

CS 636 1J2

Project 2

Natural Language Processing with Disaster Tweets

MAY 2021

# 

# 

# Team members and each one’s contribution

As mentioned above, our team members include Parvathy Neelakanta Sarma, Roma Dungarwal, and Srishti Karakoti.

We decided as a group to work on the project together over google hangout. We scheduled regular meetings to work on the project. Every member contributed equally to develop and test the project. Every member worked on different tasks such as researching public kaggle notebooks to find the most appropriate packages to use, writing functions, and writing regular expressions.

# 

# Pipeline and Methods

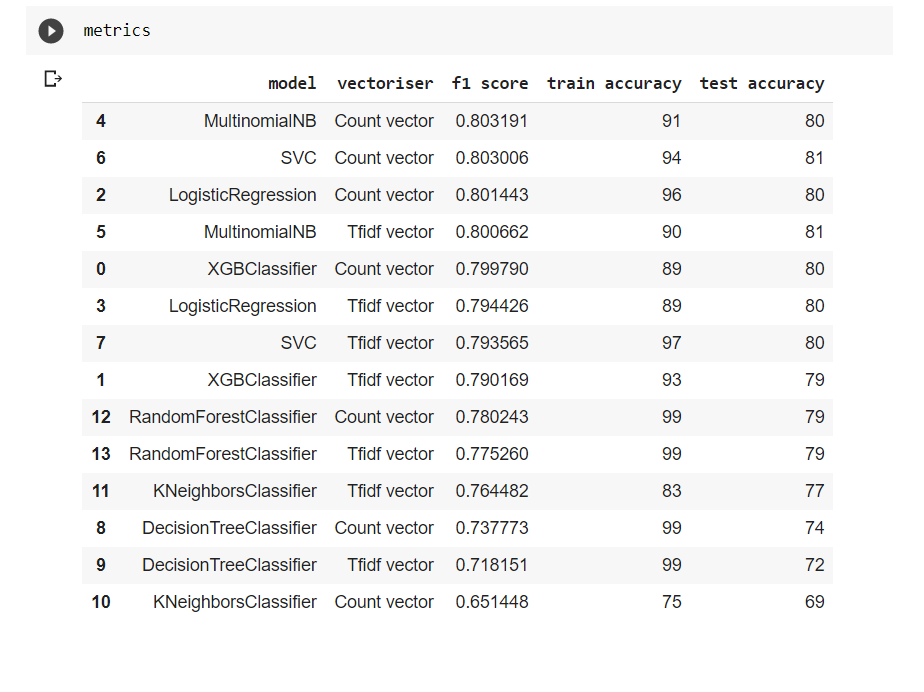
We imported required libraries and set random seed. We used the seed() method to customize the start number of the random number generator. For Exploratory Data Analysis, we loaded the data and checked for null values. Then, we investigated features such as "keywords", "location", and "tweet". Next, we created visualizations such as bar plots and histograms to understand the data. For Data Preprocessing, we wrote several functions for different tasks such as for removing URLS, removing non printable characters, removing all english stopwords, etc.

For Vectorizing the train and test data we used count vectorizer to convert a collection of text documents to a matrix of token counts. For vectorizing the train and test data, we also used Tfit vectorizer to convert a collection of raw documents to a matrix of TF-IDF features.

For creating different models [LogisticRegression, SVC, MultinomialNB, DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier] to compare results, we assigned the random state. A random\_state parameter may be provided to control the random number generator used. For each model, we used both vectorizers (count and tfidf) to fit and predict. Then we printed the model, vectorizer, F1 score, and accuracy (training and testing) in descending order. Multinomial Naive Bayes with count vectorizer resulted in the best F1 score. We used Multinomial Naive Bayes with count vectorizer to create the submission file.

# 

# Training Performance



# 

# Final testing performance and ranking in Public Leaderboard

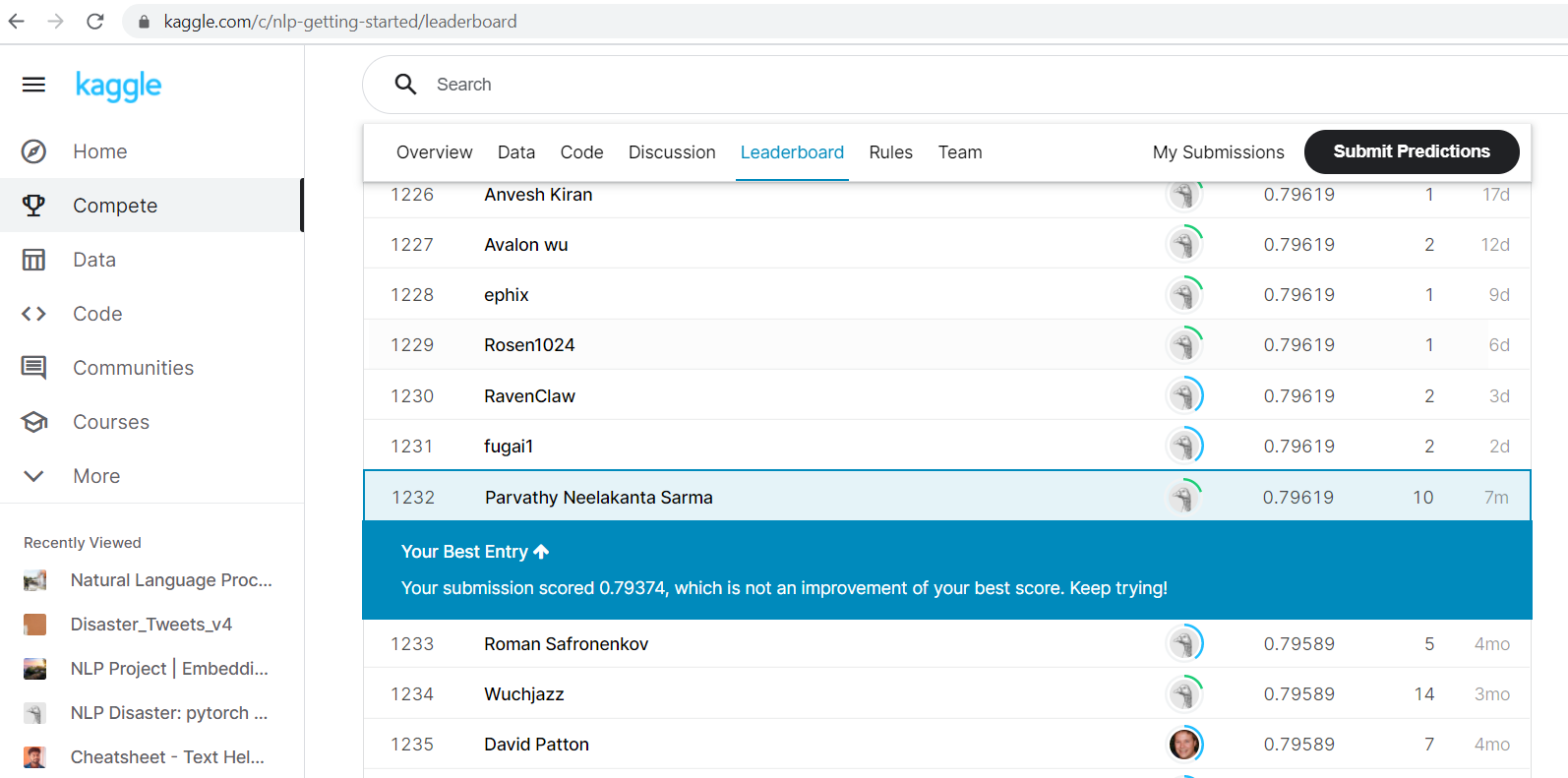
# 

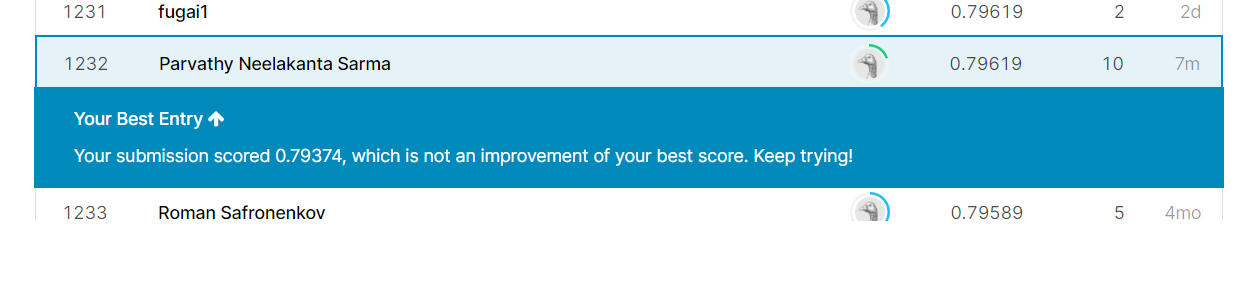
Final testing performance - 0.79619

Ranking in Public Leaderboard - 1232

# 

# Kaggle Public Leaderboard Board Ranking

****

****