UFRN

Open Data Day
Natal, Brazil

01 a 03 de março de 2018

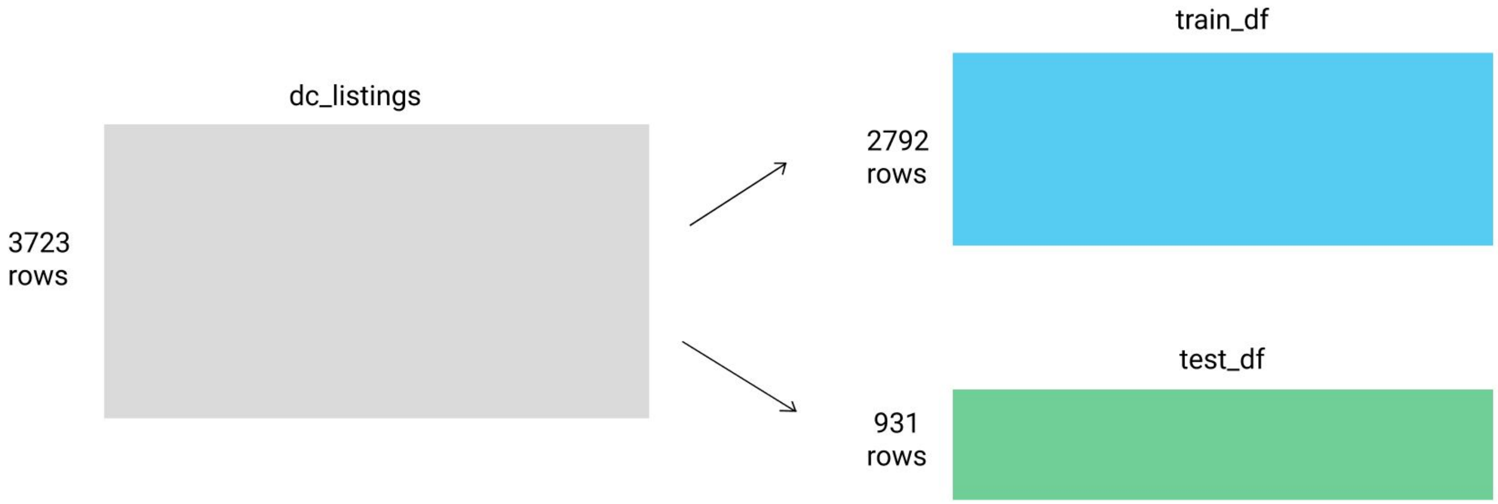
Minicourse

Machine Learning Fundamentals in Python - Evaluating a Model

Speaker:
Prof. Ivanovitch Silva (ivan@imd.ufrn.br)

Testing quality of predictions

Machine Learning Model



Error metrics

- We now need a metric that quantifies how good the predictions were on the test set.

Mean Absolute Error

$$MAE = \frac{|actual_1 - predicted_1| + |actual_2 - predicted_2| + \ \ldots + |actual_n - predicted_n|}{n}$$

Mean Squared Error

$$MSE = \frac{(actual_1 - predicted_1)^2 + (actual_2 - predicted_2)^2 + \ \ldots + (actual_n - predicted_n)^2}{n}$$

Root Mean Squared Error (RMSE)

- While comparing MSE values helps us identify which model performs better on a relative basis, it doesn't help us understand if the performance is good enough in general.
- This is because the units of the MSE metric are squared (in this case, dollars squared)

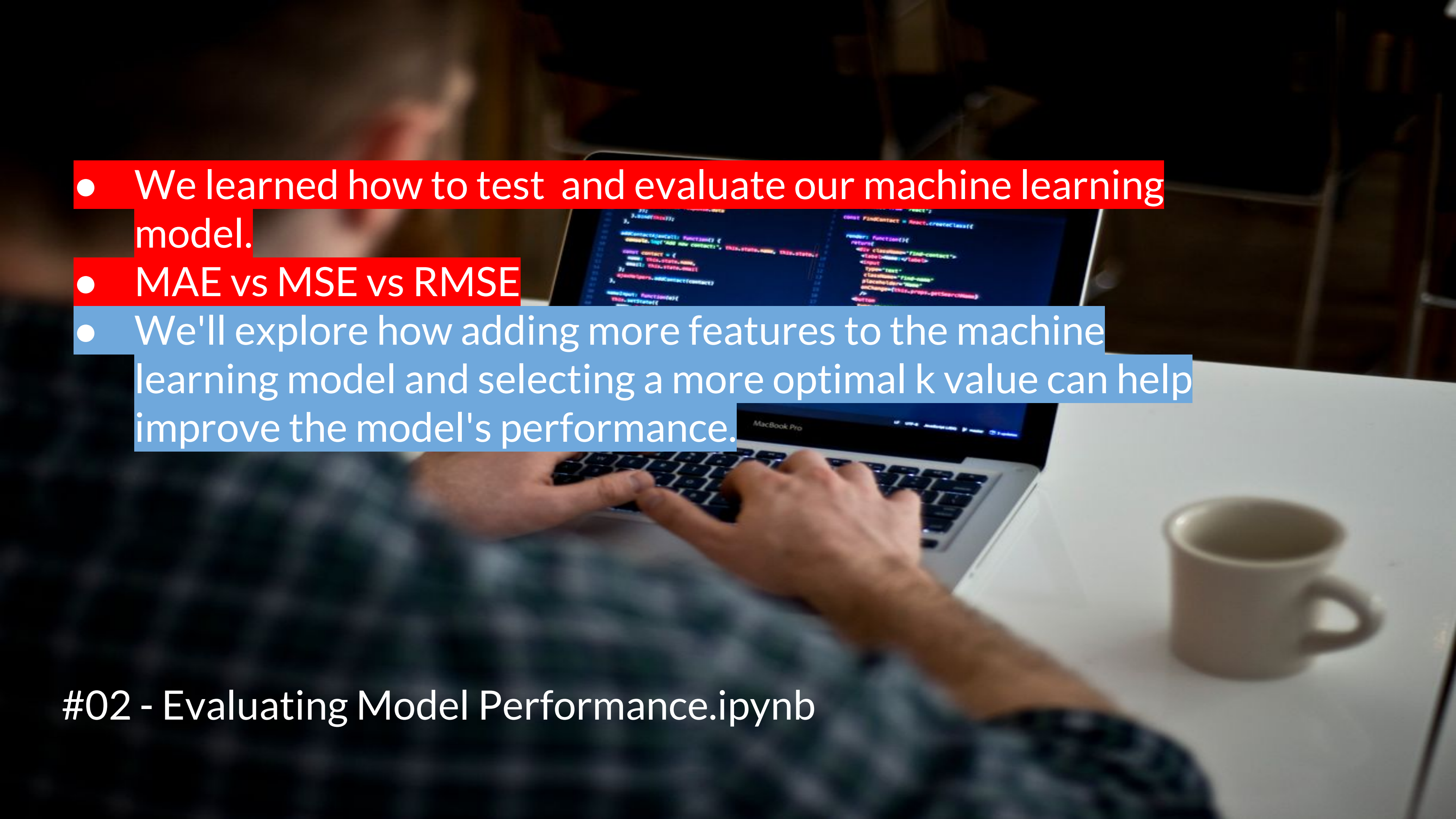
$$RMSE = \sqrt{MSE}$$

MAE vs RMSE

```
errors_one = pd.Series([5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10])
errors_two = pd.Series([5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 1000])

mae_one = np.sum(errors_one)/len(errors_one)
rmse_one = np.sqrt(np.sum(errors_one**2)/len(errors_one))
mae_two = np.sum(errors_two)/len(errors_two)
rmse_two = np.sqrt(np.sum(errors_two**2)/len(errors_two))
print("Mae_one: {}\nRmse_one: {}".format(mae_one,rmse_one))
print("Mae_two: {}\nRmse_two: {}".format(mae_two,rmse_two))
```

```
Mae_one: 7.5
Rmse_one: 7.905694150420948
Mae_two: 62.5
Rmse_two: 235.82302686548658
```



- We learned how to test and evaluate our machine learning model.
- MAE vs MSE vs RMSE
- We'll explore how adding more features to the machine learning model and selecting a more optimal k value can help improve the model's performance.

#02 - Evaluating Model Performance.ipynb