AUC-ROC curve

AUC-ROC curve is one of the most commonly used metrics to evaluate the performance of machine learning algorithms particularly in the cases where we have imbalanced datasets.

Step 1: Import libraries

```
#Receiver Operating Characteristic (ROC) metric
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# roc curve and auc score
from sklearn.datasets import make_classification
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
```

+ Code + Text

Step 2: Defining a python function to plot the ROC curves.

```
def plot_roc_curve(fpr, tpr):
    plt.plot(fpr, tpr, color='orange', label='ROC')
    plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic (ROC) Curve')
    plt.legend()
    plt.show()
```

Step 3: Generate sample data.

```
data_X, class_label = make_classification(n_samples=1000, n_classes=2, weights=[1,1], r
```

Step 4: Split the data into train and test sub-datasets.

```
trainX, testX, trainy, testy = train_test_split(data_X, class_label, test_size=0.3, ran
```

Step 5: Fit a model on the train data.

```
model = RandomForestClassifier()
model.fit(trainX, trainy)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Step 6: Predict probabilities for the test data.

```
probs = model.predict_proba(testX)
```

Step 7: Keep Probabilities of the positive class only.

```
probs = probs[:, 1]
```

Step 8: Compute the AUC Score.

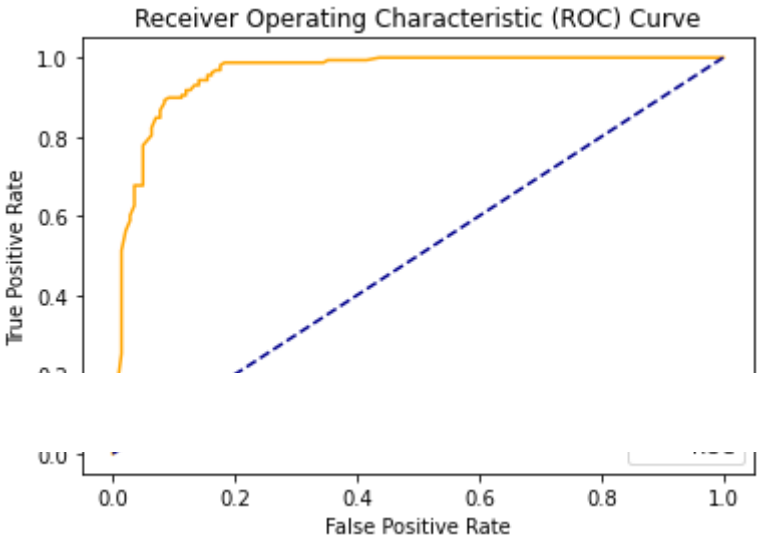
```
auc = roc_auc_score(testy, probs)
print('AUC: %.2f' % auc)
```

```
AUC: 0.96
```

Step 9: Get the ROC Curve.

```
fpr, tpr, thresholds = roc_curve(testy, probs)
```

```
plot_roc_curve(fpr, tpr)
```


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

1