A Mini Project

Report *On*

Comparative Analysis of

Heart Disease Prediction

In Subject: Machine Learning

Contents

Abstract

The Binary Logistic Regression Analysis BLRA technique has been used and applied for building the best model for Heart disease data using best subsets regression and stepwise procedures and depending on some laboratory tests such as Resting blood pressure, Serum cholestoral in mg/dl, Sasting blood sugar > 120 mg/dl, Resting electrocardiographic results (values 0,1,2) which represents explanatory variables. Also, the technique has used for classifying persons into two groups which are infected and non-infected with Heart disease. A random sample size consists of 1025 persons has been selected which represents 499 of uninfected and 526 of infected persons. The results of the analysis showed that the percentage of visible correct classification rate was about 80% which represents the high ability of the model for classification.

Chapter 1 – Introduction

1.1 Introduction

The forecast of cardiovascular disease, one of the most common heart diseases, is considered to be one of the most significant topics in the analysis of clinical data. According to the World Health Organization (WHO), cardiovascular diseases (CVDs) kills about 31% of the world's population each year, with older people at greater risk than other age groups. Through applying the technology of data mining, a new idea is provided for the prediction of heart disease, extracting clinical attributes and pathological data from large medical data sets, and generating biological hypotheses. At present, some studies have applied data mining technology to the prediction of heart disease, but there are limited studies on the important features of cardiovascular disease, while logistic regression can extract the risk factors of disease and predict the incidence probability of patients in real time. This study aims to determine the important characteristics and incidence probability of heart disease prediction, and compare the accuracy of the logistic regression algorithm used with other existing research algorithms, such as Naive Bayes, SVM and Neural Network, to determine the feasibility of the logistic regression algorithm in predicting heart disease.

1.2 Requirements

Hardware Requirement for Development of Project: (minimum)

- System Processor: Pentium P4
- Motherboard: Genuine Intel
- Memory: 512 MB of RAM, 1GB recommended.
- Display: 1024x 768 or higher-resolution display with 16 bits colors of android mobile phone.

Software Requirement for Development of Project: (minimum)

- Operating system: Windows XP or higher.

- Software: jupyter notebook
-

1.3 Design and Problem Statement

Presently, the major challenge of the medical industry is to predict the cardio vascular disease with less expensive and more reliable method to avoid the compounding effect of the disease in low income or developing countries. The early detection not only reduce the cost but also improves the quality of life.

The purpose of this project is to predict whether the person has heart disease or not. The model will analyze the given input i.e. independent variables and on that basis it will give prediction. Our main goal is building a model with higher accuracy so that it will predict the heart disease more accurately.

1.4 Aim and Scope

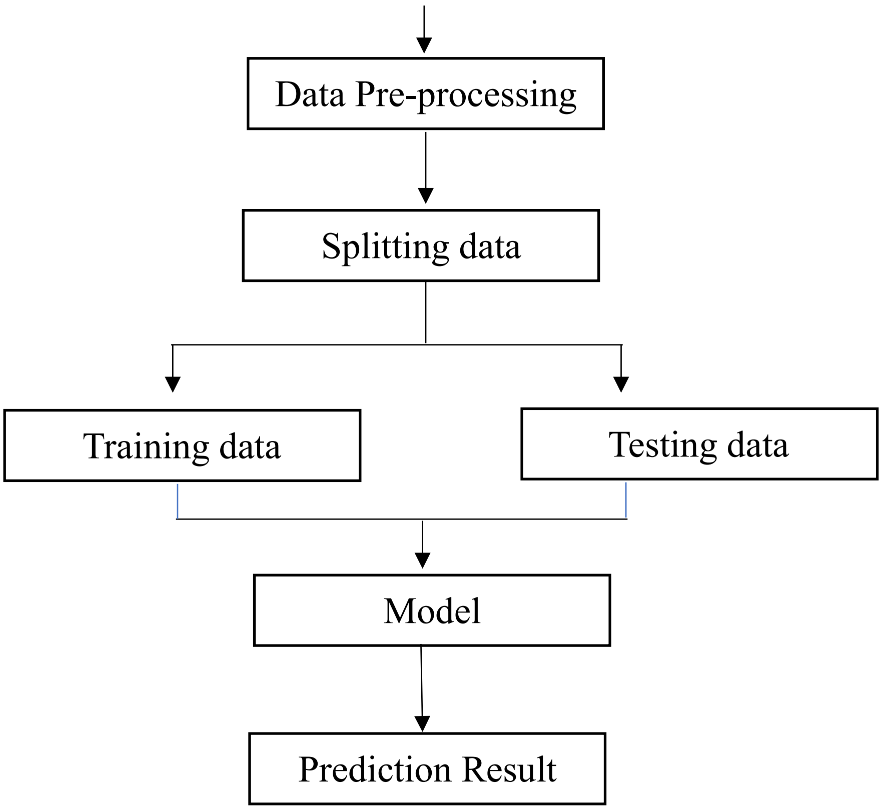
The aim of this research is to develop an efficient way to predict the presence of the cardiovascular disease. The steps as mentioned below. a. The UCI dataset is used to predict the disease.

b. The performance of the model is evaluated by 80:20 training and testing ratio of dataset.
c. To check the behaviour of the model with training and testing data.

1.5 Proposed Work

Data Acquisition

Chapter-2



Methodology

2.1 Algorithm

1. Logistic Regression : A statistical model that predicts the probability of a binary outcome based on one or more predictor variables. It is used for classification tasks and outputs probabilities that a given input point belongs to a certain class.

2. Decision Tree : A model that uses a tree-like graph of decisions and their possible consequences. It is used for both classification and regression tasks. Each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a continuous output.

3. Random Forest : An ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set.

4. Naive Bayes: A classification technique based on Bayes' Theorem with an assumption of independence among predictors. It is a simple but surprisingly powerful algorithm for predictive modeling.

5. K-Nearest Neighbours (KNN): A non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output is a class membership (classification) or a property value (regression) which is determined by majority vote or averaging the values of the k nearest neighbors.

6. Support Vector Machine (SVM): A powerful and versatile supervised machine learning model that is used for both classification and regression. It works by finding the hyperplane that best divides a dataset into classes in terms of a margin that is as wide as possible.2.2 Dataset

- The Personal Key Indicators of Heart Disease dataset contains 320K rows and 18 columns. It is a cleaned, smaller version of the 2020 annual CDC

(Centers for Disease Control and Prevention) survey data of 400k adults. For each patient (row), it contains the health status of that individual. The data was collected in the form of surveys conducted over the phone. Each year, the CDC calls around 400K U.S residents and asks them about their health status, with the vast majority of questions being yes or no questions.

- Smoking : We can see that the composition of the subset of population with Heart Disease has a higher proportion of smokers.
- Alcohol Drinking: The distribution of Yes/No is almost the same in both sets with and without Heart Disease.
- Stroke : It is clear that the population with Heart Disease has a higher proportion of people who have had a stroke, which suggests a correlation between the two.
- DiffWalkin : There are a higher number of people who have Difficulty Walking with Heart Disease as opposed to those who do not.
- Sex : It appears that Males make up a higher proportion of the population with Heart Disease
- Age Category: We see that the occurences of HeartDisease are more common in older age groups
- Race: The distribution of Race with respect to HeartDisease is nearly the same, suggesting weak correlation
- Diabetic : There is a higher proportion of diabetics in subset of people with Heart Disease
- Physical Activity : There are more physically inactive people with Heart Disease as compared to physically active people
- GenHealth : People without Heart Disease have better General Health than those with Heart Disease
- Asthma: The distribution for with/without HeartDisease is almost the same, suggesting weak correlation
- KidneyDisease: There is a bigger fraction of people with KidneyDisease and Health Disease as compared to those without HeartDisease
- SkinCancer : There is a bigger fraction of people with SkinCancer and Health Disease as compared to those without HeartDiseaseThe slope of the peak exercise ST segment

1	HeartDise	BMI	Smoking	AlcoholDr	Stroke	PhysicalHe	MentalHe	DiffWalkir	Sex	AgeCatego	Race	Diabetic	PhysicalAc	GenHealth	SleepTime	Asthma	KidneyDis	SkinCancer
2	No	16.6	Yes	No	No	3	30	No	Female	55-59	White	Yes	Yes	Very good	5	Yes	No	Yes
3	No	20.34	No	No	Yes	0	0	No	Female	80 or olde	White	No	Yes	Very good	7	No	No	No
4	No	26.58	Yes	No	No	20	30	No	Male	65-69	White	Yes	Yes	Fair	8	Yes	No	No
5	No	24.21	No	No	No	0	0	No	Female	75-79	White	No	No	Good	6	No	No	Yes
6	No	23.71	No	No	No	28	0	Yes	Female	40-44	White	No	Yes	Very good	8	No	No	No
7	Yes	28.87	Yes	No	No	6	0	Yes	Female	75-79	Black	No	No	Fair	12	No	No	No
8	No	21.63	No	No	No	15	0	No	Female	70-74	White	No	Yes	Fair	4	Yes	No	Yes
9	No	31.64	Yes	No	No	5	0	Yes	Female	80 or olde	White	Yes	No	Good	9	Yes	No	No
10	No	26.45	No	No	No	0	0	No	Female	80 or olde	White	No, borde	No	Fair	5	No	Yes	No
11	No	40.69	No	No	No	0	0	Yes	Male	65-69	White	No	Yes	Good	10	No	No	No
12	Yes	34.3	Yes	No	No	30	0	Yes	Male	60-64	White	Yes	No	Poor	15	Yes	No	No
13	No	28.71	Yes	No	No	0	0	No	Female	55-59	White	No	Yes	Very good	5	No	No	No
14	No	28.37	Yes	No	No	0	0	Yes	Male	75-79	White	Yes	Yes	Very good	8	No	No	No
15	No	28.15	No	No	No	7	0	Yes	Female	80 or olde	White	No	No	Good	7	No	No	No
16	No	29.29	Yes	No	No	0	30	Yes	Female	60-64	White	No	No	Good	5	No	No	No
17	No	29.18	No	No	No	1	0	No	Female	50-54	White	No	Yes	Very good	6	No	No	No
18	No	26.26	No	No	No	5	2	No	Female	70-74	White	No	No	Very good	10	No	No	No
19	No	22.59	Yes	No	No	0	30	Yes	Male	70-74	White	No, borde	Yes	Good	8	No	No	No
20	No	29.86	Yes	No	No	0	0	Yes	Female	75-79	Black	Yes	No	Fair	5	No	Yes	No
21	No	18.13	No	No	No	0	0	No	Male	80 or olde	White	No	Yes	Excellent	8	No	No	Yes
22	No	21.16	No	No	No	0	0	No	Female	80 or olde	Black	No, borde	No	Good	8	No	No	No
23	No	28.9	No	No	No	2	5	No	Female	70-74	White	Yes	No	Very good	7	No	No	No
24	No	26.17	Yes	No	No	0	15	No	Female	45-49	White	No	Yes	Very good	6	No	No	No
25	No	25.82	Yes	No	No	0	30	No	Male	80 or olde	White	Yes	Yes	Fair	8	No	No	No
26	No	25.75	No	No	No	0	0	No	Female	80 or olde	White	No	Yes	Very good	6	No	No	Yes
27	No	29.18	Yes	No	No	30	30	Yes	Female	60-64	White	No	No	Poor	6	Yes	No	No
28	No	34.34	Yes	No	No	21	8	Yes	Female	65-69	White	No	Yes	Fair	9	No	No	No
29	No	31.66	Yes	No	No	5	0	No	Male	60-64	White	No	Yes	Very good	5	No	No	No
30	No	24.89	No	No	No	1	0	No	Female	55-59	White	No	Yes	Very good	7	No	No	No

2.3 Implementation

Logistic Regression

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import ConfusionMatrixDisplay, classification_report, accuracy_score

for i in range(len(X_train_list)):
    print("---------------------------------------------------------------------------")
    print(f"Model with training data {i} ({data_desc[i]}):\n".upper())

    # making model for logistic regression
    clf_LR= LogisticRegression(random_state=0).fit(X_train_list[i], y_train_list[i])
    pred = clf_LR.predict(X_test)

    print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, pred)))
    print("Classification report:\n")
    print(classification_report(y_test,pred))

    print("Confusion Matrix:")
    ConfusionMatrixDisplay.from_predictions(y_test, pred, cmap='YlOrRd')
    plt.show()
    print("---------------------------------------------------------------------------")
```

[44]

```
---------------------------------------------------------------------------
MODEL WITH TRAINING DATA 0 (IMBALANCED):

Model accuracy score: 0.9128
```

Decision Tree

Decision Tree

```python
from sklearn.tree import DecisionTreeClassifier

for i in range(len(X_train_list)):
    print("---------------------------------------------------------------------------")
    print(f"Model with training data-{i} ({data_desc[i]}):\n")

    clf_dtc = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=0)
    clf_dtc.fit(X_train_list[i], y_train_list[i])

    pred = clf_dtc.predict(X_test)
    print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, pred)))
    print("Classification report:\n")
    print(classification_report(y_test,pred))

    print("Confusion Matrix:")
    ConfusionMatrixDisplay.from_predictions(y_test, pred, cmap='YlOrRd')
    plt.show()
    print("---------------------------------------------------------------------------")
```

```
---------------------------------------------------------------------------
Model with training data-0 (imbalanced):

Model accuracy score: 0.9116
```

Random Forest

Random Forest

```python
from sklearn.ensemble import  RandomForestClassifier

for i in range(len(X_train_list)):
    print("---------------------------------------------------------------------------")
    print(f"Model with training data-{i} ({data_desc[i]}):\n")

    clf_RF= RandomForestClassifier().fit(X_train_list[i], y_train_list[i])

    pred = clf_RF.predict(X_test)
    print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, pred)))
    print("Classification report:\n")
    print(classification_report(y_test,pred))

    print("Confusion Matrix:")
    ConfusionMatrixDisplay.from_predictions(y_test, pred, cmap='YlOrRd')
    plt.show()
    print("---------------------------------------------------------------------------")
```

```
---------------------------------------------------------------------------
Model with training data-0 (Imbalanced):

Model accuracy score: 0.9016
```

Naïve Bayes

Naive Bayes

```python
from sklearn.naive_bayes import GaussianNB

for i in range(len(X_train_list)):
    print("-------------------------------------------------------------------------------")
    print(f"Model with training data-{i} ({data_desc[i]}):\n")

    clf_gnb = GaussianNB()
    pred = clf_gnb.fit(X_train_list[i], y_train_list[i]).predict(X_test)

    print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, pred)))
    print("Classification report:\n")
    print(classification_report(y_test,pred))

    print("Confusion Matrix:")
    ConfusionMatrixDisplay.from_predictions(y_test,pred, cmap='YlOrRd')
    plt.show()
    print("-------------------------------------------------------------------------------")
```

```
---------------------------------------------------------------------------
Model with training data-0 (imbalanced):

Model accuracy score: 0.8473
```

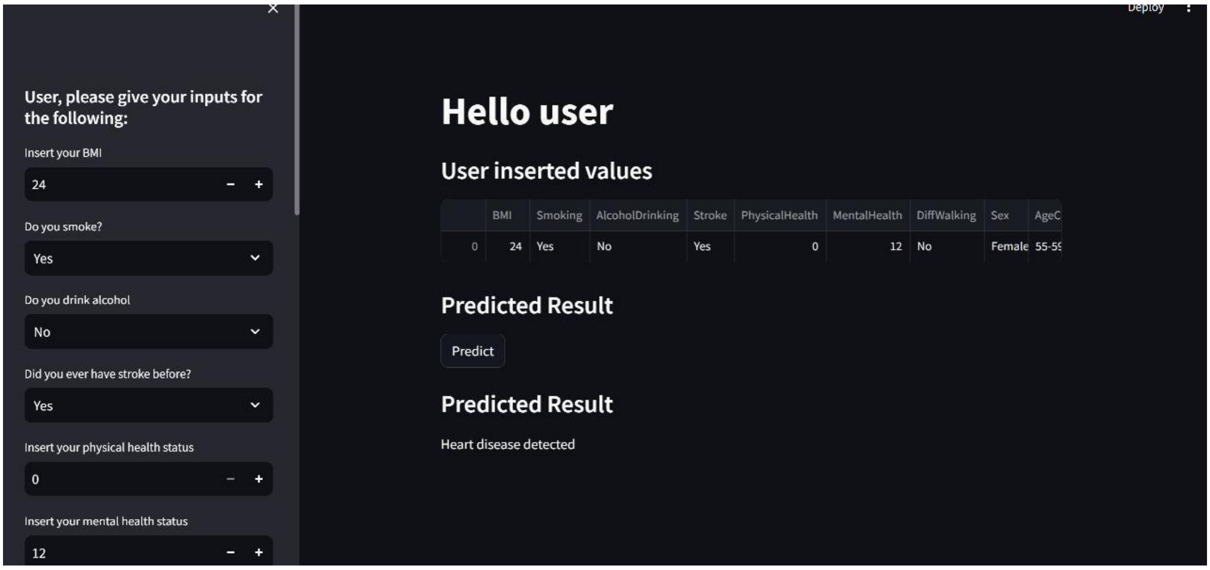
K – Nearest Neigbours (KNN)

K-Nearest Neighbours (KNN)

```python
from sklearn.neighbors import KNeighborsClassifier

for i in range(len(X_train_list)):
    print("-------------------------------------------------------------------------------")
    print(f"Model with training data-{i} ({data_desc[i]}):\n")

    clf_knn = KNeighborsClassifier(n_neighbors=5)
    clf_knn.fit(X_train_list[i], y_train_list[i])
    pred = clf_knn.predict(X_test)

    print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, pred)))
    print("Classification report:\n")
    print(classification_report(y_test,pred))

    print("Confusion Matrix:")
    ConfusionMatrixDisplay.from_predictions(y_test,pred, cmap='YlOrRd')
    plt.show()
    print("-------------------------------------------------------------------------------")
```

```
---------------------------------------------------------------------------
Model with training data-0 (imbalanced):

Model accuracy score: 0.9043
```

2.3 Results and Output



Model	**Accuracy**
Navie Bayies	84.73%
Random Forest	90.16%
K-Nearest Neighbour	90.43%
Decision Tree	91.16%
Support Vector Machine	89.86%

Chapter-3

Conclusion:

One of the important areas in industry of medical is prediction of cardiovascular disease, with the available data of the patient to predict the absence and presence of cardia disease. There are several techniques and methods are present for prediction of cardiovascular disease. In this research, Logistic Regression supervised ML algorithm are used to classify the heart disease. To improve the performance, pre-processing of corpus like Cleaning, finding the missing values are done. The vital part is feature selection, which increase the accuracy of algorithm and even focus on the behavior of the algorithm. As the behavior of Logistic regression is as training increases the accuracy of prediction also increased. The LR classifier achieved 84.87% of accuracy for training data and 80.48% accuracy for testing data with training 80% and testing 20%.

References:

- https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
- https://www.kaggle.com/code/ahmadrafiee/handling-imbalanced-1class-weight-halvinggrid/notebook
- https://www.ibm.com/in-en/topics/logistic-regression
- https://iopscience.iop.org/article/10.1088/1742-6596/1769/1/012024/pdf
- https://www.sciencedirect.com/science/article/pii/S2666285X22000449