Heart Disease Prediction

Team Mates:
2103A52003
2103A52014
2103A52166
2103A52146

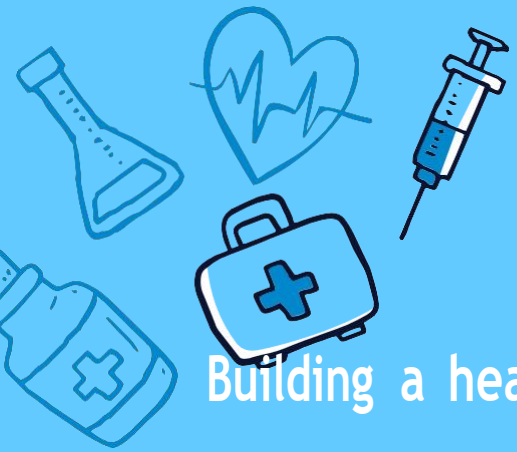
Abstract:

Heart disease is one of the most significant causes of mortality in the world today. Prediction of cardiovascular disease is a critical challenge in the area of clinical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry. We have also seen ML techniques being used in recent developments in different areas of the Internet of Things (IoT). Various studies give only a glimpse into predicting heart disease with ML techniques.

Introduction:

It is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate and many other factors. Various techniques in data mining and neural networks have been employed to find out the severity of heart disease among humans. The severity of the disease is classified based on various methods like K -Nearest Neighbor Algorithm (KNN), Support vendor machine(SVM), random forest . The nature of heart disease is complex and hence, the disease must be handled carefully.

The appropriate algorithms used for heart disease prediction are Support vendor machine(SVM), K-nearest neighbors(KNN) and Random forest.



Building a heart disease prediction project using AI/ML involves
-collecting and pre-processing data
-selecting relevant features
-choosing the appropriate machine learning algorithm
-training and evaluating the model and
-optimizing the model parameters..

Problem Statement:-

The objective of this study is to develop a machine learning model that can accurately predict the presence of heart disease in patients based on their medical history and clinical parameters. Heart disease is a major cause of morbidity and mortality worldwide, and early detection and treatment can significantly improve patient outcomes. Therefore, accurate prediction of heart disease risk can assist healthcare providers in making informed decisions and providing targeted interventions to at-risk patients. The model will be trained on a dataset of patients' medical records and clinical data, and will be evaluated on its ability to accurately classify patients as either having or not having heart disease. The results of this study can provide valuable insights for healthcare providers and researchers in the prevention and management of heart disease.

METHODOLOGY:-

Support Vector Machine:-

A support vendor machine (SVM) is a type of machine learning algorithm used for classification and regression analysis. It is commonly used for solving binary classification problems where the goal is to classify data into one of two categories. A vending machine is a machine that dispenses items such as snacks, drinks, and other goods automatically after the user inserts the correct amount of money. Vending machines are commonly used in public places such as schools, airports, and hospitals.

K-nearest neighbors:-

KNN stands for K-Nearest Neighbors and is a machine learning algorithm used for both classification and regression tasks. The algorithm determines the classification of a new data point by comparing it to the k nearest neighbors in the training set and selecting the most common classification or average value among the neighbors.

Random forest:-

Random Forest is a machine learning algorithm that is used for classification, regression, and other tasks. It works by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees. The random forest algorithm helps to reduce overfitting by using bootstrap samples of the training data and randomly selecting a subset of the features for each tree.

Training and testing of data:-

Conversion of data to training and testing

```
[64] x = d.iloc[:, :-2]
     y = d.iloc[:, -1]
     x_train, x_test, y_train, y_test = train_test_split(x, y, random_state = 0, test_size = 0.35)
```

Standard Scaler for Standardization technique

```
[65] sc_x = StandardScaler()
     x_train = sc_x.fit_transform(x_train)
     x_test = sc_x.transform(x_test)
```

Nearest Neighbours

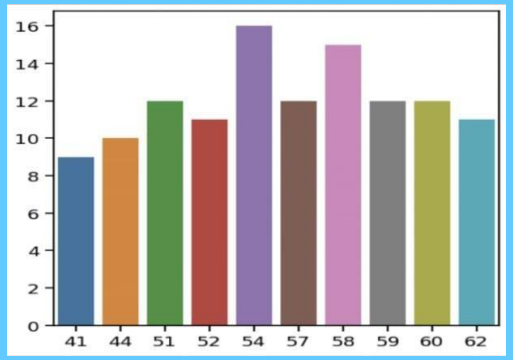
```
[66] import math
     math.sqrt(len(y_test))
```

```
9.746794344808963
```

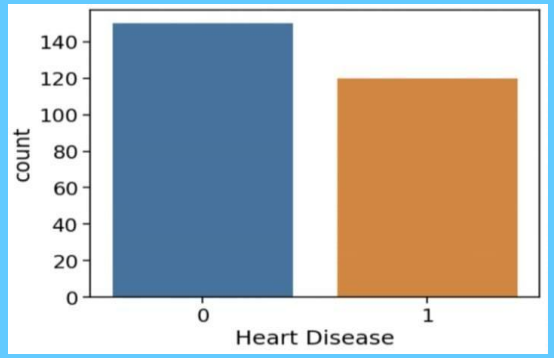
Dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Age	Sex	Chest pair	BP	Cholesterc	FBS over 1	EKG result	Max HR	Exercise a	ST depress	Slope of S	Number o	Thallium	Heart Disease	
2	70	1	4	130	322	0	2	109	0	2.4	2	3	3	1	
3	67	0	3	115	564	0	2	160	0	1.6	2	0	7	0	
4	57	1	2	124	261	0	0	141	0	0.3	1	0	7	1	
5	64	1	4	128	263	0	0	105	1	0.2	2	1	7	0	
6	74	0	2	120	269	0	2	121	1	0.2	1	1	3	0	
7	65	1	4	120	177	0	0	140	0	0.4	1	0	7	0	
8	56	1	3	130	256	1	2	142	1	0.6	2	1	6	1	
9	59	1	4	110	239	0	2	142	1	1.2	2	1	7	1	
10	60	1	4	140	293	0	2	170	0	1.2	2	2	7	1	
11	63	0	4	150	407	0	2	154	0	4	2	3	7	1	
12	59	1	4	135	234	0	0	161	0	0.5	2	0	7	0	
13	53	1	4	142	226	0	2	111	1	0	1	0	7	0	
14	44	1	3	140	235	0	2	180	0	0	1	0	3	0	
15	61	1	1	134	234	0	0	145	0	2.6	2	2	3	1	
16	57	0	4	128	303	0	2	159	0	0	1	1	3	0	
17	71	0	4	112	149	0	0	125	0	1.6	2	0	3	0	
18	46	1	4	140	311	0	0	120	1	1.8	2	2	7	1	
19	53	1	4	140	203	1	2	155	1	3.1	3	0	7	1	
20	64	1	1	110	211	0	2	144	1	1.8	2	0	3	0	
21	40	1	1	140	199	0	0	178	1	1.4	1	0	7	0	
22	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1	
23	48	1	2	130	245	0	2	180	0	0.2	2	0	3	0	
24	43	1	4	115	303	0	0	181	0	1.2	2	0	3	0	
25	47	1	4	112	204	0	0	143	0	0.1	1	0	3	0	
26	54	0	2	132	288	1	2	159	1	0	1	1	3	0	
27	48	0	3	130	275	0	0	139	0	0.2	1	0	3	0	
28	46	0	4	138	243	0	2	152	1	0	2	0	3	0	
29	51	0	3	120	295	0	2	157	0	0.6	1	0	3	0	
30	58	1	3	112	230	0	2	165	0	2.5	2	1	7	1	

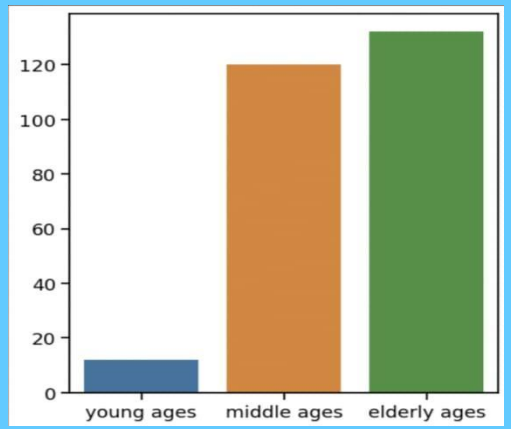
Graphs:-



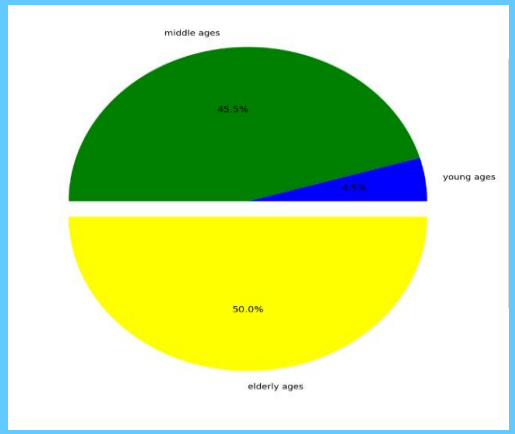
No of age factor



Yes or no of heart disease



No of young age, midddle age and elder age



Pie chart of young age, midddle age and elder age

Accuracy achieved through svm:

SVM MODEL

```
[71] from sklearn import svm
     clf = svm.SVC(kernel='rbf')
     clf.fit(x_train,y_train)
     y_pred = clf.predict(x_test)
```

```
[72] y_pred = clf.predict(x_test)
     y_pred
```

```
array([0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 0, 1, 0, 0, 1, 0])
```

```
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[44 11]
 [11 29]]
```

```
[78] X=accuracy_score(y_test,y_pred)
     print(X*100)
```

```
76.84210526315789
```

Accuracy = 76.84210526315789

Accuracy achieved through KNN:

KNN MODEL

```
[67] classifier = KNeighborsClassifier(n_neighbors = 9, p = 2, metric = 'euclidean')
     classifier.fit(x_train,y_train)
```

```
KNeighborsClassifier(metric='euclidean', n_neighbors=9)
```

```
[68] y_pred = classifier.predict(x_test)
     y_pred
```

```
array([0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0,
       1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 0, 1, 0])
```

```
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[43 12]
 [11 29]]
```

```
[79] Z=accuracy_score(y_test,y_pred)
     print(Z*100)
```

```
76.84210526315789
```

Accuracy = 76.84210526315789

Accuracy achieved through Random forest:

RANDOM FOREST MODEL

```
[75]  rf = RandomForestClassifier(n_estimators=500, random_state=12, max_depth=5)
      rf.fit(x_train,y_train)
      rf_predicted = rf.predict(x_test)
```

2s

```
[76]  rf_conf_matrix = confusion_matrix(y_test, rf_predicted)
      print("confusion matrix")
      print(rf_conf_matrix)
```

```
confusion matrix
[[45 10]
 [ 7 33]]
```

0s

```
[77]  rf_acc_score = accuracy_score(y_test, rf_predicted)
      print("Accuracy of Random Forest:",rf_acc_score*100)
```

```
Accuracy of Random Forest: 82.10526315789474
```

0s

Accuracy = 82.10526315789474

Thanks

Do you have any questions?

youíemail@fíeepik.com

+91 620 421 838

youícompany.com

CREDIT'S: This presentation template

was cíeal'ed by **Slidesgo**, including icons
by **ïlaticon** and infogíaphics & images
by **ïíeepifi**

Please keep l'his slide foí al'l'íibul'ion