2024 WQ STA 141C Final Project Report

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

March 2024

Contents

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 preprocessing 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

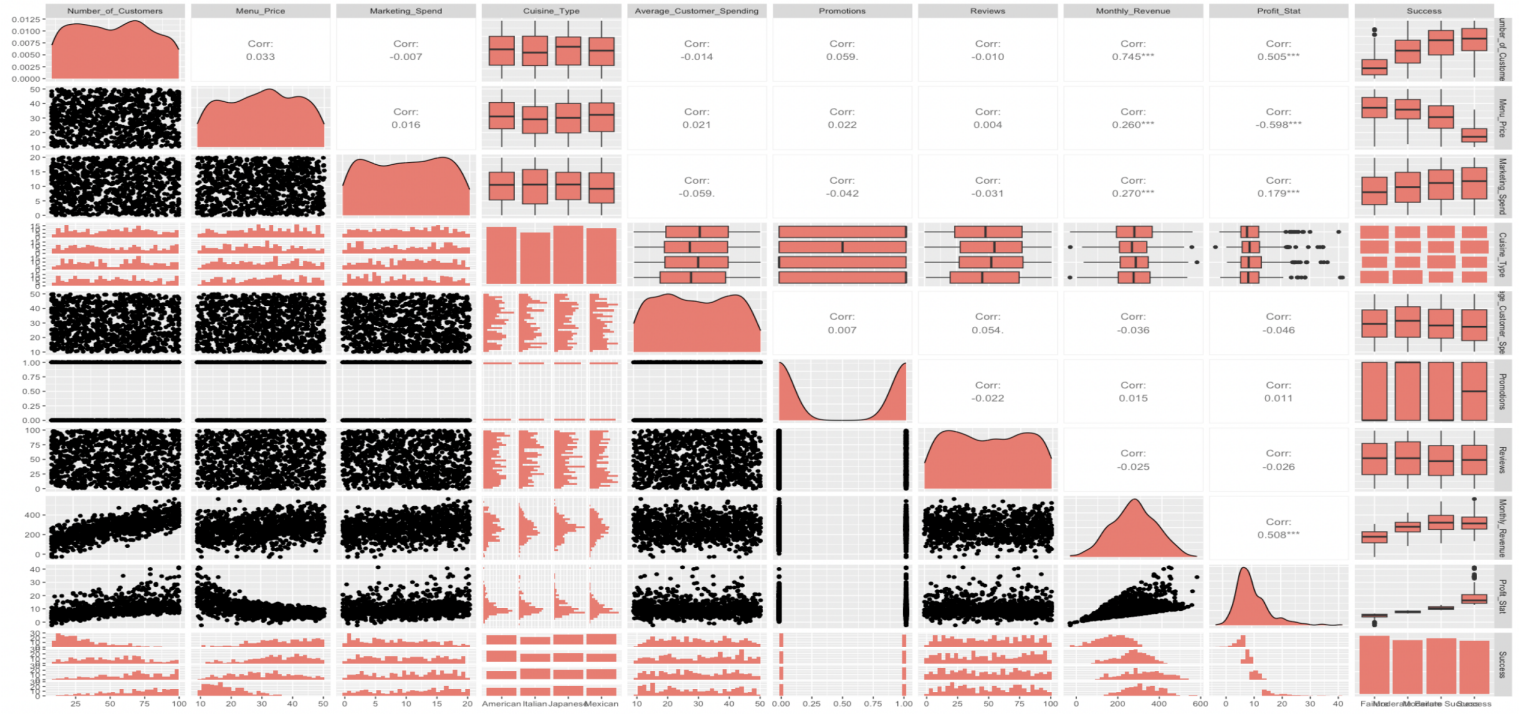


Figure 1: Pairwise correlation plots and graphs for all variables, showing little correlation

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.

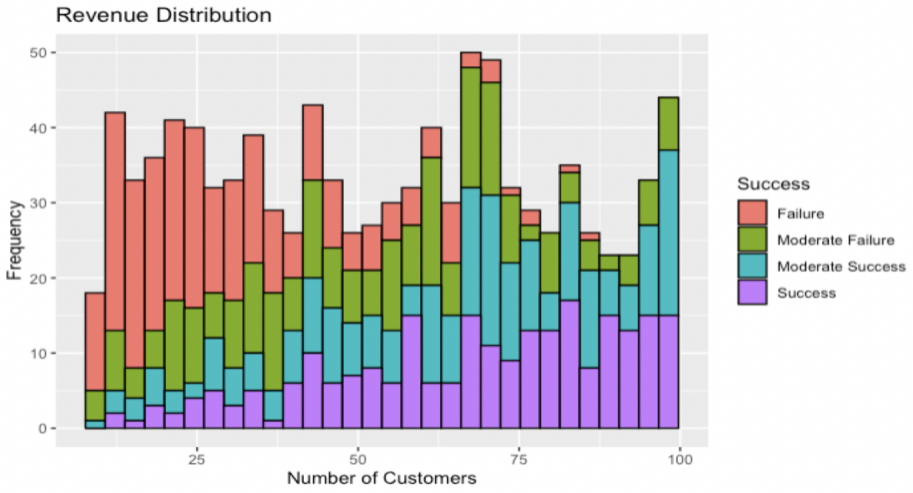


Figure 2: Histogram showing the distribution of the number of customers in restaurants, stratified by success levels

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.

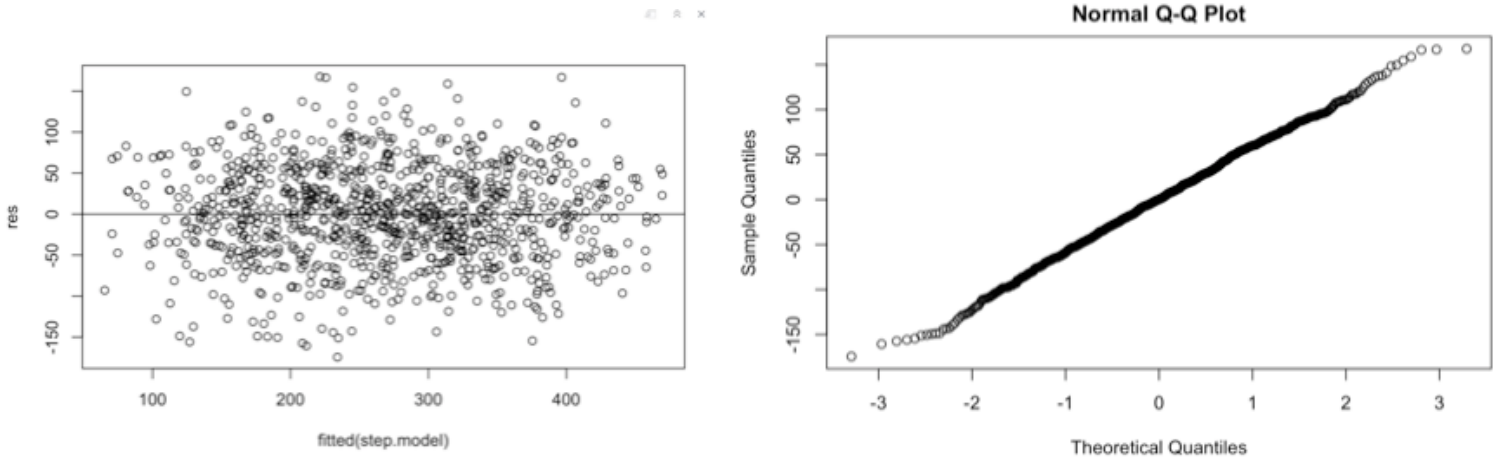
5.3 Classification Techniques

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.

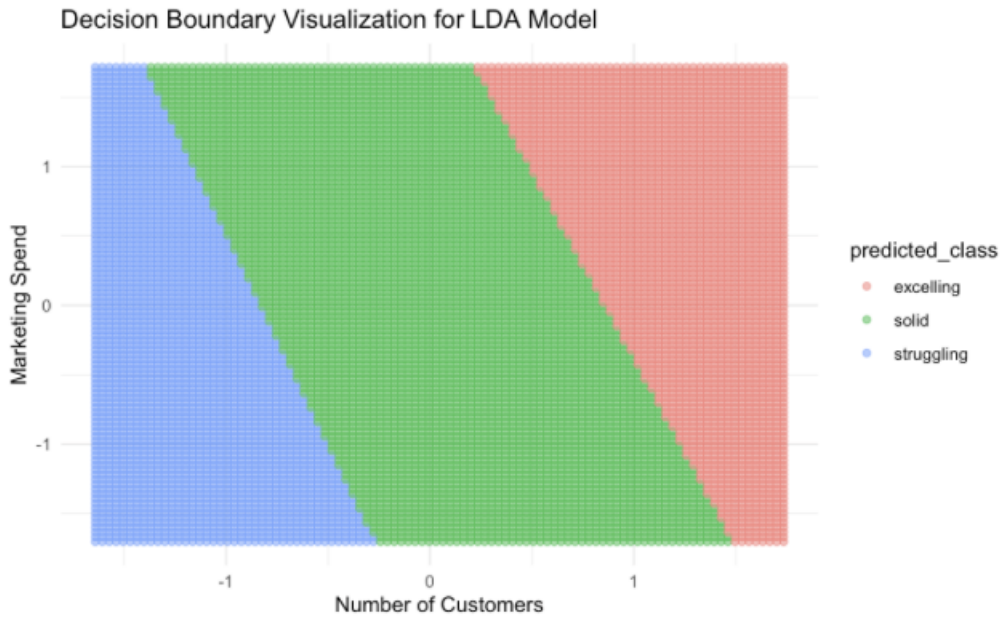


Figure 5: Graph for linear discriminant boundary line where the lines between success clearly favors higher number of customers and marketing expenditures

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.

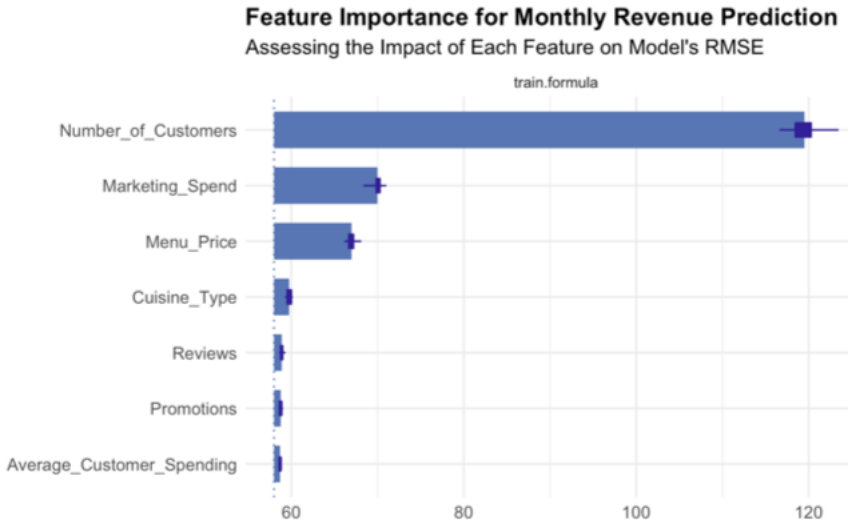


Figure 6: Root mean squared error (RMSE) loss after permutations shows the high impact of Number_of_Customers

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

```
### ALL CODE COMBINED

Data Analysis on Restaurant Revenue Data

Divided into four sections (Jared's code, Judah's code, Hubert's code, Brett's code):
1. Multiple Regression and some exploration by cuisine type
2. EDA, Decision Trees
3. CV, bootstrap, model selection
4. LDA, QDA, EDA (multicollinearity)


Jared's Code (Multiple Regression and some exploration by cuisine type):
### Preliminary
```{r}
# Load necessary libraries
library(tidyverse)
library(corrplot)
library(caret)
library(MASS)
library(leaps)
library(ISLR)
revenue <- read.csv("~/Downloads/Restaurant_revenue (1).csv")
```
```{r}
head(revenue)
unique(revenue[4])
```
```{r}
# Data grouped by cuisine:
Japanese <- revenue %>%
  filter(Cuisine_Type == 'Japanese')
Italian <- revenue %>%
  filter(Cuisine_Type == 'Italian')
American <- revenue %>%
  filter(Cuisine_Type == 'American')
Mexican <- revenue %>%
  filter(Cuisine_Type == 'Mexican')
```
```{r}
# Correlation matrix
corrplot(cor(revenue[,-4]),tl.col="black")
# Mean revenue by cuisine type
revenue %>%
  group_by(Cuisine_Type) %>%
  summarise(mean(Monthly_Revenue),mean(Number_of_Customers),mean(Menu_Price),mean(Marketing_Spend),mean(Average_Customer_Spending),mean(Promotions),mean(Reviews))
# Can add corrplot per cuisine type
```
```

````
### Multiple regression with stepwise selection (By AIC)
```{r}
# Fit a linear model where Monthly_Revenue is predicted by all other variables in the 'revenue' dataset
full <- lm(Monthly_Revenue ~ ., data = revenue)

# Perform stepwise model selection to identify the optimal model based on AIC
# 'direction = "both"' means the algorithm can both add and remove predictors to find the best model
# 'trace = FALSE' suppresses the printing of information at each step, making the output cleaner
step.model <- stepAIC(full, direction = "both", trace = FALSE)
```

By distinct cuisine types:
```{r}
# Fit a linear model predicting Monthly_Revenue from all but the 4th column in 'Japanese'.
fullj <- lm(Monthly_Revenue ~ ., data = Japanese[-4])

# Use stepwise regression to optimize model predictors, allowing both addition and removal.
step.model <- stepAIC(fullj, direction = "both", trace = FALSE)

# Output a summary of the optimized model, including coefficients and diagnostics.
summary(step.model)
```

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

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

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

Judah's code (EDA,PCA, Decision Trees):

```{r setup, include=FALSE}
# Initial Libraries
library(tidyverse)
library(ggplot2)
library(caret)
library(randomForest)
revenue <- read.csv("~/Downloads/Restaurant_revenue (1).csv")

# I'm creating a new column to re-evalue success by revenue per dollar on average menu price
revenue$Profit_Stat = revenue$Monthly_Revenue / revenue$Menu_Price
revenue <- revenue %>% mutate(Success = case_when(
  Profit_Stat > 13 ~ "Success",
  Profit_Stat <= 13 & Profit_Stat > 9 ~ 'Moderate Success',
  Profit_Stat <= 9 & Profit_Stat > 6.5 ~ "Moderate Failure",
  TRUE ~ "Failure"
))

revenue %>% head()
```
````

```
Notes:
Variables are listed below
- number of customers: count of visiting customers
- menu price: average menu prices at restaurant
- marketing spend: expenditure on marketing activities (scale isn't mentioned... so a store spent $3.475052?)
- cuisine type: type of cuisine offered
- average customer spending: average spending per customer
- promotions: binary indicator whether or not promotions were conducted
- reviews: number of reviews received by restaurant
- monthly revenue: just that

-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.

-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)
- heatmaps
- box plots
- histograms
- scatterplots (change color / size)

```

```{r}
# Visualize the distribution of the number of customers, colored by success status, to explore its impact on success.
ggplot(data = revenue, aes(x = Number_of_Customers, fill=Success)) +
  geom_histogram(binwidth = 5, color = "black") +
  labs(title = "Revenue Distribution",
       x = "Number of Customers",
       y = "Frequency")

# Investigate menu price distribution across different cuisines to explore pricing strategies and their correlation to cuisine types.
ggplot(data = revenue, aes(x = Menu_Price, fill=Cuisine_Type)) +
  geom_histogram(binwidth = 2, color = "black") +
  labs(title = "Menu Price Distribution",
       x = "Menu Price",
       y = "Frequency")

# Plot reviews against profit status by cuisine type to understand the relationship between customer feedback and profitability.
ggplot(data = revenue, aes(x=Reviews, y = Profit_Stat, color=Cuisine_Type )) +
  geom_point()

# Display the spread and distribution of profit statistics across the dataset.
ggplot(data=revenue, aes(x=Profit_Stat)) +
  geom_boxplot()

# Examine the relationship between marketing spend and monthly revenue by success, highlighting the effectiveness of marketing strategies.
revenue %>% ggplot(aes(x=Marketing_Spend, y=Monthly_Revenue, color= Success, alpha=0.5)) +
  geom_point() +
  geom_smooth(method='lm')

# Analyze the correlation between menu price and monthly revenue, suggesting higher-priced menus might lead to higher revenue.
revenue %>% ggplot(aes(x=Menu_Price, y=Monthly_Revenue)) +
  geom_point(alpha=0.5) +
  geom_smooth(method='lm')
```

```r

# Explore how marketing spend influences customer volume, differentiated by success, to assess marketing effectiveness.
revenue %>% ggplot(aes(x=Marketing_Spend, y=Number_of_Customers, color= Success, alpha=0.5)) +
  geom_point() +
  geom_smooth(method='lm')

# Evaluate the relationship between the number of customers and menu pricing, considering the success rate.
revenue %>% ggplot(aes(x=Number_of_Customers, y=Menu_Price, color= Success, alpha=0.5)) +
  geom_point() +
  geom_smooth(method='lm')

# Visualize overall success rates within the dataset to understand success distribution.
revenue %>% ggplot(aes(x=Success)) +
  geom_bar()

# Compare success rates across different cuisine types to identify any trends in cuisine popularity or success.
revenue %>% ggplot(aes(x=Cuisine_Type, fill=Success)) +
  geom_bar()
```

```r
# Random Forest model
ctrl <- trainControl(method = "cv", number = 10)

predictors <- revenue %>% dplyr::select(Number_of_Customers, Marketing_Spend, Cuisine_Type, Reviews)
target <- revenue$Success %>% as.factor()

rf_model <- train(x = predictors,
                  y = target,
                  method = "rf",
                  trControl = ctrl)

print(rf_model)
```

Hubert's Code (CV, bootstrap, model selection):

```r
# Load data
dataset <- read.csv("~/Downloads/Restaurant_revenue (1).csv")

head(dataset)
```

```r
# Data Preprocessing

library(caret)
# Convert categorical variables to factors for analysis
dataset$Cuisine_Type <- as.factor(dataset$Cuisine_Type) # Convert Cuisine_Type to factor
dataset$Promotions <- as.factor(dataset$Promotions) # Convert Promotions to factor
# Preprocess numeric features: centering and scaling (for SVM)
preprocessParams <- preProcess(dataset[, -ncol(dataset)], method = c("center", "scale"))
# Modifies original dataset with centered and scaled values for numeric features
dataset <- predict(preprocessParams, dataset)

head(dataset)
```

```r
# K-Fold Cross Validation
set.seed(123) # Ensure reproducibility
ctrl_kfold <- trainControl(method = "cv", number = 10, summaryFunction = defaultSummary)
fit_kfold <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl_kfold)
rmse_kfold <- fit_kfold$results$RMSE
```

```r
# LOOCV
ctrl_loocv <- trainControl(method = "LOOCV", summaryFunction = defaultSummary)
fit_loocv <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl_loocv)
rmse_loocv <- fit_loocv$results$RMSE
```

```r
# Monte Carlo CV
ctrl_mc <- trainControl(method = "repeatedcv", number = 10, repeats = 5, summaryFunction = defaultSummary)
fit_mc <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl_mc)
rmse_mc <- fit_mc$results$RMSE
```

```r
# Bootstrapping
ctrl <- trainControl(method = "boot", number = 1000) # 1000 bootstrap resamples
fit_boot <- train(Monthly_Revenue ~ ., data = dataset, method = "lm", trControl = ctrl)
```

```r
# Support Vector Machine
library(e1071)
set.seed(123)
ctrl <- trainControl(method = "cv", number = 10) # 10-fold CV
fit_svm <- train(Monthly_Revenue ~ ., data = dataset, method = "svmRadial", trControl = ctrl, tuneLength = 8)
```

```r
# RMSE Values (all models)

rmse_boot <- fit_boot$results$RMSE
rmse_svm <- fit_svm$results$RMSE
# Print RMSE values for all models
print(paste("K-Fold CV RMSE:", rmse_kfold))
print(paste("LOOCV RMSE:", rmse_loocv))
print(paste("Monte Carlo CV RMSE:", rmse_mc))
print(paste("Bootstrapping RMSE:", rmse_boot))
print(paste("SVM RMSE:", rmse_svm))

# Determine the model with the lowest RMSE
min_rmse_value <- min(rmse_kfold, rmse_loocv, rmse_mc, rmse_boot, rmse_svm)
min_rmse_model <- switch(which.min(c(rmse_kfold, rmse_loocv, rmse_mc, rmse_boot, rmse_svm)),
                         "K-Fold CV",
                         "LOOCV",
                         "Monte Carlo CV",
                         "Bootstrapping",
                         "SVM")

# Print the best model based on RMSE
print(paste("The best model based on RMSE is:", min_rmse_model, "with an RMSE of:", min_rmse_value))
```

```r
# R^2 Values (all models)

# Function to calculate R^2
calculateR2 <- function(actual, predicted) {
  tss <- sum((actual - mean(actual))^2)
  rss <- sum((actual - predicted)^2)
  r2 <- 1 - rss / tss
  return(r2)
}

# K-fold CV
# Generate predictions
predictions_kfold <- predict(fit_kfold, newdata = dataset)
# Calculate R² using actual values from your test set and the predictions
r_squared_kfold <- calculateR2(dataset$Monthly_Revenue, predictions_kfold)
# Print the R-squared value
print(paste("K-Fold CV R-squared:", r_squared_kfold))

# LOOCV
# Generate predictions
predictions_loocv <- predict(fit_loocv, newdata = dataset)
# Calculate R² using the actual and predicted values
r_squared_loocv <- calculateR2(dataset$Monthly_Revenue, predictions_loocv)
# Print the R-squared value for the LOOCV model
print(paste("LOOCV R-squared:", r_squared_loocv))

# Monte Carlo CV
# Generate predictions
predictions_mc <- predict(fit_mc, newdata = dataset)
# Calculate R² using the actual and predicted values
r_squared_mc <- calculateR2(dataset$Monthly_Revenue, predictions_mc)
# Print the R-squared value for the Monte Carlo CV model
print(paste("Monte Carlo CV R-squared:", r_squared_mc))
```

```
# Bootstrapping
# Generate predictions
predictions_boot <- predict(fit_boot, newdata = dataset)
# Calculate R² using the actual and predicted values
r_squared_boot <- calculateR2(dataset$Monthly_Revenue, predictions_boot)
# Print the R-squared value for the Bootstrapping model
print(paste("Bootstrapping R-squared:", r_squared_boot))

# Support Vector Machine
# Generate predictions
predictions_svm <- predict(fit_svm, newdata = dataset)
# Calculate R² using the actual and predicted values
r_squared_svm <- calculateR2(dataset$Monthly_Revenue, predictions_svm)
# Print the R-squared value for the SVM model
print(paste("SVM R-squared:", r_squared_svm))
```

```{r}
# Visualizations
library(ggplot2)

rmse_data <- model_data[model_data$Metric == "RMSE", ]
r2_data <- model_data[model_data$Metric == "R2", ]

# Visualize RMSE Values
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
  labs(title = "Model Comparison by RMSE",
       x = "Model",
       y = "RMSE") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")

# Visualize R^2 Values
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
  labs(title = "Model Comparison by R²",
       x = "Model",
       y = "R²") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```

Best Model Selection:
Given the trade-off between RMSE and R^2, 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 R^2 value. Overall, the Monte Carlo CV model stands out for prediction accuracy, while the SVM provides slightly better explanatory power.

```r
# Summary of the linear model from K-Fold CV to see coefficients
summary(fit_kfold$finalModel)

# Or directly extract coefficients
coefs <- coef(fit_kfold$finalModel)
print(coefs)
```

```r
# Check variable importance for models
# For the K-Fold model
importance <- varImp(fit_kfold, scale = FALSE)
print(importance)

# For the LOOCV model
importance_loocv <- varImp(fit_loocv, scale = FALSE)
print(importance_loocv)

# For the Monte Carlo CV model
importance_mc <- varImp(fit_mc, scale = FALSE)
print(importance_mc)

# For the Bootstrapping model
importance_boot <- varImp(fit_boot, scale = FALSE)
print(importance_boot)
```

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.

```r
library(DALEX)

# Assuming 'fit_svm' is your trained model and 'dataset' is your dataset
# Create an explainer for the SVM model
explainer_svm <- explain(fit_svm, data = dataset[, -ncol(dataset)], y = dataset$Monthly_Revenue)

# Calculate feature importance
fi_svm <- model_parts(explainer_svm, loss_function = loss_root_mean_square)

# Customize the plot with a title and remove unwanted labels
p <- plot(fi_svm) +
  labs(
    title = "Feature Importance for Monthly Revenue Prediction",
    subtitle = "Assessing the Impact of Each Feature on Model's RMSE",
    caption = ""  # Removes the default caption which may contain "train.formula"
  ) +
  theme_minimal() +  # Applies a minimalistic theme
  theme(
    plot.title = element_text(size = 14, face = "bold"),  # Customize the plot title
    plot.subtitle = element_text(size = 12),  # Customize the plot subtitle
    axis.title.x = element_text(size = 12),  # Customize x axis title
    axis.title.y = element_text(size = 12),  # Customize y axis title
    axis.text = element_text(size = 10),  # Customize axis text size
    legend.position = "none",  # Remove the legend to eliminate "train.formula"
    plot.caption = element_blank()  # Ensure the caption is blank
  ) +
  scale_fill_brewer(palette = "Pastel1")  # Use a different color palette

# Print the customized plot
print(p)
```

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:

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.

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.

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.

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.

Brett's Code (LDA, QDA, EDA (multicollinearity)):

```{r}
library(ISLR)
library(MASS)
library(caret)
library(dplyr)

revenue <- read.csv("~/Downloads/Restaurant_revenue (1).csv")
head(revenue)
#summary(revenue)
```

Multicollinearity

```{r}
library(car)

# Create linear model
model <- lm(Monthly_Revenue ~ Cuisine_Type + Average_Customer_Spending + Menu_Price
+ Marketing_Spend + Promotions+ Reviews + Number_of_Customers, data= revenue)

# Calculate the VIF for each predictor variable in the model
vif(model)
```

```{r}
# Standardize the data (mean of 0,standard deviation of 1)
revenue[1:3] <- scale(revenue[1:3])
revenue[5:8] <- scale(revenue[5:8])
#summary(revenue)
# Create the new column to distinguish class among restaurants based on conditions
# Got .688 and .718 as the lower and upper quartile respectively
revenue <- revenue %>%
 mutate(Monthly_Revenue_Class = case_when(
    Monthly_Revenue < -.688 ~ "struggling",
    Monthly_Revenue >= -.688 & Monthly_Revenue <= .718 ~ "solid",
    Monthly_Revenue > .718 ~ "excelling"
  ))

# View the updated data frame
revenue
```

Linear Discriminant Analysis

```{r}

# Assuming 'revenue' is your data set
set.seed(1) # For reproducibility
trainIndex <- sample(1:nrow(revenue), 0.7 *nrow(revenue)) #70% training
train_data <- revenue[trainIndex, ]
test_data <- revenue[-trainIndex, ]

# Fit the lda model
lda_model <- lda(Monthly_Revenue_Class ~ Cuisine_Type + Monthly_Revenue + Average_Customer_Spending + Menu_Price + Marketing_Spend + Promotions+ Reviews + Number_of_Customers, data =
train_data)

lda_model_2 <- lda(Monthly_Revenue_Class ~ Number_of_Customers + Marketing_Spend, data = train_data)
lda_model_2

# Predict
predicted <- predict(lda_model, data = train_data)
predicted2 <- predict(lda_model_2, data = train_data)
#head(predicted$x)

predicted$class <- predicted$class[1:length(test_data$Monthly_Revenue_Class)]
predicted2$class <- predicted2$class[1:length(test_data$Monthly_Revenue_Class)]

# Percentage of observations the LDA Model correctly predicted
mean1 = mean(predicted$class == test_data$Monthly_Revenue_Class)
mean1
mean2 = mean(predicted2$class == test_data$Monthly_Revenue_Class)
mean2
```

```{r}
# Generates grid for predictor variables
grid <- expand.grid(Number_of_Customers = seq(min(train_data$Number_of_Customers), max(train_data$Number_of_Customers), length.out = 100),
                    Marketing_Spend = seq(min(train_data$Marketing_Spend), max(train_data$Marketing_Spend), length.out = 100))

# Predict classes for the grid points
grid$predicted_class <- predict(lda_model_2, newdata = grid)$class

# Visualize the decision boundaries
library(ggplot2)
ggplot(data = grid, aes(x = Number_of_Customers, y = Marketing_Spend, color = predicted_class)) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  labs(title = "Decision Boundary Visualization for LDA Model", x = "Number of Customers", y = "Marketing Spend")
```

````
## Quadratic Discriminant Analysis
```{r}
# Fit qda model to training data
qda_model <- qda(Monthly_Revenue_Class ~ Cuisine_Type + Monthly_Revenue + Average_Customer_Spending + Menu_Price + Marketing_Spend + Promotions+ Reviews + Number_of_Customers, data =
train_data)
qda_model

# Make predictions on test data
predictions_QDA = data.frame(predict(qda_model, test_data))

# Add the predictions in a separate column
predictions_QDA = cbind(test_data, predictions_QDA)

# Count how good the predictions were
predictions_QDA %>%
  count(class, Monthly_Revenue_Class)

# Calculate accuracy of qda model
predictions_QDA %>%
  summarize(score = mean(class == Monthly_Revenue_Class))
```
````