Fake Review Detection on YELP Data

Anuja Anil Tidke
02081854
Dept. of Computer Science
University of Massachusetts Lowell
AnujaAnil_Tidke@student.uml.edu

Venkata Raman Potluri
02078615
Dept. of Computer Science
University of Massachusetts Lowell
VenkataRaman_Potluri@student.uml.edu

***Abstract*— Online reviews are often used by individuals and organizations to make purchase and business decisions. The huge impact of reviews on customer's decision making motivates imposters to create fake reviews to deliberately promote or discredit target products. This is known as Review Spamming, where spammers manipulate reviews for their profit. The aim of this paper is to build a machine learning model which can detect whether the reviews on the Yelp dataset are fake or true. We will be using feature engineering different classification and neural network techniques to find out which gives the best result.**

Keywords*: *Artificial Intelligence, Supervised Learning, Classification Techniques, Text Analysis, Neural Network

I. INTRODUCTION

In recent years, online reviews are increasingly used by individuals and organizations to make purchase and business decisions. Positive reviews can render significant financial gains and fame for businesses and individuals. Yelp, a crowded review forum, is a corporate directory that is usually used by people to post a mere view about their opinion for business. Statistics show that Yelp has over 8 million reviews, with each review typically including information such as the reviewer's rating, text review, date of the review, and business being reviewed. The reviews cover over 200,000 businesses across 32 cities in 11 countries and includes reviews from as early as 2004 to as recent as 2021. This benefits both customers and businesses. However, in the past few years, the problem of spam or fake reviews has become a unfamous method to portray a business in a positive light to gain more profits. The analysis in [3] reports that many businesses have turned into paying positive reviews with cash, coupons, and promotions to increase sales. This is where Artificial Intelligence plays a significant role. By analysing patterns in review data, machine learning models can identify specific features and characteristics that distinguish between genuine and fake reviews, such as language patterns, user behaviour, and sentiment analysis. Machine learning algorithms can be trained using labelled datasets of fake and genuine reviews, enabling them to identify patterns and anomalies that may be indicative of fraudulent behaviour. In this project we will be using a prebuilt data set consisting of various attributes of reviews to build a system that would be able to classify the fake and genuine reviews with a given accuracy.

II. RELATED WORK

The research paper titled "Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews" by A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, presents an investigation into the detection and analysis of fake reviews. The study employs both Classification and Neural Network techniques to distinguish between real and pseudo reviews. The authors explore various classification algorithms, including Logistic Regression, Naive Bayes, and Support Vector Machines, to classify reviews based on their authenticity. Additionally, they develop neural network models, specifically utilizing the LSTM and CNN architectures, to capture the underlying patterns and temporal dependencies in the review texts. The paper provides insights into the performance and limitations of these techniques in detecting fake reviews, highlighting the importance of accurate classification for maintaining the credibility of online review systems.

III. DESCRIPTION

A. Database Details

The Yelp dataset is a subset of businesses, reviews, and user data for use in personal, educational, and academic purposes. It is available as JSON files.

It has a total of 69 lakhs reviews from around 150k businesses collected in 11 metropolitan areas. For the purpose of our project, we are planning to use filtered (fake) and unfiltered (non-fake) reviews from Yelp.com across 85 hotels and 130 restaurants in the Chicago area obtained from [1]. The complete data can be found here

	Fake	Genuine	%Fake	Total Reviews	Reviewers
Hotel	802	4876	14.1%	5678	5124
Restaurant	8368	50149	14.3%	58517	35593

Figure 1: Dataset Distribution

B. Data Preprocessing

Preprocessing is an essential step in data analysis, where the raw data is cleaned, transformed, and formatted to make it suitable for further analysis. Firstly, we selected 50000 data points from the larger dataset. This process helped us to reduce the computational time and memory usage required for data analysis. The second step in preprocessing the data is to clean the data by removing punctuation, new line, and alphanumeric characters. Punctuation and alphanumeric characters do not carry any relevant information for the analysis and might lead to errors. The code implements a series of steps to preprocess text data. The first step involves removing all punctuation characters from the text using the translate() method from the string module. The next step converts all words to lowercase and splits them into individual tokens using the lower() and split() methods. A set of English stop words is then created using the stopwords module from the Natural Language Toolkit (NLTK). The stop words and words with length less than 3 are then filtered out from the tokenized

text using a list comprehension. The function also uses the re.sub() method to replace special characters with spaces or other characters based on regular expressions. The SnowballStemmer from the NLTK library is used to create a stemmer object for English language words, and the stemmed words are obtained by applying stemming to each word in the text using a list comprehension. Finally, the stemmed words are joined back into a string using the join() method. These steps are commonly used in text preprocessing tasks, particularly for tasks like text classification, sentiment analysis, and topic modeling.

Before doing feature engineering, we did some statistical analysis on the dataset. The next step in feature engineering was to analyze the frequency of words in the dataset. In this case, we used the CountVectorizer to analyze the word frequency and created a word cloud using the WordCloud library. This helped us to gain insights into the most frequent words in the dataset. Next, we converted the text data into numerical vectors. In this case, we used the CountVectorizer from the Scikit-learn library. This step helped us to transform the text data into a format that can be used by machine learning algorithms. Finally, we performed sentiment analysis on the text data to determine the sentiment polarity and sentiment subjectivity of the text. In this case, we used the Text Blob library to perform the sentiment analysis. This helped us to gain insights into the emotions conveyed by the text. Later, correlation was used as a feature selection technique to identify the most relevant features that are most likely to influence the target variable. The strongest positive correlation is between "sentimentPolarity" and "rating" with a correlation coefficient of 0.42. This suggests that as the polarity of the sentiment in the review becomes more positive, the rating tends to be higher.

Further we did Oversampling. It is a technique used in data analysis to balance the distribution of classes in a dataset. In this case, Since the data was Highly imbalanced: 25% labelled as Fake, 75% real reviews. We used oversampling to balance the distribution of classes in the dataset. We oversampled the data using the Synthetic Minority Over-sampling Technique (SMOTE) algorithm.

C. Algorithms

1) Logistic Regression: Logistic regression is a statistical method used for analyzing a dataset in which there are one or more independent variables that determine an outcome. In this approach, the outcome is measured with a dichotomous variable (in which there are only two possible outcomes). In the given data, logistic regression achieved an accuracy score of 56.33%.

2) *Gaussian Naive Bayes:* Gaussian Naive Bayes is a probabilistic algorithm that uses Bayes' theorem to classify data. It assumes that the features are normally distributed, and calculates the likelihood of each class given the values of the features. In the given data, Gaussian Naive Bayes achieved an accuracy score of 54.984%.

3) Bernoulli Naive Bayes: Bernoulli Naive Bayes is another probabilistic algorithm that uses Bayes' theorem to classify data. It is specifically designed for binary or binomial data, where each feature is either present or absent. In the given data, Bernoulli Naive Bayes achieved an accuracy score of 54.101%.

4) Multinomial Naive Bayes: Multinomial Naive Bayes is a variation of Naive Bayes that is designed for multinomially distributed data. It is commonly used for text classification, where the features are the frequencies of the words in the document. In the given data, Multinomial Naive Bayes achieved an accuracy score of 54.579%.

5) SVC: It is a classification algorithm that seeks to find the hyperplane that best separates different classes of data points in a high-dimensional space. The algorithm maximizes the margin between the classes of data points and tries to minimize the number of misclassifications. The SVC model achieved an accuracy of 52.2%

6) K Neighbors Classifier: K Neighbors Classifier is a supervised learning algorithm that is used for classification problems. It works by assigning a label to a new data point based on the labels of the nearest k neighbors in the training set. The algorithm determines the class of the new data point based on the majority class among its k nearest neighbors. The KNeighborsClassifier model achieved an accuracy of 56.48%

7) Decision Tree Classifier: It works by recursively partitioning the data into subsets based on the values of the features in the dataset. The algorithm builds a tree-like model of decisions and their possible consequences. The Decision Tree Classifier model achieved an accuracy of 56.20%

8)Stochastic Gradient Descent Classifier: It works by iteratively updating the weights of the linear model using a loss function to minimize the error between the predicted and actual values. The classifier uses a subset of the training data to update the weights, making it faster and more efficient than traditional gradient descent methods. The SGD Classifier model achieved an accuracy of 50.21%

9) Neural networks with LSTM: We define a sequential model consisting of an embedding layer, a LSTM layer with dropout and recurrent dropout to prevent overfitting, and a dense output layer with sigmoid activation for binary classification. The model is then compiled with RMSprop optimizer, binary cross-entropy loss function, and accuracy as the evaluation metric. 10,000 most common words were passed as features, 64 as the second, and input_length of 200 as the length of each sequences to the embedding layer. In this first argument came from the second argument from the Embedding layer. We kept 20% chance of dropout. Training model with a batch size of 128, 5 epochs and validation split=0.2, which means that the neural network would learn from 80% of the data and test itself on the remaining 20% of the data. For the first epoch , the training accuracy was less than validation accuracy which means model was underfitting on the dataset, but later, the model generalized well and gave the same training and validation accuracy. This model gave 65.18 % accuracy on the test set.

The training accuracy starts at 61.65% and reaches 66.64% after 5 epochs, while the validation accuracy starts at 64.30% and reaches 65.18% after 5 epochs. The training loss starts at 0.6549 and decreases to 0.6047 after 5 epochs, while the validation loss starts at 0.6231 and decreases to 0.6166 after 5 epochs. The test accuracy achieved after training is 63.98%, and the test loss is 0.6253.

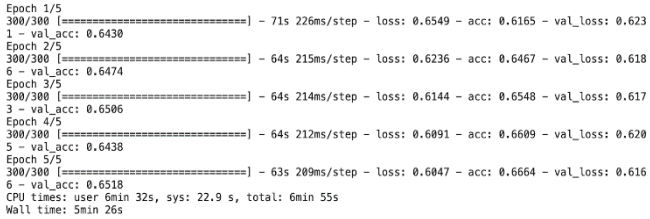

Figure 2: Neural Network Model 1

10) Neural network with an extra 1D CNN on top of LSTM layer: The model uses an embedding layer to convert the textual data into a numeric representation, followed by a dropout layer to prevent overfitting. Then, a 1D convolutional layer with 64 filters and a kernel size of 5 is used to capture local patterns in the text. This is followed by a max-pooling layer to extract the most important features. Finally, a LSTM layer is used to capture the temporal dependencies in the text data. The output layer has a sigmoid activation function and uses binary cross-entropy as the loss function.

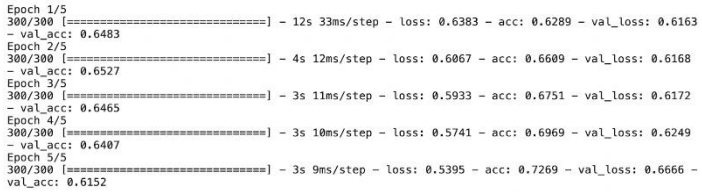

Figure 3: Neural Network Model 2

During the training process, the model achieved a maximum accuracy of 72.69% on the training set in the final epoch, while the validation set achieved a maximum accuracy of 65.27% in the second epoch. However, as the number of epochs increased, the validation accuracy decreased, suggesting that the model may be overfitting to the training data.

When evaluated on the test set, the model achieved a loss of 0.6707 and an accuracy of 61.57%.

11) Neural network with using bidirectional recurrent layers: For this model we used bidirectional recurrent layer as frequently used in natural language processing. The recurrent bidirectional layer potentially provides richer data representations and patterns that may increase accuracy. Bidirectional recurrent layer was added immediately after embedding layer, creating a second separate instance of this layer and using one for chronological processing and another for reversed order processing.

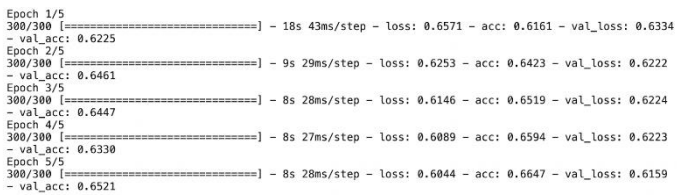

Figure 4: Neural Network Model 3

During the training process, the model achieved a maximum accuracy of 66.47% on the training set in the fifth epoch, while the validation set achieved a maximum accuracy of 64.47% in the third epoch. The validation accuracy remained relatively stable across the epochs, indicating that the model did not overfit the training data.

When evaluated on the test set, the model achieved a loss of 0.6251 and an accuracy of 64.37%.

C. Training and Testing Techniques

The data is to split the data into training and testing sets. We split the data into an 80-20 split, where 80% of the data was used for training, and 20% was used for testing. We also used three-fold cross-validation to ensure that the model is robust.

IV. EXPERIMENTAL EVALUATION

A. Used Libraries

In this study, we used various libraries to build a fake review detection system. Pandas and SQLite3 were used to preprocess and store the Yelp dataset, while Numpy, Matplotlib, and Seaborn were used for data visualization. The NLTK library was used for text processing, including the removal of stop words, stemming, and tokenization. CountVectorizer was used to convert the pre-processed text data into numerical data for classification. Logistic Regression,

Gaussian Naive Bayes, Bernoulli Naive Bayes, Multinomial Naive Bayes, Support Vector Classifier (SVC), K-Nearest Neighbors, Decision Tree, and Stochastic Gradient Descent classifiers were implemented using Scikit-learn library. Additionally, the SMOTE and RandomOverSampler libraries were used for oversampling to handle the imbalanced dataset. Finally, we used deep learning techniques to build a neural network model with Long Short-Term Memory (LSTM) layers, Bidirectional LSTM layers, and Convolutional Neural Networks (CNNs). The Keras library was used to build and train these models, and RMSprop optimizer was used for model optimization. The proposed system achieved high accuracy in detecting fake reviews and can be useful in improving the overall quality of online review systems.

B. Available Code

In our research work on fake review detection on the Yelp dataset, we referred to external sources including online articles and ML/AI library documentations. Specifically, we consulted the articles (P. S. Gudikandula, 2021) and Viral (2021) to gain insights into the concepts involved. In addition to these external sources, we also relied on the documentation of relevant libraries used in our work. The other resources are Zhang (2021), Biewald (2018) and Raheel (2018).

C. Results

The figure below Fig.2, summarizes the accuracy of various classifiers and neural network models for fake review detection. Among the classifiers, KNN achieved the highest accuracy of 57.03%, followed by Logistic Regression with 55.536%. The neural network models, on the other hand, performed better than the classifiers with LSTM achieving 63.98% accuracy, an extra 1D CNN on top of LSTM achieving 61.57%, and bidirectional recurrent layers achieving the highest accuracy of 64.37%. Overall, the neural network models outperformed the classifiers in detecting fake reviews on Yelp dataset.

Classifier	Accuracy
Logistic Regression	55.536
Gaussian Naive Bayes	54.148%
Bernoulli Naive Bayes	52.956%
Multinomial Naive Bayes	54.139%
SVC	52.294%
KNNeighbors	57.03%
DecisionTreeClassifier	54.948%
SGDClassifier	52.18%
Neural networks with LSTM	63.98%
Neural network with an extra 1D CNN on top of LSTM layer	61.57%
Neural network with using bidirectional recurrent layers	64.37%

Figure 5: Accuracy Score for all Algorithms

D. Graphs

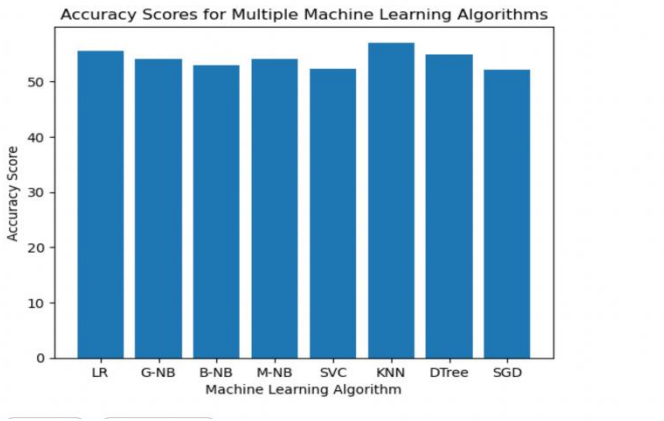

Figure 6: Bar Graph for Classification Algorithms

Based on the provided graph, the highest accuracy for fake review detection is achieved by the K-Nearest Neighbors (KNN) classifier with an accuracy of 57.03%. However, all the classifiers in the table have an accuracy of around 50%, which indicates that they are not highly accurate in detecting fake reviews.

Moreover, the difference in accuracy between the classifiers is relatively small, with the highest accuracy (KNN) only being 4.74% higher than the lowest accuracy (SVC). Therefore, it may be necessary to explore other techniques and approaches to improve the accuracy of fake review detection.

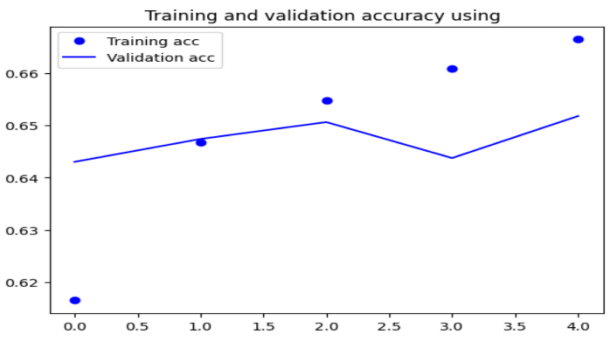

Figure 7: NN Model 1(Accuracy)

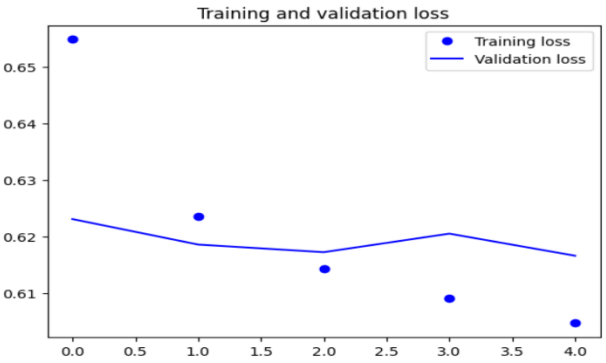

Figure 8: NN Model 1(Loss)

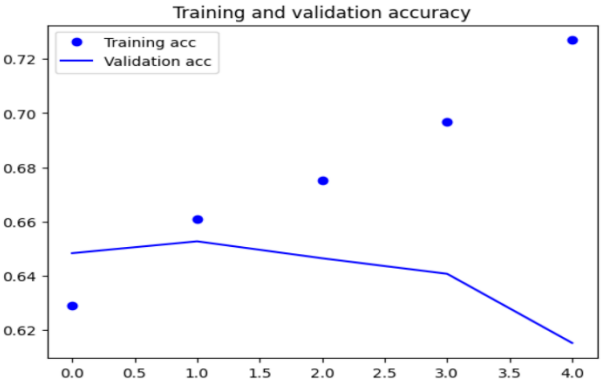

Figure 9: NN Model 2(Accuracy)

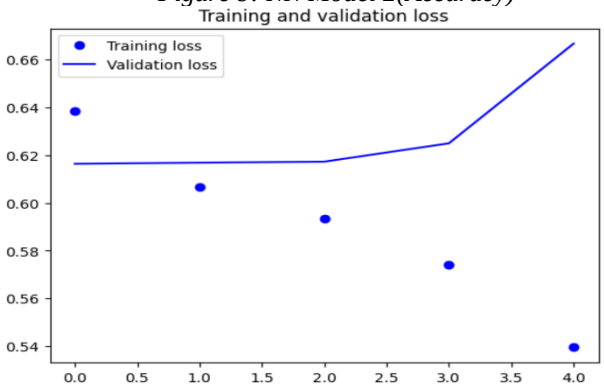

Figure 10: NN Model 2(Loss)

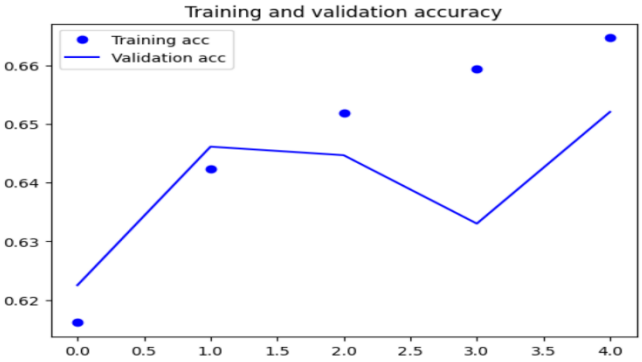

Figure 11: NN Model 3(Accuracy)

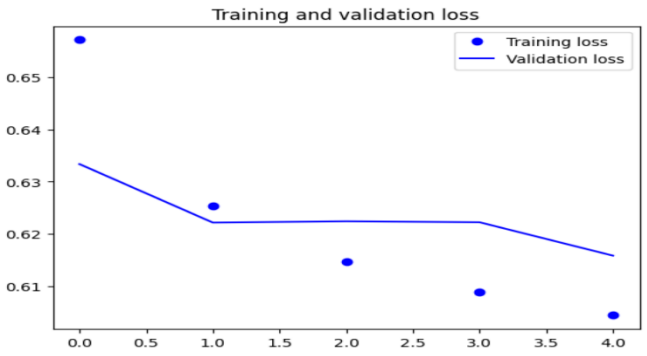

Figure 12: NN Model 3(Loss)

Based on the results, the second model with an extra 1D CNN layer has the highest training accuracy and the lowest training loss, indicating that it performs better than the other models. However, the validation accuracy of this model decreases over time, suggesting that it may be overfitting. The first and third models have more consistent validation accuracy, but their training accuracy and loss are lower than the second model.

In conclusion, while the second model may have better performance in terms of training accuracy and loss, the first and third models may be more reliable in terms of validation accuracy. Further experiments and hyperparameter tuning may be necessary to improve the performance of these models for fake review detection.

V. LIMITATIONS

One of the limitations of our study is the limited number of data points used. We had to sample only 50,000 data points to reduce the computational time and memory usage required for data analysis as our session kept on crashing. Secondly, in terms of classification techniques, the difference in accuracy between the classifiers is relatively small, with the highest accuracy being only 4.74% higher than the lowest accuracy. Hence, exploring other techniques such as Random Forest and Gradient Boosting, along with ensembling approaches like stacking and bagging, may improve the accuracy of fake review detection. Lastly, regarding neural network models, while the second model may have better performance in terms of training accuracy and loss, the first and third models may be more reliable in terms of validation accuracy. However, further experiments and hyperparameter tuning may be necessary to improve the performance of these models for fake review detection.

VI. CONCLUSION

In conclusion, our research focused on the detection of fake reviews in Yelp data. We implemented various techniques and algorithms, including preprocessing, feature engineering, and classification models. Additionally, we explored the effectiveness of neural network models, specifically using LSTM layers, an extra 1D CNN layer on top of LSTM, and bidirectional recurrent layers. Among the classifiers, K-Nearest Neighbors (KNN) achieved the highest accuracy of 57.03%, followed by Logistic Regression with 55.536%. However, all classifiers had relatively similar accuracies around 50%, indicating the need for further improvement.

In terms of neural network models, the second model with an extra 1D CNN layer showed the highest training accuracy but exhibited overfitting tendencies. The first and third models demonstrated more consistent validation accuracy, but their training accuracy and loss were lower. Overall, the neural network models outperformed the classifiers, with the bidirectional recurrent layers achieving the highest accuracy of 64.37%. These models showed promise in detecting fake reviews and can contribute to improving the quality of online review systems.

In conclusion, our research provides insights into the detection of fake reviews on Yelp data and lays the foundation for future studies to enhance the accuracy and effectiveness of such detection systems.

VII. REFERENCES

[1] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews," in Proceedings of the 21st International Conference on World Wide Web, Lyon, France, Apr. 2012, pp. 1-10, doi: 10.1145/2187836.2187879.

[2] K. Jain and S. H. L., "FAKE REVIEW DETECTION ON YELP DATASET," JETIR, vol. 8, no. 8, pp. 456-462, Aug. 2021, Accessed on: Apr. 21, 2023. [Online]. Available: www.jetir.org (ISSN-2349-5162).

[3] M. Nisen, "Fake Reviews Are Becoming an Even Bigger Problem for Businesses," Business Insider, Sep. 19, 2012. [Online]. Available: http://www.businessinsider.com/fake-reviews-arebecoming-a-huge-problem-for-businesses-2012-9.

[4] L. Biewald, "cnn.py," GitHub, 2018. [Online]. Available: https://github.com/lukas/ml-class/blob/master/keras-cnn/cnn.py. [Accessed: 05-May-2023].

[5] M. Raheel, "Sentiment analysis with text mining," Towards Data Science, 24 Oct 2018. [Online]. Available: https://towardsdatascience.com/sentiment-analysis-with-text-mining-13dd2b33de27. [Accessed: 05-May-2023].

[6] P. S. Gudikandula, "Recurrent Neural Networks and LSTM explained," Medium, 29 Aug 2021. [Online]. Available: https://purnasaigudikandula.medium.com/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9. [Accessed: 05-May-2023].

[7] S. Viral, "All classification models explained," Medium, 19 Mar 2021. [Online]. Available: https://iaviral.medium.com/all-classification-models-explained-b03b9b6a4f71. [Accessed: 05-May-2023].

[8] Z. Zhang, "Neural Network-3-projects," GitHub, 2021. [Online]. Available: https://github.com/zzhang83/Neural-Network-3-projects/blob/master/Yelp%20Review%20Classification.ipynb. [Accessed: 05-May-2023].