**Phase-2 Submission**

**Student Name:** DINESH KUMAR N

**Register Number:** 712523106004

**Institution:** PPG INSTITUTE OF TECHNOLOGY

**Department:** ELECTRONICS AND COMMUNICATION ENGINEERING

**Date of Submission:** [07/05/2025]

**Github Repository Link:** [https://github.com/Dinesh180406/NM\_DINESHKUMAR\_DS]

### **1. Problem Statement**

* *Problem: Cardiovascular diseases are the leading cause of death globally. Current risk assessments can be limited in capturing the complex factors leading to delayed diagnoses.*
* *Binary classification – predicting the presence or absence of heart disease.*
* *Impact: Early prediction enables timely interventions, personalized healthcare, better resource allocation, informs public health strategies, reduces healthcare costs, and empowers individuals to take proactive health measures, ultimately saving lives and improving well-being.*

### **2. Project Objectives**

* *Specify what the mo* *High Accuracy & Robustness: Achieve significant accuracy in predicting heart disease on unseen data.*

* *\* Feature Importance: Identify key risk factors with reasonable interpretability.*

*\* Real-World Viability: Develop a practical and efficient model for healthcare use*

* *Observed class imbalance necessitates focus on precision, recall, and F1-score, prompting exploration of imbalance handling techniques.*

### **3. Flowchart of the Project Workflow**

### 

### 

### **4. Data Description**

* *Source: Kaggle/UCI Machine Learning Repository (common origins).*
* *Data Type: Structured (tabular).*
* *Size: Hundreds to thousands of records; 10-15 features (e.g., age, blood pressure, cholesterol, ECG results).*
* *Nature: Static (point-in-time data).*
* *Target variable (if supervised learning)* *Target: Binary (presence/absence of heart disease).*

### **5. Data Preprocessing**

* *Removal: If the amount of missing data is small and random, you might choose to remove rows with missing values.*

*\* Imputation: Filling missing values with the mean, median, mode, or using more advanced imputation techniques.*

* *We identify duplicate rows using df.duplicated().sum() and can inspect them using df[df.duplicated(keep='first')].*

*The decision to remove duplicates depends on whether they represent genuine repeated measurements or errors in data collection. In this example, we assume they are genuine duplicates and remove them using*

* *Outliers are data points that significantly deviate from the rest of the data.*

*\* We use box plots (seaborn.boxplot) to visualize the distribution of numerical features and identify potential outliers.*

*\* The code demonstrates the Interquartile Range (IQR) method for outlier detection and removal in the 'chol' column as an example.*

* *We check the data types of each column using df.dtypes.*

*\* For categorical columns, we examine the value counts using df[col].value\_counts() to identify any inconsistencies in spelling or representation.*

* *Machine learning models typically require numerical input. Therefore, we need to encode categorical variables into numerical representations.*

*\* Label Encoding: Used for binary categorical features (like 'sex' where there are two unique values). It assigns a numerical label (e.g., 0 and 1) to each category.*

*\* One-Hot Encoding: Used for multi-class categorical features. It creates new binary columns for each unique category. The original column is replaced by these new columns, where a '1' indicates the presence of that category and '0' otherwise. pd.get\_dummies() is used for one-hot encoding, and drop\_first=True is often used to avoid multicollinearity.*

* *Normalize or standardize features where required. The choice between normalization and standardization depends on the specific algorithm and the distribution of the data. Standardization is often preferred as it is less affected by outliers.*

*This code provides a comprehensive starting point for preprocessing your heart disease prediction dataset. Remember to adapt each step based on the specifics of your data and the requirements of the machine learning models you intend to use. Ensure you document your decisions and the reasoning behind each transformation.*

### 

### **6. Exploratory Data Analysis (EDA)**

* *Univariate Analysis:*
  + *Distribution of Numerical Features: Histograms are used to visualize the frequency distribution of each numerical feature, providing insights into their central tendency, spread, and skewness. Box plots display the summary statistics (median, quartiles, min/max, and potential outliers) of each numerical feature.*
* *Bivariate/Multivariate Analysis:*
  + *\* Correlation Matrix: A heatmap of the correlation matrix displays the pairwise linear correlations between all numerical features. Correlation coefficients range from -1 to 1, indicating the strength and direction of the linear relationship.*

*\* Pair Plots: These scatter plot matrices visualize the pairwise relationships between multiple numerical features. They can also include the target variable with different colors for each class, allowing for a visual assessment of how features separate the target classes. (Note: Pair plots can be computationally expensive for datasets with many numerical features, so they might be limited to a subset.)*

*\*Grouped Bar Plots: For categorical features, grouped bar plots (or stacked bar plots) show the count of each target class within each categor Alright, let's dive into some Exploratory Data Analysis (EDA) for your heart disease prediction project. Here's the breakdown:*

*Heart Disease Prediction - Exploratory Data Analysis*

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

* + *Point- Biserial Correlation (for numerical features and binary target)*

*print("\n Point- Biserial Correlation with Target:")*

*for col in numerical\_cols: correlation = df[col].corr (df [TARGET\_COL]) print(f" Correlation between {col} and {TARGET\_COL}: {correlation:.2f}")*

* *Insights Summary:*
  + *Patterns and Trends: Describe any visual patterns observed in the distributions (e.g., skewness in numerical features, dominant categories in categorical features) and relationships between features (e.g., strong positive or negative correlations).*

*\* Interesting Observations: Point out any surprising or noteworthy findings, such as features with clearly different distributions for the two target classes or strong correlations between seemingly unrelated features.*

* + *Significant Chi-Square Test: Categorical features that show a statistically significant association with the target variable (low p-value in the chi-square test) are likely to be informative for the model, as the occurrence of heart disease varies across their categories.*

*\* Domain Knowledge: While not explicitly derived from the EDA code, prior medical knowledge about risk factors for heart disease can also guide the interpretation of the results and the selection of potentially influential features.*

*This EDA provides a crucial foundation for subsequent steps in the machine learning pipeline, such as feature selection, model building, and interpretation of results*

### 

### **7. Feature Engineering**

* *1. Feature Engineering:*

*\* Age Group: Created age\_group (Young Adult, Middle-Aged, Older Adult, Senior, Very Senior) by binning 'age', and age\_group\_encoded (label encoded). Justification: Captures non-linear age-related risk.*

*\* Blood Pressure Category: Created blood\_pressure\_category (Normal, Elevated, Stage 1 Hypertension, Stage 2 Hypertension) by combining and binning 'trestbps' and 'chol' (as a proxy for diastolic). Justification:Simplifies blood pressure risk.*

* *Cholesterol Ratio: Created chol\_thalach\_ratio ('chol' divided by 'thalach' as a proxy for HDL). Justification: Potentially a better risk predictor than individual cholesterol.*
* *Polynomial Feature: Created age\_chol\_interaction (interaction term between 'age' and 'chol'). Justification: Captures potential combined effects.*

*Applied PCA (n\_components=2) to scaled 'age', 'chol', 'trestbps', and 'thalach'.*

*\* Generated principal\_component\_1 and principal\_component\_2.*

*\* Justification: Reduces dimensionality, aids visualization, and can improve model training.*

*The goal of these engineered features is to provide more informative representations for the heart disease prediction model. Their effectiveness will be evaluated during model training*

* *No original features were removed in this step for demonstration. Removal decisions (due to redundancy, low relevance, or high missing values) would typically follow further analysis.*

### 

### **8. Model Building**

* *Select and implement at least 2 machine learning models.*
  + *\*Logistic Regression: Given its interpretability and efficiency in binary classification tasks, especially as a baseline model. It helps us understand the linear relationships between the features and the likelihood of heart disease.*
  + *\* Random Forest: Chosen for its ability to model complex, non-linear relationships and its generally high predictive accuracy. It's also robust to outliers and can handle a large number of features effectively.*
* *Computational resources: Training and deploying complex models like deep learning require more computational resources than simpler models.*

*In practice, it's often beneficial to try several of these models, evaluate their performance on appropriate metrics (e.g., accuracy, precision, recall, F1-score, AUC), and choose the one that best balances predictive power, interpretability, and computational efficiency for the specific heart disease prediction task and available data.*

* *Split data We'll split the dataset into training and testing sets to evaluate how well our models generalize to unseen data. Stratification will be used to maintain the proportion of positive and negative cases in both sets, which is crucial for a balanced evaluation, especially if the class distribution is imbalanced.*
* *Train models and evaluate initial performance using appropriate metrics.*
  + *Now, we'll train the selected models on the training data and then evaluate their performance on the testing data using accuracy, precision, recall, and F1-score.*
  + *For regression: This output provides the initial performance metrics for both Logistic Regression and Random Forest models on the unseen test data. These results can then be compared to determine which model performs better in its default configuration. Further steps would involve exploring hyperparameter tuning to potentially enhance the performance of each model.*

### 

### 

### **9. Visualization of Results & Model Insights**

* ***Confusion matrix****; residual plots, etc.* *A confusion matrix visually represents the performance of a classification model.*

***\*ROC curve****;* *The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds.*

***\* feature importance plot****;* *Visualizes the relative importance of each input feature in predicting the target variable (heart disease).Different models provide feature importance scores in different ways (e.g., coefficients in linear models, Gini importance in tree-based models).*

* *Include visual comparisons of model performance.*
* *Interpret top features influencing the outcome.*
* *Clearly explain what each plot shows and how it supports conclusions.]*

### 

### **10. Tools and Technologies Used**

* *Python: Widely used for data analysis, machine learning, and visualization (libraries like Pandas, NumPy, Scikit-learn, TensorFlow).*

***\*****R: Popular for statistical analysis and data visualization.*

* *Scikit-learn: For building and evaluating machine learning models (e.g., Logistic Regression, Decision Trees).*

***\**** *TensorFlow and Keras: For deep learning models like neural networks.* ***\*****Pandas: For data manipulation and preprocessing.*

***\*****Matplotlib and Seaborn: For creating informative plots and comparisons.*

***\*****NumPy: For numerical computations.*

* *SHAP and LIME: For model interpretability and feature contribution.*
* *Feature Importance Plots: To identify the most influential features.*

### 

### **11. Team Members and Contributions**

***[****List names and responsibilities.*

* *Clearly mention who worked on:*

|  |  |  |
| --- | --- | --- |
| **MEMBERS** | **ROLE** | **DESCRIPTION** |
| HARI A | *Data**Collection**&**Preprocessing* | *Responsible for acquiring the dataset, cleaning the data, and preparing it for analysis* |
| DINESH KUMAR N | *Exploratory Data Analysis & Feature Engineering* | *In charge of performing EDA, visualizing insights, and engineering features to enhance model performance* |
| *JONEYABIRAHAM Y* | *Model Building & Evaluation* | *Develops and trains the deep learning model (CNN), tunes hyperparameters, and evaluates performance metrics.* |
| *MARI RAMESH M* | *Documentation and reporting* | *Handles interpretation of results, builds the user interface (if applicable), and manages optional deployment using tools like Streamlit or Gradio.* |