NOVA IMS Information Management School

Machine Learning Project

MASTER DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

The Smith Parasite

Group 06

Amanda França, number: 20220708

Andrei Macovei, number: 20221358

Tomás Domingos, number: 20221370

Virgínia Aguiar, number: 20220707

December, 2022

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA IMS Information Management School

Introduction

This report concerns a new disease transmitted by a virus that has been discovered in England by Dr. Smith and has already affected over 5000 people. Some of the people contracting the disease are asymptomatic, but the most common symptoms include fever and tiredness. The virus has also been associated with post-disease conditions such as loss of speech, confusion, chest pain and shortness of breath. The circumstances regarding the transmission of the disease are unknown and although there is no apparent connection between the infected people, some groups seem more prone to be infected than others.

The goal of this project is to develop and validate a classification model that predicts if a patient will suffer or not from the Smith Disease based on sociodemographic, health and behavioral data. The efficiency of the predictions made by the model will be measured by the F1 score.

Exploration

In order to build the predictive model three train datasets were used, namely: Sociodemographic Data, Health Related Data, and Habits Related Data. The first dataset stores the patients' name, birth year, living region, education level and the dependent variable disease. The second dataset corresponds to patients' information such as height, weight, cholesterol value, and blood pressure. The latter includes data about the patients' smoking, drinking, and exercise habits as well as fruit and water consumption. In the interest of exploring and preprocessing the data available in an efficient manner, the three previously mentioned train datasets were joined using the *PatientId* as an index into a dataset named *Train a*nd the same was done to the test data, creating the *Test dataset.*

The first step into data analysis is data exploration. In this chapter we explored and visualized the information available to uncover insights from the start and identify possible problems.
Prior to any analysis, the Train dataset was split in two: *Train*, corresponding to 70% of the data available and *Validation*, storing the remainder 30%. This slicing was made by stratification, which means both datasets store the same percentage of patients presenting the disease versus patients without the disease. The *Train* set will be used to fit the parameters to the learning model whilst the *Validation* set will be used to get an unbiased evaluation of the model fitted. After the split, both sets along with the *Test* dataset were explored in the same manner.

The first thing that needed to be done was to check for missing values and duplicates. During this step any blank spaces were converted into NaN, and it was identified that there were 9 observations missing on the *Train* dataset and 4 on the *Validation* dataset, both under the *Education* variable but no duplicated entries. Then, data types were examined, indicating that *Smoking_Habits* and *Exercise* were stored as objects, when they should be boolean. Subsequently unique values for the observations were analyzed and it was stated that for the *Region* variable there were two values with the same name but different letter cases: 'London' and 'LONDON', which indicated an inconsistency.

For further analysis, the descriptive statistics of the datasets was examined first using the Profile Report which provided an instant exploratory data analysis with interactive reports and then with the describe function that made possible to analyze the mean, standard deviation, and quartile values of the variables. At this point, it became clear that there were incoherencies, such as two patients with the same name, and possible outliers' indications in the *Birth_Year* feature, for instance, that presented a high standard deviation. We could also observe that the High_Cholesterol and Blood_Pressure variables had the maximum value extremely high, indicating the presence of extreme values that needed to be analyzed later in more detail. Also, at this stage it was identified that the *Name* variable had high cardinality and would not be needed in the models. Furthermore, due to the mean of 0,51 in the target variable Disease, we understood that its distribution on the dataset was balanced.

Moreover, to make it more efficient to work on the data analysis, the metric and non-metric data were split into different categories, this allowed for clear visualization of the variables regarding their different nature. To observe the frequency of each categorical variable, an absolute frequency chart was created, displayed in Figure 1 and 2, and it became noticeable that while *Drinking_Habit's* and *Diabetes's* observations were split almost equally between two categories, other variables such as Fruit_Habit had a very unbalanced distribution. To analyze the distribution of the metric observations, a box plot was created for each numeric feature, as shown in Figure 3 and 4, and here we could observe some extreme values namely on the variables *Birth_Year*, *High_Cholesterol*, *Blood_Pressure*, *Mental_Health* and *Physical_Health.* To end the data exploration, a pairwise graph was plotted to study the relationship between metric features, shown in Figure 5 and 6, in which we can notice a positive correlation between *Height* and *Weight*, *Mental_Health* and *Birth_Year*, and a negative correlation between *Weight* and *Physical_Health*.

Preprocessing

Data Cleaning

After checking for possible issues on the dataset, the data cleaning process begins. It is important to state that from this point onward, whenever there was a change being made to the dataset a copy of it was kept in order to have a faster roll back to the previous step and to keep the original dataset untouched.

Initially, the *Name* column is dropped, and wrong data types and inconsistencies are treated. In this step, *Smoking_Habit* and *Exercise* are converted into booleans and the value 'LONDON' in the *Region* variable is altered to 'London'. Then, we proceed to deal with the missing data. As seen in the data exploration step, the *Education* variable is the only one presenting missing values, which are filled in here with the mode of the column, which is 'University Complete (3 or more years)'. It is important to point out that on the *Validation* set the missing values are filled in regarding the mode obtained on the *Train* dataset.

Afterwards, the outlier removal process begins. The first step was to test removing only incoherent values, this meant manually filtering out impossible blood pressure, cholesterol and birth year values. In this phase, less than 3% of the data was removed on either dataset. Then, we proceeded to test the LocalOutlierFactor, which is an algorithm that detects outliers by measuring the local deviation of a given data point with respect to its neighbors. The algorithm removed a little over 4% of data in both the *Train* and *Validation* datasets. The final outlier removal approach tested was the IQR technique, which considers as outliers any values outside of the quarter 1 and 3 fence. This was the worst performing method since it removed over 7% of the total data available. Finally, to improve the outlier removal, we assessed how the combination of the methods would perform. The LocalOutlierFactor combined with the Manual Filters method removed the same percentage as the LocalOutlierFactor alone, but this combination of approaches ensured that not only extreme values were being dropped, but also the impossible values. Regarding the IQR plus Manual Filters technique, the percentage of data kept was the same as the IQR method by itself, so this approach was discarded. Having that stated, the LocalOutlierFactor and Manual Filters combo was the one chosen to remove outliers. By doing so, 95,89% of the *Train* data and 95,83% of the *Validation* data were kept. The same process was applied on the *Test* set, removing less than 4% of its total data.

Feature Selection and Engineering

Concerning the Feature Selection and Engineering, the focus was on the manipulation of the variables. This includes the deletion, addition or mutation of the features in order to achieve better performance and greater accuracy.

The first step was to create six new datasets splitting the features of *Train*, *Validation* and *Test* into categorical and numerical in order to be able to work with the features in a more efficient and organized manner. Then, we proceeded to standardize the data using MinMaxScaler which is an estimator that scales and translates each metric feature individually such that it is in the given range of the training set. Note that for each of the datasets the same scaler was used, having the standardization based solely on the Train input. Moreover, the target variable for Train and Validation was split from the predictors variables in order to perform analysis on them.

Moving on to the feature selection, initially the variance for each numeric variable on the Train set was checked, if there were a variable with variance equal to 0 it would be excluded, however that was not the case. Then, we proceeded to measure the degree of association between independent and dependent variables using the Spearman Correlation method which, as shown in the correlation matrix heatmap in Figure 7, did not present any high correlations, meaning that by following this approach all variables would be dropped.

Continuing the feature selection, we focused on the analysis of categorical data by assessing the Chi-Square independent test that determines if two variables are independent or related to one another. The objective was to exclude from the datasets whichever features' distribution depended on the level of the other, which was the case of the *Region, Education, Smoking_Habit,* and *Water_Habit* variables. Therefore, they should be removed from the datasets before proceeding with the modeling. Then a function to visualize the proportion of the dependent variable in each possible value of a categorical feature was created in order to analyze the distribution of patients with and without the disease for each category. Visualizations are presented from Figures 8 to 12 in which it was noticeable that patients with high frequency of having the disease fall in the following categories: “consume alcohol everyday”, “no exercise”, “less than 1 piece of fruit a day”, “had diabetes” and “had a checkup 3 years ago”.

Furthermore, now regarding metric features, two more methods were implemented to better understand which variables should be kept or discarded: Recursive Feature Elimination (RFE) and Lasso Regression. The first one fits a given model and removes the weakest features until the specified number of variables

is reached by eliminating dependencies and collinearity that may exist in the model. To find the optimal number of features to keep, we applied 3 different models using cross validation to score different feature subsets and select the best collection of features. The models applied were Logistic Regression with a 71% score with 4 features, Random Forest presenting a 96% score with 5 features and Decision Tree with 91% score with 4 features. Considering the scores obtained in each cross-validated model, the final number of selected features was ambiguous, with either 4 or 5 variables being selected, namely: *Birth_Year, High_Cholesterol, Mental_Health,* and *Physical_Health, while the Blood_Pressure was not a certain pick.* Regarding the Lasso Regression approach, although it is not optimal for nonlinear relationships between dependent and independent features - which is the case of our data - we deemed it would be useful to have a better understanding of the metric features. It discarded the *Birth_Year,* whilst *Weight*, *Height*, *High_Cholesterol* and *Blood_Pressure* also seemed quite insignificant for the model.

Having done the feature selection analysis, we created two tables to better visualize the insights provided by the methods previously described: Table 1 with the metric features and Table 2 describing the categorical feature selection. On the first table, the final predictors selected depended on the most responses, meaning that whenever two of the methods indicated it was better to discard it, the feature got eliminated and vice versa. Regarding the Lasso Regression method, whenever we were unsure if the variable should be kept and the RFE technique stated that it should, it got included in the 'Try with and without' category. Later, four trial datasets were created for each set of data to store all possible combinations for the 'Try with and without' category of variables. Each dataset kept the *Drinking_Habit, Fruit_Habit, Checkup, Diabetes, Exercise, Mental_Health* and *Physical_Health* features differing solely on whether they stored Blood_Pressure and High_Cholesterol. Following is the final combination of features for each of the trial datasets:

- Trial 1: *Drinking_Habit, Fruit_Habit, Checkup, Diabetes, Exercise, Mental_Health, Physical_Health, High_Cholesterol*
- Trial 2: *Drinking_Habit, Fruit_Habit, Checkup, Diabetes, Exercise, Mental_Health, Physical_Health, Blood_Pressure*
- Trial 3: *Drinking_Habit, Fruit_Habit, Checkup, Diabetes, Exercise, Mental_Health, Physical_Health*
- Trial 4: *Drinking_Habit, Fruit_Habit, Checkup, Diabetes, Exercise, Mental_Health, Physical_Health, Blood_Pressure, High_Cholesterol*

Afterwards, the selected features were split between metric and non-metric so that the chosen categorical variables would go through the One-Hot Encoding, which is a way to convert categorical information into

a format that can be fed into the machine learning algorithms to improve prediction accuracy. Finally, to create the final datasets that will be used during the model development, the scaled selected metric features were joined with the one-hot encoded categorical variables for *Train*, *Test* and *Validation* for each of the trial sets.

Putting the Data Preprocessing together comes the last step: redoing the data exploration to ensure that the final datasets are clean and optimized for modeling. In this part, the Profile Report, Description method and visualizations were reassessed on trial 4, the one containing the combination of all the features previously selected. The visuals for this step are presented from Figure 13 to Figure 16.

Modeling

After the datasets have been cleaned and optimized, the modeling step begins. The first step was to identify possible algorithms that would perform well based on the available data. In order to do so, a function that fits the data into a range of models and assesses its results based on Accuracy, Precision, Recall, F1 score, and Area under the Roc Curve was created. This function was fed with each of the trial sets previously created and we identified that overall, the best performing models to predict if the patient is carrying the disease or not, were: Random Forests, Decision Trees, and Gradient Boosting. Therefore, those were the three algorithms selected to train the classification model. Regarding the final selection of features combos, each of the trial lots were fed into the models' training and after assessing their scores the Trial1 set was selected as the definitive dataset.

Before training the chosen algorithms, a function that uses the grid search technique for hyperparameter tuning, named *param_tuning,* was created. It divides the domain of the parameters using a discrete grid, then fits the X and Y train data for every combination of values on the grid while calculating the average F1 score obtained using stratified k-fold cross validation. This would ensure the best parameters for each of the models were picked during the model development.

Gradient Boosting

For the Gradient Boosting algorithm, the *param_tuning* function was fed with options for:

1. The number of boosting stages to perform, since the algorithm is quite powerful to overfitting a bigger number usually leads to better performance.
2. The maximum depth of the individual regression estimators, which limits the number of nodes in the tree, and the best value relies on the interaction of the input variables.
3. The learning rate, to better reduce the contribution of each tree

According to the function, the best parameters combination is a learning rate of 1, maximum depth of 3 and 50 boosting stages, achieving a weighted f1 score of 94%. These parameters were fed into the Gradient Boosting Classifier that fit X and Y train data.

Decision Tree

Some of the parameters used in the hyper parameterization of the Decision Tree algorithm were:

1. The criterion, which measures the quality of a split based on either Gini, Entropy or Log loss.
2. The splitter that defines the strategy used to split each node.
3. The maximum depth of the tree.
4. The lowest number of samples required to slice an internal node.
5. The least possible number of samples needed to be at a leaf node.
6. The number of features to consider when searching for the best split.

Considering the fine-tuned undergone by the parameters, this was the final model achieving 80% F1 score: criterion as Gini, maximum depth of 30, maximum features of 0.5, maximum leaf nodes of 5, minimum impurity decrease of 0, minimum sample leaves equal to 1, minimum sample split of 2 and splitter as best method. Then, those parameters were fed to the Decision Tree Classifier in order to fit the available data.

Random Forest

To fine-tune the Random Forest algorithm, the following parameters were fed into the *param_tuning* function:

1. Bootstrap, to select whether only samples are used when building trees.
2. The weights associated with classes.
3. The criterion to measure the quality of a split based.
4. The maximum depth of the tree.
5. The number of features to consider when searching for the best split.
6. The OOB score, to select whether to use out-of-bag samples to assess the generalization score.

The final parameters selected by the *param_tuning* function were to select samples when building the tree, the weights equal to balanced to use the values of y to adjust the weights inversely proportional to class frequencies as input data, the quality measure based on Gini, no maximum depth, maximum features set to log2, and OOB score equal to True. With these parameters the F1 score obtained was 94%.

Assessment

In order to evaluate the trained models, the *classification* function that predicts the Y value and returns a classification report with the Recall, F1 score, and Accuracy was created. By inputting the models previously created into the function, it was possible to analyze their performance and select the Random Forest as the final model to predict whether a patient will suffer or not from the Smith's Disease.

The F1 scores achieved by the Random Forest, Decision Trees, and Gradient Boosting were of 99%, 80%, and 98%, respectively. The chosen model also performed best regarding the Accuracy, Precision and Recall scoring.

Conclusion

In summary, during the execution of the project we developed three different Machine Learning models and trained them using 8 features related to patients' health, sociodemographic and habits information. We validated and compared the algorithms using stratified four-fold cross-validation, fine-tuned their parameters and assessed their performance using the Accuracy, Precision, Recall, and F1 scores and found that the best performing model to predict the dependent variable was Random Forest, achieving scores of 99%, 99%, 96% and 99% respectively. Putting it all together, the selected model was used to train the whole train dataset in order to predict the test data presented so the labels could be uploaded to Kaggle, generating a final F-1 score of 1.

References

Recursive Feature Elimination. Yellowbrick. Retrieved November 23, 2022, from https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html

Sklearn.Ensemble.GradientBoostingClassifier. Scikit-Learn. Retrieved December 18, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

Sklearn.Ensemble.RandomForestClassifier. Scikit-Learn. Retrieved December 18, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Sklearn.Model_Selection.StratifiedKFold. Scikit-Learn. Retrieved December 10, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html#sklearn.model_selection.StratifiedKFold

Sklearn.Preprocessing.MinMaxScaler. Scikit-Learn. Retrieved November 21, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Sklearn.Tree.DecisionTreeClassifier. Scikit-Learn. Retrieved December 18, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn-tree-decisiontreeclassifier

Stobierski, T. (2021, January 19). Data Wrangling: What It is & Why It is Important. Harvard Business School Online. Retrieved November 19, 2022, from https://online.hbs.edu/blog/post/data-wrangling

The University of Texas at Austin (n.d). Chi-Square Test of Independence. Statistics Online Support. Retrieved November 26, 2022, from http://sites.utexas.edu/sos/guided/inferential/categorical/chi2/

Annexes

Figure 1 - Absolute Frequency Chart for *Validation* Dataset

Figure 2 – Absolute Frequency Chart for *Train* Dataset

Figure 3 – BoxPlot for *Validation* Dataset

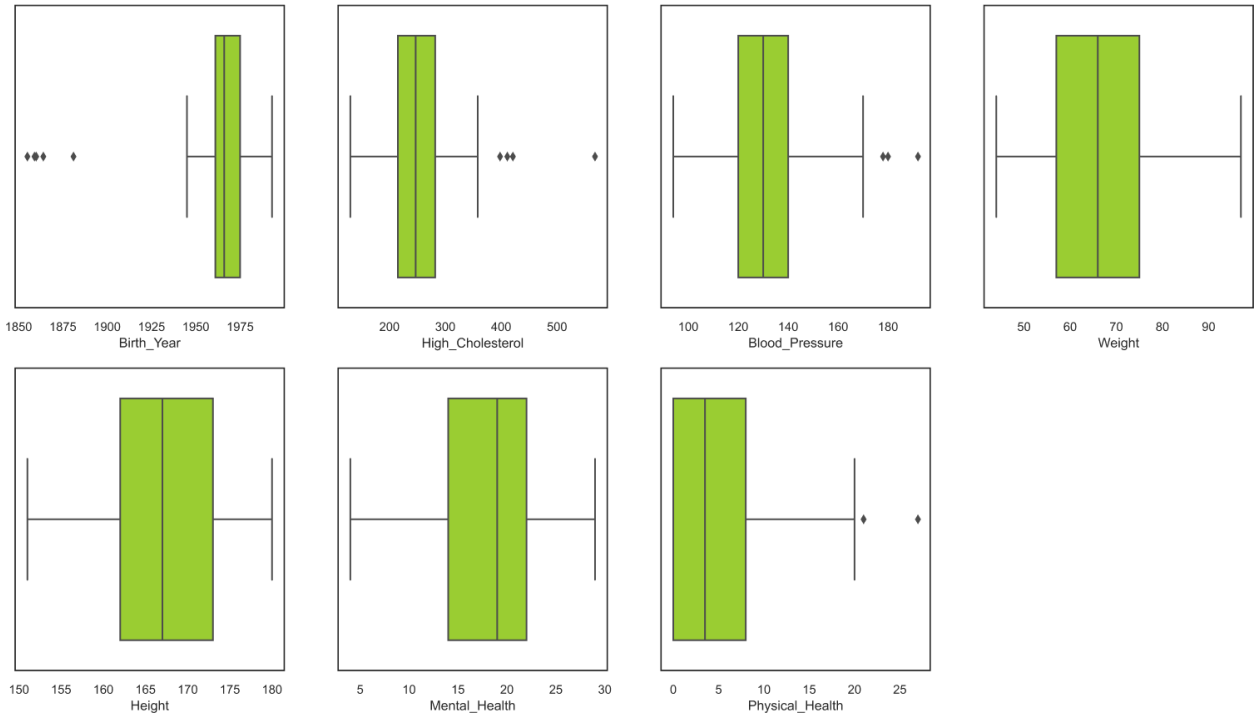

Figure 4 – Boxplot for *Train* Dataset

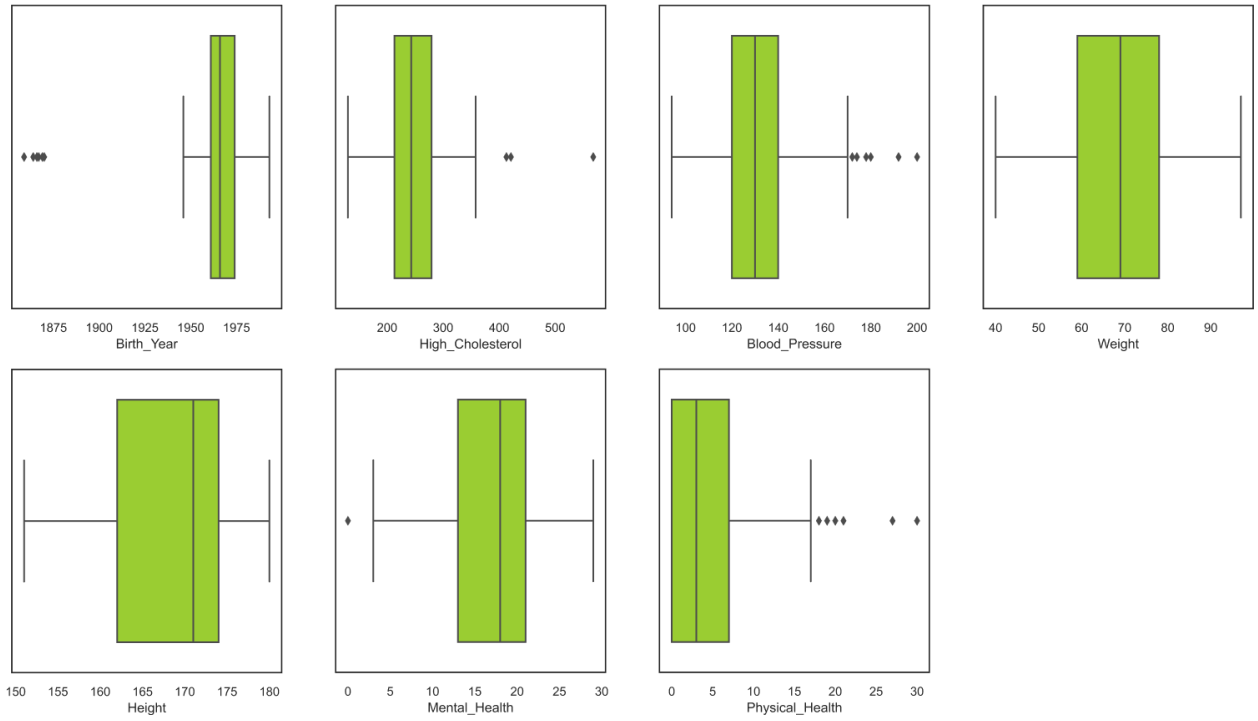

Figure 5 – Pairwise for *Validation* Dataset

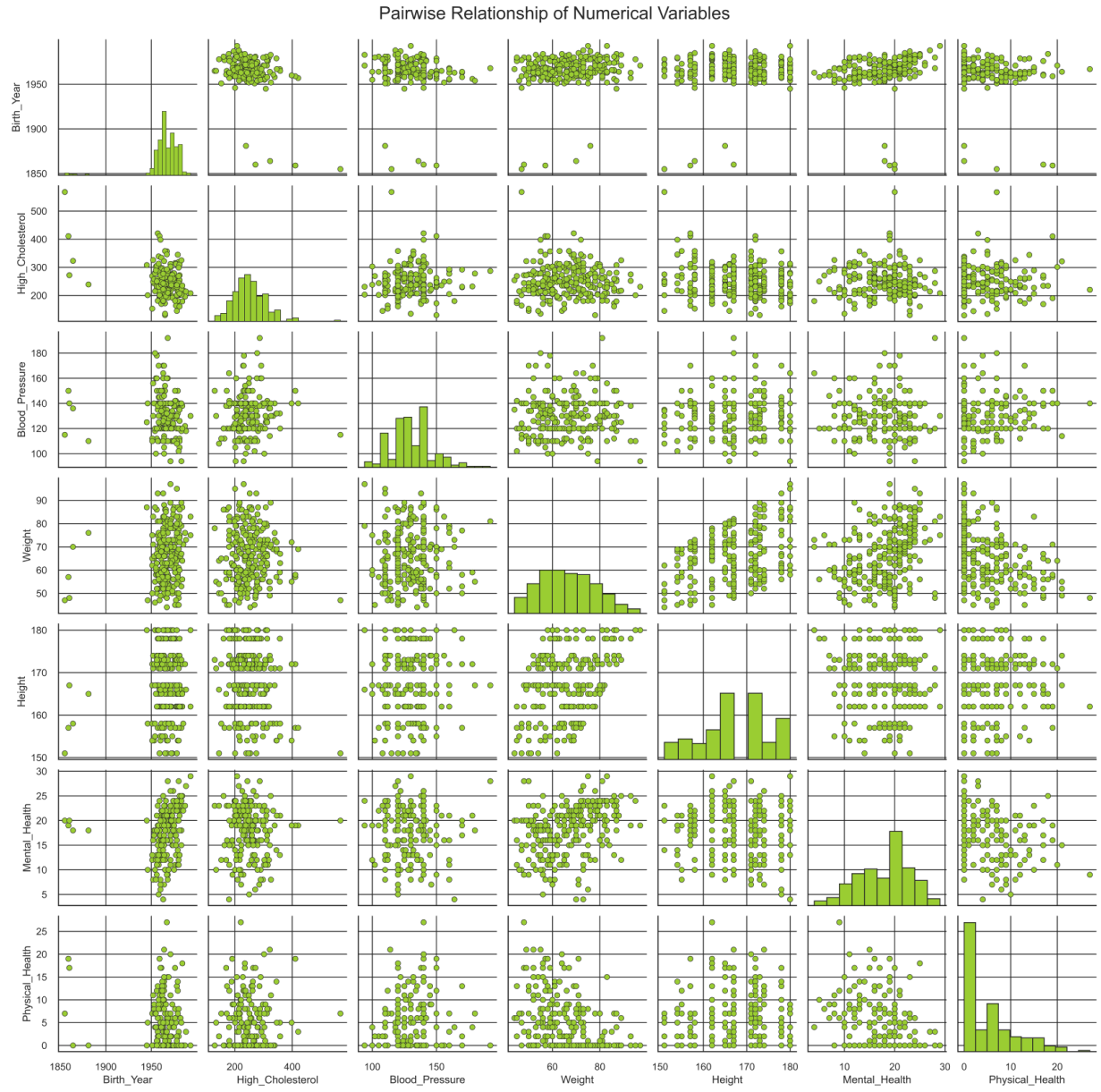

Figure 6 – Pairwise for *Train* Dataset

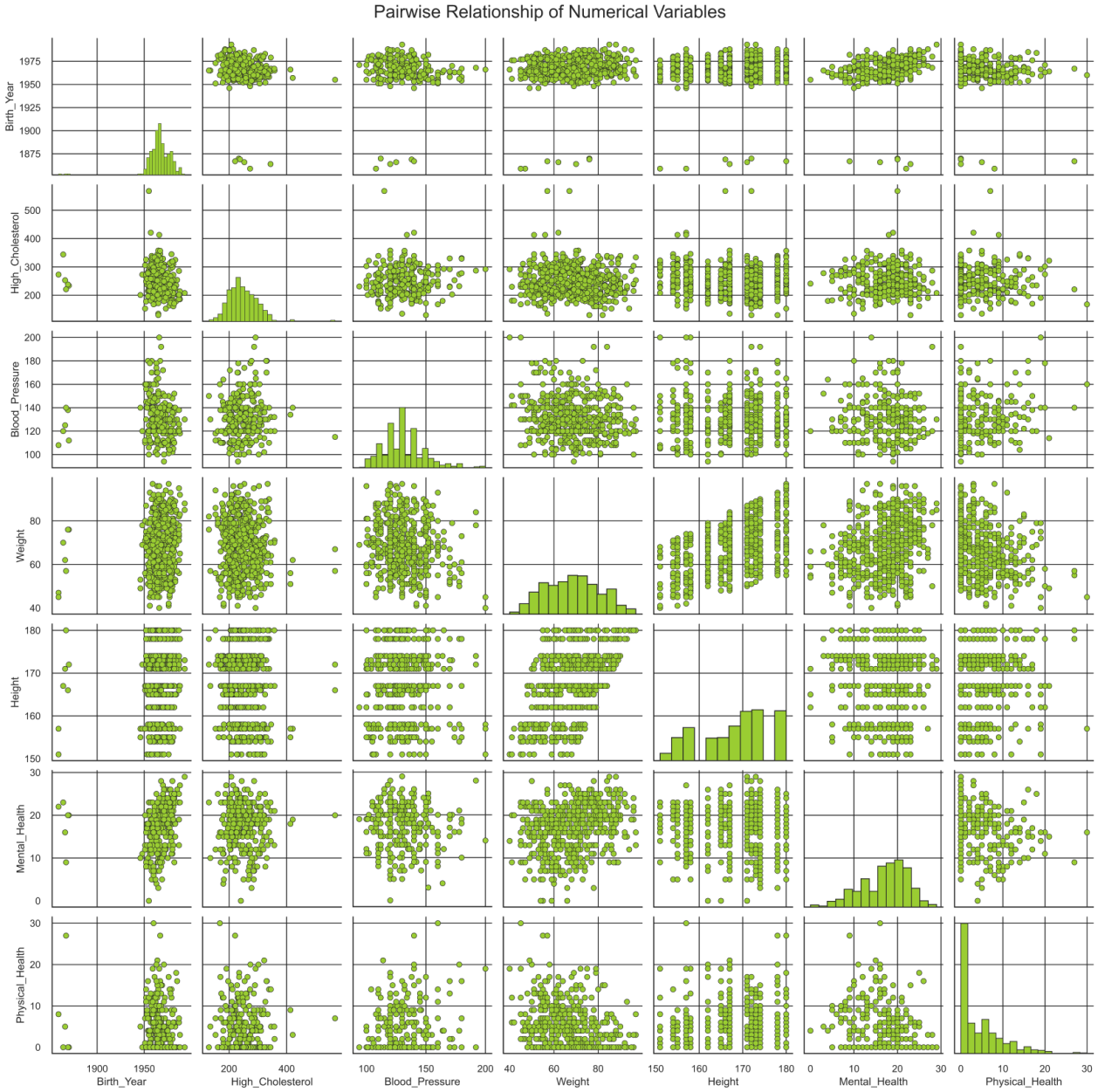

Figure 7 – Spearman Correlation Heatmap for *Train* Dataset

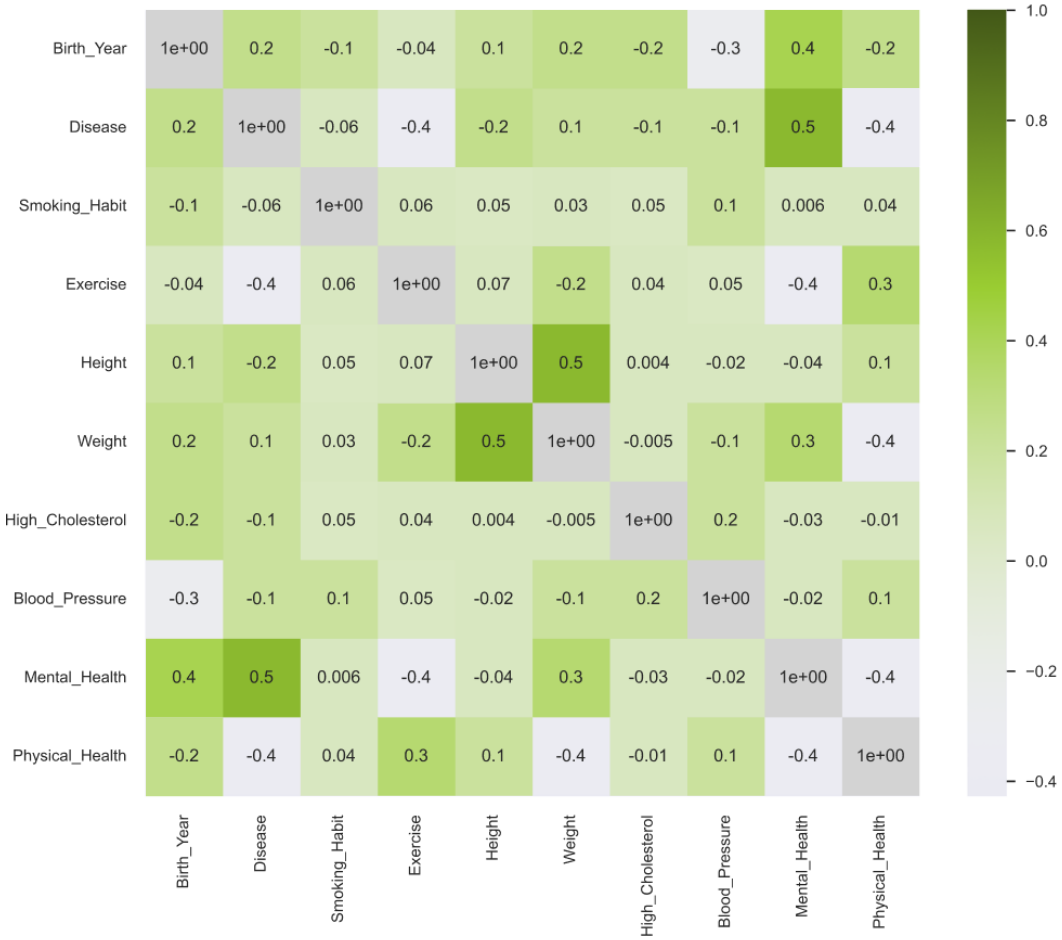

Figure 8 – Frequency and Proportion Distribution of Dependent Variable Chats for *Fruit_Habit*

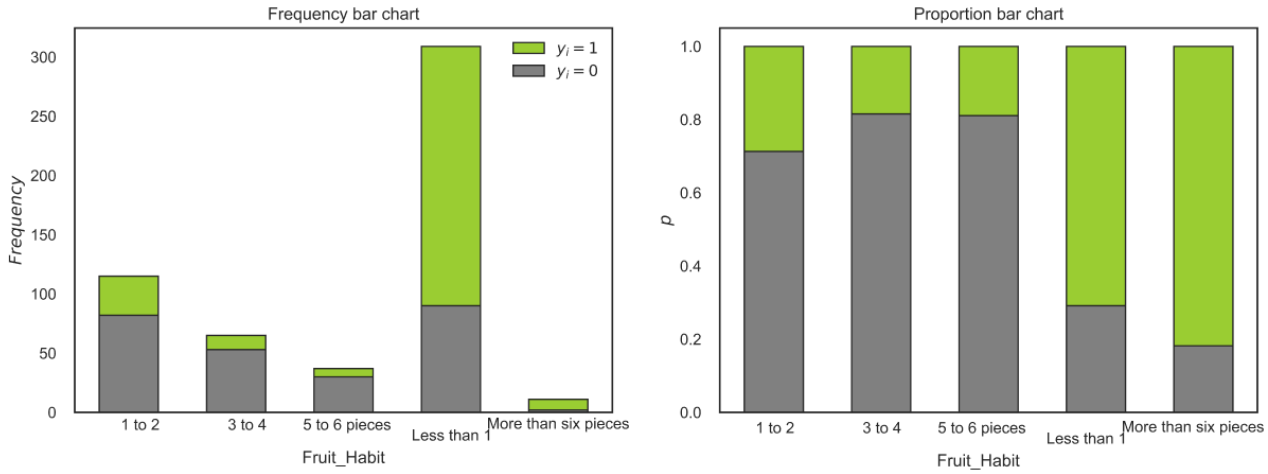

Figure 9 – Frequency and Proportion Distribution of Dependent Variable Chats for *Exercise*

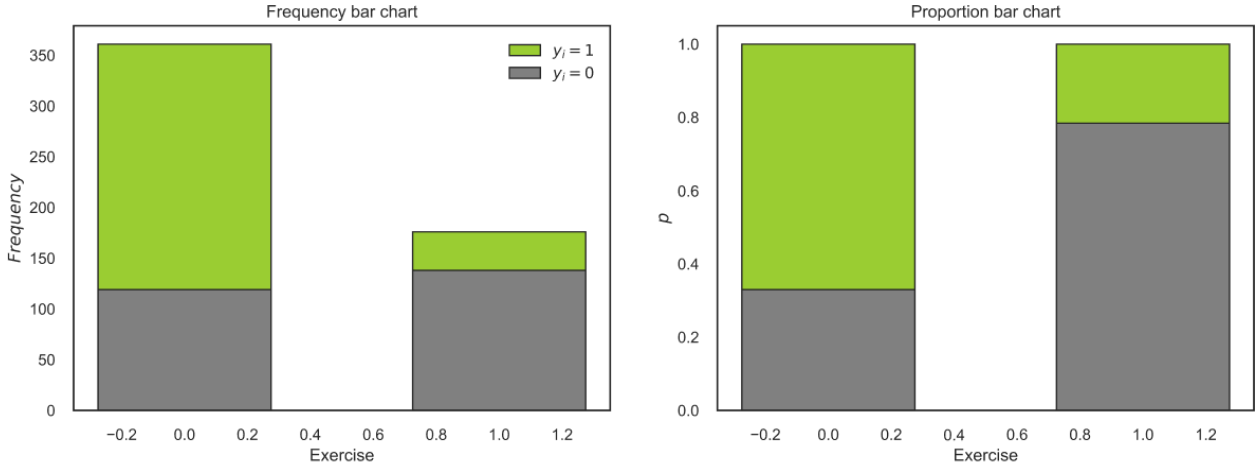

Figure 10 – Frequency and Proportion Distribution of Dependent Variable Chats for *Drinking_Habit*

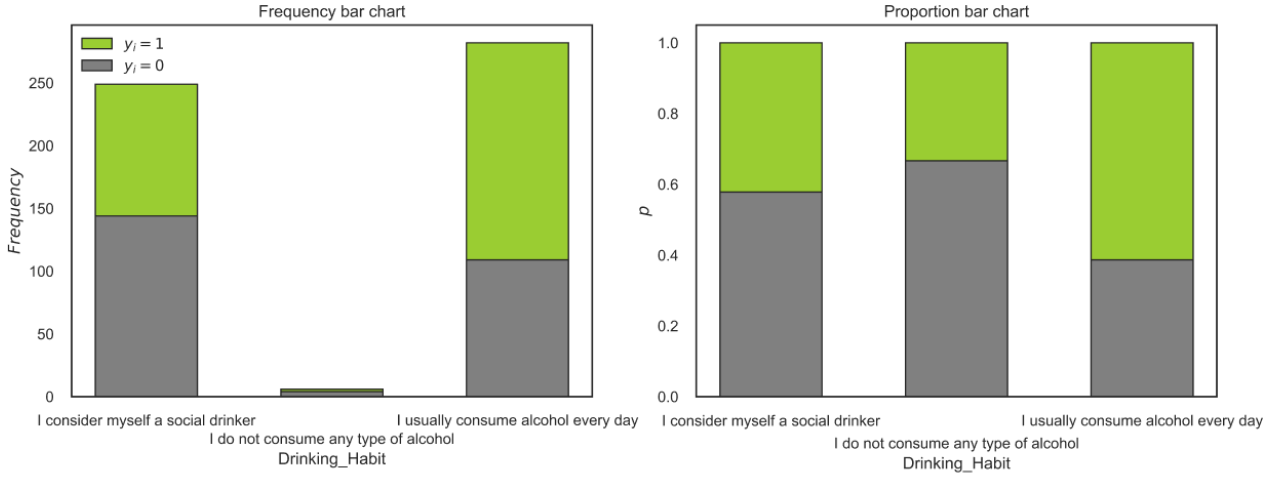

Figure 11 – Frequency and Proportion Distribution of Dependent Variable Chats for *Diabetes*

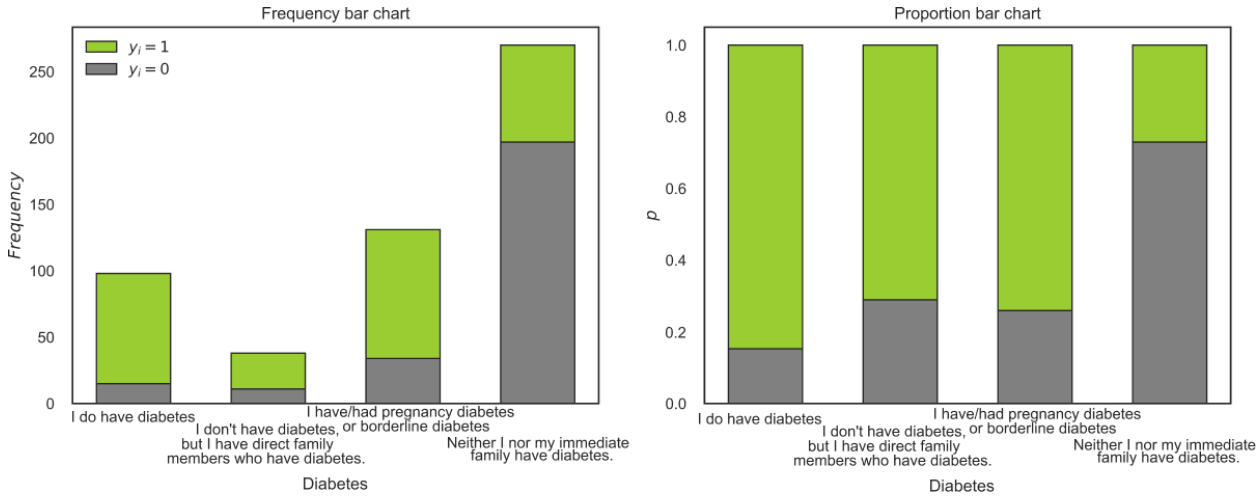

Figure 12 – Frequency and Proportion Distribution of Dependent Variable Chats for *Checkup*

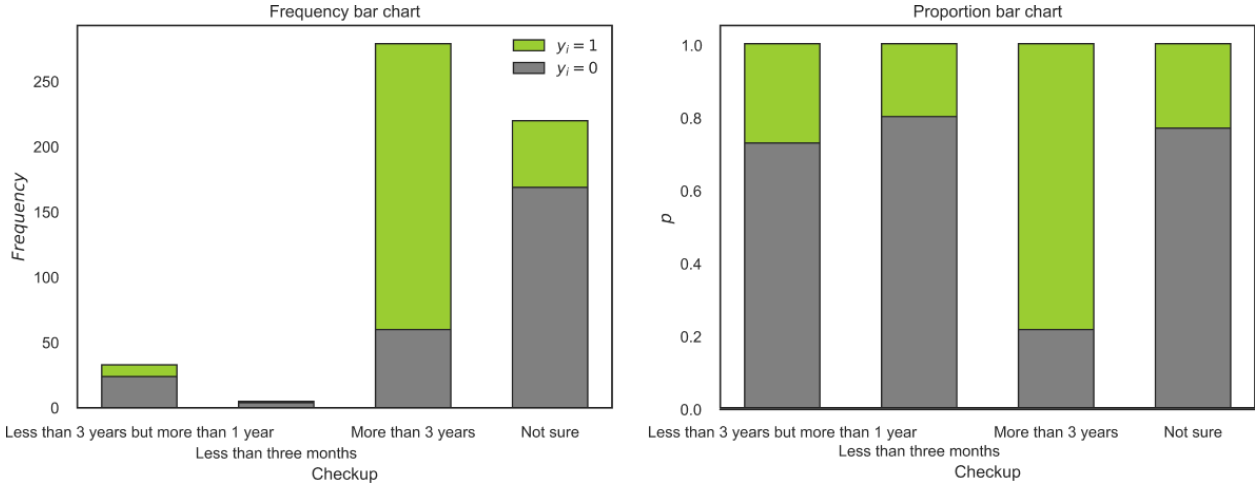

Figure 13 – Absolut Frequency Chart for *Validation Trial4* Dataset

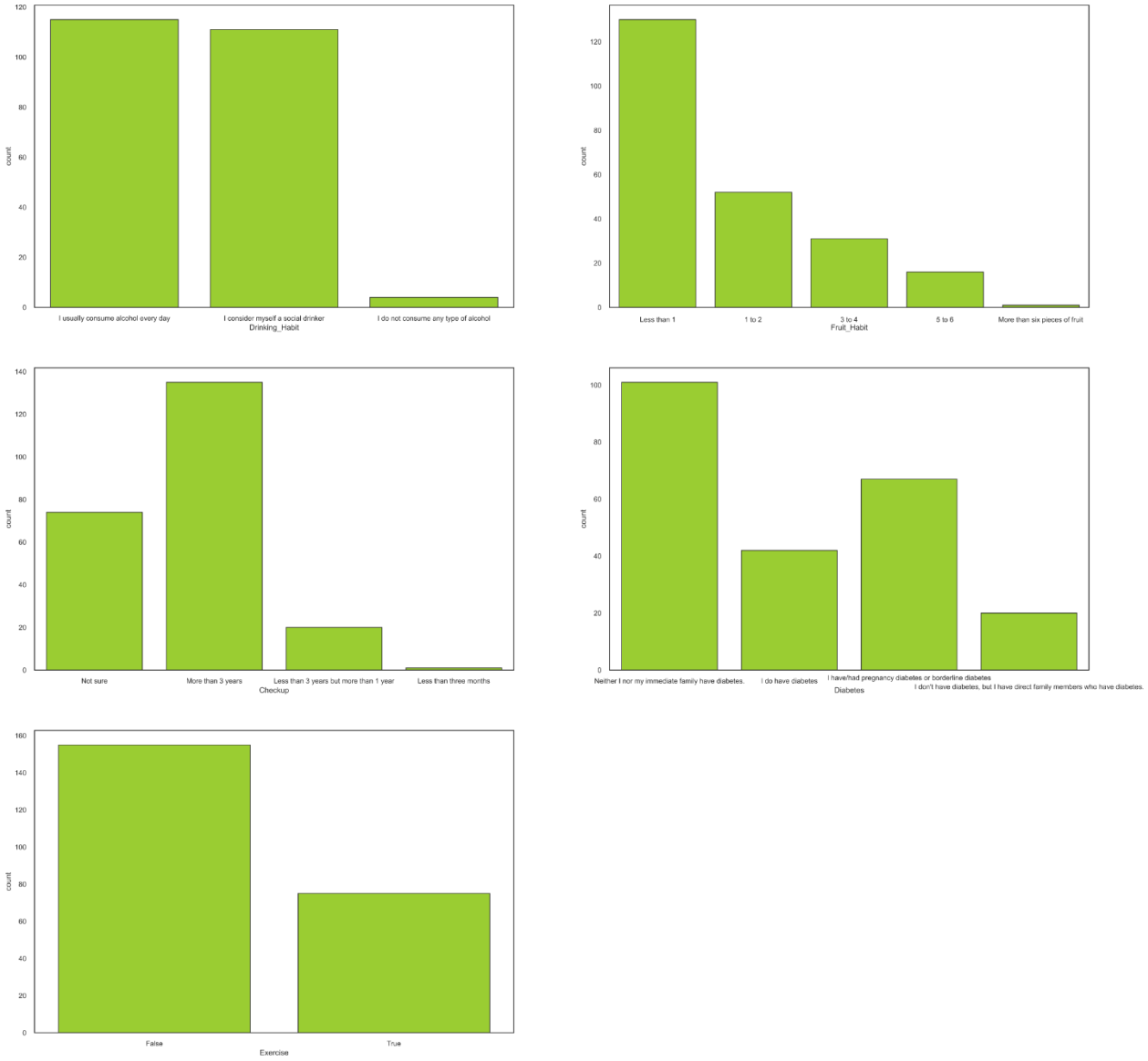

Figure 14 – Absolut Frequency Chart for *Train Trial4* Dataset

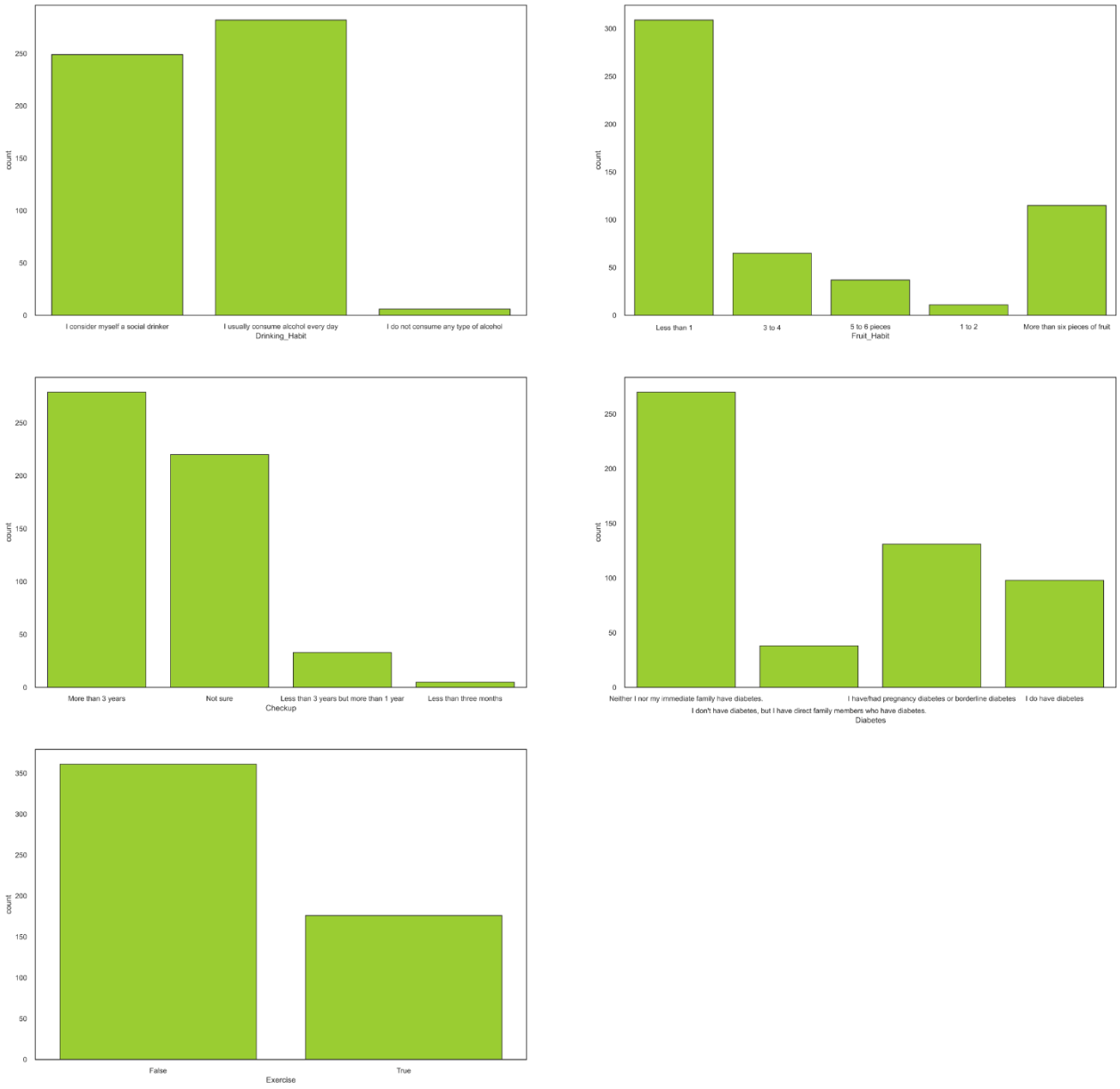

Figure 15 – BoxPlot for *Validation Trial4* Dataset

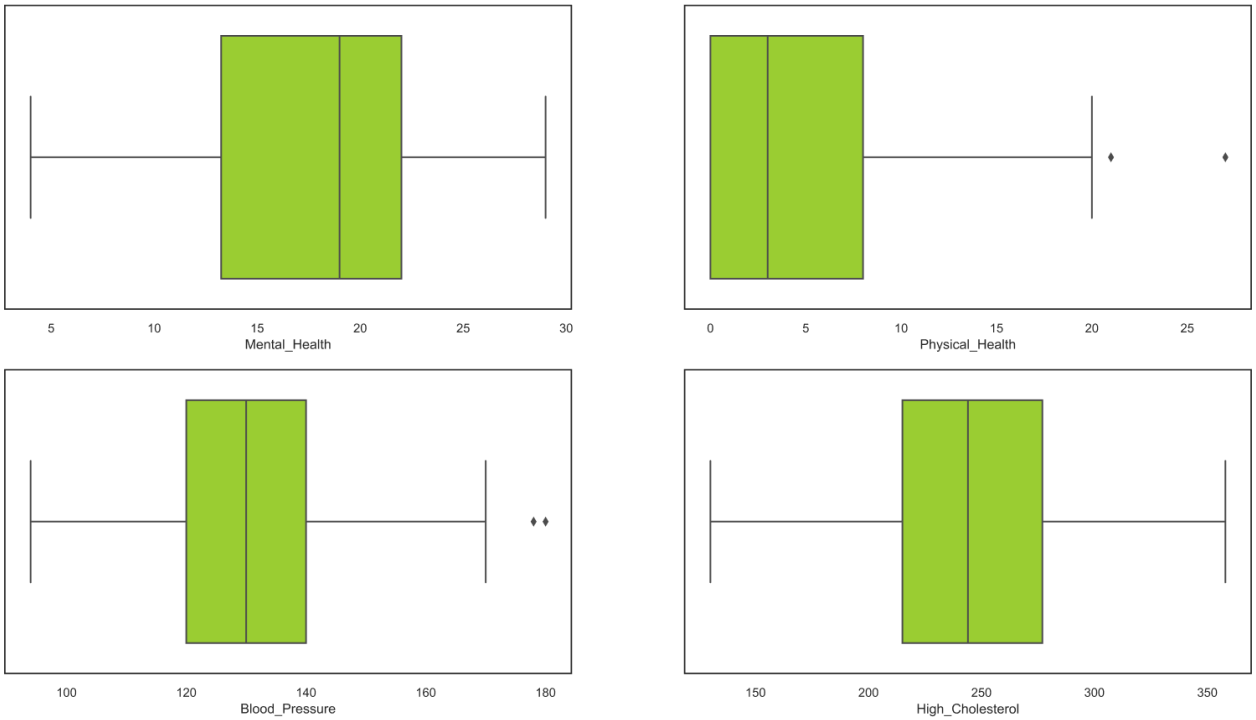

Figure 16 – BoxPlot for *Train Trial4* Dataset