Adult

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
df = pd.read_csv('adult.csv')
df.head()
```

Out[2]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hou p we
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	

Pre-process the dataset. Remove those observations containing missing values

In [3]:

```
df.isnull().sum()
```

Out[3]:

```
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 [4]:

```
df = df.replace('?',np.nan)
```

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
age                   0
workclass          2799
fnlwgt                0
education             0
educational-num       0
marital-status        0
occupation         2809
relationship          0
race                  0
gender                0
capital-gain          0
capital-loss          0
hours-per-week        0
native-country      857
income                0
dtype: int64
```

In [6]:

```
df.dropna(inplace = True)
```

In [7]:

```
df.isnull().sum()
```

Out[7]:

```
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
```

MinMaxScaler and StandardScaler

In [8]:

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

cols = df.select_dtypes(include=np.number).columns.tolist()
print(cols)

standard_transformer = Pipeline(steps=[('standard', StandardScaler())])
minmax_transformer = Pipeline(steps=[('minmax', MinMaxScaler())])


preprocessor = ColumnTransformer(
        remainder='passthrough',
        transformers=[
            ('std', standard_transformer , cols[:3]),
            ('mm', minmax_transformer , cols[3:])
        ])

minmax = preprocessor.fit_transform(df)
df_minmax = preprocessor.transform(df)
df_minmax = pd.DataFrame(df_minmax, columns = df.columns)
df_minmax.head()
```

```
['age', 'fnlwgt', 'educational-num', 'capital-gain', 'capital-loss', 'hours-per-week']
```

Out[8]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain
0	-1.02498	0.350889	-1.22156	0	0	0.397959	Private	11th	Never-married	Machine-op-inspct	Own-child
1	-0.041455	-0.945878	-0.438122	0	0	0.5	Private	HS-grad	Married-civ-spouse	Farming-fishing	Husband
2	-0.798015	1.39359	0.737034	0	0	0.397959	Local-gov	Assoc-acdm	Married-civ-spouse	Protective-serv	Husband
3	0.412481	-0.27842	-0.046403	0.0768808	0	0.397959	Private	Some-college	Married-civ-spouse	Machine-op-inspct	Husband
4	-0.344079	0.0848015	-1.61328	0	0	0.295918	Private	10th	Never-married	Other-service	Not-in-family

Transform the non-numeric columns into numeric using label encoding and one-hot encoding.

Label Encoding

In [9]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(df['marital-status'])
df['marital-status '] = le.transform(df['marital-status'])
df.head()
```

Out[9]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hou pe we
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	

In [ ]:

```

```

In [10]:

```
le = LabelEncoder()
le.fit(df['workclass'])
df['workclass'] = le.transform(df['workclass'])
df.head()
```

Out[10]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hou pe we
0	25	2	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	
1	38	2	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	
2	28	1	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	
3	44	2	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	
5	34	2	198693	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	

In [11]:

```
le = LabelEncoder()
le.fit(df['native-country'])
df['native-country'] = le.transform(df['native-country'])
natives = dict(zip(le.classes_, le.transform(le.classes_)))
print(natives)
df.head()

# one_hot = pd.get_dummies(df['native-country'])
# df = df.drop('native-country',axis = 1)
# df = df.join(one_hot)
# df.head()
```

```
{'Cambodia': 0, 'Canada': 1, 'China': 2, 'Columbia': 3, 'Cuba': 4, 'Dominican-Republic':
5, 'Ecuador': 6, 'El-Salvador': 7, 'England': 8, 'France': 9, 'Germany': 10, 'Greece': 11
, 'Guatemala': 12, 'Haiti': 13, 'Holand-Netherlands': 14, 'Honduras': 15, 'Hong': 16, 'Hu
ngary': 17, 'India': 18, 'Iran': 19, 'Ireland': 20, 'Italy': 21, 'Jamaica': 22, 'Japan':
23, 'Laos': 24, 'Mexico': 25, 'Nicaragua': 26, 'Outlying-US(Guam-USVI-etc)': 27, 'Peru':
28, 'Philippines': 29, 'Poland': 30, 'Portugal': 31, 'Puerto-Rico': 32, 'Scotland': 33, '
South': 34, 'Taiwan': 35, 'Thailand': 36, 'Trinadad&Tobago': 37, 'United-States': 38, 'Vi
etnam': 39, 'Yugoslavia': 40}
```

Out[11]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hou pe we
0	25	2	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	
1	38	2	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	
				Assoc-		Married-	Protective-						

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hour pe we
2	28	1	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	
3	44	2	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	
5	34	2	198693	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	

In [12]:

```
le = LabelEncoder()
le.fit(df['marital-status'])
df['marital-status'] = le.transform(df['marital-status'])
df.head()
```

Out[12]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours pe wee
0	25	2	226802	11th	7	4	Machine-op-inspct	Own-child	Black	Male	0	0	4
1	38	2	89814	HS-grad	9	2	Farming-fishing	Husband	White	Male	0	0	5
2	28	1	336951	Assoc-acdm	12	2	Protective-serv	Husband	White	Male	0	0	4
3	44	2	160323	Some-college	10	2	Machine-op-inspct	Husband	Black	Male	7688	0	4
5	34	2	198693	10th	6	4	Other-service	Not-in-family	White	Male	0	0	3

In [13]:

```
le = LabelEncoder()
le.fit(df['education'])
df['education'] = le.transform(df['education'])
df.head()
```

Out[13]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours pe wee
0	25	2	226802	1	7	4	Machine-op-inspct	Own-child	Black	Male	0	0	4
1	38	2	89814	11	9	2	Farming-fishing	Husband	White	Male	0	0	5
2	28	1	336951	7	12	2	Protective-serv	Husband	White	Male	0	0	4
3	44	2	160323	15	10	2	Machine-op-inspct	Husband	Black	Male	7688	0	4
5	34	2	198693	0	6	4	Other-service	Not-in-family	White	Male	0	0	3

In [14]:

```
le = LabelEncoder()
le.fit(df['relationship'])
df['relationship'] = le.transform(df['relationship'])
df.head()
```

Out[14]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hour pe wee
0	25	2	226802	1	7	4	Machine-op-inspct	3	Black	Male	0	0	4
1	38	2	89814	11	9	2	Farming-fishing	0	White	Male	0	0	5
2	28	1	336951	7	12	2	Protective-serv	0	White	Male	0	0	4
3	44	2	160323	15	10	2	Machine-op-inspct	0	Black	Male	7688	0	4
5	34	2	198693	0	6	4	Other-service	1	White	Male	0	0	3

In [15]:

```
le = LabelEncoder()
le.fit(df['race'])
df['race'] = le.transform(df['race'])
df.head()
```

Out[15]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week
0	25	2	226802	1	7	4	Machine-op-inspct	3	2	Male	0	0	40
1	38	2	89814	11	9	2	Farming-fishing	0	4	Male	0	0	50
2	28	1	336951	7	12	2	Protective-serv	0	4	Male	0	0	40
3	44	2	160323	15	10	2	Machine-op-inspct	0	2	Male	7688	0	40
5	34	2	198693	0	6	4	Other-service	1	4	Male	0	0	30

In [16]:

```
le = LabelEncoder()
le.fit(df['occupation'])
df['occupation'] = le.transform(df['occupation'])
df.head()
```

Out[16]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week
0	25	2	226802	1	7	4	6	3	2	Male	0	0	40
1	38	2	89814	11	9	2	4	0	4	Male	0	0	50
2	28	1	336951	7	12	2	10	0	4	Male	0	0	40
3	44	2	160323	15	10	2	6	0	2	Male	7688	0	40
5	34	2	198693	0	6	4	7	1	4	Male	0	0	30

In [17]:

In [17]:

```
le = LabelEncoder()
le.fit(df['income'])
df['income'] = le.transform(df['income'])
income = dict(zip(le.classes_, le.transform(le.classes_)))
print(income)
df.head()
```

```
{'<=50K': 0, '>50K': 1}
```

Out[17]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week
0	25	2	226802	1	7	4	6	3	2	Male	0	0	40
1	38	2	89814	11	9	2	4	0	4	Male	0	0	50
2	28	1	336951	7	12	2	10	0	4	Male	0	0	40
3	44	2	160323	15	10	2	6	0	2	Male	7688	0	40
5	34	2	198693	0	6	4	7	1	4	Male	0	0	30

OneHot Encoding

In [18]:

```
# le = LabelEncoder()
# le.fit(df['gender'])
# df['gender'] = le.transform(df['gender'])
# df.head()
one_hot = pd.get_dummies(df['gender'])
df = df.drop('gender',axis = 1)
df = df.join(one_hot)
df.head()
```

Out[18]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	capital-gain	capital-loss	hours-per-week	native countr
0	25	2	226802	1	7	4	6	3	2	0	0	40	3
1	38	2	89814	11	9	2	4	0	4	0	0	50	3
2	28	1	336951	7	12	2	10	0	4	0	0	40	3
3	44	2	160323	15	10	2	6	0	2	7688	0	40	3
5	34	2	198693	0	6	4	7	1	4	0	0	30	3

Sort the data frame in the descending order of “Hours-per-week”

In [19]:

```
sorted_df = df.sort_values(by=['hours-per-week'], ascending=False)
sorted_df.head()
```

Out[19]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	capital-gain	capital-loss	hours-per-week	n co
20722	43	3	286750	14	15	2	9	0	2	0	0	99	
31741	37	2	241174	9	13	2	9	0	4	0	0	99	

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	capital-gain	capital-loss	hours-per-week	native-co
31741	37	2	241174	9	13	2	9	0	4	0	0	99	
2078	33	5	162705	15	10	0	7	4	4	0	0	99	
34884	33	4	67482	8	11	0	7	4	4	0	0	99	
41994	32	2	183304	8	11	2	13	0	4	0	0	99	

Show all the information whose native country is “US”

In [20]:

```
US_df = df.loc[df['native-country'] == natives["United-States"]]
print(len(US_df))
US_df.head()
```

```
41292
```

Out[20]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	capital-gain	capital-loss	hours-per-week	native-countr
0	25	2	226802	1	7	4	6	3	2	0	0	40	3
1	38	2	89814	11	9	2	4	0	4	0	0	50	3
2	28	1	336951	7	12	2	10	0	4	0	0	40	3
3	44	2	160323	15	10	2	6	0	2	7688	0	40	3
5	34	2	198693	0	6	4	7	1	4	0	0	30	3

Subset the dataset based on male and female belonging to the column “sex”

In [21]:

```
male = df.loc[df['Male'] == 1]
male.drop(['Male','Female'], axis = 1, inplace = True)
print(len(male))
female = df.loc[df['Female'] == 1]
female.drop(['Male','Female'], axis = 1, inplace = True)
print(len(female))
```

```
30527
14695
```

In [22]:

```
male.head()
```

Out[22]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	capital-gain	capital-loss	hours-per-week	native-countr
0	25	2	226802	1	7	4	6	3	2	0	0	40	3
1	38	2	89814	11	9	2	4	0	4	0	0	50	3
2	28	1	336951	7	12	2	10	0	4	0	0	40	3
3	44	2	160323	15	10	2	6	0	2	7688	0	40	3
5	34	2	198693	0	6	4	7	1	4	0	0	30	3

In [23]:

```
# from sklearn.preprocessing import MinMaxScaler
# mm_scaler = MinMaxScaler()
# minmax = mm_scaler.fit_transform(df)
# df_minmax = mm_scaler.transform(df)
# df_minmax = pd.DataFrame(df_minmax, columns = df.columns)
```

In [24]:

```
# df_minmax.head()
```

In [25]:

```
male = df.loc[df['Male'] == 1]
male.drop(['Male','Female'], axis = 1, inplace = True)
print(len(male))
female = df.loc[df['Female'] == 1]
female.drop(['Male','Female'], axis = 1, inplace = True)
```

```
30527
```

Find the number of records, the number of individuals making more than 50k, the number of individuals making less than 50k

In [26]:

```
LessIncome = df.loc[df['income'] == income["<=50K"]]
print(len(LessIncome))
```

```
34014
```

In [27]:

```
MoreIncome = df.loc[df['income'] == income[">50K"]]
print(len(MoreIncome))
```

```
11208
```

Compute the proportion of data points from each category of “native.country.” Combine all the categories except for the one with maximum proportion into the “Other” category

In [28]:

```
proportion = (df['native-country'].value_counts()/df['native-country'].count())*100

keys = []
key_list = list(natives.keys())
val_list = list(natives.values())

for i in proportion.index:
    keys.append(key_list[val_list.index(i)])
keys = np.asarray(keys)
proportion.index = keys
pd.DataFrame(proportion).head()
```

Out[28]:

	native-country
United-States	91.309540
Mexico	1.996816
Philippines	0.625802
Germany	0.426783
Puerto-Rico	0.386980

In [29]:

```
propDf = df.copy()
propDf.loc[propDf['native-country'] != natives["United-States"], 'native-country'] = "Others"
propDf.loc[(propDf['native-country'] == natives["United-States"]), 'native-country'] = "United-States"
propDf['native-country'].unique()
```

Out[29]:

```
array(['United-States', 'Others'], dtype=object)
```

In [30]:

```
propDf.head()
```

Out[30]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	capital-gain	capital-loss	hours-per-week	native countr
0	25	2	226802	1	7	4	6	3	2	0	0	40	United State
1	38	2	89814	11	9	2	4	0	4	0	0	50	United State
2	28	1	336951	7	12	2	10	0	4	0	0	40	United State
3	44	2	160323	15	10	2	6	0	2	7688	0	40	United State
5	34	2	198693	0	6	4	7	1	4	0	0	30	United State

Convert the categories of the outcome variable “income” to 0s (for less than 50K) and 1s (for greater than 50K)

In [31]:

```
print(propDf['income'].unique())
propDf.loc[(propDf['income'] == income["<=50K"]), 'income'] = 0
propDf.loc[(propDf['income'] == income[">50K"]), 'income'] = 1
```

```
[0 1]
```

In [ ]:

In [ ]: