



MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES



MULTI-CLASS CATEGORIZATION OF
USER-GENERATED CONTENT IN A
DOMAIN SPECIFIC MEDIUM: INFERRING
PRODUCT SPECIFICATIONS FROM E-
COMMERCE MARKETPLACES

KEMAL TOPRAK UÇAR
(524116001)

MASTER THESIS
Department of Computer Engineering

Thesis Supervisor
Assoc. Prof. Dr. MUSTAFA BORAHAN TUMER

Thesis Co-Advisor
MUSTAFA KIRAÇ, PhD

ISTANBUL, 2019



MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES



MULTI-CLASS CATEGORIZATION OF
USER-GENERATED CONTENT IN A
DOMAIN SPECIFIC MEDIUM: INFERRING
PRODUCT SPECIFICATIONS FROM E-
COMMERCE MARKETPLACES

KEMAL TOPRAK UÇAR
(524116001)

MASTER THESIS
Department of Computer Engineering

Thesis Supervisor
Assoc. Prof. Dr. MUSTAFA BORAHAN TUMER


Thesis Co-Advisor
MUSTAFA KIRAÇ, PhD


ISTANBUL, 2019


MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

Kemal Toprak Uçar, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended her thesis entitled “**Multi-Class Categorization of User-Generated Content in a Domain Specific Medium: Inferring Product Specifications from E-Commerce Marketplaces**”, on 19.07.2019 and has been found to be satisfactory by the jury members.

Jury Members

Assoc. Prof. M. Borahan TÜMER (Advisor)
Marmara University, Department of Computer Engineering 

Prof. Çiğdem EROĞLU ERDEM (Jury Member)
Marmara University, Department of Computer Engineering 

Prof. Tunga Güngör (Jury Member)
Boğaziçi University, Department of Computer Engineering 

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Kemal Toprak Uçar be granted the degree of Master of Science in Department of Computer Engineering, on 07.08.19...
(Resolution no: 2019/16-Ş 2)

Director of the Institute
Prof. Dr. Bülent EKİCİ




ACKNOWLEDGEMENTS

Throughout this work, it has been an honour and pleasure to work with my advisors Assoc. Prof. Borahan Tümer and Mustafa Kır aç, PhD because of their encouragements, supports, and trusts. During the thesis, they enhanced not only my technical knowledge but also my approach to a novel research area. Besides their advisor identities, they are good friends of mine.

I would like to express my appreciation to my family, they have never gotten bored to listen to my problems for 28 years, my mother and father Filiz and Őinasi U AR, my brother, my idol. My brother has always trusted me as I have always trusted him too, and her wife, one of the mentors of mine for my most complex problems, Ali Egemen and Medina U AR, and my dearest nephew, the baby who stares at me from my phone screen, Almira U AR. Besides my family, I would like to thank to my grandmother, Bahriye Tartılacı for her strong attachment.

I would like to thank to Senem AYDIN, my wife. In these 3 years, we could have a break for our bucketlist, but I assure you we will keep on once I get my degree, I promise. Your experience from your master has always been a plus for a degree. When I acquired 0.67 F-score from my first experience, you encouraged me with your kind words then I have not given up working. To end up my project with successful results, I owe you a lot. I believe our life will be the same, we will never give up reaching happiness. Besides her, I would love to thank her family, Rahmiye, Sami, and Atakan AYDIN for their joining my excitement and sharing with me their academic advices.

I would like to thank all my cheerful friends from Denizli Anadolu Lisesi, Marmara University, and Paris, Fluctuat Nec Mergitur, for their endless support. What I learnt from my 28 years is that life does not consist of work. There is also a life after 6 p.m. and weekends. I owe them my joviality and life energy. I am sure that my accomplishments will show up in time, but happiness is never attainable without the people around me.

Lastly, I would like to tell my gratitude towards Marmara University Institute of Science and Technology to support my participation to INFUS (Intelligent and Fuzzy Systems) and iyzico for their supports on my degree.

May 2019

Kemal Toprak U ar

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
TABLE OF CONTENTS	ii
ABSTRACT	iv
ÖZET	v
ABBREVIATIONS.....	vi
LIST OF FIGURES.....	vii
LIST OF TABLES.....	viii
1. INTRODUCTION	1
1.1. General Background	1
1.2. Scope and Objective of the Study	3
2. RELATED WORK.....	6
3. MATERIAL AND METHOD	8
3.1. Data Preprocessing	9
3.1.1. HTML Tag Removal	9
3.1.2. Lowercasing	9
3.1.3. Punctuation and Number Removal.....	9
3.1.4. Stop Word Removal	10
3.1.5. Stemming and Lemmatization	10
3.2. Feature Extraction.....	10
3.2.1. Text Vectorization	11
3.2.2. Term Frequency – Inverse Document Frequency (TF-IDF)	15
3.2.2.1 Separation of Product’s Title and Description	15
3.3. Classification	18

3.3.1. Cross-Validation	20
3.3.2. Parameters	21
4. EXPERIMENTS AND RESULTS	23
4.1. Setup	23
4.1.1. Data Collection.....	23
4.2. Results and Discussion	25
4.2.1. Error Analysis.....	37
5. CONCLUSION AND FUTURE WORK.....	39
APPENDIX A.....	41
APPENDIX B.....	43
APPENDIX C.....	44
APPENDIX D.....	46
6. REFERENCES	47
7. CURRICULUM VITAE	50

ABSTRACT

MULTI-CLASS CATEGORIZATION OF USER-GENERATED CONTENT IN A DOMAIN SPECIFIC MEDIUM: INFERRING PRODUCT SPECIFICATIONS FROM E-COMMERCE MARKETPLACES

A "marketplace" is an e-commerce medium where product and inventory information is provided by varying third parties, whereas catalog service is hosted, and payments are processed by the marketplace operator. As a result of the increasing use of marketplaces, e-commerce capabilities can now be accessed by everyone. Consequently, both the number of merchants and products have been growing exponentially. Such growth raises some problems including "Does product description reflect specifications of the real one?", "Does the seller really own the product?", "Is this product legal for purchasing online?", "Is this product listed under correct category?". These problems can lead to penalties or complete close-down of the merchant as e-commerce business is regulated in most countries.

We propose a methodology to detect an accurate product category from user-generated content on e-commerce marketplaces, so that proactive removal of certain products can be automated. We present our methodology as a complete system that incorporates data collection, cleaning, and categorization. In this work, we transform unstructured text into vector representations of words during machine-learning-ready dataset preparation stage. We train ML models by a large corpus of text which includes more than half a million product descriptions. Finally, we compare our results in alternate classification algorithms and varying methodologies of vector representations. We showed that accurate predictions of text categories reaching 0.87 F-score can be obtained from user-generated text that may contain typos, special punctuation, and abbreviations, and comes from a non-moderated e-commerce medium.

Keywords: Machine learning; natural language processing; text classification; e-commerce

ÖZET

KULLANICI TARAFINDAN ÜRETİLEN İÇERİĞİN SINIFLANDIRILMASI: E-TİCARET PAZARYERLERİNDEN ÜRÜN SPESİFİKASYONLARINI ÇIKARMA

Pazaryeri, ürün ve envanter bilgilerinin çeşitli üçüncü taraflarca sağlandığı, katalog hizmetinin verildiği ve ödemelerin piyasa operatörü tarafından yönetildiği bir e-ticaret aracıdır. Pazaryerlerinin kullanımının artmasının bir sonucu olarak, e-ticaret olanağına şimdi herkes tarafından erişilebilmektedir. Bununla birlikte hem satıcı sayısı hem de ürün sayısı katlanarak artmıştır ve artmaktadır. Bu büyüme, “Ürün açıklaması, ürünün gerçek özelliklerini yansıtıyor mu?”, “Satıcı ürüne gerçekten sahip mi?”, “Bu ürün çevrimiçi satın almak için yasal mı?”, “Bu ürün doğru tür altında mı listeleniyor?” gibi bazı sorular sormamıza neden oluyor. Çoğu ülkede e-ticaret etkin olarak kullanıldığından, bu tür sorunlar yasal yaptırımlara veya satıcının etkinliklerinin tümüyle yasaklanmasına neden olabilir. Bu çalışmada, e-ticaret kullanıcısı tarafından oluşturulan içeriği kullanarak ürünün türünü belirleyen bir yöntem sunuyoruz, böylece belirli ürünlerin proaktif olarak kaldırılmasını otomatikleştiriyoruz. Yöntemimiz veri toplama, veri temizliği ve tür belirleme olarak üç ana işlemden oluşan bir sistem önermektedir. Bu çalışmada; yapılandırılmamış metni, yapay öğrenmeye hazır veri kümesi hazırlama aşamasında sözcüklerin vektörel temsillerine dönüştürüyoruz. Yapay zekâ modellerini yarım milyondan fazla ürün bilgisi içeren geniş bir metin yelpazesıyla eğitiyoruz. Son olarak, sonuçlarımızı farklı sınıflandırma algoritmaları ve vektör temsil yöntemleriyle karşılaştırdık. Sonuç olarak, ürün kategorilerinin kullanıcı tarafından oluşturulan, yazım hataları, özel noktalama işaretleri ve kısaltmalar içerebilen, denetlenmeyen bir e-ticaret sitesinden elde edilen bir metinden 0.87 F-score gibi performansla çıkartılabileceğini gösterdik.

Anahtar Kelimeler: yapay öğrenme; doğal dil işleme; metin sınıflandırma; e-ticaret

ABBREVIATIONS

ANN	: Artificial Neural Networks
CPU	: Central Processing Unit
CBOW	: Continuous Bag of Words
DRF	: Distributed Random Forest
GBM	: Gradient Boost Machine
IR	: Information Retrieval
KNN	: K-Nearest Neighbors
LDA	: Latent Dirichlet Allocation
NB	: Naïve Bayes
NLP	: Natural Language Processing
RAM	: Random Access Memory
SVD	: Singular Value Decomposition
SVM	: Support Vector Machines
TF-IDF	: Term Frequency – Inverse Document Frequency
TRY	: Turkish Lira

LIST OF FIGURES

Figure 1-1. Retail E-Commerce sales from emarketer	1
Figure 1-2. User-Generated Content and Structured Content.	4
Figure 3-1. Block diagram of the system.....	8
Figure 3-2. Structure of a neural network.....	13
Figure 3-3. Mechanism of Distributed Random Forest.	19
Figure 3-4. Mechanism of Gradient Boosting Machine.	20
Figure 3-5. Steps of cross-validation.	21
Figure 4-1. Structure of a Web Crawler.	23
Figure 4-2. Confusion Matrix.	25



LIST OF TABLES

Table 3.1. Common words table.	11
Table 3.2. Feature Set.	17
Table 4.1. Initial data distribution of products.....	24
Table 4.2. Results obtained with GBM.....	28
Table 4.3. Results obtained with DRF.	29
Table 4.4. Results after the separation of title and description features with GBM.	31
Table 4.5. Results after the separation of title and description features with DRF.	33
Table 4.6. Impacts of data preprocessing phases.	35
Table 4.7. Exhaustive parameter search results.	36
Table 4.8. Results obtained with merging highly correlated categories.	37

1. INTRODUCTION

1.1. General Background

Conducting business on the Internet is called “e-commerce;” in other words “electronic commerce.” Amazon, eBay, and Alibaba are some of the pioneer e-commerce companies and they accounted for 52% of global web sales in 2018 according to Online Marketplaces Database¹. According to the research of emarketer², retail e-commerce sales extended to \$1.538 trillion in 2015 and this figure is foreseen to approach to \$3.418 trillion by %84 growth.



Figure 1-1. Retail E-Commerce sales from emarketer

¹ <https://www.digitalcommerce360.com/product/online-marketplaces-database/>

² <https://www.emarketer.com/Article/Worldwide-Retail-Ecommerce-Sales-Will-Reach-1915-Trillion-This-Year/1014369>

The term marketplace has been increasing in e-commerce domain. A marketplace is an e-commerce medium where product and inventory information is provided, and payments are processed by the marketplace operator (Cotton & Liddicoat, 2005). Users can register on these websites, in order to sell their products. For attracting more customers, images and descriptions of the products should be descriptive and clear to understand. Besides catching customer's attention, online shopping platforms must have an ability to accurately and quickly find the desired products for the customers. In order to achieve this ability, online shopping platforms use product categories (Kanagal et al., 2012), also known as taxonomies, to organize products hierarchically from general to specific classes. Customers are able to search keywords and this guarantees the consistency of the taxonomy of similar products enabling product suggestion (Ziegler et al., 2014). The number of categories varying according to shopping platforms such as there are 27 main categories in Amazon.com while only 9 main categories exist in ebay.com since these company use different taxonomies.

Organization of a product taxonomy requires a great effort since each product description, title and related information are examined in depth and associated category or categories are assigned hierarchically. In small shopping platforms, this process can be handled manually by analysts. Employing manual product categorization is a laborious process. Besides the workload, different sellers might post same products under different categories. In big companies, there must be an automated mechanism to employ product categorization since hiring a great number of analysts is nearly impossible. Automated systems' execution time is less than humans and they are less costly.

Besides the accurate taxonomy, certification is inevitable for merchants to sell a certain type of product. Regulations are even more strict for online merchants since it is harder to verify the buyer's identity, location, and age. Some example products restricted for trading online are alcoholic beverages, weapons, or medicine. In some countries, selling cosmetics, jewelry, silk, cryptocurrencies, or gambling credits online is illegal. In addition to selling restrictions, law may also regulate installments. In Turkey, a customer may pay for any service under accommodation category with an installment plan if seller permits whereas it is prohibited to create an installment plan for products under food category. E-commerce merchants whose

business model serves as a marketplace need to know about all such restrictions. The main challenge here is to moderate each product listing posted by individual sub-merchants.

In addition to the master's thesis, a conference paper of the following study has been approved for the International Conference on Intelligent and Fuzzy Systems (INFUS) (Ucar et al., 2019).

1.2. Scope and Objective of the Study

In this work, the main problem we study is the automation of product categorization for marketplaces. To solve this problem, we apply Natural Language Processing (NLP) techniques to assign products to categories. Employing these techniques on the Turkish content generated by users is a novel approach for product categorization. Basically, the classification of a merchant's new products is automated by utilizing the existing description and classification of products. The same approach can also be used when a new product line not introduced before, or the products are more densely populated than the training data.

Our approach requires dividing the problem into smaller pieces. For instance, an online product listing may have multiple information pieces such as image, text, and tabular data. In this paper, we provide a novel approach to heterogeneous data mining by translating each data type into a standardized format. We create textual data, i.e., sequence of words that are not necessarily meaningful sentences, from each of the data types. Then, we utilize rule-based natural language generation techniques for handling structured data (i.e., in a way similar to financial news that is automatically generated from stock or forex numbers).

In addition to heterogeneous data mining, by text vectorization and representing categories using Term Frequency – Inverse Document Frequency (TF-IDF) approach we can handle imperfect language (due to user provided text with typos and no strict grammar applied) and combination of multiple languages (i.e., terms used to describe hardware such as hard disk and chipset are English, whereas user-generated product descriptions are in Turkish), and can allow for weighting alternate sections of the text differently. Hence, our approach is both new and significant.

Our work provides solutions to the following challenges:

User Generated Content: Words are at the core of text classification. Text provided by sellers is often unsuitable as input data for NLP and Information Retrieval (IR) tasks, since text may be carelessly entered by the seller. An example of a user-generated content and structured content is demonstrated in the Figure 1.2. Tokenization, sentence detection, common/stop word elimination are some of the important data cleaning tasks.

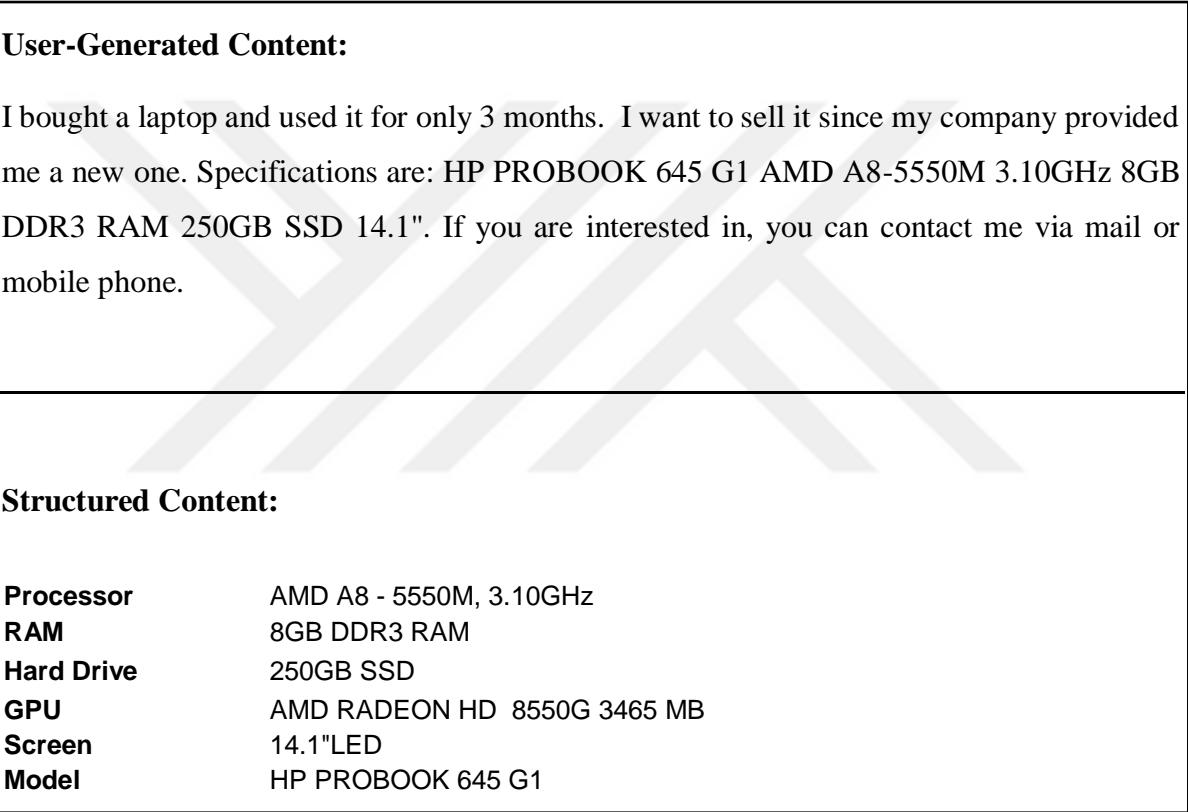


Figure 1-2. User-Generated Content and Structured Content.

Data Representation: Text data is unstructured by nature, whereas most machine learning and computational frameworks accept array-like or link-structured data. IR literature provides us with many ways of representing words as text (Ramos, 2013). Newer approaches make use of neural networks or matrix factorization techniques to provide dense numerical vectors that represent words, topics, or concepts in repository (Mikolov et al., 2013; Dumais, 2004). Word embeddings were first proposed in a neural probabilistic language model

(Bengio et al., 2003). In text classification, word embeddings formulate the word probability with the embedding probability that integrates context information.

Intermediate Model: Once (i) product description is transformed into standardized text, (ii) the standardized text is cleaned through our text quality evaluation algorithms, (iii) and the product description and product title are reduced into dense word vectors and the representatives are generated using TF-IDF. These are employed in a multi-class classification algorithm that groups product information into predefined distinct classes. Machine learning is broadly used for text categorization (Sebastiani, 2002; Joachims, 1998).

2. RELATED WORK

Machine learning is a paradigm encompassing methods that optimize a performance criterion based upon previous experience or example data (Alpaydin, 2009). There is a model which is a mathematical representation of a real-world process and this model learns from a given training data. With the growing quantity of data, machine learning applications have been rising in the academy and industry. Prognosis and prediction of cancer (Kourou et al., 2015), anomaly detection (Lane & Brodley, 1997), the YouTube video recommendation system (Davidson et al., 2010), and prediction of football results (Baio & Blangiardo, 2010) are just a few among great many machine learning applications.

NLP is a subfield of computer science that equips computers with the ability to understand and manipulate text or speech to make inference (Chowdhury, 2003). By the rapid growth of textual data since online platforms have been accessible from all areas, machine learning applications are employed for text classification. Analysis of hotel reviews (Kasper & Vela, 2011), detection of election related tweets (Yang et al., 2018), and automatic text classification based upon the author's gender (Sboev et al., 2016) are some studies where machine learning techniques are applied for text classification.

Recent classification techniques were employed to achieve better product categorization performance. A comparative study (Chavaltada et al., 2017) handles product categorization using Support Vector Machines (SVM), Naïve Bayes (NB), Artificial Neural Networks (ANN), and Logistic Regression. The results illustrate that the best accuracy was obtained using NB. A study (Ristoski et al., 2018) handles product matching in addition to product categorization for the 3 levels product taxonomy. They utilized word-embeddings in order to extract features from the text, and the Convolutional Neural Network is used to produce features from the product image for both tasks. 0.84 and 0.69 F-scores were achieved for product matching and product categorization in order. In another automatic product classifier study (Lee & Yoon, 2018), word2vec, TF-IDF, and doc2vec are employed to classify the product descriptions then they acquired 0.90 F-score for the first level of product categories. For all levels of the product taxonomy, a combination of the features from doc2vec and TF-IDF outperforms other features. The study from Stanford University (Shankar & Lin, 2011)

employs Naïve Bayes, K-Nearest Neighbors (KNN), and tree classifiers for the product categorization and the highest accuracy was acquired using the tree classifier.

In Chimera (Sun et al., 2014) the authors handle automatic product categorization by a combination of machine learning, hand-crafted rules and crowdsourcing utilizing only product's title. In addition to machine learning, researchers contributed to the classification by incorporating blacklist, whitelist, attribute and value-based classification rules to get significant results. In the study by Kozarova (Kozarova, 2015), n-grams, mutual information dictionaries, Latent Dirichlet Allocation (LDA) and word embeddings are employed using different classifiers, and they attained an F-score of up to 0.88 using word embeddings. The studies listed above apply product categorization through basic data preprocessing steps including stop word removal, stemming-lemmatization, lowercasing, and punctuation removal. In contrast, the data we dealt with came from a user-generated medium that had serious problems against properly applying stop-word elimination and stemming. On the other hand, we were able to improve accuracy of categorization by incorporating unstructured textual information in the product data, such as price and currency.

3. MATERIAL AND METHOD

The phases of the system are illustrated briefly in Figure 3.1. (Extended version can be found in Appendix D)

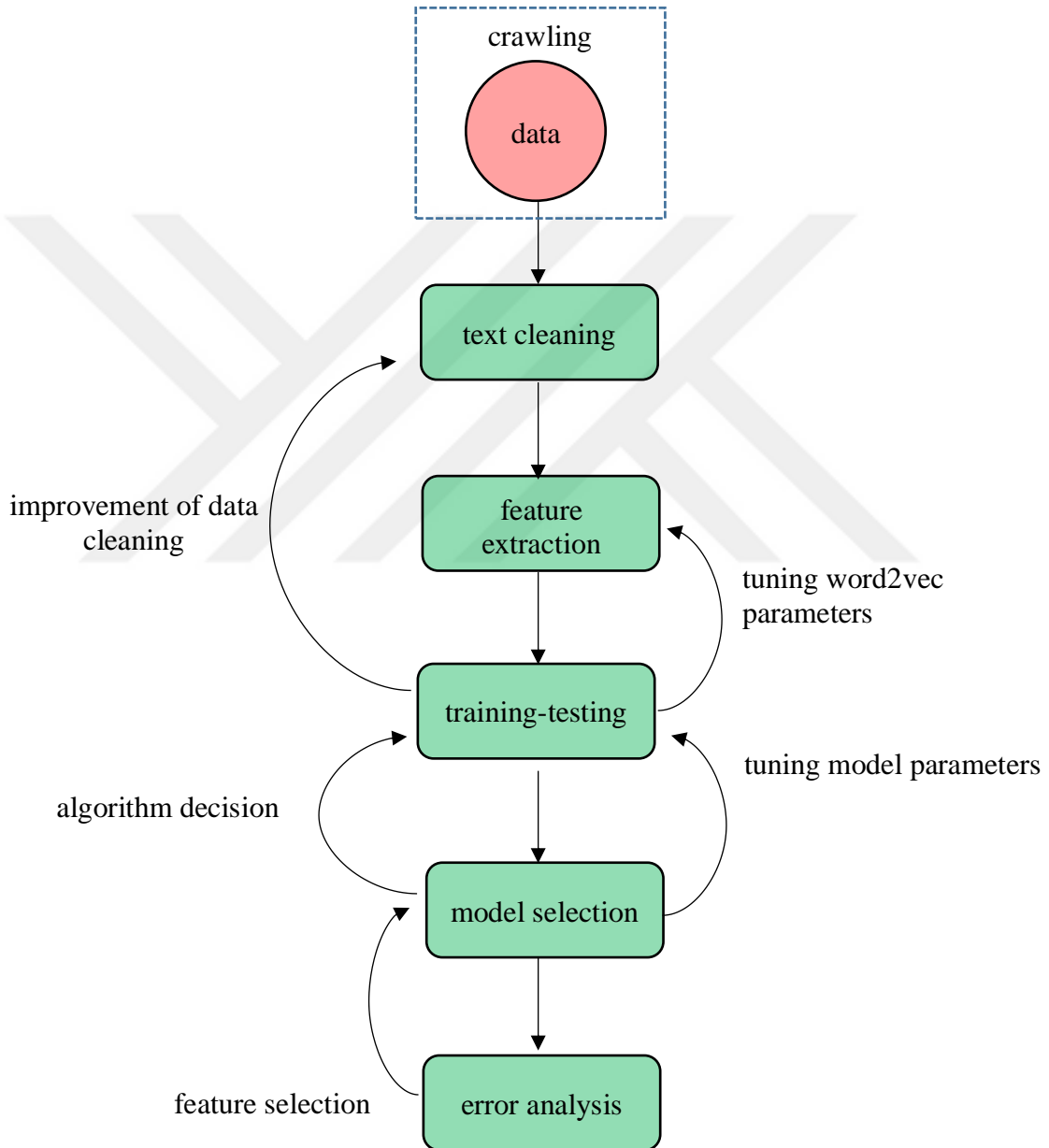


Figure 3-1. Block diagram of the system.

3.1. Data Preprocessing

Here we apply cleaning and normalization techniques to generate meaningful features for classification.

3.1.1. HTML Tag Removal

Since data is fetched using a web crawler, data obtained in HTML format exists in a crude fashion. Therefore, some control characters and HTML tags might exist in the description area. Initially these redundant values such as ‘\n’, ‘\r’, and ‘\t’ are swept from text using Regular Expression in order to start data preprocessing.

3.1.2. Lowercasing

Sentences in every language tend to capitalize the first letter of the first word. Additionally, different people capitalize different words intentionally or otherwise depending on their interpretation of the content of the word. Since the casing does not change the meaning of the word, we make all words lowercase. While lowercasing words, concerning special Turkish letters ‘ğ’, ‘ş’, ‘ü’, ‘İ’, ‘ç’, ‘ö’, and ‘ı’ we did not ignore them for lowercase and uppercase forms.

3.1.3. Punctuation and Number Removal

Words may have several forms depending on how formal it is. Further concatenated words are often used differently. (i.e., I am, antialiasing – anti-aliasing) Punctuation removal is known as one of the most effective feature normalization methods when processing on lexical data. Before the removal process, all punctuations are replaced by a space in order to prevent from concatenating unrelated words incorrectly since some people do not use space next to the punctuations. Numbers and words containing digits are also removed from text including phone numbers, quantities and other numbers with no impact on product categorization. If any word includes a number, it is also removed from the text.

3.1.4. Stop Word Removal

As common to any arbitrary text, short function words such as conjunctions, often prepositional terms and other grammatical syntax fillers may appear typically in many of product titles and descriptions. By themselves, they do not have any meaning or any impact for classification, e.g., “ve” (and), “ile” (with), “ne” (what). They were enlisted manually before the data-preprocessing step. Words common to each category are also discarded using TF-IDF. During data analysis on product text data, many common words show up in most categories and this complicates the distinction of different categories while determining words characterizing each category. Table 3.1. demonstrates some of common words which occur in product text.

As mentioned in the following section, Stemming and Lemmatization, given words were trimmed to narrow down our overall feature space.

3.1.5. Stemming and Lemmatization

Stemming and Lemmatization are techniques for text normalization in NLP. Since there exists many exceptions in the Turkish language (Can et al., 2008), we did not apply stemming or lemmatization on text. In order to narrow down our overall feature space, only the first 5 letters of each word are considered during the comparison, vector representation, or TF-IDF implementation through words. (i.e., arabadan – araba, macbook – macbo)

3.2. Feature Extraction

In this section, we merge the terms product description and title and denote them by product’s text.

Table 3.1. Common words table.

Word	Meaning	Word	Meaning
ürün	product	sadece	only
adet	quantity	lütfen	please
sıfır	zero	uygun	suitable
temiz	clean	ürünler	products
cm	cm	gün	day
tl	Turkish Lira	yok	unavailable
özel	special	yıl	year
orjinal	original	tek	unique
el	hand	takas	exchange
teslim	delivery	aynı	same
fiyat	price	bilgi	information
pazarlık	negotiation	marka	brand
kullanılmış	used	renk	color
mesaj	message	ekran	screen

3.2.1. Text Vectorization

Such models take regular text sequences as input and map each word into a numerical vector. Recently, learning distributed vector representations of words using neural attract much attention. The well celebrated Word2Vec (Mikolov et al., 2013) predicts the target word by

its neighbor words and maps words with similar meanings to nearby points in the continuous vector space. High quality word embeddings have been achieved with this simple model in application areas such as text understanding, language modeling, and machine translation. Large data sets are handled by Word2Vec thanks to the optimization of computational efficiency using Negative Sampling and Hierarchical Softmax (Rong, 2014). The model's simplicity keeps the time complexity of the training process in rather lower levels. For the use of content, Word2vec model is employed. A large corpus of text is used as an input to Word2vec to generate vector representations of words a.k.a. word embeddings. It is utilized in the problems concerning Named Entity Recognition (Nadeau & Sekine, 2007), document classification, sentiment analysis, and so on. Using only *bag of words* loses the syntactic information of words within text. Moreover, the solution we offer is also computationally feasible.

As demonstrated in Figure 3.2, structure of Word2Vec is a multi-layer neural network containing input, hidden and output layers along with their interconnections. The neural network is fully connected, and each connection has its weight value. The size of input and output layers is identical, equal to the size of vocabulary. At the input layer, one-hot vector of each word is found on the node. There is no activation function at the hidden layer, and it passes sum of weights to the output layer. At the output layer, the softmax function generates a well-defined probability distribution. The model is trained by an application of backpropagation. In each iteration, the weight values are updated regarding a loss function. Continuous Bag of Words (CBOW) and Skip-Gram are two methods supported by Word2Vec. In CBOW, a target word is predicted by taking the surrounding words into account. In opposition to CBOW, surrounding words are predicted using target word in Skip-Gram. Word analogy can be formed regarding the vector representations obtained from the Word2Vec model by applying arithmetic operations between word vectors (Rohde et al., 2006) (i.e. *king - man + woman = queen*)

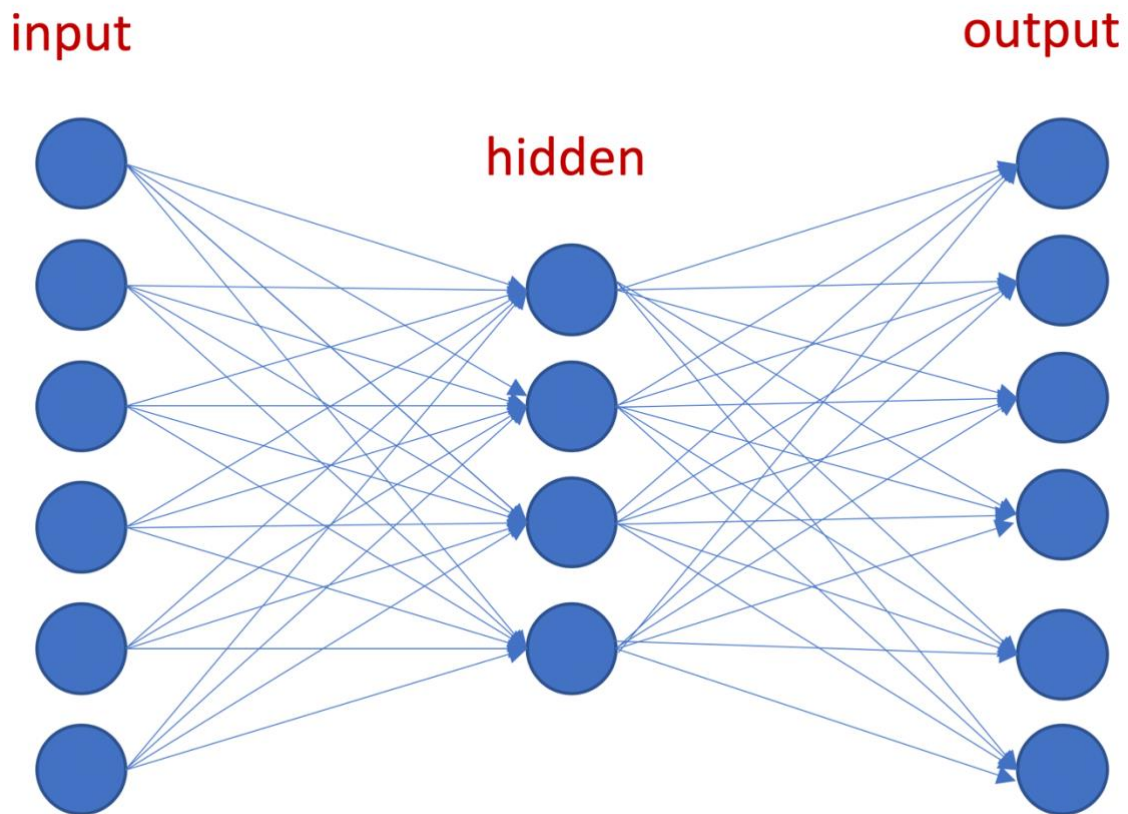


Figure 3-2. Structure of a neural network.

Products are represented by their word vectors and those displaying a higher value of the similarity criterion used in their vector entries are more likely to belong to the same category. Gensim (Rehurek & Sojka, 2010) is a pure Python library that provides similarity search, digital document indexing, and fast, scalable, and memory-efficient algorithms for Singular Value Decomposition (SVD) and LDA. Gensim realizes unsupervised semantic modelling from plain text. It outperforms other similar software in terms of robustness and efficiency. We used Gensim to generate word embeddings from the text data that we crawled. Before training, we configure algorithm parameters that are *window size*, *vector size*, *number of iterations*, *minimum count*, and *number of threads*. Window size is the maximum distance between the selected and predicted word within a sentence. Vector size is the number of dimensions of a word vector. Number of iterations is the number of epochs while model is

trained. Minimum count is the minimum number of occurrences a word to ignore. The last parameter we used is the number of threads which accelerates the model training. After Word2Vec model is initialized, the resulting embeddings are employed as a feature in the classification model to solve the product categorization problem. The vector representation of each word consists in product's text and takes the average of vectors to represent product's text and the size of vector representation is 120. The formula of vector representation of product's text is given in Equation (1) where w_i is the i^{th} word in product's text, and n is the length of product's text.

$$\vec{w} = \sum_{i=0}^n \frac{\vec{w}_i}{n} \quad (1)$$

Before generation of word embeddings, each category name is added to the end of product's text in **only training data** to increase the similarity between words that occur in categories and category names. The similarity between words is assessed by the *cosine similarity* between their vector representatives. The formula of cosine similarity is demonstrated in Equation (2) where w_1 and w_2 are word vectors.

$$similarity(w_1, w_2) = \cos(\theta) = \frac{\vec{w}_1 \cdot \vec{w}_2}{\|\vec{w}_1\| \cdot \|\vec{w}_2\|} \quad (2)$$

Similarity of each word for each category is extracted as shown in Equation (3) where c_x is the vector representation of a given category x such as “computer” or “vehicles”. This similarity measurement is executed for each category.

$$s = \sum_{i=0}^n \frac{similarity(\vec{c}_x, \vec{w}_i)}{n} \quad (3)$$

3.2.2. Term Frequency – Inverse Document Frequency (TF-IDF)

Stop words, most frequent e-commerce words, and representatives that identify each product category are extracted using TF-IDF. In Equations (4-6) we illustrate how TF-IDF is calculated.

$$TF_t = \frac{\text{(Number of times term } t \text{ appears in a document)}}{\text{(Total number of terms in the document)}} \quad (4)$$

$$IDF_t = \log_e \frac{\text{(Total number of documents)}}{\text{(Number of documents with term } t \text{ in it)}} \quad (5)$$

$$TF - IDF = TF_t \times IDF_t \quad (6)$$

For stop words and most frequent e-commerce words, only unigrams are employed. In order to determine representative words for each category, both unigrams and bigrams are obtained, and a set of words is generated for each category. In unigram, word itself is considered only while the context of each word is considered with its following pair in bigram. There might be some correlations between different categories in unigram which cause confusion in the learning algorithm, but bigrams resolve the ambiguity of words which have multiple meanings. (i.e., words such as “world” and “cup” can appear in different classes but “world cup” strongly indicates classes related with sports.) After retrieving both bigram and unigram dictionaries for each category, unigram and bigram sets of product’s text are generated and the intersection of dictionaries and their set of words are determined. Dictionary and intersection of product’s text and dictionary of each category are defined in Equation (7) where w_i is the i^{th} word in product's text, n is the number words, D_c is dictionary for a given category c , x is the size of dictionary, and I is the intersection of product’s text and dictionary. In Appendix A, some of words from unigram and bigram dictionaries are demonstrated.

$$W = \{w_0, w_1, \dots, w_n\}, \quad D_c = \{d_0, d_1, \dots, d_n\}, \quad i_c = W \cap D_c \quad (7)$$

3.2.2.1 Separation of Product’s Title and Description

During the experiments, besides the extracting dictionary features from the concatenation of product’s title and product’s description, separate features are also generated from title and

description. Similar to the implementation above, bigram and unigram dictionaries are generated for both title and description according to each category and their set of words are determined. Experiments showed that a model with the features *product title* and *description* separated outperforms one where they are combined into the *product's text*. In the results and discussion section, we indicate the effect of feature separation.

In addition to word embeddings and TF-IDF features, we also have *product's price* and *currency*. They can be distinguishing features for classification. Since product data was fetched from Turkish e-commerce platforms, majority of currency is Turkish Lira (TRY). Besides TRY, foreign currencies also exist in varied products. In order to generate a useful feature from price, all price values are converted to TRY (based on exchange rate at the time of data collection). Unless stated otherwise, there would be an ambiguity through amounts with different currencies.

In Table 3.2, features extracted from each product's text using word2vec, TF-IDF, and product information are indicated.

Table 3.2. Feature Set.

Method	Feature Set
$W_{\text{product_text}} = [v_1, v_2, \dots, v_{118}, v_{119}]$	feature ₁ , feature ₂ , feature ₃ , ... ,feature ₁₂₀
$\text{similarity}(w_{\text{category_1}}, W_{\text{product}})$	feature ₁₂₁
$\text{similarity}(w_{\text{category_2}}, W_{\text{product}})$	feature ₁₂₂
...	...
$\text{similarity}(w_{\text{category_22}}, W_{\text{product}})$	feature ₁₄₂
$i_{\text{category_1_title_unigram}}$	feature ₁₄₃
$i_{\text{category_1_title_bigram}}$	feature ₁₄₄
$i_{\text{category_1_description_unigram}}$	feature ₁₄₅
$i_{\text{category_1_description_bigram}}$	feature ₁₄₆
...	...
$i_{\text{category_22_title_unigram}}$	feature ₂₂₇
$i_{\text{category_22_title_bigram}}$	feature ₂₂₈
$i_{\text{category_22_description_unigram}}$	feature ₂₂₉
$i_{\text{category_22_description_bigram}}$	feature ₂₃₀
price	feature ₂₃₁
amount of the numbers in the text	feature ₂₃₂
amount of the words	feature ₂₃₃
length of product's text	feature ₂₃₄

3.3. Classification

After feature extraction, we evaluate two different classification algorithms which support classification with multiple labels. Distributed random forest (DRF) (Geurts et al., 2006) and Gradient Boost Machine (GBM) (Friedman, 2001) are such two ensemble models. We develop classification models using H2O (Aiello et al., 2016), a scalable machine learning framework.

When we create a machine learning model to make predictions, noise and variance are the main causes of difference in real and predicted values. Ensemble models impact to reduce these factors. The reason we use ensemble models is that different predictors trying to do the same prediction provide better performance than a single predictor alone. There are two types of ensemble models: Bagging and Boosting. In Bagging, there are several subsets from training sample which is obtained randomly with replacement. Training of their decision trees is conducted with each subset data. In consequence of training, there is an ensemble of different models and average of predictions made by these different trees are more effective than a single decision tree. In Boosting, decision trees are trained sequentially. Early trees fit simple models to the training data and examine data for errors. DRF is a bagging model which constitutes many independent predictors and combines them using model averaging techniques. Having few parameters, robustness, and performing competitive accuracy on most data sets are the advantages of DRF. In Figure 3.3 evaluation of DRF is demonstrated.

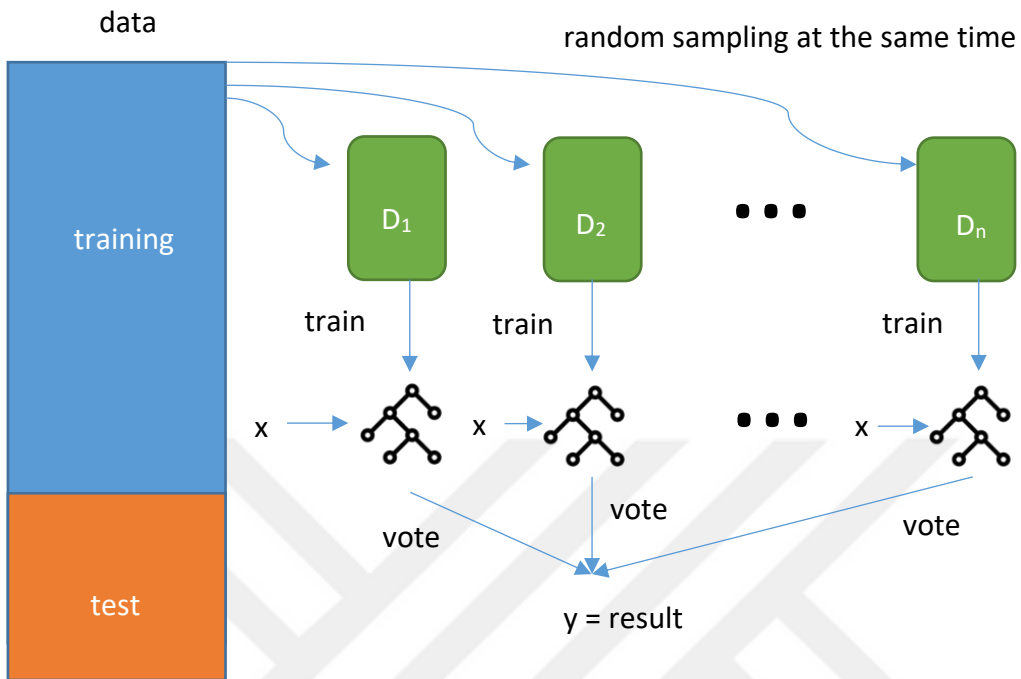


Figure 3-3. Mechanism of Distributed Random Forest.

GBM is a boosting ensemble which builds predictors sequentially (not independently) and the subsequent predictors learn from errors of the previous predictors. GBM is often a best model and directly optimizes the cost function. Besides the advantages of GBM, model may overfit, so parameters must be chosen carefully. In Figure 3.4 demonstrates how GBM is evaluated.

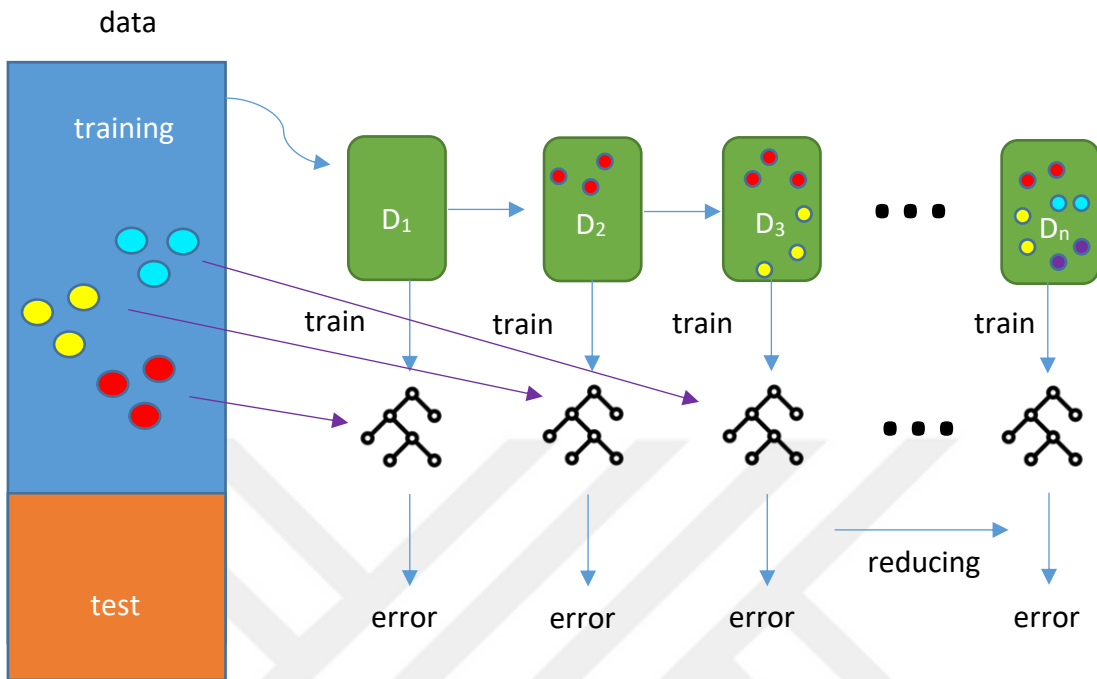


Figure 3-4. Mechanism of Gradient Boosting Machine.

During our experimental evaluation, GBM outperforms DRF in all experiments, both result sets obtained from GBM and DRF are given in result section. The data is split into a training and a test set as 70% and 30% (split ratio is selected intuitively). Separation of dataset is stratified because distribution of categories is imbalanced. Therefore, each dataset approximately contains the same percentage of samples of each category as the whole set.

3.3.1. Cross-Validation

Learning the noise and details in the training data may lead the learner to a failure of predicting significant patterns on unseen data. This is known as *overfitting*. The procedure called cross-validation is employed to avoid overfitting and tune hyperparameters with only original data (NG, 1997). We applied k-fold cross-validation which splits a given data set into a k folds where each fold is employed as a test set. During each set, it tunes the hyperparameters of the model. This approach can be costly for computation; however, it is

useful to choose hyperparameters so as to optimize the model. Figure 3.5 demonstrates how cross-validation works.

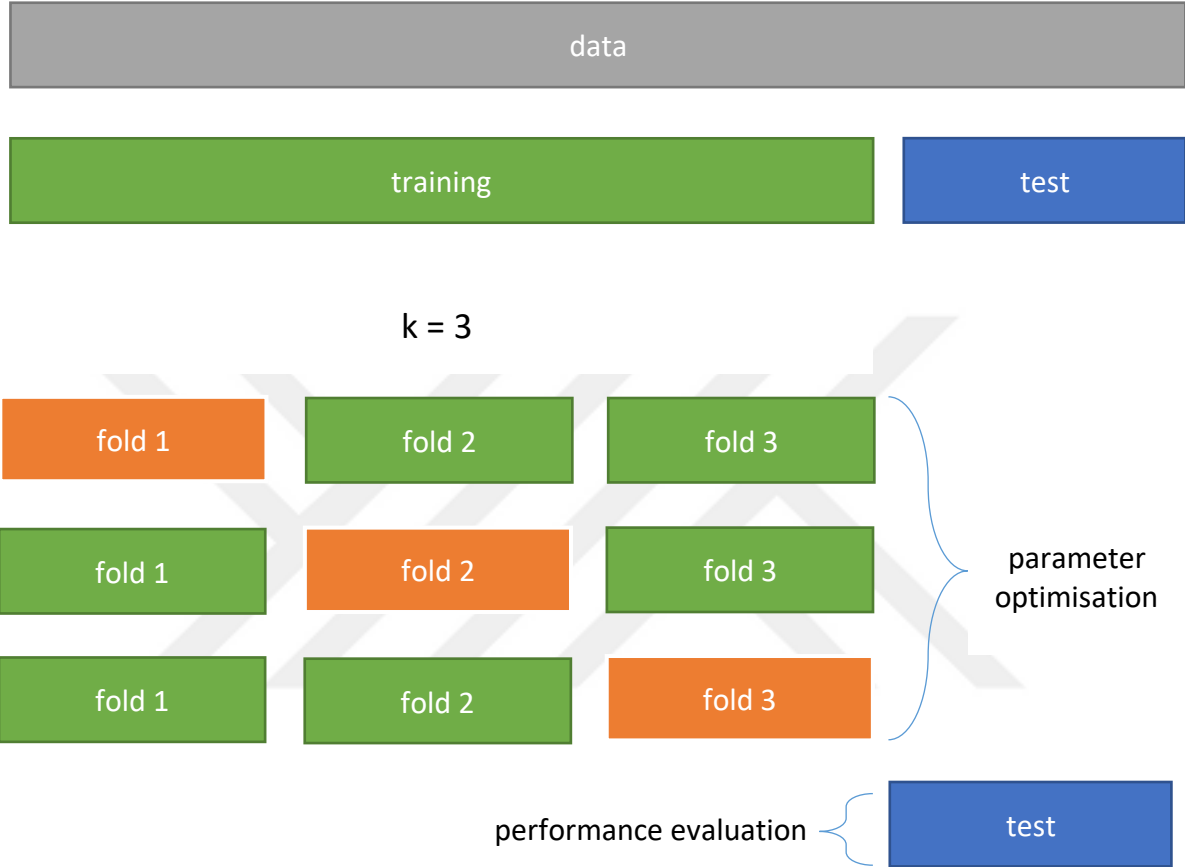


Figure 3-5. Steps of cross-validation.

3.3.2. Parameters

Parameters of classification models are tuned in compliance with results from numerous experiments. We configure the model training using parameters as follow:

- **Learning Rate:** The rate at what GBM learns when a model is built
- **Max Depth:** The maximum tree depth. Higher values increase the model complexity and it can lead to overfitting
- **Sample Rate:** The row sampling rate.

- **Column Sample Rate:** The column sampling rate. Row sampling and column sampling both improves generalization and lead to diminish validation and test set errors.
- **Number of Trees:** The size of forest. In the tree, each node corresponds to a feature from a dataset.
- **Min Rows:** Minimum number of observations. When number of observations reaches the given number, algorithm will split the leaf.

During our experiments, combinations of given parameters are examined and the values offering the most accurate results are obtained. In order to examine parameters, GridSearch is utilized. It is an exhaustive search method for parameter tuning provided by H2O. In this method, lists which include different values of parameters are given and models are trained using each combination of values from these lists.

4. EXPERIMENTS AND RESULTS

4.1. Setup

Our approach to e-commerce marketplace content classification generates a data processing and analysis pipeline that learns from batch data. The architecture of this project contains the following modules:

4.1.1. Data Collection

This module is basically a generic crawling and data scraping platform. Some sites use complex Web development techniques such as dynamic page and content loading, so we developed a Web crawler to collect data from e-commerce platforms. This module initially extracts the main product description information (i.e., product title, description, specifications list or table) in a standardized way as demonstrated in Figure 4.1. Pseudo code of a web crawler is demonstrated in Appendix B.

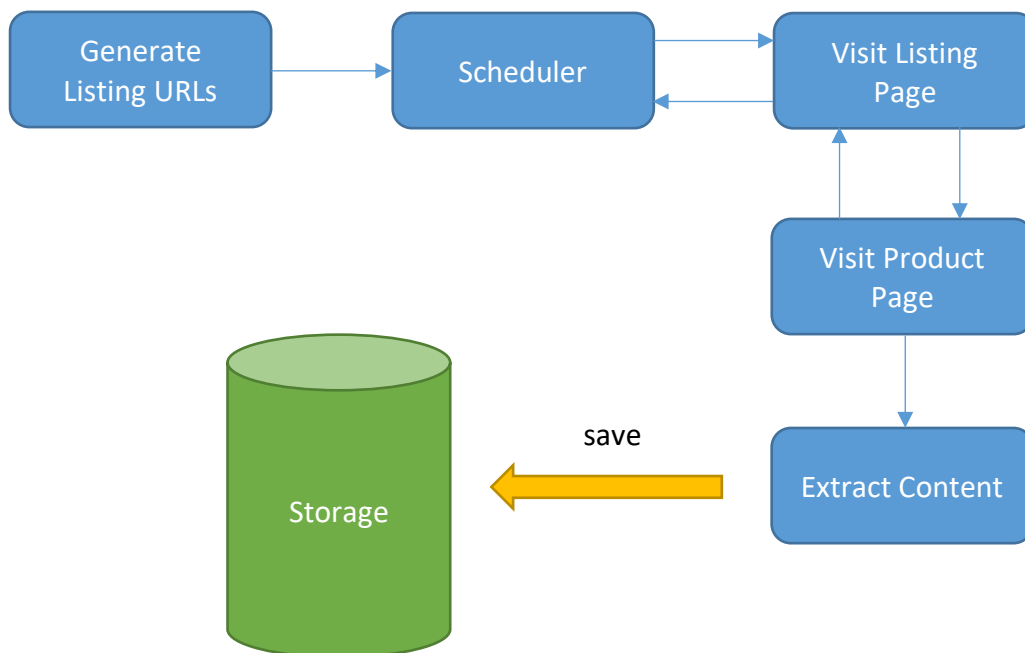


Figure 4-1. Structure of a Web Crawler.

We have fetched 1.18 million product data instances in the crawling process. When duplicates (considering product’s description) are dropped, 520K product instances with 32 different categories are left. Since the model underfits to learn categories whose percentage is less than 0.01%, these minority categories are removed from the collection and the final number appears to be 423K and the number of categories diminished to 22 ranging from Baby products to Mobile Phones, from products under Hobbies to Vehicles. Most populated categories are Computers, Sports, and Music in descending order. Product data distribution is demonstrated in Table 4.1.

Table 4.1. Initial data distribution of products.

Category Name	# Products	% In Dataset
Baby	10,511	2.5%
Garden	17,965	4.3%
Computers	43,541	10.3%
Mobile Phones	14,341	3.4%
Electronics and Gadgets	19,178	4.5%
Home Decor	29,511	7.0%
Home Electronics	22,930	5.4%
Photography	21,408	5.1%
Clothing and Accessories	26,421	6.3%
Hobbies	25,000	5.9%
Magazine and Movie	21,529	5.1%
Cosmetics	12,859	3.0%
Collections	6,695	1.6%
Music	33,224	7.9%
Office	14,661	3.5%
Toys and Video Games	13,842	3.3%
Watches	6,658	1.6%
Sports	31,651	7.5%
Jewelry	9,573	2.3%
Technical Electronics	11,110	2.6%
Vehicles	8,902	2.1%
Foods and Beverages	11,013	2.6%

4.2. Results and Discussion

Experiments were run on a computer with 2 X 16-core, 2.10 GHz Intel Xeon CPU and 128GB RAM. In order to evaluate the model performance, the *confusion matrix* also called an *error matrix* is utilized, and this matrix is demonstrated in Figure 4.2.

Predicted \ Actual	1	0
1	True Positive	False Positive
0	False Negative	True Negative

Figure 4-2. Confusion Matrix.

- True Positive (TP): Predicted 1, real value is 1
- True Negative (TN): Predicted 0, real value is 0
- False Positive (FP): Predicted 0, real value is 1
- False Negative (FN): Predicted 1, real value is 0

Recall and *prediction* metrics which are extremely important for model evaluation are measured using TP, FP, FN, and TN values. Recall is the percentage of total relevant results predicted without error by the model. Precision is the ratio of positive predictions which was correct. The formulae of recall (a.k.a. *Sensitivity*) and precision (a.k.a. *Predictive Value Positive (PVP)*) in Equation (8) and (9) where *TP*, *FP*, *TN*, and *FN* stand for *True Positive*, *False Positive*, *True Negative* and *False Negative*, respectively.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

In order to evaluate the model performance, *F-score*, also known as *F1 score* is used as a measure of an accuracy of test. F-score is the harmonic mean of Recall and Precision. This metric is useful since it balances the utilization of Recall and Precision and provides a more realistic measure for model performance. The formula of F-score is given in Equation (10) where *P* is Precision and *R* is Recall.

$$F - score = \frac{2 \times P \times R}{P + R} \quad (10)$$

We use macro average evaluation metrics (Manning et al., 2010), since standard classification measurements are based on binary classification. Equations (11-13) show how to measure macro-average precision, recall and F-score performance measurements where P_{c_x} is the precision value of category x , c is category, and R_{c_x} is the recall value of category x .

$$macro\ precision = \frac{P_{c_1} + P_{c_2} + \dots + P_{c_n}}{n} \quad (11)$$

$$macro\ recall = \frac{R_{c_1} + R_{c_2} + \dots + R_{c_n}}{n} \quad (12)$$

$$macro\ F - score = \frac{2 \times macro\ precision \times macro\ recall}{macro\ precision + macro\ recall} \quad (13)$$

Our experiments demonstrate that the augmentation of word embeddings, bigram and unigram representatives using TF-IDF achieve the best performance reaching an F-score of 0.82, a precision of 0.83, and a recall of 0.81 using GBM. Moreover, varied adjustments on word embeddings based features are compared. In the generation of word embeddings, varying the vector size and the window size, we have witnessed that the best vector representations are obtained using a vector size and a window size of 120 and 20 in respective order. In order to observe the precise effect of different types of features that represents product information, we train different classifications as follows:

- A vector representation of each product (referred to as *vecs*)
- A vectoral similarity between product and each category (referred to as *sim*)
- Bigram representatives of each category (referred to as *bigram*)
- Unigram representatives of each category (referred to as *unigram*)
- Price, length of title, length of description (referred to as *props*)

Table 4.2. and Table 4.3. demonstrate the empirical results obtained using combinations of features using GBM and DRF in order. We measured macro-averaged precisions (P), recalls (R), and F-scores (F1) at each classification.

Table 4.2. Results obtained with GBM.

Features	P	R	F-Score
vecs + props	0,776	0,832	0,803
unigram + props	0,710	0,733	0,722
bigram + props	0,343	0,751	0,471
similarity + props	0,712	0,730	0,721
vecs + unigram + props	0,781	0,818	0,799
vecs + bigram + props	0,772	0,816	0,794
vecs + similarity + props	0,759	0,791	0,775
unigram + bigram + props	0,722	0,744	0,733
unigram + similarity + props	0,790	0,807	0,798
bigram + similarity + props	0,744	0,764	0,754
vecs + unigram + bigram + props	0,791	0,838	0,813
vecs + unigram + similarity + props	0,795	0,828	0,811
unigram + bigram + similarity + props	0,776	0,800	0,788
vecs + unigram + bigram + similarity + props	0,818	0,837	0,827

Table 4.3. Results obtained with DRF.

Features	P	R	F-Score
vecs + props	0,777	0,816	0,796
unigram + props	0,693	0,710	0,701
bigram + props	0,409	0,489	0,445
similarity + props	0,747	0,764	0,756
vecs + unigram + props	0,779	0,810	0,794
vecs + bigram + props	0,777	0,815	0,796
vecs + similarity + props	0,780	0,806	0,793
unigram + bigram + props	0,709	0,727	0,718
unigram + similarity + props	0,771	0,793	0,782
bigram + similarity + props	0,754	0,774	0,764
vecs + unigram + bigram + props	0,784	0,815	0,799
vecs + unigram + similarity + props	0,789	0,813	0,801
unigram + bigram + similarity + props	0,776	0,798	0,787
vecs + unigram + bigram + similarity + props	0,794	0,819	0,806

The highest F-score is obtained from GBM and DRF with the utilization of all features as shown in Tables 4.2 and 4.3. Results indicate that GBM outperformed DRF by %2 according to the highest F-scores obtained from experiments. Among the feature sets, the most

significant one is vector representations for each product. The lowest F-score is measured to be 0.343 in GBM and 0.409 in DRF when only bigrams are used as a feature. Augmentation of vector representations and unigrams increases the model performance but adding bigrams to vectors and unigrams does not enhance the classification performance significantly. Lastly, when we combine all features, our classifier exhibited the highest F-score and precision. The highest Recall is acquired when vector representations, unigrams, and bigrams are augmented in GBM, however, the greatest Recall is obtained using the combination of all features in DRF.

The results also show that the best performances are acquired in Vehicles and Jewelry categories. Our classifier performed the worst in Collections and Garden categories. This performance issue stems from the correlation among two categories and is discussed in detail in the following section.

As we discussed under the *Separation of Product's Title and Description* section, unigram and bigram representatives are separated as *title unigram*, *title bigram*, *description unigram*, and *description bigram*. When the previous unigram and bigram features are removed and the separated features are added back into the feature set, the new features are as follows:

- Title bigram representatives of each category (referred to as *title_bigram*)
- Title unigram representatives of each category (referred to as *title_unigram*)
- Description bigram representatives of each category (referred to as *desc_bigram*)
- Description unigram representatives of each category (referred to as *desc_unigram*)

Performance improvement with the new features using GBM and DRF is observed as Tables 4.4 and 4.5 demonstrate.

Table 4.4. Results after the separation of title and description features with GBM.

Features	P	R	F-Score
vecs + title_unigram + similarity + props	0,826	0,840	0,833
vecs + title_bigram + similarity + props	0,812	0,830	0,821
vecs + desc_unigram + similarity + props	0,806	0,823	0,814
vecs + desc_bigram + similarity + props	0,806	0,823	0,814
vecs + title_unigram + title_bigram + similarity + props	0,829	0,843	0,836
vecs + title_unigram + desc_unigram + similarity + props	0,827	0,841	0,834
vecs + title_unigram + desc_bigram + similarity + props	0,827	0,841	0,834
vecs + title_bigram + desc_unigram + similarity + props	0,813	0,833	0,823
vecs + title_bigram + desc_bigram + similarity + props	0,813	0,832	0,823
vecs + title_unigram + title_bigram + desc_unigram + similarity + props	0,830	0,844	0,837
vecs + title_unigram + title_bigram + desc_bigram + similarity + props	0,830	0,844	0,837
vecs + title_unigram + desc_unigram + desc_bigram + similarity + props	0,830	0,843	0,836

vecs + title_bigram + desc_unigram + desc_bigram + similarity + props	0,814	0,834	0,824
vecs + title_unigram + title_bigram + desc_unigram + desc_bigram + similarity + props	0,831	0,845	0,838



Table 4.5. Results after the separation of title and description features with DRF.

Features	P	R	F-Score
vecs + title_unigram + similarity + props	0,803	0,829	0,815
vecs + title_bigram + similarity + props	0,785	0,818	0,801
vecs + desc_unigram + similarity + props	0,776	0,809	0,792
vecs + desc_bigram + similarity + props	0,775	0,809	0,791
vecs + title_unigram + title_bigram + similarity + props	0,806	0,832	0,819
vecs + title_unigram + desc_unigram + similarity + props	0,806	0,831	0,818
vecs + title_unigram + desc_bigram + similarity + props	0,804	0,831	0,817
vecs + title_bigram + desc_unigram + similarity + props	0,788	0,819	0,803
vecs + title_bigram + desc_bigram + similarity + props	0,785	0,819	0,802
vecs + title_unigram + title_bigram + desc_unigram + similarity + props	0,806	0,833	0,820
vecs + title_unigram + title_bigram + desc_bigram + similarity + props	0,806	0,834	0,820
vecs + title_unigram + desc_unigram + desc_bigram + similarity + props	0,806	0,832	0,819

vecs + title_bigram + desc_unigram + desc_bigram + similarity + props	0,787	0,820	0,803
vecs + title_unigram + title_bigram + desc_unigram + desc_bigram + similarity + props	0,808	0,835	0,821

As the impact of each feature on the performance has been measured, the impact of each process during data preprocessing was also examined. Following phases in the data preprocessing are applied sequentially and one after the other:

- Raw data retrieved from crawling (referred as *raw_data*)
- Lowercasing (referred as *lowercased*)
- Punctuation Removal (referred as *punctuation_removal*)
- Stop Words removal (referred as *s_words_removal*)
- E-Commerce words removal (referred as *ecom_words_removal*)
- Trimming (referred as *trimming*)

Table 4.6. Impacts of data preprocessing phases.

Phases	P	R	F-Score
raw_data	0,763	0,795	0,778
raw_data + lowercased	0,808	0,832	0,819
raw_data + lowercased + punctuation_removal	0,803	0,834	0,818
raw_data + lowercased + punctuation_removal + s_words_removal	0,813	0,840	0,826
raw_data + lowercased + punctuation_removal + s_words_removal + ecom_words_removal	0,836	0,845	0,836
raw_data + lowercased + punctuation_removal + s_words_removal + ecom_words_removal + trimming	0,831	0,845	0,838

As depicted in Table 4.6, the greatest impact was observed by lowercasing of text. Except punctuation removal, each phase has an impact on model performance. The greatest F-score was obtained by applying all of data preprocessing steps. Without trimming, the highest precision and recall values are acquired. In the future work, punctuation removal may be improved, and it might have a higher impact on results.

In order to evaluate the model parameters mentioned before in Section 3.3.2, we made an exhaustive search to achieve F-score. Table 4.7 demonstrates the 10 results obtained from exhaustive search. In the following table combinations of the parameters below are applied on the model training and results are compared.

- Number of trees (referred as n_trees)

- Learning rate (referred as *l_rate*)
- Max Depth (referred as *max_depth*)
- Column Sample Rate (referred as *col_sample_rate*)
- Min Rows (referred as *min_rows*)
- Number of bins (referred as *n_bins*)

Table 4.7. Exhaustive parameter search results.

n_trees	l_rate	max_depth	col_sample_rate	min_rows	n_bins	F-score
150	0,1	10	0,7	15	20	0,859
150	0,1	10	0,5	15	20	0,855
100	0,1	10	0,7	15	20	0,854
150	0,1	30	0,7	15	20	0,851
150	0,1	20	0,7	10	20	0,850
150	0,03	10	0,7	15	30	0,845
50	0,1	10	0,7	15	20	0,840
50	0,1	10	0,7	5	20	0,836
150	0,001	10	0,25	15	20	0,811
150	0,001	10	0,25	15	10	0,808

Results obtained from the exhaustive search indicate, among the combinations of parameters, the highest F-score is acquired reached 0,859 when number of trees is 150, learning rate is 0,1, maximum depth is 10, column sample rate is 0,7, minimum rows are 15, and number of

bins are 20. Decreasing the learning rate and the column sample rate reduces the F-score. We further extract from the experiments that increasing depth of trees does not contribute to a rise in the model performance, and the optimal depth size appears to be 10 regarding the experiments.

As we discuss in the following section, high correlations have been measured between some categories. Correlations between Home Décor - Collection and Home Décor – Garden are 0,374 and 0,112 respectively. They can also be considered under the Home Décor category as well regarding their domain. When we merge these three categories, the results on the Table 4.8 are achieved.

Table 4.8. Results obtained with merging highly correlated categories.

Precision	Recall	F-score
0.852	0.881	0.867

Results show that, the best F-score is acquired when the highly correlated categories are merged.

4.2.1. Error Analysis

Overall performance is assessed by the average of each classifier’s performance. Looking into the classifier performance for each category, Collection, Home Decor, and Garden categories seem to be the mostly misclassified categories in our classifier. In order to evaluate errors in misclassification, we analyze the correlation between categories using the misclassification rate between every pair of categories. Misclassification rate is calculated using false positive and true positive values of classifier as defined in (14) where FP_{xy} is the total number of false positive values, when our algorithm misclassifies category x as y , TP_x is the total number of true positive values for category x .

$$Corr_{xy} = \frac{FP_{xy}}{FP_x + TP_x} \quad (14)$$

The correlation between Home Decor – Collection is the highest rate: 0.374. There is an ambiguity between these categories when we look at the intersection of bigram and unigram representatives. Results show that the highest interception in Home Decor category is observed under Collection category. These representatives contain common words such as “wooden”, “mirror”, “walnut tree”, and “oil paint” which confuse the classification by even the humans. Besides Collection – Home Decor, misclassification rates between Garden – Home Decor and Technical Electronics – Computer categories are also high. Correlations between each category are indicated in Appendix C.

5. CONCLUSION AND FUTURE WORK

In this paper, we present an accurate system which handles product categorization using product’s information. Utilization of word embeddings and TF-IDF shows that a satisfactory performance is obtained using finely selected parameters in both word embedding model and classification from non-moderated user-generated text. Two supervised learning algorithms, Gradient boost machine (GBM) and distributed random forest classification (DRF) are compared and GBM outperforms DRF by around %2 in F-score. GBM achieves a categorization performance reaching an F-score of 0.87. In addition to the performance score, results demonstrate that the most significant feature set is the vector representations of each product. When we separate product’s title and description for TF-IDF features, results indicate that product’s title has is more critical in distinguishing categories. The results from the related work and the comparison of our results with the state of the art are indicated in Table 5.1. Some of the studies categorized products in a multi-level taxonomy. Only the 1st level taxanomy product categorization was taken into consideration for the comparison. Kozarova, 2015 and Lee & Yoon, 2018 outperformed our study. As mentioned in the fututre work, we are planning to employ well-moderated data to obtain higher F-scores.

Table 5.1. Comparison of the Results.

Study	Precision	Recall	F-score	Accuracy
Kozarova, 2015	-	-	0.88	-
Sun et al., 2014	0.92	-	-	0,90
Ristoski et al., 2018	0.74	0.64	0.69	0.88
Lee & Yoon, 2018	-	-	0.90	0.90
Our study	0.85	0.88	0.87	-

In the future, we plan to improve this work with several extensions. Including images and retrieving information from product's images can improve model performance. In addition to the image processing, increasing the size of vector representations is one other future plan. In order to implement this feature, we have to optimize the system and increase the computation performance since the amount of the computations in both feature extraction and model generation is going to rise. As we separate TF-IDF features as title and description, the separation of vector representation features as title and description may increase the model performance. As the results demonstrate, our classifier successfully handles the non-moderated user-generated text data which have noisy attributes. In addition to new implementations, we want to implement our model with well-moderated or benchmark data to acquire better results.

APPENDIX A

Examples from unigram and bigram dictionaries

Category	Unigram Dictionary	Bigram Dictionary
Computer	ram, işlemci (processor), usb	intel core, ram hdd, macbook pro
Garden	mm, metre (meter), su (water)	çim halı (grass carpet), geçirmeyen şalt (switch), denge makaraları (balance spool)
Vehicle	boya (paint), km, motor (engine)	sağ çamurluk (right fender), sol çamurluk (left fender), local boya (local paint)
Decoration	ağşap (wooden), halı (carpet), avize (chandelier)	yağlı boya (painting), boya tablo (painting), yemek masası (dinner table)
Technical Electric	servo (servo), dedektör (sensor), tamir (repair)	non-unique bigrams for this category
Collection	alış (buy), gümüş (silver), saat (watch)	non-unique bigrams for this category
Mobile Phone	şarj (charge), iphone, samsung	ithalat yollu (export), yanlış kusurlu (defective)
Photography	lens, canon, nikon	şarj aleti (charge unit), fotoğraf makinası (camera), canon eos xbox one, kol tamiri (controller repair)
Toys and Video Games	oyun (game), kol (controller), xbox	tükenmez kalem (pen), dolma kalem (fountain pen), ofis mobilyaları (office furnitures)
Office	ofis (office), kalem (pencil), koltuk (seat)	koşu bandı (treadmill), su geçirmez (water resistant), maç forması (jersey)
Sport	bisiklet (bicycle), kg, vites (gear)	plak yüzeyinde (record surface), yüzey gürültüsü (surface noise)
Music	plak (record), kapak (cover), müzik (music)	non-unique bigrams for this category
Hobbies	oyun (game), oyuncak (toy), yaş (age)	su arıtma (water filter), kötü koku (maladour), sıcak su (hot water)
Electronics and Gadgets	su (water), montaj (assembly), filtre (filter)	

Home Electronics	tv, hd, kamera (camera)	led tv, ev sinema (home cinema)
Clothing	beden (size), deri (leather), ayakkabı (shoes)	beden bilgisi (body size)
Foods and Beverages	kg, organik (organic), zeytin (olive)	gezen tavuk (chicken), katkı maddesi (additive), zeytin yağı (olive oil)
Jewelry	gümüş (silver), kolye (necklace), altın (gold)	bayan küpe (women's earring), bayan kolye (women's necklace), takı aksesuar (accessories)
Magazine and Movie	kitap (kitap), yayıncılık (publisher), baskı (edition)	dünya edebiyatı (dünya edebiyatı), hamur kağıra (paper pulp)
Watches	duvar (wall), saat (watch), kol (arm)	duvar saati (clock), kol saati (wristwatch)
Baby	bebek (baby), yatak (bed), beşik (cradle)	bebek arabası (baby carriage), ana kucağı (baby carrier), emniyet kemeri (seat belt)
Cosmetics	saç (hair), parfüm (perfume), tester	kalıcı makyaj (permanent makeup)

APPENDIX B

Pseude code of a web crawler

- 1:** The data set is split into k folds
- 2: Repeat for** each fold
 - 1:** Assign the group of fold except selected one as a training data set
 - 2:** Assign the selected fold as a test data set
 - 3:** A model learns on the training set and evaluate the performance on the test set
 - 4:** Keep the evaluation score and remove the model
- 3:** Summarize the behaviors of the model and hyperparameters



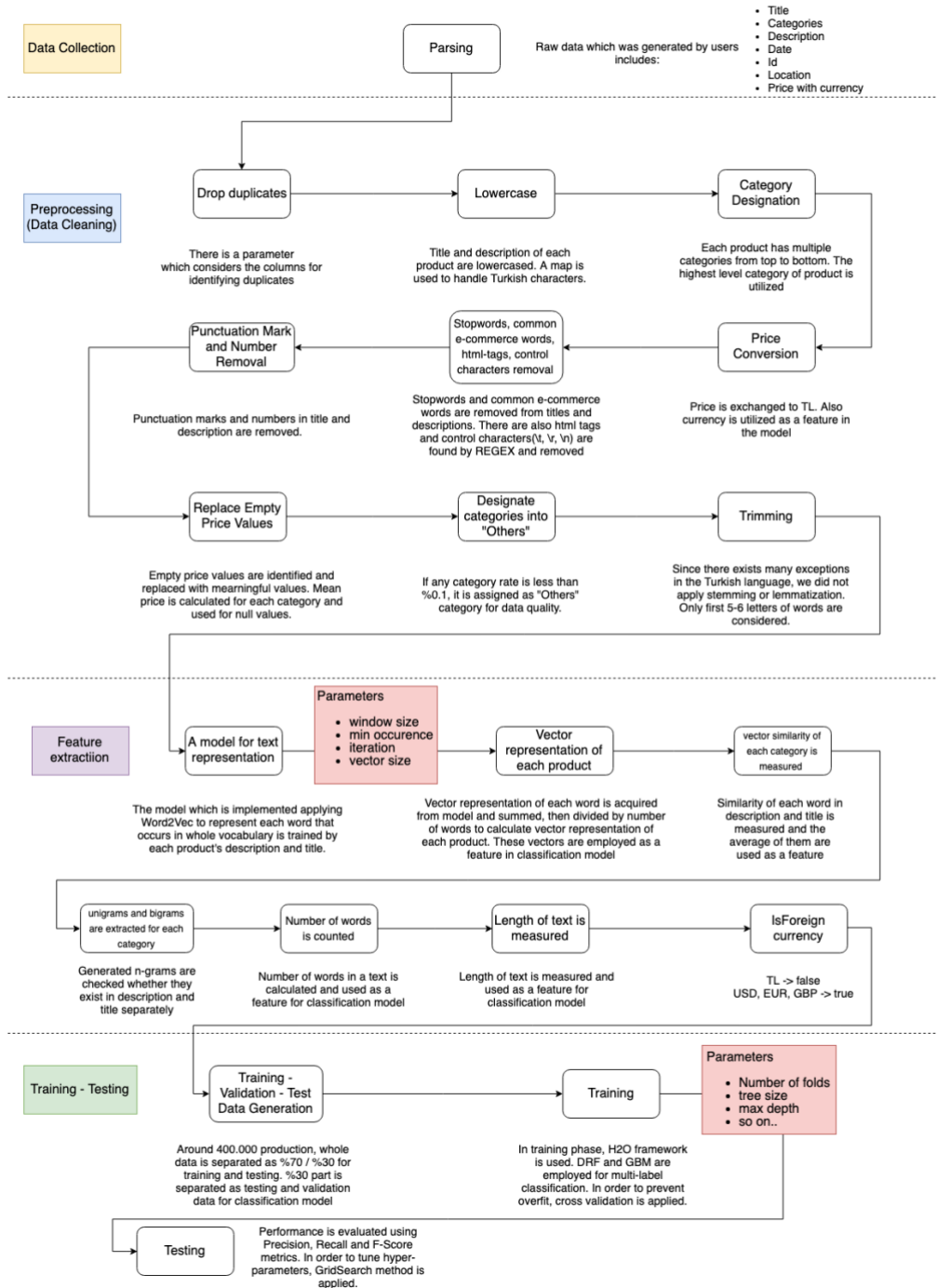
APPENDIX C Correlations between each category

Categories	Baby	Garden	Computers	Mobile Phones	Electronics & Gadgets	Home Decor	Home Electronics	Photography	Clothing & Accessories	Hobbies	Magazine & Movie
Baby		0,009	0,008	0,001	0,011	0,100	0,004	0,003	0,028	0,056	0,001
Garden	0,003		0,017	0,003	0,034	0,112	0,015	0,003	0,008	0,019	0,001
Computers	0,001	0,005		0,023	0,004	0,006	0,017	0,007	0,004	0,005	0,003
Mobile Phones	0,000	0,005	0,111		0,003	0,010	0,021	0,015	0,007	0,007	0,002
Electronics & Gadgets	0,000	0,036	0,016	0,002		0,031	0,009	0,002	0,004	0,013	0,001
Home Decor	0,007	0,032	0,007	0,001	0,015		0,007	0,002	0,016	0,016	0,008
Home Electronics	0,000	0,009	0,065	0,017	0,006	0,008		0,016	0,001	0,007	0,001
Photography	0,001	0,008	0,026	0,007	0,002	0,007	0,022		0,002	0,008	0,002
Clothing & Accessories	0,003	0,004	0,005	0,002	0,001	0,031	0,001	0,003		0,013	0,002
Hobbies	0,016	0,019	0,013	0,002	0,003	0,045	0,005	0,012	0,006		0,008
Magazine & Movie	0,001	0,002	0,019	0,000	0,000	0,011	0,005	0,001	0,002	0,016	
Cosmetics	0,003	0,027	0,010	0,002	0,009	0,044	0,006	0,004	0,015	0,007	0,003
Collections	0,002	0,014	0,010	0,000	0,009	0,204	0,022	0,009	0,016	0,027	0,034
Music	0,000	0,007	0,021	0,012	0,001	0,007	0,030	0,006	0,002	0,008	0,010
Office	0,002	0,041	0,061	0,002	0,005	0,097	0,005	0,007	0,011	0,020	0,014
Toys & Video Games	0,000	0,001	0,049	0,006	0,003	0,007	0,004	0,004	0,002	0,025	0,006
Watches	0,000	0,005	0,009	0,003	0,000	0,010	0,006	0,002	0,016	0,006	0,002
Sports	0,004	0,019	0,008	0,003	0,008	0,019	0,004	0,006	0,048	0,019	0,004
Jewelry	0,001	0,007	0,004	0,000	0,000	0,038	0,001	0,000	0,012	0,003	0,001
Technical Electronics	0,001	0,046	0,076	0,007	0,021	0,008	0,022	0,013	0,000	0,020	0,001
Vehicles	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Foods & Beverages	0,001	0,018	0,003	0,000	0,017	0,027	0,002	0,001	0,002	0,009	0,001

Categories	Cosmetics	Collections	Music	Office	Toys & Video Games	Watches	Sports	Jewelry	Technical Electronics	Vehicles	Foods & Beverages
Baby	0,006	0,003	0,004	0,002	0,000	0,000	0,019	0,001	0,000	0,000	0,001
Garden	0,006	0,007	0,008	0,012	0,001	0,001	0,027	0,002	0,020	0,000	0,023
Computers	0,001	0,002	0,012	0,010	0,007	0,000	0,003	0,000	0,006	0,000	0,001
Mobile Phones	0,004	0,008	0,023	0,002	0,004	0,002	0,008	0,000	0,001	0,000	0,001
Electronics & Gadgets	0,004	0,006	0,006	0,003	0,000	0,001	0,013	0,000	0,006	0,000	0,008
Home Decor	0,004	0,038	0,004	0,015	0,001	0,002	0,009	0,003	0,001	0,000	0,007
Home Electronics	0,001	0,006	0,087	0,003	0,001	0,001	0,004	0,000	0,008	0,000	0,000
Photography	0,001	0,004	0,012	0,005	0,001	0,001	0,008	0,000	0,004	0,000	0,000
Clothing & Accessories	0,004	0,003	0,005	0,002	0,001	0,001	0,033	0,001	0,000	0,000	0,001
Hobbies	0,000	0,011	0,012	0,007	0,013	0,001	0,039	0,001	0,005	0,000	0,002
Magazine & Movie	0,001	0,017	0,012	0,003	0,003	0,000	0,006	0,000	0,000	0,000	0,001
Cosmetics		0,007	0,008	0,012	0,001	0,001	0,020	0,001	0,002	0,000	0,009
Collections	0,000		0,014	0,016	0,004	0,047	0,013	0,009	0,000	0,000	0,005
Music	0,001	0,005		0,003	0,002	0,000	0,009	0,000	0,002	0,000	0,001
Office	0,009	0,011	0,006		0,000	0,002	0,018	0,000	0,005	0,000	0,001
Toys & Video Games	0,000	0,002	0,007	0,000		0,000	0,009	0,000	0,000	0,000	0,001
Watches	0,000	0,063	0,005	0,002	0,000		0,005	0,003	0,003	0,000	0,000
Sports	0,004	0,008	0,005	0,003	0,003	0,001		0,000	0,005	0,000	0,001
Jewelry	0,003	0,029	0,004	0,005	0,000	0,004	0,008		0,001	0,000	0,000
Technical Electronics	0,002	0,005	0,023	0,009	0,001	0,001	0,021	0,000		0,000	0,000
Vehicles	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000		0,000
Foods & Beverages	0,005	0,001	0,003	0,001	0,000	0,000	0,006	0,001	0,001	0,000	

APPENDIX D

Diagram of the overall system



6. REFERENCES

- Aiello, S., Click, C., Roark, H., Rehak, L., & Stetsenko, P. (2016) Machine Learning with Python and H2O. H2O. ai Inc.
- Alpaydin, E. (2009) Introduction to machine learning. MIT press.
- Baio, G., & Blangiardo, M. (2010) Bayesian hierarchical model for the prediction of football results. *Journal of Applied Statistics*, 37(2), 253-264.
- Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003) A neural probabilistic language model. *Journal of machine learning research*, 3, 1137-1155.
- Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H. C., & Vursavas, O. M. (2008) Information retrieval on Turkish texts. *Journal of the American Society for Information Science and Technology*, 59(3), 407-421.
- Chavaltada, C., Pasupa, K., & Hardoon, D. R. (2017) A comparative study of machine learning techniques for automatic product categorisation. In *International Symposium on Neural Networks*, 10-17.
- Chowdhury, G. G. (2003) Natural language processing. *Annual review of information science and technology*, 37(1), 51-89.
- Cotton, M., & Liddicoat, S. (2005) U.S. Patent Application No. 10/808,730.
- Davidson, J., Liebold, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., ... & Sampath, D. (2010, September). The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, 293-296.
- Dumais, S. T. (2004) Latent semantic analysis. *Annual review of information science and technology*, 38(1), 188-230.
- Friedman, J. H. (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006) Extremely randomized trees. *Machine learning*, 63(1), 3-42.

- Joachims, T. (1998) Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning, 137-142.
- Kanagal, B., Ahmed, A., Pandey, S., Josifovski, V., Yuan, J., & Garcia-Pueyo, L. (2012) Supercharging recommender systems using taxonomies for learning user purchase behavior. Proceedings of the VLDB Endowment, 5(10), 956-967.
- Kasper, W., & Vela, M. (2011) Sentiment analysis for hotel reviews. In Computational linguistics-applications conference, 231527, 45-52.
- Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015) Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal, 13, 8-17.
- Kozareva, Z. (2015) Everyone likes shopping! multi-class product categorization for e-commerce. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1329-1333.
- Lane, T., & Brodley, C. E. (1997) An application of machine learning to anomaly detection. In Proceedings of the 20th National Information Systems Security Conference, 377, 366-380.
- Lee, H., & Yoon, Y. (2018) Engineering doc2vec for automatic classification of product descriptions on O2O applications. Electronic Commerce Research, 18(3), 433-456.
- Manning, C., Raghavan, P., & Schütze, H. (2010) Introduction to information retrieval. Natural Language Engineering, 16(1), 100-103.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013) Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, 3111-3119.
- Nadeau, D., & Sekine, S. (2007) A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), 3-26.
- Ng, A. Y. (1997) Preventing "overfitting" of cross-validation data. In ICML 97, 245-253.
- Ramos, J. (2003) Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, 242, 133-142.

Rehurek, R., & Sojka, P. (2010) Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

Ristoski, P., Petrovski, P., Mika, P., & Paulheim, H. (2018) A machine learning approach for product matching and categorization. *Semantic web*, 1-22.

Rohde, D. L., Gonnerman, L. M., & Plaut, D. C. (2006) An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8(627-633), 116.

Rong, X. (2014) word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.

Shankar, S., & Lin, I. (2011) Applying machine learning to product categorization. Department of Computer Science, Stanford University.

Sboev, A., Litvinova, T., Gudovskikh, D., Rybka, R., & Moloshnikov, I. (2016) Machine learning models of text categorization by author gender using topic-independent features. *Procedia Computer Science*, 101, 135-142.

Sebastiani, F. (2002) Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1), 1-47.

Sun, C., Rampalli, N., Yang, F., & Doan, A. (2014) Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proceedings of the VLDB Endowment*, 7(13), 1529-1540.

Uçar, K. T., Tümer, M. B., & Kıracı, M. (2019) Multi-class Categorization of User-Generated Content in a Domain Specific Medium: Inferring Product Specifications from E-Commerce Marketplaces. In *International Conference on Intelligent and Fuzzy Systems*, 247-256.

Yang, X., Macdonald, C., & Ounis, I. (2018) Using word embeddings in twitter election classification. *Information Retrieval Journal*, 21(2-3), 183-207.

Ziegler, C. N., Lausen, G., & Schmidt-Thieme, L. (2004) Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 406-415.

7. CURRICULUM VITAE

Name: Kemal Toprak Uçar

Birth Date: 18.01.1991

Telephone: +90 536 956 0 444

Git: <https://github.com/ktoprakucar>

E-Mail: toprakucar@gmail.com

Education

2016 – 2019 (expected) Marmara University - MSc, Computer Engineering (English)

2009 – 2015 Marmara University - BSc, Computer Engineering (English)

2012 – 2013 EPITA (École Pour l'Informatique et les Techniques Avancées)

Work Experience

May 2017 – Present, Software Engineer @iyzico – Istanbul, Turkey

Development and improvement of Fraud Model with Machine Learning & Artificial Intelligence to **maximize payment conversion** and **minimize chargeback rates**

Development of a risk application that **examines how much risky onboarded merchant or a merchant in progress of onboarding is**. Application helps the analysts during the merchant onboarding and following cycles using application information, success payments, refund, cancel, and dispute activities to make sensible decisions

Development of a fraud application that **detects prohibited products using its title, description and image and categorize products** in order to determine whether a product is legal or not to be purchased with Natural Language Processing and Machine Learning

Implementation of RESTful web applications with Java 11, test (Unit-Integration-Functional), relational database, Spring Boot, Spring MVC, Spring Cloud, Hystrix, Redis, Groovy, and other technologies following hexagonal architecture and clean code principles

Implementation of a lightweight Java client library for Google Cloud API

June 2015 – May 2017, Software Engineer @SoftTech (Outsourced by Sade Yazılım) – Istanbul, Turkey

Migration of Calculation Service that was implemented in COBOL to Java

Enhancement of backend services and operations that was the part of “Development of Commercial Loans Disbursement Applications” in ISBANK.

Spring Batch Development in Commission, Disbursement and Guarantee Letter projects.

Refactoring in current projects.

July 2014 – September 2014, Internship @Fraunhofer IOSB – Lemgo, Germany

A GUI Implementation of SmartFactoryOWL which is an open research and demonstration platform for digital transformation. Visualization of an existing application was developed using Java. Briefly, the system contains manufacturers and conveyors. The GUI that I have implemented can configure the system parameters such as number of products, speed of production etc. and visualize the whole flow.

June 2012 – August 2012, Internship @Cargill Turkey – Istanbul, Turkey

September 2013 – June 2015, Part-Time Student Worker @Marmara University International Office

Computer Skills:

Java, Spring Boot, Spring MVC, Spring Cloud, Spring Data, RESTful Web Services, MySQL, H2, Redis, Groovy

Python, NumPy, Pandas, H2O, scikit-learn, Gensim,

Artificial Intelligence, Reinforcement Learning, Natural Language Processing

Maven, GIT, Apache Subversion (SVN)

Testing (JUnit, AssertJ, Mockito)

Agile software development, Test Driven Development (TDD), Domain Driven Design (DDD)

Performance-Scalability optimizations, Refactoring

Projects

Frauctive – iyzico

A real time fraud detection system which has been developed to prevent fraudulent transactions using artificial intelligence to **diminish chargeback rate**. In this project, data preparation, feature extraction, and model generation have been developed by our team. Performance of the classification model is examined by F1 score and it is measured around **%0,80**. Besides its performance metrics, we have also measured the model performance on the real time dashboard. Model has been developed using H2O framework in Python. A trained model is deployable and employed in a Java application. Response time of a model including feature extraction, prediction, and decision of each transaction is around **70 ms**.

On the service side, a RESTful service retrieves the requests before payment is sent to bank then marks as accept, reject, or suspicious according the score which is generated by model. The generated score is also utilized to decide the payment as 3DS or not to increase conversion rate. We achieve a **%12 increase** on conversion rate via application of **Dynamic 3DS**.

Multi-Class Categorization of User-Generated Content in a Domain Specific Medium: Inferring Product Specifications from E-Commerce Marketplaces – Marmara University

A "marketplace" is an e-commerce medium where product and inventory information is provided by varying third parties, whereas catalog service is hosted, and payments are processed by the marketplace operator. As a result of increasing use of marketplaces, e-commerce capabilities can now be accessed by everyone. Consequently, both the number of merchants and products have been growing exponentially. Such growth raises some problems including “Does product description reflect specifications of the real one?”, “Does the seller really own the product?”, “Is this product legal for purchasing online?”, “Is this product listed under correct category?”. These problems can lead to penalties or complete close-down of the merchant as e-commerce business is regulated in most countries. We propose a methodology to detect an accurate product category from user-generated content on e-commerce marketplaces, so that proactive removal of certain products can be automated. We present our methodology as a complete system that incorporates data collection, cleaning, and categorization. In this work, we transform unstructured text into vector representations of words during machine-learning-ready dataset preparation stage. We train ML models by a large corpus of text which includes more than half a million product descriptions. Finally, we compare our results in alternate classification algorithms and varying methodologies of vector representations. **We showed that accurate predictions of text categories reaching an F-score of 0.82 can be obtained from user-generated text that may contain typos, special punctuation, and abbreviations, and comes from a non-moderated e-commerce medium.**

Migration of Calculation Module from COBOL to Java – SoftTech

The task is the migration of the module that was developed in COBOL to Java platform. Initially, revisioning platform was carried to GitHub from ClearCase, building platform was

moved to Maven from Ant then structure got ready to be implemented. First, tests were defined and written then we started to code the logic of calculation service according to the tests. **Test Driven Design (TDD)** and **Domain Driven Design (DDD)** are considered during the development from beginning to end. JUnit was utilized for test environment. Around 250 unit tests and integration tests were implemented while development. Later on a long implementation period, a batch that compares the results that are both calculated by the current service and implementing service to examine the accuracy of our module. In parallel with batch development, part of web service in the module was implemented using Spring Framework. When batch test reached to the verified status, user acceptance tests are initialized then the module was deployed into production environment.

An Autonomous Quadcopter by Using NEAT – Marmara University

My final project that I have worked on with my partner. **Unmanned Air Vehicle (UAV) can be controlled without using remote controller by using Artificial Neural Networks and NEAT topology** in this project. First, system learns with respect to rewards in the environment by using neural networks then these networks are evolved by using NEAT architecture. End of project, simulation of flying object was implemented. In the simulation, object can learn how to fly to target without crashing the obstacles by rewards. This learning occurs with Neural Networks. Then, these neural networks were enhanced by NEAT architecture to try to achieve optimal flight.

Languages:

Turkish: Native

English: Advanced

French: Intermediate (ESTP Paris B1 Certificate)

Hobbies/Personality

IEEE Marmara Student Branch Chairman

Analog photography

Playing acoustic, electric guitar, ukulele and piano

Interested in some poets and writers such as Nazim Hikmet, Sabahattin Ali, Ahmet Ümit, George Orwell, Orhan Pamuk

Following some directors, actors and their works

Folk Dance (2007 National Folk Dance Winner)

Alternative rock and hard rock music

References

Assoc. Prof. M. Borahan Tümer – Marmara University

E-Mail: borahantumer@gmail.com

Mustafa Kırac, PhD. – Afiniti

E-Mail: muskirac@gmail.com

Asst. Prof. Peter Schüller – Technische Universität Wien

E-Mail: schueller.p@gmail.com

Dipl.Inf. Jens Eickmeyer – Fraunhofer IOSB-INA

E-Mail: jens.eickmeyer@iosb-ina.fraunhofer.de