Heart Attack Prediction Model

Introduction

This project tackles the crucial challenge of predicting heart attacks, aiming to identify individuals at higher risk. Utilizing a comprehensive dataset of health-related features, the model strives to:

Prioritize correct identification of potential heart attacks: Shifting the focus from pure accuracy to recall and precision prioritizes correctly identifying individuals with a higher likelihood of having a heart attack. This enhances early intervention and potentially saves lives.

Leverage diverse health data: By incorporating various health-related features, the model seeks to uncover complex relationships between these factors and the risk of heart attack, leading to more comprehensive and reliable predictions.

By prioritizing cases with higher risk and considering a wide range of health factors, this project has the potential to significantly improve the early detection and prevention of heart attacks.

This brief introduction provides a concise overview of the project's goals and approach. You can further expand on it by highlighting the specific methods used to build the model (e.g., machine learning algorithms) and the anticipated outcomes (e.g., potential impact on healthcare interventions).

Data understanding

1. Age: Age of the patient
2. sex: Sex of the patient
3. exang: exercise induced angina (1 = yes; 0 = no)
4. ca: number of major vessels (0-3)
5. cp: Chest Pain type chest pain type

   Value 1: typical angina

   Value 2: atypical angina

   Value 3: non- anginal pain

   Value 4: asymptomatic
6. trtbps: resting blood pressure (in mm Hg)
7. chol: cholestoral in mg/dl fetched via BMI sensor
8. fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

9. rest_ecg: resting electrocardiographic results Value 0: normal

Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

10. thalach: maximum heart rate achieved 11. target: 0= less chance of heart attack 1= more chance of heart attack

Data preparation

Handling Missing Values:

- df.dropna(inplace=True): This line addresses missing values (NaN) in the DataFrame df by removing rows containing any missing data.
    - Considerations:
        - Removing rows can lead to potential loss of information, especially if missingness is substantial.
        - Imputation methods can be explored to fill in missing values strategically

Handling Outliers:

- from scipy import stats: This imports the `stats` module from the SciPy library for statistical calculations.
- df = df[(np.abs(stats.zscore(df)) < 3).all(axis=1)]: This line identifies and removes outliers based on their z-scores.
- Z-scores: Measure how far a value deviates from the mean in standard deviation units.
- Threshold of 3: Values with absolute z-scores greater than 3 are considered outliers and removed.
- (axis=1): Ensures the condition applies to all columns in each row.

.

Multicollinearity:

- VIF: Measures how much the variance of an estimated regression coefficient is inflated due to multicollinearity (strong correlation between independent variables).
- Purpose: Identifying and addressing multicollinearity helps improve model performance and interpretability.

- Common Threshold: VIF values above 5 or 10 often raise concerns about multicollinearity.
- Actions: Consider removing highly correlated features, combining them, or using regularization techniques to mitigate multicollinearity.

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.

Model

Evaluation Parameter

1. Accuracy:
    - Accuracy is a measure of the overall correctness of a model. It is calculated as the ratio of correctly predicted instances to the total instances. The formula is:
      Accuracy=True Positives+True NegativesTotal InstancesAccuracy=Total Instances True Positives+True Negatives. High accuracy indicates that the model is making correct predictions across all classes.
2. Recall (Sensitivity or True Positive Rate):
    - Recall measures the ability of the model to capture all the relevant instances of the positive class. The formula is:
      Recall=True PositivesTrue Positives+False NegativesRecall=True Positives +False NegativesTrue Positives. High recall indicates that the model is good at identifying positive instances.
3. Precision:
    - Precision is the accuracy of the positive predictions made by the model. The formula is:
      Precision=True PositivesTrue Positives+False PositivesPrecision=True Positives+False PositivesTrue Positives. High precision indicates that the model's positive predictions are accurate.
4. F1 Score:
    - The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics. The formula is:
      F1 Score=2×Precision×RecallPrecision+RecallF1 Score=2×Precision+RecallPrecision×Recall. It is particularly useful when there is an imbalance between the positive and negative classes.
5. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):
    - The ROC-AUC score measures the area under the Receiver Operating Characteristic curve. This curve plots the true positive rate against the false positive rate at various threshold settings. A higher ROC-AUC score indicates better discrimination between the positive and negative classes.

Logistic Regression

Logistic Regression is a linear classification algorithm used for binary and multiclass classification tasks. The model predicts the probability of an instance belonging to a particular class by applying the logistic function to a linear combination of input features and coefficients. During training, the algorithm adjusts these coefficients using Maximum Likelihood Estimation, aiming to maximize the likelihood of the observed data. The learned model establishes a decision boundary, typically a hyperplane, that separates instances into distinct classes. During prediction, new instances are classified based on the calculated probability, with a threshold determining the final assigned class. Logistic Regression is widely employed due to its simplicity, interpretability, and effectiveness in scenarios where the relationship between features and the target variable is approximately linear.

1. Accuracy:
    - Accuracy is a measure of the overall correctness of the model. It is calculated as the ratio of correctly predicted instances to the total instances. In your case, the accuracy is 0.862069, which means the model correctly predicted the class for approximately 86.21% of the instances.
2. Recall (Sensitivity or True Positive Rate):
    - Recall is the ability of the model to capture all the relevant instances of the positive class. It is calculated as the ratio of true positives to the sum of true positives and false negatives. In your case, the recall is 0.931034, indicating that the model correctly identified about 93.10% of the actual positive instances.
3. Precision:
    - Precision is the measure of the accuracy of the positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives. In your case, the precision is 0.818182, suggesting that around 81.82% of the instances predicted as positive by the model were indeed true positives.
4. F1 Score:
    - The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, considering both false positives and false negatives. The F1 score is calculated as $2\times Precision\times RecallPrecision+Recall2\times Precision+RecallPrecision\times Recall$. In your case, the F1 score is 0.870968.
5. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):

- The ROC-AUC is a performance metric for binary classification problems. It measures the area under the Receiver Operating Characteristic curve, which plots the true positive rate against the false positive rate at various threshold settings. A higher ROC-AUC score indicates better discrimination between the positive and negative classes. In your case, the ROC-AUC is 0.869091.

Random Forest Classifier

regression tasks. It constructs a multitude of decision trees during training, each tree trained on a random subset of the dataset and a random subset of features. Through a process known as bagging (Bootstrap Aggregating), the algorithm reduces overfitting by aggregating the predictions of individual trees. During prediction, each tree "votes" on the class, and the class with the majority of votes is assigned as the final prediction. The randomness introduced in both data and feature selection enhances the model's robustness and generalization capabilities. Random Forest also provides insights into feature importance, aiding in understanding the contributions of different features to the model's predictive accuracy. Overall, the ensemble nature of Random Forest makes it a powerful and versatile algorithm, well-suited for a variety of machine learning tasks.

1. Recall (Sensitivity or True Positive Rate):
    - Recall measures the ability of the model to capture all the relevant instances of the positive class. It is calculated as the ratio of true positives to the sum of true positives and false negatives. In your case, a recall score of 0.8620689655172413 means that the model correctly identified approximately 86.21% of the actual positive instances.
2. Precision:
    - Precision is the accuracy of the positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives. Your precision score of 0.8620689655172413 indicates that around 86.21% of the instances predicted as positive by the model were indeed true positives.
3. F1 Score:
    - The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, considering both false positives and false negatives. A score of 0.8620689655172413 suggests a well-balanced performance.
4. Accuracy:
    - Accuracy is a measure of the overall correctness of the model, calculated as the ratio of correctly predicted instances to the total instances. An accuracy score of 0.8620689655172413 means that the model correctly predicted the class for approximately 86.21% of the instances.
5. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):

- The ROC-AUC score measures the area under the Receiver Operating Characteristic curve, which plots the true positive rate against the false positive rate at various threshold settings. A score of 0.8620689655172413 indicates good discriminatory power between the two classes.

XGBoost

XGBoost, short for eXtreme Gradient Boosting, is a powerful gradient boosting algorithm designed for classification and regression tasks. It builds an ensemble of decision trees sequentially, with each tree correcting the errors made by the preceding ones. During training, the algorithm optimizes a specified objective function using gradient descent, adjusting the weights of misclassified instances. XGBoost incorporates regularization terms to control overfitting, and it introduces a parallel processing approach for efficient computation. The algorithm's strength lies in its ability to handle complex relationships within data, adapt to various scenarios, and provide insights into feature importance. Its robustness, efficiency, and versatility have made XGBoost a popular choice in machine learning competitions and real-world applications**.**

1. Recall (Sensitivity or True Positive Rate):
    - Recall measures the ability of the model to capture all the relevant instances of the positive class. In your case, a recall score of 0.8275862068965517 means that the XGBoost model correctly identified approximately 82.76% of the actual positive instances.
2. Precision:
    - Precision is the accuracy of the positive predictions made by the model. A precision score of 0.8571428571428571 indicates that around 85.71% of the instances predicted as positive by the XGBoost model were indeed true positives.
3. F1 Score:
    - The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics. A score of 0.8421052631578947 suggests a good balance between precision and recall.
4. Accuracy:
    - Accuracy measures the overall correctness of the model and is calculated as the ratio of correctly predicted instances to the total instances. An accuracy score of 0.8448275862068966 means that the XGBoost model correctly predicted the class for approximately 84.48% of the instances.
5. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):
    - The ROC-AUC score measures the area under the Receiver Operating Characteristic curve, which shows the trade-off between true positive rate and false positive rate at various threshold settings. A ROC-AUC score of 0.8452380952380952 indicates good discriminatory power between the two classes.

Final Conclusion

1. Logistic Regression

- Achieves the highest recall (93.10%) among the models, indicating a strong ability to correctly identify individuals with a higher chance of a heart attack.
- Reasonable precision at 81.82%, striking a good balance between correct identifications and minimizing false positives.

2. RandomForest

- Consistent performance across all metrics with an accuracy, recall, precision, and F1 score of 86.21%.
- Offers balanced performance across multiple metrics, although not excelling in recall like Logistic Regression.

3. XGBClassifier

- Slightly lower recall compared to Logistic Regression but maintains a good precision of 85.71%.
- Competitive ROC AUC, indicating overall good model performance.

Recommendation

- 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**.**