Heart Attack Analysis and Prediction Report

Machine Learning Project

2023/24

Index

Introduction

Heart attacks, also called myocardial infarctions, are among the main reasons for death around the globe. Early detection and precise heart attack prediction may significantly improve patients' results and also lower rates of mortal deaths. The purpose of the current work is to explore how we can apply artificial intelligence methods such as machine learning in order to identify those at risk from having a heart attack. Our goal with this project involves machine learning applications together with techniques capable of forecasting the propensity to have heart attacks.

We will use the Heart Attack Analysis & Prediction Dataset accessible on Kaggle, it contains a full set of medical data from patients. It includes factors such as age, gender, blood pressure, cholesterol levels and other essential health parameters. These factors are used for making predictions while the target variable determines if someone has already had a heart attack.

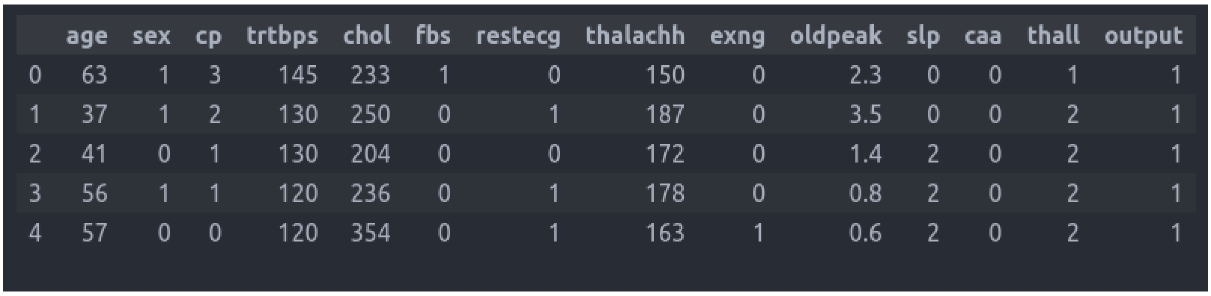
In this project, we used the Scikit-learn framework for developing and evaluating our machine learning models. Scikit-learn is a powerful and versatile library in Python, designed for data mining and data analysis. It provides a wide range of tools for model selection, preprocessing, and evaluation.

A smart project needs a multitasking approach. One side, it's necessary to process the acquired data successfully, using it for exploratory reasons. Knowing the variation of all parameters is important. Noteworthy is that EDA helps in knowing what are the patterns and possible risk factors for heart attacks. Therefore, missing values and outliers among other preprocessing issues will be handled before fitting the model.

Additionally, we evaluate the models by using appropriate metrics to ensure reliability. Finally the models will be interpreted such that we can provide insights that can have a positive influence on healthcare topics.

The Dataset

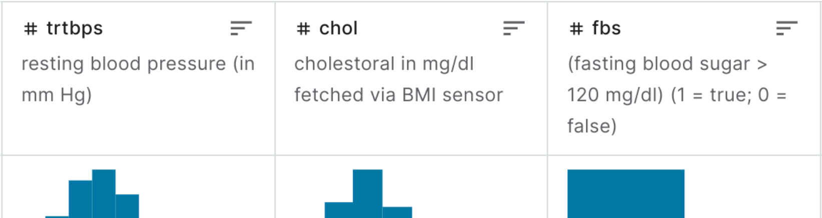
The data was provided by kaggle under the following link: https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset?select=heart.csv.
The dataset contains 303 samples with 14 features each:



	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

- Age: Age of the patien
- Sex: Sex of the patient(1 = male, 0 = female)
- cp: Chest Pain type:
    - 0: Typical angina
    - 1: Atypical angina
    - 2: Non-anginal pain
    - 3: Asymptomatic
- trtbps: Resting blood pressure (in mm Hg)
- chol: Cholesterol in mg/dl fetsched via BMI sensor
- fbs: (Fasting blood sugar > 120 mg/dl) (1 = true, 0 = false)
- restecg: Resting electrocardiographic results
    - 0: Normal
    - 1: Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
    - 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
- thalachh : maximum heart rate achieved
- exang: Exercise induced angina (1 = yes, 0 = no)
- oldpeak: ST depression induced by exercise relative to rest
- slope: Slope of the peak exercise ST segment (0-2)
- ca: Number of major vessels (0-3)
- thal: Thalassemia (0-3)
- output : 0= less chance of heart attack 1= more chance of heart attack

These features describe a lot of different symptoms of the body and are giving convincing insights on the actual health-state of the patient. Therefore we want to show that it is possible to predict the risk of a heart attack by only using these attributes.

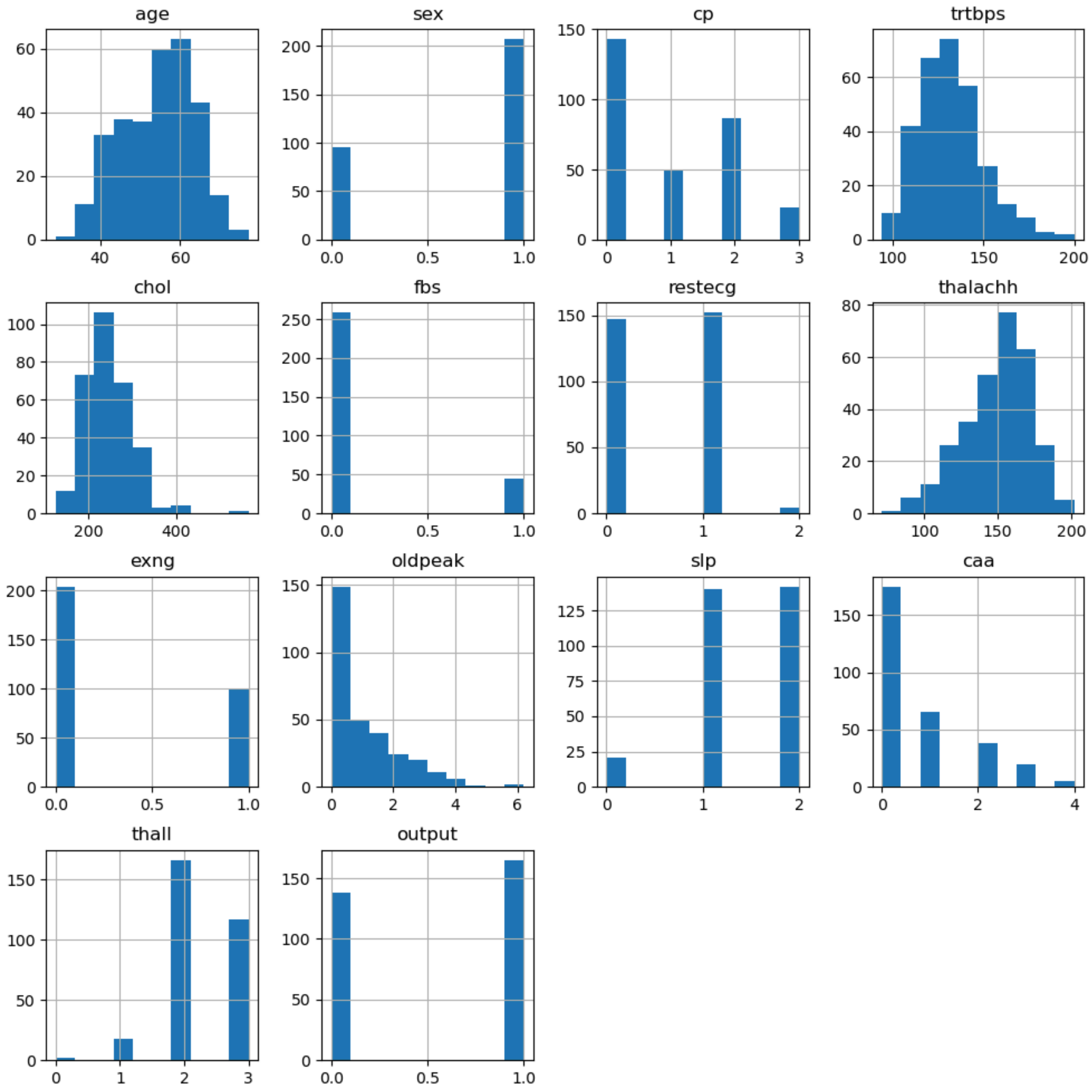


# trtbps	# chol	# fbs
resting blood pressure (in mm Hg)	cholestoral in mg/dl fetched via BMI sensor	(fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

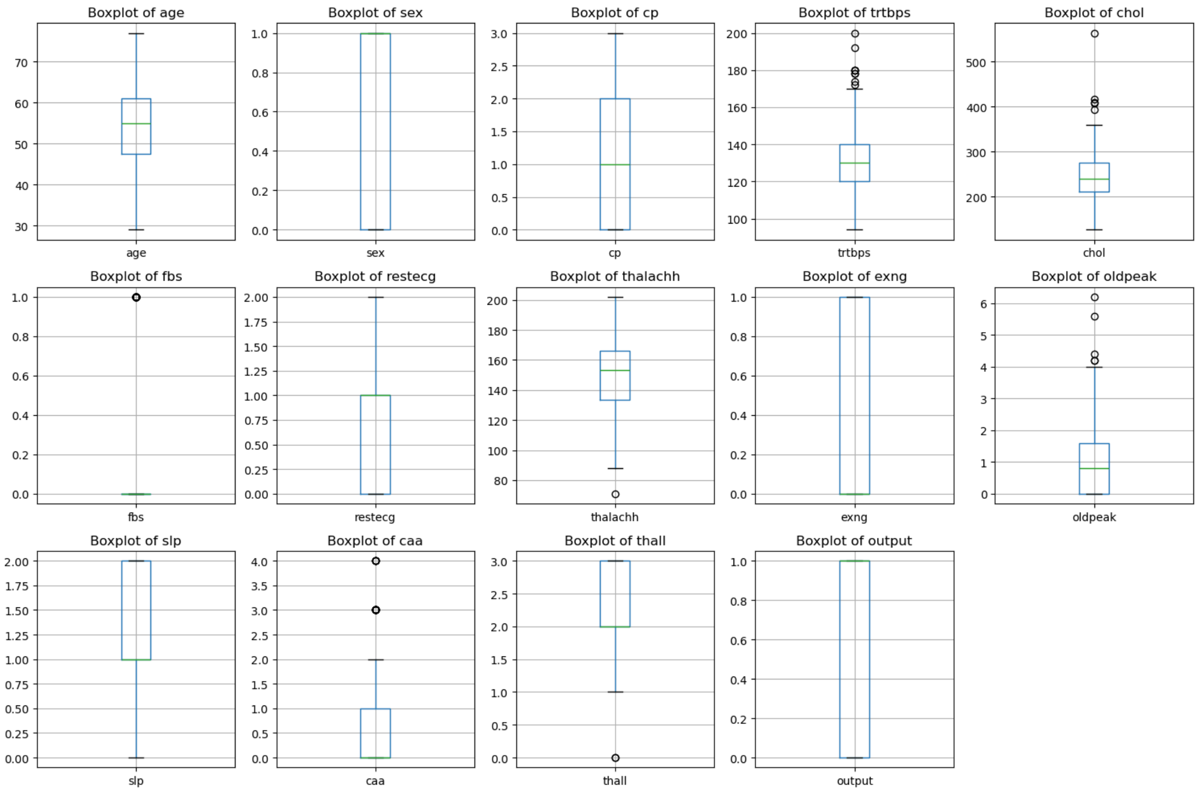
Exploratory Data-Analysis

The first step when working with a dataset is to preprocess and to manipulate the data such that it fits the model. To do that one has to know how the dataset looks like. Therefore, it is important to find out how the data is distributed, what are the minimum or maximum values for different types of attributes or if there exist some outliers. There are a lot more characteristics which are important to analyze. Therefore we decided to plot the data in histograms, because then it is easier to analyze the data.

The following image shows the histogram of how the different attributes are distributed.

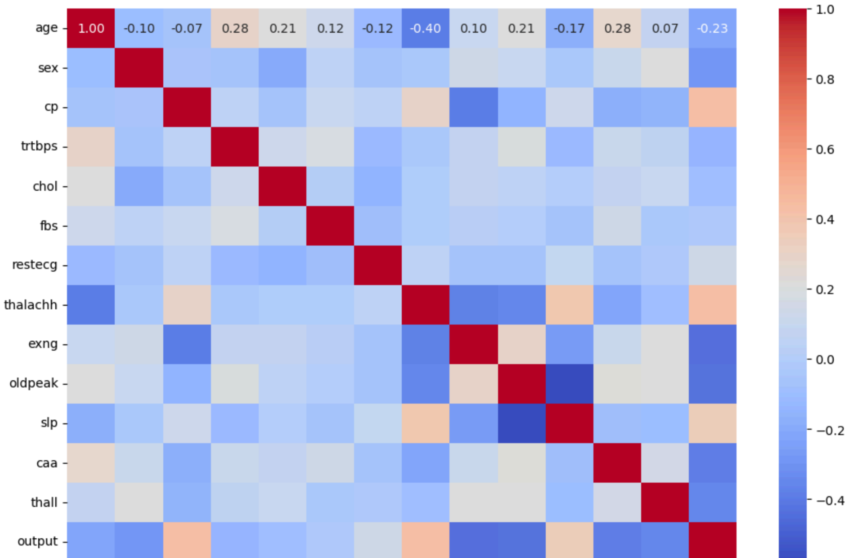


We can clearly see the binary data like sex, fbs, exng or also the output. On the other hand we have a left-skewed curve when we look at the attribute age.Here we can see that most of the people in the dataset are around 60. But as mentioned before to detect outliers or to present the range of the values there are some better plots as for example the boxplot.

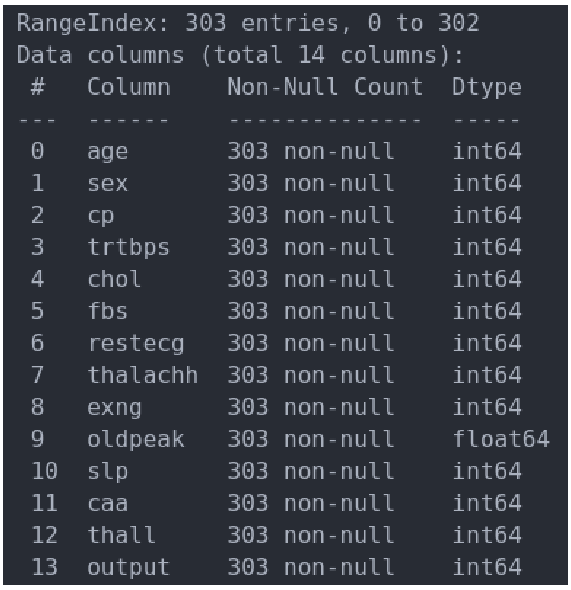


In this box plot we can see that the ranges of the different features differ a lot therefore, we need to scale the data when creating the test set such that we do not get results which are falsified, because this can lead to loss of expressiveness. Therefore we decided to scale the dataset such that the data lies between 0 and 1. Moreover, we have decided to not delete outliers since the dataset is already small and in this case outliers might be important to decide on whether a person is at risk of a heart attack.

It is also often interesting to see if and which attributes correlate.



Data-Preprocessing



```
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
```

Handling Missing Values

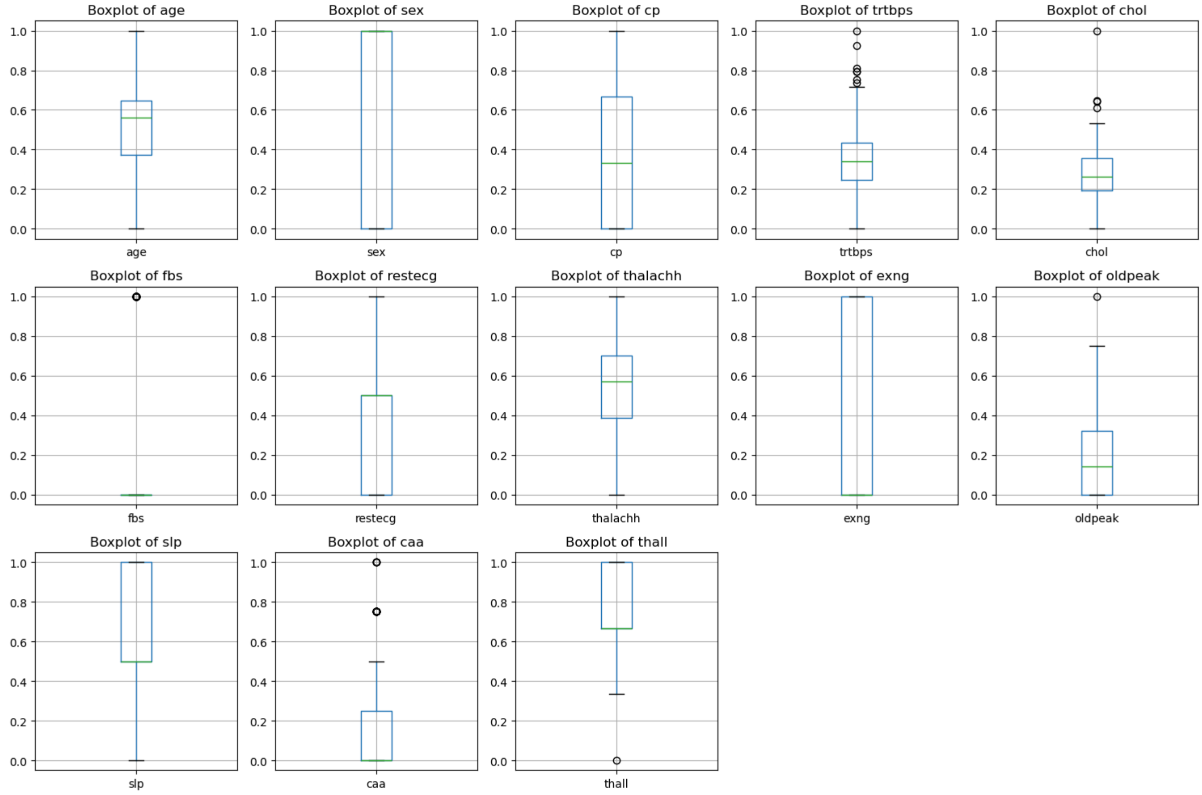
This dataset doesn’t contain missing values. So there is no need to replace any.

Splitting the Data

The dataset is split into training and testing sets with a ratio of 80:20. The training set is used to train the models, while the testing set is used to evaluate their performance.

Scaling numerical features

Numerical features are scaled using the MinMaxScaler to ensure all values fall within the same range, enhancing the performance of machine learning models. Here is the boxplot of the attributes to see the distribution after applying a MinMaxScaler to each attribute.



Oversampling

Since the two classes are evenly distributed there is no minority class. Hence, we didn't use any over- or undersampling techniques like SMOTE.

Model Training

We have used several models to predict heart attacks, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and Decision Tree (DT). Furthermore, we have used Hyperparameter tuning through GridSearchCV to fine tune the models and find the best parameters. All models were trained through a similar process with differences in their Hyperparameters.

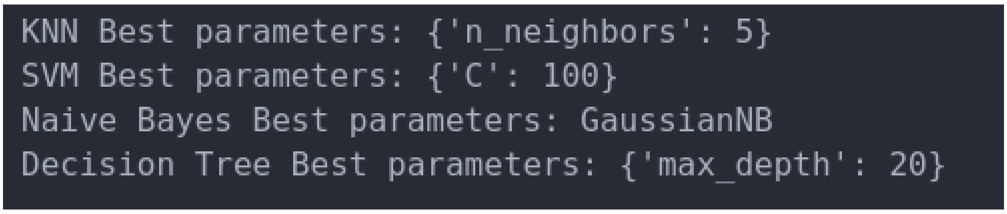
The training process
For every model we define a parameter grid. This is a dictionary that specifies the hyperparameters to be tuned and their respective ranges.

We used k-fold cross-validation within the grid search, which runs through all hyperparameters, to evaluate the performance of different combinations of hyperparameters. By incorporating k-fold cross-validation within the grid search we ensure a more reliable evaluation of the model's performance, reducing the risk of overfitting and ensuring that the chosen hyperparameters generalize well to new data.

The hyperparameters for the different models were as follows:

- KNN: number of neighbors (3, 5, 7, 9, 11)
- SVM: penalty parameter C (0.1, 1, 10, 100)
- NB: model (BernoulliNB, GaussianNB, MultinomialNB)
- DT: depth of tree (None, 10, 20, 30, 40, 50)

Then the models were trained with sci-kit learn GridSearchCV function and we received the following results:



```
KNN Best parameters: {'n_neighbors': 5}
SVM Best parameters: {'C': 100}
Naive Bayes Best parameters: GaussianNB
Decision Tree Best parameters: {'max_depth': 20}
```

With these hyperparameters, generated through cross-validation on the training set, we were able to train the models with the best settings.

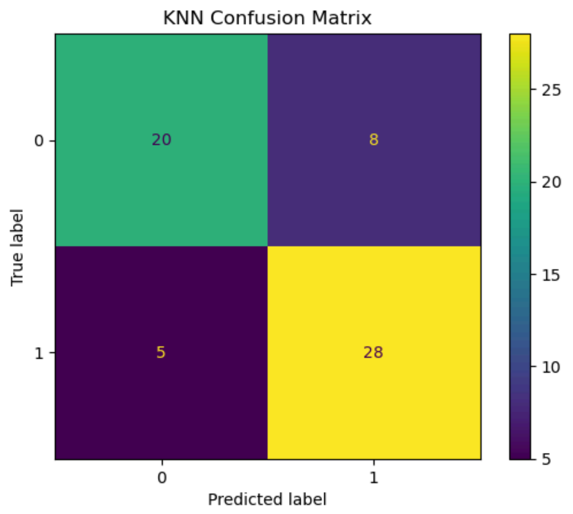
Model Evaluation

Different metrics can be used to evaluate the performance of a model. It is important to know what these metrics actually tell us before looking at the performance of our models.

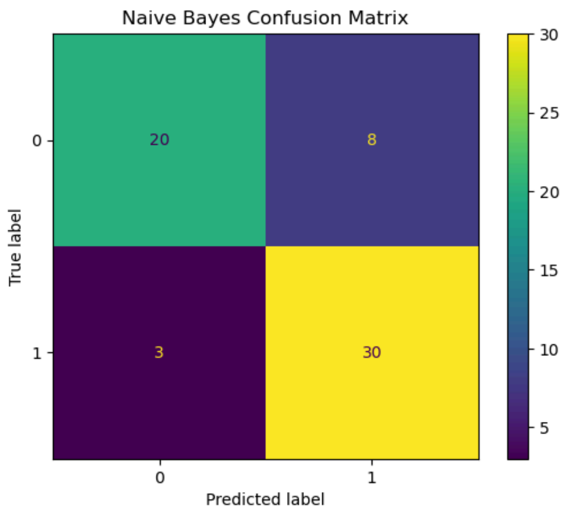
Classification Metrics:

- **Accuracy**: ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset.
- **F1 Score:** provides a single metric that balances both precision (ability of the model not to label as positive a sample that is negative) and recall (ability of the model to find all positive samples) across all classes.
- **ROC AUC Score:** area under the ROC curve. A higher ROC AUC score (closer to 1) indicates better discrimination between positive and negative instances.

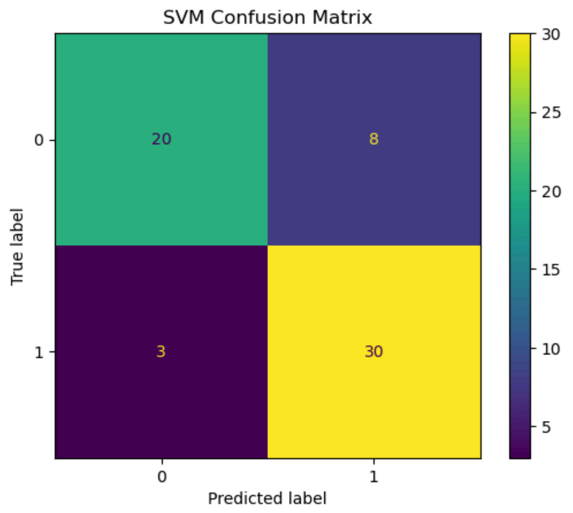
The **confusion matrix** is a very nice way to visualize a detailed breakdown of the classification performance by showing the true positives, false positives, true negatives and false negatives at once.


KNN Confusion Matrix

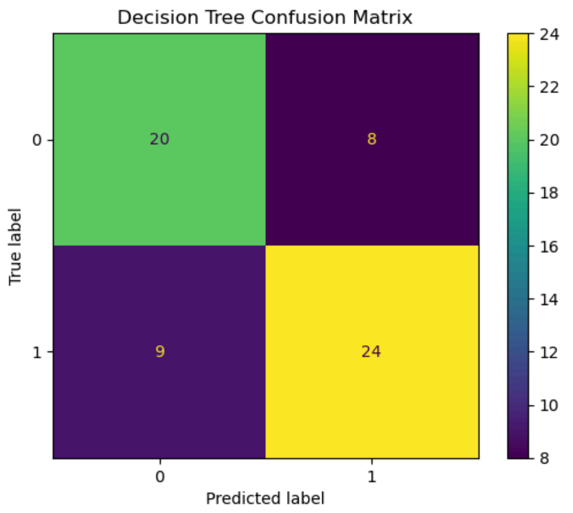
KNN Accuracy: **0.7869**
KNN Weighted F1 Score: **0.7855**
KNN ROC AUC Score: **0.8582**


Naive Bayes Confusion Matrix

Naive Bayes Accuracy: **0.8197**
Naive Bayes Weighted F1 Score: **0.8172**
Naive Bayes ROC AUC Score: **0.8799**


SVM Confusion Matrix

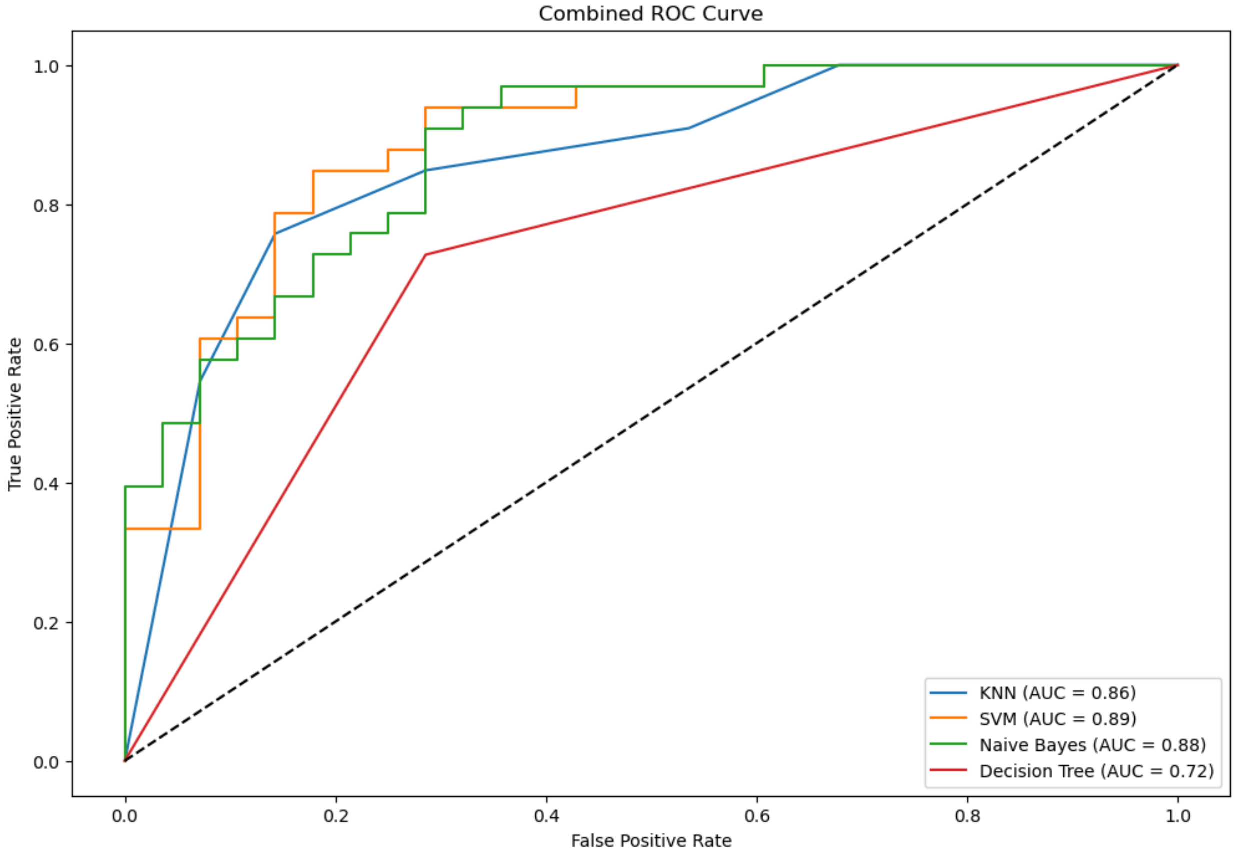
SVM Accuracy: **0.8197**
SVM Weighted F1 Score: **0.8172**
SVM ROC AUC Score: **0.8885**


Decision Tree Confusion Matrix

Decision Tree Accuracy: **0.7213**
Decision Tree Weighted F1 Score: **0.7216**
Decision Tree ROC AUC Score: **0.7208**

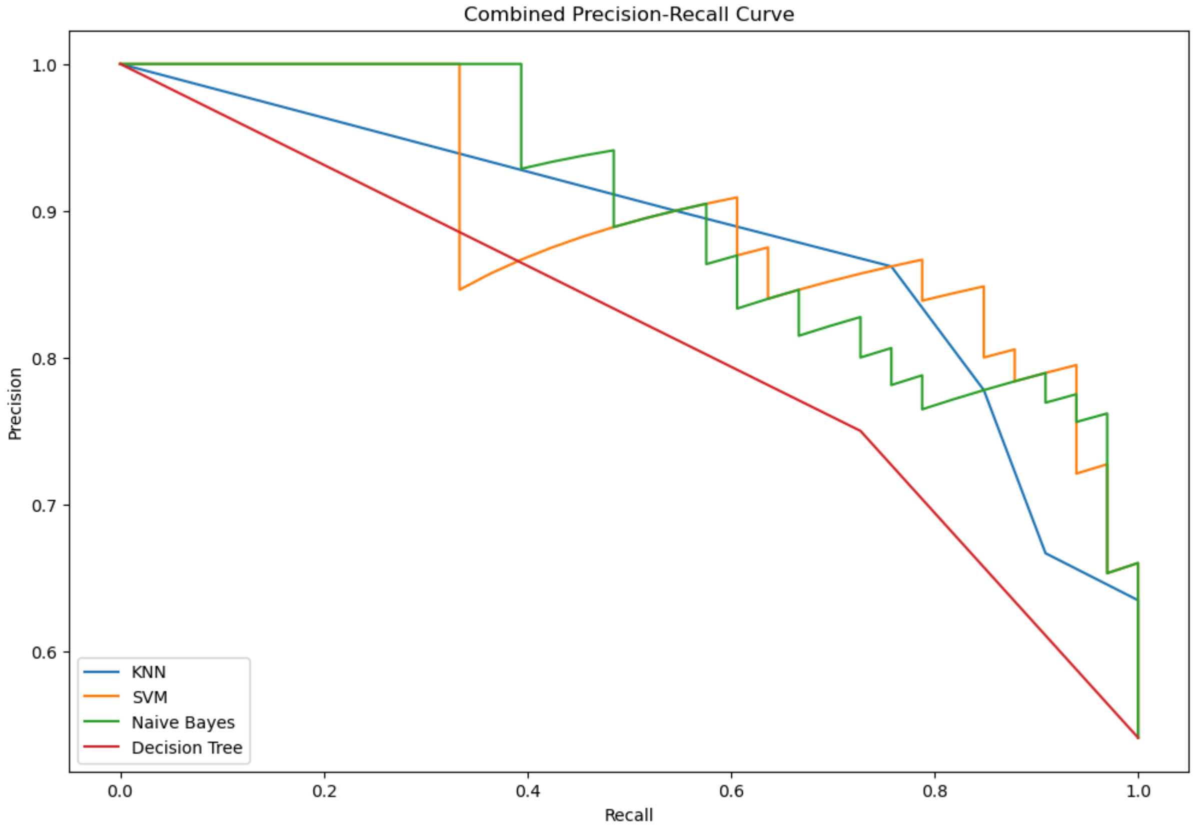
Looking at the results, SVM emerges as the top performer across all metrics, suggesting it is the most preferable model for predicting heart attacks in this dataset. Naive Bayes also performs very well and could be a practical alternative, especially for a simpler implementation. KNN shows competitive performance but slightly lags behind SVM and Naive Bayes in terms of weighted F1 and ROC AUC. Decision Tree does not generalize as effectively based on its lower scores in weighted F1 and ROC AUC.

The **ROC Curve** plots the true positive rate (TPR) against the false positive rate (FPR). The closer a model's score is to the point (0,1) the better. It is a good tool to plot the performance of multiple models against each other.

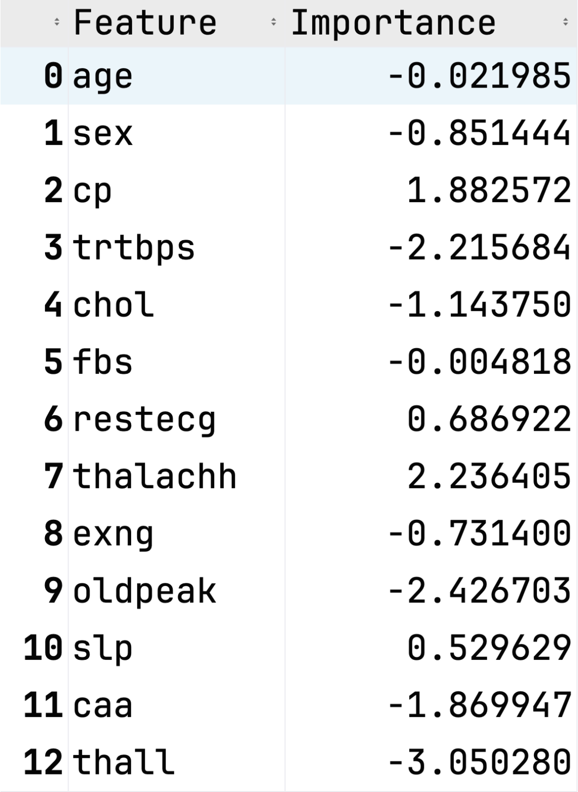


The ROC curve visually confirms the ROC AUC scores obtained earlier, reinforcing that SVM gives the highest discriminatory ability among the models evaluated for predicting heart attacks in this dataset. Naive Bayes also has a strong performance, being a practical alternative. Meanwhile, KNN and Decision Tree show slightly lower performance, aligning with their ROC AUC scores.

The **Precision-Recall Curve** plots precision against recall for different threshold values used by the model to make predictions. A higher precision at a given recall indicates fewer false positives, while a higher recall at a given precision indicates fewer false negatives.



Feature Importances for SVM:
Understanding which input features are most important in predictive models is very important for understanding how these models make decisions. Analyzing feature importances provides valuable insights into which features influence SVM's predictions.



	Feature	Importance
0	age	-0.021985
1	sex	-0.851444
2	cp	1.882572
3	trtbps	-2.215684
4	chol	-1.143750
5	fbs	-0.004818
6	restecg	0.686922
7	thalachh	2.236405
8	exng	-0.731400
9	oldpeak	-2.426703
10	slp	0.529629
11	caa	-1.869947
12	thall	-3.050280

Features such as maximum heart rate achieved (thalachh) and chest pain type (cp) emerge as significant indicators in predicting the likelihood of a heart attack. On the other hand, variables like thalassemia (thall) and ST depression induced by exercise (oldpeak) show strong negative influences, suggesting their importance in predicting a lower risk of heart attack.

Conclusion

This project successfully demonstrated the application of machine learning models for predicting heart attack risk using medical data. SVM emerged as the top-performing model, followed by Naive Bayes and KNN, with Decision Tree showing the least effectiveness. The analysis provided insights into the importance of specific health parameters in predicting heart attacks, which could be valuable for healthcare professionals.