

How to evaluate regression models?

Data Science Interview Questions around model evaluation metrics



Vimarsh Karbhari [Follow](#)

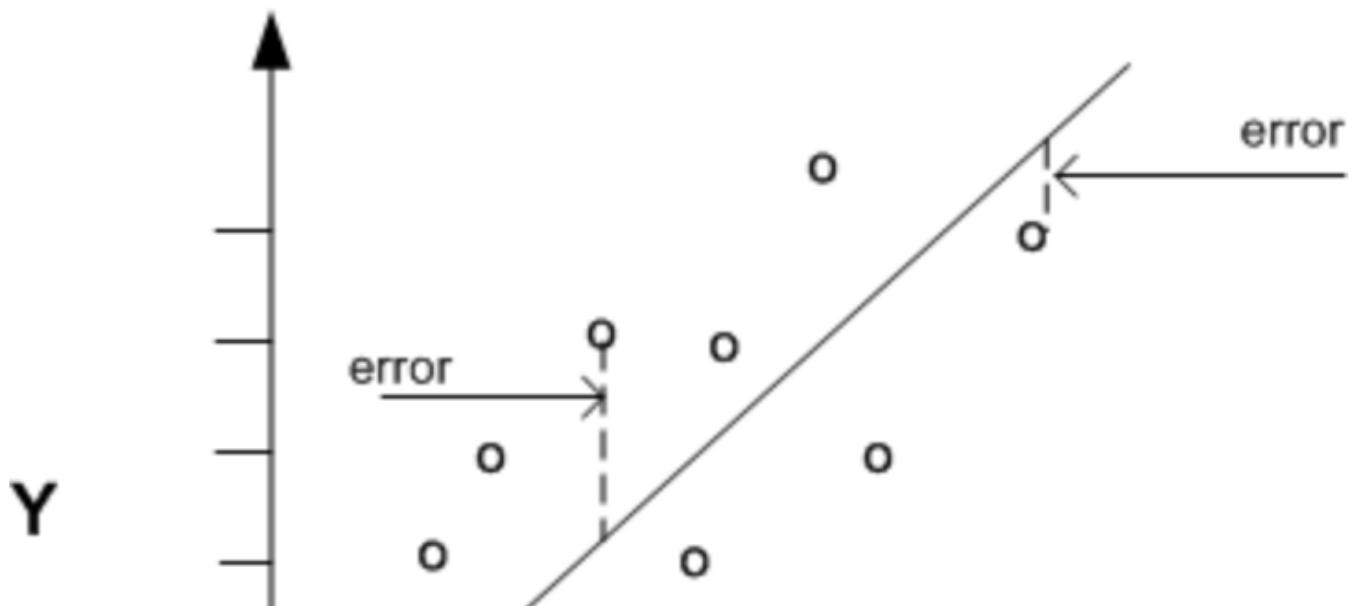
Dec 18, 2018 · 4 min read

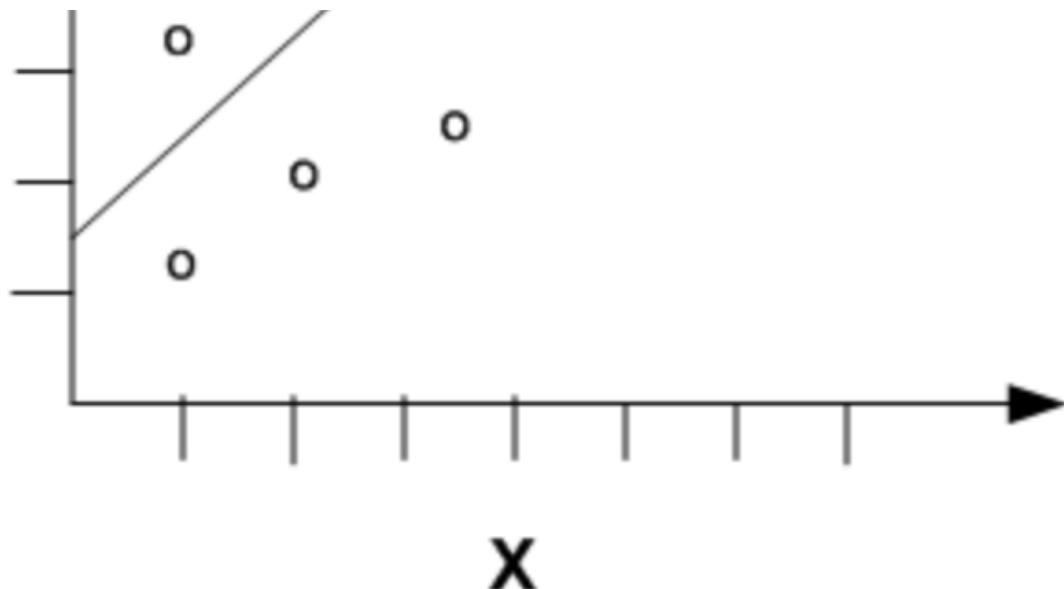
Validation and Evaluation of a Data Science Model provides more colour to our hypothesis and helps evaluate different models that would provide better results against our data. These are the metrics that help us evaluate our models.

What Big-O is to coding, validation and evaluation is to Data Science Models.

There are three main errors (metrics) used to evaluate models, **Mean absolute error**, **Mean Squared error** and **R2 score**.

Mean Absolute Error (MAE)





Source: Regression Docs

Lets take an example where we have some points. We have a line that fits those points. When we do a summation of the absolute value distance from the points to the line, we get Mean absolute error. The problem with this metric is that it is not differentiable. Let us translate this into how we can use Scikit Learn to calculate this metric.

```
>>> from sklearn.metrics import mean_absolute_error
>>> y_true = [3, -0.5, 2, 7]
>>> y_pred = [2.5, 0.0, 2, 8]
>>> mean_absolute_error(y_true, y_pred)
0.5
>>> y_true = [[0.5, 1], [-1, 1], [7, -6]]
>>> y_pred = [[0, 2], [-1, 2], [8, -5]]
>>> mean_absolute_error(y_true, y_pred)
0.75
>>> mean_absolute_error(y_true, y_pred, multioutput='raw_values')
array([0.5, 1.])
>>> mean_absolute_error(y_true, y_pred, multioutput=[0.3, 0.7])
...
0.85...
```

Mean Squared Error (MSE)

Mean Squared Error solves differentiability problem of the MAE. Consider the same diagram above. We have a line that fits those points. When we do a summation of the square of distances from the points to the line, we get Mean squared error. In Scikit learn it looks like:

```
>>> from sklearn.metrics import mean_squared_error
>>> y_true = [3, -0.5, 2, 7]
>>> y_pred = [2.5, 0.0, 2, 8]
>>> mean_squared_error(y_true, y_pred)
0.375
>>> y_true = [[0.5, 1], [-1, 1], [7, -6]]
>>> y_pred = [[0, 2], [-1, 2], [8, -5]]
>>> mean_squared_error(y_true, y_pred)
0.708...
>>> mean_squared_error(y_true, y_pred, multioutput='raw_values')
...
array([0.41666667, 1.          ])
>>> mean_squared_error(y_true, y_pred, multioutput=[0.3, 0.7])
...
0.825..
```

The mathematical representations of MAE and MSE are below:

Mean squared error

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n e_t^2$$

Root mean squared error

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$$

Mean absolute error

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |e_t|$$

Mean absolute percentage error

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right|$$

Source: Stackexchange

R2 Score

Let us take a naive approach by taking an average of all the points by thinking of a horizontal line through them. Then we can calculate the MSE for this simple model.



Source: R2 Score

R2 score answers the question that if this simple model has a larger error than the linear regression model. However, in terms of metrics the answer we need is how much larger. The R2 score answers this question. R2 score is $1 - \frac{\text{Error from Simple Average Model}}{\text{Error from Linear Regression Model}}$.

Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R² score of 0.0. In Scikit Learn it looks like:

```
>>> from sklearn.metrics import r2_score
>>> y_true = [3, -0.5, 2, 7]
>>> y_pred = [2.5, 0.0, 2, 8]
>>> r2_score(y_true, y_pred)
0.948...
>>> y_true = [[0.5, 1], [-1, 1], [7, -6]]
>>> y_pred = [[0, 2], [-1, 2], [8, -5]]
>>> r2_score(y_true, y_pred, multioutput='variance_weighted')
...
0.938...
```

```
>>> y_true = [1,2,3]
>>> y_pred = [1,2,3]
>>> r2_score(y_true, y_pred)
1.0
>>> y_true = [1,2,3]
>>> y_pred = [2,2,2]
>>> r2_score(y_true, y_pred)
0.0
>>> y_true = [1,2,3]
>>> y_pred = [3,2,1]
>>> r2_score(y_true, y_pred)
-3.0
```

Conclusion

Model evaluation leads a Data Scientist in the right direction to select or tune an appropriate model. In a Data Science Interviews, it tests the fundamentals of candidates in the same way. In any interview, knowing these values and terms for the problems being discussed is table stakes.

For more such answers to important Data Science questions, please visit Acing AI.

Subscribe to our Acing AI newsletter, I promise not to spam and its FREE!

Newsletter

Subscribe to the Acing AI/Data Science Newsletter. It is FREE!

Reducing the entropy in data science. Helping you with...

www.acingdatascience.com

We have build a new course to help people ace data science interviews. Sign up below!

Acing Data Science Interviews

3-Month course for data science interviews

www.acingdatascience.com

Thanks for reading! ☺ If you enjoyed it, test how many times can you hit  in 5 seconds. It's great cardio for your fingers AND will help other people see the story.

[Machine Learning](#)[Data Science](#)[Artificial Intelligence](#)[Interview](#)[Technology](#)[About](#) [Help](#) [Legal](#)