naive-bayes

November 18, 2024

[59]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

[60]:
```
data = 'C:\\Users\\ANUSHA\\Downloads\\adult.csv\\Naive Bayes.csv'
df = pd.read_csv(data, header=None, sep=',\s')
```

[61]:
```
df.head()
```

[61]:

	0	1	2	3	4	5	\
0	39	State-gov	77516	Bachelors	13	Never-married	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	
2	38	Private	215646	HS-grad	9	Divorced	
3	53	Private	234721	11th	7	Married-civ-spouse	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	

	6	7	8	9	10	11	12	\
0	Adm-clerical	Not-in-family	White	Male	2174	0	40	
1	Exec-managerial	Husband	White	Male	0	0	13	
2	Handlers-cleaners	Not-in-family	White	Male	0	0	40	
3	Handlers-cleaners	Husband	Black	Male	0	0	40	
4	Prof-specialty	Wife	Black	Female	0	0	40	

	13	14
0	United-States	<=50K
1	United-States	<=50K
2	United-States	<=50K
3	United-States	<=50K
4	Cuba	<=50K

[62]:
```
df.shape
```

[62]:
```
(32561, 15)
```

[63]:
```
col_names = ['age', 'workclass', 'fnlwgt', 'education', 'education_num',␣
 ↪'marital_status', 'occupation', 'relationship',
            'race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week',␣
 ↪'native_country', 'income']

df.columns = col_names
df.columns
```

[63]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education_num',
       'marital_status', 'occupation', 'relationship', 'race', 'sex',
       'capital_gain', 'capital_loss', 'hours_per_week', 'native_country',
       'income'],
      dtype='object')
```

[64]:
```
df.head()
```

[64]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	income
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

[65]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
```

```
 3   education       32561 non-null  object
 4   education_num   32561 non-null  int64
 5   marital_status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital_gain    32561 non-null  int64
 11  capital_loss    32561 non-null  int64
 12  hours_per_week  32561 non-null  int64
 13  native_country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

[66]:
```
categorical = [var for var in df.columns if df[var].dtype=='O']

print('There are {} categorical variables\n'.format(len(categorical)))
print('The categorical variables are :\n\n', categorical)
df[categorical].head()
```

```
There are 9 categorical variables

The categorical variables are :

 ['workclass', 'education', 'marital_status', 'occupation', 'relationship',
'race', 'sex', 'native_country', 'income']
```

[66]:

	workclass	education	marital_status	occupation	relationship	race	sex	native_country	income
0	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	United-States	<=50K
1	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-States	<=50K
2	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	United-States	<=50K
3	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	United-States	<=50K
4	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	Cuba	<=50K

[67]:
```
df[categorical].isnull().sum()
```

[67]:
```
workclass         0
education         0
marital_status    0
occupation        0
```

```
relationship      0
race              0
sex               0
native_country    0
income            0
dtype: int64
```

[68]:
```
for var in categorical:

    print(df[var].value_counts())
```

```
Private             22696
Self-emp-not-inc     2541
Local-gov            2093
?                    1836
State-gov            1298
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: workclass, dtype: int64
HS-grad         10501
Some-college     7291
Bachelors        5355
Masters          1723
Assoc-voc        1382
11th             1175
Assoc-acdm       1067
10th              933
7th-8th           646
Prof-school       576
9th               514
12th              433
Doctorate         413
5th-6th           333
1st-4th           168
Preschool          51
Name: education, dtype: int64
Married-civ-spouse       14976
Never-married            10683
Divorced                  4443
Separated                 1025
Widowed                    993
Married-spouse-absent      418
Married-AF-spouse           23
Name: marital_status, dtype: int64
Prof-specialty       4140
```

```
Craft-repair         4099
Exec-managerial      4066
Adm-clerical         3770
Sales                3650
Other-service        3295
Machine-op-inspct    2002
?                    1843
Transport-moving     1597
Handlers-cleaners    1370
Farming-fishing       994
Tech-support          928
Protective-serv       649
Priv-house-serv       149
Armed-Forces            9
Name: occupation, dtype: int64
Husband           13193
Not-in-family      8305
Own-child          5068
Unmarried          3446
Wife               1568
Other-relative      981
Name: relationship, dtype: int64
White                27816
Black                 3124
Asian-Pac-Islander    1039
Amer-Indian-Eskimo     311
Other                  271
Name: race, dtype: int64
Male      21790
Female    10771
Name: sex, dtype: int64
United-States                29170
Mexico                         643
?                              583
Philippines                    198
Germany                        137
Canada                         121
Puerto-Rico                    114
El-Salvador                    106
India                          100
Cuba                            95
England                         90
Jamaica                         81
South                           80
China                           75
Italy                           73
Dominican-Republic              70
Vietnam                         67
```

```
Guatemala                      64
Japan                          62
Poland                         60
Columbia                       59
Taiwan                         51
Haiti                          44
Iran                           43
Portugal                       37
Nicaragua                      34
Peru                           31
France                         29
Greece                         29
Ecuador                        28
Ireland                        24
Hong                           20
Cambodia                       19
Trinadad&Tobago                19
Laos                           18
Thailand                       18
Yugoslavia                     16
Outlying-US(Guam-USVI-etc)     14
Honduras                       13
Hungary                        13
Scotland                       12
Holand-Netherlands              1
Name: native_country, dtype: int64
<=50K    24720
>50K      7841
Name: income, dtype: int64
```

[69]:
```
for var in categorical:

    print(df[var].value_counts()/np.float(len(df)))
```

```
Private             0.697030
Self-emp-not-inc    0.078038
Local-gov           0.064279
?                   0.056386
State-gov           0.039864
Self-emp-inc        0.034274
Federal-gov         0.029483
Without-pay         0.000430
Never-worked        0.000215
Name: workclass, dtype: float64
HS-grad         0.322502
Some-college    0.223918
Bachelors       0.164461
Masters         0.052916
```

```
Assoc-voc       0.042443
11th            0.036086
Assoc-acdm      0.032769
10th            0.028654
7th-8th         0.019840
Prof-school     0.017690
9th             0.015786
12th            0.013298
Doctorate       0.012684
5th-6th         0.010227
1st-4th         0.005160
Preschool       0.001566
Name: education, dtype: float64
Married-civ-spouse       0.459937
Never-married            0.328092
Divorced                 0.136452
Separated                0.031479
Widowed                  0.030497
Married-spouse-absent    0.012837
Married-AF-spouse        0.000706
Name: marital_status, dtype: float64
Prof-specialty       0.127146
Craft-repair         0.125887
Exec-managerial      0.124873
Adm-clerical         0.115783
Sales                0.112097
Other-service        0.101195
Machine-op-inspct    0.061485
?                    0.056601
Transport-moving     0.049046
Handlers-cleaners    0.042075
Farming-fishing      0.030527
Tech-support         0.028500
Protective-serv      0.019932
Priv-house-serv      0.004576
Armed-Forces         0.000276
Name: occupation, dtype: float64
Husband           0.405178
Not-in-family     0.255060
Own-child         0.155646
Unmarried         0.105832
Wife              0.048156
Other-relative    0.030128
Name: relationship, dtype: float64
White                 0.854274
Black                 0.095943
Asian-Pac-Islander    0.031909
Amer-Indian-Eskimo    0.009551
```

```
Other                 0.008323
Name: race, dtype: float64
Male      0.669205
Female    0.330795
Name: sex, dtype: float64
United-States                 0.895857
Mexico                        0.019748
?                             0.017905
Philippines                   0.006081
Germany                       0.004207
Canada                        0.003716
Puerto-Rico                   0.003501
El-Salvador                   0.003255
India                         0.003071
Cuba                          0.002918
England                       0.002764
Jamaica                       0.002488
South                         0.002457
China                         0.002303
Italy                         0.002242
Dominican-Republic            0.002150
Vietnam                       0.002058
Guatemala                     0.001966
Japan                         0.001904
Poland                        0.001843
Columbia                      0.001812
Taiwan                        0.001566
Haiti                         0.001351
Iran                          0.001321
Portugal                      0.001136
Nicaragua                     0.001044
Peru                          0.000952
France                        0.000891
Greece                        0.000891
Ecuador                       0.000860
Ireland                       0.000737
Hong                          0.000614
Cambodia                      0.000584
Trinadad&Tobago               0.000584
Laos                          0.000553
Thailand                      0.000553
Yugoslavia                    0.000491
Outlying-US(Guam-USVI-etc)    0.000430
Honduras                      0.000399
Hungary                       0.000399
Scotland                      0.000369
Holand-Netherlands            0.000031
Name: native_country, dtype: float64
```

```
<=50K    0.75919
>50K     0.24081
Name: income, dtype: float64
```

[70]:
```
df.workclass.unique()
df.workclass.value_counts()
df['workclass'].replace('?', np.NaN, inplace=True)
df.workclass.value_counts()
```

[70]:
```
Private             22696
Self-emp-not-inc     2541
Local-gov            2093
State-gov            1298
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: workclass, dtype: int64
```

[71]:
```
df.occupation.unique()
df.occupation.value_counts()
df['occupation'].replace('?', np.NaN, inplace=True)
df.occupation.value_counts()
```

[71]:
```
Prof-specialty       4140
Craft-repair         4099
Exec-managerial      4066
Adm-clerical         3770
Sales                3650
Other-service        3295
Machine-op-inspct    2002
Transport-moving     1597
Handlers-cleaners    1370
Farming-fishing       994
Tech-support          928
Protective-serv       649
Priv-house-serv       149
Armed-Forces            9
Name: occupation, dtype: int64
```

[72]:
```
df.native_country.unique()
df.native_country.value_counts()
df['native_country'].replace('?', np.NaN, inplace=True)
df.native_country.value_counts()
```

[72]:
```
United-States                 29170
Mexico                          643
```

```
Philippines                    198
Germany                        137
Canada                         121
Puerto-Rico                    114
El-Salvador                    106
India                          100
Cuba                            95
England                         90
Jamaica                         81
South                           80
China                           75
Italy                           73
Dominican-Republic              70
Vietnam                         67
Guatemala                       64
Japan                           62
Poland                          60
Columbia                        59
Taiwan                          51
Haiti                           44
Iran                            43
Portugal                        37
Nicaragua                       34
Peru                            31
France                          29
Greece                          29
Ecuador                         28
Ireland                         24
Hong                            20
Cambodia                        19
Trinadad&Tobago                 19
Laos                            18
Thailand                        18
Yugoslavia                      16
Outlying-US(Guam-USVI-etc)      14
Honduras                        13
Hungary                         13
Scotland                        12
Holand-Netherlands               1
Name: native_country, dtype: int64
```

[73]:
```
df[categorical].isnull().sum()
```

[73]:
```
workclass         1836
education            0
marital_status       0
occupation        1843
```

```
relationship         0
race                 0
sex                  0
native_country     583
income               0
dtype: int64
```

[74]:
```
for var in categorical:

    print(var, ' contains ', len(df[var].unique()), ' labels')
```

```
workclass  contains  9  labels
education  contains  16  labels
marital_status  contains  7  labels
occupation  contains  15  labels
relationship  contains  6  labels
race  contains  5  labels
sex  contains  2  labels
native_country  contains  42  labels
income  contains  2  labels
```

[75]:
```
numerical = [var for var in df.columns if df[var].dtype!='O']

print('There are {} numerical variables\n'.format(len(numerical)))

print('The numerical variables are :', numerical)

df[numerical].head()
```

```
There are 6 numerical variables

The numerical variables are : ['age', 'fnlwgt', 'education_num', 'capital_gain',
'capital_loss', 'hours_per_week']
```

[75]:

	age	fnlwgt	education_num	capital_gain	capital_loss	hours_per_week
0	39	77516	13	2174	0	40
1	50	83311	13	0	0	13
2	38	215646	9	0	0	40
3	53	234721	7	0	0	40
4	28	338409	13	0	0	40

[76]:
```
df[numerical].isnull().sum()
```

[76]:
```
age              0
fnlwgt           0
education_num    0
capital_gain     0
```

```
capital_loss      0
hours_per_week    0
dtype: int64
```

[77]:
```
X = df.drop(['income'], axis=1)

y = df['income']
```

[78]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,␣
 ↪random_state = 0)
```

[79]:
```
X_train.shape, X_test.shape
```

[79]:
```
((22792, 14), (9769, 14))
```

[80]:
```
X_train.dtypes
```

[80]:
```
age               int64
workclass        object
fnlwgt            int64
education        object
education_num     int64
marital_status   object
occupation       object
relationship     object
race             object
sex              object
capital_gain      int64
capital_loss      int64
hours_per_week    int64
native_country   object
dtype: object
```

[81]:
```
categorical = [col for col in X_train.columns if X_train[col].dtypes == 'O']

categorical
```

[81]:
```
['workclass',
 'education',
 'marital_status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native_country']
```

[82]:
```
numerical = [col for col in X_train.columns if X_train[col].dtypes != 'O']

numerical
```

[82]:
```
['age',
 'fnlwgt',
 'education_num',
 'capital_gain',
 'capital_loss',
 'hours_per_week']
```

[83]:
```
X_train[categorical].isnull().mean()
```

[83]:
```
workclass         0.055985
education         0.000000
marital_status    0.000000
occupation        0.056072
relationship      0.000000
race              0.000000
sex               0.000000
native_country    0.018164
dtype: float64
```

[84]:
```
for col in categorical:
    if X_train[col].isnull().mean()>0:
        print(col, (X_train[col].isnull().mean()))
```

```
workclass 0.055984555984555984
occupation 0.05607230607230607
native_country 0.018164268164268166
```

[85]:
```
for df2 in [X_train, X_test]:
    df2['workclass'].fillna(X_train['workclass'].mode()[0], inplace=True)
    df2['occupation'].fillna(X_train['occupation'].mode()[0], inplace=True)
    df2['native_country'].fillna(X_train['native_country'].mode()[0],␣
 ↪inplace=True)
```

[86]:
```
X_train[categorical].isnull().sum()
X_test[categorical].isnull().sum()
X_train.isnull().sum()
X_test.isnull().sum()
```

[86]:
```
age               0
workclass         0
fnlwgt            0
education         0
education_num     0
```

```
marital_status    0
occupation        0
relationship      0
race              0
sex               0
capital_gain      0
capital_loss      0
hours_per_week    0
native_country    0
dtype: int64
```

[87]:
```
categorical
```

[87]:
```
['workclass',
 'education',
 'marital_status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native_country']
```

[88]:
```
X_train[categorical].head()
```

[88]:

	workclass	education	marital_status	occupation	\
32098	Private	HS-grad	Married-civ-spouse	Craft-repair	
25206	State-gov	HS-grad	Divorced	Adm-clerical	
23491	Private	Some-college	Married-civ-spouse	Sales	
12367	Private	HS-grad	Never-married	Craft-repair	
7054	Private	7th-8th	Never-married	Craft-repair	

	relationship	race	sex	native_country
32098	Husband	White	Male	United-States
25206	Unmarried	White	Female	United-States
23491	Husband	White	Male	United-States
12367	Not-in-family	White	Male	Guatemala
7054	Not-in-family	White	Male	Germany

[89]:
```
!pip install category_encoders
import category_encoders as ce
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: category_encoders in
c:\users\anusha\appdata\roaming\python\python39\site-packages (2.6.4)
Requirement already satisfied: pandas>=1.0.5 in
c:\programdata\anaconda3\lib\site-packages (from category_encoders) (1.4.2)
Requirement already satisfied: scipy>=1.0.0 in
```

```
c:\programdata\anaconda3\lib\site-packages (from category_encoders) (1.7.3)
Requirement already satisfied: numpy>=1.14.0 in
c:\programdata\anaconda3\lib\site-packages (from category_encoders) (1.21.5)
Requirement already satisfied: statsmodels>=0.9.0 in
c:\programdata\anaconda3\lib\site-packages (from category_encoders) (0.13.2)
Requirement already satisfied: scikit-learn>=0.20.0 in
c:\programdata\anaconda3\lib\site-packages (from category_encoders) (1.0.2)
Requirement already satisfied: patsy>=0.5.1 in
c:\programdata\anaconda3\lib\site-packages (from category_encoders) (0.5.2)
Requirement already satisfied: pytz>=2020.1 in
c:\programdata\anaconda3\lib\site-packages (from
pandas>=1.0.5->category_encoders) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\programdata\anaconda3\lib\site-packages (from
pandas>=1.0.5->category_encoders) (2.8.2)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages
(from patsy>=0.5.1->category_encoders) (1.16.0)
Requirement already satisfied: joblib>=0.11 in
c:\users\anusha\appdata\roaming\python\python39\site-packages (from scikit-
learn>=0.20.0->category_encoders) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\programdata\anaconda3\lib\site-packages (from scikit-
learn>=0.20.0->category_encoders) (2.2.0)
Requirement already satisfied: packaging>=21.3 in
c:\programdata\anaconda3\lib\site-packages (from
statsmodels>=0.9.0->category_encoders) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
c:\programdata\anaconda3\lib\site-packages (from
packaging>=21.3->statsmodels>=0.9.0->category_encoders) (3.0.4)
```

[90]:
```
encoder = ce.OneHotEncoder(cols=['workclass', 'education', 'marital_status',␣
 ↪'occupation',
                                 'relationship', 'race', 'sex',␣
 ↪'native_country'])
X_train = encoder.fit_transform(X_train)


X_test = encoder.transform(X_test)
X_train.head()
```

[90]:

	age	workclass_1	workclass_2	workclass_3	workclass_4	workclass_5	\
32098	45	1	0	0	0	0	
25206	47	0	1	0	0	0	
23491	48	1	0	0	0	0	
12367	29	1	0	0	0	0	
7054	23	1	0	0	0	0	

```
       workclass_6  workclass_7  workclass_8  fnlwgt  …  native_country_32  \
32098            0            0            0  170871  …                  0
25206            0            0            0  108890  …                  0
23491            0            0            0  187505  …                  0
12367            0            0            0  145592  …                  0
7054             0            0            0  203003  …                  0

       native_country_33  native_country_34  native_country_35  \
32098                  0                  0                  0
25206                  0                  0                  0
23491                  0                  0                  0
12367                  0                  0                  0
7054                   0                  0                  0

       native_country_36  native_country_37  native_country_38  \
32098                  0                  0                  0
25206                  0                  0                  0
23491                  0                  0                  0
12367                  0                  0                  0
7054                   0                  0                  0

       native_country_39  native_country_40  native_country_41
32098                  0                  0                  0
25206                  0                  0                  0
23491                  0                  0                  0
12367                  0                  0                  0
7054                   0                  0                  0

[5 rows x 105 columns]
```

[91]:
```
X_train.shape
```

[91]:
```
(22792, 105)
```

[92]:
```
X_test.head()
```

[92]:
```
       age  workclass_1  workclass_2  workclass_3  workclass_4  workclass_5  \
22278   27            1            0            0            0            0
8950    27            1            0            0            0            0
7838    25            1            0            0            0            0
16505   46            1            0            0            0            0
19140   45            1            0            0            0            0

       workclass_6  workclass_7  workclass_8  fnlwgt  …  native_country_32  \
22278            0            0            0  177119  …                  0
8950             0            0            0  216481  …                  0
7838             0            0            0  256263  …                  0
```

```
16505            0            0            0  147640  …                  0
19140            0            0            0  172822  …                  0

       native_country_33  native_country_34  native_country_35  \
22278                  0                  0                  0
8950                   0                  0                  0
7838                   0                  0                  0
16505                  0                  0                  0
19140                  0                  0                  0

       native_country_36  native_country_37  native_country_38  \
22278                  0                  0                  0
8950                   0                  0                  0
7838                   0                  0                  0
16505                  0                  0                  0
19140                  0                  0                  0

       native_country_39  native_country_40  native_country_41
22278                  0                  0                  0
8950                   0                  0                  0
7838                   0                  0                  0
16505                  0                  0                  0
19140                  0                  0                  0

[5 rows x 105 columns]
```

[93]:
```
X_test.shape
```

[93]:
```
(9769, 105)
```

[94]:
```
cols = X_train.columns
```

[95]:
```
from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
```

[96]:
```
X_train = pd.DataFrame(X_train, columns=[cols])
```

[97]:
```
X_test = pd.DataFrame(X_test, columns=[cols])
```

[98]:
```
X_train.head()
```

[98]:
```
     age workclass_1 workclass_2 workclass_3 workclass_4 workclass_5  \
0  0.40         0.0         0.0         0.0         0.0         0.0
1  0.50        -1.0         1.0         0.0         0.0         0.0
2  0.55         0.0         0.0         0.0         0.0         0.0
3 -0.40         0.0         0.0         0.0         0.0         0.0
4 -0.70         0.0         0.0         0.0         0.0         0.0

  workclass_6 workclass_7 workclass_8    fnlwgt  … native_country_32  \
0         0.0         0.0         0.0 -0.058906  …                0.0
1         0.0         0.0         0.0 -0.578076  …                0.0
2         0.0         0.0         0.0  0.080425  …                0.0
3         0.0         0.0         0.0 -0.270650  …                0.0
4         0.0         0.0         0.0  0.210240  …                0.0

  native_country_33 native_country_34 native_country_35 native_country_36  \
0               0.0               0.0               0.0               0.0
1               0.0               0.0               0.0               0.0
2               0.0               0.0               0.0               0.0
3               0.0               0.0               0.0               0.0
4               0.0               0.0               0.0               0.0

  native_country_37 native_country_38 native_country_39 native_country_40  \
0               0.0               0.0               0.0               0.0
1               0.0               0.0               0.0               0.0
2               0.0               0.0               0.0               0.0
3               0.0               0.0               0.0               0.0
4               0.0               0.0               0.0               0.0

  native_country_41
0               0.0
1               0.0
2               0.0
3               0.0
4               0.0

[5 rows x 105 columns]
```

[99]:
```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
```

[99]:
```
GaussianNB()
```

[100]:
```
y_pred = gnb.predict(X_test)

y_pred
```

[100]:
```
array(['<=50K', '<=50K', '>50K', …, '>50K', '<=50K', '<=50K'],
      dtype='<U5')
```

[101]:
```
from sklearn.metrics import accuracy_score

print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, y_pred)))
```

```
Model accuracy score: 0.8083
```

[102]:
```
y_pred_train = gnb.predict(X_train)

y_pred_train
```

[102]:
```
array(['>50K', '<=50K', '>50K', …, '<=50K', '>50K', '<=50K'],
      dtype='<U5')
```

[103]:
```
print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(y_train,␣
 ↪y_pred_train)))
```

```
Training-set accuracy score: 0.8067
```

[104]:
```
print('Training set score: {:.4f}'.format(gnb.score(X_train, y_train)))

print('Test set score: {:.4f}'.format(gnb.score(X_test, y_test)))
```

```
Training set score: 0.8067
Test set score: 0.8083
```

[105]:
```
y_test.value_counts()
```

[105]:
```
<=50K    7407
>50K     2362
Name: income, dtype: int64
```

[106]:
```
null_accuracy = (7407/(7407+2362))

print('Null accuracy score: {0:0.4f}'. format(null_accuracy))
```

```
Null accuracy score: 0.7582
```

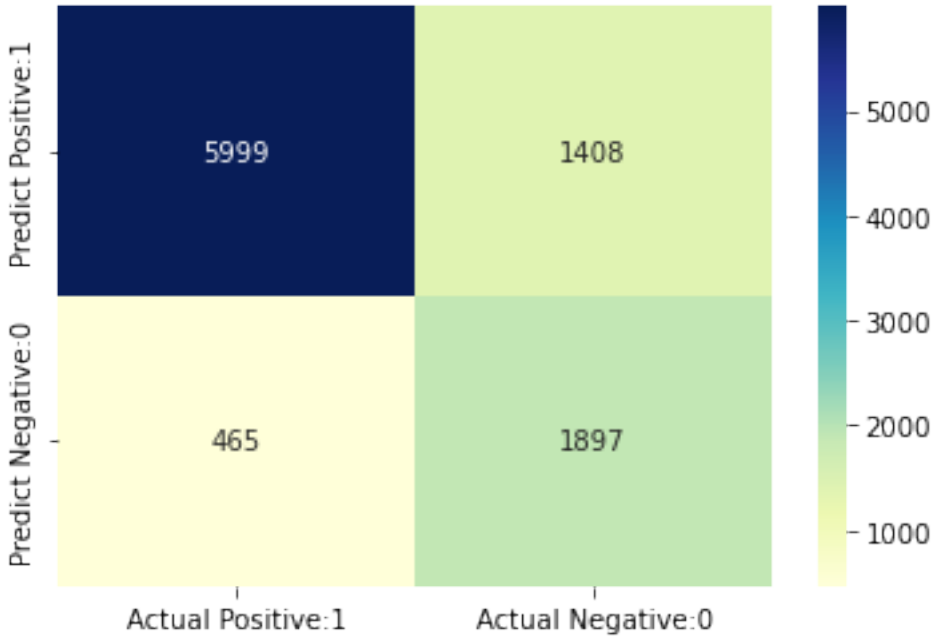
[107]:
```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)

print('Confusion matrix\n\n', cm)

print('\nTrue Positives(TP) = ', cm[0,0])
```

```
print('\nTrue Negatives(TN) = ', cm[1,1])

print('\nFalse Positives(FP) = ', cm[0,1])

print('\nFalse Negatives(FN) = ', cm[1,0])
```

```
Confusion matrix

 [[5999 1408]
 [ 465 1897]]

True Positives(TP) =  5999

True Negatives(TN) =  1897

False Positives(FP) =  1408

False Negatives(FN) =  465
```

[108]:
```
cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual␣
↪Negative:0'],
                                 index=['Predict Positive:1', 'Predict Negative:
↪0'])

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
```

[108]: <AxesSubplot:>



[109]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       <=50K       0.93      0.81      0.86      7407
        >50K       0.57      0.80      0.67      2362

    accuracy                           0.81      9769
   macro avg       0.75      0.81      0.77      9769
weighted avg       0.84      0.81      0.82      9769
```

[110]:
```
TP = cm[0,0]
TN = cm[1,1]
FP = cm[0,1]
FN = cm[1,0]
```

[111]:
```
classification_accuracy = (TP + TN) / float(TP + TN + FP + FN)

print('Classification accuracy : {0:0.4f}'.format(classification_accuracy))
```

```
Classification accuracy : 0.8083
```

[112]:
```
classification_error = (FP + FN) / float(TP + TN + FP + FN)

print('Classification error : {0:0.4f}'.format(classification_error))
```

```
Classification error : 0.1917
```

[113]:
```
precision = TP / float(TP + FP)


print('Precision : {0:0.4f}'.format(precision))
```

```
Precision : 0.8099
```

[114]:
```
recall = TP / float(TP + FN)

print('Recall or Sensitivity : {0:0.4f}'.format(recall))
```

```
Recall or Sensitivity : 0.9281
```

[115]:
```
true_positive_rate = TP / float(TP + FN)
```

```
print('True Positive Rate : {0:0.4f}'.format(true_positive_rate))
```

```
True Positive Rate : 0.9281
```

[116]:
```
false_positive_rate = FP / float(FP + TN)


print('False Positive Rate : {0:0.4f}'.format(false_positive_rate))
```

```
False Positive Rate : 0.4260
```

[117]:
```
specificity = TN / (TN + FP)

print('Specificity : {0:0.4f}'.format(specificity))
```

```
Specificity : 0.5740
```

[118]:
```
y_pred_prob = gnb.predict_proba(X_test)[0:10]

y_pred_prob
```

[118]:
```
array([[9.99999426e-01, 5.74152436e-07],
       [9.99687907e-01, 3.12093456e-04],
       [1.54405602e-01, 8.45594398e-01],
       [1.73624321e-04, 9.99826376e-01],
       [8.20121011e-09, 9.99999992e-01],
       [8.76844580e-01, 1.23155420e-01],
       [9.99999927e-01, 7.32876705e-08],
       [9.99993460e-01, 6.53998797e-06],
       [9.87738143e-01, 1.22618575e-02],
       [9.99999996e-01, 4.01886317e-09]])
```

[119]:
```
y_pred_prob_df = pd.DataFrame(data=y_pred_prob, columns=['Prob of - <=50K',␣
↪'Prob of - >50K'])

y_pred_prob_df
```

[119]:

	Prob of - <=50K	Prob of - >50K
0	9.999994e-01	5.741524e-07
1	9.996879e-01	3.120935e-04
2	1.544056e-01	8.455944e-01
3	1.736243e-04	9.998264e-01
4	8.201210e-09	1.000000e+00
5	8.768446e-01	1.231554e-01
6	9.999999e-01	7.328767e-08
7	9.999935e-01	6.539988e-06

```
8     9.877381e-01    1.226186e-02
9     1.000000e+00    4.018863e-09
```

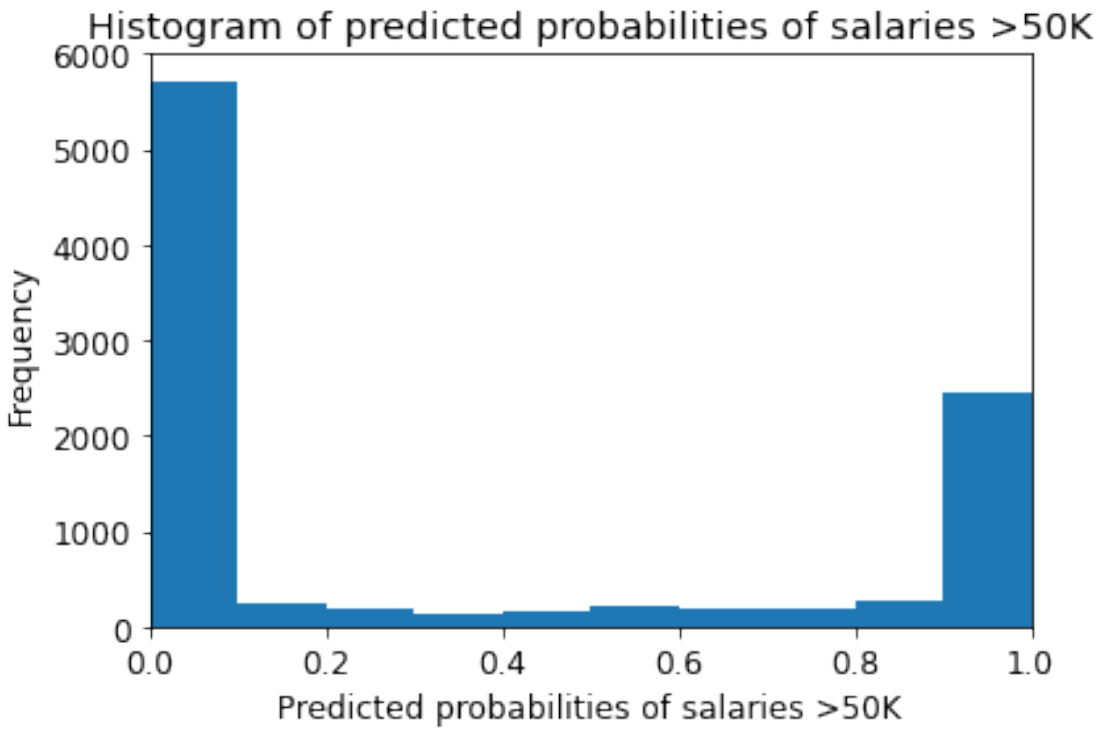
[120]:
```
gnb.predict_proba(X_test)[0:10, 1]
```

[120]:
```
array([5.74152436e-07, 3.12093456e-04, 8.45594398e-01, 9.99826376e-01,
       9.99999992e-01, 1.23155420e-01, 7.32876705e-08, 6.53998797e-06,
       1.22618575e-02, 4.01886317e-09])
```

[121]:
```
y_pred1 = gnb.predict_proba(X_test)[:, 1]
```

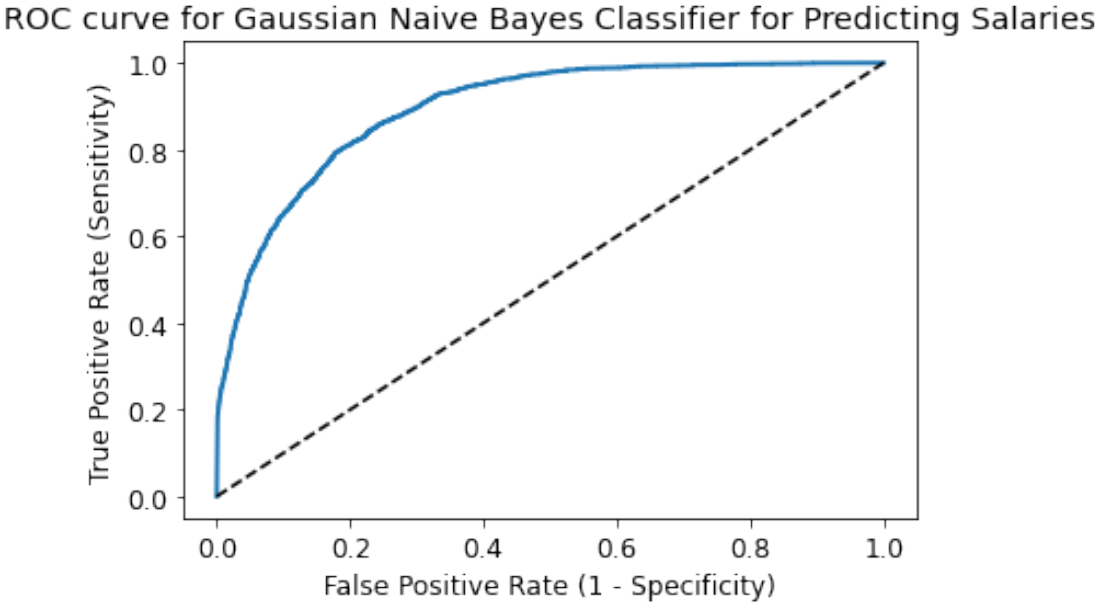
[122]:
```
plt.rcParams['font.size'] = 12
plt.hist(y_pred1, bins = 10)
plt.title('Histogram of predicted probabilities of salaries >50K')
plt.xlim(0,1)
plt.xlabel('Predicted probabilities of salaries >50K')
plt.ylabel('Frequency')
```

[122]:
```
Text(0, 0.5, 'Frequency')
```



[123]:
```
from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_pred1, pos_label = '>50K')
```

```
plt.figure(figsize=(6,4))

plt.plot(fpr, tpr, linewidth=2)

plt.plot([0,1], [0,1], 'k--' )

plt.rcParams['font.size'] = 12

plt.title('ROC curve for Gaussian Naive Bayes Classifier for Predicting␣
 ↪Salaries')

plt.xlabel('False Positive Rate (1 - Specificity)')

plt.ylabel('True Positive Rate (Sensitivity)')

plt.show()
```



[ ]: