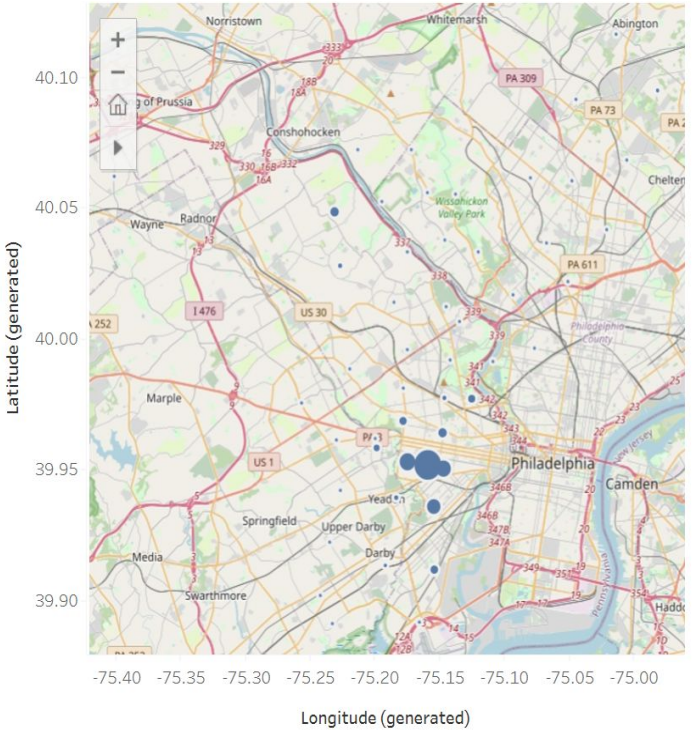
Models to predict reviews for Philadelphia Restaurants

Presented by: Anita Durg
Last Updated: Jan 5, 2023

Objective

How Philadelphia can improve their restaurant quality based on the yelp reviews?

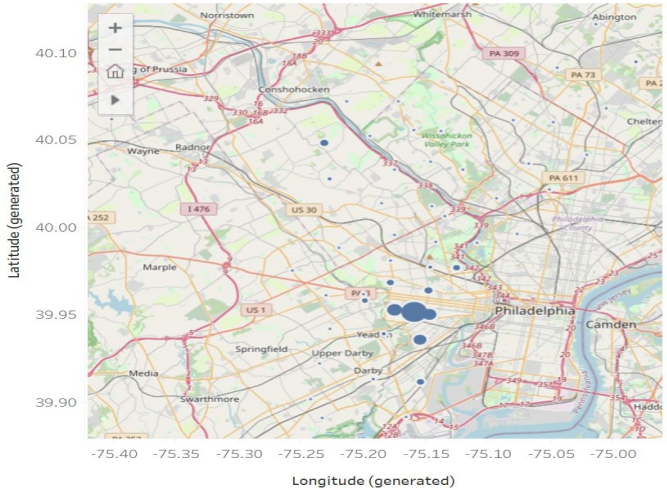


Data Science Pipeline

- 1. Sourcing and wrangling yelp reviews data
- 2. Exploratory Data Analysis
- 3. Preprocessing and applying Baseline modeling(LogisticRegression)
- 4. Apply more algorithms such as RandomForest and AdaBoostClassifier
- 5. Cross-validate with StratifiedKFold to test the ability of the modelto predict new data.
- 6. Hyperparameter tuning for better performance and fitted with test set
- 7. Choose the best model out of three.

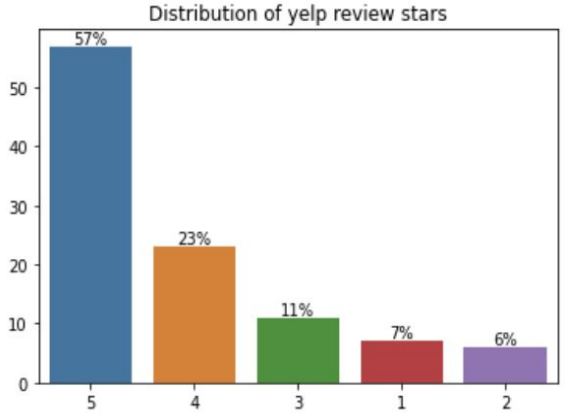
Restaurant Density

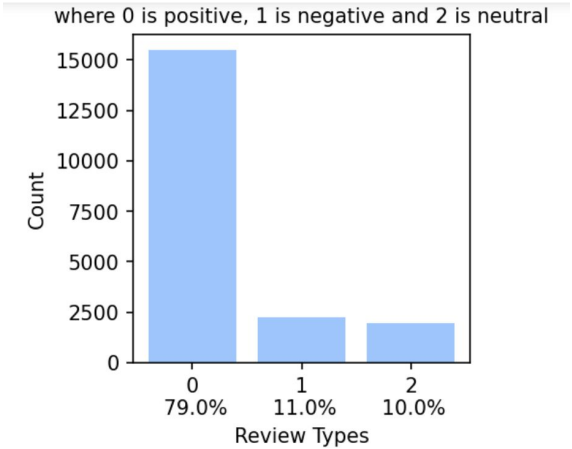
Observed lot of yelp reviews are from the central city of philadelphia.

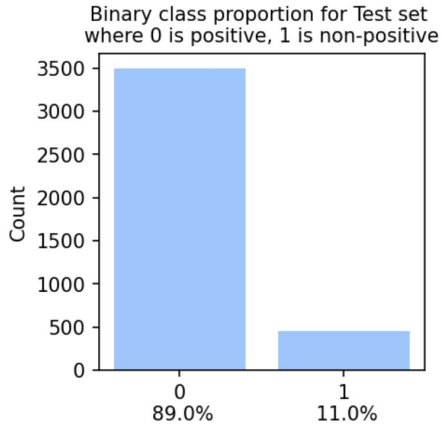



```
Out[5]: 19107    38393
        19103    22516
        19147    19999
        19106    13943
        19104     9503
        19102     9241
        19123     8866
        19148     7670
        19130     7291
        19125     6412
        19146     4952
        19128     3987
        19127     3733
        19145     3231
        19122     3130
        19143     2226
        19149     2050
        19153     1919
        19114     1759
```

Review stars proportions

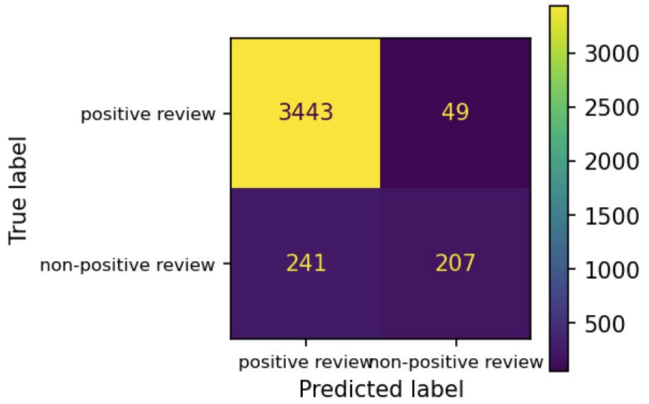






There are more reviews for 4 and 5 stars. 1 & 2 stars have very less reviews.

Classification Reports



	precision	recall	f1-score	support
0	0.93	0.99	0.96	3492
1	0.81	0.46	0.59	448
accuracy			0.93	3940
macro avg	0.87	0.72	0.77	3940
weighted avg	0.92	0.93	0.92	3940

LogisticRegression Model was our baseline model. Train Test split was done with 80/20. Followed by application of Term Frequency-Inverse Document Frequency(**TF-IFD**) on the dataset.
We notice a low Recall Score which means there are more false positive reviews. In this case, we want to increase the recall score prediction as we need to predict the negative reviews correctly to help restaurants.

Preprocessing and Hyperparameter Tuning

Models	Recall Score(basemodels)	Recall(hypertuned)	Pipeline Steps	Best Parameters
LogisticRegression	0.46	0.70	Pipeline(steps=[('o', SMOTE(random_state=42)), ('u', RandomUnderSampler(random_state=42)), ('m', LogisticRegression(C=100, max_iter=3000, solver='newton-cg'))])	{'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
RandomForestClassifier	0.18	0.49	Pipeline(steps=[('o', SMOTE(random_state=42)), ('u', RandomUnderSampler(random_state=42)), ('m', RandomForestClassifier(max_features='sqrt', n_estimators=1000, random_state=0))])	{'max_features': 'sqrt', 'n_estimators': 1000}
AdaBoostClassifier	0.50	0.78	Pipeline(steps=[('o', SMOTE(random_state=42)), ('u', RandomUnderSampler(random_state=42)), ('m', AdaBoostClassifier(learning_rate=0.8, n_estimators=88))])	{'n_estimators': 88, 'learning_rate': 0.8}

Applied StartifiedKFold Cross-validation as the target labels are highly imbalanced. Three classification algorithms were implemented with two sampling techniques(OverSampling - SMOTE and RandomUnderSampling) on each.

Conclusions

- AdaBoost Classifier seems to performing better.
- There is more scope for refining this entire project.

Acknowledgements

- DrivenData.org
- Springboard Data Science Career Track
- My mentor A J Sanchez
-