

Understanding the Factors that Raises Restaurant Revenue

Brett Loy, Darling Judah Hsu, Hubert Nguyen, Jared Duong,

March 2024

Contents

1 Bacgkround	2
2 Introduction	2
3 Data Description	2
3.1 The Data	2
3.2 Data Preprocessing	2
4 Exploratory Data Analysis	3
5 Methodology	4
5.1 Linear Regression	4
5.2 Resampling Techniques	4
5.3 Classification Techniques	5
6 Main Results	5
7 Discussion and Outlook	7
8 Conclusion	7
9 Code	8
9.1	8

1 Bacgkround

As a critical sector in the United States, the restaurant industry stands as the second-largest private employer, with approximately 15.5 million individuals, roughly 10% of the entire workforce, as reported by the National Restaurant Association.

When faced with the onslaught of the COVID-19 pandemic, the industry saw an estimated 90,000 restaurants shut their doors, with an anticipated quarter of these establishments shutting down permanently. In the face of such adversity and amidst an inflationary economy, the road to rebuilding is paved with challenges, including potential stunted growth and recovery setbacks.

Thus, we've chosen to create and analyze a variety of models in order to predict revenue and overall success of a restaurant by various metrics relating to their marketing, cuisine type, reviews, etc.

2 Introduction

Restaurant revenue is influenced by a myriad of factors, some including cuisine type, menu offerings, pricing strategies, customer service quality, marketing efforts, and external factors such as economic conditions and seasonal variations.

However, the advent of data analytics and machine learning has revolutionized this process, offering more sophisticated and accurate forecasting techniques. In particular, we plan to use regression, resampling, and classification techniques to practice class-given methods on high-variation data, while also aiming to determine the factors which most indicate restaurant success.

Some of the main problems that we are trying to address is:

- What crucial models and methods can be used to predict restaurant revenue?
- What factors most significantly predict a restaurant's revenue?

3 Data Description

3.1 The Data

The dataset we are planning to use is the following [Restaurants Revenue Prediction Dataset](#) created by MrSimple07 on Kaggle. This simulated dataset offers a diverse range of features designed to mimic the factors influencing restaurant performance, including: number of customers, average menu price, expenditure on marketing spending, cuisine type, average spending per customer, promotions, and number of reviews. The dependent variable in this problem is monthly revenue which is based on the independent variables above. This is the original datatset.

3.2 Data Preprocessing

In terms of preprocessing, because the data was simulated, the dataset was very clean (no empty values, consistent data, and moderate dimensionality meant no need for PCA). So, our only concern in terms of pre-processing lay within our dependent variable Monthly_Revenue. Success for a restaurant can be measured in

terms of profit (in which case `Monthly_Revenue` isn't particularly explanatory considering possibly high costs); though, `Monthly_Revenue` does generally show how big a business is. Thus, in our analyses, we regress on `Monthly_Revenue` and a new variable `Profit_Stat` (which is `Monthly_Revenue` scaled by `Menu_Price` since `Menu_Prices` are often representative of costs).

Furthermore, we create a new categorical variable called "Success" for classification purposes. By looking at a box plot, we categorize the restaurants such that the "Failure" label is given to restaurants with the bottom 25% of `Profit_Stat`, "Moderate Failure" to the next 25%, "Moderate Success" to the next 25% and "Success" to the upper 25%.

4 Exploratory Data Analysis

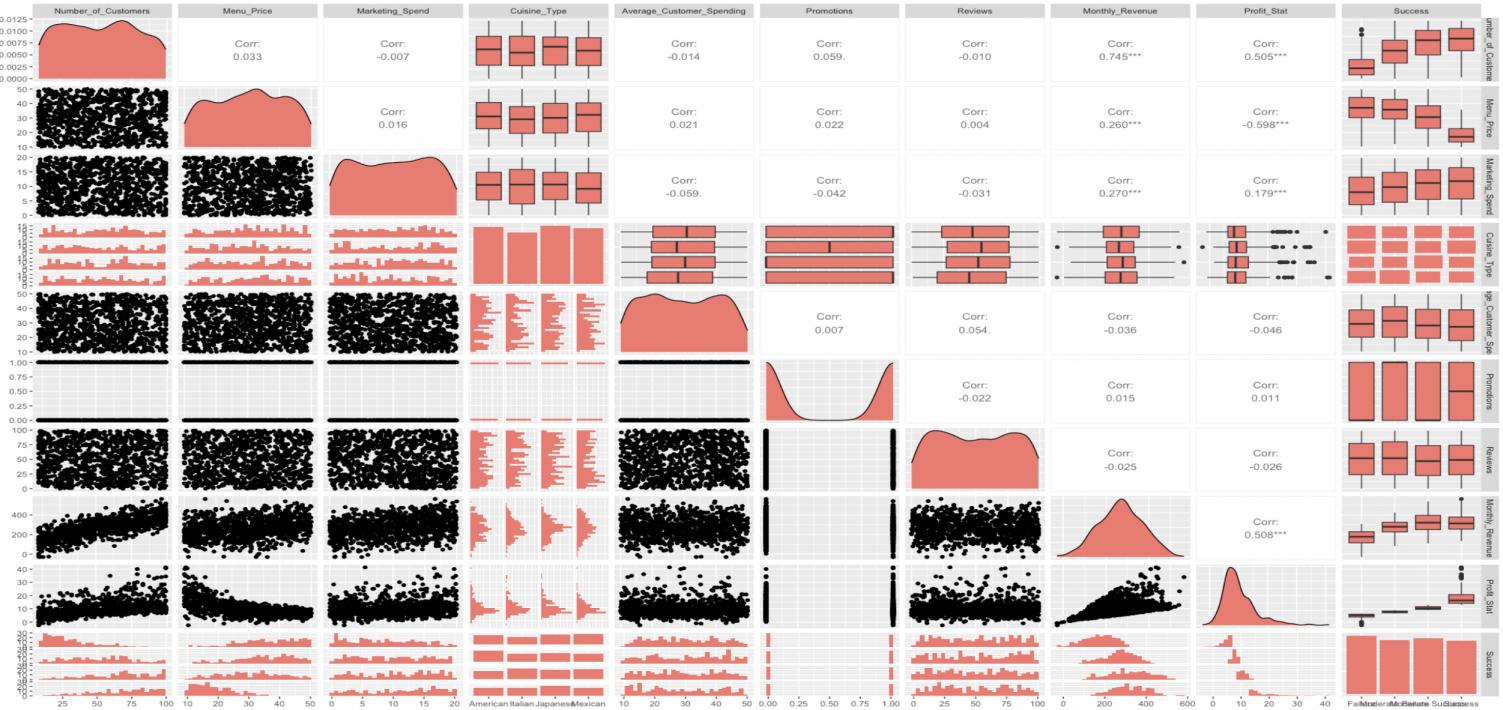


Figure 1: Pairwise correlation plots and graphs for all variables, showing little correlation

Immediately, the pairplots show that many of the variables involved may be unnecessary. In particular, the only variables that have correlations, with `Monthly_Revenue` or `Profit_Stat`, of magnitudes greater than 0.1 are `Marketing_Spend` and `Number_of_Customers` (also `Menu_Price` and `Monthly_Revenue` on `Profit_Stat` but we consider these variables blatant in predicting both dependent variables). In particular, the number of customers stands out with correlation of 0.505 and 0.745 to `Profit_Stat` and `Monthly_Revenue` respectively.

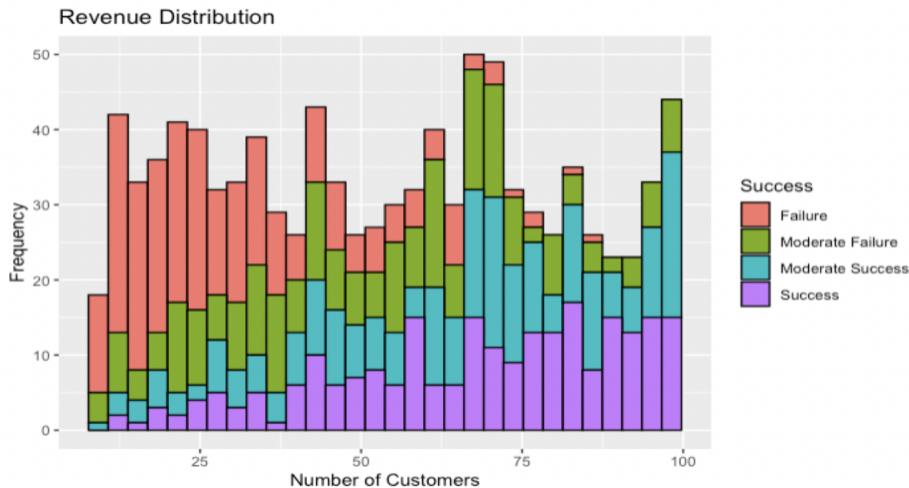


Figure 2: Histogram showing the distribution of the number of customers in restaurants, stratified by success levels

So, looking at a graph further, we see that success tends to be distributed much more across restaurants with a higher number of customers. Our primary focuses will be the number of customers and marketing expenditure.

However, it's also important to note that some variables might have combined effects on the dependent variables. So although we will take into account linear combinations of other variables, it seems rather unlikely considering the low correlation between all the predictors.

Furthermore, upon performing a calculation of the variance inflation factor (VIF), which assesses the dependence of the predictors to the response we yielded values within 1-1.018. For reference, VIFs can range from $(1, \infty)$ and 1 indicates no multicollinearity.

So although we will test combinations of variables, we expect to focus much more on the Number_of_Customers and Marketing_Spend variables

5 Methodology

5.1 Linear Regression

This will serve as a baseline for modeling the linear relationship between various predictors (e.g., menu prices, customer counts) and restaurant revenue. We will do model selection by both-ways AIC. We assume that relationships between some predictors and revenue are linear and additive, independence.

In particular, because our data is very clean and high-variation'd, we don't expect overfitting. Still, there are few data points and because of the low correlation we decided to try simple models first. Regardless, this leads us to more biased models later, including QDA.

5.2 Resampling Techniques

By employing cross-validation and bootstrapping, we aim to rigorously assess the performance of our statistical models, providing insights into their accuracy and stability across different subsets of the dataset. We assume that

these resampling techniques will offer a reliable estimate of model performance, acknowledging that such methods help mitigate overfitting and provide a more generalized understanding of the model's predictive capability.

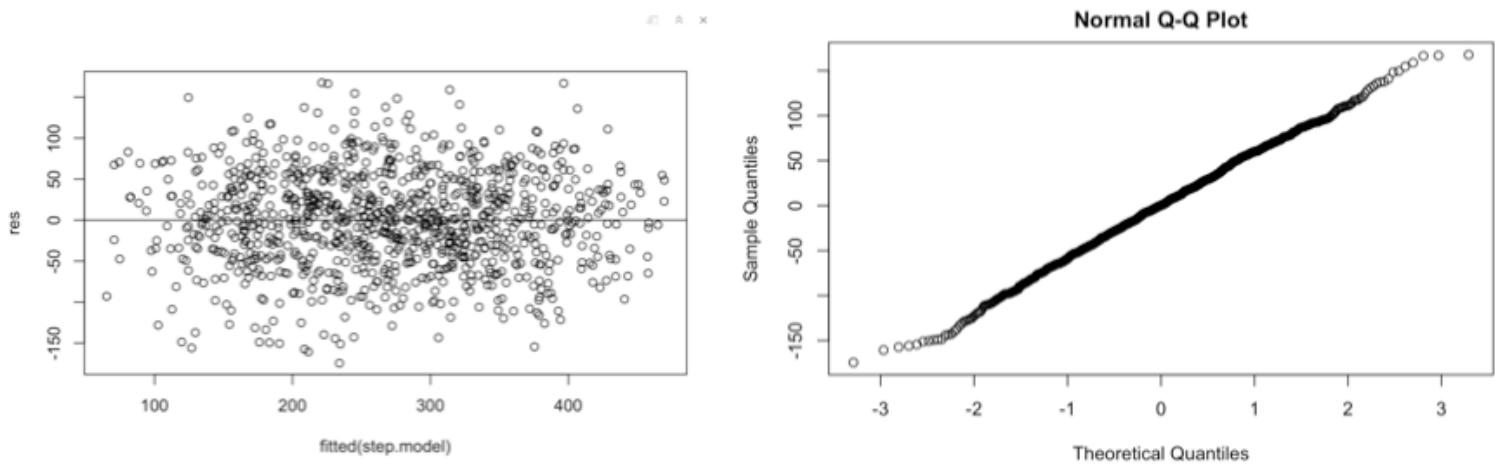
5.3 Classification Techniques

Including logistic regression, LDA (Linear Discriminant Analysis), QDA (Quadratic Discriminant Analysis), and SVM (Support Vector Machine), these methods will help identify variables that are more likely to yield higher restaurant revenues, thus informing targeted marketing strategies.

We assume that the restaurant data can be linearly or quadratically separated into distinct categories related to revenue generation. Linear models like logistic regression and LDA are effective when relationships between variables and revenue categories are linear, whereas QDA and SVM can accommodate more complex, nonlinear relationships, offering a nuanced understanding of how different factors might influence restaurant earnings across various market segments.

6 Main Results

We initially fit the simplest model, multiple regression, in order to get a baseline for the prediction task. The model assumptions of independence and normality of errors were not violated (random scatter in residuals, linear QQ-plot), and we obtained an r-squared of about .7, indicating a positive relationship, albeit not an extremely strong one. This lead us to explore further methods:



Figures 3 and Figure 4: Residual and QQ-plot for the data indicating equal variance and normality

To perform LDA and QDA the first step was to assess what variable we want to classify. This came as somewhat a struggle since we originally tested for Cuisine_Type classification which the data did not predict well. However, we encountered that the more relevant classification would need to come from Monthly_Revenue. To achieve our desired outcome, we created a separate column named Monthly_Revenue_Class. This enabled us to turn Monthly_Revenue into a categorical variable and able to be classified. Monthly_Revenue_Class was separated into 3 categories: struggling, solid, and exceeding which were split up by less than first quadrant, between first and third quadrant, and greater than the third quadrant respectively. We then ran LDA and QDA models using training data and predicted values using the testing data. Here we can visualize the decision boundary of the

LDA models with the two most significant variables because they would have the most effect on the boundary: the number of customers and the expenditures on marketing.

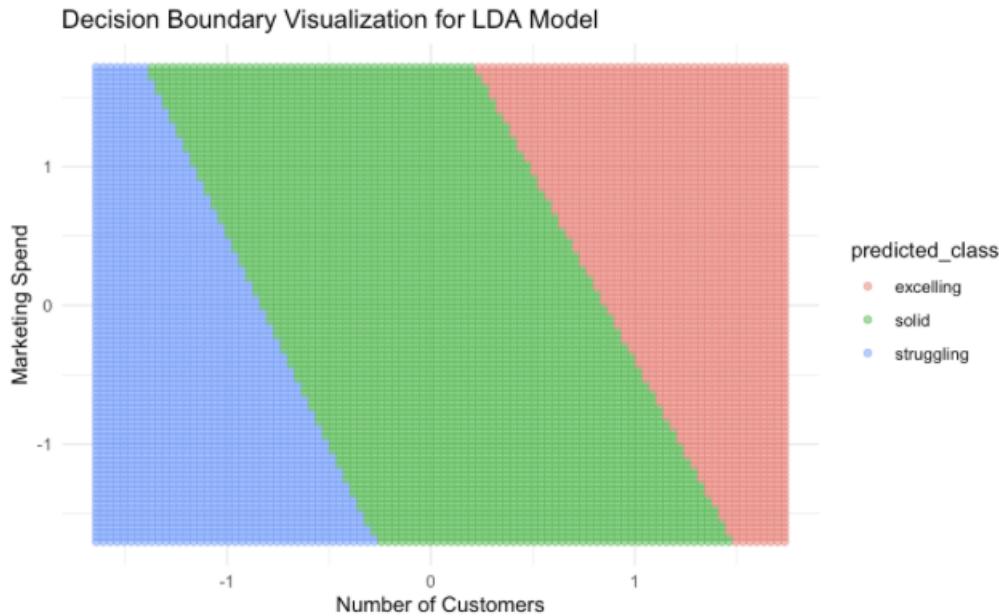


Figure 5: Graph for linear discriminant boundary line where the lines between success clearly favors higher number of customers and marketing expenditures

The accuracy of the LDA model was 41.3% while the accuracy of the QDA model was 94.6%. This intuitively makes sense because LDA is a simpler model and we had many predictor variables and complex data. On the other hand, we could assume that QDA would classify Monthly_Revenue_Class better since it offers more flexibility and higher accuracy on more complex datasets.

We also utilized feature importance for support vector machine (SVM), visualizing feature importance for monthly revenue prediction, with the feature importance measured by the increase in root mean square error (RMSE) when each feature's values are permuted. The chart indicates that the number of customers has the most significant impact on the model's prediction error for revenue, suggesting it's the most influential variable. This is followed by marketing spend, menu price, and cuisine type, which also have notable importance. Features such as reviews, promotions, and average customer spending have less impact on RMSE, indicating they may be less critical in predicting monthly revenue according to this model.

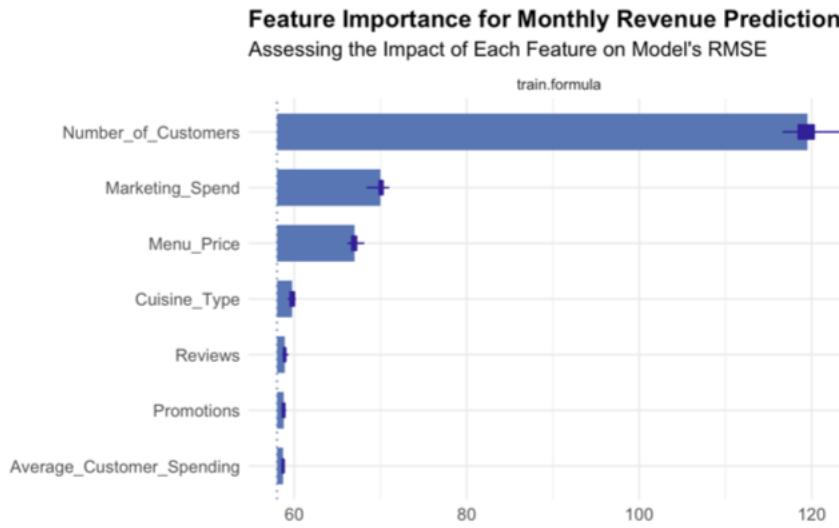


Figure 6: Root mean squared error (RMSE) loss after permutations shows the high impact of Number_of_Customers

We found that the main predictors that were most significant were Number_of_Customers, Marketing_Spend, and Menu_Price.

7 Discussion and Outlook

Regardless, there are still many things to consider. Among them, our primary concerns, limitations, and areas to grow in, are the following: measures of success, real data, locations, and more factors. One other method to look into, if we had more time, was ensembling.

Among our primary concerns, the simulated nature of our data nullifies a lot of possible real-world applications. We would've like to perform inference, but this does not make sense in a simulated setting. The best we could do is try to accurately develop models for real-world data in the future. Furthermore, monthly revenue may not be an accurate measure of success, nor our scaling of monthly revenue by menu price. Ideally, we would obtain profit data. Aspects like number of customers directly influence how much a restaurant makes intuitively, but more ad spend could be a result of a successful business, and not the other way around. Adding onto that, we'd prefer to have more factors to identify such as location (eg. it might be easier to make money in a big town), average income of residents, or proximity to schools. We'd also like to consider ensembling between models, especially between our classification and regression models. In particular, if we had a bit more data, we could even regress other helpful variables to then use for further classification.

8 Conclusion

By our findings, Number of Customers, Marketing Spend, and Menu Price were the most significant factors in a restaurant's revenue, and business owners may want to record this information to see where they can improve their business. Using this data, both existing and prospective owners can use QDA to better predict and understand aspects of a successful restaurant.

9 Code

9.1

```

7 ~ ### ALL CODE COMBINED
8
9 Data Analysis on Restaurant Revenue Data
10
11 Divided into four sections (Jared's code, Judah's code, Hubert's code, Brett's code):
12 1. Multiple Regression and some exploration by cuisine type
13 2. EDA, decision Trees
14 3. CV, bootstrap, model selection
15 4. LDA, QDA, EDA (multicollinearity)
16
17

18 Jared's Code (Multiple Regression and some exploration by cuisine type):
19 ~### Preliminary
20 ~`~`{r}
21 # Load necessary libraries
22 library(tidyverse)
23 library(corrplot)
24 library(caret)
25 library(MASS)
26 library(leaps)
27 library(ISLR)
28 revenue <- read.csv("~/Downloads/Restaurant_revenue (1).csv")
29 ~`~`{r}
30 ~`~`{r}
31 head(revenue)
32 unique(revenue[4])
33 ~`~`{r}
34 ~`~`{r}
35 # Data grouped by cuisine:
36 Japanese <- revenue %>%
37   filter(Cuisine_Type == 'Japanese')
38 Italian <- revenue %>%
39   filter(Cuisine_Type == 'Italian')
40 American <- revenue %>%
41   filter(Cuisine_Type == 'American')
42 Mexican <- revenue %>%
43   filter(Cuisine_Type == 'Mexican')
44 ~`~`{r}
45 ~`~`{r}
46 # Correlation matrix
47 corrplot(corr(revenue[, -4]), tl.col="black")
48 # Mean revenue by cuisine type
49 revenue %>%
50   group_by(Cuisine_Type) %>%
51   summarise(mean(Monthly_Revenue), mean(Number_of_Customers), mean(Menu_Price), mean(Marketing_Spend), mean(Average_Customer_Spending), mean(Promotions), mean(Reviews))
52 # Can add corrplot per cuisine type
53 ~`~`{r}

```

```

55 # ## Multiple regression with stepwise selection (By AIC)
56 ````{r}
57 # Fit a linear model where Monthly_Revenue is predicted by all other variables in the 'revenue' dataset
58 full <- lm(Monthly_Revenue ~ ., data = revenue)
59
60 # Perform stepwise model selection to identify the optimal model based on AIC
61 # 'direction = "both"' means the algorithm can both add and remove predictors to find the best model
62 # 'trace = FALSE' suppresses the printing of information at each step, making the output cleaner
63 step.model <- stepAIC(full, direction = "both", trace = FALSE)
64 ````

65
66 By distinct cuisine types:
67 ````{r}
68 # Fit a linear model predicting Monthly_Revenue from all but the 4th column in 'Japanese'.
69 fullj <- lm(Monthly_Revenue ~ ., data = Japanese[-4])
70
71 # Use stepwise regression to optimize model predictors, allowing both addition and removal.
72 step.model <- stepAIC(fullj, direction = "both", trace = FALSE)
73
74 # Output a summary of the optimized model, including coefficients and diagnostics.
75 summary(step.model)
76 ````

77
78 ````{r}
79 # Fit a linear model for Monthly_Revenue using all variables except the 4th in the 'Mexican' dataset
80 fulm <- lm(Monthly_Revenue ~ ., data = Mexican[-4])
81 # Perform stepwise regression to optimize the model, allowing both addition and removal of variables
82 step.model <- stepAIC(fulm, direction = "both", trace = FALSE)
83 # Print a summary of the optimized model to review model performance and variable significance
84 summary(step.model)
85 ````

86
87 ````{r}
88 # Fit a linear model for Monthly_Revenue using all variables except the 4th in the 'Italian' dataset
89 fulli <- lm(Monthly_Revenue ~ ., data = Italian[-4])
90 # Perform stepwise regression to optimize the model, considering both adding and removing predictors
91 step.model <- stepAIC(fulli, direction = "both", trace = FALSE)
92 # Output a summary of the stepwise optimized model for evaluation
93 summary(step.model)
94 ````

95
96 ````{r}
97 # Fit a linear model predicting Monthly_Revenue from all variables except the 4th in 'American' dataset
98 fulla <- lm(Monthly_Revenue ~ ., data = American[-4])
99 # Use stepwise regression for model selection, with bidirectional option for variable selection
100 step.model <- stepAIC(fulla, direction = "both", trace = FALSE)
101 # Display a summary of the final model, including coefficients, significance, and fit metrics
102 summary(step.model)
103 ````

104 Judah's code (EDA,PCA, Decision Trees):
105
106
107 ````{r setup, include=FALSE}
108 # Initial Libraries
109 library(tidyverse)
110 library(ggplot2)
111 library(caret)
112 library(randomForest)
113 revenue <- read.csv("~/Downloads/Restaurant_revenue (1).csv")
114
115 # I'm creating a new column to re-evalue success by revenue per dollar on average menu price
116 revenue$Profit_Stat = revenue$Monthly_Revenue / revenue$Menu_Price
117 revenue <- revenue %>% mutate(Success = case_when(
118   Profit_Stat > 13 ~ "Success",
119   Profit_Stat <= 13 & Profit_Stat > 9 ~ "Moderate Success",
120   Profit_Stat <= 9 & Profit_Stat > 6.5 ~ "Moderate Failure",
121   TRUE ~ "Failure"
122 ))
123
124 revenue %>% head()
125 ````
```

```

126 Notes:
127 Variables are listed below
128 - number of customers: count of visiting customers
129 - menu price: average menu prices at restaurant
130 - marketing spend: expenditure on marketing activities (scale isn't mentioned... so a store spent $3.475052?)
131 - cuisine type: type of cuisine offered
132 - average customer spending: average spending per customer
133 - promotions: binary indicator whether or not promotions were conducted
134 - reviews: number of reviews received by restaurant
135 - monthly revenue: just that
136
137 -A big cause for concern here is we don't know how much profit each restaurant is making because we don't know how much the ingredients for a restaurant costs. We'll have to assume that menu price would be correlated with price of ingredients and labor. So we expect higher menu price to be associated with "fancier" restaurant. So, revenue should likely be scaled by menu price otherwise, we'll just get all the "fancy" restaurants having high revenue but it's not an indicator of profit- how successful they actually are.
138
139 -For EDA, the relationships I want to check: affect of marketing on reviews, marketing on profit, marketing on number of customers, number of customers on menu price, (the thing is ig a scatterplot correlation kinda says this so we'll think in more dimensions)
140 - heatmaps
141 - box plots
142 - histograms
143 - scatterplots (change color / size)
144

145 ````{r}
146 # Visualize the distribution of the number of customers, colored by success status, to explore its impact on success.
147 ggplot(data = revenue, aes(x = Number_of_Customers, fill=Success)) +
148   geom_histogram(binwidth = 5, color = "black") +
149   labs(title = "Revenue Distribution",
150       x = "Number of Customers",
151       y = "Frequency")
152
153 # Investigate menu price distribution across different cuisines to explore pricing strategies and their correlation to cuisine types.
154 ggplot(data = revenue, aes(x = Menu_Price, fill=Cuisine_Type)) +
155   geom_histogram(binwidth = 2, color = "black") +
156   labs(title = "Menu Price Distribution",
157       x = "Menu Price",
158       y = "Frequency")
159
160 # Plot reviews against profit status by cuisine type to understand the relationship between customer feedback and profitability.
161 ggplot(data = revenue, aes(x=Reviews, y = Profit_Status, color=Cuisine_Type )) +
162   geom_point()
163
164 # Display the spread and distribution of profit statistics across the dataset.
165 ggplot(data=revenue, aes(x=Profit_Status)) +
166   geom_boxplot()
167
168 # Examine the relationship between marketing spend and monthly revenue by success, highlighting the effectiveness of marketing strategies.
169 revenue %>% ggplot(aes(x=Marketing_Spend, y=Monthly_Revenue, color= Success, alpha=0.5)) +
170   geom_point() +
171   geom_smooth(method='lm')
172
173 # Analyze the correlation between menu price and monthly revenue, suggesting higher-priced menus might lead to higher revenue.
174 revenue %>% ggplot(aes(x=Menu_Price, y=Monthly_Revenue)) +
175   geom_point(alpha=0.5) +
176   geom_smooth(method='lm')
177

```

```

177 # Explore how marketing spend influences customer volume, differentiated by success, to assess marketing effectiveness.
178 revenue %>% ggplot(aes(x=Marketing_Spend, y=Number_of_Customers, color= Success, alpha=0.5)) +
179   geom_point() +
180   geom_smooth(method='lm')
182
183 # Evaluate the relationship between the number of customers and menu pricing, considering the success rate.
184 revenue %>% ggplot(aes(x=Number_of_Customers, y=Menu_Price, color= Success, alpha=0.5)) +
185   geom_point() +
186   geom_smooth(method='lm')
187
188 # Visualize overall success rates within the dataset to understand success distribution.
189 revenue %>% ggplot(aes(x=Success)) +
190   geom_bar()
191
192 # Compare success rates across different cuisine types to identify any trends in cuisine popularity or success.
193 revenue %>% ggplot(aes(x=Cuisine_Type, fill=Success)) +
194   geom_bar()
195 ``
196
197 ````{r}
198 # Random Forest model
199 ctrl <- trainControl(method = "cv", number = 10)
200
201 predictors <- revenue %>% dplyr::select(Number_of_Customers, Marketing_Spend, Cuisine_Type, Reviews)
202 target <- revenue$Success %>% as.factor()
203
204 rf_model <- train(x = predictors,
205                      y = target,
206                      method = "rf",
207                      trControl = ctrl)
208
209 print(rf_model)
210 ````

213 Hubert's Code (CV, bootstrap, model selection):
214 ````{r}
215 # Load data
216 dataset <- read.csv("~/Downloads/Restaurant_revenue (1).csv")
217
218 head(dataset)
219 ``
220
221 ````{r}
222 # Data Preprocessing
223
224 library(caret)
225 # Convert categorical variables to factors for analysis
226 dataset$Cuisine_Type <- as.factor(dataset$Cuisine_Type) # Convert Cuisine_Type to factor
227 dataset$Promotions <- as.factor(dataset$Promotions) # Convert Promotions to factor
228 # Preprocess numeric features: centering and scaling (for SVM)
229 preprocessParams <- preProcess(dataset[, -ncol(dataset)], method = c("center", "scale"))
230 # Modifies original dataset with centered and scaled values for numeric features
231 dataset <- predict(preprocessParams, dataset)
232
233 head(dataset)
234 ``
235
236 ````{r}
237 # K-Fold Cross Validation
238 set.seed(123) # Ensure reproducibility
239 ctrl_kfold <- trainControl(method = "cv", number = 10, summaryFunction = defaultSummary)
240 fit_kfold <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl_kfold)
241 rmse_kfold <- fit_kfold$results$RMSE
242 ````
```

```

244 ````{r}
245 # LOOCV
246 ctrl_loocv <- trainControl(method = "LOOCV", summaryFunction = defaultSummary)
247 fit_loocv <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl_loocv)
248 rmse_loocv <- fit_loocv$results$RMSE
249 ````
250
251 ````{r}
252 # Monte Carlo CV
253 ctrl_mc <- trainControl(method = "repeatedcv", number = 10, repeats = 5, summaryFunction = defaultSummary)
254 fit_mc <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl_mc)
255 rmse_mc <- fit_mc$results$RMSE
256 ````
257
258 ````{r}
259 # Bootstrapping
260 ctrl <- trainControl(method = "boot", number = 1000) # 1000 bootstrap resamples
261 fit_boot <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl)
262 ````
263
264 ````{r}
265 # Support Vector Machine
266 library(e1071)
267 set.seed(123)
268 ctrl <- trainControl(method = "cv", number = 10) # 10-fold CV
269 fit_svm <- train(Monthly_Revenue ~ ., data = dataset, method = "svmRadial", trControl = ctrl, tuneLength = 8)
270 ````
271 ````{r}
272 # RMSE Values (all models)
273
274
275 rmse_boot <- fit_boot$results$RMSE
276 rmse_svm <- fit_svm$results$RMSE
277 # Print RMSE values for all models
278 print(paste("K-Fold CV RMSE:", rmse_kfold))
279 print(paste("LOOCV RMSE:", rmse_loocv))
280 print(paste("Monte Carlo CV RMSE:", rmse_mc))
281 print(paste("Bootstrapping RMSE:", rmse_boot))
282 print(paste("SVM RMSE:", rmse_svm))
283
284 # Determine the model with the lowest RMSE
285 min_rmse_value <- min(rmse_kfold, rmse_loocv, rmse_mc, rmse_boot, rmse_svm)
286 min_rmse_model <- switch(which.min(c(rmse_kfold, rmse_loocv, rmse_mc, rmse_boot, rmse_svm)),
287     "K-Fold CV",
288     "LOOCV",
289     "Monte Carlo CV",
290     "Bootstrapping",
291     "SVM")
292
293 # Print the best model based on RMSE
294 print(paste("The best model based on RMSE is:", min_rmse_model, "with an RMSE of:", min_rmse_value))
295 ````
296 ````{r}
297 # R^2 Values (all models)
298
299
300 # Function to calculate R^2
301 calculateR2 <- function(actual, predicted) {
302   tss <- sum((actual - mean(actual))^2)
303   rss <- sum((actual - predicted)^2)
304   r2 <- 1 - rss / tss
305   return(r2)
306 }
307
308 # K-fold CV
309 # Generate predictions
310 predictions_kfold <- predict(fit_kfold, newdata = dataset)
311 # Calculate R^2 using actual values from your test set and the predictions
312 r_squared_kfold <- calculateR2(dataset$Monthly_Revenue, predictions_kfold)
313 # Print the R-squared value
314 print(paste("K-Fold CV R-squared:", r_squared_kfold))
315
316 # LOOCV
317 # Generate predictions
318 predictions_loocv <- predict(fit_loocv, newdata = dataset)
319 # Calculate R^2 using the actual and predicted values
320 r_squared_loocv <- calculateR2(dataset$Monthly_Revenue, predictions_loocv)
321 # Print the R-squared value for the LOOCV model
322 print(paste("LOOCV R-squared:", r_squared_loocv))
323
324 # Monte Carlo CV
325 # Generate predictions
326 predictions_mc <- predict(fit_mc, newdata = dataset)
327 # Calculate R^2 using the actual and predicted values
328 r_squared_mc <- calculateR2(dataset$Monthly_Revenue, predictions_mc)
329 # Print the R-squared value for the Monte Carlo CV model
330 print(paste("Monte Carlo CV R-squared:", r_squared_mc))

```

```

332 # Bootstrapping
333 # Generate predictions
334 predictions_boot <- predict(fit_boot, newdata = dataset)
335 # Calculate R2 using the actual and predicted values
336 r_squared_boot <- calculateR2(dataset$Monthly_Revenue, predictions_boot)
337 # Print the R-squared value for the Bootstrapping model
338 print(paste("Bootstrapping R-squared:", r_squared_boot))
339
340 # Support Vector Machine
341 # Generate predictions
342 predictions_svm <- predict(fit_svm, newdata = dataset)
343 # Calculate R2 using the actual and predicted values
344 r_squared_svm <- calculateR2(dataset$Monthly_Revenue, predictions_svm)
345 # Print the R-squared value for the SVM model
346 print(paste("SVM R-squared:", r_squared_svm))
347 ```

350 ````{r}
351 # Visualizations
352 library(ggplot2)
353
354 rmse_data <- model_data[model_data$Metric == "RMSE", ]
355 r2_data <- model_data[model_data$Metric == "R2", ]
356
357 # Visualize RMSE Values
358 ggplot(rmse_data, aes(x = Model, y = Value, fill = Model)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  geom_text(aes(label = sprintf("%.2f", Value)), position = position_dodge(width = 0.9), vjust = -0.25, size = 3.5) +
  coord_flip() # Flips the axes for a horizontal bar chart
362 labs(title = "Model Comparison by RMSE",
      x = "Model",
      y = "RMSE") +
365 theme_minimal() +
366 theme(axis.text.x = element_text(angle = 45, hjust = 1),
      legend.position = "none")
368
369 # Visualize R^2 Values
370 ggplot(r2_data, aes(x = Model, y = Value, fill = Model)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  geom_text(aes(label = sprintf("%.4f", Value)), position = position_dodge(width = 0.9), vjust = -0.25, size = 3.5) +
  coord_flip() # Flips the axes for a horizontal bar chart
374 labs(title = "Model Comparison by R2",
      x = "Model",
      y = "R2") +
377 theme_minimal() +
378 theme(axis.text.x = element_text(angle = 45, hjust = 1),
      legend.position = "none")
380 ````

381
382 Best Model Selection:
383 Given the trade-off between RMSE and R2, if the priority is predictive accuracy (minimizing error), the Monte Carlo CV model is slightly better due to its lowest RMS. If the focus is on explaining the variance in revenue (how well the model fits the data), the SVM model is slightly better because of its highest R2 value. Overall, the Monte Carlo CV model stands out for prediction accuracy, while the SVM provides slightly better explanatory power.

```

```

385 ````{r}
386 # Summary of the linear model from K-Fold CV to see coefficients
387 summary(fit_kfold$finalModel)
388
389 # Or directly extract coefficients
390 coefs <- coef(fit_kfold$finalModel)
391 print(coefs)
392 ``
393
394 ````{r}
395 # Check variable importance for models
396 # For the K-Fold model
397 importance <- varImp(fit_kfold, scale = FALSE)
398 print(importance)
399
400 # For the LOOCV model
401 importance_loocv <- varImp(fit_loocv, scale = FALSE)
402 print(importance_loocv)
403
404 # For the Monte Carlo CV model
405 importance_mc <- varImp(fit_mc, scale = FALSE)
406 print(importance_mc)
407
408 # For the Bootstrapping model
409 importance_boot <- varImp(fit_boot, scale = FALSE)
410 print(importance_boot)
411 ``
412
413 Based on these scores, "Number_of_Customers" (41.43) is the most influential in predicting restaurant revenue, having the highest importance score. "Marketing_Spend" (15.08) is the second most influential factor is marketing expenditure. This indicates that the amount spent on marketing activities significantly affects the restaurant's revenue. "Menu_Price" (12.88) is the third most influential factor, as the price of menu items also plays a crucial role in determining revenue. This factor's importance suggests that higher menu prices can positively impact revenue, assuming that the demand remains relatively inelastic or that the restaurant successfully offers value justifying the prices.
414
415 ````{r}
416 library(DALEX)
417
418 # Assuming 'fit_svm' is your trained model and 'dataset' is your dataset
419 # Create an explainer for the SVM model
420 explainer_svm <- explain(fit_svm, data = dataset[, -ncol(dataset)], y = dataset$Monthly_Revenue)
421
422 # Calculate feature importance
423 fi_svm <- model_parts(explainer_svm, loss_function = loss_root_mean_square)
424
425 # Customize the plot with a title and remove unwanted labels
426 p <- plot(fi_svm) +
427   labs(
428     title = "Feature Importance for Monthly Revenue Prediction",
429     subtitle = "Assessing the Impact of Each Feature on Model's RMSE",
430     caption = "" # Removes the default caption which may contain "train.formula"
431   ) +
432   theme_minimal() + # Applies a minimalistic theme
433   theme(
434     plot.title = element_text(size = 14, face = "bold"), # Customize the plot title
435     plot.subtitle = element_text(size = 12), # Customize the plot subtitle
436     axis.title.x = element_text(size = 12), # Customize x axis title
437     axis.title.y = element_text(size = 12), # Customize y axis title
438     axis.text = element_text(size = 10), # Customize axis text size
439     legend.position = "none", # Remove the legend to eliminate "train.formula"
440     plot.caption = element_blank() # Ensure the caption is blank
441   ) +
442   scale_fill_brewer(palette = "Pastel1") # Use a different color palette
443
444 # Print the customized plot
445 print(p)
446 ``
447
448 The uploaded bar chart illustrates the feature importance derived from a predictive model, with the magnitude of each feature's importance indicated by the length of the bars. The importance is measured in terms of the increase in Root Mean Square Error (RMSE) when the feature is permuted (shuffled), which reflects the feature's contribution to the predictive power of the model. Here's the analysis based on the visual information:
449
450 Number_of_Customers: This feature has the longest bar, signifying the highest increase in RMSE when it is permuted. It implies that the number of customers is the most significant predictor of restaurant revenue. Intuitively, this makes sense as more customers generally lead to higher sales and revenue.
451
452 Marketing_Spend: The second-longest bar indicates that marketing spend is also a critical factor in predicting revenue. A significant increase in RMSE upon permutation suggests that how much the restaurant invests in marketing has a strong influence on attracting customers and thus on the revenue.
453
454 Menu_Price: The third factor in terms of importance is the price of the menu items. The model suggests that menu pricing is an important predictor of revenue, but less so than the number of customers or marketing spend.
455
456 Cuisine_Type: This categorical variable shows a moderate level of importance. Since this bar chart does not differentiate between different cuisine types, it's not clear which specific cuisine contributes most to the revenue prediction. However, it does indicate that the type of cuisine is a factor worth considering in the revenue model.
457

```

```

458 Brett's Code (LDA, QDA, EDA (multicollinearity)):
459 ````{r}
460 library(ISLR)
461 library(MASS)
462 library(caret)
463 library(dplyr)
464
465 revenue <- read.csv("~/Downloads/Restaurant_revenue (1).csv")
466 head(revenue)
467 #summary(revenue)
468 ``
469 ## Multicollinearity
470 ````{r}
471 library(car)
472
473 # Create linear model
474 model <- lm(Monthly_Revenue ~ Cuisine_Type + Average_Customer_Spending + Menu_Price
475 + Marketing_Spend + Promotions+ Reviews + Number_of_Customers, data= revenue)
476
477 # Calculate the VIF for each predictor variable in the model
478 vif(model)
479 ``
480
481 ````{r}
482 # Standardize the data (mean of 0,standard deviation of 1)
483 revenue[1:3] <- scale(revenue[1:3])
484 revenue[5:8] <- scale(revenue[5:8])
485 #summary(revenue)
486 # Create the new column to distinguish class among restaurants based on conditions
487 # Got .688 and .718 as the lower and upper quartile respectively
488 revenue <- revenue %>%
489 mutate(Monthly_Revenue_Class = case_when(
490   Monthly_Revenue < -.688 ~ "struggling",
491   Monthly_Revenue >= -.688 & Monthly_Revenue <= .718 ~ "solid",
492   Monthly_Revenue > .718 ~ "excelling"
493 ))
494
495 # View the updated data frame
496 revenue
497 ````

499 ## Linear Discriminant Analysis
500 ````{r}
501
502 # Assuming 'revenue' is your data set
503 set.seed(1) # For reproducibility
504 trainIndex <- sample(1:nrow(revenue), 0.7 *nrow(revenue)) #70% training
505 train_data <- revenue[trainIndex, ]
506 test_data <- revenue[-trainIndex, ]
507
508 # Fit the lda model
509 lda_model <- lda(Monthly_Revenue_Class ~ Cuisine_Type + Monthly_Revenue + Average_Customer_Spending + Menu_Price + Marketing_Spend + Promotions+ Reviews + Number_of_Customers, data =
510 train_data)
511 lda_model_2 <- lda(Monthly_Revenue_Class ~ Number_of_Customers + Marketing_Spend, data = train_data)
512 lda_model_2
513
514 # Predict
515 predicted <- predict(lda_model, data = train_data)
516 predicted2 <- predict(lda_model_2, data = train_data)
517 #head(predicted$)
518
519 predicted$class <- predicted$class[1:length(test_data$Monthly_Revenue_Class)]
520 predicted2$class <- predicted2$class[1:length(test_data$Monthly_Revenue_Class)]
521
522 # Percentage of observations the LDA Model correctly predicted
523 mean1 = mean(predicted$class == test_data$Monthly_Revenue_Class)
524 mean1
525 mean2 = mean(predicted2$class == test_data$Monthly_Revenue_Class)
526 mean2
527 ``
528
529 ````{r}
530 # Generates grid for predictor variables
531 grid <- expand.grid(Number_of_Customers = seq(min(train_data$Number_of_Customers), max(train_data$Number_of_Customers), length.out = 100),
532 Marketing_Spend = seq(min(train_data$Marketing_Spend), max(train_data$Marketing_Spend), length.out = 100))
533
534 # Predict classes for the grid points
535 grid$predicted_class <- predict(lda_model_2, newdata = grid)$class
536
537 # Visualize the decision boundaries
538 library(ggplot2)
539 ggplot(data = grid, aes(x = Number_of_Customers, y = Marketing_Spend, color = predicted_class)) +
540 geom_point(alpha = 0.5) +
541 theme_minimal() +
542 labs(title = "Decision Boundary Visualization for LDA Model", x = "Number of Customers", y = "Marketing Spend")
543 ````
```

```
545 * ## Quadratic Discriminant Analysis
546 * ````{r}
547 # Fit qda model to training data
548 qda_model <- qda(Monthly_Revenue_Class ~ Cuisine_Type + Monthly_Revenue + Average_Customer_Spending + Menu_Price + Marketing_Spend + Promotions+ Reviews + Number_of_Customers, data =
549 train_data)
550 qda_model
551 # Make predictions on test data
552 predictions_QDA = data.frame(predict(qda_model, test_data))
553
554 # Add the predictions in a separate column
555 predictions_QDA = cbind(test_data, predictions_QDA)
556
557 # Count how good the predictions were
558 predictions_QDA %>%
559 count(class, Monthly_Revenue_Class)
560
561 # Calculate accuracy of qda model
562 predictions_QDA %>%
563 summarize(score = mean(class == Monthly_Revenue_Class))
564 ````
```