Disaster Tweet Analysis

By Hrithik Reddy

Objective(NLP)

To classify(predict) Tweets whether they are related to a Disaster or Not using Deep Learning

Dataset Source and Description

Source - Kaggle
The Dataset had 7613 tweets with labels which was divided into 90 % training data and 10 % validation data which led to to distribution of 6851 and 762
The 40 % tweets of the dataset were related to a disaster and the rest were not. Thus we can conclude we have a balanced Dataset

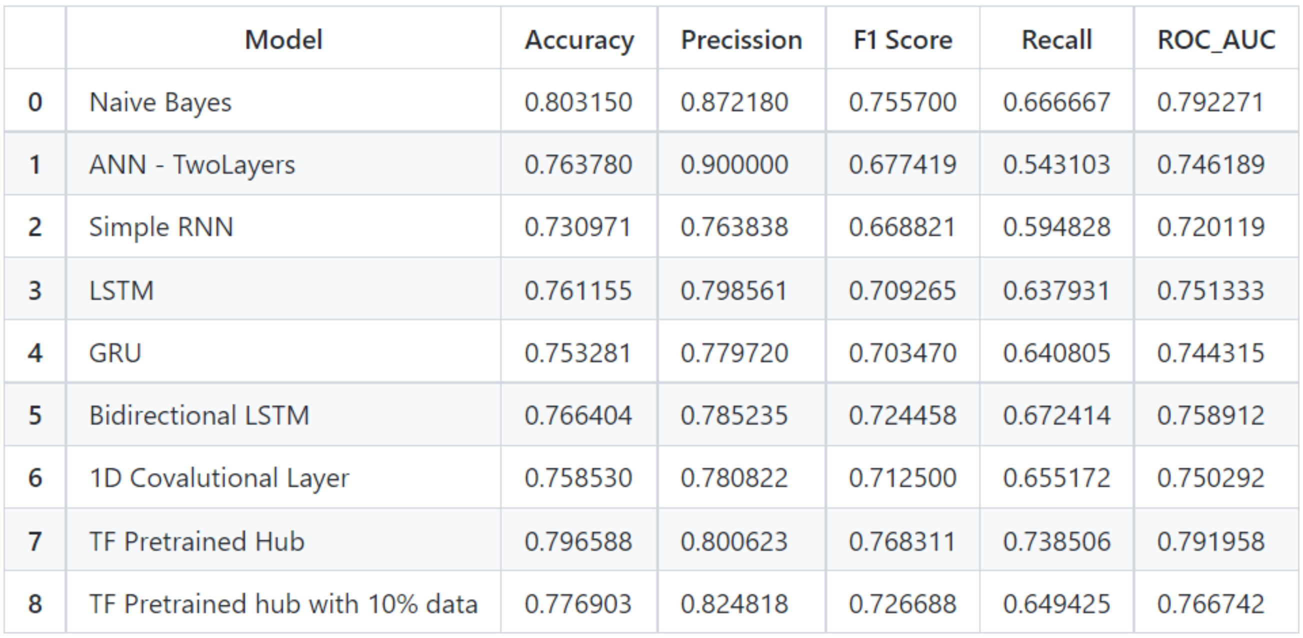
Data exploration & actions taken for Data Cleaning or Feature Engineering

The Dataset was clean by itself
The following steps were taken as Future Engineering -

- Stopwords Removal
- Tokenization - Tfid_vectorizer from sklearn and Token_Vectorizer from Tensorflow were used with max tokens of 20,000 and output-sequence length of 25
- Embedded Layer - Embedded Layer from TensorFlow was used with output dimension length of 128 adapted to the training data and an pretrained embedded model from Tensorflow hub was used which had output dimension length as 512

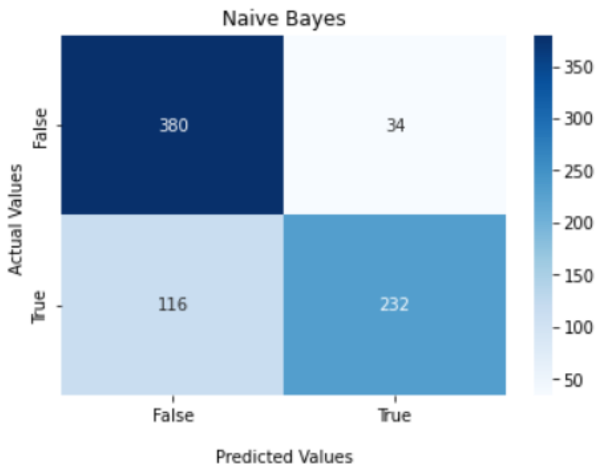
<u>Summary of Models Trained</u>

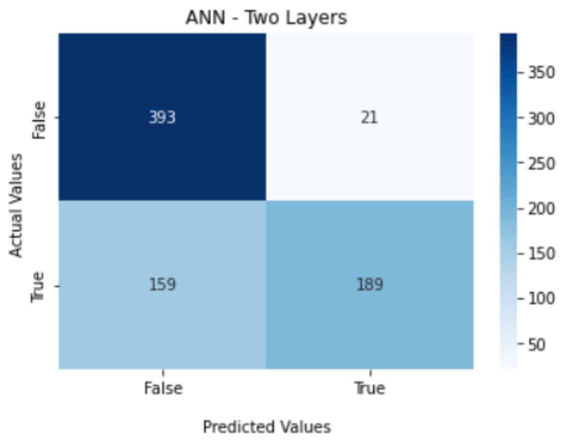
Model Metrics - on Validation Data (Test Set)

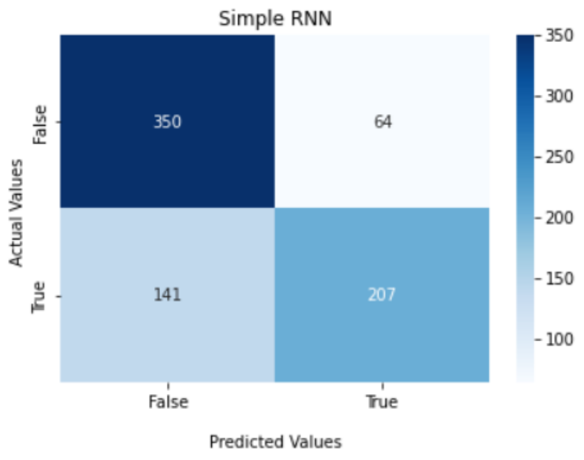


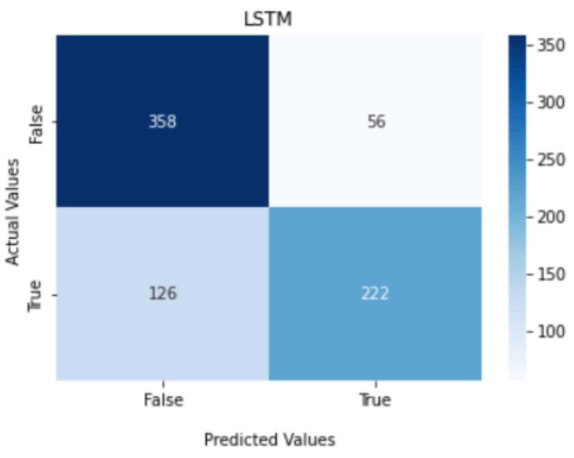
	Model	Accuracy	Precission	F1 Score	Recall	ROC_AUC
0	Naive Bayes	0.803150	0.872180	0.755700	0.666667	0.792271
1	ANN - TwoLayers	0.763780	0.900000	0.677419	0.543103	0.746189
2	Simple RNN	0.730971	0.763838	0.668821	0.594828	0.720119
3	LSTM	0.761155	0.798561	0.709265	0.637931	0.751333
4	GRU	0.753281	0.779720	0.703470	0.640805	0.744315
5	Bidirectional LSTM	0.766404	0.785235	0.724458	0.672414	0.758912
6	1D Covalutional Layer	0.758530	0.780822	0.712500	0.655172	0.750292
7	TF Pretrained Hub	0.796588	0.800623	0.768311	0.738506	0.791958
8	TF Pretrained hub with 10% data	0.776903	0.824818	0.726688	0.649425	0.766742

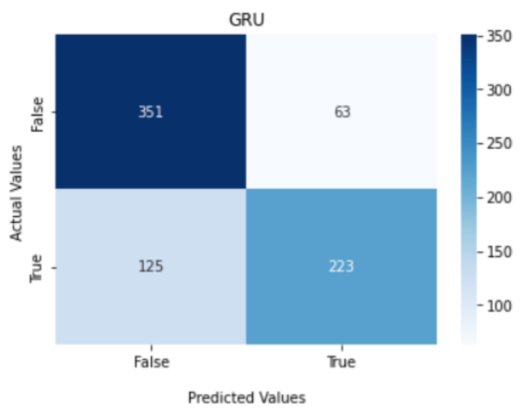
Confusion Matrix of each Model on Validation Data

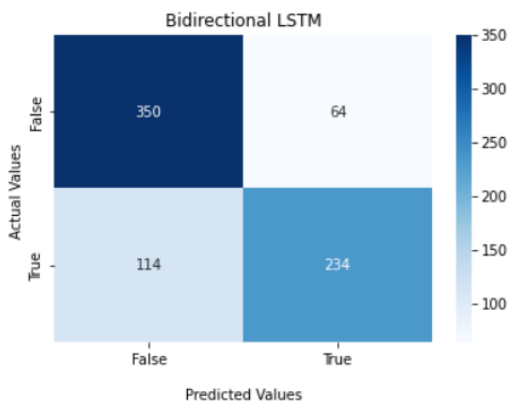


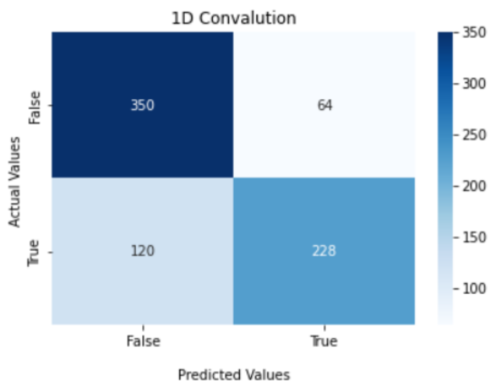


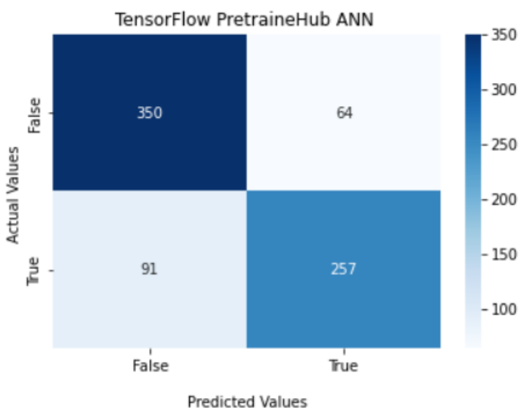


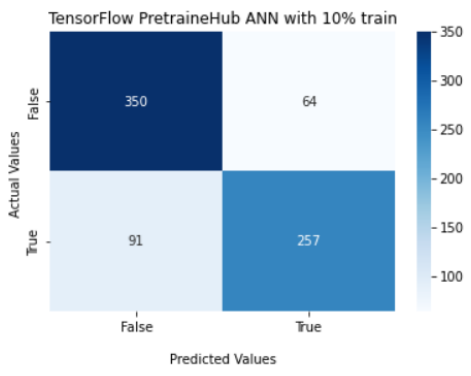












Best Model

The best model for the above objective would be voting classifier of Naive Bayes , ANN-Two Layers and TF pretrained hub ANN models

Key Findings and Insights

All models from 1 to 6 were overfitting with training data achieving an accuracy of 99% with train data and 70 - 76 % accuracy with the test data , this led to introduction of Dropout Layers , kernel_regualizer and bias_regulizer in some layers and also early stopping of training with some models being trained to as low as 2 epochs. Some models achieved 99 % accuracy on train set without regularisation terms and dropout layers .
Even after introducing regularisation and dropout layers Bidirectional LSTM achieve 99% accuracy in the second epoch.wn

In general for all the models Adam optimizer did better than SGD or mini-batch or GradientDescent

Suggestions for next steps

Compared to other Deep Learning models(Models 1 to 6) ,Bidirectional LSTM did better with our own embedding layer . I believe that a new model with a layer of Bidirectional LSTM and 2 subsequent dense layers along with Tensorflow’s pretrained hub embedding being trained on (k=10)k-fold cross validation with regularisation and dropout layers would be able to achieve an better accuracy over the test set.

Code Link -

https://github.com/Hrithik2212/Disaster-Tweet-Analysis-with-Tensoflow-NLP