Support Vector Machines

Name : Harshvardhan Singh

Roll : 1019161

Importing Datasets

```
import pandas as pd

datasets = pd.read_csv("dataset.csv")
datasets.head()
```

```
    User ID  Gender  Age  EstimatedSalary  Purchased
0  15624510    Male   19            19000          0
1  15810944    Male   35            20000          0
2  15668575  Female   26            43000          0
3  15603246  Female   27            57000          0
4  15804002    Male   19            76000          0
```

variables:

x: Age

y: Estimated Salary

```
X = datasets.iloc[:, [2,3]].values
Y = datasets.iloc[:, 4].values
```

```
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)
```

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)
```

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_Train, Y_Train)
```

```
SVC(kernel='linear', random_state=0)
```

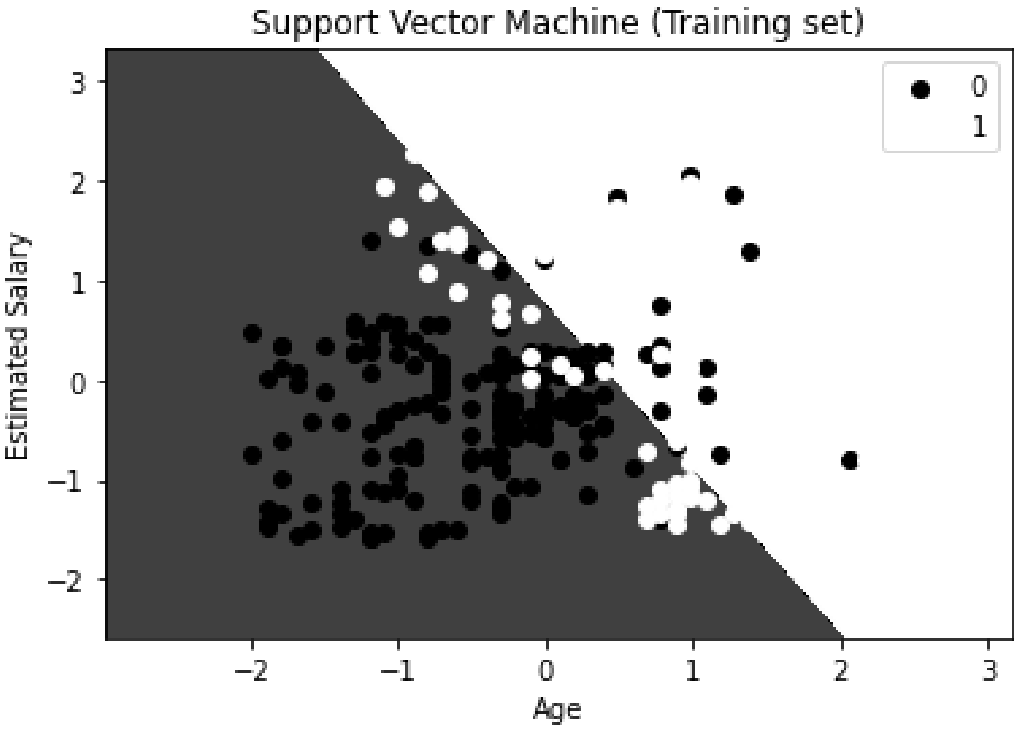
```
Y_Pred = classifier.predict(X_Test)
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_Test, Y_Pred)
```

Visualising the Training set results

```
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt
import numpy as np
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() + 1, step = 0.01),
                     np.arange(start = X_Set[:, 1].min() - 1, stop =
X_Set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('black', 'white')))
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(('black', 'white'))(i), label = j)
plt.title('Support Vector Machine (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric
RGB or RGBA sequence, which should be avoided as value-mapping will
have precedence in case its length matches with *x* & *y*.  Please use
the *color* keyword-argument or provide a 2-D array with a single row
if you intend to specify the same RGB or RGBA value for all points.
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric
RGB or RGBA sequence, which should be avoided as value-mapping will
have precedence in case its length matches with *x* & *y*.  Please use
the *color* keyword-argument or provide a 2-D array with a single row
if you intend to specify the same RGB or RGBA value for all points.
```

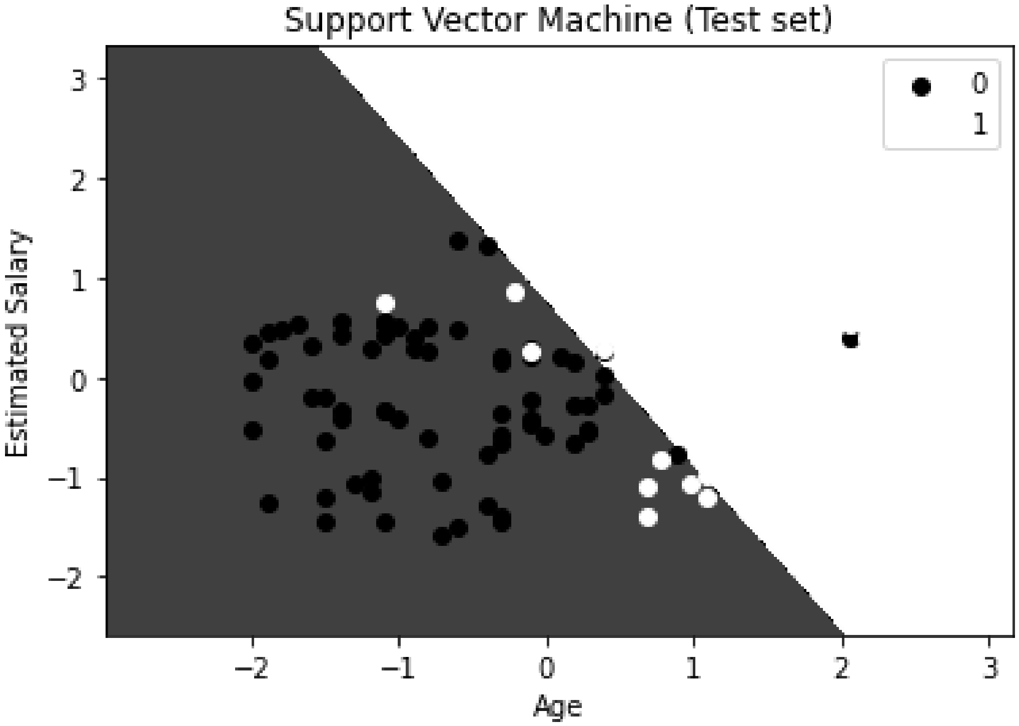


Visualising the Test set results

```
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() + 1, step = 0.01),
                     np.arange(start = X_Set[:, 1].min() - 1, stop =
X_Set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('black', 'white')))
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(('black', 'white'))(i), label = j)
plt.title('Support Vector Machine (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

```
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric
RGB or RGBA sequence, which should be avoided as value-mapping will
have precedence in case its length matches with *x* & *y*.  Please use
the *color* keyword-argument or provide a 2-D array with a single row
if you intend to specify the same RGB or RGBA value for all points.
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric
```

```
RGB or RGBA sequence, which should be avoided as value-mapping will
have precedence in case its length matches with *x* & *y*.  Please use
the *color* keyword-argument or provide a 2-D array with a single row
if you intend to specify the same RGB or RGBA value for all points.
```


Support Vector Machine (Test set)
Estimated Salary
Age
● 0