Sentiment Analysis of the Review: Using Yelp and Rotten Tomatoes Data

Master of Quantitative Economics Program

University of California, Los Angeles

Hsiang-en Ho

Faculty Advisor: Nathan Kunz

May 29, 2022

Abstract

Natural Language Process is a popular technique for processing non-numerical data. This study uses a number of machine learning models to do sentiment analysis, including Naive Bayes, Multinomial Logistic Regression, Decision Tree, Classification Tree, and Random Forest models. The hyperparameter tuning method is then used to improve and validate the models' predictions. In addition, the TF-IDF model is used to generate the new weight for the word information. Yelp and Rotten Tomatoes are the two data sources for this project; both offer text features to train the data and rating scores to test. As a result, all five machine learning algorithms have been shown to improve prediction, and tuning optimizes the forecasting process. Moreover, the success of both approaches suggests that evaluations with the same rating across platforms may share some common qualities. More methodologies for sentiment analysis are likely to complement future study, allowing us to further enhance possible textual feature analysis strategies.

Table of Contents

1. Introduction

Natural language processing (NLP) is the analysis for text data, and it's becoming more popular as more industries learn how to analyze non-numeric datasets efficiently. The rise of computers and technologies sparked a trend of ideas and applications (Koehler, Greenhalgh, and Zellner, 2015). Innovative text and figure recognition technologies have begun to assist humans in solving a variety of difficulties, such as translation and scanning.

Among these, sentiment analysis is the approach used to evaluate emotion polarity in the article. The tool, in particular, may detect positive or negative sentiments toward themes, and it can be tracked back to a very early stage. Tong (2001) conducted sentiment analysis approaches and developed numerous algorithms to address real-world situations. Wilson, Wiebe, and Hoffmann (2005) also made significant progress in analyzing the context's polarity.

As sentiment analysis becomes more widespread, more sentiment analysis is produced. There are several ways to compute the polarity score and assess an article. To begin, there are two common sentiment analysis methods. First, lexicons are a rule-based mechanism that is frequently used to generate product labels by feasible lexicons techniques (Baccianella, Esuli, & Sebastiani, 2010; Taboada, Brooke, Tofiloski, et al., 2011; Hutto &Gilbert, 2014). Consumers, for example, can use the rating scheme to choose which products they wish to buy. Also, n-gram technique is a popular approach from lexicons (Dey, Jenamani, & Thakkar, 2018; Flekova, Preoţiuc-Pietro, & Ruppert, 2015; Moreo et al., 2012). Despite the fact that it is not always publicly accessible, it is nevertheless a popular application for creating merchandise scores. The grading system can make it easier for people to make business or personal decisions.

Second, machine learning is a popular application for acquiring review score using machine learning models. He, Lin, and Alani (2011), for example, used the Joint Sentiment-Topic (JST) model, Naive Bayes (NB) models, Maximum Entropy (ME), and Support Vector Machine models (SVM). Mohammad and Turney (2010) used Maximum Entropy to train and test their seed emotion words.

Mullen and Collier (2004), on the other hand, used a Support Vector Machine to classify attitudes from data sources.

This paper uses natural language processing to investigate the reviews content (NLP). To classify the positive and negative emotions in the reviews, sentiment analysis is used. Naive Bayes, Multinomial Logistic Regression, Decision Tree, Classification Tree, and Random Forest were utilized as classifiers in this study. They are capable of combining the numeric and text features in the first two models. Vectorization and one-hot encoding are employed to aid machine learning models. The following three models, on the other hand, are capable of accurately classifying review content.

The hyperparameter tuning approach is used to increase accuracy. The algorithm for deploying the most efficient machine learning models is called Hyperparameter Tuning. Specifically, when introducing models, there are numerous parameters to set up, and implementing without knowing the feasible values can result in unimpressive predictions. As a result, tuning comes in handy because it finds the optimal settings and so optimizes the models.

The data comes directly from Yelp, which includes 6,478,914 reviews. Yelp is a website where individuals can provide business reviews for stores and businesses. Yelp has been in business since 2004 and continues to thrive, as well as making their public dataset available to the public. This research used the Yelp dataset, which was released on January 19th, 2022, and covered reviews dating back to February 2005. The data source includes information like review ID, user ID, business ID, and the number of useful, funny, and useful reviews received. The text and stars variables, in particular, are useful for training and testing sentiment analysis.

The other dataset will be examined using the same models that have been improved through hyperparameters tuning. If positive and negative reviews share comparable traits and patterns across platforms and websites, the result can be extended to a broader range to forecast polarity. Specifically, comments having the same rating across data sources may have comparable characteristics and patterns.

Rotten Tomatoes' reviews were employed as a second data source in this study. Rotten Tomatoes is also a website where viewers can post comments on movies and television shows. The reviews will then be combined to reflect the video quality. This paper's database comes from Kaggle, and it includes 1,130,017 reviews. The dataset includes review score and review content for testing our sentiment analysis algorithms.

The TextBlob package calculates the sentiment scores. The findings reveal that the higher the polarity, the higher the rating. The benchmark models, on the other hand, are set for prediction using the most common stars. As a consequence, all five models on the Yelp dataset outperform the baseline models, demonstrating that these algorithms improve predicting. The hyperparameter tweaking procedures are then utilized to maximize the models' efficiency. The findings suggest that the tweaks have improved the accuracy of all five models. Namely, they outcompete the benchmark model.

On the Rotten Tomatoes datasets, the tuning models function excellently well. The benchmark model is also set, and the result indicated that the models we train on Yelp are more accurate than the baseline model. As a result, the great prediction on Rotten Tomatoes lends credence to the theory that the reviews have some things in common. Specifically, some traits recognized as higher polarity in the higher review throughout the two data sets, and vice versa.

The TF-IDF (term frequency–inverse document frequency) is also used to recalculate the weights of terms in the context. It's a method for determining the importance and university of vectors and documents. The TF-IDF approach has a higher weight because it is more important and is commonly used in the texts. The results show that the accuracy does not exceed that of hyperparameter tuning models. Despite this, the models outperform the benchmark model.

As a result, all machine learning models outperform the benchmark models in terms of prediction. We discover that they could be used to detect emotions in documents. Furthermore, hyperparameter adjustment increased accuracy. The second conclusion is that the tuning models developed for the Yelp dataset also work well on Rotten Tomatoes, implying that comments with the same rating score may have similar characteristics.

In the future, more modules and empirical strategies for conducting sentiment analysis will be introduced. Part-of-speech analysis, for example, can identify the paragraph's informative terms. Also, the suggestion system may be useful when we wish to improve accuracy by delving deeper into the pattern at individual levels.

2. Theoretical Framework

2.1 One-hot Coding

This paper's two datasets contain both numeric and text data. To use the features from the context data for natural language analysis, the attributes need be translated to vectors (Zhao &Mao, 2017). In the scope, this is referred to as vectorized. The text component was transformed into the document-term matrix in this investigation, as well as the vectors were stacked in the matrix. The document-term matrix is then combined with non-text features in the second step.

2.2 Sentiments Score

The TextBlob Python library is used to determine the sentiment scores. TextBlob is a common natural language processing approach that includes keyword extraction, classification, and part-of-speech models. We run the sentiment analysis software, which is an API that detects emotions from content. The polarity score ranges from -1 to 1, with higher numbers representing more positive reviews and lower scores representing more negative reviews.

2.3 Sentiment Analysis Models

To produce polarity for text data, this paper primarily used machine learning approaches. The first method is to employ Naive Bayes models, which are often used to achieve the goal of document classification (Dey et al., 2016). The Naive Bayes algorithm is based on considering the conditional probability of each word and category in the article. It's highly relying on the assumption that they are independent. The vector is written as the vector given k data points x and class C:

$$\hat{y} = \underset{k \in \{1 \ldots k\}}{\operatorname{argmax}} \; \mathrm{p}(C_k) \prod_{i=1}^{n} p(x_i | C_k)$$

The second model is Multinomial Logistic Regression, which is a classical empirical strategy in the field of economics. When the dependent variable is categorical and ordinal, this regression model works perfectly (Kwak & Clayton-Matthews, 2002). The five-star rating system is the dependent variable in this study. Although it is not the complete discrete options in the latent models, it's treated as the category variable in this research.

The third model is a Decision Tree classifier. Decision Tree is a popular machine learning approach to identify the primary attributes (Song & Ying, 2015). The method was based on the idea of establishing classification rules. For example, samples with scores greater than the point will be assigned to segment A, whereas samples with scores less than the point would be assigned to segment B. The data source will be divided into multiple groups after several rounds. The fourth model is a Classification Tree. It's a similar strategy to the Decision Tree but use different algorithms.

The fifth model is the Random Forest classifier. It's another prominent technique for categorization processes in numerous domains (Breiman, 2001). It's also notable for how well it ranks the polarity score from the context (Belgiu & Drăguţ, 2016). Random Forest, like other classification tree algorithms such as Decision Tree, calculates the Gini Index for the sentiments score (Pal, 2005). According to his research, the Gini index can be written as for providing set W and class S:

$$\hat{y} = \Sigma_{i \neq j} \sum (\mathrm{f}(\mathrm{S_i}, \mathrm{W})/|\mathrm{W}|)\big(\mathrm{f}\big(\mathrm{S_j}, \mathrm{W}\big)/|\mathrm{W}|\big)$$

2.4 Hyperparameter Tuning

The sentiment score is calculated using the five models listed above. Following that, the hyperparameter tuning method is used. It is commonly utilized in the machine learning model to accurately capture associations. Elgeldawi et al. (2021) characterized the method as maximizing machine learning model predictions. The approach is carried out in this study in the traditional

manner: grid research. Scikit-learn will establish several parameters, which will subsequently be concatenated into different subsets.

Multiple parameters will be evaluated during the grid research. The cross-validation method used in this process is Repeated Stratified K Fold. Cross-validation is a widely used statistical approach for obtaining a prediction result. In addition, K-fold cross-validation is a common strategy for achieving the goal. Berrar (2019) defined the procedure as "random sampling without repeating the same baskets." The validation will look at whether the prediction performed well on the training or testing sets. Different sets' pairs will be calculated. Consequently, the precision of every conceivable combination will be displayed. Then, as a result of the hyperparameter tuning models, we chose the models with the highest accuracy.

2.5 TF-IDF

TF-IDF (term frequency–inverse document frequency) is a method for determining the relevance of words. This method will determine the weight for each word in its own unique way. Joachims (1996) divided the concept into two pieces. The Term frequency is the first section. Each text is represented by separate vectors d and t, which denote the number of words that exist in the content. Hence, the word tf(t, d) can be used to express the vocabulary' relative frequency:.

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\Sigma_{t' \in d} f_{t',d}}$$

Inverse document frequency, on the other hand, is determined from document frequency. For example, if the terms are often used and appear across multiple textual vectors, they will be classified as informative words. The logarithm will also be used to transform the phrase. |D| represents the total number of contents, and DF(t) is the percentage of the text in which the vocabulary appears at least once. The formula will then be:

$$\mathrm{idf}(t) = log\left(\frac{|D|}{\mathrm{DF(t)}}\right)$$

Combining these two algorithms yields the TF-IDF formula:

$$d^{(i)} = tf(t,d) * idf(t)$$

As a result, the TF-IDF can operate as two indicators for evaluating document words. The words will be differentiated as more essential than other words if they appear in the context more than once. Furthermore, vocabularies that are more prevalent across various contents will be regarded as more universal and important. Consequently, they provide the ideal weight for the natural language process.

3. Data

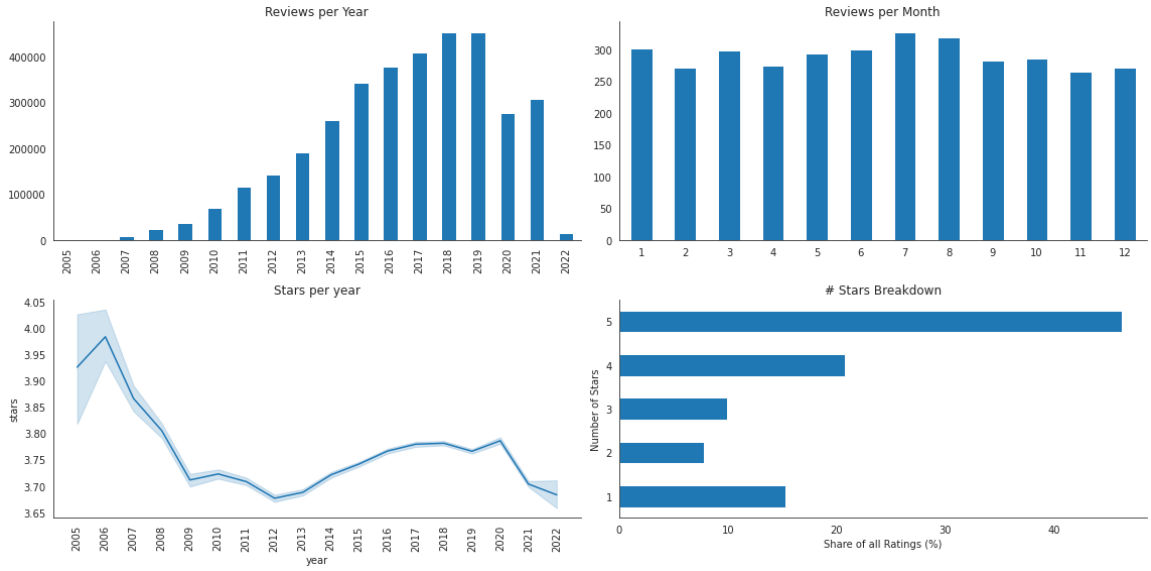
3.1 Yelp dataset

The Yelp data is downloaded directly from their website, and the most current version was released on January 19th, 2022. The most essential variable in yelp is review content, which allows us to perform sentiment analysis on the themes. The reviews are written in English, and the total number of reviews in the dataset is 6,478,914. It is then being randomly sampling to 3,495,140 reviews.

The first column in the dataset is the review's id, which is used to identify each review. The second element is the user's id, which has 1,311,610 unique values because some users write multiple comments. The third attribute is the business id, which has 149,593 unique values. The fifth to seventh attributes are the number of useful, funny, and cool votes received. Also, the ninth element is the date of the comment.

The major outcome variable in this paper is in the fourth column. The stars are given based on user feedback. Yelp uses a five-star rating system, with all reviewer scores being integers only. The higher the score, the more satisfied they are. For example, a five-star rating indicates that the users are really pleased with the product or service. The text of the reviews, on the other hand, is the most important predictor because it will be used to calculate the sentiment polarity.

Figure 1: Descriptive statistics: Yelp



The distribution of reviews and stars is shown in Figure 1. The upper left plot depicts the number of reviews by year, with more reviews from 2015 to 2019. The upper right plot shows the reviews by month, with reviews distributed evenly throughout all months. The lower left figure shows the average stars giving per year, with stars ranging from 3.7 to 4, as well as a downslope curve. The lower right figure is about the star distributions, and the five-stars comment compose the majority of the sample.

3.2 Rotten Tomatoes dataset

The second piece of data comes from Rotten Tomatoes. This movie and television show review website will aggregate users' comment in order to assess the videos. The data for this research comes from Kaggle and spans 1,130,017 reviews before December 2020.

The review score is the most important variable in the dataset. Rotten Tomatoes displays the score based on audience ratings and the overall score. 3.5/4, 4/5, or 8/10, for example. To compare the model tuning from the Yelp dataset, this study only keeps reviews using a five-star rating system. In

addition, only point-five or integer scores will be included in the dataset. Furthermore, comments before to 2000 are excluded from the sample. After cleaning, the sample has 308,303 reviews.

Figure 2: Descriptive statistics: Rotten Tomatoes

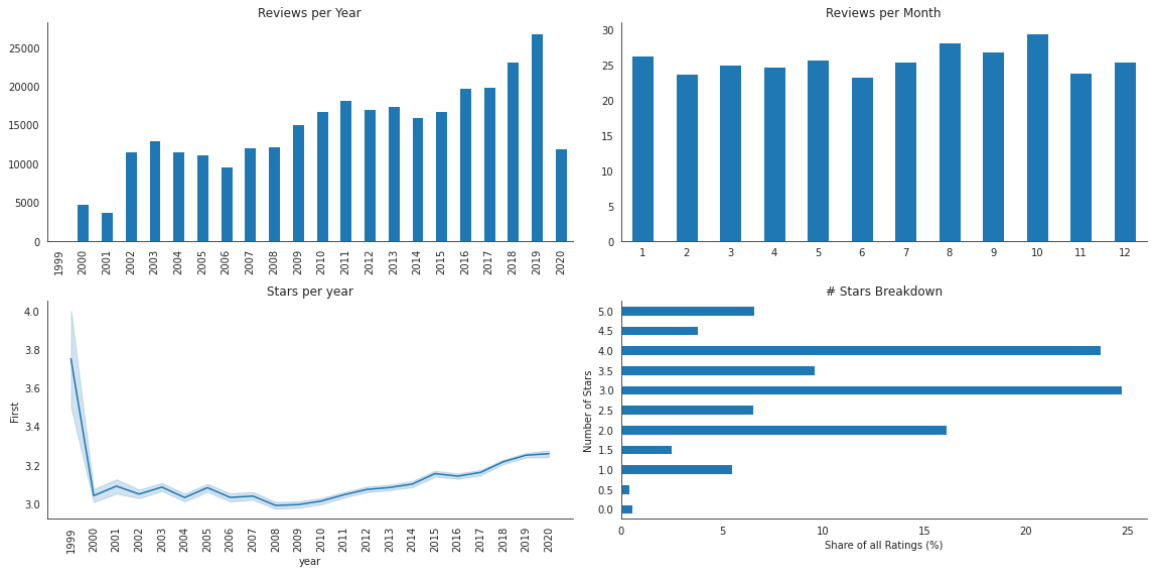


Figure 2 shows the distribution of reviews and stars. The number of reviews per year is depicted in the upper left plot, with more reviews after 2001. The plot on the upper right shows the reviews by month, with reviews evenly distributed throughout all months. The average stars giving per year is shown in the lower left figure, with stars ranging from 3 to 3.2 except for 1999, as well as a downslope curve. The star distributions are depicted in the lower right picture, and the two-, three-, and four-stars comments make up the majority of the sample.

4. Result

The first step of the analysis process is to create the sentiments score using the Python TextBlob module, which will recognize the emotion in the documents. The Yelp datasets are then divided into two sets: training and testing, with 723,252 and 602,130 observations, respectively. The next stage is to run the benchmark, Naive Bayes, Multinomial Logistic Regression, Decision Tree, Classification Tree, and Random Forest models. These models are run without specifying any parameters and are thus set as default.

The hyperparameter tuning is then performed on the Yelp dataset's testing set. We performed grid research to find the best parameters for each model in order to optimize the predictions. The tuning models will next be applied to the Rotten tomatoes to test the generality of two different data sources.

In the last stage, the TF-IDF technique was introduced. In order to train the models, we first lemmatize the text in both the training and testing sets from the Yelp dataset, removing punctuation and stop words. Following that, the training and testing sets are converted to TF-IDF matrix format. The training set is used to train our five models, such as the Naive Bayes model, and the testing set is used to test them. The technique is then repeated with the Rotten Tomatoes dataset from the lemmatization step.

4.1 Sentiments score

Figure 3: Sentiment scores distribution by stars

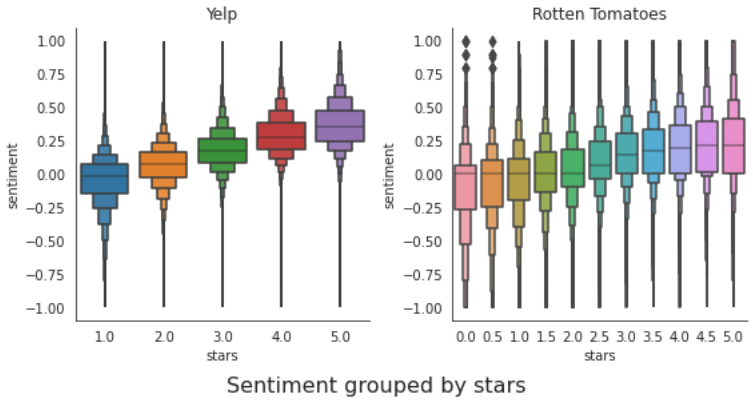


Figure 3 depicts the sentiment score distribution in both datasets. The one-star comments have a lower average score of around 0, while the five-star evaluations have a higher average score of about 0.35, as shown on the left-side plot. The right-hand figure illustrates that one-star reviews have a lower average score of around -0.1, whereas five-star reviews have a higher average score of around 0.2. In the Rotten Tomatoes dataset, the difference is lower than in the Yelp dataset. As a result, the results show that the stars have a consistently positive association with polarity ratings, which confirms the beliefs. In other words, evaluations that are identified as more positive are more likely to receive higher star ratings.

4.2 Hyperparameter tuning result

Table 1: Hyperparameter tuning result

Model	Method	Best Hyperparameters
Naive Bayes	Grid Research	Additive smoothing parameter: 100, Whether to learn class prior probabilities or not: 'True'
Multinomial Logistic Regression	Grid Research	Inverse of regularization strength: 0.1, Algorithm to use in the optimization problem: 'saga'
Decision Tree	Grid Research	The maximum depth of the tree: 5, The number of features to consider when looking for the best split: 'auto', The minimum number of samples required to be at a leaf node: 10
Classification Tree	Grid Research	The maximum depth of the tree: 5, The number of features to consider when looking for the best split: 'auto', The minimum number of samples required to be at a leaf node: 25
Random Forest	Grid Research	The number of features to consider when looking for the best split: 'sqrt', The minimum number of samples required to be at a leaf node: 5, The number of trees in the forest: 50

The grid research is used to implement the Hyperparameter tuning method. The parameters, such as the depth and leaf of the Decision Tree, are set using the Scikit-learn module. Only the combination of potential parameters is considered for efficiency. 'l2' or 'no' penalties, for example,

are not supported except for the solver hyperparameter value 'sag'. As a result, when we set the value 'sag,' they do not need to be particularly set.

4.3 Yelp dataset: Models accuracy

Table 2. Accuracy comparisons of five models: Yelp

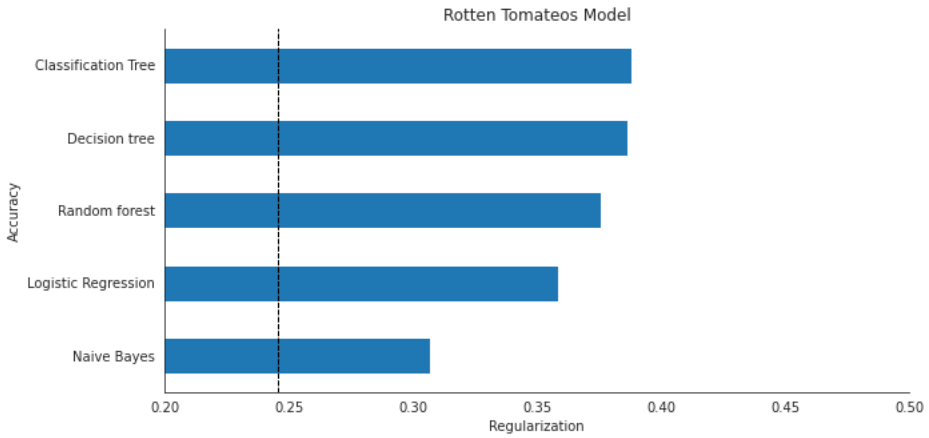
	Accuracy (in %)	
Model	**Original Model**	**Hyperparameter tuning**
Benchmark	53.54%	
Naive Bayes	64.94%	66.60%
Multinomial Logistic Regression	74.69%	75.88%
Decision Tree	66.39%	67.03%
Classification Tree	55.61%	66.95%
Random Forest	56.85%	61.53%

The accuracy comparisons of five models are shown in Table 1. The accuracy of the benchmark model is 53.54 percent. The accuracy scores of the original models are listed in the first column of the table. Precisely, the original models were built without any parameters which would be considered default. Among all, the accuracy of the Multinomial logistic regression is 74.69 percent, a 21.15 percent improvement. Moreover, all five models increase the accuracy of all models by 2.07 percent to 21.15 percent.

The hyperparameter tuning methodology is then performed, and the results are displayed in the second column. In order to enhance accuracy, the parameters are created after the hyperparameter tuning, as indicated in Table 1. All of the adjusted models have an accuracy of more than 60%. The Multinomial logistic regression still has the highest accuracy as 75.88 percent. Overall, hyperparameter adjustment increases the accuracy of all models by 0.64 percent to 11.34 percent than the original models.

4.4 Rotten Tomatoes: Models Accuracy

Figure 4: Accuracy comparisons of five models: Rotten Tomatoes



The Rotten Tomatoes dataset is then used to evaluate our hyperparameter tuning models. Figure 4 depicts the final result. Among all, the Decision Tree's accuracy is 38.63 percent, an increase of 14.06 percent. Also, the Classification Tree's accuracy is 38.8 percent, an increase of 14.23 percent. As a result, all of the models we trained and tuned on the Yelp dataset predict well on the Rotten Tomatoes dataset, boosting their accuracy by 6.14 percent to 14.23 percent.

4.5 TF-IDF model accuracy

Figure 5: TF-IDF Accuracy comparisons of five models

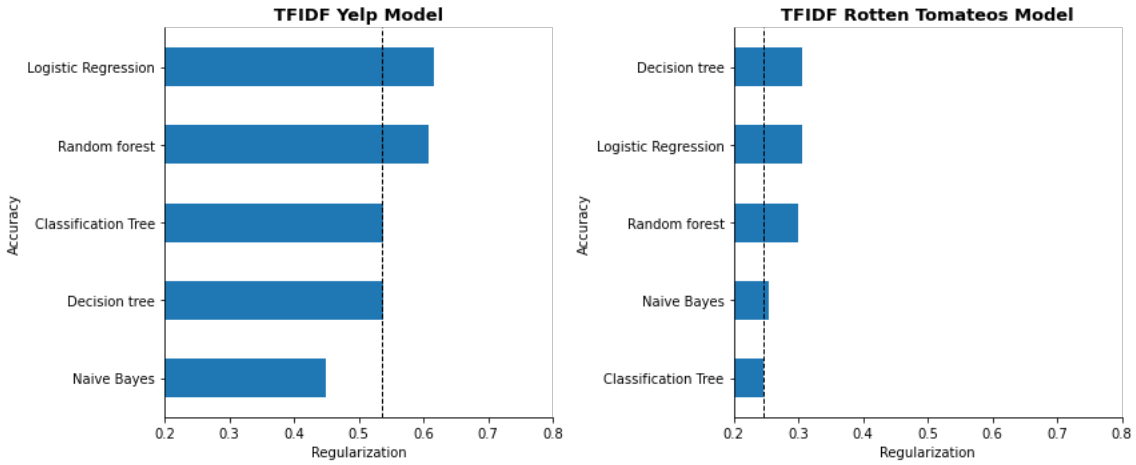


Figure 5 shows how the TF-IDF approach was applied to the five models. The Yelp dataset is on the left side of the figure. The Multinomial logistic regression has an accuracy of 61.55 percent, an improvement of 8.01 percent. Also, the Random Forest's accuracy is 60.72 percent, an increase of 7.18 percent. On the contrary, the Naive Bayes model has a 44.9 percent accuracy, which is 8.65

percent lower than the benchmark. As a result, the Multinomial logistic regression and Random Forest models outperform the benchmark model in terms of accuracy. The Classification Tree model and the Decision Tree model have similar accuracy. Only the Naive Bayes model, on the other hand, is less accurate than the benchmark model.

The right side is the TF-IDF method on the Rotten Tomatoes dataset. The Multinomial logistic regression and Decision Tree have the accuracy of 30.53 percent, an improvement of 5.96 percent. As a result, the Multinomial Logistic Regression and Random Forest models outperform the benchmark model in terms of accuracy. The Classification Tree model and the Decision Tree model have similar accuracy. Only the Naive Bayes model, on the other hand, is less accurate than the benchmark model.

5. Conclusion and Future work

In current civilization, the Natural Language Process is a popular technique for analyzing textual elements. Keyword extraction, dimension deduction, grammar check, and translation algorithms are extremely useful and convenient for people to complete their daily tasks.

This research does sentiment analysis using different machine learning models, which is a way to distinguishing positive and negative emotions from documents. The TextBlob Python module identifies the polarity scores. Then, to predict the rating scores from the reviews, the Naive Bayes, Multinomial Logistic Regression, Decision Tree, Classification Tree, and Random Forest models are used. In addition, we use hyperparameter tuning approach to optimize the process of determining the appropriate model parameters.

This study uses the Yelp dataset to train and adjust the models, which has millions of review data sources, before validating and testing the algorithm's accuracy. Furthermore, the tuning models will be used to investigate the Rotten Tomatoes dataset, which is a platform for aggregating movie and TV program reviews. Last but not least, the TF-IDF models are used to build new weights for information and the significance of words in documents.

The result demonstrates that all five machine learning models on the Yelp dataset outperform the benchmark, which predicts using the most common rating stars. The hyperparameter successfully improves the forecast. The findings, on the other side, reveal that adjusting models enhance Rotten Tomatoes accuracy. The TF-IDF models, on the other hand, perform worse than the hyperparameter models, original models, and even benchmark models in terms of prediction.

The first conclusion is made from the above result. Out five machine learning models predict well on the Yelp and Rotten Tomatoes dataset, proving they are potential to forecast the textual features in the documents. Also, the hyperparameter tuning is an excellent technique to improve the accuracy of the models.

The second conclusion is that the assessments from the two data sources may have some similarities. Specifically, the high and low rated comments may contain comparable textual vectors, and so the identified strategies work well on both.

The future work will emphasize to develop more strategies for the sentiment analysis. For instance, there are many techniques to generate polarity scores or relative analysis, including the n-grams, Lexicon, support vector machines, Bag-of-word, and Maximum Entropy. For instance, the POC (part-of-speech) analysis can come to the aid to identify the common word in the documents. Specifically, the adjacent words can be picked as a subset and find the most frequent vocabularies as the description in the comments.

Another interesting topic is the analysis for identifying persons, such as the recommendation system. Specifically, by including the same users' decisions and characteristics. Individual features might be assigned by the system. The fixed-effect model, on the other hand, may be useful in controlling unobserved characteristics across people. Our data sources include the users' IDs, allowing us to look at their review behavior individually.

6. Reference

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, *114*, 24-31.

Berrar, D. (2019). Cross-Validation.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Dey, A., Jenamani, M., & Thakkar, J. J. (2018). Senti-N-Gram: An n-gram lexicon for sentiment analysis. *Expert Systems with Applications*, *103*, 92-105.

Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. *arXiv preprint arXiv:1610.09982*.

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021, December). Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. In *Informatics* (Vol. 8, No. 4, p. 79). Multidisciplinary Digital Publishing Institute.

Flekova, L., Preoţiuc-Pietro, D., & Ruppert, E. (2015, September). Analysing domain suitability of a sentiment lexicon by identifying distributionally bipolar words. Paper presented at the annual meeting of the *Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Lisboa, Portugal.

He, Yulan; Lin, Chenghua and Alani, Harith (2011, June). *Automatically extracting polarity-bearing topics for cross- domain sentiment classification*. Paper presented at the annual meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR.

Hutto, C., & Gilbert, E. (2014, May). *Vader: A parsimonious rule-based model for sentiment analysis of social media text.* Paper presented at the annual meeting of the Proceedings of the international AAAI conference on web and social media, Ann Arbor, MI.

Joachims T. (1996, July) *A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization*, Paper presented at the annual meeting of the International Conference on Machine Learning, Bari, Italy.

Koehler, M., Greenhalgh, S., & Zellner, A. (2015, November). *Potential applications of sentiment analysis in educational research and practice–Is SITE the friendliest conference?*. Paper presented at the annual meeting of the Society for Information Technology & Teacher Education International, Conferenc Waynesville, NC.

Kwak, C., & Clayton-Matthews, A. (2002). Multinomial logistic regression. *Nursing research*, *51*(6), 404-410.

Mohammad, S., & Turney, P. (2010, June). *Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon*. Paper presented at the annual meeting of the Computational Approaches to Analysis and Generation of Emotion in Text, Los Angeles, CA.

Moreo, A., Romero, M., Castro, J. L., & Zurita, J. M. (2012). Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications*, *39*(10), 9166-9180.

Mullen, T., & Collier, N. (2004, July). *Sentiment analysis using support vector machines with diverse information sources*. Paper presented at the annual meeting of the Empirical Methods in Natural Language Processing, Barcelona, Spain.

Pal, M. (2005). Random forest classifier for remote sensing classification. *International journal of remote sensing*, *26*(1), 217-222.

Song, Y. Y., & Ying, L. U. (2015). Decision Tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130-135.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, *37*(2), 267-307.

Tong, R. M. (2001, September). An operational system for detecting and tracking opinions in on-line discussion. Paper presented at the meeting of the *ACM SIGIR 2001 Workshop on Operational Text Classification*, New Orleans, LA.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005, October). Recognizing contextual polarity in phrase-level sentiment analysis. Paper presented at the meeting of *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, Vancouver, British Columbia, Canada.

Zhao, R., & Mao, K. (2017). Fuzzy bag-of-words model for document representation. *IEEE transactions on fuzzy systems*, *26*(2), 794-804.