In [3]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import preprocessing,svm
from sklearn.linear_model import LinearRegression
```

In [8]:
```
df=pd.read_csv(r"C:\Users\Lenovo\OneDrive\Desktop\Data Sets\BMI.csv")
df
```

Out[8]:

	Gender	Height in cm	Weight	Index	Height in m	Height Square	BMI
0	Male	174	96	4	1.74	3.0276	31.708284
1	Male	189	87	2	1.89	3.5721	24.355421
2	Female	185	110	4	1.85	3.4225	32.140248
3	Female	195	104	3	1.95	3.8025	27.350427
4	Male	149	61	3	1.49	2.2201	27.476240
...	...	...	...	...	...	...	...
495	Female	150	153	5	1.50	2.2500	68.000000
496	Female	184	121	4	1.84	3.3856	35.739603
497	Female	141	136	5	1.41	1.9881	68.407022
498	Male	150	95	5	1.50	2.2500	42.222222
499	Male	173	131	5	1.73	2.9929	43.770256

500 rows × 7 columns

In [9]:
```
df.head()
```

Out[9]:

	Gender	Height in cm	Weight	Index	Height in m	Height Square	BMI
0	Male	174	96	4	1.74	3.0276	31.708284
1	Male	189	87	2	1.89	3.5721	24.355421
2	Female	185	110	4	1.85	3.4225	32.140248
3	Female	195	104	3	1.95	3.8025	27.350427
4	Male	149	61	3	1.49	2.2201	27.476240

In [10]:
```
df.tail()
```

Out[10]:

	Gender	Height in cm	Weight	Index	Height in m	Height Square	BMI
495	Female	150	153	5	1.50	2.2500	68.000000
496	Female	184	121	4	1.84	3.3856	35.739603
497	Female	141	136	5	1.41	1.9881	68.407022
498	Male	150	95	5	1.50	2.2500	42.222222
499	Male	173	131	5	1.73	2.9929	43.770256

In [11]:
```
df.describe()
```

Out[11]:

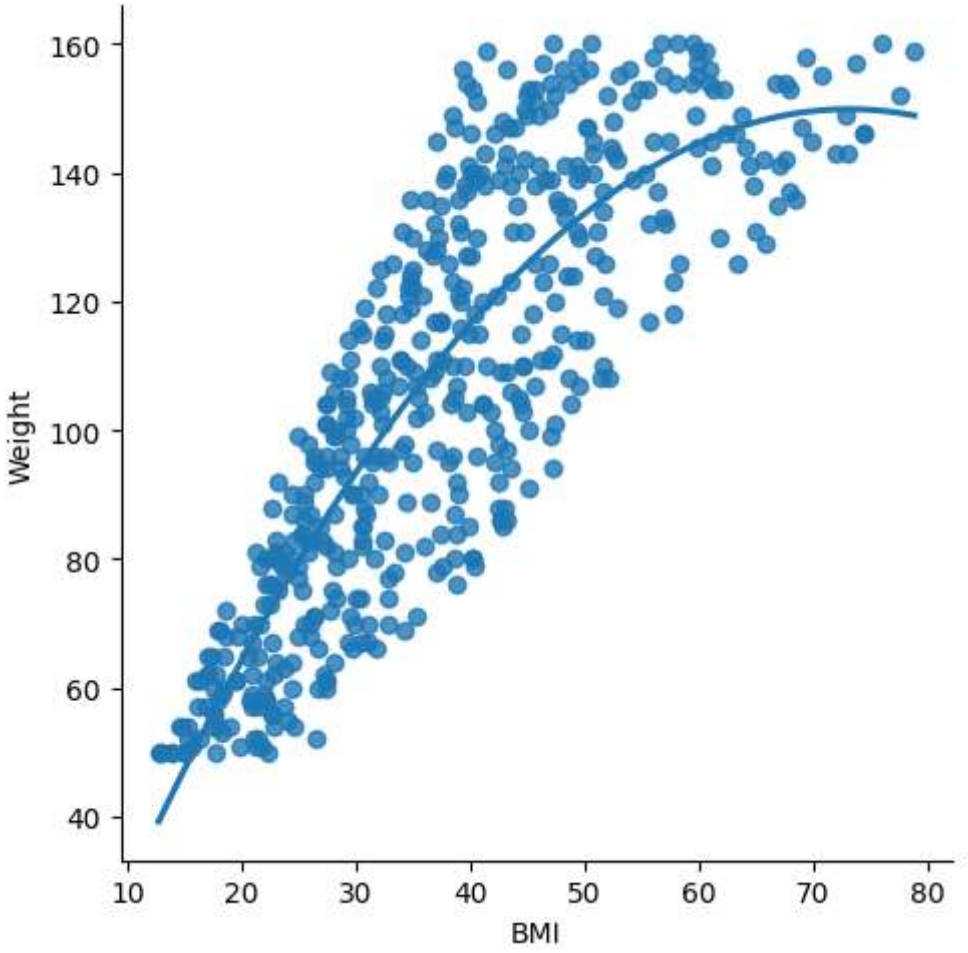
	Height in cm	Weight	Index	Height in m	Height Square	BMI
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	169.944000	106.000000	3.748000	1.699440	2.914858	37.765284
std	16.375261	32.382607	1.355053	0.163753	0.553880	13.965620
min	140.000000	50.000000	0.000000	1.400000	1.960000	12.753801
25%	156.000000	80.000000	3.000000	1.560000	2.433600	27.169353
50%	170.500000	106.000000	4.000000	1.705000	2.907050	36.956941
75%	184.000000	136.000000	5.000000	1.840000	3.385600	46.390467
max	199.000000	160.000000	5.000000	1.990000	3.960100	78.853402

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Gender         500 non-null    object 
 1   Height in cm   500 non-null    int64  
 2   Weight         500 non-null    int64  
 3   Index          500 non-null    int64  
 4   Height in m    500 non-null    float64
 5   Height Square  500 non-null    float64
 6   BMI            500 non-null    float64
dtypes: float64(3), int64(3), object(1)
memory usage: 27.5+ KB
```

In [19]:
```
sns.lmplot(x="BMI",y="Weight",data=df,order=2,ci=None)
```

Out[19]:
```
<seaborn.axisgrid.FacetGrid at 0x17a837f3f90>
```



In [34]:
```
df.fillna(method='ffill')
```

Out[34]:

	Gender	Height in cm	Weight	Index	Height in m	Height Square	BMI
0	Male	174	96	4	1.74	3.0276	31.708284
1	Male	189	87	2	1.89	3.5721	24.355421
2	Female	185	110	4	1.85	3.4225	32.140248
3	Female	195	104	3	1.95	3.8025	27.350427
4	Male	149	61	3	1.49	2.2201	27.476240
...	...	...	...	...	...	...	...
495	Female	150	153	5	1.50	2.2500	68.000000
496	Female	184	121	4	1.84	3.3856	35.739603
497	Female	141	136	5	1.41	1.9881	68.407022
498	Male	150	95	5	1.50	2.2500	42.222222
499	Male	173	131	5	1.73	2.9929	43.770256

500 rows × 7 columns

In [22]:
```
regr=LinearRegression()
```

In [24]:
```
df.columns
```

Out[24]:
```
Index(['Gender', 'Height in cm', 'Weight', 'Index', 'Height in m',
       'Height Square', 'BMI'],
      dtype='object')
```

In [26]:
```
X=np.array(df['BMI']).reshape(-1,1)
y=np.array(df['Weight']).reshape(-1,1)
```

In [31]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
regr.fit(X_train,y_train)
regr.fit(X_train,y_train)
```

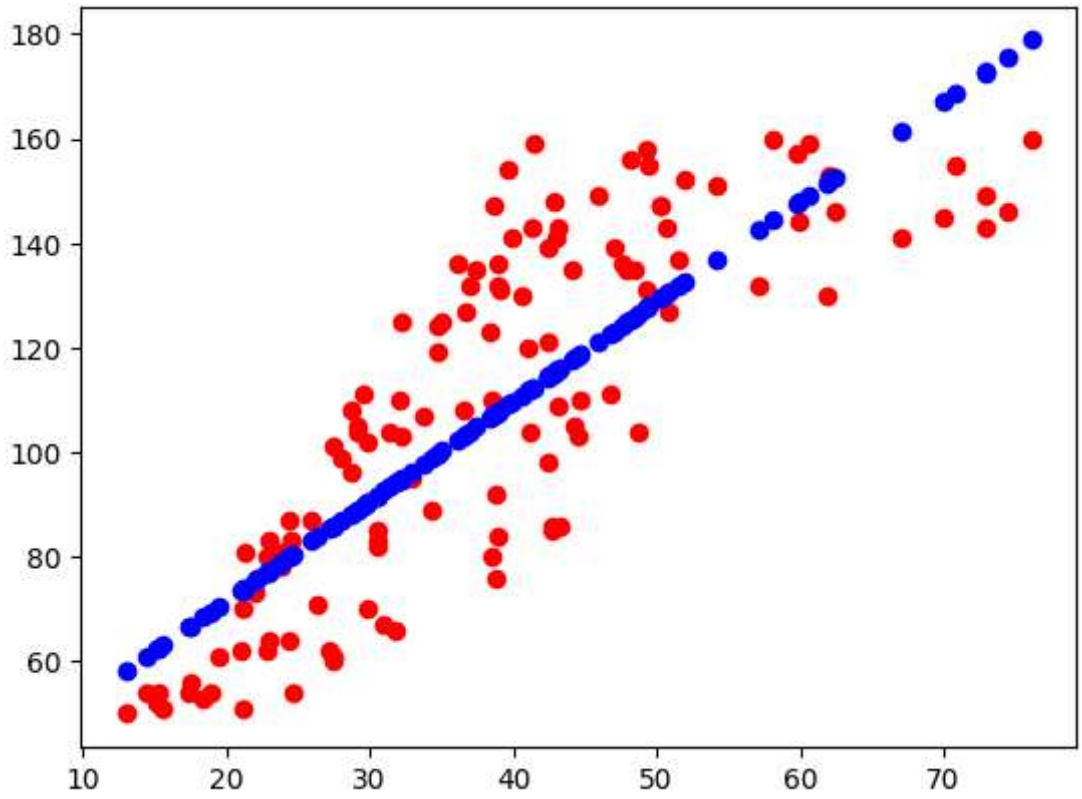
Out[31]:
```
▾ LinearRegression

LinearRegression()
```

In [32]:
```
print(regr.score(X_test,y_test))
```

```
0.6684491363261074
```

In [33]:
```
y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='r')
plt.scatter(X_test,y_pred,color='b')
plt.show()
```



In [ ]: