



YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PhD THESIS

**EVALUATION OF THE RELATIONSHIP BETWEEN
THE STABILITY OF FEATURE SELECTION
TECHNIQUES AND CLASSIFICATION
PERFORMANCE IN DATA MINING**

MUSTAFA BÜYÜKKEÇECİ

THESIS ADVISOR: PROF. MEHMET CUDİ OKUR, PhD

DOCTOR OF PHILOSOPHY

IN

COMPUTER ENGINEERING

PRESENTATION DATE: 26.08.2019

BORNOVA / İZMİR
AUGUST 2019

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of the Doctor of Philosophy.

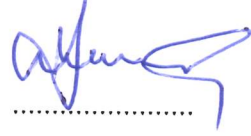
Jury Members:

Signature:

Prof. Dr. Mehmet Cudi OKUR
Yaşar University



Assoc. Prof. Dr. Murat KOMESLİ
Yaşar University



Asst. Prof. Dr. Korhan KARABULUT
Yaşar University



Asst. Prof. Dr. Mete EMİNAĞAOĞLU
Dokuz Eylül University



Asst. Prof. Dr. Samsun M. BAŞARICI
Adnan Menderes University



Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

EVALUATION OF THE RELATIONSHIP BETWEEN THE STABILITY OF FEATURE SELECTION TECHNIQUES AND CLASSIFICATION PERFORMANCE IN DATA MINING

Büyükkeçeci, Mustafa

PhD, Computer Engineering

Advisor: Prof. Mehmet Cudi OKUR, PhD

August 2019

Each year the amount of data produced and stored increases exponentially. This significant increase in both datasets and dataset sizes adversely affects data analysis techniques and algorithms, results in the production of complex models, performance losses and increased computational costs. Various data preprocessing techniques, such as feature selection, have been developed to prevent and overcome these problems. Feature selection, which is a data size (dimension) reduction technique, is used to improve analysis quality, efficiency and generalization capacity of classifiers, to reduce computational costs and to create simple and understandable models that have high classification or clustering accuracy. Besides the classification or clustering performances of the feature subsets obtained by the feature selection algorithms, stability, i.e., robustness, of the feature selection algorithm should also be tested. Stability is the measure of the sensitivity of the feature selection algorithm against the changes (perturbations) made on the training set. Algorithm with low sensitivity, i.e., a stable algorithm, produces the same or very similar results (feature subsets or ranks) after each change done in the training set, whereas algorithm with high sensitivity, i.e., an unstable algorithm, produces different results after each change. Since the results produced by an unstable algorithm will be variant, it makes it difficult to select the result set (feature set) to be used in building classification models and to establish the relationship between inputs and outputs. This undermines trust in the feature selection algorithm. Therefore, algorithm stability is an important success criterion for feature selection algorithms. In this thesis, a total of seven filter (T-Test, Bhattacharyya, Wilcoxon, ROC, Entropy, ReliefF and Decision Tree Ensemble) and two sequential (Sequential Forward Feature Selection (SFS) and Sequential Backward Feature

Selection (SBS)), or wrapper, feature selection algorithms, twelve stability measures, three classifiers and seven real-world datasets were used to determine and interpret the relationship between feature selection algorithm stability and classification performance.

Key Words: feature selection, supervised feature selection, selection algorithm stability, stability measures, classifiers, stability and classification performance



ÖZ

VERİ MADENCİLİĞİNDE ÖZNETELİK SEÇİM TEKNİKLERİNİN KARARLILIKLARI VE SINIFLANDIRMA PERFORMANSLARI ARASINDAKİ İLİŞKİNİN DEĞERLENDİRİLMESİ

Büyükkeçeci, Mustafa

Doktora Tezi, Bilgisayar Mühendisliği

Danışman: Prof. Dr. Mehmet Cudi OKUR

Ağustos 2019

Her yıl üretilen ve depolanan veri miktarı üstel olarak artmaktadır. Hem veri kümeleri hem de veri kümesi boyutlarındaki yaşanan bu önemli artış, veri analizi tekniklerini ve algoritmalarını olumsuz yönde etkileyerek karmaşık modellerin üretilmesine, performans kayıplarına ve artan hesaplama maliyetlerine neden olmuştur. Bu problemlerin önlenmesi ve üstesinden gelinmesi için, Öznitelik seçimi gibi, çeşitli veri ön işleme teknikleri geliştirilmiştir. Boyut küçültme (indirgeme) tekniği olan öznitelik seçimi, sınıflandırıcıların analiz kalitesini, verimliliğini ve genelleme kapasitesini geliştirmek, hesaplama maliyetlerini azaltmak ve yüksek sınıflandırma veya kümeleme doğruluğuna sahip basit ve anlaşılabilir modeller oluşturmak için kullanılır. Öznitelik seçim algoritmaları tarafından elde edilen öznitelik altkümelerinin sınıflandırma veya kümeleme performanslarının yanı sıra, öznitelik seçim algoritmasının kararlılığı veya sağlamlığı da test edilmelidir. Kararlılık, öznitelik seçim algoritmasının eğitim setinde yapılan değişikliklere karşı hassasiyetinin ölçüsüdür. Düşük hassasiyete sahip algoritma, yani kararlı bir algoritma, eğitim kümesinde yapılan her değişiklikten sonra aynı veya çok benzer sonuçlar (öznitelik altkümeleri veya sıraları) verirken, yüksek hassasiyete sahip algoritma, yani kararsız bir algoritma, her değişiklikten sonra farklı sonuçlar verir. Kararsız bir algoritma tarafından üretilen sonuçlar değişken olacağından, sınıflandırma modellerinin oluşturulmasında kullanılacak sonuçların (öznitelik kümesinin) seçilmesini ve girdi ve çıktılar arasındaki ilişkinin kurulmasını zorlaştırır. Öznitelik seçim algoritmasına olan güveni sarsar. Bu nedenle, algoritma kararlılığı öznitelik seçim algoritmaları için önemli bir başarı kriteridir. Bu tezde kararlılık ile sınıflandırma performansı arasındaki ilişkiyi belirlemek ve yorumlamak için toplam yedi filtreleyen (T-Testi,

Bhattacharyya, Wilcoxon, ROC, Entropi, ReliefF ve Karar Ağacı Topluluğu) ve iki ardışık seçim (Ardışık İleri Öznitelik Seçimi (SFS) ve Ardışık Geri Öznitelik Seçimi (SBS)), veya sarmalayan, öznitelik seçimi algoritması, on iki kararlılık ölçüsü, üç sınıflandırıcı ve yedi gerçek dünya veri kümesi kullanılmıştır.

Anahtar Kelimeler: öznitelik seçimi, gözetimli öznitelik seçimi, seçim algoritması kararlılığı, kararlılık ölçüleri, sınıflandırıcılar, kararlılık ve sınıflandırma performansı



ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Mehmet Cudi Okur, PhD, and members of the thesis monitoring committee, Assoc. Prof. Murat Komesli, PhD, and Asst. Prof. Mete Eminağaođlu, PhD, for their valuable help and contributions in shaping the thesis. I am also grateful to Asst. Prof. Korhan Karabulut, PhD, and Asst. Prof. Samsun M. Bařarıcı, PhD, for their supervision and guidance. Lastly, I would like to thank my father Ferhun B y kkececi and my aunt Prof. Filiz B y kkececi, M.D., for their support and encouragement.

Mustafa B y kkececi

İzmir, 2019

TEXT OF OATH

I declare and honestly confirm that my study, titled “EVALUATION OF THE RELATIONSHIP BETWEEN THE STABILITY OF FEATURE SELECTION TECHNIQUES AND CLASSIFICATION PERFORMANCE IN DATA MINING” and presented as a PhD Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Mustafa Büyükkeçeci

Signature

.....

September 16, 2019

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGEMENTS	ix
TEXT OF OATH	xi
TABLE OF CONTENTS	xiii
LIST OF FIGURES	xvii
LIST OF TABLES	xix
SYMBOLS AND ABBREVIATIONS	xxi
CHAPTER 1 INTRODUCTION	1
1.1. Literature Review on Feature Selection and Feature Selection Stability	4
CHAPTER 2 KNOWLEDGE DISCOVERY AND DATA MINING	11
CHAPTER 3 FEATURE SELECTION	15
3.1. Feature Selection Techniques.....	16
3.2. Supervised Feature Selection Techniques	19
3.2.1. Filtering Techniques	19
3.2.2. Wrapper Techniques	20
3.2.3. Embedded Techniques	21
3.2.4. Hybrid Techniques	22
3.2.5. Ensemble Techniques	23
3.3. Problems Encountered in Supervised Feature Selection	24
CHAPTER 4 SOURCES AND TYPES OF DATA, CLASSIFIERS AND CLASSIFIER EVALUATION	26
4.1. Sources and Types of Data.....	26
4.2. Classification Algorithms.....	29
4.2.1. K-Nearest Neighbor (K-NN)	29
4.2.2. Support Vector Machine (SVM).....	30
4.2.3. Naïve Bayes (NB)	31

4.2.4. Error Correcting Output Codes (ECOC)	32
4.2.5. Artificial Neural Network (ANN)	33
4.2.6. Decision Tree (DT).....	34
4.2.7. Linear Discriminant Analysis (LDA)	36
4.3. Evaluation of Classification Performance	36
4.3.1. Confusion Matrix.....	37
4.3.2. ROC (Receiver Operating Characteristics) Curve.....	38
4.4. Validation and Cross-Validation Techniques	40
4.4.1. Holdout	40
4.4.2. Resubstitution	41
4.4.3. Random Subsampling.....	41
4.4.4. Bootstrapping (0.632 Bootstrap)	41
4.4.5. K-Fold Cross-Validation (K-Fold CV).....	42
4.4.6. Leave-One-Out Cross-Validation (LOOCV <i>or</i> Rotation Estimation).....	42
4.4.7. Stratified (<i>or</i> Proportional) Holdout and K-Fold CV	43
4.5. Sampling.....	43
CHAPTER 5 STABILITY IN FEATURE SELECTION	46
5.1. Stability Measurement in Supervised Feature Selection.....	47
5.2. The Properties of Stability Measures	49
5.3. Types of Stability Measures.....	50
5.3.1. Set-Based (<i>or</i> Index-Based) Stability Measures	50
5.3.2. Weight-Based Stability Measures.....	51
5.3.3. Rank-Based Stability Measures	52
5.3.4. Frequency-Based Stability Measures	54
5.4. Computational Complexities of Stability Measures	54
5.5. Open Topics on Selection Algorithm Stability	55
CHAPTER 6 FEATURE SELECTION ALGORITHMS USED IN THE EMPRICAL STUDY.....	57
6.1. Statistical Feature Selection.....	57
6.1.1. Two Sample T-Test	57
6.1.2. Bhattacharyya Distance	58

6.1.3. Wilcoxon Rank-Sum Test (<i>or</i> Mann-Whitney U Test).....	58
6.1.4. ROC (Receiver Operating Characteristic) Curve Test.....	59
6.1.5. Entropy (<i>or</i> Shannon Entropy) Test.....	59
6.2. Relief and the ReliefF Algorithms	60
6.2.1. Methods Used to Determine the Ideal K Value	63
6.3. Feature Selection Through Ensemble Learning	65
6.4. Sequential Feature Selection	67
6.5. Hyperparameter Optimization.....	70
CHAPTER 7 EMPRICAL STUDY.....	72
7.1. Empirical Study on Filter Algorithms	74
7.2. Empirical Study on Wrapper Algorithms.....	77
7.3. Classification Performance Evaluation	81
CHAPTER 8 DISCUSSION OF EMPIRICAL STUDY RESULTS.....	84
8.1. Discussion of Stability Measures	84
8.2. Discussion of Stability and Classification Performance	86
8.2.1. Results of Filter Algorithms.....	95
8.2.2. Results of Wrapper Algorithms	96
8.3. Conclusions and Future Work.....	97
REFERENCES	100
APPENDIX 1 — K and Hyperparameters Values	107
APPENDIX 2 — Summary (Descriptive) Statistics Results	110
APPENDIX 3 — Classification Performances.....	114
APPENDIX 4 — Results of Filter Algorithms.....	115
APPENDIX 5 — Results of Wrapper Algorithms	116

LIST OF FIGURES

Figure 2.1. Stages of “Knowledge Discovery in Databases” (KDD).....	12
Figure 2.2. Data Mining Functionalities	13
Figure 3.1. General Framework of Supervised Feature Selection	16
Figure 3.2. Classification (Taxonomy) of Feature Selection Techniques.....	17
Figure 3.3. Classification of Supervised Feature Selection Methods	19
Figure 4.1. An Overview of Chapter 4.....	26
Figure 4.2. Classification of Data Sources.....	27
Figure 4.3. Classification of Data Types.....	28
Figure 4.4. An Example of Two Different ECOC Matrices (Output Codes) Created for a Four-Class Problem.....	32
Figure 4.5. The ROC Space and the ROC Curve for a Classifier	39
Figure 4.6. Illustration of Random, Stratified and Cluster Sampling Techniques	44
Figure 5.1. Classification of Feature Selection Methods in Terms of Result Representations	47
Figure 5.2. Comparison Results Represented by Matrices	50
Figure 5.3. Comparison Results Represented by a Vector.....	50
Figure 6.1. Pseudocode of the Relief Algorithm	61
Figure 6.2. Pseudocode of the ReliefF Algorithm	62
Figure 6.3. Feature Weights vs. K-Values Graph	64
Figure 6.4. Pseudocode of the SFS Algorithm.....	68
Figure 6.5. Pseudocode of the SBS Algorithm	68
Figure 6.6. Pseudocode of the BDS Algorithm	69
Figure 7.1. General Framework of the Empirical Study	74
Figure 7.2. Screenshot of T-Test Results After Three Runs	76
Figure 7.3. Screenshot of Discriminant Classifier Results After Three Runs.....	79
Figure 7.4. Screenshot of the Confusion Matrices	82

Figure 7.5. ROC Curves and AUC Values Created for Two Different Feature Subsets.....83



LIST OF TABLES

Table 4.1. Confusion Matrix	37
Table 4.2. List of Various Classification Model Evaluation Metrics.....	38
Table 4.3. Interpretation of AUC Values	39
Table 5.1. List of Stability Measures Classified According the Representation of the Output of the Feature Selection Technique	48
Table 5.2. List of Set-Based Stability Measures	51
Table 5.3. Pearson’s Correlation Coefficient, Spearman’s Rho and Kendall’s Tau	53
Table 5.4. Formulas of Canberra and Weighted Canberra Distance.....	53
Table 5.5. Computational Complexities of the Stability Measures.....	55
Table 7.1. List of Datasets Taken from the UCI and Kaggle Machine Learning Repository and Their Properties	73
Table 7.2. Minimum Number of Samples Taken from Datasets in Each Run (Iteration).....	75
Table 7.3. An Example of Average Stability Results Table of All Filtering Techniques on Vehicle Dataset	77
Table 7.4. List of Learners for the ECOC and the Ensemble of Learners	78
Table 7.5. An Example of Predictive Accuracy Table of the SFS Algorithm Used with LDA, K-NN, and NB Classifiers on Breast Cancer Dataset	80
Table 7.6. An Example of Average Stability Results Table of the SFS Algorithm Used with LDA, K-NN, and NB Classifiers on Breast Cancer Dataset	81
Table 8.1. List of Selected Filter and Wrapper Algorithms	88
Table 8.2. Classification Performances of Selected Feature Subsets from (a) Abalone, (b) Australian, (c) Breast Cancer, (d) Sat, (e) Seeds, (f) Vehicle and (g) Wine Datasets	90



SYMBOLS AND ABBREVIATIONS

Symbols	Explanation
$\mathbf{F}, \vec{\mathbf{F}}$	Feature set — Vector of features
\mathbf{F}_i	Feature i
$\mathbf{S}, \vec{\mathbf{S}}$	Selected feature subset — Vector of selected feature subsets
\mathbf{S}_i	Selected feature i
$\mathbf{W}, \vec{\mathbf{W}}$	Weight of features — Vector of feature weights
\mathbf{W}_i	Weight of feature i
$\mathbf{C}, \vec{\mathbf{C}}$	Set of classes — Vector of classes
\mathbf{C}_i	Class i
\mathbf{D}, \mathbf{N}	Dataset (population)
\mathbf{d}	Number of features
\mathbf{n}	Number of samples
\mathbb{R}^n	n -Dimensional feature space
\mathbb{R}^x	x -Dimensional feature space
\mathbb{R}^y	y -Dimensional feature space
$\mathbf{argmax}(\mathbf{P})$	Maximization of target function P
$\mathbf{argmin}(\mathbf{P})$	Minimization of target function P
ℓ	Length of code
$\mathbf{f}(\varphi)$	Activation function
\mathbf{b}	Bias
\mathbf{Acc}, \mathbf{a}	Accuracy
\mathbf{r}	Correlation coefficient
\mathbf{A}	Upper triangular matrix
\mathbf{S}	Symmetric matrix
$\vec{\mathbf{R}}$	Result vector
\mathbf{ff}	Fitness function
ρ	Rho
τ	Tau

Abbreviations	Explanation
FSA	Feature Selection Algorithm
SFS	Sequential Forward Selection
SBS	Sequential Backward Selection
BDS	Bidirectional Selection/Search/Elimination
LDA	Linear Discriminant Analysis
NB	Naïve Bayes
K – NN	K Nearest Neighbor
SVM	Support Vector Machine
ECOC	Error Correcting Output Code
DTE	Decision Tree Ensemble
PPV, NPV	Positive or Negative Predictive Value
TP, TN	True Positive and True Negative
FP, FN	False Positive and False Negative
ROC	Receiver Operating Characteristic
AUC	Area under the ROC Curve
CV	Cross-Validation
LOOCV	Leave One Out Cross-Validation
SRS	Simple Random Sampling
SRSWR	Simple Random Sampling with Replacement
SRSWOR	Simple Random Sampling without Replacement
HD	Hamming Distance
JI	Jaccard Index
CI	Cosine Index (Similarity)
SDC	Sorensen—Dice Coefficient
OC	Overlapping Coefficient
LM	Lustgarten’s Measure
NM	Nogueira’s Measure
CD	Canberra Distance
WCD	Weighted Canberra Distance
PCC	Pearson’s Correlation Coefficient
SRCC	Spearman’s Rank Correlation Coefficient
KRCC	Kendall’s Rank Correlation Coefficient
Corr.	Correlation
Coef.	Coefficient
ER	Error Rate
F1	F1 Score
Sens.	Sensitivity
Spec.	Specificity
Opt.	Optimized

CHAPTER 1

INTRODUCTION

Increased computer usage has significantly increased both the data production rate and the amount of data stored on the computer. The datasets that contain qualitative and/or quantitative values obtained by various ways such as observation, experiment and measurement may contain valuable but hidden information that can be used for various purposes. This hidden information can be revealed by processing, i.e., analyzing, the datasets. This process, in which unprocessed, i.e., raw, data is made valuable, is called the “Knowledge Discovery in Databases” or shortly KDD. Steps involved in the KDD process are preparation, dataset acquisition, data preprocessing, transferring data to the data warehouse, data mining, presentation, and evaluation. However, datasets may contain several problems that may adversely affect this process. At this point, data preprocessing techniques are used to identify and fix these problems. One of these techniques is feature selection.

The increasing number of samples, i.e., records, and features contained in the datasets also increased the need and interest in the choice of a size reduction technique. Today, datasets belonging to various disciplines such as medicine, biology, chemistry, economy, logistics, astronomy and actuary can contain thousands of features. However, these datasets might include noisy, redundant (duplicate or adds no extra information) and/or irrelevant (uncorrelated with the class tag or adds no useful information) features that do not contribute to the analysis process. Such features, which do not contribute to the model to be created, are detected and eliminated with the help of feature selection techniques. By discarding the irrelevant features, the time needed for analysis, processing power and the amount of memory is reduced, simple models with high generalization performances are aimed to be achieved.

Feature selection is essentially an optimization problem. The purpose of this problem is to find the subsets of features $S = \{S_i | i = 1, 2, 3, \dots, m\}$ that maximize a target function, P , $argmax(P)$, from the feature set $F = \{F_i | i = 1, 2, 3, \dots, n\}$, that belongs to a dataset, D , by satisfying $S \subseteq F$ and $m < n$ conditions. For this reason, feature selection algorithms are expected to find the smallest possible feature subset that provides the highest possible accuracy rate by eliminating noisy, redundant and

irrelevant features. Another desirable property expected from the feature selection algorithms is that they have a stable structure. While the selection algorithms that have stable structures produce very similar or same results (feature subset(s)) after the changes, i.e., perturbations, made in the training sets, while the ones with unstable structures produce results that show the lowest similarity or completely different from each other. Varying results produced by unstable algorithms mislead the user in detecting the final result set and undermine the trust in the algorithm and analysis process.

The stability of a selection algorithm is the measure of robustness and the sensibility of the algorithm against the changes in the training set. The algorithms which produce a low cardinality subset or have high stability, but poor classification or clustering performance, make it difficult to obtain successful results from the analysis process. Therefore, it is inaccurate to evaluate both the feature selection and the selection algorithm stability regardless of the classification or clustering performance. As can be seen, it is difficult to distinguish between these concepts and see them independently.

Aim of the thesis: When the studies in the literature are examined, it is seen that the relation between the selection algorithm stability and the classification performance of the selected features, i.e., selection algorithm performance, is not studied sufficiently. The main purpose of this thesis is to inquire into whether such a relationship exists. For this purpose, filtering and wrapper algorithms are used. In this way, the relationship between stability and classification performance is tested among selection algorithms of different types instead of a single type of selection algorithm. According to the test results, which selection technique is more stable and successful in terms of classification, whether there is a relationship between classification and feature selection stability, if so, how it can be interpreted, how to compare different stability measures, and the effect of different classifiers on the stability of wrapper algorithms have been observed.

The contributions of the thesis are as follows:

1. An extensive literature review on feature selection, feature selection stability, and stability metrics,
2. A fair comparison of different feature selection algorithms that belong to filter (T-Test, Bhattacharyya, Wilcoxon Rank-Sum, ROC, Entropy and ReliefF), embedded

and ensemble (Decision Tree Ensemble) and wrapper (SFS and SBS) families by performing an extensive set of experiments on real-world datasets.

3. A fair comparison of various stability metrics,
4. An experimental framework to inquire into the relationship between the stability and the classification performance of the selected features, and
5. A data preprocessing step using descriptive statistics to summarize the datasets statistically and to find out the underlying data-driven factors that can cause feature selection instability and reduced performance metric values.

Structure of the thesis: This thesis is divided into eight chapters. The first chapter introduces the feature selection process and a brief literature review of both feature selection and feature selection stability. In the second chapter, Knowledge Discovery in Databases and Data Mining, concepts are briefly explained to make it easier to express where and how the feature selection is used. Chapter three provides material on feature selection, feature selection according to the machine learning type and supervised feature selection techniques. Then, the chapter is completed by mentioning various problems encountered in the supervised feature selection. The fourth chapter, summarizes the data sources, data types, various classification and regression algorithms that are used to select features, methods used to assess the performance of the classifiers and various validation and sampling methods used in the literature. In the fifth chapter, selection algorithm stability, which is the basis of this thesis, stability measurement in the supervised feature selection, properties of stability measures, types and computational complexities of stability measures and the problems encountered in the measurement of stability are discussed in detail. In chapter six, filter and wrapper supervised feature selection algorithms used in the empirical study and the Bayesian Hyperparameter Optimization are briefly argued. In the seventh chapter, first, the properties of the datasets used during the empirical studies are debated. Then, the empirical study framework and how filter and sequential feature selection algorithms were used during experiments are explained in detail. In the last chapter, observed drawbacks of stability measures and the effect of hyperparameter optimization over stability measures are argued. Later, the main concern of the thesis, the relationship between feature selection stability and classification, is summarized and findings, the

contribution of the thesis and concluding remarks are presented. Finally, an outlook of future works and some comments are stated.

1.1. Literature Review on Feature Selection and Feature Selection Stability

Feature selection and extraction are two different data preprocessing techniques, which are often confused. As outlined in the research study of Khalid et al. (2016), and the “Chapter 6” of the book by Abe (2005), feature selection is used for data size reduction while feature extraction is used for data transformation. The feature selection takes place in three different ways according to the data used. These are: supervised feature selection for labeled, unsupervised feature selection for unlabeled and semi-supervised feature selection for both labeled and unlabeled data. In the study of Chin et al. (2015), the purpose of the feature selection, feature selection process and feature selection techniques are explained in general. Also, supervised, unsupervised and semi-supervised feature selection algorithms used for gene selection and the references of the studies in which they were used were tabulated. In addition to this study, research studies of Chandrashekar and Şahin (2014), and Li et al. (2016) and the book chapter of Wang et al. (2016) can be examined to obtain general information about the feature selection process.

Studies on feature selection often concentrate on supervised techniques. Therefore, various sources, e.g., books, articles, theses, projects, source codes, etc., are related to supervised feature selection. One of these studies is the literature survey of Kumar and Munz (2014), which provides a comprehensive overview of the subject. In this study, the concept of relevant and irrelevant features, the steps of the feature selection process, the developments in the feature selection and the areas where the feature selection is used were transferred to the reader through real-world examples. Also, authors classify supervised feature selection techniques according to the type of search (sequential, exponential and random), selection type (filter or feature ranker, wrapper and embedded) and data mining tasks (classification and clustering) that used to create feature subsets and explained in general terms. A study on the same topic, Dash and Lui (1997), performed feature selection using a set of synthetic datasets and supervised feature selection techniques. The authors also grouped and examined the feature selection techniques by type (feature subset generation/creation) (complete, heuristic

and random) and by evaluation criteria (distance, information gain, correlation, consistency, and classifier error rate measures).

In this thesis, supervised feature selection techniques were examined in five different classes such as filter, wrapper, hybrid, embedded and ensemble. Filtering techniques are the most preferred selection technique since they have low computing cost, scalable structure and are easy to implement. Lazar et al. (2012) represented one of the research studies conducted on filtering techniques. In general, the authors discussed how to solve the problem of feature selection and how to solve this problem by using filtering techniques in the bioinformatics field. In the study, the filter techniques were divided into two classes as performing space search using sorting and optimization algorithms. The authors also presented the methods and metrics of these classes to the reader using tables. However, in this study, algorithms have not been evaluated empirically. Instead, an extensive literature about the subject provided. As filtering techniques do not perform feature selection, they sort the features with the help of an evaluation criterion such as distance, statistical testing, information gain, e.g., Fast Correlation-Based Filtering (FCBF) (Yu and Liu, 2003), probabilistic selection (Liu and Setiono, 1996), Laplacian Score (He et al., 2005), Pearson Chi-Square (Biesiada and Duch, 2007), Bhattacharya Distance (Guorong et al., 1996), etc. For this reason, the selection process takes place according to a user-set, i.e., defined, threshold value, i.e., selection criterion.

The wrapper algorithms perform the selection process by using a classifier. The study was done by Kohavi and John (1997) related to wrapper algorithms, primarily mentions feature subset selection problems, relevant and irrelevant feature concepts and finding optimal features. Afterward, the authors summarized the filter and wrapper algorithms and mentioned Hill Climbing and Best-First methods which can be used as a forward search strategy in the wrapper algorithms. The study also included a topic related to the overfitting problem, which is one of the biggest problems of wrapper algorithms. Finally, in the experimental study, using synthetic and real datasets, algorithms were compared with each other and the results were tabulated. Another problem with wrapper algorithms is that they are slow. The main reason for this problem is that the search results are evaluated by a classifier specified in each iteration. However, Wang et al. (2015), argued that they had accelerated the wrapper selection technique by embedding the K-NN classifier in it. To prove these theses, they also used

three wrapper algorithms (SFS, IWSS, and IWSSr) and compared the algorithms experimentally using microarray datasets before and after acceleration.

Hybrid selection methods are obtained by combining two different selection methods. The studies were done by Hsu et al. (2011) and Sebbana and Nock (2001), hybrid selection methods are created by using filter and wrapper algorithms and tested. Among the studies in the field of supervised feature selection, embedded and hybrid methods have the least share. In the doctoral study of Saeys (2004), embedded selection methods Weighted Naive Bayes and Linear SVM (Support Vector Machine) algorithms are used to classify nucleic acid sequences. In the study of Peng et al. (2005) presented a two-stage feature selection algorithm by combining mRMR (Minimum Redundancy Maximum Relevance) with other wrappers to form a hybrid selection strategy and compare its performance with different selectors and using three different classifiers and four datasets. The biggest problem of the embedded methods is that the results obtained are dependent on the classifier and the computational costs increases as the data size become larger.

In the supervised feature selection, most of the studies performed on filter and ensemble methods. It is possible to create ensemble methods in different ways, but they are generally created by combining the results obtained from different feature selection algorithms with a consortium function. Thus, it is aimed to avoid the problem of the appropriate selection algorithm selection and to obtain the subset of the components with high clustering or classification performance. An ensemble model was built in the study of Seijo-Pardo et al. (2015). The authors used Chi-Square, Information Gain, mRMR and ReliefF filtering algorithms, and SVM-RFE and FS-P embedded algorithms on the datasets obtained from UCI machine learning repository. Since all the selection algorithms mentioned in the study were filtering algorithms, the results were first combined with an integration algorithm called SVM-Rank and then tested with the SVM classifier. Tuv et al. (2009) proposed a novel feature selection ensemble algorithm that is based on the Tree-Based Ensemble named ACE (Artificial Contrasts with Ensembles). The authors compared the performance of the algorithm with competitive algorithms such as RFE (Recursive Feature Elimination), Relief, CFS (Correlation-Based Feature Selection), CFS-gen (CFS with Genetic Search) and FCBS (Fast Correlation Based Filter).

For feature selection algorithms being stable is a desirable quality as much as the features they have chosen having a high classification performance. There are various measures in the literature used to measure selection algorithm stability. In the studies of Awanda et al. (2012), Khoshgoftaar et al. (2003), and Chelvan and Perumal (2017), a list of several stability measures were presented to the reader along with the studies that they were used. In the study conducted by Kuncheva (2007), in addition to the proposed stability measure (Kuncheva Index), the properties that any stability measure should have are also identified. In the doctoral study of Noguera (2018), primarily the properties that the stability measures must have were listed in five items then fifteen measures in the literature have been tested to see how many of these properties that they provide. The author then argued that there are no measures that provide all the properties except the one she proposed.

Studies have been done related to the algorithm and dataset-driven factors that cause instability, the solutions that can be used to solve the problem of instability and the determination of the relationship between stability and classification performance in the literature. In the doctoral study of Alelyani (2013), the reasons that cause the algorithm instability were investigated and argued that the problem was originated from data. The author found that the noisy datasets caused both poor learning performance and instability, and used a method called SLRMA (Supervised Noise Reduction via Low-Rank Matrix Approximation) to reduce noisy data. In the study, for the unlabeled data, a novel method named Local SVD (Local Singular Value Decomposition) was also mentioned. Han (2007), proposed a theoretical framework in his doctoral study to explain the relationship between stability and accuracy. Through this framework, the author claims that the relationship between the stability of selection algorithms and the accuracy of the selected features depends on the number of samples. Also, an empirical variance reduction framework that increases the algorithm stability used when the number of samples is low proposed and tested on synthetic and real datasets using the SVM-RFE (Recursive Feature Elimination for Support Vector Machines) and ReliefF algorithms. To measure the classification performance, SVM and K-NN (1-NN) classifiers, and to measure stability Kuncheva Index was used.

Feature selection has a wide range of applications. One of these areas is the field of health practices as in the doctoral study carried out by Kamkar (2016). In the related

thesis study, on the electronic medical records, Tree LASSO (Least Absolute Shrinkage and Selection Operator), Predictive Grouping Elastic Net (pg-EN), Covariance LASSO (C-LASSO), L1-Norm SVM, Covariance SVM (C-SVM) and Graphical SVM (graph-SVM) methods were used. The author mentioned nearly all stability and classification measures and in the experimental stage used Jaccard Index and Spearman's Rank Correlation Coefficient as stability measures and Positive or Negative Predictive Value, sensitivity, specificity, F1-Score and AUC for classification performance assessment.

In the study of Haury et al. (2011), aiming to determine the relationship between algorithm stability and classification performance, the effect of feature selection techniques on accuracy, stability and interpretability of molecular signatures were tested using four different datasets on breast cancer, T-Test, Entropy, Bhattacharyya, Wilcoxon Rank-Sum, SVM-RFE, GFS, LASSO and Elastic Net feature selection techniques and Nearest Centroids (NC), K-NN, Linear SVM, Linear Discriminant Analysis (LDA) and Naïve Bayes classifiers. To measure the classification performance, 10-Fold Cross-Validation (10-Fold CV) and the area under the ROC Curve (AUC) were used. Stability measurement was determined by the similarity of the molecular signatures generated by the methods.

Wang et al. (2011) used nine real-world software measurement dataset and seven filter-based selection algorithms: Chi-Square, Information Gain, Gain Ratio, ReliefF, Weighted ReliefF (weights features by distance), Symmetrical Uncertainty and Signal-to-Noise (S2N) and the Logistic Regression classifier for performance testing in their study. The classification performance was measured by the area under the ROC Curve (AUC) and one-way ANOVA test and the algorithm stability was tested with Kuncheva Index.

Drotár and Smékal (2014) used five feature selection algorithms on four micro-series and two biomedical datasets of Parkinson's disease. The stability of the feature selection algorithms, including Tree-Based Feature Selection Ensemble, LASSO, mRMR (minimum redundancy, maximum relevance) and Iterative Relief (iRelief) were measured by the Hamming Distance stability measure. The classification performance of the selected features was performed using the AdaBoost algorithm and Matthews Correlation Coefficient (MCC). In the study of Kalousis et al. (2007), filtering algorithms such as Chi-Square, Symmetrical Uncertainty, ReliefF and SVM-

RFE were used on the datasets related to proteomics, genomics and text mining. The algorithm stability was determined by Pearson's Correlation Coefficient, Spearman's Rank Correlation Coefficient, and Tanimoto Distance. The authors also commented on the relationship between algorithm stability and classification performance under a separate title ("Stability and Classification Performance").

In the study of Yang et al. (2016), in which the feature selection algorithms and ensemble feature selection algorithms were compared, factors affecting the stability of selection algorithm and ensemble feature selection subjects were argued. In addition to these, various questions that are the foundation of the study, such as what is the ensemble feature selection, how to combine the obtained results, how the performance is measured were answered under the related titles. The authors also mentioned Spearman's Rank Correlation Coefficient, Hamming and Jaccard Distance and Kuncheva Index stability measures that they used for stability measurement in detail. Lastly, the individual classification performances of the algorithms were compared with the performances they show when they form an ensemble method. In the study of Saeys et al. (2008), they argued that the ensemble feature selection methods have higher stability and classification performance, especially in high-dimensional datasets with a small sample and higher feature quantities. Symmetrical Uncertainty, Relief, SVM-RFE, and Random Forest feature selection algorithms were used individually first on real-world datasets of six different diseases then used by creating an ensemble method. Selection algorithm stability measurement was done using Pearson's Correlation Coefficient, Spearman's Rank Correlation Coefficient, and Jaccard Index, and classification performances were measured using SVM, Random Forest, and K-NN classifiers.

In the study of Dunne et al. (2002), instability experienced during feature subset selection was tried to be solved by wrapper algorithms using a sequential search such as Sequential Forward Selection, Sequential Backward Selection/Elimination, and Random Hill Climbing. In this study, the reasons for instability were also explained and a novel wrapper algorithm that uses parallel and genetic algorithms as search techniques have been proposed. The authors used K-NN as the classifier and Average Hamming Distance (AHD) and Average Normalized Hamming Distance (ANHD) for the stability measurement.

Succinctly, in supervised feature selection studies univariate and multivariate filtering techniques such as ReliefF, Information Gain, Symmetrical Uncertainty, and Chi-Square are mostly used. As for induction or performance testing, SVM and K-NN classifiers are preferred more than other algorithms. Robustness of the selection algorithms is mostly measured using Jaccard and Kuncheva Index, Hamming and Canberra Distances and Pearson's and Spearman's Rank Correlation Coefficients.



CHAPTER 2

KNOWLEDGE DISCOVERY AND DATA MINING

In order to understand the purpose and importance of feature selection, first, it is necessary to mention the terms “Knowledge Discovery in Databases” and “Data Mining”. The fact that computers become an indispensable part of our daily life has led to an increase in data production rate and thus the amount of data stored in databases. A growing pile of data has led to the concept of “Knowledge Discovery in Databases” or shortly KDD, which is transmuting the data into a valuable form through analysis. In other words, it is a set of operations made to add meaning to the raw data. In this thesis, the KDD process is divided into six phases and is illustrated in Figure 2.1.

1. Preparation: In this step, understanding the problem, collecting the necessary pre-information, determining the purpose, selecting the algorithm and methods that will help to reach the goal, are realized. Thus, a process management plan is created to save resources, labor and time.

2. Obtaining the dataset: The next step is to obtain the appropriate dataset to be used for analysis. Data are obtained from two different sources, primary and secondary. Primary data resources consist of data from which the researcher personally acquired through experiment, observation, and surveys. Secondary data sources are data that are collected and compiled by others in various media and formats. Detailed information on data sources and types is given in the 3rd chapter.

3. Data preprocessing: Data preprocessing is a step towards improving the quality of the knowledge discovery process. In this step, out of date, missing, excessive, inconsistent and noisy data are extracted and cleared of flaws by various approaches. Also, data from different sources are combined for to be compatible with one another. Data conversion, data reduction and data discretization are also performed at this stage. The importance of this step can be explained by the axiom of garbage in, garbage out. The real-world data obtained from internal and external sources include the defects mentioned in this subtitle. The use of defective data in the analysis process results in misleading and inaccurate analysis results. For this reason, the more data is cleared from its defects, the higher quality and accurate the obtained results will be.

4. Transferring data to databases and data warehouses: In this step, the data that is cleared from its defects, combined and transformed is transferred to specialized databases in the relational structure called the data warehouse. The data to be used for analysis, reporting, query and similar purposes are stored here and updated periodically.

5. Data mining: In this step, the selected data mining algorithm that has the property of prediction (such as classification, prediction, time series analysis) or description (such as clustering, summation, association rules) is applied to the appropriate dataset which obtained from the data warehouse and patterns (model, rule) are determined.

6. Interpretation and evaluation: At the last stage, the results (patterns) are presented to the end-user by various presentation techniques and interpreted and evaluated according to the criteria such as validity, innovation and usefulness. Also, action plans (course of action) are determined in order to reach the objective.

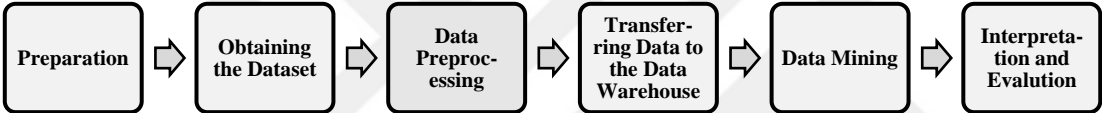


Figure 2.1. Stages of “Knowledge Discovery in Databases” (KDD)

During the KDD process, raw data is processed to reveal hidden, previously unknown, useful, understandable and interpretable patterns and to turn into information. Data mining and its functions are located at the center of this process. Data mining is a data analysis method that is formed by using a combination of different disciplines such as statistics, machine learning, databases, mathematical modeling, artificial intelligence, information, and management systems. It has two functions, as depicted in Figure 2.2. The first function is the description. There may be relations in the high-volume databases that are difficult to detect and find. For this reason, the description (depiction) function of data mining is used to identify structures (patterns) hidden within the database. The second function of data mining is the prediction. This function predicts and models the future values using previously known (existing) values. Data mining is used for a variety of purposes, such as clustering, classification, prediction, the discovery of association rules, detection of anomalies or outliers, summarization and regression in many areas of high-volume data.

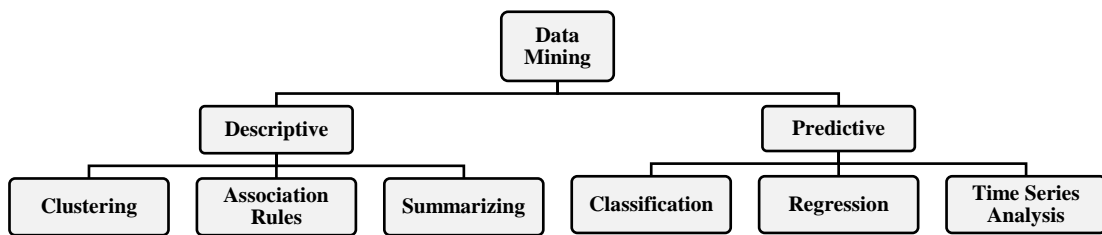


Figure 2.2. Data Mining Functionalities

The increasing number of features along with the amount of data adversely affect the operation of the analysis methods and techniques used in this process. For example, it causes building models that are difficult to interpret, complex, inadequate and faulty generalization capability. More than that the need for computational power and time for analysis increases. Feature selection is a data preprocessing technique which is used to prevent these problems and to improve the performance of models created by classification and clustering algorithms. Therefore, it is an important part of the data analysis process.

CHAPTER 3

FEATURE SELECTION

There may be only one, or tens, hundreds even thousands of features, either quantitative or categorical format, in the datasets. However, an excessive number of features, i.e., high dimensions, results in problems such as prolonging the analysis process, high variance, noise, a tendency to overfitting, multicollinearity, complex and unintelligible analysis results. This situation is called the curse of dimensionality. An excessive number of features increases in storage and data transfer costs. To avoid these problems, features that are noisy, redundant and irrelevant must be filtered out from the dataset. This is called data dimension (size) reduction, feature selection, variable selection or variable subset selection.

Feature selection is used to reduce the size of the datasets, to minimize the computational cost of data analysis, to obtain accurate, complete and understandable, i.e., simple and plain, analysis results, to increase scalability and to create generalized and easy-to-update models. Therefore, feature selection is used to improve the quality of classification and clustering functions of data mining. However, while all these objectives are achieved, it is aimed at the loss of information between the original dataset and the selected subset of the features is minimized or nonexistent. Because the loss of information affects the quality of data analysis negatively, it is necessary to transfer the information as much as possible.

The feature selection is occasionally confused with the feature extraction process. Although both methods are data preprocessing methods that are used to reduce the size, they are completely different from one another. Feature selection is to find the subsets of features $S = \{S_i | i = 1, 2, 3, \dots, m\}$ that maximize a target function, P , $argmax(P)$, from the feature set $F = \{F_i | i = 1, 2, 3, \dots, n\}$, that belongs to a dataset, D , by satisfying the following conditions $S \subseteq F \wedge m < n$. However, feature extraction is transforming, X dimensional dataset, \mathbb{R}^x , to a smaller size, Y , with the following conditions $\mathbb{R}^x \rightarrow \mathbb{R}^y \wedge \mathbb{R}^x < \mathbb{R}^y$ by using methods such as PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) or SVD (Singular Value Decomposition).

Feature selection algorithms generally perform the selection process in four steps. The first step is to select, i.e., generate, a candidate feature subset from the original dataset.

This step is followed by the evaluation of the selected candidate subset according to the specified evaluation criteria such as distance, correlation, misclassification rate, etc. The third step is to apply the stopping criterion. The stopping criterion is used to prevent the algorithms from entering an infinite loop and to limit the processing (run) time. In the cases where the stopping criterion is not provided, the algorithm returns to the beginning and repeats the steps. The stopping criterion may be dependent, e.g., selection continues until there is an increase in classification accuracy, or independent, e.g., selection continues until the maximum number of iterations is reached, on the evaluation criterion. The final step of the selection process, which is not a part of the selection process, is testing. Its purpose is to evaluate the performance of the selected feature subset for supervised learning through a classifier and for unsupervised learning through a clustering algorithm with the testing dataset. The general structure of the supervised feature selection process is shown in Figure 3.1.

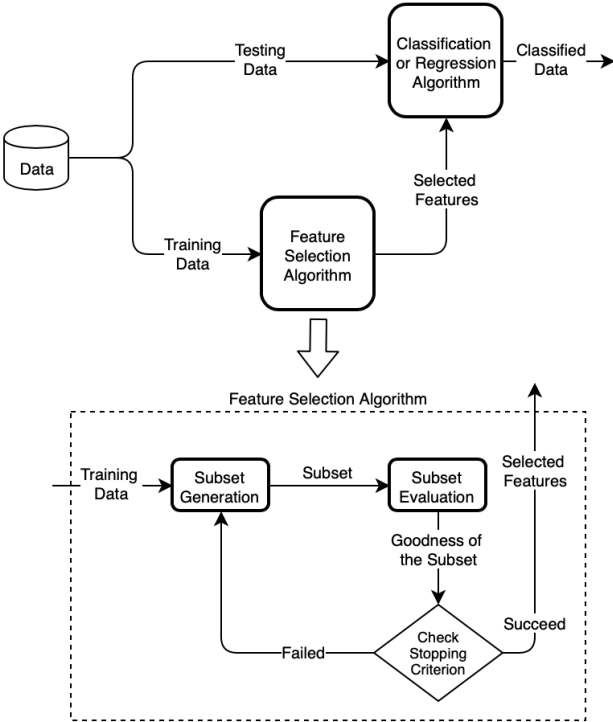


Figure 3.1. General Framework of Supervised Feature Selection

3.1. Feature Selection Techniques

Feature selection can be supervised, i.e., labeled, unsupervised, i.e., unlabeled, or semi-supervised, i.e., semi-labeled, according to the class variable, as depicted in Figure 3.2.

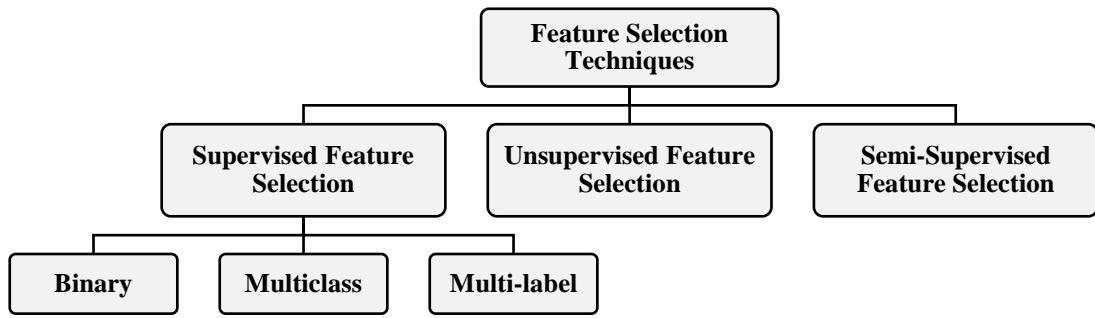


Figure 3.2. Classification (Taxonomy) of Feature Selection Techniques

- Supervised Feature Selection:** Studies in the literature are mostly concentrated on this type of feature selection. Supervised feature selection is performed on labeled datasets. Labeled data is the name given to the data containing cause-and-effect or input-output relations. In supervised feature selection, the selection algorithms consider the relationship between the features and the class label. For that, they use an evaluation criterion or a classifier. However, data labeling is a time-consuming process. Also, it is not possible to separate and label each condition encountered with certainty. Supervised feature selection can be performed on binary, multiclass or multi-label classification problems. In binary (two-class) classification problems, samples are assigned to one and only one class label. Performed tests for the detection of a particular disease, for example, diabetes, is the most typical example of a binary classification problem. The result of the test for the patient undergoing the test is either sick or healthy (not sick). In multiclass problems, the class label is more than two. However, as the binary classification problems, each sample can be assigned to one and only one class label. For example, classifying cars according to their color is a multiclass problem. Finally, in multi-label classification problems, each sample has more than one class label. A cinema movie having sci-fi, horror and adventure genres concurrently, is an example of a multi-label classification problem. There are many articles, books, compilations and electronic resources on supervised feature selection. The studies belonging to Jiliang et al. (2014), Guyon and Elisseeff (2003), and Huang (2015), can be examined for more detailed information on supervised feature selection.
- Unsupervised Feature Selection:** Unsupervised feature selection is performed on unlabeled datasets. Unlabeled data does not contain cause-and-effect or input-output relations. Therefore, no class labels that guide the selection process exist

and the process becomes more difficult than the supervised selection. In order to overcome this problem, a strategy, such as creating labels using clustering algorithms and transforming the process into something similar to supervised learning is used. Unlabeled data are often encountered because detecting data labels and labeling all data is difficult, time-consuming and cumbersome. However, the number of studies on unsupervised feature selection is small, compared to the supervised feature selection. Comprehensive research work, carried out by Salem et al. (2014), in which the unsupervised feature selection process is described in general, can be examined to obtain more detailed information on the subject.

- **Semi-supervised Feature Selection:** Semi-supervised feature selection is performed using both labeled and unlabeled data. Internet shopping sites are one of the best examples of semi-supervised data sources. They contain labeled data such as product categories, and unlabeled data such as product reviews that are written in texts. Generally, the amount of unlabeled data used in semi-supervised feature selection is more than the labeled data. However, semi-supervised feature selection algorithms, for example, graph-based, e.g., Laplacian SVM, or low-density separation, e.g., Transductive SVM (TSVM), do not take the majority into account. They are capable of processing both labeled and unlabeled samples in the same dataset. In recent years, studies on semi-supervised feature selection have increased, due to high costs of data labeling and the fact that unlabeled data can be easily obtained. One of the most recent studies is a detailed literature survey conducted by Sheikhpour et al. (2017).
- **Unified Feature Selection:** Feature selection methods are usually algorithms that can operate on a single data type. However, selection algorithms that can work on both supervised and unsupervised datasets have been proposed. These algorithms are called unified feature selection algorithms. Unified feature selection algorithms should not be confused with semi-supervised feature selection algorithms, because the datasets that are used in the semi-supervised feature selection include both labeled and unlabeled data. Unified feature selection is related to the selection algorithm, not a data type. More information about the unified feature selection can be obtained from the study of Zhao and Liu (2007) and Han and Kim (2018), which outlines the general outline of the subject.

3.2. Supervised Feature Selection Techniques

Although feature selection methods can be categorized in different ways by considering different criteria, within the scope of this thesis, they are classified and examined by regarding their structures and the way they express their results, i.e., outputs, respectively. Feature selection methods according to their structures, e.g., types of searching, evaluating and selecting the features, are divided into five different classes: filter, wrapper, embedded, hybrid and ensemble, as shown in Figure 3.3. Although the basis of hybrid and ensemble methods are the filter, wrapper, and embedded methods, they have been examined as a separate class in this study because of their structural differences.

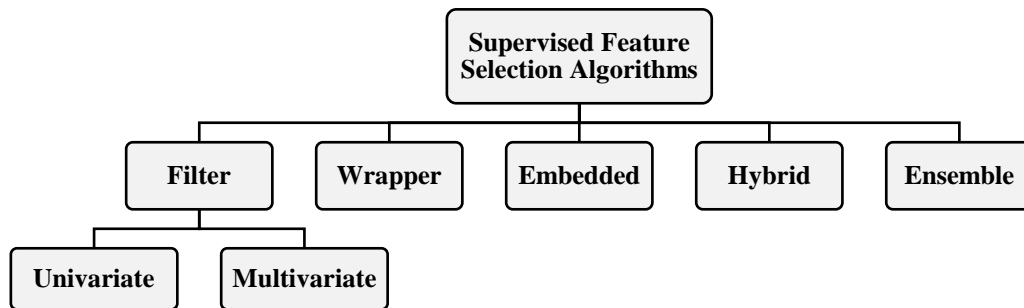


Figure 3.3. Classification of Supervised Feature Selection Methods

3.2.1. Filtering Techniques

Filtering methods, or rankers, fundamentally consist of an evaluation function and a selection criterion. Evaluation function that uses distance, information gain, error rate, and correlation metrics, ranks the entire feature set and sorted features are selected by a user-defined selection criterion, i.e., generally a threshold value. Filtering methods are resistant to overfitting problems, scalable, diverse, fast and easy to implement. Also, they do not require any classification algorithm, since they perform feature selection using mathematical and statistical evaluation functions. However, since filtering methods cannot exactly select features, they are usually used as a preparation stage for feature selection. Filtering methods are divided into two subclasses as univariate and multivariate.

- **Univariate Filtering Techniques:** Univariate filtering techniques do not evaluate the relationship between features. In other words, the usefulness of each feature is evaluated independently, i.e., individually, one by one. This causes the features that

are useless alone but valuable in combination with another feature(s), i.e. feature dependencies, to be ignored. This is the major disadvantage of univariate filtering techniques. F-Score, Chi-Square Test and Information Gain are examples of univariate filtering techniques.

- **Multivariate Filtering Techniques:** The independent evaluation of features is a problem that directly affects the performance of classification. In order to overcome this problem, multivariate filtering methods have been proposed. Multivariate filtering methods reveal the mutual relationship between features using statistical methods. Fast Correlation-Based Filtering (FCBF), Markov Blanket Filter (MBF) and ReliefF algorithms are examples of multivariate filtering methods.

3.2.2. Wrapper Techniques

Wrapper methods perform feature selection in three steps using a classification (induction) algorithm instead of mathematical or statistical functions. In the first step, the wrapper algorithm uses a search method (algorithm) to generate (select) a subset from all possible feature subset spaces. In the second step, the estimated accuracy or error rate of the generated subset is measured by a classification algorithm which is used as a black box, i.e., without considering the internal workings of the algorithm. In the last step, the stopping criterion, for example, the classification error rate, is controlled. If the stopping criterion is met, for example, if the classification error rate does not decrease any more, the algorithm stops, if not, it returns to the first step to select a new subset.

The use of wrapper methods in problems with high dimensional datasets is computationally costly. There are two reasons for this. The first reason is that the number of feature subsets that can be created depends directly on the number of features. For example, from a dataset that contains n features, 2^n subsets ($2^n - 1$ proper subsets) of features can be generated. As can be seen here, an increase in the number of features, also increases the number of subsets not linearly but exponentially. To overcome this problem, that is included in NP-Hard class, exponential (Beam Search, Branch and Bound, etc.), sequential (Sequential Forward, Sequential Backward, Bidirectional, etc.) and random search (Simulated Annealing, Hill Climbing, Genetic Algorithm, etc.) strategies are being used. The second reason is that

each feature subset formed is being tested by the classifier. For these reasons, wrapper methods are not suitable to use on today's large datasets. Another problem encountered in wrapper methods is that the selection process is entirely dependent on the classification algorithm. Changes in the classification algorithm affect the selected feature subsets. Thus, the results obtained are specific to the algorithm and far from being general.

However, in contrast to filtering methods, wrapper methods not only recognize the relationship between the features but also the relationship between the features and the classifier as well. Therefore, they generally provide better (high classification performance) results than the filtering methods. Wrapper algorithms can work with different classifiers and search methods. Instead of a fixed classifier or search method, different classifiers and search methods that are appropriate to the studied datasets can be selected, tested and evaluated according to various criteria. Sequential Forward Feature Selection (SFS), Sequential Backwards Feature Selection/Elimination (SBS/SBE), Naïve Bayes and ID3 (Iterative Dichotomiser 3) can be given as examples for wrapper algorithms.

3.2.3. Embedded Techniques

In embedded methods, feature selection occurs while the classifier is being trained. This means learning and selection processes are not separated and works together as a single step. To summarize both sentences, feature selection is performed by the classifier's guidance and results not only with a selected feature subset but also with a trained classifier. Therefore, embedded techniques have fewer calculation costs than wrapper methods. Like wrappers, they perform feature selection regarding the classification performance, e.g., estimated accuracy or error rate. However, since the subset performances are not tested by the classifier one by one, they work faster than the wrapper methods. They also are more resistant to overfitting problems. The major disadvantage of the embedded techniques is that they are dependent on the classification algorithm used, in the same way, that wrapper algorithms do. Thus, the obtained feature set is dependent on the used classification algorithm, and therefore the results of the different classifiers do not match.

Embedded techniques are divided into two classes: forming a tree and pruning, and regularization. CART (Classification and Regression Trees for Machine Learning)

algorithm can be given as an example of the first class. The CART algorithm forms a binary decision tree and can prune the tree according to several criteria. The CART algorithm performs feature selection in two steps. In the first step, it uses all the features to form the tree by assigning one of them as root (the one with the highest Gini Index or gain) and others as child and leaf nodes (again using gain). In the last step, eliminates the redundant and irrelevant features from the tree by pruning it (pruning also can be done while forming the tree). Thus, the feature selection is realized during tree construction. For the second class, the LASSO (Least Absolute Shrinkage and Selection Operator) algorithm can be given as an example. The LASSO algorithm is essentially a regression analysis method. However, since it assigns a coefficient to each feature as a result of the analysis, it can also be used as a feature selection technique. At the end of the process, the features with zero as a coefficient are eliminated. The remaining features form the result set. Apart from these two algorithms, SVM-RFE (Recursive Feature Elimination for Support Vector Machines), Winnow, Elastic Nets (EN) and FS-P (Feature Selection Perceptron) algorithms can be given as examples of the embedded methods.

3.2.4. Hybrid Techniques

Hybrid methods are created by using filter and wrapper methods together. The main idea is to use the output of the filtering method as input to the wrapper method. In this approach, it is aimed to produce faster and more efficient results using both a statistical method and a classification algorithm in the selection process and to eliminate the individual disadvantages of filter and wrapper methods. In hybrid approaches, the selection process takes place in four steps. In the first step, the filter method evaluates and sorts the entire feature set. In the second step, a rough elimination (selection) takes place on the sorted feature set according to a predetermined threshold value (selection criterion) to form a feature subset. In the third step, the wrapper algorithm steps in and performs search and evaluation on the selected feature subset, respectively. In the last step, the stopping criterion is checked. If the criterion is provided, the algorithm stops, if not, it returns to the third step to select a new subset.

Hybrid techniques can be formed by using different filter and wrapper algorithms. The combined use of filtering and wrapper methods increases the probability of obtaining subsets of features with higher classification performance than when used individually.

Also, unlike filtering, wrapper and embedded methods, hybrid methods do not depend on the performance of a single algorithm. The major disadvantage of hybrid methods is that the selection process is carried out twice. Lastly, even though the number of elements of the feature subset to be given as input to the wrapper algorithm is reduced by the filtering method, ($2^n \rightarrow 2^a, a < n$), in some cases, the required computational requirement may still be high. ReliefF-GA (ReliefF with Genetic Algorithm) and GRASP (Greedy Randomized Adaptive Search Procedure) algorithms can be given as examples of hybrid methods.

3.2.5. Ensemble Techniques

It is often difficult for the user to choose the appropriate technique and algorithm without having any technical knowledge (background) of both the dataset to be studied and the algorithm to be used. This problem is tried to be overcome with ensemble techniques. Ensemble methods are mainly an adapted version of ensemble learning that is used for classification problems, to the feature selection process. The main purpose of this technique is to combine techniques instead of selecting.

Ensemble methods are based on the principle of performing the feature selection process more than once and combining the obtained results into a single feature subset. There are three different ensemble building approaches: data diversity, function diversity, and hybrid approach. In data diversity (variety of data), new datasets created by sampling the original dataset, are analyzed using a fixed single selection algorithm. In functional diversity (variety of functions), a single dataset (the original one) is analyzed by multiple selection algorithms without sampling. The hybrid approach is the combination of other techniques and the datasets are generated by sampling are analyzed by different selection algorithms. Different feature subsets obtained in each of the approaches are combined using a combination function, e.g., mean rank aggregation, and a final result set is obtained. These approaches can be performed using various algorithms of filtering, wrapper and embedded.

Ensemble methods can work in high dimensional datasets and are resistant to overfitting. Moreover, they can provide good results with the aid of methods such as resampling, in cases where the dataset has a small sample size. The major disadvantage of ensemble approaches is that their working patterns and structures become rather complicated at times and consequently the comprehensibility of the results is

significantly reduced. Another aspect to be considered is the combination function because it has the power to directly affect the result set. Finally, although the ensemble methods often produce better feature subsets than a single algorithm, this is not always guaranteed. Ensemble approaches also have a limit, as is the case with each approach. As an example, to ensemble methods, the Random Forest which is generated by forming multiple Decision Trees, GEFS (Genetic Ensemble Feature Selection) and GBFS (Gradient Boosted Feature Selection) can be given.

3.3. Problems Encountered in Supervised Feature Selection

Some of the problems encountered in the feature selection are itemized in this section. Each of these articles can be considered as a subject of improvement.

- **Establishing the relationship between quantity and performance:** Feature selection algorithms try to maximize the classification or clustering performance while minimizing the number of features. Therefore, feature selection is essentially an optimization problem. Yet, it is not easy to ensure both conditions at the same time and reducing the number of features does not always guarantee performance improvements.
- **Ensuring data quality:** Used dataset, especially the training set, affects the performance of the feature selection algorithms directly. Therefore, the datasets should be cleared of various flaws before going into the feature selection. For example, small sample size, high dimension, wrong class labels, and unbalanced class distribution, missing and noisy samples can adversely affect the results of the selection process. There is also another problem that is associated with highly correlated features called multicollinearity. It is difficult to determine the effect of highly correlated features on the results and also such features may also produce misleading results. Hence, a good feature set should include elements that are closely correlated to the classifier, but as uncorrelated as possible to one another.
- **Adding an extra layer to the model building process:** As mentioned earlier, feature selection is a data preprocessing technique used to reduce the size of the datasets. However, this process creates an extra layer during the modeling process. It extends the time required to create a model and increases complexity. This problem especially can be observed in filter selection algorithms easily. Since the filtering-type algorithms sort the features instead of selecting, the selection process

remains to be completed by the user or another selection algorithm. Only then learning and interpretation stages can take place.

- **Setting the algorithm and hyperparameters:** Although it is not unique to the algorithms used in feature selection, it is time-consuming and difficult to determine the selection algorithm parameters and hyperparameters. Essentially, it is possible to interpret this process as a searching problem. To determine the optimal parameters, algorithms must be run several times and the result changes should be observed and interpreted. Also, since these values are problem-specific, they should be determined again as the problem changes.
- **Scalability:** Both horizontal (number of samples) and vertical (number of features) growth in datasets increases the amount of processing power and time and memory needed to analyze by the selection algorithms. While the overall methods show high performance in small datasets, they tend to lose their performance as the size increases. Therefore, the ability to obtain and guarantee scalability is a serious problem encountered in feature selection.
- **Feature selection on unsupervised, multi-labeled and streaming data:** Studies on feature selection often focus on supervised problems (datasets). Therefore, studies on unsupervised (Doquire and Verleysen, 2011; Li et al., 2014; Zang et al., 2014), multi-labeled (Dy and Brodley, 2004; Qian and Zhai, 2013) and streaming (Zheng and Zhang, 2016; Wu et al., 2013; Wang et al., 2017) datasets are relatively few. However, nowadays, the analysis need of streaming data is more notable than in the past. Streaming data is the name given to the data which are generated continuously and in real-time, in large or infinite volumes, dynamically developing over time and generally obtained from sources such as computer networks, social networks, internet and location services. As can be understood from the definition, streaming data analysis is a rather up-to-date and promising subject. In addition to that, topics such as big data feature selection and online feature selection are also open to research and development.

CHAPTER 4

SOURCES AND TYPES OF DATA, CLASSIFIERS AND CLASSIFIER EVALUATION

In this chapter, firstly the data sources and data types (subtitle 4.1) are presented. Then the classification algorithms used to evaluate the performance of the feature subset(s), during the selection process, e.g., wrapper algorithms, and/or after the selection process (subtitle 4.2), are presented. These titles were followed by the accuracy measures (subtitle 4.3) and the validation methods (subtitle 4.4), used to evaluate the classifier (predictor) performance, respectively. Finally, some of the sampling techniques (subtitle 4.5) that can be used to change (for perturbation) the training datasets are mentioned. Figure 4.1 shows an overview of chapter 4 and places where the corresponding subtitles are being used in feature selection.

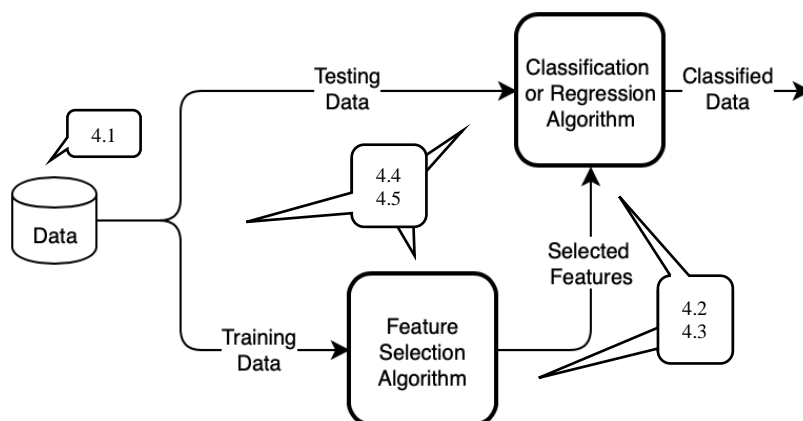


Figure 4.1. An Overview of Chapter 4

4.1. Sources and Types of Data

Qualitative and quantitative datasets to be used in feature selection and classification problems can be obtained by the researcher personally through experiments, observations, and surveys, as well as from written and oral sources such as existing researches, reports, domain experts, newspapers, magazines, papers and articles. For this reason, data sources are examined in two groups as primary and secondary. Primary data sources are preferred because the data collection process is under the control of the researcher, real-time data can be collected and the data is direct and

reliable. However, secondary data sources are quite fast and economical to create. Also, hence they contain mostly out-of-date information, they are a summary of many sources on the subject. Both primary and secondary data sources are examined in two subgroups. These are observation and survey and internal and external. Internal data is generally high-security level data that organizations obtain from their operations. External data, however, is the data that organizations obtain from outside. The classification of data sources is shown in Figure 4.2.

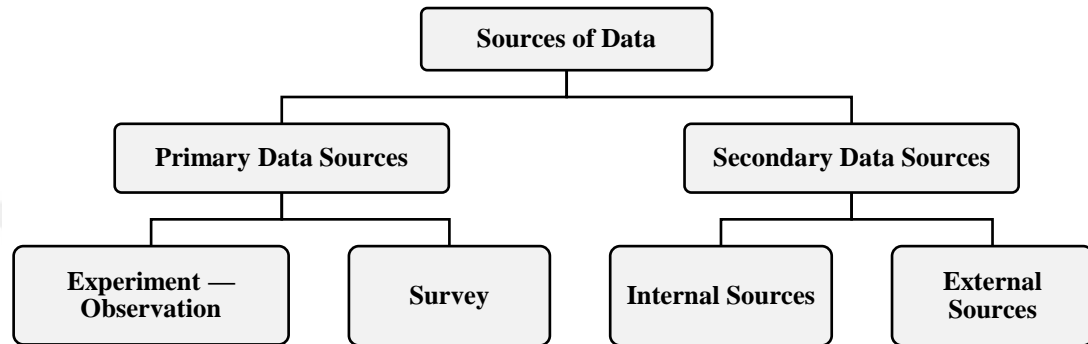


Figure 4.2. Classification of Data Sources

In addition to the real data, synthetic, i.e., artificial, datasets that were firstly proposed by Donald B. Rubin (1993) are being used in the studies. The synthetic dataset is the name given to the datasets generated by various algorithms and data-generating software to mimic real-life data (datasets). Synthetic data does not mean fictional data. Synthetic data are generated with the help of statistical models, e.g., Linear or Non-Linear Regression, obtained from the original dataset and carry the character of the original dataset from which they are produced. This model building process is called a synthesizer build. After the completion of synthesizer build synthetic data can be produced as much as desired. In addition to the synthetic datasets, semi-synthetic datasets which were first proposed by Roderick J. A. Little (1993) are also being used. Semi-synthetic datasets are created by synthesizer builds, just like synthetic datasets, but this time instead of the entire dataset, only the critical data is generated. The rest of the datasets stays untouched. Thus, real and synthetic data would be used together. Synthetic and semi-synthetic data can be generated from datasets derived from primary and secondary data sources. The classification of data types is shown in Figure 4.3.

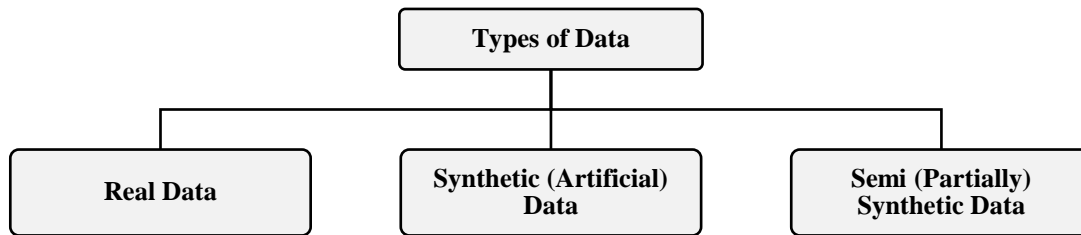


Figure 4.3. Classification of Data Types

The controlled experimental environment can be provided according to the necessities and conditions determined using synthetic and semi-synthetic datasets. For example, according to the experimental scenario to be applied, the amount of noisy data can be increased or reduced and the limits of the algorithm can be determined more clearly. Besides, they can also be used to eliminate data defects, such as skewed datasets. Synthetic and semi-synthetic datasets also serve as a verification technique. After performing classification or feature selection with synthetic datasets, performance evaluation is performed with the actual datasets. Lastly, especially the synthetic datasets give anonymity to the data and provide privacy and confidentiality because actual data may include personal, private or confidential information. Since synthetic data does not contain such information, they cannot be traced back. Thus, privacy and confidentiality are preserved. Despite all these advantages, synthetic datasets are more dependent on the model (synthesizer build) than the semi-synthetic ones. Thus, quality is determined by the model. Moreover, in some cases, it is difficult or impossible to create a model.

In addition to the given taxonomy, data can be divided into two groups: static (offline) or dynamic (online, live or streaming data). Static data as can be understood from the name is the data that does not change over time. Records that belong to databases are the best example of static data. Once they are registered, they do not change. Dynamic datasets are constantly changing and developing over time. Data obtained from sensors or the internet is an example of live data. Nowadays, the use of dynamic datasets is increasing with the help of dynamic data production and consequently the need for analysis and new analysis methods.

4.2. Classification Algorithms

In this section, various classification algorithms used in the performance evaluation during and/or after the supervised feature selection process are briefly mentioned without specifying any order of importance. The purpose of classification, which is a form of supervised learning, is to assign the data (observations) to the classes (class labels) that have been defined previously. In other words, it is to predict the class in which the data may be included. The classification process is performed by using the results, data distributions and characteristics of the algorithm in the learning (model building) phase. For example, loan applications can be divided into three classes such as low, medium and high-risk groups. Any credit application will be categorized into one of these three classes, according to the applicant's previous credit score(s) and/or other criteria. Classification is often used as a basic data mining function in many areas such as finance, banking, security (fraud detection), medicine (diagnosis), astrophysics and space sciences, text mining, image and voice recognition.

If the target value of the data to be assigned is categorical (discrete), for example, the classification of individuals according to education level, then classification analysis is performed, if the target value is continuous, for example, the future value of a currency, regression analysis is performed. However, regression analysis methods, such as Linear and Logistic Regression, can also be used for classification problems. Therefore, classification and regression algorithms are used to evaluate the performance of the feature subset(s) obtained during, e.g., embedded algorithms, or after the selection process, e.g., wrapper algorithms. Within the scope of this thesis, various of the classifiers have been briefly explained.

4.2.1. K-Nearest Neighbor (K-NN)

K-NN (Altman, 1992) algorithm is widely used in solving classification and regression problems. The algorithm requires two hyperparameters: a user-specified k value (number of neighboring points) and a distance metric. Although it is a rather costly method, the number of neighboring points is usually determined by trial and error. However, there are some generally accepted approaches for the determination of this value. One of them is, where n is the number of samples (number of records), taking the k as $\lfloor \sqrt{n} \rfloor$. For example, for a dataset that includes 90 samples, k can be taken as 9. Alternate to this method, the value of k can also be determined with the help of

Bayesian Hyperparameter Optimization. However, no matter which method is preferred, the k value should be cautiously determined as it plays a direct role in the classification process. As distance metrics, Euclid, Mahalanobis, Chebychev, Hamming, Cosine, Spearman can be used. The K-NN algorithm generally works in three steps:

- Each sample, let's call the number of samples by n , is marked to represent a point on the n -dimensional space (\mathbb{R}^n).
- The distance between the point in which its class wished to be determined to the other points, is calculated by a user-defined distance metric.
- After the distances are sorted from smallest to largest (increasing distance), the nearest (from the top of the ordered list) k are selected and their classes are examined. The point with an unknown class is assigned to the class where the majority are. At this stage, in the binary classification problems when k value is an even number, a draw may occur. For this reason, an odd k value is chosen in binary classification problems.

4.2.2. Support Vector Machine (SVM)

Support Vector Machine (Cortes and Vapnik, 1995) is a supervised learning algorithm used in classification (fundamentally in binary classification) and regression problems. The general aim in this approach is to separate the existing classes with the help of a hyperplane that is equidistant to both classes. Examples of both classes closest to the hyperplane are called support vectors. Support vectors form two boundary lines parallel to each other, leaving the hyperplane in the middle. There are no samples of any class within these boundary lines and the name of this area is the margin. The algorithm tries to separate the classes by tampering both the boundary width (can be reduced and expanded) and the position (angle) of the border. However, it is not always possible to separate the samples with a line. For the sample sets that cannot be divided linearly, a nonlinear support vector machine which uses quadratic or cubic functions (curves) can be used. This time, the algorithm performs classification by tampering with the boundary width and position generated by these functions.

The best advantages of Support Vector Machines are that they perform well even in low-sample datasets, and they are resistant to overfitting and noisy data. Support

Vector Machines are essentially binary classification algorithms. However, most of the problems tackled in daily life are multiclass problems. Therefore, in order to use the SVM algorithm in multiclass classification problems, “One vs. One” and “One vs. Rest (All)” strategies were proposed. In the “One vs. One” (OVO) strategy, classes are selected as binary groups. Classes that remain out of the group, are not considered. Therefore, for n class, the SVM algorithm produces $(n^2 * n/2)$ models. In the “One vs. Rest” (OVR or OVA) strategy, each time a class gets selected (also called as a positive class) and all the remaining classes (also called as a negative class) are treated as one single class. Therefore, for n class, the SVM algorithm produces n models. The biggest disadvantage of the SVM algorithm is that both methods are extremely costly in cases where the number of classes is high.

4.2.3. Naïve Bayes (NB)

Bayes Classifiers (Pedro and Michael, 1997), which can be used in datasets with categorical (discrete) class labels, perform classification according to the probability based on Bayes’ Theorem. The Bayes’ Theorem is used to calculate the probability that samples are included in a particular class. Generally, when the algorithm encounters a sample with an unknown class, it calculates this value for all the classes and assigns the data to the highest possible class. According to Bayes’ theorem,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

A and B are events, $P(A|B)$, the likelihood of event A occurring given that B is true. $P(B|A)$, the likelihood of event B occurring given that A is true and $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other.

Bayes Classifiers, which are exemplary of conditional probabilistic classifiers, are frequently used since they are fast, easy to apply and often have high classification performance. However, Bayes Classifiers have several disadvantages. Firstly, Bayes Classifiers do not consider the relationships between the features. Thus, they are called “naïve” Bayes Classifiers. However, features in real life, like the relationship between disease symptoms such as exhaustion, cough, and fever, are often related to each other. To consider the relationships between the features, Bayes Belief Networks (BBN) is being used. The second disadvantage is that the algorithm behaves as if all the features have the same importance (weight). Finally, the last disadvantage is that the Bayes

Classifiers do not work on datasets with a continuous class label. For this reason, Gauss Naïve Bayes (GNB) classifier is used for classification problems that contain continuous data.

4.2.4. Error Correcting Output Codes (ECOC)

ECOC (Dietterich and Bakiri, 1995) algorithm is an ensemble method that is created by combining binary classifiers, e.g., SVM, for solving multiclass classification problems. As mentioned before, multiclass problems have more than two class labels. Therefore, ECOC algorithm represents each class with a fixed length unique binary codes that are created randomly or using a specific method, e.g., c number of classes are coded with 0s and 1s in the length of ℓ where $\ell \geq \log_2 c$, in order to transmute the problem into binary classification. After that, the ECOC matrix is created by combining these codes. Each row in the matrix refers to a different class. Therefore, for each column, a binary classifier is assigned and run separately. As referred, classifiers are responsible only for the column they assigned and cannot interfere with each other. An example of two different 4×4 ECOC matrices created for four classes, $C = \{C_i | i = 1, \dots, 4\}$ is shown in Figure 4.4. Given the example, it can easily be seen that four classes can also be represented by 2 bits (00,01,11,10). However, expressing classes with more bits such as in the first matrix increases the error tolerance of the algorithm.

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 4.4. An Example of Two Different ECOC Matrices (Output Codes) Created for a Four-Class Problem

When a sample with an unknown class is encountered, each classifier generates code with the same length, ℓ , as the code generated for the classes, and the generated code is compared with the class tags using Hamming Distance. The sample is then assigned to the same or the nearest class. The major disadvantage of this method is that a classifiers' mistake, makes the entire classification process inaccurate. It is possible to construct an ECOC classifier from classifiers such as SVM, Decision Tree, K-NN, Discriminant Analysis or Naïve Bayes.

4.2.5. Artificial Neural Network (ANN)

Artificial Neural Network is the name of the system, which mimics the real biological nerve cells and is formed by artificial neural cells called nodes or neurons clustering in the form of layers. ANNs are used for many purposes such as image and signal processing, pattern recognition and analysis, classification, abnormality detection, optimization, and association. Artificial neurons, which form the ANNs and where the information is processed, consists of input (data), the weight value of the input, transfer function, activation function, and output parts. Inputs are processed through these layers respectively.

Artificial neurons are interconnected and interacting with each other as biological nerve cells. Thus, each node that is working in parallel, transmits its input after processing to the other node or nodes to which it's connected. For this reason, ANNs are being designed as layers. These layers are called input, hidden and output, respectively. Networks consisting of only input and output layers are called single-layer, networks consisting of input, hidden and output layers are called multi-layered neural networks.

- 1. Input Layer:** The input layer is the layer that receives data from outside and transmits the input (data) it receives to the network. Since each neuron in the input layer receives an input, the number of neurons (nodes) are equal to the number of inputs.
- 2. Hidden Layer:** Hidden layer is the layer between the input and output layer. The hidden layer's job is to process the data that comes from the input layer and to transmit the processed data to the output layer. The number of hidden layers may vary depending on the architecture that is applied. Some ANNs do not have a hidden layer, while some may have more than one. Likewise, the total number of neurons in the hidden layer may vary, as it is not dependent on the number of neurons in the input and output layers.
- 3. Output Layer:** The output layer is the last layer and its task is to convert the data coming from the hidden layer into the desired output. All values, x_i , that enter into the hidden and output layers are multiplied by their weight, w_i , first and then added. At this stage, they form the $(\sum_i(x_i w_i) + b)$ transfer function by adding the bias, b , value. The value obtained is controlled by the activation function, $f(\varphi)$, e.g.,

sigmoid, hyperbolic tangent, etc. Essentially, the activation function is used to limit the transfer function. Thus, it is determined whether or not the neuron will be activated.

There are two types of ANNs, feed-forward, and backpropagation (the backward propagation of errors). The backpropagation ANNs are used in both classification and feature selection. Classification in backpropagation ANNs is performed in two steps. In the first step, the ANNs are trained with the training datasets. Generally, the algorithm checks the target output with the output it produced each time, regulates the weights entering the nodes (usually the initial weights are determined as random values), w , therefore completes the learning process. This process is completed when the training error rate, i.e., the difference between the algorithm output and the target output, is reduced to an acceptable limit when it does not reduce any further or the output generated by the algorithm is identical to the target output. In the last step, the system is tested with data that has not been encountered before. When the performance (classification accuracy) reaches the desired level or does not increase any further the best features, i.e., those weigh more than others, are automatically selected. Therefore, ANNs are an example of embedded techniques.

Artificial Neural Networks have properties such as self-learning and generalization ability, tolerance against inaccurate and incomplete data, parallel operation, and adaptability to unknown situations. However, ANNs also learn with examples as same as humans. Therefore, the quality of the training set directly affects the learning process. Occasionally it is difficult and costly to determine the appropriate network structure and parameters or adapt the problem to the network. In this case, the network structure may become complex. While generating behaviors that are hard to explain, complex networks also require more processing power, memory and time. Also, ANNs may perform poorly in small datasets.

4.2.6. Decision Tree (DT)

Decision Tree classifiers are frequently used since they are reliable, simple, have low computational complexity and can work with high dimensional datasets with both continuous and discrete variables. Decision Trees consist of a root, inner (or test) and leaf (or terminal or decision) nodes and branches. Root and inner nodes represent

features, branches represent the values that features may possess and the leaf nodes refer to the classes or preferences (decision).

Size (number of nodes) of the decision tree, plays an important role in the classification performance. For example, large trees work slowly and have an overfitting tendency, while small-sized, i.e., bushy, trees are not successful in the classification despite their fast work. Therefore, after the tree is completely formed, pruning is used to obtain the tree with the most suitable size. Pruning is a technique used to remove sections (reducing tree size) that do not contribute to the classification from the tree to increase the classification accuracy and speed of the classification process and reduce the complexity of the final classifier. The pruning process can be carried out both during and after the formation of the tree. Algorithms such as ID3, C4.5, CART (Classification and Regression Trees) and CHAID (Chi-square Automatic Interaction Detection) can be used to form decision trees. However, each algorithm can form the Decision Tree in different ways.

Each part leading from root node to leaf node on the tree can be expressed with an “if-then” phrase. These sentences are called classification rules. It is preferred to express the results with classification rules, especially since the large tree structures might be difficult to understand. Classification rules are usually written from top to bottom, namely from the root node to leaf node. In between if-then (antecedent part), the conditions (each node and their values, starting from the root node until it reaches the leaf node) are written. If it passes through more than one inner node, a “and” conjunction is added between the past nodes. Finally, after “then” (consequent part), the class label is written. For example, for an online shopping site “*If the monthly visit of the customer is less than two, and the average monthly expenditure is less than 100 TL then create a customer-specific promotional coupon.*” classification rule, can be created using a decision tree.

Decision Tree classifiers select features that are effective (or useful or relevant) to the classification automatically during the tree construction with the help of the splitting criterion, e.g., Information Gain or Gain Ratio. Here it must be noted that there are two main types of splitting criterion: univariate (splitting the instance space is performed according to the value of a feature) and multivariate (splitting the instance space is performed according to the values of several features). Information Gain, Gain Ratio, impurity, distance, etc. are examples of univariate splitting criterion.

In order to select relevant features, first, all the features are evaluated with a splitting criterion. Then the features are sorted according to the evaluation results and the best feature is assigned as the root node. According to the feature on the root node, the remaining (the ones that are not used) features, and the stopping criterion, the tree splits and expands. This is how feature selection is performed. More information about Decision Tree classifiers can be found in the survey by Rokach and Maimon (2005).

4.2.7. Linear Discriminant Analysis (LDA)

The Linear Discriminant Analysis proposed by Sir Ronald Aylmer Fisher (1936), is a parametric classification method that aims to determine the relationship between two or more continuous independent variables or variables and a categorical dependent variable (class variable or class label). Therefore, it is also used as a dimension reduction technique. LDA can be used for binary and multiclass problems. LDA performs classification by reflecting the samples onto a line. Thus, the size of the n -dimensional dataset is reduced. However, because the size is reduced, the samples are often positioned very close to one another or on top of each other. In this case, the discrimination process becomes difficult and the rate of finding the class labels is reduced. In order to avoid this problem, the algorithm tries to find the point that provides a lower variance between the class elements, and a higher variance among the classes by changing the angle and direction of the projected line. After the location of the line is found, the class label of the unknown samples can be found. Discriminant analysis, which is one of the multivariate statistical techniques, is used for the detection of the most effective discriminative variables, apart from the classification function. LDA classifiers are fast, require low memory and CPU usage and easy to implement and interpret.

4.3. Evaluation of Classification Performance

The main purpose of feature selection is to find the smallest feature subsets that can produce accurate classification models. Therefore, the classification performance of the selected subsets should be measured. Various classification model assessment measures are being used for this task and they are visualized by using confusion matrix which is a specific table, and ROC Curve, which is a graphical plot.

4.3.1. Confusion Matrix

Confusion matrix, which is an $n \times n$ square matrix/table with the class size n , e.g., for binary classification problems it is 2×2 matrix, is generated by tabulating the number of true and false estimates made by the classification algorithms. In the confusion matrix, rows represent actual classes and columns represent estimated classes (could be vice versa). The accuracy rate of a classifier is obtained by dividing the number of correct estimates by the total number of estimates and the error rate is equal to one minus the accuracy rate. The confusion matrix of a binary classifier is shown in Table 4.1.

Table 4.1. Confusion Matrix

		Predicted Class		Total
		A	B	
Actual Class	A	a (TP)	b (FP)	a + b
	B	c (FN)	d (TN)	c + d
Total		Positive Estimates	Negative Estimates	

True positive, true negative, false positive and false negative values can be visualized using the confusion matrix. For example, suppose that for a binary classification problem we are calling the instances that belong to class *A* as positive and the instances that belong to class *B* as negative. True positives (TP) are the total number of positive instances, i.e., that belongs to the class *A*, estimated correctly, i.e., correctly predicted or labeled, by the classifier. True negatives (TN) are the total number of negative instances, i.e., that belongs to the class *B*, estimated correctly, i.e., correctly predicted or labeled, by the classifier. False positives (FP) are the total number of negative instances, i.e., that belongs to the class *B*, estimated wrongly, i.e., incorrectly predicted or labeled, by the classifier. False negatives (FN) are the total number of positive instances, i.e., that belongs to the class *A*, estimated wrongly, i.e., incorrectly predicted or labeled, by the classifier.

Once these values are calculated other classification performance criteria such as accuracy, error rate, sensitivity, specificity, Positive Prediction Value (PPV), Negative Prediction Value (NPV), F1-Score, etc. can be calculated (see Table 4.2) with the help of the confusion matrix. In cases where there are more than two classes ($n > 2$), after creating the confusion matrix for all classes, the “One vs. Rest” strategy can be used.

In this method, one class is selected and named as a positive class, each time, until there is no class left, and regarding all the remaining classes named as a negative class, and treated as a single class, performance values are calculated one by one. A list of metrics for classification model evaluation can be found in the review by Hossin and Sulaiman (2015).

Table 4.2. List of Various Classification Model Evaluation Metrics

Metric	Formula
Accuracy Rate	$(TP + TN)/(TP + TN + FN + FP)$
Error Rate	$1 - \text{Accuracy Rate}$
Sensitivity	$TP/(TP + FN)$
Specificity	$TN/(TN + FP)$
Positive Prediction Value (PPV)	$TP/(TP + FP)$
Negative Prediction Value (NPV)	$TN/(TN + FN)$
Prevalence	$(TP + FN)/(TP + TN + FN + FP)$
F1 Score	$\frac{2 * PPV * \text{Sensitivity}}{PPV + \text{Sensitivity}}$

4.3.2. ROC (Receiver Operating Characteristics) Curve

ROC Curve is another technique used to visualize and measure the accuracy of binary classifiers (can also be used in multi-classification problems) in two-dimensional ROC space. In order to draw the ROC space, true positive (TP, 1- specificity) and false positive (FP) values are needed. ROC Space has false positive values on the x-axis, and true positive values on the y-axis where both values are in the range of [0,1]. Therefore, the ROC Curve is a continuous function defined between (0,0) and (1,1) coordinates. The upper left corner of the ROC space ((0,1) coordinate) is the most ideal point and it is called the perfect classification. After true positive and false positive values are calculated according to each cut point (decision thresholds) determined by the user and the curve is created by marking it on the space.

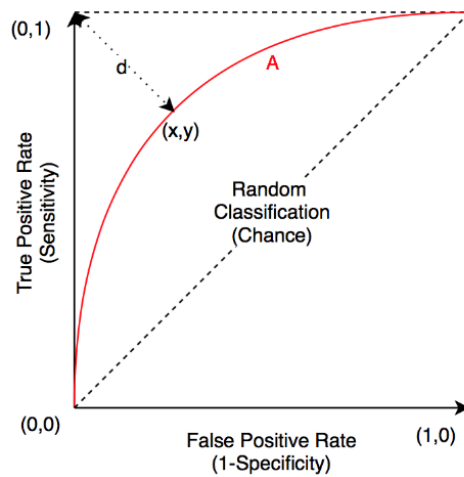


Figure 4.5. The ROC Space and the ROC Curve for a Classifier

The success of the classifier is determined by the area under the curve (AUC). Therefore, the larger the area, the more successful the classifier is. If the area under the curve is 1, this means that the classifier correctly classifies all the samples without making any mistakes. If the area is 0.5 (area under the diagonal line), this means that the classifier randomly, i.e., by chance, performs the classifying process. In other words, this is the same as classifying samples with a coin flip. Lastly, if the area is smaller than 0.5, this means that the classifier is working poorly and making more false selections than the right ones. Table 4.3 shows an example of how to interpret the classification performance considering the area under the curve. The taxonomy in this table is just to give an idea to the reader and can be changed.

Table 4.3. Interpretation of AUC Values

AUC	Classification Performance
0.91— 1.00	Very Good
0.81— 0.90	Good
0.71— 0.80	Mediocre – Fair
0.61— 0.70	Poor
0.51— 0.60	Very Poor
≤ 0.50	Valueless

The TP and FP values can also be expressed by points placed in the ROC space instead of the ROC Curve. In that case, instead of the area, the distance of points to the upper left corner (0,1) is considered and Euclidean Distance,

$$d = \sqrt{(1 - \text{TPR})^2 + (1 - \text{FPR})^2} \quad (2)$$

is used as a distance measure (Figure 4.5). The closer the dots are to the corner, the better the performance of the classifier. And for multiclass problems, as in the confusion matrix, for each class a single ROC Curve is generated using the “One vs. Rest” strategy and the AUC value is calculated. For a multiclass ROC analysis, the study of Landgrebe and Duin (2007) can be examined.

4.4. Validation and Cross-Validation Techniques

Datasets are separated into training, testing, and verification or training and testing sets by using validation and cross-validation methods. The main purpose of using these techniques is to determine the classifier performance. However, they also can be used to set the classifier parameters. The difference between validation and cross-validation methods is that in validation the training and testing of the classifier is performed only once. In this section Holdout, Resubstitution and Random Subsampling Validation and Bootstrapping, K-Fold and Leave-One-Out Cross-Validation techniques are briefly mentioned.

4.4.1. Holdout

Holdout validation is one of the most widely used and simplest validation techniques, which can be performed in two different ways.

- **Training, validation and testing sets:** In the first approach, the initial dataset is randomly separated into three disjoint, i.e., mutually exclusive, training, validation and testing sets. The training set consists of approximately half of the total data and its task is to train the classifier. The remaining data is separated into two, half of it as the testing and the other half as the validation set. The testing set, which consists of unused data, is used to measure the performance, i.e., accuracy, of the classifier and the validation set is used to set the algorithm parameters.
- **Training and testing sets:** In the second approach, the validation set is added to the training set. In this case, the initial dataset is separated into two disjoint sets as approximately 70% of the dataset for the training and 30% of the dataset for the testing.

The data in sets can vary in both applications. The main point to be considered here is that the training set is always bigger in ratio than the other sets. Otherwise, the training of the classifier would be incomplete and incorrect. Holdout validation has two

disadvantages. The first is where to split the dataset. Holdout validation works only once (that is why it is not a cross-validation technique), unlike other methods, therefore, a poorly made separation of dataset causes us to have misleading results. The second one is that the initial dataset may not be large enough to be separated into two or three parts. In the Holdout validation method, the overall accuracy is equal to the accuracy shown in the testing set:

$$\text{Acc} = a_{\text{test}} . \quad (3)$$

4.4.2. Resubstitution

Resubstitution is a validation method where the entire initial dataset is used for training first, and then it is used for testing. However, this technique is not preferred since it leads to the problem of overfitting in most of the machine learning algorithms. Also, the error rate in this technique turns out to be quite low. The overall accuracy rate is equal to the accuracy that is shown in the testing set (see Formula 3).

4.4.3. Random Subsampling

Random subsampling is a variation of Holdout method, therefore, as in the Holdout, the initial dataset is separated into training and testing sets. This time, however, the training set consists of randomly selected, i.e., sampled from the initial dataset without replacement, data from the initial dataset and the data that is remaining from the selection, i.e. unselected data, create the testing set. After the partitions are created, the training is carried out first and then the testing takes place. This process is repeated n times. The overall accuracy in random subsampling validation method is calculated as the average of the accuracy rate in each iteration:

$$\text{Acc} = \frac{1}{n} \sum_{i=1}^n a_i . \quad (4)$$

4.4.4. Bootstrapping (0.632 Bootstrap)

In Bootstrapping data reserved in the training and testing sets are sampled uniformly with replacement from the initial dataset. Therefore, in testing or training set, the same data can be chosen more than once. For example, in the test set, they may be three sample number 2. However, same data cannot be in both the training and the testing

set at the same time. Data that did not get selected in the training set are assigned to the test set. This process is repeated n times. Bootstrapping technique is usually preferred when the dataset is not large. In Bootstrapping, the possibility of each data getting selected is $1/n$ and the possibility of not getting selected is $(1 - 1/n)$. Since the selection process is repeated n times, this probability becomes $(1 - 1/n)^n$ for each data and this is approximately equal to $e^{-1} \approx 0.368$ or 38.6%. This value is the probability of any data within the dataset that will not be selected for training. Therefore, the training set will reserve approximately 63% of data from the initial dataset. That is why this method is also called as 0.632 Bootstrap. The overall accuracy of Bootstrapping is equal to the sum of accuracy in the testing and the training sets:

$$\text{Acc} = \frac{1}{n} \sum_{i=1}^n (0.632 * a_{\text{test}_i} + 0.368 * a_{\text{train}_i}). \quad (5)$$

4.4.5. K-Fold Cross-Validation (K-Fold CV)

In this technique, the initial dataset is separated into k equal-sized disjoint, i.e., mutually exclusive, subsets and the training and testing of the classifier are repeated k times. If the initial dataset cannot be separated evenly, one subset can contain more samples than the other. In each iteration, one subset is used for testing and the others are used for training purposes. Therefore, each data used in the testing set once, and in the training set $(k - 1)$ times. The overall accuracy in this technique is calculated as the average of the accuracy rate in each iteration (see Formula 4, where $k = n$).

4.4.6. Leave-One-Out Cross-Validation (LOOCV or Rotation Estimation)

In LOOCV, the initial dataset is separated into subsets, as in the K-Fold Cross-Validation. This time, however, the number of subsets, k , is equal to the number of instances (data points or records), n , that forms the initial dataset. In each iteration, the selected classifier is trained $(n - 1)$ times, tested only once. LOOCV is the only method that uses, i.e., utilizes, as much as data for training, however, its computing costs are rather high, especially in large volume datasets. The accuracy in LOOCV is calculated as the average of the accuracy rate in each iteration (see Formula 4).

4.4.7. Stratified (or Proportional) Holdout and K-Fold CV

In general, stratification, which is a sampling method, is used to ensure that each class is equally represented across training and testing sets. It is possible to use stratification in Holdout or K-Fold CV techniques. This method often used to reduce the variance and to provide diversity in cases where the number of instances per class is unbalanced, i.e., where the data has asymmetric distribution (skewed), can be applied in two different ways. The first way, the samples are taken equally from each sample group, i.e., class. The second way, the number of samples, i.e., sample sizes, in each group is taken into consideration in determining the size of each stratum. More samples are taken from the group that has more samples. In this study, a stratified 10-Fold CV was used during the empirical studies on sequential feature selection algorithms.

4.5. Sampling

Sampling is selecting a certain amount of data, i.e., sample, that represents the basic characteristics of the dataset, i.e., population, from which they were taken, for analysis and reporting. Sampling is a data reduction technique. Especially since it is costly to process high volume datasets, dealing with a part of it rather than the whole set, helps to reduce computing costs. Sampling is also used for determining feature selection stability. With sampling, different sample subsets from the same probabilistic distribution are generated and the responses of the feature selection algorithms to these samples are measured and evaluated by stability measures. In this thesis, the most commonly used techniques such as random, stratified and cluster sampling (Figure 4.6) are mentioned.

- **Random Sampling (or Simple Random Sampling, SRS):** Random sampling is the simplest sampling technique where each data, i.e., an individual for the population, that forms the dataset, i.e., population, is selected independently and with equal probability, i.e., chance. For example, the probability of selecting data from a dataset that contains n sample is $1/n$ in each selection. There are two types of random sampling: with replacement and without replacement.

1. In simple random sampling with replacement (SRSWR), each data is selected randomly from the dataset, are re-added or replaced into the dataset. Therefore, any data that is selected may be selected more than once. The total number of datasets, with n number of samples (data), that can be selected from a dataset

containing N data is equal to N^n and each dataset can be selected with the possibility of $1/N^n$.

2. In simple random sampling without replacement (SRSWOR), each data is selected randomly from the dataset, are not re-added or replaced into the dataset. Therefore, any data that is selected cannot be selected again. The total number of datasets, with n number of samples (data), that can be selected from a dataset containing N data is equal to $\binom{N}{n}$, and each dataset can be selected with the possibility of $1/\binom{N}{n}$.

- **Stratified Sampling:** In this technique, which is used to reduce variance, the sample set is firstly divided into two or more non-overlapping groups, called strata, which may contain equal or different numbers of samples. The formation of strata is done according to a particular feature (purpose, criterion or goal), such as age, educational status or income, and the examples that fit the property of strata, form the strata samples. Strata should be homogeneous in itself, but heterogeneous to one another. As mentioned, each sample belongs to one and only one stratum, i.e., mutually exclusive, and sampling is performed by random selection from each stratum. Therefore, in the strata, each sample has the same probability of being selected. Rare (numerically less than others) samples are less likely to be selected and represented in larger clusters. Stratified sampling helps to increase this possibility.
- **Cluster Sampling:** In cluster sampling, data are divided into sets according to a specific feature (purpose, criterion or goal). Sets may contain equal or different numbers of data. However, each data belongs to only one set. Sets are selected by random selection and the data in the selected set constitute the dataset. In this technique instead of selecting individual data, a group of data is selected.

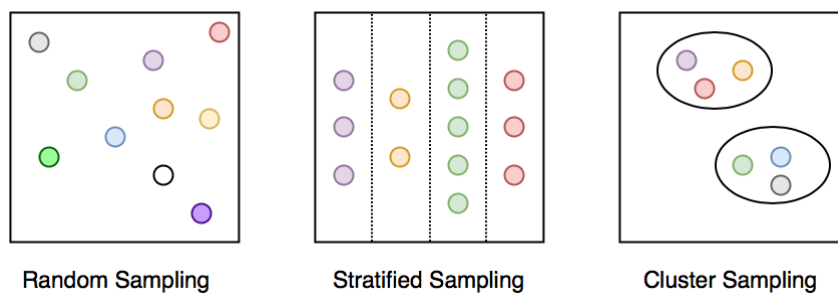


Figure 4.6. Illustration of Random, Stratified and Cluster Sampling Techniques

Although cluster sampling and stratified sampling appears to be similar in structure, they are completely different. While in the cluster sampling clusters are being selected, in stratified sampling, samples are being selected from the strata. Apart from these mentioned methods, there are more sampling techniques. One of these is adaptive sampling. In this technique, sampling starts with a small sample subset and stops when the number of samples reaches a sufficient size. Sufficient size may vary depending on the intended use of the sample subset. For example, the classification performance for classification will start to vary with the increase in the number of samples and becomes stable at some point. In this case, the point at which it becomes stationary gives sufficient size. The most frequent used sampling method in data mining and stability determination is random sampling. Comprehensive information about the purposes of sampling in the data mining process can be obtained from the research studies done by Weiß (2008) and Khandar and Dani (2010).

CHAPTER 5

STABILITY IN FEATURE SELECTION

As mentioned earlier, feature selection algorithms attempt to select the smallest possible feature subset, i.e., a set or vector of features with the lowest possible cardinality, that maximizes classification or clustering performance within the original dataset. However, during this process, the stability of the selection algorithm is relatively being neglected. The stability of the selection algorithms is measured by its sensitivity to changes (perturbations) made in the training set. Changes can be done by obtaining different data subsets by a sampling method, adding/subtracting samples to the original dataset, reordering the samples or features, adding noisy and/or outlier samples. Any stable algorithm must be affected by the changes in the training dataset as low as possible or not at all. In this study, only the stability of the supervised feature selection techniques and stability measures are considered.

A stable algorithm should not produce different results when it is run on datasets generated from the same probabilistic distribution. Therefore, to determine the stability the algorithm results are checked after each change is made in the training dataset. A stable algorithm, i.e., insensitive to changes, produces results (feature subsets or ranks) that are identical or very similar, while unstable one, i.e., sensitive to changes, produces results that are slightly similar or completely different. Stability is a serious problem that needs to be addressed, as it makes it difficult to validate and interpret the selected features. Therefore, it is necessary to evaluate the algorithms not only according to their classification or clustering accuracy but also to their stability. In order to increase stability, methods such as data preprocessing to reduce faulty data and class variance and ensemble feature selection methods are generally used.

Stability largely depends on the dataset. Unbalanced class distributions, skewed data, outliers, noisy values, features that carry similar information or close correlation, i.e., multicollinearity or multi-dependency (if the relations between independent variables are not linear), insufficient number of samples and high dimensions are important factors that affect the algorithm stability. Besides, using a feature selection algorithm that is not suitable for the dataset and/or incorrectly setting parameters and/or hyperparameters of the feature selection algorithm also affects the stability. Therefore,

it is necessary to select the appropriate algorithm for the correct stability measurement and to use the same algorithmic parameters in each iteration, if any. Otherwise, the determination of stability will be performed incorrectly.

5.1. Stability Measurement in Supervised Feature Selection

Supervised feature selection algorithms represent (or express) their results in three different ways: by feature ranking (scoring), weighting and indexing (see Figure 5.1). Although it is possible to select a subset of features by scoring all the features first, then sorting them according to their scores and lastly selecting some of the features according to a certain threshold value (selection criterion), it is not possible to perform the opposite. Therefore, it is necessary to perform stability measurement according to the algorithm outputs.

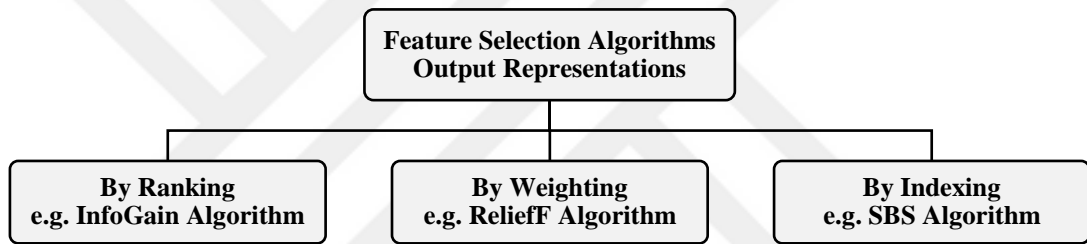


Figure 5.1. Classification of Feature Selection Methods in Terms of Result Representations

Feature selection algorithms have three different categories of stability measurement according to the output (result) expressions.

- **Stability measurement using rank:** Feature selection algorithms, which represent results in terms of feature ranks, e.g., Information Gain algorithm, create a vector of features with a descending relevancy move from left to right. The stability measurement is performed by calculating the distance or correlation between the feature rank vectors obtained after each change was done in the training set using Spearman's ρ (rho), Kendall's τ (tau), Canberra Distance, etc.
- **Stability measurement using weight:** Feature selection algorithms, which represent results in terms of feature weights, e.g., ReliefF algorithm, assign a weight to all features ranging from 0 to 1 (most useful features) or -1 and 1

according to their relevancy. The stability measurement is performed by calculating the correlation between the feature weight sets obtained after each change done in the training set using only Pearson's Correlation Coefficient.

- **Stability measurement using index:** Feature selection algorithms, which represent results in terms of feature indexes, e.g., SBS algorithm, stability measurement is performed according to the locations, i.e., places, of features in the selected subset. Selected features can be expressed using either binary strings, 1 means feature is selected and 0 means feature is not selected, as $\vec{S} = [11001 \dots]$, or a simple vector of feature indices, as $\vec{S} = \{1,2,5 \dots\}$. The stability measurement is performed by using set-based (index-based) stability measures such as Sørensen-Dice Coefficient, Kuncheva and Jaccard Index, Tanimoto Distance, etc.

Several stability measures and the related studies are listed in Table 5.1. A more detailed explanation of stability measures can be found in the literature in reviews by Awada et al. (2013) and Mohana and Perumal (2016).

Table 5.1. List of Stability Measures Classified According the Representation of the Output of the Feature Selection Technique

FSA Output	Measure	Reference
Rank	Kendall's τ (tau)	Shabbir et al. (2014)
	Spearman's ρ (rho)	Kalousis et al. (2007)
	Canberra Distance Weighted Canberra Distance	Jurman et al. (2008)
Weight	Pearson's Correlation Coefficient (PCC)	Kalousis et al. (2007)
Set (Index)	Sørensen-Dice Coefficient	Yu et al. (2008)
	Jaccard Index	Saeys et al. (2008)
	Tanimoto Index	
	Kuncheva Index	Kuncheva (2007)
	C, CV, CW_{rel}	Somol and Novovicova (2010)
	Average Normal Hamming Distance	Dunne et al. (2002)
	Lustgarten's Measure	Lustgarten et al. (2002)

5.2. The Properties of Stability Measures

In the literature, properties that stability measures should have, are listed under different titles (Nogueira and Brown, 2015; Nogueira and Brown 2016; Mohanaand and Perumal, 2016). Therefore, based on the publications reviewed, five properties are mentioned in here without considering any order of importance.

- 1. To have limits (Upper and lower bounds):** The results of stability measures should be ranged in closed bounded intervals. According to this property, stability result of an algorithm, e.g., x , should be greater than or equal to a and less than or equal to b , i.e., $a \leq x \leq b$, ($x \in [a, b]$). For this reason, the results of stability measures in the literature are bounded on $[0,1]$ or $[-1,1]$.
- 2. To be monotonic:** The higher the similarity between results, the greater the stability value should be. For example, if the subsets s_1 and s_2 are selected from a dataset, i.e., ($s_1, s_2 \subset D$), then the greater the $n(s_1 \cap s_2)$, or $|s_1 \cap s_2|$, the greater the stability. For feature rankers (scorers) the monotonicity property is provided by considering the similarities in feature rankings or scores.
- 3. To have a mechanism to correct the coincidental overlap (result by chance):** Especially, in the case when the number of features in the original dataset is small, the chance of selecting similar feature subsets can be expected to be high. Similarly, the increase in the number of selected feature subsets and/or the number of selected features (feature subsets with high cardinality) also increase this probability. Thus, a constant value is required to correct the coincidental overlap, i.e., intersection by chance, issue.
- 4. To be symmetrical:** According to the symmetric property, if x is equal to y , then y must be equal to x . For this reason, the stability value should not change depending on the order of feature subsets. For example, for s_1 and s_2 feature subsets, stability value of (s_1, s_2) and (s_2, s_1) must be the same. Otherwise, the measure would not be symmetrical. The symmetrical property is not related to the order of elements in the feature subsets.
- 5. To be independent of quantity:** Results of the selection algorithms can be in various sizes, i.e., lengths. For this reason, the stability measure should be able to work on the result subsets with a different number of elements (cardinality). For example, a feature subset with x number of elements should be compared with a

feature subset with y number of elements, even if x and y are not equal ($x \neq y$). Some stability measures in the literature, for example, the Kuncheva Index, work only on result sets of equal cardinalities.

5.3. Types of Stability Measures

The stability value of feature selection algorithms is determined in 4 steps. In the first step, changes are made in the training data. In the second step, the algorithm is executed. The first and second steps are repeated in the determined amount and all the feature subsets selected during each iteration (repetition) are collected. In the third step, regarding the algorithm's output representation these subsets are compared in pairs (except frequency-based stability measures because they do not use pairwise comparison) by using stability measures. In the last step, the average of the stability values is taken to determine the final stability value of the algorithm. The result of each comparison can be represented using upper triangular, (A in Figure 5.2), or symmetric matrix, (\mathcal{S} in Figure 5.2). Both matrices are square matrices of size $n \times n$, where n is the number of iterations (or the total number of changes made in the dataset). Besides matrices, a vector (\vec{R} in Figure 5.3) can also be used to represent comparison results.

$$A = \begin{bmatrix} 1 & r_{12} & r_{13} & r_{14} \\ 0 & 1 & r_{23} & r_{24} \\ 0 & 0 & 1 & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathcal{S} = \begin{bmatrix} 1 & r_{12} & r_{13} & r_{14} \\ r_{21} & 1 & r_{23} & r_{24} \\ r_{31} & r_{32} & 1 & r_{34} \\ r_{41} & r_{42} & r_{43} & 1 \end{bmatrix}$$

Figure 5.2. Comparison Results Represented by Matrices

$$\vec{R} = [r_{12}, r_{13}, r_{14}, r_{21}, r_{23}, r_{24}, r_{31}, r_{32}, r_{34}, r_{41}, r_{42}, r_{43}]$$

Figure 5.3. Comparison Results Represented by a Vector

5.3.1. Set-Based (or Index-Based) Stability Measures

Set-based (index-based) stability measures compare feature subsets results in pairs using basic set operations (intersection, union, set difference, etc.) in order to evaluate the algorithm's stability value. For n number of results, where n is bigger than 1, $(n^2 - n)/2$ pairwise comparisons are needed and the final stability value of the algorithm is determined by taking the average of stability values. If the result is equal to 1 then the result sets are the same, while 0 or -1 means they are completely different.

The generalized formula of set-based stability measures is intersection over the union of two result sets, e.g., a and b , $((a \cap b) / (a \cup b))$. If the result sets are expressed using binary strings instead of feature indices, stability value is evaluated with the help of logical operators, “and” logical operator is used for intersection and “or” is used for the union, without changing the stability measure formula. In the scope of this study, X and Y express result vectors, n expresses the number of features in the original dataset and c expresses cardinality of the selected feature subset, several set-based stability measures which have been examined or used are listed in Table 5.2.

Table 5.2. List of Set-Based Stability Measures

Measure	Formula	Bound	Measures
Jaccard Distance. (JD)	$J(X, Y) = \frac{ X \cap Y }{ X \cup Y }$	[0,1]	Dissimilarity
Jaccard Index (JI)	$JI = 1 - J(X, Y)$		Similarity
Sørensen-Dice Coefficient (SDC)	$SD(X, Y) = \frac{2 X \cap Y }{ X + Y }$	[0,1]	Similarity
Sørensen-Dice Using Jaccard Coef.	$J(X, Y) = \frac{SD(X, Y)}{2 - SD(X, Y)}$	[0,1]	Similarity
Jaccard Coef. Using Sørensen-Dice	$SD(X, Y) = \frac{2J(X, Y)}{1 + J(X, Y)}$	[0,1]	Similarity
Cosine Distance (CD)	$C(X, Y) = \frac{ X \cap Y }{\sqrt{ X }\sqrt{ Y }}$	[0,1]	Dissimilarity
Cosine Index (CI) (Cosine Similarity)	$CI = 1 - C(X, Y)$		Similarity
Overlapping Coefficient (OC)	$O(X, Y) = \frac{ X \cap Y }{ X }, \frac{ X \cap Y }{ Y }$	[0,1]	Similarity
Hamming Distance (HD)	$H(X, Y) = \frac{\#(X \neq Y)}{n}$	[0,1]	Dissimilarity
Hamming Index (HI)	$HI = 1 - H(X, Y)$		Similarity
Intersection	$I(X, Y) = X \cap Y $	[0, ∞]	Similarity
Kuncheva Index (KI)	$KI(X, Y) = \frac{ X \cap Y n - c^2}{nc - c^2}$	[-1,1]	Similarity
Lustgarten’s Measure (LM)	$L(X, Y) = \frac{ X \cap Y - \frac{ X Y }{n}}{\min(X , Y) - \max(0, X + Y - n)}$	[-1,1]	Similarity

5.3.2. Weight-Based Stability Measures

The results of feature selection algorithms that weights all the features, e.g., ReliefF, are expressed by vectors of feature weights. As the algorithm assigns weights to each feature, the result sets have fixed length and have the same number of the features in

the original dataset. For example, for a dataset with n number of features, weights are represented with a vector \vec{w} as $\vec{w} = \{\vec{w}_i | i = 1, \dots, n\}$. The similarity between the weight vectors is only measured by Pearson's Correlation Coefficient, which takes values between -1 and +1. The plus and minus signs indicate the direction of the relationship and the numerical values indicate the strength of the relationship. If the coefficient is less than zero, i.e., $r < 0$, it means that the two variables are moving in the opposite direction, e.g., one is increasing while the other is decreasing. If the coefficient is greater than zero, i.e., $r > 0$, it means that the two variables are moving in the same direction, e.g., both increasing. If the coefficient is equal to zero, i.e., $r = 0$, it means that the two variables are independent, i.e., not in a relationship with each other. The direction or the degree of the correlation coefficient, does not refer to the cause and effect relationship. The strength of the relationship can be classified as very weak, weak, medium, strong and very strong regarding the value of the correlation coefficient (class names and ranges can be changed). In Table 5.3, i expresses the i^{th} element of X_i and Y_i result vectors, \bar{X} and \bar{Y} express the average of the result vectors and n expresses the number of features in the original dataset, Pearson's Correlation Coefficient (PCC) formula is given.

5.3.3. Rank-Based Stability Measures

The results of feature selection algorithms that rank (scores) all the features according to a criterion, e.g., distance or relationship (correlation), are expressed by vectors of feature indexes. The similarity between these vectors can be determined with the help of Spearman's and Kendall's Rank Correlation Coefficient, Canberra Distance or Weighted Canberra Distance. The Spearman's Rank Correlation Coefficient (SRCC) or Spearman ρ (rho), expresses the uniform relationship between the rank of two discrete variables, i.e., feature rank vectors, and its strength with the correlation coefficient indicated by ρ . Like all correlation coefficients, ρ is in the range of $[-1, +1]$ and the plus and minus signs refer to the direction of the relationship, and the numerical values refer to the strength of the relationship. If the coefficient is less than zero, i.e., $\rho < 0$, means that the two variables are moving in the opposite directions, being greater than zero, i.e., $\rho > 0$, means both variables are moving in the same direction and if the coefficient is equal to zero, i.e., $\rho = 0$, means the two variables have an independent relationship.

The Kendall's Rank Correlation Coefficient (KRCC) or Kendall's τ (tau), in contrast to the Spearman's Rank Correlation Coefficient analysis, consider concordant and discordant pairs. Each observation pair (feature pair) that belong to the \bar{X} and \bar{Y} variables (two different feature ranking vectors), $(X_i, Y_i) - (X_j, Y_j)$, if $X_i > X_j$ and $Y_i > Y_j$ or $X_i < X_j$ and $Y_i < Y_j$ conditions are met, they are concordant and if $X_i > X_j$ and $Y_i < Y_j$ or $X_i < X_j$ and $Y_i > Y_j$ conditions are met, they are discordant. Concordant and discordant pairs are found for each feature pair to calculate Kendall's Rank Correlation Coefficient. Spearman's and Kendall's Rank Correlation Coefficient formulas are given in Table 5.3, where i is the i^{th} element of X_i and Y_i result vectors, n is the number of features in the original dataset and CP , concordant and DP , discordant feature pairs.

Table 5.3. Pearson's Correlation Coefficient, Spearman's Rho and Kendall's Tau

Measure	Formula	Bound
Pearson's Corr. Coef. (PCC)	$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}}$	$-1 \leq r \leq 1$
Spearman's Rho (SRCC)	$\rho = 1 - \frac{6 \sum_{i=1}^n (X_i - Y_i)^2}{(n^3 - n)}$	$-1 \leq \rho \leq 1$
Kendall's Tau (KRCC)	$\tau = \frac{\# \text{ of } CP - \# \text{ of } DP}{\frac{n^2 - 1}{2}}$	$-1 \leq \tau \leq 1$

The similarity between rank vectors can also be measured by the Canberra Distance. If the similarity between the two vectors is low, the Canberra Distance takes high values because it is inversely proportional to the rate of similarity. Therefore, the closer the value to zero, the greater the similarity. However, in some cases, considering only the top k numbers of features instead of the whole rank list may make it easier to interpret the analysis and the analysis result. In this case, Weighted Canberra Distance, which is the weighted version of the Canberra distance (Jurman et al., 2008) is used. Canberra and Weighted Canberra Distance formulas are given in Table 5.4., where i is the i^{th} element of X_i and Y_i result vectors, n is the number of features in the original dataset and k is the number of features considered, i.e., top k position of the ranked feature subset.

Table 5.4. Formulas of Canberra and Weighted Canberra Distance

Measure	Formula	Bound
Canberra Distance (CD)	$d_{CD} = \sum_{i=1}^n \frac{ X_i - Y_i }{ X_i + Y_i }$	$0 \leq CD(X, Y) \leq \infty$
Weighted Canberra Distance (WCD)	$d_{WCD} = \sum_{i=1}^n \frac{ \min(X_i, k + 1) - \min(Y_i, k + 1) }{\min(X_i, k + 1) + \min(Y_i, k + 1)}$	$0 \leq CWD(X, Y) \leq \infty$

5.3.4. Frequency-Based Stability Measures

Stability measures mentioned so far use pairwise comparison in principle. However, in addition to these measures, frequency-based measures were proposed and being used to assess feature selection stability. In this approach, unlike other stability measures, pairwise comparison is not used, instead, it uses the frequency of occurrence of a feature or a feature set for measurement. Therefore, computing costs of frequency-based measures are generally low. Measures proposed by Goh and Wong (2016), Guzman-Martinez and Alaiz-Rodriguez (2011) and Lausser et al. (2013) in their studies are examples of this kind. Since it is aimed to use various examples of stability measures in this thesis, the frequency-based measure proposed by Nogueira (2018) in her doctoral thesis is also included. Nogueira's stability measure formula is,

$$(z) = 1 - \frac{\frac{1}{d} \sum_{f=1}^d S_f^2}{\bar{k} \left(1 - \frac{\bar{k}}{d}\right)} \quad (6)$$

where d is the number of features, S_f^2 is the sample variance of the selection of the f^{th} feature and \bar{k} is the arithmetic average of the selected feature quantities. Nogueira (2018) also shared MATLAB, R and Python implementation of her measure and prepared an online stability calculator which only requires a selected feature matrix in binary form. Finally, the relevant doctoral study also provides information on other frequency-based stability measures.

5.4. Computational Complexities of Stability Measures

Since most of the stability measures (index, rank or weight) perform $(l * (l - 1)/2)$ comparisons for l number of results, their computational complexity is quite close to one another. Frequency-based measures, a binary matrix in which selected features are represented by 1 (if selected) and 0 (if not selected) is used and frequencies are calculated by traversing all columns and rows. For example, in an $N \times l$ result matrix,

where N is the number of features and l is the results (iterations), finding feature frequencies has a computational complexity of $O(N * l)$. However, if l value gets close to N , then the computational complexity increases to $O(N^2)$. In Table 5.5, the computational complexities of the stability measures used in this study are presented, where N is the total number of features in the dataset, n is the total number of features (length or quantity) in the feature subset and l is the number of results.

Table 5.5. Computational Complexities of the Stability Measures

Stability Measure	Computational Complexity
Average Normal Hamming Distance Jaccard Index Sorensen-Dice Kuncheva Index Tanimoto Overlapping Coefficient	$O\left(\frac{n(l^2 - l)}{2}\right)$
Canberra Distance Weighted Canberra Distance Pearson Correlation Coefficient Spearman's Rho Kendall's Tau	$O\left(\frac{N(l^2 - l)}{2}\right)$
Frequency Based Measures	$O(Nl)$

5.5. Open Topics on Selection Algorithm Stability

Some of the problems related to algorithm stability are considered in this section. Each of these topics can also be considered as a subject of improvement.

- The relationship between stability and performance:** It is quite difficult to prove “algorithms that have high stability values, selects feature subsets with higher accuracy rate (classification performance)”, “there is a relationship between stability value and classification accuracy” or “there is a relationship between an x stability measure and classification accuracy” arguments if you consider feature selection, feature selection stability and classification or clustering performance of the selected feature subsets concepts independently. The number of factors that effects feature selection, algorithmic stability, and classification are quite high and this subject is still open for study.
- Determination and elimination of the cause of instability:** Knowing the reasons for instability and what needs to be done to overcome these reasons are quite important in creating high-performance models. Several studies, e.g., Alelyani

(2003), Dittman et al. (2012) and Yang et al. (2013) largely consider the dataset as the cause of the algorithm instability. For example, datasets with inadequate (small) sample sizes are naturally unstable. In these studies, the effects of dataset characteristics, such as feature dependencies, number of samples and features and class distributions, on different selection algorithms are succinctly presented to the readers by conducting experimental studies. In addition to data-driven factors, instability may be based on the algorithm. For example, the stability value of the wrapper algorithms is determined especially by the classification and search algorithm to be used. Thus, detection and elimination of the cause/causes and the development of more stable algorithms by nature are subjects that are still open to be worked on.

- **New stability measures:** As mentioned earlier, feature selection algorithms represent their results in three different ways: by indexing selected features, by assigning weight (scores) to the features or by ranking the features. For this reason, stability measurement takes place in terms of index, weight or rank. At this point, set-based (index-based) stability measures have the majority with more than ten measures, while there is only one weight-based stability measure. Besides, no measure that can work on two or more result representations. This makes it difficult to compare stability measures with one another and assess the results. Lastly, as previously mentioned in section 5.2, measures have different properties, however, only one measure (Nogueria, 2018) that has all the properties mentioned has been proposed so far.
- **Stability measures for unsupervised and semi-supervised feature selection:** Studies on selection algorithm stability often concentrate on supervised problems (datasets) in feature selection. Studies related to the stability of the selection algorithm in unlabeled, semi-labeled and streaming data has not been found in the literature search.

CHAPTER 6

FEATURE SELECTION ALGORITHMS USED IN THE EMPRICAL STUDY

This chapter provides brief information about the T-Test, Bhattacharyya Distance, Wilcoxon Sum-Rank Test, ROC Curve, Entropy, ReliefF and Decision Tree Ensemble of Learners (DTE) filtering and SFS, SBS and BDS wrapper algorithms and Bayesian Hyperparameter Optimization used in the experimental stage.

6.1. Statistical Feature Selection

Instead of selecting features, filtering techniques sort them with the help of an evaluation function. Various statistical techniques (tests) can be used as an evaluation function. Statistical techniques are divided into two groups as univariate and multivariate. In this thesis, feature selection performed using parametric (works with data that have a normal distribution) and nonparametric (distribution-free) statistical tests that belong to univariate class: Two-Sample T-Test, Bhattacharyya Distance, Wilcoxon Rank-Sum Test, ROC, and Entropy. A detailed description of the tests used can be found in the book by Theodoridis and Koutroumbas (2003).

6.1.1. Two Sample T-Test

Two-Sample T-Test is a parametric test,

$$t = \frac{\mu_{1i} - \mu_{2i}}{\sqrt{\frac{\sigma_{1i}^2}{n_1} + \frac{\sigma_{2i}^2}{n_2}}} \quad (7)$$

where μ_{ji} and σ_{ji} are the sample average and variance of i^{th} feature of the j^{th} ($j = 1, 2$) class and n_1 and n_2 are the sample sizes. To perform feature selection using the Two-Sample T-Test, firstly t -value of each feature is calculated according to the formula. Features with higher t -values are effective in the classification. In other words, they are the relevant features. Then, after ranking all features according to their t -value from largest to smallest, features are selected using a user-determined threshold value

(selection criterion). Two-Sample T-Test can be applied to classes with normal distribution, homogeneous variance and includes no outlier samples.

6.1.2. Bhattacharyya Distance

Bhattacharyya Distance is a parametric test¹,

$$d_B = \frac{1}{4} \ln \left(\frac{1}{4} \left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} + 2 \right) \right) + \frac{1}{4} \left(\frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \right) \quad (8)$$

where μ_j and σ_j are the average and variance of the j^{th} ($j = 1,2$) class. Bhattacharyya distance gives the divergence between two different classes (variance) in the metric distance, i.e., taking values in the range between 0 and infinite. Therefore, the larger the test value, the more different the two classes. Bhattacharyya distance can be applied to classes with normal distribution, homogeneous variance and includes no outlier samples.

6.1.3. Wilcoxon Rank-Sum Test (or Mann-Whitney U Test)

In contrast to other methods, Wilcoxon Rank-Sum Test is a non-parametric or distribution-free hypothesis test that is used to compare independent samples (data) drawn from populations (classes) with a non-normal distribution and cannot be transformed into a normal distribution with logarithmic transformation. In general, the Wilcoxon Rank-Sum Test is based on forming a ranked list by ranking values of each feature in the dataset in ascending order, beginning from 1 for the smallest value, according to their places. If k number of observations are tied for the i^{th} rank, then each one gets assigned a value using the formula $(i + \frac{k-1}{2})$. Then the list is divided into two regarding the class labels, and ranks of the observations for each class are added. Results are named as W_1 and W_2 , i.e., the summation of ranks for the first and the second classes, and p value is determined by the W values and number of observations, n_1 and n_2 , per class. There is a W distribution table prepared for datasets with 20 or fewer observations. By this means, p value can be determined directly with the help of the table. For observations above this number, p value is determined using the normal distribution (Z-Statistics). The Wilcoxon Rank-Sum Test is resistant to outliers and noisy values as other rank-sum tests. However, this method can only be

¹ Simplified version of Bhattacharyya Distance for two class (binary) problems.

used for two-class problems. For problems with more than two classes, Kruskal-Wallis Rank-Sum Test or “One vs. All” strategies may be preferred.

6.1.4. ROC (Receiver Operating Characteristic) Curve Test

The ROC curve is mainly used to represent the classification performance rate graphically. Y -axis of the graph is the true positive ratio (TPR) and the x -axis is the true negative ratio (TNR), the relationship between the sensitivity and the selectivity of the classifier is expressed. However, it is also possible to use the ROC Curve for feature selection. To do this, the probability distribution function of the features according to the classes (note that the probability density function is used for continuous random variables) needs to be considered. If the distribution of the two classes overlaps, the ROC Curve takes a form of a line ($y = 1$) between the coordinates (0,1) and (1,1). As the distributions begin to separate from one another, they slowly begin to form a curve. At this point, the area between the curve and ($y = 1$) line or between the curve and the random classifier ($(x = y)$ line) is examined. If the line is considered, the distributions are overlapping when the area is equal to 0 and the distributions are completely disjoint when the area is equal to 0.5. If the random classifier curve is to be considered, the opposite would be the case. Thus, the class separation capability of any feature is determined.

6.1.5. Entropy (or Shannon Entropy) Test

Entropy is essentially a measure of the uncertainty of a probabilistic distribution. It is a nonparametric test,

$$Entropy = - \sum_{i=1}^n p_i \log_2(p_i) \quad (9)$$

where n is the number of classes and p_i is the ratio of the frequency of the i^{th} class to the total number of samples (probability of the i^{th} class). Entropy value is equal to 0 if all the samples belong to the same class, 1 if they are divided equally between classes, and in between 0 and 1 if they are randomly distributed. Thus, the closer this value is to 1, the better. However, entropy alone is not enough for feature selection. To determine which feature is more important in the classification, it is necessary to consider the information gain provided by all the features. Starting from the high gain

provider, ranked features are then selected based on the threshold value. The information gain of a feature, f , in a class, X , is,

$$Gain(X, f) = I(X, f) = Entropy(X) - Entropy(X, f). \quad (10)$$

6.2. Relief and ReliefF Algorithms

Relief is a supervised feature selection algorithm developed by Kira and Rendell (1992) to rank features by assigning feature weights. However, Relief can only work on two-class (binary) datasets. Therefore, Kononenko (1997) proposed a multiclass version of the Relief and named as ReliefF. ReliefF can work on noisy, incomplete, discrete and continuous datasets and can detect conditional dependencies between features (independent variables). It can be used in both classification and regression (named RReliefF) problems. ReliefF assigns a weight ranging from -1 to 1 to all the features, like its ancestor, and feature selection is performed regarding these values. The features with a weight close or equal to 1 are effective (relevant features) in classifying. Whereas, the features with a weight close or equal to -1 are ineffective in classifying (noisy, redundant and/or irrelevant). Note that zero is the initial value of each feature. The Relief algorithm (Figure 6.1) starts by setting all the feature weights to zero (the first step in Figure 6.1). In the second step, it selects a random sample from the sample space (step 3 in Figure 6.1) and checks its class. In the third step, the algorithm finds two samples (instances) closest to the selected sample from the same class (named nearest hit, H) and the other class (named nearest miss, M) using the Euclidean distance (step 4 in Figure 6.1). Finally, the weight calculation is performed (step 6 in Figure 6.1) in two different ways for discrete,

$$diff_{Discrete}(a, I_1, I_2) = \begin{cases} 0 & \text{if } value(a, I_1) = value(a, I_2) \\ 1 & \text{if otherwise} \end{cases} \quad (11)$$

and continuous,

$$diff_{Continuous}(a, I_1, I_2) = \frac{|value(a, I_1) - value(a, I_2)|}{max(a) - min(a)} \quad (12)$$

features. However, the purpose of both calculation methods is to determine the changes in the feature values. Normalization is also performed in this step to ensure the results to be in the range of $[-1,1]$. From the second step, the algorithm repeats the steps as many times as the user determines, for example, m times. When the algorithm terminates, the weight value of each feature would be determined. Features are ranked by these values if necessary and are selected based on a threshold value specified by the user.

1	for $n = 1$ to N (<i>Number of features</i>)
	Set all feature weights $w[n]$ to 0.0
2	for $i = 1$ to m (<i>User defined parameter</i>)
3	Select an random instance R_i
4	Find nearest H and nearest M
5	for $a = 1$ to N (<i>Number of features</i>)
6	$w[a] = w[a] - \frac{\text{diff}(a, R_i, H)}{m} + \frac{\text{diff}(a, R_i, M)}{m}$
7	end

Figure 6.1. Pseudocode of the Relief Algorithm
(Robnik-Sikonja and Kononenko, 1997)

Unlike its predecessor, the ReliefF (Figure 6.2) algorithm is capable of working on both multiclass and incomplete datasets. However, both algorithms work-alike (note that feature weight measurement is the same in both algorithms, see Formulae 11 and 12). ReliefF algorithm finds k number of closest neighbors using Manhattan distance, instead of finding the two closest neighbors. Thus, the algorithm is less affected by unnecessary and noisy data.

1	for $n = 1$ to N (<i>Number of features</i>)
	Set all feature weights $w[n]$ to 0.0
2	for $i = 1$ to m (<i>User defined parameter</i>)
3	Select an random instance R_i
4	Find k nearest H_j
5	for each class $C \neq \text{class}(R_i)$
6	from class C find k nearest misses $M_j(C)$
7	for $a = 1$ to N
8	$w[a] = w[a] - \sum_{j=1}^k \frac{\text{diff}(a, R_i, H_j)}{m k}$ $+ \sum_{C \neq \text{class}(R_i)} \left[\frac{P(C)}{1 - P(\text{class}(R_i))} \sum_{j=1}^k \frac{\text{diff}(a, R_i, M_j(C))}{m k} \right]$
9	end

Figure 6.2. Pseudocode of the ReliefF Algorithm

(Robnik-Sikonja and Kononenko, 1997)

The user-specified k value is directly involved in the determination of the feature weights and consequently with the selection. Therefore, it is important to find an optimal k value. However, this is not easy. In the literature, there are two generally accepted approaches used for finding this value. The first approach is to take the square root of the number of samples that the current dataset has, for example, $k = \sqrt{100} = 10$, for 100 samples. The second approach is to start the k value from 10 and find an optimal value by trial and error. In the scope of this study, search methods were used to find the optimal k value, and these methods are mentioned in the next chapter.

Both Relief and ReliefF algorithms also require a user-specified number of iterations and a threshold value. The number of iterations, which is not a fixed value, is determined by the user according to the number of samples in the dataset. Since the algorithm will select as many samples as the number specifies, selecting the number of iterations more than the number of samples will not affect the result. It is also possible to consider the number of iterations as a percentage of sampling. For example, the number of iterations for 10% of 100 samples can be selected as 10.

As mentioned before feature filtering algorithms, such as Relief and ReliefF, do not select features. Feature subsets are created based on a threshold value. Therefore, the threshold value is an important hyperparameter that directly affects the feature selection, as the k value. This value, although there are a variety of approaches such as averaging or using Chebyshev Inequality, is generally determined intuitively and

can be re-determined according to the classification performance of the obtained feature subset. However, this way is rather time-consuming and prone to error for problems with high values. Therefore, another method that can be used for algorithms that produce results as weights, is to consider the amount of increase between feature weights. The point with an increase more than normal can be considered as a threshold value. However, this method is difficult to use in result sets with close weight values or multiple increment points.

6.2.1. Methods Used to Determine the Ideal K Value

As mentioned in the previous chapter, the ReliefF algorithm requires a user-defined k value that can directly affect the result. Therefore, finding the ideal k value is quite important in terms of algorithm performance. However, since the k value depends on the dataset (it is not a fixed value) and it must be recalculated for each dataset to be studied. Various methods are being used to calculate the k value. Some of these methods are, such as setting the k value to the square root of the number of samples, generally accepted intuitive and practical approaches. Within the scope of this study, the optimal k value was determined using brute force and genetic algorithm.

1. **Brute Force (or Exhaustive) Search:** Brute force search is based on the principle of testing all the possible values of the k hyperparameter one by one. Therefore, it is time-consuming, especially in cases where the number of records (observations) is high. However, brute force search is frequently preferred for parameter optimization in machine learning algorithms, e.g., setting k values for K-Means and K-NN algorithms, since it can be easily adapted and applied to many problems. In addition to this, the brute force search can guarantee to find the global optimum solution, unlike random search. Brute force search can be performed in two ways.
 - **Searching forward:** With the help of a loop statement, starting from 1, k value is incremented by one in each iteration until it reaches the total number of samples, n , ($k = 1, 2, 3, \dots, n$). Meanwhile, all feature weight values are observed. The k value, in which all the weights become constant (stable), is taken as the optimum solution.
 - **Searching backward:** With the help of a loop statement, starting from the total number of samples, n , k value is decremented by one in each iteration until it

reaches 1, ($k = n, n-1, n-2, \dots, 1$). As the same with forward search, all feature weight values are observed. But this time, k value, in which all the weights start to change, is taken as the optimum solution.

Both approaches have $O(n)$ complexity where n is the number of observations or the size of the search space. The search results can be expressed by using an $n \times \vec{W}$ matrix, \vec{W} is the vector that contains the weight values of the features and n is the number of observations or feature weights versus k value graph (see Figure 6.3). Brute force search is quite costly, especially in cases where the number of samples is high. Therefore, instead of increasing or decreasing the k value one by one, increasing or decreasing it by multiple of tens or higher values in each iteration can reduce the time and the processing power needed to find the solution. However, in this case, in order to determine the exact value, it is necessary to perform a brute force search again at the relevant range.

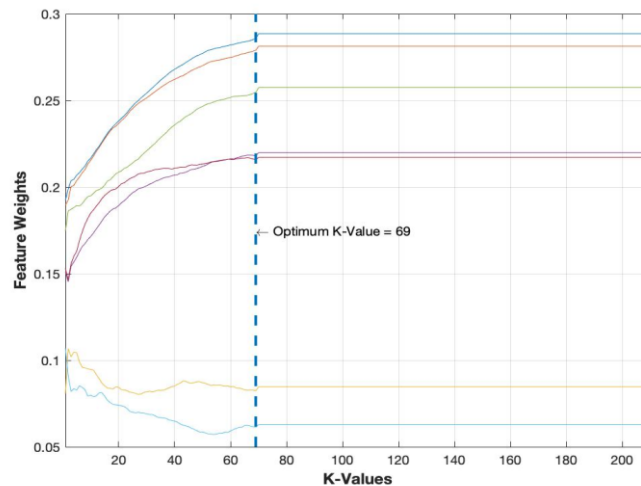


Figure 6.3. Feature Weights vs. K-Values Graph

- Genetic Algorithm Search:** Genetic algorithms, inspired by Darwin's theory of evolution, are the techniques used to find complete and/or approximate solutions for optimization and search problems using biological processes such as reproduction, selection, mutation, and crossover. Genetic algorithms start working with a randomly generated result set called population, and in each iteration (generation) by improving the quality of the population try to find the best solution. In general, to find the best solution, each individual in the population is evaluated according to a pre-defined fitness function. Thus, good solutions (individuals) are

selected and crossover with other selected individuals to transfer their characteristics, i.e., genes, to the next generation. This process usually ends when the maximum number of generations specified by the user is met or when a satisfactory level of success is reached.

In order to determine the ideal k value, it is necessary to handle integer-valued constraints. Therefore, genetic algorithm parameters must be set to perform integer constraint optimization. As in the same with brute force search, after setting the lower bound of the k value to 1 and the upper bound to the number of observations, n , (algorithm will perform integer search between 1 and n) in each iteration fitness function, ff , is evaluated for each k value to check the feature weights. This process ends when the maximum number of generations is reached. As a result, the ideal k value and the corresponding weight vector, \vec{W} , are obtained. More detailed information on integer and decimal constrained optimization problems can be found in the publication of Deep et al. (2009).

Genetic algorithms operate in parallel by their nature and are easily adaptable to many problems. However, since genetic algorithms are population-based, they may be as time-consuming and costly as the brute force searches in some problems. In addition to this, the parameters required by the algorithm, for example, the mutation operator or the stopping criterion, may not be easy to detect since they are usually dependent on trial and error. Finally, unlike the brute force search, they may not always guarantee to find the best global solution in the search space. However, they may find the best local solution. In addition to these methods, Simulated Annealing, Hill Climbing, and Tabu Search can be used to determine the ideal k value.

6.3. Feature Selection Through Ensemble Learning

It is possible to perform feature selection through ensemble learning. Ensemble learning is a classification method based on the principle of creating a strong classifier using more than one individually trained weak classifiers, i.e., collection of learners. The weak classifier is the name given to classifiers that have a slightly better performance than random, i.e. depending on luck, classifiers. Although each classifier is weak by itself, high classification performance and easy scalability for high volume datasets are achieved when they form an ensemble. In addition to these, the final result is not depended on the performance of a single classifier.

After creating the ensemble learning using strategies such as different learning algorithms, different parameters of the same algorithm, different training sets, etc., the results obtained are combined with techniques such as the majority and weighted voting for classification problems and the average and weighted average for the regression problems. The most widely used ensemble learning techniques are Bagging, Boosting, Stacking and Random Forest.

- **Bagging (Bootstrap Aggregation):** The bagging method generally consists of three steps. In the first step, training sets are created by using simple random sampling with replacement (SRSWR) selection procedure. Generated datasets have the same size as the original datasets. In the second step, each training set is given as an input to a classifier and the training of the classifiers is carried out in parallel. In the last step, the classifier results are combined using majority voting and the final result is obtained. Since Bagging is a variance reducing technique, it is suitable for high variance with low bias datasets.
- **Boosting:** Boosting method generally consists of four steps. In the first step, as the same with Bagging, training sets are created by using simple random sampling with replacement (SRSWR) selection procedure. In this step, an equal weight value is assigned to each sample. In Boosting, classifiers work in series instead of parallel. Therefore, the output of a model becomes the input of the next model. In the second step, training is performed by giving the first training set to the classifier. In the third step, faulty sample weights are updated according to the errors made in classifying. Weighted samples have a higher chance to be selected for the next time. Process returns to the beginning and repeats as much as the number of predetermined classifiers. In the last step, the classifier results are combined with a majority voting and the final result is obtained. Since Boosting is a bias reducing technique, it is suitable for high bias, low variance datasets.
- **Stacking:** Stacking method consists of three steps. The first step is the same as Bagging and Boosting. In the second step, the training of the classifiers is carried out in parallel by giving each training set as an input to a classifier, as in Bagging. In the last step, the selection process is performed. However, instead of voting, the selection process is performed by giving all the classifier results as an input to another classifier, also named as meta classifier, that was not used during the learning process.

- **Random Forests (Random Decision Forests):** The Random Forests method constructs various Decision Trees to perform feature selection in three steps. The first step is to obtain samples of the same size as the original dataset, by using simple random sampling with replacement (SRSWR) selection procedure. This step is followed by the creation of the trees. Generated training sets are distributed in a manner that each Decision Tree gets a different set. Each tree that forms the Random Forest works in parallel and independent from one another, i.e., trees cannot interfere with each other. Because the pruning is not performed, large trees may form. The main purpose of the pruning is to reduce the unnecessary complexity, to increase the estimation power and to avoid the overfitting problem. Random Forest algorithm, however, by randomly selecting both the training set (bootstrap aggregation) and the features that form the tree (random subspace) prevent the overfitting issue. Therefore, the pruning is not performed on the trees used. The last step is to combine the results. Combining is performed by taking majority voting for the classification and the regression is performed by taking the average of the results.

Decision Trees allow the measurement of the relative importance of each feature on classification due to its structure. Therefore, they can be used together with the above-mentioned methods as an ensemble feature selection method. Decision trees are not mentioned again since they have been mentioned in the section of classification algorithms. Within the scope of this thesis, an ensemble learning algorithm was created by using decision trees and used in feature selection as a filtering technique.

6.4. Sequential Feature Selection

Wrapper methods, as mentioned earlier, carry out the selection process in three steps. These are: forming a feature subset, determining the classification performance of the formed subset and evaluation, respectively. The biggest problem encountered in this process is forming the feature subsets, because the increase in the number of features, also increases the number of features that can be created exponentially and rarifies the feature selection process. In order to overcome this problem, exponential, sequential and randomized search methods are integrated into the wrapper algorithms. In this thesis, sequential search techniques are used. Sequential Forward Selection (SFS), Sequential Backwards Selection/Elimination (SBS or SBE) and Bidirectional Search

(BDS) algorithms perform sequential search and evaluate all elements within the feature set based on the classification performance, one at a time. They start searching in three different ways, either with an empty feature set, or a complete feature set or with both empty and complete sets at the same time. Therefore, they are divided into three different groups.

- 1. Sequential Forward Selection Algorithm (SFS):** The Sequential Forward Selection (see Figure 6.4) is a bottom-up search strategy, which starts searching with an empty feature set and adds the feature, which increases the classifier's accuracy to the feature subset in each iteration. The search continues until there is no increase in classifier performance. Features added to the feature subset cannot be removed.

1	$i = 0$
	<i>Start with an empty feature set $Y_i = \{\emptyset\}$</i>
2	<i>do</i>
3	<i>Select the best feature $x = \operatorname{argmax}_{x \notin Y_i} J(Y_i + x)$</i>
4	$Y_{i+1} = Y_i + \{x\}$
5	$i = i + 1$
6	<i>while</i> ($J(Y_i + x) > J(Y_i)$)

Figure 6.4. Pseudocode of the SFS Algorithm

- 2. Sequential Backward Selection/Elimination Algorithm (SBS or SBE):** The Sequential Backward Elimination (see Figure 6.5) is a top-down search strategy, which starts searching with the complete feature set and extracts the feature, which reduces the classifier's accuracy from the feature subset in each iteration. The search continues until there is no decrease in classifier performance. Features removed from the feature subset cannot be added back again.

1	$i = 0$
	<i>Start with all features $Y_0 = \{1, \dots, N\}$</i>
2	<i>do</i>
3	<i>Select the worst feature $x = \operatorname{argmax}_{x \in Y_i} J(Y_i - x)$</i>
4	$Y_{i+1} = Y_i - \{x\}$
5	$i = i + 1$
6	<i>while</i> ($J(Y_i - x) > J(Y_i)$)

Figure 6.5. Pseudocode of the SBS Algorithm

3. Bidirectional Search Algorithm: In this method, SFS and SBS algorithms work in parallel. In the search space, both algorithms try to reach the same result set without changing algorithms search strategies. SFS algorithm is searching upside down while the SBS algorithm is searching from top to bottom. Algorithms are not allowed to interfere with each other. Therefore, the features selected by the SFS algorithm, cannot be removed from the feature subset by the SBS algorithm. Likewise, the features eliminated by the SBS algorithm cannot be added back to the feature subset by the SFS algorithm. The BDS algorithm is given in Figure 6.5.

	$i = 0$
1	Start with an empty feature set $F_i = \{\emptyset\}$ Start with all features $B_0 = \{1, \dots, N\}$
2	do
	Select the best feature
3	$x = \operatorname{argmax}_{x \notin F_i \wedge x \in B_i} J(F_i + x)$
4	$F_{i+1} = F_i + \{x\}$
	Remove the worst feature
5	$x = \operatorname{argmax}_{x \notin F_{i+1} \wedge x \in B_i} J(B_i - x)$
6	$B_{i+1} = B_i - \{x\}$
7	$i = i + 1$
8	while($F \neq B$)

Figure 6.6. Pseudocode of the BDS Algorithm

SFS, SBS and BDS algorithms can be used for both classification and regression problems. In either case, they produce results in feature subsets. All sequential feature selection algorithms are greedy algorithms, which choose the closest alternative to the target and try to improve it in each step and reach the result. In other words, they try to find the global optimum solution by using the local optimum solutions. However, greedy approaches do not guarantee to find the global optimum, i.e., best, solution because of the risk of being stuck on the local optimum solutions in the search space. This problem also called the nesting effect, is likely to occur since there is no backtracking mechanism in sequential feature selection algorithms. Besides their drawbacks, greedy approaches are fast, have low computational needs and easy to implement. Lastly, it is important to note that instead of SFS and SBS algorithms, SFFS (Sequential Floating Forward Selection) and SFBS (Sequential Floating Backward Selection) algorithms that have backtracking capabilities can be used.

6.5. Hyperparameter Optimization

The aim of the optimization is to minimize or maximize a target (objective) function, $f(x)$, $\operatorname{argmin}_x f(x)$ or $\operatorname{argmax}_x f(x)$, by adjusting the variables on hand according to various constraints (conditions) if any, and ensuring the valuable resources such as labor, time, cost, equipment to be used with maximum efficiency. In supervised feature selection, optimization techniques are generally used to determine the selection algorithm parameters and hyperparameters. For example, the ReliefF algorithm requires an integer k value that maximizes all feature weight values. Optimum k value can also be found by navigating through the entire search space individually with the help of a brute force search. However, when the number of samples and features is high, this method has a high computational cost. Therefore, in order to reduce computational costs Bayesian (or Bayes) Hyperparameter Optimization that tries to minimize or maximize deterministic or stochastic scalar target function, can be used.

Algorithm performance is increased by detecting the most appropriate parameters and hyperparameters for the problem. This process is called parameter or hyperparameter optimization. The parameter or model parameter is the properties or variables that the algorithm detects and uses, using the dataset during the training process. The determination of weights in Artificial Neural Networks by the algorithm can be given as an example of a model parameter. The hyperparameter or model hyperparameter is properties or variables that cannot be detected by the algorithm by using the dataset during the training process. Thus, these variables are given to the algorithm by the user, such as the k value that is needed by the ReliefF algorithm. Hyperparameters, like parameters, can take continuous, discrete or binary values. In this thesis, classification algorithms used by wrapper algorithms were first run with default values and then with values determined with Bayesian Hyperparameter Optimization.

The Bayesian Hyperparameter Optimization which essentially is a black box optimization method (used in situations where the structure of the target function is not clearly defined), is using the Gaussian process which is a probabilistic process to determine the values that maximize the performance of classification or regression algorithms. With the Gaussian process, firstly a probabilistic model, such as minimizing the error rate of the target function, is created. Then this model and a gain function, such as expected improvement or probability of improvement, are used to

find the best possible hyperparameters and tried on the actual (target) function. The probabilistic model is trained (updated) according to the obtained values. These steps are repeated considering the stopping criterion, e.g., no change in the error rate, a certain number of iterations or a time limit. Bayesian Optimization considers past results in contrast to the grid and random search. Therefore, it has better performance.

The aim of all the problems encountered in real life is to produce the highest quality solutions with the shortest possible time and a minimum cost. But it is often difficult for all of these to happen at the same time. It is likely to result in a gain in one point and a loss on the other. For example, while the optimization of classification and regression algorithms provide performance improvement, the computational cost of this process can increase exponentially regarding the dataset studied. In addition to this problem, the algorithm optimization depends on the dataset. Even if the algorithm does not change, algorithm variables (hyperparameters) should be optimized again when the dataset is changed. Therefore, algorithm optimization adds an extra layer and costs to the analysis process. Finally, Bayesian Optimization is one of the techniques that can be used. Apart from this method, Grid Search, Random Search, Gradient-Based and Evolutionary Hyperparameter Optimization techniques can also be used.

CHAPTER 7

EMPRICAL STUDY

In the scope of this study, the relationship between the stability and the classification, i.e. prediction, performance of feature selection algorithms, were evaluated and compared experimentally with T-Test, Bhattacharyya, Wilcoxon Rank-Sum, ROC Curve and Entropy Tests and ReliefF, Decision Tree Ensemble Learning filter algorithms and Sequential Forward Selection (SFS) and Sequential Backward Selection/Elimination (SBS) wrapper algorithms. The reason why the experiments were conducted using filter and wrapper algorithms is that the filter methods are faster and computationally less intensive than other methods, perform model-independent feature selection and resistant to overfitting and the wrapper algorithms can be used with different search techniques and/or classification algorithms. However, in order not to lose the focus on the main topic of the thesis, only the classification algorithms used by the wrapper algorithms have been changed by keeping the search technique (greedy search) constant.

The theoretical framework of this study is prepared to explain whether there is a relationship between feature selection stability and classification performance of the selected features by using various datasets and feature selection algorithms. Since, it is not directly possible to assess any possible relationship between stability and classification performance, first feature selection algorithms in terms of stability and then selected features in terms of classification performance measures were compared. Feature selection stability was evaluated with various stability metrics to create grounds for a fair comparison. This also gives an idea to the readers on how to evaluate the stability of feature selection algorithms sensibly. Comparisons of the results were carried out first within the same selection approach method, and then with the other method. Experiments are conducted on seven real-world datasets, with different instances and feature sizes and the number of classes, gathered from the UCI and Kaggle Machine Learning Repository, according to the frequencies of being downloaded and preferred in the studies examined during the literature overview. The properties of datasets used are listed in Table 7.1 and sorted according to the number of classes in ascending order.

Table 7.1. List of Datasets Taken from the UCI and Kaggle Machine Learning Repository and Their Properties

Dataset Name	Feature Data Type	Features and Instances	Number of Classes	Number of Class Instances	Class Ratios
Abalone	Categorical	8—4177	3	1528	36.58%
	Integer			1307	31.29%
	Real			1342	32.13%
Australian	Categorical	14—690	2	307	44.49%
	Integer			383	55.51%
	Real				
Breast Cancer	Real	9—683	2	444 239	65.01% 34.99%
Sat	Integer	36—6435	6	1533	23.82%
				703	10.92%
				1358	21.10%
				626	9.73%
				707	10.99%
				1508	23.43%
Seeds	Real	7—210	3	70	33%
				70	33%
				70	33%
Vehicle	Integer	18—846	3	411	48.58%
				217	25.65%
				218	25.77%
Wine	Integer	13—178	3	59	33.15%
	Real			71	39.89%
				48	26.97%

Before performing feature selection, the properties, i.e., characteristics, of the datasets are analyzed and summarized using descriptive statistics, i.e., summary statistics (see Appendix 2). To determine the shape of the class distributions: skewness and kurtosis, to measure central tendency: mode, median and average and to measure variation: range, variance and standard deviation measures are used. The results are tabulated and visualized using histograms and scatter plots. In addition to these, missing values, abrupt changes, and outliers were also checked in this step because they can significantly affect the feature selection process negatively. All codes are implemented in MATLAB (R2018a/b and R2019a) using Statistics and Machine Learning and

Bioinformatics Toolbox. The calculations are performed on Intel Core i7 CPU running at 2.8 GHz with 16GB of onboard 1600 MHz RAM and 64-bit macOS Mojave operating system. The general, framework of the empirical study is depicted in Figure 7.1.

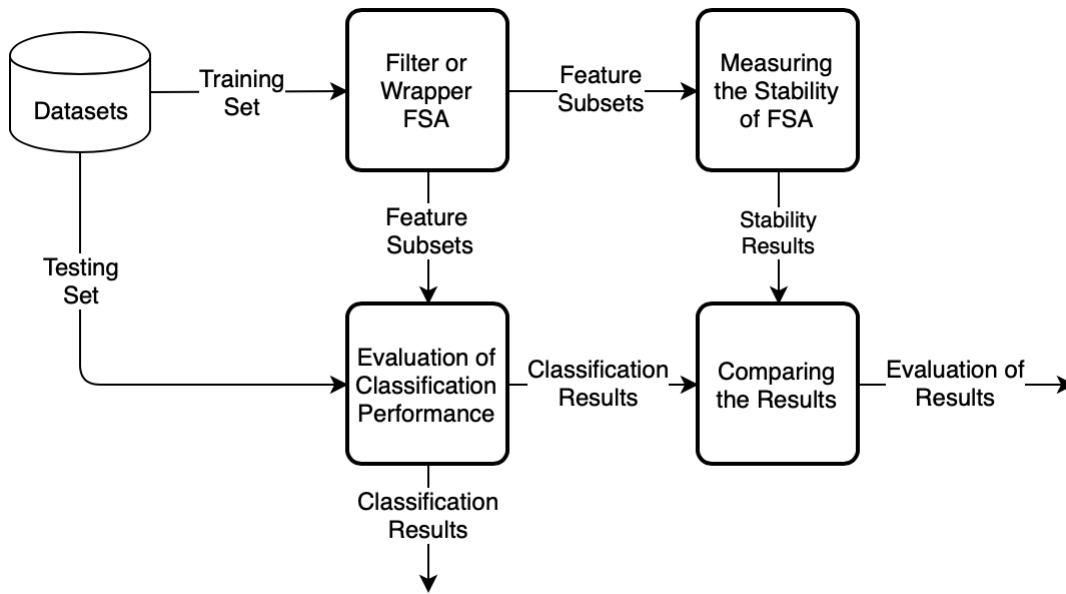


Figure 7.1. General Framework of the Empirical Study

7.1. Empirical Study on Filter Algorithms

Filter algorithms used in empirical study rank and sort (order) features using different evaluation functions (strategies). For example, T-Test, Bhattacharyya and Wilcoxon Rank-Sum tests use statistical hypothesis tests, Entropy test uses mutual information, ReliefF algorithm uses the weighted version of K-Nearest Neighbor (K-NN) algorithm, Ensemble of Learners uses Decision Trees, and ROC Curve uses the area between the ROC Curve and the random classifier slope ($(x = y)$ line). In order to perform statistically significant experiments, all the algorithms were run ten times using stratified random sampling. Stratified random sampling was used both to make changes in the training set and to represent and obtain samples (data) from each class, i.e., to reduce variances caused by the imbalanced class sample distributions. It should be noted that stratified sampling is almost identical to simple random sampling when the number of classes is high. The minimum stratified sample size was calculated and given in Table 7.2 for each dataset using the formula,

$$\text{Sample Size (Finite Population)} = \frac{\frac{z^2 p(1-p)}{e^2}}{1 + \left(\frac{z^2 p(1-p)}{e^2 N}\right)} \quad (13)$$

where z is the Z-Score², p is the population proportion, e is the margin of error and N is the population. This formula can be used for datasets with the normal distribution. However, it can be applied for the datasets besides the normal distribution by using Central Limit Theorem. Central Limit Theorem expresses that, as the number of samples taken from a non-normal distribution increases, the distribution of the sample means approaches to a normal distribution. Therefore, it is necessary to keep the sample quantity as much as possible on the left of right-skewed datasets. It should be noted that for unlimited populations only quotient part of same the formula is used.

Table 7.2. Minimum Number of Samples Taken from Datasets in Each Run (Iteration)

Dataset	Minimum Number of Samples
Abalone	380
Australian	260
Breast Cancer	260
Sat	400
Seeds	150
Vehicle	280
Wine	150

After the features are ranked, the results were visualized using the bar graph that contains all the feature indices on the x -axis and the criterion value (statistical test results or feature weights) used for ranking on the y -axis. As mentioned before, feature filtering techniques require a threshold value to form feature subset(s) (or to select features). In order to determine this value, the average of scores (weights) assigned to each feature by the evaluation function over ten runs were taken and a heat map is created. In this way, the color changes observed on the heat map have helped to determine the threshold values, i.e., points. As the final step, the classification performance of the subset(s) was/were tested. Figure 7.2 depicts T-Test results after

² Z-Score depends on the desired confidence interval, in this study it is 95%.

three runs in a 2D bar graph and a heat map created by averaging criterion values. As is seen, feature ranking performed using the data diversity ensemble strategy. Each algorithm was executed ten times with the same parameters on the same dataset and obtained feature weights were sorted according to their weights to create heat maps and perform feature subset creation.

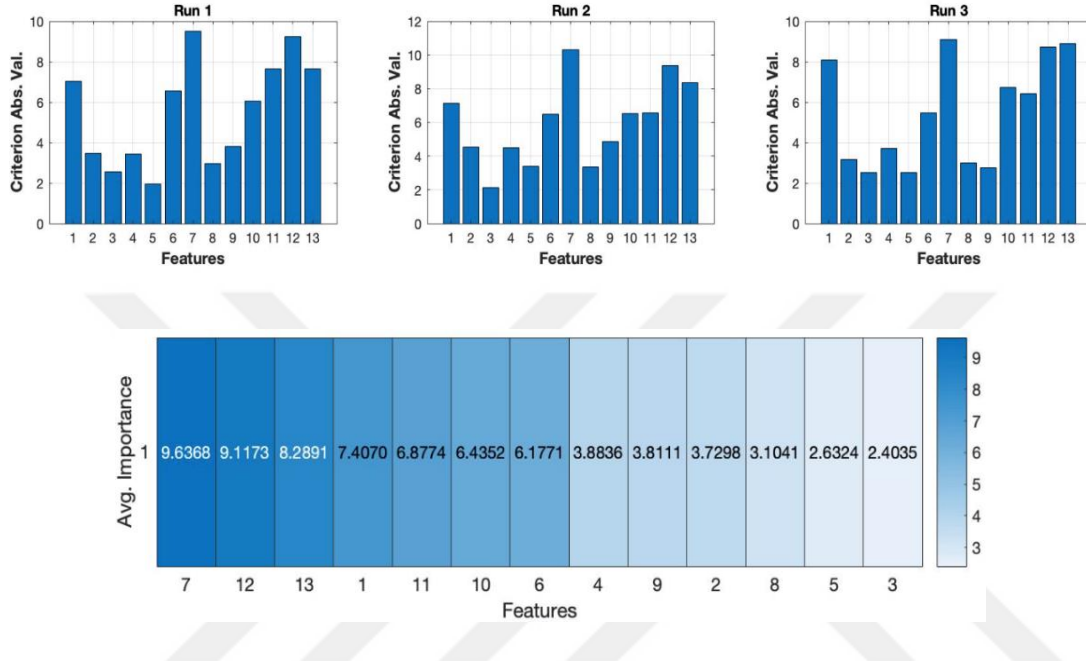


Figure 7.2. Screenshot of T-Test Results After Three Runs

Stability of filter algorithms was evaluated using rank-based (Canberra Distance, Weighted Canberra Distance, Pearson Correlation Coefficient, Spearman's and Kendall's Rank Correlation Coefficients) measures and tabulated to simplify the comparison. To average correlation coefficients, they were transformed into stability values using Fisher's Z Transform. Firstly, all the correlation coefficients were converted from r -value to z -value using the following formula,

$$z_i = \tanh^{-1}(r_i) = \frac{1}{2} \ln \left(\frac{1 + r_i}{1 - r_i} \right) \quad (14)$$

and then after taking the arithmetic average of the z -values, obtained values again converted to r -values using the following formula,

$$r = \tanh(\bar{z}) = \frac{e^{2\bar{z}} - 1}{e^{2\bar{z}} + 1} \quad (15)$$

to determine the algorithm's stability value. Therefore, two separate MATLAB functions were implemented to calculate the stability values of filtering techniques. The first function is used to calculate stability measures and the second function is used to average rank correlation coefficients. Table 7.3 is an example of the stability results table. Stability measures such as Canberra Distance and Weighted Canberra Distance don't have an upper bound. Therefore, feature scaling techniques such as min-max normalization can be used to limit the results between 0 (selected feature sets are the same) and 1 (selected feature sets are completely different). Besides min-max normalization, standardization, mean normalization and unit length scaling can be used as a feature scaling technique. Average stability values of filtering techniques in terms of different measures over ten runs on the Vehicle dataset are tabulated in Table 7.3 and given as an example (see Appendix 4 for all tables). It should be noted that the results of Canberra Distance and Weighted Canberra measures are not normalized.

Table 7.3. An Example of Average Stability Results Table of All Filtering Techniques on Vehicle Dataset

Ranking Algorithm	CD	WCD (Top 5 Features)	PCC	SRCC	KRCC
TTest	2.9904	1.2554	0.7809	0.7560	0.6042
Entropy	2.7297	0.9318	0.8956	0.7797	0.6089
Bhattacharyya	2.7878	1.1242	0.8548	0.8040	0.6209
ROC	2.0179	0.4834	0.9214	0.8583	0.7108
Wilcoxon	2.9199	0.8895	0.8863	0.7129	0.5539
ReliefF	3.2138	1.5520	0.7927	0.7765	0.6022
Decision Tree Ensemble	0.2555	0.0571	0.9991	1.0000	1.0000

7.2. Empirical Study on Wrapper Algorithms

As mentioned earlier, wrapper algorithms require a classifier for feature selection. In this thesis, in order to see the effect of different classifiers on the feature selection process, Linear Discriminant Analysis, K-Nearest Neighbor and Naïve Bayes classifiers (induction algorithms) were used as the evaluation function for the Sequential Forward Selection and Sequential Backward Selection. It should be noted that additional classifiers, e.g., ECOC classifier can also be used. As mentioned before, the ECOC classifier is an ensemble learning (meta) method of binary classifiers, i.e.,

learners, such as Discriminant Analysis, K-Nearest Neighbor, Naïve Bayes, Decision Trees, SVM, Linear Classifier, etc., using “One vs. One” or “One vs. Rest (All)” strategy to solve multiclass problems. Table 7.4 represents a list of learners which can be used in ECOC and Ensemble Learning.

Table 7.4. List of Learners for the ECOC and the Ensemble of Learners

Classifier	Learners
ECOC	Discriminant Analysis
	K Nearest Neighbor
	Linear Classifier
	Naive Bayes
	Support Vector Machines
	Classification Trees
Ensemble of Learners	Decision Tree
	Discriminant Analysis
	K Nearest Neighbor

As in filter methods, each classifier used with the wrapper algorithms was run on the same dataset ten times to assure statistical significance. Therefore, during the experiments, each classifier was run two hundred times in total, first with the default parameters, then with the hyperparameters determined by the Bayes Hyperparameter Optimization (see Appendix 1) to observe the changes in stability values and classification performance. In this process, stratified 10-Fold Cross-Validation was used both to make changes in the training set and to represent and obtain samples (data) from each class, i.e., to reduce variances caused by the imbalanced class sample distributions across each fold. Here it must be noted that, as mentioned in chapter 6.4, both SFS and SBS algorithms use a greedy search strategy but with different starting conditions (search directions). Different search strategies can be applied to select features however, it won’t be objective to compare different search families. For this reason, only the greedy search strategy was used in the experiments conducted with wrappers.

In order to increase ease-of-interpretation, results are visualized using graphs and tables. Numbers of elements selected in each run, error rates (Misclassification error or MCE), the change of accuracy and frequency of selected features were represented using 2D line and bar graphs. In Figure 7.3, the number of selected features obtained

after running the discriminant classifier three times on the same dataset are shown on the x -axis, the misclassification error rates are on the y -axis, and the accuracy of the selected feature subset is shown in the upper right corner. For example, for the first run, six features were selected and they have 63% classification accuracy. Also, changes in the accuracy during each run is represented using a 2D line graph (shown in the lower right-hand corner) where the x -axis is the number of runs and y -axis is the accuracy values. Lastly, the 2D bar graph in the lower left-hand corner depicts frequencies of selected features (each bar represents the frequency of occurrence of a feature).

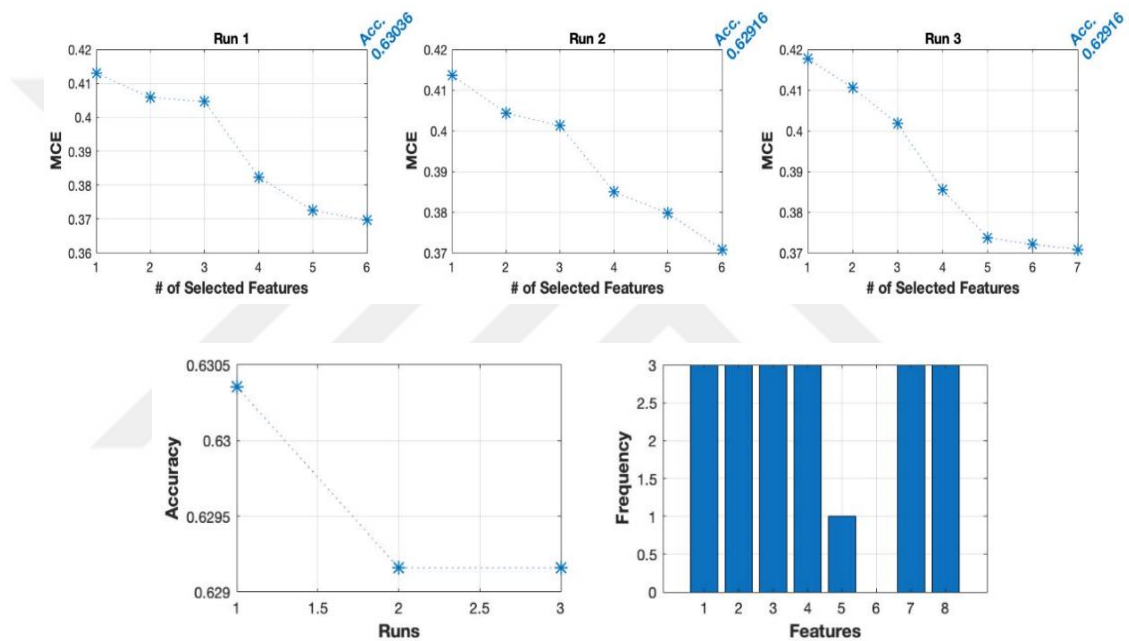


Figure 7.3. Screenshot of Discriminant Classifier Results After Three Runs

Based on the selected features minimum, the average and maximum predictive accuracy of the SFS algorithm on the Breast Cancer dataset after ten runs are tabulated in Table 7.5 and given as an example (see Appendix 5 for all tables). The results in these tables were used to compare the accuracy rates of the classifiers with the default and optimized parameters.

Table 7.5. An Example of Predictive Accuracy Table of the SFS Algorithm Used with LDA, K-NN, and NB Classifiers on Breast Cancer Dataset

Classifier	Minimum Accuracy	Average Accuracy	Maximum Accuracy
LDA	0.9619	0.9625	0.9634
Opt. LDA	0.9634	0.9649	0.9663
K-NN	0.9575	0.9619	0.9678
Opt. K-NN	0.9722	0.9747	0.9780
NB	0.9634	0.9649	0.9663
Opt. NB	0.9707	0.9729	0.9751

The stability of the wrapper algorithms was evaluated using set-based (Hamming Distance, Jaccard and Cosine Index, Sorensen–Dice and Overlapping Coefficient and Lustgarten’s measures) and frequency-based (Nogueira’s measure) measures. On the contrary to rank/weight-based stability calculation, all stability values were calculated using a single MATLAB function. Average stability values of the SFS algorithm using LDA, K-NN and NB classifiers and their optimized versions in terms of different measures over ten runs are tabulated in Table 7.6 and given as an example (see Appendix 5 for all tables). As mentioned before, each stability measure used in the empirical study have different characteristics. Therefore, comparing stability values within a row is meaningless, but comparing stability values within each column allows a comparison of classifiers based on a specific stability measure. For example, the NB Classifier has the highest stability values for all measures whereas the K-NN classifier has the lowest stability values.

Table 7.6. An Example of Average Stability Results Table of the SFS Algorithm Used with LDA, K-NN, and NB Classifiers on Breast Cancer Dataset

Classifier	HD	JI	CI	SDC	OC	LM	NM
LDA	0.7951	0.6893	0.8103	0.8047	0.7944 0.8374	0.3856	0.5893
Opt. LDA	0.8741	0.7267	0.8206	0.8190	0.8389 0.8056	0.5244	0.7212
K-NN	0.6568	0.4988	0.6430	0.6362	0.6593 0.6407	0.2437	0.3122
Opt. K-NN	0.7778	0.6799	0.8101	0.8005	0.8552 0.7848	0.4004	0.5500
NB	0.9407	0.8667	0.9285	0.9238	0.9444 0.9222	0.6333	0.8739
Opt. NB	0.9037	0.8167	0.8899	0.8881	0.9000 0.8833	0.5167	0.8039

7.3. Classification Performance Evaluation

Classification performances of the features (before, see Appendix 3, and after feature selection) were evaluated by Linear Discriminant Analysis, K-Nearest Neighbor and Naïve Bayes classifiers in terms of error and accuracy rate, sensitivity, specificity, F1 Score and AUC (area under the ROC Curve) performance measures. In the literature, various metrics for classification model evaluation are presented. They assess classification performance in different ways. The main aim of this study is not to analyze different evaluation metrics but to determine any possible relation with stability and classification performance. Therefore, the metrics used during the study are enough to reflect the main characteristics of the classifier. In cases where the same feature set is selected more than once, only one of the results is considered, since the performance metrics will generate the same results.

The confusion matrices, which are given as an example in Figure 7.4, have also been used to visualize the true and false ratios of the selected feature subsets. In each matrix, the rows correspond to the true classes and the columns correspond to the predicted classes. Correct predictions are located in diagonal cells and incorrect predictions located in off-diagonal cells. Lastly, the precision and recall values are shown on the right-hand side and the positive predictive values and false discovery rates are shown on the bottom of each chart.

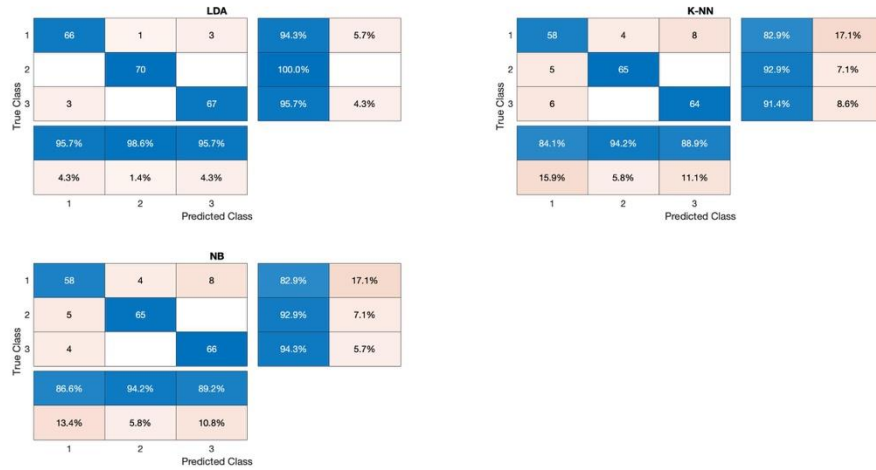


Figure 7.4. Screenshot of the Confusion Matrices

The AUC value of each class was calculated after the ROC Curve (with the “One vs. All” strategy when the number of classes is more than two) plotted for each class. In Figure 7.5, the ROC Curves and the AUC values of two different feature subsets selected from the same dataset by the Discriminant Analysis classifier are shown. On each ROC Curve graph, selected features, true positive ratios (TPR - on the y-axis) and false positive ratios (FPR - on the x-axis) can be seen. The red squares in the graphs refer to the optimal operating point (OOP) of the ROC Curve for the respective class. The optimal operating point can be calculated in a variety of ways, but in this study, the calculation used by MATLAB has been abided. Therefore, the optimal operating point is defined as the coordinates of a point (FPR and TPR values) that the ROC Curve was cut by a line that is drawn with a specific slope starting from the upper left corner ((0,1) or the point where the FPR value is 0 and the TPR value is 1). The slope of the straight line can be calculated using the formula,

$$S = \frac{(TN + FP) [Cost(TP + FN|TN + FP) - Cost(TN + FP|TN + FP)]}{(TP + FN) [Cost(TN + FP|TP + FN) - Cost(TP + FN|TP + FN)]} \quad (16)$$

where TP is equal to true positive, TN is equal to true negative, FP is equal to false positive and FN is equal to false negative rates (The MathWorks, Inc., 2019).

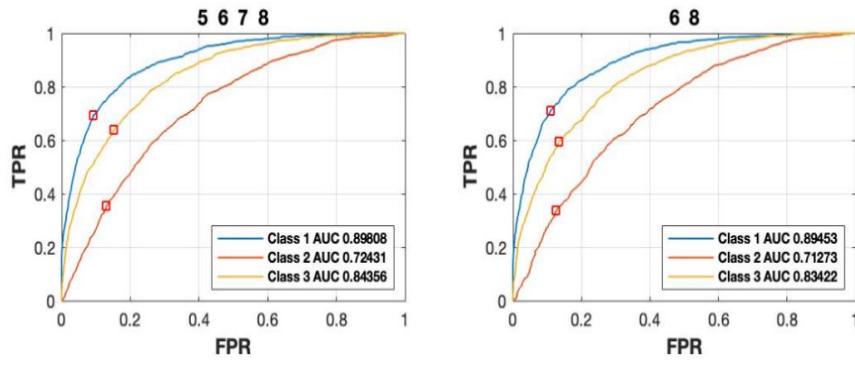


Figure 7.5. ROC Curves and AUC Values Created for Two Different Feature Subsets



CHAPTER 8

DISCUSSION OF EMPIRICAL STUDY RESULTS

8.1. Discussion of Stability Measures

Since each stability measure even of the same type, evaluates the selected feature subsets or the selection algorithm differently, considering a single measure may not be sufficient and objective enough to select the stable algorithm (method). Therefore, in the scope of this work, four rank, one weight, six set (index), and one frequency-based stability measures were used. About the rank and weight-based stability measures which were used for evaluating the stability of the feature filtering algorithms, i.e., feature rankers, these conclusions can be stated:

- Rank and weight-based stability measures cannot be used in feature subsets of different sizes. Therefore, feature subsets to be compared should be equal-sized (should have equal cardinality). However, in each iteration (change in the training dataset), the number of elements (cardinality) in the feature subsets, may vary depending on the results and user-defined threshold value. For example, if the threshold value (results of statistical tests or feature weights) is set as greater than or equal to 0.15, four features can be selected in the first iteration and three in the second iteration from the Abalone dataset using the Bhattacharyya Distance. As can be seen, trying to form equal sized feature subsets may result in adding or removing undesired features to/from the selected feature subsets.
- Canberra Distance is basically the weighted version of Spearman's Footrule. When its formula is examined, it can be seen that stability value and the number of features is directly proportional. In other words, increasing the number of features increases the stability value as well. In order to overcome this problem, a weighted version of Canberra Distance is proposed. Weighted Canberra Distance, only considers top k values (positions) of the ranked features, i.e., top k features. However, this value is intuitively determined by the user and has an impact on the stability value.
- Although both Canberra and Weighted Canberra Distance measures have a lower bound, they do not have an upper bound. This means they obtain a value between zero and infinite. It is possible to scale (normalize) the value between zero and one

in order to satisfy the “to have limits” property but this time normalization result would be affected by outlier values if there is any.

- It is observed that the correlation-based stability measures produce higher stability values than the distance measures (Canberra and Weighted Canberra Distance).
- Although only Pearson’s Correlation Coefficient is parametric, which means it is depended on the distribution of data, it is observed that mostly all three correlation measures produce results close to each other.

About the set (index) and frequency-based stability measures which are used for evaluating the stability of the wrapper algorithms these conclusions can be stated:

- Since the wrapper feature selection algorithms produce feature subsets, stability values are calculated with set-based stability measures. The general formula of these measures is the ratio of intersection over the union of two result sets. Therefore, the problem is seen in Canberra Distance also appears in some of the set-based measures. The increase in the cardinality of the selected feature subset casually causes an increase in the rate of similarity and stability values as well. This causes intersection by chance issue. In order to overcome this problem, some set-based stability measures have a constant for correcting the stability result. One of these measures is the Kuncheva Index. However, Kuncheva Index can be used in feature subsets with equal cardinalities. Since it is difficult to achieve this condition in all cases, Lustgarten et al. (2009) proposed a measure in which feature subsets of different quantities could be compared. Therefore, the measure of Lustgarten was used instead of Kuncheva.
- Regarding the results, it is observed that measures such as Hamming, Cosine, Jaccard, Tanimoto, Sørensen-Dice and Overlapping Coefficient are producing rather similar results. In fact, this is expected because the Jaccard, Tanimoto and Sørensen-Dice measures have very similar formulas and can be generalized using the Tversky Index (Amos,1997) given in the formula below,

$$TI(X, Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|} \quad (17)$$

Setting $\alpha = \beta$ values to 1 will produce the Tanimoto Coefficient and Jaccard Index and $\alpha = \beta$ values to 0.5 will produce the Sorensen-Dice's Coefficient. However,

Lustgarten's and Nogueira's measures are not in a correlation either with other measures or among themselves.

- As mentioned earlier, each stability measure even of the same type assesses the stability of the feature selection algorithm differently. In this study, only one frequency-based measure was used. Other frequency-based measures can be included in the empirical study process for a more objective evaluation.

Lastly, the following can be said about the impact of optimization on stability measures:

- Parameters of all the classifiers (including Decision Tree Ensemble) used in this study are optimized using Bayesian Hyperparameter Optimization. It is observed that the stability and the classification performance of the feature selection algorithms are positively affected.

8.2. Discussion of Stability and Classification Performance

Feature selection has three interdependent objectives. The first objective is not to lose much information during or after the selection process. At this point, finding and removing redundant and irrelevant features should not be considered as loss of information because these features have no significant impact on the classification process. The second objective is to select the smallest possible feature subset(s) or feature subset(s) with low cardinality as possible. This objective helps to reduce the computational cost of data analysis and create simple classification models. The final objective is the classification performance of the resulting subset of features must be, in the worst case, the same as using all features of the original dataset, but in the best case, better than using all features of the original dataset. Therefore, deterioration due to feature selection in the learning process should be considered. In addition to these objectives, the selection algorithm must also be stable. The reliability of a feature selection algorithm generating different results every time it works is questionable. At this point the question of whether there is a relationship between selection stability and classification performance arises. This question can be thought of as a closed-ended, i.e., binary, question and answered with "yes" or "no" responses, but the possibility of being in a relationship only in some cases should also be taken into consideration before reaching a final judgment. Therefore, three different answers can be given to this question.

- If the answer is “yes” then selection algorithm stability and classification performance of the selected feature subsets are dependent concepts and there is a relation between them. This means that both stability value and classification or regression performance of the selected feature subset(s) move in tandem: they both increase or decrease.
- If the answer is “no” then selection algorithm stability and classification performance of the selected feature subsets are two independent concepts and there is no relation between them. This means stability value and classification or regression performance of the selected feature subset(s) move in opposite directions. When one of them increases the other decreases or vice versa.
- Finally, if the answer is “not always” then in some cases selection algorithm stability and classification performance of the selected feature subsets are dependent concepts and there is a relationship between them. In this case, it should be determined which condition(s) justify the relationship between these concepts.

In order to assess any possible relationship between stability and classification performance, first, feature selection methods that have low stability values were discarded using two-stage selection wherein each stage different selection criterion is used. A strict selection policy was used to select as stable algorithms as possible. Here, it should be noted that using different criteria may lead to different conclusions about the relationship between stability and classification performance. Later, results (selected feature subsets) of the feature selection algorithms that succeed in both selections were used to determine a possible relationship.

In the first stage, two different selection criteria (threshold) were used. The first selection criterion is to take the average of the stability measures (column-wise mean) for each dataset and to ignore algorithms that have stability values less than the average. For Canberra and Weighted Canberra measures, because they are not normalized, algorithms that have stability values greater than the average were ignored. In this way, a total of 10 feature filters (rankers), which is approximately 20.5% of all filter methods used, and a total of 20 wrappers (12 SFS and 8 SBS), which is approximately 24% of all wrappers used, were selected.

The second selection criterion is to set a general threshold defined regarding the average stability values of both filter and wrapper algorithms over all datasets. The

average for correlation correlations measures is 0.90, 0.50 for Lustgarten’s and Nogueira’s, 1.45 for Canberra and Weighted Canberra (for these metrics, the closer the value is to zero, the more stable the algorithm is), and 0.80 for the remaining metrics. Note that all averages are rounded up to the nearest tenth. This time 25 feature filters and 8 wrappers satisfied the threshold. In the second stage, the algorithms selected in both cases were selected and the classification performances of their feature subsets were used to make comments on the relationship between the stability and the classification performance. In Table 8.1 the list of selected algorithms is represented.

Table 8.1. List of Selected Filter and Wrapper Algorithms

Dataset	Filter (Ranker) Techniques Selection Criteria		Wrapper Techniques Selection Criteria	
	Dataset Based Average	General Average	Dataset Based Average	General Average
Abalone	ROC DTE	ROC DTE	SFS – Opt. DA SFS – Opt. NB SBS – NB	SFS – Opt. DA SFS – Opt. NB SBS – NB
Australian	DTE	DTE	SFS – DA SFS – Opt. DA SFS – Opt. NB	SFS – DA SFS – Opt. DA
Breast Cancer	ReliefF DTE	T-Test Bhattacharyya ROC ReliefF DTE	SFS – Opt. KNN SFS – NB SBS – DA SBS – KNN	SFS – Opt. KNN SFS – NB SBS – DA SBS – KNN
Sat	T-Test DTE	DTE	SFS – NB SBS – NB SBS – Opt. NB	None
Seeds	DTE	All	SFS – Opt. DA SFS – Opt. NB SBS – DA	None
Vehicle	ROC DTE	DTE	SBS – DA SBS – Opt. DA	SBS – Opt. DA
Wine	DTE	T-Test Bhattacharyya ROC ReliefF DTE	SFS – DA SFS – Opt. KNN	SFS – Opt. KNN

In each table below (Table 8.2), the name of the dataset from which the feature selection is performed appears in the header. The classification algorithms and the performance metrics (classification model evaluation metrics) used to measure the classification performance of feature subsets are located just below the title. The values highlighted with bold prints indicate the performance before feature selection and the

values with regular prints indicate the performance after feature selection. The feature selection technique is written to the first column on the left-hand side, and the indices of the selected features whose performance is tested are shown after this column.

As mentioned before, one of the problems with the filter (ranker) methods is to find an optimal cutoff point to form feature subsets. A method that can be used for determining this point is to consider the amount of increase between feature weights. The point with a decrease more than normal can be considered as a threshold value. However, this strategy is difficult to use if feature weights are close to each other and/or there is more than one decrement point. In this case, e.g., for the ReliefF algorithm on the Breast Cancer dataset, more than one feature subset was created and tested. Algorithms that select the same features were grouped in a single row and performances of the feature subsets selected from the same dataset were not shared when they yield similar results.

Table 8.2. Classification Performances of Selected Feature Subsets from (a) Abalone, (b) Australian, (c) Breast Cancer, (d) Sat, (e) Seeds, (f) Vehicle and (g) Wine Datasets

Abalone																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
	ALL	0.544	0.455	0.528	0.650	0.495	0.665 0.715 0.872	0.497	0.502	0.452	0.668	0.446	0.560 0.569 0.738	0.515	0.484	0.147	0.874	0.215	0.632 0.700 0.868
Filters Wrappers	{6}	0.542	0.457	0.588	0.621	0.524	0.651 0.705 0.866	0.432	0.567	0.426	0.626	0.411	0.526 0.539 0.653	0.536	0.463	0.695	0.565	0.567	0.653 0.705 0.866
Wrappers	{6, 7}	0.536	0.463	0.557	0.614	0.514	0.650 0.705 0.868	0.454	0.545	0.422	0.645	0.415	0.534 0.551 0.685	0.534	0.465	0.352	0.764	0.400	0.657 0.704 0.868
	{5, 6, 8}	0.550	0.449	0.547	0.646	0.507	0.665 0.716 0.875	0.497	0.502	0.445	0.681	0.446	0.563 0.582 0.722	0.539	0.460	0.431	0.727	0.453	0.664 0.705 0.878
	{4, 5, 6, 8}	0.553	0.446	0.548	0.647	0.508	0.665 0.717 0.876	0.493	0.506	0.442	0.675	0.441	0.558 0.575 0.725	0.526	0.473	0.252	0.826	0.325	0.658 0.703 0.876
	{5, 6, 7, 8}	0.550	0.449	0.536	0.652	0.501	0.665 0.716 0.875	0.474	0.525	0.424	0.662	0.422	0.543 0.564 0.711	0.532	0.467	0.309	0.794	0.371	0.658 0.705 0.876
	{3, 4, 5, 6, 8}	0.548	0.451	0.540	0.644	0.501	0.664 0.718 0.874	0.482	0.517	0.430	0.664	0.428	0.547 0.569 0.719	0.540	0.459	0.372	0.757	0.415	0.651 0.707 0.875
	{3, 5, 6, 7, 8}	0.549	0.450	0.534	0.650	0.499	0.664 0.717 0.874	0.474	0.525	0.437	0.659	0.431	0.548 0.557 0.710	0.539	0.460	0.394	0.738	0.426	0.648 0.707 0.874

(a)

Australian																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
None	ALL	0.857	0.142	0.918	0.809	0.852	0.927 0.927	0.660	0.339	0.570	0.7336	0.599	0.651 0.651	0.805	0.194	0.677	0.908	0.756	0.896 0.896
Filters	{8}	0.855	0.144	0.925	0.798	0.850	0.843 0.843	0.855	0.144	0.925	0.798	0.850	0.862 0.862	0.855	0.144	0.925	0.798	0.850	0.842 0.842
Wrappers	{8, 14}	0.856	0.143	0.928	0.799	0.852	0.874 0.874	0.739	0.260	0.771	0.712	0.724	0.742 0.742	0.697	0.302	0.381	0.950	0.528	0.869 0.869

(b)

Breast Cancer																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
None	ALL	0.960	0.039	0.981	0.920	0.969	0.995 0.995	0.950	0.049	0.970	0.912	0.962	0.941 0.941	0.963	0.036	0.954	0.979	0.971	0.992 0.987
Filters	{2, 3, 6}	0.951	0.048	0.984	0.891	0.963	0.991 0.991	0.941	0.058	0.957	0.912	0.955	0.934 0.934	0.959	0.040	0.963	0.949	0.968	0.991 0.991
	{3, 6}	0.942	0.057	0.981	0.870	0.957	0.988 0.988	0.934	0.065	0.954	0.895	0.949	0.925 0.925	0.947	0.052	0.957	0.928	0.959	0.987 0.987
Wrappers	{1, 2, 3, 6, 7, 8, 9}	0.961	0.038	0.984	0.920	0.971	0.994 0.994	0.947	0.052	0.968	0.907	0.959	0.938 0.938	0.966	0.033	0.959	0.979	0.973	0.992 0.988
	{1, 2, 3, 6, 7, 9}	0.960	0.039	0.984	0.916	0.970	0.994 0.994	0.956	0.043	0.970	0.928	0.966	0.949 0.949	0.960	0.039	0.959	0.962	0.969	0.991 0.988
	{1, 2, 3, 6, 7, 8}	0.960	0.039	0.984	0.916	0.970	0.994 0.994	0.953	0.046	0.970	0.920	0.964	0.945 0.945	0.966	0.033	0.959	0.979	0.973	0.993 0.990
	{1, 2, 6, 8}	0.961	0.038	0.984	0.920	0.971	0.994 0.994	0.953	0.046	0.970	0.920	0.964	0.945 0.945	0.963	0.036	0.957	0.974	0.971	0.993 0.990
	{1, 2, 6, 7}	0.960	0.039	0.984	0.916	0.970	0.994 0.994	0.953	0.046	0.966	0.928	0.964	0.947 0.947	0.957	0.042	0.959	0.953	0.967	0.993 0.993

(c)

Sat																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
None	ALL	0.839	0.160	0.967	0.993	0.972	0.997 0.997 0.987 0.912 0.968 0.968	0.906	0.093	0.981	0.995	0.982	0.988 0.980 0.938 0.842 0.948 0.927	0.795	0.204	0.803	0.971	0.847	0.972 0.995 0.981 0.904 0.927 0.953
Filters	{14, 17, 18, 20, 21, 22}	0.821	0.178	0.953	0.990	0.961	0.995 0.993 0.984 0.909 0.962 0.962	0.847	0.152	0.968	0.988	0.966	0.978 0.972 0.916 0.710 0.908 0.878	0.792	0.207	0.786	0.984	0.856	0.977 0.982 0.983 0.903 0.933 0.946

(d)

Seeds																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
None	ALL	0.966	0.033	0.942	0.978	0.949	0.993 0.999 0.995	0.890	0.109	0.828	0.921	0.834	0.875 0.950 0.928	0.900	0.100	0.828	0.935	0.846	0.967 0.994 0.989
Filters	{1}	0.871	0.128	0.800	0.907	0.805	0.928 0.990 0.974	0.838	0.161	0.800	0.857	0.767	0.828 0.935 0.871	0.866	0.133	0.800	0.900	0.800	0.937 0.990 0.973

(e)

Vehicle																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
None	ALL	0.771	0.228	0.805	0.758	0.781	0.891	0.684	0.315	0.686	0.754	0.705	0.720	0.587	0.412	0.600	0.680	0.619	0.640
							0.863						0.638						0.712
							0.991						0.916						0.873
Filters	{5, 6, 14, 18}	0.631	0.368	0.902	0.427	0.719	0.699 0.674 0.913	0.665	0.334	0.705	0.680	0.690	0.693 0.611 0.901	0.427	0.572	0.221	0.756	0.299	0.516 0.661 0.816
	{5, 6, 8, 12}	0.568	0.431	0.827	0.413	0.675	0.720 0.697 0.735	0.625	0.374	0.669	0.655	0.657	0.662 0.621 0.829	0.488	0.511	0.520	0.570	0.527	0.575 0.702 0.850
Wrappers	{1, 3, 4, 5, 8, 10, 13, 14, 15, 17, 18}	0.763	0.236	0.805	0.756	0.780	0.887 0.854 0.989	0.687	0.312	0.661	0.733	0.680	0.697 0.656 0.938	0.573	0.426	0.574	0.652	0.591	0.657 0.726 0.862
	{1, 3, 7, 8, 10, 12, 13, 14, 15}	0.724	0.275	0.759	0.742	0.747	0.849 0.832 0.965	0.666	0.333	0.654	0.779	0.693	0.716 0.650 0.878	0.580	0.419	0.661	0.606	0.637	0.657 0.686 0.857
	{1, 3, 4, 5, 7, 8, 10, 12, 13, 14}	0.734	0.266	0.754	0.749	0.746	0.865 0.841 0.981	0.664	0.335	0.688	0.728	0.697	0.708 0.642 0.866	0.568	0.431	0.654	0.593	0.627	0.644 0.702 0.865
	{1, 2, 7, 8, 10, 12, 13, 14, 15, 17, 18}	0.728	0.271	0.785	0.719	0.754	0.847 0.844 0.960	0.644	0.355	0.652	0.726	0.671	0.689 0.645 0.846	0.576	0.423	0.642	0.604	0.623	0.655 0.692 0.837
	{1, 7, 8, 10, 12, 13, 15, 16, 17, 18}	0.716	0.283	0.790	0.710	0.754	0.852 0.834 0.953	0.634	0.365	0.674	0.740	0.691	0.707 0.613 0.828	0.574	0.425	0.674	0.597	0.641	0.669 0.687 0.816

(f)

Wine																			
FSA	Selected Features	LDA						K-NN						NB					
		Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC	Acc.	ER	Sens.	Spec.	F1	AUC
None	ALL	0.988	0.011	1	0.991	0.991	1 0.999 1	0.735	0.264	0.881	0.932	0.873	0.907 0.768 0.712	0.977	0.022	0.966	1	0.982	0.998 0.996 0.999
Filters	{6, 10}	0.837	0.162	0.796	0.941	0.831	0.957 0.940 0.992	0.910	0.089	0.949	0.957	0.933	0.953 0.915 0.928	0.865	0.134	0.813	0.949	0.849	0.968 0.952 0.992
Wrappers	{7, 10, 13}	0.949	0.050	0.949	0.966	0.941	0.993 0.989 0.999	0.747	0.252	0.881	0.899	0.845	0.890 0.798 0.717	0.943	0.056	0.915	0.974	0.931	0.996 0.992 0.999
	{1, 7, 10, 11, 13}	0.971	0.028	0.949	0.991	0.965	0.998 0.997 0.999	0.752	0.247	0.881	0.899	0.845	0.890 0.803 0.727	0.977	0.022	0.966	0.983	0.966	0.998 0.998 1
	{1, 7, 8, 10, 13}	0.966	0.033	0.949	0.991	0.965	0.997 0.995 0.999	0.752	0.247	0.881	0.899	0.845	0.890 0.803 0.727	0.955	0.044	0.966	0.983	0.966	0.998 0.993 0.998

(g)

8.2.1. Results of Filter Algorithms

The following conclusions can be drawn from the result tables:

1. In filtering techniques selecting a redundant subset of features is at the discretion of the user. Different cutoff points select different features and may lead to different, in some cases positive and some cases negative, classification performances even if the stability of the selection algorithm does not change. For example, it is possible to select two different subsets of features with two different classification performances using the ReliefF algorithm on the Breast Cancer dataset.
2. Classification performances of the selected features vary according to the induction algorithm used. This is due to the way the classifier works. Therefore, the classification performances differ for the same feature set. For example, the LDA classifier has quite close but better performance than the Naïve Bayes Classifier on the Abalone dataset. A similar situation applies to feature selection algorithms. Selection algorithms can select different features from the same dataset. Therefore, instead of comparing the algorithms with each other, it will be fairer to compare the performance values obtained before and after the selection using the same algorithm.
3. Selection algorithms select only a few features, for example, one or two, especially in cases where the number of features is small, and this may be considered as reasonable for these sets. However, as the number of features in the original dataset increases, the number of features that can be added to the selected features subset is expected to increase. Because it is not reasonable to represent all datasets with one or two attributes. In this study, this situation was observed for Vehicle and Sat datasets. Also, for these sets increase in the number of selected features usually increases classification accuracy.
4. Classification performance is adversely affected, even in case of stable feature selection, from high variability of the features in the datasets. Table A.2.1 displays several attributes with high variabilities in datasets. Consider as an example, the Breast Cancer standard deviation (or variance) values. For this dataset, standard deviations are very high in comparison with the feature averages, which is the most likely cause of lower classification performance values after feature selection.

5. When the selected algorithms were examined, it was observed that the DTE algorithm was mostly chosen. Therefore, it can be said that DTE is the most stable algorithm according to the other algorithms used and the selection strategy applied.

8.2.2. Results of Wrapper Algorithms

The following conclusions can be drawn from the result tables:

1. As observed in filtering techniques, in general, classification accuracies obtained after feature selection may be close to, equal to or greater than the accuracies obtained before feature selection. For example, using only features 6 and 7 causes an increase in the performance of the Naïve Bayes classifier on the Abalone dataset. However, the same feature subset has achieved a fairly close value for the K-NN classifier, although it does not cause an increase in performance. As expected, different classifiers lead to different results. Therefore, if the classifiers are compared to each other, it is always difficult to give solid evidence, i.e., strong evidence, of the relationship between property selection stability and property subset classification accuracy.
2. A notable result is a dramatic increase in the sensitivity value of the selected feature subsets in Abalone and Australian datasets. For example, the sensitivity of the Naive Bayes classifier on the Abalone dataset increased from 0.147 to 0.695 after feature selection. This result indicates that the Naïve Bayes classifier is inversely affected by the data composition and feature selection has increased the ability to detect the correct positive rate. The increase in sensitivity value indicates that the ability to detect a positive result also increases after a stable feature selection. However, this tendency is not clearly present in the specificity values which reflects the ability to detect the ratio of correct negative cases. In fact, these results are consistent with the definitions of sensitivity and specificity because theoretically, sensitivity and specificity are inversely proportional, meaning that as sensitivity increases, specificity decreases and vice versa.
3. In order to assess the relationship between stability and classification performance, algorithms with high stability were selected first. After applying the first selection criterion, it was observed that the number of selected SFS and SBS algorithms were very close. However, SBS algorithms failed to meet (or pass over) the second selection criterion. If the feature subsets selected by the SBS algorithm are

examined, it can be seen that the SBS algorithm generally generates feature subsets with high cardinality and shares fewer common features with the SFS algorithm's results. This is the result of the algorithm's unstable behavior. Therefore, it can be concluded that for this selection strategy SBS algorithms are more unstable than SFS algorithms.

4. As can be seen in Table 8.1, the number of wrapper algorithms using default parameters is less than the number of optimized ones. This means that for this selection strategy optimized algorithms are more stable than the unoptimized ones. However, because none of the unoptimized wrapper algorithms were selected, it is not possible to compare the classification performances of the two algorithms.
5. As expected, wrapper algorithms generally had higher classification performances than the filters in the datasets where both filtering and winding algorithms are selected. For example, in the abalone dataset, wrapper algorithms are more successful than the filter algorithm.

8.3. CONCLUSIONS AND FUTURE WORK

In this thesis, the relationship between the stability of feature selection and the classification, i.e., prediction performance, is compared by using filter and wrapper supervised feature selection methods. Benchmark datasets including Abalone, Australian, Breast Cancer, Seeds, Vehicle and Wine have been used for the empirical study. The study is based on feature selection methods with higher stability values. The observed results indicate that under the constraints of the empirical study, there is a general tendency of maintaining or increasing positive classifier metric values after stable feature selection. However, some contrary cases have also been detected and reported in the thesis. The effects of data composition and higher variabilities of individual features have been noted to be the cause of instability in feature selection and reduced accuracy values. In other words, feature selection stability profoundly depends on the quality of the training set. Sets including few, skewed, noisy and outlier samples and having high kurtosis cause degradation in both stability and performance. The results of filter and wrapper feature selection algorithms are also similar concerning stability and classification accuracy. Finally, it may be concluded that the stability of feature selection is an important positive factor for classification accuracy and it should be checked before implementing a classification algorithm.

For the empirical studies, labeled datasets with known statistical properties have been used. However, since the labeling is very costly and it is not always possible to classify data with clear lines, unlabeled datasets are encountered quite often. For these reasons, experiments can be carried out to determine the relationship between the stability and the clustering performance of unsupervised selection methods.

Feature selection is a compute-intensive task, which means it requires lots of computation and therefore computational power, especially when dealing with high dimensional datasets. For this reason, it is not possible to perform empirical studies on most of today's high dimensional datasets using an ordinary computer. Grid computing (or cloud computing), which is a processor architecture that combines various computer resources to reach a common objective or supercomputers can be used to perform feature selection on high dimensional datasets. Therefore, experiments performed in this thesis can be repeated using a grid computing (or cloud) environment.

Selection algorithm stability is defined as the reaction, i.e., robustness, shown by the algorithm against the perturbations done in the training set. These changes can be made on the training set in various ways. Therefore, as a new study, experiments can be performed to determine the optimal amount of change (distortion) in the training dataset without affecting the stability of the selection algorithm and the effect of different validation, cross-validation and sampling techniques on stability.

In this thesis, only one of the frequency-based stability measurements was used. A more extensive experiment can be performed, including the measures that are not used in this study, especially the frequency-based ones, by preserving the current test environment or by making various changes. Selection algorithm instability can be either data-driven, originate from data characteristics, or algorithm-driven, originate from an incorrectly selected algorithm or set parameter or hyperparameter. In this thesis, the classifiers used by the wrapper algorithm are optimized using the Bayes Hyperparameter Optimization method. However, there are different optimization techniques besides the Bayes method. Therefore, the stability of the wrapper algorithms or other selection algorithms that use a classifier to check the selected features classification accuracy can be observed by applying different optimization techniques.

Lastly, irrelevant (non-correlated with the class tag or has no effect on the classification) and redundant (containing information contained in one or more features or duplicated as content) features, as mentioned in the research study conducted by Yu and Liu (2004), are affecting both the performance and the stability of the algorithm directly. In addition to this, the trade-off between bias (underfitting) and variance (overfitting) also causes the same problem. Munson and Caruana (2009) argued in their study that variance reducing methods such as Bagging is a solution to this problem. Therefore, the effect of relevance vs. redundancy and bias vs. variance trade-off on feature selection stability and selected feature subsets classification performance can be determined experimentally.



REFERENCES

- Abe, S., 2005, Support Vector Machines for Pattern Classification, *Advances in Pattern Recognition*, 189–195p.
- Alelyani, S., 2013, On feature selection stability: a data perspective, PhD Thesis, Arizona State University, 124p.
- Altman, N.S., 1992, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician*, 46(3):175–185p.
- Awada, W., Khoshgoftaar, T.M., Dittman, D., Wald, R. and Napolitano, A., 2012, A review of the stability of feature selection techniques for bioinformatics data, 2012 IEEE 13th International Conference on Information Reuse and Integration (IRI), 356–363p.
- Biesiada, J. and Duch, W., 2007, Feature selection for high-dimensional data—a Pearson redundancy based filter, *Computer Recognition Systems 2: Advances in Soft Computing*, 45:242–249p.
- Chandrashekar, G. and Sahin, F., 2014, A survey on feature selection methods, *Computers and Electrical Engineering*, 40(1):16–28p.
- Chelvan, P.M. and Perumal, K., 2017, A comparative analysis of feature selection stability measures, *International Conference on Trends in Electronics and Informatics (ICEI)*, 124–128p.
- Chin, A.J., Mirzal, A., Haron, H. and Hamed, H.N.A., 2015, Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(5):971–989p.
- Cortes, C. and Vapnik, V.N., 1995, Support-vector networks, *Machine Learning*, 20(3):273–297p.
- Dash, M. and Liu, H., 1997, Feature selection for classification, *Intelligent Data Analysis*, 1(1-4):131–156p.
- Deep, K., Singh, K.P., Kansal, M.L. and Mohan, C., 2009, A real coded genetic algorithm for solving integer and mixed integer optimization problems, *Applied Mathematics and Computation*, 212(2):505–518p.
- Dietterich, T.G. and Bakiri, G., 1995, Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research*, 2(1):263–286p.
- Dittman, D.J., Khoshgoftaar, T.M., Wald, R. and Napolitano, A., 2012, Similarity analysis of feature ranking techniques on imbalanced DNA microarray datasets,

- 2012 IEEE International Conference on Bioinformatics and Biomedicine, 398–402p.
- Doquire, G. and Verleysen, M., 2011, Feature selection for multi-label classification problems, International Work-Conference on Artificial Neural Networks 2011: Advances in Computational Intelligence, 9–16p.
- Drotar, P. and Smekal, Z., 2014, Stability of feature selection algorithms and its influence on prediction accuracy in biomedical datasets, TENCON 2014 - 2014 IEEE Region 10 Conference, 1–5p.
- Dunne, K., Cunningham, P. and Azuaje, F., 2002, Solutions to instability problems with sequential wrapper-based approaches to feature selection, Journal of Machine Learning Research, 22p.
- Dy, J.G. and Brodley, C.E., 2004, Feature selection for unsupervised learning, Journal of Machine Learning Research 5, 845–889p.
- Fisher, R.A., 1936, The use of multiple measurements in taxonomic problems, Annals of Eugenics, 7(2):179–188p.
- Goh, W.W. and Wong, L., 2016, Evaluating feature selection stability in next-generation proteomics, Journal of Bioinformatics and Computational Biology, 14(5):23p.
- Guorong, X., Peiqi, C. and Minhui, W., 1996, Bhattacharyya distance feature selection, Proceedings of 13th International Conference on Pattern Recognition, 2:195–199p.
- Guyon, I. and Elisseeff, A., 2003, An introduction to variable and feature selection, Journal of Machine Learning Research 3, 1157–1182p.
- Guzman-Martinez, R., and Alaiz-Rodriguez, R., 2011, Feature selection stability assessment based on the Jensen-Shannon divergence, In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), 597–612p.
- Han, D. and Kim, J., 2018, Unified simultaneous clustering and feature selection for unlabeled and labeled data, IEEE Transactions on Neural Networks and Learning Systems, 29(12):6083–6098p.
- Han, Y., 2012, Stable feature selection: theory and algorithms, PhD Thesis, Binghamton University, 87p.
- Haury, A.C., Gestraud, P. and Vert, J.P., 2011, The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures, PLoS ONE, 6(12):12p.
- He, X., Cai, D. and Niyogi, P., 2005, Laplacian score for feature selection, NIPS'05

- Proceedings of the 18th International Conference on Neural Information Processing Systems, 507–514p.
- Hossin, M. and Sulaiman, M., 2015, A review on evaluation metrics for data classification evaluations, *International Journal of Data Mining & Knowledge Management Process*, 5(2):1—11p.
- Hsu, H.H., Hsieh, C.W. and Lu, M.D., Hybrid feature selection by combining filters and wrappers, *Expert Systems with Applications*, 38(7):8144–8150p.
- Huang, S.H., 2015, Supervised feature selection: a tutorial, *Artificial Intelligence Research*, 4(2):22–37p.
- Jiliang, T., Salem, A. and Huan, L., 2014, Feature selection for classification: a review, *Data Classification: Algorithms and Applications*, 29p.
- Jurman, G., Merler, S., Barla, A., Paoli, S., Galea, A. and Furlanello, C., 2008, Algebraic stability indicators for ranked lists in molecular profiling, *Bioinformatics*, 24(2):258–264p.
- Kalouisis, A., Prados, J. and Hilario, M., 2007, Stability of feature selection algorithms: a study on high-dimensional spaces, *Knowledge and Information Systems*, 12(1):95–116p.
- Kamkar, I., 2016, Building stable predictive models for healthcare applications a data-driven approach, PhD Thesis, Deakin University, 212p.
- Khalid, S., Khalil, T. and Nasreen, S., 2014, A survey of feature selection and feature extraction techniques in machine learning, 2014 Science and Information Conference, London, 372–378p.
- Khandar, P.V. and Dani, S., 2010, Knowledge discovery and sampling techniques with data mining for identifying trends in data sets, *International Journal on Computer Science and Engineering*, 7-11p.
- Khoshgoftaar, T.M., Fazelpour, A., Wang, H. and Wald, R., 2013, A survey of stability analysis of feature subset selection techniques, 2013 IEEE 14th International Conference on Information Reuse and Integration (IRI), 424–431p.
- Kira, K. and Rendell, L.A., 1992, The feature selection problem: traditional methods and a new algorithm, *AAAI'92 Proceedings of the 9th National Conference on Artificial Intelligence*, 129–134p.
- Kohavi, R. and John, H.G., 1997, Wrappers for feature subset selection, *Artificial Intelligence*, 97(1–2):273–324p.
- Kononenko, I., 1997, Overcoming the myopia of inductive learning algorithms with RELIEFF, *Applied Intelligence*, 7(1):39–55p.
- Kumar, V. and Minz, S., 2014, Feature selection: a literature review, *Smart Computing*

- Review, 4(3):211–229p.
- Kuncheva, L.I., 2007, A stability index for feature selection, AIAP'07 Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi Conference: Artificial Intelligence and Applications, 390–395p.
- Landgrebe, T.C.W. and Duin, R.P.W., 2007, Approximating the multiclass ROC by pairwise analysis, Pattern Recognition Letters, 28(13):1747–1758p.
- Lausser, L., Müssel, C., Maucher, M. and Kestler, H.A., 2013, Measuring and visualizing the stability of biomarker selection techniques, Computational Statistics, 28(1):51–65p.
- Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., Schaetzen, V., Duque, R., Bersini, H. and Nowe, A., 2012, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 9(4):1106–1119p.
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J. and Liu, H., 2017, Feature selection: a data perspective, ACM Computing Surveys (CSUR) Surveys, 50(6): 1–94p.
- Li, L., Liu, H., Ma, Z., Mo, Y., Duan, Z., Zhou, J. and Zhao, J., 2014, Multi-label feature selection via information gain, International Conference on Advanced Data Mining and Applications 2014: Advanced Data Mining and Applications, 345–355p.
- Little, R.J.A., 1993, Statistical analysis of masked data, Journal of Official Statistics, 9:407–426p.
- Liu, H. and Setiono, R., 1996, A probabilistic approach to feature selection—a filter solution, ICML'96 Proceedings of the Thirteenth International Conference on Machine Learning, 319–327p.
- Lustgarten, J. L., Gopalakrishnan, V. and Visweswaran, S., 2009, Measuring stability of feature selection in biomedical datasets, AMIA: Annual Symposium Proceedings, 406–410p.
- Mohana, C.P. and Perumal, K., 2016, A survey on feature selection stability measures, International Journal of Computer and Information Technology, 05(01):98–103p.
- Munson, M.A. and Caruana, R., 2009, On feature selection, bias-variance, and bagging, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 144–159p.
- Nogueira, S. and Brown, G., 2015, Measuring the stability of feature selection with applications to ensemble methods, International Workshop on Multiple Classifier Systems, 135–146p.

- Nogueira, S. and Brown, G., 2016, Measuring the stability of feature selection, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 442–457p.
- Nogueira, S., 2018, Quantifying the stability of feature selection, PhD Thesis, University of Manchester, 126p.
- Nogueira, S., How to Measure the Stability of Feature Selection, Retrieved January 10, 2019 from <http://www.cs.man.ac.uk/~gbrown/stability/>
- Peng, H., Long, F. and Ding, C., 2005, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1226–1238p.
- Qian, M. and Zhai, C., 2013, Robust unsupervised feature selection, IJCAI 2013 Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence, 1621–1627p.
- Robnik—Sikonja, M. and Kononenko, I., 1997, An adaptation of RELIEF for attribute estimation in regression, ICML '97 Proceedings of the 14th International Conference on Machine Learning, 296–304p.
- Rubin, D.B., 1993, Discussion: statistical disclosure limitation, Journal of Official Statistics, 9(2):461–468p.
- Saeyns, Y., Abeel, T. and Peer, Y.V., 2008, Robust feature selection using ensemble feature selection techniques, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 313–325p.
- Saeyns, Y., 2004, Feature Selection for Classification of Nucleic Acid Sequences, PhD. Thesis, Ghent University, Faculty of Sciences, 171p.
- Salem, A., Jiliang, T. and Huan, L., 2013, Feature selection for clustering: a review, Data Clustering: Algorithms and Applications, 33p.
- Sebban, M. and Nock, R., 2002, A hybrid filter wrapper approach of feature selection using information theory, Pattern Recognition, 35(4):835–846p.
- Seijo-Pardo, B., Bolón-Canedo, V., Porto-Díaz, I. and Alonso-Betanzos, A., 2015, Ensemble feature selection for rankings of features, International Work-Conference on Artificial Neural Networks 2015: Advances in Computational Intelligence, 29–42p.
- Shabbir, A., Javed, K., Ansari, Y. and Babri, H.A., 2014, Stability of feature ranking algorithms on binary data, Pakistan Journal of Engineering and Applied Sciences, 15:76–86p.
- Sheikhpour, R., Sarram, M.A., Gharaghani, S. and Chahooki, M.A.Z., 2017, A survey on semi-supervised feature selection methods, Pattern Recognition, (64):141–

158p.

- Somol, P. and Novovicova, J., 2010, Evaluating the stability of feature selectors that optimize feature subset cardinality, Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition, 956–966p.
- The MathWorks, Inc., Statistics and Machine Learning Toolbox: User's Guide (Release 2019a), Retrieved March 6, 2019 from https://www.mathworks.com/help/pdf_doc/stats/index.html
- Theodoridis, S. and Koutroumbas, K., 2003, Pattern Recognition Second Edition, Elsevier Academic Press, 163–207p.
- Tuv, E., Borisov, A., Runger, G. and Torkkola, K., 2009, Feature selection with ensembles, artificial variables, and redundancy elimination, Journal of Machine Learning Research, 10:1341–1366p.
- Wang, A., An, N., Chen, G., Li, L. and Alterovitz, G., 2015, Accelerating wrapper-based feature selection with K-nearest-neighbor, Knowledge-Based Systems, 83:81–91p.
- Wang, H., Khoshgoftaar, T.M. and Liang, A.Q., 2011, Stability and classification performance of feature selection techniques, 10th International Conference on Machine Learning and Applications and Workshops, 151–156p.
- Wang, H., Wang, G., Zeng, X. and Peng, S., 2017, Online streaming feature selection based on conditional information entropy, IEEE International Conference on Big Knowledge, 230–235p.
- Wang, S., Tang J. and Liu H., 2016, Feature selection, Encyclopedia of Machine Learning and Data Mining, 1–9p.
- Wei, C.H., 2008, Sampling in data mining, Encyclopedia of Statistics in Quality and Reliability, 5p.
- Wu, X., Yu, K., Ding, W., Wang, H. and Zhu, X., 2013, Online feature selection with streaming features, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(5):1178–1192p.
- Yang, P., Zhou, B.B., Yang, Jean J.Y.H. and Zomaya, A.Y., 2013, Stability of feature selection algorithms and ensemble feature selection methods in bioinformatics, Biological Knowledge Discovery Handbook, 333–352p.
- Yu, L. and Liu, H., 2003, Feature selection for high-dimensional data: a fast correlation-based filter solution, ICML'03 Proceedings of the Twentieth International Conference on Machine Learning, 856–863p.
- Yu, L. and Liu, H., 2004, Efficient feature selection via analysis of relevance and

- redundancy, *The Journal of Machine Learning Research*, 5:1205–1224p.
- Yu, L., Ding, C. and Loscalzo, S., 2008, Stable feature selection via dense feature groups, In *KDD'08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 803–811p.
- Zhang, L., Hu, Q., Duan, J. and Wang, X., 2014, Multi-label feature selection with fuzzy rough sets, *International Conference on Rough Sets and Knowledge Technology 2014: Rough Sets and Knowledge Technology*, 121–128p.
- Zhao, Z. and Liu, H., 2007, Spectral feature selection for supervised and unsupervised learning, In *ICML'07: Proceedings of the 24th International Conference on Machine Learning*, 1151–1157p.
- Zheng, H.T. and Zhang, H., 2016, Online streaming feature selection using sampling technique and correlations between features, *Asia-Pacific Web Conference 2016: Web Technologies and Applications*, 43–55p.

APPENDIX 1 — K AND HYPERPARAMETERS VALUES

Hyperparameter optimization algorithm builds a model of the target (objective) function, e.g., for classification error rate or the accuracy, and assumes that the model contains some errors, i.e., noise. Then regarding the model algorithm calculates the best observed feasible and the best estimated feasible points (shown in first and second lines respectively for each dataset in tables A.1.1). The best observed feasible point is the point with the lowest returned value from objective function evaluations and the best estimated feasible point is the point with the lowest estimated mean value according to the latest model of the objective function.

ReliefF algorithm requires the number of nearest neighbors, i.e., k value, defined as a positive integer scalar by the user. The optimal k value can be determined by using search techniques as mentioned in chapter five. In table A.1.2 the feature, i.e., predictor, weights ranged between -1 and 1 and optimal neighbor size of each feature in each dataset is given.

Table A.1.1. Hyperparameter Values of (a) LDA, NB, (b) K-NN and (c) DTE Classifiers

Dataset	Linear Discriminant Analysis (LDA)			Naive Bayes (NB)		
	Delta	Gamma	Discrimination Type	Distribution Names	Kernel Smoothing Window Width	Kernel
Abalone	0.01018173	0.001258772	diagLinear	kernel	0.002513588	normal ³
	0.01018173	0.001258772	diagLinear	kernel	0.002513588	normal
Australian	1.0278E-06	0.995505612	pseudoLinear	kernel	0.497089546	epanechnikov
	0.00060281	0.995111918	linear	kernel	0.497089546	epanechnikov
Breast Cancer	0	0	diagQuadratic	kernel	0.169635846	normal
	0	0	diagQuadratic	kernel	0.355934713	triangle
Sat	0	0	pseudoQuadratic	kernel	0.025133996	normal
	0	0	quadratic	kernel	0.025615217	normal
Seeds	4.5828E-06	0.116646232	diagLinear	kernel	5.089383567	normal
	4.5828E-06	0.116646232	diagLinear	kernel	5.090231692	epanechnikov
Vehicle	0	0	pseudoQuadratic	kernel	0.027485513	normal
	0	0	pseudoQuadratic	kernel	0.027485513	normal
Wine	2.4361E-05	0.019658287	linear	normal	-	normal
	0	0	quadratic	normal	-	normal

(a)

³ Normal means Gaussian Distribution.

K-Nearest Neighbor (K-NN)					
Dataset	Number of Neighbors	Distance Metric	Distance Weighting Function	Minkowski Distance Exponent	Standardize Predictors
Abalone	89	chebychev	squaredinverse	-	false
	89	chebychev	squaredinverse	-	false
Australian	12	hamming	inverse	-	true
	12	hamming	inverse	-	true
Breast Cancer	16	cosine	equal	-	true
	16	cosine	equal	-	true
Sat	1	euclidean	squaredinverse	-	false
	1	euclidean	squaredinverse	-	false
Seeds	2	seuclidean	inverse	-	false
	20	mahalanobis	inverse	-	false
Vehicle	8	mahalanobis	inverse	-	false
	8	mahalanobis	inverse	-	false
Wine	1	minkowski	inverse	0.815434059	true
	1	minkowski	inverse	0.815434059	true

(b)

Decision Tree Ensemble (Ensemble of Learners)							
Dataset	Method	Number of Learning Cycles	Learning Rate for Shrinkage	Minimum Leaf Size	Maximum Number of Splits	Split Criterion	Number of Predictors to Select at Random for Each Split
Abalone	Bag	388	1	57	552	gdi	5
	Bag	388	1	57	552	gdi	5
Australian	AdaBoostM1	500	0.18135	227	473	deviance	All
	AdaBoostM1	500	0.18135	227	473	deviance	All
Breast Cancer	RUSBoost	310	0.90705	1	679	gdi	All
	Bag	376	1	3	103	deviance	7
Sat	Bag	477	1	3	5230	gdi	32
	Bag	477	1	3	5230	gdi	32
Seeds	AdaBoostM2	102	0.01461	5	11	gdi	All
	AdaBoostM2	128	0.13229	4	2	gdi	All
Vehicle	Bag	163	1	5	23	deviance	18
	Bag	476	1	3	72	deviance	18
Wine	Bag	52	1	5	4	gdi	1
	Bag	84	1	25	163	gdi	1

(c)

Table A.1.2. Weights and K Values of Each Feature in the Datasets

Number of Features	Abalone		Australian		Breast Cancer		Sat		Seeds		Vehicle		Wine	
	Weights	K	Weights	K	Weights	K	Weights	K	Weights	K	Weights	K	Weights	K
1	0.1056	1439	-0.0006	382	0.2227	443	0.1427	1517	0.2885	69	0.0171	410	0.1447	69
2	0.0592	1447	0.0074	382	0.3756	443	0.1391	1525	0.2814	69	0.0068	410	0.0740	70
3	0.0639	1447	0.0135	382	0.3548	443	0.0977	1531	0.0847	69	0.0224	410	0.0157	70
4	0.0135	1401	0.0081	382	0.2576	442	0.0915	1533	0.2199	69	0.0081	410	0.0475	70
5	0.0639	1447	0.0495	382	0.1908	442	0.1489	1521	0.2574	69	0.0036	410	0.0286	70
6	0.0415	1447	0.0181	382	0.4660	443	0.1450	1525	0.0629	69	0.0092	410	0.1213	70
7	0.0509	1447	0.0197	381	0.2326	443	0.0929	1530	0.2171	69	0.0178	410	0.1862	70
8	0.0592	1447	0.5135	382	0.2953	442	0.0895	1517			0.0177	410	0.0590	70
9			0.2118	382	0.0672	443	0.1458	1524			0.0187	410	0.0489	70
10			0.0216	380			0.1474	1503			0.0044	410	0.1314	70
11			-0.0002	382			0.0903	1526			0.0142	410	0.1046	70
12			-0.0007	382			0.0868	1509			0.0176	410	0.1953	70
13			0.0026	382			0.1521	1524			0.0084	410	0.1758	70
14			0.0035	313			0.1488	1517			0.0139	410		
15							0.0981	1525			0.0080	410		
16							0.0971	1518			0.0073	410		
17							0.1647	1522			0.0098	410		
18							0.1707	1519			0.0346	410		
19							0.1026	1527						
20							0.0984	1524						
21							0.1559	1526						
22							0.1625	1521						
23							0.0969	1525						
24							0.0937	1528						
25							0.1422	1526						
26							0.1433	1513						
27							0.0964	1520						
28							0.0902	1532						
29							0.1487	1524						
30							0.1545	1525						
31							0.0931	1525						
32							0.0901	1533						
33							0.1442	1533						
34							0.1511	1503						
35							0.0889	1530						
36							0.0862	1526						

APPENDIX 2 — SUMMARY (DESCRIPTIVE) STATISTICS RESULTS

- **Mode** is the most common and **median** is the middle number.
- **Skewness** is a measure of the asymmetry of the distribution around its mean. Distribution with a long-right tail is right-skewed (or skewed to the right or positively skewed) and the mean of the distribution is located on the right of the peak. Distribution with a long-left tail is left-skewed (or skewed to the left or negatively skewed) and the mean of the distribution is located on the left of the peak. The normal distribution, that has symmetric tails, has a skewness of zero.
- **Kurtosis** is a measure of the flatness of the distribution peak. Distributions with kurtosis less than 3 are called platykurtic. Platykurtic distributions produce fewer outliers than the normal distribution. The univariate normal distribution has a kurtosis of 3. Distributions with kurtosis greater than 3 are called leptokurtic. Leptokurtic distributions produce more outliers than the normal distribution.
- **Abrupt Changes** is a list of features (columns) with data values that show a sudden (unexpected) change in their values more than the mean of the regarding the feature.
- **Outliers** is a list of features with data values that have more than three standard deviations from the mean of the regarding the feature.

Table A.2.1. Summary Statistics of (a) Abalone, Australian, Breast Cancer, Seeds and Vehicle, (b) Sat and Wine Datasets

	Ft.	Min	Mode	Mean	Median	Max	Variance	Standard Deviation	Skewness	Kurtosis
Abalone	1	0.075	0.55	0.52399	0.545	0.815	0.014422	0.12009	-0.63964	3.0631
	2	0.055	0.45	0.40788	0.425	0.65	0.0098486	0.09924	-0.60898	2.9531
	3	0	0.15	0.13952	0.14	1.13	0.0017495	0.041827	3.1277	78.933
	4	0.002	0.2225	0.82874	0.7995	2.8255	0.24048	0.49039	0.53077	2.9749
	5	0.001	0.175	0.35937	0.336	1.488	0.049268	0.22196	0.71884	3.593
	6	0.0005	0.1715	0.18059	0.171	0.76	0.012015	0.10961	0.59164	3.0825
	7	0.0015	0.275	0.23883	0.234	1.005	0.019377	0.1392	0.6207	3.5299
	8	1	9	9.9337	9	29	10.395	3.2242	1.1137	5.3265
Australian	1	0	1	0.67826	1	1	0.21854	0.46748	-0.7632	1.5825
	2	13.75	31.57	31.568	28.625	80.25	140.5	11.853	1.1534	4.1748
	3	0	1.5	4.7587	2.75	28	24.782	4.9782	1.4856	5.2489
	4	1	2	1.7667	2	3	0.18495	0.43006	-1.1509	2.6555
	5	1	8	7.3725	8	14	13.566	3.6833	-0.06904	2.1484
	6	1	4	4.6928	4	9	3.9693	1.9923	0.46739	2.8145
	7	0	0	2.2234	1	28.5	11.199	3.3465	2.885	14.111
	8	0	1	0.52319	1	1	0.24982	0.49982	-0.092854	1.0086
	9	0	0	0.42754	0	1	0.2451	0.49508	0.29295	1.0858
	10	0	0	2.4	0	67	23.648	4.8629	5.1413	53.453
	11	0	0	0.45797	0	1	0.24859	0.49859	0.16871	1.0285
	12	1	2	1.929	2	3	0.089289	0.29881	-1.9405	9.6616
	13	0	0	184.01	160	2000	29639	172.16	2.7439	22.774
	14	1	1	1018.4	6	1e+05	2.714e+07	5210.1	13.112	216.11
Breast Cancer	1	0	0.6	3.9607	3.63	10	8.0139	2.8309	0.59328	2.4073
	2	0	0.5	2.6515	0.91	9.99	9.528	3.0867	1.2006	3.0463
	3	0	0.17	2.722	0.99	10	9.082	3.0136	1.1415	2.9946
	4	0	0.41	2.3204	0.85	9.98	8.3537	2.8903	1.4761	3.891
	5	0.02	1.25	2.7388	1.79	9.96	4.9872	2.2332	1.6499	5.0191
	6	0	0.29	3.0625	0.86	9.99	13.395	3.6599	0.98079	2.2089
	7	0.01	0.17	2.9417	2.2	9.97	5.9892	2.4473	1.0545	3.1384
	8	0	0.02	2.3707	0.8	10	9.3631	3.0599	1.3948	3.4507
	9	0	0.62	1.1147	0.6	9.92	3.1308	1.7694	3.363	14.546
Seeds	1	10.59	11.23	14.848	14.355	21.18	8.4664	2.9097	0.39703	1.9129
	2	12.41	13.47	14.559	14.32	17.25	1.7055	1.306	0.38381	1.891
	3	0.8081	0.8823	0.871	0.87345	0.9183	0.00055835	0.023629	-0.5341	2.8346
	4	4.899	5.236	5.6285	5.5235	6.675	0.19631	0.44306	0.52172	2.2045
	5	2.63	3.026	3.2586	3.237	4.033	0.14267	0.37771	0.13342	1.8998
	6	0.7651	2.129	3.7002	3.599	8.456	2.2607	1.5036	0.39879	2.9065
	7	4.519	5.001	5.4081	5.223	6.55	0.24155	0.49148	0.55788	2.1507
Vehicle	1	73	89	93.678	93	119	67.807	8.2345	0.38059	2.4608
	2	33	43	44.862	44	59	38.067	6.1699	0.26233	2.0734
	3	40	66	82.089	80	112	248.74	15.772	0.10703	2.0202
	4	104	197	168.94	167	333	1120.4	33.472	0.39001	3.293
	5	47	64	61.694	61	138	62.225	7.8883	3.8148	32.653
	6	2	7	8.5674	8	55	21.171	4.6012	6.7664	61.024
	7	112	150	168.84	157	265	1105.2	33.245	0.6047	2.3807
	8	26	31	40.934	43	61	61.02	7.8116	0.04776	2.1339
	9	17	19	20.583	20	29	6.7192	2.5921	0.76932	2.6022
	10	118	144	148	146	188	210.7	14.516	0.2559	2.2274
	11	130	170	188.63	178.5	320	985.64	31.395	0.65066	3.1105
	12	184	327	439.91	364	1018	31220	176.69	0.83435	2.7783
	13	109	186	174.7	173	268	1059.3	32.546	0.27973	2.5056
	14	59	72	72.462	71.5	135	56.055	7.487	2.0689	14.299
	15	0	1	6.3771	6	22	24.19	4.9184	0.77242	3.0807
	16	0	11	12.599	11	41	79.767	8.9312	0.6881	2.8528
	17	176	188	188.93	188	206	37.994	6.1639	0.2481	2.4023
	18	181	198	195.63	197	211	55.336	7.4388	-0.22594	2.1843

(a)

	Ft.	Min	Mode	Mean	Median	Max	Variance	Standard Deviation	Skewness	Kurtosis
Sat	1	39	67	69.4	68	104	185.12	13.606	0.022394	2.2825
	2	27	79	83.595	87	137	523.6	22.882	-0.67248	2.7791
	3	53	104	99.291	101	140	277.09	16.646	-0.12244	2.1137
	4	33	83	82.593	81	154	357.12	18.898	0.89439	4.2544
	5	39	67	69.15	68	104	183.91	13.561	0.036165	2.2885
	6	27	75	83.244	85	137	523.79	22.886	-0.6558	2.7327
	7	50	104	99.111	101	145	277.69	16.664	-0.1195	2.1026
	8	29	83	82.497	81	157	358.76	18.941	0.89968	4.2555
	9	40	67	68.912	67	104	181.46	13.471	0.044402	2.2993
	10	27	79	82.893	85	130	522.68	22.862	-0.64933	2.7133
	11	50	104	98.853	100	145	276.78	16.637	-0.10303	2.0965
	12	29	83	82.388	81	157	360.28	18.981	0.91849	4.2754
	13	39	67	69.29	68	104	185.03	13.603	0.034387	2.2898
	14	27	79	83.477	85	137	522.12	22.85	-0.66724	2.7846
	15	50	104	99.311	101	145	277.82	16.668	-0.12211	2.1072
	16	29	83	82.645	81	154	358.42	18.932	0.87953	4.2056
	17	40	67	69.046	68	104	183.27	13.538	0.038249	2.2969
	18	27	79	83.171	85	130	524.64	22.905	-0.65936	2.7399
	19	50	104	99.15	100	145	279.48	16.718	-0.11911	2.0894
	20	29	83	82.603	81	157	362.35	19.036	0.88535	4.1978
	21	39	67	68.839	67	104	181.15	13.459	0.052706	2.3158
	22	27	79	82.861	84	130	523.69	22.884	-0.64943	2.7205
	23	50	104	98.95	100	145	279.88	16.73	-0.10688	2.08
	24	29	83	82.469	81	157	363.69	19.071	0.90191	4.2211
	25	39	67	69.162	68	104	184.43	13.581	0.036583	2.3012
	26	27	75	83.373	85	131	519.96	22.803	-0.67457	2.8066
	27	50	104	99.215	100	140	275.98	16.613	-0.11728	2.1199
	28	29	83	82.661	81	154	360.67	18.991	0.88838	4.2097
	29	39	67	68.944	68	104	182.05	13.493	0.046731	2.3077
	30	27	79	83.146	85	130	521.99	22.847	-0.66002	2.7603
	31	50	104	99.112	100	145	279.03	16.704	-0.12093	2.0933
	32	29	83	82.618	81	157	362.66	19.044	0.88271	4.2045
	33	39	67	68.728	67	104	179.6	13.402	0.055951	2.3263
	34	27	75	82.859	84	130	520.61	22.817	-0.65343	2.74
	35	50	104	98.926	100	145	278.74	16.695	-0.10762	2.0818
	36	29	83	82.505	81	157	363.07	19.054	0.89699	4.2338
Wine	1	11.03	12.37	13.001	13.05	14.83	0.65906	0.81183	-0.051047	2.1377
	2	0.74	1.73	2.3363	1.865	5.8	1.248	1.1171	1.0309	3.2573
	3	1.36	2.28	2.3665	2.36	3.23	0.075265	0.27434	-0.17521	4.0786
	4	10.6	20	19.495	19.5	30	11.153	3.3396	0.21125	3.4408
	5	70	88	99.742	98	162	203.99	14.282	1.0889	5.0128
	6	0.98	2.2	2.2951	2.355	3.88	0.39169	0.62585	0.085907	2.1541
	7	0.34	2.65	2.0293	2.135	5.08	0.99772	0.99886	0.025129	2.1106
	8	0.13	0.26	0.3618	0.34	0.66	0.015489	0.12445	0.44635	2.347
	9	0.41	1.35	1.5909	1.555	3.58	0.32759	0.57236	0.51277	3.5057
	10	1.28	2.6	5.0581	4.69	13	5.3744	2.3183	0.86124	3.3374
	11	0.48	1.04	0.9574	0.965	1.71	0.052245	0.22857	0.020913	2.632
	12	1.27	2.87	2.6117	2.78	4	0.50409	0.70999	-0.30469	1.9103
	13	278	520	746.89	673.5	1680	99167	314.91	0.76134	2.725

(b)

Table A.2.2. List of Features Indices in Each Dataset that has Missing and Outlier Values and Abrupt Changes

Dataset	Missing Values	Outlier Values	Abrupt Changes
Abalone	None	All	1,4,5,8
Australian	None	All Features Except 5	2,3,5,6,7,9,10,11,13,14
Breast Cancer	None	All Features Except 1	All
Sat	None	4,8,12,16,20,24,28,32,36	All
Seeds	None	7	All Features Except 3
Vehicle	None	4,5,6,7,9,11,12,14,16,17	All
Wine	None	2,3,4,5,9,10,11,13	1,2,4,5,6,7,9,10,12,13



APPENDIX 3 — CLASSIFICATION PERFORMANCES

Table A.3.1. Classification Performances of All Datasets before Feature Selection

Dataset	Classifier	Accuracy	Error Rate	Sensitivity	Specificity	F1 Score	Prevalence	AUC Values
Abalone	LDA	0.5448	0.4551	0.5281	0.6508	0.4950	0.3658	0.6651 0.7152 0.8728
	K-NN	0.4974	0.5025	0.4528	0.6689	0.4468		0.5609 0.5692 0.7382
	NB	0.5154	0.4845	0.1472	0.8742	0.2157		0.8684 0.7009 0.8684
Australian	LDA	0.8579	0.1420	0.9185	0.8093	0.8519	0.4449	0.9278 0.9278
	K-NN	0.6608	0.3391	0.5700	0.7336	0.5993		0.6518 0.6518
	NB	0.8057	0.1942	0.6775	0.9086	0.7563		0.8966 0.8966
Breast Cancer	LDA	0.9604	0.0395	0.9819	0.9205	0.9699	0.6500	0.9950 0.9950
	K-NN	0.9502	0.0497	0.9707	0.9121	0.9620		0.9414 0.9414
	NB	0.9633	0.0366	0.9549	0.9790	0.9713		0.9928 0.9875
Sat	LDA	0.8390	0.1609	0.9673	0.9932	0.9727	0.2382	0.9976 0.9973 0.9873 0.9120 0.9688 0.9684
	K-NN	0.9067	0.0932	0.9810	0.9951	0.9826		0.9880 0.9801 0.9382 0.8423 0.9486 0.9273
	NB	0.7954	0.2045	0.8036	0.9710	0.8476		0.9721 0.9954 0.9819 0.9042 0.9278 0.9531
Seeds	LDA	0.9666	0.0333	0.9428	0.9785	0.9496	0.3333	0.9934 0.9998 0.9957
	K-NN	0.8904	0.1095	0.8285	0.9214	0.8345		0.8750 0.9500 0.9285
	NB	0.9000	0.1000	0.8285	0.9357	0.8467		0.9679 0.9945 0.9895
Vehicle	LDA	0.7718	0.2281	0.8053	0.7586	0.7815	0.4858	0.8915 0.8636 0.9916
	K-NN	0.6843	0.3156	0.6861	0.7540	0.7050		0.7200 0.6388 0.9161
	NB	0.5874	0.4125	0.6009	0.6804	0.6198		0.6400 0.7124 0.8733
Wine	LDA	0.9887	0.0112	1	0.9915	0.9916	0.3314	1 0.9997 1
	K-NN	0.7359	0.2640	0.8813	0.9327	0.8739		0.9070 0.7680 0.7120
	NB	0.9775	0.0224	0.9661	1	0.9827		0.9982 0.9965 0.9998

APPENDIX 4 — RESULTS OF FILTER ALGORITHMS

Table A.4.1. Stabilities of Filter Methods

FS Method	Dataset	CD	WCD (Top5)	PCC	SRCC	KRCC	Dataset	CD	WCD (Top5)	PCC	SRCC	KRCC
T-Test	Abalone	1.1989	1.0484	0.8524	1.0000	1.0000	Australian	0.9992	0.3262	0.9756	0.9324	0.8201
Entropy		1.3385	1.2535	0.8875	0.8083	0.6734		1.2758	0.6076	0.9727	0.9206	0.7898
Bhat.		1.4231	1.3022	0.9151	0.7512	0.5996		1.1609	0.5119	0.9531	0.9320	0.8223
ROC		1.0481	0.9460	0.9313	1.0000	1.0000		1.0083	0.4312	0.9566	0.9382	0.8258
Wilcoxon		1.4696	1.2061	0.6884	0.5957	0.4470		1.4014	0.5681	0.9128	0.8842	0.7379
Relieff		1.7559	1.5995	0.7354	1.0000	1.0000		1.0179	0.3086	0.9959	0.9179	0.7987
DTE		0	0	0.9999	1.0000	1.0000		0	0	1.0000	1.0000	1.0000
T-Test		Breast Cancer	0.8648	0.7133	0.9762	0.9881		1.0000	Sat	4.3354	1.0249	0.9140
Entropy	1.7618		1.5568	0.5919	0.7054	0.5540	6.1431	1.5676		0.8724	0.8430	0.6737
Bhat.	1.2046		0.9017	0.9097	0.9195	1.0000	5.4321	1.4345		0.9032	0.8796	0.7201
ROC	0.9081		0.8037	0.9803	0.9691	1.0000	8.3759	2.2023		0.6677	0.6549	0.4773
Wilcoxon	0.9720		0.5771	0.9766	0.8168	0.6692	5.7365	2.6079		0.9270	0.8883	0.7123
Relieff	0.2074		0.1292	0.9832	0.9999	1.0000	5.7672	1.5969		0.9214	0.8617	0.6700
DTE	0.0571		0.0571	0.9995	1.0000	1.0000	0.6403	0		0.9999	0.9939	0.9570
T-Test	Seeds		0.5267	0.5267	0.9942	0.9985	1.0000	Vehicle		2.9904	1.2554	0.7810
Entropy		0.2349	0.2349	0.9970	1.0000	1.0000	2.7297		0.9318	0.8957	0.7798	0.6089
Bhat.		0.7276	0.7276	0.9941	0.9991	1.0000	2.7878		1.1242	0.8548	0.8040	0.6210
ROC		0.3486	0.3486	0.9929	1.0000	1.0000	2.0179		0.4834	0.9214	0.8583	0.7108
Wilcoxon		0.0790	0.0790	0.9977	1.0000	1.0000	2.9199		0.8895	0.8863	0.7120	0.5539
Relieff		0.1037	0.1037	0.9969	1.0000	1.0000	3.2138		1.5520	0.7927	0.7766	0.6022
DTE		0	0	1.0000	1.0000	1.0000	0.2555		0.0571	0.9991	1.0000	1.0000
T-Test		Wine	0.4117	0.2141	0.9914	1.0000	1.0000					
Entropy	0.5164		0.2230	0.9893	1.0000	1.0000						
Bhat.	0.8762		0.5918	0.9931	0.9628	0.8727						
ROC	0.3056		0.2362	0.9857	1.0000	1.0000						
Wilcoxon	0.6014		0.3099	0.9915	0.9663	0.8996						
Relieff	0.9336		0.7384	0.9795	0.9747	0.9102						
DTE	0.2654		0.1658	0.9990	1.0000	1.0000						

Table A.4.2. List of Selected Features from Datasets

FS Method	Ranked Features		
	Abalone	Australian	Breast Cancer
T-Test	4,6,7,2,5,1,3,8	8,9,10,5,7,6,3,14,4,2,12,13,11,1	6,3,2,7,1,8,4,5,9
Entropy	6,4,5,3,7,2,1,8	14,10,8,7,9,5,3,6,12,4,2,13,1,11	2,9,8,6,3,4,5,7,1
Bhattacharyya	6,4,7,5,2,3,1,8	14,8,10,7,9,5,12,3,4,2,6,13,11,1	6,2,3,8,4,7,5,1,9
ROC	6,4,7,2,3,5,1,8	8,10,9,5,7,14,3,6,4,2,13,12,11,1	2,3,7,6,5,1,4,8,9
Wilcoxon	8,6,4,5,7,3,1,2	8,13,1,10,11,9,12,7,5,4,2,3,14,6	9,3,2,7,6,5,8,4,1
Relieff	4,6,2,7,1,5,3,8	8,9,5,10,7,6,3,4,2,14,13,11,12,1	6,2,3,8,4,7,1,5,9
DTE	6,8,4,7,1,5,3,2	8,5,14,13,7,2,9,3,11,1,4,6,10,12	3,6,7,2,1,5,8,4,9
	Seeds	Vehicle	Wine
T-Test	2,1,7,5,4,3,6	14,18,3,1,8,6,11,7,4,12,9,15,13,16,5,17,2,10	7,12,13,1,11,10,6,4,2,9,8,3,5
Entropy	1,2,7,4,5,3,6	6,5,14,18,15,3,16,1,8,10,11,12,4,17,9,13,2	7,12,13,10,11,1,6,2,8,4,9,3,5
Bhattacharyya	2,1,7,5,4,3,6	6,14,5,15,18,3,1,16,8,10,11,12,4,17,9,13,2	7,13,12,11,1,10,6,2,4,9,8,3,5
ROC	2,1,7,4,5,3,6	6,14,18,3,11,1,8,12,4,7,9,13,5,16,15,17,2,10	10,13,7,1,12,11,6,4,9,2,5,8,3
Wilcoxon	6,3,7,5,1,2,4	14,6,8,18,11,13,5,3,12,9,7,4,15,1,16,17,2,10	4,8,10,12,7,11,1,2,13,6,9,5,3
Relieff	1,2,5,7,4,3,6	18,3,9,8,1,7,12,14,11,15,4,17,6,13,2,16,10,5	7,12,13,1,10,6,11,2,8,9,4,5,3
DTE	1,7,6,2,5,4,3	6,12,5,8,14,10,18,1,3,4,2,15,13,7,17,16,11,9	10,6,5,11,4,1,2,3,7,8,9,12,13
	Sat		
T-Test	17,18,21,22,5,13,14,29,33,6,1,34,9,30,2,19,10,25,20,23,15,26,31,24,11,7,16,27,35,32,12,8,3,28,36,4		
Entropy	18,17,22,20,14,16,21,30,24,34,6,13,8,29,32,28,5,26,4,33,2,12,10,25,1,36,9,19,15,23,31,7,27,3,11,35		
Bhattacharyya	18,17,22,14,21,13,30,20,6,29,34,16,5,33,10,2,9,24,26,25,1,32,28,8,12,4,36,19,15,23,31,27,7,11,3,35		
ROC	18,14,17,6,2,20,21,22,16,13,30,5,24,29,8,33,34,1,26,9,10,12,32,28,19,25,15,4,36,23,7,11,31,27,3,35		
Wilcoxon	25,29,21,13,33,17,1,9,5,36,32,24,28,20,16,12,8,4,18,7,19,22,11,23,6,3,14,15,30,10,31,27,2,35,34,26		
Relieff	18,17,22,30,21,14,34,33,6,13,29,5,2,10,1,9,26,25,19,23,20,15,16,24,7,31,35,27,11,3,36,32,28,8,4,12		
DTE	17,20,18,22,21,14,6,16,19,24,26,23,15,2,10,33,34,36,30,4,9,27,25,28,13,11,29,12,8,3,35,5,32,1,7,31		

APPENDIX 5 — RESULTS OF WRAPPER ALGORITHMS

Table A.5.1. Stabilities of SFS and SBS Algorithms on (a) Abalone, Australian, Breast Cancer, (b) Sat, Seeds, Vehicle and (c) Wine Datasets

FS Method	Dataset	Classifier	HD	JI	CI	SDC	OC		LM	NM
SFS	Abalone	DA	0.8667	0.7667	0.8456	0.8413	0.7944	0.9056	0.4815	0.7306
		Opt. DA	0.9083	0.8256	0.9033	0.8984	0.9389	0.8778	0.5580	0.8138
		K-NN	0.6528	0.5806	0.7123	0.7051	0.7637	0.6756	0.2531	0.2836
		Opt. K-NN	0.9056	0.6444	0.7089	0.6963	0.6778	0.7667	0.6469	0.6296
		NB	0.7806	0.5315	0.6507	0.6415	0.6630	0.6593	0.3840	0.5093
		Opt. NB	0.9750	0.9000	0.9414	0.9333	0.9444	0.9556	0.8667	0.8946
SBS	Abalone	DA	0.6389	0.5875	0.7380	0.7249	0.7582	0.7453	0.2648	0.2063
		Opt. DA	0.6750	0.6353	0.7744	0.7601	0.8381	0.7407	0.3401	0.2067
		K-NN	0.8028	0.7968	0.8859	0.8824	0.9144	0.8643	0.5889	0.1685
		Opt. K-NN	0.6444	0.5894	0.7286	0.7275	0.7356	0.7237	0.2099	0.2186
		NB	0.9500	0.9333	0.9600	0.9600	0.9600	0.9600	0.5056	0.8933
		Opt. NB	0.6611	0.4137	0.5994	0.5614	0.7185	0.5722	0.3840	0.2940
SFS	Australian	DA	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8667	1.0000
		Opt. DA	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8667	1.0000
		K-NN	0.9365	0.6148	0.7549	0.7333	0.7333	0.8222	0.7867	0.6864
		Opt. K-NN	0.7873	0.4571	0.6463	0.5862	0.7788	0.6683	0.6768	0.3969
		NB	0.6984	0.4285	0.5838	0.5565	0.5889	0.6420	0.3435	0.3432
		Opt. NB	0.9143	0.7452	0.8398	0.8284	0.8889	0.8148	0.6896	0.7569
SBS	Australian	DA	0.6873	0.5996	0.7463	0.7414	0.7200	0.7826	0.2748	0.3445
		Opt. DA	0.8238	0.8055	0.8915	0.8900	0.8609	0.9252	0.5137	0.4341
		K-NN	0.7000	0.4566	0.6114	0.5892	0.6778	0.5953	0.3288	0.3711
		Opt. K-NN	0.7270	0.6848	0.8141	0.8085	0.8585	0.7813	0.3529	0.3311
		NB	0.8254	0.7411	0.8537	0.8466	0.8862	0.8356	0.4486	0.6435
		Opt. NB	0.7317	0.6393	0.7735	0.7695	0.7794	0.7758	0.3084	0.4443
SFS	Breast Cancer	DA	0.7951	0.6893	0.8103	0.8047	0.7944	0.8374	0.3856	0.5893
		Opt. DA	0.8741	0.7267	0.8206	0.8190	0.8389	0.8056	0.5244	0.7212
		K-NN	0.6568	0.4988	0.6430	0.6362	0.6593	0.6407	0.2437	0.3122
		Opt. K-NN	0.7778	0.6799	0.8101	0.8005	0.8552	0.7848	0.4004	0.5500
		NB	0.9407	0.8667	0.9285	0.9238	0.9444	0.9222	0.6333	0.8739
		Opt. NB	0.9037	0.8167	0.8899	0.8881	0.9000	0.8833	0.5167	0.8039
SBS	Breast Cancer	DA	0.6765	0.6222	0.7503	0.7476	0.7228	0.7831	0.2656	0.2838
		Opt. DA	0.8642	0.8448	0.9114	0.9109	0.9079	0.9159	0.5096	0.6204
		K-NN	0.6864	0.6496	0.7694	0.7689	0.7661	0.7735	0.2537	0.2821
		Opt. K-NN	0.8815	0.8532	0.9205	0.9181	0.9356	0.9102	0.5833	0.6970
		NB	0.8123	0.7938	0.8794	0.8794	0.8794	0.8794	0.4185	0.4571
		Opt. NB	0.7407	0.6804	0.7988	0.7962	0.7794	0.8236	0.3252	0.4418

(a)

FS Method	Dataset	Classifier	HD	JI	CI	SDC	OC		LM	NM
SFS	Sat	DA	0.6204	0.3081	0.4716	0.4613	0.4666	0.4988	0.1604	0.1745
		Opt. DA	0.7722	0.4953	0.6487	0.6452	0.7016	0.6027	0.3697	0.4761
		K-NN	0.5926	0.4453	0.6142	0.6072	0.5953	0.6474	0.1399	0.1815
		Opt. K-NN	0.6117	0.3840	0.5538	0.5449	0.5404	0.5856	0.1690	0.2082
		NB	0.9136	0.6454	0.7675	0.7643	0.7467	0.7945	0.6312	0.7114
		Opt. NB	0.7654	0.4168	0.5718	0.5660	0.5826	0.5731	0.3488	0.4080
SBS	Sat	DA	0.7272	0.7153	0.8336	0.8317	0.8315	0.8397	0.2257	0.0667
		Opt. DA	0.7000	0.6560	0.7906	0.7887	0.8094	0.7759	0.2520	0.2613
		K-NN	0.8377	0.8339	0.9089	0.9086	0.9157	0.9029	0.3122	0.1597
		Opt. K-NN	0.8796	0.8778	0.9336	0.9332	0.9329	0.9349	0.4600	0.2772
		NB	0.7846	0.7627	0.8653	0.8631	0.8956	0.8394	0.3888	0.3337
		Opt. NB	0.7735	0.7494	0.8586	0.8547	0.8956	0.8297	0.4072	0.3132
SFS	Seeds	DA	0.7492	0.6652	0.7901	0.7812	0.7711	0.8274	0.3694	0.4918
		Opt. DA	0.8921	0.8244	0.9025	0.8966	0.9141	0.9030	0.4852	0.7825
		K-NN	0.6984	0.5119	0.6353	0.6249	0.6815	0.6111	0.3352	0.3636
		Opt. K-NN	0.7429	0.6396	0.7510	0.7445	0.7852	0.7300	0.3546	0.4857
		NB	0.8190	0.7433	0.8347	0.8333	0.8389	0.8333	0.3556	0.6333
		Opt. NB	0.9143	0.8822	0.9295	0.9284	0.9378	0.9233	0.4481	0.8234
SBS	Seeds	DA	0.8127	0.7433	0.8487	0.8433	0.8563	0.8522	0.4204	0.6047
		Opt. DA	0.6857	0.6169	0.7556	0.7447	0.7674	0.7663	0.3111	0.3368
		K-NN	0.6476	0.5312	0.6867	0.6740	0.6648	0.7352	0.3019	0.2859
		Opt. K-NN	0.6698	0.5930	0.7418	0.7268	0.7678	0.7475	0.3722	0.2929
		NB	0.8032	0.7144	0.8150	0.8124	0.8370	0.7981	0.3565	0.6034
		Opt. NB	0.8127	0.7607	0.8407	0.8375	0.8667	0.8211	0.3713	0.6141
SFS	Vehicle	DA	0.7123	0.6409	0.7777	0.7706	0.8123	0.7575	0.3449	0.3678
		Opt. DA	0.7432	0.6797	0.8109	0.8028	0.7880	0.8504	0.4317	0.4121
		K-NN	0.6296	0.4189	0.5599	0.5581	0.5582	0.5652	0.1724	0.2409
		Opt. K-NN	0.6568	0.4878	0.6605	0.6381	0.7092	0.6609	0.3105	0.3136
		NB	0.7864	0.4092	0.5364	0.5148	0.4900	0.6342	0.4058	0.3708
		Opt. NB	0.8383	0.6077	0.7476	0.7434	0.7305	0.7734	0.4883	0.6263
SBS	Vehicle	DA	0.8173	0.8006	0.8875	0.8872	0.8822	0.8933	0.4211	0.4037
		Opt. DA	0.8827	0.8696	0.9295	0.9291	0.9254	0.9342	0.5670	0.5887
		K-NN	0.7617	0.7081	0.8320	0.8231	0.7667	0.9158	0.4548	0.4494
		Opt. K-NN	0.7741	0.7563	0.8610	0.8566	0.8728	0.8581	0.4168	0.2788
		NB	0.7901	0.7486	0.8555	0.8517	0.8790	0.8397	0.4253	0.5104
		Opt. NB	0.7321	0.5631	0.7173	0.7077	0.6749	0.7796	0.3389	0.4588

(b)

FS Method	Dataset	Classifier	HD	JI	CI	SDC	OC		LM	NM
SFS	Wine	DA	0.8957	0.8211	0.8948	0.8916	0.9175	0.8787	0.4753	0.7913
		Opt. DA	0.6923	0.5266	0.6515	0.6458	0.6485	0.6663	0.2762	0.3726
		K-NN	0.6872	0.3997	0.5407	0.5261	0.6030	0.5101	0.2917	0.2934
		Opt. K-NN	0.9419	0.8504	0.9154	0.9067	0.9867	0.8622	0.7273	0.8495
		NB	0.8479	0.6865	0.8160	0.8029	0.7893	0.8702	0.5563	0.6705
		Opt. NB	0.8462	0.6815	0.8053	0.7980	0.7693	0.8566	0.5152	0.6697
SBS	Wine	DA	0.7863	0.7446	0.8540	0.8494	0.8650	0.8521	0.4812	0.4753
		Opt. DA	0.6547	0.5939	0.7307	0.7217	0.7577	0.7223	0.2690	0.2449
		K-NN	0.7265	0.5863	0.7328	0.7268	0.7254	0.7525	0.2984	0.4529
		Opt. K-NN	0.6222	0.5361	0.6855	0.6782	0.7225	0.6637	0.2120	0.2077
		NB	0.7897	0.7402	0.8498	0.8429	0.9162	0.7977	0.4150	0.5446
		Opt. NB	0.7624	0.7098	0.8336	0.8241	0.8905	0.7962	0.4291	0.4633

(c)



Table A.5.2. Classification Accuracy of Sequential Forward and Backward Feature Selection Algorithms

	Abalone SFS			Abalone SBS			Australian SFS			Australian SBS		
	Min Acc.	Avg. Acc.	Max Acc.	Min Acc.	Avg. Acc.	Max Acc.	Min Acc.	Avg. Acc.	Max Acc.	Min Acc.	Avg. Acc.	Max Acc.
DA	0.5504	0.5530	0.5549	0.5454	0.5502	0.5547	0.8565	0.8565	0.8565	0.8609	0.8636	0.8681
Opt. DA	0.5514	0.5529	0.5549	0.5446	0.5493	0.5545	0.8565	0.8565	0.8565	0.8754	0.8793	0.8826
K-NN	0.4992	0.5039	0.5104	0.4994	0.5061	0.5123	0.8551	0.8570	0.8594	0.8087	0.8339	0.8580
Opt. K-NN	0.5140	0.5169	0.5214	0.5449	0.5539	0.5616	0.8551	0.8630	0.8783	0.8739	0.8814	0.8884
NB	0.5344	0.5396	0.5432	0.5356	0.5386	0.5430	0.8565	0.8671	0.8739	0.8710	0.8755	0.8783
Opt. NB	0.5396	0.5409	0.5430	0.5310	0.5358	0.5430	0.8580	0.8614	0.8681	0.8652	0.8728	0.8783
	Breast Cancer SFS			Breast Cancer SBS			Sat SFS			Sat SBS		
DA	0.9619	0.9625	0.9634	0.9619	0.9638	0.9649	0.8305	0.8379	0.8409	0.8416	0.8431	0.8443
Opt. DA	0.9634	0.9649	0.9663	0.9678	0.9691	0.9707	0.8720	0.8748	0.8763	0.8645	0.8692	0.8744
K-NN	0.9575	0.9619	0.9678	0.9634	0.9669	0.9707	0.8942	0.9073	0.9148	0.9105	0.9121	0.9134
Opt. K-NN	0.9722	0.9747	0.9780	0.9766	0.9776	0.9795	0.8915	0.9037	0.9166	0.9105	0.9126	0.9153
NB	0.9634	0.9649	0.9663	0.9678	0.9687	0.9693	0.8034	0.8065	0.8078	0.7998	0.8018	0.8034
Opt. NB	0.9707	0.9729	0.9751	0.9722	0.9732	0.9751	0.8238	0.8300	0.8329	0.8219	0.8261	0.8308
	Seeds SFS			Seeds SBS			Vehicle SFS			Vehicle SBS		
DA	0.9619	0.9681	0.9762	0.9667	0.9714	0.9762	0.7329	0.7629	0.7931	0.7825	0.7870	0.7931
Opt. DA	0.9571	0.9629	0.9667	0.9571	0.9614	0.9667	0.7955	0.8251	0.8428	0.8440	0.8462	0.8534
K-NN	0.8952	0.9095	0.9238	0.9000	0.9271	0.9476	0.7187	0.7283	0.7447	0.7128	0.7338	0.7506
Opt. K-NN	0.9619	0.9686	0.9714	0.9619	0.9686	0.9714	0.7506	0.7994	0.8357	0.8132	0.8264	0.8392
NB	0.9381	0.9471	0.9524	0.9286	0.9438	0.9524	0.5816	0.5994	0.6170	0.6158	0.6296	0.6418
Opt. NB	0.9333	0.9490	0.9571	0.9238	0.9476	0.9571	0.7104	0.7186	0.7281	0.6927	0.7154	0.7352
	Wine SFS			Wine SBS								
DA	0.9775	0.9888	0.9944	0.9888	0.9938	1.0000						
Opt. DA	0.9719	0.9927	1.0000	0.9944	0.9994	1.0000						
K-NN	0.7809	0.8882	0.9607	0.9326	0.9449	0.9607						
Opt. K-NN	0.9607	0.9702	0.9831	0.9775	0.9865	0.9944						
NB	0.9719	0.9798	0.9944	0.9831	0.9888	0.9944						
Opt. NB	0.9775	0.9809	0.9888	0.9831	0.9888	0.9944						

Table A.5.3. List of Selected Feature Subsets from (a) Abalone, Australian and Breast Cancer, (b) Seeds and Vehicle, (c) Wine and (d) Sat Datasets

	Abalone		Australian		Breast Cancer	
	SFS	SBS	SFS	SBS	SFS	SBS
DA	4,5,6,8 5,6,8 4,6,7	4,5,6,8 1,2,5,6,8 3,4,5,6,7,8 1,2,3,5,6,7,8 1,3,4,5,6,8 1,2,4,5,7,8 5,6,8	8,14	5,6,7,8,9,10,12,13,14 3,4,5,7,8,9,10,11,12,13,14 4,5,7,8,9,12,13,14 1,2,5,6,7,8,9,12,13,14 3,5,6,7,8,9,13,14 5,8,9,10,13,14 3,4,5,7,8,9,10,13,14 4,5,8,9,10,11,12,14 3,5,8,9,10,11,13,14	1,2,4,6,9 1,2,4,6,7,8 1,2,4,6 1,2,5,6 1,2,6,8	1,2,3,4,5,6,8 1,4,5,6,7,8 1,4,5,6,7,8,9 1,2,6,8,9 1,2,6,7,9 1,5,6,7,8,9 1,3,6,7,8
Opt. DA	5,6,8 4,5,6,8 5,6,7,8	2,4,5,6,7,8 5,6,8 1,2,3,4,5,7,8 5,6,7,8 1,2,4,5,6,8 4,5,6,8 1,2,4,5,6,7,8	8,14	1,2,3,4,5,6,7,8,9,10,12,13,14 2,3,4,5,6,7,8,9,10,13,14 2,3,4,5,6,7,8,9,10,11,13,14 1,3,4,5,6,7,8,9,10,12,13,14 1,2,3,4,5,6,8,9,10,13,14 1,2,3,4,5,6,7,8,9,10,13,14 1,2,3,4,5,6,8,9,10,12,13 3,4,5,6,8,9,10,12,13,14 2,3,4,5,6,7,8,9,10,13	1,2,6 2,3,6 1,2,6,8	1,2,4,6,7,8,9 1,2,3,4,6,7,8 1,3,4,6,7,8,9 1,2,3,4,6,8
K-NN	5,6,8 1,4,5,6,8 1,5,6,8 1,3,4,5,6,7 1,2,4,5,7 1,2,4,5,6,8 1,3,4,6,7	1,3,4,5,6,7 1,2,3,4,5,6,7,8 1,3,4,5,6,7,8 1,2,4,5,6,8 1,2,4,5,6,7 1,2,3,4,5,6,7	4,8 8,12 8	4,8 4,5,7,8,9,11,12 4,6,7,8,9,11 4,5,7,8,9,11 1,3,5,8,10 1,4,6,8,12 4,5,7,8,9,10,11 5,8,9	2,5,6,8 1,2,6 1,2,5,6 1,2,3,5,6,7 1,3,5,6 1,3,5,6,7,9 3,6,7,9	1,3,5,6,8,9 1,3,4,5,6,7,8 2,5,6,7,8,9 1,3,5,6,7,9 1,2,4,5,6,8 1,3,4,6,7,9
Opt. K-NN	3,8 3 2	1,3,4,6,7 1,2,4,5,7 1,3,5,6,7 1,2,4,5,6,8 1,2,3,4,5,7 1,4,5,6,7 1,2,3,4,6	4,8 8 4,8,9 4,5,8,9,11,12,13 4,6,7,8,9,13 4,5,8,9,10,12,13	3,4,5,6,7,8,9,12,13,14 3,4,5,6,8,9,12,13 1,4,5,6,8,9,11,12,13 3,4,5,6,8,9,11,12,13 4,5,6,8,9,10,11,12,13,14 4,5,6,8,9,10,11,13 1,4,5,6,7,8,9,10,11,12,13,14 1,2,3,4,5,6,7,8,9,10,11,13,14 3,4,5,6,8,10,11,12,13	1,2,6,8 1,2,6,7 1,2,4,6,7 1,2,3,6,7,8 1,2,3,6,7,9 1,2,3,6,7,8,9	1,2,3,6,7,8 1,2,3,6,7,8,9 1,2,3,5,6,7,8,9 1,2,3,6,8,9 1,2,3,6,7,9
NB	5,6,8 6,8 4,5,8 4,6,8 4	3,4,5,6,8 3,5,6,7,8	5,7,8,9,13 5,6,8,9,13 2,4,5,6,8,9,12 1,2,3,4,5,6,8,9,12 8,12 3,4,5,6,8,9,12 3,8,12	2,4,5,6,8,9,13 3,4,5,6,8,9,13 1,2,3,4,5,6,8,9,11,12,13 2,4,5,6,8,9 2,3,4,5,6,8,9,13 2,3,4,5,6,8,9,12,13 2,3,4,5,6,8,9,11,12,13 3,4,5,6,8,9,12 2,4,5,6,8,9,11,13	1,2,6 1,2,6,8	1,2,3,4,6,7,8 1,2,3,6,7,8,9 1,2,4,6,7,8,9 1,3,4,5,6,7,8 1,3,4,6,7,8,9
Opt. NB	6 6,7	6 4,6,7 5,6,7,8 5,6,8 4,5,6,7,8 3,4,6 3,5,6 3,4,6,7,8 3,4,5,6	4,8,13 3,8,13 4,8 1,4,5,8,9,13	2,4,5,8,9,10,11,13 2,4,5,8,9,10,13 3,4,5,8,9,10,12,13 4,5,8,9,10,11,12,13 1,2,4,5,6,8,9,10,12,13 2,3,4,5,6,8,9,10,11 1,2,3,4,5,6,8,9,10 2,3,5,6,8,9	2,5,6,8 1,2,6,8 2,6,8 2,3,6,8	1,2,5,6,7,8 2,3,5,6,7,8 1,2,3,4,6,7,8 1,2,6,7,8 1,3,4,5,6,8 2,3,5,6,8

(a)

	Seeds		Vehicle	
	SFS	SBS	SFS	SBS
DA	2,4,5,6,7 1,2,4,7 1,4,6,7 1,6,7 1,2,4,6,7 1,4,5,6,7	2,4,5,6,7 2,4,7 2,4,6,7 4,5,6,7 2,3,4,6,7	1,2,3,6,8,9,11,13,14,17 1,2,3,4,5,6,8,11,13,14,17,18 1,2,3,4,5,6,8,10,11,13,14,17,18 1,3,8,10,11,14,17,18 1,3,8,9,10,11,13,14,15,17,18 1,2,3,4,5,6,8,11,13,14,15,16,17,18 1,2,3,4,5,6,8,11,12,13,14,15,16,17,18 1,2,5,6,10,13,14,15,17,18	1,2,3,4,5,6,7,8,11,12,13,14,16,17,18 1,2,3,4,5,6,8,11,13,14,15,16,17,18 1,2,3,4,5,6,7,8,12,13,14,15,16,17,18 1,2,3,4,5,6,7,9,11,12,13,14,15,17,18 1,2,3,4,5,6,7,8,9,12,13,15,16,17,18 1,2,3,4,5,6,7,8,9,13,14,15,17,18 1,2,3,4,5,6,7,8,12,13,15,16,17,18 1,2,3,4,5,6,8,12,13,14,15,16,17,18
Opt. DA	1,2,6,7 2,6,7 1,2,4,6,7 1,6,7	1,2,3,5,6,7 3,4,5,6,7 2,6,7 4,5,6,7 2,4,5,6,7 2,4,6,7	1,3,4,5,7,8,10,11,12,13,14,15,16,17,18 1,3,4,5,8,10,13,14,15,17,18 1,2,3,4,5,7,8,10,12,13,14,15,16,17,18 1,3,7,8,10,12,13,14,15 1,3,4,5,7,8,10,12,13,14 1,7,8,10,12,13,15,16,17,18 1,2,3,5,7,8,10,11,12,13,14,15,16,17,18 1,2,7,8,10,12,13,14,15,17,18	1,2,3,4,5,7,8,9,11,12,13,14,15,16,17,18 1,2,3,4,5,7,8,12,13,14,15,16,17,18 1,2,3,5,7,8,9,11,12,13,14,15,16,17,18 1,2,3,5,7,8,10,11,12,13,14,15,16,17,18 1,2,3,4,5,7,8,10,12,13,14,15,16,17,18 1,2,3,4,5,7,8,11,12,13,15,16,17,18 1,2,3,4,5,7,8,11,12,13,14,15,16,17,18
K-NN	1,6 1,2,6,7 1,5,7 1,4,5,7 1,4,7	3,4,7 2,5,6,7 1,2,4,5,6,7 4,5,7 2,5,7 2,4,7 2,4,6,7	1,2,3,7,8,9,10 3,6,7,8,11,12,14,18 1,3,5,6,11,12,17,18 1,3,5,6,7,8,10,15 1,2,3,5,7,10 3,5,6,7,8,11,12,14,17 1,2,3,6,7,8,9,10 3,6,7,11,12,14,18 6,7,8,11,14,17,18	1,2,3,4,5,6,7,8,10,11,13,14,15,16,17,18 1,2,3,4,5,7,8,10,11,13,14,15,16,17,18 1,2,3,6,8,10,11,13,14,15,17,18 1,2,5,6,7,10,11,13,14,15,17,18 2,3,5,10,11,13,17,18 1,2,3,5,6,10,11,13,15,17,18 1,3,5,6,8,9,10,11,13,15,17,18 1,2,5,6,10,11,13,15,17,18
Opt. K-NN	2,4,5,7 2,5,7 1,4,7 1,3,4,7 1,2,4,7 1,2,3,4,7	2,3,4,6,7 2,3,6,7 3,4,7 2,3,4,7 1,2,3,4,5,6,7 1,2,3,4,6,7 1,4,7 1,2,3,4,7	3,4,6,8 1,3,5,6,8 1,2,3,4,5,6,7,8,9,10,12,13,17,18 1,2,3,4,5,6,7,8,10,12,13,17,18 1,2,3,7,8,10,12,14,17,18 1,3,6,7,8,13,14,17,18 1,2,3,4,5,6,7,8,12,13,17,18 2,3,4,5,6,8,11,13 1,3,5,6,8,10,17,18 1,3,6,8,14,17,18	1,2,4,5,6,7,8,9,10,11,12,13,15,16,17,18 1,2,3,4,5,6,7,8,9,10,12,14,16,17,18 1,2,4,5,6,7,10,12,13,17,18 1,2,3,4,5,6,7,8,9,11,12,13,15,17,18 1,2,3,4,5,7,8,9,10,11,12,13,14,15,17,18 1,2,4,5,7,8,10,12,13,16,17,18 1,2,3,4,5,6,7,8,9,10,11,12,13,15,17,18
NB	2,3,6,7 1,3,6,7 1,6,7	2,3,6,7 2,6,7 1,6,7 1,3,6,7 3,4,6,7	13,14,18 4,5,8,10,11,12,13,14,15,16,18 8,10,13,14,18 5,8,18 1,14	2,3,4,5,7,8,9,10,11,12,14,15,16,18 3,4,5,6,8,10,11,12,14,15,16 1,2,3,5,6,8,9,10,11,12,13,14,15,16 3,5,6,7,8,10,11,12,14,15,16 3,4,5,6,7,8,10,11,12,15,16,18 1,3,4,5,6,7,8,9,10,11,14,15,16 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
Opt. NB	2,3,6,7 2,3,4,6,7 1,3,6,7	2,3,6,7 1,6,7 2,3,4,6,7 3,4,5,6,7	5,6,7,15,16,17 5,6,7,15,17 1,5,6,7,15,18 5,6,7,14,15,17 1,5,6,7,14,15,18 5,6,12,14,15,17 1,5,6,7,14,15 6,7,14,15	3,5,6,9,12,14,15 4,5,6,7,9,10,11,14,15,16,17 1,2,3,4,5,6,9,11,15,16,17 2,3,4,5,6,9,12,14,15,16,17 5,6,12,14,15,17 4,5,6,12,14,15,16 4,5,6,12,14,15,16,17 4,5,6,7,14,15,16

(b)

	Wine	
	SFS	SBS
DA	1,3,4,7,10,11,13 1,3,4,7,10,13 1,3,4,5,7 1,2,3,4,7,10,13 1,3,4,6,7,10,11,13	1,2,3,4,7,8,10,11,12,13 1,3,4,7,10,11,12,13 1,2,3,4,7,8,9,10,11,13 1,3,4,7,8,10,11,13 1,2,3,4,8,10,11,12,13 1,2,3,4,6,7,8,9,10,11,12,13 2,3,4,7,8,9,10,12,13 1,2,3,4,6,7,8,10,11,12,13
Opt. DA	1,5,7,11,13 1,3,4,6,7,8,9,10 2,5,6,7,10,13 1,3,5,7,11,13 1,7,10,11 1,4,7,10	1,3,4,7,8,9,10,11,12,13 2,5,7,8,10,11,13 1,3,4,7,11,12,13 1,3,6,7,11,13 1,2,3,4,7,8,9,10,12,13 1,2,3,4,5,7,8,9,10,12,13
K-NN	2,10,12,13 7,8,9,10 7,10 6,7,10 1,3,6,7,9,10,12 6,7,8,10,12 1,2,6,7,9,10	1,3,6,7,9,10 1,2,4,9,10,11,12 1,2,3,4,7,9,10 1,3,6,7,9,10,12 1,2,3,4,6,9,10,12 1,3,6,9,10,12 1,7,9,10 1,3,4,7,9,11,12
Opt. K-NN	7,10,13 1,7,10,11,13 1,7,8,10,13	1,4,5,6,7,10,12,13 1,3,4,8,9,11,13 1,3,4,9,10,11,12,13 1,5,7,11,13 1,5,7,10,12,13 1,4,5,6,7,8,9,10,11,12,13 1,2,4,7,8,9,10,12,13
NB	1,7,11 1,4,5,7,9,10,11,13 1,3,5,7,11,13 1,2,7,11,13 1,7,11,13 1,4,7,11,13 1,5,7,11	1,3,4,7,10,11,13 1,3,4,10,11,12,13 1,3,4,5,6,7,10,11,13 1,3,4,7,9,10,11,13 1,3,4,5,7,9,10,11,13 1,3,4,5,6,8,9,10,11,13 1,2,3,4,5,6,7,8,10,11,12,13
Opt. NB	1,7,11,13 1,3,4,7,11,13 1,3,4,7,10,11,13 1,4,5,7,11 1,4,7,11,13 1,2,7,11,13 1,5,7,11	1,3,4,5,7,9,10,11,13 1,5,7,10,11,13 1,3,4,7,10,11,13 1,5,7,9,10,11,13 1,2,3,4,5,6,7,8,10,11,12,13 1,3,5,7,8,9,10,11,13 1,3,4,7,9,10,11,13 1,2,3,4,5,7,8,9,10,11,12,13 1,3,4,5,7,10,11,13

(c)

	Sat	
	SFS	SBS
DA	2,5,6,13,14,17,18,19,20,23,24,25,26,27,33,34 9,10,14,17,18,23,24,31,33 4,7,8,10,16,17,18,19,20,21,23,31,33 5,9,10,17,18,19,22,23,24,30,31,32,33,36 2,4,8,10,12,13,15,17,18,19,20,21,24,26,27,28,34 4,17,18,19,20,21,24,25,27,28,32,33,34,35,36 4,14,15,17,18,19,21,22,25,27,34 2,11,17,18,19,27 1,6,13,14,17,18,19,23,24,25,27,28,30,33 4,12,16,17,18,19,21,22,25,26,27,28,29,35	1,2,4,5,6,7,8,9,10,11,13,14,16,18,19,20,21,22,23,24,26,27,29,30,31,32,33,34,36 2,3,4,6,7,8,10,12,13,14,16,17,18,19,21,22,23,26,27,28,29,30,32,33,34,35,36 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,32,33,34,35,36 1,2,3,4,5,6,8,9,10,12,13,14,15,17,18,19,21,22,24,25,27,28,29,30,32,33,34,35,36 2,3,4,5,6,8,10,11,13,15,17,18,19,20,21,22,23,25,26,28,30,31,32,33,34,35,36 1,2,3,4,5,6,7,8,10,12,13,14,15,17,18,19,20,21,22,24,25,27,28,30,31,32,33,34,35,36 1,2,3,4,5,6,7,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 3,4,5,6,7,8,10,11,12,13,15,16,17,18,19,20,21,22,23,24,25,27,28,30,31,32,33,34,35,36 2,3,4,5,7,8,10,11,13,14,16,18,19,21,22,23,24,26,27,28,30,31,32,33,34,35 4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,27,28,29,31,32,33,34
Opt. DA	3,11,15,17,18,20,21,23,26 3,11,15,17,18,20,23,24,25,26 3,9,13,14,15,17,18,20,23,31 3,9,11,17,18,20,21,22,23,25,26 3,11,15,17,18,20,23,24,25,26,31 7,13,14,16,17,18,19,20,23,24,25,27 3,11,13,14,16,17,18,19,20,23,24,27,28,31 3,11,17,18,19,20,21,22,23,25,26 3,4,11,13,15,17,18,20,21,23,24,25,26,35 7,13,14,16,17,18,19,20,23,24,25,27,31	3,4,6,8,9,13,14,17,18,19,21,22,23,24,25,26,27,28,29,31,33,34 1,3,4,6,7,8,9,10,11,13,14,16,18,19,20,21,22,23,25,27,28,29,30,33,34,35,36 1,2,4,5,7,8,11,12,13,14,15,16,18,19,20,21,22,23,25,26,27,28,29,30,31,33,34 1,2,4,5,7,8,10,11,13,18,19,20,22,23,24,25,26,27,28,29,30,31,33,34 1,2,3,4,7,8,9,10,11,13,14,17,18,19,20,21,22,24,25,26,27,28,29,31,33,34 1,3,4,5,7,8,9,10,11,13,14,16,17,18,19,20,22,23,24,25,27,28,29,30,31,33,34 3,4,5,6,8,9,10,11,13,14,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,33,34 1,2,4,6,7,11,12,13,14,15,18,19,20,21,22,23,25,26,27,28,31,32,33 1,2,4,5,8,9,10,11,13,14,15,17,18,19,20,22,23,25,27,28,29,30,31,33 1,2,3,4,5,7,8,9,11,12,13,14,16,17,18,19,20,21,22,23,25,27,28,29,30,31,32,33,34,35,36
K-NN	1,3,4,5,6,9,10,11,12,13,14,16,17,18,20,21,22,24,26,27,29,32,34,35 1,3,4,5,9,11,12,13,14,17,18,20,22,23,24,28,29,30,32,33 4,5,11,13,14,16,17,19,21,22,24,25,28,32,34,36 2,4,5,7,10,12,15,16,18,19,20,21,22,24,25,26,27,28,29,33,34,35,36 1,2,10,12,14,16,17,18,20,21,22,24,25,29,30,33 2,3,4,5,11,12,13,16,17,18,21,22,24,26,27,28,29,32,34 1,2,9,10,11,12,13,14,16,18,20,21,22,24,25,26,27,28,29,32,33,34,36 4,14,17,20,21,22,25,28,29,30,35,36 1,2,3,5,8,9,11,12,13,14,16,18,20,22,24,25,26,27,28,29,33,34,36 1,2,4,5,6,9,12,16,18,21,22,24,27,29,33,36	1,2,4,5,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 1,2,4,5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,32,33,34,35,36 2,4,5,7,8,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 1,2,4,5,6,7,10,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,32,33,34,35,36 1,2,3,4,5,8,9,10,11,12,14,16,17,18,19,20,21,22,23,24,25,27,28,29,30,32,33,34,35,36 1,2,3,4,5,6,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 1,2,4,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,29,30,31,32,33,34,35,36 1,2,3,4,5,6,8,9,10,11,12,14,15,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 1,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,31,32,33,34,35,36 1,2,3,4,5,6,8,9,10,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35
Opt. K-NN	1,2,9,10,11,12,13,16,18,20,21,22,24,25,26,27,29,32,33,34,36 2,9,13,14,16,22,24,27,28,29,30,36 1,2,4,5,8,11,14,17,18,19,20,22,24,25,28,29,33,35,36 7,9,10,12,13,14,16,17,18,22,24,25,27,29,36 1,7,8,9,11,14,15,18,21,22,24,25,28,34,36 2,3,5,10,11,14,16,17,18,20,22,24,25,26,27,29,32,33,34,36 13,16,18,21,22,24,27,28,29,32 1,3,13,16,18,21,22,24,26,29,32 5,8,9,13,14,16,17,18,20,21,22,24,27,28,29,30,32,36 3,4,5,10,12,13,16,18,21,22,24,26,32,34	1,2,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,29,30,31,32,33,34,35,36 1,2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,32,33,34,35 1,2,3,4,5,6,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,27,29,30,32,33,34,35 1,2,3,4,5,6,8,9,11,12,13,14,15,16,17,18,19,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 1,2,3,4,5,7,8,9,10,12,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,29,31,32,33,34,35,36 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35

Table Continues ↓

NB	<p>11,17,18,20,22,28,36 10,17,18,20,24,26,28,36 10,17,18,20,24,28 12,17,18,20,22,28,36 10,17,18,20,28</p>	<p>2,4,6,8,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,31,32,33,34,35,36 2,3,4,6,8,10,12,14,15,16,17,18,19,20,21,22,24,25,28,32,33,34,36 2,3,4,6,10,11,12,13,14,15,16,17,18,19,20,21,22,24,27,28,29,32,33,34,35,36 1,2,3,4,5,6,8,10,11,12,13,14,15,16,17,18,19,20,21,22,24,26,27,28,29,32,33,34,35,36 1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30,32,33,34,35,36 2,4,6,8,9,10,11,12,14,15,16,17,18,20,21,22,23,24,26,27,28,29,30,32,33,35,36 2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,20,21,22,23,24,26,27,28,30,32,33,34,35,36 2,3,4,6,7,8,9,10,12,13,14,15,16,17,18,19,20,21,22,24,26,27,28,30,32,33,34,35,36 1,2,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21,22,23,24,26,27,28,29,30,32,33,35,36</p>
Opt. NB	<p>4,10,12,18,29,32 2,3,4,13,18,20,21,24,28,33,34 2,4,10,12,13,16,18,24,29,32,34 2,4,12,13,16,18,22,28,29,36 2,3,4,13,16,18,20,21,24,28,33,34 4,10,12,13,18,29,32,34 2,4,10,16,18,21,24,25,28,34 2,6,9,12,16,18,19,24,25,29,32,34 2,4,13,16,18,21,24,34</p>	<p>2,3,4,6,8,9,10,11,12,13,14,17,18,19,20,21,22,24,25,26,28,29,31,32,33,34,36 2,4,6,9,10,11,12,14,16,17,18,20,22,24,25,26,28,29,32,33,34,36 1,2,4,6,7,8,9,10,11,12,14,15,16,17,18,20,21,22,23,24,25,26,27,28,29,30,32,33,34,35,36 2,3,4,6,9,10,11,12,14,16,17,18,20,22,24,25,26,28,29,30,33,34,36 2,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,28,29,31,32,33,34,35,36 2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,28,29,30,31,32,33,34,35,36 2,4,5,6,8,10,11,12,13,14,15,16,17,18,20,21,22,23,24,25,26,29,30,32,33,34,35,36 1,2,3,4,6,8,10,12,14,15,16,17,18,19,20,21,22,23,24,25,26,28,29,30,32,33,34,36 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,28,29,31,33,34,35,36 1,2,4,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,32,33,34,36</p>

(d)