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

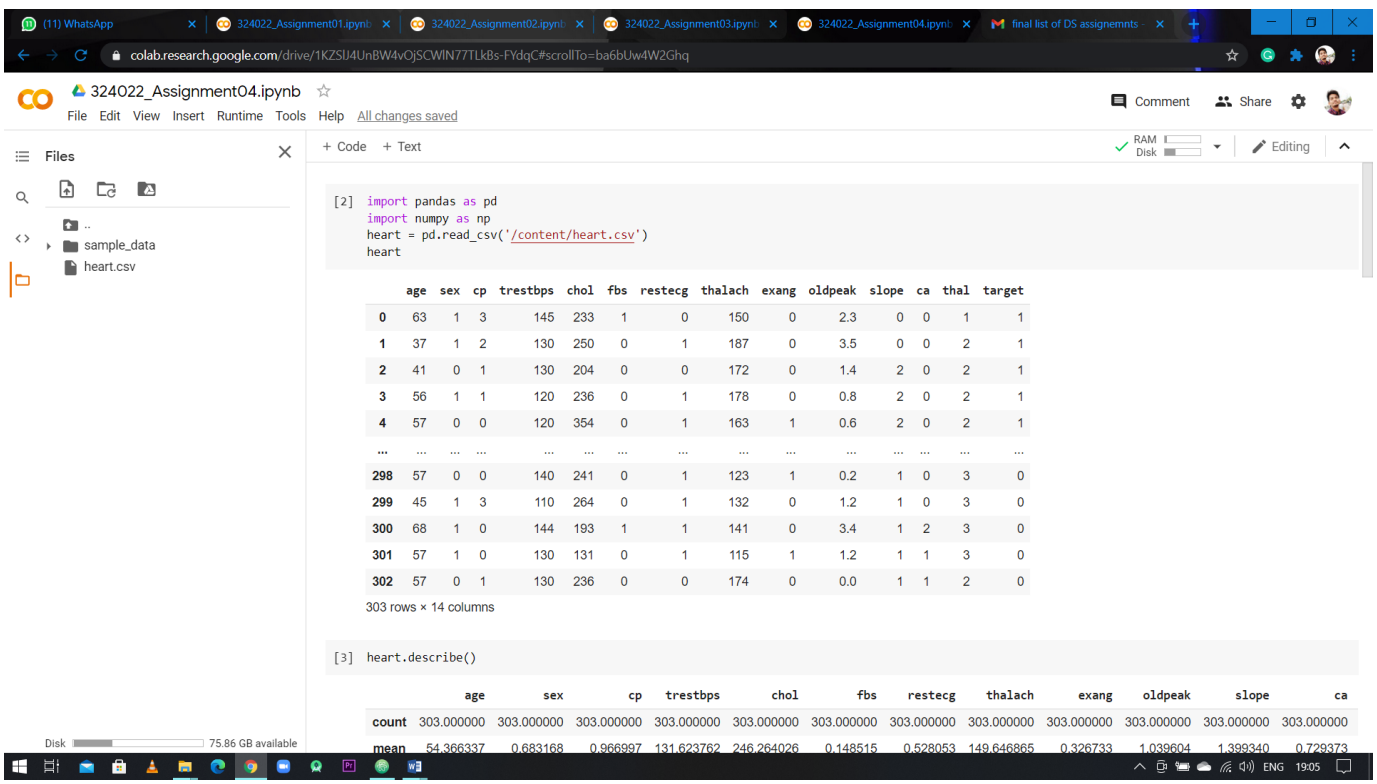
Theory: Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. 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:



(11) WhatsApp | 324022_Assignment01.ipynb | 324022_Assignment02.ipynb | 324022_Assignment03.ipynb | 324022_Assignment04.ipynb | final list of DS assignemnts

colab.research.google.com/drive/1KZSIJ4UnBW4vOjSCWIN77TLkBs-FYdqC#scrollTo=ba6bUw4W2Ghq

324022_Assignment04.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

..
sample_data
heart.csv

+ 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

ENG 19:05

(11) WhatsApp | 324022_Assignment01.ipynb | 324022_Assignment02.ipynb | 324022_Assignment03.ipynb | 324022_Assignment04.ipynb | final list of DS assignemnts

colab.research.google.com/drive/1KZSIJ4UnBW4vOjSCWIN77TLkBs-FYdqC#scrollTo=ba6bUw4W2Ghq

324022_Assignment04.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

..
sample_data
heart.csv

+ 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: 0.7049180327868853
```

Disk 75.86 GB available

ENG 19:05

(11) WhatsApp | 324022_Assignment01.ipynb | 324022_Assignment02.ipynb | 324022_Assignment03.ipynb | 324022_Assignment04.ipynb | final list of DS assignemnts

colab.research.google.com/drive/1KZSIJ4UnBW4vOjSCWIN77TLkBs-FYdqC#scrollTo=ba6bUw4W2Ghq

324022_Assignment04.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

..
sample_data
heart.csv

+ Code + Text

RAM Disk Editing

[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("Naïve 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))
```

Disk 75.86 GB available

ENG 19:06

(11) WhatsApp | 324022_Assignment01.ipynb | 324022_Assignment02.ipynb | 324022_Assignment03.ipynb | 324022_Assignment04.ipynb | final list of DS assignemnts

colab.research.google.com/drive/1KZSIJ4UnBW4vOjSCWIN77TLkBs-FYdqC#scrollTo=ba6bUw4W2Ghq

324022_Assignment04.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

..
sample_data
heart.csv

+ Code + Text

RAM Disk Editing

[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("Naïve 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))
```

```
Naïve Bayes Classifier Model Score is: 0.8524590163934426
Confusion Matrix is:
 [[25  6]
 [ 3 27]]
Precision Score is: 0.8181818181818182
```

[13]
```
#USING RANDOM FOREST CLASSIFIER
```

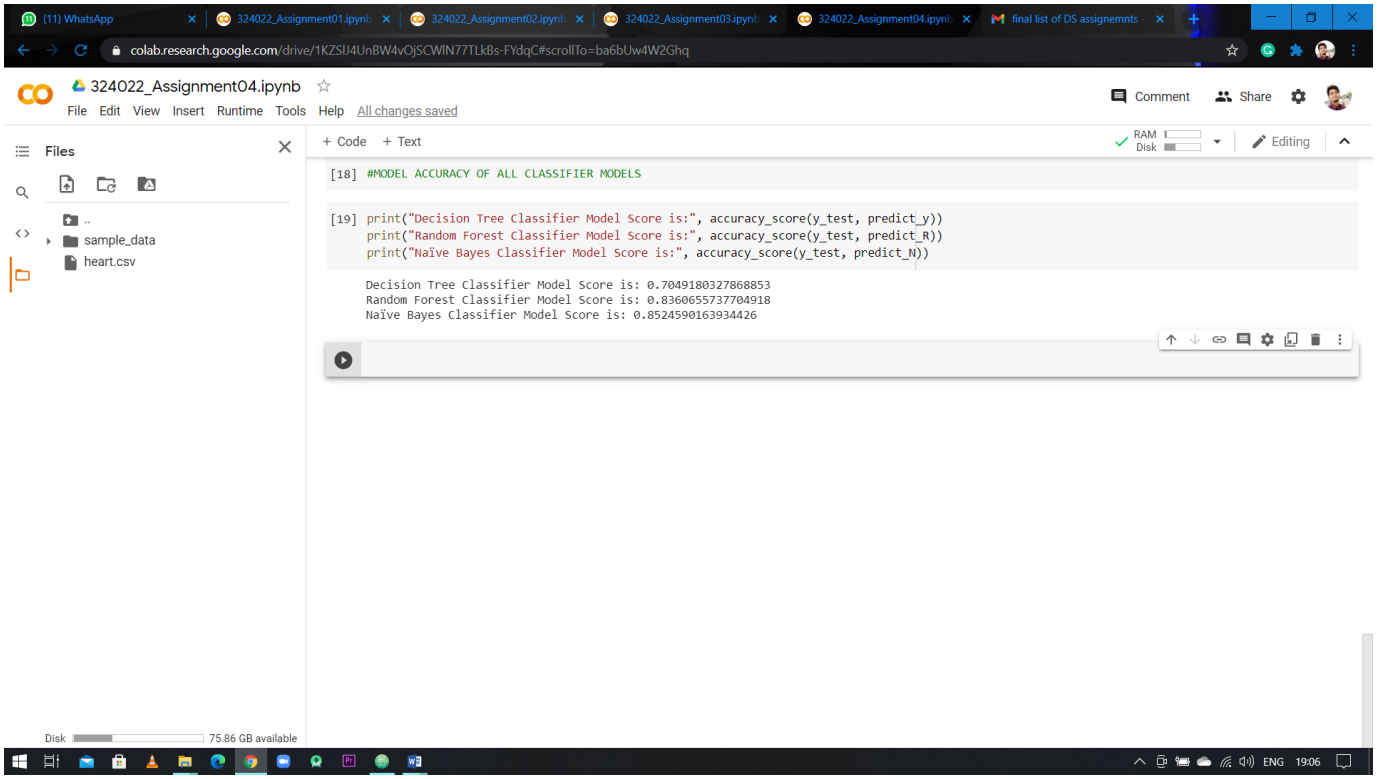
[17]
```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
predict_R = model.predict(X_test)

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

ENG 19:06



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.