Implementation of Linear Regression using scikit-learn

P. Usha sree

RA1911042020051

- AIM: To program the implementation of Implementation of Linear Regression using scikit-learn

1) Importing the libraries

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

2)Importing the dataset

In [4]:
```
df =pd.read_csv(r"C:\data sets\stroke.csv", encoding='ANSI')
df
```

Out[4]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

5110 rows × 12 columns

3)Handle the missing data

In [6]:
```
from sklearn.impute import SimpleImputer
```

In [7]:
```
df.describe()
```

Out[7]:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	36517.829354	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	21161.721625	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	67.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	17741.250000	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	36932.000000	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	54682.000000	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

In [8]:
```
df["bmi"]=df["bmi"].replace(np.NaN,df["bmi"].mean())
print(df["bmi"])
```

```
0       36.600000
1       28.893237
2       32.500000
3       34.400000
4       24.000000
          ...
5105    28.893237
5106    40.000000
5107    30.600000
5108    25.600000
5109    26.200000
Name: bmi, Length: 5110, dtype: float64
```

4)Handling Categorical data

In [9]:
```
cat_cols =['gender','ever_married','work_type','Residence_type','smoking_status']
from sklearn.preprocessing import LabelEncoder
for each_col in cat_cols:
    Le=LabelEncoder()
    df[each_col]=Le.fit_transform (df[each_col])
```

In [10]:
```
df
```

Out[10]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	1	67.0	0	1	1	2	1	228.69	36.600000	1	1
1	51676	0	61.0	0	0	1	3	0	202.21	28.893237	2	1
2	31112	1	80.0	0	1	1	2	0	105.92	32.500000	2	1
3	60182	0	49.0	0	0	1	2	1	171.23	34.400000	3	1
4	1665	0	79.0	1	0	1	3	0	174.12	24.000000	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	0	80.0	1	0	1	2	1	83.75	28.893237	2	0
5106	44873	0	81.0	0	0	1	3	1	125.20	40.000000	2	0
5107	19723	0	35.0	0	0	1	3	0	82.99	30.600000	2	0
5108	37544	1	51.0	0	0	1	2	0	166.29	25.600000	1	0
5109	44679	0	44.0	0	0	1	0	1	85.28	26.200000	0	0

5110 rows × 12 columns

5)Splitting the data set into training and testing

In [11]:
```
x=df[['id','gender','age','hypertension','heart_disease','ever_married','work_type','Residence_type','avg_glucose_level','bmi',
      'smoking_status']].values
y=df[['stroke']].values
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size =0.8)
```

In [12]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[12]:
```
((4088, 11), (1022, 11), (4088, 1), (1022, 1))
```

2. Now we have to implement Linear regression model and predict the values based on our data set.

6) Linear Regression using Scikit-learn

In [13]:
```
from sklearn.linear_model import LinearRegression
```

In [24]:
```
lr =LinearRegression()
lr.fit(x,y)
print("Our model has been Trained!!")
```

```
Our model has been Trained!!
```

In [28]:
```
y_pred =lr.predict(x_test)
y_pred
```

Out[28]:
```
array([[ 0.12658376],
       [-0.0079362 ],
       [-0.02809805],
       ...,
       [ 0.22041728],
       [ 0.01090751],
       [-0.02017515]])
```

In [30]:
```
from sklearn.metrics import mean_squared_error, mean_absolute_error
mean_squared_error(y_test,y_pred)
```

Out[30]:
```
0.03776576024885075
```

In [31]:
```
mean_absolute_error(y_test,y_pred)
```

Out[31]:
```
0.08805156187116507
```

In [ ]:

❖ RESULT: Implementation of Linear Regression using scikit-learn has been implemented on a data set.