Business Reviews and Data Analysis on Yelp Dataset

Harshika Shrivastava
016019120
San Jose State University
harshika.shrivastava@sjsu.edu

Sanika Vijaykumar Karwa
016688815
San Jose State University
sanika.vijaykumarkarwa@sjsu.edu

Swapna Kotha
016007979
San Jose State University
swapna.kotha@sjsu.edu

Tirupati Venkata Sri Sai Rama Raju Penmatsa
016037047
San Jose State University
tirupativenkatasrisairama.penmatsa@sjsu.edu

Abstract—**Feedback and reviews are assets for any successful business, it helps in increasing sales and provide scope for improvement. With the Yelp dataset, we attempted to use machine learning models in this study to comprehend customer evaluations of businesses. To anticipate user ratings based on user reviews, we have tried to adopt a variety of models. Then, to comprehend how various methods, including linear models vs. other models, function, we tried to develop several models for the same objective. Additionally, we experimented with datasets and used a variety of methodologies in the research to try and improve the models' efficacy.**

Keywords—tagging, machine learning models, yelp, review analysis, sentiment analysis, false review detection

I. Introduction

In this project, we've used a variety of strategies to forecast what ratings a user will provide based on the reviews they've read. Other features including sentiment analysis, false review identification, and category prediction based on reviews have also been incorporated. This project's major goal is to comprehend how learning functions. To evaluate the Yelp dataset[1] and comprehend the pattern in the data, we have put data preparation techniques into practice. We have also tried to evaluate different patterns in the data to see if learning is possible or not. This project's major goal is to put the two-step learning strategy we acquired in class into practice and use theoretical knowledge in experiments. We have used a variety of algorithms, including decision trees, random forests, and gradient-boosting machines to obtain the result. Through this research, we gained actual knowledge of various machine learning principles, including data preprocessing, pattern analysis, model implementation, and model evaluation. This project served as a stepping stone for us to enhance our analytical skills because it required an intense understanding of the problem and thinking creatively to solve it. Additionally, we had to master several Python libraries and machine-learning ideas in order to construct different machine-learning algorithms.

The project has given us an in-depth understanding of machine learning and its applications. It is also interesting to learn about how a simple task such as prediction can be implemented using various algorithms and techniques. This knowledge will certainly help us gain more insights into business analytics, where the most important aspect is to understand customer behavior.

II. About the Dataset

The data that we used comes from the Yelp Dataset Challenge [1]. It is a subset of Yelp's businesses, reviews and user data that has been made publicly available for different uses. This dataset contains 5 JSON files namely, business, checkin, reviews, tip and user.

Business.json - Contains business data along with location data, attributes, and categories.

Review.json - Contains full review text data with the user_id that wrote the review and the business_id for whom the review is written.

User.json - Contains user data including the user's friend mapping and all the other metadata that is associated with the user.

Checkin.json - Contains check-ins on a business

Tip.json - Contains tips written by a user on a business. They are written so as to convey fast suggestions and are generally shorter than reviews.

For our implementation, we have converted the JSON files to csv files. For few additional tasks, we have also taken a subset of data to do additional tasks i.e we have used Californian Restaurants only for the implementation of our models.

III. Methods

In this section, we are going to discuss various concepts that we need to understand in the implementation of the project.

A. Data cleaning

The files are initially in json format and are afterwards transformed into csv files for additional use. Second, the null values that were previously contained in the dataset have been eliminated because they are comparatively inconsequential to the actual data. Thirdly, the company's top categories and related reviews have been combined. This is accomplished by use the sqldf library, which makes it simple to import the dataframe into the database and perform operations using straightforward SQL commandCs. It enables SQL queries to be used to query any Pandas Dataframe. In order to integrate the business dataset with the appropriate reviews, the CSV files are lastly combined using the Business ID as a unique identifier key.

For easier processing of models, we have used a smaller subset of data i.e. taking the business classified as “Restaurants” in state “California”.

B. *Pre-processing*

The following methods are used to pre-process the dataset:

- **TF-IDF Transformer** - This transformer stands for term frequency - inverse document frequency. The TF-IDF is a metric that assesses a word's significance to a document within a collection of papers. This is accomplished by counting the number of times a word appears in our document and paying close attention to the inverse document frequency, or the number of times the same term appears in other documents. TF-IDF is a score that is consequently assigned to each word in every document.
- **Count Vectorizer** - As machines do not understand characters and words, we need to convert them into numbers to be understood. CountVectorizer is a method of converting textual data to numerical data. It converts collection of text documents to a vector of term/token counts. It creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. It also enables preprocessing of text data prior to generating vector representation.
- **Stemming** - This is a technique which is used to extract the base form of words by removing affixes from them. It is used to process unstructured text and simplifies the task of analyzing the processed task. It used the stem of the word. We have tried using stemming in our project but did not continue using it as it was affecting the performance of the model. This was done using the Python’s NLTK library.
- **Lemmatization** - This technique uses the meaning behind the words to extract the lemma. To do this, it has a detailed dictionary which the algorithm can look through to link the form back to its lemma. t is important to consider each word's morphological analysis. For lemmatization as well, we used the Python’s NLTK library.

C. *Improving models by data balancing*

The target dataset contains data that has an uneven number of distributions. As a result, there is a significant bias in the class distribution, which could lead some algorithms to completely disregard the minority class. In our dataset, the imbalance is brought on by the quantity of reviews, where the class "5" represents the majority.

To address this problem, we can randomly resample the training dataset. The two fundamental techniques for classifying imbalances are oversampling and undersampling. While random oversampling replicates samples from the minority class, random undersampling includes eliminating examples from the majority class from the training dataset.

IV. Implementation and Example Analysis

The functionalities implemented in our project are the following as given below and along with these 4 functionalities we also have an UI for the project which brings together everything and this will be explained below:

A. *Rating Prediction Model*

In this model, the main functionality is to predict the rating of the given user review.
This has been achieved by the usage of some linear machine learning models like linear SVC, Multinomial Naive Bayes and used ensemble models like Decision Trees, XGBoost.
We have retrieved the dataset from the reviews.json file and more filtered it down to reviews only from restaurants to reduce the variance down to only just the restaurant reviews. This could help us in understanding and gaining the finer details and nuances in linguistics of one category. We further narrowed down the reviews just from California. Initially, baseline performance was done on the imbalance dataset, and then further has been expanded to multiple sampling techniques and balancing of data like random oversampling and random undersampling. Comparison of all the three models have been done and as assumed, the f1score of the baseline model is quite biased towards the imbalanced class. Our key takeaway was the performance improvement that has been achieved by the random Oversampling, not only was there an almost 30% performance increase in terms of f1scores but also there hasn't been a lot of variances among the different classes.

B. *Tagging Prediction Model*

Tagging prediction or Review categorization, this model’s main goal is to help the app developers or forum moderators to classify/tag the reviews based on the review the user has been provided. A tag is a key-value pair that you define. Data has been collected m from the reviews.json file and data has been formatted to limit the dataset to only 5 categories namely, active-life, Restaurants, beauty & spa, Automotive, Shopping. These are some of the top categories with huge amounts of data. Once the dataset has been created, manual random sampling has been done to create a balanced dataset. Motivation to limit the categories to 5, initially was to gather reasonable amounts of the data and take categories that are different from each other, which would help the model in determining them in a better and accurate way. The models that we used include LinearSVC.

C. *Sentiment Analysis*

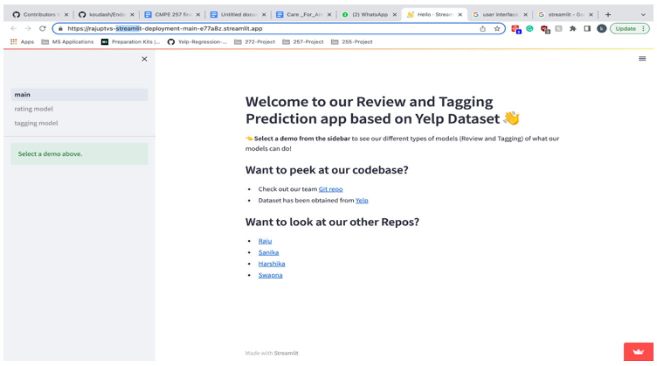
We initially started with sentiment analysis to get an overview of review classification into two main categories which are positive and negative sentiments. The reviews which are loaded from the JSON file are loaded into a data frame. For sentiment analysis, we have only used the two attributes of the review_text and review_start. The new attribute label is added to the data frame according to the star rating. The reviews with a star rating above 3 are labeled as 1 and the reviews below three are labeled as 0. Then the text reviews are converted to a vector on the frequency of the occurrence of each word using count vectorization while tokenizing the words both unigrams and bigrams to perverse the sentiment of the review. The model is trained by using logistic regression with liblinear analyzer The model can predict the sentiment with an accuracy of 93.7%. We also extended the sentiment analysis model with three sentiments positive sentiment, neutral sentiment, and negative sentiment. The reviews with a star rating above 3 are labeled as 2 and the reviews below three are labeled as 1 and reviews with a rating equal to 3 are labeled as 0. This model can predict the sentiment with an accuracy of 86.6%.

D. False Review Detection

In this part of project, we have built some classification models to detect whether the review given is real or fake. We have used hr_modified.csv dataset in this model, the link is given in the appendix section [2]. We have also compared various classification models for the same task such as LinearSVC, decision trees and a lot of other models. We also tried to tune the hyperparameters to get good results with our model on the test data. The future scope of this function will be to include the RNN model to generate the fake reviews and we will train our false review detection model on ample amount of training data, to get more accuracy.

E. User Interface(UI)

We have implemented a user interface where the user can access the rating and tagging prediction model so that it can be developed through this user-friendly webpage rather than running the notebooks. This is done using the Python's streamlit library which is an open-source library that assists in making easy and beautiful custom web pages to build and deploy powerful models and data apps.



Welcome to our Review and Tagging Prediction app based on Yelp Dataset 👋

👈 **Select a demo from the sidebar** to see our different types of models (Review and Tagging) of what our models can do!

Want to peek at our codebase?

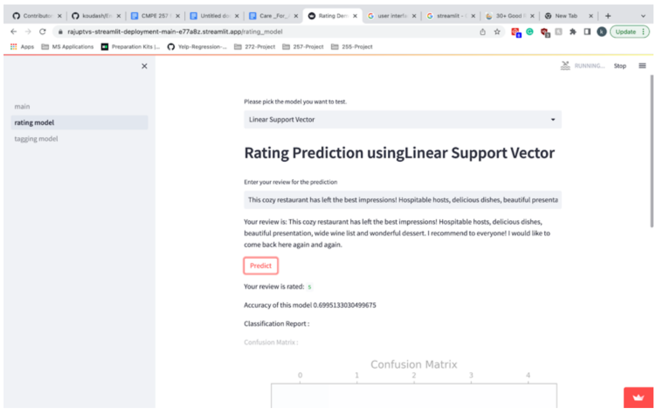
- Check out our team Git repo
- Dataset has been obtained from Yelp

Want to look at our other Repos?

- Raju
- Sanika
- Harshika
- Swapna

Fig. 1. Homepage of UI using Streamlit – python library

This is the home page for our application, where the user can select a rating model or tagging model. If the user selects the rating model, the user will be redirected to a page where the user can enter a text review and based on the rating the model can predict the star rating of the given text review.



Please pick the model you want to test.

Linear Support Vector

Rating Prediction usingLinear Support Vector

Enter your review for the prediction

This cozy restaurant has left the best impressions! Hospitable hosts, delicious dishes, beautiful presenta

Your review is: This cozy restaurant has left the best impressions! Hospitable hosts, delicious dishes, beautiful presentation, wide wine list and wonderful dessert. I recommend to everyone! I would like to come back here again and again.

Your review is rated: 5

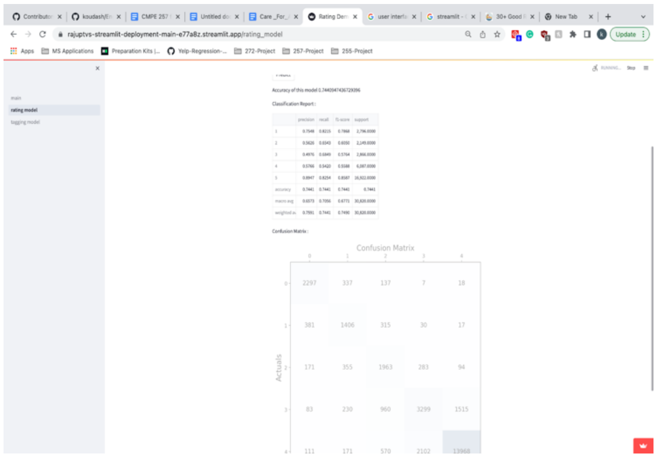
Accuracy of this model 0.6995133030499675

Classification Report :

Confusion Matrix :

Fig. 2. Rating prediction page

The user is also given the option to choose from the two models Linear Support Vector and Linear Support Vector with Oversampling. The rating page also provides the information like accuracy and confusion matrix of that particular model selected.



Accuracy of this model 0.7440947436729396

Classification Report :

	precision	recall	f1-score	support
1	0.7548	0.8215	0.7868	2,796.0000
2	0.5626	0.6543	0.6050	2,149.0000
3	0.4976	0.6849	0.5764	2,866.0000
4	0.5766	0.5420	0.5588	6,087.0000
5	0.8947	0.8254	0.8587	16,922.0000
accuracy	0.7441	0.7441	0.7441	0.7441
macro avg	0.6573	0.7056	0.6771	30,820.0000
weighted av	0.7591	0.7441	0.7490	30,820.0000

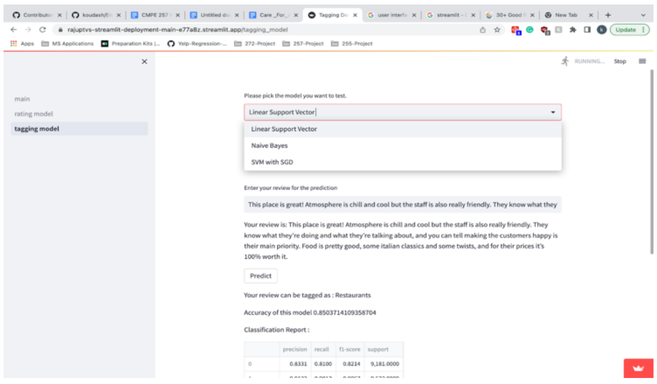
Confusion Matrix :

Confusion Matrix

Actuals	0	1	2	3	4
0	2297	337	137	7	18
1	381	1406	315	30	17
2	171	355	1963	283	94
3	83	230	960	3299	1515
4	111	171	570	2102	13968

Fig. 3. Rating prediction page showing classification report and confusion matrix

If the user selects the tagging, the user will be redirected to a page where the user can enter a text review and based on the rating the model can be classified into the category it belongs to.



Please pick the model you want to test.

Linear Support Vector

Enter your review for the prediction

This place is great! Atmosphere is chill and cool but the staff is also really friendly. They know what they

Your review is: This place is great! Atmosphere is chill and cool but the staff is also really friendly. They know what they're doing and what they're talking about, and you can tell making the customers happy is their main priority. Food is pretty good, some italian classics and some twists, and for their prices it's 100% worth it.

Your review can be tagged as : Restaurants

Accuracy of this model 0.8503714109358704

Classification Report :

	precision	recall	f1-score	support
0	0.8331	0.8100	0.8214	9,181.0000

Fig. 4. Tagging Model Page in UI

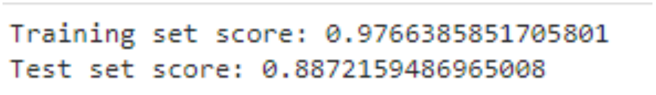
V. Results and Comparisons

A. *Comparing various models for Rating Prediction*

We have trained various models for the task of rating prediction based on the review given by the user. The models were tested on full and half dataset. The full datasets were computationally expensive accounting for testing of one model. The following results show the comparison between various classification models for the same task.

Using Full Dataset:

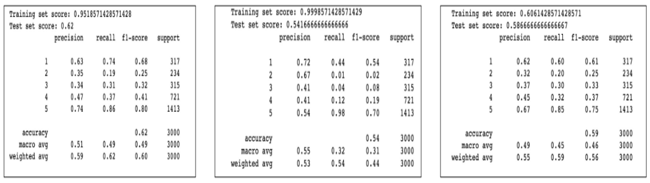
a. *Linear SVC*



```
Training set score: 0.9766385851705801
Test set score: 0.8872159486965008
```

Fig. 5. Results of Linear SVC on Full Dataset

Using Half Dataset:



```
Training set score: 0.9518571428571428
Test set score: 0.62
              precision    recall  f1-score   support

           1       0.63      0.74      0.68       317
           2       0.35      0.19      0.25       234
           3       0.34      0.31      0.32       315
           4       0.47      0.37      0.41       721
           5       0.74      0.86      0.80      1413

    accuracy                           0.62      3000
   macro avg       0.51      0.49      0.49      3000
weighted avg       0.59      0.62      0.60      3000
```

```
Training set score: 0.9998571428571429
Test set score: 0.5416666666666666
              precision    recall  f1-score   support

           1       0.72      0.44      0.54       317
           2       0.67      0.01      0.02       234
           3       0.41      0.04      0.08       315
           4       0.41      0.12      0.19       721
           5       0.54      0.98      0.70      1413

    accuracy                           0.54      3000
   macro avg       0.55      0.32      0.31      3000
weighted avg       0.53      0.54      0.44      3000
```

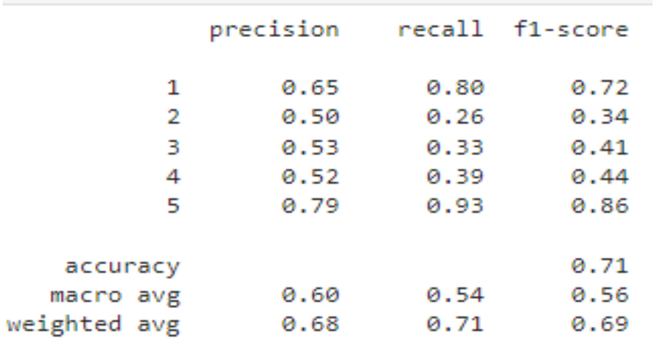
```
Training set score: 0.6061428571428571
Test set score: 0.5866666666666667
              precision    recall  f1-score   support

           1       0.62      0.60      0.61       317
           2       0.32      0.20      0.25       234
           3       0.37      0.30      0.33       315
           4       0.45      0.32      0.37       721
           5       0.67      0.85      0.75      1413

    accuracy                           0.59      3000
   macro avg       0.49      0.45      0.46      3000
weighted avg       0.55      0.59      0.56      3000
```

Fig. 6. Results of LinearSVC, Random Forest and Adaboost respectively on Half Dataset

From the image above, the accuracies of LinearSVC, Random Forest and Adaboost are 62%, 54% and 59% respectively.

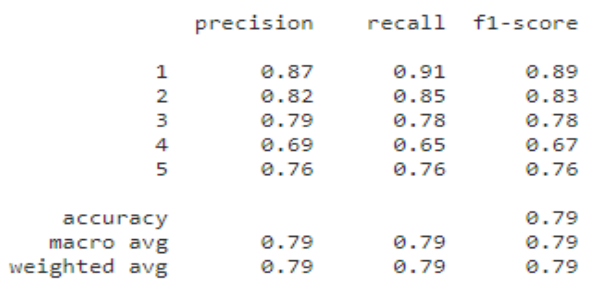
B. *Improving Rating Prediction Models*

Dealing with unbalanced data was one of the key issues that our models have had to deal with. Our models were not functioning well because they were being trained on imbalanced data.
Below results show comparison between F1 score of various classes, before balancing and after balancing using oversampling and undersampling of data.



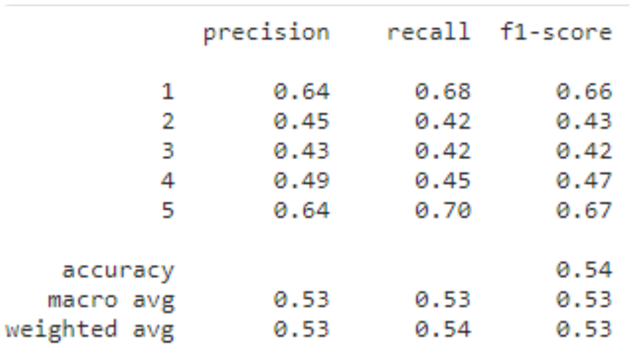
```
              precision    recall  f1-score

           1       0.65      0.80      0.72
           2       0.50      0.26      0.34
           3       0.53      0.33      0.41
           4       0.52      0.39      0.44
           5       0.79      0.93      0.86

    accuracy                           0.71
   macro avg       0.60      0.54      0.56
weighted avg       0.68      0.71      0.69
```

Fig. 7. F-1 score and precision before balancing of data



```
              precision    recall  f1-score

           1       0.87      0.91      0.89
           2       0.82      0.85      0.83
           3       0.79      0.78      0.78
           4       0.69      0.65      0.67
           5       0.76      0.76      0.76

    accuracy                           0.79
   macro avg       0.79      0.79      0.79
weighted avg       0.79      0.79      0.79
```

Fig. 8. F-1 score and precision after data balancing using oversampling

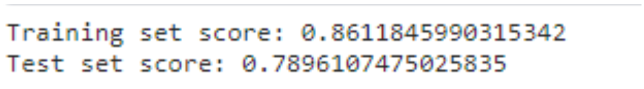


```
              precision    recall  f1-score

           1       0.64      0.68      0.66
           2       0.45      0.42      0.43
           3       0.43      0.42      0.42
           4       0.49      0.45      0.47
           5       0.64      0.70      0.67

    accuracy                           0.54
   macro avg       0.53      0.53      0.53
weighted avg       0.53      0.54      0.53
```

Fig. 9. F-1 score and precision after data balancing using undersampling

When we balanced the imbalance data, we used various methods for performing sampling of data. Below is the result comparison. Test scores for oversampling and undersampling for the LinearSVC model is shown below:

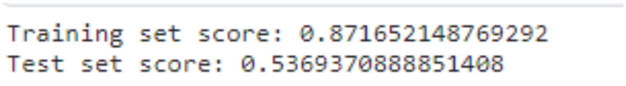
For oversampling using Random oversampling:



```
Training set score: 0.8611845990315342
Test set score: 0.7896107475025835
```

Fig. 10. Results of LinearSVC using oversampling of data

For undersampling using Random undersampling:



```
Training set score: 0.871652148769292
Test set score: 0.5369370888851408
```

Fig. 11. Results of LinearSVC using undersampling of data

From the above scores, we can conclude that for linear models oversampling is good approach than undersampling.

C. *Comparing various models for Rating Prediction using Oversampling and Undersampling respectively*

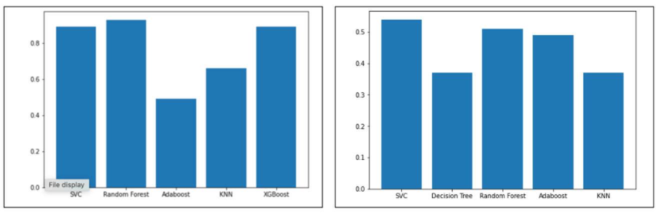
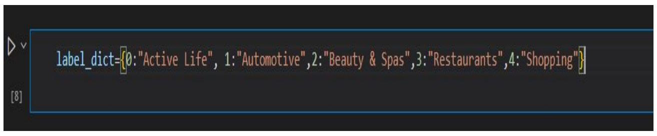


Fig. 12. Oversampling and undersampling comparison in different classifier

In the above results, the x-axis shows various models, and the y-axis has the corresponding F-1 scores.

D. Tagging Model Results

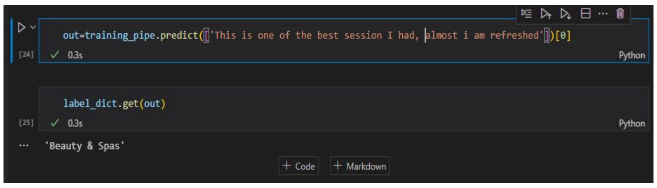
When checking results for tagging model, we have divided our categories into 5 main categories which are very different from one another as shown below in the image. These categories are namely Active Life, Automotive, Beauty and Spas, Restaurants and Shopping.



```
[8]  label_dict={0:"Active Life", 1:"Automotive",2:"Beauty & Spas",3:"Restaurants",4:"Shopping"}
```

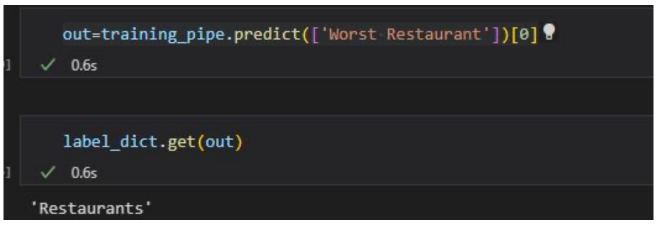
Fig. 13. Different categories used in Tagging model

The following figures show how our model can predict the categories, based on the review given in the text input.



```
out=training_pipe.predict(['This is one of the best session I had, almost i am refreshed'])[0]
[24]  ✓ 0.3s                                                         Python

label_dict.get(out)
[25]  ✓ 0.3s                                                         Python
...  'Beauty & Spas'
```

Fig.14. Output of Tagging Model using Beauty and Spa Category



```
out=training_pipe.predict(['Worst Restaurant'])[0]
✓ 0.6s

label_dict.get(out)
✓ 0.6s

'Restaurants'
```

Fig. 15. Output of Tagging model using Restaurant category

E. False Review Detection

When we tried to understand the Yelp Dataset, we realized that many of the businesses have fake reviews as well. We have implemented various classification algorithms to classify whether the review is fake or real.

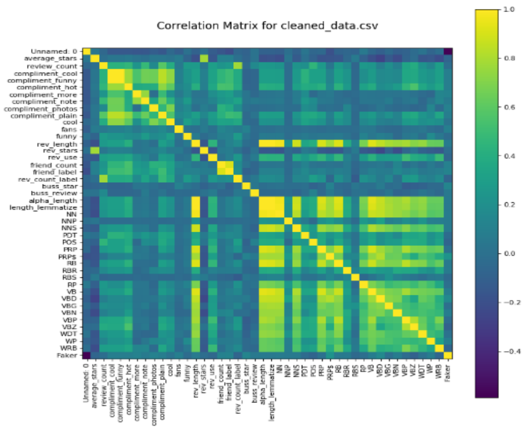
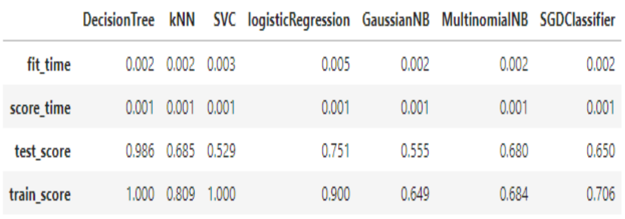


Fig. 16. Correlation matrix of the Yelp dataset

Below, the table shows the comparison of various algorithms for the task of classification on the same dataset. The table shows fit_time comparison which is based on the time taken by the model for training on the training data. Train and test score shows, how our models have performed on training data and test data.



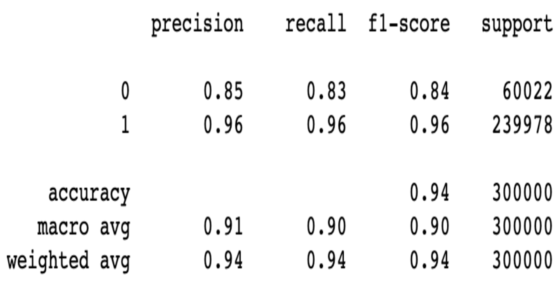
	DecisionTree	kNN	SVC	logisticRegression	GaussianNB	MultinomialNB	SGDClassifier
fit_time	0.002	0.002	0.003	0.005	0.002	0.002	0.002
score_time	0.001	0.001	0.001	0.001	0.001	0.001	0.001
test_score	0.986	0.685	0.529	0.751	0.555	0.680	0.650
train_score	1.000	0.809	1.000	0.900	0.649	0.684	0.706

Fig. 17. Table showing various comparisons by different models on training data.

As from the table above it seems that the SVC model (support vector machine) is not giving very good accuracy. As the goal of this project is to understand linear models, we also tried to tune the hyperparameters such as gamma C in the model.

F. Sentiment Analysis

In this analysis, we trained a model using regression sentiment prediction based on the review given by the user. The following shows the result metrics and the prediction for the sample test data.



```
              precision    recall  f1-score   support

           0       0.85      0.83      0.84     60022
           1       0.96      0.96      0.96    239978

    accuracy                           0.94    300000
   macro avg       0.91      0.90      0.90    300000
weighted avg       0.94      0.94      0.94    300000
```

Fig. 18. Result metrics and prediction for the sample test data

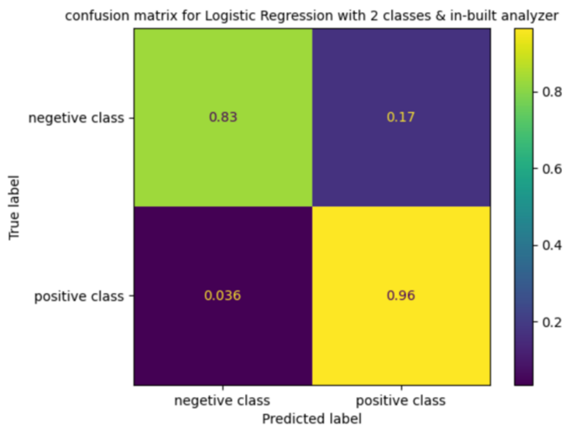


Fig. 19. Confusion matrix for logistic regression used for sentiment analysis

VI. Conclusion

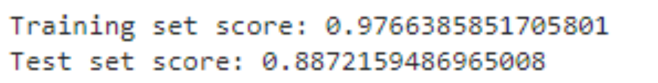
After implementing various techniques, we can conclude the following things about Yelp dataset and machine learning. The following are our learning from the 2-step learning approach:

A. Do you have enough data to learn?

We have trained our models on California restaurants from Yelp dataset, and we got good prediction results with the same. Further, we divided our dataset into half and trained our model on this dataset.
In second case, we are getting good accuracy in prediction and hence we can say that if we are able to predict with half dataset correctly, then we have enough data to learn and to give good results.

B. Are you implementing two-step learning process?

Our project has models in which we are implementing the two-step learning process.
From the below scores in the figure for the Linear SVC model, we can say that $E_{in} \rightarrow 0$ as our training accuracy is 97%. Also, as our training and test scores are close enough to each other, we can conclude that $E_{out} \rightarrow E_{in}$. Hence, we can say we have learned in a two-step process successfully and learning is feasible for us because we can utilize that in getting good predictions.



```
Training set score: 0.9766385851705801
Test set score: 0.8872159486965008
```

Fig. 20. Result comparison between train and test dataset for LinearSVC

C. How have you improved your models?

We have trained various models and compared them with each other as to which gives more accuracy. We also experimented with features and correlation matrix to improve the efficiency. We then perform Data Preprocessing again as we discover the different nuances that our data has like the imbalance of data and hence, balance the data using random oversampling and random undersampling data to improve the model.

References

[1] Yelp Dataset- (link: https://www.yelp.com/dataset)

[2] Y. Chen and F. Xia, "Restaurants' Rating Prediction Using Yelp Dataset," 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA), 2020,pp.113-117,doi:10.1109/AEECA49918.2020.9213704.

[3] T. Doan and J. Kalita, "Sentiment Analysis of Restaurant Reviews on Yelp with Incremental Learning," 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 697-700, doi: 10.1109/ICMLA.2016.0123.K. Elissa, "Title of paper if known," unpublished.

[4] H. S. and R. Ramathmika, "Sentiment Analysis of Yelp Reviews by Machine Learning," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 700-704, doi: 10.1109/ICCS45141.2019.9065812Y.

[5] Nabiha Asghar, "Yelp Dataset Challenge: Review Rating Prediction," 2016, doi: 10.48550/arXiv.1605.05362.

[6] Mohesen Alam, Benjamin Cevallos, Oscar Flores, Randall Lunetto, Kotaro Yayoshi Jongwook Woo, "Yelp Dataset Analysis using Scalable Big Data," 202, doi: 10.48550/arXiv.2104.08396

[7] A. Sihombing and A. C. M. Fong, "Fake Review Detection on Yelp Dataset Using Classification Techniques in Machine Learning," 2019 International Conference on contemporary Computing and Informatics (IC3I), 2019, pp. 64-68, doi: 10.1109/IC3I46837.2019.9055644.

Appendix

[1] Project Demo Link: https://rajuptvs-streamlit-deployment-main-e77a8z.streamlit.app/

[2] Clean Datasets: https://drive.google.com/drive/folders/1ZZQRmJ9G19Y1m9oQK2_bPa39LfepY1hf?usp=sharing

[3] Source Code Link: https://github.com/harshika14/CMPE-257_ProjectTeam12