Heart Disease Prediction

Eshita Pradhan
Computer Science & Engineering
International Institute Of Information Technology
Naya Raipur, India
eshita21100@iiitnr.edu.in

Riya Yadav
Computer Science & Engineering
International Institute Of Information Technology
Naya Raipur, India
riya21100@iiitnr.edu.in

Abstract:

Healthcare expenditures are overwhelming national and corporate budgets due to asymptomatic diseases including cardiovascular diseases. Therefore, there is an urgent need for early detection and treatment of such diseases. Machine learning is one of the trending technologies used in many spheres around the world, including the healthcare industry, for predicting diseases. The aim of this study is to identify the most significant predictors of heart disease and predict the overall risks by using linear regression and logistic regression and compare their accuracy. Thus, the binary logistic model which is one of the classification algorithms in machine learning is used in this study to identify the predictors. Further, data analysis is carried out in Python using Google Colab notebook in order to validate the linear and logistic regression.

Keywords-machine learning, linear regression, logistic regression, classification algorithms, heart disease

1. Introduction

The number of deaths due to cardiovascular diseases and increased by 41% between 1990 and 2013, climbing from 12.3 million deaths to 17.3 million deaths globally. In addition to that, half of the deaths in the United States and other developed countries are due to the same issue. Therefore, early detection of heart diseases is required to reduce health complications. Machine learning has been widely used in the modern healthcare sector for diagnosing and predicting the presence of diseases using data models. Linear and Logistic regression is one such relatively used machine learning algorithm for studies involving risk assessment of complex diseases. Thus, the study intends to identify the most significant predictors of cardiovascular diseases and predict the overall risk by using linear and logistic regression and analyzing their accuracy.

2. Background of the study

The dataset used for the logistic regression analysis is available on the Kaggle website(https://www.kaggle.com), from an ongoing cardiovascular study in Framingham, Massachusetts. The classification goal of this study is to predict whether the patient has a 10-year risk of future heart disease. The Framingham dataset consists of 4238 records of patient data and 15 attributes. The data analysis is carried out in Python programming by using Google Colab which is a more flexible and powerful data science application software.

3. Machine Learning (ML)

Machine learning is widely used in almost many fields in the world including the healthcare sector. Machine learning is an application of artificial intelligence (AI) that provides systems with the ability to automatically learn and improve from experience without being explicitly programmed.Further, machine learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world. There are two major categories of problems often solved by machine learning i.e.

regression and classification. Mainly, the regression algorithms are used for numeric data and classification problems include binary and multi-category problems. Machine learning algorithms are further divided into two categories such as supervised learning and unsupervised learning. Basically, supervised learning is performed by using prior knowledge in output values whereas unsupervised learning does not have predefined labels hence the goal of this is to infer the natural structures within the dataset. Therefore, the selection of a machine learning algorithm needs to be carefully evaluated.

4. Methodology

4.1 Workflow of Machine Learning ModelBuilding

Figure 3 indicates the steps followed in order to build the linear and logistic regression model in machine learning.

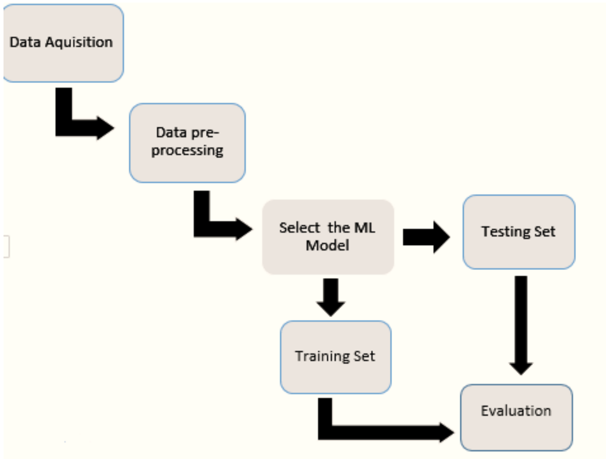


Figure 3: Workflow of Logistic Regression Model

4.1.1 Data Acquisition

The dataset is collected from the Kaggle website.

4.1.2 Data Pre-Processing

In order to build up a more accurate ML model, data preprocessing is required. Data preprocessing is the process of cleaning the data. This includes the identification of missing data, noisy data and inconsistent data.

4.1.3 Select Machine Learning Model

The pre-processed data are identified using machine learning algorithms.

a) Input Variables of the study

The data set consists of 14 input variables and predicted values. ML model is based on the identification of the dependent variable. First, we have used a linear regression model. Probability is ranged between 0 and 1, where the probability of something certain to happen is 1, and 0 is something unlikely to happen. But in linear regression, we are to predict an absolute number, which can range outside 0 and 1. That's why we need to use binary logistic regression which is one of the classification algorithms due to the target variable being categorical.

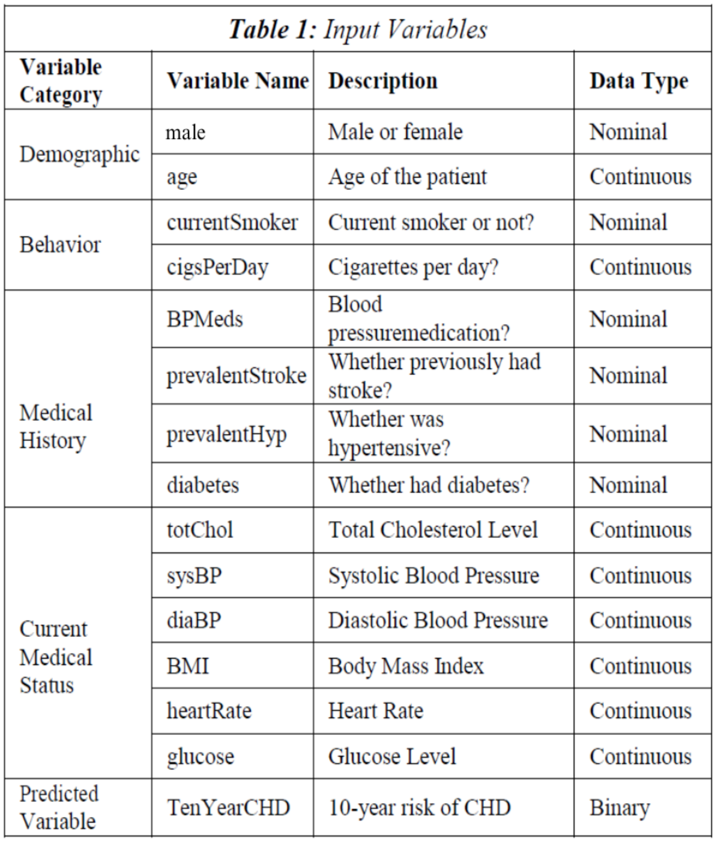


***Table 1:** Input Variables*

Variable Category	Variable Name	Description	Data Type
Demographic	male	Male or female	Nominal
	age	Age of the patient	Continuous
Behavior	currentSmoker	Current smoker or not?	Nominal
	cigsPerDay	Cigarettes per day?	Continuous
Medical History	BPMeds	Blood pressuremedication?	Nominal
	prevalentStroke	Whether previously had stroke?	Nominal
	prevalentHyp	Whether was hypertensive?	Nominal
	diabetes	Whether had diabetes?	Nominal
Current Medical Status	totChol	Total Cholesterol Level	Continuous
	sysBP	Systolic Blood Pressure	Continuous
	diaBP	Diastolic Blood Pressure	Continuous
	BMI	Body Mass Index	Continuous
	heartRate	Heart Rate	Continuous
	glucose	Glucose Level	Continuous
Predicted Variable	TenYearCHD	10-year risk of CHD	Binary

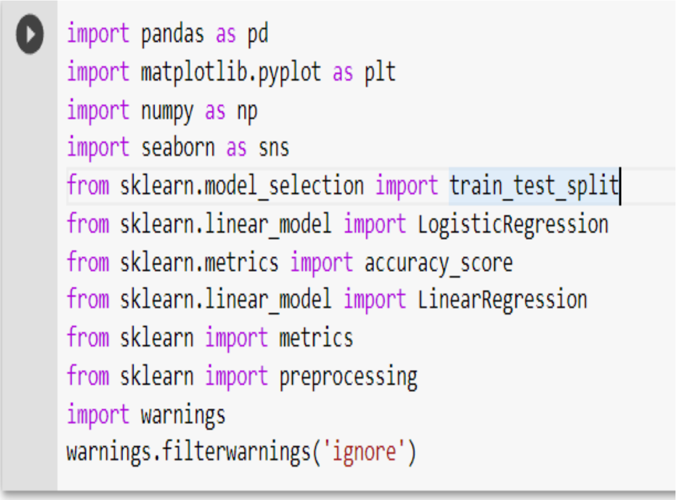
Table 1: Input Variable

5. Data Analysis

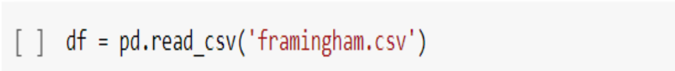
It is carried out using Colab using Python. The following steps were implemented in order to process the linear and logistics regression model.

5.1 Loading Data and Other Required Libraries

It has loaded the heart prediction data using the Framingham CSV file into Google Colab in order to build the linear and logistic regression model. In addition to that, required libraries which are used as supportive applications are loaded. It has removed the education field from the database.



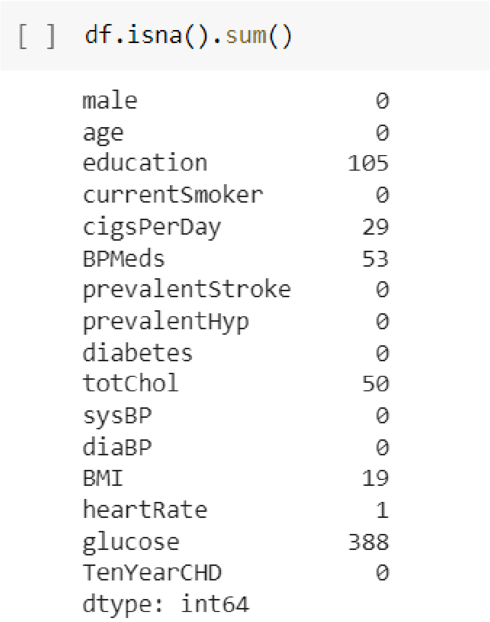
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn import preprocessing
import warnings
warnings.filterwarnings('ignore')
```



```
[ ] df = pd.read_csv('framingham.csv')
```

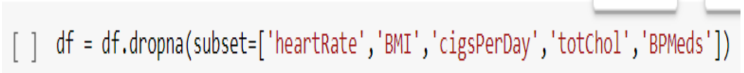
5.2 Identify Missing Values

Further, the number of missing values has been identified for cleaning the existing dataset. The summarized total number of missing values based on the attributes are given below.

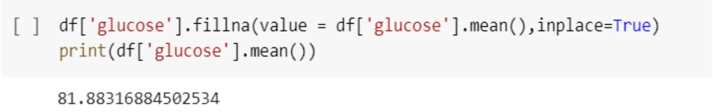


```
[ ]  df.isna().sum()

     male                 0
     age                  0
     education          105
     currentSmoker        0
     cigsPerDay          29
     BPMeds              53
     prevalentStroke      0
     prevalentHyp         0
     diabetes             0
     totChol             50
     sysBP                0
     diaBP                0
     BMI                 19
     heartRate            1
     glucose            388
     TenYearCHD           0
     dtype: int64
```

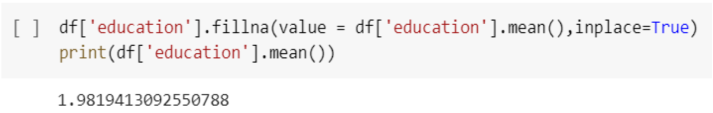
Then, the total percentage of missing values in the column was identified using Pandas Dataframe 3.58 % of the entire dataset (rows) with missing values are excluded. It has used the Pandas dropna() method which was used to analyze the drop rows/columns with Null values. In this case, the missing values in the 'glucose' and 'education' column are being filled with the mean value of the non-missing values in that column.



```
[ ] df = df.dropna(subset=['heartRate','BMI','cigsPerDay','totChol','BPMeds'])
```

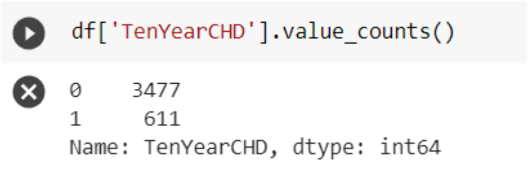


```
[ ] df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
    print(df['glucose'].mean())

    81.88316884502534
```



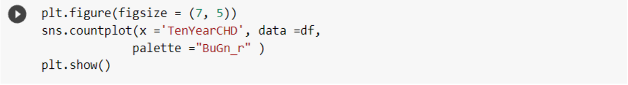
```
[ ] df['education'].fillna(value = df['education'].mean(),inplace=True)
    print(df['education'].mean())

    1.9819413092550788
```

The descriptive figures related to 10-year risk of CHD have indicated below.

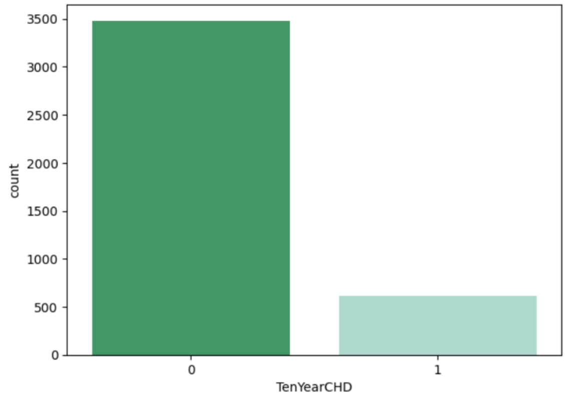


```
df['TenYearCHD'].value_counts()
```

```
0    3477
1     611
Name: TenYearCHD, dtype: int64
```

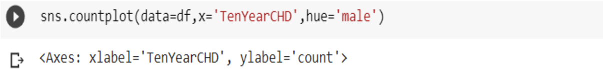


```
plt.figure(figsize = (7, 5))
sns.countplot(x ='TenYearCHD', data =df,
            palette ="BuGn_r" )
plt.show()
```



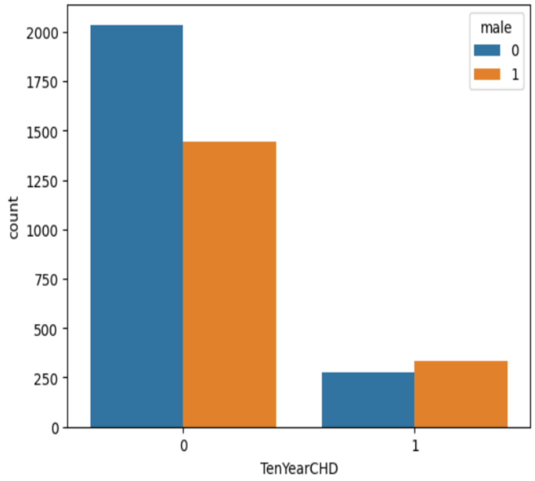
According to the above data, there are 3477 patients with no heart disease and 611 patients with a risk of heart disease.

The x-axis TenYearCHD is a binary variable indicating whether or not a person developed CHD in ten years. The hue male represents the gender of the person.

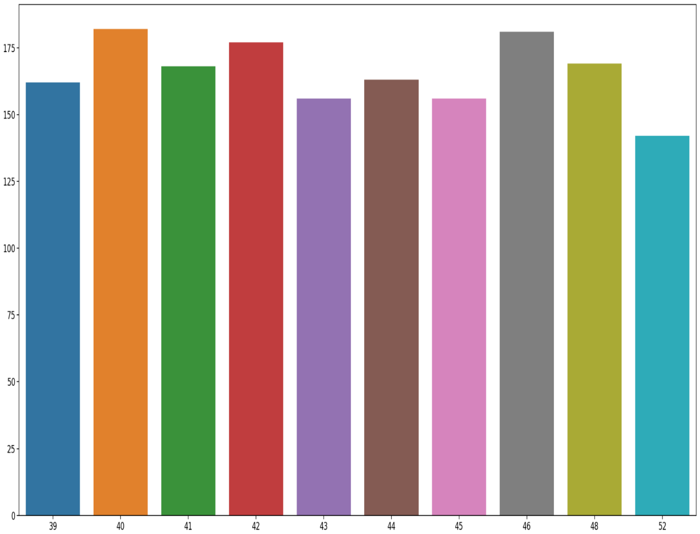


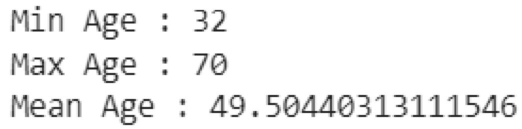
```
sns.countplot(data=df,x='TenYearCHD',hue='male')
```

```
<Axes: xlabel='TenYearCHD', ylabel='count'>
```



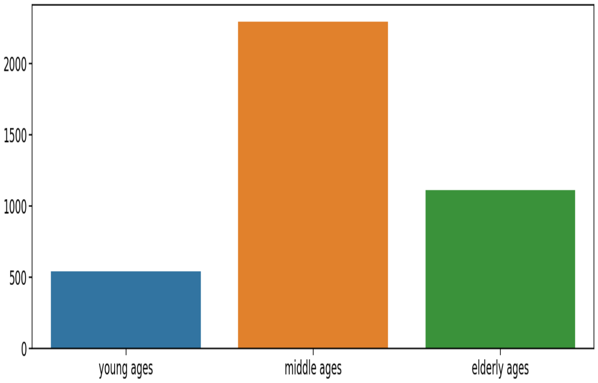
Here we will be checking the 10 ages and their counts.



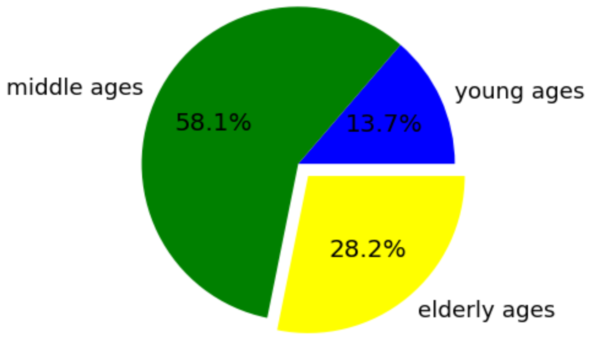


```
Min Age : 32
Max Age : 70
Mean Age : 49.50440313111546
```

We can observe that the minimum age of patient in the data set is 32 years and maximum age is 70 years.We should divide the age attribute into 3 parts i.e. young,middle and elder ages.

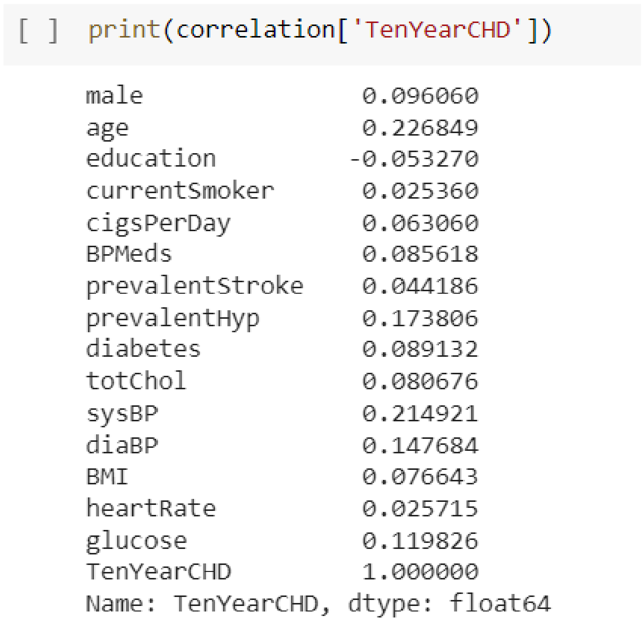


A pie chart is generated which states that middle-aged people have higher chances to have heart disease.



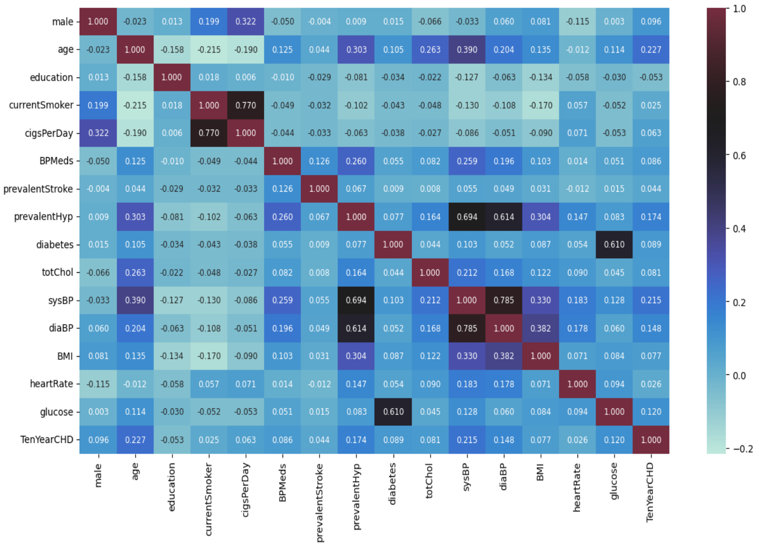
5.3 Implementing Linear Regression

Once we have collected the data, we would use linear regression to build a model that predicts the risk of heart disease based on these risk factors. The model would take the form of a linear equation. We have Calculates the pairwise correlation between columns (variables) of a Pandas DataFrame. The resulting correlation matrix is a square matrix where each element represents the correlation between two columns.



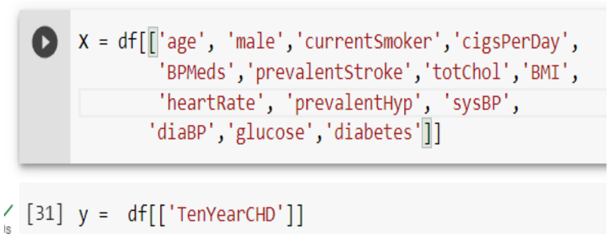
```
[ ]  print(correlation['TenYearCHD'])

     male                 0.096060
     age                  0.226849
     education           -0.053270
     currentSmoker        0.025360
     cigsPerDay           0.063060
     BPMeds               0.085618
     prevalentStroke      0.044186
     prevalentHyp         0.173806
     diabetes             0.089132
     totChol              0.080676
     sysBP                0.214921
     diaBP                0.147684
     BMI                  0.076643
     heartRate            0.025715
     glucose              0.119826
     TenYearCHD           1.000000
     Name: TenYearCHD, dtype: float64
```

The resulting correlation matrix would provide information on the linear relationships between various risk factors for coronary heart disease (CHD).



	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
male	1.000	-0.023	0.013	0.199	0.322	-0.050	-0.004	0.009	0.015	-0.066	-0.033	0.060	0.081	-0.115	0.003	0.096
age	-0.023	1.000	-0.158	-0.215	-0.190	0.125	0.044	0.303	0.105	0.263	0.390	0.204	0.135	-0.012	0.114	0.227
education	0.013	-0.158	1.000	0.018	0.006	-0.010	-0.029	-0.081	-0.034	-0.022	-0.127	-0.063	-0.134	-0.058	-0.030	-0.053
currentSmoker	0.199	-0.215	0.018	1.000	0.770	-0.049	-0.032	-0.102	-0.043	-0.048	-0.130	-0.108	-0.170	0.057	-0.052	0.025
cigsPerDay	0.322	-0.190	0.006	0.770	1.000	-0.044	-0.033	-0.063	-0.038	-0.027	-0.086	-0.051	-0.090	0.071	-0.053	0.063
BPMeds	-0.050	0.125	-0.010	-0.049	-0.044	1.000	0.126	0.260	0.055	0.082	0.259	0.196	0.103	0.014	0.051	0.086
prevalentStroke	-0.004	0.044	-0.029	-0.032	-0.033	0.126	1.000	0.067	0.009	0.008	0.055	0.049	0.031	-0.012	0.015	0.044
prevalentHyp	0.009	0.303	-0.081	-0.102	-0.063	0.260	0.067	1.000	0.077	0.164	0.694	0.614	0.304	0.147	0.083	0.174
diabetes	0.015	0.105	-0.034	-0.043	-0.038	0.055	0.009	0.077	1.000	0.044	0.103	0.052	0.087	0.054	0.610	0.089
totChol	-0.066	0.263	-0.022	-0.048	-0.027	0.082	0.008	0.164	0.044	1.000	0.212	0.168	0.122	0.090	0.045	0.081
sysBP	-0.033	0.390	-0.127	-0.130	-0.086	0.259	0.055	0.694	0.103	0.212	1.000	0.785	0.330	0.183	0.128	0.215
diaBP	0.060	0.204	-0.063	-0.108	-0.051	0.196	0.049	0.614	0.052	0.168	0.785	1.000	0.382	0.178	0.060	0.148
BMI	0.081	0.135	-0.134	-0.170	-0.090	0.103	0.031	0.304	0.087	0.122	0.330	0.382	1.000	0.071	0.084	0.077
heartRate	-0.115	-0.012	-0.058	0.057	0.071	0.014	-0.012	0.147	0.054	0.090	0.183	0.178	0.071	1.000	0.094	0.026
glucose	0.003	0.114	-0.030	-0.052	-0.053	0.051	0.015	0.083	0.610	0.045	0.128	0.060	0.084	0.094	1.000	0.120
TenYearCHD	0.096	0.227	-0.053	0.025	0.063	0.086	0.044	0.174	0.089	0.081	0.215	0.148	0.077	0.026	0.120	1.000

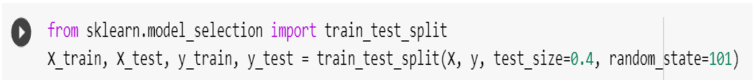
On the basis of the correlation matrix, we have chosen the input variables whose correlation with the target variable is positive.



```
X = df[['age', 'male','currentSmoker','cigsPerDay',
         'BPMeds','prevalentStroke','totChol','BMI',
         'heartRate', 'prevalentHyp', 'sysBP',
        'diaBP','glucose','diabetes']]

[31] y =  df[['TenYearCHD']]
```

5.3.1 Training and testing set

Data set was separated into training and testing sets for the evaluation process. This has been done using the scikit learn library. Training data is 60% and testing data is 40% and we can train our linear regression model on the training set and evaluate its performance on the testing set.

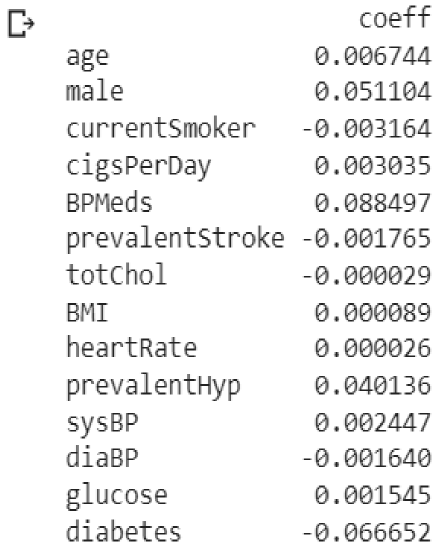


```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
```

5.3.2 Results

We would use regression analysis to estimate these coefficients based on the data we have collected. We would also assess the model's accuracy by calculating the root mean squared error (RMSE) or R-squared value.

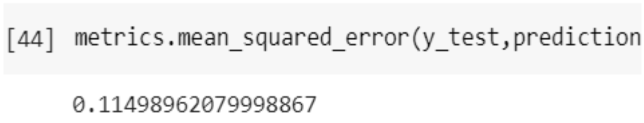
Input variables and their corresponding coefficients in the linear regression model.



```
                    coeff
age              0.006744
male             0.051104
currentSmoker   -0.003164
cigsPerDay       0.003035
BPMeds           0.088497
prevalentStroke -0.001765
totChol         -0.000029
BMI              0.000089
heartRate        0.000026
prevalentHyp     0.040136
sysBP            0.002447
diaBP           -0.001640
glucose          0.001545
diabetes        -0.066652
```

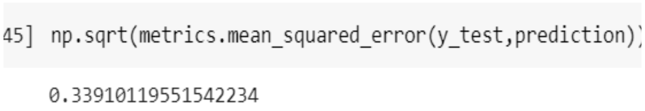
MSE measures the average squared difference between the actual and predicted values of the target variable.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$



```
[44] metrics.mean_squared_error(y_test,prediction

     0.11498962079998867
```

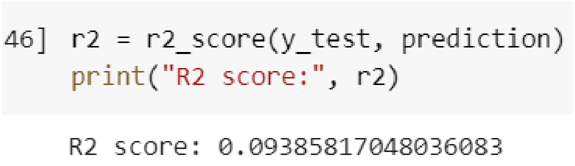
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$



```
45] np.sqrt(metrics.mean_squared_error(y_test,prediction)

    0.33910119551542234
```

R-squared indicates that the independent variables in the model explain only a small proportion i.e. 8.09% of the variance in the dependent variable. This means that the linear regression model is not a good fit for the data.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

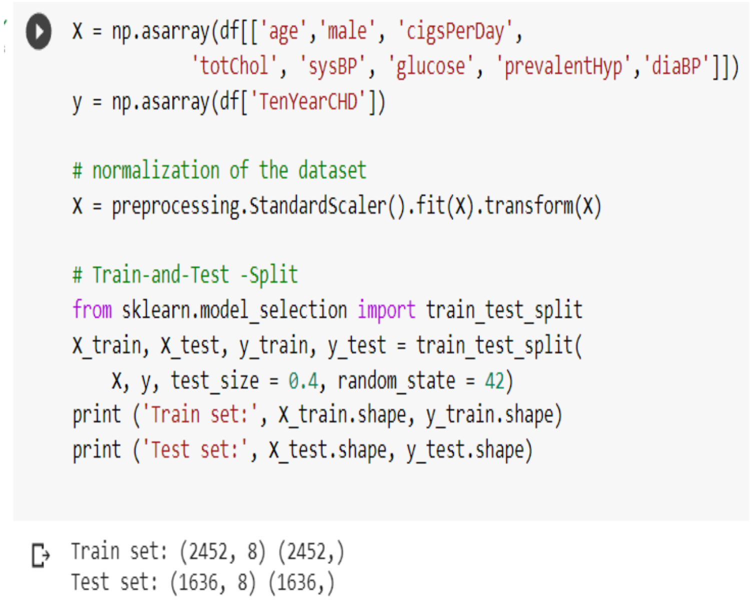


```
46] r2 = r2_score(y_test, prediction)
    print("R2 score:", r2)

    R2 score: 0.09385817048036083
```

This means that approximately 9.38% of the variance in the dependent variable can be explained by the independent variables in the regression model. In other words, the model is able to explain only a small portion of the variation in the dependent variable.

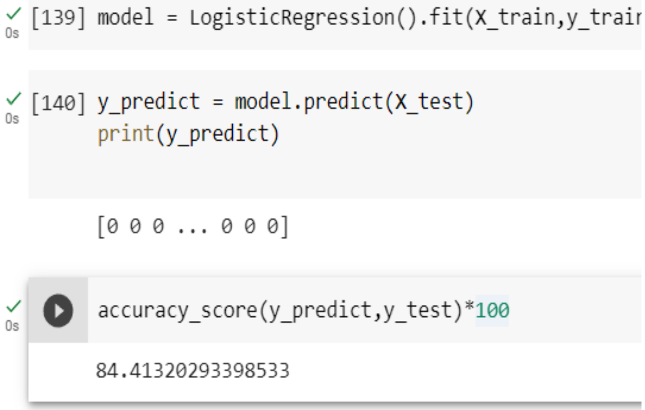
5.4 Implementing Logistics Regression

For pre-processing we are using Standardscaler. This is because some of the features in the dataset, such as age and blood pressure, have a wide range of values that are not bounded between 0 and 1. When using MinMaxScaler to scale such features, the range of values may be compressed and important information may be lost.



```
X = np.asarray(df[['age','male', 'cigsPerDay',
                   'totChol', 'sysBP', 'glucose', 'prevalentHyp','diaBP']])
y = np.asarray(df['TenYearCHD'])

# normalization of the dataset
X = preprocessing.StandardScaler().fit(X).transform(X)

# Train-and-Test -Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.4, random_state = 42)
print ('Train set:', X_train.shape, y_train.shape)
print ('Test set:', X_test.shape, y_test.shape)

Train set: (2452, 8) (2452,)
Test set: (1636, 8) (1636,)
```

Training data is 60% and testing data is 40% and we can train our linear regression model on the training set and evaluate its performance on the testing set.



```
[139] model = LogisticRegression().fit(X_train,y_trair

[140] y_predict = model.predict(X_test)
      print(y_predict)

      [0 0 0 ... 0 0 0]

accuracy_score(y_predict,y_test)*100

84.41320293398533
```

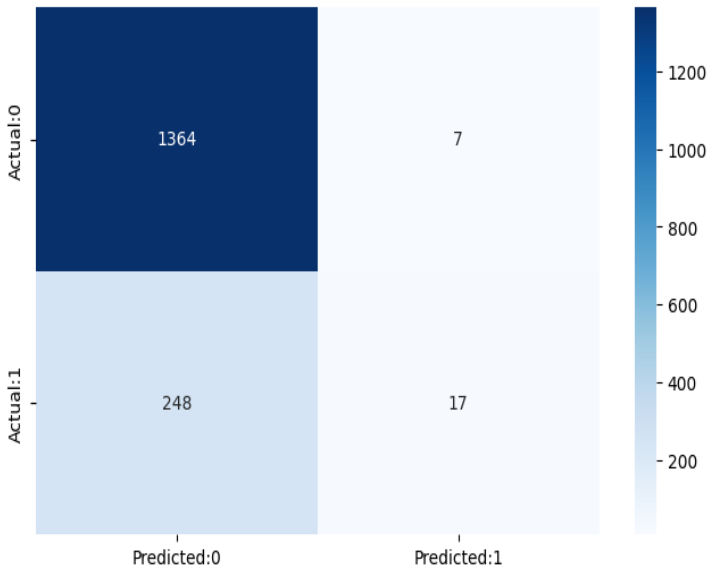
Training a logistic regression model using the input features X_train and corresponding target labels y_train, which will be used to make predictions on new data in the future. The accuracy of the model is 84.413.

5.5 Confusion Matrix Outcomes

This has been used to indicate the summary of prediction results including correct and incorrect on a classification problem.

Further, this was used to not only errors but also types of errors. The segments of the confusion matrix indicate the following parameters. True Positives (TP): cases which are predicted yes (they have the disease), and they do have the disease. True Negatives (TN): cases which are predicted no, and they do not have the disease.False Positives (FP): cases which are predicted yes, but they do not actually have the disease (Type I error). False Negatives (FN): cases which are predicted no, but they actually do have the disease (Type II error). The following outcome indicates the confusion matrix of the dataset.

According to the outcome of the confusion matrix,



Correct predictions (1364+ 17) =1381

Incorrect predictions (248 +7) =255

Therefore,

True Positives:17

True Negatives:1364

False Positives:7(Type I error)

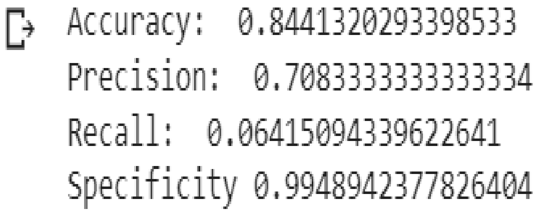
False Negatives:248(Type II error)

Terms	Formula
Accuracy of the model (overall, how often the classifier correct)	(TP + TN) / (TP + TN + FP + FN).
Sensitivity or True Positive Rate (when it is actually yes, how often does it predict yes)	TP / (TP + FP).
The precision of a machine learning model is dependent on both the negative and positive samples.	TP / (TP + FP).
Specificity or True Negative Rate (when it is actually no, how often does it predict no)	TN/(TN+FP)

Table 2: Terms and Formula

With analyzing confusion matrix data, it is evident that the model is more highly specific than sensitive.

Further, the negative values in the model are predicted more accurately than the positives.



```
Accuracy:  0.8441320293398533
Precision:  0.7083333333333334
Recall:  0.06415094339622641
Specificity 0.9948942377826404
```

6. Conclusion

Linear regression assumes a linear relationship between the dependent variable and the independent variables, which may not be appropriate for binary outcomes. Logistic regression, on the other hand, uses a logistic function to model the probability of a binary outcome based on the independent variables. The aim of this study is to evaluate the risk of 10-year CHD using 14 input variables. The attributes are selected after the backward elimination process considering the P values which are lower than 5%. Therefore, the logistic regression model is derived through P values of the variables <0.05. According to the logistic regression outcome, men are more susceptible to heart disease than women. Age, number of cigarettes per day and systolic blood pressure are the odds of CHD. However, there is no significant change in the total cholesterol level and the glucose level. But, the level of glucose has a negligible change in odds. The model is more specific than sensitive. Further, the accuracy of the model is

0.84413.

References

[1]. Mozaffarian, D., Benjamin, E., Go, A., Arnett, D., Blaha, M.Cushman, M. et al. (2015). Heart Disease and Stroke Statistics—2015, Update. Circulation, 131(4). doi:10.1161/cir.0000000000000152.

[2]. Das, S., Dey, A., Pal, A., & Roy, N. (2015).Applications of Artificial Intelligence in Machine Learning: Review and Prospect. International Journal of Computer Applications,115(9),31-41.

doi:10.5120/20182-2402

[3]. Abduljabbar, R., Dia, H., Liyanage, S., & Bagloee, S. (2019). Applications of Artificial Intelligence in Transport: An Overview. Sustainability, 11(1), 189.

doi: 10.3390/su11010189

[4]. Strecht, Pedro & Cruz, Luís & Soares, Carlos & Moreira, João & Abreu, Rui. (2015). A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance