**Project – Python for Data Analysis**

*Yvann Vincent*

*Antoine Serafini*

This summary will guide you through our work on the Spambase Dataset (https:// https://archive.ics.uci.edu/ml/datasets/Spambase), a dataset grouping together 57 variables to predict whether an email is a spam or not.

**First, we want to tell you that our API is mainly usable on the website developed for the project:** [**https://spamails.herokuapp.com/**](https://spamails.herokuapp.com/)

Our work has been divided on 3 parts:

- the data analysis of our dataset

- the supervised learning to predict if an email is a spam or not

- the conception of a Spam Email Prediction (API and Website)

Then we will give our conclusions on this work.

• Data Analysis

The data analysis part was our first comprehension with the dataset. Due to multiple plots to better understand it, we tried to do differents correlations ranking to get the most interesting variables for the supervised learning part to get the best accuracy on our predictions. To see different correlations, we used several tools like pivot\_table or PCA.

• Supervised Learning

On this part, we tried to predict if an email is a spam or not due to the different variables in the dataset.

To get the best prediction, we had to estimate three parameters:

- what feature to use

- what model to use

- how to tune its parameters to fit the dataset

With our work about correlations in the Data Analysis part, we tried to reduce the number of variables used for training the model with 4 different dataframes (all the variables, the 17th best, the 22nd best, the 10 best and worse). After testing, it has been found that the dataframe with all the variables gave us the best accuracy.

To find the best model, we also tried by trial and error several models like SVC, Trees and Random Forests, Linear Classification (LDA, Naive Bayes), Logistic Regression or Deep Learning. The one maximizing the accuracy on its prediction was the K-Nearest-Neighbours (KNN). Then we had to tune KNN’s parameters to fit the dataset. On this part, we used plot to see the evolution of accuracy according to the parameters to verify we weren’t overfitting.

After determining everything to maximize the accuracy of our model, we were able to predict near 93% of our test set.

• API and Website

The last step to completion was to develop and API to make our prediction model accessible to everyone. We maid it into a website called SPAMAILS, you can check it out here: <https://spamails.herokuapp.com/>

For this, we have writer an API to parse data of a text input and turn it into the required 57 parameters, so we could fit into our prediction model. This prediction model allowed us to make several conclusions about the Spam dataset.

• Conclusion

After having done Data Analysis and Supervised Learning on the Spam dataset, we really enjoyed working on a database relevant to everyday life. By working on this project, we attained a better understanding of spam structures. It served as a great example of how to use AI knowledge for practical applications.

However, our IA showed that the database doesn’t have the same accuracy with real email that on itself. It is important to keep in mind that the Spam dataset has been assembled in 1998. Spam emails back in the day were different from today’s standard. Communication in the early 2000s was not as spontaneous as it is today. Messages were longer and denser in content due to slow network making extremely spontaneous communication unrealistic.

The average number of capital letters in spam emails from 1998 is already a lot longer than modern spam emails in whole. Since modern spams are notably shorter, it makes a frequency based approach like in the dataset inaccurate and overly sensitive to small variations.

As a final conclusion, a frequency-based approach is not sufficient for the more common short texts from today. For instance, in a text of 10 words, having the word ‘you’ appear once gives a frequency of 10%, enough to automatically classify the email as a spam.

To correctly train a prediction model common shorts texts, both word frequency and word count are necessary. FYI, a word frequency of 50% but a count of 1 would mean the text is 2 words long. A simple message like “Thank you” would therefore not be classified as a spam with correct model training.