Lab-3 Classification

Introduction:

Classification is a supervised machine learning technique used to predict a categorical label or class based on input data. In this lab, we applied classification techniques to predict whether a patient has diabetes based on health-related attributes using the Naive Bayes and Decision Tree algorithms. The model was trained on a dataset (`diabetes_data.csv`) and evaluated using accuracy, ROC AUC score, and confusion matrix to compare performance.

DataSets

diabetes_data.csv

1 to 50 of 768 entries Filter

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0
7	196	90	0	0	39.8	0.451	41	1
9	119	80	35	0	29	0.263	29	1
2	90	68	42	0	38.2	0.503	27	1
4	111	72	47	207	37.1	1.39	56	1
3	180	64	25	70	34	0.271	26	0
7	133	84	0	0	40.2	0.696	37	0
7	106	92	18	0	22.7	0.235	48	0
9	171	110	24	240	45.4	0.721	54	1
7	159	64	0	0	27.4	0.294	40	0
0	180	66	39	0	42	1.893	25	1
1	146	56	0	0	29.7	0.564	29	0
2	71	70	27	0	28	0.586	22	0
7	103	66	32	0	39.1	0.344	31	1

3.1 Naive Bayes Classification

Implementation Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report
# Load the dataset from CSV
df = pd.read_csv('diabetes_data.csv')
# Split the data into features and target
X = df.drop(columns='Outcome')
y = df['Outcome']
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Initialize the Naive Bayes classifier
nb_classifier = GaussianNB()
# Train the model
nb_classifier.fit(X_train, y_train)
# Make predictions
y_pred_nb = nb_classifier.predict(X_test)
# Evaluate the model
accuracy_nb = accuracy_score(y_test, y_pred_nb)
print("Kishor Lab-3 Naive Bayes Classification")
print(f"Naive Bayes Accuracy: {accuracy_nb:.2f}")
print("\nClassification Report:\n", classification_report(y_test, y_pred_nb))
```

Output SnapShot:

```
Kishor Lab-3 Naive Bayes Classification
Naive Bayes Accuracy: 0.74

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.79      0.80       151
           1       0.62      0.66      0.64        80

    accuracy                           0.74       231
   macro avg       0.72      0.73      0.72       231
weighted avg       0.75      0.74      0.75       231
```

3.2 Decision Tree Classification

Implementation Code

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
# Load the dataset from CSV
df = pd.read_csv('diabetes_data.csv')
# Split the data into features and target
X = df.drop(columns='Outcome')
y = df['Outcome']
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Initialize the Decision Tree classifier
dt_classifier = DecisionTreeClassifier(criterion='entropy', random_state=42)
# Train the model
dt_classifier.fit(X_train, y_train)
# Make predictions
y_pred_dt = dt_classifier.predict(X_test)
# Evaluate the model
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print("Kishor Lab-3 Decision Tree Classification")
print(f"Decision Tree Accuracy: {accuracy_dt:.2f}")
print("\nClassification Report:\n", classification_report(y_test, y_pred_dt))
```

Output SnapShot:

```
Kishor Lab-3 Decision Tree Classification
Decision Tree Accuracy: 0.73

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.78      0.79       151
           1       0.60      0.62      0.61        80

    accuracy                           0.73       231
   macro avg       0.70      0.70      0.70       231
weighted avg       0.73      0.73      0.73       231
```

3.3 Comparing Accuracy

Implementation Code

```python
from sklearn.metrics import confusion_matrix, roc_auc_score
# Calculate confusion matrices
conf_matrix_nb = confusion_matrix(y_test, y_pred_nb)
conf_matrix_dt = confusion_matrix(y_test, y_pred_dt)
# Calculate ROC AUC scores
roc_auc_nb = roc_auc_score(y_test, y_pred_nb)
roc_auc_dt = roc_auc_score(y_test, y_pred_dt)
# Print comparison results
print("Kishor Lab-3 Comparing Naive Bayes and Decision Tree Classifiers")
print("\nNaive Bayes vs Decision Tree Classifier Performance:\n")
print(f"Naive Bayes Accuracy: {accuracy_nb:.2f}")
print(f"Decision Tree Accuracy: {accuracy_dt:.2f}")
print(f"Naive Bayes ROC AUC: {roc_auc_nb:.2f}")
print(f"Decision Tree ROC AUC: {roc_auc_dt:.2f}")
print("\nConfusion Matrix - Naive Bayes:\n", conf_matrix_nb)
print("\nConfusion Matrix - Decision Tree:\n", conf_matrix_dt)
```

Output SnapShot:

```
Kishor Lab-3 Comparing Naive Bayes and Decision Tree Classifiers

Naive Bayes vs Decision Tree Classifier Performance:

Naive Bayes Accuracy: 0.74
Decision Tree Accuracy: 0.73
Naive Bayes ROC AUC: 0.73
Decision Tree ROC AUC: 0.70

Confusion Matrix - Naive Bayes:
 [[119  32]
 [ 27  53]]

Confusion Matrix - Decision Tree:
 [[118  33]
 [ 30  50]]
```