0yozvgxmp

September 10, 2024

[1]:

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, accuracy_score

# Step 1: Load Dataset (Using Iris dataset as an example, replace with your␣
 ↪dataset)
from sklearn.datasets import load_iris
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

# Step 2: Train-Test Split
X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)

# Step 3: Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 4: Model Development

# 4.1 Logistic Regression
lr = LogisticRegression(max_iter=200)
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

# 4.2 k-Nearest Neighbors (k-NN)
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

```
y_pred_knn = knn.predict(X_test)

# 4.3 Decision Tree
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)

# Step 5: Model Evaluation

# Logistic Regression Metrics
print("Logistic Regression Classification Report:\n")
print(classification_report(y_test, y_pred_lr))

# k-Nearest Neighbors Metrics
print("k-NN Classification Report:\n")
print(classification_report(y_test, y_pred_knn))

# Decision Tree Metrics
print("Decision Tree Classification Report:\n")
print(classification_report(y_test, y_pred_dt))

# Step 6: Comparison of Accuracy
accuracy_lr = accuracy_score(y_test, y_pred_lr)
accuracy_knn = accuracy_score(y_test, y_pred_knn)
accuracy_dt = accuracy_score(y_test, y_pred_dt)

print(f"Logistic Regression Accuracy: {accuracy_lr:.4f}")
print(f"k-NN Accuracy: {accuracy_knn:.4f}")
print(f"Decision Tree Accuracy: {accuracy_dt:.4f}")

# You can save this comparison in a markdown file or generate a PDF from a␣
  ↪Jupyter Notebook for your GitHub submission.
```

```
<ipython-input-1-d201c2c95b31>:2: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of
pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better
interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

```
Logistic Regression Classification Report:

              precision    recall  f1-score   support
```

```
           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45

k-NN Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45

Decision Tree Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45

Logistic Regression Accuracy: 1.0000
k-NN Accuracy: 1.0000
Decision Tree Accuracy: 1.0000
```

[ ]: