SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

15IT313J MAJOR PROJECT

A STUDY ON IMPROVED PREDICTION SYSTEM FOR CORONARY HEART DISORDER

Guide
Mr. Kottilingam Kottursamy

RA1711008010139
Vishesh Hupta

ABSTRACT

- Our principle commitment is to predict with least number of attributes and with best accuracy.
- The reduction of dimension in primarily done using backward elimination.
- Our solution utilizes 4 different algorithms namely k-Nearest Neighbor, Naïve Bayes, Random Forest, Decision Tree for prediction.
- The user can add values of attributes through a web based portal.
- The user data is then taken and identified with the trained data.

INTRODUCTION

- Data mining is a critical task during the technique of knowledge discovery in databases in which insightful techniques are utilised as a part of request to obtain patterns.
- The fundamental point of records mining is to determine fresh patterns on behalf of customers and to disclose the facts patterns to run important and significant data for the clients.
- Heart is one of the most important organs of human body. To define malfunction of heart a common general term is used which is cardiac/mind disease. There are various forms of mind bug like heart attack, heart failure, CHD etc.
- Though there are different forms there are some common risks factor which decides whether someone will be at risk of heart disease or not.

LITERATURE SURVEY

S.No	Paper	Authors	Year	Algorithms Used	Dataset	Language Used	Finding	Advantages
1	Predictive Modeling of Hospital Mortality for Patients With Heart Failure by Using an Improved Random Survival Forest	FEN MIAO, YUN-PENG CAI, YU-XIAO ZHANG, XIAO-MAO FAN AND YE LI	2017 IEEE Access	IMPROVED RANDOM SURVIVAL FOREST	MIMIC II clinical database	R version 3.4.1	OOB C-statistics value of 0.821.	The model separated the 1-year survivors and non-survivors with much greater accuracy than previous heart failure models,
2	An Artificial Neural Network Based Pattern Classification Algorithm for Diagnosis of Heart Disease	Balasaheb Tarle and Sudarson Jena	2017 IEEE	ANN and five-fold cross validation	Heart Disease Dataset from UCI Machine Learning Repository	Python	83% classification accuracy	reduces complexity and increases accuracy
3	Diagnosis of Heart Disease Using a Mixed Classifier	Sarawut Meesri, Suphakant Phimoltares, Atchara Mahaweerawat	2017 ICSEC	Naïve Bayes approachrpe, Support Vector Machine, K-Nearest Neighbor, Multi-Layer Perceptron and ANN	Heart Disease Dataset from UCI Machine Learning Repository	Python	Accuracy:86 % FPR: 9.76%	Method outperforms the other techniques in terms of accuracy and false positive rate.
4	Heart Disease Prediction System Using CART-C	Priyal Chotwani, Asmita Tiwari, Vikas Deep, Purushottam Sharma	2018 ICCCI	All Possible-MV algo, CART-C algo	Heart Disease Dataset from UCI Machine Learning Repository	Python		This dataset considers various attributes which can lead to heart disease. These factors are usually excluded or ignored in large hospitals.

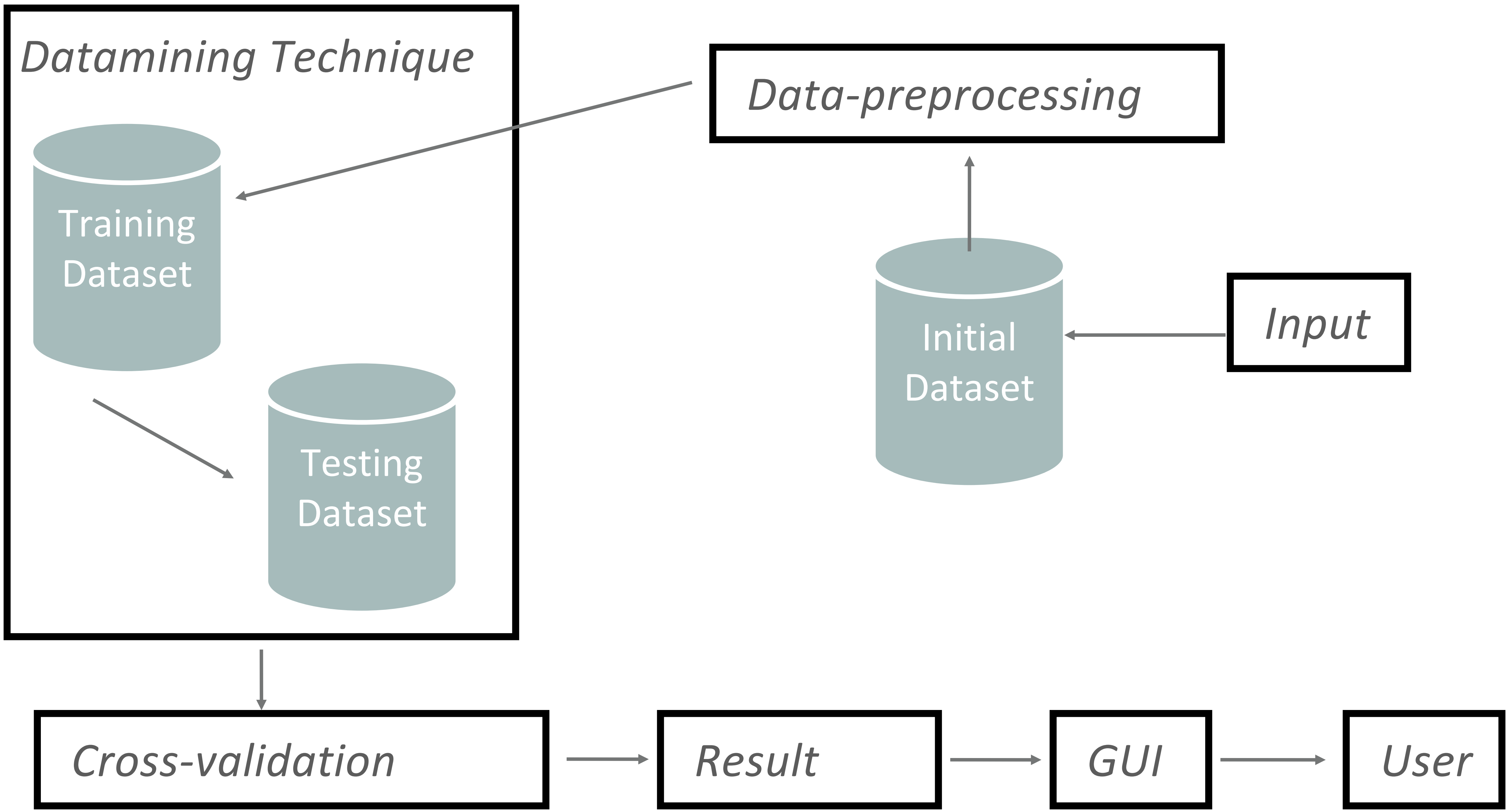
5	A Scalable Solution for Heart Disease Prediction using Classification Mining Technique	Rashmi G Saboji and Prem Kumar Ramesh	2017 IEEE	Random Forest, Naive Bayes	Heart Disease Dataset from UCI Machine Learning Repository	Java, Python	98% accuracy	More Scalable as used HDFS for supporting large dataset.
6	An Ensemble Based on Distances for a kNN Method for Heart Disease Diagnosos	Alberto Palacios Pawlovsky	2018 IEEE	kNN, Naive Bayes, Decision Tree, Support Vector Machine	Heart Disease Dataset from UCI Machine Learning Repository	Python	85% accuracy	Used Ensemble Based Distance for kNN which increases the accuracy of the Algorithm
7	Heart Disease Prediction Using Data Mining Techniques	Abhisek Rairikar, Vedant Kulkarni, Vikas Sabale, Harshavarshan Kale, Anuradha Lamgunde	2017 I2C2	KNN, Decision Trees=s, Naive Bayes	Heart Disease Dataset from UCI Machine Learning Repository	Python, HTML, CSS	Algithem Searching Time: Naive:- 58, Decision Tree:- 31, kNN:- 2	Better Result Accuracy, Reduced Time Complexity
8	Web Analytics Support System for Prediction of Heart Disease Using Naïve Bayes Weighted Approach (NBwa)	Priyanga and Dr. Naveen	2017 AMS	Naïve Bayes Weighted Approach	Heart Disease Dataset from UCI Machine Learning Repository	Python	86% accuracy	when the number of records was increased above 500 records minimal change in the accuracy was observed.

INFERENCE FROM SURVEY

- It is observed that all the papers used the same parameters which they thought were suitable for the disease prediction.
- Most of the paper’s parameter were ranging from 13 to 15.
- 90% of the research was done using UCI Machine Learning repository.

ARCHITECTURE DIAGRAM

Overview of the System

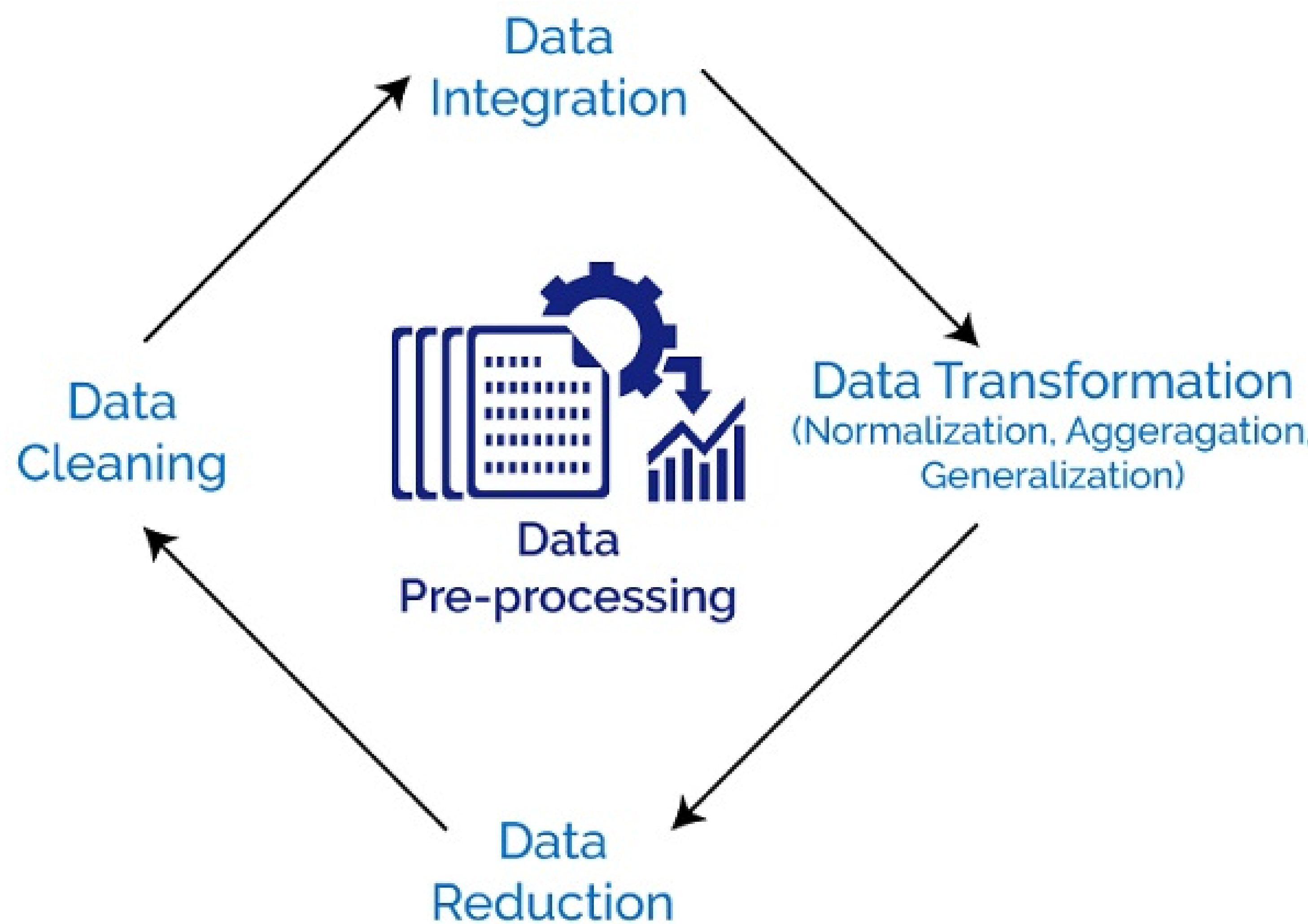


OBJECTIVE OF THE PROJECT

- Reducing the no. of attributes to determine the possibility of heart disease.
- Reduce False Positive Rate (FPR).
- Overcoming the limitation of locally or temporally stable association with continually updating the data and algorithm.
- Use analytics for clinical decision support.
- Use analytics for better care coordination.

MODULES

DATA PREPROCESSING



DATA PREPROCESSING

Data Preprocessing involved following steps:

- Data Cleaning
- Feature Selection

DATA PREPROCESSING

Data Cleaning:

- The missing values in dataset were represented via -9.
- The data set contained initially 75 columns, removing columns with missing values reduced it to 55 columns
- The Imputer function of scikit-learn library was used.
- The values were replaced using mean strategy.

```
from sklearn.preprocessing import Imputer

imputer = Imputer(missing_values=-9, strategy='mean', axis=0)

imputer = imputer.fit(X)

X = imputer.transform(X)
```

DATA PREPROCESSING

Feature Selection

- Feature selection was done to reduce the number of parameters from 55 to 10.
- Backward Elimination was used to achieve this goal.
- The significance level used is 0.01

```
Import statsmodels.formula.api as sm
def backwardElimination(x, sl):
#backward elemination\n",
    numVars = len(x[0])
    for i in range(0, numVars):
        regressor_OLS = sm.OLS(Y, x).fit()
        maxVar = max(regressor_OLS.pvalues).astype(float)
        if maxVar > sl:
            for j in range(0, numVars - i):
                if (regressor_OLS.pvalues[j].astype(float) == maxVar):
                    print(j)
                    print(x[:,j][0:10])
                    x = np.delete(x, j, 1)
    regressor_OLS.summary()
    return x

SL = 0.01
X_opt = X
X_Modeled = backwardElimination(X_opt, SL)
```

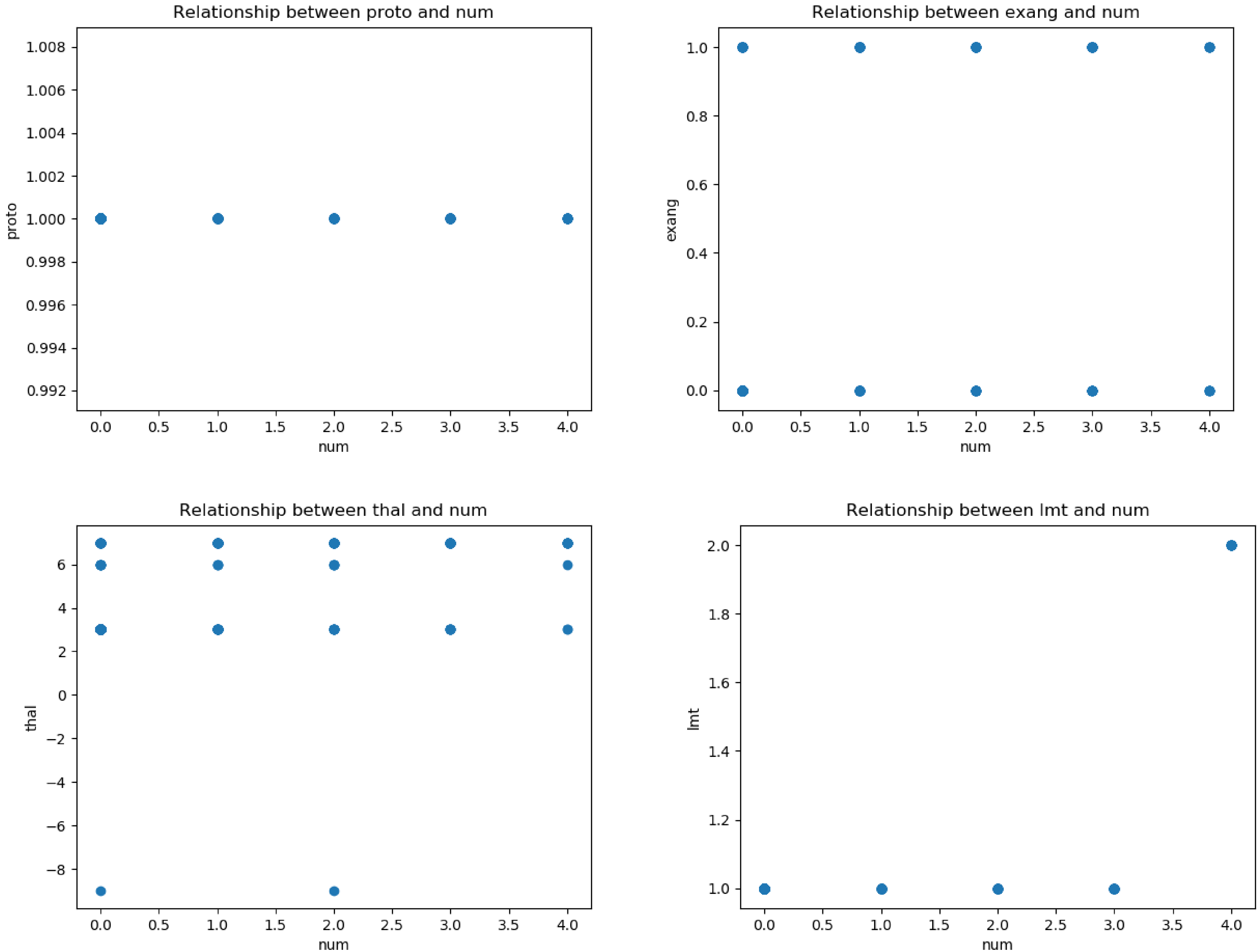
DATA PREPROCESSING

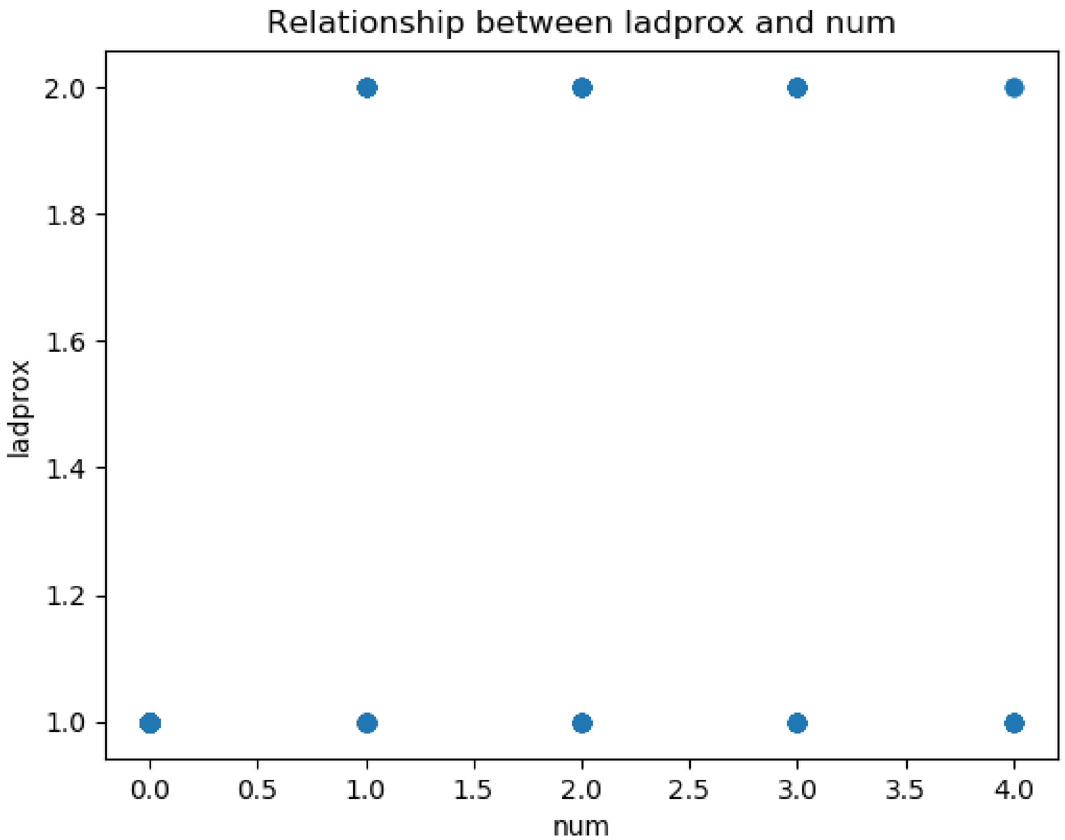
Initial Features

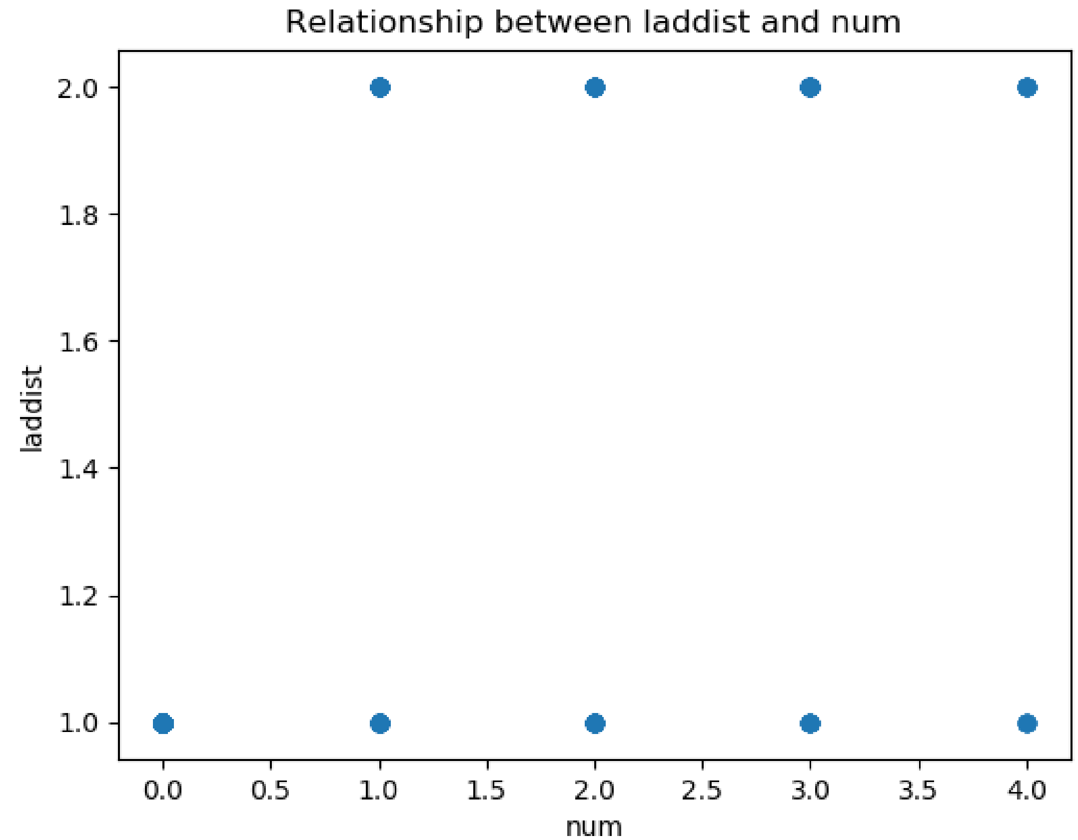
['id','ccf','age','sex','painloc','painexer','relrest','pncaden','cp','trestbps','htn','chol','smoke','cigs','years','fbs','dm','famhist','restecg','ekgmo','ekgday','ekgyr','dig','prop','nitr','pro','diuretic','proto','thaldur','thaltime','met','thalach','thalrest','tpeakbps','tpeakbpd','dummy','trestbpd','exang','xhypo','oldpeak','slope','rldv5','rldv5e','ca','restckm','exerckm','restef','restwm','exeref','exerwm','thal','thalsev','thalpul','earlobe','cmo','cday','cyr','num','lmt','ladprox','laddist','diag','cxmain','ramus','om1','om2','rcaprox','rcadist','lvx1','lvx2','lvx3','lvx4','lvf','cathef','junk','name']

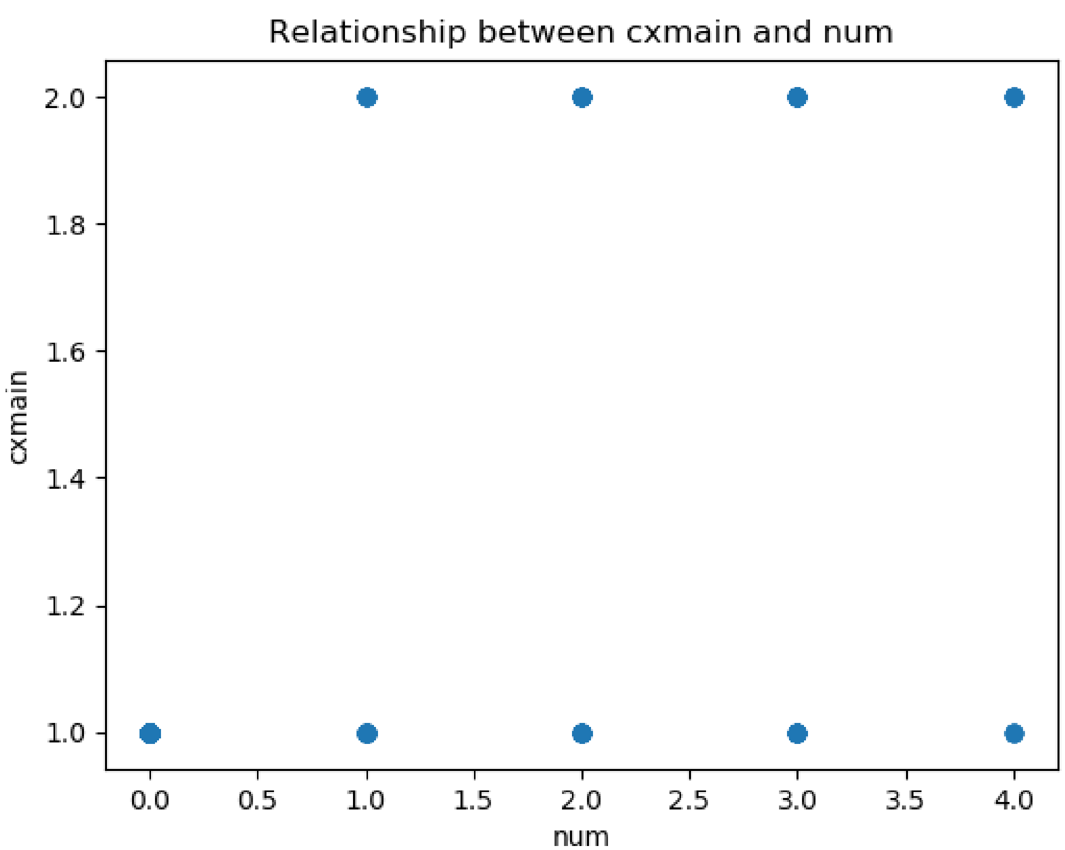
Final Features

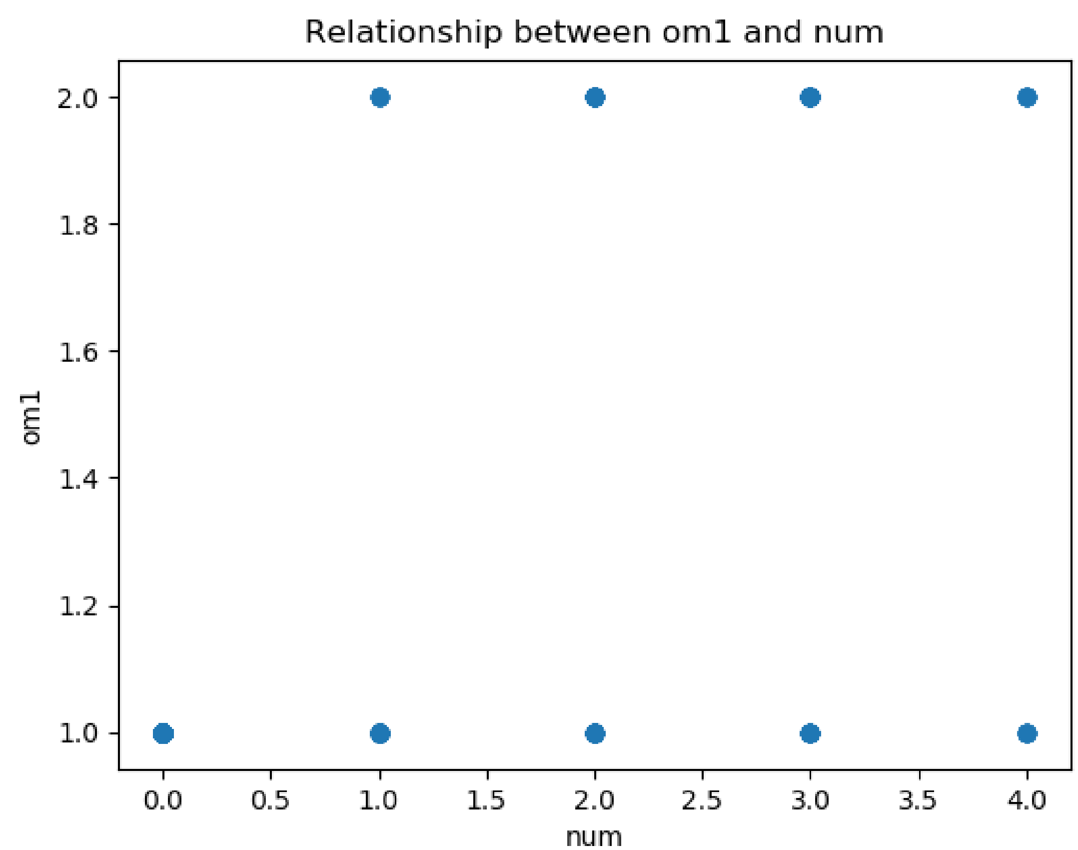
- 'proto‘
- 'exang‘
- 'thal‘
- 'lmt‘
- 'ladprox‘
- 'laddist‘
- 'cxmain‘
- 'om1‘
- 'rcaprox‘
- 'rcadist'

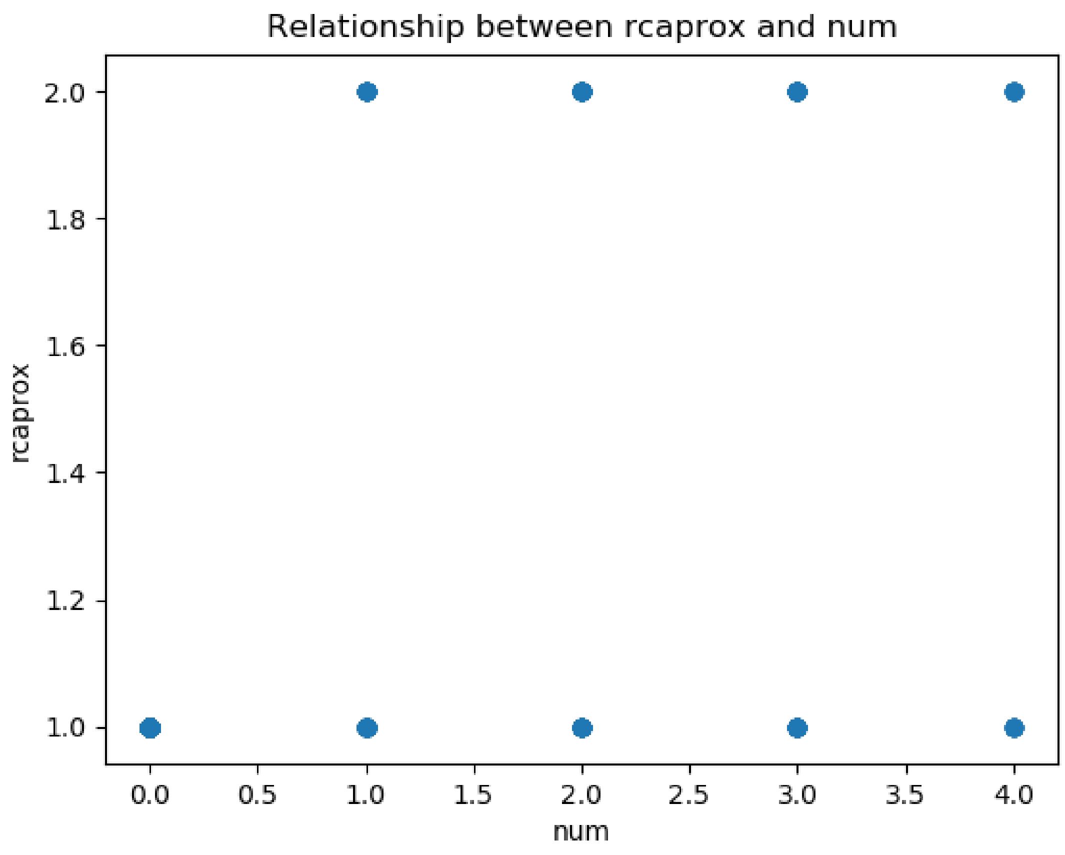

Relationship between proto and num
proto
1.008
1.006
1.004
1.002
1.000
0.998
0.996
0.994
0.992
num
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Relationship between exang and num
exang
1.0
0.8
0.6
0.4
0.2
0.0
num
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Relationship between thal and num
thal
6
4
2
0
−2
−4
−6
−8
num
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
Relationship between lmt and num
lmt
2.0
1.8
1.6
1.4
1.2
1.0
num
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0

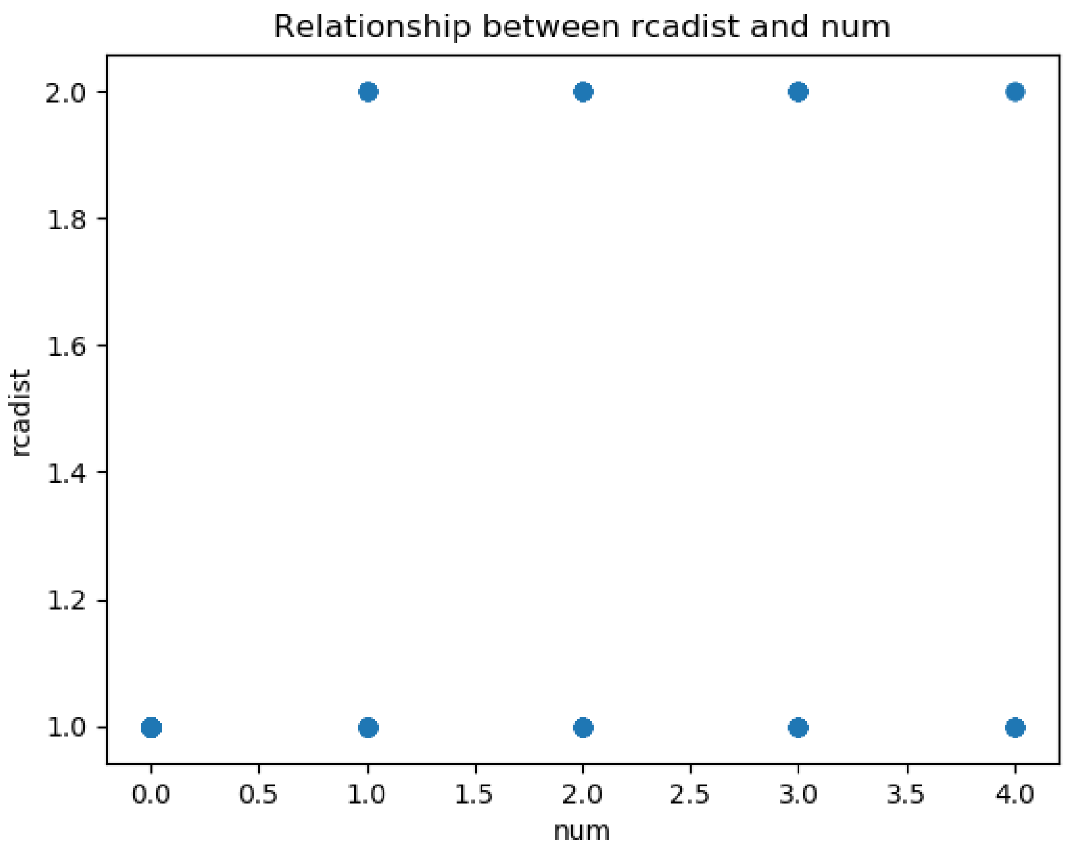

Relationship between ladprox and num
ladprox
num
2.0
1.8
1.6
1.4
1.2
1.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0


Relationship between laddist and num
laddist
num
2.0
1.8
1.6
1.4
1.2
1.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0

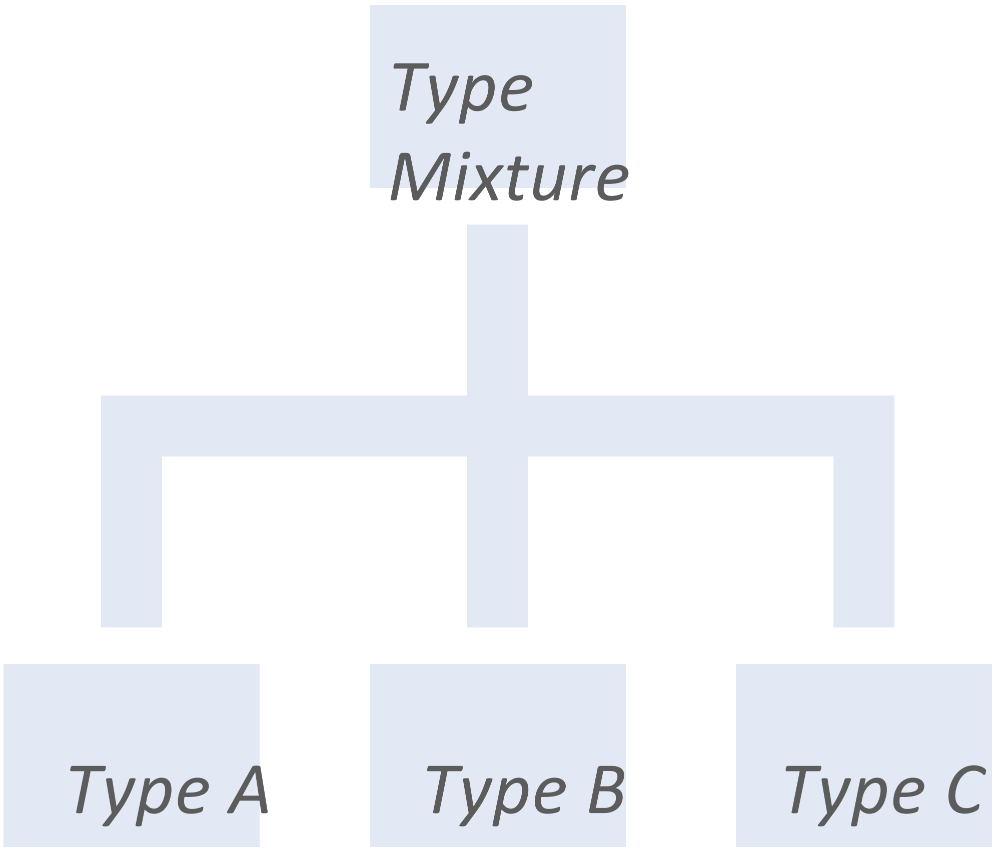

Relationship between cxmain and num
cxmain
num
2.0
1.8
1.6
1.4
1.2
1.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0


Relationship between om1 and num
om1
num
2.0
1.8
1.6
1.4
1.2
1.0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0


Relationship between rcaprox and num
rcaprox
num
2.0
1.8
1.6
1.4
1.2
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Relationship between rcadist and num
rcadist
num
2.0
1.8
1.6
1.4
1.2
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Classification Model



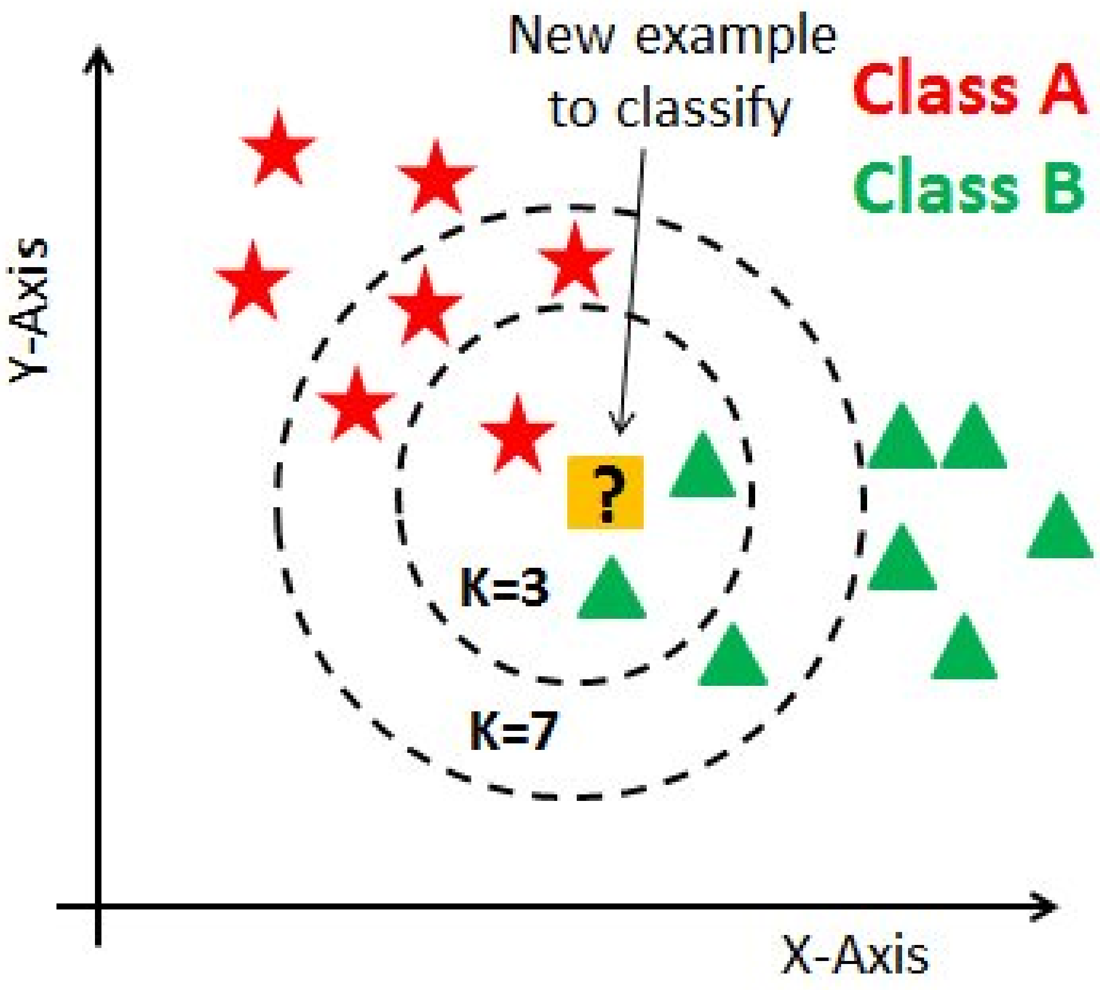
Classification Model

The Classification Model used:-

- Knn
- Naïve Bayes
- Decision Tree
- Random Forest

These classifiers were optimized using hyper parameters that were selected using GridSearchCV

KNN



KNN

```
knn = KNeighborsClassifier()
paramKnn = [{'n_neighbors':[1,4,5,6,8,10], 'algorithm':['ball_tree', 'kd_tree', 'brute', 'auto'], 'leaf_size':[10, 30, 50], 'metric':['minkowski', 'euclidean']}]
gridKnn = GridSearchCV(estimator=knn, param_grid=paramKnn, cv=10, scoring='accuracy', return_train_score=True)
gridKnn.fit(X_train, Y_train)
best_estimators.append({'estimator':gridKnn.best_estimator_, 'accuracy':gridKnn.best_score_, 'param':gridKnn.best_params_, 'test_score':gridKnn.cv_results_})
results.append(gridKnn.cv_results_)
knn=gridKnn.best_estimator_
knn_accuracy = cross_val_score(estimator=knn, X=X_train, y=Y_train, cv=10)
knn_y_pred = knn.predict(X_test)
knn_cm = confusion_matrix(Y_test, knn_y_pred)
```

Accuracy:- 84.6%

Best Parameters:- {'algorithm': 'ball_tree‘}, {'leaf_size': 30},

{'metric': 'minkowski‘}, {'n_neighbors': 1}

Confusion Matrix:-

```
[29,  0,  0,  0,  0],
[ 2,  5,  2,  0,  0],
[ 0,  1,  7,  0,  0],
[ 0,  1,  1,  6,  0],
[ 0,  0,  1,  1,  1]]
```

NAÏVE BAYES

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

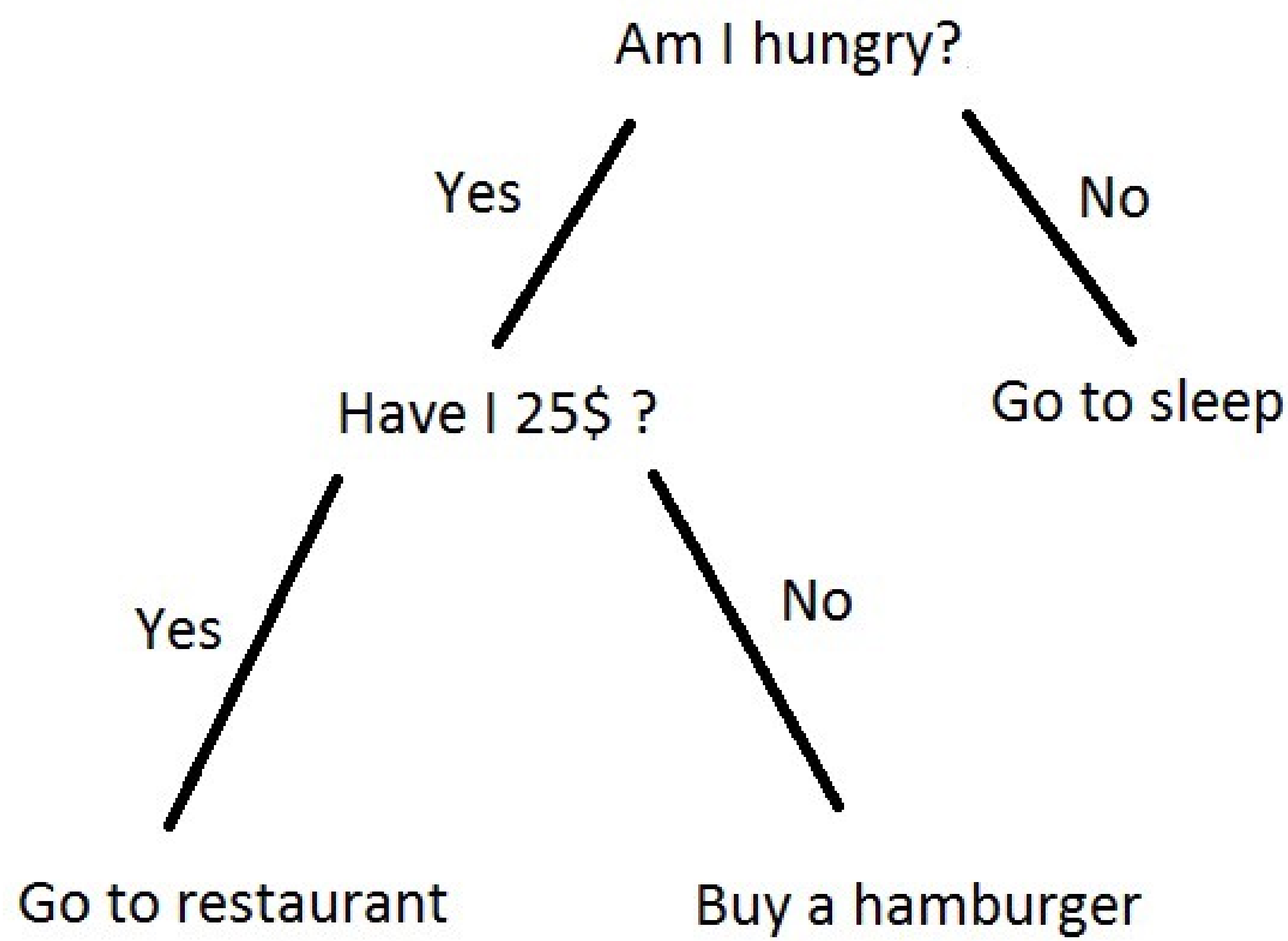
$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

NAÏVE BAYES

```
nb = GaussianNB()
paramNb = [{}]
gridNb = GridSearchCV(estimator=nb, param_grid=paramNb, cv=10,
scoring='accuracy', return_train_score=True)
gridNb.fit(X_train, Y_train)
best_estimators.append({'estimator':gridNb.best_estimator_,
'accuracy':gridNb.best_score_, 'param':gridNb.best_params_,
'test_score':gridNb.cv_results_})
results.append(gridNb.cv_results_)
nb=gridNb.best_estimator_
nb_accuracy = cross_val_score(estimator=nb, X=X_train, y=Y_train, cv=10)
nb_y_pred = knn.predict(X_test)
nb_cm = confusion_matrix(Y_test, nb_y_pred)
```

```
Accuracy:- 84.62%

Confusion Metrics

[29,  0,  0,  0,  0]

[ 0,  9,  0,  0,  0]

[ 0,  3,  4,  1,  0]

[ 0,  0,  1,  7,  0]

[ 0,  0,  0,  0,  3]
```

DECISION TREES

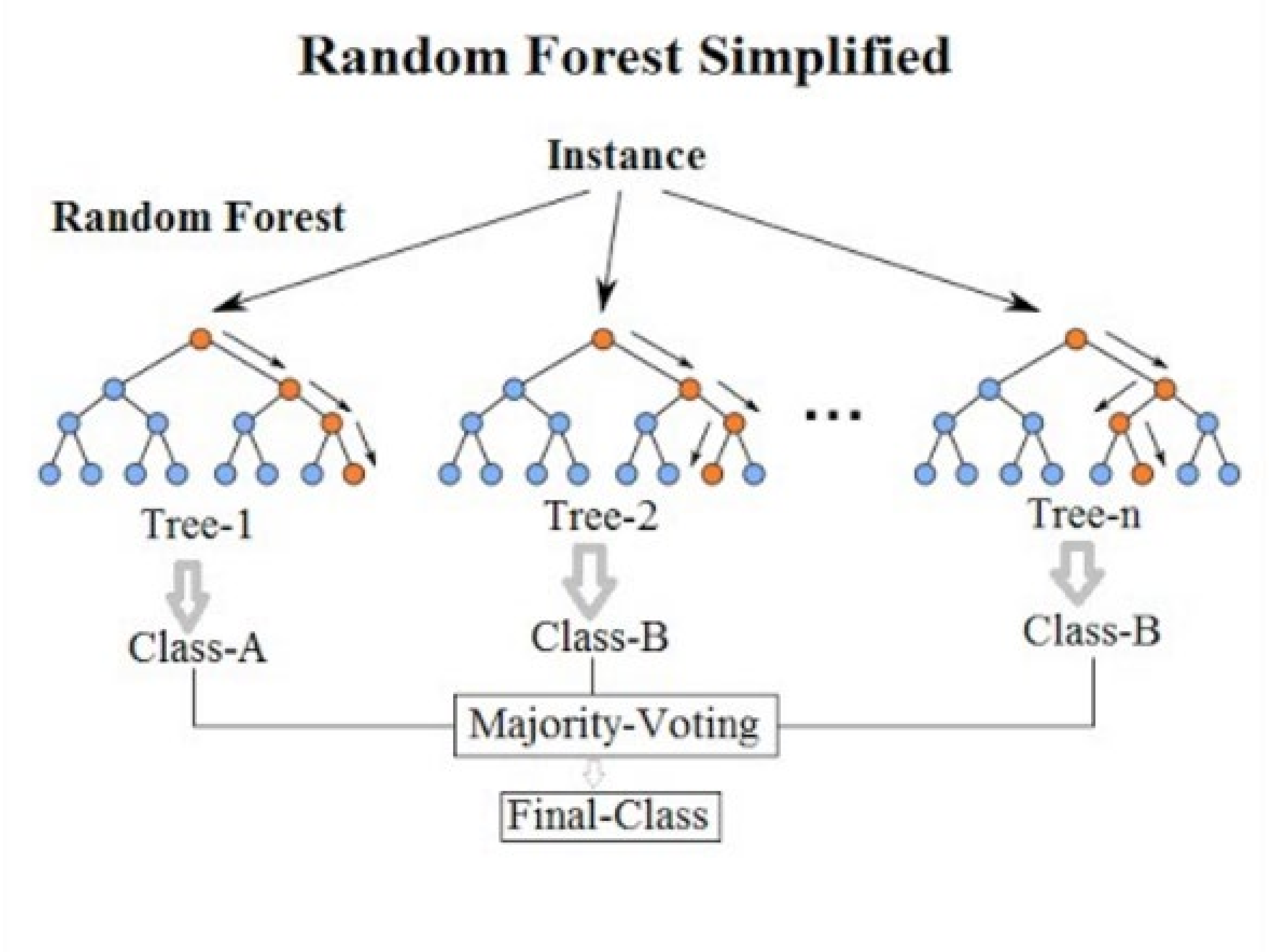


DECISION TREES

```
dt = DecisionTreeClassifier()
paramDt = [{'criterion':['gini', 'entropy'], 'splitter':['best', 'random']}]
gridDt = GridSearchCV(estimator=dt, param_grid=paramDt, cv=10,
scoring='accuracy', return_train_score=True)
gridDt.fit(X_train, Y_train)
best_estimators.append({'estimator':gridDt.best_estimator_,
'accuracy':gridDt.best_score_, 'param':gridDt.best_params_,
'test_score':gridDt.cv_results_})
results.append(gridDt.cv_results_)
dt=gridDt.best_estimator_
dt_accuracy = cross_val_score(estimator=dt, X=X_train, y=Y_train, cv=10)
dt_y_pred = dt.predict(X_test)
dt_cm = confusion_matrix(Y_test, dt_y_pred)
```

```
Accuracy:- 93.77%

Best Parameters:- {'criterion': 'entropy', 'splitter': 'best'}

Confusion Metrics

[29,  0,  0,  0,  0],

[ 0,  9,  0,  0,  0],

[ 0,  0,  8,  0,  0],

[ 0,  0,  0,  8,  0],

[ 0,  0,  0,  1,  2]
```

RANDOM FOREST

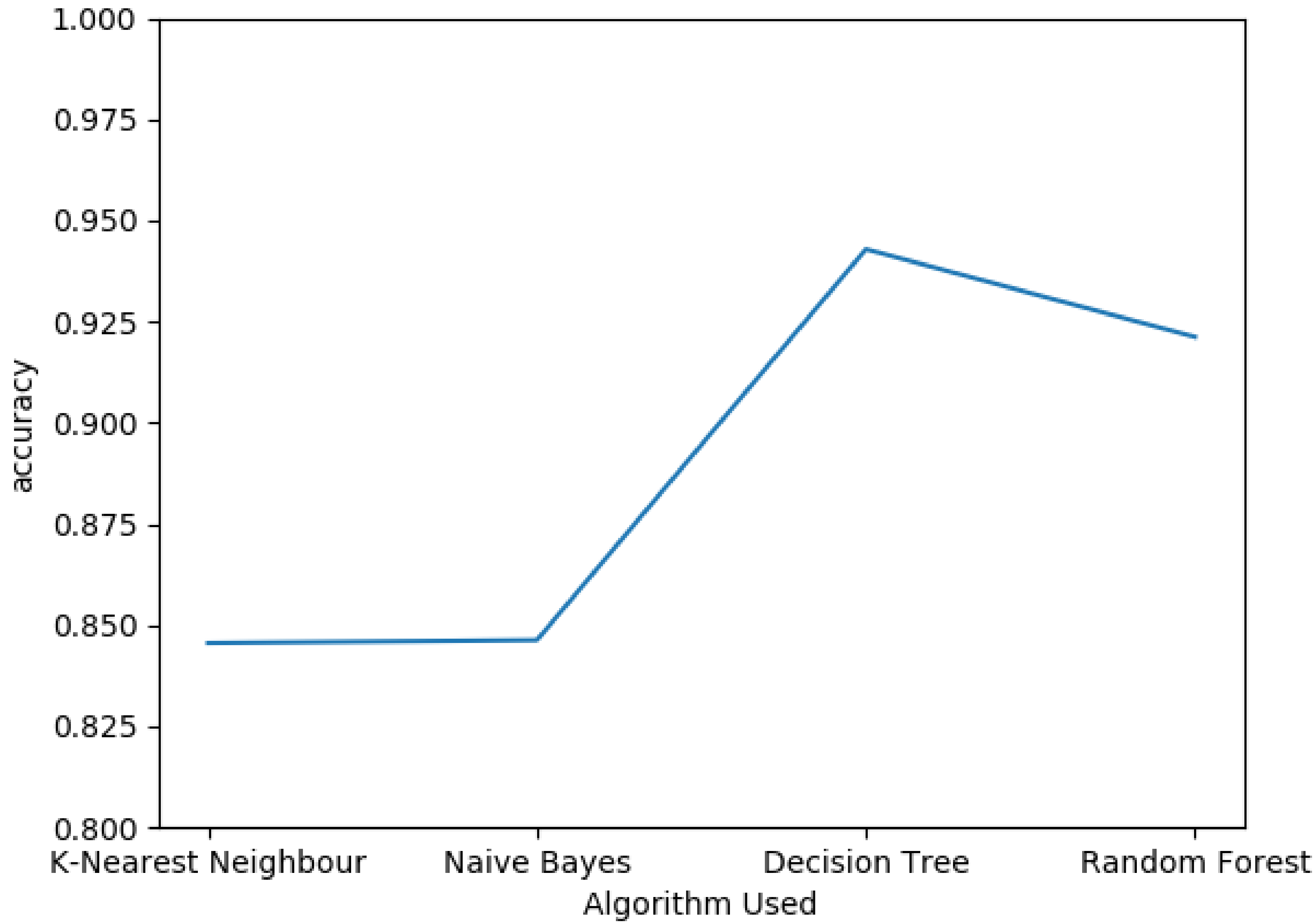


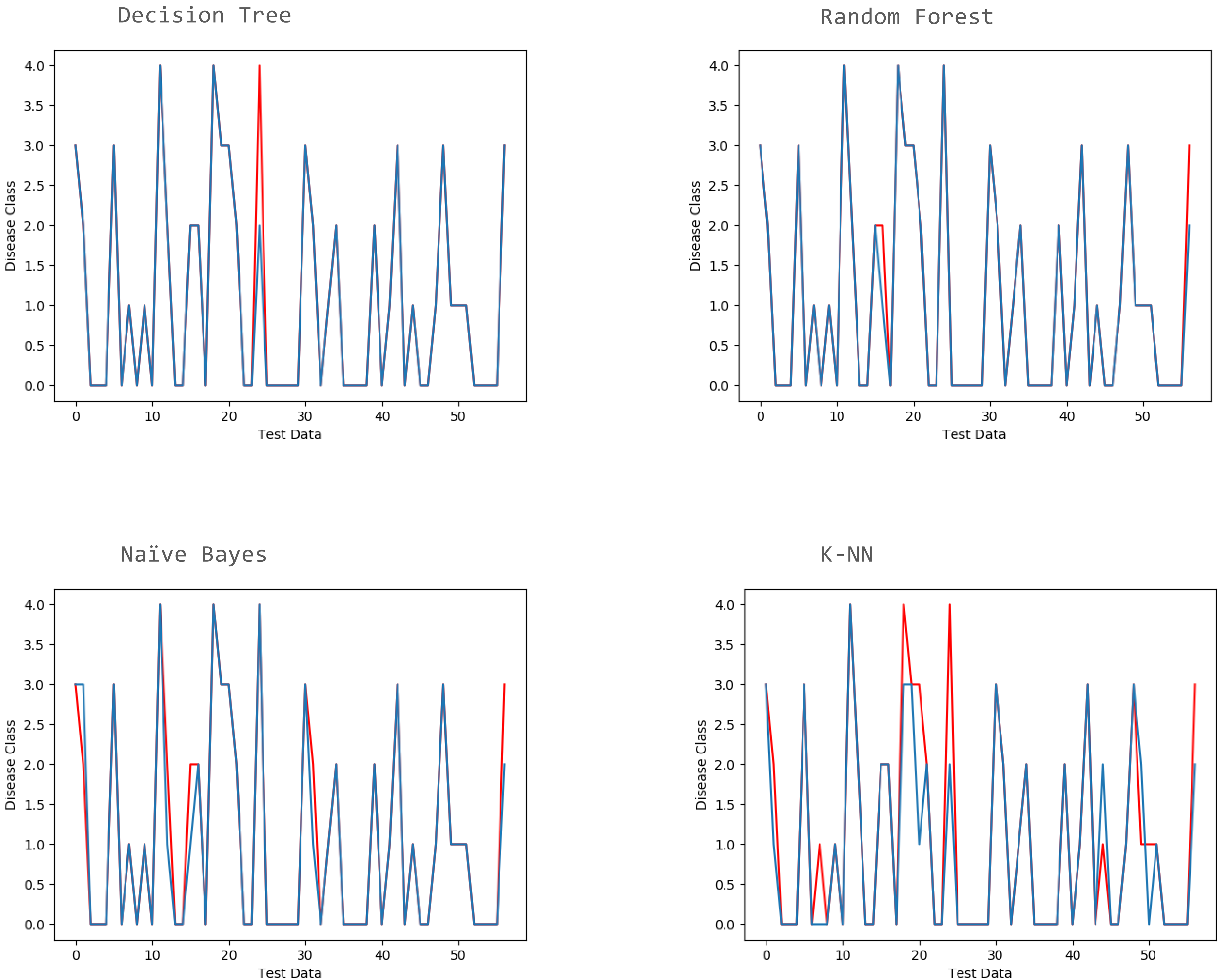
RANDOM FOREST

```
dt = DecisionTreeClassifier()
paramDt = [{'criterion':['gini', 'entropy'], 'splitter':['best', 'random']}]
gridDt = GridSearchCV(estimator=dt, param_grid=paramDt, cv=10,
scoring='accuracy', return_train_score=True)
gridDt.fit(X_train, Y_train)
best_estimators.append({'estimator':gridDt.best_estimator_,
'accuracy':gridDt.best_score_, 'param':gridDt.best_params_,
'test_score':gridDt.cv_results_})
results.append(gridDt.cv_results_)
dt=gridDt.best_estimator_
dt_accuracy = cross_val_score(estimator=dt, X=X_train, y=Y_train, cv=10)
dt_y_pred = dt.predict(X_test)
dt_cm = confusion_matrix(Y_test, dt_y_pred)
```

```
Accuracy:- 90.74%

Best Parameters:- {'criterion': 'entropy', 'n_estimators': 10}

Confusion Metrics

[29,  0,  0,  0,  0],

[ 0,  9,  0,  0,  0],

[ 0,  2,  6,  0,  0],

[ 0,  0,  0,  8,  0],

[ 0,  0,  0,  1,  2]
```

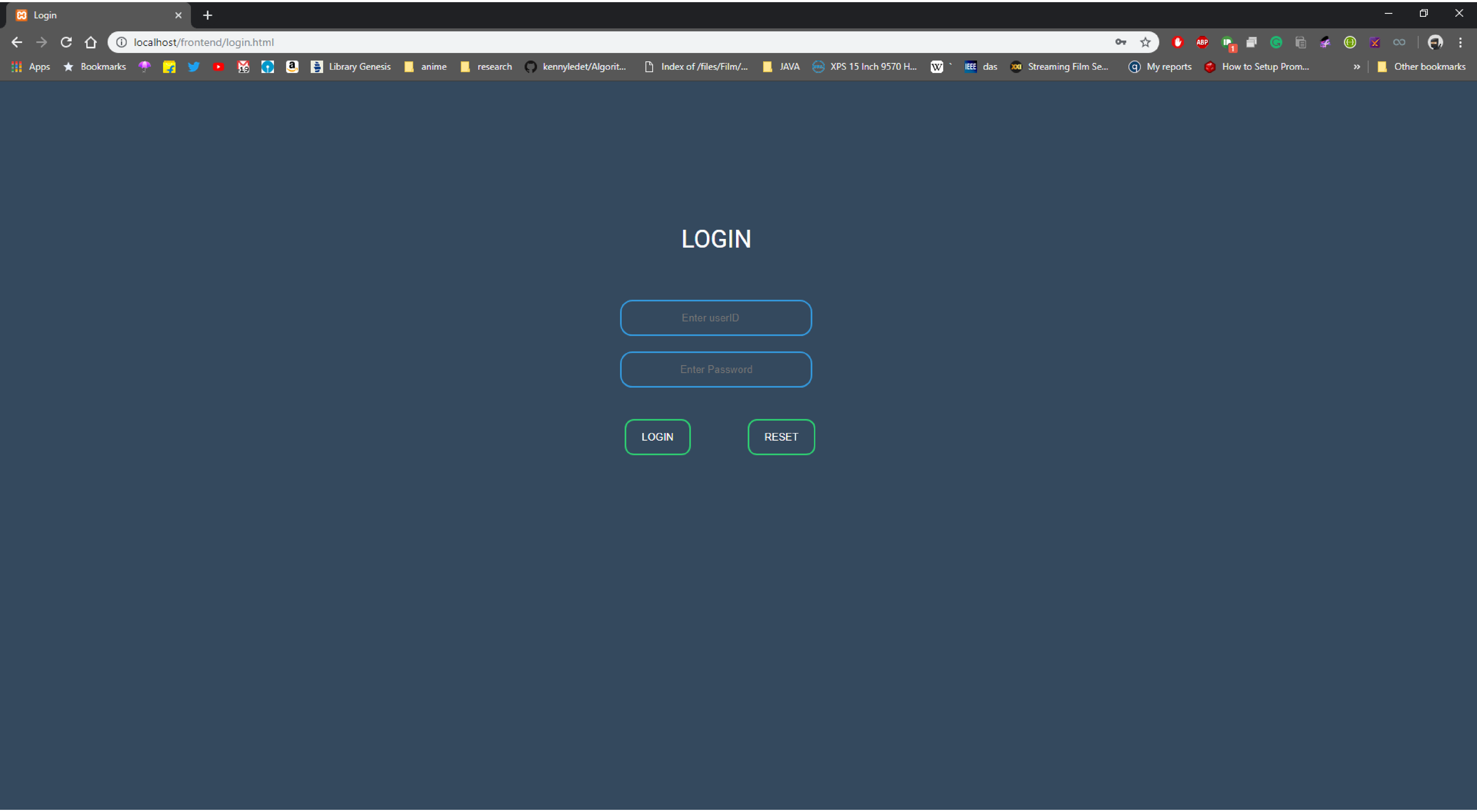
ACCURACY VS MODEL

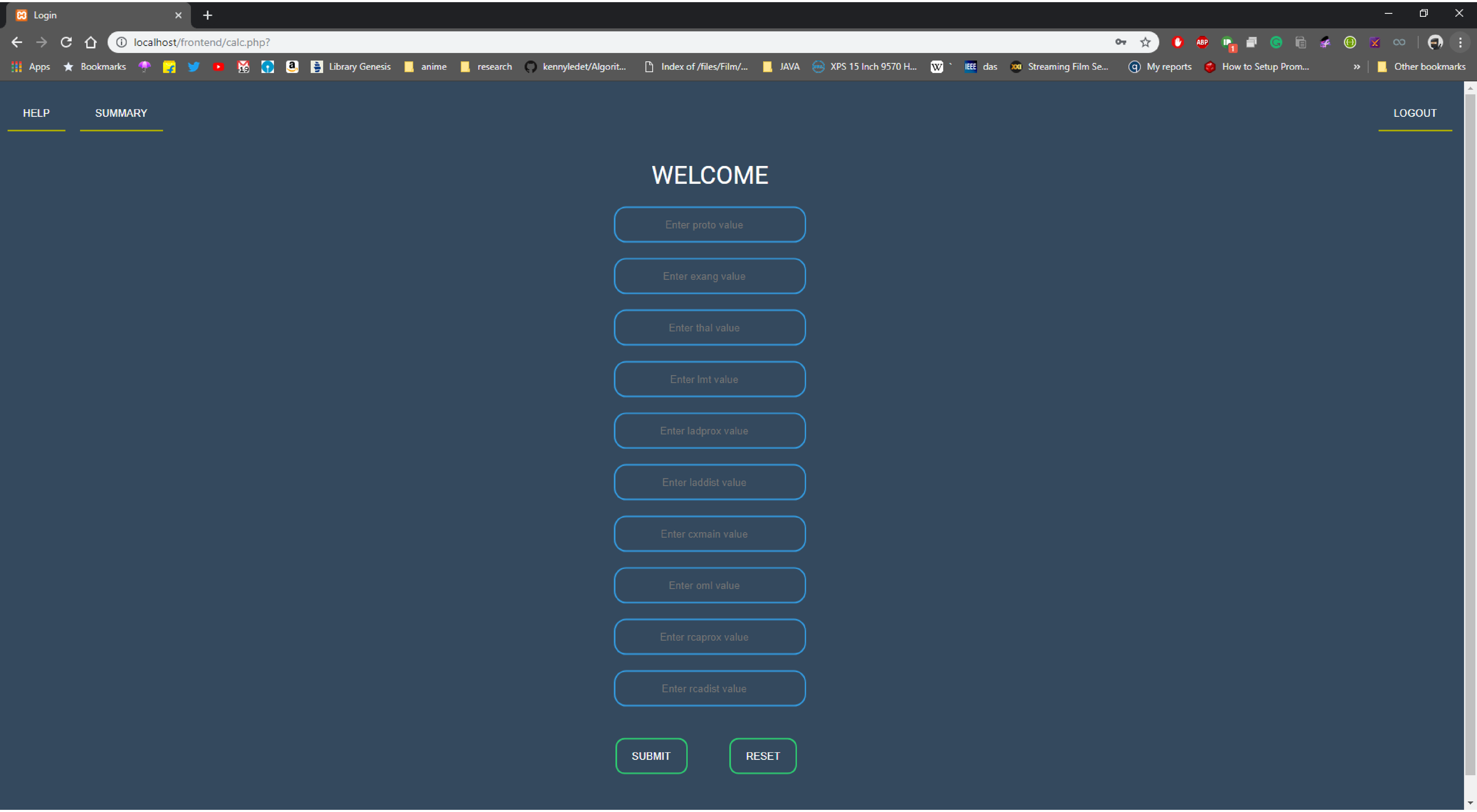




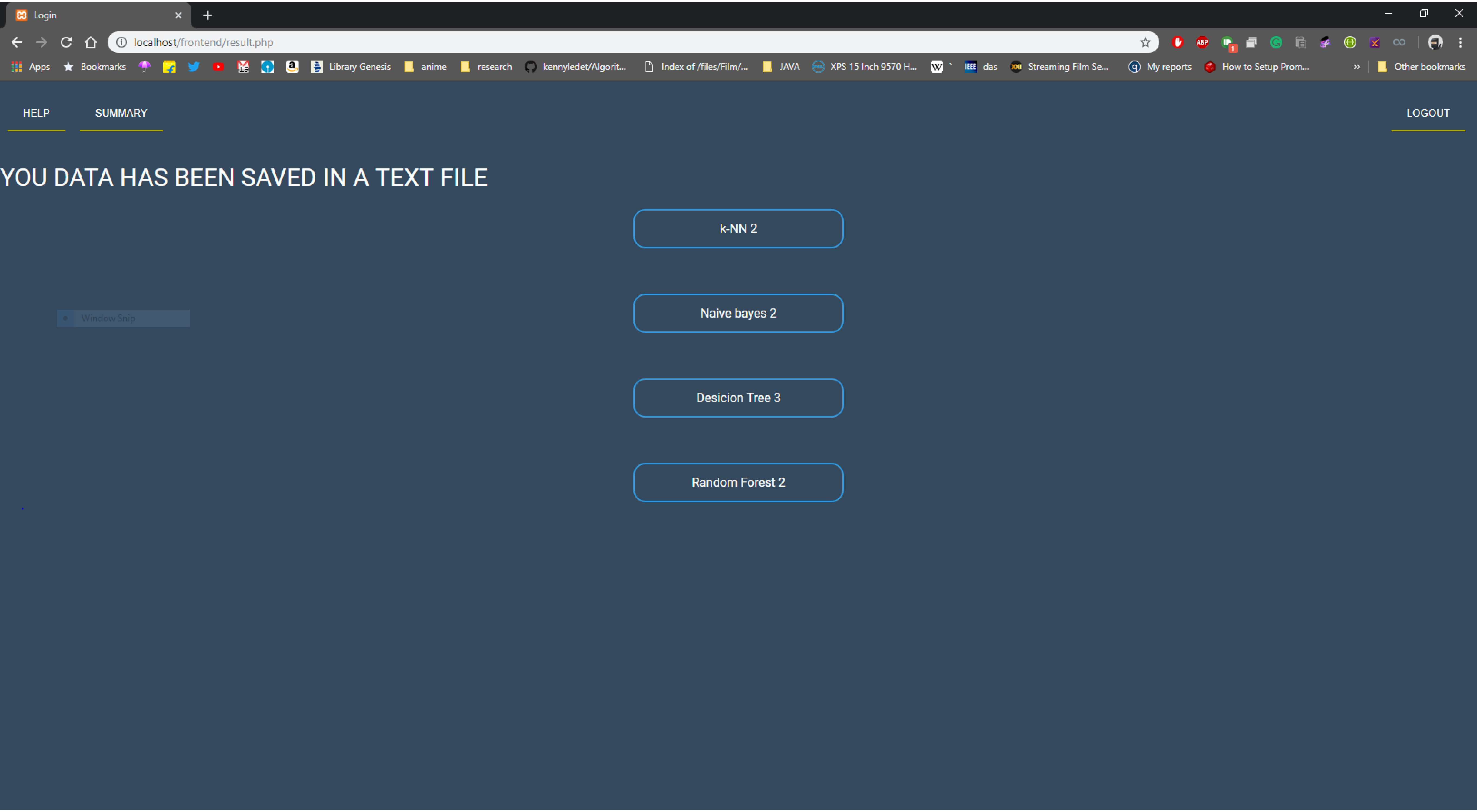
Decision Tree
Random Forest
Naïve Bayes
K-NN
Disease Class
Test Data
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
10
20
30
40
50

Sample ScreenShots


Login
localhost/frontend/login.html
Apps
Bookmarks
Library Genesis
anime
research
kennyledet/Algorit...
Index of /files/Film/...
JAVA
XPS 15 Inch 9570 H...
das
Streaming Film Se...
My reports
How to Setup Prom...
Other bookmarks
LOGIN
Enter userID
Enter Password
LOGIN
RESET



Login
localhost/frontend/calc.php?
Apps
Bookmarks
Library Genesis
anime
research
kennyledet/Algorit...
Index of /files/Film/...
JAVA
XPS 15 Inch 9570 H...
das
Streaming Film Se...
My reports
How to Setup Prom...
Other bookmarks
HELP
SUMMARY
LOGOUT
WELCOME
Enter proto value
Enter exang value
Enter thal value
Enter lmt value
Enter ladprox value
Enter laddist value
Enter cxmain value
Enter oml value
Enter rcaprox value
Enter rcadist value
SUBMIT
RESET


Login
localhost/frontend/result.php
Apps
Bookmarks
Library Genesis
anime
research
kennyledet/Algorit...
Index of /files/Film/...
JAVA
XPS 15 Inch 9570 H...
das
Streaming Film Se...
My reports
How to Setup Prom...
Other bookmarks
HELP
SUMMARY
LOGOUT
YOU DATA HAS BEEN SAVED IN A TEXT FILE
k-NN 2
Naive bayes 2
Desicion Tree 3
Random Forest 2

REFERENCES

- Abhishek Rairikar, Vedant Kulkarni, Vikas Sabale, Harshavardhan Kale, HEART DISEASE PREDICTION USING DATA MINING TECHNIQUES, 2017.
- Alberto Palacios Pawlovsky, An Ensemble Based on Distances for a kNN Method for Heart Disease Diagnosis, 2017
- Balasaheb Tarle, An Artificial Neural Network Based Pattern Classification Algorithm for Diagnosis of Heart Disease, 2017.
- Rashmi G Saboji, A Scalable Solution for Heart Disease Prediction using Classification Mining Technique, 2017
- FEN MIAO1,2, YUN-PENG CAI1,2, YU-XIAO ZHANG3 , XIAO-MAO FAN1,2, AND YE LI, Predictive Modeling of Hospital Mortality for Patients With Heart Failure by Using an Improved Random Survival Forest, 2018
- Priyal Chotwani, Vikas Deep, Asmita Tiwari, Purushottam Sharma, Heart Disease Prediction System Using CART-C, 2018

PUBLICATION

Accepted for Conference in ICIoT 2019

Dear author,

Thank you for sending us your article and giving us the chance to consider your work. We are pleased to inform that your paper has been accepted for oral presentation in the conference.

Manuscript Id: ICIOT014
Title: "A Study on Improved Prediction System for Coronary Heart Disorder"
Recommendation: Accepted for oral presentation in the conference
Plagiarism :21%

We will communicate later regarding publication in Journals.

Kindly send your revised manuscript after reducing the plagiarism percentage and complete the registration process as early as possible in the link http://srmiciot.ml/

Kindly contact us for any queries.

Thanks and Regards,

Organising Committee-ICIoT2019,
Dept of CSE,
Faculty of Engg & Tech,
SRMIST,Kattankulathur.

“

THANKYOU