Using overfitting to evaluate Regression Models

(Linear and Non-Linear)

Brais Galvan Sotelo (19563)
CS550 - NPU

Table of Contents

Introduction

We are going to use overfitting to compare two regression models, a linear (Model 1) and a non-linear model (Model 2). Overfitting is generally refer to training the model with too much data, and because of this, the model learns from noisy data points as well as the correct ones, generating inaccuracies in the model.

For our model we are provided with data. We will use 50% of the data for training the model, and 25% of the data for validating and the last 25% for testing the model. We will use the training and validation data for calculating the mean squared error (MSE) of the model, to be able to compare the models and see which has more overfitting issues.

Our objective is to pick the better of the two model to make a prediction on our test data.

Linear Model Implementation - Overview

Regression Equation(y) = a + bx

Intercept(a) = $(\Sigma Y - b(\Sigma X)) / N$

Slope(b) = $(N\Sigma XY - (\Sigma X)(\Sigma Y)) / (N\Sigma X^2 - (\Sigma X)^2)$

N = Number of values or elements

ΣXY = Sum of the product of first and Second Scores

ΣX = Sum of First Scores

ΣY = Sum of Second Scores

ΣX^2 = Sum of square First Scores

X	Y
1	1.8
2	2.4
3.3	2.3
4.3	3.8
5.3	5.3
1.4	1.5
2.5	2.2
2.8	3.8
4.1	4
5.1	5.4

Model Data (N = 10)

Linear Model Implementation - Calculations

Slope $(N\Sigma XY - (\Sigma X)(\Sigma Y)) / (N\Sigma X^2 - (\Sigma X)^2)$

$b = ((10)(120.8) - (31.8)(32.5)) / ((10)(121.34) - (31.8)^2)$

$b = 0.863177681$

Intercept $(\Sigma Y - b(\Sigma X)) / N)$

$a = (32.5 - 0.863177681(31.8)) / 10$

$a = 0.505094974$

Regression Equation: $0.505094974 + 0.863177681(x)$

X	Y	X*Y	X*X
1	1.8	1.8	1
2	2.4	4.8	4
3.3	2.3	7.59	10.89
4.3	3.8	16.34	18.49
5.3	5.3	28.09	28.09
1.4	1.5	2.1	1.96
2.5	2.2	5.5	6.25
2.8	3.8	10.64	7.84
4.1	4	16.4	16.81
5.1	5.4	27.54	26.01
31.8	**32.5**	**120.8**	**121.34**

Non-Linear Model Implementation - Overview

Regression Equation(y) = $a + bx^2$

Intercept(a) = $(\Sigma Y - b(\Sigma \underline{P})) / N$

Slope(b) = $(N\Sigma \underline{P}Y - (\Sigma \underline{P})(\Sigma Y)) / (N\Sigma \underline{P}^2 - (\Sigma \underline{P})^2)$

Where $\underline{P} = X * X$

Similar to the linear model, but we replace X by X * X (P)

X	Y
1	1.8
2	2.4
3.3	2.3
4.3	3.8
5.3	5.3
1.4	1.5
2.5	2.2
2.8	3.8
4.1	4
5.1	5.4

Model Data (N = 10)

Non-Linear Model Implementation - Calculations

Slope $(N\Sigma PY - (\Sigma P)(\Sigma Y)) / (N\Sigma P^2 - (\Sigma P)^2)$

$b = ((10)(509.762) - (121.34)(32.5)) / ((10)(2329.986) - (121.34)^2)$

$b = 0.134562411$

Intercept $(\Sigma Y - b(\Sigma P)) / N)$

$a = (32.5 - 0.134562411(121.34)) / 10$

$a = 1.6172197$

Regression Equation: $1.6172197 + 0.134562411(x^2)$

X	Y	P (X*X)	P * P	P * Y
1	1.8	1	1	1.8
2	2.4	4	16	9.6
3.3	2.3	10.89	118.5921	25.047
4.3	3.8	18.49	341.8801	70.262
5.3	5.3	28.09	789.0481	148.877
1.4	1.5	1.96	3.8416	2.94
2.5	2.2	6.25	39.0625	13.75
2.8	3.8	7.84	61.4656	29.792
4.1	4	16.81	282.5761	67.24
5.1	5.4	26.01	676.5201	140.454
31.8	32.5	121.34	2329.986	509.762

MSE Calculations, comparison, and test predictions

Step 1 - Applying Regression Equations to each data point in the Training and Validation data for both Model 1 and 2.
Step 2 - Calculate MSE for training data and validation
Step 3 - Comparing MSEs
Step 4 - Calculating ŷ values for Test Data using the better model

Model 1:

Regression Equation = 0.505094974 + 0.863177681(x)

Model 2:

Regression Equation = $1.6172197 + 0.134562411(x^2)$

Training Phase (50% of data)	
X	Y
1	1.8
2	2.4
3.3	2.3
4.3	3.8
5.3	5.3
1.4	1.5
2.5	2.2
2.8	3.8
4.1	4
5.1	5.4

Validation Phase (25% of	
X	Y
1.5	1.7
2.9	2.7
3.7	2.5
4.7	2.8
5.1	5.5

Test Phase (25% of data)
X
1.4
2.5
3.6
4.5
5.4

Step 1 : Apply regression equation to get ŷ values for Training and Validation data

Training Phase (50% of data)		Model 1: Linear Model	Model 2: Non-Linear Model	Validation Phase (25% of data)		Model 1: Linear Model	Model 2: Non-Linear Model
X	Y	$y = 0.505094974 + 0.863177681(x)$	$y = 1.6172197 + 0.134562411(x^2)$	X	Y	$y = 0.505094974 + 0.863177681(x)$	$y = 1.6172197 + 0.134562411(x^2)$
1	1.8	1.368272655	1.751782112	1.5	1.7	1.799861496	1.919985126
2	2.4	2.231450336	2.155469346	2.9	2.7	3.008310249	2.74888958
3.3	2.3	3.353581322	3.08260436	3.7	2.5	3.698852394	3.459379112
4.3	3.8	4.216759003	4.105278687	4.7	2.8	4.562030075	4.589703368
5.3	5.3	5.079936684	5.397077836	5.1	5.5	4.907301148	5.11718802
1.4	1.5	1.713543728	1.880962027				
2.5	2.2	2.663039177	2.458234771				
2.8	3.8	2.921992481	2.672189005				
4.1	4	4.044123467	3.879213836				
5.1	5.4	4.907301148	5.11718802				

Step 2: Calculate MSE for training data and validation

Mean Square Error is the value we utilize to understand how closely a model fits the data, the lower the MSE value is, the lower the error of the model for fitting the data.

MSE is calculated by the sum of the difference in y values squared and then divided by the number of data points.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2.$$

Model 1 Training Data:

MSE = (1\10)* ((1.368272655 - 1.8)^2 + (2.231450336 - 2.4)^2 + …) = **0.282254947**

Model 1 Validation Data:

MSE = (1\5)* ((1.799861496 - 1.7)^2 + (3.008310249 - 2.7)^2 + …) = **0.999663301**

Model 2 Training Data:

MSE = (1\10)* ((1.751782112 - 1.8)^2 + (2.155469346 - 2.4)^2 + …) = **0.235555579**

Model 2 Validation Data:

MSE = (1\5)* ((1.919985126 - 1.7)^2 + (2.74888958 - 2.7)^2 + …) = **0.864155017**

Step 3: Comparing MSEs values

Model 1 :

Training MSE = 0.282254947
Validation MSE = 0.999663301

max(Training_MSE, Validation_MSE) / min(Training_MSE, Validation_MSE)

= Validation MSE / Training MSE = 0.999663301 / 0.282254947 = **3.541703391**

Model 2 :

Training MSE = 0.235555579
Validation MSE = 0.864155017

max(Training_MSE, Validation_MSE) / min(Training_MSE, Validation_MSE)

= Validation MSE / Training MSE = 0.864155017 / 0.235555579 = **3.668582255**

We determine **Model 1** to be better since 3.541703391 < 3.668582255.

Step 4: Calculating ŷ values for Test Data using the better model (Model 1)

Training Phase (50% of data)		Model 1: Linear Model	Model 2: Non-Linear Model	Validation Phase (25% of data)		Model 1: Linear Model	Model 2: Non-Linear Model	Test Phase (25% of data)	Model 1: Linear Model
X	Y	$y = 0.505094974 + 0.863177681(x)$	$y = 1.6172197 + 0.134562411(x^2)$	X	Y	$y = 0.505094974 + 0.863177681(x)$	$y = 1.6172197 + 0.134562411(x^2)$	X	$y = 0.505094974 + 0.863177681(x)$
1	1.8	1.368272655	1.751782112	1.5	1.7	1.799861496	1.919985126	1.4	1.713543728
2	2.4	2.231450336	2.155469346	2.9	2.7	3.008310249	2.74888958	2.5	2.663039177
3.3	2.3	3.353581322	3.08260436	3.7	2.5	3.698852394	3.459379112	3.6	3.612534626
4.3	3.8	4.216759003	4.105278687	4.7	2.8	4.562030075	4.589703368	4.5	4.389394539
5.3	5.3	5.079936684	5.397077836	5.1	5.5	4.907301148	5.11718802	5.4	5.166254452
1.4	1.5	1.713543728	1.880962027						
2.5	2.2	2.663039177	2.458234771						
2.8	3.8	2.921992481	2.672189005						
4.1	4	4.044123467	3.879213836						
5.1	5.4	4.907301148	5.11718802						

Conclusion

After evaluating the two models, we found that the linear regression model (Model 1) had less overfitting issues than the non-linear regression model (Model 2). Because the linear model was better, we used the linear regression equation that we calculated using the testing data to make predictions for the test data.