Predicting Cardiovascular Disease Risk Using Naïve Bayes Classifier: A Comprehensive Analysis

Authored by: Nourhan Ahmed, Hazem Zakaria, Ahmed Emad, Khaled Mohamed, Omar Nabil

Under the supervision of: Dr. Ibrahim Youssef

Department of Systems and Biomedical Engineering, Cairo University.

KEYWORD

Binary classification,
Naïve Bayes classifier,
Cardiovascular risk,
Outlier removal,
Descriptive statistics,
Standardization,
Model accuracy,
Feature selection,
Model evaluation.

ABSTRACT

This research project explores the use of the Naïve Bayes classifier for predicting the risk of cardiovascular disease. The study utilizes the Cardiovascular Risk Data from Kaggle and applies data preprocessing techniques such as outlier removal and feature standardization. Through exploratory data analysis, the relationships between features and the target class are examined. The Naïve Bayes classifier is implemented and trained on the dataset, with accuracy as the evaluation metric. The results demonstrate the potential effectiveness of the classifier in predicting cardiovascular risk. This research contributes to understanding cardiovascular risk factors and showcases the practical application of machine learning techniques in healthcare.

1 Introduction

The aim of this project was to perform binary classification using the Naïve Bayes (NB) classifier.z

Cardiovascular diseases (CVDs) are a major global health concern, causing a significant number of deaths each year. Early identification and accurate prediction of cardiovascular risk factors play a crucial role in preventing and managing these conditions. In this project, we explore the application of the Naïve Bayes (NB) classifier for binary classification to predict the risk of cardiovascular disease The dataset chosen for this study is the Cardiovascular Risk Data, sourced from Kaggle. This dataset contains various attributes such as age, gender, blood pressure, cholesterol levels, and lifestyle factors. By analyzing these attributes, we aim to develop a classification model that can effectively classify individuals into high or low risk categories for cardiovascular disease in this project, we will perform data preprocessing steps to handle outliers and standardize the numerical features. Descriptive statistics will be calculated to summarize the central tendency and dispersion of the data. The dataset will then be split into training and testing subsets. Using the training data, we will implement the NB classifier from scratch and train it on the dataset. Subsequently, we will evaluate the model's performance by predicting the classes of the testing data. The accuracy of the classifier will be calculated to assess its effectiveness in predicting cardiovascular risk.

2 Methods

Before applying the NB classifier, the data underwent several preprocessing steps:

Data loading: The dataset was imported into a Jupyter notebook using the Pandas library.

Outlier removal: Statistical methods were employed to identify and remove outliers from the numerical attributes, excluding the 'id' column.

Descriptive statistics: Measures of central tendency (mean, median, mode) and dispersion (standard deviation, range) were calculated for the data with no outliers.

Standardization: The features were standardized using the calculated descriptive statistics.

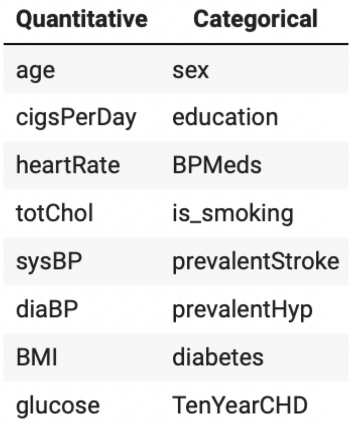
2.1 Dataset Selection

The Cardiovascular Risk Data dataset from Kaggle was chosen for the binary classification experiment. This dataset contains a variety of attributes related to cardiovascular health, including:

- “Id”: the ID of each patient.
- “age”: the age of the patient.
- “education”: categorized into level 1, 2, 3 and 4.
- “sex”: whether the patient is male or female.
- “is_smoking”: whether or not the patient is a current smoker.

- “cigsPerDay”: number of cigarettes consumed per day.
- “BPMeds”: whether or not the patient was on blood pressure medications.
- “prevalentStroke”: whether or not the patient had previously had a stroke.
- “prevalentHyp”: whether or not the patient was hypertensive.
- “diabetes”: whether or not the patient has diabetes.
- “totChol”: total cholesterol level.
- “sysBP”: systolic blood pressure.
- “diaBP”: diastolic blood pressure.
- “BMI”: body mass index.
- “heartRate”: heart rate of the patient.
- “glucose”: glucose level.
- “TenYearCHD”: 10-year risk of coronary heart disease (the target label).

Here is the classification of the features:



Quantitative	Categorical
age	sex
cigsPerDay	education
heartRate	BPMeds
totChol	is_smoking
sysBP	prevalentStroke
diaBP	prevalentHyp
BMI	diabetes
glucose	TenYearCHD

As shown above, the “id” column is not included in the table as it is not considered a feature that will affect our target label but rather used to uniquely identify each row.

2.2 Data Preprocessing

2.2.1 Data Loading

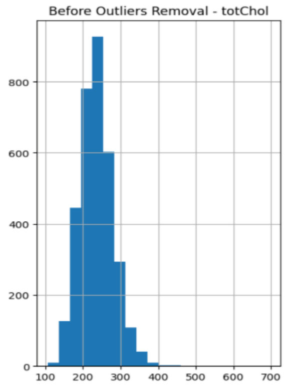
The dataset was imported into a Jupyter notebook using the Pandas library. The Pandas library provides efficient data manipulation and analysis capabilities. We used “**label encoding**” which involves transforming categorical variables into numerical representations to facilitate the analysis and modeling tasks in machine learning. It is a commonly used technique that assigns a unique numerical value to each distinct category within a given column. This enables machine learning algorithms to effectively process the categorical data.

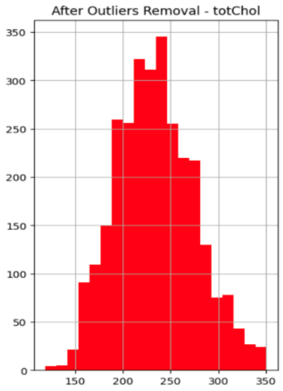
In the specific context mentioned, the **'sex'** and **'is_smoking'** columns are subjected to label encoding. Label encoding replaces the original categories in these columns with corresponding numerical values. For instance, in the **'sex'** column, the categories 'male (M)' and 'female (F)' are encoded as 1 and 0, respectively. Similarly, in the **'is_smoking'** column, the categories 'YES' and 'NO' are transformed into 1 and 0, respectively.

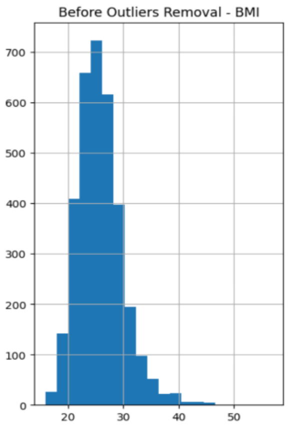
2.2.2 Outlier Detection and Removal

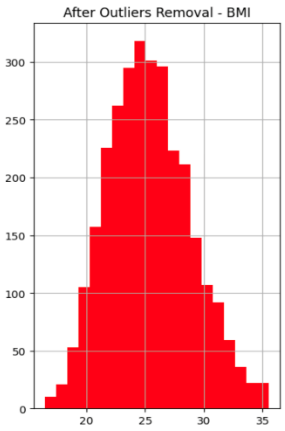
To ensure the robustness of the analysis, outliers were identified and removed from the dataset using a suitable statistical method. Numeric columns, excluding specific categorical features, were selected for outlier analysis. The **interquartile range (IQR)** method was employed to detect outliers by defining a threshold based on the IQR. The identified outliers were subsequently removed from the dataset, excluding the 'id' column, which is meaningless to remove any outliers from. The categorical features were also excluded from the outlier removal process as they only have specific values (binary values), which makes it insensible to remove any data from them.

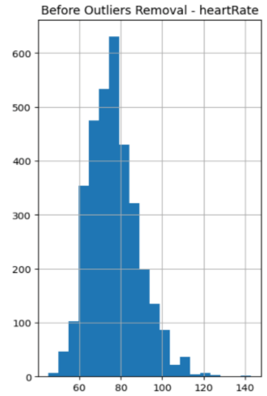
Displaying the before and after effect of removing outliers from the dataset:

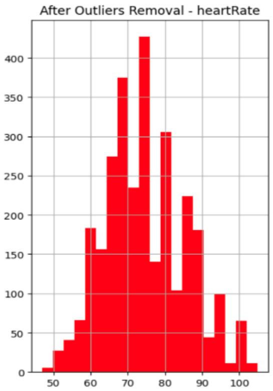


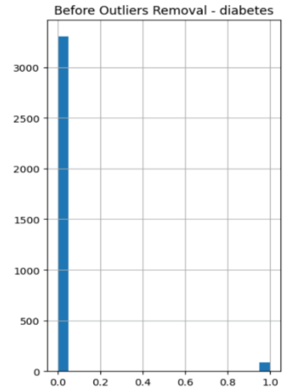


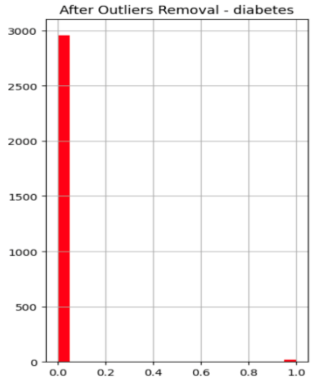












*Note that categorical features, as we mentioned, did not participate in the removal process, however, due to removal of outliers from the quantitative columns, the whole row of data is deleted and this surely affects the count of categorical rows. Here is an example for the “diabetes” feature.

2.2.3 Handling Missing Values

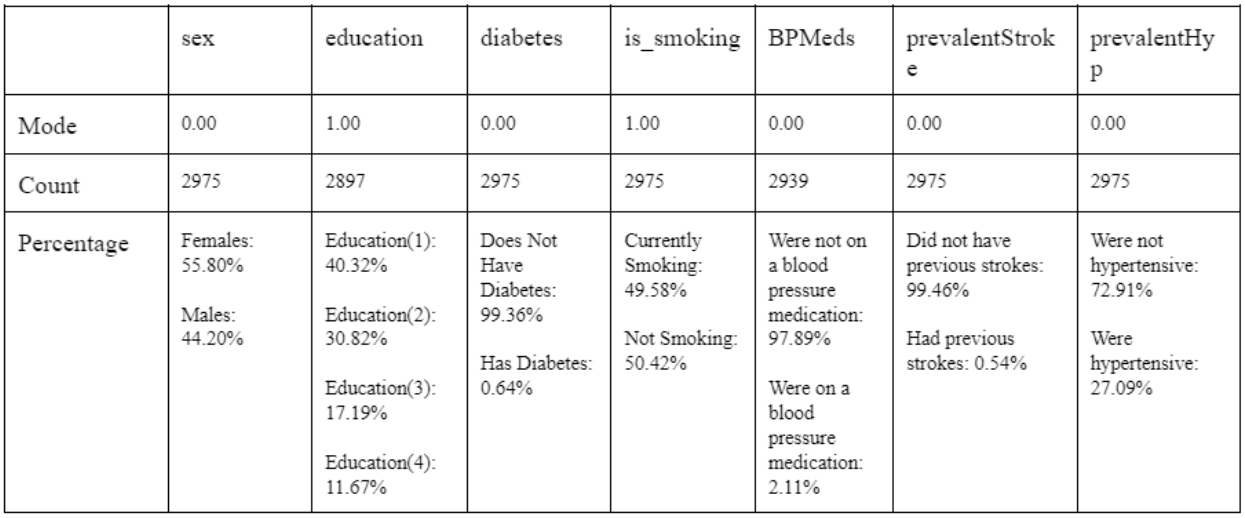
Missing values are carefully addressed. We applied the **imputation strategy** as the missing values in the dataset are filled with the mean value of each respective column which avoids reducing the sample size to maintain statistical power.

2.3 Descriptive Statistics

Descriptive statistics were calculated to quantitatively summarize the central tendency and dispersion of the data. Measures of central tendency, including the mean, median, and mode, provided insights into the typical values of the attributes. Measures of dispersion, such as the standard deviation, range, and interquartile range, helped understand the variability and spread of the data.

	age	cigsPerDay	totChol	sysBP	diaBP	BMI	heartRate	glucose
Mode	40.00	0.00	240.00	120.00	80.00	22.91	75.00	75.00
Mean	49.057479	9.133581	234.367097	129.327227	81.517311	25.389858	74.891056	78.313682
Median	48.00	1.00	232.00	127.00	81.00	25.14	75.00	77.00
Standard deviation	8.495008	11.645535	40.988083	18.265915	10.540621	3.513721	10.771652	11.014433
Range	38.00	50.00	232.00	101.00	61.00	19.05	58.00	64.00
Minimum	32.00	0.00	119.00	83.50	52.00	16.48	47.00	47.00
Maximum	70.00	50.00	351.00	184.50	113.00	35.53	105.00	111.00
Count	2975	2957	2942	2975	2975	2964	2974	2697

These statistical measures offer a comprehensive overview of the dataset, allowing us to understand the distribution and characteristics of the variables being studied. Further analysis of the dataset was done, calculating the counts and percentages of categorical data in different columns. The count values represent the frequency of occurrence for each value, while the percentages represent the proportion of each value in relation to the total number of non-null observations in the respective column. By obtaining these counts and percentages, we gain insights into the distribution and prevalence of different characteristics within the dataset.

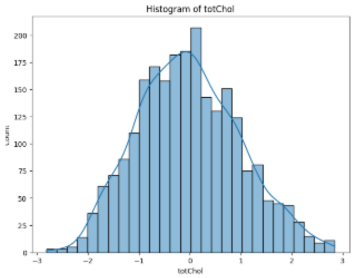


	sex	education	diabetes	is_smoking	BPMeds	prevalentStroke	prevalentHyp
Mode	0.00	1.00	0.00	1.00	0.00	0.00	0.00
Count	2975	2897	2975	2975	2939	2975	2975
Percentage	Females: 55.80% Males: 44.20%	Education(1): 40.32% Education(2): 30.82% Education(3): 17.19% Education(4): 11.67%	Does Not Have Diabetes: 99.36% Has Diabetes: 0.64%	Currently Smoking: 49.58% Not Smoking: 50.42%	Were not on a blood pressure medication: 97.89% Were on a blood pressure medication: 2.11%	Did not have previous strokes: 99.46% Had previous strokes: 0.54%	Were not hypertensive: 72.91% Were hypertensive: 27.09%

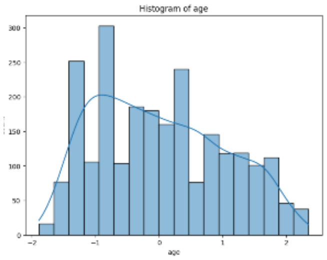
2.4 Feature Standardization - Histograms and Distribution Analysis

To ensure that features were on a similar scale and to prevent certain features from dominating the analysis due to their larger magnitudes, feature standardization was applied. This involved transforming the data such that each feature had a mean of 0 and a standard deviation of 1.

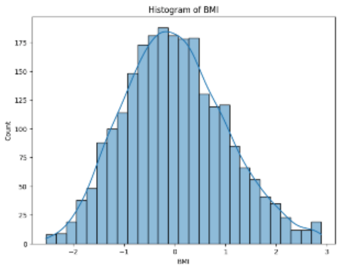
Standardization was performed using the calculated descriptive statistics. Only the quantitative features were standardized which are (age, cigsPerDay, heartRate, totChol, sysBP, diaBP, BMI and glucose). Here are the distributions of some features after standardization:



Apparently similar to Normal distribution

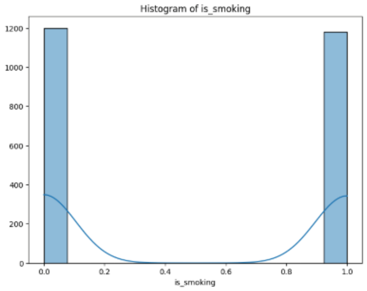


Not similar to any known distribution

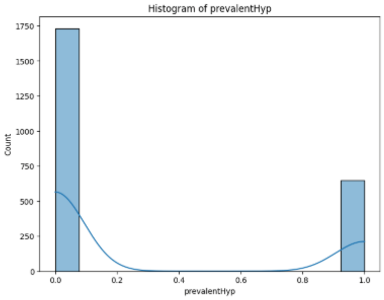


Apparently similar to Normal distribution

Since the categorical features did not undergo the standardization process, their "after standardization distributions" were not plotted. Yet here are the shapes of their distributions:



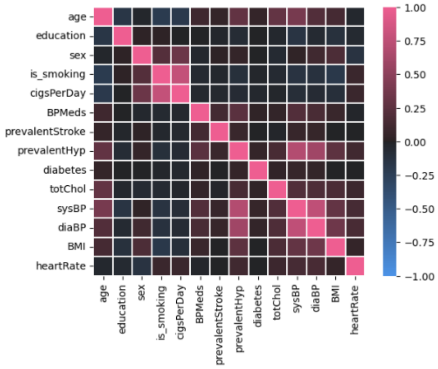
Not similar to any known distribution



Not similar to any known distribution

2.5 Heat Map

The heat map was employed as a visualization technique to analyze the dataset and examine potential relationships between the variables under investigation.By visually inspecting the heat map, patterns of color intensity can be observed, indicating the strength and direction of the relationship between variables. Areas of high color intensity indicate a stronger relationship, while areas with lower intensity suggest a weaker or no relationship.

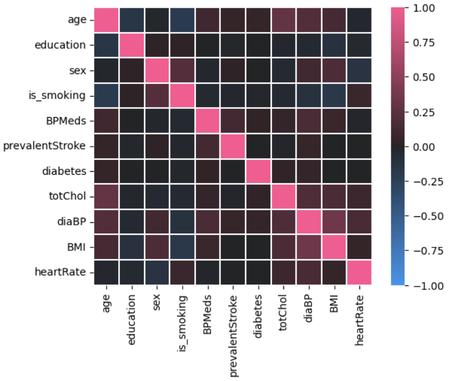


As shown in the figure .The heat map shows that there is a strong relationship between the columns:

- “sysBP” with “diaBP”
- “sysBP” with “prevalentHyp”
- “diaBP” with “prevalentStroke”
- “cigsPerDay” with “is_smoking”

In order to use Naive Bayes, all the features should ideally be independent of each other. Therefore, it is recommended to remove some of the features with relationships to improve the accuracy of the model and ensure the assumption of feature independence in Naive Bayes, so we decided to remove the following features : “sysBP” , “cigsPerDay” and “prevalentHyp”.

The resulting heat map is shown in the figure below:



2.6 Data Partitioning

The preprocessed dataset was randomly split into two partitions: **80% training data** and **20% testing data**. This partitioning ensured that the model was trained on a sufficiently large dataset while retaining a separate portion for evaluating its performance on unseen data. The target class(TenYearCHD) is included in both the X_train and Y_train data sets for classification reasons discussed later.

2.7 Exploratory Data Analysis (EDA)

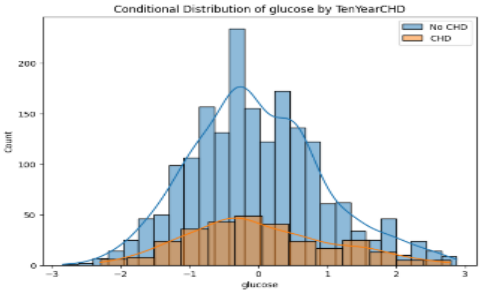
2.7.1 Statistical Tests for Normality

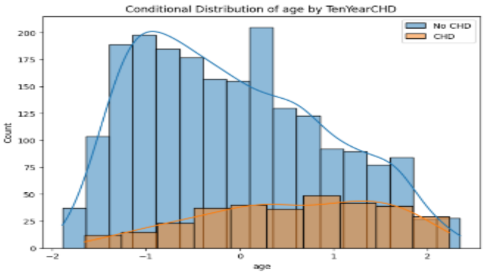
Null and Alternative hypotheses of **Quantitative** features using **Shapiro-Wilk** test: the ‘shapiro’ function from the SciPy library is used to test the null hypothesis that the data is normally distributed against the alternative hypothesis that the data is not normally distributed. To perform the Shapiro-Wilk test, the test statistic (W) is calculated based on the observed data. The test statistic measures the discrepancy between the observed data and the expected values under the assumption of normality. The p-value associated with the test statistic is then compared to a chosen significance level (0.05) to make a decision. If the p-value is greater than 0.05, we fail to reject the null hypothesis. It indicates that there is not enough evidence to conclude that the data significantly deviates from a normal distribution. In other words, we can consider the data as likely being normally distributed. On the other hand, if the p-value is smaller than the chosen significance level, we reject the null hypothesis. It suggests that there is sufficient evidence to conclude that the data significantly departs from a normal distribution. In this case, we consider the data as not being normally distributed. It's important to note that the Shapiro-Wilk test is sensitive to departures from normality, especially for smaller sample sizes. Therefore, even minor deviations from normality can lead to rejecting the null hypothesis.

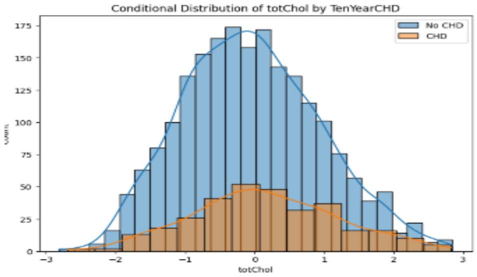
Null and Alternative hypotheses of **Categorical** features using **Chi-square** test**:** performed between categorical features and the target variable ‘TenYearCHD’. It uses the ‘chi2_contingency’ function from the scipy.stats module to calculate the chi-square statistic and p-value. The code iterates over each categorical feature creating a contingency table using ‘pd.crosstab’ to calculate the frequency counts. The chi-square test is then conducted on the contingency table, and the results are printed, indicating whether there is an association between each feature and the target variable. A significance level of 0.05 is used to determine the presence of association. The null hypothesis indicates that there is no association between the categorical feature and the target variable, while the alternative hypothesis indicates that there is an association between the categorical feature and the target variable.

2.7.2 *Conditional Distributions*

Conditional distributions of each feature on the target class (TenYearCHD) were analyzed and visualized. This analysis helped understand how the features varied across different classes, providing insights into their predictive capabilities and potential relationships with the target variable. Here is a visualization of the conditional distributions of some features:







2.8 *Naïve Bayes Classifier Implementation*

2.8.1 *Library Selection*

- Scikit-learn (‘sklearn’): efficient implementation of Naive Bayes classifiers.
- NumPy (‘numpy’): provides support for handling arrays.
- Pandas (‘pandas’): provides data structures like DataFrames, which allows splitting data into training and testing sets.

2.8.2 *Training the NB Classifier*

Naive Bayes (NB) classifier is trained using a Gaussian implementation. The dataset is split into a training set and a testing set. The training set is used for training the NB classifier, while the testing set is used for evaluation. The NB classifier is applied to the training set using a function called **‘naive_bayes_gaussian’**, which is assumed to implement the Gaussian NB algorithm. The predicted labels are obtained, and the accuracy of the model is calculated by comparing the predicted labels with the true labels.

2.9 *Model Evaluation*

2.9.1 *Accuracy Calculation*

The trained NB classifier was used to predict the classifications of the testing data. The accuracy of the model was calculated by comparing the predicted labels with the true labels of the testing data. Accuracy provides an overall measure of the model's performance in correctly classifying instances.

2.9.2 *Comparison with Standard Python Packages*

To validate the results obtained from the custom implementation, the performance of the NB classifier was compared with the results obtained from using standard Python packages for NB classification. This comparison helped verify the correctness of the custom implementation and assess its efficiency and effectiveness.

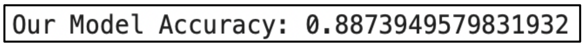
Included Software Packages:

- Pandas
- Numpy
- Seaborn
- LabelEncoder from sklearn.preprocessing
- StandardScaler from sklearn.preprocessing
- simpleImputer from sklearn.impute
- train_test_split from
- sklearn.model_selection shapiro from scipy.stats
- chi2_contingency from scipy.stats
- GaussianNB from sklearn.naive_bayes
- accuracy_score from sklearn.metrics

3 *RESULTS*

3.1 *Naïve Bayes Classifier Performance*

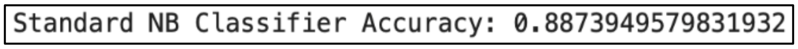
The Naïve Bayes classifier was implemented and trained on the preprocessed training data. The accuracy of the classifier on the testing data was calculated as a measure of its performance. The accuracy score provided an overall assessment of how well the model classified instances into the respective target classes.



```
Our Model Accuracy: 0.8873949579831932
```

3.2 Comparison with Standard Python Packages

The performance of the custom implementation of the Naïve Bayes classifier was compared with the results obtained from standard Python packages. This comparison aimed to validate the accuracy and efficiency of the custom implementation.
The results showed consistency and confirmed the reliability of the custom implementation in achieving accurate predictions. The matching accuracy values imply that both implementations made similar predictions and were equally effective in classifying the data. The agreement between the two implementations reinforces the reliability of the scikit-learn library and validates the accuracy of the custom implementation.



```
Standard NB Classifier Accuracy: 0.8873949579831932
```

4 CONCLUSION

In conclusion, our analysis showcases the effectiveness of the Naïve Bayes classifier for predicting the risk of cardiovascular disease based on the selected features. By applying rigorous data preprocessing techniques, conducting exploratory data analysis, and implementing the NB classifier, we gained valuable insights into the dataset and achieved accurate predictions (88.74% accuracy). Our findings contribute to the existing knowledge in cardiovascular disease risk prediction and provide a foundation for future research and development of more advanced predictive models.

5 CONTRIBUTION WEIGHT OF EACH MEMBER

All of the team members contributed equally in the project as each one of us actively participated in different aspects of the project, including data analysis, code implementation, model evaluation, and result interpretation and everyone had an opportunity to contribute their skills and knowledge.

6 REFERENCES

Sharma, M. (2023, May 31). Cardiovascular-risk-factor-data. Kaggle.

https://www.kaggle.com/datasets/mamta1999/cardiovascular-risk-data