# 

# 

# CSE422 Lab Project Report

| **Group No: 05 CSE422 Lab Section 09**  **Fall 23** |
| --- |

| **ID** | **NAME** |
| --- | --- |
| Zabir Ramiz | 21201394 |
| Sajida Hossain | 21201603 |
| Smita Biswas | 21201193 |
| Masiat Mohammad Momshad | 21241017 |

### 

### 

### 

### 

### **Table of contents:**

| Serial No. | Content | Page No. |
| --- | --- | --- |
| 1 | Introduction | 01 |
| 2 | Dataset description | 02-07 |
| 3 | Dataset pre-processing | 08-12 |
| 4 | Feature scaling | 13 |
| 5 | Dataset splitting | 13 |
| 6 | Model training and testing | 13-14 |
| 7 | Model selection/Comparison analysis | 15-21 |
| 8 | Conclusion | 21-22 |

## **Introduction:**

This report presents a machine learning project focused on predicting cardiovascular diseases among patients using medical data collected from different tests. Cardiovascular diseases are a significant concern in healthcare. Thus, the project aims to develop a model that can assist healthcare professionals in early detection and intervention by analyzing the medical records of a potential patient.

## **Dataset Description:**

**Source:**

* Link: <https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset/data>
* Reference: Ulianova, S. (2019). *Cardiovascular Disease dataset* [Data set].

**Dataset Description:**

* The dataset has a total of 13 features

1. id:

- A unique identifier for each record or individual in the dataset.

- Data Type: Integer.

2. age:

- The age of the individual. In our dataset, it is in days.

- Data Type: Integer.

3. gender:

- The gender of the individual. In our dataset, it is encoded as 1 for female, 2 for male

- Data Type: Integer .

4. height:

- The height of the individual, in centimeters.

- Data Type: Integer.

5. weight:

- The body weight of the individual, in kilograms.

- Data Type: Float.

6. ap\_hi:

- Systolic blood pressure. This is the higher number in a blood pressure reading.

- Data Type: Integer.

7. ap\_lo:

- Diastolic blood pressure. This is the lower number in a blood pressure reading

- Data Type: Integer.

8. cholesterol:

- A measure of cholesterol levels, possibly encoded categorically 1 for normal, 2 for above normal, 3 for well above normal.

- Data Type: Integer.

9. gluc:

- A measure of glucose levels, which might also be encoded categorically similar to cholesterol.

- Data Type: Integer.

10. smoke:

- Indicates whether the individual smokes tobacco. 0 for no, 1 for yes.

- Data Type: Integer.

11. alco:

- Indicates alcohol consumption. 0 for no, 1 for yes.

- Data Type: Integer.

12. active:

- Indicates whether the individual is physically active. 0 for no, 1 for yes.

- Data Type: Integer.

13. cardio:

- Indicates the presence of cardiovascular disease, the target variable for analysis or prediction. 0 for absence, 1 for presence.

- Data Type: Integer.

Problem Type:

* As the target is to detect the presence or absence of cardiovascular disease it is a classification problem.

Data Points:

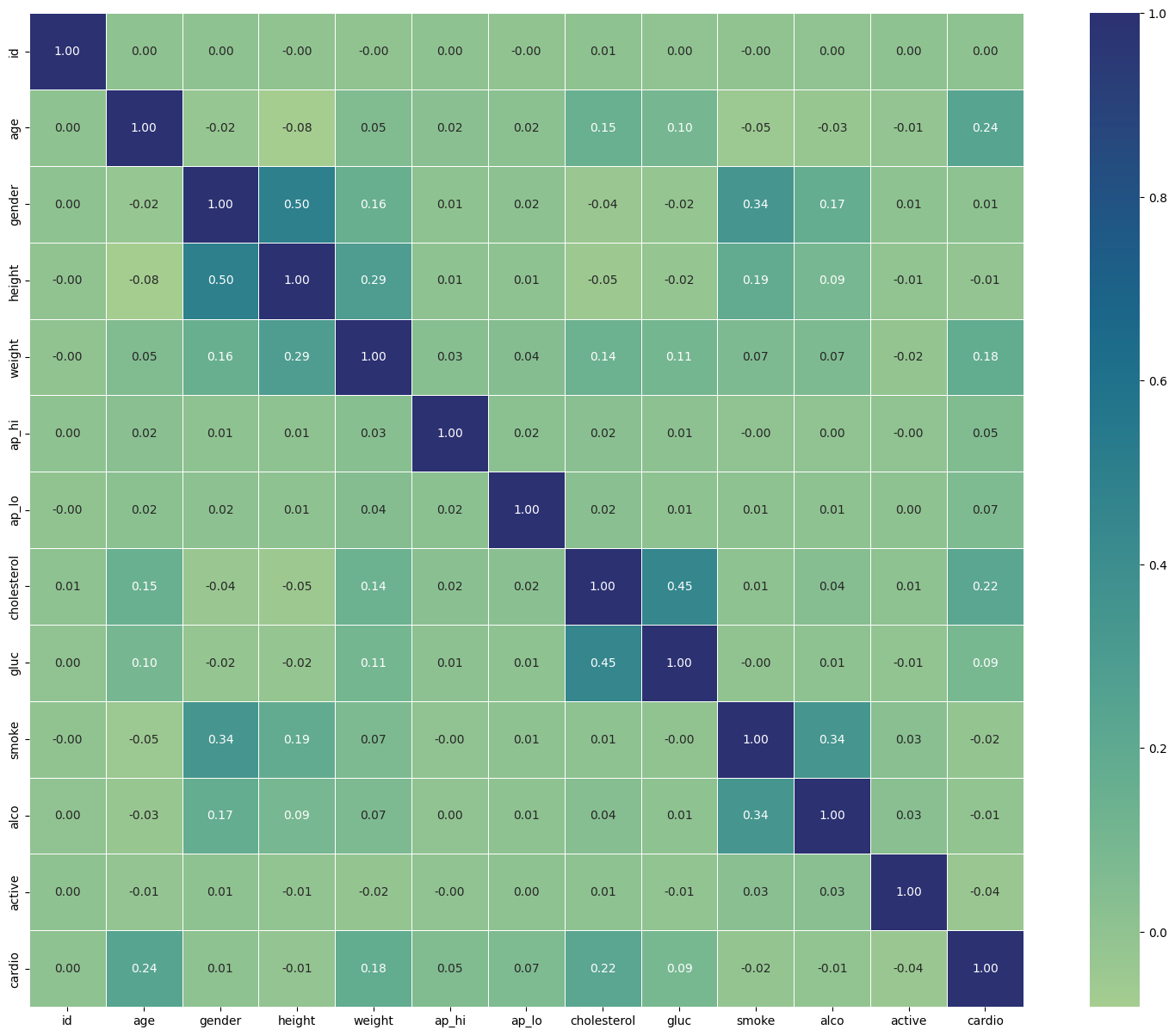
* The dataset has 70,000 data points

Feature Type:

* 7 categorical features: ['gender', 'cholesterol', 'gluc', 'smoke', 'alco', 'active', 'cardio']
* 6 quantitative features: ['id', 'age', 'height', 'weight', 'ap\_hi', 'ap\_lo']

Correlation of Features:

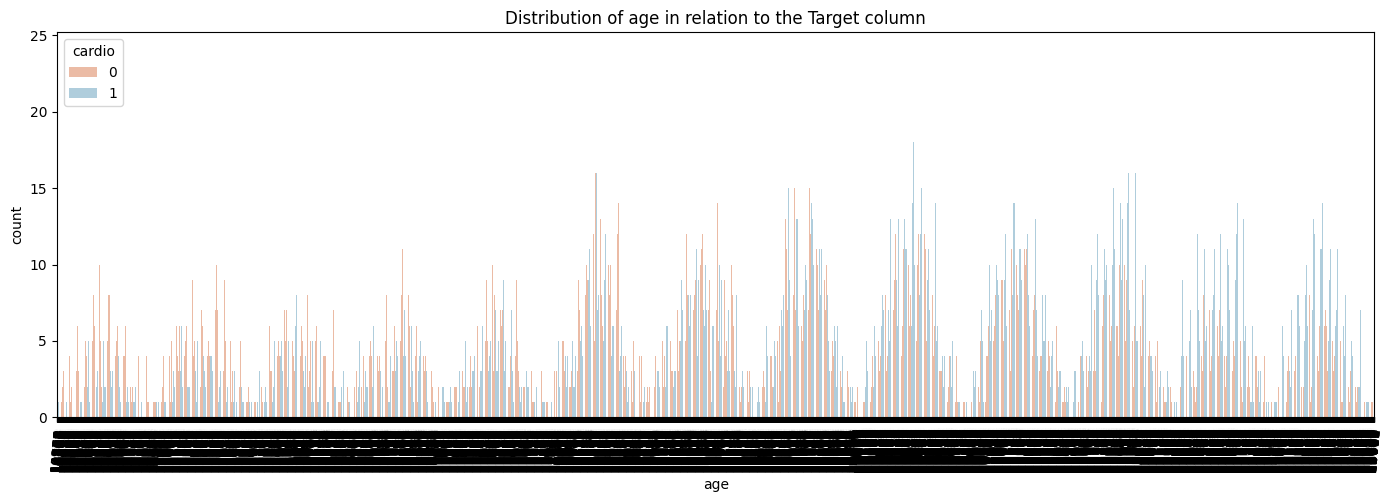
* Heatmap generated by seaborn library annotating with corresponding correlation till 2 floating points

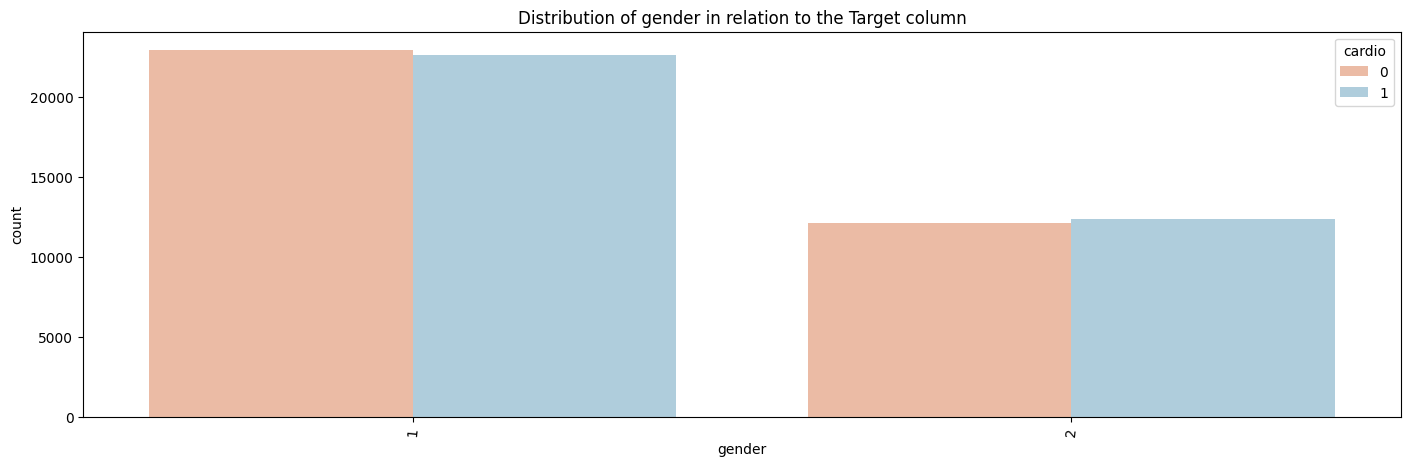


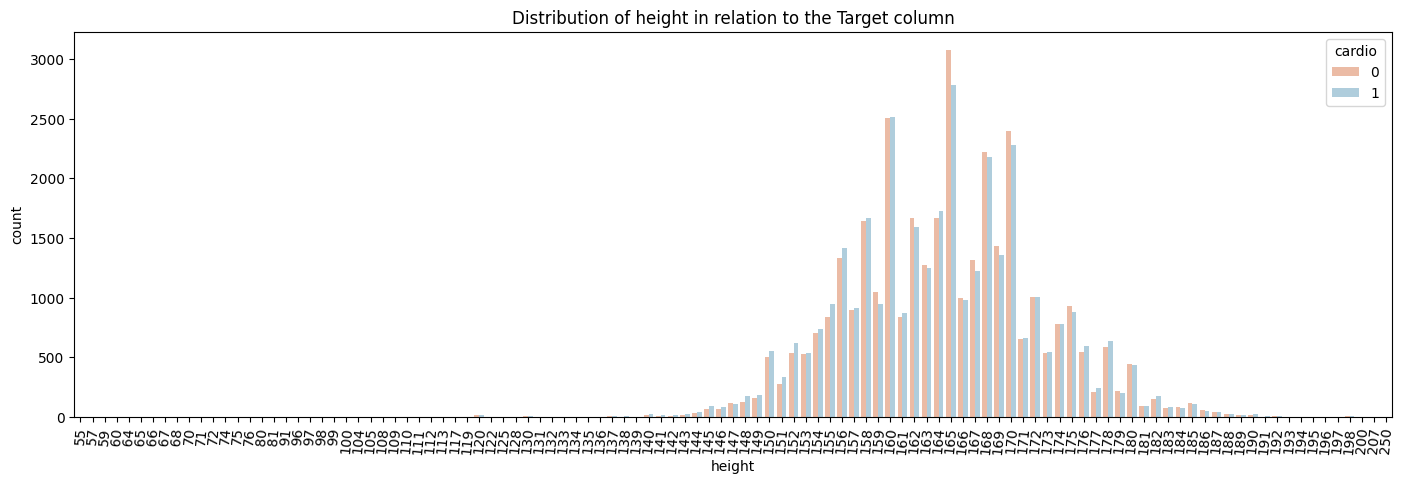
**Dataset Balance:**

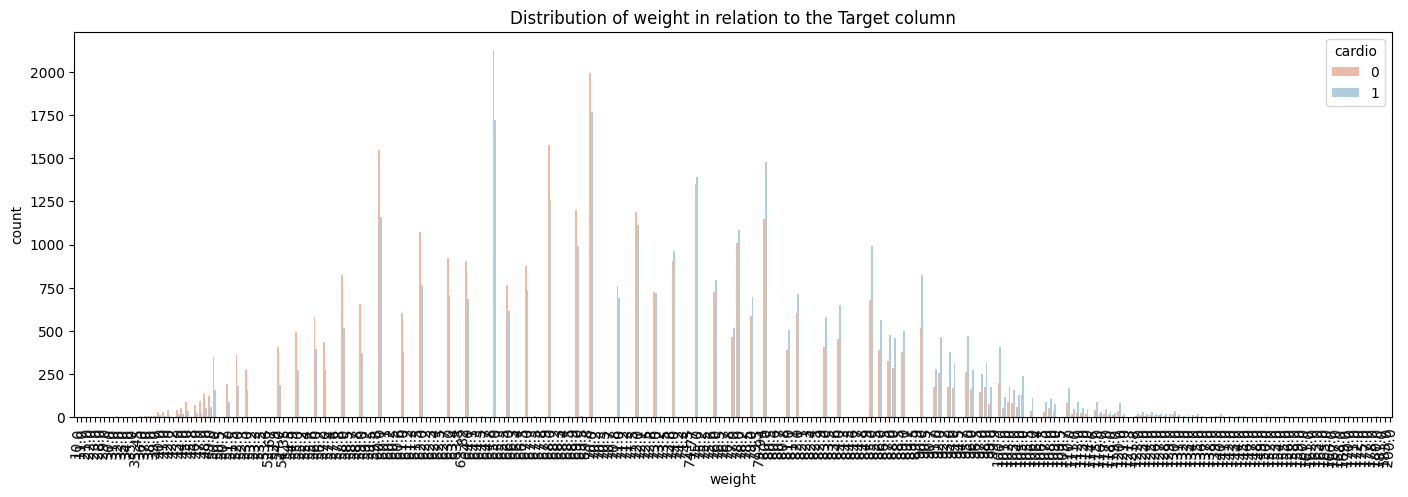
* For the output feature, most of the unique columns are evenly distributed in relation to the target column, Stroke.
* Below is the representation of the 13 columns in barchart

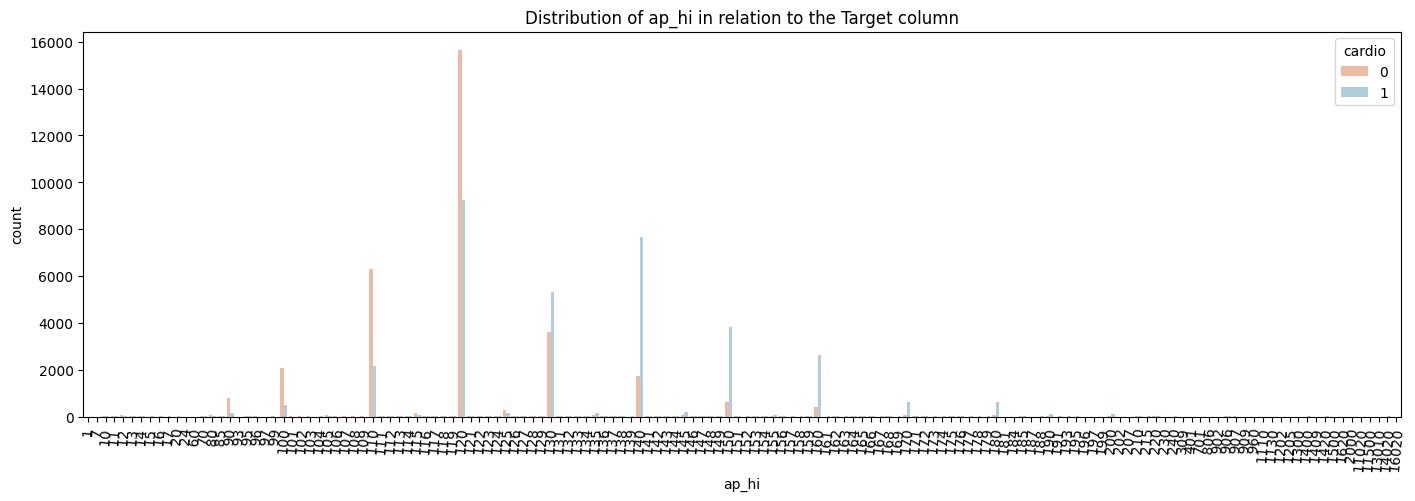


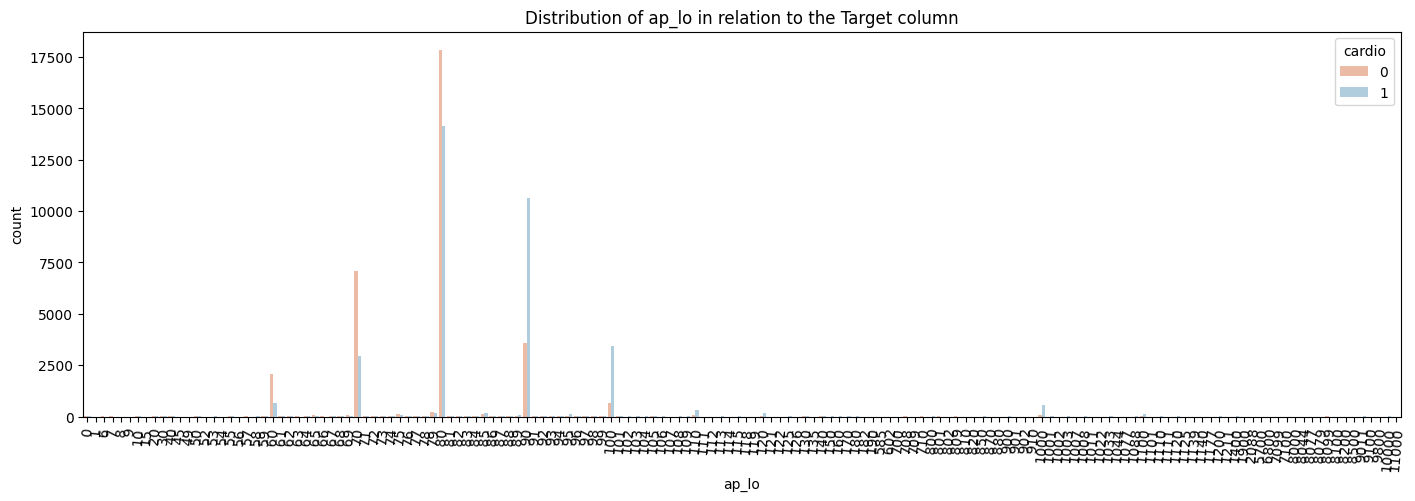


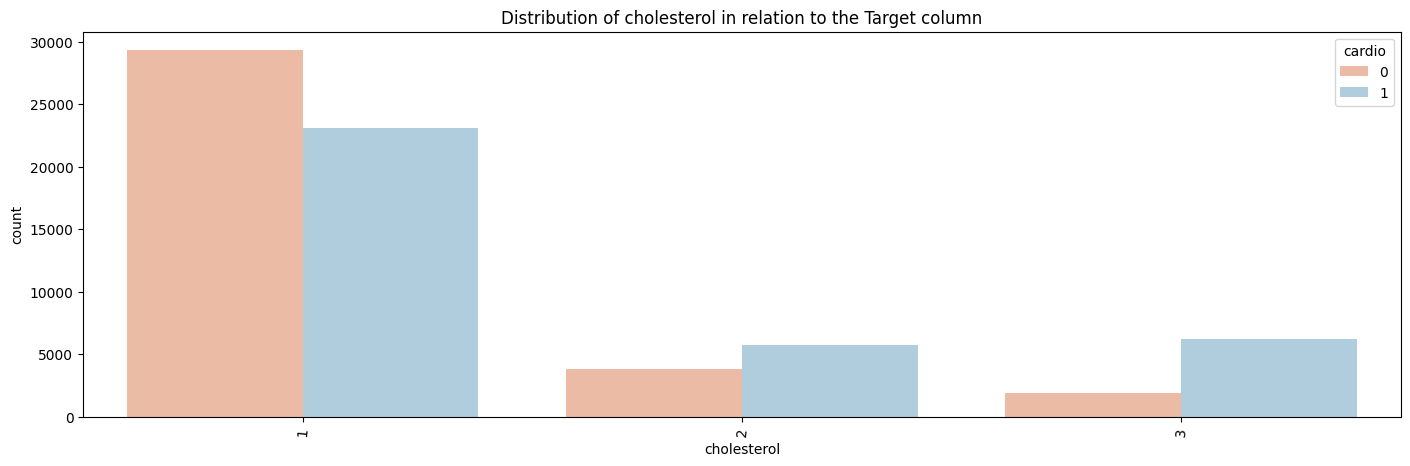


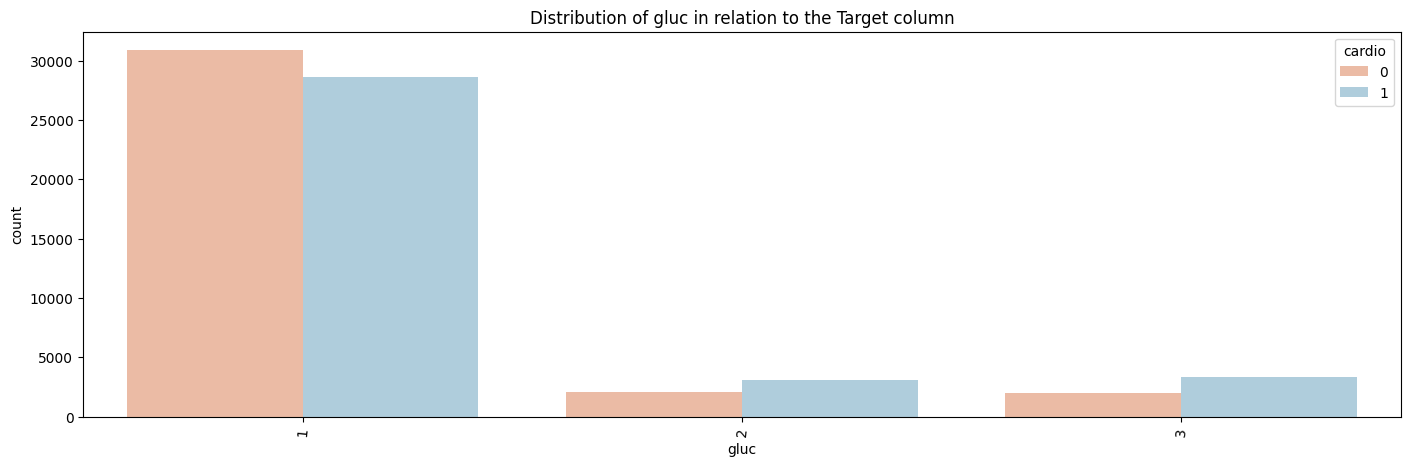


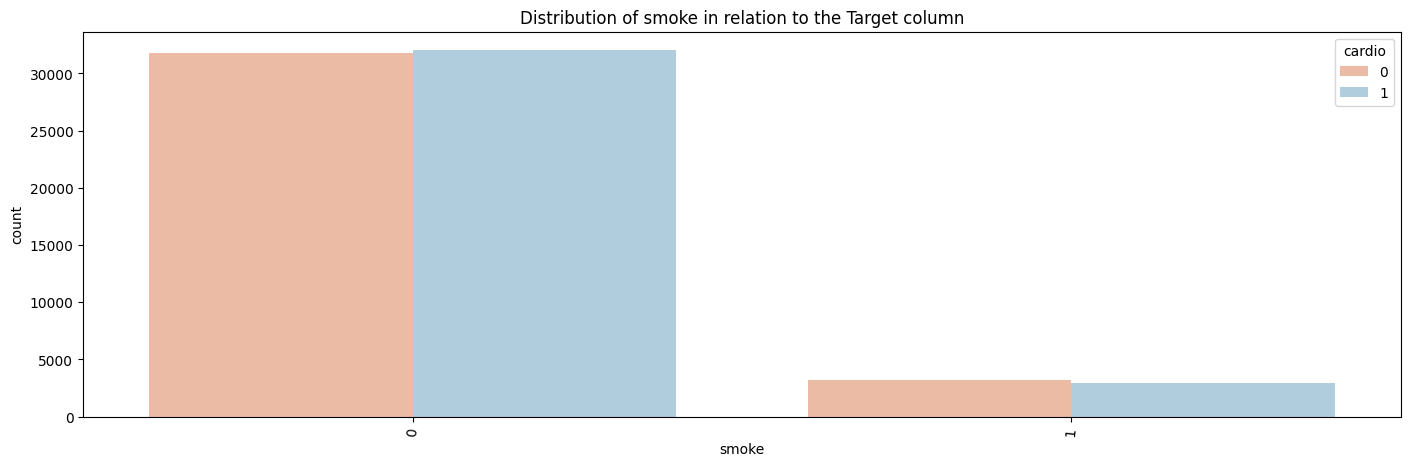


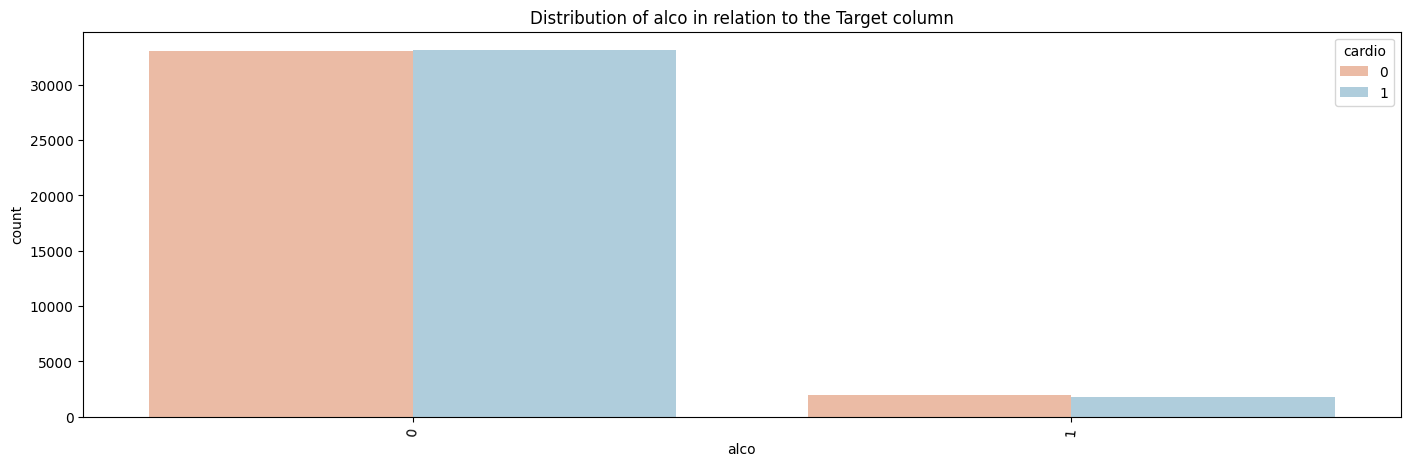


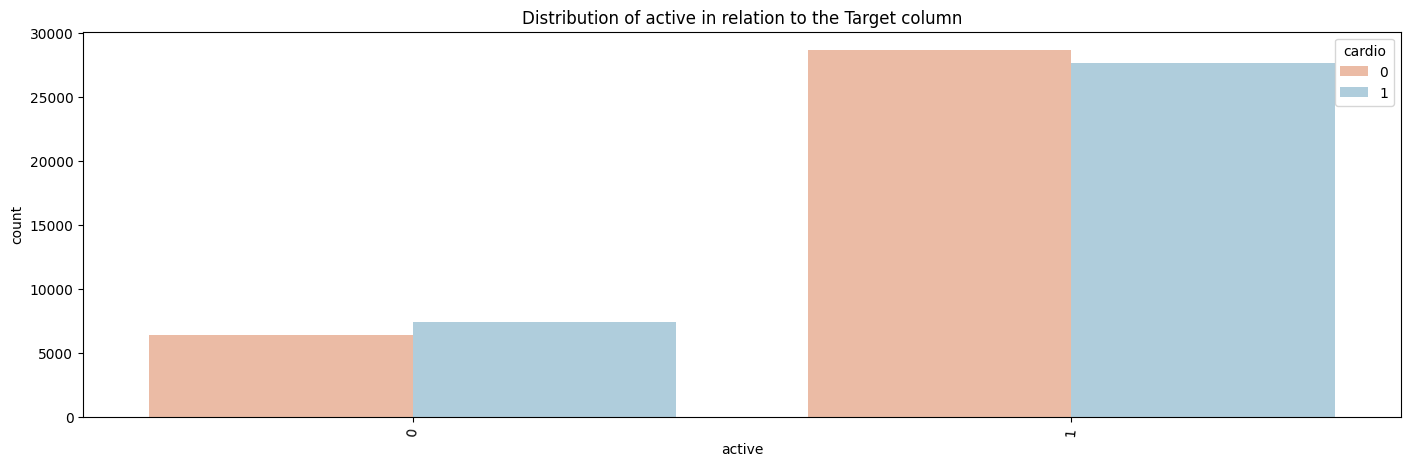


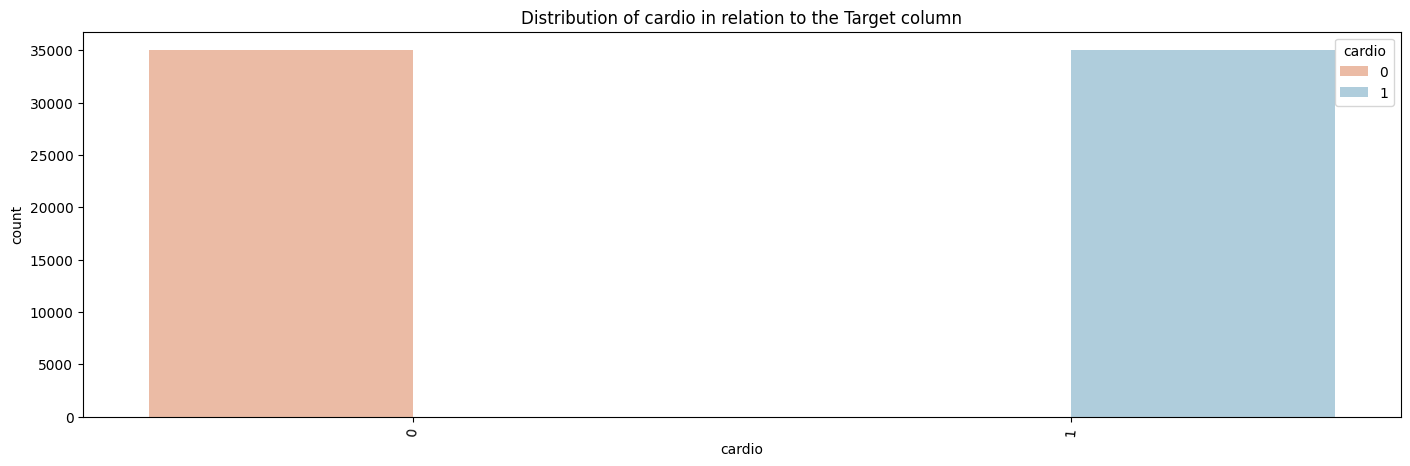












## 

## **Dataset Pre-processing:**

**Faults:**

* Null values: Our dataset did not have any null values. So as instructed we manually inserted null values into the dataset.
* Negative Blood pressure values: ‘ap\_high’ and ‘ap\_lo’ columns had negative values in the columns.
* Outliers: We detected some columns that had unrealistic values, like the column ‘height’ had values over 200 and ‘weight’ had values below 20 etc.
* Duplicate values: The dataset also had many duplicate values in many rows.

**Solutions:**

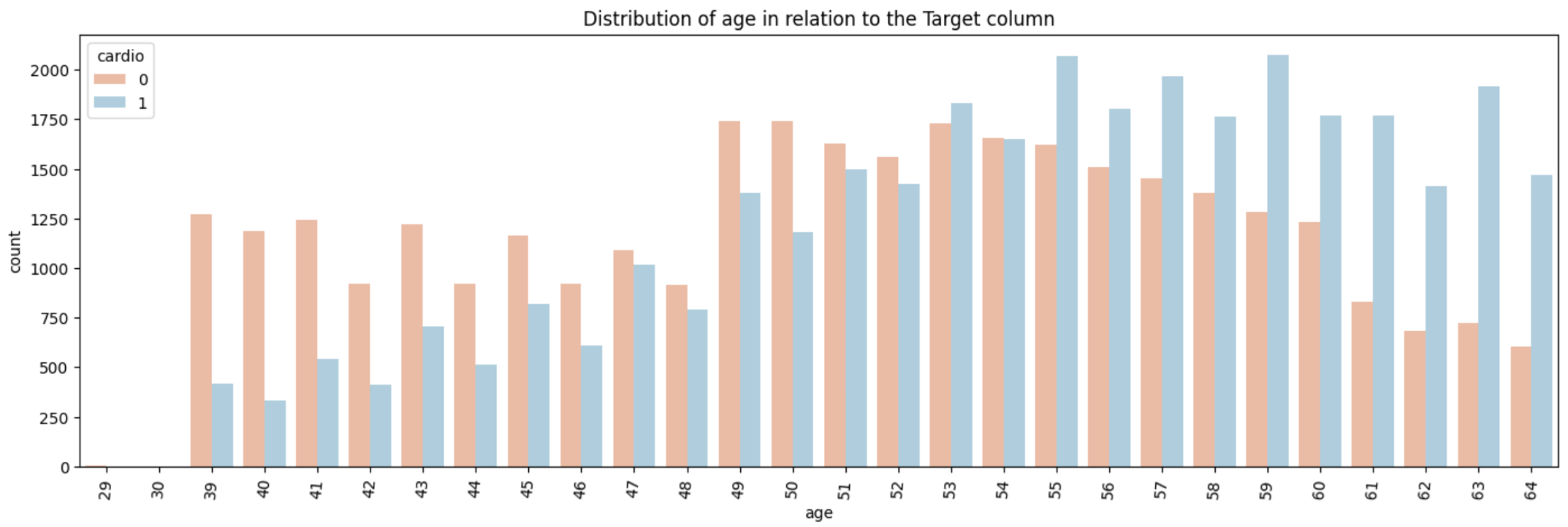
Deleting Rows and Columns:

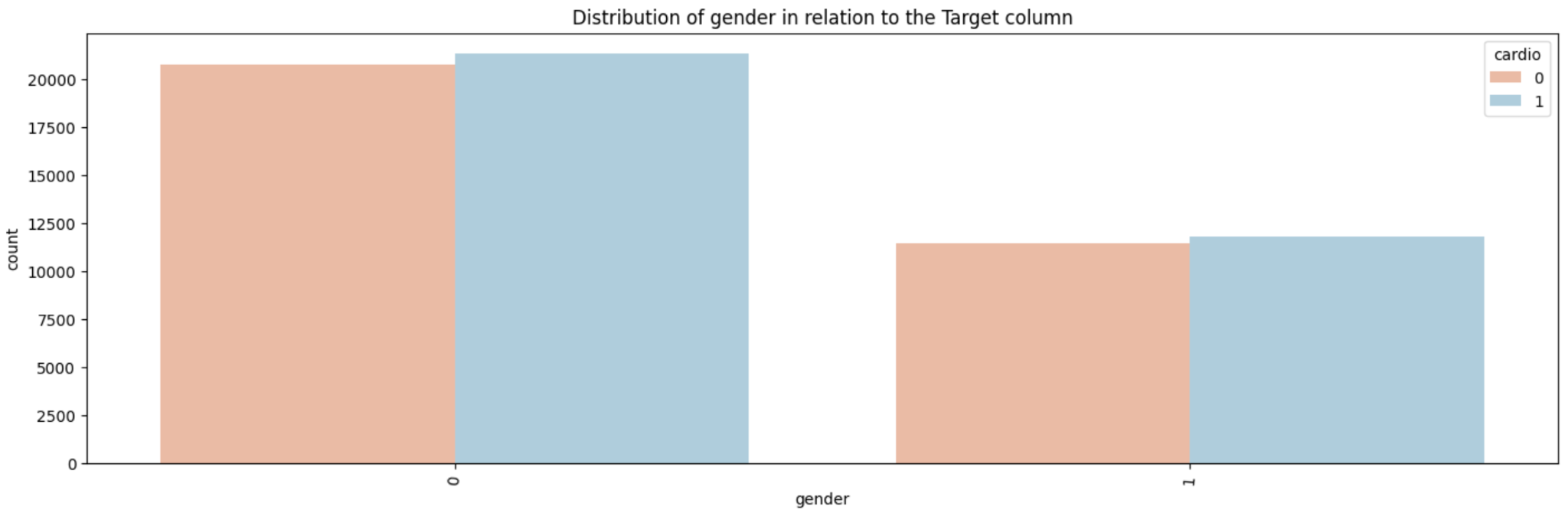
* Deleting ID Column: First of all we checked the correlation between the columns in the dataset and found that the “ID” column had the lowest correlation with the other columns. So we dropped it.
* Deleting Null rows: We found out how many null values were there and the rows which contained them. Then we deleted those rows.
* Deleting rows with outliers: We filtered out the outlier values by deleting the rows that had unrealistic values.
* Deleting rows with duplicate values.

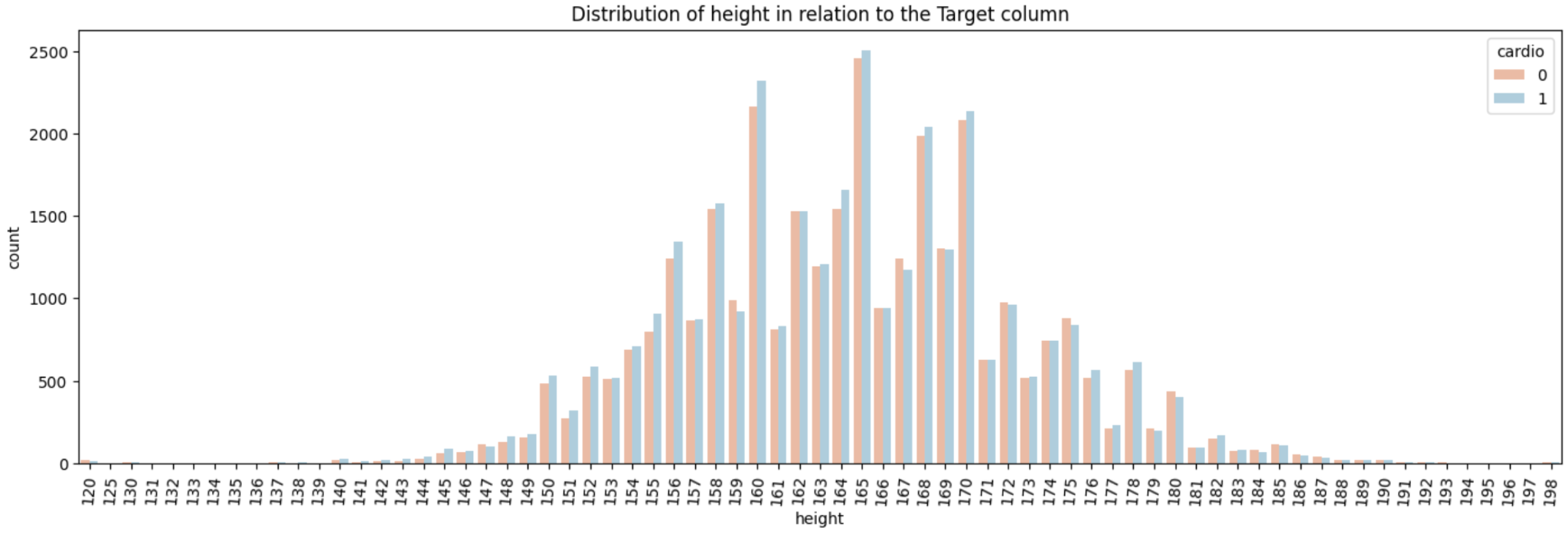
Encoding Rows and Columns:

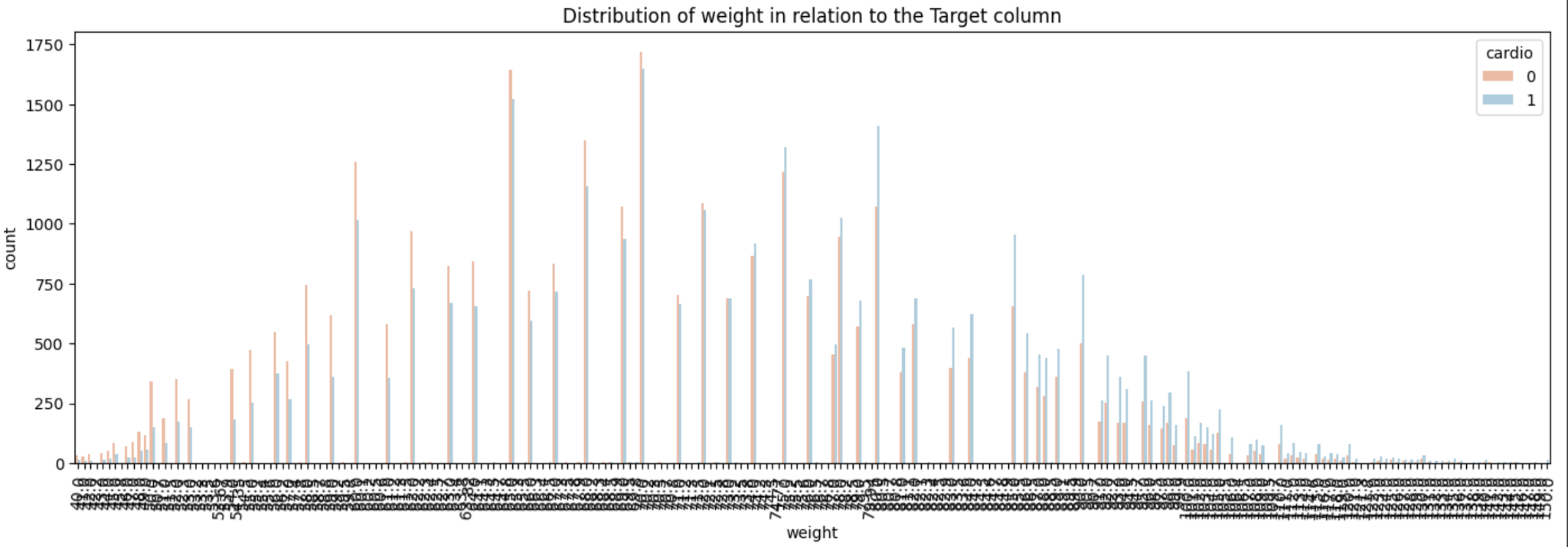
* ‘Age’ column was measured in days, we encoded it by dividing the column values by 365 and changing the unit to years.
* The ‘gender’ column had values 1 and 2, we mapped them to 0 and 1. 0 meaning female and 1 meaning male.
* The ‘Cholesterol’ column had values 1,2 and 3, we mapped them between 0 to 1, where 0 meant low cholesterol, 0.5 meant medium and 1 meant high cholesterol.
* ‘Ap\_high’ and ‘Ap\_lo’ columns were encoded to absolute values, as blood pressure cannot be negative.
* The ‘gluc’ column had values 1,2 and 3, we mapped them between 0 to 1, where 0 meant low gluc, 0.5 meant medium and 1 meant high gluc.

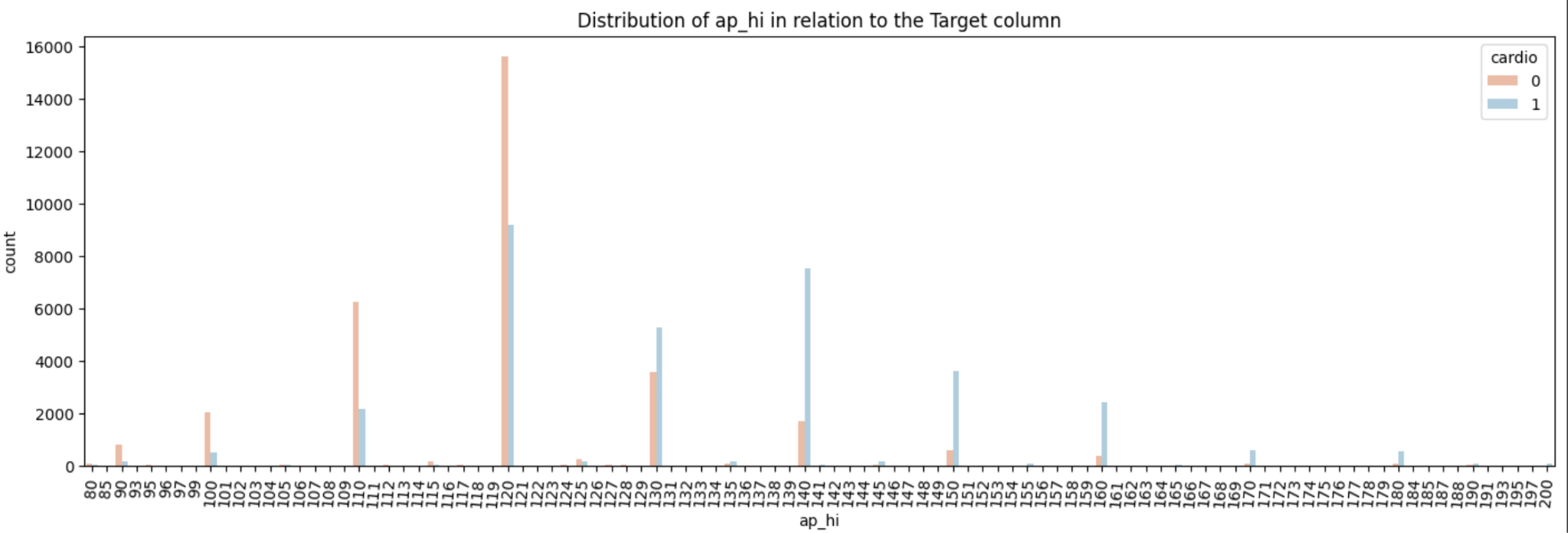
**Dataset visualization after preprocessing:**

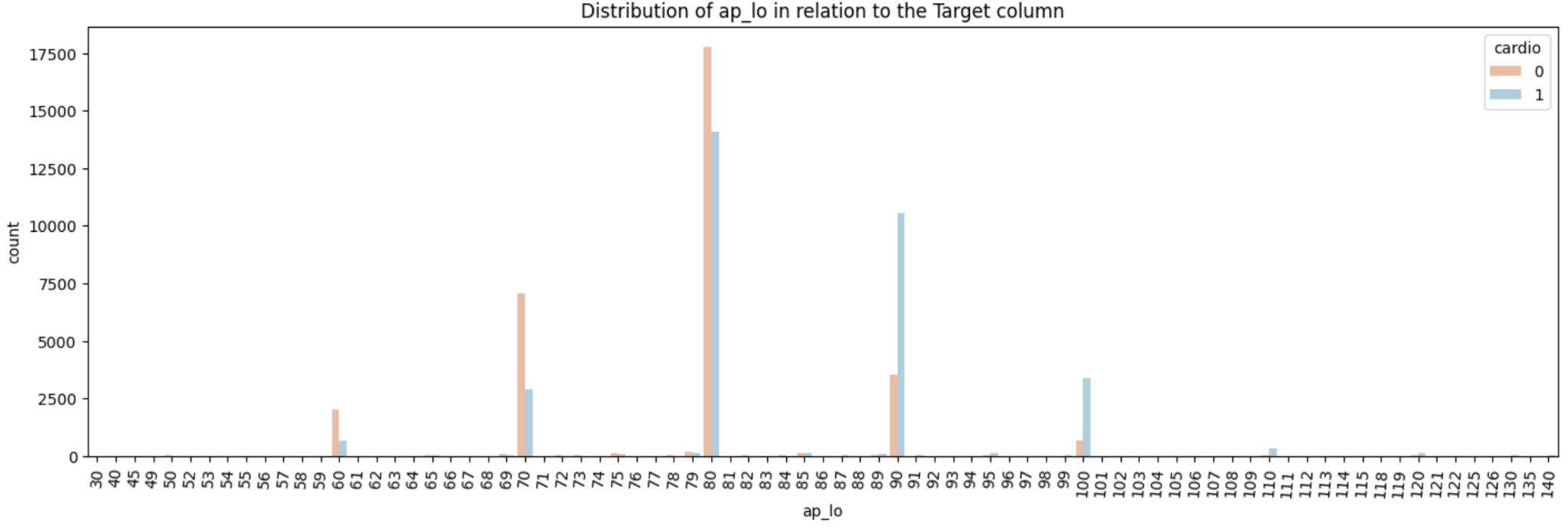
****

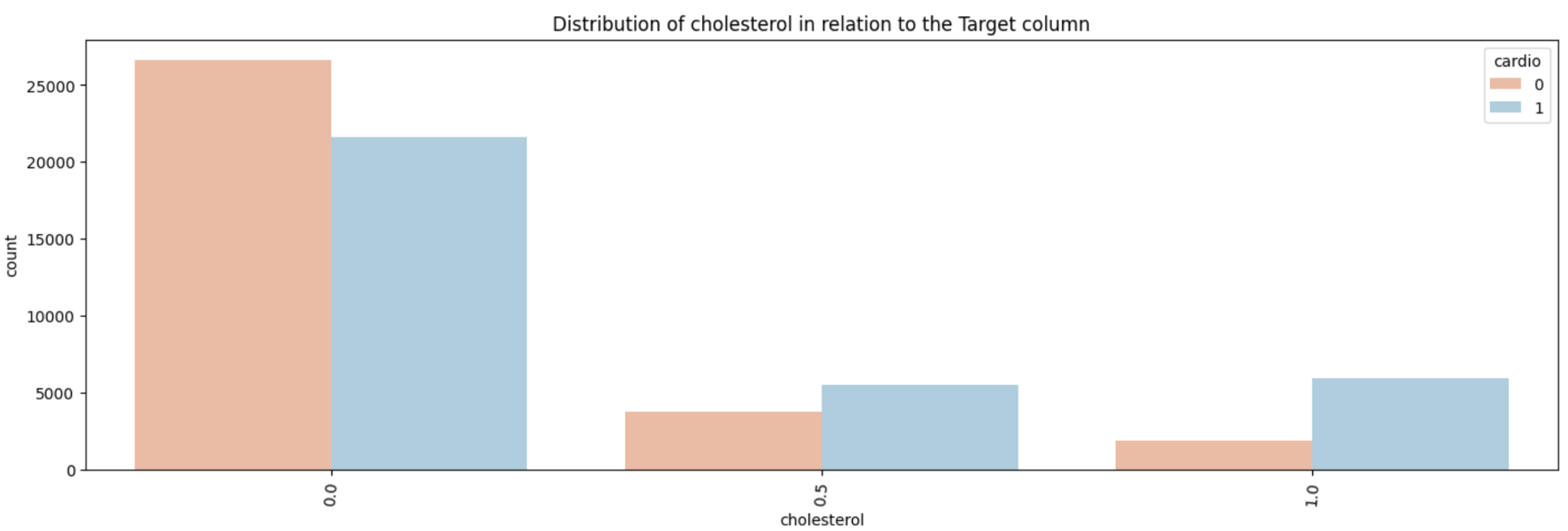
****

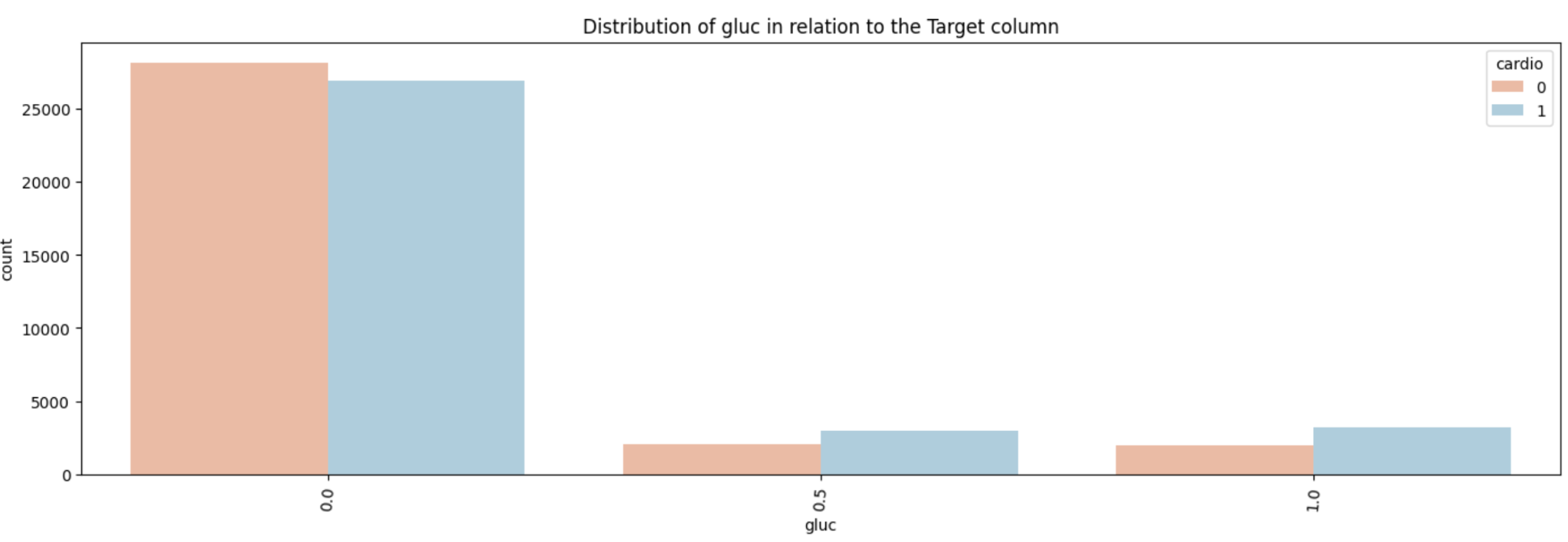


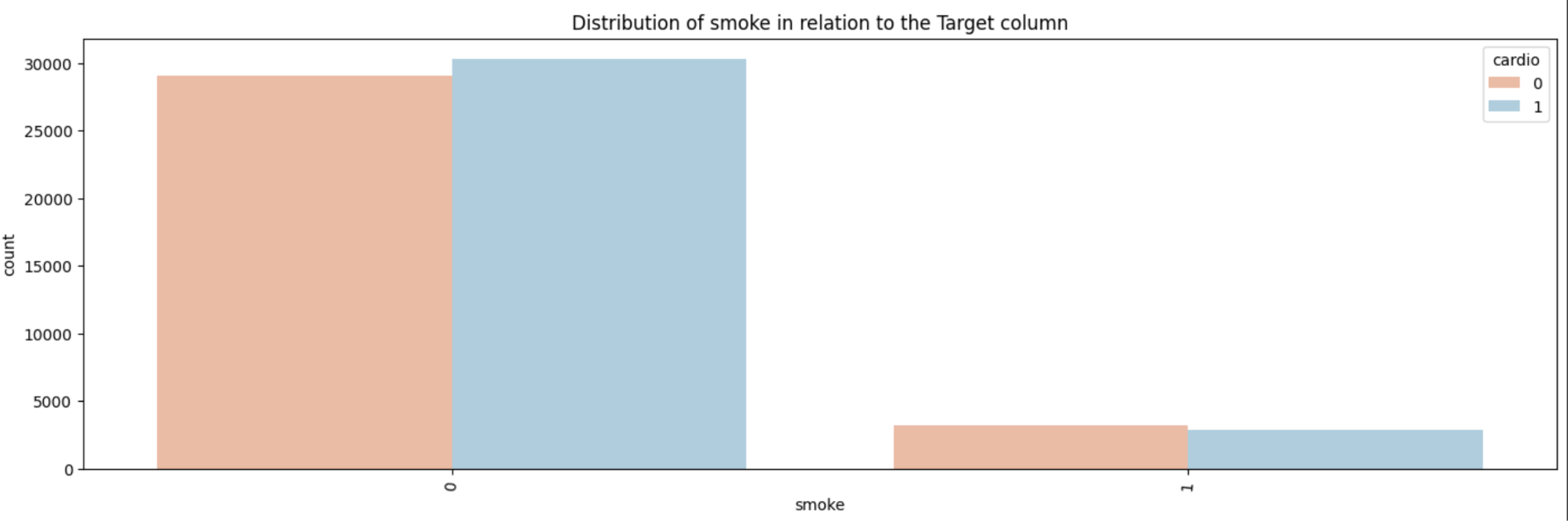


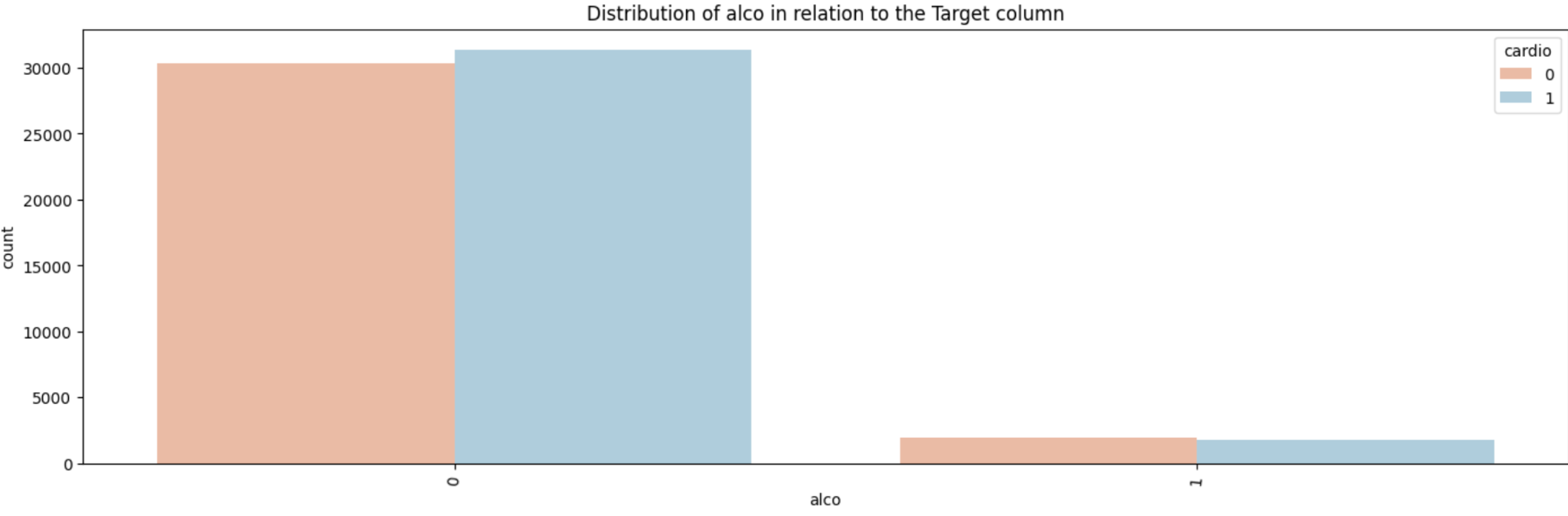


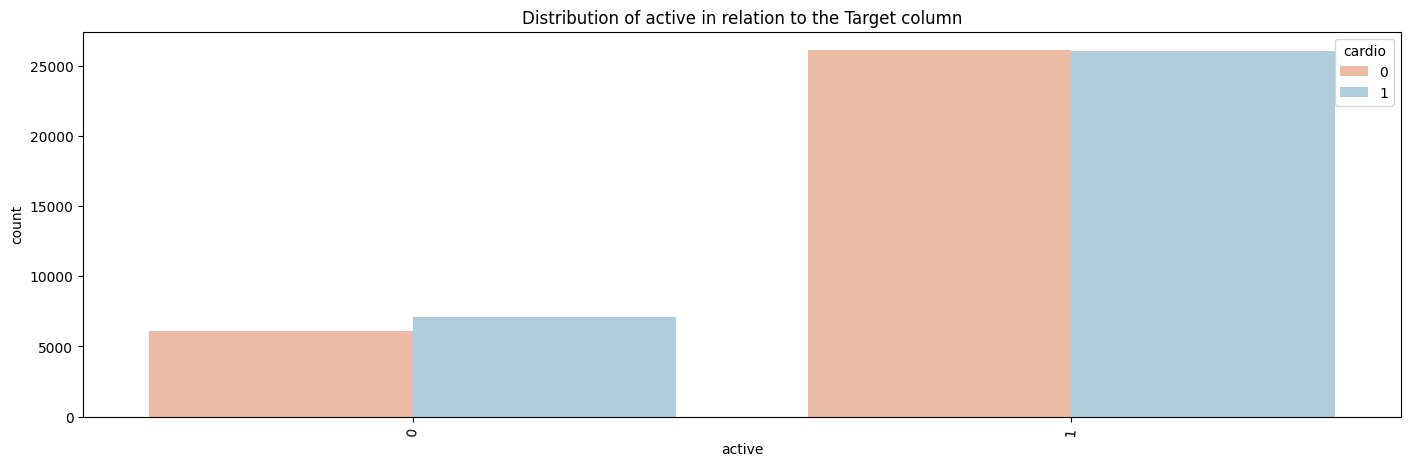


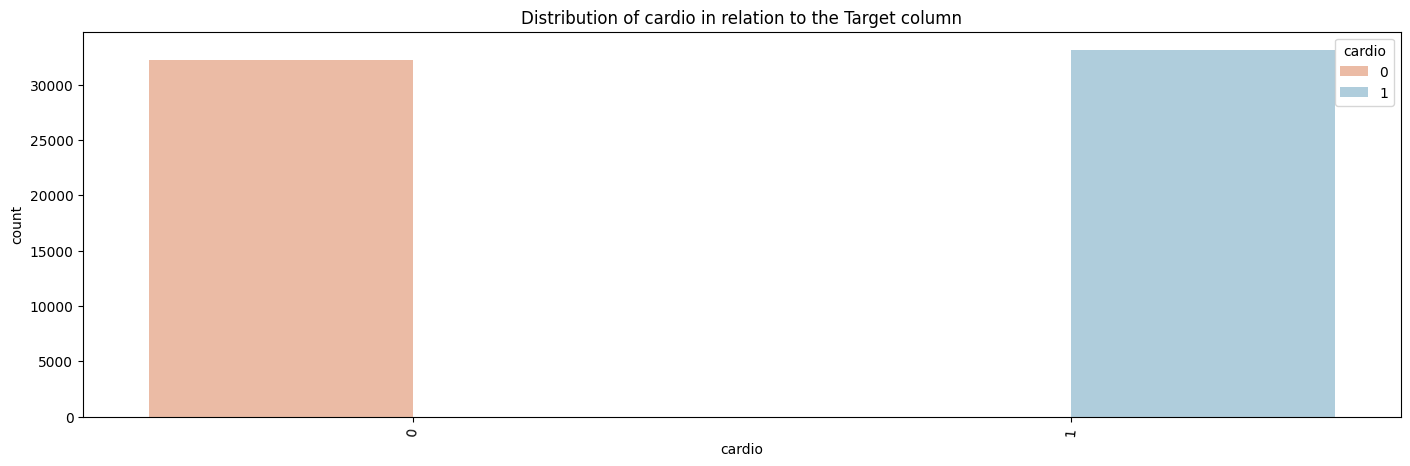












## **Feature Scaling:**

* The model Support Vector Machine(SVM) benefits heavily from feature scaling. Thus, we applied the min max scaler on the features to fit all the features within the range of 0-1.  
  This scaling significantly improved the runtime of the SVM model.

## **Dataset Splitting:**

* While splitting a random\_state = 42 was declared for the data points to have a equal probability of being split every time this random state is called
* We used stratified=y to keep the ratio of the target column same in both the test and train splits
* 70% of the dataset, numbering 45737 data points were split to make the train dataset
* 30% of the dataset, numbering 19602 data points were split to make the test dataset

## **Model Training and Testing:**

* **KNN (for classification problem) :**

K-Nearest Neighbors (KNN) is a simple and versatile classification algorithm used for both binary and multiclass classification problems.

KNN is a non-parametric learning method that doesn't require training before producing predictions and doesn't make any assumptions about the distribution of the underlying data. KNN is comparatively easy to implement, however the algorithm can be computationally expensive, especially with large datasets, as it requires calculating distances for each prediction.

* **Random Forest(for classification problem):**

Random Forest is a versatile ensemble learning method that is used for classification tasks.

This algorithm builds a forest of decision trees, each trained on a random subset of the training data. The final classification is determined by a majority vote (for example, "voting" for the most common class among all the trees) or by averaging the predicted probabilities across the trees. It is effective in handling complex tasks, dealing with high-dimensional data, and providing robust predictions.

* **Logistic Regression (for classification):**

Logistic Regression is a statistical method commonly used in machine learning for binary classification problems.

Logistic Regression is often a good starting point for binary classification tasks due to its simplicity, interpretability, and efficiency. It can handle any number of classes.

* **Naive Bayes (for classification problem):**

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem. It is commonly used for classification tasks, particularly in situations where there are multiple features, and the goal is to predict the class or category of a given instance.

Despite its simplicity and the "naive" assumption of feature independence, Naive Bayes often performs well in practice, especially in text classification and spam filtering applications.

* **Support vector machine (for classification/regression problem):**

SVM is a popular classification algorithm that works by finding the hyperplane in an N-dimensional space (where N is the number of features) that distinctly classifies the data points.

SVM can handle high-dimensional feature spaces and establish difficult decision boundaries, which makes it useful for multiclass classification.

* **Gradient Boosting(for classification problem):**

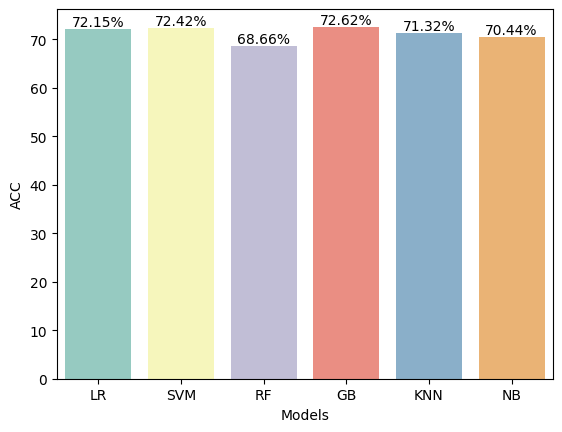
Gradient Boosting is an ensemble learning technique that combines the predictions of multiple weak learners to create a strong predictive model.

The primary idea behind gradient boosting is to iteratively improve the model's performance by correcting the errors made by previous models. Gradient boosting minimizes a loss function by iteratively moving in the direction of the negative gradient of the loss.

## 

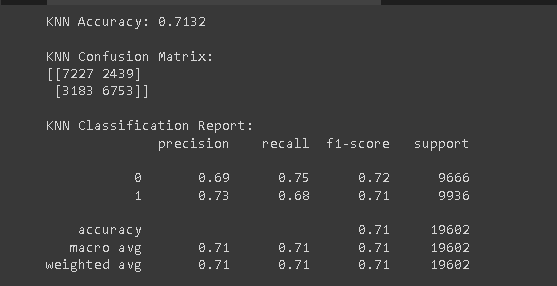
## **Model Selection/Comparison Analysis:**

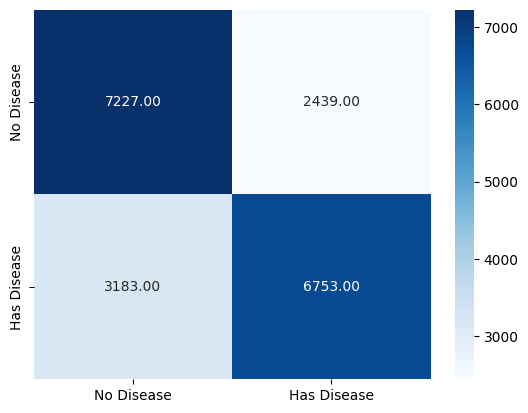
1. Barchart showcasing prediction accuracy of all model



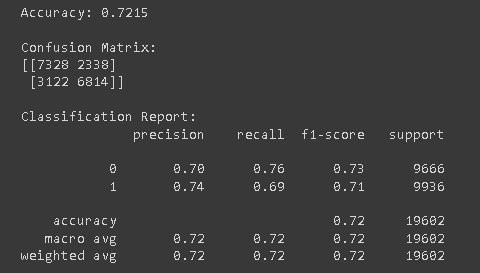
1. Precision,recall comparison of each model and confusion matrix

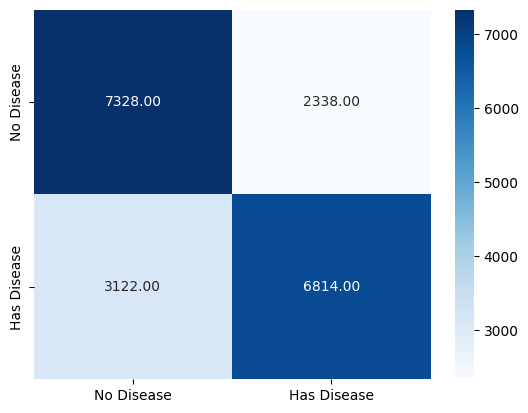
* Precision measures the accuracy of the positive predictions made by the model. It is the ratio of true positive predictions to the total predicted positives.
* Recall measures the ability of the model to capture all the relevant instances. It is the ratio of true positive predictions to the total actual positives.
* **KNN**

****

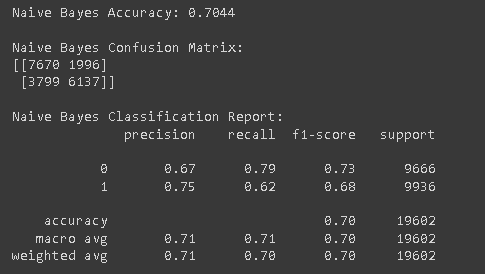
****

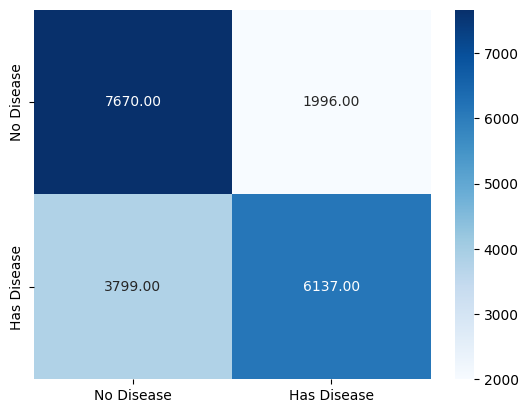
* **Logistic Regression**

****

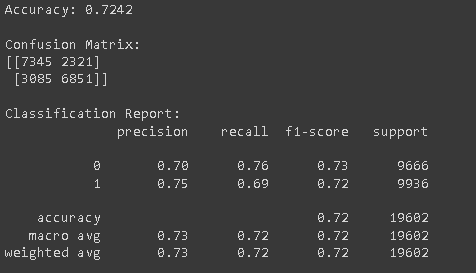
****

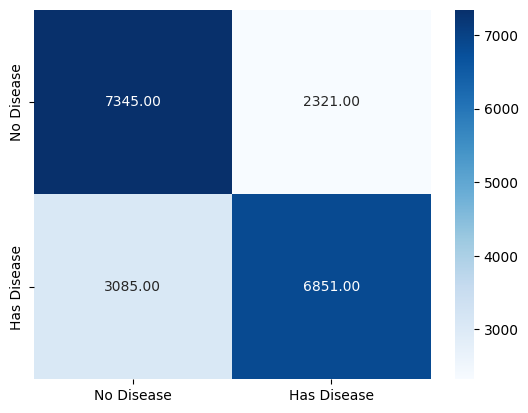
* **Naive Bayes**

****

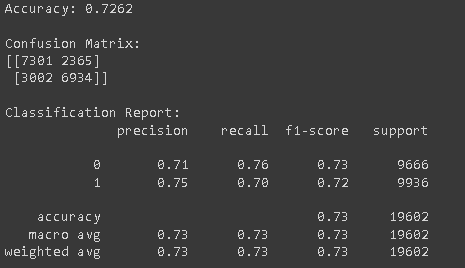
****

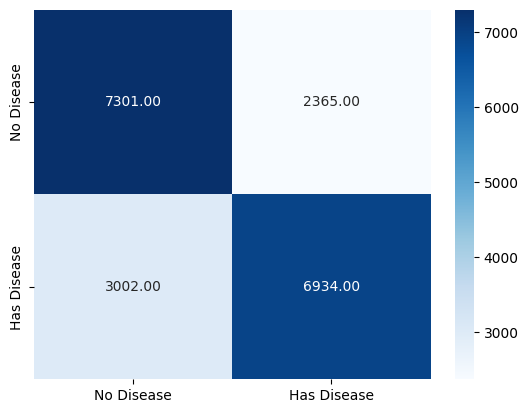
* **Support Vector Machine**

****

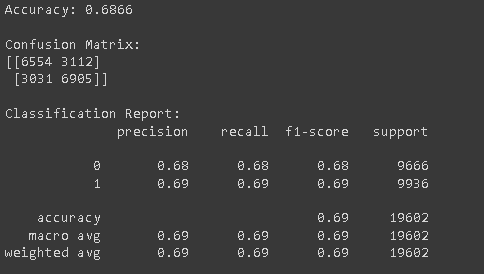
****

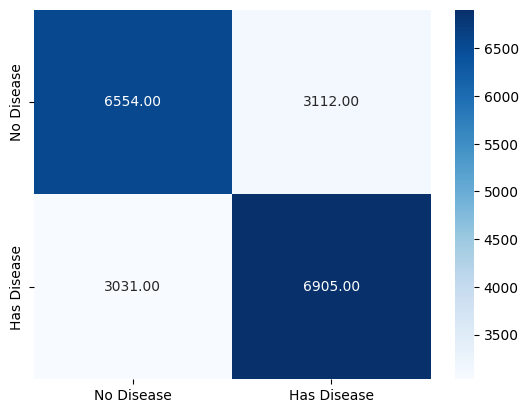
* **Gradient Boosting Classifier**

****

****

* **Random Forest Classifier**

****

****

## **Conclusion:**

The goal of this project was to create a model that could predict Cardiovascular Diseases among individuals using their medical information. We managed to train and test a dataset of 70000 data points which could accurately predict the target almost 73% of the time. Additionally, cardiovascular diseases are absent in most of the population; the prediction and recall parameters are a better metric to understand the accuracy of the prediction.

Some more in depth analysis of the dataset and more correlated data points would have helped with better accuracy.

Overall, this project shows how machine learning techniques can be used to solve important diagnostic problems and improve public health.