

Nikit Gokhe
Class: TY Comp D1
Roll No. 324022
Gr No. 21810522

Assignment 4

Problem Statement:

Build a Data model in Python using any classification model (Decision Tree or Naïve Bayes)and infer the result using accuracy score.

Compare different classification models (not limited to NB and DT only) with respect to feature selection and accuracy. Infer the result : which model best suits the dataset chosen .

Objectives:

- 1) Compare different classification models
- 2) Decide which model best suits for the dataset

Theory: Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Decision trees can help organizations structure and automate (complex) information. Decision trees are decision models that answer a specific question based on a question structure and certain conditions.

Dataset:

This dataset obtained from Kaggle.com

Link: <https://www.kaggle.com/volodymyrgavrysh/heart-disease>

Expected Output/sample code:

The screenshot shows a Google Colab notebook interface. The browser tabs at the top include WhatsApp and several assignment notebooks. The notebook title is "324022_Assignment04.ipynb". The left sidebar shows a file explorer with a folder named "sample_data" containing a file "heart.csv". The main code cell contains the following Python code:

```
[2] import pandas as pd
import numpy as np
heart = pd.read_csv('/content/heart.csv')
heart
```

The output of the code is a preview of the "heart" DataFrame, showing 303 rows and 14 columns. The columns are: age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, and target. The preview shows the first 5 rows, followed by an ellipsis, and then rows 298, 299, 300, 301, and 302.

	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 x 14 columns

The next code cell contains the command:

```
[3] heart.describe()
```

The output of this command is a summary statistics table for the "heart" DataFrame:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.398340	0.729373



324022_Assignment04.ipynb ☆

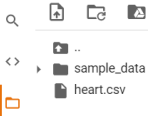
File Edit View Insert Runtime Tools Help All changes saved

Comment

Share



Files



+ Code + Text

RAM Disk

Editing

[3] heart.describe()

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

[4] heart.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
```

Disk 75.86 GB available



324022_Assignment04.ipynb ☆

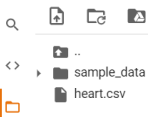
File Edit View Insert Runtime Tools Help All changes saved

Comment

Share



Files



+ Code + Text

RAM Disk

Editing

[5] import seaborn as sns
import matplotlib.pyplot as plt

[6] X = heart.drop('target', axis = 'columns')
y = heart.target
len(X), len(y)

(303, 303)

[7] 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)
len(X_train), len(X_test), len(y_train), len(y_test)

(242, 61, 242, 61)

[8] #USING DECISION TREE CLASSIFIER

[9] from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
predict_y = model.predict(X_test)

[16] from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import roc_auc_score

print("Decision Tree Classifier Model Score is:", accuracy_score(y_test, predict_y))
print("Confusion Matrix is:\n", confusion_matrix(y_test, predict_y))
print("Precision Score is:", precision_score(y_test, predict_y))

Decision Tree Classifier Model Score is: 0.704918032786853

Disk 75.86 GB available

colab.research.google.com/drive/1KZSI4UnBW4vOjSCWIN77TLk8s-PYdqC#scrollTo=ba6bUw4W2Ghq

324022_Assignment04.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- heart.csv

```
[8] #USING DECISION TREE CLASSIFIER

[9] from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
predict_y = model.predict(X_test)

[16] from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score, confusion_matrix
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import roc_auc_score

print("Decision Tree Classifier Model Score is:", accuracy_score(y_test, predict_y))
print("Confusion Matrix is:\n", confusion_matrix(y_test, predict_y))
print("Precision Score is:", precision_score(y_test, predict_y))

Decision Tree Classifier Model Score is: 0.7049180327868853
Confusion Matrix is:
[[22  9]
 [ 9 21]]
Precision Score is: 0.7

[11] #USING NAIVE_BAYES CLASSIFIER MODEL

[12] from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
predict_N = model.predict(X_test)

print("Naive Bayes Classifier Model Score is:", accuracy_score(y_test, predict_N))
print("Confusion Matrix is:\n", confusion_matrix(y_test, predict_N))
print("Precision Score is:", precision_score(y_test, predict_N))

Naive Bayes Classifier Model Score is: 0.8524590163934426
Confusion Matrix is:
[[25  6]
 [ 3 27]]
Precision Score is: 0.81818181818182
```

Disk 75.86 GB available

colab.research.google.com/drive/1KZSI4UnBW4vOjSCWIN77TLk8s-PYdqC#scrollTo=ba6bUw4W2Ghq

324022_Assignment04.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- heart.csv

```
[11] #USING NAIVE_BAYES CLASSIFIER MODEL

[12] from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
predict_N = model.predict(X_test)

print("Naive Bayes Classifier Model Score is:", accuracy_score(y_test, predict_N))
print("Confusion Matrix is:\n", confusion_matrix(y_test, predict_N))
print("Precision Score is:", precision_score(y_test, predict_N))

Naive Bayes Classifier Model Score is: 0.8524590163934426
Confusion Matrix is:
[[25  6]
 [ 3 27]]
Precision Score is: 0.81818181818182

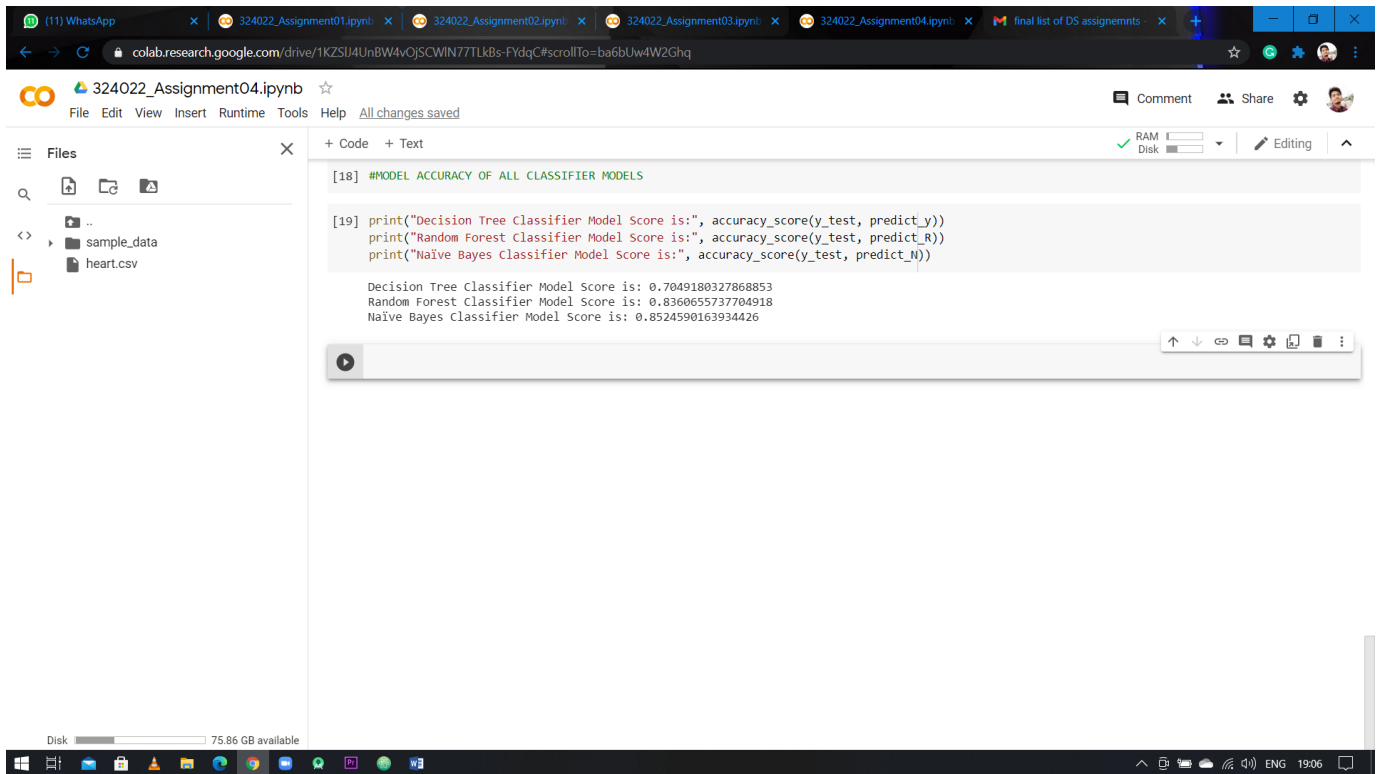
[13] #USING RANDOM FOREST CLASSIFIER

[17] from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
predict_R = model.predict(X_test)

print("Random Forest Classifier Model Score is:", accuracy_score(y_test, predict_R))
print("Confusion Matrix is:\n", confusion_matrix(y_test, predict_R))
print("Precision Score is:", precision_score(y_test, predict_R))

Random Forest Classifier Model Score is: 0.8360655737704918
Confusion Matrix is:
[[24  7]
 [ 3 27]]
Precision Score is: 0.7941176470588235
```

Disk 75.86 GB available



The screenshot shows a Google Colab notebook interface. The browser tabs at the top include WhatsApp and several assignment notebooks. The notebook title is '324022_Assignment04.ipynb'. The left sidebar shows a file explorer with 'sample_data' and 'heart.csv'. The main code cell contains a comment and three print statements for model accuracy scores. The output shows the following results:

```
[18] #MODEL ACCURACY OF ALL CLASSIFIER MODELS

[19] print("Decision Tree Classifier Model Score is:", accuracy_score(y_test, predict_y))
      print("Random Forest Classifier Model Score is:", accuracy_score(y_test, predict_R))
      print("Naive Bayes Classifier Model Score is:", accuracy_score(y_test, predict_N))

Decision Tree Classifier Model Score is: 0.7049180327868853
Random Forest Classifier Model Score is: 0.8360655737704918
Naive Bayes Classifier Model Score is: 0.8524590163934426
```

Inference :

1. Inductive inference to approximate a target function which will produce discrete values.
2. It is widely used, robust to noisy data, and considered a practical method for learning disjunctive expressions. Appropriate Problems for Decision Tree Learning.