Project: Spam and Ham mail classification Maximilian Kroslak

• **Abstract**

Spam mails are an everyday life issue. In 2024, more than 45% of all mails worldwide were identified as spam. Detecting spam efficiently is essential for email filtering systems. The goal of this project is to classify email as “spam” or “ham” using machine learning algorithms.

For this task we used Logistic regression model and Support vector machines (SVM). We used two methods of text vectorization: Bag of words and TF-IDF (term frequency-inverse document frequency).

• **Introduction**

Spam email classification is a important task for email filtering systems to reduce unwanted emails. The goal of this project is to build models that can accurately differentiate between spam mails and non spam mails.

For this project we used this dataset: https://www.kaggle.com/datasets/manoj223/email-spam-csv?select=spam.csv. Number of positive examples was 87% and negative only 13%. I was afraid that the number of negative examples was too low, which would lead to underfitting, but the end result was quite satisfactory.

This dataset contains two columns: Category (identifies mail as spam or ham) and Message (contains content of the email). Spam.csv (name of our dataset) contains 5572 different emails, which should be enough for training our models. We split the data randomly using train_test_split method from sklearn to 0.8 training and 0.2 test.

• **Context**

Spam classification has been widely studied in machine learning. Popular approaches include: Naïve Bayes Classifier: Known for its simplicity and efficiency in text classification. Logistic Regression: A linear model that performs well with balanced datasets. Support Vector Machines (SVM): Effective in handling high-dimensional data like text. Deep Learning: Models such as LSTMs and transformers (e.g., BERT) have shown state-of-the-art performance but require significant computational resources.

• **Description and justification of methods used**

Since the task of this project was to differentiate emails between spam and ham, we choose classification models: Logistic regression and SVM.

First we loaded the data from spam.csv file, then separated the text of the mail from mail category (spam or ham). Next we changed the label of spam and ham to 1 and 0 (spam = 1, ham = 0). Then we change the messages to lowercase and removed the stop words (words with no “value” in this problem). For stop words we used this dataset: https://www.kaggle.com/datasets/amirhoseinsedaghati/english-stopwords. Next we vectorised our text. We used bag of words method and TF-IDF method and fitted them into our models to train. Finaly we tested our models on test data.

• **Brief description of technical issues that you have encountered**

Even though the Spam classification is widely studied in machine learning, the datasets for this task that I found were really disappointing. The number of mails was low or some of the columns were missing description. Some datasets were really chaotic with too many features. The dataset I used in this project was the best I could find, even though the number of emails was lower than wanted.

The only technical issues I had were probably only with pandas, which could not read some of the csv files because of special encoding. To resolve this issue I used "chardet" import, which should have find out which encoding was used. Unfortunately, even when I specified in pandas the encoding used in the csv file, pandas returned the same error, that the csv uses different encoding. Then I found dataset that panda could read and I could continue my work. The problem happened next day, when pandas could not read the csv file from yesterday. Then I tried loading the csv that did not work before (the one from Introduction) and now for some unknow reason it did. This was kind of weird.

Another technical issue could be the imbalance in the dataset used. The dataset used contained significantly more ham emails than spam, potentially leading to more biased models.

• **Experimental evaluation** (tables, graphs, their interpretation and what conclusions do you draw from them)

For this problem I wanted to compare Logistic regression and SVM model. For text vectorisation we used bag of words method and TF-IDF (term frequency-inverse document frequency) and compare, which is better suited for which model.

Logistic regression results:

Method	Training accuracy	Test accuracy
Bag of words	99.4%	97.3%
TF-IDF	96%	86.1%

SVM results:

Method	Training accuracy	Test accuracy
Bag of words	99.5%	97.4%
TF_IDF	99.7%	97.7%

(These results were based on 0.8 0.2 training/ test split)

Bag of words (Bow) performed well with both Logistic regression and SVM. However, SVM achieved slightly higher accuracy in both training and testing.

TF-IDF showed better results with SVM by 10% higher accuracy. TF-IDF reduces the influence of frequently used words and rises the importance of words that are rarer. As we can see from the table, this seems to be a good option to use in SVM models, but bad at Logistic regression.

Later I tried splitting the data into 0.7 training and 0.3 testing, but the results were slightly worse. Another experiment was to not remove stop words. With the stop words the results were almost the same. The training accuracy was slightly higher, but the test accuracy was slightly worse, but not much. Since the dataset is so much imbalanced, the TF-IDF method can assign too much importance to rarer words, which can result in Logistic regression effectiveness.

- **Conclusion**

I think the project was successful, but the result was quite surprising. I thought I would get maximum of 93% accuracy, but I got 97%+. This could be because of the imbalance between spam an ham count in the dataset, since the number of ham emails was almost 7 times more.

- **References**

git hub: https://github.com/Maax02/Spam-mail-classification/tree/main

spam dataset: https://www.kaggle.com/datasets/manoj223/email-spam-csv?select=spam.csv

stop words dataset: https://www.kaggle.com/datasets/amirhoseinsedaghati/english-stopwords