

- ▶ Welcome
- ▶ Introduction: Machine Learning concepts
- ▶ Module 1. The Predictive Modeling Pipeline
- ▶ Module 2. Selecting the best model
- ▶ Module 3. Hyperparameter tuning
- ▶ Module 4. Linear Models
- ▶ Module 5. Decision tree models
- ▶ Module 6. Ensemble of models
- ▼ **Module 7. Evaluating model performance**

✓ Quiz M7.05

Note: For each question **make sure you select all of the correct options**— there may be more than one! Don't forget to use the sandbox notebook if you need.

Question 1 (1/1 point)

What is the default score in scikit-learn when using a regressor?

☒ a) R^2 ✓

☐ b) mean absolute error

☐ c) median absolute error

EXPLANATION

solution: a)

Each regressor in scikit-learn uses the R^2 score as a default metric: [documentation](#)

You have used 1 of 1 submissions


Question 2 (1/1 point)

If we observe that the values returned by

`cross_val_scores(model, X, y, scoring="r2")` increase after changing the model parameters, it means that the latest model:

☒ a) generalizes better ✓

model with
simple baselines

Quiz M7 

Choice of cross-
validation

Quiz M7 

Nested cross-
validation

Quiz M7 


Classification
metrics

Quiz M7 

Regression
metrics

Quiz M7 

Wrap-up quiz

Wrap-up quiz 

Main take-away

► Conclusion

► Appendix

EXPLANATION

solution: a)

You have used 1 of 1 submissions

Question 3 (1/1 point)

If all the values returned by

`cross_val_score(model_A, X, y, scoring="neg_mean_squared_error")`

are strictly lower than those returned by

`cross_val_score(model_B, X, y, scoring="neg_mean_squared_error")`,

it means that model_B generalizes:

☒ a) better than `model_A` ✓

☐ b) worse than `model_A`

EXPLANATION

solution: a)

Lower **error** values means a better model. Considering the **negative** error (i.e. multiplying by `-1`) reverses this relationship. In general, the scoring parameter passed to scikit-learn model selection utilities expects the convention that `"higher is better"`, hence the slightly counter intuitive use of `"neg_mean_squared_error"` instead of `"mean_squared_error"` in scikit-learn.

You have used 1 of 1 submissions

```
cross_val_scores(model, X, y, scoring="neg_mean_squared_error")
```

are:

- ☐ a) guaranteed to be positive or zero
- ☒ b) guaranteed to be negative or zero ✓
- ☐ c) can be either positive or negative depending on the data

EXPLANATION

solution: b)

The mean squared error is always positive or zero (never negative) since:

- taking the square of a value maps a negative value to a positive value, zero to zero and a positive value to a positive value;
- the average of an array of positive values is a positive value.

Therefore the **negative** mean squared error is guaranteed to be negative or zero by reversing the sign of the mean squared error (i.e. multiplying by `-1`).

You have used 1 of 1 submissions

YOUR EXPERIENCE

According to you, this whole 'Regression metrics' lesson was:

- ☐ **Too easy, I got bored**
- ☐ **Adapted to my skills**
- ☐ **Difficult but I was able to follow**
- ☐ **Too difficult**

To follow this lesson, I spent:

- ☐ **less than 30 minutes**
- ☐ **30 min to 1 hour**
- ☐ **1 to 2 hours**
- ☐ **2 to 4 hours**
- ☐ **more than 4 hours**
- ☐ **I don't know**

Submit

FORUM (EXTERNAL RESOURCE)

✚ New topic 

[Home](#) > [M7. Evaluating model performance](#) > [M7. Quiz M7.05](#)

There are no more M7. Quiz M7.05 topics. Ready to [start a new conversation](#)?

About...

Help and Contact

Terms of use

Terms and conditions

