

Support Vector Machine

In [1]:

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

In [2]:

```
dataset = pd.read_csv('F:/Class_ML/Support Vector Machine/Ad_File.csv')
```

In [3]:

```
dataset.head()
```

Out[3]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0

In [4]:

```
X = dataset.iloc[:, [2, 3]]
y = dataset.iloc[:, 4]
```

In [5]:

```
dataset.head()
```

Out[5]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0

In [6]:

```
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 = 5)
```

In [7]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [8]:

```
# X_train
```

In [9]:

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', 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_deprecated',
    kernel='linear', 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
cm = confusion_matrix(y_test, y_pred)
```

In [12]:

```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test,y_pred)
```

In [13]:

```
accuracy
```

Out[13]:

```
0.85
```

In [14]:

```
from sklearn.model_selection import GridSearchCV
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, 0.6, 0.7, 0.8, 0.9]}]
```

In [15]:

```
grid_search = GridSearchCV(estimator = classifier,  
                           param_grid = parameters,  
                           scoring = 'accuracy',  
                           cv = 10,  
                           n_jobs = -1)
```

In [16]:

```
grid_search = grid_search.fit(X_train, y_train)
```

C:\Users\Hemanth Varma\Anaconda3\lib\site-packages\sklearn\model_selection_search.py:814: DeprecationWarning: The default of the `iid` parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results when test-set sizes are unequal.
DeprecationWarning)

In [17]:

```
accuracy = grid_search.best_score_
```

In [18]:

```
accuracy
```

Out[18]:

0.91

In [19]:

```
grid_search.best_params_
```

Out[19]:

```
{'C': 10, 'gamma': 0.3, 'kernel': 'rbf'}
```

In [20]:

```
grid_search.best_estimator_
```

Out[20]:

```
SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,  
    decision_function_shape='ovr', degree=3, gamma=0.3, kernel='rbf',  
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.  
    001,  
    verbose=False)
```

In [21]:

```

classifier = SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.3, kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True,
    tol=0.001, verbose=False)
classifier.fit(X_train, y_train)

```

Out[21]:

```

SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.3, kernel='rbf',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.
001,
    verbose=False)

```

In [22]:

```

y_pred = classifier.predict(X_test)

```

In [23]:

```

y_pred

```

Out[23]:

```

array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
    1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0,
    0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1,
    0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0,
    0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1], dtype=int64)

```

In [24]:

```

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

```

Out[24]:

```

array([[61,  5],
       [ 2, 32]], dtype=int64)

```

In [25]:

```

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

```

In [26]:

```

accuracy

```

Out[26]:

```

0.93

```