CLASSIFYING HEART DISEASE USING MACHINE LEARNING

A Project Report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

CH SHIVA, 1215316314

P AKHIL, 1215316316

M CHAKRADHAR, 1215316339

T ATREYA PAVAN KUMAR, 1215316357

Under the esteemed guidance of

VINOD BABU P

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM

(Deemed to be University)

VISAKHAPATNAM

APRIL 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled "**CLASSIFYING HEART DISEASE USING MACHINE LEARNING**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

Registration Numbers	**Names**	**Signatures**
1215316314	**CH SHIVA**	
1215316316	**P AKHIL**	
1215316339	**M CHAKRADHAR**	
1215316357	**T ATREYA PAVAN KUMAR**	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project entitled "**CLASSIFYING HEART DISEASE USING MACHINE LEARNING**" is a bona-fide record of work carried out by **CH SHIVA (1215316314), P AKHIL (1215316316), M CHAKRADHAR (1215316339), T ATREYA PAVAN KUMAR (1215316357)** students submitted in partial fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide	**Head of the Department**
Vinod Babu P	**Dr. K. Thammi Reddy**
Assistant Professor	**Professor**

ACKNOWLEDGEMENT

The project opportunity we had was a great chance for learning and professional development. Therefore, we consider ourselves very lucky individuals as we were provided with an opportunity to be a part of it.

Bearing in mind previous we are using this opportunity to express my deepest gratitude and special thanks to Vinod Babu P sir who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep us on the correct path.

We express our deepest thanks to Vinod Babu P for taking part in useful decisions & giving necessary advice and guidance, as mentor. We choose this moment to acknowledge their contribution gratefully.

We perceive this opportunity as a big milestone in our career development. We will strive to use gained skills and knowledge in the best possible way, and we will continue to work hard improve, in order to attain desired career objectives. Hope to continue cooperation with you in the future.

Sincerely,

CH SHIVA, 1215316314

P AKHIL, 1215316316

M CHAKRADHAR, 1215316339

T ATREYA PAVAN KUMAR, 1215316357

TABLE OF CONTENTS

1. ABSTRACT:

In this age of information science handling data became intelligent and straightforward. For that purpose, there are various tools, techniques, and methods that are proposed. We applied various data science algorithms like KNN(k-nearest neighbor) is usually used for classification and multivariate analysis and within the same way SVM(Support Vector Machine) could be a classification algorithm wont to classify data using training examples to classify data and find a prediction for the target.

2. INTRODUCTION:

Machine Learning is employed round the world in various organizations in various fields. The healthcare industry cannot be negligible. Machine Learning and AI are playing a really crucial role in predicting existence or absence of disorders, lung diseases and more. We are predicting heart diseases well ahead using information that's data given by hospital accustomed provide a good value of data to the doctors in order that they can prescribe the patient with better treatment. The algorithms accustomed predict heart diseases are Decision Tree Classifier, K Neighbours Classifier, Random Forest Classifier, Principal Component Analysis and Support Vector Classifier. We applied many operations on data so as to form data efficient and consistent without outliers, missing values, etc. So, while applying algorithms they will perform more efficiently. We used Train Test Split and also K-Fold Cross Validation to coach and test the efficiency of algorithm when the information is applied. The accuracy of every algorithm is given within the comparison table.

3. EXISTING SYSTEM:

- All patients' details are stored by the hospitals in their respective hospital’s databases or physically in paper format but they are not using it for any purpose.
- Drawbacks:

  1. Difficult to research for the doctor to predict whether a patient has cardiovascular disease or not.

  2. When the first symptoms of cardiovascular disease are ignored, the patient might find yourself with drastic consequences during a short span of your time.

4. PROBLEM STATEMENT:

Heart diseases are the foremost reason behind deaths round the globe that's more humans die yearly from heart diseases than from the opposite cause. An estimated 18 million humans died from heart attacks in 2016, representing thirty one percent of all global deaths of this eight five percent humans are died because of heart attacks – WHO. Most Heart diseases are often prevented by taking some risk factors into consideration like age, diet, obesity, physical inactivity, etc. These all patients' details are stored by the hospitals in their respective hospital’s databases or physically in paper format but they are not using it for any purpose.

5. SYSTEM METHODOLOGY (Proposed System):

Heart disease is that the main reason behind demises of many people among all other diseases. Because of the dearth of resources within the health industry the prediction of cardiovascular diseases occasionally could also be an issue. This problem will be taken care by adopting Machine learning techniques.

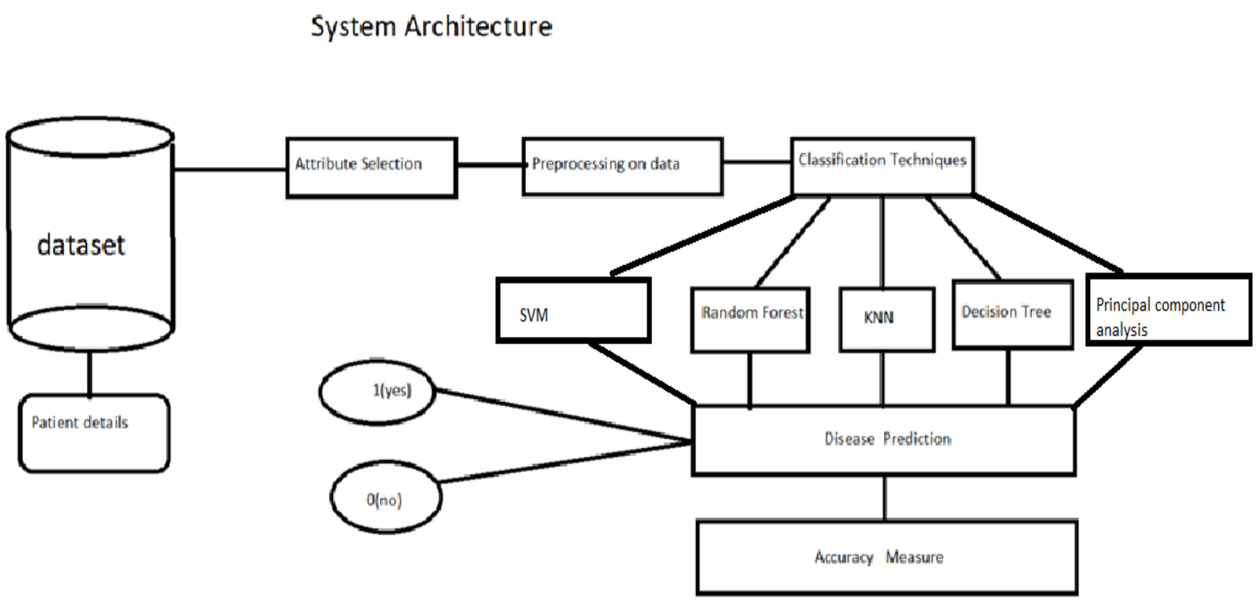


Figure 1: System Architecture

- The working of the system architecture is described step by step:

1. Dataset contains all the heart patients' details.

2. In attribute selection process we select the required attributes used to analyse the data which are important and we drop unnecessary attributes which are not required.

3. Then we pre-process data such as filling missing values, etc.

4. Then we applied some data science algorithms to analyse and predict the condition of the heart patient which is vulnerable or not.

5. Finally, we measured accuracy of each algorithm we applied to the data.

5.1. LIBRARIES:

We imported several libraries in python for the project such as:

- NumPy: to figure with arrays.
- Pandas: to figure with CSV(Comma Separated Values) files and data frames.
- matplotlib: to form charts using pyplot.
- Sklearn: It provides algorithms both supervised and unsupervised.
- Seaborn: For analytics and data visualization of Dataset.
- SimpleImputer: Used to handle missing values.
- IteretiveImputer: Used to handle missing values.

5.2. DATASET DESCRIPTION:

Dataset contains following features:

S.No	Attribute Name	Attribute Description
1.	Age	Patients age
2.	Sex	Gender of the Patient
3.	CP	chest pain kind that it's far whether the affected person is dealing with normal angina-0,strange angina-1,non-anginal pain-2 and sooner or later asymptotic-3
4.	trestbps	Patients resting blood pressure in mm/Hg
5.	Chol	Cholesterol level of patient in mg/dl
5.	Fbs	If fbs(fasting blood sugar)=1(if>a hundred and twenty mg/dl), (else 0)
7.	Restecg	0-normal, 1-having ST-T, 2-hypertrophy(resting electrocardiographic results)
8.	thalach	Patients max coronary heart rate achieved
9.	exang	Whether the patient does exercise(1=yes and 0=no)
10.	Oldpeak	ST despair prompted by means of exercising relative to rest
11.	Slope	1-upsloping, 2-flat, 3-downsloping

12.	Ca	0-three vessels, range of most important vessels colored by using flourosopy
13.	Thal	3-normal, 6-fixed defect, 7-reversable illness thalassemia
14.	Target	0= no coronary heart disease 1=heart disease

5.3. ATTRIBUTE SELECTION:

In this step we select the required columns which are useful to our project in the dataset and remove unnecessary columns from the dataset so that our storage space is saved and optimized.

We remove unnecessary columns to increase the accuracy of the algorithm and optimise storage space.

By removing all the unnecessary columns we can better analyse our dataset because unnecessary columns reduces the performance of algorithm.

For Example, Attributes such as persons name, address, phone numbers, etc, are useless to data analysis because we cannot get any patterns from that kind of data.

Here in this project we are using pandas module to handle the data, in python. We can remove the columns by using pandas function 'DROP'.

'Drop' is a python function used to remove both rows and columns.

Example code of drop method can be seen below is written in python programming language:

#importing pandas module

#importing dataset

#removing one column

Dataset = pandas.drop('column name',axis=1,inplace=1)

#**or when you want to remove multiple attributes**

Dataset = pandas.drop(['column name1', 'column name2'],axis=1,inplace=1)

Where,

- Dataset = the dataset we are using to analyse dataset
- Drop() = the method used to remove columns here
- Column names is the column/columns you want to remove
- Axis=1 is the term used to column where axis=0 specifies row.

- If inplace=1 it makes changes permanently in the dataset where as inplace=0 makes temporary changes to dataset which is temporary.

5.4. DATA PRE-PROCESSING:

- ➢ Data in the real world is tedious.
  - Such as incomplete : missing attribute values
  - For example: Job=’’ →missing data in occupation
- ➢ Noisy : Containing errors or outliers
  - For Example: Billing Amount=’-1’ → When we are having such values it results in Inaccurate results.
- ➢ Inconsistent : Containing discrepancies in data.
  - For Example: Age=’39’ Birthday='03/08/1998’ →here birthdate does not match age.
- ➢ Incomplete : contains missing values

Handling Missing Values example in python:

#pandas is used to handle dataset

# Imputer class is also used to handle missing values here

#importing dataset

#1. OMISSION:

- ➢ Removal Of rows → .dropna(axis=0)

#2. IMPUTATION

- ➢ Fill with Zero → .SimpleImputer(strategy=’constant’, fill_value=0)
- ➢ Imputer Mean → . SimpleImputer(strategy=’mean’)

- Imputer Median → .SimpleImputer(strategy='median')
- Imputer Mode→ .SimpleImputer(strategy='mode')
- Iterative imputer → .IterativeImputer()

5.5. UNDERSTANDING THE DATA - HISTOGRAM:

A histogram is a precise visualization of the spread of numerical data. It's approximated that the probability distribution of endless variables and was early given by Karl Pearson. It differs from a chart, within the sense that a chart relates two variables, but a histogram relates just one.

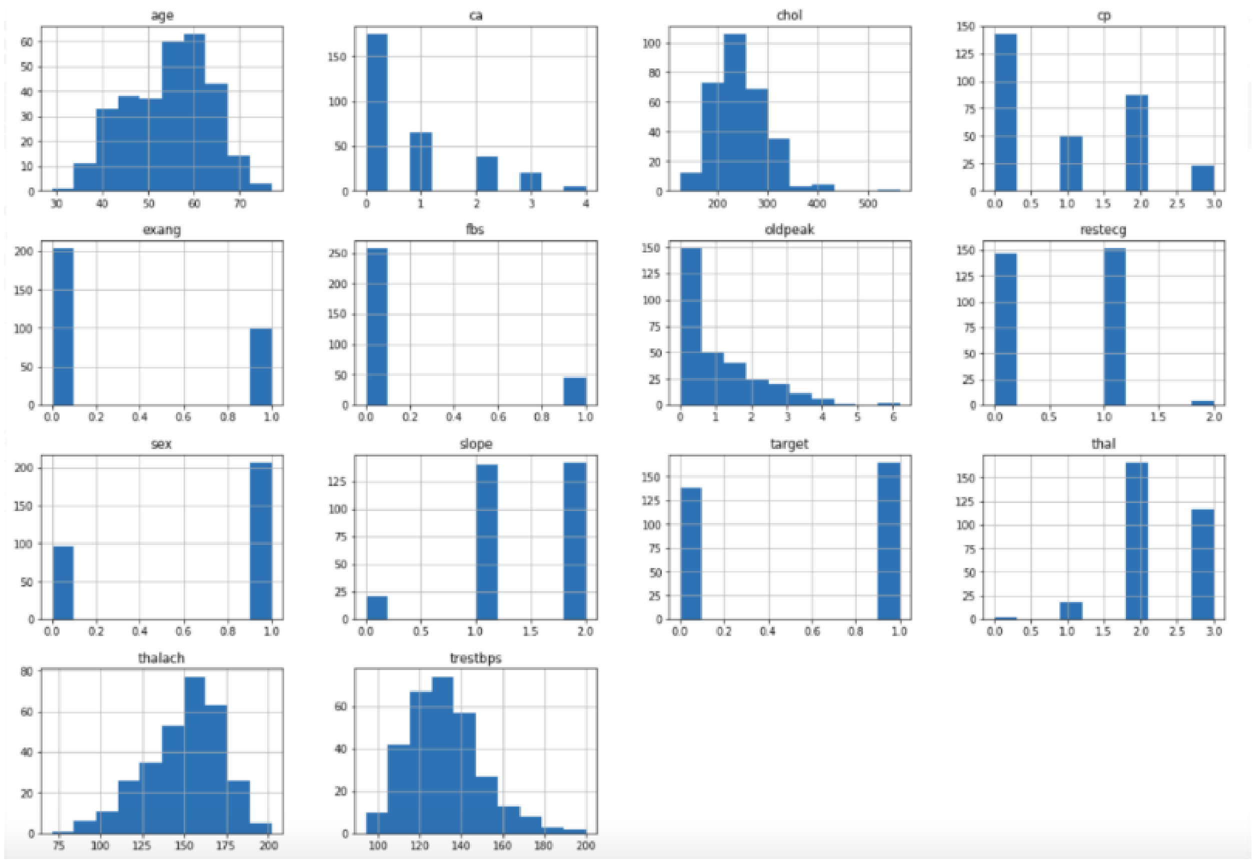


Figure 2: Histogram for our dataset

5.6. K-FOLD CROSS VALIDATION:

The general procedure is as follows:

1. Shamble the set containing the data randomly.

2. Break the set containing the data into k categories.

For each group:

1. Take one set as testing set.
2. Take remaining sets as train sets.
3. Calculate the accuracy of each group.
4. Finally, calculate the average and obtain the final accuracy.

10 Fold Cross Validation

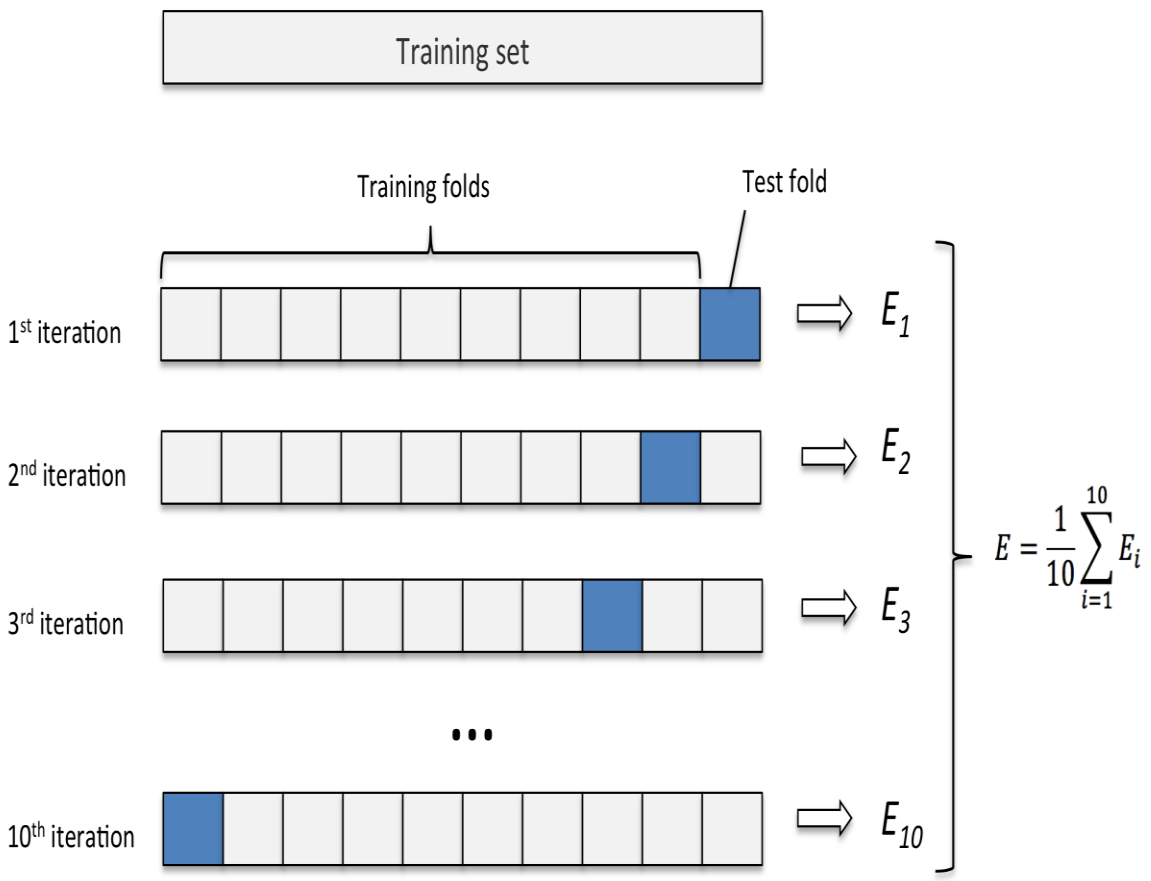


Figure 3: 10 fold cross validation

5.7. Decision Tree Classification:

This algorithm is generally used as classification algorithm. It is a Supervised Machine Learning where the data is continuously split per a specific parameter.

5.7.1 DECISION TREE CONTAINS:

1. Root Nodes

2. Internal Nodes

3. Leaf Nodes or Leaves

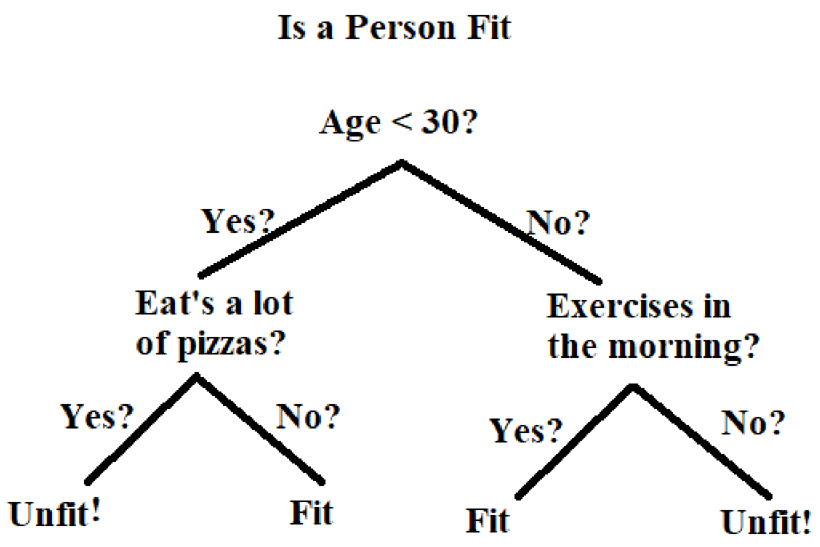


Figure 4: Decision tree Example

To comprehend the idea of Decision Tree thinks about the above model. The choice hubs are questions like 'What's the age?', 'Does he exercise?', 'Does he eats a lot of pizzas'? What's more, the leaves speak to results like either 'fit', or 'unfit'.

Two measures are accustomed decide the most effective attribute:

1. Entropy
2. Information Gain

1. Entropy:

Entropy calculates the degradation or uncertainty there in the information. It is wont to decide how a choice Tree can split the info.

2. Information Gain:

Information Gain (IG) is that the most essential measure wont to construct a desire Tree.
It indicates what proportion of "information" a particular feature/ variable gives us about
the remaining outcome.

Information Gain is critical due to the fact it wont to select the variable that first-class splits
the data at each node of a desire Tree. The variable with the very pleasant IG is employed to
separate the information at the basis node.

5.8. K NEIGHBORS CLASSIFIER:

5.8.1. K-NN ALGORITHM:

K-NN(K-Nearest Neighbors) is one simplest algorithm in which every foremost notable classification algorithms are currently among the trade because of its simplicity and accuracy.

5.8.2. K-NN REPRESENTATION:

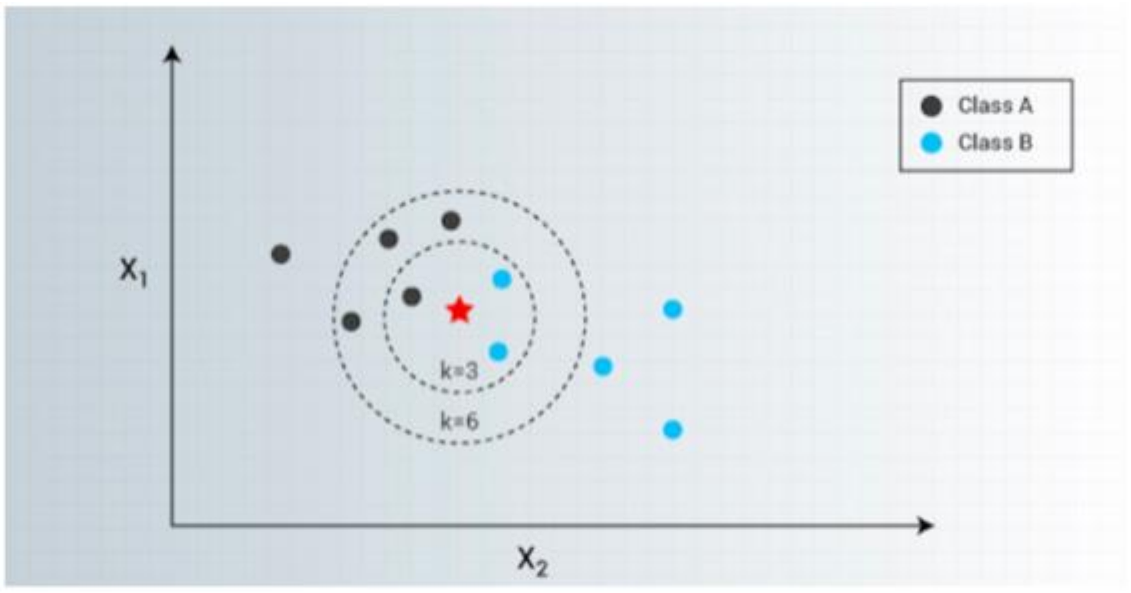


Figure 5: KNN Representation

K-NN might likewise be a straightforward calculation that stores each single accessible case and arranges new cases upheld a closeness live (e.g., separation capacities). KNN has been utilized in measurable estimation and example acknowledgment as of currently toward the beginning of the Seventies as a non-parametric procedure. The calculation expects that comparable things exist in distance. As it were, substances that area unit comparable exist along.

5.8.3. HOW DOES THE K-NN ALGORITHM WORK?

K is usually an odd number if the amount of classes is two. When K=one, then the algorithm chooses one nearest neighbor. But when k=one it can be a disadvantage due to outliers but when k is 3 or more is often the best case.

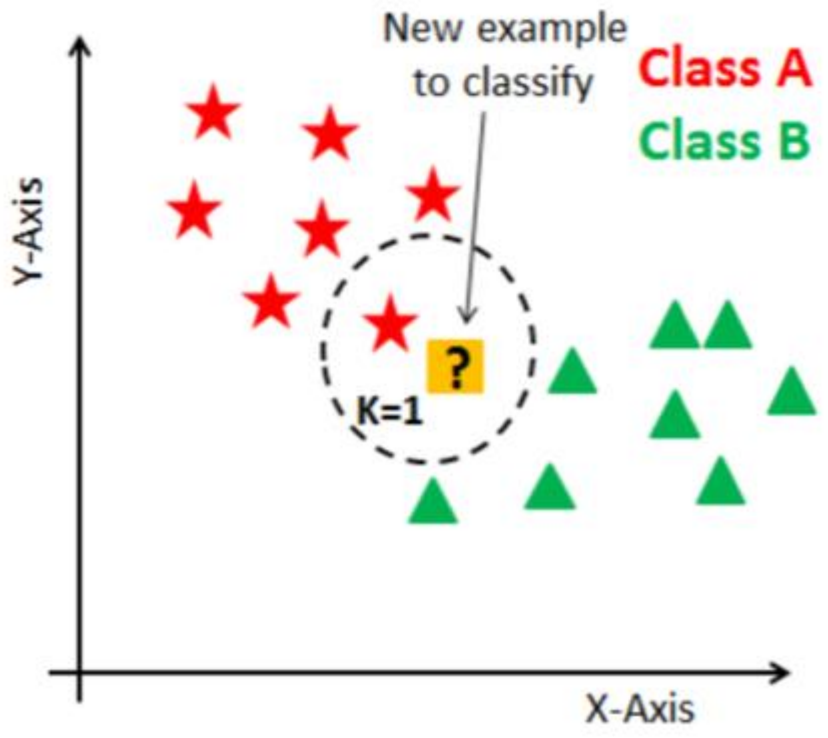


Figure 5: Classifying a new example using KNN

Calculate the following :

1. Compute the distance using various distance measures.
2. Select number of neighbors required for you.
3. Select the nearest neighbors.
4. Obtain the result based on majority value.

5.8.4. APPLICATIONS OF KNN:

1. Data Mining
2. Health Industry
3. Agriculture

5.8.5. PROS OF KNN:

1. KNN Implementation is extraordinarily simple.
2. No Coaching Period.
3. It does not learn anything from the dataset but it memorizes and stores dataset.

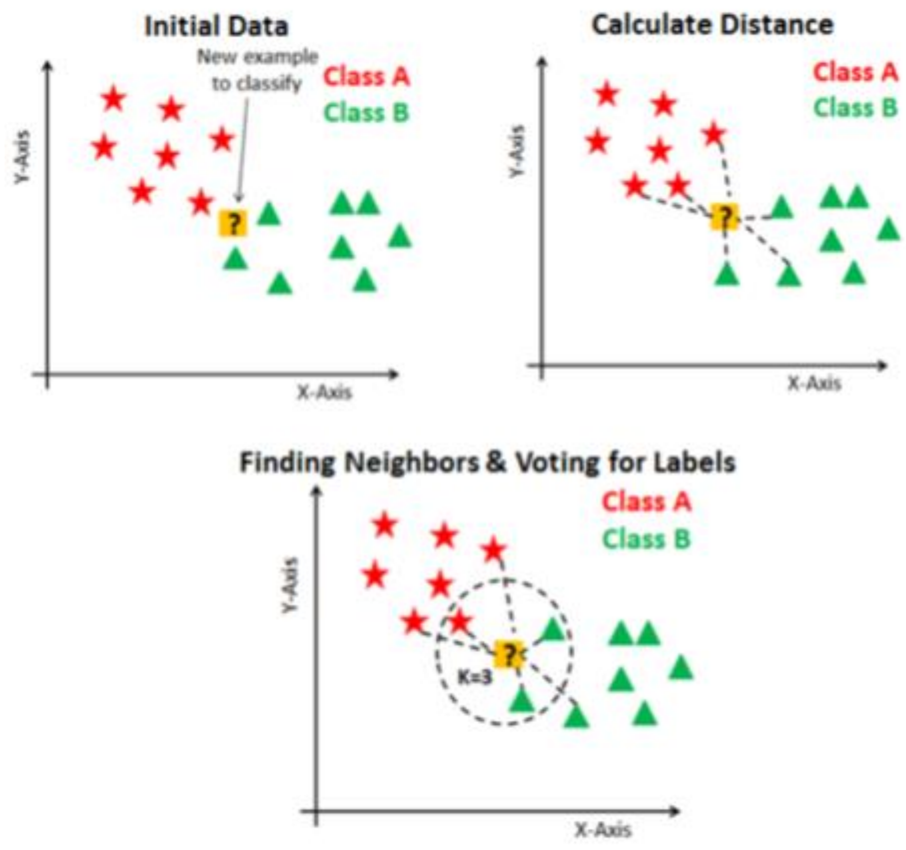


Figure 7: KNN process

The length between the datapoint and its neighbors are measured using:

5.8.6. DISTANCE FUNCTIONS:

It is simply a distance live between a try of samples p and alphabetic character in n-dimensional feature space: In that mostly used distance measures are Euclidean distance, Minkowski distance and Manhattan distance.

The figure shows Euclidean distance

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

5.9. RANDOM FOREST CLASSIFIER:

5.9.1. RANDOM FOREST ALGORITHM:

This algorithm could be a supervised classification rule. This algorithm creates the forest with a variety of decision trees.In general, there are a lot of trees within the forest there is a lot of sturdy the forest feels like. Within the same approach within the random forest classifier, the upper the amount of trees within the forest offers the high the accuracy results.
If you acknowledge the choice tree rule. You'd probably be thinking square measure we tend to make a lot of call trees and therefore the approach will we tend to produce a lot of call trees. As all the calculation of nodes choice goes to be constant for constant dataset.

To model a lot of trees to form the forest you are not attending to use constant apache of constructing the selection with data gain or Gini index approach. If you are not alert to the ideas of call tree classifier, Please pay a short while on the below articles, As you'd prefer to perceive however the selection tree classifier works before you learn the operating nature of the random forest rule.

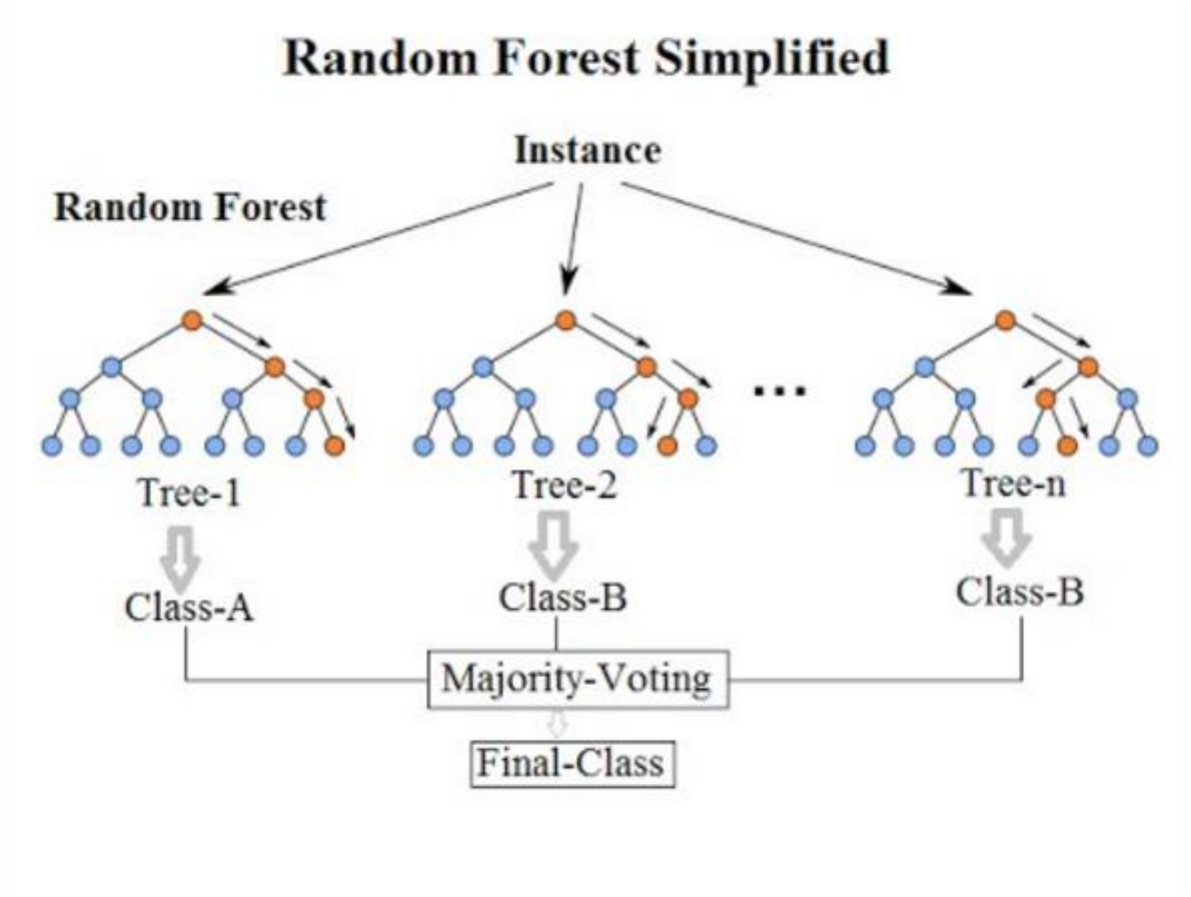


Figure 8: Random Forest Classifier

This algorithm can anticipate a remarkable objective (result) for the equivalent to investigate the data. At that point by considering each imagined objective region unit going to be determined. Assume the 100 arbitrary decision zone unit foreseeing somewhere in the range of 3 unique objectives p, q, r then the votes of p is nothing anyway out of 100 irregular call trees what share forecast is p. Moreover for the other a couple of goals (q, r). In the event that p is acquiring high votes. Suppose out of a hundred irregular elective tree sixty zone units anticipating the objective territory unit going to be p. At that point, the last word arbitrary timberland restores the p on account of the undeniable reality of the foreseen target.

This notion of the vote is understood as **majority selection.**

5.9.2. RANDOM FOREST ALGORITHM APPLICATIONS:

Some applications of random forest algorithms which are widely used are:

1. Banks
2. Health Industry
3. Stock Market Exchange
4. E-commerce Applications

5.9.3. ADVANTAGES OF RANDOM FOREST:

1. Random Forest uses bagging algorithm.

2. Random Forest frequently used to clear up each classifications additionally as regression difficulty.

3. Random Forest toils higher accompanied by each categorized and continuous variables.

4. This algorithm automatically identify and make certain operations on missing data.

5. No function scaling needed.

6. Handles nonlinear parameters efficiently

7. This algorithm can routinely manage lacking values.

8. This algorithm is typically sturdy to deviation and need to take care of them automatically.

9. This algorithm is extremely stable

10. This algorithm is comparatively much less collided through noise.

5.10. SUPPORT VECTOR CLASSIFIER:

A Support Vector Machine in short SVM is a discriminatory classifier described by a dividing hyperplane. In other words, given labelled data (supervised), this algorithm outputs an optimised hyperplane which categorizes new examples.

5.10.1. APPLICATIONS:

This algorithm is used in several companies to handle several real world problems such as:

1. This algorithm is applied in content and also in hypertext association.

2. Arrangement of pictures additionally can be calculated utilizing SVMs'. Exploratory outcomes displays that SVMs' accomplish fundamentally greater hunt precision than customary question improvising plots succeeding 3 to 4 rounds of important input. That is frequently additionally valid for a picture division frameworks, along with those utilising a modified shape SVM that makes use of the exclusive methodology as proposed by 'Vapnik'.

3. Classification of satellite data.

4. Hand-written characters recognition.

5. Biological sciences.

5.10.2. ALGORITHM:

1. SVM desires to discover the optimum line with the condition of accurately classifying either class.

2. Follow the condition: solely check out the different hyperplanes (e.g. separate lines), hyperplanes that classify training correctly.

3. Performance optimization: choose one that enlarges the margin.

5.10.3. PROS:

1. SVM projects comparatively nicely when there is a clear margin of separation between classes.

2. SVM is greater effectual in high dimensional spaces.

3. SVM is fantastic in situations where the number of dimensions is higher than the number of samples.

4. SVM is highly memory well-organised (efficient).

This classifier objectives setting up a hyperplane that can divide the classes as a good deal as viable via adjusting the distance between the fact points and the hyperplane.

There are various kernels build on which the hyperplane is determined.

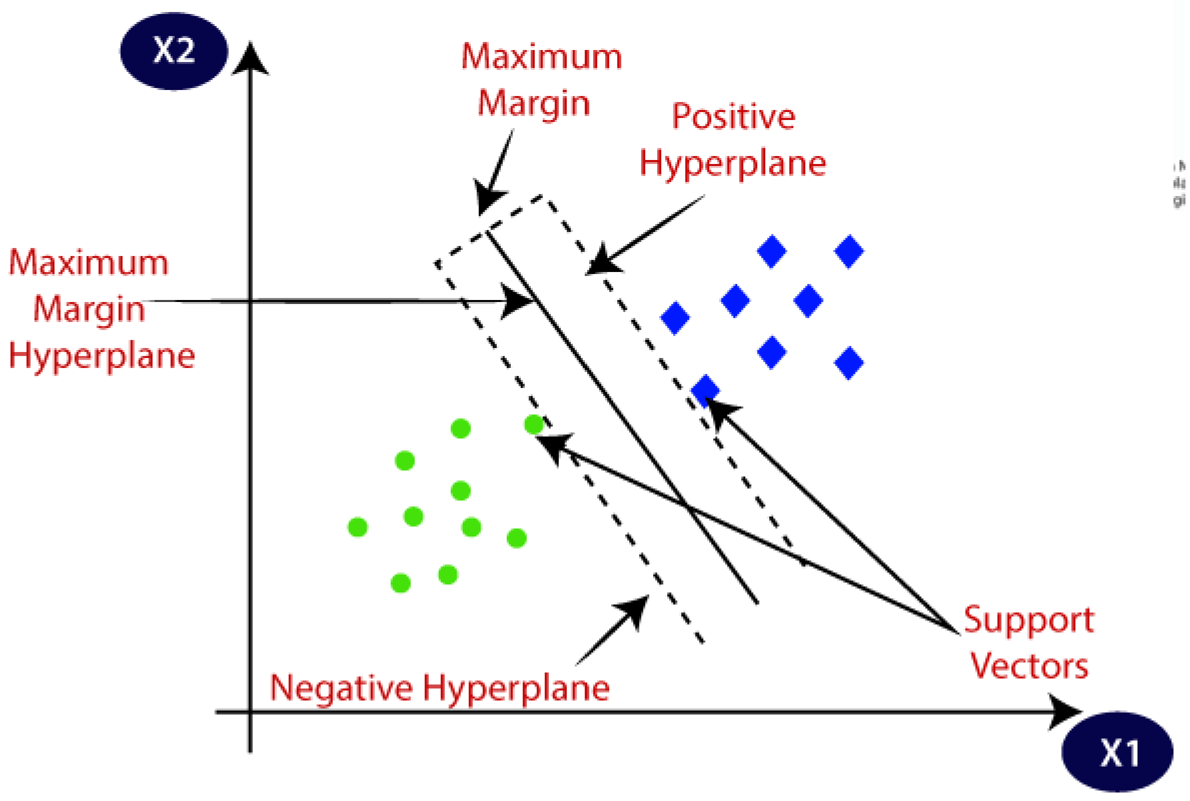


Figure 9: Support Vector Classifier

5.11. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis is an approach to carry out sturdy patterns in a dataset by suppressing variations. It is used to clean information units to make it convenient to explore and analyse.

The Principal Component Analysis algorithm is based on some arithmetical ideas namely:

1. Variance and Covariance
2. Eigen Vectors and Eigen values

Principal Component Analysis (PCA) could be an analytical method that utilizes an orthogonal change which turns a group of correlated variables to a group of uncorrelated variables.

PCA could be almost all generally used device in investigational (exploratory) information analysis and in machine learning for forecasting models. Moreover, PCA is an unsupervised analytical technique wont to test the interdependencies among a group of variables. It's also called a general correlational analysis where regression decides a line of finest fit.

5.11.1. APPLICATIONS:

1. Recognition of image
2. Facial recognition
3. Computer vision projects

5.11.2. ADVANTAGE OF PRINCIPAL COMPONENT ANALYSIS:

1. Eliminate Correlated Features
2. Enhances Algorithm Performance
3. Eliminate Over-fitting
4. Enhances Visualization

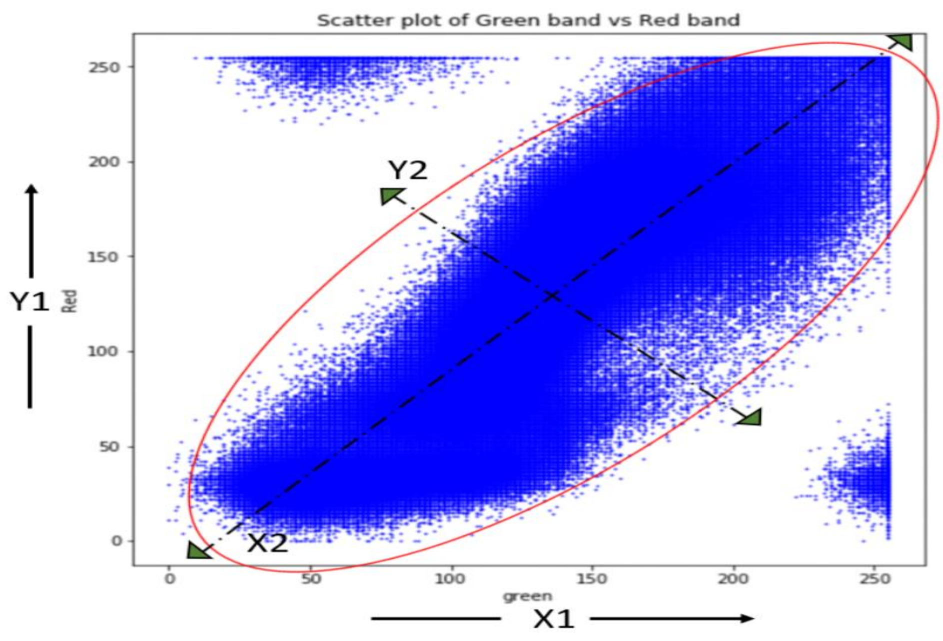


Figure 10: Principal Components

6. SYSTEM REQUIREMENTS:

6.1 HARDWARE REQUIREMENTS:

1. Processor	:	Minimum I3 processor is required because I3 processor can train an average of 100 examples per second.
2. RAM	:	Minimum 4 SGB
3. Storage	:	Minimum 256 GB

6.2 SOFTWARE REQUIREMENTS:

1. Operating system	:	Any latest operating system available preferably Windows operating system.
2. Programming Language	:	Python
3.Python packages	:	Numpy, Matplotlib, Pandas, Sklearn(Scikit Learn)

7. SAMPLE CODE AND OUTPUT:

7.1 SAMPLE CODE:

#here we import all the required libraries for program execution

#this function is used to import dataset in our case it is named as heart.csv

```
data=pd.read_csv('heart.csv')
```

#this head function by default displays the starting 5 rows of dataset otherwise we can give any n number inside function to display n rows of dataset

```
data.head()
```

#this displays histogram of every variable as

```
data.hist()
```

#it returns the count of each target values

```
data.target.value_counts()
```

#this plots the count of each target values in bar chart

```
sns.countplot(x="target", data=data, palette="bwr")
plt.show()
```

#this plots the count of males and females in bar chart

```
sns.countplot(x='sex', data=data, palette="mako_r")

plt.xlabel("Sex (0 = female, 1= male)")

plt.show()
```

# this plots the max heart rate achieved by both disease and non-disease patients

```
plt.scatter(x=data.age[data.target==1],

y=data.thalach[(data.target==1)], c="green")

plt.scatter(x=data.age[data.target==0],

y=data.thalach[(data.target==0)], c = 'black')

plt.legend(["Disease", "No Disease"])

plt.xlabel("Age")   plt.ylabel("Maximum Heart Rate")

plt.show()
```

#this iloc is used to allocate rows and columns which here we stored in x and y variables [number of rows, number of columns]

```
x = data.iloc[:,:-1].values
y = data.iloc[:,13].values
```

# here we are assigning k fold cross validation with 10 splits which means 10 fold cross

```
kf=KFold(n_splits=10)
```

#here we are taking empty lists

```
s= []
scores= []
```

#logic when applying KNN to the dataset with K-fold cross validation

```
for train_data, test_data in kf.split(x):
#    print("train:",train_data,"test:",test_data)
    x_traink, x_testk = x[train_data], x[test_data]
    y_traink, y_testk = y[train_data], y[test_data]
    algorithm = KNeighborsClassifier(n_neighbors=3)
    algorithm.fit(x_traink, y_traink)
    y_Predictionk = algorithm.predict(x_testk)
    print("Accuracies in each step:",metrics.accuracy_score(y_testk, y_Predictionk))
    scoresknn=[metrics.accuracy_score(y_testk, y_Predictionk)]
    sknn=sknn+scoresknn
```

#logic when applying Decision trees to the dataset with K-fold cross validation

```
sdt=[]

scoresdt=[]

for train_data, test_data in kf.split(x):

#    print("train:",train_data,"test:",test_data)

    x_traind, x_testd = x[train_data], x[test_data]

    y_traind, y_testd = y[train_data], y[test_data]

    algorithm = DecisionTreeClassifier()

    algorithm.fit(x_traind, y_traind)

    y_Predictiond = algorithm.predict(x_testd)

    print("Accuracies in each step:",metrics.accuracy_score(y_testd, y_Predictiond))

    scoresdt=[metrics.accuracy_score(y_testd, y_Predictiond)]

    sdt=sdt+scoresdt

print(sum(sdt)/len(sdt))
```

#logic when applying Random forest to the dataset with K-fold cross validation

```
srf=[]

scoresrf=[]

for train_data, test_data in kf.split(x):

#    print("train:",train_data,"test:",test_data)

    x_trainr, x_testr = x[train_data], x[test_data]

    y_trainr, y_testr = y[train_data], y[test_data]

    algorithm = RandomForestClassifier()

    algorithm.fit(x_trainr, y_trainr)

    y_Predictionr = algorithm.predict(x_testr)

    print("Accuracies in each step:",metrics.accuracy_score(y_testr, y_Predictionr))

    scoresrf=[metrics.accuracy_score(y_testr, y_Predictionr)]

    srf=srf+scoresrf

print(sum(srf)/len(srf))
```

#logic when applying Support Vector Machine to the dataset with K-fold cross validation

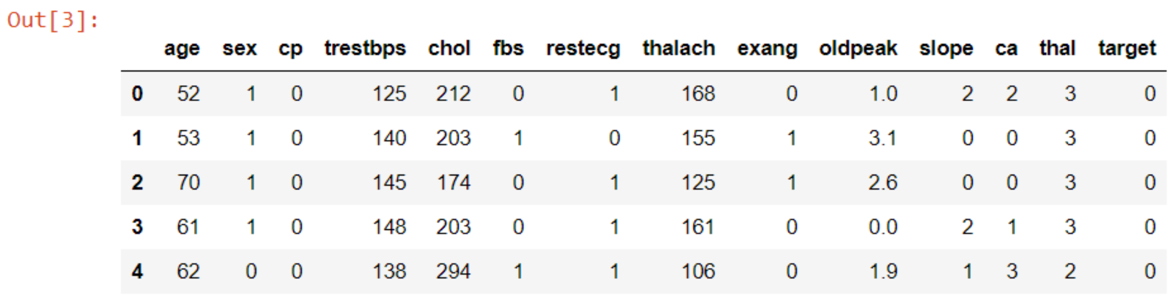
```
for train_data, test_data in kf.split(x):
#    print("train:",train_data,"test:",test_data)
    x_train, x_test = x[train_data], x[test_data]
    y_train, y_test = y[train_data], y[test_data]
    svm = SVC()
    svm.fit(x_train, y_train)
    y_Prediction = svm.predict(x_test)
    print("Accuracies in each step:",metrics.accuracy_score(y_test, y_Prediction))
    scores=[metrics.accuracy_score(y_test, y_Prediction)]
    s=s+scores

for train_data, test_data in kf.split(x):
#    print("train:",train_data,"test:",test_data)
    x_train, x_test = x[train_data], x[test_data]
    y_train, y_test = y[train_data], y[test_data]
    svm = SVC()
    svm.fit(x_train, y_train)
    y_Prediction = svm.predict(x_test)
    print("Accuracies in each step:",np.mean(y_test==y_Prediction))
    scores=[metrics.accuracy_score(y_test, y_Prediction)]

    s=s+scores
```

#logic when applying Principal Component Analysis to the dataset with K-fold cross validation

```
sp=[]

scoresp=[]

for train_data, test_data in kf.split(x):

#    print("train:",train_data,"test:",test_data)

    x_trainp, x_testp = pca.fit_transform(x[train_data]), pca.transform(x[test_data])

    y_trainp, y_testp = y[train_data], y[test_data]

    algorithm = DecisionTreeClassifier()

    algorithm.fit(x_trainp, y_trainp)

    y_Predictionp = algorithm.predict(x_testp)

    print("Accuracies :",metrics.accuracy_score(y_testp, y_Predictionp))

    scoresp=[metrics.accuracy_score(y_testp, y_Predictionp)]

    sp=sp+scoresp

print(sum(sp)/len(sp))
```

7.2 OUTPUTS:

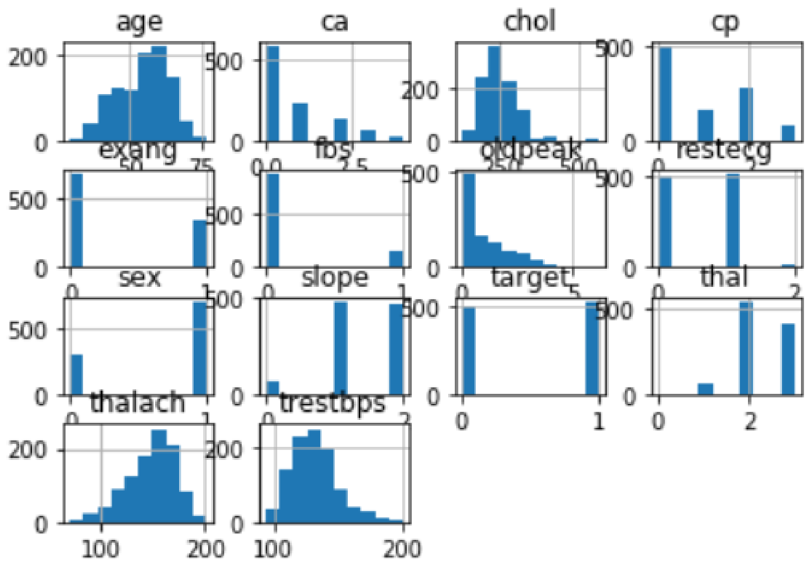
data.head() output:



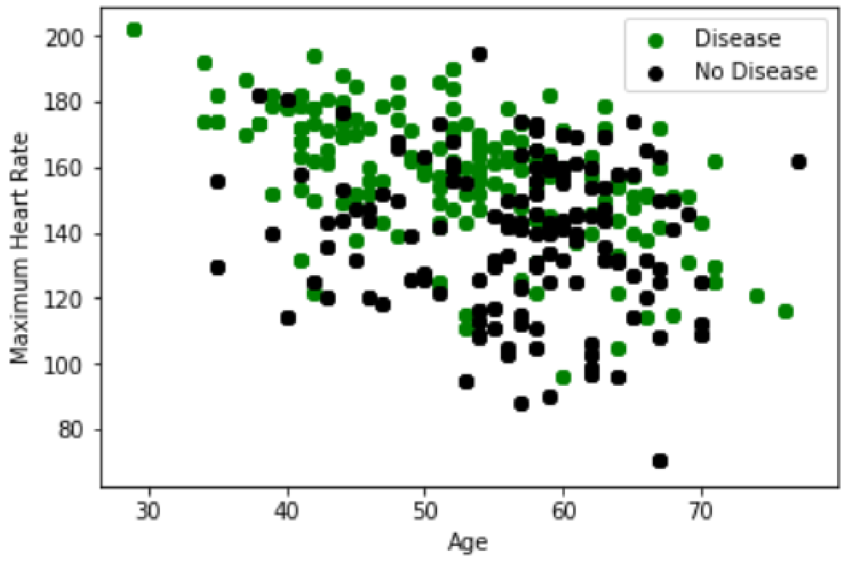
Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

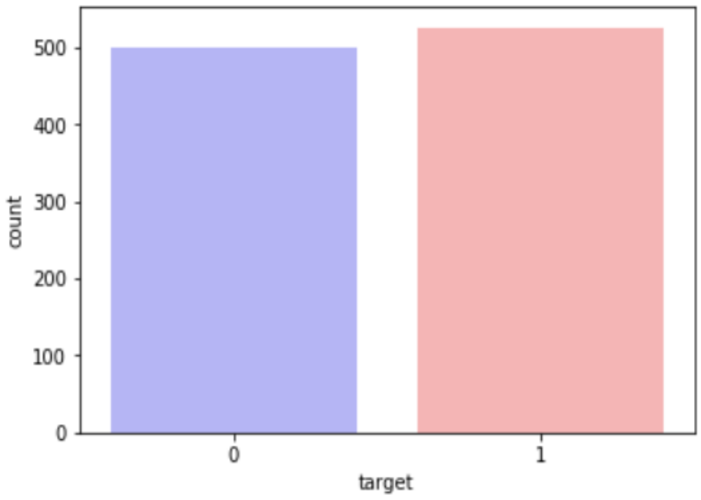
Data.hist() output:



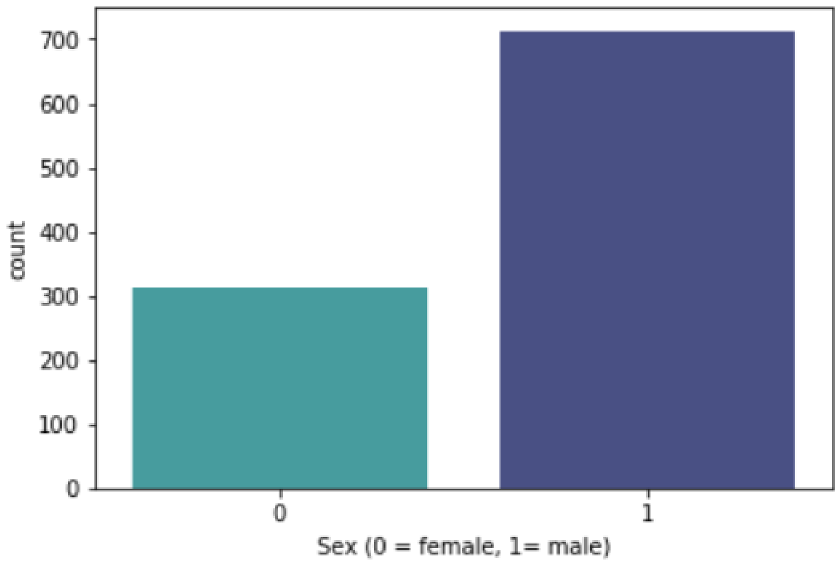
Maximum heart rate achieved output:



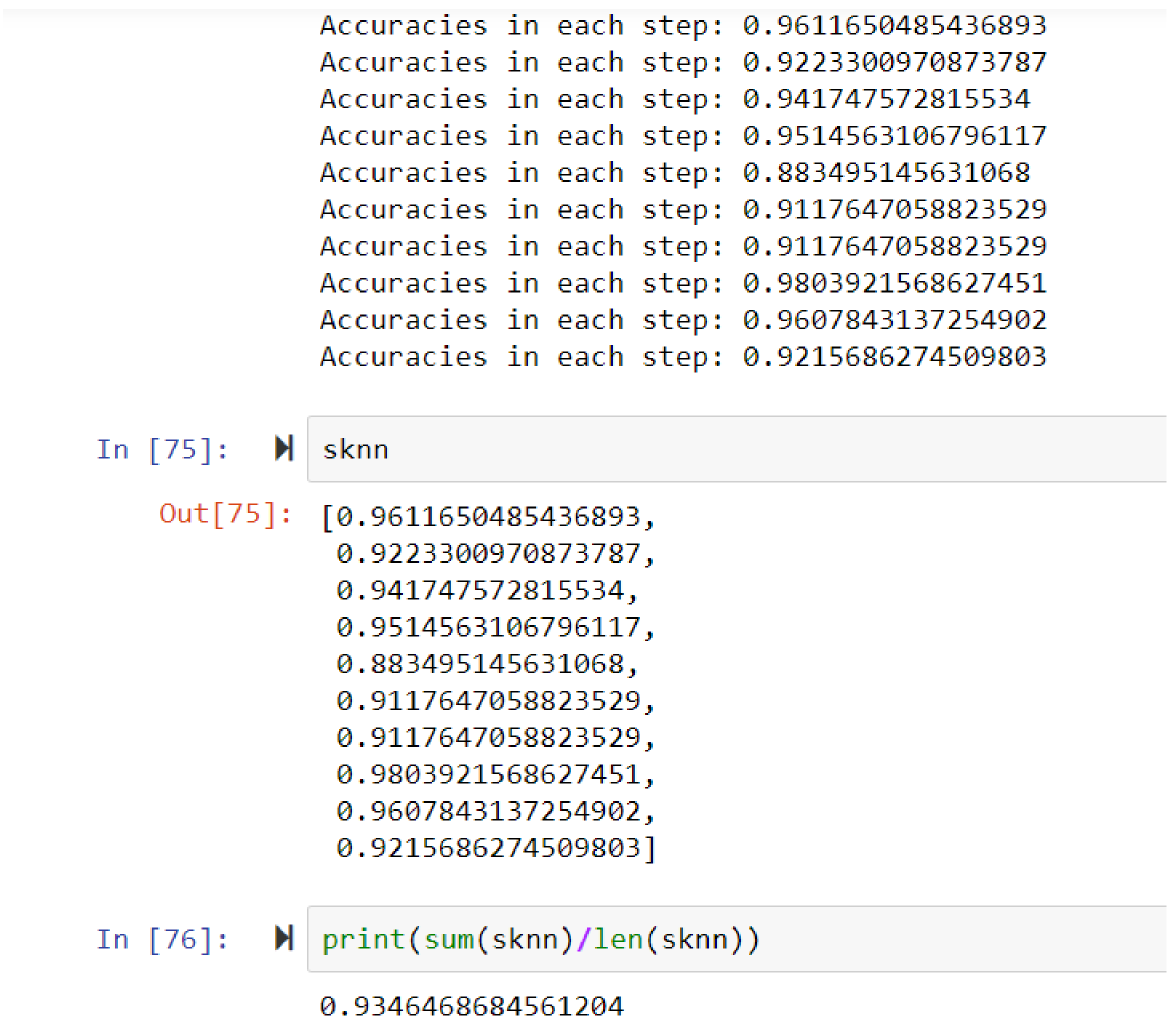
#this plots the count of each target values in bar chart



#bar chart of number males and females in the dataset

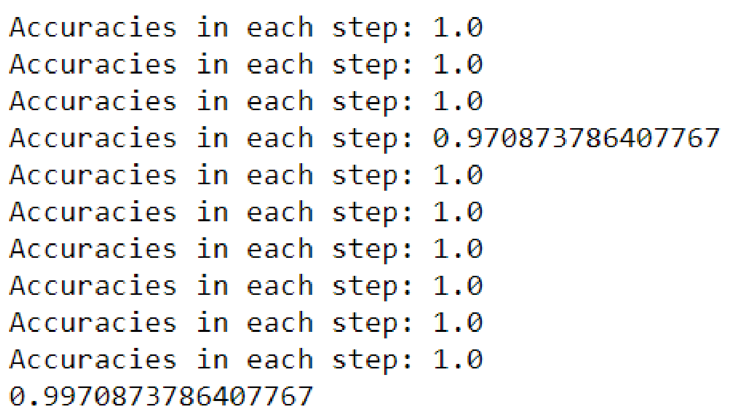


#output of KNN



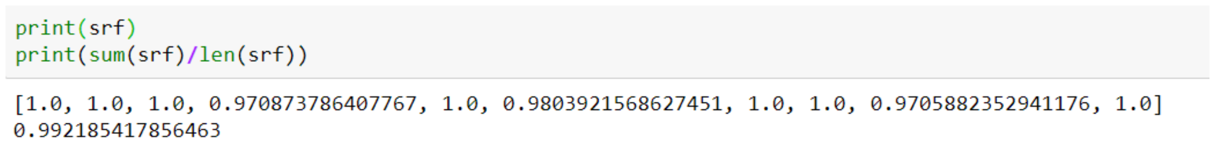
```
Accuracies in each step: 0.9611650485436893
Accuracies in each step: 0.9223300970873787
Accuracies in each step: 0.941747572815534
Accuracies in each step: 0.9514563106796117
Accuracies in each step: 0.883495145631068
Accuracies in each step: 0.9117647058823529
Accuracies in each step: 0.9117647058823529
Accuracies in each step: 0.9803921568627451
Accuracies in each step: 0.9607843137254902
Accuracies in each step: 0.9215686274509803

In [75]:  sknn

Out[75]:  [0.9611650485436893,
           0.9223300970873787,
           0.941747572815534,
           0.9514563106796117,
           0.883495145631068,
           0.9117647058823529,
           0.9117647058823529,
           0.9803921568627451,
           0.9607843137254902,
           0.9215686274509803]

In [76]:  print(sum(sknn)/len(sknn))

          0.9346468684561204
```

#output of Decision tree



```
Accuracies in each step: 1.0
Accuracies in each step: 1.0
Accuracies in each step: 1.0
Accuracies in each step: 0.970873786407767
Accuracies in each step: 1.0
Accuracies in each step: 1.0
Accuracies in each step: 1.0
Accuracies in each step: 1.0
Accuracies in each step: 1.0
Accuracies in each step: 1.0
0.9970873786407767
```

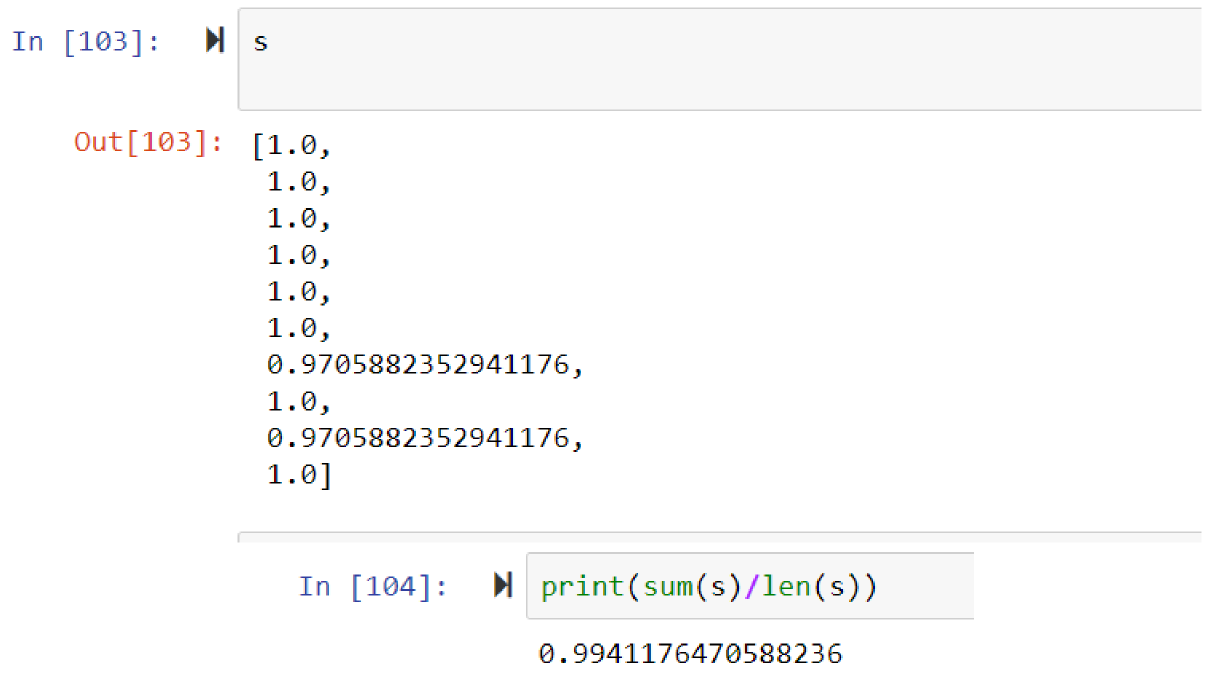
#output of random forests



```
print(srf)
print(sum(srf)/len(srf))
```

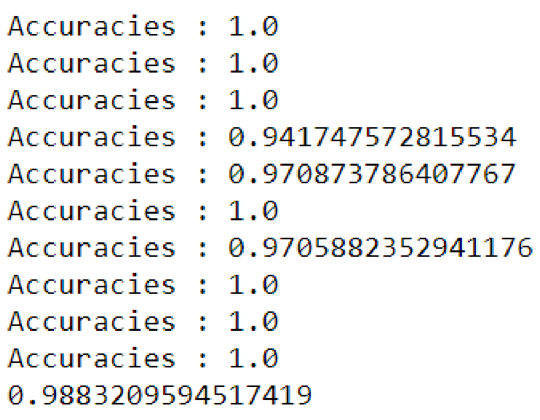
```
[1.0, 1.0, 1.0, 0.970873786407767, 1.0, 0.9803921568627451, 1.0, 1.0, 0.9705882352941176, 1.0]
0.992185417856463
```

#output of SVM



```
In [103]:  s

Out[103]: [1.0,
           1.0,
           1.0,
           1.0,
           1.0,
           1.0,
           0.9705882352941176,
           1.0,
           0.9705882352941176,
           1.0]

In [104]:  print(sum(s)/len(s))

           0.9941176470588236
```

#output of PCA



```
Accuracies : 1.0
Accuracies : 1.0
Accuracies : 1.0
Accuracies : 0.941747572815534
Accuracies : 0.970873786407767
Accuracies : 1.0
Accuracies : 0.9705882352941176
Accuracies : 1.0
Accuracies : 1.0
Accuracies : 1.0
0.9883209594517419
```

8. COMPARISON TABLE:

	Decision tree	**Random Forest**	**KNN**	**SVM**	**PCA**
Accuracy	99.7%	99.5%	93%	99.4%	85.2%
Computation time	6.237	8.526	4.217	12.456	18.725
Advantages	Very simple to elucidate for minor-sized decision trees	Random forest is less compact to elucidate(interprett)	Does requires no training	Working well even if the dataset is not linearly separable	Easy to compute
Disadvantages	Large trees can be difficult to interpret	It requires more time to train	Does not work with a large dataset	High memory usage	Depend on previous knowledge
Used	classification	Classification & regression	Clustering	Classification	Classification & regression

9. FINAL ANALYSIS AND CONCLUSION:

In our project, we first pre-processed the data first then we have applied k-fold cross validation and data science algorithms. The following are the accuracies of each machine learning algorithms we applied.

1.	K Neighbors Classifier	:	93%.
2.	Decision Tree Classifier	:	99.7%.
3.	Random Forest Classifier	:	99.5%.
4.	Principal Component analysis	:	85.2%.
5.	Support vector machine	:	99.4%.

During this project, various classification methods are executed on the dataset to predict heart diseases. Classification algorithms are used to predict set of relations between attributes within the datasets to create an accurate classifier. The most contribution of this project is to achieve high calculation accuracy for early diagnosis of heart diseases. Finally, an intelligent system is developed to see the danger of heart diseases on the idea of assumed parameters.

10. REFERENCES:

1. "Random Forests for Regression as a Weighted Sum of k-Potential Nearest Neighbours". Donghua Yu ; Guojun Liu ; Maozu Guo ; Xiaoyan Liu ; Shuang Yao.

2. A Novel Density Peaks Clustering Halo Node Assignment Method Based on K-Nearest Neighbor Theory. Limin Wang ; Wei Zhou ; Honghuan Wang ; Milan Parmar ; Xuming Han.

3. Usage of Machine Learning for Strategic Decision Making at Higher Educational Institutions. Yuri Nieto ; Vicente Gacía-Díaz ; Carlos Montenegro ; Claudio Camilo González ; Rubén González Crespo.

4. Efficient and Private Scoring of Decision Trees, Support Vector Machines and Logistic Regression Models Based on Pre-Computation,Martine De Cock ; Rafael Dowsley ; Caleb Horst ; Raj Katti ; Anderson C. A. Nascimento ; Wing-Sea Poon ; Stacey Truex.

5. A missing power data filling method based on improved random forest algorithm. Wei Deng ; Yixiu Guo ; Jie Liu ; Yong Li ; Dingguo Liu ; Liang Z.

6. Random Forests for Regression as a Weighted Sum of **k** -Potential Nearest Neighbors Pablo Fernández-González ; Concha Bielza ; Pedro Larrañaga.

7. A Feature Selection Based Serial SVM Ensemble Classifier Jianjun Cao ; Guojun Lv ; Chen Chang ; Hongmei Li.

8. Network Security Situation Prediction Based on MR-SVM
Jingjing Hu ; Dongyan Ma ; Chen Liu ; Zhiyu Shi ; Huaizhi Yan ; Changzhen Hu

9. Group-Wise Principal Component Analysis for Exploratory Intrusion Detection
José Camacho ; Roberto Therón ; José M. García-Giménez ; Gabriel Maciá-Fernández ; Pedro García-Teodoro.

10. Local and Global Randomized Principal Component Analysis for Nonlinear Process Monitoring
Ping Wu ; Lingling Guo ; Siwei Lou ; Jinfeng Gao