In this notebook, I did data processing and saved the Data as an csv file

Importing all the necessary libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

In [2]:
```
name_cols = []
with open('adult.names') as file:
    for f in file.readlines():
        if not str(f).startswith('|') and ':' in str(f):
            name_cols.append(str(f).split(':')[0])
name_cols.append('Salary')
```

In [3]:
```
df = pd.read_csv('adult.data', names=name_cols)
```

In [4]:
```
df.head()
```

Out[4]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	Salary
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

The Object columns had values with prevailing whiespaces in them, so we remove those whitespaces.

In [5]:
```
object_cols = []
for i, enum in enumerate(df.dtypes):
    if enum=='object':
        object_cols.append(i)

for i in range(len(object_cols)):
    object_cols[i] = df.dtypes.index[object_cols[i]]

def correct_names(name):
    if name.startswith(" ") or name.endswith(" "):
        return name.strip(" ")
    else:
        return name

for i in object_cols:
    df[i] = df[i].apply(correct_names)
```

Since there were a lot of inputs as Country,so I reduced the number of inputs by only taking the top 20 countries by number of people with having salary more than 50K and putting the rest into Others.

In [8]:
```
native_country_new = df[df['Salary']==">50K"]['native-country'].value_counts().index[:20]
def get_country(name):
    if name not in native_country_new or name=='?':
        return "Others"
    else:
        return name

df['native-country'] = df['native-country'].apply(get_country)
```

There were 16 different inputs for education, so I reduced them to less than 10.

In [9]:
```
def change_edu_level(name):
    if name=="HS-grad":
        return "High School"
    elif name in ["Bachelors","Some-college"]:
        return "Bachelors"
    elif name in ["11th", "9th", "7th-8th", "5th-6th", "10th", "1st-4th", "12th", "Preschool", "compulsory"]:
        return "Compulsory"
    elif name in ["Assoc-acdm", "Assoc-voc"]:
        return "Associate"
    else:
        return name

df['education'] = df['education'].apply(change_edu_level)
```

Similar names had different inputs.

In [10]:
```
def get_mar(name):
    if name in ['Married-civ-spouse', 'Married-spouse-absent', 'Married-AF-spouse']:
        return "Married"
    elif name in ['Divorced', 'Separated']:
        return "Divorced"
    else:
        return name

df['marital-status'] = df['marital-status'].apply(get_mar)
```

"?" were unknown values which were like the Null input, so I put them as "other-services".

In [11]:
```
def remove_qm(name):
    if name == '?':
        return 'Other-service'
    else:
        return name

df['workclass'] = df['workclass'].apply(remove_qm)
df['occupation'] = df['occupation'].apply(remove_qm)
```

In [12]:
```
df
```

Out[12]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States
1	50	Self-emp-not-inc	83311	Bachelors	13	Married	Exec-managerial	Husband	White	Male	0	0	13	United-States
2	38	Private	215646	High School	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States
3	53	Private	234721	Compulsory	7	Married	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States
4	28	Private	338409	Bachelors	13	Married	Prof-specialty	Wife	Black	Female	0	0	40	Cuba
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32556	27	Private	257302	Associate	12	Married	Tech-support	Wife	White	Female	0	0	38	United-States
32557	40	Private	154374	High School	9	Married	Machine-op-inspct	Husband	White	Male	0	0	40	United-States
32558	58	Private	151910	High School	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States
32559	22	Private	201490	High School	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States
32560	52	Self-emp-inc	287927	High School	9	Married	Exec-managerial	Wife	White	Female	15024	0	40	United-States

32561 rows × 15 columns

In [13]:
```
df[df['Salary']=='>50K']
```

Out[13]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
7	52	Self-emp-not-inc	209642	High School	9	Married	Exec-managerial	Husband	White	Male	0	0	45	United-States
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States
9	42	Private	159449	Bachelors	13	Married	Exec-managerial	Husband	White	Male	5178	0	40	United-States
10	37	Private	280464	Bachelors	10	Married	Exec-managerial	Husband	Black	Male	0	0	80	United-States
11	30	State-gov	141297	Bachelors	13	Married	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	India
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32539	71	Other-service	287372	Doctorate	16	Married	Other-service	Husband	White	Male	0	0	10	United-States
32545	39	Local-gov	111499	Associate	12	Married	Adm-clerical	Wife	White	Female	0	0	20	United-States
32554	53	Private	321865	Masters	14	Married	Exec-managerial	Husband	White	Male	0	0	40	United-States
32557	40	Private	154374	High School	9	Married	Machine-op-inspct	Husband	White	Male	0	0	40	United-States
32560	52	Self-emp-inc	287927	High School	9	Married	Exec-managerial	Wife	White	Female	15024	0	40	United-States

7841 rows × 15 columns

In [14]:
```
df.to_csv('AdultDataCleared.csv')
```

In [ ]:
```

```