**Python/Deep Learning Project Report**

**Classification of News into Categories Based on Headlines & Short Description**

**Team ID: 2**

**Member 1: Akhil Teja Kanugolu Class ID: 11**

**Member 2: Geetanjali Makineni Class ID:13**

**Github Link:**

<https://github.com/geetamakineni/PYTHON-PROJECT>

**Dataset link:**

<https://www.kaggle.com/rmisra/news-category-dataset>

**Overall Architecture:**

Data Pre- Processing

Feature Engineering

Data Set

Data Pre- Processing

Machine Learning

Deep Learning

(Decision Tree, Random Forest, Naïve Bayes, SVM) (CNN)

**Work**:

Basically, we divided the whole project into 3 components as

* Dataset Preparation
* Feature Engineering
* Model Training

1. Dataset Preparation: The first step here is dataset preparation where we load the dataset and perform the basic preprocessing. The dataset is further split into training, validation, and test sets.
2. Feature Engineering – In this step, the raw dataset is transformed further into flat features. This mainly includes process in which we create new features from the existing features. We use the Count Vector matrix notation where we check variance and drop some of the features based on the threshold it handles.
3. Model Training – The final most step we use is the Model Building where a machine learning model is here trained on the dataset. We implement the models like Naïve Bayes Classifier, Convolution Neural Network and Decision Tree Model and find which model gives best accuracy.

**Process:**

Initially, we imported the required packages and loaded the dataset as below. And we also concatenated the headline as well as the short description into the single attribute named ‘Combined\_H&SD’ for gaining more data to categorize the category of news type. And performed custom function process\_data() where we applied stemmer which helped to remove duplicate words & removed the stop words, special characters. Based on the Data with 41 News categories using SMOTE function we splitted the Data with equal for increasing the efficiency. Thereafter, we classified the Data into Train, Development, Test Data: Train data is used for the training different Model, Development Data for tuning the Parameters, Test Data used for validating the different models trained. And visualized the Words which was processed with Word Cloud.

In the Data preprocessing, we did vectorization using BOW (Bag of Words). Later, we processed the data with tokenizer from nltk corpus library. This is classification model problem as the predicted value belongs to category. So, we use one hot label encoder to mask the predicted category.

Next, we will use feature Reduction based on the Variance Threshold=0.001 which removes the data that will not have the impact on the prediction. Data sampling was done because the categories are unequally distributed which may overfit or underfit some categories with more data or less data. This will help to train the data equally for every category using the SMOTE function.

Moving Next, Training of model using Machine Learning Models:

1. Decision Tree Model: Created the Decision Tree Model with dtc\_model and trained the model and got the accuracy 31% and viewed the F1 score with help of classification report
2. Random Forest Model: Created the Random Forest Model with rf\_model and trained the model and got the accuracy 34% and viewed the F1 score with help of classification report
3. Multinomial Naïve Bayes Classification: Created the Multinomial Naïve Bayes Model with nb\_model and trained the model and got the accuracy 52% and viewed the F1 score with help of classification report.

By Comparing other models, we got better accuracy for the Development Data. So, we predicted the data for test data & which resulted the accuracy of 54%. And when we process the data as we used Bag of words to vectorize it will not give the order of words, so we created custom function reverse vocabulary. Finally appended the words for each category helps for the prediction based on whole training data.

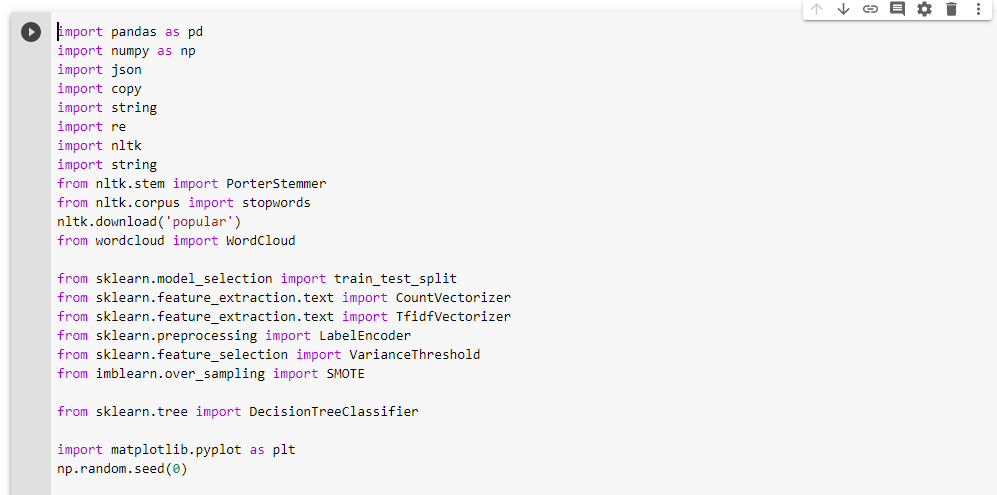
1. Support Vector Classification: Created the Decision Tree Model with svc\_model and trained the model and got the accuracy 54% and viewed the F1 score with help of classification report.

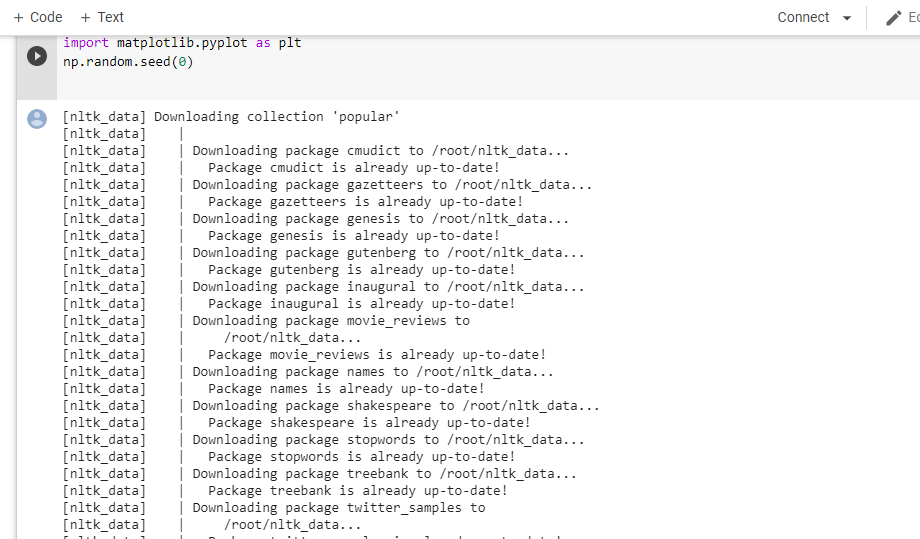
As we got accuracy more than Multinomial Naïve Bayes, we performed same steps as Multinomial Naïve Bayes Model on test Data such as Reverse vocabulary & viewed the words for the prediction.

Using Deep learning Model:

Here we are trying to perform Convolution Neural Network. After importing the packages required for the Analysing the Dataset, we will read the JSON file and store to df. Apart from Machine learning model we tried different techniques to pre-process the Data like started with viewing the categories by using the group by. Then removing the empty data& short and later combined the headline and short description with the space. Calculated the max length of words for padding the Data. Later the category variable converted into ID. Thereafter glove embedding to remove the duplicates from getting the Stanford library words using inbuilt function. And splitted the Data into training and Test data.

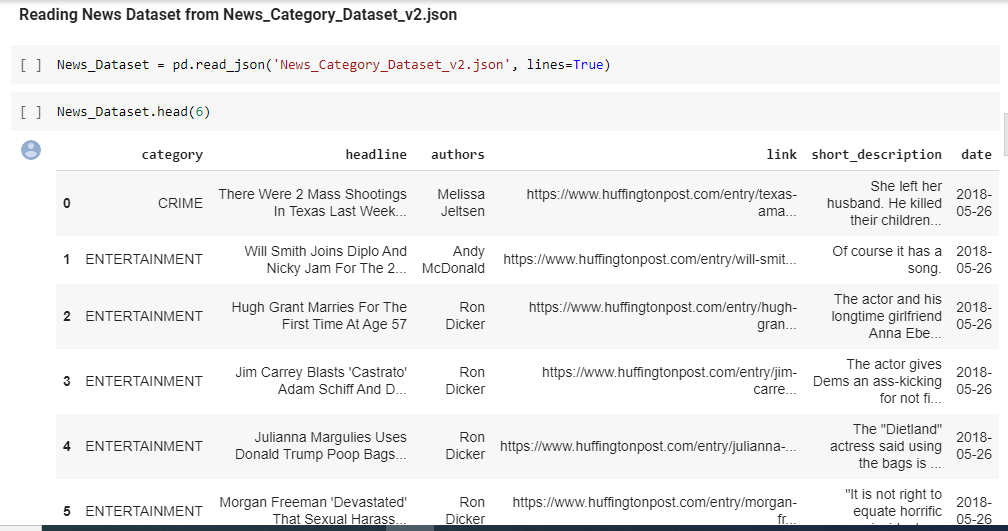
Step 1:

For this we started with importing the packages:



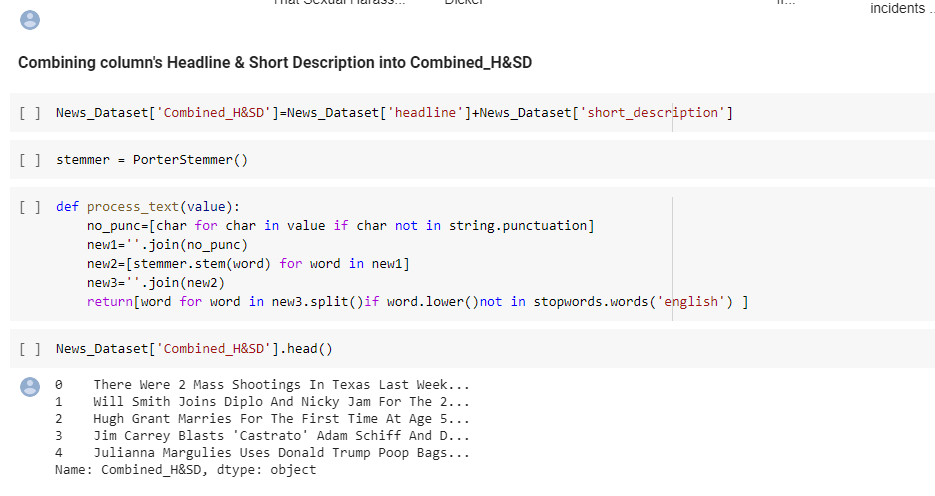
Step 2:

Here, we are reading the news dataset from the json file we have and viewing the sample Data with head function



Step 3:

Combining of column’s headline’s and short description into a single attribute which helps to get the sufficient data for prediction of Category.And the Combined headline is cleaned using stemmer and process\_text() function



Step 4:

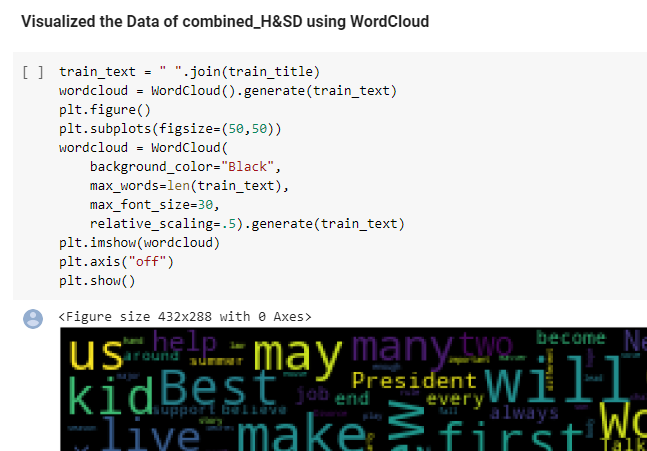
Splitting of the Dataset into train, test and development.

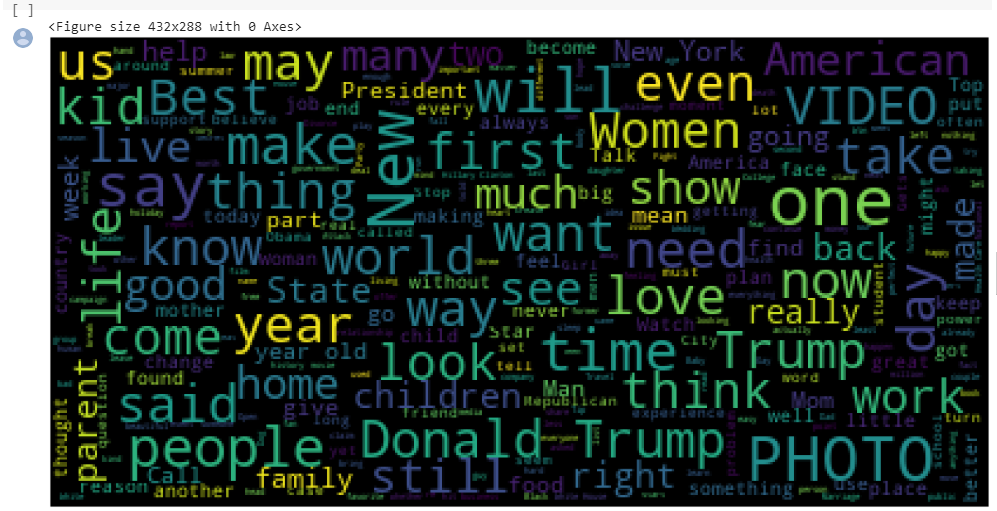
Training data is used for training out the model and Development data for tuning and checking the hyper parameters and test data to check how the model is performing.



Step 5:

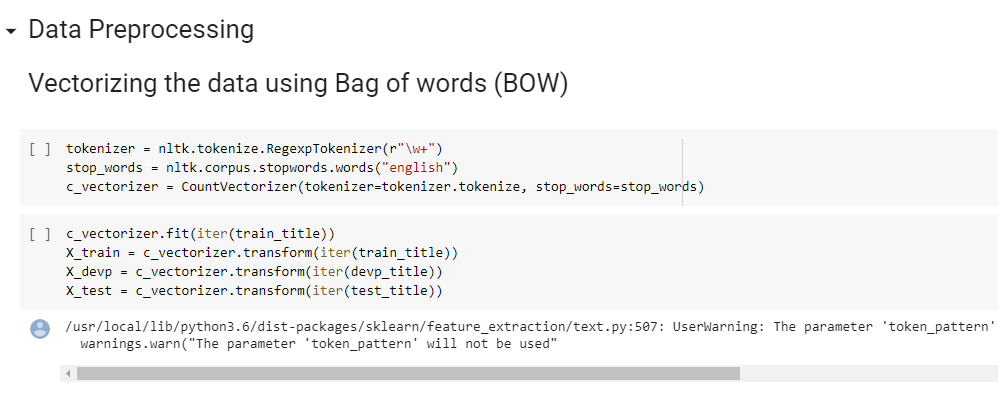
Visualized the data combined using WordCloud which gives the unique words of training Data which helps for the Prediction category





Step 6:

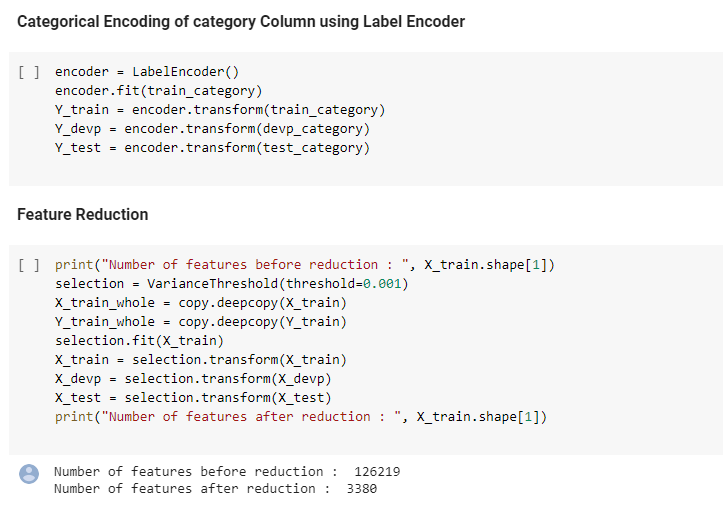
The next step is Data Preprocessing. Here, we are vectorizing the data using a Bag of Words (BOW) and we performs tokenizer on the train, test, Development.



Step 7:

Encoding the column categories are done using the label encoder for all the categories. After that the features are reduced.

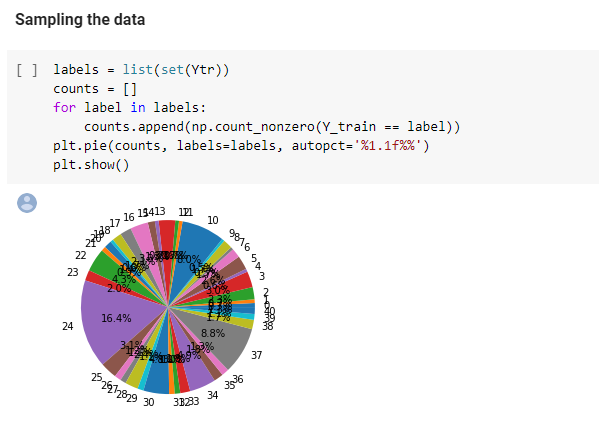
If we clearly see the features before reduction are 126219 and after reduction are 3380 The threshold which we took is 0.001



Step 8:

In Data Sampling,

We have counted the number of total labels and plotted them using a pie chart distribution model.



Step 9:

We can clearly have a look that the class labels are here not distributed uniformly.

So, we had to use SMOT and then over sampled the classes which are lowest in the number. This is done because we can samples can be equally distributed helps for efficient prediction of category.

Step 10:

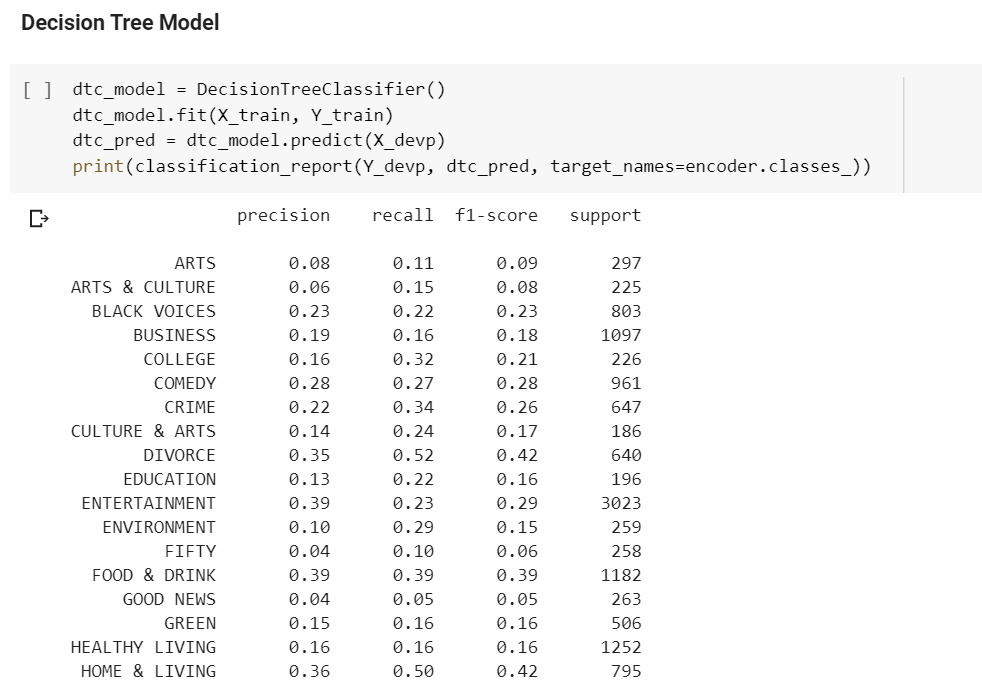
**Model Training:**

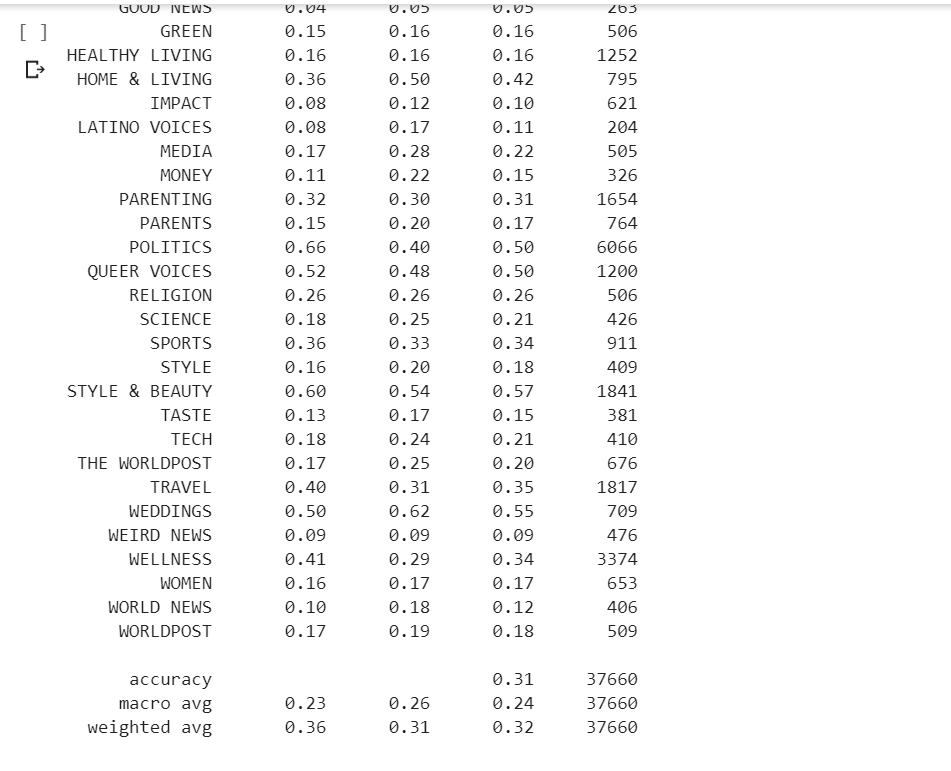
The following are the models we used to train our data:

Here we performed all Machine learn models Decision Tree, Random forest, SVC, Multinomial Naïve Bayes comparing all these accuracies we got High for SVC with 54 % and Multinomial Naïve Bayes with 52 %. So, we can select this two:

**Decision Tree:**

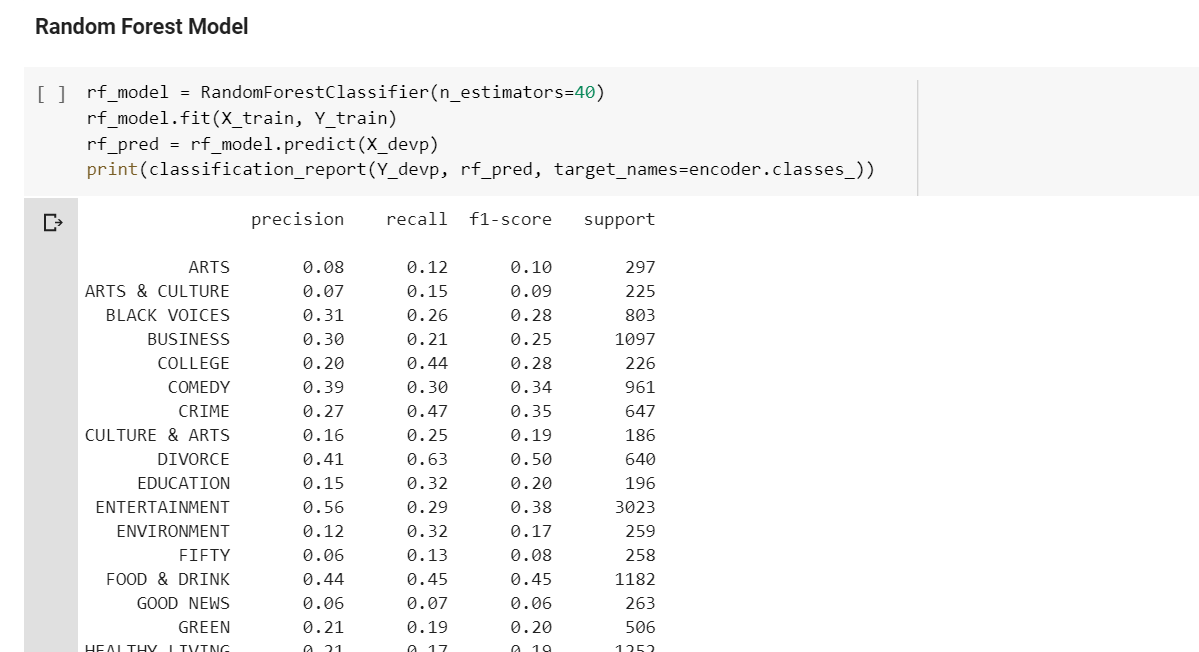
In decision tree model for every we can observe the every category the precision was less so we can say the accuracy was low with 31%.

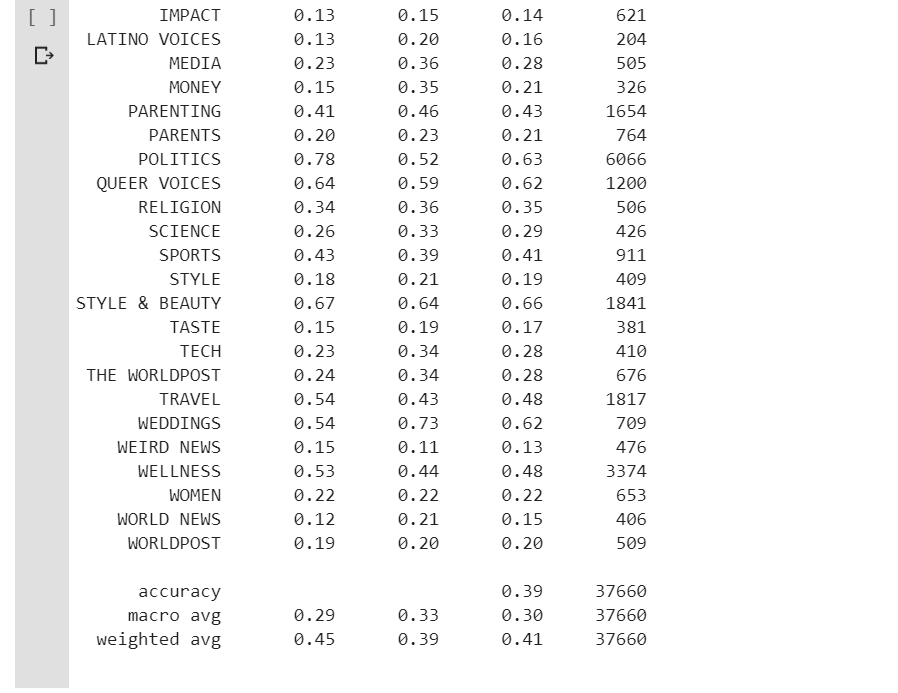




**Random Forest Model:**

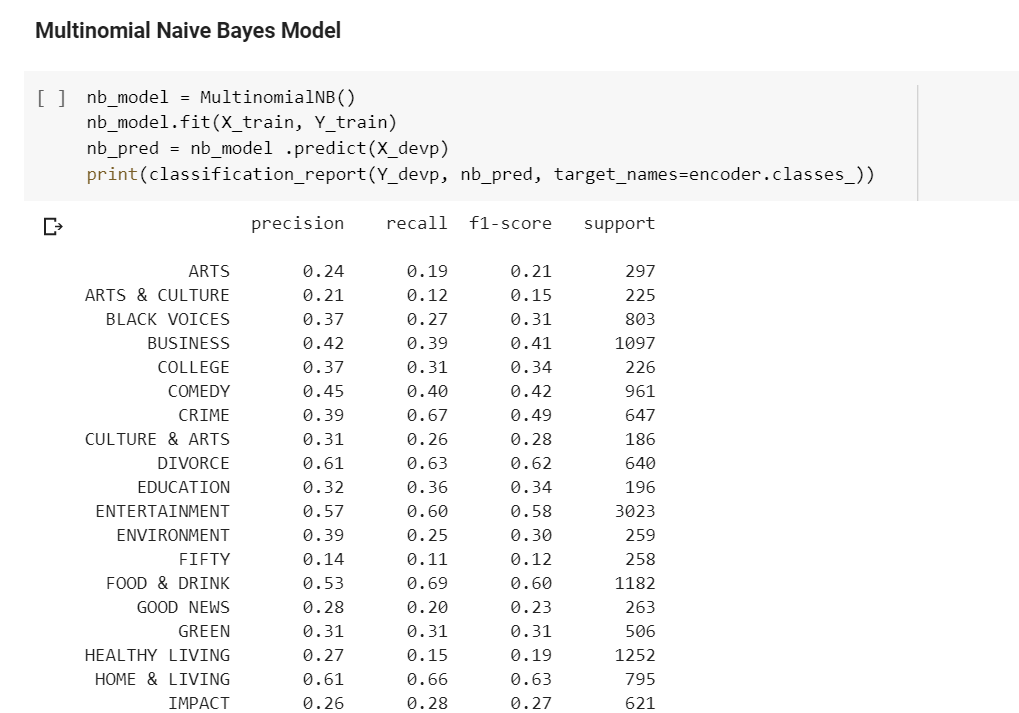
In Random forest model for every we can observe the every category the precision was less so we can say the accuracy was low with 39% so we reject this model.

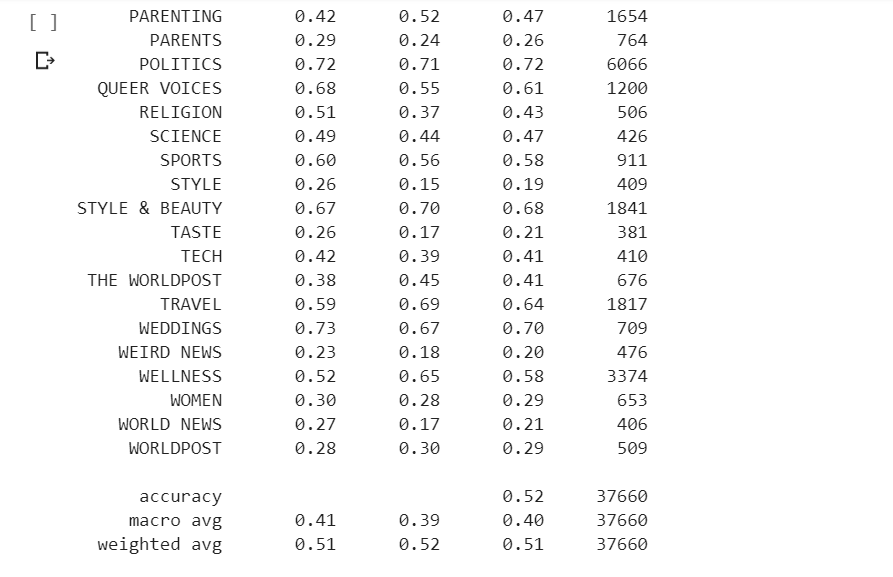




**Multinomial Naïve Bayes Model:**

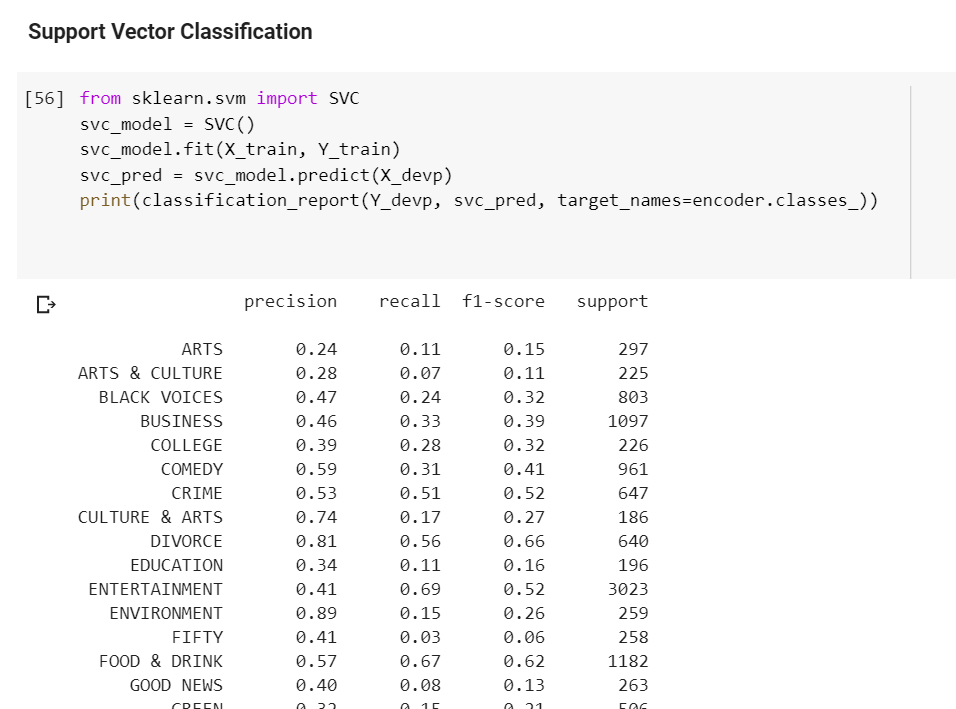
In Multinomial Naïve Bayes Model we got accuracy for development data with 52%. So later we check with test data where we got accuracy with 54%. According to this we can say that this model is better than previously mentioned models.

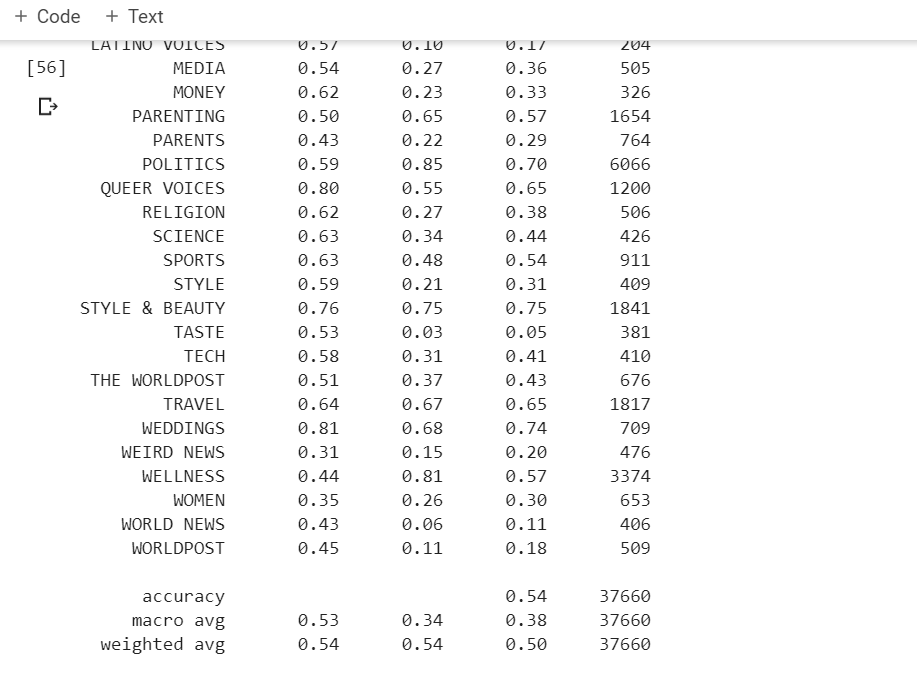


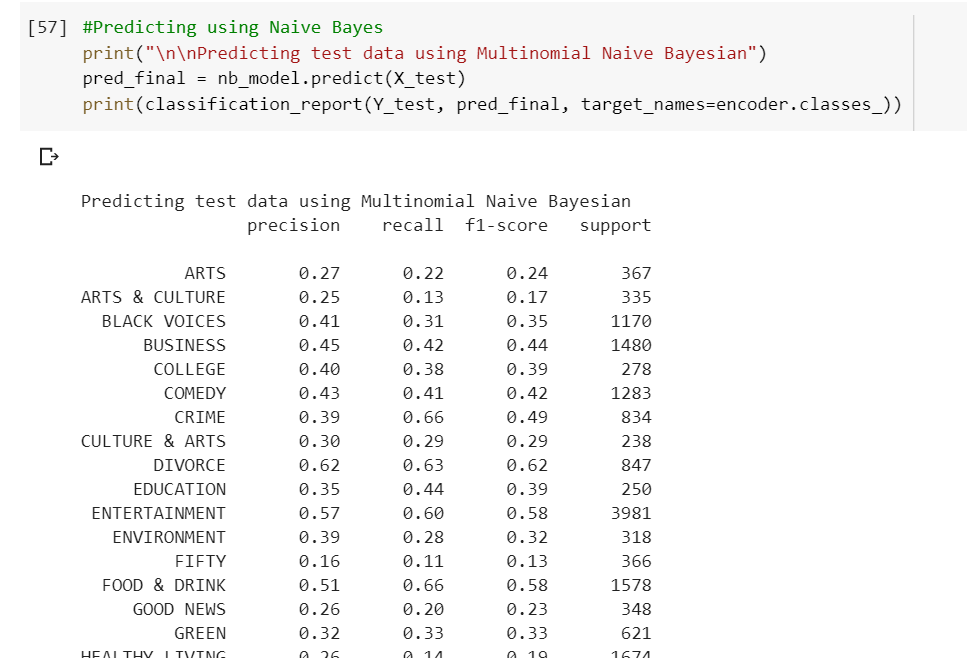


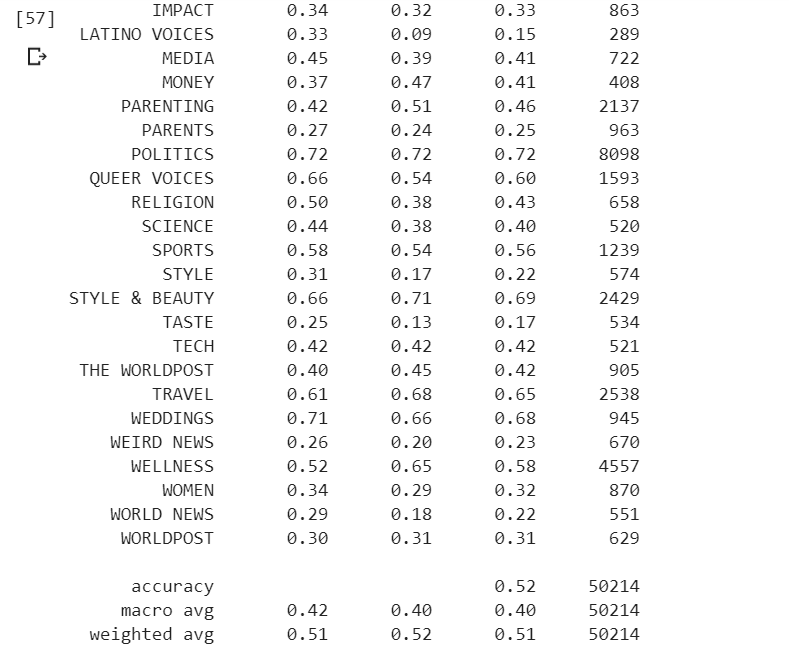
**Support Vector Classification:**

Comparing with Random Forest model this model also have better accuracy with 54% with development data. But, while we check with test data we got accuracy with 51%. So we finalised multi-nomial naïve bias model from the machine learning models.

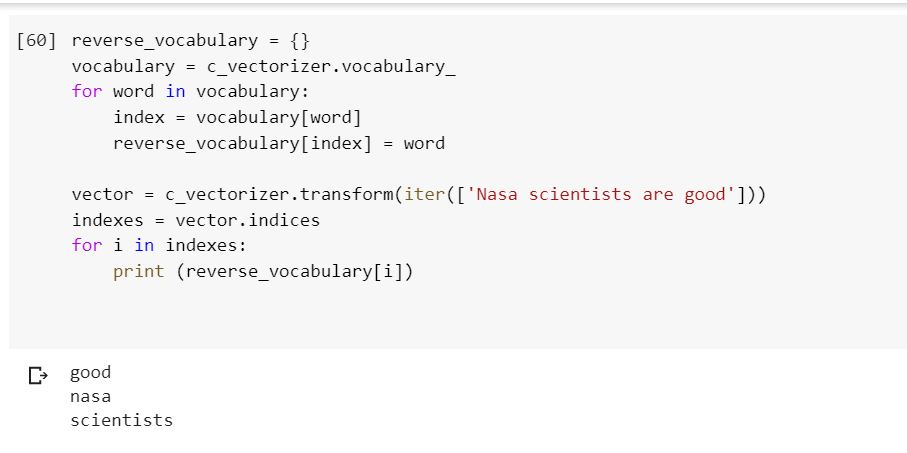








While we use vectorizer from the BOW (bag of words) the order of words will be not saved so we use reverse vocabulary function to save the order.



Viewing the words using the model multi-nomial naive bias for help in predicting the categories.

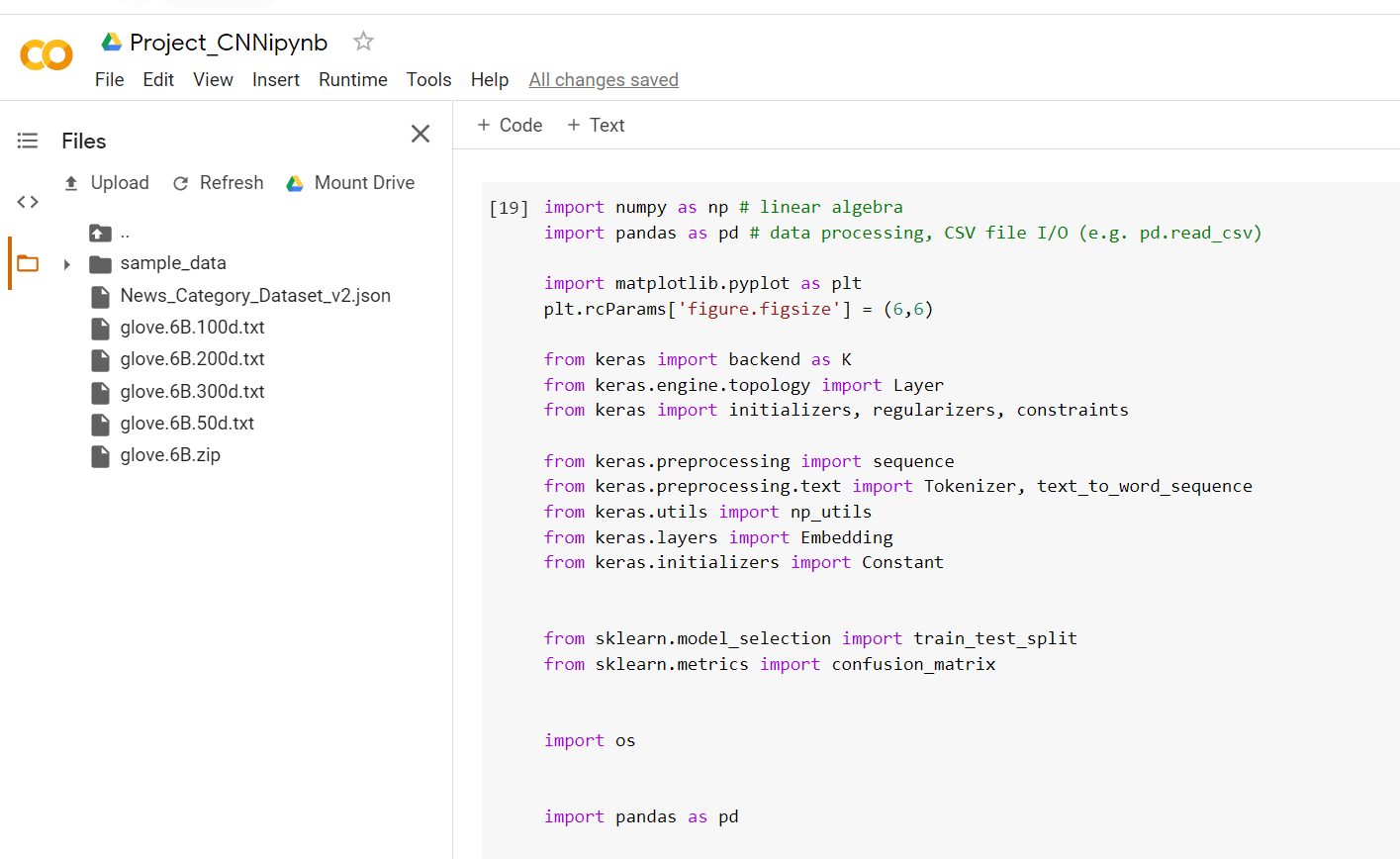


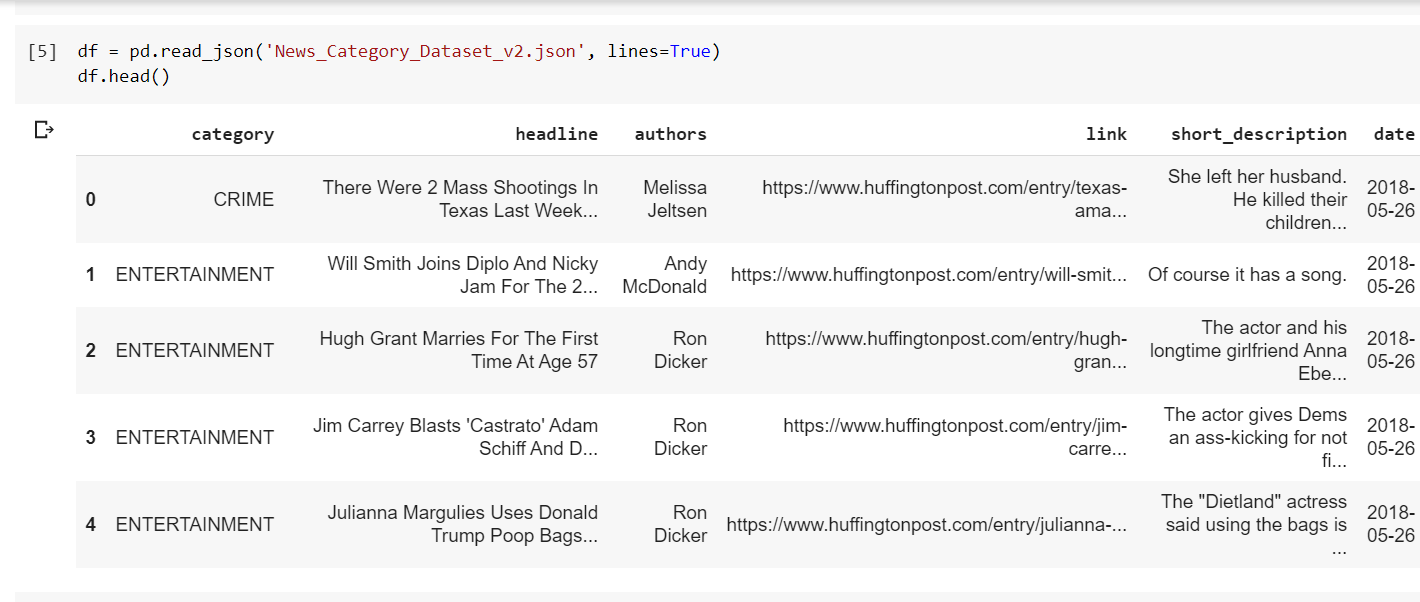


**Step 11:**

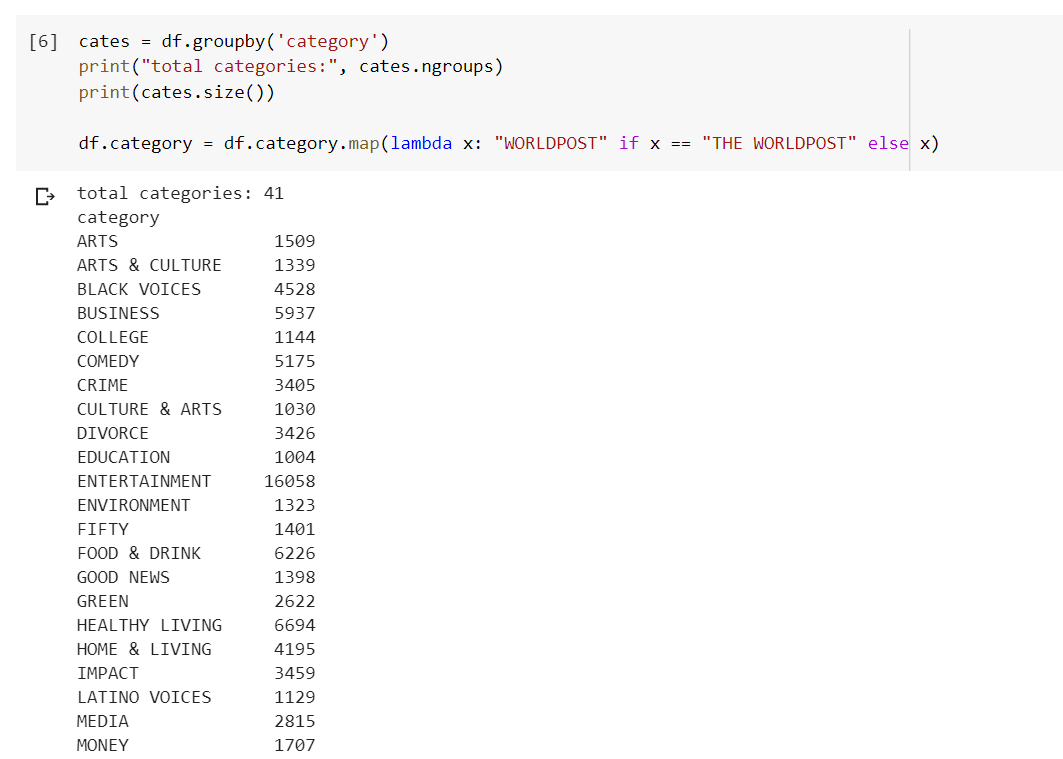
**Deep Learning: Convolution Neural Network**

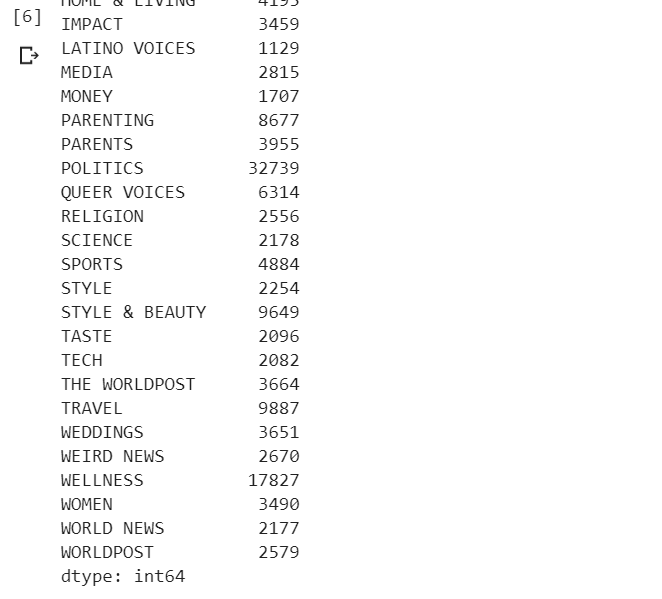
After completion of machine learning models we trained CNN model from the initial stage like data prepossessing.



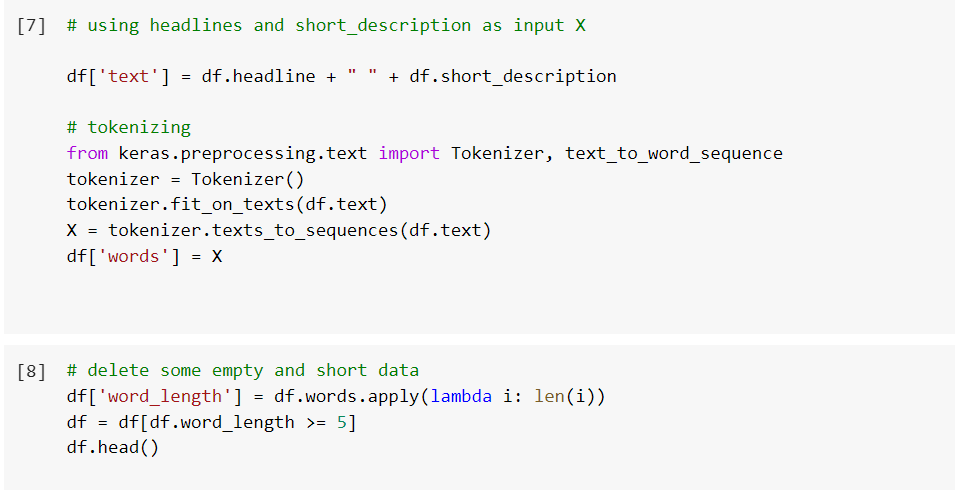


Checking the category types using groupby function

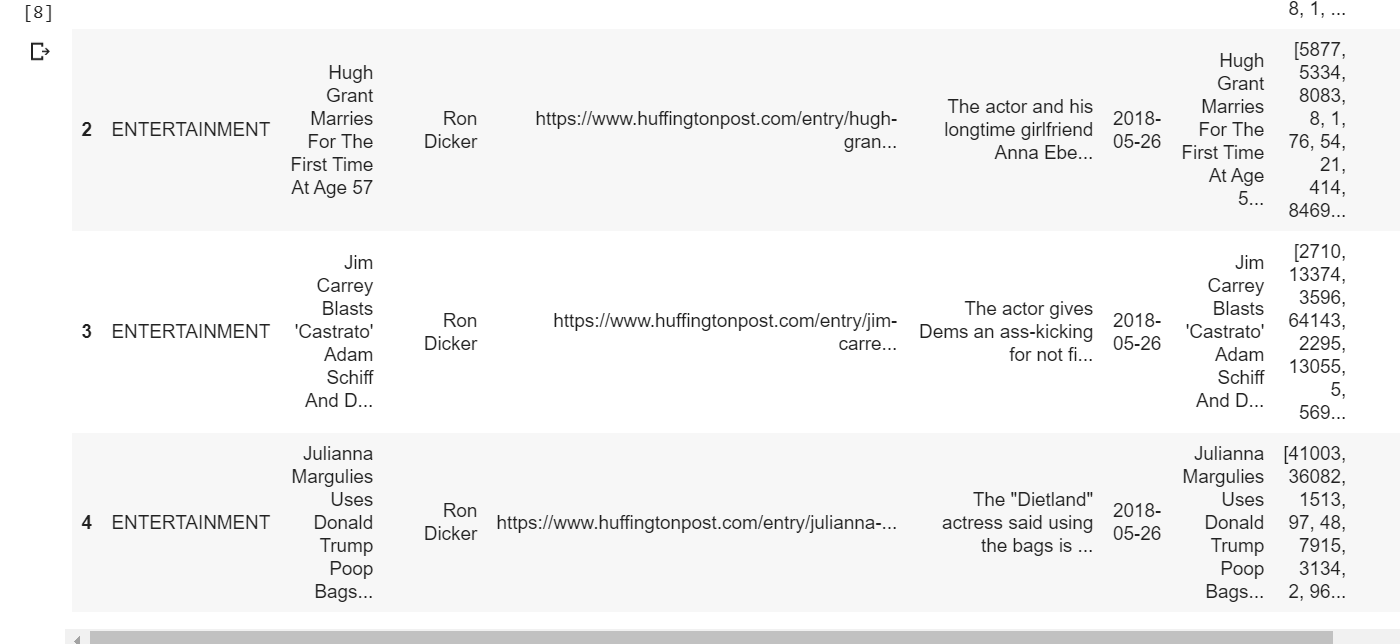




Combining the short description and headline as a single attribute and deleting the sentences with word length less than 5.



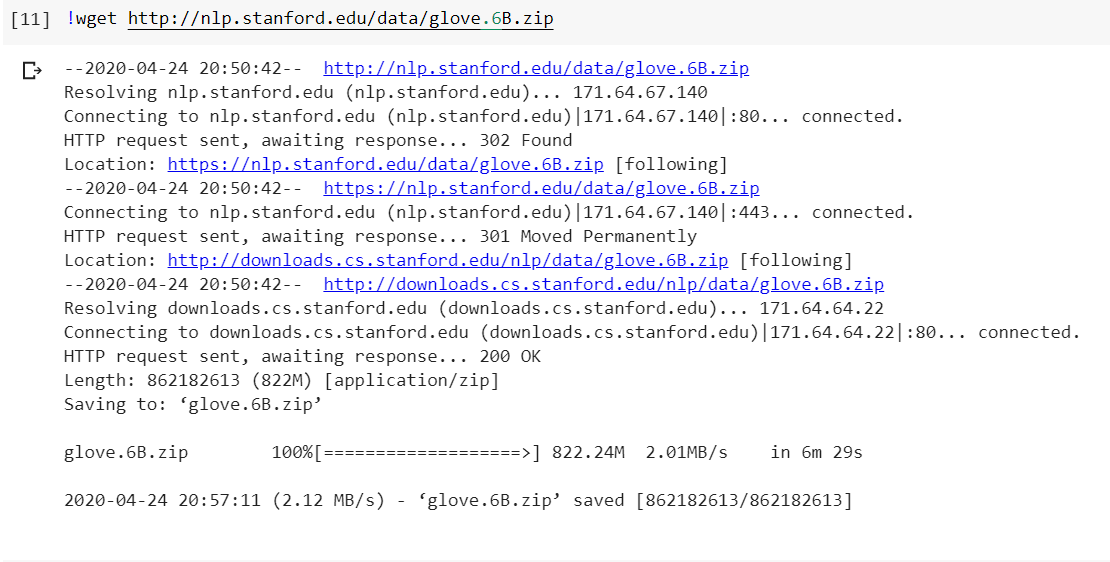




Padding was done with 50



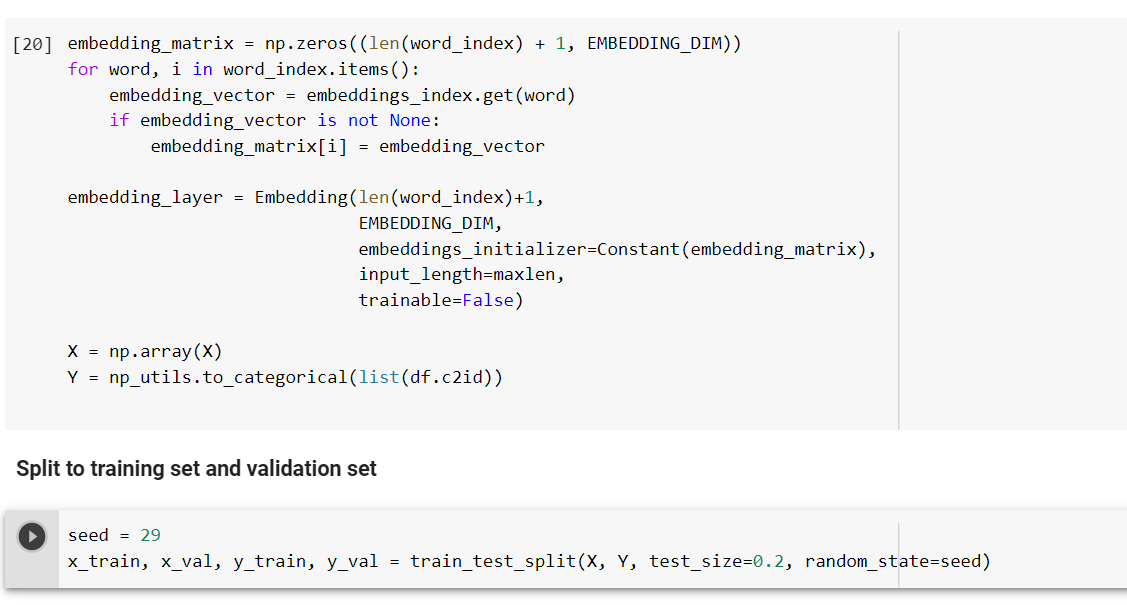
Glove embedding was done from the below link we downloaded the zip file and unzip for the embedding.



Glove Embedding was performed:

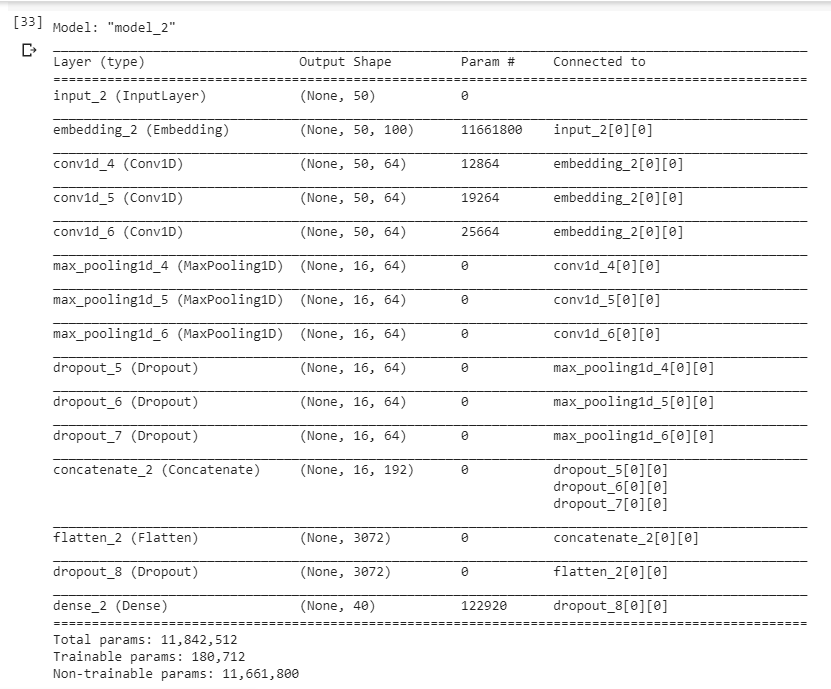


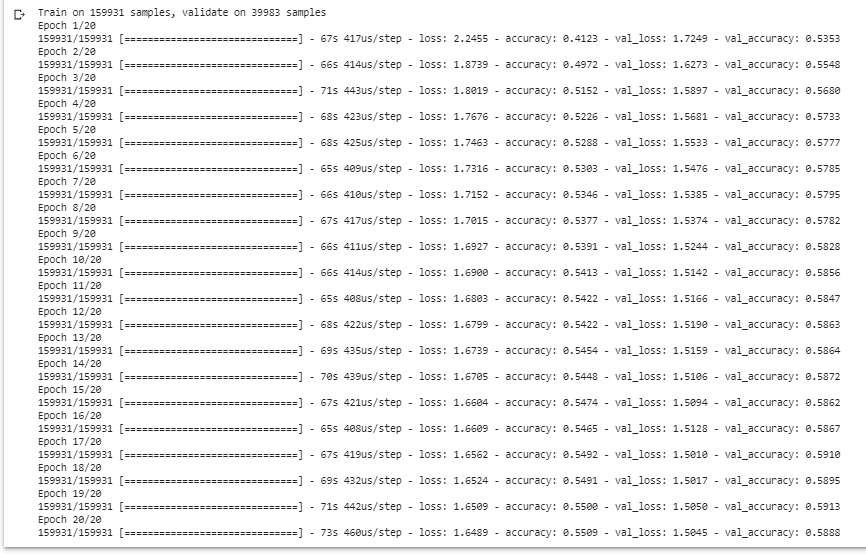
We created the embedding layer with embedding\_dim with length 100 and later we applied array to the x which is input and the category into id.



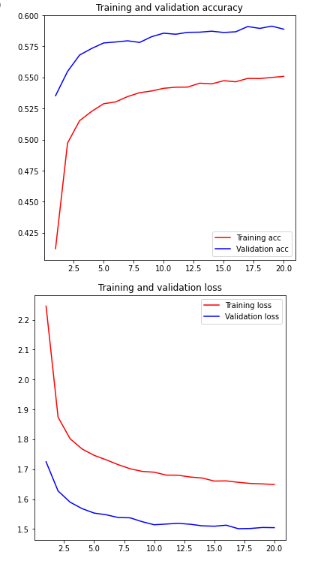
Below we can observe the structure of the CNN model where we can see the embedding layer, dense layer and the output layer.

After we applied the clubbed the headline and short description as single attribute and Deleted some empty and short data with length less than 5. And padding length 50 was applied later converted Categories to ID. Later the tokenization was applied through glove embedding.



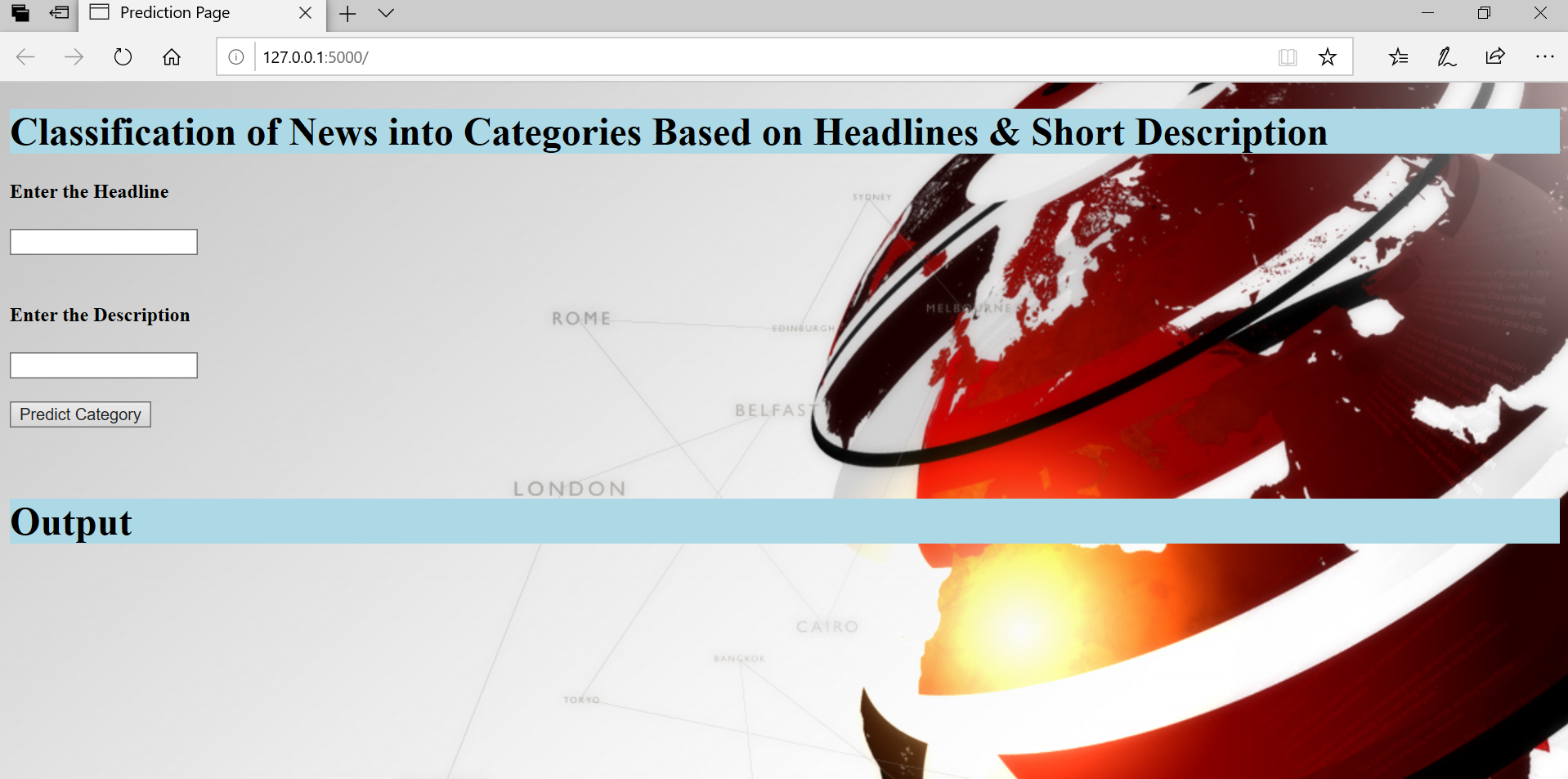


We can observe the epochs and the Validation accuracy & Validation Loss



From the above graphs we can observe the accuracy was increased for the test data compared to training data and same with validation loss we got better compared to the training data. But we can see our validation loss was greater than 1 which says our model was not upto the mark. For getting better low loss we try to change number of layers and change the activation layers but I can’t see the better loss. I got fluctuations over the loss. Because the category data was not sufficient for the training data which created the under fitting. So, our loss was more. Basically we can say that our model was sensitive noise. For rectifying this we try to change data pre processing techniques other than bag of words and glove embedding.

Finally created the pickle file.



**Team-work division:**

* Geetanjali Makineni –
* Dataset Preparation:
  + Combining Column Headline & Short Description
  + Stemming
* Feature Engineering
  + Text Processing
  + Sampling
  + Model Selection
* Decision Tree Model
* Random Forest Model
  + Loss & Accuracy
* Akhil Teja Kanugolu –
* Dataset Preparation:
  + Splitting Train, Test, Development
  + Visualization of Combined H&SD
* Feature Engineering
  + Vectorization
  + Feature Reduction based on Threshold
  + Model Selection
* Multinomial Naïve Bayes Model
* Support Vector Classification
* CNN
  + Hosting Static Webpage

**Challenges Facing:**

Since, we used a dataset with around 200K records, it can take more time to run the models.

More time consumed when pre-processing the data and cleaning it up.

Visualizing the data using Word Cloud is a little complex.

As the Data was Large the Run time was more while running the Machine Learning Models.

Validation loss was greater than 100%

**Future Work:**

* Here, we can also use some other machine learning other than Random fores, SVC, Decision Tree, Multinomial Naïve Bayes as well as deep learning algorithms other than CNN on our data set and we can see which model can give the better accuracy.
* We can here also induce some more other methods in feature engineering as well as parameters and can check how it might affect the accuracy of the model.