Predictive Modeling and Analysis of Online Food Delivery Services

Anuj Prabhu

1 April 2024

Contents

Background 1
Problem Statement 1
Loading the Dataset 1
General Summary 2
Data Cleaning 3
Exploratory Data Analysis 6
Data Visualization 13
Conclusion 18
Dataset link 18

Background

- In the modern era of digital convenience, online food ordering platforms have become an integral part of many people's lives, offering a convenient way to access a wide range of culinary options. Understanding the dynamics of customer behavior and satisfaction within this domain is crucial for platform operators to enhance service quality and cater to evolving consumer preferences.
- The dataset under analysis contains comprehensive information collected from an online food ordering platform over a long period of time. It encompasses demographic attributes such as age, gender, marital status, occupation, and educational qualifications of customers, as well as location-specific details like their latitude, longitude, and pin code data. Additionally, it includes crucial feedback from customers regarding their satisfaction with the service, alongside the outcome of their orders.
- With the aim of delving into the intricate relationship between demographic/location factors and online food ordering behavior, the project embarks on an exploratory journey. Through rigorous analysis and modeling techniques, it seeks to uncover valuable insights that can guide decision-making processes and improve service quality within the online food-ordering landscape.

Problem Statement

- The objective of this project is to **analyze the impact of demographic attributes on customer feedback regarding their orders and the resultant order output**. By examining factors such as age, gender, location, and any other relevant demographic information, we aim to discern patterns in customer satisfaction and identify potential correlations between demographics and feedback sentiment. Thus, this analysis will provide **valuable** insights for **improving customer experience** and optimizing order fulfillment processes.

Loading the Dataset

```
#Loading the Dataset
online_foods <- read.csv("./data/onlinefoods.csv", header = TRUE, sep = ",", stringsAsFactors = TRUE)
```

- We load the "onlinefoods" dataset from our local directory and specify the parameters for loading the data into our project. To facilitate data manipulation, we convert strings to factors, leveraging the ease of handling factors. The dataset is then stored in the variable named online_foods.

General Summary

```
#general summary
head(online_foods)
```

```
##   Age Gender Marital.Status Occupation  Monthly.Income
## 1  20 Female         Single    Student       No Income
## 2  24 Female         Single    Student  Below Rs.10000
## 3  22   Male         Single    Student  Below Rs.10000
## 4  22 Female         Single    Student       No Income
## 5  22   Male         Single    Student  Below Rs.10000
## 6  27 Female        Married   Employee More than 50000
##   Educational.Qualifications Family.size latitude longitude Pin.code Output
## 1              Post Graduate           4  12.9766   77.5993   560001    Yes
## 2                   Graduate           3  12.9770   77.5773   560009    Yes
## 3              Post Graduate           3  12.9551   77.6593   560017    Yes
## 4                   Graduate           6  12.9473   77.5616   560019    Yes
## 5              Post Graduate           4  12.9850   77.5533   560010    Yes
## 6              Post Graduate           2  12.9299   77.6848   560103    Yes
##    Feedback   X
## 1  Positive Yes
## 2  Positive Yes
## 3 Negative  Yes
## 4  Positive Yes
## 5  Positive Yes
## 6  Positive Yes
```

```
summary(online_foods)
```

```
##       Age           Gender              Marital.Status          Occupation
##  Min.   :18.00   Female:166   Married          :108    Employee      :118
##  1st Qu.:23.00   Male  :222   Prefer not to say: 12    House wife    :  9
##  Median :24.00                Single           :268    Self Employeed: 54
##  Mean   :24.63                                         Student       :207
##  3rd Qu.:26.00
##  Max.   :33.00
##          Monthly.Income Educational.Qualifications  Family.size
##  10001 to 25000 : 45    Graduate     :177          Min.   :1.000
##  25001 to 50000 : 69    Ph.D         : 23          1st Qu.:2.000
##  Below Rs.10000 : 25    Post Graduate:174          Median :3.000
##  More than 50000: 62    School       : 12          Mean   :3.281
##  No Income      :187    Uneducated   :  2          3rd Qu.:4.000
##                                                    Max.   :6.000
##     latitude       longitude        Pin.code      Output         Feedback
##  Min.   :12.87   Min.   :77.48   Min.   :560001   No : 87   Negative : 71
##  1st Qu.:12.94   1st Qu.:77.57   1st Qu.:560011   Yes:301   Positive :317
##  Median :12.98   Median :77.59   Median :560034
##  Mean   :12.97   Mean   :77.60   Mean   :560040
##  3rd Qu.:13.00   3rd Qu.:77.63   3rd Qu.:560068
##  Max.   :13.10   Max.   :77.76   Max.   :560109
##    X
```

```
##  No : 87
##  Yes:301
##
##
##
##
```

```
str(online_foods)
```

```
## 'data.frame':    388 obs. of  13 variables:
##  $ Age                       : int  20 24 22 22 22 27 22 24 23 23 ...
##  $ Gender                    : Factor w/ 2 levels "Female","Male": 1 1 2 1 2 1 2 1 1 1 ...
##  $ Marital.Status            : Factor w/ 3 levels "Married","Prefer not to say",..: 3 3 3 3 3 1 3 3 3
##  $ Occupation                : Factor w/ 4 levels "Employee","House wife",..: 4 4 4 4 4 1 4 4 4 4 ..
##  $ Monthly.Income            : Factor w/ 5 levels "10001 to 25000",..: 5 3 3 5 3 4 5 5 5 5 ...
##  $ Educational.Qualifications: Factor w/ 5 levels "Graduate","Ph.D",..: 3 1 3 1 3 3 1 3 3 3 ...
##  $ Family.size               : int  4 3 3 6 4 2 3 3 2 4 ...
##  $ latitude                  : num  13 13 13 12.9 13 ...
##  $ longitude                 : num  77.6 77.6 77.7 77.6 77.6 ...
##  $ Pin.code                  : int  560001 560009 560017 560019 560010 560103 560009 560042 560001 56
##  $ Output                    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Feedback                  : Factor w/ 2 levels "Negative ","Positive": 2 2 1 2 2 2 2 2 2 2 ...
##  $ X                         : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

- The dataset comprises of 388 observations across 13 variables. Notably, variables such as Monthly.Income and Educational.Qualifications exhibit numerous factors that are very similar in nature. Additionally, the dataset contains a variable labeled "X," for which the dataset owner has not provided clarification regarding its significance or definition. Therefore, further exploration of variable "X" is warranted to ascertain its purpose and relevance within the dataset.

```
#understand "X" variable better
unique_values <- online_foods %>%
  distinct(X)
unique_values
```

```
##     X
## 1 Yes
## 2  No
```

- Upon examination, the variable "X" is observed to have values labeled as "Yes" and "No," which lack clarity and fail to provide explanatory context. In addition to an absence of information about this variable on the dataset's source website, its ambiguity renders it irrelevant for our analysis. Consequently, we proceed to the data cleaning phase of this project, where the variable "X" will be removed from consideration.

Data Cleaning

```
online_foods <- online_foods %>%
  select(-X) #removing "X" variable from online_foods because no background has been provided about thi

#check for NA values in dataset
has_na <- any(is.na(online_foods))
has_na #no NA values in dataset
```

```
## [1] FALSE
```

- We exclude the "X" variable from further analysis. Additionally, after confirming the absence of NA

(missing) values in the dataset, our next step involves addressing the issue of numerous similar factors by amalgamating them where appropriate.

```
#Excluding "Prefer not to say" Marital Status values from analysis (handling outlying data points)
online_foods$Marital.Status <- as.character(online_foods$Marital.Status)
online_foods <- online_foods %>%
  filter(online_foods$Marital.Status != "Prefer not to say")
online_foods$Marital.Status <- as.factor(online_foods$Marital.Status)

#Combining like factors
online_foods$Occupation <- as.character(online_foods$Occupation)
online_foods$Occupation <- case_when(
  online_foods$Occupation %in% c("Employee", "Self Employeed") ~ "Employed",
  #online_foods$Occupation == "House wife" ~ "Housewife",
  TRUE ~ as.character(online_foods$Occupation) # Keep other values unchanged
)
online_foods$Occupation <- as.factor(online_foods$Occupation)

online_foods$Monthly.Income <- as.character(online_foods$Monthly.Income)
online_foods$Monthly.Income <- case_when(
  online_foods$Monthly.Income %in% c("10001 to 25000", "25001 to 50000") ~ "Average Monthly Income",
  online_foods$Monthly.Income == "More than 50000" ~ "High Monthly Income",
  online_foods$Monthly.Income %in% c("No Income", "Below Rs.10000") ~ "Low/No Monthly Income",
  TRUE ~ as.character(online_foods$Monthly.Income) # Keep other values unchanged
)
online_foods$Monthly.Income <- as.factor(online_foods$Monthly.Income)

online_foods$Educational.Qualifications <- as.character(online_foods$Educational.Qualifications)
online_foods$Educational.Qualifications <- case_when(
  online_foods$Educational.Qualifications %in% c("Post Graduate", "Ph.D") ~ "Higher Education",
  online_foods$Educational.Qualifications %in% c("School", "Uneducated") ~ "Lower/No Education",
  TRUE ~ as.character(online_foods$Educational.Qualifications) # Keep other values unchanged
)
online_foods$Educational.Qualifications <- as.factor(online_foods$Educational.Qualifications)

online_foods$Output <- as.character(online_foods$Output)
online_foods$Output <- case_when(
  online_foods$Output == "No" ~ "Unsuccessful",
  online_foods$Output == "Yes" ~ "Successful",
  TRUE ~ as.character(online_foods$Output) # Keep other values unchanged
)
online_foods$Output <- as.factor(online_foods$Output)
str(online_foods)
```

```
## 'data.frame':    376 obs. of  12 variables:
##  $ Age                        : int  20 24 22 22 22 27 22 24 23 23 ...
##  $ Gender                     : Factor w/ 2 levels "Female","Male": 1 1 2 1 2 1 2 1 1 1 ...
##  $ Marital.Status             : Factor w/ 2 levels "Married","Single": 2 2 2 2 2 1 2 2 2 2 ...
##  $ Occupation                 : Factor w/ 3 levels "Employed","House wife",..: 3 3 3 3 3 1 3 3 3 3 ..
##  $ Monthly.Income             : Factor w/ 3 levels "Average Monthly Income",..: 3 3 3 3 3 2 3 3 3 3 .
##  $ Educational.Qualifications: Factor w/ 3 levels "Graduate","Higher Education",..: 2 1 2 1 2 2 1 2 1
##  $ Family.size                : int  4 3 3 6 4 2 3 3 2 4 ...
##  $ latitude                   : num  13 13 13 12.9 13 ...
##  $ longitude                  : num  77.6 77.6 77.7 77.6 77.6 ...
##  $ Pin.code                   : int  560001 560009 560017 560019 560010 560103 560009 560042 560001 56
```

```
##  $ Output                    : Factor w/ 2 levels "Successful","Unsuccessful": 1 1 1 1 1 1 1 1 1 1 .
##  $ Feedback                  : Factor w/ 2 levels "Negative ","Positive": 2 2 1 2 2 2 2 2 2 2 ...
```

- After consolidating similar factors across multiple variables to alleviate ambiguity, we have enhanced the clarity of our data for exploratory analysis. For instance, the "Monthly.Income" variable originally encompassed values such as "Below Rs.10000", "More than 50000", and "No Income". As it can be observed, one label had a rupee symbol while others did not. These labels presented an inconsistency in the representation of income levels. To rectify this, we have transformed these labels into **three simplified categories**: **"Low/No Monthly Income", "Average Monthly Income", and "High Monthly Income"**. This streamlined data enables a clearer understanding of the relationship between a customer's monthly income and their online food ordering behavior. Similar refinement procedures have been applied to other variables, enhancing their interpretability and facilitating further analysis.

```
#reordering factors according to customs levels
online_foods$Marital.Status <- factor(online_foods$Marital.Status,
                                      levels = c("Single", "Married"))
online_foods$Occupation <- factor(online_foods$Occupation,
                                  levels = c("Student", "House wife", "Employed"))
online_foods$Monthly.Income <- factor(online_foods$Monthly.Income,
                                      levels = c("Low/No Monthly Income", "Average Monthly Income",
                                                 "High Monthly Income"))
online_foods$Educational.Qualifications <- factor(online_foods$Educational.Qualifications,
                                                  levels = c("Lower/No Education", "Graduate",
                                                             "Higher Education"))
online_foods$Output <- factor(online_foods$Output,
                                      levels = c("Unsuccessful", "Successful"))
summary(online_foods)
```

```
##       Age           Gender    Marital.Status      Occupation
##  Min.   :18.00   Female:161   Single :268    Student   :207
##  1st Qu.:22.75   Male  :215   Married:108    House wife:  8
##  Median :24.00                               Employed  :161
##  Mean   :24.56
##  3rd Qu.:26.00
##  Max.   :33.00
##                 Monthly.Income      Educational.Qualifications  Family.size
##  Low/No Monthly Income :209    Lower/No Education: 14          Min.   :1.000
##  Average Monthly Income:109    Graduate          :172          1st Qu.:2.000
##  High Monthly Income   : 58    Higher Education  :190          Median :3.000
##                                                               Mean   :3.303
##                                                               3rd Qu.:4.000
##                                                               Max.   :6.000
##     latitude       longitude        Pin.code               Output
##  Min.   :12.87   Min.   :77.48   Min.   :560001   Unsuccessful: 81
##  1st Qu.:12.94   1st Qu.:77.57   1st Qu.:560011   Successful  :295
##  Median :12.98   Median :77.59   Median :560034
##  Mean   :12.97   Mean   :77.60   Mean   :560040
##  3rd Qu.:12.99   3rd Qu.:77.63   3rd Qu.:560067
##  Max.   :13.10   Max.   :77.76   Max.   :560109
##       Feedback
##  Negative : 66
##  Positive :310
##
##
##
```

```
##
```

```
head(online_foods)
```

```
##   Age Gender Marital.Status Occupation        Monthly.Income
## 1  20 Female         Single    Student Low/No Monthly Income
## 2  24 Female         Single    Student Low/No Monthly Income
## 3  22   Male         Single    Student Low/No Monthly Income
## 4  22 Female         Single    Student Low/No Monthly Income
## 5  22   Male         Single    Student Low/No Monthly Income
## 6  27 Female        Married   Employed   High Monthly Income
##   Educational.Qualifications Family.size latitude longitude Pin.code     Output
## 1           Higher Education           4  12.9766   77.5993   560001 Successful
## 2                   Graduate           3  12.9770   77.5773   560009 Successful
## 3           Higher Education           3  12.9551   77.6593   560017 Successful
## 4                   Graduate           6  12.9473   77.5616   560019 Successful
## 5           Higher Education           4  12.9850   77.5533   560010 Successful
## 6           Higher Education           2  12.9299   77.6848   560103 Successful
##    Feedback
## 1  Positive
## 2  Positive
## 3 Negative
## 4  Positive
## 5  Positive
## 6  Positive
```

- Following the consolidation of similar factors and the standardization of variable representations, we proceed to recode the factor levels of variables according to custom categories. This ensures uniformity across variables and enhances the interpretability of factors. With our data now cleaned and standardized, we are well-equipped to conduct exploratory analysis and derive meaningful insights from the dataset.

Exploratory Data Analysis

- Firstly, we want to determine the location of the online food ordering platform. Initially, we consider that the monthly income of its consumers is expressed in rupees, indicating a connection to South Asian countries. However, the specific city, state, and country where the app is based remains unknown. Hence, we initiate an exploration to address this uncertainty.

```
#understand which region orders are being placed from; we know currency is rupees but do not know count
coordinates_df <- data.frame(
  latitude = c(online_foods[which.min(online_foods$latitude), "latitude"],
               online_foods[which.max(online_foods$latitude), "latitude"],
               online_foods[which.min(online_foods$longitude), "latitude"],
               online_foods[which.max(online_foods$longitude), "latitude"]),
  longitude = c(online_foods[which.min(online_foods$latitude), "longitude"],
                online_foods[which.max(online_foods$latitude), "longitude"],
                online_foods[which.min(online_foods$longitude), "longitude"],
                online_foods[which.max(online_foods$longitude), "longitude"])
)
coordinates_df
```

```
##   latitude longitude
## 1  12.8652   77.5240
## 2  13.1020   77.5864
## 3  12.9105   77.4842
```

```
## 4  12.9967   77.7582
```

- First, we extract the four farthest points from where consumers in this sample have placed orders. Then, we use the Google Maps API to **reverse geocode** the latitude and longitude values into street addresses that we can recognize easily. **Reverse geocoding** is the process of converting latitude and longitude coordinates into a human-readable address or location.

```
API_KEY <- Sys.getenv("API_KEY")
register_google(key = API_KEY)
```

```
region_info <- suppressWarnings(apply(coordinates_df, 1, function(row) { #apply function(row) to each r
  revgeocode(c(row["longitude"], row["latitude"]), output_type = "more")
}))
```

```
## i <https://maps.googleapis.com/maps/api/geocode/json?latlng=12.8652,77.524&key=xxx>
## !   00, Bellanduru, Mahadevapura Zone, Bangalore, Nagegowdanapalya, Bengaluru, Uttarahalli -Manavartl
## !   61, Nagegowdanapalya, Bengaluru, Uttarahalli -Manavarthekaval, Karnataka 560109, India
## !   01, Nagegowdanapalya, Bengaluru, Uttarahalli -Manavarthekaval, Karnataka 560109, India
## !   VG8F+3J Bengaluru, Karnataka, India
## !   Unnamed Road, Nagegowdanapalya, Bengaluru, Uttarahalli -Manavarthekaval, Karnataka 560062, India
## !   Nagegowdanapalya, Bengaluru, Karnataka 560109, India
## !   Talaghattapura, Bengaluru, Karnataka 560109, India
## !   Uttarahalli -Manavarthekaval, Karnataka, India
## !   Bengaluru South, Karnataka, India
## !   Bengaluru, Karnataka, India
## !   Bengaluru Urban, Karnataka, India
## !   Bangalore Division, Karnataka, India
## !   Karnataka, India
## !   India
## i <https://maps.googleapis.com/maps/api/geocode/json?latlng=13.102,77.5864&key=xxx>
## !   10, Yelahanka Satellite Town, Nakkala Halli, Bengaluru, Karnataka 560064, India
## !   4H2P+RH Bengaluru, Karnataka, India
## !   Yelahanka Satellite Town, Yelahanka, Bengaluru, Karnataka, India
## !   Yelahanka New Town, Bengaluru, Karnataka, India
## !   Bengaluru, Karnataka 560064, India
## !   Bengaluru Urban, Karnataka, India
## !   Bengaluru, Karnataka, India
## !   Bangalore Division, Karnataka, India
## !   Karnataka, India
## !   India
## i <https://maps.googleapis.com/maps/api/geocode/json?latlng=12.9105,77.4842&key=xxx>
## !   Maniunath Gobj Center 90, Harsha Layout, Kengeri, Bengaluru, Karnataka 560060, India
```

```
## !   11, Udaya Layout, Manganahalli, Bengaluru, Karnataka 560060, India
## !   WF6M+5M Bengaluru, Karnataka, India
## !   Harsha Layout, Kengeri Satellite Town, Bengaluru, Karnataka, India
## !   Kengeri, Bengaluru, Karnataka 560060, India
## !   Bengaluru, Karnataka 560060, India
## !   Bengaluru South, Karnataka, India
## !   Bengaluru, Karnataka, India
## !   Bengaluru Urban, Karnataka, India
## !   Bangalore Division, Karnataka, India
## !   Karnataka, India
## !   India
## i <https://maps.googleapis.com/maps/api/geocode/json?latlng=12.9967,77.7582&key=xxx>
## !   32, near MVJ COLLEGE, AKG COLONY, Belatur Colony, Brindavan Layout, Bengaluru, Karnataka 560067,
## !   XQW5+M7 Bengaluru, Karnataka, India
## !   Brindavan Layout, Bengaluru, Karnataka 560067, India
## !   Belatur Colony, Krishnarajapuram, Bengaluru, Karnataka 560067, India
## !   Bengaluru, Karnataka 560067, India
## !   Bengaluru South, Karnataka, India
## !   Bengaluru, Karnataka, India
## !   Bengaluru Urban, Karnataka, India
## !   Bangalore Division, Karnataka, India
## !   Karnataka, India
## !   India
```

```
region_info #orders are placed from Begaluru, Karnataka, India.
```

```
## [1] "00, Bellanduru, Mahadevapura Zone, Bangalore, Nagegowdanapalya, Bengaluru, Uttarahalli -Manavar
## [2] "10, Yelahanka Satellite Town, Nakkala Halli, Bengaluru, Karnataka 560064, India"
## [3] "Maniunath Gobj Center 90, Harsha Layout, Kengeri, Bengaluru, Karnataka 560060, India"
## [4] "32, near MVJ COLLEGE, AKG COLONY, Belatur Colony, Brindavan Layout, Bengaluru, Karnataka 560067
```

- We apply the revgeocode() function to each pair of coordinate values and extract more information about them. We find out that the online food ordering platform is based in Bengaluru, Karnataka, India.
- Next, we build Logistic Regression Models to predict the **Feedback** on orders and the **Output Status** of orders by consumers of the online food ordering app.

Building Logistic Regression Models

```
# Set seed for reproducibility of test
set.seed(123)

# Splitting data into 80% training and 20% testing sets
```

```r
train_index <- sample(1:nrow(online_foods), 0.8 * nrow(online_foods))
train_data <- online_foods[train_index, ]
test_data <- online_foods[-train_index, ]

# Logistic Regression Model for Feedback
feedback_model <- glm(Feedback ~ ., data = train_data, family = binomial)

# Summary of the model
summary(feedback_model)
```

Feedback Model

```
##
## Call:
## glm(formula = Feedback ~ ., family = binomial, data = train_data)
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             1.110e+03  4.193e+03   0.265
## Age                                    -7.839e-02  1.143e-01  -0.686
## GenderMale                             -5.264e-01  4.843e-01  -1.087
## Marital.StatusMarried                  -2.751e-01  6.868e-01  -0.401
## OccupationHouse wife                    1.934e+01  1.092e+03   0.018
## OccupationEmployed                     -1.588e+00  1.285e+00  -1.236
## Monthly.IncomeAverage Monthly Income    1.046e+00  1.239e+00   0.844
## Monthly.IncomeHigh Monthly Income       2.181e+00  1.364e+00   1.600
## Educational.QualificationsGraduate      2.735e+00  1.043e+00   2.621
## Educational.QualificationsHigher Education  2.651e+00  1.065e+00   2.490
## Family.size                             5.474e-02  1.643e-01   0.333
## latitude                               -9.786e+00  5.355e+00  -1.828
## longitude                               8.942e+00  4.416e+00   2.025
## Pin.code                               -2.995e-03  7.594e-03  -0.394
## OutputSuccessful                        3.588e+00  4.977e-01   7.210
##                                        Pr(>|z|)
## (Intercept)                             0.79128
## Age                                     0.49286
## GenderMale                              0.27709
## Marital.StatusMarried                   0.68874
## OccupationHouse wife                    0.98587
## OccupationEmployed                      0.21656
## Monthly.IncomeAverage Monthly Income    0.39856
## Monthly.IncomeHigh Monthly Income       0.10970
## Educational.QualificationsGraduate      0.00876 **
## Educational.QualificationsHigher Education  0.01278 *
## Family.size                             0.73894
## latitude                                0.06761 .
## longitude                               0.04289 *
## Pin.code                                0.69330
## OutputSuccessful                       5.61e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 273.53  on 299  degrees of freedom
## Residual deviance: 148.30  on 285  degrees of freedom
## AIC: 178.3
##
## Number of Fisher Scoring iterations: 16
```

- Firstly, we obtain a train-test split of the online-foods dataset. This dataset is split according to **80%** training data and **20%** testing data.
- Next, we model the Feedback variable based on every other attribute present in the dataset. To model the Feedback variable, we build a logistic regression model which predicts binary outcomes. The binary outcomes are "Positive" feedback or "Negative" feedback in this case. This is also why the family parameter of the model is set to be "binomial".
- Lastly, we obtain a summary of our model. The summary is as follows:
  - Generally, lower AIC values indicate better regression models. An AIC value of 178.3 indicates that the model fits reasonably well to the dataset.
  - The residual deviance is significantly lower than the null deviance, which suggests that the model accounts for a considerable amount of variation in the data.

```
# Predictions on the test set
feedback_predictions <- predict(feedback_model, newdata = test_data, type = "response")

# Convert probabilities to classes (Positive/Negative)
feedback_predictions_class <- ifelse(feedback_predictions > 0.5, "Positive", "Negative")

# Confusion Matrix for Feedback
feedback_confusion_matrix <- table(Actual = test_data$Feedback, Predicted = feedback_predictions_class)
print(feedback_confusion_matrix)
```

```
##            Predicted
## Actual      Negative Positive
##   Negative         7        8
##   Positive         4       57
```

- Next, we initialize predictions on the test data using the logistic regression model we built above. Thus, we predict values for the Feedback column using our own model.
- Then, we build a confusion matrix to compare our predicted values to the actual values that the Feedback column takes on.
  - We can see the number of false positive, false negative, true positive, and true negative observations based on our model.

```
# Extracting values from the confusion matrix
true_negative <- feedback_confusion_matrix[1, 1]
false_positive <- feedback_confusion_matrix[1, 2]
false_negative <- feedback_confusion_matrix[2, 1]
true_positive <- feedback_confusion_matrix[2, 2]

# Calculate accuracy
accuracy <- (true_positive + true_negative) / sum(feedback_confusion_matrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.8421053
```

```
# Calculate precision
precision <- true_positive / (true_positive + false_positive)
cat("Precision:", precision, "\n")
```

```
## Precision: 0.8769231
```

```r
# Calculate recall
recall <- true_positive / (true_positive + false_negative)
cat("Recall (Sensitivity):", recall, "\n")
```

```
## Recall (Sensitivity): 0.9344262
```

- Lastly, we calculate the **Accuracy**, **Precision**, and **Recall** of the model.
- **The Feedback model is 84% accurate, 88% precise, and has a recall of 93%.**

Output Model

- We perform a similar procedure to build and analyze the **Output Model** that models the output status of orders based on every other attribute in the dataset.

```r
# Logistic Regression Model for Output
output_model <- glm(Output ~ ., data = train_data, family = binomial)

# Summary of the model
summary(output_model)
```

```
##
## Call:
## glm(formula = Output ~ ., family = binomial, data = train_data)
##
## Coefficients:
##                                          Estimate Std. Error z value
## (Intercept)                             -9.337e+02  3.620e+03  -0.258
## Age                                     -1.121e-01  9.333e-02  -1.201
## GenderMale                               3.331e-01  4.093e-01   0.814
## Marital.StatusMarried                   -5.902e-01  5.887e-01  -1.003
## OccupationHouse wife                    -1.405e+00  1.352e+00  -1.039
## OccupationEmployed                      -1.539e+00  1.517e+00  -1.015
## Monthly.IncomeAverage Monthly Income     8.518e-01  1.488e+00   0.572
## Monthly.IncomeHigh Monthly Income        1.033e+00  1.561e+00   0.661
## Educational.QualificationsGraduate      -2.196e+00  9.561e-01  -2.297
## Educational.QualificationsHigher Education -1.971e+00  1.007e+00  -1.957
## Family.size                              3.336e-02  1.479e-01   0.226
## latitude                                -4.089e+00  4.523e+00  -0.904
## longitude                               -1.614e+00  3.738e+00  -0.432
## Pin.code                                 1.992e-03  6.515e-03   0.306
## FeedbackPositive                         3.368e+00  4.703e-01   7.161
##                                         Pr(>|z|)
## (Intercept)                               0.7965
## Age                                       0.2297
## GenderMale                                0.4157
## Marital.StatusMarried                     0.3160
## OccupationHouse wife                      0.2989
## OccupationEmployed                        0.3103
## Monthly.IncomeAverage Monthly Income      0.5670
## Monthly.IncomeHigh Monthly Income         0.5083
## Educational.QualificationsGraduate        0.0216 *
## Educational.QualificationsHigher Education   0.0503 .
## Family.size                               0.8216
## latitude                                  0.3660
## longitude                                 0.6658
```

```
## Pin.code                                    0.7598
## FeedbackPositive                           8.04e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.14  on 299  degrees of freedom
## Residual deviance: 195.38  on 285  degrees of freedom
## AIC: 225.38
##
## Number of Fisher Scoring iterations: 5
```

```
# Predictions on the test set
output_predictions <- predict(output_model, newdata = test_data, type = "response")

# Convert probabilities to classes (Successful/Unsuccessful)
output_predictions_class <- ifelse(output_predictions > 0.5, "Successful", "Unsuccessful")

# Confusion Matrix for Output
output_confusion_matrix <- table(Actual = test_data$Output, Predicted = output_predictions_class)
print(output_confusion_matrix)
```

```
##               Predicted
## Actual         Successful Unsuccessful
##   Unsuccessful          6            9
##   Successful           55            6
```

```
# Extracting values from the confusion matrix
true_negative <- output_confusion_matrix[1, 1]
false_positive <- output_confusion_matrix[1, 2]
false_negative <- output_confusion_matrix[2, 1]
true_positive <- output_confusion_matrix[2, 2]

# Calculate accuracy
accuracy <- (true_positive + true_negative) / sum(output_confusion_matrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.1578947
```

```
# Calculate precision
precision <- true_positive / (true_positive + false_positive)
cat("Precision:", precision, "\n")
```

```
## Precision: 0.4
```

```
# Calculate recall (sensitivity)
recall <- true_positive / (true_positive + false_negative)
cat("Recall (Sensitivity):", recall, "\n")
```

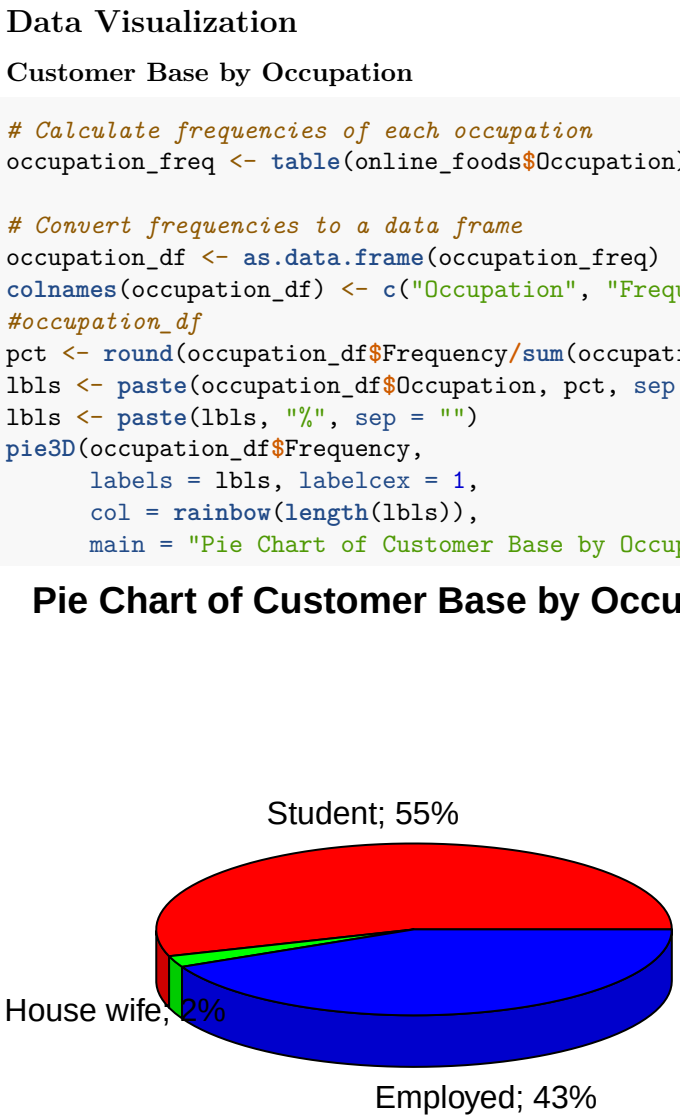
```
## Recall (Sensitivity): 0.09836066
```

- Upon evaluating the Output model, it's noteworthy that the model exhibits an accuracy of only **16%**, precision of **40%**, and a recall of **10%**.
- This could be due to a class imbalance issue in the dataset, which occurs when one class (e.g., "Successful") is much more prevalent than the other. This can lead to biased model performance metrics.
- Clearly, there is substantial room for improvement in this model, suggesting avenues for future enhancement.

Data Visualization

Customer Base by Occupation

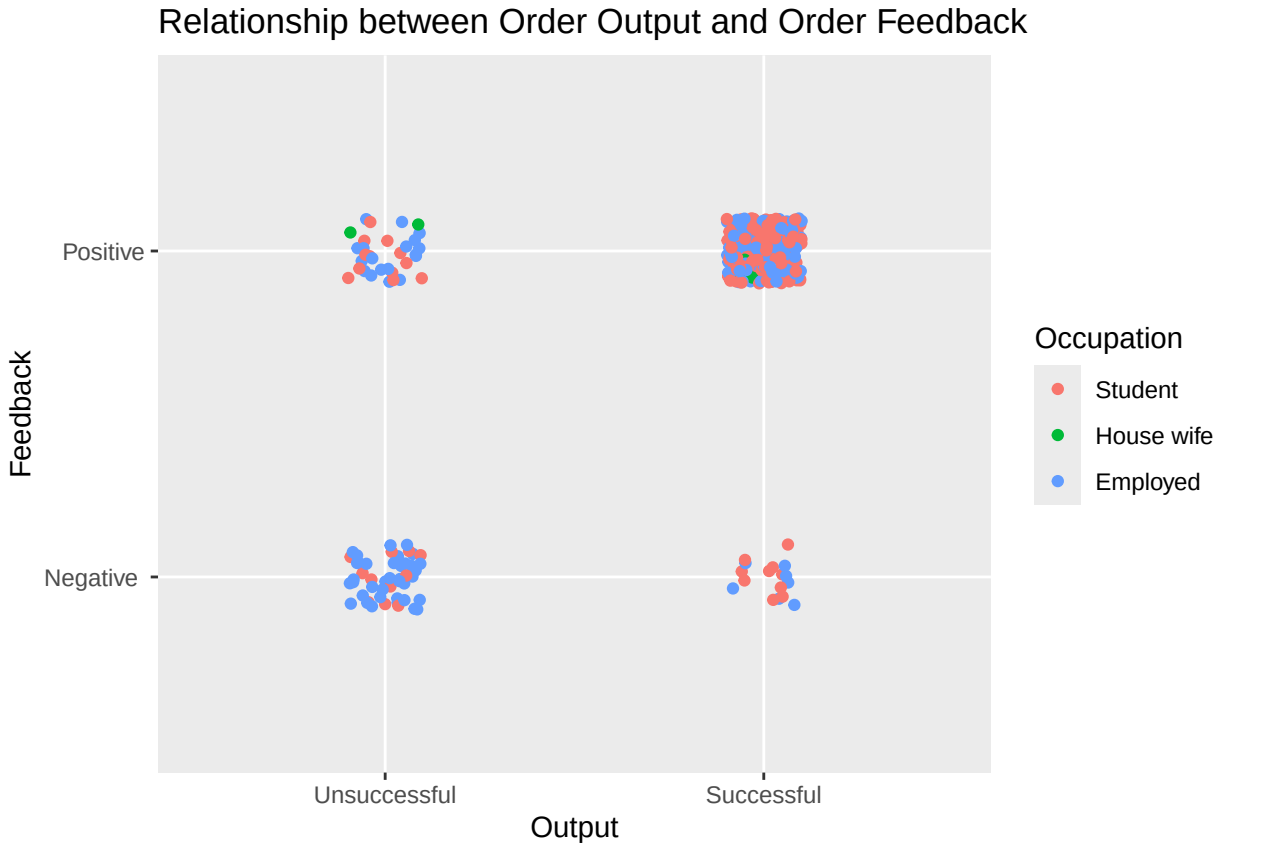
```
# Calculate frequencies of each occupation
occupation_freq <- table(online_foods$Occupation)

# Convert frequencies to a data frame
occupation_df <- as.data.frame(occupation_freq)
colnames(occupation_df) <- c("Occupation", "Frequency")
#occupation_df
pct <- round(occupation_df$Frequency/sum(occupation_df$Frequency)*100)
lbls <- paste(occupation_df$Occupation, pct, sep = "; ")
lbls <- paste(lbls, "%", sep = "")
pie3D(occupation_df$Frequency,
      labels = lbls, labelcex = 1,
      col = rainbow(length(lbls)),
      main = "Pie Chart of Customer Base by Occupation")
```



- In this analysis, we utilize a three-dimensional pie chart to visually represent the distribution of the sample according to their occupations.
- The findings reveal that a significant portion (**55%**) of the consumer base of the online food ordering app comprises students, while housewives constitute a much smaller proportion (**2%**).
- Notably, housewives are distinctly categorized as an occupation group within this dataset, possibly indicating regional biases specific to Bengaluru, India.

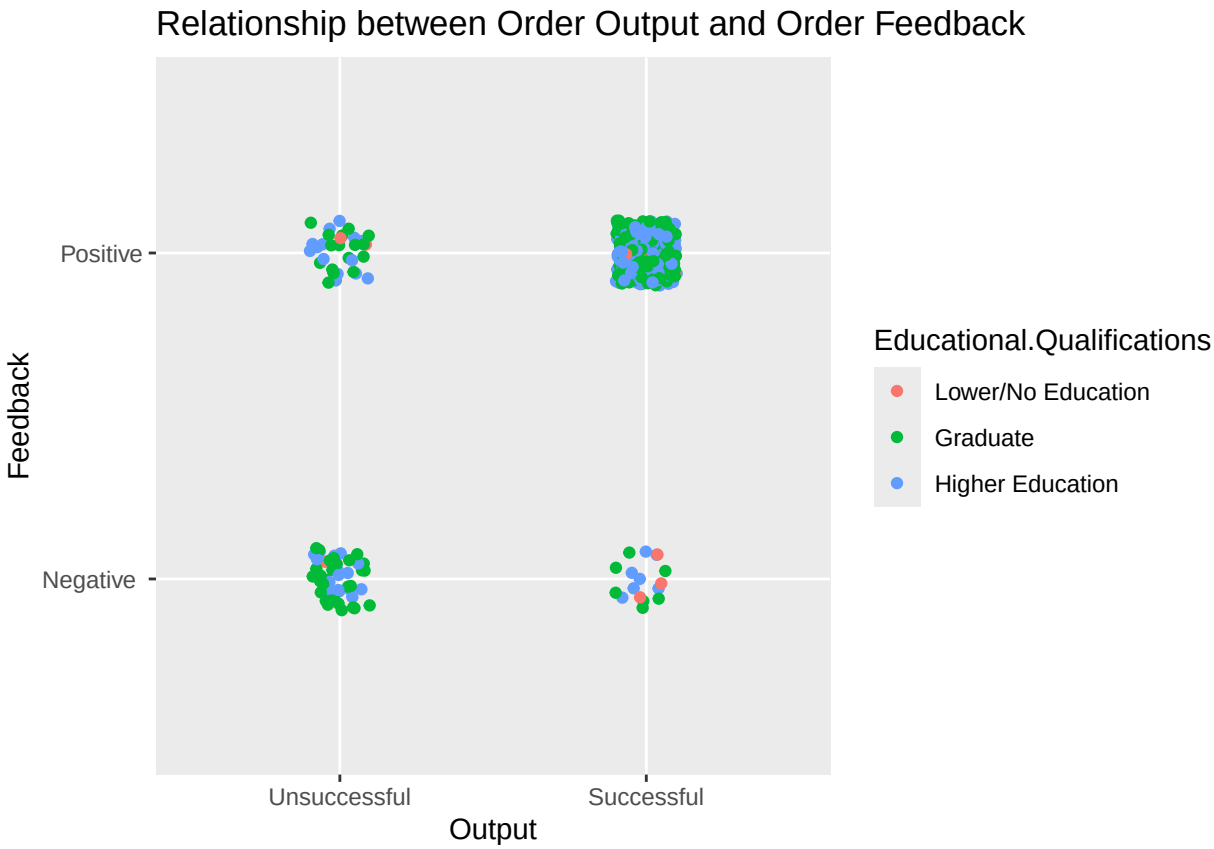
Relationship between Order Output and Order Feedback Sentiment

```
ggplot(online_foods, aes(x = Output, y = Feedback)) +
  geom_point(aes(color = Occupation), position = position_jitter(width = 0.1, height = 0.1)) +
  labs(title = "Relationship between Order Output and Order Feedback",
       x = "Output", y = "Feedback")
```



- In this scatter plot (jittered to reduce noise and overlap among points), we observe the relationship between customer feedback sentiment and the success of their orders.
- We can see that the majority of data points are clustered in the "positive" feedback and "successful" output region, indicating that most orders which are successfully completed have positive feedback reviews. This is symmetric for the exact opposite scenario, where orders that are unsuccessful receive negative reviews.
- Interestingly, a significant portion of unsuccessful orders still receive positive reviews, implying a level of compassion and understanding among Bengaluru, India residents when utilizing online food ordering platforms.
- Furthermore, students are primarily clustered in the "Positive, Successful" region, indicating their tendency to leave positive reviews for successful orders. Conversely, employees tend to leave the most negative reviews for unsuccessful orders.

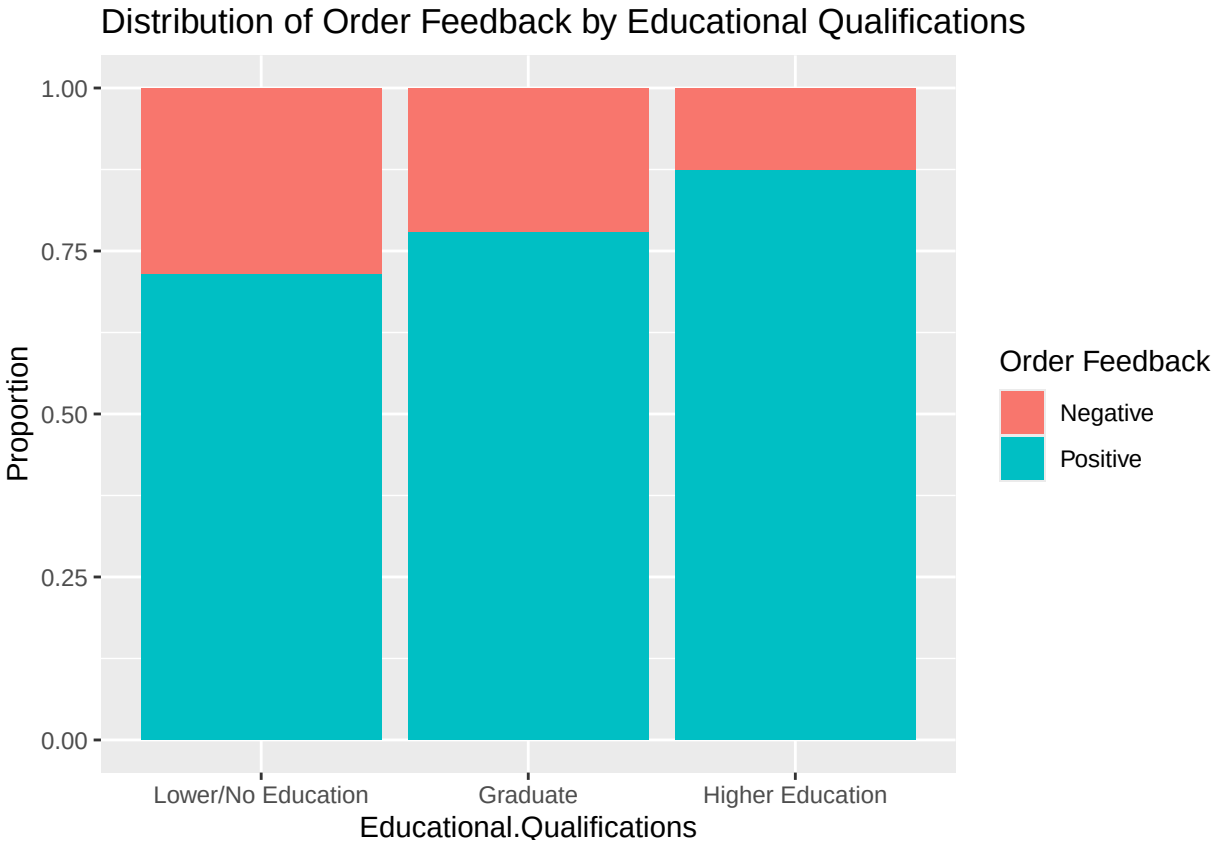
```
ggplot(online_foods, aes(x = Output, y = Feedback)) +
  geom_point(aes(color = Educational.Qualifications),
             position = position_jitter(width = 0.1, height = 0.1)) +
  labs(title = "Relationship between Order Output and Order Feedback",
       x = "Output", y = "Feedback")
```



- Likewise, the plot above is recreated, with the distinction that the points are now grouped based on the educational qualifications of the customers.

Distribution of Order Feedback Sentiment by Educational Qualification

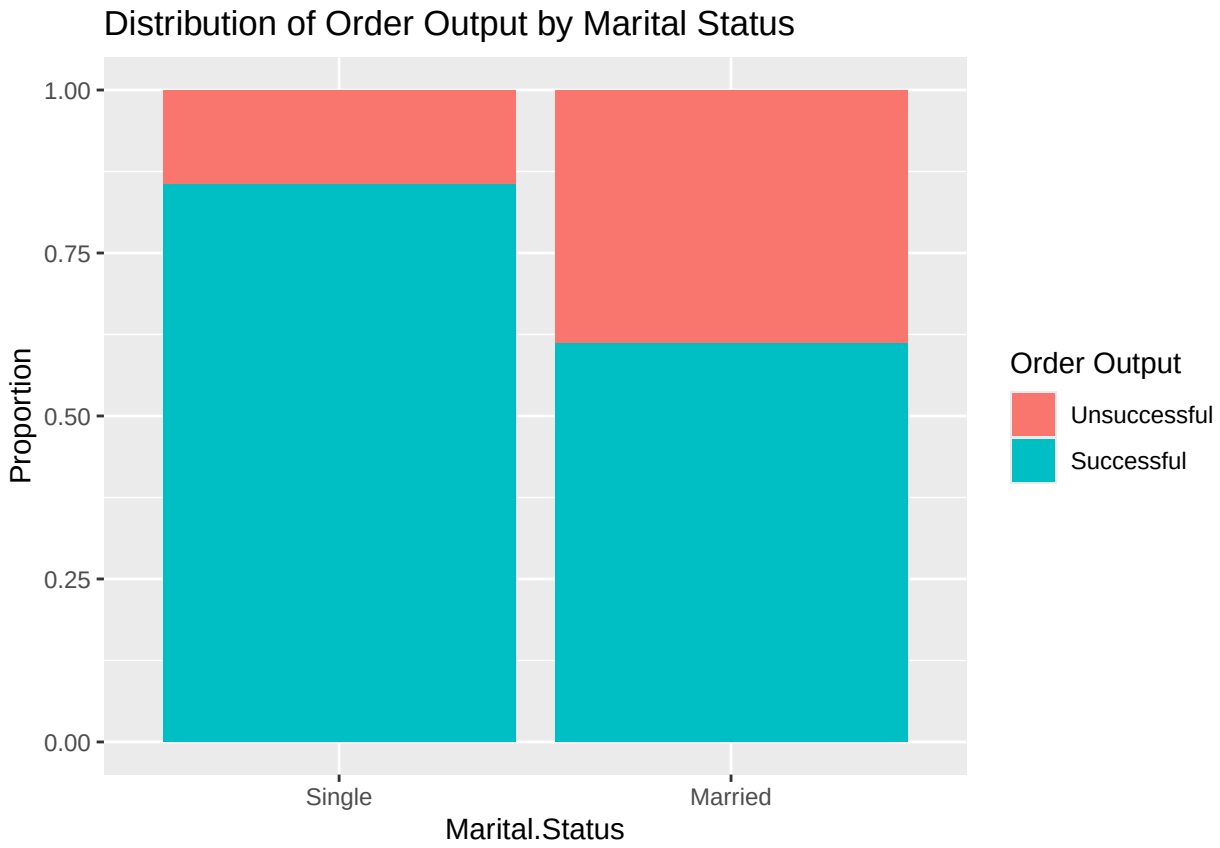
```
ggplot(online_foods, aes(x = Educational.Qualifications, fill = Feedback)) +
  geom_bar(position = "fill") +
  labs(title = "Distribution of Order Feedback by Educational Qualifications", y = "Proportion", fill =
```



- Individuals with higher educational qualifications demonstrate a propensity to provide positive feedback on their orders significantly more often compared to those with lower levels of education.

Distribution of Order Output by Marital Status

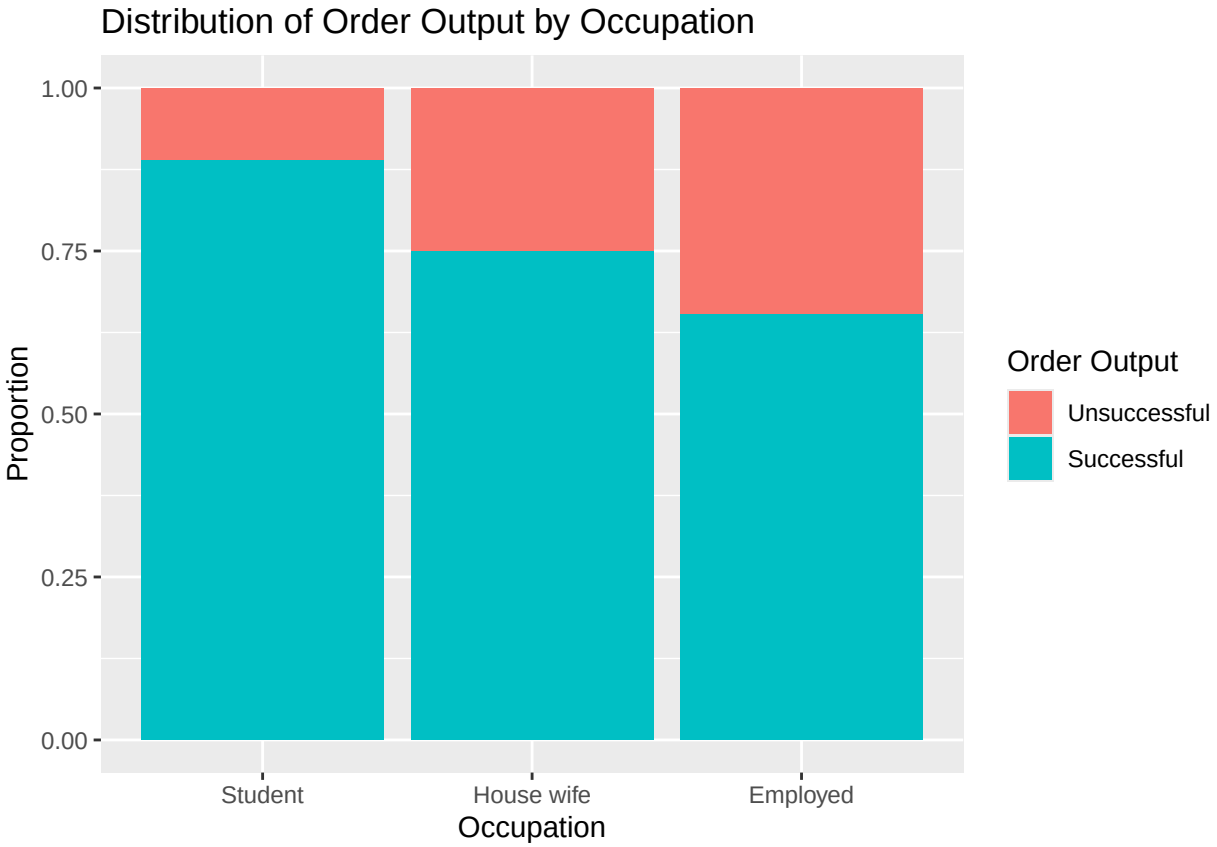
```
ggplot(online_foods, aes(x = Marital.Status, fill = Output)) +
  geom_bar(position = "fill") +
  labs(title = "Distribution of Order Output by Marital Status", y = "Proportion", fill = "Order Output
```



- This plot suggests that married individuals tend to encounter a notably higher proportion of unsuccessful orders compared to single individuals. This trend may correlate with the observation that students in Bengaluru tend to leave more positive reviews on orders. This connection likely stems from the fact that students, who are typically single, experience a higher proportion of successful order transactions. (**based on regional biases of India, assuming students are generally single**)

Distribution of Order Output by Occupation

```
ggplot(online_foods, aes(x = Occupation, fill = Output)) +
  geom_bar(position = "fill") +
  labs(title = "Distribution of Order Output by Occupation", y = "Proportion", fill = "Order Output")
```



- The bar plot indicates a significant disparity in successful order experiences among different demographic groups. Specifically, students receive a substantially higher number of successful orders from the online food ordering platform compared to housewives and employed individuals. Conversely, employed individuals exhibit the lowest proportion of successful order experiences relative to other groups.

Conclusion

- Throughout this project, we've effectively executed data cleaning, analysis, and visualization tasks to glean valuable insights into the ordering behaviors of Bengaluru, India residents on an online food ordering platform. By systematically refining the dataset, exploring its nuances through analytical techniques, and crafting insightful visualizations, we've uncovered significant patterns and trends shaping consumer preferences and habits in the local online food delivery landscape.

Dataset link

- Kaggle link to dataset source: dataset