Project 3: Reddit Data Scraping and Building Classification Model

Saba Suhail

Overview

Nate Silver and co. at FiveThirtyEight have agreed to hear my pitch for a story in two weeks. I need to make a narrative on how to create a Reddit post that will get the most engagement from Reddit users.

This project will involve web-scraping, NLP and classification models.

Use of PRAW:

- 11353 data-points which after dropping duplicates become 10958 entries

-Post_id, post_title, authors_link karma, authors_comment_karma etc adding to 19 columns per entry

Understanding the project

Step 1

Scraped data from Reddit

Cleaning

EDA

Step 2

Dummies for subreddits

NLP for post_title:

1. Countvectorization
2. Tfidfvectorization

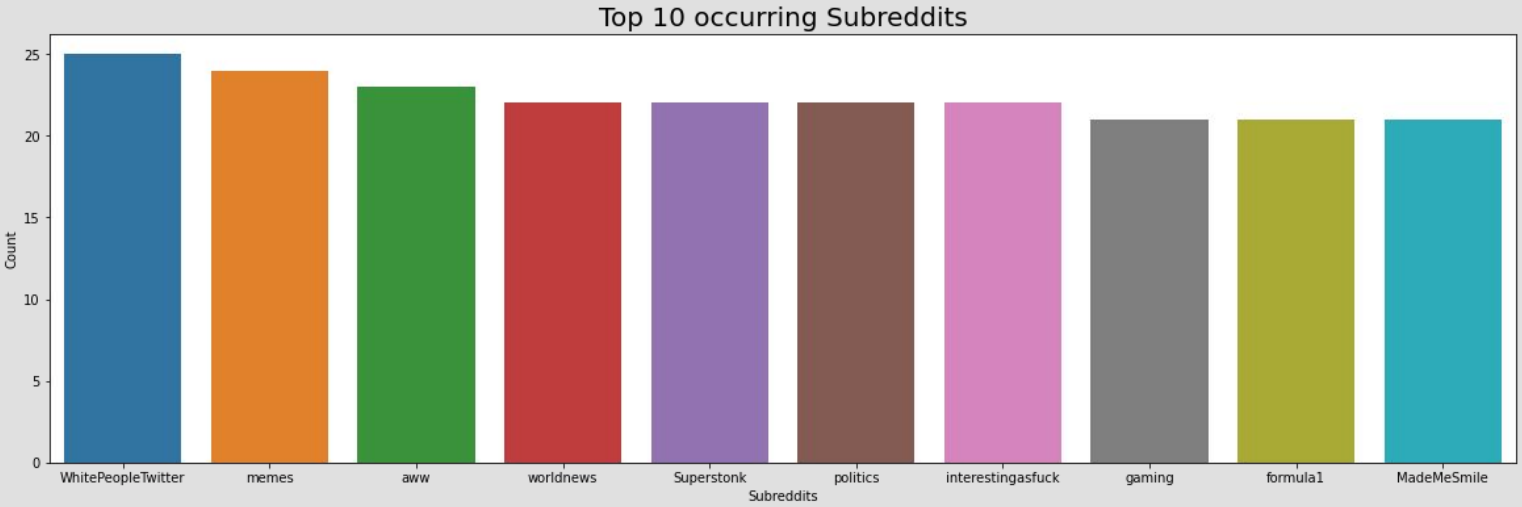
Make new dataframes

Step 3

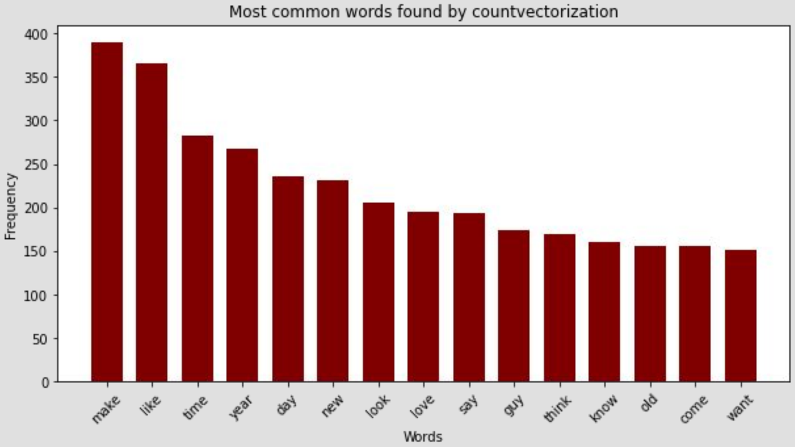
Classification model evaluation:

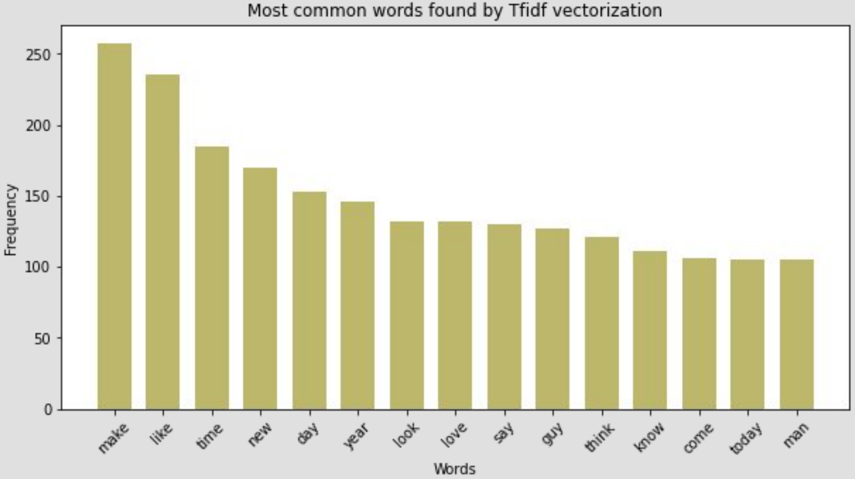
1. RandomForestClassifier
2. RandomForestClassifier with Gridsearch
3. KNN
4. LogisticRegression
5. Ensemble Methods: Decision tree, Bagging, Decision Stump

Understanding the data



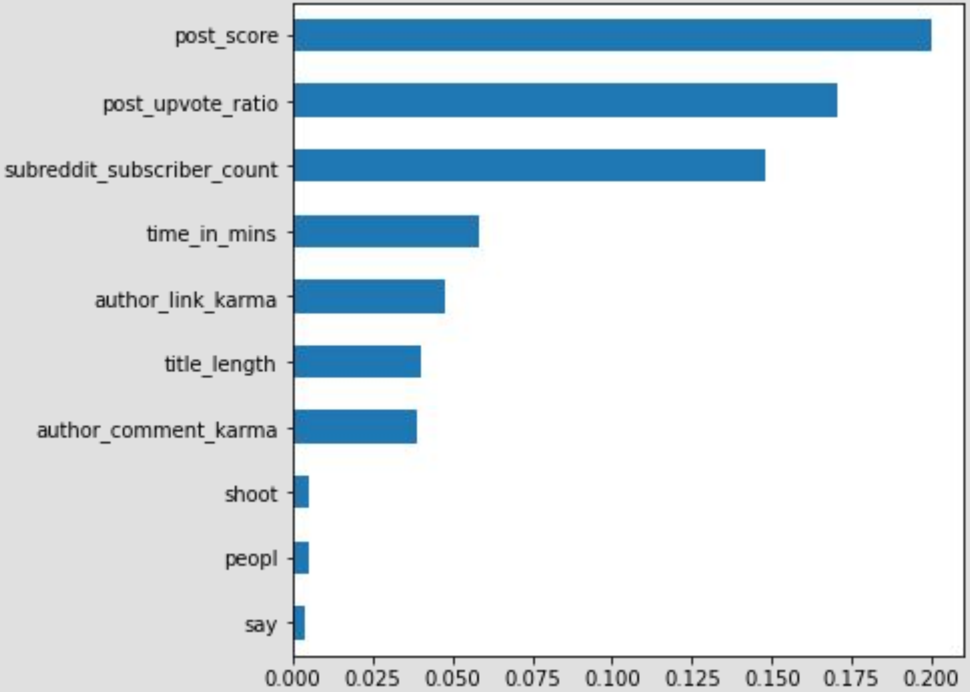
Popular subreddits





Not same but quite similar

RandomForestClassifier for Countvectorized Data



The best parameters on the training data are:

```
 max_depth: 23
 n_estimators: 170
best max_depth: 23
best n_estimators: 170
Random Forest Score: 0.78 +- 0.035
```

Confusion Matrix:

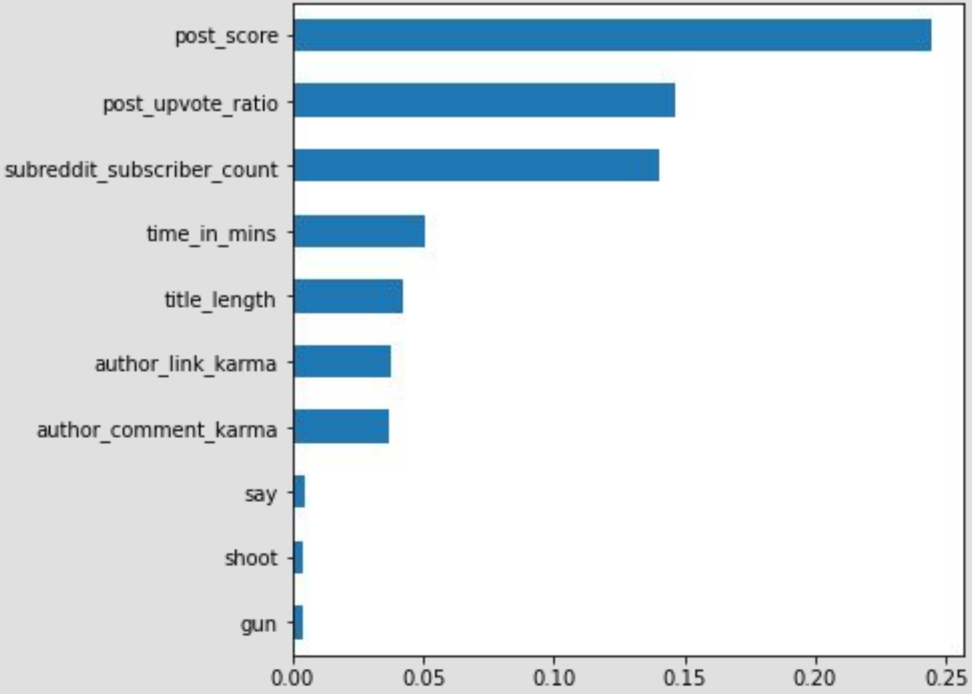
```
[[1085  283]
 [ 328 1044]]

  precision    recall  f1-score   support

           0       0.77      0.79      0.78      1368
           1       0.79      0.76      0.77      1372

    accuracy                           0.78
```

RandomForestClassifier for Tfidfvectorized Data



The best parameters on the training data:

```
 max_depth: 23
 n_estimators: 170
best max_depth: 23
best n_estimators: 170
Random Forest Score: 0.78 +- 0.030
```

Confusion Matrix:

```
[[1101  267]
 [ 339 1033]]
              precision    recall  f1-score   support

           0       0.76      0.80      0.78      1368
           1       0.79      0.75      0.77      1372

    accuracy                           0.78
```

Conclusion

1. Bagging performed best followed closely by RandomForestClassifier, Logistic regression and Decision tree
2. The complexity of algorithms should be considered into account while dealing with huge datasets-use better GPUs for complex algorithms
3. Respective models' accuracy scores on countvectorised and tfidfvectorised data are comparable
4. Best results for k=5 for KNN

Recommendations

1. Post in a subreddit with high subscriber count, tag as many as possible
2. The company should employ an author with high karma(if possible)
3. Have a decent title length
4. Politics, funny, antiwork etc popular topics. Use them
5. Increase upvotes. Ask whoever you know, bots also if possible

Points to ponder

1. Removed post_num_comments as it would be leaking data
2. Time_in_min can be removed
3. Varying max_features in NLP algo will give different results
4. Sample selection problem
5. Don't use things not known apriori-post upvote ratio, post_score, time_in_min for posts yet to be posted (depends on the approach to problem and stage when I need to make predictions)