```
# k-Fold Cross Validation

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv(r'C:\Users\Hanshu\Desktop\excel data_ML\Churn_Modelling.csv')
x = dataset.iloc[:, [2,3]].values
y = dataset.iloc[:, -1].values


from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
x[:, 0] = le.fit_transform(x[:, 0])


#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x = sc.fit_transform(x)

# Splitting the dataset into the Training set and Test set
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
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(x_train , y_train)


# Predicting the Test set results
y_pred = classifier.predict(x_test)


# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

from sklearn.metrics import accuracy_score
ac = accuracy_score(y_test, y_pred)
print(ac)

bias = classifier.score(x_train, y_train)
bias

variance = classifier.score(x_test, y_test)
variance


# you can add implement auc & roc

# Applying k-Fold Cross Validation
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(classifier, x_train, y_train, cv=10)
print("Accuracy: {:.2f}%".format(accuracies.mean()*100))
#print("Standard Deviation: {:.2f}%".format(accuracies.std()*100))
```

```
# staratified cv
```