CAPSTONE PROJECTS

By Saurabh Kolambkar

Introduction:

My capstone projects were based on Natural Language Processing and Machine Learning. The First Project was car popularity prediction in which I implemented Machine Learning and news category prediction had Natural Language Processing.

Problem Statement:

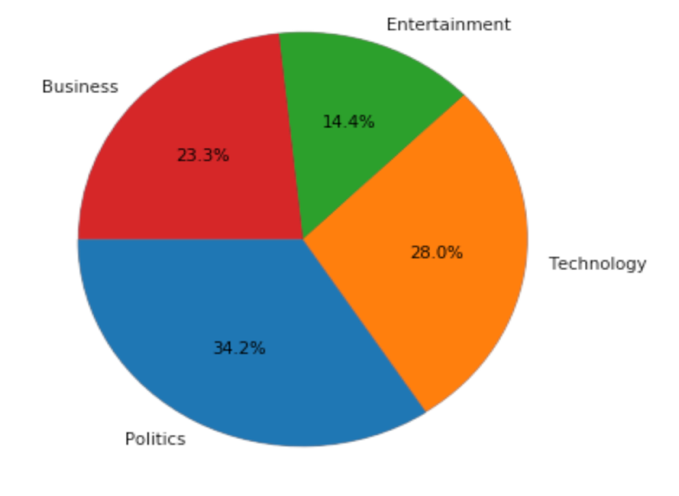
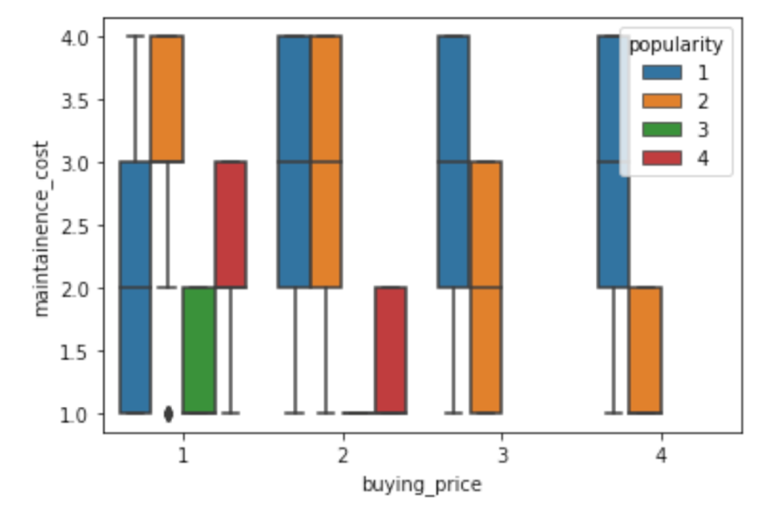
1. Car Popularity Prediction:
   1. In this project, the train dataset consisted of various features a car has and the target variable was “popularity” ranging from 0-4.
   2. Depending upon the features, we had to predict the car popularity on test dataset.
   3. Practiced ways to tune a random forest to get optimum model setup
2. In News Category prediction:
   1. There was just 1 dependent variable “Story”, which consisted the story of the news and the target variable was “Category” which ranged from 0-3, each number denoting a category of news.
   2. Learnt to implement very important preprocessing technique called lemmatization.
   3. Since machine learning only works on numerical values and not on strings, it was quite essential to understand the string conversion methods.

Skills:

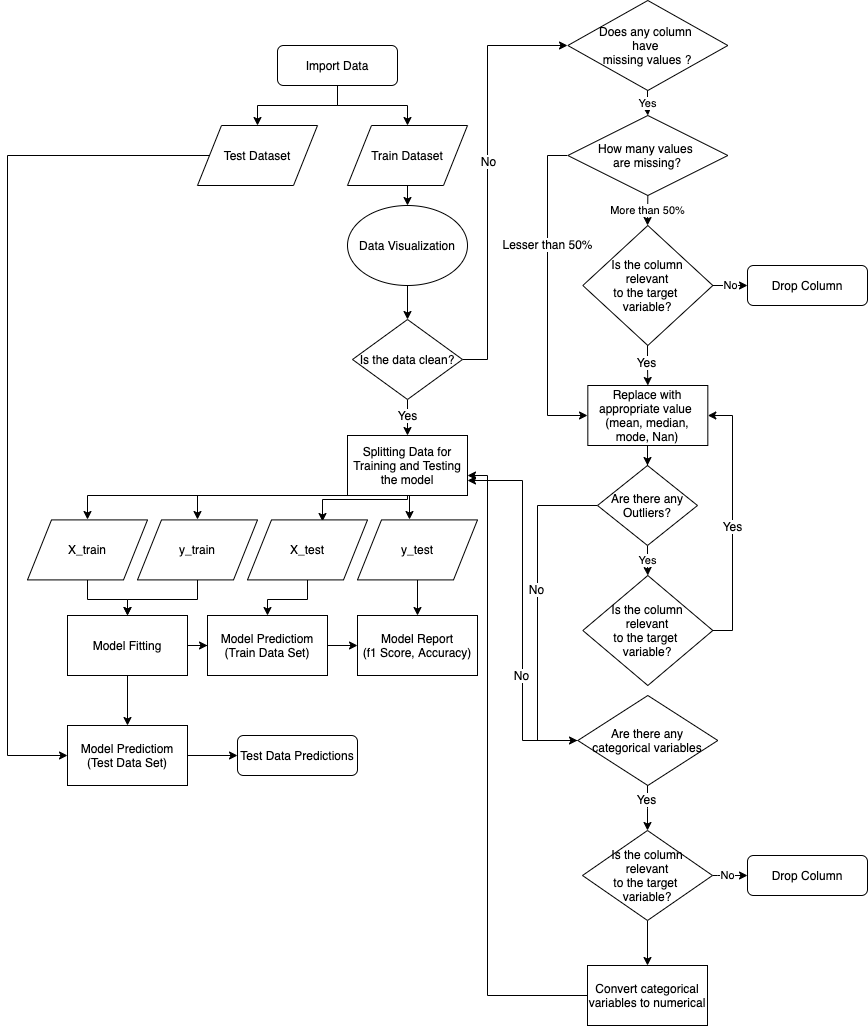
* Data Preprocessing
* Data Visualization
* Data Analysis
* Machine Learning
* Natural Language Processing

Some Basic Visualization:

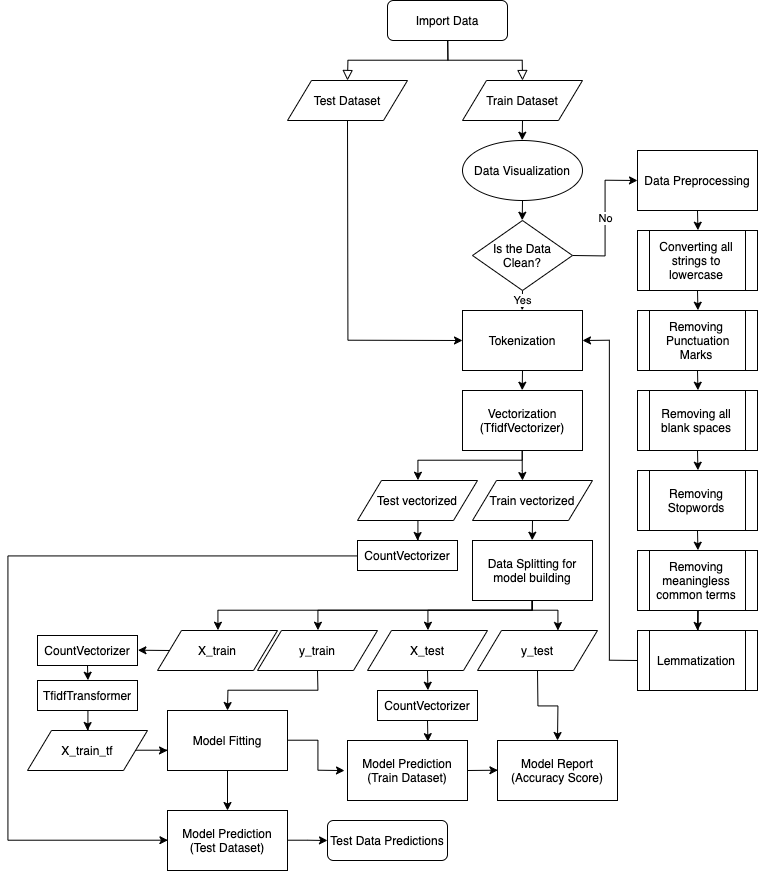
Car Prediction Model Analysis News Category Analysis



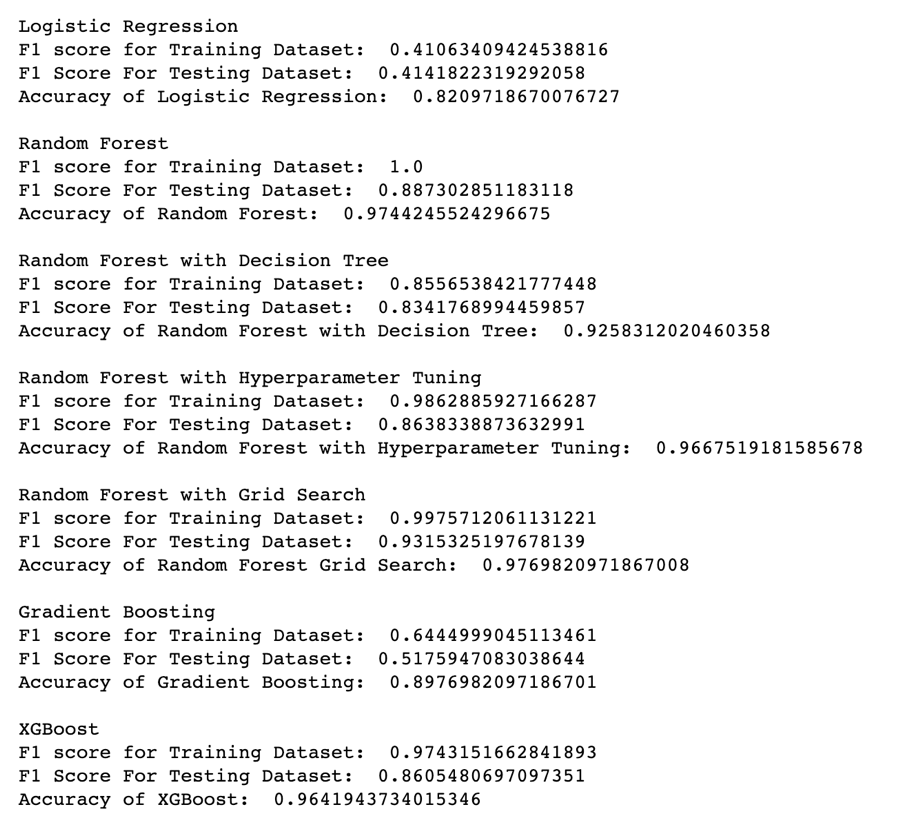
* The box plot shows the relation between the “maintenance cost” and “buying cost” based on popularity. If you observe, the more popular the car is, usually the maintenance reduces.
* In news popularity, the pie chart denotes the average count of words in each story. And as it seems, politics has the highest average word count in its stories.
* Workflow for ML:



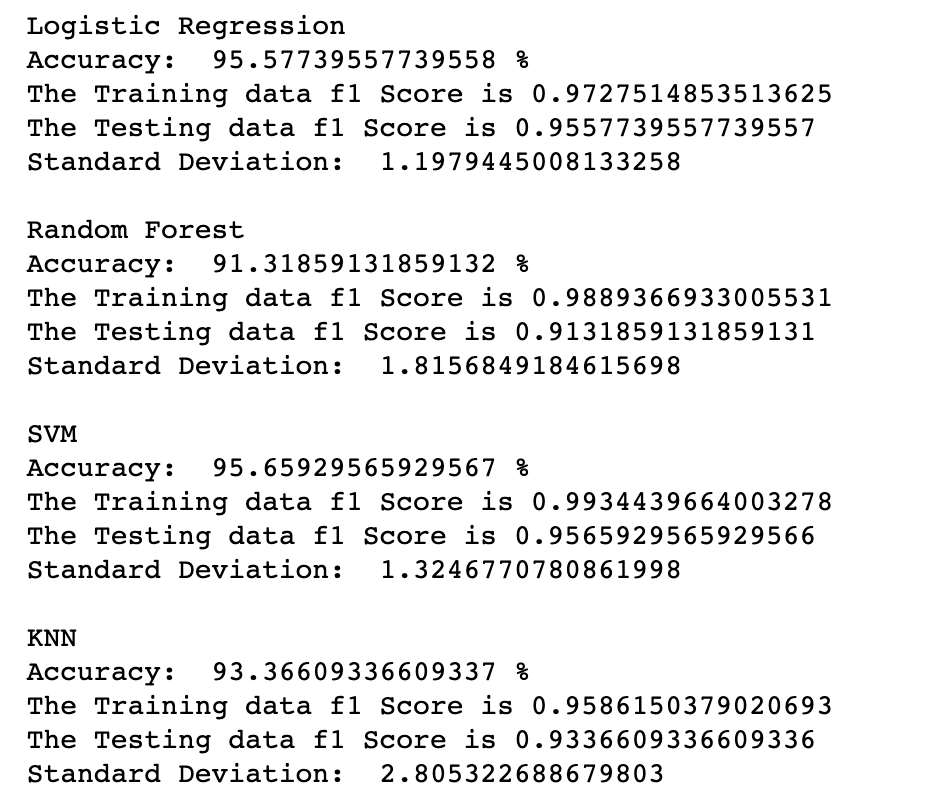
Workflow for NLP:



Models implemented In Car Prediction Model:



Models implemented In News Category Model:



Conclusion:

As the models in car prediction models,

* Logistic regression performed poorly.
* The random forest with its default parameters had decent Accuracy but the model was overfitting.
* When I implemented decision tree, the scores were decent but not the best in the pack
* In Hyperparameter tuning, I did tune the model to get optimum parameters, and it actually it did drop the accuracy by a little bit but my achievement was the Random Forest model wasn’t overfitting anymore.
* When I used Grid Search, the algorithm gave me the best parameters to fit in the model and after passing the parameters as an input to Random Forest classifier, it gave the best output amongst the path.

As the models News Prediction models,

* It is very important the way you clean the data for model performance.
* There are multiple ways of performing data cleaning but choosing the most efficient one is the most important aspect.
* Also make sure you don’t clean the data to an extent that the words would stop making any sense. Be practical. Implementing Lemmatization and Stemming on the same dataset might be one of them.
* The Logistic Regression Algorithm was straight up best performing algorithm amongst the pack.
* Support Vector Machine (SVC) was on the same level of Logistic Regression Model

Future Scope:

Various other techniques are yet to be explored. In future I am planning to implement POS(Parts of Speech) and BOW(Bag of Words) to understand Natural Language Processing in depth.