Final-Term Project
Submitted By:
Ramya Vattikuti

Table of Contents

Introduction

In this project, we aim to explore and evaluate various machine learning models for predicting the presence of heart disease. The dataset used in this study is sourced from the UCI Machine Learning Repository and is commonly known as the "Heart Disease Dataset." The models compared in this project include:

- **Random Forest Classifier** (RF)
- **Bidirectional Long Short-Term Memory (Bi-LSTM)** neural network
- **K-Nearest Neighbors (KNN)** classifier

The project investigates the effectiveness of each algorithm in terms of classification accuracy and provides a detailed evaluation of their performance using key metrics, such as precision, recall, accuracy, False Positive Rate (FPR), False Negative Rate (FNR), True Skill Statistic (TSS), and Heidke Skill Score (HSS).

Description of the Project

The primary objective of this project is to predict heart disease presence based on patient data that includes features such as age, sex, blood pressure, cholesterol levels, and more. Each machine learning algorithm is trained on the same dataset and evaluated using metrics commonly used in classification tasks.

- **Heart Disease Dataset**: The dataset contains 303 samples with 13 features. The target variable represents whether or not a person has heart disease (binary classification).
- **Algorithms Evaluated**: Random Forest (RF), Bi-LSTM, and KNN. These models are selected for their varying approaches to classification problems, with RF being an ensemble learning method, Bi-LSTM using deep learning for sequence data, and KNN relying on distance metrics.

The results from these algorithms will be compared to determine which performs best for this particular task.

Objectives

The main objectives of this project are:

- To build and train three machine learning models (Random Forest, Bi-LSTM, and KNN) for predicting heart disease.

- To evaluate each model using standard metrics, including confusion matrix, classification report, and additional evaluation metrics (accuracy, precision, recall, FPR, FNR, TSS, HSS).
- To compare the performance of these algorithms and make an informed decision on which one is the most suitable for heart disease prediction.

Methodology

The methodology involves several key steps:

1. **Data Preprocessing**:
   The data is first fetched from the UCI repository. It is then preprocessed to handle missing values and formatted appropriately for the machine learning models.
2. **Model Building**:
   Three models are built and evaluated:
   - **Random Forest Classifier**: An ensemble method that works by building multiple decision trees and combining their results.
   - **Bi-LSTM**: A deep learning model that captures sequential patterns in the data. Bi-LSTM is chosen for its ability to handle complex data relationships.
   - **K-Nearest Neighbors (KNN)**: A non-parametric classifier that predicts based on the majority class of nearby data points.
3. **Cross-Validation**:
   10-fold cross-validation is used to evaluate the performance of Random Forest and KNN. This technique ensures that the models are validated on different subsets of the data to avoid overfitting.
4. **Performance Evaluation**:
   Key performance metrics such as accuracy, precision, recall, FPR, FNR, TSS, and HSS are calculated and compared across the three models.

Implementation Summary

1. **Random Forest**:
   - 100 decision trees are used in the model.
   - 10-fold cross-validation is applied to evaluate the model's performance.
   - The confusion matrix and classification report are generated to assess the model.
2. **Bi-LSTM**:
   - A sequential deep learning model is created using Keras.
   - The model uses two layers of bidirectional LSTM to capture patterns in the dataset.
   - The model is trained for 10 epochs with binary cross-entropy loss and Adam optimizer.

3. **KNN**:
    - A KNN classifier with 5 neighbors is used.
    - A simple imputer is applied to handle missing data before training.
    - 10-fold cross-validation is used to evaluate the model's performance.
4. **Evaluation**:
    - The models' performance is evaluated using various metrics, and results are compared in a summary table.

Running Code

Source Code Link: Final Term Project

Code Annotations

The code is divided into clear sections, each with a specific task:

1. **Dataset Loading**:
   The `fetch_ucirepo` function is used to load the Heart Disease dataset. It fetches both features (`X`) and targets (`y`) from the dataset.
2. **Model Training and Evaluation**:
    - Each model (Random Forest, Bi-LSTM, KNN) is created and trained using the respective algorithms and techniques.
    - Cross-validation is applied to assess model performance.
    - Confusion matrix and classification reports are used to evaluate performance.
3. **Metrics Calculation**:
   A custom function `evaluation_metrics` calculates and returns important metrics like accuracy, precision, recall, etc., based on the confusion matrix.
4. **Results Presentation**:
   A DataFrame is created to display the results in a tabular format, making it easy to compare the performance of the models.

Comparison between Algorithms

- **Random Forest**:
    - Performs well with an ensemble approach, reducing variance through averaging multiple decision trees.
    - Provides relatively high accuracy but may overfit with too many trees or on noisy data.

- **Bi-LSTM**:
  - Deep learning model with a good ability to capture sequential patterns and complex relationships.
  - Suitable for more complex datasets but may take longer to train, requiring more computational power.
- **KNN**:
  - Simple to implement and understand. Performs well for small datasets but can struggle with large or high-dimensional data.
  - The performance can degrade if the number of neighbors is not well chosen or if the dataset contains noise.

Which Algorithm to Choose:

- **Best for Simplicity and Speed**: If you need a fast, easy-to-implement model, **KNN** is a good choice. However, it may not perform as well with larger datasets or more complex relationships.
- **Best for Performance with Large Data**: If computational resources are available, **Bi-LSTM** could be the best choice due to its ability to capture complex relationships and handle sequential data. It is ideal for more intricate datasets where patterns are harder to extract with traditional models.
- **Best for Robustness and Generalization**: **Random Forest** strikes a good balance between performance and generalization. It is less prone to overfitting compared to deep learning models and is relatively quick to train.

Conclusion

In this project, three classification algorithms—Random Forest, Bi-LSTM, and KNN—were compared on the Heart Disease dataset. After evaluating their performance based on various metrics, it is concluded that:

- **Random Forest** performed well in terms of general accuracy and robustness.
- **Bi-LSTM** provided the highest performance in terms of capturing complex data relationships but required more computational resources.
- **KNN** is easy to implement and works well for small datasets, but its performance could degrade on larger or more complex datasets.

The choice of algorithm should depend on the dataset's complexity, the available computational resources, and the specific performance criteria that are most important for the task at hand.