In [79]:
```
import pandas as pd
import numpy as np
from sklearn import preprocessing
```

In [80]:
```
df=pd.read_csv("Placement_Data_Full_Class.csv")
```

In [81]:
```
#printing the dataframe
df
```

Out[81]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No
...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No

215 rows × 15 columns

In [82]:
```
df.head()
```

Out[82]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	e
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	

In [83]:
```
df.tail()
```

Out[83]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
210	211	M	80.6	Others	82.0	Others	Commerce	77.6	Comm&Mgmt	No
211	212	M	58.0	Others	60.0	Others	Science	72.0	Sci&Tech	No
212	213	M	67.0	Others	67.0	Others	Commerce	73.0	Comm&Mgmt	Yes
213	214	F	74.0	Others	66.0	Others	Commerce	58.0	Comm&Mgmt	No
214	215	M	62.0	Central	58.0	Others	Science	53.0	Comm&Mgmt	No

In [84]:
```
df.describe()
```

Out[84]:

	sl_no	ssc_p	hsc_p	degree_p	etest_p	mba_p	salary
count	215.000000	215.000000	215.000000	215.000000	215.000000	215.000000	148.000000
mean	108.000000	67.303395	66.333163	66.370186	72.100558	62.278186	288655.405405
std	62.209324	10.827205	10.897509	7.358743	13.275956	5.833385	93457.452420
min	1.000000	40.890000	37.000000	50.000000	50.000000	51.210000	200000.000000
25%	54.500000	60.600000	60.900000	61.000000	60.000000	57.945000	240000.000000
50%	108.000000	67.000000	65.000000	66.000000	71.000000	62.000000	265000.000000
75%	161.500000	75.700000	73.000000	72.000000	83.500000	66.255000	300000.000000
max	215.000000	89.400000	97.700000	91.000000	98.000000	77.890000	940000.000000

In [85]:
```
df.mean()
```

```
/tmp/ipykernel_9635/3698961737.py:1: FutureWarning: The default value
of numeric_only in DataFrame.mean is deprecated. In a future version,
it will default to False. In addition, specifying 'numeric_only=None'
is deprecated. Select only valid columns or specify the value of numer
ic_only to silence this warning.
  df.mean()
```

Out[85]:
```
sl_no          108.000000
ssc_p           67.303395
hsc_p           66.333163
degree_p        66.370186
etest_p         72.100558
mba_p           62.278186
salary      288655.405405
dtype: float64
```

In [86]:
```
df.mode()
```

Out[86]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
0	1	M	62.0	Central	63.0	Others	Commerce	65.0	Comm&Mgmt	No
1	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
210	211	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
211	212	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
212	213	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
213	214	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
214	215	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

215 rows × 15 columns

In [87]:
```
#quqntile is used for percentage of data 0.5 means 50% 0.25 means 25% e
df.quantile(0.5)
```

```
/tmp/ipykernel_9635/1219827425.py:2: FutureWarning: The default value
of numeric_only in DataFrame.quantile is deprecated. In a future versi
on, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  df.quantile(0.5)
```

Out[87]:
```
sl_no          108.0
ssc_p           67.0
hsc_p           65.0
degree_p        66.0
etest_p         71.0
mba_p           62.0
salary      265000.0
Name: 0.5, dtype: float64
```

In [88]:
```
df.count()
```

Out[88]:
```
sl_no             215
gender            215
ssc_p             215
ssc_b             215
hsc_p             215
hsc_b             215
hsc_s             215
degree_p          215
degree_t          215
workex            215
etest_p           215
specialisation    215
mba_p             215
status            215
salary            148
dtype: int64
```

In [89]:
```
df.shape
```

Out[89]:
```
(215, 15)
```

In [90]:
```
df.size
```

Out[90]:
```
3225
```

In [91]:
```
df.isna()
```

Out[91]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
210	False	False	False	False	False	False	False	False	False	False	False
211	False	False	False	False	False	False	False	False	False	False	False
212	False	False	False	False	False	False	False	False	False	False	False
213	False	False	False	False	False	False	False	False	False	False	False
214	False	False	False	False	False	False	False	False	False	False	False

215 rows × 15 columns

In [92]:
```
#checks the not available/null values
df.isna().sum()
```

Out[92]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary           67
dtype: int64
```

In [93]:
```
#gives you detail about the data type of the object
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   sl_no           215 non-null    int64  
 1   gender          215 non-null    object 
 2   ssc_p           215 non-null    float64
 3   ssc_b           215 non-null    object 
 4   hsc_p           215 non-null    float64
 5   hsc_b           215 non-null    object 
 6   hsc_s           215 non-null    object 
 7   degree_p        215 non-null    float64
 8   degree_t        215 non-null    object 
 9   workex          215 non-null    object 
 10  etest_p         215 non-null    float64
 11  specialisation  215 non-null    object 
 12  mba_p           215 non-null    float64
 13  status          215 non-null    object 
 14  salary          148 non-null    float64
dtypes: float64(6), int64(1), object(8)
memory usage: 25.3+ KB
```

In [94]:
```
df.min()
```

Out[94]:
```
sl_no                   1
gender                  F
ssc_p               40.89
ssc_b             Central
hsc_p                37.0
hsc_b             Central
hsc_s                Arts
degree_p             50.0
degree_t         Comm&Mgmt
workex                  No
etest_p              50.0
specialisation     Mkt&Fin
mba_p                51.21
status           Not Placed
salary             200000.0
dtype: object
```

In [95]:
```
df.max()
```

Out[95]:
```
sl_no                 215
gender                  M
ssc_p                89.4
ssc_b              Others
hsc_p                97.7
hsc_b              Others
hsc_s             Science
degree_p             91.0
degree_t         Sci&Tech
workex                Yes
etest_p              98.0
specialisation     Mkt&HR
mba_p                77.89
status             Placed
salary             940000.0
dtype: object
```

In [96]:
```
#A standard deviation (or σ) is a measure of how dispersed the data is
#Low standard deviation means data are clustered around the mean,
#and high standard deviation indicates data are more spread out.
df.std()
```

```
/tmp/ipykernel_9635/3332421159.py:4: FutureWarning: The default value
of numeric_only in DataFrame.std is deprecated. In a future version, i
t will default to False. In addition, specifying 'numeric_only=None' i
s deprecated. Select only valid columns or specify the value of numeri
c_only to silence this warning.
  df.std()
```

Out[96]:
```
sl_no          62.209324
ssc_p          10.827205
hsc_p          10.897509
degree_p        7.358743
etest_p        13.275956
mba_p           5.833385
salary      93457.452420
dtype: float64
```

In [97]:
```
#to check any particular column we do
print(df['gender'].min()) #alternatively can be done as  "df.gender.min
print(df['gender'].max())
print(df['gender'].count())
print(df['gender'].mode())
#std() , quantile(), mean() cannot be used with the string value column
```

```
F
M
215
0    M
Name: gender, dtype: object
```

In [98]:
```
#for numeric column ssc_p
print(df['ssc_p'].min())
print(df['ssc_p'].max())
print(df['ssc_p'].std())
print(df['ssc_p'].quantile(0.25))
print(df['ssc_p'].quantile(0.5))
print(df['ssc_p'].count())

#all methods can be done here as it is numeric column
```

```
40.89
89.4
10.827205398231452
60.599999999999994
67.0
215
```

In [99]:
```
#for checking the data types
df.dtypes
```

Out[99]:
```
sl_no              int64
gender            object
ssc_p            float64
ssc_b             object
hsc_p            float64
hsc_b             object
hsc_s             object
degree_p         float64
degree_t          object
workex            object
etest_p          float64
specialisation    object
mba_p            float64
status            object
salary           float64
dtype: object
```

In [100]:
```
#checking null values before performing the normalization
df.isna().sum()
```

Out[100]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary           67
dtype: int64
```

In [101]:
```
#the salary is having 67 null values or not available values
# we need to deal with this either by filing it or removing it
#we are going to fill it using fill na
sal=df['salary'].fillna(df['salary'].mean())
print(newdf.isna().sum())
```

```
0
```

In [102]:
```
#so we have new filled values column so we delete the old one and apper
df1=df.drop(['salary'],axis=1)
df1
```

Out[102]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No
...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No

215 rows × 14 columns

In [103]:
```
#concatinate the new sal column using the pandas.concat method it takes
#1. array , columns or data frame to concatenate and
#2 . Axis =1 means column and axis=0 means row
new_df=pd.concat([df1,sal],axis=1)
new_df
```

Out[103]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No
...	...	...	...	...	...	...	...	...	...	...
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&Mgmt	No
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&Tech	No
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&Mgmt	Yes
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&Mgmt	No
214	215	M	62.00	Central	58.00	Others	Science	53.00	Comm&Mgmt	No

215 rows × 15 columns

In [104]:
```
#now checking again for the null values
new_df.isna().sum()
```

Out[104]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary            0
dtype: int64
```

In [105]:
```
#using the LabelEncoder for transforming the categorical into quantitat
labelencoder=preprocessing.LabelEncoder()
new_df['gender']=labelencoder.fit_transform(new_df['gender'])
new_df['ssc_b']=labelencoder.fit_transform(new_df['ssc_b'])
new_df['hsc_b']=labelencoder.fit_transform(new_df['hsc_b'])
new_df['hsc_s']=labelencoder.fit_transform(new_df['hsc_s'])
new_df['degree_t']=labelencoder.fit_transform(new_df['degree_t'])
new_df['workex']=labelencoder.fit_transform(new_df['workex'])
new_df['specialisation']=labelencoder.fit_transform(new_df['specialisat
new_df['status']=labelencoder.fit_transform(new_df['status'])
new_df
```

Out[105]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p
0	1	1	67.00	1	91.00	1	1	58.00	2	0	55.0
1	2	1	79.33	0	78.33	1	2	77.48	2	1	86.5
2	3	1	65.00	0	68.00	0	0	64.00	0	0	75.0
3	4	1	56.00	0	52.00	0	2	52.00	2	0	66.0
4	5	1	85.80	0	73.60	0	1	73.30	0	0	96.8
...	...	...	...	...	...	...	...	...	...	...	...
210	211	1	80.60	1	82.00	1	1	77.60	0	0	91.0
211	212	1	58.00	1	60.00	1	2	72.00	2	0	74.0
212	213	1	67.00	1	67.00	1	1	73.00	0	1	59.0
213	214	0	74.00	1	66.00	1	1	58.00	0	0	70.0
214	215	1	62.00	0	58.00	1	2	53.00	0	0	89.0

215 rows × 15 columns

In [107]:
```
#generating column name for the normalization
column=[]
for columns in new_df:
    column.append(columns)
column
```

Out[107]:
```
['sl_no',
 'gender',
 'ssc_p',
 'ssc_b',
 'hsc_p',
 'hsc_b',
 'hsc_s',
 'degree_p',
 'degree_t',
 'workex',
 'etest_p',
 'specialisation',
 'mba_p',
 'status',
 'salary']
```

In [109]:
```
#now applying min max normalisation
from sklearn.preprocessing import MinMaxScaler
scaler =MinMaxScaler()
new_df=scaler.fit_transform(new_df)
new_df=pd.DataFrame(new_df,columns=column)
new_df
```

Out[109]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex
0	0.000000	1.0	0.538240	1.0	0.889621	1.0	0.5	0.195122	1.0	0.0
1	0.004673	1.0	0.792414	0.0	0.680890	1.0	1.0	0.670244	1.0	1.0
2	0.009346	1.0	0.497011	0.0	0.510708	0.0	0.0	0.341463	0.0	0.0
3	0.014019	1.0	0.311482	0.0	0.247117	0.0	1.0	0.048780	1.0	0.0
4	0.018692	1.0	0.925788	0.0	0.602965	0.0	0.5	0.568293	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
210	0.981308	1.0	0.818594	1.0	0.741351	1.0	0.5	0.673171	0.0	0.0
211	0.985981	1.0	0.352711	1.0	0.378913	1.0	1.0	0.536585	1.0	0.0
212	0.990654	1.0	0.538240	1.0	0.494234	1.0	0.5	0.560976	0.0	1.0
213	0.995327	0.0	0.682540	1.0	0.477759	1.0	0.5	0.195122	0.0	0.0
214	1.000000	1.0	0.435168	0.0	0.345964	1.0	1.0	0.073171	0.0	0.0

215 rows × 15 columns

In [ ]:

In [ ]: