SVC k-Fold Cross Validation

Importing the libraries

In [7]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

In [8]:

```
dataset = pd.read_csv('/Users/myyntiimac/Desktop/Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, -1].values
```

Feature Scaling

In [9]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X = sc.fit_transform(X)
```

Splitting the dataset into the Training set and Test set

In [12]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0
```

Training the Kernel SVM model on the Training set

In [13]:

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train, y_train)
```

Out[13]:

```
SVC(random_state=0)
```

Predicting the Test set results

In [36]:

```
y_pred = classifier.predict(X_test)
```

Making the Confusion Matrix

In [37]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[64  4]
 [ 3 29]]
```

In [38]:

```
bias = classifier.score(X_train, y_train)
bias
```

Out[38]:

```
0.9066666666666666
```

In [39]:

```
variance = classifier.score(X_test, y_test)
variance
```

Out[39]:

```
0.93
```

In [40]:

```
#ROC AND AUC
from sklearn.metrics import roc_curve, roc_auc_score
```

In [41]:

```
# Compute the False Positive Rate (FPR), True Positive Rate (TPR), and thresholds
fpr, tpr, thresholds = roc_curve(y_test, y_pred)
```

In [42]:

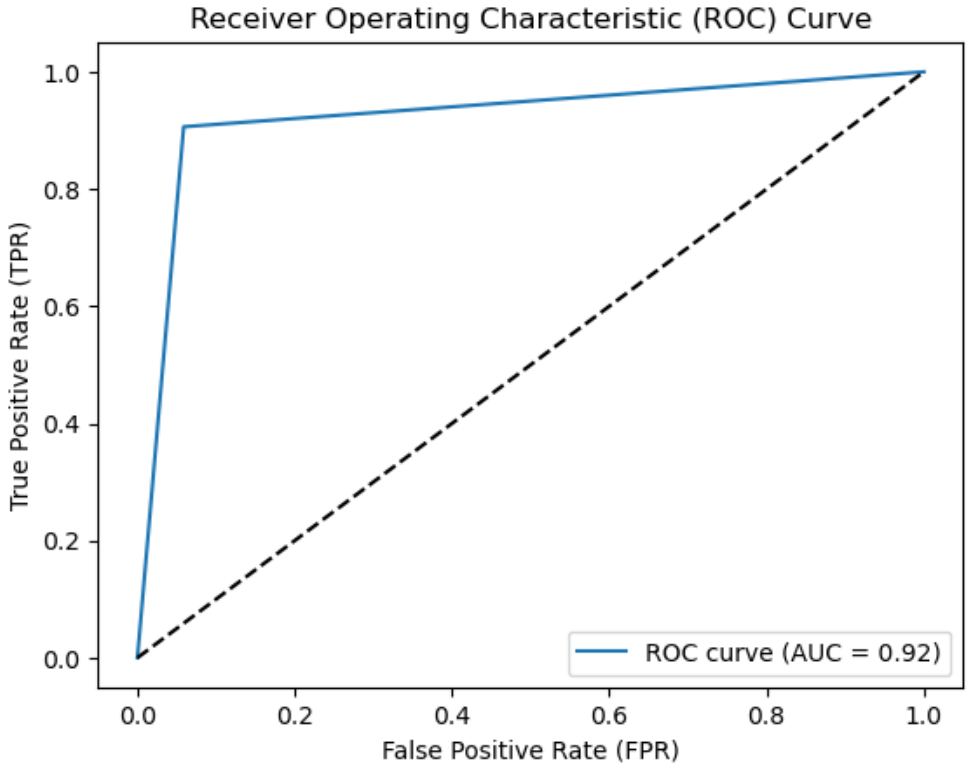
```
auc = roc_auc_score(y_test, y_pred)
auc
```

Out[42]:

```
0.9237132352941176
```

In [44]:

```
# Plotting the ROC curve
plt.plot(fpr, tpr, label='ROC curve (AUC = {:.2f})'.format(auc))
plt.plot([0, 1], [0, 1], 'k--')  # Random guess line
plt.xlabel('False Positive Rate (FPR)')
plt.ylabel('True Positive Rate (TPR)')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
```



Insight:In the case of AUC = 0.92, the model demonstrates reasonable discriminative ability, but there is still room for improvement. It correctly ranks 92% of the positive samples higher than the negative samples, on average, across different classification thresholds. However, it might misclassify some instances, leading to false positives or false negatives.

Applying k-Fold Cross Validation

In [45]:

```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

```
Accuracy: 90.00 %
Standard Deviation: 6.83 %
```

Applying gridserach

In [15]:

```
from sklearn.model_selection import GridSearchCV
```

In [16]:

```
# Define the parameter grid
parameters = [{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
              {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.1, 0.2, 0.3, 0.4, 0.5
```

In [18]:

```
# Create an object of GridSearchCV
grid_search = GridSearchCV(estimator=classifier, param_grid=parameters, cv=10)
grid_search = grid_search.fit(X_train, y_train)
best_accuracy = grid_search.best_score_
best_parameters = grid_search.best_params_
print("Best Accuracy: {:.2f} %".format(best_accuracy*100))
print("Best Parameters:", best_parameters)
```

```
Best Accuracy: 91.00 %
Best Parameters: {'C': 1, 'gamma': 0.7, 'kernel': 'rbf'}
```

In [ ]:

```

```