Reddit is Broken

By Josh Slizinov

Agenda

- Problem Statement
- Solution Proposition
- Gathering Data & EDA
- Modeling
- Model Performance
- Improvements
- Recommendation
- Questions

Problem

Something is wrong with the reddit servers. New posts are being posted to random subreddits! Until the servers are fixed, reddit employees have to manually direct posts to the correct subreddit! It's taking too long!

I've decided to offer my pro-bono services to reddit in exchange for a data science job upon graduation.

Question

Can a model be developed that will automatically classify loose reddit posts to the correct subreddit by analyzing the textual makeup of its title and description?

Solution Proposition

Create a model that will analyze the relevant text in a reddit post and classify it as belonging to r/microgrowery or r/sandwiches.

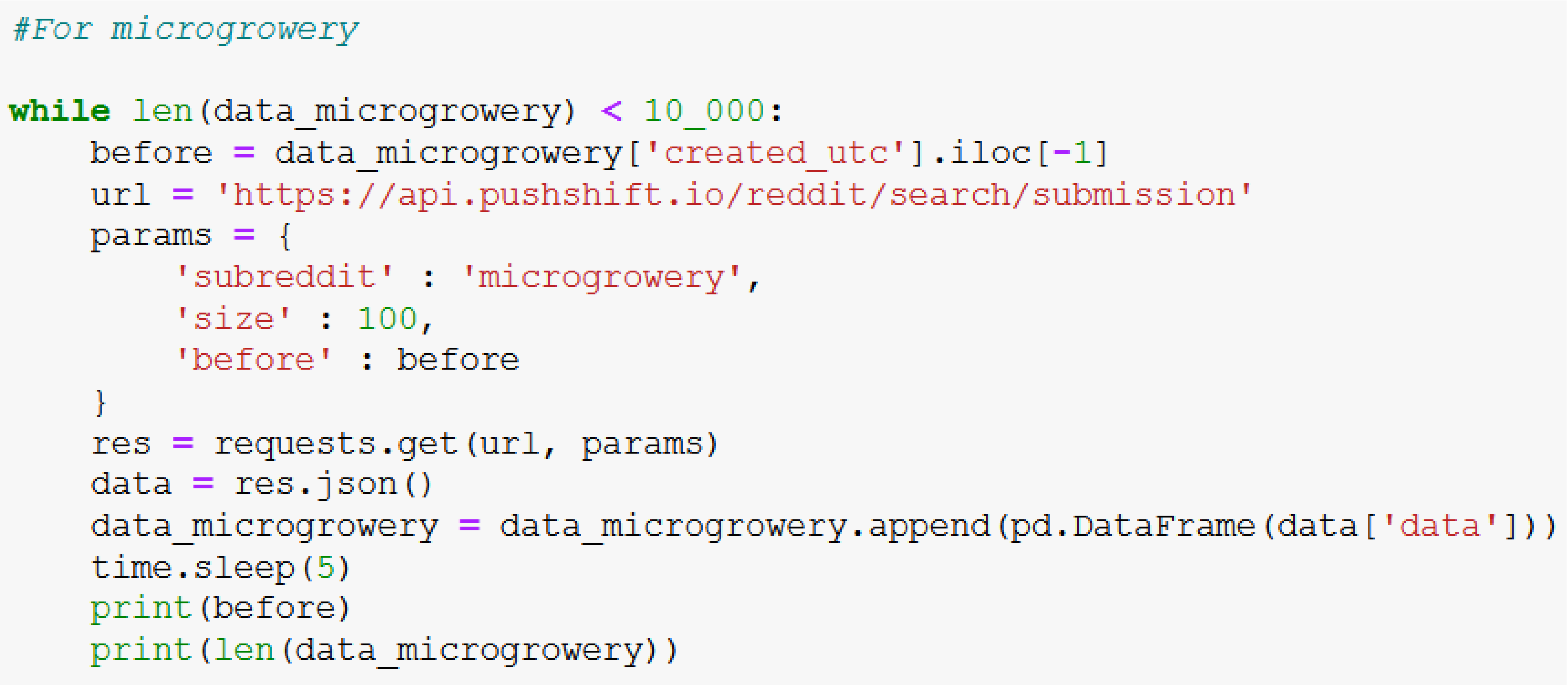
Models used: Random Forest and KNN

Gathering Data & EDA Process

1. Pushshift Reddit API to get initial data
2. Initial feature selection
3. Standardization & lemmatization
4. Distribution analysis
5. CountVectorize
6. Top word analysis
7. Sentiment analysis

Pushshift API

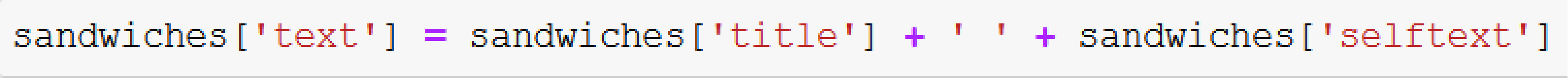
- Used requests library to gather initial 100 rows of data
- Wrote while loop to obtain remaining data
    - 10 second delay for each request



```
#For microgrowery

while len(data_microgrowery) < 10_000:
    before = data_microgrowery['created_utc'].iloc[-1]
    url = 'https://api.pushshift.io/reddit/search/submission'
    params = {
        'subreddit' : 'microgrowery',
        'size' : 100,
        'before' : before
    }
    res = requests.get(url, params)
    data = res.json()
    data_microgrowery = data_microgrowery.append(pd.DataFrame(data['data']))
    time.sleep(5)
    print(before)
    print(len(data_microgrowery))
```

Initial Feature Selection

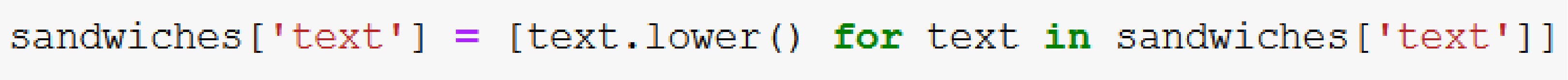
- Only using ‘selftext’ and ‘title’ columns
  - Combine



```
sandwiches['text'] = sandwiches['title'] + ' ' + sandwiches['selftext']
```

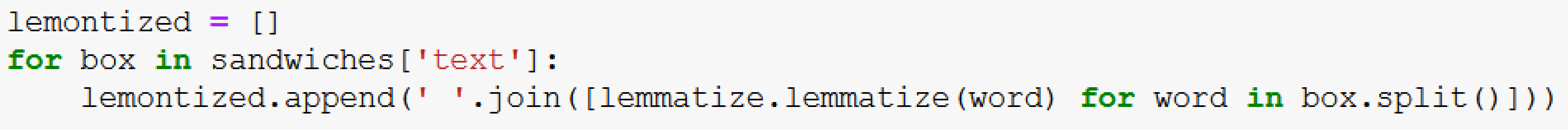
Standardization & Lemmatization

- Make all text lowercase



```
sandwiches['text'] = [text.lower() for text in sandwiches['text']]
```

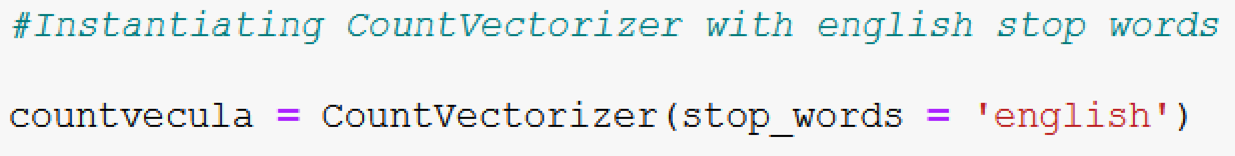
- Lemmatize using nltk WordNetLemmatizer



```
lemontized = []
for box in sandwiches['text']:
    lemontized.append(' '.join([lemmatize.lemmatize(word) for word in box.split()]))
```

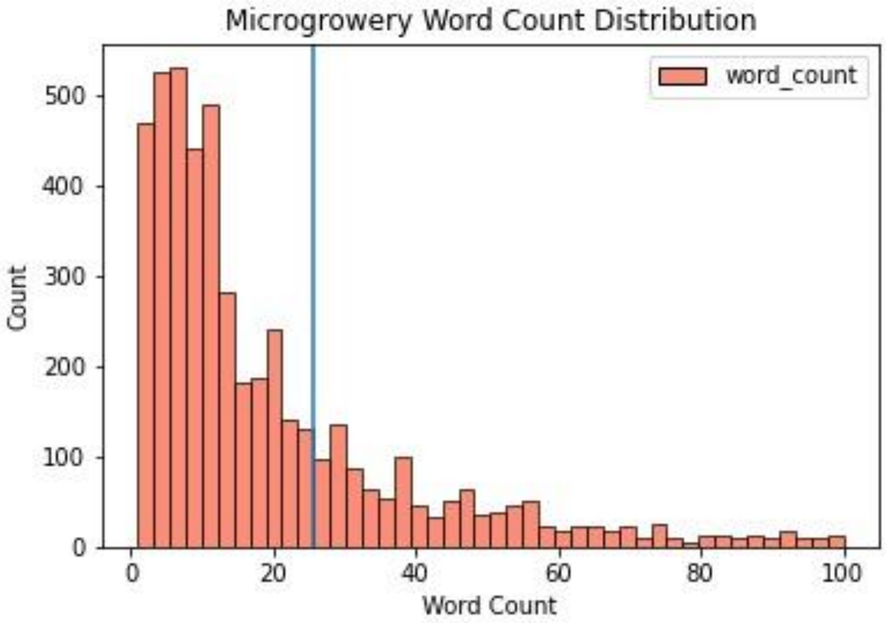
CountVectorization

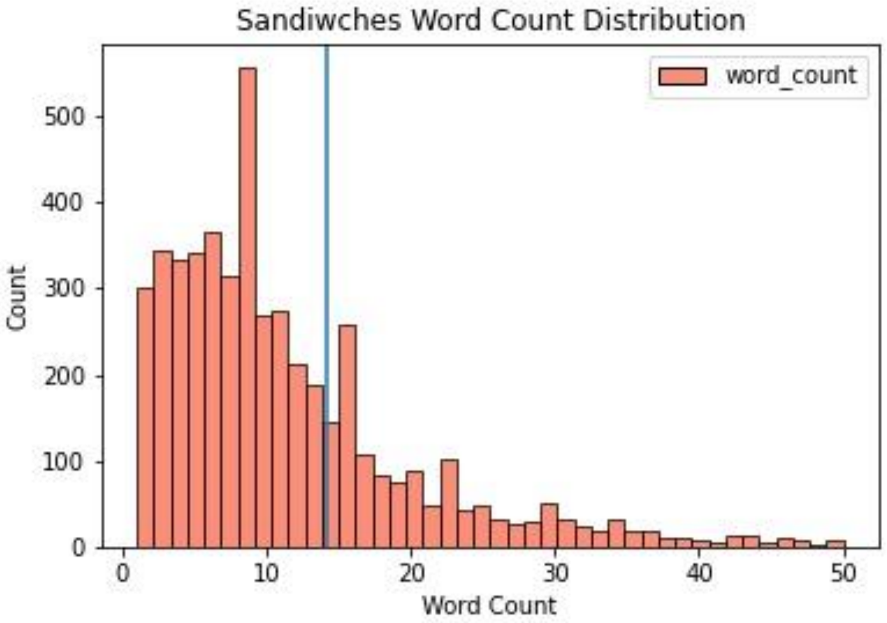
- CountVectorizer from sklearn to convert data to matrix form with English stop words



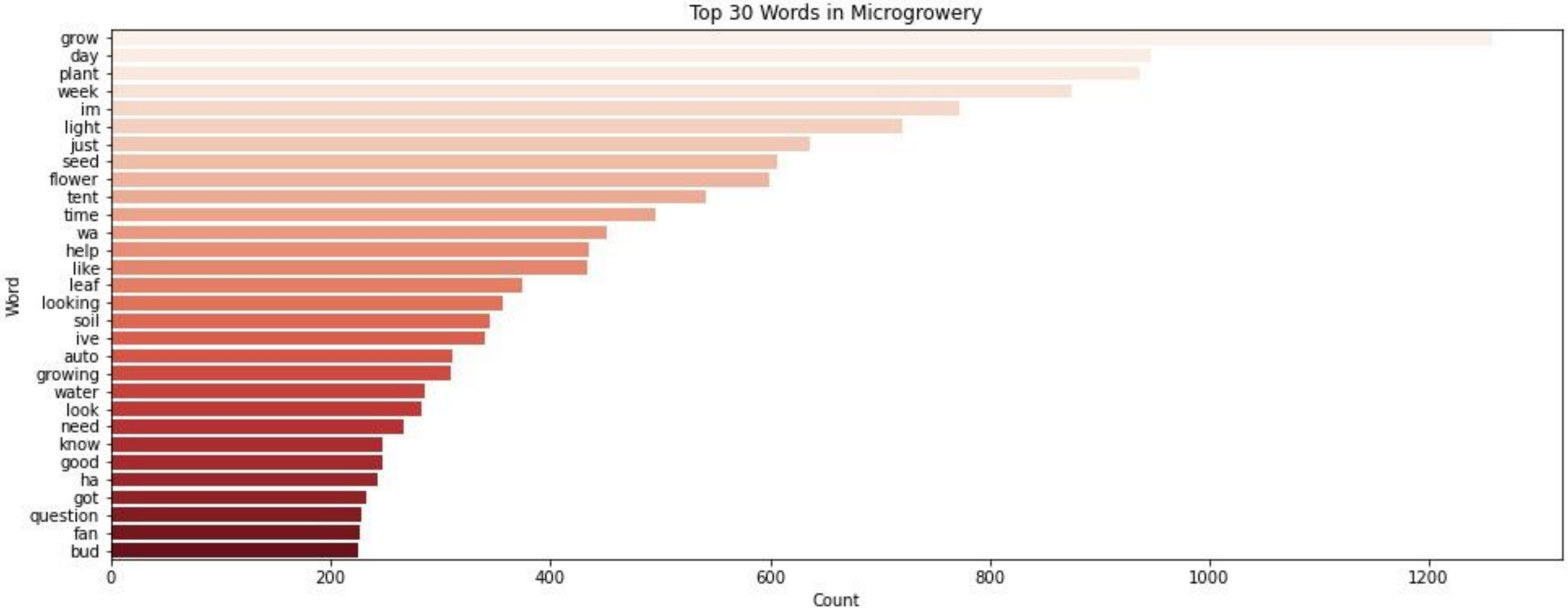
```
#Instantiating CountVectorizer with english stop words

countvecula = CountVectorizer(stop_words = 'english')
```

Distribution Analysis

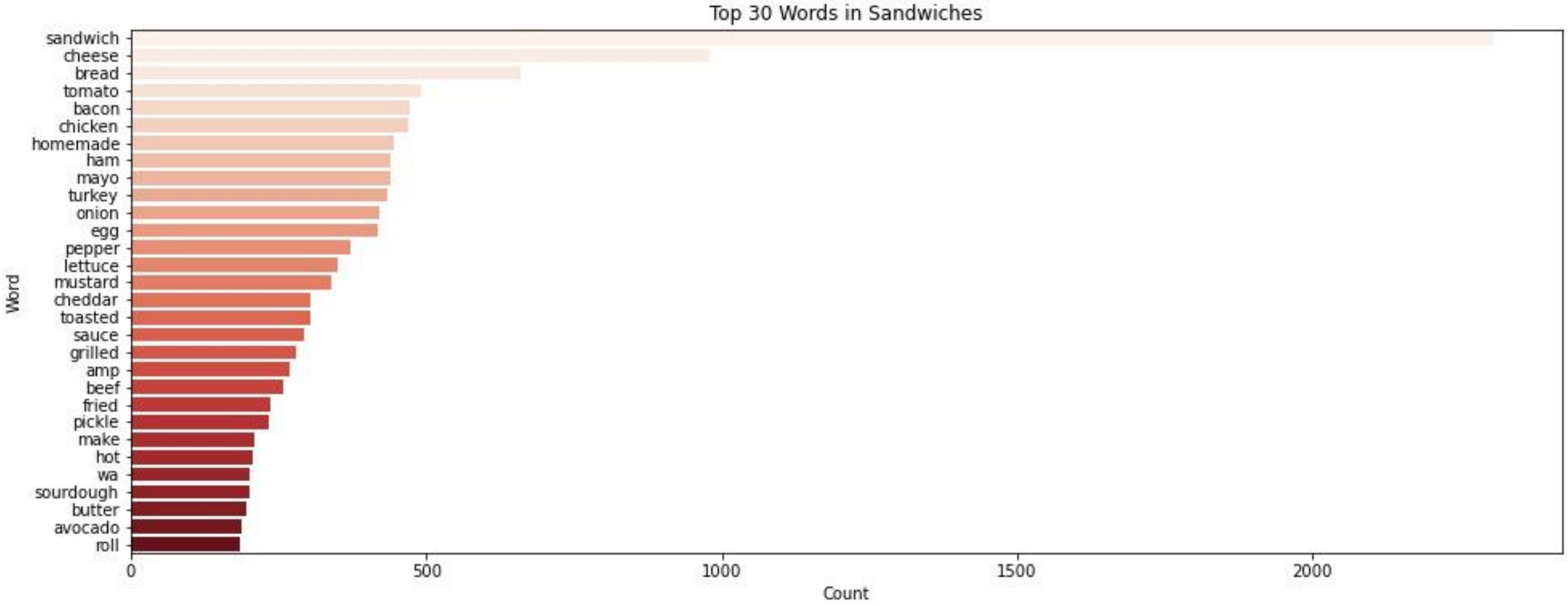




Top Word Analysis

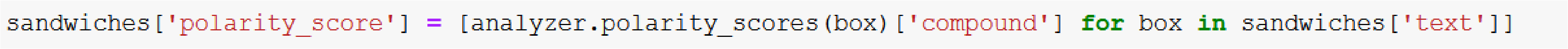


Top Word Analysis



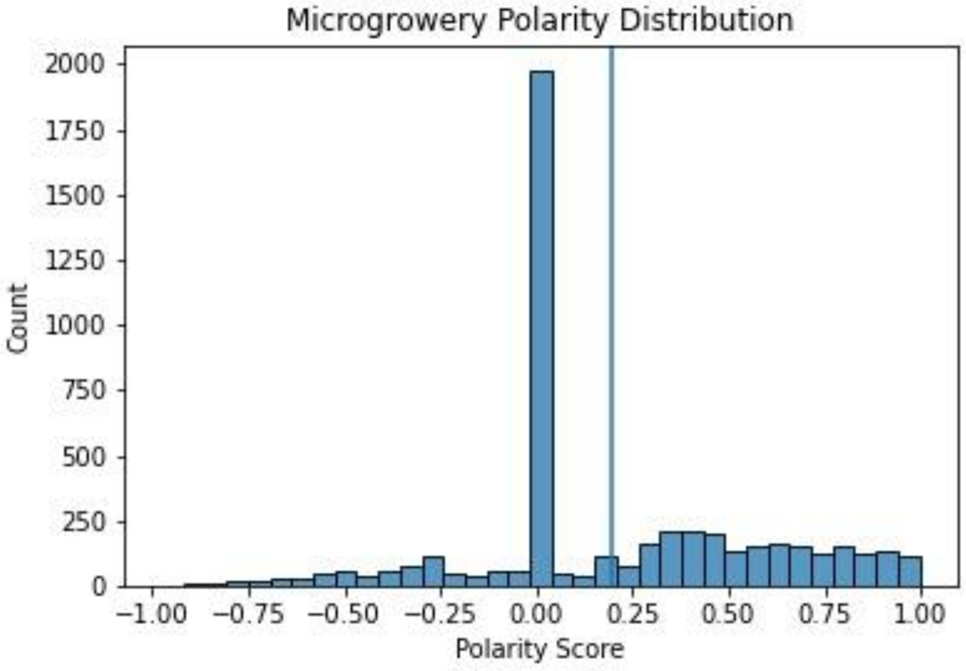
Sentiment Analysis

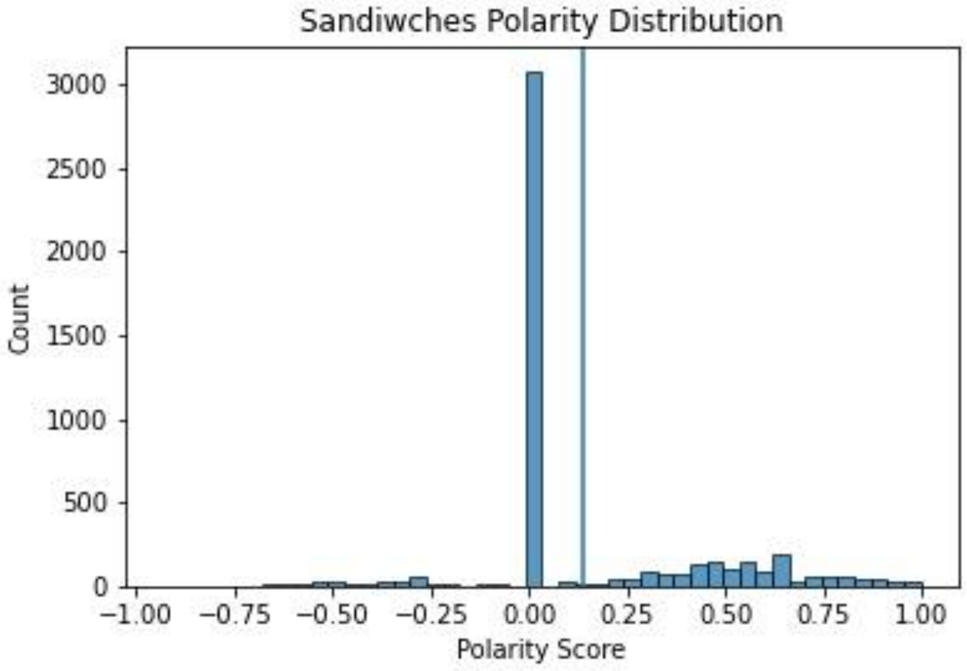
- vaderSentiment to determine polarity score for each text



```
sandwiches['polarity_score'] = [analyzer.polarity_scores(box)['compound'] for box in sandwiches['text']]
```

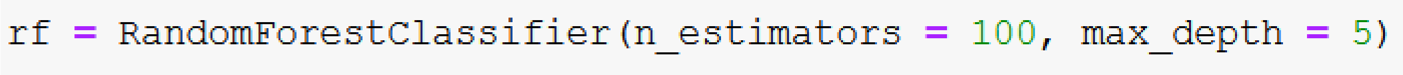
Sentiment Analysis





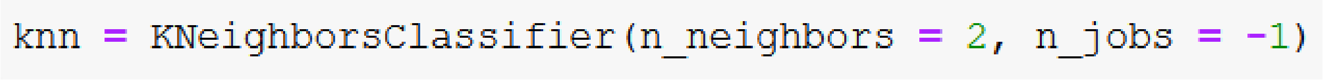
Modeling

- Random Forest



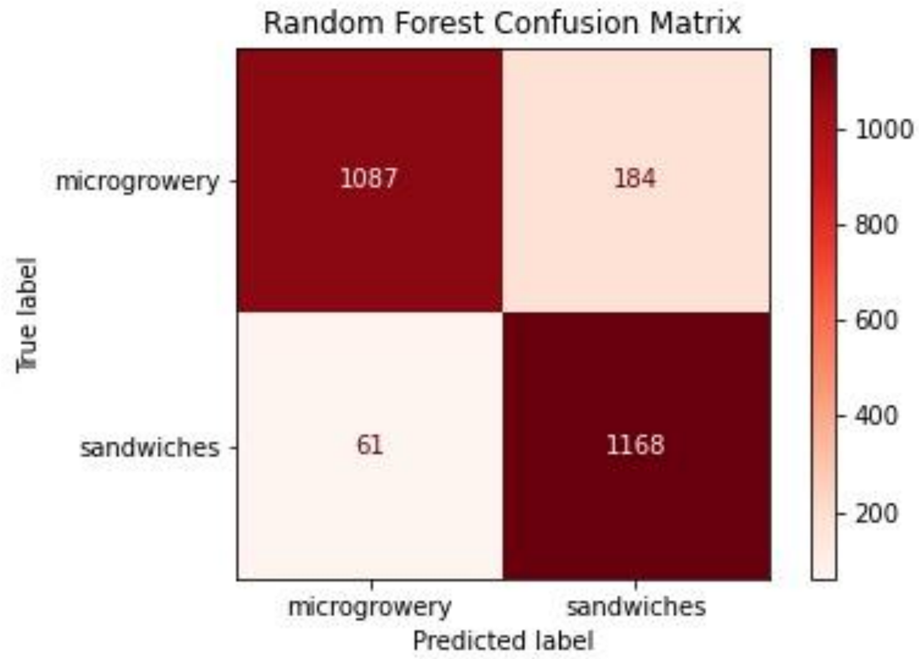
```
rf = RandomForestClassifier(n_estimators = 100, max_depth = 5)
```

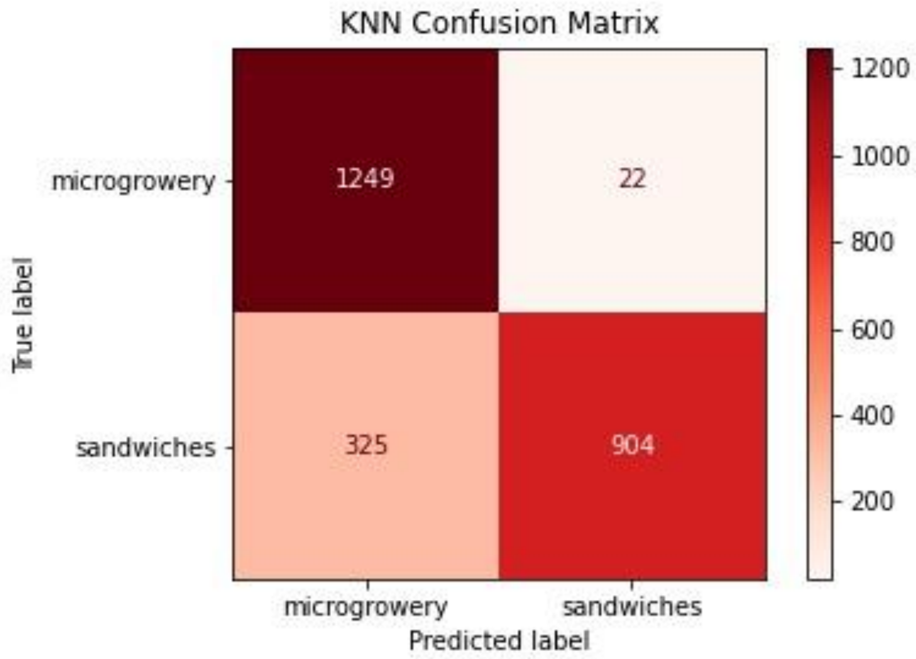
- KNN



```
knn = KNeighborsClassifier(n_neighbors = 2, n_jobs = -1)
```

Model Performance





Train accuracy: 0.9148
Test accuracy: 0.902

Train accuracy: 0.9244
Test accuracy: 0.8612

Improvements

1. GridSearch models
2. Test on other subreddits
3. Pipeline to make code more efficient
4. Extra computing power!
5. Test more models

Recommendation

Because this model IS able to predict, with high accuracy, the classification of subreddit posts based on text, I recommend that reddit employees use this model to assist them in classifying their subreddit posts until they can fix their servers.

Questions?