Heart Attack Prediction Using Machine Learning

Angela Nikolova

Summary

This project focuses on predicting heart attack from patient data. Using a dataset with 13 features such as age, cholesterol, chest pain type, and others, various classification models were trained and evaluated. The one that performed best in terms of accuracy and F1-score was selected for deployment using a Gradio interface for user interaction.
`https://github.com/nikolovaaangela/MLDM-Project`

1 Introduction

Heart attack is one of the leading causes of death worldwide. Early detection and preventive measures can greatly reduce the risk. This project aims to build an accurate model that can classify whether a patient is likely to have heart attack, based on routine clinical data.

2 Dataset Description

The dataset used in the project contains 303 records with 13 features and 1 target variable.

Features

- Age
- Sex
- Chest Pain Type (cp)
- Resting Blood Pressure (trestbps)
- Cholesterol (chol)
- Fasting Blood Sugar (fbs)
- Resting ECG (restecg)
- Max Heart Rate (thalach)
- Exercise Induced Angina (exang)
- ST depression (oldpeak)
- Slope of ST (slope)
- Number of major vessels (ca)
- Thalassemia (thal)

Target Variable

Target: Presence (1) or absence (0) of heart disease.

3 Data Preprocessing

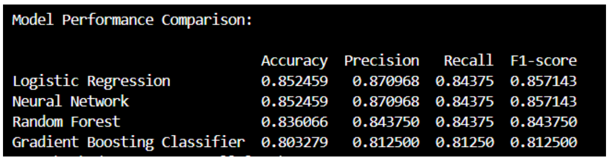
- Checked for missing values (none found).
- Standardized features using `StandardScaler`.
- Split dataset into 80% training and 20% testing sets.

4 Modeling and Evaluation

The following models were trained:

- Logistic Regression
- Random Forest Classifier
- Gradient Boosting Classifier
- Neural Network (ANN using Keras)

Their performance was evaluated and compared in the following dataframe, comparing their Accuracy, Precision, Recall and F1 Score which was later a crucial factor for determining the best-performing model:



```
Model Performance Comparison:

                              Accuracy  Precision   Recall  F1-score
Logistic Regression           0.852459   0.870968  0.84375  0.857143
Neural Network                0.852459   0.870968  0.84375  0.857143
Random Forest                 0.836066   0.843750  0.84375  0.843750
Gradient Boosting Classifier  0.803279   0.812500  0.81250  0.812500
```

Figure 1: Model Performance Comparison

5 Feature Importance

To determine which clinical parameters most influence heart attacks prediction, we extracted feature importance from both Gradient Boosting Classifier and Logistic Regression trained models.

The results show that while both models identify `ca`, `thal`, `oldpeak`, and `cp` as important predictors, their rankings vary:

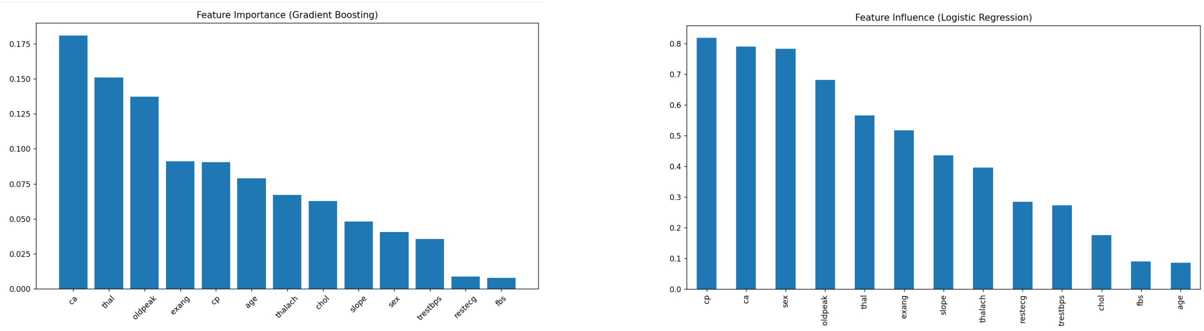
- **Gradient Boosting**, being a non-linear model, captures complex feature interactions and assigns higher importance to `ca`, `thal`, and `oldpeak`.

- **Logistic Regression**, a linear model, highlights `cp` and `sex` more important, which probably have a stronger linear correlation with heart disease.

- Features like `thalach`, `slope`, and `exang` show mid-level importance in both models, indicating a consistent predictive value.

In general, this comparison displays the strength of certain clinical features while also evaluating how the model type affects feature interpretation.

The top contributing features were:



(a) Gradient Boosting

(b) Logistic Regression

Figure 2: Feature importance comparison between Gradient Boosting and Logistic Regression

- cp
- thalach
- oldpeak
- ca
- slope

6 Confusion Matrix

According to the F1 scores, the Logistic Regression turned out to be the most efficient model for our data. Therefore, the confusion matrix shows that our model correctly identified 39 out of 45 patients who had a heart attack (true positives), and only misclassified 2 (false negatives), making it highly accurate.

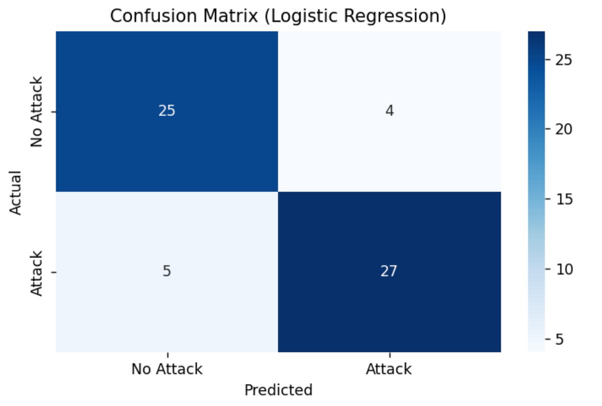


Figure 3: Confusion Matrix (LR)

7 ROC curves

The Receiver Operating Characteristic curves provide a comparative evaluation of the classification performance of all implemented models. As illustrated, all models (Logistic

Regression, Random Forest, Gradient Boosting, and Neural Network) perform significantly better than random guessing (represented by the diagonal dashed line). Each model achieves a high true positive rate while maintaining a low false positive rate across most thresholds. Notably, Logistic Regression and Neural Network demonstrate the steepest rise toward the top-left corner, suggesting better performance. The tight clustering of the curves also indicates the consistency of the chosen classifiers.

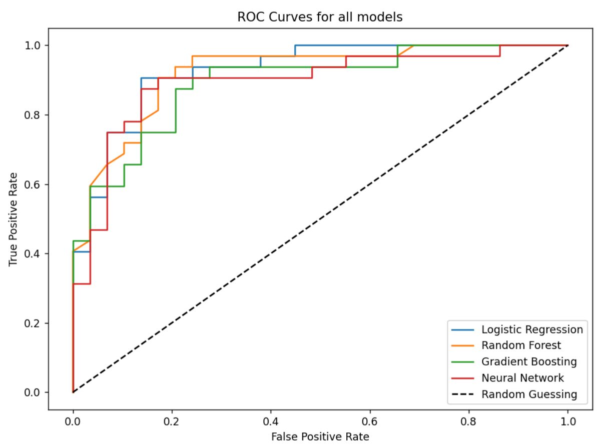


Figure 4: ROC curves (all models)

8 Decision Tree

To better understand feature interactions and how individual decisions are made, we trained a Decision Tree classifier and visualized the split structure. The visualization highlights `cp`, `thal`, and `oldpeak` as dominant decision points for classifying patients as ”risky”.

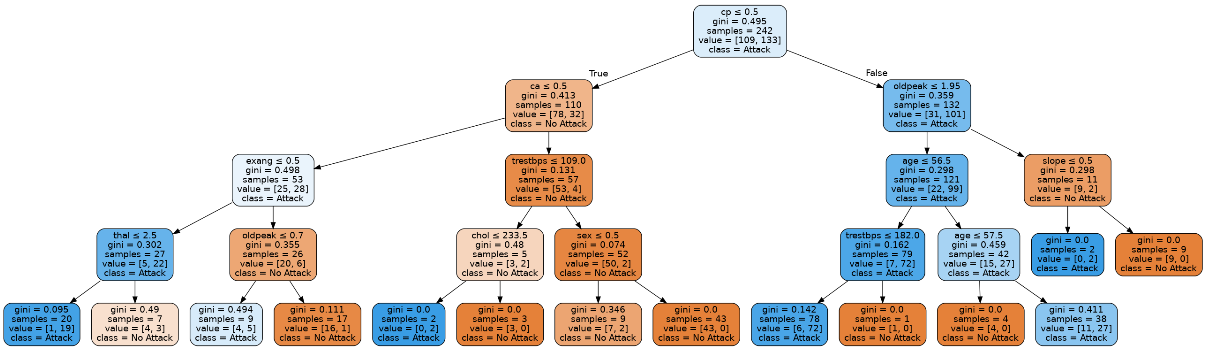


Figure 5: Decision Tree

Each node in the tree represents a decision based on a threshold and includes the Gini impurity, number of samples, and the class distribution in the form [No Attack, Attack].

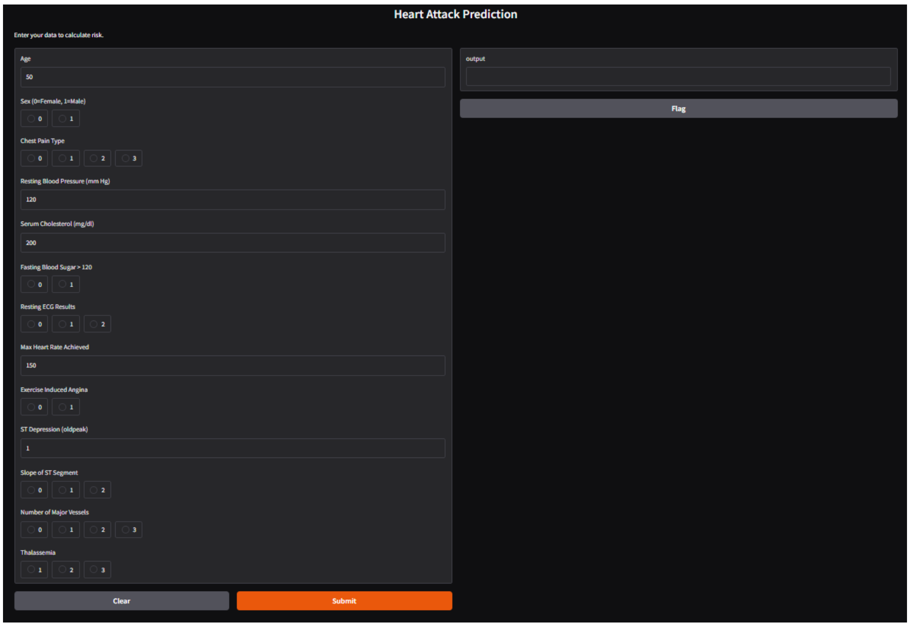
The color of each node indicates the predicted class:

- **Orange** nodes represent predictions of *No Attack*
- **Blue** nodes represent predictions of *Attack*
- Darker shading indicates higher confidence

The tree reveals that features such as `cp`, `thal`, and `oldpeak` play key roles in determining heart attack risk.

9 Best Model and Deployment

The selected model (based on performance aka highest F1 Score) was saved using `joblib` and integrated into a user-friendly web interface using `Gradio`. Users can input patient data and receive a prediction about the likelihood of heart attack.



Heart Attack Prediction

Enter your data to calculate risk.

Age: 50

Sex (0=Female, 1=Male): ○ 0 ○ 1

Chest Pain Type: ○ 0 ○ 1 ○ 2 ○ 3

Resting Blood Pressure (mm Hg): 120

Serum Cholesterol (mg/dl): 200

Fasting Blood Sugar > 120: ○ 0 ○ 1

Resting ECG Results: ○ 0 ○ 1 ○ 2

Max Heart Rate Achieved: 150

Exercise Induced Angina: ○ 0 ○ 1

ST Depression (oldpeak): 1

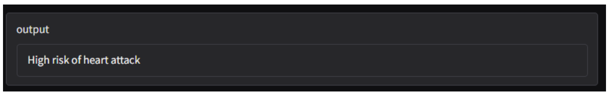
Slope of ST Segment: ○ 0 ○ 1 ○ 2

Number of Major Vessels: ○ 0 ○ 1 ○ 2 ○ 3

Thalassemia: ○ 1 ○ 2 ○ 3

output

Figure 6: User Interface



output: High risk of heart attack

Figure 7: Output

To practically apply the trained model, I developed a simple interactive web interface using the **Gradio***. This user-friendly UI allows users to manually input patient data. The system loads the best-performing saved model (selected based on F1-score during evaluation) and predicts whether the individual is at risk of experiencing a heart attack. This deployment shows how the model could be integrated in real-life solution and potentially assist in preventive decision-making.

* The website can be launched after running `python predictor.py` in the terminal which gives us a local URL. If I were to generate a public link, it would expire after a week which wouldn't be convenient. The accessibility of the website is a challenge for further development.

10 Conclusion

This project explored the use of machine learning techniques to predict heart attack risk based on patient data. Multiple classification models were trained, evaluated, and compared using performance metrics such as accuracy, precision, recall, and F1-score. Emphasis was placed not only on predictive performance but also on model interpretability and usability. To demonstrate real-world applicability, the final model was deployed through a user-friendly Gradio web interface. The project highlights the potential of data-driven approaches in supporting early detection and clinical decision-making in healthcare.

References and Links

The implementation of certain model architectures, training and evaluation procedures were guided by publicly available examples and documentation. These sources were adapted and modified to fit the context and structure of this project.

- **Github** : https://github.com/nikolovaaangela/MLDM-Project
- Exploratory Data Analysis (previous project of mine):
  https://heart-attack-predictions-eda.netlify.app/
- Heart Attack Dataset:
  https://www.kaggle.com/datasets/pritsheta/heart-attack/data
- Scikit-learn Documentation: https://scikit-learn.org
- Keras Documentation: https://keras.io
- Gradio Documentation: https://www.gradio.app/docs
- Graphviz Documentation: https://graphviz.org/documentation/
- Code References :
  https://www.geeksforgeeks.org/
  https://www.w3schools.com
  https://www.tensorflow.org
  https://keras.io/guides/
  https://codefinity.com/courses