Heart Attack Prediction Model

Introduction

This project aims to develop a predictive model for heart attack prediction using a dataset containing various health-related features. The goal is to maximize the recall and precision metrics instead of accuracy, focusing on the importance of correctly identifying cases with a higher chance of heart attack.

Features

The dataset includes the following features:

- **age**: age of the patient
- **sex** : sex of the patient
- **cp** : chest pain type

0 = typical angina
1 = atypical angina
2 = non-anginal pain
3 = asymptomatic

- **trtbps** : resting blood pressure in mm Hg
- **chol** : cholestoral in mg/dl
- **exng** : exercise induced angina

1 = yes
0 = no

- **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
- **slp** : slope
- **caa** : number of major vessels
- **thall** : thalium stress test result
- **target** :

0 = less chance of heart attack
1 = more chance of heart attack

Assignment Objectives

1. **Data Exploration and Analysis:**
   - Explore the dataset and analyze relationships between features.
   - Identify correlations and visualize feature distributions.
2. **Data Pre-processing:**
   - Handle missing values, outliers, and address unbalanced data.
   - Perform feature engineering, including handling correlated features and scaling.
3. **Model Building:**
   - Split the dataset into training and testing sets.
   - Choose appropriate models for heart attack prediction (e.g., Logistic Regression, Random Forest,XG boost).
4. **Model Evaluation:**
   - Evaluate models using recall and precision metrics.
   - Visualize the confusion matrix for better understanding of model performance.
5. **Presentation:**
   - Create a non-code report using Jupyter Notebook or export as PDF.
   - Summarize findings, insights, and visualizations from the analysis.
6. **Explanation:**
   - Explain reasoning behind preprocessing steps, model selection, and metric choices.
   - Discuss the impact of certain features on heart attack prediction.

Libraries/Package Used

1. Importing Libraries:

   1. %matplotlib inline: This line enables inline plotting within a Jupyter Notebook environment, allowing visualizations to be displayed directly below code cells.
   2. numpy: NumPy is a fundamental library for numerical computing in Python, providing efficient array operations and mathematical functions.

3. pandas: Pandas is a powerful library for data manipulation and analysis, offering data structures like DataFrames and Series for working with tabular data.
4. matplotlib.pyplot: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
5. seaborn: Seaborn builds on Matplotlib, providing a high-level interface for creating informative and aesthetically pleasing statistical graphics.
6. sklearn.model_selection: This module from scikit-learn offers tools for splitting datasets into training and testing sets, crucial for model evaluation.
7. collections: This built-in Python module contains the Counter class, used for counting hashable objects.
8. sklearn.linear_model: This module provides various linear models for classification and regression, including LogisticRegression.
9. sklearn.metrics: This module houses a collection of metrics for evaluating model performance, such as accuracy, precision, recall, F1-score, and AUC-ROC.
10. sklearn.ensemble: This module contains ensemble methods like RandomForestClassifier, which combine multiple base models for improved performance.
11. xgboost: XGBoost is a powerful library for gradient boosted decision trees, often used for classification and regression tasks.
12. warnings: This module allows for managing warning messages generated during code execution.

2. Setting Warning Filter:

13. warnings.simplefilter("ignore"): This line suppresses warning messages, which can be helpful for streamlining output but should be used with caution as it might mask potential issues.

Conclusion

- Logistic Regression emerges as the most suitable model for heart attack prediction, given its high recall and reasonable precision. Correctly identifying individuals with a higher chance of a heart attack is crucial for this task.
- Further optimization of hyperparameters and feature engineering may enhance the model's performance.
- Consider conducting additional analyses to understand feature impact on predictions and identify areas for improvement.