

**Project**

**Titled**

|  |
| --- |
| **Heart Disease Prediction** |

|  |  |
| --- | --- |
| ***Class and Section*** | BS Mathematics Semester(A) |
| ***Session*** | 2021-2025 |
| ***Submitted to*** | Dr Nimra Tariq |
| ***Submission Deadline*** | 14-05-2024 |

***Submitted by:***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Sr No*** | ***Name*** | ***Roll No*** | ***Signature*** |
| ***1*** | ***Khansa Fatima*** | ***BSMM-F21-011*** |  |
| ***2*** | ***Fiza Hanif*** | ***BSMM-F21-007*** |  |
| ***3*** | ***Kainat Asghar*** | ***BSMM-F21-019*** |  |
| ***4*** | ***Waseem Akram*** | ***BSMM-F21-003*** |  |

**Marks Obtained:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Signature of Project Advisor:** \_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| 1. **Abstract** |
| This report studies class of heart disease prediction using different python libraries and it’s importance in various fields. This study emphasizes the preprocessing phase using Python to ensure accurate predictive modeling. Key steps include data cleaning, handling missing values, feature selection, normalization, and transformation. Leveraging libraries like Pandas, NumPy, and Scikit-learn, we enhance data quality for predictive models. Our results show that thorough preprocessing significantly improves model performance, establishing a solid foundation for predictive analytics. |
| 1. **Introduction** |
| **Heart Disease Prediction:**  Heart disease prediction involves using statistical and computational methods to forecast the likelihood of an individual developing heart disease based on various risk factors. These risk factors can include demographic information (age, gender), clinical measurements (blood pressure, cholesterol levels), medical history (diabetes, family history of heart disease), and lifestyle habits (smoking, physical activity, diet).  **Example:**  **Install Necessary Libraries:**  Ensure you have the necessary libraries installed. The libraries we used in this process are Pandas, Numpy and Scikit-learn for different purposes.   * **Bash**   pip install pandas numpy scikit-learn  **Load the Dataset:**  We use Pandas to load and inspect the dataset. This dataset is often available as a CSV file.   * **Python**   import pandas as pd  # Load the dataset  url = "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data"  columns = [  "age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach",  "exang", "oldpeak", "slope", "ca", "thal", "target"  ]  df = pd.read\_csv(url, names=columns)  # Display the first few rows of the dataset  print(df.head())  **Handle Missing Values:**  The dataset may contain missing or ambiguous values, which need to be handled appropriately.  python  # Replace '?' with NaN  df.replace('?', pd.NA, inplace=True)  # Convert all columns to numeric values  df = df.apply(pd.to\_numeric, errors='coerce')  # Handle missing values (e.g., by filling with the mean or median)  df.fillna(df.median(), inplace=True)  print(df.info()) # Check for any remaining issues  **Prepare the Data:**  Split the dataset into features (X) and target (y), and then into training and testing sets.  python  from sklearn.model\_selection import train\_test\_split  # Define features and target  X = df.drop(columns="target")  y = df["target"]  # Split the data into training and testing sets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  print(f"Training set size: {X\_train.shape}")  print(f"Testing set size: {X\_test.shape}")  **Steps of Preprocessing:**   1. **Data Cleaning**: Remove duplicates and correct inconsistent entries. For example, ensure that all patient records are complete and accurate, eliminating any outliers that do not represent typical patient data. 2. **Handling Missing Values**: Address missing data by using imputation methods such as filling in with the mean, median, or mode of the relevant feature, or employing predictive algorithms to estimate missing values. 3. **Feature Selection:** Select the most relevant features that contribute to heart disease prediction, such as age, cholesterol levels, blood pressure, and family history. Techniques like correlation analysis and recursive feature elimination help identify these key features. 4. **Normalization and Scaling:** Normalize numerical features to ensure they contribute equally to the model. For instance, scale features like cholesterol levels and blood pressure using Min-Max scaling or Z-score standardization. 5. **Data Transformation:** Transform categorical variables into numerical format using techniques like one-hot encoding for variables such as gender or chest pain type. Apply transformations like log scaling to features with skewed distributions. 6. **Data Splitting:** Split the dataset into training and testing sets to evaluate the model’s performance accurately. A common split is 70% for training and 30% for testing, ensuring that the model is trained on a diverse and representative subset of the data. |
| 1. **Experimental work/Coding:**   Here we use python and its libraries for preprocessing of the data using data in the form of csv file from Kaggle and the coding we used is given below: |
|  |
| 1. **Results:**   Result we got for this data set are: |
| 1. **Conclusion :** |
| In heart disease prediction, the preprocessing phase is critical to ensure the accuracy and reliability of predictive models. This study highlights the importance of systematic data cleaning, effective handling of missing values, careful feature selection, and appropriate normalization and scaling. By transforming raw medical data into a structured format, we enhance the dataset's quality, making it suitable for machine learning algorithms. Employing Python's robust libraries like Pandas, NumPy, and Scikit-learn, we can efficiently execute these preprocessing steps, resulting in improved model performance. Thorough preprocessing not only boosts prediction accuracy but also lays a solid foundation for subsequent stages of predictive analytics, ultimately contributing to better patient outcomes and more informed healthcare decisions. |
|  |