PYTHON FOR DATA ANALYSIS

Spambase dataset

TABLE OF CONTENTS

EXPLORING THE DATASET

- First, we started to read the dataset just to see and get familiar with the elements inside.
- For this, we decided to use many functions like :
    - spam.infos() → to see the types of variables
    - spam.isnull().sum() → to check if there is any missing values
    - spam.shape () → to obtain the number of rows and columns
    - spam,columns → to obtain the name of columns in order to visualize all the variables that we may need in the future

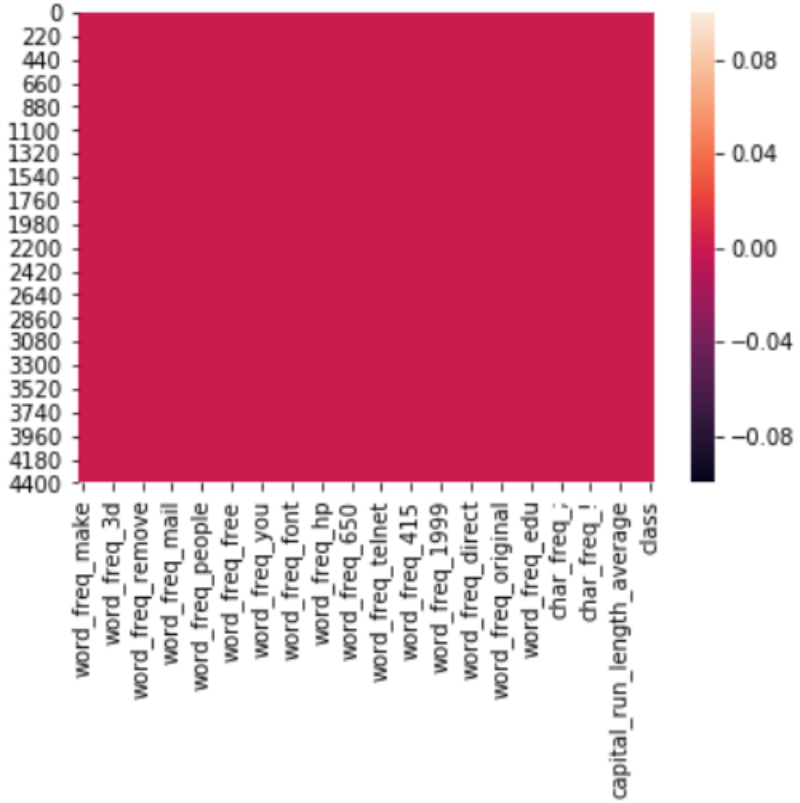
- After visualizing the dataset, we wanted to know the number of spam/not spam.

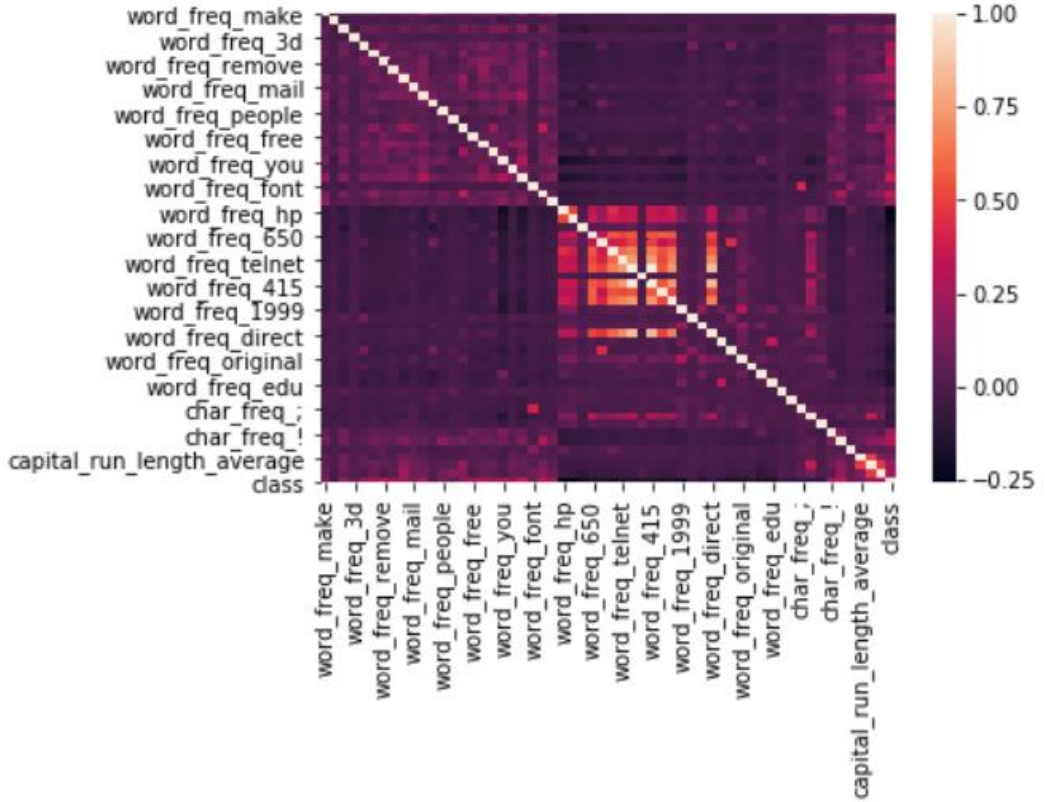
  → spam['class'].value_counts() which has shown us :

```
Out[17]: 0    2788
         1    1813
         Name: class, dtype: int64
```

    - 0 means the email is not a spam
    - 1 means the email is a spam

- To represent the distribution spam/not spam, we called the matplot library to build an histogram.





We created a matrix to show the missing values and a correlation using the library seaborn

CLEANING THE DATASET

- After discussion , we have decided to drop the features which have correlation less than <0.2. We now only have 20 variables left :

```
Out[163]: Index(['word_freq_our', 'word_freq_over', 'word_freq_remove',
                 'word_freq_internet', 'word_freq_order', 'word_freq_receive',
                 'word_freq_free', 'word_freq_business', 'word_freq_email',
                 'word_freq_you', 'word_freq_your', 'word_freq_000', 'word_freq_money',
                 'word_freq_hp', 'word_freq_hpl', 'char_freq_!', 'char_freq_$',
                 'capital_run_length_longest', 'capital_run_length_total', 'class'],
                dtype='object')
```

- We will test our future models on both data set and based on the accuracy score we will decide which dataset is better for prediction :
  - The one with all the features
  - The other one with only 20 features

BUILDING THE MODEL

- We have built different models using different libraries such as sckiti from Python :
  - Logistic Regression → using sklearn.linear_model
  - Random Forest → using sklearn.ensemble
  - Boost → using xgboost
  - Naives Bayes → using sklearn.naive_bayes

- We also built a KNN algorithm which consists to have a cluster.

- After spliting the data into train_set and test_test.
- We trained our models on the train_set and then we did a prediction on the test_set.
- We got different accuracies score for different models and dataset :
  - The model with the highest accuracy score is the Boosting one.
  - Also, we tried this model on both dataset : the one with all the variables and the one with only 20 variables

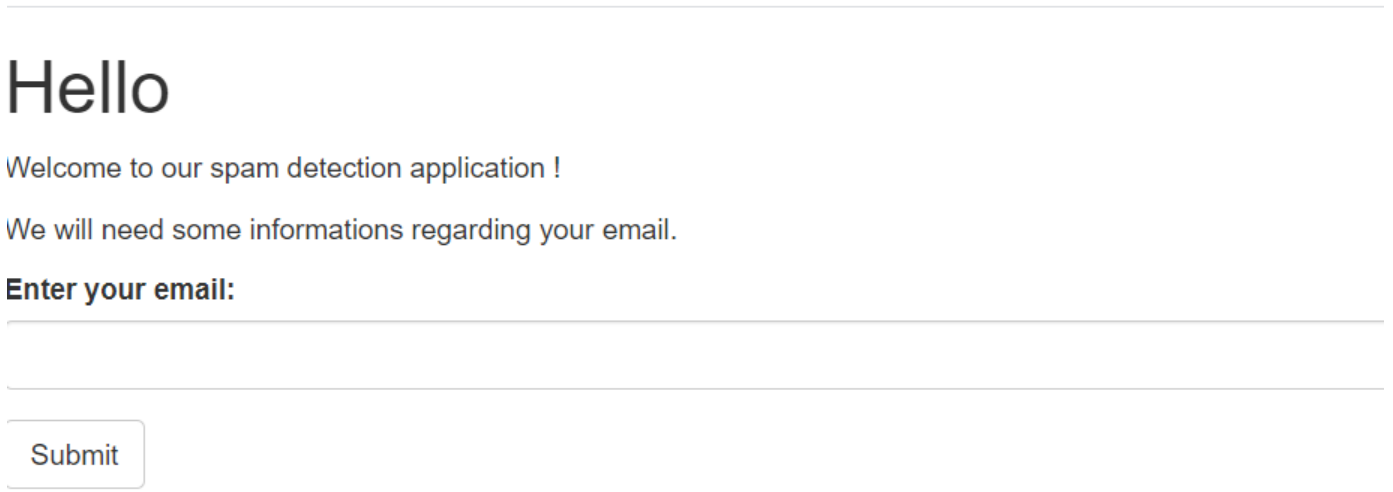
- Conclusion : The highest accuracy score is obtained using all the variables and by the boosting model.

API FLASK

- We decided to go with the Flask method to do our API. For that we have followed the same method as our teacher explained during the class.
- We have chosen the software Visual Studio Code in which we have app.py where we imported different libraries from flask such as :
  - Bootstrap
  - FlaskForm
  - StringField,FloatField etc….
- We have 3 different files :
  - App.py where we have created the form in which the person will enter his/her email and submit
  - Prediction_form.html and base.html → we created these two files during our last session in class with our teacher.

Mon Application Flask

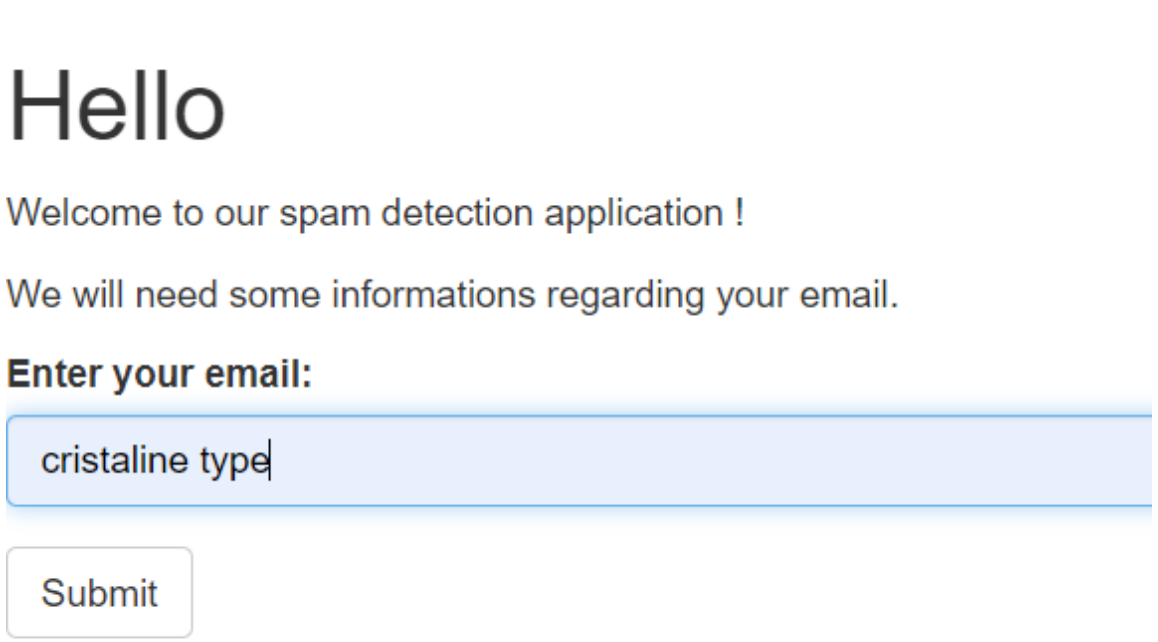
This is how our form looks like

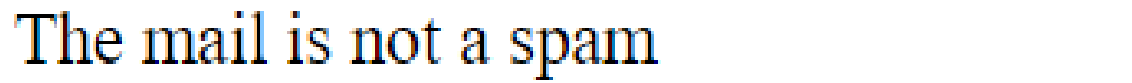


API FLASK

- Now that we have our application, the task here was to predict if the mail entered is a spam or not.
- In order to do this, we have created different functions :
  - Def comptage (mail) → this function counts the percentage of each feature present in the mail (which we just entered).
  - Def fonctionfinal (mail) → remodel the function comptage and return a dataframe in which the frequency of each feature is clearly seen.

- After obtaining the frequency of each feature, we define the function def prediction() where will predict if the email is a spam or not based on our built model previously.
- In our case the boosting model has the highest accuracy score, so we decided to use this one and predict the given mail on this model.





The display message looks like this stating if the email is a spam or not