Twitter Message Data Transformation : SENTIMENT ANALYSIS

By:
Sri Harish Reddy Alla
(2824635)

OUTLINE

- Data Preparation
- Data Preprocessing
  - ❖Dropping columns technique
  - ❖Dataset Splitting
  - ❖Text Tokenization
  - ❖Pad sequence technique
- Modeling
  - ❖Model structure
  - ❖Model Compiling and Training
  - ❖Prediction In Action
  - ❖Model Evaluation Metrics Used
- Results in Accuracy and Comparisons
- Discussion and Conclusions

INTRODUCTION

- Sentiment Analysis is one of the applications of the Natural Language Processing techniques.
- Various NLP techniques are used to analyse social media posts and know what customers think about their products.
- This helps in understanding the issues and problems that their customers are facing by using their products.
- Sentiments is feelings, emotions, opinions likes/dislikes bad/good etc.

INTRO CONTD..

- Sentiments analysis is a task in Natural Language Processing and Information Extraction that looks for writers' feelings reflected in favourable or unfavourable remarks, queries, and requests by looking through a huge number of documents.
- In the study of human behaviour, we take user sentiment and emotion out of plain text to determine their opinions.
- The aim is to determine whether a text's opinion is good or negative.

DATA PREPARATION

- Data preparation was done by reading the dataset from the input directory (my google drive).
- The datasets are in csv format.
- Using pandas python library We were able to read the dataset in a dataframe format as shown by the output in the next slide.
- The source of the dataset have two copies,
- 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 PREPARATION CONTD..

```
[13] df
```

	0	1	2	3	4	5
0	4	3	Mon May 11 03:17:40 UTC 2009	kindle2	tpryan	@stellargirl I loooooooovvvvvveee my Kindle2. ...
1	4	4	Mon May 11 03:18:03 UTC 2009	kindle2	vcu451	Reading my kindle2... Love it... Lee childs i...
2	4	5	Mon May 11 03:18:54 UTC 2009	kindle2	chadfu	Ok, first assesment of the #kindle2 ...it fuck...
3	4	6	Mon May 11 03:19:04 UTC 2009	kindle2	SIX15	@kenburbary You'll love your Kindle2. I've had...
4	4	7	Mon May 11 03:21:41 UTC 2009	kindle2	yamarama	@mikefish Fair enough. But i have the Kindle2...
...	...	...	...	...	...	...
1599995	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...
1599996	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com - Very cool to hear old Walt interv...
1599997	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you ready for your MoJo Makeover? Ask me f...
1599998	4	2193602064	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	tinydiamondz	Happy 38th Birthday to my boo of alll time!!! ...
1599999	4	2193602129	Tue Jun 16 08:40:50 PDT 2009	NO_QUERY	RyanTrevMorris	happy #charitytuesday @theNSPCC @SparksCharity...

1600498 rows × 6 columns

DATA PREPROCESSING

- Data pre-processing is a crucial part as it helps in enhancing dataset review, processing and clean-up.
- The techniques makes sure that the dataset is ready to be fed to the model.
- In pre-processed format, the dataset can be precisely understandable by the model.

DROPPING COLUMNS TECHNIQUE

- This stage started with dropping columns technique.
- Renaming of remaining columns was hence done.
- Column 0 was renamed to sentiment whereas column 1 was renamed to tweet.

```
[19] df
```

	sentiment	tweet
0	4	@stellargirl I loooooooovvvvvveee my Kindle2. ...
1	4	Reading my kindle2... Love it... Lee childs i...
2	4	Ok, first assesment of the #kindle2 ...it fuck...
3	4	@kenburbary You'll love your Kindle2. I've had...
4	4	@mikefish Fair enough. But i have the Kindle2...
...	...	...
1599995	4	Just woke up. Having no school is the best fee...
1599996	4	TheWDB.com - Very cool to hear old Walt interv...
1599997	4	Are you ready for your MoJo Makeover? Ask me f...
1599998	4	Happy 38th Birthday to my boo of alll time!!! ...
1599999	4	happy #charitytuesday @theNSPCC @SparksCharity...

1600498 rows × 2 columns

0s completed at 8:03 PM

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.

```
display(f"X Train Shape: {X_train.shape}")
display(f'X Test Shape: {X_test.shape}')
display(f'Y Train Shape: {Y_train.shape}')
display(f'Y Test Shape: {Y_test.shape}')
```

```
'X Train Shape: (1072333,)'
'X Test Shape: (528165,)'
'Y Train Shape: (1072333,)'
'Y Test Shape: (528165,)'
```

TEXT TOKENIZATION.

- We had to use a maximum of 10000 words to tokenize.
- Tensorflow method is more preferable in this project to tokenize text 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 making the dataset ready for modelling stage in a pre-processed 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.

- Padding of sequences is employed to ensure that all sequences in a list are of the same length, in the pre-processing 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.

```
[28] X_train = pad_sequences(x_training_text_sequences)
     X_test = pad_sequences(x_testing_text_sequences, maxlen=X_train.shape[1])

[29] print(f"\n{X_train.shape}")
     print(f"\n{X_test.shape}")

     (1072333, 116)

     (528165, 116)
```

MODEL STRUCTURE

- Deep learning models are trained using neural network architecture using of data which are well labelled using multiple layers.
- Deep Learning models do exceed human level performance.
- These architectures have the ability to do feature learning from the dataset.
- The sentiment analysis model was designed to have 8 layers with an output layer with softmax activation.
- The summary screenshort below shows the layers added to model, output shape and the parameters which will be used for training, non-training.

MODEL STRUCTURE CONTD..

```
[32] model = Model(input_shape, input_layer)
     model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 116)]             0

 embedding (Embedding)       (None, 116, 20)           10549620

 conv1d (Conv1D)             (None, 114, 32)           1952

 max_pooling1d (MaxPooling1D  (None, 38, 32)           0
 )

 conv1d_1 (Conv1D)           (None, 36, 64)            6208

 max_pooling1d_1 (MaxPooling  (None, 12, 64)           0
 1D)

 conv1d_2 (Conv1D)           (None, 10, 128)           24704

 global_max_pooling1d (Globa  (None, 128)              0
 lMaxPooling1D)

 dense (Dense)               (None, 5)                 645

=================================================================
Total params: 10,583,129
Trainable params: 10,583,129
Non-trainable params: 0
_________________________________________________________________
```

MODEL COMPILING AND TRAINING

- Compiling the model was the first step to take followed by the training.
- The model is forced to compile by default parameters if you don't compile it.
- The compiling was done under loss function with sparse_categorical_crossentopy parameter, metric function with metric index parameter, and SGD as the optimizer parameter.

PREDICTION IN ACTION

- Prediction was done on the training set of the data and the testing set of the data respectively.

```
[36] Y_Train_Prediction = model.predict(X_train)

     33511/33511 [==============================] - 63s 2ms/step

     Y_Train_Prediction

     array([[8.1325871e-01, 9.7002376e-05, 2.3234845e-04, 9.2695111e-05,
             1.8631919e-01],
            [9.4964880e-01, 5.8282516e-05, 1.2421465e-04, 5.1498042e-05,
             5.0117161e-02],
            [9.5935565e-01, 4.0632713e-05, 8.8568049e-05, 3.5367651e-05,
             4.0479690e-02],
            ...,
            [9.3233937e-01, 7.9205951e-05, 1.6855488e-04, 6.9700247e-05,
             6.7343026e-02],
            [2.1159044e-01, 8.6222295e-05, 2.1761740e-04, 8.5885571e-05,
             7.8801990e-01],
            [7.5387782e-01, 2.1659809e-05, 5.6393579e-05, 1.7058634e-05,
             2.4602708e-01]], dtype=float32)

[38] Y_Train_Prediction = numpy.argmax(Y_Train_Prediction, axis=1)

[39] Y_Train_Prediction

     array([0, 0, 0, ..., 0, 4, 0])
```

```
[40] Y_Test_Split_Prediction = model.predict(X_test)

     16506/16506 [==============================] - 30s 2ms/step

[41] Y_Test_Split_Prediction

     array([[9.7554225e-01, 2.1755044e-05, 4.5876357e-05, 1.8101196e-05,
             2.4371980e-02],
            [2.9054362e-01, 1.7023560e-05, 5.2222160e-05, 1.5242744e-05,
             7.0937192e-01],
            [6.7196507e-03, 2.8424477e-06, 9.3999615e-06, 2.7600781e-06,
             9.9326539e-01],
            ...,
            [1.8392679e-01, 1.6591116e-04, 3.9265159e-04, 1.7198692e-04,
             8.1534266e-01],
            [9.7274077e-01, 1.5232890e-05, 3.3041717e-05, 1.0770190e-05,
             2.7200231e-02],
            [9.4094557e-01, 3.0399722e-05, 6.5048225e-05, 2.3792167e-05,
             5.8935143e-02]], dtype=float32)
```

MODEL EVALUATION METRICS USED

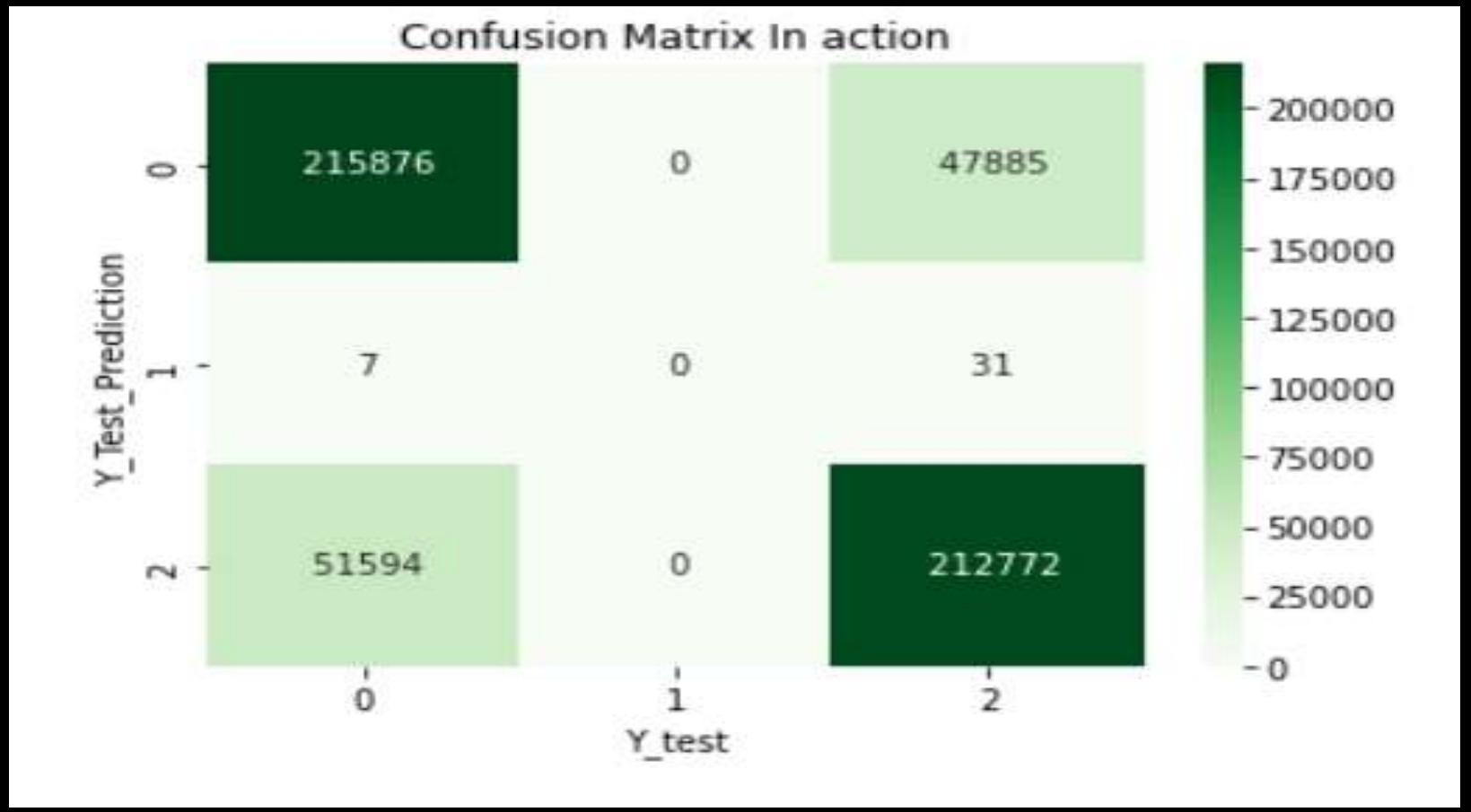
- The model metrics tries to do performance evaluation on a model.
- It is important to note the model evaluation metrics used in this project.
- Below screenshots try to give a highlight on that.

	precision	recall	f1-score	support
0	0.81	0.82	0.81	263761
2	0.00	0.00	0.00	38
4	0.82	0.80	0.81	264366
accuracy			0.81	528165
macro avg	0.54	0.54	0.54	528165
weighted avg	0.81	0.81	0.81	528165

RESULTS IN ACCURACY AND COMPARISONS

- The output helps by providing a know how the model predicts.
- Using the model metrics print out, The accuracy was not that so 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:

CONFUSION MATRIX OF THE Y TEST SPLIT



DISCUSSION AND CONCLUSIONS

- The model was a bit good but with less performance, on the accuracy.
- 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.

THE FOLLOWING SHOULD BE RECOMMENDED:

- Thresholding, 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.
- Incase 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.

CONTD…

- 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, 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.

CONTD…

- 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.

THANK YOU!