Name : Yash Dilipkumar Upadhyay

Sap Id : 60004190128

Branch : Computer Engineering

Division : B4

DATA WAREHOUSE MINING

EXPERIMENT NO 2

Part A :

Code :

#Defining all the imports import pandas import matplotlib.pyplot from sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler, OrdinalEncoder, LabelEncoder from sklearn.naive_bayes import GaussianNB from sklearn.tree import DecisionTreeClassifier from sklearn.metrics import confusion_matrix, accuracy_score, classification_report import seaborn as sns

#Loading dataset

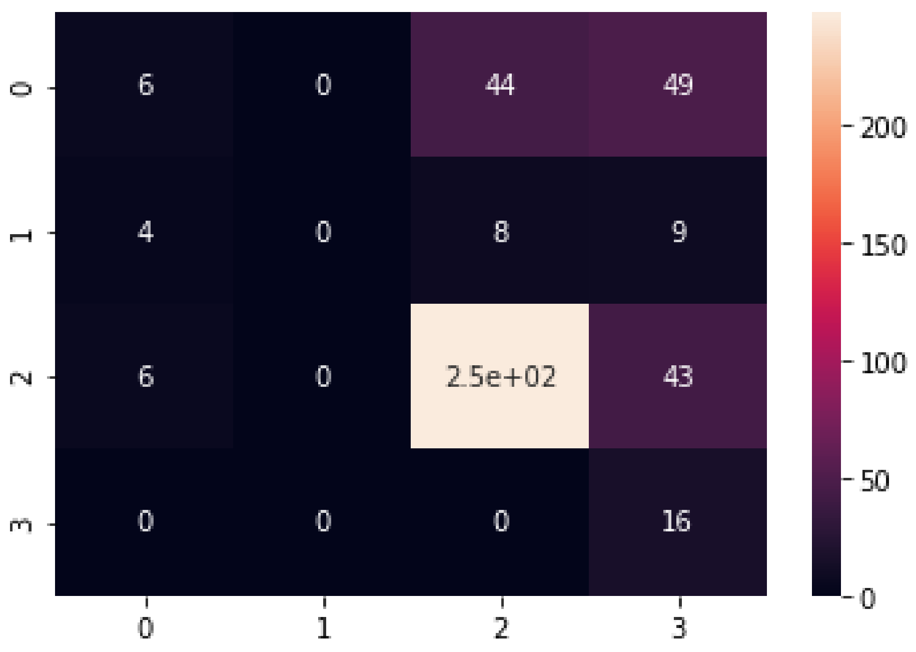
dataset = pandas.read_csv('car.data', names=['buying','maint','doors','persons','lug_boot','safety','class'],sep=',')

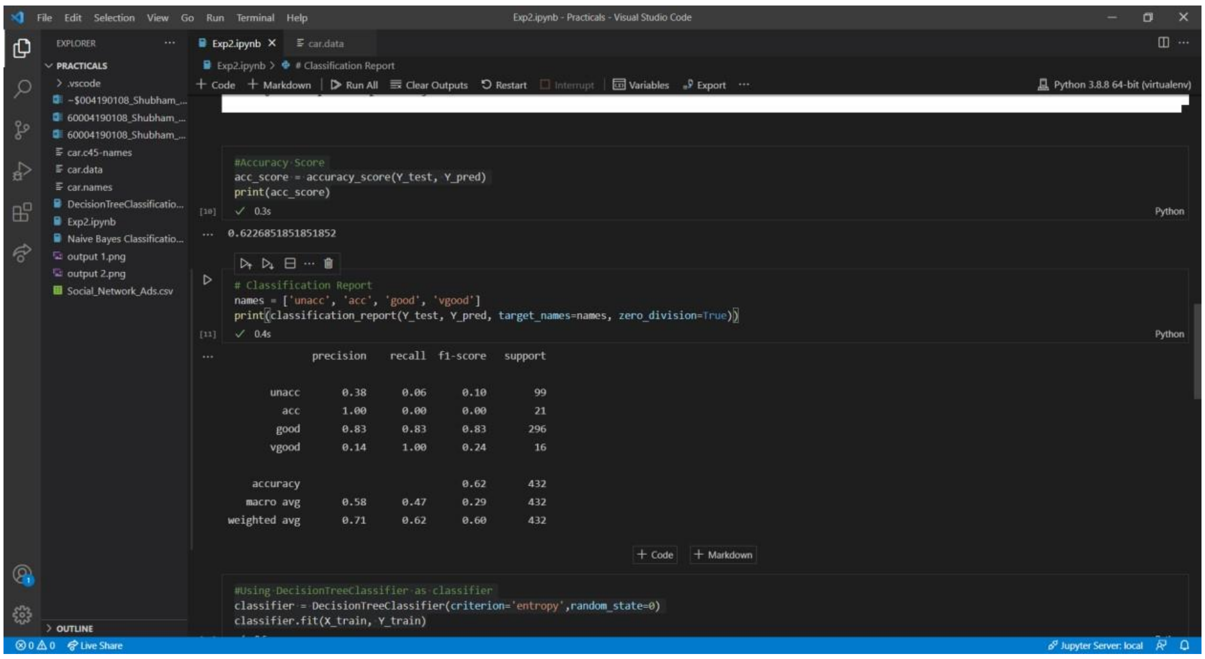
#Dividing dataset into Features and Class

```
X = dataset.iloc[:,:-1].values
Y = dataset.iloc[:,-1].values

#Splitting the Train and Test datasets from given Dataset
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25,
random_state=0) #Preprocessing Feature values oe = OrdinalEncoder()
X_train = oe.fit_transform(X_train)
X_test = oe.transform(X_test)

#Preprocessing Class values
le = LabelEncoder()
Y_train = le.fit_transform(Y_train)
Y_test = le.transform(Y_test)

#Using GaussianNB as
classifier classifier =
GaussianNB()
classifier.fit(X_train, Y_train)

#Prediction
Y_pred = classifier.predict(X_test)

#Confusion Matrix conf_matrix =
confusion_matrix(Y_test, Y_pred)
sns.heatmap(conf_matrix, annot=True)

#Accuracy Score acc_score =
accuracy_score(Y_test, Y_pred)
print(acc_score)
```

```
# Classification Report names = ['unacc', 'acc', 'good', 'vgood']
print(classification_report(Y_test, Y_pred, target_names=names,
zero_division=True)) #Using DecisionTreeClassifier as classifier classifier =
DecisionTreeClassifier(criterion='entropy',random_state=0)
classifier.fit(X_train, Y_train)

#Prediction
Y_pred = classifier.predict(X_test)

#Confusion         Matrix         conf_matrix         =
confusion_matrix(Y_test,                         Y_pred)
sns.heatmap(conf_matrix, annot=True)

#Accuracy Score acc_score =
accuracy_score(Y_test, Y_pred)
print(acc_score)

# Classification Report names = ['unacc', 'acc', 'good', 'vgood']
print(classification_report(Y_test, Y_pred, target_names=names,
zero_division=True))
```
Output-

Gaussian Naïve Bayes-





```
#Accuracy Score
acc_score = accuracy_score(Y_test, Y_pred)
print(acc_score)
```

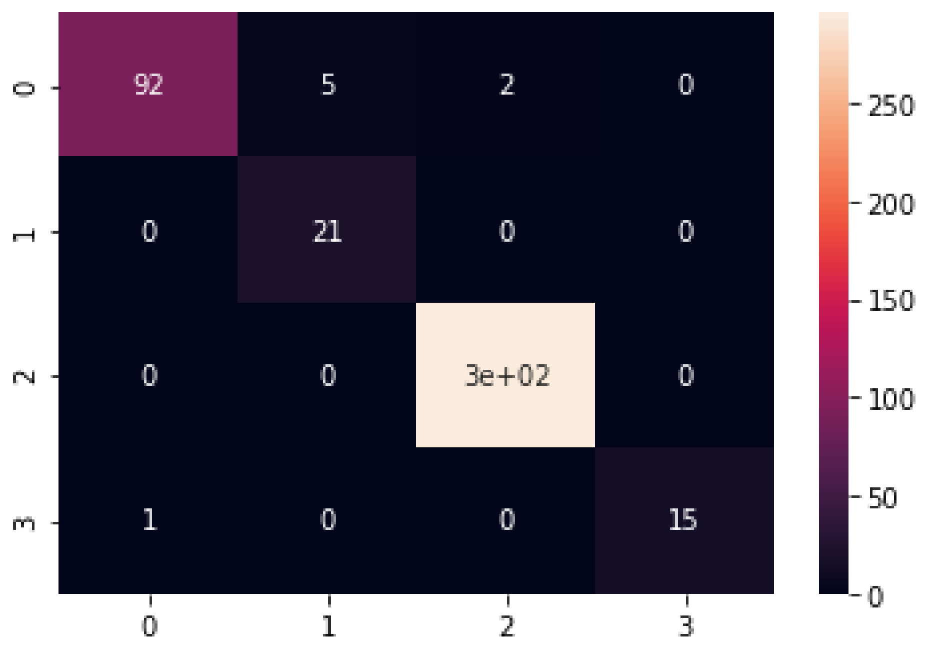
```
0.6226851851851852
```

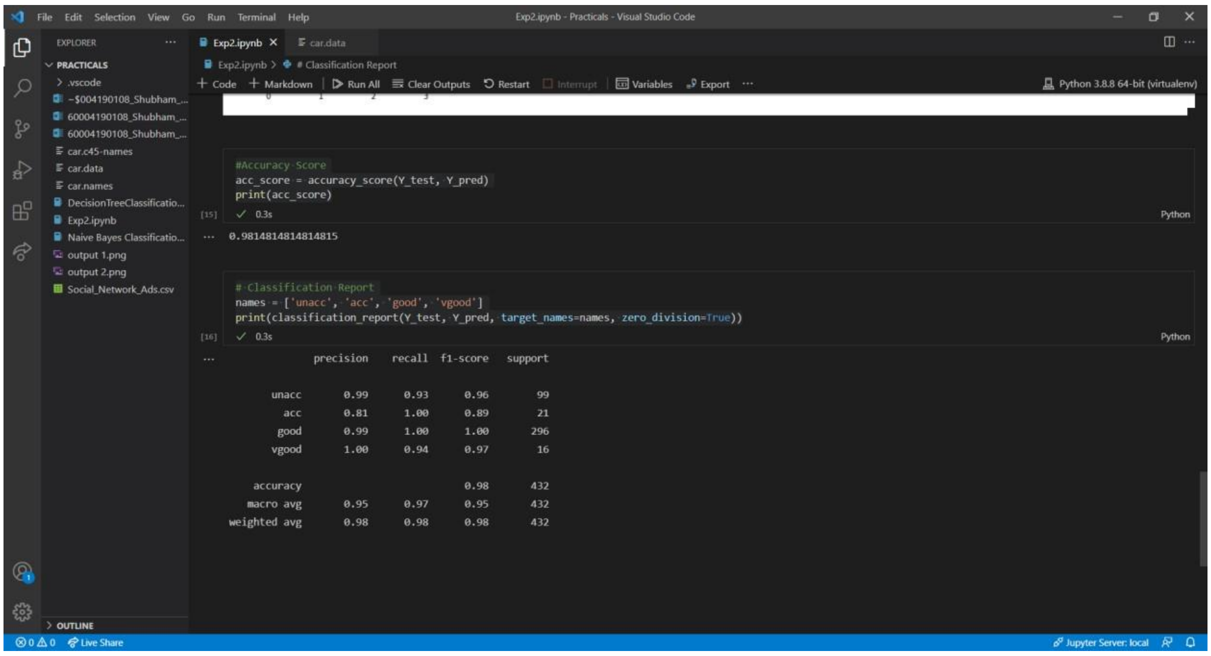
```
# Classification Report
names = ['unacc', 'acc', 'good', 'vgood']
print(classification_report(Y_test, Y_pred, target_names=names, zero_division=True))
```

```
              precision    recall  f1-score   support

       unacc       0.38      0.06      0.10        99
         acc       1.00      0.00      0.00        21
        good       0.83      0.83      0.83       296
       vgood       0.14      1.00      0.24        16

    accuracy                           0.62       432
   macro avg       0.58      0.47      0.29       432
weighted avg       0.71      0.62      0.60       432
```

```
#Using DecisionTreeClassifier as classifier
classifier = DecisionTreeClassifier(criterion='entropy',random_state=0)
classifier.fit(X_train, Y_train)
```

Decision Tree-





```
#Accuracy Score
acc_score = accuracy_score(Y_test, Y_pred)
print(acc_score)
```
[15] ✓ 0.3s

```
0.9814814814814815
```

```
# Classification Report
names = ['unacc', 'acc', 'good', 'vgood']
print(classification_report(Y_test, Y_pred, target_names=names, zero_division=True))
```
[16] ✓ 0.3s

```
              precision    recall  f1-score   support

       unacc       0.99      0.93      0.96        99
         acc       0.81      1.00      0.89        21
        good       0.99      1.00      1.00       296
       vgood       1.00      0.94      0.97        16

    accuracy                           0.98       432
   macro avg       0.95      0.97      0.95       432
weighted avg       0.98      0.98      0.98       432
```