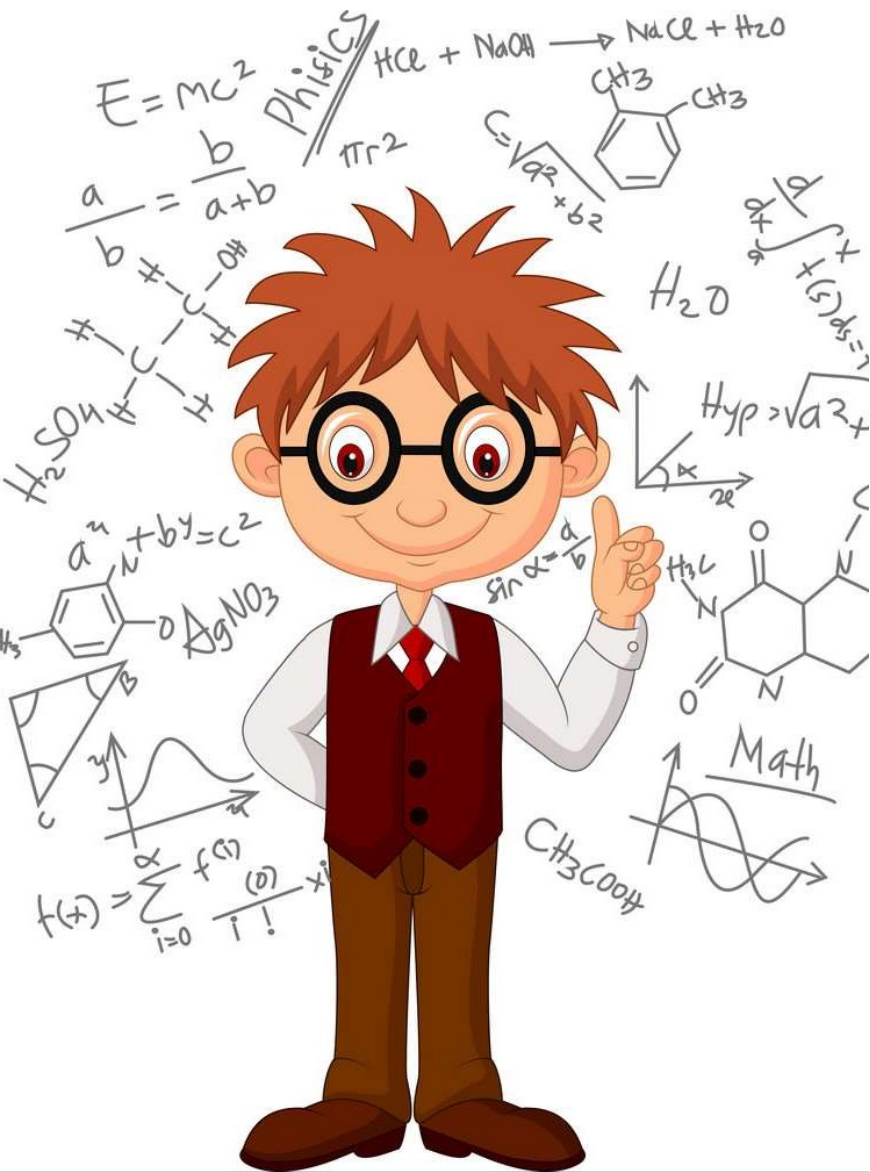


# Predictive Data Analysis

Lecturer: Marina Iantorno

E-mail: [miantorno@cct.ie](mailto:miantorno@cct.ie)





In today's class we will cover:

- ☐ Multiple Regression Model
- ☐ Dummy variables creation
- ☐ Polynomial Regression Model
- ☐ Practice in Python

# Regression Models

---

## MULTIPLE LINEAR REGRESSION

A solid orange horizontal bar spanning the width of the slide, located at the bottom.

# Multiple Regression Model

The idea of regression is to build a model that estimates or predicts one quantitative variable ( $y$ ) by using at least one other quantitative variable ( $x$ ). Simple Linear Regression uses one  $x$  variable to estimate the  $y$ . Multiple linear regression, on the other hand, uses more than one  $x$  variable.

# Multiple Regression Model

There are many situations in which we need to involve more than one variable to make our prediction, especially if the  $y$  variable is hard to track down, in other words, when the value cannot be measured straight up, and you need more than one piece of information to help get a handle on what its value will be

For example, if you want to estimate the price of the gold today, it would be hard to do it with only one variable. You may need multiple indicators such as recent gold price, the price of other commodities on the market that move against the gold, and many other possible economic conditions associated with the price of gold.

# Multiple Regression Model

In the simple linear regression, the line is represented by  $y = b_0 + b_1x_1$ , where  $a$  is the y-intercept and  $b$  is the slope.

In the multiple regression model, we work with many variables, namely  $x_1, x_2, x_3, \dots, x_k$ , and we will use some or all those variables to estimate  $y$  where each  $x$  variable is taken to the first power, and the linear equation looks like  $y = b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k$ .

# Multiple Regression Model

In order to conduct multiple regression analysis, we need to do a pre-work before actually running the regression. Here are the steps:

1. Generate a list of potential variables, independent(s) and dependent
2. Check the data.
3. Use the independent variables in the analysis to find the best fitting model.
4. Use the best fitting model to make prediction about the dependent variable.

# Multiple Regression Model

1. Generate a list of potential variables, independent(s) and dependent:

This step will likely take more time than any other steps, except maybe the data-collection process. Deciding which x variable may be candidates for consideration in your model is a deal-breaking step, because you cannot go back and collect more data after the analysis is over.



# Multiple Regression Model


Let's see the steps with an example. Suppose that you are working for a consultant company, and you are in charge to analyse the performance of 50 startups companies in the US. Your task is to decide in what company it is better to invest. Let's have a closer look to our data (50\_Startups.csv).

R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	California	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94


← Variables

# Multiple Regression Model

Potential Independent Variables



R&D Spend	Administration	Marketing Spend	State	Profit
165349.2	136897.8	471784.1	New York	192261.83
162597.7	151377.59	443898.53	California	191792.06
153441.51	101145.55	407934.54	Florida	191050.39
144372.41	118671.85	383199.62	New York	182901.99
142107.34	91391.77	366168.42	Florida	166187.94



$$y = b_0 + b_1 \cdot X_1 + b_2 \cdot X_2 + b_3 \cdot X_3 + b_4 \cdot D_1$$



$$\text{Profit} = b_0 + b_1 \cdot \text{R\&D} + b_2 \cdot \text{Admin} + b_3 \cdot \text{Mkt} + b_4 \cdot \text{State}$$

Dependent Variable

# Multiple Regression Model

Careful here. We need to encode the categorical data, and when we have only 2 options, by doing so, we are creating what is called “Dummy variables”. A dummy variable is an alternative to the categorical data to proceed with the ML models. We will find it 0 when the values are replaced by 0 and 1. Something that we need to think of is that we cannot include more than 1 Dummy variable, and this is because we could fall under multicollinearity, which means that we are working with data that is already included in the analysis. The principle is that, in probabilities, we could say that:


$$D_2 = 1 - D_1$$

Using one dummy variable, we will be including in the analysis the existence of the complement.

# Multiple Regression Model

Conclusion:

When we work with dummy variables, **always** omit one of them.

$$y = b_0 + b_1 * X_1 + b_2 * X_2 + b_3 * X_3 + b_4 * D_1 + b_5 * D_2$$


# Multiple Regression Model

## 2. Check the data.

In this step you need to ensure that the variables will have their pair (y-value and x-values) and pick the correct independent variables that will affect the results of the dependent variable.

There are some assumption to proceed with the Multiple Linear Regression Model.

# Multiple Regression Model

## Assumptions:

- Linearity (the model can be represented with a straight line)
- Homoscedasticity (the error to be the same between the independent variables).
- Normality of the data (no many outliers in the sample)
- Independence between the independent variables to avoid multicollinearity.

*\*Note: As we will use Python, the library ScikitLearn automatically takes care of selecting statistically significant features to make accurate predictions, therefore, we don't need to check these assumptions one by one.*

# Multiple Regression Model

3. Use the independent variables in the analysis to find the best fitting model.

We already identified the group of  $x$  variables; we are ready to find the best-fitting model for the data. At this point, we have to find the coefficients to put in for  $b_0$ ,  $b_1$ , and so on. Finding the best-fitting linear equation is like finding the best-fitting line in simple linear regression, except that you are not finding a line, you are estimating a best-fitting plane for the data.

# Multiple Regression Model

4. Use the best fitting model to make prediction about the dependent variable.

After we found the model, we check the accuracy. Remember that the accuracy is the determination coefficient, and we will accept it when it is greater than 50%.



# Multiple Regression Model

Let's try it in Python!

# Multiple Regression Model

Some important outputs.

```
# Predicting the Test set results
#Remember that we need to check our training results on the Test set but we can't plot a graph
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2) #we display values with only 2 decimals after the comma
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

1<sup>st</sup> argument

2<sup>nd</sup> argument

y\_pred = the prediction profit

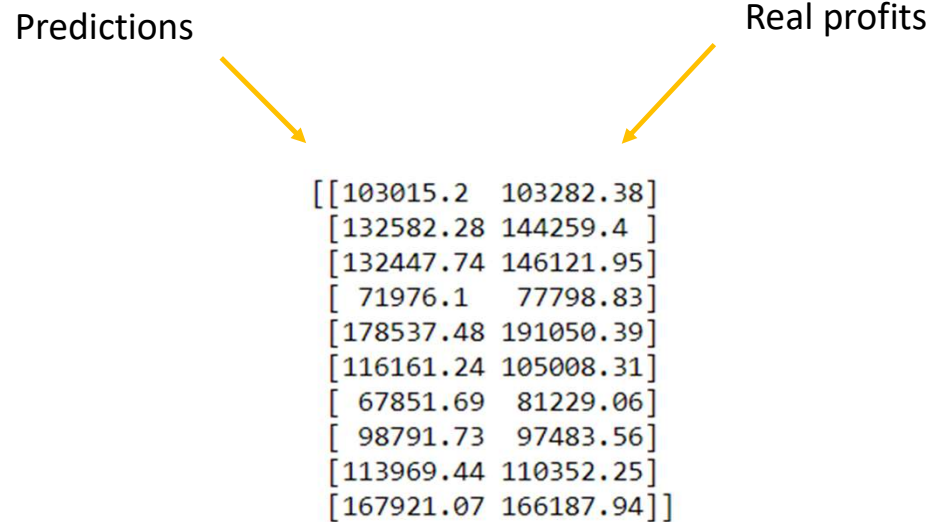
y\_test = the real profit that comes from the 20% of the dataset

# Multiple Regression Model

Some important outputs.

Predictions

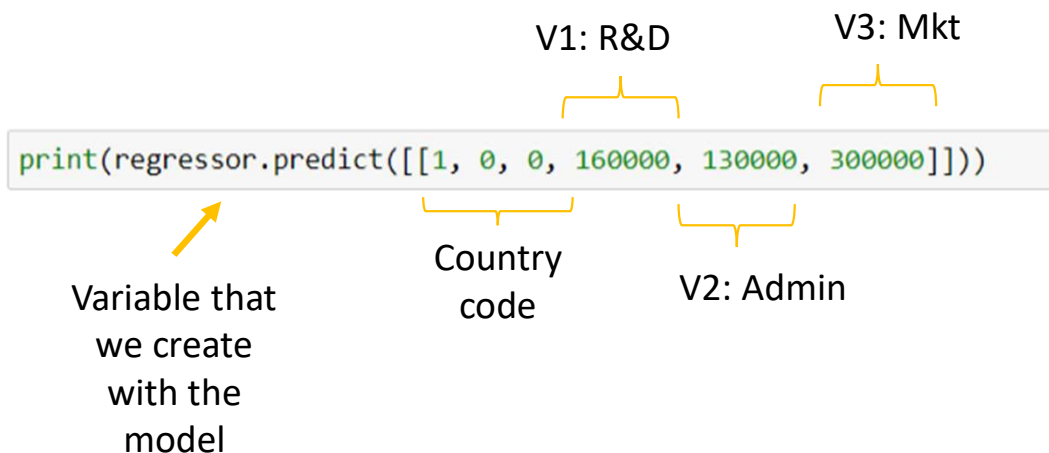
Real profits



[[103015.2 103282.38]  
[132582.28 144259.4 ]  
[132447.74 146121.95]  
[ 71976.1 77798.83]  
[178537.48 191050.39]  
[116161.24 105008.31]  
[ 67851.69 81229.06]  
[ 98791.73 97483.56]  
[113969.44 110352.25]  
[167921.07 166187.94]]

# Multiple Regression Model

Suppose that you want to make a prediction on how much profit would make a company that is in California and spent \$160,000 in R&D, \$130,000 in Administration and \$300,000 in Marketing.



# Multiple Regression Model

Some important outputs.

How to read our linear regression function?

$\text{Profit} = 86.6 * \text{Dummy State 1} - 873 * \text{Dummy State 2} + 786 * \text{Dummy State 3} + 0.773 * \text{R\&D} + 0.0329 * \text{Administration} + 0.0366 * \text{Marketing Spend} + 42467.53$

```
a =  
[ 8.66e+01 -8.73e+02  7.86e+02  7.73e-01  3.29e-02  3.66e-02]  
The interception is:  
42467.529248536506
```

# Regression Models

---

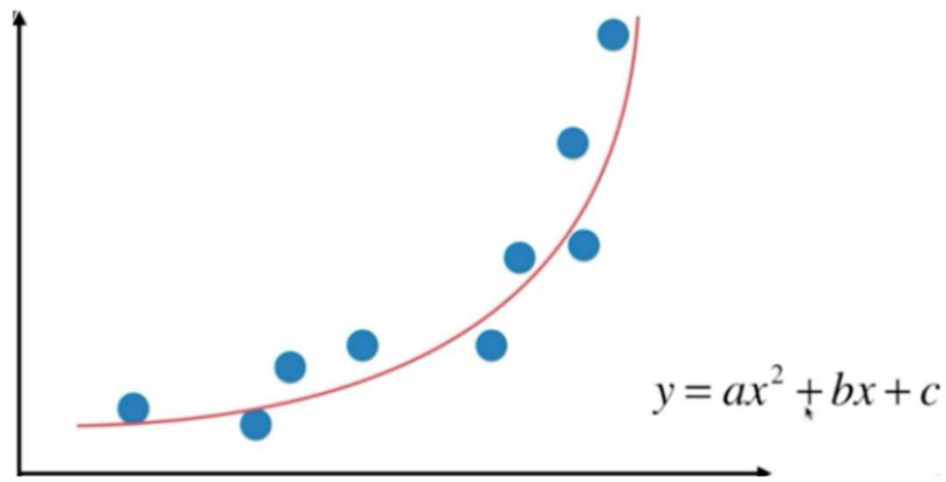
POLYNOMIAN REGRESSION



# Polynomial Regression Model

Sometimes, we discard the multiple linear regression because there is no linear correlation between the variables, however, it could be that another kind of connection exists. Here is when the Polynomial regression appears. The most general example could be associated with a quadratic function, which has only one variable on the power of another one.

# Polynomial Regression Model



As it happens with the other models, the datapoints are approximately drawing a curve with a parabolic effect due to the X squared, and the one that we can see in the graph is the curve that better fits with these observations.



# Polynomial Regression Model

This kind of model is commonly used for those events that shouldn't be considered linear. For example:

- How does certain disease spread?
- How does a pandemic spread?
- Immigration vs population and resources.
- Territory vs population.

# Polynomial Regression Model

We should keep in mind that the function of this model is a polyonomy, and therefore we should get something as follow:

$$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_nx_1^n$$

In the end, our goal is still to find the coefficients to be able to predict y based on what happens to the different independent variables.

# Polynomial Regression Model

Suppose that we are working in the HR department of a company, and there is a candidate who is very experienced and we want to hire, but when the moment to talk about salary arises, the candidate says that he expects to earn \$160,000 yearly. We ask him why and he says that this was his salary at the previous job. We will use then a Polynomial Regression Model to predict the salary of this candidate in his previous job.

# Polynomial Regression Model

We collected data to proceed. We looked on Job websites for the company where the candidate worked, and we collect the data for different positions, from Business Analyst to the CEO. This is the data that is in our dataset called “Position\_Salaries.csv”.

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000

# Polynomial Regression Model

Suppose that this person was a Regional Manager for two years, his salary should be between \$150,000 and \$200,000.

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000

We will consider that the person worked on a level between 6 and 7, so we will consider that he was on the level 6.5

# Polynomial Regression Model

What are our steps then?

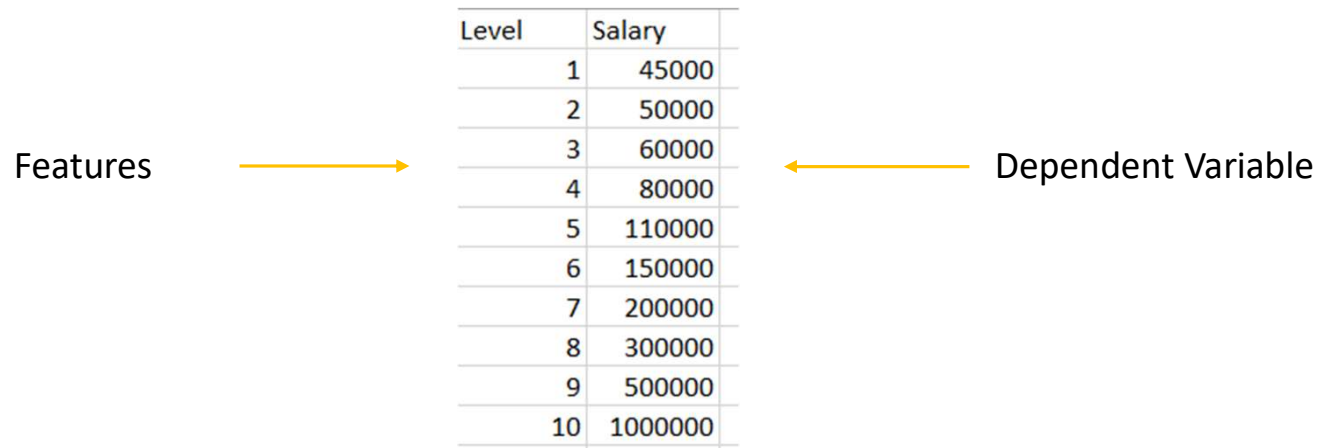
1. Data pre-processing.
2. Build the Polynomial Regression Model.
3. Train the model to understand the correlation levels between the positions and the salaries.
4. Make the prediction

# Polynomial Regression Model

Something important to consider in this case, is that we will not split the data into the training set and test set in this occasion during the Data Pre-Processing. We have just a few observations. Besides, we want to get the prediction straight away for the level between 6 and 7, and we need to use as much data as possible to get an accurate result.

# Polynomial Regression Model

Another observation is that we don't really need to do the encoding step, because the categorical variables are represented by levels, therefore the second column could be considered as the data already encoded. We will know the position according to the level.



Level	Salary
1	45000
2	50000
3	60000
4	80000
5	110000
6	150000
7	200000
8	300000
9	500000
10	1000000



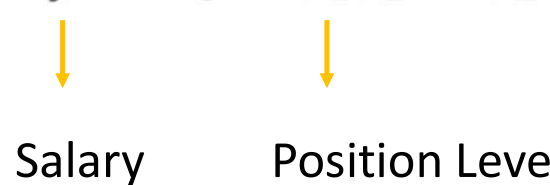
# Polynomial Regression Model

Let's try it in Python!

# Polynomial Regression Model

Some important outputs

We need to build the model, so we could think of our model as follow:

$$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_nx_1^n$$


Salary      Position Level

**THAT'S ALL FOR TODAY**

**THANK YOU**

