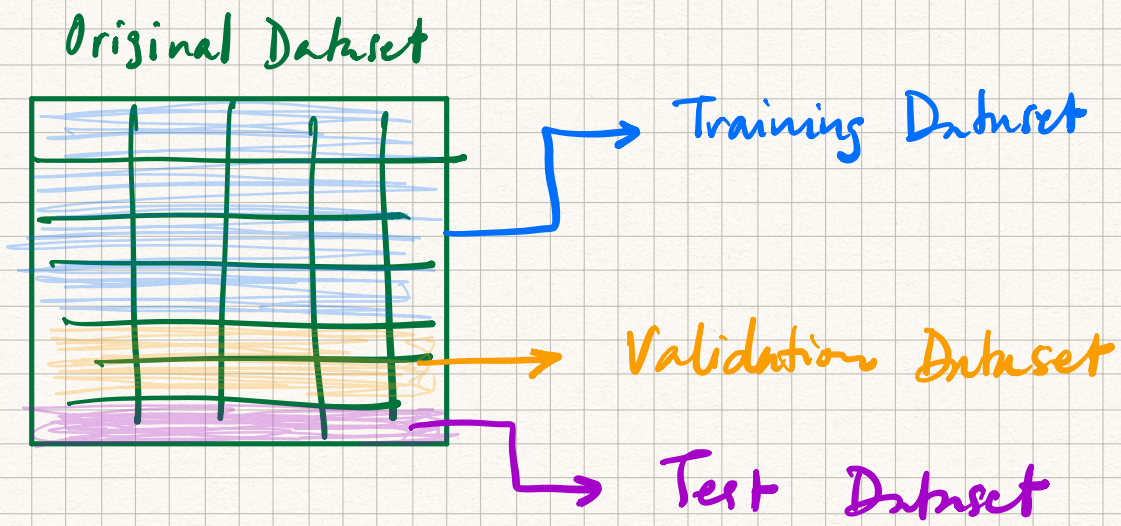
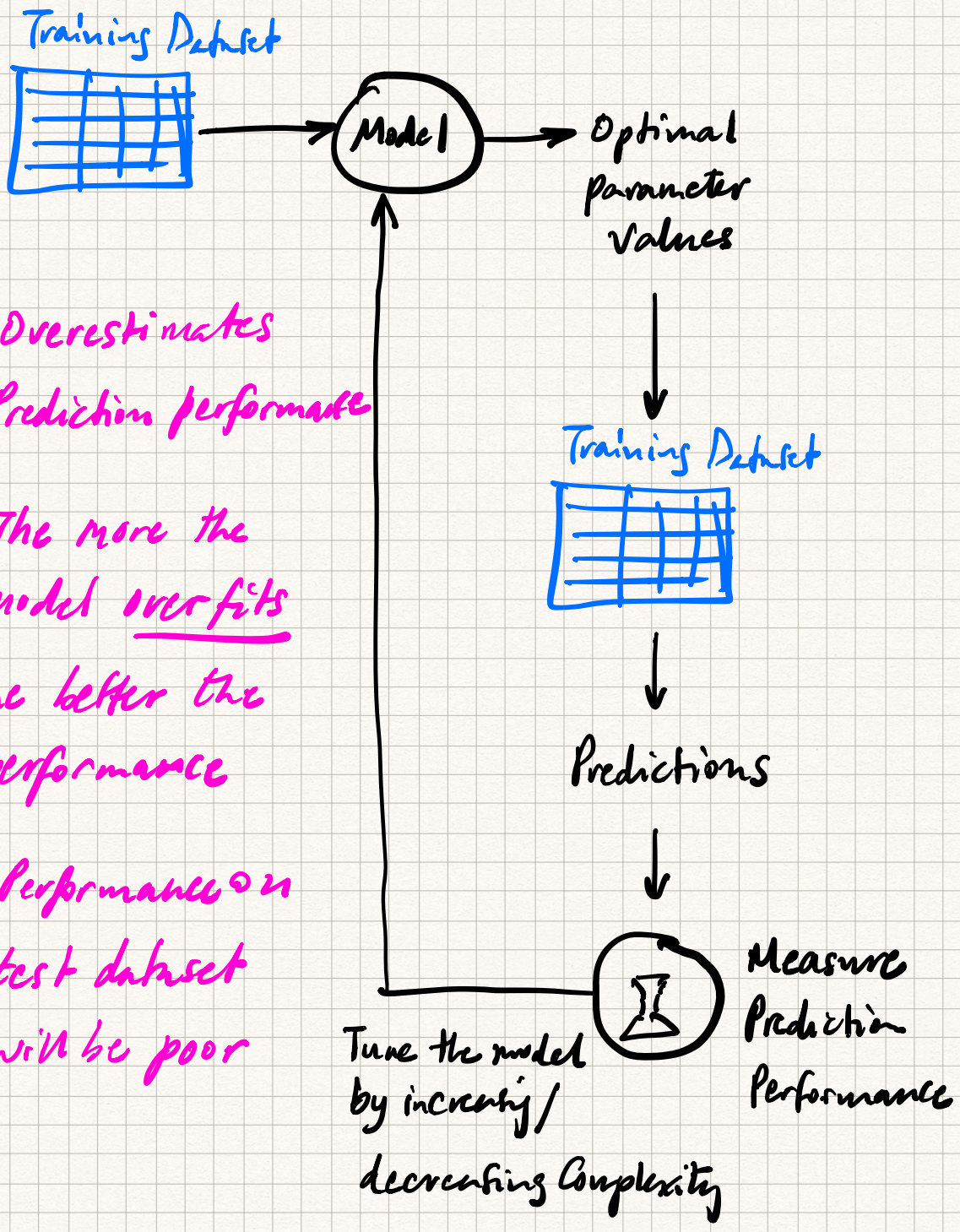
TRAINING, VALIDATION, AND TEST DATASETS

Why split up the data this way and what happens if you don't?

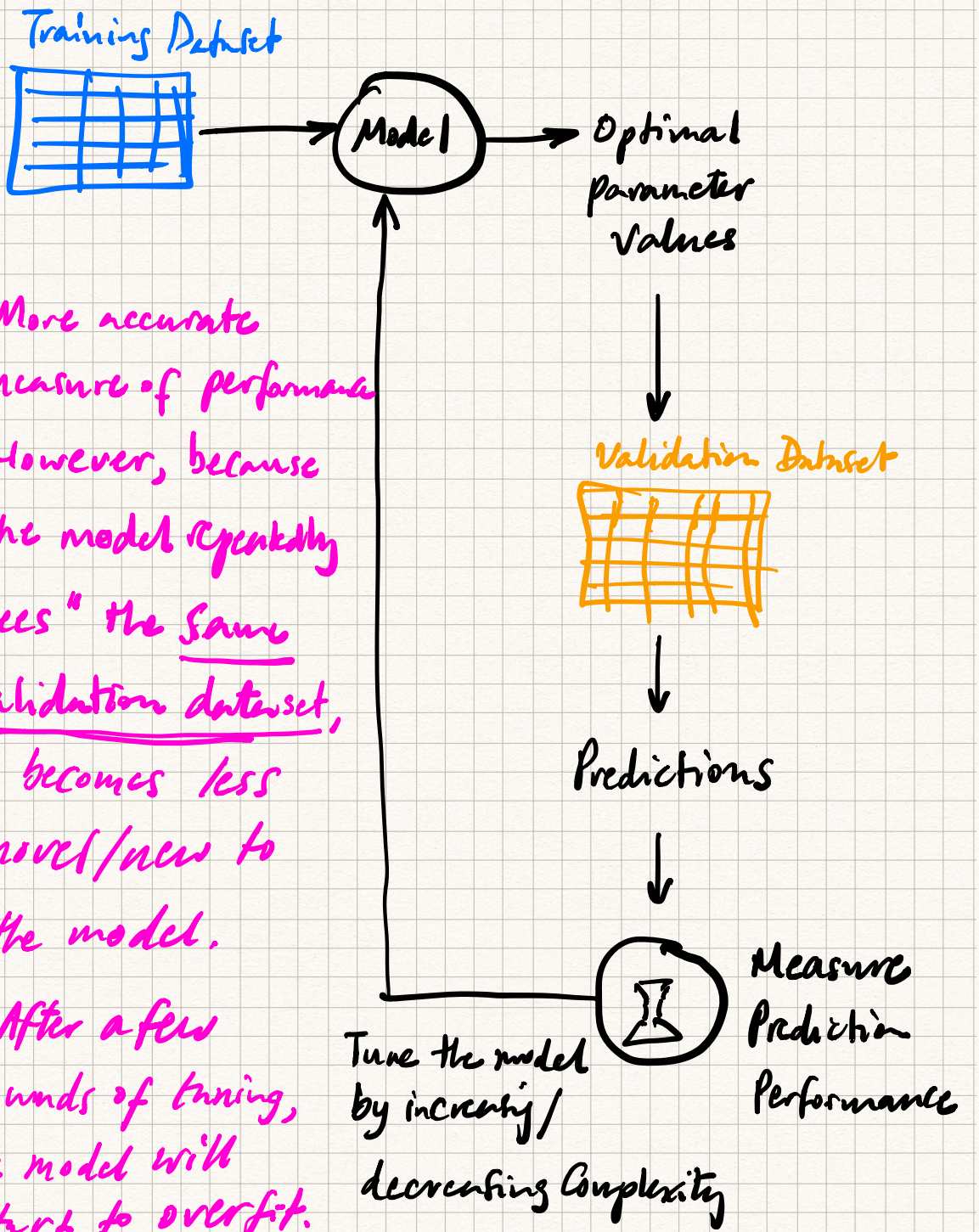

Original Dataset
Training Dataset
Validation Dataset
Test Dataset

Finding the Best Model - Scenario 1



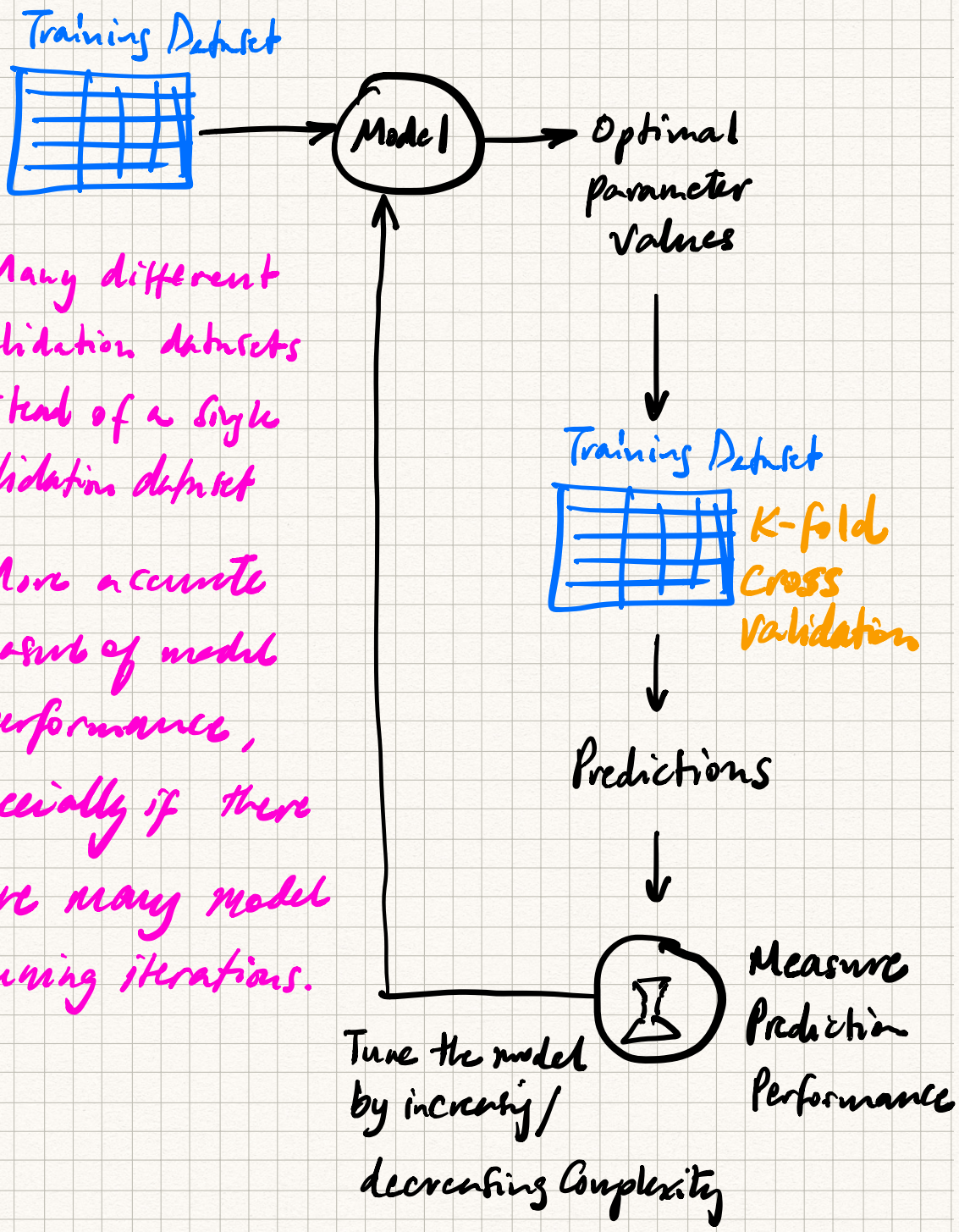
- Overestimates Prediction performance
- The more the model overfits the better the performance
- Performance on test dataset will be poor

Finding the Best Model – Scenario 2



- More accurate measure of performance
- However, because the model repeatedly "sees" the same validation dataset, it becomes less novel/new to the model.
- After a few rounds of tuning, the model will start to overfit.

Finding the Best Model - Scenario 3



- Many different validation datasets instead of a single validation dataset
- More accurate measure of model performance, especially if there are many model tuning iterations.

Once the best model is found using k-fold cross validation on the training dataset, measure the performance of the model on the test dataset as a sanity check.

Typically, performance on the test dataset will be worse than the performance achieved through k-fold cross validation on the training dataset.

If performance on the test dataset differs a great deal from the performance on the training dataset,

Something is wrong.

Usually, it means that the test dataset is not representative - it differs in some way(s) from the training dataset.

Even when the performance of the model(s) on the training and test datasets are in accord, the model may start to predict less well as more and more predictions are made.

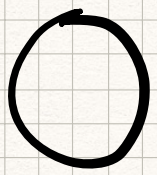
This means the model needs retraining.

Finally, there is a way to keep data "fresh" without doing k-fold cross validation.

The trick is to build multiple versions of the same model.

E.g.

Linear Regression with Lasso regularization



Linear Regression w/ Ridge regularization

Each version sees the training dataset just once and sees the test dataset just once.