A Study on How Data Quality Influences Machine Learning Predictability and Interpretability for Tabular Data

by

Humra Ahsan

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master

of

Computing and Information Systems

YOUNGSTOWN STATE UNIVERSITY

May, 2022

A Study on How Data Quality Influences Machine Learning Predictability and Interpretability for Tabular Data

Humra Ahsan

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

Humra Ahsan, Student Date

Approvals:

Alina Lazar, Thesis Advisor Date

Dr. Feng Yu, Committee Member Date

Dr. Yong Zhang, Committee Member Date

Dr. Salvatore A. Sanders, Dean of Graduate Studies Date

ABSTRACT

Today data is the most important part of any organization, as data is everywhere around us. Most companies produce large amount of data that is essential for the decision making process. In this context, many machine learning and artificial intelligence methods can be used for analysis and prediction. To understand the data quality and make efficient use of the data, several pre-processing steps are necessary. In various fields of study and industry, machine learning is becoming the dominant problem-solving technique. Machine learning models are now being used to solve a variety of real-world problems in a variety of disciplines, ranging from retail and finance to medicine and healthcare which demands high predictive accuracy. Understanding data quality and feature engineering are some of the most critical parts of any machine learning project. Mostly, companies manage tabular data that needs to be converted into numerical data.

However, this improved predictive accuracy has often been achieved through increased model complexity which leads to a lack of transparency. The major disadvantage is that the models' inner workings are hidden from the user because it prevents even an experienced professional from interpreting and understanding the reasoning behind the system and how some decisions are made. The quality and quantity of data used to train machine learning algorithms are directly related to their predicted ability. Quality data leads to accurate predictions that in turn leads to accurate explanations. In many cases, it is important to know how predictions are made. The research is focused on the effect of data quality and feature engineering on training different tabular datasets using different machine learning models and the ranking of features in terms of their importance to the prediction.

The results are compared in terms of performance accuracy to find which feature set and which model works best.

Acknowledgements

I would first like to acknowledge and express my deepest gratitude to my thesis advisor Dr. Alina Lazar of the School of Computer Science, Information and Engineering Technology at Youngstown State University. This thesis would not have been possible without her constant support and guidance. Her expertise in this field made the project easier. Her level of patience, knowledge, and ingenuity is something I will always keep aspiring for.

I would also like to thank my committee members Dr. Feng Yu and Dr. Yong Zhang for their precious time and advice during my thesis process.

I would like to express my sincere gratitude to the Department of Computer Science, Information and Engineering Technology and the Graduate Studies at Youngstown State University for furnishing me with the financial support to complete this thesis.

I would like to offer my special thanks to my parents Mr. Azhar Ahsan and Mrs. Naheed Sultana for their constant love and support which kept me motivated and confident. My accomplishments and success is due to their unwavering support and belief in me. I also express my heartfelt thanks to my siblings for their perceptive suggestions and trust in my decision-making skills. Last but not the least, I owe my deepest gratitude towards my husband, Mr. Adil Ansari for his impeccable support, constant reinforcement, and valuable guidance. Thank you.

Table of Contents

List of Figures

List of Tables

1 Introduction

Machine learning (ML) is a sub-field of Artificial Intelligence (AI) that provides methods and algorithms for building accurate software applications used in decision making. Classification models predict outcomes using previous data as input to predict new output values.

Machine learning models require a huge volume of data for training. Explainability [1] and reliability of these models is directly proportional to the quality of data they are fed with. These models are deployed in high-stakes settings; relatively small errors in the training data can lead to large scale errors in a model's output. Low quality of data may lead to diverged model outcomes and thus adversely impact human trust in these systems. The quality of data plays a vital role in making accurate predictions as the machine learning process is totally dependent on data. Many companies and enterprises make use of Machine Learning algorithms to build models for analyzing huge and complex data, and provide faster and accurate results. They rely on the decisions made by these models in identifying risks and profitable opportunities. Data cleaning [2], preprocessing and qualitative analysis and feature engineering [3] plays a vital role in the decision making process.

Reliable Machine Learning models predict comprehensive decisions that are easier for a human to understand. Higher the interpretability of a model, easier it is for someone to understand why certain predictions or decisions have been made. Machine Learning algorithms are complex and until recently, have been notorious for being black boxes [4], there have been significant concerns about their functional transparency. The opaque nature has concerned developers of system behavior. It can be difficult to

understand why a model is predicting a particular response to sets of data input. There is a need to understand the internal processes of these models and explain the high-stakes decision making to regulatory authorities and stakeholders. The interpretability [5] of these Machine Learning models is becoming a key component in fostering trust and confidence in Machine Learning systems and increasing their adoption in industrial applications.

This study will focus on the impact of data quality on the predictability and interpretability of Machine Learning models. Relevance, accuracy and completeness are some of the dimensions to data quality used for training a model. In our work, we will use different techniques to assess and improve the input data quality, use them to train different Machine Learning models and evaluate its influence on predictability and accuracy of models.

2 Related Works

Data quality is a key metric to make accurate and informed decisions. It is a basic building block of a Machine Learning pipeline for improving the performance of a model. Most of the industries like healthcare, manufacturing, energy and utilities, etc. acquire data from automated AI systems. Usually, the quality of data generated by these systems are not befitting for analysing and building Machine Learning models.

For accurate decision making data needs to be cleaned and precisely analysed by the data scientists for building models. Data cleaning [2] is a time consuming process that takes most of the time for modelling process. There is no one such absolute framework built to address data quality issues

directly as the datasets vary in nature. It is difficult to clean data manually, researchers have proposed systems that automate data pre-processing steps and taking care of data quality.

Schelter et al. [6] presented a system to automates data quality verification in a scalable manner to meet the demands of current production use cases. During their work, they have addressed the most common dimensions of data quality: completeness, consistency and accuracy. They build a declarative API, that allowed specification of constraints on datasets and translation of these constraints to compute metrics. These metrics eventually allowed users to evaluate the constraints. They demonstrated their system on an on-demand video platform. The system performed check for completeness, consistency and predictability on this dataset. Completeness is calculated as a ratio of non-null values in a particular column. Consistency is measured by providing metrics on the data types, count of unique values, size of data set, range of values and a predicate matching metric. Uniqueness is referred to the unique values in a column. Distinctness corresponds to the ratio of unique rows in a column. Standard summarized statistics are implemented for numerical columns that include min, max, mean, standard deviation, histogram and entropy. Correlation and Mutual Information is also included to measure the amount of association between two columns.

Shrivastava et al. [7] addressed five serious issues that affect the experience of a data scientist in the process of performing data quality inspection. These issues are repetitive in nature, time consuming, require unorganised exploration, hard to reproduce, non-visual and non-interactive. This paper introduced Data Quality Advisor (DQA) to evaluate data qual-

ity in an automated, interactive and scalable manner. Data quality operations are performed by validators. A comprehensive structure for accessing quality of the data with the ability to automate, scale and generate human interpretable results is defined. It has its components divided into four stages. The DQA Core is the second stage of the framework that is responsible for performing all the checks and validations. Operations are abstracted out to perform checks in the framework by the validators. The Validators have three primary features: checker function, validity record and execution backend. The checker function performs the check operations. For example, the missing value detection validators's checker function is a method that checks the presence of missing values and detect the location of those instances in the dataset. The output of a checker function is the validity record that contains vital information of the check in unencoded format required to populate the GUI and in the encoded format for the next operation to perform. In this paper, a library of validators has been built for the Data Quality Advisor toolkit to avoid the various data quality issues in the data pipeline. These validators are categorised as: general validators, AI validators, Time and Series validators, transformers and others. General validators are then categorized on the basis of checks performed like: value checks, uniqueness, duplicate value, statistical and correlations.

Metaweb Technologies, Inc. [8] created a software called OpenRefine, which is an open-source project supported by the community. It is originally written and conceived by David Huynh. It is a powerful tool to operate on inconsistent data: cleaning and transforming it from one form into other; and extending it with web services and external data. The tool

is capable of importing data in different formats, explore datasets in a few seconds, cell transformations, deal with multiple value cells, create links between datasets, partition and filter data easily with regex, use named-entity extraction to automatically identify topics in full-text fields and perform advanced operations on data using the general refine expression language.

Another common problem [9] is that real-world tabular datasets often contain categorical columns with string entries. However, training ML models on such data generally requires a numerical representation of all entries. Considering string entries as unordered—categories gives well-framed statistical analysis. In such situations, categories are assumed to be mutually exclusive and unrelated, with a fixed known set of possible values and they get encoded using vector representation of the entries.

3 Data Cleaning and Pre-processing Methods

In this thesis, we explore the effect of data cleaning and feature engineering on the results of training Machine Learning models on different tabular datasets.

3.1 Data Import and Data Wrangling

Data import is the process of uploading data from external sources and combining it with the collected data by analytics. Data Wrangling [10] is the process of transforming and mapping complex datasets consists of raw data into a desired format for easy access and analysis. It is also called as data cleaning, data remediation or data munging. It can be manual or an automated process. It has a variety of processes. Every project has a

unique method depending on the dataset used and the desired goal.

3.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) [11] is the process of investigating datasets used by data scientists to recognize different patterns and relations, spot anomalies, test hypothesis and validate assumptions through summary of their main characteristics usually by statistics and graphical representations and other data visualization methods. It is an important step in analyzing the data of any data science project. EDA provide insights from the data.

3.3 Data Imputation

Imputation [12] is a process of filling the missing data in the dataset columns with some substitute value to keep most of the data/information of the dataset. This technique is used because removing the data from the dataset every time is impractical and can reduce the size of the dataset to a large extent, which not only raises concerns for biasing the dataset but also results in incorrect analysis.

One way to deal with missing values is to replaces them with a specified placeholder. In this case, we replaces the missing values with the most frequent item in the specific columns. This approach works well when the columns with missing values are categorical. Another simple way to deal with missing values is to replace them with the median value. This works well when the dataset has numerical columns.

3.4 Categorical Encoding

In many ML tasks, the data sets comprises of text or categorical values (non-numerical). Few algorithms can handle categorical values very well but most of the algorithms expect numerical values. There are several approaches [9] to convert categorical values to numerical values, each with its own trade-off and impacts on the feature set. In this thesis, one-hot-encoder [13] and label encoder is used for the conversion task. Label encoding is very simple and is used when the categorical feature is ordinal type. In this technique each label is converted into an integer value.

The numeric values can be misinterpreted by algorithms in label encoding as they have some sorting order in them. This ordering problem of label encoding is eliminated in one-hot-encoding approach but does have a drawback of adding more columns to data set. This technique is used when the features are nominal. Every category value is first converted to a new column and then mapped with a binary variable either 0 or 1 (0 indicates the absence and 1 indicates the presence of that category).

4 Machine Learning Methods

Machine Learning uses statistical concepts to enable computers to "learn" without explicit programming. A Machine Learning algorithm produces a model, which is a mathematical expression that represents data in the context of a problem. A logistic approach fits best when the task that the Machine is Learning is based on two values, or a binary classification. Few of the supervised Machine Learning methods discussed below are used to train the models in this thesis.

4.1 Logistic Regression

Logistic regression [14] is a supervised Machine Learning algorithm used for classification problems which gives discrete output. It is faster than other supervised classification techniques like support vector machines (SVM) [15, 16] or ensemble methods [17] but has a lower accuracy. Logistic regression is a statistical method that predicts the output for a dependent binary variable from one or more independent variables. This statistical analysis is used to find the befitting model to understand the relationship between the dependent and the independent variable(s) by estimating probabilities from generating the coefficients of logistic regression equation. It is also used to predict the probability of an event to occur or a choice being made.

Logistic regression is a classification model rather than regression model. It is a simple and efficient supervised Machine Learning algorithm used for binary and linear classification tasks.

4.2 Gradient Boosting Methods

Gradient boosting technique [18] is a type of Machine Learning method which is well known for its accuracy and prediction speed for huge and complex data. Among all, Xgboost [19] is one of the most popular gradient boosting implementation, but other implementation improve upon it (Catboost [20] and LightGBM [21]). The overall prediction error can be minimized by combining the next best possible model and the previous models with gradient boosting technique. The target values for each case is set according to the error gradient with respect to the prediction to minimize the prediction error. The target values depend on how much of a change in case predictions impacts the overall prediction error, hence,

named "gradient boosting".

4.3 MLjar Supervised

Mljar-supervised [22] is an Automated Machine Learning Python package which is used to explaining and understanding the tabular data. MLjar saves the time for data scientist. It has four working modes: explain, perform, compete and optuna. It pre-processes the data like missing value imputation and converting categorical features, constructs many Machine Learning models and finds the best model by performing hyper-parameters tuning, creates the detailed markdown report for each model. It has a vast set of algorithms: Baseline, Linear, Random Forest, Extra Trees, LightGBM, Xgboost, CatBoost, Neural Networks and more. It can handle binary classification, multi-class classification, and regression problems. MLjar can perform advanced feature engineering, like: golden features, feature selection, text and time transformation. MLjar-supervise uses the most frequent class as the Baseline algorithm. Decision trees can be easily visualized with the dtreeviz for better understanding of the data. Feature importance is generated based on permutation and SHAP explanations [23] are computed for each algorithm.

4.4 Feature Importance

Feature importance [24] is the techniques that set a score to all the input features for a given model based on their usefulness in predicting a target variable. Higher the score value, larger will be the effect of a specific feature on the model that is being used to predict a certain variable. Basically, feature importance rank "importance" of each feature on the basis of their

effect on model's prediction. Feature importance is extremely useful for (i) understanding the relationship between features and the target value (ii) improving the performance of a model (iii) interpreting and communicating the model to stakeholders.

5 Tabular Datasets Description

5.1 In-Vehicle Coupon Recommendation

The dataset [25] is obtained from a survey conducted on Amazon Mechanical Turk. Different driving scenarios are given to the person taking the survey including the time, weather, destination, passenger, etc., and then asked whether a person will accept coupon if he will be driving in the given scenario. This is a multivariate dataset consisting of 26 attributes.

5.2 Heart Disease

The "target" field in this dataset determines the presence of heart disease in a patient. This dataset [26] is derived from four databases : Switzerland, Cleveland, Hungary and the VA Long Beach. It comprises of 76 attributes out of which only a subset of 14 attributes have been used in our Machine Learning experiments to determine the absence or presence of heart disease. The only database used by Machine Learning researchers is Cleveland database which has concentrated mainly on distinguishing presence from absence. The presence is determined on a scale of 0 to 4 (0 being no presence).

6 Experiments and Results

All the experiments were performed using the Jupyter Notebooks [27] platform provided by the Ohio Supercomputing Center [28]. We used a Python conda environment [29] to install all the Python packages necessary. Machine Learning algorithms were implemented using the scikit-learn Python package [30].

6.1 Scikit-learn with In-Vehicle Coupon Recommendation Dataset

The brief description of the columns in the dataset are given in Table 1.

6.1.1 Data Wrangling

There are **missing values** that require imputation in the dataset. The "car" column has 108 non-null values which means more than 99 percent values are null or missing. This column has no significant importance as it does not contain useful information. so we decided to drop it off. There are 5 other columns with missing values that have "object" data type which means they all are strings. After performing visualization of the missing values using the matrix in Figure 1, we have found that the patterns of missingness in the dataset is random. To understand it further, the correlation between every two columns is checked by heatmap in Figure 2. There is no dependence between the occurrence of missing values of two variables. So we can do data imputation on them. Though data imputation can only be done after splitting the data as it cannot be performed on the whole dataset directly. The basic data wrangling part is finished here.

Name	Description
destination	destination of the coupon user
passenger	passenger in the scenario
weather	weather during the scenario
time	time of the day
coupon	coupon type
expiration	duration of coupon expiration
gender	sex of the user
age	age of the user
marital status	user's marital status
has_children	number of children a user has
education	academic qualification of the user
occupation	occupation of the user
income	income range of the user
car	type of car a user has
bar	count of user's bar visits in a month
coffee house	count of user's coffee house visits in a month
carry away	how many times the user gets take-away food in a month
restaurantLessThan20	number of times user visits a restaurant with an average expense per person of less than $20 every month
restaurant20To50	number of times user visits a restaurant with average expense per person of $20 - $50 every month
toCoupon_GEQ5min	user has to drive more than 5 minutes to the restaurant/bar to use the coupon
toCoupon_GEQ15min	user has to drive more than 15 minutes to the restaurant/bar to use the coupon
toCoupon_GEQ25min	user has to drive more than 25 minutes to the restaurant/bar to use the coupon
direction_same	whether the restaurantbar is in the same direction as your current destination
direction_opp	whether the restaurantbar is in the same direction as your current destination
target	probability of coupon acceptance

Table 1: Column Description of In-Vehicle Coupon Recommendation Dataset

SimpleImputer, scikit-learn's imputation implementation was used to perform imputation. All missing categorical values were replaces by the "most_frequent" value. For numerical data, the "median" was used for replacing missing values along each column.

To perform categorical encoding we split our data into two sets:

Simple Encoding (Plan A): For the nominal features that we believed were strong predictors, we performed both frequency encoding and target encoding.

One-Hot Encoding (Plan B): For all other categorical features, we performed one-hot encoding.

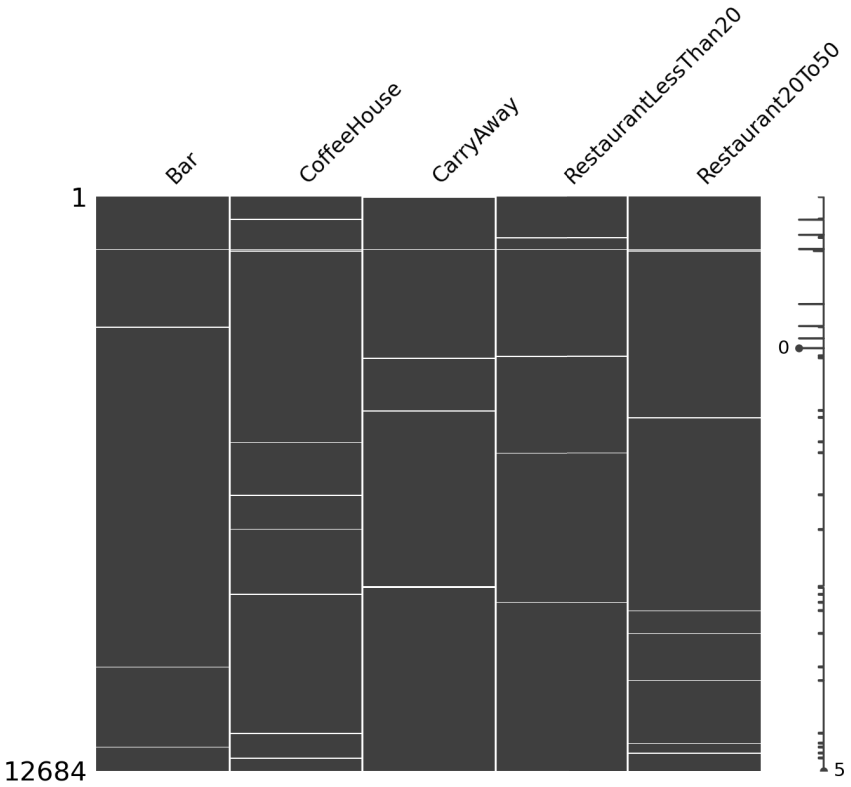

Figure 1: Matrix Plot

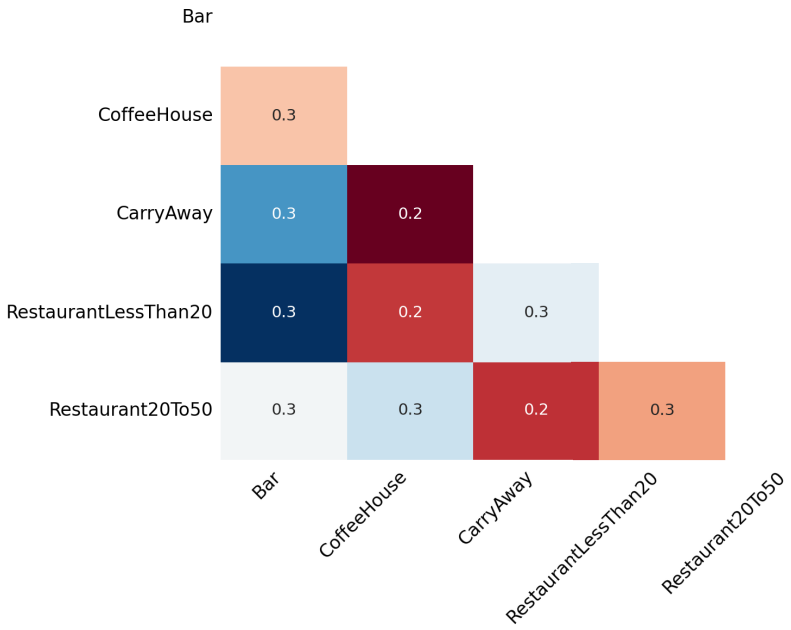

Figure 2: Heat Map

To analyze the dataset, summary of statistics pertaining to the dataframe columns are computed and showed in Table 2. The column **"GEQ5min"** has only one single value: 1. Showing no variance at all, so, we can drop it. According to the dataset description, this column means driving distance to the restaurant/bar for using the coupon is greater than 5 minutes, so all the restaurant/bars are at least five minutes away from the driver.

6.1.2 Exploratory Data Analysis

The **count plot** is used to represent the occurrence(counts) of the observation present in the categorical variable using the seaborn library. It uses the concept of a bar chart for the visual depiction.

Categorical Features

The counting plots of the categorical features shows that the dataset has

	temp	has any children	GEQ 5min	GEQ 15min	GEQ 25min	dir same	dir opp	Y
mean	63.30	0.41	1.0	0.56	0.11	0.21	0.78	0.56
std	19.15	0.49	0.0	0.49	0.32	0.41	0.41	0.49
min	30.0	0.0	1.0	0.0	0.0	0.0	0.00	0.0
25%	55.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
50%	80.0	0.0	1.0	1.0	0.0	0.0	1.0	1.0
75%	80.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0
max	80.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 2: Descriptive Statistics of In-Vehicle Coupon Recommendation Dataset

two kinds of categorical data: ordinal and nominal. The one hot encoder is applied to nominal features however, ordinal data should be mapped into numerical data to preserve the inner order.

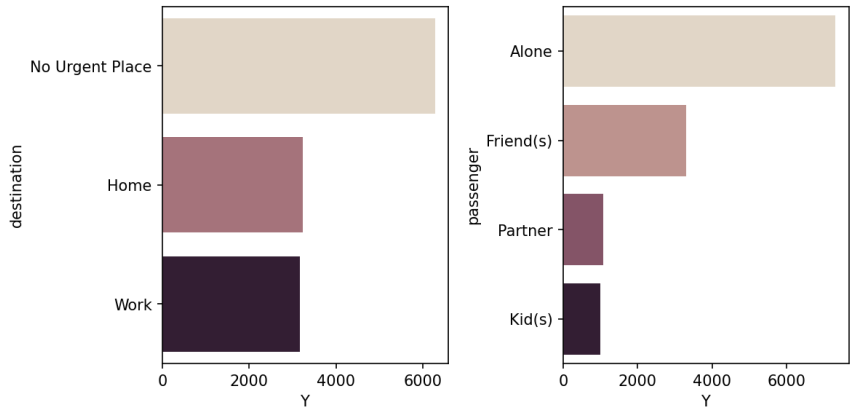

Figure 3: Count Plot of Destination and Passenger with Target

destination: Most of the drivers are driving to a non urgent place and there are almost equal number of drivers driving to the "Home" and "Work", as shown in Figure 3.

passanger: Almost 90% of the passangers are travelling alone, as shown in Figure 3.

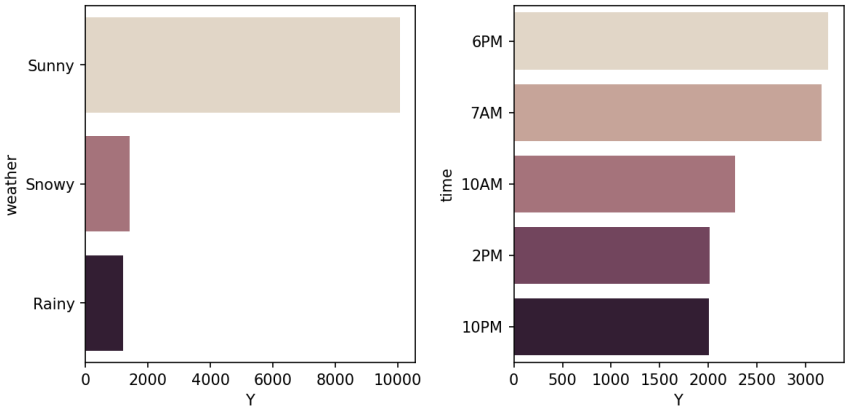

Figure 4: Count Plot of Weather and Time with Target

weather: The highest number of observations are obtained when the weather was sunny, as showed in Figure 4.

time: Most of the drivers are driving during the evening between 6PM to 7PM. There are significant number of drivers driving during the other time of the day, as shown in Figure 4.

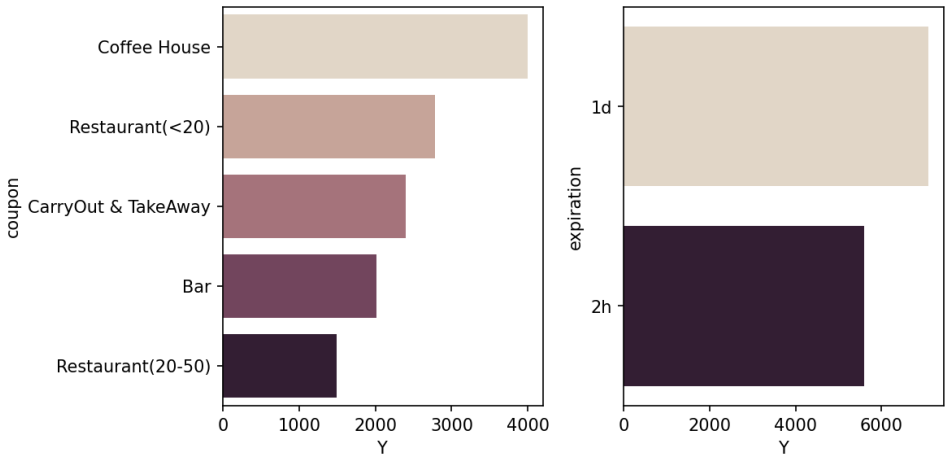

Figure 5: Count Plot of Coupon and Expiration with Target

coupon: Most of the coupons are collected by the drivers from the Coffee House, as showed in Figure 5.

expiration: The coupons that expire in one day has larger occurances

than the coupons expiring in two hours, as shown in Figure 5.

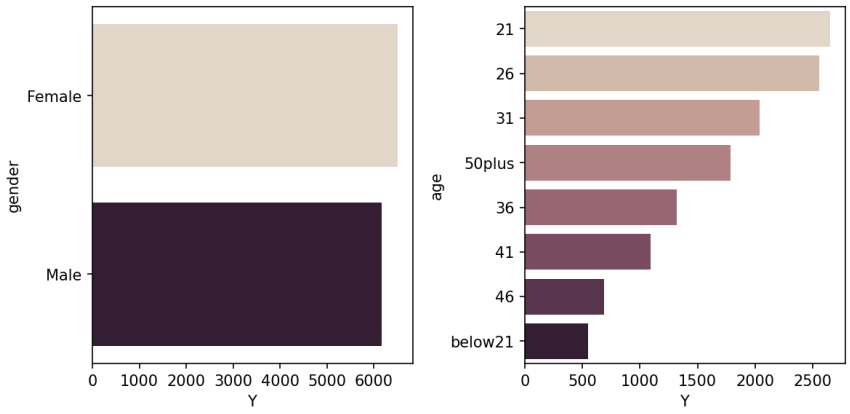

Figure 6: Count Plot of Gender and Age with Target

gender: This column has almost equal distribution of instances for both male and female drivers, as shown in Figure 6.

age: The drivers of the age of 21 and 26 shows the highest occurrence, as shown in Figure 6.

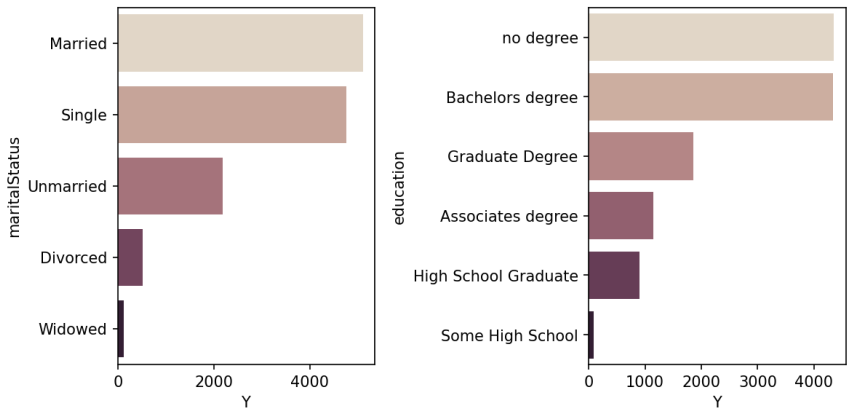

Figure 7: Count Plot of MaritalStatus and Education with Target

maritalStatus: The drivers that are married and single has larger occurances while widowed drivers has very few occurrences, as shown in Figure 7.

education: The drivers with Some college - no degree and Bachelors degree has equal number of occurrences, as showed in Figure 7.

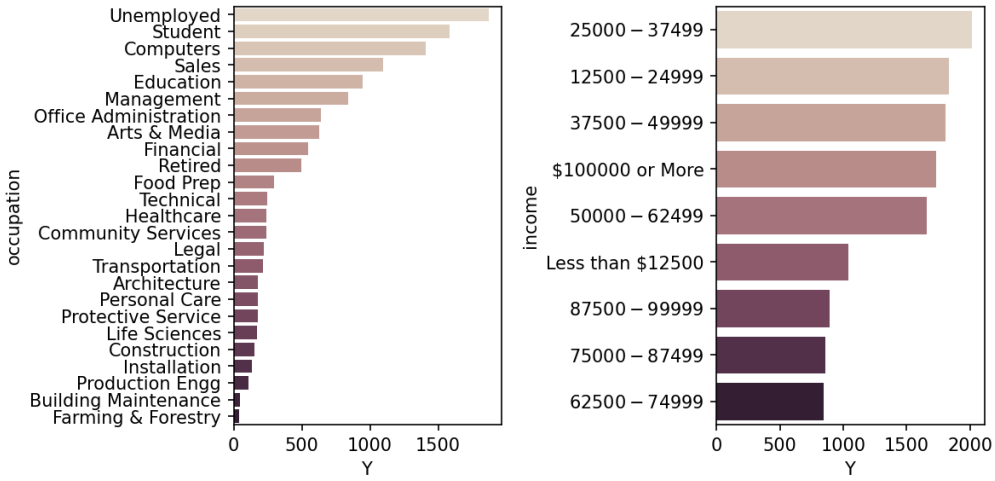

Figure 8: Count Plot of Occupation and Income with Target

occupation: The "occupation" column has many levels. This column shows the occupation of the driver, as shown in Figure 8. If we apply one-hot encoder on it, it will greatly increase the sparsity of the data.

income: The "income" column has 9 different income ranges: $37500 - $49999, $62500 - $74999, $12500 - $24999, $75000 - $87499, $50000 - $62499, $25000 - $37499, $100000 or More, $87500 - $99999 and Less than $12500, as shown in Figure 8.

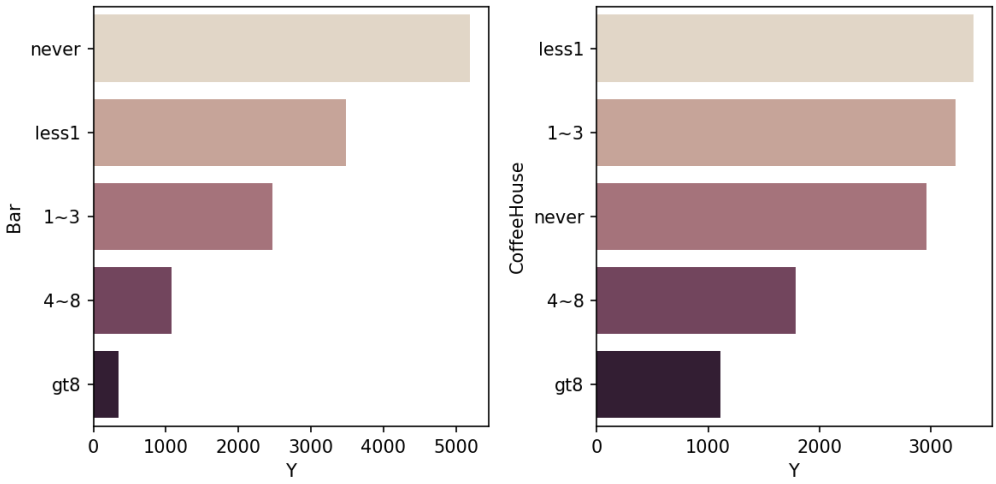

Figure 9: Count Plot of Bar and Coffee House with Target

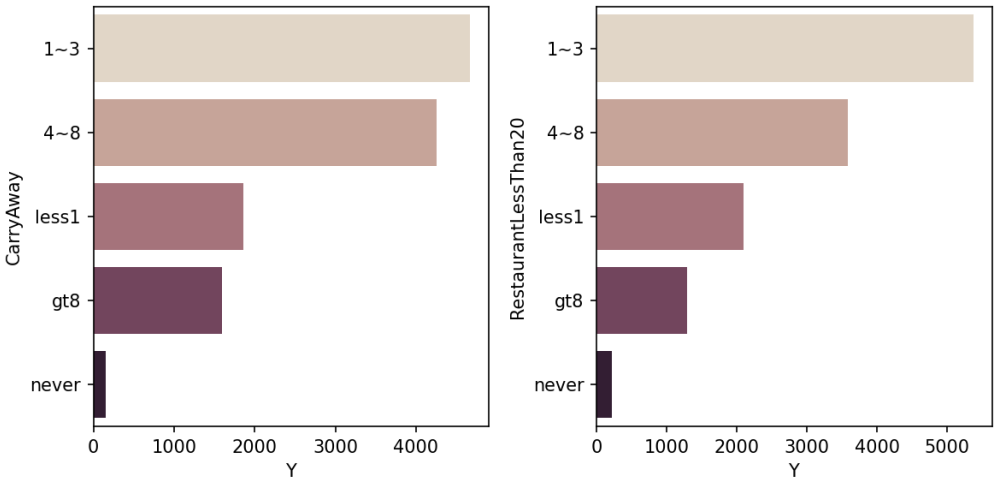

Figure 10: Count Plot of Carry Away and RestaurantLessThan20 with Target

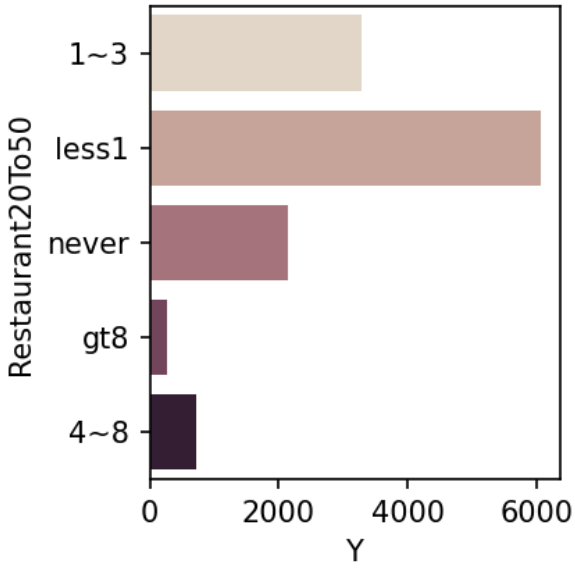

Figure 11: Count Plot Restaurant20t050 with Target

Bar, CoffeeHouse, CarryAway, RestaurantLessThan20 and Restaurant20to50: The columns "Bar", "CoffeeHouse", "CarryAway", "RestaurantLessThan20" and "Restaurant20to50" shows 5 different categories: never, less1, 1 3, 4 8, gt8. These column have the occurrence of how many times a passenger visits to a bar, a coffee house, a restaurant where per person average expense is $20 - $50, or less than $20 and gets take-away food every month, as shown in Figure 9 and Figure 10.

Nominal Features

destination: People that has no urgent place to go has a higher probability to accept the coupon, as shown in Figure 12.

passenger: If the passengers in car are friends of the driver, they are more likely to accept the coupon, as shown in Figure 12.

weather: People tend to accept the coupon when it is sunny, as shown in Figure 12.

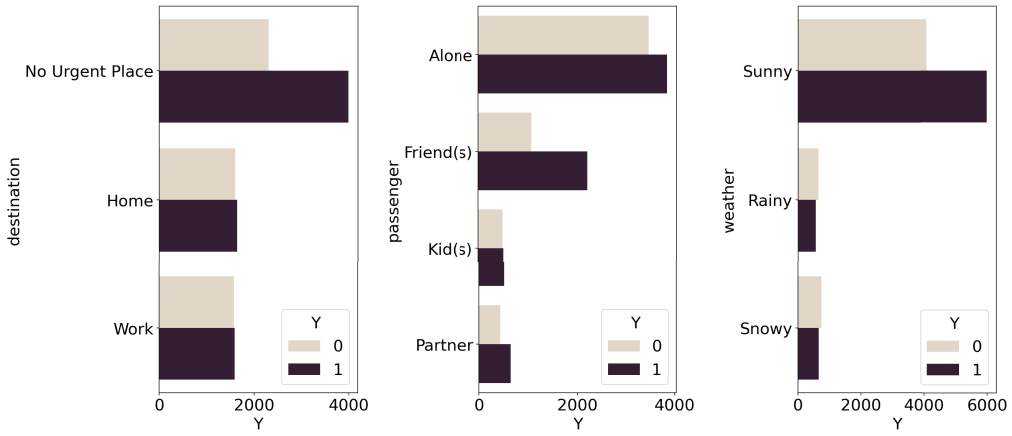

Figure 12: Relationship Between destination, passenger and weather with target

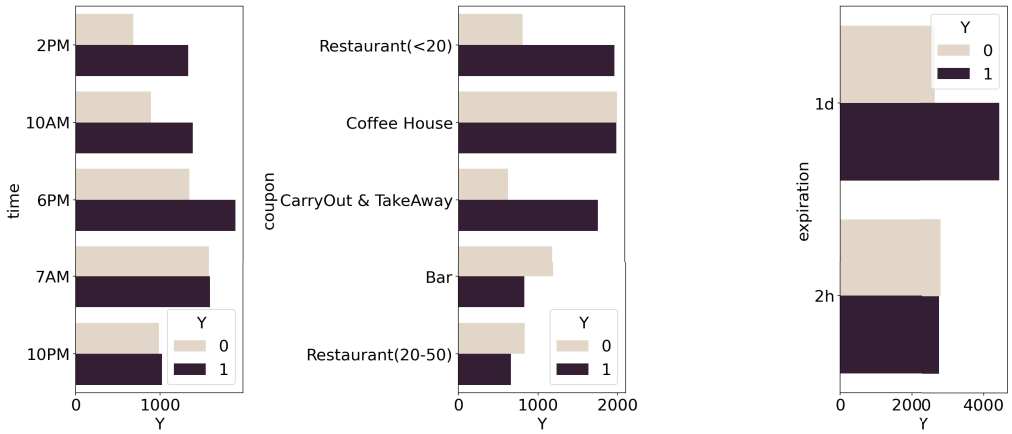

Figure 13: Relationship between time, coupon and expiration with target

time: If the time is too early or too late, the probability of accepting the coupon is lower, as shown in Figure 13.

coupon: If the coupon is of a coffee house, the probability of accepting the coupon is just the same as rejecting it. If the coupon is of a cheap restaurant or carry out, most people will accept the coupon. If the coupon is of a Bar of expensive Restaurant, people tend to refuse it, as shown in Figure 13.

expiration: People are more likely to accept a coupon that expires in one

day than one in two hours, as shown in Figure 13.

Figure 14: Relationship Between gender, maritalstatus and education with target

gender: There is no much difference between gender, as shown in Figure 14.

maritalStatus: Single people are most likely to accept the coupon, as shown in Figure 14.

education: Some college, Bachelor or high school graduate are more likely to accept the coupon, as shown in Figure 14.

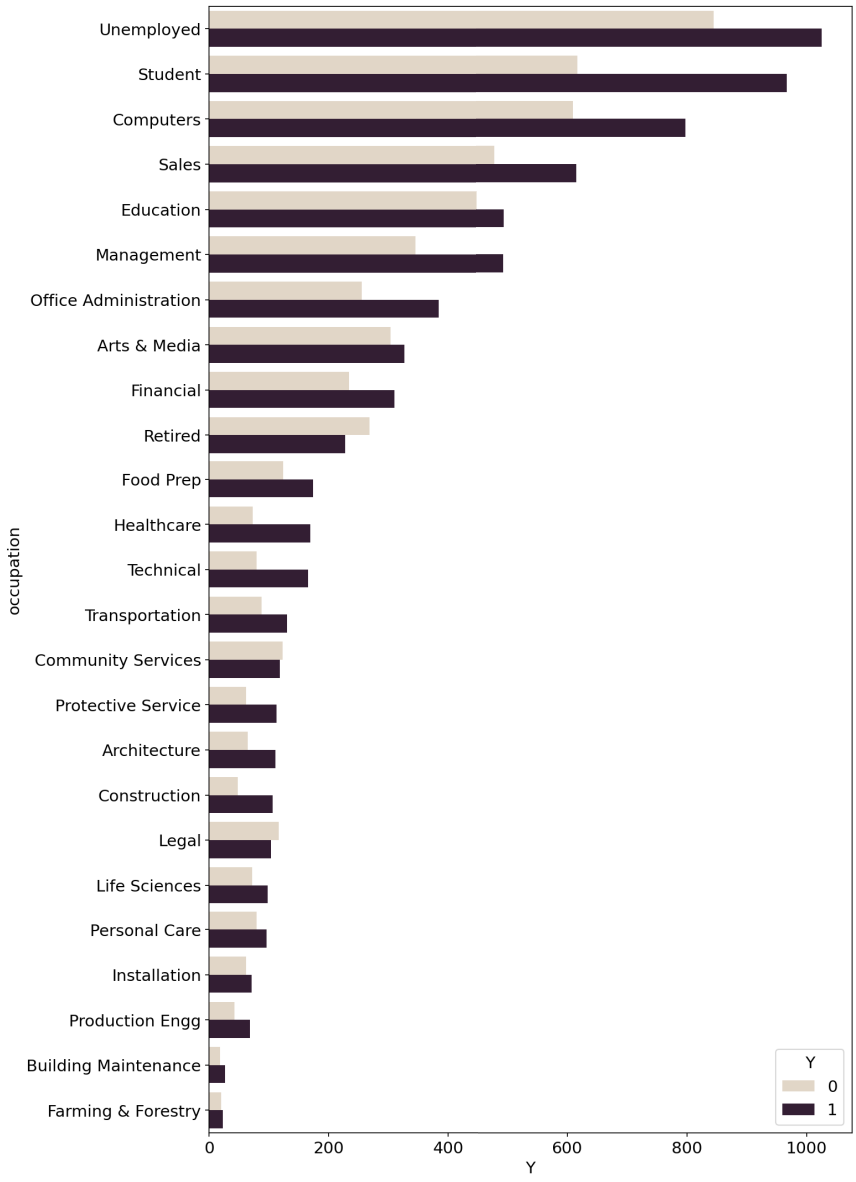

Figure 15: Relationship between occupation and target

Figure 15 shows some differences among jobs.

From these count plots, we develop an insight that all nominal features are strong predictors. Thus, to improve their predictive power we will perform frequency and target encoding on them.

Numerical Features

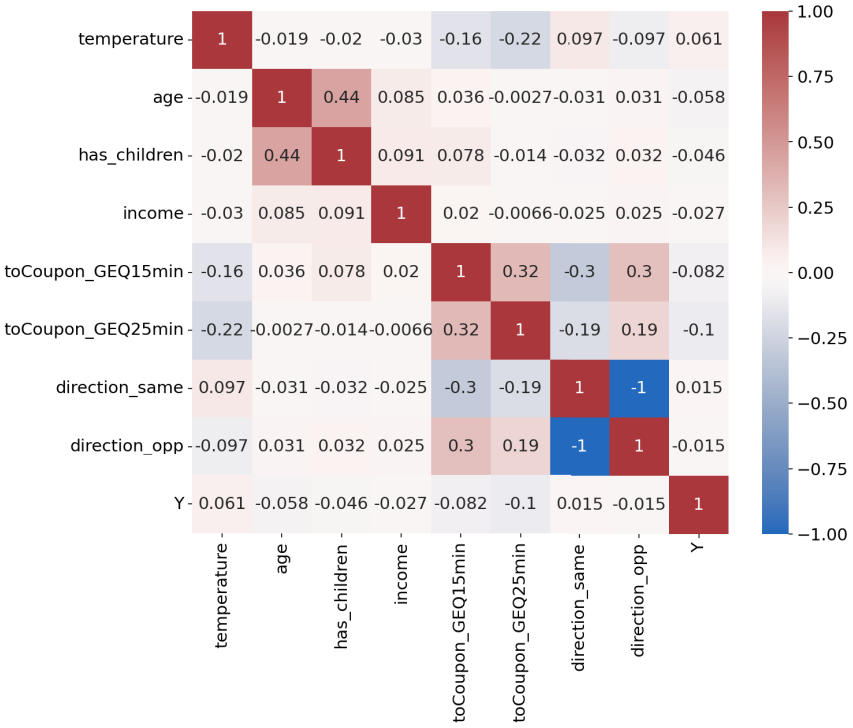

Figure 16: Correlation of numerical features

The heatmap in Figure 16 shows, there are two columns sharing the same information: "direction_same" and "direction_opp". They indicate whether the restaurant/bar is in the same direction as your current destination. So we dropped the "direction_opp" column.

Besides, there are correlations among those frequency columns: Bar, CoffeeHouse, CarryAway, RestaurantLessThan20, Restaurant20To50.

The correlations between categorical features and the target shows the strong predictors (correlation $>= 0.1$): destination, passanger, weather, time, coupon, and expiration.

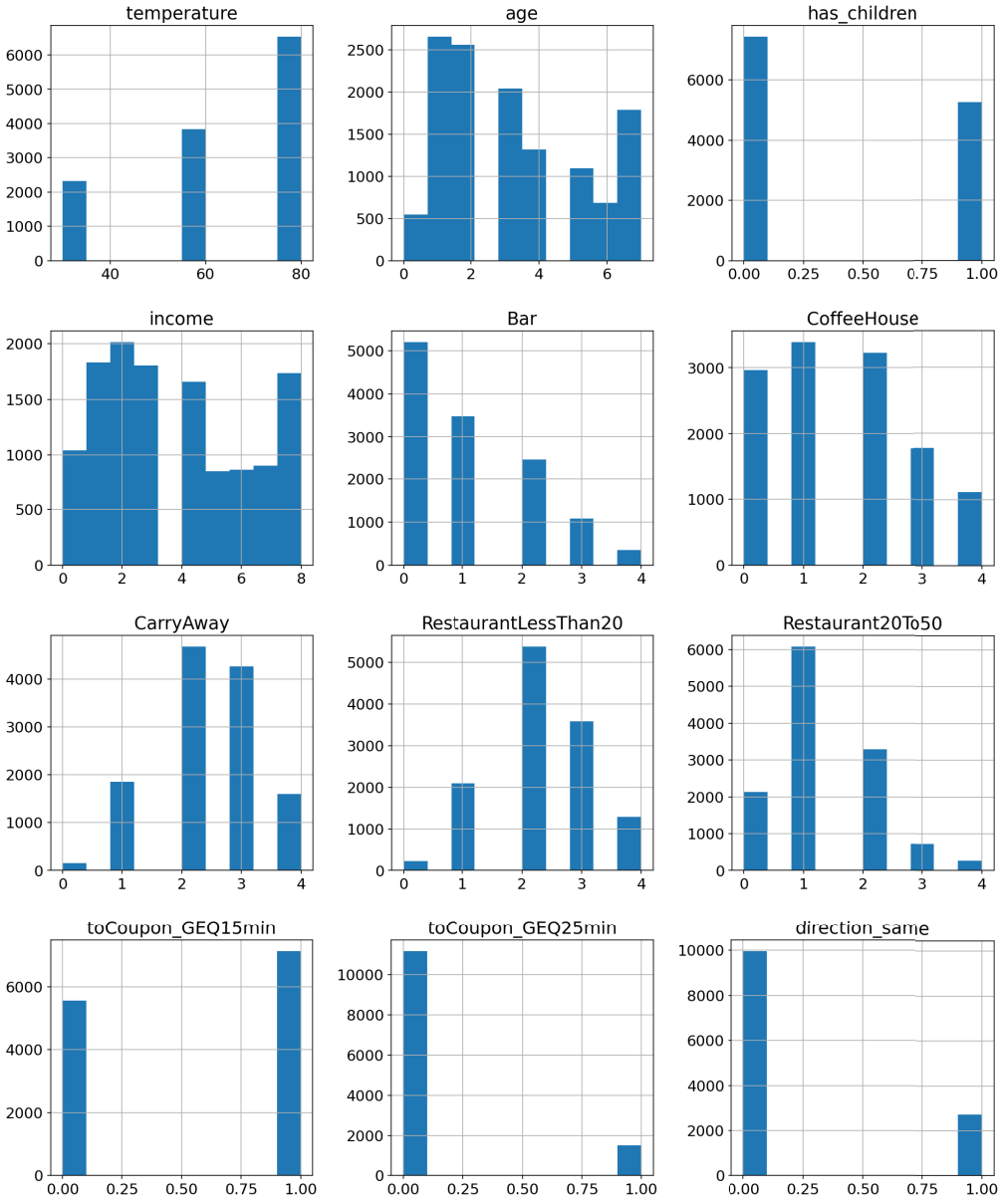

Figure 17: Histogram

After dropping the "toCoupon_GEDQ5min" feature, there are still two features about the driving distance to the coupon's location: "toCoupon_GEQ15min" and "toCoupon_GEQ25min". They have two possible values either 1 indicating yes or 0 indicating no to the question: is the travel time to the restaurant/bar to use the coupon is greater than 15 minutes/25 minutes? We should be able to combine these two columns into one with ordinal data inside: greater than 5 minutes and less than 15 minutes, greater than 15 minutes and less than 25 minutes, greater than 25 minutes. We believe doing that should be better than treat those two columns using one-hot encoder just like different categories.

Now, we explore the relationships between some of the features and the target.

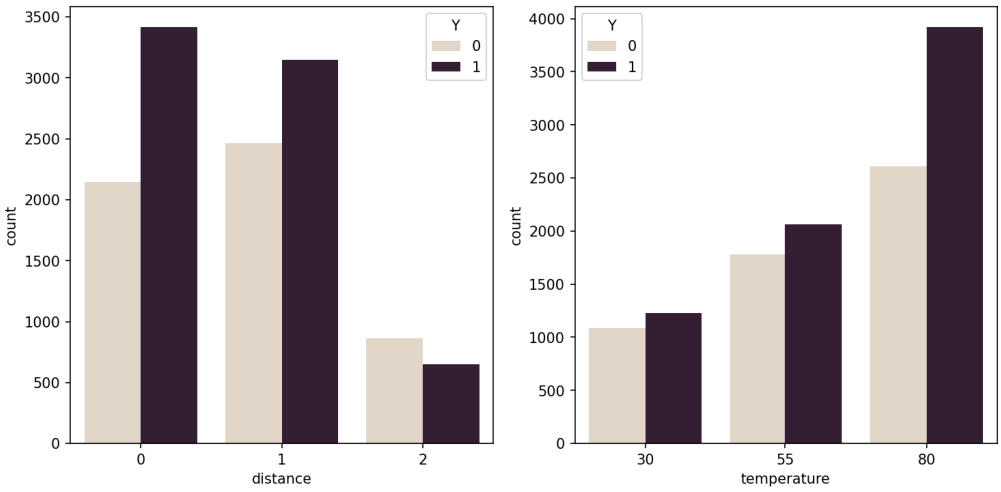

Figure 18: Relationship between distance and temperature

The plot in Figure 18 shows some subtle, positive correlations between short distance, high temperature and the probability to accept the coupon in the scenario.

Data Pre-processing

From the exploratory data analysis, we have almost all categorical variables. Feature engineering can be performed on splitted data in two sections: data imputation and categorical encoding. Data imputation is done by replacing missing value with most frequent item. For categorical encoding, we will do frequency encoding and target encoding on the categorical features that we believe are strong predictors and One-Hot encoding will be applied to all other categorical features.

To compare the effect of different feature engineering we made two plans:

Simple Encoding (Plan A) is to do frequency and target encoding for strong predictors we observed in EDA part, and one-hot encoding for other categorical features.

One-hot Encoding (Plan B) is to apply one-hot encoding for all the categorical features.

After performing frequency encoding and k-fold target encoding we got the correlation between the all categorical features and the target. These correlations depicts that the strong predictors (correlation $>= 0.1$) are destination, passanger, weather, time and expiration. Although not all of the features in the strong_predictors list are really strong predictors. So, to reduce the dimensions of the dataset we want to implement the frequency encoding and feature encoding on them. For Data preprocessing, Sklearn's Pipeline and ColumnTransformer are used to do simple imputation, standardization, and one hot encoder for columns of mixed data types.

After performing data preprocessing, Simple Encoding (Plan A) has 33 features in it. Now, data preprocessing can be performed for One-

Hot Encoding (Plan B), which uses one-hot encoder for all the categorical features.

6.1.3 Model Training

We had frequency encoded and target encoded data of Plan A, and one hot encoded data of Plan B. We built basic models on these data plans and checked the performance between them.

Basic Models

The basic models we chose are faster to train and they are: Naive Bayes, Decision Tree, Logistic Regression, K Nearest Neighbor, and linear Support Vector Machine. RandomizedSearchCV will be used to choose the hyperparameters for both simple encoding (Plan A) and one-hot encoding (Plan B) from the same parameter grid and then results will be compared.

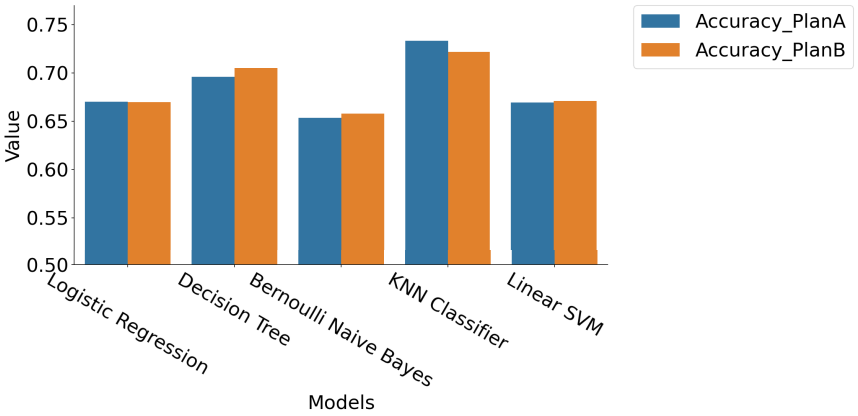

Figure 19: Testing accuracy of basic models

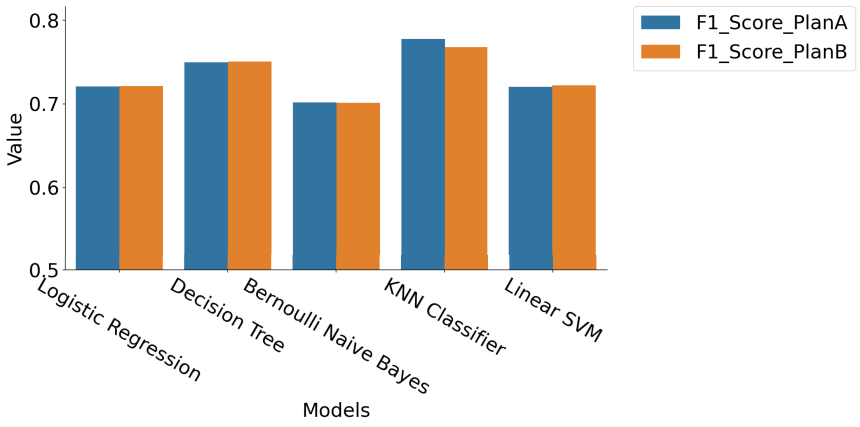

Figure 20: Testing F1 Scores of Basic Models

In Figures 19 and 20 , we visualize that the knn classifier built on data of simple encoding (Plan A) has a better accuracy. This is quite interpretable since knn classifier relies on distances, and our feature engineering in simple encoding (Plan A) transformed sparse categorical data into numerical frequency features, by doing that, distance among the feature space started to make more sense.

Decision tree built on data of one-hot encoding (Plan B) performs better than simple encoding (Plan A), that is because decision tree is good at splitting categorical data, so one-hot encoded features are more suitable for decision tree. As for F1 scores, knn classifier reached the highest testing F1 score on data of simple encoding (Plan A), indicating the best overall predictive accuracy behind it.

To improve the prediction results, we have two approaches: feature expansion and advanced models.

Features Expansion

Clustering method cab be used to cluster the training data and add the cluster label as new feature. Two popular clustering methods can be used

to apply feature expansion: hierarchical clustering and k-means clustering. However, hierarchical clustering is slower than k-means clustering, thus, we are using k-means for this feature expansion task.

K-prototypes Clustering: Since we have mixed data types, K-means is not appropriate here because euclidean distance is meaningless for categorical data. There are two variations of K-means clustering: K-modes is capable of clustering on categorical data, K-prototypes clustering is suitable for mixed data types.

There is a concern for the clustering algorithm: to include the target feature or not. If we include the target to get the cluster labels, and use those labels to classify the target, the information about target will have leaked into the features, and the accuracy will be overly optimistic. But when we evaluate them on the test set, this bias will diminish and disappear. Thus, we can also use k-fold cross validation to alleviate the data leakage problem. To determine which number of clusters to use in the K-prototypes clustering, we plotted a silhouette diagram

The silhouette score is the mean silhouette coefficient over all the instances within the cluster which is used to calculate the goodness of a clustering technique. It's value lies between -1 and +1. A coefficient value close to +1 indicates that the instance is inside its own cluster and farther away from other clusters. A coefficient value close to 0 indicates that the instance is close to a cluster boundary. Finally, a coefficient value close to –1 would mean that the instance is assigned to the wrong cluster.

According to the Figure 21, the silhouette score appears better at $k = 2, 3, 4, 5$. We are choosing $k = 4$ as the number of clusters in K-prototypes clustering for data in simple encoding (Plan A).

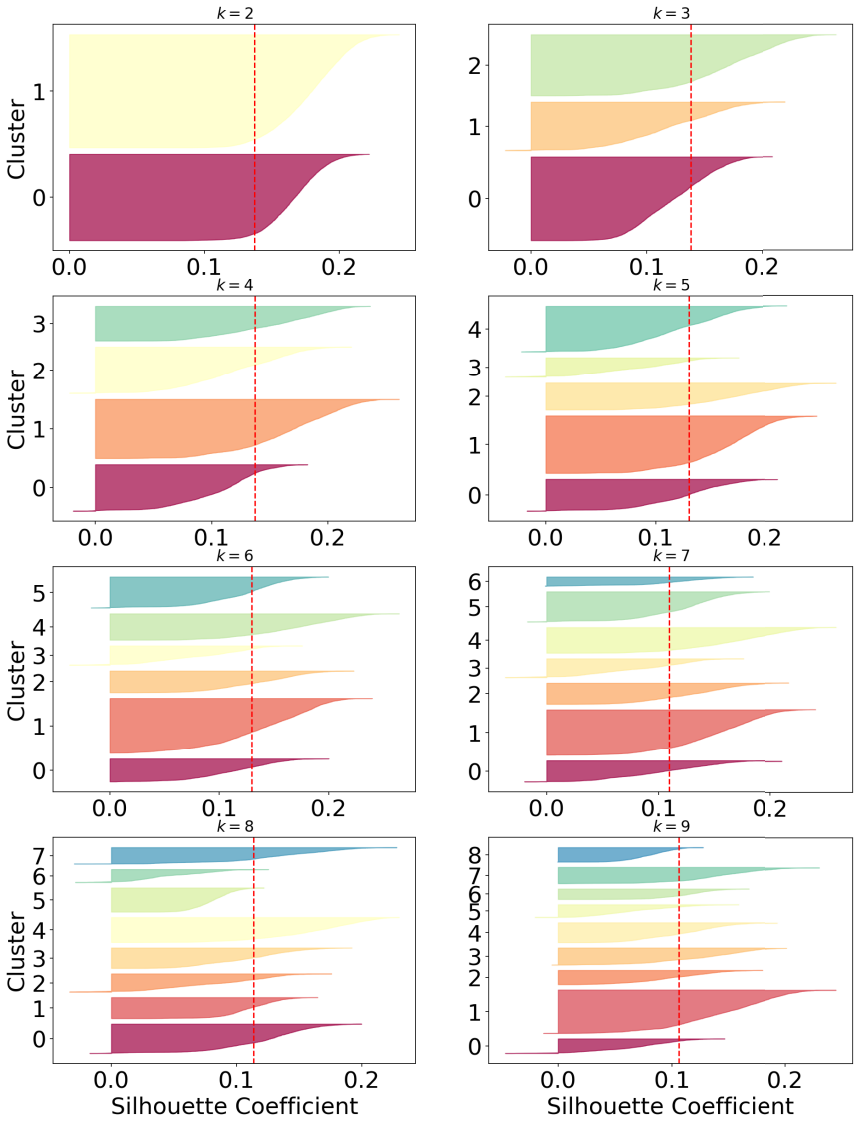

Figure 21: Silhouette

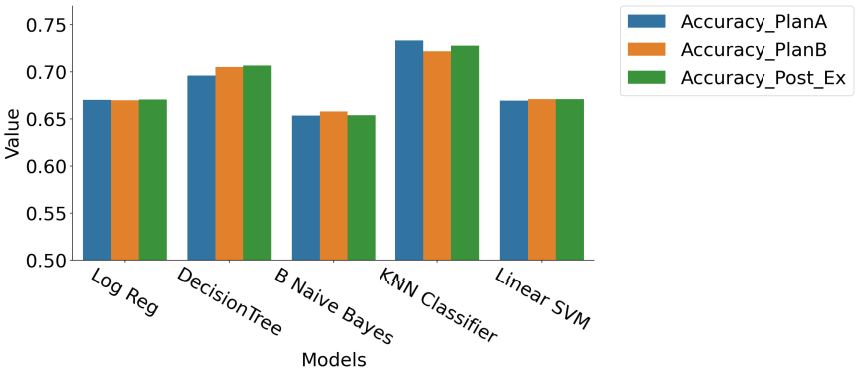

Figure 22: Testing accuracy post feature expansion

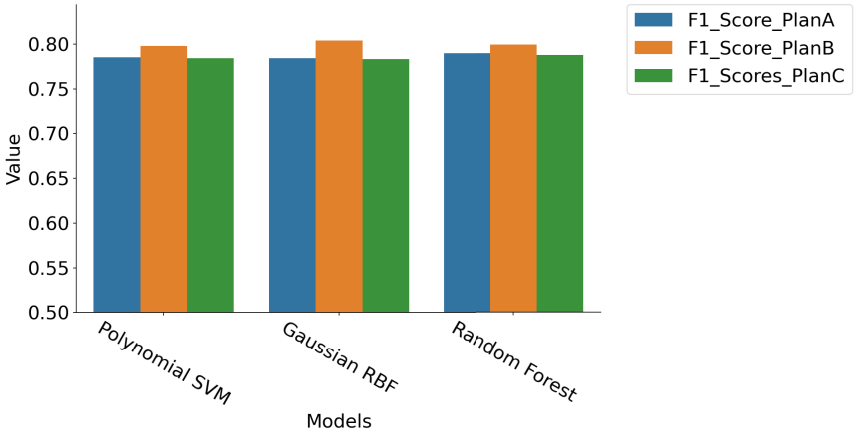

Figure 23: Testing F1 scores post feature expansion

Retrain Models After Adding Cluster Labels

According to Figures 22 and 23, feature expansion using K-prototypes clustering decreased the performance of KNN model, increased the performance of decision tree induction model. Both the highest accuracy and F1 scores were achieved by KNN classifier using data in Simple Encoding (Plan A). Since we used the same random search grids, better results could have been achieved using different hyperparameters. Anyway, it is already enough to see the power of feature expansion using K-prototypes clustering.

Advance Models

The advance models we picked are: polynomial SVM, Gaussian RBF SVM, and Random Forest

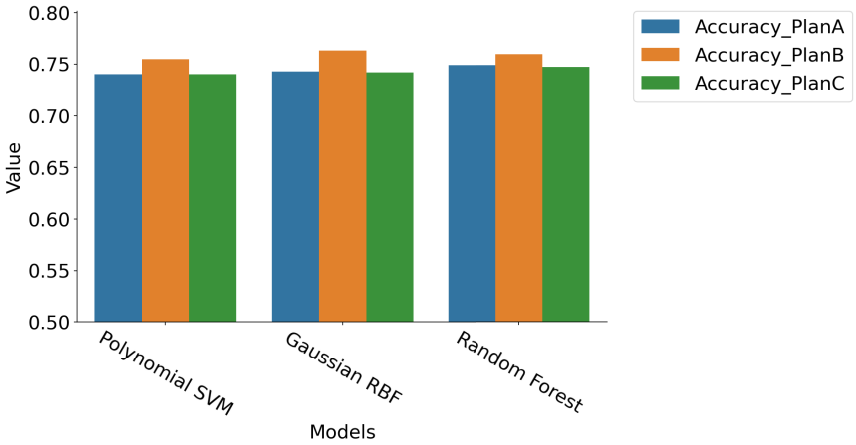

Figure 24: Testing accuracy of advance models

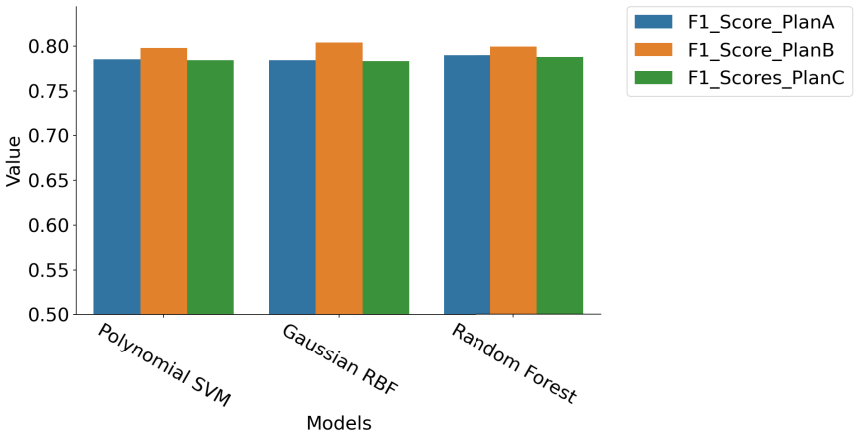

Figure 25: Testing F1 scores of advance models

Based on the Figures 24 and 25, data of one-hot encoding (plan B) achieved the highest accuracy and F1 score by Gaussian RBF SVM given the limited grid search hyperparameter space. However, the training time of one-hot encoding (Plan B) is more than twice as long as plan A and C

due to its high dimensions and sparsity. Besides, the features in plan A and C may achieve higher accuracy with different hyperparameters.

As we broke down the training time of three advanced models, we noticed that SVM models spent more time than Random Forest. One of the reason is that SVM in scikit learn does not support parallel computation.

Detailed Model Evaluation

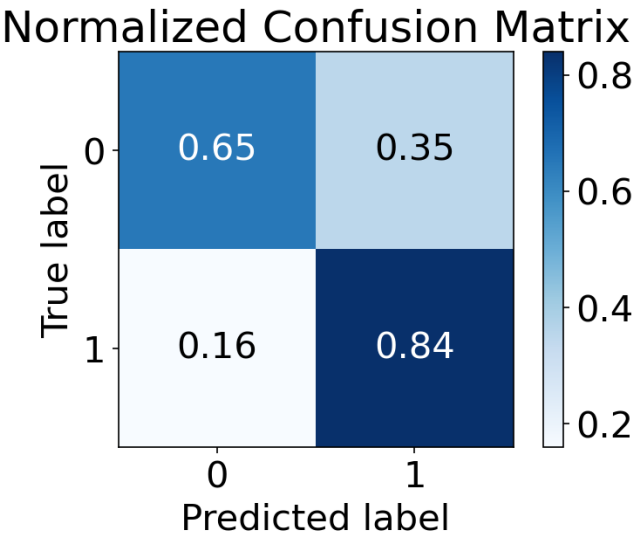

Figure 26: Confusion Matrix

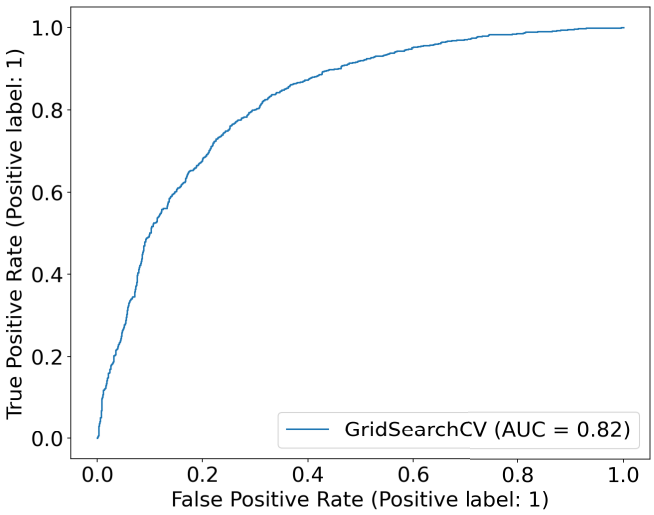

Figure 27: ROC Curve

Figures 26 and 27 show, in the detailed model evaluation Gaussian RBF SVM model trained on data in plan A has a high false positive rate.

6.1.4 Visualization of Feature Importance

We set a threshold that only those features whose importance is higher than 0.015 can be in this plot. For data A and C, the feature importance plots in Figures 28 and 29 are basically the same, and the cluster labels we added by K-prototypes clustering does not show up in the feature importance plot, which means it is not that important for the Random Forest model. If we check the top 10 most important features, they are coupon type and occupation type's target and frequency encoding, how frequent to go to coffee houses, education and time's target encoding, income, and age.

For feature set B, we can see that the frequency features: CoffeeHouse, Bar, CarryAway, RestaurantLessThan20, and Restaurant20To50 are on the upper part of the plot in Figure 30, indicating their importance. Numerical features like income, age, temperature, and distance are quite

important as well as the Education level. Coupon features like coupon type and coupon expiration time are also in the plot. Indeed, there are some similarity between data in B and A, C.

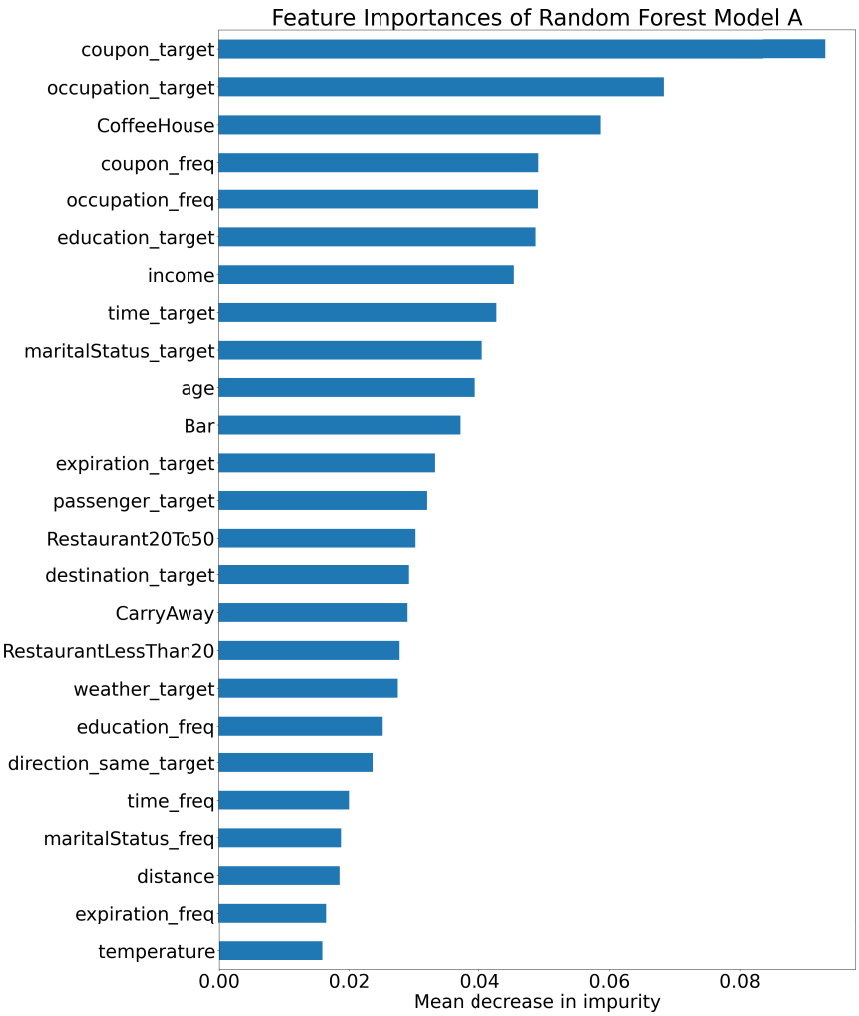

Figure 28: Feature importance of model A

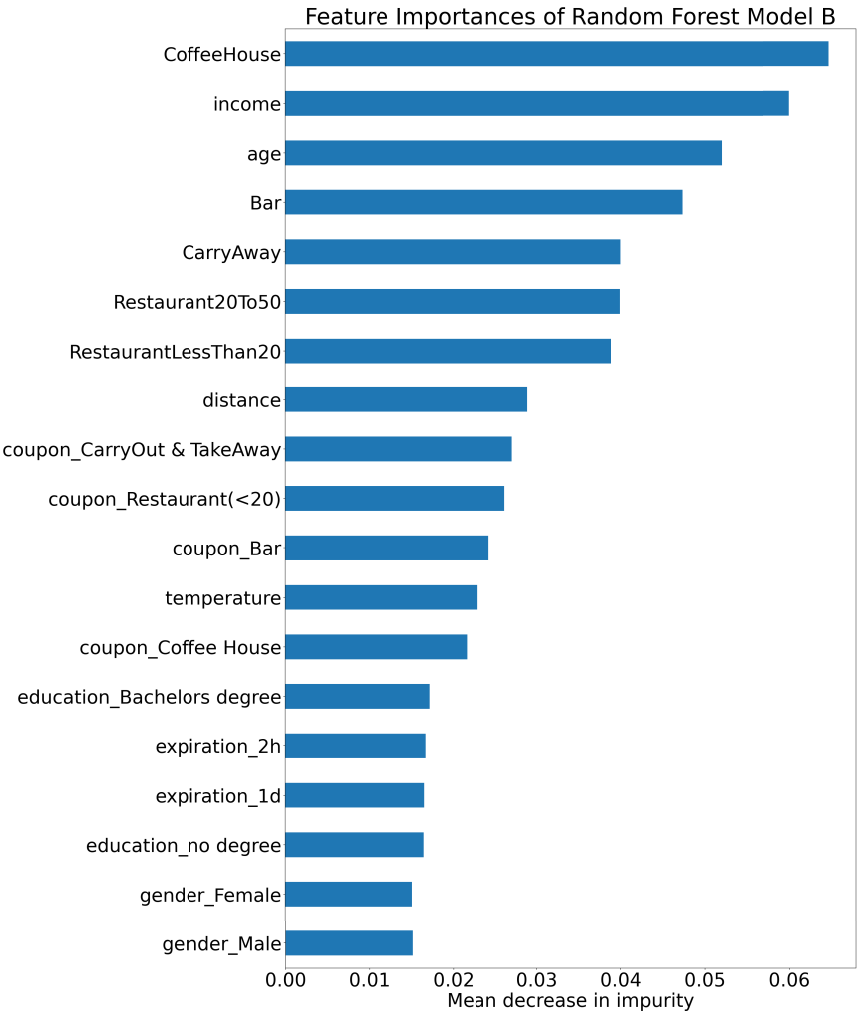

Figure 29: Feature importance of model B

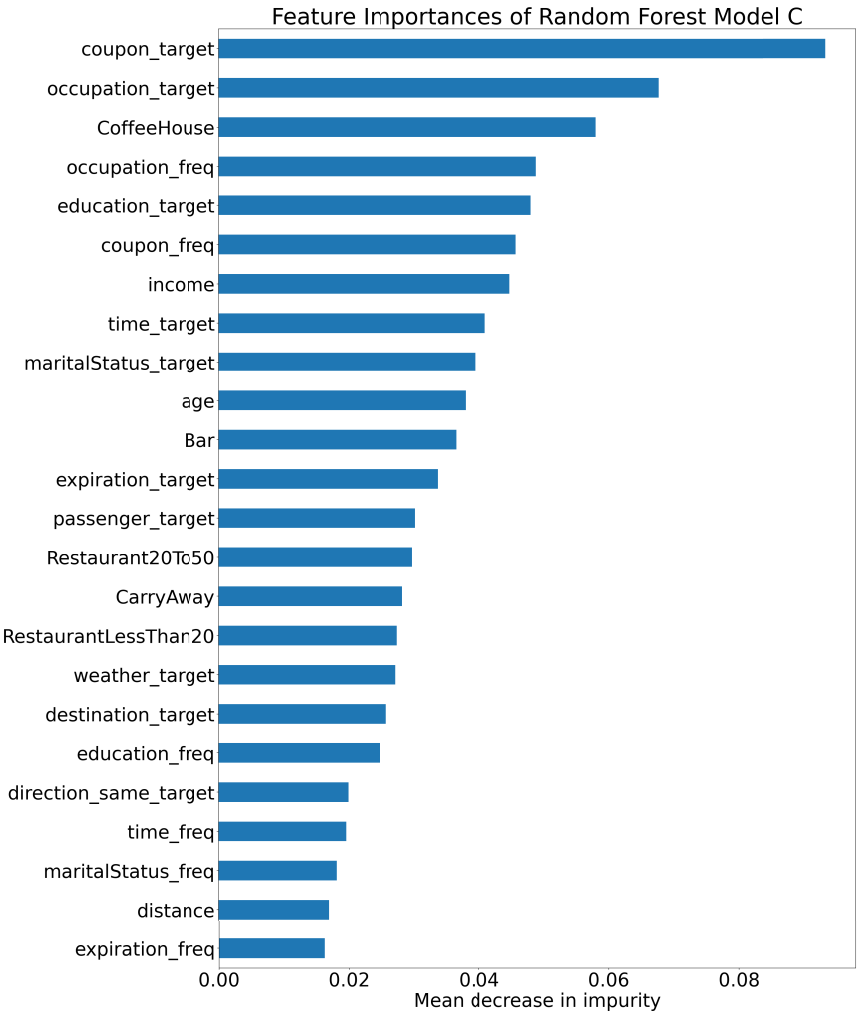

Figure 30: Feature importance of model C

Name	Description
age:	Patient's age in years
sex:	Patient's gender (1: male, 0: female)
chest pain:	Chest pain type experienced by the person (0: typical angina, 1: atypical angina, 2: non-anginal, 3: asymptomatic)
rest bp:	Resting blood pressure of the patient when admitted to the hospital (mm Hg)
chol:	Serum cholesterol measurements of the patient in mg/dl (unit)
fbs:	Fasting blood sugar value of an individual comparative to 120mg/dl (1: $>$ 120mg/dl, 0: $<$ 120mg/dl).
rest ecg:	Resting electrocardiographic results of the patient (0: normal , 1: ST-T wave abnormality , 2: ventricular hypertrophy)
max heart rate:	Patient's maximum heart rate
exercise angina:	Exercise induced angina (0: no, 1: yes)
ST dep:	Exercise induced by ST depression relative to rest (position on the ST plot)
slope:	Peak exercise slope's ST segment (0: upsloping , 1: flat , 2: downsloping)
major vessels	count of major vessels (0–3)
thal:	Thalassemia, a blood disorder (1: normal , 2: fixed defect , 3: reversible defect)
target:	presence of a heart disease in a patient (0 = no, 1 = yes)

Table 3: Column Description of Heart Disease Dataset

6.2 Scikit-learn with Heart Disease Dataset

The brief description of the abbreviations used in dataset are given in the Table 3

	age	sex	chest pain	rest bp	chol	fbs	rest ecg	max heart rate
mean	54.36	0.68	0.96	131.62	246.26	0.14	0.52	149.64
std	9.08	0.46	1.03	17.53	51.83	0.35	0.52	22.90
min	29.0	0.0	0.0	94.0	126.0	0.0	0.0	71.0
25%	47.50	0.0	0.0	120.0	211.0	0.0	0.0	133.50
50%	55.0	1.0	1.0	130.0	240.0	0.0	1.0	153.0
75%	61.0	1.0	2.0	140.0	274.50	0.0	1.0	166.0
max	77.0	1.0	3.0	200.0	564.0	0.0	2.0	202.0

	exercise angina	ST dep	slope	major vessels	thal	target
mean	0.32	1.03	1.39	0.72	2.31	0.54
std	0.46	1.16	0.61	1.02	0.61	0.49
min	0.0	0.0	0.0	0.0	0.0	0.0
25%	0.0	0.0	1.0	0.0	2.0	0.0
50%	0.0	0.80	1.0	0.0	2.0	1.0
75%	1.0	1.60	2.0	1.0	3.0	1.0
max	1.0	6.20	2.0	4.0	3.0	1.0

Table 4: Descriptive Statistics of Heart Disease Dataset

6.2.1 Data Wrangling

As per the dataset description, feature "thal" ranges from 1-3 but we have seen 4 different values and "major_vessels" ranges from 0-3 but we have seen 5 different values by printing the unique values across the columns. Then, we found the count of invalid and out of range values in the dataset. In the "thal" column we were getting two values as '0' and in "major_vessels" column five entries as '4', which were not expected and out of range. We decided to replace all these out of range values with "Null". Now, the dataset contains missing values in the two columns. Later, we replaced "Null" with median for these numerical features.

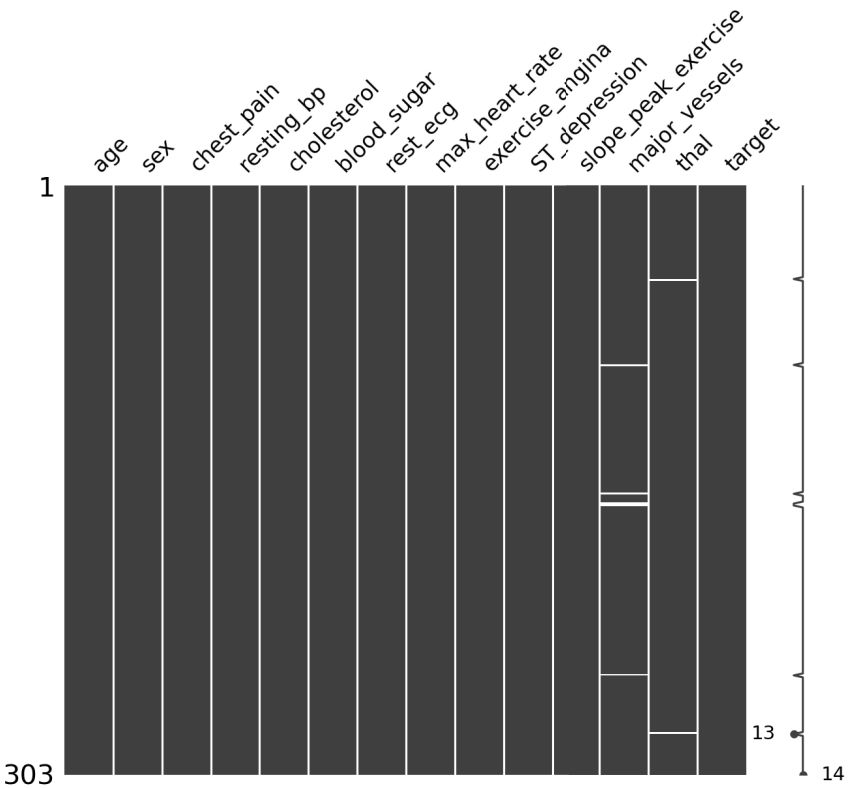

Figure 31: Matrix Plot

We have identified following distinct features as:

Categorical Features	Numerical Features
sex	age
chest_pain	resting_bp
blood_sugar	cholestrol
rest_ecg	max_heart_rate
exercise_angina	ST_dep
slope	major_vessels
thal	-

Table 5: Classification of features

6.2.2 Exploratory Data Analysis

Categorical Features

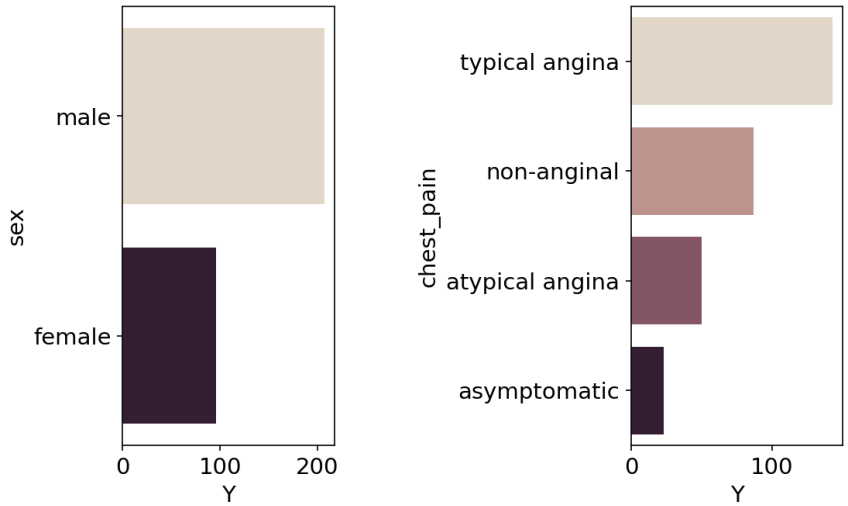

Figure 32: Count plot of sex and chest_pain with target "Y"

sex: the count of "Male" patients is twice than "Female", as show in Figure 32

chest_pain: patients with "typical angina" has highest occurrences, as show in Figure 32

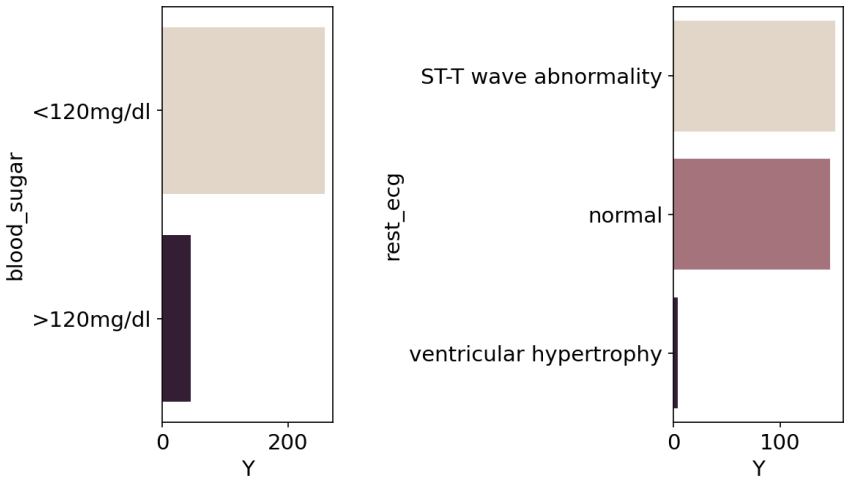

Figure 33: Count plot of blood_sugar and rest_ecg with target "Y"

blood_sugar: the count of patients having blood sugar level "<120mg/dl" is almost 90%, as shown in Figure 33.

rest_ecg: patients showing ST-T wave abnormality and normal has equal occurrences while there are few patients with ventricular hypertrophy, as shown in Figure 33.

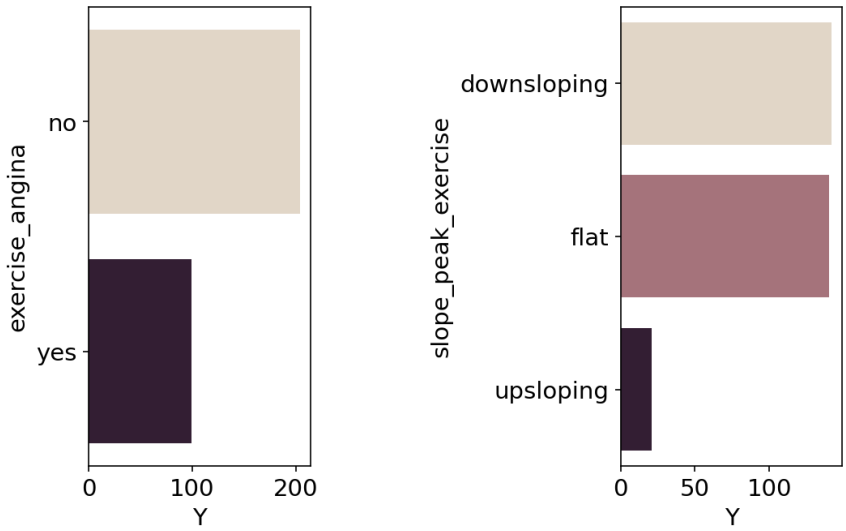

Figure 34: Count plot of exercise_angina and slope_peak_exercise with target "Y"

exercise_angina: patients with exercise_angina has a greater count, as shown in Figure 34.

slope_peak_exercise: Most of the people had either Flat or Downsloping peak, as shown in Figure 34.

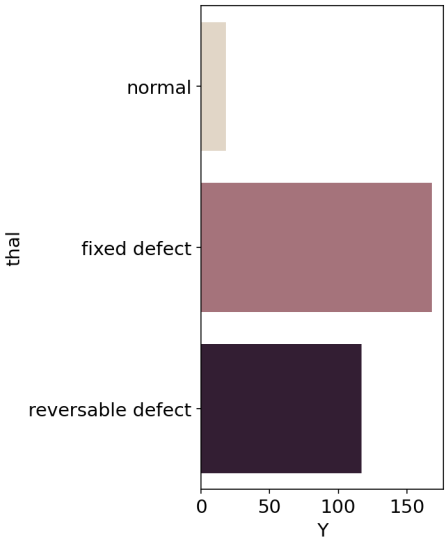

Figure 35: Count plot of thal with target "Y"

thal: there are significant number of patients with fixed defect thalassemia and reversible defect thalassemia. Patients with normal thalassemia showing a negligible count, as shown in Figure 35.

By looking at the counting plot of the categorical features, we realized that there are two kinds of categorical data: **ordinal** and **nominal**, we will apply one-hot encoder to nominal features and map the ordinal data into numerical to preserve the inner order.

Nominal Features

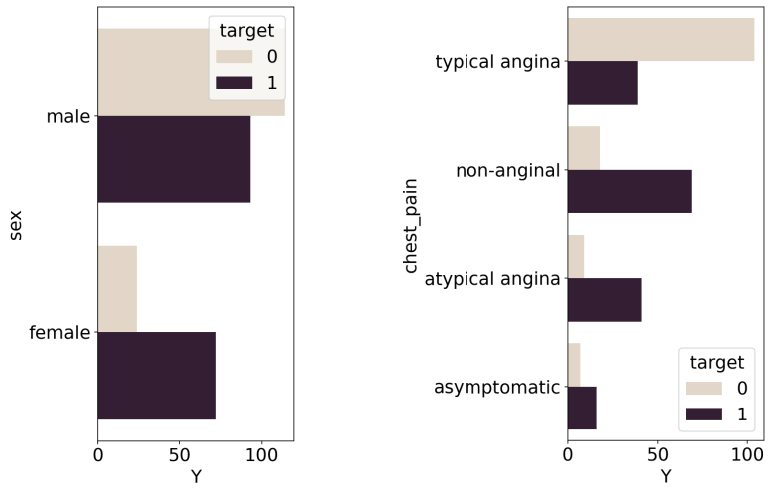

Figure 36: Relationship of sex and chest_pain_type with target "Y"

sex: Women experience heart attacks more than men, as shown in Figure 36.

chest_pain: A commonality among heart disease patients is the presence of non-anginal type of chest pain, as shown in Figure 36.

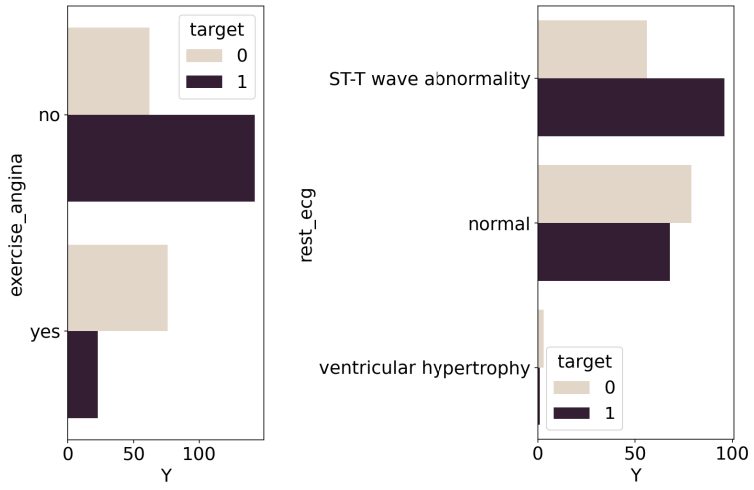

Figure 37: Relationship of exercise_angina and rest_ecg with target "Y"

exercise_angina: people with angina induced without exercise have more heart disease than people with angina induced by exercise, as shown in Figure 37.

rest_ecg: patients that had ST-wave abnormality are susceptible to a heart disease, as shown in Figure 37.

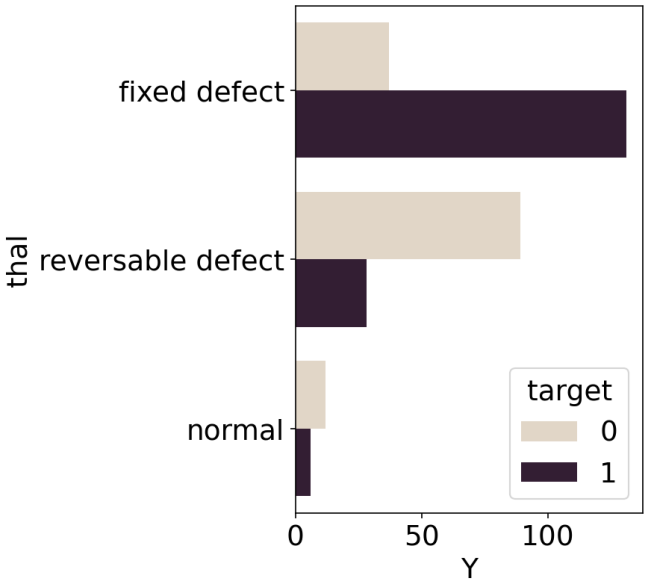

Figure 38: Relationship of thal with target ”Y”

thal: Fixed defect type of thalassemia patients are most likely to have a heart disease, as shown in Figure 38.

Numerical Features

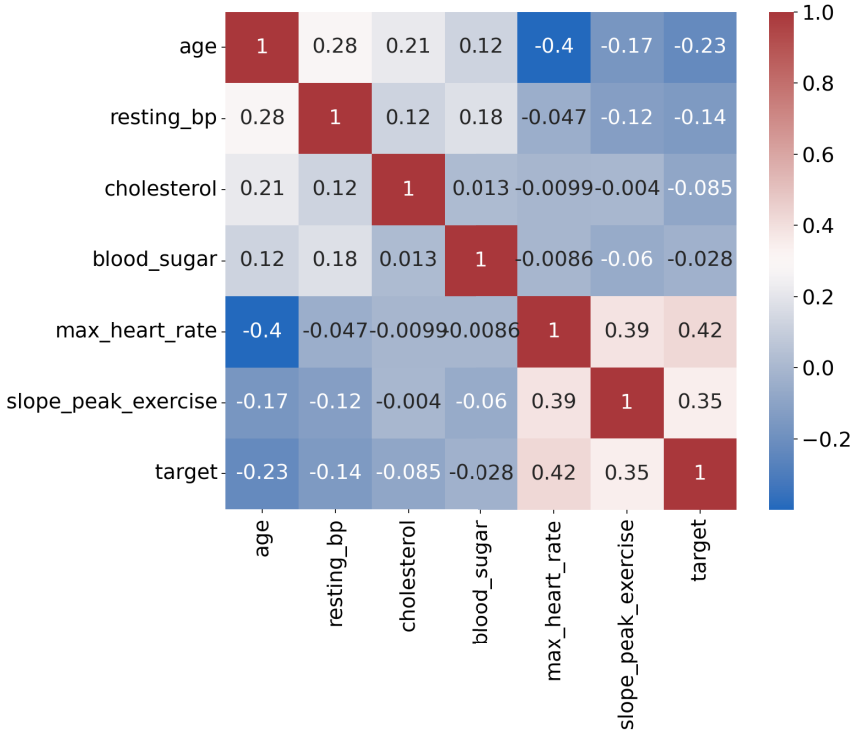

Figure 39: Correlation of numerical features for heart disease dataset

In Figure 39, "max_heart_rate" and "slope_peak_exercise" is showing a strong positive correlation with target.

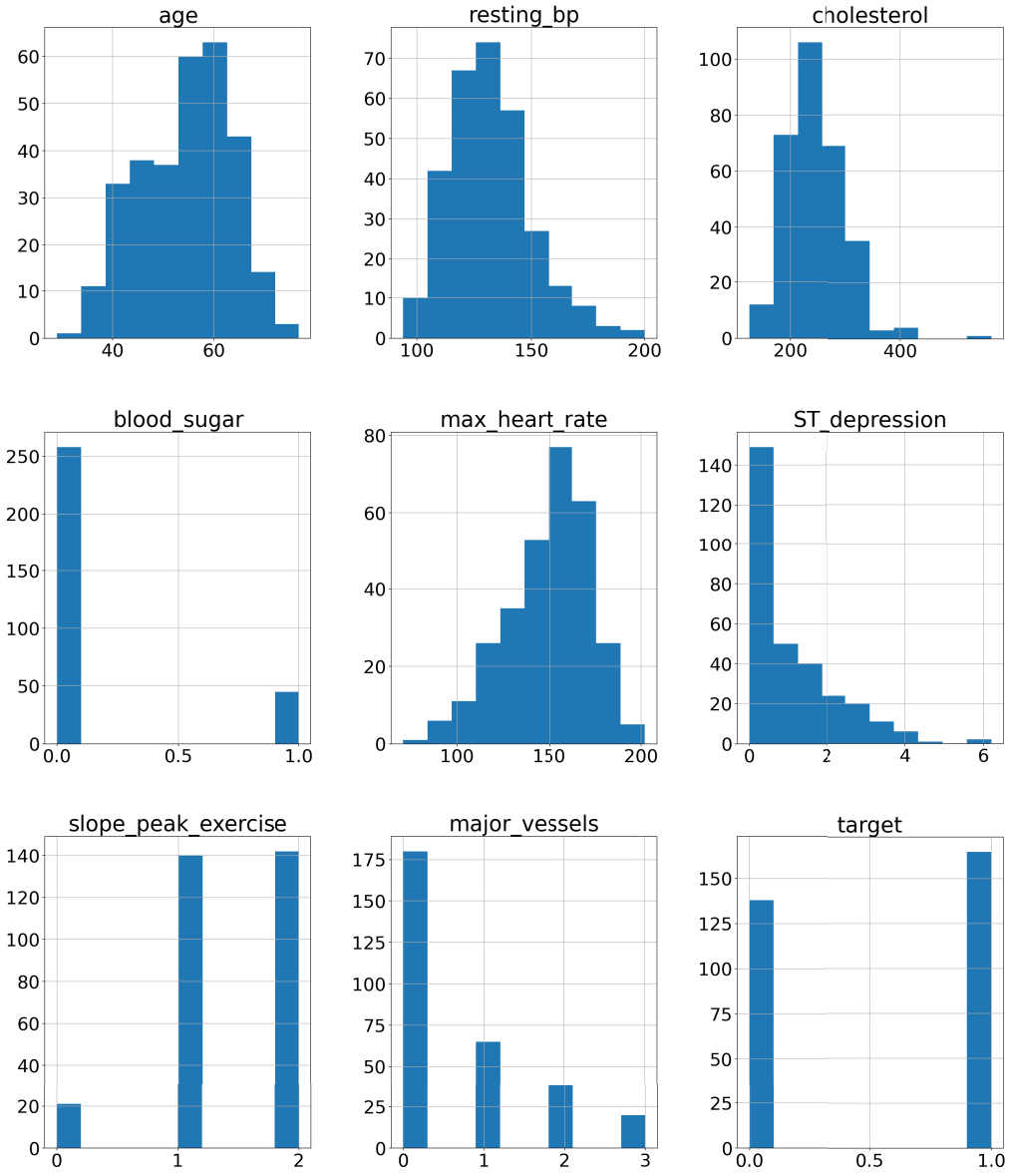

Figure 40: Histogram for heart disease dataset

From Figure 40, we can see the frequency distribution of numerical as well as mapped ordinal columns.

Simple Encoding (Plan A) and **One-Hot Encoding (Plan B)** has been used to compare the effect of different feature engineering.

After performing frequency encoding and k-fold target encoding we get the correlation between the all categorical features and the target. These correlations depicts that the strong predictors (correlation >= 0.1) are sex, chest_pain, slope_peak_exercise, major_vessels and thal.

After data preprocessing, we now have 22 features in Simple Encoding (plan A). Now, we can do data preprocessing for one-hot encoding (Plan B), which basically uses OneHotEncoder for all categorical features.

6.2.3 Model Training

We will build basic models on the data of simple encoding (Plan A) and data of one-hot encoding (Plan B) to check and compare the performance of plan A and plan B.

Basic Models

The basic models we chose are: Logistic Regression, Decision Tree, Naive Bayes, K Nearest Neighbor, and linear Support Vector Machine. RandomizedSearchCV will be used to choose the hyperparameters for both Plan A and Plan B from the same parameter grid and then results will be compared.

From Figure 41 and 42, we can see that Logistic Regression built on data of Plan A and Plan B is showing similar performance and has a better accuracy than others. Decision tree built on data of plan B performs better than plan A. However, Bernoulli Naive Bayes, KNN Classifier and Linear SVM has a equal value of accuracy. As for F1 scores, Linear Regression reached the highest score, thus indicating best overall predictive accuracy on both data of plan A and plan B.

To improve the prediction results, we have two approaches: feature

expansion and advanced models.

Features Expansion

To apply feature expansion we used k-prototype clustering method which is a variation of k-means clustering and is suitable to use for mixed data types.

To determine which number of clusters to use in the K-prototypes clustering, we plotted a silhouette diagram.

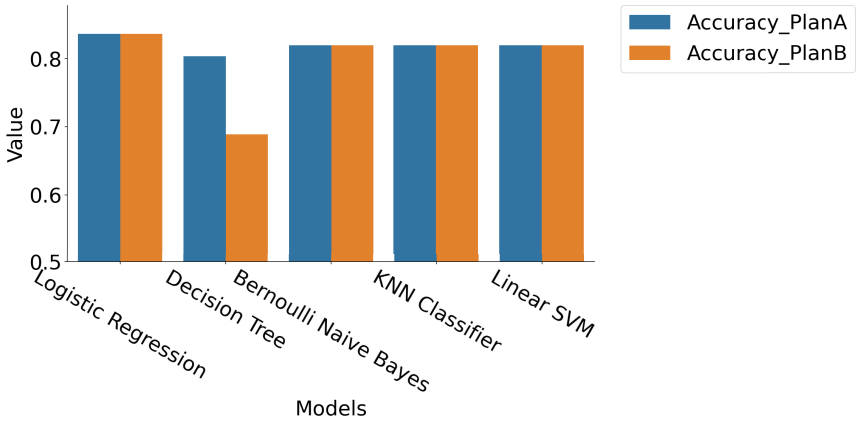

Figure 41: Testing accuracy of basic models

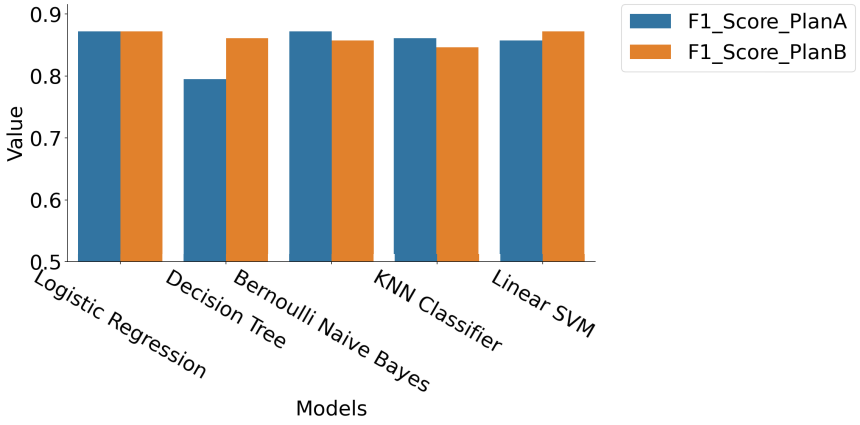

Figure 42: Testing F1 score of basic models

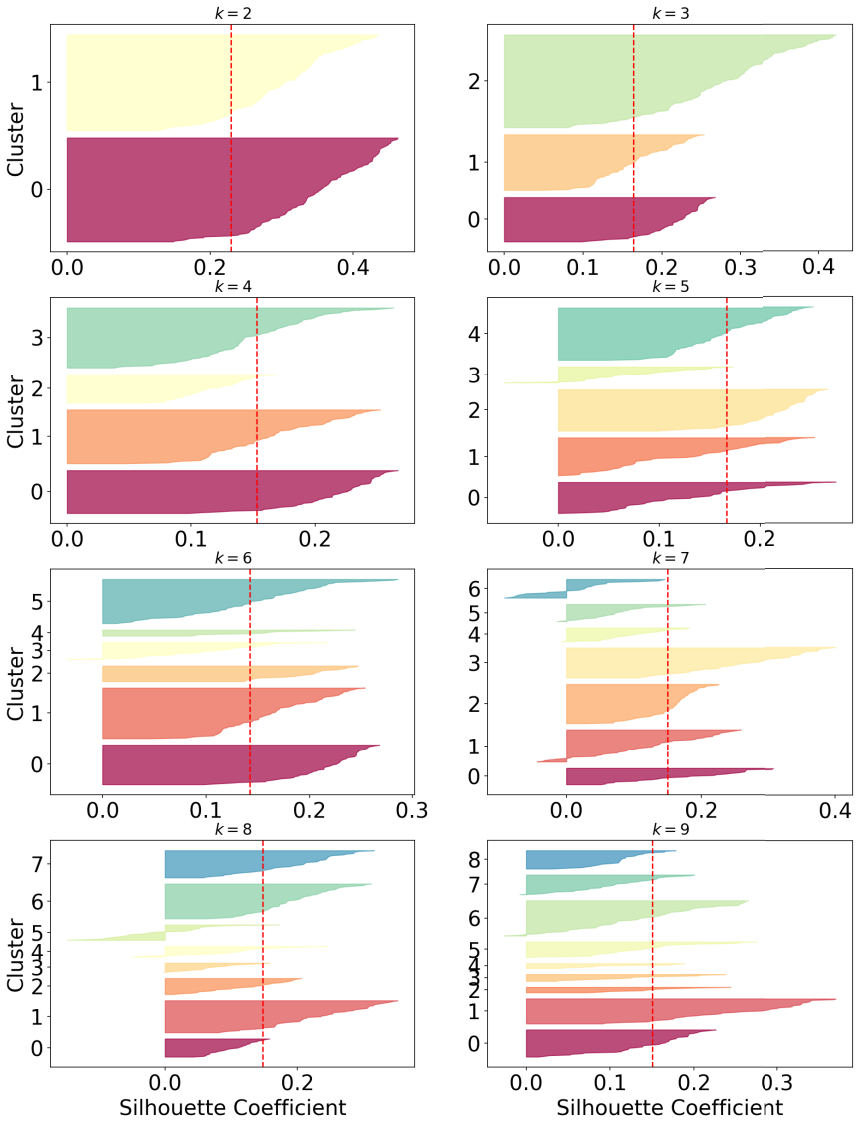

Figure 43: Silhouette

From Figure 43, we chose $k = 2$ as the number of clusters of the k-prototype clustering for data in plan A.

Retrain Models After Adding Cluster Labels

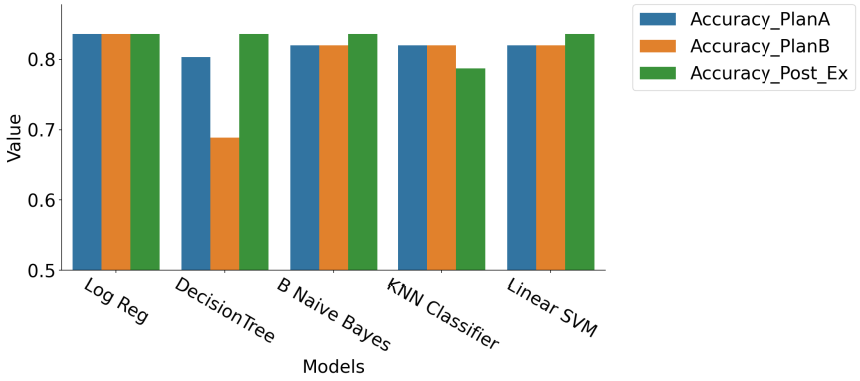

Figure 44: Testing accuracy post feature expansion

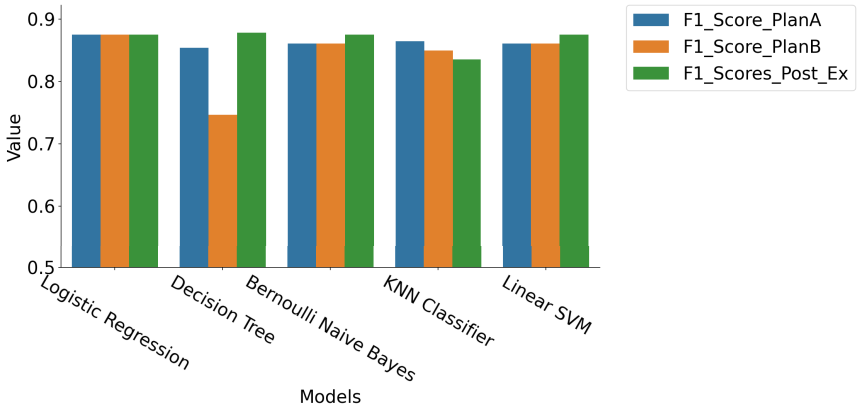

Figure 45: Testing F1 scores post feature expansion

From Figures 44 and 45, We could see that feature expansion using k-prototypes clustering increased the performance of all models. Both the highest accuracy and F1 scores were achieved after this feature expansion. Post feature expansion, Logistic Regression performs the best as we observed in the previous plan as well. Logistic Regression has the highest F1 score after feature expansion. The feature expansion using clustering features does not help the decision tree model to achieve better performance.

Advance Models

The Advance models we picked are: polynomial SVM, Gaussian RBF SVM, and Random Forest.

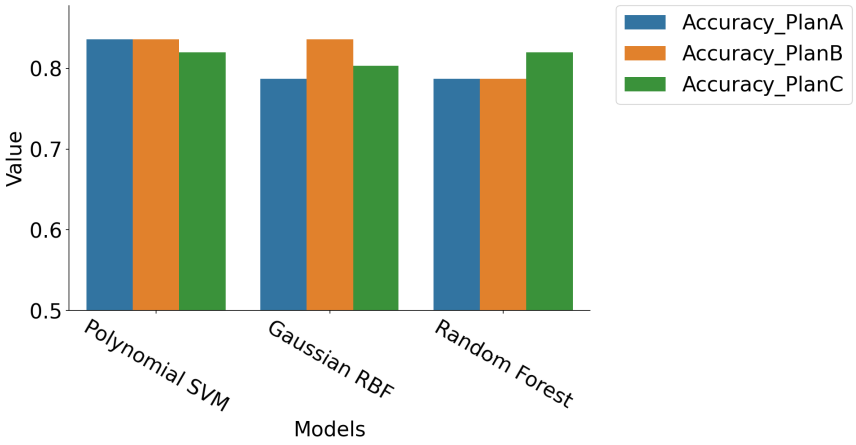

Figure 46: Testing accuracy of advance models for heart disease dataset

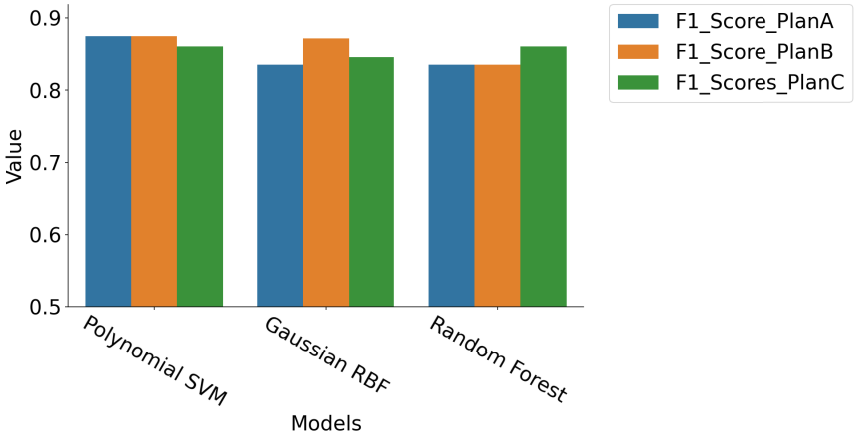

Figure 47: Testing F1 score of advance models for hear disease dataset

According to the Figures 46 and 47, feature set of plan A and plan B achieved the equal and highest accuracy and F1 score by Polynomial SVM. However, the training time of plan B is more than twice as long as Plan A and Plan C.

Detailed Model Evaluation

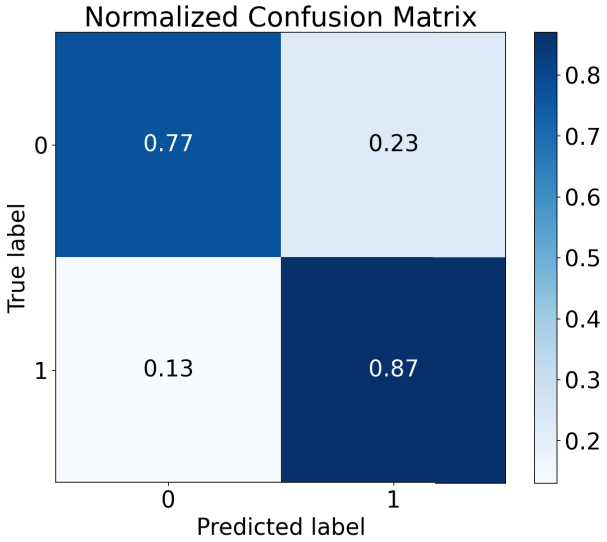

Figure 48: Confusion matrix for heart disease

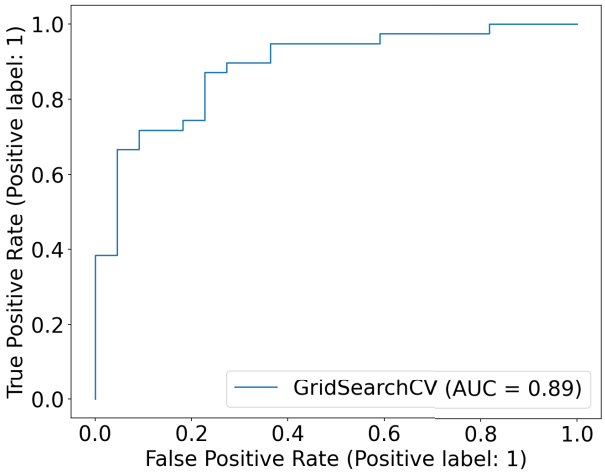

Figure 49: ROC curve for heart disease

In Figures 48 and 49, confusion matrix and roc curve shows the detailed model evaluation. Polynomial SVM model trained on data in plan A has a high false positive rate.

6.2.4 Visualization of Feature Importance

Since we trained Random Forest models, we can have a look of the feature importance by plotting them.

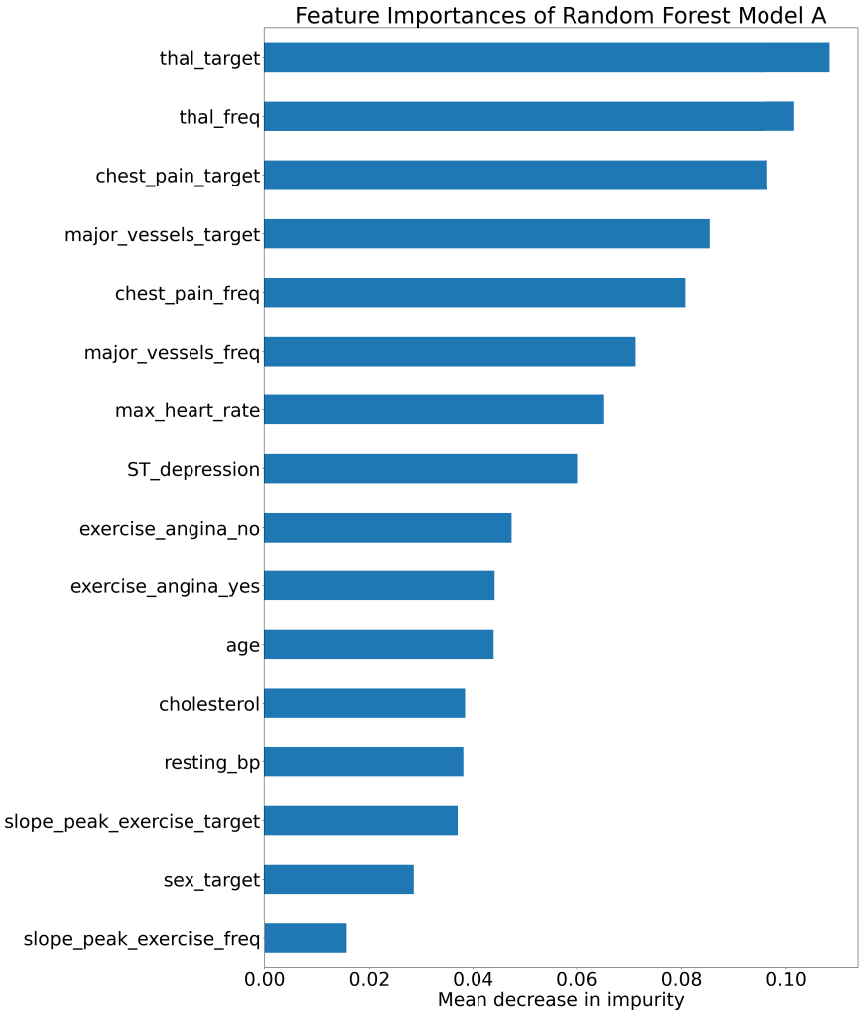

Figure 50: Feature Importance of plan A for heart disease

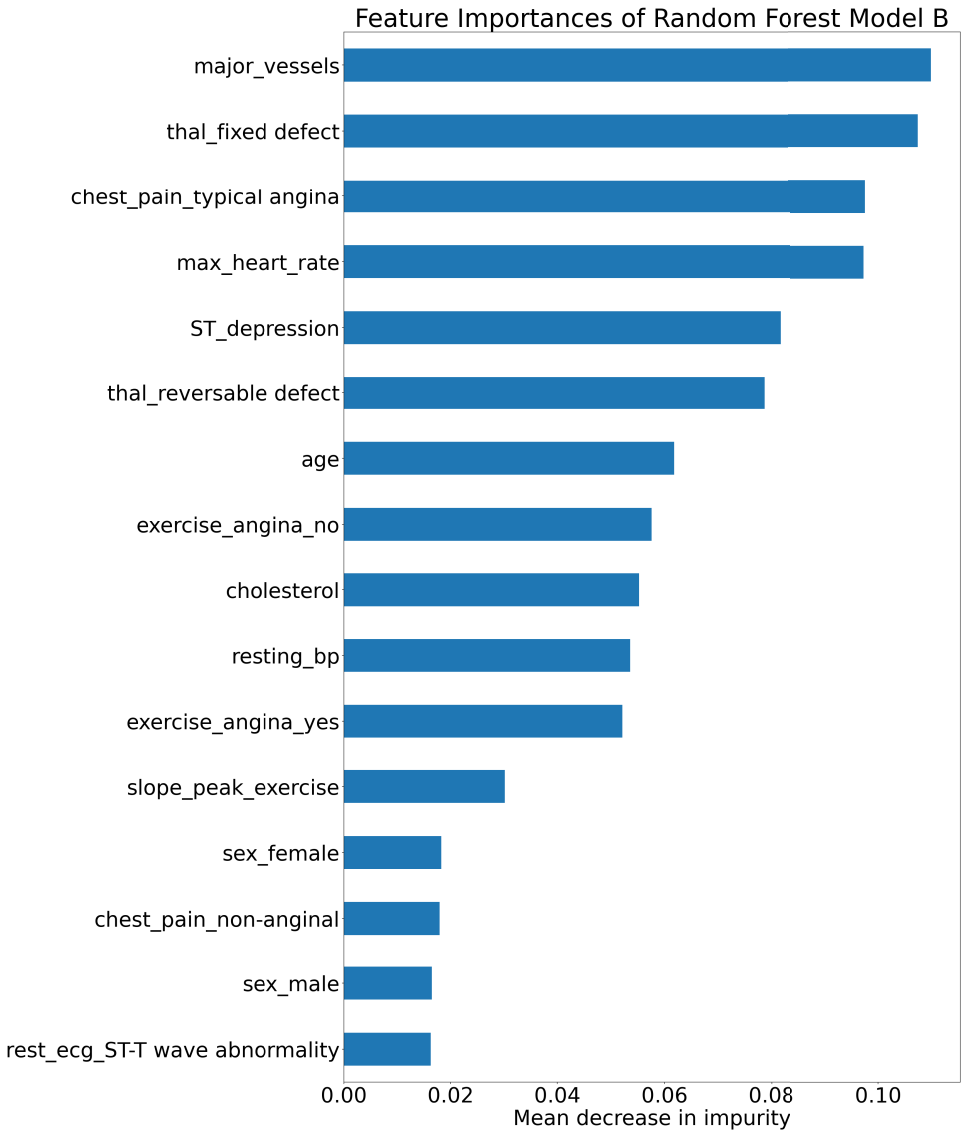

Figure 51: Feature Importance of plan B for heart disease

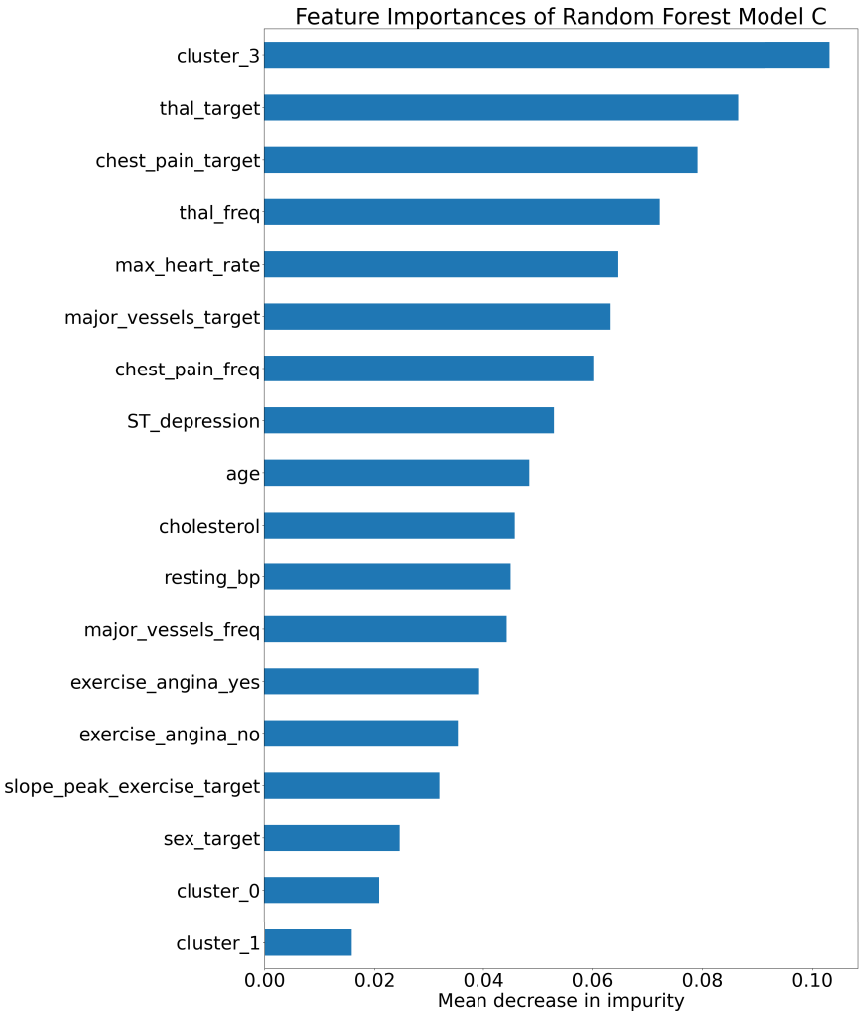

Figure 52: Feature Importance of plan C for heart disease

We set a threshold that only those features whose importance is higher than 0.015 can be in this plot. For data A and C, the feature importance plots, shown in Figures 50 and 52 are basically the same, and the cluster labels we added by the K-prototypes clustering do not show up in the feature importance plot, which means it is not that important at least in Random Forest model. If ee check the top 10 most important features, they are thal, chest_pain, major_vessels's and target and frequency encoding, ST_depression, exercise_angina_no and exercise_angina_yes.

For feature set B, shown in Figure 51 we can see that the features: major_vessels, thal_fixed defect, chest_pain_typical angina, max_heart_rate and ST_depression are on the upper part of the plot, indicating their importance. Features like age, exercise_angina, cholesterol and resting_bp are quite important as well. Indeed, there are some similarity between data in B and A, C.

In this chapter, we have shown the experimental results and compared their different aspects in detail. A general discussion is given in the last section.

7 Conclusion

The research aimed at evaluating the impact of data quality on interpretability and predictability of ML models. The first dataset had inconsistencies in the form of missing or incorrect values; thus we obtained mixed performance results with the different ML models. Gaussian RBF SVM was the most accurate with 76% outcome prediction accuracy, while other models' accuracy results varied between 62% to 70%. The Heart Disease dataset was fairly complete with less than 1% of quality issues; consequently, the accuracy results for the majority of Machine Learning models were fairly consistent in the range of 81% to 84%. Thus, a better quality data did improve the interpretability and outcome prediction accuracy of the ML models.

8 References

[1] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[2] Xu Chu, Ihab F Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206, 2016.

[3] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.

[4] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[5] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

[6] Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger. Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 11(12):1781–1794, 2018.

[7] Shrey Shrivastava, Dhaval Patel, Anuradha Bhamidipaty, Wesley M. Gifford, Stuart A. Siegel, Venkata Sitaramagiridharganesh Ganapavarapu, and Jayant R. Kalagnanam. Dqa: Scalable, automated and

interactive data quality advisor. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2913–2922, 2019.

[8] Kelli Ham. Openrefine (version 2.5). http://openrefine.org. free, open-source tool for cleaning and transforming data. *Journal of the Medical Library Association : JMLA*, 101:233–234, 07 2013.

[9] Patricio Cerda and Gaël Varoquaux. Encoding high-cardinality string categorical variables. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[10] Otmane Azeroual. Data wrangling in database systems: purging of dirty data. *Data*, 5(2):50, 2020.

[11] John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, MA, 1977.

[12] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.

[13] Shuntaro Okada, Masayuki Ohzeki, and Shinichiro Taguchi. Efficient partition of integer optimization problems with one-hot encoding. *Scientific reports*, 9(1):1–12, 2019.

[14] Raymond E Wright. Logistic regression. 1995.

[15] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[16] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support

vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[17] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[18] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[19] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[20] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. Catboost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

[21] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[22] Aleksandra Płońska and Piotr Płoński. Mljar: State-of-the-art automated machine learning framework for tabular data. version 0.10.3, 2021.

[23] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global under-

standing with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[24] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. Evaluating feature importance estimates. 2018.

[25] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

[26] D Aha and Dennis Kibler. Instance-based prediction of heart-disease presence with the cleveland database. *University of California*, 3(1):3–2, 1988.

[27] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. *Jupyter Notebooks-a publishing format for reproducible computational workflows.*, volume 2016. 2016.

[28] Ohio Supercomputer Center. Ohio supercomputer center, 1987.

[29] Maurizio Petrelli. Setting up your python environment, easily. In *Introduction to Python in Earth Science Data Analysis*, pages 3–9. Springer, 2021.

[30] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine

learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.