Lab 1 - Linear Regression

Import necessary libraries

In [1]:

1. Loading dataset from csv file

In [2]:

1.1 Inspecting the dataset

In [3]:

Out[3]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59.0	F	32.1	NaN	157.0	93.2	38.0	4.0	4.8598	87.0	151
1	48.0	M	21.6	87.0	183.0	103.2	70.0	NaN	3.8918	69.0	75
2	NaN	NaN	30.5	93.0	156.0	93.6	41.0	NaN	4.6728	85.0	141
3	24.0	M	NaN	84.0	198.0	131.4	40.0	5.0	4.8903	89.0	206
4	NaN	M	23.0	101.0	192.0	125.4	52.0	4.0	4.2905	80.0	135

In [4]:

```
Summary statistics:
```

Out[4]:

	AGE	BMI	BP	S1	S2	S3	S4	S5	S6	Y
count	409.000000	405.000000	407.000000	408.000000	406.000000	408.000000	415.000000	411.000000	416.000000	454.000000
mean	48.322738	26.446173	94.266929	188.909314	115.142365	49.756127	4.038000	4.632813	91.257212	153.988987
std	13.149722	4.416022	13.855572	34.450393	30.170183	13.145004	1.267534	0.520792	11.612361	78.006636
min	19.000000	18.100000	62.000000	97.000000	43.400000	22.000000	2.000000	3.258100	58.000000	25.000000
25%	38.000000	23.300000	84.000000	164.750000	95.100000	40.000000	3.000000	4.269700	83.000000	88.000000
50%	50.000000	25.700000	92.330000	186.000000	112.900000	48.000000	4.000000	4.605200	91.000000	142.000000
75%	59.000000	29.300000	104.500000	209.000000	135.200000	58.000000	5.000000	4.997200	98.000000	214.000000
max	79.000000	42.200000	133.000000	301.000000	242.400000	99.000000	9.090000	6.107000	124.000000	346.000000

In [5]:

```
Data dimensions: (454, 11)

Column names: Index(['AGE', 'SEX', 'BMI', 'BP', 'S1', 'S2', 'S3', 'S4', 'S5', 'S6', 'Y'], dtype='object')

Data types:
 AGE    float64
SEX     object
BMI    float64
BP     float64
S1     float64
S2     float64
S3     float64
S4     float64
S5     float64
S6     float64
Y        int64
dtype: object
```

2. Data preprocessing

2.1. Dealing with Missing Values

Missing Values indicator

In [6]:

Out[6]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	False	False	False	True	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	True	False	False	False
2	True	True	False	False	False	False	False	True	False	False	False
3	False	False	True	False	False	False	False	False	False	False	False
4	True	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
449	False	False	False	False	False	False	False	False	False	False	False
450	False	False	False	False	False	False	True	True	False	False	False
451	False	False	False	True	True	False	False	False	False	False	False
452	False	False	True	False	True	False	True	False	False	False	False
453	False	False	False	False	False	False	False	False	False	False	False

454 rows × 11 columns

Number of missing values for each feature

In [7]:

Out[7]:
```
AGE    45
SEX    43
BMI    49
BP     47
S1     46
S2     48
S3     46
S4     39
S5     43
S6     38
Y       0
dtype: int64
```

Mean Imputation for AGE

In [8]:

```
The mean age is: 48.32273838630807
```

In [9]:
```
# Show the imputation result for the Missing Values in AGE
```

Out[9]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59.000000	F	32.1	NaN	157.0	93.2	38.0	4.00	4.8598	87.0	151
1	48.000000	M	21.6	87.0	183.0	103.2	70.0	NaN	3.8918	69.0	75
2	48.322738	NaN	30.5	93.0	156.0	93.6	41.0	NaN	4.6728	85.0	141
3	24.000000	M	NaN	84.0	198.0	131.4	40.0	5.00	4.8903	89.0	206
4	48.322738	M	23.0	101.0	192.0	125.4	52.0	4.00	4.2905	80.0	135
5	23.000000	M	22.6	89.0	139.0	64.8	61.0	2.00	4.1897	68.0	97
6	36.000000	F	22.0	NaN	160.0	99.6	50.0	3.00	3.9512	82.0	138
7	66.000000	NaN	26.2	114.0	255.0	185.0	56.0	4.55	4.2485	92.0	63
8	48.322738	F	32.1	83.0	NaN	119.4	42.0	NaN	4.4773	NaN	110
9	29.000000	M	NaN	85.0	180.0	93.4	43.0	4.00	5.3845	88.0	310

In [10]:
```
# Show the imputation result for the Missing Values in AGE
```

Out[10]:
```
AGE     0
SEX    43
BMI    49
BP     47
S1     46
S2     48
S3     46
S4     39
S5     43
S6     38
Y       0
dtype: int64
```

Mode Imputation for SEX

In [11]:
```

```

```
Mode sex : M
```

In [12]:
```
# Show the imputation result for the Missing Values in SEX
```

Out[12]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59.000000	F	32.1	NaN	157.0	93.2	38.0	4.00	4.8598	87.0	151
1	48.000000	M	21.6	87.0	183.0	103.2	70.0	NaN	3.8918	69.0	75
2	48.322738	M	30.5	93.0	156.0	93.6	41.0	NaN	4.6728	85.0	141
3	24.000000	M	NaN	84.0	198.0	131.4	40.0	5.00	4.8903	89.0	206
4	48.322738	M	23.0	101.0	192.0	125.4	52.0	4.00	4.2905	80.0	135
5	23.000000	M	22.6	89.0	139.0	64.8	61.0	2.00	4.1897	68.0	97
6	36.000000	F	22.0	NaN	160.0	99.6	50.0	3.00	3.9512	82.0	138
7	66.000000	M	26.2	114.0	255.0	185.0	56.0	4.55	4.2485	92.0	63
8	48.322738	F	32.1	83.0	NaN	119.4	42.0	NaN	4.4773	NaN	110
9	29.000000	M	NaN	85.0	180.0	93.4	43.0	4.00	5.3845	88.0	310

Dealing with Missing Values in the othe Columns

In [13]:
```

```

In [14]:
```
# Show the result of the Missing Values imputation
```

Out[14]:
```
AGE    0
SEX    0
BMI    0
BP     0
S1     0
S2     0
S3     0
S4     0
S5     0
S6     0
Y      0
dtype: int64
```

In [15]:
```
# Show the result of the Missing Values imputation
```

Out[15]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59.000000	F	32.100000	94.266929	157.000000	93.2	38.0	4.000	4.8598	87.000000	151
1	48.000000	M	21.600000	87.000000	183.000000	103.2	70.0	4.038	3.8918	69.000000	75
2	48.322738	M	30.500000	93.000000	156.000000	93.6	41.0	4.038	4.6728	85.000000	141
3	24.000000	M	26.446173	84.000000	198.000000	131.4	40.0	5.000	4.8903	89.000000	206
4	48.322738	M	23.000000	101.000000	192.000000	125.4	52.0	4.000	4.2905	80.000000	135
5	23.000000	M	22.600000	89.000000	139.000000	64.8	61.0	2.000	4.1897	68.000000	97
6	36.000000	F	22.000000	94.266929	160.000000	99.6	50.0	3.000	3.9512	82.000000	138
7	66.000000	M	26.200000	114.000000	255.000000	185.0	56.0	4.550	4.2485	92.000000	63
8	48.322738	F	32.100000	83.000000	188.909314	119.4	42.0	4.038	4.4773	91.257212	110
9	29.000000	M	26.446173	85.000000	180.000000	93.4	43.0	4.000	5.3845	88.000000	310

2.2. Data Encoding

Encoding SEX column

In [16]:

In [17]:
```
# Show the result of SEX Encoding
```

Out[17]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59.000000	0	32.100000	94.266929	157.0	93.2	38.0	4.000	4.8598	87.0	151
1	48.000000	1	21.600000	87.000000	183.0	103.2	70.0	4.038	3.8918	69.0	75
2	48.322738	1	30.500000	93.000000	156.0	93.6	41.0	4.038	4.6728	85.0	141
3	24.000000	1	26.446173	84.000000	198.0	131.4	40.0	5.000	4.8903	89.0	206
4	48.322738	1	23.000000	101.000000	192.0	125.4	52.0	4.000	4.2905	80.0	135

2.3. Normalization

In [18]:

In [92]:
```
# Show the result of the Normaliation
```

Out[92]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	0.666667	0.0	0.580913	0.454464	0.294118	0.250251	0.207792	0.282087	0.562217	0.439394	151
1	0.483333	1.0	0.145228	0.352113	0.421569	0.300503	0.623377	0.287447	0.222437	0.166667	75
2	0.488712	1.0	0.514523	0.436620	0.289216	0.252261	0.246753	0.287447	0.496578	0.409091	141
3	0.083333	1.0	0.346314	0.309859	0.495098	0.442211	0.233766	0.423131	0.572923	0.469697	206
4	0.488712	1.0	0.203320	0.549296	0.465686	0.412060	0.389610	0.282087	0.362385	0.333333	135

2.4. Preparing the Data for Training

Split the Features X from the Target Y

In [114]:

Split the Features X and the Target y into Training/Testing sets

In [115]:
```
# Use only one feature
```

In [116]:

3. Creating and Training the Linear Regression Model

In [117]:

In [118]:
```
# The mean squared error
```

```
Mean squared error: 4078.94
```

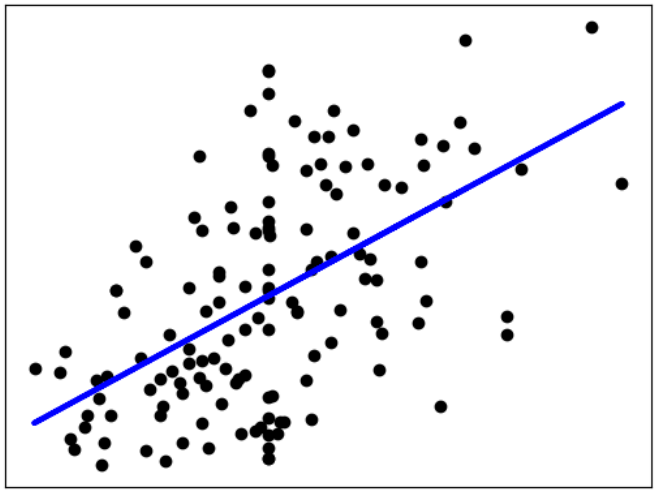
In [119]:
```
# The coefficients
```

```
Coefficients:
 [246.74363727]
```

In [120]:
```
# The coefficient of determination: 1 is perfect prediction
```

```
Coefficient of determination: 0.27
```

In [122]:
```
# Plot outputs
```



In [ ]:
```
```