Model Selection

Part A: Model Selection

Pavlos Protopapas

Institute for Applied Computational Science
Harvard

Model Selection

Model selection is the application of a principled method to determine the complexity of the model, e.g. choosing a subset of predictors, choosing the degree of the polynomial model etc.

A strong motivation for performing model selection is to avoid **overfitting**, which we saw can happen when:

- there are too many predictors:
    - the feature space has high dimensionality
    - the polynomial degree is too high
    - too many cross terms are considered
- the coefficients values are too **extreme (we have not seen this yet)**

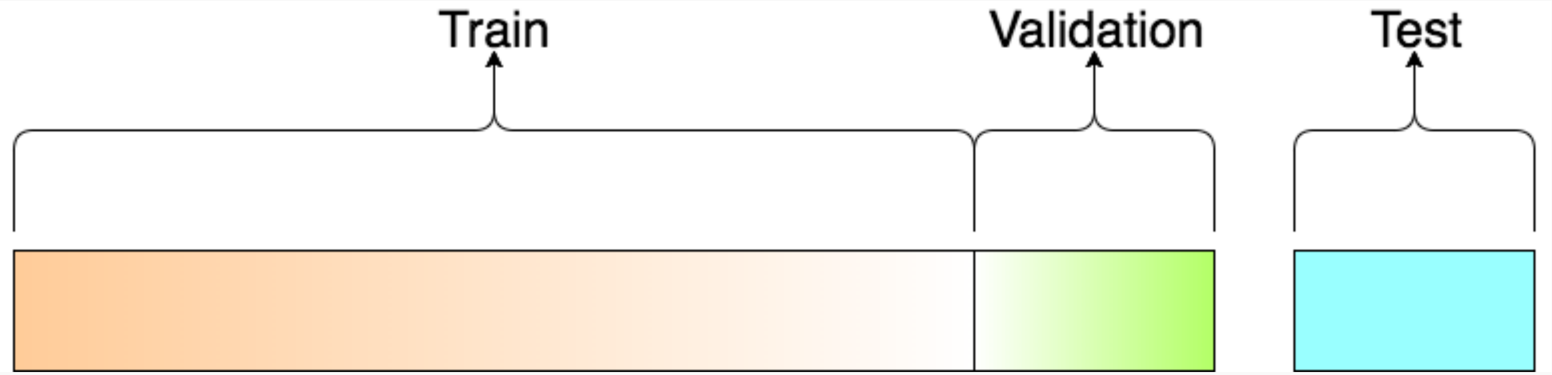
Generalization Error

We know to evaluate the model on both train and test data, because models that do well on training data may do poorly on new data (overfitting).

The ability of models to do well on new data is called generalization.

The goal of model selection is to choose the model that generalizes the best.

Train-Validation-Test



We use this to train a model

We use this to select model

We use this to report model performance

Model Selection

Question:

How many different models when considering J predictors (only linear terms) do we have?

Example: 3 predictors $(\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3})$

- Models with 0 predictor:

  M0:

- Models with 1 predictor:

  M1: X_1

  M2: X_2

  M3: X_3

- Models with 2 predictors:

  M4: $\{X_1, X_2\}$

  M5: $\{X_2, X_3\}$

  M6: $\{X_3, X_1\}$

- Models with 3 predictors:

  M7: $\{X_1, X_2, X_3\}$

2^J models

Stepwise Variable Selection and Validation

Selecting optimal subsets of predictors (including choosing the degree of polynomial models) through:

- stepwise variable selection - iteratively building an optimal subset of predictors by optimizing a fixed model evaluation metric each time.
- validation - selecting an optimal model by evaluating each model on validation set.

Stepwise Variable Selection: Forward method

In **forward selection**, we find an ‘optimal’ set of predictors by iterative building up our set.

1. Start with the empty set P0, construct the null modelM0.

2. For $k = 1, \dots, J$:

2.1 Let M_{k-1} be the model constructed from the best set of $k-1$ predictors, P_{k-1}.

2.2 Select the predictor X_{n_k}, not in P_{k-1}, so that the model constructed from $P_k = X_{n_k} \cup P_{k-1}$ optimizes a fixed metric (this can be p -value, F -stat; validation MSE, R^2, or AIC/BIC on training set).

2.3 Let M_k denote the model constructed from the optimal P_k.

3. Select the model M amongst $\{M_0, M_1, \dots, M_J\}$ that optimizes a fixed metric (this can be validation MSE, R^2; or AIC/BIS on training set)

Stepwise Variable Selection Computational Complexity

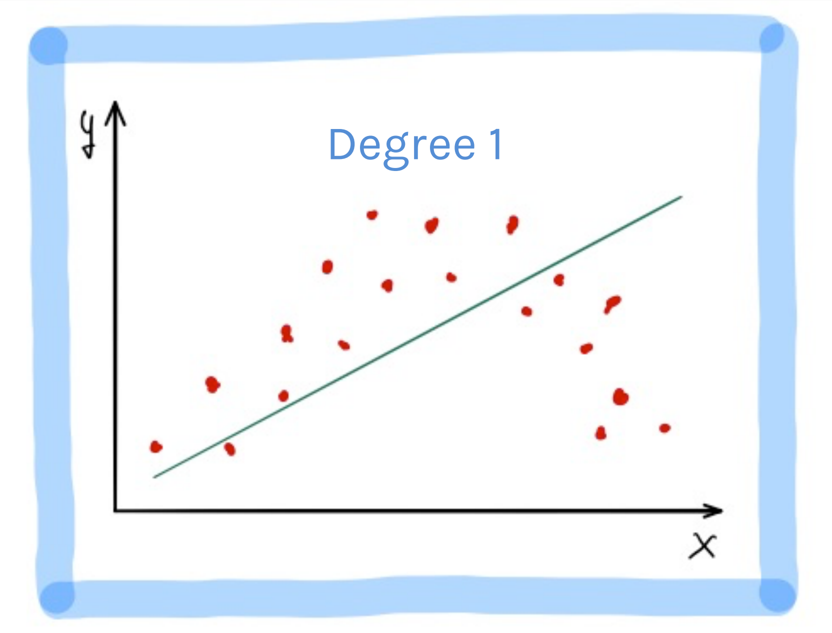
How many models did we evaluate?

- 1st step, ***J* Models**
- 2nd step, *J-1* **Models** (add 1 predictor out of *J-1* possible)
- 3rd step, *J-2* **Models** (add 1 predictor out of *J-2* possible)
- …

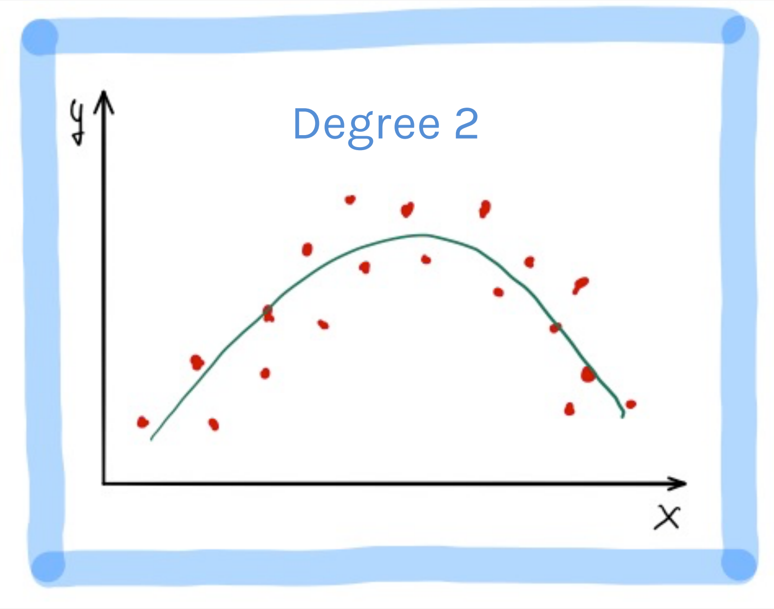
$$O(J^2) \ll 2^J \text{ for large } J$$

Choosing the degree of the polynomial model

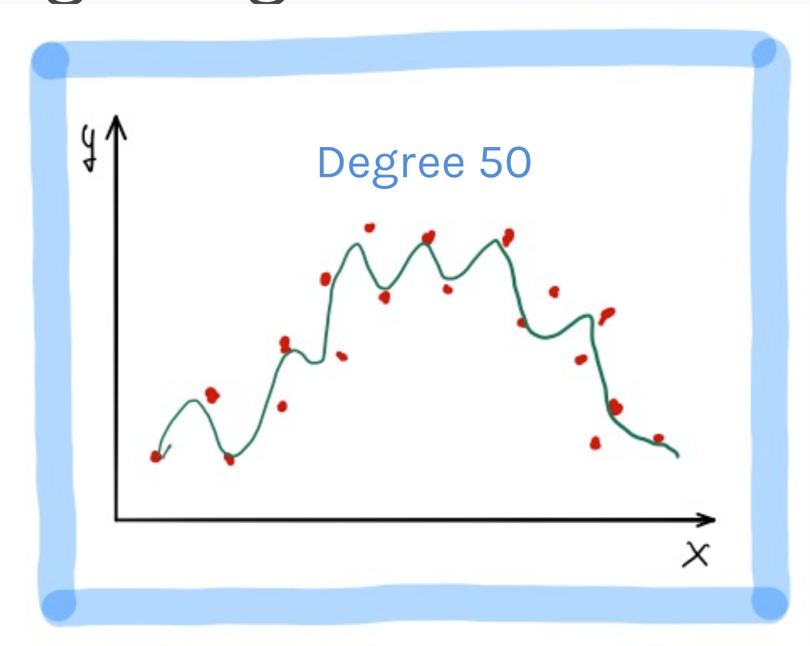
Fitting a polynomial model requires choosing a degree.



Underfitting: when the degree is too low, the model cannot fit the trend.

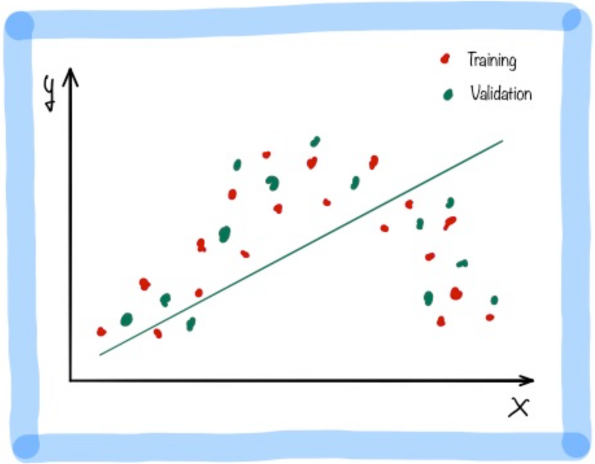


We want a model that fits the trend and ignores the noise.

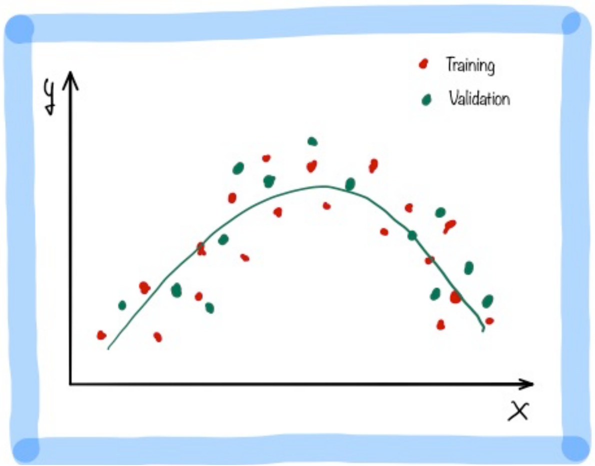


Overfitting: when the degree is too high, the model fits all the noisy data points.

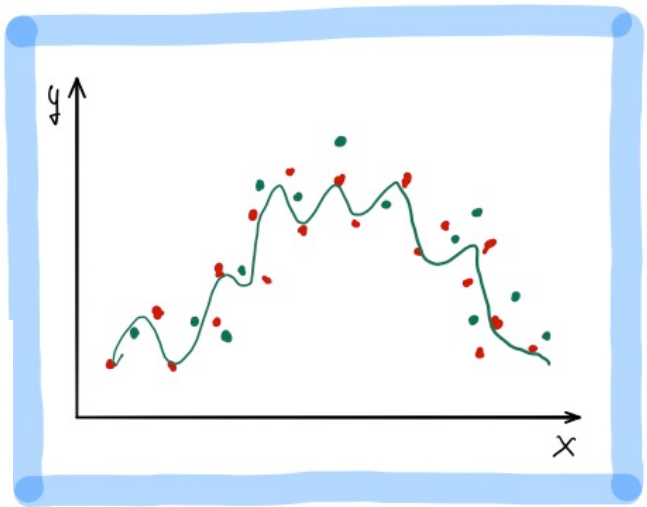
Underfitting: train and validation error is high.

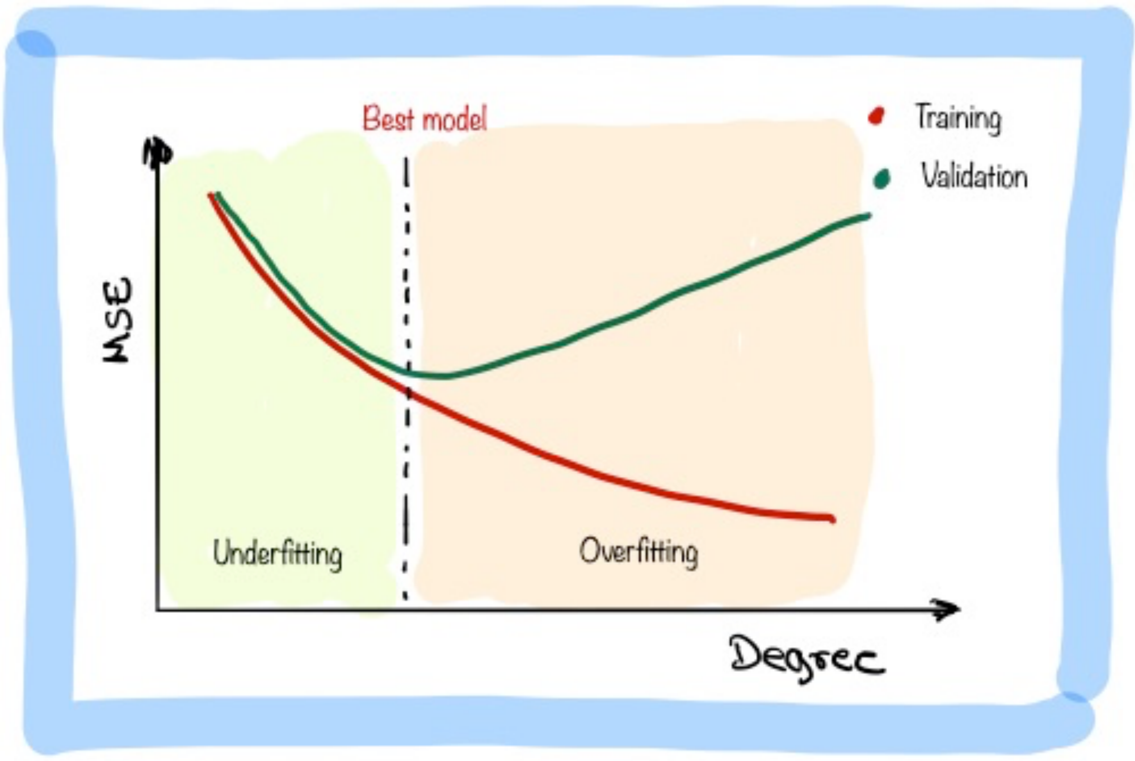


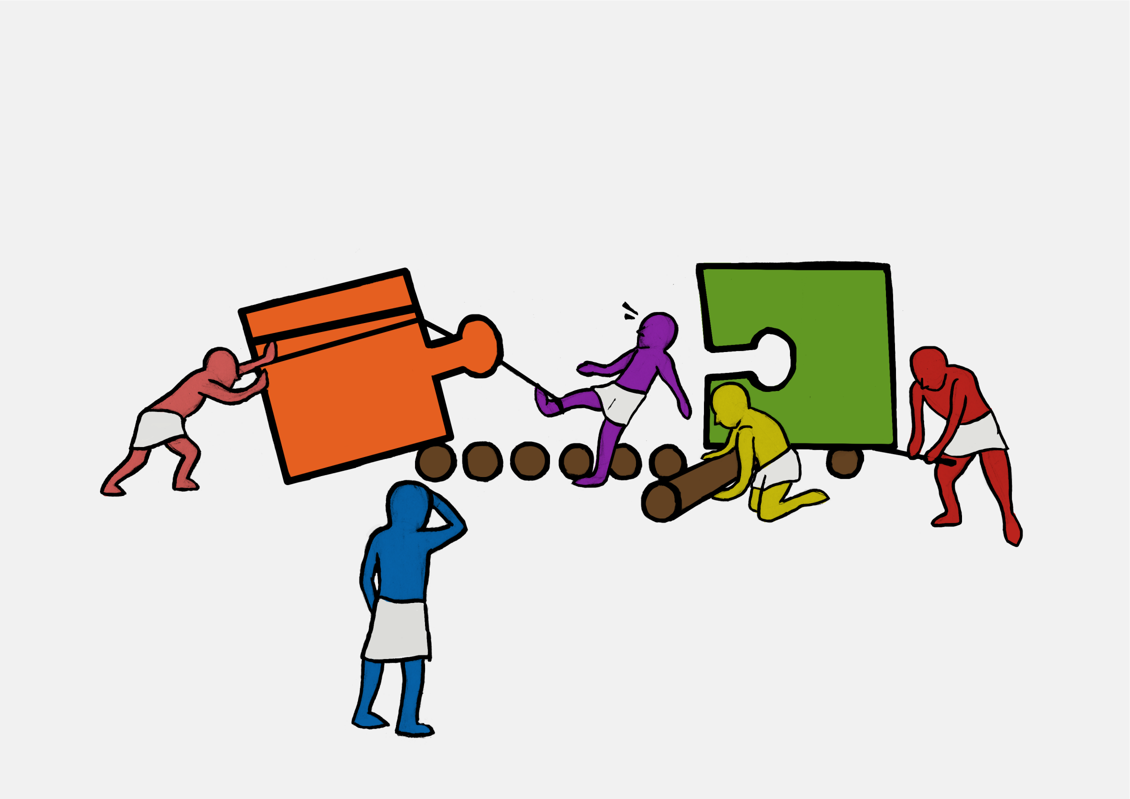
Best model: validation error is minimum.



Overfitting: train error is low, validation error is high.



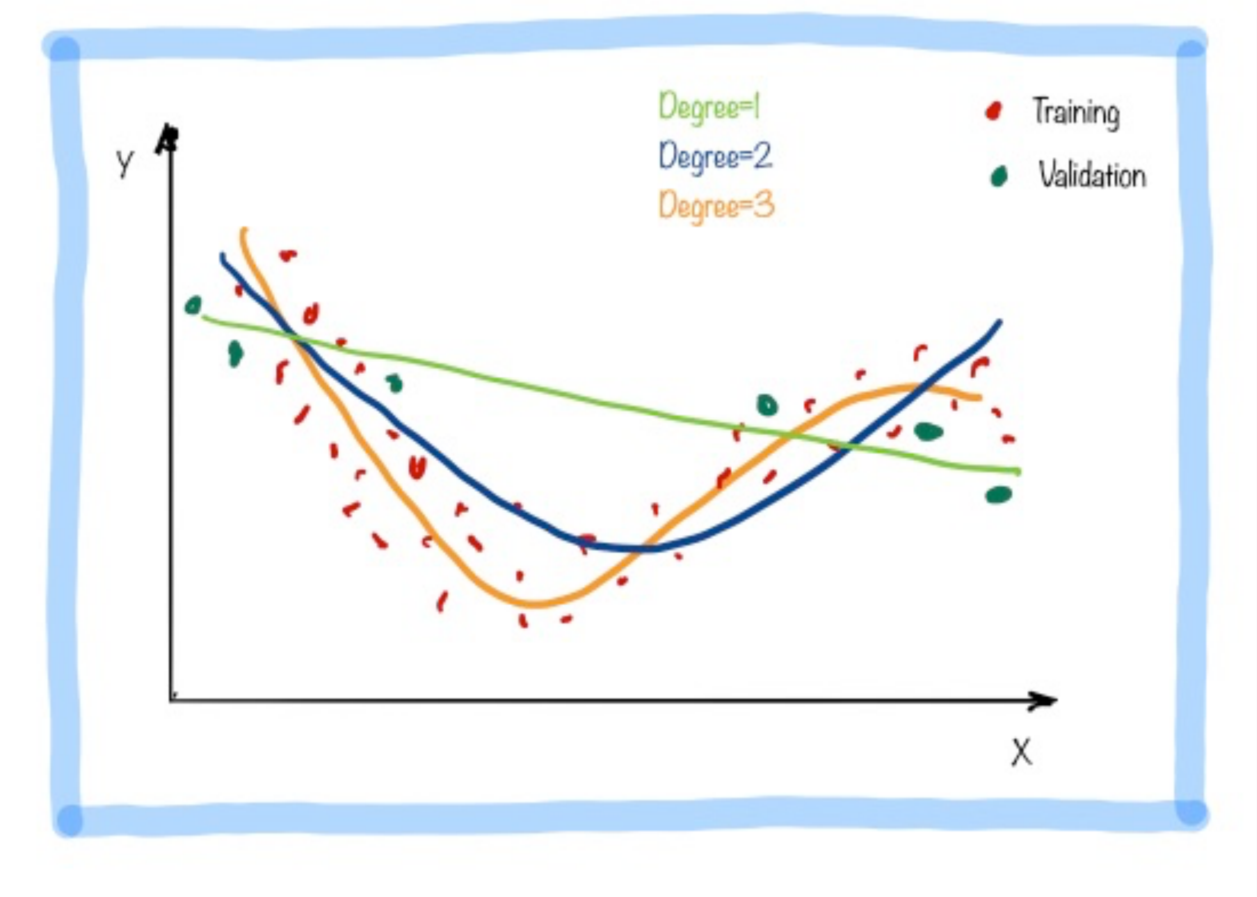




Exercise C.1

Cross Validation: Motivation

Using a single validation set to select amongst multiple models can be problematic - **there is the possibility of overfitting to the validation set**.



It is obvious that degree=3 is the correct model but the validation set by chance favors the linear model.

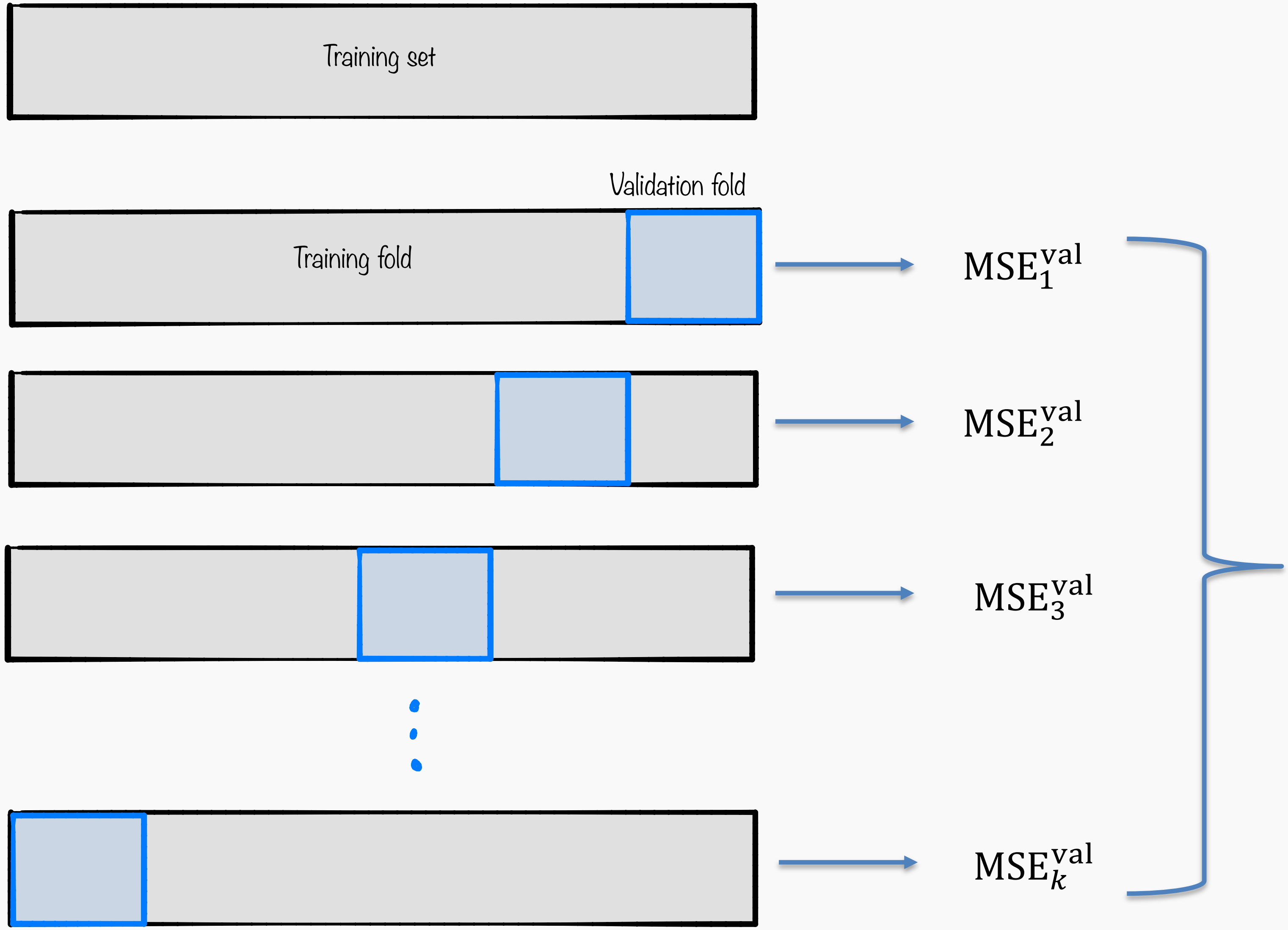
Cross Validation: Motivation

Using a single validation set to select amongst multiple models can be problematic - **there is the possibility of overfitting to the validation set**.

One solution to the problems raised by using a single validation set is to evaluate each model on **multiple** validation sets and average the validation performance.

One can randomly split the training set into training and validation multiple times **but** randomly creating these sets can create the scenario where important features of the data never appear in our random draws.

Cross Validation



$\mathrm{MSE}_1^{\mathrm{val}}$

$\mathrm{MSE}_2^{\mathrm{val}}$

$\mathrm{MSE}_3^{\mathrm{val}}$

$\mathrm{MSE}_k^{\mathrm{val}}$

$$\mathrm{MSE}^{\mathrm{val}} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i^{\mathrm{val}}$$

K-Fold Cross Validation

Given a data set $\{X_1, \ldots, X_n\}$, where each $\{X_1, \ldots, X_n\}$ contains *J* features.

To ensure that every observation in the dataset is included in at least one training set and at least one validation set we use the **K-fold validation**:

- split the data into *K* uniformly sized chunks, $\{C_1, \ldots, C_K\}$
- we create *K* number of training/validation splits, using one of the *K* chunks for validation and the rest for training.

We fit the model on each training set, denoted $\hat{f}_{C_{-i}}$, and evaluate it on the corresponding validation set, $\hat{f}_{C_{-i}}(C_i)$. The ***cross validation is the performance*** of the model averaged across all validation sets:

$$CV(\text{Model}) = \frac{1}{K}\sum_{i=1}^{K} L(\hat{f}_{C_{-i}}(C_i))$$

where *L* is a loss function.

Leave-One-Out

Or using the ***leave one out*** method:

- validation set: $\{X_i\}$
- training set: $X_{-i} = \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n\}$

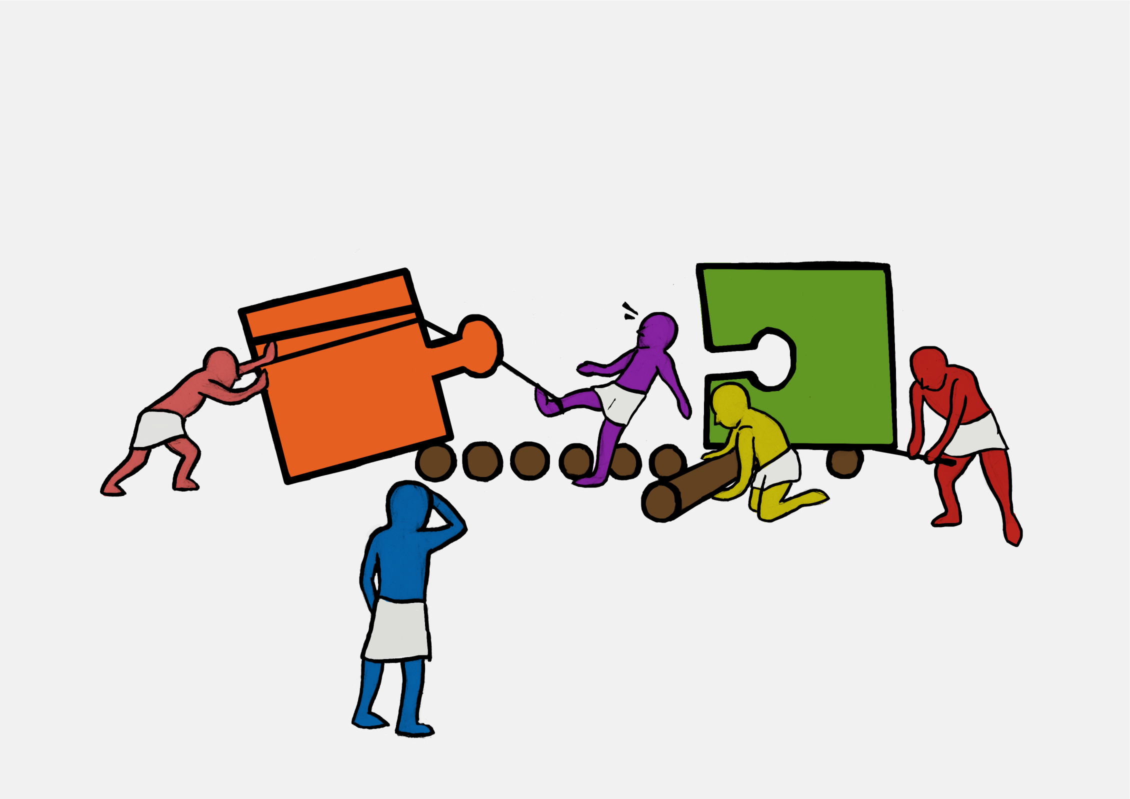
for $i = 1, \ldots, n$:

We fit the model on each training set, denoted $\hat{f}_{X_{-i}}$, and evaluate it on the corresponding validation set, $\hat{f}_{X_{-i}}(X_i)$.

The ***cross validation score*** is the performance of the model averaged across all validation sets:

$$CV(\text{Model}) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{f}_{X_{-i}}(X_i))$$

where L is a loss function.



Exercise C.2