**CRUZ, JUAN CARLOS M.**

**FINAL CAPSTONE PROJECT**

**IE 198**

Raw data: SPAM Dataset

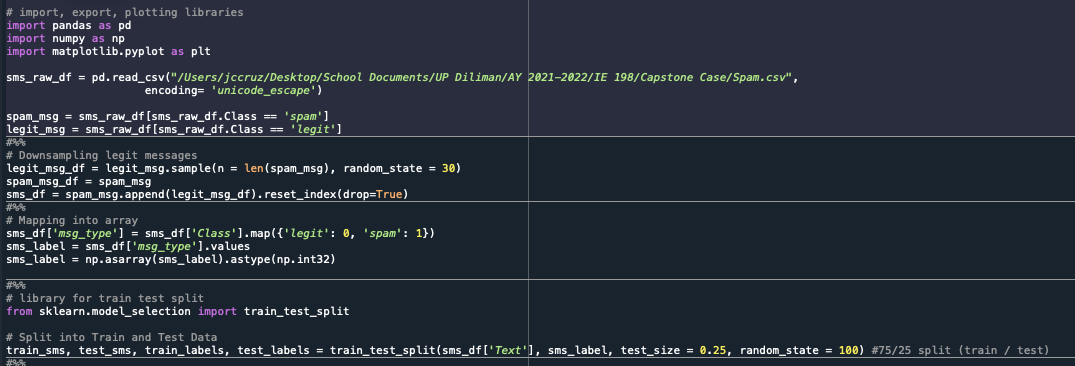
• 5,572 messages

* 4,825 legitimate messages (86.6%)
* 747 spam messages (13.41%)

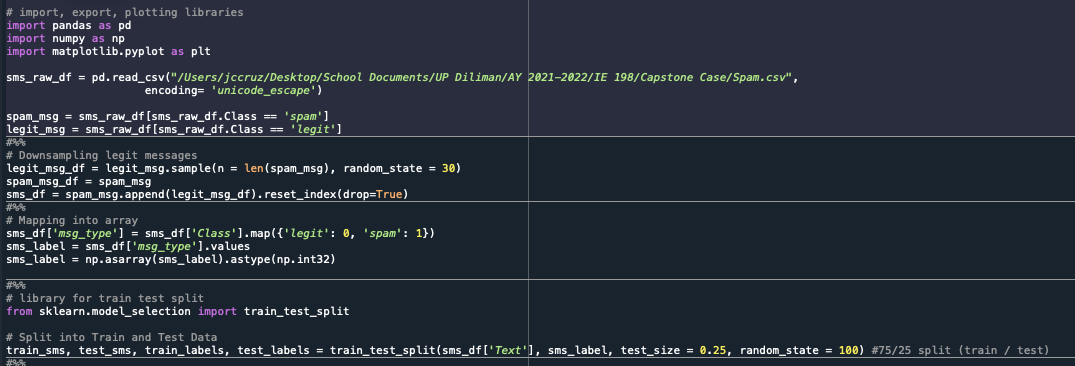
**Task**: Create a predictive model to predict Spam and Legit messages.

**STEPS TO CREATING THIS PREDICTIVE MODEL**

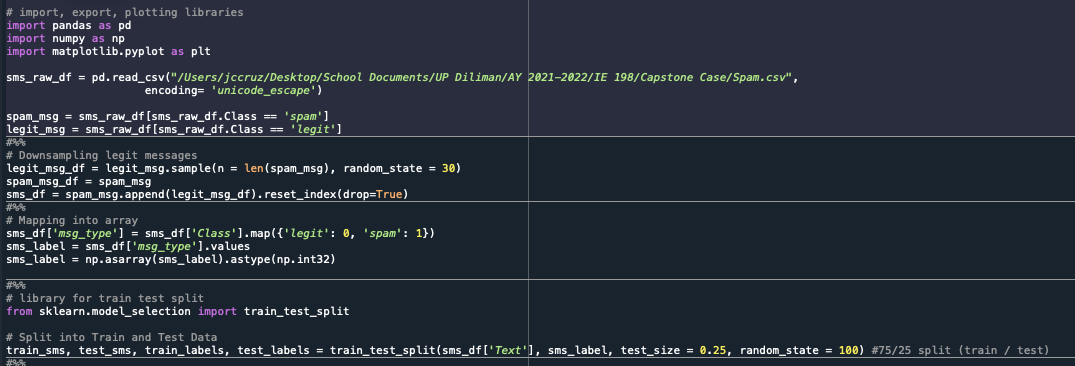
1. **Pre-processing**
   1. Data was imported into Spyder



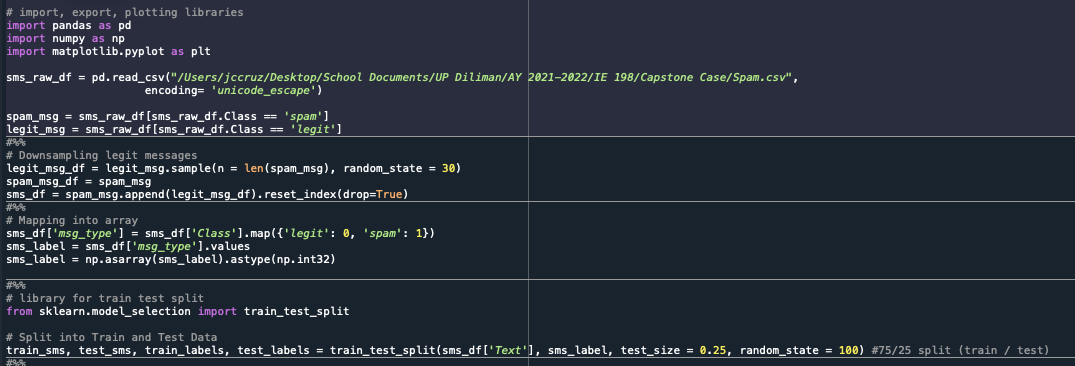
* 1. Since the data contains an imbalanced proportion of legitimate and spam messages, the first step is to **downsample the data** so that the number of legit messages and spam messages to be fed into the model are equal. (n = 747)



* 1. After this data was downsampled, the predictor variable “Class” was **mapped into another column** to express its numeric equivalent (for use when running, evaluating and assessing the model’s accuracy) [0 = Legit, 1 = Spam]

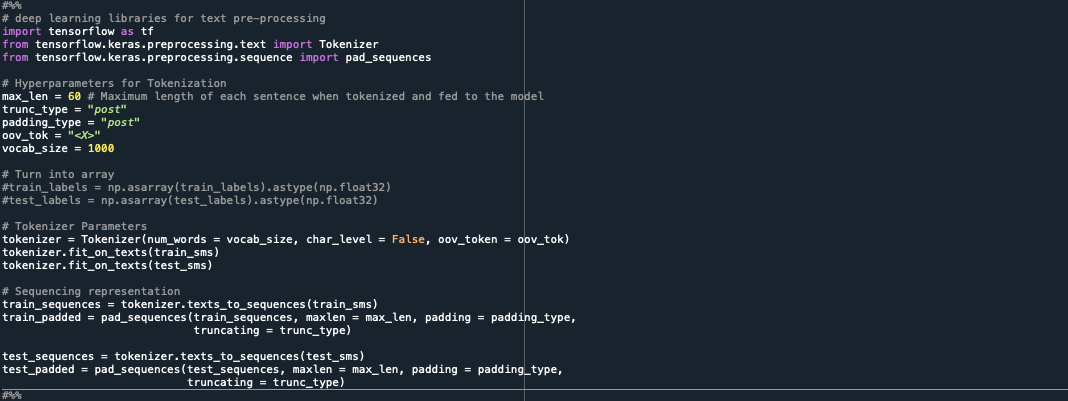


* 1. The predictor and class variables were split into test and training data. (25% test / 75% training)



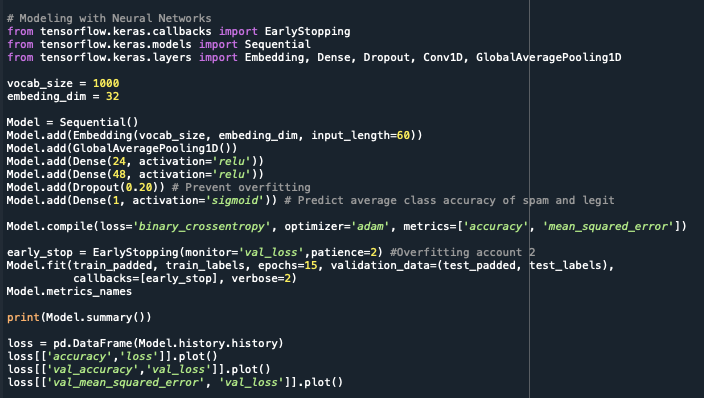
* 1. The “Text” predictor data was fed through a “Tokenizer” module which assigned a corresponding numerical value to every unique word found in the dataset. After both the testing and training data were tokenized, each instance of data was fed through the “texts\_to\_sequences” and “pad\_sequences” function to represent the “Text” data as an array of numbers, with each subarray representing a sentence in the preprocessed dataset (each subarray is formed by padding each formed sequence according on the maximum length parameter [see: max\_len variable which is 60 numbers (words)]).

The Tokenizer in question employed a vocabulary size of 1000, meaning that 1000 unique words were identified from the train and test data and used as the numerical reference when retrieving the numerical array equivalent of said data.

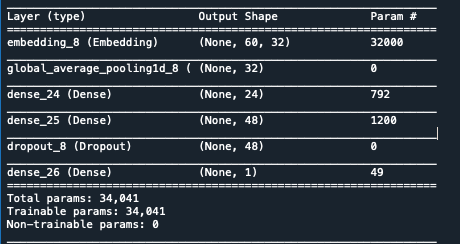


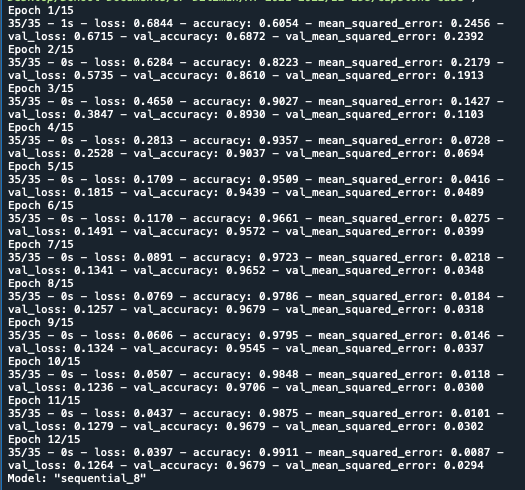
* 1. The data is now fully preprocessed. It is ready to be fed into the Model.

1. **Running the Model**



* 1. Details of the Model
     1. Artificial Neural Network with **five** (5) hidden layers and **one** (1) output layer
        1. An Embedding layer with input\_dim = 1000, output\_dim = 32, and input\_length = 60 (input\_length corresponding to the length of each subarray from the preprocessed data)
        2. A Global Average Pooling 1D layer
        3. Two density layers with 24 and 48 nodes respectively, using the ReLu activation function
        4. A Dropout layer which removes 20% of the data to prevent overfitting
        5. An output layer with a sigmoid function
     2. Compiled with the “binary\_crossentropy” loss function, uses the “Adam” optimizer and identifies accuracy and MSE metrics
     3. EarlyStopping function which checks the val\_loss at a patience value of 2
     4. Model is tested for 15 epochs
  2. Results and Values





**MODEL STOPS AT 12/15 EPOCHS**

Accuracy of Model: 99.11%

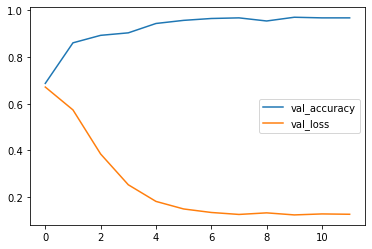
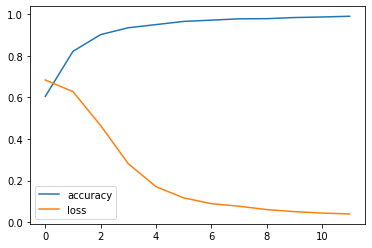
Value of Accuracy: 96.79%

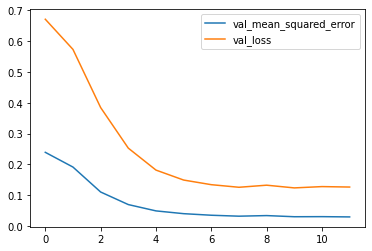
Loss value: 0.0397

MSE: 0.0087

Value of MSE: 0.0294

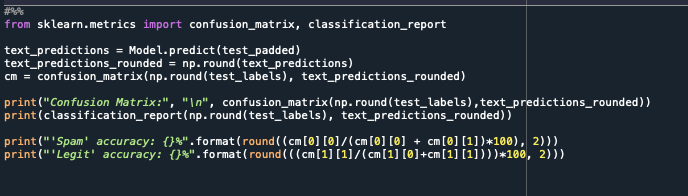
**GRAPHS**



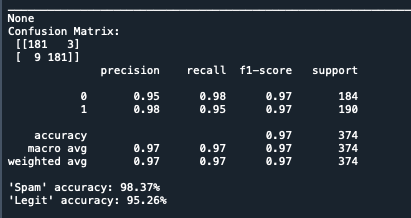


**CONFUSION MATRIX, CLASSIFICATION REPORT AND AVERAGE CLASS ACCURACY OF PREDICTIONS**

**Code**



**RESULTS**



**INTERPRETATION**

The model has 98.37% accuracy of correctly predicting a message as spam, while it has a 95.26% accuracy of correctly predicting a message as legitimate. The recall values of each legit and spam prediction are 98% and 95%, respectively, while both have an F1-score of 97%.