Machine Learning:

Reddit Classifications

By: Shirley Lin

Dsir-111

Problem Statement:

Problem Statement:

We aim to correctly sort Reddit posts into their related subreddits for this project using the classification model with the highest test score and the most balanced bias/variance tradeoff.

Recipes

1) Data Collection
2) Preprocessing the data
3) Modeling and Evaluation
4) Model selection
5) Prediction

Recipe 1: Data Collection

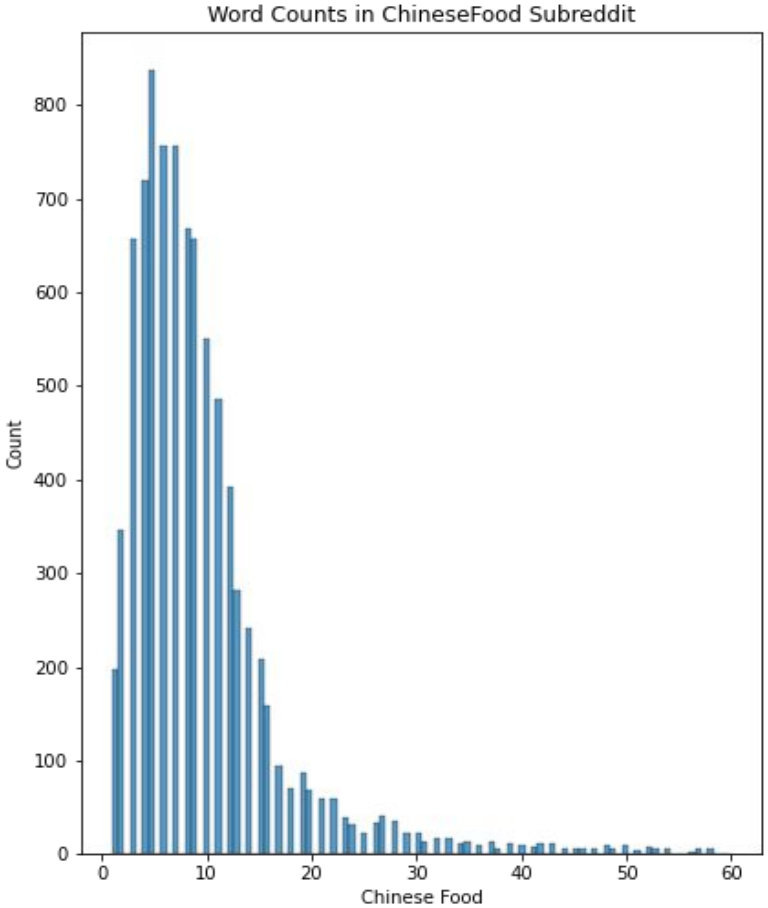
Ingredients

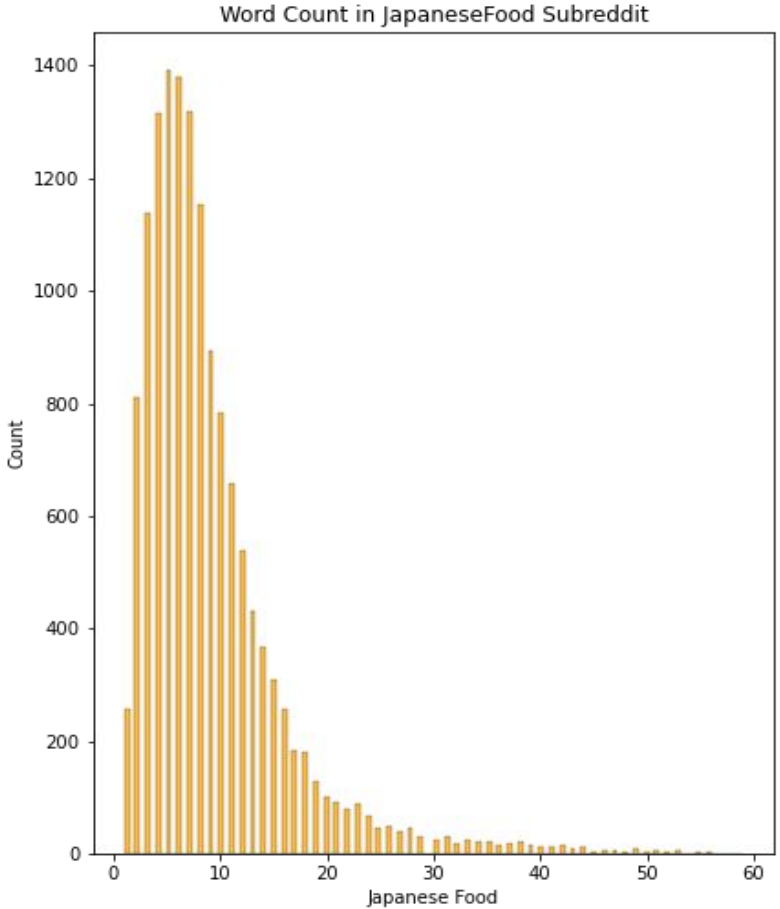
- Pushshift's API
- Subreddits:
  - Japanese Food (15,000 posts)
  - Chinese Food (9,000 posts)

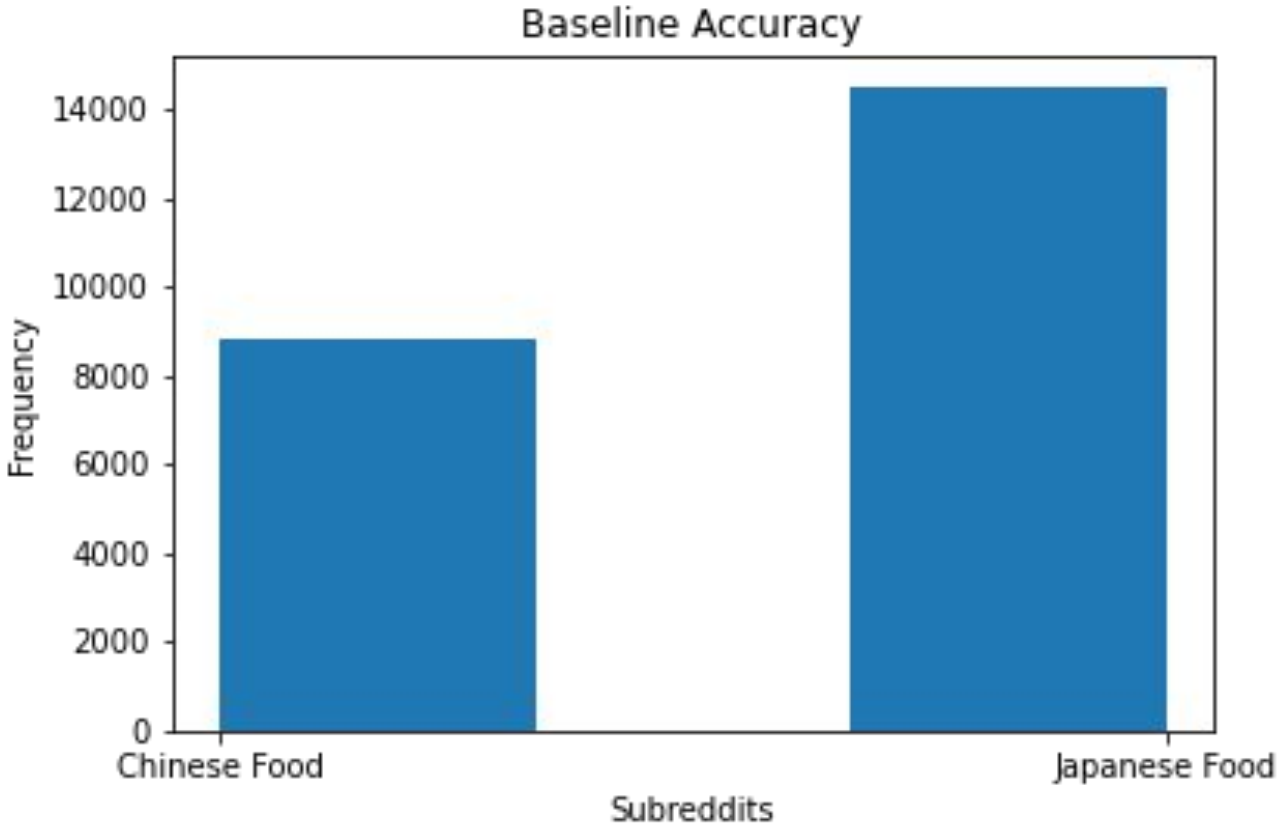
Preparation

1. Create a function to automatically request and pull intended amounts of posts under different subreddits.
2. Set sleep time parameter in accounts for number of requests per second by Reddit.

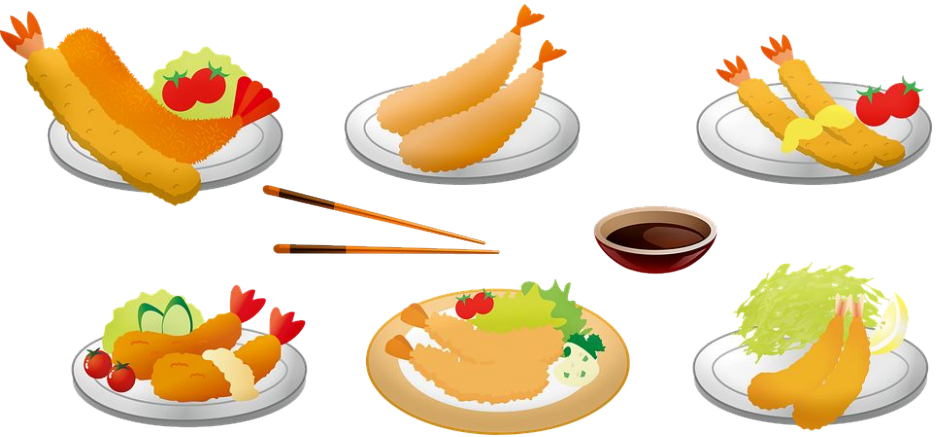
Word Count Distributions






Baseline Accuracy
14000
12000
10000
8000
6000
4000
2000
0
Frequency
Chinese Food
Japanese Food
Subreddits

Recipe 2: Preprocessing

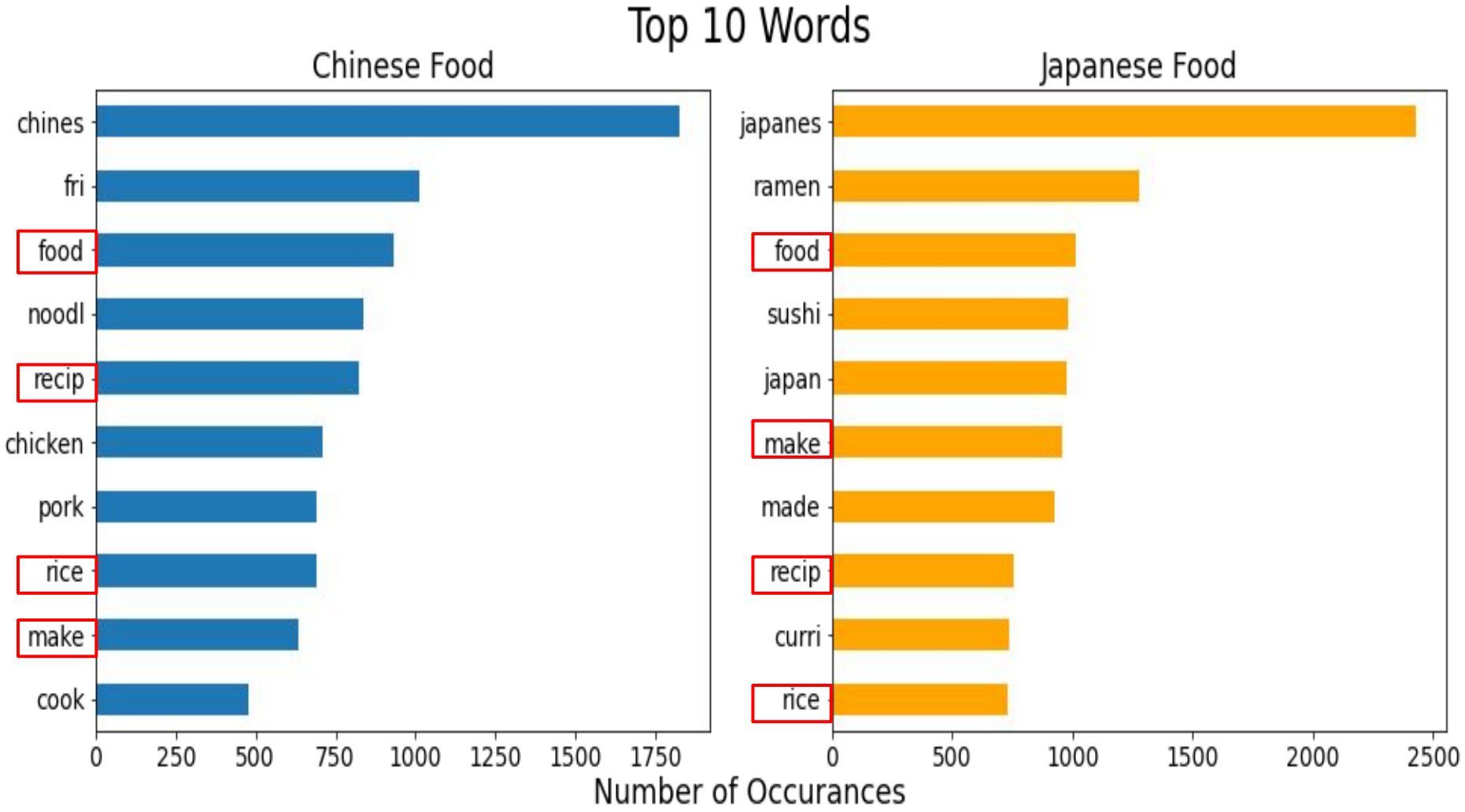


Ingredients

- RegexpTokenizer()
- WordNetLemmatizer()
- PorterStemmer()
- Stopwords - English

Preparation

1. Create for loops to remove special characters from the title of the posts


Top 10 Words
Chinese Food
chines
fri
food
noodl
recip
chicken
pork
rice
make
cook
0
250
500
750
1000
1250
1500
1750
Japanese Food
japanes
ramen
food
sushi
japan
make
made
recip
curri
rice
0
500
1000
1500
2000
2500
Number of Occurances

Recipe 3: Modeling

Ingredients

- Transformers:
  - CountVectorizer
  - TfidfVectorizer
- Estimators:
  - LogisticRegressionCV
  - Multinomial Naive Baye
  - Bernoulli Naive Baye
  - Random Forest Classifier
- Preparation:
  - GridSearch

Recipe 4: Model Evaluation

- Highest training score: (0.9981)
  - Random Forest Classifier with CountVectorizer
- Highest Testing score (0.91)
  - Logistic Regression with TfidfVectorizer
- Most balanced model (0.0279)
  - Bernoulli Naive Baye with either vectorizer.

```
 ======Logistic Regression======
Count Vectorizer Training Score: 0.9730019284336833
Count Vectorizer Testing Score: 0.9089545844044559
Difference: 0.0640473440292274

Tfidf Vectorizer Training Score: 0.9622884079708592
Tfidf Vectorizer Testing Score: 0.910025706940874
Difference: 0.052262701029985204

 ======Multinomial Naive Bayes======
Count Vectorizer Training Score: 0.9358795800299978
Count Vectorizer Testing Score: 0.9068123393316195
Difference: 0.029067240698378294

Tfidf Vectorizer Training Score: 0.9300407113777587
Tfidf Vectorizer Testing Score: 0.895458440445587
Difference: 0.0345822709321717

 ======Bernoulli Naive Bayes======
Count Vectorizer Training Score: 0.9343796871652025
Count Vectorizer Testing Score: 0.9063838903170522
Difference: 0.027995796848150234

Tfidf Vectorizer Training Score: 0.9343796871652025
Tfidf Vectorizer Testing Score: 0.9063838903170522
Difference: 0.027995796848150234

 ======Random Forest Classifier======
Count Vectorizer GS Training Score: 0.9981787015213199
Count Vectorizer GS Testing Score: 0.8883890317052271
Difference: 0.10978966981609284
```

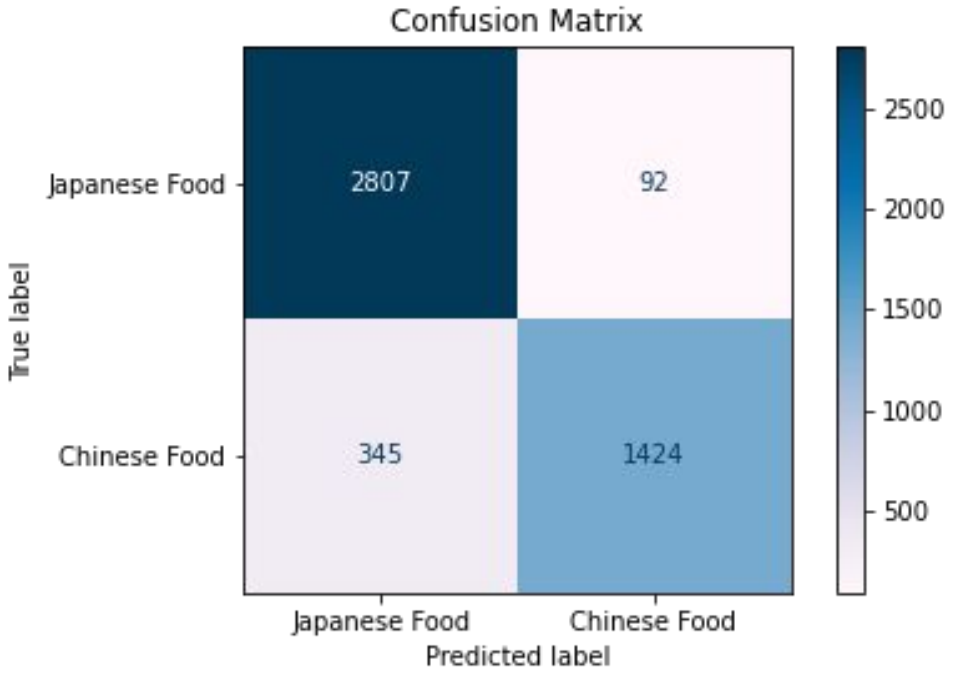
Recipe 5: Model Selection

Ingredients

- CountVectorizer with Bernoulli Naive Bayes
- Training score = 0.934
- Testing score = 0.906
- Score difference = 0.028

Recipe 6: Predictions

- Accuracy: 0.906
- Precision: 0.939
- Sensitivity & Type II Error: 0.804
- Specificity & Type I Error: 0.968
- F1-score: 0.867



Conclusion:

- For this project, we successfully created a model using CountVectorizer and Bernoulli Naive Bayes Classifier to distinguish and separate posts into their corresponding Reddits with 90% accuracy.
- One drawback for this model is that the dataset used in this project was not evenly distributed because Chinese food subreddit had fewer than 10,000 total posts by the time we created the model. Thus, we also had to limit the number of posts for the Japanese food subreddit to minimize the bias towards the majority class.
- The next step is to train the model on other subreddits and see how the model performs with a larger dataset, and more evenly distributed dataset.