In [70]:

```
# Importing the packages

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
```

In [71]:

```
df = pd.read_csv('C:/Users/dell/Desktop/Dataset.csv')
```

In [99]:

```
df.head()
```

Out[99]:

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi	job designation	education
0	Nabin	Technical	M	25000	NaN	yes	under graduate	15000	2	2
1	Saransh	Medical	M	45000	NaN	No	post graduate	10000	1	1
2	Serena	Technical	F	20000	NaN	yes	post graduate	18000	2	1
3	Manisha	Technical	F	15000	NaN	No	under graduate	3000	2	2
4	Rajiv	Technical	M	35000	NaN	yes	graduate	12000	2	0

In [73]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Name                20 non-null     object 
 1   Job                 20 non-null     object 
 2   Gender              20 non-null     object 
 3   Monthly Income      20 non-null     object 
 4   Rent amount paid    0 non-null      float64
 5   Look after family   20 non-null     object 
 6   Education           20 non-null     object 
 7   Monthly expendi     20 non-null     int64  
dtypes: float64(1), int64(1), object(6)
memory usage: 1.4+ KB
```

In [74]:

```
# Let's do some repalcing of values

df['Job'].replace('Techical', 'Technical', inplace=True)
df
```

Out[74]:

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi
0	Nabin	Technical	M	25000	NaN	yes	under garduate	15000
1	Saransh	Medical	M	45000	NaN	No	post graduate	10000

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi
2	Serena	Technical	F	20000	NaN	yes	post garduate	18000
3	Manisha	Technical	F	15000	NaN	No	under garduate	3000
4	Rajiv	Technical	M	35000	NaN	yes	graduate	12000
5	Dilnawaz	Technical	M	30000	NaN	yes	graduate	20000
6	Pratima	Finance	F	25000	NaN	No	graduate	5000
7	pooja	Medical	F	23000	NaN	No	under garduate	13000
8	Parichaya	Technical	M	6000	NaN	No	garduate	5000
9	Melisha	Technical	F	28000	NaN	yes	under garduate	10000
10	Anmol	Finance	M	25000	NaN	No	graduate	4000
11	Ankita	Medical	F	40000	NaN	No	graduate	10000
12	Dinesh	Technical	M	30000	NaN	yes	under garduate	20000
13	Dilip	Technical	M	30,000	NaN	yes	garduate	15000
14	Rohit	Technical	M	30000	NaN	No	graduate	5000
15	Faija	Finance	F	50000	NaN	yes	post garduate	25000
16	Rojesh	Medical	M	60000	NaN	No	post garduate	25000
17	Sabin	Medical	M	40000	NaN	No	graduate	10000
18	sargoon	Technical	F	40000	NaN	No	under garduate	10000
19	Sushma	Medical	F	15000	NaN	No	under garduate	5000

In [75]:

```
## Replacing


df['Education'].replace('garduate', 'graduate', inplace=True)

df['Education'].replace('post garduate', 'post graduate', inplace=True)

df['Education'].replace('under garduate', 'under graduate', inplace=True)
df
```

Out[75]:

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi
0	Nabin	Technical	M	25000	NaN	yes	under graduate	15000
1	Saransh	Medical	M	45000	NaN	No	post graduate	10000
2	Serena	Technical	F	20000	NaN	yes	post graduate	18000
3	Manisha	Technical	F	15000	NaN	No	under graduate	3000
4	Rajiv	Technical	M	35000	NaN	yes	graduate	12000
5	Dilnawaz	Technical	M	30000	NaN	yes	graduate	20000
6	Pratima	Finance	F	25000	NaN	No	graduate	5000
7	pooja	Medical	F	23000	NaN	No	under graduate	13000
8	Parichaya	Technical	M	6000	NaN	No	graduate	5000
9	Melisha	Technical	F	28000	NaN	yes	under graduate	10000
10	Anmol	Finance	M	25000	NaN	No	graduate	4000
11	Ankita	Medical	F	40000	NaN	No	graduate	10000
12	Dinesh	Technical	M	30000	NaN	yes	under graduate	20000
13	Dilip	Technical	M	30,000	NaN	yes	graduate	15000
14	Rohit	Technical	M	30000	NaN	No	graduate	5000
15	Faija	Finance	F	50000	NaN	yes	post graduate	25000
16	Rojesh	Medical	M	60000	NaN	No	post graduate	25000

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi
17	Sabin	Medical	M	40000	NaN	No	graduate	10000
18	sargoon	Technical	F	40000	NaN	No	under graduate	10000
19	Sushma	Medical	F	15000	NaN	No	under graduate	5000

In [76]:

```
## Let's make the use of label encoder job and education column

from sklearn.preprocessing import LabelEncoder
```

In [77]:

```
label_job = LabelEncoder()
label_education = LabelEncoder()
```

In [78]:

```
df['job designation'] = label_job.fit_transform(df['Job'])
df['education'] = label_education.fit_transform(df['Education'])
df
```

Out[78]:

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi	job designation	education
0	Nabin	Technical	M	25000	NaN	yes	under graduate	15000	2	2
1	Saransh	Medical	M	45000	NaN	No	post graduate	10000	1	1
2	Serena	Technical	F	20000	NaN	yes	post graduate	18000	2	1
3	Manisha	Technical	F	15000	NaN	No	under graduate	3000	2	2
4	Rajiv	Technical	M	35000	NaN	yes	graduate	12000	2	0
5	Dilnawaz	Technical	M	30000	NaN	yes	graduate	20000	2	0
6	Pratima	Finance	F	25000	NaN	No	graduate	5000	0	0
7	pooja	Medical	F	23000	NaN	No	under graduate	13000	1	2
8	Parichaya	Technical	M	6000	NaN	No	graduate	5000	2	0
9	Melisha	Technical	F	28000	NaN	yes	under graduate	10000	2	2
10	Anmol	Finance	M	25000	NaN	No	graduate	4000	0	0
11	Ankita	Medical	F	40000	NaN	No	graduate	10000	1	0
12	Dinesh	Technical	M	30000	NaN	yes	under graduate	20000	2	2
13	Dilip	Technical	M	30,000	NaN	yes	graduate	15000	2	0
14	Rohit	Technical	M	30000	NaN	No	graduate	5000	2	0
15	Faija	Finance	F	50000	NaN	yes	post graduate	25000	0	1
16	Rojesh	Medical	M	60000	NaN	No	post graduate	25000	1	1
17	Sabin	Medical	M	40000	NaN	No	graduate	10000	1	0
18	sargoon	Technical	F	40000	NaN	No	under graduate	10000	2	2
19	Sushma	Medical	F	15000	NaN	No	under graduate	5000	1	2

In [79]:

```
## Let's use one hot encoder for Gender

Gender_value = pd.get_dummies(df.Gender)
Gender_value
```

Out[79]:

	F	M
0	0	1
1	0	1
2	1	0
3	1	0
4	0	1
5	0	1
6	1	0
7	1	0
8	0	1
9	1	0
10	0	1
11	1	0
12	0	1
13	0	1
14	0	1
15	1	0
16	0	1
17	0	1
18	1	0
19	1	0

In [80]:

```
merged = pd.concat([df,Gender_value],axis = 'columns')
merged
```

Out[80]:

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi	job designation	education	F	M
0	Nabin	Technical	M	25000	NaN	yes	under graduate	15000	2	2	0	1
1	Saransh	Medical	M	45000	NaN	No	post graduate	10000	1	1	0	1
2	Serena	Technical	F	20000	NaN	yes	post graduate	18000	2	1	1	0
3	Manisha	Technical	F	15000	NaN	No	under graduate	3000	2	2	1	0
4	Rajiv	Technical	M	35000	NaN	yes	graduate	12000	2	0	0	1
5	Dilnawaz	Technical	M	30000	NaN	yes	graduate	20000	2	0	0	1
6	Pratima	Finance	F	25000	NaN	No	graduate	5000	0	0	1	0
7	pooja	Medical	F	23000	NaN	No	under graduate	13000	1	2	1	0

	Name	Job	Gender	Monthly Income	Rent amount paid	Look after family	Education	Monthly expendi	job designation	education	F	M
8	Parichaya	Technical	M	6000	NaN	No	graduate	5000	2	0	0	1
9	Melisha	Technical	F	28000	NaN	yes	under graduate	10000	2	2	1	0
10	Anmol	Finance	M	25000	NaN	No	graduate	4000	0	0	0	1
11	Ankita	Medical	F	40000	NaN	No	graduate	10000	1	0	1	0
12	Dinesh	Technical	M	30000	NaN	yes	under graduate	20000	2	2	0	1
13	Dilip	Technical	M	30,000	NaN	yes	graduate	15000	2	0	0	1
14	Rohit	Technical	M	30000	NaN	No	graduate	5000	2	0	0	1
15	Faija	Finance	F	50000	NaN	yes	post graduate	25000	0	1	1	0
16	Rojesh	Medical	M	60000	NaN	No	post graduate	25000	1	1	0	1
17	Sabin	Medical	M	40000	NaN	No	graduate	10000	1	0	0	1
18	sargoon	Technical	F	40000	NaN	No	under graduate	10000	2	2	1	0
19	Sushma	Medical	F	15000	NaN	No	under graduate	5000	1	2	1	0

In [91]:

```
## Now we will drop the unnecessary columns
# 0 represent finance,1 represent medical and 2 represents technical. Likewise, 0 represe
nt graduate,1 represent
# post graduate and 2 represents under graduate

final_df = merged.drop(['Job','Gender','Education','M'],axis = 'columns')
final_df
```

Out[91]:

	Name	Monthly Income	Rent amount paid	Look after family	Monthly expendi	job designation	education	F
0	Nabin	25000	NaN	yes	15000	2	2	0
1	Saransh	45000	NaN	No	10000	1	1	0
2	Serena	20000	NaN	yes	18000	2	1	1
3	Manisha	15000	NaN	No	3000	2	2	1
4	Rajiv	35000	NaN	yes	12000	2	0	0
5	Dilnawaz	30000	NaN	yes	20000	2	0	0
6	Pratima	25000	NaN	No	5000	0	0	1
7	pooja	23000	NaN	No	13000	1	2	1
8	Parichaya	6000	NaN	No	5000	2	0	0
9	Melisha	28000	NaN	yes	10000	2	2	1
10	Anmol	25000	NaN	No	4000	0	0	0
11	Ankita	40000	NaN	No	10000	1	0	1
12	Dinesh	30000	NaN	yes	20000	2	2	0
13	Dilip	30,000	NaN	yes	15000	2	0	0
14	Rohit	30000	NaN	No	5000	2	0	0
15	Faija	50000	NaN	yes	25000	0	1	1
16	Rojesh	60000	NaN	No	25000	1	1	0
17	Sabin	40000	NaN	No	10000	1	0	0
18	sargoon	40000	NaN	No	10000	2	2	1
19	Sushma	15000	NaN	No	5000	1	2	1

In [100]:

```
final_df
```

Out[100]:

	Name	Monthly Income	Rent amount paid	Look after family	Monthly expendi	job designation	education	F
0	Nabin	25000	NaN	yes	15000	2	2	0
1	Saransh	45000	NaN	No	10000	1	1	0
2	Serena	20000	NaN	yes	18000	2	1	1
3	Manisha	15000	NaN	No	3000	2	2	1
4	Rajiv	35000	NaN	yes	12000	2	0	0
5	Dilnawaz	30000	NaN	yes	20000	2	0	0
6	Pratima	25000	NaN	No	5000	0	0	1
7	pooja	23000	NaN	No	13000	1	2	1
8	Parichaya	6000	NaN	No	5000	2	0	0
9	Melisha	28000	NaN	yes	10000	2	2	1
10	Anmol	25000	NaN	No	4000	0	0	0
11	Ankita	40000	NaN	No	10000	1	0	1
12	Dinesh	30000	NaN	yes	20000	2	2	0
13	Dilip	30,000	NaN	yes	15000	2	0	0
14	Rohit	30000	NaN	No	5000	2	0	0
15	Faija	50000	NaN	yes	25000	0	1	1
16	Rojesh	60000	NaN	No	25000	1	1	0
17	Sabin	40000	NaN	No	10000	1	0	0
18	sargoon	40000	NaN	No	10000	2	2	1
19	Sushma	15000	NaN	No	5000	1	2	1

In [104]:

```
final_df.rename(columns = {'Name':'name'},inplace = True)
final_df
```

Out[104]:

	Name	Monthly Income	Rent amount paid	Look after family	Monthly expendi	job designation	education	F
0	Nabin	25000	NaN	yes	15000	2	2	0
1	Saransh	45000	NaN	No	10000	1	1	0
2	Serena	20000	NaN	yes	18000	2	1	1
3	Manisha	15000	NaN	No	3000	2	2	1
4	Rajiv	35000	NaN	yes	12000	2	0	0
5	Dilnawaz	30000	NaN	yes	20000	2	0	0
6	Pratima	25000	NaN	No	5000	0	0	1
7	pooja	23000	NaN	No	13000	1	2	1
8	Parichaya	6000	NaN	No	5000	2	0	0
9	Melisha	28000	NaN	yes	10000	2	2	1
10	Anmol	25000	NaN	No	4000	0	0	0
11	Ankita	40000	NaN	No	10000	1	0	1
12	Dinesh	30000	NaN	yes	20000	2	2	0

	Name	Monthly Income	Rent amount paid	Look after family	Monthly expendi	job designation	education	F
13	Dilip	30,000	NaN	yes	15000	2	0	0
14	Rohit	30000	NaN	No	5000	2	0	0
15	Faija	50000	NaN	yes	25000	0	1	1
16	Rojesh	60000	NaN	No	25000	1	1	0
17	Sabin	40000	NaN	No	10000	1	0	0
18	sargoon	40000	NaN	No	10000	2	2	1
19	Sushma	15000	NaN	No	5000	1	2	1

In [105]:

```
new_df = final_df[['Monthly expendi','job designation','education','F']]
new_df
```

Out[105]:

	Monthly expendi	job designation	education	F
0	15000	2	2	0
1	10000	1	1	0
2	18000	2	1	1
3	3000	2	2	1
4	12000	2	0	0
5	20000	2	0	0
6	5000	0	0	1
7	13000	1	2	1
8	5000	2	0	0
9	10000	2	2	1
10	4000	0	0	0
11	10000	1	0	1
12	20000	2	2	0
13	15000	2	0	0
14	5000	2	0	0
15	25000	0	1	1
16	25000	1	1	0
17	10000	1	0	0
18	10000	2	2	1
19	5000	1	2	1

In [108]:

```
## Let's create a target column

target_Col = final_df[['Monthly Income']]
target_Col
```

Out[108]:

	Monthly Income
0	25000
1	45000
2	20000
3	15000
4	35000

	Monthly Income
5	30000
6	25000
7	23000
8	6000
9	28000
10	25000
11	40000
12	30000
13	30,000
14	30000
15	50000
16	60000
17	40000
18	40000
19	15000

In [114]:

```
## Let's convert 30,000 into 30000 because , can create problem while model fitting

target_Col['Monthly Income'].replace('30,000', '30000', inplace=True)
target_Col
```

Out[114]:

	Monthly Income
0	25000
1	45000
2	20000
3	15000
4	35000
5	30000
6	25000
7	23000
8	6000
9	28000
10	25000
11	40000
12	30000
13	30000
14	30000
15	50000
16	60000
17	40000
18	40000
19	15000

In [115]:

```
## Let's create a fit model
```

```
reg = linear_model.LinearRegression()
reg.fit(new_df,target_Col)
```

Out[115]:

```
LinearRegression()
```

In [117]:

```
reg.coef_
```

Out[117]:

```
array([[ 1.08987352e+00, -5.79921387e+03, -6.24898355e+02,
        -3.31975292e+03]])
```

In [119]:

```
reg.intercept_
```

Out[119]:

```
array([27696.71454805])
```

In [122]:

```
reg.predict([[18000,2,1,1]])
```

Out[122]:

```
array([[31771.35883832]])
```

In [124]:

```
reg.predict([[25000,0,1,1]])
```

Out[124]:

```
array([[50998.90120631]])
```

In [125]:

```
reg.predict([[5000,1,2,1]])
```

Out[125]:

```
array([[22777.31863114]])
```