Spam detection using text classification techniques

ABSTRACT

This project compares different machine learning text classification techniques in detecting spam text. Spam Text Message is used as the dataset (1). Techniques which are introduced are: DistilBert, Support Vector Machine, K-nearest Neighbours and Multinomial Naive Bayes Classifier. Among all the algorithms, the DistilBert model gets the highest f1 score.

1. https://www.kaggle.com/datasets/team-ai/spam-text-message-classification

1 Introduction

Today, spamming messages is one of the biggest issues faced by everyone in the 21st century. In such a world, internet messages is mostly shared by everyone to share the information and files because of their easy way of communication and for their low cost. But such messages are mostly affecting the professionals as well as individuals by the way of sending spam messages. Every day, the rate of spam messages is increasing. Such spam messages are mostly sent by people to earn income or for any advertisement for their benefit. This increasing amount of spam messages causes traffic congestion and waste of time for those who are receiving that spam. The real cost of spam message is very much higher than one can imagine. Sometimes, the spam also have some links which have malware. And also, some people will get irritated once they see their inbox which is having more spam mails. Sometimes, the users easily get trapped into financial fraud actions, by seeing spam such as job alert spam and offer spam messages. It may also cause the person to have some mental stress. To reduce all these risks, there are NLP techniques which will detect spam mail and non-spam emails. Spam messages can be filtered either manually or automatically. Nowadays, it is obvious that manual text classification alone can’t meet the needs.

Based on the background, automatic text classification has emerged. It can help people summary the text accurately and quickly from the mass of text information. Automatic classification has drawn more and more attention in recent few years no matter in the academic or in the industry area, and it is a topic worth discussing.

Text classification is the process of assigning labels to text according to its content, it is one of the fundamental tasks in Natural Language Processing (NLP). NLP methods change the human language to numeral vectors for machine to calculate, with these word embeddings, researchers can do different tasks such as sentiment analysis, machine translation and natural language inference.

Chapter 2 provides basic information about Natural Language Processing and Text Classification. Chapter 3 provides the theoretical basis for all the ML techniques used for spam detection in this project. Chapter 4 describes used dataset, the implementation of the algorithms and results of the tests. Chapter 5 contains a summary of the results and conclusions from the research.

2.1 NATURAL LANGUAGE PROCESSING

Nowadays, thanks to advances in natural language processing, computers are able to understand and process language. NLP is a branch of science that straddles computer science, artificial intelligence, and linguistics. NLP uses techniques for automatically understanding, analyzing, and generating language by computer. By "natural language" we mean here the language used in everyday communication by humans, as opposed to artificial creations such as programming languages. NLP covers a broad spectrum of issues, ranging from the mundane, such as counting the number of occurrences of words in a text or automatic word splitting, to innovative solutions, such as automatic answers.

The Internet search engine answers questions automatically and automatically translates speech into other languages in real time. Analysis can involve both text and speech, but the object of processing is usually text. Although mastery of natural language is a fundamental and one of the first skills that humans learn, teaching this skill to a computer is not an easy task. The challenges that researchers face stem from the highly imprecise and ambiguous nature of language. It can be observed on syntax, semantics and pragmatics levels.

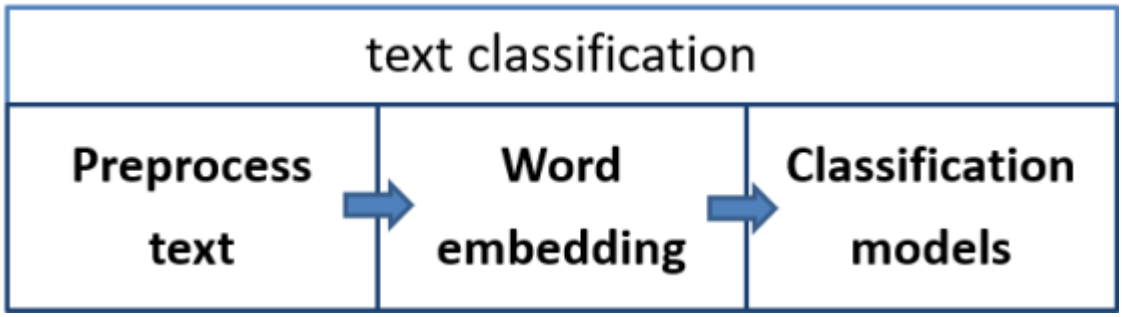
Often, in order to successfully solve more complex natural language processing tasks, it is necessary to perform less complex steps beforehand. For example, in text classification, usually the input text is segmented to extract the individual words in the text. Then, each word is subjected to stemming. This way, the same word in two grammatical forms can be treated as one, what effectively reduces the size of the problem. Often performed is also tagging of parts of speech and filtering out uninteresting groups of words, such as conjunctions, numerals, pronouns and prepositions. Finally, the resulting set of words

is fed to the input of the classifier, which determines which group the text belongs to.

2.2 TEXT CLASSIFICATION

Classification refers to the process of dividing objects with the same attributes into the same category. From the grammatical level, the text is a written form of expression consisting of words, phrases, sentences and paragraphs. Text classification is a supervised machine learning method, in which all text categories are defined in advance. In text classification, sometimes the text not just belong to one class, for example, if a sports superstar married to a singer, this news can be belonged to both classes of sports and entertainment, this

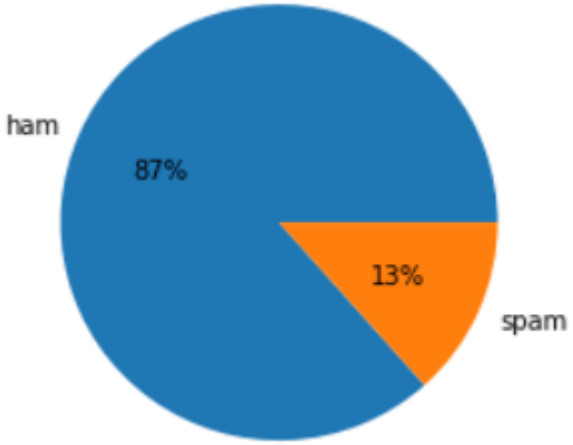
situation is called ’multilabels’, which will not be discussed here. In this project every message is mapped to just one class. The process of text classification can be summarized as below



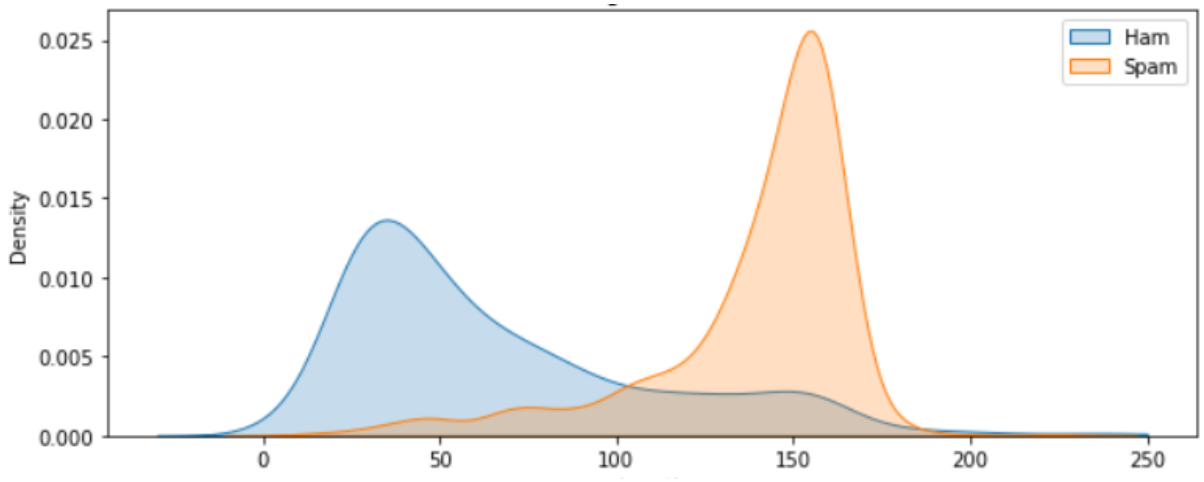
3. Description and preprocessing of the dataset

3.1 Description

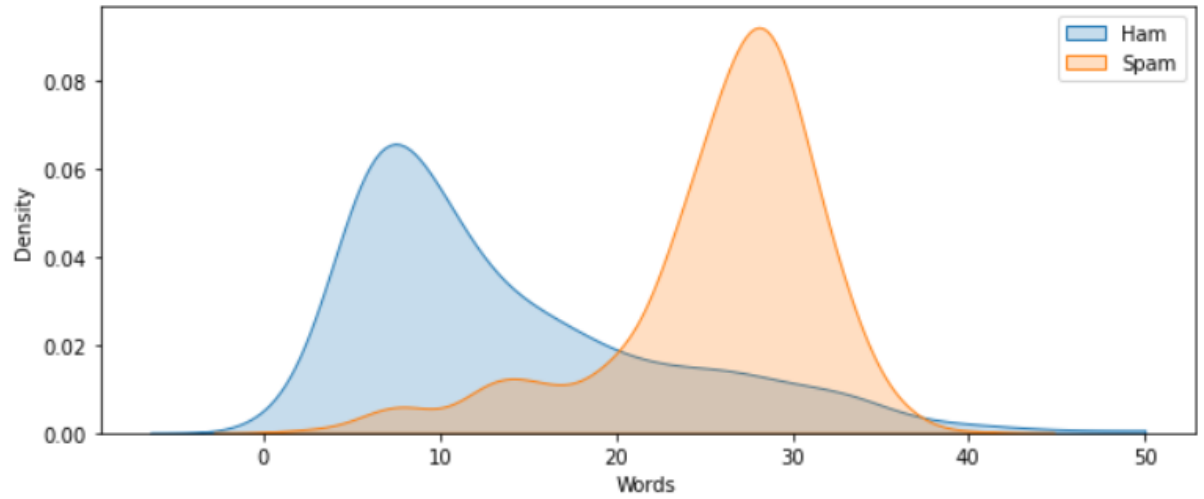
Dataset is of size 5572x2 and contains of „v1” which consists of labels and „v2” column which contains messages. During preprocessing, labels „ham” and „spam” are replaced with 0 and 1 and column „v1” and „v2” are renamed to „is_spam” and „content”. Dataset in only 13% consists of spam data so the data is skewed:



Therefore for evaluation we will use f1 score metric which is good metric for skewed datasets. Length of spam messages are around 150 and not spam messages are usually 25 chars long:



No. of words is also higher for spam messages:



3.2 Feature creation

Feature creation is the creation of the features according to the raw data. It is on the basis of these features that the training of the classification ML models will be done. Created features are: no. of words in message, no. of chars in mess., no. of uppercase chars…, no. of upercase words …, presence of words „free” or „win”, presence of link.

3.3 Preprocessing data

During preprocessing, lemmatization was applied on the used dataset. Then stop words, digits and punctuation was removed from every message.

Since the whole dataset is in one file, dataset was splitted into training and testing data in ratio 80:20.

4.Desription of used methods and their implementation

4.1 DistilBert

First NLP technique used was DistilBert which is a lighter version of Bert model. BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary. As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert_base_uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

Data was tokenized so that batches of sequences could be fed into the model at the same time. Model chosen during tokenization step was „distilbert-bert-uncased". Sentences were padded/truncated to the max. Length of 96 chars. Encodings got batched to a TensorSliceDataset object, so that each key in the batch encoding corresponds to hyperparameters named according to the model trained. Later on model was fine-tuned and trained. Model outputs logits which are converted to probabilities that sentences are spam.

DistilBert model results are: 964 True Negatives, 2 False Positives, 2 False Negatives, 147 True Positives.

This gives Precision of 98,6577% and the same value of recall and f1 score.

4.2 K-Nearest Neighbors

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.
The k value in the k-NN algorithm defines how many neighbors will be checked to determine the classification of a specific query point. For example, if k=1, the instance will be assigned to the same class as its single nearest neighbor. Defining k can be a balancing act as different values can lead to overfitting or underfitting. Lower values of k can have high variance, but low bias, and larger values of k may lead to high bias and lower variance. The choice of k will largely depend on the input data as data with more outliers or noise will likely perform better with higher values of k. Overall, it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

Grid search cross-validation techniques were used to determine the hyperparameters of model and train model on them. Using the confusion matrix, measures of the quality of the classification system was given: 964 True Negatives, 2 False Positive, 46 False Negatives, 103 True Positives.

This gives Precision of 98,0995%, Recall of 60,403% and f1 score of 74,7685%

4.3 Multinomial Naive Bayes Classifier

Bayes rule is used to calculate the posterior probability using prior probability which might be related. There are two events, event A and event B. The conditional probability (also known as a posterior probability) of event A under the condition that event B has occurred refers to the probability of event A occurring under the condition that event B occurs. The conditional probability can be expressed as P(A | B),P (B) is called a prior probability, P(A, B) is the joint probability of A and B.

p(A|B) = P(A, B) P(B) = P(A)P(B)/P(B)

if P(A | B) is equal to P(A),which meas the posterior probability of event A has nothing to do with event B, we can say event A and B are independent, otherwise, they are dependent.

As the features of dataset have discrete frequency counts, Multinomial type of Naive Bayes model was used. As before grid search techniques using cross-validation were used to determine the optimal value of the alfa hyperparameter of model and the model was trained on this hyperparameter. Thanks to confusion matrix we know that there are: 959 True Negatives, 7 False Positives, 14 False Negatives, 135 True Positives.

This gives Precision of 99,3237, Recall of 87,248% and f1 score of 92,857%

4.4 Support Vector Machine

Supportive vector machines(SVM) is a classifier defined by a separating hyperplane. It is a supervised learning model, given labeled training data, the algorithm outputs an optimal hyperplane which can maximize the margin between 2 classes. SVM has its unique advantages in solving the problem of high-dimensional space vector and it is also memory efficient. The system uses a hyperplane that has been found through training and learning to classify the sample space into two categories. When the problem to be solved is

linearly separable, the optimal hyperplane requires the maximum of the optimal hyperplane based on the correct classification. Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of elements. In this thesis, Multisvm is trained with one-against-all approach. One-against-all approach builds as many binary classifiers as there are classes, each trained to separate one class from the rest. To predict a new instance, multisvm iterate each classifier until the first classifier which assigns the new instance as its class member is found.

As before grid search techniques using cross-validation were used to determine the hyper-parameters of our model and train this model on them. Using the confusion matrix, measures of the quality of the classification system are given: 964 True Negatives, 2 False Positives, 13 False Negatives,136 True Positives. This gives Precision of 97.794, Recall of 89,262 and F1 score of 93,3334%.

5. Results and conclusion

5.1 Results

If we summarize the obtained results, here is what we get:

DistilBert f1 score: 98,6577%

SVM f1 score: 93,3334%.

Multinomial Naive Bayes f1 score: 92,857%

KNN f1 score 74, 7685%

We can see that DistilBERT is the ML classification algorithm that provides the best results. However, based on the scores, we can see that there is no significant difference in precision and accuracy between these algorithms apart from KNN.

5.2 Conclusion

Project started by loading a dataset of spam messages and created features on the raw data using Feature Engineering. Once features were created, program analyzed the data made available on the basis of these features, before being able to do data preprocessing which consisted in removing the presence of stop words, punctuation, digits and lemmatize the words. In addition, different Machine Learning classification algorithms were fine-tuned. To do this, it was useful for fine-tuning some of these algorithms to use search grid techniques using cross-validation to evaluate the performance of the model. DistilBERT and other transformer models are to be preferred when we would like to push performance to its maximum and to optimize the avoidance of True Positive misclassification (given by the recall score). However, if these few percent more can be neglected, classical classification algorithms such as Multinomial Naive Bayes and SVM can still be preferred because of their simplicity of understanding and implementation.