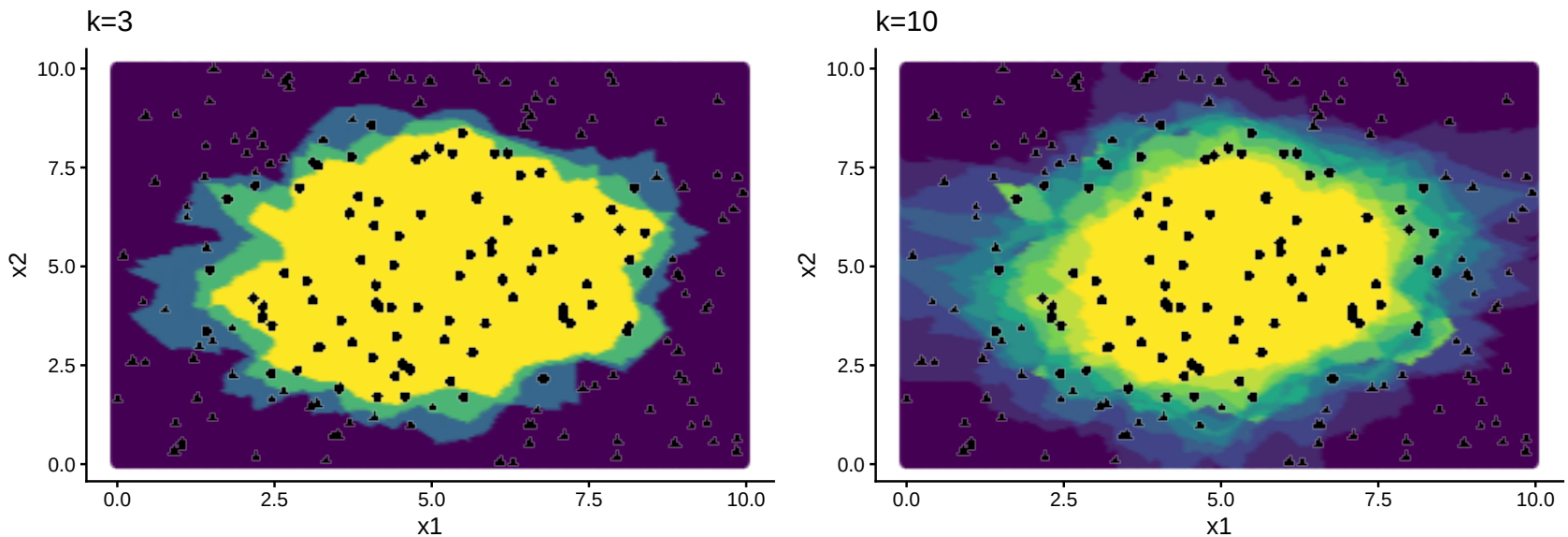
Cross-validation

Ryan Miller

Grinnell College
Statistics

Introduction

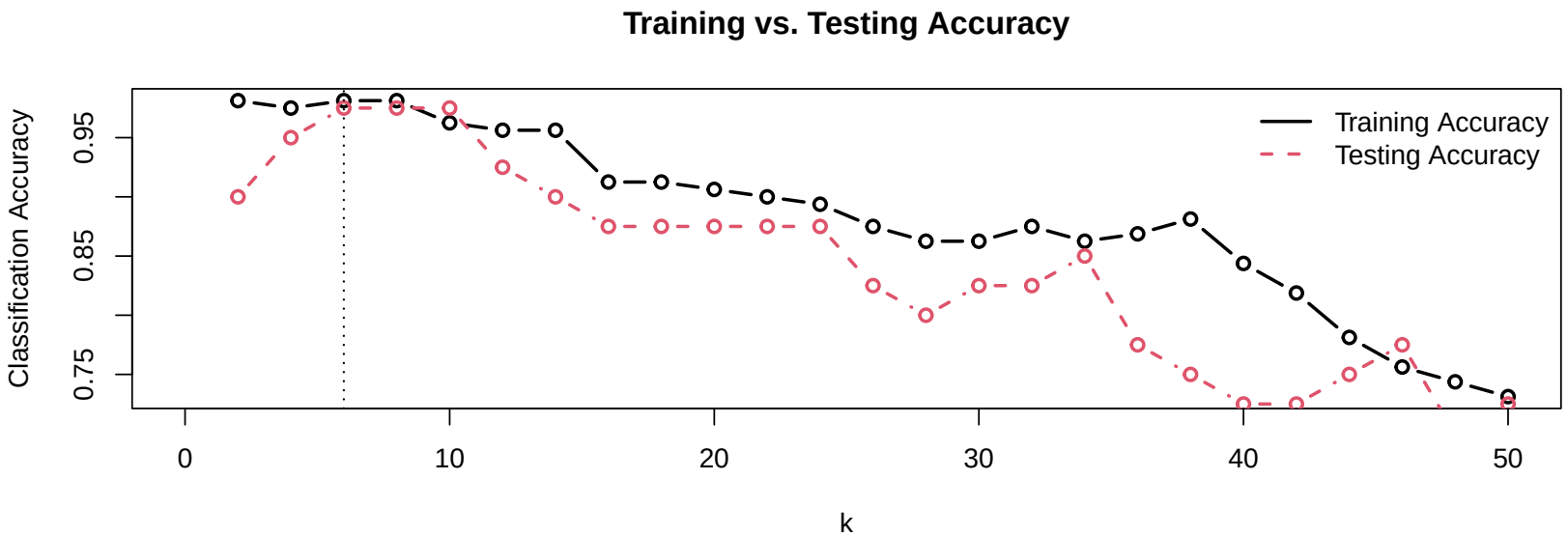
So far, we've been working with toy data involving the classification of healthy vs. unhealthy samples:



Which is the better value of k?

Introduction (cont.)

Below are the training and test set performances with these data for various KNN models:



Should we choose the k with the highest test set accuracy?

Test Data vs. Model Selection

- Choosing k using the test set produces performance estimates that are too optimistic
- In our example, $k = 6$ was "cherry-picked"
  - it was the value that got lucky on this specific test set from a range of k that might perform similarly
- So, how might we approach selecting k without overusing the test set?

Single Validation

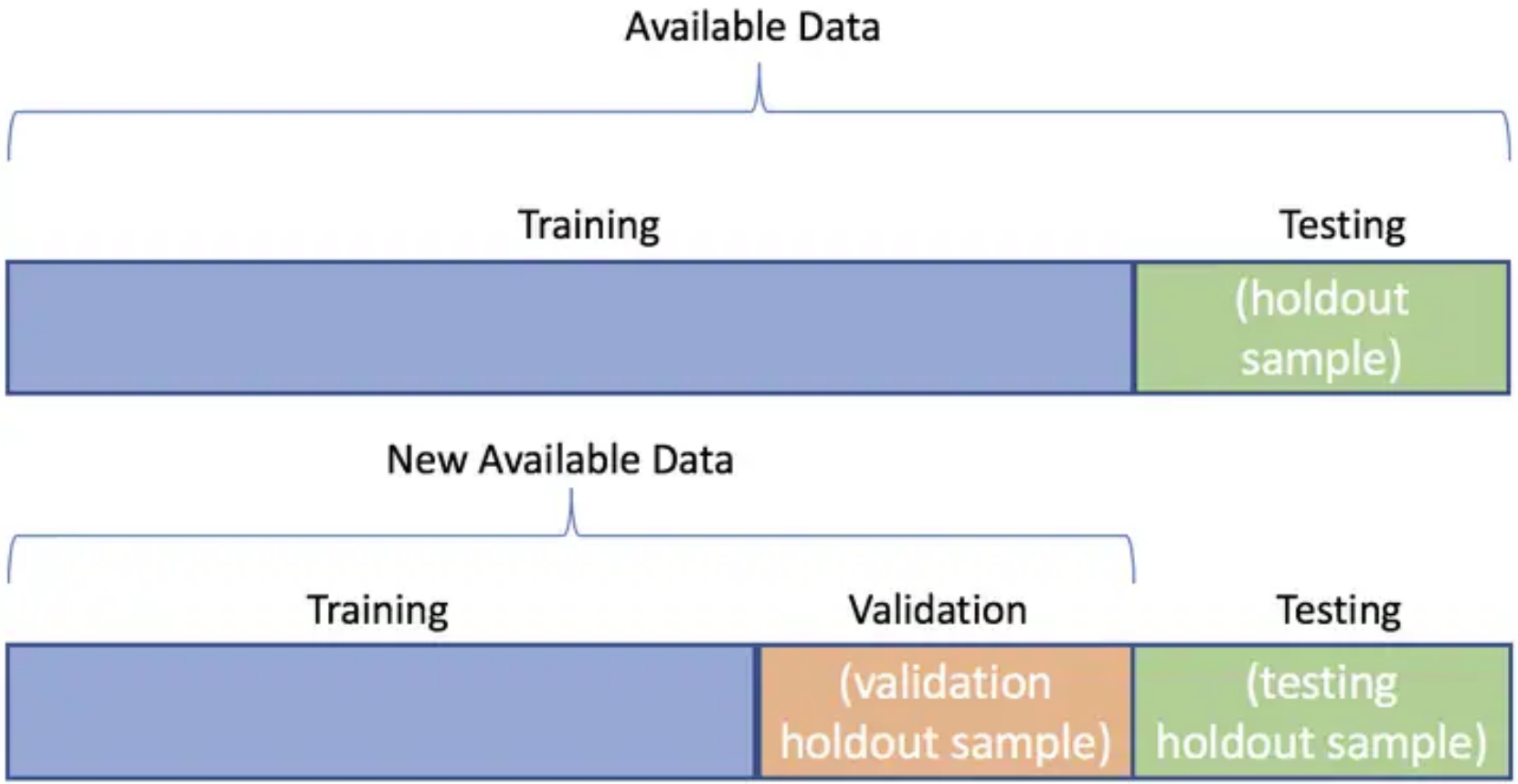


image credit: https://algotrading101.com/learn/train-test-split/

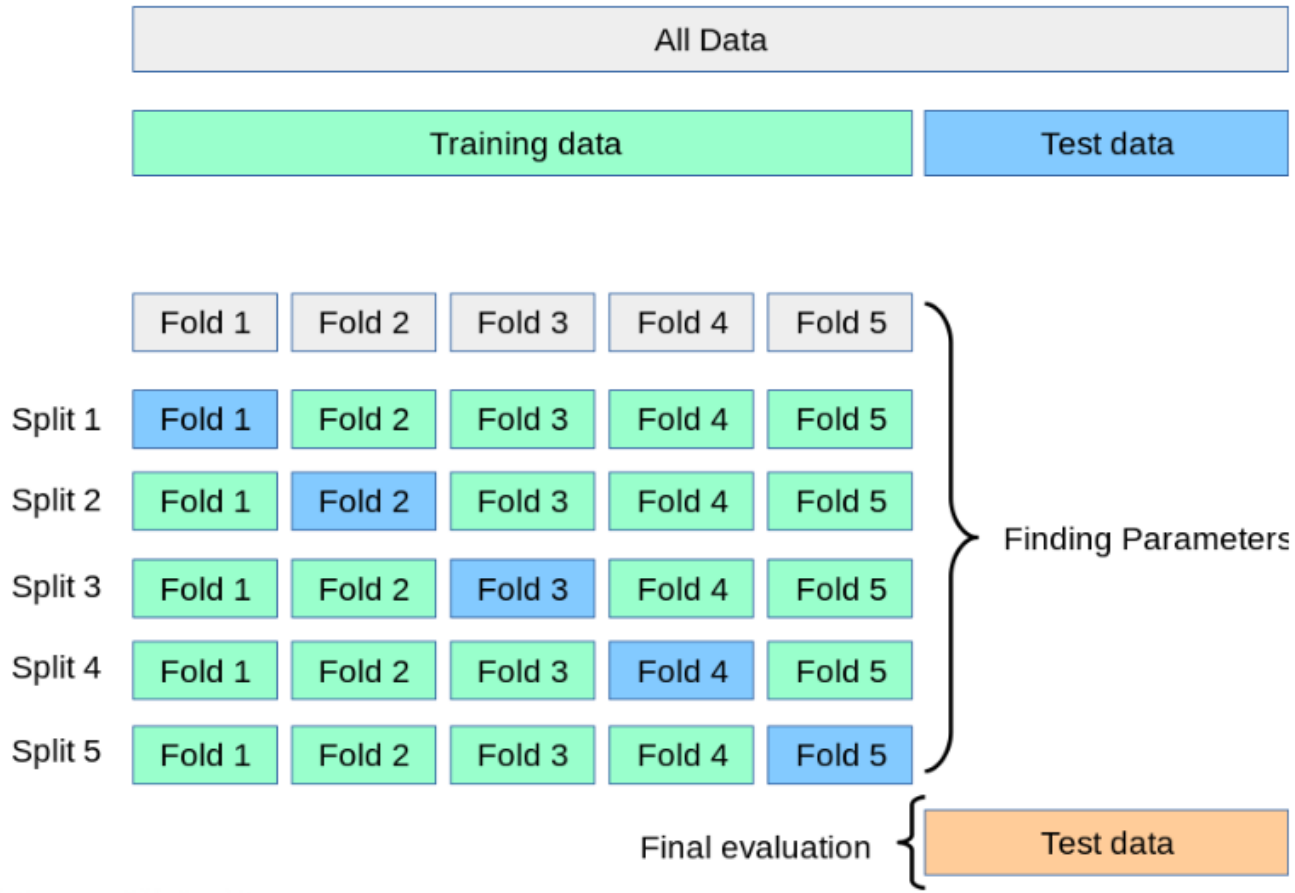
Single Validation (cont.)

Carving off a separate validation set from the training data has a few disadvantages:

1. We have a similar "cherry-picking" problem, which might lead to us picking "lucky" hyperparameter configurations rather than optimal ones
2. If we're using 10-25% of our data for validation and 10-25% for testing, we might not have much data to train on

Cross-validation

Cross-validation provides more robust estimates of model performance by repeating the training-validation process using different "folds" of data. 5-fold cross-validation is illustrated below:



Cross-validation (cont.)

Pseudocode of 5-fold cross-validation:

```
## Assign n obs into k folds
fold_id = sample(1:k, size = n)

## Loop through each fold:
for i in 1:k
   train_X = samples[fold_id != i]
   train_y = labels[fold_id != i]
   eval_X = samples[fold_id == i]
   eval_y = labels[fold_id == i]

   model.fit(train_X, train_y)
   pred[fold_id == i] = model.predict(eval_X, eval_y)

## Calculate performance
score(pred)
```

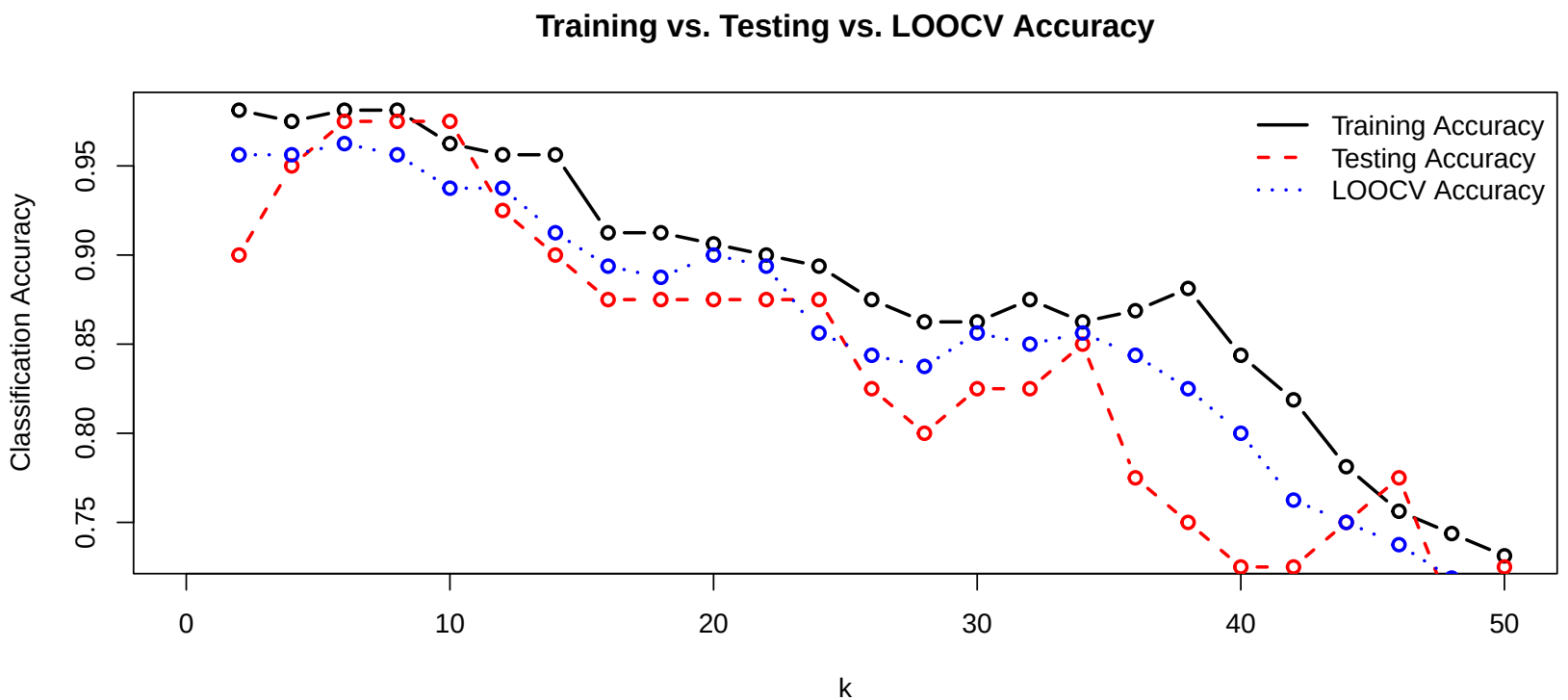
Cross-validation (cont.)

Cross-validation (CV) is a general term describing methods of repeated data-splitting used to evaluate the performance of a model on data that was not used in the training process:

1. *k*-**fold cross-validation** - the training data are partitioned into *k* equally sized folds. This approach is *non-exhaustive*, meaning we won't get the same answer every time.
2. **Leave-one-out cross-validation (LOOCV)** - training and validation are repeated *n* times, holding out each individual sample as its own validation set. This approach is *exhaustive*, but can be computationally expensive.

Example - LOOCV

Cross-validation helps us *avoid overfitting* by producing reasonable estimates of performance without using the test set:



k-fold or LOOCV?

- Compared to *k*-fold CV, LOOCV is a *higher variance procedure*
  - Each training set uses almost entirely the same data
  - Thus, performance estimates are sensitive to the specific sample, and will vary more from sample to sample
- *k*-fold cross-validation offers *better generalization*, but can be unfeasible for very small sample sizes

How many folds?

- Too many folds leads to higher variance estimates of performance, as the training sets become more similar
  - Too few folds can lead to biased estimates due lack of training data ($k = 3$ will only use $2/3$ of the training set for model training per iteration)
  - 5-10 folds generally provides a good middle ground
- Additionally, since a model must be trained for each fold, compute time scales approximately linearly with k
  - Thus, $k = 10$ is recommended for smaller datasets and computationally inexpensive models
  - $k = 5$ is recommended for larger datasets and/or computationally demanding models

Grid Search

Cross-validation is generally paired with methods for evaluating hyperparameter configurations, with **grid search** being one of the most common:

k	Distance	Weight
5	euclidean	uniform
10	euclidean	uniform
5	manhattan	uniform
10	manhattan	uniform
5	euclidean	distance
10	euclidean	distance
5	manhattan	distance
10	manhattan	distance

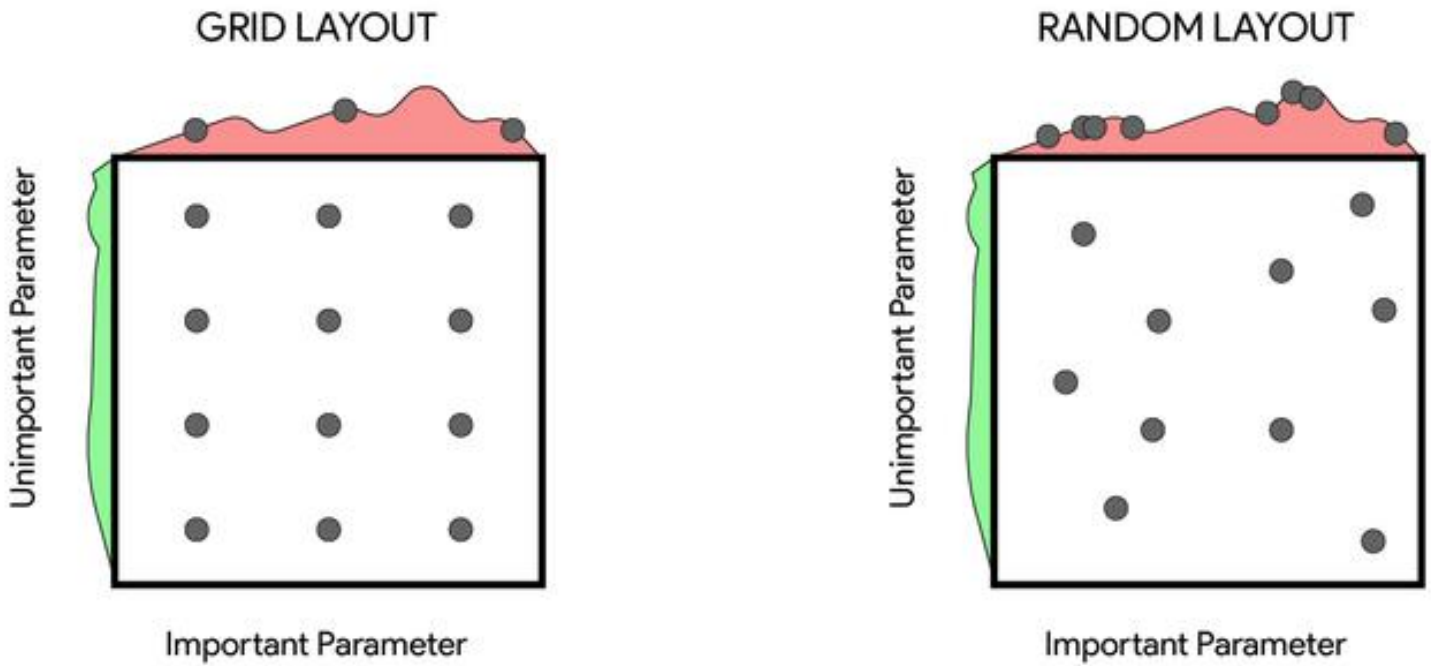
Python will allow us to evaluate each row of this grid via cross-validation *using the same fold assignments*.

Randomized Search

- Grid search can be computationally expensive, especially when you'd like to explore a broad range of values for several different hyperparameters
- **Randomized search** is an alternative method that allows you specify distributions to be randomly sampled from for each hyperparameter
  - For KNN, you might sample k from a Poisson distribution with $\mu = \sqrt{n}$

Grid Search vs. Randomized Search

An underappreciated fact is that the uniform spacing in grid searches can sometimes prevent us from finding optimal values for important parameters:



What to Know for Our Next Quiz

- Understand the basic steps behind k-fold cross-validation, including:
  - Dividing the data into k equally-sized parts
  - Training a model using the samples in k-1 of these parts and using it to make predictions on the left out fold
  - Repeating until all folds have been left out exactly once
- Understand cross-validation reduces the infleunce of a single "lucky" data split
- Understand how k-fold cross-validation can be combined with approaches like grid search and randomized search to make informed choices of hyperparameters and data pre-processing steps