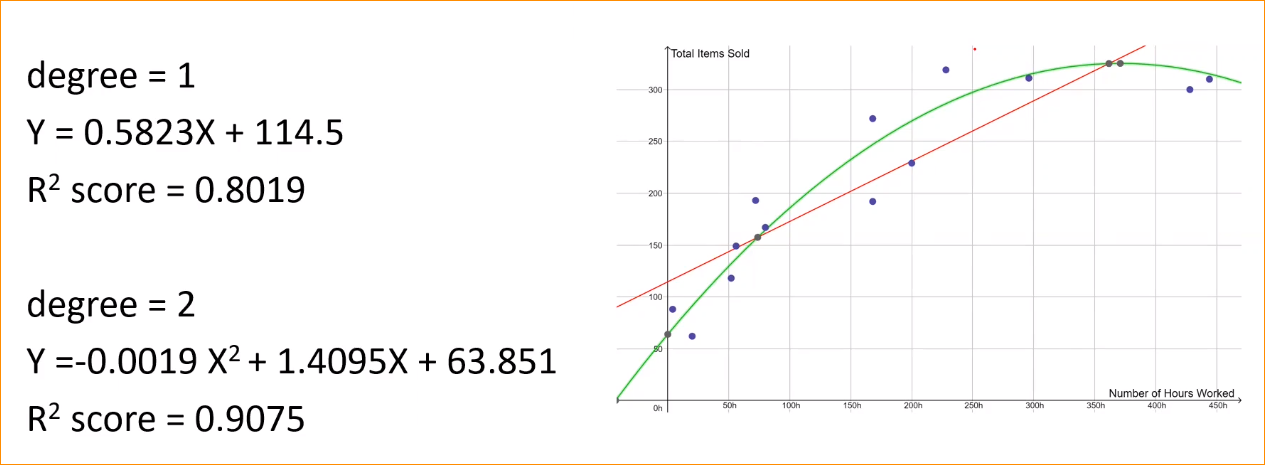
**How choose the degree of the polynomial –**

Different degree model give different perfoamce.



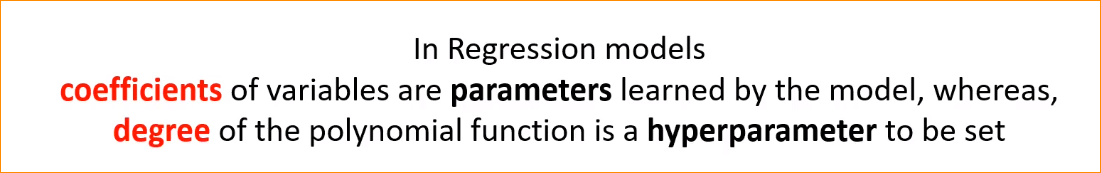
In here we can go with degree 2 because R^2 value is 0.9075.

But how to choose good model with out R^2 vgalue.

For that you should have idea of Parameters and Hyper parameters

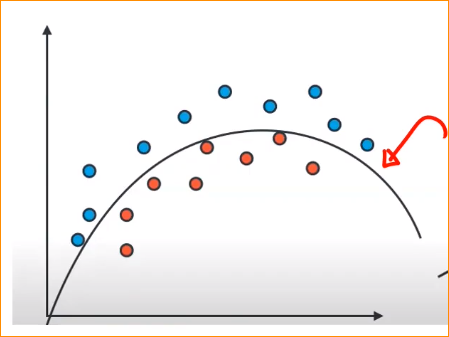
**Parameters and Hyper parameters**

|  |  |
| --- | --- |
| **Parameters** | **Hyper parameters** |
| Internal to the model (that mean – information learnt by the model during the training. Learning parameters during the training Example Theta 0 , theta 1 … those all parameters value recognize by model during the training.) | External to the model – (In here the parameters determine by Engineer Example – what is the degree of the polynomial funcation. Before beginning the training, we will set the hyper parameters. ) |
| The value of the parameter depend on the dataset.(model try to find parameter to best fit.) | Independent of the data set. |
| Specified or estimated while training the model. | Set before the beginning of the training of the model |
| Learned & set by the model by it self. | Set manually by a ML engineer or practitioner |
| Values of parameter can be estimated by the optimization algorithms, such as **Gradient Descent or it can be optimization technique. We use different algorithms that use to lean by the parameters.** | In here figuring out what would be the best one. Base on the outcome.  Values of hyper parameters can be estimated by **hyper parameter tuning.** |



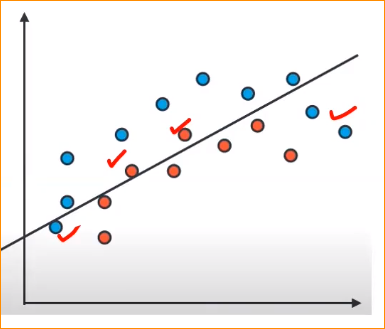
**Under fitting models-**

You have 2 groups of data. Blue and red. This graph well fit and explain the two groups.

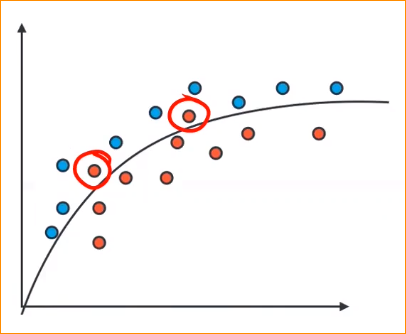


If you go one-degree polynomial. the graph looks like this –

In here you make so many mistakes in the training data- that call **Under fitting** . Check under fitting – error is too large.

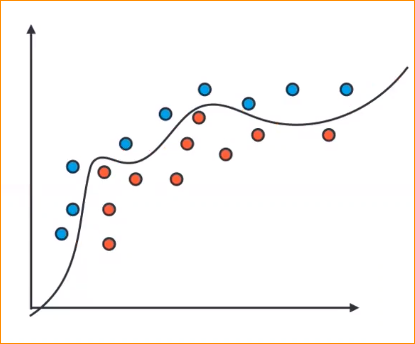


**Overfitting models-**



In we have best fitting line but 2 customers are going to the wrong side. It is may be making errors. But if you need perfect model,

You increase the degree and trying to find the good graph well fit to your training data. So you will come up with line like this.

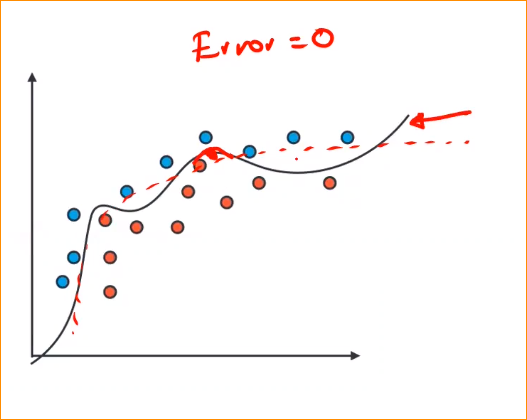


Here the error is 0.

Well fit model and all the blue customers one side and red another side.

But this is not good model.

Why – if you go with this model in the real world example –

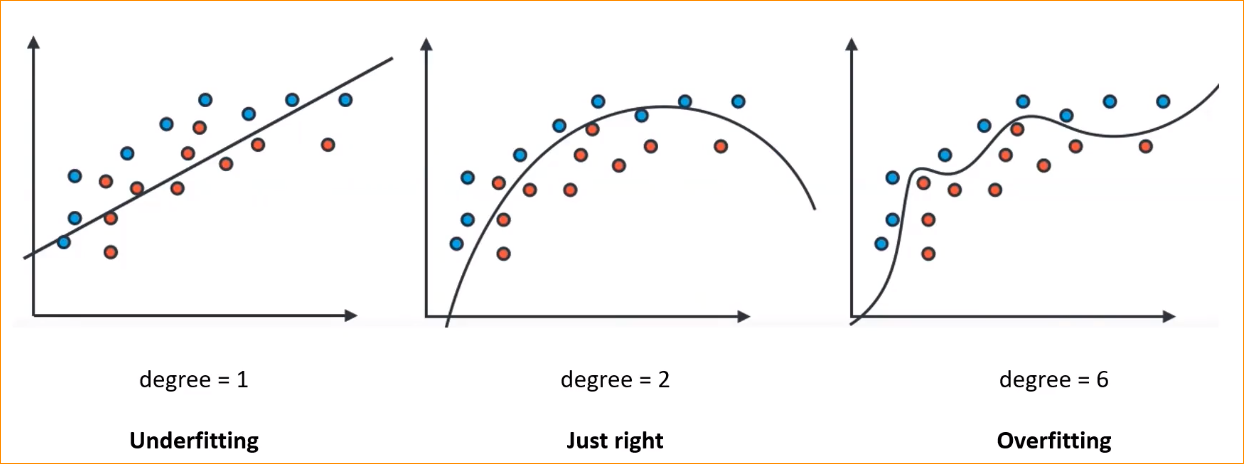


In real time if we have blue customer in here but this model recognizes as a Red in this model

Actual pattern could be this one

So that mean if you have 100% fit model . in the real word example it will predict wrong value. That’s call overfitting. Is call generalization issue. To recognize the overfitting you have to check the model with new data.

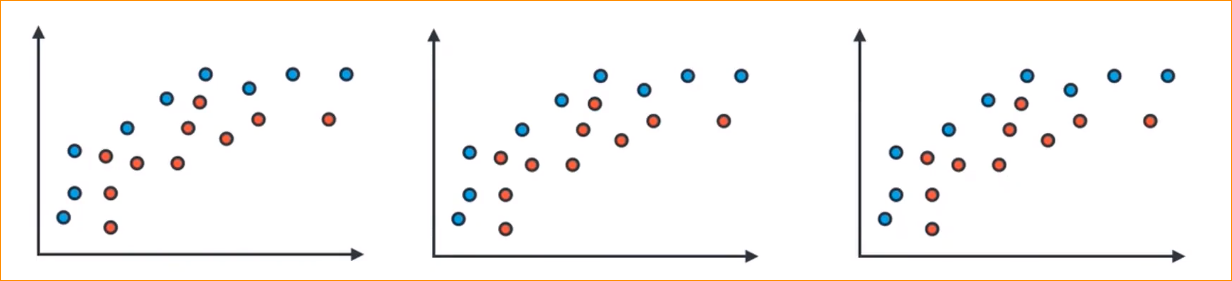
Best one in this example - degree 2



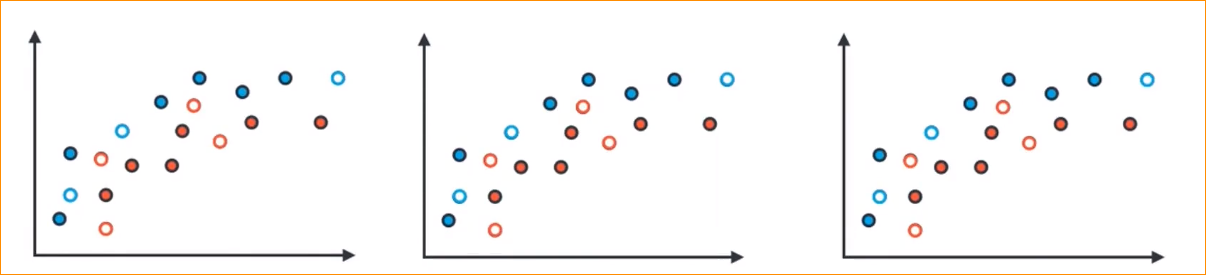
**Training vs test Data**

What is the pattern behind the error-?

Now we have a data like this –

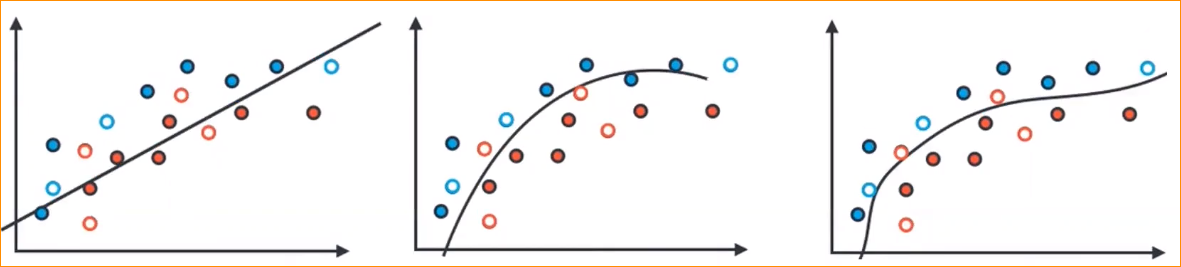


Then we are going to split it training and testing. Why – we want to check how it is performing our test data as well.

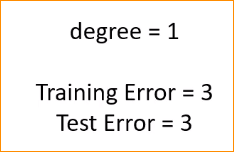
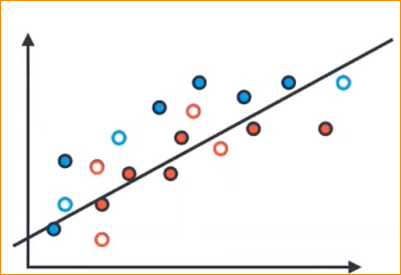


Then we will train our models- - Training data

To lover order to higher - Testing data



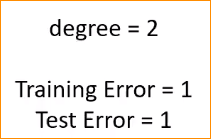
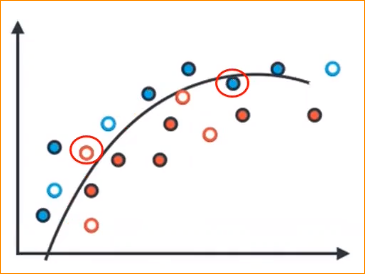
1. Degree 1



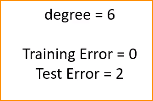
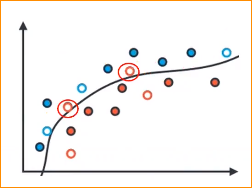
* Here 2 red color one gone wrong side. And one blue gone to wrong side. There are 3 errors in the training data.(Purple)
* Testing data (Green) – two red color gone wrong side and one blue color goes to wrong side. 3 testing error

1. Degree 2

Training and testing error has been reduce.



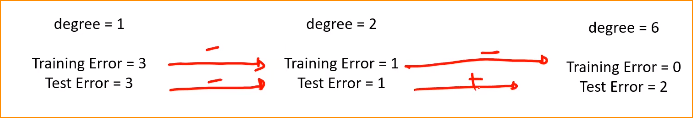
1. Degree 6



In here there is no training error-

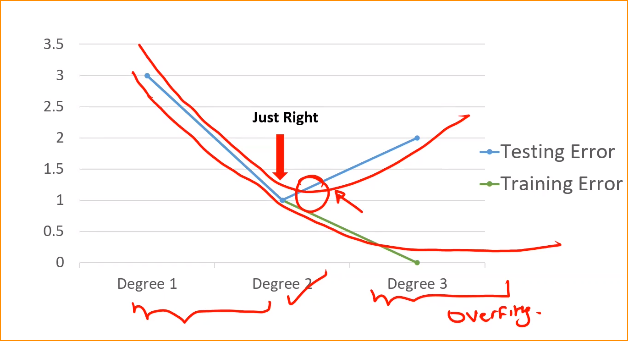
But 2 testing errors.

Summary of errors –



Increase the degree-

**Traning vs Testing data**



Training data error reduce but testing data reduce but after it will increased.

There will be minimum point to the testing data. When that start increasing we call start over fitting.

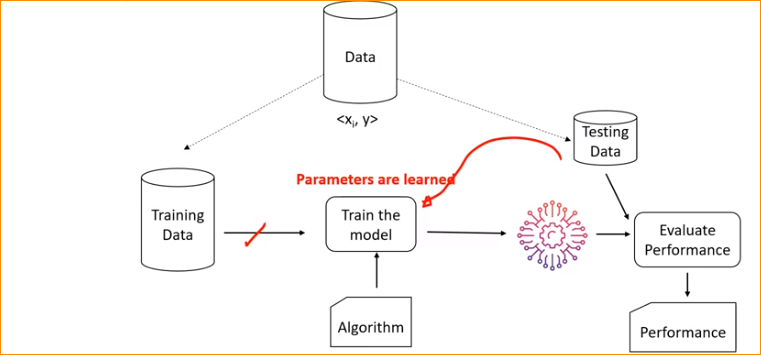
Now we can recognize what is the correct point.

Remember – The supervise learning . We are not touch any of testing data until we find the best model. So here we change the degree and should performance of test data. But that is not the right practice.

So what is the solution for that?

**Learning parameters**

Simple process- get the dataset divide two as training and testing. Use training data made model and get testing data and test. Adjust the degrees to best fit model

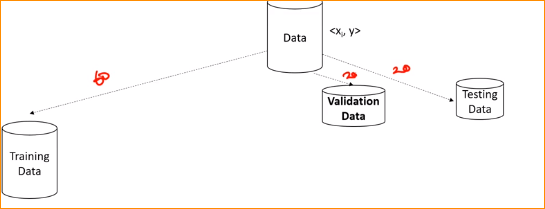


But this is not right practice-

**Choosing the Hyper parameters**

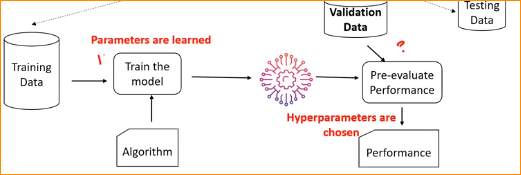
We will split data in to three potions. 3rd one is validation data set.

Usually divide data in to 60, 20 ,20 , but can divide more.



Now once we train the model, we will do evaluation from the validation data. And check the performance.

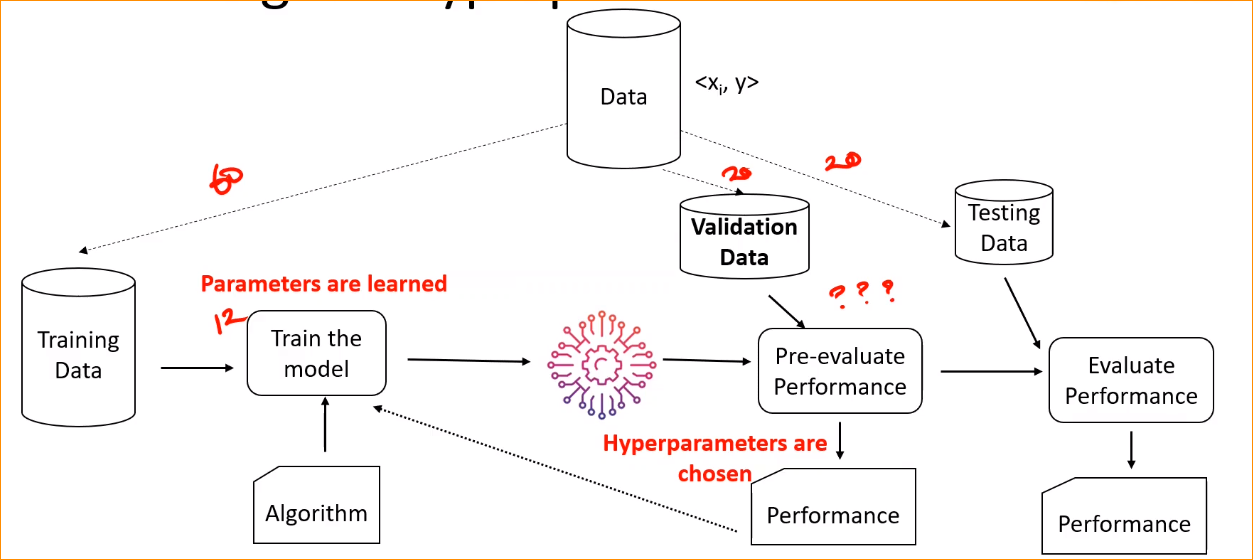
Example – use the degree one and check the performance of the degree one.



Check Performance??

Change degree-1,2,3

Like this we will create different type of parameters. Then we choose the one that best performing parameter. With the best performing hyper parameter , the training model apply to the validation data. After we made a final decision, apply the test data to evaluate the performance.



**Validation Error –**

In here we should get minimum point of validation error.

