Analysis of Method – Report

The DS Methodology that I decided to follow was the CRISP-DM Method because it lent itself perfectly to my approach. I applied it by structuring my approach based on the iterative life-cycle process.

1.) Understand Business Needs: Use data to predict review in terms of number of stars that user I gives business j
2.) Data understanding: EDA, understanding how data can be used to answer the question.
3.) Data Preparation: Different Models require the data to be preprocessed differently.
4.) Modelling: Create the model
5.) Evaluation: Run Model on test data and see how it fares
6.) Deployment: This analysis report falls under this category.

The challenge I found most difficult was picking the combination of variables to create smaller sub datasets. I did this my visualizing the correlation and manually putting them into groups. Alternatively, a feature importance method could have been used.

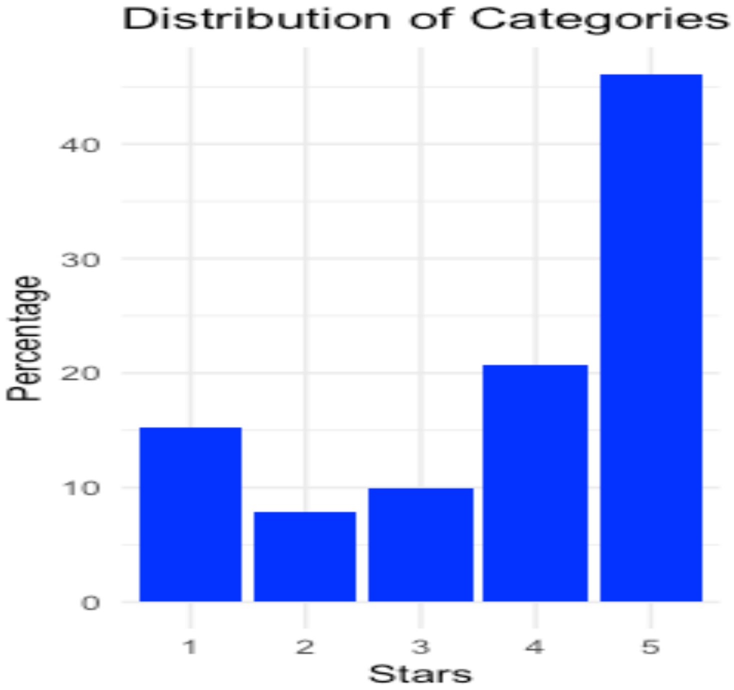
Introduction

The aim of the model is to predict the number of stars a business will receive in a review. Since it can only receive a discrete number between 1-5 inclusive, I've treated this as a classification problem rather than a regression problem and hence changed variable storage type to a factor variable.

We have been given 5 datasets out of which I chose to work with user_small, review_small and business. json. I used user_id and business id to merge them all to create a single dataset called Merged_data1.

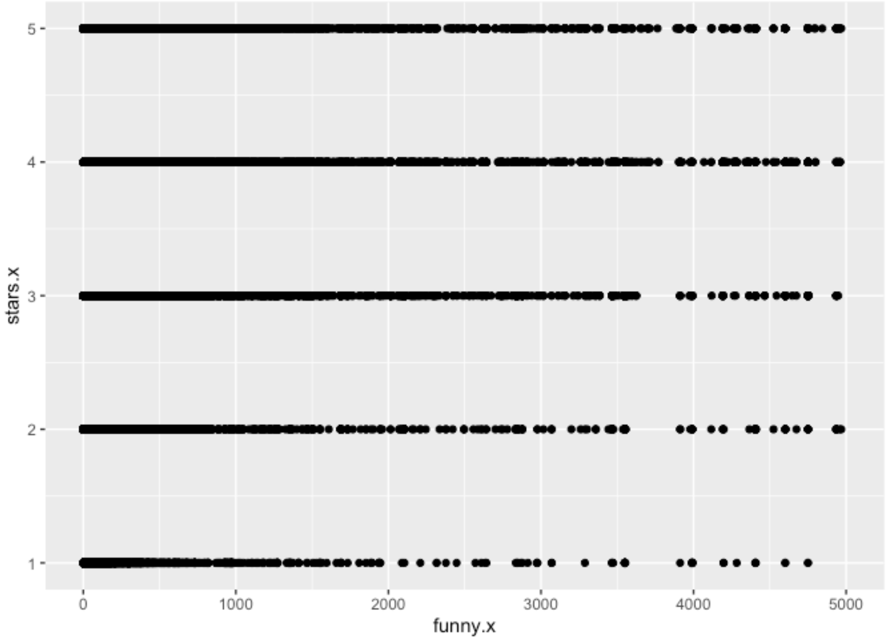
Data Preprocessing and EDA

Merged_data1 has 87 columns, 279878 observation and a mix of different types of variables. Since the brief is to predict stars.x, it is essential to first understand its distribution.



Plotting its distribution, we observe an imbalance in classes which can prove to be problematic. To account for this, I oversampled underrepresented classes to the same level as the largest class.

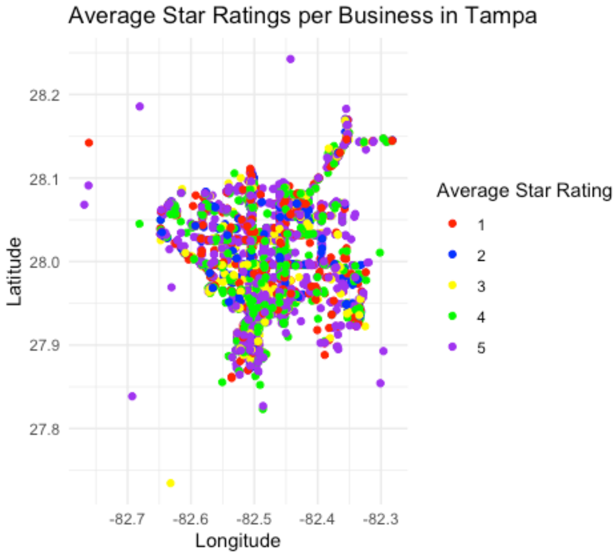
To understand the relationship between different variables and stars.x I used the ggplot2 package to visualize any patterns.



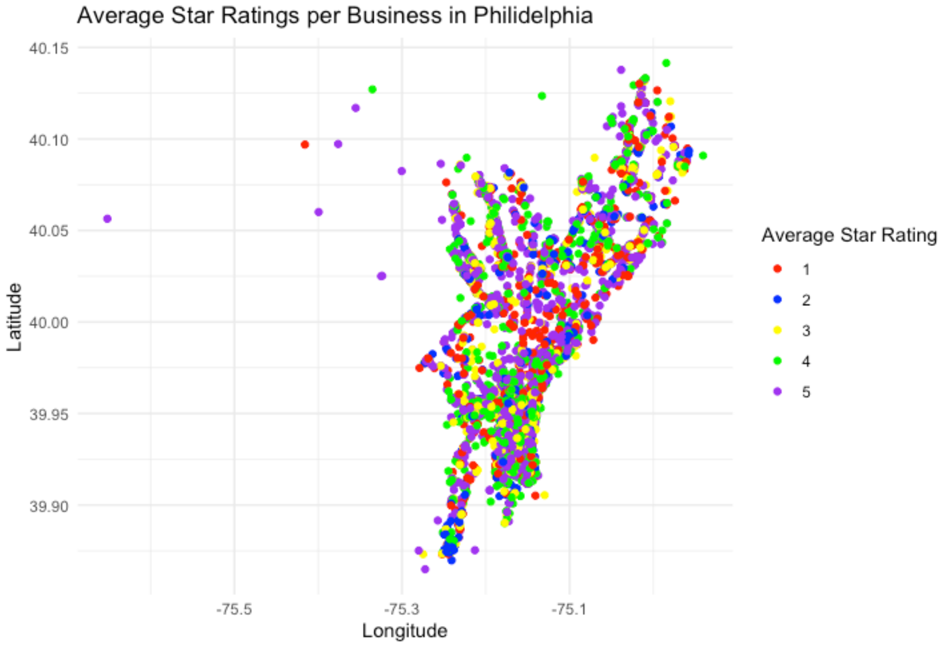
We observe here as funny.x increases, it is more probable that the business has a higher rating.

Variables which didn't showcase a strong correlation I discarded from the data hence creating smaller dataset "subdata" with lower dimensionality and noise.

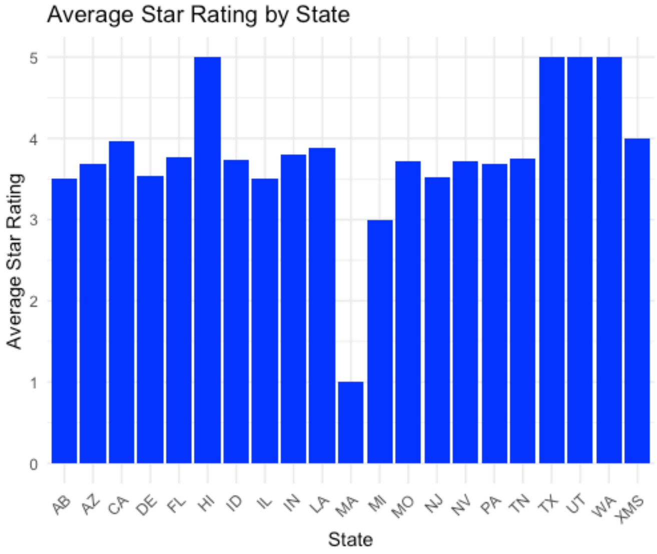
To investigate the impact of location on rating I use the latitude and longitude variables. My hypothesis was that if the business was in a better neighborhood, it would have a higher rating and hence clustering of classes would be observed.



Tampa didn't have any clear indications of clustering of rating of the same class whereas Philadelphia did as seen below which could be because Philadelphia houses more tourist who are more likely to leave reviews.



As we can see, there seems to be a concentration of 1 star rating in the Centre but higher ratings in the suburbs which are the nicer areas.



Here we can see the difference in average ratings by state.

MODELS

The first model I decided to run was a simple classifier that always predicts 5 as that's the highest class in the data. The accuracy of this model was 46%. Which I will treat as baseline

The next model I decided to run was a KNN model. K-Nearest Neighbors (KNN) is a simple, non-parametric classification algorithm that assigns a class to a new data point based on the majority class among its k-nearest neighbors in the training set. It calculates distances between data points, considering the closest `k` neighbors to determine the class. My dataset included only stars.x, latitude and longitude and the model had an accuracy of 45%. I believe it didn't fare too well because as seen in the Tampa and Philadelphia example there weren't prominent patterns to learn from.

Another model known to do well as a classifier is a neural network model. Neural networks involve layers of interconnected nodes (neurons) that process input data through weights and activation functions. They learn complex patterns by adjusting these weights during training, through backpropagation. The output layer classifies input by assigning probabilities to each class, based on learned patterns. This model yields an accuracy of 57%. While this is better than the KNN model, a neural network setup should be significantly more powerful. I believe it didn't perform well because it is prone to overfitting as it hones an exceptional learning rate and may have mistaken noise for complex patterns.

Random Forest Model Analysis – Main Model

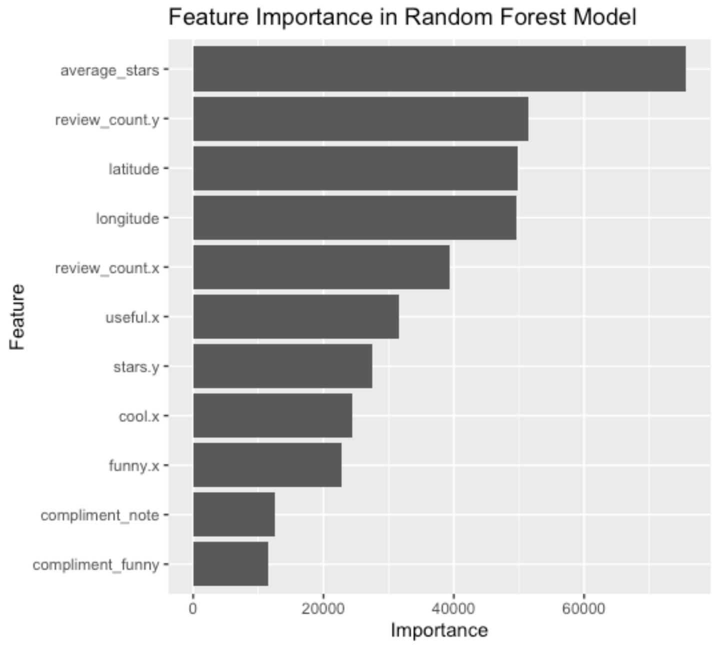
Random Forest is a robust ensemble learning technique ideal for classifying complex datasets with numerous predictors like ours. By constructing multiple decision trees, each on a distinct bootstrap sample from the balanced dataset `balanced_data`, it captures a diverse range of data aspects. Each tree in the ensemble considers a random subset of features for splitting nodes, preventing any single predictor from dominating the decision process. This strategy enhances the model's robustness and prevents overfitting, which can be a significant risk.

The method does not rely on a singular loss function during training. Instead, it uses measures like Gini impurity and entropy to assess the quality of splits within individual trees—where $Gini = 1 - \sum (p_i)^2$ and $Entropy = - \sum p_i \log_2(p_i)$, with p_i representing the proportion of samples within the i^{th} class. These criteria allow the model to evaluate and minimize misclassification risk and data disorder, respectively. Gini impurity quantifies the purity of a node, indicating the likelihood of a new sample being incorrectly classified if randomly assigned according to the distribution of samples in the node. Entropy is used to measure the level of disorder or uncertainty in the data, with higher entropy reflecting more heterogeneity in the class distribution within a node.

Feature importance emerges naturally from the Random Forest algorithm, with more predictive features resulting in greater impurity reduction and, consequently, a higher importance score. This is quantified post-training through metrics such as Mean Decrease in Impurity and Mean Decrease in Accuracy.

The method's structure inherently immune against missing data and outliers because it can address gaps in data through surrogate splits which is utilizing correlated features when primary splits fails, or by imputing missing values. Since Random Forest is an ensemble of decision trees, and decision trees are non-parametric, they don't assume a particular distribution for the data. Therefore, they are less affected by outliers than parametric methods that make specific assumptions about distribution, like linear regression.
To enhance the model's performance and robustness, k-fold cross-validation was employed.

The model achieved an accuracy of 94% in the test data and had a sensitivity of 0.97, 0.99, 0.98, 0.90, and 0.84 for each class 1-5 respectively.



Limitations

While Random Forest is a robust and powerful model, it's important to note its limitations, such as the lack of transparency in how individual trees make decisions (the "black box" nature) and the computational complexity with very large datasets.

Conclusion

Given more time and computational resources such as GPU and CPU, there are many enhancements I would have liked to have made to the process to utilize the full dataset. Sentiment analysis could have been conducted on the text of the reviews, feature selection could have been done on the entirety of Merged_data1 and through clustering selection, one impactful variable from each cluster could be used to create multiple sub datasets to test which ones work the best with the different models. Additionally, the dataframe attributes could have been better utilised to compare and predict stars based on the similarity of the businesses as well through clustering.Lastly, an intricate neural network could have been created, and an ensemble of all of these models to create a supremely accurate predictor.

References

1. Grolemund, G. & Wickham, H. 2016. R for Data Science: Import, Tidy, Transform Visualise, and Model Data.
2. Hastie, T.; Tibshirani, R.; Friedman, J. H.; & Friedman, J. H. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Vol. 2). New York: Springer
3. James, G.; Witten, D.; Hastie, T.; & Tibshirani, R. 2021 (2nd Ed.). An Introduction to Statistical Learning with Applications in R. Springer.