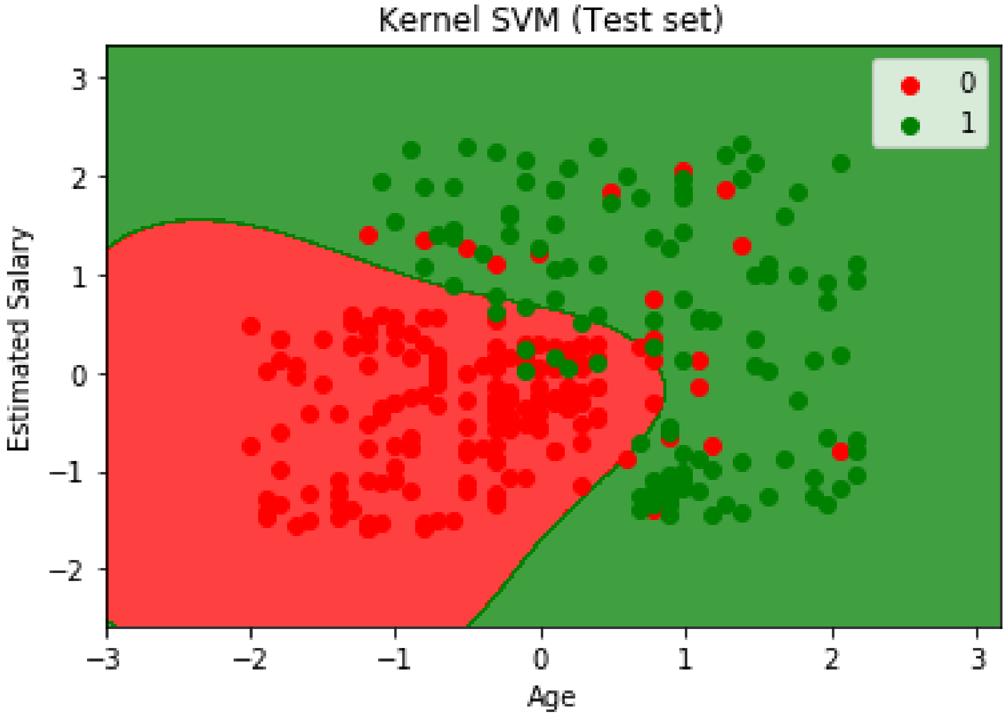
Kernal SVM

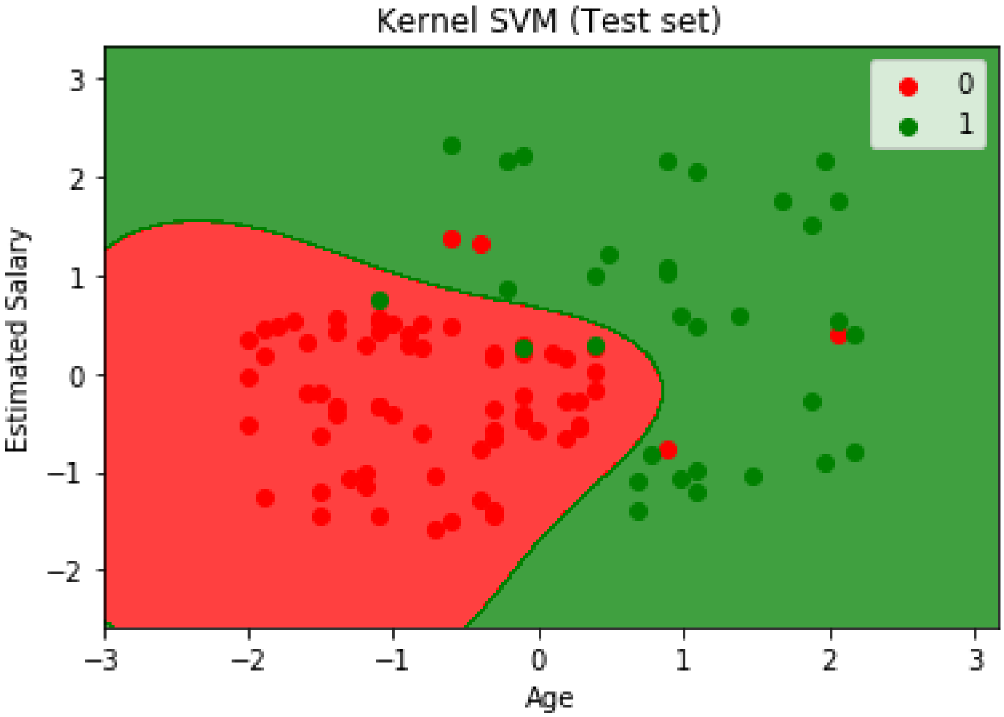
September 20, 2018

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: dataset = pd.read_csv("/media/coea/A4F698A8F6987BEC/A-Z/SVM/SVM/Social_Network_Ads.csv")

In [3]: x = dataset.iloc[:,[2,3]].values

In [4]: y = dataset.iloc[:,4].values

In [ ]: from sklearn.cross_validation import train_test_split

In [6]: x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = .25,random_state = 0)

In [ ]: from sklearn.preprocessing import StandardScaler
        sc_x = StandardScaler()
        x_train = sc_x.fit_transform(x_train)
        x_test = sc_x.transform(x_test)

In [8]: from sklearn.svm import SVC

In [9]: classifier = SVC(kernel = 'rbf',random_state=0)
        classifier.fit(x_train,y_train)

Out[9]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
          decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
          max_iter=-1, probability=False, random_state=0, shrinking=True,
          tol=0.001, verbose=False)

In [10]: y_pred = classifier.predict(x_test)

In [11]: from sklearn.metrics import confusion_matrix
         cn = confusion_matrix(y_test,y_pred)

In [12]: cn

Out[12]: array([[64,  4],
               [ 3, 29]])
```

```
In [13]: # Visualising the Test set results
         from matplotlib.colors import ListedColormap
         X_set, y_set = x_train, y_train
         X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max()
                              np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max()
         plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X
                      alpha = 0.75, cmap = ListedColormap(('red', 'green')))
         plt.xlim(X1.min(), X1.max())
         plt.ylim(X2.min(), X2.max())
         for i, j in enumerate(np.unique(y_set)):
             plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                         c = ListedColormap(('red', 'green'))(i), label = j)
         plt.title('Kernel SVM (Test set)')
         plt.xlabel('Age')
         plt.ylabel('Estimated Salary')
         plt.legend()
         plt.show()
```



```
In [14]: # Visualising the Test set results
         from matplotlib.colors import ListedColormap
         X_set, y_set = x_test, y_test
         X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max()
                              np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max()
         plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('Kernel SVM (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```


Kernel SVM (Test set)
Estimated Salary
Age
● 0
● 1