YELP Reviews Classification

Shyam Kumar Kannuru
Sai Sumanth Muvva
Sri Venkata Sai Anoop Bulusu

***Abstract*—In this project, we aimed to classify Yelp reviews into positive and negative sentiment using machine learning techniques. We collected a dataset of Yelp reviews and ratings and used various machine learning algorithms to train models that could predict the sentiment of a given review. We compared the performance of different algorithms and found that some were more effective than others at accurately classifying the reviews. We compare the results of various algorithms and demonstrate how machine learning can be used to address this problem. In the end, we were able to achieve high levels of accuracy by performing various pre-processing, feature engineering and regularization techniques on the data before training the models, which shows the potential of machine learning for understanding and analyzing customer feedback from online review platforms like Yelp. In this paper, we show that the XGBOOST Classifier achieved the best performance, with an AUROC of 0.969.**

I. Keywords

NumPy, Pandas, Count Vectorizer, TF-IDF Vectorizer, Feature scaling (l2 regularization), Machine learning models (Logistic regression, Support Vector Machine, Multinomial Naïve-Bayes Classifier, XGBoost Classifier, Random Forest Classifier, MLP Classifier,), GridSearchCV, accuracy, F-1 score, AUROC

II. Introduction

Yelp reviews can be a valuable resource for consumers looking to make informed decisions about where to eat, shop, and spend their money. They provide a platform for customers to share their experiences and opinions about a wide range of businesses, which can help others decide whether or not to patronize a particular establishment.

With the proliferation of online review platforms, businesses are increasingly relying on customer feedback to evaluate their performance and make improvements. But it might take a lot of time and resources to manually read and comprehend a lot of customer evaluations. Businesses may simply and rapidly acquire insights into consumer sentiment and take steps to enhance their goods and services by utilizing machine learning algorithms to automatically identify reviews as positive or negative. I Therefore, in this project we aim to find a solution to tackle this problem easily investigating several machine learning techniques and assess how well they work using a Yelp reviews dataset.

III. Data Description

The data for this project is made up of reviews from Yelp. Each review includes the text of the customer's feedback. The data-set is diverse and contains a large amount of information that can be used to train and evaluate machine learning models for the classification of Yelp reviews.

A. About the data:

- The data-set is constructed by assigning stars 1 and 2 to the negative class, and stars 3 and 4 to the positive class.
- A total of 280,000 training samples and 19,000 testing samples are randomly selected for each polarity, resulting in 560,000 training samples and 38,000 testing samples in total.
- The negative polarity is assigned class 1, and the positive polarity is assigned class 2.
- The features in the data-set are:
  I. Feature 1: Class index
  II. Feature 2: Review text

The dataset can be found on Kaggle[1]

IV. Data Prepossessing and Feature Engineering

A. Data Cleaning

- The initial data cleaning on the data-set included removing all the null values and the empty data fields.
- Then, we performed text cleaning which includes removing all the unwanted information from the reviews such as: removing the URL's, removing the punctuation, removing non-ASCII characters, removing the special characters, removing numbers, removing extra white spaces wherever applicable, converting the text into lowercase.

B. Vectorization

To classify the Yelp reviews as positive or negative, we implemented various machine learning and Deep learning models. So, for the models to train and classify we need to convert the text (reviews) into numerical representation.

Therefore, After cleaning the data, we have also created count vectorizer matrix as well as the TF-IDF matrix on the data as one of the data pre-processing techniques. We have used both count vectorizer matrix and TF-IDF matrix to train various machine learning models.

- Count Vectorizer:
  Count Vectorizer[2] is a method for tokenizing a collection of text documents and building a vocabulary of tokens(also called n-grams). This vocabulary can be used to encode new documents in the form of a vector, with each dimension of the vector representing a different token(number). The length of the vector is the number of tokens in the vocabulary, and the value of each dimension is the number of tokens that the token appears in the encoded document. In this method we set the minimum frequency to 5.
  For example, if the vocabulary consists of tokens 'Boy','Girl', 'Toy', then the sentence "I am a Boy" could be encoded as the vector [0,0,0,1]. This is known as a count vector.

- Tf-idf Vectorizer:
  Tf-IDF (term frequency-inverse document frequency) Vectorizer [3] is a method for encoding a collection of text documents by transforming each document into a vector of documents. The tf-idf vectorizer combines two statistics: term frequency and inverse document frequency. Term Frequency (word in a document) is the ratio of the number of times the word appears in the document to the total number of words in the document. Whereas Inverse document frequency is calculated by dividing the total number of documents containing the word, and then taking the logarithm of that quotient.

C. *Feature Engineering*

After cleaning the data, we have applied the following feature engineering on the data and added new features into it.

- Number of words in each review: Added a column which gives number of words in each review

- Average length of each word in each review: Added a column which gives average length of each word in each review

- Number of characters in each review: Added a column which gives number of characters in each review

- Number of unique words in each review: Added a column which gives number of unique words in each review

V. Exploratory Data Analysis (EDA)

Once we cleaned and prepared the data, and added new features to it, we explored the dataset through exploratory data analysis (EDA).

Exploratory data analysis (EDA) plays a vital role in any data analysis process, and this is also true for analysing Yelp reviews. EDA involves using various techniques to explore and summarize a dataset, with the goal of better understanding its characteristics and identifying any patterns or trends that may be present.

In the context of Yelp reviews, EDA could involve things like looking at the distribution of ratings for different businesses, identifying common words or phrases used in positive and negative reviews, or examining the relationships between different factors (such as no. of unique words, word count, or review length).

We examined the distribution of positive and negative reviews to see if there is any unequal distribution in the train and test datasets as shown in figure 1. This could help us choose the right performance metric. The negative class is marked as 1 and positive class is marked as 2.

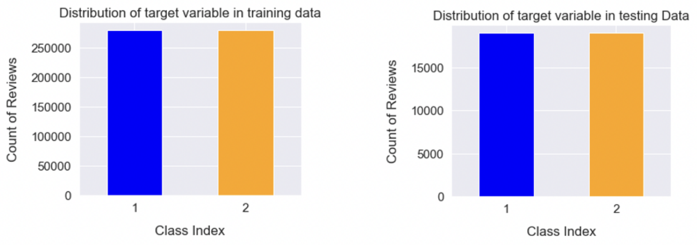


Fig 1: Class variable Distribution

Examined the relationships between different factors (such as frequency of reviews and review length). This could help us understand which reviews creates more impact based on their length. Here in our study, we found that most of the reviews are from 0-250 characters length. So, our main focus is on those reviews. This is shown in figure 2.

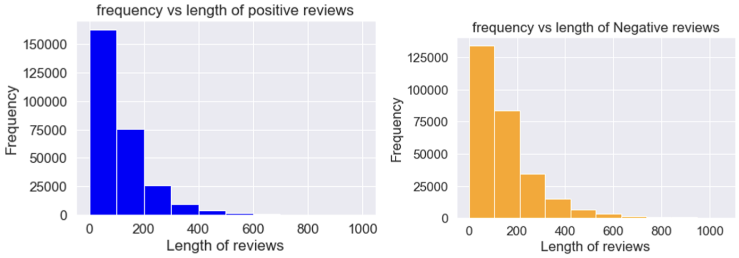


Fig 2: Frequency vs Review length

We identified frequent words or phrases used in positive and negative reviews using uni-gram, bi-gram and tri-gram plots (Fig. 3, 4, and 5)

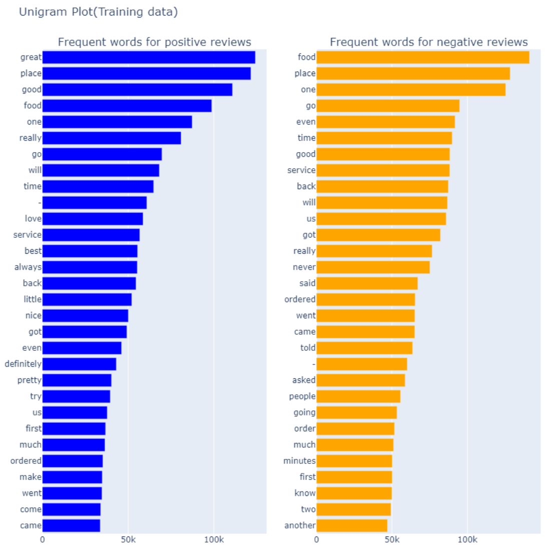


Fig 3: Uni-gram Plot

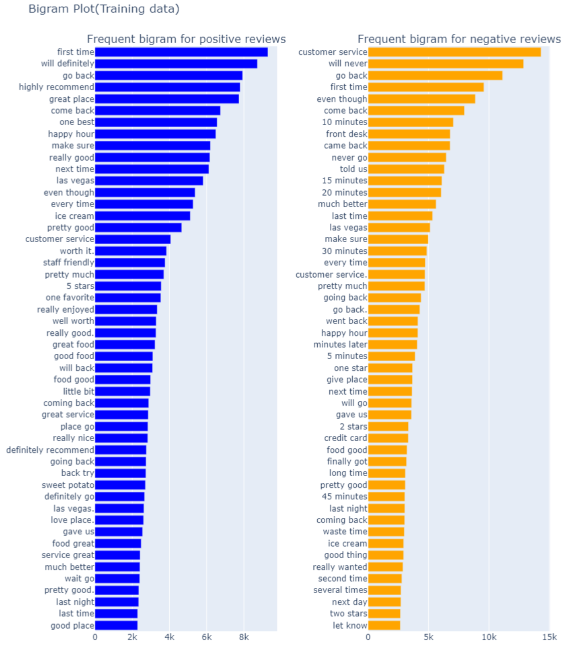


Fig 4: Bi-gram Plot

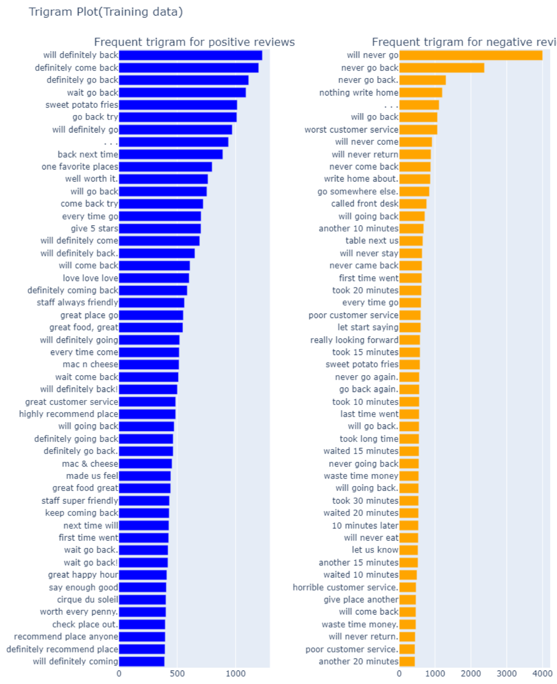


Fig 5: Tri-gram Plot

We also plot word clouds for negative and positive review data to see which words are occurring the most respectively (Fig. 6 and 7)

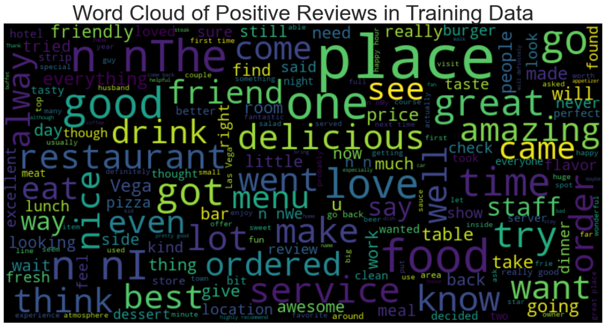


Fig 6: Word cloud for positive reviews

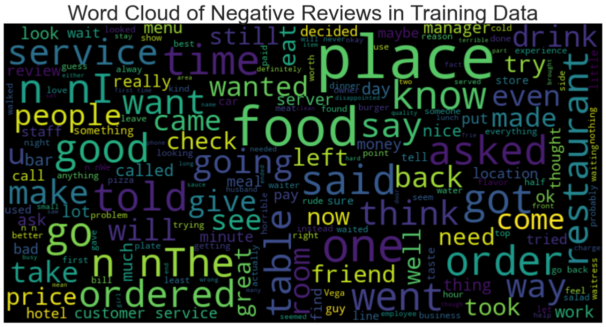


Fig 7: Word cloud for negative reviews

Overall, conducting EDA on Yelp reviews helped us gain valuable insights that can guide further analysis or decision-making.

VI. Experiments

After converting the text data into numerical representation using the above-mentioned methods, these numerical vectors are inputted to the machine learning models. All the models were trained using both the above-mentioned methods(count vectorizer and tf-idf vectorizer).

- Logistic regression:
  - Logistic regression is a commonly used statistical technique for predicting binary outcomes(In our case whether a review is positive or negative). The model learns the weights by minimizing the difference between the predicted probabilities and the observed outcomes in the training data, which is typically done using an optimization algorithm, such as gradient descent, which updates the weights iteratively based on the error in the prediction. These learned weights are then used to make predictions on the test data.
  - To tune the hyperparameters in the logistic regression model the GridSearchCV method was used. The hyperprarameters which were tuned included the regularization parameter which controls the strength of the regularization term in the loss function and the penalty term (type of regularization used) which helps to prevent overfitting.
  - The best hyperparameters('C' = 10 and 'l' = 'l2') were therefore chosen to train the model.
  - The AUROC for the logistic Regression is 0.935 as shown in figure 8

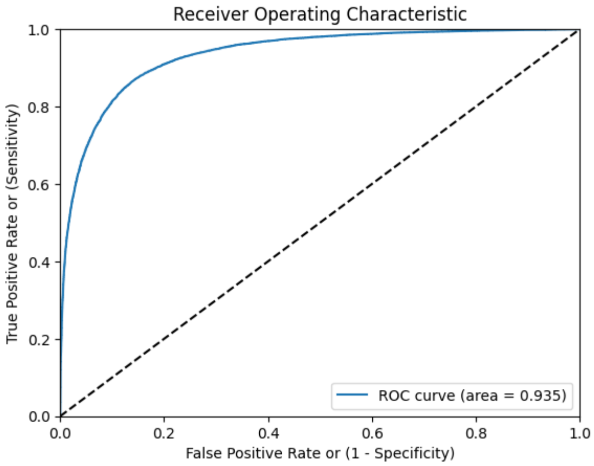


Fig 8: AUROC for the logistic Regression

- Naive Bayes Classifier:
  - We also implemented the Multinomial Naïve Bayes classifier. This model is based on one of the popular theorems i.e., Bayes Theorem with the naïve assumption that each pair of features are independent of the other.
  - We hyper-tuned the alpha parameter using the GridsearhCV method. The optimum value for alpha which we obtained was 0.001. This alpha term helps to ensure the probabilities are well-behaved when the words from the train data do not appear in the test data. The multinomial naïve bayes with tf-idf method was not performing well.
  - The AUROC for the Naive Bayes Classifier is 0.870 as shown in figure 9

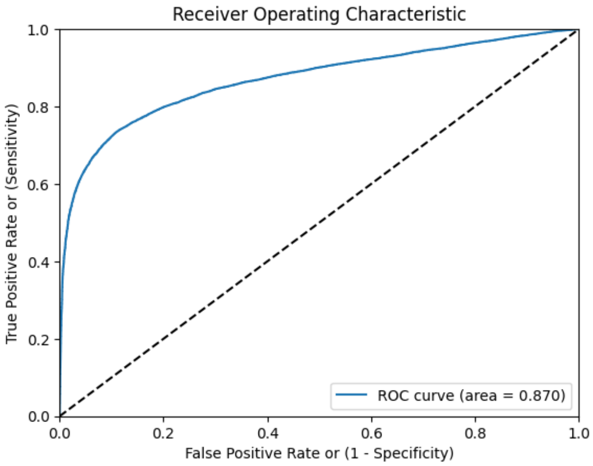


Fig 9: AUROC for the Naive Bayes Classifier

- Support Vector Machine:
  - The linearSVC is one of the popular models used in text classification tasks. The LinearSVC which we built was trained using the hinge loss function and L2 regularization. Using the GridSearchCV we hypertuned the regularization parameter and the tolerance for the stopping criteria. The model is then trained using these parameters for both the Count vectorizer and Tf-idf vectorizer methods. The model with Count vectorizer was performing better.
  - The AUROC for the Support Vector Machine is 0.738 as shown in figure 10

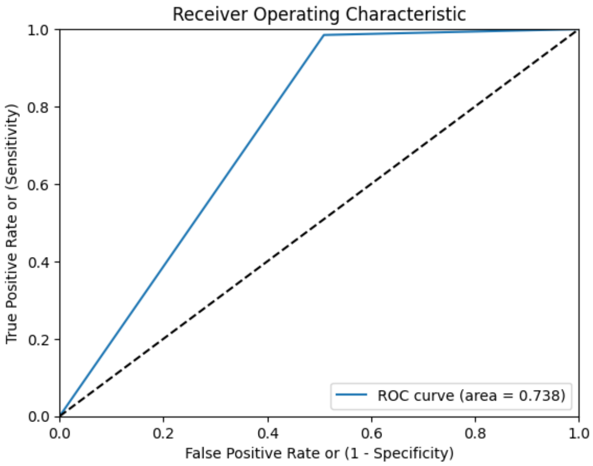


Fig 10: AUROC for the Support Vector Machine

- Ensemble approach:
  - 1. Random Forest:
    A random forest classifier is a machine learning algorithm that uses a collection of decision trees to predict the class or label of a given input. To train a random forest classifier, we first need to create a collection of decision trees by randomly sampling the features in the training dataset and training each tree on the subset of features. The predictions made by the individual trees are then combined to make a final prediction using a technique such as majority voting or weighted averaging. The random forest model we have trained has train accuracy of 99.95 and the test accuracy of 85.92 we can clearly see that the model is overfitting and cannot be used to classify the reviews correctly.
  - The AUROC for the Random Forest is 0.938 as shown in figure 12
  - 2. XGBoost classifier:
    We tried using Decision tree and logistic regression as the base learners for the gradient boosting algorithm. The model with decision tree as the base learner was not performing well but the model performs better with logistic regression as the base learner. The model was performing better if we either use a Count vectorizer or Tf-idf vectorizer.
  - The AUROC for the XGBoost classifier is 0.969 as shown in figure 11

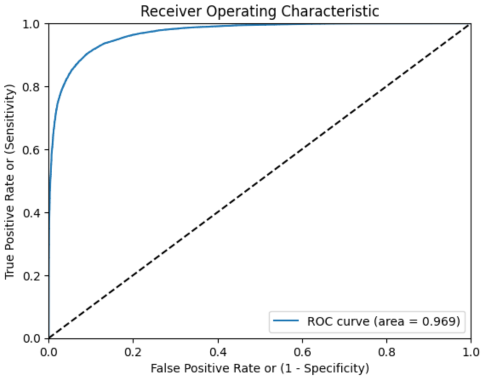


Fig 11: AUROC for the XGBOOST Classifier

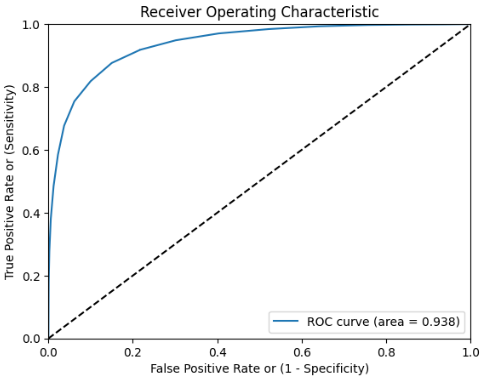


Fig 12: AUROC for the Random forest

- Artificial Neural Network(ANN) Model:
  - We have built an Artificial neural network model in order to classify the reviews into positive and negative.
  - Initially we tried using an ANN model with architecture of 1 input layer, 1 hidden layer, 1 output layer.
  - we have used Adam optimizer and cross entropy loss as the loss function.
  - The ANN model is overfitting on the training data with the current hyper parameters and hence it is not providing good results in classifying the reviews

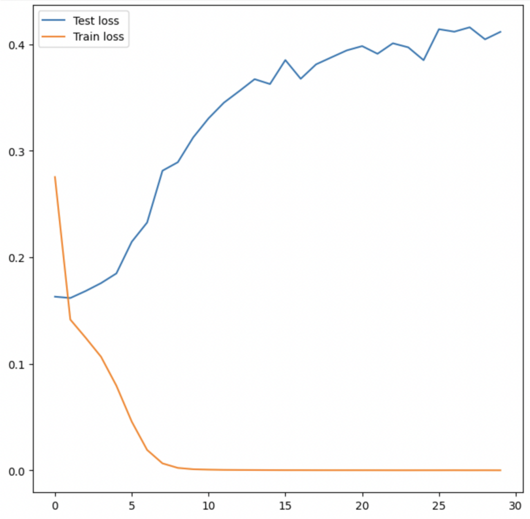
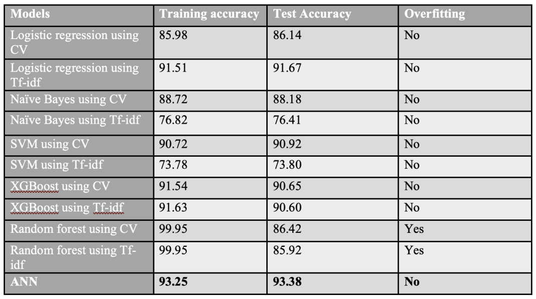


Fig 13: Train and test loss of ANN Model

VII. Results

Since our dataset is balanced, we can use accuracy as an evaluation metric to compare the models. So, for each of the above trained models we calculated the train and test accuracy which helped us to determine if a model is prone to overfitting or not. The figure 14 shows us the accuracies achieved by different models. The models trained using ensemble approaches yielded good accuracies. The XGBoost classifier performs well on the test data too, but the random forest classifier performs poorly on the test data causing the model to overfit this may be because since the random forest are based on decision trees, they can easily learn to memorize the training data due to the complex patterns in the training data. The ANN model is also performing poorly as it is overfitting on the training data.



Models	Training accuracy	Test Accuracy	Overfitting
Logistic regression using CV	85.98	86.14	No
Logistic regression using Tf-idf	91.51	91.67	No
Naïve Bayes using CV	88.72	88.18	No
Naïve Bayes using Tf-idf	76.82	76.41	No
SVM using CV	90.72	90.92	No
SVM using Tf-idf	73.78	73.80	No
XGBoost using CV	91.54	90.65	No
XGBoost using Tf-idf	91.63	90.60	No
Random forest using CV	99.95	86.42	Yes
Random forest using Tf-idf	99.95	85.92	Yes
ANN	**93.25**	**93.38**	**No**

Fig 14: Results

VIII. Conclusion and Future work

After performing the experiments on various above trained models the best accuracies as well as the best AUROC score can be seen for the logistic regression and XGBOOST model. The ANN is overfitting and performing poorly on the data. Future implementations that can be done in order to prevent the ANN model from over fitting are:

1. more and better hyper parameter tuning can be performed

2. use layer wise regularization such as L1 or L2 regularization

3. use dropouts

References

1) https://www.yelp.com/dataset
2) https://scikit-learn.org/stable/modules/generated/sklearn.feature$_{e}xtraction.text.CountVectorizer.html$
3) https://scikit-learn.org/stable/modules/generated/sklearn.feature$_{e}xtraction.text.TfidfVectorizer.html$
4) https://towardsdatascience.com/building-a-sentiment-analysis-model-using-yelp-reviews-and-ml-ensemble-methods-80e45db6d0c7
5) Jader Abreu, Luis Fred, David Macedo, and Cleber Zanchettin, 2019. “Hierarchical Attentional Hybrid Neural Networks for Document Classification”
6) Reinstein, David and Christopher Snyder, 2005. “The Influence of Expert Reviews on Consumer Demand for Experience Goods: A Case Study of Movie Critics,” The Journal of Industrial Economics, Vol. 53, No. 1, 27-51.
7) J. Stoppleman, “Why Yelp Has a Review Filter”, http:// officialblog.yelp.com/2009/10/why-yelp-has-a-review-filter.html
8) Peter Hajas, Louis Gutierrez and Mukkai S. Krishnamoorthy (Department of Computer Science, RPI, Troy, NY), 2014. “Analysis of Yelp Reviews” - Research Gate publication (263736290)
9) https://www.ics.uci.edu/ vpsaini/