Improve Sentiment Classification of Restaurant Reviews with Machine Learning

Springboard - DSC

Capstone Project 2

- by Anita Durg

- January 2023

Introduction

There are many platforms like Yelp where consumers post their experiences about the restaurants which will help both the merchants as well as the prospective consumers. We wanted to analyze the reviews for restaurants based in the city of Philadelphia and predict the class(Positive as 0 and Negative as 1) by implementing a few Machine learning

algorithms. The goal is to predict class 1 as accurately as possible in order to help merchants on the platform to improve their services

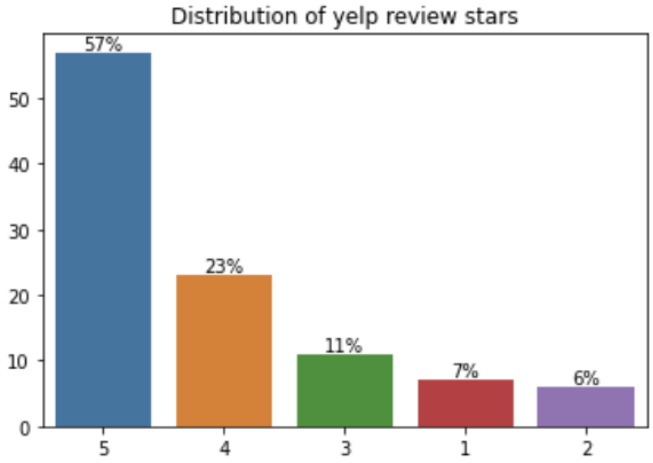
https://github.com/anidurg/Springboard_2022/tree/main/Capstone%20Project%202

Approach

Data Acquisition and Wrangling

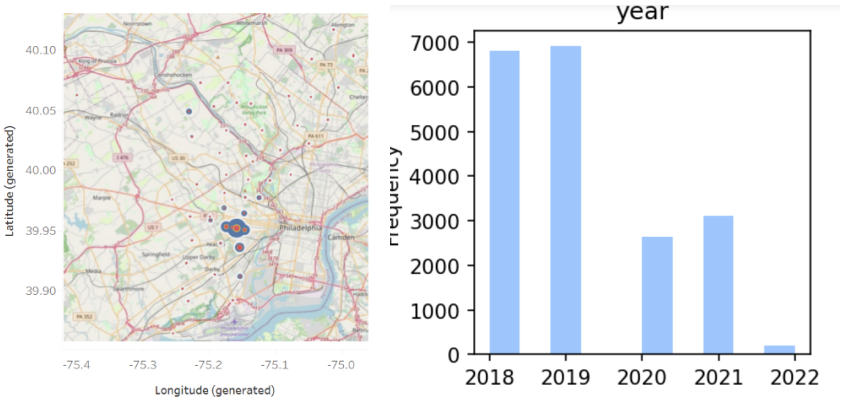
I used two datasets viz., yelp_academic_dataset_business.json and yelp_academic_dataset_review.json. Both were obtained from drivendata.org.

Due to the sheer size of the dataset, I concentrated to analyze on only one city Philadelphia as this had more reviews and I analyzed the reviews from 2018-2022. Merged both datasets. Changed the date column type to date format. Ensured no duplicated values and no null values. Visualized the distribution using a bar chart of the review stars whose values are ranging from 1 to 5.



Story Telling and Inferential Statistics

Based on the above graph, we notice that people are generous in posting their reviews when they are satisfied. Thereby, we noticed more positive reviews and very few negative reviews. The data is certainly an imbalanced dataset. I observed that lot of reviews were coming from the restaurants which are located in the central region. Used tableau for this.



I observed that more reviews are available for 2018-2019. Very less for 2020-2021. I am speculating this could be due to covid cases. After analyzing the columns, I decided to keep only the Text and review stars columns for further sentiment analysis. Review star is our target column.

Baseline Modeling

Implemented Logistic Regression algorithm as a baseline model. To efficiently and accurately identify the sentiment in restaurant reviews, word embedding techniques such as TF-IDF (which is a statistical measure used to determine the mathematical significance of words in documents is used to represent words mathematically).

The following are the steps:

a. Stratified Train test split was done with an 80/20 split size.
b. Fit the train set with tf-idf object. Transform both the test and train set
c. Fit the baseline model with multi-class as one-versus-rest on the transformed feature and predict the score. Evaluation scores were assessed with a confusion matrix and classification report function which showed low recall (sensitivity or true positive rate).
d. Followed the same steps as above for binary classification and scores were obtained. The training score is almost 100% which is a sign of overfitting. The recall score was 0.46

Extended Modeling

Because the target column is highly imbalanced, we implemented sampling techniques. Applied two types viz., SMOTE and RandomUnderSampling. Then these sampled data were fit with the baseline model and reports were evaluated. Three algorithms were applied viz., RandomForestClassifiers, AdaBost, and XGBoost as we wanted to see if any other model might be better which can predict the scores

accurately or which can bring up the recall score. Observed that AdaBoost and XGboost performed better than the other two. AdaBoost scores were the best. Stratified CrossValidation was done for all the models and results were similar. Feature interpretability was performed for three models to see which feature is responsible for the predictions. For logisticRegression, used coef_, RandomForest-feature_importances_, and SHAP for XGBoost.

	No technique(imbalanced)	1	0.71	**0.50**	0.59	448
AdaBoost	SMOTE	1	0.47	**0.74**	0.58	448
	Undersampling	1	0.39	**0.83**	0.53	448
accuracy before					0.92	3940
accuracy After SMOTE					0.88	3940
accuracy After Undersampling					0.83	3940
	No technique(imbalanced)	1	0.76	**0.52**	0.62	448
XGBoost	SMOTE	1	0.63	**0.71**	0.67	448
	Undersampling	1	0.42	**0.88**	0.57	448
accuracy before					0.93	3940
accuracy After SMOTE					0.92	3940
accuracy After Undersampling					0.85	3940

Hyperparameter tuning was implemented for each model to improve their Recall score so that models can predict the minority class accurately.

Findings

Summary of the four models:

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	0.52	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	0.79	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}
XGBoostClassifier	0.52	0.71	Pipeline(steps=[('o', SMOTE(random_state=42)), ('u', RandomUnderSampler(random_state=42)), ('m', XGBoostClassifier(learning_rate=0.3, n_estimators=100))])	{'n_estimators': 100, 'learning_rate': 0.3}

- AdaBoost performed the best as the recall scores for the minor target class scored high with sampling and tuning.
- Higher the Recall (sensitivity or True positive rate) score, the better the chances that businesses can improve their quality.
- I was hoping that XGBoost would perform the best with a better Recall score than AdaBoost. Due to time constraints, I could not do hyperparameter tuning on XGBoost model.

Conclusions and Future Work

Fours models were implemented and results were predicted accurately out of which Adaboost classifier performed the best. Recall scores improved from 50% to 79%. This should help the restaurant to improve its quality if the prediction shows a negative review score.

Future work:

- More work on text cleaning using re library
- More work on fine-tuning the hyperparameters
- Implement Word2Vec instead of TF-IDF

Recommendations for the clients

- For imbalanced learning, recall is typically used to measure the coverage of the minority class. Since our data is highly imbalanced, I highly recommend a client consider the model which produces a good Recall score which should be close to 1.0.
- We want to ensure that if the review text is negative, then the model should predict that correctly as a negative review and predict the low review star rather than predicting a high star. This will help consumers to look for a better restaurant. At the same, this should help the restaurant to better their services.

Consulted Resources

http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://shap.readthedocs.io/en/latest/index.html

https://github.com/slundberg/shap

https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/

https://shap.readthedocs.io/en/latest/example_notebooks/tabular_examples/tree_based_models/Basic%20SHAP%20Interaction%20Value%20Example%20in%20XGBoost.html