**Python/Deep Learning Project Report**

**Classification of News into Categories Based on Headlines**

**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>

**Features:**

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

**Dataset link:**

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

**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 labelled 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’.

**Step1:** Imported All the required packages

A screenshot of a cell phone

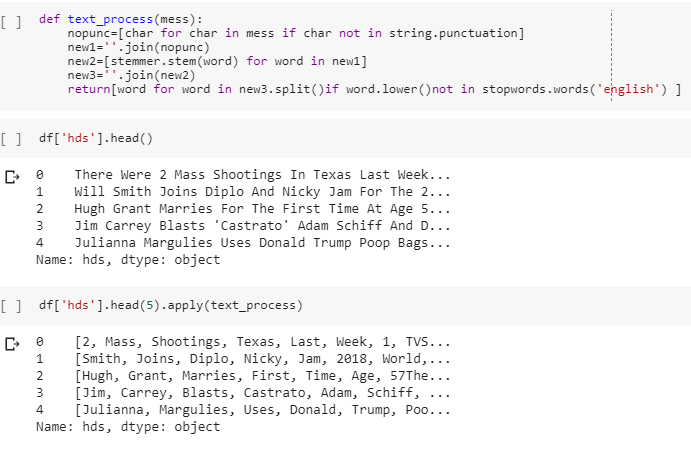
Description automatically generated

**Step 2:** Read the Data from Json file and stored in News\_Dataset and to View the sample Data we used head function.

A screenshot of a social media post

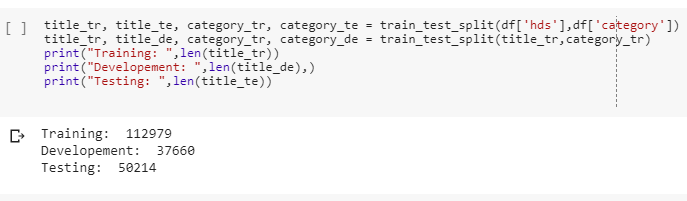
Description automatically generated

**Step 3:** Cleaned the Data using custom created function(text\_process()) and included stemmer to filter the data.



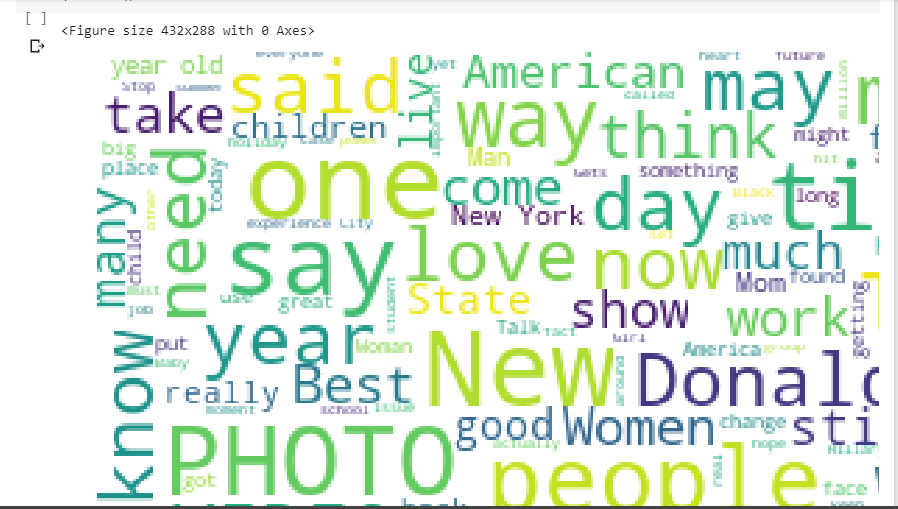
**Splitting of Dataset:**

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.



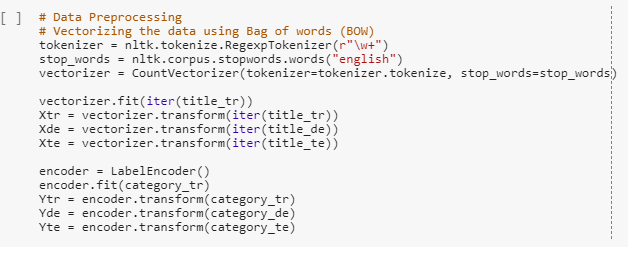
Using wordCloud we visualized our data:



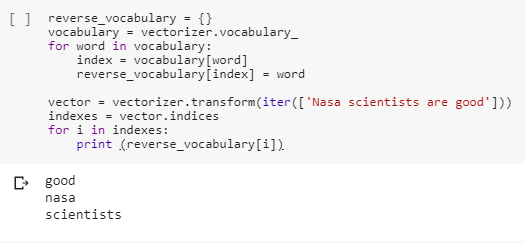


**Data Preprocessing:**

Vectorizing the Data using Bag of Words:

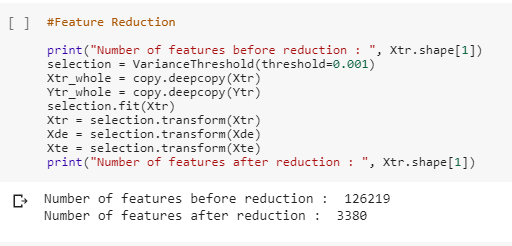


Lets look at what exactly is this vectorizer doing. We will first create reverse dictionary from the vectorizer. Iterating over the vectorized sentence ”Nasa scientists are good”. We get the vector to be representative of three words "good", "nasa" and "scientists". The order has been changed because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



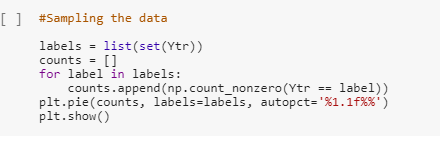
**Feature Reduction:**

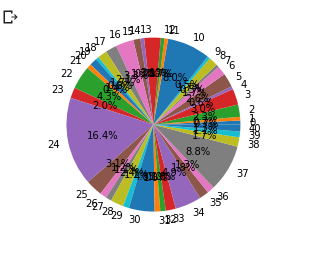
We can check the variance of the feature and drop them based on a threshold:



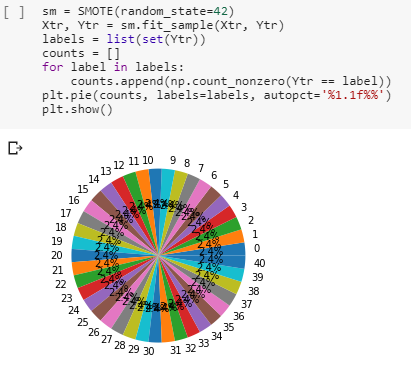
**Sampling of the Data:**

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.

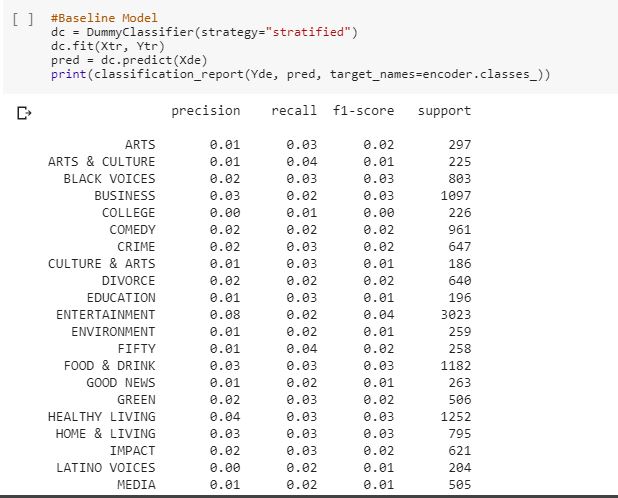


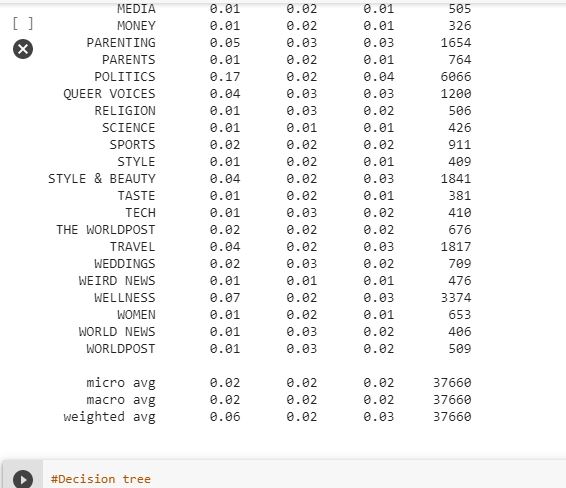
**Model Training:**

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

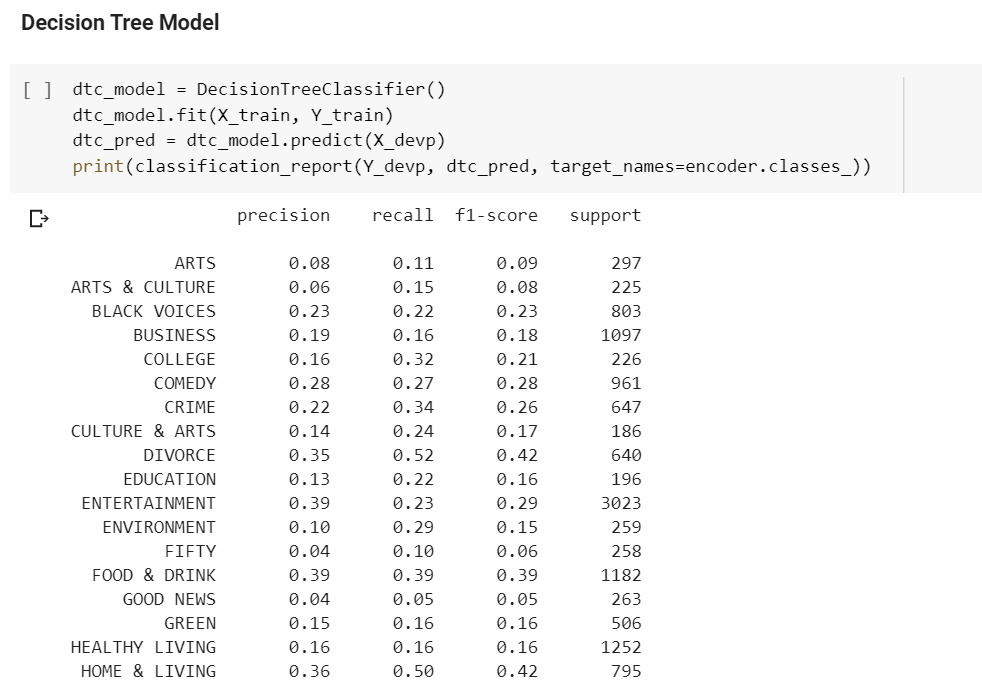
**Baseline Model:**

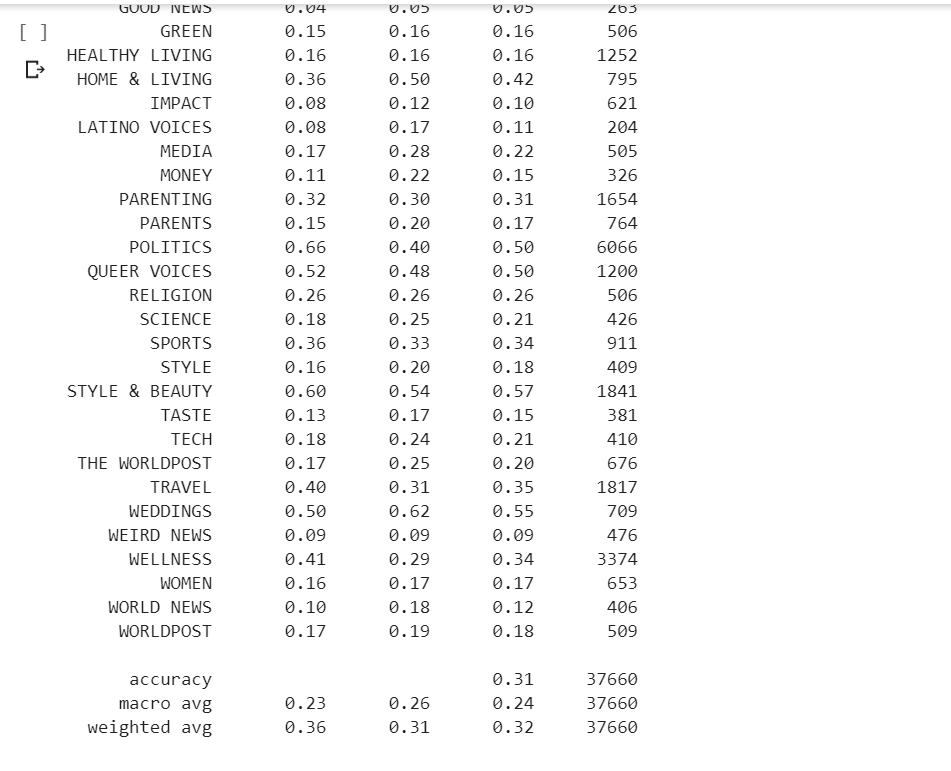
“stratified”: generates predictions by respecting the training set’s class distribution.



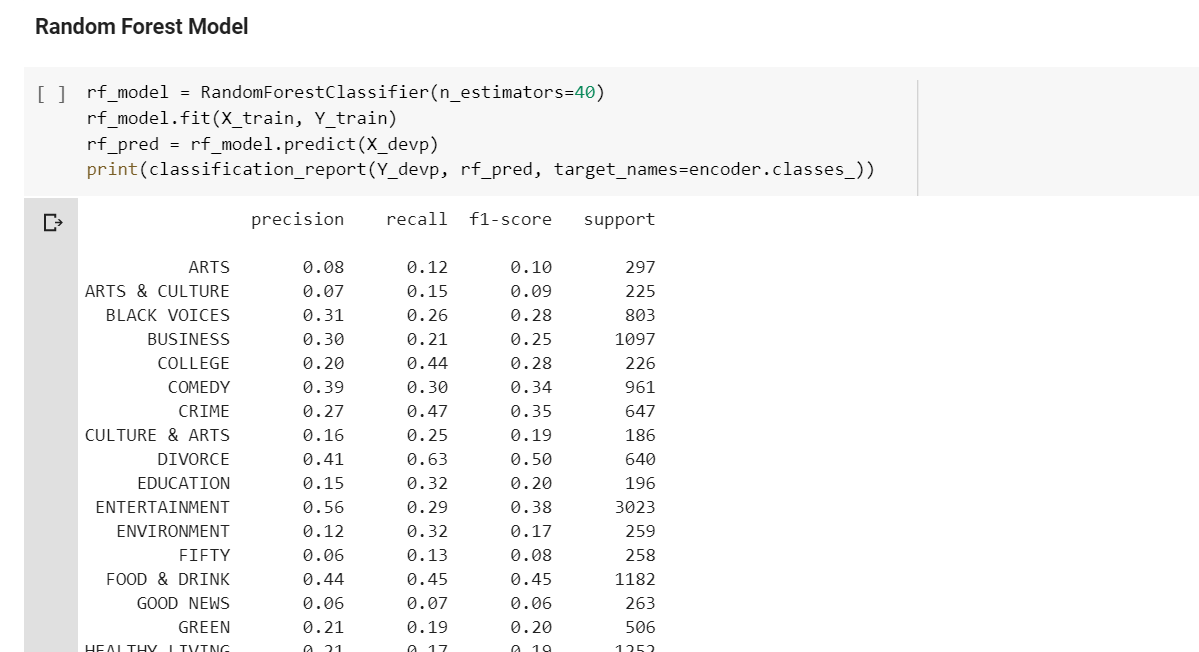


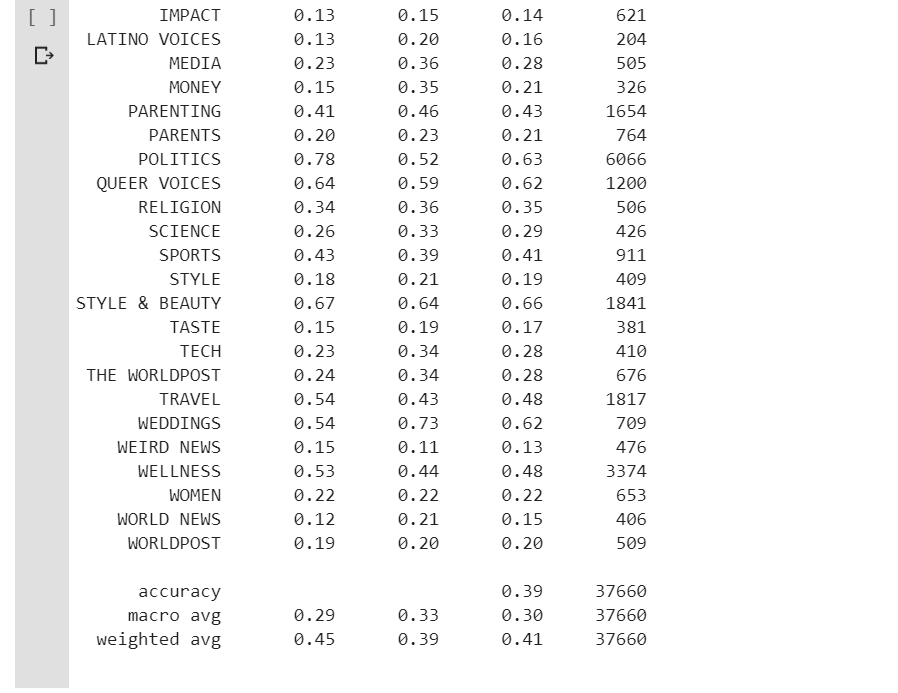
**Decision Tree:**



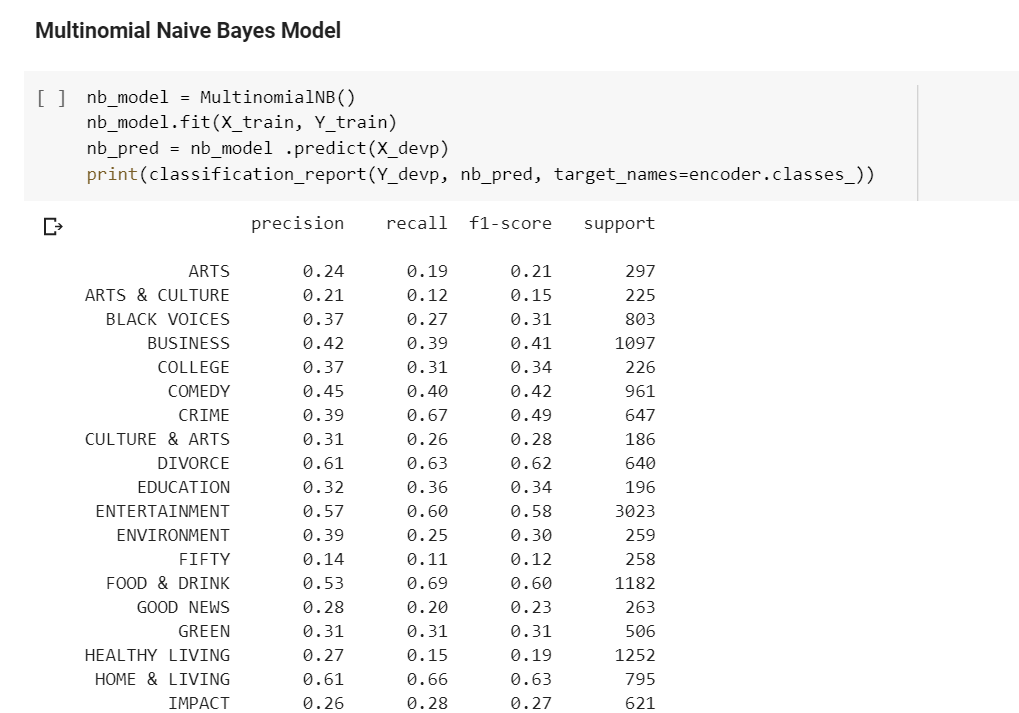


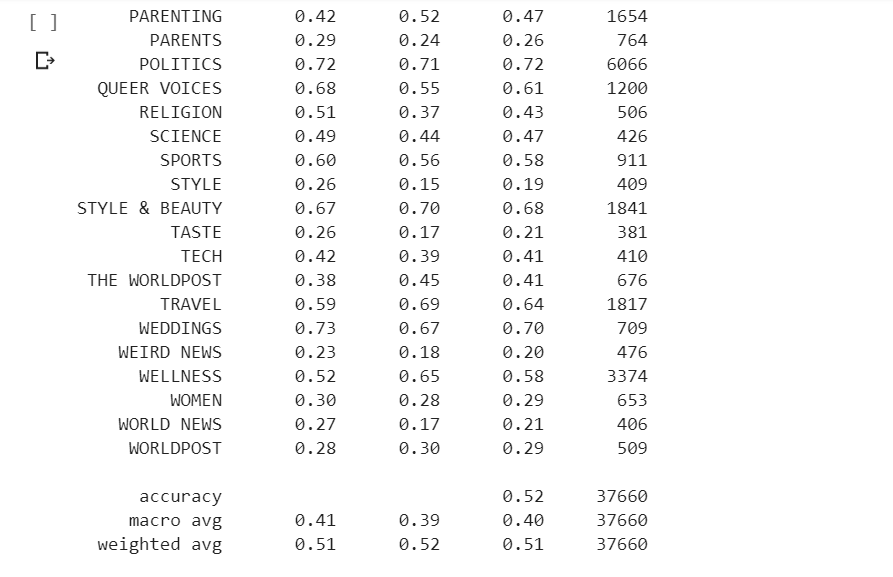
**Random Forest Model:**



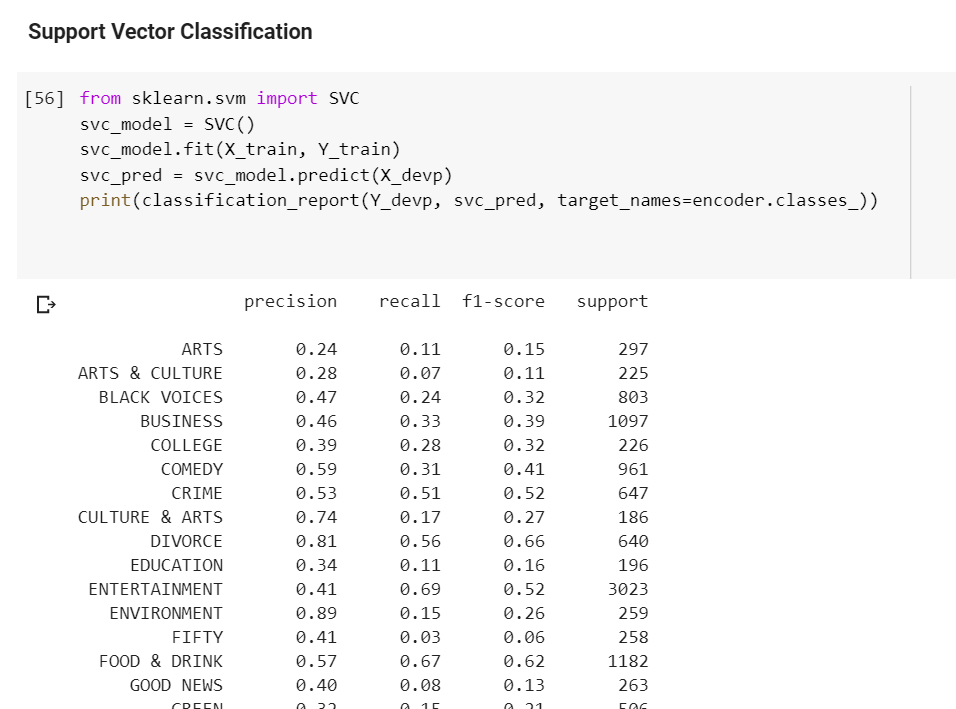


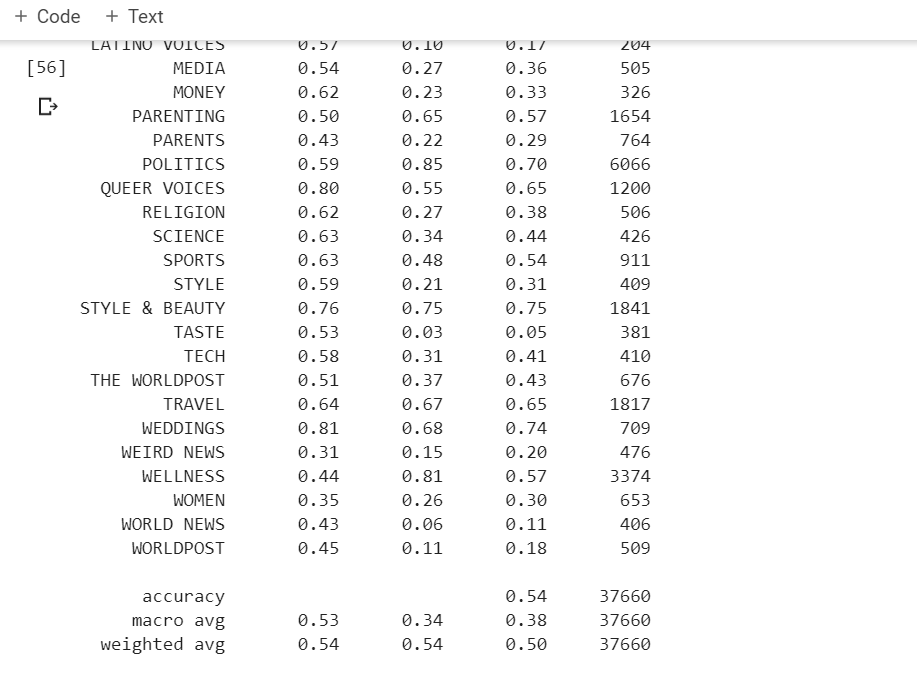
**Multinomial Naïve Bayes Model:**

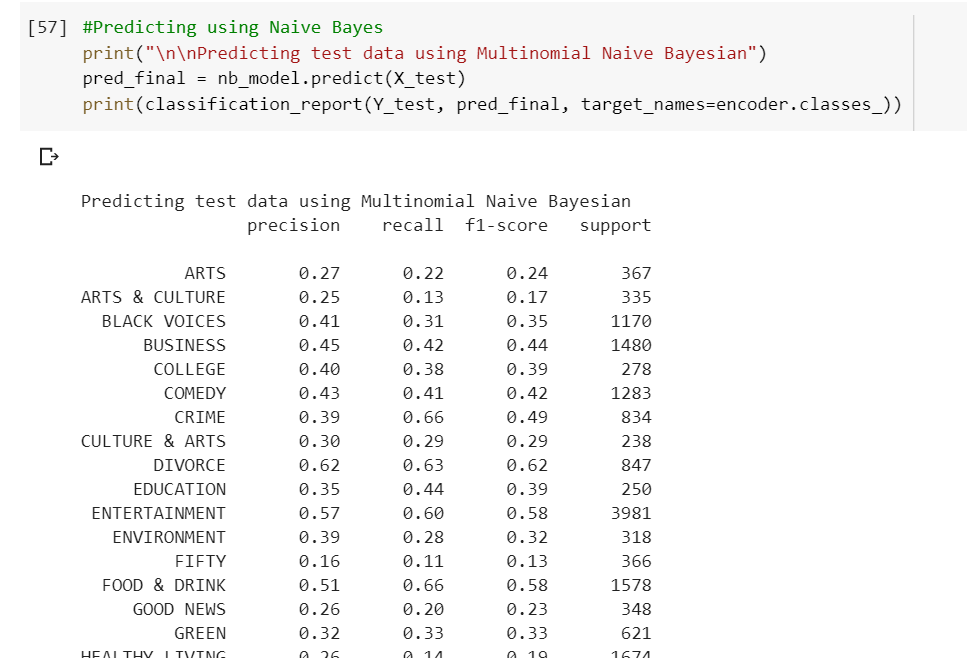


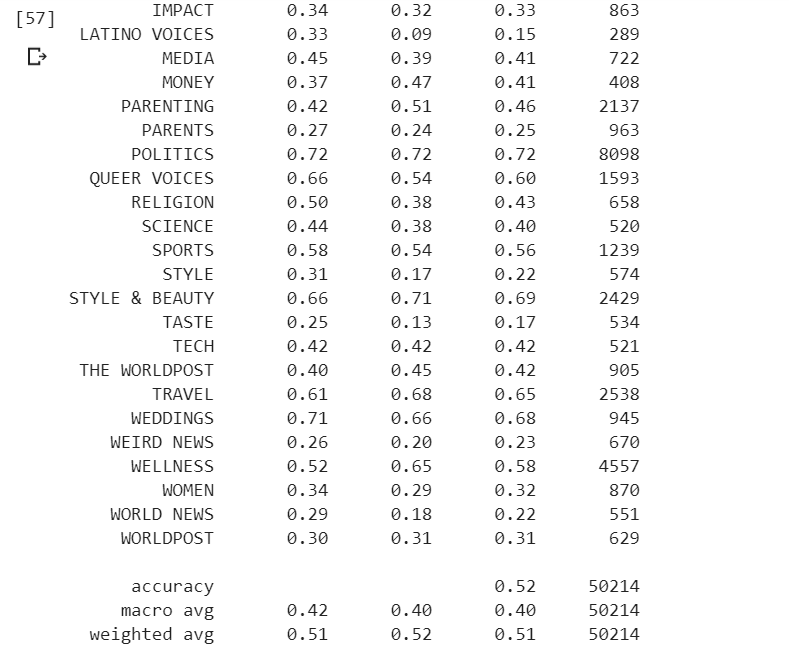


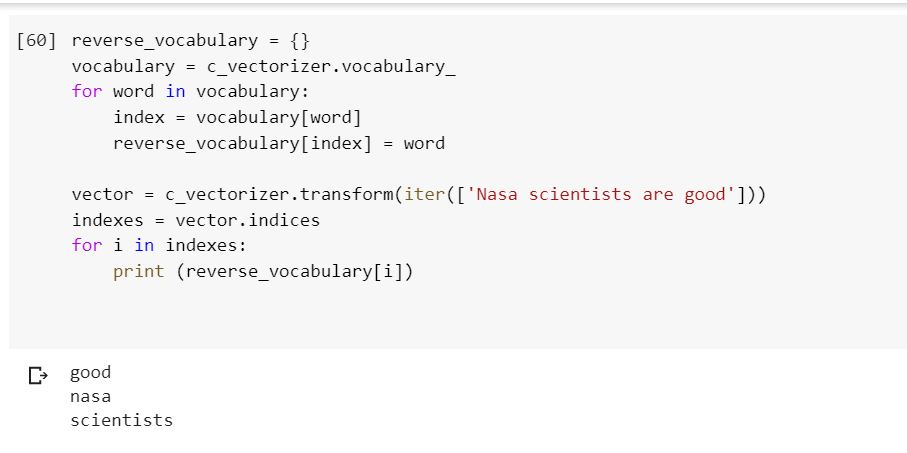
**Support Vector Classification:**





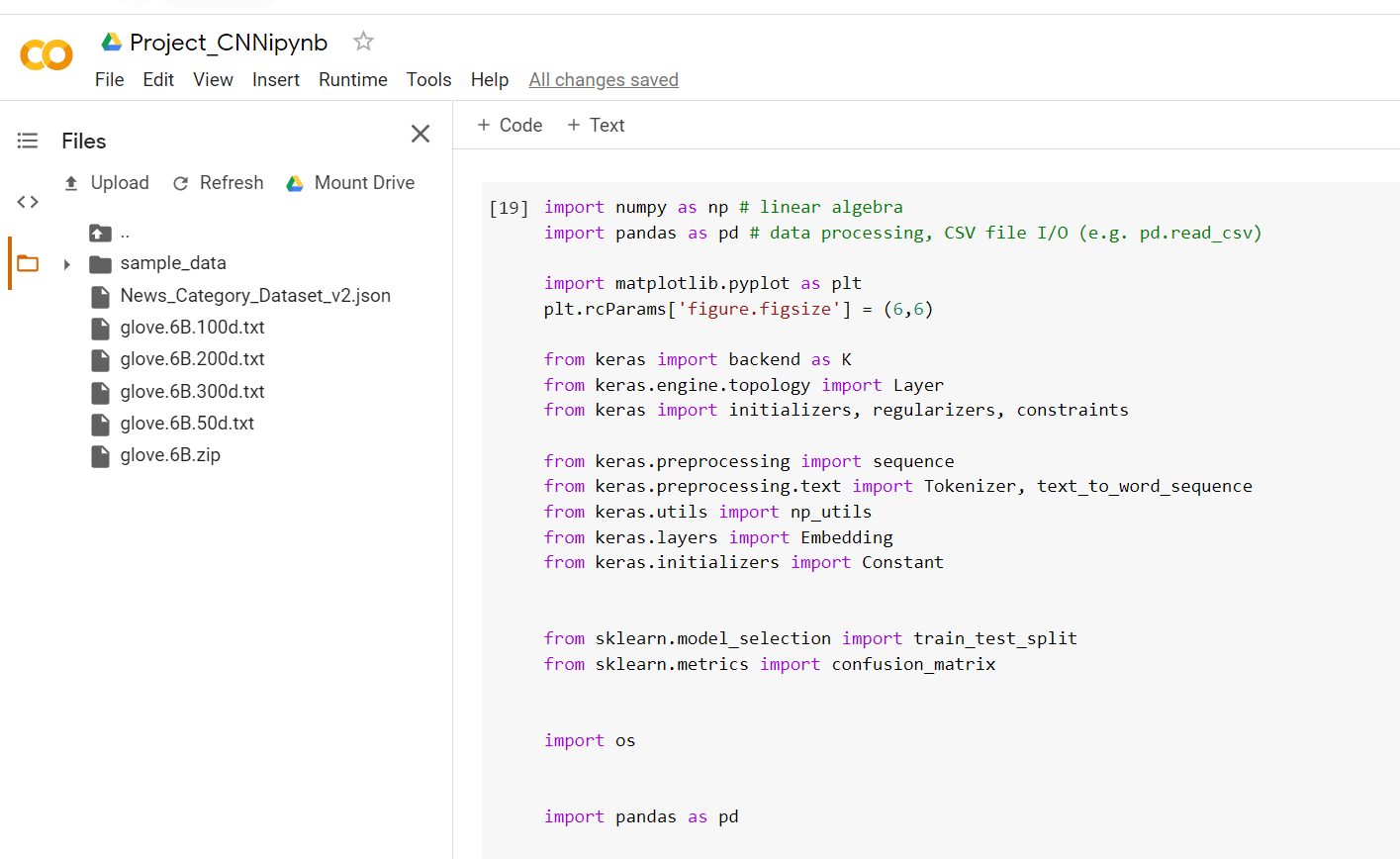


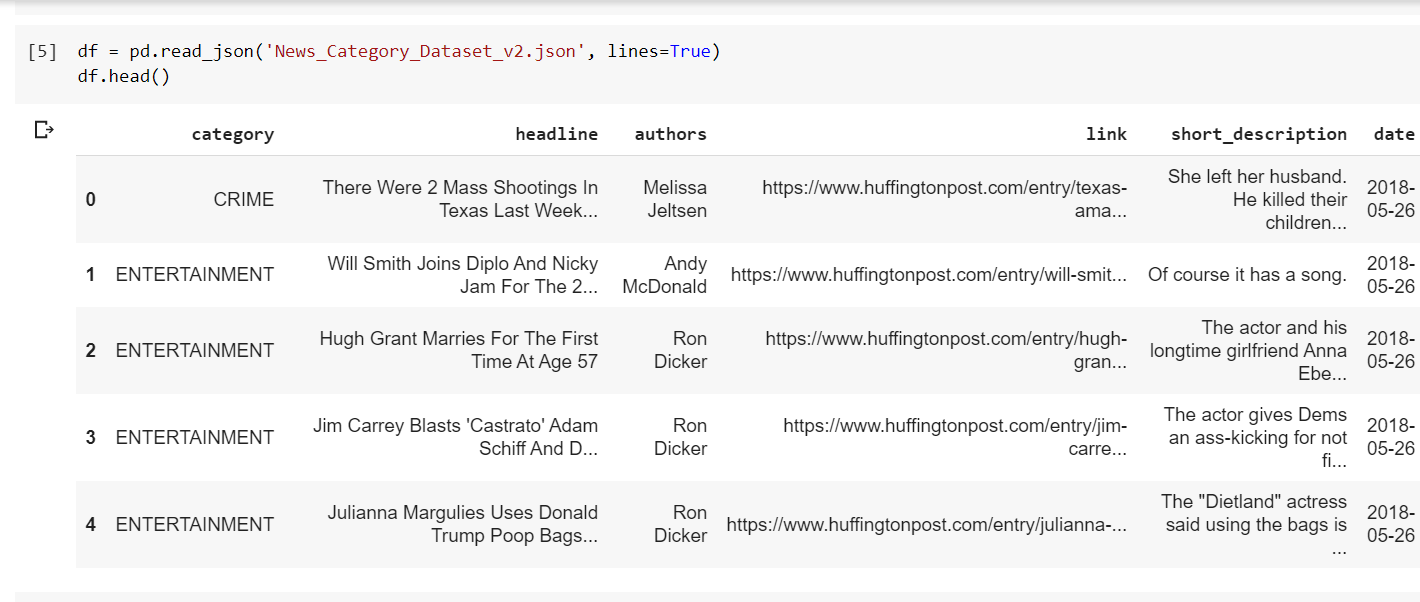


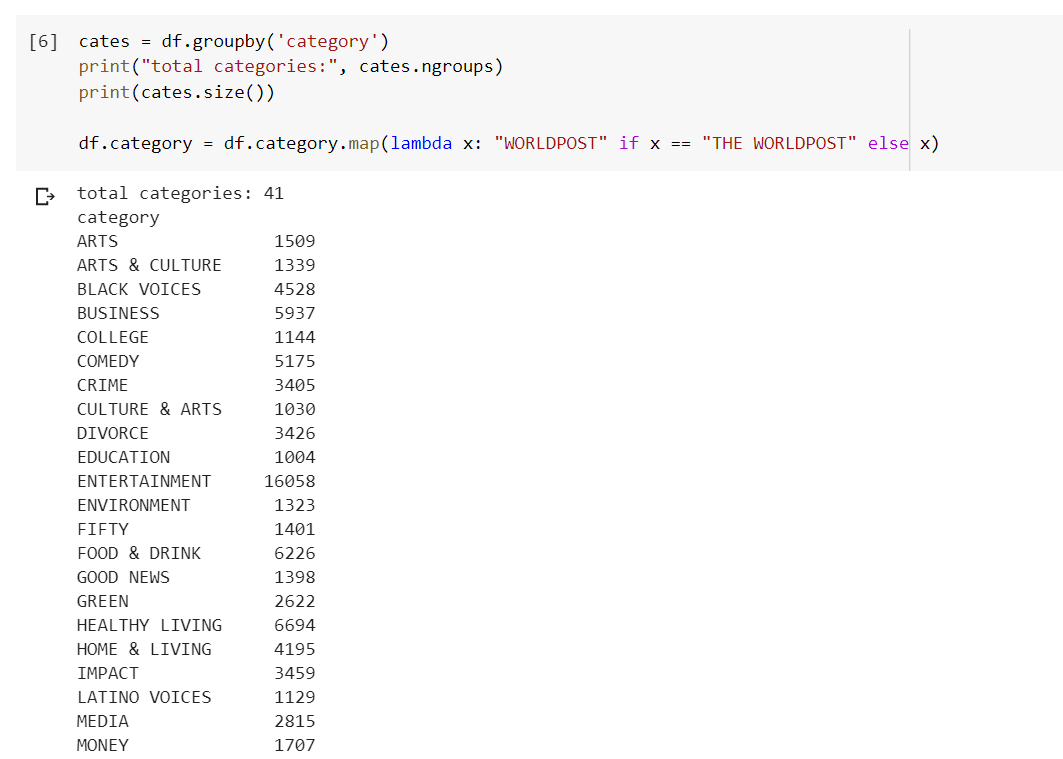


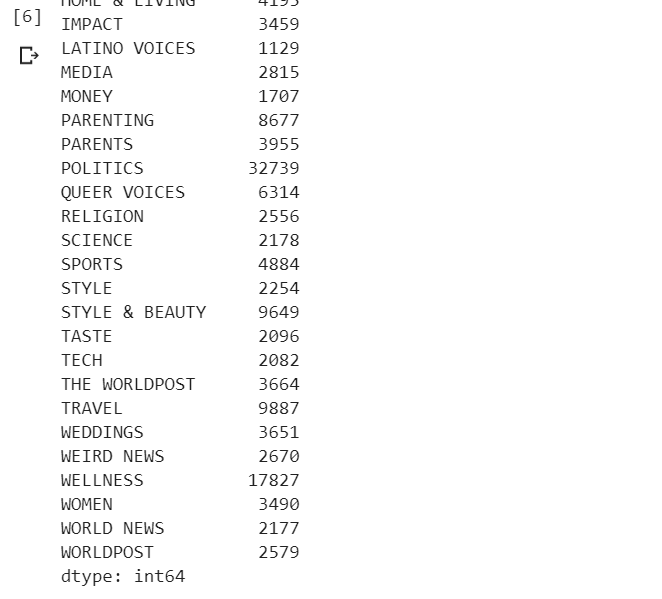


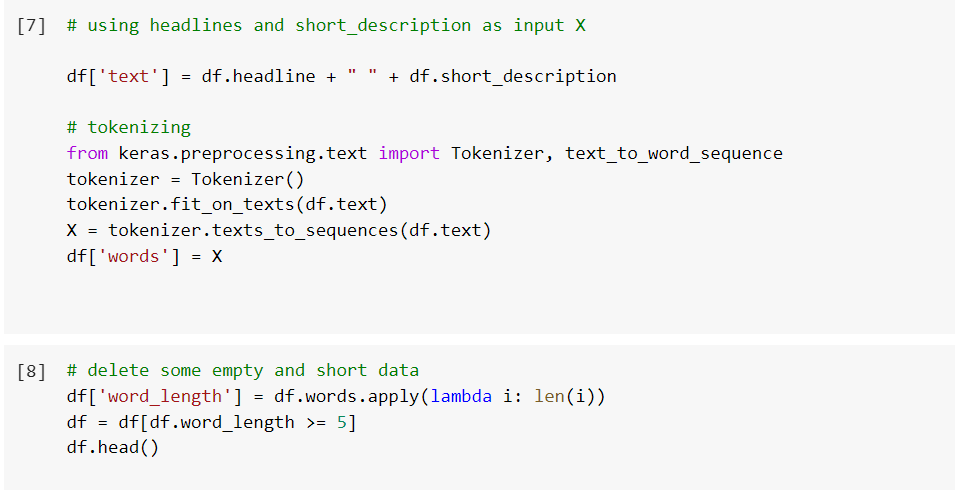




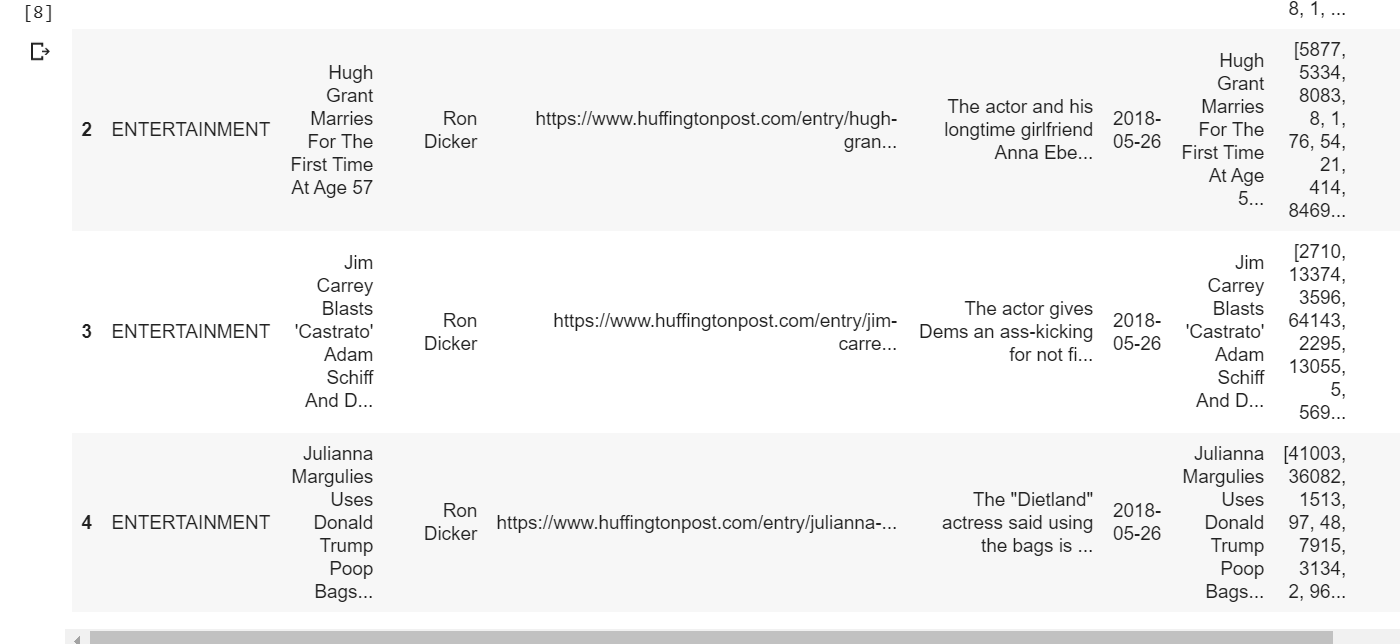




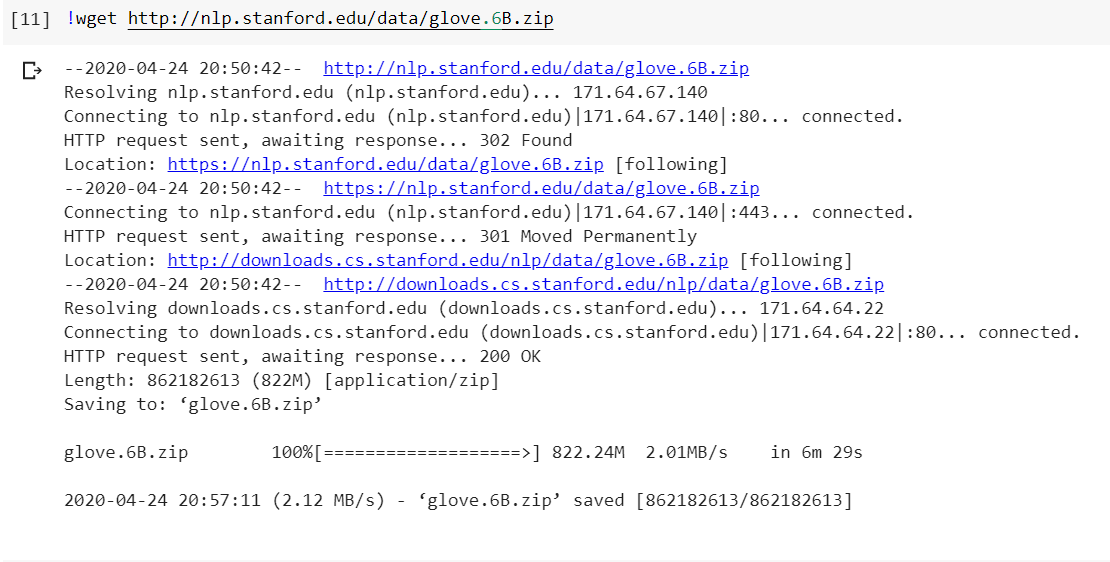




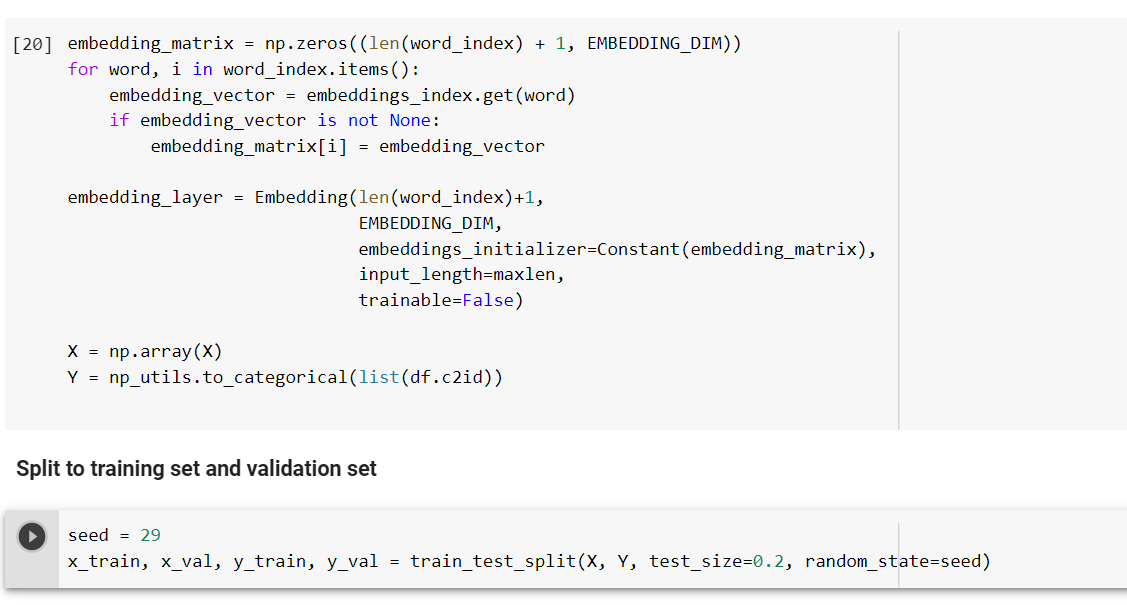












**Challenges Faced:**

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.