In [1]:
```
import pandas as pd
from pandas.plotting import scatter_matrix
import numpy as np
from numpy import percentile
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
data =pd.read_csv("census_income.csv")
data.head()
```

Out[2]:

	Age	Workclass	Fnlwgt	Education	Education_num	Marital_status	Occupation	Relationship	Race	Sex	Capital_gain	Capital_loss	Hours_per_week	Native_country	Income
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
2	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
3	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
4	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<=50K

In [3]:
```
data.shape
```

Out[3]:
```
(32560, 15)
```

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32560 entries, 0 to 32559
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Age             32560 non-null  int64 
 1   Workclass       32560 non-null  object
 2   Fnlwgt          32560 non-null  int64 
 3   Education       32560 non-null  object
 4   Education_num   32560 non-null  int64 
 5   Marital_status  32560 non-null  object
 6   Occupation      32560 non-null  object
 7   Relationship    32560 non-null  object
 8   Race            32560 non-null  object
 9   Sex             32560 non-null  object
 10  Capital_gain    32560 non-null  int64 
 11  Capital_loss    32560 non-null  int64 
 12  Hours_per_week  32560 non-null  int64 
 13  Native_country  32560 non-null  object
 14  Income          32560 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

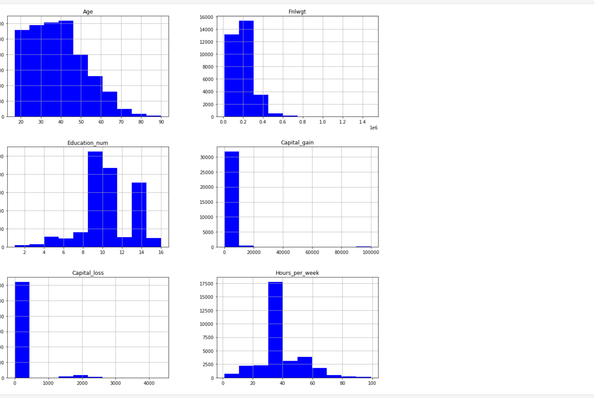
In [5]:
```
data.isnull().sum()
```

Out[5]:
```
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
Income            0
dtype: int64
```

In [6]:
```
data.value_counts()
```

Out[6]:
```
Age  Workclass         Fnlwgt  Education     Education_num  Marital_status       Occupation          Relationship    Race   Sex     Capital_gain  Capital_loss  Hours_per_week  Native_c
ountry  Income
25   Private           195994  1st-4th       2              Never-married       Priv-house-serv     Not-in-family   White  Female  0             0             40              Guatema
la      <=50K     3
23   Private           240137  5th-6th       3              Never-married       Handlers-cleaners   Not-in-family   White  Male    0             0             55              Mexico
<=50K     2
38   Private           207202  HS-grad       9              Married-civ-spouse  Machine-op-inspct   Husband         White  Male    0             0             48              United-
States  >50K      2
30   Private           144593  HS-grad       9              Never-married       Other-service       Not-in-family   Black  Male    0             0             40              ?
<=50K     2
49   Self-emp-not-inc  43479   Some-college  10             Married-civ-spouse  Craft-repair        Husband         White  Male    0             0             40              United-
States  <=50K     2
                                                                                                                                                                                  ..
31   Private           128567  HS-grad       9              Married-civ-spouse  Craft-repair        Husband         White  Male    0             0             40              United-
States  <=50K     1
                       128493  HS-grad       9              Divorced            Other-service       Not-in-family   White  Female  0             0             25              United-
States  <=50K     1
                       128220  7th-8th       4              Widowed             Adm-clerical        Not-in-family   White  Female  0             0             35              United-
States  <=50K     1
                       127610  Bachelors     13             Married-civ-spouse  Prof-specialty      Wife            White  Female  0             0             40              United-
States  >50K      1
90   Self-emp-not-inc  282095  Some-college  10             Married-civ-spouse  Farming-fishing     Husband         White  Male    0             0             40              United-
States  <=50K     1
Length: 32536, dtype: int64
```

In [11]:
```
data.hist(figsize =(15,15) , color ='b')
plt.show()
```



In [13]:
```
data.describe()
```

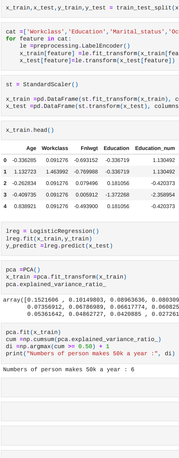
Out[13]:

	Age	Fnlwgt	Education_num	Capital_gain	Capital_loss	Hours_per_week
count	32560.000000	3.256000e+04	32560.000000	32560.000000	32560.000000	32560.000000
mean	38.581634	1.897818e+05	10.080590	1077.615172	87.306511	40.437469
std	13.640642	1.055498e+05	2.572709	7385.402999	402.966116	12.347618
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178315e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783630e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370545e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

In [14]:
```
# filling ? values
```

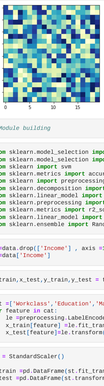
In [7]:
```
data[data =='?'] =np.nan
for col in ['Workclass','Occupation','Native_country']:
    data[col].fillna(data[col].mode()[0] , inplace =True)
```

In [18]:
```
sns.countplot(data['Income'] ,palette ='mako',data =data)
plt.show()
```



In [20]:
```
data.boxplot( figsize= (8,8) ,color= 'b')
plt.show()
```


In [25]:
```
data = np.random.random(( 20 , 20 ))
plt.imshow( data , cmap = 'YlGnBu' , interpolation = 'nearest' )
plt.show()
```



In [28]:
```
# Module building
```

In [10]:
```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn import preprocessing
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler ,LabelEncoder
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

In [8]:
```
x =data.drop(['Income'] , axis =1)
y =data['Income']
```

In [11]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y ,test_size =0.2 , random_state =0)
```

In [12]:
```
cat =['Workclass','Education','Marital_status','Occupation','Relationship','Race','Sex','Native_country']
for feature in cat:
    le =preprocessing.LabelEncoder()
    x_train[feature] =le.fit_transform(x_train[feature])
    x_test[feature]=le.transform(x_test[feature])
```

In [13]:
```
st = StandardScaler()

x_train =pd.DataFrame(st.fit_transform(x_train), columns =x.columns)
x_test =pd.DataFrame(st.transform(x_test), columns =x.columns)
```

In [14]:
```
x_train.head()
```

Out[14]:

	Age	Workclass	Fnlwgt	Education	Education_num	Marital_status	Occupation	Relationship	Race	Sex	Capital_gain	Capital_loss	Hours_per_week	Native_country
0	-0.336285	0.091276	-0.693152	-0.336719	1.130492	-0.406808	-0.609391	-0.899268	0.394607	0.699993	-0.146565	-0.217349	1.995908	0.292605
1	1.132723	1.463992	-0.769988	-0.336719	1.130492	-0.406808	-0.373007	-0.899268	0.394607	0.699993	-0.146565	-0.217349	0.774635	0.292605
2	-0.262834	0.091276	0.079496	0.181056	-0.420373	0.926089	-0.609391	-0.277542	0.394607	-1.428586	-0.146565	-0.217349	1.100308	0.292605
3	-0.409735	0.091276	0.005912	-1.372268	-2.358954	-1.739704	-0.136623	-0.277542	0.394607	0.699993	0.144548	-0.217349	-0.446637	0.292605
4	0.830921	0.091276	-0.493900	0.181056	-0.420373	-0.406808	1.281681	2.209359	-1.962629	-1.428586	-0.146565	-0.217349	-0.039546	0.292605

In [15]:
```
lreg = LogisticRegression()
lreg.fit(x_train,y_train)
y_predict =lreg.predict(x_test)
```

In [16]:
```
pca =PCA()
x_train =pca.fit_transform(x_train)
pca.explained_variance_ratio_
```

Out[16]:
```
array([0.1521606 , 0.10149803, 0.08963636, 0.08036999, 0.07618551,
       0.07356912, 0.06786989, 0.06617774, 0.06082527, 0.06017398,
       0.05361642, 0.04862727, 0.0420885 , 0.02726131])
```

In [17]:
```
pca.fit(x_train)
cum =np.cumsum(pca.explained_variance_ratio_)
di =np.argmax(cum >= 0.50) + 1
print("Numbers of person makes 50k a year :", di)
```

```
Numbers of person makes 50k a year : 6
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```