

Model validation

02 July 2024

12:02

We need to evaluate the vast majority of every model we build. For most applications, this is determined by predictive accuracy. This is the key to improving and iterating on models.

Many people make a huge mistake when measuring predictive accuracy. They measure it on their *training data* and compare those predictions to the target values in the *training* problem with this approach and how to solve it in a moment, but let's think about it differently.

First we must summarise model quality in a single metric instead of looking at each prediction, which is not useful as one it would take way too long and two individually it doesn't summarise the model's performance as there will be a mix of good and bad predictions.

To solve this issue we use something called **Mean Absolute Error (MAE)**

```
error = actual - predicted  
individual case
```

With the MAE metric, we take the absolute value of each error. This converts each error to a positive number. We then take the average of those absolute errors. This is our measure of model performance. In English, it can be said as

On average, our predictions are off by about X.

To calculate this we import the MAE function from sklearn

```
from sklearn.metrics import mean_absolute_error  
predicted_home_prices = melbourne_model.predict(X)  
mean_absolute_error(y, predicted_home_prices)
```

ations model quality is
the model.

make predictions with
training data. You'll see the
but how we'd do this first.

individual case,
tell us anything

For the

r to a positive
el quality. In plain

