In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

In [2]:
```
#downloading and importing file as csv
df = pd.read_csv('train.csv')
df
```

Out[2]:

	# age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	-0.049105	-0.044642	0.160855	-0.046985	-0.029088	-0.019790	-0.047082	0.034309	0.028017	0.011349	346
1	-0.070900	0.050680	-0.089197	-0.074528	-0.042848	-0.025739	-0.032356	-0.002592	-0.012908	-0.054925	104
2	0.001751	-0.044642	-0.070875	-0.022885	-0.001569	-0.001001	0.026550	-0.039493	-0.022512	0.007207	49
3	0.048974	0.050680	0.081097	0.021872	0.043837	0.064134	-0.054446	0.071210	0.032433	0.048628	180
4	0.041708	0.050680	0.061696	-0.040099	-0.013953	0.006202	-0.028674	-0.002592	-0.014956	0.011349	110
...	...	...	...	...	...	...	...	...	...	...	...
326	0.012648	-0.044642	-0.020218	-0.015999	0.012191	0.021233	-0.076536	0.108111	0.059881	-0.021788	233
327	0.019913	-0.044642	-0.057941	-0.057314	-0.001569	-0.012587	0.074412	-0.039493	-0.061177	-0.075636	63
328	0.023546	-0.044642	0.070319	0.025315	-0.034592	-0.014466	-0.032356	-0.002592	-0.019197	-0.009362	288
329	0.016281	0.050680	0.072474	0.076958	-0.008449	0.005575	-0.006584	-0.002592	-0.023645	0.061054	131
330	-0.009147	0.050680	-0.030996	-0.026328	-0.011201	-0.001001	-0.021311	-0.002592	0.006209	0.027917	42

331 rows × 11 columns

In [3]:
```
df.shape
```

Out[3]: (331, 11)

In [4]:
```
df.describe()
```

Out[4]:

	# age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
count	331.000000	331.000000	331.000000	331.000000	331.000000	331.000000	331.000000	331.000000	331.000000	331.000000	331.000000
mean	-0.000214	-0.000293	0.000053	-0.001045	-0.001423	-0.001220	0.000440	-0.001473	-0.001594	0.000099	149.722054
std	0.047207	0.047618	0.048627	0.047621	0.047371	0.046936	0.047303	0.046591	0.048088	0.047263	76.796907
min	-0.107226	-0.044642	-0.090275	-0.102071	-0.126781	-0.115613	-0.102307	-0.076395	-0.126097	-0.137767	25.000000
25%	-0.034575	-0.044642	-0.035307	-0.036656	-0.035968	-0.029967	-0.032356	-0.039493	-0.035817	-0.034215	84.000000
50%	0.005383	-0.044642	-0.008362	-0.005671	-0.004321	-0.004445	-0.000584	-0.002592	-0.005145	-0.001078	138.000000
75%	0.038076	0.050680	0.030979	0.032775	0.027326	0.028905	0.026550	0.034309	0.032124	0.027917	202.000000
max	0.110727	0.050680	0.170555	0.132044	0.153914	0.198788	0.181179	0.185234	0.133599	0.135612	346.000000

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 331 entries, 0 to 330
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   # age   331 non-null    float64
 1   sex     331 non-null    float64
 2   bmi     331 non-null    float64
 3   bp      331 non-null    float64
 4   s1      331 non-null    float64
 5   s2      331 non-null    float64
 6   s3      331 non-null    float64
 7   s4      331 non-null    float64
 8   s5      331 non-null    float64
 9   s6      331 non-null    float64
 10  target  331 non-null    int64  
dtypes: float64(10), int64(1)
memory usage: 28.6 KB
```

Viewing the Data Shapes

In [6]:
```
df.head()
```

Out[6]:

	# age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	-0.049105	-0.044642	0.160855	-0.046985	-0.029088	-0.019790	-0.047082	0.034309	0.028017	0.011349	346
1	-0.070900	0.050680	-0.089197	-0.074528	-0.042848	-0.025739	-0.032356	-0.002592	-0.012908	-0.054925	104
2	0.001751	-0.044642	-0.070875	-0.022885	-0.001569	-0.001001	0.026550	-0.039493	-0.022512	0.007207	49
3	0.048974	0.050680	0.081097	0.021872	0.043837	0.064134	-0.054446	0.071210	0.032433	0.048628	180
4	0.041708	0.050680	0.061696	-0.040099	-0.013953	0.006202	-0.028674	-0.002592	-0.014956	0.011349	110

In [7]:
```
#create x columns by dropping the 'target' column
x = df.drop('target',axis=1)
#creating thr y column by calling out the 'target' column
y = df['target']
```

In [8]:
```
x
```

Out[8]:

	# age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
0	-0.049105	-0.044642	0.160855	-0.046985	-0.029088	-0.019790	-0.047082	0.034309	0.028017	0.011349
1	-0.070900	0.050680	-0.089197	-0.074528	-0.042848	-0.025739	-0.032356	-0.002592	-0.012908	-0.054925
2	0.001751	-0.044642	-0.070875	-0.022885	-0.001569	-0.001001	0.026550	-0.039493	-0.022512	0.007207
3	0.048974	0.050680	0.081097	0.021872	0.043837	0.064134	-0.054446	0.071210	0.032433	0.048628
4	0.041708	0.050680	0.061696	-0.040099	-0.013953	0.006202	-0.028674	-0.002592	-0.014956	0.011349
...	...	...	...	...	...	...	...	...	...	...
326	0.012648	-0.044642	-0.020218	-0.015999	0.012191	0.021233	-0.076536	0.108111	0.059881	-0.021788
327	0.019913	-0.044642	-0.057941	-0.057314	-0.001569	-0.012587	0.074412	-0.039493	-0.061177	-0.075636
328	0.023546	-0.044642	0.070319	0.025315	-0.034592	-0.014466	-0.032356	-0.002592	-0.019197	-0.009362
329	0.016281	0.050680	0.072474	0.076958	-0.008449	0.005575	-0.006584	-0.002592	-0.023645	0.061054
330	-0.009147	0.050680	-0.030996	-0.026328	-0.011201	-0.001001	-0.021311	-0.002592	0.006209	0.027917

331 rows × 10 columns

In [9]:
```
y
```

Out[9]:
```
0     346
1     104
2      49
3     180
4     110
     ... 
326   233
327    63
328   288
329   131
330    42
Name: target, Length: 331, dtype: int64
```

In [10]:
```
x.shape
```

Out[10]: (331, 10)

In [11]:
```
y.shape
```

Out[11]: (331,)

In [12]:
```
#splitting cells into train and test set
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

In [13]:
```
x_train
```

Out[13]:

	# age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
251	-0.020045	-0.044642	0.004572	0.097616	0.005311	-0.020729	0.063367	-0.039493	0.012553	0.011349
38	-0.005515	-0.044642	0.008883	-0.050428	0.025950	0.047224	-0.043401	0.071210	0.014823	0.003064
166	0.041708	-0.044642	-0.008362	-0.057314	0.000063	-0.013176	0.151726	-0.076395	-0.060237	-0.017646
258	0.038931	-0.044642	0.006728	0.025315	0.092078	0.008707	0.063367	-0.039493	0.009436	0.032059
259	-0.009147	0.050680	0.001339	-0.002228	0.079612	0.070084	0.033914	-0.002592	0.026714	0.081764
...	...	...	...	...	...	...	...	...	...	...
299	-0.056370	0.050680	-0.060097	-0.036656	-0.088254	-0.070833	-0.013948	-0.039493	-0.078141	-0.104630
247	0.030811	0.050680	0.032595	0.049415	-0.040096	-0.043589	-0.069172	0.034309	0.063017	0.003064
294	0.038076	0.050680	-0.013751	-0.015999	-0.035968	-0.021982	-0.013948	-0.002592	-0.025952	-0.001078
302	-0.052738	-0.044642	0.030440	-0.074528	-0.023584	-0.011335	-0.002903	-0.002592	-0.030751	-0.001078
199	-0.045472	-0.044642	-0.073030	-0.081414	0.083740	0.027809	0.173816	-0.039493	-0.004220	0.003064

264 rows × 10 columns

In [14]:
```
x_test
```

Out[14]:

	# age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
171	-0.070900	-0.044642	0.092953	0.012691	0.020446	0.042527	0.000779	0.000360	-0.054544	-0.001078
26	-0.027310	0.050680	0.060618	0.049415	0.085116	0.086368	-0.002903	0.034309	0.037814	0.048628
213	0.023546	0.050680	0.061696	0.062039	0.024574	-0.036073	-0.091262	0.155345	0.133396	0.081764
89	-0.089063	-0.044642	-0.041774	-0.019442	-0.066239	-0.074277	0.008142	-0.039493	0.001144	-0.030072
295	-0.041840	-0.044642	0.047685	0.059744	0.127771	0.128016	-0.024993	0.108111	0.063893	0.040343
...	...	...	...	...	...	...	...	...	...	...
119	0.023546	-0.044642	0.019662	-0.012556	0.083740	0.038769	0.063367	-0.002592	0.066048	0.048628
103	0.110727	0.050680	0.006728	0.028758	-0.027712	-0.007264	-0.047082	0.034309	0.002008	0.077622
207	0.016281	0.050680	-0.045007	0.063187	0.010815	-0.000374	0.063367	-0.039493	-0.030751	0.036201
78	-0.001882	-0.044642	0.054152	0.066495	0.072732	0.056619	-0.043401	0.084863	0.084495	0.048628
230	0.096197	-0.044642	0.040140	-0.057314	0.045213	0.060690	-0.021311	0.036154	0.012553	0.023775

67 rows × 10 columns

In [15]:
```
x_train.shape
```

Out[15]: (264, 10)

In [16]:
```
x_test.shape
```

Out[16]: (67, 10)

In [17]:
```
y_train
```

Out[17]:
```
251    48
38    174
166    39
258   109
259   142
     ... 
299    70
247   208
294    83
302   172
199    57
Name: target, Length: 264, dtype: int64
```

In [18]:
```
y_train.shape
```

Out[18]: (264,)

In [19]:
```
y_test
```

Out[19]:
```
171    200
26     186
213    242
89     185
295    258
      ... 
119    262
103    277
207    102
78     192
230    180
Name: target, Length: 67, dtype: int64
```

In [20]:
```
y_test.shape
```

Out[20]: (67,)

In [21]:
```
#fitting a linear regression model to the training set
model = LinearRegression()
model.fit(x_train,y_train)
```

Out[21]: LinearRegression()

In [22]:
```
model.coef_
```

Out[22]:
```
array([  33.34605548, -204.10407179,  602.18660893,  217.98837792,
        -720.64941739,  387.48678118,   63.29757377,   75.96080854,
         747.31627008,   32.22288286])
```

In [23]:
```
model.intercept_
```

Out[23]: 148.48368177806714

In [26]:
```
#predicting the dependent variable using the testing set
y_pred = model.predict(x_test)
y_pred
```

Out[26]:
```
array([174.99663409, 188.88071595, 266.26089248, 141.58613593,
       211.14400835, 198.94776525, 157.28124506, 249.50517695,
       127.57539957,  72.8176131 , 184.00617483, 116.92266602,
       208.37752375, 103.20636735,  65.12223698, 190.96734593,
       164.48744776,  70.46735987, 156.29777123, 182.42015386,
        60.68222761, 125.72284698, 239.58365918, 116.97121792,
        97.43421659, 208.9593951 , 142.42091563,  81.24969286,
       161.86283334,  92.55229113, 100.28704753, 153.63103166,
       116.52574533, 117.55193336,  97.9803489 , 211.37134109,
       189.21238427, 174.42247787, 258.03773578, 235.71033832,
       167.57642283, 189.69301945, 256.91926056, 133.71636105,
       116.87266221, 123.12246222, 160.14059867, 147.76388835,
       210.52614898, 152.43968901, 186.29567365,  86.51636506,
       128.85781915,  50.55324256,  51.19944654, 119.29924424,
       120.95585655, 111.22316223, 105.32369452,  54.00630204,
       125.10832994, 108.24019044, 176.89827672, 172.9360025 ,
        96.61182298, 213.58179037, 174.95899793])
```

In [28]:
```
#Calculating the mean squared error to measure the model's performance
mse = mean_squared_error(y_test,y_pred)
mse
```

Out[28]: 3233.560103535633

In [29]:
```
print("Mean Squared Error:",mse)
```

```
Mean Squared Error: 3233.560103535633
```

In [ ]:

In [ ]: