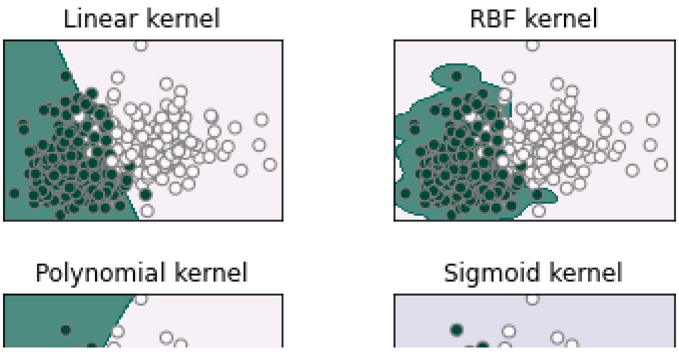
```
#Importing the necessary packages and libaries
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn import svm, datasets
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
breast_cancer= datasets.load_breast_cancer()
#Store variables as target y and the first two features as X (sepal length and sepal width of the iris flow
X = breast_cancer.data[:, :2]
y = breast_cancer.target
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state = 0)
linear = svm.SVC(kernel='linear', C=1, decision_function_shape='ovo').fit(X_train, y_train)
rbf = svm.SVC(kernel='rbf', gamma=1, C=1, decision_function_shape='ovo').fit(X_train, y_train)
poly = svm.SVC(kernel='poly', degree=3, C=1, decision_function_shape='ovo').fit(X_train, y_train)
sig = svm.SVC(kernel='sigmoid', C=1, decision_function_shape='ovo').fit(X_train, y_train)
#stepsize in the mesh, it alters the accuracy of the plotprint
#to better understand it, just play with the value, change it and print it
h = .01
#create the mesh
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),np.arange(y_min, y_max, h))
# create the title that will be shown on the plot
titles = ['Linear kernel','RBF kernel','Polynomial kernel','Sigmoid kernel']
for i, clf in enumerate((linear, rbf, poly, sig)):
 #defines how many plots: 2 rows, 2columns=> leading to 4 plots
 plt.subplot(2, 2, i + 1) #i+1 is the index
 #space between plots
 plt.subplots_adjust(wspace=0.4, hspace=0.4)
 Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
# Put the result into a color plot
 Z = Z.reshape(xx.shape)
 plt.contourf(xx, yy, Z, cmap=plt.cm.PuBuGn, alpha=0.7)
# Plot also the training points
 plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.PuBuGn, edgecolors='grey')

 plt.xlim(xx.min(), xx.max())
 plt.ylim(yy.min(), yy.max())
 plt.xticks(())
 plt.yticks(())
 plt.title(titles[i])
plt.show()
linear_pred = linear.predict(X_test)
poly_pred = poly.predict(X_test)
rbf_pred = rbf.predict(X_test)
sig_pred = sig.predict(X_test)
```




```
# retrieve the accuracy and print it for all 4 kernel functions
accuracy_lin = linear.score(X_test, y_test)
accuracy_poly = poly.score(X_test, y_test)
accuracy_rbf = rbf.score(X_test, y_test)
accuracy_sig = sig.score(X_test, y_test)
print("Accuracy Linear Kernel:", accuracy_lin)
```

```
Accuracy Linear Kernel: 0.8947368421052632
```

```
print("Accuracy Polynomial Kernel:", accuracy_poly)
print("Accuracy Radial Basis Kernel:", accuracy_rbf)
print("Accuracy Sigmoid Kernel:", accuracy_sig)
# creating a confusion matrix
cm_lin = confusion_matrix(y_test, linear_pred)
cm_poly = confusion_matrix(y_test, poly_pred)
cm_rbf = confusion_matrix(y_test, rbf_pred)
cm_sig = confusion_matrix(y_test, sig_pred)
print(cm_lin)
print(cm_poly)
print(cm_rbf)
print(cm_sig)
```

```
Accuracy Polynomial Kernel: 0.8947368421052632
Accuracy Radial Basis Kernel: 0.8421052631578947
Accuracy Sigmoid Kernel: 0.5877192982456141
[[40  7]
 [ 5 62]]
[[41  6]
 [ 6 61]]
[[37 10]
 [ 8 59]]
[[ 0 47]
 [ 0 67]]
```