Support Vector Machines - 2¶

In [1]:
```
# Importig Libraries
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.preprocessing import StandardScaler

from sklearn import svm
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report


from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split, cross_val_score

import warnings
warnings.filterwarnings('ignore')
```

EDA

In [2]:
```
train_data = pd.read_csv('SalaryData_Train(1).csv')
test_data = pd.read_csv('SalaryData_Test(1).csv')
```

In [3]:
```
train_data
```

Out[3]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	M
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	M
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	M
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	M
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Fem
...	...	...	...	...	...	...	...	...	
30156	27	Private	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Fem
30157	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	M
30158	58	Private	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Fem
30159	22	Private	HS-grad	9	Never-married	Adm-clerical	Own-child	White	M
30160	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Fem

30161 rows × 14 columns

In [4]:
```
train_data
```

Out[4]:

	age	workclass	education	educationno	maritalstatus	occupation	relationship	race	s
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	M
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	M
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	M
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	M
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Fem
...	...	...	...	...	...	...	...	...	
30156	27	Private	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Fem
30157	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	M
30158	58	Private	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Fem
30159	22	Private	HS-grad	9	Never-married	Adm-clerical	Own-child	White	M
30160	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Fem

30161 rows × 14 columns

In [5]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30161 entries, 0 to 30160
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            30161 non-null  int64 
 1   workclass      30161 non-null  object
 2   education      30161 non-null  object
 3   educationno    30161 non-null  int64 
 4   maritalstatus  30161 non-null  object
 5   occupation     30161 non-null  object
 6   relationship   30161 non-null  object
 7   race           30161 non-null  object
 8   sex            30161 non-null  object
 9   capitalgain    30161 non-null  int64 
 10  capitalloss    30161 non-null  int64 
 11  hoursperweek   30161 non-null  int64 
 12  native         30161 non-null  object
 13  Salary         30161 non-null  object
dtypes: int64(5), object(9)
memory usage: 3.2+ MB
```

In [6]:
```
test_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15060 entries, 0 to 15059
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            15060 non-null  int64 
 1   workclass      15060 non-null  object
 2   education      15060 non-null  object
 3   educationno    15060 non-null  int64 
 4   maritalstatus  15060 non-null  object
 5   occupation     15060 non-null  object
 6   relationship   15060 non-null  object
 7   race           15060 non-null  object
 8   sex            15060 non-null  object
 9   capitalgain    15060 non-null  int64 
 10  capitalloss    15060 non-null  int64 
 11  hoursperweek   15060 non-null  int64 
 12  native         15060 non-null  object
 13  Salary         15060 non-null  object
dtypes: int64(5), object(9)
memory usage: 1.6+ MB
```

In [7]:
```
train_data.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
age	30161.0	38.438115	13.134830	17.0	28.0	37.0	47.0	90.0
educationno	30161.0	10.121316	2.550037	1.0	9.0	10.0	13.0	16.0
capitalgain	30161.0	1092.044064	7406.466611	0.0	0.0	0.0	0.0	99999.0
capitalloss	30161.0	88.302311	404.121321	0.0	0.0	0.0	0.0	4356.0
hoursperweek	30161.0	40.931269	11.980182	1.0	40.0	40.0	45.0	99.0

In [8]:
```
test_data.describe().T
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
age	15060.0	38.768327	13.380676	17.0	28.0	37.0	48.0	90.0
educationno	15060.0	10.112749	2.558727	1.0	9.0	10.0	13.0	16.0
capitalgain	15060.0	1120.301594	7703.181842	0.0	0.0	0.0	0.0	99999.0
capitalloss	15060.0	89.041899	406.283245	0.0	0.0	0.0	0.0	3770.0
hoursperweek	15060.0	40.951594	12.062831	1.0	40.0	40.0	45.0	99.0

In [9]:
```
# Dropping columns which are not required

train_data = train_data.drop(['education', 'native'],axis = 1)
train_data
```

Out[9]:

	age	workclass	educationno	maritalstatus	occupation	relationship	race	sex	capital
0	39	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	2
1	50	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	
2	38	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	
3	53	Private	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
4	28	Private	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	
...	...	...	...	...	...	...	...	...	
30156	27	Private	12	Married-civ-spouse	Tech-support	Wife	White	Female	
30157	40	Private	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
30158	58	Private	9	Widowed	Adm-clerical	Unmarried	White	Female	
30159	22	Private	9	Never-married	Adm-clerical	Own-child	White	Male	
30160	52	Self-emp-inc	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15

30161 rows × 12 columns

In [10]:
```
# Dropping columns which are not required

test_data = test_data.drop(['education', 'native'],axis = 1)
test_data
```

Out[10]:

	age	workclass	educationno	maritalstatus	occupation	relationship	race	sex	capit
0	25	Private	7	Never-married	Machine-op-inspct	Own-child	Black	Male	
1	38	Private	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	
2	28	Local-gov	12	Married-civ-spouse	Protective-serv	Husband	White	Male	
3	44	Private	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	
4	34	Private	6	Never-married	Other-service	Not-in-family	White	Male	
...	...	...	...	...	...	...	...	...	
15055	33	Private	13	Never-married	Prof-specialty	Own-child	White	Male	
15056	39	Private	13	Divorced	Prof-specialty	Not-in-family	White	Female	
15057	38	Private	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	
15058	44	Private	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	
15059	35	Self-emp-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	

15060 rows × 12 columns

Feature Encoding

In [11]:
```
from sklearn.preprocessing import LabelEncoder
train_data = train_data.apply(LabelEncoder().fit_transform)
train_data
```

Out[11]:

	age	workclass	educationno	maritalstatus	occupation	relationship	race	sex	capitalgain
0	22	5	12	4	0	1	4	1	24
1	33	4	12	2	3	0	4	1	0
2	21	2	8	0	5	1	4	1	0
3	36	2	6	2	5	0	2	1	0
4	11	2	12	2	9	5	2	0	0
...	...	...	...	...	...	...	...	...	...
30156	10	2	11	2	12	5	4	0	0
30157	23	2	8	2	6	0	4	1	0
30158	41	2	8	6	0	4	4	0	0
30159	5	2	8	4	0	3	4	1	0
30160	35	3	8	2	3	5	4	0	107

30161 rows × 12 columns

In [12]:
```
test_data = test_data.apply(LabelEncoder().fit_transform)
test_data
```

Out[12]:

	age	workclass	educationno	maritalstatus	occupation	relationship	race	sex	capitalgain
0	8	2	6	4	6	3	2	1	0
1	21	2	8	2	4	0	4	1	0
2	11	1	11	2	10	0	4	1	0
3	27	2	9	2	6	0	2	1	87
4	17	2	5	4	7	1	4	1	0
...	...	...	...	...	...	...	...	...	...
15055	16	2	12	4	9	3	4	1	0
15056	22	2	12	0	9	1	4	0	0
15057	21	2	12	2	9	0	4	1	0
15058	27	2	12	0	0	3	1	1	73
15059	18	3	12	2	3	0	4	1	0

15060 rows × 12 columns

In [13]:
```
# Splitting data into test data and train data

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(train_data,train_data['Salary
```

In [14]:
```
x_train
```

Out[14]:

	age	workclass	educationno	maritalstatus	occupation	relationship	race	sex	capitalgain
27112	18	2	8	2	2	0	4	1	0
13212	6	2	8	2	5	0	4	1	0
24111	22	2	12	2	3	5	4	0	0
22501	16	2	8	0	7	4	4	0	0
3872	10	3	8	5	0	1	4	1	0
...	...	...	...	...	...	...	...	...	...
29802	19	2	11	2	3	5	4	0	0
5390	24	4	9	2	6	0	4	1	49
860	20	2	8	2	2	0	4	1	0
15795	39	4	6	2	7	5	4	0	0
23654	28	2	9	5	2	1	4	1	0

21112 rows × 12 columns

In [15]:
```
x_test
```

Out[15]:

	age	workclass	educationno	maritalstatus	occupation	relationship	race	sex	capitalgain
217	11	2	14	4	9	1	4	1	0
24912	24	2	8	2	2	0	1	1	0
17780	36	2	8	2	7	0	4	1	0
12484	14	2	8	2	2	0	4	1	0
8890	17	1	12	3	9	1	4	0	0
...	...	...	...	...	...	...	...	...	...
20990	15	2	9	2	11	2	1	0	0
23327	30	4	9	2	2	0	4	1	0
24639	24	2	12	5	9	1	2	0	117
7738	7	2	8	4	11	1	4	1	0
3882	9	5	12	4	9	1	4	1	0

9049 rows × 12 columns

In [16]:
```
y_train
```

Out[16]:
```
27112    1
13212    0
24111    1
22501    0
3872     0
        ..
29802    0
5390     1
860      1
15795    0
23654    0
Name: Salary, Length: 21112, dtype: int32
```

In [17]:
```
y_test
```

Out[17]:
```
217      1
24912    0
17780    1
12484    0
8890     0
        ..
20990    0
23327    1
24639    1
7738     0
3882     0
Name: Salary, Length: 9049, dtype: int32
```

Building model with Grid Search CV

In [ ]:
```
clf = SVC()
param_grid = [{'kernel':['rbf'],'gamma':[50,5,10,0.5],'C':[15,14,13,12,11,10,0.1,
gsv = GridSearchCV(clf,param_grid,cv=10)
gsv.fit(x_train,y_train)
```

In [ ]:
```
gsv.best_params_ , gsv.best_score_
```

In [ ]:
```
clf = SVC(C= 15, gamma = 50)
clf.fit(x_train , y_train)
y_pred = clf.predict(x_test)
acc = accuracy_score(y_test, y_pred) * 100
print("Accuracy =", acc)
confusion_matrix(y_test, y_pred)
```

In [ ]:
```

```