**MIT xPRO Data Engineering Certificate**

**Linear Regression**

Linear regression is a machine learning algorithm that is used to predict values within a continuous range (i.e., stocks’ price and height) rather than classifying them into categories (i.e., male or female).

The values that you want to predict are the dependent variable(s). The predictions can be made using one or more independent variables.

There are two types of linear regression: simple and multiple.

**Simple Linear Regression**

Simple linear regression is the most basic form of linear regression. Its mathematical formulation is based on the traditional equation of the line:

𝑦=𝑚𝑥+𝑏

where

𝑥 is the independent variable, 𝑦 is the dependent variable, 𝑚 is the slope, and 𝑏 is the intercept with the y-axis.

In the context of linear regression, 𝑥 represents the input variable that is used to make the prediction, whereas 𝑦represents the prediction. Additionally, in the context of machine learning, 𝑚is the weight and 𝑏 is the bias.

One of the most important assumptions of linear regression is that the data has a linear relationship. However, in the real world, data is often not perfectly linear, but rather it will be in the form of a cluster of data points on a scatter *plot*. The goals of linear regression are to compute the line of best fit that describes the linear properties of the data and to determine how well this line fits the data in the cluster of points.

**Simple Linear Regression by Hand**

To understand the concepts above, let’s try to compute the result for a linear regression example.

Let’s consider the dataset below:

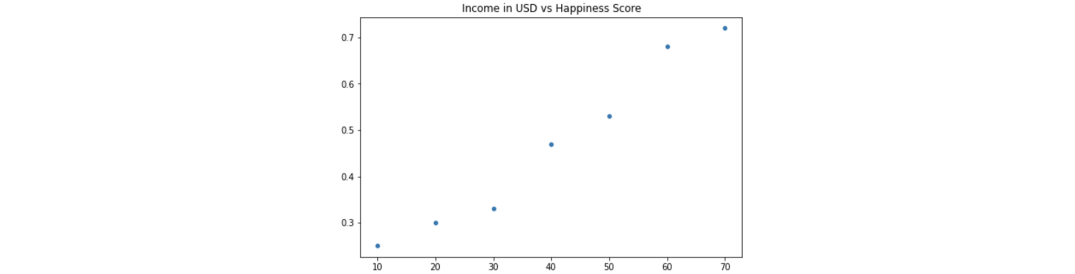
Income and Happiness Score Dataset

|  | Income in USD | Happiness Score |
| --- | --- | --- |
| 0 | 10 | 0.25 |
| 1 | 20 | 0.30 |
| 2 | 30 | 0.33 |
| 3 | 40 | 0.47 |
| 4 | 50 | 0.53 |
| 5 | 60 | 0.68 |
| 6 | 70 | 0.72 |

You can *plot* the data above in Python as a scatter *plot* using the Seaborn *library,* a Python *library* for data visualization. The Seaborn *library* is widely used in data science as it integrates well with pandas data structures, such as *series* and *dataframes*.

To install Seaborn, run the following command in a Terminal window:

pip install seaborn



From the *plot* above, you can observe that the data follows a linear pattern. Therefore, as an example, you can use linear regression to make an accurate prediction of the actual happiness score for an income equal to 30.

Given the data points above, you can compute the line of best fit that will help you to make your prediction:

| Income in USD (𝑥) | Happiness Score (𝑦) | 𝑥𝑦 | 𝑥2 | 𝑦2 |
| --- | --- | --- | --- | --- |
| 10 | 0.25 | 2.5 | 100 | 0.0625 |
| 20 | 0.30 | 6 | 400 | 0.09 |
| 30 | 0.33 | 9.9 | 900 | 0.1089 |
| 40 | 0.47 | 18.8 | 1,600 | 0.2209 |
| 50 | 0.53 | 26.5 | 2,500 | 0.2809 |
| 60 | 0.68 | 40.8 | 3,600 | 0.4624 |
| 70 | 0.72 | 50.4 | 4,900 | 0.5184 |
|  |  |  |  |  |
| ∑𝑥 | ∑𝑦 | ∑𝑥𝑦 | ∑𝑥2 | ∑𝑦2 |
| 270 | 3.03 | 152.4 | 13,900 | 1.6815 |

LaTeX:  \begin{equation}

m = \frac{n\sum{xy} - \sum{x}\sum{y}}{n\sum{x^2}- (\sum{x})^2} = \frac{7(152.4) - (270)(3.03)}{7(13,900) - 270^2} = \frac{248.7}{24400} = 0.01

\end{equation}

LaTeX: \begin{equation}

b = \frac{\sum{y}- m\sum{x}}{n} = \frac{3.03 - (0.01)(270)}{7} = \frac{0.33}{7} = 0.047

\end{equation}

So, for this dataset, the line of best fit has the equation:

𝑦=0.01𝑥+0.047

Now, to predict the happiness score for the value 30, you can just substitute 30 for and compute the prediction.

𝑦30=0.01∗30+0.047=0.347

This result, although not that different from the one you had initially with the data, represents a much more accurate result for the happiness score.

Now that you have seen how simple linear regression works, let’s have a look at multiple linear regression.

**Multiple Linear Regression**

Multiple linear regression can be viewed as a natural extension of simple linear regression. The main difference is that now multiple sets of data are used to make the prediction.

Therefore, the equation for multiple linear regression becomes:

𝑦=𝑚1𝑥1+𝑚2𝑥2+...+𝑚𝑛𝑥𝑛+𝑏,

where

𝑥1,𝑥2,...,𝑥𝑛 represent the data points of all the independent variables used to make the prediction, 𝑚1,𝑚2,...,𝑚𝑛 are the weights, and *b* is the bias.

A practical application of multiple linear regression would be, for example, to predict the price of a house (the dependent variable) based on multiple factors, such as the number of bedrooms, the square footage, and the distance from downtown. All of these variables represent the independent variables.