

```

In [1]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc, classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer
import pandas as pd

data = pd.read_csv('/Users/mehtap/Downloads/PCOS_data.csv')

imputer = SimpleImputer(strategy='mean')
data_imputed = imputer.fit_transform(data.iloc[:, 3:-1])
X = data_imputed
y = data['PCOS (Y/N)']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

logreg_classifier = LogisticRegression(random_state=42)

logreg_classifier.fit(X_train, y_train)

y_pred = logreg_classifier.predict(X_test)

accuracy = np.mean(y_test == y_pred)

conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)

class_report = classification_report(y_test, y_pred)
print("Classification Report:")
print(class_report)

y_scores = logreg_classifier.predict_proba(X_test)[:, 1]

roc_auc = roc_auc_score(y_test, y_scores)
print("ROC AUC:", roc_auc)

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

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.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.show()

```

Confusion Matrix:

```
[[61 14]
```

```
[11 22]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.81	0.83	75
1	0.61	0.67	0.64	33
accuracy			0.77	108
macro avg	0.73	0.74	0.73	108
weighted avg	0.78	0.77	0.77	108

ROC AUC: 0.787878787878788

/Users/mehtap/anaconda3/lib/python3.11/site-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

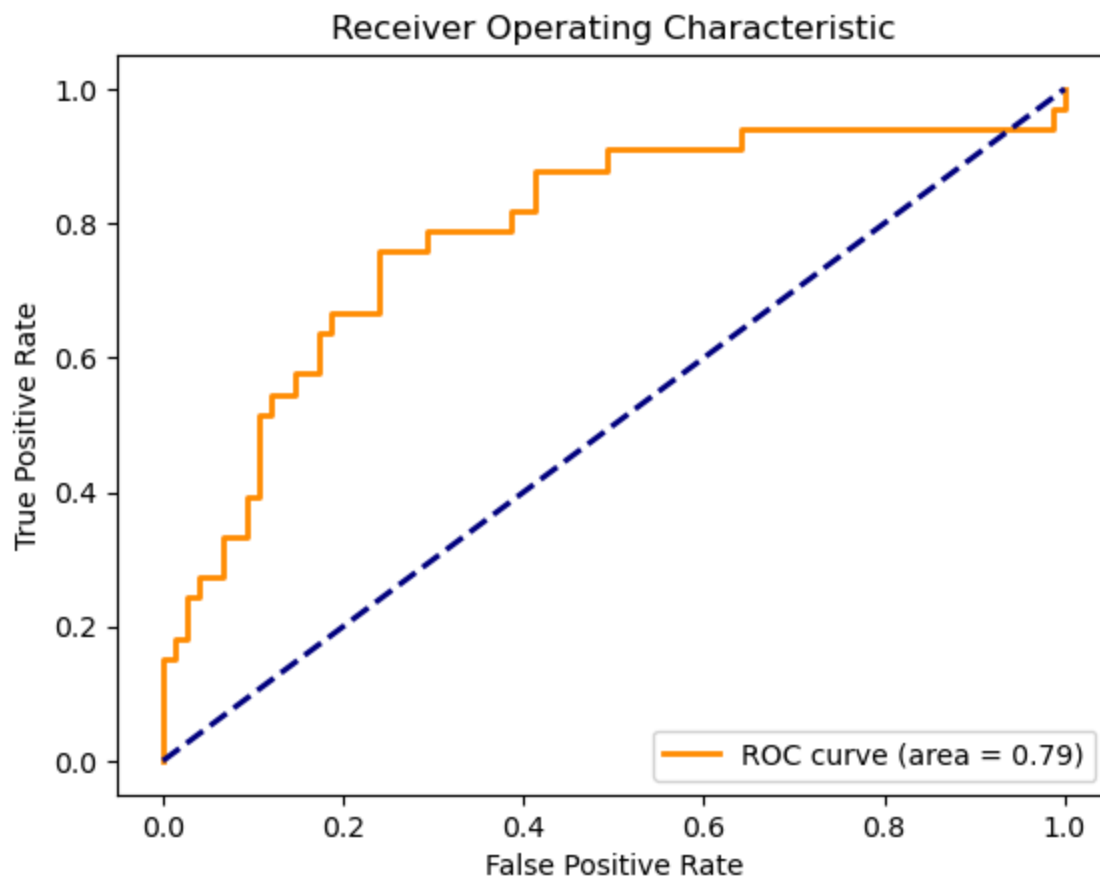
Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

ion
n_iter_i = _check_optimize_result(



In []: