**Sentiment Analysis on Twitter data**

**Abstract**:

Identifying opinions and sentiments from tweets is called as “Twitter Sentiment Analysis”. The main idea behind Twitter sentiment analysis is to determine the polarity of the tweet and classifying them into positive or negative sentiment tweet. In this project I compared Bidirectional Encoder Representations from Transformers (BERT) model with Random Forest Classifier for to determine the polarity of the twitter data. At first, the raw tweets are pre-processed by removing the urls, white spaces, characters that are not letters, all the stopwords, punctuations, and lemmatization. Then, utilized pre-trained BERT pre-processor and encoder from tensor flow and random forest classifier for sentiment classification. I have evaluated the model performance with metrics like confusion matrix, accuracy, precision, recall, f1 score, and auc.

**Introduction**:

Now a days, Internet is becoming very popular and more important. It is serving as a cost-effective platform and used of social media become a necessary daily activity as it is used for social interaction. Several social media platforms like Facebook, Instagram, blogs, reviews, tweets are being processed for extracting the people’s opinions about any particular product feedback, organization, medical experience, professional knowledge, or any situation. The attitude and feeling behind any opinion plays a vital role in evaluating behaviour of an individual which is known as sentiment.

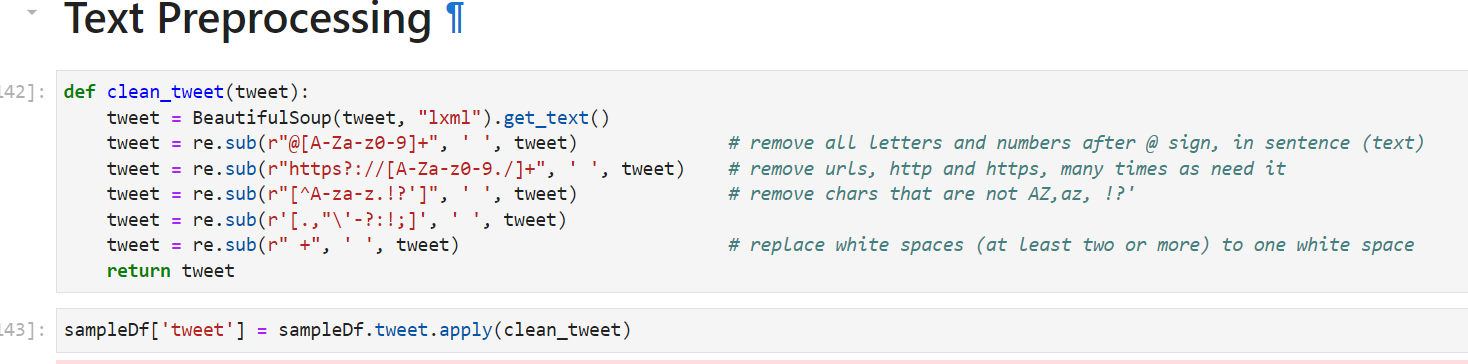
Different Machine learning methods were introduced to perform sentiment analysis those are categorized under unsupervised and supervised leaning methods. This project is mainly contributed to compare the performance of sentiment analysis using Bidirectional Encoder Representations from Transformers (BERT), a popular and effective model which makes used of Transformer that learns contextual relation between words in a text The BERT models use the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after, hence the name: Bidirectional Encoder Representations from Transformers and Random Forest Classifier constructs multiple decision trees and finally merges them together to gain absolute and stable value of class.

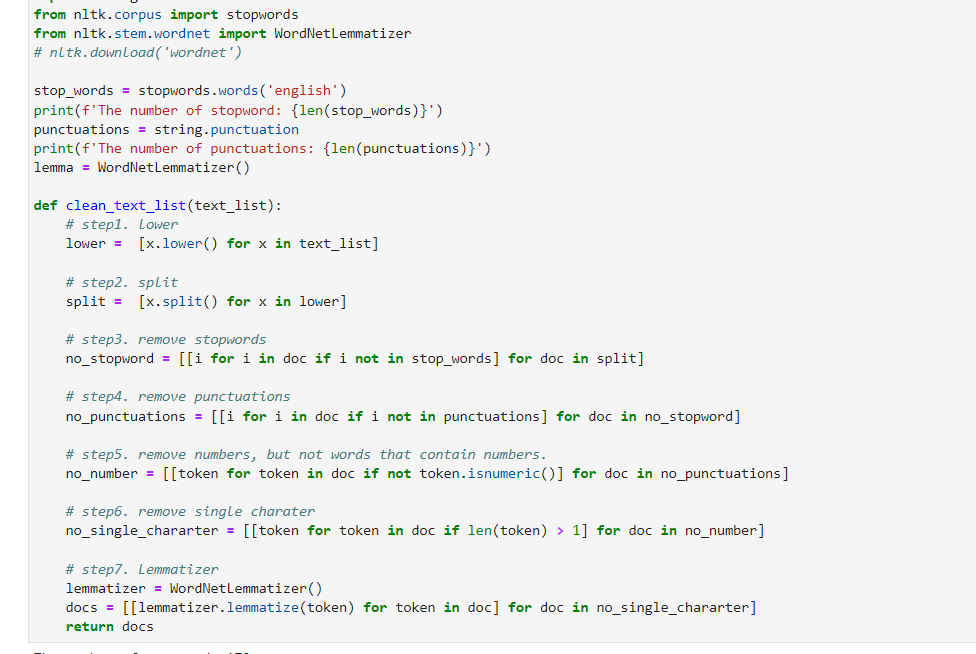
The objective of this project was to build an effective approach based on the BERT model and Random Forest Classifier for Twitter sentiment analysis. First, I explored the raw twitter data and pre-processing is done by creating a balanced dataset and removing the urls, white spaces, characters that are not letters, removed all the stopwords, punctuations and lemmatized the tweets, then for BERT, I build word embedding layers with an input layer, dropout and dense layers to build BERT model by utilizing pre-trained Tensor flow BERT model on plain text. For Random Forest classifier, I build bigram bag of words and transformed into a matrix of TF-IDF features for train data and applied same transformation on test data. Implemented both BERT pre-trained model and Random Forest Classifier model with training data, fine-tuned and evaluated on test dataset and predicted the output using test data. Performance metrics carried out to measure the performance of each model to compare.

**Technical Approach**:

Step-1: Pre-processing of data.

The raw tweets in the dataset generally result in a very noisy due to people’s random and creative use of social media. Tweets have certain special features, i.e., emojis, emoticons, hashtags and user mentions, coupled with typical web constructs, such as email addresses and URLs, and other noisy sources, such as phone numbers, percentages, money amounts, time, date, generic numbers, stop words and punctuations. The pre-processing step is needed in order to remove all the words that are irrelevant for sentiment analysis.





Step-2: BERT model and Random Forest Classifier

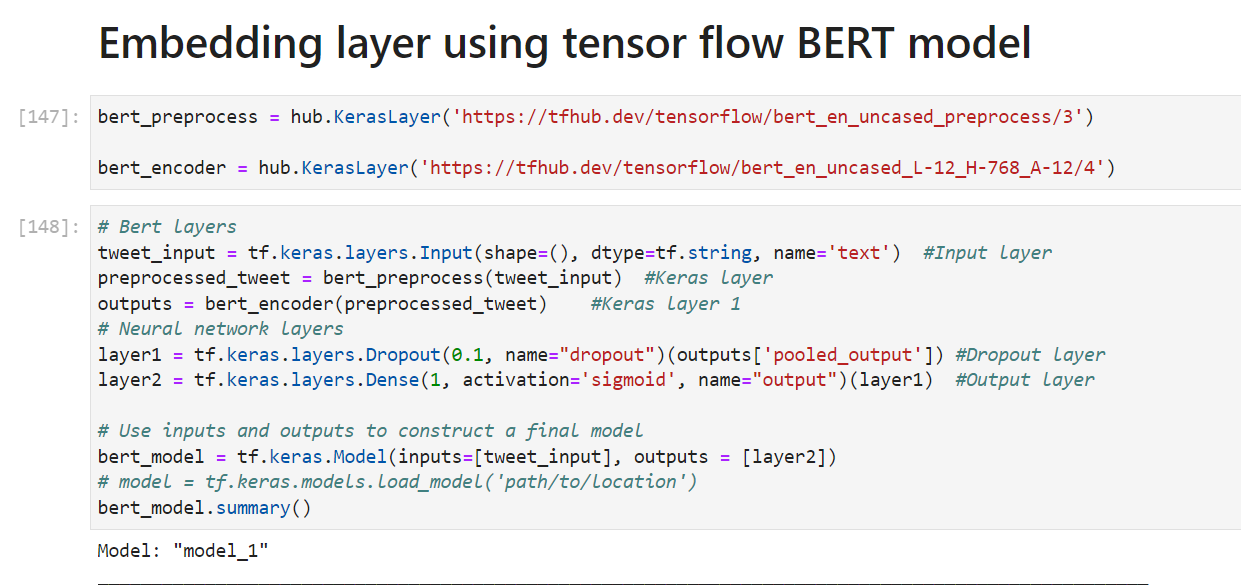
BERT model is used for emotion classification. The meaning of a word in a given sentence depends on the other words surrounding it. The BERT feeds all input at once to handle dependencies among words. The BERT-base model uses 12 transformer encoders, we can easily fine-tune the BERT model to get the desired results.

In this project I followed the following steps to create a mask and the encoder representation of the BERT using Tensorflow with Keras library for emotion classification:

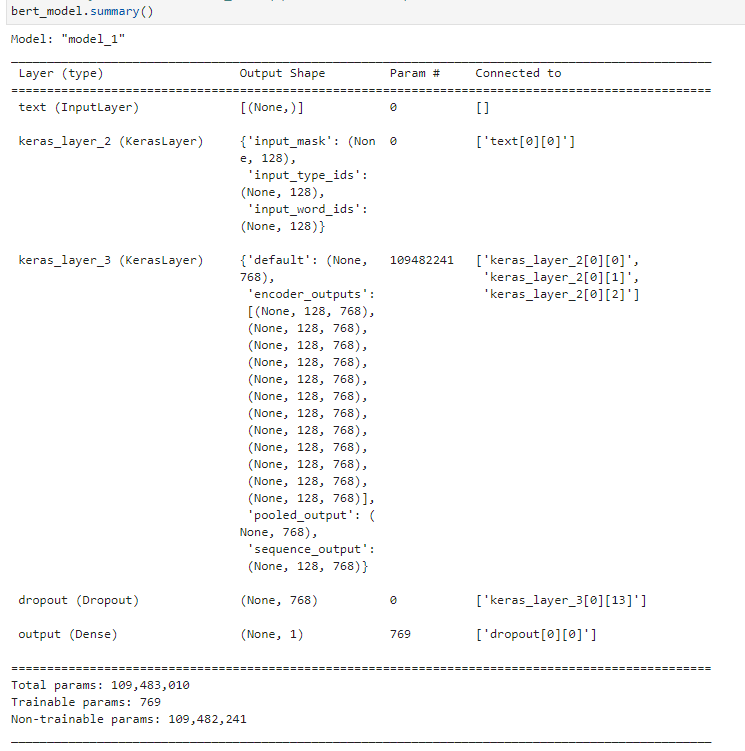
* Dividing the collected data into training and testing sets using train-test split.



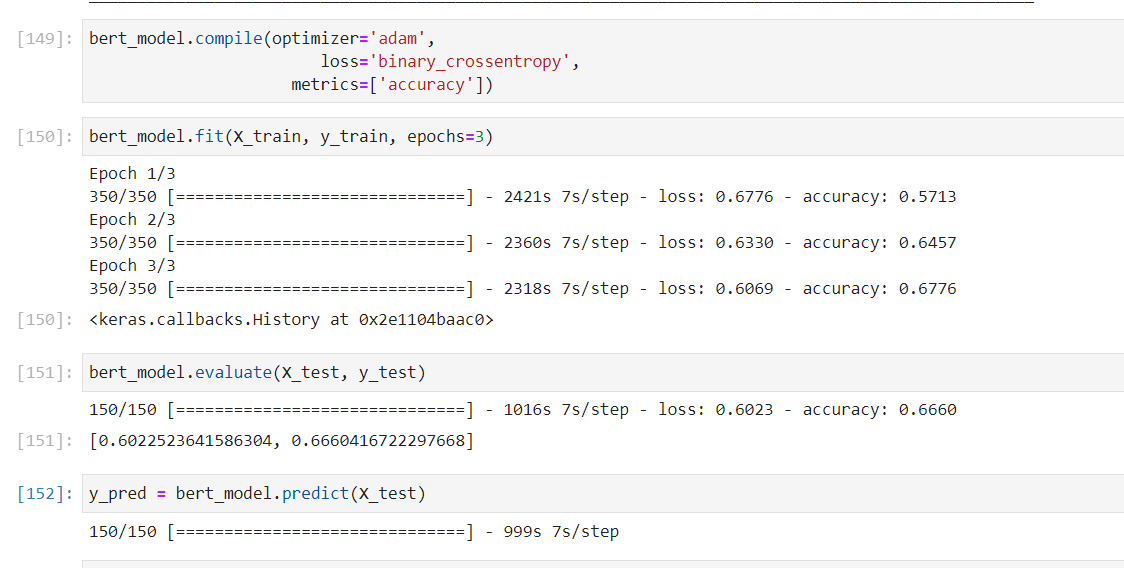
* Build word embedding layers on pre-processed twitter data with an input layer, dropout and dense layers to build BERT model.



Summary of the model created with embedding layers –



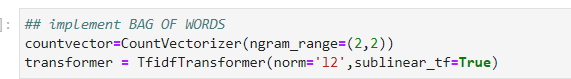
* Compiling, model fitting on training data and evaluating the model on test data.
* Used Adam as our optimizer, binary\_crossentropy as our loss function, and accuracy as our accuracy metric. Fine-tuning the model for 3 epochs.



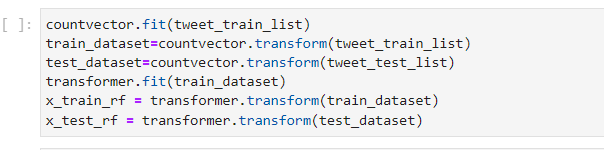
Random Forest Classifier is an ensemble of decision tree classifiers which will output a combined prediction value of each tree in the ensemble. Each decision tree is constructed by using a random subset of the training data with a fixed probability distribution. The deeply grown decision trees has low bias and high variance. Hence, they can learn irregular patterns and overfit their training sets. Random forests give improvement over just bagged trees because they decorrelates the trees in the Random Forest.

In this project I involved following steps of random forest algorithm

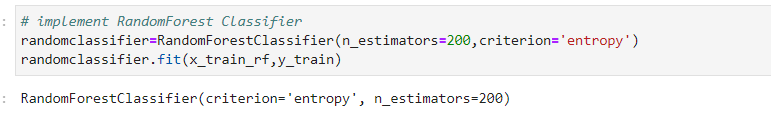
* Converted text documents into a matrix of bigram counts.



* Bag of words transformed into a matrix of TF-IDF features for train data and test data.



* Random forest model implemented with 200 decision trees to average the predictions.



**Results and Analysis:**

Evaluated the performance of both the models with following metrics.

**Confusion matrix** -

It is a table with 4 different combinations of predicted and actual values.

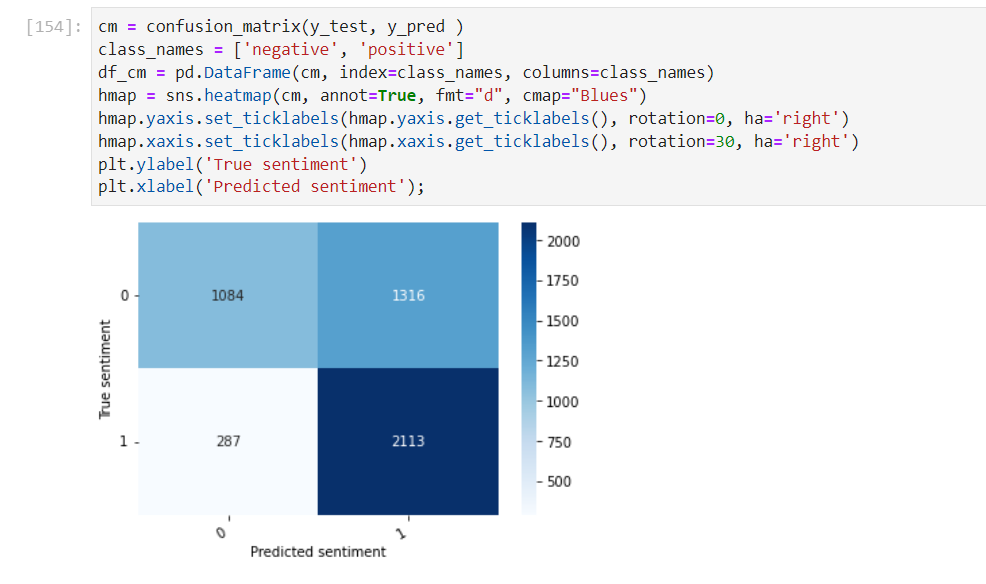
True Positives: Model predicted positive and it’s true.

True Negatives: Model predicted negative and it’s true.

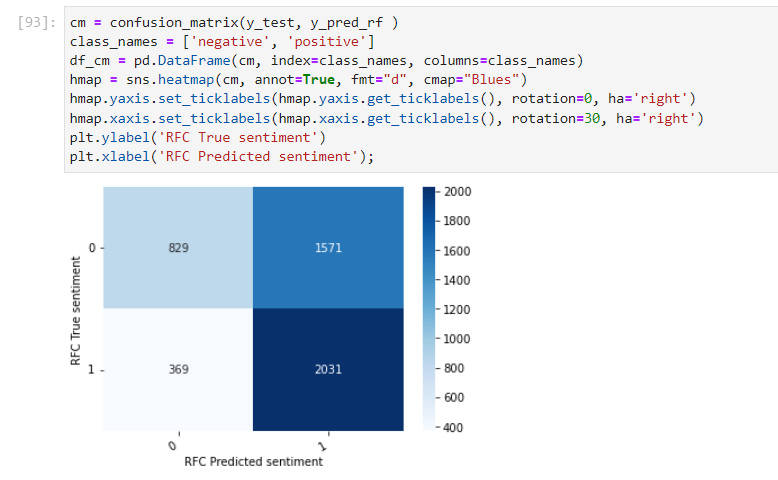
False Positives : Model predicted positive and it’s false.

False Negatives: Model predicted negative and it’s false.

For BERT:



For Random Forest:



**Accuracy -** From all the classes (positive and negative), how many of them model have predicted correctly.

**Recall** - From all the positive classes, how many model predicted correctly.

**Precision** - From all the classes (positive and negative), model have predicted as positive, how many are actually positive.

**F1 score** - The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

**AUC** **(Area Under the ROC Curve)** - it is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

For BERT model:



For Random Forest:



**Conclusion** –

The BERT model performed well on predicted sentiment of tweets compared to random forest classifier with F1 Score – 0.72 and AUC score – 0.77 which is reasonably good compared random forest model with F1 Score – 0.60 and auc of 51%.