**MACHINE LEARNING LAB**

**PROJECT REPORT**



**Submitted By:**

Mahnoor Aftikhar 22-CS-194

Minahil Hijab 22-CS-194

Section: A

**Submitted to:**

Lab Instructor, Syeda Aimal Fatima Naqvi

**Department of Computer Science,**

**HITEC University,**

# Heart Attack Risk Classifier

## Motivations

The impact of Machine Learning in healthcare is beyond what humans have achieved in centuries. In the recent years, Machine Learning has gained immense popularity in the precise and accurate classification of various medical conditions. Although several better techniques of Deep Learning have been introduced, the Machine Learning algorithms are still known for their high accuracies especially for classification problems. As a part of Machine Learning Course, this project aims to implement a Machine Learning model for the Heart Attack Risk Classification.

## Introduction

A Heart Attack, medically known as Myocardial Infarction is a condition in which the blood supply to the heart is blocked by a blood clot, usually formed by the buildup of fat, cholesterol and other substances in the arteries of the heart (coronary arteries) [1]. Heart disease is the leading cause of death for men, women, and people of most racial and ethnic groups [2]. One person dies every 33 seconds from cardiovascular disease. In 2022, 702,880 people died from heart disease. That's the equivalent of 1 in every 5 deaths [2] [3]. Heart disease cost about $252.2 billion from 2019 to 2020 [2]. Heart attacks are a leading cause of death globally, and early risk prediction can significantly reduce mortality rates.

## Objectives

* The goal of this project is to understand and analyze the real-life impacts of Machine Learning in health care systems.
* The project utilizes five different classification algorithms on the Heart Attack Risk Dataset to compare and analyse performance of different algorithms.
* The projects aims to visualize and understand the effects of various algorithms on a single dataset through different performance metrics.

## Data Analysis

## Reasons for Choosing this Dataset

Following are the main reasons for choosing this dataset in particular:

1. The dataset is fit for testing multi-class classification algorithms.
2. The dataset has good, relevant and balanced and diverse set of features best suited for classification problems.
3. The dataset is not an image dataset.
4. The dataset is neither too large, nor too small and therefore, can be easily trained on an average CPU (cost efficient).

## Dataset Description

The Heart Attack Risk Prediction Dataset [4] is a synthetic dataset designed to model real-life factors influencing heart attack risks. It contains 50,000 rows and 20 features, capturing various demographic, clinical, lifestyle, and diagnostic attributes. This dataset is ideal for exploring classification problems, feature engineering, and data visualization techniques. The further description is provided below:

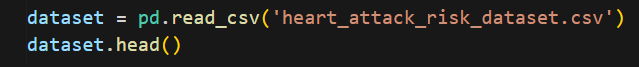
* Rows: 50,000 (each row represents a unique individual)
* Columns: 20 features related to heart attack risk
* Target Variable: Heart\_Attack\_Risk (Low, Moderate, High)

| **Feature Name** | **Description** | **Type** |
| --- | --- | --- |
| **Age** | Age of the individual (18–90 years). | Numeric (Integer) |
| **Gender** | Gender of the individual (Male/Female). | Categorical |
| **Smoking** | Smoking status (0 = Non-Smoker, 1 = Smoker). | Binary |
| **Alcohol\_Consumption** | Alcohol consumption status (0 = No, 1 = Yes). | Binary |
| **Physical\_Activity\_Level** | Level of physical activity (Low, Moderate, High). | Categorical |
| **BMI** | Body Mass Index (15–40, rounded to 2 decimal places). | Numeric (Float) |
| **Diabetes** | Diabetes status (0 = No, 1 = Yes). | Binary |
| **Hypertension** | Hypertension status (0 = No, 1 = Yes). | Binary |
| **Cholesterol\_Level** | Cholesterol level in mg/dL (150–300). | Numeric (Float) |
| **Resting\_BP** | Resting blood pressure in mmHg (90–180). | Numeric (Integer) |
| **Heart\_Rate** | Resting heart rate in beats per minute (60–130). | Numeric (Integer) |
| **Family\_History** | Family history of heart disease (0 = No, 1 = Yes). | Binary |
| **Stress\_Level** | Level of stress (Low, Moderate, High). | Categorical |
| **Chest\_Pain\_Type** | Type of chest pain (Typical, Atypical, Non-anginal, Asymptomatic). | Categorical |
| **Thalassemia** | Type of thalassemia (Normal, Fixed defect, Reversible defect). | Categorical |
| **Fasting\_Blood\_Sugar** | Fasting blood sugar level (0 = <120 mg/dL, 1 = >=120 mg/dL). | Binary |
| **ECG\_Results** | Results of electrocardiogram (Normal, ST-T abnormality, Left Ventricular Hypertrophy). | Categorical |
| **Exercise\_Induced\_Angina** | Angina caused by exercise (0 = No, 1 = Yes). | Binary |
| **Max\_Heart\_Rate\_Achieved** | Maximum heart rate achieved during stress tests (100–200 bpm). | Numeric (Integer) |
| **Heart\_Attack\_Risk** | Risk level for heart attack (Low, Moderate, High). | Categorical (Target) |

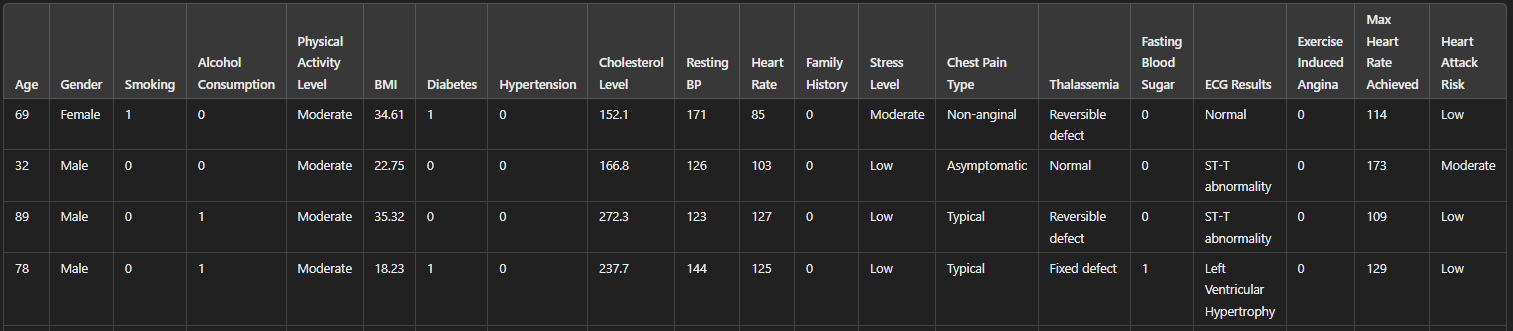
## Dataset Visualization & Code Description:

From the Python ‘head()’ function, we can obtain a general idea of how the dataset and its various features looks like:

### Code

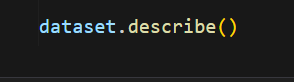


### Ouput



The class-wise count, mean, standard deviation, min, mac etcetera type details con be obtained by the Python function ‘describe()’as follows:

### Code



### Ouput

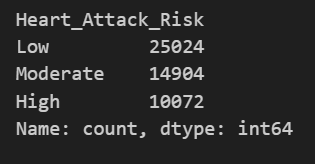


Further target variable values’ information can be visualized by Python function ‘value\_count()’ as follows:

### Code



### Ouput

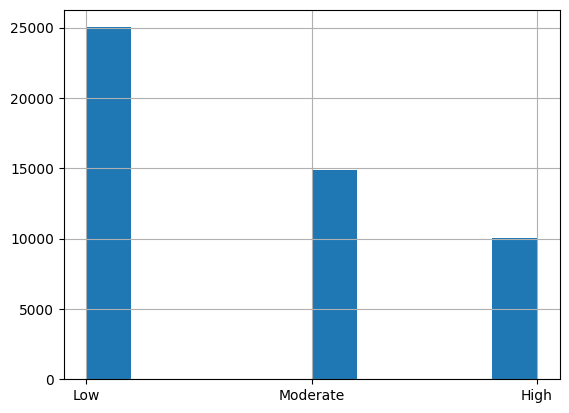


The graphical representation of above value count can be represented through a histogram obtained through the Python function ‘.hist()’

### Code

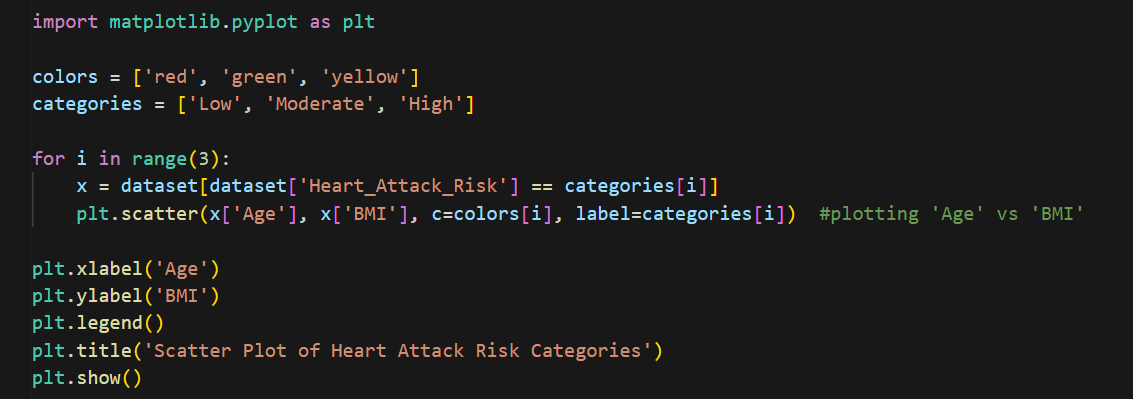


### Ouput

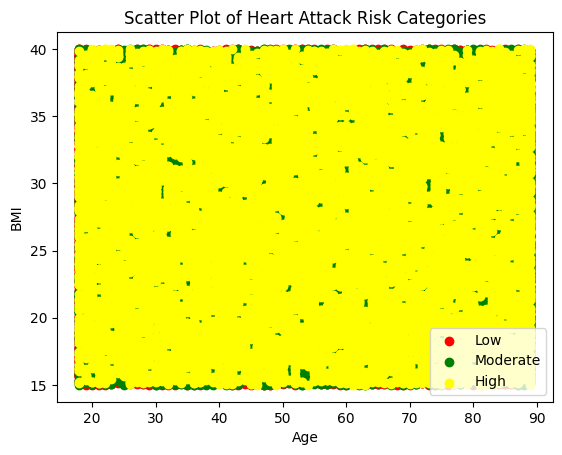


The 2D scatter plot of the classes Age vs BMI on target variable can be represented as follows:

### Code

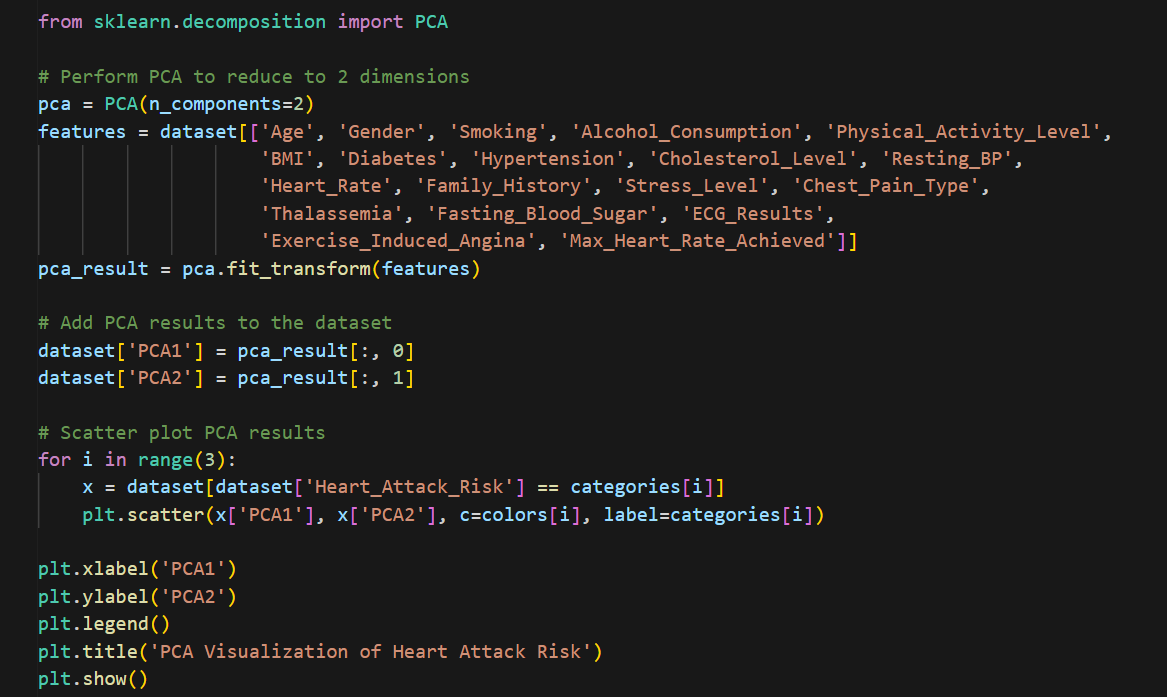


### Ouput

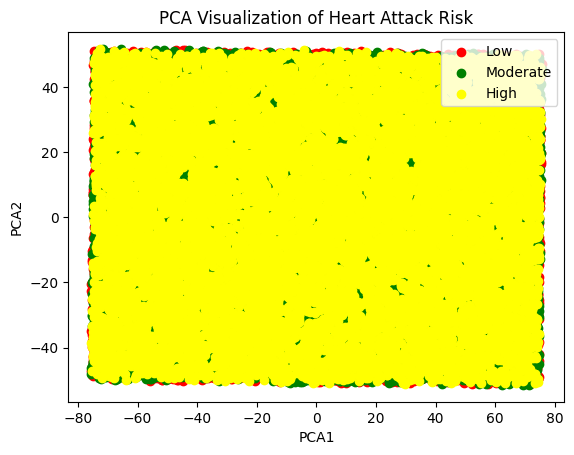


However, a 2D plot is not enough to visualize the overall multinomial dataset so an n-dimensional Scatter Plot can be obtained as follows:

### Code



### Ouput



## Methodology

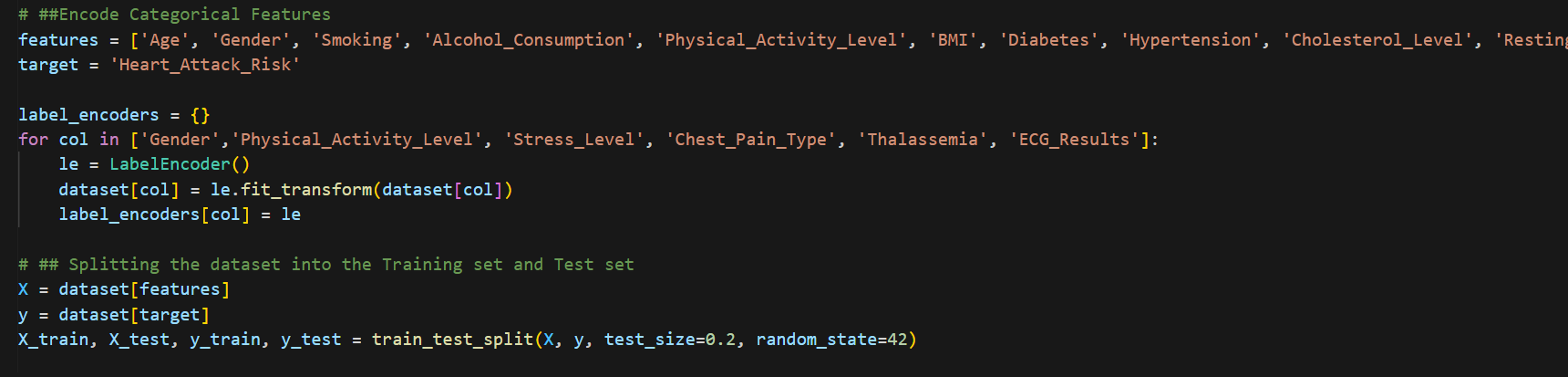
This project utilizes five different Supervised Learning Classification algorithms i.e. Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Logistics Regression, Decision Tree and Naïve Bayes to classify and predict heart attack risk among individuals with diverse demographic background and medical histories using the Heart Attack Risk Dataset [4]. This section provides the details of methodology of the project.

## Data Preprocessing

The dataset preprocessing steps like feature selection and dimensionality reduction were beyond the scope of this project. However, techniques like feature scaling and label encoding have been applied as follows:

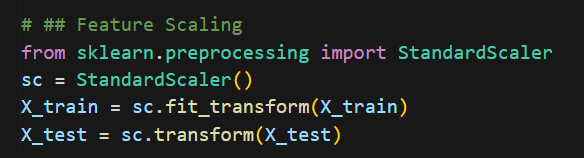
### Label Encoding

**Label Encoding** is a method used to convert categorical (non-numeric) data into numerical form, which is required by many machine learning algorithms. It assigns a unique integer value to each category in a feature.



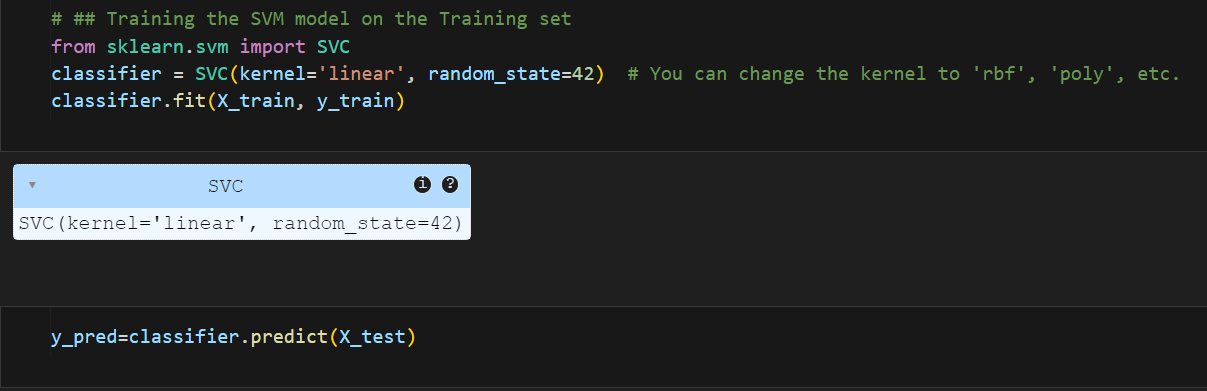
### Feature Scaling

**Feature scaling** is a technique used to normalize or standardize the range of independent variables (features) in a dataset. The goal is to ensure that all features contribute equally to the model's performance and avoid any feature dominating due to differences in scale.



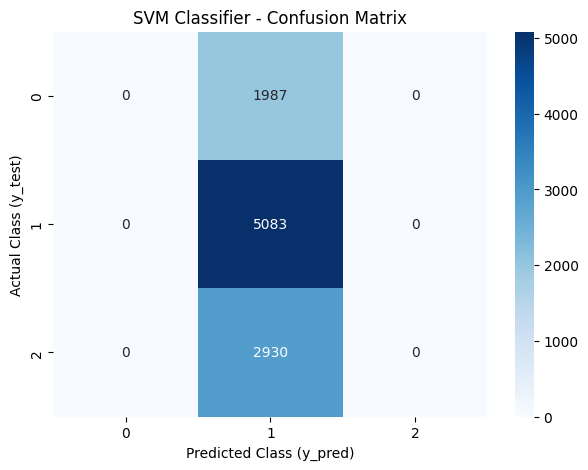
## SVM Classifier

An **SVM (Support Vector Machine) classifier is** a supervised machine learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates data points into different classes with the maximum margin between the classes. SVMs can handle linear and non-linear data using kernels to map inputs into higher-dimensional spaces. The following code of python was used to train the SVM model on the training dataset:



### Confusion Matrix

The Confusion Matrix for the SVM model’s predictions is as follows:



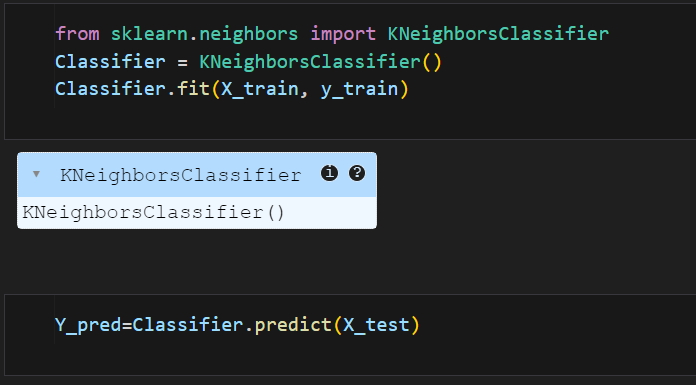
### Performance Metrics

The performance metrics of the SVM model are presented in the table below:

|  |  |
| --- | --- |
| **Accuracy** | 0.5083 |
| **Precision** | 0.17 |
| **Recall** | 0.33 |
| **F1-Score** | 0.22 |

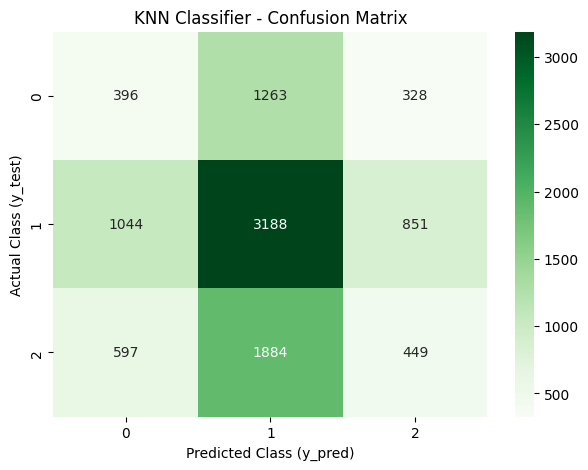
## KNN Classifier

The **K-Nearest Neighbors (KNN)** classifier is a supervised machine learning algorithm used for classification and regression tasks. It predicts the class of a data point based on the majority class of its KKK nearest neighbors in the feature space, determined by a distance metric (e.g., Euclidean distance). KNN is simple, intuitive, and works well for smaller datasets. The following code of python was used to train the KNN model on the training dataset:



### Confusion Matrix

The Confusion Matrix for the KNN model’s predictions is as follows:



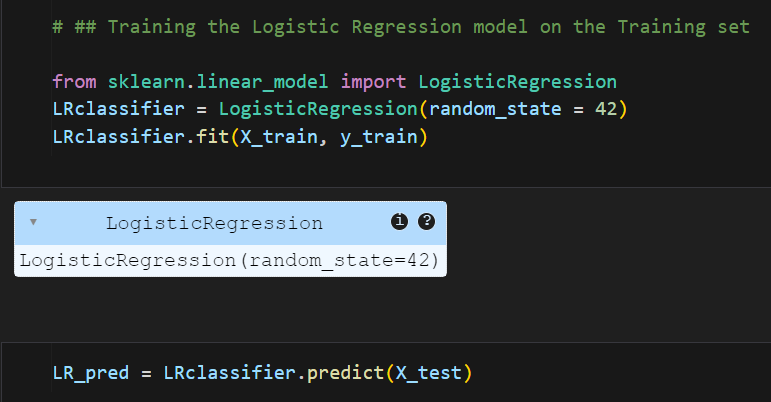
### Performance Metrics

The performance metrics of the KNN model are presented in the table below:

|  |  |
| --- | --- |
| **Accuracy** | 0.4033 |
| **Precision** | 0.32 |
| **Recall** | 0.33 |
| **F1-Score** | 0.32 |

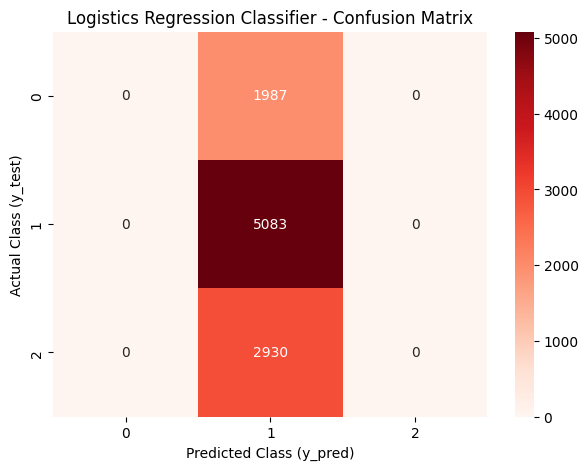
## Logistic Regression Classifier

**Logistic Regression** is a supervised machine learning algorithm used for binary and multi-class classification tasks. It models the relationship between input features and the probability of a class label using a logistic (sigmoid) function. It predicts probabilities and classifies data based on a threshold, making it effective for linearly separable datasets. The following code of python was used to train the Logistic Regression model on the training dataset:



### Confusion Matrix

The Confusion Matrix for the Logistic Regression model’s predictions is as follows:



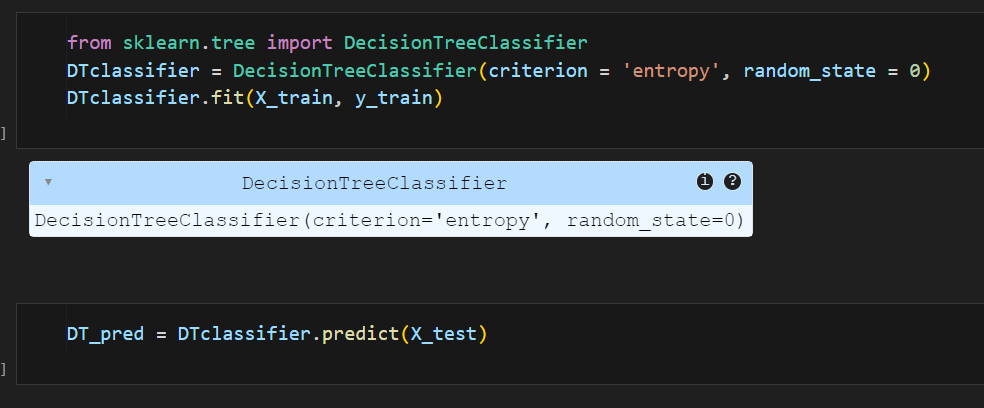
### Performance Metrics

The performance metrics of the Logistic Regression model are presented in the table below:

|  |  |
| --- | --- |
| **Accuracy** | 0.5083 |
| **Precision** | 0.17 |
| **Recall** | 0.33 |
| **F1-Score** | 0.22 |

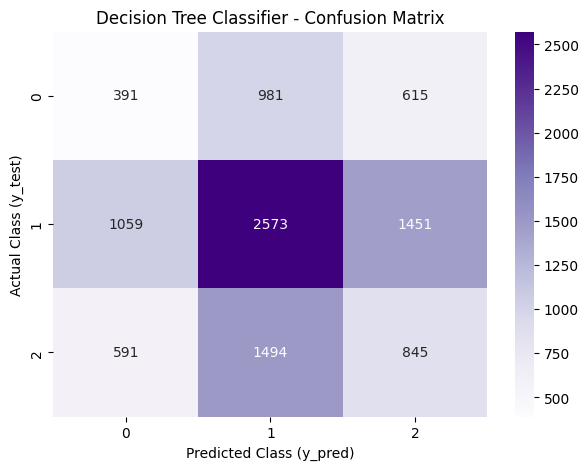
## Decision Tree Classifier

A **Decision Tree classifier** is a supervised machine learning algorithm used for classification and regression tasks. It works by splitting the data into subsets based on feature values, forming a tree-like structure where each node represents a decision rule. The tree predicts outcomes by traversing from the root to a leaf node based on feature conditions. The following code of python was used to train the Decision Tree model on the training dataset:



### Confusion Matrix

The Confusion Matrix for the Decision Tree model’s predictions is as follows:



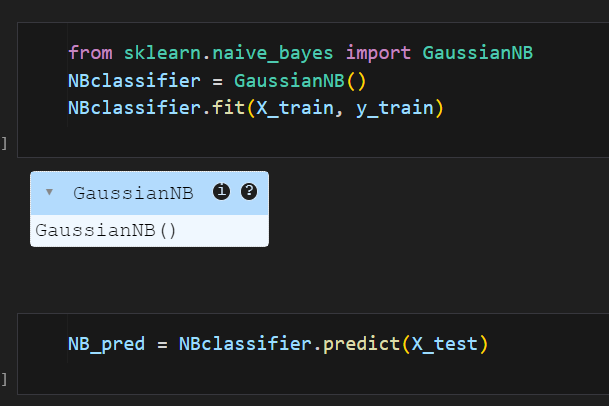
### Performance Metrics

The performance metrics of the Decision Tree model are presented in the table below:

|  |  |
| --- | --- |
| **Accuracy** | 0.3809 |
| **Precision** | 0.33 |
| **Recall** | 0.33 |
| **F1-Score** | 0.33 |

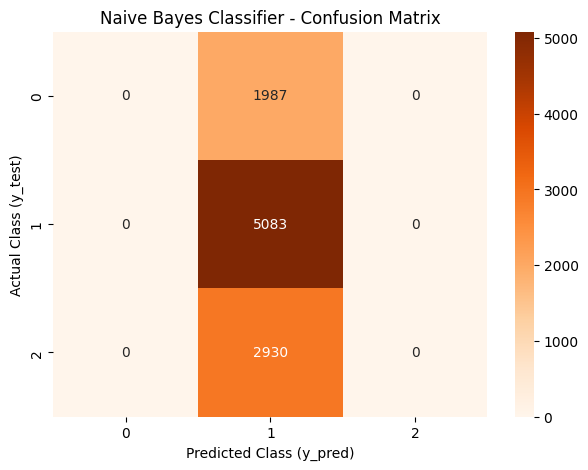
## Naïve Bayes Classifier

Naive Bayes is a supervised machine learning algorithm used for classification tasks. It is based on Bayes' Theorem and assumes that features are independent of each other (a "naive" assumption). Despite this simplification, it performs well for tasks like text classification and spam detection due to its efficiency and simplicity. The following code of python was used to train the Naïve Bayes model on the training dataset:



### Confusion Matrix

The Confusion Matrix for the Naïve Bayes model’s predictions is as follows:



### Performance Metrics

The performance metrics of the Naïve Bayes model are presented in the table below:

|  |  |
| --- | --- |
| **Accuracy** | 0.5083 |
| **Precision** | 0.17 |
| **Recall** | 0.33 |
| **F1-Score** | 0.22 |

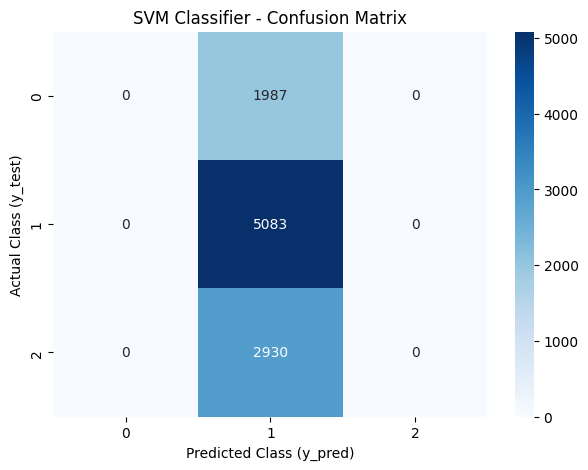
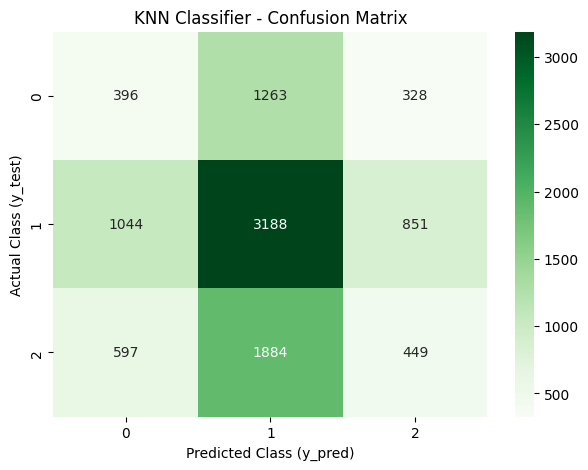
## Results and Discussion

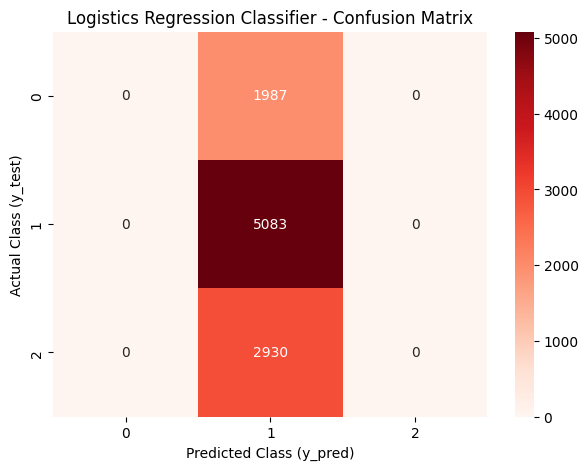
This section presents a comparative analysis of all five classifiers used in the heart attack risk prediction process.

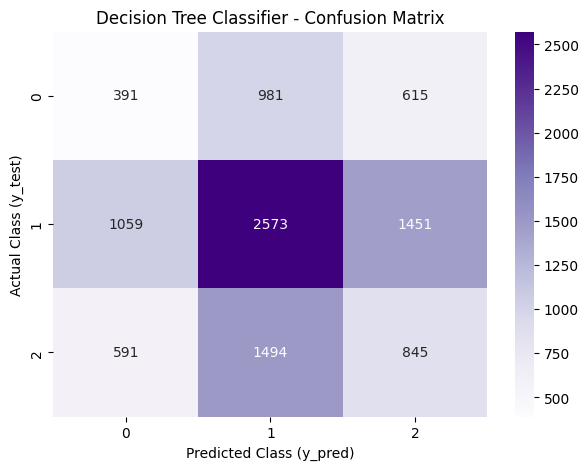
## Result Visualization

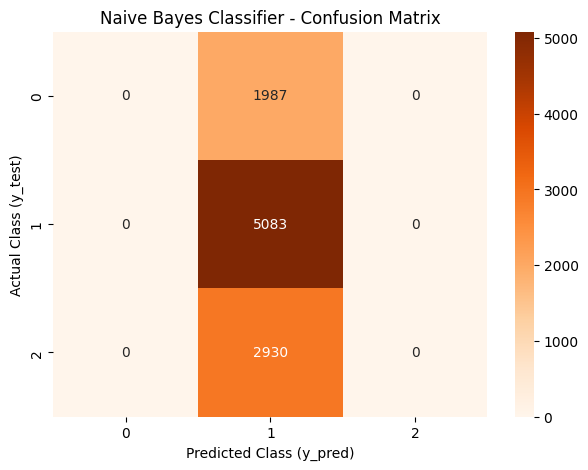
The Results of all five classifiers can be visualized by the confusion matrices provided below:

### Confusion Matrices







## Comparison of Algorithms

This section presents the comparative analysis of performance metrics of all five classifiers used in the project. The table below shows accuracy, precision, recall and f1 scores of all five models. The highest accuracy achieved is 50.83%. Overall performance of all five models is not up to the mark. The possible reasons for this poor performance are discussed in the next section.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Performance Metrics | SVM | KNN | Logistics Regression | Decision Tree | Naïve Bayes |
| Accuracy | 0.5083 | 0.4033 | 0.5083 | 0.3809 | 0.5083 |
| Precision | 0.17 | 0.32 | 0.17 | 0.33 | 0.17 |
| Recall | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 |
| F1-Score | 0.22 | 0.32 | 0.22 | 0.33 | 0.22 |

### Reasons for Poor Model Performance

Following are the possible reasons for the model’s poor performance:

1. Lack of Feature Engineering
   * The use of advanced feature engineering techniques such as dimensionality reduction and feature selection were beyond the scope of this project.
2. Dataset Limitation
   * The utilized dataset is very small and might have resulted in poor model generalizability.
3. Underfitting
   * Due to the several reasons mentioned above and below, the model is clearly suffering from underfitting.
4. Class Imbalance Problem
   * The class misbalancing can be clearly viewed from the dataset visualization.

### Ways to Improve Model Performance

Possible ways to improve model performance are:

1. Use of Advanced Feature Engineering Techniques

* The use of advanced feature engineering techniques such as dimensionality reduction and feature selection can enhance the model’s performance.

1. Changing Dataset

* Using a relatively large dataset with improvised features can result in better model generalization.

1. Optimization:
   * + Techniques like Hyperparameter Tuning, optimizing k and the distance metric in KNN, tuning the C, kernel, and gamma parameters in SVM and max\_depth, min\_samples\_split, and min\_samples\_leaf decision tree, experimenting with C (inverse of regularization strength) and penalty type (l1, l2) in Logistic Regression and checking for different variants (Gaussian, Multinomial, Bernoulli) in Naïve Bayes can signifacntly improved performance of each model.
2. Data Augmentation
   * Handling Class Imbalance through oversampling techniques (e.g., SMOTE), undersampling for the majority class and class weighting in models (e.g., class\_weight='balanced' for SVM and Logistic Regression) can improve model performance.

## Conclusion

The Impact of Machine Learning in healthcare systems is huge. The proposed project shows the great potential of Machine Learning Algorithms in the field of AI in Medicine. The project successfully developed and evaluated multiple machine learning models, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Logistic Regression, and Naive Bayes, to classify heart attack risk based on diverse health-related features. Through rigorous preprocessing, feature selection, and model evaluation, valuable insights were obtained regarding the strengths and weaknesses of each algorithm for this classification task.

## Future Prospects

While the results of this project are promising, there are several directions for future work:

1. **Incorporation of Advanced Models**
   * Explore deep learning techniques, such as neural networks, to capture more complex patterns in the data.
   * Investigate ensemble methods like Random Forests, Gradient Boosting, or XGBoost for improved performance.
2. **Larger and Diverse Datasets**
   * Expand the dataset to include more diverse populations and additional health features for generalizability.
   * Incorporate real-time data collection methods, such as wearable device metrics, to enhance predictive accuracy.
3. **Addressing Class Imbalance**
   * Implement techniques like SMOTE (Synthetic Minority Oversampling Technique) to handle any imbalance in the target class, ensuring fair evaluation of model performance.
4. **Feature Enhancement**
   * Integrate more domain-specific features, such as genetic factors, medication history, and lifestyle details.
   * Perform automated feature selection using advanced techniques to further optimize the model.
5. **Deployment and Usability**
   * Develop a user-friendly application or dashboard for healthcare providers to make predictions accessible.
   * Ensure the model complies with ethical and legal standards, emphasizing patient data privacy and model transparency.

## References

[1] M. Clinic. "Heart Attack." <https://www.mayoclinic.org/diseases-conditions/heart-attack/symptoms-causes/syc-20373106> (accessed.

[2] CDC., "National Center for Health Statistics Mortality Data on CDC WONDER," ed: US Department of Health and Human Services, CDC, 2021.

[3] S. S. Martin *et al.*, "2024 heart disease and stroke statistics: a report of US and global data from the American Heart Association," *Circulation,* vol. 149, no. 8, pp. e347-e913, 2024.

[4] kaggle.com. "Heart Attack Risk Dataset." <https://www.kaggle.com/datasets/arifmia/heart-attack-risk-dataset/data> (accessed.