ANALYZING CUSTOMER REVIEWS IN TURKISH USING MACHINE LEARNING AND DATA SCIENCE METHODOLOGIES

A THESIS SUBMITTED TO
THE FACULTY OF ACHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

MIGENA CEYHAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

FEBRUARY, 2021

Approval of the thesis:

THESIS TITLE: ANALYZING CUSTOMER REVIEWS IN TURKISH USING MACHINE LEARNING AND DATA SCIENCE METHODOLOGIES

submitted by Migena Ceyhan in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Computer Engineering, Epoka University** by,

Assoc. Prof. Dr. Sokol DERVISHI ______________________
Dean, Faculty of Architecture and Engineering

Dr. Ali Osman TOPAL ______________________
Head of Department, Computer Engineering, EPOKA University

Assoc. Prof. Dr. Dimitrios KARRAS ______________________
Supervisor, Computer Engineering Dept., EPOKA University

Examining Committee Members:

Prof. Dr. Betim Çiço ______________________
Computer Engineering Dept., EPOKA University

Prof. Dr. Neki Frashëri ______________________
Natural and Technical Sciences Dept., Academy of Sciences of Albania

Prof. Dr. Kozeta Sevrani ______________________
Applied Statistics and Informatics Dept., Tirana University

Assoc. Prof. Dr. Carlo Ciulla ______________________
Computer Engineering Dept., EPOKA University

Assist. Prof. Dr. Edlira Martiri ______________________
Applied Statistics and Informatics Dept., Tirana University

Date: 25.02.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Migena Ceyhan

Signature:

ABSTRACT

ANALYZING CUSTOMER REVIEWS IN TURKISH USING MACHINE LEARNING AND DATA SCIENCE METHODOLOGIES

Ceyhan, Migena
Ph.D., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Dimitrios KARRAS

Digital life, especially after the introduction of Web 2.0, has significantly altered human relations, providing all people the “right of public speech”. Ideas, emotions, and opinions on many topics are generously shared in virtual environments. A new age global and digital Mouth of World is shaping the society where knowledge is the most influential power. Being fed by social media data highly dynamic in either amount or shape, automatic handling is indispensable.

Natural Language Processing, in cooperation with Machine Language techniques, has an important say in analyzing written textual data. Traditional techniques exploited in the literature are empowered when hybrid ones are applied, in accordance also with the characteristic properties of the language used and the domain-specific data. Although all the subsequent steps of the text classification chain are important, adequate feature selecting has a notable huge impact on accurate classification prediction.

In this study, a simple classification of the sentiment polarity of comments in document level of subjective texts in Turkish is done. Different domains include reviews of customers towards company products, movies, and healthcare services, deciding on the positivity or negativity of the comments. Another domain includes doctors' notes on patients' symptoms aiming to predict and thus recommend some of the most often used medical tests according to general doctors' procedures.

The features used included a part of or all distinct words roots together with their binary or frequency information. Linear or vector analysis of the feature sets was done employing Machine Learning algorithms provided by the Weka tool. Hybrid features set was proposed and found more efficient combining binary vectors and frequency meta-features from nodes and leaves of J48 tree classifier for all or a set of correlation-based selected features, improving both prediction accuracy and classification performance.

Keywords: Machine Learning; Natural Language Processing; Sentiment Analysis; Turkish; Meta-Features Selection

ABSTRAKT

ANALIZIMI I KOMENTEVE TE PËRDORUESVE NË TURQISHT DUKE PËRDORUR TË NXËNIT E MAKINËS DHE METODOLOGJI TË SHKENCAVE TË PËRPUNIMIT TË TË DHËNAVE

Ceyhan, Migena
Doktoraturë, Departamenti i Inxhinierise Kompjuterike
Udhëheqësi: Assoc. Prof. Dr. Dimitrios KARRAS

Bota dixhitale, veçanërisht pas futjes së teknologjisë Web 2.0, ka ndryshuar ndjeshëm marrëdhëniet njerëzore, duke u mundësuar dhe lehtësuar të gjithë aktorëve të saj "të drejtën e fjalës publike". Idetë, ndjenjat dhe mendimet në lidhje me shumë tema ndahen me zemërgjerësi dhe janë të kudogjendura në mjedise virtuale. Një epokë e re globale dhe dixhitale e "Fjalëve të Hallkut" po formëson shoqërinë tonë, ku dija është pushteti më ndikues. Gjatë procesit të përftimit të dijes, të furnizuar kryesisht nga të dhënat, shumë dinamike qoftë në sasi apo në formë të mediave sociale, trajtimi i tyre automatik është domosdoshmëri e pashmangshme.

Përpunimi i Gjuhëve Natyrore, në bashkëpunim me teknikat e Gjuhës së Makinës kanë një rol të rëndësishëm në analizimin e të dhënave të shkruara. Teknikat tradicionale të shfrytëzuara fuqizohen kur zbatohen në mënyrë hibride, në përputhje me vecoritë karakteristike të gjuhës së përdorur dhe tipareve specifike të lëmive të të dhënave. Megjithë rëndësinë e të gjitha hapave të njëpanjëshëm në zinxhirin e klasifikimit të teksteve, zgjedhja e duhur e karakteristikave që do analizohen kanë një ndikim të dukshëm në saktësinë e hamendësimit të klasës së dokumenteve.

Në këtë studim është bërë klasifikimi i komenteve subjektive në gjuhën turke. Lëmitë e të dhënave përfshijnë vlerësime të klientëve ndaj: produkteve të kompanive të ndryshme, filmave dhe shërbimeve të kujdesit shëndetësor. Në këto fusha synohet matja e saktësisë së metodave të propozuara në lidhje me hamendësinë e pozitivitetit ose negativitetit të komenteve të përdoruesve. Një lëmi tjetër përfshin shënimet e mjekëve mbi simptomat e pacientëve, duke synuar rekomandimin e disa nga analizave mjekësore më të përdorura bazuar në procedurat e përgjithshme të mjekëve. Karakteristikat apo tiparet e përdorura në analizë përfshinë të gjitha apo një pjesë të rrënjëve të fjalëve të ndryshme të përdorura në tekstet që u analizuan, prandaj këto tre koncepte do të përdoren në mënyrë të ndërsjelltë në këtë studim.

Analiza lineare ose vektoriale e bashkësive të fjalëve, së bashku me informacionin e tyre binar ose të frekuencës të përdorimit në secilin prej teksteve të studiuar, u bë me anë të algoritmave të fushës së Mësimit të Makinës të nxjerra nga programi i klasifikimit Weka. Bashkësia e tipareve hibride, e propozuar në këtë studim, si kombinim i vektorëve të informacionit binar të tipareve dhe frekuencës së meta-karakteristikave të nyjeve dhe gjetheve që dolën si rezultat i klasifikimit përmes pemës J48, rezultoi si më efikase për të gjithë karakteristikat ose nënbashkësi të tyre të zgjedhura sipas korrelimit, në përmirësimin e saktësisë së parashikimit dhe të performancës së klasifikimit.

Fjalët kyçe: Të Nxënit e Makinës; Përpunimi i Gjuhëve Natyrore; Klasifikimi i Teksteve; Gjuha Turke; Zgjedhja e Meta-Tipareve të Gjuhës

....

Dedicated to my family

ACKNOWLEDGEMENTS

I would like to express my special thanks to my supervisor Assoc. Prof. Dr. Dimitrios KARRAS for his continuous guidance, encouragement, motivation, and support during all the stages of my thesis. I sincerely appreciate the time and effort he has spent to improve my experience during my graduate years.

I am also deeply thankful to Assoc. Prof. Dr. Zeynep ORHAN for co-advising me and being always on my side with her precious support, without whom I would not be able to complete my studies. Special thanks go also to my former advisor Assoc. Prof. Dr. Elton DOMNORI for his understanding.

My sincere acknowledgments go to my thesis progress committee members, Prof. Dr. Betim ÇIÇO, Assoc. Prof. Dr. Arban UKA, Dr. Ali Osman TOPAL for their comments and suggestions throughout the entire thesis.

I am especially grateful to Dr. Mine MERCAN, who was a motivation engine for me, I appreciate the comfort and pleasure of every moment working with her, making me enjoy the work.

I would like to thank my whole family for being always at my side. Sincerely, without their help, this work wouldn’t be possible. I deeply thank my son, Eren, for showing patience and understanding during this period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
NLP	Natural Language Processing
ML	Machine Learning
SA	Sentiment Analysis
OMSA	Sentiment Analysis and Opinion Mining
SSL	Semi Supervised Learning
NB	Naïve Bayes
SVM	Support Vector Machine
MNB	Muntinomial Naïve Bayes
BMNB	Binarized Multinomial Naïve bayes
BNB	Bernoulli NB
IMDb	Internet Movie Database
MA	Morphological Analyzer
MD	Morphological Disambiguator
Acc	Accuracy
P	Precision
R	Recall
FM	FMeasure
CM	Confusion Matrix
TF-IDF	Term Frequency – Inverse Document Frequency
WLLR	Weighted Logaritmic Likelihood Ratio
CFS	Correlation based Feature Selection

CHAPTER 1

INTRODUCTION

Along with the progress in the digital world and the introduction of technology in each space of our living, social media has taken its place in an important corner of our lives, by revealing essential data sources for general use. Whether when releasing the Web 2.0 at the beginning of this century the creators pre-estimated or not the impact that it would bring, they provided every person having access and will to use this technology with the power to share, and on the other side, they supplied the part collecting this kind of data with the power to use this information. The importance of this instrument is getting evident each passing day. The web is now a rich data ecosystem created cooperatively by all Internet users without any cultural or physical borders and offering a diverse information ocean.

It is natural to people at the present time to criticize, comment on what they like or dislike, and share their personal view or experience after common everyday activities, such as watching a movie, listening to a new piece of music, testing a technology device, or even on a news, administrative policy, etc. through social media sites, portals, blogs, vlogs, forums, YouTube channels, etc. without hesitating. While most individuals would feel lazy, unwilling or suspicious to answer direct and formal, quantitative questions or surveys about products or experiences when asked from a company or a decision taking institution, they don’t mind expressing detailed evaluations to their peers. Either if they do this to protect each other, to "punish" the companies that they were unsatisfied with, to share something they enjoyed so that their acquaintances get informed about, to show off, to get profits from firms for advertising their products or services; all the reasons behind such disclosure remain a topic of study of social and psychological sciences, while what is also important is the fact that this is happening frequently. The general common characteristic of social media data is that they are raw, unstructured meta-knowledge, in different formats, such as textual, audio, video, pictures, url's, hashtags, networks, digital history, deduced personal information, etc. This data is in need to be analyzed, quantified, generalized or specified, and converted to numbers and summarized, consequently converted to materials where important decisions will rely on. Analyzing this kind of data is crucial especially for large companies after introducing to the market a new product or service, in order to get insight on the weak or strong sides of their products, so that they can modify them to be closer to a best liked version in their future product release. Similar analysis is also useful for consumers to get a rough idea before deciding to buy a new product. However, the analysis process may be very long, time and effort consuming because of the large amount and complexity of information. Social media and general sentiment text analysis is of much valuable use, accomplishing the task of extracting pure gold out of raw mineral. When you have the raw materials containing precious elements, and possess the appropriate technology to extract worthy stuff from it, you are considered wealthy. Otherwise, when one of the steps of the process fails, it can be exhausting and too much labor and time consuming, not justifying the efforts and expenses.

Being heavily populated with user-generated content, commenting mainly on politics, sports, news in general, social media posts; reviews of goods, services, entertainment events and much more, it offers a decent source for their interpretation and classification, which is at the same time too much work and time consuming to manually analyze. Recently, this ocean of data has become a valuable tool for extracting consumer opinions for a range of purposes ranging from customer experience management to monitoring public opinion. The huge volume of data guarantees reliability and comprehensibility for most users. These two facts on social media feedback make them preferable in the decision-making process as regarding brand analysis, business intelligence, stock market forecasting and image monitoring (Balahur, Mihalcea, & Montoyo, 2014).

In general, in the technical applications of these fields, natural language processing (NLP) and sentiment analysis (SA) are used. NLP is a branch of Artificial Intelligence (AI) which deals with human language analysis and understanding. NLP's goal is to develop and build a framework that will analyze and generate human languages. NLP techniques on derived data use Machine Learning (ML) algorithms and Data Science methodologies.

SA seeks to classify and extract subjective knowledge in source materials through the use of NLP, text analysis, computational linguistics and the automated classification of texts attempting to assess the author's attitude according to a subject or the entire document. This finds its position widely in the study of feedback and social media, from marketing to customer service. Public preferences play a major role in developing new products, preparing future plans and delivering tailored goods according to the profiles of customers. Hence, discovering reliable and cost-effective ways of using the views, desires, behaviors and thoughts of their consumers in real time is a critical and beneficial problem for businesses. In addition to its difficulties, SA remains worth pursuing for several domains, with comments being one of the easiest to access and featuring a lot of opinions (Dave, Lawrence, & Pennock, 2003).

This study is limited to revealing the polarity into positive or negative of written subjective documents belonging to some specific domains of reviews or health data in Turkish, in order to train systems with known data and offer a reliable and efficient way of predicting the polarity of unknown data accordingly.

The following sections of this chapter will deal with the motivation and need of sentiment analysis, a definition, its types and uses and similar works in this area.

Chapter 2, after providing an outline of the general sentiment analysis process, it will reveal important details about the types and characteristics of data gathered, data pre-processing steps, following with explaining the experimental setup, feature selection aspects and proposed methodologies. Consequetively, Machine Languages algorithms and the classification process through them will be described.

The results will be revealed and interpreted in Chapter 3.

Chapter 4 will include important conclusive remarks regarding this study and potential future ones.

Subsequently will be references and appendices.

1.1. Sentiment Analysis and Opinion Mining

Sentiment analysis is the interpretation of a document as positive, negative or neutral. SA is also referred to as Opinion Extraction, Opinion Mining, Sentiment Mining and Subjectivity Analysis (Jurafsky, 2018).

1.2. The need for Sentiment Analysis

With the development of internet in recent years, it has become possible to find opinions everywhere. Social networks such as blogs, Facebook, Twitter, news portals and e-commerce sites, have facilitated this situation.

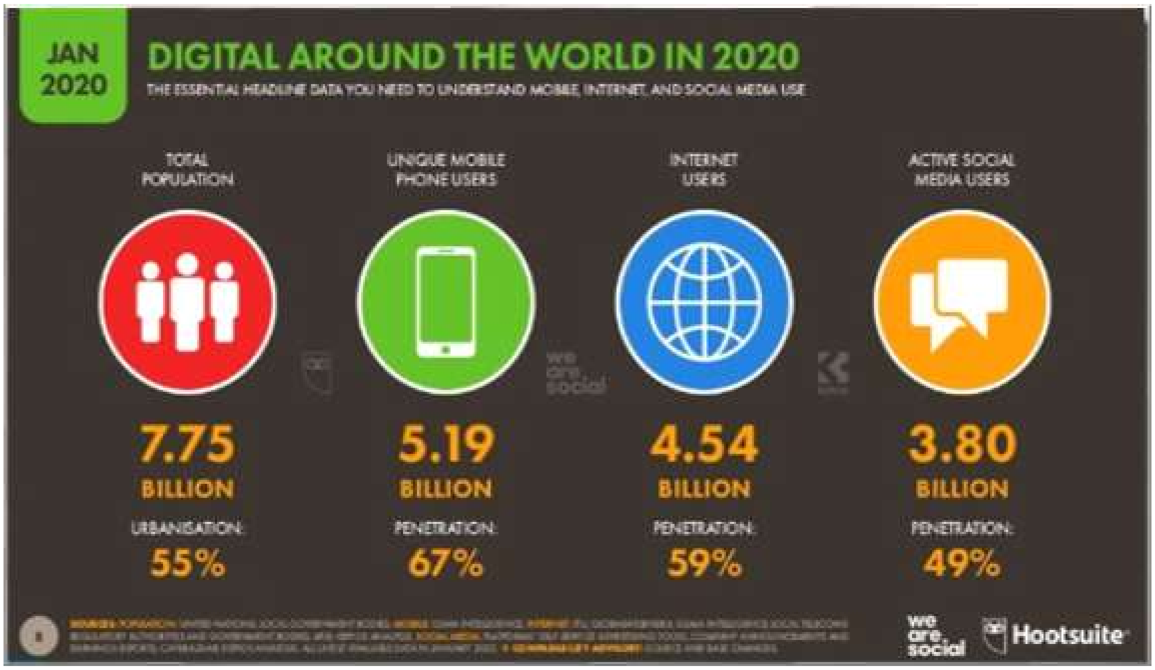


Figure 1. Worldwide internet, social media and mobile usage statistics in 2020

Figure 2. Worldwide internet, social media and mobile usage statistics in 2021

Statistics are an important indicator in terms of understanding the impact and the economic size of studies to be conducted in this field. Furthermore, statistics give us a better idea of the subject. As it can be seen in Figure 1 (Digital in 2020) and Figure 2 (Digital in 2021), the number of people using the digital world has reached billions. The rapid increase in the amount of data on the internet and the impact of internet usage on trade also change the relationship between the producer and the consumer. The annual growth in social media active users of approximate 400 million people overcomes a lot the growth in population of approximate 120 million people (Digital in 2020), (Digital in 2021). Nowadays, both producers and consumers share their opinions and experiences on social media. These data are the first source of reference for a subject, brand, product or service.

Considering only Twitter, millions of tweets are posted and new accounts are opened, with 500 million tweets are sent daily by May 2020 (Sayce, 2020)). It is estimated that out of 7.8 billion of the world population, more than 4.6 billion individuals (58% of the complete population) is dynamic in web, with North Americans having the most elevated entrance rate with around 95% of their population (Internet World Stats, 2020). Nonetheless, these incredible numbers exceed the users' tracking capacity, thus, increasing the need for auxiliary tools day by day.

As an illustration of a social media platform, Facebook gets visited at least daily by 74% of its clients in the US (Zephoria - Digital Marketing, 2020). People share their opinions and get quite influenced by others' opinions, too. This power of social media can also explain quite the common exploitation of it from the political and business leaders around the world. Each post is followed by countless comments, which reveal a lot about the support or opposition related to the post. Products are associated with several reviews from people who either suggest or complain about the product.

As an instance of a corporation that has appreciably advanced currently is Netflix. The wide variety of paying streaming subscribers differs as 21, 66 and 182 millions of customers respectively in 2011, 2015, and 2020. Only during the first six months of

2020, under the favorable effects of the pandemic situation, 26 millions of new subscribers have chosen this company (Statista, 2020). Before starting to watch for hours people don't mind querying for some moments others' opinions. Sometimes, however, this process can take more than minutes because of the abundancy of the reviews.

Reviews can be of great help in analyzing customer feedback and one of the areas that can benefit a lot from similar research is the health system. People are very concerned about health and well-being issues, so nearly all patients or potential patients want to consult and learn the experiences of other people before they receive any health service. The Internet has become the first contact of people in need of medical advice, even before consulting medical experts.

According to Turkey's 2017 annual report, 22 percent of total health spending (almost $7.9 billion, recognizing that the overall per capita expenditure is $445 and the population was $80,745,000) was invested in the private health sector, which is a substantial amount (Global Health Expenditure Database (GHED), 2020). Hospitals are very interested in holding and growing their customer share and they are mindful that this way passes by assessing the satisfaction and reducing the dissatisfaction of their patients and their families. According to the facts of the Turkish Statistical Institute, health sector spending has risen significantly, particularly during the last 15 years. While the national healthcare expenditure amounted to 8,248 million Turkish Liras, this amount increased to 94,750 million Turkish Liras in 2014 (Tüfekci & Asığbulmuş, 2016).

For this reason, it is not enough to only learn from good experiences, people want to learn more from adverse experiences in order to take steps before anything unpleasant can happen. For this purpose detecting all negative feedback is a priority, and it is important to reach all potential negative comments as this method is automated, besides the consistency of the tests (Tüfekci & Asığbulmuş, 2016).

Although so many sources of information might be useful, considering their abundance and quantity, they can become a difficult-to-digest information overload. While information is easily reachable in the internet for online shopping, it is easy to get lost among hundreds of comments for a single product. Instead, it would be useful to provide a more understandable summary. At this point, sentiment analysis applications emerge, which interpret and classify information. Tools to summarize or automatically color the data according to their polarity, positiveness or negativeness, are crucial in this area. An example to this can be seen in

, Figure 4 and d)

Figure 5 a, b, c and d. (Kim, Ganesan, & Sondhi, 2017).

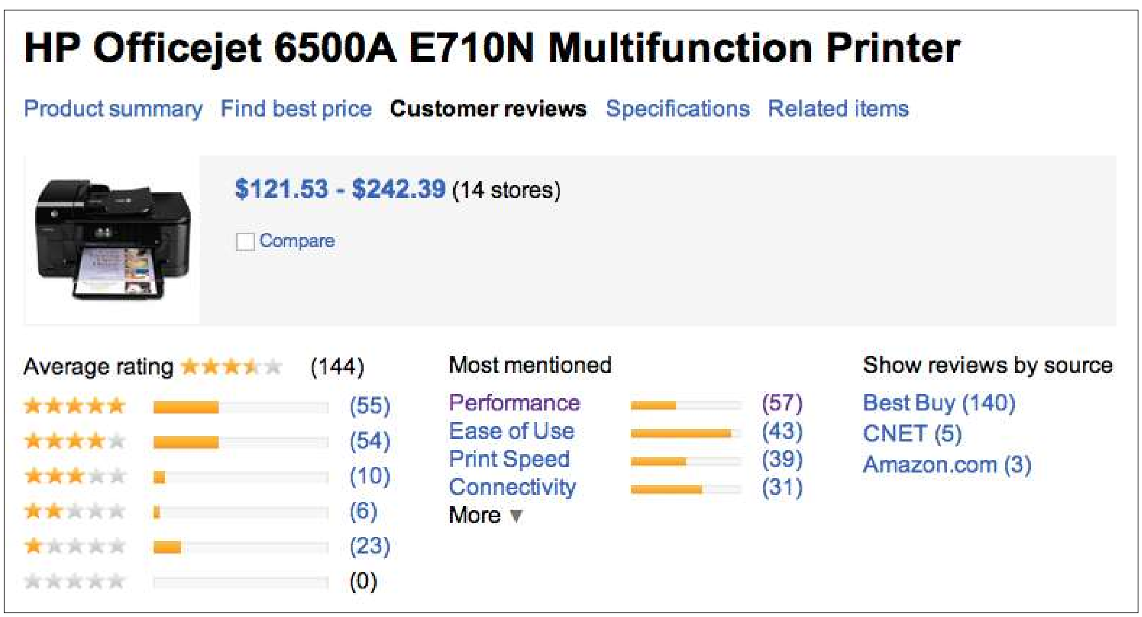


Figure 3. Bing product reviews (Jurafsky, 2018)

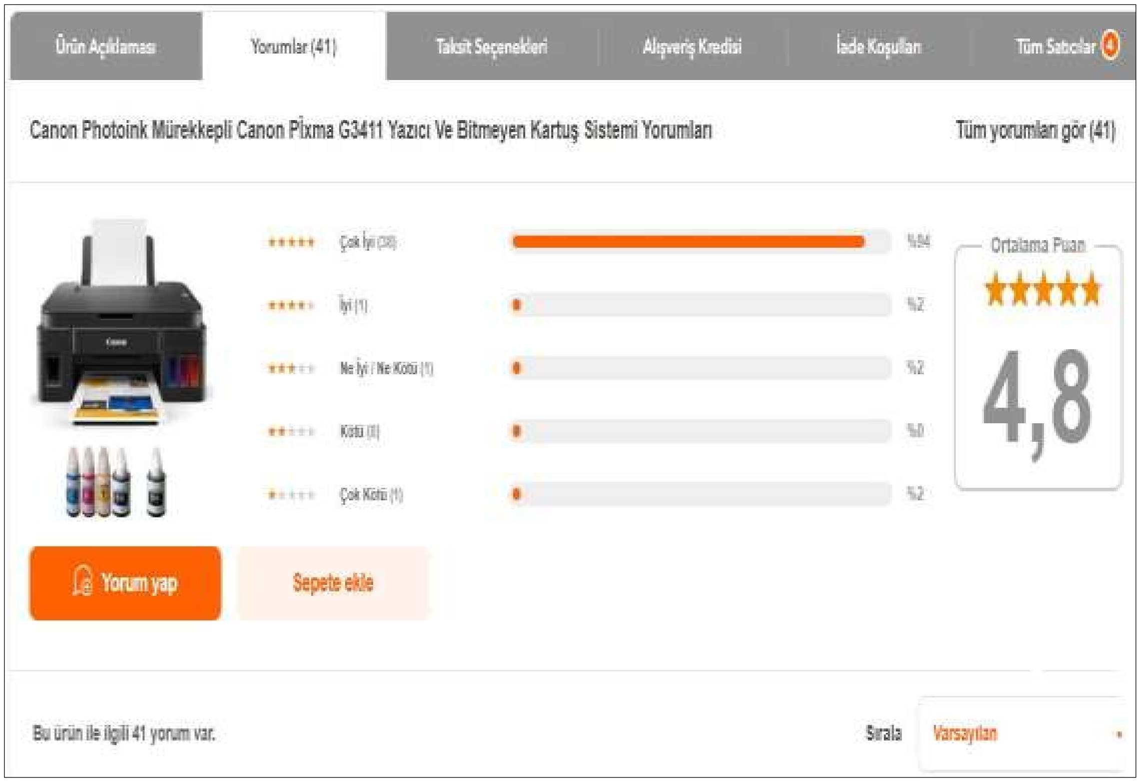
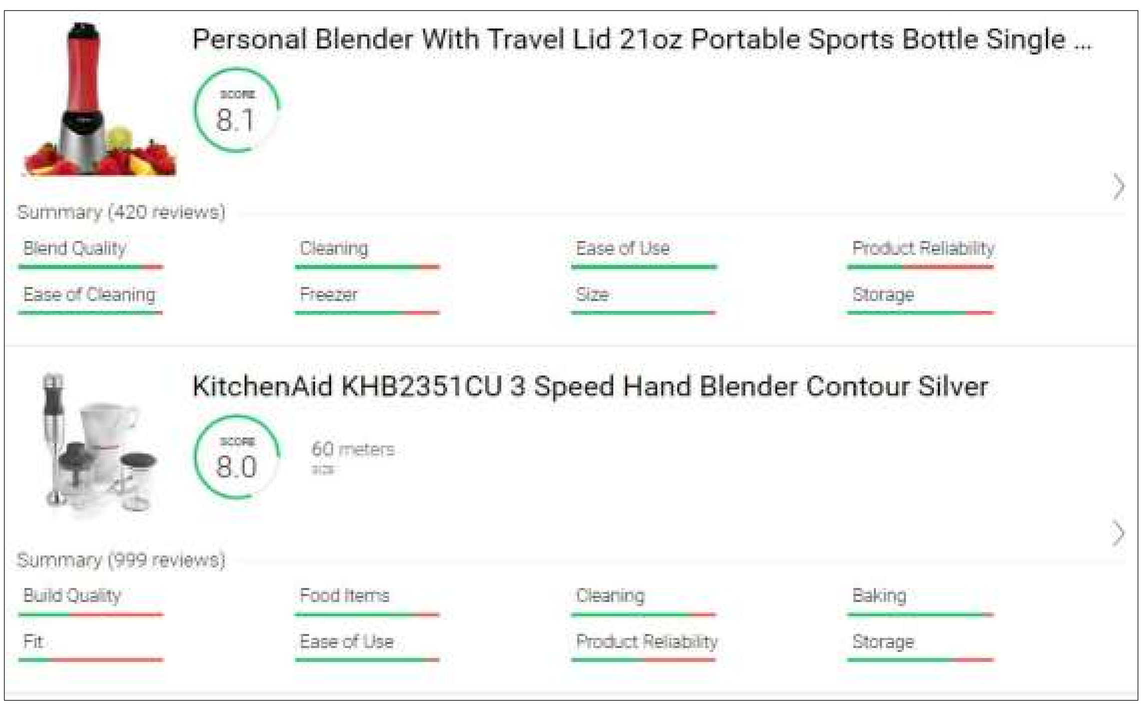


Figure 4. HepsiBurada (AllHere) product reviews



a)


Canon Color Image CLASS LBP622Cdw Wireless Mobile Ready Duplex Laser Printer Compact Size White
SCORE
7.2
/10
TOTAL REVIEWS
277
OLD REVIEWS
0
UNVERIFIED
22
SPAM TEST
PASS
Review Summary - 255 recent, verified reviews
Detailed Mode
PRINT QUALITY
print, printing, prints, toner, printers
188
+69%
-31%

b)


SETUP
set, easy, printer, instructions, wifi
51
+76%
-24%
EASE OF USE
set, setup, easy to set, easy to setup, wifi
34
+76%
-24%
DEVICE COMPATIBILITY
printer, mac, wireless, network, connected
34
+47%
-53%
NETWORK
printer, network, connect, connected, wi fi
32
+59%
-41%
SOFTWARE PLATFORM
print, error, google, linux, ip address
30
+43%
-57%

c)

d)

Figure 5.a,b,c,d. thereviewindex Amazon product English reviews aspects

1.3. Where and how to use Sentiment Analysis

Sentiment analysis is used in many areas. Some examples are (Jurafsky, 2018):

- The film industry seeks answers to questions such as: Are comments positive or negative; What will the revenue be; and Can the continuation of the film be shot, and, then takes decisions accordingly.
- Trade and production: gives directions to production strategies by using it as the source of research into questions such as: what do people think about the products and what are the products’ liked and disliked aspects.

- Opinion polling: Appropriate policies are adopted by analysing the public's views on topics such as what is the level of consumer confidence or is hopelessness increasing.
- Politics: Political life is especially used in predicting election outcomes.
- Forecasting: It is deemed appropriate to make predictions on any subjects using sentiment analysis.
- Statistics provide a decent answer to why entrepreneurs and businessmen should include search operations and social networks in their marketing plans and why it is impossible for them to remain in the market without them. Below are some striking data on the topic:
- People who use social networks and social media regularly make up 84% of internet users and this rate is growing rapidly.
- Google (by far) is the most popular search engine. By 2020, Google accounts for over 79% of the total desktop search traffic.
- Google performs more than 5.5 billion searches every day.
- 97% of consumers search for a company online.
- Users refer 34% to the first results of their search results.
- The rate of referring to the top four results in the search results is 83% (Freeland, 2017)
- Positive reviews cause confidence of youth between the ages of 18 and 34 to increase by 91%.
- 72% of users employing Google search visited a store eight km from their search location.
- 88% of customers are known to call or visit a store within 24 hours after searching from a mobile device (Daneghyan, 2019).

1.4. What are some benefits of Sentiment Analysis

The following can be mentioned as benefits of SA (Botego, 2014):

- Providing brands with a 360-degree perspective on internet, providing an effective reputation management.
- Automatically performing a function which is increasingly impossible to do manually.
- Providing the necessary tool for the accurate determination of the companies' product development, pricing and customer relationship policy through instant, daily, weekly and monthly reports.
- Responding promptly and immediately to crisis and preventing increase in costs due to crisis by listening to ideas and opinions about the brand's products and services.
- Providing the opportunity to see customer expectations not only through brand and product name, but also with sector-related keywords.
- Ensuring the company's competitive advantage by following not only its own brands but also the contents of rival companies.
- Scanning decentralized web platforms and following all targeted sources of content from a single interface.
- Providing institutions with economic benefits by reducing customer relationship and reputation management costs.

SA helps to determine people's emotions and opinions about events, services, products, other people, and so on. While positive feedback encourages the company, negative feedback assumes undertaking deterrent steps. Sometimes firms get help from SA when they want to compare themselves to other firms. Apart from that, it is an important area for marketing managers, politicians, online product managers, entrepreneurs and anyone who needs consumer reviews (Yurt, 2014).

1.5. Basic information on sentiment analysis

SA also gets support from many different areas. Text analysing, Natural Language Processing, Data Mining and Artificial Intelligence are some of them. The number of studies on the subject has especially increased in recent years. However, studies conducted in Turkish are not so many.

SA can be dealt with at different levels (Jurafsky, 2018):

- *Simple level*: Classifies all available data as a whole, only as positive or negative.
- *Complex level*: Classifies the data only as positive or negative, but also assigns a positive or negative rating ranging, for example from one to five.
- *Advanced level*: Examines and classifies the target, source and complex attitudes. For example, if a cell phone is being examined, it deals with issues such as which features of the cell phone (screen, performance, processor, etc.) should be considered and how to interpret their comments separately as positive and negative.

SA may also differ when it comes to handling the data it uses (Alpaydın, 2018):

- *Document-Level*: It works on a document. The simplest level is comparison. A document corresponds to an opinion.
- *Sentence-Level*: A document can contain many emotions. Sentences should be separated. The general approach is to base on the previous sentence. Recent approaches are based on the type of the sentence (conditional clauses, interrogative clauses, humorous clauses, etc.).

- *Aspect-based*: This approach is different compared to others. It works on the properties of the spoken object.

1.6. Methods used in Sentiment Analysis

Machine learning methods are used for SA. In short, Machine Learning is the study of providing a computer with a human feature by means of software and it is a sub discipline of Artificial Intelligence (Amasyalı F. , 2008). Regarding the data, Machine Learning is divided into four main categories (Alpaydın, 2018):

- *Supervised Learning:* Simply defined, supervised learning is carried out through the classification process. Supervised learning is a method used to find the class, that is the label of a sample whose type is unknown, by making use of known types, that is, labelled ones. The training criteria set is necessary for the classification process. For an example given in the classification process, it is based on the method of finding the class to which it belongs based on the training criteria set. Finding labelled data for supervised learning is difficult and expensive, because human support is needed to label the data.

- *Unsupervised Learning:* Unsupervised learning is done through clustering. Labels of samples are not clear in unsupervised learning. Clustering is based on the logic of taking similar samples into the same cluster. Finding unlabelled data for unsupervised learning is easy and cheap.

- *Semi Supervised Learning (SSL):* Semi supervised learning, which can also be called a rough combination of supervised and unsupervised learning, is actually a classification process. It is the process of using the small amount of available labelled data to label unlabelled data with similar features in the same environment.

- *Reinforcement Learning:* Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize cumulative reward. It does not need labelled input/output pairs. Instead the focus is on finding a balance between exploration of uncharted territory and exploitation of current knowledge. Many reinforcement learning algorithms for this context use dynamic programming techniques. They can be used in autonomous vehicles or in learning to play a game against a human opponent (Auer, Jaksch, & Ortner, 2010).

Besides the main categories of learning mentioned above, lately many other learning methods have emerged, such as ensemble learning as a combination of different methods and patterns, self-learning, learning through feedback, feature learning, robot learning, association rules learning, etc.

1.7. State of the Art

Although many challenges exist, the research in this area has a rising trend and it promises high returns in the business world, and even in many other related application domains. An extensive literature review conducted by Mostafa (2013) suggests that most Sentiment Analysis applications might be classified into four distinct categories: product reviews, movie reviews, politically oriented abstractions and stock market predictions. In this study overall efforts to analyze sentiments in general topics, especially in four selected domains will be discussed, such as online reviews about healthcare, products, movies, and patients' anamnesia.

Initially, some previous works, the state of the art of the sentiment analysis of texts in general and especially in Turkish will be covered.

(Bollen, Mao, & Zeng, 2010) held a study on social networks sentiment tracking stock market based on daily Twitter text content by using two mood tracking tools, the first one measuring positive vs. negative mood and the second tool measuring mood in terms of 6 dimensions (Calm, Alert, Sure, Vital, Kind, and Happy). The tools were used to detect the public's response to the presidential election and Thanksgiving Day in 2008, getting an accuracy of 86.7% in predicting the daily collective mood.

From a study on hotels from TripAdvisor comments (Bulchand-Gidumal, Melián-González, & López-Valcárcel, 2013) it was indicated that client satisfaction from hotels of similar characteristics differs in 11.38% depending on hotel location.

(Buche, Chandak, & Zadgaonkar, 2013) conducted a research focusing on the area of OMSA. One important problem in sentiment analysis of product reviews is to produce summary of opinions based on product features. After providing a survey about the main tasks and approaches of SA, an overall picture of what is involved in developing a software system for opinion mining on the basis of survey and analysis has been provided.

(Burnap, Williams, & Rana, 2013) tried to forecast social tension, which marks the loss of normal relations between groups in online communities, by using computational Twitter analysis in a study case relating to detecting racism feelings in individual tweets, concluding on the importance of combining approach performed better than some traditional ML algorithms.

(Choi, Ko, Kim, & Kim, 2014) held a study to classify terrorism related documents. They used WordNet hierarchy to get information about word similarity and n-gram data frequencies in news domain. The proposed method effectively extracts context words from the text and identifies terrorism-related documents

1.7.1. SA in Reviews

SA on reviews is one of the best known domains (Dave, Lawrence, & Pennock, 2003)

In one study (Na, 2010) the opinion mining of movie reviews from discussion board threads, user reviews, critics' reviews and bloggers' postings were performed. Reviews sentence length, lexicon, and parts-of-speech information was considered, concluding that opinion holder words like verbs and adjectives were used more in reviews, respective to an abundance of nouns used in bloggers or critics posting. The most commonly used positive and negative words and their patterns per each domain were determined.

Zhuang (2006) applied ML techniques movie reviews dataset in order to summarize the opinion polarity. They aimed to retrieve specific feature sets in the text and the expressed opinion, for example, "sound effects" and "excellent".

Pang (2002) similarly utilized support vector machines (SVM) acquiring 82.9% of precision in categorizing opinion summaries of movie reviews. They made use of both single words (unigrams) and pairs of consecutive words (bigrams) to classify the comments, as well of different classification algorithms, concluding that Naive Bayes, Maximum Entropy and SVM, have performed better in text classification than in sentiment classification. In fact, it is difficult to get better outcomes, because of specific characteristics of natural languages, but in specific domains, we can get good enough outcomes.

Thet, Na and Khoo (2008) conducted studies classifying online movie reviews. In one of them, they used machine learning and information extraction techniques by correctly determining the pronouns and co-referencing them to relate them to different aspects, such as the cast, the director, the effects, etc. and the overall rating of the movie. In the next task, the authors tried to reveal also the strength of the polarity of the comment according to a suggested computerized method, by taking into consideration the grammatical dependency structure of each clause analyzed according to a computational linguistics approach.

An advantage of the lexicon-based approach as compared to more generally used machine learning is that in the former, training set need not be labeled previous to the classification. They work according to text grammar analysis principles, while the elder fits the algorithms to the training set characteristic patterns. It is interesting that besides being inferior to machine learning methods in specific domains, lexicon-based methods can be quite better for wider domain sets.

For instance, a lexicon-based approach (Taboada, 2011) is used for six distinct corpora from various domains, with a 75–80% accuracy.

ML techniques, on the other hand, result more efficient for a distinctive domain, with 86.4% accuracy for a movie review summarization for a given dataset (Pang B. &., 2004). Once again it is implied that, although showing a weaker performance in data classification within one domain due to training dataset pattern overfitting, lexicon-based methods are more robust and show better results in cross-domain text classification process, in the current work, getting better scores in blog postings and video game reviews.

Sindhwani (2008) build a semi-supervised lexical model by merging lexical sentiment information, unlabeled data, and labeled training data. In three of the domains used in the study, such as products, political, and movie reviews, this strategy outdid purely supervised and competing semi-supervised approach.

There exist few studies in other natural languages, generally not applying very different methodologies than those used in text classification in English. Nevertheless, they are important in their field as novel applications in other languages. In a study in the Spanish language that has been conducted by Martínez-Cámara (2011) for movie reviews classification, they used several ML techniques and attained a high accuracy of 86.84% when SVM was applied.

The SA region is newly being studied in morphologically rich languages, and not much research has been published in the field. Turkish is not an exception for languages other than English. New words can be extracted in Turkish by adding suffices which makes SA a challenging task. Working with the Turkish language in itself is difficult, where only a limited number of studies in sentiment analysis area exist.

One of them (Kaya, 2013) studied the sentiment analysis of Turkish political columns on web documents. Their approach considered transfer learning in Turkish. In transfer learning, the aim is to extract needed knowledge from one or more tasks and then to transfer extracted information into a target task. In this work, the unigrams and the bigrams together with polar Turkish terms are used as classification characteristics to categorize unseen documents. The authors used four different classifiers: Naive Bayes, Maximum Entropy, Support Vector Machine, and the character-based n-gram language model and compared their accuracies. They concluded that Maximum Entropy and n-gram language model is more efficient than Support Vector Machine and Naive Bayes classifiers. The classification accuracy in different cases ranges from 65% to 77%. On the other side, several works have studied causal association rule mining.

Erogul (2009) investigated sentiment in two movie datasets in the Turkish language, applying English language sentiment analysis approaches.

Turksent (2010) is an annotation tool developed specifically for manual sentiment analysis of Turkish social media posts.

Yet another study in the Turkish language compares methods of text representation (Amasyalı M. F., 2012).

An ambitious study (Vural, 2013) aimed to determine the polarity of movie reviews by transtlating Sentistrength library to Turkish. They used a large corpus of Turkish movie reviews and they stated that although the framework was unsupervised, the performance approached the performance of supervised polarity classification.

Amanet (2017) studied Twitter data using the emotion categories like “Happy", “Appreciation" etc., defining the most effective word sets for each emotion. Turkmen (2016) worked on the aspect-based sentiment polarity of online customer reviews.

In Kanburoğlu (2018) thesis adjective clustering was used to automatically guess Turkish movie review scores of 76% accuracy. Through this study it was possible to measure also the reliability of the two popular sentiment lexicons SenticNet and SentiWordNet, resulting in a moderate level of agreement between lexicons and human judgments with an accuracy of 79%.

Orhan (TurkLang, 2014) automatically predicted the text polarity in customer product comments domain by making use of language characteristic features of the reviews and by utilizing ML techniques with a high level of correctness. An optimistic upcoming aim of their research is to categorize texts on any topic.

1.7.2. SA in Healthcare Systems

Engineering and computational areas such as AI and NLP help the analysis of data from various fields. Reviews can be of great help in analyzing customer feedback and one of the areas that can benefit a lot from similar research is the health system. People are very concerned about health and well-being issues, so nearly all patients or potential patients want to consult and learn the experiences of other people before they receive any health service. The Internet became like the first contact of people with medical knowledge, even before consultation of medical experts.

A research on health care in Turkey showed that the first combined selection criteria in hospitals in Turkey were based on a specialist doctor (43.7%), the second was based on confidence (40.2%), while the next came access facilities (33.9%) and overall satisfaction (23.1). Those are accompanied by preferences, accessibility of rates, advice from relatives and acquaintances, and institutional study. Since trust and doctor preference is important in the health care system, people make sure they get accurate details from family or friends, or even unknown individuals willing to share their experiences. Disliking a hospital was primarily measured by inadequate exams (28.1%), incompetence of the doctor (24%), high rates (19.1%), accompanied by unethical behaviour, lack of empathy, inadequate cleanliness, medical supplies, insufficient physical conditions, etc.

Very few interdisciplinary studies are available that apply statistical methodologies to automatically detect opinions in data collected from the health system. In a very interesting study, diagnoses of mental disorder were targeted. The study was conducted on depression, mania and healthy adults diagnosed beforehand. With Naïve Bayes, Bayesian Logistic Regression and Support Vector Classifier machine learning algorithms, the semantic categories found in the Turkish version of the General Inquirer Harward III dictionary were used in addition to syntactic characteristics. A mobile platform that detects disturbed psychological conditions in patients exchanging messages was designed to build specific dictionaries for each psychological disorder (Orhan, Mercan, & Gökgöl, 2020).

A very interesting study (Zunic, Corcoran, & Spasic, 2020) on health and well-being social media comments examined 86 studies in this area and found out that “on average, accuracy is around 80%, and it does not fall below 70%. This is well below accuracy achieved in SA of movie reviews, which is typically well over 90%. In SA of service and product reviews, the results are closer to those in health and well-being with just more than 80% for service reviews and just below 80% for product reviews. However, the performance still lags behind the state of the art achieved in these 2 domains when measured by F-score, which was found to be below 60% on average. F-measure achieved on service and product reviews is found to be above 70% and 80%, respectively. In summary, the performance of SA of health narratives is much poorer than that in other domains".

1.7.3. SA in Health Recommandation Systems

According to World Health Organization definition e-health is the usage of information and communication technologies (ICT's) in the health domain. E-health can be of use in administering the patients' treatment, research area, health education and monitoring the public health (eHealth, 2020). E-health and m-health applications have become an increasing trend, where the delivery of healthcare and services by electronic means and via mobile communication devices has led to various opportunities for this sector.

People in the health industry are looking forward to the best solutions proposed in order to get the most out of the huge but chaotic treasure of data. The term Recommender Systems has emerged as systems which help its users to select items, products or information they are interested in out of the chunk of data available in the cyberspace.

The technology advancement have given rise to enormous amount of data in the digital domain. Recommender Systems facilitates the users in selection of the most appropriate things from a large amount of data available by using data mining techniques along with prediction algorithms. Nowadays recommender systems are of wide use in health care industry in order to provide better health services to patient, as well as to facilitate doctors and hospital staff to make decisions (Kamran & Javed, 2015).

A powerful ICT-based tool able to make health care delivery more effective and efficient was defined by Rodrigues (2010) as Health Information System (HIS) which are recommender systems for the health domain. Being an intensive industry, where reliable and quick information is of vital importance for planning and monitoring proper health care services at organizational, regional, national and international levels, using HIS quickly became imperative and very widespread due to the huge amount of available data and data sources for healthcare, the inclusion of many people and not only the experts, for management and strategic planning issues as well as research purposes (Haux, 2006). In these circumstances the consistent database was maintained by the hospital server to update the queries instantly by the user.

Other research includes suggesting to use correlations among nursing diagnoses, outcomes and interventions to create a recommender system for constructing nursing care plans, leaded by Duan, Street & Xu (2011).

(Kuo & Chung, 2012) developed a healthcare information system for Comprehensive geriatric assessment (CGA) by integrating medical-related technologies with information technology.

(Nasiri, Minaei, & Amir, 2016) studied chronic patients and tried to predict any other associative diseases.

As a sequel to the above studies comes this one attempting to implement, as one component of it, a recommender system about the examinations to be asked from patients based on their symptoms and anamnesis.

CHAPTER 2

DATA AND METHODS

For SA, the following procedure has been applied:

2.1. Algorithm:

The algorithm steps are:

- Collect data.
- Convert data into text.
- Ensure that data are corrected and audited.
- Extract important and decisive words and phrases used in positive and negative reviews in the data.
- Perform morphological analysis of data.
- Reduce to one the number of multiple results from morphological analysis.
- Process the texts and extract the features planned to be used.
- Determine assessment methods.
- Type the Naïve Bayes method and try it with the specified features.

- Prepare csv files of above features to use the Naïve Bayes method in Weka program and test Naïve Bayes.

- Use a new method to make a classification with the help of a vocabulary derived from important words or phrases that are important only in positive or negative comments.

- Compare the results in the program and those in Weka.

The work flowchart of the steps performed in order to complete the text categorization task is shown in Figure 6 as follows:

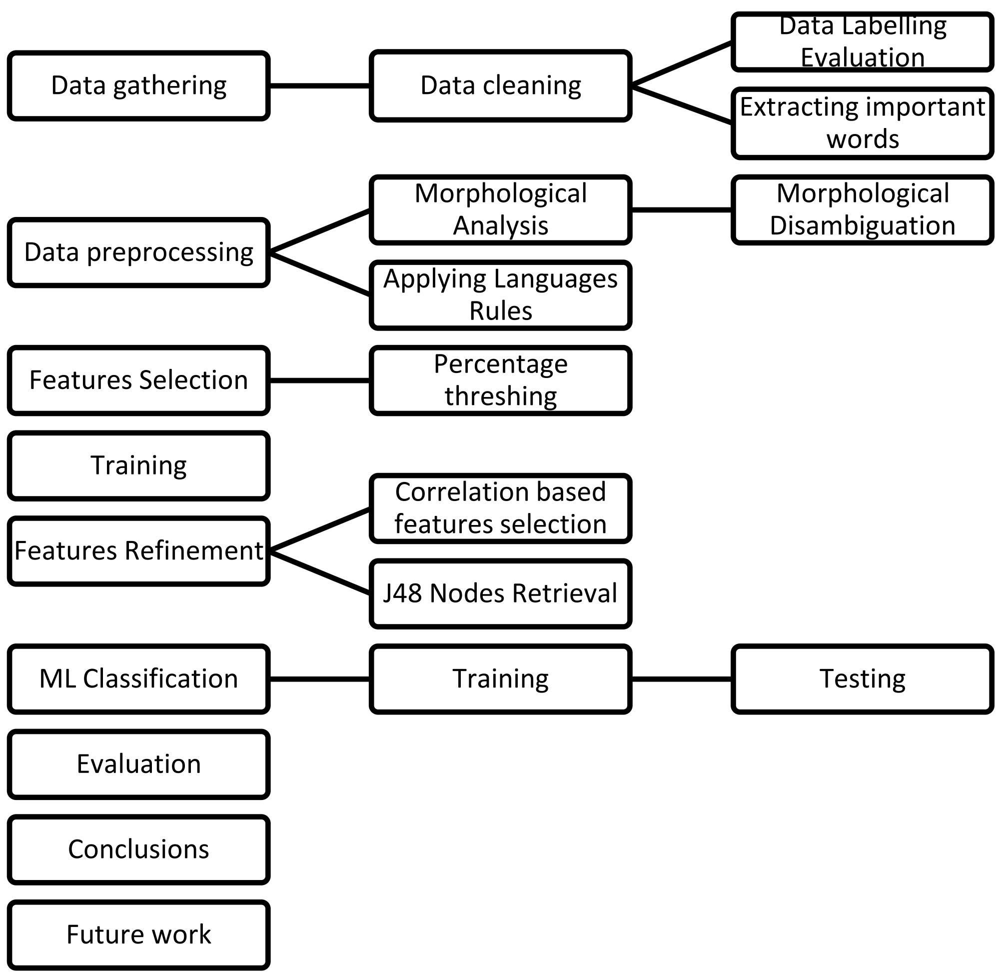


Figure 6. The up-down work flowchart for automatical data classification

2.2. Data Description

Data collecting is for sure one of the most principal steps in SA. Good, reliable data is the starting point for a good classification task. Especially when social media data is concerned, it becomes more important because of the specifics related to such data, regarding fairness, misspelling errors, abreviations, emoticons, mistyping errors, and for Turkish data especially weather the usage of Turkish specific characters was done correctly matters a lot in the final results. First of all, if you are going to use unlabeled data and assume their polarity automatically, this is the main issue. Another parameter is the length of the data, the number and format. The domain of the data is also one of the determining factors for the result of a successful text analysis. All these concerns are going to be explained in detail in the following section. (Orhan Z. , Giriş, Ceyhan, & Mercan, 2014)

This thesis was conceived to be non-domain dependent in order for the results to be stronger, wider and more reliable. This is achieved by using some sample datasets from different domains. Besides the diversity of data domains, the components of this study are constrained to special domains, as it is normal to most SA studies, but all the methods are applicable to all of the domains used without considerate difference. Each domain is selected as pilot data in a certain field, being impossible to process, even download and gather all the data in a specific domain. The size of the data extracted and used in each of the domains is enough to reliably categorize each of the results.

2.2.1. Data Collection

In this study four main datasets have been exploited. Details about the datasets is shown as follows:

- Customer Product Reviews Dataset
- Movie Reviews Dataset
- Hospital healthcare Review Dataset
- Patients Anamnesis Dataset.

2.2.1.1. Customer Product Reviews Dataset

The preprocessing of raw data that involves finding the reviews and extracting them from the web sites, correcting some errors, etc. are essential tasks since the actual data are extracted from several places. There could be more than one comment per page in the case of positive comments, therefore the program checked all the comments and related information in the web page. Noticeably, the number of positive data is higher than the number of negative data. The detailed statistics about the data are presented in (Ceyhan, Orhan, & Karras, RevOpiD '18, 2018). Table 1 shows statistics about number of positive, negative and total comments downloaded, number of data used for training and test for each category, and information about the words and features used both in unigram and bigram methods, while sections a) and b) of Table 2 show sample text from each class of data.

Table 1: Customer Review Training and Test data information

Class	Documents Download	Dataset			Unigram		Bigram	
		Total	Train	Test	Words	Features	Words	Features
Pos	2497	220	198	22	5643	1115	6640	5067
Neg	392	100	90	10	6710	1245	7874	4089
Pos+Neg	2899	320	288	32	-	1890	-	8186

Table 2: Sample from Customer Review positive and negative data.

a) Positive sample data
Yüksek kaliteye, maximum performansa ihtiyaç duyuyorsanız bu ürünü kaçırmayın. *If you need high quality, maximum performance don't miss this product*
b) Negative sample data
Buzdolabımın aynı arızayı tekrarlamasından bıktım. xxxx çözüm yerine hep aynı parçayı değiştirerek tek amacının para kazanmak olduğunu bana ispatladı. Firmadan şikayetçiyim. Konu hakkında yetkili biri ile görüşmek istiyorum. *I am tired of my fridge keeping making the same defect. xxxx proved to me that their only goal is to make money by replacing the same part instead of finding the solution. I am complaining about the company. I would like to speak to someone in charge of the subject.*

2.2.1.2. Movie Reviews Reviews Dataset

The data were collected manually. The data were obtained from the www.sinemalar.com and www.beyazperde.com sites. These sites were chosen because they are popular in Turkey and contain enough comments. While for positive data movies with high IMDb (Internet Movie Database) (www.imdb.com) scores were used, negative data were obtained from low-score movies. 919 positive and 615 negative comments were collected as training data. A total of 50 comments was collected for the test data, with 25 positive and 25 negative comments. (IJSS, 2020)

Data information is shown in Table 3. In Table 4, the first collection of the data is given in an excel line. The comments along with the title of the film, the correction of some spelling mistakes in these comments as well as the words that have a positive or negative effect along with the comment's class information were maintained. (ICMS XXIII Proceedings, 2020)

Table 3: Training and Test data information

Comment Type	Total	Training	Test
Positive	919	894	25
Negative	615	590	25

Table 4: Examples from positive and negative training data.

Film title	Original comment	Corrected and translated comment	Important words of comment	Class
Başlangıç *Inception*	Geleceğin **kült** filmlerinden biridir. **Muhakkak izlemek lazım**, zihinsel arşivde bulundurmak lazım. Oyunculuklar, senaryo **harika**ydı. İlla ki eksik bir şey söylemek gerekirse son sahnelerin fazla uzatıldığını söyleyebilirim, zaten 148 dakika bir film bence (istisnai durumlar dışında) çoktur**.** **Başarılı** bir film. **Mutlaka** izleyin derim...	*One of the **cult** films of the future. It is a **definitely must watch** and should be kept in mental archive. The acting, the script was **wonderful**. If I need to say something negative, I can say the final scenes were too long, I think 148 minutes (apart from exceptional cases) is too much for a film. A **successful** film. You should **absolutely** watch it...*	kült *(cult)*, muhakkak izlemek lazım *(definitely must watch)*, harika *(wonderful)*, başarılı *(successful)*, mutlaka *(absolutely)*	Pos
Lucy *Lucy*	Cidden çok **değişik** bir film. Ve bence biraz da **saçma** olmuş.	*Seriously, it is a very **different** movie. Plus, a little **absurd**, I think.*	değişik *(different)*, saçma *(absurd)*	Neg

2.2.1.3. Hospital healthcare Reviews Dataset

One part of this study is based on the patient comments rating on the health care they were provided in a private hospital, chosen as a pilot casestudy in health care area.

Numerous comments of satisfaction or complaints toward doctors, nurses or other hospital personnel, as well as health service quality, personnel attitude towards them, hygiene, price, etc. were gathered during the last 7 years in a private hospital. (CMBEBIH'17)

The data were selected out of 2018 positive comments and 1394 negative comments. Since the number of data is large enough, in order not to create huge datasets, which could be too much time consuming for the system, they were divided into four datasets of 500 positive and 500 negative comments, then the averages of all of the datasets were evaluated. Out of each dataset 450 of the positive and the same amount of negative comments were used for training, while 50 positive and 50 negative comments were randomly divided to be used for testing. Information about data is shown in Table 1. Each comment was stored into a separate text file which was further processed before applying the feature selection methods and ML algorithms, as will be shown in detail in the following sections. Another dataset, Dataset 5, was build again of 500 reviews for each class, but the number of the average words per comments were chosen to be similar, given that the negative comments tend to be much longer and detailed than the positive ones. Information on datasets is explained further in Table 5, Table 6 and Table 7.

Table 5. Healthcare Training and Test data information

Collected comments	**Total comments**	**Train comments**	**Test comments**	**Class**
	500	335	165	Positive
	500	335	165	Negative

Table 6: Examples from Healthcare Negative data.

ID	**Thanking**	**Date**	**Explanation**
1	.	…	İlginiz ve güler yüzünüz için teşekkür ederiz. *Thank you for yor care and smiley manners.*
2		…	Prof. Dr. Xxx Xxx’e ilgisinden dolayı teşekkür ederim. *I thank Prof. Dr. Xxx Xxx for his care.*
3	Xxx Xxx	…	Temizlik hizmetleri gayet başarılı, tüm personelin güler yüzlülüğü dikkatlerden kaçmayacak gibi. *The cleaning service was quite effective, the smiley mood of all the staff cannot be ignored.*

Table 7: Examples from Healthcare Negative data.

ID	Category	To Whom	Date	Mode	Name Surname	Topic:	Explanation
1	Hasta Hizmetleri *Patient Service*	Xxx Xxx	…	Şahsen *(Perso-nally)*	PR: 5511xx/ Xxx Xxx	Hizmet Kalitesi / İlgi Bilgilendirme *Service Quality / Interest & Informing*	Hastamız … tarihinde Dr. Xxx Xxx 'e ilgisizlikten. *Our patient complains for the negligence of Dr. Xxx Xxx during her visit on ...)*
2	Hemsireler *Nurses*	Xxx Xxx	…	İnternet *Internet*	Xxx Xxx	Hemşire Şikayeti/ Kadın Doğum *Nurse Complaint/ Birth Unit*	… hemşireyi çağırdım … *… I called the nurse …*
3	Bekleme Sureleri *Waiting Time*	Xxx Xxx	…	Anket *Survey*	Xxx Xxx	Bekleme Süresi/ Radyoloji *Waiting Time / Radiology*	Radyoloji Bölümü ... *Radiology Unit…*

2.2.1.4. Patients Anamnesis Dataset

The data are obtained anonymously from all departments of a private hospital, however only the internal medicine department data that consist of 1000 entries are utilized as a pilot case study. In this hospital, 1440 active tests exist and only 750 of them were requested in the previous year and 200 of them are requested daily in general. The laboratory secretariat and the medical expert need to spend their valuable time for the selection of the appropriate tests as shown in Figure 21 and Figure 22. In the context of this study the 20 most important test are considered. The data are first collected according to the ICD-10 diagnosis system, however the results of these inputs are unsatisfactory and the input is changed to the free text of symptoms. The most frequent 20 tests data were collected from 1000 patients, as can be shown in Table 8. The typical entry of some partial data is provided in Table 9. (CMBEBIH 2017)

Table 8. Patients Anamnesis Training and Test data information

Tests	Positive Negative	Training (90%)	Test (10%)	Total Pos+Neg
T01	PosT01 = 893 NegT01 = 107	0.9xPosT01 0.9x(1000-PosT01)	0.1xPosT01 0.1x(1000-PosT01)	1000
T02	PosT02 = 646 NegT02 = 354	0.9xPosT02 0.9x(1000-PosT02)	0.1xPosT02 0.1x(1000-PosT02)	1000
T…	…	…	…	1000
T20	PosT20 = 123 NegT20 = 877	0.9xPosT20 0.9x(1000-PosT20)	0.1xPosT20 0.1x(1000-PosT20)	1000

Table 9. Patients Anamnesis Data

Age	G	ID	Symptoms	Tests						
				T 01	**T 02**	**T 03**	**…**	**T 18**	**T 19**	**T 20**
31-40	E	441	halsizlik yorgunluk kilo kaybı midede yanma kabızlık *weakness fatigue weight loss heartburn constipation*	P	P	P		N	N	N
31-40	B	447	hipotroidi için levotiron 100mcg 1*1 alıyor sırt ağrısı için skolyoz için osteocare 2*1 alıyormuş. oroferon dr 1*1 alıyormuş. *takes 100 mcg 1*1 levotiron for hypothyroid takes osteocare 2*1 for back pain scoliosis takes oroferon dr 1*1*	P	N	N		N	N	N
31-40	E	501	ayaklarda yanma *feet burning feeling*	P	P	P		N	N	N
11-20	B	542	2-3 gündür yorgunluk var sabahları uyku hali var ellerde ayaklarda titremeleri oluyormuş midede 1 senedir ağrı varmış. batın usg normalmiş *last 2-3 days fatigue in the morning drowsiness trembling in the hands last one year has stomach pain. Batin usg normal*	P	P	P		P	N	N

2.3. Proposed Methodology for Data Processing

2.3.1. Data Preprocessing

The data were transcribed. As word roots were to be used, morphological analysis was necessary. Special software was used for this. The data were transferred to the appropriate format for morphological analysis by means of C program.

This section describes the modifications conducted on the extracted data.

2.3.1.1. Stemming

In order not to swell too much the words set, their stems are used to reduce storage, thus improve performance of operations. Stemming is very important in SA. (Lovins, 1968) defined stemming as “a procedure to reduce all words with the same stem to a common form, usually by stripping each word of its derivational and inflectional suffixes”.

Stemming is language dependent. The process ignores words meanings, leading to errors. “Light” inflectional suffix-stripping stemming includes only plural (‘-s’), or tense suffix (‘-ed’, ‘-ing’) removal, while more sophisticated stemming removes also derivational suffix and prefixes (such as ‘-ment’, ‘-ably’, ‘-ship’, ‘pre-‘, etc.). However, under-stemming should be avoided to converge to same meaning roots as much as possible, as well as over-stemming should (organization should be differentiated from organ, create from creation, as (Porter, 1980)’s stemmer does.

In many languages, several adjacent, space or hyphen separated units or even words come together to compose more complex grammatical structures. Turkish is a very agglutinative language, similarly to Swahili, Quechua, and most Altaic languages

(Indurkhya & Damerau, 2010). An example to this can be shown the single Turkish word çöplüklerimizdekilerdenmiydi (çöp-lük-ler-i-miz-de-ki-ler-den-mi-y-di) meaning "was it from those that were in our garbage cans?" (Hankamer, 1986).

Similar agglutinative aspects may be found in other languages such as Finnish, Turkish, where the noun "ev" (house) may take on the form "evler" (the houses), "evlerim" (my houses), and "evlerimde" (in my houses). For these two languages at least, the automatic removing of suffixes does not present a real and complex task (Solak, 1994).

Many problems with stemming occur because of misspelled or mistyped words. This is very common in informal reviews from social media, where people tend to write in abbreviated way, without correcting the mistakes, and what even more often tendency is, they omit the characters specific to Turkish replacing them witht their English alphabet corresponding ones. A list of Turkish alphabet specific letters and their English alphabet correspondents are as follows: **ç → c, ğ → g, ı → i, ö → o, ş → s, ü → u,** knowing that all the replacing characters (**c, g, i, o, s** and **u**) are also part of Turkish alphabet, making it very difficult to automatically correct these. This spoils a lot all the stemming process, not finding the correct stems and largely affects all the classification results.

A correct stemming contributes in the consequent important steps, which are Morphological Analysis and Disambiguation

2.3.1.2. Morphological Analysis

In linguistics, morphology is the identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes, part of speeches, intonations and stresses, or implied context.

Morphological Analysis (MA) does not accept text data in the format shown in natural textual format and extracted data should be converted to the appropriate form. One important issue is deciding or knowing where a sentence starts or ends. In social media, people do not often use punctuations or obey the grammar rules, making this task difficult. It is assumed that every '.' indicates the end of a sentence at the risk of some minor errors to overcome this issue and for simplicity. "<S>" and "</S>" tags annotate the start and end of a sentence. The morphological analysis input and output formats for movie dataset can be shown in Table 10.

Table 10: Morphological analysis input and output formats for the sentence: "Cidden çok değişik bir film. Ve bence biraz da saçma olmuş" (*Seriously, it is a very different movie. Plus, a little absurd, I think*)

Morphological analysis input	Morphological analysis output
<DOC>	<DOC> <DOC>
<TITLE>	<TITLE> <TITLE>
<S>	<S> <S>
xwqLucyxwq	xwqLucyxwq xwqLucyxwq *UNKNOWN*
</S>	</S> </S>
</TITLE>	</TITLE> </TITLE>
<S>	<S> <S>
Cidden	Cidden (*seriously*) cidden +Adverb
çok	çok (*many*) çok +Det çok (*very*) çok +Adverb çok (*a lot of*) çok +Adj çok (*ample*) çok +Postp+PCAbl
değişik	değişik (*different*) değişik +Adj
bir	bir (*a*) bir +Det bir (*only*) bir +Adverb bir (*another*) bir +Adj bir (*one*) bir +Num+Card

film . </S> <S>	film (*movie*) film +Noun+A3sg+Pnon+Nom . +Punc </S> </S> <S> <S>
Ve	Ve (*and*) ve +Conj
bence	bence (*mine*) ben +Noun+A3sg+Pnon+Equ bence (*as for me*) ben +Pron+Pers+A1sg+Pnon+Equ
biraz	biraz (*a little*) biraz +Adverb biraz (*a bit*) biraz +Adj
da	da (*and*) da +Conj
saçma	saçma (*buckshot*) saçma +Noun+A3sg+Pnon+Nom saçma (*nonsense*) saçma +Adj saçma (*don't sparkle*) saç +Verb+Neg+Imp+A2sg saçma (*irradiate*) saç +Verb+Pos^DB+Noun+Inf2+A3sg +Pnon+Nom
olmuş	olmuş (*has come out*) ol +Verb+Pos+Narr+A3sg olmuş (*done*) ol +Verb+Pos+Narr^DB+Adj+Zero
</S> </DOC>	</S> </S> </DOC> </DOC>

2.3.1.3. Morphological Disambiguation

The main issue araising from the whole MA process is the multiple results obtained. Another process to get rid of the irrelevant morphological resolutions. A morphological disambiguator (MD) tool is used to find the optimal analysis output for a sentence. MD is an NLP tool that examines the linguistic context of phrases to determine the closest and optimal morphological result matching the other parts of the sentence. This is a tough process of recursive disambiguity, that is, you try to reduce the redundant resolution by making first a standing guess about the adjacent parts of speech, and especially in Turkish where close to half of the words have a disambiguous set of morphological outputs. This makes the whole process a difficulat and time-consuming task. The MD tool for Turkish used in this study claims a success of 96% (Yuret, D. & Türe, F, 2006)

A study (Bekir & Karaoğlan, 2003) showed that Turkish had a compression rate of 84.6% after stemming with (Oflazer, 1993)'s Morphological Analyzer and Disambiguator, compared to 36.4% compression after (Porter, 1980)'s stemmer in datasets containing 376,187 and 567,574 initial words, resulting in 41,370 and 18,384 distinct terms, and 6,363 and 11,671 distinct stems, respectively. Although the disambiguation in Turkish is reported up to an extent of 96%, there are some problems. Some examples are shown as following:

> **Yazıcı** ilk olarak kartuşlu yazıcıdan bıktığım için almıştım. Sürekli baskı yapmam gerekn bir işim var. İlk aldığımda 1000 sayfa hatta 1002 sayfa siyahbeyaz ve renkli baskı yaptım 3 gün sonra toneri **yenilemem** gerekti ama son baskı ile ilk baskı arasında fark neredeyse yoktu bence mükemmel ötesi bir ürün masa üstü **bilgisayarıma** bağladım ağ üzerinden **laptopumla** istediğim yerden bağlanıyorum süper mi? bence çok süper. Tavsiye konusuna gelince ilk arkadaşıma tavsiye ettim ilk bin sayfayı bastığım 3 gün içerisinde ardından bir okula tavsiye ettim toplamda 11 tane aynı yazıcıdan aldırdım ve arkadaşlarım harika diyor.
>
> *I bought the* ***printer*** *firstly because I was tired of the cartridge printer. I have a job where I have to constantly print. When I first bought it, I printed 1000 pages or even 1002 pages in black and white and color. I had to renew the toner after 3 days, but there was almost no difference between the last print and the first print. I think it is a product beyond perfect. I connected it* ***to my*** *dersktop* ***computer*** *through network I connect it* ***to my laptop*** *from anywhere is it super? I think it's really super. As for the*

recommendation, I recommended it first to my friend, within the 3 days I printed the first thousand pages, and then I recommended it to a school I had 11 people buy from the same printer in total and my friends say it's great.

Over-stemming examples:

yazıcı **yazı**+Noun+A3sg+Pnon+Nom^DB+Noun+Agt+A3sg+Pnon+Nom

yenilemem **ye**+Verb^DB+Verb+Pass^DB+Verb+Able+Neg+Aor+A1sg

bağlanıyorum **bağ**+Noun+A3sg+Pnon+Nom^DB+Verb+Acquire+Pos+Prog1+A1sg

yazıcı means ***printer***, while **yazı** means a ***written thing***, so the meaning changes.

yenilemem should have been stemmed as **yenileme**-m (my *renewing, my getting a new one* as the noun ***renewing*** deriving from verb stem ***renew***), and not as any ot the following: **yenile-me**-m (the negative form of the verb ***renew*** as ***I don't renew***), or **yeni**-le-me-m (the noun *my renewing* deriving from adjective stem ***new, renew*, *renewing, my renewing***), or further **yen**-il-e-me-m (*I cannot be beaten-in a game*),.**ye**-nil-e-me-m (*I cannot be **eat**en*), yielding a totally different meaning from the verb ***eat*** = **ye.**

Similarily, bağlanıyorum (bağla-n-ı-yor-um) meaning *I connect, I get connected* should have been stemmed as the verb bağla(mak), or shortly *(to) connect* rather than the noun stem bağ in bağ -la (as in bağ-la-n-ı-yor-um) meaning with a *link*, *lace*, *bond, vineyard*, etc.

Correct-stemming examples:

bilgisayarıma **bilgisayar**+Noun+A3sg+P1sg+Dat ardından **ardından**+Adverb

ardından should not have mistakenly be stemmed as **art**-ı-n-dan where **art** is a noun meaning ***back, rear, end, sequel***, and **ardından** is an adverb meaning ***after, subsequently***.

bilgisayarıma = **bilgisayar**+ım+a (*to my computer*), which should not mistakenly yield the stem **bilgi** (***knowledge***) as **bilgi**+**sayar**+ım+a (standing for to ***my knowledge counter***)

Insufficient-stemming example:

laptopumla **laptopum**+Noun+A3sg+Pnon+Ins

laptopumla → has become laptopum-la (*with + my laptop*), which can be further divided as laptop-um-la (*with+my+laptop*) to yield the stem laptop.

Josh **oyunculuğu** ve **yakışıklılığı** mimikleri ve **tatlılığı** herşeyiyle filme müthiş uyum sağlıyor. *Josh's **acting**, **handsomennes**, mimics and **sweetness** is in wonderful harmony with the movie*.

The stemmings:

- oyunculuğu **oyun**+Noun+A3sg+Pnon+Nom^DB+Noun+Agt+A3sg+Pnon+Nom^DB+Noun+Ness+A3sg+P3sg+Nom

  Here, **oyun** is ***play/act*** instead of **oyunculuk** →***acting***, but the system at least didn't select any of the following as stem candidates as **o**→***he/she/it*** or **oy**→***vote*** or **oyuncu**→***player/actor***, either the correct one **oyunculuk**→***acting***.

- yakışıklılığı **yakışık**+Noun+A3sg+Pnon+Nom^DB+Adj+With^DB+Noun+Ness+A3sg+P3sg+Nom

  Here, **yakışık**→ ***suitability*** was selected instead of **yakışıklılık**→ ***handsomeness,*** or:

- tatlılığı **tatlı**+Adj^DB+Noun+Ness+A3sg+P3sg+Nom

  where: **tatlı** →*sweet* was elected rather than **tatlılık**→*sweetness*

All of the above problems arise because of mistaken decisions taken from the disambiguation algorithm, even though the text is correctly entered and disregarding the contextual settings, but this is to-be-expected in a mechanical process. Moreover, if mistakes persist similiarly, we can get considerate same stems, which, even though carrying the wrong meaning, counts the same in building up the frequence of used words, resulting in additional significant features for machine learning process.

A sample of the disambiguity input and output for movie reviews dataset is shown in Table 11, and for customer reviews dataset can be found in Table 38

Table 11: Morphological and disambiguity analysis outputs for Movie Reviews Data for the sentence: "Cidden çok değişik bir film. Ve bence biraz da saçma olmuş" *(Seriously, it is a very different movie. Plus, a little absurd, I think)*

Morphological Analysis Output – Disambiguity Analysis Input	**Disambiguity Analysis Output**
<DOC> <DOC> <TITLE> <TITLE> <S> <S> xwq**Lucy**xwq xwq**Lucy**xwq*UNKNOWN* </S> </S> </TITLE> </TITLE>	<DOC> <DOC> <TITLE> <TITLE> <S> <S> xwq**Lucy**xwqxwq**Lucy**xwq*UNKNOWN* </S> </S> </TITLE> </TITLE>
<S> <S> **Cidden (*seriously*) cidden +Adverb**	<S> <S> Cidden cidden +Adverb
çok (many) çok +Det **çok (*very*) çok +Adverb** çok (*a lot of*) çok +Adj çok (*ample*) çok +Postp+PCAbl	çok çok +Adverb
değişik (*different*) değişik +Adj	değişik değişik+Adj
bir (*a*) bir +Det bir (*only*) bir +Adverb	bir bir +Det

bir (*another*) bir +Adj bir (*one*) bir +Num+Card	
film (*movie*) film+Noun+A3sg+Pnon+Nom	film film +Noun+A3sg+Pnon+Nom
. +Punc	+Punc
</S> </S>	</S> </S>
<S> <S>	<S> <S>
Ve (*and*) ve +Conj	Ve ve +Conj
bence (*mine*) ben +Noun+A3sg+Pnon+Equ **bence (*as for me*) ben +Pron+Pers+A1sg+Pnon+Equ**	bence ben +Pron+Pers+A1sg+Pnon+Equ
biraz (*a little*) biraz +Adverb **biraz (*a bit*) biraz +Adj**	biraz biraz +Adj
da (*and*) da +Conj	da da +Conj
saçma (*buckshot*) saçma +Noun+A3sg+Pnon+Nom saçma (*nonsense*) saçma +Adj saçma (*don't sparkle*) saç+Verb+Neg+Imp +A2sg saçma (*irradiate*) saç +Verb+Pos^DB +Noun+Inf2+A3sg+Pnon+Nom	saçma saçma +Noun+A3sg+Pnon+Nom
olmuş (*has come out*) ol+Verb+Pos+Narr +A3sg olmuş (*done*) ol +Verb+Pos+Narr^DB +Adj+Zero	olmuş ol +Verb+Pos+Narr+A3sg
</S> </S>	</S> </S>
</DOC> </DOC>	</DOC> </DOC>

2.4. Features Preparing

Feature selection is one of the most important step in text classification. But before selecting the features set, it is preferred to consider some semantic rules in order to make use of language related resources.

2.4.1. Determining Some Rules on Data

Negativity indicating words and suffixes and amount indicating words and their effect in to related words have been determined in the customer product dataset, as following:

2.4.1.1. Determining Negativity

Negativity indicating **words** (negates the words before it)

değil	*(not)*
ne … ne…	*(nor… neither)*

Negativity indicating **suffixes**

-me/-ma	*(not)*
-mez/-maz	*(not)*
-meden/-madan	*(without)*
-sız/-siz	*(without something)*
-mı…-ki/-mu –ki	*(is it…)*
-meksizin/-maksızın	*(without …ing)*

2.4.1.2. Determining Amount

Amount indicating words

çok	*(very)*
epeyce	*(much)*
fazla	*(too much)*
pek	*(enough)*
az	*(a bit)*
biraz	*(a little)*

The information types about the words in the sentence: “Bu ürünü çok sevmedim (*I didn’t like this product much*)” before and after applying negativity and strengthening rules is shown in Table 12.

Table 12: The information types about the words in the sentence “Bu ürünü çok sevmedim (*I didn’t like this product much*)” before and after applying negativity and strengthening rules

			PosNeg		**Strength**	
			Before	**After**	**Before**	**After**
Bu	Bu	bu	true	true	1	1
ürünü	ürün	ürünü	true	true	1	1
Çok	çok	çok	true	true	1	1
~~sevmedim~~	~~sev~~	~~sevmedim~~	~~true~~	~~false~~	~~1~~	~~2~~
	sev-			true	~~1~~	2

2.4.2. Part of Speech and Mode Information

Eight different feature sets were features were used in the movie dataset. Before the classification all important words or group of words were extracted for each comment. Half of the methods used all the words as features, the other half made use of the important words in determining the class. Another important decision is using the mode of the verbs which can be easily extracted from the morphological analysis of words, such that whenever imperative, necessity, optative moods of verbs or words deriving from verbs are used in the text, then new features are created. The same happens also for words used in negation. Both unigrams and bigrams were used. The feature sets are as following:

Single word root and Positive/Negative aspect of **All** words

severim *(I like) =>* ***sev****Pos*

sevmem *(I don't like) =>* ***sev****Neg*.

Single word root, Positive/Negative aspect and **Modes** of **All** words

imperative, necessity modes:

seyredin *(watch it!)=>* ***watch****PosImp*,

seyretmemelisiniz *(you shouldn't watch) =>* ***watch****NegNeces*.

Binary word root and Positive/Negative aspect of **All** words

bu filmi sevdim *(I liked this movie)*

=> ***bu****Pos****Film****Pos* and ***film****Pos****Sev****Pos*.

Binary word root, Positive/Negative aspect and **Modes** of **All** words

bu filmi seyretmeyin *(don't watch this movie)*

=> ***bu****PosNoMode****Film****PosNoMode* and ***film****PosNoMode****Seyret****NegImp*

Single word root and Positive/Negative formation of **Important** words

Single word root, Positive/Negative aspect and **Mode** of **Important** words

Binary word root and Positive/Negative aspect of **Important** words

Binary word root, Positive/Negative aspect and **Mode** of **Important** words

2.4.3. The dictionary method

Another method developed with the movie dataset includes the classification with the help of a dictionary extracted from the important words and phrases only in positive or negative reviews. Accordingly, we made a simple score calculation for test reviews. If the words in the test review appear in the positive words dictionary, we calculated its positive score as the word's rate of appearance in the positive word dictionary. We made the same calculation for the negative words, and assigned the highest scores as the result. The results of this method were also compared to the results of different versions of Naïve Bayes classifiers used for this method.

2.4.4. Organization of the Experiments

2.4.4.1. Features Selection

There are four main **features selection methods** used in this study, namely Words List, Words Vector, Nodes Vector and Hybrid Nodes Vector. The concepts words, terms, features, stems or roots can be used interchangeably, since they mean the same. However the term features is used mainly, together with words.

Each of the methods includes the **binary** and **frequency** versions, each of them including or not some **threshing** ratio. For each of the methods it is possible to attain the **unigram** and **bigram** results, which are obtained for **Term Frequency** (TF), **Term Frequency – Inversed Document Frequency** (TF-IDF) and for **Weighted Log Likelihood Ratio** (WLLR) models separately.

The tf-idf model is preferred, because it is a numerical value showing how critical a word is to a document in the dataset. It is frequently utilized as a weighted factor. The characteristic of tf-idf is that while increasing proportionally with the repeated recurrence of a word showing up in the document, it is on the other hand counterbalanced by the recurrence of the word within the dataset. This is important to differentiate important words from general terms, which tend to have high frequency (Wang_2017).

Weighted Log Likelihood Ratio (WLLR) (Nigam, Mccallum, Thrun, & Mitchell, 2000) is a supervised method finding the weight of a term according to the following formula:

$$\boldsymbol{WLLR(w_t\,, c_j) \;=\; P(w_t \mid c_j) \;\times\; log(\frac{P(w_t \mid c_j)}{P(w_t \mid \neg c_j)})} \qquad \textbf{\textit{Equation 1}}$$

where;

w_t is the t_{th} feature whose scores is being calculated, c_j is the $j_t h$ emotion class,

P$(w_t \mid c_j)$ is number of appearances of wt in c_j divided by number of all features in c_j,

P$(w_t \mid \neg c_j)$ is number of appearances of w_t in $\neg c_j$ divided by number of all features in $\neg c_j$

By using the equation above, a feature in class j will have a high rank, if it appears frequently in class c and infrequently in classes other than c. So, the features that are the most distinctive ones can be extracted and these features would best represent data in the vector space.

Firstly, for each dataset, positive and negative lists of terms and their respective frequencies are prepared from respective class documents. The lists of all used terms together with their frequencies build domain dependent dictionaries polarized according to the classes of the text categorization. These are further used to determine the features sets.

The following feature selection approaches are used:

Words List method: All word roots coming from positive and negative comments and their frequencies contribute respectively to the calculated positive value and negative value of a document. The highest value determines the class of the document.

Words Binary List feature selection method: Initially all words (roots) are considered as candidate features. According to which thresh ratio and erase method is going to be used, the positive words list and negative words list are built. If the word used in the positive comment exists in the positive words list, then it contributes to the positivity of the comment as much as the frequency of the word used inside the same comment. The same is done for the negativity of the word from the positive comment, if the word appears in the negative word list for the given thresh and erase method, the frequency of appearance in the positive document is added to the negativity value of the comment. This is done for all the words of the comment. This process is repeated for all positive comments in the train set. The whole process is done also for all negative data from the train set. At the end of the process, a comment will have a positive and a negative value which will be the estimated class value (P for positive, N for negative determined from the difference of the negative value from the positive value calculated). Again all the above is repeated for the test data set, for positive and negative comments.

Words Frequency List feature selection method: The difference from Words Binary List method is that each word value occurrence in the document is multiplied by the frequency coming from the positive and negative words list of words. This calculation will affect the total positivity and negativity value of each comment.

Words Vector method: all words from positive and negative training comments and their correspondent frequencies occurring in each document are visualized in a matrix, together with their class information.

Words Binary Vector method: All distinct words coming from train documents become the first row of the vector, while all the documents (id's or names) coming from the train dataset become the first column of the vector. If a word appears in the comment, its value is placed as 1, if it doesn't, its value equals 0. The last column contains the information for positivity or negativity of the comment. In the same way also the test data file is prepared, with the exact same first row. For words appearing in test dataset but not in train dataset, no value will be kept, obviously.

Words Frequency Vector feature selection method: The only difference from Words Binary Vector method is that not only the occurrence is determined, but also a numeric value is calculated regarding the number of occurrences of the word in the document, the frequency coming from the positive words list and the frequency coming from the negative words list.

Selected Words Vector method: selected words after erasing some of the features from one or both lists of the positive and negative training comments and their correspondent frequencies occurring in each document are visualized in a matrix, together with their class information.

Selected Words Binary Vector feature selection method: Instead of taking all the words out of the train data test, according to the thresh value and erase method, a set of those words is taken as the feature set of the vector. The rest is done similarly to the Words Binary Vectore feature selection method.

Selected Words Frequency Vector feature selection method: Instead of taking all the words out of the train data test, according to the thresh value and erase method, a set of those words is taken as the first row of the matrix. The rest is done similarly to the Words Frequency Vector feature selection method.

Nodes Vector method: the nodes resulted from the J48 ML classification of the Words Frequency Vector method are the new features of a similar matrix as in the Words Frequency Vector method.

Hybrid Nodes Vector method: the new matrix has all features that Words Vector method has, with the difference that for non-node features only binary information is kept, while for nodes the respective frequency induced from the branching level of J48 tree is kept for each appearance of the words in each document. This method is a fusion of Words Binary Vector method applied for non-node features and Words Frequency Vector method applied for node features.

A representation of the features binary vector for **n features** and **m documents** is shown in Table 13, where D_i stands for the i'th document name, $1<=i<=m$, F_j stands for the j'th feature, $1<=j<=n$.

Table 14 shows a representation of the features frequency vector for the same set, while Table 15 and Table 16 show the situation when **k nodes** are obtained from the J48 decision tree, and N_x stands for the x'th node, such that $1<=x<=k<n$.

Table 13: Features Binary Vector representation

	F_1	F_2	F_3	...	F_j	...	F_n	Class
D_1	0	0	1	...	0	...	1	Pos
D_2	1	0	0	...	1	...	0	Neg
...	...	...	...	...	...	...	...	...
D_i	1	0	1	...	1	...	0	Pos
...	...	...	...	...	...	...	...	...
D_m	0	0	0	...	0	...	1	Pos

Table 14: Features Frequency Vector representation

	F_1	F_2	F_3	...	F_j	...	F_n	Class
D_1	0	0	2	...	0	...	1	Pos
D_2	1	0	0	...	5	...	0	Neg
...	...	...	...	...	...	...	...	...
D_i	2	0	1	...	4	...	0	Pos
...	...	...	...	...	...	...	...	...
D_m	0	0	0	...	0	...	3	Pos

Table 15: Nodes Vector representation

	N_1	N_2	...	N_x	...	N_k	Class
D_1	4	0	...	0	...	7	Pos
D_2	4	5	...	12	...	0	Neg
...	...	...	...	...	...	...	...
D_i	4	5	...	12	...	7	Pos
...	...	...	...	...	...	...	...
D_m	0	0	...	0	...	7	Pos

Table 16: Hybrid Nodes Vector representation

	F_1	F_2	F_3	...	$F_j(N_x)$	...	F_n	Class
D_1	0	0	1	...	0	...	1	Pos
D_2	1	0	0	...	12	...	0	Neg
...	...	...	...	...	...	...	...	...
D_i	1	0	1	...	12	...	0	Pos
...	...	...	...	...	...	...	...	...
D_m	0	0	0	...	0	...	1	Pos

As it is clearly seen, the Hybrid Nodes Vector method differs from the respective Words Vector ones in emphasizing the importance of the nodes frequencies, while disregarding the other non-node words ones, just keeping their binary information.

2.4.5. Features Erasing

Once all the stems are determined, feature selection task is done, being one of the most important ones in SA procedure. All the stems will be used as the initial set of the features for various methods. After the unique stems and their respective frequencies of their occurrence are specified for all individual documents and all the dataset as a whole, it is easy to combine them and form the positive and negative lists of words used, according to the source the comments were gathered from. This process involves some normalization related to the length of texts and the overall documents number in each class. The obtained positive and negative lists of words together with their frequencies provide the basic information needed for effective feature selection, followed by machine learning classification. Although of commonly highest occurrences, most of the times the top words of each list fail from being discriminative, because they are mutual for both classes. Pronouns, articles, auxiliary verbs and other grammar based words can be an example for this category. According to one of the approaches, the common terms are reduced from both lists, or from just one of them, adjusting the value of that term in the other list accordingly, leaving thus more robust feature values. The elimination process is done according to some thresh values. Following some preliminary concepts, each of the methods and some of the data clearing techniques together with their parameters will be defined.

Erase methods: Some terms which fall between determined thresh values are erased from one or both lists, according to the erase method.

Erase from both lists erase method: The terms of similar frequencies whose values appear in each of the lists fall within given thresh percentage are erased from both the positive and negative lists.

Erase from one list erase method: The terms within thresh percentage of similar frequencies are erased from the list that they appear less often, while their frequencies difference is kept in the other respective list.

2.4.6. Determining Threshing

Thresh values: Some of the words appearing in similar ratio in both positive and negative word lists have been eliminated from both lists according to this ratio value.

2.4.6.1. Frequency Threshing

In some studies a number of mostly used terms is determined according to the respective frequency of the term appearing in each of the polarity lists, keeping only this certain number of terms and disregarding all the remaining ones. This can be done by using any of the abovementioned erase methods. Suppose most frequent 200 terms will be kept for each class. One way is to get rid of all other terms lower than 200'th position in the respective list. This can leave many terms that are frequently used, although they are not class determining, since most probably they are common for both classes. This method is not preferred since we loose much important information and get vwry low results. Moreover, determining a fixed number of features to be left is not a good idea, because this number should be in proportion with the overall amount of initial terms.

2.4.6.2. Percentage Threshing

Percentage threshing could be a better proposal, leaving a certain percentage of the overall words and erasing all the others. Besides overcoming the dataset size problem, it will still leave mostly nondiscriminating features.

Erasing values with a frequency similarity criteria is a better solution. After determining some percentages, all the values appearing in both lists less than the determined threshing percentages will be deleted according to the usage of erase from both or erase smallest method respectively from both lists, or from the list where they appear less, leaving it in the opposite list, but with the frequency reduced with the value of the frequency form the list of the term where it was erased from, actually leaving the absolute value of the frequencies differences.

Suppose WPi and WNj are the same word and they appear in both positive and negative lists, and their respective values are N and P. The value will be erased from the list it appears less, and still stay in the other list, its new frequency evaluated as $|N-P|/(N+P)$, ensuring always a positive value for the frequency.

For experimental reasons different values of threshing percentages were applied, such as 5, 25, 50 and 75, from very small to very large values. The result of threshing is measured and discussed in the results and conclusion sections for some of the datasets used in this task.

2.4.6.3. Smart Threshing: Correlation-based Features Seection

Applying treshing according to usage percentage it is not the best solution, because even sparce features can play an important role in text classification. For this reason letting the more sophisticated Machine Learning and Data Science methods decide on the “uninteresting” features removal is much brighter solution. For this reason

Correlation-based feature selection (CFS) is an algorithm which utilizes the filter technique for choosing the attributes according to the principal that “a good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other.” (Hall, 2000). Hall suggests that “CFS evaluates a subset by considering the predictive ability of each one of its features individually and also their degree of redundancy (or correlation)”.

The CFS algorithm applies a heuristic which measures the usefulness of individual features for predicting the class label along with the level of inter-correlation among them. The extremely correlated and unrelated features are avoided. The difference between CFS and other methods is that it provides a “heuristic merit” for a feature subset instead of each feature independently. This means that given a heuristic function, the algorithm can decide on its next moves by selecting the option that maximizes the output of this function. Heuristic functions can also be designed to minimize the cost to the goal.

CFS work on filters or wrappers principles. Filter methods are generally faster than wrappers, but they do not take into account the classifier, which can be a disadvantage.

Ignoring the specific heuristics and biases of the classifier might lower the classification accuracy. CFS-correlation based feature selection method has the advantages of having less computational complexity compared to other method, being at the same time less prone to overfitting. However, this method can be heavily dependent on the model, so it can fail to fit the data well.

Wrappers tend to perform better in selecting features since they take the model hypothesis into account by training and testing in the feature space. This leads to the big disadvantage of wrappers, the computational inefficiency, which is more apparent as the feature space grows. Unlike filters, they can detect feature dependencies.

For calculating the necessary correlations a number of information based measures of association were projected such as: the uncertainty coefficient, the gain ratio or the minimum description length principle. The best results achieved with the gain ratio used for feature-class correlations and symmetrical uncertainty coefficients used for feature inter correlations.

The CFS method was used in this study to reduce the number of features, without diminuishing the overall accuracy.

One of the easiest way of classification improvement is by using feature reduction, which when done effectively can speed up the classification process, while increasing also efficiency (Aliwy & Abdul Ameer, 2017).

2.5. Methodologies of Machine Learning for Classification

2.5.1. Tools Used: Weka

Classification is achieved using ML methods provided by a data mining tool named Weka. Weka provides a collection of many well-known ML algorithms to test and train the classifiers.Weka is the name of software developed for machine learning at the Waikato University. It includes most of the machine learning algorithms and methods that today are widely used (Witten I. H., et al., 1999), (Witten & Frank, 2005). Thanks to the fact that it is developed in the Java language and its libraries come in jar files, it has become more common to integrate it in programs written in the JAVA language. Weka has a completely modular design, and with the features it contains, it can use the data sets to perform operations such as visualization, data analysis, business smart applications and data mining. Weka software comes with an .arff extension support sui generis. However, in the Weka software, there are tools which convert CSV (comma-separated values) files, which allow data to be stored in a tabled structured format, to the ARFF (attribute relation file format) format, which describes a list of instances sharing a set of attributes (Şeker, 2013).

It is possible to make classifications within the existing data sets by using one of the classification algorithms loaded in the WEKA. Moreover, it is possible to use separate sets for testing and validation.

2.5.2. Determining Evaluation Methods

Classification is the process of identifying the new observations whose classes are unknown by using the pre-classified data. In this study different ML classifiers were used:

To measure the program's success, Acc-accuracy, P-Precision, and R-Recall as well as the FM-FMeasure, the harmonic mean for P and R, were used. In addition, an error matrix (CM-Confusion matrix) showing the distribution of evaluation by classes was found.

True Positive (TP): Comments you anticipated positive are actually positive.
True Negative (TN): Comments you anticipated negative are actually negative.
False Positive (FP): Comments you anticipated positive are actually negative.
False Negative (FN): Comments you anticipated negative are actually positive.

For example, we have 50 positive and 50 negative data. Let's assign them a value with a classification system. Suppose that out of the 50 positive, 40 are positive and 10 are negative, and out of the 50 negative, 30 are negative and 20 are positive. In this case, a CM like the one shown in Table 17 will come out:

Table 17: Confusion Matrix

Original	**Assigned**	
	Positive	**Negative**
Positive	40 (TP)	10 (FP)
Negative	20 (FN)	30 (TN)

The total number of elements is *N*=100. While calculating the value of *Acc*, the ratio of all correctly classified samples to all data is checked. *P*, *R*, and *FM* values are calculated separately for each class. While *Pc* belongs to class *c*, it is the ratio of the data found as class c to all the data that belong to class c, that is, *Pc* is the accuracy rate of the values found for *c*. While *Rc* belongs to class *c*, it is the ratio of data found as class *c* to the all data that belong to class *c*, that is, *Rc* shows how much of the class *c* data can be covered or found as class c. The calculations for the case study explained in Table 17 are shown in Table 18. Meanwhile, *FM* is the harmonic mean of *P* and *R* and it is used to prevent systems from giving low *R* results when *P* is high, or low *P* results when *R* is low; it is calculated as *FM=2PR/(P+R).*

Table 18: Positive and Negative Precision and Recall calculations

	Precision	**Recall**
Positive	P_Ol = $\frac{40}{40+20}$	R_Ol = $\frac{40}{40+10}$
Negative	P_Olz = $\frac{30}{30+10}$	R_Olz = $\frac{30}{30+10}$

$$\boldsymbol{FM\_Ol = \frac{2\ x\ P\_Ol\ x\ R\_Ol}{P\_Ol + R\_Ol} = \frac{2\ x\ 0.67\ x\ 0.8}{0.67+0.8} = 0.73}$$ ***Equation 1***

$$\boldsymbol{FM\_Olz = \frac{2\ x\ P\_Olz\ x\ R\_Olz}{P\_Olz + R\_Olz} = \frac{2\ x\ 0.75\ x\ 0.6}{0.75+0.6} = 0.67}$$ ***Equation 2***

$$\boldsymbol{Acc = \frac{TP+TN}{TP+FP+TN+FN} = \frac{40+30}{40+10+30+20} = \frac{70}{100} = 0.7}$$ ***Equation 3***

In this document, P, R, FM and Acc values will be shown in percentages.

Together with the accuracies, the True Positive (TPR) and False Positive Rates (FPR) in percentage for the negative set of data were evaluated for the health care dataset. While TRP is aimed to have the highest values as close as possible to 100, the opposite is true for FPR, ideally close enough to 0. This was done to identify the Feature method – ML algorithm pair with the highest precision in not classifying negative data wrongly because health issues are so delicate that nobody would like to risk not to identify problems already happened to other reviewers, so that they can possibly avoid them

2.5.3. ML Classification Algorithms

For the classification of subjects, some ML algorithms are used. A classification algorithm can be used to build a model that predicts the outcome class for a given dataset. The datasets can come from a variety of domains, as it is the case in this study. Depending upon the dimensionality of the dataset, the attribute types, sparsity, missing values, etc., one algorithm might give better predictive accuracy than most others (Piccini, 2019). In this section the algorithms used through this study will be briefly discussed.

Each of used classifiers, the characteristics, and their elaboration in different studies, together with some strong and weak points of them will be mentioned in the following sections. Diverse methods used in different case studies and methods chosen from disticntive categories were explained, emphasizing the most commonly used ones.

2.5.3.1. Naïve Bayes Method

Naïve Bayes (NB) classifier is one of the supervised learning methods among machine learning methods. Although it is difficult and expensive to prepare the training set for supervised learning methods, the reason for choosing it is the ease and high success of testing with Naïve Bayes.

In the text classification application, words, phrases or their properties are used as features. According to the NB formula, in order to find the class of a document, when that class is given, it is necessary to calculate the likelihood for each term/word, and to multiply it by the prior likelihood of that class.

The Naïve Bayes classifier is a generative probabilistic classification method uses estimator classes under known priori probability and class conditional probability using simple and strong independence assumptions between each pair of features based on Bayes' theorem to efficiently compute the conditional probabilities. This (not so) much naïve assumption can provide a reasonable trade-off between performance and computational costs. The basic idea of Naïve bias is to calculate the probability that document D belongs to class C. The Bayes Theorem says the conditional probability of an outcome can be computed using the conditional probability of the cause of the outcome. Using the prior probability, the posterior probability can be computed - which is the probability that an event belonging to class C will occur given that another event x has occurred. The Naive Bayes classifier uses the input variable to choose the class with the highest posterior probability (Piccini, 2019).

The classifier is obtained with the maximum a posteriori (MAP) decision rule. As some values in the results are very small and can thus not be fully displayed and fit in the memory, they may be rounded to zero, which may invalidate the results of analysis. To avoid this, instead of maximizing the multiplication of probabilities, the method of maximizing the sum of their logarithm is preferred and the formula is applied accordingly. In the new formula, instead of the maximum probability class, the class with the maximum log score is selected. Since the logarithm function is a monotone increasing function, the MAP decision rule remains the same. Another issue that should be considered here is that in a certain class some terms/words might not be mentioned at all. In this case, the value of conditional probability is 0. In order to avoid 0 as a product for inexistent term, since the logarithm 0 is undefined, the Laplace smoothing method is used, adding one to the counting results (Vryniotis, 2013).

Even though there are more successful methods than this, NB is more effective in terms of processing time and memory usage, since is not computationally intensive, and it requires a small amount of training data, resulting in training time significantly smaller as opposed to alternative methods (Vryniotis, 2013). (John & Langley, 1995). Although this assumption may be wrong in many cases, it has been shown that there are many theoretical explanations behind this classifier's unreasonable effectiveness (Zhang, 2004).

Despite its simplicity and extremely simple assumptions, NB has been successfully applied to many complicated real life problems, from e-mail and document classification to sentiment and opinion classification. Even if the probability calculations are of low quality and not very realistic for the selected class, classification decisions are quite good (Manning, Raghavan, & Schütze, 2008), as the main goal is not to calculate real probabilities, but to compare classes with each other and decide, the decision is generally correct and the model works well (Vryniotis, 2013).

NB can also perform well when features are interdependent. In addition, the resulting generative model is easy to interpret and explain. Moreover, NB as a generative classifier may be recommended for smaller sample sizes due to its inherent regularization, making it less likely to overfit compared with discriminative classifiers. However, NB is not capable of modeling interaction effects among features. Thus, it is expected to perform relatively well for problems with strong individual signal words and straightforward relationships between the text features and the respective classes, e.g., for two-class sentiment classification exhibiting strong polarity, spam filtering, document classification or simple forms of promotion content detection, The Naïve Bayes is effective enough to classify the text in many domains, although it is less accurate than other discriminative methods, such as SVM (Wang_2017).

For larger samples sizes, as more data has substantial impact on the ability to identify the different types of expressions contained in a particular dataset more comprehensively, better performances of machine learning methods is expected. NB is suitable in situations with limited processor and memory resources. In addition, since the training is very fast, it is also one of the most suitable choices in applications in which the short training time is important. It is also used as a method for comparing and testing alternatives in many applications.

The distribution of the data assumed by the naïve algorithm can be Gaussian, Bernoulli or Multinomial. Accordingly, there are some types of NB method, Multinomial NB (MNB), Binarized Multinomial NB (BMNB) and Bernoulli NB (BNB) being some of them. They may give different results because they use different models.

Multinomial Naïve Bayes (MNB)

The Multinomial Naive Bayes uses the multinomial distribution, which is the generalization of the binomial distribution. In other words, the multinomial distribution models the probability of rolling a ***k*** sided die ***n*** times.

Multinomial Naive Bayes is used frequently in text analytics because it has a bag of words assumption - which is the position of the words doesn't matter. It also has an independence assumption – supposing that the features are all independent. (Piccini, 2019)

MNB is generally used in cases when more than one appearance of terms is important in the classification problem. Topic classification can be given as an example. The multinomial model is more suitable when database is large. However, there are some serious problems identified with this model. One of them is the problem in handling rare categories that contain only few training documents. Enhancing the method by giving weight to improve the performance of rare categories could be proposed as a solution. Other improvements on performance of text classification could be explored as ways to search the dependencies among attributes.

Binarized (Boolean) Multinomial Naïve Bayes Model (BMNB)

BMNB: Despite being very similar to MNB, instead of looking at the frequency/number of a term/word usage, BMNB this only looks at whether it is used or not. The assumption here is that instead of number of uses of a term, what affects the model's success is its usage or not (Jurafsky, 2018). The training and testing algorithm in MNB remains the same only that instead of the frequency, if the term appears in the text, 1 is used, and if not, 0 is used.

One drawback of BMNB is that continuous features have to be preprocessed and discretized by binning, which can discard useful information. For this reason BMNB is suitable for cases when frequencies do not play an important role in classification. Sentiment analysis is a good field of application for the use of BMNB because in these texts, it is important whether a negative word such as “bad” is used or not, rather than the number of times it is used. Finally, the BMNB method is useful for applications such as spam email or unwanted content detection, where it is important that certain words are not used.

Bernoulli Naïve Bayes Model (BNB)

For each term in the vocabulary, BNB produces a value of 1, if present in the document, if not, it produces 0. It is very different from MNB in that it does not look at the number of uses of a term, but takes into account the values for the terms that are not used. On the other hand, MNB does not consider the values of the terms that are not used (Manning, Raghavan, & Schütze, 2008). For this reason BNB can make a lot of mistakes in correctly categorizing the polarity of the values when it is not suited to the domains, thus achieving a lower accuracy in some cases, because it does not look at the number of uses while classifying long documents. Conversely, it is sensitive to whether unrelated terms are used. The Bernoulli Distribution is appropriate for binary variables, like determining the gender.

Since NB is easy for implementation and computation, it is used for pre-processing, i.e. for vectorization. Performance of NB is very poor when features are highly correlated and, thus is highly sensitive to feature selection. Many improvements were attempted for NB classifier, some of them modifying the probability calculating technique, reducing the number of features, etc. (Aliwy & Abdul Ameer, 2017). Schneider (2005) used uncomplicated transformation to simple modifications of the Naive Bayes text classifier.

Straightforward transformation is working on removing duplicate words effectively in a document. Yuan (2010) optimized Naïve Bayes text classification by “calculating posterior probability and reducing dimension of feature words of text". The results for experiment specified that the enhanced way has higher efficiency than the original algorithm. Singhal & Shanma (2014) optimized the performance of Naive Bayes algorithms by removing the features that are redundantly correlated before giving the dataset to classifier. This optimization is potential through the correlation based feature selection (CFS) algorithm as preprocessing to Naive Bayes classifier for training purpose.

2.5.3.2. Support Vector Machine (SVM)

Sequential Minimum Optimization (SMO) is a classification and regression algorithm that trains a Support Vector Classifier (SVC). A SVC is a classifier that takes a set of data and predicts the membership of each data according to the belonging classes (Platt, 1999). Support Vector Machine (SVM) are discriminative classifiers, which identify a margin-maximizing the n dimensional space, so called the hyperplane, to separate the data into classes. SVM algorithm represents the text document as a vector where the dimension is the number of distinct keywords. If the document size is large then the dimensions of the hyperspace in text classification are enormous, which causes high computational cost. Differently from some other classification methods, the SVM need both positive and negative training set, to seek for the decision surface that best separates the positive from the negative data in hyper plane. They were initially developed as binary linear classifiers (Cortes & Vapnik, 1995), but can be extended to non-linear problems of higher dimensionality through the use of kernels that can accommodate any functional form (Schölkopf & Smola, 2001). Computing the parameters of the margin-maximizing hyperplane can be computationally costly depending on the sample size and number of features (Pang, Lee, & Vaithyanathan, 2002).

Some improvements and modifications can be done to SVM to increase the efficiency, and hence its accuracy. The feature extraction and reduction can be used to reduce the dimensionality. Ageev & Dobrov (2003) analyzed the influence of different parameters for SVM, such as different kernel functions, feature space reduction (about 80%), and relative weight of different errors on performance of text categorization and tuning the strategy depending on subject area, resulting in an increase by 1-5% of the accuracy. Rennie & Rifkin (2001) compared the Support Vector and Naive Bayes Machines in classifying multilayered text. They found that when using the SVM as portion of an Error-Correcting Output Codes (ECOC) scheme (a meta method which combines many binary classifiers in order to solve the multi-class problem), it effected positively the task of classifying multilayered text such as news article categorization and sentiment prediction, as they can deal well with high dimensionality (Aliwy & Abdul Ameer, 2017)

SVM classifier method is outstanding from other with its effectiveness to improve performance of text classification, used as a feature extractor. As a result, a new feature vector is normalized as the input of SVM. The trained SVMs can classify unknown texts successfully, also by combing with Bayes to reduce number of features, thus reducing the number of dimensions. SVM is more capable to solve the multi-label class classification and especially in text classification tasks. SVM is very effective not only in high dimensional spaces, but also in cases whether the number of dimensions (or features/columns) is greater than the number of samples (rows). It is also effective in high dimensions, such as images. It is considerably memory efficient due to its own advantage of kernel mapping to high-dimensional feature spaces (Aliwy & Abdul Ameer, 2017), because it uses a subset of the data to learn support vectors (Piccini, 2019). However the performance of the method is likely to get poorer in this case, which is considered as a disadvantage of SVM (Wang_2017).

At the same time, SVM have been argued to be less prone to overfitting. Therefore, SVM performs similarly to a simple method like NB, but worse than more flexible methods like ANN and RF. (Aliwy & Abdul Ameer, 2017).

2.5.3.3. Decision Tree

A decision tree classification algorithm is an inductive learning method using a labelled training dataset to stratify or segment the predictor space into multiple regions. Each such region has only a subset of the training dataset. To predict the outcome for a given (test) observation, first, we determine which of these regions it belongs to. Once its region is identified, its outcome class is predicted according to all the training observations included in that region.

The rules used to stratify the predictor space can be graphically described in a tree-like flow-chart, hence the name of the algorithm, with the only difference that these decision trees are drawn upside down. A tree can classify the document by running through the query structure from root until it reaches a certain leaf, which represents the goal for the classification of the document. DT is a flowchart such as tree structures, each internal node indicate test on document, each branch acts outcome of the test, and each leaf node holds a class label.

Since most of training data will not fit in memory, the construction of decision tree becomes inefficient due to swapping of training tuples.

ID3 is one of the most well-known decision tree learning algorithms. It has extensions like C4.5 and C5.

J48 is a decision tree algorithm for generating a pruned or un-pruned C4.5 decision tree. It uses formulas based on information theory to evaluate how much good a test extracting the maximum amount of information from a set of cases, given the constraint that only one attribute is tested (Quinlan, 1993).

Decision trees (DT) are widely utilized. When decision tree is used for text classification it consist tree internal node are label by term, branches departing from them are labeled by test on the weight, and leaf node are represent corresponding class labels.

To handle this issue can handle numeric and categorical data. New method is proposing as FDT to handle the multi-label document which reduce cost of induction, and [26] presented decision-tree-based symbolic rule induction system for text categorization which also improves text classification.

Modifications to the learner or the algorithm itself, as well as extraction, selection or reduction of the features can contribute to improving DT.

Vateekul & Kubat (ICDMW'09) aimed on reducing the costs of Imbalanced, Large Scale, and Multi-label Data, where implementing decision trees computations is very difficult and costly. The researchers implemented FDT ("fast decision - tree induction"), as a two parts technique for feature set preselection and induction of several trees, each for a different data subset.

Harrag and friends (NDT'09, 2009) used a decision tree algorithm in classifying Arabic text documents. They suggested hybrid techniques of document frequency threshold by using embedded information gain criterion and the preferable feature selection criterion, where they got accuracies of 0.93 and 0.91 for two different datasets.

Badgujar & Sawant (2016) utilized of L' Hospital Rule which eases the calculation process and improves the efficiency of decision making algorithm, resulting in improved effect on C4.5 than the ID3 in three aspects, such as node count, rule count and time complexity.

Galathiya and friends (2012) compared ID3, C4. 5 and C5.0 and implemented a new classification system by using cross validation, feature selection, reduced error pruning and model complexity, resulting in a faster, more accurate and efficient system, less complex and using lower memory. In another study (2012) they proposed an implementation of C5.0 with feature selection, cross validation, model complexity and reduced error pruning. They aimed to solve over fitting problem and succeeded to reduce the error ratio within less time, by using RGUI with WEKA packages where the input to algorithm is a fixed set of attributes.

Pandya & Pandya (2015) also found C5.0 as more efficient and accurate, using lower memory compared to other DT. They used WEKA packages to implement a version of C5.0, by using cross-validation in the testing, relevant features selection and reduced error pruning technique, and obtained more reliable estimation, gaining 1 to 3% accuracy. (Aliwy & Abdul Ameer, 2017)

To sum up, DT have some advantages and drawbacks.

The decision tree classification method is outstanding from other decision support tools with several advantages like its simplicity in understanding and interpreting, even for non-expert users. Decision tree classification models can easily handle qualitative predictors without the need to create dummy variables. Missing values are not a problem either. Interestingly, decision tree algorithms are used for regression models as well. The same library that you would use to build a classification model, can also be used to build a regression model after changing some of the parameters.Decision trees are capable to learn disjunctive expressions and seem convenient for noisy document classification. (Aliwy & Abdul Ameer, 2017). Although the decision tree-based classification models are very easy to interpret, they are not robust. One major problem with decision trees is their high variance. One small change in the training dataset can give an entirely different decision trees model. Learning of decision tree algorithms cannot guarantee to return the globally optimal decision tree. Another issue is that their predictive accuracy is generally lower than some other classification models, such as “Random Forest” models, for which decision trees are the building blocks. (Piccini, 2019)

2.5.3.4. Voted Perceptron

The Voted Perceptron (VP) method is based on the perceptron algorithm of Rosenblatt Frank (1957). The algorithm takes advantage of data that are linearly separable with large margins.

The Voted Perceptron (Freund & Schapire, 1999), is a variant using multiple weighted perceptrons. The perceptron algorithm starts with an initial zero prediction vector. It predicts the label of a new instance. If this prediction differs from the actual label, the prediction vector is updated. If the prediction is correct, then it is not changed. The process then repeats with the next example. It has been shown that if the data are linearly separable, then the perceptron algorithm will make a finite number of mistakes, and therefore, if repeatedly cycled through the training set, will converge to a vector which correctly classifies all of the examples.

The algorithm starts a new perceptron every time an example is wrongly classified, initializing the weights vector with the final weights of the last perceptron. Each perceptron will also be given another weight corresponding to how many examples do they correctly classify before wrongly classifying one, and at the end the output will be a weighted vote on all perceptrons.

Moreover, the number of mistakes is upper bounded by a function of the gap between the positive and negative examples. In the voted-perceptron algorithm, more information is stored during training respectively to SVM. This elaborate information is used to generate better predictions on the test data. The information maintained during training is the list of all prediction vectors that were generated after each and every mistake. For each such vector, the number of iterations it “survives” until the next mistake is made are counted. This is referred as the “weight” of the prediction vector. To calculate a prediction the binary prediction of each of the prediction vectors is computed and all these predictions are combined by a weighted majority vote. The weights used are the survival times. This makes intuitive sense as “good” prediction vectors tend to survive for a long time and thus have larger weight in the majority vote (Freund & Schapire, 1999).

The perceptron is a linear classifier, therefore it will never get to the state with all the input vectors classified correctly if the training set is not linearly separable, i.e. if the positive examples cannot be separated from the negative examples by a hyperplane. In this case, no "approximate" solution will be gradually approached under the standard learning algorithm, but instead, learning will fail completely.

Although the perceptron initially seemed promising, it was quickly proved that perceptrons could not be trained to recognize many classes of patterns. This caused the field of neural network research to stagnate for many years, before it was recognized that a feedforward neural network with two or more layers (also called a multilayer perceptron) had greater processing power than perceptrons with one layer (also called a single-layer perceptron).

It should be kept in mind, however, that the best classifier is not necessarily that which classifies all the training data perfectly. Indeed, if we had the prior constraint that the data come from equivariant Gaussian distributions, the linear separation in the input space is optimal, and the nonlinear solution is overfitted.

Manabu Sassano (2008) performed an experiment to find out that running the perceptron algorithm in a higher dimensional space using kernel functions produces very significant improvements in performance. He witnessed that while the voted perceptron is comparable to Vapnik's SVM in terms of accuracy, it is simpler to implement and a faster algorithm, better than SVM considering learning time and prediction speed. For these reasons, VP is a strong alternative to SVM in classification tasks in NLP as well as ranking tasks.

2.5.3.5. Random Forest

Random Forest (RF) is an ensemble learning method that grows a multitude of randomized, uncorrelated decision trees, each of which is trained on a random subset of the training data (Breiman, 2001). Ensemble learning methods are meta-algorithms that combine several machine learning methods into a single predictive model to increase the overall performance (Piccini, 2019). To build a random forest, you need to choose the total number of trees and the number of samples for each individual tree. Later, for each tree, the set number of samples with replacement and features are selected to train the decision tree using this data. The larger the number of predictors, the more trees need to be grown for good performance. These trees predictions can then be aggregated to provide a single prediction from a series of predictions. The outputs from all the separate models are aggregated into a single prediction as part of the final model. In terms of regression, the output is simply the average of predicted outcome values. In terms of classification, the category with the highest frequency output is chosen. Each decision tree casts a vote for the class of the test example. The most popular class determines the final prediction of the RF classifier. This procedure is called bagging (Breiman, 1996).

There are different ways to introduce randomness and decorrelate the individual decision trees, e.g., through random feature selection and randomly chosen data subsets (Breiman, 2001). While individual decision trees are prone to overfitting due to their high flexibility, RF overcomes this issue by combining a multitude of decision trees on a heterogeneous randomly drawn subset of variables.

RF is a discriminative classifier. As RF is more robust to noise and outliers (Breiman, 2001), we expect consistently high performance across all social media datasets. Moreover, given their hierarchical structure, RF can learn complex interactions between features, perform automatic feature selection, and model highly non-linear data. This leads to the belief that RF can deal well with both content and more complex sentiment classification, where higher context understanding is required, as signals are subtly embedded in the text and spread across features. The training time of RF increases linearly with the number of decision trees in the ensemble. As each tree is grown individually, processing can be easily parallelized. This makes RF scalable and computationally efficient, enabling quick training of classifiers (Hartmann, Huppertz, Schamp, & Heitmann, 2019).

Moreover, Random forest is one of the most popular bagging algorithms. Bagging offers the advantage of allowing many weak learners to combine efforts to outdo a single strong learner. It also helps in the reduction of variance, hence eliminating the overfitting of models in the procedure (Corporate Finance Institute).

The bootstrapping and feature bagging process outputs varieties of different decision trees rather than just a single tree applied to all of the data. Using this approach, the models that were trained without some features will be able to make predictions in aggregated models even with missing data. Moreover, each model trained with different subsets of data will be able to make decisions based on different structure of the underlying data/population. Hence, in aggregated model they will be able to make prediction even when the training data doesn't look exactly like what we're trying to predict. (Piccini, 2019)

Research on other applications beyond text classification suggests RF to be among the top performing methods given their versatile structures (Hartmann, Huppertz, Schamp, & Heitmann, 2019).

2.5.3.6. Bayesian Linear Regression

Regression Analysis

Regression Analysis is a statistical method for examining the relationship between two or more variables. There are many different types of Regression analysis, of which linear regression is one of the most common.

Linear Regression

Linear Regression models describe the relationship between a set of variables and a real value outcome. For example, input of the mileage, engine size, and the number of cylinders of a car can be used to predict the price of the car using a regression model.

Regression differs from classification in how its error is defined. In classification, the predicted class is not the class in which the model is making an error. In regression, for example, if the actual price of a car is $5000 and we have two models which predict the price to be $4500 and $6000, then we would prefer the former because it is less erroneous than $6,000. We need to define a loss function for the model, such as Least Squares or Absolute Value. The drawback of linear regression is that it assumes that a single straight line is appropriate as a summary of the data.

Bayesian Linear Regression

In the Bayesian world, linear regression is formulated using probability distributions rather than point estimates. The dependent variable, Y, is not estimated as a single value, but is assumed to be drawn from a probability distribution. Y is generated from a normal distribution with a mean and variance. Bayesian Linear Regression aims to find the posterior distribution for the model parameters rather than determining a single "optimal" value for the model.

In contrast to ordinary least squares (OLS), there is a posterior distribution for the model parameters that is proportional to the likelihood of the data multiplied by the prior probability of the parameters.

One of the advantages of this approach is that if we have domain knowledge (priors), or an idea about the model parameters, we can include them in our model. The major advantage of Bayesian processing is that you can incorporate the use of previous or assumed knowledge and update the current state of beliefs. You can incorporate prior information about a parameter and form a prior distribution for future analysis.

One of the shortcomings of Bayesian analysis is that it does not tell you how to select a prior. There is no single correct way to choose a prior. This approach requires skills to translate subjective prior beliefs into a mathematically formulated prior. Any misunderstanding can generate misleading results (Piccini, 2019). A non Bayesian Regression model or simply a linear regression model overfits the data, meaning the unknown part for a certain value of independent variable becomes too precise when calculated. Bayesian Linear Regression relaxes this fact, saying that there is uncertainty involved by incorporating "Predictive Distribution" (Zaman, 2016).

2.5.3.7. Logistic Regression

Logistic Regression is one of the most used Machine Learning algorithms for binary classification. It is a simple algorithm that can be used as a performance baseline, it is easy to implement and it will do well enough in many tasks. The building block concepts of Logistic Regression can also be helpful in deep learning while building neural networks. Logistic Regression is the go-to method for binary classification providing discrete binary outcome between 0 and 1. A simple example of a Logistic Regression problem would be an algorithm used for cancer detection that takes screening picture as an input and should tell if a patient has cancer (1) or not (0) (Donges, 2018), (Freedman, 2009).

Logistic Regression measures the relationship between the dependent variable, which is the label we want to predict) and the more independent variables (the features), by using underlying logistic function to estimate probabilities.

The logistic Sigmoid-Function used in LR, is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits using a threshold classifier.

To maximize the likelihood that a random data point gets classified correctly, Maximum Likelihood Estimation approach is used to estimate parameters in statistical models. An optimization algorithm, such as Newton's Method or Gradient Descent can be used to find maximum (or minimum) of many different functions, including the likelihood function (Donges, 2018). Multinomial logistic regression, also known as multi-class LR, maximum entropy (MaxEnt) classifier, and conditional maximum entropy model, is a classification method that generalizes logistic regression to multiple-class problems. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function. (Wang_2017)

Logistic Regression is a widely used technique because it is very efficient, does not require too many computational resources, it's highly interpretable, it doesn't require input features to be scaled, it doesn't require any tuning, it's easy to regularize, and it outputs well-calibrated predicted probabilities.

Like linear regression, logistic regression does work better when attributes that are unrelated to the output variable are removed, as well as attributes that are very similar (correlated) to each other. Therefore Feature Engineering plays an important role in regards to the performance of Logistic and also Linear Regression. Another advantage of Logistic Regression is that it is incredibly easy and quick to implement and very efficient to train. It can be possible to start with a Logistic Regression model as a benchmark and try using more complex algorithms from there on.

Logistic Regression is appropriate to be used when data can be linearly separable. A disadvantage of it is that we can’t solve non-linear problems with logistic regression since its decision surface is linear and a line that separates these 2 classes can’t be drawn without a huge error. In those situations it would be a much better choice to use a simple decision tree.

Logistic Regression is also not one of the most powerful algorithms and can be easily outperformed by more complex ones. Another disadvantage is its high reliance on a proper presentation of your data. This means that logistic regression is not a useful tool unless you have already identified all the important independent variables. Since its outcome is discrete, Logistic Regression can only predict a categorical outcome. It is also an Algorithm that is known for its vulnerability to overfitting (Donges, 2018)..

Logistic VS. Linear Regression

Logistic regression gives a discrete outcome, while the outcome of linear regression is continuous. A good example of a continuous outcome would be a model that predicts the value of a house, calculating a different value based on parameters like its size or location. Logistic regression, on the other hand, can specify a discrete outcome, whether you have one thing (you have cancer) or another (you have no cancer) (Donges, 2018).

2.5.3.8. Bagging (Bootstrap Aggregation)

Bagging and boosting are the two main methods of ensemble machine learning.

Ensemble Learning

An ensemble method is a participant of a bigger group of multi-classifiers and it is a machine learning platform that helps multiple models in training through the use of the same learning algorithm.

Multi-classifiers are a group of multiple learners, running into thousands, with a common goal that can fuse and solve a common problem. Another category of multi-classifiers is hybrid methods. The hybrid methods use a set of learners, but unlike the multi-classifiers, they can use distinct learning methods.

Learnin faces many challenges. Some of them are the errors occurred mainly due to bias, noise, and variance. The accuracy and stability of machine learning are guaranteed by ensemble methods such as bagging and boosting. Multiple classifiers combinations reduce variance, especially where classifiers are unstable, and they are important in presenting more reliable results than a single classifier.

The application of either bagging or boosting requires the selection of a base learner algorithm first. For example, if one chooses a classification tree, then boosting and bagging would be a pool of trees with a size equal to the user's preference (Corporate Finance Institute).

Bagging

Bagging (Bootstrap Aggregation) is an ensemble machine learning method which takes several weak models which specialize in distinct sections of the feature space, aggregating the predictions coming from every model to reach the utmost purpose. The main purpose is to select the best prediction used in regression and statistical classification when we want to reduce the variance or overfitting of a decision tree. Bagging comprises of the following steps.

Bootstrap Sampling

Several subsets of data can be obtained from the training data chosen randomly with replacement. This collection of data will be used to train decision trees. Bagging will construct n decision trees using bootstrap sampling of the training data. As a result, we will have an ensemble of different models at the end.

Aggregation

The outputs from all the separate models are aggregated into a single prediction as part of the final model. In terms of regression, the output is simply the average of predicted outcome values. In terms of classification, the category with the highest frequency output is chosen.

The bagging technique is useful for both regression and statistical classification. Bagging is used with decision trees, where it significantly raises the stability of models in the reduction of variance and improving accuracy, which eliminates the challenge of overfitting. Unlike boosting, bagging involves the training a bunch of individual models in a parallel way. The advantage of using Bootstrap aggregation is that it allows the variance of the model to be reduced by averaging multiple estimates that are measured from random samples of a population data. (Piccini, 2019) One disadvantage of bagging is that it introduces a loss of interpretability of a model. The resultant model can experience lots of bias when the proper procedure is ignored. Despite bagging being highly accurate, it can be computationally expensive and this may discourage its use in certain instances (Corporate Finance Institute).

2.5.3.9. AdaBoost

AdaBoost is an iterative ensemble method. It builds a strong classifier by combining multiple weak performing classifiers. The final classifier is the weighted combination of several weak classifiers. It fits a sequence of weak learners on different weighted training data. If prediction is incorrect using the first learner, then it gives higher weight to observation which have been predicted incorrectly. Being an iterative process, it continues to add learner(s) until a limit is reached in the number of models or accuracy. (Piccini, 2019).

Initially, AdaBoost selects a training subset randomly and gives equal weight to each observation. If prediction is incorrect using the first learner then it gives higher weight to observation which have been predicted incorrectly. The model is iteratively training by selecting the training set based on the accurate prediction of the last training. Being an iterative process, the model continues to add multiple learners until a limit is reached in the number of models or accuracy. (Piccini, 2019)

It is possible to use any base classifier with AdaBoost. This algorithm is not prone to overfitting and it is easy to implement.

One of the downsides of AdaBoost is that it is highly affected by outliers because it tries to fit each point perfectly. It is computationally slower as compared to other boost methods, such as XGBoost, and it can be used both for classification and regression problem. (Piccini, 2019)

2.5.3.10. Alternating decision tree

An alternating decision tree (ADTree) is a ML classifier which generalizes decision trees and has connections to boosting.

Differently from binary classification trees such as CART (Classification and regression tree) or C4.5 in which an instance follows only one path through the tree, an ADTree consists of an alternation of decision nodes, which specify a predicate condition, and prediction nodes, which contain a single number. An instance is classified by an ADTree by following all paths for which all decision nodes are true, and summing any prediction nodes that are traversed.

Original boosting algorithms typically used either decision stumps (trees with one level of decision nodes) or decision trees as weak hypotheses. Individual decision stumps are weighted according to their boosting iterations, which is used in voting process for the final classification.

Boosting a simple learner results in an unstructured set of hypotheses, making it difficult to infer correlations between attributes. Alternating decision trees introduce structure to the set of hypotheses by requiring that they build off a hypothesis that was produced in an earlier iteration. The resulting set of hypotheses can be visualized in a tree based on the relationship between a hypothesis and its "parent."

Another important feature of boosted algorithms is that the data is given a different distribution at each iteration. Instances that are misclassified are given a larger weight while accurately classified instances are given reduced weight.

ADTrees are typically as robust as boosted decision trees and boosted decision stumps. ADTree is preferred for its visualization and for requiring less iterations, while having similar test errors to other trees. On the other hand sometimes it can be prone to overfitting, thus memorizing instead of learning the patterns (Freund & Mason, ICML'99).

2.6. Aim and Novelty of the Study

The goal of this study is to propose a new method for feature selection based on Machine Learning algorithms and Data Science methodologies, in order to improve the overall accuracy of the classification prediction of subjective texts used in different domains as part of this research.

The importance of hybrid methodology related to different usages of Machine Learning algorithms in combination with plain ML methods has currently become very essential. The importance of data does not lie only on their own, but with the help of data science methodologies, computer processed data can become a much important tool in data analyst or scientist hands as a result of his/her high intuitive capabilities. So, the original data can be “served” again in a more refined way, as a “sauce” on top of the “meal” composed of all the data, with an improved “taste” and much more impact on the final result. Data science methodologies serve for this.

One of the alternative methods is the introduction of meta features. Meta features selection methodologies combined with ensemble learning has been righteously emphasized lately, because they give a better insight and impressively improve the results. In many studies different meta features are combined with textual features in a hybrid form to get fuller results. Besides the tendency to meta features usage, additive meta features have been used rather than the ones derived from the own data.

In this study the new method of refining the original feature set after training it first with a well known ML decision tree algorithm, such as J48, is proposed. As a result of this process some new feature sets are built derived as inner nodes of the resultant J48 tree.

A sample branch of the visualization of J48 tree can be shown in Figure 7.

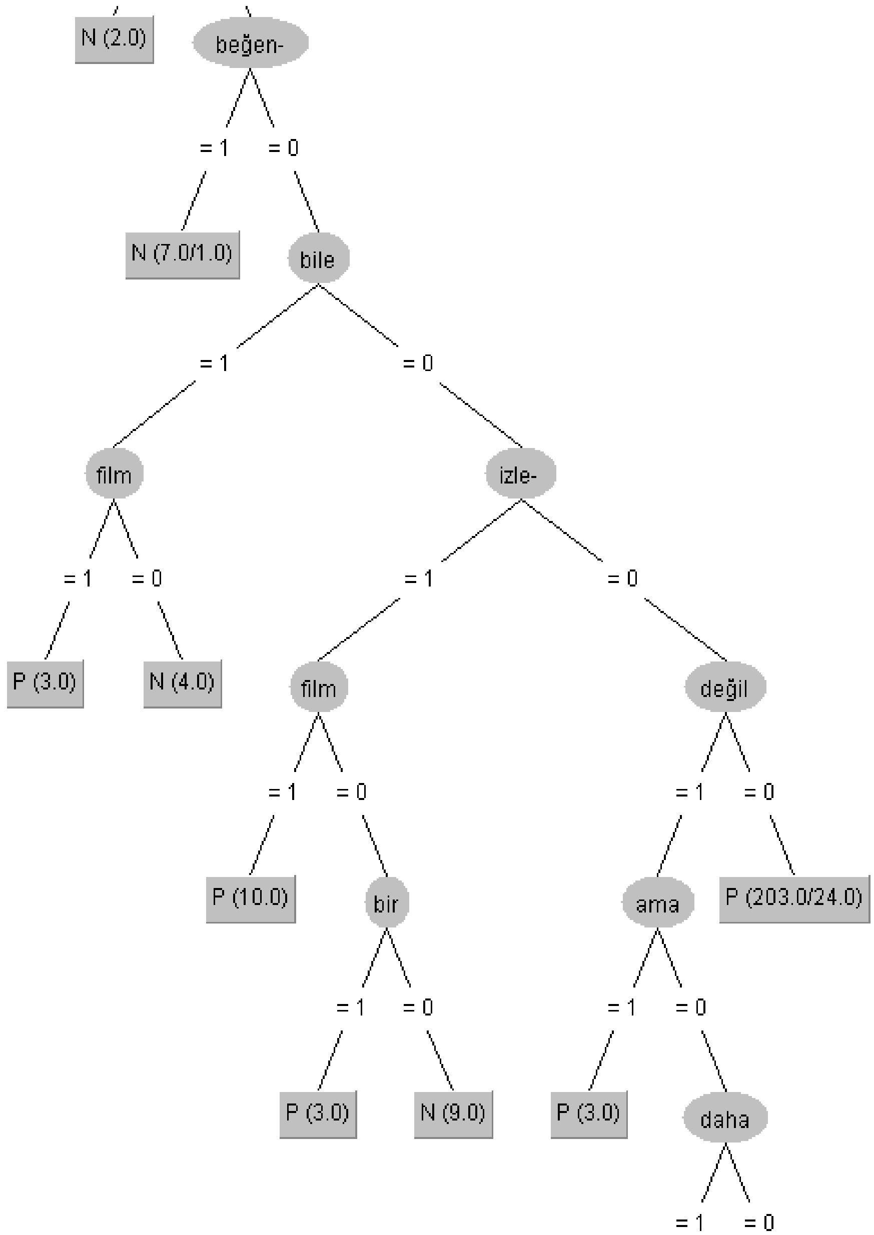


Figure 7 A branch of the J48 tree of unigrams binary vector method for movie reviews

Other sample branches of the visualization of J48 tree are shown in Figure 31, Figure 32 and Figure 33 of Appendix B.

In this visualization the features **beğen-** (*not like*), **bile** (*even)*, **film** (*movie*), **izle-** (*not watch*), **film** (*movie*), **değil** (*not*), **bir** (*a(n)*), **ama** (*but*) are nodes of the J48 tree.

The whole tree is considered, with all the nodes counted in the appearance order, and they are given values according to their level in descending order, all the apparances being counted. An example of the nodes and their respective values for the customer product reviews is shown in Table 19. In this tree, **gönder** *(send)* is the root, thus having the highest frequency, while **ben** *(I)* is the node of the lowest level of the tree, thus having a frequency of one, then come the nodes of one level higher, and so on. The reason why **firma** *(company)* has the highest frequency at all is because it appears twice.

Table 19: The binary J48 tree nodes and their respective frequencies in the customer product reviews

Turkish	English	Frequency
ben	I	1
bu	this	2
tl	Turkish Lira	2
sorun	problem	3
iste	request	3
kalite	quality	4
gerek	need	4
ilk	first	4
servis	service	5
şikayet	complaint	5
müşteri	customer	6
gönder	send	8
firma	company	10

CHAPTER 3

RESULTS AND DISCUSSION

3.1. Machine Learning for Classification Results

Machine learning based studies can be categorized into supervised, unsupervised and semi-supervised topics. Feature engineering and feature selection are also vital in a machine learning pipeline.

Supervised learning is one of the most used approaches in ML domain. Although supervised learning is successful in a rich set of applications, it has many challenges

Lastly, the features for train and test data prepared by the above mentioned feature methods yield .arff extension files, which are fed to Weka Classification tool. Several existing Machine Learning algorithms are used and their results are evaluated.

In this section the results will be discussed for each domain and approach used in the study.

3.1.1. Machine Learning versus Dictionary based methods: Movie review results

This part of the research studies the impact of different feature sets out of the whole text and out of some special words bearing a subjective meaning regarding positivity and negativity. Movies dataset was used for this purpose. Besides the dictionary method, also the effect of parts of speech, more specifically the imperative, necessity, optative moods of verbs or words deriving from verbs used in the text was tested. Negation was considered, and both unigrams and bigrams were used. As a result eight feature sets were prepared, according to the procedure discussed in section 2.4.2 in page 65. Then, the respective .arff files were fed to Weka tool. For the classification process, naïve and simple, but yet strong and high performance versions of Naïve Bayes algorithm were used. Multinomial NB (MNB), Binarized Multinomial NB (BMNB) were applied to the full set of words or special words. In addition to accuracy other measures were also calculated, such as Precision (P), Recall (R) and F-Measure (FM) for each class.

The results of experiments with Weka, the confusion matrixes and respective accuracies for NB binary results, Multinomial NB and for the Dictionary Method are given in Figure 24, Figure 25, and in Figure 26 of Appendix B.

NB results of experiments with Weka are given in the following tables. The classification results of P, R and FM values of the positive and negative classes for the dictionary, MBNB and MNB methods are given in Table 20, Table 21 and

Table 22.

When we look at the P, R and FM values, we encounter the accuracy rates for NB. All singular words gave more balanced P and R values. When important words are used and according to their being single or binary, the P and R values are more unevenly distributed among classes.

Table 20: P, R, FM values of positive and negative classes according to the dictionary method and features

	Dictionary method							
Measure	**Feature0**	**Feature1**	**Feature2**	**Feature3**	**Feature4**	**Feature5**	**Feature6**	**Feature7**
P_Pos	58	58	75	75	79	82	81	81
P_Neg	**100**	**100**	94	94	90	90	87	87
R_Pos	**100**	**100**	**96**	**96**	92	92	88	88
R_Neg	28	28	68	68	76	80	80	80
FM_Pos	74	74	84	84	85	**87**	85	85
FM_Neg	44	44	79	79	83	85	83	83

Table 21: P, R, FM values of positive and negative classes according to MBNB and features

	MBNB							
Measure	**Feature0**	**Feature1**	**Feature2**	**Feature3**	**Feature4**	**Feature5**	**Feature6**	**Feature7**
P_Pos	92	92	92	92	92	92	77	80
P_Neg	**96**	**96**	92	92	92	92	95	95
R_Pos	**96**	**96**	92	92	92	92	**96**	**96**
R_Neg	92	92	92	92	92	92	72	76
FM_Pos	94	94	92	92	92	92	86	87
FM_Neg	94	94	92	92	92	92	82	84

In the new method, fluctuations of P, R values among classes are high, especially when all words are used, while when important words are used, sounder results are obtained. Since the difference between scores is very close and no threshold value was determined for this difference, these results can be considered natural. In future studies, this situation can be corrected by first eliminating the common features used in positive and negative reviews with certain methods and then determine the difference between them by selecting the most appropriate level experimentally.

Table 22: P, R, FM values of positive and negative classes according to MNB and features

	MNB							
Measure	**Feature0**	**Feature1**	**Feature2**	**Feature3**	**Feature4**	**Feature5**	**Feature6**	**Feature7**
P_Pos	92	92	92	92	88	88	80	80
P_Neg	**96**	**96**	92	92	92	92	95	95
R_Pos	**96**	**96**	92	92	92	92	**96**	**96**
R_Neg	92	92	92	92	88	88	76	76
FM_Pos	94	94	92	92	90	90	87	87
FM_Neg	94	94	92	92	90	90	84	84

The accuracy rates for each method used with the movie database relative to the eight feature sets are shown in Table 23 and visualized in Figure 8.

Table 23: Methods accuracy rates by features

	Accuracies							
Method	**Feature0**	**Feature1**	**Feature2**	**Feature3**	**Feature4**	**Feature5**	**Feature6**	**Feature7**
Binary NB	**94**	**94**	92	92	92	92	84	86
Frequency NB	**94**	**94**	92	92	90	90	86	86
Weka Results	**94**	**94**	**94**	**94**	92	90	88	88
Dictionary Met	64	64	82	82	84	86	84	84

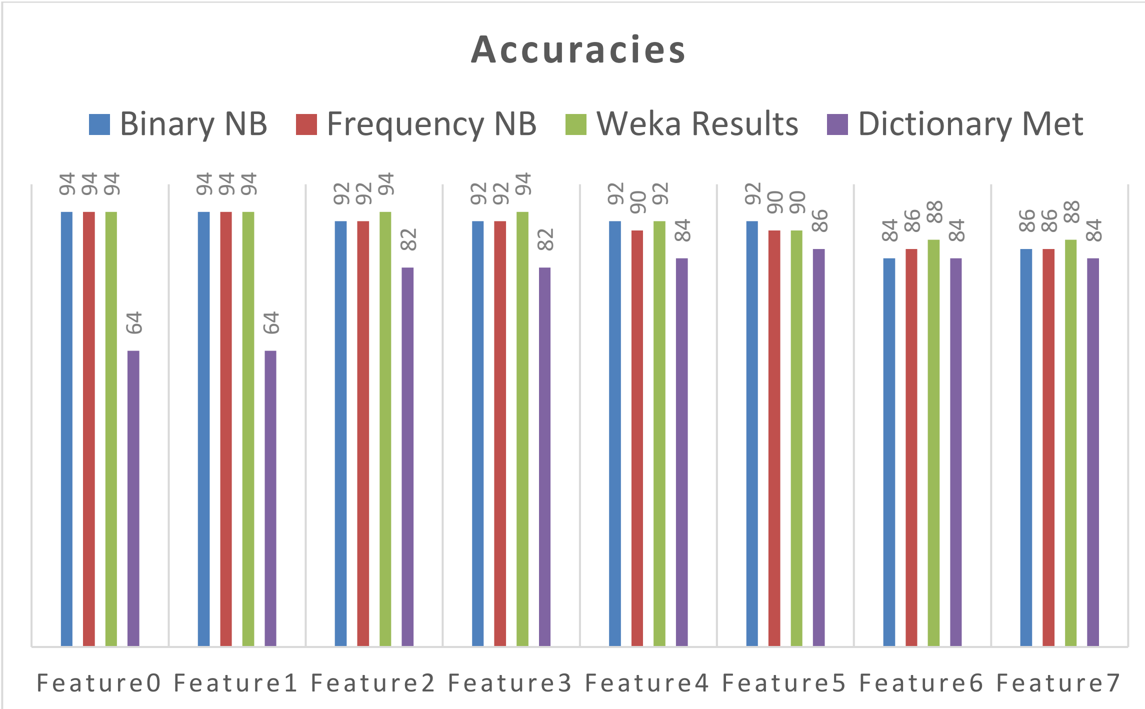


Figure 8. Accuracy rates of methods by features

According to their accuracy, NB methods give approximately the same results. The best results have been obtained for NB methods, with all single root words, and 94% accuracy was achieved. When binary words were used, the success rate decreased slightly. When only important words were used, the decrease was higher, and the accuracy rate fell to 84-86%, especially when binary words were used. The fact that single words are more successful than binary words may be attributed to reasons such as the lower number of produced features, not obtaining binary words at the training phase due to their lower probability of occurrence, and the increase in the number of produced features due to production of unnecessary features. In addition, out of the NB methods, MNB and MBNB gave approximate results. The most important factor here is that the comments in the used data are short and the frequency or presence/absence information gives approximately the same result.

The result obtained in the newly proposed method, is different compared to the ones in the full NB methods. When all of the single root words were used, the accuracy rate increased to 64% due to the many commonalities in the positive and negative comments. Nonetheless, the aim of this method is not to use all the words. Therefore, when the score was calculated by using the previously produced dictionary of important positive and negative words, as expected, accuracy rates increased from 64% to 84-86%, showing significant improvement. In addition, when looking at the binary phrases in all words, as probability of finding commonalities in the positive and negative is reduced, binary phrases gave better results than single ones in terms of success.

3.1.2. The effect of Erase Methods and Thresholds Percentages

3.1.2.1. Hospital health care review results without feature erasing

In this part of the study the impact of different feature selection methods with regard to some well known ML classifiers is been tested for different settings of the healthcare dataset. For datasets 1-4 their average is used, while for dataset 5 the actual results are displayed. Partitioning the data into four groups and then utilize their average has been done purely for performance improvement reasons, trying to keep the total size not much higher than 500 documents per dataset, since the number of features per set is quite considerable for such a size of data. The chosen ML algorithms are Naïve Bayes, Sequencial Minimal Optimization (SMO) algorithm used to train support vector machines (SVM), and J48 decision tree. The entire set of features has been used in this section of the study, so no erase method has been used, thus EN standing for Erase None is specified. The features selection methods being tested are Binary Lists (BL), Frequency Lists (FL), Binary Vector or otherwise mentioned as Binary Words (BW), and Frequency Vector, otherwise stated as Frequency Words (FW).

The average accuracy results for feature selection methods according to three classifiers is shown in Table 24 for datasets 1-4 and in Table 25 for dataset 5.

Table 24: The average accuracy percentages of datasets 1-4 from three algorithms

Methods	**Binary List**	**Frequency List**	**Binary Vector**	**Frequency Vector**
Naïve Bayes	83	90	91	89
SMO	**91**	90	**95**	**96**
J48	85	**91**	93	91

Table 25: The average accuracy percentages for Dataset 5 from three algorithms

Methods	**Binary List**	**Frequency List**	**Binary Vector**	**Frequency Vector**
Naïve Bayes	68	92	**98**	96
SMO	**86**	**93**	96	**97**
J48	73	92	92	92

The visualization of the accuracy results between differnet healthcare review datasets is shown in Figure 9.

As shown in Table 24 and Table 25 and in Figure 9 for each ML classifier used, Words or otherwise named as Vector feature selection method outperforms the others with the best accuracy for NB as 98% and with the poorest result for J48 as 92% for dataset 5, while the highest score is 95% for SMO as opposite to 91% for the average of datasets 1-4. On the other hand Word Frequencies have very high accuracies, with the peak value of 97% with SMO algorithm and dataset 5, and raning from 89%-96% for datasets 1-4 and 92%-97% for dataset 5. The Frequency List method show similar results, ranging respectively from 90% to 91% and 92% to 93% accuracy for datasets 1-4 nd 5 respectively. The poorest results are obtained from the Binary List feature selection method, ranging from 83% to 91% for datasets 1-4, whereas 68% to 86% accuracy for dataset 5.

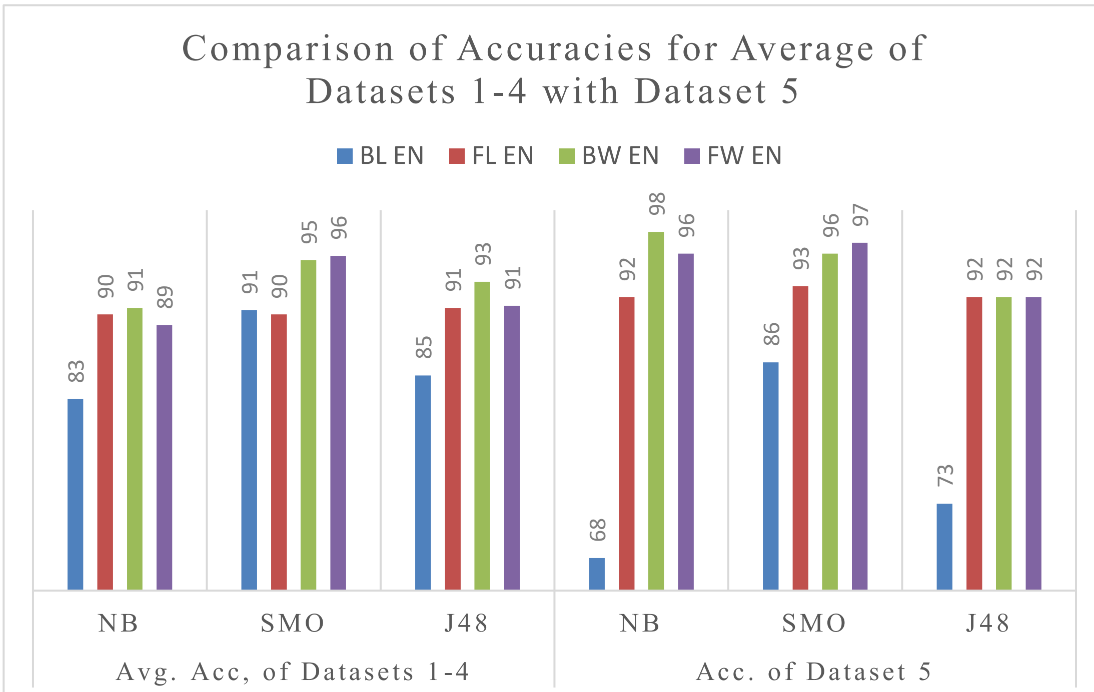


Figure 9. Hospital healthcare reviews results

According to the results, the best ones are obtained from Binary Words feature selection method. In normal circumstances, Frequency Words feature selection is expected to give better results than any other one, but because of short reviews, Word Frequencies has no priority over Words method, as it would have been expected. Moreover, the superiority of Frequency Words feature selection method over the Binary one would be expected when the common terms are trimmed out of the positive and negative lists, which is not the case in this setting.

3.1.2.2. The Movie Reviews dataset case and feature erasing

Feature sets prepared according to the methods explained in the previous section are the key attributes applied to train the machine learning algorithms in order to reckognize the characteristics of the supplied data. The following step is system testing, which provides us with the classification accuracy values, needed to evaluate the system.

Several existing Machine Learning algorithms are used and their results are evaluated. The results obtained from the following algorithms: Voted Perception (VP), Bayesian Linear Regression (BLR), and Random Forest (RF) are shown in Table 26 and Table 27 for Binary List and Frequency List feature selection methods, respectively, in combination or not with erasing according to a thresh value.

Table 26: ML Results for Binary List feature selection method

VP	BLR	RF	Erase Mode	Thresh
74	54	70	No Erase	No Thresh
75	59	**79**	Erase From Both Lists	25%
75	73	78	Erase From Both Lists	50%
74	54	70	Erase For Smallest	25%
71	60	69	Erase For Smallest	50%

Table 27: Results for Frequency List feature selection method

VP	BLR	RF	Erase Mode	Thresh
84	84	83	No Erase	No Thresh
90	89	87	Erase From Both Lists	25%
88	**90**	87	Erase From Both Lists	50%
84	84	87	Erase For Smallest	25%
84	84	85	Erase For Smallest	50%

Further, the effects of 25% and 75% threshing values and both erasing methods are measured when Voted Perception (VP), Bayesian Linear Regression (BLR), Random Forest (RF) and Logistic Regression (LOG). Here LOG function and RF tree was added respective to the prior casestudy. Their results can be seen in Table 28:

Table 28: Accuracy values (%) for all feature selection methods and their ML classifiers

VP	BLR	RF	LOG	Feature Method
90	89	87	88	Frequency List EraseBoth 25% thresh
88	**90**	87	88	Frequency List EraseBoth 50% thresh
79	80	83	89	All Words Binary
85	89	84	79	All Words Frequency
86	88	83	72	Selected Words Binary EraseBoth 25% thresh
83	**90**	87	78	Selected Words Frequency EraseBoth 50% thresh

When comparing Binary and Frequency List feature selection methods, Frequency List method clearly outdoes the Binary one to an extent 90% to 79% of respective accuracies.

It can be noted that the highest accuracies of 90% can be obtained from Frequency List feature selection method together when erasing the common terms from both positive and negative word lists within the thresh values 25% and 50% of the values, respectively for Voted Perception and Bayesian Logistic Regression ML algorithms. The same accuracy is obtained also with the Selected Words Frequency EraseBoth 50% feature selection method when classified with Bayesian Logistic Regression ML algorithms.

In general Bayesian Logistic Regression ML algorithm is very accurate, giving 89% value for Frequency List EraseBoth 25% thresh and All Words Frequency feature selection method. While Bayesian Linear Regression does not give a good result with All Words Binary feature selection method, Logistic Regression has an accuracy of 89%.

One can thus conclude that all feature selection methods have similar results with different ML algorithms. All the results are quite high for Turkish movie reviews when compared with previous studies in this field, making this one a promising study for Turkish and generally in this area.

When comparing Binary and Frequency List feature selection methods, Frequency List clearly outdoes the Binary one to an extent of 83% to 54% for the lower bound and 90% to 79% for the respective highest accuracies.

In Binary List features selection method, generally for all the classifiers the results improve when some of the features get erased. The improvement when 25% of the common features are removed only from one of the lists is lower than when both lists are compressed, to a value up to 9% for the Random Forest algorithm. Similarly, even though the highest result is achieved with the combination Random Forest, Erase from both lists with 25% thresh, the 50% thresh value, in general, achieves highest results, up to a difference of 14%. When erase methods get compared, erasing from both lists method is much more efficient (up to 13%) than erasing from only the list with lower value, while adjusting the value from the opposite list as the difference of both.

In Frequency List features selection method, we get higher accuracies always when diminishing some of the nearby frequency features from one or both lists. The improvement can be up to 13% for Bayesian Linear Regression with erasing from both lists to 50% of threshing. Again the greatest effect is shown when 50% thresh is applied opposed to 25%, and when erasing features from both lists compared to one side removal. The combinations of Voted Perception and Bayesian Linear Regression give the highest 90% accuracy with respective 25% and 50% of feature erasing from both of the lists.

To sum up, 50% thresh with Bayesian Linear Regression and 25% thresh with Voted Perception get the highest accuracies of 90% polarity classification.

3.1.2.3. The hospital dataset case with different threshings

Many studies have revealed that one of the most efficient improvement on the results of the classification process lies in the extraction or/and reduction of the features (Aliwy & Abdul Ameer, 2017). Even though the initial drive is to keep as much features as you have, the general literature review goes for getting rid of many unsignificant words for the document polarity. One way is to erase common features from both positive and negative lists when their difference falls within a specified threshold, as **Erase From Both (EB)** method implies, while another possibility is erasing from the list the lowest value and keeping in the opposite list the difference of the frequencies, when this difference falls within the given threshold ratio, according to **Erase Smallest (ES)** method.

The following thresholding values are taken accordingly, as 5%, 25%, 50% and 75% of the difference of frequencies of a common term appearing both in the positive and negative list of features, opposite to no threshing at all **(No Erase – NE)**.

As the erasing methods names suggest, Erase From Both List method reduces the number of features according to the features selection method, while erasing the smallest value just tunes the diference in the frequencies of appearance of a word, without really reducing the featue from the overall features set.

Subsequently, in Table 29, Table 30, Table 31 and Table 32 the results of different threshing and erasing methods are shown for both binary and frequency versions of the Lists and Vector features method, when **Naïve Bayes (NB)**, **Support Vector Machine (SVM)**, **J48**, **Voted Perceptron (VP)**, and **Bayesian Linear Regression (BLR)** ML classifiers are applied for the healthcare dataset.

The choice of classifiers has been done to include representatives of important categories, such as bayes (NB, BLR), functions (SMO, VP), and trees (J48), to see how different classifiers are affected from features selection options. The above results are visualized in the charts shown in Figure 10,

Figure ***11***, Figure 12 and Figure 13, as following:

Generally all the results are improved by feature removal, or feature frequency fine tuning between the positive and negative terms list.

The least increase in accuracy due to features reduction happens in Binary Vector, and next in the Frequency Vector method.

Table 29: ML results for different threshings for Binary List method

NB	**SVM**	**J48**	**VP**	**BLR**	**Erase Method**	**Erase Percent**
68	86	73	89	76	No Erase	-
90	89	88	90	79	Erase Both	5%
86	89	82	90	83	Erase Both	25%
84	89	88	90	84	Erase Both	50%
84	89	84	89	90	Erase Both	75%
88	88	82	89	77	Erase Smallest	5%
88	89	77	87	83	Erase Smallest	25%
91	91	85	90	88	Erase Smallest	50%
93	**93**	**89**	**92**	**92**	Erase Smallest	75%

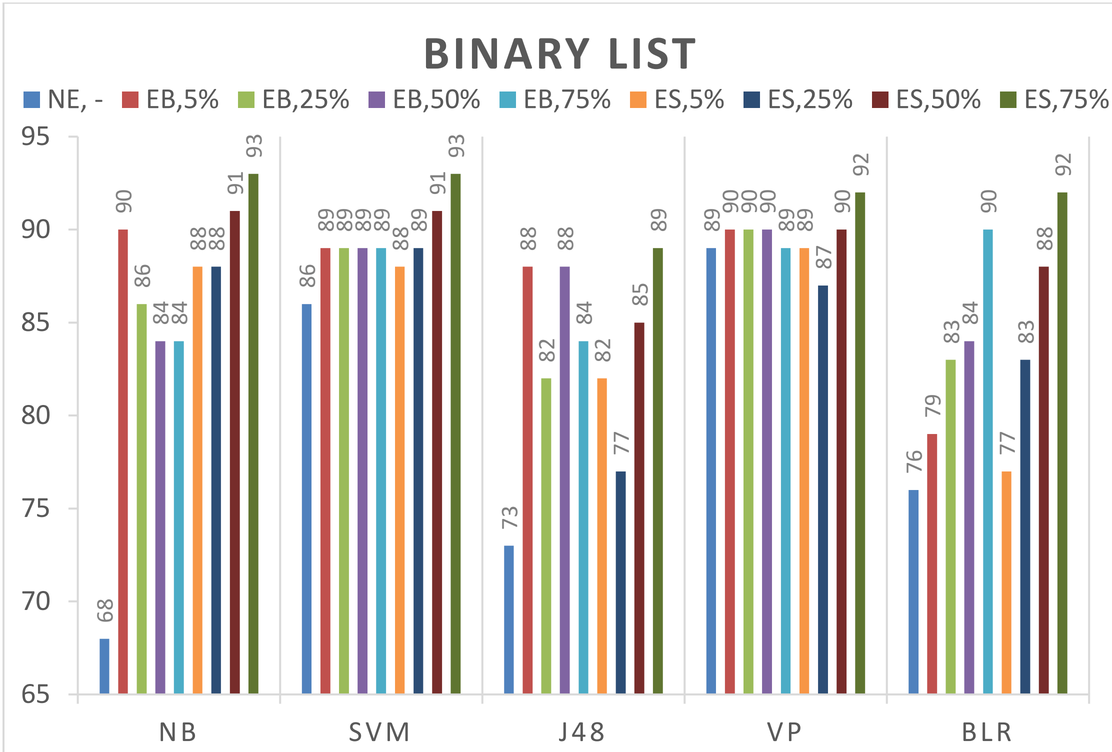


Figure 10. The effect of different threshing percentages according to ML algorithms with the Binary List feature method

The features reduction or frequency adjustion as a result of common terms in both positive and negative lists of words improves visibly the accuracy for all the classifiers with the Binary List Method, with Erasing From Smallest List and adjusting the frequency value for the other list when common values fall within 75% threshing, with NB and SMO achieving the highest score of 93% of accuracy for these threshing values. Generally erasing from one list has higher impact on the results, with higher threshing giving best results, but it is interesting that even a small threshing such as 5% with erasing from both lists improves the result drastically with 22% for NB and 11% for J48. VP and SVM profit the least, while BLR profits the most from features reduction and frequency adjusting according to different threshing values. Frequency Lists method profits more from feature reducing, with the highest values for VP, and for BLR when 75% threshing is applied achieving the highest score of 96% in this case.

Table 30: ML results for different threshings for Frequency List method

NB	SVM	J48	VP	BLR	Erase Method	Erase Percent
92	93	92	90	89	No Erase	-
92	93	92	90	89	Erase Both	5%
91	94	94	88	90	Erase Both	25%
90	94	93	89	91	Erase Both	50%
94	94	93	95	**96**	Erase Both	75%
92	93	92	90	89	Erase Smallest	5%
92	94	93	89	90	Erase Smallest	25%
92	93	91	89	91	Erase Smallest	50%
93	93	90	87	88	Erase Smallest	75%

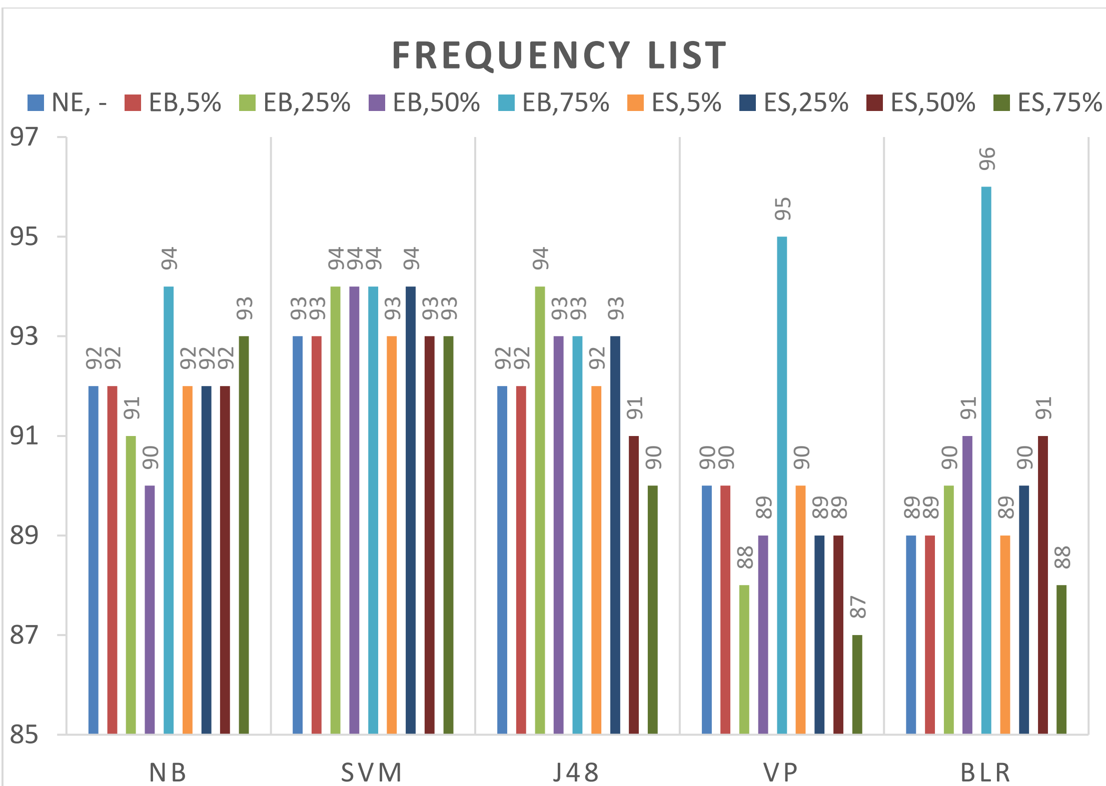


Figure 11. The effect of different threshing percentages according to ML algorithms with the Frequency List feature method

Table 31: ML results for different threshings for Binary Vector method

NB	SVM	J48	VP	BLR	Erase Method	Erase Percent
98	**95**	**94**	92	**92**	No Erase	-
97	**95**	92	82	**92**	Erase Both	5%
97	**95**	92	**94**	**92**	Erase Both	25%
97	93	92	91	89	Erase Both	50%
95	94	92	93	**92**	Erase Both	75%
98	**95**	**94**	91	**92**	Erase Smallest	5%
98	**95**	**94**	92	**92**	Erase Smallest	25%
98	**95**	**94**	93	**92**	Erase Smallest	50%
98	**95**	**94**	93	**92**	Erase Smallest	75%

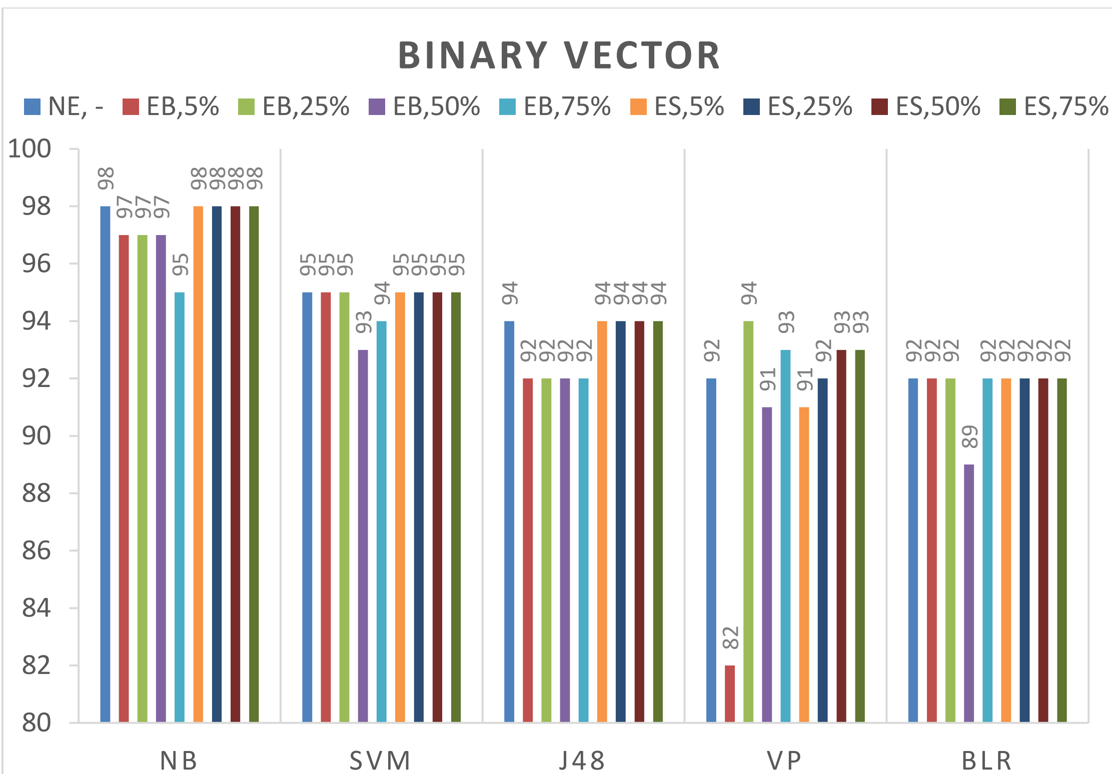


Figure 12. The effect of different threshing percentages according to ML algorithms with the Binary Vector feature method

Table 32: ML results for different threshings for Frequency Vector method

NB	SVM	J48	VP	BLR	Erase Method	Erase Percent
94	55	**92**	88	**95**	No Erase	-
93	83	91	89	94	Erase Both	5%
93	56	91	87	94	Erase Both	25%
93	57	91	90	**95**	Erase Both	50%
91	**91**	**92**	**94**	**95**	Erase Both	75%
94	55	**92**	87	**95**	Erase Smallest	5%
94	55	**92**	89	**95**	Erase Smallest	25%
94	55	**92**	91	**95**	Erase Smallest	50%
94	55	**92**	90	**95**	Erase Smallest	75%

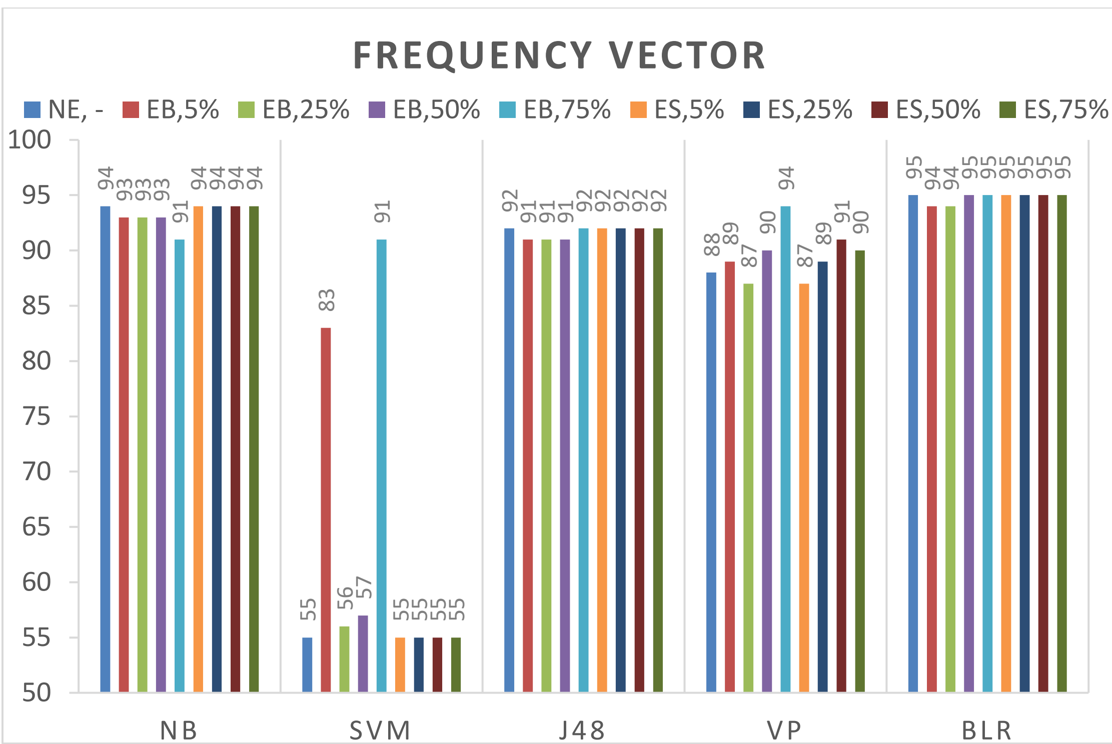


Figure 13. The effect of different threshing percentages according to ML algorithms with the Frequency Vector feature method

The Binary Vector method is not too much affected from feature erasing from one or both lists. This is also shown in the results. With the full set of features, when no erasing or just erasing from one list does not make any change, provided only the existence or absence of a feature is regarded, thus, no feature is really reduced from the feature set. The best results of 98% accuracy are obtained from NB classifier with all the features. Since the frequencies do not matter in Binary Vector method, the results of No Erase and Erase Smallest methods almost do not differ.

In the Frequency Vector method, NB, J48 and BLR classifiers give better result with full set of features, while SVM is strongly improved with 28% when 5% of the features are reduced, and with 36% increase in accuracy when submitted to 75% of featueres reduction. VP gains up to 7% from erasing some of the features. The highest score of 95% for this method is achieved with BLR full features set.

Despite the small accuracy increase, Vector methods gain much in terms of efficiency in the case of Erase From Both lists method, where the number of the features drop drastically, while the Erase Smallest method does not really affect neither the result, nor the time consumption.

In the Binary List method the most appropriate threshing ratio seems to be 75% for fine tuning the result with Erase Smallest method, even though just as little as 5% contributes to the improvement of the result. J48 classifier’s improvement look arbitrary, while the other classifiers look increasing with the features reduction. NB also gains the most from 5% features reduction in this method.

3.1.2.4. Hospital health care review results with feature erasing and TNR and FNR measures

In another setting of Dataset 5 of the hospital dataset, not only results of accuracy, but also of the TNR and FNR measues, for no erasing, erasing from one list the smallest value while adjusting the frequency in the opposite list with their difference, or erasing from both lists same values when their frequencies fall into the interval of 75% threshing.

Together with the accuracies, the True Positive (TPR) and False Positive Rates (FPR) in percentage for the negative set of data were evaluated. This was done to identify the Feature method – ML algorithm pair with the highest precision in not classifying negative data wrongly because health issues are so delicate that nobody would like to risk not to identify problems already happened to other reviewers, so that they can possibly avoid them.

While TRP is aimed to have the highest values as close as possible to 100, the opposite is true for FPR, ideally close enough to 0. The test results of the accuracy, TNR, and FNR of texts by using each of the methods when applied to the ML techniques, are given in Table 33 and Figure 14 and Figure 15 show the results of the study.

As seen from the numerical and visual results, only Binary Lists (Erase None) does not profit from equilibrating the datasets according to the text length rather than only dataset size, for more accurate normalization, while all the other methods get obviously higher results.

Table 33: ML results (Accuracy, TNR and FNR) for hospital care reviews dataset for different Feature selection and erase methods

Method	**Erase (%)**	**Accuracy**			**TNR**			**FNR**		
		NB	**SVM**	**J48**	**NB**	**SVM**	**J48**	**NB**	**SVM**	**J48**
Binary Lists (BL)	None	68	86	73	55	93	**99**	19	22	52
	Both 75	84	89	84	86	86	78	19	7	11
	Smallest 75	93	93	89	91	91	85	6	**5**	6
		NB	**SVM**	**J48**	**NB**	**SVM**	**J48**	**NB**	**SVM**	**J48**
Freq Lists (FL)	None	92	93	92	93	92	91	10	6	6
	Both 75	94	94	93	**98**	**98**	90	9	9	**4**
	Smallest 75	93	93	90	91	92	87	6	6	7
		NB	**SVM**	**J48**	**NB**	**SVM**	**J48**	**NB**	**SVM**	**J48**
Binary Words (BW)	None	**98**	96	92	**98**	94	92	**2**	**2**	8
	Both 75	95	94	92	96	94	92	5	6	8
	Smallest 75	**98**	95	94	**98**	96	95	**2**	5	7
		NB	**SVM**	**J48**	**NB**	**SVM**	**J48**	**NB**	**SVM**	**J48**
Freq Words (FW)	None	96	**97**	92	**98**	**97**	94	6	**3**	10
	Both 75	91	91	92	89	90	93	7	8	10
	Smallest 75	94	55	92	96	87	96	9	77	12

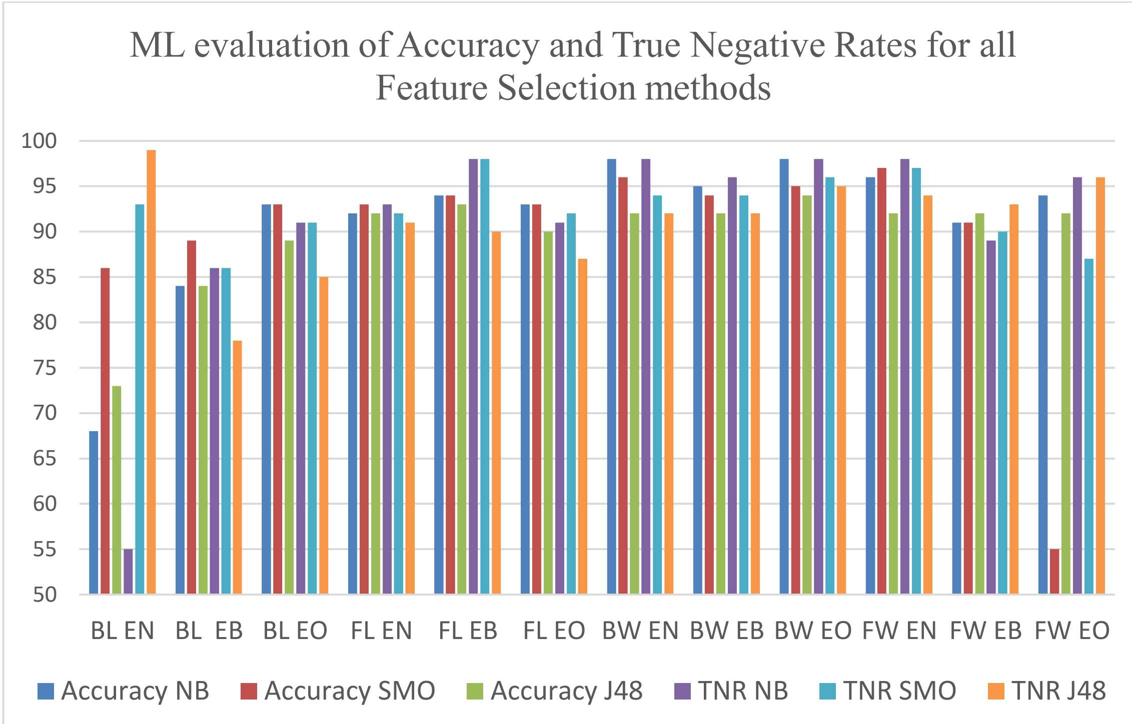


Figure 14. ML evaluation of Accuracy and True Negative Rates for all Feature methods

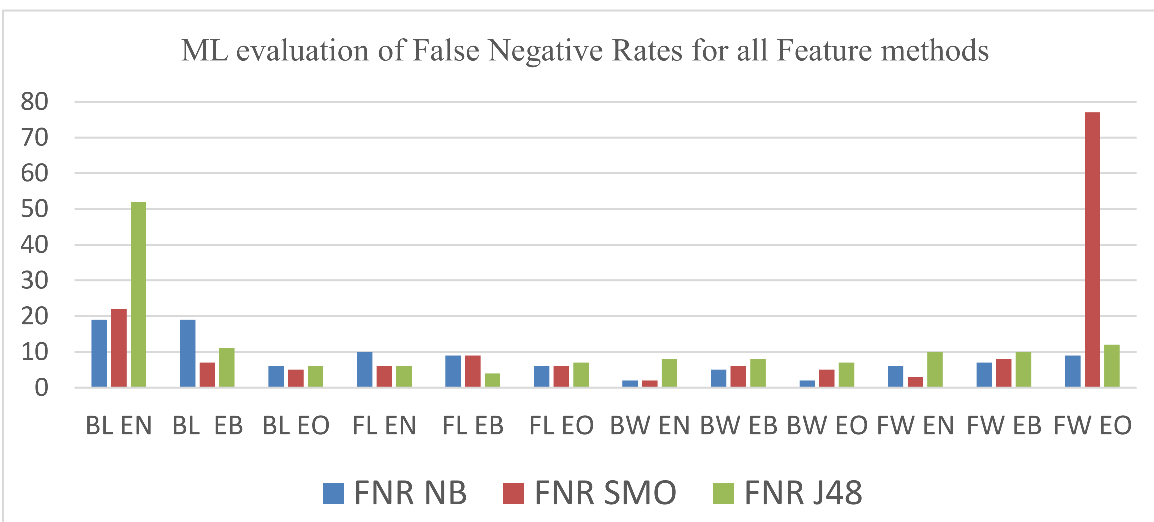


Figure 15. ML evaluation of False Negative Rates for all Feature methods

In Dataset 5, for each ML classifier used, Binary Lists has the poorest performance, both accuracy-wise (Acc. of NB of 68%), and TNR-FNR-wise (TNR of NB 55%, FNR of J48 52%). These results are worse when no erasing is done, while erasing from one list is superior to erasing from both lists, especially for the FNR value (getting up to 5-6% for all ML algorithms). When erasing is performed, more relevant information is processed, thus especially when one entry having the largest value remains in the list, the information provided is more significant, since List methods process linear data. This is especially visible in the FNR result of J48.

Frequency List methods have better outcomes compared to Binary List methods, understandably emphasizing the importance of using frequency information in linear computational processing. The highest accuracy of 94% is reached for this method when both similar values are erased from both positive and negative lists of words at a threshold of 75%, for NB and SMO. These results are supported by high values of 98% of TNR, too.

The best prediction is achieved with Binary Words method and Naïve Bayes (98% accuracy, 98% TNR and only 2% FNR), when all features are used (this is the case for no erasing or erasing from one list, being binary the information is not different). The next good prediction value is for SMO with no erasing (96% accuracy, 98% TNR, and 2% FNR). SMO, on the other hand, gets the highest results for Frequency Words method, with 97% accuracy, 97% TNR, and 3% FNR. The same ML algorithm, SMO, surprisingly gets a very poor result for Frequency Words method with erasing from one list with the threshing of 75%, with 55% accuracy, 87% PNR and a record of 77% of FNR.

According to the results, the best ones are obtained from Binary Words feature selection method, combined with Naïve Bayes feature selection method. In normal circumstances, Word Frequency feature selection is expected to give better results than any other one, but because of short reviews, Word Frequencies have no clear superiority.

3.1.3. Medical Tests Recommendation results

In this part of the study dedicated the medical test recommendation according to patients' anamnesis data, Random Forest, Bagging, Alternating Decision Tree and Adaptive Boosting (AdaBoostM1) ML classifiers are used. Random Forest is for constructing a forest of random trees (Breiman, 2001). Bagging is a class for bagging a classifier to reduce variance, and it can do classification and regression depending on the base learner (Breiman, 1996). Alternating Decision Tree is a class for generating a decision tree (Freund & Mason, 1999). Adaptive Boost is a boosting technique, which is a machine learning *meta-algorithm* that aims to *iteratively* build an *ensemble* of *weak learners*, in an attempt to generate a strong overall model. Detailed information about different classifiers and their characteristics is provided in section 2.5.3 above.

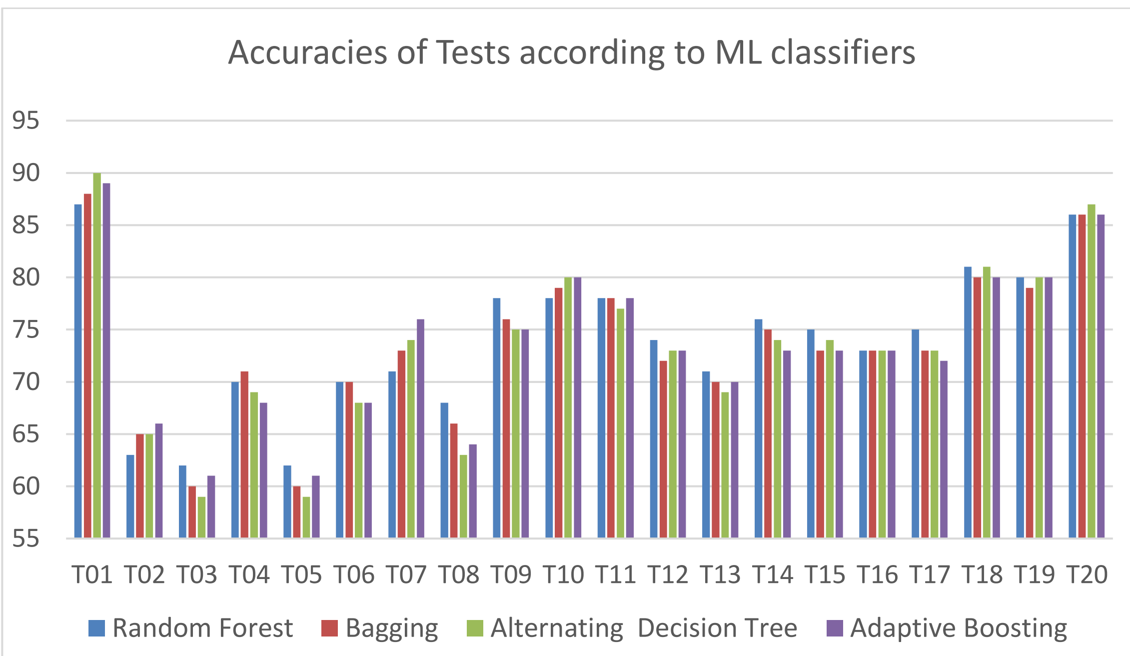


Figure 16. Medical tests accuracies results

Table 34: The average accuracy of the system as percentages for each classifier and each medical test.

Test Nr	**Random Forest**	**Bagging**	**Alternating Decision Tree**	**Adaptive Boosting**
T01	87	88	**90**	89
T02	63	65	65	**66**
T03	**62**	60	59	61
T04	70	**71**	69	68
T05	**62**	60	59	61
T06	**70**	**70**	68	68
T07	71	73	74	**76**
T08	**68**	66	63	64
T09	**78**	76	75	75
T10	78	79	**80**	**80**
T11	**78**	**78**	77	**78**
T12	**74**	72	73	73
T13	**71**	70	69	70
T14	**76**	75	74	73
T15	**75**	73	74	73
T16	**73**	**73**	**73**	**73**
T17	**75**	73	73	72
T18	**81**	80	**81**	80
T19	**80**	79	**80**	**80**
T20	86	86	**87**	86

The results depicted that some tests are easier to be detected with a high confidence from the symptoms, like T01 and T20, however, some medical tests can only be decided by careful examination of the expert like T03 and T05. The type of ML classifier used has only a slight difference (in the range of 1-4 percent) on the detection of the tests. Therefore, using any of these ML classifiers can be helpful for final decision.

3.1.4. Product review results with the novel Nodes Methods

The results of accuracies obtained from the unigram features of Words Vector, Nodes Vector and Nodes Hybrid Vector binary and frequency methods for TF (Term Frequency), TF-IDF (Term Frequency – Inverse Document Frequency) and WLLR (Weighted Log Likelihood Ratio) models for Product Review Data are shown respectively in Table 35, Table 36 and Table 37.

Table 35: Accuracies for TF model

NB	**SMO**	**J48**	**VP**	**BLR**	**Method**
96	98	93	75	95	Words Binary Vector
96	98	95	86	95	Words Frequencies Vector
94	95	95	92	84	Nodes Binary Vector
92	93	95	93	93	Nodes Frequencies Vector
95	98	95	93	**99**	Hybrid Nodes and Words

Table 36: Accuracies for TF-IDF model

NB	**SMO**	**J48**	**VP**	**BLR**	**Method**
75	23	75	2	25	Words Frequencies Vector
85	85	95	93	90	Nodes Frequencies Vector
93	90	95	**97**	94	Hybrid Nodes and Words

Table 37: Accuracies for WLLR model

NB	SMO	J48	VP	BLR	Method
97	98	95	75	93	Words Binary Vector
93	98	94	97	97	Words Frequencies Vector
93	95	95	92	84	Nodes Binary Vector
91	92	94	93	93	Nodes Frequencies Vector
99	98	95	**99**	**99**	Hybrid Nodes and Words

The results of the Binary and Frequency Nodes Vector, and especially those for Hybrid Nodes and Words Vector methods are striking. For the first two, attaining close to much more time and effort requiring methods results with such a reduced, yet too powerful features set of nodes of the J48 decision tree. For the final classification process a wide range of both simple and complex classifiers were used, in order to have a wide impact of the results. While NB, SMO and J48 give best results with the binary version, more complicated and strong algorithms like VP and BLR profit from the frequency values calculated reversely out of the level and depth of the nodes of the decision tree. This is an evidence that, provided the features are determinant and meaningful, their frequencies is a reliable information.

The best results are acquired from the Hybrid Nodes Frequency information and all the words features binary information. Here the results of document term frequency, term frequency – inverse document frequency and for the weighted logarithmic likelihood ratio. More sophisticated models such as WLLR and TF-IDF outdo the results compared to TF model, since they stress out the frequencies of terms regarding their overall appearance in other documents, emphasizing the most determinant ones. BLR algorithm achieved the highest score of 99% accuracy, hence it is a good candidate as a classifier.

The fact that this method outperforms the other methods in all the cases, achieving a higher result for different classifiers is not a sporadic event, instead it is a motivating sign of development in the sentiment analysis and in ML area.

In this study meta-level feature selection was performed. This is a novelty in SA. Although the concept of meta-level SA approaches have been described in literature, to the extent of our knowledge, none of them deals directly with words as features, but rather with auxiliary information, such as numbers of documents used in a study, number of aspects analyzed, etc. Other studies use negativity suffices, other parts of speech features as meta information, separated from the semantics of the features itself. Our approach targets a completely new features set out of all the original set of features by using the nodes and leafs of the J48 tree formed by Weka tool output with different options as parameters. Meta-level features preparation of All Words Binary with J48 Nodes Frequency Vector provides a powerful features set. The results are very promising.

Methods using only nodes information for unigrams are almost as good as the ones obtained from all the words information used, being fairly more efficient.

The new hybrid method using binary words and frequency nodes information for unigrams shows higher performance, while for bigrams this method shows similar results. Given that the bigram features set are very large and sparse at the same time, significant bigrams should be first selected, contributing indirectly also to negation, asciification and uncommon misspelling and mistyping problem resolving, then hybrid method should be tested. The two results could be used in combination.

The results involving the meta features with J48 tree nodes are very promising and just a good start point to be elaborated deeply in future studies from our part or in general.

CHAPTER 4

CONCLUSION

With the advancement of technology and the increase in data through which people express their emotions and opinions in electronic environments, sentiment analysis and opinion mining have become an important area. The exponential increase in the studies in this field in recent years and the allocation of astronomical budgets to studies and applications in this field make it even more attractive. Using today's facilities, it is possible to develop many applications from which people can benefit a great deal.

Even though being an uprising research area, much has to be done yet. Besides traditional careful information retrieval, data preprocessing and ML classifications, preparing strong and significant features sets remains crucial.

While in other countries there is a lot of work in this area, studies in Turkish have just started and are still new. Sentiment Analysis is a challenging task, especially for morphologically rich languages, like Turkish.

Products customer reviews, movie reviews, hospital healthcare reviews and anamneses data are used to be able to recommend medical tests or predict the comments polarity as positive or negative.

Although we applied NB and simple techniques with film reviews, we obtained very good results with these simple and high performance techniques. In addition, although the manual selection of terms in the dictionary extraction method, which we call the new method, is more difficult, it didn't yield more successful results compared to traditional machine learning methods. However, dictionaries extracted from a large number of reviews will give better results.

An advantage of the lexicon-based approach as compared to more generally used machine learning is that in the former, training set need not be labeled previous to the classification. They work according to text grammar analysis principles, while the elder fits the algorithms to the training set characteristic patterns. It is interesting that besides being inferior to machine learning methods in specific domains, lexicon-based methods can be quite better for wider domain sets. Once again it is implied that, although showing a weaker performance in data classification within one domain due to training dataset pattern overfitting, lexicon-based methods are more robust and show better results in cross-domain text classification process.

In order to obtain better results from the dictionary method, important words can be selected more carefully, their numbers can be increased and in particular, binary terms can be selected more meticulously. The important words coming from new reviews can also be added to the dictionary. Similarly, although the method containing modal terms is an original invention, the expected high success was not achieved due to the low number of modal terms mentioned in the reviews; however, the success rate can be increased by selecting features from only small word clusters which contain specific terms instead of using all terms in such methods.

Some results from the movie review dataset imply that simple and high performance ML techniques, such as Naïve Bayes give very good results. According to their accuracy, NB methods give approximately the same results. The best results have been obtained for NB methods, with all single root words and 94% accuracy was achieved. When binary words were used, the success rate decreased slightly. When only important words were used, the decrease was higher, and the accuracy rate fell to 84-86%, especially when binary words were used. The fact that single words are more successful than binary words may be attributed to reasons such as the lower number of produced features, not obtaining binary words at the training phase due to their lower probability of occurrence, and the increase in the number of produced features due to production of unnecessary features. In addition, out of the NB methods, Multinomial NB and Multinomial Binarized NB gave approximate results. The most important factor here is that the comments in the used data are short and the frequency or presence/absence information gives approximately the same result.

When all of the single words were used, the accuracy rate was low due to the many commonalities in the positive and negative comments. Nonetheless, the aim of this method is not to use all the words. Therefore, when the score was calculated by using the previously produced dictionary of important positive and negative words, as expected, accuracy rates increased from 64% to 84-86%, showing significant improvement. In addition, when looking at the binary phrases in all words, as probability of finding commonalities in the positive and negative is reduced, binary phrases gave better results than single ones in terms of success.

When we look at the P, R and FM values for NB, all singular words gave more balanced P and R values. When important words are used and according to their being single or binary, the P and R values are more unevenly distributed among classes. Fluctuations of P, R values among classes are high, especially when all words are used, while when important words are used, sounder results are obtained. Since the difference between scores is very close and no threshold value was determined for this difference, these results are considered natural. This situation can be corrected by first eliminating the common features used in positive and negative reviews with certain methods and then determine the difference between them by selecting the most appropriate level experimentally.

To sum up, a sentiment analysis study was performed using different domains reviews from social media or health domain by using the polarity information of the words appearing in each of the positive or negative comment group. This study evaluates as features, together with their frequency of usage or not. Entire or reduced sets of features after common terms elimination from one or both lists are prepared. The unique roots of the words for each class information and their respective frequencies serve to train a system with machine learning known methods to further forecast the polarity of the test set of data. The highest precision is achievable when a set of meaningful features with high positive or negative significance and respective frequencies is obtained as a result of clearing the similar frequency terms falling inside a determined thresh value from both positive and negative lists.

Similarly, although the method containing modal terms is an original invention, the expected high success was not trivial due to the low number of modal terms mentioned in the reviews; however, the success rate can be increased by selecting features from only small word clusters which contain specific terms instead of using all terms in such methods. When the dictionary extraction method is used together with modal terms and especially with machine learning methods, the hybrid method will yield better results.

Rather than minimum values removal (dealing also with uncommon misspelling and mistyping problems), percentages modification or removal, ML supported correlation features selection attribute selection methods are more useful.

Together with Term Frequency model, Term Frequency – Inversed Document Frequency and Weighted Log Likelihood Ratio are beneficial in obtaining better features sets, not very much negatively affected from coefficients used during data normalization and balancing.

Semantic based features are very interesting (like negativity implying words, meaning strengthening or diminishing words effect, necessity or imperative words, etc., but they are to be used only in combination with other main methods, to improve the whole result.

Bigrams are very powerful, but powerful features sets should for sure be carefully selected because of very sparse features (which can also induce contribution to negation problem resolving.

Meta-level features preparation of All Binary Words Vector with J48 (Quinlan, 1993) Nodes Frequency Vector provides a powerful features set.

Smart threshing contribute in higher performance without significant information loss. It is important because it improved the overall performance of the classification process without effecting the results. This method was used prior to Hybrid Nodes features selection method.

The results of this study are not restricted only to reviews and it is not domain dependant. Since a wide range of people from different backgrounds use social media, they generate important data which has high profit potential and are appropriate for many applications. This application is inexpensive in terms of cost items as well; therefore, it is widely and easily applicable in many areas. This study can be improved by changing the application area and applying further rules specific to Turkish.

In this study meta-level feature selection was performed. This is a novelty in SA. Although the concept of meta-level SA approaches have been described in literature, to the extent of our knowledge, none of them deals directly with words as features, but rather with auxiliary information, such as numbers of documents used in a study, number of aspects analyzed, etc. Other studies use negativity suffices, other parts of speech features as meta information, separated from the semantics of the features itself.

Methods using only nodes information for unigrams are almost as good as the ones obtained from all the words information used, being fairly more efficient.

The new hybrid method using binary words and frequency nodes information for unigrams shows higher performance, while for bigrams this method shows similar results. Given that the bigram features set are very large and sparse at the same time, significant bigrams should be first selected, contributing indirectly also to negation, asciification and uncommon misspelling and mistyping problem resolving, then hybrid method should be tested. The two results could be used in combination.

Our approach targets a completely new features set out of all the original set of features by using the nodes and leafs of the J48 tree formed by Weka tool output with different options as parameters. Meta-level features preparation as a combination of All Words Binary Vector with J48 Nodes Frequency Vector provides a powerful features set.

The results are very promising. The highest results in this study were achieved using the Hybrid Nodes and Words Vector method, scoring 99% of accuracy for multiple settings, such as Bayesian Linear Regression (BLR) classifier in TF model, and with Naïve Bayes, Voted Perceptron, and Bayesian Linear Regression classifiers in combination with WLLR model. TF-IDF model and Voted Perceptron classifier combination scored 97% of accuracy as the best result for that model, too.

The other novel approach combines smart threshing, which contribute in higher performance without significant information loss. Rather than modification or removal of values induced from some percentages range frequency similarity, ML supported attribute Correlation Features Selection (CFS) methods are more useful for more significant feature refinement, reducing the feature sets in favor of performance and predicted accuracy. Although such methods do not contribute directly to the accuracy improvement, their value is enormous. CFS techniques were successfully used for feature selection in this study.

Besides Term Frequency model, Term Frequency – Inversed Document Frequency and Weighted Log Likelihood Ratio models used in this study are beneficial in obtaining better features sets, by emphasizing the significant class determining terms and even complementing the negative effects of the manipulations used during data normalization and balancing.

Semantic based features are very interesting (like negativity implying words, meaning strengthening or diminishing words effect, necessity or imperative words, etc., but they are to be used only in combination with other main methods, to improve the whole result.

Together with prediction accuracies, other measures are important to consider, especially in accordance with the domain. i.e. in health domain sometimes being sure about the results is more important than having many positive comments in the purpose.

One can thus conclude that all the feature selection methods proposed have high results with different ML algorithms for Turkish movie reviews when compared with previous studies in this field, making this one a promising study for Turkish and generally in this area. Forthcoming research might exploit machine learning techniques for more significant feature refinement, reducing the feature sets in favor of performance and predicted accuracy. These results are very promising, and to the extent of our knowledge, this is one of the rare studies to produce results for Turkish SA. This study is valuable since it enables the efficient and automatic classification of large amounts of reviews and comments on products, which is very precious in today's market in order to make profitable choices and even on individual site before buying a product. A few simple grammar rules, for a very complex and lesser studied language, namely Turkish are applied and very good results are obtained.

The results involving the meta features with J48 tree nodes are very promising and just a good start point to be elaborated deeply in future studies from our part or in general, which makes us proud of such a discovery. We believe that these inferences will enlighten us and other researchers of the field. The promising results of the study show that language usage can be good indicator of efficient text classification.

REFERENCES

Bibliography

Ageev, M. S., & Dobrov, B. V. (2003). Support Vector Machine Parameter Optimization for Text Categorization Problems. *ISTA*, 165-176.

Aliwy, A. H., & Abdul Ameer, E. H. (2017, No). Comparative Study of Five Text Classification Algorithms with their Improvements. *International Journal of Applied Engineering Research, 12*(14), 4309-4319. Retrieved from http://www.ripublication.com

Alpaydın, E. (2018). *Yapay Öğrenme.*

Amanet, H. (2017). *Sentiment analysis in turkish social media texts.* Karadeniz Technical University.

Amasyalı, F. (2008). Yeni Makine Öğrenmesi Metotları ve İlaç Tasarımına Uygulanması. *Doktora Tezi, Fen Bilimleri Enstitüsü*. İstanbul, Turkey: Yıldız Teknik Üniversitesi.

Amasyalı, M. F. (2012). Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması" - "A Comparison of Text Representation Methods for Turkish Text Classification. *EMO Bilimsel Dergi, 2*(4).

Auer, P., Jaksch, T., & Ortner, R. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research, 11*, 1563–1600.

Badgujar, M. G., & Sawant, K. (2016). Improved C4.5 Decision Tree Classifier Algorithms for Analysis of Data Mining Application. *International Journal, 1*(8).

Balahur, A., Mihalcea, R., & Montoyo, A. (2014). Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications. *Computer Speech Language, 28*, 1-6.

Bekir, D., & Karaoğlan, B. (2003). Stemming in Agglutinative Languages: A Probabilistic Stemmer for Turkish. *Lecture Notes in Computer Science (2869): Proceedings of Computer and Information Sciences - ISCIS 2003, 18th International Symposium, Antalya, Turkey, November 3-5*, *2869*, pp. 244-251. Antalya, Turkey. doi:10.1007/978-3-540-39737-3_31

Bollen, J., Mao, H., & Zeng, X.-J. (2010). Twitter Mood Predicts the Stock Market. *Journal of Computational Science, 2*. doi:10.1016/j.jocs.2010.12.007

Botego. (2014, 10 04). Retrieved from Sentiment analizi (Duygu analizi): http://www.sentimentanalizi.com/

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140.

Breiman, L. (2001). Random Forests. *Machine Learning, 5*(1), 5-32.

Buche, A., Chandak, D. M., & Zadgaonkar, A. (2013). Opinion Mining and Analysis: A survey. *IJNLC, 2*(3), 39-49.

Bulchand-Gidumal, J., Melián-González, S., & López-Valcárcel, B. (2013). A social media analysis of the contribution of destinations to client satisfaction with hotels. *International Journal of Hospitality Management, 35*, 44-47. doi:10.1016/j.ijhm.2013.05.003

Burnap, P., Williams, M., & Rana, O. (2013). Detecting tension in online communities with computational Twitter analysis. *Technological Forecasting and Social Change, 95*(9), 96–108.

Ceyhan, M., Orhan, Z., & Domnori, E. (2017, March). Health service quality measurement from patient reviews in Turkish by opinion mining. *IFMBE Proceedings of International Conference on Medical and Biological Engineering in Bosnia and Herzegovina. 62*, pp. 649-653. Singapore: Springer. doi:10.1007/978-981-10-4166-2

Ceyhan, M., Orhan, Z., & Domnori, E. (March, 2017). E-Medical Test Recommendation System Based on the Analysis of Patients' Symptoms and Anamnesis. *IFMBE Proceedings of International Conference on Medical and Biological Engineering in Bosnia and Herzegovina, CMBEBIH. 62*, pp. 654-659. Singapore: Springer. doi:10.1007/978-981-10-4166-2

Ceyhan, M., Orhan, Z., & Karras, D. (2018). Sentiment Polarity Classification of Turkish Product Reviews for Measuring and Summarizing User Satisfaction. In A. DL (Ed.), *Proceedings of the Workshop on Opinion Mining, Summarization and Diversification, RevOpiD '18. 1*, pp. 1-10. Baltimore, Maryland, New York, NY, USA: Association for Computing Machinery. doi:10.1145/3301020.3303752

Ceyhan, M., Orhan, Z., & Karras, D. (2020, August). An Approach for Movie Review Classification in Turkish. *European Journal of Engineering and Formal Sciences, 4*(2), 1-9. Retrieved 2020, from http://journals.euser.org/files/articles/ejef_v4_i2_20/Ceyhan.pdf

Ceyhan, M., Orhan, Z., & Karras, D. (2020). Sosyal Medyadaki Film Yorumlarının Fikir Madenliği ile Otomatik Sınıflaması. *International Journal of Social Sciences - Uluslararas{\i} Sosyal Bilimler Dergisi, 4*(20), 1-31. Retrieved 2020, from http://www.sobider.net/FileUpload/ep842424/File/film_makale_(1).pdf

Ceyhan, M., Orhan, Z., & Karras, D. (2020). Towards machine evaluation of human produced comments in Turkish. *ICMS XXIII Proceedings on 23rd International Conference on Multidisciplinary Studies: "Resilience for Survival"*, *II*, pp. 236-241. Cambridge, UK. Retrieved from http://books.euser.org/files/proceedings/icms23_proceedings_v2_ISBN_9781649991584.pdf?v=5

Choi, D., Ko, K., Kim, H., & Kim, P. (2014). Text analysis for detecting terrorism-related articles on the web. *Journal of Network and Computer Applications, 38*, 16-21. doi:10.1016/j.jnca.2013.05.007

Corporate Finance Institute. (n.d.). *Bagging (Bootstrap Aggregation) A machine learning ensemble used to improve the accuracy and stability of algorithms in regression and statistical classification*. Retrieved from Corporate Finance Institute: https://corporatefinanceinstitute.com/resources/knowledge/other/bagging-bootstrap-aggregation/

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*, 273-297. doi:http://dx.doi.org/10.1007/BF00994018

Daneghyan, R. (2019). *54 SEO Statistics That Will Impact Your Business in 2020*. Retrieved 2020, from https://www.safaridigital.com.au/blog/seo-statistics-2019

Dave, K., Lawrence, S., & Pennock, D. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of the 12th international conference on World Wide Web* (pp. 519–528). Budapest, HU: ACM.

Digital in 2020. (2020). Retrieved from Digital in 2020: https://wearesocial.com/digital-2020

Digital in 2021. (2021). Retrieved from https://wearesocial.com/digital-2021

Donges, N. (2018, 04 23). *machinelearning-blog.com Tutorials and explanations about applied Machine Learning*. Retrieved from The Logistic Regression Algorithm: https://machinelearning-blog.com/2018/04/23/logistic-regression-101/

Duan, L., Street, W., & Xu, E. (2011). Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems 5 (2)*, 169-181.

eHealth. (2020). Retrieved from World Health Organization: https://www.who.int/ehealth/en/

Eroğul, U. (2009). Sentiment analysis in Turkish. *Middle East Technical University, Ms Thesis, Computer Engineering*.

Freedman, D. A. (2009). Statistical Models: Theory and Practice. In *Statistical Models: Theory and Practice* (p. 128). Cambridge University Press.

Freeland, A. G. (2017). *SEO and Social Secrets Every Entrepreneur Must Know and Why*. Retrieved from https://www.huffpost.com/entry/6-seo-and-social-secrets-_b_5505296

Freund, Y., & Mason, L. (1999). The alternating decision tree learning algorithm., (pp. 124-133). Bled, Slovenia.

Freund, Y., & Mason, L. (1999). The Alternating Decision Tree Learning Algorithm. *In Proceedings of the Sixteenth International Conference on Machine Learning* (pp. 124–133). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. doi:10.5555/645528.657623

Freund, Y., & Schapire, R. E. (1999). Large Margin Classification Using the Perceptron Algorithm. *Machine Learning, 37*(3), 277-296.

Galathiya, A. S., Ganatra, A. P., & Bhensdadia, C. K. (2012). Classification with an improved Decision Tree Algorithm. *International Journal of Computer Applications, 46*(23), 1-6.

Galathiya, A. S., Ganatra, A. P., & Bhensdadia, C. K. (2012). Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning. *International Journal of Computer Science and Information Technologies, 3*(2), 3427-3431.

Global Health Expenditure Database (GHED). (2020, October 2). Retrieved from World Health Organization: https://apps.who.int/nha/database

Hall, M. A. (1998). Correlation-based feature selection for machine learning. *PhD thesis*. Hamilton, New Zealand: Department of Computer Science, University of Waikato.

Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the 17th International Conference on Machine Learning* (pp. 359–366). San Francisco, Calif, USA: Morgan Kaufmann.

Hankamer, J. (1986). Finite state morphology and left to right phonology. *In Proceedings of the Fifth West Coast Conference on Formal Linguistics.* Stanford, CA.

Hartmann, J., Huppertz, J., Schamp, C., & Heitmann, M. (2019). Comparing automated text classification methods. *International Journal of Research in Marketing, 36*(1), 20-38. doi:https://doi.org/10.1016/j.ijresmar.2018.09.009

Haux, R. (2006). Health information systems – past, present, future. *International Journal of Medical Informatics, 75*, 268—281.

Improving Arabic text categorization using decision trees. (2009). *First International Conference on Networked Digital Technologies* (pp. 110-115). IEEE.

Indurkhya, N., & Damerau, F. (2010). *Handbook of natural language processing* (Vol. 2). Chapman & Hall/CRC.

Internet World Stats. (2020). Retrieved from https://www.internetworldstats.com/stats.htm

Joglekar, S. (2016, 03 06). *SACHIN JOGLEKAR'S BLOG*. Retrieved from ADABOOST A (small) introduction to Boosting: https://codesachin.wordpress.com/tag/adaboost/

John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338–345). Morgan Kaufmann Publishers.

Jurafsky, D. (2018). *What is sentiment analysis*. Retrieved 2018, from https://web.stanford.edu/class/cs124/lec/sentiment.pptx

Kamran, M., & Javed, A. (2015). A Survey of Recommender Systems and Their Application in Healthcare. *Technical Journal, University of Engineering and Technology (UET) , 20*(IV).

Kanburoglu, A. B. (2018). *Graph clustering approach to sentiment analysis.* Işık University.

Kavipriya, P., & Karthikeyan , K. (2017, November). A Comparative Study of Feature Selection Algorithms in Data Mining. *International Journal of Advanced Research in Computer and Communication Engineering, 6*(11). doi:10.17148/IJARCCE.2017.61109 58 7

Kavipriya, P., & Karthikeyan, K. (2017, November). A Comparative Study of Feature Selection Algorithms in Data Mining. *International Journal of Advanced Research in Computer and Communication Engineering, 6*(11). doi:10.17148/IJARCCE.2017.61109 58 7

Kaya, M. (2013). *Sentiment analysis of Turkish political columns with transfer learning.* Middle East Technical University.

Kim, H. D., Ganesan, K., & Sondhi, P. (2017). *Comprehensive Review of Opinion Summarization*. Retrieved from http://kavita-ganesan.com/sites/default/files/survey_opinionSummarization.pdf

Kuo, N., & Chung, Y. (2012). The Application of Healthcare Information System for Comprehensive Geriatric Assessment. *National Chengchi University & Airiti Press Inc, Vol. 17*(2), 87-98.

Lovins, J. B. (1968). Development of a Stemming Algorithm. *Linguistics, Mechanical Translation and Computational, 11*(1-2), 22-31.

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press. doi:http://nlp.stanford.edu/IR-book/html/htmledition/properties-of-naive-bayes-1.html

Martínez-Cámara E., M.-V. M.-L. (2011). Opinion Classification Techniques Applied to a Spanish Corpus. *In: Muñoz R., Montoyo A., Métais E. (eds) Natural Language Processing and Information Systems. NLDB 2011. Lecture Notes in Computer Science. 6716.* Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-22327-3_17

Mostafa, M. M. (2013). More than words: Social networks' text mining for consumer brand sentiments. *Expert Syst. Appl*, *40*, pp. 4241-4251. doi:https://doi.org/10.1016/j.eswa.2013.01.019

Mountassir, A., Benbrahim, H., & Berrada, I. (2013). Sentiment classification on arabic corpora: A preliminary cross-study. Document numérique. *16*(1), 73-96. doi:https://doi.org/10.3166/DN.16.1.73-96

Na, J. T. (2010). Comparing sentiment expression in movie reviews from four online genres. *Online Information Review, 34*(2), 317-338. doi:10.1108/14684521011037016

Nasiri, M., Minaei, B., & Amir, K. A. (2016). Dynamic recommendation: disease prediction and pre-vention using recommender system. *1*(1), 13-17. doi:doi:10.15171/ijbsm.2016.04

Nigam, K., Mccallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2-3), 103-134.

Oflazer, K. T. (1993). Two Level Description of Turkish Morphology. *In: Proceedings of EACL'98.* Utrecht, The Netherlands.

Orhan, Z., Giriş, S. F., Ceyhan, M., & Domnori, E. (2014). Customer Satisfaction Measurement Tool by Analysing Turkish Product Reviews. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 7*(1), 12–18. Retrieved from https://dergipark.org.tr/en/pub/tb

Orhan, Z., Giriş, S. F., Ceyhan, M., & Mercan, M. (2014). Customer Satisfaction Measurement Tool by Analysing Turkish Product Reviews. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve*

Mühendisliği Dergisi, 7(1), 12-18. Retrieved 2016, from https://dergipark.org.tr/en/pub/tbbmd/issue/22247/238816

Orhan, Z., Mercan, M., & Gökgöl, M. K. (2020). A New Digital Mental Health System Infrastructure for Diagnosis of Psychiatric Disorders and Patient Follow-Up by Text Analysis in Turkish. In *Proceedings of International Conference on Medical and Biological Engineering* (In: Badnjevic A., Škrbić R., Gurbeta Pokvić L. (eds) ed., Vol. 73, pp. 395-402). Springer, Cham. Retrieved from http://doi-org-443.webvpn.fjmu.edu.cn/10.1007/978-3-030-17971-7_59

Pandya, R., & Pandya, J. (2015). C5.0 algorithms to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications, 117*(16).

Pang, B. &. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *ArXiv, cs.CL*. doi:0409058

Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Proceedings Of The ACL-02 Conference On Empirical Methods In Natural Language Processing - EMNLP '02.* doi:https://doi.org/10.3115/1118693.1118704

Piccini, N. (2019, July 16). *R-bloggers*. Retrieved from 101 Machine Learning Algorithms for Data Science with Cheat Sheets: https://www.r-bloggers.com/2019/07/101-machine-learning-algorithms-for-data-science-with-cheat-sheets/

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization,. In *in Schölkopf, B., Burges, C., and Smola, A. (Eds.), Advances in Kernel Methods - Support Vector Learning* (pp. 185-208). MIT Press.

Porter, M. (1980). An algorithm for suffix stripping. *Program, 14*, 130-137.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA.: Morgan Kaufmann Publishers.

Rennie, J. D., & Rifkin, R. (2001). Improving Multiclass Text Classification with the Support Vector Machine. *Computer Science*.

Rodrigues, Joel J. P. C. (2010). Preface. In *Health Information Systems: Concepts, Methodologies, Tools, and Applications* (pp. i-vi). Hershey, PA.

Rosenblatt, F. (1957). *The Perceptron: A Perceiving and Recognizing Automaton.* Cornell Aeronautical Laboratory, Buffalo, New York.

Sassano, M. (2008). An Experimental Comparison of the Voted Perceptron and Support Vector Machine in Japanese Analysis Tasks. *International Joint Conference on Natural Language Processing*, 829-834.

Sassano, M. (2008). An Experimental Comparison of the Voted Perceptron and Support Vector Machine in Japanese Analysis Tasks. *ACLWEB*, 829-834.

Sayce, D. (2020). *The Number of tweets per day in 2020*. Retrieved September 2020, from https://www.dsayce.com/social-media/tweets-day/

Schneider, K. M. (2005). Techniques for improving the performance of naive bayes for text classification. *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 682-693). Berlin Heidelberg: Springer.

Schölkopf, B., & Smola, A. J. (2001). Learning With Kernels.

Sindhwani V., M. P. (2008). Document-Word Co-regularization for Semi-supervised Sentiment Analysis. *Eighth IEEE International Conference on Data Mining*, (pp. 1025-1030). Pisa. doi:10.1109/ICDM.2008.113

Singhal, M., & Sharma, R. (2014, August). Optimization of Naïve Bayes Data Mining Classification Algorithm. *International Journal for research in applied Science and Engineering Technology, 2*(VIII).

Solak, A. C. (1994). *Effects of Stemming on Turkish Text Retrieval. Technical Report BU-CEIS-94-20. .* Bilkent University, Department of Computer Engineering and Information Science, Ankara, Turkey.

Statista. (2020). Retrieved from https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/

Şeker, Ş. E. (2013). *İş Zekası ve Veri Madenciliği (Weka ile)* (Vol. 1). Turkiye: CİNİUS.

Taboada, M. B. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*, 267–307. doi:10.1162/COLI_a_00049

Thet, T. N. (2008). Sentiment classification of movie reviews using multiple perspectives. *Proceedings of the international conference on Asian digital libraries (ICADL)*, 184–193.

Turkmen, H. (2016). *Discovering product features from Turkish reviews by using aspect based sentiment analysis.* Kocaeli University.

Turksent. (2010). Retrieved May 2017, from Annotation tool developed specifically for manual sentiment analysis of social media posts: http://www.turksent.com

Tüfekci, N., & Asığbulmuş, H. (2016). The Factors that Effective in the Choice of Hospital and Patient Satisfaction: The Sample of Isparta. *Journal of Current Researches on Health Sector, 6*(2), 71-92.

Vateekul, P., & Kubat, M. (2009). Fast induction of multiple decision trees in text categorization from large scale, imbalanced, and multi-label data. *IEEE International Conference Data Mining Workshops* (pp. 320-325). IEEE.

Vryniotis, V. (2013). *Machine Learning Tutorial: The Naive Bayes Text Classifier*. Retrieved from Machine Learning Blog & Software Development News: http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier

Vural, A. G. (2013). A framework for sentiment analysis in Turkish: Application to polarity detection of movie reviews in Turkish. *Computer and Information Sciences III*, 437-445.

Wang, Y., Zhou, Z., Jin, S., & Liu, D. &. (2017, 10). Comparisons and Selections of Features and Classifiers for Short Text Classification. *IOP Conference Series: Materials Science and Engineering, 261*, 12-18. doi:10.1088/1757-899x/261/1/012018

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition.* San Francisco: Morgan Kaufmann.

Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., & Cunnigham, S. J. (1999). Weka: Practical Machine Learning Tools and Techniques with Java Implementations. *Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems*, (pp. 192-196).

Yuan, L. (2010). An improved Naive Bayes text classification algorithm in Chinese information processing. *Science*, pp. 267-269.

Yuret, D., & Türe, F. (2006). Learning morphological disambiguation rules for Turkish. *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (pp. 328-334). USA: Association for Computational Linguistics. doi:10.3115/1220835.1220877

Yurt, E. A. (2014). *Duygu Analizi*. Retrieved from http://www.slideshare.net/webender/duygu-analizi

Zaman, B. (2016, December 8). *Stack Exchange*. Retrieved from Cross Validated: What's the advantages of bayesian version of linear regression, logistic regression etc: https://stats.stackexchange.com/questions/43436/whats-the-advantages-of-bayesian-version-of-linear-regression-logistic-regress

Zephoria - Digital Marketing. (2020). Retrieved from https://zephoria.com/top-15-valuable-facebook-statistics/

Zhang, H. (2004). The optimality of Naive Bayes. *3*.

Zhuang, L. J.-y. (2006). Movie Review Mining and Summarization. *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM).*

Zunic, A., Corcoran, P., & Spasic, I. (2020). Sentiment Analysis in Health and Well-Being: Systematic Review. *JMIR medical informatics, 8*(1). doi:10.2196/16023

APPENDIX A

Customer Products Review Data Samples

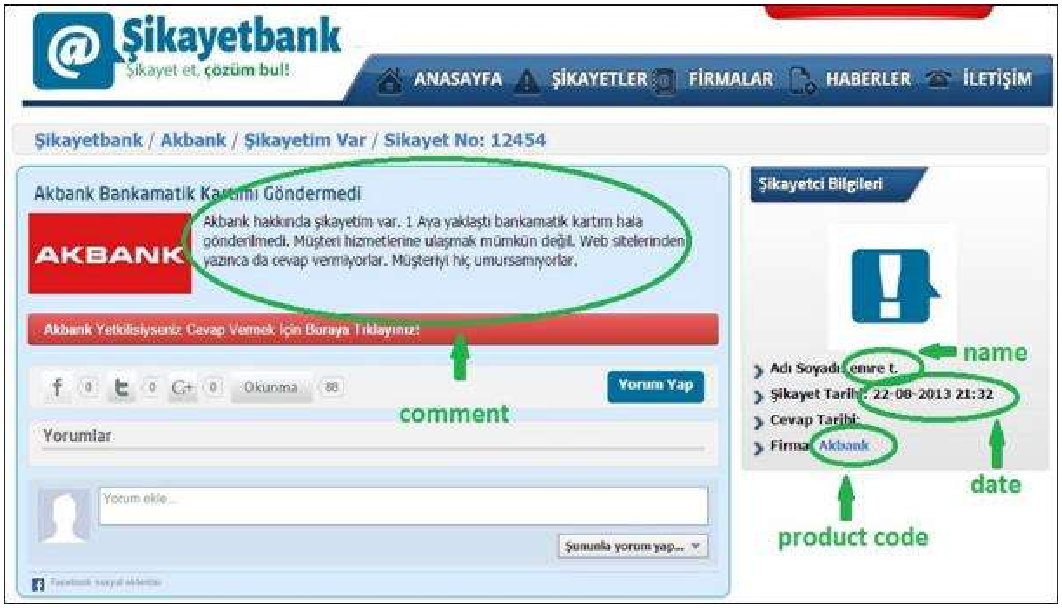


Figure 17. The structure of negative reviews on the web site

Figure 18. The structure of negative reviews on the web site

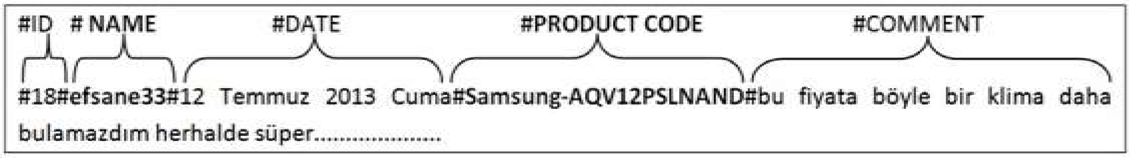


Figure 19. The extracted text data format of one product comment

Patients Anamnesis Data Samples

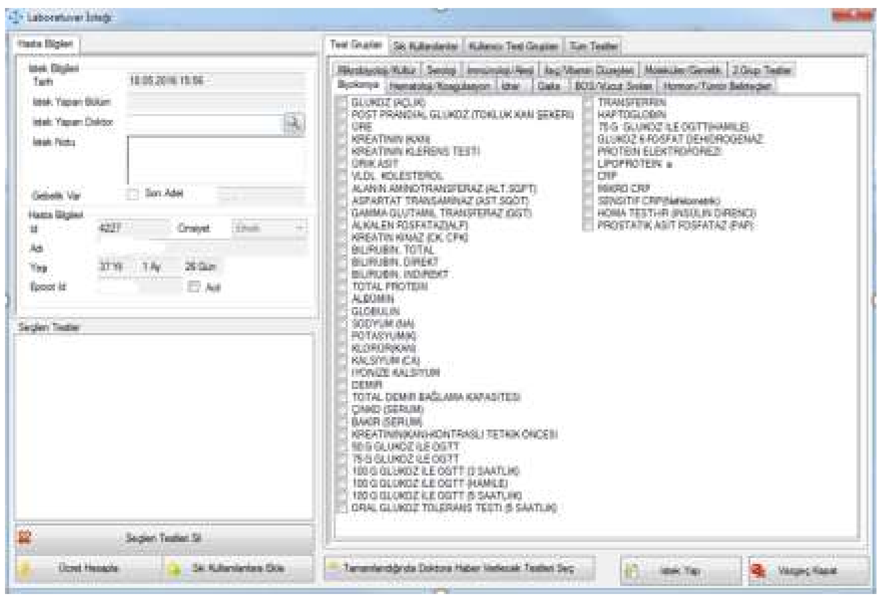


Figure 20. The classified groups of medical tests**.**

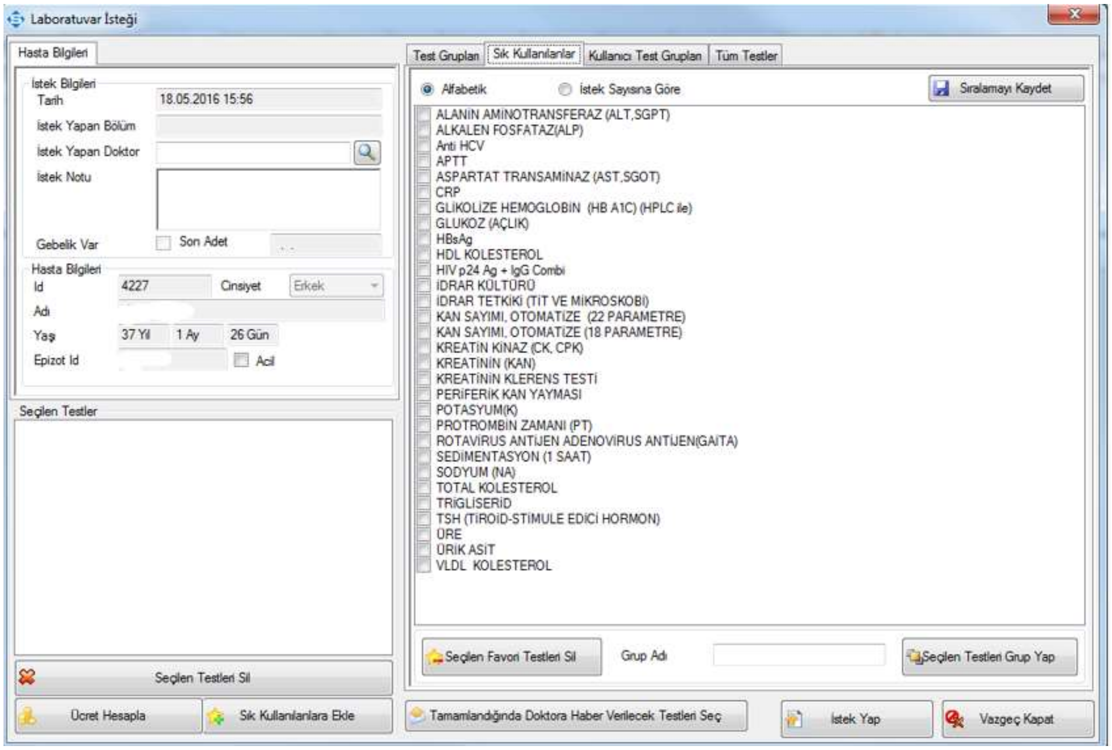


Figure 21. Most frequently used medical tests for fast access.

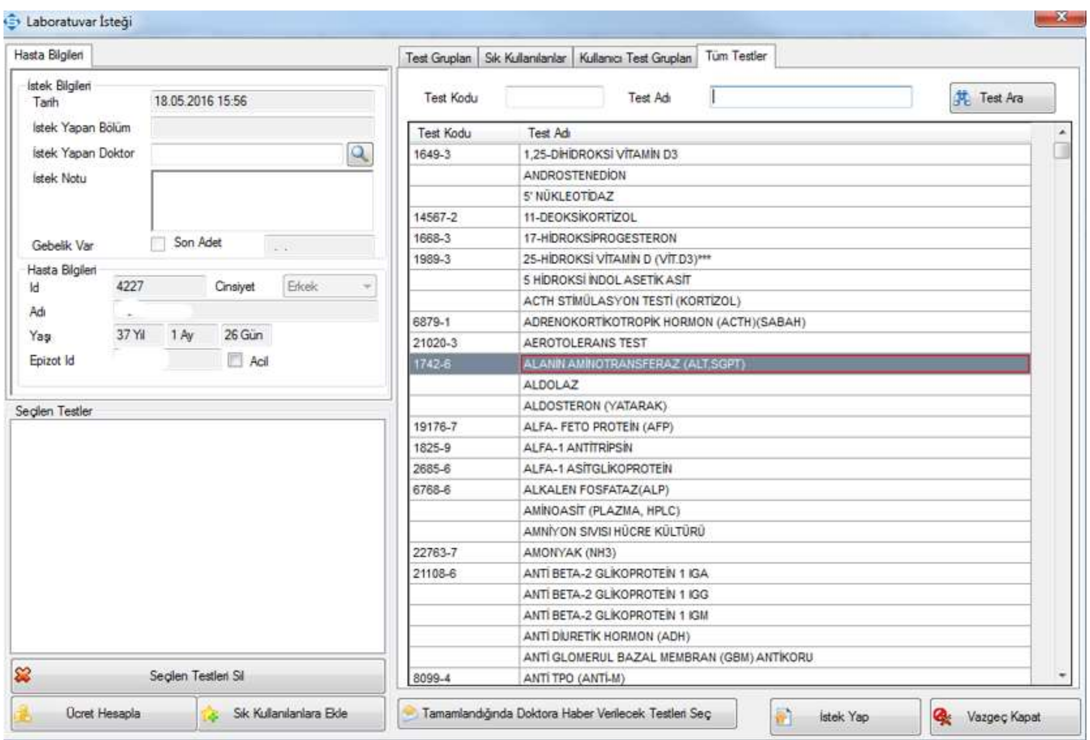


Figure 22 Selection screen for all medical tests

Table 38: Customer Product reviews Data Morphological Analysis and Disambiguation

Morphological Analysis Input Data (Positive example) ==> Partial Morphological Analysis Result			
<S>	<S>	<S>	
yüksek	yüksek	yüksek (high)	+Adj
kaliteye	yüksek	yük (load)	+Noun+A3sg+Pnon+Nom^DB
maximum			+Verb +Zero+Cond+A1pl
performansa	kaliteye	kalite (quality)	+Noun+A3sg+Pnon+Dat
ihtiyaç	,	,	+Punc
duyuyorsanız	maximum	maximu (the maxim?)	+Noun+A3sg+P1sg+Nom
bu	maximum	maximum (maximum)	+Noun+A3sg+Pnon+Nom
ürünü			
kaçırmayın	performansa	performans (performance)	+Noun+A3sg+Pnon+Dat
.			
</S>	ihtiyaç	ihtiyaç (need)	+Noun+A3sg+Pnon+Nom
<S>			
	duyuyorsanız	duy (feel)	+Verb+Pos+Prog1+Cond+A2pl
	...		

Disambiguated Sample Data		
<S>	<S>	
yüksek	yüksek	+Adj
kaliteye	kalite	+Noun+A3sg+Pnon+Dat
,	,	+Punc
maximum	maximu	+Noun+A3sg+P1sg+Nom
performansa	performans	+Noun+A3sg+Pnon+Dat
ihtiyaç	ihtiyaç	+Noun+A3sg+Pnon+Nom
duyuyorsanız	duy	+Verb+Pos+Prog1+Cond+A2pl
bu	bu	+Det
ürünü	ürün	+Noun+A3sg+P3sg+Nom
kaçırmayın	kaç	+Verb^DB+Verb+Caus+Neg+Imp+A2pl
.	.	+Punc
</S>	</S>	
<S>	<S>	
.	.	+Punc
</S>	</S>	

APPENDIX B

```
erased from both:   hafta    N:18        P:14
erased from both:   fena     N:0         P:1
erased from both:   baskı    N:1         P:6
erased from both:   işlev    N:0         P:1
erased from both:   bari     N:0         P:0
erased from both:   işlem    N:19        P:10
erased from both:   servi    N:13        P:1
erased from both:   akşam    N:2         P:1
erased from both:   düşün-   N:0         P:4
erased from both:   adres    N:15        P:0
erased from both:   çarp     N:0         P:0
erased from both:   üst      N:6         P:7
erased from both:   s        N:0         P:3
erased from both:   inç      N:0         P:0
erased from both:   sabah    N:1         P:1
```

Figure 23 A view of features erased from both positive and negative lists of customer product reviews domain, since their respective frequencies fall within a certain thresh percentage

Figure 24 NB (Binary) Results for Movies Dataset.

```
SONUCLAR ÖZELLİK 0 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       24        1 |          ||
 OLUMSUZ=1  |        2       23 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.923077    0.960000    0.941176 |
 OLUMSUZ=1  |  0.958333    0.920000    0.938775 |
==================================================

Doğruluk oranı Acc=  0.940000
```

```
SONUCLAR ÖZELLİK 1 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       24        1 |          ||
 OLUMSUZ=1  |        2       23 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.923077    0.960000    0.941176 |
 OLUMSUZ=1  |  0.958333    0.920000    0.938775 |
==================================================

Doğruluk oranı Acc=  0.940000
```

```
SONUCLAR ÖZELLİK 2 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       23        2 |          ||
 OLUMSUZ=1  |        2       23 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.920000    0.920000    0.920000 |
 OLUMSUZ=1  |  0.920000    0.920000    0.920000 |
==================================================

Doğruluk oranı Acc=  0.920000
```

```
SONUCLAR ÖZELLİK 3 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       23        2 |          ||
 OLUMSUZ=1  |        2       23 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.920000    0.920000    0.920000 |
 OLUMSUZ=1  |  0.920000    0.920000    0.920000 |
==================================================

Doğruluk oranı Acc=  0.920000
```

```
SONUCLAR ÖZELLİK 4 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       23        2 |          ||
 OLUMSUZ=1  |        3       22 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.884615    0.920000    0.901961 |
 OLUMSUZ=1  |  0.916667    0.880000    0.897959 |
==================================================

Doğruluk oranı Acc=  0.900000
```

```
SONUCLAR ÖZELLİK 5 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       23        2 |          ||
 OLUMSUZ=1  |        3       22 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.884615    0.920000    0.901961 |
 OLUMSUZ=1  |  0.916667    0.880000    0.897959 |
==================================================

Doğruluk oranı Acc=  0.900000
```

```
SONUCLAR ÖZELLİK 6 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       24        1 |          ||
 OLUMSUZ=1  |        6       19 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.800000    0.960000    0.872727 |
 OLUMSUZ=1  |  0.950000    0.760000    0.844444 |
==================================================

Doğruluk oranı Acc=  0.860000
```

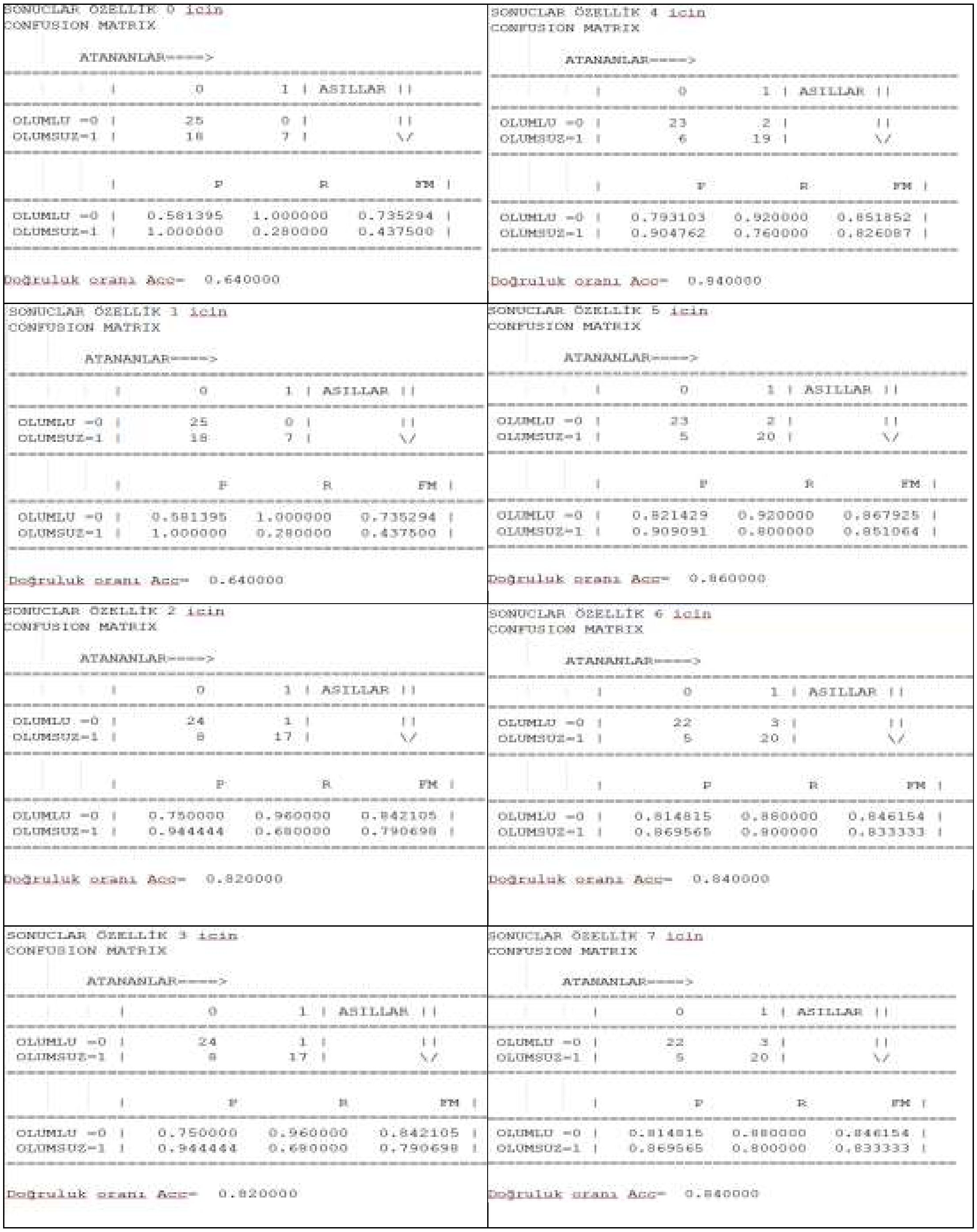
```
SONUCLAR ÖZELLİK 7 icin
CONFUSION MATRIX

        ATANANLAR====>
==================================================
            |        0        1 | ASILLAR ||
==================================================
 OLUMLU  =0 |       24        1 |          ||
 OLUMSUZ=1  |        6       19 |          \/
==================================================

            |        P           R          FM |
==================================================
 OLUMLU  =0 |  0.800000    0.960000    0.872727 |
 OLUMSUZ=1  |  0.950000    0.760000    0.844444 |
==================================================

Doğruluk oranı Acc=  0.860000
```

Figure 25 MNB Results for Movies Dataset.



SONUCLAR ÖZELLİK 0 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	25	0	||
OLUMSUZ=1	18	7	\/

	P	R	FM
OLUMLU =0	0.581395	1.000000	0.735294
OLUMSUZ=1	1.000000	0.280000	0.437500

Doğruluk oranı Acc= 0.640000

SONUCLAR ÖZELLİK 4 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	23	2	||
OLUMSUZ=1	6	19	\/

	P	R	FM
OLUMLU =0	0.793103	0.920000	0.851852
OLUMSUZ=1	0.904762	0.760000	0.826087

Doğruluk oranı Acc= 0.840000

SONUCLAR ÖZELLİK 1 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	25	0	||
OLUMSUZ=1	18	7	\/

	P	R	FM
OLUMLU =0	0.581395	1.000000	0.735294
OLUMSUZ=1	1.000000	0.280000	0.437500

Doğruluk oranı Acc= 0.640000

SONUCLAR ÖZELLİK 5 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	23	2	||
OLUMSUZ=1	5	20	\/

	P	R	FM
OLUMLU =0	0.821429	0.920000	0.867925
OLUMSUZ=1	0.909091	0.800000	0.851064

Doğruluk oranı Acc= 0.860000

SONUCLAR ÖZELLİK 2 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	24	1	||
OLUMSUZ=1	8	17	\/

	P	R	FM
OLUMLU =0	0.750000	0.960000	0.842105
OLUMSUZ=1	0.944444	0.680000	0.790698

Doğruluk oranı Acc= 0.820000

SONUCLAR ÖZELLİK 6 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	22	3	||
OLUMSUZ=1	5	20	\/

	P	R	FM
OLUMLU =0	0.814815	0.880000	0.846154
OLUMSUZ=1	0.869565	0.800000	0.833333

Doğruluk oranı Acc= 0.840000

SONUCLAR ÖZELLİK 3 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	24	1	||
OLUMSUZ=1	8	17	\/

	P	R	FM
OLUMLU =0	0.750000	0.960000	0.842105
OLUMSUZ=1	0.944444	0.680000	0.790698

Doğruluk oranı Acc= 0.820000

SONUCLAR ÖZELLİK 7 icin
CONFUSION MATRIX

ATANANLAR====>

	0	1	ASILLAR
OLUMLU =0	22	3	||
OLUMSUZ=1	5	20	\/

	P	R	FM
OLUMLU =0	0.814815	0.880000	0.846154
OLUMSUZ=1	0.869565	0.800000	0.833333

Doğruluk oranı Acc= 0.840000

Figure 26 Dictonary Method’s Results for Movies Dataset.

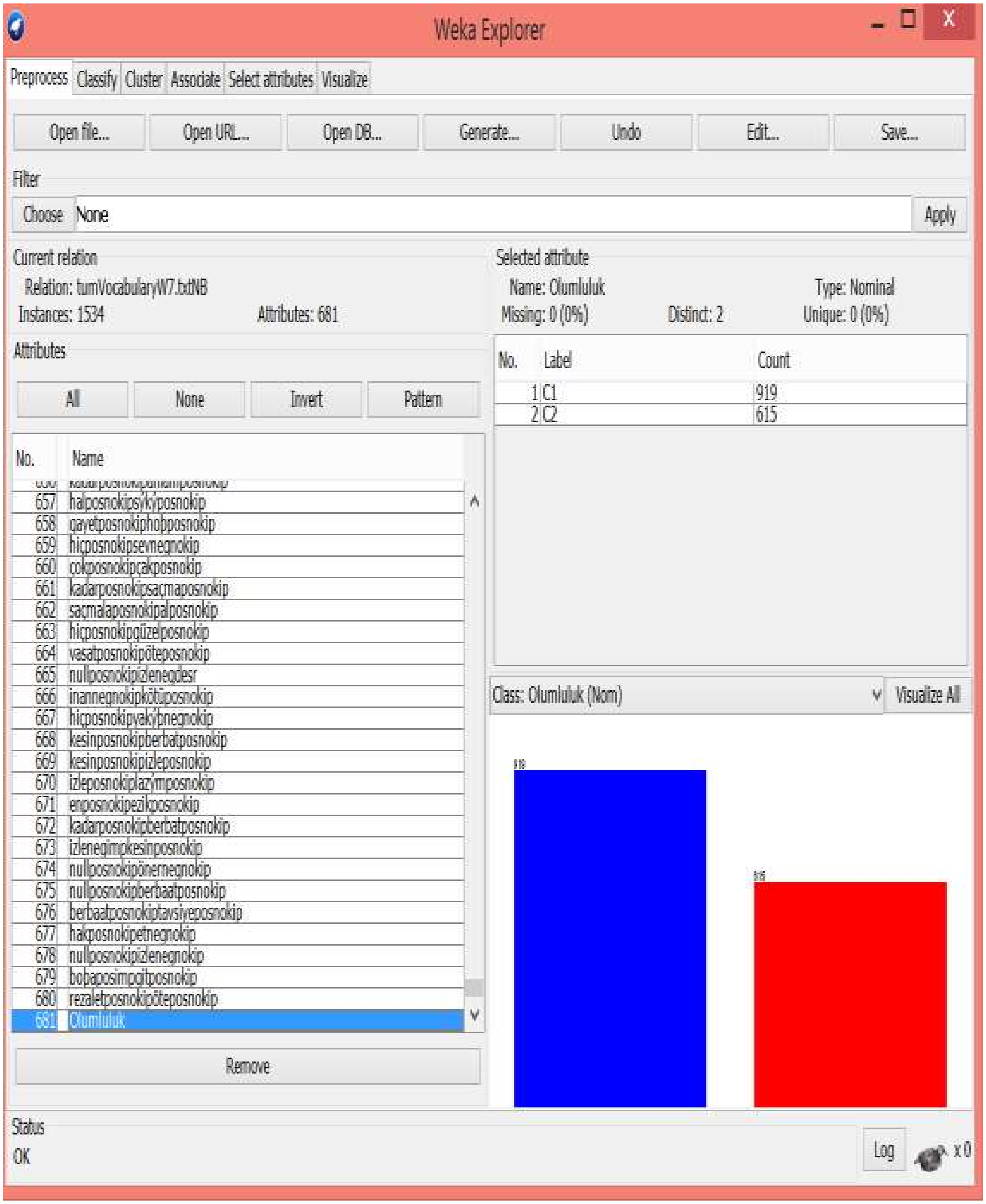


Figure 27 Weka preprocessing panel

Classifier output

```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayesMultinomial
Relation:     tumVocabularyW7.txtNB
Instances:    1534
Attributes:   681
[list of attributes omitted]
Test mode:user supplied test set:     50instances

=== Classifier model (full training set) ===

The independent probability of a class
--------------------------------------
C1      0.5989583333333334
C2      0.4010416666666667

The probability of a word given the class
-----------------------------------------
        C1      C2
nullposnokipkültposnokip       9.560229445506689E-4    2.937720329024678E-4
nullposnokipmuhakkakposnokip   5.736137667304014E-4    2.937720329024678E-4
muhakkakposnokipizleposnokip   5.736137667304014E-4    2.937720329024678E-4
nullposnokiplazýmposnokip      5.736137667304014E-4    2.937720329024678E-4
nullposnokipharikaposnokip     0.04340344168260037     2.937720329024678E-4
nullposnokipbaþarýposnokip     0.005544933078393882    2.937720329024678E-4
nullposnokipmutlakaposnokip    0.00248565965583174     2.937720329024678E-4
nullposnokiphayranposnokip     0.006309751434034417    2.937720329024678E-4
nullposnokipmükemmelposnokip   0.03422562141491395 5   2.937720329024678E-4
nullposnokipefsaneposnokip     0.012428298279158 7     2.937720329024678E-4
```

Figure 28 Weka accuracies for each feature for the movies database

```
Time taken to build model: 0.02 seconds

=== Predictions on test split ===

inst#,    actual, predicted, error, probability distribution
     1      1:C1       1:C1         *0.966  0.034
     2      1:C1       1:C1         *1      0
     3      1:C1       1:C1         *0.745  0.255
     4      1:C1       1:C1         *0.994  0.006
     5      1:C1       1:C1         *1      0
     6      1:C1       1:C1         *0.995  0.005
     7      1:C1       1:C1         *0.984  0.016
     8      1:C1       1:C1         *0.596  0.404
     9      1:C1       1:C1         *0.996  0.004
    10      1:C1       1:C1         *0.984  0.016
    11      1:C1       1:C1         *0.991  0.009
    12      1:C1       1:C1         *0.995  0.005
    13      1:C1       1:C1         *0.599  0.401
    14      1:C1       1:C1         *0.984  0.016
    15      1:C1       1:C1         *0.553  0.447
    16      1:C1       1:C1         *0.599  0.401
    17      1:C1       1:C1         *0.984  0.016
    18      1:C1       1:C1         *0.994  0.006
    19      1:C1       1:C1         *0.599  0.401
    20      1:C1       1:C1         *0.993  0.007
    21      1:C1       1:C1         *0.829  0.171
    22      1:C1       1:C1         *0.745  0.255
    23      1:C1       1:C1         *0.958  0.042
    24      1:C1       1:C1         *0.993  0.007
    25      1:C1       1:C1         *0.994  0.006
    26      2:C2       1:C1      +  *0.599  0.401
    27      2:C2       1:C1      +  *0.599  0.401
    28      2:C2       1:C1      +  *0.882  0.118
    29      2:C2       2:C2          0.013 *0.987
    30      2:C2       2:C2          0.014 *0.986
    31      2:C2       2:C2          0.395 *0.605
    32      2:C2       2:C2          0      *1
    33      2:C2       1:C1      +  *0.918  0.082
    34      2:C2       2:C2          0.163 *0.837
    35      2:C2       2:C2          0.325 *0.675
    36      2:C2       2:C2          0.061 *0.939
    37      2:C2       2:C2          0.042 *0.958
    38      2:C2       2:C2          0.016 *0.984
    39      2:C2       2:C2          0.04  *0.96
    40      2:C2       2:C2          0.007 *0.993
```

Figure 29 Weka accuracies for each feature for the movies database

```
    40       2:C2       2:C2          0.007 *0.993
    41       2:C2       2:C2          0.245 *0.755
    42       2:C2       2:C2          0.493 *0.507
    43       2:C2       2:C2          0     *1
    44       2:C2       2:C2          0.007 *0.993
    45       2:C2       2:C2          0.081 *0.919
    46       2:C2       1:C1      +  *0.599  0.401
    47       2:C2       2:C2          0.035 *0.965
    48       2:C2       2:C2          0.035 *0.965
    49       2:C2       2:C2          0.107 *0.893
    50       2:C2       1:C1      +  *0.599  0.401

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          44               88      %
Incorrectly Classified Instances         6               12      %
Kappa statistic                          0.76
Mean absolute error                      0.1841
Root mean squared error                  0.3061
Relative absolute error                 36.8246 %
Root relative squared error             60.0514 %
Total Number of Instances               50

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.24      0.806     1         0.893      0.952    C1
                 0.76      0         1         0.76      0.864      0.952    C2
Weighted Avg.    0.88      0.12      0.903     0.88      0.878      0.952

=== Confusion Matrix ===

  a  b   <-- classified as
 25  0 |  a = C1
  6 19 |  b = C2
```

***Figure** 30* Weka test results for Movies Dataset.

Table 39 Weka NBM Results for Movies Dataset.

=== Run information === Scheme:weka.classifiers.bayes.NaiveBayesMultinomial Relation: tumVocabularyW0.txtNB Instances: 1534 Attributes: 2021 Correctly Classified Instances 47 94 % Incorrectly Classified Instances 3 6 % === Confusion Matrix === a b <-- classified as 24 1 | a = C1 2 23 | b = C2	=== Run information === Scheme:weka.classifiers.bayes.NaiveBayesMultinomial Relation: tumVocabularyW4.txtNB Instances: 1534 Attributes: 405 Correctly Classified Instances 46 92 % Incorrectly Classified Instances 4 8% === Confusion Matrix === a b <-- classified as 24 1 | a = C1 3 22 | b = C2
=== Run information === Scheme:weka.classifiers.bayes.NaiveBayesMultinomial Relation: tumVocabularyW1.txtNB Instances: 1534 Attributes: 2083 Correctly Classified Instances 47 94 % Incorrectly Classified Instances 3 6 % === Confusion Matrix === a b <-- classified as 24 1 | a = C1 2 23 | b = C2	=== Run information === Scheme:weka.classifiers.bayes.NaiveBayesMultinomial Relation: tumVocabularyW5.txtNB Instances: 1534 Attributes: 414 Correctly Classified Instances 45 90 % Incorrectly Classified Instances 5 10 % === Confusion Matrix === a b <-- classified as 23 2 | a = C1 3 22 | b = C2
=== Run information === Scheme:weka.classifiers.bayes.NaiveBayesMultinomial Relation: tumVocabularyW2.txtNB Instances: 1534 Attributes: 9523 Correctly Classified Instances 47 94 % Incorrectly Classified Instances 3 6 % === Confusion Matrix ===	=== Run information === Scheme:weka.classifiers.bayes.NaiveBayesMultinomial Relation: tumVocabularyW6.txtNB Instances: 1534 Attributes: 668 Correctly Classified Instances 44 88 % Incorrectly Classified Instances 6 12 % === Confusion Matrix ===

<table>
<tr>
<td>a b <-- classified as
24 1 | a = C1
2 23 | b = C2</td>
<td>a b <-- classified as
25 0 | a = C1
6 19 | b = C2</td>
</tr>
<tr>
<td>=== Run information ===
Scheme:weka.classifiers.bayes.NaiveBayesMultinomial
Relation: tumVocabularyW3.txtNB
Instances: 1534
Attributes: 9656

Correctly Classified Instances 47 94 %
Incorrectly Classified Instances 3 6 %
=== Confusion Matrix ===

a b <-- classified as
24 1 | a = C1
2 23 | b = C2</td>
<td>=== Run information ===
Scheme:weka.classifiers.bayes.NaiveBayesMultinomial
Relation: tumVocabularyW7.txtNB
Instances: 1534
Attributes: 681

Correctly Classified Instances 44 88 %
Incorrectly Classified Instances 6 12 %
=== Confusion Matrix ===

a b <-- classified as
25 0 | a = C1
6 19 | b = C2</td>
</tr>
</table>

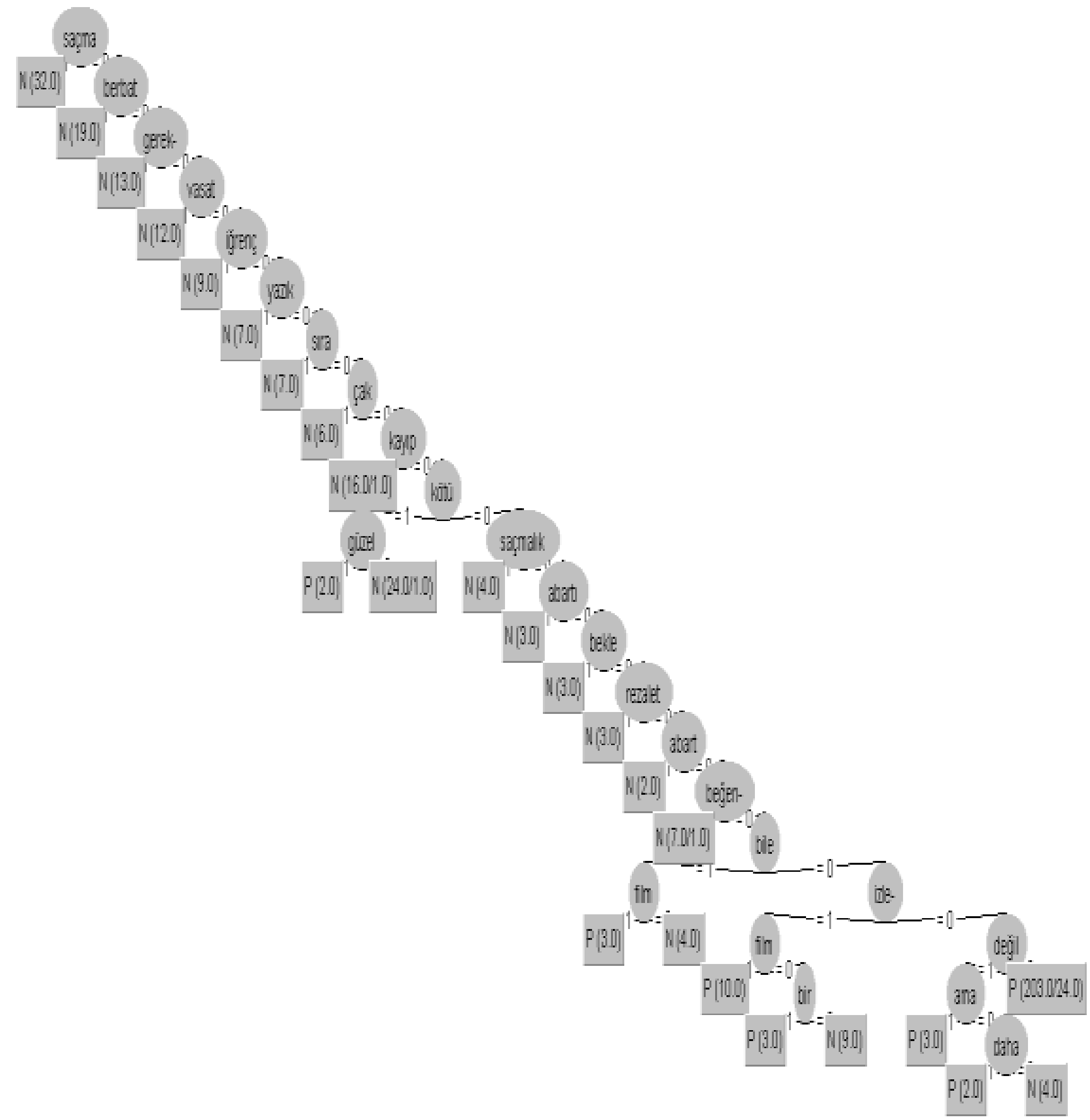


Figure 31 A sample view of the J48 tree of unigrams binary vector method for movie reviews

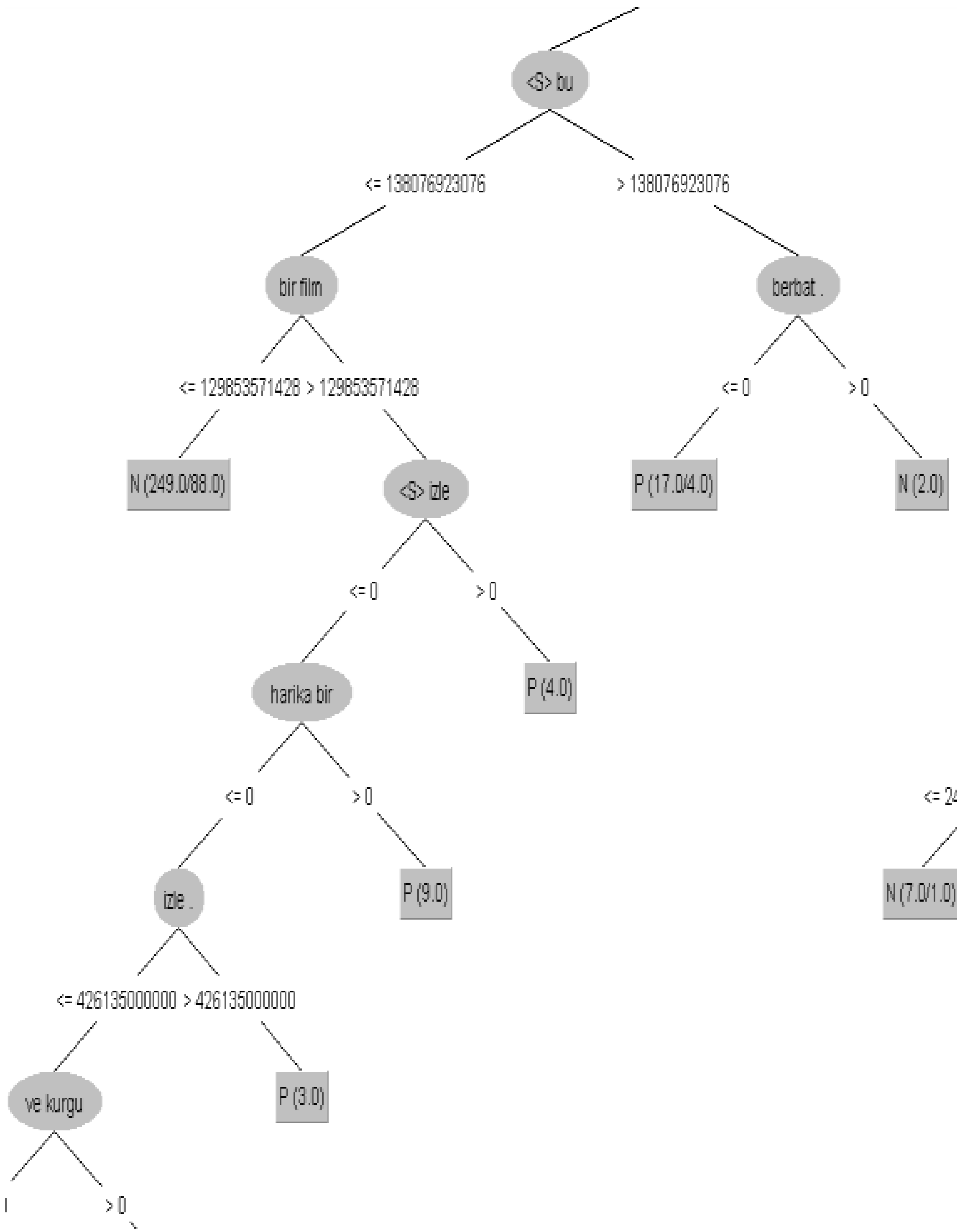


Figure 32 A branch of the J48 tree of bigrams frecuency vector method for movie reviews

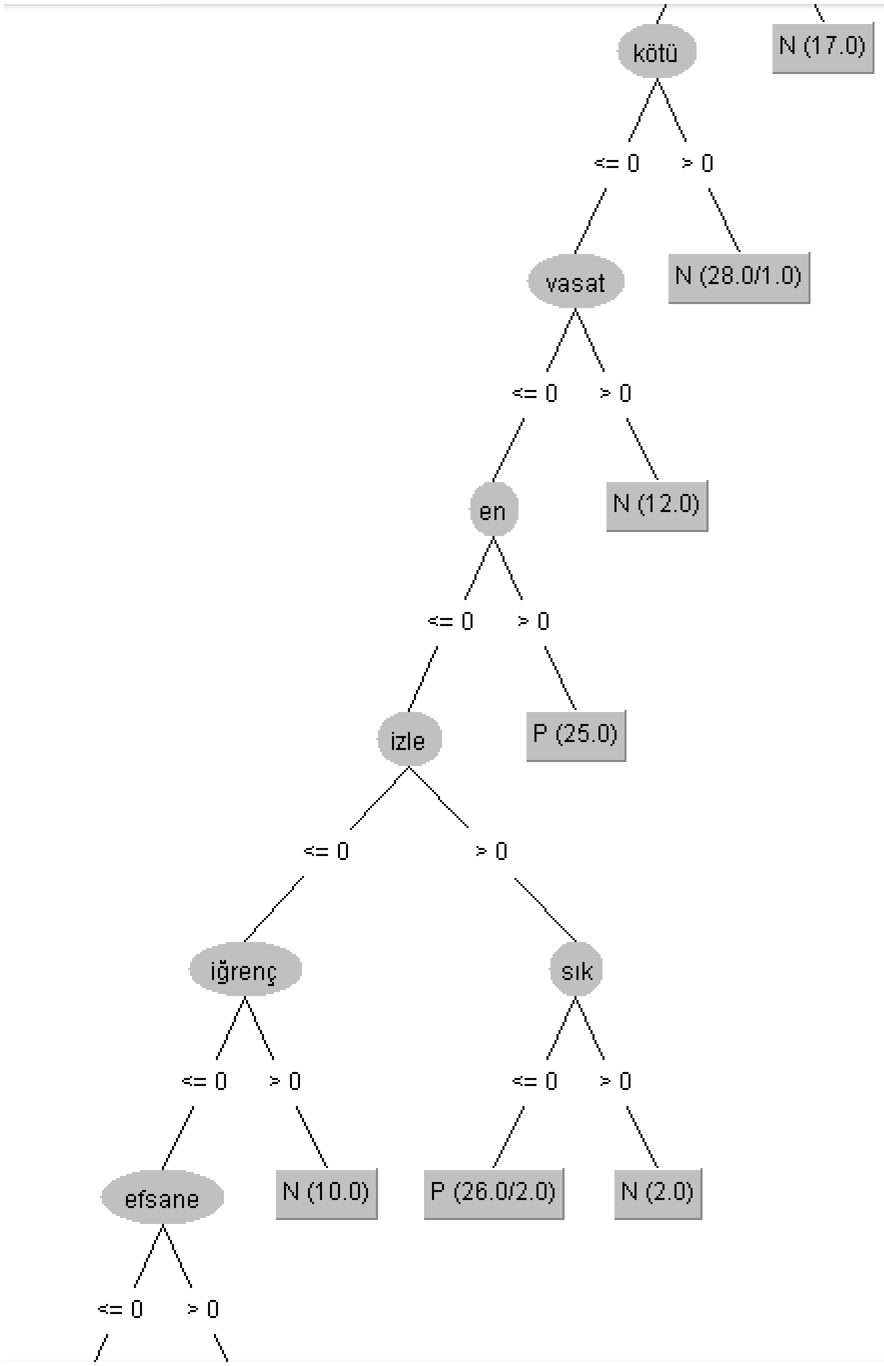


Figure 33 A branch of the J48 tree of unigrams frequency vector method for movie reviews

Figure 34 Word cloud of mostly used customers' product reviews words list

Figure 35 Word cloud of mostly used health service satisfaction words list

Nr	Turkish	Translation	Nr	Turkish	Translation
1	firma	company	26	gönder	send
2	ver	give	27	çalış	work
3	ile	with	28	hal	status
4	talep	request	29	mağduriyet	victimization
5	fatura	invoice	30	ara	call
6	başla	start	31	yetkili	authorized
7	para	money	32	iste-	request (not)
8	süper	super	33	garanti	warranty
9	iade	return	34	güncelle	update
10	geri	back	35	şikayet	complaint
11	tarih	history	36	önce	before
12	gerek	need	37	mağdur	victim
13	için	for	38	çok	very
14	bir	one	39	bilgi	information
15	bil	know	40	daha	more
16	sonra	after	41	gerçekten	really
17	gel-	come (not)	42	gel	come
18	ben	I	43	kullan	use
19	konu	subject	44	kadar	until
20	oyun	game	45	ve	and
21	hız	speed	46	kendi	own
22	ilgi-	interest (not)	47	tv	TV
23	iste	request	48	tl	Turkish lira
24	her	each	49	iş	business
25	yine	again	50	performans	performance

Nr	Turkish	Translation
51	ol	be
52	ne	what
53	sor	ask
54	durum	status
55	servis	service
56	yap-	do (not)
57	yani	so
58	el	hand
59	böyle	such
60	de	also
61	kalite	quality
62	bu	this
63	bi	one
64	at	through
65	al	get
66	yok	no
67	karşıla	meet
68	yıl	year
69	tüket	consume
70	yer	place
71	pişman	regret
72	gibi	as
73	acil	urgent
74	fazla	much
75	şey	thing
76	yap	do
77	yan	side
78	güzel	beautiful
79	şekil	figure

Nr	Turkish	Translation
80	gün	day
81	et-	meat (not)
82	özellik	property
83	tekrar	again
84	tümsayılar	integers
85	gör	see
86	büyük	big
87	kul	ash
88	diğer	other
89	rağmen	despite
90	müthiş	wonderful
91	belir	appear
92	teslim	delivery
93	defa	times
94	kez	times
95	söyle	say
96	çözüm	solution
97	kredi	credit
98	kat	floor
99	numara	number
100	ulaş-	reach (not)
101	cevap	reply
102	ancak	but
103	bekle	wait
104	hata	error
105	mükemmel	excellent
106	merkez	center
107	hala	still
108	samsung	samsung

Nr	Turkish	Translation
109	tasarım	design
110	yıka	wash
111	com	com
112	neden	cause, why
113	telefon	telephone
114	muhteşem	spectacular
115	iyi	good
116	mesaj	message
117	müşteri	customer
118	özellikle	especially

Nr	Turkish	Translation
119	fakat	but
120	tavsiye	advice
121	teknoloji	technology
122	fiyat	price
123	hafta	week
124	işlem	operation
125	düşün-	think (not)
126	şube	branch
127	diye	so that

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Migena Ceyhan

Nationality: Albanian

Date and Place of Birth: 26.06.1978, Shkoder, Albania

Marital Status: Married

Phone: +355 696703027
Fax:

E-mail: mceyhan@epoka.edu.al

EDUCATION

BSc. in Computer Engineering and Information Sciences, Bilkent University, Ankara, Turkey. July 2001

MSc. in Computer Engineering and Information Sciences, Bilkent University, Ankara, Turkey. July 2001

MSc. in Information Systems Development, Hogeschool van Arnhem en Nijmegen, Arnhem, The Netherlands. November 2003

ACADEMIC EXPERIENCE:

University of Shkoder, Part time Lecurer: 2010-2012

Fatih University, Full time Lecurer: 2012-2016

University of Shkoder, Full time Lecurer: 2016-2021

AWARDS:

3rd price at Balcanic Mathematical Olimpiads, April 1996.

Partecipating 37th IMO (International Mathematics Olimpiad), June 1996.

FOREIGN LANGUAGES:

English: C1

French: Good

Turkish: Very good

German: Basic

Italian: Very good

PUBLICATIONS (Journals)

M Migena Ceyhan, Zeynep Orhan, Dimitrios Karras, Senol Dane. Sentiment Analysis of Hospital Service Satisfaction, J Res Med Dent Sci, 2020, 8(5): 6-12. eISSN No.2347-2367: pISSN No.2347-2545. Available Online at: www.jrmds.in <https://www.jrmds.in/articles/sentiment-analysis-of-hospital-service-satisfaction.pdf> reachable at https://www.jrmds.in/inpress.html

Ceyhan Migena, Orhan Zeynep, Karras Dimitrios. Automatic Classification of Film Reviews on Social Media through Opinion Mining. Journal of Social Research and Behavioral Sciences (Sosyal Araştırmalar ve Davranış Bilimleri Dergisi), SADAB, 2020, Volume 6, Issue 10, p. 130-154. www.sadab.org ISSN:2149-178X. Accessed from http://www.sadab.org/FileUpload/bs701867/File/ceyhan_movie_reviews.pdf

Ceyhan, M., Orhan, Z., & Karras, D. (2020, August). An Approach for Movie Review Classification in Turkish. European Journal of Engineering and Formal Sciences, 4(2), 1-9. Retrieved 2020, from http://journals.euser.org/files/articles/ejef_v4_i2_20/Ceyhan.pdf

Ceyhan, M., Orhan, Z., & Karras, D. (2020). Sosyal Medyadaki Film Yorumlarının Fikir Madenliği ile Otomatik Sınıflaması. International Journal of Social Sciences - Uluslararas{\i} Sosyal Bilimler Dergisi, 4(20), 1-31. Retrieved 2020, from http://www.sobider.net/FileUpload/ep842424/File/film_makale_(1).pdf

Index Copernicus ICV=55.09

Orhan, Z., Giriş, S. F., Ceyhan, M., & Mercan, M. (2014). CUSTOMER SATISFACTION MEASUREMENT TOOL BY ANALYSING TURKISH PRODUCT REVIEWS. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 7(1), 12-18. Retrieved 2016, from https://dergipark.org.tr/en/pub/tbbmd/issue/22247/238816

ORAL PRESENTATIONS (Conferences)

Ceyhan, M., Orhan, Z., & Domnori, E. (2017, March). Health service quality measurement from patient reviews in Turkish by opinion mining. *IFMBE*

Proceedings of International Conference on Medical and Biological Engineering in Bosnia and Herzegovina. 62, pp. 649-653. Singapore: Springer. doi:10.1007/978-981-10-4166-2

Ceyhan, M., Orhan, Z., & Domnori, E. (March, 2017). E-Medical Test Recommendation System Based on the Analysis of Patients' Symptoms and Anamnesis. *IFMBE Proceedings of International Conference on Medical and Biological Engineering in Bosnia and Herzegovina, CMBEBIH. 62*, pp. 654-659. Singapore: Springer. doi:10.1007/978-981-10-4166-2

Ceyhan, M., Orhan, Z., & Karras, D. (2018). Sentiment Polarity Classification of Turkish Product Reviews for Measuring and Summarizing User Satisfaction. In A. DL (Ed.), *Proceedings of the Workshop on Opinion Mining, Summarization and Diversification, RevOpiD '18. 1*, pp. 1-10. Baltimore, Maryland, New York, NY, USA: Association for Computing Machinery. doi:10.1145/3301020.3303752

Ceyhan, M., Orhan, Z., & Karras, D. (2020, 30-31 June). Towards machine evaluation of human produced comments in Turkish. *ICMS XXIII Proceedings on 23rd International Conference on Multidisciplinary Studies: asResilience for Survival"*, *II*, pp. 236-241. Cambridge, UK. Retrieved from http://books.euser.org/files/proceedings/icms23_proceedings_v2_ISBN_9781649991584.pdf?v=5