REPUBLIC OF TURKEY
FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL
AND APPLIED SCIENCE

AGE AND GENDER IDENTIFICATION BY SMS TEXT MESSAGES

AHMAD JAMAL KHDR

Master Thesis
Software Engineering Department
Supervisor: Assoc. Prof. Cihan VAROL

DECEMBER 2018

REPUBLIC OF TURKEY

FIRAT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

AGE AND GENDER IDENTIFICATION BY SMS TEXT MESSAGES

MASTER THESIS

AHMAD JAMAL KHDR

162137113

Submission date to Institute of Applied Science: 05 November 2018

Thesis Presentation Date: 07 December 2018

Thesis Supervisor : Assoc. Prof. Cihan VAROL (SHSU)

Jury Member : Prof. Asaf VAROL (FU)

Jury Member : Asst. Prof. Mehmet KAYA (ADYU)

NOVEMBER 2018

DECLARATION

I, Ahmad Jamal Khdr, declare that the master's by research exegesis entitled *Age and Gender Identification by SMS Text Messages* is no more than 16,560 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references, and footnotes. This exegesis contains no material that has been submitted previously for the award of any other academic degree or diploma, in whole or in part, except for the passages and single words that are quoted. Except where otherwise indicated, this exegesis is my own work. It is being submitted for a master's degree in software engineering.

DEDICATION

I dedicate my thesis to my family and many teachers. A special feeling of gratitude to my loving parents, Jamal Ormizyar and Awaz Mohammed, whose words of encouragement and push for tenacity still ring in my ears. My sister, Chrakhan, has never left my side and is very special to me.

I also dedicate this thesis to my wife who has supported and encouraged me throughout the process. I will always appreciate all the teachers, especially Dr. Cihan, for helping me develop my technology skills and Prof. Asaf for helping me and giving me a place among the software engineering family members at the technology faculty.

I dedicate this work and give special thanks to my lovely wonderful twin daughters, Niva and Lava, for being there for me throughout the entire master's program. Both of you have been my best inspiration and the joy of my life.

ACKNOWLEDGMENTS

I would like to thank the committee members who were more than generous with their expertise and precious time. A special thanks to Dr. Cihan VAROL, the committee chairman, for his countless hours of reflection, reading, encouragement, and most of all patience throughout the entire process. I would like to acknowledge and thank my school division for allowing me to conduct my research and providing any assistance requested. Special thanks to Dr. Tao Chen and her supervisor Associate Professor Dr. Min-Yen Kan for creating the SMS dataset and published it at National University of Singapore official website for research use. Finally, I would like to thank the teachers, mentors, and administrators in our school division that assisted me with this project. Their excitement and willingness to provide feedback made the completion of this research an enjoyable experience.

TABLE OF CONTENTS

Page No

ABSTRACT

Age and gender identification from text documents became a popular subject for researchers within the text classification field. Over the last decades, the number of text-based social network applications such as Facebook messenger, Twitter, and short message services, has increased at a rapid space. That is why texting has become the most popular method of communication that has users' attention all around the globe.

This research aims to predict the age for 8 different age ranges and to identify the gender of a text sender from their short text messages. The reason behind this research is that some people fake their age and gender in text-based messaging applications. Linguistic psychology shows how certain words and writing styles of different people can be used to identify their age and gender.

In recent decades, researchers used different sets of features for age and gender identification of an author. However, feature set identifications will always be a barrier for researchers. In this study, 25 different experiments were applied for the Naïve Bayes, Support Vector Machine (SVM), and J48 algorithms based on changing the parameter settings to prepare a feature for the identification of age according to different age classes and gender. The text that an author used was preprocessed in different stages.

To design a module for SVM, Naïve Bayes, and J48, Weka (data mining software) was used. The reason behind using these three algorithms is that SVM is the most accurate and powerful algorithm used in text classification, Naïve Bayes is the fastest at building a module, and J48 has the ability to choose the most biased features, can classify data without complex calculations, and it has ability to handle incomplete or noisy data. However, it still took a long time to create a module. The Short Message Service (SMS) text messages used for the training and testing stages in this study can be found on kaggle.com under the name "The National University of Singapore SMS Corpus".

The highest accuracy for age prediction was in experiment number six, which yielded 70.9823% by SVM, Later, after the whole dataset been used as a training set and one of the age classes been used as a testing set at a time. The highest result recorded for the age between 16 to 20 that included 20649 instances by using the same parameters, it was 91.3361% which recorded by SVM algorithm, while the highest record for gender

identification was in experiment number three, also gained by SVM, which it was 79.5869% according to application of different parameter settings.

Keywords: Age prediction, Classifier, Gender identification, J48 Decision Tree, Machine Learning, Naïve Bayes, Psychology Linguistic, Support Vector Machine.

ÖZET

SMS Metin Mesajları İle Yaş ve Cinsiyet Belirleme

Yazı sınıflandırılıması alanında yaş ve cinsiyet ayrımı yapılması araştırmacılar için popüler bir konu olmuştur. Yakın zamanda, Facebook, Twitter, ve kısa mesajlaşma servisleri gibi yazı tabanlı sosyal iletişim ağlarının kullanımı oldukça artmıştır. Dolayısıyla kısa mesajlaşma Dünya çapında insanların ilgisini çeken en popüler iletişim mekanizması olmuştur.

Bu çalışmada, kısa mesajlar üzerinden, mesajı yazan kişinin yaşı ve cinsiyetini tahmin etmeye çalıştık. Bu konunun seçilmesindeki temel neden, bazı insanların bilinçli olarak yazılı mesajlar da yaşını veya cinsiyetini yanlış göstermeye çalışmasıdır. Dil bilimi bazı seçilen kelimelerin ve yazım şeklinin kişinin yaşını ve cinsiyetini tahmin etmede kullanılabileceğini göstermiştir.

Eski çalışmalarda yaş ve cinsiyet ayrımı yapılmasında farklı nitelik verileri kullanılmıştır. Nitelik verilerini tespit etmek her zaman araştırmacılar için bir problem olacaktır. Bu çalışmada 25 farklı deney Naive Bayes, Destek Vektör Makinesi (DVM), ve J48 algoritmaları için uygulanarak nitelik verileri yaratılmıştır.

Seçilen üç algoritmadan, DVM yazı sınıflandırılması alanında en doğru ve başarılı oldugu için, Naive Bayes en hızlı bir şekilde modeli kurduğu için, J48 ise veriyi kompleks hesaplamalara katmadan, yarım veya gürültülü verinin üstesinden gelebildiği için seçildi. Lakin yine de verinin büyük olmasından dolayı işlem uzun zaman aldı. Bu çalışmanın temelini oluşturan veri kaggle.com sayfasından “The National University of Singapore SMS Corpus” isimli linkten indirilebilir.

Yaşı tahmin etme de, en yüksek başarı altıncı deneyde %70.9823 ile DVM tarafından elde edilirken, doğru cinsiyeti tahmini en yüksek başarı oranı 16-20 yaş arası 20,649 örnekli grupta, yine DVM ile %91.3361 ile elde edildi.

Anahtar Kelimeler: Yaş tahmini, Sınıflandırıcı, Cinsiyet tanımlaması, J48 Karar Ağacı, Makine Öğrenmesi, Naïve Bayes, Psikoloji Dili, Destek Vektör Makinesi.

LIST OF FIGURES

Page No

LIST OF TABLES

Page No

ABBREVIATIONS

ANN : Artificial Neural Network

ASCII : American Standard Code for Information Interchange

BNC : British National Corpus

BoW : Bag of Words

CNN : Convolutional Neural Network

EER : Equal Error Rate

FAR : False Acceptance Rate

FNMR : False Non-Match Rate

FNR : False Negative Rate

FPR : False Positive Rate

FRR : False Rejection Rate

IDFT : Inverse Document Frequency within Text

LDA : Linear Discriminant Analysis

LIWC : Linguistic Inquiry and Word Count

LSTM : Long Short Term Memory neuronet

NUS : National University of Singapore

NV : Naïve Bayes

PCA : Principle Command Analysis

PNN : Probabilistic Neural Network

RBF : Radial Basis Function

ReLU : Rectified linear unit

SMO : Sequential minimal optimization

SMS : Short Message Services

SVM : Support vector machine

TFT : Term Frequency within Text

1. INTRODUCTION

Nowadays, the most common media type in our daily life is still text. We use text within all applications of social networking like Twitter and Facebook, also email, blogs, and chat rooms as web applications and built-in SMS texting applications in mobile devices. These also mostly text-based.

In this research, we deal with two questions for text-based digital forensics.

1. Are we able to anticipate the gender of an author who typed a short text document?
2. Are we able to predict the age range or exact age of the same author?

The motivation behind those two questions is that people often fake both their age and gender the text conversations. In online communities, anonymity is very important [1]. People do not have to be honest with their identities, address, gender, age, or name.

In many criminal situations, evidence of the author's real identity is often hidden by criminals to avoid being caught. Therefore, it is important to design and create an effective technique for tracing identity in digital forensics [2]. Gender and age identification will be an area of interest in our research.

To give clarification about the identification problem for gender and age lets suppose the latest case, Myspace mom, released in (www.foxnews.com). Lori Drew (13 years old, female) and some others laid claim to be a teenage boy (called Josh, 13 years old) on Myspace and made a friendship with Megan Meier.

In Myspace Megan and Josh were exchanging text messages for more than a thirty days when Josh rudely terminated their friendship, because he told Megan that she was harsh. Megan later committed suicide. Obviously, this type of case comprehensibly is mentioned to the techniques that based on texting for identifying gender and age, and thus, it could help how protecting children that using their mobile phones and on the internet [3].

According to a psychological research, one can measure physical/mental health and emotion of a person by utilizing the words that he or she uses [4] and proposed that each author has a unique style attitude, which referred to the writer's profile.

As a forensic identification tool, researchers are using this textual evidence for authorship identification. However, we need to be careful, because sometimes an author is able to change his/her real identity and with an opposite gender speak like a teenager or an adult.

2. LITERATURE REVIEW

Cheng and colleagues, examined author gender prediction for multi-genre, content-free text, and short length messages. They used machine-learning algorithms (J48, SVM, and Bayesian logistic regression) [2]. However, they noted a problem with their paper as it is not similar to other types of author gender prediction on three points:

- Higher level of predicting gender is abstraction (and it is not similar to authorship attribution because it is missing the user's candidate set).

- Comparing the length of SMS text messages from the Emails. Classic text documents are usually smaller with special linguistics elements such as emoticons.

- The structures of internet text documents diversify from author to author according to the situation such as private message, internet chat, etc.

In this paper, Cheng et al. treat the problem as a binary classification problem because we have only two results, whether the author is male or female. According to their research, the gender prediction procedure can be explained in four steps:

- Collecting appropriate text messages to be used later as the dataset.

- Recognizing those features that could be helpful for gender prediction.

- Automatic extraction of the values of the feature out of every message.

- Creating a categorization model to predict the gender of an author through text messages.

They collected their datasets from the Reuters newsgroup and emails, and for the implementation, they extracted features by using Python and then the classifiers were applied in MATLAB. First, Bayesian-based logistic regression was applied, and then each AdaBoost decision tree and SVM classifiers one by one was used. Ultimately, the results showed that SVM performance is better than the other two methods for both datasets, and the performance of the Bayesian logistic regression had less effect on the training set size examples related with the quick change in the AdaBoost (200 iterations) decision tree.

The SVM classifier generated the best classification results, with an accuracy of 82.23% and 76.75% according to both datasets. The researchers indicated that the genders of the Reuters news group dataset are hard to be predicted rather than the emails. For testing the suggested set of features, they applied SVM to the sub-dataset that had text messages that contain 100 words in a time by using one feature set.

When they used dimension reduction for features through processing only the significant features compared to 3.77 seconds before feature reduction, they got a faster extraction that took only 1.35 seconds to get a result. However, after applying SVM classification on the sub-dataset, they used only 157 features, and they yielded 82.1% accuracy, a drop of 3.03% compared to the accuracy before feature reduction. Thus, they could set the importance level of testing to be lower or higher, to choose and decide between times, cost, and accuracy.

Mansur et al. classified gender in a few steps where they used a collection of suitable text messages as their dataset. First, they identified features that are most important for gender classification. Later, feature values were extracted from messages and to create a model of classification for predicting the author's gender for a text message. They used the Enron email dataset to train the classifiers. This dataset contains 517,431 emails; it took three and a half years to collect the data from 150 users [5].

For the classification, they used the SVM algorithm then applied it to software named LibSVM. For model selection, they performed a grid search on the C and gamma parameters, using a radial basis function (RBF) kernel throughout. In addition, to getting their result, first, the researchers trained the SVM model on the email dataset, using a validation set to pick the model parameters, and then the researchers evaluated the accuracy of the model on a holdout test set of emails from the Enron dataset, as well as the text of 100 different reviews of Google Play applications. On the test data from the email dataset, the accuracy was 93.67% (13081/13965), very close to the 93.44% accuracy on the validation set.

According to Brocardo et al. both the n-gram analysis and supervised learning technique combined to verify authorship in short texts. Enron emails , which holds 87 users, were used as the dataset and led to very promising results consisting of an equal error rate (EER) of 14.35% for message blocks of 500 characters [6]. They conducted the authorship analysis from three various point of views, including authorship identification, authorship verification, and authorship profiling or characterization. Recognition of authorship determines the most likely user of a target document in a list of known persons. Authorship verification checks that a predetermined document whether has been written by a specific person or not. Authorship profiling or characterization could determines the user's character (such as gender, age, and race) of an anonymous document.

The researchers used an n-gram model for language processing and selecting features. They applied the n-gram in two steps: first of all, the user profile is achieved by takeout n-

grams from testing documents, then, a user-specific threshold is computed to the verification phase.

However, in the result, two errors happened corresponding to the false rejection rate (FRR), which referred to false non-match rate (FNMR) or false positive rate (FPR), and the other error corresponded to the false acceptance rate (FAR), also referred to as the false match rate (FMR) or false negative rate (FNR).

Miller et al. were trying to classify gender from Twitter users by utilizing Perceptron and Naïve Bayes through selecting 1 to 5-gram features from tweeted text. They used stream applications to classify gender to deal with the speed and volume of tweet traffic. Since, text like tweets cannot be evaluated easily by using a traditional dictionary, therefore n-gram features were applied to characterize streaming tweets [7]. In their paper, the large number of 1 to 5-grams needed a subset to determine gender. Thus, the researchers chose informative n-gram features by using numerous selection algorithms. For a best instance, the Perceptron and Naïve Bayes stream algorithms made accuracy, balanced accuracy, and F-measure above 99%.

For the data set and feature extraction, the researchers downloaded 36,238 unlabeled tweets by using a tweeter-streaming API. Then the tweets manually labeled to male and female, after which the researchers kept a tweet from each user then they deleted the non-English language texts and unclear genders. After this, the number of dataset users was 3000 while 60% of them were female. However, the data separated into two equal sub-datasets for training and testing to extract and select useful features of the users. Each count of the n-grams was used to find a feature that is why the researchers faced a problem because the higher orders of n-grams will find a new feature in text characters, whereas to show every tweets a 1-grams to 5-grams were handled.

The researchers faced few disadvantages in their research. By increasing the value of n, the number of features exponentially enhanced too. In another word, when 95 1-grams were achieved, then 95 = 9025 2-grams would be required. If the researchers used every possible 1 through 5-gram for each 95 characters, then 7,820,126,495 features should be stored for each instance.

Miller et al. used Weka for data mining by applying six features and selecting algorithms for training sets. They used some algorithms including chi-square, information gain, information gain ratio, relief, symmetrical uncertainty, and filtered attribute evaluation. In addition, for their result, the researchers utilized two modest stream mining algorithms,

which were Perceptron and Naïve Bayes. The first one, performed with a relatively high precision (97%), and its balanced accuracy was 94%. The result shows that Perceptron worked more better than Naïve Bayes that could score the corresponding rates between 90% and 100% for all metrics.

Deitrick et al. used Balanced Winnow neural network algorithm to to assigns three parameters that used for classification. First, they created a learning rule (model) then, if any mistakes happened in the classification, it will update its rule. For each new sentence, the learner makes an estimate by a predefined neural network function, and after that make a comparison of the predicted value and the real class. If the estimation is true then the learner will up to dated itself and for a new sentence [8].

Also, they used a Balanced Winnow neural network. This algorithm describes three factors that will be used for classification: the α factor is used to promote the rule through multiplication and increasing the value in the rule, β (similarly) is the demotion parameter and constantly decreases the values stored by the model and, θ (threshold) in charge of biasing the prediction. These factors will control which rule will be learning.

Deitrick et al. used a modified Balanced Winnow neural network. This new algorithm modifies the Balanced Winnow to make an improvement in the learning process. It creates a margin M, which means an estimation created by the modified Balanced Winnow is a mistake when the true class multiplied by the score function and the result gives a value equal or less than M (this certifies that modifications will happened when the estimation is completely true).

The popular Enron email corpus was used for this paper. The Federal Energy Regulatory Commission made this dataset during the study of the Houston-based energy company's 2001 bankruptcy.

The researchers used different sets of parsed emails for testing learning algorithms. It has 436 stylometric features, character-based features, syntactic features, word-based features, and function words.

As for their result, Deitrick et al. achieved through the Balanced Winnow neural network roughly 56 percentage accuracy for each stylometric and word-based features.

Montero et al. utilized two sets of annotated corpora in their research, private journals and weblogs. They researchers used the Weka API to apply two types of classifiers, which were support vector machine (SVM) and C4.5 decision tree (J48). These classifiers are well

defined in the text classification field and they are a good representation of both statistical (J48) and function-based (SVM) machine-learning approaches [9].

In addition, for their result, the J48 classification algorithm, BoW feature sets and Emo and BoW feature sets had equal performance, however their correctness was extensively less than that obtained with SVM. However, using the emotion feature sets lonely, the J48 algorithm was sensitive to determine of emotion features.

Zhangand et al. used blog posts from several blog hosting sites and blog search engines as their dataset; the data set consists of 3,226 blog entries, each with a gender label. Out of the 3,226 posts, 1,551 (48.08%) were written by males and 1,672 (51.92%) were written by females. The average length of these blogs is 422 words [10].

Zhangand et al. used IG, Naïve Bayes, SVM, and LDA (linear discriminant analysis) algorithms. However, the researchers faced some problems where they could not apply the generic dimension reduction technique, such as PCA since the feature space is too big. However, this problem was solved by two common methods for feature selection, filtering and the wrapper. In filtering, features are ranked by metric and only the top features are retained for classification. In addition, the wrapper was used if the new features improved the classification accuracy, then the new features were added to the existing set. As for their result, the best prediction accuracy that could be achieved according to the algorithms and classifier was 72.1017%.

Peersman et al. chose a corpus of 1,537,283 Flemish Dutch posts from the Belgian social networking site Netlog. First, they preprocessed the data by extracting the last post of each contact and saved them as a separate document. Later they tokenized the dataset, where general emoticons were interpreted, normalizing each token to lowercase and reducing all four or more repetitive characters to a minimum, for example, “hellooooo” to “hello”. After that, they grouped the data by using the profile data. Finally, they selected features by applying the chi-square ($\chi 2$) [11].

Peersman et al. used the SVM learning package Liblinear and the greatest outcomes they obtained was by balancing the set of data according to gender by a top accuracy of 88.8%.

Sboev et al. used two sets of methods, first, they used a set of various machine-learning algorithms in a wide range utilized for classification of texts, gradient boosting classifier, adaptive boosting classifier (adaboosting), extra trees, PNN (sigma = 0.1), random forest, SVM with linear kernel, and ReLU (1 hidden layer). Second, implicated topologies of ANN

and convolutional neural network (CNN) that applies convolutional filters to successive windows for a certain sequence to obtain global features by max-pooling which composed with Long Short term memory neuronet (LSTM).

As for their result, ReLU was the most efficient classification algorithm that had an accuracy of 74%, and complicated neural network models (CNN+LSTM) show upper levels of accuracy at 86%. However, the weakness of these kind of models is a complexity in features evaluation because these models doing the processes on input data in a non-linear way [12].

It is obvious that most of the studies mentioned here used SVM and scored high results for age and gender predicting by classifying text documents. However, there are differences between earlier studies and our study which are that most of the studies listed here used emails, web blogs, journals, and tweets as datasets, compared to data from SMS text messages, which we used in our research. Text from emails, blogs, journals, tweets have too many characters and complicated, while text messages are short with a maximum number of characters for one message is 140.

3. DATA PREPROCESSING

3.1. Corpus

The main goal behind this thesis and its research is to specify the gender of the writer and predicting their age according to each classes through SMS text messages. For that purpose, a dataset that has many text samples was needed, where it was easy for us to source different writing styles of both genders of different ages. There were different datasets for the researchers; each one of those datasets had different attributes according to gender, age, and the text message content.

For my research, I used the National University of Singapore SMS Dataset that holds 67,093 short messages in both Mandarin and English (native speaker and non-native speaker) languages [13]. In addition, the British National Corpus (BNC) dataset [14] is another corpus that researchers can use. This corpus consists of 100 million words of British English. The Oxford University Press created the BNC in the 1980s and the beginning of 1990s. It includes one hundred million words of text and texts from an extensive range of categories such as speeches, fiction, academics, newspapers, and magazines).

The BNC is connected to other English corpora that we created, which suggests a unique insight into differences in the English language.

3.2. National University of Singapore SMS Dataset

One of the most widely used datasets to identify gender and age from text messages is the National University of Singapore SMS Corpus. Messages are short text documents sent from one person to another from their mobile phones. The number of characters changes from one device to another with the latest smartphones providing only 70 characters while the old models provided 140 characters. These text documents are a method of personal communication that are an important communication artifact of our present digital era. The dataset used in my research has SMS messages gathered from users who knew their messages would be used in a research project and be made public. Two languages can be recognized in this dataset, Singapore English and Mandarin Chinese.

This SMS messages corpus was collected for research at the Computer Science Department at the National University of Singapore. This dataset has of 67,093 SMS messages gathered from the corpus on March 9, 2015, from both Chinese (Mandarin) and Singapore English languages. Those messages were mostly collected from university students. They were collected from volunteers who knew that donating their messages would make the messages available for public use later. The data collectors gathered as much metadata about the messages and their senders as possible; therefore, the researchers could get different types of analyses. T. Chen and M.Y. Kan collected those messages.

3.3. Data Preparation

3.3.1. Remove the Unnecessary Columns and Add Quotation Marks

The first preprocessing to be done is cleaning the data. There are many columns in the Excel sheet of the dataset that are not needed because only the content of the text message is being used for age and gender identification. The sender data will be kept because it can be used to group content together. One distinct sender will always have the same age and same gender.

The following columns can be safely removed: receiver, send time, collected time, collected method, (no name yes/no), country, input method, experience, frequency, phone model, collector, smartphone, language, city.

Also, during this step, a formula is used in Excel to make sure that quotation marks in the string are properly escaped (i.e. ” becomes \”), and that quotes surround the text.

This was not the first formula I tried. Excel was inconsistent in terms of exporting a CSV with quotes. Sometimes it would use a triple quote (“““), while other times it would use no quote. To be safe, the following formula was used to surround the content in quotation marks while still in the Excel worksheet.

=CONCATENATE(CHAR(34),SUBSTITUTE(Original!B3,CHAR(34),CONCATENATE(“\”,CHAR(34))),CHAR(34)) **Eq. 1**

This has the effect of **“ + the content with quotes escaped + “**

However, the original formula that was used was simplified:

= SUBSTITUTE(Original!F2,CHAR(34),CONCATENATE(“\”,CHAR(34))) **Eq. 2**

This has the effect of: **The content with quotes escaped**

3.3.2. Remove Records That Are Useless for Classification

Those records that do not have a valid age and do not have a valid gender been removed because there is nothing to train on, or verify test results on.

There are 10,992 records that do not have a valid age. As it turns out, none of the records with an invalid age has a valid gender, so those have also been removed.

3.3.3. Remove Invalid Gender Records

There are 41,241 records left after this is done. These all have text content, a valid age, and a valid gender.

Depending on what the predictive accuracy is for age, the records removing this step might be used later. Since they have a valid age, they might be useful for training or testing, as long as it is only for age and not for gender. Since less than 10% of the records were removed in step 3.3.3, they probably will not be used at all.

3.3.4. Remove the Line Breaks, Tabs, and Duplicate Instances

Some of the text contains line breaks and tabs. These will cause problems when trying to load the file into Weka. The following formula replaces every carriage return (char 13), line break (char 10), and tab (char 9) with spaces. The space is important in case the line break or tab character is separating two words.

=SUBSTITUTE(SUBSTITUTE(SUBSTITUTE('Step1c'!B11,CHAR(13),"
"),CHAR(10)," "), CHAR(9), " ") **Eq. 3**

There are many duplicate SMS messages. When they are removed using an in-place filter in Excel (Select Data → Filter → Advanced), it leaves 38,623 unique SMS messages, with both a valid age and a valid gender.

In the case where duplicate SMS content has not been removed, then this would have had the same effect as oversampling a particular class. It would no longer represent what happens, generally. This would have introduced errors in the model, leading to a poorer predictive accuracy in the future.

3.3.5. Filter the Remaining Erroneous or Missing Data

- In the original content, there are 23 instances that contain "= #N/A", which indicates that the original source either contained erroneous or missing data. Since the content is what will be used for classification, these instances were removed using another in-place filter in Excel. This left a total of 38,600 SMS messages.
- There are also 12 instances where VCARDs are sent. These contain no useful data yet and caused trouble for loading the ARFF file into Weka. This left 38588 SMS messages.
- A new column titled "CSV" was created that will be copied and pasted into a text file to build the ARFF. This is simply a combination of the other fields with a "clean" on the outside to remove non-printable characters using the following formula: =CLEAN(CONCATENATE(A2,",",B2,",",C2,",",D2)) **Eq. 4**

3.3.6. Create the ARFF Header

The ARFF needs to have three attributes for the data, defined in the following way:

@RELATION sms
@ATTRIBUTE _content_ STRING
@ATTRIBUTE _age_ (16→20, 21→25, 26→30, 31→35, 36→40, 41→45, 46→50, 51→60)
@ATTRIBUTE _gender_ (male, female)
@ATTRIBUTE _sender_ (male, female)
@DATA

3.3.7. Fix Remaining Parsing Errors

- There were a few other parsing errors found during the Weka load. These included several instances of the backslash (\) not being escaped. A search and replace for "\" to "|\\" fixed this.
- There were some cases where the gender was specified as "Female" instead of "female", and again, a quick search and replace fixed these.

3.4. Statistical information (verification)

When the ARFF loads, Weka reports the following information about the data:

Total instances: 38,588

Content: 35,197 distinct values, 32,616 unique values, 0 missing values

Age: 8 distinct values, 0 unique values, 0 missing values

Gender: 2 distinct values, 0 unique values, 0 missing values

Sender: 146 distinct values, 1 unique value, 0 missing values

According to these statistics, it appears that the data was loaded and processed correctly.

3.4.1. Dataset Imbalance

It is important to consider that there are different numbers of instances between classes of gender and classes of age. The breakdown for gender is:

Table 3.1. Gender Breakdown

Gender	Instances	Percentage ratio
Male	26242	68.01
Female	12346	31.99

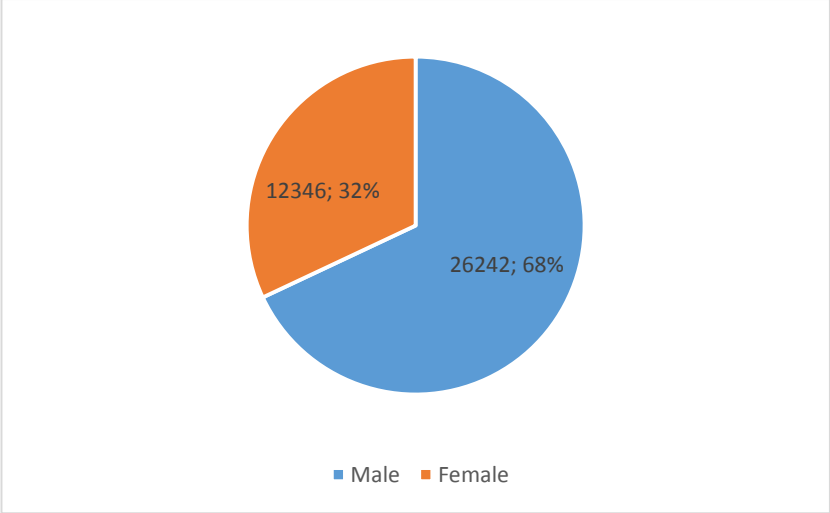

Figure 3.1. Gender Breakdown

While the breakdown for age is:

Table 3.2. Age Breakdown

Age range	Instances	Instances percentage
16–20	20649	53.51
21–25	14289	37.03
26–30	2665	6.91
31–35	118	0.31
36–40	595	1.55
41–45	240	0.62
46–50	10	0.03
51–60	20	0.05

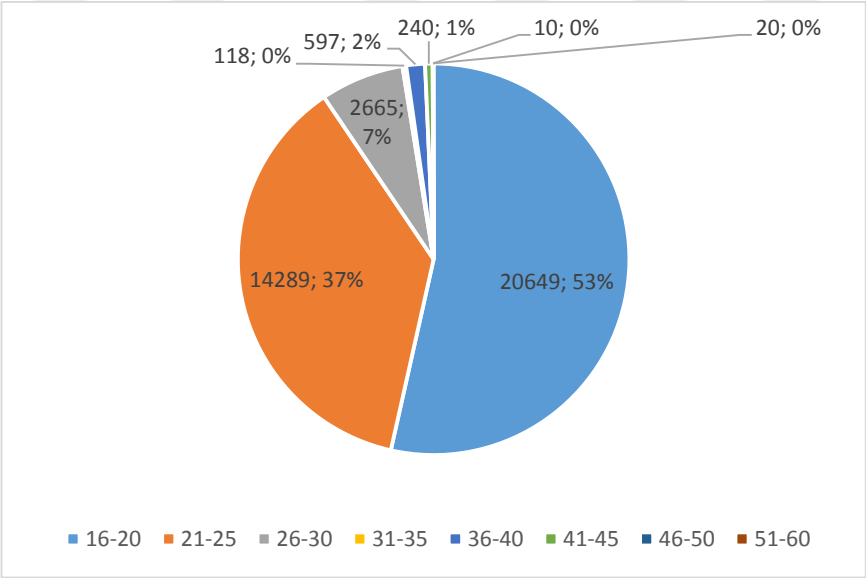

Figure 3.2. Age Breakdown

It is important to remember that age is highly imbalanced outside of the 16–20 and 21–25 age groups. This is because there simply is not enough data in the other categories. This can sometimes be accounted for using techniques such as undersampling and oversampling, but because of the goals here, it will not be needed.

Significantly, it does mean that there are minimum targets to achieve in order to judge the worth of the classification. There are two measures: first, it should be better than random and second, it should be better than assuming the most popular class.

- **Better than random**

Since there are only two genders, a random guess will achieve 50% accuracy on average. In the case of age, there are eight classes, meaning that a random guess will yield 12.5% average accuracy.

- **Better than guessing the most popular class**

For gender, the most popular class is male with approximately 68%. For age, the most popular class is 16–20, with 53.5% of the instances.

3.5. Filter Implementation

3.5.1. Convert the String into a Word Frequency Vector

Text classification requires that the text is converted into something that can add meaning to the search. Weka uses a filter called "StringToWordVector" to convert a string into many attributes, each containing popular words and the frequency of these words.

The filter has many options and variations will be tried to see what works best for this data.

In the following steps, we can see how to apply the StringToWordVector filter:

Choose → Unsupervised → Attribute → StringToWordVector

By clicking on the "StringToWordVector" label, it will bring up a screen of options. The choices of options are described in the tables later.

The main thing to remember is that the (attributeindices) value must be 1 so that only content is converted into a word vector.

After selecting that option and applying it, the default parameters will expand the size of the dataset to 1,007 attributes.

3.5.2. The Remove Filter

When performing classification, it is important that Weka does not make use of the sender. If the attribute is used, then it could relate the sender's ID to an age or gender. This might be useful for building a sender profile, but it will do nothing for using the content to make predictions about age or gender.

To remove the attribute, it requires using another filter:

Choose → Unsupervised → Attribute → Remove

Once again, the options have to be edited by clicking on the label. The only thing to change is the attribute-indices, which has to match the sender that in our case is line number 3 in sequence. After clicking apply, the sender will be removed from the list.

3.6. Classification

There are several classification experiments to run. Some of them are going to use the full dataset to train and testing using cross-validation. For those experiments, there is going to be one large ARFF file with all of the data in it.

The main objective of this comparison is to clarify which classifier of Naïve Bayes, support vector machine, and J48 decision trees work the best for the determination of age and gender from the analysis of SMS text messages. As such, the majority of experiments involved running all three of the classifiers multiple times while varying different parameter settings.

3.6.1. Training and Testing the Data

a. Full Dataset – 10 - folds cross-validation

It is possible to test the Naïve Bayes classifier using this file, and several experiments are conducted with it. However, the SVM and J48 decision trees are bogged down by the number of instances and it is not practical to use the full file for classification. As such, this research will not contain full 10-fold cross-validation experiments for anything other than Naïve Bayes.

As will be seen in the results, the training split seems to be very good. It appears that using the 25% split is almost as effective as using 75% (as evidenced by the SMO performance using 10-fold cross-validation versus the 25% training split).

b. Dataset splitting

A few experiments were accomplished using the 10-fold cross-validation and it seemed that the 25% and %75 split was sufficient to do the bulk of the experiments on. Different combinations of parameters were tried, as can be seen in the results chart.

Table 3.3. All experiments and results

Predicting	Method	IDFT	TFT	Word Counts	Lowercase	Stemmer	Stopwords	Tokenizer options	Average Accuracy	NaiveBayes Accuracy	NaiveBayes Cmp Rnd	NaiveBayes Cmp Maj	SVM Accuracy	SVM Cmp Rnd	SVM Cmp Maj	J48 Trees Accuracy	J48 Trees Cmp Rnd	J48 Trees Cmp Maj
Age	10 fold CV	FALSE	FALSE	FALSE	FALSE	Null	Null	WordTokenizer	55.8438%	55.8438%	4.47	1.04						
	75% training	FALSE	FALSE	FALSE	FALSE	Null	Null	WordTokenizer	63.3991%	55.6961%	4.46	1.04	69.5968%	5.57	1.30	64.9045%	5.19	1.21
	75% training	FALSE	FALSE	FALSE	TRUE	Null	Null	WordTokenizer	65.6681%	56.6083%	4.53	1.06	70.7888%	5.66	1.32	69.6071%	5.57	1.30
	25% training	FALSE	FALSE	FALSE	TRUE	Null	Null	WordTokenizer	63.6513%	55.9863%	4.48	1.05	70.3811%	5.63	1.32	64.5866%	5.17	1.21
	75% training	TRUE	TRUE	TRUE	FALSE	Null	Null	WordTokenizer	60.6982%	47.5381%	3.80	0.89	69.9008%	5.59	1.31	64.6557%	5.17	1.21
	75% training	TRUE	TRUE	TRUE	TRUE	Null	Null	WordTokenizer	61.4894%	48.6265%	3.89	0.91	70.9823%	5.68	1.33	64.8595%	5.19	1.21
	75% training	FALSE	FALSE	FALSE	TRUE	Lovins	Null	WordTokenizer	63.2033%	56.2973%	4.50	1.05	69.5138%	5.56	1.30	63.7988%	5.10	1.19
	75% training	FALSE	FALSE	FALSE	TRUE	Snowball	Null	WordTokenizer	63.8587%	56.6083%	4.53	1.06	70.3811%	5.63	1.32	64.5866%	5.17	1.21
	75% training	FALSE	FALSE	FALSE	TRUE	Null	Rainbow	WordTokenizer	63.7527%	59.3138%	4.75	1.11	68.6880%	5.50	1.28	63.2563%	5.06	1.18
	75% training	FALSE	FALSE	FALSE	TRUE	Null	Null	N-Gram	62.9557%	54.7113%	4.38	1.02	69.2754%	5.54	1.29	64.8803%	5.19	1.21
	75% training	FALSE	FALSE	FALSE	TRUE	Lovins	Rainbow	N-Gram	62.1241%	55.0638%	4.41	1.03	67.8484%	5.43	1.27	63.4601%	5.08	1.19
Gender	10 fold CV	FALSE	FALSE	FALSE	FALSE	Null	Null	WordTokenizer	64.8051%	64.8051%	5.18	1.21						
	75% training	FALSE	FALSE	FALSE	FALSE	Null	Null	WordTokenizer	73.0026%	64.1132%	5.13	1.20	78.8363%	6.31	1.47	76.0582%	6.08	1.42
	10 fold CV	FALSE	FALSE	FALSE	TRUE	Null	Null	WordTokenizer	71.6855%	63.7841%	5.10	1.19	79.5869%	6.37	1.49			
	75% training	FALSE	FALSE	FALSE	TRUE	Null	Null	WordTokenizer	72.4589%	62.8797%	5.03	1.18	78.0726%	6.25	1.46	76.4244%	6.11	1.43
	10 fold CV	TRUE	TRUE	TRUE	FALSE	Null	Null	WordTokenizer	65.8365%	65.8365%	5.27	1.23						
	75% training	TRUE	TRUE	TRUE	FALSE	Null	Null	WordTokenizer	73.4978%	65.8443%	5.27	1.23	78.6946%	6.30	1.47	75.9545%	6.08	1.42
	10 fold CV	TRUE	TRUE	TRUE	TRUE	Null	Null	WordTokenizer	66.0698%	66.0698%	5.29	1.23						
	75% training	TRUE	TRUE	TRUE	TRUE	Null	Null	WordTokenizer	74.6795%	66.5906%	5.33	1.24	78.3456%	6.27	1.46	79.1021%	6.33	1.48
	25% training	TRUE	TRUE	TRUE	TRUE	Null	Null	WordTokenizer	73.5634%	66.2555%	5.30	1.24	78.0484%	6.24	1.46	76.3864%	6.11	1.43
	75% training	TRUE	TRUE	TRUE	TRUE	Lovins	Null	WordTokenizer	73.3112%	66.2382%	5.30	1.24	77.9586%	6.24	1.46	75.7368%	6.06	1.42
	75% training	TRUE	TRUE	TRUE	TRUE	Snowball	Null	WordTokenizer	73.6750%	66.5903%	5.33	1.24	78.0484%	6.24	1.46	76.3864%	6.11	1.43
	75% training	TRUE	TRUE	TRUE	TRUE	Null	Rainbow	WordTokenizer	74.5597%	69.3791%	5.55	1.30	77.9275%	6.23	1.46	76.3726%	6.11	1.43
	75% training	TRUE	TRUE	TRUE	TRUE	Null	Null	N-Gram	73.1949%	66.3211%	5.31	1.24	77.6787%	6.21	1.45	75.5848%	6.05	1.41
	75% training	TRUE	TRUE	TRUE	TRUE	Lovins	Rainbow	N-Gram	72.9323%	66.0309%	5.28	1.23	77.1293%	6.17	1.44	75.6366%	6.05	1.41

The first set of experiments performed on gender involved comparing the parameters:

c. TFT + IDFT + Output Word Counts

These three parameters work together to change the weight of words that appear most frequently. Depending on the settings, rare terms are considered more informative than frequent terms, and the frequency may be high in one string but low in the overall set and alternatively, high in the overall set but low or absent for a particular instance. The value of varying these parameters depends on the nature of the text, which is why it is worth experimenting with them.

d. Lowercase

Convert all of the tokens to lower case before building the vector. This will have an impact depending on factors such as how commonly occurring and/or how important proper names are.

Looking at the results shows that the best settings for these are TFT = True; IDFT = True; Output word counts = True; Lowercase = True.

After recognizing the best of these parameters, the next set of experiments were run using these fixed as stated. The parameters that varied are stemmer, stop words and the tokenizer.

e. Experimenting with Stemmers and Stop word removal

When choosing which previously found best settings to test the new parameters with, I looked at the results. It is necessary to choose subsets because it would simply take too long to test everything. For example, some of the experiments building J48 models took several hours (one of the runs at this stage took more than 10 hours to complete).

- **Age**

For predicting age, the SVM had the best accuracy at 70.9823% by using the parameter's value as the following: (IDFT: true, TFT: true, output word counts: true, and lowercase letters: true). However, I did not choose these parameter settings because the average accuracy between the three classifiers is only 61.4894%.

The reason for the drop is that Naïve Bayes did very poorly for these settings on this dataset.

Compare this to another set of parameters (IDFT: false, TFT: false, output word counts: false, and lowercase letters: true). The SVM was still the best at 70.7888%, while the average between classifiers is 65.6681%. Consequently, we chose these parameter settings to move on to the next set of experiments, which it was finding the age range according to eight different classes, each class contains different number of instances as it shown earlier in the table 3.2.

In addition, SVM recorded the highest result for the age class between 16 years old to 20, which it was 91.3361%. However, we can notice huge differences between the classes, because each class recorded different result according to the instance number.

- **Gender**

For predicting gender, the SVM had the best accuracy at 79.5869% using and the accuracy average is 71.6855% where the value of the parameters are (IDFT: false, TFT: false, output word counts: false, and lowercase letters: true). Better accuracy average recorded at experiment number 8 where the value of parameters were as the following: (IDFT: true, TFT: true, output word counts: true, and lowercase letters: true).

However, for age classification, the best average accuracy was for the parameters (IDFT: false, TFT: false, output word counts: false, and lowercase letters: true). As such, this is what the set of experiments for gender will be using.

f. Comparison of N-Gram against Word Tokenization

One of the goals of this research is to determine whether applying n-gram modeling improves the predictive accuracy of the text in the determination of age and/or gender. The method that I have chosen is to try n-gram modeling (using at longest, trigrams) with the best of the parameter settings used to this point.

Table 3.4. Comparing word tokenizer and n-gram modeling

	Average	Best
Age Word Tokenizer	61.4%	71.0%
Age N-Gram	62.7%	69.3%
Gender Word Tokenizer	71.2%	79.6%
Gender N-Gram	73.0%	77.7%

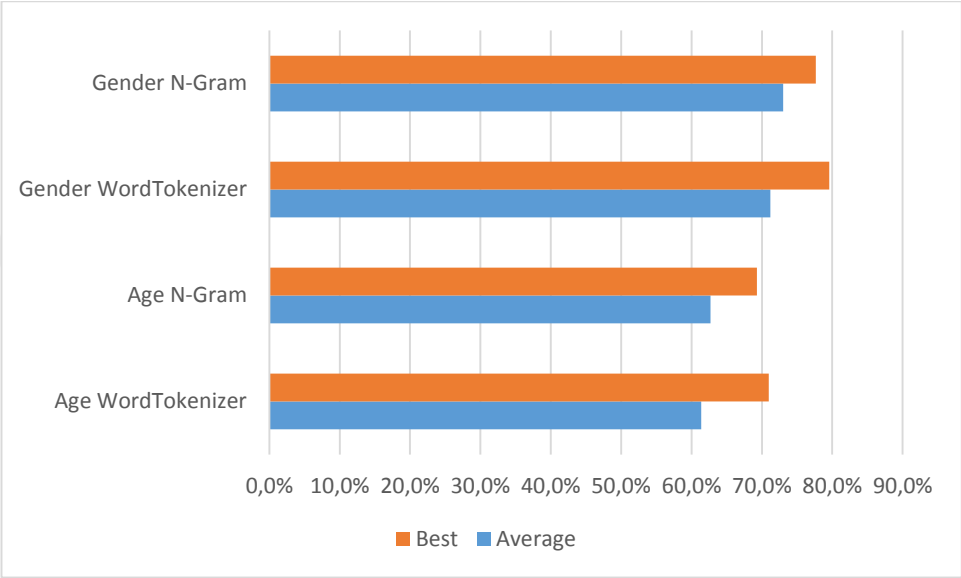

Figure 3.3. N-gram Modeling and Word Tokenizer Comparison

4. FEATURE SET DESCRIPTION

In linguistics, which features are good to separate male writers from their female counterparts? Men and women of different ages have different styles for expressing and explaining the same subject, and this made many researchers interested in identifying gender and predicting age. In the past few years, important changes have been noted in the way men and women of different ages are using different styles of speaking. Let us assume that we have a text document and it contains words related to fashion such as “design“, “tailor“, “cut“, “style“, and “makeup“. It is obvious that the person who wrote a paragraph that contains these words will be mostly defined as female rather than male and the age will be defined after identifying which kind of style or haircut or designer is mentioned in that paragraph. In other terms, if we have another text document that contains words related to sport and news, the probability of gender prediction will be higher to show that it is male [15] and for age prediction it will also depends more details about the type of sports and news they were talking about will define the age.

While analyzing age standards, researchers have discovered that youths mostly mention friends and moods in their writing, where most people in their twenties are more interested in writing about their college life. Those in their thirties are writing about their work, holidays, languages, and marriage.

Different genders of different ages use several of the same words and phrases in their writing, but each expresses themselves in a different way. For example, when a male mentions food, they are pointing to food as their passion, which restaurant serves the best meal, but when females are using the same word, their intention is to discuss diet, health, cooking, and going to a fancy restaurant for dinner. In addition, there is another example of the use of the phrase, “I am fine!”. Males use it to show that they really feel fine, but females are mostly expressing that they are not fine and something is bothering them.

Another example is the use of the word “aging”, which males use for wisdom and more life experience, but females use this word when speaking about marriage, white hairs, or wrinkles.

As with gender, there are differences that are noted with different age groups when they write about different subjects. According to earlier studies, this diversity and difference can be seen from the point of view of human psychology.

In general, the differences and the similarities between genders' speech, intonation, and grammatical features have been studied by scientists. "Females are using delicate and sweeter-sounding words like 'honey' and 'oh my goodness'" said Robin Lakoff, a contributor to feminist linguistics, while tough and stronger words are used by males like "damn"(Braun, 2004: 13). Further, different words are used by both genders in different frequencies.

For example, intensifying adverbs are mostly used by women, like "deeply", "really", "strongly", and multiple question marks or exclamation marks. In general, during conversations, indirect orders are made by females while more directives words are used by males. Comparing females to males, they are using more grammatical language and their speaking is close to standards, while dialectical speech is more common among males. For example, note the differences when asking someone out for a shopping trip, females mostly write, "Does anybody want to go out shopping??? I don't have anything to wear!!!" while a male may write, "I am going out to pick up some clothes".

Another feature, which is the length of the sentence, can be considered to differentiate between both genders. In general, females write longer sentences than males do. In another explanation, females talk more about emotions and personal subjects than men, who speak in a less dramatic tone with fact-based topics [16].

Stylometry [17] is the application for investigation of linguistic style, typically written language, to analyze the differences in literary style between one writer and genre to another. We cannot only use stylometry to recognize writing style, but it can be helpful in identifying both the author's age and gender. In the following section, those features that help us to define writing according to gender and age will be noted by using various stylometrics. All features are listed in the appendix.

4.1. Character-Based Features

In this part, we will explain the text analysis from each character. In general, the text contains 27 stylometric features that will be used in studies of author attribution [17]. First, the total number of characters will be counted, which is all the digits, letters, spaces, and punctuation. The stylometric feature to be analyzed is the total number of letters, comprised of both upper and of lowercase letters. In addition to each uppercase character (A–Z), all

digits (0–9), white spaces, and special characters are counted. There are some examples that are given in Table 5 where shows the character-based features of the texts [16] [17] [18].

Table 4.1. Character-Based Features

Feature	**Description**
Total numbers of characters	Alphabets, digits, and special characters
The entire number of letters	all small and capital letters
Capital characters total number	A–Z
Digital characters (10 characters)	0–9
Each white space character	White space
All special characters (22)	“, #, $, %, &, (,), *, +, _, /, <, =, >, @, \, ^, _, }, {, |, ~

4.2. Word-Based Features

This section will explain word analysis by applying 11 statistical measures [19], comprising the characters per word average number, all existing words, and every word created from three characters as a minimum.

Another technique to find every existing number of words is Hapax legomena [20], which is not repeated within the text. The term “Hapax legomena” means that each word must not match the repetition condition, in other terms, it must be written only one time in a particular text by a specific writer, but this does not mean that the writer must mention that specific word only one time during all of their writing. Because maybe that special word might be used later on in another writing.

Hapax dislegomena [20] is a different evaluation technique, which indicates double events.

As with Hapax legomena, these words can be used by the same authors in other writing, but they may have repeated the same word two times in a particular text.

One more measure that could be helpful for estimating the richness of vocabulary, is called Yule’s K equation [21]. This explains the variety of words in a text document used by the writer. As we can see in equation 5, Yule’s K can be calculated as the following:

$$Yule\ K\ = 10^4(-\frac{1}{N}+\sum_{i=1}^{v} Vi\ \left(\frac{i}{N}\right)^2) \qquad \textbf{Eq. 5}$$

Simpson's D [22] is an equation that refers to a situation where if two words are selected randomly from a text, the chance of choosing the exact same word would be very large. If the outcome is zero, even no diversity is meant at all, but it refers to infinite diversity, therefore, the higher the diversity, the smaller the value of Simpson's D. Simpson's D can be calculated by using equation 6.

$$Simpson\ D = \sum_{i=1}^{v} Vi \frac{i}{N} \frac{i-1}{N-1} \quad \textbf{Eq. 6}$$

Honore's R equation [23] is used for estimating the richness of the text. Honore's R referring that the bigger value of Hapax legomena make the text be richer. The texts' richness can be accomplished by The Honore's R is the word's number in the text that has been written once as the total number of words ratio, as we can see in equation 7.

$$Honore's\ R = \frac{100\ Log_{10} N}{1 \frac{Hapax\ Legomena}{V}}, \quad \textbf{Eq. 7}$$

This part can also calculate entropy by using equation 8 to estimate the data's randomness.

$$Entropy = \sum_{i=1}^{N} Vi \left(-log_{10} \frac{i}{N}\right) \frac{i}{N}. \quad \textbf{Eq. 8}$$

Examples of word-based features are noted in table 6.

Table 4.2. Word-Based Feature

Features	**Description**
The entire number of words	The entire number of all words in the text
Each word's average length	
Richness of the vocabulary	Total number of different words
Long words total number	Words longer than 6 characters
Short words total number	1–3-character words
Hapax legomena	Words that occur only once
Hapax dislegomena	Words that occur only twice
K measure of Yule	Measure of vocabulary richness
D measure of Simpson	Measure of diversity
R measure of Honore	Measure of vocabulary richness
Measurement of entropy	Measure of disorder of dataset

In the last few decades, researchers could found the words that used by people related to their mental and physical health condition [24] [25]. Studies found that the researchers use positive words in many ways like “gorgeous“, “lovely“, “cute“ and a few negative emotions like “pain“, “annoyed“ and “nasty“, also cognitive words like “learn“, “realize“, “design“, and “compose“. Furthermore, in a document those words pronunciation can change from one part to another part [26]. James W. Pennebaker created software called LIWC (Linguistic Inquiry and Word Count). Thousands of words been categorized by Martha E. Francis and Roger J. Booth into 68 groups. While inputting a text into LIWC, we noticed in the output the number of words that exist in each of the 68 categories that a writer used [26].

In this research, according to the word-based feature extraction, those 68 categories were treated as part of age and gender identification. Table 7 explains some example feature sets of LIWC.

Table 4.3. LIWC Feature Sample

Features	**Sample of words in the features**
Certainty	Always, Never
Assent	Ok, Never, Agree
Insight	Think, Know, Consider
Negation	Never, No, Not
Tentative	Perhaps, Assume, Maybe, Guess
Sadness	Cry, Sorrow, Grief, Alone, Sad
Anger	Hate, Kill, Annoyed
Negative emotions	Ugly, Smelly, Hurt, Nasty
Positive emotions	Cute, Nice, Sweet, Lovely
Anxiety	Afraid, Anxious, Nervous, Worried

4.3. Syntactic-based Features

The writing style of any writer will be extracted from the syntactic features of the sentences contained therein. For this purpose, these features contain the number of commas, semicolons, single quotes, periods, exclamation marks, several exclamation marks, question marks, several question marks, and counting ellipses to find out how many times punctuation was used by males and females in their writing.

As an example of informal writing, to express our feelings in a better way we use more than one question mark or exclamation mark. Females in particular use more question or exclamation marks than males [27].

The syntactic features is listed in Table 8.

Table 4.4. Syntactic-Based Features

Features	**Description**
Total number of commas	,
Total number of single quotes	‘
Total number of colons	:
Total number of period counters	.
Total number of question marks	?
Total number of multiple question marks	????
Total number of exclamation marks	!
Total number of multiple exclamation marks	!!!!
Total number of semicolons	;
Total number of ellipsis	...

4.4. Structurally-Based Features

Everyone has a different way of organizing their writing layout and those differences could be related to how they switch from a paragraph to another or maybe the length of the paragraphs. This research is about investigating SMS texts messages exchanged between mobile phones. The interesting feature about these kinds of text is that they are flexible, which means that writers rarely use general writing rules about how the paragraphing or spacing should be. Another interesting feature of SMS text messages according to their content is that they contain useful information; you can tell a short story in less than 140 characters (according to the old mobile phones) and 70 characters according to smartphones.

We counted the total number of sentences and paragraphs, the number of words in each paragraph, the average number of characters, words, and sentences for each paragraph, and the overall number of blank lines in the whole text just to extract the features.

Table 4.5 shows examples of features that were categorized in the structurally-based features category.

Table 4.5. Structural-Based Features

Feature	**Description**
Total number of sentences	In the case each sentence should contain one verb or adverb at least
Total number of paragraphs	In the case of pressing enter
Average number of sentences per paragraph	In the case of writing more than one sentence
Average number of words per paragraph	In the case of pressing the space bar to write a new word
Average number of characters per paragraph	Characters and spaces in the case of pressing enter to create a new paragraph
Average number of words per sentence	In the case where "." or "," is written
Total number of blank lines	In the case of pressing the space bar to create a space between words

4.5. Function Word-Based Features

Words that the author uses in the sentence to create grammatical bonds with other words are called function words (other than this they do not have important lexical meaning). Also, sometimes when the authors are using meaningless words to express their mood or feelings, this can be considered as another example of a function word. To analyze this part, function words are mostly used, but in general, there are thousands of words that were not used.

We divided function-based features into six different groups. Article words usually can be noticed before nouns to show if they mean a general or specific thing. Pro-words are words that are able to substitute of another for a complete sentence. Pronouns are words that substitute of nouns in a sentence so that we can talk about the same thing or person without repeating the name. Auxiliary verbs add grammatical or functional meaning to the related clause and they are a different category of the function-based feature. Conjunctions are used to connect sentences and phrases. Finally, interjections are used to show emotions. To avoid the presence of a number of zeros in case there was more than one in the results, we did not study all of these kind of word. We can see in table 10 some examples that explain the extracted words in this part of the analysis.

Table 4.6. Function-Based Features

Feature	**Description**
Article words total number	A, An, The
Pro-sentence words total number	Yes, No, Okay, Amen
Pronoun words total number	a, an, all, any, another, anything, anyone, anybody, each, both, either, few, everything, everyone, everybody, he, she, his, her, him, hers, himself, herself, I, me, my, mine, myself, ours, our, ourselves, it, its, itself, many, more, most, much, neither, either, nothing, no one, nobody, others, other, several, some, someone, something, somebody, this, that, these, those, there, their, theirs, they, them, themselves, us, we, what, which, who, whose, whatever, whichever, whoever, you, your, yours, yourself, yourselves, yes, no
Auxiliary verbs total number	Am, is, isn't, are, aren't, was, wasn't, were, weren't, be, being, can, can't, could, couldn't, shall, shan't, should, shouldn't, will, won't, would, wouldn't, may, might, dare, do, don't, does, doesn't, did, didn't, have, haven't, has, hasn't, had, hadn't, having, must, need, ought
Conjunction words total number	Him, his, himself, she, I, me, mine, most, my, myself, our, ours, ourselves, that, their, theirs, them, themselves, it, its, itself, neither, one, no one, nobody, nothing, other, others, several, some, somebody, someone, something, many, more
Total number of interjection words	Aah, aha, ahem, ahh, argh, aww, aw, bah, boo, booh, brr, duh, eek, eep, eh, ehm, eww, gah, gee, grr, hmm, humph, harumph, huh, hurrah, ich, yuck, yak, meh, eh, mhm uh-hu, mm, mmh, muahaha, mwahaha, nah, nuh-uh, oh, ooh-la-la, oh-lala, ooh, oomph, umph, oops, ow, oy, pew, pff, phew, psst, sheesh, jeez, shh, shoo, tsk-tsk, uh-uh, oh-oh, uh-uh, uhh, err, wee, whee, whoa, wow, yahoo, yay, yeah, yee-haw, yoo-yoo, yah-uh, yuck, mwah, neener-neener, zowie, zoinks, yow, yikes, va-va-voom, ugh, tchah, rah, sis-boom-bah, shh, ole, lah-de-dah, hup, ich, hubba-hubba, ho-hum…

5. MACHINE LEARNING

Machine learning is a program which is written for a computer and let the computer to imitate the intelligent reactions and abilities of a human. The computers are attempting this by using training data through a program that already collected this data just for this task or by indicating the executions of earlier software for the same program. There are many effective applications able to predict the actions of customers or robot performance optimization by analyzing collected datasets [28].

In some cases, for specific systems, programmers may be incapable of writing a program immediately, therefore, the system tries to learn from several other cases, so it can easily identify a specific issue. The significant mission of machine learning is to perform an algorithm that can recognize and tell the difference between particular input data and the relation of the same data with the classes according to the different feature samples [29].

According to speech recognition, a programmer should modify and change the signals to ASCII codes but how a human is able to recognize more than one accent is a problem in this context that cannot be explained. Or maybe sometimes people are trying to use several words or phrases to explain and describe the same thing according to their particular culture, gender, age, etc. Machine learning is a process for gathering the maximum number of training data of different accents of people of different ages, gender, and other standards, therefore trying to assign these data to define a certain word [30].

During recording data in different times and places and in noisy situations, we face another problem, but here, we can still look forward that we need to solve that problem, instead of writing a new program for every problem. Nonetheless, in some situations, such as network packet routing, it is not possible to write an accurate program that could solve every problems. By taking advantage of machine-learning techniques, we are able to train the system by specifying its training set to support the system in making decisions related to network traffic or destination changes [31].

In this thesis, we used machine-learning algorithms to design a system to specify the gender of an author and their age in a range of four years, for example between 20 and 24. This was achieved by defining a training set in the system, where each category allowed the system to learn the identification standards according to the sets of features. We can see in

Figure 5.1 three datasets holding texts written by male, female, and unknown authors of different and unknown age.

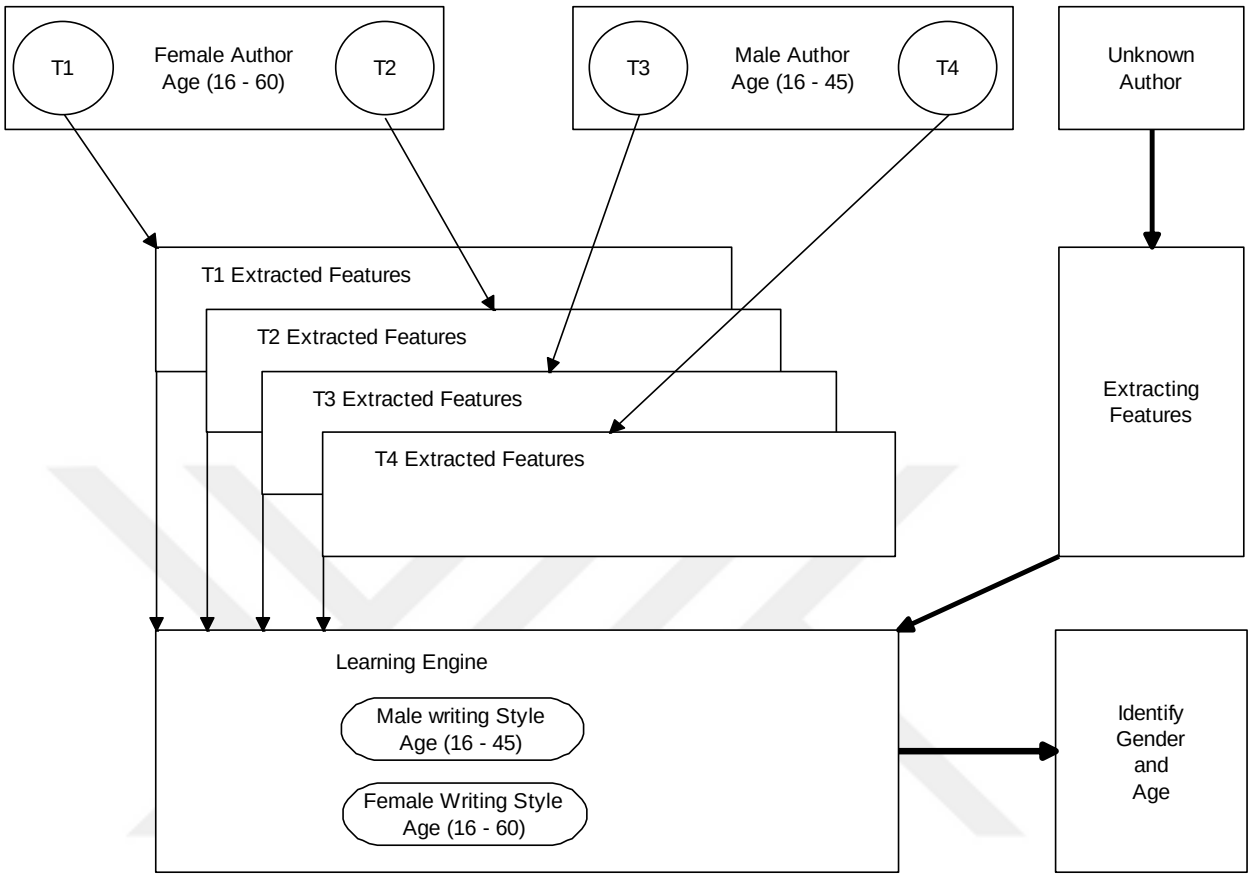

Figure 5.1. Process of identifying the gender of an author

After extracting all the features from the text, we submit the results to the learning engine to train the system. After training the system, the male and female styles of writing are extracted by the engine. During the analysis level, when a written text by a different gender-age-unknown author is submitted, the system selects and extracts the specific features from the text. Then it compares the existing writing styles with the extracted features, and finally, the system can identify the gender and predict the age range of the author.

Each application of machine learning has two major parts. The first one is learning the association, it describes the rule of association by assigning the conditional probabilities of the assigned problem. The second one is classification, every problem is classified to a related class, which has already been located. If the problem related to the recognizing author by a submitted text as a teenager, a middle-aged or an adult, and an elderly, there will be three different classes defined as the teenager class, middle-aged class and elderly class.

According to the rules of learning-association, for one of these three classes, the system will be in charge to provide the input text [32].

For classification techniques, there are different algorithms to be used. However, for training the system when we choose a classifier, the first point we take into consideration is if there is no training data at all or there is but how big is the training dataset? Is there a large training dataset? Is the dataset has little entities? Or, is it a massive dataset? Therefore, the first step in the machine-learning field is to prepare an appropriate training dataset. In the real world, for almost every application, we need a large training corpus to make a system with high performance [32]. If the training set is not labeled, the solution will be to use a particular staff specialist in this field to write the rules. Therefore, queries such as (A and B) or (C and D) must be written as result = Y.

If my training dataset were small, it would be better for applying any high bias classifiers such as, in some situations, Naïve Bayes classifiers perform better than other classifiers [33]. However, to apply all algorithms on bigger training data corpus depends on the advantages of that classifier that we choose to solve our problem.

In the next section, the pros and cons of the Naïve Bayes classifier, J48 decision tree, and support vector machine are briefly explained.

5.1. Naïve Bayes Classifier

Because of the simplicity of this algorithm, it is widely used. This algorithm collects useful information faster than other discriminative algorithms like neural network and random forest, where less training data been used. During real applications, Naïve Bayes behaves successfully, the reason for choosing Naïve Bayes is because it is easy, fast, and reliable, however, its inability to recognize interactions between criteria is one of the disadvantages of this classifier. For more clarification, if there is a client who likes tea, he may also like sugar or sweets, but he might hate eating sweets with tea or mix sugar with tea. In Naïve Bayes, it is unable to recognize the idea that someone who might like eating sweets and drinking tea in separate, while never having them at the same time [34].

5.2. J48 Decision Tree

Classification is creating a model procedure of classes from a dataset that includes class labels. To find the way that the attributes-vector proceeds for a some cases, we used a decision tree algorithm. For the newly generated cases, the classes will be found according to the training cases [35]. This algorithm produces rules to predict the target variable. The critical distribution of the data is easily understandable with the tree classification algorithm support [36]. J48 is an extension of ID3. The extra features of J48 are rule derivation, pruning the decision trees, rating the missing values, value ranges of the continuous attribute.

J48 is an open source Java and the processing C4.5 algorithm. The Weka software gives some options relating to tree pruning so that the probability of overfitting pruning could be utilize as a tool for précising.

In other algorithms, the categorization is repeatedly implemented until every single leaf is clean, therefore, the categorization should have a top accuracy. The decision tree algorithm produces the rules from which a specific identity of that data is produced. Until the tree learns to balance accuracy and flexibility, it will generalize its objective progressively.

5.3. Support Vector Machine

In an overfitting circumstance, the SVM can yield a dataset that is both precise and explicit, even though the sets from the data may not appear detectable. For text classification, the SVM algorithm is highly recommended because of the high dimensionality of its input vectors [37]. The cons of the SVM classifier is that its memory is too complicated and intensive to be clarified to users that have fixed knowledge.

In 1979, the idea of the SVM came to Vladimir Vapnik, but in 1995, an official paper was submitted for the first time in this field, written by Vapnik [38]. In this research, new machine-learning algorithms are used because, in high dimensional data, a hyperplane will be found that has the ability to input data into three classes: female, male, and the age range.

The input data was not always devisable linearly, that is the reason why through the SVM classifier the kernel was defined and the data was put in a higher dimensional space where the SVM classifier could simply classify the data into the three stated separable groups [39].

In general, we faced some computational difficulties caused by casting data in a higher dimensional space, also, some over fitting occurred. The SVM classifier has a problem that could not interacting straightly with the higher dimensional data. Furthermore, for estimating the likeliness of unseen data in the system (VC-dimension) there is a measure that can be simply calculated differently from some other machine-learning algorithms, which they do not contain this kind of measure.

Generally, a number of researchers mentioned that, along with practicing, the SVM classifier is fruitful in classifying the input information data into corresponding classes, also, it can be utilized to solve regression problems. Modern SVM is different from traditional algorithms in three distinguishable ways, which are soft margins, kernel, and optimal hyperplane [40]. To be clearer, it will be discussed in the following paragraphs. When there is a linear discriminant function, the training set is considered as linearly separable and able to match classes of the entire training sets easily. In linearly separable problems, usually there are unlimited numbers of support vectors that split classes. Vapnik and Lerner [41] chose the hyperplane that has the largest space between the nearest instant and hyperplane; it is clarified in Figure 5.2.

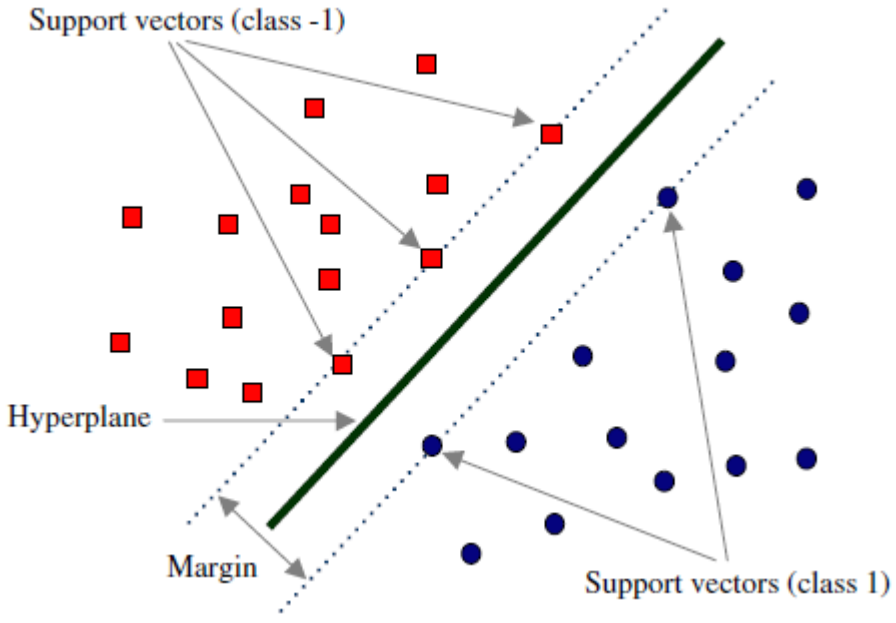

Figure 5.2. Hyperplane within SVM algorithm

The best hyperplane is one that splits circles and triangles by taking into account the closest instances. In the previous linear classifiers, a pattern x is a class of, $y = \pm 1$, which

transforms the pattern into the function of feature vector $\varphi(x)$, where $\hat{y}(x) = w^T\varphi(x) + b$, the parameters w (weight vector: is defined as a vector that has right angle with hyperplane), and b is bias which can determined by running on a training dataset $(x_1, y_1), ..., (x_n, y_n)$, thus $\varphi(x)$ will always be selected by someone who solves the issue.

Selecting the best hyperplane can be find out by using optimization that is expressed in equation the following equation:

$$\min \rho(w|b) = \frac{1}{2}w^2 \qquad \textbf{Eq. 9}$$

where,

$$\forall i\ yi(w^t\varphi(x) + b) \geq 1$$

It is hard to find the solution to equation 9 because the bonds are too complicated. Using the duality of the Lagrangian function, we can simplify this problem and, therefore, one can solve this kind of dual problem by using the following mathematical expression:

$$\max D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i\alpha_i y_j\alpha_j\varphi(x_i)^T\varphi(x_j) \qquad \textbf{Eq. 10}$$

where

$$\begin{cases} \forall i\ \alpha i \geq 0, (lagrange\ multiplier) \\ \sum_i yi\alpha i = 0. \end{cases}$$

The hyperplane direction (w*) can be found out of the solution α^* of the above given formula.

$$w^* = \sum_i \alpha_i{}^* y_i\varphi(x_i) \qquad \textbf{Eq. 11}$$

This was the reason behind finding w* in a simple formula and reconstructing the linear function of the discriminant, as the following equation [16]:

$$\hat{y} = w^{*T} + b^* = \sum_{i=1}^{n} y_i\alpha_i{}^*\varphi(x_i)^T\varphi(x_j) + b \qquad \textbf{Eq. 12}$$

If the prediction of both age and gender is correct, no modifications need to be done. However, if the estimation was mistaken, the factors that explained the hyperplane will be moved ahead to the exact point where the mistake took palace.

The value of scalar is indicated as the rate of the learning and defining how far the parameters will be moved. Selecting the learning rate can safely affect the iteration number until convergence is happening on a linearly separable set.

In the sample space, both equations 10 and 11 can only be a part of dot products, therefore, the equation that behaves with these types of issues, we do not have to calculate φ (x), as a replacement to (x), it is contingent to obtain scalar product (dot product). For nonlinear separable spaces, Vapnik [42] et al proposed to select a kernel function K(x,ẋ), which is capable of performing φ(x) in a higher dimensional feature space. For noise issues, may it is far away to get a strict devisor for the given classes, therefore, Vapnik and Cortes (1995) suggested soft margins [39] to let some of the instants overstep the devisor using positive slack variables (which measure the degree of misclassification of the data x_i) ε = (εI , ... , εn). Furthermore, by using another parameter c they must take control of the greatness of the violation. This caused a change in equation 9 to equation 13.

$$\text{Min } \rho\,(\mathrm{w}, \mathrm{b}, \varepsilon) = \frac{1}{2}w^2 + \mathrm{c}\sum_{i=1}^{n}\varepsilon\mathrm{I} \qquad \textbf{Eq. 13}$$

where,

$$\begin{cases} \forall i \quad yi(w^t\varphi(x) + \mathrm{b}) \geq 1 - \varepsilon \\ \forall i \quad \varepsilon \geq 0 \end{cases}$$

In addition, Eq. 10 will replaced to Eq. 14 for the case of duality.

$$\text{Max } \mathrm{D}(\alpha) = \sum_{i=1}^{n}\alpha - \frac{1}{2}\sum_{i,j=1}^{n} y_i\alpha_i y_j\alpha_j k_{i,j} \qquad \textbf{Eq. 14}$$

where,

$$\begin{cases} \forall i\ c \geq\ \alpha i \geq 0, \\ \sum_i yi\alpha i = 0, \\ K\ is\ the\ kernal\ values\ matrix. \end{cases}$$

In addition, Eq. 12 will change to Eq. 15 as the following:

$$\hat{\mathrm{y}} = \mathrm{w}^{*\mathrm{T}} + b^* = \sum_{i=1}^{n} y_i{\alpha_i}^* k(x_i, x) + b^* \qquad \textbf{Eq. 15}$$

The above-mentioned subjects were just an explanation of what SVM basics are and other studies in this field but are beyond of the scope of this work.

5.4. Stemmer

In the last few decades, there was a big change in data growth around the world. About 60 years ago, this new direction started when a community of researchers went through the first experience of scientific submission, which derived from many different fields [43], was later boosted by the arrival and the subsequent socialization of the internet. The need of researchers made them to search among these collection of big data for information and supporting the performance of some mechanisms to support the task [44]. Today, the process of text classification consists of these types of mechanisms that contain perfection, pervasion, and specialized fields of research in digital forensics. The purpose of text classification is to evaluate documents and treat them automatically to fetch some measures of the data that makes it easy for the users to search and discover the necessary information. By searching and discovering information we mean a high level of standards that do not include only specific questions (e.g., how old are you?), but documents that researchers relate to a phrase or set of phrases that would be helpful and particular (e.g., cute girl), or expressions of an abstract search that indicate that the user did not decide what they are searching for. Those kind of specific questions, phrases, and terms are called attributes and they characterize the user's input to the system. Therefore, the output can be easily ranked depending on the rate of conformed between the attributes and the attribute doing calculation during the current process. A stemmer purpose is to get the word's stem, we mean the structural root, by clearing the affixes that hold lexical or grammatical information around the word. For each situations, those affixes do not customize the word's image where the semantic simplicity has been proven in the literature, mostly in short text documents [45] and highly inflected languages [46], according to precision and recall.

5.5. Purpose of Stemming Algorithms

A stemming algorithm's aim is particular and clear. Yet, it is very hard to yield this goal systematically because no language follows a strict number of rules entirely. This is the reason we do not have perfect stemming that is capable of extracting any stem of any phrase accurately and independent of its features. There are three main purposes behind using stemming algorithms:

The first purpose depends on collecting words in relation to their topic and putting them together in a group. Many words that originally came from the same stem belong to the same term (e.g., write, written, writer). These words were created by using suffixes, prefixes, or infixes, however, in the English language, suffixes will only be considered where infixes and prefixes change the word's meaning and deleting them would cause errors in defining topics [47]. Sometimes there is an exception in very inflected languages like Dutch and German [48], or in some documents that contain some particular subjects such as chemistry, physics, biology, and/or medicine, where suffixes and prefixes are the only way to keep up the meaning of the word. There are two kind of derivations that can be supposed through these suffixes [45]. In the first situation, grammatical information reflected by inflectional derivations related to the gender, age, case, number, or mood. Those derivations have no effect on the original word in its meaning or in the part of speech, which is the classified as linguistic category of the word such as noun, verb, or adjective. In contrast, according to an existing word, derivational suffixes deal with a new word construction, which sometimes shares the meaning, or it may not (e.g., words terminating in -OUS, -IVE, -TION). Deleting those suffixes from created words will give us the stem of that word, which is almost the root of its structure, and then by matching the stems of words, we can identify related words thematically.

The second purpose of a stemming algorithm is precisely connect to the process of retrieving information. The existence of stem word's gives the process of retrieving information some level to be improved, therefore due to the subjects we can focus on the capability for indexing of the text documents, as stems cluster their concepts or attributes expansion to gain more accurate results. The development of attributes is modified by switching the terms, which it contains, with their connected subjects that are also exist in the collection, or by gathering these subjects according to the original attributes. This adjustment can be done transparently and automatically for users where the system will be able to suggest more than one developed formulations of the attributes for users and give them options to decide whether if any of them are more defined or particularly better for any required information. Even if the attribute expansion is way better in principle since the user has more information on what is happening, with the result of stemming cannot be happening right away, as usually stems are complicated and humans cannot understand them [49]. Finally, the combination of word that have the same stem will lead to a decreasing of the dictionary to be considered in the process, while in the unprocessed collection of documents,

the whole vocabulary can be diminished into a set of stems or subjects. This gets us to a point where we can decrease the required space to that used for saving the structures used by the system of retrieving the information and then also ease the computational load on the system.

5.6. Errors in Stemming

In stemming, it could be find that main errors happened, over stemming and under stemming. The first one happens when two words that each has different stems from each other, but they stemmed under the same root. This can also can be identified as a false positive. The second one, under stemming, is when two words should not be stemmed to the same root. This can be identified as a false negative. Lovins stemmer increases the under-stemming errors while at the same time proved that light stemming reduces the over-stemming errors. In contrast, snowball stemmers increase the over-stemming errors while reducing the under-stemming errors u = at the same time. [50] [51].

5.7. Stemmers Used in This Research

5.7.1. Lovins

Lovins proposed this stemmer first in 1986; it was an effective and popular stemmer back then. It implemented the test on 294 endings within the table, 35 transformation rules, and 29 conditions, which have been arranged on a longest match principle [52]. The longest suffix within the word will be removed by the Lovins stemmer. After removing the ending, by using a different table, the word will be recorded and makes several changes to transforming these stems into useful words. According to Lovins, a single pass algorithm will always remove one suffix as the maximum number from one word. The advantages of Lovins algorithm is its speed and ability to deal with deleting double letters within words, for example, “Planning” is converted into “Plan”, and deal with many irregular plurals nouns such as “Ox“ and “Oxen“.

The drawbacks of the Lovins stemmer is the waste of data. Likewise, mostly in the table of endings, all suffixes are non-existent. Sometimes this stemmer is quite inaccurate

and repeatedly fails to create new phrases from the stems or to identify the stems of similar meaning phrases. The cause of this is that the author used the technical vocabulary [53].

5.7.2. Porters Stemmer or Snowball Stemmer

Porters stemmer, created in 1980, is one of the most common stemming algorithms [54] [55]. Lots of adjustments and developments have been conducted and proposed on the main version of this algorithm. It is based off the number suffixes in the English language, which is almost 1,200, and they have been made up from mixing smaller and simpler suffixes. Porters stemmer goes through five steps and within each step, rules are implemented until one of them matches with the conditions. If any rule is approved, then the suffix will be removed correspondingly, and it moves to the next step and waits to be implemented. The response of the stem at the end of the fifth step will be brought back. The rule looks like the following: <condition> <suffix> → <new suffix> for instance, a rule (m>0) EED → EE means, "If the phrase contains minimum one vowel and consonant with EED at the end, that suffix will be converted into EE". Therefore, "freed" is converted into "free" but "seed" stays as it is. This method includes around 60 rules and they are simple to understand. Porter created a more accurate framework for stemming known as "Snowball". The main aim behind this framework is to help programmers and allow them to enhance their own stemmers for other languages. At present, there are snowball stemmers working on many Scandinavian, Germanic, Romance, and Uralic languages as well as Turkish, Russian, and English languages. According to the errors of stemming, we conclude that the Porter stemmer has a reduced error rate when compared with the Lovins stemmer. Yet, the Lovins stemmer is a better stemmer that creates better data reduction. Apparently, the Lovins algorithm is greater than the Porter algorithm because of its massive endings list. However, one of the advantages is Lovins Stemmer works faster and it has a very effective way to deal with time, and because of the wide suffix set plus the five steps of the Porter stemmer compared to the Lovins stemmer, the Lovins needs only two main steps to delete a suffix. [56]

5.7.3. N-Gram

This is an impressive method and it is language-agnostic. In this algorithm, the string-similarity process is utilized to change the word expansion to its stem. An n-gram is a string of (n) from a continuous text section where characters will be taken out. To be more accurate, an n-gram is an extraction of a set of n sequential characters extracted from a phrase. The major concept from this procedure is that identical words will have a high attribution of n-grams. The extracted word would be called a digram if the n equals two and called a trigram if the n is equal to three. Let us take word "CLASSIFICATION" as an example, for its results in the term of the digrams: *C, CL, LA, AS, SS, SI, IF, FI, IC, CA, AT, TI, IO, ON, N* and the trigrams would be: **C, *CL, CLA, LAS, ASS, SSI, SIF, IFI, FIC, ICA, CAT, ATI, ION, ON*, N**, where '*' refers to a space of padding. In digrams n+1 can be noted and n+2 would be noted within the trigrams in a word that includes n characters. Most stemmers are language-specific. Commonly, a value of 4 or 5 is selected for n. After that, for all the n-grams, a text document is evaluated and analyzed. Apparently, the root of any word generally takes a place less often than its morphological form. This means that, in general, the word has an affix to connect with. According to the inverse document frequency (IDF), typical statistical analysis can be utilized to identify them. The advantage of this stemmer is that it is language-agnostic, effective, and useful in many implementations. The disadvantage is it requests an enormous amount of storage and memory to create and save the n-grams and indexes where it is not a very common procedure.

5.8. Weka

In my research, I used Weka version 3.8 to select the SVM, J48 and Naïve Bayes classifiers. The Weka is a flightless bird that has a curious nature and it is only found in New Zealand. Weka [57] is a common machine-learning software suite written in Java, developed at the Waikato University in New Zealand, under a GNU general public license. There are two ways to apply any algorithms to the dataset, either directly from the Weka GUI or Weka can be called from Java code. Weka has tools for preprocessing data, classification, regression, clustering, and association rules. In addition, it is appropriate to develop new schemes of machine learning and it is possible to apply Weka to big data[56].

The main features of Weka contain 49 tools for data preprocessing, 79 regression/algorithms classifications, eight algorithms for clustering, three rule-finder-association algorithms, 15 evaluators for attribute/subset, and ten feature selection search algorithms.

The main interface of the software has three graphical user interfaces: “The Explorer”, which is used for exploring and analyzing data, “The Experimenter”, which includes the environment of the experiment, and “The Knowledge Flow”, which contains a new interface inspired by the model processing [58]. Figure 5.3 presents the Weka Explorer interface used in this research.

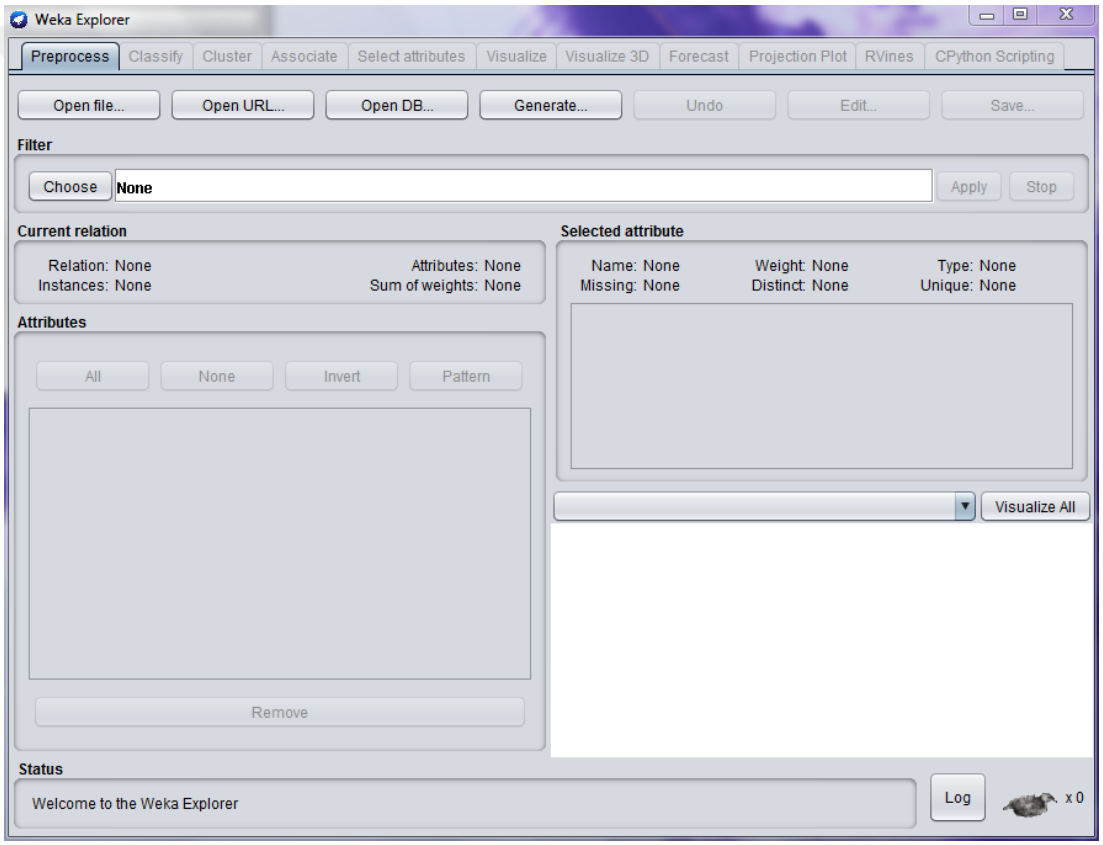

Figure 5.3. Weka Explorer Interface

6. ANALYSIS OF RESULTS

The results of this research are multifaceted in that several different comparisons are made on the way to the larger experiment. The goal at every point is to maximize the predictive accuracy for both age and gender using only the contents of SMS text messages.

Early in the experimental process, it was determined that "10-fold cross-validation" could not be used in place of intentionally split training and testing sets.

Table 6.1. 10-fold cross-validation experiments result of age and gender prediction

Predicting	Method	Correctness (predictive accuracy)		
		Naïve Bayes	SVM	J48
Age	10-Fold Cross-Validation	55.8438%	×	×
		64.8051%	×	×
		63.7841%	79.5869%	×
		62.8797%	×	×
Gender	10-Fold Cross-Validation	65.8365%	×	×

Despite this, there were five experiments running using this method but only Naïve Bayes and one-time SVM experiments were successful. The remaining four experiments and all experiments of J48 took more than 36 hours without building any model. In our study, the time for completing the tests for 10-fold cross-validation experiments was excessive.

Consequently, a training set comprised of 75% of the instances was specified, with the remaining 25% to be used for testing.

Table 6.2. 75/25% training/testing data split experiments result of age prediction

Predicting	Method	Correctness (predictive accuracy)		
		Naïve Bayes	SVM	J48
Age	75% training set 25% testing set	55.6961%	69.5968%	64.9045%
		56.6083%	70.7888%	69.6071%
		47.5381%	69.9008%	64.6557%
		48.6265%	70.9823%	64.8595%
		56.2973%	69.5138%	63.7988%
		56.6083%	70.3811%	64.5866%
		59.3138%	68.6880%	63.2563%

Table 6.3. 75/25% training/testing data split experiments result of gender prediction

Predicting	Method	Correctness (predictive accuracy)			Average
		Naïve Bayes	SVM	J48	Accuracy
Gender	75% training set 25% testing set	64.1132%	78.8363%	76.0582%	73.0026%
		62.8797%	78.0726%	76.4244%	72.4589%
		65.8443%	78.6946%	75.9545%	73.4978%
		66.5906%	78.3456%	79.1023%	74.6795%
		66.2382%	77.9586%	75.7368%	73.3112%
		66.5903%	78.0484%	76.3864%	73.6750%
		69.3791%	77.9275%	76.3726%	74.5597%

After duplicating the experiments (in terms of parameter settings) that used 10-fold cross-validation, the outputs were compared against those produced by the 75/25 split. Given that they were not drastically different, it meant two things:

a. The 75% training split is generally representative of the entire dataset.

b. It was no longer necessary to use 10-fold cross-validation.

The next questions determined by the experiments are whether it is important to weight the term frequency TFT and IDFT differently. This is important because these parameters can yield better results, depending on the nature of the text. Simultaneously, and for the same reasons, the "Convert tokens/words to lowercase" option was varied.

Table 6.4. Result of age and gender identification according to the TFT and IDFT

Algorithms	Age	Gender
Naïve Bayes	56.6083%	66.5906%
SVM	70.7888%	78.3456%
J48	69.6071%	79.1023%

The results in the table 6.1 highlight the differences according to the individual accuracy for each algorithm. Same thing for the experiments within the table 6.2, 6.3 and 6.4 highlight the differences:

a. Using the default settings resulted in the best average accuracy for predicting age.

b. Using the modified settings resulted in a better average accuracy for predicting gender.

c. The majority of experiments, regardless of the classifier, showed that conversion to lowercase yielded slightly better results. This may suggest that the presence of proper case makes a difference or that all-cap (i.e., YELLING) is frequent enough to skew the results if the words are not normalized.
d. The best performing classifier for age prediction was the SVM that recorded 70.9823%, for gender identification it was J48 that recorded 79.1023% where it was slowest to build a model. In spite of Naïve Bayes recording the lowest accuracy to identify age and gender, it was fastest compared to other algorithms at building models.

The experiments to follow involved comparing the use of a stemming algorithm (Lovins and Snowball) and a stop word algorithm (Rainbow). The parameters that gave the best accuracy in the previous experiments were used for these. The total number of 14 experiments showed the following:

a. The use of the stemming algorithms (either Lovins or Snowball) did not significantly change the results for any of the algorithms.
b. The use of the stop word remover, Rainbow, caused the Naïve Bayes algorithm to perform better. However, even with this improvement, it was still inferior to the support vector machine and the J48 decision trees.
c. The use of Rainbow appeared to decrease the predictive accuracy of the support vector machine, although perhaps not significantly.
d. Rainbow had virtually no effect on J48.

Given the empirical evidence with respect to dealing with stop words using the Rainbow algorithm, it suggests that the "StringToWordVector" filter in Weka made satisfactory choices with respect to the words. For this particular text, removing stop words is clearly unnecessary.

Looking at word stemming tells a similar story. Application of either the Lovins or the Snowball stemmer seems to have little if any, effect on the predictive accuracy for either age or gender. This is likely due to the nature of the text. It is possible that SMS text messages are generally terse or may more often be comprised of a subset of the language where the words are shorter and have less variation. This would make sense for the purpose of brevity, where SMS text authors often seem to keep things short.

The final set of experiments in table 14 for age and table 15 for gender used n-gram modeling to determine whether the age or gender could be determined more accurately.

Table 6.5. Applying n-gram to identify age

Predicting	Method	Correctness (predictive accuracy)		
		Naïve Bayes	SVM	J48
Age	75% training set 25% testing set	54.7113%	69.2754%	64.8803%
	n-gram	55.0638%	67.8484%	63.4601%

Table 6.6. Applying n-gram to identify gender

Predicting	Method	Correctness (predictive accuracy)		
		Naïve Bayes	SVM	J48
Gender	75% training set 25% testing set	66.3211%	77.6787%	75.5848%
	n-gram	66.0309%	77.1293%	75.6366%

For each of the three classification algorithms, the n-gram modeling was first applied and then classified using the method of splitting dataset into 75% for training and the remaining 25% for testing. For the sake of completeness, one run for each of the classification algorithms used all of the parameters. The results were as follows:

a. When predicting age, the n-gram modeling produced considerably more attributes.
b. Interestingly, when predicting gender, the n-gram modeling produced about the same number of attributes – albeit different, since many were digrams and trigrams – than the standard word tokenizer using the other parameters.
c. None of the algorithms performed significantly better using n-gram modeling.

N-gram modeling generally took longer to build the model. The longest was the final test, predicting age used 1,025 attributes and J48 took more than 18 hours to run.

After all the experiments to define best algorithm according to the parameter settings and filters to predict age, we needed to get more accurate result whether few data is acceptable for doing this task or not. Therefore, at first we took the whole dataset and used it as a training set, later separated the age into eight different classes, after that we used one class at a time as a testing set to predict the age.

Table 6.7. Instances and prediction accuracy according to the used algorithm for each age class

Age range	Instances	Prediction Accuracy	Naïve Bayes	SVM	J48
16 – 20	20649	91.3361%	×	✓	×
21 – 25	14289	72.6223%	×	×	✓
26 – 30	2665	35.4221%	×	×	✓
31 – 35	118	12.7119%	✓	×	×
36 – 40	597	38.8610%	✓	×	×
41 – 45	240	54.1667%	✓	×	×
46 – 50	10	30%	✓	×	×
51 – 60	20	30.0047%	✓	×	×

As we can see from the table 6.6, the SVM has recorded the highest result compared to Naïve Bayes and J48 algorithms, because the age instance number for class 16 to 20 has a better effect for recording higher accuracy.

In mean while the less-best accuracy can be noticed in the same table, which recorded by Naïve Bayes, it was 30% while we had 10 instances only to do the task.

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

For the classification of SMS message content with the intention of identifying the author's age and gender, the best classification algorithm tested was the support vector machine.

The length of time it took to build models varied between the algorithms, the size of the training set, and the number of attributes for the task. Naïve Bayes was the fastest and J48 decision trees was the slowest.

Using other techniques such as word stemming or removing stop word, had little or no effect on the predictive accuracy for either goal.

Using n-gram modeling caused the classification algorithms to behave differently, as was obvious by the attributes that were created for the purpose of prediction. However, ultimately the n-gram modeling yielded no advantage.

During preprocessing, the content of the SMS messages could have been converted to lowercase letters. This was not done intentionally because it may turn out that maintaining the mixed (original) case yields a better accuracy.

Since Weka provided the option for conversion to lowercase during the "StringToWordVector" filter, that will be used to create the word frequency vector as it is better to perform it there. That way, it can be tried both ways.

Two examples of where leaving the mixed case has the potential to improve the model are:

A: Proper names are more likely to be identified ("Eve" the person versus "eve" the time of day).

B: Using improper capitalization may be indicative of a certain age group.

7.2. Future Work

The current work was conducted on English short text messages only. This work can be enhanced to use in different languages as well. The project's robustness could be improved to deal with the structure of shorter messages, that is, messages of less than 70 characters.

REFERENCES

[1] **H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik,** "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155-161.

[2] **N. Cheng, R. Chandramouli, and K. Subbalakshmi,** "Author gender identification from text," *Digital Investigation*, vol. 8, pp. 78-88, 2011.

[3] **A. Press,** "Family Shunned Over MySpace Hoax, Teen's Suicide," December 7th 2007.

[4] **M. L. Newman, J. W. Pennebaker, D. S. Berry, and J. M. Richards,** "Lying words: Predicting deception from linguistic styles," *Personality and social psychology bulletin*, vol. 29, pp. 665-675, 2003.

[5] **M. M. B. Wolfe,** "Gender Classification of Mobile Application Reviews."

[6] **M. L. Brocardo, I. Traore, S. Saad, and I. Woungang,** "Authorship verification for short messages using stylometry," in *Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on*, 2013, pp. 1-6.

[7] **Z. Miller, B. Dickinson, and W. Hu,** "Gender prediction on twitter using stream algorithms with n-gram character features," *International Journal of Intelligence Science*, vol. 2, p. 143, 2012.

[8] **W. Deitrick, Z. Miller, B. Valyou, B. Dickinson, T. Munson, and W. Hu,** "Author gender prediction in an email stream using neural networks," *Journal of Intelligent Learning Systems and Applications*, vol. 4, p. 169, 2012.

[9] **C. S. Montero, M. Munezero, and T. Kakkonen,** "Investigating the role of emotion-based features in author gender classification of text," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2014, pp. 98-114.

[10] **C. Zhang and P. Zhang, "Predicting gender from blog posts,"** *University of Massachussetts Amherst, USA*, 2010.

[11] **C. Peersman, W. Daelemans, and L. Van Vaerenbergh,** "Predicting age and gender in online social networks," in *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, 2011, pp. 37-44.

[12] **A. Sboev, T. Litvinova, I. Voronina, D. Gudovskikh, and R. Rybka,** "Deep Learning Network Models to Categorize Texts According to Author's Gender and to Identify Text Sentiment," in *Computational Science and Computational Intelligence (CSCI), 2016 International Conference on*, 2016, pp. 1101-1106.

[13] **T. Chen and K. MIN-YEN,** "The National University of Singapore SMS Corpus," 2017.

[14] **L. Burnard,** "Users Reference Guide British National Corpus Version 1.0," 1995.

[15] **B. Stein, N. Lipka, and P. Prettenhofer,** "Intrinsic plagiarism analysis," *Language Resources and Evaluation,* vol. 45, pp. 63-82, 2011.

[16] **L. Breiman,** "Random forests," *Machine learning,* vol. 45, pp. 5-32, 2001.

[17] **A. Abbasi and H. Chen,** "Applying authorship analysis to extremist-group web forum messages," *IEEE Intelligent Systems,* vol. 20, pp. 67-75, 2005.

[18] **M. Corney, O. De Vel, A. Anderson, and G. Mohay,** "Gender-preferential text mining of e-mail discourse," in *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, 2002, pp. 282-289.

[19] **D. J. Hand and K. Yu,** "Idiot's Bayes—not so stupid after all?," *International statistical review,* vol. 69, pp. 385-398, 2001.

[20] **D. Fradkin and I. Muchnik,** "Support vector machines for classification," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science,* vol. 70, pp. 13-20, 2006.

[21] **E. Stamatatos, N. Fakotakis, and G. Kokkinakis,** "Computer-based authorship attribution without lexical measures," *Computers and the Humanities,* vol. 35, pp. 193-214, 2001.

[22] **E. Nel and C. Omlin,** "Machine Learning Algorithms for Packet Routing in Telecommunication Networks," *Bellville, South Africa,* 2004.

[23] **E. E. Abdallah, A. Otoom, O. Abu-Aisheh, D. Omari, and G. Salem,** "Detecting email forgery using random forests and naive Bayes classifiers," in *Proceeding of International Conference on Computer and Software Engineering (ICCSE)*, 2012.

[24] **F. Iqbal, H. Binsalleeh, B. C. Fung, and M. Debbabi,** "Mining writeprints from anonymous e-mails for forensic investigation," *digital investigation,* vol. 7, pp. 56-64, 2010.

[25] **F. Mosteller and D. L. Wallace,** *Applied Bayesian and classical inference: the case of the Federalist papers*: Springer Science & Business Media, 2012.

[26] **F. Peng, D. Schuurmans, S. Wang, and V. Keselj,** "Language independent authorship attribution using character level language models," in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, 2003, pp. 267-274.

[27] **F. J. Tweedie, S. Singh, and D. I. Holmes,** "Neural network applications in stylometry: The Federalist Papers," *Computers and the Humanities,* vol. 30, pp. 1-10, 1996.

[28] **A. Genkin, D. D. Lewis, and D. Madigan,** "Large-scale Bayesian logistic regression for text categorization," *Technometrics,* vol. 49, pp. 291-304, 2007.

[29] **H. Baayen, H. van Halteren, A. Neijt, and F. Tweedie,** "An experiment in authorship attribution," in *6th JADT*, 2002, pp. 29-37.

[30] **H. Deng, G. Runger, and E. Tuv,** "Bias of importance measures for multi-valued attributes and solutions," in *International Conference on Artificial Neural Networks*, 2011, pp. 293-300.

[31] **D. I. Holmes,** "A stylometric analysis of Mormon scripture and related texts," *Journal of the Royal Statistical Society. Series A (Statistics in Society),* pp. 91-120, 1992.

[32] **T. Tenbrink,** "Relevance in spatial navigation and communication," in *International Conference on Spatial Cognition*, 2012, pp. 358-377.

[33] **D. Diermeier and M. Trepanier,** "Measuring reputation," *Kellogg School of Management,* 2009.

[34] **G. James, D. Witten, T. Hastie, and R. Tibshirani,** *An introduction to statistical learning* vol. 112: Springer, 2013.

[35] **M. Jordan,** "Advanced Topics in Learning & Decision Making," *Course material available at www. cs. berkeley. edu/~ jordan/courses/281B-spring01,* 2004.

[36] **G. Kaur and A. Chhabra,** "Improved J48 classification algorithm for the prediction of diabetes," *International Journal of Computer Applications,* vol. 98, 2014.

[37] **A. K. Ameen and B. Kaya,** "Detecting Spammers in Twitter Network," *International Journal of Applied Mathematics, Electronics and Computers,* vol. 5, pp. 71-75, 2017.

[38] **K. Santosh, R. Bansal, M. Shekhar, and V. Varma,** "Author profiling: Predicting age and gender from blogs," *Notebook for PAN at CLEF,* pp. 119-124, 2013.

[39] **Ö. ÖZMEN, K. Ahmad, and A. Engin,** "Sınıflandırıcıların Kalp Hastalığı Verileri Üzerine Performans Karşılaştırması," *Fırat Üniversitesi Mühendislik Bilimleri Dergisi,* vol. 30, pp. 153-159, 2018.

[40] **K. Huang, Z. Zhou, I. King, and M. R. Lyu,** "Improving naive Bayesian classifier by discriminative training," in *Proceedings International Conference on Neural Information Processing (ICONIP 05), Taipei, Taiwan,* 2005.

[41] **V. Vapnik and A. Lerner,** "Generalized portrait method for pattern recognition," *Automation and Remote Control,* vol. 24, pp. 774-780, 1963.

[42] **B. E. Boser, I. M. Guyon, and V. N. Vapnik,** "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory,* 1992, pp. 144-152.

[43] **H. LUHN,** "A stoical approach to mechanized encoding and searching of literary information," *IBM Jouranl of Research and Development,* vol. 1, pp. 390-317, 1957.

[44] **C. Bell and K. P. Jones,** "Towards everyday language information retrieval systems via minicomputers," *Journal of the American Society for information Science,* vol. 30, pp. 334-339, 1979.

[45] **R. Krovetz,** "Viewing morphology as an inference process," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval,* 1993, pp. 191-202.

[46] **M. Popovič and P. Willett,** "The effectiveness of stemming for natural-language access to Slovene textual data," *Journal of the American Society for Information Science,* vol. 43, pp. 384-390, 1992.

[47] **D. A. Hull,** "Stemming algorithms: A case study for detailed evaluation," *Journal of the American Society for Information Science,* vol. 47, pp. 70-84, 1996.

[48] **C. Moral, A. de Antonio, R. Imbert, and J. Ramírez,** "A survey of stemming algorithms in information retrieval," *Information Research: An International Electronic Journal,* vol. 19, p. n1, 2014.

[49] **E. M. Voorhees,** "Query expansion using lexical-semantic relations," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994, pp. 61-69.

[50] **D. P. Chris,** "Another stemmer," in *ACM SIGIR Forum*, 1990, pp. 56-61.

[51] **C. D. Paice,** "An evaluation method for stemming algorithms," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994, pp. 42-50.

[52] **J. B. Lovins,** "Development of a stemming algorithm," *Mech. Translat. & Comp. Linguistics,* vol. 11, pp. 22-31, 1968.

[53] **A. G. Jivani,** "A comparative study of stemming algorithms," *Int. J. Comp. Tech. Appl,* vol. 2, pp. 1930-1938, 2011.

[54] **M. F. Porter,** "An algorithm for suffix stripping," *Program,* vol. 14, pp. 130-137, 1980.

[55] **M. F. Porter,** "Snowball: A language for stemming algorithms," ed, 2001.

[56] **V. Guevara,** "Author Note This is a final project COMP 4910 for the bachelors of computing science from the Thompson Rivers University supervised by Mila Kwiatkowska."

[57] **P. K. P. Lakshmi,** "A Study on Author Identification through Stylometry," *International Journal on Computer Science & Communication Networks,* vol. 2, pp. 653-657, 2012.

[58] **M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten,** "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter,* vol. 11, pp. 10-18, 2009.

CURRICULUM VITA

Ahmad Jamal KHDR

E-mail:	ahmedhormizyar@gmail.com
Nationality:	Iraqi
Place of birth:	Sulaimanyah
Date of birth:	31 / 01 / 1988
Marital status:	Married

EDUCATION

2016 – 2018 Master Degree, Software Engineering department, Technology Faculty, Firat University, Elazig, Turkey.

2009 – 2013 Bachelor degree, Computer Science department, College of Science, Cihan University, Erbil, Iraq.

2007 – 2009 Diploma degree, Electric and Electronic Department, Sulaymaniyah Technical Institute.

WORK EXPERIENCE

- Junior Salesman at ABB Company for electric, robotics, power, heavy electrical equipment and automation technology 2008/2009.
- I.T technician at Ministry of Endowment and Religious Affair 2008 /2012.
- I.T technician at Diabetic Child Association 2012/2016 Volunteered.
- Data entry at Toyota Cihan Motors 2013-08-01 / 2013-09-01.
- Senior Data Entry Manager at Toyota Cihan Motors 2013-09-02 / 2013-12-31.
- Hawler Computer Science Institute 2014 — 2016.
- Research Assistance at Selahaddin University 2016 – present.

PUBLICATION

- ÖZMEN, Ö., Ahmad, K. H. D. R., & Engin, A. V. C. I. (2018). *Sınıflandırıcıların Kalp Hastalığı Verileri Üzerine Performans Karşılaştırması*. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 30(3), 153-159.

- KHDR, A., VAROL, C., (2018). *Age and Gender Identification by SMS text Messages*, 3rd international conference on artificial intelligence and data processing(IDAP). 28-30 Sept 2018.