In Vehicle Coupon Recommendation – Multiclass Classification

Group 8
Siyi Yu
Roswita Hede
Aishwarya Kavitkar
Priyanshi Rastogi

Table of Contents

Introduction

1.Project Definition

From retailers to manufacturers to small businesses, everyone uses discount coupons in their marketing strategies to not only increase sales but also improve customer retention. While it doesn't guarantee consumers will profit it somehow gives us a reassuring and satisfying sense of "business". Therefore, it becomes crucial for companies to utilize this marketing strategy to achieve their sales and profit goals.

2.Problem Setting

The dataset is dedicated to recommending in-vehicle mobility system coupons to users, and we predict whether customers will accept the vouchers. Based on the dataset, only 56.84 % of drivers receive the coupon. This might imply that the profit is not optimal since it doesn’t fulfill the company's target or the KPI of the campaign.
In order to solve the problem, we need to predict the acceptance rate of the coupon.
Therefore, we decided to build a classification model that can accurately predict whether users will accept their coupons, while reducing marketing costs as much as possible.

Data Source: The data provided to us is part of a research effort in which a group of scientists and professionals studied a new classification model based on the Bayesian ruleset and collected this dataset from a survey on Amazon Mechanical Turk for benchmarking this new model. Because the survey is conducted with real people, it can be assumed that the dataset is from the real world.

Data Description

This project's dataset was taken from UCI Machine Learning Repository website (https://archive.ics.uci.edu/ml/datasets/in-vehicle+coupon+recommendation).
The dataset records twenty one different features to estimate the acceptance of the coupon.

Feature	Description	Measurement
Destination	Destination of the driver	Categorical
Passenger	The passengers in the car	Categorical
Weather	The weather when the vouchers are distributed	Categorical
Temperature	The temperature when the vouchers are distributed	Numerical
Time	The time when the vouchers are distributed	Categorical
Coupon	Coupon type offered by company	Categorical
Expiration	The expiration date of coupon	Categorical
Gender	The gender of the driver	Categorical
Age	The age of the driver	Categorical
Marital Status	The marital status of driver	Categorical
has_Children	Has children or not	Categorical
education	The education status of driver	Categorical
occupation	The occupation status of driver	Categorical
income	The income status of driver	Categorical
Bar	The frequency of going to bar every month	Categorical
CoffeeHouse	The frequency of getting coffee every month	Categorical
CarryAway	The frequency of getting take-away food every month	Categorical

RestaurantLessThan20	The frequency of going to restaurant with price less than $20 every month	Categorical
Restaurant20To50	The frequency of going to restaurant with price less than $20-$50 every month	Categorical
toCoupon_GEQ15min	Driving time for using the coupon is greater than 15 minutes	Categorical
toCoupon_GEQ25min	Driving time for using the coupon is greater than 25 minutes	Categorical
toCoupon_GEQ5min	Driving time for using the coupon is greater than 5 minutes	Categorical
direction_same	same direction as current destination	Numerical
direction_opp	opposite direction as current destination	Numerical
Y	Target variable whether the coupon is accepted or not	Categorical

Exploratory Data Analysis

1. Handling Null Values

Based on the table below, among the variables, the ‘Car’, ‘Bar’, ‘CoffeeHouse’, ’Carryaway’, ’RestaurantLessThan20’, and ‘Restaurant20to50’ contain null values.

Based on the descriptive analysis below, since more than 90% of ‘car’ variables are null variables, we decide to remove it from the dataset. In addition, we perform an imputation step to those variables by changing the null value with its mode.

```
destination                 0
passanger                   0
weather                     0
temperature                 0
time                        0
coupon                      0
expiration                  0
gender                      0
age                         0
maritalStatus               0
has_children                0
education                   0
occupation                  0
income                      0
car                     12502
Bar                       107
CoffeeHouse               217
CarryAway                 150
RestaurantLessThan20      129
Restaurant20To50          189
toCoupon_GEQ5min            0
toCoupon_GEQ15min           0
toCoupon_GEQ25min           0
direction_same              0
direction_opp               0
Y                           0
dtype: int64
```

2.Handling Duplicate Data

We also checked if there are duplications on the dataset. We found that there are 74 duplicates and eliminate those dataset.

```
df.duplicated().sum()
```

```
74
```

3.Features Transformation

Two categories in the dataset, namely 'age' and 'occupation', have more categories than the other features. Because of that, we decided to categorize it to be more simple. The 'age' has eight categories, which are changed into five categories, while the 'occupation' has twenty-five categories and are changed into eight categories. In addition, we combined three redundant variables (ToCoupon_GEQ5min, ToCoupon_GEQ15min, ToCoupon_GEQ25min) into one column as 'driving distance'.

4.Statistical Analysis

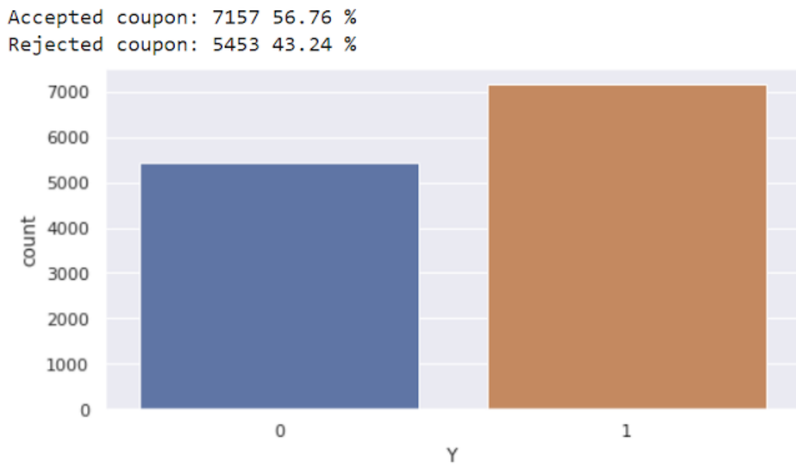
4.1 Descriptive Analysis

For the numerical variables, we do the descriptive analysis. The standard deviation of direction_opp and direction_same is equal. They might contain the same information, therefore in the data pre-processing, we will conduct correlation tests for those variables. The mean, max, min and quartile value of ToCoupon_GEQ5min are the same, which means the variable only has one unique value.

	temperature	has_children	toCoupon_GEQ5min	toCoupon_GEQ15min	toCoupon_GEQ25min	direction_same	direction_opp	Y
count	12610.000000	12610.000000	12610.0	12610.000000	12610.000000	12610.000000	12610.000000	12610.000000
mean	63.267248	0.414512	1.0	0.559794	0.116019	0.215543	0.784457	0.567565
std	19.153386	0.492657	0.0	0.496432	0.320260	0.411215	0.411215	0.495434
min	30.000000	0.000000	1.0	0.000000	0.000000	0.000000	0.000000	0.000000
25%	55.000000	0.000000	1.0	0.000000	0.000000	0.000000	1.000000	0.000000
50%	80.000000	0.000000	1.0	1.000000	0.000000	0.000000	1.000000	1.000000
75%	80.000000	1.000000	1.0	1.000000	0.000000	0.000000	1.000000	1.000000
max	80.000000	1.000000	1.0	1.000000	1.000000	1.000000	1.000000	1.000000

4.2 Distribution of Class

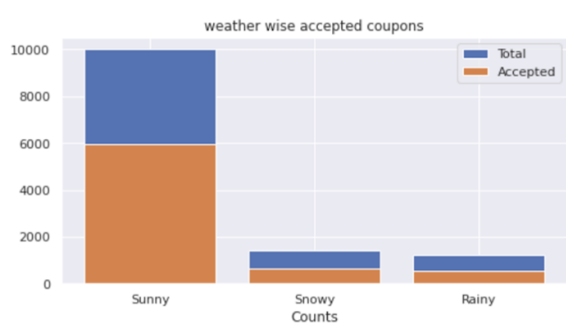
The ratio between class is balance (Class 1 (Accepted)= 56.76% and Class 0 (Rejected)= 43.24%.



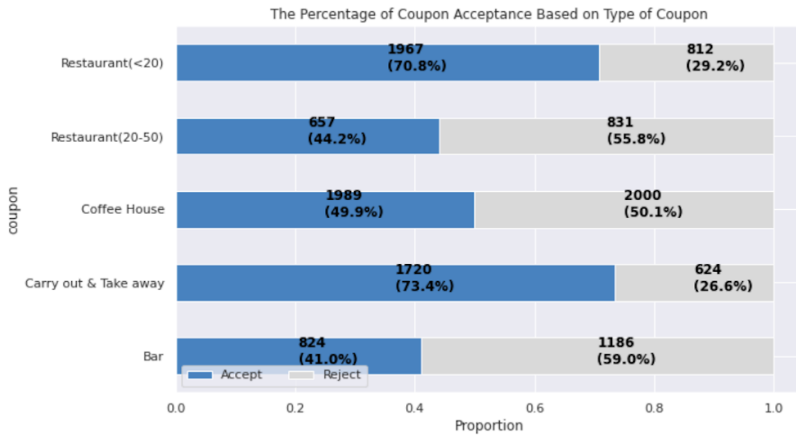
4.3 Univariate Analysis

Since all features are categorical, we want to look at the discrete distributions of various features with the target.

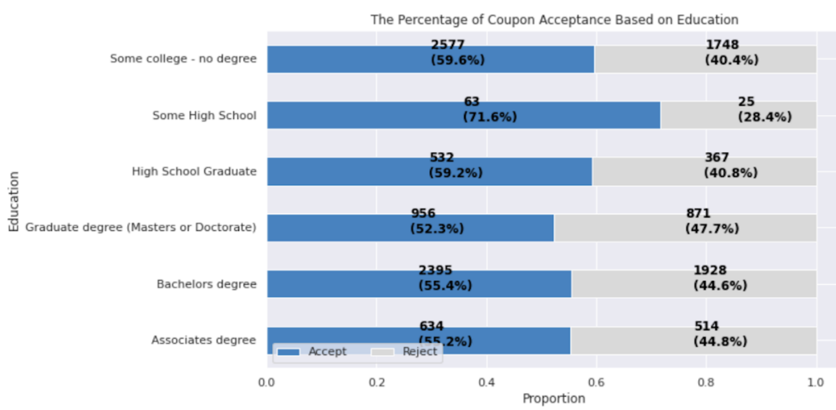
As clearly evident from the plot below, the bar plots show the weather and expiration date for accepting coupons. Most drivers receive the coupon when the weather is sunny and the coupon expiration date is longer.



In addition, we can see that most of the coupons accepted by consumers are coupons from cheap restaurants (restaurants with prices less than $20), carry-out and take-away. On the other hand, the customer is most likely to reject coupons from expensive restaurants and bars.

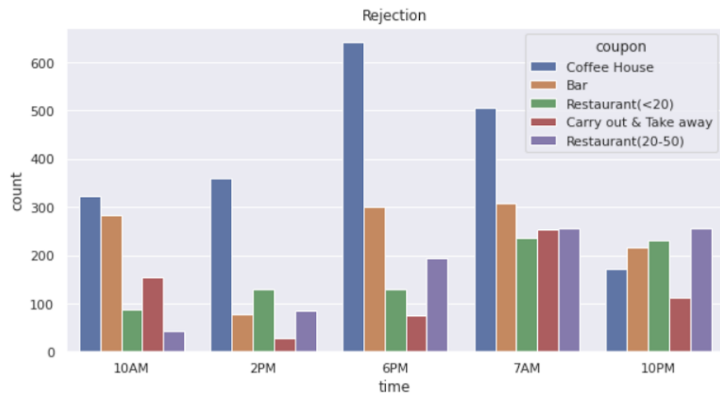


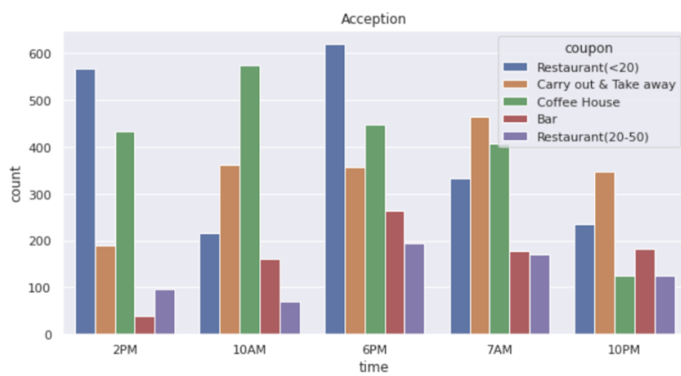
We are also looking at the driver's education and get insight that most customers who accept the coupon are high school students, followed by the high school graduate and college students.



4.4 Bivariate Analysis

We want to know the acceptance rate based on the time and the type of coupon. It can be seen from the graph that the driver mostly accepts vouchers from cheaper restaurants (Restaurants <$20), carry-out, and coffee houses at 10AM, 2PM, and 6PM.





Future Engineering

1.Label Encoding

The original dataframe contains both object and int data type. For better training our model, we need to convert categorical columns to numeric columns. After applying LabelEncoder, all columns are numerical.

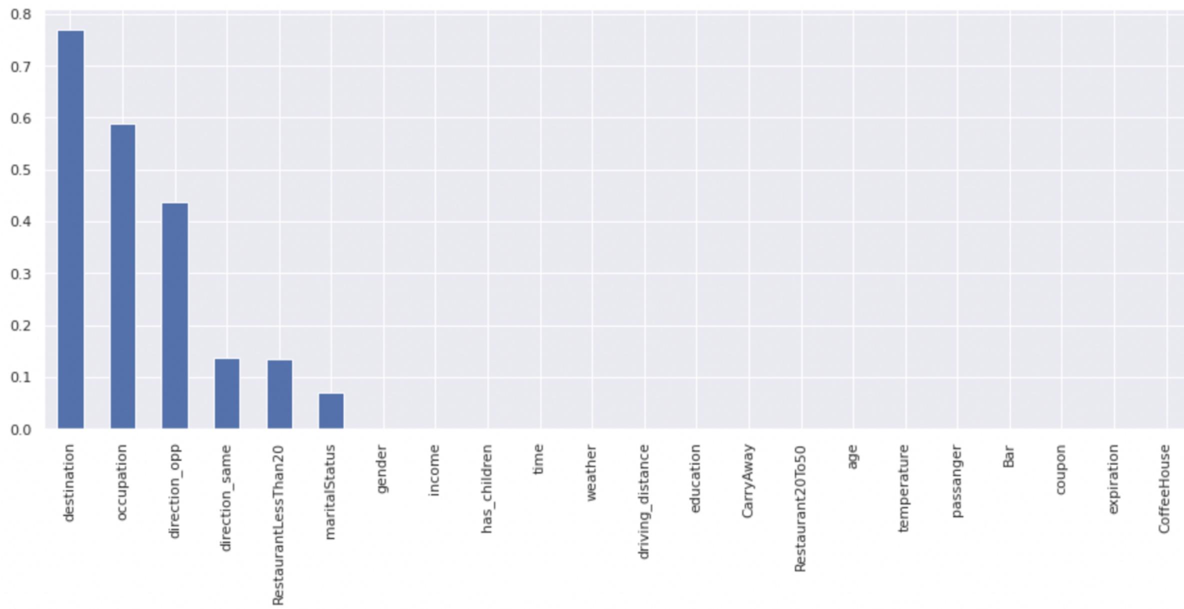
#	Column	Non-Null Count	Dtype
0	destination	12610 non-null	object
1	passanger	12610 non-null	object
2	weather	12610 non-null	object
3	temperature	12610 non-null	int64
4	time	12610 non-null	object
5	coupon	12610 non-null	object
6	expiration	12610 non-null	object
7	gender	12610 non-null	object
8	age	12610 non-null	object
9	maritalStatus	12610 non-null	object
10	has_children	12610 non-null	int64
11	education	12610 non-null	object
12	occupation	12610 non-null	object
13	income	12610 non-null	object
14	Bar	12610 non-null	object
15	CoffeeHouse	12610 non-null	object
16	CarryAway	12610 non-null	object
17	RestaurantLessThan20	12610 non-null	object
18	Restaurant20To50	12610 non-null	object
19	toCoupon_GEQ5min	12610 non-null	int64
20	toCoupon_GEQ15min	12610 non-null	int64
21	toCoupon_GEQ25min	12610 non-null	int64
22	direction_same	12610 non-null	int64
23	direction_opp	12610 non-null	int64
24	Y	12610 non-null	int64

dtypes: int64(8), object(17)

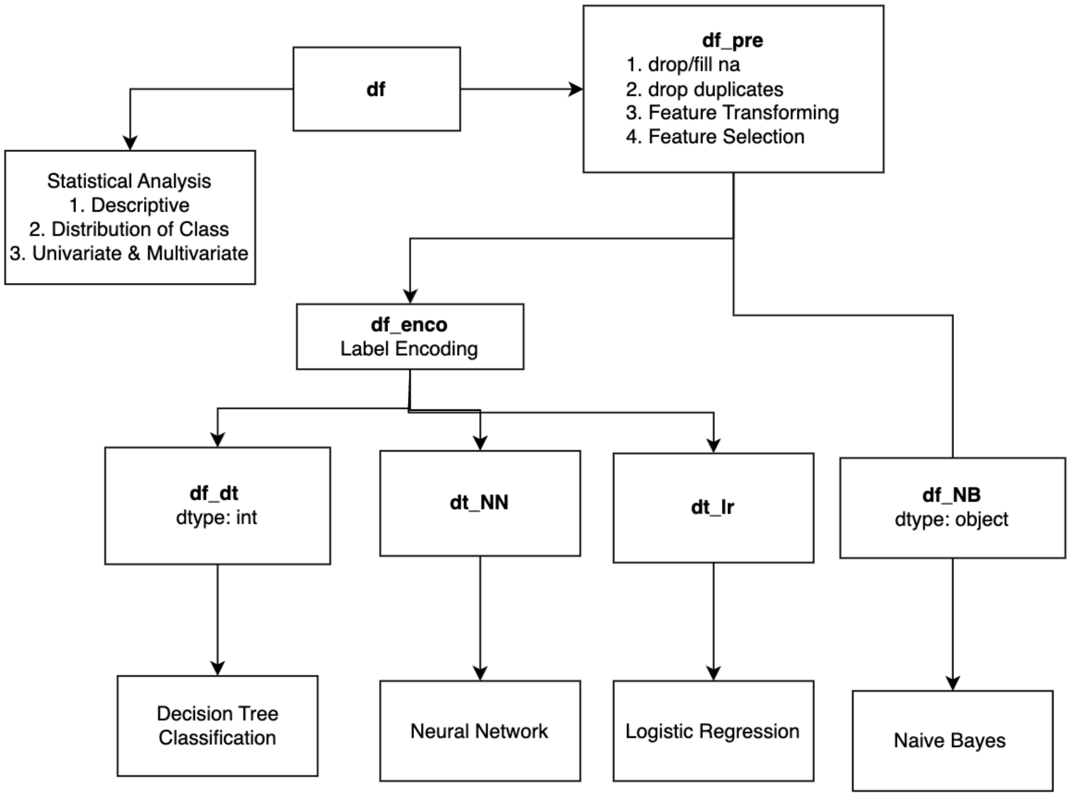
#	Column	Non-Null Count	Dtype
0	passanger	12610 non-null	int64
1	weather	12610 non-null	int64
2	temperature	12610 non-null	int64
3	time	12610 non-null	int64
4	coupon	12610 non-null	int64
5	expiration	12610 non-null	int64
6	gender	12610 non-null	int64
7	age	12610 non-null	int64
8	maritalStatus	12610 non-null	int64
9	has_children	12610 non-null	int64
10	education	12610 non-null	int64
11	occupation	12610 non-null	int64
12	income	12610 non-null	int64
13	Bar	12610 non-null	int64
14	CoffeeHouse	12610 non-null	int64
15	CarryAway	12610 non-null	int64
16	RestaurantLessThan20	12610 non-null	int64
17	Restaurant20To50	12610 non-null	int64
18	direction_same	12610 non-null	int64
19	Y	12610 non-null	int64
20	driving_distance	12610 non-null	int64

2.Feature Selection

Since the dataset has a higher dimension, in order to avoid overfitting, we performed a dimension reduction by looking at the correlation between predictors and target using the Chi-Square test and comparing the p-value test. If the P-value is higher, then the variable is less important (it has a lower correlation with the target). Based on the bar graph below, the p-value of ‘destination’ has a higher p-value; we decided to remove it from the dataset.



3.Dataframe Selection



Machine Learning Algorithm

Splitting the dataset:
Before performing the algorithm, the in-vehicle coupon recommendation dataset is splitted into 75% training data and 25% test data.

1.Decision Tree

- 1.1 Splitting Method:
- Continuous Target Variable:
- Reduction in Variance
- Categorical target variable:
- Information Gani
- Gani Impurity
- Chi-Square

Since our target variables are categorical 0 and 1, we use information gani to split the dataset. Information Gain is a measure of how much information a feature provides about a class. Information gain helps to determine the order of attributes in the nodes of a decision tree.

$$Gain = E_{parent} - E_{children}$$

Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations. It determines how a decision tree chooses to split data. If the sample is completely homogeneous the entropy is zero and if the sample is equally divided it has entropy of one.

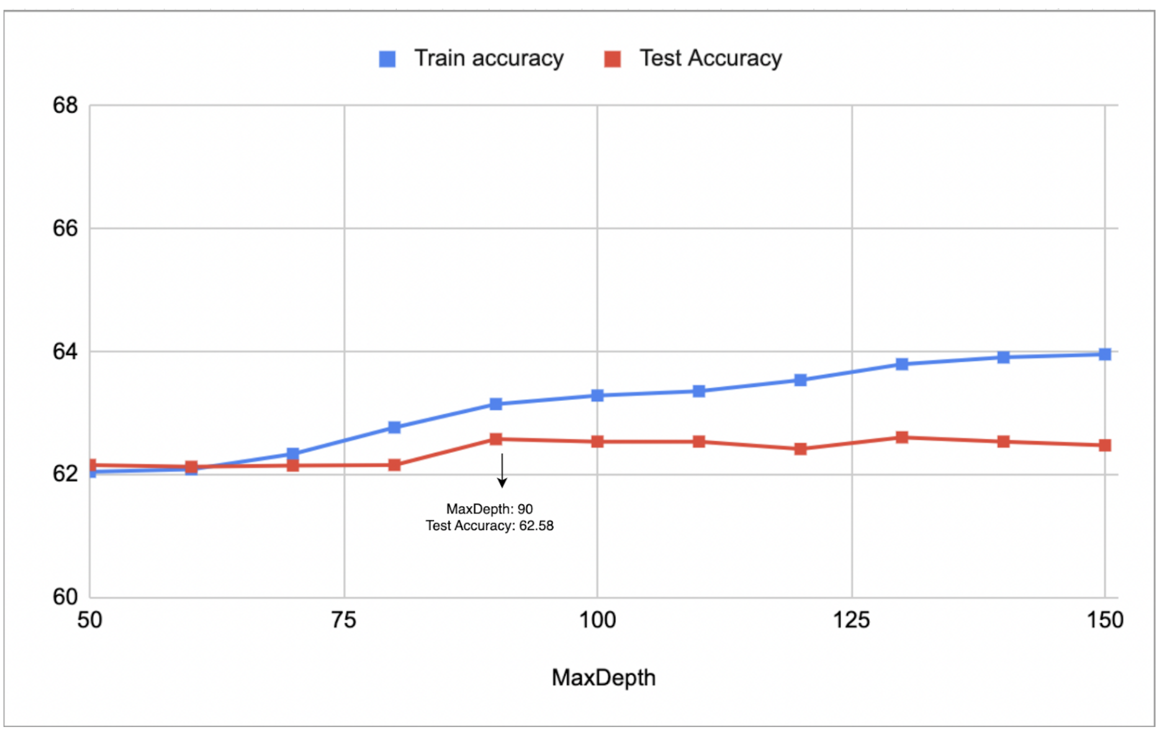
$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

1.2 Splitting Steps

Step1: Calculate the entropy of parent node independently
Step2: Calculate the entropy of each split using the weighted average entropy of child nodes
Step3: Choose the split with maximum Gani information
Step4: Repeat iteratively until finishing constructing the whole tree.

1.3 Max Depth

The desired max_depth of decision tree is 90 with accuracy 62.58%. If set max_depth too high, then the decision tree might simply overfit the training data without capturing useful patterns. This will cause testing errors to increase. If set max_depth too low, that is not good as well. Might be giving the decision tree too little flexibility to capture the patterns and interactions in the training data. This will also cause the testing error to increase.



1.4 Result

Decision Tree Classification Report

	precision	recall	f1-score	support
0	0.70	0.64	0.67	729
1	0.56	0.63	0.59	532
accuracy			0.63	1261
macro avg	0.63	0.63	0.63	1261
weighted avg	0.64	0.63	0.63	1261

2.Naive Bayes Classifier

2.1 Introduction

Naive Bayes algorithm is one of the supervised algorithms used to estimate the classification model. This algorithm is based on Bayes' theorem and has a strong assumption that the features are conditionally independent given the target class.
Bayes's theorem uses prior knowledge to determine the probability of an event. The equation for Bayes Theorem is

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Because, Naive Bayes classifier is based on the Bayes's theorem, than it can be written by

$$P(y_i|x_1,\ x_2,\ \dots,\ x_n) = \frac{P(x_1, x_2, \dots, x_n|y_i)P(y_i)}{P(x_1, x_2, \dots, x_n)}$$

Where $P(y_i|x_1,\ x_2,\ \dots,\ x_n)$ is the posterior probalility of target given the predictors, $P(y_i)$ is the prior probability of target and $P(x_1,\ x_2,\ \dots,\ x_n|y_i)$ is the likelihood and $P(x_1,\ x_2,\ \dots,\ x_n)$ is the marginal probability (probability of predictor)
Naive Bayes Classifier is a simple and faster algorithm compared to generative classifiers as well as robust to noise in the data. Although Naive Bayes Classifier can work with less data, finding the best parameter requires a large amount of data.

2.2 Naive Bayes Classifier Steps

The dataset that is used in this project is categorical, therefore, we perform a naive bayes classifier. Below are the steps to perform Naive Bayes Classifier.

1. Calculate the prior probability of class $P(Y)$
2. Calculate the likelihood $P(X|Y)$
3. Calculate the predictor prior $P(X)$
4. Calculate the posterior probability $P(Y|X)$
5. Evaluate the accuracy, F1 score and recall of the final model.

2.3 Result

In order to evaluate the performance of the Naive Bayes Classifier, we use accuracy and F1 score to assess the model performance. Accuracy is the proportion of prediction that is classified correctly by the model. At the same time, F1 scores are used to evaluate the preciseness and robustness of the model and are usually more accurate than accuracy if the class has imbalanced data. Since our data is balanced, then both measurements can be used. In addition, we also want to look at the confusion matrix of the prediction model. The confusion matrix is one of the metrics to evaluate how the classification algorithm performs. The confusion matrix of the Naive Bayes classifier is

```
Confusion Matrix

[[ 744  607]
 [ 447 1355]]
```

From the confusion matrix above, we got the result that out of the 1351 actual instances of ‘Accepted’, the model correctly predicted 744 of them, and out of 1802 actual instances of ‘Rejected’, it predicted 1355 of them. In addition, out of 3153 coupons, the classification model estimated 2099 correctly.

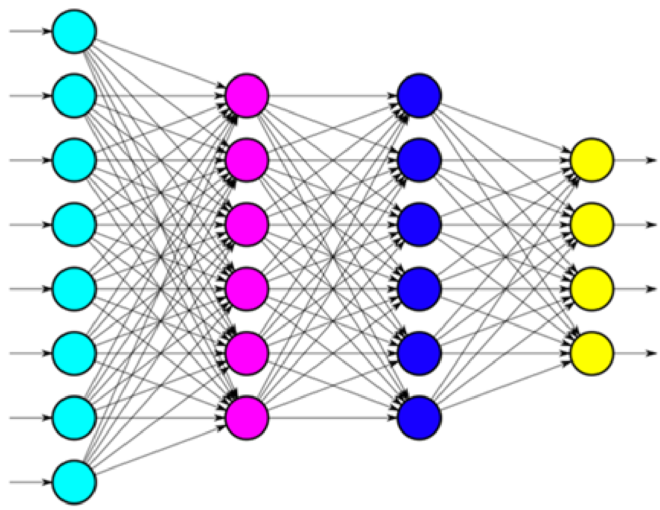
Naive Bayes Classification Report

	precision	recall	f1-score	support
Accept	0.69	0.75	0.72	1802
Reject	0.62	0.55	0.59	1351
accuracy			0.67	3153
macro avg	0.66	0.65	0.65	3153
weighted avg	0.66	0.67	0.66	3153

Based on the classification report above, the accuracy of the classifier model is 67%. It means the model predicts whether the driver will accept or reject the coupon correctly by 67%. It can be seen that the F1 score of the classifier model is 0.72, which means the model is good.

3. Artificial Neural Network

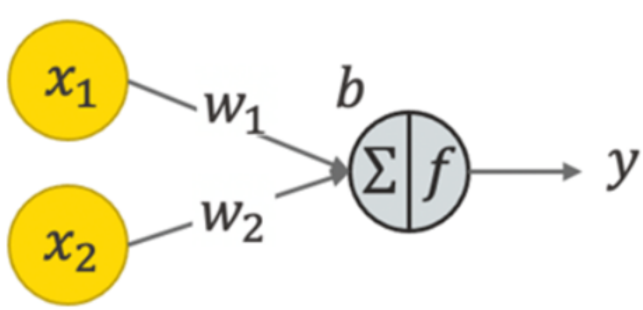
3.1 Introduction



Artificial Neural Network is made up of layers . The layers are connected with nodes which in turn are interconnected and they contain the Activation Function .

Input Layer :

The purpose of the input layer is to receive input the values of the explanatory attributes . Most of the times the number of input nodes is equal to the number of explanatory variables . “input layer” gives the pattern with the help of which the network communicates with the “hidden” layers. A basic structure of a single node is displayed below :



Hidden Layer :

The hidden layer applies transformations to the input values within the network. In hidden layer, the actual processing is done via a system of weighted ‘connections’. There may be one or more hidden layers. The values entering a hidden node multiplied by weights, a set of predetermined numbers stored in the program. The weighted inputs are then added to produce a single number.

Output Layer :

The hidden layers then link to an ‘output layer‘. Output layer receives connections from hidden layers or from input layer. It returns an output value that corresponds to the prediction of the response variable. The active nodes of the output layer combine and change the data to produce the output values. The ability of the neural network to provide useful data manipulation lies in the proper selection of the weights. This is different from conventional information processing.

3.2 Artificial Neural Network Parameter

3.2.1. ADAM as gradient descent optimization algorithm
3.2.2. Activation function :
Output layer : sigmoid/ hard_sigmoid - binary classification . Sigmoid or Logistic Activation Function: Sigmoid function maps any input to an output ranging from 0 to 1. For small values (<-5), sigmoid returns a value close to zero, and for large values (>5) the result of the function gets close to 1. Hidden layers: relu or tahn 3. Metrics: accuracy as the data is balancedLoss function: Binary cross entropy . Computes the cross-entropy loss between true labels and predicted labels. We use this cross-entropy loss when there are only two label classes (assumed to be 0 and 1). Number of epochs - number times that the learning algorithm will work through the entire training dataset. Batch_size - number of training examples utilized in one iteration Number of layers: 2-3 (empirically) . Adding more layers to NN can help addressing bias, not variance.Regularization parameters/ Cross-Validation or Early Stopping to reduce overfitting .Validation split 0.2, CV with 5 folds

3.3 Result

3.3.1 Best Model 1

Epoch : 50
Batch Size : 512
Training Accuracy : 64.89% (0.84%)
Test Accuracy : 62.36% (2.36%)
Executing Time : 4.52835097200022
Loss – Accuracy :

0

99/99 [==============================] - 0s 2ms/step - loss: 0.6037 - accuracy: 0.6717

1

99/99 [==============================] - 0s 2ms/step - loss: 0.6129 - accuracy: 0.6597

2

99/99 [==============================] - 0s 2ms/step - loss: 0.6212 - accuracy: 0.6486

3

99/99 [==============================] - 0s 2ms/step - loss: 0.6178 - accuracy: 0.6489

4

99/99 [==============================] - 0s 2ms/step - loss: 0.6154 - accuracy: 0.6527

5

99/99 [==============================] - 0s 2ms/step - loss: 0.6156 - accuracy: 0.6606

6

99/99 [==============================] - 0s 1ms/step - loss: 0.6049 - accuracy: 0.6622

7

99/99 [==============================] - 0s 2ms/step - loss: 0.5904 - accuracy: 0.6876

8

99/99 [==============================] - 0s 1ms/step - loss: 0.6168 - accuracy: 0.6441

9

99/99 [==============================] - 0s 1ms/step - loss: 0.6192 - accuracy: 0.6438

Graph :

3.3.2 Best Model 2

Epoch : 9
Batch Size : 256
Training Accuracy : 61.94% (1.46%)
Test Accuracy : 58.48% (0.08%)
Executing Time : 2.128152268999999
Loss – Accuracy :
0

99/99 [==============================] - 0s 843us/step - loss: 0.5992 - accuracy: 0.6755

1

99/99 [==============================] - 0s 934us/step - loss: 0.6151 - accuracy: 0.6521

2

99/99 [==============================] - 0s 951us/step - loss: 0.5928 - accuracy: 0.6752

3

99/99 [==============================] - 0s 815us/step - loss: 0.6211 - accuracy: 0.6457

4

99/99 [==============================] - 0s 813us/step - loss: 0.6053 - accuracy: 0.6603

5

99/99 [==============================] - 0s 815us/step - loss: 0.6103 - accuracy: 0.6597

6

99/99 [==============================] - 0s 851us/step - loss: 0.6068 - accuracy: 0.6695

7

99/99 [==============================] - 0s 866us/step - loss: 0.6207 - accuracy: 0.6457

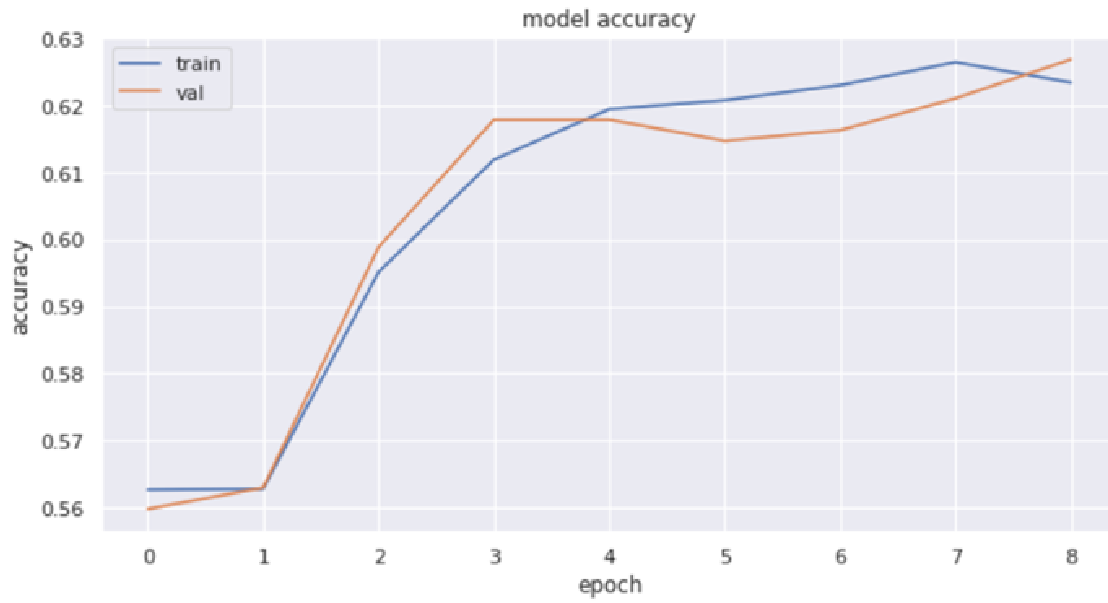
8

99/99 [==============================] - 0s 840us/step - loss: 0.5994 - accuracy: 0.6794

9

99/99 [==============================] - 0s 864us/step - loss: 0.6187 - accuracy: 0.6537

Graph :



3.4 Conclusion

We monitored the validation loss along with the training loss by changing the number of epochs and the batch size . So we trained the data for 9 , 50 and 100 epochs with a batch size of 512 , 256 , 128 and 64 . The validation loss is increasingly fluctuating and to balance it off we increased the batch size and the best models which came out were where the epoch is 9 and 50 along with the batch size of 256 and 512 respectively . The Loss – Accuracy has been monitored and it is increasing which shows that the model is generalizing well .

4. Logistic Regression

4.1 Introduction

Logistic Regression is one of the supervised learning algorithms used for classification and the most common case is binary logistic regression where the outcome is binary (yes or no).Mathematically, a logistic regression model predicts P(Y=1) for a class given X.

$$P = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots\beta_n X_n)}} = \frac{1}{1+e^{-(\beta_0+\sum \beta_i X_i)}}.$$

Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same −

1. All the input variables should be numericalThe target variables must be binary always and the desired outcome is represented by the factor level 1.
2. There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
3. We must include meaningful variables in our model.
4. We should choose a large sample size for logistic regression.

4.2 Logistic Regression Steps

The dataset that is used in this project is categorical, therefore, we encoded the data using ordinal encoding. Below are the steps to perform Logistic Regression:

1. Create a class and declare the required parameters. Parameters: maxIteration, epsilon, learning rate.
2. Calculate Predict function
3. Calculate Sigmoid function
4. Calculate Cost function
5. Calculate Gradient
6. Evaluate the accuracy, F1 score for train and test data.

4.3 Result

To evaluate the performance of the logistic regression model, we checked the accuracy of the training and test data. For this particular dataset, the accuracy is 57% on the training data and 55% on the test data. This means that the model has a 55% chance of predicting whether the driver correctly accepts or rejects the coupon. The model has an f1 value of 0.0537%, which is terrible, so we tried different algorithms to get better predictions.

```
weather                    -1232.000742
temperature                  -14.295561
time                         282.349682
coupon                      -874.680537
expiration                  3150.334295
gender                      -751.494809
age                           -5.202212
maritalStatus               -117.948137
has_children                 406.878712
education                   -513.661597
occupation                  -396.919064
income                      -190.610312
Bar                          -51.632982
CoffeeHouse                  555.432924
CarryAway                    109.638355
RestaurantLessThan20         -86.646859
Restaurant20To50             -10.813246
direction_same             -1322.192119
driving_distance            -121.837517
dtype: float64
The train F1 score of the model is 0.05378858746492049
The train recall score of the model is 0.02816556453588048
The train precision score of the model is 0.5958549222797928
```

Also, we tried to see how the accuracy varied by doing different splits on the data but found that the accuracy remains constant for all different split sizes. We also tried changing the precision by giving the parameter various different values. We found that the lower the learning rate and the higher the iteration value, the highest accuracy.

Sample	Learning Rate	Maximum Iteration	Test Accuracy	Train Accuracy	Executing time
1.	0.3	10,000	0.5098350253807107	0.5194544301120745	48 seconds
2.	0.1	10,000	0.5126903553299492	0.5195601607105096	50 seconds
3.	0.05	1000	0.5717005076142132	0.5710509621484458	9 seconds
4.	0.01	1000	0.5104695431472082	0.5220976950729541	5seconds

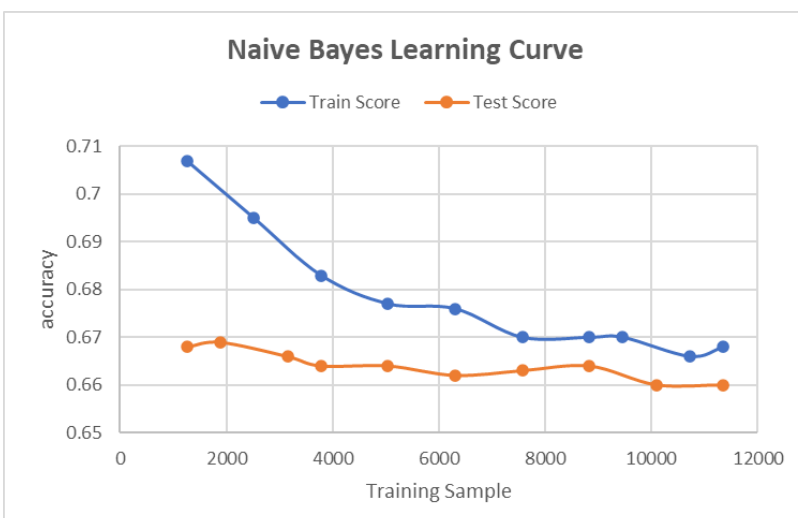
Bias Variance Tradeoff

The important sources of learning errors in a model are noise, bias, and variance. It is essential to evaluate those three to decide whether the model is good.
The below equation defines Bias Variance Tradeoff

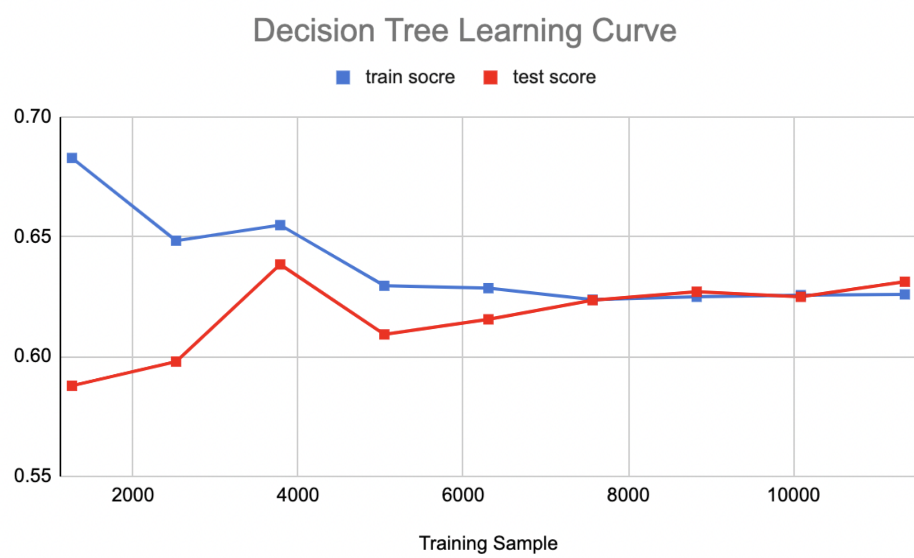
$$Expected\ Error\ = E(h_D(x) - \overline{f}(x))^2 + E(\overline{y}(x) - y)^2 + E(\overline{h}(x) - \overline{y}(x))^2$$

Bias-variance helps to evaluate the ability of the model to compute the difference between actual and predicted values. The model is good if it has low bias and low variance. High bias means the outputs of the prediction model are too far from the actual value, which means the model has poor performance. It is overfitting if the model cannot capture information from the dataset and has high bias and low variance. On the other hand, it is underfitting if it performs well on the train set but poorly on the test set (low bias, high variance).
We are using a learning curve to evaluate the impact of adding more training and whether the model prediction suffers more from a variance error or a bias error.

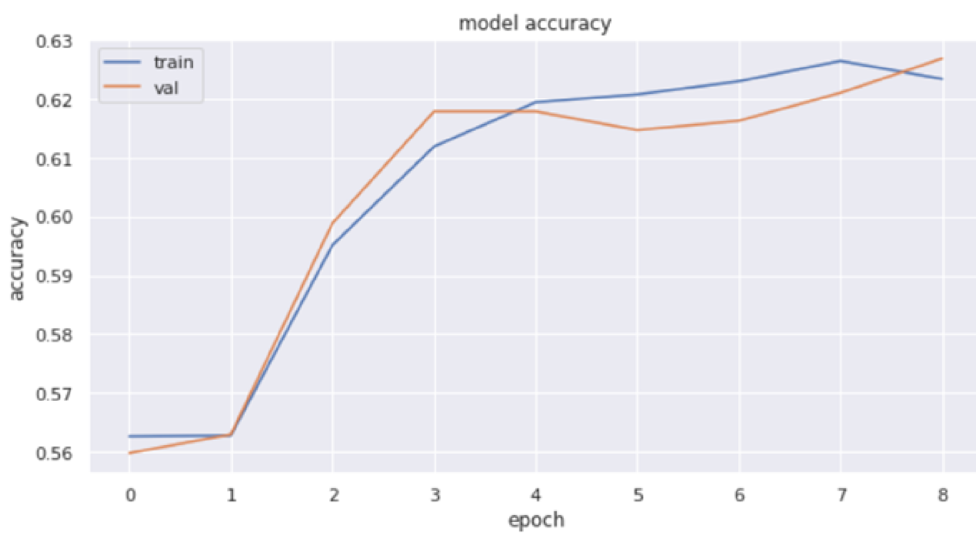
Learning Curve
Based on the line graph below, it can be seen that all the classification models have low variance and high bias. It means the model does have signs of underfitting.
Based on the learning curve and the mean accuracy, we could conclude that the Naive Bayes Classifier has the best ability to predict whether the driver will accept the voucher or not, given the predictors.



Logistic Regression Learning Curve
Train Score
Test Score
62
60
58
56
54
52
50
2000
4000
6000
8000
10000
Traning Sample



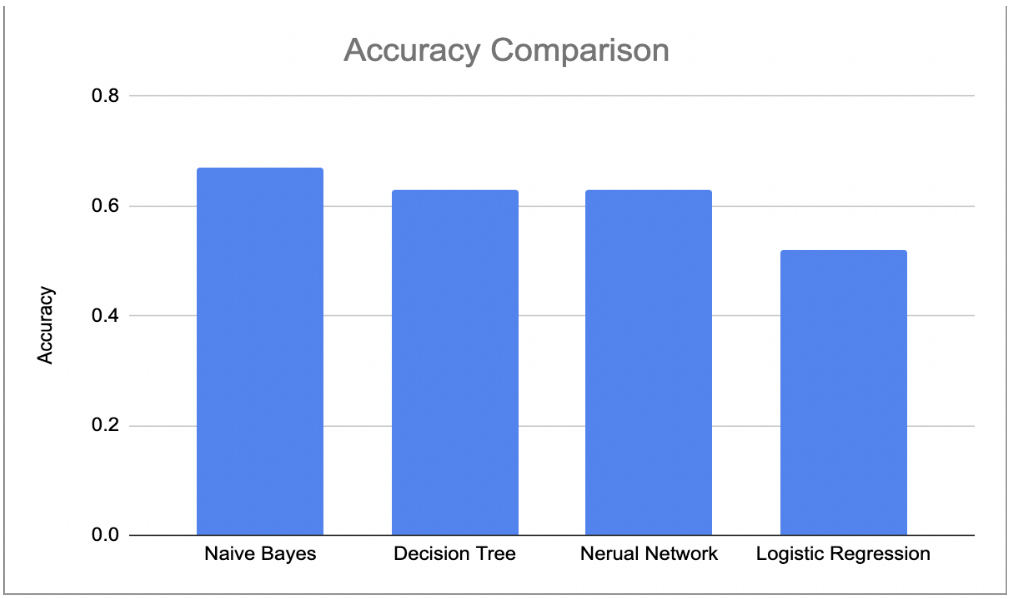
Decision Tree Learning Curve
train socre
test score
0.70
0.65
0.60
0.55
2000
4000
6000
8000
10000
Training Sample



Model	Mean Accuracy (Train)	Mean Accuracy (Test)
Naïve Bayes Classifier	0.6782	0.664
Logistic Regression	0.546	0.5456
Decision Tree	0.6382	0.6172
Neural Network (9 Epoch)	0.588	0.5924
Neural Network (50 Epoch	0.6314	0.6365

Conclusion:

The nature of dataset presented challenges that need to be managed in order to generate general, usable predictive models. Feature engineering and encoding was the crucial since numerical data is required for performing the selected models and the data present in the original dataset is categorical. We developed four classification models to understand which one works the best for this particular dataset. Based on the learning curve above for four models, it can be seen that all the classification models have low variance and high bias. It means the model does have signs of underfitting. From the mean accuracy and runtime table, we could conclude that the Naive Bayes Classifier has the best ability to predict whether the driver will accept the voucher or not, given the predictors.



	Run Time
Naive Bayes	2.86 secs
Decision Tree	19.43 secs
Logistic Regression	48.11 secs
Neural Network model1	4.5 secs
Neural Network model2	2.12 secs

Citation:

https://hshan0103.medium.com/understanding-bias-variance-trade-off-from-learning-curve-a64b4223bb02

GitHub Link:

https://github.com/yusykf7/In-Vehicle-Coupon-Recommendation-Multiclass-Classification