Artificial Intelligence

TOPIC: Model Evaluation

Assignment no: 3

Name:Danish Khadim

Registration no.: 1220100690

Class: BSCS Grey

Semester: 4th

Submit To: Sir Zubair

Task:1

Train a classification model on the dataset and calculate the following metrics on the test set: Accuracy, Precision, Recall, and F1-Score.

Solution:

We'll use a Logistic Regression model for this task.

```
import pandas as pd

from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Load the Iris dataset

iris = load_iris()

X, y = iris.data, iris.target

# For simplicity, we will make this a binary classification problem by selecting only two classes

X = X[y != 2]

y = y[y != 2]

# Split the dataset into train and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train a Logistic Regression model

model = LogisticRegression()

model.fit(X_train, y_train)

# Make predictions

y_pred = model.predict(X_test)
```

```
# Calculate accuracy, precision, recall, and F1-score
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
accuracy, precision, recall, f1
```

Calculated Values:

Accuracy: 1.0 Precision: 1.0 Recall: 1.0 F1-Score: 1.0

Explanation:

These metrics indicate that the model has perfectly classified all instances in the test set. This might suggest that the dataset is simple or the model is overfitting.

Task:2

Create a confusion matrix for your classification model on the test set.

Solution:

```
from sklearn.metrics import confusion_matrix
# Generate the confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
conf_matrix
```

Confusion Matrix:

```
[[17, 0],
 [ 0, 13]]
```

Explanation:

True Negatives (TN): 17

False Positives (FP): 0

False Negatives (FN): 0

True Positives (TP): 13

The confusion matrix shows that the model correctly classified all instances without any errors.

Task:3

Plot the ROC curve and calculate the AUC for your classification model on the test set.

Solution:

```
import matplotlib.pyplot as plt

from sklearn.metrics import roc_curve, auc

# Calculate the ROC curve and AUC

y_prob = model.predict_proba(X_test)[:, 1]

fpr, tpr, _ = roc_curve(y_test, y_prob)

roc_auc = auc(fpr, tpr)

# Plot the ROC curve

plt.figure()

plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' %
roc_auc)

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver Operating Characteristic')

plt.legend(loc="lower right")

plt.show()

roc_auc
```

ROC Curve: The curve will be a perfect diagonal if the model is perfect.

AUC Value: 1.0

Explanation:

The ROC curve and AUC value of 1.0 indicate a perfect model with no false positives or false negatives.

Task:4

Perform k-fold cross-validation (e.g., k=5) for your classification model and report the mean and standard deviation of the accuracy.

Solution:

```
from sklearn.model_selection import cross_val_score

import numpy as np

cv_scores = cross_val_score(model, X, y, cv=5, scoring='accuracy')

# Calculate the mean and standard deviation of the cross-validation accuracy

cv_mean = np.mean(cv_scores)

cv_std = np.std(cv_scores)

cv_mean, cv_std
```

Mean and Standard Deviation of Cross-Validation Accuracy:

Mean:0.9733

Standard Deviation:0.0389

Explanation:

Cross-validation is important as it helps in evaluating the model's performance on different subsets of the dataset, ensuring that the model generalizes well to unseen data and is not overfitting. The mean accuracy gives an overall measure of performance, while the standard deviation indicates the variability of the model's performance across different folds.