Capstone Project

Coronavirus Tweet Sentiment Analysis

Sandesh Gaikwad

Content

- Problem Statement
- Exploratory Data Analysis
- Preprocessing of Tweets
- Modeling and comparing results
- Conclusions

Problem Statement

- Use Different machine learning models to classify the tweets by sentiments

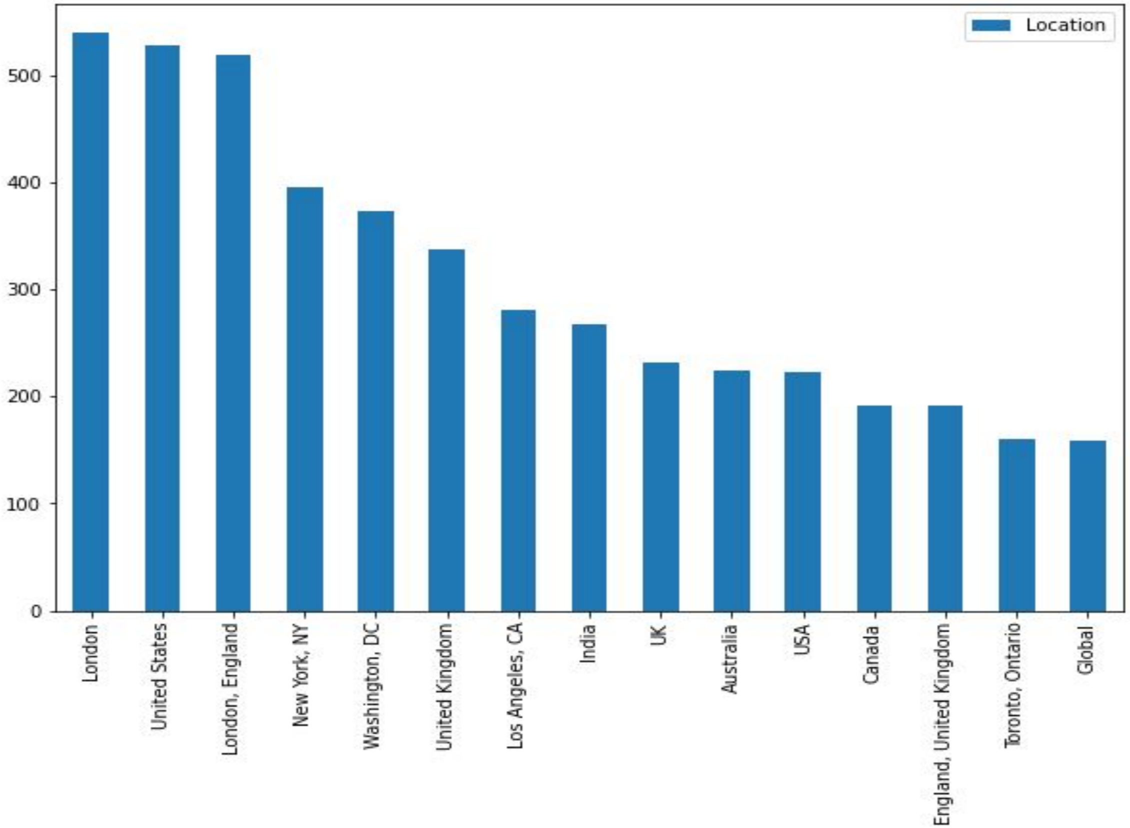
- Find which method gives best results and do evaluations on test data

Data Summary

- UserId and ScreenId - These two columns are identity columns for people who are using twitter and are irrelevant for our classification
- Location - At which location tweeted
- Tweet At - date of tweet, data is available for one month from 16/03/2020 to 14/04/2020
- Original Text - Actual tweet text
- Label - there are five labels for tweet - Extremely Negative , Negative , Neutral , Extremely Positive , Positive

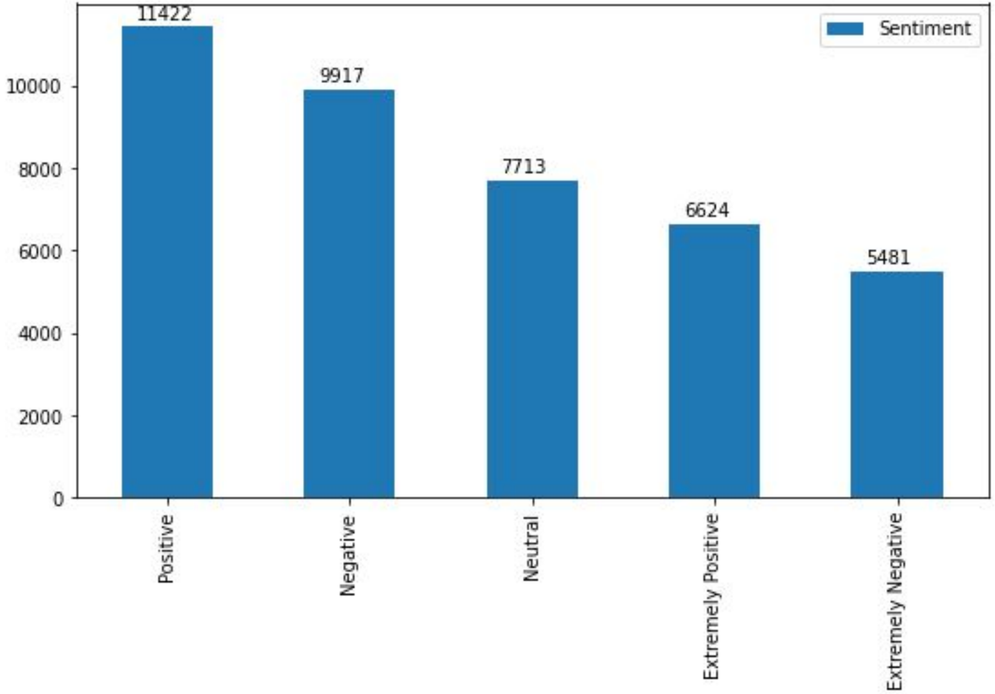
Top Tweet Locations

- Most tweets are from london and other part of Uk
- Other are from eastern CIties of USA



Sentiments

- We can use try most of ml algorithms as the data is not unbalanced
- Extreme tweets are smaller in number compared to the positive negative and neutral



Tweet Preprocessing

- Remove user tages:
  - User tags don't give any information about the content of the tweet these are just usernames
- Remove links
  - links of the websites do not give any specific information about the content of the tweet
- Remove stop words and punctuation
  - In these classification we are using tokenization based machine learning approach, so stop words just don’t and any value in predicting anything as they are present everywhere

Tweet Preprocessing

- Use stemming :
  - Stemming is the process of producing morphological variants of a root/base word.
  - A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”.

Vectorization

- Countvectorizer - Text into a vector on the basis of the frequency (count) of each word
- TFidfvectorizer - Term frequency-inverse document frequency
- Hyperparameters
  - max_df - includes words that occur less than 80 % of documents
  - min_df - includes words that occur more than 20 times

Understating Misclassification

- Five Classes : Extremely Negative , Negative , Neutral , Positive , Extremely Positive

- Three Classes : We will merge
    - Extremely Negative and Negative to one class negative
    - Extremely Positive and Positive into one class positive
    - We will keep Neutral as Neutral

Ordinal Error

- Some misclassification are okay but some aren't - ‘Extremely Negative’ to ‘Negative’ is okay but ‘Extremely Positive’ is not okay
- We will encode these classes and get RMSE error to check ordinal error from our test data
- Encoding
    - ‘Extremely Negative’ : -3
    - ‘Negative’ : -2
    - ‘Neutral : 0
    - ’ ‘Positive’ : 2
    - ‘Extremely Positive’ : 3

Naive Bayes (Baseline Model)

- The Multinomial Naive Bayes algorithm is a Bayesian learning approach
- It calculates each sentiment's likelihood for a word and outputs the sentiment with the greatest chance
- We will use this model as a baseline model

Multiclass Logistic Regression

- there are 5 classes. Hence, we need to train 5 different logistic regression classifiers.
- When training the classifier for label 1, we will treat input data with class 1 labels as true samples and all other classes as false sample.
- This will continue for all the classes.

Ensemble of tree

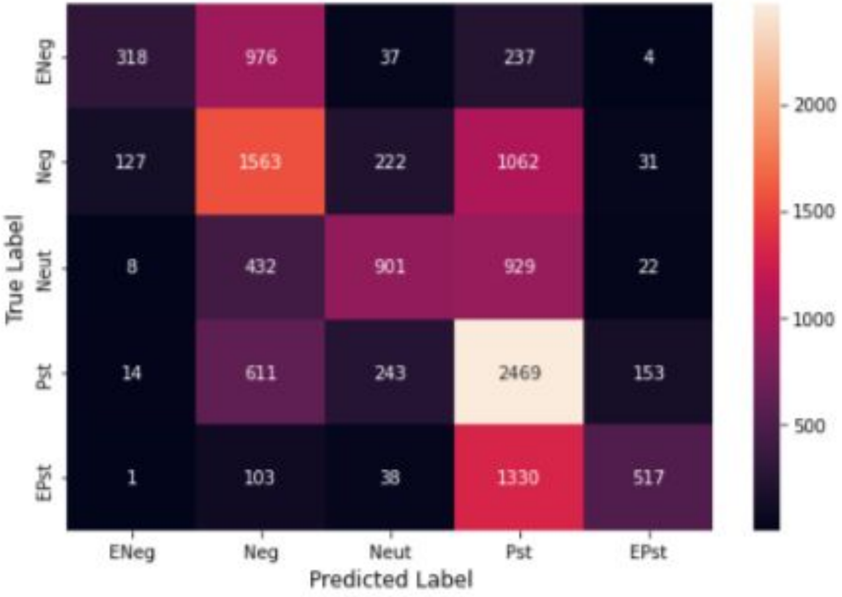
- Bagging: -
    - Random Forest :

- Boosting :
    - Xgboost :
    - Catboost :

Model Comparison

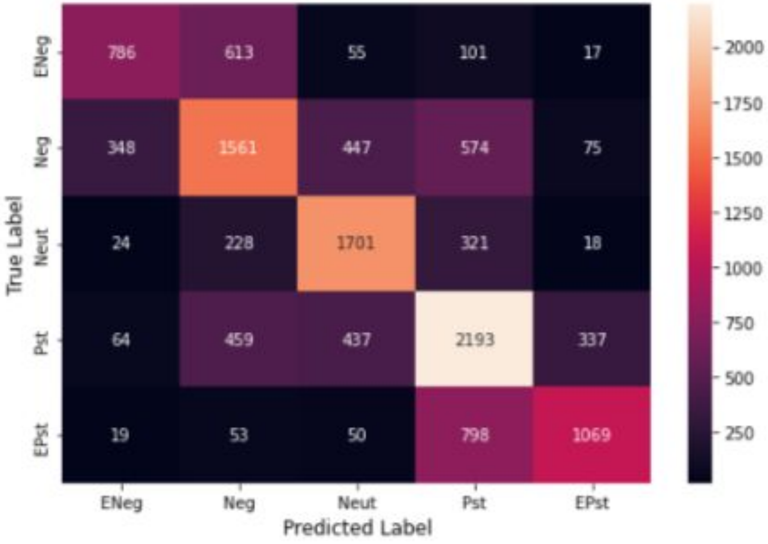
Model	RMSE	Accuracy (5 Labels)	Accuracy (3 Labels)
Naive Bayes (baseline model)	1.83	0.49	0.68
Xgboost	1.74	0.51	0.74
Random Forest	1.57	0.51	0.69
Logistic Regression	1.74	0.52	0.74
Catboost (best performing model)	1.5	0.59	0.76

Naive Bayes vs Catboost



True Label \ Predicted Label	ENeg	Neg	Neut	Pst	EPst
ENeg	318	976	37	237	4
Neg	127	1563	222	1062	31
Neut	8	432	901	929	22
Pst	14	611	243	2469	153
EPst	1	103	38	1330	517

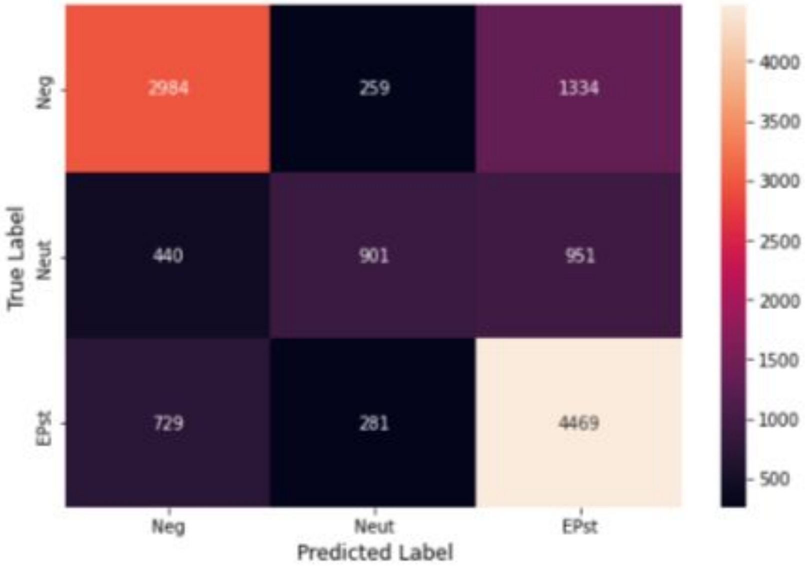
Naive bayes model



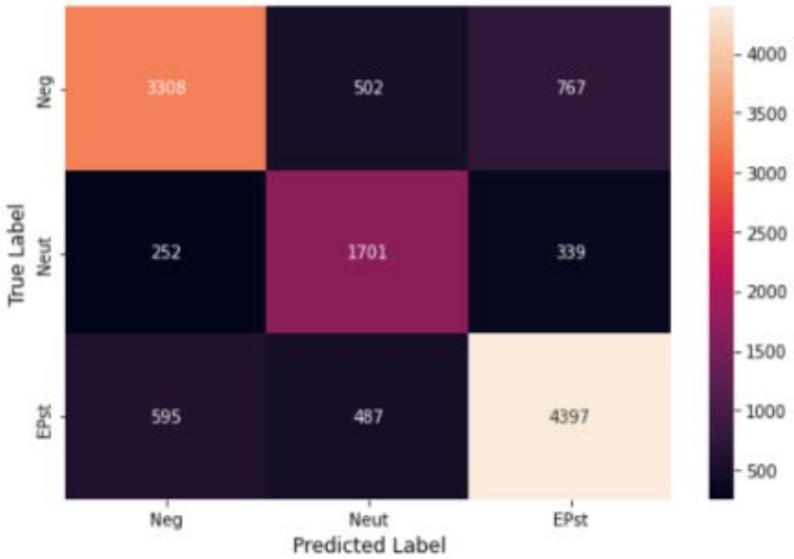
True Label \ Predicted Label	ENeg	Neg	Neut	Pst	EPst
ENeg	786	613	55	101	17
Neg	348	1561	447	574	75
Neut	24	228	1701	321	18
Pst	64	459	437	2193	337
EPst	19	53	50	798	1069

Catboost Model

Naive Bayes vs Catboost



Naive Bayes model



Catboost model

Conclusion

- Catboost gives the best performance among the all models with 59% accuracy

- Catboost also gives smallest Rmse and 76% accuracy if we consider only 3 labels

- The models from scikit learn does not support ordinal errors. The current classification weights every error in the same way but some sentiments are closer to other that needs to be considered. The use of classification models with ordinal errors will improve performance.