Google Developer Groups
On Campus • Canadian International College

GDG on Campus CIC
Data Science Track
Final Project

Malak Khaled El Hussieny
Email: malakkhaleds2mm@gmail.com

Heart Attack Prediction Model

Project Overview

This project applies the **Data Science Life Cycle** to predict the risk of heart attack based on patient health indicators. Using machine learning, we analyze clinical features (e.g., age, blood pressure, glucose, troponin levels) to classify patients into **positive (at risk)** or **negative (not at risk)** categories.

About The Data Source

Dataset: Heart Attack Clinical Dataset from Kaggle.
Format: CSV file (Heart Attack.csv)
Size: 1,319 patient records, 9 features (age, gender, pulse, blood pressure, glucose, kcm, troponin, class).
Note: This dataset was obtained from https://www.kaggle.com/datasets/bharath011/heart-disease-classification-dataset/data .

First: Data Exploration

- Displayed the first few rows of the dataset using **df.head().**
- Used **df.info()** and **df.describe()** to understand column types and statistics.
- Checked unique values for gender and class.
- Visualized distributions using:
    1. Histogram for age
    2. Pie chart for gender distribution
    3. Bar chart for heart attack counts
    4. Created a correlation heatmap to identify relationships between features.

Second: Data Cleaning & Preprocessing

- **Renamed columns for clarity:** impluse → impulse, pressurehight → systolic_bp, pressurelow → diastolic_bp.
- Handled **outliers in the impulse column** by removing values outside 30–220.
- Verified no missing values using **df.isnull().sum().**
- Encoded the target variable class using **LabelEncoder**.

Third: Feature Engineering & Scaling

- The Added a new feature **pulse_pressure = systolic_bp - diastolic_bp.**
- Scaled all features using **StandardScaler** to improve model performance.

Fourth: Model Building

- Split the dataset into training (80%) and test (20%) sets.
- Applied **Logistic Regression** for binary classification.
- Trained the model on the training data.

Fifth: Model Evaluation

Evaluated performance with:

- Confusion matrix (plotted as a heatmap)
- Accuracy score
- Classification report (Precision, Recall, F1-score, Macro avg, Weighted avg).

Sixth: Results

- Accuracy: 83%
- Negative Class: Precision = 0.77, Recall = 0.80, F1 = 0.78
- Positive Class: Precision = 0.87, Recall = 0.84, F1 = 0.86
- Macro Avg F1: 0.82 (balanced performance across classes)
- Weighted Avg F1: 0.83 (slightly favors Positive class, since dataset has more positive cases)

Seventh: Comparing LR with SVM model

- Applied Support Vector Machine (SVM) with three kernels: **linear, polynomial, and RBF**.
- Results:
    - Linear kernel: Accuracy = ~0.81
    - Polynomial kernel: Accuracy = ~0.70
    - RBF kernel: Accuracy = ~0.76
- The **linear kernel** performed best.
- Compared Logistic Regression (83%) vs. Best SVM (linear kernel, ~81%).

- Logistic Regression slightly outperformed SVM, though both showed comparable performance.

Eighth: Conclusions

- The Logistic Regression model achieved good performance (83% accuracy).
- The Logistic Regression model performs better at predicting Positive (at risk) patients, which is critical in a medical setting since false negatives (missed risks) can be dangerous.
- There is still room for improvement:
    - Using k-fold cross-validation on the training set to choose the best SVM kernel, instead of picking the kernel based only on test accuracy.
    - Perform hyperparameter tuning to optimize model performance.