In [56]:
```
import pandas as pd
data= pd.read_csv('https://raw.githubusercontent.com/SivadineshPonrajan/Machine-Learning-Notebooks/master/Dataset/Churn_Modelling.csv')
data.head()
```

Out[56]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1

In [57]:
```
process=data.iloc[ :,[3,4,6,7,8,9,10,11,12,13] ]
process.head()
```

Out[57]:

	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	41	1	83807.86	1	0	1	112542.58	0
2	502	France	42	8	159660.80	3	1	0	113931.57	1
3	699	France	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	43	2	125510.82	1	1	1	79084.10	0

In [58]:
```
import numpy as np
for i in range(len(process.iloc[:,1])):
  if(process.iloc[:,1][i]=="France"):
    process.iloc[:,1][i]= np.int64(1)
  elif(process.iloc[:,1][i]=="Spain"):
    process.iloc[:,1][i]=np.int64(2)
  else:
    process.iloc[:,1][i]=np.int64(3)
process.head()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[58]:

	CreditScore	Geography	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	1	42	2	0.00	1	1	1	101348.88	1
1	608	2	41	1	83807.86	1	0	1	112542.58	0
2	502	1	42	8	159660.80	3	1	0	113931.57	1
3	699	1	39	1	0.00	2	0	0	93826.63	0
4	850	2	43	2	125510.82	1	1	1	79084.10	0

In [ ]:
```
x,y=process.iloc[:,:-1],process.iloc[:,-1]
```

In [ ]:
```
from sklearn import preprocessing
x = preprocessing.StandardScaler().fit(x).transform(x.astype(float))
x_train,x_test,y_train,y_test=x[:8000:],x[8000::],y[:8000:],y[8000::]
```

In [61]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(8000, 9)
(2000, 9)
(8000,)
(2000,)
```

In [62]:
```
from sklearn.linear_model import SGDClassifier
sgd_clf = SGDClassifier(random_state=50,average=True)
sgd_clf.fit(x_train,y_train)
```

Out[62]:
```
SGDClassifier(alpha=0.0001, average=True, class_weight=None,
              early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
              l1_ratio=0.15, learning_rate='optimal', loss='hinge',
              max_iter=1000, n_iter_no_change=5, n_jobs=None, penalty='12',
              power_t=0.5, random_state=50, shuffle=True, tol=0.001,
              validation_fraction=0.1, verbose=0, warm_start=False)
```

In [69]:
```
print(y_test[30:40:])
sgd_clf.predict(x_test[30:40:])
```

```
8030    1
8031    0
8032    0
8033    0
8034    0
8035    0
8036    0
8037    0
8038    0
8039    0
Name: Exited, dtype: int64
```

Out[69]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [64]:
```
from sklearn import metrics
print("Train set acc: " ,(metrics.accuracy_score(y_train, sgd_clf.predict(x_train))))
print("Test set acc: " ,(metrics.accuracy_score(y_test, sgd_clf.predict(x_test))))
```

```
Train set acc:  0.80375
Test set acc:  0.8105
```

In [73]:
```
training_score = sgd_clf.score(x_train, y_train)
testing_score = sgd_clf.score(x_test, y_test)

print("training_score: "+str(training_score))
y_pred= sgd_clf.predict(x_test)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix: For Training ")
print(cm)

print("testing_score: "+str(testing_score))
y_pred=sgd_clf.predict(x_train)

cm = confusion_matrix(y_train, y_pred)
print("Confusion Matrix: For Testing ")
print(cm)
```

```
training_score: 0.80375
Confusion Matrix: For Training
[[1590   20]
 [ 359   31]]
testing_score: 0.8105
Confusion Matrix: For Testing
[[6280   73]
 [1497  150]]
```

In [75]:
```
from sklearn.metrics import classification_report
cr = classification_report(y_train, y_pred)

print("classification Report:")
print(cr)
```

```
classification Report:
              precision    recall  f1-score   support

           0       0.81      0.99      0.89      6353
           1       0.67      0.09      0.16      1647

    accuracy                           0.80      8000
   macro avg       0.74      0.54      0.52      8000
weighted avg       0.78      0.80      0.74      8000
```

In [ ]:
```

```