CIND 820 Capstone Project Abstract: In-Vehicle Coupon Recommendation

Sucheta Sikdar (ID: 501074722)

Supervisor: Dr. Ceni Babaoglu

Toronto Metropolitan University

Date of Submission: 2024-10-21

Toronto Metropolitan University

Table of Contents

Abstract

Coupons are a great way for customers to save money on their purchases. It is also a great way for business to attract customers. Coupons create a win-win situation for both companies and customers and hence by offering a correct coupon to users can lead to users to become frequent customers (Niralidedaniya, 2023). If businesses can find the right customers who will use their coupons, then it will help businesses. It will also be interesting to predict what type of coupon will be accepted by a customer based on various attributes about the customer. Luckily, there exists a publicly available dataset called In-Vehicle Coupon Recommendation at UCI Machine Learning Repository from 2020 which describes different driving scenarios of multiple clients such as destination, time, coupon, expiration, gender, age, marital status, whether they have children, education, occupation, income, car, the number of times they go to the bar per month, the number of times they go to a coffee shop per month, the number of times that they buy take away food per month, if customer's average expense per person at restaurants is less than 20 dollars a month, if customer's average expense per person at a restaurant is between 20 dollars to 50 dollars per month, driving distance to the restaurant/bar for using the coupon is greater than 15 minutes, driving distance to the restaurant/bar for using the coupon is greater than 25 minutes, whether the restaurant/bar is in the same direction as destination, whether the restaurant/bar is in the opposite direction as destination, whether the coupon is accepted (*UCI Machine Learning Repository*, 2020).

When printing the info of the dataset in VSCode using Python, the following was observed:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64
 4   time                  12684 non-null  object
 5   coupon                12684 non-null  object
 6   expiration            12684 non-null  object
 7   gender                12684 non-null  object
 8   age                   12684 non-null  object
 9   maritalStatus         12684 non-null  object
 10  has_children          12684 non-null  int64
 11  education             12684 non-null  object
 12  occupation            12684 non-null  object
 13  income                12684 non-null  object
 14  car                   108 non-null    object
 15  Bar                   12577 non-null  object
 16  CoffeeHouse           12467 non-null  object
 17  CarryAway             12533 non-null  object
 18  RestaurantLessThan20  12554 non-null  object
 19  Restaurant20To50      12495 non-null  object
 20  toCoupon_GEQ5min      12684 non-null  int64
 21  toCoupon_GEQ15min     12684 non-null  int64
 22  toCoupon_GEQ25min     12684 non-null  int64
 23  direction_same        12684 non-null  int64
 24  direction_opp         12684 non-null  int64
 25  Y                     12684 non-null  int64
dtypes: int64(8), object(18)
memory usage: 2.5+ MB
None
PS C:\Users\suche\OneDrive\Desktop\Sucheta_Sikdar_CIND_820>
```

Figure 1: Info on the In-Vehicle Coupon Recommendation Dataset

As seen in Figure 1, the dataset has 12684 records. The dataset has 25 features and 1 target column. Some of the data like marital status and gender are categorical and some of the data like age and temperature are numerical. Using code provided by Niralidedaniya (2023), it was found that the target classes are partially balanced. If the target classes were highly unbalanced, then this dataset could not be used because the results of supervised learning algorithms used to make predictions would skew towards the class with the class with higher percentage of records.

```
Accepted coupon: 7210 56.843 %
Rejected coupon: 5474 43.157 %
PS C:\Users\suche\OneDrive\Desktop\Sucheta_Sikdar_CIND_820>
```

Figure 2: Distribution of Target Classes

It should also be noted that this dataset comes with many missing values. Hence, this dataset requires preprocessing before it can be analyzed with machine learning algorithms. Using the code provided by Niralidedaniya (2023), we can see the features which have missing values.

```
                      missing_count  missing_percentage
destination                       0            0.000000
passanger                         0            0.000000
weather                           0            0.000000
temperature                       0            0.000000
time                              0            0.000000
coupon                            0            0.000000
expiration                        0            0.000000
gender                            0            0.000000
age                               0            0.000000
maritalStatus                     0            0.000000
has_children                      0            0.000000
education                         0            0.000000
occupation                        0            0.000000
income                            0            0.000000
car                           12576           99.148534
Bar                             107            0.843582
CoffeeHouse                     217            1.710817
CarryAway                       151            1.190476
RestaurantLessThan20            130            1.024913
Restaurant20To50                189            1.490066
toCoupon_GEQ5min                  0            0.000000
toCoupon_GEQ15min                 0            0.000000
toCoupon_GEQ25min                 0            0.000000
direction_same                    0            0.000000
direction_opp                     0            0.000000
Y                                 0            0.000000
```

Figure 3: Distribution of missing values in the dataset

The objective of this project will be to:

- Find the best predictive classification algorithm for the In-Vehicle Coupon Recommendation dataset (2020) after evaluation of various supervised learning classification algorithms introduced to us in *CMTH 642 – Data Analytics: Advanced Methods* like Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbours (k-NN) on the dataset.
- Using a correlation matrix find out which attributes are highly correlated to the target of the customer accepting or rejecting a coupon.
- Find whether we can attain a dataset with fewer dimensions using these 3 methods: Stepwise Regression, Forward Feature Selection and Backward Feature Elimination methods learnt by us in *CMTH 642 – Data Analytics: Advanced Methods*.
- Find the limitations of this dataset.

Data analysis for this project will be done using Python. Pandas and numpy libraries will be used. Seaborn and Matplotlib libraries will be used for visualizations. Supervised learning techniques will require the scikit-learn library (*Supervised Learning*, n.d.). Evaluation of various classification algorithms will be done by comparing evaluation metrics like the Accuracy, Precision and Area under the Curve (AUC) of each algorithm.

GitHub Link: https://github.com/suchetasikdar1/CIND820/

Introduction

Coupons are a great way for customers to save money on their purchases and feel special that they are getting a discount. Businesses can attract customers with coupons. If businesses can find the right customers who will use their coupons, then it will help businesses survive and grow. Businesses can then retain existing customers and attract new customers. If businesses know beforehand as to which customers to target for their coupons, then it will save them both money and effort in marketing and sending coupons (Ahmed et al., 2024).

It will also be interesting to predict what type of coupon will be accepted by a customer based on various attributes about the customer. There exists a publicly available dataset called In-Vehicle Coupon Recommendation at UCI Machine Learning Repository from 2020 which describes different driving scenarios of multiple clients and whether the coupon is accepted (*UCI Machine Learning Repository*, 2020).

This project will use the knowledge that has been gained in previous courses of the Data Analytics, Big Data, and Predictive Analytics Certificate Program taught at Toronto Metropolitan University. What has been learned can be implemented in this project.

The objective of this project will be to:

- Find the best predictive classification algorithm for the In-Vehicle Coupon Recommendation dataset (2020) after evaluation of various supervised learning classification algorithms introduced to us in *CMTH 642 – Data Analytics: Advanced Methods* like Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbours (k-NN) on the dataset. What has been learned will be implemented.

- Using a correlation matrix find out which attributes are highly correlated to the target of the customer accepting or rejecting a coupon. A visual of a correlation matrix will be very effective. **Since most of the features are categorical, the discreet values will be converted to corresponding numerical values using encoding.**
- Find whether we can attain a dataset with fewer dimensions using these 3 methods: Stepwise Regression, Forward Feature Selection and Backward Feature Elimination methods learnt by us in *CMTH 642 – Data Analytics: Advanced Methods*. This will help eliminate some noise. **Dimensionality reduction is the major contribution of this project to this dataset.**
- Find the limitations of this dataset. The flaws of this dataset need to be explored.

There is some existing and related work done on this topic already. Exploratory data analysis to get a better picture of our data is also necessary.

Literature Review

The original data was collected by Wang et al. (2017) using Amazon Mechanical Turk about users interacting with a mobile recommendation system. They used rule set models that are used for classification and decision making to understand and predict consumers' response to coupons in different contexts (Wang et al., 2017). Wang et al. (2017) used Bayesian approach and Disjunctive Normal Form classifiers. An example is that if a combination of a subset of input is met, then the output is satisfied. Wang et al. (2017) compared performance in accuracy by calculating and comparing Area under the ROC curve for their deduced final two Bayesian Rule Sets with other machine learning methods including different types of decision tree algorithms. Wang et al. (2017) compared with C4.5, CART, Lasso, RIPPER and TopK. Unlike other machine learning algorithms, a Bayesian approach looks at previous choices. Wang et al. (2017) deduced that that their Bayesian approach had competitive performance. Their methods are more complex compared to the simpler supervised learning classification algorithms introduced to us in *CMTH 642 – Data Analytics: Advanced Methods.* It is worth checking how this dataset performs against simpler supervised learning classification algorithms introduced to us in *CMTH 642 – Data Analytics: Advanced Methods.* It is possible that using simpler supervised learning classification algorithms like Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbours (k-NN) might yield similar competitive results too.

Same dataset was used by Niralidedaniya (2023). A lot of the data understanding, Data preparation and Exploratory Data Analysis on the dataset was already done by Niralidedaniya (2023). Niralidedaniya (2023) experimented with below thirteen machine learning classification models with hyper parameter tuning:

1. Logistic Regression

2. K-Nearest Neighbor
3. Decision Tree
4. Support Vector Classification with rbf kernel
5. LinearSVC
6. Gaussian Naive Bayes
7. Random Forest
8. GBDT
9. Bagging Classifier
10. AdaBoost Classifier
11. Gradient Boosting Classifier
12. ExtraTrees Classifier
13. HistGradientBoosting Classifier

Niralidedaniya (2023) compared the test log loss and test AUC score of each of the thirteen ML models with different encoded data matrices to find the best model and best encoding method with their best hyper parameter. Niralidedaniya (2023) found that Ordinal Encoding and One Hot Encoding perform best than other encoding techniques. It was observed that XGB Classifier, Support Vector Classification, Hist Gradient Boosting, and Random Forest Classifier models perform best than other models. Bagging, AdaBoost, Gradient Boosting, and ExtraTrees Classifier models perform best but they are overfitted (Niralidedaniya, 2023). Niralidedaniya (2023) stated that their Training AUC Score is very high with a value of 1, and a training log loss is almost 0. Basic classification models like Logistic Regression, K-Nearest Neighbor, Decision Tree, LinearSVC, and Gaussian Naive Bayes models didn't do well for this problem (Niralidedaniya, 2023). Niralidedaniya (2023) took classification up a notch by the use of stacking classifier on the four models: XGB Classifier, Support Vector Classification, Hist Gradient Boosting, and Random Forest Classifier with their best parameters. It would still be interesting to see and compare how the basic classification algorithms introduced to us in *CMTH 642 – Data Analytics: Advanced Methods* like Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbours (k-NN) perform on the dataset. A generated visual correlation matrix will be insightful. A dataset with fewer dimensions can be attainted using these 3 methods: Stepwise Regression, Forward Feature Selection and Backward Feature Elimination methods learnt by us in *CMTH 642 – Data Analytics: Advanced Methods*. It will also be useful to explore the limitations of this dataset.

Depari et al. (2022) used the same dataset. Depari et al. (2022) used RapidMiner 9.10.001 tool in predicting customer’s responses to in-vehicle coupon recommendations. Depari et al. (2022) compared the accuracy percentage, class precision, and execution time for three algorithms (Random Forest, Naïve Bayes and Decision Tree) on the dataset after doing a detailed descriptive analysis of the data. The descriptive analysis was very insightful about the dataset. For example, Depari et al. (2022) found that the data contained mostly married females who like to travel alone on a sunny day around 6 PM. Most of them have attended college, yet didn’t graduate (Depari et al., 2022). For those who have an occupation, it states that most of them earn an income of around $25000 - $37499 (Depari et al., 2022). It was also mentioned that the destination is mostly the No Urgent Place such as Coffee House, which provides a coupon that expires in one day (Depari et al., 2022). It will be interesting to find more inferences from the data with the help of plots for each feature. This will help with finding the limitations of the dataset provided. Depari et al. (2022) found that predictive analytics results showed that random forest achieved the highest accuracy with 77.65% overall accuracy percentage, yet required the most time to process. However, the decision tree algorithm acquired the highest confidence level of 0.750 for prescriptive analysis (Depari et al., 2022). Prescriptive analytics is data analysis that is used to explain and determine what is the next action plan and why should they do it (Depari et al., 2022). It will be interesting to compare with 2 more algorithms: Logistic Regression and k-Nearest Neighbours (k-NN) used on the dataset. There was no dimension reduction done by Depari et al. (2022).

This dataset was also used by Atiq et al. (2022). This dataset comes with missing values. This could cause problems in the prediction analysis. To circumvent this problem Atiq et al. (2022) analysed the impact of four different imputation techniques (Frequent value, mean, KNN, MICE) to replace the missing values and use them to create prediction models. Atiq et al. (2022) then applied six classifier algorithms (Naïve Bayes, Deep Learning, Logistic Regression, Decision Tree, Random Forest, and Gradient Boosted Tree). Atiq et al. (2022) found that KNN imputation with Deep Learning classifier gave the most accurate outcome. Atiq et al. (2022) applied SMOTE (Synthetic Minority Oversampling Technique) for oversampling the dataset in order to have positive and negative instances divided into 50% / 50%. This led to a perfectly balanced dataset for target value. Below is a table by Atiq et al. (2022) that is used to show Accuracy of Classifiers on Actual and Oversampled Dataset using various imputation Techniques. **These pre-existing findings will be useful in dealing with the missing values in the dataset.** Also, there was no dimension reduction done by Atiq et al. (2022).

Imputation Technique	Classifier	Accuracy (Without Over Sampling)	Accuracy (With Over Sampling)
Frequent Value	Random Forest	74.2	76.1
	Decision Tree	70.6	69.5
	Logistic Regression	68.433	69.6
	Gradient Boosted Tree	74.338	75.4
	Naive Bayes	66.253	67.1
	Deep Learning	75.73	60.4
Mean Imputation	Random Forest	74.7	73.1
	Decision Tree	68.82	67.3
	Logistic Regression	69.2	67.5
	Gradient Boosted Tree	73.4	73.1
	Naive Bayes	65.9	68.3
	Deep Learning	57.40	100
KNN	Random Forest	74.32	72.3
	Decision Tree	71.91	68.1
	Logistic Regression	68.8	66.2
	Gradient Boosted Tree	73.2	72.4
	Naive Bayes	65.6	64.9
	Deep Learning	57.16	83.27
MICE	Random Forest	74.3	71.9
	Decision Tree	70.2	67.9
	Logistic Regression	66.4	68.1
	Gradient Boosted Tree	73.2	72.9
	Naive Bayes	66.6	65.3
	Deep Learning	100	80.61

Table 1: Accuracy of Classifiers on Actual and Over Sampled Dataset provided by Atiq et al. (2022).

Patil et al. (2019) used a different dataset for E-coupons to predict coupon usage behaviour. It is interesting that to train the gradient boosting classifier, they used 45 “train periods” that simulated the test timing (Patil et al., 2019). When comparing with other algorithms like logistic regression, SVM, random forest, neural networks, Patil et al. (2019) found that gradient boosting was the single best classifier. By having train periods, Patil et al. (2019) made interesting observations such as the more the coupon is viewed, the probability to buy using the coupon code increases. Patil et al. (2019) also observed that customers tend to purchase the same coupon over and over again. The dataset for in-vehicle coupon response deficient in data over a periodic basis to help uncover such patterns. This is a limitation of the in-vehicle coupon dataset.

Ahmed et al. (2024) used a different dataset, Dunnhumby data for a particular grocery retail company. Ahmed et al. (2024) proposed two different models: one for predicting customer churn and the other for coupon redemption model and both those models used XGBoost Classifier Model. As seen in earlier papers, it is good to use a variety of models for comparison. However, Ahmed et al. used only XGBoost Classifier Model. Ahmed et al. (2024) also stated that their dataset was not large enough and therefore it led to uncertainty about the model performance on a larger dataset in the future.

Given all the existing data analysis that has been done on coupon redemption, it is obvious that it is crucial for businesses to help them grow and survive. When we attain a dataset with fewer dimensions using these 3 methods: Stepwise Regression, Forward Feature Selection and Backward Feature Elimination methods learnt by us in *CMTH 642 – Data Analytics: Advanced Methods*, we can have more concise information to be used for decision making and not have noise. **Dimensionality reduction is the major contribution of this project to this dataset.** A visual correlation matrix will show which attributes are highly correlated to the target of the customer accepting or rejecting a coupon. A visual correlation matrix is a very effective way to display a lot of information. This project also gives an opportunity to compare and find the best predictive classification algorithm for In-Vehicle Coupon Recommendation dataset (2020) after evaluation of various supervised learning classification algorithms introduced to us in *CMTH 642 – Data Analytics: Advanced Methods* like Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbours (k-NN) on the dataset. There could be so many other factors that effect customer coupon redemption. Therefore, we need to also explore the limitations of this dataset. Hence, it is worth proceeding with this project.

Data Description

The dataset is in csv format. It is a publicly available dataset called In-Vehicle Coupon Recommendation at UCI Machine Learning Repository from 2020 which describes different driving scenarios of multiple clients and whether the coupon is accepted (*UCI Machine Learning Repository*, 2020). The dataset is large with 12684 records and 25 features. The below figure provides some information on the dataset:

```
in_vehicle_coupon_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 26 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64
 4   time                  12684 non-null  object
 5   coupon                12684 non-null  object
 6   expiration            12684 non-null  object
 7   gender                12684 non-null  object
 8   age                   12684 non-null  object
 9   maritalStatus         12684 non-null  object
 10  has_children          12684 non-null  int64
 11  education             12684 non-null  object
 12  occupation            12684 non-null  object
 13  income                12684 non-null  object
 14  car                   108 non-null    object
 15  Bar                   12577 non-null  object
 16  CoffeeHouse           12467 non-null  object
 17  CarryAway             12533 non-null  object
 18  RestaurantLessThan20  12554 non-null  object
 19  Restaurant20To50      12495 non-null  object
 20  toCoupon_GEQ5min      12684 non-null  int64
 21  toCoupon_GEQ15min     12684 non-null  int64
 22  toCoupon_GEQ25min     12684 non-null  int64
 23  direction_same        12684 non-null  int64
 24  direction_opp         12684 non-null  int64
 25  Y                     12684 non-null  int64
dtypes: int64(8), object(18)
memory usage: 2.5+ MB
```

Figure 4: In-Vehicle Coupon Recommendation dataset information

Feature	Feature Description	Possible Values
destination	Destination	No Urgent Place, Home, Work
passenger	Passenger	Alone, Friend(s), Kid(s), Partner
weather	Weather Type	Sunny, Rainy, Snowy
temperature	Temperature	Numerical value
time	Time	2PM, 10AM, 6PM, 7AM, 10PM
coupon	Coupon type	Restaurant (<20), Coffee House, Carry out & Take away, Bar, Restaurant (20-50)
expiration	Expiration	1d, 2h
gender	Gender	Female, Male
age	Age	below21, 50plus, 36, 41, 31, 26, 46, 21
maritalStatus	Marital Status	Unmarried partner, Single, Married partner, Divorced, Widowed
has_children	Whether they have children	Numerical value – 0 or 1
education	Education	Some college – no degree, Bachelors degree, Associates degree, High School

		Graduate, Graduate degree (Masters or Doctorate), some High School
occupation	Occupation	Unemployed, Architecture & Engineering, Student, Education & Training & Library, Healthcare Support, Healthcare Practitioners & Technical, Sales & Related, Management, Arts Design Entertainment Sports & Media, Computer & Mathematical, Life Physical Social Science, Personal Care & Service, Community & Social Services, Office & Administrative Support, Construction & Extraction, Legal, Retired, Installation Maintenance & Repair, Transportation & Material Moving, Business & Financial, Protective Service,

		Food Preparation & Serving Related, Production Occupations, Building & Grounds Cleaning & Maintenance, Farming Fishing & Forestry
income	Income	37500-49999, 62500-74999, 12500 – 24999, 75000-87499, 50000 – 62499, 25000 – 37499, $100000 or More, 87500 – 99999, Less than $12500
car	Car	Scooter and motorcycle, crossover, Mazda5, do not drive, car that is too old to install Onstar
bar	The number of times they go to the bar per month	Never, less 1, 1~3, gt8, 4~8
CoffeeHouse	The number of times they go to a coffee shop per month	Never, less1, 4~8, 1~3, gt8
CarryAway	The number of times that they buy take away food per month	Never, less1, 4~8, 1~3, gt8

RestaurantLessThan20	if customer's average expense per person at restaurants is less than 20 dollars a month	Never, less1, 4~8, 1~3, gt8
Restaurant20To50	if customer's average expense per person at a restaurant is between 20 dollars to 50 dollars per month	Never, less1, 4~8, 1~3, gt8
toCoupon_GEQ5min	driving distance to the restaurant/bar for using the coupon is greater than 5 minutes	Numerical value
toCoupon_GEQ15min	driving distance to the restaurant/bar for using the coupon is greater than 15 minutes	Numerical value
toCoupon_GEQ25min	driving distance to the restaurant/bar for using the coupon is greater than 25 minutes	Numerical value
direction_same	whether the restaurant/bar is in the same direction as destination	Numerical value
direction_opp	whether the restaurant/bar is in the opposite direction as destination,	Numerical value

Y	whether the coupon is accepted. This is the target.	0 or 1. 1 if coupon is accepted. 0 if coupon is rejected.

Table 2: Feature Description in In Vehicle Coupon Redemption dataset

The below plots for distribution of categorical value features were generated with the help of code by Inyama (2023).

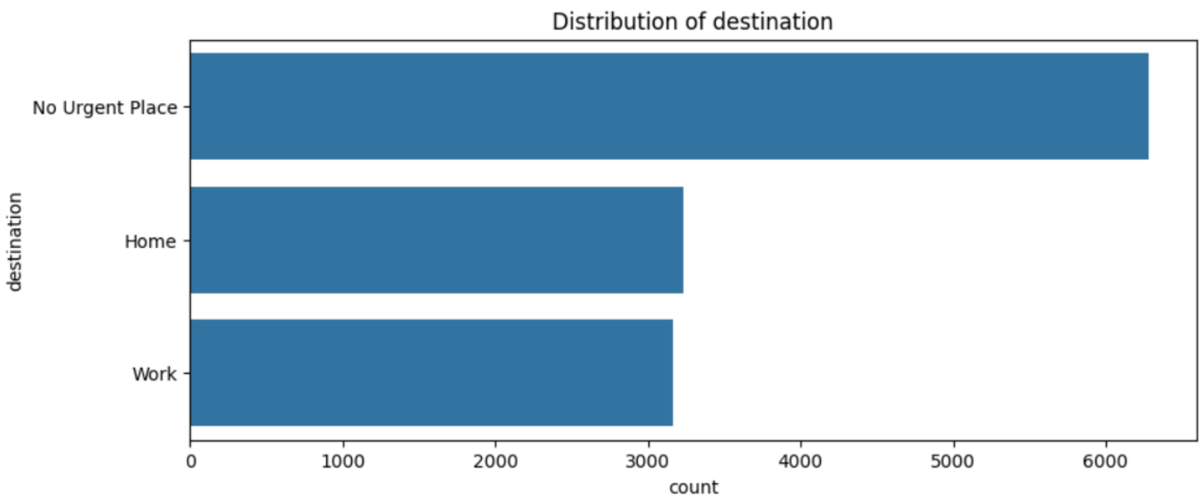

Figure 5: Distribution of destination feature

Observation: Most people's destination is not an urgent place

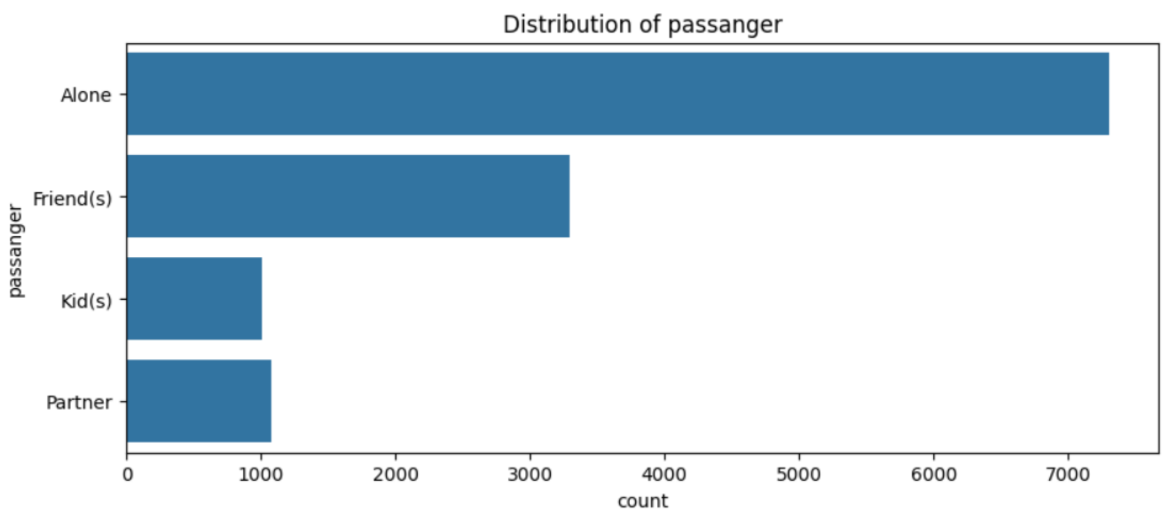

Figure 6: Distribution of passenger feature

Observation: Most people travel alone

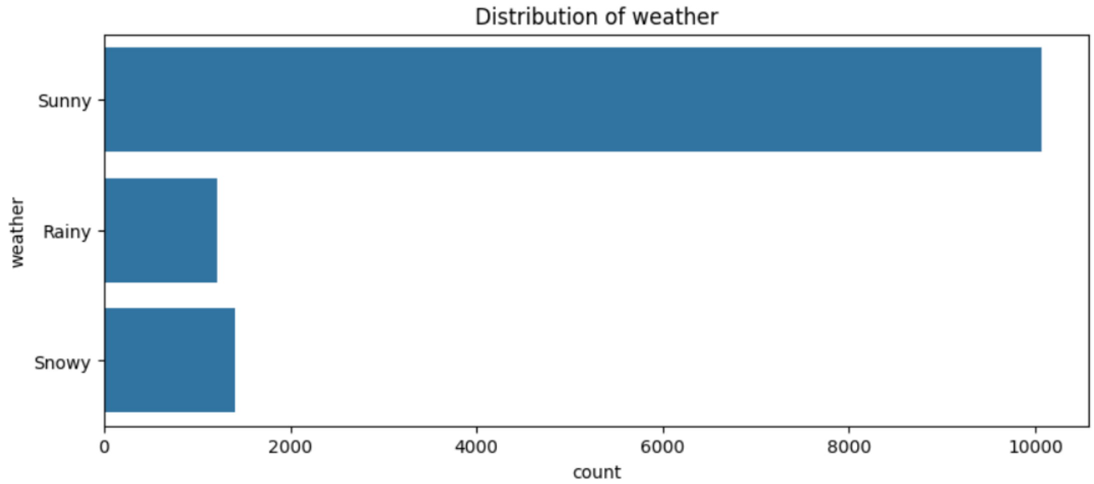

Figure 7: Distribution of weather feature

Observation: Most people travel on a sunny day

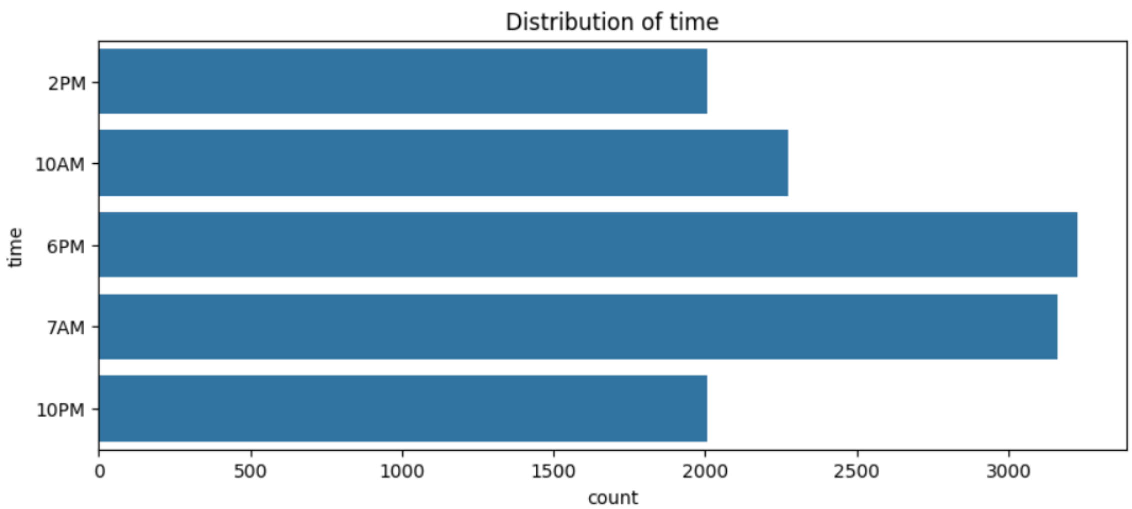

Figure 8: Distribution of time feature

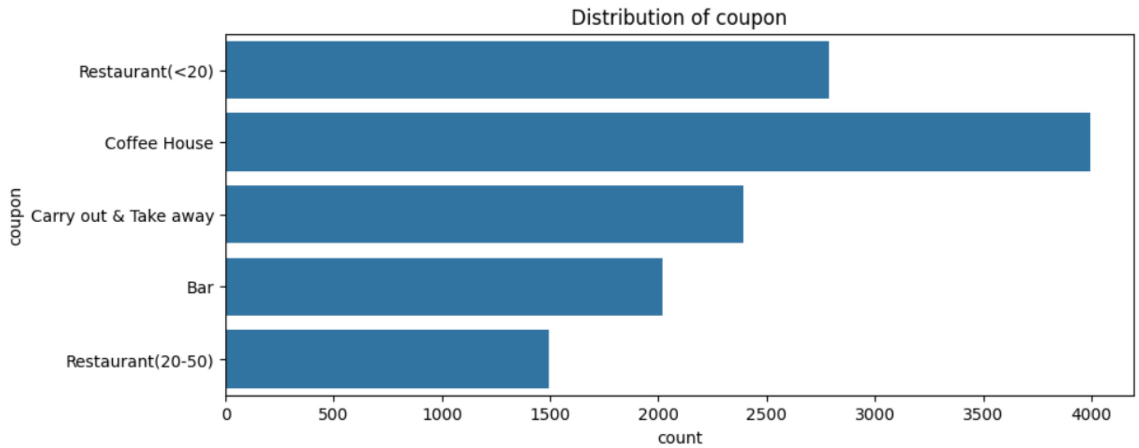

Figure 9: Distribution of coupon feature

Observation: Most people go to Coffee House

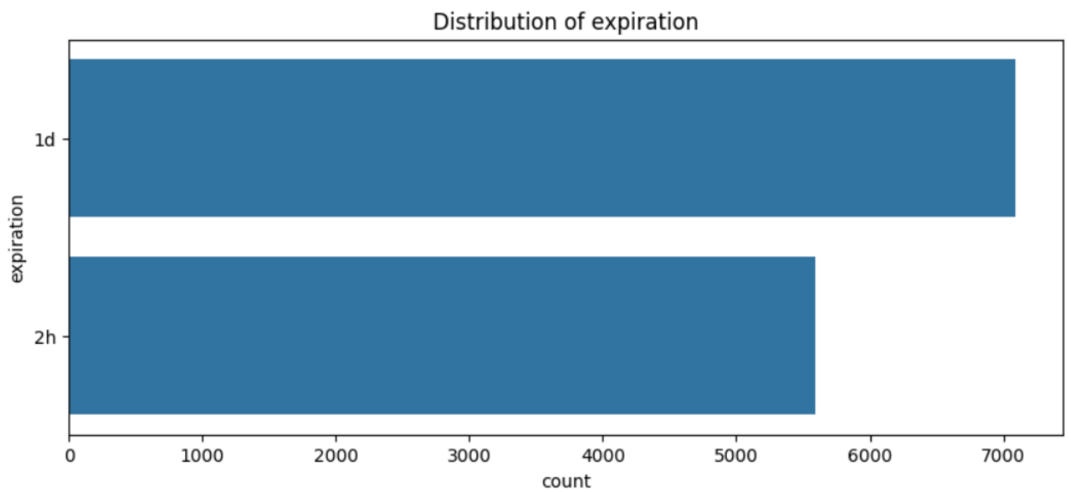

Figure 10: Distribution of expiration feature

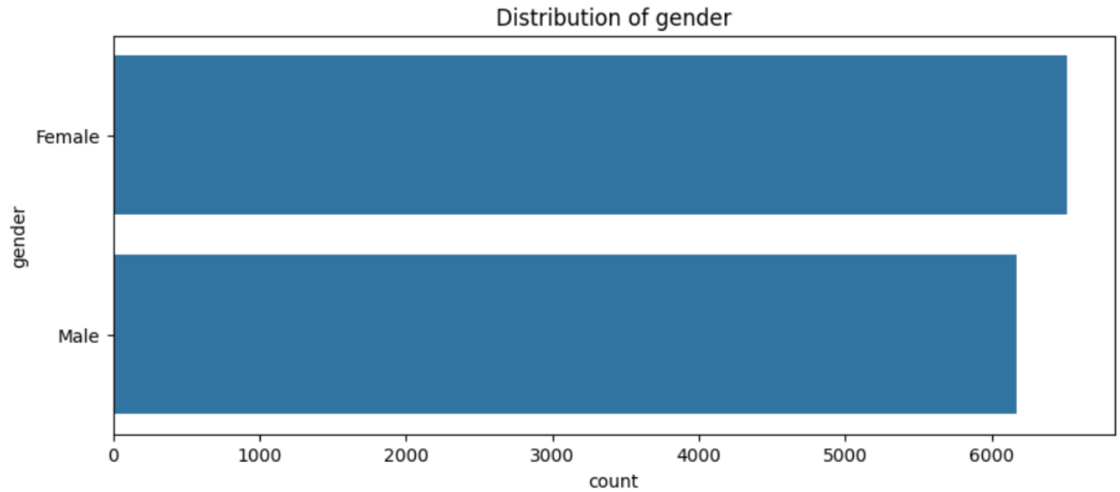

Figure 11: Distribution of gender feature

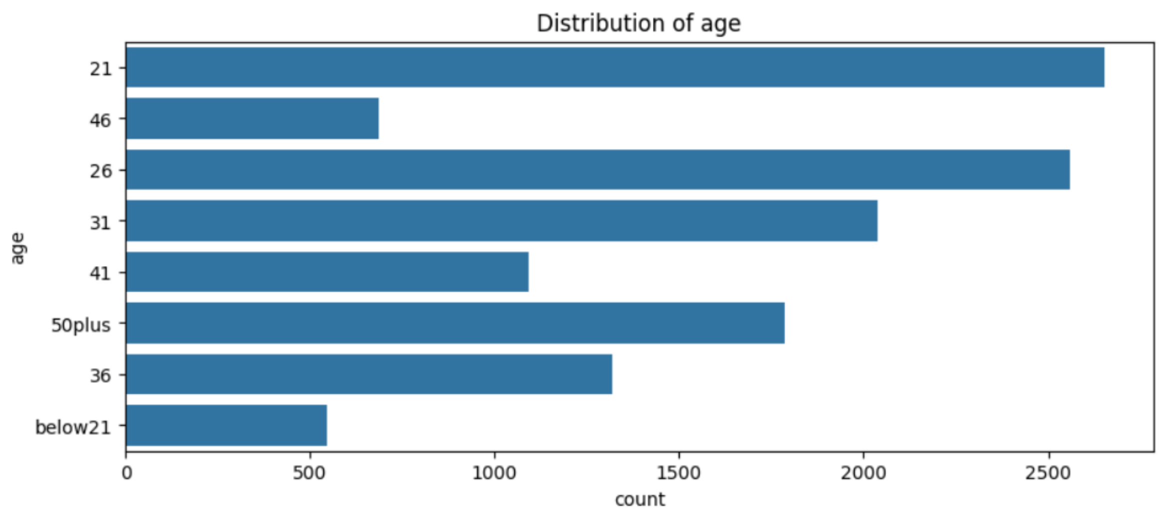

Figure 12: Distribution of age feature

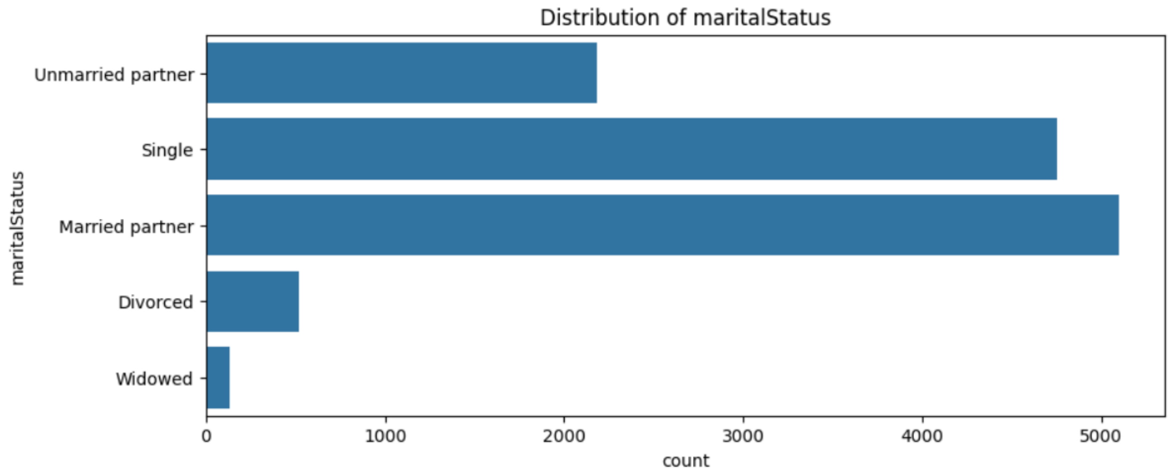

Figure 13: Distribution of marital status feature

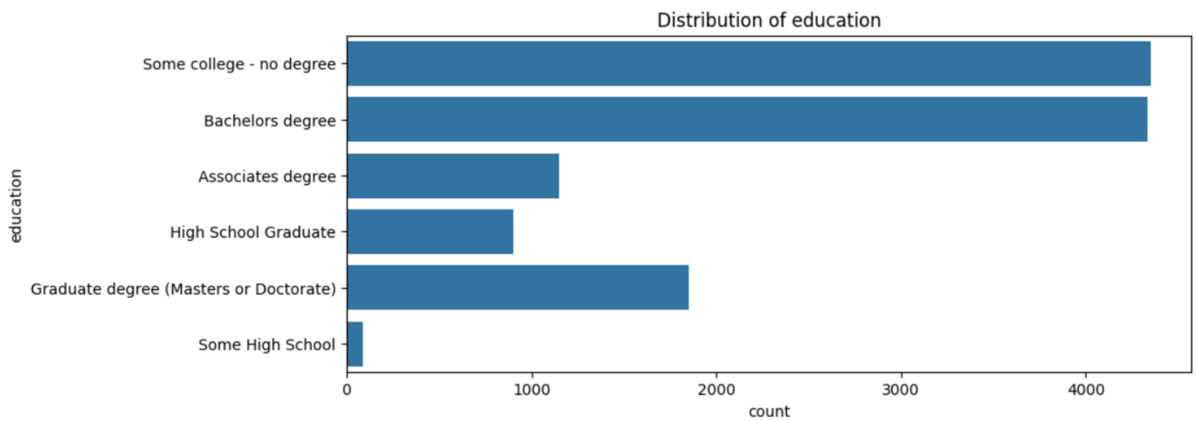

Figure 14: Distribution of education feature

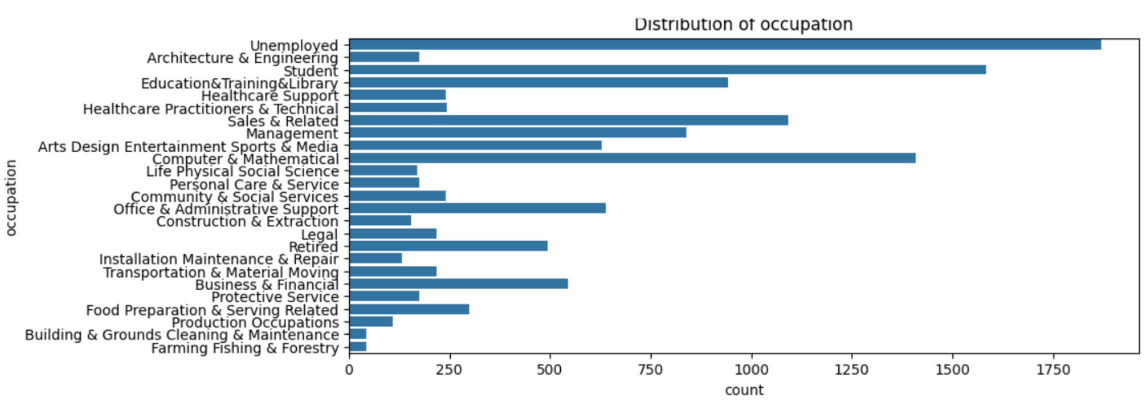

Figure 15: Distribution of occupation feature

Observation: Occupation has too many categories.

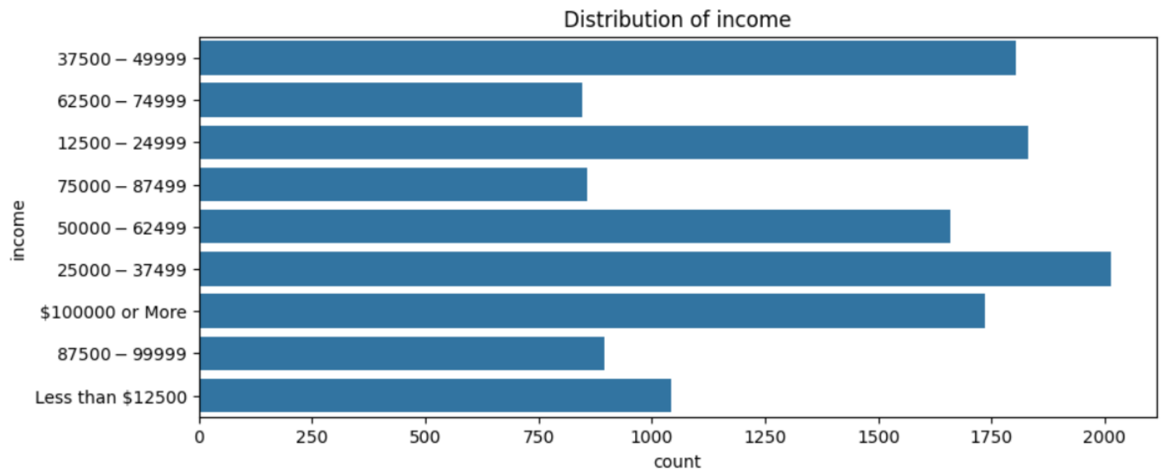

Figure 16: Distribution of income feature

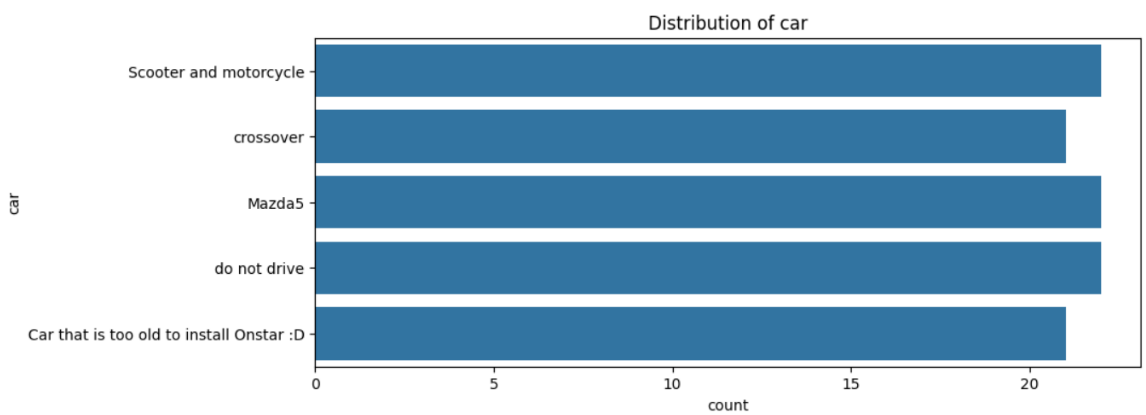

Figure 17: Distribution of car feature

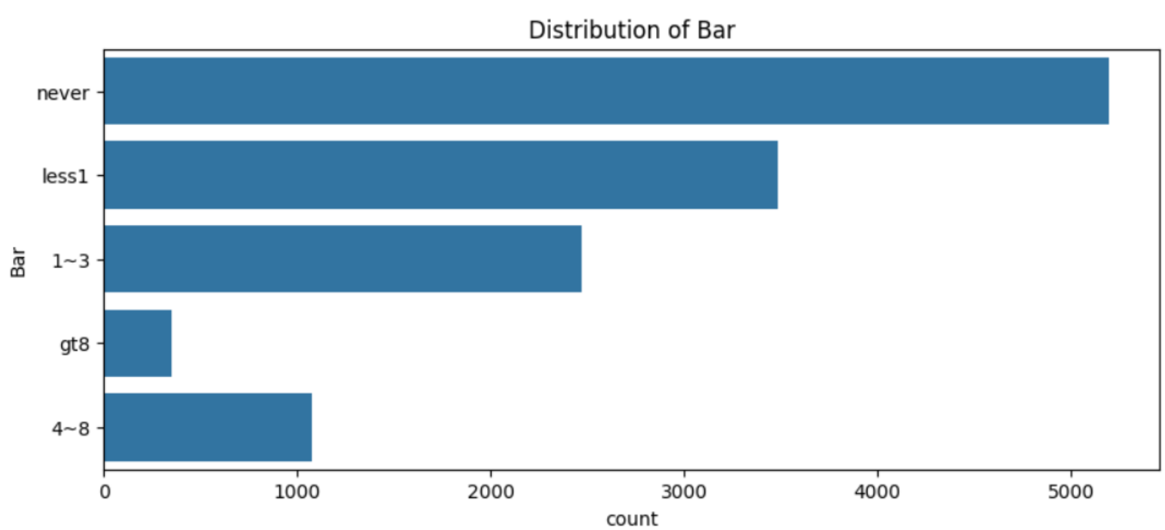

Figure 18: Distribution of bar feature

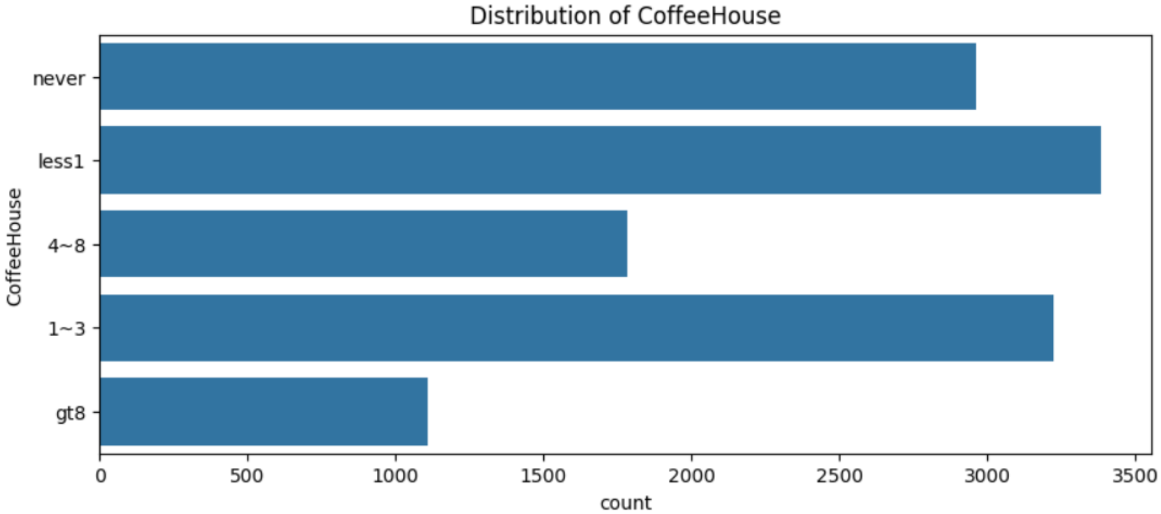

Figure 19: Distribution of Coffee House feature

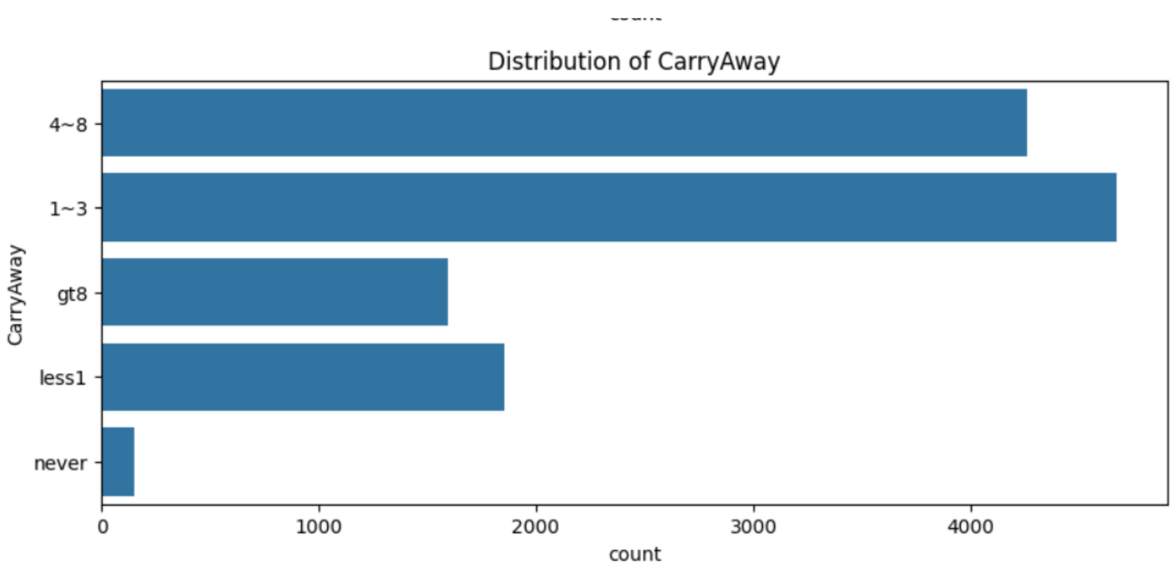

Figure 20: Distribution of Carry Away feature

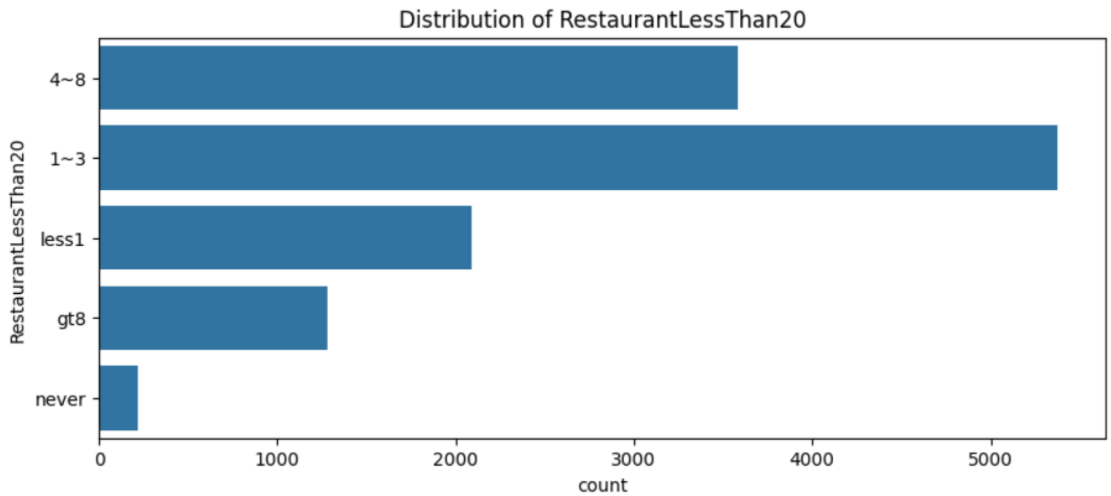

Figure 21: Distribution of Restaurant Less Than 20 feature

Observation: Most popular frequency for RestaurantLessThan20 feature is 1 to 3

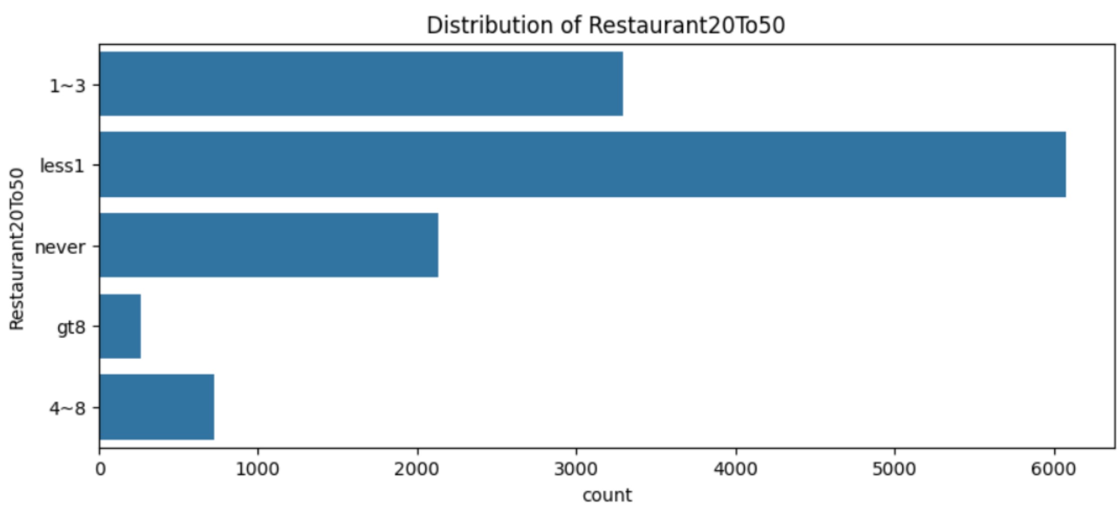

Figure 22: Distribution of Restaurant 20 to 50 feature

Observation: Most popular frequency for Restaurant20To50 feature is less 1

Using code from Inyama (2023), we can see the boxplots of the numerical data in numerical columns.

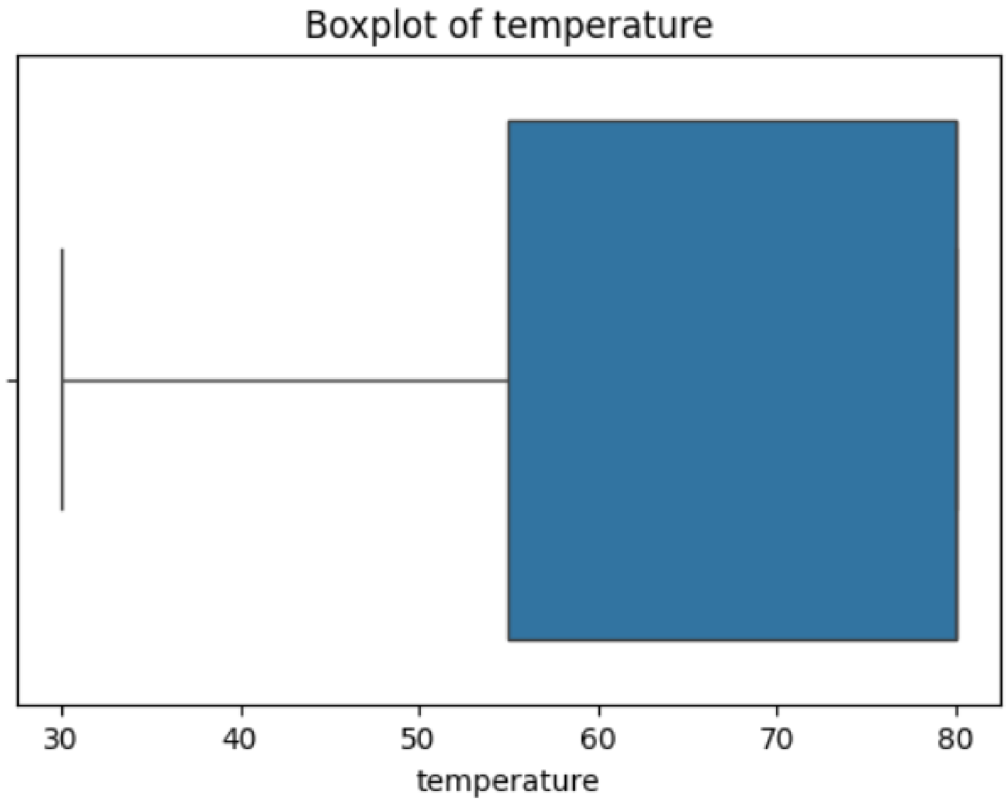

Figure 23: Boxplot of temperature

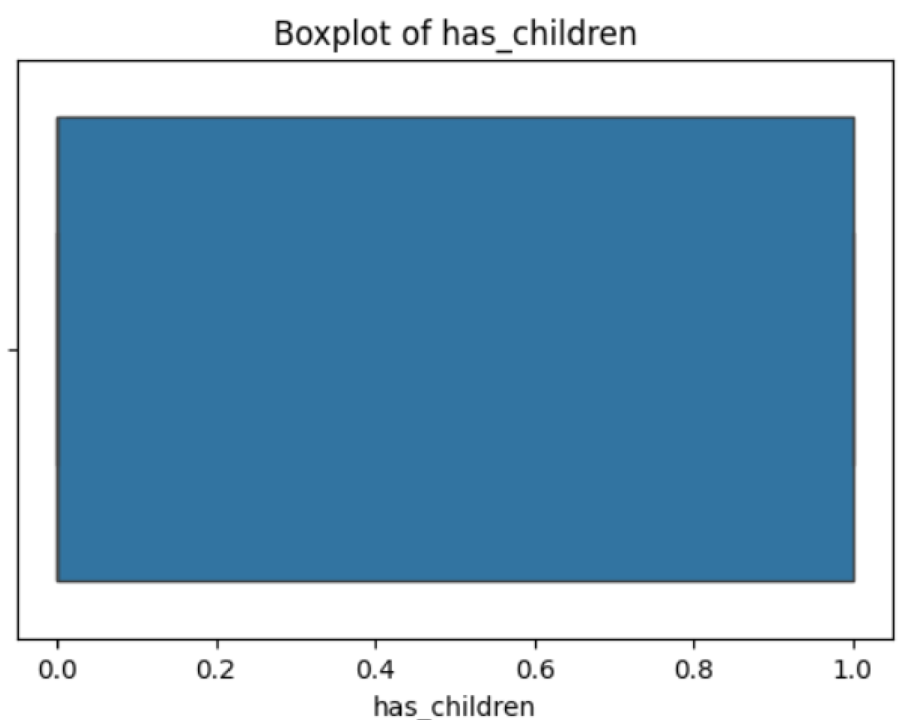

Figure 24: Boxplot of has children

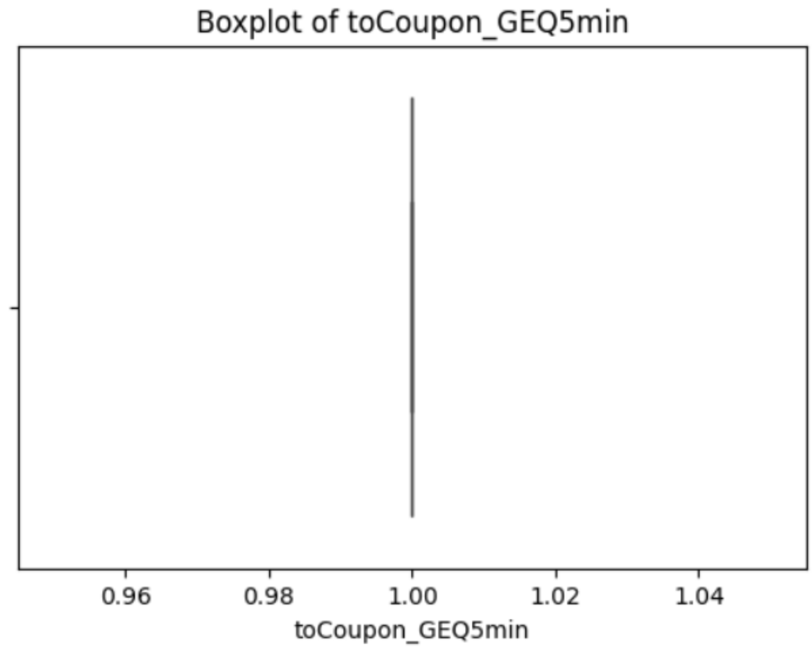

Figure 25: Boxplot of to coupon GEQ 5 min

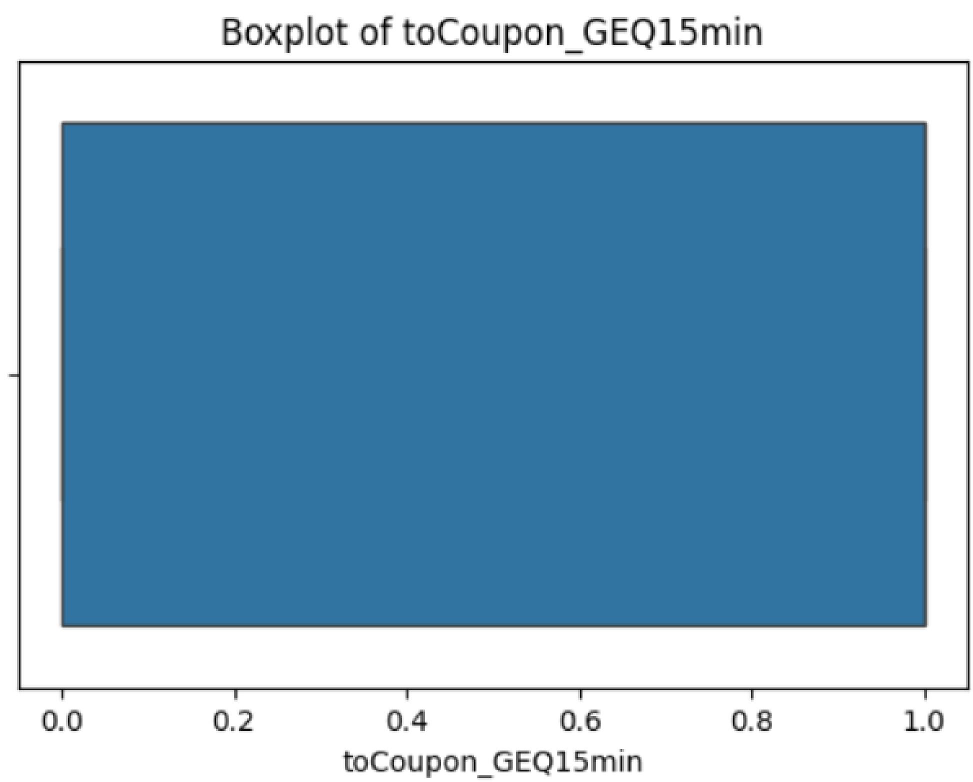

Figure 26: Boxplot of to coupon GEQ 15 min

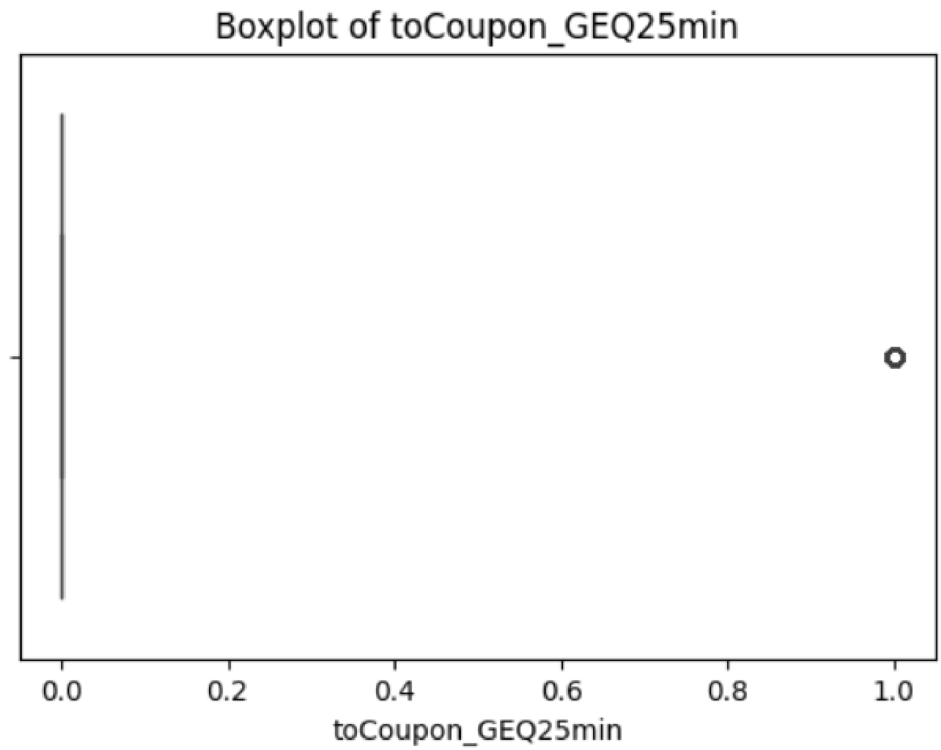

Figure 27: Boxplot of to coupon GEQ 25 min

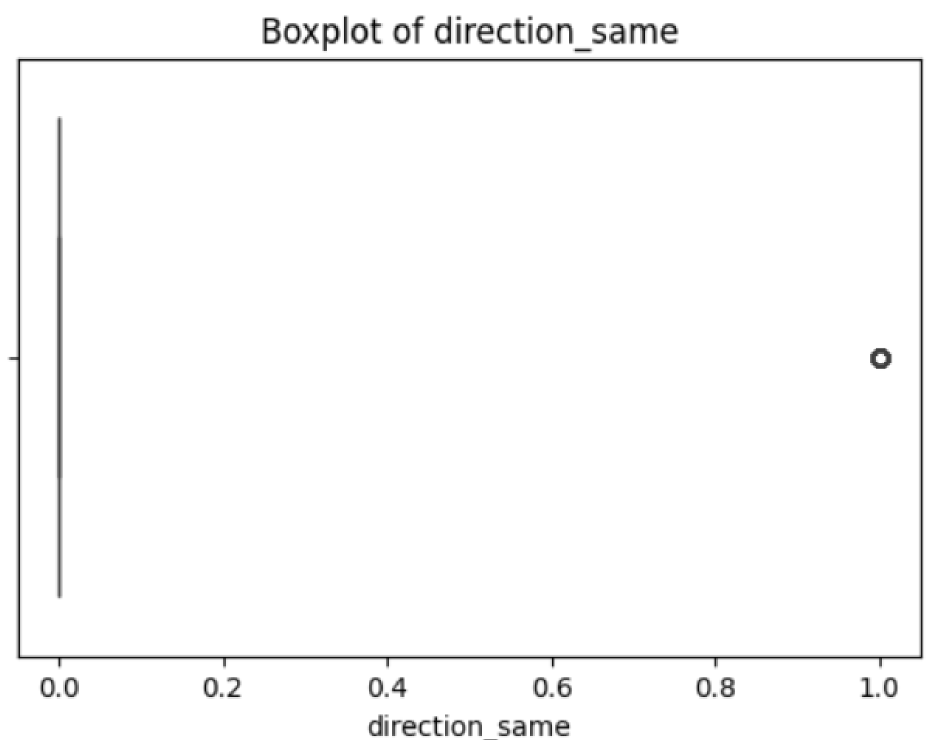

Figure 28: Boxplot of direction same

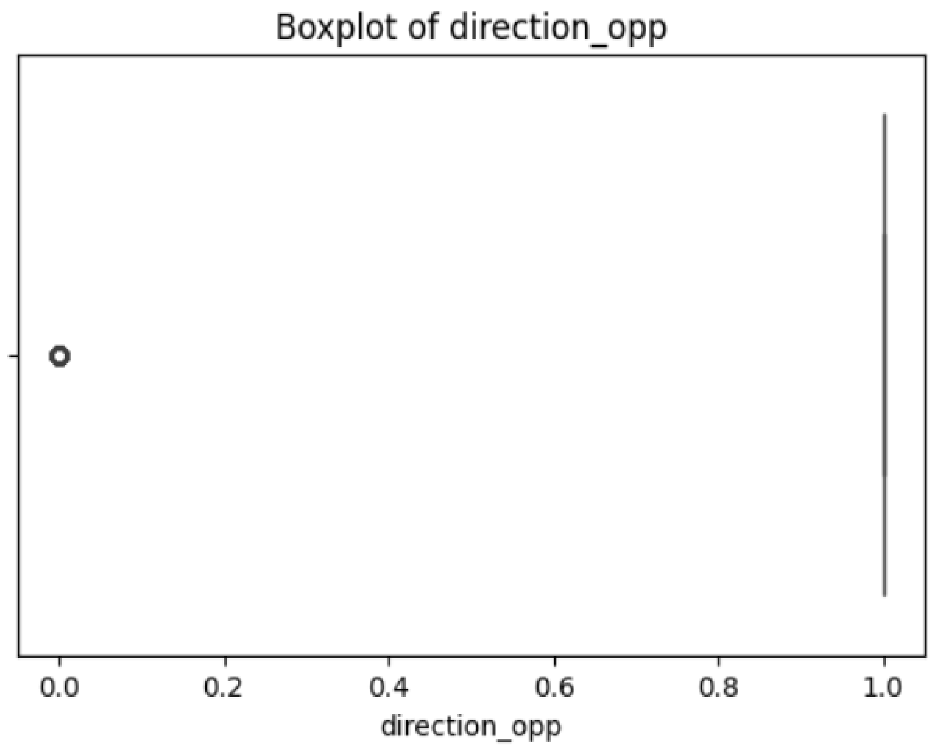

Figure 29: Boxplot of direction opposite

Using code provided by Niralidedaniya (2023), it was found that the target classes are partially balanced. If the target classes were highly unbalanced, then this dataset could not be used because the results of supervised learning algorithms used to make predictions would skew towards the class with the class with higher percentage of records.

```
Accepted coupons: 7210 56.843 %
Rejected coupons: 5474 43.157 %
```

Figure 30: Distribution of Target Classes

It should also be noted that this dataset comes with many missing values. Hence, this dataset requires preprocessing before it can be analyzed with machine learning algorithms. Using the code provided by Niralidedaniya (2023), we can see the features which have missing values.

Is there any missing value present or not? True

	missing_count	missing_percentage
destination	0	0.000000
passanger	0	0.000000
weather	0	0.000000
temperature	0	0.000000
time	0	0.000000
coupon	0	0.000000
expiration	0	0.000000
gender	0	0.000000
age	0	0.000000
maritalStatus	0	0.000000
has_children	0	0.000000
education	0	0.000000
occupation	0	0.000000
income	0	0.000000
car	12576	99.148534
Bar	107	0.843582
CoffeeHouse	217	1.710817
CarryAway	151	1.190476
RestaurantLessThan20	130	1.024913
Restaurant20To50	189	1.490066
toCoupon_GEQ5min	0	0.000000
toCoupon_GEQ15min	0	0.000000
toCoupon_GEQ25min	0	0.000000
direction_same	0	0.000000
direction_opp	0	0.000000
Y	0	0.000000

Figure 31: Distribution of missing values in the dataset

There are too many missing values for car feature. To avoid unreliable results, the car feature will be dropped from the dataset. toCoupon_GEQ5min has the same value for all rows. This does not add any useful information to the analysis. toCoupon_GEQ5min will be dropped from the dataset. Occupation feature has too many categories. This creates too much noise and distracts from clear analysis. Occupation feature will be dropped from the dataset.

```
#Drop column car as it has too many missing values
#Drop column toCoupon_GEQ5min because there is no variability in its value
#Drop column occupation because it has too many categories that leads to a lot of noise
in_vehicle_coupon_data = in_vehicle_coupon_data.drop(['car', 'occupation', 'toCoupon_GEQ5min'], axis=1)
in_vehicle_coupon_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64 
 4   time                  12684 non-null  object
 5   coupon                12684 non-null  object
 6   expiration            12684 non-null  object
 7   gender                12684 non-null  object
 8   age                   12684 non-null  object
 9   maritalStatus         12684 non-null  object
 10  has_children          12684 non-null  int64 
 11  education             12684 non-null  object
 12  income                12684 non-null  object
 13  Bar                   12577 non-null  object
 14  CoffeeHouse           12467 non-null  object
 15  CarryAway             12533 non-null  object
 16  RestaurantLessThan20  12554 non-null  object
 17  Restaurant20To50      12495 non-null  object
 18  toCoupon_GEQ15min     12684 non-null  int64 
 19  toCoupon_GEQ25min     12684 non-null  int64 
 20  direction_same        12684 non-null  int64 
 21  direction_opp         12684 non-null  int64 
 22  Y                     12684 non-null  int64 
dtypes: int64(7), object(16)
memory usage: 2.2+ MB
```

Figure 32: Information of dataset after dropping features: car, toCoupon_GEQ5min, occupation

From Table 1, Atiq et al. (2022) demonstrated pretty good accuracy when using frequent value imputation for missing values in the dataset along with the algorithms: Random Forest, Decision Tree, Logistic Regression, Gradient Boosted Tree, Naïve Bayes and Deep Learning. Therefore, this project is going to use frequent value imputation too for the remaining missing values. This can be easily done with the help of code snippet provided by Niralidedaniya (2022).

```
[13] # frequent value / mode imputation for missing values in data. This code snippet has been taken from Niralidedaniya (2022).
     in_vehicle_coupon_data['Bar']=in_vehicle_coupon_data['Bar'].fillna(in_vehicle_coupon_data['Bar'].value_counts().index[0])
     in_vehicle_coupon_data['CoffeeHouse']=in_vehicle_coupon_data['CoffeeHouse'].fillna(in_vehicle_coupon_data['CoffeeHouse'].value_counts().index[0])
     in_vehicle_coupon_data['CarryAway']=in_vehicle_coupon_data['CarryAway'].fillna(in_vehicle_coupon_data['CarryAway'].value_counts().index[0])
     in_vehicle_coupon_data['RestaurantLessThan20']=in_vehicle_coupon_data['RestaurantLessThan20'].fillna(in_vehicle_coupon_data['RestaurantLessThan20'].value_counts().index[0])
     in_vehicle_coupon_data['Restaurant20To50']=in_vehicle_coupon_data['Restaurant20To50'].fillna(in_vehicle_coupon_data['Restaurant20To50'].value_counts().index[0])
     #Lets check for missing values again
     print('Is there any missing value present?',in_vehicle_coupon_data.isnull().values.any())

     Is there any missing value present? False
```

Figure 33: Mode / frequent value imputation for missing values

Using ordinal encoding for the categorical values in the categorical features, the covariance matrix was generated. Strangely, direction_same feature has same values as direction_opp feature in the covariance matrix.

```
from sklearn.preprocessing import OrdinalEncoder
encoder = OrdinalEncoder()
encoded_in_vehicle_coupon_data = encoder.fit_transform(in_vehicle_coupon_data)
encoded_in_vehicle_coupon_data = pd.DataFrame(encoded_in_vehicle_coupon_data, columns=in_vehicle_coupon_data.columns)
encoded_in_vehicle_coupon_data.cov()
```

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	Bar	CoffeeHouse	CarryAway	Restaurant
destination	0.504658	-0.078836	-0.035245	-0.015041	0.420743	-0.010986	-0.011497	0.002407	-0.003339	0.001797	...	-0.005564	-0.007850	-0.005959	
passanger	-0.078836	0.887317	0.035703	0.040508	-0.543328	0.023840	0.034968	-0.009838	0.010415	-0.016771	...	0.022005	-0.012488	-0.017730	
weather	-0.035245	0.035703	0.401445	0.210926	-0.021266	0.125609	0.005569	-0.008552	-0.027806	-0.008874	...	0.010249	-0.005865	-0.026723	
temperature	-0.015041	0.040508	0.210926	0.587031	-0.065117	0.133050	0.047207	-0.009767	-0.046974	0.002288	...	0.010868	0.008828	-0.027345	
time	0.420743	-0.543328	-0.021266	-0.065117	2.072183	0.092426	-0.060532	-0.002523	-0.025719	0.005786	...	-0.009895	-0.011495	0.001266	
coupon	-0.010986	0.023840	0.125609	0.133050	0.092426	1.818577	0.099355	0.004809	0.006944	-0.000149	...	-0.015862	-0.004406	-0.004440	
expiration	-0.011497	0.034968	0.005569	0.047207	-0.060532	0.099355	0.246532	-0.000314	0.007207	-0.005098	...	-0.006669	-0.009675	-0.002079	
gender	0.002407	-0.009838	-0.008552	-0.009767	-0.002523	0.004809	-0.000314	0.249842	-0.067526	0.023960	...	-0.120201	0.045669	-0.009576	
age	-0.003339	0.010415	-0.027806	-0.046974	-0.025719	0.006944	0.007207	-0.067526	4.950184	-0.328927	...	0.712399	-0.002177	0.216170	
maritalStatus	0.001797	-0.016771	-0.008874	0.002288	0.005786	-0.000149	-0.005098	0.023960	-0.328927	0.693751	...	-0.118302	0.014938	-0.022649	
has_children	-0.002347	0.016028	0.003950	-0.007441	-0.005145	-0.006923	0.003918	-0.039384	0.335703	-0.177913	...	0.127863	-0.010256	0.003538	
education	0.011758	0.001656	0.015185	0.022296	-0.008154	-0.001837	-0.008597	0.014140	0.363653	0.107978	...	0.116209	0.011414	0.049950	
income	-0.025464	-0.005402	-0.051117	-0.047347	-0.022016	0.002894	-0.013671	0.032841	0.267308	0.150888	...	0.053494	-0.134355	0.120053	
Bar	-0.005564	0.022005	0.010249	0.010868	-0.009895	-0.015862	-0.006669	-0.120201	0.712399	-0.118302	...	2.407485	0.378947	0.134812	
CoffeeHouse	-0.007850	-0.012488	-0.005865	0.008828	-0.011495	-0.004406	-0.009675	0.045669	-0.002177	0.014938	...	0.378947	2.365560	0.195541	
CarryAway	-0.005959	-0.017730	-0.026723	-0.027345	0.001266	-0.004440	-0.002079	-0.009576	0.216170	-0.022649	...	0.134812	0.195541	1.194358	
RestaurantLessThan20	0.002886	-0.033238	-0.003905	-0.001147	-0.004725	0.016685	-0.006039	0.025848	-0.074016	-0.001212	...	0.139119	0.306658	0.113426	
Restaurant20To50	-0.000269	-0.042455	0.005444	0.002082	0.019281	0.011179	-0.000336	-0.000155	-0.004063	0.055322	...	0.375264	0.321059	0.193181	
toCoupon_GEQ15min	0.049593	0.030170	-0.038262	-0.059057	0.005156	-0.088045	0.010531	-0.001743	0.029335	-0.020447	...	0.015285	-0.003662	0.002056	
toCoupon_GEQ25min	0.045391	-0.060297	-0.041579	-0.053675	0.136276	-0.049269	-0.005304	0.000444	-0.000045	0.001348	...	0.002514	0.001584	0.001954	
direction_same	-0.024310	-0.103995	0.004609	0.030548	0.184128	-0.040432	0.006848	-0.000923	-0.007565	0.005645	...	-0.000763	0.011396	0.003997	
direction_opp	0.024310	0.103995	-0.004609	-0.030548	-0.184128	0.040432	-0.006848	0.000923	0.007565	-0.005645	...	0.000763	-0.011396	-0.003997	
Y	-0.000671	0.024082	0.031006	0.023241	-0.033780	0.064804	-0.031952	0.010886	-0.038836	0.010348	...	-0.058434	-0.110180	-0.026371	

Figure 34: Covariance matrix on the dataset after doing ordinal encoding on the categorical values of the dataset.

Since direction_same has same covariance values as direction_opp. It makes sense to just have one on them and reduce the noise.

```
in_vehicle_coupon_data = in_vehicle_coupon_data.drop(['direction_opp'], axis=1)
in_vehicle_coupon_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12684 entries, 0 to 12683
Data columns (total 22 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   destination           12684 non-null  object
 1   passanger             12684 non-null  object
 2   weather               12684 non-null  object
 3   temperature           12684 non-null  int64 
 4   time                  12684 non-null  object
 5   coupon                12684 non-null  object
 6   expiration            12684 non-null  object
 7   gender                12684 non-null  object
 8   age                   12684 non-null  object
 9   maritalStatus         12684 non-null  object
 10  has_children          12684 non-null  int64 
 11  education             12684 non-null  object
 12  income                12684 non-null  object
 13  Bar                   12684 non-null  object
 14  CoffeeHouse           12684 non-null  object
 15  CarryAway             12684 non-null  object
 16  RestaurantLessThan20  12684 non-null  object
 17  Restaurant20To50      12684 non-null  object
 18  toCoupon_GEQ15min     12684 non-null  int64 
 19  toCoupon_GEQ25min     12684 non-null  int64 
 20  direction_same        12684 non-null  int64 
 21  Y                     12684 non-null  int64 
dtypes: int64(6), object(16)
memory usage: 2.1+ MB
```

Figure 35: Dataset information after data cleaning

With all the steps above, the data is cleaned and ready to use. We can also generate a correlation matrix.

```python
#Get the correlation matrix using ordinal encoding for the categorical values
encoder = OrdinalEncoder()
encoded_in_vehicle_coupon_data = encoder.fit_transform(in_vehicle_coupon_data)
encoded_in_vehicle_coupon_data = pd.DataFrame(encoded_in_vehicle_coupon_data, columns=in_vehicle_coupon_data.columns)
correlation_matrix = encoded_in_vehicle_coupon_data.corr()
print(correlation_matrix)
```

```
                      destination  passanger   weather  temperature      time  \
destination              1.000000  -0.117811 -0.078305    -0.027633  0.411437
passanger               -0.117811   1.000000  0.059821     0.056127 -0.400690
weather                 -0.078305   0.059821  1.000000     0.434497 -0.023316
temperature             -0.027633   0.056127  0.434497     1.000000 -0.059041
time                     0.411437  -0.400690 -0.023316    -0.059041  1.000000
coupon                  -0.011468   0.018767  0.147008     0.128771  0.047612
expiration              -0.032594   0.074764  0.017702     0.124090 -0.084691
gender                   0.006779  -0.020896 -0.027003    -0.025504 -0.003507
age                     -0.002112   0.004969 -0.019725    -0.027556 -0.008030
maritalStatus            0.003036  -0.021376 -0.016816     0.003585  0.004826
has_children            -0.006707   0.034542  0.012657    -0.019716 -0.007256
education                0.008793   0.000934  0.012733     0.015460 -0.003009
income                  -0.014554  -0.002329 -0.032758    -0.025091 -0.006210
Bar                     -0.005048   0.015055  0.010426     0.009142 -0.004430
CoffeeHouse             -0.007185  -0.008619 -0.006018     0.007491 -0.005192
CarryAway               -0.007676  -0.017223 -0.038593    -0.032657  0.000804
RestaurantLessThan20     0.003497  -0.030378 -0.005306    -0.001289 -0.002826
Restaurant20To50        -0.000255  -0.030401  0.005795     0.001833  0.009035
toCoupon_GEQ15min        0.140684   0.064544 -0.121698    -0.155332  0.007218
toCoupon_GEQ25min        0.197240  -0.197595 -0.202572    -0.216254  0.292231
direction_same          -0.083328  -0.268830  0.017712     0.097085  0.311467
Y                       -0.001906   0.051614  0.098800     0.061240 -0.047377
```

```
                        coupon  expiration    gender       age  maritalStatus  \
destination          -0.011468   -0.032594  0.006779 -0.002112       0.003036
passanger             0.018767    0.074764 -0.020896  0.004969      -0.021376
weather               0.147008    0.017702 -0.027003 -0.019725      -0.016816
temperature           0.128771    0.124090 -0.025504 -0.027556       0.003585
time                  0.047612   -0.084691 -0.003507 -0.008030       0.004826
coupon                1.000000    0.148383  0.007134  0.002314      -0.000132
expiration            0.148383    1.000000 -0.001264  0.006523      -0.012328
gender                0.007134   -0.001264  1.000000 -0.060720       0.057552
age                   0.002314    0.006523 -0.060720  1.000000      -0.177495
maritalStatus        -0.000132   -0.012328  0.057552 -0.177495       1.000000
has_children         -0.010422    0.016020 -0.159956  0.306306      -0.433628
education            -0.000724   -0.009198  0.015029  0.086833       0.068872
income                0.000871   -0.011180  0.026677  0.048782       0.073555
Bar                  -0.007581   -0.008656 -0.154986  0.206363      -0.091539
CoffeeHouse          -0.002124   -0.012669  0.059404 -0.000636       0.011661
CarryAway            -0.003012   -0.003832 -0.017530  0.088903      -0.024881
RestaurantLessThan20  0.010651   -0.010471  0.044519 -0.028640      -0.001253
Restaurant20To50      0.005592   -0.000457 -0.000209 -0.001232       0.044802
toCoupon_GEQ15min    -0.131571    0.042740 -0.007028  0.026571      -0.049471
toCoupon_GEQ25min    -0.112780   -0.032977  0.002743 -0.000063       0.004997
direction_same       -0.073007    0.033584 -0.004496 -0.008279       0.016504
Y                     0.097019   -0.129920  0.043969 -0.035241       0.025083
```

```
                      ...    income       Bar  CoffeeHouse  CarryAway  \
destination           ... -0.014554 -0.005048    -0.007185  -0.007676
passanger             ... -0.002329  0.015055    -0.008619  -0.017223
weather               ... -0.032758  0.010426    -0.006018  -0.038593
temperature           ... -0.025091  0.009142     0.007491  -0.032657
time                  ... -0.006210 -0.004430    -0.005192   0.000804
coupon                ...  0.000871 -0.007581    -0.002124  -0.003012
expiration            ... -0.011180 -0.008656    -0.012669  -0.003832
gender                ...  0.026677 -0.154986     0.059404  -0.017530
age                   ...  0.048782  0.206363    -0.000636   0.088903
maritalStatus         ...  0.073555 -0.091539     0.011661  -0.024881
has_children          ... -0.000569  0.167292    -0.013537   0.006573
education             ...  0.001157  0.039789     0.003942   0.024282
income                ...  1.000000  0.013998    -0.035469   0.044603
Bar                   ...  0.013998  1.000000     0.158793   0.079502
CoffeeHouse           ... -0.035469  0.158793     1.000000   0.116333
CarryAway             ...  0.044603  0.079502     0.116333   1.000000
RestaurantLessThan20  ...  0.053753  0.077191     0.171650   0.089352
Restaurant20To50      ...  0.018483  0.163138     0.140805   0.119233
toCoupon_GEQ15min     ...  0.006196  0.019852    -0.004799   0.003791
toCoupon_GEQ25min     ...  0.003940  0.005002     0.003179   0.005519
direction_same        ...  0.017048 -0.001197     0.018042   0.008906
Y                     ... -0.023949 -0.076033    -0.144629  -0.048717
```

```
                      RestaurantLessThan20  Restaurant20To50  \
destination                       0.003497         -0.000255
passanger                        -0.030378         -0.030401
weather                          -0.005306          0.005795
temperature                      -0.001289          0.001833
time                             -0.002826          0.009035
coupon                            0.010651          0.005592
expiration                       -0.010471         -0.000457
gender                            0.044519         -0.000209
age                              -0.028640         -0.001232
maritalStatus                    -0.001253          0.044802
has_children                     -0.074252         -0.000614
education                         0.026678          0.081681
income                            0.053753          0.018483
Bar                               0.077191          0.163138
CoffeeHouse                       0.171650          0.140805
CarryAway                         0.089352          0.119233
RestaurantLessThan20              1.000000          0.111437
Restaurant20To50                  0.111437          1.000000
toCoupon_GEQ15min                 0.004797         -0.001461
toCoupon_GEQ25min                -0.003029          0.008245
direction_same                    0.003103          0.007554
Y                                -0.011137         -0.056268
```

	toCoupon_GEQ15min	toCoupon_GEQ25min	direction_same \
destination	0.140684	0.197240	-0.083328
passanger	0.064544	-0.197595	-0.268830
weather	-0.121698	-0.202572	0.017712
temperature	-0.155332	-0.216254	0.097085
time	0.007218	0.292231	0.311467
coupon	-0.131571	-0.112780	-0.073007
expiration	0.042740	-0.032977	0.033584
gender	-0.007028	0.002743	-0.004496
age	0.026571	-0.000063	-0.008279
maritalStatus	-0.049471	0.004997	0.016504
has_children	0.078211	-0.013722	-0.031620
education	-0.019017	-0.010812	0.001205
income	0.006196	0.003940	0.017048
Bar	0.019852	0.005002	-0.001197
CoffeeHouse	-0.004799	0.003179	0.018042
CarryAway	0.003791	0.005519	0.008906
RestaurantLessThan20	0.004797	-0.003029	0.003103
Restaurant20To50	-0.001461	0.008245	0.007554
toCoupon_GEQ15min	1.000000	0.324984	-0.303533
toCoupon_GEQ25min	0.324984	1.000000	-0.192319
direction_same	-0.303533	-0.192319	1.000000
Y	-0.081602	-0.103633	0.014570

	Y
destination	-0.001906
passanger	0.051614
weather	0.098800
temperature	0.061240
time	-0.047377
coupon	0.097019
expiration	-0.129920
gender	0.043969
age	-0.035241
maritalStatus	0.025083
has_children	-0.045557
education	0.043023
income	-0.023949
Bar	-0.076033
CoffeeHouse	-0.144629
CarryAway	-0.048717
RestaurantLessThan20	-0.011137
Restaurant20To50	-0.056268
toCoupon_GEQ15min	-0.081602
toCoupon_GEQ25min	-0.103633
direction_same	0.014570
Y	1.000000

Figure 36: Correlation matrix after cleaning In Vehicle Coupon dataset

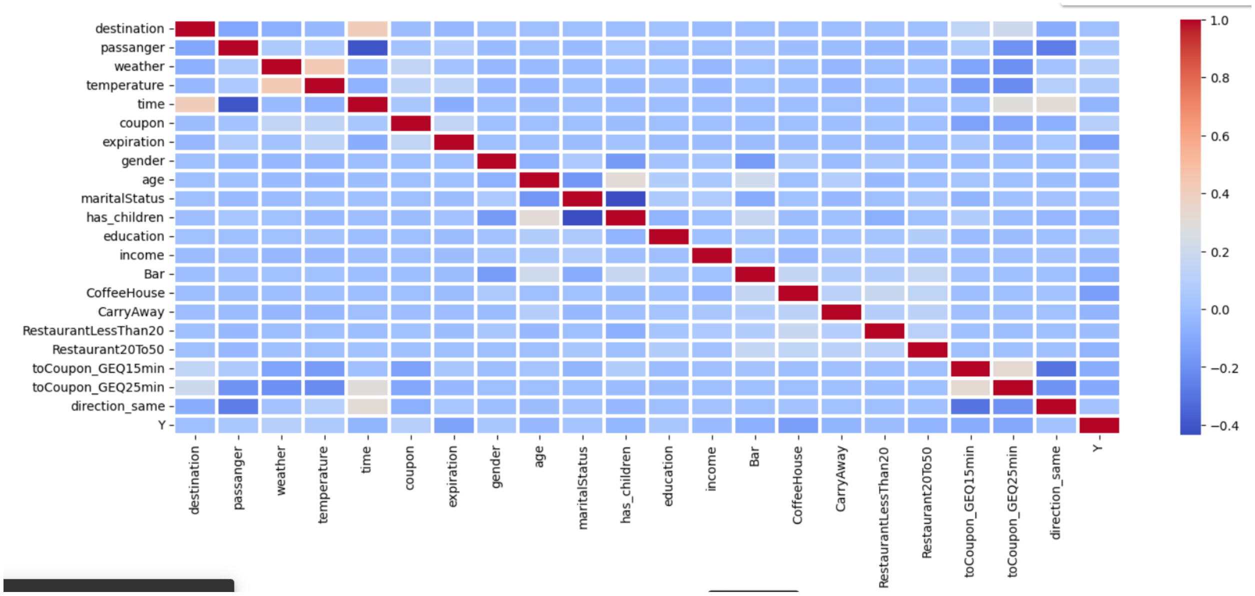

Figure 37: Heat map of correlation matrix

A heatmap to visualize the correlation between features is also effective.

There seems to be a correlation between time and destination, between temperature and weather, between marital status and has children, between passenger and time, between to coupon GEQ 15 min and same direction.

Methodology

The methodology being followed is based on the video by Babaoglu. (2018, January 6).

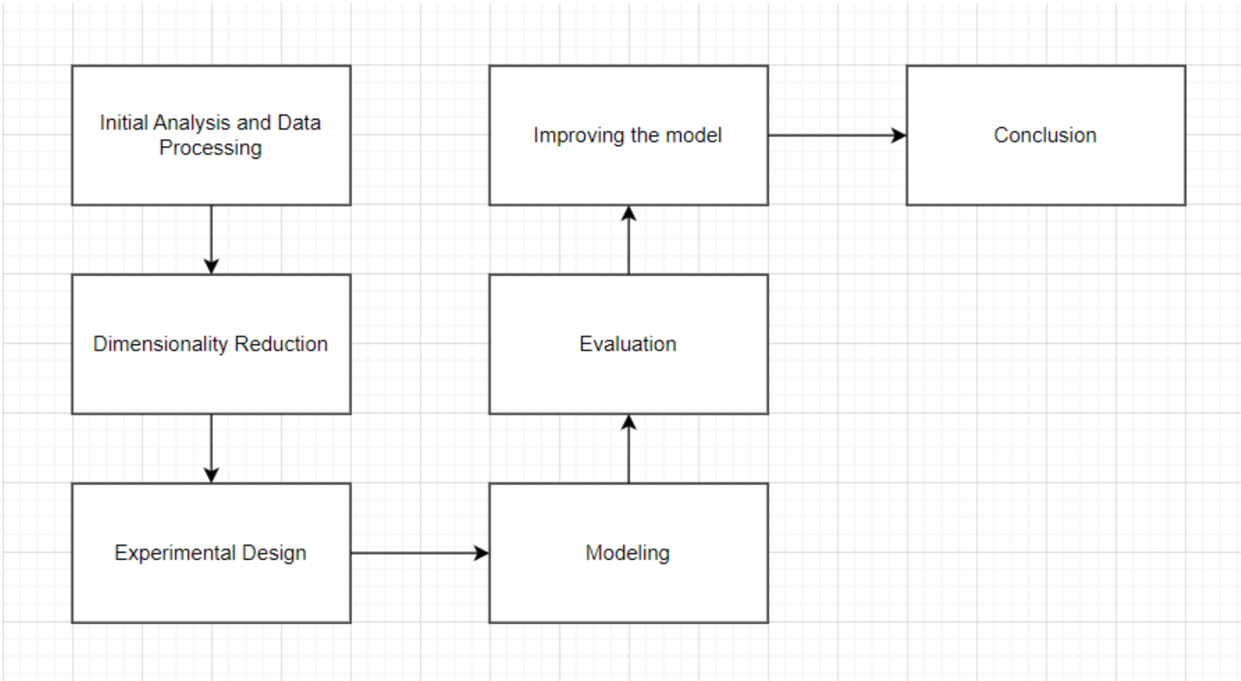

Figure 38: Proposed methodology for this project

In the Initial Analysis, the data will be described and examined. Univariate Analysis will be done to look at the distribution of each attribute. Check for missing values will be conducted. Data Processing will also be done and data will be cleaned.

During dimensionality reduction, a dataset with fewer dimensions using these 3 methods: Stepwise Regression, Forward Feature Selection and Backward Feature Elimination will be acquired.

Experimental Design will involve splitting data into a training set and test set to run classification algorithms over.

Modeling will involve running Random Forest, Decision Tree, Logistic Regression, Naïve Bayes, k-Nearest Neighbours (k-NN) on the dataset.

Evaluation of various classification algorithms will be done by comparing evaluation metrics like the Accuracy, Precision and Area under the Curve (AUC) of each algorithm.

Based on evaluation and comparison of the performance of various classification algorithms on the dataset, the model might be improved.

A conclusion is arrived at based on results.

References

Ahmed, N., & Umair, M. (2024). *Churn prediction using machine learning: A coupon optimization technique*. In World Journal of Advanced Engineering Technology and Sciences (Vols. 12–12, Issue 02, pp. 332–354) [Journal-article]. https://wjaets.com/sites/default/files/WJAETS-2024-0310.pdf

Atiq, R., Fariha, F., Mahmud, M., Yeamin, S. S., Rushee, K. I., & Rahim, S. (2022, October 8). *A comparison of missing value imputation techniques on coupon acceptance prediction*. MECS PRess. https://www.mecs-press.org/ijitcs/ijitcs-v14-n5/IJITCS-V14-N5-2.pdf

Babaoglu C. (n.d.). CMTH 642 – Data Analytics: Advanced Methods course [PowerPoint slide printouts]

Babaoglu C. (2018, January 6). *How to conduct data analysis process Systematically* [Video]. YouTube. https://www.youtube.com/watch?v=NMffLbFql5k

Inyama C. (2023). *In-Vehicle-Coupon-Recommendations/coupon_Project (2).ipynb at main · chrisinyama/In-Vehicle-Coupon-Recommendations*. GitHub. https://github.com/chrisinyama/In-Vehicle-Coupon-Recommendations/blob/main/coupon_Project%20(2).ipynb

Depari, G. S., Shu, E., Fachriza, C. A., Chow, J., Wijaya, J., & Winata, R. (2022). Customer's responses towards in-vehicle coupon recommendation an implementation of business analytics concept. *Jurnal Ekonomi* (Vol. 11, Issue 02) [Journal-article] https://download.garuda.kemdikbud.go.id/article.php?article=2994923&val=22616&title=CUSTOMER%27S%20RESPONSES%20TOWARDS%20IN-

VEHICLE%20COUPON%20RECOMMENDATION%20AN%20IMPLEMENTATION%20OF%20BUSINESS%20ANALYTICS%20CONCEPT

Niralidedaniya. (2023, January 19). In Vehicle Coupon Recommendation — a Machine learning classification case study. *Medium*. https://medium.com/@niralidedaniya/in-vehicle-coupon-recommendation-a-machine-learning-classification-case-study-df67e7835703

Patil, Y., Pawar, O., & Ingle, D. R. (2019). Coupon Purchase Prediction using Machine Learning. In *IJSRD - International Journal for Scientific Research & Development* (Vol. 7, Issue 02) [Journal-article]. https://ijsrd.com/articles/IJSRDV7I21146.pdf

Supervised learning. (n.d.). Scikit-learn. https://scikit-learn.org/stable/supervised_learning.html

UCI Machine Learning Repository. (2020). https://archive.ics.uci.edu/dataset/603/in+vehicle+coupon+recommendation

Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., & MacNeille, P. (2017). A Bayesian framework for learning rule sets for interpretable classification. In Maya Gupta (Ed.), *Journal of Machine Learning Research* (Vols. 1–37) [Journal-article]. https://jmlr.csail.mit.edu/papers/volume18/16-003/16-003.pdf