Kulvir Singh

19BCE2074

Digital Assignment 4

Question 1 – Decision Tree

Problem Statement :

Building a Classifier Using Decision Tree to predict the normal and anomaly users.

- Use scikit-learn 0.24.1 Package to perform the process
- Print out the Accuracy and Confusion Matrix of Classification
- Document the step by step process and upload with output and Code

Procedure :

1. Load Labeled Training Data
2. Load and create Test Data
3. Using Decision Tree Classifier
4. Find the accuracy score and confusion at the end

Data has been taken from the internet and dataset file can be downloaded from this link - https://www.kaggle.com/saurabh00007/diabetescsv

Code :

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import tree
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedig
ree', 'age', 'label']
pima = pd.read_csv("diabetes.csv")
pima.columns = col_names
```

```
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigr
ee']
X = pima[feature_cols]
y = pima.label
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_
state=1)
dtc = DecisionTreeClassifier()
clf = dtc.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Code Screenshot:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import tree
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
pima = pd.read_csv("diabetes.csv")
pima.columns = col_names
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols]
y = pima.label
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1)
dtc = DecisionTreeClassifier()
clf = dtc.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Output Screenshots :

```
Accuracy: 0.6883116883116883
[[115  31]
 [ 41  44]]
              precision    recall  f1-score   support

           0       0.74      0.79      0.76       146
           1       0.59      0.52      0.55        85

    accuracy                           0.69       231
   macro avg       0.66      0.65      0.66       231
weighted avg       0.68      0.69      0.68       231
```

Question 2 – Naïve Bayes

Problem Statement :

Building a Classifier Using Naïve Bayes to predict the normal and anomaly users.

- Use scikit-learn 0.24.1 Package to perform the process
- Print out the Accuracy and Confusion Matrix of Classification
- Document the step by step process and upload with output and Code

Procedure :

1. Load Labeled Training Data

2. Load and create Test Data

3. Using Naïve Bayes Classifier

4. Find the accuracy score and confusion at the end

Data has been taken from the internet and dataset file can be downloaded from this link - https://www.kaggle.com/saurabh00007/diabetescsv

Code :

```
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import tree
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedig
ree', 'age', 'label']
pima = pd.read_csv("diabetes.csv")
pima.columns = col_names
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigr
ee']
X = pima[feature_cols]
y = pima.label
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_
state=1)
gnb = GaussianNB()
clf = gnb.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
```

```
print(classification_report(y_test, y_pred))
```

Code Screenshot:

```
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import tree
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
pima = pd.read_csv("diabetes.csv")
pima.columns = col_names
feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
X = pima[feature_cols]
y = pima.label
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1)
gnb = GaussianNB()
clf = gnb.fit(X_train,y_train)
y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Output Screenshots :

```
Accuracy: 0.7748917748917749
[[128  18]
 [ 34  51]]
              precision    recall  f1-score   support

           0       0.79      0.88      0.83       146
           1       0.74      0.60      0.66        85

    accuracy                           0.77       231
   macro avg       0.76      0.74      0.75       231
weighted avg       0.77      0.77      0.77       231
```