INT 303 BIG DATA ANALYTICS

Lecture9: Model Selection and Cross Validation

Pengfei FAN
pengfei.fan@xjtlu.edu.cn

Xi'an Jiaotong-Liverpool University
西交利物浦大學

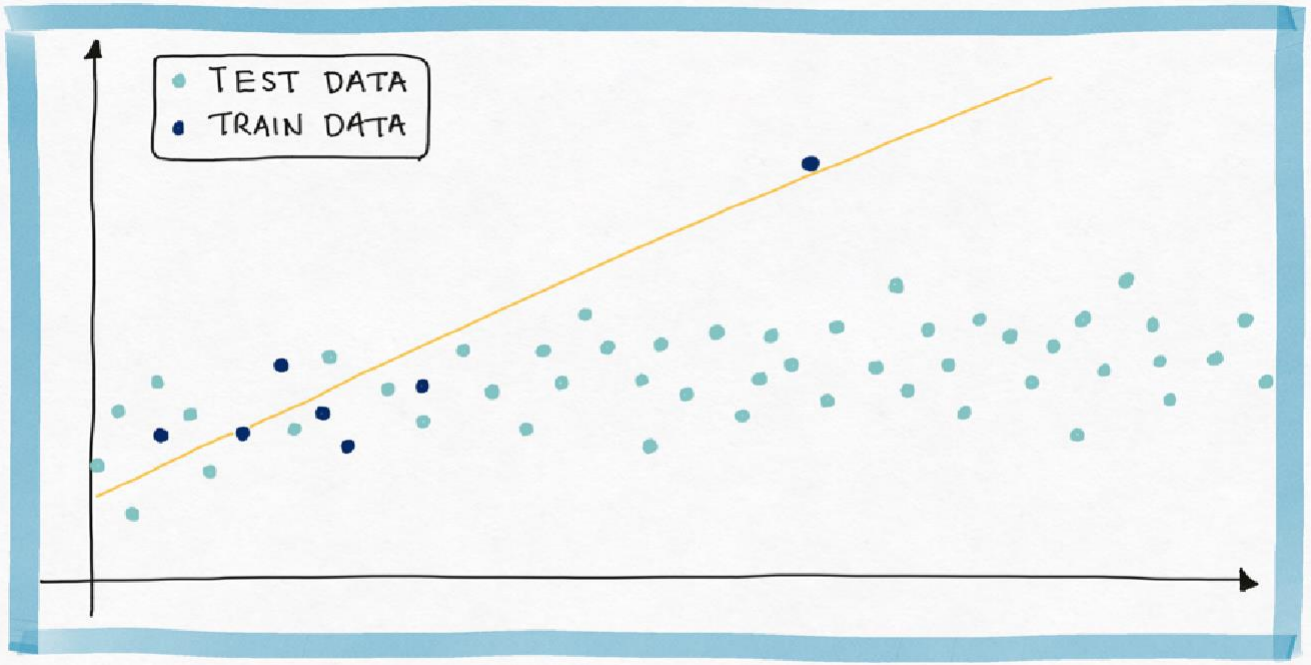
Outline

- Model Selection
  - Using Validation
  - Using Cross Validation

Model Selection

Evaluation: Training Error & Test Error

We need to evaluate the fitted model on new data, data that the model did not train on, the test data.



The training MSE here is 2.0 where the test MSE is 12.3.

The training data contains a strange point – an outlier – which confuses the model.

Fitting to meaningless patterns in the training is called **overfitting.**

Overfitting: Another Motivation for Model Selection

Finding subsets of significant predictors is an important for model interpretation. But there is another strong reason to model using the smaller set of significant predictors: to avoid overfitting.

Definition

Overfitting is the phenomenon where the model is unnecessarily complex, in the sense that portions of the model captures the random noise in the observation, rather than the relationship between predictor(s) and response.

Overfitting causes the model to lose predictive power on new data.

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.

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**

Model Selection

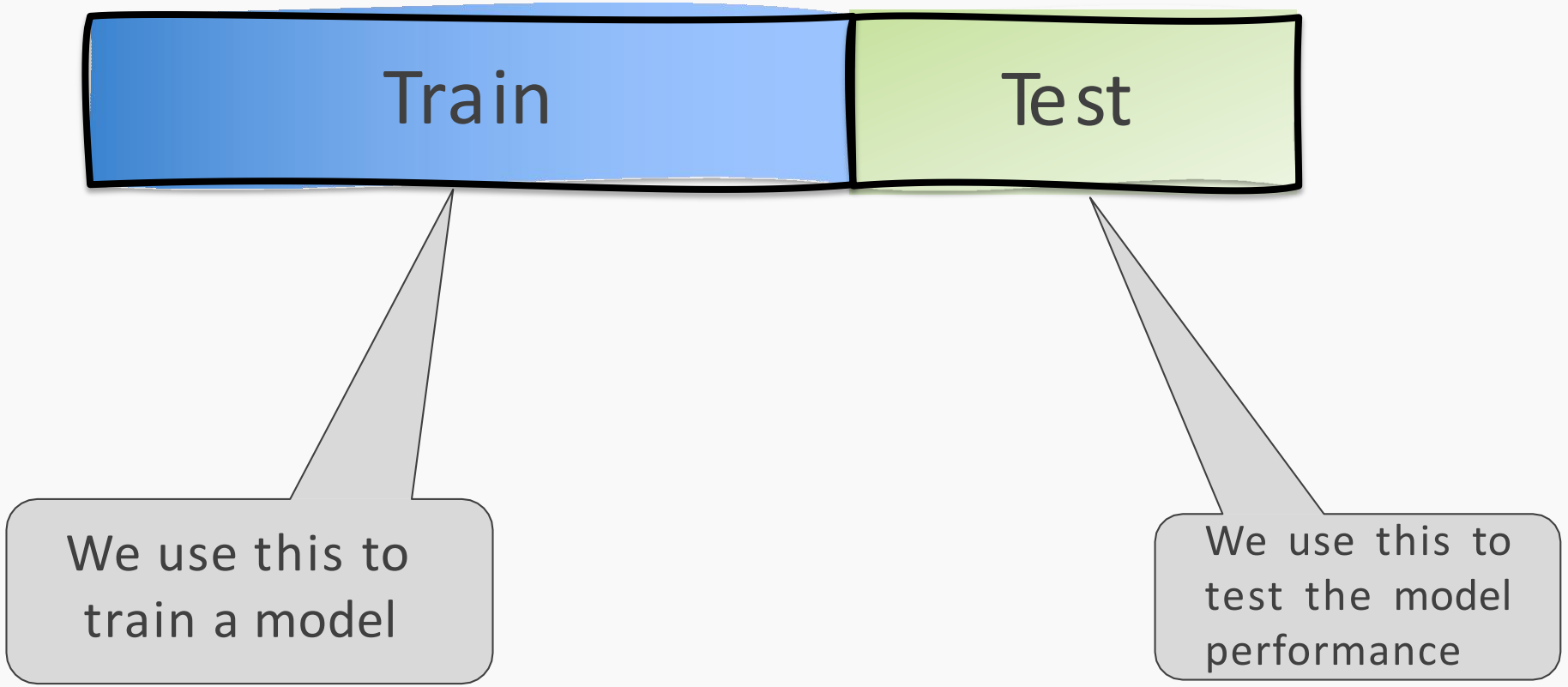
Model selection typically consists of the following steps:

1. split the training set into two subsets: training and ***validation***
2. multiple models (e.g. polynomial models with different degrees) are fitted on the training set; each model is evaluated on the validation set
3. the model with the best validation performance is selected
4. the selected model is evaluated one last time on the testing set

Train-Test split

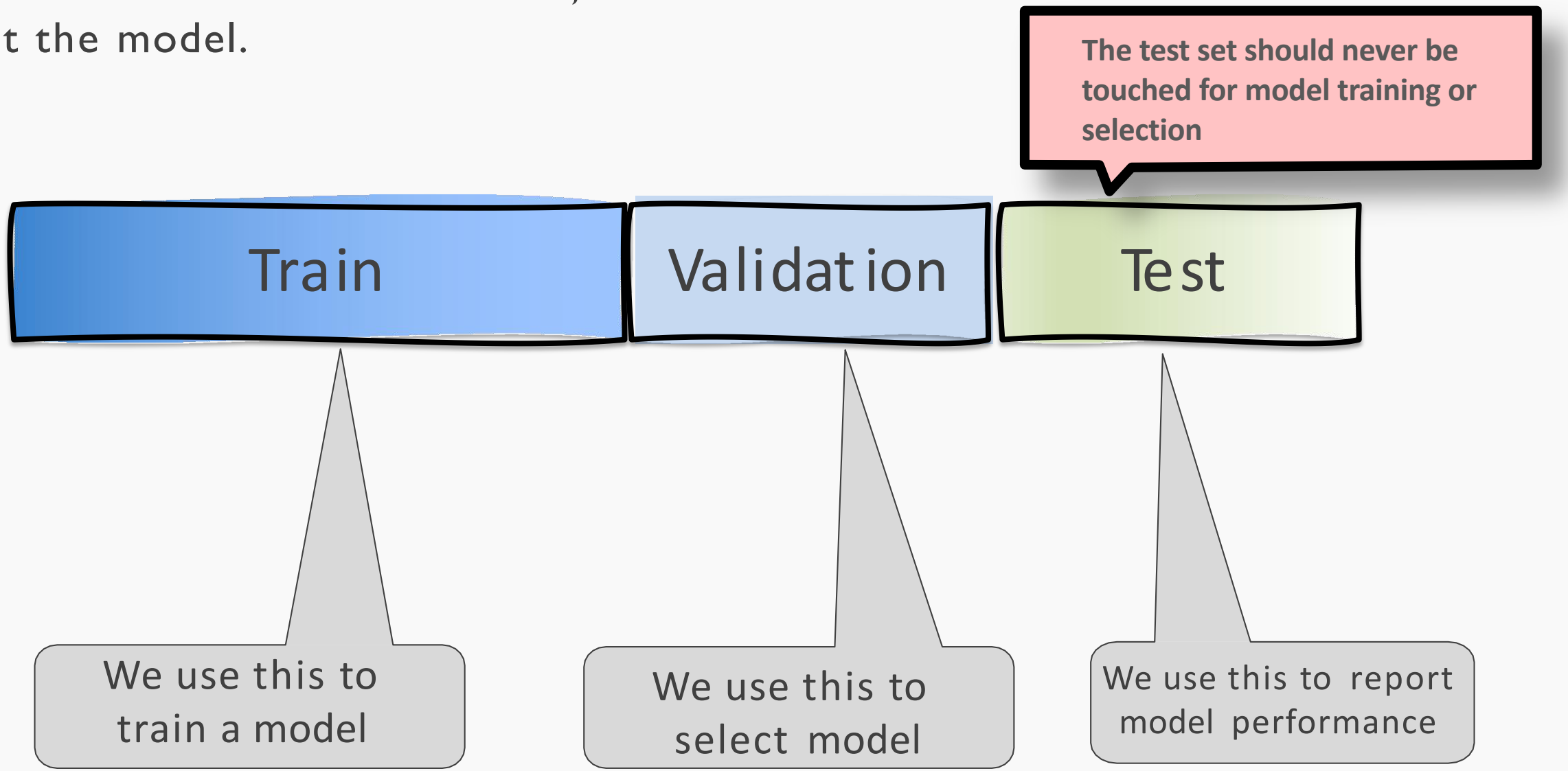
How do we select a model?

So far, we have been using train/test splits



Train-Validation-Test

We introduce a different sub-set, which we called validation and we use it to select the model.



Model Selection

Ways of model selection:

- Exhaustive search
- Greedy algorithms
- Fine tuning hyper-parameters
- Regularization

Model Selection

Ways of model selection:

- **Exhaustive search**
- Greedy algorithms
- Fine tuning hyper-parameters
- Regularization

Model Selection: How many models?

Question:

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

Example: 3 predictors (X_1, X_2, 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

Model Selection

Ways of model selection:

- Exhaustive search
- **Greedy algorithms**
- Fine tuning hyper-parameters
- Regularization

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.
- 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 model M0.

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

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_{nk}, not in P_{k-1}, so that the model constructed from $P_k = X_{nk} \cup P_{k-1}$ 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 $\{M_0, M_1, \ldots, M_J\}$ 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)
- …

$O(J^2) << 2^J$ for large J

Model Selection

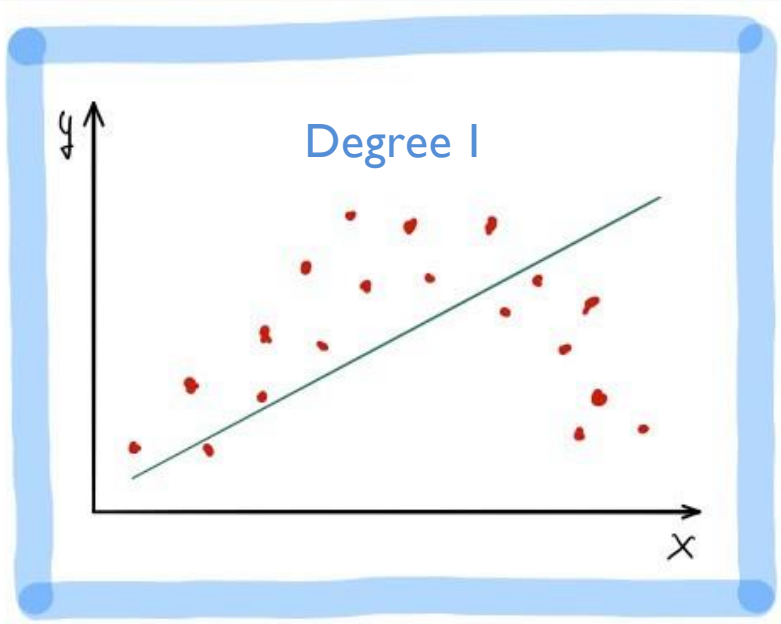
Ways of modelselection:

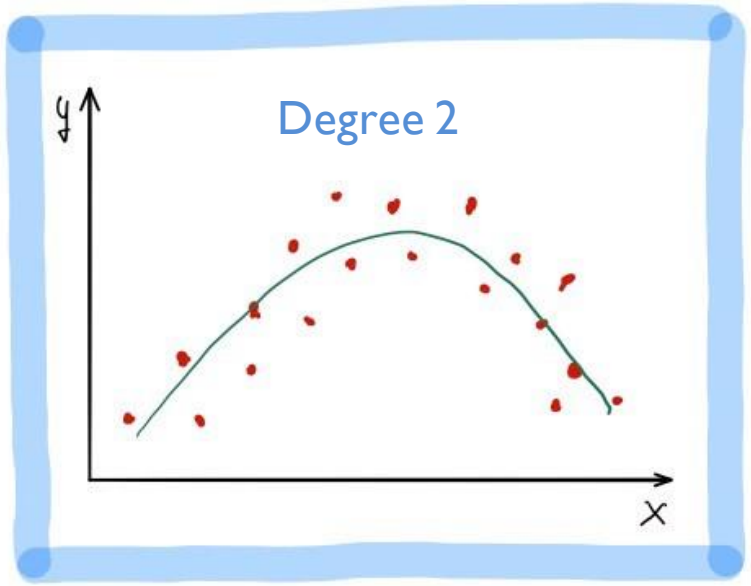
- Exhaustive search
- Greedy algorithms
- **Fine tuning hyper-parameters**
- Regularization

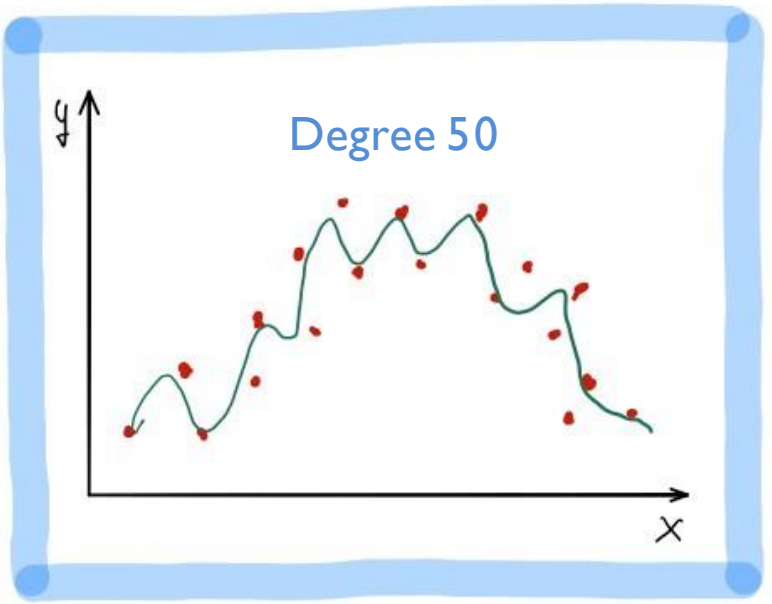
Choosing the degree of the polynomial model

kNN: k was a hyper-parameter

We turn model selection into choosing a hyper-parameter. For example, polynomial regression requires choosing a degree – this can be thought as model selection – and we select the model by tuning the hyper-parameter.





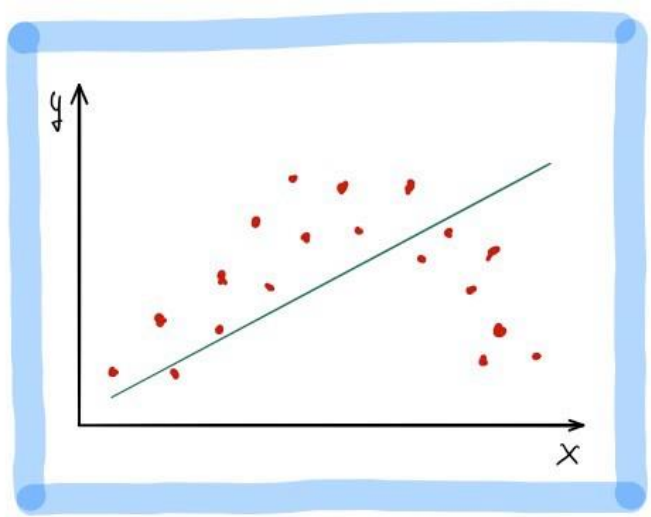


Underfitting: when the degree is too low, the model cannot fit the trend.

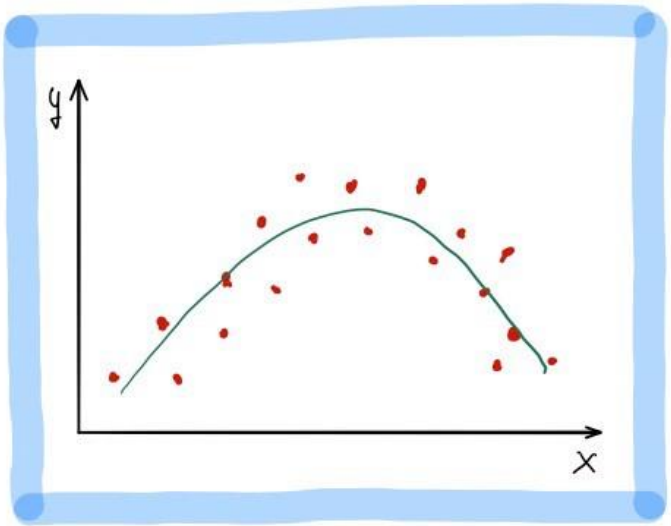
We want a model that fits the trend and ignores the noise.

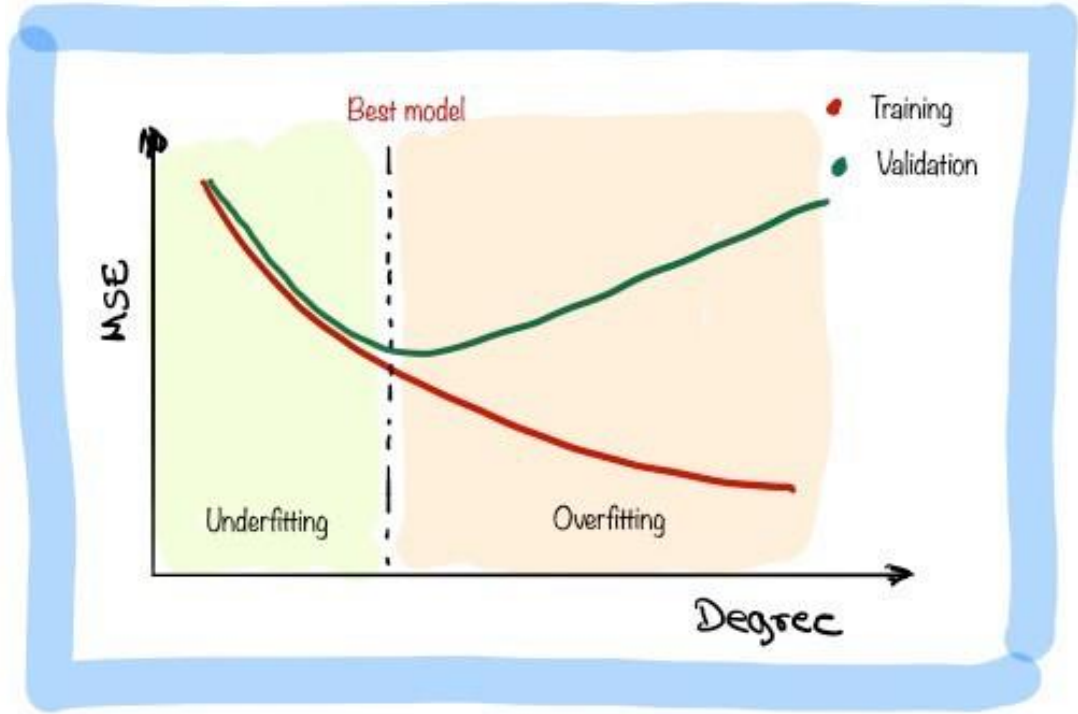
Overfitting: when the degree is too high, the model fits all the noisy data points.

Underfitting: train and validation error is high.

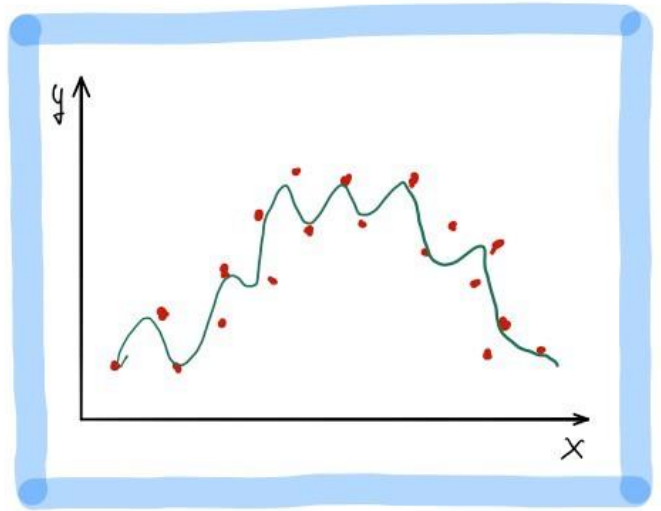


Best model: validation error is minimum.



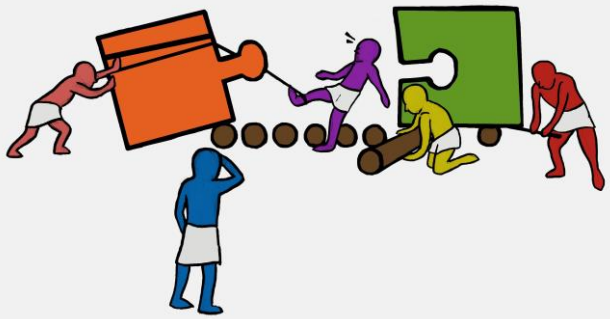


Overfitting: train error is low, validation error is high.



?

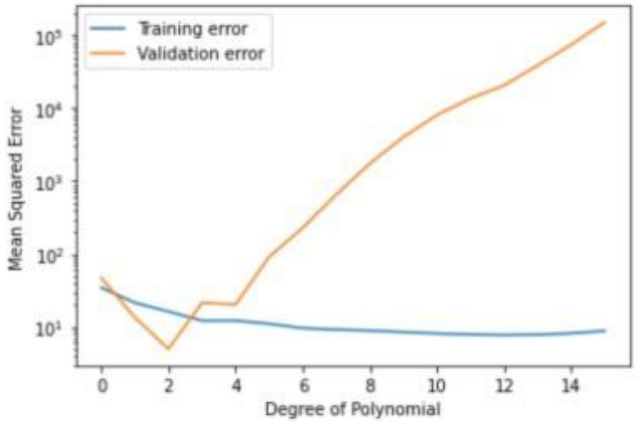
What are the parameters of the models and what are the hyperparameters?



Description

🏋️ Exercise: Best Degree of Polynomial with Train and Validation sets

The aim of this exercise is to find the **best degree** of polynomial based on the MSE values. Further, plot the train and validation error graphs as a function of degree of the polynomial as shown below.



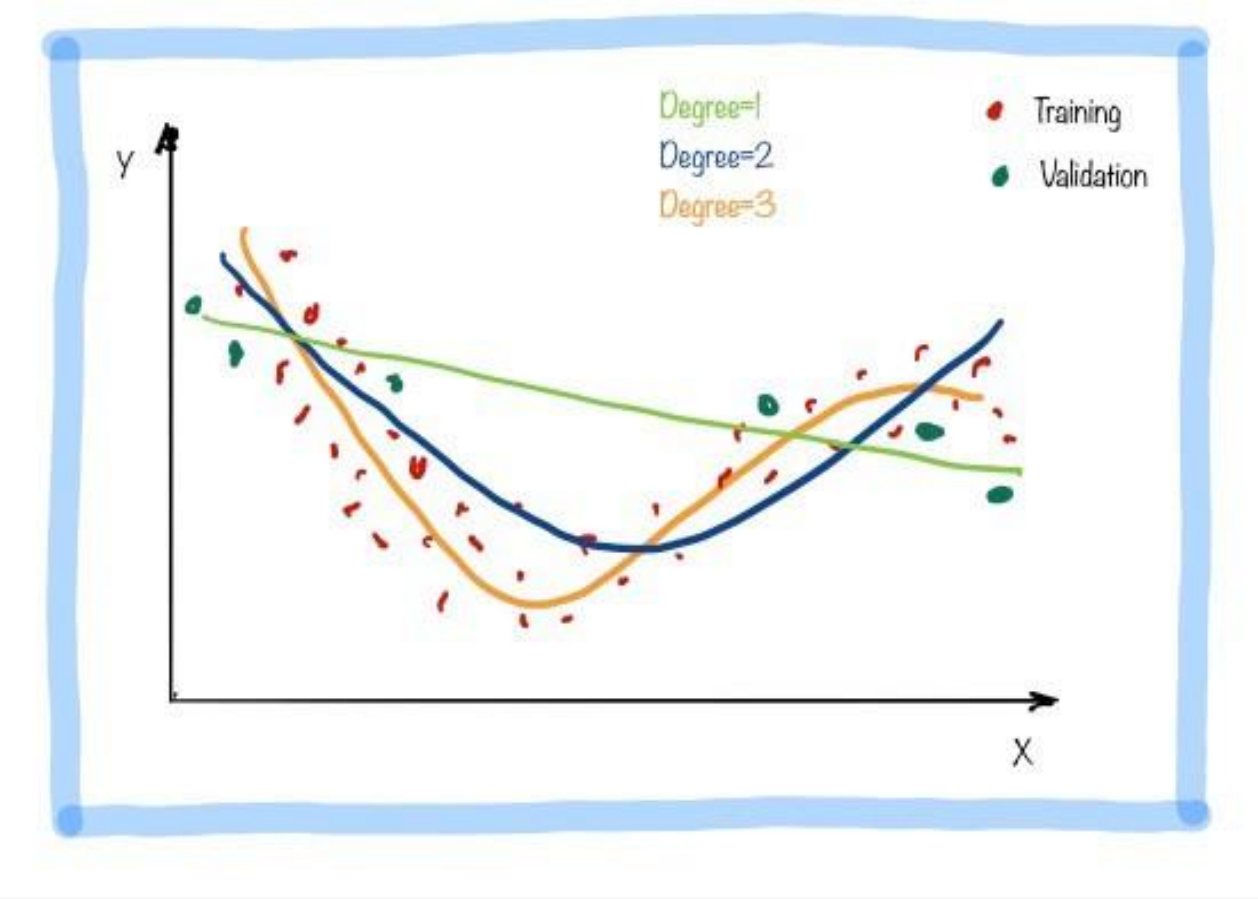
Instructions:

- Read the dataset and split into train and validation sets.
- Select a max degree value for the polynomial model.
- Fit a polynomial regression model on the training data for each degree and predict on the validation data.
- Compute the train and validation error as MSE values and store in separate lists.
- Find out the best degree of the model.
- Plot the train and validation errors for each degree.

Model Selection with Cross Validation

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.

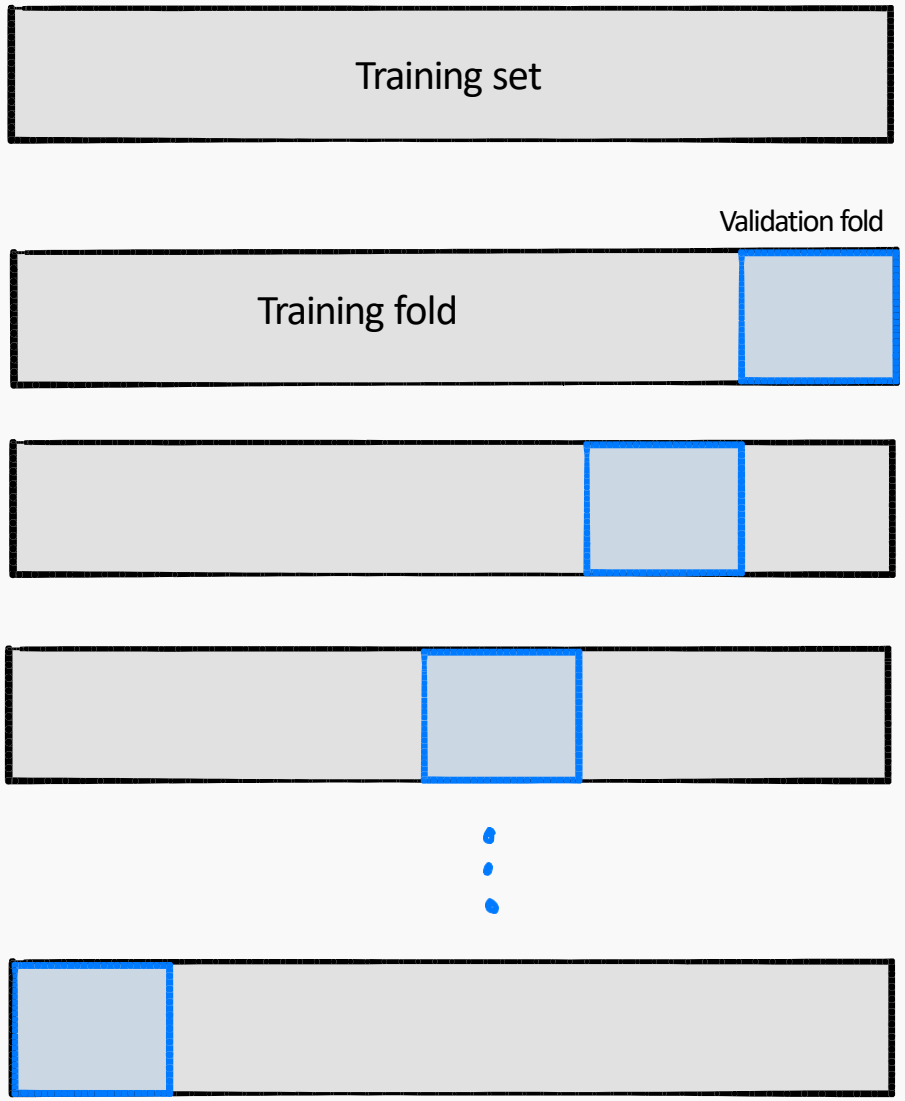
Cross Validation: Motivation

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



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

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

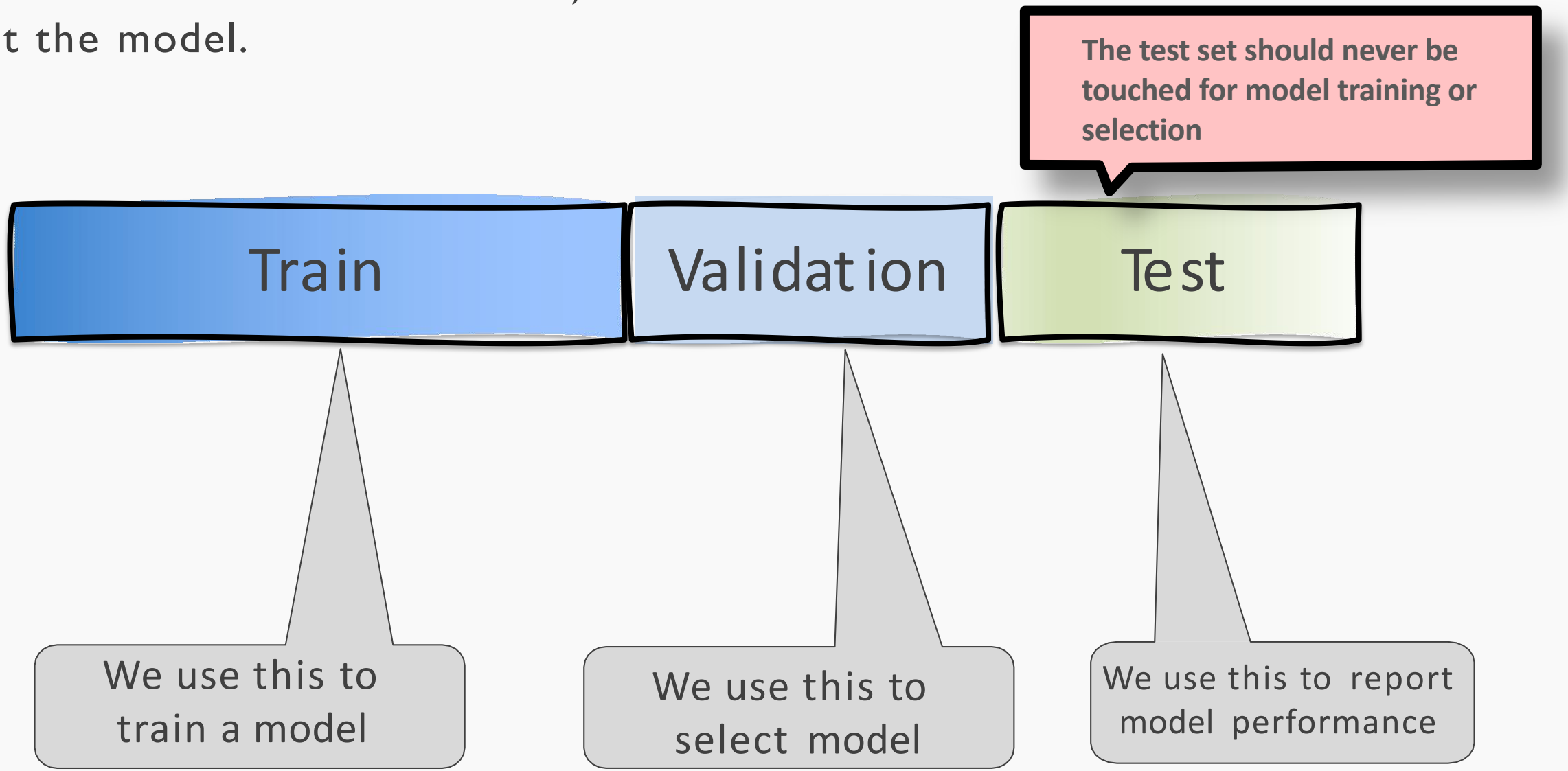
$$\mathrm{MSE}_3^{\mathrm{val}}$$

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

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

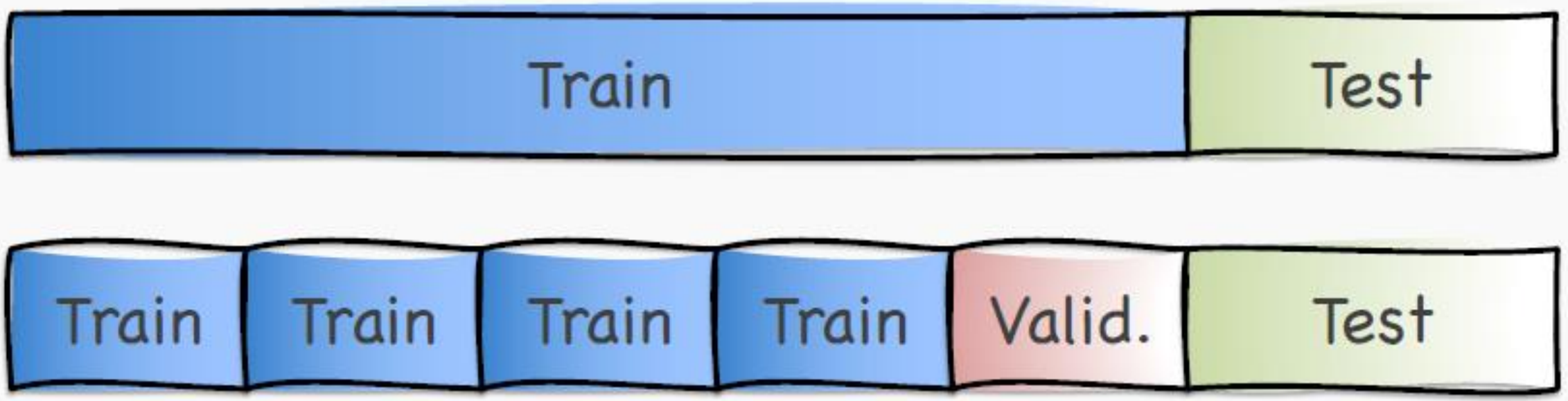
Train-Validation-Test

We introduce a different sub-set, which we called validation and we use it to select the model.

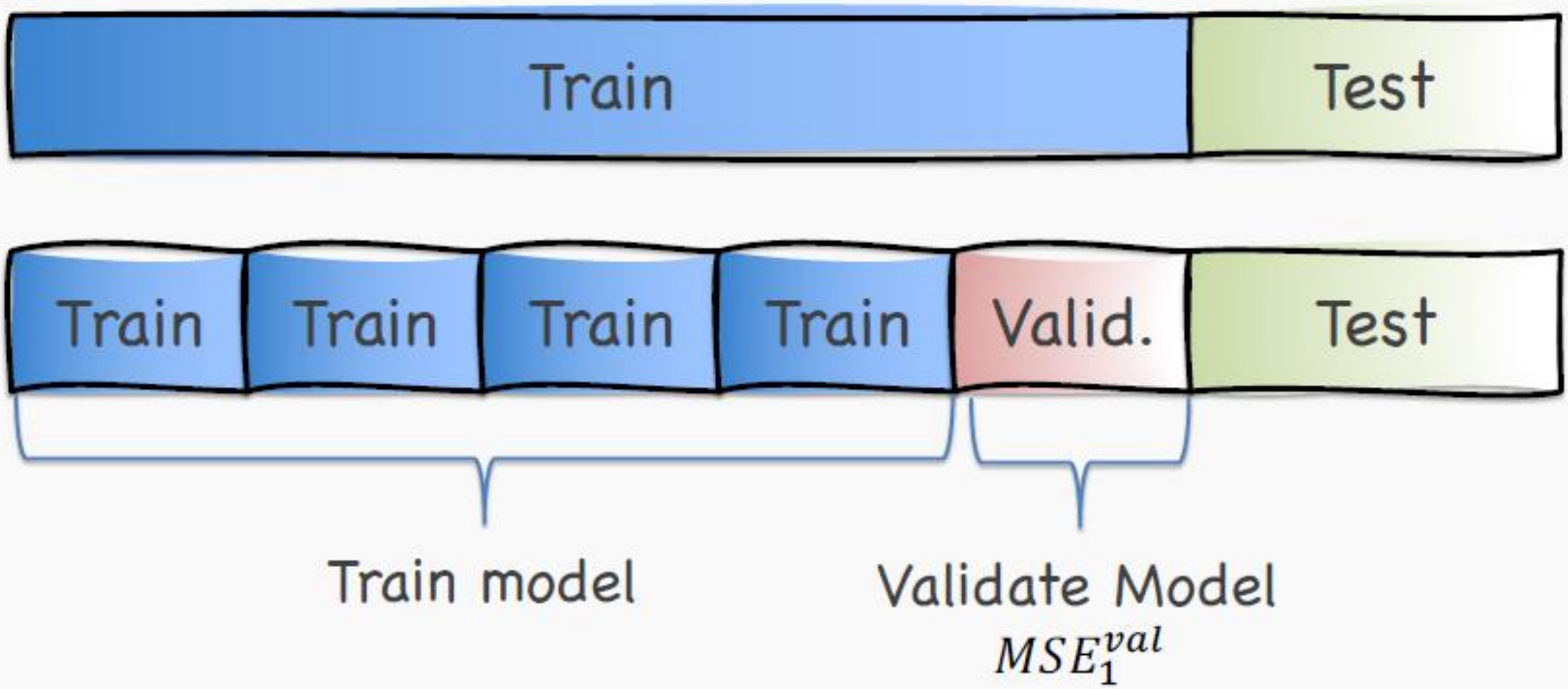


Cross Validation

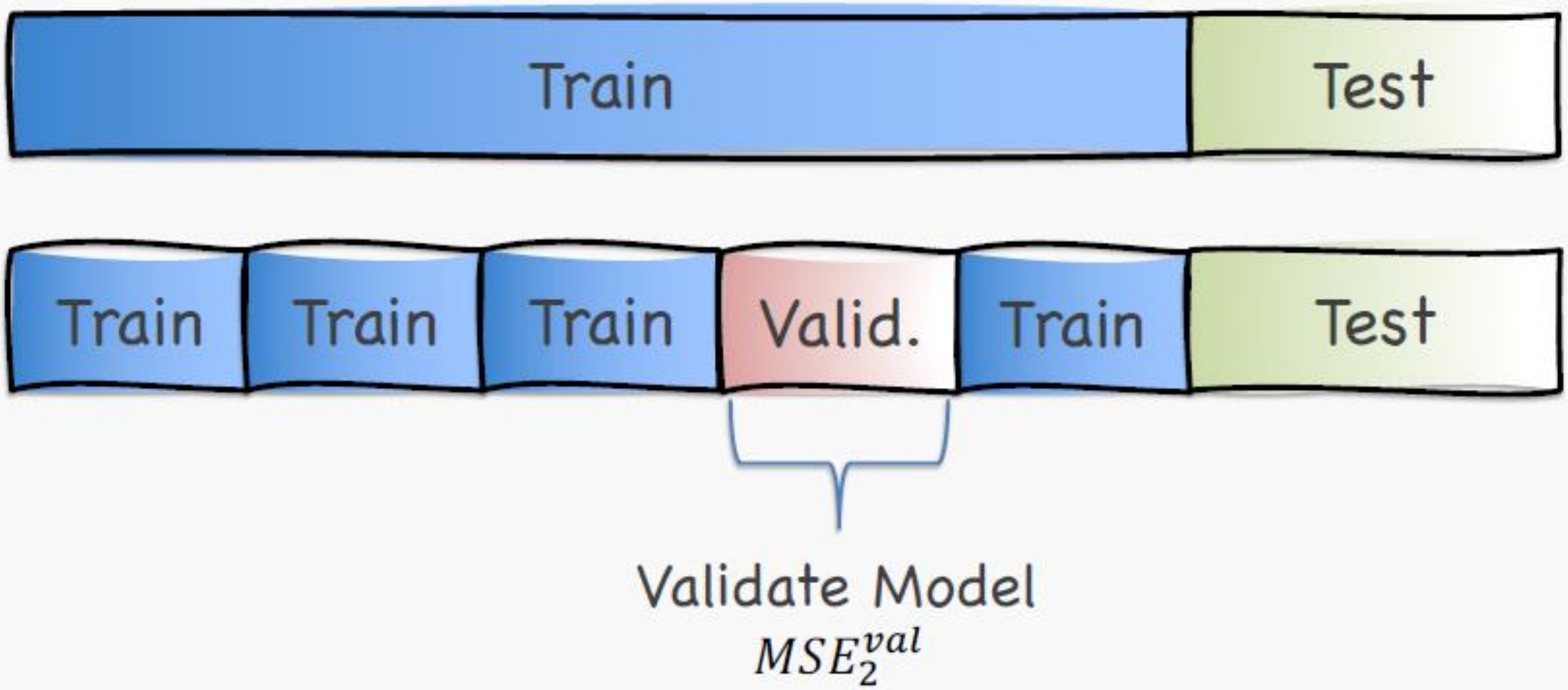
Cross Validation



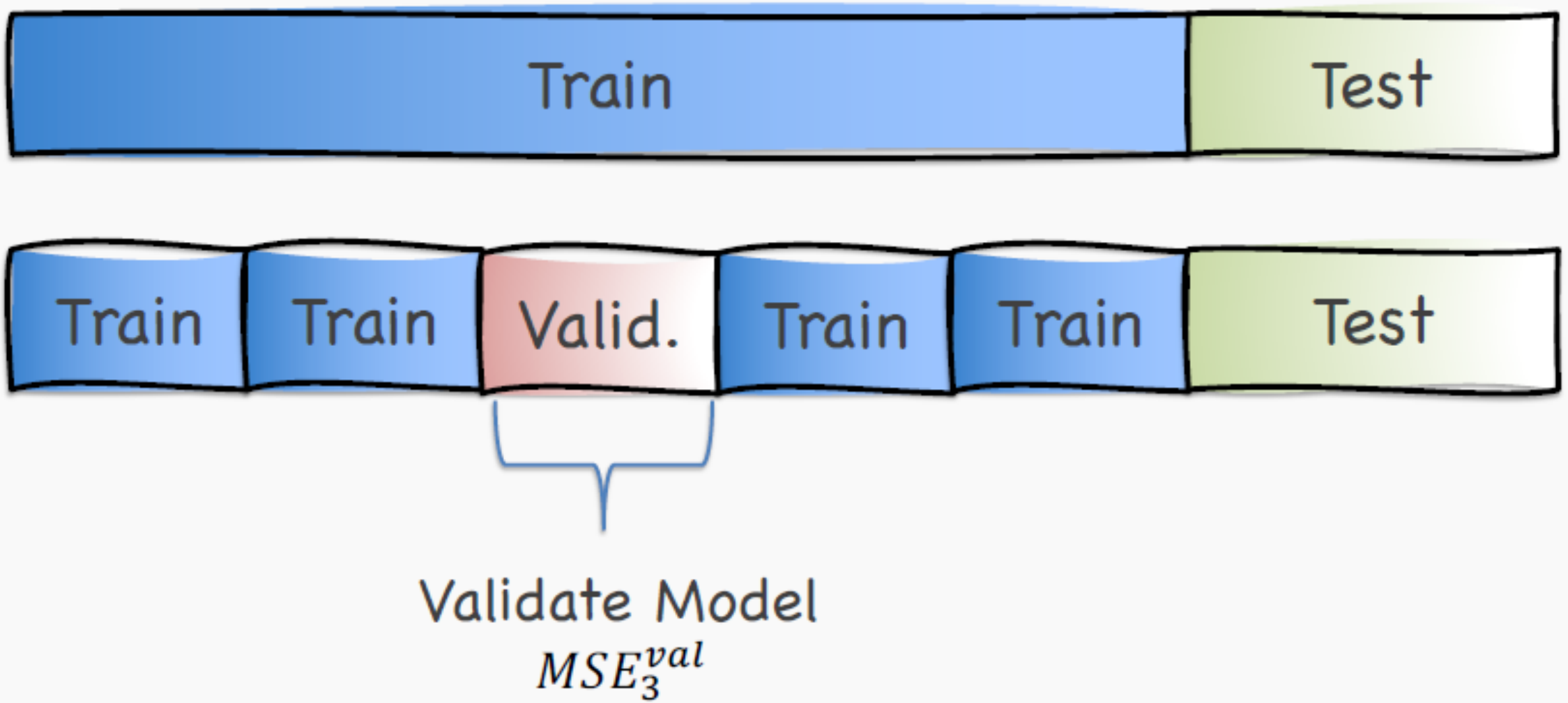
Cross Validation



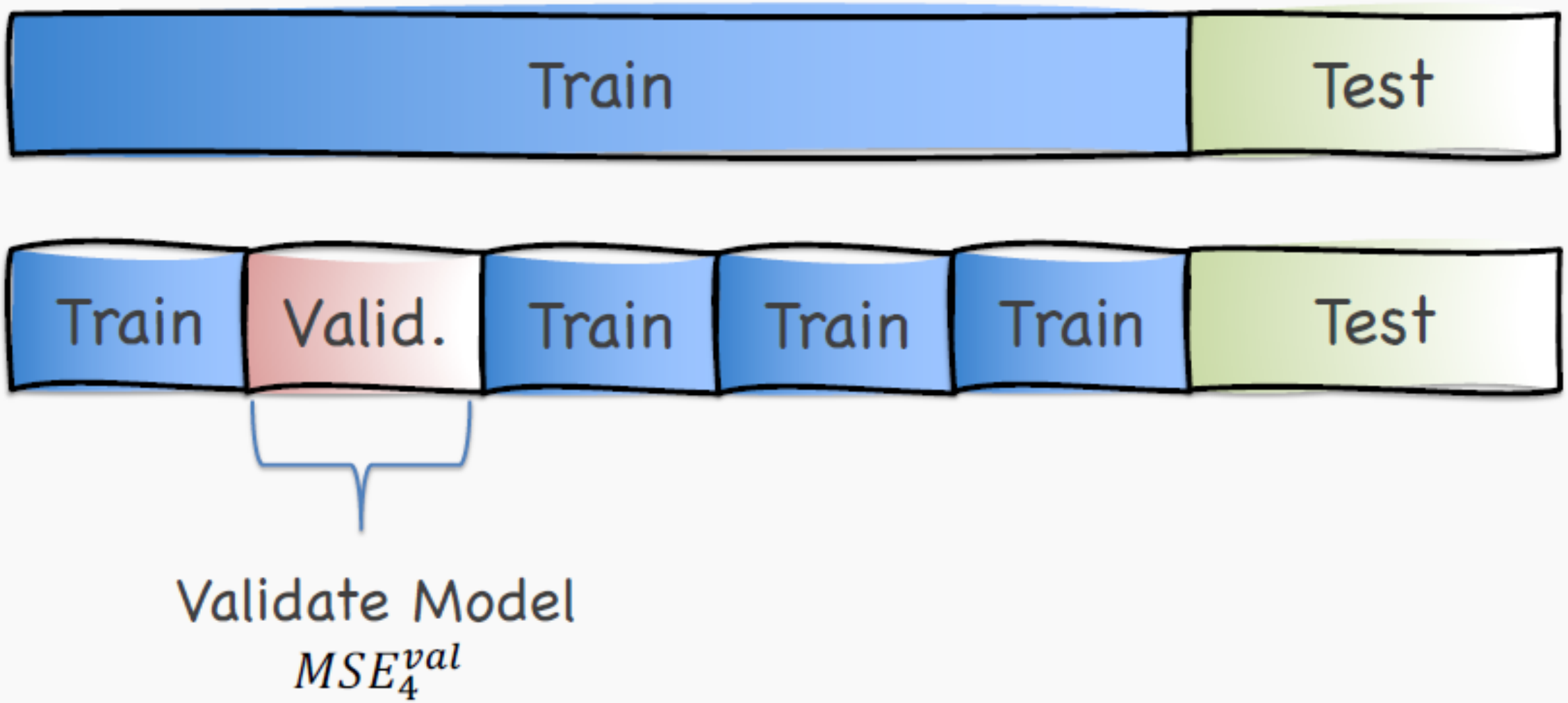
Cross Validation



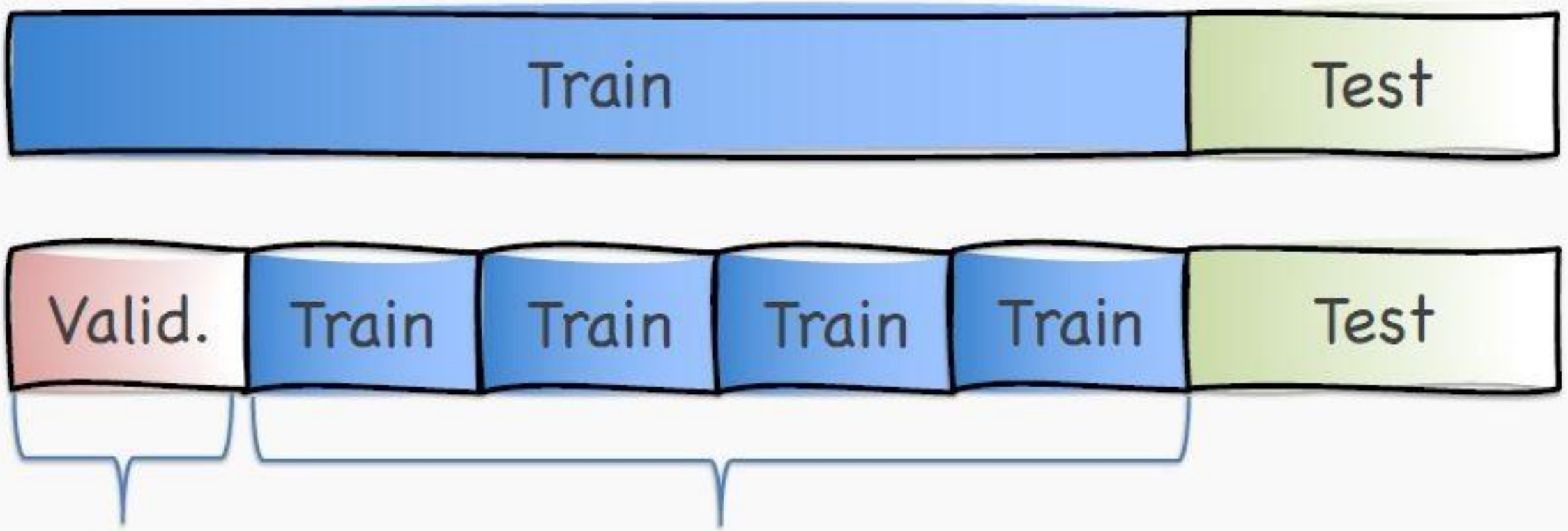
Cross Validation



Cross Validation



Cross Validation



$$MSE^{val} = \frac{1}{5}\sum_{i=1}^{5} MSE_i^{val}$$

K-Fold Cross Validation

Given a data set $\{X_1, ..., X_n\}$, where each $\{X_1, ..., X_n\}$ 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, $\{C_1, ..., C_k\}$
- 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.

The model used to fit the data

The data to fit

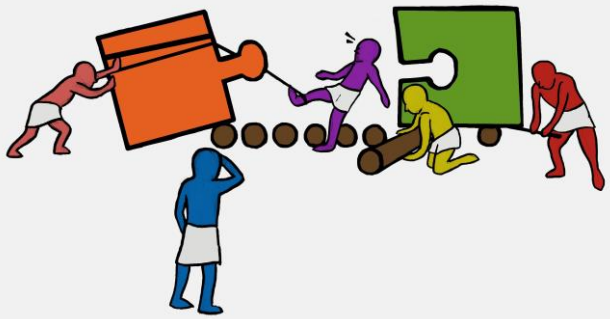
The target variable to predict on

```
sklearn.model_selection.cross_validate(estimator, X, y,
                                       scoring, cv, return_train_score)
```

Number of folds

Strategy to evaluate the performance of the cross-validated model on the test set.
Use "neg_mean_squared_error" for regression

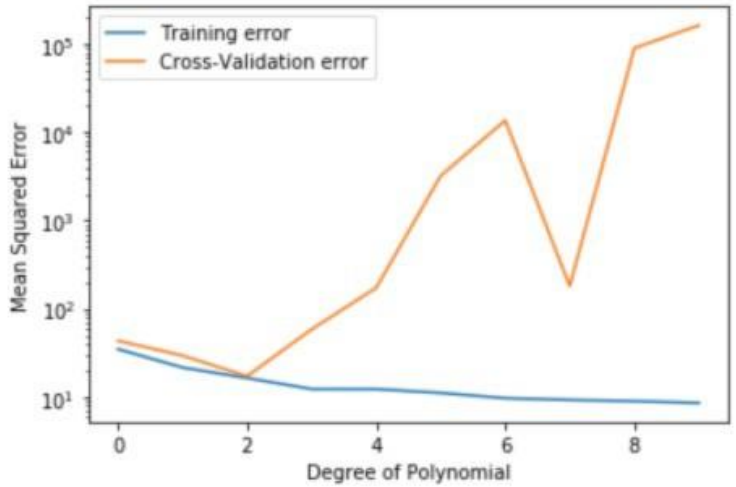
Set to True to include train scores



Description

🏋 Exercise: Best Degree of Polynomial using Cross-validation

The aim of this exercise is to find the **best degree** of polynomial based on the MSE values. Further, plot the train and cross-validation error graphs as shown below.



Instructions:

- Read the dataset and split into train and validation sets.
- Select a max degree value for the polynomial model.
- For each degree:
  - Perform k-fold cross validation
  - Fit a polynomial regression model for each degree on the training data and predict on the validation data
- Compute the train, validation and cross-validation error as MSE values and

KNN Revisited

Recall a simple, intuitive, non-parametric model for regression - the kNN model. We saw that it is vitally important to select an appropriate *k* for the data.

If the *k* is too small then the model is very sensitive to noise (since a new prediction is based on very few observed neighbors), and if the k is too large, the model tends towards making constant predictions.

A principled way to choose *k* is through *K*-fold cross validation.

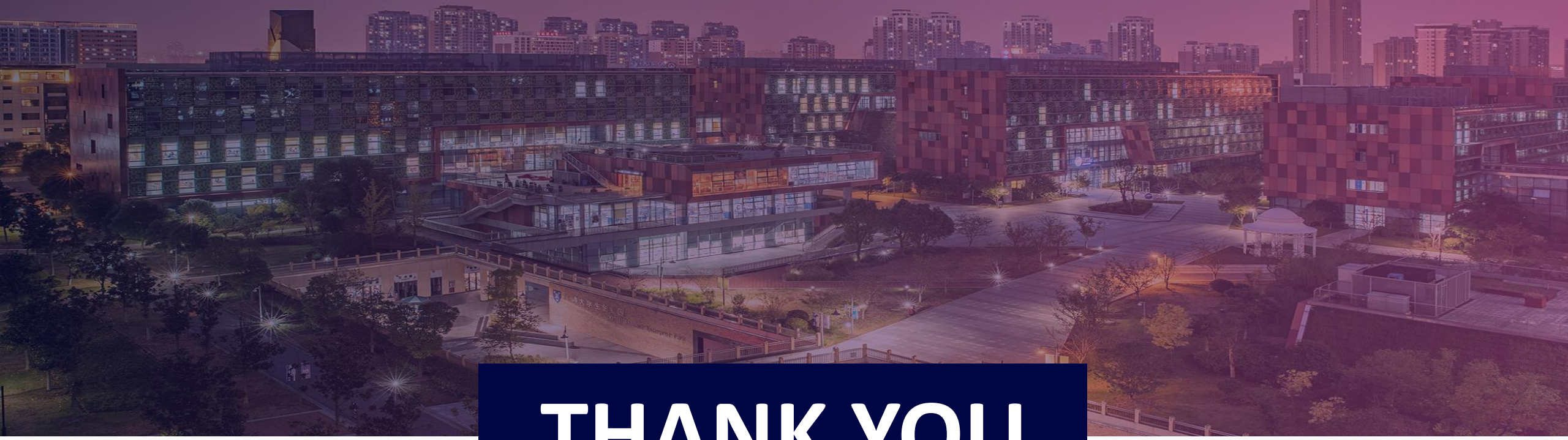
Choosing number of folds?

A higher k (number of folds) means that each model is trained on a larger training set and tested on a smaller test fold. In theory, this should lead to a *lower* prediction error as the models see more of the available data.

A lower k means that the model is trained on a smaller training set and tested on a larger test fold. Here, the potential for the data distribution in the test fold to differ from the training set is bigger, and we should thus expect a *higher* prediction error on average.

Careful Considerations

- Time-series dataset
  - We can still use cross-validation for time-series datasets using some other technique such as time-based folds.
- Unbalanced dataset
  - One way to consider is the use of stratified sampling instead of splitting randomly. Here is a fantastic blog article discussing how to handle this situation.
- Nested cross-validation
  - The nested keyword comes to hint at the use of double cross-validation on each fold. The hyper-parameter tuning validation is achieved using another k-fold splits on the folds used to train the model.
- Overfitting
  - We rely on using a validation dataset for early stopping and parameter tuning different from the testing set during the building of models.



THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大學