Capstone Project

Sentiment Analysis

By,
Yamini

Steps followed in Sentiment analysis

1. Problem statement
2. Missing value analysis
3. Data Cleaning and Analysis
4. Data Visualization
5. Feature Engineering and Selection
6. Splitting the data into train and test set
7. Dealing with imbalanced set
8. Model building and hyperparameter tuning
9. Model Validation using evaluation metrics
10. Model Selection

Problem statement

This challenge asks you to build a classification model to predict the sentiment of COVID-19 tweets. The tweets have been pulled from Twitter and manual tagging has been done and the names and usernames have been given codes to avoid any privacy concerns.

The following are the variables for analysis:

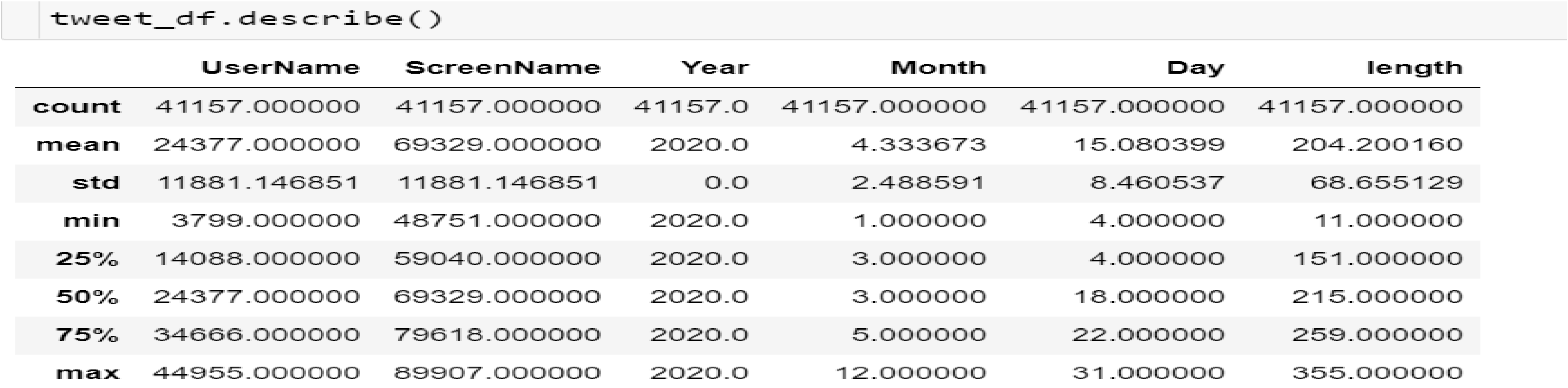
1. User name
2. Screen name
3. Location
4. Tweet At
5. Original Tweet
6. Label

Data pre-processing

- There are a total of 41157 rows and 6 columns of raw data obtained from Twitter for Sentiment analysis.
- There are 8590 missing values in the location variable.
- These are filled with Not disclosed/not provided.
- There are 4 object and 2 integer variables from which date variable is converted into datetime and some other features like month, date, year values are obtained separately.
- The raw tweet variable is given which is important for predicting the sentiment of the tweet. This raw tweets are cleaned by removing punctuations and stop words. One more feature is created from raw tweet named as length which gives the length of the tweet.
- New features are created from the new cleaned tweets using tfidf vectorizer which gives important words that are present in each document and some other parameters specified.

EDA

- After bringing out new data for analysis from the raw data, the statistics are found from the describe method and pandas profiling.



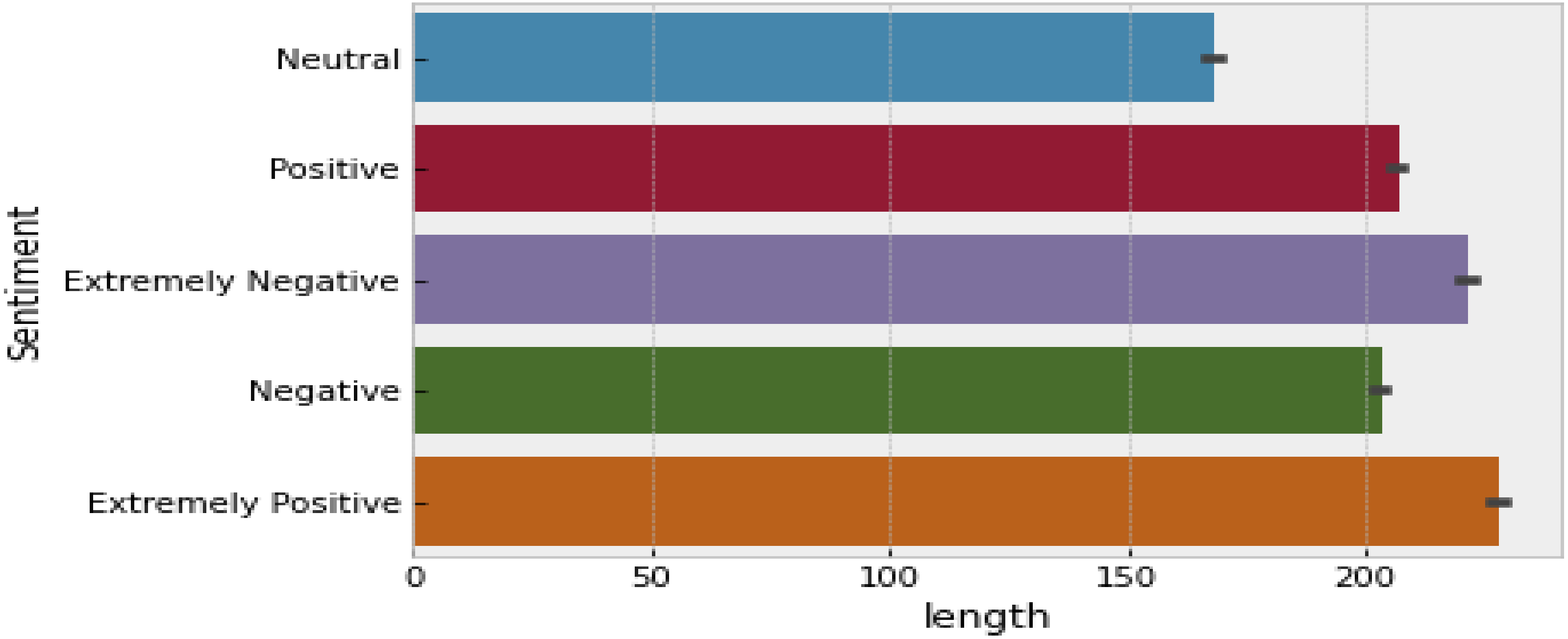
```
tweet_df.describe()
```

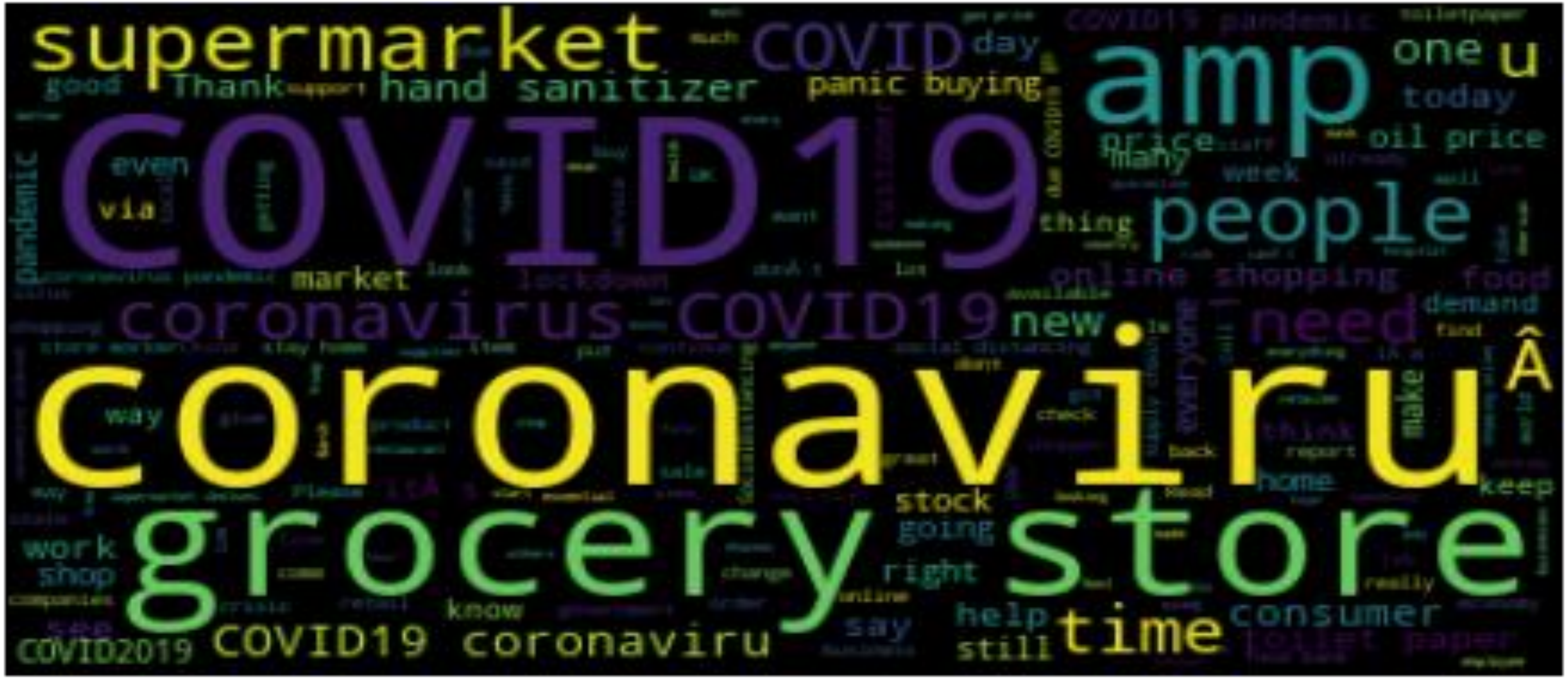
	UserName	ScreenName	Year	Month	Day	length
count	41157.000000	41157.000000	41157.0	41157.000000	41157.000000	41157.000000
mean	24377.000000	69329.000000	2020.0	4.333673	15.080399	204.200160
std	11881.146851	11881.146851	0.0	2.488591	8.460537	68.655129
min	3799.000000	48751.000000	2020.0	1.000000	4.000000	11.000000
25%	14088.000000	59040.000000	2020.0	3.000000	4.000000	151.000000
50%	24377.000000	69329.000000	2020.0	3.000000	18.000000	215.000000
75%	34666.000000	79618.000000	2020.0	5.000000	22.000000	259.000000
max	44955.000000	89907.000000	2020.0	12.000000	31.000000	355.000000

- Note: describe() gives only stats on numeric data and there is much more analysis remaining.

Next…->

Data Visualization – Title vs length of the tweet

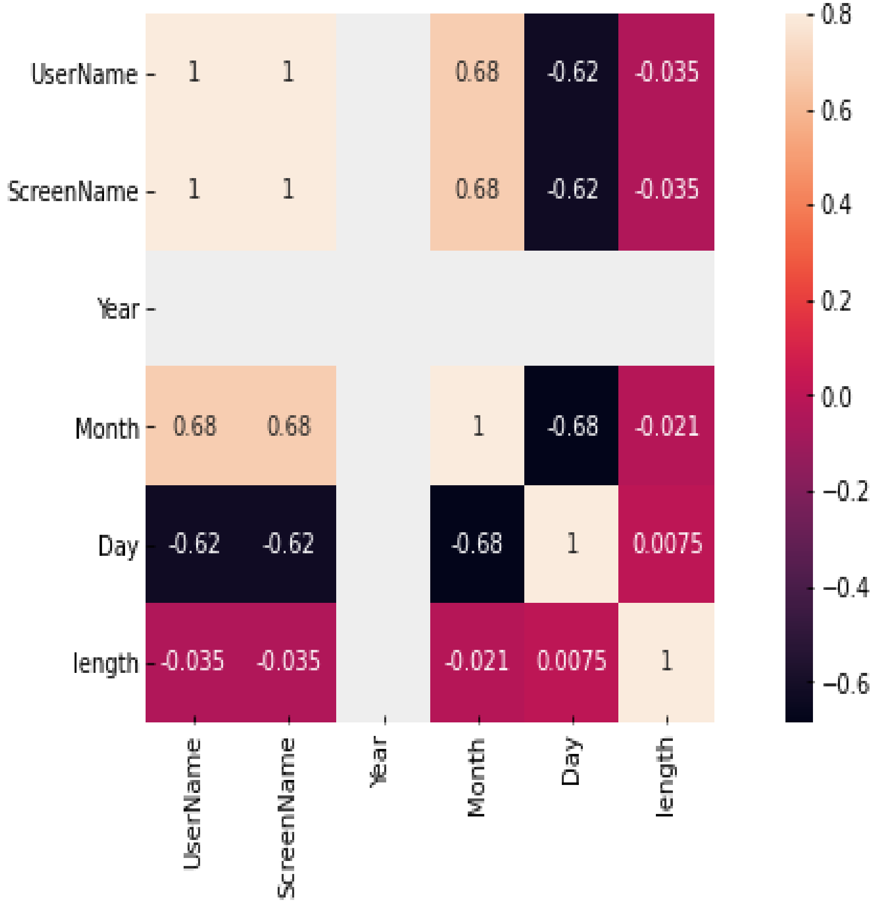




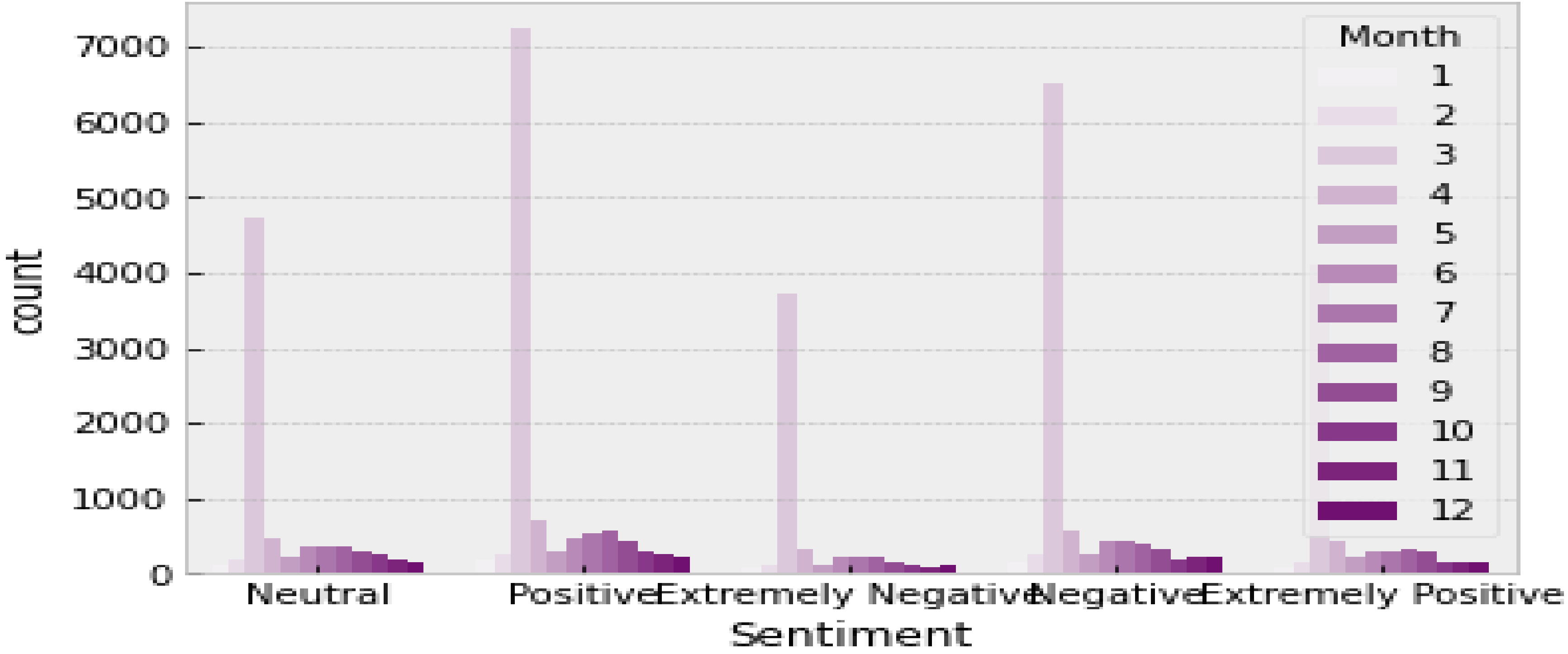
Twitters users terms/words used in their tweets

Correlation matrix

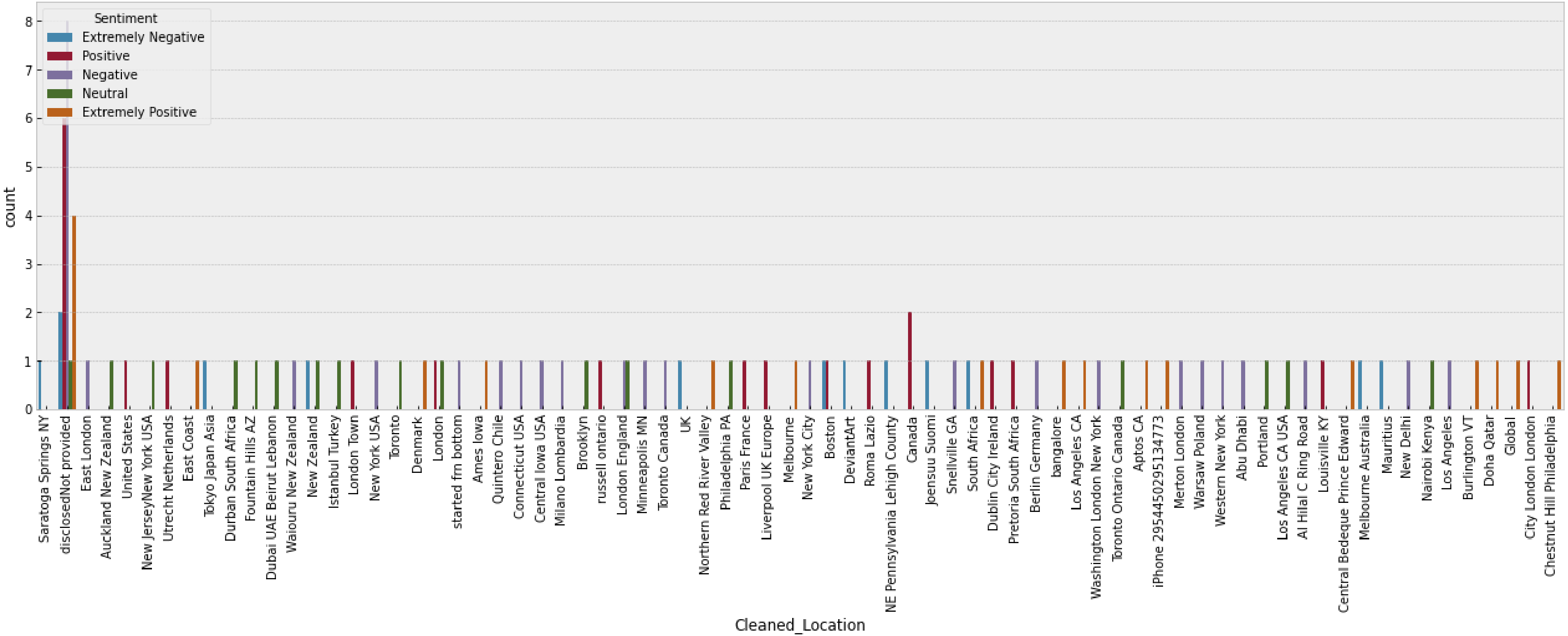
- Correlation matrix only has integer variables in it and can say the correlation that exists between the numeric variables with percentage.
- User name, Screen name are getting correlated with month and they are not used for analysis and are just unique values. So can disregard them.
- Length is showing a correlation of 0.0075,-0.021 with Day and Month then they are not highly correlated variables. So all these variables can be considered for analysis.



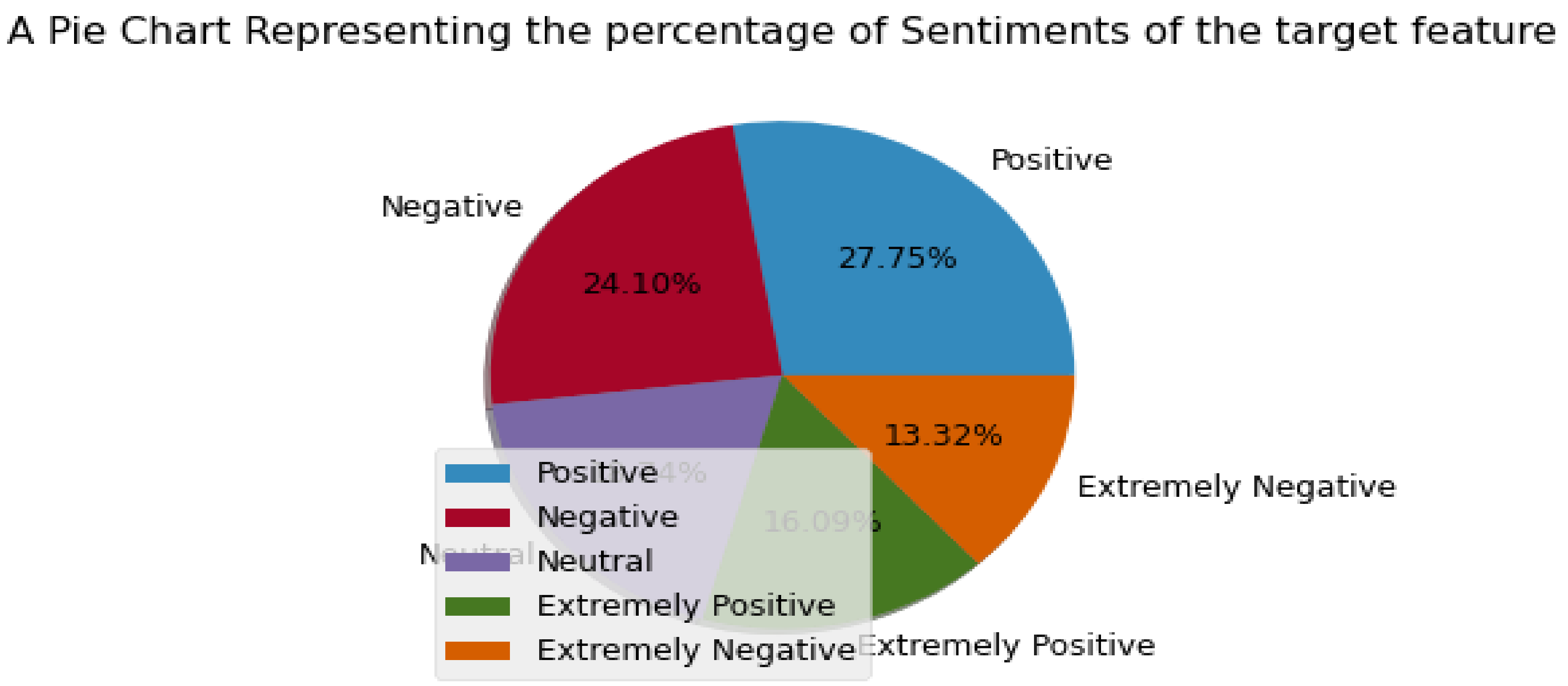
	UserName	ScreenName	Year	Month	Day	length
UserName	1	1		0.68	-0.62	-0.035
ScreenName	1	1		0.68	-0.62	-0.035
Year						
Month	0.68	0.68		1	-0.68	-0.021
Day	-0.62	-0.62		-0.68	1	0.0075
length	-0.035	-0.035		-0.021	0.0075	1



The different classes in Sentiment(target) vs the length of the tweet in every month



This is the sample data taken from population to find some location tweets☝



A pie chart representing the imbalance in the target classes

Feature engineering and selection

- Dropped the redundant variables which cannot be used for classification.
- Creating new features using Hash encoding for Location variable so that there no data loss and also helps in getting required dimensions for analysis.
- Combining all the necessary features and splitting the dataset into train and test sets.
- After splitting the dataset, apply SMOTE(Synthetic minority over-sampling technique) used for balancing the imbalance present in the target label by generating synthetic data to it.

Models used for the multi-class classification

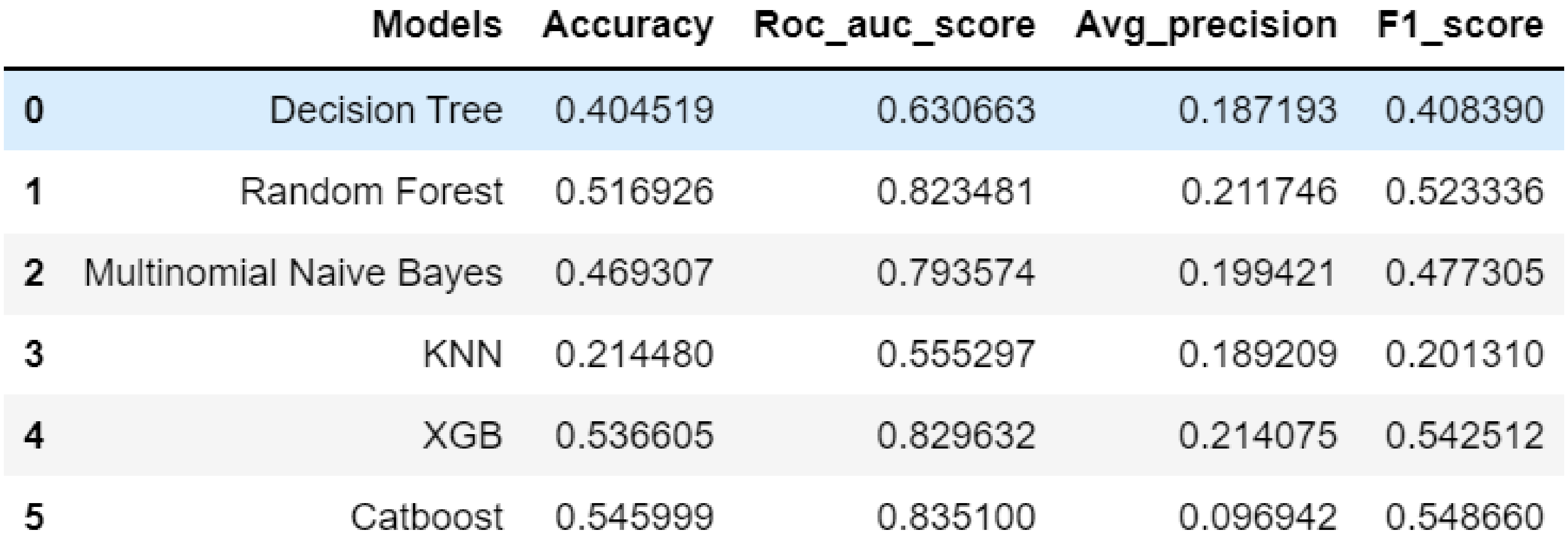
1) Decision tree
2) Random Forest
3) Multinomial Naïve Bayes
4) KNN
5) XGB Classifier
6) CatBoost

Sentiment…..?

Evaluation metrics

- There are many evaluation metrics for Classification. They are Accuracy, Precision, Recall, Roc-auc score, f1-score, kappa score, classification report and confusion matrix..etc.
- The metrics I used for evaluating the models for this multi-class classification is Roc-auc score, F1-score(Geometric mean of precision and recall).
- I have chosen these metrics because accuracy may not be good to use for evaluation for imbalanced set. Hence using the metric which can be used in differentiating and working as a evaluation metric for imbalanced classification.

Results on the test set

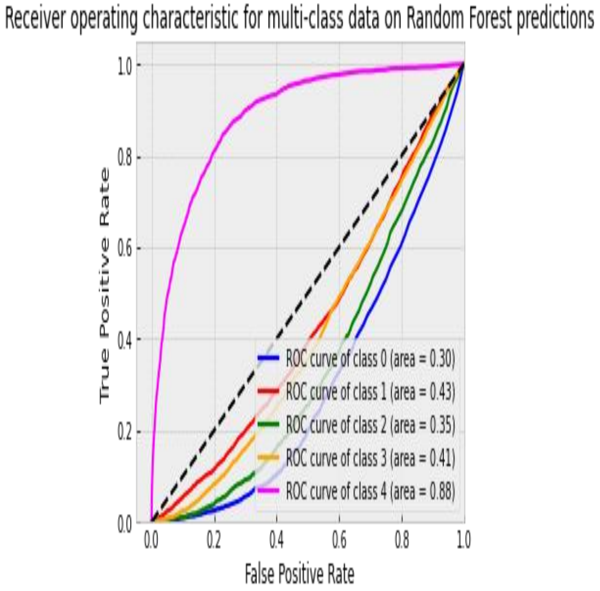


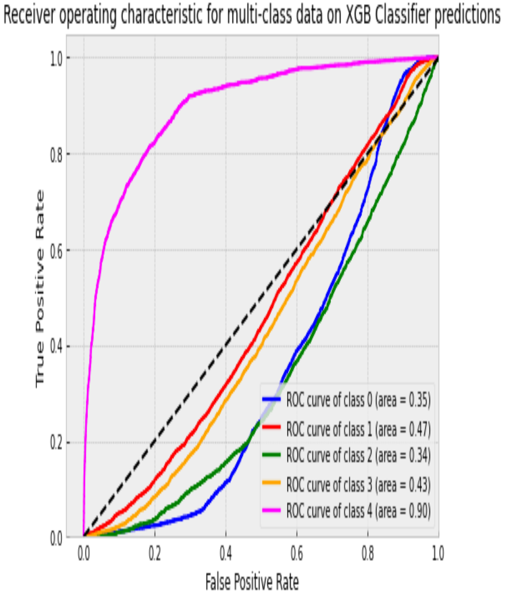
	Models	Accuracy	Roc_auc_score	Avg_precision	F1_score
0	Decision Tree	0.404519	0.630663	0.187193	0.408390
1	Random Forest	0.516926	0.823481	0.211746	0.523336
2	Multinomial Naive Bayes	0.469307	0.793574	0.199421	0.477305
3	KNN	0.214480	0.555297	0.189209	0.201310
4	XGB	0.536605	0.829632	0.214075	0.542512
5	Catboost	0.545999	0.835100	0.096942	0.548660

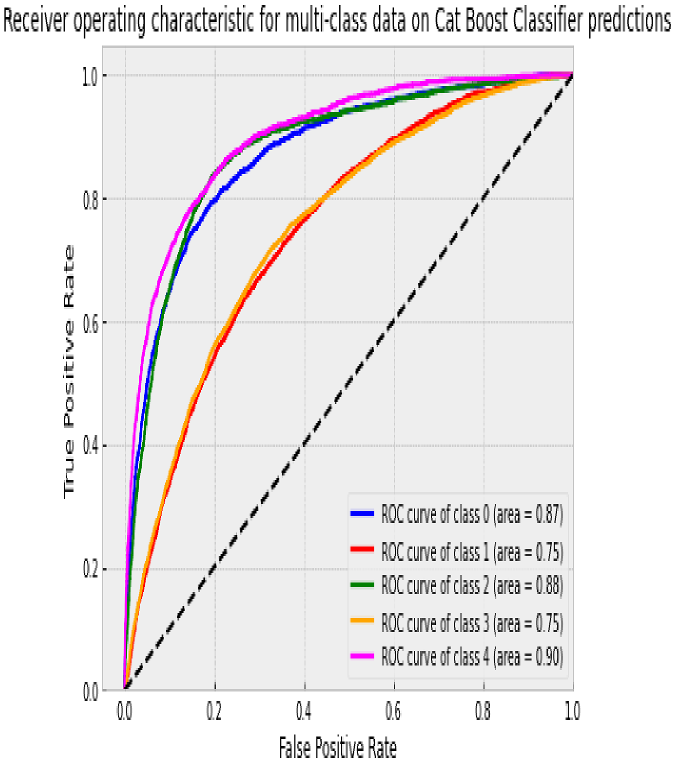
Model Selection

CatBoost

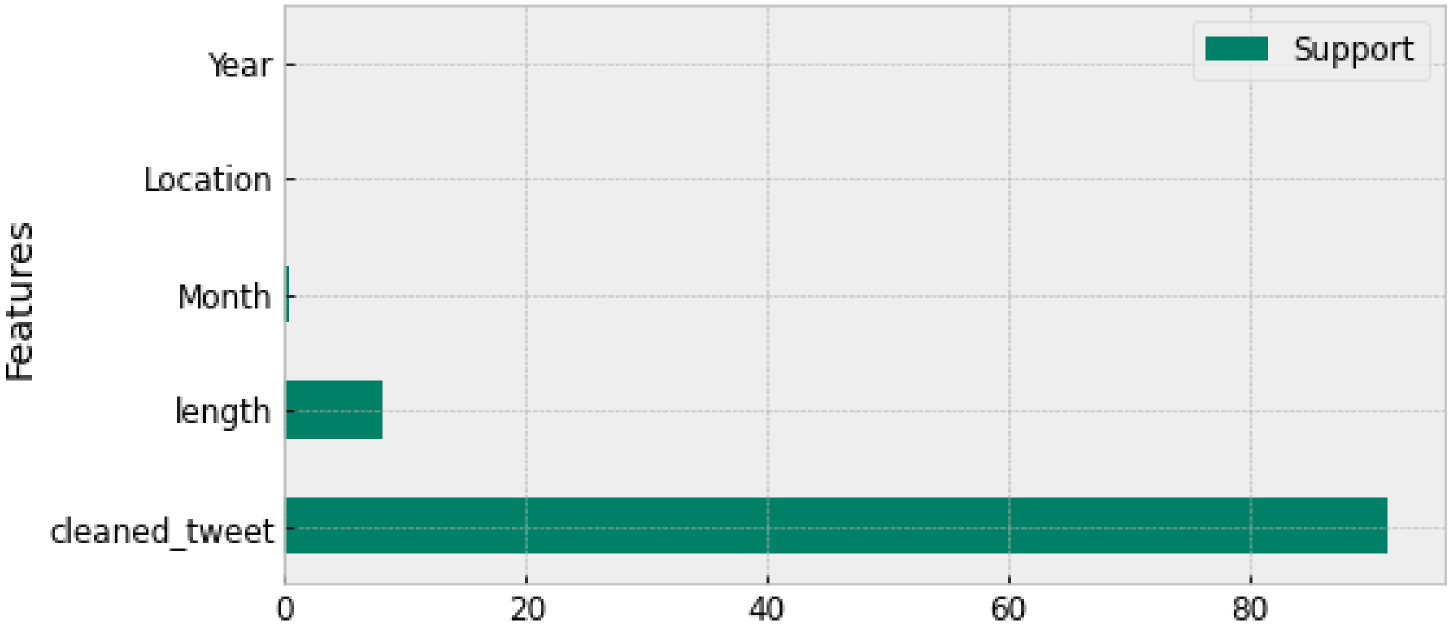
Model Evaluation







Picture says how CatBoost is the best model for predicting the Sentiment(class).



Feature importance plot from CatBoost model

Conclusion

- Here the most important feature that gives most of the correct predictions is due to the cleaned_tweet feature which contains the cleaned tweet of the tweets given in the data.
- The model which understands the language and terms used in it has given the correct predictions compared to the model which couldn’t capture or understand the data and has underfit to the data.
- From the models used for prediction the models which are underfit the data are KNN and decision tree. As KNN is a lazy learner it couldn’t properly learn the data and decision tree is one more underfit model as one tree couldn’t capture the population tweets.
- When Random forest is used it has shown the improvement and still random forest can be considered as overfit model where it has learned the training data and also gave good score on test set then it couldn’t differentiate all the classes properly. It didn’t fetch the results on the minority class.
- XGB Classifier has given good score and also precision is high compared to other models then didn’t show good results on minority class.
- CatBoost was performed very well capturing all the classes and improved performance on all the levels of Sentiment present in the data.

Challenges

- Huge dataset
- Memory crash problem
- Missing values
- Feature selection and the good number of features unknown as there are many tweets with many words in it.
- Loss of information and data security concerns
- Computation time
- Hyperparameter tuning takes long time and sometimes kernel dies.

Thank you😊