In-vehicle Coupon Recommendation Analysis

Submission Data: December 8, 2022

Tomoki Kyotani

Table of Contents

1. Project Overview / Problem Statement

In today's world, it is very important for companies to tailor their marketing and advertisement to individual customers to effectively utilize their resources and reach/attract more customers.

The goal of this project is to build a machine learning model to predict whether a driver accepts the provided coupon or not. Utilizing this model, based on the demographics of a driver, companies can perform personalized marketing and provide coupons that are most likely to be accepted by drivers.

2. Dataset Overview

The dataset used for the project is called "in-vehicle coupon recommendation Data Set" and can be found at UCI Machine Learning Repository at:

https://archive.ics.uci.edu/ml/datasets/in-vehicle+coupon+recommendation.

Tong Wang (University of Iowa) and Cynthia Rudin (Duke University) collected this dataset via a survey on Amazon Mechanical Turk and used it for their published paper "A bayesian framework for learning rule sets for interpretable classification". The survey describes driving scenarios with different conditions such as destination, passengers, weather, temperature, etc.. and asks the people surveyed if he/she would accept a provided coupon if he/she was a driver.

3. Data Dictionary

Feature	*Data Type*	*Possible Values / Description*
destination	String	No Urgent Place, Home, Work
passanger	String	Alone, Friend(s), Kid(s), Partner (feature meaning: who are the passengers in the car)
weather	String	Sunny, Rainy, Snowy
temperature	Integer	55, 80, 30
time	String	2PM, 10AM, 6PM, 7AM, 10PM
coupon	String	Restaurant(<$20), Coffee House, Carry out & Take away, Bar, Restaurant($20-$50)
expiration	String	1d, 2h (the coupon expires in 1 day or in 2 hours)
gender	String	Female, Male
age	String	21, 46, 26, 31, 41, 50plus, 36, below21
maritalStatus	String	Unmarried partner, Single, Married partner, Divorced, Widowed
has_Children	Integer	1, 0
education	String	Some college - no degree, Bachelors degree, Associates degree, High School Graduate, Graduate degree (Masters or Doctorate), Some High School
occupation	String	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	String	$37500 - $49999, $62500 - $74999, $12500 - $24999, $75000 - $87499, $50000 - $62499, $25000 - $37499, $100000 or More, $87500 - $99999, Less than $12500
car	String	Scooter and motorcycle, Mazda5, do not drive, crossover, Car that is too old to install Onstar :D (feature meaning: type of car)
Bar	String	never, less1, 1~3, gt8, nan4~8 (feature meaning: how many times do you go to a bar every month?)
CoffeeHouse	String	never, less1, 4~8, 1~3, gt8, nan (feature meaning: how many times do you go to a coffeehouse every month?)
CarryAway	String	1~3, 4~8, gt8, less1, never (feature meaning: how many times do you get take-away food every month?)
RestaurantLessThan20	String	4~8, 1~3, less1, gt8, never (feature meaning: how many times do you go to a restaurant with an average expense per person of less than $20 every month?)
Restaurant20To50	String	1~3, less1, never, gt8, 4~8, nan (feature meaning: how many times do you go to a restaurant with average expense per person of $20 - $50 every month?)
toCoupon_GEQ5min	Integer	0,1 (feature meaning: driving distance to the restaurant/bar for using the coupon is greater than 5 minutes)
toCoupon_GEQ15min	Integer	0,1 (feature meaning: driving distance to the restaurant/bar for using the coupon is greater than 15 minutes)
toCoupon_GEQ25min	Integer	0, 1 (feature meaning: driving distance to the restaurant/bar for using the coupon is greater than 25 minutes)
direction_same	Integer	0, 1 (feature meaning: whether the restaurant/bar is in the same direction as your current destination)
direction_opp	Integer	1, 0 (feature meaning: whether the restaurant/bar is in the same direction as your current destination)
Y	Integer	1, 0 (whether the coupon is accepted)

4. Exploratory Data Analysis

In this dataset, there are 12,684 instances with 26 features as shown in the table above. Among these features, the target variable is ***'Y'***, which indicates whether the coupon provided is accepted or not by a driver.

The table below shows the missing (null) values in the dataset. Among all the listed features, the feature 'car' has the highest number of missing values, which accounts for 99% of the entire dataset. I will remove this entire 'car' feature as well as perform some imputation for other features with missing values during the data preprocessing stage.

Feature	***# of Null (% of entire dataset)***
car	12,576 (99%)
Bar	107 (0.8%)
CoffeeHouse	217 (1.7%)
CarryAway	151 (1.2%)
RestaurantLessThan20	130 (1%)
Restaurant20To50	189 (1/5%)

There is one non-binary continuous feature, 'temperature'. Although the data type of this feature is integer, the values of this features in the provided dataset are either 30, 55, or 80, and it does not contain any other value.

'temperature' Attribute - Unique Value Counts:

Class/Label	***Count***
30	2316
55	3840
80	6528

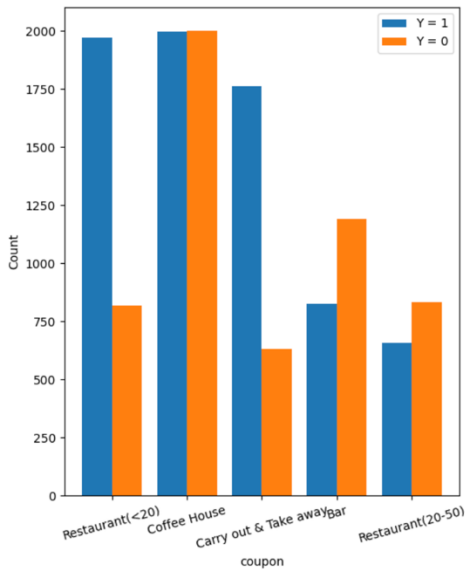
Regarding the target variable, which shows whether a driver accepted a coupon or not ('Y' attribute), and the provided coupon types ('coupon' attribute), the counts of each class/label in these features are shown below:

'Y' Feature – Unique Value Counts:

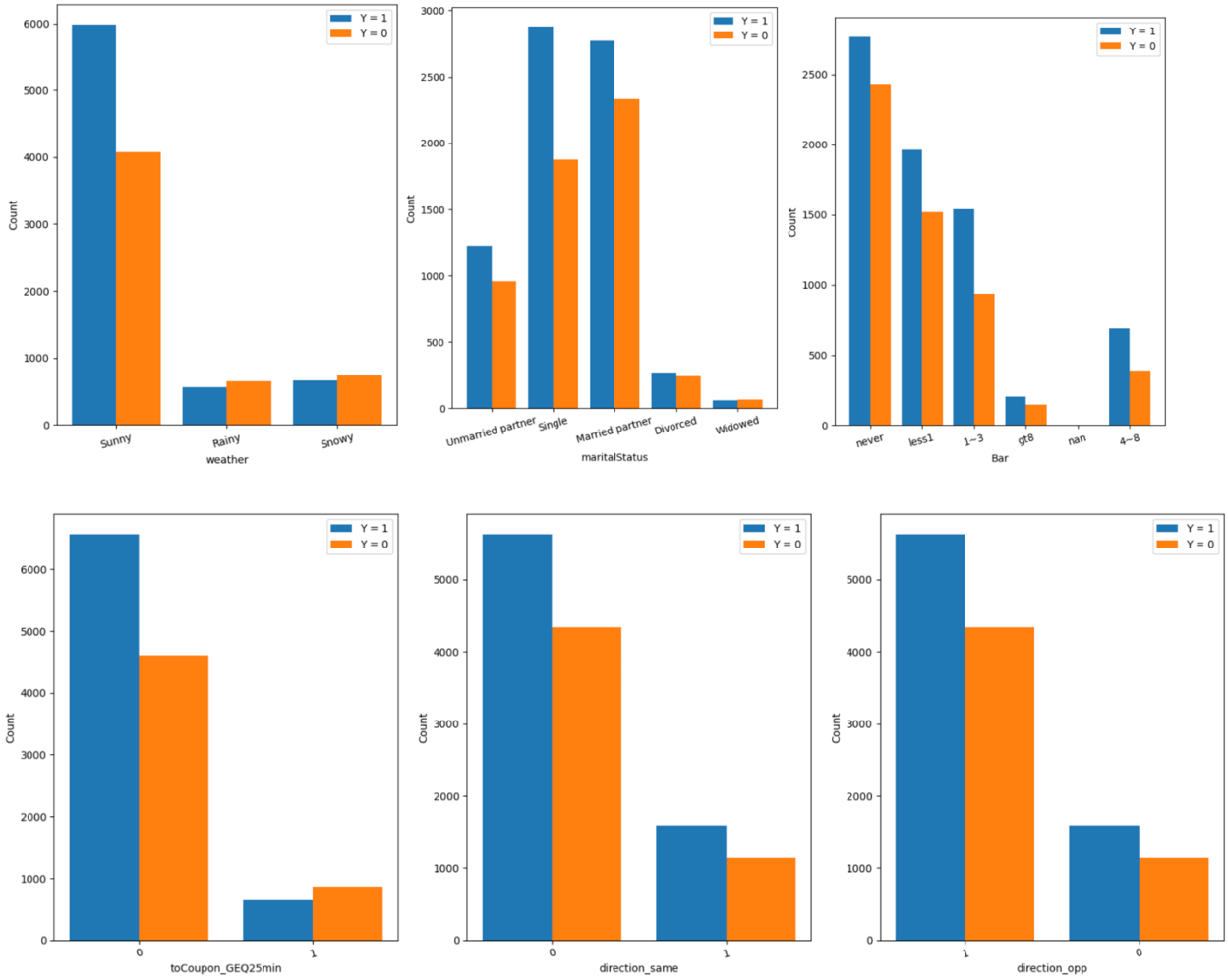
Class/Label	***Count***
1	7210
0	5474

'coupon' Feature - Unique Value Counts:

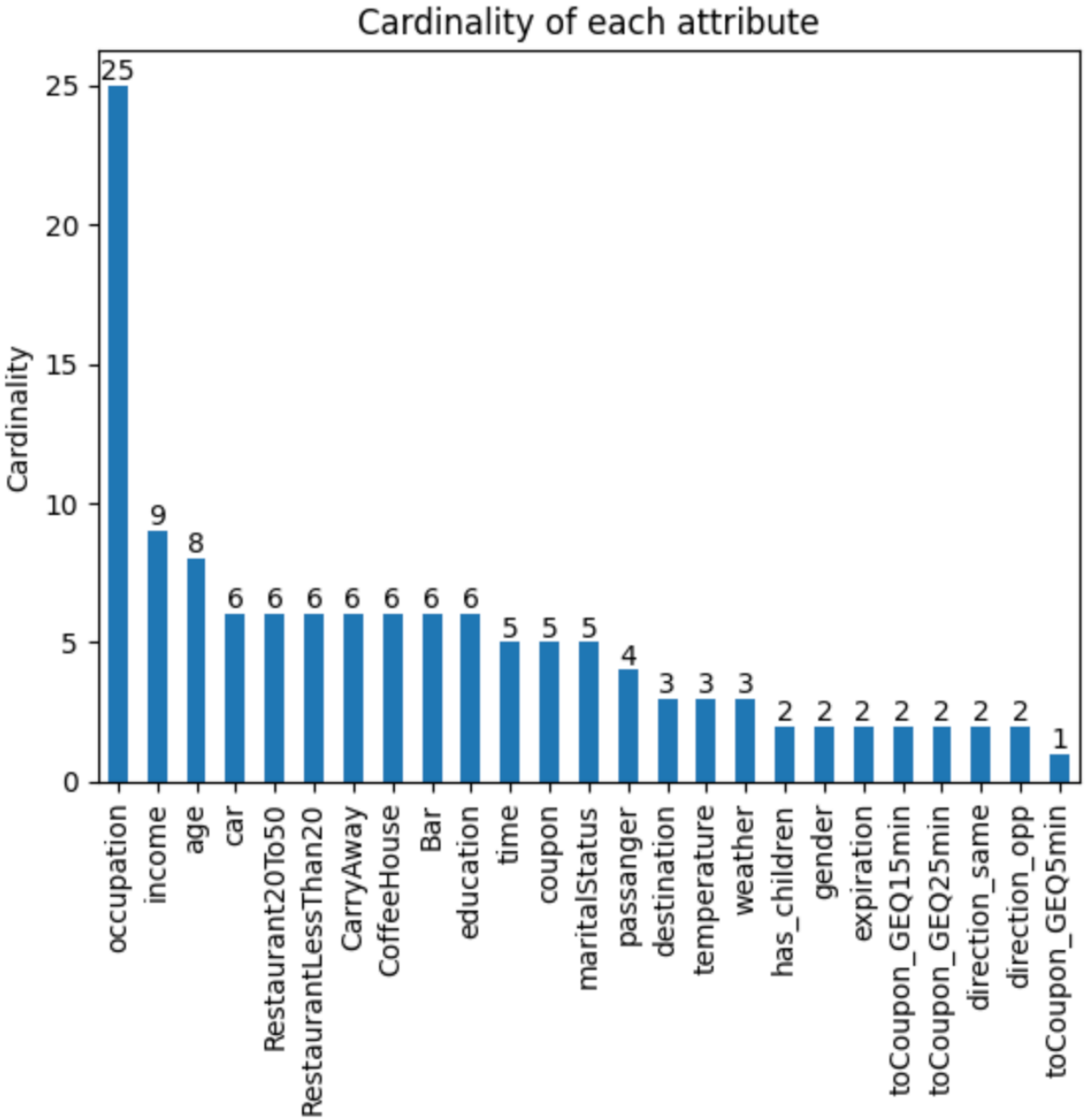
Class/Label	***Count***
Coffee House	3996
Restaurant(<20)	2786
Carry out & Take away	2393
Bar	2017
Restaurant(20-50)	1492



When looking at the distributions within each feature, the features such as ‘weather’, ‘martialStatus’, ‘Bar’, ‘toCoupon_GEQ25min’, ‘direction_same’, and ‘direction_opp’ are highly imbalanced (*Distributions of all features are listed in the Appendix – A. Distributions of each feature):



The cardinalities of each feature are shown below. The ‘occupation’ feature has the highest cardinality among all features and has 25 unique values. If algorithms chosen for this project require predictors to be continuous such as logistic regression, I need to perform some encoding to convert categorical predictors to numerical. However, if I choose One Hot Encoding, the feature with high cardinality like ‘occupation’ will generate 25 new attributes, and this can cause performance issues as well as the curse of dimensionality. In case such issues result in poor performance of the model, I will try the Target Encoding or dropping this feature and see how it will improve the performance.


Cardinality of each attribute
Cardinality
0
5
10
15
20
25
25
9
8
6
6
6
6
6
6
6
5
5
5
4
3
3
3
2
2
2
2
2
2
2
1
occupation
income
age
car
Restaurant20To50
RestaurantLessThan20
CarryAway
CoffeeHouse
Bar
education
time
coupon
maritalStatus
passanger
destination
temperature
weather
has_children
gender
expiration
toCoupon_GEQ15min
toCoupon_GEQ25min
direction_same
direction_opp
toCoupon_GEQ5min

5. Data Preprocessing

Before starting training models, I will first impute missing values in five of the features, namely 'Bar', 'CoffeHouse', 'CarryAway', 'RestaurantLessThan20', and 'Restaurant20To50'.

Feature	***# of Null (% of entire dataset)***
car	12,576 (99%)
Bar	107 (0.8%)
CoffeeHouse	217 (1.7%)
CarryAway	151 (1.2%)
RestaurantLessThan20	130 (1%)
Restaurant20To50	189 (1/5%)

Due to its high percentage of missing values (99%), I dropped the entire column for the 'car' feature.

For the other five features, I built a decision tree for each of the features and used it to predict their missing values. Each of the decision tree was trained with the features besides the 'Bar', 'CoffeHouse', 'CarryAway', 'RestaurantLessThan20', and 'Restaurant20To50' as predictors and with each of these five features as a target variable.

I explored multiple values (3, 5, 8, 10, and 12) of the maximum tree depth ('max_depth' hyperparameter), and the best performance was obtained when 'max_depth' = 12 for all decision trees. The decision trees for each of the five target variables ('Bar', 'CoffeHouse', 'CarryAway', 'RestaurantLessThan20', and 'Restaurant20To50') had the training accuracies of 0.96. 0.89, 0.94, 0.94, and 0.90 and the test accuracies of 0.94, 0.89, 0.92, 0.92, and 0.89 respectively.

Using these decision trees, I predicted and imputed the missing values. Now, the dataset has 12,684 instances with 25 attributes with no missing values.

6. Model Training and Methodology

As mentioned in the Exploratory Data Analysis section 4, the dataset has only one continuous feature, and all other features are categorical. The decision-tree-based algorithm is likely to perform well with such a dataset, and I decided to use the Random Forest algorithm and AdaBoost algorithm with a stump as a weak learner. I also explored the Logistic Regression and K-Nearest Neighbors (KNN) algorithms.

I first trained each model using the entire dataset. Next, I performed the feature selecting using a decision tree and trained the models with the same algorithms to see if there is any performance improvement. In a classifier decision tree, the higher level a feature is located at in a decision tree, the more important the feature is since it reduces more impurity. Thus, the features that are located above the level 4 of the decision tree were selected for the purpose of feature selection. After this feature selection, the number of predictors was reduced to 10, and these 10 predictors

include 'coupon', 'CoffeeHouse', 'Bar', 'toCoupon_GEQ25min', 'expiration', 'temperature', 'direction_same', 'passanger', 'age', and 'income'.

For all the model training process, the dataset was split into training:test = 80:20 in order to see performance of each model against both the training and the new (test) data.

At the end, I compared each model's performance metrics such as accuracy, precision, recall, training time, prediction time, and selected the best performing model to recommend in order to achieve the project objective.

Below is the information about hyperparameters and their values I tuned in each model. For both the entire dataset and the dataset with feature selection, I tuned the hyperparameters in the same way.

- RandomForest
    - Maximum tree depth (max_depth): 8, 10, 12
    - Number of trees (#_trees): 100, 150, 200
- AdaBoost
    - Number of classifiers (#_classifiers): 100, 300, 500, 700, 1000
- Logistic Regression
    - Learning rate (learning_rate): 0.001, 0.005, 0.01
- KNN
    - Number of neighbors (#_K): 6, 9, 12
    - Distance method: manhattan, Euclidean

7. Model Performance

The table below lists the performance metrics from all the models trained in the section above:

Model	Feature Selection	Dataset	Hyperparameter 1	Hyperparameter 2	Training Time (s)	Prediction Time (s)	Precision - Denied	Recall - Denied	Precision - Accepted	Recall - Accepted	Accuracy
RandomForest	No	Training	max_depth = 12	#_trees = 100	103.574	4.576	0.91	0.73	0.82	0.95	0.85
RandomForest	No	Test	max_depth = 12	#_trees = 100	103.574	1.151	0.74	0.56	0.72	0.85	0.73
AdaBoost	No	Training	#_claasifiers = 100	NA	14.850	0.028	0.62	0.14	0.59	0.93	0.59
AdaBoost	No	Test	#_claasifiers = 100	NA	14.850	0.008	0.63	0.13	0.59	0.94	0.59
Logistic Regression	No	Training	learning_rate = 0.01	NA	3.599	0.002	0.69	0.16	0.60	0.94	0.61
Logistic Regression	No	Test	learning_rate = 0.01	NA	3.599	0.001	0.64	0.15	0.59	0.93	0.60
KNN	No	Training	#_K = 12	distance = manhattan	0.001	40.612	0.70	0.73	0.79	0.76	0.75
KNN	No	Test	#_K = 12	distance = manhattan	0.001	10.146	0.64	0.67	0.74	0.72	0.70
RandomForest	Yes	Training	max_depth = 12	#_trees = 100	63.409	4.457	0.80	0.65	0.77	0.88	0.78
RandomForest	Yes	Test	max_depth = 12	#_trees = 100	63.409	1.126	0.70	0.57	0.71	0.81	0.71
AdaBoost	Yes	Training	#_claasifiers = 100	NA	6.215	0.028	0.61	0.08	0.58	0.96	0.58
AdaBoost	Yes	Test	#_claasifiers = 100	NA	6.215	0.007	0.54	0.08	0.58	0.95	0.57
Logistic Regression	Yes	Training	learning_rate = 0.05	NA	1.557	0.001	0.43	0.99	0.71	0.01	0.44
Logistic Regression	Yes	Test	learning_rate = 0.05	NA	1.557	0.000	0.43	1.00	0.79	0.01	0.43
KNN	Yes	Training	#_K = 6	distance = manhattan	0.001	18.671	0.71	0.80	0.84	0.76	0.78
KNN	Yes	Test	#_K = 6	distance = manhattan	0.001	4.692	0.62	0.73	0.76	0.66	0.69

From the performance metrics, the three best-performing models are:

- Random Forest model with max_depth = 12 and #_trees = 100 without feature selection
  - This model has the highest accuracies from the training dataset (0.85) and the test dataset (0.73).
  - This model performed better in the training dataset but performed worse when facing the new dataset from the test. Thus, this model has low bias and high variance and has some overfitting issue.
  - This model has a high Recall score for the 'Accepted' class (0.85 from the test dataset). Thus, using this model, 85% of the data can be correctly predicted out of actual 'Accepted' class data.
  - On the other hand, this model has a low Recall score for the 'Denied' class (0.56 from the test dataset). Thus, using this model, only 56% of the data can be correctly predicted out of actual 'Denied' class data. This model is way better at predicting the 'Accepted' class data.
  - With the dataset without feature selection, this model took the longest time (103.6 seconds) to train. Thus, if the business has requirements where they need to

constantly update/train the model with a new dataset and thus a shorter training time, this might not be the best model for them.

- KNN model with #_K = 12 and disntance_metrics = manhattan without feature selection
  - This model has the accuracies of 0.75 from the training dataset and 0.70 from the test.
  - From the metrics above metrics, this model also has somewhat high variance and thus has some overfitting issue.
  - Comparing to the other two best-performing models, this model has more balanced performance metrics. For instance, the Recall for the 'Accepted' and 'Denied' classes are 0.72 and 0.67 respectively, and there is as large gap between these values as the other two models.
  - This model requires very short time to train (0.01 for both training and test datasets) but longer time to predict. This is because the calculation is happening only during the prediction.

- Random Forest model with max_depth = 12 and #_trees = 100 with feature selection
  - This model has the accuracy of 0.78 from the training dataset and 0.71 from the test dataset. This model also has some overfitting issue due to it high variance.
  - This model has a high Recall score for the 'Accepted' class (0.81 from the test dataset). Thus, using this model, 81% of the data can be correctly predicted out of actual 'Accepted' class data.
  - On the other hand, this model has a low Recall score for the 'Denied' class (0.57 from the test dataset). Thus, using this model, only 57% of the data can be correctly predicted out of actual 'Denied' class data. This model is way better at predicting the 'Accepted' class data.
  - This model took the second longest time (63.4 seconds) to train. It does not take as long as the first Random Forest model mentioned above, but we still need to take this model's training time into consideration if the business requires the faster training time.

8. Conclusion

Among the three models mentioned in the section 6, I recommend the ***Random Forest model -1 with max_depth = 12 and #_trees = 100 with feature selection*** (Selected features are: 'coupon', 'CoffeeHouse', 'Bar', 'toCoupon_GEQ25min', 'expiration', 'temperature', 'direction_same', 'passanger', 'age', and 'income').

In this model, the top 5 important features are:

- 'expiration'… shows whether the coupon expires in 1 day or in 2 hours
- 'direction_same'… whether the restaurant/bar is in the same direction as your current destination
- 'income'… income of the driver
- 'coupon'… type of a coupon

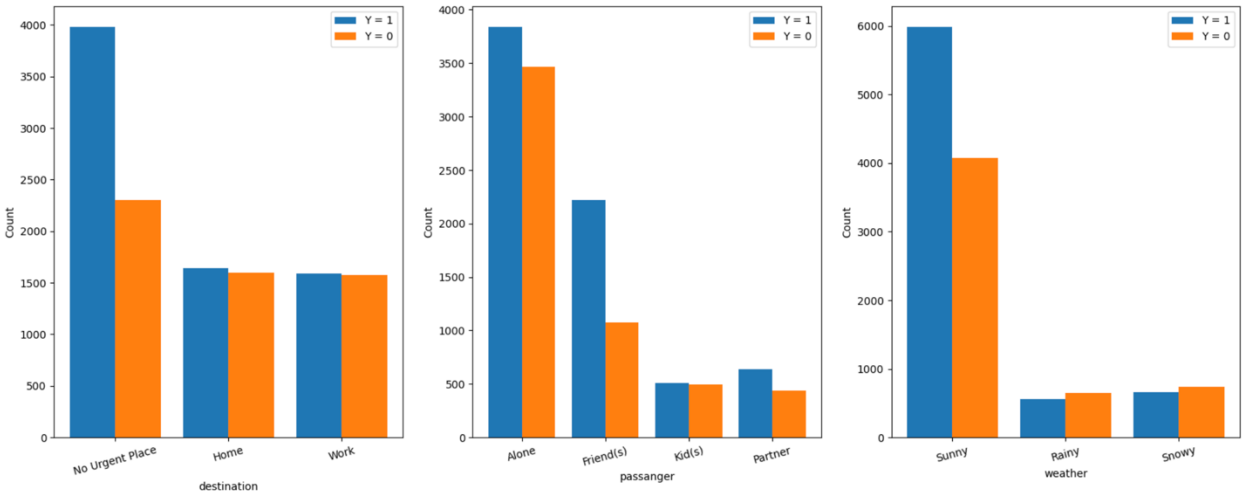
- ‘toCoupon_GEQ25min’… whether driving distance to the restaurant/bar for using the coupon is greater than 25 minutes

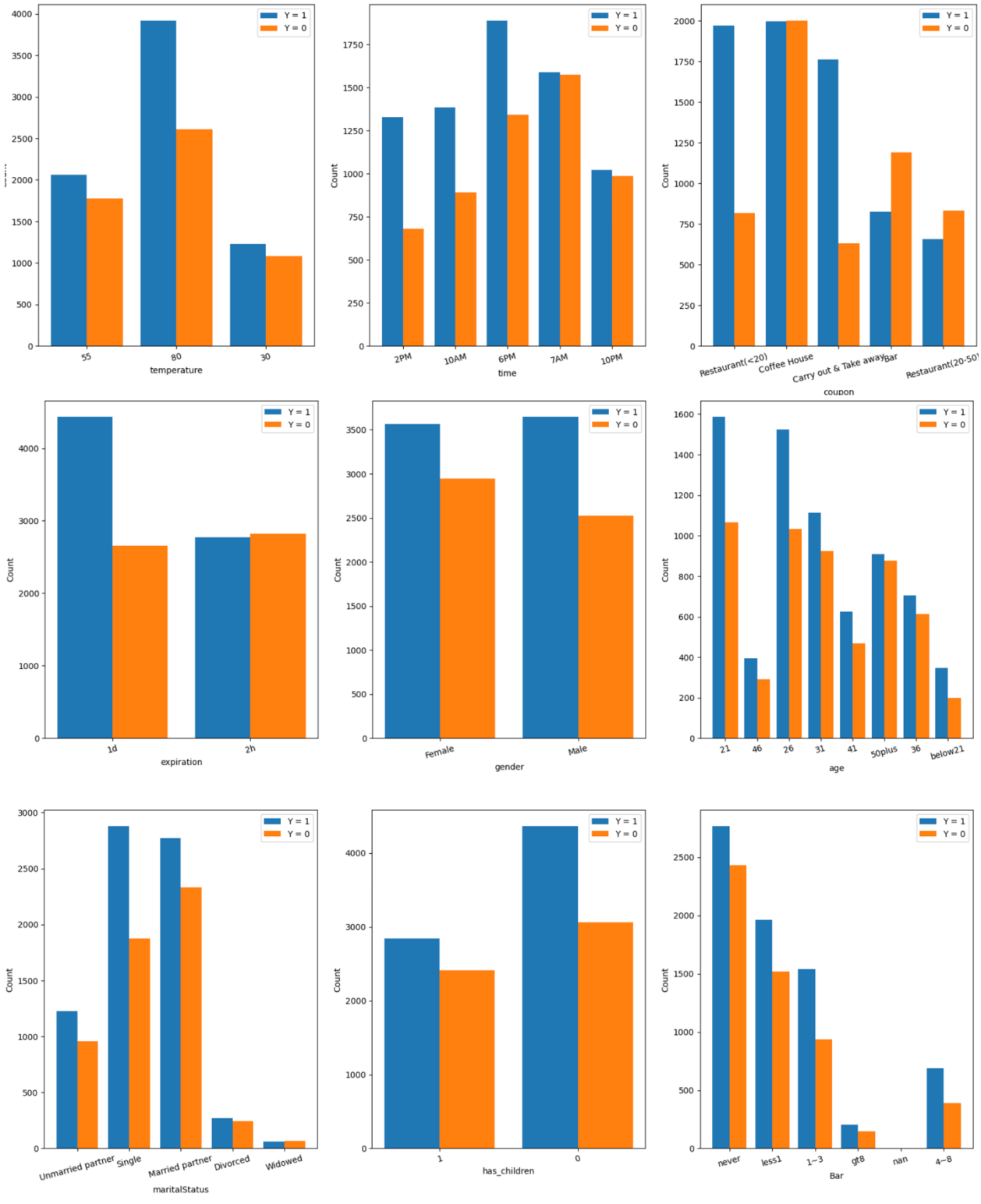
This model has accuracies of 0.78 from the training dataset and 0.71 from the test dataset. Although the other Random Forest model -2, which was mentioned first in the section 6, has higher accuracy (0.85 from the training and 0.73 from the test), the selected Random Forest model -1 took about 40 seconds faster to train than the other Random Forest model -2 (63.4 vs 103.5 seconds). Since the accuracy of the Random Forest model -2 from the test dataset was 0.73, which is only 0.02 higher than the selected model -1, using the selected model -1, we can reduce the training time dramatically while losing only 0.02 accuracy.

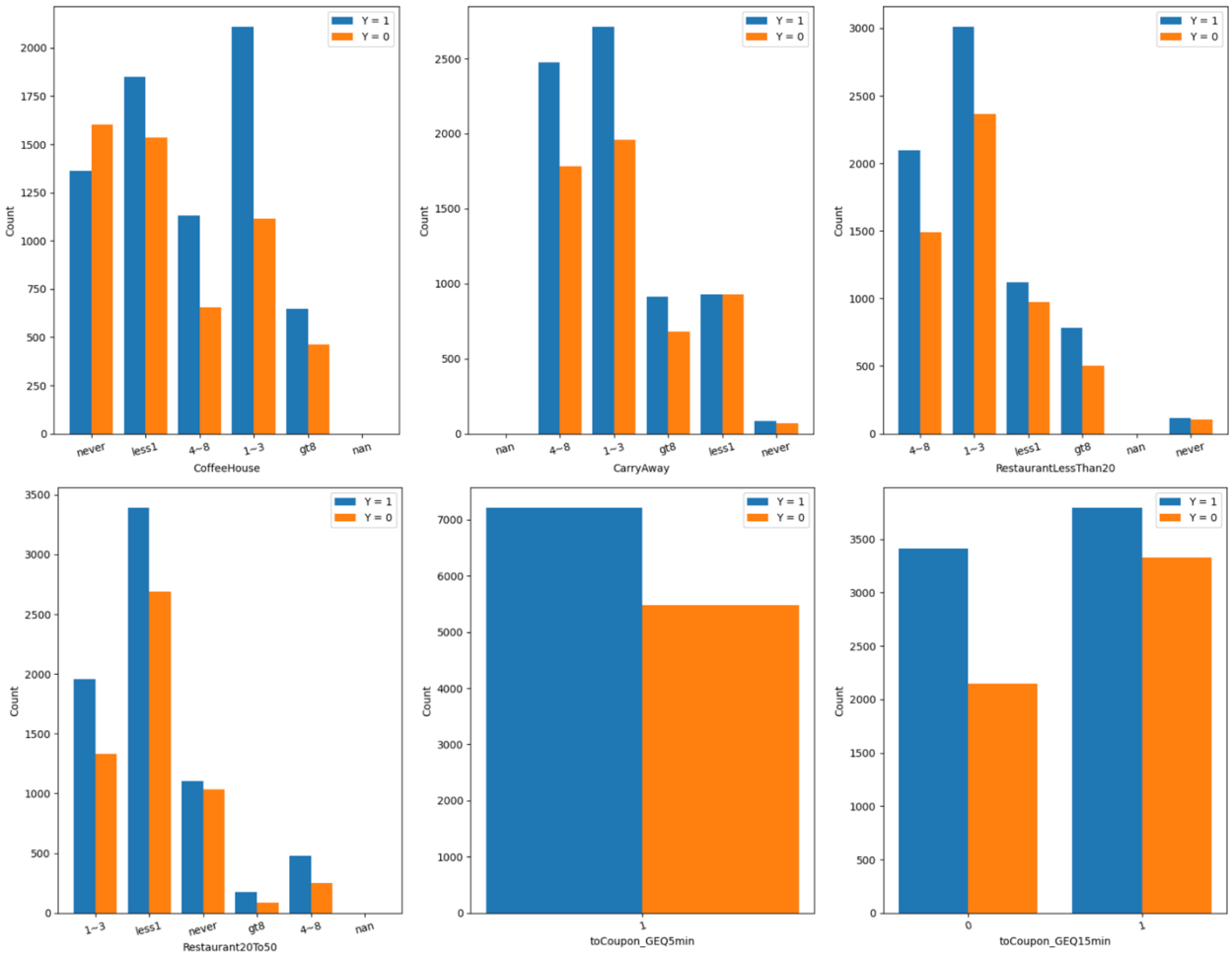
In addition, the selected model has a high Recall score for the ‘Accepted’ class (0.81). In this business scenario, it is important to have a high score for this metric because we do not want to miss any opportunities to recommend coupons to drivers. As it is just a coupon, it does not require lots of costs to recommend one to a driver. Thus, we want to focus more on distributing coupons to drivers with any potential of accepting the given coupons and do not need to concern too much about getting denied by the drivers.

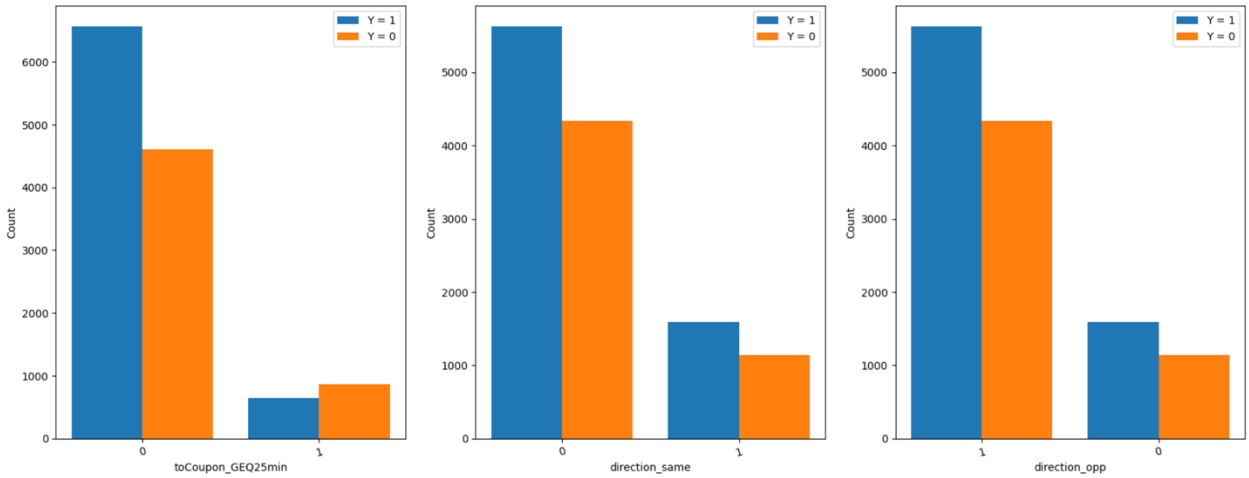
9. Appendix

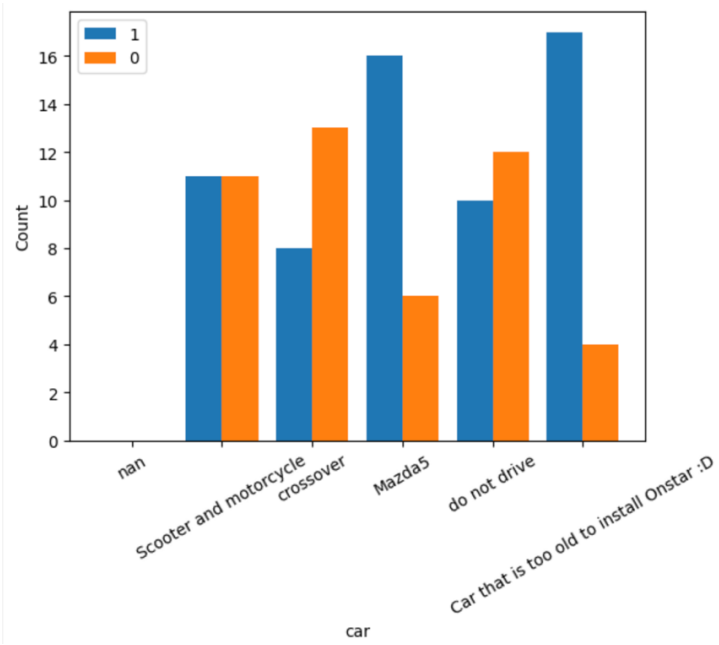
A. Distributions of each feature

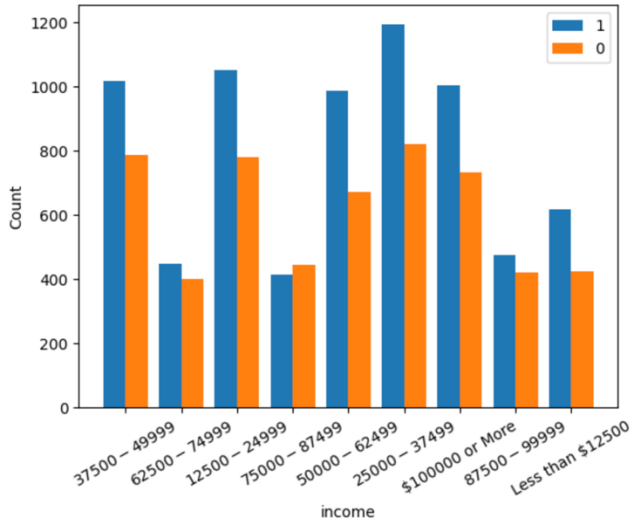


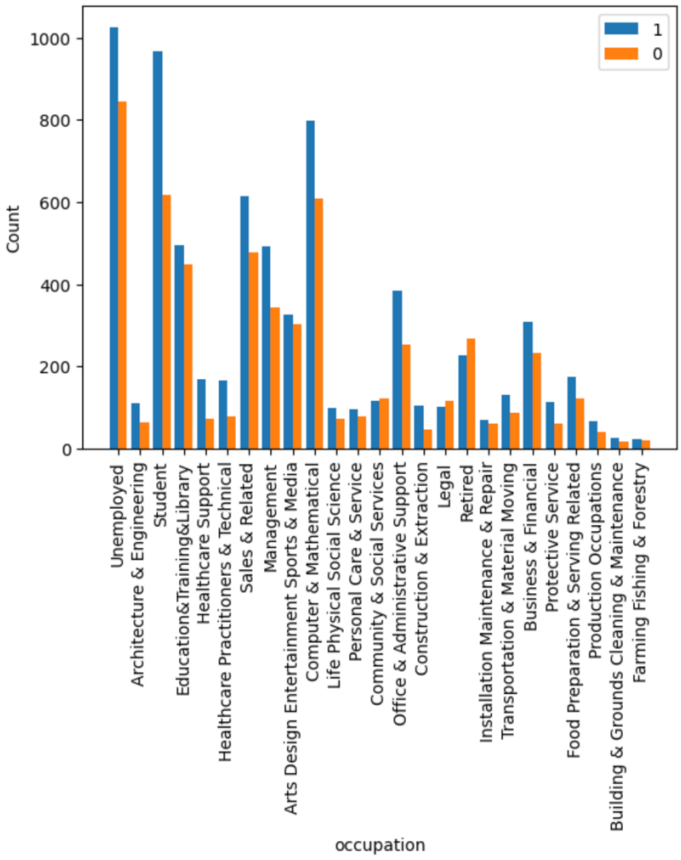

Y = 1
Y = 0
temperature
55
80
30
time
2PM
10AM
6PM
7AM
10PM
Count
coupon
Restaurant(<20)
Coffee House
Carry out & Take away
Bar
Restaurant(20-50)
expiration
1d
2h
gender
Female
Male
age
21
46
26
31
41
50plus
36
below21
maritalStatus
Unmarried partner
Single
Married partner
Divorced
Widowed
has_children
1
0
Bar
never
less1
1~3
gt8
nan
4~8

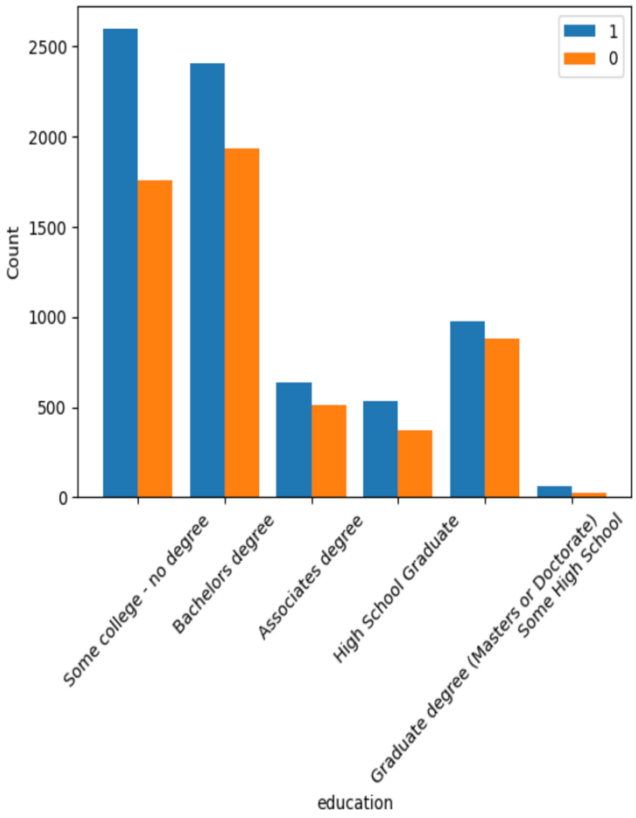

Y = 1
Y = 0
Count
never
less1
4~8
1~3
gt8
nan
CoffeeHouse
nan
4~8
1~3
gt8
less1
never
CarryAway
4~8
1~3
less1
gt8
nan
never
RestaurantLessThan20
1~3
less1
never
gt8
4~8
nan
Restaurant20To50
1
toCoupon_GEQ5min
0
1
toCoupon_GEQ15min


Y = 1
Y = 0
Count
toCoupon_GEQ25min
direction_same
direction_opp


1
0
Count
nan
Scooter and motorcycle
crossover
Mazda5
do not drive
Car that is too old to install Onstar :D
car


1
0
Count
37500 – 49999
62500 – 74999
12500 – 24999
75000 – 87499
50000 – 62499
25000 – 37499
$100000 or More
87500 – 99999
Less than $12500
income


1
0
Count
1000
800
600
400
200
0
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
occupation


1
0
Count
2500
2000
1500
1000
500
0
Some college - no degree
Bachelors degree
Associates degree
High School Graduate
Graduate degree (Masters or Doctorate)
Some High School
education