Precily Report

Submitted By : Ajay Kumar Kushwaha

Vidyajaykushwaha@gmail.com

https://github.com/kush1912/Text-Classification

+91 8349649515

Language : Python

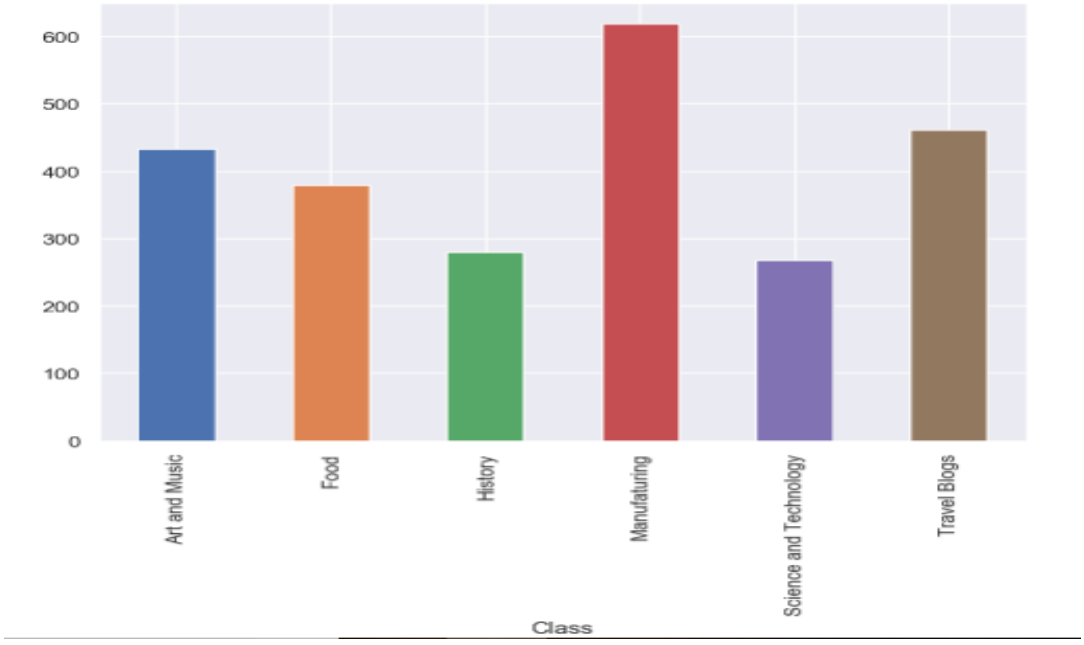
Tools & Technolgies : Spyder, Jupyter Notebook, Chrome driver, google collab

SCRAPING:

- *Website Scraped :* Youtube
- *Libraries used :* Beautiful Soup, Selenium, Requests
- *Data Fields:* Video ID, Title, Description, Class
- Dataset contains 500 entries per class. It was mandatory to collect to 1700 entries per class but I was blocked by google for 1 day due to very fast scrapping. Hence only this entries are possible. Task was to demonstrate handling Big data which is quite visible

I have used Beautiful soup library of python to pull the information of html page of youtube. Youtube does not gives the full video length on the first search page when you scroll down it shows the list of other videos. Hence it was a *challenging* to get the list of videos in that big amount.

To overcome this challenge I have used chrome web driver with selenium and used a function which firstly loads the page by scrolling it 150 times after a period of one sec. And then it receives the list of all the videos. You can get video ID, Title, from there but to get the full description you have to go to the link of that video and then scrap description from there.



DATA CLEANING:

Since the Description had a lot of noise in the form of html tags, number, urls, etc.

- Html tags removed
- Urls, email ids, and all kind of links removed
- Numbers removed
- Punctuation removed and the whole text of title and Description converted into small case.

After carefully observing the cleaned data :

TF-IDF has been used as feature extraction method. TF-IDF score represents the relative importance of a term in the document and the entire corpus. TF-IDF score is composed by two terms: the first computes the normalized Term Frequency (TF), the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

I have applied the TF-IDF vector to both the fields, Title and Description, and extracted their features. TF-IDF was a choice because the content of the title was very small which didn't give any substantial meaning in most of the observation. Hence bag of words was an obvious choice, as it does not provide any information about the semantic relation between and only focuses on lexical features.

Similar is the case with Description of the video where we see that most of the description text does not contain any information regarding the class of the video. In fact there were many entries which did not even have any description. Most of them had word like subscribe, comment, like etc. which did not help us in anyway rather lead to biasing in the model. So bag word technique is used where features have been given weight according to their frequency in the whole documents.

Model Selection :

Category 1: SVM

- Linear classifier can not be selected because it is good for **binary classification** which are linearly separable. But this is a multiclass classification, hence it is not possible to linearly separate the classes accurately.
- So, our choice was narrowed down to Naïve Bayes and SVM. Both have very different approaches of dealing the problem one of them is probabilistic and other is geometric. While Naïve Bayes treats each feature as independently while SVM separates them by finding the interaction between the support vectors. And also we considered the TF-IDF and in most of the cases SVM is chosen with that.
- Since our featured were not independent of each other SVM was a considered choice which gives us accuracy of 85% with Description and 98% with titles.
- I have also applied Naïve Bayes where we observe that the difference between both the models is not really substantial because it also yields accuracy of 85% with Description and 96% with titles, which is slightly less than the SVM model.

- Hence Both the Models classify the classed really well. Although Naïve Bayes has an **advantage** over SVM as it takes less computation time compared to the SVM. Which in case of very large data set had been very useful and preferred over SVM
- Confusion matrix of both with description and with title have been plotted with green and blue colour respectively.

Category 2: XGBOOST

- Random Forest is a bagging algorithm. It reduces variance. In such a case, we can build a robust model (reduce variance) through bagging. Bagging is when you create different models by resampling your data to make the resulting model more robust.

- Boosting models are type of ensemble models part of tree based models. Boosting reduces variance, and also reduces bias. It reduces variance because you are using multiple models (bagging). It reduces bias by training the subsequent model by telling him what errors the previous models made.

- Shallow Neural Network gives a training accuracy of 96% but it is after adding 4-5 layers which can not be called a shallow network. Even though with heavy loss. Secondly reducing layers leads to decrease in the training accuracy upto 50 %.

- I have applied XGBOOST where we observe that the difference between both the models is indifference because it also yields accuracy of 85% with Description and 85% with titles, while Random forest gives an accuracy of 85% with Description and 96% with titles.

- Then also we choose Xgboost over Random Forest as reduces variance and bias both. While random forest can't tolerate robustness in the data. It basically merges several tress made narrows them down. A light change in dataset can cause difference in the tree constructed. And it also reduces the chances of over fitting the data

- Confusion matrix of both with description and with title have been plotted with green and blue colour respectively.

Over all we observe that accuracy based on every model is nearly equal to 85% and with titles it varies a lot. But Xgboost can be regarded as the best model or Random Forest Classifier if the robustness of dataset is guaranteed. Because they give almost Equivalent accuracy and also they reduce chances of over fitting by reducing biasing and variance

Category 3: Haven't worked with any of these models till now so can't comment on that.