SVM on annual income classification

May 16, 2018

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

In [2]:
```
names = ['age','workclass','fnlwgt','education','education-num','marital-status' ,  'oc
'capital-gain','capital-loss','hours-per-week','native-country','income']
df_train = pd.read_csv('adult.data.txt',index_col=False,delim_whitespace=True,names=nam
df_test = pd.read_csv('adult_test.txt',index_col=False,delim_whitespace=True,names=name
df_train.dropna()
df_test.dropna()
```

Out[2]:
```
        age         workclass  fnlwgt     education  education-num  \
0        25           Private  226802          11th              7
1        38           Private   89814       HS-grad              9
2        28         Local-gov  336951    Assoc-acdm             12
3        44           Private  160323  Some-college             10
5        34           Private  198693          10th              6
7        63  Self-emp-not-inc  104626   Prof-school             15
8        24           Private  369667  Some-college             10
9        55           Private  104996       7th-8th              4
10       65           Private  184454       HS-grad              9
11       36       Federal-gov  212465     Bachelors             13
12       26           Private   82091       HS-grad              9
14       48           Private  279724       HS-grad              9
15       43           Private  346189       Masters             14
16       20         State-gov  444554  Some-college             10
17       43           Private  128354       HS-grad              9
18       37           Private   60548       HS-grad              9
20       34           Private  107914     Bachelors             13
21       34           Private  238588  Some-college             10
23       25           Private  220931     Bachelors             13
24       25           Private  205947     Bachelors             13
25       45  Self-emp-not-inc  432824       HS-grad              9
26       22           Private  236427       HS-grad              9
27       23           Private  134446       HS-grad              9
28       54           Private   99516       HS-grad              9
```

```
29      32  Self-emp-not-inc  109282  Some-college             10
30      46         State-gov  106444  Some-college             10
31      56  Self-emp-not-inc  186651          11th              7
32      24  Self-emp-not-inc  188274     Bachelors             13
33      23         Local-gov  258120  Some-college             10
34      26           Private   43311       HS-grad              9
...    ...               ...     ...           ...            ...
16248   25           Private  242136       HS-grad              9
16249   31           Private  112115       HS-grad              9
16250   49      Self-emp-inc   77132       HS-grad              9
16252   60           Private  117909     Assoc-voc             11
16253   39           Private  229647     Bachelors             13
16254   38           Private  149347       Masters             14
16255   43         Local-gov   23157       Masters             14
16256   23           Private   93977       HS-grad              9
16257   73      Self-emp-inc  159691  Some-college             10
16258   35           Private  176967  Some-college             10
16259   66           Private  344436       HS-grad              9
16260   27           Private  430340  Some-college             10
16261   40           Private  202168   Prof-school             15
16262   51           Private   82720       HS-grad              9
16263   22           Private  269623  Some-college             10
16264   64  Self-emp-not-inc  136405       HS-grad              9
16266   55           Private  224655       HS-grad              9
16267   38           Private  247547     Assoc-voc             11
16268   58           Private  292710    Assoc-acdm             12
16269   32           Private  173449       HS-grad              9
16270   48           Private  285570       HS-grad              9
16271   61           Private   89686       HS-grad              9
16272   31           Private  440129       HS-grad              9
16273   25           Private  350977       HS-grad              9
16274   48         Local-gov  349230       Masters             14
16275   33           Private  245211     Bachelors             13
16276   39           Private  215419     Bachelors             13
16278   38           Private  374983     Bachelors             13
16279   44           Private   83891     Bachelors             13
16280   35      Self-emp-inc  182148     Bachelors             13

           marital-status         occupation    relationship  \
0           Never-married  Machine-op-inspct       Own-child
1      Married-civ-spouse    Farming-fishing         Husband
2      Married-civ-spouse    Protective-serv         Husband
3      Married-civ-spouse  Machine-op-inspct         Husband
5           Never-married      Other-service   Not-in-family
7      Married-civ-spouse     Prof-specialty         Husband
8           Never-married      Other-service       Unmarried
9      Married-civ-spouse       Craft-repair         Husband
10     Married-civ-spouse  Machine-op-inspct         Husband
```

11	Married-civ-spouse	Adm-clerical	Husband
12	Never-married	Adm-clerical	Not-in-family
14	Married-civ-spouse	Machine-op-inspct	Husband
15	Married-civ-spouse	Exec-managerial	Husband
16	Never-married	Other-service	Own-child
17	Married-civ-spouse	Adm-clerical	Wife
18	Widowed	Machine-op-inspct	Unmarried
20	Married-civ-spouse	Tech-support	Husband
21	Never-married	Other-service	Own-child
23	Never-married	Prof-specialty	Not-in-family
24	Married-civ-spouse	Prof-specialty	Husband
25	Married-civ-spouse	Craft-repair	Husband
26	Never-married	Adm-clerical	Own-child
27	Separated	Machine-op-inspct	Unmarried
28	Married-civ-spouse	Craft-repair	Husband
29	Never-married	Prof-specialty	Not-in-family
30	Married-civ-spouse	Exec-managerial	Husband
31	Widowed	Other-service	Unmarried
32	Never-married	Sales	Not-in-family
33	Married-civ-spouse	Protective-serv	Husband
34	Divorced	Exec-managerial	Unmarried
...	...	...	...
16248	Divorced	Machine-op-inspct	Not-in-family
16249	Never-married	Machine-op-inspct	Not-in-family
16250	Married-civ-spouse	Exec-managerial	Husband
16252	Married-civ-spouse	Prof-specialty	Husband
16253	Never-married	Tech-support	Not-in-family
16254	Married-civ-spouse	Prof-specialty	Husband
16255	Married-civ-spouse	Exec-managerial	Husband
16256	Never-married	Machine-op-inspct	Own-child
16257	Divorced	Exec-managerial	Not-in-family
16258	Married-civ-spouse	Protective-serv	Husband
16259	Widowed	Sales	Other-relative
16260	Never-married	Sales	Not-in-family
16261	Married-civ-spouse	Prof-specialty	Husband
16262	Married-civ-spouse	Craft-repair	Husband
16263	Never-married	Craft-repair	Own-child
16264	Widowed	Farming-fishing	Not-in-family
16266	Separated	Priv-house-serv	Not-in-family
16267	Never-married	Adm-clerical	Unmarried
16268	Divorced	Prof-specialty	Not-in-family
16269	Married-civ-spouse	Handlers-cleaners	Husband
16270	Married-civ-spouse	Adm-clerical	Husband
16271	Married-civ-spouse	Sales	Husband
16272	Married-civ-spouse	Craft-repair	Husband
16273	Never-married	Other-service	Own-child
16274	Divorced	Other-service	Not-in-family
16275	Never-married	Prof-specialty	Own-child

```
16276           Divorced     Prof-specialty   Not-in-family
16278  Married-civ-spouse     Prof-specialty         Husband
16279           Divorced       Adm-clerical       Own-child
16280  Married-civ-spouse    Exec-managerial         Husband

                    race     sex  capital-gain  capital-loss  hours-per-week  \
0                  Black    Male             0             0              40
1                  White    Male             0             0              50
2                  White    Male             0             0              40
3                  Black    Male          7688             0              40
5                  White    Male             0             0              30
7                  White    Male          3103             0              32
8                  White  Female             0             0              40
9                  White    Male             0             0              10
10                 White    Male          6418             0              40
11                 White    Male             0             0              40
12                 White  Female             0             0              39
14                 White    Male          3103             0              48
15                 White    Male             0             0              50
16                 White    Male             0             0              25
17                 White  Female             0             0              30
18                 White  Female             0             0              20
20                 White    Male             0             0              47
21                 Black  Female             0             0              35
23                 White    Male             0             0              43
24                 White    Male             0             0              40
25                 White    Male          7298             0              90
26                 White    Male             0             0              20
27                 Black    Male             0             0              54
28                 White    Male             0             0              35
29                 White    Male             0             0              60
30                 Black    Male          7688             0              38
31                 White  Female             0             0              50
32                 White    Male             0             0              50
33                 White    Male             0             0              40
34                 White  Female             0             0              40
...                  ...     ...           ...           ...             ...
16248              Black    Male             0             0              40
16249              White    Male             0             0              40
16250              White    Male             0             0              40
16252              White    Male          7688             0              40
16253              White  Female             0          1669              40
16254              White    Male             0             0              50
16255              White    Male             0          1902              50
16256              White    Male             0             0              40
16257              White  Female             0             0              40
16258              White    Male             0             0              40
16259              White  Female             0             0               8
```

16260	White	Female	0	0	45
16261	White	Male	15024	0	55
16262	White	Male	0	0	40
16263	White	Male	0	0	40
16264	White	Male	0	0	32
16266	White	Female	0	0	32
16267	Black	Female	0	0	40
16268	White	Male	0	0	36
16269	White	Male	0	0	40
16270	White	Male	0	0	40
16271	White	Male	0	0	48
16272	White	Male	0	0	40
16273	White	Female	0	0	40
16274	White	Male	0	0	40
16275	White	Male	0	0	40
16276	White	Female	0	0	36
16278	White	Male	0	0	50
16279	Asian-Pac-Islander	Male	5455	0	40
16280	White	Male	0	0	60

	native-country	income
0	United-States	<=50K.
1	United-States	<=50K.
2	United-States	>50K.
3	United-States	>50K.
5	United-States	<=50K.
7	United-States	>50K.
8	United-States	<=50K.
9	United-States	<=50K.
10	United-States	>50K.
11	United-States	<=50K.
12	United-States	<=50K.
14	United-States	>50K.
15	United-States	>50K.
16	United-States	<=50K.
17	United-States	<=50K.
18	United-States	<=50K.
20	United-States	>50K.
21	United-States	<=50K.
23	Peru	<=50K.
24	United-States	<=50K.
25	United-States	>50K.
26	United-States	<=50K.
27	United-States	<=50K.
28	United-States	<=50K.
29	United-States	<=50K.
30	United-States	>50K.
31	United-States	<=50K.

```
32     United-States  <=50K.
33     United-States  <=50K.
34     United-States  <=50K.
...              ...     ...
16248  United-States  <=50K.
16249  United-States  <=50K.
16250         Canada   >50K.
16252  United-States   >50K.
16253  United-States  <=50K.
16254  United-States   >50K.
16255  United-States   >50K.
16256  United-States  <=50K.
16257  United-States  <=50K.
16258  United-States  <=50K.
16259  United-States  <=50K.
16260  United-States  <=50K.
16261  United-States   >50K.
16262  United-States  <=50K.
16263  United-States  <=50K.
16264  United-States  <=50K.
16266  United-States  <=50K.
16267  United-States  <=50K.
16268  United-States  <=50K.
16269  United-States  <=50K.
16270  United-States  <=50K.
16271  United-States  <=50K.
16272  United-States  <=50K.
16273  United-States  <=50K.
16274  United-States  <=50K.
16275  United-States  <=50K.
16276  United-States  <=50K.
16278  United-States  <=50K.
16279  United-States  <=50K.
16280  United-States   >50K.

[15060 rows x 15 columns]
```

In [3]:
```
from sklearn.preprocessing import LabelEncoder
# Hint: Now use a for loop over the elements in `le_category` and update df_le #TODO
encoder = LabelEncoder()

ohc_category = ['workclass', 'relationship'
                 ,'native-country','occupation']
le_category=['sex','income','education','race','marital-status']

df_ohc_train = pd.get_dummies(df_train,columns=ohc_category)
df_ohc_test = pd.get_dummies(df_test,columns=ohc_category)
```

```python
df_le_train = df_ohc_train.copy()
df_le_test = df_ohc_test.copy()
for i in range (len(le_category)):
    df_le_train[le_category[i]]  =encoder.fit_transform(df_le_train[le_category[i]])
    df_le_test[le_category[i]]  =encoder.fit_transform(df_le_test[le_category[i]])

df_le_test.head(6)
```

Out[3]:

```
   age  fnlwgt  education  education-num  marital-status  race  sex  \
0   25  226802          1              7               4     2    1
1   38   89814         11              9               2     4    1
2   28  336951          7             12               2     4    1
3   44  160323         15             10               2     2    1
4   18  103497         15             10               4     4    0
5   34  198693          0              6               4     4    1

   capital-gain  capital-loss  hours-per-week             ...                \
0             0             0              40             ...
1             0             0              50             ...
2             0             0              40             ...
3          7688             0              40             ...
4             0             0              30             ...
5             0             0              30             ...

   occupation_Farming-fishing  occupation_Handlers-cleaners  \
0                           0                             0
1                           1                             0
2                           0                             0
3                           0                             0
4                           0                             0
5                           0                             0

   occupation_Machine-op-inspct  occupation_Other-service  \
0                             1                         0
1                             0                         0
2                             0                         0
3                             1                         0
4                             0                         0
5                             0                         1

   occupation_Priv-house-serv  occupation_Prof-specialty  \
0                           0                          0
1                           0                          0
2                           0                          0
3                           0                          0
4                           0                          0
5                           0                          0
```

```
   occupation_Protective-serv  occupation_Sales  occupation_Tech-support  \
0                           0                 0                        0
1                           0                 0                        0
2                           1                 0                        0
3                           0                 0                        0
4                           0                 0                        0
5                           0                 0                        0

   occupation_Transport-moving
0                            0
1                            0
2                            0
3                            0
4                            0
5                            0

[6 rows x 79 columns]
```

In [4]:
```
X_df_train = np.array(df_le_train.drop(['income'],axis=1))
y_train = np.array(df_le_train['income'])
X_df_test = np.array(df_le_test.drop(['income'],axis=1))
y_test = np.array(df_le_test['income'])

nsamples,nfeatures=X_df_train.shape
print (' number of Train samples:{0} and number of features :{1}'.format(nsamples,nfea
nsamples,nfeatures=X_df_test.shape
print (' number of Test samples:{0} and number of features :{1}'.format(nsamples,nfeatu
```

```
 number of Train samples:32560 and number of features :78
 number of Test samples:16281 and number of features :78
```

In [5]:
```
Xtr_mean = np.mean(X_df_train,axis=0)
Xtr_std = np.std(X_df_train,axis=0)
Xtr_scale = (X_df_train-Xtr_mean)/Xtr_std[None,:]
Xts_scale = (X_df_test-Xtr_mean[None,:])/Xtr_std[None,:]
```

In [6]:
```
from sklearn import svm
C_test= [0.1,0.5,1]
gam_test = [0.005,0.01,0.05,0.1]
nC = len(C_test)
ngam = len(gam_test)
acc = np.zeros((nC,ngam))
for i,c in enumerate(C_test):
    for j,g in enumerate(gam_test):
        svc = svm.SVC(probability=False,  kernel="rbf", C=c, gamma=g,verbose=11)
        svc.fit(Xtr_scale,y_train)
```

```
                yhat_ts = svc.predict(Xts_scale)
                acc[i,j]=np.mean(yhat_ts == y_test)
```

```
[LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVI
```

```
In [7]: print(acc)
```

```
[[0.84343714 0.84503409 0.8381549  0.82740618]
 [0.85123764 0.8522818  0.84613967 0.84024323]
 [0.85412444 0.85510718 0.84810515 0.84294577]]
```

```
In [9]: position=np.argmax(acc)
        m, n = divmod(position, 4)
        print('greatest accuracy using RBF is',acc[m,n])
        print('optimal C is',C_test[m])
        print('optimal gamma is',gam_test[n])
```

```
greatest accuracy is 0.8551071801486395
optimal C is 1
optimal gamma is 0.01
```

```
In [11]: for i,c in enumerate(C_test):
             for j,g in enumerate(gam_test):
                 svc = svm.SVC(probability=False,  kernel="sigmoid", C=c, gamma=g,verbose=11)
                 svc.fit(Xtr_scale,y_train)
                 yhat_ts = svc.predict(Xts_scale)
                 acc[i,j]=np.mean(yhat_ts == y_test)
```

```
[LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVM][LibSVI
```

```
In [12]: print(acc)
```

```
[[0.84626251 0.84847368 0.78342854 0.76303667]
 [0.85191327 0.84681531 0.77808488 0.77747067]
 [0.85166759 0.8352681  0.77857625 0.77089859]]
```

```
In [13]: position=np.argmax(acc)
         m, n = divmod(position, 4)
         print('greatest accuracy using sigmoid is',acc[m,n])
         print('optimal C is',C_test[m])
         print('optimal gamma is',gam_test[n])
```

```
greatest accuracy using sigmoid is 0.8519132731404705
optimal C is 0.5
optimal gamma is 0.005
```

In [ ]: The SVM using RBF kernel, C=1, gamma=0.01 yields highest accuracy.

In [14]:
```
svc = svm.SVC(probability=False,  kernel="rbf", C=1, gamma=0.01,verbose=11)
svc.fit(Xtr_scale,y_train)
yhat_ts = svc.predict(Xts_scale)
acc1= np.mean(yhat_ts == y_test)
print('Accuaracy = {0:f}%'.format(100*acc1))
l=0
m=0
n=0
for i in range(y_test.shape[0]):
    if y_test[i]==0:
        l+=1
        if yhat_ts[i]==1:
            m+=1
print('{0:f}% of the people earn less than 50ks are classified as those earn more thar
print('{0:d} people earn less than 50ks in total in test data'. format(l))
l=0
for i in range(y_test.shape[0]):
    if y_test[i]==1:
        l+=1
        if yhat_ts[i]==0:
            n+=1
print('{0:f}% of the people earn more than 50ks are classified as those earn less thar
print('{0:d} people earn more than 50ks in total in test data'. format(l))
```

```
[LibSVM]Accuaracy = 85.510718%
6.151990% of the people earn less than 50ks are classified as those earn more than 50ks
12435 people earn less than 50ks in total in test data
41.445658% of the people earn more than 50ks are classified as those earn less than 50ks
3846 people earn more than 50ks in total in test data
```