```
#MAJOR PROJECT 1
#Shoe size dataset


#1. Import the data
import pandas as pd
df=pd.read_csv('/content/wo_men.csv')
df
```

	time	sex	height	shoe_size
0	04.10.2016 17:58:51	woman	160.0	40.0
1	04.10.2016 17:58:59	woman	171.0	39.0
2	04.10.2016 18:00:15	woman	174.0	39.0
3	04.10.2016 18:01:17	woman	176.0	40.0
4	04.10.2016 18:01:22	man	195.0	46.0
...	...	...	...	...
96	17.10.2016 12:37:09	woman	170.0	39.0
97	17.10.2016 13:12:48	woman	183.0	39.0
98	19.10.2016 17:07:53	woman	173.0	40.0
99	29.10.2016 20:28:33	woman	160.0	37.0
100	30.10.2016 11:57:57	woman	168.0	39.0

101 rows × 4 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   time       101 non-null    object 
 1   sex        100 non-null    object 
 2   height     100 non-null    float64
 3   shoe_size  100 non-null    float64
dtypes: float64(2), object(2)
memory usage: 3.3+ KB
```

```
df=df.dropna()
```

```
df.shape
```

```
(100, 4)
```

```
df.size
```

```
200
```

```
#2. Preprocessing
df=df.drop(['time','sex'],axis=1)
df
```

	height	shoe_size
0	160.0	40.0
1	171.0	39.0
2	174.0	39.0
3	176.0	40.0
4	195.0	46.0
...	...	...
96	170.0	39.0
97	183.0	39.0
98	173.0	40.0
99	160.0	37.0
100	168.0	39.0

100 rows × 2 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100 entries, 0 to 100
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   height     100 non-null    float64
 1   shoe_size  100 non-null    float64
dtypes: float64(2)
memory usage: 2.3 KB
```

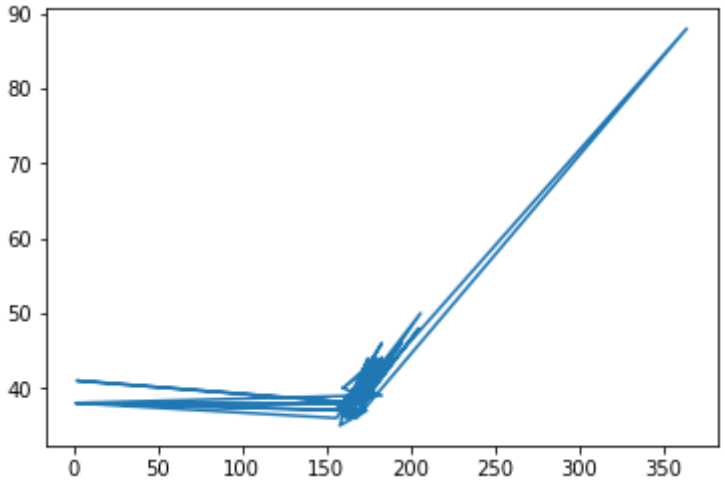
```
df.groupby(['height','shoe_size']).size()
```

```
height  shoe_size
1.63    38.0         1
1.68    38.0         1
1.73    38.0         1
1.84    41.0         1
155.00  37.0         1
                    ..
187.00  44.0         1
195.00  46.0         1
205.00  48.0         1
206.00  50.0         1
```

```
364.00  88.0         1
Length: 68, dtype: int64
```

```
#3. Data Visualisation
import matplotlib.pyplot as plt
plt.plot(df['height'],df['shoe_size'])
```

```
[<matplotlib.lines.Line2D at 0x7f448d7e7690>]
```




```
#4. Divide into input and output
#input - height
#output- shoe_size
import numpy as np
x =df.iloc[:,0:1].values
x
```

```
array([[160.  ],
       [171.  ],
       [174.  ],
       [176.  ],
       [195.  ],
       [157.  ],
       [160.  ],
       [178.  ],
       [168.  ],
       [171.  ],
       [165.  ],
       [175.  ],
       [163.  ],
       [158.  ],
       [159.  ],
       [183.  ],
       [155.  ],
       [172.  ],
       [164.  ],
       [158.  ],
       [174.  ],
       [164.  ],
       [168.  ],
       [168.  ],
       [163.  ],
       [160.  ],
```

```
            [183.  ],
            [161.  ],
            [162.  ],
            [165.  ],
            [164.  ],
            [161.  ],
            [163.  ],
            [169.  ],
            [171.  ],
            [163.  ],
            [159.  ],
            [180.  ],
            [168.  ],
            [170.  ],
            [168.  ],
            [180.  ],
            [183.  ],
            [170.  ],
            [172.  ],
            [163.  ],
            [168.  ],
            [  1.84],
            [169.  ],
            [206.  ],
            [165.  ],
            [171.  ],
            [165.  ],
            [168.  ],
            [180.  ],
            [160.  ],
            [183.  ],
            [160.  ],
```

```
y=df.iloc[:,1].values
y
```

```
array([40. , 39. , 39. , 40. , 46. , 37. , 38. , 39. , 38. , 41. , 39. ,
       44. , 38. , 37. , 38. , 44. , 37. , 39. , 39. , 35. , 37. , 37. ,
       38. , 38. , 37. , 37. , 46. , 38. , 36. , 37. , 36. , 37. , 39. ,
       40. , 39. , 38. , 36. , 42. , 38. , 38. , 38. , 42. , 44. , 40. ,
       39. , 38. , 38. , 41. , 38. , 50. , 38. , 40. , 37. , 38. , 44. ,
       40. , 44. , 36.5, 40. , 39. , 37. , 38. , 38. , 44. , 40. , 42. ,
       39. , 88. , 36. , 39. , 42. , 48. , 36. , 42. , 42. , 41. , 36. ,
       38. , 37. , 38. , 38. , 39. , 39. , 38. , 38. , 39. , 40. , 40. ,
       38. , 38. , 38. , 41. , 44. , 40. , 37. , 39. , 39. , 40. , 37. ,
       39. ])
```

```
#5. Train and Test variables
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)
```

```
print(x.shape)
print(x_train.shape)
print(x_test.shape)
```

```
(100, 1)
(75, 1)
```

```
(25, 1)
```

```
print(y.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(100,)
(75,)
(25,)
```

```
#6. Scaling the data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.fit_transform(x_test)
```

```
#7. Run a Classifier/Regressor/Clusterer
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
#8.MODEL FITTING
model.fit(x_train,y_train)
```

```
LinearRegression()
```

```
#9.PREDICT THE OUTPUT
y_pred = model.predict(x_test)
y_pred #predicted output
```

```
array([42.84591879, 37.41181763, 37.95522774, 30.34748611, 36.86840751,
       35.23817716, 27.63043553, 38.49863786, 41.21568844, 35.78158728,
       34.69476704, 40.12886821, 31.97771646, 29.26066588, 40.12886821,
       32.52112658, 34.69476704, 31.97771646, 35.23817716, 34.69476704,
       35.78158728, 37.41181763, 39.04204798, 54.80094135, 31.97771646])
```

```
y_test
```

```
array([46., 40., 39., 40., 41., 40., 37., 42., 44., 39., 38., 44., 37.,
       37., 39., 36., 38., 37., 40., 38., 40., 38., 40., 48., 38.])
```

```
#INDUVIDUAL PREDICTION
a=scaler.transform([[172]])
```

```
s=model.predict(a)
s
```

```
array([36.86840751])
```

```
a
```

```
array([[0.34]])
```

```
#As we can see that the induvidual accuracy is almost about the range 92% to 97%
#We  can say that the model is designed or created almost accurately.
```