

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
class LR:
    def __init__(self):
        self.m=0 ## slope
        self.b=0 ## intercept
    def fit(self,x_train, y_train):
        num=0
        div=0

        for i in range(x_train.shape[0]):
            num += (x_train.iloc[i] - x_train.mean()) * ( y_train.iloc[i]- y_train.mean())
            div += (x_train.iloc[i] - x_train.mean()) **2
        self.m = num/div
        self.b= y_train.mean()- (self.m * x_train.mean())

        return self.m , self.b

    def predict(self, x):
        # print(self.m, self.b)
        return ((self.m * x) + self.b)

    def intercept(self):
        #print(self.b)
        return self.b
    def coef_(self):
        # print(self.m)
        return self.m
```

```
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Load the diabetes dataset
x, y = datasets.load_diabetes(return_X_y=True)
```

```
df= pd.read_csv('ins.csv')
```

```
df.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

```
from sklearn.model_selection import train_test_split
```

```
y= df['charges']
x= df['bmi']
x_train, x_test, y_train, y_test = train_test_split(x,y,random_state= 2)
```

```
model= LR()
```

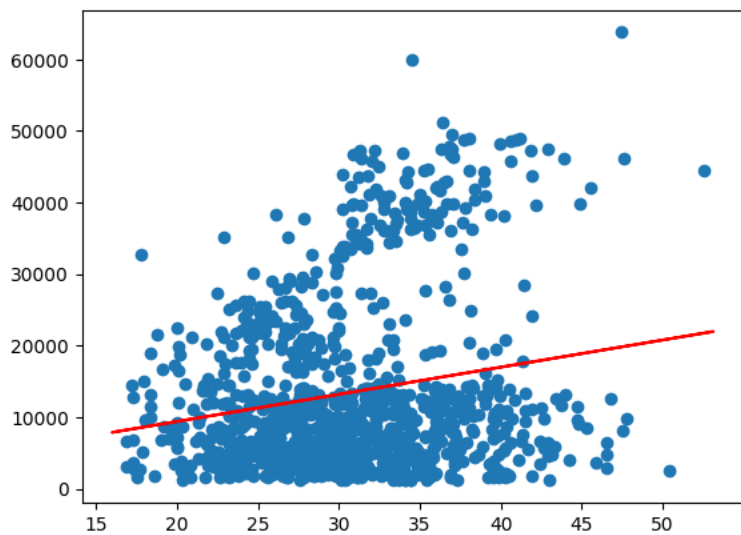
```
model.fit(x_train,y_train)
```

```
(379.36797676692873, 1811.7996948058662)
```

```
model.predict(x_test[2])
```

```
14330.942928114515
```

```
plt.scatter(x_train, y_train)
plt.plot(x_test, model.predict(x_test) , color= 'red')
plt.show()
```



```
intercept=model.intercept()  
coeff= model.coef_  
print("Intercept: ",intercept, "coeff: ", coeff)
```

```
Intercept: 1811.7996948058662 coeff: 379.36797676692873
```

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
```

```
x_train = x_train.values.reshape(-1, 1)  
y_train = y_train.values.reshape(-1, 1)
```

```
lr.fit(x_train, y_train)
```

```
▼ LinearRegression  
LinearRegression()
```

```
lr.intercept_
```

```
array([1811.79969481])
```

```
lr.coef_
```

```
array([[379.36797677]])
```