# **CIS 612 Big Data & Para Database**

**Project Report**

**Twitter Message data transformation: Sentiment Analysis**



**Name: Sri Harish Reddy Alla**

**CSUID: 2824635**

# **Project Title:** Twitter Message data transformation: Sentiment Analysis

# **DataSet Preparation:**

Data preparation was done by reading the dataset from the input directory (my google drive). The datasets are csv format. Using pandas python library. We were able to read the dataset in a dataframe format as shown by the output below.

Graphical user interface, table

Description automatically generated

Fig : Dataset dataframe view.

Having two copies of dataset as per from the source of the dataset, training and testing dataset, we read the two dataset and happened to do the merging to have one dataset. This therefore, is expected to be divided and splitted into two during data preprocessing. This will have our predefined training and testing data splits as opposed to the original two sets of data provided from the dataset source.

# **Data Preprocessing:**

This is a crucial part as it helps in enhancing dataset review, processing and cleanup techniques applications to making sure the dataset is ready to be fed to the model. In preprocessed format, the dataset can be precisely understandable by the model.

## **Dropping columns technique:**

This stage started with dropping columns technique. Having dropped several columns, we resolved to have the below dataframe view.

Graphical user interface, table

Description automatically generated with medium confidence

Fig : Dataframe after several columns dropped.

Renaming of remaining columns was hence done. Column 0 was renamed to sentiment whereas column 1 was renamed to tweet.

## **Dataset Splitting:**

This was splitted by the help of the train\_test\_split predefined method from sklearn python library. With a test size of 0.33 dataset samples and a random state threshold of 0 below was the output shape of each split.

Text, letter

Description automatically generated

## **Text Tokenization:**

Since the dataset is a bit big, to save time, we had to use a maximum of 10000 words to tokenize.

Using Tensorflow to tokenize texts is a more precise and concise way of working with tensors. The method is more preferable in this project instead of using NLTK Library as proposed. However, using either should work as expected. Tokenization is therefore done, to have a tokenized text content. This helps in the vectorization process and hence making the dataset ready for modeling stage in a preprocessed format.

With vectorization and indexing checks, the dataset reveals 527562 independent tokens.

The word Index dictionary file is saved to the output folder found in the root folder (working directory).

## **Pad Sequences technique:**

To ensure that all sequences in a list are of the same length, padding of sequences is employed in the preprocessing stage. By default, this facilitates padding 0 to each of the sequences, until every sequence in the list has the same length as the longest sequence.

Graphical user interface, application, Teams

Description automatically generated

# **Modeling:**

## **Model structure:**

Ever thought of how models work? Well, I guess not really. Deep learning models are trained using neural network architecture. To be precise, the models are trained using a set of data which are as well labeled using multiple layers.

Deep Learning models do exceed human level performance. These architectures have the ability to do feature learning from the dataset, therefore, you do not have to necessarily do the features extraction process manually.

The sentiment analysis model was designed to have 8 layers with an output layer with softmax activation.

Below is the screenshot of the model summary. The summary shows the layers added to model, output shape and the parameters which will be used for training, non-training.

Table

Description automatically generated

## **Model Compiling and Training:**

To train the model, compiling the model was the first step to take followed by the training. Without compiling the model, the model is forced to compile by default parameters. The compiling was done under loss function with sparse\_categorical\_crossentopy parameter, metric function with metric index parameter, and SGD as the optimizer parameter.

Below was the summary after training our model on with the dataset splits, and with about at least 10 epochs.

Graphical user interface, text

Description automatically generated with medium confidence

## **Prediction In Action:**

To have a comparison on prediction, prediction was simple and clear on the dataset split. Prediction was done on the training set of the data and the testing set of the data respectively.

**Y Train Prediction test:**

Graphical user interface, text, application, email

Description automatically generated

**Y Test Prediction test:**

Table

Description automatically generated

## **Model Evaluation Metrics Used:**

The model metrics tries to do performance evaluation on a model. It is important to note the model evaluation metrics used in this project. Below screenshots try to give a highlight on that.

Table

Description automatically generated

# **Results in Accuracy and Comparisons**

The output helps by providing a know how the model predicts. With that said, using the model metrics print out, The accuracy was not that bad as the accuracy threshold on testing split set was 0.81 which shows a good starting point of the model performance.

Confusion matrix of the Y test split data was as below:

Diagram

Description automatically generated

**Overall Screenshots of Execution:**

These screenshots will be helpful to know how execution is done through Google colab.

The pre-defined dataset is imported from Google drive to Google colab.

We have to set the path, where Dataset is located in Google drive.

**Step 1: Dataset Preparation:**

Graphical user interface, text, application, email

Description automatically generated

**Step 2:**

Graphical user interface, text, application, email

Description automatically generated

**Step 3:**

Graphical user interface, application, table

Description automatically generated

**Step 4: Data Preprocessing & Feature Engineering**

Graphical user interface, text, application, email

Description automatically generated

**Step 5:**

Graphical user interface, text, application, email

Description automatically generated

**Step 6: Tokenization**

Graphical user interface, text, application, email

Description automatically generated

**Step 7:**

Graphical user interface, text, application

Description automatically generated

**Step 8: Padding Sequences**

Graphical user interface, text, application, email

Description automatically generated

**Step 9: Modelling**

Graphical user interface, text

Description automatically generated

**Step 10:**

Graphical user interface, text, application, email

Description automatically generated

**Step 11: Model Training**

Graphical user interface, text, application, email

Description automatically generated

**Step 12: Prediction**

Graphical user interface, text, application

Description automatically generated

**Step 13: Prediction**

Graphical user interface, text, application, email

Description automatically generated

**Step 14: Performance Metrics Evaluation**

Graphical user interface, application

Description automatically generated

**Step 15: Metrics Evaluation**

Graphical user interface, text, application, email

Description automatically generated

# **Discussion and Conclusions:**

On the look of the accuracy, the model was a bit good but with less performance. To enhance performance, the model might be able to have a high capability compared to the current accuracy.

Having a better dataset preprocessing approach should enhance a better performance on our model. Least but not least, the following should be recommended:

Thresholding, this is another crux of the big data analysis. Threshold might affect the model in one way or another. It is advisable to use a threshold value of 0.5. However, it is again advisable to use the best threshold that works best on different types of models. Threshold simply rules out the projected probability scores into a class label. In case of normalized probabilities, for example in the range of 0 and 1, and no threshold value is chosen, then the threshold value to use is always defaulted to 0.5.

Dataset Resampling, a simple and concise resampling technique method may improve the model in the best way possible. “Generally, resampling techniques for estimating model performance operate similarly: a subset of samples are used to fit a model and the remaining samples are used to estimate the efficacy of the model. This process is repeated multiple times and the results are aggregated and summarized. The differences in techniques usually center around the method in which subsamples are chosen”*.* — Page 69, [Applied Predictive Modeling](https://amzn.to/2Fmrbib), 2013. K-Fold technique should be a recommendation on this sentiment model. However, one might try to enumerate between K-Fold and bootstrap resampling methods as well.

In conclusion, other applicable data preprocessing and modeling techniques are not limited to be used in trying to enhance a better improvement on the model performance. A better implementable approach increases chances of having a better engineered model.

# **References:**

1. [Opinion Mining and Sentiment Analysis](http://www.amazon.com/gp/product/1601981503/ref=as_li_ss_tl?ie=UTF8&camp=1789&creative=390957&creativeASIN=1601981503&linkCode=as2&tag=algoshopa-20) by Bo Pang and Lillian Lee.
2. [Foundations of Statistical Natural Language Processing](http://www.amazon.com/gp/product/0262133601/ref=as_li_ss_tl?ie=UTF8&camp=1789&creative=390957&creativeASIN=0262133601&linkCode=as2&tag=algoshopa-20&l=as2&o=1&a=0262133601) by Christopher Manning and Hinrich Schuetze
3. [Introduction to Information Retrieval](http://www.amazon.com/gp/product/0521865719/ref=as_li_ss_tl?ie=UTF8&camp=1789&creative=390957&creativeASIN=0521865719&linkCode=as2&tag=algoshopa-20&l=as2&o=1&a=0521865719) by Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze
4. [Speech and Language Processing](http://www.amazon.com/gp/product/0131873210/ref=as_li_ss_tl?ie=UTF8&camp=1789&creative=390957&creativeASIN=0131873210&linkCode=as2&tag=algoshopa-20&l=as2&o=1&a=0131873210) by Daniel Jurafsky and James Martin