TÉCNICO LISBOA

STATISTICAL METHODS IN DATA MINING

In Vehicle Coupon Recommendation — A Machine Learning Classification Case Study

Autores:

José Pedro Lopes (100001) jose.pedro.rodrigues.lopes@tecnico.ulisboa.pt
João Medeiros Loureiro (99987) joao.miguel.loureiro@tecnico.ulisboa.pt
Tiago Costa (100094) tiagomascosta@tecnico.ulisboa.pt
Miguel Luís Rente Lourenço (100044) miguel.rente.l@tecnico.ulisboa.pt

Grupo 2

2022/2023 – 1º Semestre, P1

Conteúdo

1 Objective

Forecasting whether a customer will approve or decline a coupon poses a challenging task, and it's not feasible to simply suggest it to all customers due to cost considerations. Consequently, in this scenario, we will make predictions regarding a customer's likelihood to accept or decline the provided coupon, taking into account the customer's profile and past interactions. If the customers accept the coupon are labeled as Y=1 and if the customers reject the coupon are labeled as Y=0, so this problem can be posed as a binary class classification problem.

The dataset used for training, validation, and testing are subsets of data that was collected via a survey on Amazon Mechanical Turk with different driving scenarios (DOI: 10.24432/C5GS4P) has 26 features, 12684 instances and has missing values (NULL).

2 Exploratory Data Analysis

Initially, we engage in an exploratory data analysis (EDA), which is a pivotal phase within the data analysis journey, facilitating comprehension of our data, detection of patterns, and revelation of valuable insights.

Our features are possible to divide into different categories: User, contextual, coupon, and target attributes.

This paper will use the same meaning for features as the ones used in the paper "A Bayesian Framework for Learning Rule Sets for Interpretable Classification" [1].

The first step involved examining the frequency of the Y data. Based on this analysis, it can be determined that the dataset is partially balanced, with approximately 56.84% of acceptance class labels and around 43.16% of reject class labels.

2.1 Data Cleaning

Cleaning our data is crucial in order to reduce noise and irrelevant values of the dataset.

First, we noticed that the variable "toCoupon_GEQ5min"only has a categorical value of one type. This means that it doesn't add any information to our problem so we chose to remove this feature.

Secondly, we checked for missing values. This analysis is presented in the table 1.

Feature	Missing Values (%)
Car	99.15
Bar	0.84
Carry Away	1.19
Coffee House	1.71
Restaurant 20 to 50	1.49
Restaurant less than 20	1.02

Tabela 1: Missing Values in Data

As shown above the feature "Car"has 99,15% of data missing which means that little information is provided so it's better to remove it completely from the data to avoid inaccuracies in the modeling.

The rest of the missing values are residual for the other features. However, the total number of rows with at least one missing feature is 605, corresponding to approximately 5%. In order to not lose any data points and maximize the information it can provide, we decided to fill the missing values with the mode of the rest of the data.

Finally, we ensured that there were no duplicate features in our data.

2.2 Correlation Analysis

Since we're dealing with categorical variables an ordinary correlation matrix wouldn't work since it is only used for numerical variables. With this in mind, we decided to perform a Cramérs's V analysis. This metric measures the correlation between categorical variables. We ran this analysis to understand which features are highly correlated. We also defined a lower threshold of 0.40, since values above it can indicate a strong correlation between features.

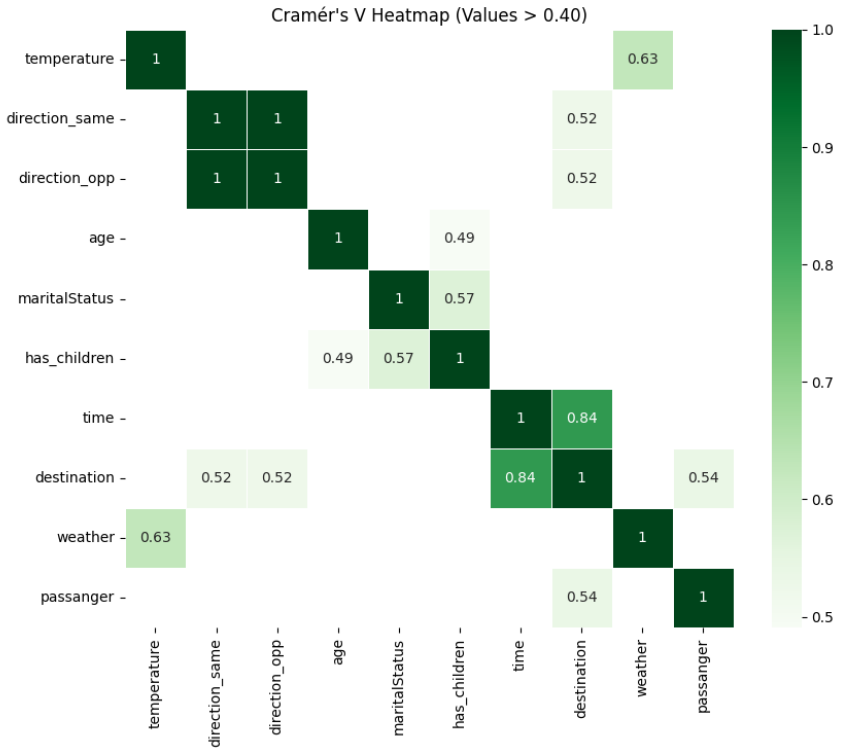

Figura 1: Bar plot of destination vs time

Some conclusions to be taken from the heatmap above are:

- direction_opp and direction_same are totally correlated because one is exactly the opposite of the other, so we removed the direction_opp feature.

- **time and destination**, **temperature and weather**, **maritalStatus and has_children**, destination and passenger and direction_same and destination are highly correlated features. The ones in bold were merged into one feature since these ones have the higher correlation values. The other pairs (the ones not in bold) have a feature that was already merged with another, which stopped us from merging them as well.

2.3 Bi-variate Analysis

Since we did not get any valuable information from an univariate analysis, we decided to perform it instead with two features, to understand more about our data.

Here we combined two features, taking into account the previous correlation analysis.

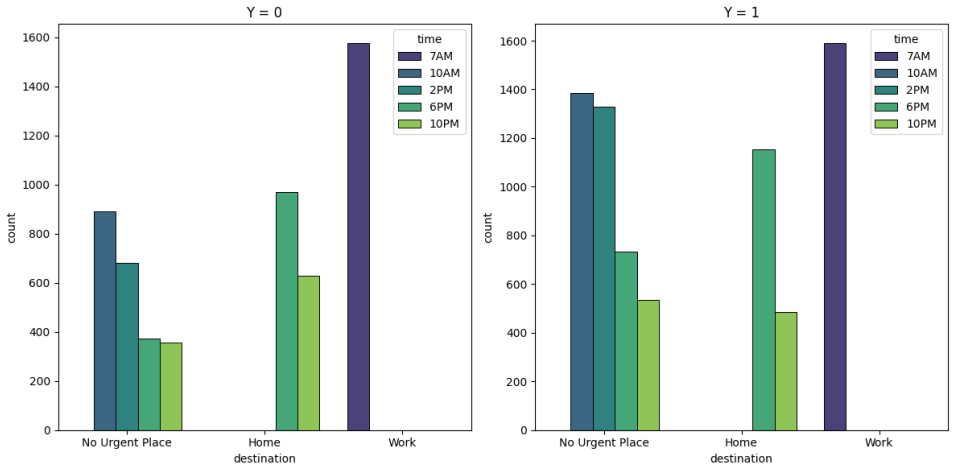

Figura 2: Bar plot of destination vs time

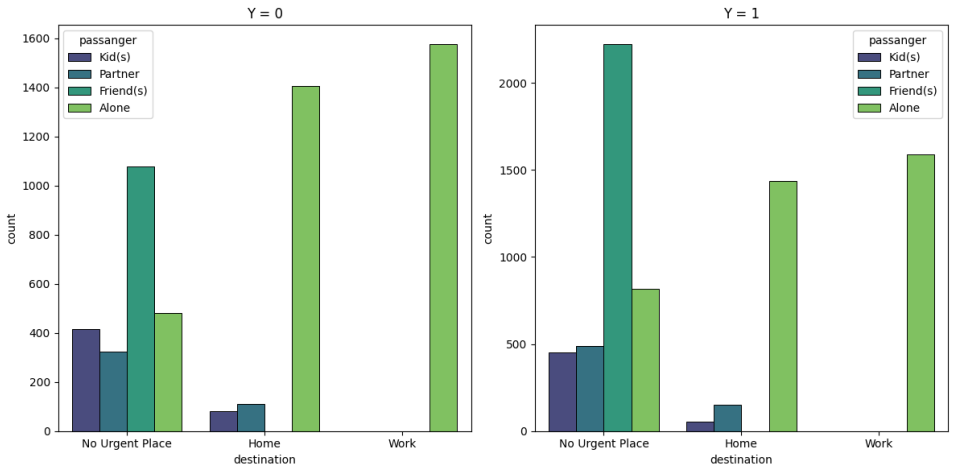

Figura 3: Bar plot of destination vs passanger

From the observation presented in figure 2 and figure 3 we can state that the users who go along with friends are likely to be going to a not urgent place; At 7 AM only workers are driving and they are alone; At 10 AM and 2 PM all users are just going to no urgent places.

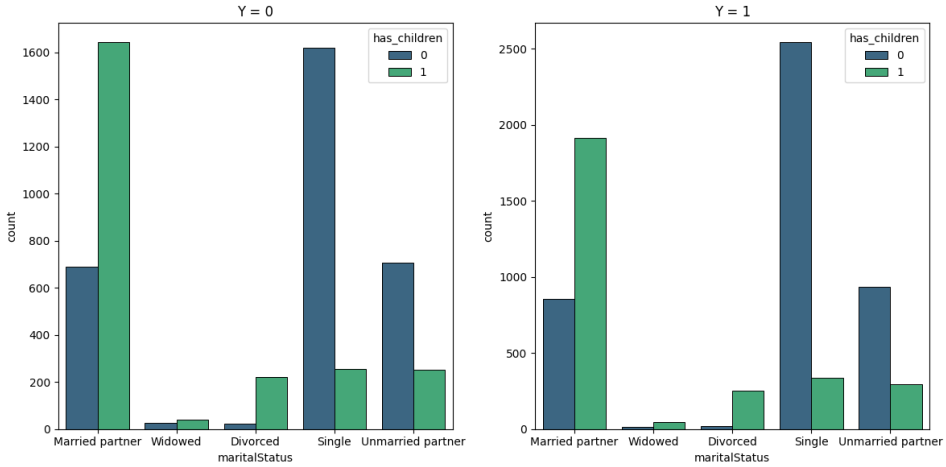

Figura 4: Bar plot of has_Children vs maritalStatus

From fig 4 we can see a significant correlation between "maritalStatus"and "has_children", the distribution trends are consistent between Y = 0 and Y = 1.

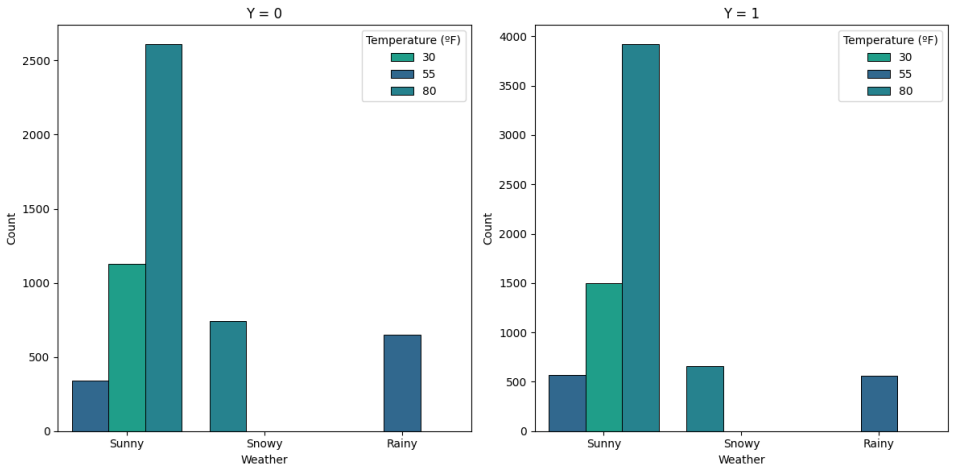

Figura 5: Bar plot of weather vs temperature

Aside from the high correlation between weather and temperature, one more time following the same distribution trends between Y = 0 and Y = 1, we can conclude that in rainy days we observe always 55ºF, and in Snowy Weather we observe always 30ºF.

2.4 Feature Aggregation

Feature aggregation is the process of combining multiple features to reduce dimensionality, capture complex behaviors, reduce noise, and potentially enhance model performance and interpretability.

There are a few features to tell the driving time to the restaurant/bar/coffee house, so it is possible to combine them into one.

The new feature ("toCoupon") will have three different categorical values "0", "1"and "2", meaning driving distance is less than or equal to 15 min, greater than 15 min and less than or equal to 25 min and greater than 25 min, respectively. The distribution of this feature is presented in figure 6

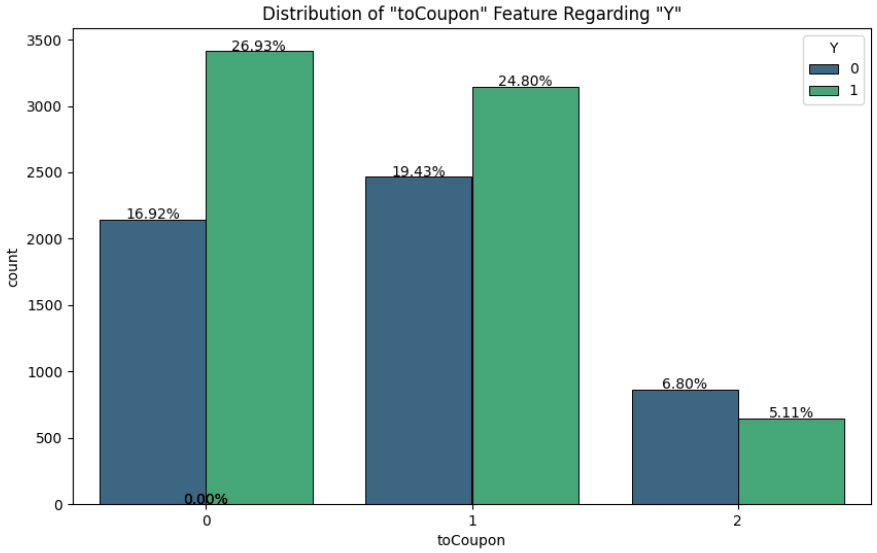

Figura 6: "toCoupon"distribution regarding "Y"

2.5 Feature Binning

Binning can help simplify the data, reducing noise or variance, and making the data more meaningful and easier to work with, especially when working with categorical variables that have many unique values.

The "occupation"feature has 25 variables of distinct values, which creates a lot of sparsity in the data matrix after encoding so we decided to convert it into a new feature using binning.

Firstly, we calculated the acceptance ratio for each occupation and then binned it into five categories, namely: low, medium_low, medium, medium_high, and high. To define the boundaries of these bins, we utilized quantiles, ensuring a data-driven categorization.

2.6 Feature Extraction

Feature extraction plays a vital role in transforming raw data into meaningful and efficient representations that can be better processed by machine learning models. In our context we find

three feature extraction possibilities, these conclusions came from the correlation and bivariate analysis

By doing so we are consolidating the essence of two features into one, therefore reducing the dimensionality of our dataset without losing important information.

3 Data Encoding

Data encoding is crucial for ensuring accurate storage, transmission, and retrieval of information, enabling seamless communication between systems, preserving data integrity, and safeguarding data.

For that reason, we tested different encoding techniques.

3.1 Ordinal Encoding

The main idea behind ordinal encoding is to preserve the inherent order of the data in the encoding.

It's important to note that ordinal encoding is suitable only for ordinal data. Using it for nominal data can be misleading. For our dataset, ordinal encoding will only be used for:

- age: The age categories have a natural order (from youngest to oldest).
- education: There is an inherent order in the level of education.
- income: The income brackets have a clear ranking from lower to higher income.
- Bar, Coffee House, Carry Away, Restaurant Less Than 20, Restaurant 20 To 50: Ordinal variables indicating frequency.
- Occupation: Is binned based on acceptance ratio.
- toCoupon

3.2 One-Hot Encoding

One-hot encoding is a method where each unique value of a categorical feature with N distant categories is represented by N binary column, with a "1"indicating the presence of the value and a "0"indicating its absence, is best suited for nominal variables where there's no inherent order to the categories.

For that reason will be used in:

- Passenger
- Coupon
- marital_hasChildren
- temperature_weather
- time_destination

3.3 Binary Encoding

To simplify we will use binary for features with a binary nature, which are:

- expiration: Binary in nature (2h = 0 or 1d = 1).
- Y: It's already binary
- direction_same: Already binary categorical variables.

3.4 Label Encoding

Label encoding is a technique for converting categorical values into numerical labels. For each unique category in a given feature, label encoding assigns an integer starting from 0 up to $N - 1$, where N is the number of distinct categories for the feature.

- Gender: Male = 0, Female = 1

4 Modeling

For modeling our dataset we use multiple machine-learning algorithms. Firstly, the data was split into a training + validation set and a test set, the split ratio is 75% for training + validation and 25% for testing.

Then in order to do cross-validation across every algorithm used a 5-fold cross-validation was set up using the KFold class, so the data could be shuffled before splitting to ensure randomness.

To have the best model for each algorithm a grid with different hyperparameters was defined and iterated in each model which allowed us to better tune our hyperparameters for the tested models.

We decided to try many different classifiers and choose the best one considering some metrics (the ones on the "x axis"of the graph below). Some assumptions we previously made in order to help us test the models more efficiently is that whatever model we choose, it should be suited to a binary output and should deal effectively with the high number of features we have. We also noticed that our data doesn't follow Gaussian distribution trends which led us to not include it in our tests. All the algorithms tested are presented below (7).

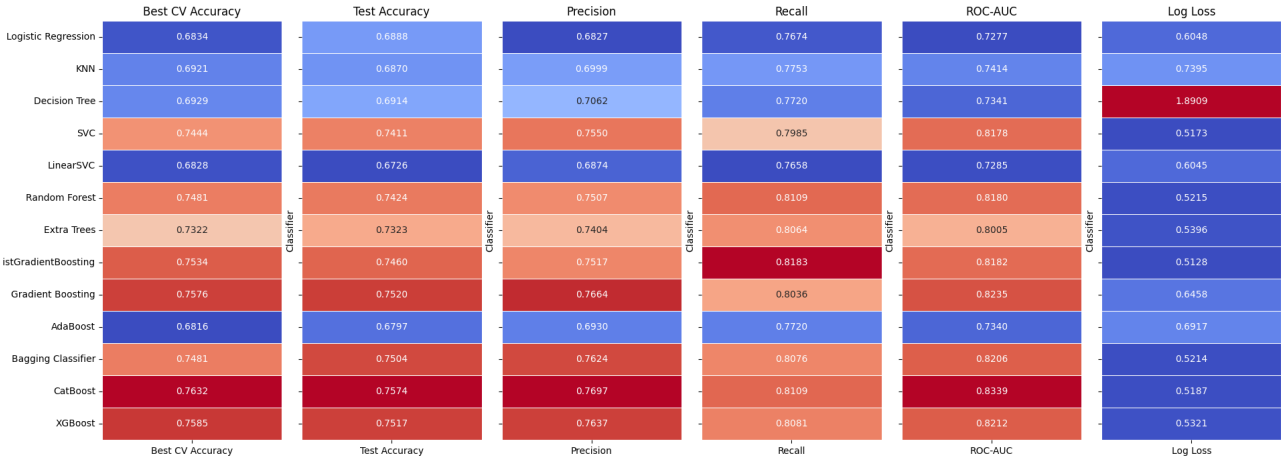

Figura 7: Model metrics values

From this information, there are a few interesting key points that are worth mentioning.

Test Accuracy is a measure of how well the classifier performs on unseen data, in this evaluation metric the XGBoost classifier has the highest Best CV Accuracy at 0.7585 and Test Accuracy at 0.7517. This model given its architecture behaves well with binary output which explains its excellent performance. Close contenders in this category are the Gradient Boosting and Random Forest classifiers.

Precision is a metric that tells us about the true positive rate, here the Gradient Boosting has the highest precision at 0.7664 which indicates that the classifier has fewer false positives.

On the other side, recall indicates that the classifier has fewer false negatives. The Gradient Boosting and the AdaBoost classifiers tie for the highest recall at 0.8183 and 0.8206, respectively.

ROC-AUC indicates the capability to distinguish between the positive and negative classes, where Gradient Boosting achieves the highest value at 0.8235.

The lower the Log Loss metric the better the classifier's performance is, the Gradient Boosting classifier has the second-lowest log loss at 0.5128, while XGBoost has the lowest log loss at 0.5321.

From this information, it is evident that the best classifiers are XGBoost, CatBoost, Gradient Boosting, AdaBoost, and Random Forest.

4.1 Stacking Classifier

In order to have a better model, the best classification models with their best parameters were combined to produce a meta-model, once again this classifier was trained using k-fold for cross-validation.

The final results are presented in the table below.

Metrics	Values
Test Accuracy	0.762
Test Precision	0.772
Test Recall	0.819
ROC-AUC	0.837
Log Loss	0.496

Figura 8: Stacking Classifier metrics values

5 Conclusion

5.1 Learning about the problem

Firstly it is important to address some things we learned from this problem and some limitations of our study. Apart from the correlations between variables that we stated previously, we tried to understand which features played a more preponderant role in coupon acceptance. For that we used an attribute of one of our models, Random Forest, that allowed us to measure the feature importance. For each tree in the Random Forest, the decrease in impurity is achieved by splitting each feature. This measures how much each feature contributes to improving the quality of a split in that tree. Using this method we plotted the graph below.

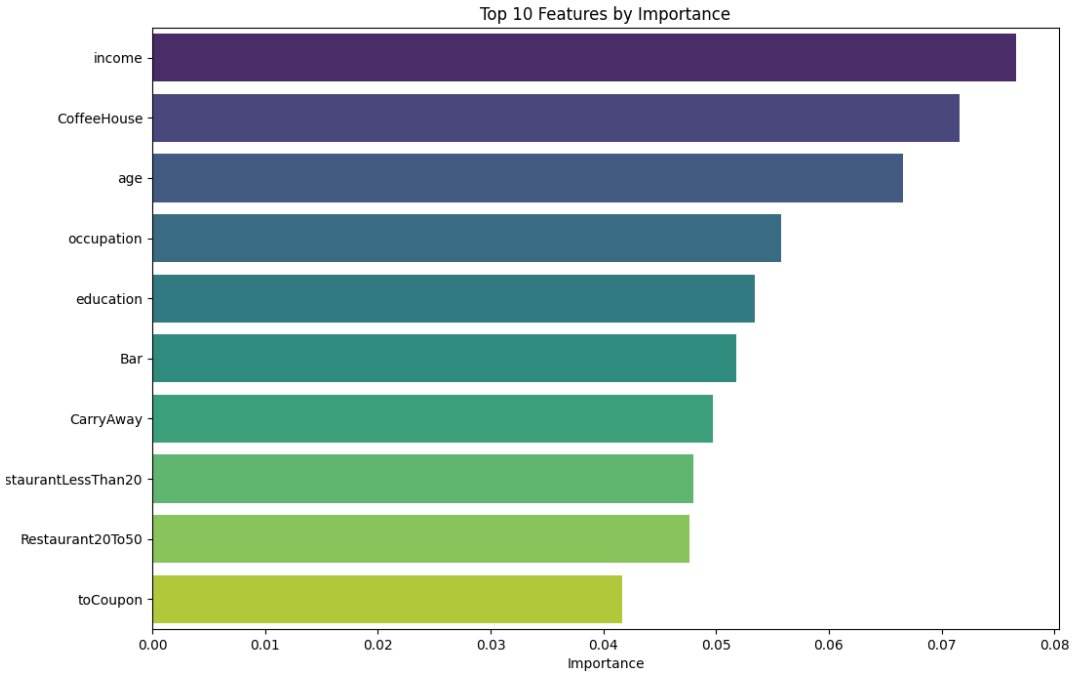

Figura 9: Feature importance

This allowed us to conclude that accepting the coupon is a matter of convenience. The closer an individual is to the coupon the more likely he is to accept it (see figure 6). We have noticed that income is the variable which more influences coupon acceptance. Curious about how much income affects coupon acceptance, we plotted a density graph to evaluate this relationship.

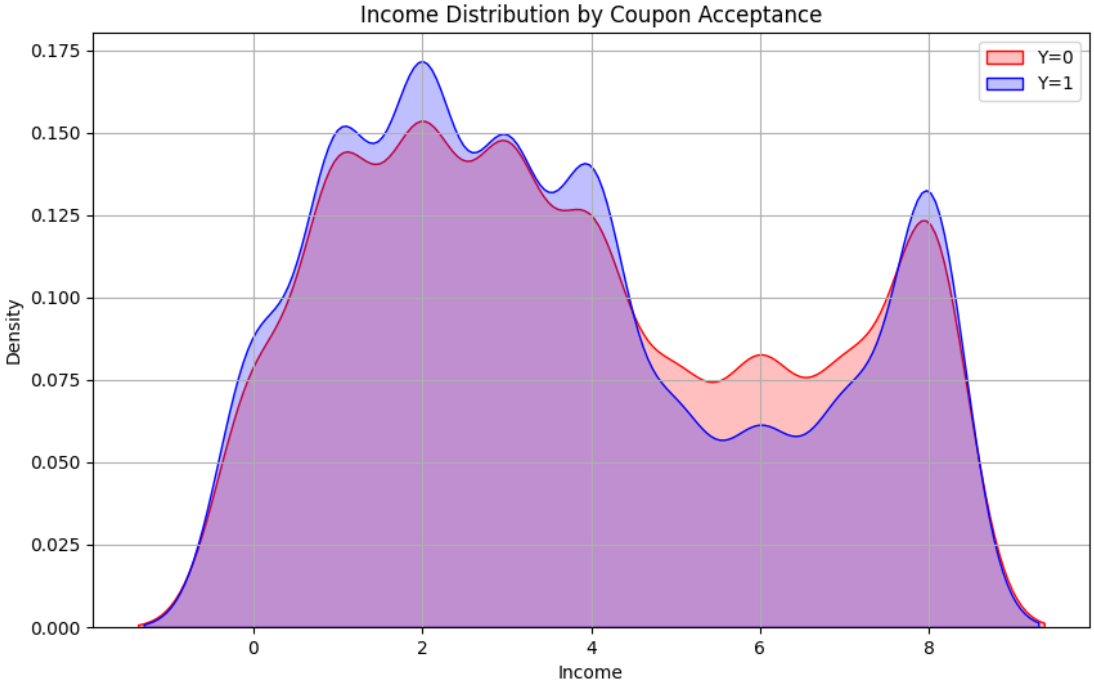

Figura 10: Density Function of Income for Y = 0 and Y = 1

Analyzing this plot, where the income increases from 0 to 8, helps us conclude that it is more likely that an individual with a lower income accepts a coupon than to refuse it. Also, people with higher incomes have a bigger tendency to not accept the coupons than to accept them, except for people with the highest incomes where the opposite happens, this small group (+100k of income) tends to accept more than refuse them.

5.2 Limitations and Suggestions

It's time to talk about some limitations and suggestions for future work.

It's worth mentioning that despite our efforts to reduce dimensionality through feature engineering, having to encode the features increased its number significantly. Allying this with the around 12 000 data points we have (which we consider are few for a classification problem) led us to reach an accuracy of about 76% which we consider very good given the data context but when considering a classification problem we expect higher values of accuracy.

We would like to suggest to the research team a few changes that in our opinion would improve the quality of their future studies. Firstly, there should be a previous analysis of the variables to be studied and how interesting or important they are for the relevance of the

subject. We noticed in our bivariate analysis that there were many correlated features and some irrelevant variables to the study case. For example, collecting data about temperature (with such low granularity, only 3 different categories) and weather is predictable to not be productive since the different weather states are very likely to be always related to the same temperatures, not adding any relevant information to the problem.

Some of the data was also irrelevant such as the car variable in which 99% of the data was missing which made us wonder why it was included in the dataset. Speaking of irrelevant data, many duplicates were found. Treating and fixing this data consumed some valuable time that could be used for more relevant topics of the study like modelling.

Finally, we would suggest including more data points given the dimensionality of the problem. After cleaning the data (before EDA), we were working with 23 features which we consider to be a lot since we have only 12 000 points, neglecting the modeling which is the scope of our job.

5.3 Overview

Our main metric for this problem was accuracy, since this is a classification problem, the accuracy metric provides us with the necessary information on the efficiency of this task. Even though tackling this problem wasn't easy, due to the high complexity of the data collected, we used the methods learned in class to treat data efficiently, having achieved a relatively high accuracy of 76,15%, through a stacking classifier. We are certain that if the amount of data was higher, we would have achieved higher accuracy, unfortunately, we only had 12610 instances of data, after removing duplicates, which is relatively low for a classification problem. Overall the project was a success, we treated data according to our judgment and it resulted in high accuracy in the end, proving that the methods used were adequate for the problem at hand.

Referências

[1] Finale Doshi-Velez Yimin Liu Erica Klampfl Perry MacNeille Tong Wang, Cynthia Rudin. *A Bayesian Framework for Learning Rule Sets for Interpretable Classification.* Journal of Machine Learning Research, 2017.