In [5]:
```python
#Employee Salary Prediction Using adult csv
#Load your libaray
import pandas as pd
```

In [7]:
```python
data=pd.read_csv("adult 3.csv")
```

In [9]:
```python
data
```

Out[9]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

48842 rows × 15 columns

In [11]:
```python
data.shape
```

Out[11]: (48842, 15)

In [15]:
```python
data.head()
```

Out[15]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K

In [17]:
```python
data.tail()
```

Out[17]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

In [23]:
```python
#Null values
data.isna()
```

Out[23]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
48838	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
48839	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
48840	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
48841	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

48842 rows × 15 columns

In [25]:
```python
data.isna().sum()
```

Out[25]:
```
age                0
workclass          0
fnlwgt             0
education          0
educational-num    0
marital-status     0
occupation         0
relationship       0
race               0
gender             0
capital-gain       0
capital-loss       0
hours-per-week     0
native-country     0
income             0
dtype: int64
```

In [27]:
```python
print(data.occupation.value_counts())
```

```
occupation
Prof-specialty       6172
Craft-repair         6112
Exec-managerial      6086
Adm-clerical         5611
Sales                5504
Other-service        4923
Machine-op-inspct    3022
?                    2809
Transport-moving     2355
Handlers-cleaners    2072
Farming-fishing      1490
Tech-support         1446
Protective-serv       983
Priv-house-serv       242
Armed-Forces           15
Name: count, dtype: int64
```

In [31]:
```python
print(data.gender.value_counts())
```

```
gender
Male      32650
Female    16192
Name: count, dtype: int64
```

In [43]:
```python
print(data['marital-status'].value_counts())
```

```
marital-status
Married-civ-spouse       22379
Never-married            16117
Divorced                  6633
Separated                 1530
Widowed                   1518
Married-spouse-absent      628
Married-AF-spouse           37
Name: count, dtype: int64
```

In [45]:
```python
print(data['education'].value_counts())
```

```
education
HS-grad         15784
Some-college    10878
Bachelors        8025
Masters          2657
Assoc-voc        2061
11th             1812
Assoc-acdm       1601
10th             1389
7th-8th           955
Prof-school       834
9th               756
12th              657
Doctorate         594
5th-6th           509
1st-4th           247
Preschool          83
Name: count, dtype: int64
```

In [47]:
```python
print(data['workclass'].value_counts())
```

```
workclass
Private             33906
Self-emp-not-inc     3862
Local-gov            3136
?                    2799
State-gov            1981
Self-emp-inc         1695
Federal-gov          1432
Without-pay            21
Never-worked           10
Name: count, dtype: int64
```

In [60]:
```python
data.occupation.replace({'?':'Others'},inplace=True)
```

In [62]:
```python
print(data.occupation.value_counts())
```

```
occupation
Prof-specialty       6172
Craft-repair         6112
Exec-managerial      6086
Adm-clerical         5611
Sales                5504
Other-service        4923
Machine-op-inspct    3022
Others               2809
Transport-moving     2355
Handlers-cleaners    2072
Farming-fishing      1490
Tech-support         1446
Protective-serv       983
Priv-house-serv       242
Armed-Forces           15
Name: count, dtype: int64
```

In [66]:
```python
data
```

Out[66]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	Others	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

48842 rows × 15 columns

In [72]:
```python
data.workclass.replace({'?':'Notlisted'},inplace=True)
```

In [74]:
```python
print(data['workclass'].value_counts())
```

```
workclass
Private             33906
Self-emp-not-inc     3862
Local-gov            3136
Notlisted            2799
State-gov            1981
Self-emp-inc         1695
Federal-gov          1432
Without-pay            21
Never-worked           10
Name: count, dtype: int64
```

In [76]:
```python
data=data[data['workclass']!='Without-pay']
data=data[data['workclass']!='Never-worked']
```

In [78]:
```python
print(data['workclass'].value_counts())
```

```
workclass
Private             33906
Self-emp-not-inc     3862
Local-gov            3136
Notlisted            2799
State-gov            1981
Self-emp-inc         1695
Federal-gov          1432
Name: count, dtype: int64
```

In [80]:
```python
data.shape
```

Out[80]: (48811, 15)

In [88]:
```python
data=data[data['education']!='5th-6th']
data=data[data['education']!='1st-4th']
data=data[data['education']!='Preschool']
```

In [90]:
```python
print(data['education'].value_counts())
```

```
education
HS-grad         15768
Some-college    10873
Bachelors        8025
Masters          2657
Assoc-voc        2061
11th             1809
Assoc-acdm       1599
10th             1387
7th-8th           952
Prof-school       834
9th               756
12th              657
Doctorate         594
Name: count, dtype: int64
```

In [92]:
```python
data.shape
```

Out[92]: (47972, 15)

In [106…
```python
#redundancy
data.drop(columns=['education'],inplace=True)
```

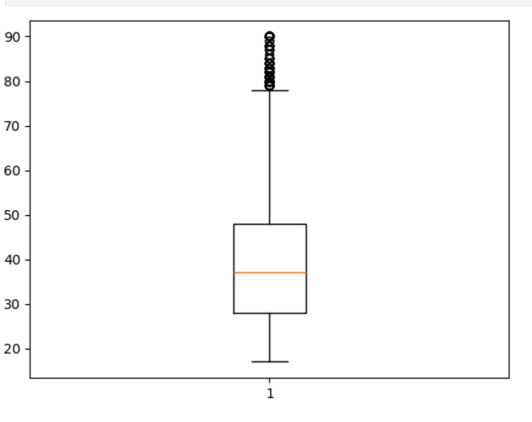
In [96]:
```python
data
```

Out[96]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	Notlisted	103497	Some-college	10	Never-married	Others	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

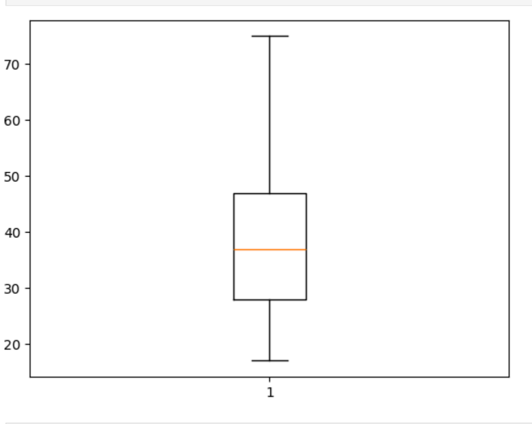
47972 rows × 15 columns

In [110…
```python
#outlier
import matplotlib.pyplot as plt
plt.boxplot(data['age'])
plt.show()
```



In [116…
```python
data=data[(data['age']<=75)& (data['age']>=17)]
```

In [118…
```python
plt.boxplot(data['age'])
plt.show()
```



In [126…
```python
#Label encoding
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data['workclass']=encoder.fit_transform(data['workclass'])
data['marital-status']=encoder.fit_transform(data['marital-status'])
data['occupation']=encoder.fit_transform(data['occupation'])
data['relationship']=encoder.fit_transform(data['relationship'])
data['race']=encoder.fit_transform(data['race'])
data['gender']=encoder.fit_transform(data['gender'])
data['native-country']=encoder.fit_transform(data['native-country'])
data
```

```
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['workclass']=encoder.fit_transform(data['workclass'])
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['marital-status']=encoder.fit_transform(data['marital-status'])
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['occupation']=encoder.fit_transform(data['occupation'])
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['relationship']=encoder.fit_transform(data['relationship'])
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['race']=encoder.fit_transform(data['race'])
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['gender']=encoder.fit_transform(data['gender'])
C:\Users\Akshay Besekar\AppData\Local\Temp\ipykernel_12820\3093134225.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data['native-country']=encoder.fit_transform(data['native-country'])
```

Out[126…

	age	workclass	fnlwgt	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	3	226802	7	4	6	3	2	1	0	0	40	39	<=50K
1	38	3	89814	9	2	4	0	4	1	0	0	50	39	<=50K
2	28	1	336951	12	2	11	0	4	1	0	0	40	39	>50K
3	44	3	160323	10	2	6	0	2	1	7688	0	40	39	>50K
4	18	2	103497	10	4	8	3	4	0	0	0	30	39	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	3	257302	12	2	13	5	4	0	0	0	38	39	<=50K
48838	40	3	154374	9	2	6	0	4	1	0	0	40	39	>50K
48839	58	3	151910	9	6	0	4	4	0	0	0	40	39	<=50K
48840	22	3	201490	9	4	0	3	4	1	0	0	20	39	<=50K
48841	52	4	287927	9	2	3	5	4	0	15024	0	40	39	>50K

47619 rows × 14 columns

In [128…
```python
x=data.drop(columns=['income'])
y=data['income']
```

In [132…
```python
x
```

Out[132…

	age	workclass	fnlwgt	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country
0	25	3	226802	7	4	6	3	2	1	0	0	40	39
1	38	3	89814	9	2	4	0	4	1	0	0	50	39
2	28	1	336951	12	2	11	0	4	1	0	0	40	39
3	44	3	160323	10	2	6	0	2	1	7688	0	40	39
4	18	2	103497	10	4	8	3	4	0	0	0	30	39
...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	3	257302	12	2	13	5	4	0	0	0	38	39
48838	40	3	154374	9	2	6	0	4	1	0	0	40	39
48839	58	3	151910	9	6	0	4	4	0	0	0	40	39
48840	22	3	201490	9	4	0	3	4	1	0	0	20	39
48841	52	4	287927	9	2	3	5	4	0	15024	0	40	39

47619 rows × 13 columns

In [134…
```python
y
```

Out[134…
```
0        <=50K
1        <=50K
2         >50K
3         >50K
4        <=50K
         ...
48837    <=50K
48838     >50K
48839    <=50K
48840    <=50K
48841     >50K
Name: income, Length: 47619, dtype: object
```

In [136…
```python
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
x=scaler.fit_transform(x)
x
```

Out[136…
```
array([[0.13793103, 0.5       , 0.14512876, ..., 0.        , 0.39795918,
        0.95121951],
       [0.36206897, 0.5       , 0.05245126, ..., 0.        , 0.5       ,
        0.95121951],
       [0.18965517, 0.16666667, 0.21964867, ..., 0.        , 0.39795918,
        0.95121951],
       ...,
       [0.70689655, 0.5       , 0.09446153, ..., 0.        , 0.39795918,
        0.95121951],
       [0.0862069 , 0.5       , 0.12800425, ..., 0.        , 0.19387755,
        0.95121951],
       [0.60344828, 0.66666667, 0.18648211, ..., 0.        , 0.39795918,
        0.95121951]])
```

In [138…
```python
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest=train_test_split(x,y, test_size=0.2, random_state=23, stratify=y)
```

In [140…
```python
xtrain
```

Out[140…
```
array([[0.44827586, 0.5       , 0.10248458, ..., 0.        , 0.39795918,
        0.95121951],
       [0.17241379, 0.16666667, 0.08567128, ..., 0.        , 0.34693878,
        0.95121951],
       [0.75862069, 0.83333333, 0.11752469, ..., 0.        , 0.39795918,
        0.04878049],
       ...,
       [0.0862069 , 0.66666667, 0.24567574, ..., 0.        , 0.39795918,
        0.85365854],
       [0.53448276, 0.5       , 0.14505705, ..., 0.        , 0.39795918,
        0.95121951],
       [0.01724138, 0.5       , 0.0515217 , ..., 0.        , 0.35714286,
        0.95121951]])
```

In [148…
```python
#machine learning algorithm
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(xtrain, ytrain) #input and output training data
predict=knn.predict(xtest)
predict
```

Out[148…
```
array(['<=50K', '<=50K', '>50K', ..., '<=50K', '<=50K', '<=50K'],
      dtype=object)
```

In [150…
```python
from sklearn.metrics import accuracy_score
accuracy_score(ytest,predict)
```

Out[150… 0.8167786644267114

In [158…
```python
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(xtrain, ytrain) #input and output training
predict1=lr.predict(xtest)
predict1   #predicted value
```

Out[158…
```
array(['<=50K', '<=50K', '>50K', ..., '<=50K', '<=50K', '<=50K'],
      dtype=object)
```

In [160…
```python
from sklearn.metrics import accuracy_score
accuracy_score(ytest,predict1)
```

Out[160… 0.8197186056278875

In [164…
```python
from sklearn.neural_network import MLPClassifier
clf=MLPClassifier(solver='adam',hidden_layer_sizes=(5,2), random_state=2, max_iter=2000)
clf.fit(xtrain, ytrain)
predict2=clf.predict(xtest)
predict2
```

Out[164…
```
array(['<=50K', '<=50K', '>50K', ..., '<=50K', '<=50K', '<=50K'],
      dtype='<U5')
```

In [166…
```python
from sklearn.metrics import accuracy_score
accuracy_score(ytest,predict2)
```

Out[166… 0.8391432171356573

In [ ]: