**Spam/Ham Text Detection using Dense network, LSTM and**

**Bi-LSTM architectures in TensorFlow2**

**Purpose**

To implement Dense, Long Short Term Memory (LSTM) and Bidirectional-LSTM (Bi-LSTM) deep learning models in TensorFlow2 Keras API for spam/ham texts Detection.

**Approach**

* Dataset Loading and Exploration.
* Preparing Train and Test Data
* Model training using 3 of the above mentioned deep learning models.
* Choosing the final model with the best Accuracy .
* Using Final trained classifier to classify the new messages.

**Spam Data Exploration**

Libraries for

* Reading data,exploring and plotting.
* Train test split
* Text preprocessing of Deep Learning
* Modelling

Text Dataset is a tab separated (\t) text file.

**Statistics summary**

* 5,572 labels and messages
* two unique labels indicating for “ham” and “spam”
* Lesser unique messages (5,169) than total message count(5,572) indicating some repeated messages.
* duplicates = df[df.duplicated()],shows there are 403 duplicated messages.
* The top label and top message are “ham” and “Sorry, I’ll call later” respectively.
* Data is imbalanced as the number of ham is 4,825 compared to 747 spam messages.
* Popular ham sms = “Sorry, I’ll call later” and Popular spam sms = “Please call our customer service…”
* On average, the ham message has length of 73 words whereas spam message has 138.

**WordCloud and Bar charts Visualisation**

* Create a separate data frame for ham and spam texts and convert it to a numpy array to generate WordCloud.
* Extract words most commonly found in ham and spam messages, remove meaningless stop words such as “the”, “a” , “is” etc, and plot it.
* According to the Bar plot ,there are more frequent ham messages (85%) than spam (15%).

**Ways to handle imbalanced data**

* Using appropriate evaluation metrics .
* Resampling the Dataset (oversampling/upsampling or undersampling/downsampling)
* Putting different resampled Datasets together.

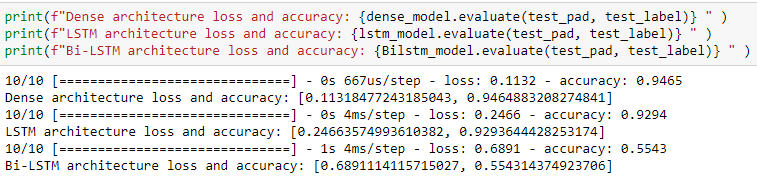
**Downsampling the Majority class(Ham)**

After Downsampling , 747 messages in each class.

**Prepare train/test data and pre-process text**

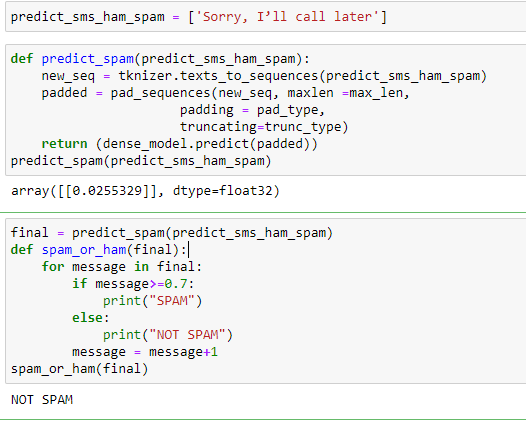
* 80% of data were used for training and 20% for testing purposes.
* Convert labels to numpy arrays to fit deep learning models.
* Text pre-processing which includes Tokenization, Sequencing and Padding.

**Compare three different models and select a final one**

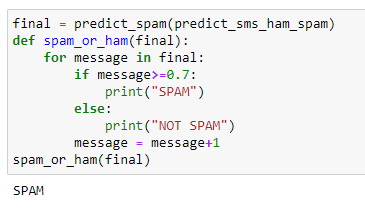
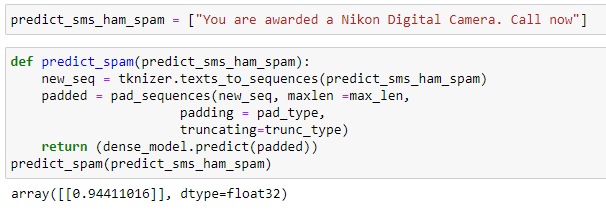
****

The dense spam detection model outperformed other two models in terms of Accuracy .Hence,this will be trained and evaluated.

**Case 1: Given the text from our original data(Raw text from given dataset)**



**Case 2: Given the New data**

****

Dense spam model correctly classifies the first case as Ham and second case as Spam .

**Improvement**

Trying

* more sampling approaches like upsampling ,SMOTE etc
* using different hyper-parameters.
* sample size increment.
* Machine Learning Classifiers