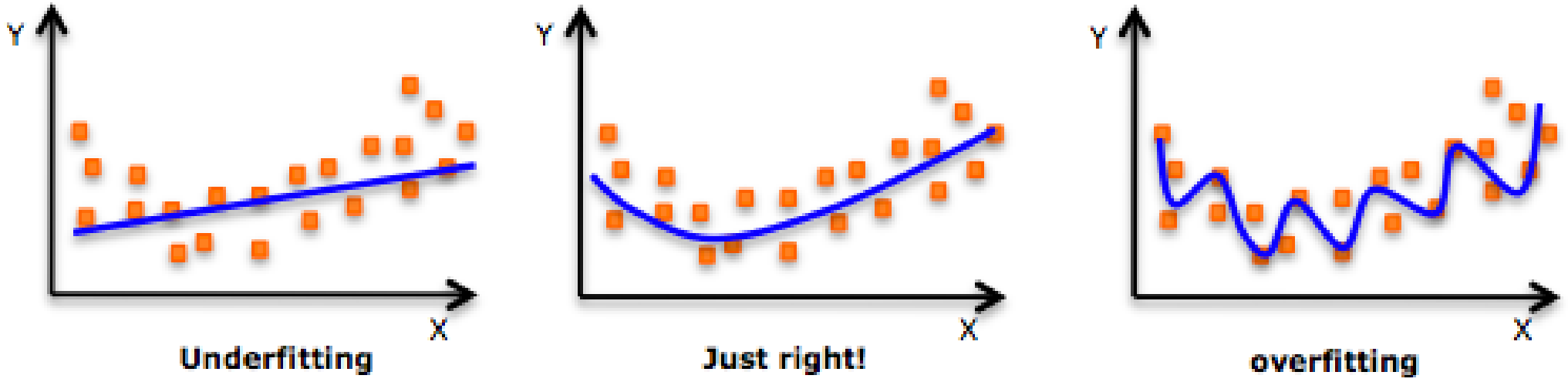
3.03 Train/Test Split + Cross Validation

What is Underfitting/Overfitting a Model?

- **Underfitting:** Model doesn't fit training data and is not generalizable to other data sets
- **Overfitting:** Model will be very accurate on training data but is not generalizable to other data sets

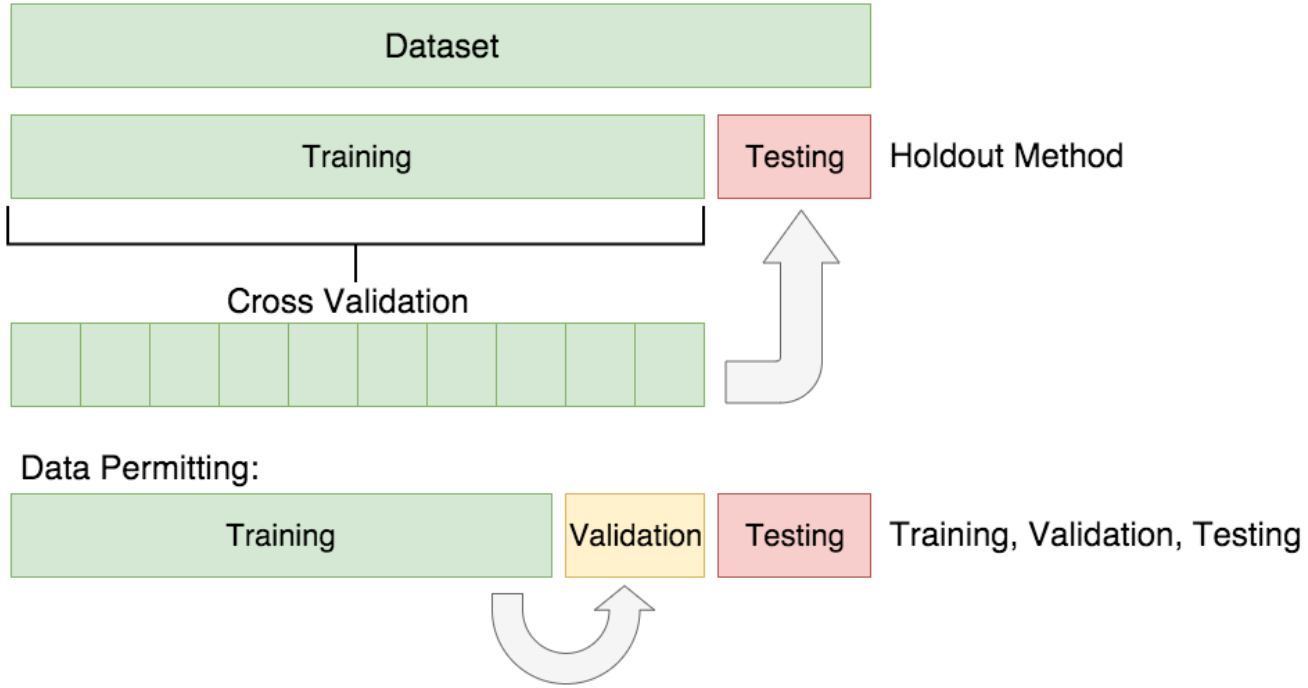


Splitting our data:
THE HOLDOUT METHOD: Train/Test Split

- Training Set
  - Used to train the model
- Testing Set
  - Used to test model performance on unseen data
- Advantages
  - Fast! Simple! Computationally inexpensive!
- Disadvantages
  - Eliminating data! Imperfect splits!

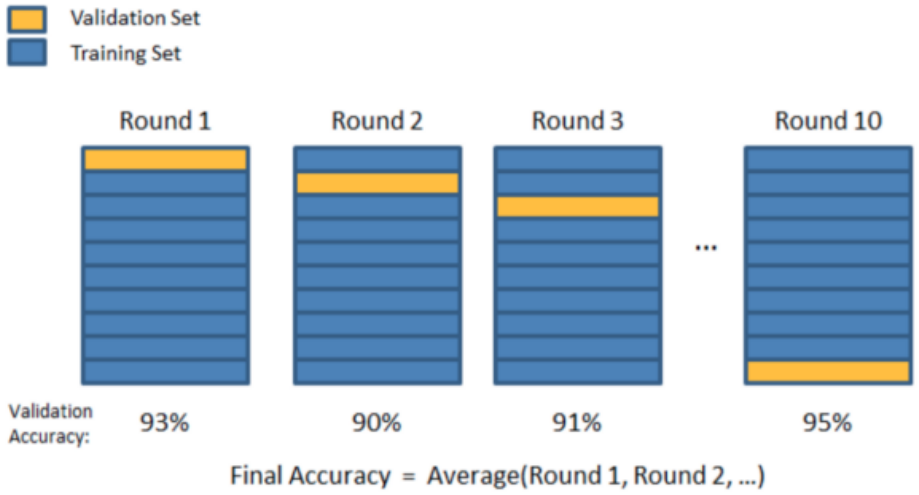
Splitting our data: Cross Validation

- Overcoming Train/Test Split limitation i.e. what if split is not random?
- Train Data is used for cross validation and resultant model is then applied on test data set(s)



K-Folds Cross Validation

- We split our **train data** into *k* different subsets (or folds).
- We use k-1 subsets to train our data and leave the last subset (or the last fold) as test data.
- We then average the model against each of the folds and then finalize our model. After that we test it against the test set.



How many folds should we choose?

- **With a large number of folds:**
  - Error due to bias is low
  - Variance is quite high
  - Computationally expensive

- **With a low number of folds:**
  - Error due to variance is low
  - The error due to bias will be large
  - Computationally cheaper

- **Thus…**
  - For large datasets, k=3/5 is typically ok

Three-way data splits

If model selection and performance are to be computed simultaneously, three disjoint data sets are best.

- **Training set:** a set of example used for learning
- **Validation set:** a set of examples used to further optimise / tune model
- **Testing set:** a set of examples used **ONLY** to assess the performance of the fully-trained model

Validation and testing must be separate data sets. Once you have the final model set, you cannot do any additional tuning after testing.

Cross-Validation Procedure

1. Divide data into training, validation, testing sets

2. Select model and training parameters (e.g., k)

3. Train the model using the training set

4. Evaluate the model using the training set

5. Repeat 2-4 selecting different models and tuning parameters

6. Select the best performing model

7. Assess the best model with the final testing set