Experiment No. 6: Understanding Accuracy Parameters

Aim

To understand the various accuracy parameters and write a Python program to read a dataset and apply decision tree classifier and measure various accuracy parameters.

Source Code

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, roc_curve
import matplotlib.pyplot as plt

# Read data
data = pd.read_csv('data.csv')
label_encoder = LabelEncoder()
data['activity'] = label_encoder.fit_transform(data['activity'])

# Feature selection
X = data[['time', 'timestamp', 'x-acceleration', 'y-acceleration', 'z-acceleration']]
y = data['activity']

# Preprocessing
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model creation and training
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(X_train, y_train)

# Predictions
y_pred_decision_tree = decision_tree_model.predict(X_test)

# Evaluation metrics
accuracy_decision_tree = accuracy_score(y_test, y_pred_decision_tree)
print(f"Accuracy on the test set (Decision Tree): {accuracy_decision_tree}")
```

```
# Confusion matrix
confusion_decision_tree = confusion_matrix(y_test, y_pred_decision_tree)
print("Confusion Matrix for Decision Tree:")
print(classification_decision_tree)

# Cross-validation scores
cv_scores_decision_tree = cross_val_score(decision_tree_model, X, y, cv=5)
print("Cross-Validation Scores for Decision Tree:", cv_scores_decision_tree)

# ROC Curve
n_classes = len(label_encoder.classes_)
for i in range(n_classes):
    y_one_vs_all = (y_test == i)
    y_score = decision_tree_model.predict_proba(X_test)[:, i]
    fpr, tpr = roc_curve(y_one_vs_all, y_score)
    plt.plot(fpr, tpr, label=f'ROC curve (area={roc_auc[i]:.2f}) for class {label_encoder.cl

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic for Decision Tree (One-vs-All)')
plt.legend(loc='lower right')
plt.show()
```

Output

The experiment produced the following results:

1. Decision Tree Classification Report showing:
    - Precision, recall, and F1-score for each class
    - Support values
    - Overall accuracy of 0.99
    - Weighted average scores of 0.99
2. Cross-validation scores for Decision Tree: [0.34141859, 0.21639939, 0.33853196, 0.28504617, 0.16092307]
3. ROC Curves showing the performance for each class:
    - Downstairs (area = 0.99)
    - Jogging (area = 1.00)
    - Sitting (area = 1.00)
    - Standing (area = 1.00)
    - Upstairs (area = 0.99)
    - Walking (area = 1.00)