**Name: Kartik Jolapara** **SAPID:** 60004200107

**DIV:** B/B1 **Branch:** Computer Engineering

# DMW

**Aim:** Implementation of Classification algorithm Using 1.

Decision Tree ID3 and 2. Naïve Bayes algorithm Perform

the experiment in Python.

Read any dataset from UCI dataset repository

## Theory:

### Decision tree ID3:

In simple words, a decision tree is a structure that contains nodes (rectangular boxes) and edges(arrows) and is built from a dataset (table of columns representing features/attributes and rows corresponds to records). Each node is either used to make a decision (known as decision node) or represent an outcome (known as leaf node).ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes(divides) features into two or more groups at each step.ID3 uses a top-down greedy approach to build a decision tree. In simple words, the top-down approach means that we start building the tree from the top and the greedy approach means that at each iteration we select the best feature at the present moment to create a node.

### Naive Bayes:

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

To start with, let us consider a dataset.

Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies the conditions as fit("Yes") or unfit("No") for playing golf.

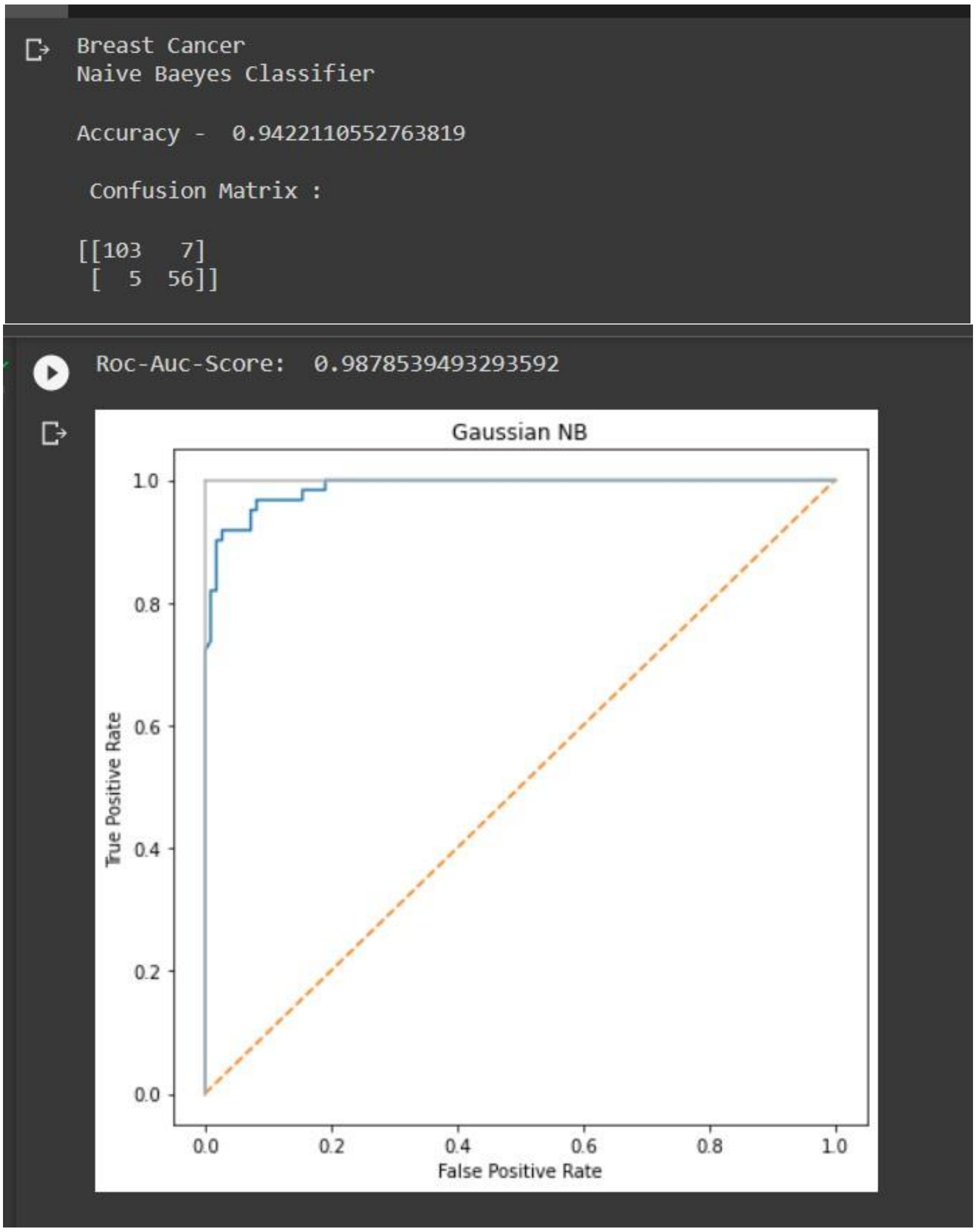
## Dataset – 1 :

```
import pandas as pd import numpy as np from sklearn.model_selection import
train_test_split from sklearn.naive_bayes import GaussianNB from
sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score, accu
racy_score, classification_report import matplotlib.pyplot as plt from
sklearn import tree from sklearn.tree import DecisionTreeClassifier from
sklearn.model_selection import cross_val_score from sklearn.pipeline import
make_pipeline from sklearn.preprocessing import StandardScaler from
sklearn.svm import SVC from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder import seaborn as sns

#Dataframe creation df =
pd.read_csv("/content/data.csv") df =
df.dropna()

# Train test split  y = df.diagnosis x
= df.drop(['diagnosis','id'],axis=1)

X_train, X_val, Y_train, Y_val = train_test_split(x, y, test_size = 0.3, ran
dom_state=255)

# Gaussian model gnb =
GaussianNB()
gnb.fit(X_train, Y_train)
 y_val_pred = gnb.predict(X_val)
y_train_pred = gnb.predict(X_train)
nb_cancer = gnb.score(X_train, Y_train)*100

# Prediction and Confusion Matrix print("Breast
Cancer ") print("Naive Baeyes Classifier")
print(f"\nAccuracy - ", gnb.score(X_train, Y_train))
print(f"\n Confusion Matrix : \n")
con_matrix = confusion_matrix(Y_val,y_val_pred) print(con_matrix)

#Plotting of AUCROC
y_score = gnb.predict_proba(X_val)[ :,1]
```

```
false_positive, true_positive, threshold = roc_curve(Y_val, y_score)
print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score)) print()
plt.subplots(1, figsize = (7, 7)) plt.title("Gaussian NB")
plt.plot(false_positive, true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False Positive Rate")
plt.show()
 dt =
DecisionTreeClassifier()
dt.fit(X_train, Y_train)
 y_val_pred = dt.predict(X_val)
dt_cancer = dt.score(X_val, Y_val)*100

#Prediction and Confusion Matrix
print("\n\nDecission Tree") print(f"\nAccuracy -
", dt.score(X_val, Y_val)) print(f"\n Confusion
Matrix : ") print("\n", confusion_matrix(Y_val,
y_val_pred))

#Plotting of AUCROC y_score =
dt.predict_proba(X_val)[ :,1]
 false_positive, true_positive, threshold = roc_curve(Y_val,
y_score) print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score))
print()  plt.subplots(1, figsize = (7, 7)) plt.title("Gaussian NB")
plt.plot(false_positive, true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False Positive Rate")
plt.show()

# Kfolds cross validation pipeline = make_pipeline(StandardScaler(),
RandomForestClassifier(n_estimato rs=100, max_depth=4)) X = X_train y =
Y_train scores = cross_val_score(pipeline, X, y, cv=10, n_jobs=1)
print('Cross Validation accuracy scores: %s' % scores)
# Random Forest Classifier m
= RandomForestClassifier()
```

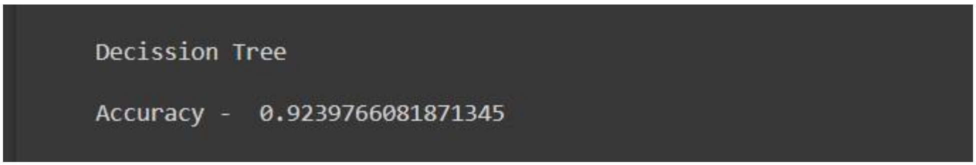
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) #
print(X_train.head(), X_test.head(), y_train.head(), y_test.head())
m.fit(X_train, y_train) print(f"\nAccuracy of Random forest classifier is
{m.score(X_test, y_test)}"
)
```

**Output :-**

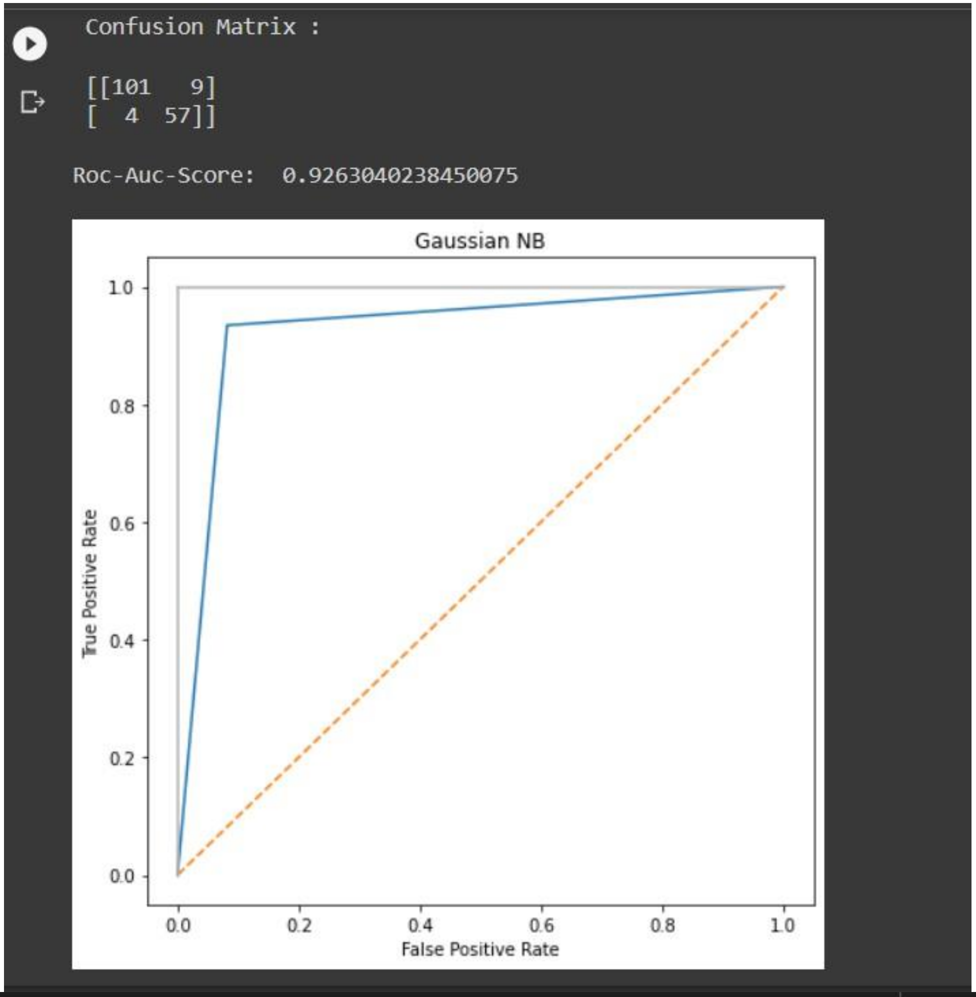


```
Breast Cancer
Naive Baeyes Classifier

Accuracy -  0.9422110552763819

 Confusion Matrix :

[[103   7]
 [  5  56]]
```

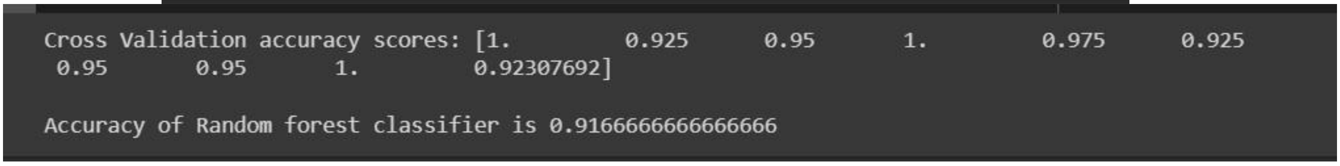
```
Roc-Auc-Score:  0.9878539493293592
```



```
Decission Tree

Accuracy -  0.9239766081871345
```

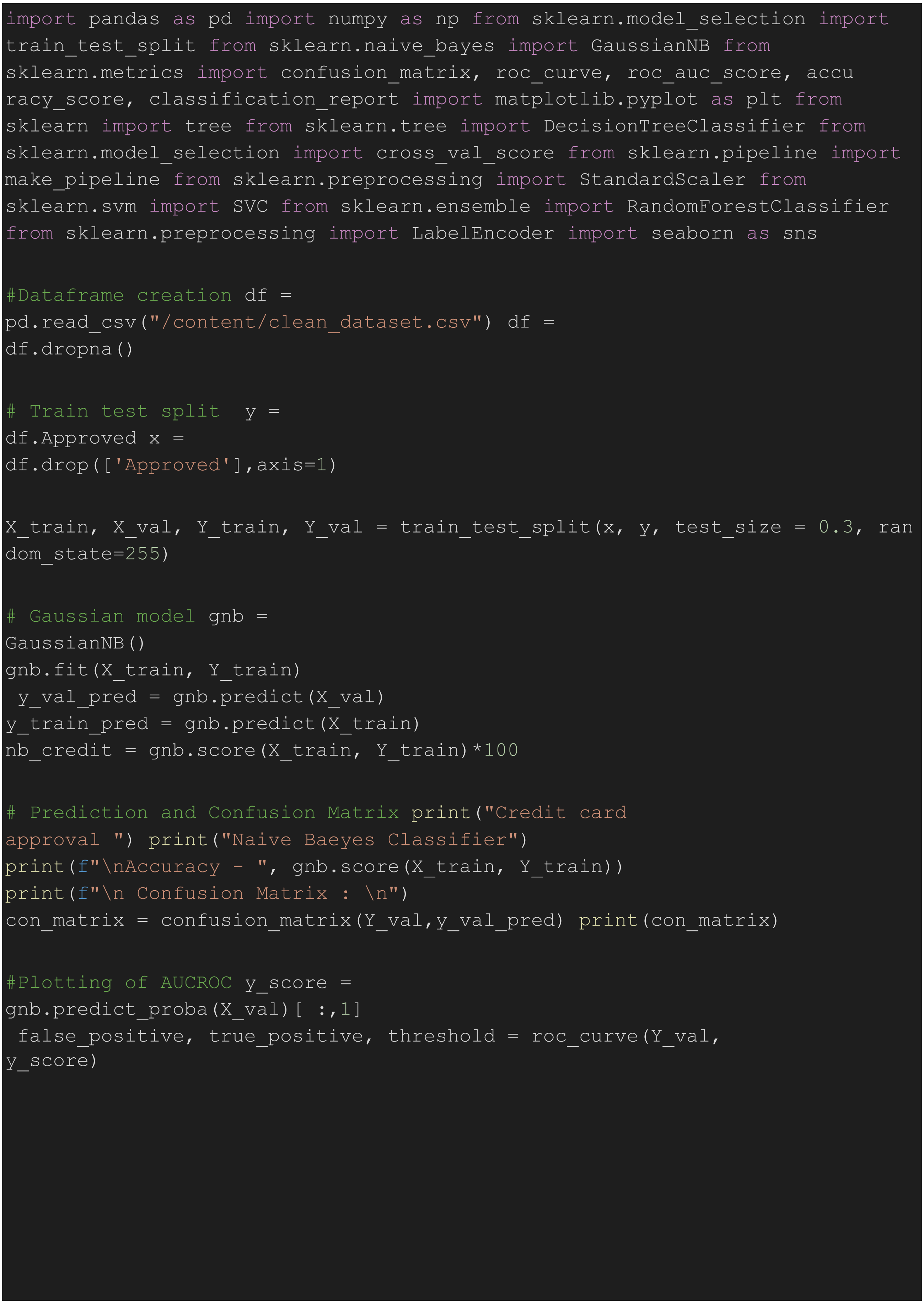


```
Confusion Matrix :

[[101   9]
 [  4  57]]

Roc-Auc-Score:  0.9263040238450075
```



```
Cross Validation accuracy scores: [1.         0.925      0.95       1.         0.975      0.925
 0.95       0.95       1.         0.92307692]

Accuracy of Random forest classifier is 0.9166666666666666
```

**Dataset – 2 :**

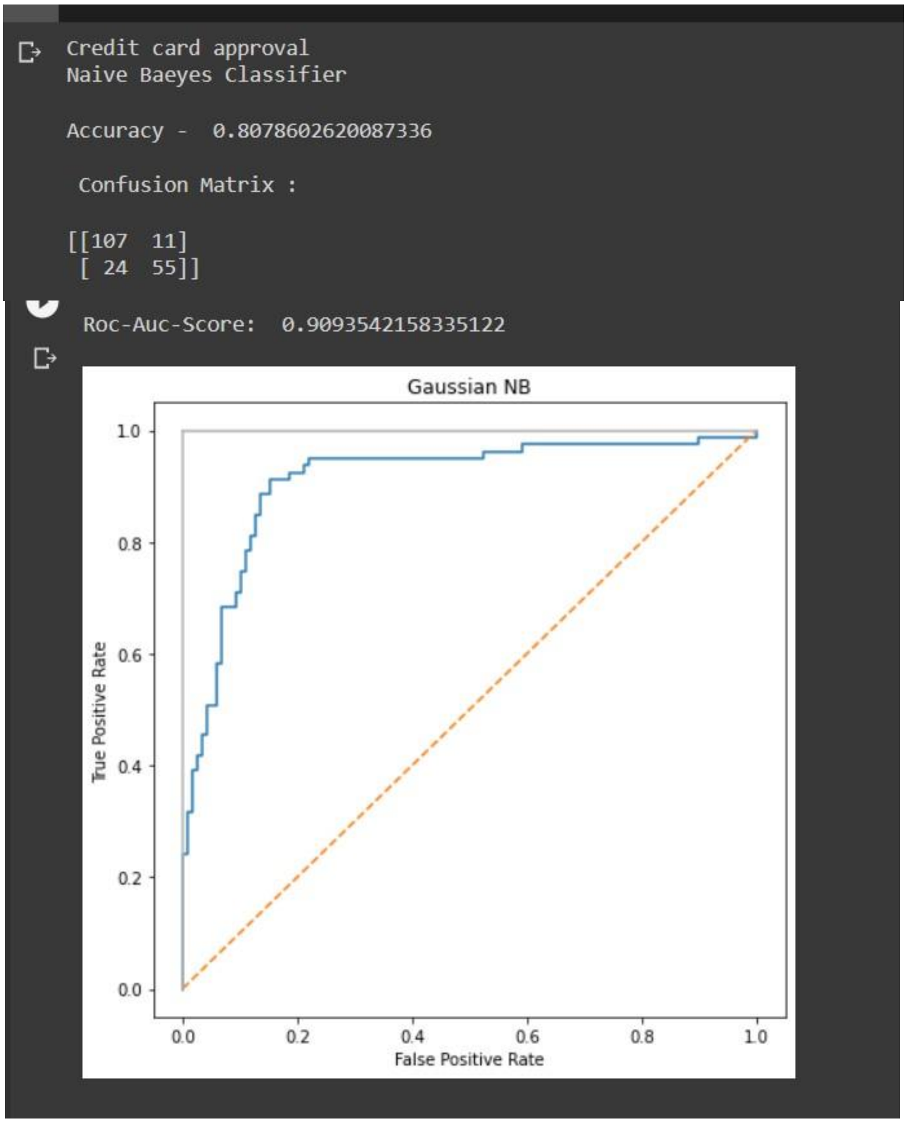


```
import pandas as pd import numpy as np from sklearn.model_selection import
train_test_split from sklearn.naive_bayes import GaussianNB from
sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score, accu
racy_score, classification_report import matplotlib.pyplot as plt from
sklearn import tree from sklearn.tree import DecisionTreeClassifier from
sklearn.model_selection import cross_val_score from sklearn.pipeline import
make_pipeline from sklearn.preprocessing import StandardScaler from
sklearn.svm import SVC from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder import seaborn as sns

#Dataframe creation df =
pd.read_csv("/content/clean_dataset.csv") df =
df.dropna()

# Train test split  y =
df.Approved x =
df.drop(['Approved'],axis=1)

X_train, X_val, Y_train, Y_val = train_test_split(x, y, test_size = 0.3, ran
dom_state=255)

# Gaussian model gnb =
GaussianNB()
gnb.fit(X_train, Y_train)
 y_val_pred = gnb.predict(X_val)
y_train_pred = gnb.predict(X_train)
nb_credit = gnb.score(X_train, Y_train)*100

# Prediction and Confusion Matrix print("Credit card
approval ") print("Naive Baeyes Classifier")
print(f"\nAccuracy - ", gnb.score(X_train, Y_train))
print(f"\n Confusion Matrix : \n")
con_matrix = confusion_matrix(Y_val,y_val_pred) print(con_matrix)

#Plotting of AUCROC y_score =
gnb.predict_proba(X_val)[ :,1]
 false_positive, true_positive, threshold = roc_curve(Y_val,
y_score)
```

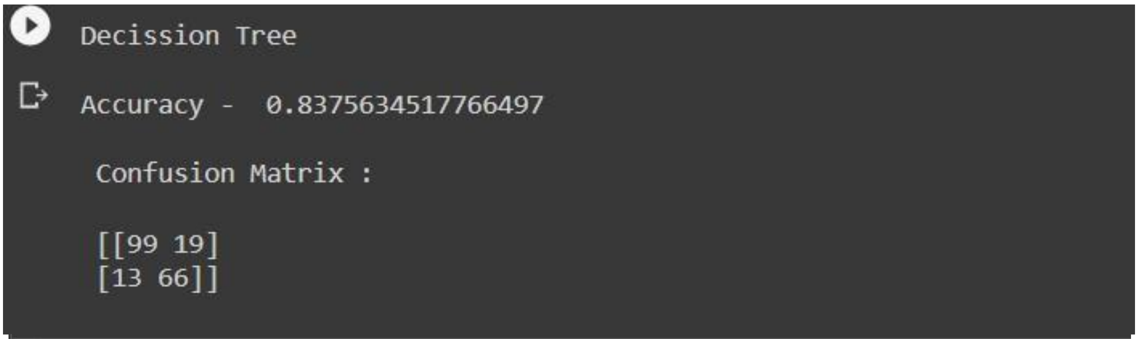
```
print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score))
print()  plt.subplots(1, figsize = (7, 7))
plt.title("Gaussian NB") plt.plot(false_positive,
true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False
Positive Rate") plt.show()
 dt =
DecisionTreeClassifier()
dt.fit(X_train, Y_train)
 y_val_pred = dt.predict(X_val)
dt_credit = dt.score(X_val, Y_val)*100

#Prediction and Confusion Matrix
print("\n\nDecission Tree") print(f"\nAccuracy -
", dt.score(X_val, Y_val)) print(f"\n Confusion
Matrix : ") print("\n", confusion_matrix(Y_val,
y_val_pred))

#Plotting of AUCROC y_score =
dt.predict_proba(X_val)[ :,1]
 false_positive, true_positive, threshold = roc_curve(Y_val,
y_score) print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score))
print()
plt.subplots(1, figsize = (7, 7)) plt.title("Gaussian
NB")
plt.plot(false_positive, true_positive) plt.plot([0,
1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False
Positive Rate") plt.show()

# Kfolds cross validation pipeline = make_pipeline(StandardScaler(),
RandomForestClassifier(n_estimato rs=100, max_depth=4)) X = X_train y =
Y_train scores = cross_val_score(pipeline, X, y, cv=10, n_jobs=1)
print('\n\nCross Validation accuracy scores: %s' % scores)
# Random Forest Classifier m
= RandomForestClassifier()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
# print(X_train.head(), X_test.head(), y_train.head(), y_test.head())
m.fit(X_train, y_train) print(f"\nAccuracy of Random forest classifier is
{m.score(X_test, y_test)}" )
```

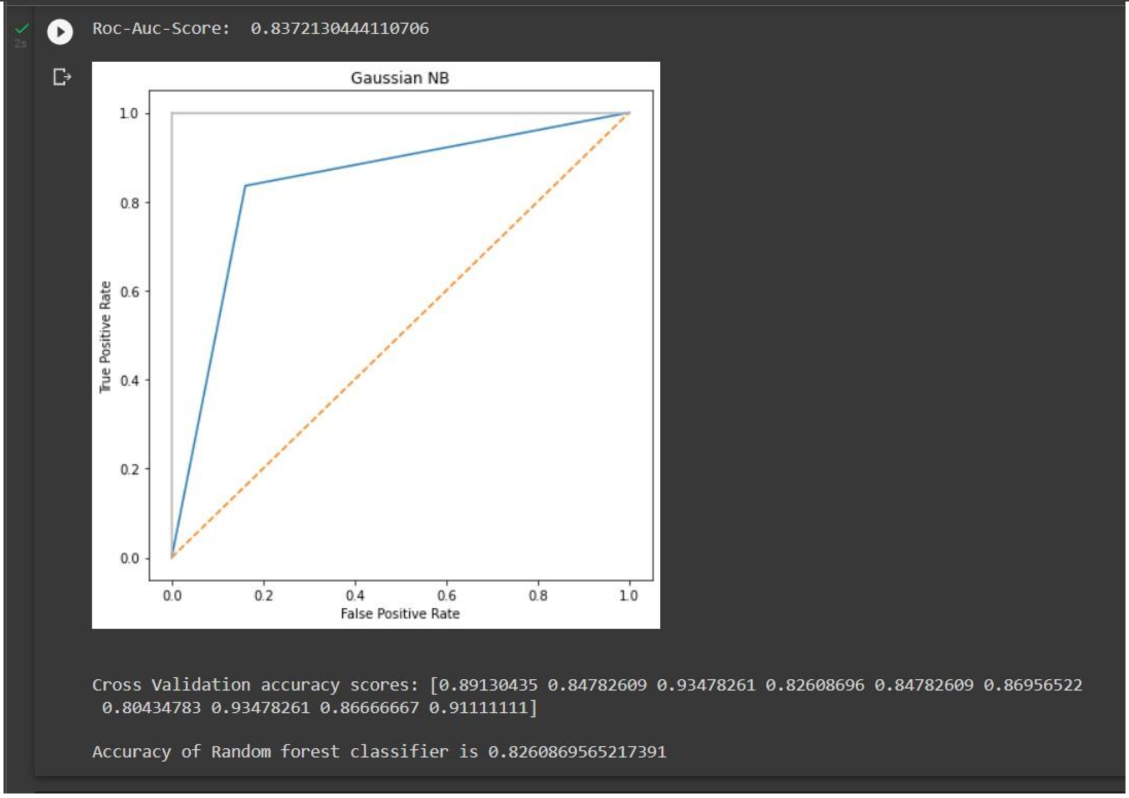
## Output :-



```
Credit card approval
Naive Baeyes Classifier

Accuracy -  0.8078602620087336

 Confusion Matrix :

[[107  11]
 [ 24  55]]

Roc-Auc-Score:  0.9093542158335122
```



```
Decission Tree

Accuracy -  0.8375634517766497

 Confusion Matrix :

 [[99 19]
 [13 66]]
```



```
Roc-Auc-Score:  0.8372130444110706
```

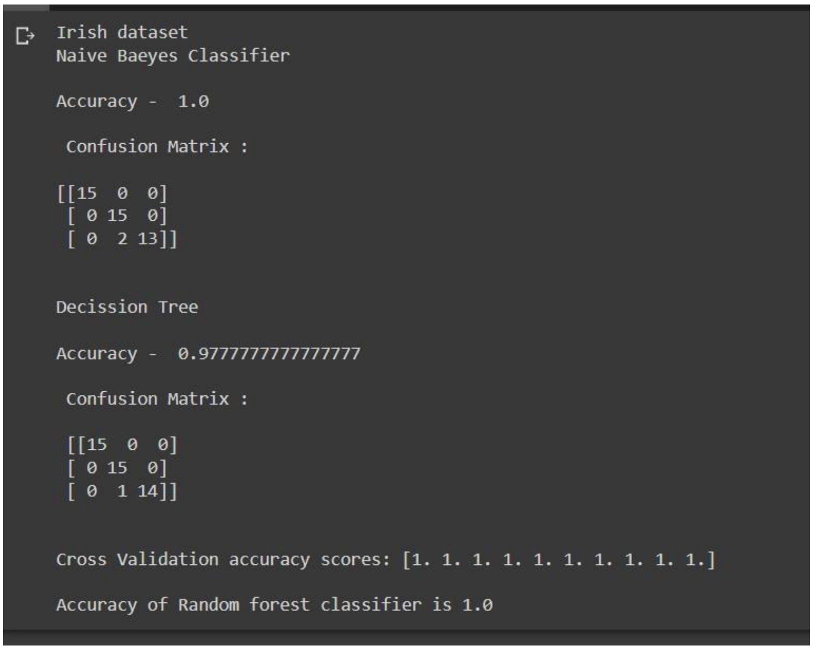
```
Cross Validation accuracy scores: [0.89130435 0.84782609 0.93478261 0.82608696 0.84782609 0.86956522
 0.80434783 0.93478261 0.86666667 0.91111111]

Accuracy of Random forest classifier is 0.8260869565217391
```

**Dataset – 3 :**

```
import pandas as pd import numpy as np from sklearn.model_selection import
train_test_split from sklearn.naive_bayes import GaussianNB from
sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score, accu
racy_score, classification_report import matplotlib.pyplot as plt from
sklearn import tree from sklearn.tree import DecisionTreeClassifier from
sklearn.model_selection import cross_val_score from sklearn.pipeline import
make_pipeline from sklearn.preprocessing import StandardScaler from
sklearn.svm import SVC from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder import seaborn as sns

#Dataframe creation df =
pd.read_csv("/content/Iris (1).csv") df =
df.dropna()

# Train test split  y =
df.Species x =
df.drop(['Species'],axis=1)

X_train, X_val, Y_train, Y_val = train_test_split(x, y, test_size = 0.3, ran
dom_state=255)

# Gaussian model gnb =
GaussianNB()
gnb.fit(X_train, Y_train)
 y_val_pred = gnb.predict(X_val)
y_train_pred = gnb.predict(X_train)
nb_iris = gnb.score(X_train, Y_train)*100

# Prediction and Confusion Matrix print("Audit
risk  ")
print("Naive Baeyes Classifier") print(f"\nAccuracy
- ", gnb.score(X_train, Y_train)) print(f"\n
Confusion Matrix : \n")
con_matrix = confusion_matrix(Y_val,y_val_pred) print(con_matrix)

# Decision Tree dt =
DecisionTreeClassifier()
dt.fit(X_train, Y_train)
```

```
y_val_pred = dt.predict(X_val) dt_iris
= dt.score(X_val, Y_val)*100

#Prediction and Confusion Matrix
print("\n\nDecission Tree") print(f"\nAccuracy -
", dt.score(X_val, Y_val)) print(f"\n Confusion
Matrix : ") print("\n", confusion_matrix(Y_val,
y_val_pred))

# Kfolds cross validation pipeline = make_pipeline(StandardScaler(),
RandomForestClassifier(n_estimato rs=100, max_depth=4)) X = X_train y =
Y_train scores = cross_val_score(pipeline, X, y, cv=10, n_jobs=1)
print('\n\nCross Validation accuracy scores: %s' % scores)
# Random Forest Classifier m
= RandomForestClassifier()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) #
print(X_train.head(), X_test.head(), y_train.head(), y_test.head())
m.fit(X_train, y_train) print(f"\nAccuracy of Random forest classifier is
{m.score(X_test, y_test)}" )
```

## Output :-



```
Irish dataset
Naive Baeyes Classifier

Accuracy -  1.0

 Confusion Matrix :

[[15  0  0]
 [ 0 15  0]
 [ 0  2 13]]


Decission Tree

Accuracy -  0.9777777777777777

 Confusion Matrix :

 [[15  0  0]
 [ 0 15  0]
 [ 0  1 14]]


Cross Validation accuracy scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

Accuracy of Random forest classifier is 1.0
```

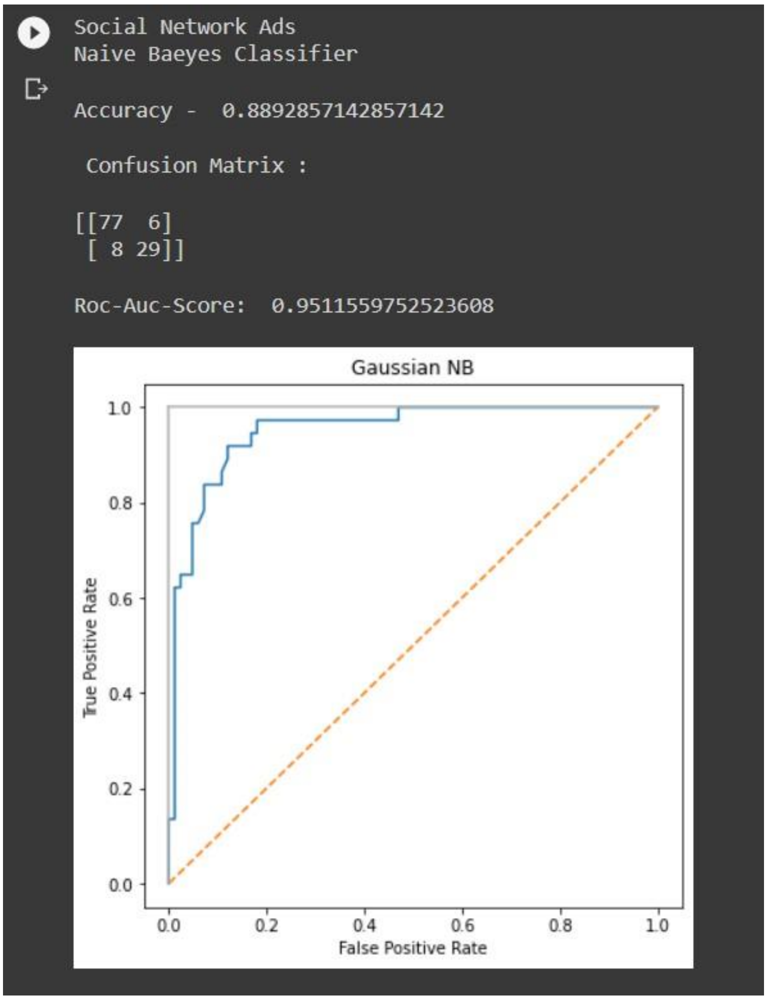
## Dataset – 4 :

```
import pandas as pd import numpy as np from sklearn.model_selection import
train_test_split from sklearn.naive_bayes import GaussianNB from
sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score, accu
racy_score, classification_report import matplotlib.pyplot as plt from
sklearn import tree from sklearn.tree import DecisionTreeClassifier from
sklearn.model_selection import cross_val_score from sklearn.pipeline import
make_pipeline from sklearn.preprocessing import StandardScaler from
sklearn.svm import SVC from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder import seaborn as sns

#Dataframe creation df =
pd.read_csv("/content/Social_Network_Ads.csv") df =
df.dropna()

# Train test split  y =
df.Purchased x =
df.drop(['Purchased'],axis=1)

X_train, X_val, Y_train, Y_val = train_test_split(x, y, test_size = 0.3, ran
dom_state=255)

# Gaussian model gnb =
GaussianNB()
gnb.fit(X_train, Y_train)
 y_val_pred = gnb.predict(X_val)
y_train_pred = gnb.predict(X_train)
nb_social = gnb.score(X_train, Y_train)*100

# Prediction and Confusion Matrix print("Social
Network Ads ") print("Naive Baeyes Classifier")
print(f"\nAccuracy - ", gnb.score(X_train, Y_train))
print(f"\n Confusion Matrix : \n")
con_matrix = confusion_matrix(Y_val,y_val_pred) print(con_matrix)

#Plotting of AUCROC
y_score = gnb.predict_proba(X_val)[ :,1]
```

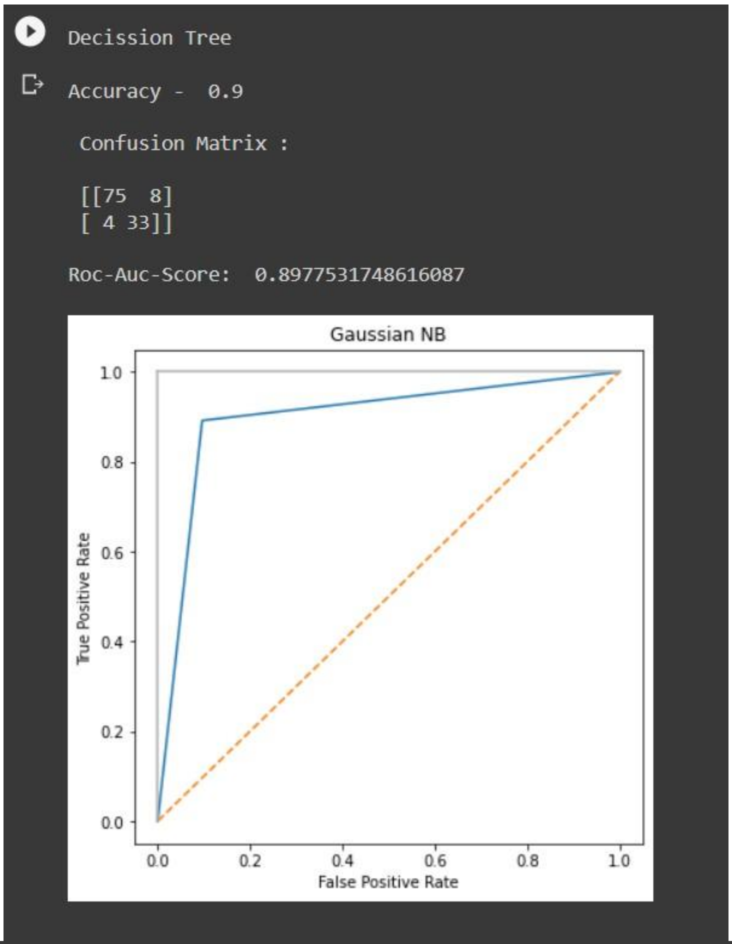
```
false_positive, true_positive, threshold = roc_curve(Y_val, y_score)
print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score)) print()
plt.subplots(1, figsize = (6, 6)) plt.title("Gaussian NB")
plt.plot(false_positive, true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False Positive Rate")
plt.show()
 dt =
DecisionTreeClassifier()
dt.fit(X_train, Y_train)
 y_val_pred = dt.predict(X_val)
dt_social = dt.score(X_val, Y_val)*100

#Prediction and Confusion Matrix
print("\n\nDecission Tree") print(f"\nAccuracy -
", dt.score(X_val, Y_val)) print(f"\n Confusion
Matrix : ") print("\n", confusion_matrix(Y_val,
y_val_pred))

#Plotting of AUCROC y_score =
dt.predict_proba(X_val)[ :,1]
 false_positive, true_positive, threshold = roc_curve(Y_val,
y_score) print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score))
print()  plt.subplots(1, figsize = (6, 6)) plt.title("Gaussian NB")
plt.plot(false_positive, true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False
Positive Rate") plt.show()

# Kfolds cross validation pipeline = make_pipeline(StandardScaler(),
RandomForestClassifier(n_estimato rs=100, max_depth=4)) X = X_train y =
Y_train scores = cross_val_score(pipeline, X, y, cv=10, n_jobs=1)
print('\n\nCross Validation accuracy scores: %s' % scores)
# Random Forest Classifier
```

```
m = RandomForestClassifier()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) #
print(X_train.head(), X_test.head(), y_train.head(), y_test.head())
m.fit(X_train, y_train) print(f"\nAccuracy of Random forest classifier is
{m.score(X_test, y_test)}" )
```

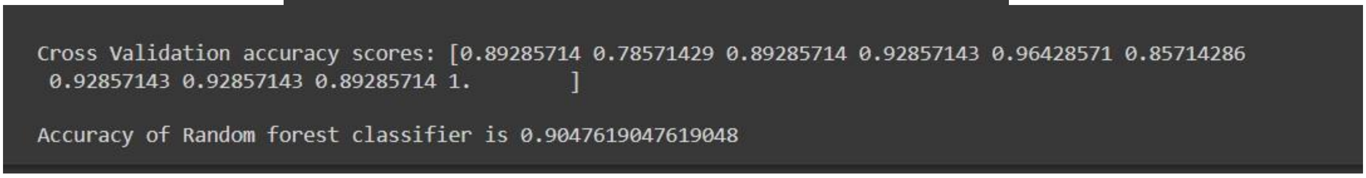
## Output :-



```
Social Network Ads
Naive Baeyes Classifier

Accuracy -  0.8892857142857142

 Confusion Matrix :

[[77  6]
 [ 8 29]]

Roc-Auc-Score:  0.9511559752523608
```



```
Decission Tree

Accuracy -  0.9

 Confusion Matrix :

 [[75  8]
 [ 4 33]]

Roc-Auc-Score:  0.8977531748616087
```



```
Cross Validation accuracy scores: [0.89285714 0.78571429 0.89285714 0.92857143 0.96428571 0.85714286
 0.92857143 0.92857143 0.89285714 1.        ]

Accuracy of Random forest classifier is 0.9047619047619048
```

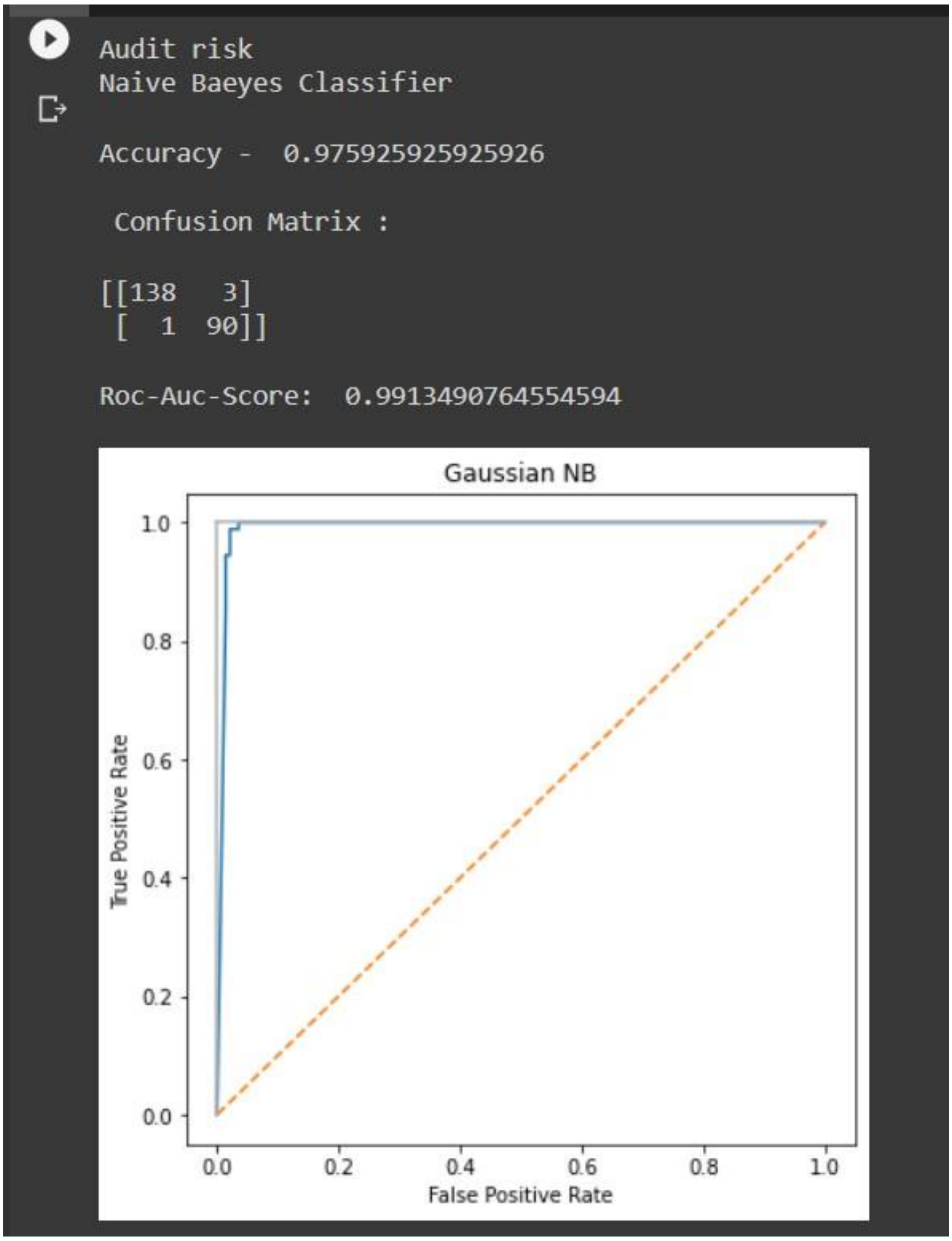
## Dataset – 5 :

```
import pandas as pd import numpy as np from sklearn.model_selection import
train_test_split from sklearn.naive_bayes import GaussianNB from
sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score, accu
racy_score, classification_report import matplotlib.pyplot as plt from
sklearn import tree from sklearn.tree import DecisionTreeClassifier from
sklearn.model_selection import cross_val_score from sklearn.pipeline import
make_pipeline from sklearn.preprocessing import StandardScaler from
sklearn.svm import SVC from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder import seaborn as sns

#Dataframe creation df =
pd.read_csv("/content/audit_data.csv") df =
df.dropna()

# Train test split  y =
df.Risk x =
df.drop(['Risk'],axis=1)

X_train, X_val, Y_train, Y_val = train_test_split(x, y, test_size = 0.3, ran
dom_state=255)

# Gaussian model gnb =
GaussianNB()
gnb.fit(X_train, Y_train)
 y_val_pred = gnb.predict(X_val)
y_train_pred = gnb.predict(X_train)
nb_audit = gnb.score(X_train, Y_train)*100

# Prediction and Confusion Matrix print("Audit risk
")       print("Naive       Baeyes       Classifier")
print(f"\nAccuracy - ", gnb.score(X_train, Y_train))
print(f"\n Confusion Matrix : \n")
con_matrix = confusion_matrix(Y_val,y_val_pred) print(con_matrix)

#Plotting of AUCROC
y_score = gnb.predict_proba(X_val)[ :,1]
```

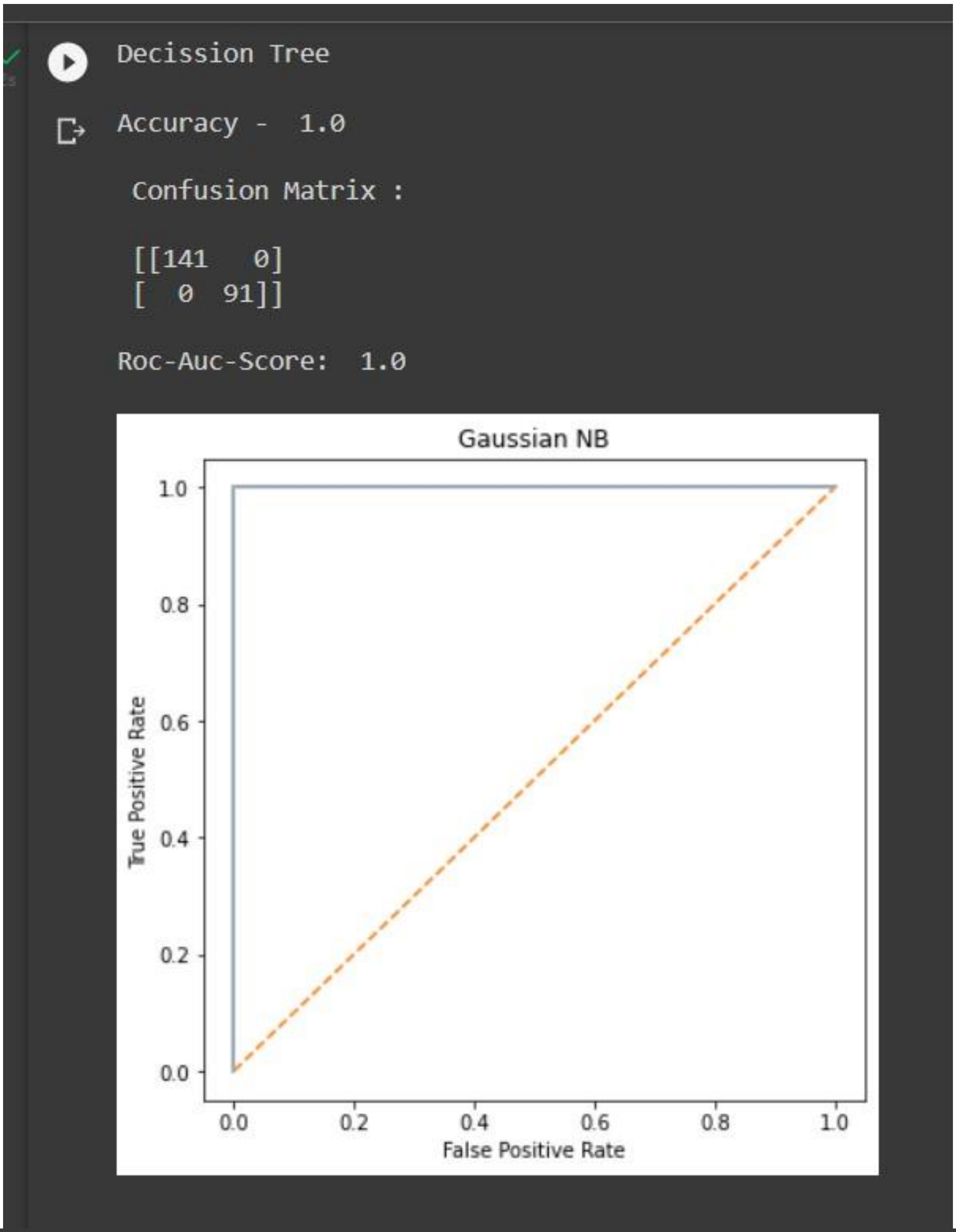
```
false_positive, true_positive, threshold = roc_curve(Y_val, y_score)
print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score)) print()
plt.subplots(1, figsize = (7, 7)) plt.title("Gaussian NB")
plt.plot(false_positive, true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False Positive Rate")
plt.show()
 dt =
DecisionTreeClassifier()
dt.fit(X_train, Y_train)
 y_val_pred = dt.predict(X_val)
dt_audit = dt.score(X_val, Y_val)*100

#Prediction and Confusion Matrix
print("\n\nDecission Tree") print(f"\nAccuracy -
", dt.score(X_val, Y_val)) print(f"\n Confusion
Matrix : ") print("\n", confusion_matrix(Y_val,
y_val_pred))

#Plotting of AUCROC y_score =
dt.predict_proba(X_val)[ :,1]
 false_positive, true_positive, threshold = roc_curve(Y_val,
y_score) print("\nRoc-Auc-Score: ", roc_auc_score(Y_val, y_score))
print()  plt.subplots(1, figsize = (7, 7)) plt.title("Gaussian NB")
plt.plot(false_positive, true_positive) plt.plot([0, 1], ls = "--")
plt.plot([0,0], [1,0], c="0.7"), plt.plot([1,1], c="0.7")
plt.ylabel("True Positive Rate") plt.xlabel("False Positive Rate")
plt.show()

# Kfolds cross validation pipeline = make_pipeline(StandardScaler(),
RandomForestClassifier(n_estimato rs=100, max_depth=4)) X = X_train y =
Y_train scores = cross_val_score(pipeline, X, y, cv=10, n_jobs=1)
print('\n\nCross Validation accuracy scores: %s' % scores)
# Random Forest Classifier m
= RandomForestClassifier()
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) #
print(X_train.head(), X_test.head(), y_train.head(), y_test.head())
m.fit(X_train, y_train) print(f"\nAccuracy of Random forest classifier is
{m.score(X_test, y_test)}" )
```

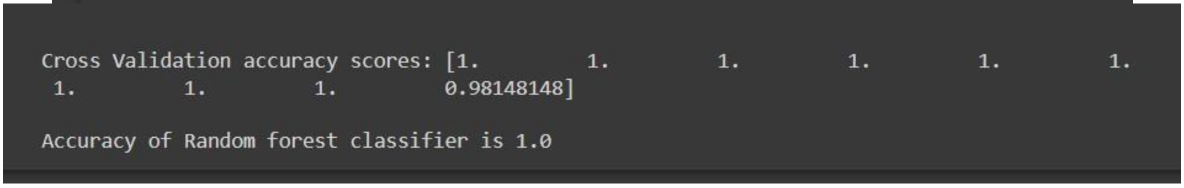
## Output :-



```
Audit risk
Naive Baeyes Classifier

Accuracy -  0.975925925925926

 Confusion Matrix :

[[138   3]
 [  1  90]]

Roc-Auc-Score:  0.9913490764554594
```



```
Decission Tree

Accuracy -  1.0

 Confusion Matrix :

 [[141   0]
 [  0  91]]

Roc-Auc-Score:  1.0
```



```
Cross Validation accuracy scores: [1.         1.         1.         1.         1.         1.
 1.         1.         1.         0.98148148]

Accuracy of Random forest classifier is 1.0
```

**<u>Conclusion:</u>**

Thus, we have successfully implemented Classification algorithm using Decision Tree ID3 and Naïve Bayes algorithm