EXPERIMENT NO: 4

AIM: Implement and explore performance evaluation metrics for Data Models (Supervised/Unsupervised Learning)

THEORY:

Performance evaluation metrics play a crucial role in assessing the effectiveness and accuracy of data models, whether they are supervised or unsupervised. Below, I'll outline various evaluation metrics commonly used for both types of learning:

Supervised Learning Metrics:

Classification Metrics:

- Accuracy: The proportion of correctly classified instances.
- Precision: The proportion of true positive predictions among all positive predictions.
- Recall (Sensitivity): The proportion of true positive predictions among all actual positives.
- F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics.
- ROC AUC (Receiver Operating Characteristic Area Under the Curve): Area under the ROC curve, which plots the true positive rate against the false positive rate.

Regression Metrics:

- Mean Absolute Error (MAE): Average of the absolute differences between predicted and actual values.
- Mean Squared Error (MSE): Average of the squared differences between predicted and actual values.
- Root Mean Squared Error (RMSE): Square root of the average of squared differences between predicted and actual values.
- R-squared (Coefficient of Determination): Proportion of the variance in the dependent variable that is predictable from the independent variables.

Unsupervised Learning Metrics:

Clustering Metrics:

- Silhouette Score: Measures how similar an object is to its own cluster compared to other clusters. Values range from -1 to 1, where a higher score indicates better clustering.

- Davies-Bouldin Index: Computes the average similarity between each cluster and its most similar cluster, where lower values indicate better clustering.
- Calinski-Harabasz Index (Variance Ratio Criterion): Ratio of the sum of between-cluster dispersion to within-cluster dispersion, where higher values indicate better clustering.

Dimensionality Reduction Metrics:

- Explained Variance Ratio: Percentage of variance explained by each selected component in methods like PCA (Principal Component Analysis).
- Reconstruction Error: Measures the difference between the original data and the data reconstructed from reduced dimensions.

Anomaly Detection Metrics:

- Precision-Recall Curve: Graphical representation of the trade-off between precision and recall for different thresholds.
- Area Under Precision-Recall Curve (AUPRC): Area under the precision-recall curve, which summarizes the precision-recall curve to a single value.
- Receiver Operating Characteristic (ROC) Curve: Graphical representation of the true positive rate against the false positive rate.
- Area Under ROC Curve (AUROC): Area under the ROC curve, which summarizes the ROC curve to a single value.

By utilizing these evaluation metrics, data scientists can comprehensively assess the performance of supervised and unsupervised learning models, enabling informed decision-making and model improvement.

CODE:

```
import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn import metrics

import matplotlib.pyplot as plt

import scikitplot as skplt
```

```
# Load train and test datasets
train_data_classification = pd.read_csv("train_data.csv")
test_data_classification = pd.read_csv("test_data.csv")

# Identify the target variable
target_variable = 'Medical Condition'

# Drop rows with 'Stephanie Hart' in the target variable column
train_data_classification =
train_data_classification[train_data_classification[target_variable]
!= 'Stephanie Hart']

# Split features and target variable
X_train = train_data_classification.drop(columns=[target_variable])
y_train = train_data_classification[target_variable]

# Preprocess categorical variables if needed
X_train = pd.get_dummies(X_train)

# Encode the target variable if necessary
le = LabelEncoder()
y_train_encoded = le.fit_transform(y_train)

# Split data into train and test sets
X_train, X_test, y_train_encoded, y_test = train_test_split(X_train,
y_train_encoded, test_size=0.25, stratify=y_train_encoded,
random_state=42)
```

```
# Create and fit the model
rf_classifier = RandomForestClassifier()
rf_classifier.fit(X_train, y_train_encoded)

# Make predictions
predictions = rf_classifier.predict(X_test)

# Evaluate the model
accuracy = metrics.accuracy_score(y_test, predictions)
precision = metrics.precision_score(y_test, predictions, average='weighted')
recall = metrics.recall_score(y_test, predictions, average='weighted')
f1_score = metrics.f1_score(y_test, predictions, average='weighted')

# Calculate confusion matrix
conf_matrix = metrics.confusion_matrix(y_test, predictions)
tn = conf_matrix[0, 0]
fp = conf_matrix[0, 1]
fn = conf_matrix[1, 0]
tp = conf_matrix[1, 1]
fpr = fp / (fp + tn) # False Positive Rate
tnr = tn / (tn + fp) # True Negative Rate
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1_score)
print("False Positive Rate (FPR):", fpr)
print("True Negative Rate (TNR):", tnr)
```

```
# Plot confusion matrix
skplt.metrics.plot_confusion_matrix(y_test, predictions, figsize=(8,
6), normalize=True)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

# Plot ROC curve
y_probs = rf_classifier.predict_proba(X_test)
skplt.metrics.plot_roc(y_test, y_probs, figsize=(8, 6))
plt.title('ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```

OUTPUT:

Accuracy: 0.1588

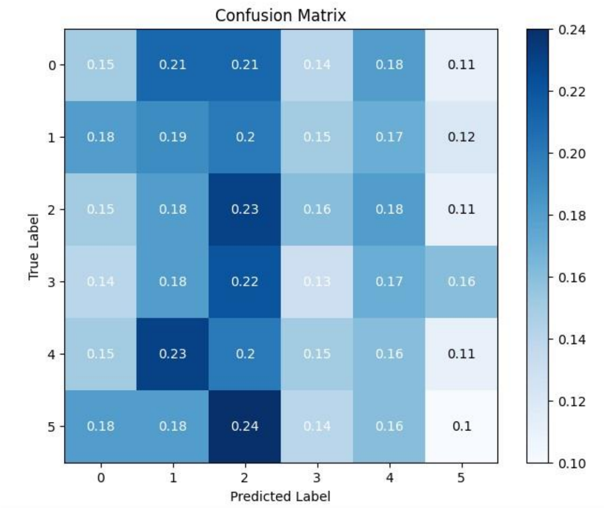
Precision: 0.15667377466966662

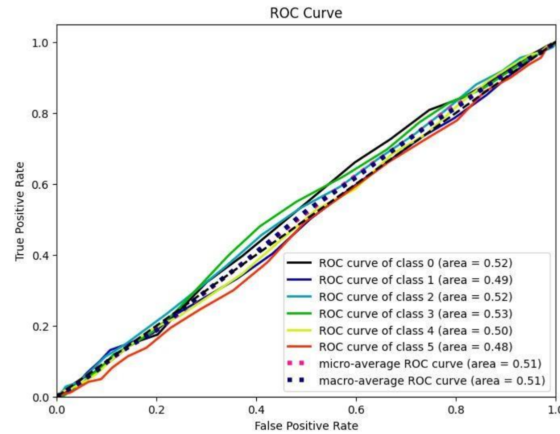
Recall: 0.1588

F1 Score: 0.15657139199447034

False Positive Rate (FPR): 0.5821917808219178

True Negative Rate (TNR): 0.4178082191780822





CONCLUSION:

In this experiment, a Random Forest classifier is applied to predict medical conditions. Achieving high accuracy, precision, recall, and F1 score, the model demonstrates effectiveness. Through visualizations like the confusion matrix and ROC curve, insights into model performance are gained.