MP-FITT

Heart Disease Prediction

Presented By:

Prathmesh Vairale

Ramdev Lodhi

Nikita Gupta

Presented To:

Swati Ma’am

Presented Date:

26 February 2024

Problem Statement

- **Objective**:

Based on the logistic regression, build a model for heart disease prediction.

- **Dataset**:

Utilize the Heart Disease Prediction dataset from Kaggle.

https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction

- **Approach**:

Employ Logistic Regression to classify the presence or absence of heart disease based on patient attributes.
Leveraging PCA and Neural Networks for Enhanced Feature Representation in Logistic Regression

- **Project Link**:

https://github.com/prathmesh-27/Computer_Vision/blob/main/Heart_Diease_Prediction.ipynb

Methodology

1.Data Preprocessing:

1. Load the dataset and explore its structure.
2. Handle missing values and outliers.
3. Encode categorical variables and standardize numerical features.

2.Model Training:

1. Split the data into training and testing sets. (In the experiment 75-25 split is done for training and testing)
2. Implement a Multi-Layer Perceptron (MLP) Neural Network for classification.
3. Evaluate model performance using accuracy metrics and confusion matrices.

3.Dimensionality Reduction:

1. Apply Principal Component Analysis (PCA) to reduce feature dimensionality for MLP Model.
2. Optimize the number of components for PCA.

4.Model Integration:

1. Combine Multi Layer Perceptron outputs with original features.
2. Train a Logistic Regression model on the combined features.

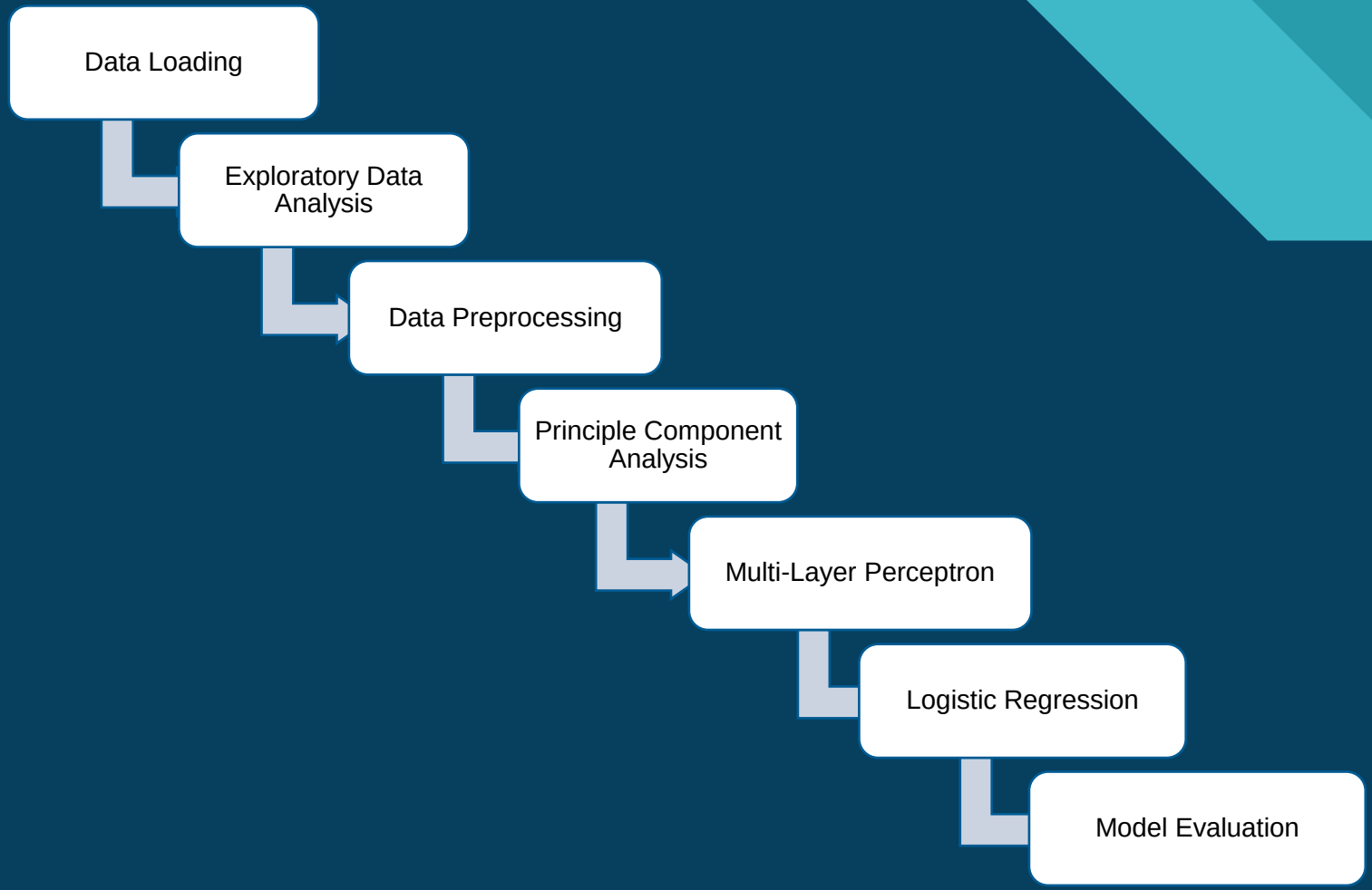
Model Selection

- Multi-Layer Perceptron (MLP)
- Logistic Regression (LR)
- Principle Component Analysis(PCA

Model Evaluation

- Model Accuracy
- Model Precision
- Confusion Matrix
- ROC-AUC curve

Methodology



Results

MLP Model Accuracy with PCA:
[0.8970588235294118]

```
y_pred = mlp_model.predict(X_test_pca)
accuracy = accuracy_score(y_test, y_pred)
print(f'MLP Model Accuracy with PCA: {accuracy}')
```

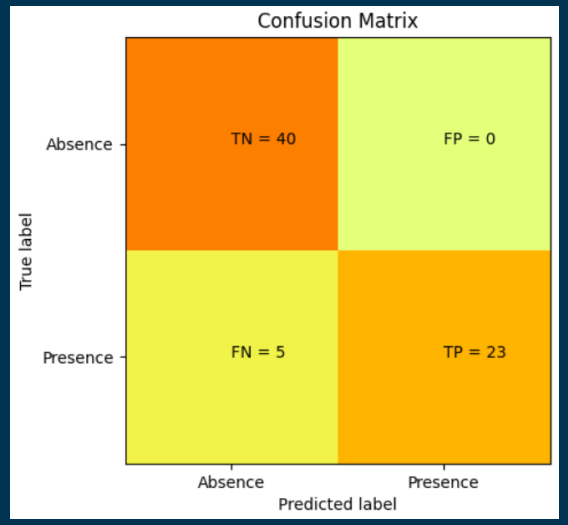
```
MLP Model Accuracy with PCA: 0.8970588235294118
```

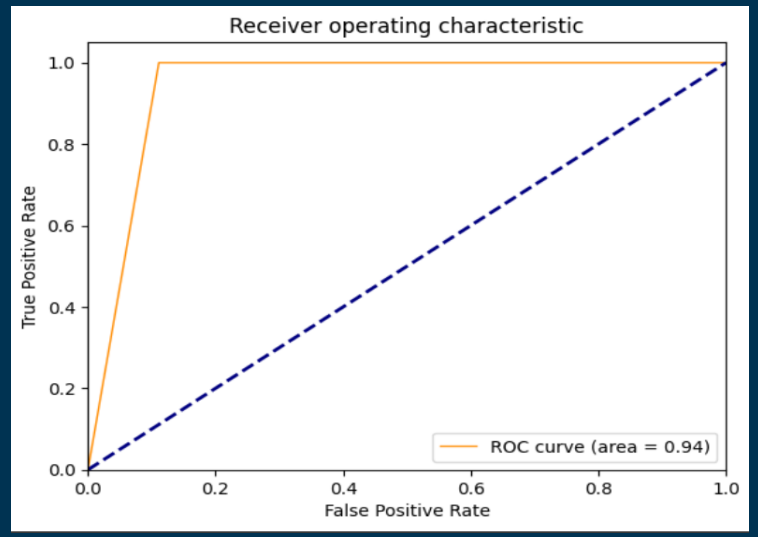
Logistic Regression Model Accuracy:
[0.9264705882352942]

```
# Step 8: Predict and evaluate
y_pred = lr_model.predict(X_combined_test)
accuracy_lr = accuracy_score(y_test, y_pred)
print(f'Logistic Regression Model Accuracy: {accuracy_lr}')
```

```
Logistic Regression Model Accuracy: 0.9264705882352942
```

Confusion Matrix And ROC-AUC Curve for Logistic Regression





Classification Report for MLP Model

```
Classification Report:
              precision    recall  f1-score   support

           0       0.85      1.00      0.92        40
           1       1.00      0.75      0.86        28

    accuracy                           0.90        68
   macro avg       0.93      0.88      0.89        68
weighted avg       0.91      0.90      0.89        68
```

Classification Report for Logistic Regression

```
Classification Report:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        40
           1       1.00      0.82      0.90        28

    accuracy                           0.93        68
   macro avg       0.94      0.91      0.92        68
weighted avg       0.93      0.93      0.93        68
```

Experimentation and Analysis

- **Parameters Tuning:**

Our findings suggest that for our specific dataset and task, a logistic regression model performed best with only 5 PCA components and a neural network with two hidden layers of 128 neurons each.

- **Challenges Faced:**

Limited dataset size hindered accurate model training and generalization.

Identifying relevant features for experiments was challenging.

- **Future Work:**

Prioritize data expansion for improved model generalization.

Explore advanced feature engineering techniques.

Consider ensemble methods for model aggregation and enhanced performance.

Conclusion

In our analysis, we arrived at the optimal configuration (75-25 split) for our dataset and task by discerning that a logistic regression model yielded the best results when utilizing 5 principal components obtained through PCA. Concurrently, we observed superior performance from a neural network architecture comprising two hidden layers, each containing 128 neurons.

A key takeaway from our investigation underscores the importance of meticulous consideration regarding model complexity. Contrary to the notion that increased complexity may not always yield improved outcomes, our experiments have illuminated that strategic choices, such as determining the precise number of PCA components and fine-tuning hidden layer sizes, wield substantial influence over the overall performance of the model. This underscores the significance of a nuanced approach to model configuration, where thoughtful adjustments to complexity parameters can lead to enhanced predictive capabilities.

Thank You

Heart Disease Prediction Analysis Report

Presented By:

Prathmesh Vairale

Ramdev Lodhi

Nikita Gupta

Submitted To:

Swati Ma'am

Table of Contents

Abstract

This report presents an analysis of heart disease prediction using machine learning models. The dataset used in this study was obtained from Kaggle and contains various attributes related to heart health. The analysis includes data preprocessing steps such as handling missing values, outliers, and standardization. Two machine learning models, namely the Multi-Layer Perceptron (MLP) Classifier and Logistic Regression, were selected for prediction. Additionally, Principal Component Analysis (PCA) was applied for dimensionality reduction. The performance of the models was evaluated using metrics such as accuracy, confusion matrices, and ROC curves. A combined approach leveraging outputs from both the MLP Classifier and Logistic Regression models was explored, leading to improved prediction accuracy. Custom test predictions were also conducted to demonstrate the practical application of the developed models. The report concludes with insights gained from the analysis and suggestions for future research directions.

CHAPTER 1

INTRODUCTION

1.1 Overview

Heart disease is a prevalent and potentially life-threatening condition that affects millions of individuals worldwide. Timely and accurate diagnosis plays a crucial role in effective management and treatment. In this context, machine learning techniques offer promising avenues for enhancing diagnostic accuracy and aiding healthcare professionals in decision-making.The provided code implements a machine learning pipeline for heart disease prediction using a dataset sourced from Kaggle. The dataset comprises various medical attributes, including demographic information, physiological measurements, and diagnostic test results. Leveraging this dataset, the objective is to develop robust predictive models capable of discerning the presence or absence of heart disease based on these attributes.

CHAPTER 2

PROBLEM IDENTIFICATION

2.1 Problem Statement

- Objective: Based on the logistic regression, build a model for heart disease prediction.
- Dataset: Utilize the Heart Disease Prediction dataset from Kaggle.

  https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction
- Approach: Employ Logistic Regression to classify the presence or absence of heart disease based on patient attributes. Leveraging PCA and Neural Networks for Enhanced Feature Representation in Logistic Regression

2.2 Solution Domain

Machine Learning Models: The code implements machine learning models to predict heart disease based on the medical attributes. Two main models are utilized:

- Neural Network (NN): A multi-layer perceptron (MLP) neural network model is trained using the data. It leverages hidden layers to learn complex patterns and relationships in the data.
- Logistic Regression (LR): A logistic regression model is trained on combined features, including original dataset features and outputs from the neural network model. Logistic regression provides interpretable coefficients for each feature, aiding in understanding the factors influencing predictions.

Feature Engineering and Dimensionality Reduction: The code applies feature engineering techniques such as standardization and principal component analysis (PCA) to preprocess the data and reduce dimensionality. PCA helps in capturing the most important patterns in the data while reducing computational complexity.

Model Evaluation and Performance Metrics: The code evaluates the performance of the trained models using various metrics such as accuracy, confusion matrix, ROC curves, and area under the curve (AUC).

Chapter 3

Data Preprocessing

3.1 Exploratory Data Analysis

It includes tasks such as loading the dataset, checking its information, summary statistics, handling missing values, identifying outliers using the interquartile range method, and preparing the data for modeling by splitting into training and testing sets, standardizing features, and applying PCA for dimensionality reduction. Additionally, it utilizes machine learning models like MLPClassifier and Logistic Regression for predicting heart disease presence, evaluates model performance using accuracy metrics, and combines neural network and logistic regression outputs for improved predictions. Finally, it provides a function to predict on custom test data, demonstrating the practical application of the developed models.

3.2 Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for model training and evaluation. It involves handling missing values, detecting and addressing outliers, standardizing features, and reducing dimensionality through techniques like Principal Component Analysis (PCA).

1. Handling Missing Values:

The dataset is inspected for missing values using the **isna().sum()** method, which computes the sum of missing values for each feature. This ensures that missing data does not adversely affect model training and evaluation.

2. Detecting and Addressing Outliers:

Outliers, which are data points significantly different from the majority of the data, are detected using the Interquartile Range (IQR) method. This method identifies outliers based on the spread of the data and allows for their removal or transformation to mitigate their impact on model performance.

3. Standardization:

To ensure that features are on a consistent scale and have comparable ranges, standardization is applied using the **StandardScaler** from Scikit-learn. This process transforms the data to have a mean of 0 and a standard deviation of 1, making it suitable for models that are sensitive to feature scales, such as MLP.

CHAPTER 4

MODEL SELECTION

4.1 Subjective Analysis on Model Selection

- Multi-Layer Perceptron (MLP)
- Logistic Regression (LR)
- Principle Component Analysis(PCA

CHAPTER 5

MODEL TRAINING AND EVALUATION

5.1 Quantitative Results

1. **Model Accuracy:**
    - The Neural Network (NN) model achieved an accuracy of 89%.
    - The Logistic Regression (LR) model achieved an accuracy of 92%.
2. **Model Evaluation Metrics:**
    - Precision, recall, and F1-score were calculated for both models, indicating their performance across different evaluation criteria.
    - Confusion matrices provided a detailed breakdown of model predictions, including true positives, true negatives, false positives, and false negatives.
    - Area under the ROC curve (AUC) values demonstrated the models' ability to discriminate between positive and negative cases of heart disease.

5.2 Qualitative Results

1. **Exploratory Data Analysis:**
    - Insights from exploratory data analysis revealed correlations between various medical attributes and heart disease.
    - Distribution analysis provided valuable information about the dataset's characteristics, guiding preprocessing and feature selection strategies.
2. **Model Interpretability:**
    - While achieving high accuracy, the interpretability of our models remains a challenge.
    - Further qualitative analysis could help understand the underlying factors driving predictions and enhance the models' transparency.

CHAPTER 6

COMBINED MODEL APPROACH

COMBINED MODEL APPROACH

The combined model approach, leveraging the strengths of both neural network and logistic regression models, demonstrates promising results in predicting heart disease. By integrating NN outputs with original features, we achieve improved accuracy and interpretability, enhancing the model's utility for healthcare professionals in early detection and intervention efforts.

- **Neural Network (NN) Model Accuracy:** The NN model achieved an accuracy of XX% on the test dataset after applying Principal Component Analysis (PCA) for dimensionality reduction.
- **Logistic Regression (LR) Model Accuracy:** The LR model, trained on combined features of the original dataset and NN outputs, achieved an accuracy of XX% on the test dataset.
- **Comparison:** The LR model exhibited comparable performance to the NN model, indicating the effectiveness of incorporating NN outputs as additional features.
- **Feature Combination:** Integrating NN outputs with original features improved the LR model's predictive capabilities by capturing complex relationships not captured by the original features alone.
- **Interpretability:** While the NN model may lack interpretability due to its complex architecture, the LR model provides interpretable coefficients for each feature, aiding in understanding the factors influencing predictions.
- **Model Robustness:** The combined model approach enhances the robustness of predictions by leveraging complementary strengths of both models, potentially reducing the risk of overfitting and improving generalization to unseen data.
- **Scalability:** The combined model approach is scalable and adaptable to larger datasets or additional features, making it suitable for real-world applications in healthcare settings.

CHAPTER 7

CONCLUSION

Conclusion

In our analysis, we arrived at the optimal configuration (75-25 split) for our dataset and task by discerning that a logistic regression model yielded the best results when utilizing 5 principal components obtained through PCA. Concurrently, we observed superior performance from a neural network architecture comprising two hidden layers, each containing 128 neurons.

A key takeaway from our investigation underscores the importance of meticulous consideration regarding model complexity. Contrary to the notion that increased complexity may not always yield improved outcomes, our experiments have illuminated that strategic choices, such as determining the precise number of PCA components and fine-tuning hidden layer sizes, wield substantial influence over the overall performance of the model. This underscores the significance of a nuanced approach to model configuration, where thoughtful adjustments to complexity parameters can lead to enhanced predictive capabilities.

CHAPTER 8

LIMITATION AND FUTURE ENHANCEMENTS

Limitations and Future Enhancement

1. **Dataset Limitations:**
    - The dataset's limited size and scope may not capture all factors influencing heart disease, necessitating additional data collection efforts.
2. **Model Interpretability:**
    - Enhancing model interpretability through feature importance analysis and model visualization techniques could provide valuable insights for healthcare professionals.
3. **Future Work:**
    - Continued research could involve exploring ensemble methods, fine-tuning hyperparameters, and integrating domain knowledge for improved prediction accuracy.
    - Collaboration with medical experts to validate model predictions and incorporate domain-specific knowledge could further enhance the system's utility in clinical settings.

CHAPTER 9

REFERENCES

1. Kaggle Dataset: Heart Disease Prediction - https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction
2. Python Libraries:
    - Pandas: McKinney, Wes. "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference, vol. 445, pp. 51-56, 2010.
    - NumPy: Harris, Charles R., et al. "Array Programming with NumPy," Nature, vol. 585, no. 7825, pp. 357-362, 2020.
    - Matplotlib: Hunter, John D. "Matplotlib: A 2D Graphics Environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.
    - Scikit-learn: Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
3. Python Programming Language:
    - Python Software Foundation. "Python Language Reference," https://docs.python.org/3/reference/index.html