In [50]:
```python
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')
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 [2]:
```python
data =pd.read_csv("census_income.csv")
```

In [3]:
```python
data
```

Out[3]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32555	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
32556	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
32557	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
32558	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
32559	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

32560 rows × 15 columns

In [4]:
```python
data.shape
```

Out[4]:
```
(32560, 15)
```

In [5]:
```python
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]:
```python
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 [7]:
```python
data.nunique()
```

Out[7]:
```
Age                  73
Workclass             9
Fnlwgt            21647
Education            16
Education_num        16
Marital_status        7
Occupation           15
Relationship          6
Race                  5
Sex                   2
Capital_gain        119
Capital_loss         92
Hours_per_week       94
Native_country       42
Income                2
dtype: int64
```

In [8]:
```python
data.columns
```

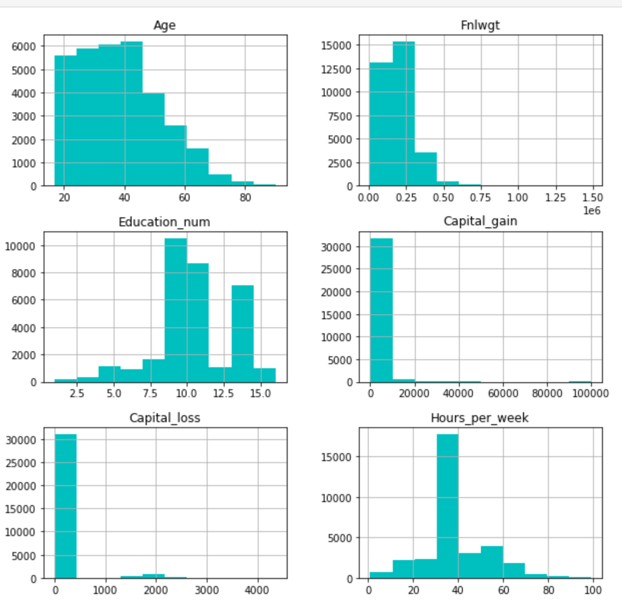
Out[8]:
```
Index(['Age', 'Workclass', 'Fnlwgt', 'Education', 'Education_num',
       'Marital_status', 'Occupation', 'Relationship', 'Race', 'Sex',
       'Capital_gain', 'Capital_loss', 'Hours_per_week', 'Native_country',
       'Income'],
      dtype='object')
```

In [9]:
```python
data.head()
```

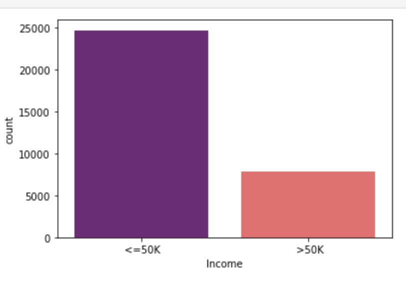
Out[9]:

	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 [10]:
```python
data.hist(figsize =(10,10) , color ='c')
plt.show()
```



In [11]:
```python
sns.countplot(data['Income'] ,palette ='magma' ,data =data)
plt.show()
```



In [12]:
```python
data['Income'].value_counts()
```

Out[12]:
```
 <=50K    24719
 >50K      7841
Name: Income, dtype: int64
```

In [13]:
```python
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]:
```python
data['Workclass'].value_counts()
```

Out[14]:
```
 Private             22696
 Self-emp-not-inc     2541
 Local-gov            2093
 ?                    1836
 State-gov            1297
 Self-emp-inc         1116
 Federal-gov           960
 Without-pay            14
 Never-worked            7
Name: Workclass, dtype: int64
```

In [15]:
```python
data["Occupation"].value_counts()
```

Out[15]:
```
 Prof-specialty       4140
 Craft-repair         4099
 Exec-managerial      4066
 Adm-clerical         3769
 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
```

In [16]:
```python
data["Age"].value_counts()
```

Out[16]:
```
36    898
31    888
34    886
23    877
35    876
     ...
83      6
88      3
85      3
86      1
87      1
Name: Age, Length: 73, dtype: int64
```

In [17]:
```python
data["Education"].value_counts()
```

Out[17]:
```
 HS-grad         10501
 Some-college     7291
 Bachelors        5354
 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
```

In [18]:
```python
data["Fnlwgt"].value_counts()
```

Out[18]:
```
164190    13
203488    13
123011    13
148995    12
126675    12
          ..
325573     1
140176     1
318264     1
329205     1
257302     1
Name: Fnlwgt, Length: 21647, dtype: int64
```

In [19]:
```python
data["Education_num"].value_counts()
```

Out[19]:
```
9     10501
10     7291
13     5354
14     1723
11     1382
7      1175
12     1067
6       933
4       646
15      576
5       514
8       433
16      413
3       333
2       168
1        51
Name: Education_num, dtype: int64
```

In [20]:
```python
data["Marital_status"].value_counts()
```

Out[20]:
```
 Married-civ-spouse       14976
 Never-married            10682
 Divorced                  4443
 Separated                 1025
 Widowed                    993
 Married-spouse-absent      418
 Married-AF-spouse           23
Name: Marital_status, dtype: int64
```

In [21]:
```python
data["Relationship"].value_counts()
```

Out[21]:
```
 Husband           13193
 Not-in-family      8304
 Own-child          5068
 Unmarried          3446
 Wife               1568
 Other-relative      981
Name: Relationship, dtype: int64
```

In [22]:
```python
data["Race"].value_counts()
```

Out[22]:
```
 White                 27815
 Black                  3124
 Asian-Pac-Islander     1039
 Amer-Indian-Eskimo      311
 Other                   271
Name: Race, dtype: int64
```

In [23]:
```python
data["Sex"].value_counts()
```

Out[23]:
```
 Male      21789
 Female    10771
Name: Sex, dtype: int64
```

In [24]:
```python
data["Capital_gain"].value_counts()
```

Out[24]:
```
0        29849
15024      347
7688       284
7298       246
99999      159
         ...
1111         1
2538         1
22040        1
4931         1
5060         1
Name: Capital_gain, Length: 119, dtype: int64
```

In [25]:
```python
data["Capital_loss"].value_counts()
```

Out[25]:
```
0       31041
1902      202
1977      168
1887      159
1848       51
        ...
2080        1
1539        1
1844        1
2489        1
1411        1
Name: Capital_loss, Length: 92, dtype: int64
```

In [26]:
```python
data["Hours_per_week"].value_counts()
```

Out[26]:
```
40    15216
50     2819
45     1824
60     1475
35     1297
      ...
82        1
94        1
92        1
74        1
87        1
Name: Hours_per_week, Length: 94, dtype: int64
```

In [27]:
```python
data["Native_country"].value_counts()
```

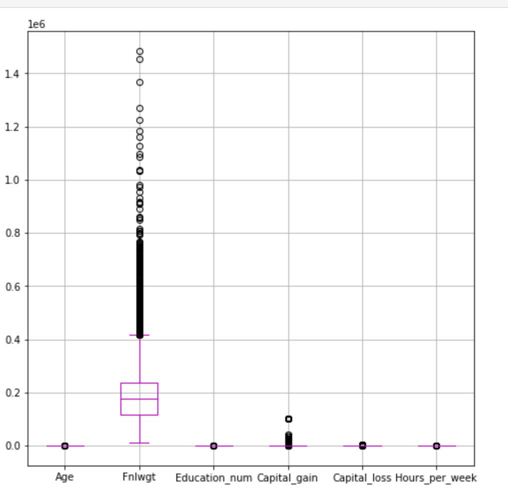
Out[27]:
```
 United-States                 29169
 Mexico                          643
 ?                               583
 Philippines                     198
 Germany                         137
 Canada                          121
 Puerto-Rico                     114
 El-Salvador                     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
```

In [28]:
```python
# filling ? values
```

In [31]:
```python
data[data =='?'] =np.nan
```

In [35]:
```python
for col in ['Workclass','Occupation','Native_country']:
    data[col].fillna(data[col].mode()[0] , inplace =True)
```

In [43]:
```python
data.boxplot( figsize= (8,8) ,color= 'm')
plt.show()
```



In [66]:
```python
# building module
```

In [36]:
```python
x =data.drop(['Income'], axis =1)
y =data['Income']
```

In [38]:
```python
x_train,x_test,y_train,y_test = train_test_split(x,y ,test_size =0.2 , random_state =0)
```

In [43]:
```python
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 [45]:
```python
ss = StandardScaler()

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

In [46]:
```python
x_train.head()
```

Out[46]:

	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	0.838921	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 [47]:
```python
x_test.head()
```

Out[47]:

	Age	Workclass	Fnlwgt	Education	Education_num	Marital_status	Occupation	Relationship	Race	Sex	Capital_gain	Capital_loss	Hours_per_week	Native_country
0	0.104417	0.777634	0.034739	1.216606	-0.032657	-0.406808	-0.373007	-0.899268	0.394607	0.699993	-0.146565	-0.217349	-0.039546	0.292605
1	0.545120	0.091276	-0.017853	1.216606	-0.032657	0.926089	-0.609391	1.587634	0.394607	-1.428586	-0.146565	-0.217349	0.367545	0.292605
2	-0.409735	-2.654157	0.120597	-1.372268	-2.358954	1.592537	-1.554927	-0.277542	0.394607	0.699993	-0.146565	-0.217349	-0.039546	-3.031157
3	-1.291139	0.091276	-0.049148	1.216606	-0.032657	0.926089	-1.318543	0.965908	0.394607	-1.428586	-0.146565	-0.217349	-1.993583	0.292605
4	1.499974	0.091276	-0.795486	-1.372268	-2.358954	-0.406808	0.336145	2.209359	0.394607	-1.428586	-0.146565	-0.217349	4.764126	0.292605

In [48]:
```python
loreg = LogisticRegression()
loreg.fit(x_train,y_train)
y_predict =loreg.predict(x_test)
```

In [51]:
```python
pca =PCA()
x_train =pca.fit_transform(x_train)
pca.explained_variance_ratio_
```

Out[51]:
```
array([0.1521606 , 0.10149803, 0.08963636, 0.08030999, 0.07618551,
       0.07356912, 0.06786989, 0.06617774, 0.06082527, 0.06017398,
       0.05361642, 0.04862727, 0.0420885 , 0.02726131])
```

In [54]:
```python
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 [ ]: