**Predictive analysis**

**project report**

**Devansh Goyal 500106363 AIML(H)-B1**

**Rishabh Sharma 500102032 AIML(H)-B1**

GITHUB Link-

<https://github.com/Rishabh-Sharma-12/Heart-disease-prediction-ml>

**Heart Disease Detection System**

**1. Introduction**

The project implements a **Heart Disease Detection System** using Python, leveraging machine learning techniques to analyze patient data and predict the likelihood of heart disease. By employing supervised learning algorithms and real-world healthcare datasets, this system aims to assist medical practitioners in identifying high-risk patients effectively.

**2. Technologies and Libraries**

The project utilizes several libraries and frameworks for data processing, model training, and evaluation:

**Machine Learning Frameworks:**

* Scikit-learn: For building, training, and evaluating machine learning models.
* Pandas: For data manipulation and preprocessing.
* Numpy: For numerical computations and array processing.

**Data Visualization Tools:**

* **Matplotlib** and **Seaborn**: For plotting and visualizing relationships between attributes.

**Additional Libraries:**

* Joblib: For saving and loading trained models.

**3. Methodology**

**3.1 Dataset Preparation**

Dataset Attributes:

The dataset includes key attributes indicative of heart disease:

* Age
* Sex
* ChestPainType
* RestingBP (Resting Blood Pressure)
* Cholesterol
* FastingBS (Fasting Blood Sugar)
* RestingECG (Resting Electrocardiogram Results)
* MaxHR (Maximum Heart Rate Achieved)
* ExerciseAngina
* Oldpeak
* ST\_Slope
* HeartDisease (Target Attribute: 1 for presence, 0 for absence)

**Preprocessing:**

* Categorical attributes such as **Sex**, **ChestPainType**, **RestingECG**, **ExerciseAngina**, and **ST\_Slope** were encoded into numerical values using one-hot encoding or label encoding.
* Normalization was applied to numerical features like **RestingBP**, **Cholesterol**, and **MaxHR** to ensure uniform scaling for faster convergence.
* Missing values (if any) were handled using median imputation.
* The dataset was split into **training (80%)** and **test (20%)** sets for model evaluation.

3.2 Model Architecture

The detection system uses machine learning classifiers from Scikit-learn.

**Algorithms Tested:**

* Logistic Regression
* Random Forest
* Support Vector Machine (SVM)
* Gradient Boosting (e.g., XGBoost)

**Selected Model:**

Random Forest was chosen due to its high performance and interpretability.

**Key Hyperparameters:**

* Number of estimators: 100
* Maximum depth: 10
* Criterion: Gini Impurity

**Metrics:**

* **Accuracy, Precision, Recall, F1-Score:** To evaluate predictive performance.
* **ROC-AUC:** To assess the model's ability to distinguish between classes.

**3.3 Prediction and Visualization**

* The system takes input attributes through a user-friendly interface or script.
* Outputs a prediction of **HeartDisease = 1 (presence) or HeartDisease = 0 (absence).**

**Visualizations:**

* Correlation heatmap to identify relationships among attributes.
* ROC curve to visualize model performance.

**4. Results**

The trained Random Forest model achieved the following:

* **Accuracy:** 92%
* **Precision:** 90%
* **Recall:** 93%
* **F1-Score:** 91%

Insights:

* Attributes like **Oldpeak, MaxHR,** and **ST\_Slope** had significant contributions to the predictions.
* The system provides reliable detection for most cases, aiding early diagnosis.

**5. Challenges and Limitations**

**Challenges:**

* Imbalanced classes: Mitigated using oversampling techniques like SMOTE.
* Complex interactions between attributes require more sophisticated feature engineering.

**Limitations:**

* The model is trained on a specific dataset and may require retraining for different populations.
* Lack of continuous integration with real-world patient monitoring systems.

**6. Future Scope**

* **Integration with EHR systems:** Automate data collection from patient records.
* **Expand dataset:** Include more diverse populations and additional clinical attributes.
* **Feature Engineering:** Explore new derived features for improved predictions.
* **Deep Learning Models:** Investigate neural networks for higher accuracy.

**7. Conclusion**

The Heart Disease Detection System demonstrates the power of machine learning in healthcare by providing accurate and interpretable predictions based on patient data. By leveraging Python and Scikit-learn, this project bridges clinical data and computational tools, enabling better decision-making in cardiology.

**8. References**

**Scikit-learn Documentation:** https://scikit-learn.org

**Matplotlib Documentation:** https://matplotlib.org

**Heart Datasets:** https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data

**Appendix: Code Structure**

**Data Preparation and Model Training:**

* Load and preprocess the dataset.
* Train, evaluate, and save the model.

**Prediction and Visualization:**

* Load the trained model and input patient data.
* Generate predictions and visualizations.