PROJECT REPORT ON

“HEART DISEASE PREDICTION MODEL”

Data Science and Data Analytics

Batch -B5

Submitted by-

Akshat Srivastava

Devesh Kumar Singh

Shreyas Bind

Satyam Kumar Maurya

Submitted to

Mrs. Purbadri Ghoshal Ma’am

TABLE OF CONTENT

ABSTRACT

This report represents the project assigned to BATCH B5 students for the partial fulfillment of COMP 484, Machine Learning and Data Science and Analytics, given by the Quantum Learnings platform.
Cardiovascular diseases are the most common cause of death worldwide over the last few decades in the developed as well as underdeveloped and developing countries. Early detection of cardiac diseases and continuous supervision of clinicians can reduce the mortality rate. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. In this project, we have developed and researched about models for heart disease prediction through the various heart attributes of patient and detect impending heart disease using Machine learning techniques like, KNN ,SVM and Logistic Regression on the dataset available in mentorrbuddy Website, further evaluating the results using confusion matrix and cross validation. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. Keywords: Machine Learning, KNN,Logistic regression, CrossValidation , Cardiovascular Diseases, Support Vector Machine, Logistic Regression

INTRODUCTION

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications. This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithms.

Problem Definition

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

Motivation

DS&DA Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore behind the models, and further implement Logistic Regression model to train the obtained data. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of heart disease, in this project we have developed a model which classifies if patient will have heart disease in ten years or not based on various features (i.e. potential risk factors that can cause heart disease) using logistic regression. Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

Objective: The main objective of developing this project are: 1. To develop machine learning model to predict future possibility of heart disease by implementing best model. 2. To determine significant risk factors based on medical dataset which may lead to heart disease. 3. To analyze feature selection methods and understand their working principle.

DATASET

The dataset is available on mentorrbuddy.com which is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. It provides patient information which includes over 300 and 14 attributes. The attributes include: age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting, sugar blood, resting electrocardiographic results, maximum heart rate, exercise induced angina, ST depression induced by exercise, slope of the peak exercise, number of major vessels, and target ranging from 0 to 2, where 0 is absence of heart disease. The data set is in csv (Comma Separated Value) format which is further prepared to data frame as supported by pandas library in python.

IMPORTING LIBRARIES AND READING FROM CSV

```
# importing necessary modules

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,recall_score,precision_score
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
```

```
# read the data from dataset (.csv)
df = pd.read_csv(r'D:\heart_disease.csv')
```

DATA PREPARATIONS

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

CHECKING FOR NULL VALUES

```
#checking for null values
df.isnull().sum()
```

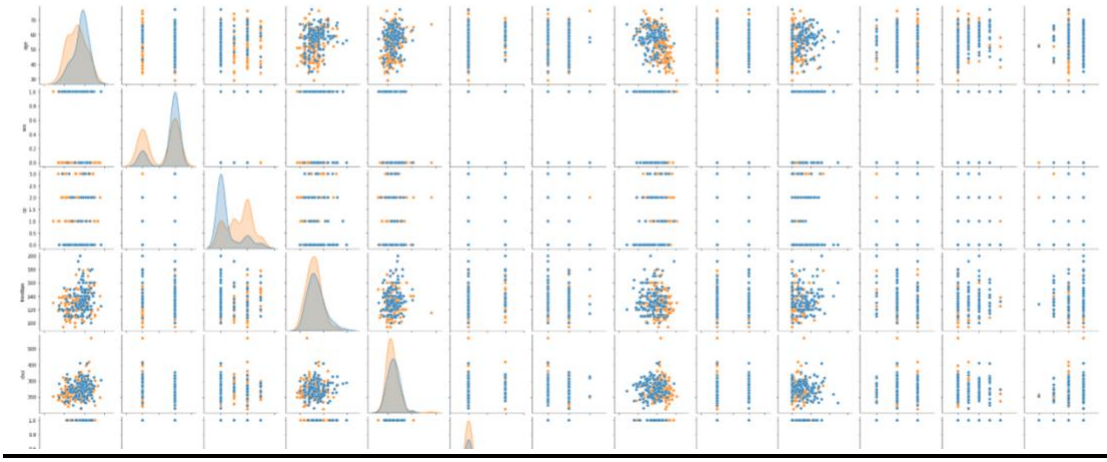
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

From this we can say that dataset has no null values.

CHECKING DISTRIBUTION OF TARGET VARIABLESWITH ATTRIBUTES

```
# checking distributiion of target with other attributes
sns.pairplot(df, hue = 'target')
```

```
<seaborn.axisgrid.PairGrid at 0x19689e4beb0>
```



By using the pairplot feature of matplotlib it can be seen thatBy distributing the attributes between the targets as positive or negative we can see that the data is not linearly separable at any attribute hence the logistic regression can not be directly used .

EXPLORATORY ANALYSIS

Heatmap using Seaborn Library is plotted for checking correlation

```
# checking correlations with other variable
plt.figure(figsize = (15,15))
sns.heatmap(df.corr() , annot = True)
plt.show()
```

FEATURE SELECTION

```
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

as the features supplied are all related for predicting the target and attributes and independent so the categorical data is independently selected. So the model was prepared without dropping any column

SEPERATING OUR TARGET VARIABLE WITH OTHER ATTRIBUTES

```
# seperating our target variable
x = df.iloc[ : , :-1]
y = df.iloc[ : , -1]
```

```
x
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2

303 rows × 13 columns

```
y
```

```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64
```

FEATURE SCALING

Since the feature with a higher value range starts dominating when calculating we need to scale the data.

```
# feature scaling
sc = StandardScaler()
x = sc.fit_transform(x)
```

SPLIT THE DATA INTO TRAINING AND TESTING SET

```
# split the data into training and testing
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.2, random_state = 43)
```

MODEL SELECTION

METHODS AND ALGORITHMS USED:

KNN(K Nearest Neighbours):

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. o K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

Logistic Regression:

This type of statistical analysis (also known as *logit model*) is often used for predictive analytics and modeling, and extends to applications in machine learning. In this analytics approach, the dependent variable is finite or categorical: either A or B (binary regression) or a range of finite options A, B, C or D (multinomial regression). It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.

Support Vector Machine:

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the

hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

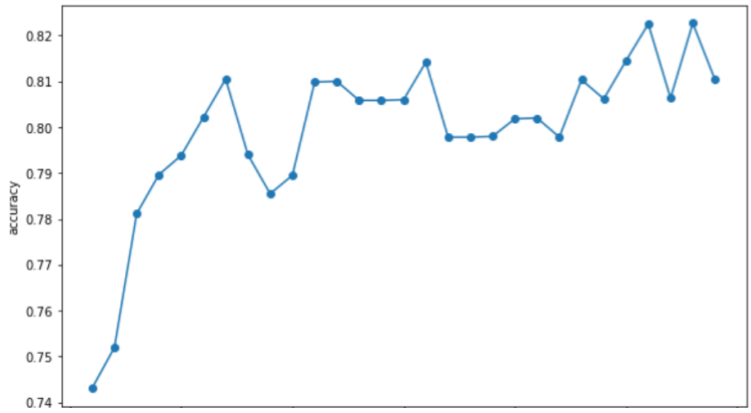
KNN Algorithm

```
# choosing the value of n in KNN
accuracy = []
for i in range(1,30):
    knn = KNeighborsClassifier(n_neighbors = i)
    score = cross_val_score(knn,x_train,y_train,cv=10)
    accuracy.append(score.mean())
```

```
plt.figure(figsize = (10,6))
plt.plot(range(1,30),accuracy,marker = 'o')
plt.xlabel('k')
plt.ylabel('accuracy')

# from fig weget n = 14
```

```
Text(0, 0.5, 'accuracy')
```




```
## model selection
```

```
# KNN Classification
knn = KNeighborsClassifier(n_neighbors = 14)
knn.fit(x_train,y_train)
y_pred = knn.predict(x_test)
```

```
#accuracy,recall and precision in KNN
y_pred = knn.predict(x_test)
acc = accuracy_score(y_test,y_pred)
cm = confusion_matrix(y_test,y_pred)
print(acc)
print(recall_score(y_test,y_pred))
print(precision_score(y_test,y_pred))
print(cm)
```

```
0.9180327868852459
1.0
0.868421052631579
[[23  5]
 [ 0 33]]
```

Support Vector Machine

```
# support vector machine(rbf)
svc = SVC() #default hyperparameters rbf
svc.fit(x_train,y_train)
y_pred_svc = svc.predict(x_test)
from sklearn import metrics
print('accuracy score:',accuracy_score(y_test,y_pred_svc))
```

```
accuracy score: 0.8688524590163934
```

```
# support vector machine(linear)
svc1 = SVC( kernel = 'linear')
svc1.fit(x_train,y_train)
y_pred_li = svc1.predict(x_test)
print('accuracy score:',accuracy_score(y_test,y_pred_li))
```

```
accuracy score: 0.8852459016393442
```

```
# support vector machine(polynomial)
svc2 = SVC(kernel = 'poly')
svc2.fit(x_train,y_train)
y_pred2 = svc2.predict(x_test)
print('accuracy score:',metrics.accuracy_score(y_test,y_pred2))
```

```
accuracy score: 0.8360655737704918
```

Logistic Regression

```
# logistic regression
lr = LogisticRegression()
lr.fit(x_train,y_train)
y_pred_lr = lr.predict(x_test)
print('accuracy score:',metrics.accuracy_score(y_test,y_pred_lr))
```

```
accuracy score: 0.8852459016393442
```

EVALUATION METRICS

For the evaluation of our output from our training the data, the accuracy was analyzed “Confusion matrix”

Confusion Matrix

A confusion matrix, also known as an error matrix, is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. The key to the confusion matrix is the number of correct and incorrect predictions are summarized with count values and broken down by each class not just the number of errors made.

Accuracy

The accuracy is calculated as: Accuracy = $TP+TN/\ TP+TN+FP+FN$

where,

• True Positive (TP) =Observation is positive, and is predicted to be positive.

• False Negative (FN) = Observation is positive, but is predicted negative

• True Negative (TN) = Observation is negative, and is predicted to be negative.

• False Positive (FP) =Observation is negative, but is predicted positive

The obtained accuracy during training the data after feature selection using KNN and during testing was 91.8%

Recall

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN). Recall is calculated as: Recall = $TP\ /TP+FN$ The obtained recall during training the data after feature selection using KNN was and during testing was 1.0

Precision

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled as positive is indeed positive (a small number of FP). Precision is calculated as: Precision = $TP\ /TP+FP$ The obtained precision during training the data after feature selection using KNN was 0.86

DISCUSSION ON RESULTS

When performing various methods of feature selection, testing it was found that backward elimination gave us the best results among others. The various methods tried were KNN with and without KFold, Recursive Feature Elimination with Cross Validation. The accuracy that was seen in them ranged around 90% and 91.8% being maximum. The precision of KNN is 86%. And the recalls is 1.

MODEL USED	ACCURACY (IN %)
1. KNN	91.8
2. LOGISTIC REGRESSION	88.5
3. SVM(RBF)	86.8
4. SVM(LINEAR)	88.5
5. SVM(POLYNOMIAL)	83.6

Hence from accuracy score we get KNN is best model for applied dataset

CONCLUSION

The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. This project resolved the feature selection i.e. KNN behind the models and successfully predict the heart disease, with 91% accuracy. The model used was Logistic Regression. Further for its enhancement, we can train on models and predict the types of cardiovascular diseases providing recommendations to the users, and also use more enhanced models.

REFRENCES

1. Stackoverflow.com

2. Academia.edu

3. Githubs.org

4. Sklearn.org

5. Seaborn.org