KHUSHI KATHURIA 2K19CSUN04012 BTECH CSE DSML

In [70]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"]=(20,10)
```

IMPORTING CSV FILES

In [2]:

```
df1= pd.read_csv("adult_dataset.csv")
df1
```

Out[2]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relatio
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-
2	66	?	186061	Some-college	10	Widowed	?	Unm
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unm
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Owr
...	...	...	...	...	...	...	...	
32556	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Hu:
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unm
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Owr

32561 rows × 15 columns

DATA PREPROCESSING

In [3]:

```
df1.isnull().sum()
```

Out[3]:

```
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 [4]:

```
df1[df1.workclass=="?"]
```

Out[4]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relatio
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-
2	66	?	186061	Some-college	10	Widowed	?	Unm
14	51	?	172175	Doctorate	16	Never-married	?	Not-in-
24	61	?	135285	HS-grad	9	Married-civ-spouse	?	Hu:
44	71	?	100820	HS-grad	9	Married-civ-spouse	?	Hu:
...	...	...	...	...	...	...	...	
32533	35	?	320084	Bachelors	13	Married-civ-spouse	?	
32534	30	?	33811	Bachelors	13	Never-married	?	Not-in-
32541	71	?	287372	Doctorate	16	Married-civ-spouse	?	Hu:
32543	41	?	202822	HS-grad	9	Separated	?	Not-in-
32544	72	?	129912	HS-grad	9	Married-civ-spouse	?	Hu:

1836 rows × 15 columns

In [5]:

```
df1= df1[df1.workclass!="?"]
df1= df1[df1.workclass!="Never-worked"]
df1
```

Out[5]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relatio
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unm
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Owr
5	34	Private	216864	HS-grad	9	Divorced	Other-service	Unm
6	38	Private	150601	10th	6	Separated	Adm-clerical	Unm
...	...	...	...	...	...	...	...	
32556	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Hu
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unm
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Owr

30718 rows × 15 columns

In [6]:

```
df1.groupby("native.country")["native.country"].agg('count')
df1=df1[df1["native.country"]!='?']
df1.groupby("native.country")["native.country"].agg('count')
```

Out[6]:

```
native.country
Cambodia                         18
Canada                          107
China                            68
Columbia                         56
Cuba                             92
Dominican-Republic               67
Ecuador                          27
El-Salvador                     100
England                          86
France                           27
Germany                         128
Greece                           29
Guatemala                        63
Haiti                            42
Holand-Netherlands                1
Honduras                         12
Hong                             19
Hungary                          13
India                           100
Iran                             42
Ireland                          24
Italy                            68
Jamaica                          80
Japan                            59
Laos                             17
Mexico                          610
Nicaragua                        33
Outlying-US(Guam-USVI-etc)       14
Peru                             30
Philippines                     188
Poland                           56
Portugal                         34
Puerto-Rico                     109
Scotland                         11
South                            71
Taiwan                           42
Thailand                         17
Trinadad&Tobago                  18
United-States                 27504
Vietnam                          64
Yugoslavia                       16
Name: native.country, dtype: int64
```

In [7]:

```
df1["hours.per.week"].unique()
```

Out[7]:

```
array([18, 40, 45, 20, 35, 55, 76, 50, 42, 25, 32, 90, 60, 48, 70, 52, 72,
       39,  6, 65, 80, 67, 99, 30, 75, 12, 26, 10, 84, 38, 62, 44,  8, 28,
       59,  5, 24, 57, 34, 37, 46, 56, 41, 98, 43, 15,  1, 36, 47, 68, 54,
        2, 16,  9,  3,  4, 33, 23, 22, 64, 51, 19, 58, 63, 53, 96, 66, 21,
        7, 13, 27, 14, 77, 31, 78, 11, 49, 17, 85, 87, 88, 73, 89, 97, 94,
       29, 82, 86, 91, 81, 92, 61, 74, 95], dtype=int64)
```

In [8]:

```
df1.groupby("marital.status")["marital.status"].agg('count')
```

Out[8]:

```
marital.status
Divorced                  4214
Married-AF-spouse           21
Married-civ-spouse       14065
Married-spouse-absent      370
Never-married             9726
Separated                  939
Widowed                    827
Name: marital.status, dtype: int64
```

In [9]:

```
df1.groupby("education.num")["education.num"].agg('count')
```

Out[9]:

```
education.num
1       45
2      151
3      288
4      557
5      455
6      820
7     1048
8      377
9     9840
10    6678
11    1307
12    1008
13    5044
14    1627
15     542
16     375
Name: education.num, dtype: int64
```

In [10]:

```
df1.groupby("occupation")["occupation"].agg('count')
```

Out[10]:

```
occupation
Adm-clerical         3721
Armed-Forces            9
Craft-repair         4030
Exec-managerial      3992
Farming-fishing       989
Handlers-cleaners    1350
Machine-op-inspct    1966
Other-service        3212
Priv-house-serv       143
Prof-specialty       4038
Protective-serv       644
Sales                3584
Tech-support          912
Transport-moving     1572
Name: occupation, dtype: int64
```

In [11]:

```
df1.groupby("relationship")["relationship"].agg('count')
```

Out[11]:

```
relationship
Husband           12463
Not-in-family      7726
Other-relative      889
Own-child          4466
Unmarried          3212
Wife               1406
Name: relationship, dtype: int64
```

In [12]:

```
df1.groupby("sex")["sex"].agg('count')
```

Out[12]:

```
sex
Female     9782
Male      20380
Name: sex, dtype: int64
```

In [13]:

```
df1.groupby("race")["race"].agg('count')
```

Out[13]:

```
race
Amer-Indian-Eskimo      286
Asian-Pac-Islander      895
Black                  2817
Other                   231
White                 25933
Name: race, dtype: int64
```

In [14]:

```
df1.groupby("age")["age"].agg('count')
```

Out[14]:

```
age
17    328
18    447
19    594
20    629
21    621
     ...
84      8
85      3
86      1
88      3
90     35
Name: age, Length: 72, dtype: int64
```

In [15]:

```
df1.groupby("income")["income"].agg('count')
```

Out[15]:

```
income
<=50K    22654
>50K      7508
Name: income, dtype: int64
```

In [16]:

```
df1=df1[df1.age<=60]
df1
df1.groupby("age")["age"].agg('count')
```

Out[16]:

```
age
17    328
18    447
19    594
20    629
21    621
22    674
23    824
24    752
25    799
26    745
27    789
28    808
29    774
30    813
31    851
32    789
33    837
34    836
35    828
36    852
37    828
38    791
39    786
40    765
41    769
42    741
43    743
44    704
45    706
46    711
47    683
48    523
49    555
50    575
51    571
52    455
53    448
54    394
55    386
56    343
57    337
58    344
59    332
60    276
Name: age, dtype: int64
```

In [17]:

```
df1=df1.drop(["fnlwgt","education.num","capital.gain","capital.loss"],axis='columns')
```

In [18]:

```
ages=df1.age
ages
```

Out[18]:

```
3        54
4        41
5        34
6        38
10       45
         ..
32556    22
32557    27
32558    40
32559    58
32560    22
Name: age, Length: 28356, dtype: int64
```

In [19]:

```
bins=[20,30,40,50,60]
age1=pd.cut(ages,bins)
age1=age1.cat.codes
age1
```

Out[19]:

```
3        3
4        2
5        1
6        1
10       2
        ..
32556    0
32557    0
32558    1
32559    3
32560    0
Length: 28356, dtype: int8
```

In [20]:

```
df1["age.group"]= age1
```

In [21]:

```
df1=df1.drop(["age"], axis='columns')
```

In [22]:

```
df1.groupby("education")["education"].agg('count')
```

Out[22]:

```
education
10th             743
11th             995
12th             358
1st-4th          131
5th-6th          254
7th-8th          427
9th              404
Assoc-acdm       985
Assoc-voc       1254
Bachelors       4809
Doctorate        327
HS-grad         9236
Masters         1529
Preschool         39
Prof-school      492
Some-college    6373
Name: education, dtype: int64
```

In [23]:

```
primary=df1[df1.education=="1st-4th"]
primary
```

Out[23]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours
26	Private	1st-4th	Married-civ-spouse	Craft-repair	Not-in-family	White	Male	
219	Self-emp-not-inc	1st-4th	Married-civ-spouse	Transport-moving	Husband	White	Male	
1258	Self-emp-not-inc	1st-4th	Widowed	Craft-repair	Other-relative	White	Female	
2541	Self-emp-not-inc	1st-4th	Married-civ-spouse	Exec-managerial	Husband	White	Male	
3485	Private	1st-4th	Married-civ-spouse	Farming-fishing	Husband	White	Male	
...	...	...	...	...	...	...	...	
31956	Private	1st-4th	Married-spouse-absent	Other-service	Own-child	Other	Female	
32108	Private	1st-4th	Married-civ-spouse	Other-service	Wife	Asian-Pac-Islander	Female	
32333	Private	1st-4th	Married-civ-spouse	Handlers-cleaners	Other-relative	White	Male	
32418	Private	1st-4th	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
32439	Private	1st-4th	Married-civ-spouse	Machine-op-inspct	Wife	Amer-Indian-Eskimo	Female	

131 rows × 11 columns

In [24]:

```
primary["education"]='primary'
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:1: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

In [25]:

```
primary
```

Out[25]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours
26	Private	primary	Married-civ-spouse	Craft-repair	Not-in-family	White	Male	
219	Self-emp-not-inc	primary	Married-civ-spouse	Transport-moving	Husband	White	Male	
1258	Self-emp-not-inc	primary	Widowed	Craft-repair	Other-relative	White	Female	
2541	Self-emp-not-inc	primary	Married-civ-spouse	Exec-managerial	Husband	White	Male	
3485	Private	primary	Married-civ-spouse	Farming-fishing	Husband	White	Male	
...	...	...	...	...	...	...	...	
31956	Private	primary	Married-spouse-absent	Other-service	Own-child	Other	Female	
32108	Private	primary	Married-civ-spouse	Other-service	Wife	Asian-Pac-Islander	Female	
32333	Private	primary	Married-civ-spouse	Handlers-cleaners	Other-relative	White	Male	
32418	Private	primary	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
32439	Private	primary	Married-civ-spouse	Machine-op-inspct	Wife	Amer-Indian-Eskimo	Female	

131 rows × 11 columns

In [26]:

```
secondary1=df1[df1.education=="5th-6th"]
secondary1["education"]='secondary'
secondary1
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[26]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.
27	Private	secondary	Married-civ-spouse	Other-service	Husband	White	Male	
142	Private	secondary	Divorced	Craft-repair	Not-in-family	White	Female	
226	Self-emp-not-inc	secondary	Married-civ-spouse	Sales	Husband	White	Male	
643	Private	secondary	Married-civ-spouse	Transport-moving	Husband	Amer-Indian-Eskimo	Male	
774	Self-emp-not-inc	secondary	Married-civ-spouse	Exec-managerial	Husband	White	Male	
...	...	...	...	...	...	...	...	
31511	Private	secondary	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	
31632	Private	secondary	Married-civ-spouse	Other-service	Husband	White	Male	
31670	Private	secondary	Never-married	Machine-op-inspct	Own-child	White	Male	
32255	Local-gov	secondary	Never-married	Handlers-cleaners	Other-relative	White	Male	
32358	Private	secondary	Married-spouse-absent	Farming-fishing	Not-in-family	White	Male	

254 rows × 11 columns

In [27]:

```
secondary2=df1[df1.education=="7th-8th"]
secondary2["education"]='secondary'
secondary2
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[27]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
3	Private	secondary	Divorced	Machine-op-inspct	Unmarried	White	Female	
212	Private	secondary	Never-married	Handlers-cleaners	Not-in-family	Black	Male	
216	Private	secondary	Married-civ-spouse	Handlers-cleaners	Husband	Other	Male	
218	Self-emp-not-inc	secondary	Married-civ-spouse	Craft-repair	Husband	White	Male	
277	Private	secondary	Married-civ-spouse	Farming-fishing	Husband	White	Male	
...	...	...	...	...	...	...	...	
32168	Private	secondary	Married-civ-spouse	Craft-repair	Husband	White	Male	
32374	Private	secondary	Married-spouse-absent	Machine-op-inspct	Not-in-family	White	Male	
32416	Local-gov	secondary	Never-married	Other-service	Other-relative	Black	Female	
32445	Private	secondary	Divorced	Machine-op-inspct	Not-in-family	White	Female	
32521	Private	secondary	Married-civ-spouse	Craft-repair	Husband	White	Male	

427 rows × 11 columns

In [28]:

```
secondary3=df1[df1.education=="9th"]
secondary3["education"]='secondary'
secondary3
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[28]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
197	Local-gov	secondary	Widowed	Handlers-cleaners	Unmarried	White	Male	
963	Private	secondary	Married-civ-spouse	Other-service	Husband	White	Male	
1081	Private	secondary	Never-married	Machine-op-inspct	Not-in-family	White	Male	
1110	Private	secondary	Divorced	Other-service	Not-in-family	White	Female	
1116	Private	secondary	Never-married	Handlers-cleaners	Own-child	White	Male	
...	...	...	...	...	...	...	...	
32166	Private	secondary	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	
32263	Private	secondary	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
32316	Private	secondary	Never-married	Machine-op-inspct	Own-child	Black	Male	
32460	Private	secondary	Married-civ-spouse	Transport-moving	Husband	Black	Male	
32474	Private	secondary	Married-civ-spouse	Craft-repair	Husband	White	Male	

404 rows × 11 columns

In [29]:

```
secondary4=df1[df1.education=="10th"]
secondary4["education"]='secondary'
secondary4
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[29]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.
6	Private	secondary	Separated	Adm-clerical	Unmarried	White	Male	
28	Self-emp-inc	secondary	Never-married	Transport-moving	Not-in-family	White	Male	
29	Private	secondary	Never-married	Prof-specialty	Not-in-family	White	Male	
31	Self-emp-inc	secondary	Widowed	Exec-managerial	Unmarried	White	Female	
195	Private	secondary	Widowed	Adm-clerical	Unmarried	White	Female	
...	...	...	...	...	...	...	...	
32457	Private	secondary	Never-married	Other-service	Own-child	White	Male	
32510	Private	secondary	Never-married	Adm-clerical	Not-in-family	Black	Male	
32513	Private	secondary	Divorced	Other-service	Not-in-family	Black	Female	
32529	Private	secondary	Married-civ-spouse	Transport-moving	Husband	White	Male	
32552	Private	secondary	Married-civ-spouse	Handlers-cleaners	Husband	Amer-Indian-Eskimo	Male	

743 rows × 11 columns

In [30]:

```
HS1=df1[df1.education=="11th"]
HS1["education"]='High School'
HS1
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[30]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
16	Private	High School	Divorced	Transport-moving	Not-in-family	White	Male	
21	Private	High School	Separated	Sales	Not-in-family	White	Female	
61	Self-emp-inc	High School	Never-married	Exec-managerial	Other-relative	White	Male	
241	Self-emp-not-inc	High School	Married-civ-spouse	Craft-repair	Own-child	White	Male	
247	Private	High School	Married-civ-spouse	Craft-repair	Husband	White	Male	
...	...	...	...	...	...	...	...	
32420	Private	High School	Married-civ-spouse	Other-service	Husband	White	Male	
32466	Self-emp-not-inc	High School	Married-spouse-absent	Craft-repair	Not-in-family	White	Male	
32499	Private	High School	Divorced	Machine-op-inspct	Unmarried	White	Female	
32502	Private	High School	Never-married	Prof-specialty	Own-child	White	Male	
32525	Private	High School	Married-civ-spouse	Sales	Husband	White	Male	

995 rows × 11 columns

In [31]:

```
HS2=df1[df1.education=="12th"]
HS2["education"]='High School'
HS2
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[31]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
178	Private	High School	Divorced	Craft-repair	Not-in-family	White	Male	
954	Local-gov	High School	Married-civ-spouse	Transport-moving	Husband	White	Male	
1012	Private	High School	Never-married	Machine-op-inspct	Not-in-family	White	Male	
1093	Private	High School	Never-married	Other-service	Own-child	White	Female	
1187	Self-emp-inc	High School	Married-civ-spouse	Exec-managerial	Husband	White	Male	
...	...	...	...	...	...	...	...	
32354	Private	High School	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
32368	Private	High School	Never-married	Other-service	Own-child	White	Male	
32410	Private	High School	Never-married	Adm-clerical	Own-child	White	Male	
32482	Private	High School	Married-civ-spouse	Craft-repair	Husband	White	Male	
32538	Private	High School	Never-married	Protective-serv	Own-child	Black	Male	

358 rows × 11 columns

In [32]:

```
HSG=df1[df1.education=="HS-grad"]
HSG["education"]='High School Graduate'
HSG
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[32]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
5	Private	High School Graduate	Divorced	Other-service	Unmarried	White	Female	
34	Self-emp-not-inc	High School Graduate	Never-married	Exec-managerial	Not-in-family	Black	Male	
36	Private	High School Graduate	Never-married	Sales	Not-in-family	White	Male	
51	Private	High School Graduate	Widowed	Sales	Not-in-family	White	Female	
71	Self-emp-not-inc	High School Graduate	Married-civ-spouse	Exec-managerial	Husband	White	Male	
...	...	...	...	...	...	...	...	
32542	State-gov	High School Graduate	Separated	Adm-clerical	Own-child	White	Female	
32549	Private	High School Graduate	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
32558	Private	High School Graduate	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
32559	Private	High School Graduate	Widowed	Adm-clerical	Unmarried	White	Female	
32560	Private	High School Graduate	Never-married	Adm-clerical	Own-child	White	Male	

9236 rows × 11 columns

In [33]:

```
Bach1=df1[df1.education=="Bachelors"]
Bach1["education"]='Bachelors'
Bach1
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[33]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours
12	Private	Bachelors	Widowed	Other-service	Not-in-family	White	Female	
19	Private	Bachelors	Separated	Sales	Not-in-family	White	Male	
20	Private	Bachelors	Never-married	Exec-managerial	Not-in-family	White	Male	
33	Private	Bachelors	Divorced	Exec-managerial	Not-in-family	White	Male	
40	Private	Bachelors	Divorced	Exec-managerial	Unmarried	White	Male	
...	...	...	...	...	...	...	...	
32507	Local-gov	Bachelors	Married-civ-spouse	Prof-specialty	Husband	White	Male	
32512	Private	Bachelors	Married-civ-spouse	Prof-specialty	Husband	White	Male	
32516	Local-gov	Bachelors	Never-married	Adm-clerical	Own-child	Black	Female	
32536	Private	Bachelors	Married-civ-spouse	Exec-managerial	Husband	Asian-Pac-Islander	Male	
32539	Private	Bachelors	Never-married	Exec-managerial	Not-in-family	White	Female	

4809 rows × 11 columns

In [34]:

```
Bach2=df1[df1.education=="Prof-school"]
Bach2["education"]='Bachelors'
Bach2
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[34]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours
11	Self-emp-not-inc	Bachelors	Never-married	Prof-specialty	Not-in-family	White	Male	
15	Private	Bachelors	Divorced	Prof-specialty	Not-in-family	White	Male	
32	Private	Bachelors	Divorced	Exec-managerial	Not-in-family	White	Male	
37	Private	Bachelors	Never-married	Prof-specialty	Not-in-family	White	Female	
50	Self-emp-not-inc	Bachelors	Never-married	Prof-specialty	Not-in-family	White	Male	
...	...	...	...	...	...	...	...	
31892	Self-emp-not-inc	Bachelors	Married-civ-spouse	Prof-specialty	Husband	White	Male	
31908	Private	Bachelors	Never-married	Prof-specialty	Own-child	Asian-Pac-Islander	Male	
31918	Private	Bachelors	Married-civ-spouse	Exec-managerial	Husband	Asian-Pac-Islander	Male	
32290	Self-emp-inc	Bachelors	Married-civ-spouse	Prof-specialty	Husband	White	Male	
32449	Private	Bachelors	Married-civ-spouse	Prof-specialty	Husband	White	Male	

492 rows × 11 columns

In [35]:

```
Mast=df1[df1.education=="Masters"]
Mast["education"]='Masters'
Mast
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[35]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours
13	Private	Masters	Separated	Exec-managerial	Not-in-family	White	Male	
17	Private	Masters	Divorced	Exec-managerial	Not-in-family	White	Male	
39	Private	Masters	Divorced	Prof-specialty	Not-in-family	White	Female	
41	Private	Masters	Divorced	Exec-managerial	Unmarried	White	Female	
43	Private	Masters	Divorced	Prof-specialty	Unmarried	White	Female	
...	...	...	...	...	...	...	...	
32509	Private	Masters	Divorced	Sales	Not-in-family	White	Female	
32518	Private	Masters	Married-civ-spouse	Prof-specialty	Wife	White	Female	
32546	Private	Masters	Divorced	Other-service	Not-in-family	Other	Female	
32554	Private	Masters	Never-married	Tech-support	Not-in-family	Asian-Pac-Islander	Male	
32555	Private	Masters	Married-civ-spouse	Exec-managerial	Husband	White	Male	

1529 rows × 11 columns

In [36]:

```
Doc=df1[df1.education=="Doctorate"]
Doc["education"]='Doctorate'
Doc
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[36]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
10	Private	Doctorate	Divorced	Prof-specialty	Unmarried	Black	Female	
38	Self-emp-not-inc	Doctorate	Never-married	Prof-specialty	Not-in-family	White	Female	
99	Private	Doctorate	Married-civ-spouse	Prof-specialty	Husband	White	Male	
101	Private	Doctorate	Married-civ-spouse	Prof-specialty	Husband	White	Male	
183	State-gov	Doctorate	Never-married	Exec-managerial	Not-in-family	White	Female	
...	...	...	...	...	...	...	...	
32315	Local-gov	Doctorate	Married-civ-spouse	Exec-managerial	Husband	White	Male	
32350	Private	Doctorate	Married-civ-spouse	Prof-specialty	Husband	White	Male	
32443	Local-gov	Doctorate	Divorced	Exec-managerial	Not-in-family	White	Female	
32477	Private	Doctorate	Divorced	Prof-specialty	Not-in-family	White	Female	
32535	Private	Doctorate	Married-civ-spouse	Prof-specialty	Husband	White	Male	

327 rows × 11 columns

In [37]:

```
pre=df1[df1.education=="Preschool"]
pre["education"]='Pre-School'
pre
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[37]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours
1106	Private	Pre-School	Married-spouse-absent	Machine-op-inspct	Not-in-family	White	Male	
1153	Private	Pre-School	Married-civ-spouse	Other-service	Husband	White	Male	
1678	Private	Pre-School	Married-civ-spouse	Farming-fishing	Other-relative	White	Male	
3260	Private	Pre-School	Married-civ-spouse	Machine-op-inspct	Wife	Asian-Pac-Islander	Female	
4424	Local-gov	Pre-School	Never-married	Machine-op-inspct	Not-in-family	White	Female	
5042	Local-gov	Pre-School	Married-civ-spouse	Other-service	Husband	White	Male	
6773	Private	Pre-School	Never-married	Other-service	Other-relative	White	Female	
7211	Private	Pre-School	Married-civ-spouse	Farming-fishing	Husband	White	Male	
7787	Private	Pre-School	Married-civ-spouse	Other-service	Not-in-family	White	Male	
10198	Private	Pre-School	Married-civ-spouse	Farming-fishing	Husband	White	Male	
10374	Private	Pre-School	Never-married	Farming-fishing	Not-in-family	White	Male	
11261	Private	Pre-School	Never-married	Farming-fishing	Not-in-family	White	Male	
11856	Private	Pre-School	Never-married	Other-service	Own-child	White	Male	
13209	Private	Pre-School	Never-married	Machine-op-inspct	Not-in-family	White	Male	
14107	Private	Pre-School	Married-civ-spouse	Other-service	Husband	White	Male	
17344	Private	Pre-School	Never-married	Farming-fishing	Not-in-family	White	Male	
18656	Private	Pre-School	Married-civ-spouse	Machine-op-inspct	Wife	White	Female	
20267	Private	Pre-School	Never-married	Farming-fishing	Not-in-family	White	Male	
22845	Private	Pre-School	Never-married	Machine-op-inspct	Own-child	White	Female	
23203	Local-gov	Pre-School	Never-married	Handlers-cleaners	Own-child	White	Female	
23541	Private	Pre-School	Never-married	Other-service	Not-in-family	White	Female	
23861	Private	Pre-School	Married-spouse-absent	Adm-clerical	Own-child	White	Male	
24227	Private	Pre-School	Never-married	Farming-fishing	Not-in-family	White	Male	

	workclass	education	marital.status	occupation	relationship	race	sex	hours
25289	Private	Pre-School	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	
25611	Private	Pre-School	Married-civ-spouse	Craft-repair	Husband	White	Male	
25737	Private	Pre-School	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
26087	Private	Pre-School	Separated	Other-service	Unmarried	White	Female	
26098	Private	Pre-School	Never-married	Other-service	Own-child	White	Female	
26198	Private	Pre-School	Never-married	Other-service	Not-in-family	Asian-Pac-Islander	Female	
26554	Private	Pre-School	Never-married	Handlers-cleaners	Not-in-family	White	Male	
26799	Private	Pre-School	Never-married	Farming-fishing	Not-in-family	White	Male	
27295	Private	Pre-School	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
27389	Private	Pre-School	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	
27931	Local-gov	Pre-School	Never-married	Adm-clerical	Own-child	Black	Female	
28939	Private	Pre-School	Never-married	Machine-op-inspct	Not-in-family	Black	Male	
31891	State-gov	Pre-School	Never-married	Other-service	Not-in-family	White	Male	
32262	Private	Pre-School	Never-married	Other-service	Not-in-family	White	Female	
32381	Private	Pre-School	Married-civ-spouse	Machine-op-inspct	Other-relative	Black	Male	
32446	Private	Pre-School	Divorced	Other-service	Not-in-family	Other	Male	

In [38]:

```
Drop=df1[df1.education=="Some-college"]
Drop["education"]='Dropout'
Drop
```

```
C:\Users\Khushi\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
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-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[38]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
4	Private	Dropout	Separated	Prof-specialty	Own-child	White	Female	
23	Private	Dropout	Married-civ-spouse	Transport-moving	Husband	White	Male	
30	Private	Dropout	Separated	Other-service	Not-in-family	White	Male	
42	Private	Dropout	Divorced	Adm-clerical	Unmarried	White	Female	
53	Private	Dropout	Never-married	Exec-managerial	Not-in-family	White	Female	
...	...	...	...	...	...	...	...	
32530	Private	Dropout	Never-married	Adm-clerical	Own-child	White	Male	
32537	Private	Dropout	Divorced	Adm-clerical	Unmarried	White	Female	
32550	State-gov	Dropout	Divorced	Adm-clerical	Other-relative	White	Female	
32551	Self-emp-not-inc	Dropout	Married-civ-spouse	Craft-repair	Husband	White	Male	
32556	Private	Dropout	Never-married	Protective-serv	Not-in-family	White	Male	

6373 rows × 11 columns

In [39]:

```
df2 = pd.concat([primary,secondary1,secondary2,secondary3,secondary4,HS1,HS2,HSG,Bach1,
Bach2,Mast,Doc,pre,Drop])
```

In [40]:

```
df2
```

Out[40]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
26	Private	primary	Married-civ-spouse	Craft-repair	Not-in-family	White	Male	
219	Self-emp-not-inc	primary	Married-civ-spouse	Transport-moving	Husband	White	Male	
1258	Self-emp-not-inc	primary	Widowed	Craft-repair	Other-relative	White	Female	
2541	Self-emp-not-inc	primary	Married-civ-spouse	Exec-managerial	Husband	White	Male	
3485	Private	primary	Married-civ-spouse	Farming-fishing	Husband	White	Male	
...	...	...	...	...	...	...	...	
32530	Private	Dropout	Never-married	Adm-clerical	Own-child	White	Male	
32537	Private	Dropout	Divorced	Adm-clerical	Unmarried	White	Female	
32550	State-gov	Dropout	Divorced	Adm-clerical	Other-relative	White	Female	
32551	Self-emp-not-inc	Dropout	Married-civ-spouse	Craft-repair	Husband	White	Male	
32556	Private	Dropout	Never-married	Protective-serv	Not-in-family	White	Male	

26117 rows × 11 columns

In [41]:

```
df2.reset_index(drop=True, inplace=True)
```

In [42]:

```
df2
```

Out[42]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.p
0	Private	primary	Married-civ-spouse	Craft-repair	Not-in-family	White	Male	
1	Self-emp-not-inc	primary	Married-civ-spouse	Transport-moving	Husband	White	Male	
2	Self-emp-not-inc	primary	Widowed	Craft-repair	Other-relative	White	Female	
3	Self-emp-not-inc	primary	Married-civ-spouse	Exec-managerial	Husband	White	Male	
4	Private	primary	Married-civ-spouse	Farming-fishing	Husband	White	Male	
...	...	...	...	...	...	...	...	
26112	Private	Dropout	Never-married	Adm-clerical	Own-child	White	Male	
26113	Private	Dropout	Divorced	Adm-clerical	Unmarried	White	Female	
26114	State-gov	Dropout	Divorced	Adm-clerical	Other-relative	White	Female	
26115	Self-emp-not-inc	Dropout	Married-civ-spouse	Craft-repair	Husband	White	Male	
26116	Private	Dropout	Never-married	Protective-serv	Not-in-family	White	Male	

26117 rows × 11 columns

In [43]:

```
df_temp=df2[df2.income == ">50K"]
df_temp
```

Out[43]:

	workclass	education	marital.status	occupation	relationship	race	sex	hours.per.
3	Self-emp-not-inc	primary	Married-civ-spouse	Exec-managerial	Husband	White	Male	
16	Private	primary	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
43	Private	primary	Married-civ-spouse	Farming-fishing	Husband	White	Male	
58	Private	primary	Married-civ-spouse	Exec-managerial	Husband	White	Male	
80	Private	primary	Married-civ-spouse	Craft-repair	Husband	Black	Male	
...	...	...	...	...	...	...	...	
26090	Private	Dropout	Married-civ-spouse	Other-service	Husband	White	Male	
26092	Private	Dropout	Married-civ-spouse	Craft-repair	Husband	White	Male	
26097	Self-emp-not-inc	Dropout	Married-spouse-absent	Craft-repair	Own-child	White	Male	
26101	Private	Dropout	Married-civ-spouse	Sales	Husband	White	Male	
26111	Private	Dropout	Married-civ-spouse	Exec-managerial	Husband	White	Male	

6479 rows × 11 columns

grp2=df2.groupby("sex") number=[] for sex, df_temp in grp2: number.append(df_temp["sex"].count())

plt.rcParams['patch.edgecolor'] = 'black' cols=['c','m'] plt.pie(number, labels=['female','male'], colors=cols) plt.title("different sex woking above 50K") plt.show()

IMPORTING TRAIN TEST SPLIT

In [53]:

```
from sklearn.model_selection import train_test_split
```

In [54]:

```
X_train, X_test, y_train, y_test = train_test_split(features, label, test_size=0.3,rand
om_state=12000)
```

NAIVE BAYES

In [55]:

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
```

In [56]:

```
from sklearn import metrics
```

In [57]:

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
gnb_acc=metrics.accuracy_score(y_test, y_pred)
```

```
Accuracy: 0.7595712098009189
```

In [58]:

```
from sklearn.metrics import accuracy_score,recall_score,precision_score,confusion_matri
x,f1_score
print("precision score : "+str(precision_score(y_test, y_pred))) # tp/tp+fp
print("accuracy score : "+str(accuracy_score(y_test, y_pred))) # total correct
print("recall score : "+str(recall_score(y_test, y_pred)))   # tp/tp+fn
print("f1 score : "+str(f1_score(y_test, y_pred)))
average_precision=precision_score(y_test, y_pred)
```

```
precision score : 0.513671875
accuracy score : 0.7595712098009189
recall score : 0.672978505629478
f1 score : 0.5826318121400089
```

ABOVE SCORES ARE DERIVED BY APPLYING NAIVE BAYES CLASSIFICATION

KNN CLASSIFICATION

In [64]:

```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [65]:

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
knn_acc=metrics.accuracy_score(y_test, y_pred)
```

```
Accuracy: 0.790454313425217
```

In [66]:

```
from sklearn.metrics import accuracy_score,recall_score,precision_score,confusion_matri
x,f1_score
print("precision score : "+str(precision_score(y_test, y_pred))) # tp/tp+fp
print("accuracy score : "+str(accuracy_score(y_test, y_pred))) # total correct
print("recall score : "+str(recall_score(y_test, y_pred)))   # tp/tp+fn
print("f1 score : "+str(f1_score(y_test, y_pred)))
average_precision=precision_score(y_test, y_pred)
```

```
precision score : 0.5807453416149069
accuracy score : 0.790454313425217
recall score : 0.5742067553735927
f1 score : 0.5774575398867731
```

THESE SCORES ARE DERIVED FROM KNN CLASSIFICATION

In [67]:

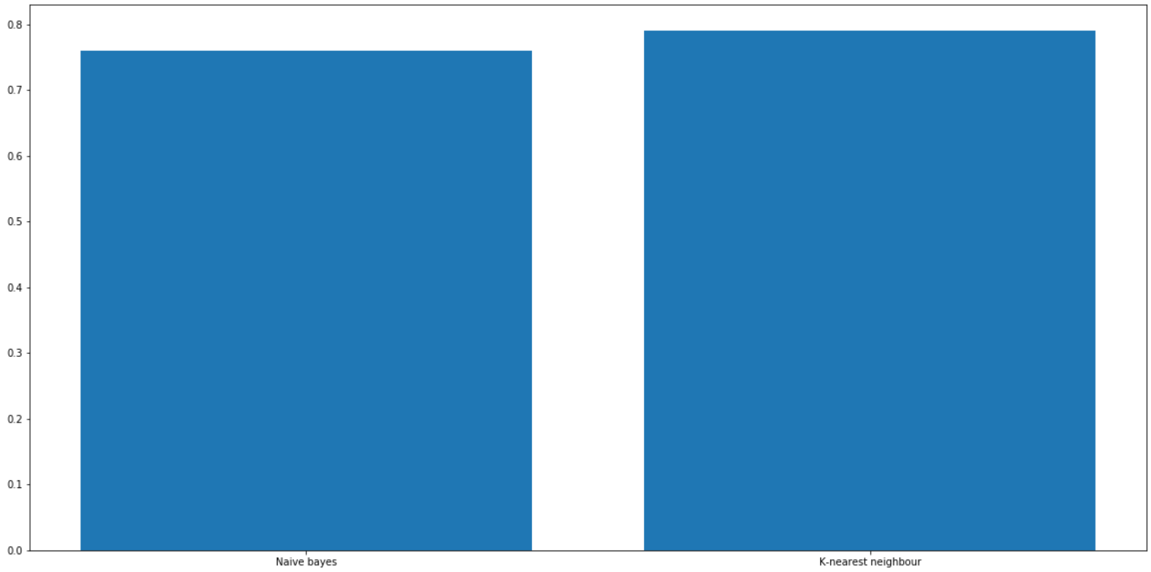
```
predicted= clf.predict([[2, 7, 2, 2, 1, 4, 1, 32, 25, 1]])
print("Predicted Value:", predicted)
```

```
Predicted Value: [0]
```

COMPARISON OF NAIVE BAYES AND KNN CLASSIFICATION

In [69]:

```
y=[gnb_acc,knn_acc]
x=['Naive bayes', 'K-nearest neighbour']
plt.bar(x,y)
plt.show()
```



BY THE GRAPH ABOVE WE CAN SEE THAT KNN IS MORE PREFERABLE THAN NAIVE BAYES IN THIS DATASET AS THE PRECISSION,ACCURACY AND RECALL VALUES ARE MORE IN KNN CLASSIFIER

In [ ]: