Project 3: Web APIs and NLP

Anand Ramchandani

Problem Statement

Collect posts from 2 subreddits and use NLP to train a classifier to distinguish between posts from the subreddits 'r/beer' and 'r/wine'

Plan to Achieve Project Goals

1. Gather data using Reddit's API
2. Clean Data
3. Data Exploration
4. Identify relevant features and conduct feature engineering
5. Vectorize using CountVectorizer and TF-IDF
6. Modelling using Logistic Regression, Multinomial Naive Bayes and Random Forest
7. Refinements and Hyper Parameter Tuning
8. Re-evaluate and select best model

REASON FOR CHOICE

1. *Both are commonly consumed alcoholic beverages*
2. *Both have passionate followings*
3. *Both have lower alcohol %*
4. *Similar words (bottle, taste, etc.)*
5. *Different enough to allow for classification, similar enough to be challenging*

BUSINESS RELEVANCE

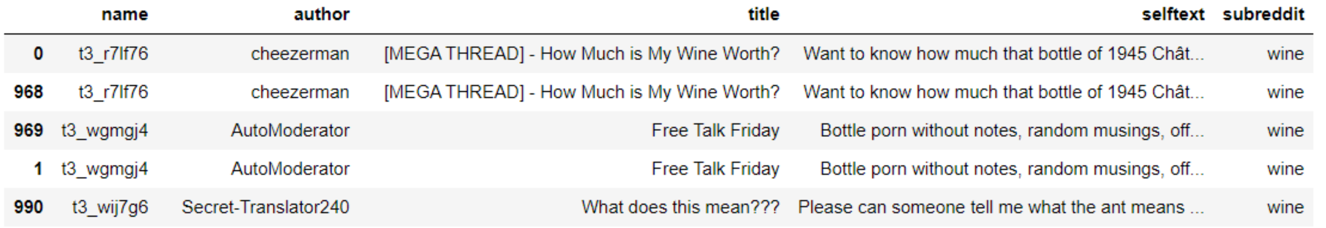
- Aid search engines in identifying between the 2 beverages
- Model could be used for businesses sorting out/filtering email/queries/feedback
- Example - Could be used to see how successful a marketing campaign was by filtering online comments after beer or wine commercial airs

Data Extraction/ Web Scraping

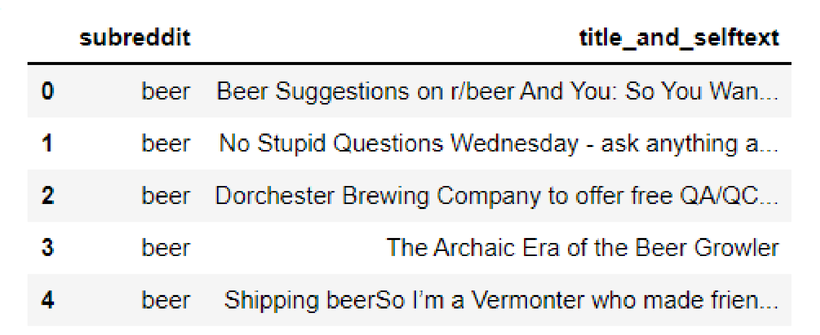
- Downloaded user posts and comments using pushshift.io Reddit API
- Identified urls
- Called API
- Converted API output to .json
- Gathered 1,000 posts per subreddit

Data Cleaning / Feature Engineering

1. Check for duplicates and remove them
2. Check for null values
3. Merge title and selftext posts into 1 column (this also addressed a lot of missing data seen in selftext)



	name	author	title	selftext	subreddit
0	t3_r7lf76	cheezerman	[MEGA THREAD] - How Much is My Wine Worth?	Want to know how much that bottle of 1945 Chât...	wine
968	t3_r7lf76	cheezerman	[MEGA THREAD] - How Much is My Wine Worth?	Want to know how much that bottle of 1945 Chât...	wine
969	t3_wgmgj4	AutoModerator	Free Talk Friday	Bottle porn without notes, random musings, off...	wine
1	t3_wgmgj4	AutoModerator	Free Talk Friday	Bottle porn without notes, random musings, off...	wine
990	t3_wij7g6	Secret-Translator240	What does this mean???	Please can someone tell me what the ant means ...	wine



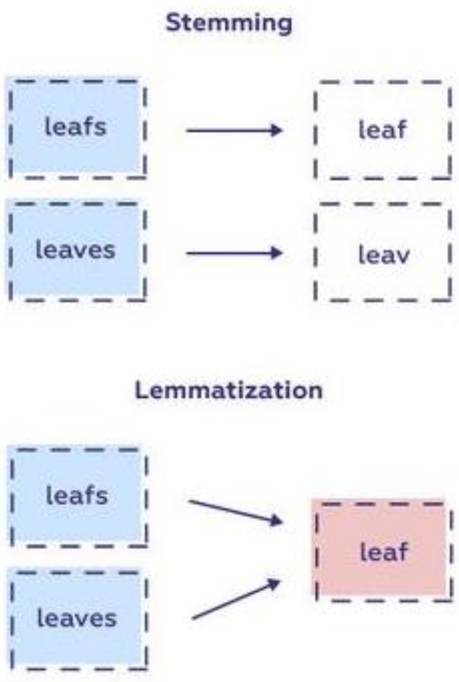
	subreddit	title_and_selftext
0	beer	Beer Suggestions on r/beer And You: So You Wan...
1	beer	No Stupid Questions Wednesday - ask anything a...
2	beer	Dorchester Brewing Company to offer free QA/QC...
3	beer	The Archaic Era of the Beer Growler
4	beer	Shipping beerSo I'm a Vermonter who made frien...

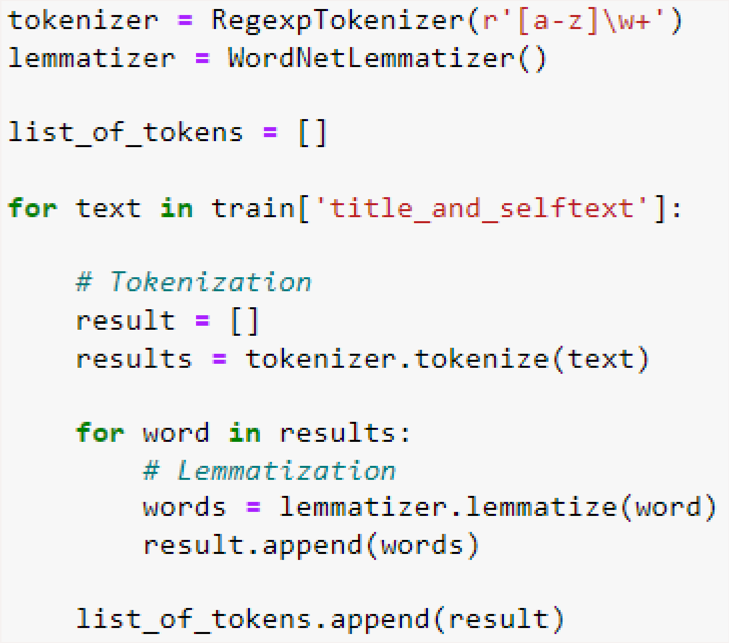
Data Cleaning – Pre Modelling Cleaning

1. Make text lowercase
2. Remove HTML special entities
3. Remove Hyperlinks
4. Remove Punctuation
5. Split 's, 't, ‘ve
6. Remove whitespace
7. Remove characters beyond Basic Multilingual Plane (BMP) of Unicode
8. Removal of Stop Words

Tokenization and Lemmatization

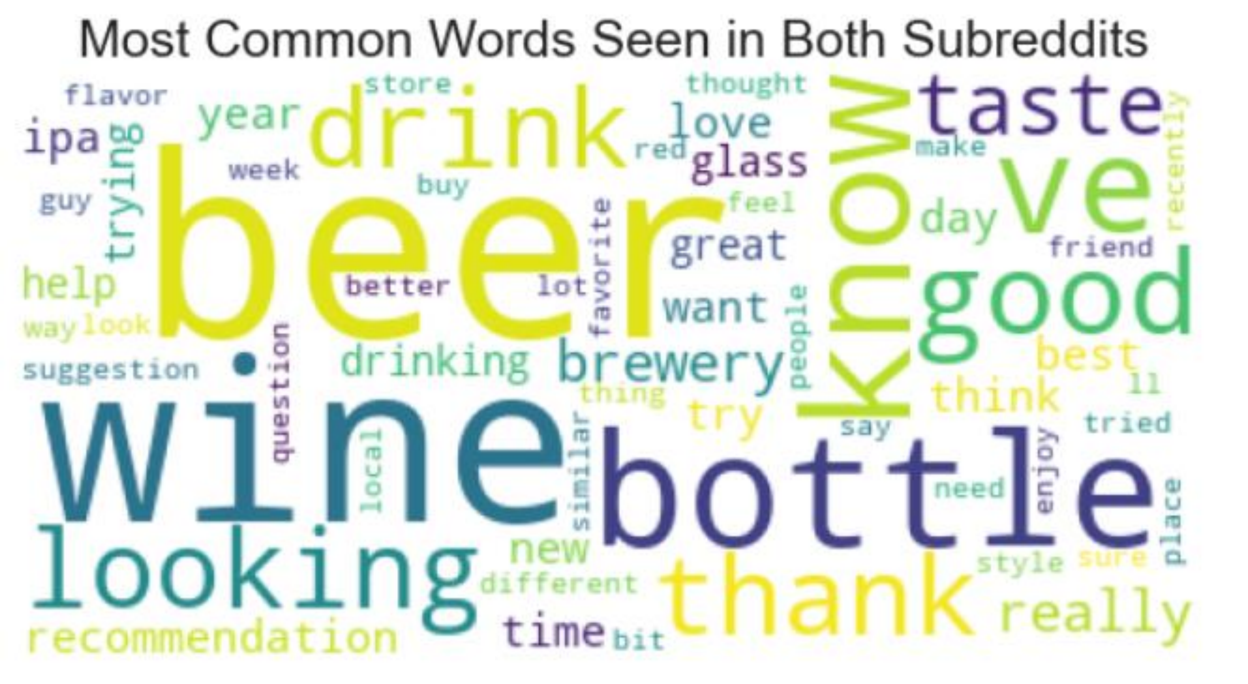
Using this method to shorten root words and remove redundancy



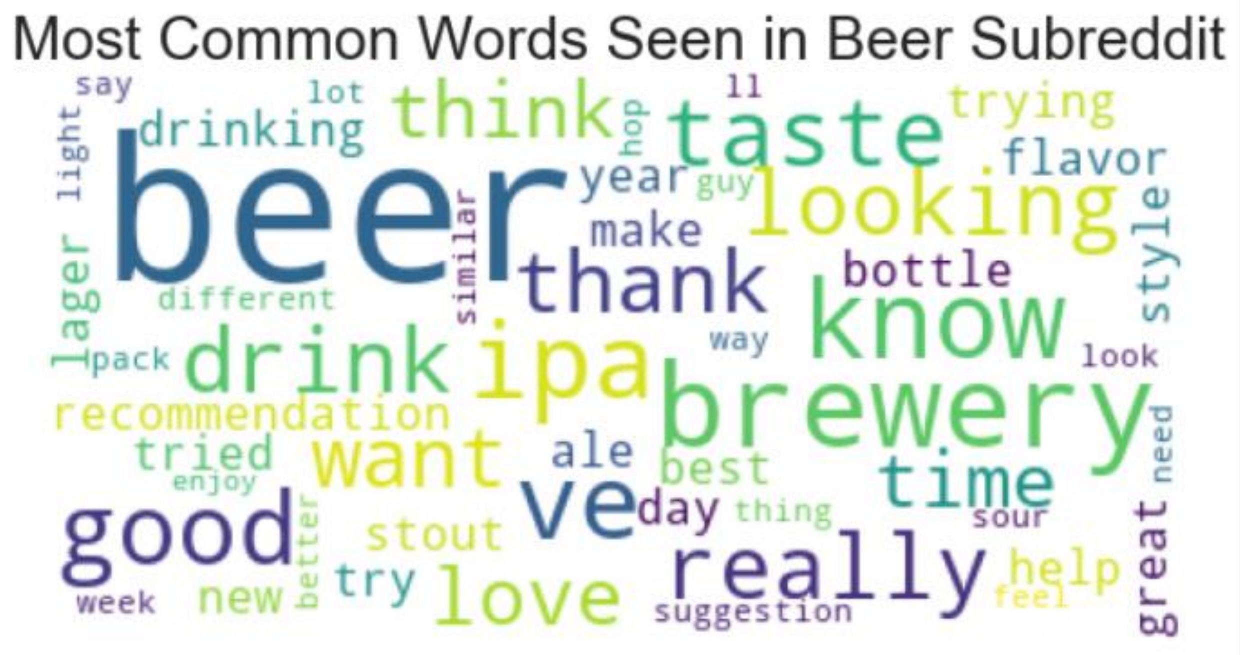


```
tokenizer = RegexpTokenizer(r'[a-z]\w+')
lemmatizer = WordNetLemmatizer()

list_of_tokens = []

for text in train['title_and_selftext']:

    # Tokenization
    result = []
    results = tokenizer.tokenize(text)

    for word in results:
        # Lemmatization
        words = lemmatizer.lemmatize(word)
        result.append(words)

    list_of_tokens.append(result)
```

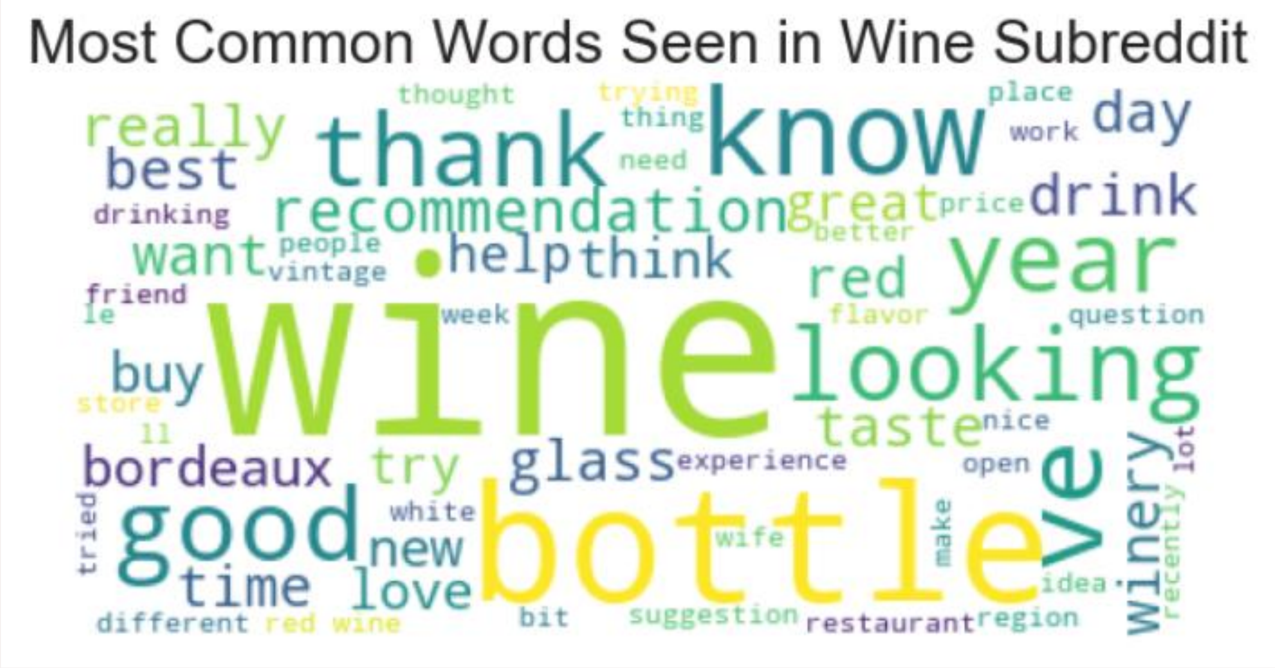
Word Cloud – Common Words Seen in Both Subreddits



Word Cloud – Common Words Seen in Beer Subreddits



Word Cloud – Common Words Seen in Wine Subreddits


Most Common Words Seen in Wine Subreddit

Data Modelling Plan

Vectorizers used

- CountVectorizer
- TF-IDF Vectorizer

Models Used

- Logistic Regression
- Multinomial Naïve Bayes
- Random Forest

Compared the following Models:

- Model 1: CountVectorizer with Logistic Regression
- Model 2: CountVectorizer with MultinomialNB
- Model 3: CountVectorizer with Random Forest
- Model 4: TF-IDF Vectorizer with Logistic Regression
- Model 5: TF-IDF Vectorizer with MultinomialNB
- Model 6: TF-IDF Vectorizer with Random Forest

Data Modelling Plan – How to analyse Models selected

1. Conduct GridSearch CV (pipe vectorizer with model chosen)
2. Evaluate accuracy score and prediction on test data set
3. Re-evaluate after Hyperparameter Tuning
4. Draw Confusion Matrix
5. Calculate Sensitivity, Specificity, Precision
6. Draw ROC curve
7. Make evaluation of which model is best

Model Evaluation

Comparison of Models based on Accuracy and AUC Score on Test Data

	CountVectorizer				TF-IDF			
	Accuracy (Train)	Accuracy (Test)	%	ROC-AUC Score (Test)	Accuracy (Train)	Accuracy (Test)	%	ROC-AUC Score (Test)
Logistic Regression	0.9987	0.9644	3.5035	0.99	0.9885	0.9585	3.0290	1
Multinomial Naïve-Bayes	0.9936	0.9663	2.7460	0.99	0.9834	0.9585	2.5246	0.99
Random Forest	1.0000	0.9508	4.9223	0.98	1.0000	0.9611	3.8860	0.98

Model Evaluation

Comparison of Models based on Sensitivity, Specificity and Precision

	CountVectorizer			TF-IDF		
	Sensitivity	Specificity	Precision	Sensitivity	Specificity	Precision
Logistic Regression	0.938	0.982	0.974	0.925	0.982	0.974
Multinomial Naïve-Bayes	0.969	0.965	0.951	0.944	0.969	0.956
Random Forest	0.931	0.987	0.98	0.931	0.982	0.974

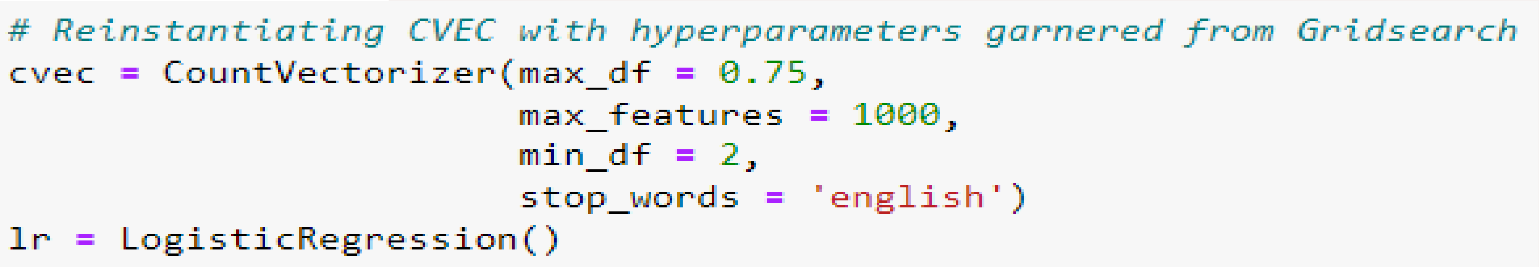
- Choose Sensitivity/Recall if the idea of false positives is far better than false negatives
- Choose Precision if you want to be more confident of your true positives
- Choose Specificity if you want to cover all true negatives, meaning you don’t want any false alarms, you don’t want any false positives

CountVectorizer with Logistic Regression – Model Chosen

Reason for choice:

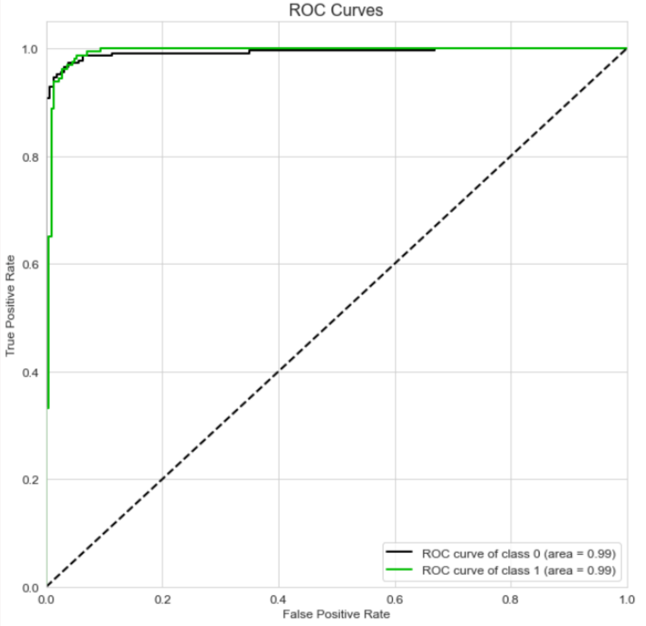
- Good Accuracy (Test) Score - 0.9644
- 0.99 – Good ROC-AUC Score
- Precision would best suit our needs and the model has 0.974 precision

Hyperparameters used for CountVectorizer with Logistic Regression



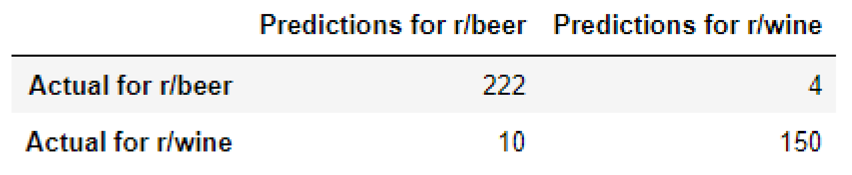
```
# Reinstantiating CVEC with hyperparameters garnered from Gridsearch
cvec = CountVectorizer(max_df = 0.75,
                       max_features = 1000,
                       min_df = 2,
                       stop_words = 'english')
lr = LogisticRegression()
```

ROC Curve
- CountVectorizer with Logistic Regression

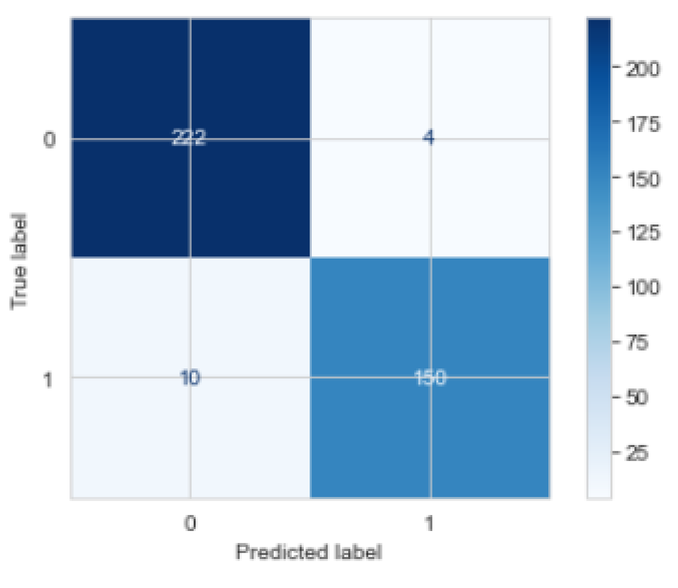


- The CountVectorizer with Logistic Regression has an ROC of 0.9 for both r/beer and r/wine
- Our model is able to properly classify the post between these two different subreddits

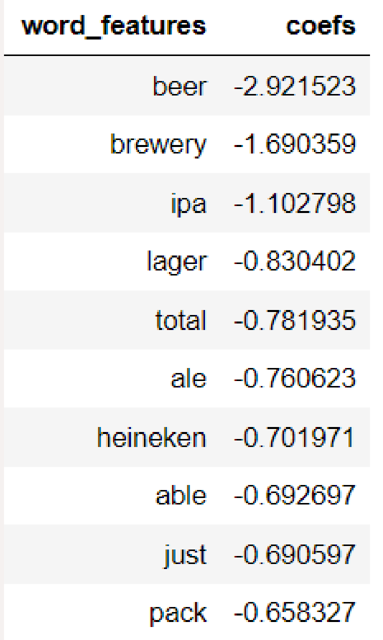
Confusion Matrix
- CountVectorizer with Logistic Regression



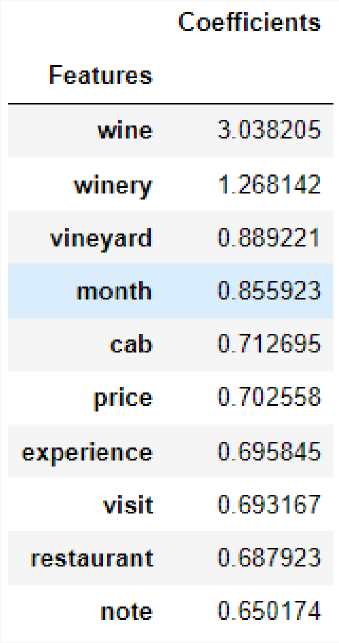
	Predictions for r/beer	Predictions for r/wine
Actual for r/beer	222	4
Actual for r/wine	10	150



Feature Importance in r/beer and r/wine - CountVectorizer with Logistic Regression



word_features	coefs
beer	-2.921523
brewery	-1.690359
ipa	-1.102798
lager	-0.830402
total	-0.781935
ale	-0.760623
heineken	-0.701971
able	-0.692697
just	-0.690597
pack	-0.658327



	Coefficients
Features	
wine	3.038205
winery	1.268142
vineyard	0.889221
month	0.855923
cab	0.712695
price	0.702558
experience	0.695845
visit	0.693167
restaurant	0.687923
note	0.650174

Conclusions

- CountVectorizer with Logistic Regression chosen as most apt model for our classification purposes
- Key words between the 2 subreddits help immensely in message classification
- Differences outweigh similarities
- Can be a useful tool to aid marketing or strategy
- Model works well but will fail if message is too general

Recommendations

- Improve removal of noise words
- Try more models
- Collect more training data
- Can do more data cleaning (e.g. increase number of stop words)
- Better Gridsearching methods to optimize model selection
- May not be as accurate in long run if new words used
- We could test this model on similar alcohol subreddits like whiskey, gin, vodka etc.

Thanks for listening!

Any questions?