FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Text Mining - A Toolbox for Text Classification

Paulo Sérgio Vieira da Costa

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rui Carlos Camacho (FEUP)

Co-Supervisor: Célia Talma Gonçalves (ISCAP)

April 26, 2020

Text Mining - A Toolbox for Text Classification

Paulo Sérgio Vieira da Costa

Mestrado Integrado em Engenharia Informática e Computação

April 26, 2020

Abstract

In this thesis we developed a text mining tool capable of doing the pre-processing of text documents as well as their classification. The main focus was the development of a platform capable of executing operations of data extraction, natural language processing, data classification and evaluation of the constructed models, given a corpus of labeled documents. This is deeply involved with predictive analysis, used in several business and data mining research fields. This document classification tool was tested with a sentiment analysis dataset, about movie reviews, where the documents are classified with a positive or negative polarity. The goal was to achieve an accurate predictive classification, while presenting the user with a user-friendly interface, for easier interaction with the platform. We have also made an evaluation and comparison between several Machine Learning algorithms, using accuracy as the evaluating metric, to understand in which circumstances we could achieve the best results.

Resumo

Nesta tese desenvolvemos uma ferramenta de text mining capaz de fazer o pré-processamento de documentos de texto assim como a sua classificação. O foco principal foi o desenvolvimento de uma plataforma capaz de executar operações de extração de data, processamento de linguagem natural, classificação e avaliação dos modelos contruídos, dado um conjunto de documentos rotulados. Isto encontra-se profundamente análise preditiva, usado em vários campos comperciais e de pesquisa sobre extração de data. Esta ferramenta de classificação de documentos foi testada num dataset de análise sentimental, acerca de críticas de filmes, onde os documentos estão classificados com polaridade positiva ou negativa. O objetivo foi obter um classificação preditiva precisa, enquanto é apresentado ao utilizador uma interface amigável, para uma interação fácil com a plataforma. Também fizemos uma avaliação e comparação entre vários algoritmos de Machine Learning, para perceber em que circustâncias conseguimos obter os melhores resultados.

Acknowledgements

I would like to thank everyone that helped me in the path of this experience. To my parents and my sister, for helping me become the person I am today. To the rest of my family and friends. A especial thank you to my grandmother, Fernanda, to which this achievement would bring more proudness than it brings myself.

Paulo Sérgio Vieira da Costa

"Rather than accepting confusion and uncertainty, we create superstitions and beliefs that make us feel like we have control over our lives. In fact to not form a superstition or belief or guess about the world around us is to be powerless, even when the superstitions are unconnected to reality."

Michael Stevens

Contents

List of Figures

List of Tables

Abbreviatures and Symbols

ARFF	Attribute Relationship File Format
CRISP-DM	Cross-Industry Process for Data Mining
CSV	Comma Separated Values
DM	Data Mining
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
IDF	Inverse Document Frequency
K-NN	K-Nearest Neighbors
KDD	Knowledge Discovery in Databases
MNB	Multinomial Naive Bayes Classifier
NB	Naive Bayes Classifier
NER	Named Entity Recognition
NLTK	Natural Language Toolkit
NPV	Negative Predictive Value
PDF	Portable Document Format
PPV	Positive Predictive Value
PRC	Precision-recall Curve
ROC	Receiver Operating Characteristic Curve
RF	Random Forest
SVM	Support Vector Machine
TF	Term Frequency
TM	Text Mining
TN	True Negatives
TP	True Positives
TPR	True Positive Rate

Chapter 1

Introduction

This project was dedicated to the construction of a framework capable of offering the user tools to select, process and classify data from a corpus of documents. The goal was to create a dynamic platform, capable of executing operations of data pre-processing and document classification, presented with a user-friendly interface to facilitate the interaction with the system. Throughout this thesis, we presented the proposed methodology used to achieve our goals, the different modules of the application and the different operations that together compose the selection, pre-process and classification actions, applied to several datasets. We have done an analysis of the state of the art approaches used in data mining to achieve this solution, compared different document processing techniques and specified how this framework was developed. Then, we describe several algorithms used in natural language processing and for text classification, used to extract important features and implementing a predictive analysis approach. This framework is capable of classifying different text datasets, after learning the causality between the weight of important features present in the corpus and their polarity label, we have used several data mining algorithms in order to evaluate the accuracy in the classification of the datasets.

1.1 Context

Text mining evolved to offer an approach to efficiently retrieving information from text, creating tools for the analysis of the causality between the features present in processed datasets and their positive or negative polarity. Processing and categorization of data are the two main text mining processes for the manipulation of extracted data. The interest for predictive analysis in several different fields has increased over the past years and this work can be used to make better predictions about the future. It could be useful, for example in stock market, for precision of values, accordingly to specific scenarios. It is a complex process of deriving quality information though the knowledge gained from the associations found in the previously categorized dataset, in order to create a model capable of achieving a high accuracy in the classification stage.

1.2 Motivation

With the increment of available information, there is a necessity for the creation of a framework capable of extracting knowledge in a dynamic and efficient way. This serves as a motivation for the devotion of extracting information from a dataset and transform it into a structure for further processing and classification, a process known as data mining. The discovery of patterns of relationship between the entities is essential for predictive modeling, through the derivation of associating rules between the relations present in the documents. This is also essential for further extraction and visualization of data. With the rising necessity of working with larger and larger datasets, it is important to try to maximize the efficiency in the processing of the dataset but also the quality of the predictive analysis. There is a demand for the increase of efficiency in retrieving information, in a way of dynamically matching the expectations of rate of evolution in this field. This induces interest for the creation of tools suited for the processing and derivation of knowledge in a sustainable and efficient way.

1.3 Objectives

In this thesis we have implemented a platform capable of loading and processing a corpus of labelled documents. With this application, the user can select one or more pre-processing operations to apply to a dataset, which then will be processed. After, it is generated a vocabulary of the features present in the processed dataset and it's generated an attribute-relationship file, containing for each of the documents the weight of the features present in the vocabulary. The goal is now to understand the influence that each feature present has in the document label, that can be positive or negative. From that file, it is now possible to apply several classification algorithms, with the goal of making an accurate prediction in the classification of the dataset, based on the previous knowledge gained. The main objective is to achieve an accurate prediction in the classification of the datasets. We have done an in-depth analysis of the different accuracy scores obtained from the classification algorithms, in datasets processed after applying several pre-processing operations.

1.4 Dissertation Structure

The present dissertation is composed by five chapters. Chapter 2, describes the process of data mining, as well as the key concepts related to this topic. Chapter 3, details in depth the text mining process, with the analysis of the system architecture. Chapter 4 presents the case study and the conclusions about the model evaluation. Finally, Chapter 5 presents the achieved results and details about the future work.

Chapter 2

Data Mining

2.1 Data Mining

Data mining is the process of extraction of knowledge from the associations of features present in the dataset. By finding and studying the relationships present in the data, it's is possible to map important information, classify it and build models (1) for prediction of future behaviours, based on the knowledge obtained from previous data processing. This is inherently connected with machine learning, a task known as predictive analysis. Data Mining is composed of two main steps - Data Pre-processing / Preparation and Data Processing / Mining. Data Pre-processing involves the operations of data cleaning, selection and transformation, while Data Processing includes data mining, pattern evaluation, and knowledge representation. It is a field with a high commercial value, since that the extraction of knowledge is a very important topic for knowledge discovery in databases that every major company has. It is capable of being applied to predictive analysis, a crucial task for every scenario simulation.

2.1.1 Data Mining Steps

Firstly, it is important to understand the key concepts and main steps that compose the data mining process in order to further explore how the derivation of information is achieved. Bellow are described the main steps of data mining.

- **Data Pre-processing / Preparation** - Data Preparation is based in operations of data cleaning, data selection and data transformation.

  Data Selection - It is important to select relevant data for the purpose of the study in question. The dataset should be robust and the information for the research should be gathered from a reliable source (2). To achieve the goal of accurately classifying a dataset, it is important to choose a labelled one, with each document being assigned with a positive or negative sentiment. This will be later important in the classification stage, in order to

extract a relationship between the corpus and their label, while applying different classification algorithms, to see which ones are capable of achieving the best accuracy score in this predictive analysis process.

Data Cleaning - An essential stage for the pre-processment of the documents. It is comprised with several operations necessary for the cleaning of irrelevant data, giving control to the user of which features is he interested in retrieving, and defining how the tokenization of features shall be arranged. This contains a set of very important operations, like the removal of special characters, stemming, lemmatization, named entity recognition, in a way of reducing the number of actual relevant words present in the dataset, for the future extraction of the most important features present in the corpus (3).

Data Transformation - Converting the data from one structure or format to a different one. In this particular case, that would refer to the processing of the csv file or the text file and converting them into a data structure in which the processing operation will take place (4).

- **Data Processing / Mining** - This phase composes a different number of processes, such as the detection of anomalies, to ensure that the data received has no errors, pattern evaluation, associative learning, made with the mapping of relationships between the entities present in the text and clustering, with the aim of grouping entities with similar characteristics and the classification of the data based on the previous mapping, to minimize the modeling error and maximize the data classification accuracy (5).

These are the main steps that compose the data mining process. Bellow it will be represented a few more important concepts that occur in this dynamic derivation of knowledge.

1. **Natural Language Processing** - An essential stage that involves natural language processing. In this stage, the documents are being pre-processed and there are applied a series of operations to sanitize the text, such as the removal of special characters, synonyms substitution, stemming, lemmatization and named entity recognition (6). These will be the operations applied to the corpus of documents to be ready for the processing stage.

2. **Clustering** - This is the process of grouping into clusters segments of information that share similar characteristics between them. A similarity function is used to understand the degree between them, being the clusters formed by different types of entities present in the corpus of documents. It is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) (7).

3. **Regression** - Main process for the association of relationships between variables. This can be specially applied for predictive analysis, where there are defined causal relationships between the independent and dependent variables present in the corpus. Both regression and

classification are essential features in predictive analysis. In regression one is predicting a numerical quantity whereas in classification one is predicting a non numerical variable (8).

4. **Classification** - This function refers to the attribution of predefined classes to the processed corpus. Based on the tags present in the dataset, it will be divided into a training and a test set, normally 70-30%, in which the training set (70%) of the dataset will be used for the study of the relationship between the weight of the entities present in the corpus and their respective tags, to then apply to the test set, with a different number of algorithms, such as the multinomial naive-bayes, k-nearest neighbor and decision trees. After this process it will be possible to analyze the accuracy achieved in the classification of the test set (9).

It is also important to state the difference between supervised and unsupervised learning techniques, as the data mining algorithms belong to one of the two categories. In the first one, the knowledge is obtained from data that hasn't been categorized before, untreated data, analyzing common properties and acting based in the differences present in the database. Unsupervised learning is specially important in clustering and neural networks creation. In contrast, supervised learning relies in the analysis of training data to produce a function that will map similar scenarios, based in previous deducted knowledge. This will be the main focus around this topic, where it will be extracted relationships from a previously categorized dataset. This one in particular is used for sentiment analysis, as a polarity based dataset labeled as positive or negative.

2.1.2 The CRISP-DM Methodology

CRISP-DM stands for Cross-Industry Process for Data Mining. The CRISP-DM methodology (10) provides a structured approach to planning a data mining project. It is a robust, flexible and powerful methodology, used in data mining to create platforms, providing an approach to this issue. It is composed of six main stages - Business understanding, Data understanding, Data preparation, Modeling, Evaluation and Deployment. A schema of the methodology can be seen in Figure 2.1.

- **Business Understanding -** Setting the objectives, from a business perspective, define a project plan to achieve the proposed goals and lay out the criteria that will be used to determine whether the project is successful from a business point of view.
- **Data Understanding -** Initial data collection, data description, including format, quantity and number of attributes. It involves the mapping of key attributes, the properties and relationships between them and a verification of the quality of the selected data.
- **Data Preparation -** Selection of data, including deciding about which data will be included and removed, cleaning of data, construction of the acquired data, with the new generated attributes and the integration of the processed data in the system.
- **Modeling -** Selection of the modeling technique, with the specification of the algorithms used for processing and classification of data, the modeling assumptions, such as the class

attributes having an uniform distribution, no missing values allowed, etc. The generation of a test design, that describes the intended plan for training, testing and evaluating the models and then building the models with the set parameters, description and interpretation of the models and finally model assessment. This stage includes the interpretation of the models according to the domain knowledge, the data mining success criteria, the desired test design, summarization of results, list the qualities of the generated models and ranking their quality in relation to each other.

- **Evaluation -** Assessment of data mining results achieved, with the summarization of results in words of business success criteria, including a final statement regarding whether the project meets the initial stated business objectives or not. It should also include a review of the process, highlighting the tasks that have been missed and those that should be repeated, in order to prepare the generated models for further actions.

- **Deployment -** Development of a deployment plan, with a summarization of the chosen deployment strategy, including the necessary steps and how to perform them. It should also be developed a monitoring and maintenance plan. Lastly, there is a production of a final report, the final written report of the data mining engagement, including the previous versions, description of the features and a statement of the achieved results. There is often a final presentation, a meeting at the conclusion of the project at which the results are presented to the customer.

Figure 2.1: The CRISP-DM Methodology.

2.1.3 Classification Algorithms

Classification (9) is a very important task in data mining, where data is divided in a test and training set, where the training set will be used to study the relationships between the features and their respective classification. After, the test set will be classified accordingly to the knowledge extracted from the training set. After that, it will be made an evaluation of the model created. There are several classification algorithms that will be analyzed in this thesis - Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbor (K-NN), J48 and Random Forest (RF).

1. **Naive Bayes** - Naive Bayes is a technique for classifiers construction, where it is possible to create models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from a finite set. In Naive Bayes classifier, it is assumed that the value of a particular feature is independent of the value of any other feature, given the class variable, considering that each of the features contribute independently to the probability of influencing the class label. Working with a probabilistic classification method, based on applying Bayes' theorem with strong independence assumptions between the features, the naive Bayes classifiers can be trained very efficiently in a supervised learning position (11).

2. **Random Forest** - This is an algorithm for data classification and regression, that involves a multitude of decision trees at training, while outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. They are used as a predictive model to go from observations about an item, represented in the branches, to conclusions about the item's value, represented in the leaves. It creates a model for decisions and their possible consequences, including the chance of event outcomes, resource costs, and utility. When the target variable can take a discrete set of values they are called classification trees. Decision trees where the target variable can take continuous values are called regression trees. In these types of structures, leaves represent class labels and branches represent conjunctions of features that lead to the class labels. By using this model it's possible to do a predictive analysis, classifying the model based on previous input variables (12).

3. **K-Nearest Neighbor** - This is a distance-based algorithm used for data classification and regression, where the output depends on whether K-NN is used for classification or regression. In the first case, the output is a class member. An object is classified by a vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It uses a weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. A common weighting algorithm consists in giving each neighbor a weight of 1/x, being x the distance to the neighbor. If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In K-NN regression, the output works as the property value for the object. This value is the average of the values of k nearest neighbors (13).

4. **Support Vector Machines** - Support Vector Machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a training set, with each element marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. This algorithm is normally used for dataset classification, with the binary class tags. An SVM model is a representation of the attributes as points in space, mapped so that the attributes of the separate categories are divided by a gap. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. Support vector machines can be used both in linear and non-linear classification (14).

5. **Decision Trees** - A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is an algorithm that only contains conditional control statements. A supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning decision rules derived from the data features. A decision tree typically starts with a single node, which branches into possible outcomes. A decision node, represented by a square, shows a decision to be made, with the end node showing the final outcome of a decision path (15).

2.2 Text Mining

Text Mining is a subfield of data mining, using a similar approach, applied particularly to the extraction of knowledge from text files. There are several tasks that Text Mining can address:

- **Information Retrieval** - Collecting and preparing a corpus of document, after selecting relevant and robust data for the purpose of the study. To achieve the goal of accurately classifying a dataset, first it's important to focus of the data selection process. This includes the selection of a relevant dataset, with a description of the data that has been acquired, including its format and quantity, giving the number of records and fields in each table. Evaluation of the quality of data and the ability of meeting the established requirements are some of the main tasks present in information retrieval (2).

- **Natural Language Processing** - Use of several operations to analyze the original text, parsing it to be ready for classification. This involves operations of special characters removal, stopwords removal, stemming, lemmatization, synonyms substitution and Named Entity Recognition (NER) (16).

- **Term Weighting** - The Term Frequency–Inverse Document Frequency (TF-IDF) is an effective metric for calculating the weight of the attributes present in the documents, relative to the whole corpus. Term frequency (TF) deals with counting the number of occurrences of each attribute present in the dataset, for each document. The Inverse Document Frequency

measures how much information a word provides. The TF-IDF aims to reflect how important a word is to a document relative to the corpus, increasing proportionally to the number of times a word appears in the document and decreasing by the number of documents in the corpus that contain the word. By this way, a high weight in tf–idf is reached by a high term frequency and a low document frequency of the term in the whole dataset. By assigning the weight of a term proportionally to its occurrences in all the dataset, it is possible to do a more accurate extraction of the importance of features present in the corpus. This allows for the creation of vocabularies, containing a unique set of the words present in the dataset. This is important for mapping the important features, accordingly to their weight. (17)

- **Sentiment Analysis** - It is a branch of natural language processing, that aims to extract the polarity and subjectivity present in text, in order to create a sentiment(18). The polarity is usually defined as 0 or 1, representing False or True respectively. The polarity is defined as a negative to positive sentiment, while the subjectivity ranges from a personal belief to a known fact, represented from 0 to 1. The most common type of datasets used for sentiment analysis are polarity based, where the documents present in the corpus are classified as either positive or negative. This is relevant for the classification stage, where the processed datasets are classified, based in previous obtained knowledge about the relationships between the weight of the words present in the corpus and their polarity label.

- **Predictive Analysis** - Predictive Analysis is an essential feature of text mining, and data extraction in general. Feature extraction has the goal of using processed data to forecast results, a process known as knowledge discovery (19). By splitting the processed dataset into a train and test set, it is possible to use classification algorithms, such as Naive Bayes and K-Nearest Neighbor to study the relationships between the attributes and their tag in the classified training set to predict the classification in the test set.

2.3 Model Evaluation

Model evaluation refers to the metrics and methods to assess the finality of the models constructed. Sensitivity and specificity are statistical measures of the performance of a binary classification test, with most important parameters are the accuracy, precision, recall and f1 score. The accuracy score refers to the percentage of the train set that was accurately classified, after being trained. The precision score refers to the ratio between the number of true positives and the sum of the total true positives and false positives. The precision is the ability of the classifier not to inaccurately label as positive a sample that is negative. The recall score is the ratio between the number of true positives and the sum of the true positives with the number of false negatives. The recall is the ability of the classifier to find all the positive samples. The F-Measure score can be interpreted as a weighted harmonic mean of the precision and recall, reaching its worst score at zero and its best value at one. True positives, refer to the number of positive instances correctly classified as positive, true negatives refer to the number of negative instances correctly classified as negative,

false positives are the number of negative instances incorrectly classified as positive and false negatives are the number of positive instances incorrectly classified as negative. The confusion matrix represents a visualization of the true positives/negatives and false positives/negatives. TP Rate is the true positive rate, represented as the ratio between true positives and the total of positives. FP Rate is the false positive rate, represented as the ratio between the false positives and the total of negatives. ROC Area is a performance measurement that represents degree of separability. It tells how much model is capable of distinguishing between classes. The ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR), with TPR on the y-axis and FPR on the x-axis. Positive predictive value (PPV) is calculated as TP / (TP + FP), representing the ratio between true positive values and the sum of true positives with false positives. Precision-recall curve (PRC) is a plot of precision (PPV) vs. recall (sensitivity) for all potential cut-offs for a test. Sensitivity can be calculated as TP / (TP + FN). All of these formulae can be consulted in Table 2.1.

Formulas used in our study	
Sensitivity or True Positive Rate (TPR)	$TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = 1-FNR$
Specificity or True Negative Rate (TNR)	$TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = 1-FPR$
Precision or Positive Predictive Value (PPV)	$PPV = \frac{TP}{TP+FP} = 1-FDR$
Negative Predictive Value (NPV)	$NPV = \frac{TN}{TN+FN} = 1-FOR$
Fall Out or False Positive Rate (FPR)	$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} = 1-TPR$
Miss Rate or False Negative Rate (FNR)	$FNR = \frac{FN}{P} = \frac{FN}{FN+TP} = 1-TNR$
Accuracy (ACC)	$ACC = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN}$
F1-Score (F1)	$F1 = \frac{PPV*TPR}{PPV+TPR} = \frac{2TP}{2TP+FP+FN}$
Cohen's Kappa (k)	$k = \frac{p_o-p_e}{1-p_e} = 1-\frac{1-p_0}{1-p_e}$

Table 2.1: Formulas used in the evaluation stage.

2.4 Machine Learning and Text Mining Tools

Python was the selected programming language for the implementation of this project, for the availability of critical modules necessary for natural language processing, data extraction and classification. A wide set of existing tools is available for the practice of Data Mining. Some of them include:

2.4.1 Text Mining Tools

- **Weka** - Weka (20) is a software that provides several machine learning algorithms for data mining tasks. It contains algorithms for data preparation, classification, regression, clustering, association rules mining and visualization. It is essential for predictive analysis, achieved with the classification of the datasets, by previously generating an attribute relationships file with the weight of the features associated with their tag, as a way of further extract knowledge from the datasets.

- **NLTK** - Natural Language Toolkit (NLTK) (21) is an essential module for natural language processing. It provides tools such as stopword list, a module for stemming and lemmatization and a word tokenizer. These are all essential modules for the document preprocessing stage. It also provides a wordnet dictionary, used for substituting the attributes present in the text for their synonyms.

- **Spacy** - Spacy (22) is a module used for the named entity recognition. The aim of the use of this tool was to map entities present in the corpus, identifying them, from a set of entities present in a library, and replacing the words that compose it with a unique token. For example, Nile River would be replaced for Nile_River and identified as a Location (LOC), as a way of reducing the number of total tokens in the text, prior to the extraction of features. This module is divided into a different number of classes for the recognition of entities, such as Person, Location, Event and Date, making it possible to identify and map the entities found in the text. More of this usage is further detailed in chapter 3, in the section about pre-processment stage, named entity recognition.

- **Scikit-learn** - Scikit-learn (23) is an essential module for feature extraction and model classification. It was used the TfidfVectorizer function to create a matrix of the tf-idf values along the corpus, as a way of doing the term weighting. It also has a splitter function, with the goal of dividing the set into a test and training subset. It provides algorithms for the classification, such as Multinomial Naive Bayes and K-Nearest Neighbor. It was possible to combine these tools to measure and evaluate the model created, with the extraction of data about the accuracy, recall and f1 scores.

- **RapidMiner** - RapidMiner (24) is software platform that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. It is used for business applications as well as for research and application development, supporting all the steps of the machine learning process, such as data preparation, results visualization, model validation and optimization.

- **KNIME** - Konstanz Information Miner (KNIME) (25) is an open-source tool for data analytics, reporting and integration platform. KNIME integrates several components for machine learning and data mining, using a modular pipeline for data processing. It also provides the user with a graphic interface capable of performing different data operations, such

as pre-processing (Extraction, Transformation, Loading) or modeling, data analysis and visualization.

2.4.2 Other Tools

- **Tkinter** - Tkinter (26) is a Python module used for the creation of the user interface. It was possible to create and combine the several windows with different scripts, accordingly to the option selected from the user. From choosing to load a file, to the pre-processing operations and the creation of an attribute relationship file for the model classification, the user was provided with a different set of options to apply to the dataset being analysed.
- **MySql** - MySQL (27) is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL is a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions. By using it, it was possible to store the datasets, as well as their class attributes.

2.5 Related Work

With major classification tools already available to the public, is was created a platform capable of doing pre-processing operations do a dataset, chosen by user, and then export it into an output file, with the processed dataset. Then, it can then be integrated with other platforms, such as Weka (20) or RapidMiner (24) for processing operations. The major platform studied was Orange (28), a machine learning and data mining tool for data analysis through Python scripting and visual programming. It includes several features, such as interactive data analysis and component-based assembly of data mining procedures. In the selection and design of components, it is focused on the flexibility and with an intention let the user write simple and clear scripts in Python, for the implementations of computationally-intensive tasks. Orange is intended both for experienced users and programmers, as well as data mining students.

Chapter 3

Proposed Framework / Methodology

This chapter describes in a detailed way the implementation of the proposed framework. The main goal was to provide the user with a flexible and dynamic interface, capable of offering tools for text mining. It was possible to load and process a set of documents, in a csv file format and apply several different combinations of pre-processing operations to create different datasets. After this, it is generated a dictionary, created an attribute relationship file, composed of a matrix of the weight of the words and the class polarity, for each one of the datasets. Next, they are classified with a set of algorithms, such as Naive Bayes and K-Nearest Neighbor. In the end, the models are evaluated in words of the accuracy and other precision scores achieved using these predictive models. The framework architecture diagram is represented in Figure 3.1.

3.1 Framework Architecture

The framework is divided in four main modules - Loading, Pre-processing, Processing and Evaluation. Bellow it is described in detail about the actions that each one of these modules executes and the interactions between them, working altogether.

1. **Dataset Load** - The first step executed is loading and saving the dataset, in the csv file. After the initial dataset load, it is possible for the user to see each document and their respective tag. Then, it is generated a vocabulary with all the unique words present in the corpus. After this, the application goes to the pre-processing stage, where the user can choose between different sanitizing operations to apply to the dataset. The load view can be seen in Figure 3.2.

2. **Pre-processing Module** - The second step of the framework is pre-processing (29). It will be displayed a number of operations that the user can choose to apply to the texts, such as character removal, stop-words removal, named entity recognition, stemming, lemmatization and synonyms substitution. After the user selects the wanted operations, they will be processed in a pipeline, where operations like stemming and lemmatization are the last ones

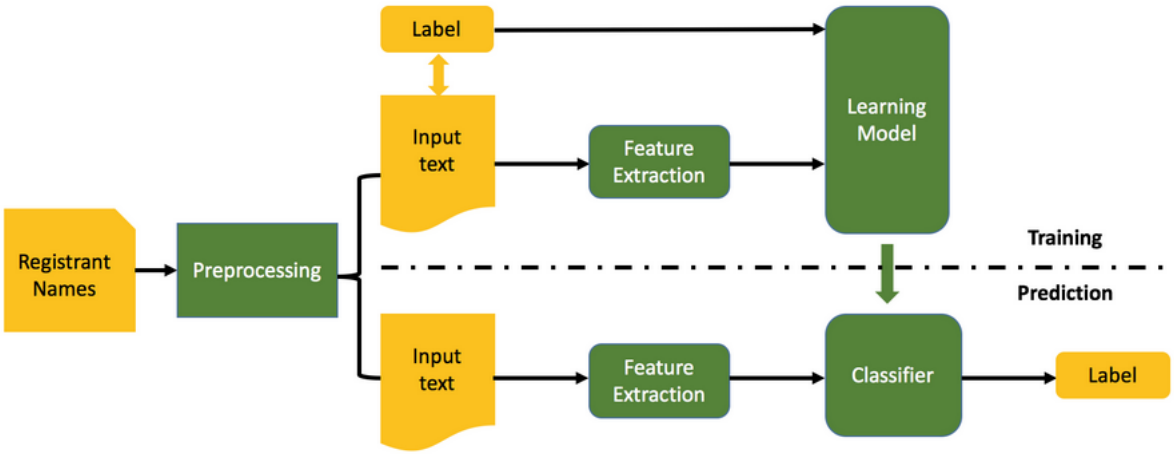

Figure 3.1: Framework Architecture Diagram.

processed, as due to their nature, it couldn't be applied stemming before synonyms substitution for example, as it wouldn't be a dictionary synonym match for a stemmed word. After this operations it will be displayed in the terminal a dataframe with corpus of documents after the natural language processing was applied and their respective tag, and the user reaches the processing stage. The pre-processing view, responsible for these operations is represented in Figure 3.3.

- **Character Removal** - The goal of this operation is to remove the special characters present in the text. It is used a Regular Expressions matching function to maintain only characters classified as words. The filters applied were '\w+', that matches any word character (alphanumeric underscore), with the '+' quantifier matching one or more of the preceding words and '\r' to include the carry over. All the other characters were removed, as a way of providing a tokenization of the words present without unnecessary characters in the pre-processment stage. It's important to notice that the underscore for example, is maintained, because underscore words are recognized named entities, and others, such as distance, monetary or time measures that are important recognized entries .
- **Stop-words Removal** - This operation removes all the stopwords contained in the list defined by the NLTK package. This mainly removes conjunctions, prepositions and other disposable words present in the dataset. It's worth noticing that all of these pre-processing operations have the goal of reducing the number of number of words present in the processed dataset.
- **Stemming** - Stemming refers to the process of reducing derived words to their word stem, base or root form. The stem isn't necessary to be identical to the morphological root of the word, and the objective is to map related words to the same stem. For example, the word 'connecting', 'connects' and 'connected' would all be reduced to

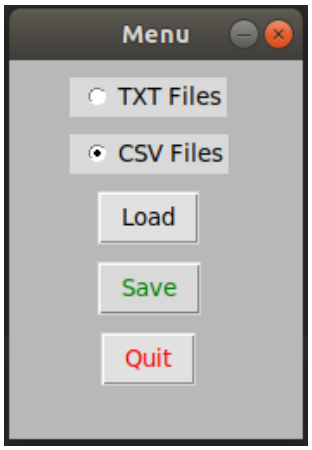

Figure 3.2: Main menu

'connect' and stored as a single feature. With stemming is possible to reduce derived words to the same root, reducing the number of words present in the document corpus.

- **Lemmatization** - Lemmatization operates in a similar fashion to stemming, but it acts by analyzing different forms of a word as a single item, identified by the word's lemma. A Lemma is a representation of the canonical form for each rendered words.
- **N-Grams** - N-Grams represent contiguous sequences of n-items in a sentence. N can be one, two or any other positive integer, but usually it's not considered a very large N because big sequences of the words rarely appear in different documents. In the toolbox, it is given the user the option of choosing the number of ngrams to apply in the document processing, were by default, the number of n-grams used is 1.
- **Named Entity Recognition** - The goal of this operation is to recognize and map entities present in the text. With the usage of the spacy module, it was possible to identity 16 different types of entities(30).
  - PERSON - People, including fictional.
  - NORP - Nationalities, religious or political groups.
  - FAC - Buildings, airports, highways, bridges, etc.
  - ORG - Companies, agencies and institutions.
  - GPE - Countries, cities and states.
  - LOC - Non-GPE locations, mountain ranges and bodies of water.
  - PRODUCT - Objects, vehicles and foods.
  - EVENT - Named natural catastrophes, wars, sports events, etc.
  - WORK OF ART - Titles of books, songs, etc.
  - LAW - Named law documents.
  - LANGUAGE - Any named language.
  - DATE - Absolute or relative dates or periods.

- TIME - Times smaller than a day.
- PERCENT - Percentage, including "%".
- MONEY - Monetary values, including unit.
- QUANTITY - Measurements, of weight or distance.
- ORDINAL - "first", "second", etc.
- CARDINAL - Numerals that don't fall under another type.

It is important to notice that when any of these kinds of entities are identified in the corpus, the words that compose them are substituted by a single words, separated by underscores. For example, 'Albert Einstein', would be replaced by Albert_Einstein and classified as a Person.

- **Synonyms Analysis** - This operation aims to substitute the words present in the documents with their first synonym present in the dictionary. The goal of this feature is to reduce the number of words used for processing, by eliminating words that are synonyms of each other.

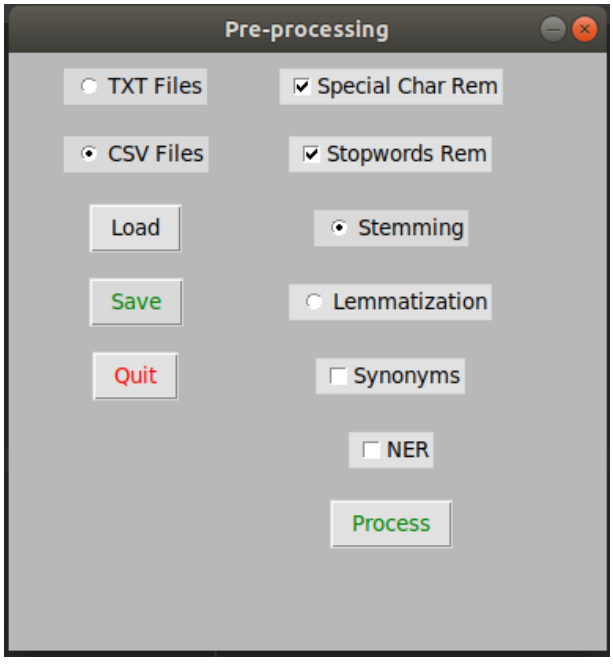

Figure 3.3: Pre-processing menu

3. **Processing Module** - Following the creation of a vocabulary, it is done a calculation of the weight of the words present in the documents. To achieve this, it will be applied the term-frequency-inverse document frequency - tf-idf - that represents the frequency of the inverse-term of the frequency that each word has in the document. It was possible to identify the most common features present in the corpus and also create a dictionary of words based in their frequency. This dataset contains a large number of documents, and as such, it was

important to use a tf-idf vectorizer, that allows for the storage of each set of words from each document in a vector associated with the tf-idf value for each word. This is important to understand the relationship between the frequency of the words present in the dataset and their impact in determining the positive or negative polarity of a document. With this analysis, it is possible to extract the most important features present in the dataset, displaying to the user a matrix with the index, tf-idf vectors and the classification for each document to then be analysed with a feature selection model. With this information, it was possible to generate an attribute relationship file with the represented term weighting of each feature in the documents and their respective positive or negative label, in order to map the causality degree of such features with their document polarity label. The processing view, responsible for these operations is represented in Figure 3.4.

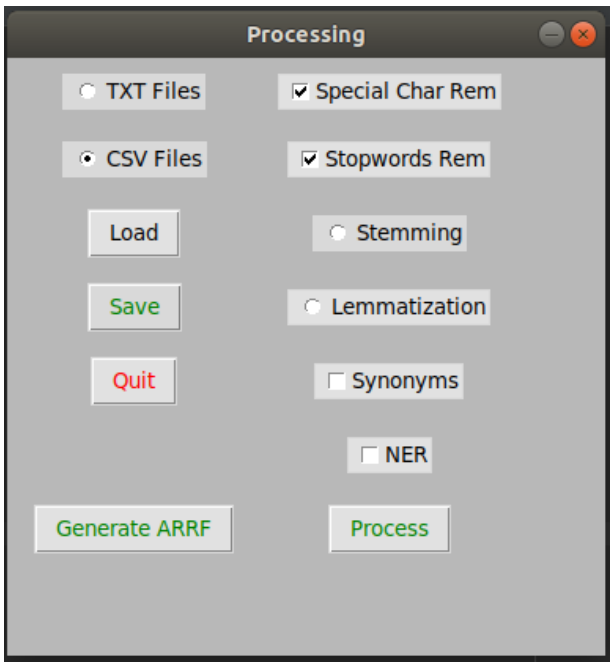

Figure 3.4: Processing menu

4. **Model Building** - After the words in the corpus are weighted and the relationships between the features are identified, the next step is to build a classification model. The user will then be able to export the created dataset in a format suitable for Weka (20). With the generated ARFF file (31), it is then possible to use several different classifiers to the created model. Then, the processed corpus will be split into a training and test set, using the knowledge extracted from the training set to classify the test set. In this case, the model will be subjected to a predictive analysis, using several classification algorithms present in Weka, such as Multinomial Naive Bayes and J48 in order to classify the processed datasets as positive or negative and analyse the accuracy of the classification.

5. **Evaluation -** The final phase is the evaluation of the model created, using metrics such as the accuracy of the classification, F1 score and the confusion matrix. It was made a comparison between the different classification algorithms and the results achieved with each one of them. This stage also includes analyses of different score types, of some characteristics as the number of false positives and negatives. The confusion matrix allows the visualization of the performance of an algorithm, with each row representing the instances in a predicted class and each column representing the instances in an actual class (32). In this way, it is possible to understand the error ratio, between the predicted instances of the class and actual instances that belong to the class. There were evaluated several parameters, based around the accuracy of the classification of the test set. These parameters are the Sensitivity, Specificity, Precision, False Positive Rate, False Negative Rate, Accuracy, F1-Score and also an analysis of the confusion matrix, as a way of understanding the number of false positives and negatives present in the classified dataset.

3.2 Attribute Relationship

In the processing stage, a model is generated and the goal now is now to understand the relationship between the most important features present in the corpus and their weight in influencing a positive or negative label. There is created a dictionary with an unique set of words present in the corpus, associated with an index. From that dictionary, it will be possible of generating a vector of tf-idf weights, for each document present in the corpus, containing the index of the feature and their respective tf-idf score, as well as a positive or negative labelling. Then, for the whole corpus, it will be generated a matrix of these tf-idf vectors, and after that operation it will be generated an ARFF file containing this data. ARFF stands for attribute relationship file and it will map the relationships between the score of each feature present in the corpus and their sentiment tag. Bellow it is presented a document with the tf-idf vectors corresponding to each document in the corpus, as well as their polarity tags. This ARFF file is composed of two parts, the header and the body. The file header is represented in Figure 3.5 and the file body is represented in Figure 3.6.

Where, for each document, it's represented a line, containing the index of the document, ranging from 0 to the number of files contained in the corpus. It is also representing the tf-idf weight of each ones of the words contained in the vocabulary and the sentiment polarity as the final attribute, that represents the class, stating 'pos' for positive and 'neg' for a negative sentiment. After the creation of a vocabulary of words, the matrix composed of the tf-idf values is used to observe the most notorious features present in the corpus and their class, as a preparation for future predictive analysis, using correlation algorithms, such as Information Gain. It's also relevant to state that these values were normalized, ranging from 0 to 1 as their weight value. In the case of these datasets, it was extracted the 5000 most important features, according to their weight score. After this, it's located the file body, following the @data tag, where it will be created a line for each document present in the corpus, with the information stated above. As such, the processed data is

```
@relation polarity
@attribute 10 real
@attribute ability real
@attribute able real
@attribute absolutely real
@attribute across real
@attribute act real
@attribute acting real
@attribute action real
@attribute actor real
@attribute actors real
(...)
@attribute class {pos, neg}
```

Figure 3.5: ARFF Header after applying operations of Special Character and Stopwords Removal.

represented in this form. For this example, the first value refers to the index, 0, that means that this is the first document in the dataset, the following values are the weight of all the features in that document and the last one, the binary class, with pos indicating a positive sentiment. The same rules apply to the last line, the final document, represented with the index, feature weights and a negative sentiment, represented in Figure 3.6.

```
@data
0, 0.6115995809754082, 0.37646205525906042, 0.5605170185988092, 0.5135166556742356,
0.4863232841258714, 0.5074541934925921, 0.7214608098422191, (...), pos
1, 0.7907755278982137, 0.5710530701645918, 0.5961845129926823, 0.5605170185988092,
0.7907755278982137, 0.7907755278982137, 0.7214608098422191, (...), pos
(...)
63744, 0.877080149633748, 0.8808541824320326, 0.8665440980679653,
0.6372906223539109, 0.691334676784403, 0.5278793380032229, (...), neg
```

Figure 3.6: ARFF Body after applying operations of Special Character and Stopwords Removal.

Chapter 4

Case Study and Experiences

For the present study, it was analyzed a corpus of sixty-four thousand movie reviews with a binary tag of zero for negative sentiment and one for positive sentiment. Half of them are classified as negative and the other half is classified as positive, which means the dataset is balanced. This was the dataset chosen for analysis and after the loading of the csv file containing it, it is given the user a possibility of choosing which pre-processing operations he wants to apply. After that, the user can choose a set of pre-processment operations to apply to the corpus. In this case, the results were achieved with the processing of the classified polarity dataset.

4.1 Dataset Characterization

The dataset is composed of sixty-five thousand document reviews, represented in a csv file, like it can be seen bellow. The name of this dataset is Sentiment Polarity Dataset Version 2.0 and it can be found in Kaggle's website (33). This dataset is stored in a database, using MySql. In this database, is is stored two different dataset attributes, the words of the original documents and their classification - zero or one, representing one for positive and zero for negative sentiment. Then, the dataset is imported to the database and the corpus is ready to be pre-processed in the main program module. This dataset is represented in Figure 4.1.

4.2 Dataset Processing

For the purpose of the present thesis, it was applied three different sets of operations for the pre-processing of the datasets - Special Character Removal, Stopwords Removal and Stemming. These operations are processed, it is displayed to the user the original text and text after the operations are applied. It will be also possible to visualize the percentage of text that was reduced after the pre-processing. Then it is created a vocabulary with all the terms present in the dataset and there are created tf-idf vectors, with the index of each term, per document, and their respective tf-idf values. By this way, it is possible to map each document feature with their weight relative to the whole corpus. After this, the file is ready to be

	fold_id	cv_tag	html_id	sent_id	text	tag
	0 – 9	1000 unique values	42 – 29.9k	0 – 111	63652 unique values	neg, pos, neg
28	0	cv001	18431	2	did anybody know this film existed a week before it opened ?	pos
29	0	cv001	18431	3	the plot is deceptively simple .	pos
30	0	cv001	18431	4	george washington carver high school is having student elections .	pos
31	0	cv001	18431	5	tracy flick (reese witherspoon) is an over-achiever with her hand raised at nearly every question , way , way , high .	pos
32	0	cv001	18431	6	mr . " m " (matthew broderick) , sick of the megalomaniac student , encourages paul , a popular-but-slow jock to run .	pos

Figure 4.1: CSV dataset

exported to an ARFF file, in order to complete the classification with the selected algorithms.

In Table 4.1 it can be seen the result of the pre-processment operation in dataset, with the operations of Special Character Removal, Stopwords Removal and Stemming, with the 'Original Corpus' column displaying the original text documents, one line per document, in the 'class' column is displayed 'pos' or 'neg' for a positive or negative sentiment and in the 'Processed Corpus' is displayed the result, after these operations were processed. In Table 4.3 it can be seen the TF-IDF Polarity matrix, where for each document is stored the index, the tf-idf weight for each of the features and in the last column is stored the sentiment polarity. If a number is zero, in a position (x,y) it means that feature x is not present in document y. This matrix is the essential base for the extraction of features and the creation of the ARFF file with the processed data. This attribute relationship file will then be imported into Weka, where it can be used different algorithms for model classification. In this case, it was chosen Multinomial Naive Bayes, Support Vector Machines, J48 and K-Nearest Neighbor as the algorithms for the classification process.

Dataset Pre-processing			
Index	Original Corpus	Processed Corpus	Class
0	films adapted from comic books have had plenty of success , whether they're about superheroes	film adapt comic book plenti success whether superhero	pos
1	for starters , it was created by alan moore (and eddie campbell) , who brought the medium	creat alan moor eddi campbel brought medium	pos
2	to say moore and campbell thoroughly researched the subject of jack the ripper would be like	say moor campbel thoroughli research subject jack ripper would like	pos
3	the book (or " graphic novel , " if you will) is over 500 pages long and	book graphic novel 500 page long	pos
4	in other words , don't dismiss this film because of its source.	word dismiss film sourc	pos
(...)	(...)	(...)	(...)
32356	blaring on the soundtrack , the brotherly duo of doug and steve butabi (chris kattan and will	blare soundtrack brotherli duo doug steve butabi chri kattan	neg
32357	there is one crucial difference , however– these guys speak	one crucial differ howev guy speak	neg
32358	that little fact has been used as justification for the film's existence , that the butabis' new	littl fact use justif film exist butabi newfound capac speech would open whole new	neg
32359	the doors opened by director john fortenberry and screenwriters steve koren , ferrell , and kattan	door open director john fortenberri screenwrit steve koren ferrel kattan	neg

Table 4.1: Example of Pre-processing a dataset with 32359 features, after applying Special Character Removal, Stopwords Removal and Stemming operations.

4.2.1 Model Evaluation

After the classification of the processed corpus, the model is evaluated in terms of accuracy, using the parameters described in the previous chapter. The dataset was divided 66-34, into two subsets of training(66%) and the remainder testing data, suggested by Weka. With the classification algorithms, it was possible to study the percentage of accuracy of this predictive analysis, with different scores criteria. These values values are displayed to the user in the console for visualization. It was possible to import the generated ARFF file to Weka for further classification and visualization of the created model, with the classification made by several different algorithms available in Weka.

Then, it will be done an analysis on eight processed datasets - D1, D2, D3, D4, D1', D2', D3' and D4', where the first four elements of the list represent datasets where it were ap-

Vocabulary	
Index	Features
10	0
90	1
abil	2
abl	3
absolut	4
accept	5
achiev	6
(...)	(...)
written	4993
wrong	4994
ye	4995
year	4996
yet	4997
york	4998
young	4999

Table 4.2: Vocabulary of 5 000 features.

TF-IDF Matrix											
Index / Features	10	abl	absolut	act	action	actor	stress	actual	add	(...)	Class
0	0.000	0.0000	0.0000	0.7634	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	pos
1	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	pos
2	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.8098	0.0000	(...)	pos
3	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	pos
4	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	pos
(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)	(...)
32355	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	neg
32356	0.8973	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	(...)	neg
32357	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	neg
32358	0.000	0.5179	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.8396	(...)	neg
32359	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	(...)	neg

Table 4.3: TF-IDF-Matrix of 32360 documents and 5 000 features.

plied data processing operations and the last four elements represent a different version of the previous datasets, where the same operations where executed for feature selection and after, Information Gain algorithms where applied. D1 serves as the baseline, where all the operations where applied - Special Character Removal, Stopwords Removal, Stemming, Synonym Analysis and Name Entity Recognition. In D2, it was applied all the operations except Synonym Analysis, in D3 were applied all the options except Name Entity Recog-

nition and in D4 it was applied all of the operations except Synonym Analysis and Name Entity Recognition. In D1', D2', D3' and D4' it was applied the same operations as in the previous datasets, respectively, with an additional algorithm of Information Gain for feature selection. It was used a Ranker Search Method, that evaluates each attribute and lists the results in a rank order, using the full training set with 5000 features. Bellow, it is represented an example of the final classification scores achieved with applying the Multinomial Naive Bayes and Support Vector Machines algorithms in dataset D4. In the conclusions, it will be made a comparison of the accuracy scores achieved with each of the algorithms.

Summary	
Correctly Classified Instances	7316 - 67.03 %
Incorrectly Classified Instances	3686 - 32.966 %
Kappa Statistic	0.340
Mean Absolute Error	0.329
Root Mean Squared Error	0.574
Relative Absolute Error	65.932 %
Root Relative Squared Error	114.831 %
Total Number of Instances	1100

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.655	0.314	0.678	0.655	0.667	0.341	0.670	0.618	pos
0.686	0.345	0.663	0.686	0.674	0.341	0.670	0.611	neg
0.670	0.329	0.671	0.670	0.670	0.341	0.670	0.614	avg

Confusion Matrix		
a	b	<— classified as
3625	1908	a = pos
1719	3750	b = neg

Table 4.4: Support Vector Machines accuracy scores in D4.

Summary	
Correctly Classified Instances	7588 - 68.969 %
Incorrectly Classified Instances	3414 - 31.030 %
Kappa Statistic	0.379
Mean Absolute Error	0.372
Root Mean Squared Error	0.44
Relative Absolute Error	74.47 %
Root Relative Squared Error	89.790 %
Total Number of Instances	1100

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.667	0.287	0.702	0.667	0.684	0.380	0.761	0.763	pos
0.713	0.333	0.679	0.713	0.696	0.380	0.761	0.748	neg
0.690	0.310	0.690	0.690	0.690	0.380	0.761	0.756	avg

Confusion Matrix		
a	b	<— classified as
3688	1845	a = pos
1569	3900	b = neg

Table 4.5: Multinomial Naive Bayes accuracy scores in D4.

4.3 Experiences and Results

In this section it will be done an in-depth analysis about the conclusions obtained from the processing of the eight datasets - D1, D2, D3, D4, D1', D2', D3' and D4', with the accuracy being the main indicator of the success obtained in the classification.

- **D1 -** Dataset after applying special character removal, stopwords removal, stemming, synonym analysis and name entity recognition.
- **D2 -** Dataset after applying stopwords removal, stemming and synonym analysis.
- **D3 -** Dataset after applying special character removal, stopwords removal, stemming and synonym analysis.
- **D4 -** Dataset after applying special character removal, stemming and synonym analysis.
- **D1' -** Dataset after applying special character removal, stopwords removal, stemming, synonym analysis and name entity recognition operations and applying Information Gain.
- **D2' -** Dataset after applying special character removal, stemming and synonym analysis operations applying attribute selection with Information Gain.
- **D3' -** Dataset after applying special character removal, stopwords removal, stemming, synonym analysis and applying attribute selection with Information Gain.
- **D4' -** Dataset after applying special character removal, stemming and synonym analysis and applying attribute selection with Information Gain.

In Table 4.7 it can be seen the percentage of word reduction achieved with each of the dataset processing. word reduction is calculated using the percentage of the number of words reduced when applied each one of the operations. The highest reduction was obtained in D3, with 53.79% word reduction.

Word Reduction (%)				
Datasets/ Operations	D1	D2	D3	D4
Special Char Removal	19.08	N/A	19.08	19.08
Stopwords Removal	34.42	43.0	34.42	N/A
Stemming	0.19	0.18	0.19	0.25
Synonyms Substitution	0.11	0.11	0.12	0.1
NER	1.71	N/A	N/A	N/A
Total	54.52	43.29	53.79	19.33

Table 4.6: word Reduction in the 4 datasets.

After the pre-processing operations are applied, it is displayed the percentage of word reduction after each one of the functions and it is created a dictionary, with 5000 features extracted

from each of the datasets. The features are selected accordingly to their tf-idf weight, from highest to lowest, targeting the most important ones and allowing reduction of unimportant features present in the corpus. After this, it is constructed the TF-IDF matrix, with all the term weights represented for 5000 features in 32360 documents. This matrix will then be used to generate an ARFF file, with the goal of extracting the relationships between the feature weight and their polarity class. Then, the datasets are ready for classification and each one of them will be classified with a training set of 66% and a testing set of 34%, with the aim of predicting the classes of the testing set, based in the previous derived knowledge gained from the staining set. It is also applied a feature selection algorithm known as Information Gain in datasets D1', D2', D3' and D4'. This algorithm works by evaluating the worth of an attribute, by measuring the information gain of each class, respectively. Despite the different accuracy scores obtained with the datasets, there is almost no deviation from the score in the processed and the derived dataset, due to the Information Gain algorithm using the full training set for attribute selection. The datasets are classified using Multinomial Naive-Bayes, K-Nearest Neighbor, Decision Trees and Support Vector Machines algorithms, with different accuracy scores achieved in each one of them. These scores can be consulted in Table 4.7.

Accuracy Scores (%)								
Datasets/ Classifiers	D1	D2	D3	D4	D1'	D2'	D3'	D4'
Multinomial Naive-Bayes	68.41	68.38	68.28	68.97	68.41	68.38	68.27	68.97
Decision Tree	52.95	53.04	52.99	53.24	52.94	53.04	52.99	53.24
K-NN	54.25	54.27	54.17	54.33	54.26	54.27	54.17	54.32
SVM	66.43	66.21	66.50	67.03	66.43	66.21	66.50	67.03

Table 4.7: Classification algorithms Accuracy scores.

4.4 Results Analysis

There will be used three metrics to do an evaluation on the created classification models - Accuracy, F1 Score and the Confusion Matrix. From the accuracy perspective the algorithm to achieve the highest was Multinomial Naive Bayes, with a classification score of 68.97 %, obtained in D4. The F1-Score is a score that relates the Precision and Recall, being obtained from the Confusion Matrix. It's a formula that relates the Precision with the Recall. The Precision is a measure of accuracy, provided that a class label has already been predicted. It is a score that represents from of all the classes, how many of them were correctly predicted. The Recall score or Sensitivity is the true positive rate that calculates how often it predicts positive. The highest one was achieved by Multinomial Naive Bayes, with 0.69 Precision score obtained in D4. The

Confusion Matrix is a matrix that combines True positives - shows that a model correctly predicted Positive cases as Positive, False positive - shows that a model incorrectly predicted Negative cases as Positive, False Negative - shows that an incorrectly model predicted Positive cases as Negative and True Negative - shows that a model correctly predicted Negative cases as Positive. The highest True Positive Rate and lowest False Positive Rate was also achieved with Multinomial Naive Bayes, with a 0.667 and 0.287 rate, respectively. All of these formulas can be consulted in Table 2.1. With this, it can be concluded that Multinomial Naive Bayes was the most accurate and precise algorithm in the classification of the datasets, followed by Support Vector Machines. This data can be consulted in the following tables:

Summary	
Correctly Classified Instances	7527 - 68.418 %
Incorrectly Classified Instances	3475 - 31.585 %
Kappa Statistic	0.368
Mean Absolute Error	0.376
Root Mean Squared Error	0.450
Relative Absolute Error	75.242 %
Root Relative Squared Error	90.083 %
Total Number of Instances	11002

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.666	0.298	0.298	0.541	0.666	0.680	0.369	0.757	pos
0.702	0.459	0.334	0.675	0.702	0.089	0.369	0.757	neg
0.684	0.460	0.316	0.685	0.684	0.084	0.369	0.757	avg

Confusion Matrix		
a	b	<— classified as
3686	1847	a = pos
4386	3841	b = neg

Table 4.8: Multinomial Naive Bayes accuracy scores in D1.

Summary	
Correctly Classified Instances	7523 - 68.378 %
Incorrectly Classified Instances	3679 - 33.621 %
Kappa Statistic	0.367
Mean Absolute Error	0.377
Root Mean Squared Error	0.451
Relative Absolute Error	75.393 %
Root Relative Squared Error	90.283 %
Total Number of Instances	1102

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,665	0,297	0,694	0,665	0,679	0,368	0,755	0,759	pos
0,703	0,335	0,675	0,703	0,688	0,368	0,755	0,742	neg
0,684	0,316	0,684	0,684	0,684	0,368	0,755	0,751	avg

Confusion Matrix		
a	b	<— classified as
3680	1853	a = pos
1626	3843	b = neg

Table 4.9: Multinomial Naive Bayes accuracy scores in D2.

Summary	
Correctly Classified Instances	7512 - 68.227 %
Incorrectly Classified Instances	3490 - 31.721 %
Kappa Statistic	0.365
Mean Absolute Error	0.376
Root Mean Squared Error	0.450
Relative Absolute Error	75.334 %
Root Relative Squared Error	90.110 %
Total Number of Instances	11002

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.665	0.299	0.692	0.665	0.678	0.366	0.756	0.761	pos
0.701	0.335	0.674	0.701	0.687	0.366	0.756	0.743	neg
0.683	0.317	0.683	0.683	0.683	0.366	0.756	0.752	avg

Confusion Matrix		
a	b	<— classified as
3680	1853	a = pos
1637	3832	b = neg

Table 4.10: Multinomial Naive Bayes accuracy scores in D3.

Summary	
Correctly Classified Instances	7588 - 68.969 %
Incorrectly Classified Instances	3414 - 31.031 %
Kappa Statistic	0.379
Mean Absolute Error	0.372
Root Mean Squared Error	0.449
Relative Absolute Error	74.471 %
Root Relative Squared Error	89.7903 %
Total Number of Instances	11002

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.667	0.287	0.702	0.667	0.684	0.380	0.761	0.763	pos
0.713	0.333	0.679	0.713	0.696	0.380	0.761	0.748	neg
0.690	0.310	0.690	0.690	0.690	0.380	0.761	0.756	avg

Confusion Matrix		
a	b	<— classified as
3688	1845	a = pos
1569	3900	b = neg

Table 4.11: Multinomial Naive Bayes accuracy scores in D4.

Summary	
Correctly Classified Instances	7316 - 66.433 %
Incorrectly Classified Instances	3686 - 33.566 %
Kappa Statistic	0.328
Mean Absolute Error	0.335
Root Mean Squared Error	0.579
Relative Absolute Error	67.132 %
Root Relative Squared Error	115.870 %
Total Number of Instances	11002

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.646	0.317	0.673	0.646	0.659	0.329	0.664	0.613	pos
0.683	0.354	0.656	0.683	0.669	0.329	0.664	0.606	neg
0.664	0.335	0.665	0.664	0.664	0.329	0.664	0.609	avg

Confusion Matrix		
a	b	<— classified as
3573	1960	a = pos
1733	3736	b = neg

Table 4.12: Support Vector Machines accuracy scores in D1.

Summary	
Correctly Classified Instances	7316 - 66.206 %
Incorrectly Classified Instances	3686 - 33.793 %
Kappa Statistic	0.324
Mean Absolute Error	0.338
Root Mean Squared Error	0.581
Relative Absolute Error	67.586 %
Root Relative Squared Error	116.262 %
Total Number of Instances	11002

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.647	0.322	0.670	0.647	0.658	0.324	0.662	0.611	pos
0.678	0.353	0.655	0.678	0.666	0.324	0.662	0.604	neg
0.662	0.338	0.662	0.662	0.662	0.324	0.662	0.607	avg

Confusion Matrix		
a	b	<— classified as
3578	1955	a = pos
1763	3706	b = neg

Table 4.13: Support Vector Machines accuracy scores in D2.

Summary	
Correctly Classified Instances	7316 - 67.497 %
Incorrectly Classified Instances	3686 - 32.503 %
Kappa Statistic	0.330
Mean Absolute Error	0.335
Root Mean Squared Error	0.578
Relative Absolute Error	67.005 %
Root Relative Squared Error	114.761 %
Total Number of Instances	1102

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,648	0,318	0,674	0,648	0,660	0,330	0,665	0,613	pos
0,682	0,352	0,657	0,682	0,669	0,330	0,665	0,606	neg
0,335	0,665	0,665	0,665	0,330	0,665	0,610	0,665 avg	

Confusion Matrix		
a	b	<— classified as
3625	1908	a = pos
1719	3750	b = neg

Table 4.14: Support Vector Machines accuracy scores in D3.

Summary	
Correctly Classified Instances	7316 - 67.03 %
Incorrectly Classified Instances	3686 - 32.966 %
Kappa Statistic	0.340
Mean Absolute Error	0.329
Root Mean Squared Error	0.574
Relative Absolute Error	65.932 %
Root Relative Squared Error	114.831 %
Total Number of Instances	1100

Detailed Accuracy By Class								
TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.655	0.314	0.678	0.655	0.667	0.341	0.670	0.618	pos
0.686	0.345	0.663	0.686	0.674	0.341	0.670	0.611	neg
0.670	0.329	0.671	0.670	0.670	0.341	0.670	0.614	avg

Confusion Matrix		
a	b	<— classified as
3625	1908	a = pos
1719	3750	b = neg

Table 4.15: Support Vector Machines accuracy scores in D4.

Chapter 5

Conclusions and Future Work

It was successfully created a text-mining toolbox for document classification.

5.1 Conclusions

It was created a platform capable of processing a text corpus, from the implementation of modules responsible for the loading of the files, to pre-processing, classifying, processing, predicting and evaluating modules. It was implemented an interactive application, a toolbox for text classification where the user can load a sentiment analysis dataset, a corpus of documents with polarity-based classification used for testing this tool, being able to choose from a different set of operations to be applied to the corpus. With the pre-processing module, for the natural language processing and the processing module, for the creation of vocabularies, term weighting and classification algorithms. It can also be distinguished it's compatibility with Weka and the ability of visualizing different information about the processed dataset, and further model classification. It is given the user the ability of retrieving knowledge from text data as well as analyzing the accuracy of the created models.

5.2 Future Work

For the future, it could be an interesting hypothesis of experimenting with other classified datasets, extending to other fields of sentiment analysis, such as entity subjectivity. In fact, this could be integrated with this polarity analysis, to extract relationships between not only the features and their polarity but also to the subjectivity involved in the statements. Overall it was a very interesting experience, to combine sentiment polarity analysis with data mining.

Bibliography

[1] Pavani Kuntala and Gary L Drescher. Data mining model building using attribute importance. May 15 2007. US Patent 7,219,099.

[2] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[3] Jeonghee Yi Tetsuya Nasukawa. Sentiment analysis - capturing favorability using natural language processing. 2003.

[4] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[5] Ronaldo Goldschmidt and Emmanuel Passos. *Data mining: um guia prático*. Gulf Professional Publishing, 2005.

[6] Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*. Springer, 2015.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[8] Zhenyu Pan, Jie Xue, Yang Gao, Honghao Wang, and Guanling Chen. Revealing the relations between learning behaviors and examination scores via a prediction system. In *Proceedings of the 2018 2Nd International Conference on Computer Science and Artificial Intelligence*, CSAI '18, pages 414–419, New York, NY, USA, 2018. ACM.

[9] Davide Magatti, Fabio Stella, and Marco Faini. A software system for topic extraction and document classification. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '09, pages 283–286, Washington, DC, USA, 2009. IEEE Computer Society.

[10] Ruben Sipos, Dmitriy Fradkin, Fabian Moerchen, and Zhuang Wang. Log-based predictive maintenance. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1867–1876, New York, NY, USA, 2014. ACM.

[11] Kamal Nigam Andrew McCallum. A comparison of event models for naive bayes text classication. 1998.

[12] L. Breiman. Machine learning. 2001.

[13] Zhongheng Zhang. Introduction to machine learning: k-nearest neighbors. June 2016.

[14] Suthaharan S. Introduction to machine learning: k-nearest neighbors. June 2016.

[15] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

[16] John Bauer Christopher D. Manning, Mihai Surdeanu. The stanford corenlp natural language processing toolkit. 2014.

[17] John Bauer Christopher D. Manning, Mihai Surdeanu. Term-weighting approaches in automatic text retrieval.

[18] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. 2008.

[19] Colleen McCue. *Data mining and predictive analysis: Intelligence gathering and crime analysis*. Butterworth-Heinemann, 2014.

[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[21] Jianqiang Li, Yu Zhao, and Bo Liu. Fully automatic text categorization by exploiting wordnet. In *Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, AIRS '09, pages 1–12, Berlin, Heidelberg, 2009. Springer-Verlag.

[22] Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd, 2018.

[23] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[24] Vijay Kotu and Bala Deshpande. *Predictive analytics and data mining: concepts and practice with rapidminer*. Morgan Kaufmann, 2014.

[25] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

[26] John W Shipman. Tkinter 8.4 reference: a gui for python. *New Mexico Tech Computer Center*, 2013.

[27] Michael Widenius, David Axmark, and Kaj Arno. *MySQL reference manual: documentation from the source*. " O'Reilly Media, Inc.", 2002.

[28] Janez Demšar, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, Anže Starič, et al. Orange: data mining toolbox in python. *The Journal of Machine Learning Research*, 14(1):2349–2353, 2013.

[29] Yong Shib Emma Haddi, Xiaohui Liu. The role of text pre-processing in sentiment analysis. 2013.

[30] Martins Bruno Won Miguel, Murrieta-Flores Patricia. Ensemble named entity recognition (ner): Evaluating ner tools in the identification of place names in historical corpora. 2018.

[31] Tina R Patil, SS Sherekar, et al. Performance analysis of naive bayes and j48 classification algorithm for data classification. *International journal of computer science and applications*, 6(2):256–261, 2013.

[32] I Rajeshwari and K Shyamala. Study on performance of classification algorithms based on the sample size for crop prediction. In *International Conference On Computational Vision and Bio Inspired Computing*, pages 1051–1058. Springer, 2019.

[33] Michael Speriosu Michael Speriosu. Twitter polarity classification with label propagation over lexical links and the follower graph. 2011.