import the packages

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Read the data

In [2]:
```
visa_df=pd.read_csv(r"C:\Users\omkar\OneDrive\Documents\Data science\Naresh IT\N
visa_df.head(2)
```

Out[2]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training
0	EZYV01	Asia	High School	N	N
1	EZYV02	Asia	Master's	Y	N

Standardization

- Standardization is a technqiue scale all the data under one scale
- Different columns has different values also different units
- Some column values has bigger values, some column values has lesser values
- So it is important to keep all the values under one scale
- We have two methods are there
  - Standardization
  - Normalization
- Standarization
  - Z-score
  - Z score varies values from -3 to 3

$$Z = \frac{x - \mu}{\sigma}$$

- Normalization
  - min max scalar
  - values ranges from 0 to 1

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

task

- step-1: take prevailing wage column : **visa_df['prevailing_wage']**
- step-2: calculate the mean valye prevailing wage:mean= **visa_df['prevailing_wage'].mean**
- step-3: calculate the standard deviation of prevailaing vage:std= **visa_df['prevailing_wage'].std**
- step-4: Calculate the Nr : step-1 - step-2:Nr= **visa_df['prevailing_wage']-mean**
- step-5: divide the step4/step3

In [3]:
```
d={'name':"dharma",
  "age":[20,30]}
pd.DataFrame(d)
```

Out[3]:

	name	age
0	dharma	20
1	dharma	30

In [6]:
```
mean = visa_df['prevailing_wage'].mean()
std = visa_df['prevailing_wage'].std()
Nr= visa_df['prevailing_wage'] - mean
visa_df['prevailing_wage_z'] =Nr/std
visa_df
```

Out[6]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_t
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 13 columns

task-2

- Compare the two columns
- Get the maximum value and minimum value of the indexes
- The both indexes should match

idxmax- idxmin

In [10]:
```
maxx=visa_df["prevailing_wage"].idxmax(),visa_df["prevailing_wage_z"].idxmax()
minn=visa_df["prevailing_wage"].idxmin(),visa_df["prevailing_wage_z"].idxmin()

maxx,minn
```

Out[10]:
```
((21077, 21077), (20575, 20575))
```

In [12]:
```
max_val=visa_df["prevailing_wage"].max(),visa_df["prevailing_wage_z"].max()
min_val=visa_df["prevailing_wage"].min(),visa_df["prevailing_wage_z"].min()

max_val,min_val
```

Out[12]:
```
((319210.27, 4.634101837909902), (2.1367, -1.4096818992891214))
```

StandardScalar

- StandardScalar same as Z-score but by using pacakge
- It is under sklearn package
- In the sklearn we have preprocessing

- Read the package
- Save the package
- Apply fit transform
- Compare 3 coumns
  - One is original
  - Manually we did z-score
  - Column with package

In [13]:
```
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
visa_df['prevailing_wage_ss']=ss.fit_transform(visa_df[['prevailing_wage']])
```

- Single square bracket is series
- Double square bracket is Data frame
- Whenever you see the shape error apply double square bracket

In [14]:
```
visa_df[['prevailing_wage','prevailing_wage_z','prevailing_wage_ss']]
```

Out[14]:

	prevailing_wage	prevailing_wage_z	prevailing_wage_ss
0	592.2029	-1.398510	-1.398537
1	83425.6500	0.169832	0.169835
2	122996.8600	0.919060	0.919079
3	83434.0300	0.169991	0.169994
4	149907.3900	1.428576	1.428604
...	...	...	...
25475	77092.5700	0.049923	0.049924
25476	279174.7900	3.876083	3.876159
25477	146298.8500	1.360253	1.360280
25478	86154.7700	0.221504	0.221509
25479	70876.9100	-0.067762	-0.067763

25480 rows × 3 columns

Normalization

- Read the data agian
- step-1: take prevailing wage column : **visa_df['prevailing_wage']**

- step-2: calculate the min value prevailing wage:min= **visa_df['prevailing_wage'].min**
- step-3: calculate the min value prevailing wage:max= **visa_df['prevailing_wage'].max**
- step-4: Calculate the Nr : step-1 - step-2:Nr= **visa_df['prevailing_wage']-min**
- step-5: DR= step-3-step-2
- step-6: divide the step4/step5

In [15]:
```
x_max = visa_df['prevailing_wage'].max()
x_min = visa_df['prevailing_wage'].min()
Nr = visa_df['prevailing_wage'] - x_min
visa_df['prevailing_wage_min_max'] = Nr/(x_max - x_min)
visa_df[['prevailing_wage', 'prevailing_wage_min_max']]
```

Out[15]:

	prevailing_wage	prevailing_wage_min_max
0	592.2029	0.001849
1	83425.6500	0.261345
2	122996.8600	0.385312
3	83434.0300	0.261371
4	149907.3900	0.469616
...	...	...
25475	77092.5700	0.241505
25476	279174.7900	0.874579
25477	146298.8500	0.458311
25478	86154.7700	0.269895
25479	70876.9100	0.222033

25480 rows × 2 columns

package name: **MinMaxScalar**

In [16]:
```
from sklearn.preprocessing import MinMaxScaler
mms = MinMaxScaler()
visa_df['prevailing_wage_min_max_ss'] = mms.fit_transform(visa_df[['prevailing_w

visa_df[['prevailing_wage','prevailing_wage_min_max','prevailing_wage_min_max_ss
```

Out[16]:

	prevailing_wage	prevailing_wage_min_max	prevailing_wage_min_max_ss
0	592.2029	0.001849	0.001849
1	83425.6500	0.261345	0.261345
2	122996.8600	0.385312	0.385312
3	83434.0300	0.261371	0.261371
4	149907.3900	0.469616	0.469616
...	...	...	...
25475	77092.5700	0.241505	0.241505
25476	279174.7900	0.874579	0.874579
25477	146298.8500	0.458311	0.458311
25478	86154.7700	0.269895	0.269895
25479	70876.9100	0.222033	0.222033

25480 rows × 3 columns

In [ ]: