**Python/Deep Learning Project Report**

**Classification of News into Categories**

**Based on Headlines**

**Team ID: 12**

**Member 1: Sushma Manne Class ID: 22**

**Member 2: Sushma Sri Surapaneni Class ID:35**

**Member 3: Bhavana Deepthi Kota Class ID: 19**

**Motivation:**

Nowadays on the Internet there are a lot of sources that generate immense amounts of daily news. In addition, the demand for information by users has been growing continuously, so it is crucial that the news is classified to allow users to access the information of interest quickly and effectively. This way, the machine learning model for automated news classification could be used to identify topics of untracked news and/or make individual suggestions based on the user prior interests.

**Objective:**

Our aim is to build models that take news headline and short description as input and output news category. So, we have used different machine learning algorithms and also used Convolution Neural Network and found which model yields better performance by measuring the accuracy.

**Data and Features:**

**Data:**

Our dataset is a kaggle dataset that contains almost 200k news headlines from the year 2012 to 2018 obtained from Huffpost. News in this dataset belong to 40 different labels. Each news record consists of several attributes from which we are using only Category, Headline and Short description in our analysis. In addition, we combine data attributes Headline and Short description into the single attribute as the input data for classification.

**Features:**

We can divide the whole project into 3 components as follows:

**Dataset Preparation:**

The first step is the Dataset Preparation Step which includes the process of loading a dataset and performing basic pre-processing. The dataset is then splitted into train and validation and test sets.

**Feature Engineering:**

The next step is the Feature Engineering in which the raw dataset is transformed into flat features which can be used in a machine learning model. This step also includes the process of creating new features from the existing data. So, we have implemented count vector as features in our work.

Count Vector is a matrix notation of the dataset in which every row represents a document from the corpus, every column represents a term from the corpus, and every cell represents the frequency count of a particular term in a particular document.

Also we have checked the variance and dropped some of the features based on threshold.

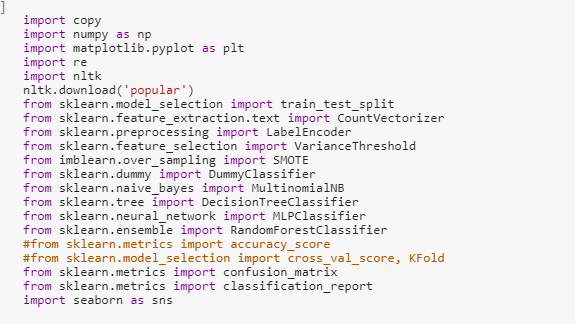
**Model Training:**

The final step is the Model Building step in which a machine learning model is trained on a labeled dataset. Here in our project we have implemented Naive Bayes Classifier, Decision Tree Model and Convolution Neural Network.

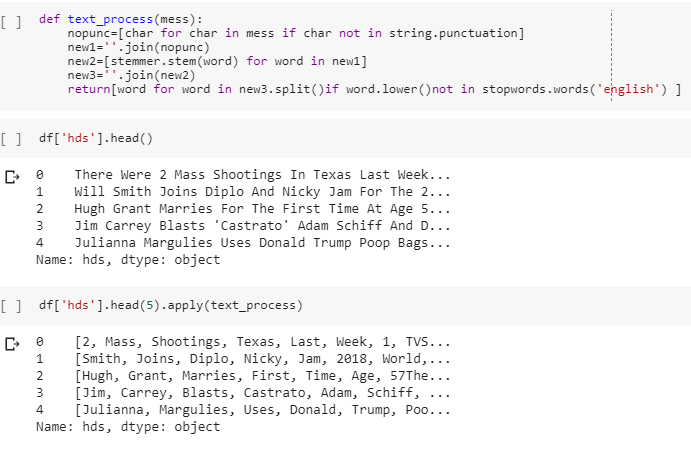
**Related Work:**

First we have Imported Libraries and Loaded the dataset as follows. Also we have concatenated both the headline and short description and stored it into attribute ‘hds’.



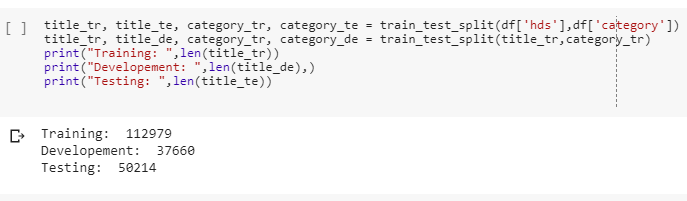


We have cleaned the headline as follows:



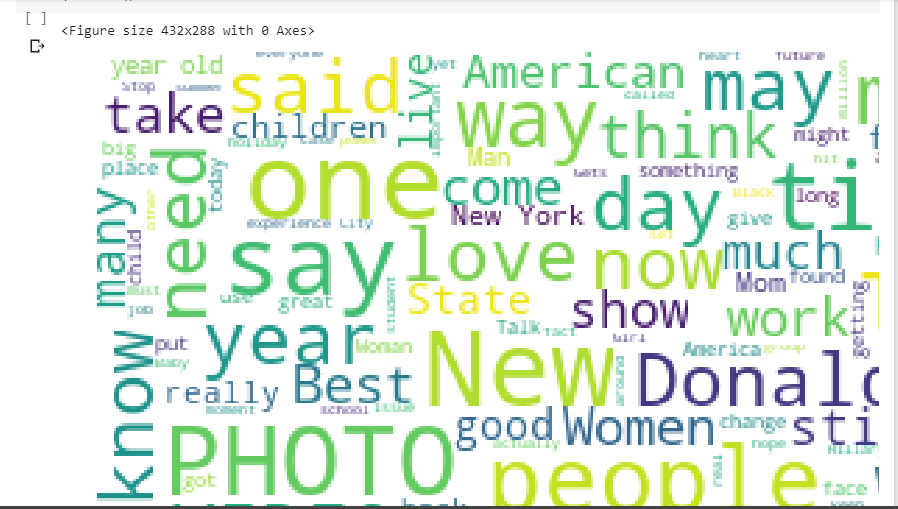
**Splitting of Dataset:**

Splitted data into 3 parts - training, development and test. We will use training data to train out model and use development data to check and tune hyper parameters. And finally use test data to see how our model performs.



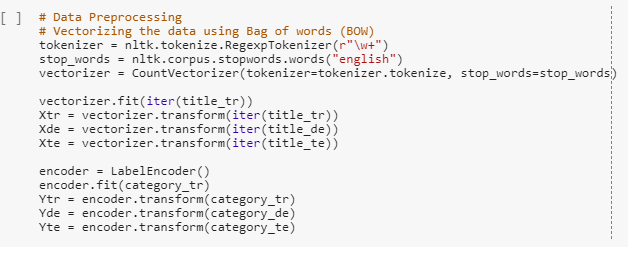
Using wordCloud we visualized our data:



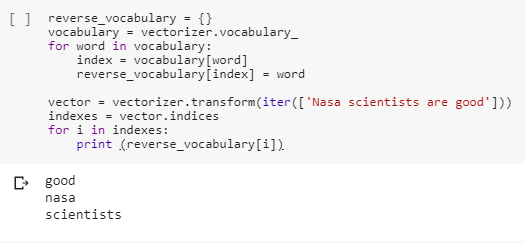


**Data Preprocessing:**

Vectorizing the Data using Bag of Words:

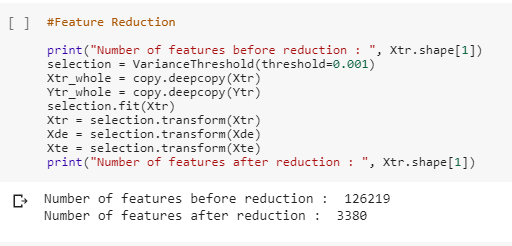


Lets look at what exactly is this vectorizer doing. We will first create reverse dictionary from the vectorizer. Iterating over the vectorized sentence ”Nasascientistsare good”. We get the vector to be representative of three words "good", "nasa" and "scientists". The order has been changed because bag of words does not preserve order.



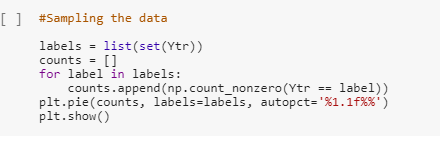
**Feature Reduction:**

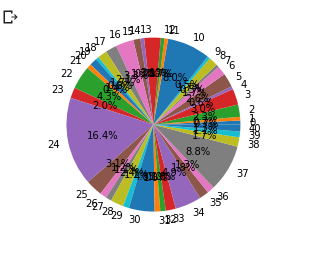
We can check the variance of the feature and drop them based on a threshold:



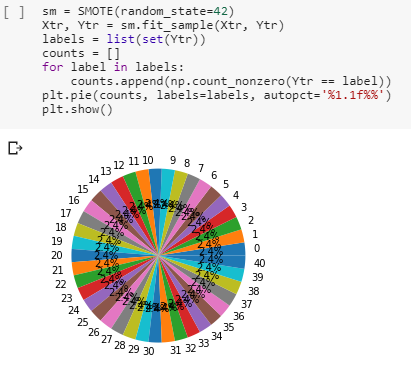
**Sampling of the Data:**

We counted the number of different labels in dataset and plotted a pie chart distribution.





As we can see the class labels are not uniformly distributed so we used SMOT and over sample the classes which are less in number so that classes are equally distributed.

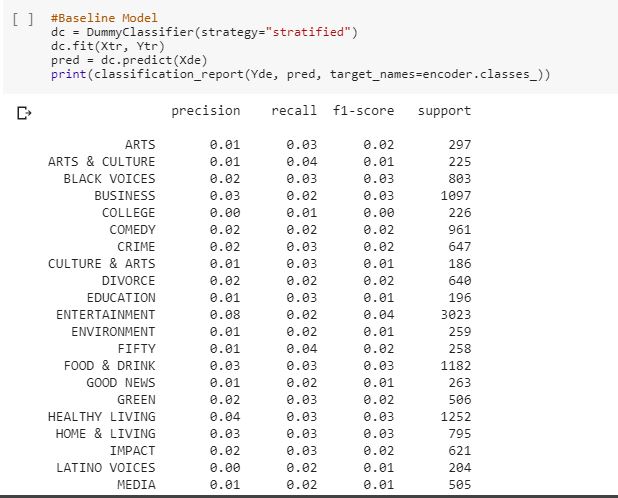


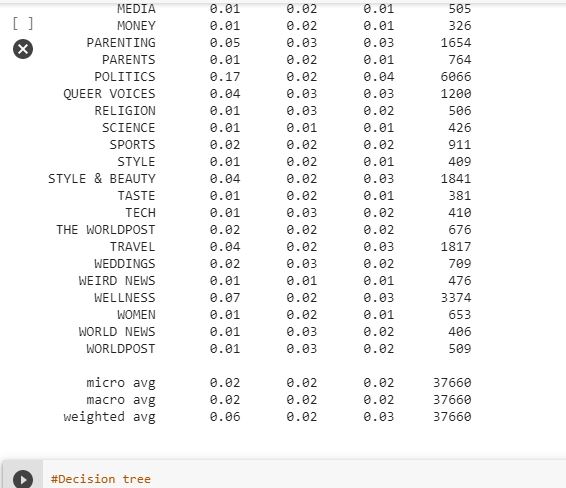
**Model Training:**

The following are the models we used to train our data:

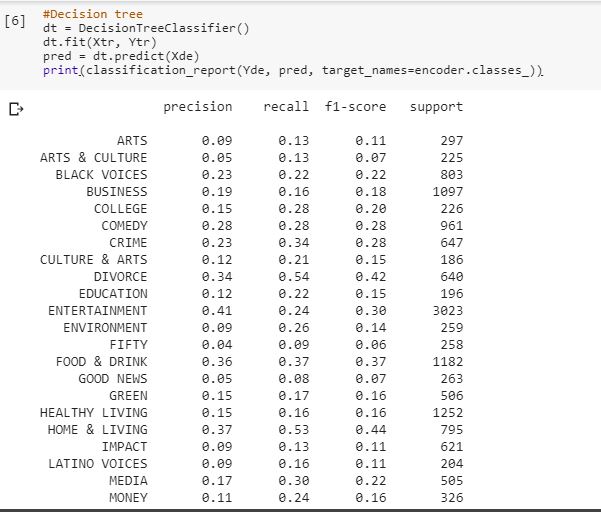
**Baseline Model:**

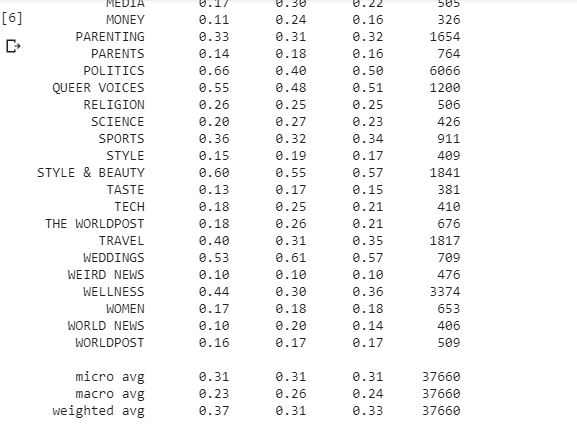
“stratified”: generates predictions by respecting the training set’s class distribution.



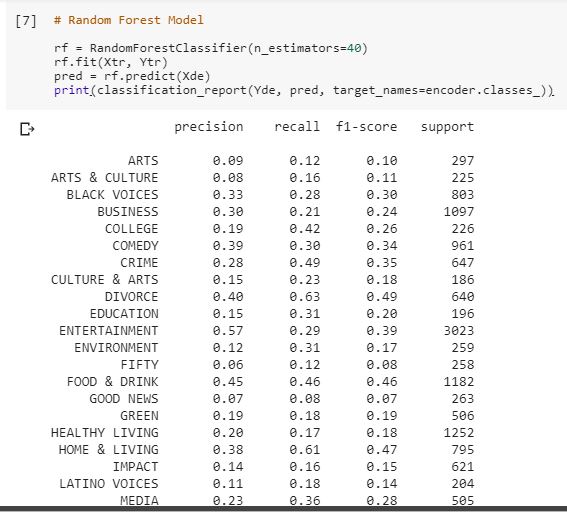


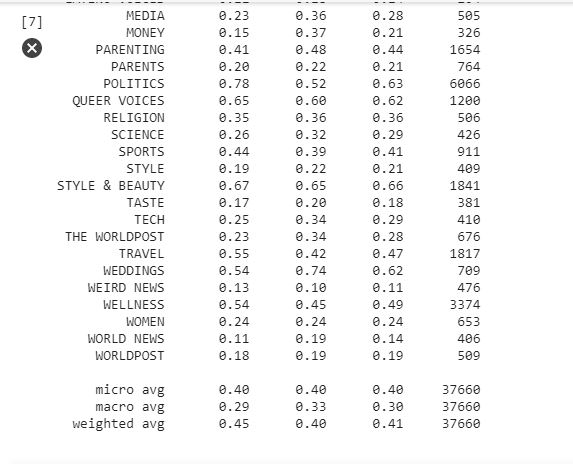
**Decision Tree:**



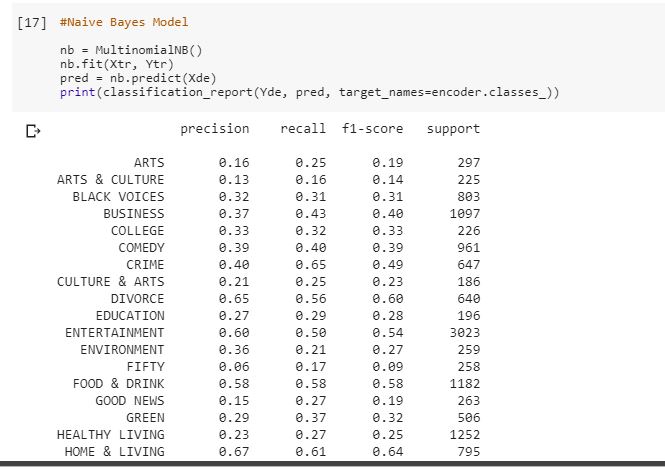


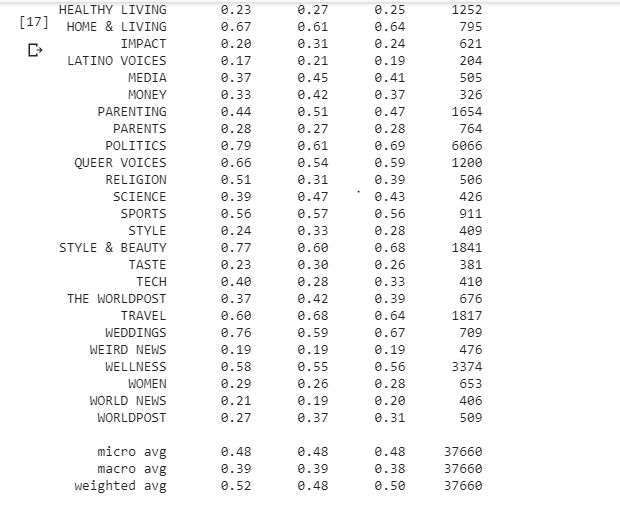
**Random Forest Model:**





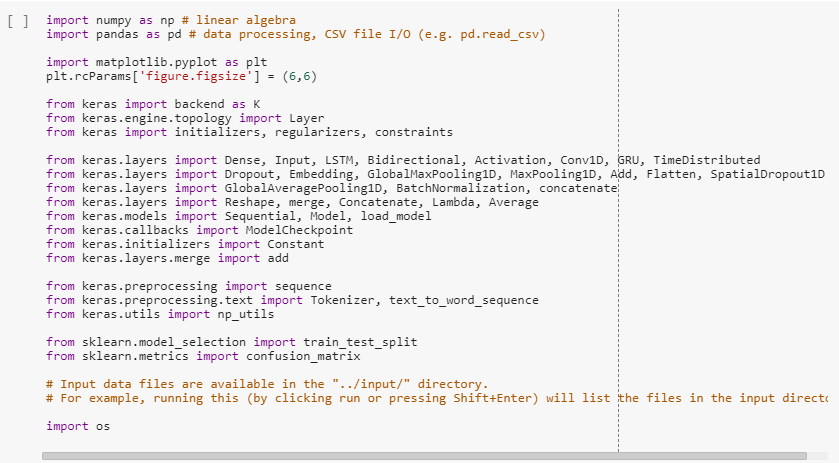
**Multinomial Naive Bayes Approach:**

****

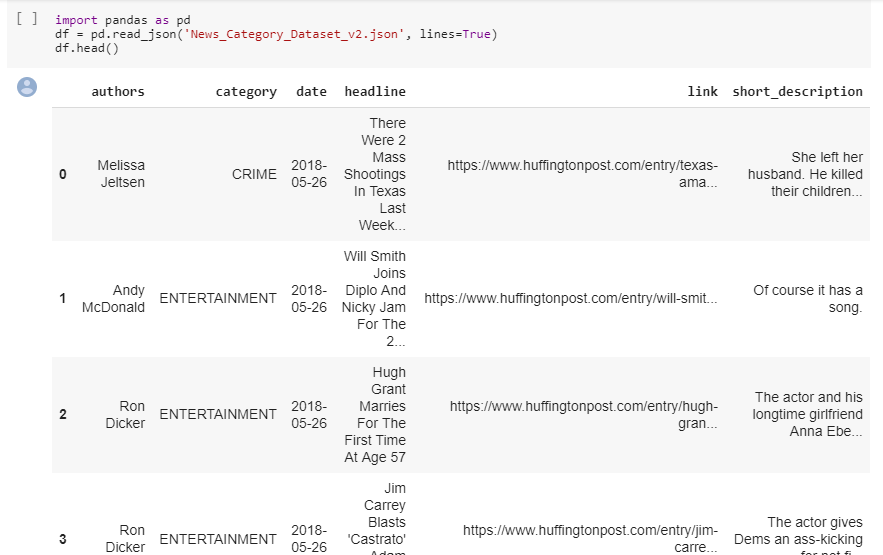
****

**Convolution Neural Network:**

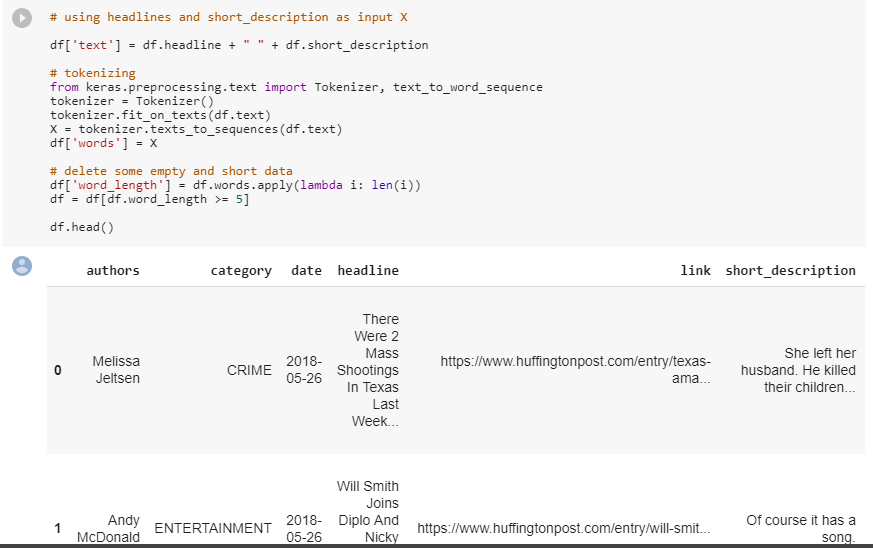
For CNN model we first imported libraries and loaded the data and then data pre processing is done. We have used glove embedding here. And then splitted the data and applied CNN model to the processed data. And observed the performance of the model. The execution of our CNN model is as follows:

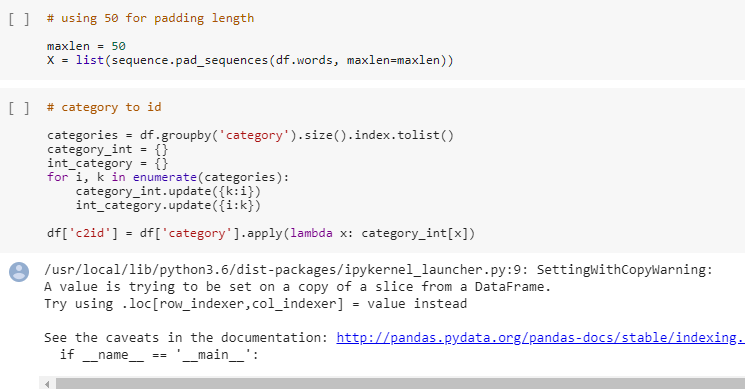


Loaded the dataset:

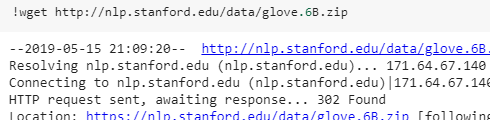


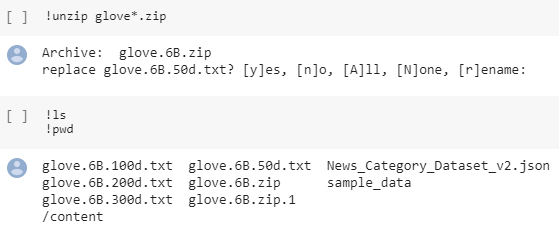
**Data Preprocessing:**

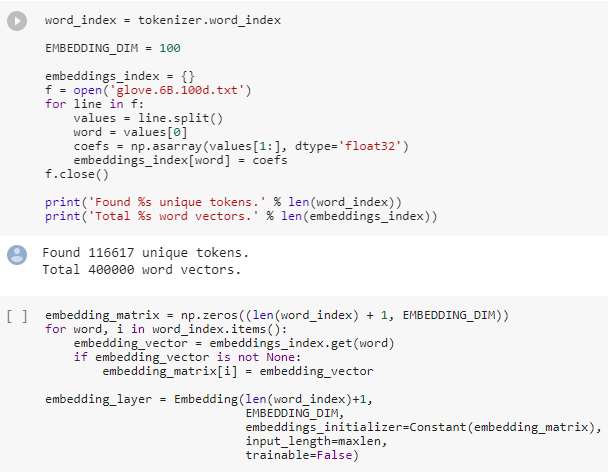




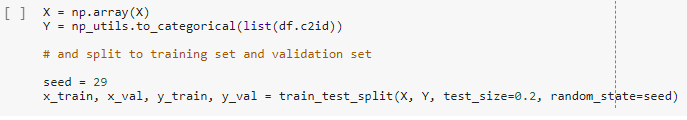
**Glove Embedding:**



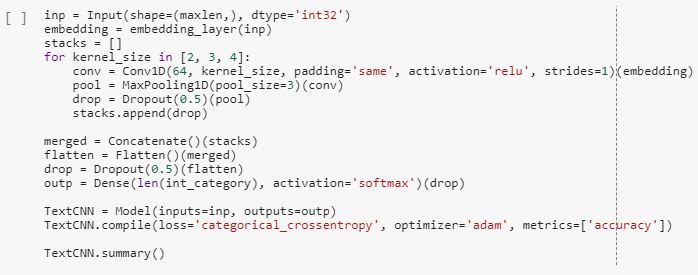


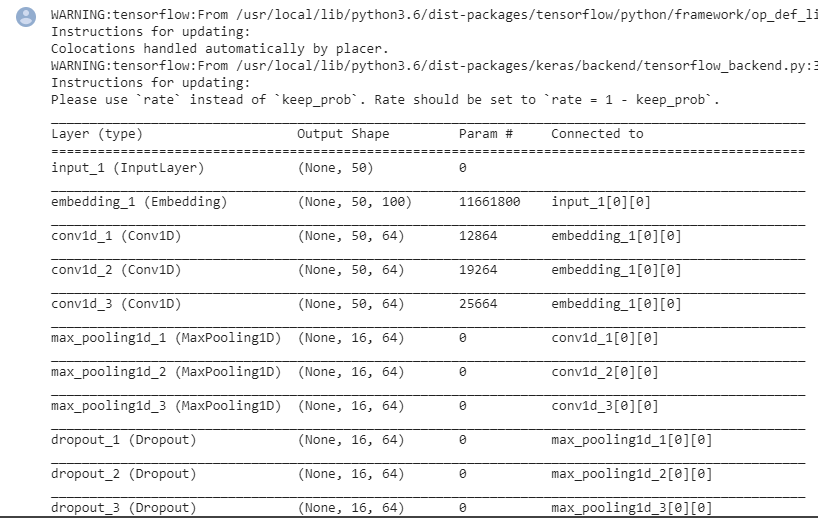


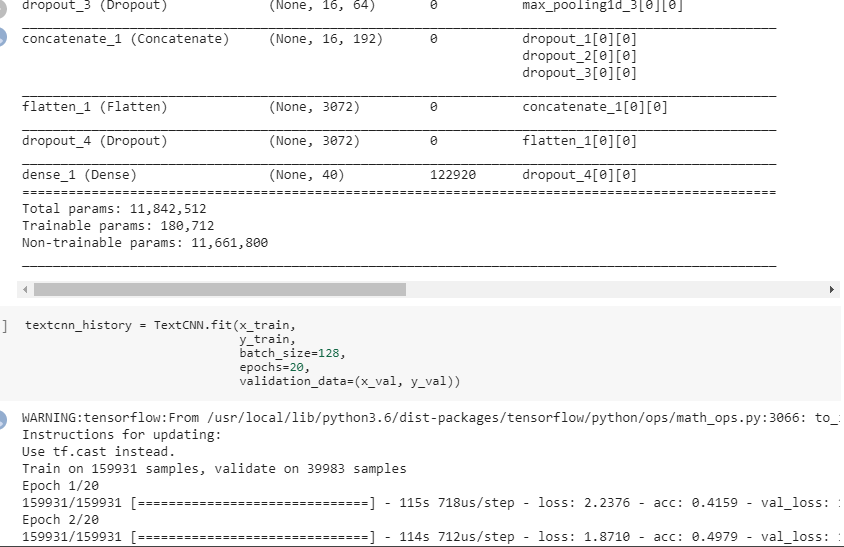
**Splitting the Dataset:**

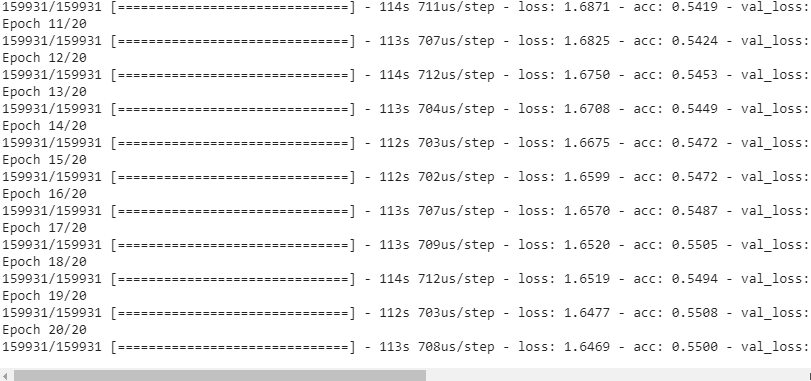


**CNN Model:**

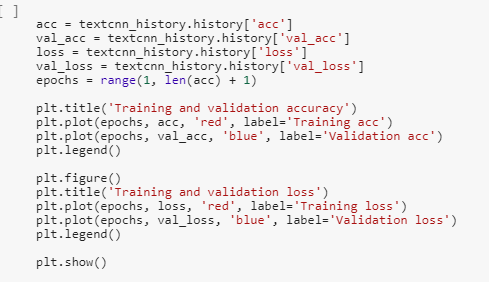




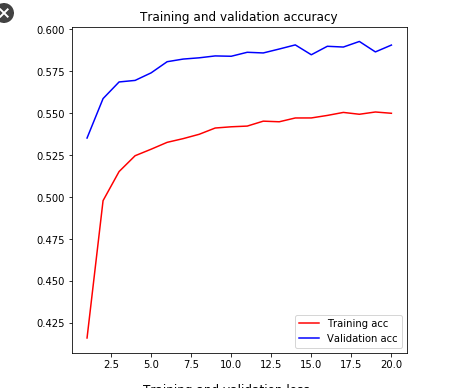




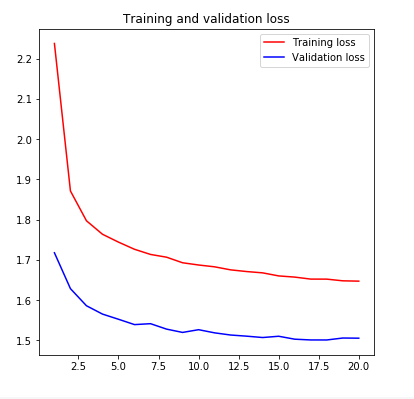
**Plotted loss and accuracy:**



**Accuracy Curve:**



**Loss Curve:**



**Overall Architecture:**

Feature Engineering

(Count Vectorization,Glove Embedding

Tokenization

Data PreProcessing

Data Set

Model Evaluation

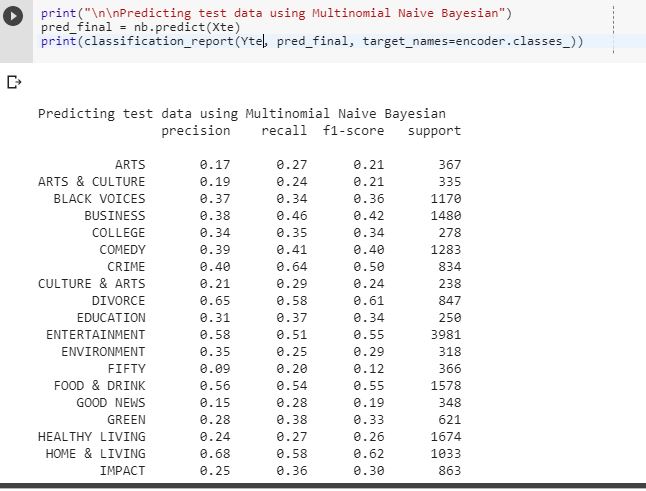
Model Building

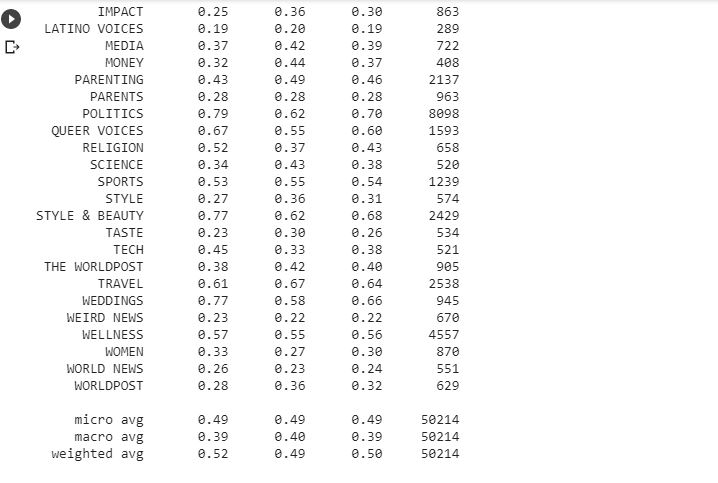
(CNN,Decision Tree, Random Forest, Naive Bayes)

**Evaluation and Results:**

We have performed different machine learning and deep learning algorithms like Multinomial Naive Bayes, Random Forest, Decision Tree and Convolution Neural Network on the dataset we have chosed. We have evaluated the performance of machine learning models by considering the F1 score. The F1 scores for Multinomial Naive Bayes is 0.50, Random Forest Model is 0.41, Decision Tree is 0.33, Baseline model is 0.03. From those scores we can conclude that Naive Bayes model yields a better performance when compared to other algorithms.

So we have predicted on test data using Multinomial Naive Bayes approach and obtained the following results.





We have also checked if there are any incorrectly categorized data while performing Naive Bayes model as it got more accuracy as follows:

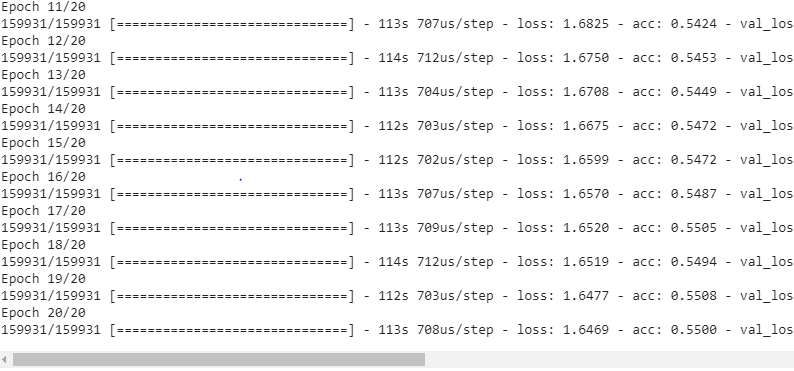


We can see from above that all the words are correctly related to the category they are in.

**Result of CNN :**

CNN is generally used for image classification. Dividing the picture into pixels and then divide each pixel. It can be used for the labeled data as well. So we have tried this on our data set also as it is labeled data.

When applied Convolution Neural Network to the train set we got an accuracy of 55% which is more when compared to the other algorithms we applied.



**Challenges Faced:**

As we have used dataset containing around 200K records it took a lot of time to run the models and also there is so much for the data preprocessing phase. Overall we consumed more time for cleaning the data and preprocessing the data as required.

**Future Work:**

* We can use other machine learning and deep learning algorithms on the data set and can see which model gives better accuracy.
* We can also induce other methods in feature engineering and also parameters and can see how it affects the accuracy of the model.