Random Forest Model on Prediction of Users' Reviews Using Text Mining

2023-12-02

Student ID: U2016609

Github link: https://github.com/xichuqing/Random-forest-model-for-prediction-of-users-review-using-text-mining (https://github.com/xichuqing/Random-forest-model-for-prediction-of-users-review-using-text-mining)

Tabula statement
We're part of an academic community at Warwick. Whether studying, teaching, or researching, we're all taking part in an expert conversation which must meet standards of academic integrity. When we all meet these standards, we can take pride in our own academic achievements, as individuals and as an academic community. Academic integrity means committing to honesty in academic work, giving credit where we've used others' ideas and being proud of our own achievements. In submitting my work I confirm that:

1. I have read the guidance on academic integrity provided in the Student Handbook and understand the University regulations in relation to Academic Integrity. I am aware of the potential consequences of Academic Misconduct.

2. I declare that the work is all my own, except where I have stated otherwise.

3. No substantial part(s) of the work submitted here has also been submitted by me in other credit bearing assessments courses of study (other than in certain cases of a resubmission of a piece of work), and I acknowledge that if this has been done this may lead to an appropriate sanction.

4. Where a generative Artificial Intelligence such as ChatGPT has been used I confirm I have abided by both the University guidance and specific requirements as set out in the Student Handbook and the Assessment brief. I have clearly acknowledged the use of any generative Artificial Intelligence in my submission, my reasoning for using it and which generative AI (or AIs) I have used. Except where indicated the work is otherwise entirely my own.

5. I understand that should this piece of work raise concerns requiring investigation in relation to any of points above, it is possible that other work I have submitted for assessment will be checked, even if marks (provisional or confirmed) have been published.

6. Where a proof-reader, paid or unpaid was used, I confirm that the proofreader was made aware of and has complied with the University's proofreading policy.

7. I consent that my work may be submitted to Turnitin or other analytical technology. I understand the use of this service (or similar), along with other methods of maintaining the integrity of the academic process, will help the University uphold academic standards and assessment fairness.

Privacy statement
The data on this form relates to your submission of coursework. The date and time of your submission, your identity, and the work you have submitted will be stored. We will only use this data to administer and record your coursework submission.
Related articles Reg. 11
Academic Integrity (from 4 Oct 2021)
Guidance on Regulation 11 Proofreading Policy
Education Policy and Quality Team Academic Integrity (warwick.ac.uk)

Introduction

The internet enables individuals to share their opinions by posting text on various platforms. It becomes crucial for businesses to analyse users' reviews since it can improve their service quality based on the analysis by establishing relevant strategies.

OSEMN framework

This paper will base on the OSEMN framework (Obtain, Scrub, Explore, Model, and iNterpret) to predict Yelp users' ratings on businesses. OSEMN is a data science methodology which prioritizes a direct route to model analysis with efficiency and less preliminary business understanding process (Saltz, Sutherland and Hotz 2022). The focus is to construct and select models. Selecting model should be paid caution to since the Yelp ratings can be considered as categorical, which are incompatible with some linear models. Additionally, the consumers' reviews are text contents, so the computational algorithms is needed for model construction. OSEMN framework satisfies this priority.Thus, this paper uses classification random forest models with text mining to predict users' reviews on Yelp.

Methodology

A. Obtain and Scrub

The original datasets are available on the Yelp website (Yelp, 2019). It contains ratings and other relevant information about users and businesses. Some sample data are randomly drawn for quicker processing. After taking the overlap and merging three of the original datasets, dropping all the missing values and the variables extremely skewed, a sample size of 279,878 data was generated.

B. Explore

summary statistics

To capture the features of the variables that will be included in the model, summary statistics is shown as follows:

variable	Min	Max	Median	Mean
stars.y	1	5	4	3.75
funny.y	0	346	0	0.33
useful.y	0	1182	0	1.18
cool.y	0	400	0	0.50
average_stars	1	5	3.88	3.746
review_count.y	0	8363	24	114.6
review_count.x	5	568	136	370
fans	0	2547	0	11.8
stars.x	1	5	4	3.751
compliment_plain	0	8974	0	30.34
latitude	27.56	53.65	38.6	35.89
longitude	-120.1	-74.66	-86.18	-89.64

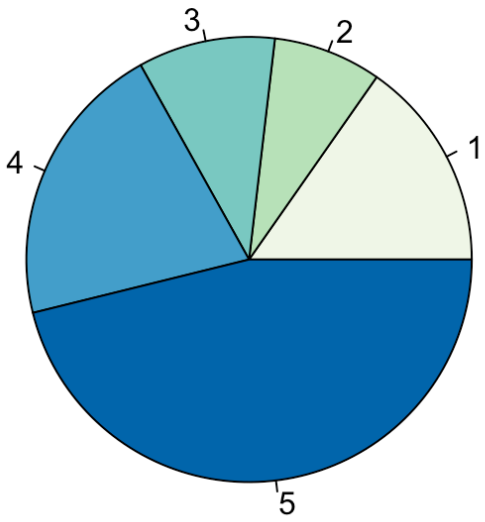
variable	Min	Max	Median	Mean
is_open	0	1	1	0.83
text	\	\	\	\

The relevant variables used in this paper are described in the table below:

Variables	Description
text	users' text comments
stars.y	the response variable, discrete number range from 1 to 5, means the rates users gave to the business
stars.x	the average stars the business got
average_stars	the average stars the users gave to every business
fans	the number of fans of the user
review_count.y	discrete, means the accumulate review count of the user
review_count.x	discrete, means the accumulate review count of the business
funny.x	discrete, meaning the number of users who think the comment is funny
useful.x	discrete, meaning the number of users who think the comment is useful
cool.x	discrete, meaning the number of users who think the comment is cool
is_open	binary, 1 represents the business is opened while 0 means it is closed
compliment_plain	users' information, discrete
longitude	longitude of the business, continuous
latitude	latitude of the business, continuous

Visualization

The variables with median smaller than mean are skewed to the lower values. In contrast, the median of *stars.y* is bigger than the mean, indicating the data skews towards higher values. As can be seen in the below pie chart, it indicates the portion of each class of stars. Most people rated 5 and 4 stars, while a few people rated 2 and 3.



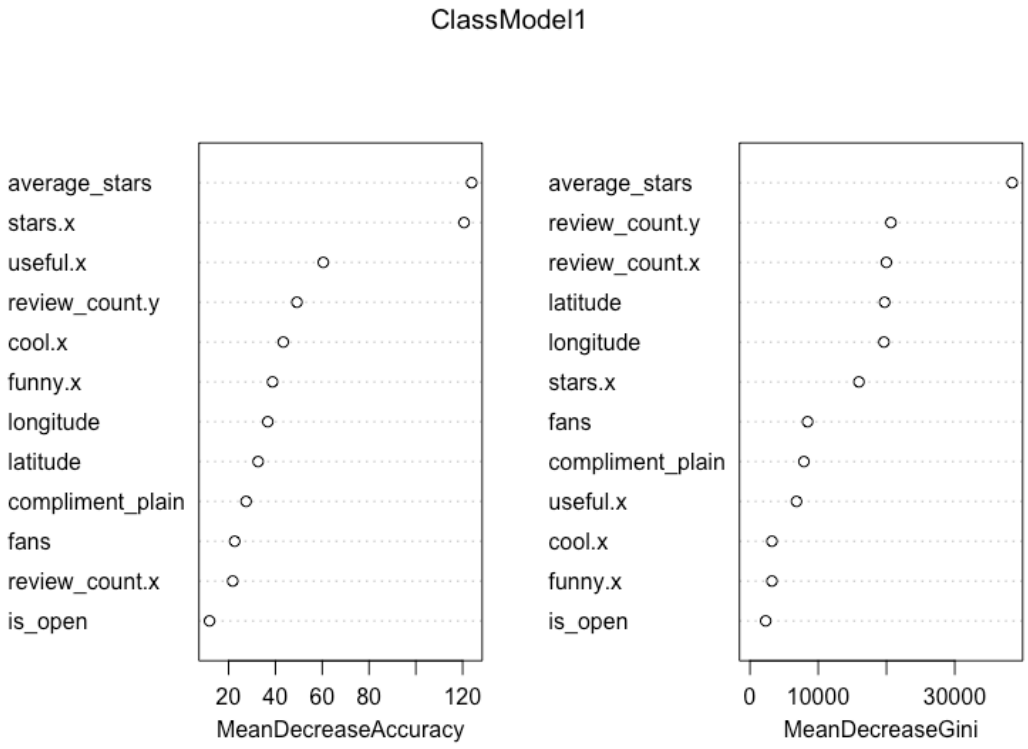
After the tokenization, a word cloud with decreasing word frequency is presented below. As can be seen, most of the words are positive or neutral.

D. Model

Random Forest Classification Model 1

Random forest is a supervised learning algorithm and often be used in regression and classification problems (Karthika, Murugeswari and Manoranjithem, 2019). *M* training samples are constructed to build multiple decision trees which will be merged together to gain a stable value with variance be averaged, where *M* usually equals the square roots of the sample size (Hegde and Padma, 2017). *stars.y* is discrete numeric data with only 5 outcomes, so the prediction should be discrete as well to obtain the precise ratings. Thus, *stars.y* will be treated as categorical variable in classification tree. After drawing 10,000 testing data randomly, the models below are constructed based on the rest of the training data.

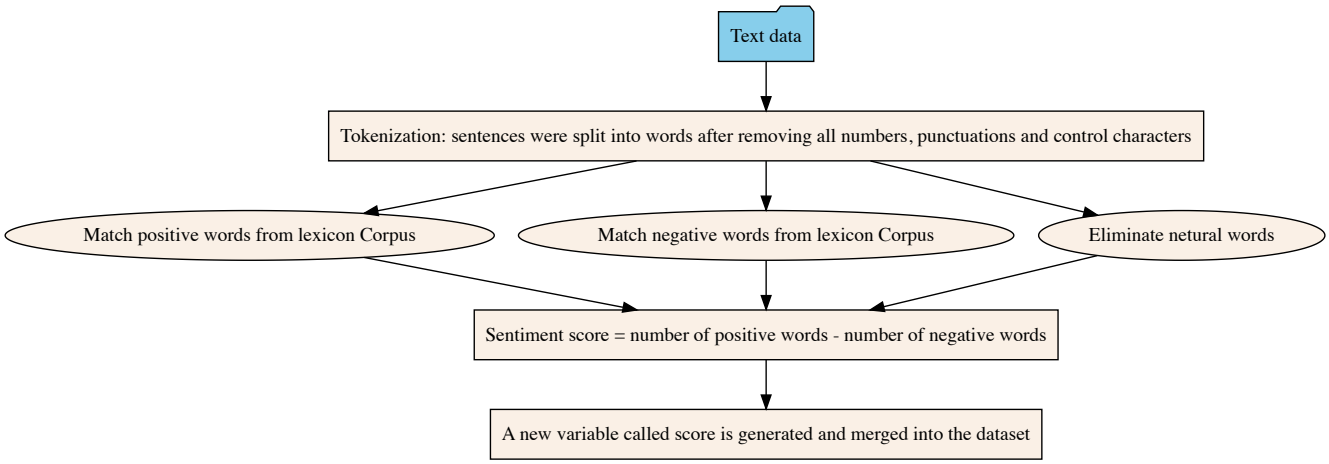
A classification tree model (Class Model 1) is constructed for predicting *stars.y.* Since we can deduced that *stars.y* is correlated with *stars.x* and *average_stars*, these variables will be included. Additionally, *review_count.x, review_count.y, useful.x, funny.x, cool.x, latitude, longitude, is_open, compliment_plain* and *fans* are included to test the importance of the variables.



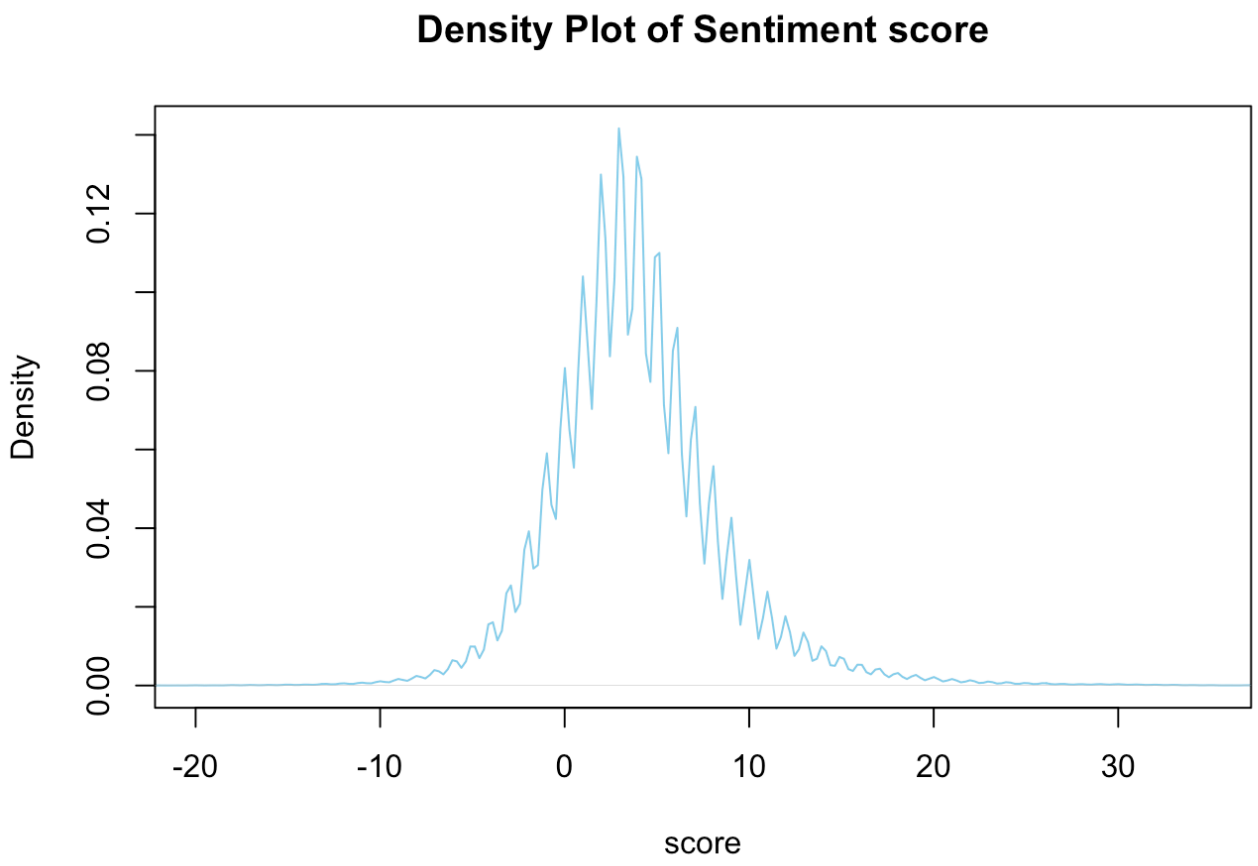
Gini index represents the node impurity. Summing up the Gini decreases for each individual variable over all trees in the forest gives a variable importance (Hegde and Padma, 2017). In general, the higher value of mean decrease accuracy and mean decrease Gini, the more important the variable is to the model. As can be seen in the graph above, *average_stars* is important to explain the response variable. To improve the model further, text mining will be used for generating sentiment score.

Text Mining

Text data presents users' attitudes, it gives implications on the ratings. Text Mining is used for exploring the attitudes. Such as sentiment analysis, a computational study of people's opinions, emotions, and attitudes towards an entity (Medhat, Hassan and Korashy, 2014), can provide sentiment scores according to the negative and positive opinion words in the text. Based on the lexicon corpus provided by Murali (2017), the process of calculating sentiment score is as follows:



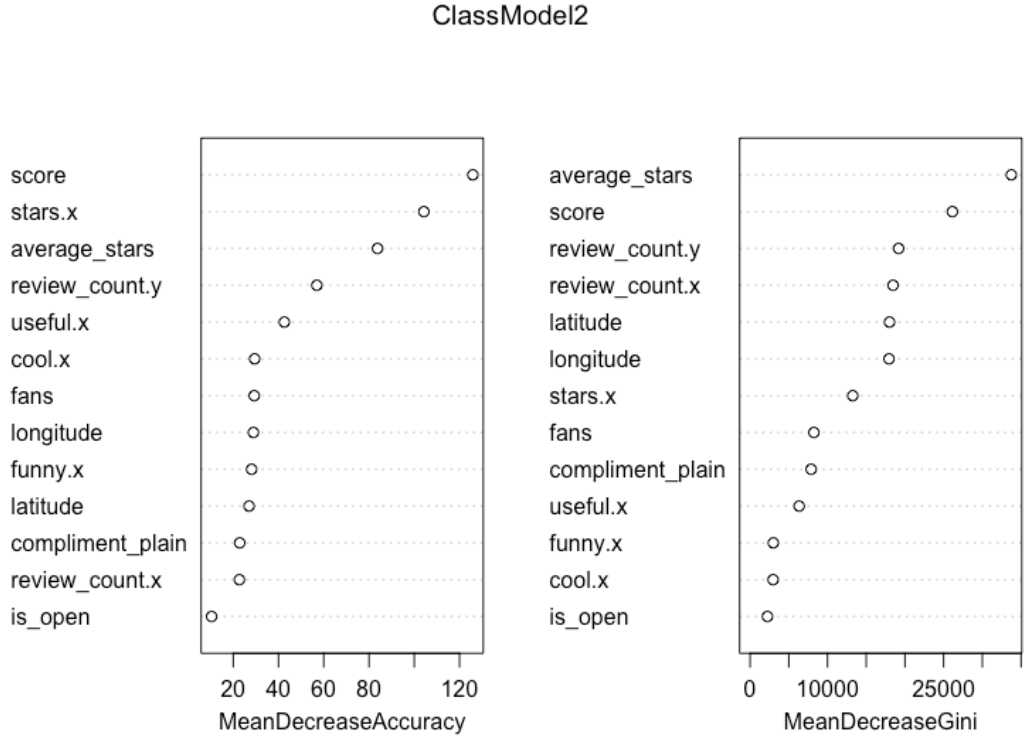
- If the score > 0, the text has an overall 'positive opinion'
- If the score = 0, the text has an overall 'neutral opinion'
- If the score < 0, the text has an overall 'negative opinion'



A density plot of the sentiment score is shown above. It asymptotically follows a normal distribution with a mean of 3.89. The minimum score is -66, while the maximum score is 57. It implies most of the users hold positive opinions, which is in line with the *stars.y* pie chart and the word cloud, indicating a relation between *score* and the response variable *stars.y*.

Random Forest Classification Model 2

Class Model 2 with 80 trees is constructed based on Class Model 1 with *score* included to see the impact of sentiment score. The variable importance graph is as follows:



As can be seen, *score* tops the list, meaning it is the most important variable among all.

E. Result and Interpretation

Performance

The above Class Models' performance table is shown as follows:

Model	OOB error rate	variables at each split	Number of tree
Class Model 1	41.93%	3	80
Class Model 2	39.04%	3	80

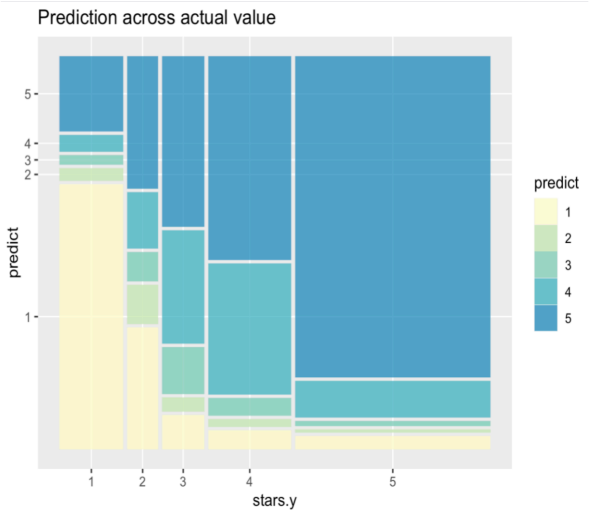
Some observations do not add to the boostrap sample when constructing the model and these are referred to "out of bag data", which are useful for estimating generalization error and variable importance (Prajwala, 2015). As can be seen, model with *score* performs better with lower OOB error rate, indicating a higher accuracy for predicting.

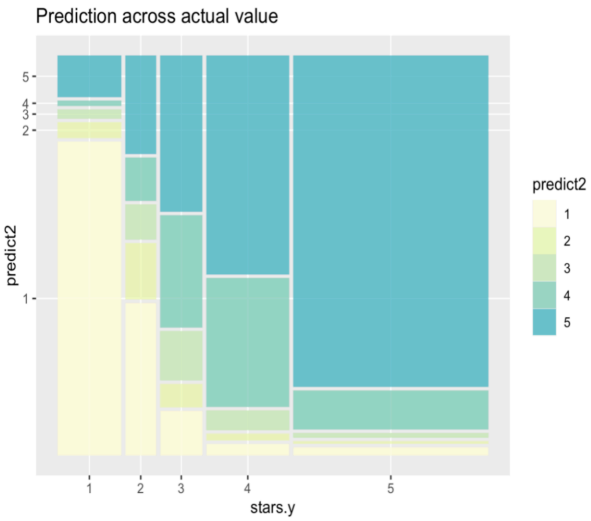
Prediction

Testing data is used to test the model's performance. In terms of accuracy and kappa, class model 2 performs better, which means the correct prediction probability and agreement between sets of observations are relatively high. As can be seen in the table below, including sentiment scores improves the model's performance by 2.44% in accuracy.

Model	Class Model 1	Class Model 2
Accuracy	59.73%	62.17%
Kappa	0.3794	0.4218

See the confusion matrix in the appendix, the sensitivity and accuracy for different classes diverge. For class 5, the sensitivity and balanced accuracy values are high. Take class model 2 as the case: for class 1 and 5, the model performs well. It has 82% and 86% sensitivity, 87% and 75% accuracy, and 93% and 64% specificity, respectively. But for class 2, 3, and 4, the sensitivity is 15%, 13%, and 33%, respectively, and the balanced accuracy is 56%, 55%, and 61%, respectively, which are low. The Mosaic plot of prediction and actual value is presented below:





The rectangular tile represents a combination of levels from prediction and actual value. Tile size is proportional to the number of predictions falling into the correct actual class. As can be seen, the proportion of correct predictions of class 1 and 5 are high, followed by class 4.

Interpretation

Sentiment analysis of text can improve the accuracy of predictions on stars ratings. The classification model 2 can generate the most accurate prediction with a 62.17% probability for the testing data. For 1 and 5 stars, the true positive and negative cases are well identified by the model, as well as the accuracy rate, while stars 2 and 3 have the opposite situation.

Discussion

Limitation

The lexicon-corpus-based approach in sentiment analysis is unable to find opinion words with domain and context-specific orientation, and this will cause bias during the text analysis.
Although *average_stars* can represent the pattern of the user when he rate the business, reverse causality may still occur between *stars.y* and *average_stars*. Increasing *stars.y* will increase *average_stars*, but increasing in *average_stars* may not lead to increase in *stars*.

Difficulty

It is hard to determine which model fits this dataset the best. When facing with large size of data, testing all the proposed model is time consuming. Considering there are only 5 outcomes for *stars.y*, i decided to treat it as categorical and use decision tree. However, i tried different explaining variables in the model and trained many times, the accuracy of the model was still below 60% with high error rate. Then, i used sentiment analysis to assign sentiment score to the data and it improves the model performance and finally the accuracy reached above 60%.

Conclusion

The classification random forest model can perform generally well in predicting users' review on businesses, especially when the actual stars ratings are 1 or 5. Sentiment score gives good implication on ratings. More advanced natural language techniques which can find opinion words with context-specific orientation are suggested for further study.

Reference

Hegde, Y. and Padma, S.K., 2017, January. Sentiment analysis using random forest ensemble for mobile product reviews in Kannada. In *2017 IEEE 7th international advance computing conference (IACC)* (pp. 777-782). IEEE.

Karthika, P., Murugeswari, R. and Manoranjithem, R., 2019, April. Sentiment analysis of social media network using random forest algorithm. In *2019 IEEE international conference on intelligent techniques in control, optimization and signal processing (INCOS)* (pp. 1-5). IEEE.

Medhat, W., Hassan, A. and Korashy, H., 2014. *Sentiment analysis algorithms and applications: A survey.* Ain Shams engineering journal, 5(4), pp.1093-1113.

Murali, S. (2018). *Sentiment-Analysis-of-Twitter-Data-by-Lexicon-Approach.* [online] GitHub. Available at: https://github.com/Surya-Murali/Sentiment-Analysis-of-Twitter-Data-by-Lexicon-Approach/tree/master (https://github.com/Surya-Murali/Sentiment-Analysis-of-Twitter-Data-by-Lexicon-Approach/tree/master) [Accessed 27 Nov. 2023].

Prajwala, T.R., 2015. A comparative study on decision tree and random forest using R tool. *International journal of advanced research in computer and communication engineering*, *4*(1), pp.196-199.

Saltz, J., Sutherland, A. and Hotz, N., 2022. *Achieving Lean Data Science Agility Via Data Driven Scrum.*,Proceedings of the 55th Hawaii International Conference on System Sciences

Yelp (2019). *Yelp Dataset*. [online] Available at: https://www.yelp.com/dataset. (https://www.yelp.com/dataset.)

Appendix

Class Model 1:

```
Call:
 randomForest(formula = stars.y ~ average_stars + stars.x + review_count.x +      fans +
review_count.y + useful.x + funny.x + cool.x + latitude +      longitude + is_open + comp
liment_plain, data = train, ntree = 80,      importance = T)
               Type of random forest: classification
                     Number of trees: 80
No. of variables tried at each split: 3

        OOB estimate of  error rate: 41.93%
Confusion matrix:
      1    2    3     4      5 class.error
1 28396 1582 1026  1904   8337   0.3115287
2  6231 2722 1590  3415   7292   0.8719059
3  3096 1155 3539  8022  11038   0.8681937
4  2893 1065 3467 18225  30446   0.6751105
5  4869  991 1928 12808 103841   0.1655135
```

Confusion Matrix of Class Model 1:

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5
         1 1077  238   91   99  162
         2   50   76   37   39   35
         3   38   57  125   92   59
         4   67  110  303  691  453
         5  305  259  457 1076 4004

Overall Statistics

               Accuracy : 0.5973
                 95% CI : (0.5876, 0.6069)
    No Information Rate : 0.4713
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3794

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.7007   0.1027   0.1234   0.3460   0.8496
Specificity            0.9303   0.9826   0.9726   0.8834   0.6034
Pos Pred Value         0.6461   0.3207   0.3369   0.4255   0.6563
Neg Pred Value         0.9448   0.9320   0.9078   0.8441   0.8182
Prevalence             0.1537   0.0740   0.1013   0.1997   0.4713
Detection Rate         0.1077   0.0076   0.0125   0.0691   0.4004
Detection Prevalence   0.1667   0.0237   0.0371   0.1624   0.6101
Balanced Accuracy      0.8155   0.5427   0.5480   0.6147   0.7265
```

Class Model 2:

```
Call:
 randomForest(formula = stars.y ~ score + average_stars + stars.x +      review_count.x +
fans + review_count.y + useful.x + funny.x +      cool.x + latitude + longitude + is_open
+ compliment_plain,      data = train, type = "class", ntree = 80, importance = T)
               Type of random forest: classification
                     Number of trees: 80
No. of variables tried at each split: 3

        OOB estimate of  error rate: 39.04%
Confusion matrix:
      1    2    3     4      5 class.error
1 32878 2186  999  1251   3931   0.2028610
2  7813 3296 1796  2898   5447   0.8448941
3  3349 1644 3663  8014  10180   0.8635754
4  1862 1067 3309 18740  31118   0.6659298
5  2507  932 1854 13213 105931   0.1487178
```

Confusion Matrix of Class Model 2:

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4    5
         1 1254  293  117   58   96
         2   62  108   61   34   29
         3   36   67  129  101   53
         4   20   83  295  668  477
         5  165  189  411 1136 4058

Overall Statistics

               Accuracy : 0.6217
                 95% CI : (0.6121, 0.6312)
    No Information Rate : 0.4713
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4218

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            0.8159   0.1459   0.1273   0.3345   0.8610
Specificity            0.9334   0.9799   0.9714   0.8907   0.6404
Pos Pred Value         0.6898   0.3673   0.3342   0.4329   0.6810
Neg Pred Value         0.9654   0.9349   0.9081   0.8429   0.8379
Prevalence             0.1537   0.0740   0.1013   0.1997   0.4713
Detection Rate         0.1254   0.0108   0.0129   0.0668   0.4058
Detection Prevalence   0.1818   0.0294   0.0386   0.1543   0.5959
Balanced Accuracy      0.8746   0.5629   0.5494   0.6126   0.7507
```