ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Cevher ÖZDEN

THE EFFECTS OF PREPROCESSING METHODS ON PREDICTION OF TRAFFIC ACCIDENT SEVERITY

DEPARTMENT OF COMPUTER ENGINEERING

ADANA, 2018

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

THE EFFECTS OF PREPROCESSING METHODS ON PREDICTION OF TRAFFIC ACCIDENT SEVERITY

Cevher ÖZDEN

MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was received and approved the award of degree of the Master of Science by the board of jury on

…….....	.....	.....
Assoc. Prof. Dr. Zekeriya TÜFEKÇİ	Prof. Dr. S. Ayşe ÖZEL	Assoc. Prof. Dr. Serdar YILDIRIM
SUPERVISOR	MEMBER	MEMBER

This MSc Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.
Registration Number:

Prof. Dr. Mustafa GÖK
Director Institude of Natural and Applied Science

This study was supported by Research Projects Unit Ç.U.
Project Number:

ABSTRACT

MSc THESIS

THE EFFECTS OF PREPROCESSING METHODS ON PREDICTION OF TRAFFIC ACCIDENT SEVERITY

Cevher ÖZDEN

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
Year: 2018, Pages: 49
Jury : Assoc. Prof. Dr. Zekeriya TÜFEKÇİ
: Prof. Dr. S. Ayşe ÖZEL
: Assoc. Prof. Dr. Serdar YILDIRIM

The purpose of this thesis is to investigate the effects of different preprocessing approaches on the prediction accuracy of classifiers regarding the severity of traffic accidents. For this aim, six different classification methods, including J48, Ibk, Random Forest, OneR, Naïve Bayes and SMO have been used on an imbalanced dataset consisting of 99% nonfatal and 1% fatal traffic accidents that took place in Adana between 2005 and 2015. Various undersampling and oversampling approaches are tried to solve the imbalance problem and improve the classification accuracy. Then, the results of each method are compared to determine the best classifier and preprocessing method. Accordingly, SMO has attained higher accuracy in nearly all analyses, and it has produced the highest scores with the undersampled dataset consisting of equal amount of nonfatal and fatal instances.

Keywords: Traffic accident; injury severity, classification; preprocessing; machine learning.

ÖZ

YÜKSEK LİSANS TEZİ

ÖNİŞLEME YÖNTEMLERİNİN TRAFİK KAZALARININ ŞİDDETİNİN TAHMİNİ ÜZERİNE ETKİSİ

Cevher ÖZDEN

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Danışman : Doç. Dr. Zekeriya TÜFEKÇİ
Yıl: 2018, Sayfa: 49
Jüri : Doç. Dr. Zekeriya TÜFEKÇİ
: Prof. Dr. S. Ayşe ÖZEL
: Doç. Dr. Serdar YILDIRIM

Bu çalışmanın amacı farklı veri ön işleme yöntemlerinin trafik kazalarının şiddetini sınıflamadaki tutarlılığı üzerindeki etkisini incelemektir. Bu amaçla altı farklı sınıflama yöntemi, J48, Ibk, Random Forest, OneR, Naïve Bayes ve SMO, kullanılarak 2005-2015 yılları arasında Adana ilinde meydana gelen trafik kazalarını içeren ve %99 yaralanmayla, %1 ölümle sonuçlanan kazalardan oluşan veri seti üzerinde sınıflama yapılmıştır. Çeşitli veri azaltım ve veri çoğaltım yaklaşımları denenerek, verideki dengesizlikten kaynaklanan problem çözülmeye ve sınıflama tutarlılığı arttırılmaya çalışılmıştır. Analiz sonuçlarına göre en iyi sınıflayıcı yöntem ve veri ön işleme yöntemi belirlenmiştir. Buna göre, SMO neredeyse tüm analizlerde daha üstün bir performans sergilemiştir, ve en yüksek tutarlılık oranlarına ise eşit oranlarda ölümlü ve yaralanmalı kaza içeren, veri azaltımı uygulanmış veri kümesiyle ulaşmıştır.

Anahtar Kelimeler: Trafik kazası; yaralanma şiddeti, sınıflandırma; ön işlem; makine öğrenmesi.

GENİŞLETİLMİŞ ÖZET

Çalışmada Türkiye'nin en büyük metropollerinden birisi olan Adana'da gerçekleşen trafik kazaları ve kaza anına ait meteorolojik parametreleri içeren veri kümesi kullanılmıştır. Kaza verileri, trafik polislerince tutulan kayıtlardan oluşmaktadır ve Adana Trafik Şube Müdürlüğü'nden temin edilmiştir. Ayrıca kaza saatleri ve günleri dikkate alınarak Adana Meteoroloji Bölge Müdürlüğü'nden mevcut olan tüm meteorolojik gözlemler temin edilmiştir. Böylece, kazalar üzerinde etkili olabileceği düşünülen bütün çevresel bilgilere ulaşmak amaçlanmıştır. Elde edilen iki veri kümesi birleştirilerek elde edilen son veri kümesinde 14 kaza parametresi ve 10 meteorolojik parametre girdi olarak yer almıştır. Bu girdiler kullanılarak meydana gelen trafik kazaları sonuçları itibariyle ölümle ve yaralanmayla sonuçlanan olmak üzere sınıflandırılmaya çalışılmıştır. Elde edilen veri kümesi 25.015 kaza içermektedir, ancak bu kazaların sadece 246'sı (%0,63) ölümle sonuçlanmıştır. Diğer bir ifadeyle toplam kazaların %99,37 gibi tamamına yakını yaralanmalı kazalardan oluşmaktadır. Veri kümesi bu haliyle oldukça dengesiz bir yapıdadır. Tüm vakaları yaralanmalı olarak sınıflayıp hiçbir ölümlü vaka doğru sınıflanmasa dahi %99'un üzerinde bir başarı elde edilecektir.

Veri kümesindeki bu çarpıklığı gidermek amacıyla veri azaltımı ve veri çoğaltımı olmak üzere iki farklı yöntem uygulanmıştır. Veri çoğaltımı için, çoğunluk sınıfı olan yaralanmalı kazalara ait vakalar sabit tutulmuş ve ölümlü kazalar kopyalanarak sırasıyla 10 kat, 50 kat ve 100 kat arttırılmıştır. Ayrıca Python programlama dilinde hazırlanmış "Imbalance Learn" kütüphanesine ait SMOTE fonksiyonu kullanılarak ölümlü kazalar, yaralanmalı kazalarla sayıca denk olacak şekilde çoğaltılmıştır. Veri çoğaltımı aşamasında 4 farklı yeni veri kümesi elde edilmiştir. Elde edilen veri kümeleri üzerinde altı farklı sınıflayıcı kullanılarak, analiz sonuçları karşılaştırılmıştır. Yöntemlerin uygulanmasından önce veri kümelerinden eksik verilerin temizlenmesi, veri ayrıklaştırması gibi ön işlemler

yapılmıştır. Kullanılan sınıflayıcılar Karar Ağacı (J48), Random Forest, K En Yakın Komşu (Ibk), Naive Bayes, Destek Vektör Makinesi (SMO) ve OneR olup, tamamı 10 Kat Çapraz Doğrulama yapılarak WEKA 3.8 yazılımı ile uygulanmıştır. 10 Kat Çapraz Doğrulama için her veri kümesi rasgele bir şekilde 10 alt bölüme ayrılmıştır. Ölümlü ve yaralanmalı kazalar her alt bölüme eşit sayıda düşecek şekilde ve rasgele olarak dağıtılmıştır. Her döngüde 1 alt küme test amacıyla ayrılarak geriye kalan 9 alt küme eğitim amaçlı kullanılmış ve elde edilen modeller ayrılan test kümesi üzerinde denenmiştir. Yapılan analizler neticesinde en başarılı yöntem olarak SMO öne çıkmıştır. SMO en yüksek sınıflama başarısını ölümlü kazaların 100 kat arttırılarak elde edilen üçüncü veri kümesinde elde etmiştir ve sınıflama tutarlılıkları %62.2 yaralanmalı kaza, %58.9 ölümlü kaza ve %60.6 toplam tutarlılık şeklinde kaydedilmiştir.

Çalışmanın veri azaltımı aşamasında, çoğunluk sınıfı olan yaralanmalı kazaların sayısı, ölümlü kazaların 5 katı (1-5), 3 katı (1-3) ve eşit miktarı (1-1) olacak şekilde ve rasgele olarak azaltılmıştır. Ayrıca Imbalance Learn Python kütüphanesine ait Random Undersampler fonksiyonu kullanılarak yaralanmalı kazaların sayısı ölümlü kazaların sayısıyla aynı olacak şekilde azaltılarak dördüncü bir veri kümesi daha elde edilmiştir. Elde edilen veri kümeleri üzerinde çalışma için seçilen altı sınıflama yöntemi 10 kat Çapraz Doğrulama yapılarak uygulanmıştır. Analiz sonuçlarına göre en başarılı yöntem olarak yine SMO öne çıkmıştır. Bu bölümde SMO en yüksek başarısını eşit sayıda ölümlü ve yaralanmalı kaza içeren üçüncü veri kümesi olan 1-1 Veri kümesi ile elde etmiştir ve sınıflama tutarlılıkları %53.8 yaralanmalı kaza, % 69.2 ölümlü kaza ve % 61.5 toplam tutarlık şeklindedir.

Buna göre en başarılı yöntem olarak SMO ön plana çıkarken, 1-1 Azaltılmış Veri Kümesi bu yöntemin en yüksek tutarlılık gösterdiği veri kümesi olarak belirlenmiştir. Çalışmanın sonraki aşamasında 1-1 Azaltılmış Veri Kümesi üzerinde dört farklı nitelik seçimi algoritması çalıştırılarak sınıflamada daha belirleyici olan girdiler saptanmış ve SMO sınıflayıcısı seçilen girdilerle tekrar uygulanmıştır.

Kullanılan nitelik seçimi algoritmaları “Gain Ratio”, “Relief”, “Information Gain” ve “Correlation Based Feature Selection (CFS)” olup, tamamı WEKA 3.8 ortamında uygulanmıştır. Nitelik seçimi analiz sonuçlarına göre, CFS üç nitelik (3: Kaza Lokasyonu, 8: Geçit, 9: Trafik Kontrol), Information Gain beş nitelik (9: Trafik Kontrol, 3: Kaza Lokasyonu, 1: Gün, 8: Geçit, 2: Kaza Saati), Gain Ratio beş nitelik (13: Engel Nesne, 8: Geçit, 3: Kaza Lokasyonu, 9: Trafik Kontrol, 11: Banket), ve son olarak Relief iki nitelik (9: Trafik Kontrol, 2: Kaza Saati) belirlemiştir. SMO yöntemi 1-1 azaltılmış veri seti üzerinde belirlenen niteliklerle tekrar uygulanmıştır. Uygulama sonuçlarına göre en yüksek tutarlılık CFS ile belirlenen üç nitelik ve Gain Ratio ile belirlenen beş nitelik ile elde edilmiştir. Tutarlılık oranları yaralanmalı kazalar için %52.8, ölümlü kazalar için %69.9 ve toplamda %61.4 şeklidedir.

Çalışmanın son kısmında, 1:1 azaltılmış veri kümesi üzerinde Hata Duyarlılık Analizi (Cost Sensitive Analysis) Bagging ve Adaboost yöntemleri ile birlikte iki farklı hata matrisi kullanılarak uygulanmıştır. Ayrıca aynı analizler bir önceki aşamada CFS ile belirlenen 3 nitelik (3, 8, 9) ile tekrar edilmiştir. Analiz sonuçlarına göre, hata matrisinde ufak değişikliklerin diğer sınıf tutarlılığında büyük kayıplara sebebiyet verdiği belirlenmiştir. Ayrıca bu kısımda en yüksek tutarlılık oranlarına üç nitelik ve varsayılan hata matrisi kullanılarak uygulanan Adaboost uygulamasıyla elde edilmiştir. Tutarlılık oranları yaralanmalı kazalar için %51.6, ölümlü kazalarda %68.7 ve toplamda %60.2 olarak kaydedilmiştir.

Sonuç olarak, SMO tüm analizlerde başarısıyla ön plana çıkmış ve 100 kat arttırılmış veri kümesi, SMOTE veri kümesi, RandomUndersampler veri kümesi ve 1-1 azaltılmış veri kümelerinde birbirine yakın sonuçlar üretmiştir. F-skoru da dikkate alındığında SMO en yüksek başarısını 1-1 azaltılmış veri kümesi üzerinde ve tüm nitelikler kullanılarak yapılan analiz ile elde etmiştir. Ancak en iyi tutarlılık oranlarının dahi %60 seviyesinin biraz üzerinde olduğu dikkate alınırsa, çalışmada ele alınan veri ön işleme yöntemlerinin sınıflama tutarlılığı üzerinde sadece sınırlı bir iyileşme sağladığı görülmektedir. Bunun muhtemelen en büyük nedeni kaza

tutanaklarında kaydedilen verilerin kazaların oluş şekli ve sonucu üzerinde yeterli bilgi içermemesinden kaynaklanmaktadır. Özellikle kazaya karışan sürücüler hakkında sosyo-ekonomik bilgilerin olmaması ve ayrıca araçlarla ilgili teknik bilgilerin kaydedilmemesi veri setinin oluşturulmasında büyük eksiklikler olarak göze çarpmaktadır.

ACKNOWLEDGMENTS

I would like to express my gratitude to the Prof. Dr. Selma Ayşe ÖZEL and Assoc. Prof. Dr. Zekeriya TÜFEKÇİ for their valuable contributions and recommendations. This research was supported by General Directorate of Security-Traffic Services Department and Turkish State Meteorological Service.

CONTENTS **PAGES**

LIST OF TABLES **PAGES**

LIST OF FIGURES **PAGES**

LIST OF ABBREVIATIONS

ACC	: Total Accuracy
AUC	: Area Under Curve
CART	: Classification and Regression Tree
CCNFC	: Correctly Classified Nonfatal Class
CRO	: Crossing Point
DAY	: Crash Day of Week
DBM	: Dynamic Bayesian Network model
DIV	: Roadway Division
DTC	: Decision Tree Classifier
FF	: False Fatal
GI	: Gini Index
GSR	: Total Global Solar Radiation
GST	: Ground Surface Temperature
KNN	: k Nearest Neighbor
LOC	: Crash Location
LR	: Logistic Regression
MAR	: Pavement Marking
MCL	: Mean Cloudiness
MLP	: Multilayer Perceptron
MNMP	: Minimum Mean Pressure
MNTP	: Minimum Temperature
MXMP	: Maximum Mean Pressure
MXTP	: Maximum Temperature
MRH	: Mean Relative Humidity
MWS	: Mean Wind Speed
NBC	: Naive Bayes Classifier

PER	: Crash Time Period
ROC	: Receiver Operator Curve
SHO	: Shoulder
SLO	: Sloppiness
SMO	: Sequential Minimal Optimization
SMOTE	: Synthetic Minority Over-sampling Technique
SVM	: Support Vector Machine
SUR	: Roadway Surface
TP	: Total Precipitation
TNFR	: True Nonfatal Rate
TRA	: Existence of Traffic Control
TSI	: Turkish Statistics Institute
WHO	: World Health Organization

1. INTRODUCTION

Improvements in data storage capacity and processing units have brought about many changes in the way the data is used. Data warehouses contain enormous amount of data, which continues to increase constantly. But largeness of data does not necessarily present valuable information by itself. And, data mining is the term for extracting valuable information from the patterns in data and converting it into useful knowledge (Sayın, 2013). The main purpose of data mining is to support decision making. Data mining can be used in any data-related field. And traffic is one of the fields where data mining has not fulfilled its potential so far.

Each year, over 1 million people lose their lives in traffic accidents and another 50 million people are subject to injuries of varying severity throughout the world (WHO, 2015). In addition, traffic accidents are one of the major sources of suffering to victims, their relatives and countries. According to WHO, traffic accidents are responsible for economic losses equivalent to around 3 % of GDP in developing countries (WHO, 2016). The number of motor vehicles was increased two-time from 2005 to 2015 in Turkey (Table 1.1). On the other hand, traffic accidents were increased more in the same period. Around 1% of Turkey's population die in traffic accidents and 4% incur injuries (TSI, 2018).

Table 1.1. Summary statistics of traffic accidents in Turkey (TSI, 2016)

Year	Total vehicle	Number of traffic accidents	Killed persons		Injured persons	
			Number	Ratio to population	Number	Ratio to population
2005	11,145,826	620,789	4,505	0.06	154,086	2.14
2006	12,227,393	728,755	4,633	0.06	169,080	2.32
2007	13,022,945	825,561	5,007	0.07	189,057	2.68
2008	13,765,395	950,120	4,236	0.06	184,468	2.58
2009	14,316,700	1,053,346	4,324	0.06	201,380	2.78
2010	15,095,603	1,106,201	4,045	0.05	211,496	2.87
2011	16,089,528	1,228,928	3,835	0.05	238,074	3.19
2012	17,033,413	1,296,634	3,750	0.05	268,079	3.54
2013	17,939,447	1,207,354	3,685	0.05	274,829	3.58
2014	18,828,721	1,199,010	3,524	0.05	285,059	3.67
2015	19,994,472	1,313,359	7,530	0.09	304,421	3.86

The aim of this thesis is to analyze the effects of different preprocessing techniques over classification accuracy of traffic accidents and build an efficient decision support system for traffic authorities in their combat against traffic accidents. For this purpose, a dataset containing a total of 25,015 traffic accident records kept by traffic officers is used in the study. The dataset covers the period between 2005 and 2015. Also, meteorological observations are obtained from Turkish State Meteorological Services to include additional attributes to the dataset, which could have contributed to the occurrences of the accident. The dataset used in this study is quite imbalanced, over 99% of which consists of accidents resulting only in injury and less than 1% of fatal accidents. In order to overcome the imbalanced structure of the dataset, both undersampling and oversampling approaches are adopted along with different machine learning methods in the study, and the best method and dataset combination are determined. Subsequently, feature selection is applied to observe if it makes any improvement in the classification results. Subsequently, cost sensitive classifier with bagging and boosting methods are applied. And, the results are compared to find the best solution in terms of both dataset and method applied. In light of the findings, the efficiency of the accident

reports kept by traffic officers is discussed, and new suggestions are made to enable better representation of the causes of traffic accidents.

2. PREVIOUS STUDIES

There are many studies in literature that have tried to develop injury severity models using meteorological parameters. The previous studies mainly employed both statistical and machine learning models.

Osman et al. (2016) researched the injury severity of traffic accidents involving trucks in working areas and the related parameters that contributed to the occurrence of accidents. Their dataset included one meteorological parameter classified as Wet or Dry. Kaplan and Prato (2012) analyzed the risk factors related to the severity of traffic accidents in USA. Collision type, the features and behaviors of drivers, urban infrastructure, and environment were among the factors investigated. The dataset included one meteorological parameter characterized as either Good or Adverse. In their study, Kim et al. (2013) employed mixed logit model to investigate the severity levels of traffic accidents in California, USA. Their dataset included driver features and meteorological parameters. Similarly, Xie et al. (2012) studied the injury severity levels of traffic accidents and related factors. They also included meteorological parameters in their dataset. On the other hand, Shon and Shin (2010) used certain machine learning methods on traffic accident dataset recorded in Korea. Their primary target is to compare the accuracy outcomes of methods they included in their study. Consequently, no significant difference could be detected between methods, while the wearing protection was found the most important parameter in the study. Wu et al. (2016) investigated the severity of traffic accidents that took place in New Mexico between 2010 and 2011. In their study, different logit models were developed to detect the effective features on the accident severity. Their dataset included meteorological parameters. As a result, five (5) factors were determined to be important on the severity of accidents for rural area, while six (6) parameters were found significant for urban areas.

Sun and Sun (2015) studied the relation between accident occurrence and speed using Bayes Neural Network and they could predict the traffic accident

occurrence with 76.4% of accuracy using speed and state of congestion. Prato et al. (2011) researched the traffic accidents that took place in Israel between 2003 and 2006 using Neural Networks. They intended to determine patterns and contributing factors, and consequently, they concluded five patterns including driver and victims features like age and profession. Taamneh et al. (2016) investigated traffic accidents in Abu Dhabi using several machine learning methods. They concluded that the victims consisted mainly of 18—30 years old people, and nationality, collision type, gender, year and age are the most relevant factors for accidents. Castro and Kim (2015) studied the traffic accidents that took place in UK between 2010 and 2012 using several machine learning methods. Maneuver, lightning and road condition were determined as the most effective factors on the occurrence of traffic accidents. Oiu et al. (2014) investigated the factors that affect the severity of traffic accidents using machine learning methods on a dataset consisting of accidents between 2008 and 2010 in Beijing. They determined meteorological conditions, protective gears, road division and gender as the most effective factors on accident severity. Dadashova et al. (2016) studied the factors that contribute to the occurrence of traffic accident in Spain using decision tree and random forest methods, and they concluded the road design as the most effective parameter. Ozden and Acı used Naïve Bayes, Multilayer Perceptron, Support Vector Machine, Decision Tree, K-Nearest Neighbor and Logistic Regression methods for binary classification of traffic accidents, either fatal or non-fatal (Ozden and Acı, 2018). They used traffic accident records combined with meteorological data between 2005 and 2015. The data set contains 24 attributes including information about road, location, time and weather. As a result, they determined that DTC and KNN provided higher accuracy, and Mean Cloudiness, Traffic Control Signs and Ground Surface Temperature played more important roles in classification result.

Some of the studies that used weather parameters and machine learning methods are listed in the Table 2.1.

Table 2.1. Previous studies on injury severity

Authors	Methods	Weather information
Taamneh et al. (2016)	Artificial Neural Network, Hierarchical Clustering	Clear, rainy, dusty, fog
Chen et al. (2016)	Decision Table/Naive Bayes Hybrid Classifier	Clear, snow, rain
Taamneh et al. (2016)	Naive Bayes, Multilayer Perceptron, Decision Tree	Clear, rainy, dusty, fog
Castro and Kim (2016)	Bayesian Network, Decision Tree, Artificial Neural Network	Fine without high winds, Raining without/with high winds
Chen et al. (2016)	Support Vector Machine	Sunny, adverse
Zeng and Huang (2014)	Back-Propagation Neural Network	Clear, not clear
Qiu et al. (2014)	Particle Swarm Optimization	Sunny, rain, other
Li et al. (2016)	Support Vector Machine	Clear, other
Prato et al. (2011)	Feed-Forward Back-Propagation Neural Network	Clear, rainy, hot, foggy, not specified
Tavakoli et al. (2011)	Classification and Regression Tree	Clear, fog, rain, snow, stormy, cloudy, dusty
Kunt et al. (2011)	Multilayer Perceptron Neural Networks, Genetic Algorithm	Clear, snowy, rainy, cloudy
Chang and Wang (2006)	Classification and Regression Tree	Clear, rain, or fog
Aci and Ozden (2018)	Predicting the Severity of Motor Vehicle Accident Injuries in Adana-Turkey Using Machine Learning Methods and Detailed Meteorological Data	Mean wind speed, mean pressure, maximum temperature, minimum temperature, mean cloudiness, mean relative humidity, solar radiation, surface temperature, precipitation

The previous studies have mainly aimed to find the best classifier for a given traffic dataset without investigating the effect of various preprocessing. Another aim of these studies has been to determine the factors that promote the occurrence of traffic accidents. However, traffic accident datasets consist of mostly nonfatal accidents that cause injury or property damage, and only the minor part of these datasets includes fatal accidents involving one or more casualties. And due to this

nature of the traffic accident datasets, researchers have to deal with the problem of imbalanced distribution. And, the most important peculiarity of this thesis is that the main aim is to analyze the efficiency of two main preprocessing approaches i.e. oversampling and undersampling along with different feature selection algorithms on the accuracy of classifiers given a dataset consisting 99% of nonfatal and 1% of fatal accidents.

3. MATERIAL AND METHODS

3.1. Accident Data and Meteorological Data

The study is based on a dataset consisting of traffic accident reports and meteorological parameters pertaining to Adana between 2005 and 2015. Traffic data was obtained from the Traffic Services Department in Adana, and the meteorological parameters were provided by Turkish State Meteorological Services. Traffic data contains 13 attributes such as (1) Day of Week, (2) Crash Time Period, (3) Location of Accident, (4) weather status, (5) Division of Road, (6) Roadway Surface, (7) Sloppiness of Road, (8) Crossing, (9) Traffic Control, (10) Pavement Marking, (11) Shoulder, (12) Construction, and (13) Deterrent Object on Road. Data discretization was applied on the traffic dataset. On the other hand, Meteorological data include 10 parameters, which are (1) Mean Wind Speed (m/sec), (2) Maximum Mean Pressure (hPa), (3) Maximum Temperature (°C), (4) Minimum Mean Pressure (hPa), (5) Minimum Temperature (°C), (6) Average Cloudiness, (7) Mean Relative Humidity (%), (8) Total Global Solar Radiation (cal/cm²), (9) Daily Precipitation (mm) and (10) Surface Temperature (°C). All parameters are normalized within the range of 0 and 1. Descriptive Statistics are given in Table 3.1.

Table 3.1. Descriptive statistics for meteorological data

Attribute Name	Symbol	Min.	Max.	Mean	St.D.
Mean Wind Speed (m/sec)	MWS	0.0	4.2	1.2	0.5
Maximum Mean Pressure (hPa)	MXMP	999.9	1027.4	1010.6	5.2
Maximum Temperature (°C)	MXTP	8.0	39.9	26.7	7.6
Minimum Mean Pressure (hPa)	MNMP	995.9	1023.0	1007.3	4.9
Minimum Temperature (°C)	MNTP	-3.0	27.6	15.5	7.1
Mean Cloudiness	MCL	0.0	10.0	3.8	1.5
Mean Relative Humidity (%)	MRH	27.8	95.3	70.2	12.9
Total Global Solar Radiation (cal/cm²)	GSR	29.4	673.8	390.4	137.8
Total Precipitation (mm)	TP	0.0	53.0	1.1	4.9
Ground Surface Temperature (°C)	GST	-6.1	26.4	13.2	7.8

The initial dataset contained 25,015 accidents, and only 246 of these accidents resulted in death, while more than 99% of the accidents were injury-related and non-fatal cases. Because of this imbalanced structure of dataset, any classifier would attain over 99% of total accuracy by classifying all cases as nonfatal accident. And, the aim of this study is to try different approaches to overcome this imbalanced structure of the dataset. For this purpose, both undersampling and oversampling methods are employed in the study along with six machine learning methods, including Decision Tree Classifier (J.48), Random Forest, OneR, K Nearest Neighbor (Ibk), Support Vector Machine (SMO) and Naïve Bayes. All machine learning methods were implemented in Weka 3.8 Software.

To oversample the minority class (fatal instances), all fatal accidents were copied, and their number is increased by 10-time, 50-time and 100-time. Also, the SMOTE Function of Imbalanced Learn Python API was used in Jupyter Notebook environment to obtain another oversampled dataset for comparison purpose. So, four different subdatasets were created

To undersample the majority class (nonfatal instances), the number of nonfatal instances was randomly reduced to 5-fold (1,230), 3-fold (738) and 1-fold (246) of the number of fatal instances (246). For comparison purpose, Random

Undersampler function of Imbalanced Learn Python API was used to produce another undersampled dataset consisting of equal amount of fatal and nonfatal instances (246-246). Thus, four different undersampled subdatasets were obtained. And, six classification methods were applied to four oversampled and four undersampled subdatasets to determine the best classifier and subdataset. Then, four different feature selection algorithms, including Information Gain, Gain Ratio, Relief and Correlation Based Feature Selection (CFS) were applied to the determined subdataset. And, the best classifier was re-applied with the selected attributes to see if it improves the accuracy. For the last analysis, Cost Sensitive Classifier with Bagging and AdaBoost was applied to the subdataset to determine the effects of different cost penalization scenarios over accuracy.

3.2. k Nearest Neighbor (kNN) Method

The K Nearest Neighbor method is based on the distance computation, in which classification is made with respect to the majority votes of the nearest neighbors (Cover and Hart, 1968). Training process consists of storing feature vectors and labels of the training instances. During classification, the unlabeled instance is assigned to the label of its k nearest neighbors. If there is only 1 neighbor, then the instance is classified as the same as the object nearest to it. In the case of two classes, k has to be an odd integer. Euclidean distance is used as the distance function in the study:

$$d(x,y) = \|x-y\| = \sqrt{(x-y).(x-y)} = (\sum_{i\_1}^{m}((x_i - y_i)^2))^{1/2}$$

where x is the object to be classified and y is the training instance such that x and y are in $X = R^m$. Figure 3.1. shows the process of KNN classification.

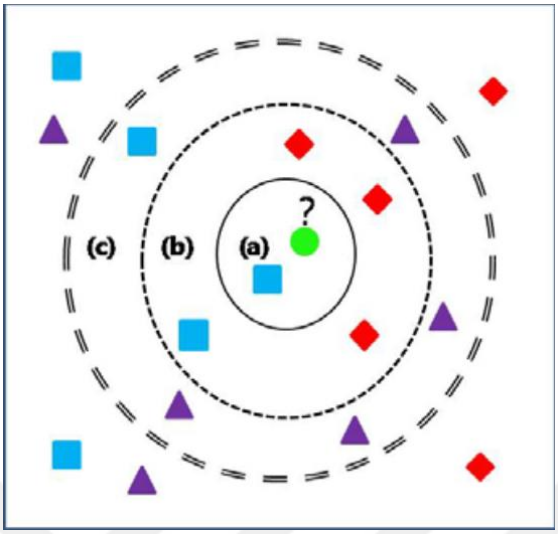

Figure 3.1. kNN classification (Bronshtein, 2017)

In Figure 3.1, the circle (? marked) depends on the *k* value of 1, 5, or 10. And, its class can be queried using the classes of the instances at (a), (b) and (c). KNN performs well with multi-modal classes as it decides considering the small neighborhood of similar instances. Therefore, even if the target class is multi-modal, KNN might still yield a good accuracy. On the other hand, the downside of the KNN is that it measures all the features equally to evaluate the similarities, which could result in bad classification, especially in the case of a small subset of useful features.

3.3. Decision Tree Classifier (DTC)

Decision Tree is expressed as a recursive partition of the instance space (Breiman et al., 1984). It has a structure consisting of nodes of a rooted tree. Apart from the root node, other nodes have one incoming edge. And, a node with outgoing edges is test node, while the remaining nodes are leaves. In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a definite discrete function of the input attributes (Rokach, 2016).

Each leaf is attributed to a class based on the attributes of the proper target. The leaf might have a vector showing the probability of the target. Classification is

made starting from the root of the tree down to a leaf with respect to the outcome of the tests through the path. Breiman et al. (1984) indicates that complexity of a decision tree has a profound effect on the accuracy and it can be explicitly managed by the stopping criteria and the pruning method. There are different top-bottom decision trees like ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), and CART (Breiman et al., 1984). These algorithms have greedy nature and build the decision tree in a top–bottom, iterative manner. In each iteration, the algorithm partitions the training examples considering the outcome of a discrete function. The most appropriate function is selected by certain splitting criteria. Then, each node further subdivides the training examples into smaller subsets, until no split gains sufficient splitting score or a stopping criterion is satisfied.

In the study, DTC was implemented using the J48 implementation of WEKA. This version of DTC is based on the C4.5 originally devised by Quinlan (2013). J48 uses the normalized version of Information Gain for building trees as the splitting criteria. It has both reduced error pruning and normal C4.5 pruning option. It has C4.5 pruning option as default, which is kept in this study.

3.4. Random Forest

Random Forest is a class of ensemble methods especially designed for decision tree classifiers (Kam, 1995). The logic behind its structure is that it combines predictions made by many decision trees. In a random forest algorithm, each tree is produced based on a bootstrap sample and the values of a distinct set of random vectors. The random vectors are produced based on a fixed probability distribution. The structure of generating a random forest is based on sampling a dataset with replacement, then selecting m variables from p variables randomly and creating a tree in this way, after creating more trees by repeating the same procedures, the results are combined eventually.

3.5. OneR

OneR is a simple but accurate classification algorithm that produces one rule for each predictor in the data, then chooses the rule with the minimum error as its "one rule" (Holte, 1993). In order to create a rule for a predictor, a frequency table is formed for every predictor against the target. Thus, OneR yields rules only slightly less accurate than other modern classification methods; however, the rules produced by OneR are simple for human interpretation (Sayad, 2018).

OneR Algorithm (Sayad, 2018):

For each predictor,

For each value of that predictor, make a rule as follows;

Count how often each value of target (class) appears

Find the most frequent class

Make the rule assign that class to this value of the predictor

Calculate the total error of the rules of each predictor

Choose the predictor with the smallest total error.

3.6. SMO Support Vector Machine

Support Vector Machine (SVM) is a kernel-based learning algorithm where only a part of the training set is used in the solution (these are called the support vectors), and the aim of learning is to maximize a margin around the decision surface (Bernhard et al., 1992). SVM classification is made in the following order: first map the input vectors into one feature space (possibly a higher dimension), either linearly or nonlinearly, which is related to the selection of the kernel function; then within the feature space, seek an optimized linear division, i.e. construct a hyperplane which separates two classes (Chong et al., 2005). SVM is implemented in the WEKA environment.

3.7. Naive Bayes

The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors (Russel and Norvig, 2003). A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes Classifier (NBC) assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence (Sayad, 2018). NBC is an effective and simple method for classification problems. It can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. It is based on the Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. It is well known method and has been defined in many statistics and machine learning books before (Bishop, 2006; Mitchell, 1997).

3.8. Cost Sensitive Analysis

Cost Sensitive Learning is a type of analysis in data mining that takes the misclassification costs (and possibly other types of cost) into consideration (Saltelli, 2002). The goal of this type of learning is to minimize the total cost. The key difference between cost sensitive learning and cost insensitive learning is that cost sensitive learning treats the different misclassifications differently. Cost-insensitive learning does not take the misclassification costs into consideration. The goal of this type of learning is to pursue a high accuracy of classifying examples into a set of known classes. The imbalanced datasets occur in many real-world applications where the class distributions of data are highly imbalanced. Cost sensitive learning is a common approach to solve this problem.

The misclassification cost values can be given by domain experts or learned via other approaches. In cost-sensitive learning, it is usually assumed that such a cost matrix is given and known. For multiple classes, the cost matrix can be easily extended by adding more rows and more columns (Ling and Sheng, 2008).

3.9. Bagging

Bagging, short for "bootstrap aggregating", is an ensemble learning approach which generates multiple exemplars of a predictor to lead to an aggregated learner by taking the combination of their outputs using a fixed rule (Breiman, 1996). It provides a way to present variability between the different models. Creation of the multiple exemplars is done via making bootstrap replicates of the learning set. Logic behind the bootstrap creation is treated as follows. Assume that we have a dataset X = {x1, ..., xm} with m data points. If we generate a new dataset XBagged whose instances are randomly drawn from the original dataset as the same number of instances with replacement, it is the case where some number of data points are repeated containing duplicates in XBagged and some others in the original dataset are not included. This difference between bootstrap models is exactly what we want to give rise to diversity among the models in the ensemble. An iterative process is performed by repeating this procedure K times and resulting in K randomly generated datasets (Tüysüzoğlu, 2016).

3.10. Adaboost

Boosting is an approach to machine learning based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules (Freund and Schapire, 1999). The AdaBoost algorithm was the first practical boosting algorithm. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak

learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner (Yıldırım, 2010).

3.11. SMOTE Function

SMOTE is an over-sampling approach which creates "synthetic" examples of the minority class instead of oversampling with replacement (Chawla et al., 2002). For this purpose, certain operations like rotation and skew are performed to perturb real data. In this way, synthetic examples are produced in a less application-specific manner through operating in feature space rather than data space. SMOTE uses k minority class nearest neighbors where k is randomly chosen depending upon the required amount of oversampling and it generates one sample in the direction of each neighbors. This results in a larger and less specific decision regions for the minority class (Chawla et al., 2002).

3.12. Feature Selection Algorithms

The main aim of machine learning is to approximate the functional relationship between the input and the output variables. However, the output is not necessarily determined by the complete set of the input features, and it is sometimes decided only by a subset of them. When data and time are abundant, it is fine to use all the input features, even the irrelevant ones, to approximate the underlying function between the input and the output. But in practice, there are two problems which may be evoked by the irrelevant features involved in the learning process (Deng, 1998).

1.The irrelevant input features will induce greater computational cost.

2.The irrelevant input features may lead to overfitting.

Another motivation for feature selection is that, since our goal is to approximate the underlying function between the input and the output, it is reasonable and important to ignore those input features with little effect on the output, so as to keep the size of the approximator model small.

In the study, four types of Feature Selection Algorithms are used in this thesis, including Information Gain, Gain Ratio, Relief and Correlation Based Feature Selection (CFS) algorithm implemented in WEKA. Entropy is commonly used to characterize the purity of an arbitrary collection of dataset. It is the foundation of feature selection methods (Novakovic et al., 2009). Entropy of *T* is:

$$H(T) = -\sum_{t \in T} p(t) log_2^{(p(t))}$$

Here, $p(t)$ is the marginal probability density function for random variable *T*. If the observed values of *T* in the training dataset are portioned according to the values of a second feature *X* and the entropy of *T* with respect to the partition is less than the entropy of *T* prior to the partitioning, then it is accepted that there exists a relationship between features *T* and *X*. The entropy of *T* after observing *X* is given:

$$H(T|X) = -\sum_{x \in X} p(x) \sum_{t \in T} p(t|x) log_2^{(p(t|x))}$$

Here, $p(t|x)$ is the conditional probability of *t* given *x*.

3.12.1. Information Gain

Considering the entropy as a criterion of impurity in a training set, Information Gain (IG) can be defined to reflect additional information about T provided by X representing the proportion as the entropy of T decreases (Mitchell, 1997). It is given by the formula below:

$$IG = H(T) - H(T|X) = H(X) - H(X|T)$$

The information gained about T after observing X is equal to the information gained about X after observing T. IG is biased towards features that have more values even if they are not more informative, which is the main weakness of IG.

3.12.2. Gain Ratio

Gain Ratio (GR) proposes nonsymmetrical measure and is introduced to eliminate the bias problem of IG (Novakovic et al., 2009). It is described by the formula below:

$$GR = \frac{IG}{H(X)}$$

IG is normalized through dividing by the entropy of X, and vice versa. Thanks to this normalization, GR falls in the range [0,1], where GR=1 indicates that X fully predicts T, while GR=0 shows that there exists no relation between X and T.

3.12.3. Relief

Relief evaluates the value of each feature through repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different class. This feature evaluation assigns a weight to each feature in terms of the ability of the feature to distinguish among the classes, and then selects the features with weights over a predefined threshold as relevant features (Karegowda et al., 2010).

3.12.4. Correlation Based Feature Selection (CFS)

CFS takes into account the interactions between attributes. It evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy among them. Correlation coefficient is used to estimate correlation between subset of attributes and classes as well as inter-correlations between the features (Karegowda et al., 2010). Relevance of a group of features grows with the correlation between features and classes and decreases with growing inter-correlation.

3.13. Performance Metrics

There are many different approaches used in literature to measure the performance of classifiers. In the study, *accuracy* and *F-Measure* are used for this purpose. *Accuracy* is measured by counting the proportion of correctly predicted examples in a dataset (WEKA, 2013). WEKA produces detailed accuracy scores and 2x2 confusion matrix for each analysis, as there are two classes in the study.

The *True Positive (TP)* is the number of correctly classified examples among all examples that are classified as positive which is the class of interest.

The *False Positive (FP)* is the number of incorrectly classified examples as positive among all examples which are not in the positive class.

The *True Negative (TN)* represents the number of negative instances and classified as so.

The *False Negative (FN)* is the number of instances that are positive but classified as negative.

The *Precision* is the proportion of correctly classified examples of class of interest among all examples classified in that class. It is calculated as follows:

$$Precision = TP/(TP + FP)$$

The *Recall* is the proportion of correctly classified examples of class of interest among all examples in that class. In this study, *Recall* is used to represent *Accuracy*. It is calculated as follows:

$$Recall = TP/(TP + FN)$$

The *F-Measure* is calculated from the following formula:

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The *Macro Average* is simply the mean of *F-Measures* of two classes.

4. RESULTS AND DISCUSSION

This part of the study consists of four main sub sections. In the first section, oversampling approach is employed to overcome the problems caused by the imbalanced structure of the dataset. For this purpose, the minority class (fatal instances) was copied by 10-time, 50-time and 100 time. Also, SMOTE function of the Imbalance Learn Python API was used to oversample the fatal instances to the same number of non-fatal instances. Thus, six machine learning methods (J48, Ibk, Random Forest, Naïve Bayes, SMO, OneR) were applied to the derived four oversampled datasets.

In the second section of this part, undersampling approach is employed and the number of nonfatal instances (majority class) was gradually reduced to 5-fold, 3-fold and 1-fold (equal amount) of the number of fatal instances. Also, Random Undersampler function of Imbalance Learn Python API was used to equalize the numbers of nonfatal and fatal instances. Then, the six classification methods (J48, Ibk, Random Forest, Naïve Bayes, SMO, OneR) were applied to the four undersampled datasets, as well. Further details and the results are given below for each dataset.

In the third section, four different feature selection algorithms, including Gain Ratio with Ranker search method, Relief with Ranker search method, Information Gain with Ranker search method and Correlation Based Feature Selection (CFS) with BestFirst search method, are implemented to determine the features to be included in the further analysis. And the best classifier and sub-dataset is re-applied with the selected features to see if it improves the accuracy.

In the fourth and last section of this part of the study, cost sensitive classifier is applied with bagging and adaboost on the determined subdataset and features. Different cost matrices are tried to penalize both classes to observe how the accuracy changes.

4.1. The Results with Oversampling Methods

In this section of the study, four different oversampling methods were applied to obtained higher accuracy in the classification. For this purpose, the data cleaning is applied, and all unrepairable data are excluded from the dataset. After this process, 22,490 accidents left. All the excluded instances are nonfatal accidents, and the number of fatal accidents remains the same (246). Then, non-fatal instances were kept the same, and the fatal instances are, respectively, copied by 10-time, 50-time and 100-time to increase their number to bring gradual balance to the dataset. Also, the SMOTE function of the imbalanced-learn Python API was applied. The SMOTE function oversamples the rare event by using bootstrapping and k-nearest neighbor to synthetically create additional observations of that event.

4.1.1 The Results with 10-Time Increased Fatal Accidents

For the first analysis, the dataset was randomly separated into 10 subdatasets in a way that each contains 90% training and 10% testing set. Then, oversampling was applied to the fatal accidents in all subdatasets by copying 10-time. Then, 13 out of 24 attributes are properly discretized. And finally, the six classification methods (J48, Ibk, Random Forest, Naïve Bayes, SMO, OneR) are applied 10-fold in WEKA environment. The mean results are summarized in Table 4.1.

Table 4.1. The mean results of classification methods on the 10-time increased dataset

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	99.6	100	97.9	100	99.4	99.8
		Fatal	99.8	100	80.5	100	7.6	3.9
		Total	99.6	100	96.3	100	91.1	91.1
	F-Measure	Non-fatal	0.997	1	0.980	1	0.956	0.953
		Fatal	0.978	0.999	0.801	0.999	0.134	0.074
		Macro-Average	0.988	0.999	0.890	0.999	0.545	0.514
TESTING	Accuracy (%)	Non-fatal	98.3	99.9	97.6	99.0	99.4	99.7
		Fatal	3.2	0.8	12.6	0.4	3.6	3.6
		Total	97.4	98.9	96.8	98.1	98.5	98.8
	F-Measure	Non-fatal	0.986	0.994	0.984	0.990	0.992	0.994
		Fatal	0.025	0.014	0.073	0.003	0.046	0.056
		Macro-Average	0.506	0.504	0.528	0.497	0.519	0.525

The results indicate that despite their high accuracy rates with training set, all methods performed badly in the classification of fatal instances in the test set. OneR, the simplest methods of all, provided the highest score for fatal instances by 12.6% accuracy. And, Ibk and Random Forest produces the lowest scores with slightly over 0%. The methods are quite inefficient and increasing the number of fatal instances by 10-time did not suffice.

4.1.2. The Results of the Current Study with 50-Time Increased Fatal Accidents

For the second analysis in this study, the same steps are followed. The dataset was randomly separated into 10 subdatasets in a way that each contains 90% training and 10% testing set, and all fatal accidents were copied 50-time, thus their number is increased to 12,300 against the number of nonfatal instances (22,244). Then, the same discretization is made for 13 out of 24 attributes. And lastly, the six classification methods are applied 10-fold. The mean results are summarized in Table 4.2 below.

Table 4.2. The mean results of classification methods on the 50-time increased dataset

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	99.4	100	94.9	100	72.8	97.4
		Fatal	100	100	100	100	53.4	11.9
		Total	99.6	100	96.6	100	66.4	69.0
	F-Measure	Non-fatal	0.997	1	0.974	1	0.743	0.807
		Fatal	0.994	1	0.951	1	0.513	0.206
		Macro-Average	0.995	1	0.962	1	0.628	0.506
TESTING	Accuracy (%)	Non-fatal	98.1	99.9	94.8	99.0	72.5	97.4
		Fatal	4.9	0.8	15.0	0.4	42.5	8.5
		Total	97.2	98.9	94.0	98.1	72.2	86.9
	F-Measure	Non-fatal	0.985	0.994	0.969	0.990	0.838	0.886
		Fatal	0.0348	0.014	0.047	0.003	0.029	0.050
		Macro-Average	0.5103	0.504	0.508	0.497	0.433	0.468

According to the results, all methods performed quite well in training set; however, only Naïve Bayes showed an important increase in its classification accuracy of fatal instances (42.5%) with the testing set. This method was followed by OneR, SMO, J48, Random Forest and Ibk. In terms of F-Measures and accuracy result of fatal instances, methods did not perform well enough; after all, for binary classification, each of two target classes has already 50% chance of correct identification just like tossing coin.

4.1.3. The Results of the Current Study with 100-Time Increased Fatal Accidents

For the third analysis, after randomly separating the dataset into 10 subdatasets, all fatal accidents (246) were copied 100-time, which increases their number to 24,600 against the number of nonfatal instances (22,244). The same discretization process was applied for 13 out of 24 attributes. Subsequently, the six

classification methods are applied 10-fold, and the mean results are summarized in Table below.

Table 4.3. The mean results of classification methods on the 100-time increased dataset

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	99.3	100	94.9	100	48.5	62.5
		Fatal	100	100	100	100	76.4	69.5
		total	99.6	100	97.4	100	62.4	66
	F-Measure	Non-fatal	0.996	1	0.974	1	0.564	0.648
		Fatal	0.996	1	0.975	1	0.669	0.671
		Macro-Average	0.996	1	0.974	1	0.616	0.659
TESTING	Accuracy (%)	Non-fatal	97.9	99.8	94.8	90	48.2	62.2
		Fatal	6	0.8	15	0.4	68.6	58.9
		Total	51.9	50.3	54.9	45.2	58.4	60.5
	F-Measure	Non-fatal	0.984	0.994	0.969	0.99	0.649	0.764
		Fatal	0.039	0.015	0.047	0.003	0.025	0.029
		Macro-Average	0.511	0.504	0.508	0.496	0.337	0.396

The results of the analysis have revealed that only Naïve Bayes and SMO improved their performance in classifying the fatal instances, while the accuracy of other four methods even degraded compared to the earlier analysis. Another important point is that Naïve Bayes's accuracy for non-fatal instances decreased to some degree, and SMO performed better considering both class accuracy and F-Measures. It should be noted that after separating the dataset into 10 subdatasets, oversampling methods were applied only to training set and no further processing was made on the testing test. Therefore, the testing set has still its imbalanced structure, which explains the low F-Measure of fatal instances in the testing set.

4.1.4. The results of the current study with SMOTE Function

For the fourth and last analysis in the oversampling section of the study, SMOTE function of Imbalanced-Learn Python API was used to synthetically create additional fatal instances to even up the numbers of both target classes. So, following the application of SMOTE function, the number of fatal-instances increased to 22,244, which is the same number of non-fatal instances. Following the same discretization process of 13 out of 24 attributes, the dataset is randomly separated into 10 sub datasets in a way that each contains 90% training and 10% testing set, which are the same steps as the previous two analyses carried out in this study. Subsequently, the six classification methods are applied 10-fold, and the mean results are summarized in Table 4.4 below.

Table 4.4. The mean results of classification methods with SMOTE function

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	97.5	99.9	87.6	100	52.4	64.2
		Fatal	98.8	100	69.5	100	83.1	72.6
		Total	98.15	99.95	78.55	100	67.75	68.4
	F-Measure	Non-fatal	0.984	1	0.803	1	0.611	0.670
		Fatal	0.984	1	0.764	1	0.724	0.697
		Macro-Average	0.984	1	0.783	1	0.667	0.683
TESTING	Accuracy (%)	Non-fatal	93.7	98.4	83.5	90.4	47.8	63.5
		Fatal	9.7	4.5	22.2	12.1	56.3	50.8
		Total	51.7	51. 5	52.9	51.3	52.1	57.2
	F-Measure	Non-fatal	0.963	0.949	0.902	0.945	0.671	0.761
		Fatal	0.029	0.128	0.112	0.022	0.045	0.078
		Macro-Average	0.496	0.538	0.507	0.483	0.358	0.419

According to the results, performances of Naïve Bayes and SMO slightly deteriorated compared to the previous analysis, while other methods improved their results. Yet, the results are still quite bad and none of the methods can be seen as a good classifier.

To sum up the oversampling section of the study, the best results were obtained with the dataset containing 100-time increased fatal instances. With this dataset, SMO provided superior classification results in terms of both fatal (58.9%) and non-fatal instances (62.2%), and it was closely followed by Naïve Bayes, which produced the highest score for fatal instances (68.6%) but lower score on non-fatal instances (48.2%).

4.2. The Results with Undersampling Methods

In this part of the study, undersampling approach was embraced to overcome the problem caused by the imbalanced structure of the dataset. For this purpose, the number of fatal instances (246) was kept the same, while the number of non-fatal

instances (22,244) was randomly reduced to 5-fold (1,230), 3-fold (738) and the same number (246) of fatal instances. So, three new datasets (1:5, 1:3, 1:1) were reproduced through undersampling the nonfatal instances. In addition, Random Undersampler function of the imbalanced-learn Python API was used to produce another new dataset consisting of equal number of fatal and nonfatal instances. This function randomly reduces the majority class instances to equalize all class instances. Consequently, the six classification methods (J48, Random Forest, OneR, Ibk, Naïve Bayes, SMO) were applied to the four undersampled datasets. The results are given in the following subsections.

4.2.1. The Results of the Current Study with 1-5 (Fatal-Nonfatal) Undersampled Dataset

For the first analysis of the undersampling section, all fatal accidents (246) were kept the same and not changed, while the number of nonfatal instances was randomly reduced to 5-fold (1,230) of the fatal class. Random selection was made in MS Excel using the built-int Rand function. Then, as before, 13 out of 24 attributes were properly discretized. The final dataset was randomly separated into 10 sub datasets in a way that each contains 90% training and 10% testing set. And finally, the six classification methods (J48, Ibk, Random Forest, Naïve Bayes, SMO, OneR) were applied 10-fold in WEKA environment. The mean results are summarized in Table 4.5 below.

Table 4.5. The mean results of the classification methods with 1-5 (Fatal-Nonfatal) dataset

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	99.9	100	99.8	100	96.8	99.8
		Fatal	4.0	99.6	4.2	99.6	18.6	5.5
		Total	83.9	99.92	83.9	99.92	84.9	75.8
	F-Measure	Non-fatal	0.912	0.9	0.920	1	0.915	0.832
		Fatal	0.077	0.108	0.080	0.108	0.251	0.105
		Macro-Average	0.494	0.889	0.500	5.939	0.583	0.468
TESTING	Accuracy (%)	Non-fatal	99.6	99.1	99.6	85.2	96.5	90.4
		Fatal	3.6	2.4	3.2	21.2	14.2	13.5
		Total	98.7	98.2	98.6	84.6	95.7	88.8
	F-Measure	Non-fatal	0.994	0.990	0.993	0.916	0.978	0.904
		Fatal	0.063	0.026	0.050	0.026	0.062	0.0542
		Macro-Average	0.528	0.508	0.521	0.471	0.520	0.479

The results indicated that all methods performed poorly in classifying fatal instances, while they were quite good at nonfatal instances, which was an expected result considering the still high proportion of the nonfatal instances in the dataset. On the other hand, Ibk produced highest score with 21.2% and it was followed by Naïve Bayes and SMO.

4.2.2. The results of the current study with 1-3 (Fatal-Nonfatal) Undersampled Dataset

For the second analysis of the undersampling section, all fatal accidents (246) were kept the same and the number of nonfatal instances was randomly reduced to 3-fold (738) of the fatal class. Random selection was made in MS Excel using the built-int Rand function. Then, 13 out of 24 attributes were properly discretized. The final dataset was randomly separated into 10 sub datasets in a way that each contains 90% training and 10% testing set. And finally, the six

classification methods (J48, Ibk, Random Forest, Naïve Bayes, SMO, OneR) were applied 10-fold in WEKA environment. The mean results are summarized in Table 4.6 below.

Table 4.6. The mean results of the classification methods with 1-3 (Fatal-Nonfatal) dataset

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	94.4	100	96.4	99.4	91.7	99.7
		Fatal	28.0	100	19.4	90.4	28.4	6.5
		Total	78.4	100	77.2	99.3	75.8	78.4
	F-Measure	Non-fatal	0.879	1	0.863	0.996	0.851	0.863
		Fatal	0.314	1	0.322	0.901	0.369	0.119
		Macro-Average	0.596	1	0.593	0.948	0.610	0.491
TESTING	Accuracy (%)	Non-fatal	95.9	96.9	92.0	79.8	90.7	99.3
		Fatal	14.1	8.5	7.3	29.3	20.2	4.5
		Total	94.4	96.1	91.2	79.3	90.1	98.3
	F-Measure	Non-fatal	0.971	0.980	0.953	0.883	0.947	0.991
		Fatal	0.050	0.038	0.015	0.036	0.039	0.052
		Macro-Average	0.510	0.509	0.484	0.459	0.493	0.522

According to the results, the methods provided poor classification results once again. Ibk produced highest score for fatal instances with 29.3%; however, its classification score on nonfatal instances was the lowest among the methods. Naïve Bayes provided best result considering both classes, but its score on fatal instances was still bad.

4.2.3. The RESULTS of the Current Study with 1-1 (Fatal-Nonfatal) Undersampled Dataset

For the third analysis of the undersampling section, fatal accidents were kept the same once again, and the number of nonfatal instances was randomly reduced to

the same amount (246) as the fatal class (246). Random selection was similarly made in MS Excel using the built-int Rand function, and the same discretization and separation into 10-fold subdatasets were repeated for each to contain 90% training and 10% testing set. And finally, the six classification methods (J48, Ibk, Random Forest, Naïve Bayes, SMO, OneR) were applied 10-fold in WEKA environment. The mean results are summarized in Table 4.7 below.

Table 4.7. The mean results of the classification methods with 1-1 (Fatal-Nonfatal) dataset

			J48	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	83.8	100	72.3	100	60.4	56.5
		Fatal	76.4	100	64.9	100	71.1	71.2
		Total	80.1	100	68.6	100	65.7	66.8
	F-Measure	Non-fatal	0.808	1	0.697	1	0.637	0.652
		Fatal	0.792	1	0.690	1	0.674	0.682
		Macro-Average	0.800	1	34.8	1	0.656	0.667
TESTING	Accuracy (%)	Non-fatal	64.4	57.4	52.4	54.1	52.7	53.8
		Fatal	49.1	56.9	44.7	57.2	66.1	69.2
		Total	56.8	57.2	48.6	55.6	59.4	61.5
	F-Measure	Non-fatal	0.752	0.727	0.684	0.699	0.699	0.685
		Fatal	0.025	0.026	0.018	0.023	0.091	0.094
		Macro-Average	0.388	0.376	0.351	0.361	0.395	0.390

The dataset prepared for this section's analysis contained equal amounts of fatal and nonfatal classes. However, the results showed that no significant classification accuracy can be achieved. The total accuracy of all methods only slightly surpasses the 50% threshold. In terms of the F-measure and accuracy scores for both classes, SMO produced the best results. And yet, it classified fatal instances with 69.2% accuracy and nonfatal instances with 53.8% accuracy. It was closely followed by Naïve Bayes.

4.2.4. The Results of the Current Study with RandomUndersampler Function

For the fourth and last analysis in the undersampling section of the study, RandomUndersampler function of Imbalanced-Learn Python API was employed to reduce the number of nonfatal instances. After applying RandomUndersampler function, the number of nonfatal instances decreased to 246, which is the same number of fatal instances. Following the same discretization process of 13 out of 24 attributes, the dataset was randomly separated into 10 sub datasets, which was the same process for all analyses. Subsequently, the six classification methods were applied 10-fold, and the mean results are summarized in Table 4.8 below.

Table 4.8. The mean results of the classification methods with RandomUndersampler function

			J4.8	Random Forest	OneR	Ibk	Naive Bayes	SMO
TRAINING	Accuracy (%)	Non-fatal	78.6	100	73.2	100	50.9	64.1
		Fatal	75.4	100	63.5	100	75.5	70.5
		Total	77.0	100	68.4	100	63.2	67.3
	F-Measure	Non-fatal	0.774	1	0.698	1	0.579	0.662
		Fatal	0.765	1	0.666	1	0.672	0.683
		Macro-Average	0.769	1	0.682	1	0.625	0.672
TESTING	Accuracy (%)	Non-fatal	60.6	57.2	53.8	54.4	44.1	55.2
		Fatal	59	57.3	41.4	50.8	73.2	66.2
		Total	59.8	57.3	47.6	52.6	58.6	60.7
	F-Measure	Non-fatal	0.753	0.651	0.694	0.702	0.609	0.709
		Fatal	0.028	0.097	0.017	0.021	0.025	0.028
		Macro-Average	0.390	0.374	0.356	0.361	0.317	0.368

According to the results, SMO was the most successful methods considering both class accuracy scores. And, it was followed by J48, Naïve Bayes, Random Forest, Ibk and OneR. None of the methods can be seen as good classifiers once again, as even the most successful classification of SMO was only around 61%.

So far, Naïve Bayes and SMO have produced comparably better classification results, which are accompanied by J48 in the last analysis. The highest classification scores were obtained with 1-1 Undersampled dataset and RandomUndersampled dataset, which are the latest two datasets. SMO is the superior classifier among the six methods in both analyses. So, SMO is chosen for further analysis in the following section. However, it is a little bit harder to choose among two datasets due to similar scores of SMO, therefore, F-Measures is taken into consideration and 1-1 Undersampled dataset is chosen.

4.3. Feature Selection with SMO and 1-1 Undersampled Dataset

In this section, SMO and 1-1 Undersampled Dataset are used to see the effects of feature selection algorithms over classification accuracy. Feature Selection Algorithms choose the best set of features that contribute to the class value. And, they allow to make classification with fewer input variables, which could improve the performance of the classifier. For this purpose, four different types of feature selection algorithms are used, including Gain Ratio with Ranker search method, Relief with Ranker search method, Information Gain with Ranker search method and Correlation Based Feature Selection (CFS) with BestFirst search method. The analyses were carried out with cross validation method implemented in WEKA, which is different from the previous analyses where the cross validation is manually applied because in the previous sections we apply oversampling in which we copied samples in the training sets. To avoid having the same examples in both training and testing sets, cross validation was done manually. This is the main cause for the higher F-Measure scores of fatal instances in the Table 4.9. SMO classifier was reapplied to give a comparison basis for the implementation with the selected features. The results on testing set are summarized in the Table 4.9 below.

Table 4.9. The classification results of SMO on 1-1 undersampled dataset with/without feature selection

	Methods		1	2	3	4	5
TESTING	Selected Features		None	3,8,9	9,3,1,8,2	13,8,3,9,11	9,2
	Accuracy (%)	Non-fatal	54.1	52.8	50.8	52.8	50.8
		Fatal	58.1	69.9	66.7	69.9	70.3
		Total	56.1	61.4	58.7	61.4	60.6
	F-Measure	Non-fatal	0.552	0.578	0.552	0.578	0.563
		Fatal	0.570	0.644	0.618	0.644	0.641
		Macro-Average	0.561	0.611	0.585	0.611	0.602

1: SMO without Feature Selection, 2: SMO with CfsSubsetEval + BestFirst, 3: SMO with InfoGain + Ranker, 4: SMO with GainRatio + Ranker, 5: SMO with Relief + Ranker

All feature selection implementations positively affected and improved the accuracy scores of SMO classifier. CfsSubsetEval produced three features (3: Location, 8: Crossing, 9: Traffic Control). Information Gain gave five features (9: Traffic Control, 3: Location, 1: Day of Week, 8: Crossing, 2: Crash Time Period). GainRatio suggested five features (13: Deterrent Object, 8: Crossing, 3: Location, 9: Traffic Control, 11: Shoulder). And lastly, Relief suggested two features (9: Traffic Control, 2: Crash Time Period). Accordingly, the highest scores were obtained with SMO with CfsSubsetEval and SMO with GainRatio, both of which attained the same level of accuracy and F-measures. However, CfsSubsetEval produced its results with fewer features.

4.4. Cost Sensitive Analysis with Bagging and Adaboost

In this section, cost sensitive analysis is carried out with Bagging method and different bag sizes and cost scenarios are tried. Firstly, default 1:1 cost rate is considered. The results are given in Table 4.10 below.

Table 4.10. The classification results of cost sensitive analysis with Bagging

	Bag Size		100	90	80	70	60
TESTING	Accuracy (%)	Non-fatal	54.9	53.7	56.5	55.3	53.7
		Fatal	54.9	54.9	57.3	60.2	56.9
		Total	54.9	54.3	56.9	57.7	55.3
	F-Measure	Non-fatal	0.549	0.54	0.567	0.567	0.545
		Fatal	0.549	0.545	0.571	0.587	0.56
		Macro-Average	0.549	0.543	0.569	0.577	0.553

The initial results indicated that 70 Bag Size yielded better accuracy in the classification. Then, the analysis was repeated for 70 Bag Size with three input variables (3, 8, 9) which were selected by CfsSubsetEval and Best Search method in the previous section. In addition, two different cost scenarios were tried to see the changes in classification accuracy. The results are given in Table 4.11 below.

Table 4.11. The classification results of cost sensitive analysis with different cost matrices

	Cost Matrix		0	1	0	1.5
			1.5	0	1	0
TESTING	Accuracy (%)	Non-fatal	32.5		93.5	
		Fatal	82.5		11.8	
		Total	57.5		52.6	
	F-Measure	Non-fatal	0.434		0.664	
		Fatal	0.66		0.199	
		Macro-Average	0.547		0.432	

Increasing cost matrix for fatal instances improved the class accuracy to 82.5%, however, it reduced the accuracy of nonfatal instances. The similar case occurred for nonfatal instances, as well. Even slight changes in cost matrix resulted in high bias for one class. In the final step, we applied Cost Sensitive Analysis with

Adaboost using full dataset and the selected three input variables, as well. The results are as follows.

Table 4.12. The classification results of cost sensitive analysis with Adaboost on three selected features

	Feature Selection		None	3,8,9
TESTING	Accuracy (%)	Non-fatal	51.2	51.6
		Fatal	67.1	68.7
		Total	59.1	60.2
	F-Measure	Non-fatal	0.556	0.564
		Fatal	0.621	0.633
		Macro-Average	0.589	0.599

Cost sensitive analysis produced better results when applied with Adaboost. And feature selection further improved its accuracy. However, the classifier performance is around 60%, which is quite low for binary classification problems.

5. CONCLUSIONS

The main purpose of the study is to determine how different preprocessing methods affect the prediction accuracy of classification algorithms. For this purpose, a dataset containing traffic accidents and weather information between 2005-2015 for Adana-Turkey is used. The peculiarity of this dataset is that it has a quite imbalanced structure that contains a total of 25,015 accidents, of which only 246 accidents are fatal and the rest non-fatal. In order to solve the imbalance problem, two different approaches are used, which are undersampling the majority class (nonfatal instances in this case) and oversampling the minority class (fatal instances) along with other preprocessing methods such as data discretization, feature selection and cost sensitive analysis.

In the oversampling section, the number of fatal instances is gradually increased by copying them 10-time, 50-time and 100-time, while nonfatal instances are kept the same. Also, SMOTE function of Imbalance Learn Python API is used in Jupyter Notebook environment to obtain equal amounts of classes. The SMOTE function is widely used for oversampling the minority cases. In oversampling section, SMO attained the highest accuracy scores with the 100-time increased fatal instances, which are 58.9% fatal accuracy, 62.2% nonfatal accuracy and 60.5% total accuracy.

In the undersampling section, the number of nonfatal instances is gradually reduced to 5-fold (5:1), 3-fold (3:1) and 1-fold (1:1) of the fatal instances and the number of fatal instances is kept the same this time. Also, Random Undersampler function of the Imbalance Learn Python API is used to produce another dataset consisting of equal amounts of both classes. After the implementation of six classification methods (J48, Random Forest, Ibk, OneR, Naïve Bayes, SMO), the results demonstrated that SMO attained the best scores with 1:1 Undersampled dataset with 69.2% fatal accuracy, 53.8% nonfatal accuracy and 61.5 % total

accuracy. However, even these best scores are not good enough considering the nature of binary classification.

SMO and 1:1 dataset are chosen for further analyses. First, four different feature selection algorithms are run on 1:1 dataset. And the SMO is re-applied with the selected features. Accordingly, the highest scores are obtained with three features (3: Location, 8: Crossing, 9: Traffic Control) selected by CfsSubsetEval and Best Search method as 69.9% fatal accuracy, 52.8% nonfatal accuracy and 61.4% total accuracy, which are not much different from the ones attained with full dataset in the earlier section.

In the last step, Cost Sensitive Analysis is applied with Bagging and Adaboost Methods on 1:1 Undersampled dataset using two different cost matrices. Also, the analyses are repeated with the three features selected above. And the results are compared. As a result, it is determined that even slight changes in cost matrices result in high bias and improves the accuracy of one class, while it deteriorates the accuracy of other class. In this section, the highest scores are attained using Adaboost with the three selected features and default cost matrix as 68.7% fatal accuracy, 51.6% nonfatal accuracy and 60.2% total accuracy.

In conclusion, SMO has produced better results in each scenarios especially it attained very similar accuracy rates with 100-time oversampled, SMOTE, 1:1 undersampled and Random Undersampled datasets, while its highest are recorded with its application on 1:1 undersampled dataset. However, even these scores are slightly over 60%. Therefore, it can be said that the preprocessing methods applied in the study have only limited effects on the classification accuracy and the desired level of accuracy could not be attained. The most probable reason behind this is the fact that the dataset used in the study lacks many important parameters for the occurrence of traffic accidents and the current form of the accident reports do not represent the accidents enough to derive pattern. Especially, information about drivers involved in the accidents should be included in the accident reports. Age, education, gender, profession, income level, wearing glasses and health status of the

drivers can be considered for this purpose. Even the reason they might state to cause accident can be noted. Another important source of information is vehicles. Date of manufacture, registry date, repair history, type of vehicle e.g. station wagon, cabriolet, etc. and many more parameters can be considered for the renewal of the accident reports. In conclusion, it is highly recommended for the authorities in charge to reorganize accident reports to give more detailed information about the occurrence of the accidents.

REFERENCES

Ajmani, S., and Jadhav, K., Kulkarni, S., A., 2006. Three-Dimensional QSAR Using the k-Nearest Neighbor Method and Its Interpretation, J. Chem. Inf. Model. 46 (2006) 24–31. doi:10.1021/CI0501286.

Bernhard, B. E., Guyon, I. M., Vapnik, V. N. (1992). "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on Computational learning theory – COLT '92. p. 144. doi:10.1145/130385.130401. ISBN 089791497X.

Bishop, C., M., 2006. Pattern Recognition and Machine Learning. Springer-Verlag Berlin, Heidelberg 2006 ISBN:0387310738

Breiman, L., 1996. Bagging Predictors. Machine Learning. 24(2): 123-140. https://link.springer.com/article/10.1007%2FBF00058655 (accessed Junei 23 2018).

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., 1984. Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Bronshtein, A., 2017. "A Quick Introduction to K-Nearest Neighbors Algorithm" https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7 (accessed: June, 23 2018).

Castro, Y., and Kim, Y., J., 2016. Data mining on road safety: factor assessment on vehicle accidents using classification models, Int. J. Crashworthiness. 21 (2016) 104–111. doi:10.1080/13588265.2015.1122278.

Chang, L., Y., and Wang, H., W., 2006. Analysis of traffic injury severity: An application of non-parametric classification tree techniques, Accid. Anal. Prev. 38 (2006) 1019–1027. doi:10.1016/j.aap.2006.04.009.

Chawla, V., N., Bowyer, K., W., Hall, L., O., and Kegelmeyer, W., P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16 (2002) 321-357.

Chen, C., Zhang, G., Qian, Z., Tarefder, R., A., and Tian, Z., 2016. Investigating driver injury severity patterns in rollover crashes using support vector machine models, Accid. Anal. Prev. 90 (2016) 128–139. doi:10.1016/j.aap.2016.02.011.

Chen, C., Zhang, G., Yang, J., and Milton, J., C., 2016. An explanatory analysis of driver injury severity in rear-end crashes using a decision table/Naïve Bayes (DTNB) hybrid classifier, Accid. Anal. Prev. 90 (2016) 95–107. doi:10.1016/j.aap.2016.02.002.

Chong, M. Abrahan, A., Papprzycki, M., 2005. Traffic Accident Analysis Using Machine Learning Paradigms. Informatica 29 (2005) 89-98.

Cover, T. M., Hart, P. E., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory, IT-13(1):21–27, 1967. https://ieeexplore.ieee.org/document/1053964/ (accessed June, 24 2018).

Dadashova, B., Ramierz, B., A., McWilliams, J., M., and Izquierdo, F., A., 2016. The identification of patterns of interurban road accident frequency and severity using road geometry and traffic indicators. Transportation Research Procedia 14 (2016) 4122 – 4129

Deng, K. 1998. Online memory based general purpose system classifier. PhD Thesis. The Robotics Institute School of Computer Science Carnegie Mellon University.

Freund, Y., Schapire, R. E., 1999. A Short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780. https://cseweb.ucsd.edu/~yfreund/papers/IntroToBoosting.pdf (accessed June, 24 2018)

Holte, C., R., 1993. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. University of Ottawa. https://www.mlpack.org/papers/ds.pdf (accessed June, 24 2018)

Kam, H, T., 1995. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

Kaplan, S., and Prato, C., G., 2012. Risk factors associated with bus accident severity in the United States: A generalized ordered logit model, J. Safety Res. 43 (2012) 171–180. doi:10.1016/j.jsr.2012.05.003.

Karegowda, A., G., Manjunath, A., S., Jayaram, M., A., 2010. Comparative Study of Attribute Selection using Gain Ratio and Correlation Based Feature Selection. International Journal of Information Technology and Knowledge Managemenet July-December 2010, 2, 2, 271-277.

Kim, J., K., Ulfarsson, G., F., Kim, S., and Shankar, V., N., 2013. Driver-injury severity in single-vehicle crashes in California: A mixed logit analysis of heterogeneity due to age and gender, Accid. Anal. Prev. 50 (2013) 1073–1081. doi:10.1016/j.aap.2012.08.011.

Kunt, M., M., Aghayan, I., and Noii, N., 2011. Prediction for traffic accident severity: comparing the artificial neural network, genetic algorithm, combined genetic algorithm and pattern search methods, Transport. 26 (2011) 353–366. doi:10.3846/16484142.2011.635465.

Li, Z., Liu, P., Wang, W., and Xu, C., 2012. Using support vector machine models for crash injury severity analysis, Accid. Anal. Prev. 45 (2012) 478–486. doi:10.1016/j.aap.2011.08.016.

Li, D., Ranjitkar, P., Zhao, Y., Yi, H., and Rashidi, S., 2016. Analyzing pedestrian crash injury severity under different weather conditions, Traffic Inj. Prev. (2016) 1–4. doi:10.1080/15389588.2016.1207762.

Ling, C., X., Sheng, V., S., 2008. Cost-Sensitive Learning and the Class Imbalance Problem, *Encyclopedia of Machine Learning*. C. Sammut (Ed.). Springer."

MATLAB. "fitcknn, Fit k-nearest neighbor classifier" https://www.mathworks.com/help/stats/fitcknn.html (accessed January 9, 2017).

MATLAB. "fitcnb, Train multiclass naive Bayes model" https://www.mathworks.com/help/stats/fitcnb.html (accessed January 9, 2017).

MATLAB. "fitctree, Decision Trees - MATLAB & Simulink", https://www.mathworks.com/help/stats/classification-trees-and-regression-trees.html (accessed January 9, 2017).

Mitchell, T., 1997. Machine Learning. McGraw-Hill, Inc. New York, NY, USA ISBN:0070428077 9780070428072

Novakovic, J., Strbac, P., Bulatovic, D., 2009. Toward optimal feature selection using ranking methods and classification algorithms. Yugoslav Journal of Operations Research 21 (2011), 1, 119-135. DOI: 10.2298/YJOR1101119N http://elib.mi.sanu.ac.rs/files/journals/yjor/41/yujorn41p119-135.pdf (accessed June, 24, 2018).

Osman, M., Paleti, R., Mishra, S., and Golias, M., M., 2016. Analysis of Injury Severity of Large Truck Crashes in Work Zones, Accid. Anal. Prev. 97 (2016) 261–273. doi:10.1016/j.aap.2016.10.020.

Ozden, C., ACI, Ç., İ., 2018. Predicting the Severity of Motor Vehicle Accident Injuries in Adana-Turkey Using Machine Learning Methods and Detailed Meteorological Data. International Journal of Intelligent Systems and Applications in Engineering, 6(1), 72-79.

Prato, C., G., Gitelman, V., and Bekhor, S., 2011. Pattern Recognition and Classification of Fatal Traffic Accidents in Israel: A Neural Network Approach. Journal of Transportation Safety & Security, 3:304–323, 2011

Qiu, C., Wang, C., Fang, B., and Zuo X., 2014. A Multiobjective Particle Swarm Optimization-Based Partial Classification for Accident Severity Analysis, Applied Artificial Intelligence, 28:6, 555-576

Qiu, C., Wang, C., and Zuo, X., 2013. A novel multi-objective particle swarm optimization with *K* -means based global best selection strategy, Int. J. Comput. Intell. Syst. 6 (2013) 822–835. doi:10.1080/18756891.2013.805584.Risch, T., Canli, T., Khokhar, A., Yang, J., Munagala, K. and Silberstein, A., 2009. Decision Tree Classification, in: Encycl. Database Syst., Springer US, Boston, MA, 2009: pp. 765–769.

doi:10.1007/978-0-387-39940-9_554.

Quinlan, J., R., 1986. Induction of decision trees, Machine Learning 1, 81, 106. Reprinted in Shavlik and Dietterich eds., Readings in Machine Learning.

Quinlan, J., R., 1993. C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, California.

Xie, Y., Zhao, K., and Huynh, N., 2012. Analysis of driver injury severity in rural single-vehicle crashes, Accid. Anal. Prev. 47 (2012) 36–44. doi:10.1016/j.aap.2011.12.012.

Rokach, L. "Classification and Regression Tree Lecture Notes-Chapter 9", http://www.ise.bgu.ac.il/faculty/liorr/hbchap9.pdf (Erişim Tarihi: 19 Kasım 2016).

Russell, S., *Norvig*, P., 2003. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

Saltelli, A. 2002. Sensitivity Analysis for Importance Assessment. Risk Analysis. **22** (3): 1–12. doi:10.1111/0272-4332.00040.

Sayın, E., K., 2013. Importance of Data Preprocessing For Improving Classification Performance on CAC Data Set. Kadir Has University. Master of Science Thesis, İstanbul, 46s.

Sayad, S., "Lecture Notes", http://www.saedsayad.com/ (Erişim Tarihi: 17 Eylül 2016).

Sayad, S., "One R Algorithm", http://www.saedsayad.com/oner.htm (Erişim Tarihi: 05 Mayıs 2018).

Shon, Y., S., and Shin, H., 2010. Pattern recognition for road traffic accident severity in Korea. Ergonomics, 44:1, 107-117

Sun, J., and Sun, J., 2015. A Dynamic Bayesian Network Model For Real Time Crash Prediction Using Traffic Speed Condition Data. Transportation Research Part C 54 (2015) 176-186.

Taamneh, M., Alkheder, S., and Taamneh, S., 2016. Data-mining techniques for traffic accident modeling and prediction in the United Arab Emirates, J.

Transp. Saf. Secur. (2016) 1–21. doi:10.1080/19439962.2016.1152338.

Taamneh, M., Taamneh, S., and Alkheder, S., 2016. Clustering-based classification of road traffic accidents using hierarchical clustering and artificial neural networks, Int. J. Inj. Contr. Saf. Promot. (2016) 1–8. doi:10.1080/17457300.2016.1224902.

Tavakoli, K., A., and Shariat-Mohaymany, A., and Ranjbari, A., 2011. A Data Mining Approach to Identify Key Factors of Traffic Injury Severity, PROMET - Traffic&Transportation. 23 (2011) 11–17. doi:10.7307/ptt.v23i1.144.

TSI, Turkish Statistical Institute. Transportation Statistics http://www.turkstat.gov.tr/PreTablo.do?alt_id=10514

Tüysüzoğlu, G. 2016. Sparse coding based ensemble classifiers combined with acive learning framework for data classification. Msc Thesis. Graduate School of Science Engineering and Technology, Istanbul Techical University.

WEKA, 2013. WEKA Manual for Version 3-7-8. The University of Waikato, Hamilton, New Zealand. http://statweb.stanford.edu/~lpekelis/13_datafest_cart/WekaManual-3-7-8.pdf (accessed: June, 24 2018).

WHO, 2015. "Global Status Report on Road Safety 2015", http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/ (Erişim Tarihi: 21 Kasım 2016)

WHO, 2016. "Road Traffic Injuries - Fact Sheet", http://www.who.int/mediacentre/factsheets/fs358/en/ (Erişim Tarihi:02 Ekim 2016)

Wu, Q., Zhang, G., Zhu, X., Liu, X., C., and Tarefder, R., 2016. Analysis of driver injury severity in single-vehicle crashes on rural and urban roadways, Accid. Anal. Prev. 94 (2016) 35–45. doi:10.1016/j.aap.2016.03.026.

Yıldırım, A. 2010. Analysis and Classification of spelling paradigm eeg data and an attempt for optimization of channels used. Msc Thesis. Graduate School of

Natural and Applied sciences of middle east technical university.

Zeng, Q., and Huang, H., 2014. A stable and optimized neural network model for crash injury severity prediction, Accid. Anal. Prev. 73 (2014) 351–358.

CURRICULUM VITAE

Cevher ÖZDEN was born in İzmir in 1980. After completing elementary education in Tarsus/Mersin, he graduated from Anatolian High School of Meteorology in Ankara and was appointed to Agri Airport as Meteorologist. He was then appointed to İncirlik Airbase in Adana in 2000 and graduated from the Department of Agronomics, Faculty of Agriculture, University of Cukurova in 2004. He received MSc degree in 2008 and PhD degree in 2015 from the same department. He also gradutated from the Department of Computer Engineering, University of Cukurova in 2013, and started to pursue MSc degree in Computer Engineering in the same year. He has been working in Turkish State Meteorological Service for 19 years, serving in different provinces and positions, including the Director position of External Relations. He is currently stationed in Antalya and attends the Faculty of Law in Akdeniz University. He is fluent in English (YDS 96.25) and French (YDS 87.50) and has working knowledge of German (YDS 71.25) and Spanish (YDS 61.25). He is married and the father of one boy.