Double-click (or enter) to edit

Project: Machine Learning of Salary and Demographic Factors Name: Shaohua Feng Supervisor:

Double-click (or enter) to edit

```
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
# read in data loaded in google drive
file_path_1 = '/content/drive/My Drive/adult.data'
adult_1= pd.read_csv(file_path_1,header=None)
file_path_2 = '/content/drive/My Drive/adult.test.txt'
adult_2= pd.read_csv(file_path_2,header=None)
adult=pd.concat([adult_1, adult_2], ignore_index=True)
```

```
# add column names
cols=['age','workclass','fnlwgt','education','education-num','marital-status','occupation','relationship','race','sex','capital-gain','capital-loss','hours-per-week','native-country','l
adult.columns=cols
adult.head(10)
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	l
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-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	<

```
# add y column to data frame. y=1 for label '>50k' and y=0 for label '<=50k'
adult['y']=np.where(adult['label']==' >50K',1,0)
adult.head(20)
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	na
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	
10	37	Private	280464	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	80	
11	30	State-gov	141297	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	
12	23	Private	122272	Bachelors	13	Never-married	Adm-clerical	Own-child	White	Female	0	0	30	
13	32	Private	205019	Assoc-acdm	12	Never-married	Sales	Not-in-family	Black	Male	0	0	50	
14	40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	0	0	40	

```
print(adult.describe())
adult.dtypes
adult.info()
```

```
                age        fnlwgt  education-num  capital-gain  capital-loss  \
count  48842.000000  4.884200e+04   48842.000000  48842.000000  48842.000000
mean      38.643585  1.896641e+05      10.078089   1079.067626     87.502314
std       13.710510  1.056040e+05       2.570973   7452.019058    403.004552
min       17.000000  1.228500e+04       1.000000      0.000000      0.000000
25%       28.000000  1.175505e+05       9.000000      0.000000      0.000000
50%       37.000000  1.781445e+05      10.000000      0.000000      0.000000
75%       48.000000  2.376420e+05      12.000000      0.000000      0.000000
max       90.000000  1.490400e+06      16.000000  99999.000000   4356.000000

       hours-per-week             y
count    48842.000000  48842.000000
mean        40.422382      0.160538
std         12.391444      0.367108
min          1.000000      0.000000
25%         40.000000      0.000000
50%         40.000000      0.000000
75%         45.000000      0.000000
max         99.000000      1.000000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             48842 non-null  int64
 1   workclass       48842 non-null  object
 2   fnlwgt          48842 non-null  int64
 3   education       48842 non-null  object
 4   education-num   48842 non-null  int64
```

```
 5   marital-status  48842 non-null  object
 6   occupation      48842 non-null  object
 7   relationship    48842 non-null  object
 8   race            48842 non-null  object
 9   sex             48842 non-null  object
 10  capital-gain    48842 non-null  int64
 11  capital-loss    48842 non-null  int64
 12  hours-per-week  48842 non-null  int64
 13  native-country  48842 non-null  object
 14  label           48842 non-null  object
 15  y               48842 non-null  int64
dtypes: int64(7), object(9)
memory usage: 6.0+ MB
```

```
# explore: find factor levels
print(adult['workclass'].unique())
print(adult['occupation'].unique())
print(adult['native-country'].unique())
print(type(adult['occupation']))
```

```
[' State-gov' ' Self-emp-not-inc' ' Private' ' Federal-gov' ' Local-gov'
 ' ?' ' Self-emp-inc' ' Without-pay' ' Never-worked']
[' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specialty'
 ' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
 ' Farming-fishing' ' Machine-op-inspct' ' Tech-support' ' ?'
 ' Protective-serv' ' Armed-Forces' ' Priv-house-serv']
[' United-States' ' Cuba' ' Jamaica' ' India' ' ?' ' Mexico' ' South'
 ' Puerto-Rico' ' Honduras' ' England' ' Canada' ' Germany' ' Iran'
 ' Philippines' ' Italy' ' Poland' ' Columbia' ' Cambodia' ' Thailand'
 ' Ecuador' ' Laos' ' Taiwan' ' Haiti' ' Portugal' ' Dominican-Republic'
 ' El-Salvador' ' France' ' Guatemala' ' China' ' Japan' ' Yugoslavia'
 ' Peru' ' Outlying-US(Guam-USVI-etc)' ' Scotland' ' Trinadad&Tobago'
 ' Greece' ' Nicaragua' ' Vietnam' ' Hong' ' Ireland' ' Hungary'
 ' Holand-Netherlands']
<class 'pandas.core.series.Series'>
```

```
adult['workclass']=adult['workclass'].replace(' ?',None)
adult['occupation']=adult['occupation'].replace(' ?',None)
adult['native-country']=adult['native-country'].replace(' ?',None)
```

```
print(adult['occupation'].unique())
print(adult['occupation'].unique())
print(adult['occupation'].unique())
```

```
[' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specialty'
 ' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
 ' Farming-fishing' ' Machine-op-inspct' ' Tech-support' None
 ' Protective-serv' ' Armed-Forces' ' Priv-house-serv']
[' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specialty'
 ' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
 ' Farming-fishing' ' Machine-op-inspct' ' Tech-support' None
 ' Protective-serv' ' Armed-Forces' ' Priv-house-serv']
[' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specialty'
 ' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
 ' Farming-fishing' ' Machine-op-inspct' ' Tech-support' None
 ' Protective-serv' ' Armed-Forces' ' Priv-house-serv']
```

```
# check how many missing vales in columns workclass, occupation and native-country
print(adult['workclass'].isnull().sum())
```

```
print(adult['occupation'].isnull().sum())
print(adult['native-country'].isnull().sum())
```

```
2799
2809
857
```

```
# charactegorical columns
cols_cat=['workclass','fnlwgt','education','marital-status','occupation','relationship','race','sex','native-country']

for x in cols_cat:
  adult[x] = adult[x].astype('category')
  #print(x)

adult.dtypes
```

```
age                  int64
workclass         category
fnlwgt            category
education         category
education-num        int64
marital-status    category
occupation        category
relationship      category
race              category
sex               category
capital-gain         int64
capital-loss         int64
hours-per-week       int64
native-country    category
label               object
y                    int64
dtype: object
```

```
# delete missing value
adult_cleaned=adult.dropna()
print(len(adult_cleaned))
```

```
45222
```

```
print("Check for NaN values:")
print(adult_cleaned.isna().any())
```

```
Check for NaN values:
age               False
workclass         False
fnlwgt            False
education         False
education-num     False
marital-status    False
occupation        False
relationship      False
race              False
sex               False
capital-gain      False
capital-loss      False
hours-per-week    False
native-country    False
label             False
```

```
y                  False
dtype: bool
```

```
# Grouped by factors
factor_cols=['workclass','education','marital-status','occupation','relationship','race','sex','native-country']

for factor_col in factor_cols:
    # Group by the current factor column and calculate the mean
    grouped_data = adult.groupby(factor_col)['y'].mean().reset_index()

    # Sort the grouped data by percentage of salary>50,000
    sorted_data = grouped_data.sort_values(by='y', ascending=False)
    # Print the results
    print(f"Grouped by {factor_col}:\n{sorted_data}\n")

    # plot the sorted data
    #plt.bar(sorted_data[factor_col], sorted_data['y'])
    #plt.xlabel(f'factor_col')
    #plt.ylabel('% salary>50,000')
    #plt.title(f'Group Mean from Highest to Lowest')
    #plt.show



##############
#grouped_df = df.groupby('Category')['Value'].mean().reset_index()

# Sort the DataFrame by mean values
#sorted_df = grouped_df.sort_values(by='Value', ascending=False)

# Plot the sorted data
#plt.bar(sorted_df['Category'], sorted_df['Value'])
#plt.xlabel('Category')
#plt.ylabel('Mean Value')
#plt.title('Group Mean from Highest to Lowest')
#plt.show()


```

40	Yugoslavia	0.260870
23	Japan	0.260870
0	Cambodia	0.250000
21	Italy	0.238095
8	England	0.236220
1	Canada	0.214286
10	Germany	0.213592
29	Philippines	0.206780
16	Hong	0.200000
4	Cuba	0.181159
2	China	0.163934
38	United-States	0.163602
11	Greece	0.163265
17	Hungary	0.157895
33	Scotland	0.142857
34	South	0.139130
30	Poland	0.137931
20	Ireland	0.135135
36	Thailand	0.100000
22	Jamaica	0.094340
6	Ecuador	0.088889
24	Laos	0.086957
37	Trinadad&Tobago	0.074074
32	Puerto-Rico	0.065217
31	Portugal	0.059701
39	Vietnam	0.058140
7	El-Salvador	0.058065
13	Haiti	0.053333
15	Honduras	0.050000
28	Peru	0.043478
26	Nicaragua	0.040816
25	Mexico	0.034700
12	Guatemala	0.034091
3	Columbia	0.023529
5	Dominican-Republic	0.019417
27	Outlying-US(Guam-USVI-etc)	0.000000
14	Holand-Netherlands	0.000000

```
# I am very interested in the relationship between salary level and race and education combination
factor_columns = ['race', 'education']

for combination in adult.groupby(factor_columns):
    group_name = combination[0]
    group_data = combination[1]
    mean_salary = group_data['y'].mean()

    print(f"Factors: {', '.join(f'{col}={val}' for col, val in zip(factor_columns, group_name))} | Salary>50,000: {mean_salary}")

```

```
Factors: race= Black, education= 1st-4th | Salary>50,000: 0.041666666666666664
Factors: race= Black, education= 5th-6th | Salary>50,000: 0.0
Factors: race= Black, education= 7th-8th | Salary>50,000: 0.022222222222222223
Factors: race= Black, education= 9th | Salary>50,000: 0.036036036036036036
Factors: race= Black, education= Assoc-acdm | Salary>50,000: 0.11801242236024845
Factors: race= Black, education= Assoc-voc | Salary>50,000: 0.10909090909090909
Factors: race= Black, education= Bachelors | Salary>50,000: 0.19047619047619047
Factors: race= Black, education= Doctorate | Salary>50,000: 0.5625
Factors: race= Black, education= HS-grad | Salary>50,000: 0.048314606741573035
Factors: race= Black, education= Masters | Salary>50,000: 0.27972027972027974
Factors: race= Black, education= Preschool | Salary>50,000: 0.0
Factors: race= Black, education= Prof-school | Salary>50,000: 0.38095238095238093
Factors: race= Black, education= Some-college | Salary>50,000: 0.07977736549165121
Factors: race= Other, education= 10th | Salary>50,000: 0.09090909090909091
Factors: race= Other, education= 11th | Salary>50,000: 0.0
Factors: race= Other, education= 12th | Salary>50,000: 0.0
Factors: race= Other, education= 1st-4th | Salary>50,000: 0.0
Factors: race= Other, education= 5th-6th | Salary>50,000: 0.043478260869565216
Factors: race= Other, education= 7th-8th | Salary>50,000: 0.0
Factors: race= Other, education= 9th | Salary>50,000: 0.0
Factors: race= Other, education= Assoc-acdm | Salary>50,000: 0.2
Factors: race= Other, education= Assoc-voc | Salary>50,000: 0.0
Factors: race= Other, education= Bachelors | Salary>50,000: 0.1
Factors: race= Other, education= Doctorate | Salary>50,000: 0.3333333333333333
Factors: race= Other, education= HS-grad | Salary>50,000: 0.01904761904761905
Factors: race= Other, education= Masters | Salary>50,000: 0.15384615384615385
Factors: race= Other, education= Preschool | Salary>50,000: 0.0
Factors: race= Other, education= Prof-school | Salary>50,000: 0.8
Factors: race= Other, education= Some-college | Salary>50,000: 0.08235294117647059
Factors: race= White, education= 10th | Salary>50,000: 0.046632124352331605
Factors: race= White, education= 11th | Salary>50,000: 0.03367003367003367
Factors: race= White, education= 12th | Salary>50,000: 0.05242718446601942
Factors: race= White, education= 1st-4th | Salary>50,000: 0.025510204081632654
Factors: race= White, education= 5th-6th | Salary>50,000: 0.02891566265060241
Factors: race= White, education= 7th-8th | Salary>50,000: 0.04645476772616137
Factors: race= White, education= 9th | Salary>50,000: 0.03600654664484452
Factors: race= White, education= Assoc-acdm | Salary>50,000: 0.17178362573099415
Factors: race= White, education= Assoc-voc | Salary>50,000: 0.18469217970049917
Factors: race= White, education= Bachelors | Salary>50,000: 0.28646573784475404
Factors: race= White, education= Doctorate | Salary>50,000: 0.5247148288973384
Factors: race= White, education= HS-grad | Salary>50,000: 0.11518637484126391
Factors: race= White, education= Masters | Salary>50,000: 0.370954003407155
Factors: race= White, education= Preschool | Salary>50,000: 0.0
Factors: race= White, education= Prof-school | Salary>50,000: 0.5106951871657754
Factors: race= White, education= Some-college | Salary>50,000: 0.13510603940144256
```

```
# correlation matrix
cor_matrix = adult_cleaned.corr()
print(cor_matrix)
```

```
                     age  education-num  capital-gain  capital-loss  \
age             1.000000       0.037623      0.079683      0.059351
education-num   0.037623       1.000000      0.126907      0.081711
capital-gain    0.079683       0.126907      1.000000     -0.032102
capital-loss    0.059351       0.081711     -0.032102      1.000000
hours-per-week  0.101992       0.146206      0.083880      0.054195
y               0.182661       0.260062      0.168588      0.115861

                hours-per-week         y
age                   0.101992  0.182661
education-num         0.146206  0.260062
capital-gain          0.083880  0.168588
capital-loss          0.054195  0.115861
```

```
hours-per-week        1.000000  0.177195
y                     0.177195  1.000000
<ipython-input-17-d8ef474a4b8f>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid c
  cor_matrix = adult_cleaned.corr()
```