Camille Gengania (305520126)
Stacy Deng (305537216)
Katia Uribe (605619705)

Statistics 101C Final Report

1) Introduction

For this project, we are using the Yelp Challenge dataset titled Data_Final provided in class for training and testing our prediction models. The dataset consists of information from 11,204 unique businesses based in 22 cities across California and 3530 unique users. The Data_Final file contains 18 variables regarding businesses and users including business/user ID, average stars (rating), review/user ratings (useful, funny, cool), and in the case of users, variables such as Elite Squad status and user fan count.

In this report, we are training and evaluating models on the dataset to predict Yelp reviewers' Elite status based on the variables "User_Review_count," "User_Useful_count," "User_Funny_count," "User_Cool_count," and "User_Fans." We used four supervised learning algorithms: Random Forest, LDA and QDA, logistic regression, and K-Nearest Neighbors to build our prediction models. In Section 2 of the paper, we describe the steps used to pre-process the data from Yelp. In Section 3, we provide the details on the different models we test out in R to predict Elite status. In Section 4, we analyze the performance of these models and discuss our final conclusions.

2) Pre-processing Step

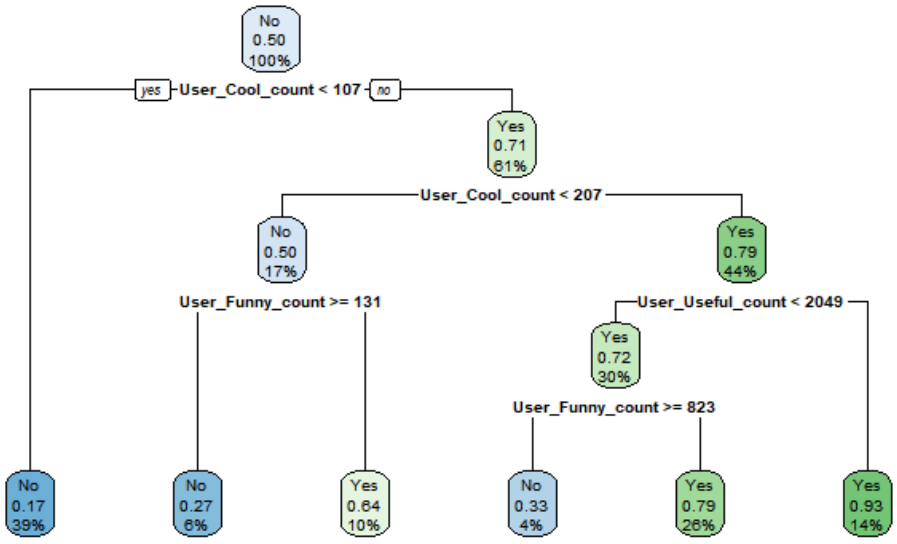
Using R, we first fix errors in our dataset such as correcting "20,20" to "2020" under the Elite variable and "Santa Barbra" to "Santa Barbara" under the city variable. We also create and factor a new column called "Elite_status" with the levels "Yes" and "No" to indicate whether or not a reviewer has elite status on Yelp. Because we have 42418 observations under "Yes" and 11427 under "No," we have to balance out the dataset so that each has 11427. We will then split the dataset at 80:20 for our training set and testing set.

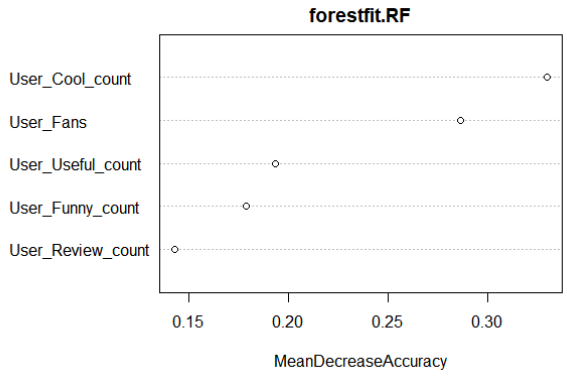
3) Experimental Setup

To train our prediction models, we used the following four training models:

3.1) Random Forest

The first training model we tried is Random Forest. Using the rpart.plot function, we plotted a decision tree graph consisting of the nodes for User_Cool_count, User_Funny_count, User_Useful_count, and User_Review_count.





We also created variance importance plots using the varImpPlot function for both permutation importance and Gni importance with the same variables and additionally, User_Fans. From these graphs, we can determine that User_Cool_count is the most important variable for our Random Forest model since it is the root node in the tree plot and the top variable in the variable importance plots.

With our Random Forest model, we get 2,169 true positive predictions, 2,177 true negative, 89 false positive, and 108 false negative, giving us an accuracy score of 0.9569.

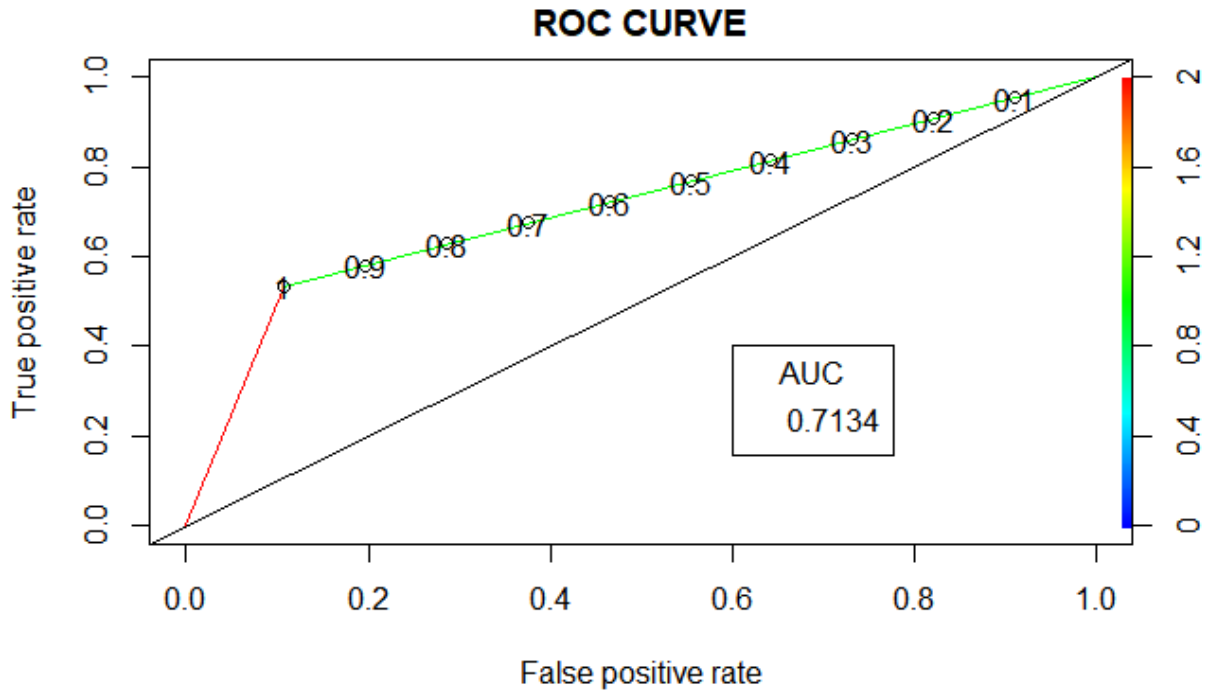
3.2) LDA and QDA

The next models we tried are Linear and Quadratic Discriminant Analysis (LDA and QDA). Once again, we used the variables User_Cool_count, User_Funny_count, User_Useful_count, User_Review_count, and User_Fans to predict Elite_status. First, we used LDA, which assumes each class has the same covariance matrix. Our results yielded an accuracy of 0.6532, a True Positive rate of 0.4167, and a test error of 0.3468 for this model. For our QDA model which assumes different covariance matrices for each class, we get an accuracy of 0.5931, a True Positive rate of 0.2656, and a test error of 0.4069.

3.3) Logistic Regression

We ran our third model using logistic regression, following the binomial distribution. We used Elite_status as our response variable, with User_Cool_count, User_Funny_count, User_Useful_count, User_Review_count, and User_fans as our predictors. The results from our regression shows that the accuracy of our model is 0.7149, with a true positive rate of 0.5344 and a test error of 0.2851.

Plotting an AUC-ROC curve, we can see that the AUC is 0.7134. The closer AUC is to 1, the better the measure of separability, showing that our logistic regression model is able to distinguish between Elite and non-Elite members relatively well.



3.4) K-Nearest Neighbors (KNN)

Our next model training model uses K-Nearest Neighbors. Using K-fold cross validation, we discovered that the best value of K for our model was 5. Using KNN, the confusion matrix consists of 1965 true positive observations, 2065 true negatives, 335 false positives, and 206 false negatives. The accuracy of KNN is 0.8816, with a true positive rate of 0.8543, and a test error of 0.1184.

3.5) Support Vector Machine (SVM)

The final model we created using the Support Vector Machine. The type we used is C-classification with the radial/Gaussian kernel in which only nearby training observations affect the test observation predictions. The confusion matrix consists of 1832 true positive observations, 1727 true negative, 453 false positive, and 558 false negative. Here, we get an accuracy score of 0.7786, a True Positive rate of 0.7922, and a test error of 0.2214.

4) Results and Analysis

Based on our graphs, we can conclude that the Random Forest achieved the highest accuracy at 0.9569, followed by the K-Nearest Neighbors at 0.8816.

Without looking at accuracy, we can see that by looking at the diagnostic graphs for our logistic regression, we see that the residuals all seem to be following the line, and the scale location also follows a parabolic shape, so logistic regression may not be the best choice of a model.