In Vehicle Coupon Recommendation Project Report

1. Problem Statement

Issuing coupons is a very effective promotional method in many different businesses. However, how and when to issue the coupon is very important. Here, we have an in-vehicle coupon recommendation data set collected from a survey on Amazon Mechanical Turk(source). The survey describes different driving scenarios including the destination, current time, weather, passenger, personal information, etc. and then asks the survey takers whether they would accept the coupon from a restaurant, coffee shop, bar etc. if they were the drivers.

Based on the data set, I used different tools in python to clean the data, utilized a variety of exploratory techniques to visualize and analyze the data. Then, I tried many different supervised machine learning models to predict whether a person would use the coupon recommended to him/her in a certain driving scenario.

After training a series of models, I finally tuned a Support Vector Classifier that was able to achieve an accuracy of 92%, which is 12% higher than I expected.

2. Data Wrangling

The data set, In Vehicle Coupon Recommendation, has a total 12684 observations and 26 columns, including 25 features and 1 label column representing whether the driver would accept the coupon. Figure 2-1 shows the basic information of the original data set. All features are categorical data in data type object(string) and int64.

In the data set, there were 74 duplicated columns, so I dropped them first. The column 'car' has 12576 missing values, which is about 99.1% of the total observations. Therefore, I removed this feature from the data set. Beside, the column 'Bar' has 107 missing values; 'CoffeeHouse' has 217; 'CarryAway' has 151, 'RestaurantLessThan20' has 130; 'Restaurant 20To50' has 189. The total number of observations with missing values was 4.8%. Since there was enough data for training the models, I dropped all those columns. Based on the description of the data set, the column 'toCoupon_GEQ5min' has mean 1.0 and standard deviation 0.0. That means this column only has value 1, so I drop this column. After the data cleaning process, the data set had 12007 observations and 24 columns.

The main tool I used in the Data Wrangling section is pandas.

```
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
```

Figure 2-1

3. Exploratory Data Analysis

First, I get the statistical information of the data set. Figure3-1 is the table of the statistical data. 57% of the observations accepted the coupon and 43% did not. Only 11.6% think the driving distance for using the coupon is more than 25 mins. 21.6% of people think the direction for using the coupon is the same as the destination, and 78.4% think in the opposite direction. Most people think that the coupon is always provided when the weather is not too cold, normally higher than 55 Fahrenheit.

	temperature	has_children	toCoupon_GEQ15min	toCoupon_GEQ25min	direction_same	direction_opp	Y
count	12007.000000	12007.000000	12007.000000	12007.000000	12007.000000	12007.000000	12007.000000
mean	63.301408	0.408845	0.559507	0.116266	0.215957	0.784043	0.568418
std	19.131641	0.491641	0.496467	0.320556	0.411502	0.411502	0.495317
min	30.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	55.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
50%	80.000000	0.000000	1.000000	0.000000	0.000000	1.000000	1.000000
75%	80.000000	1.000000	1.000000	0.000000	0.000000	1.000000	1.000000
max	80.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 3-1

Second, I tried a histogram plot to see the distribution of each feature. The interesting fact I found was that the distribution of 'direction_same' and the 'direction_opp' are the same. See (Figure 3-1) Then I tried a category plot on the 'direction_same' feature hue by 'direction_opp', and found that all the value 0 in 'direction_same' are corresponding to value 1 in 'direction_opp' and all the value 1 in 'direction_same' are corresponding to value 0 in 'direction_opp'. This means that these two features are indicating the same thing. (See Figure 3-2) Therefore, I dropped the 'direction_opp' column.

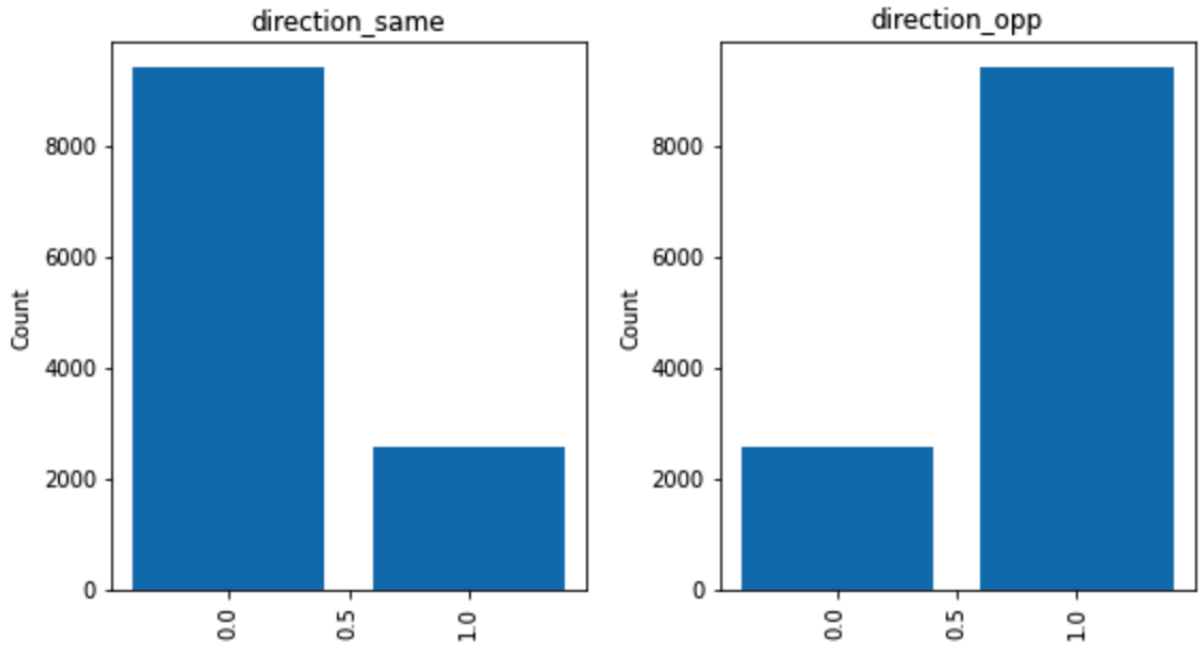


Figure 3-1

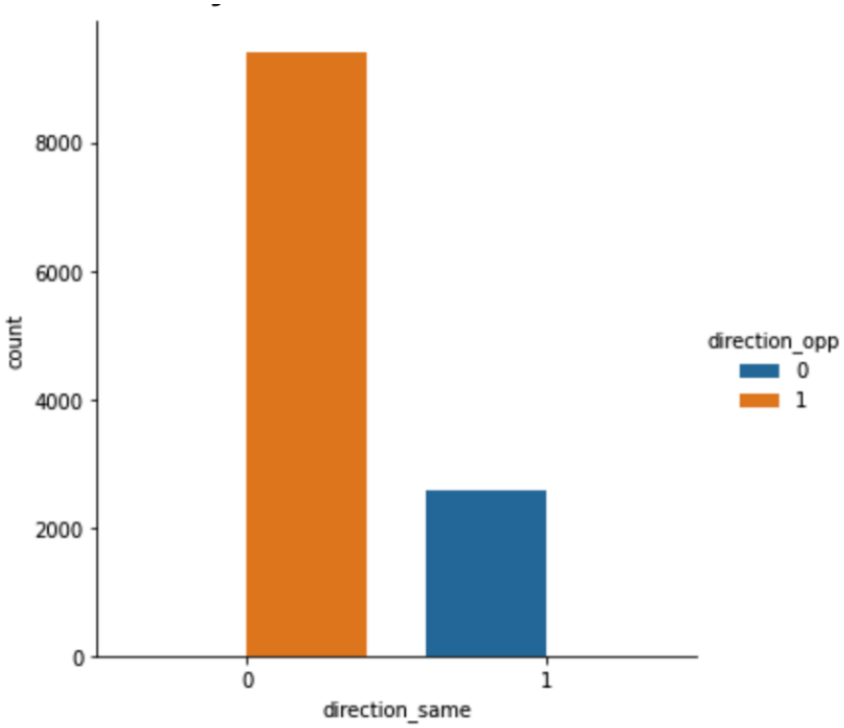


Figure 3-2

Third, I applied count plot on the the data set hue by the label ‘Y’, and found that people more likely accept the coupons when they are going to some not urgent place, or going with their friend, or in sunny day, or in lower temperature, or in the afternoon, or the coupon is less than $20, or coupon is for takeaway, or the coupon expire in a day, or the location is near. Also, People who are younger(< 30 years old), single, and don't have children are more likely to accept the coupons.

After this process, the data set has 12007 observations and 23 columns. The main tools I used in this section are numpy, pandas, seaborn, matplotlib.

4. Pre-processing & Training Data Development

Since all the features are in categorical type, I wanted to create dummy variables on all the features using feature engineering technique.

In this section, I created 113 columns from 22 features. The main tools are pandas, numpy.

5. Model Selection

In this section, I tested 6 different supervised learning models, including: logistic regression, random forest, k-nearest neighbors, support vector classifier, Naive Bayes, and gradient boosting. To handle the data imbalance problem, I used f1-score as the metrice for the models. To optimize the hyperparameters for each model, I used grid search cross validation.

Classification Report

Logistic Regression

	precision	recall	f1-score	support
0	0.64	0.57	0.60	1028
1	0.70	0.75	0.73	1374
accuracy			0.68	2402
macro avg	0.67	0.66	0.67	2402
weighted avg	0.67	0.68	0.67	2402

Random Forest

	precision	recall	f1-score	support
0	0.71	0.55	0.62	1028
1	0.71	0.83	0.77	1374
accuracy			0.71	2402
macro avg	0.71	0.69	0.70	2402
weighted avg	0.71	0.71	0.71	2402

K-Nearest Neighbors

	precision	recall	f1-score	support
0	0.68	0.60	0.64	1028
1	0.73	0.79	0.76	1374
accuracy			0.71	2402
macro avg	0.70	0.70	0.70	2402
weighted avg	0.71	0.71	0.71	2402

Figure 5-1

Support Vector Classifer

	precision	recall	f1-score	support
0	0.93	0.87	0.90	1028
1	0.90	0.95	0.93	1374
accuracy			0.92	2402
macro avg	0.92	0.91	0.91	2402
weighted avg	0.92	0.92	0.92	2402

Bernoulli Naive Bayes

	precision	recall	f1-score	support
0	0.60	0.57	0.59	1028
1	0.69	0.71	0.70	1374
accuracy			0.65	2402
macro avg	0.65	0.64	0.65	2402
weighted avg	0.65	0.65	0.65	2402

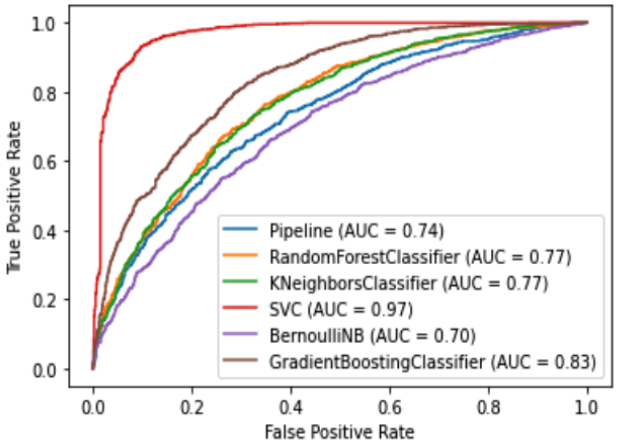
Gradient Boosting

	precision	recall	f1-score	support
0	0.74	0.67	0.70	1028
1	0.77	0.82	0.80	1374
accuracy			0.76	2402
macro avg	0.76	0.75	0.75	2402
weighted avg	0.76	0.76	0.76	2402

Figure 5-2

Figure 5-1 and Figure 5-2 above are the results of the models using the best hyperparameter set from grid search cross validation. All the models have a higher f1-score on label 1 than babel 0, which means that all of them do better on predicting positive values. Bernoulli Naive Bayes model has the lowest accuracy, which is 0.65. The Random Forest has the lowest recall score on label 0, which is 0.55. This means that it cannot predict the negative item very well.

The Support Vector Classifier model is the best model, it has the highest scores on every metric. The accuracy is 0.92, which is at least 14% higher than other models. Even though the f1-score of label 1 is higher than label 0, they are very close. The Support Vector Classifier model does the best on optimizing true negatives compared to the five other models with f1-score. Figure 5-3 and Figure 5-4 are the plot about ROC-AUC values and ROC-AUC scores of all models. The Support Vector Classifier got much better results than other models, which are AUC=0.97 and ROC-AUC=0.91. Therefore, my final models will be the Support Vector Classifier model. The optimized hyperparameters from grid search are gamma='scale',kernel='poly'.



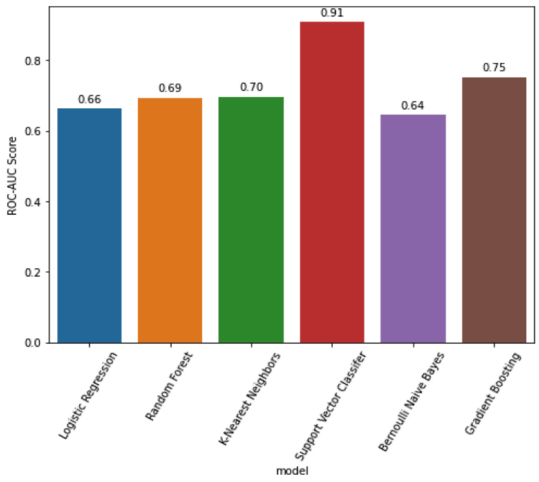


Figure 5-3 Figure 5-4

The main tools I used in the model selection section are pandas, numpy, skiearn.

6. Final Thought

About this project, I have two ideas for further improvement and research.

First, since this survey is for general catering business, I think the researchers can also ask people their food and drink preferences, what area they live in, and how much they usually spend on meal, drinks.

Second, I think a similar research can be applied to customers in a big shopping mall. There is no doubt that by learning the behavior of the customers can help increase the revenue. The survey takers are more easy to get in the shopping mall because they are more free. Last but not least, the survey would be more accurate or less noisy since people in the shopping mall are the exact target population.