Email Spam Classification
Final Project

A.Y. 2022-2023

Foundations of Data Science

Aur Marina Iuliana, 1809715

Ilaria Gagliardi, 1796812

Sophia Balestrucci, 1713638

Viktoriia Vlasenko, 2088928

Michele Musacchio, 2070948

Contents

1 Abstract **1**

2 Introduction **1**

3 Dataset and Benchmark **1**

4 Proposed method explained **1**

4.1 Preprocessing 1

4.2 Model building 2

4.3 Performance evaluation 2

5 Experimental results **2**

6 Conclusions and Future work **2**

6.1 Future work 3

1 Abstract

With this project, our intent is to study different classifiers for the email spam problem. First we preprocessed the dataset in order to select the features that best optimized the learning process of our classifiers, based on basic natural language processing techniques. Then we implemented two classifiers: Logistic Regression and Naive Bayes, two of the models seen in class, and choose others classifiers to compare: Ada Boost, Extra Trees and KNN. Lastly we computed some metrics to evaluate performances of these classifiers on our dataset, to define which has the best results.

2 Introduction

The email spam classification is a well known problem and a very important one for every day life. Just think that since 2003, spam constitutes 80% to 85% of email messages sent worldwide. So it's interesting study how it works and what are the best methods to exploit in order to obtain the best results for the classification. We decided to observe different models on the same dataset, to compare performances and define which one has the best score, but also to study weaknesses and strenghts of those classifiers.

The structure of the notebook is the following:
- Preprocessing and standardization (by Balestrucci, Musacchio, Vlasenko)
- Metrics implementation (by all team memebers)
- Model building and performance evaluation (by Gagliardi, Aur)

3 Dataset and Benchmark

We used a dataset taken from Kaggle , where we have 5172 rows (the email samples) and 3000 features, that are the most common words in all the emails; for each row it's stored the count of occurences of each word in each email.

Material used:
- course material, like assignment and slides
- python libraries, like Scikit-learn, an open-source library for predictive data analysis, spaCy and ntlk.

4 Proposed method explained

4.1 Preprocessing

For the preprocessing part, we filtered the features used some basic natural language processing techniques, like removing the stop words, that are the most common words which don't add much information (like articles, prepositions, etc), or lemmatization, that consists of grouping inflected forms under a single base form of the words.

This should improve the overall performance of every classifier, since it reduces the number of feature with no or little information.

4.2 Model building

Then we choose five models to be trained with this modified dataset. Two of them, Logistic Regression and Naive Bayes are implemented in order to understand in details the behaviour of such classifiers.

Instead, three of them are taken from scikit-learn library: Ada Boost, Extra Trees and K Nearest Neighbors. For each of the latter ones, we computed the GridSearchCV in order to find the optimal values of parameters, according to a score based on accuracy.

4.3 Performance evaluation

In the end, we computed different metrics to evaluate classifiers performances. In particular we considered:

- Confusion matrices, accuracy, precision and recall
- Computation time
- Equal error rate
- ROC curve
- Cross validation

5 Experimental results

Overall, the metrics agree on defining Extra Trees as the best classifier for our dataset, while KNN results to be the worst.

The reason for Extra Trees to be the best is probably on the fact that it chooses features randomly while computing the decision trees, which reflects a primary aspect of this kind of problem and dataset, that is the order of the words is not relevant.

While K Nearest Neighbors searches for the most similar emails and probably, with the given dataset that doesn't have the structure of the text but is composed only of occurences of words, this approach leads to be wrong often.

6 Conclusions and Future work

For the preprocessing part, we tried different techniques and we observed that not all of them was useful to increment performances of the classifiers, on the contrary reducing too much leads to a loss of information, so it needs to find the right balance in filtering features.

For the modeling part, we understood that the results of classification task is not given only by the model used but also by the dataset and the kind of problem. So in order to achieve the best numbers, it needs to study the problem in details, find the basic charatteristics and only then choose a model that best suites those.

6.1 Future work

To improve this work, we can try different feature selection. For example, we can try to introduce a study over the most common words that appear in spam emails and attach to every of them a bigger weight.

References

[1] “Email Classification Research Trends: Review and Open Issues”, Mujtaba, Ghulam and Shuib, Liyana and Raj, Ram and Majeed, Nahdia and Al-Garadi, Mohammed, 2017

[2] “Comparative Analysis of Classification Algorithms for Email Spam Detection”, Shafi’i Muhammad Abdulhamid, Maryam Shuaib, Oluwafemi Osho, Idris Ismaila and John K. Alhassan, 2018