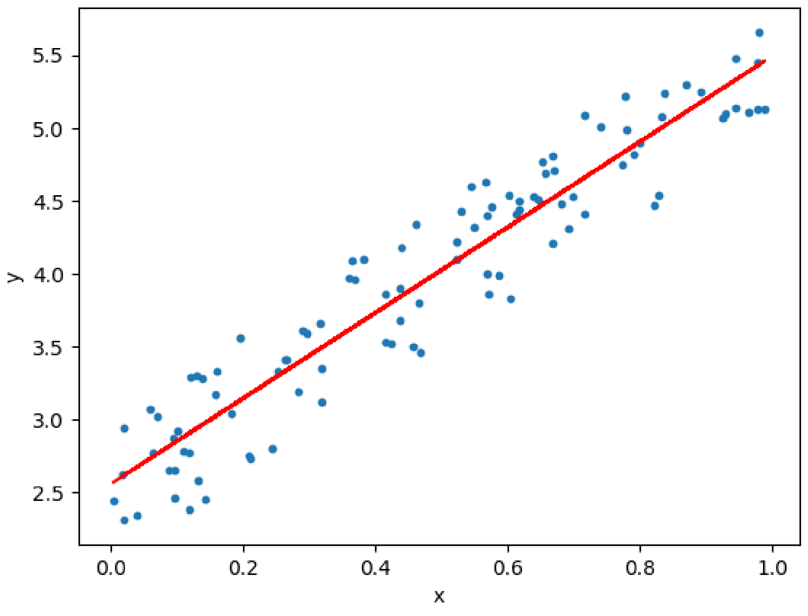
T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

Ex 2: Simple Linear Regression



Working Principle:

The basic concept behind linear regression is to find the best fit line that passes through the data plot.

In order to achieve this best fit line, we have to start with a simple line (either stochastic or fixed) and work our way up to find the perfect fit. This can be done by simply calculating the distance between the line and every single point in the dataset.

Which introduces us to the Loss Function.

Loss Function:

The loss function used in the linear-regression is 'Mean Squared Error (MSE)' where the difference between the line and the data points are squared and averaged. This loss function can be plotted and minimized to find the best fit line.

Mathematically, the MSE can be represented by the formula:

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)^2$$

Minimizing the Loss Function:

The loss function is minimized using the "Gradient Descent Algorithm". In short, Gradient Descent Algorithm can find the local minima of a function by comparing the previous values proceeding with a fixed step size. A very popular example would be trying to navigate to the bottom of the hill where we tend to proceed until we cannot feel the slope of the hill declining – indicating that we have reached the ground.

Gradient Descent Algorithm:

The equation of a line is mathematically represented as:

$$Y = mX + c$$

The predicted value is given by:

$$\bar{y}_i = mx_i + c$$

Hence, the MSE function is given by:

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)^2 \quad \rightarrow \quad E = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

In order to find the find the next step(Let us assume m=0 and c=0 and the step size be L=0.0001), we need to find the partial derivative after differentiating once with 'm' and once with 'c'. Which will leave us with two values 'Dm' and 'Dc'.

$$D_m = \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - \bar{y}_i) \qquad D_c = \frac{-2}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)$$

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

Now we can update the values of 'm' and 'c' to find the next best fit line.

$$m = m - L \times D_m$$

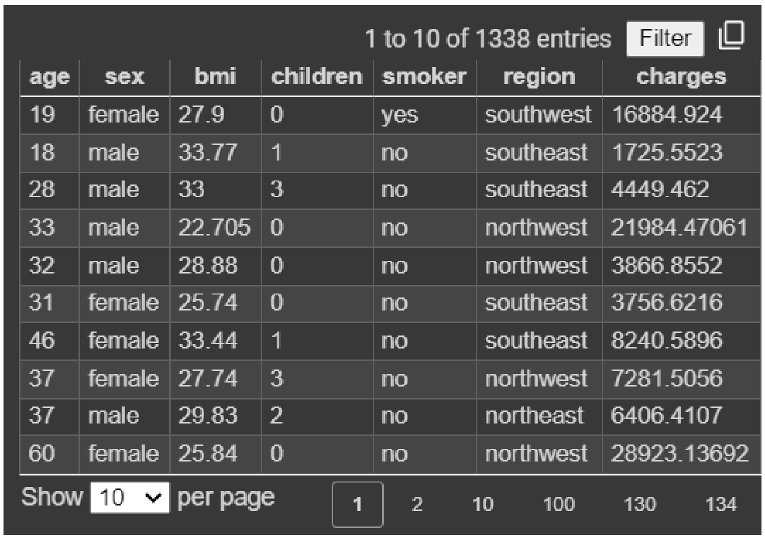
$$c = c - L \times D_c$$

Therefore, by repeating this process continuously, we can achieve the best fit line.

Code:

Let us find the relationship between the BMI(Body Mass Index) of a person and the average medical insurance claimed by the person. To analyze this relationship using Linear Regression with Gradient Descent Algorithm, let us consider the following dataset.

Dataset:



1 to 10 of 1338 entries Filter

age	sex	bmi	children	smoker	region	charges
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.47061
32	male	28.88	0	no	northwest	3866.8552
31	female	25.74	0	no	southeast	3756.6216
46	female	33.44	1	no	southeast	8240.5896
37	female	27.74	3	no	northwest	7281.5056
37	male	29.83	2	no	northeast	6406.4107
60	female	25.84	0	no	northwest	28923.13692

Show 10 per page 1 2 10 100 130 134

Dataset Description: Health Insurance Costs Dataset

This dataset contains information related to health insurance costs for individuals. It includes the following features:

1. Age: The age of the primary beneficiary for the health insurance.

2. Sex: The gender of the insurance contractor, categorized as female or male.

3. BMI (Body Mass Index): A measure that provides an understanding of body weight relative to height. It is calculated as weight in kilograms divided by the square of height in meters. A BMI ideally falls between 18.5 and 24.9, indicating normal weight.

4. Children: The number of children covered by the health insurance or the number of dependents.

5. Smoker: A binary variable indicating whether the individual is a smoker or non-smoker.

6. Region: The residential area of the beneficiary within the United States, categorized into northeast, southeast, southwest, and northwest.

7. Charges: The individual medical costs billed by the health insurance company.

Code:

#Importing Libraries and configuring plotscale

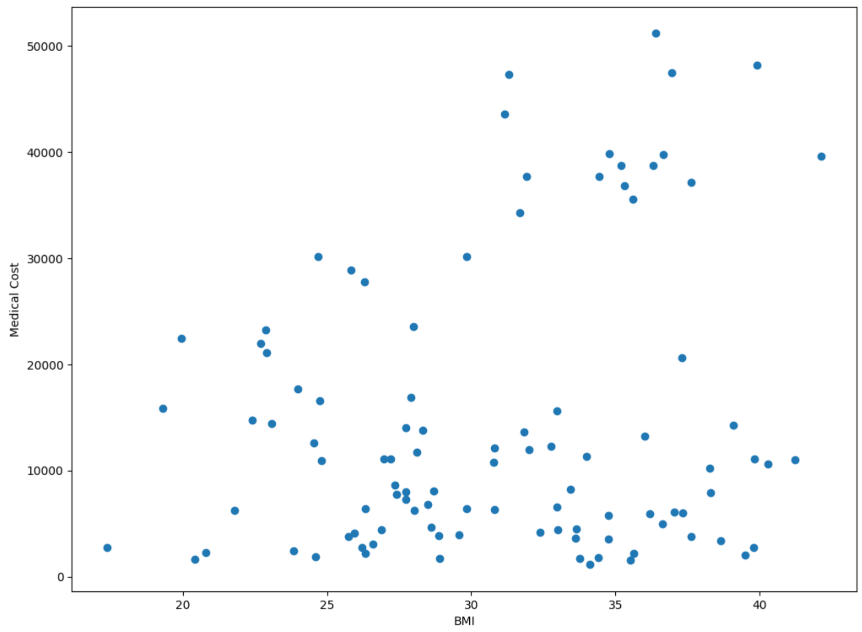
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (12.0, 9.0)
```

#Importing the data set and slicing

```
data = pd.read_csv('/content/insurance.csv')

X= data.iloc[:100,2].values # BMI

Y = data.iloc[:100,-1].values #Medical Insurance Claimed

plt.scatter(X,Y)
```

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

#Initial Scatter Plot:



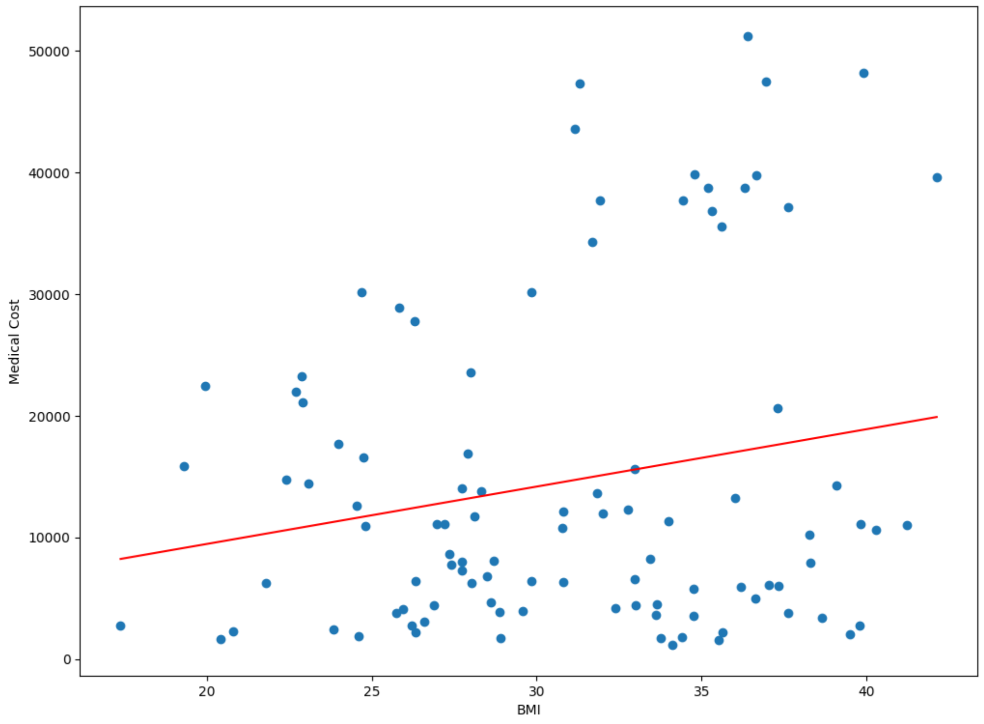
#Applying Gradient Descent Algorithm

```
m = 0
c = 0

L = 0.0001  # The learning Rate
epochs = 1000  # The number of iterations to perform gradient descent

n = float(len(X)) # Number of elements in X

# Performing Gradient Descent
for i in range(epochs):
    Y_pred = m*X + c  # The current predicted value of Y
    D_m = (-2/n) * sum(X * (Y - Y_pred))  # Derivative wrt m
    D_c = (-2/n) * sum(Y - Y_pred)  # Derivative wrt c
    m = m - L * D_m  # Update m
    c = c - L * D_c  # Update c

print (m, c)
```

#Plotting the predicted

```
Y_pred = m*X + c

plt.scatter(X, Y)
plt.plot([min(X), max(X)], [min(Y_pred), max(Y_pred)], color='red')  # 
regression line
```

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025



Inference:

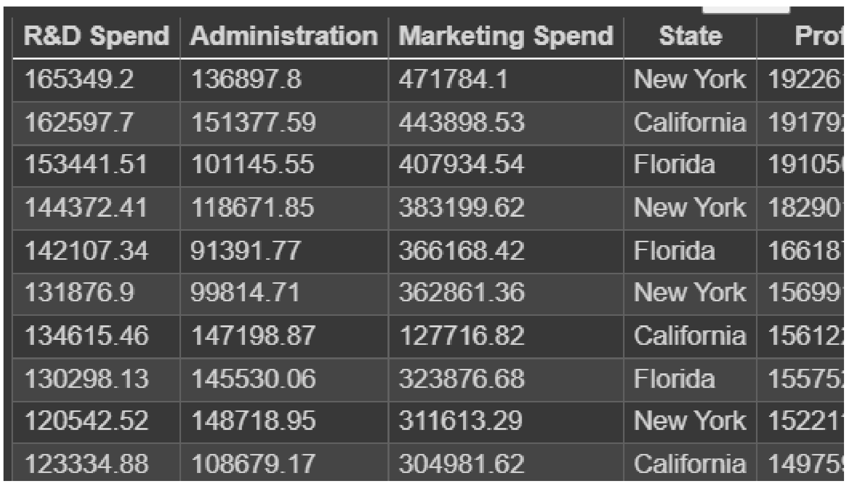
After applying linear regression using the gradient descent algorithm to analyze the relationship between BMI and medical costs, we observe a clear pattern: as BMI increases, medical costs also tend to increase. This inference aligns with the fundamental concept of linear regression, which aims to identify the best-fit line that minimizes the mean squared error between the predicted and actual values. By iteratively adjusting the parameters of the regression model, we find a line that optimally represents the relationship between BMI and medical costs in the dataset. Thus, this analysis suggests that BMI serves as a significant predictor for estimating medical expenses.

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

Multiple Linear Regression:

Multiple linear regression extends the concept of simple linear regression by accommodating multiple inputs that collectively influence the output. Rather than just one input affecting the outcome, this technique considers several factors simultaneously. By fitting a line (or a plane in higher dimensions) through the data points, multiple linear regression helps us understand how these various inputs collectively contribute to the observed outcomes. It's a versatile and powerful tool in statistics and machine learning, enabling us to predict outcomes based on multiple variables, thereby enhancing our understanding of complex relationships within the data.

Dataset Used:



R&D Spend	Administration	Marketing Spend	State	Prof
165349.2	136897.8	471784.1	New York	19226
162597.7	151377.59	443898.53	California	19179
153441.51	101145.55	407934.54	Florida	19105
144372.41	118671.85	383199.62	New York	18290
142107.34	91391.77	366168.42	Florida	16618
131876.9	99814.71	362861.36	New York	15699
134615.46	147198.87	127716.82	California	15612
130298.13	145530.06	323876.68	Florida	15575
120542.52	148718.95	311613.29	New York	15221
123334.88	108679.17	304981.62	California	14975

The above listed is a dataset containing the financial details involved in operating 50 startups.

- R&D Spend: Amount of money spent on Research and Development.

- Administration: Amount of money spent on administrative tasks.

- Marketing Spend: Amount of money spent on marketing activities.

- State: Location of the company (categorical variable).

- Profit: The profit earned by the company.

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

Code:

#Importing Libraries

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

#Importing dataset and slicing:

```
dataset = pd.read_csv('/content/50_Startups.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

#Feature scaling and encoding the data in dataset:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(),
[3])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
```

#Splitting the data into Training and Testing.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)
```

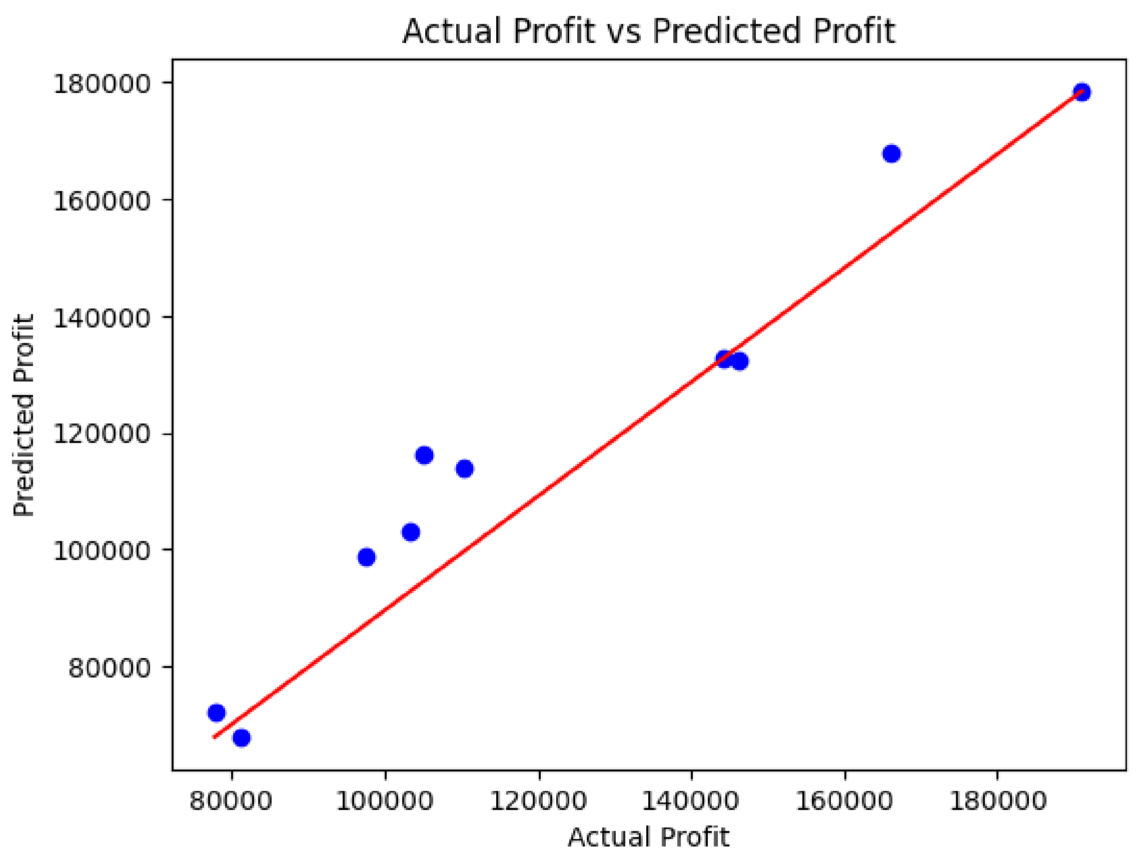
#Training the Multiple Linear Regression Model on the Training Data

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Plotting
plt.scatter(y_train, X_train color='blue')
```

T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

#Predicting the Testing data results:

```
y_pred = regressor.predict(X_test)
# np.set_printoptions(precision=2)
print(np.concatenate((y_pred.reshape(len(y_pred),1),
y_test.reshape(len(y_test),1)),1))

# Assuming you've already defined y_pred and y_test
y_pred = regressor.predict(X_test)

# Plotting
plt.scatter(y_test, y_pred, color='blue')

plt.plot([min(y_test),max(y_test)],[min(y_pred),max(y_pred)],
color="red")
plt.title('Actual Profit vs Predicted Profit')
plt.xlabel('Actual Profit')
plt.ylabel('Predicted Profit')
plt.show()
```



T. Mohamed Nadhim
3122 21 3002 058
ECE A 2025

Inference:

The multiple linear regression analysis revealed that R&D spending, administration costs, marketing expenses, and the state of operation all affect company profits. R&D and marketing spending positively correlate with profit, while administration costs have a weaker impact. Additionally, the state of operation also influences profitability.

Project Link:

https://github.com/Nadhim/ML-Lab/tree/main/Experiment_1%20-%20Linear%20Regression