Machine Learning for Natural Language Processing

Arieda Muço[1]

Central European University

[1] based on material from Andreas Mueller (slides and book)

Missing observations

Missing Values

Missing values can be encoded in many ways

- Numpy has no standard format for it (often np.NaN)
- Sometimes you will encounter missing values encoded as: 999 (you can go on adding more "9" here), "???", "?", "np.inf", "N/A", "Unknown", "." ?
- Often missingness is informative
  - Checking other covariates
  - Code missing as indicator 1 if missing and zero otherwise.
- If you use variables in the dataset to predict if missing category, make sure your outcome is not one of them. Why? Can you show this?

Do

Understand your data. Some common imputation methods for replacing missing values are

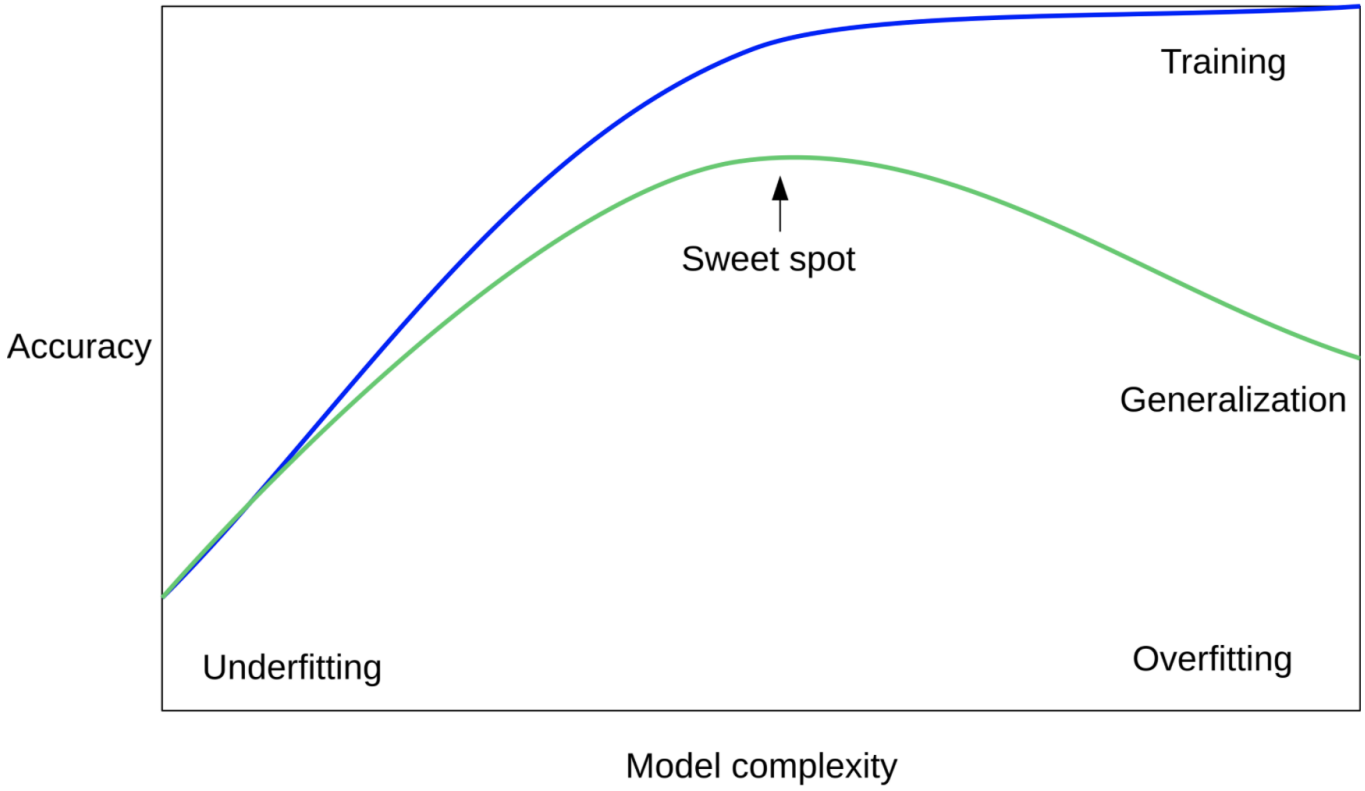
- Mean or Median – conditioning for other covariates. Regression models
- K-nearest neighbours
  - Find k nearest neighbors that have non-missing values
  - Fill in all missing values using the average of the neighbors
- Model-Driven Imputation. Train regression model for missing values (retrain model after filling in)

Don't

- Drop observations with the default drop missings command (Stata/Python/R...)
- Start doing analysis without understanding the missings
  - Algorithms don't understand missings and will drop them. You will end up with a selected sample.
  - If your missings are coded as "9999" will skew the data and your results are not valid

Cross-Validation

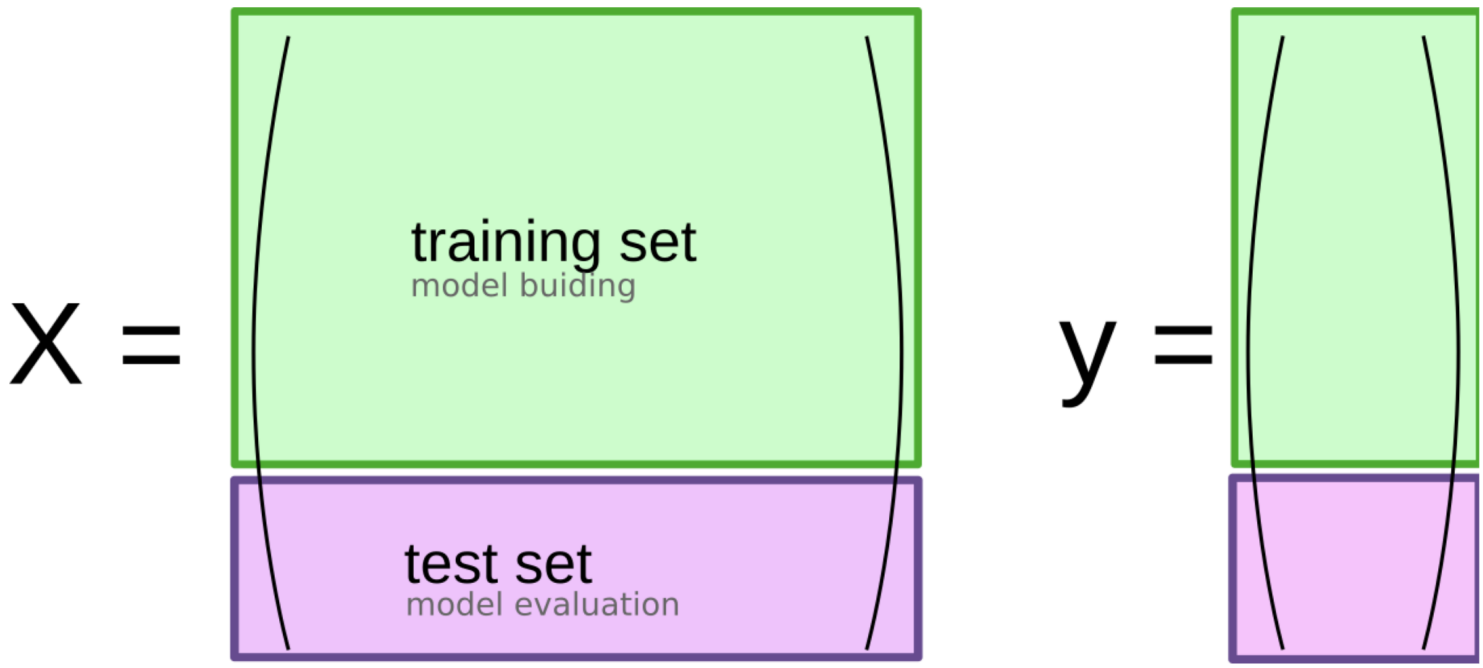
Overfitting and Underfitting



Questions

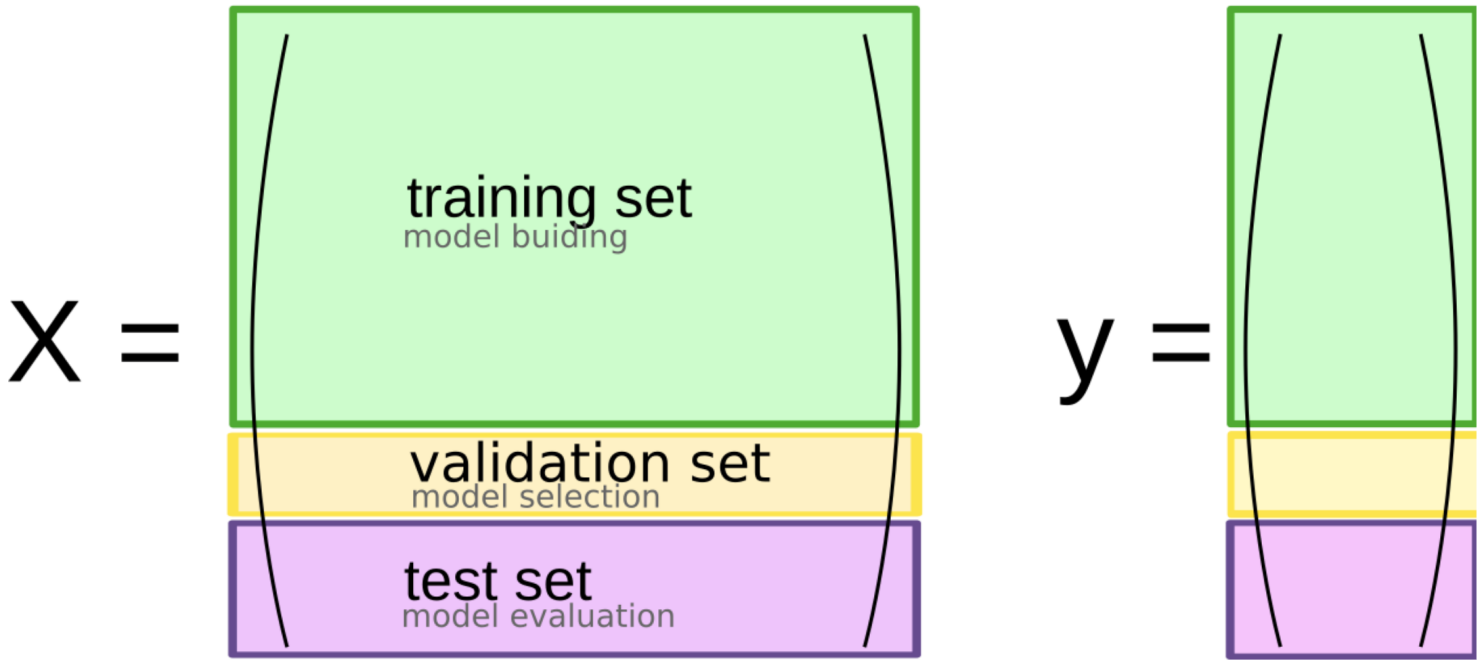
- How are regularization techniques related with overfitting?
- What are the regularization techniques we have seen?
- Why is it important to scale variables/features when performing regularization?

We've seen



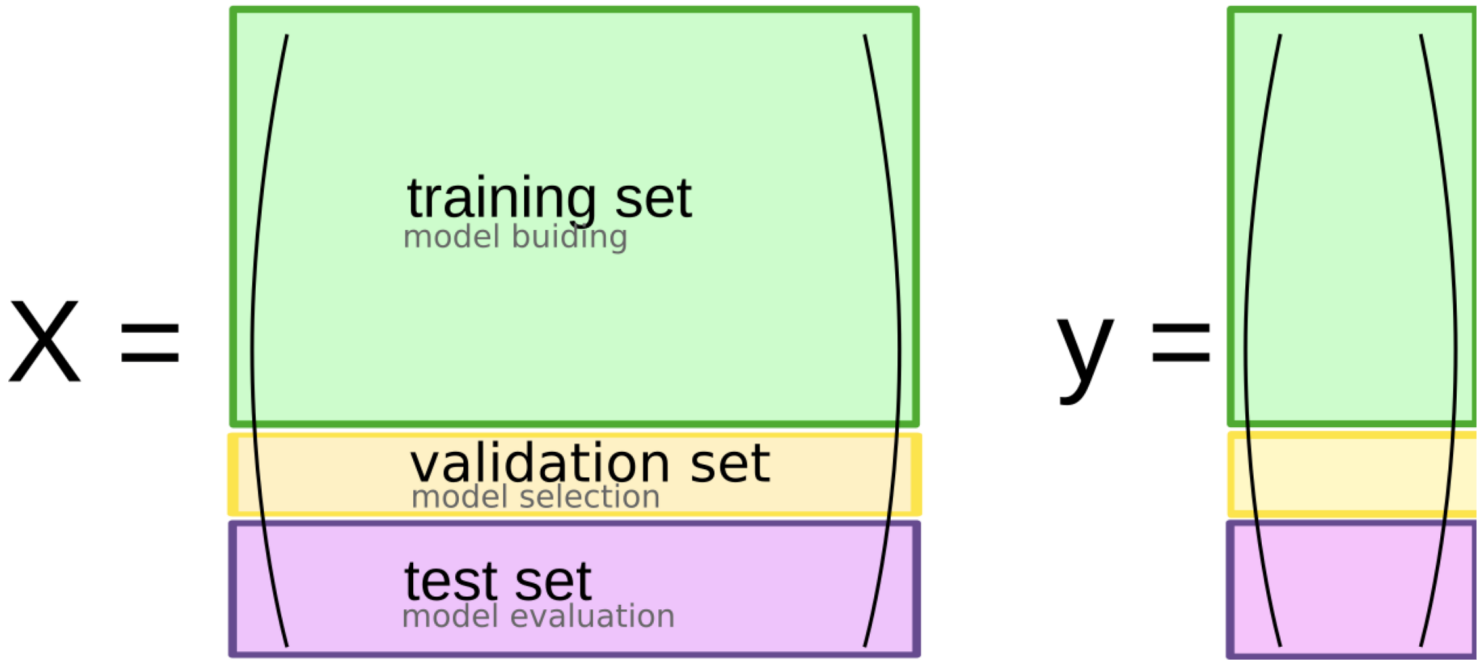
We've seen

Better: Threefold split

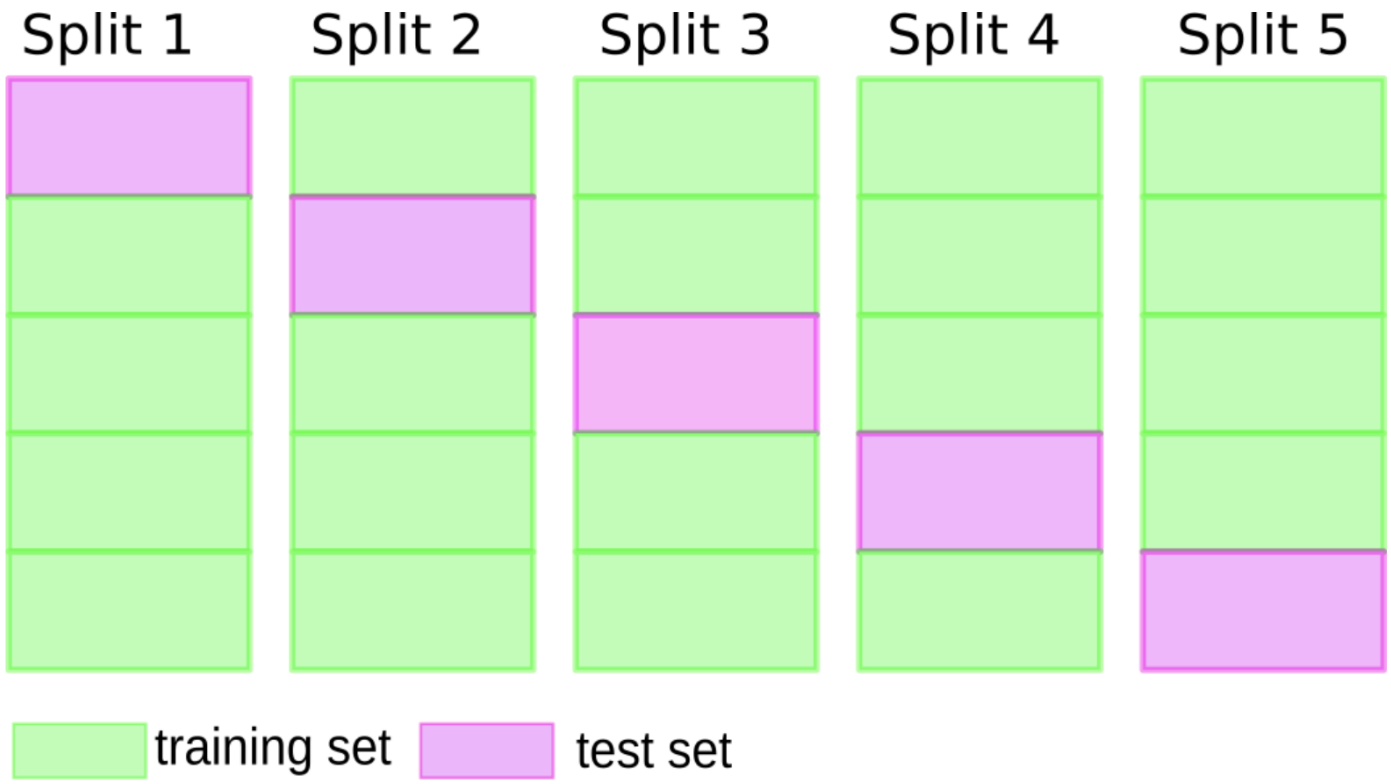


We've seen

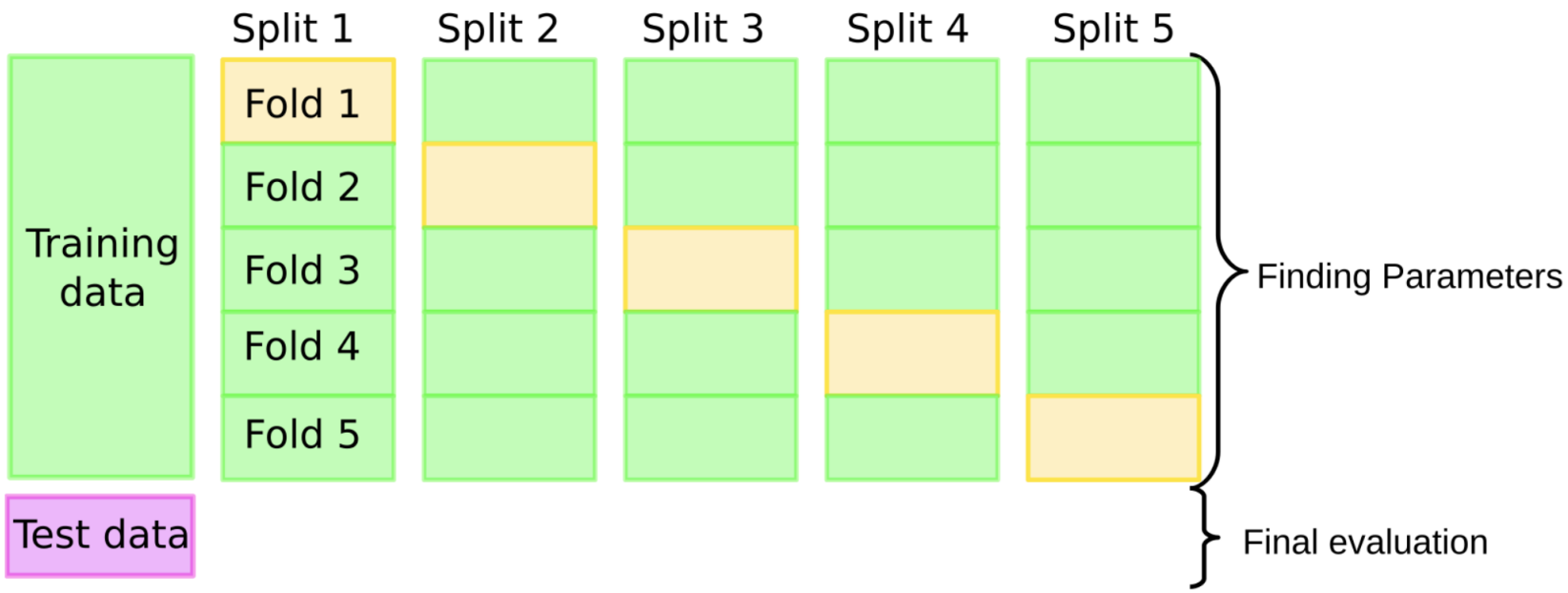
Better: Threefold split



Cross-Validation



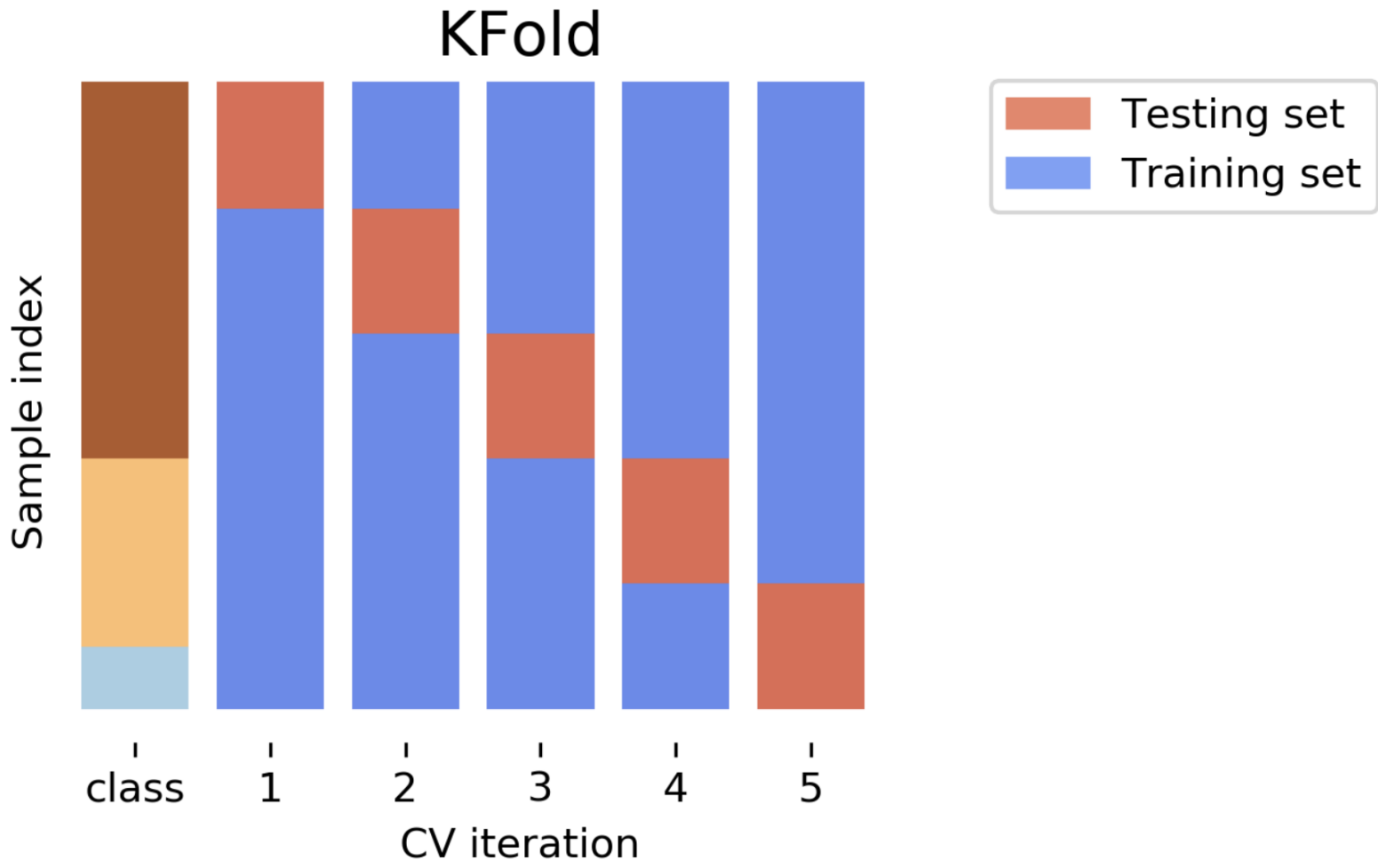
Cross-Validation

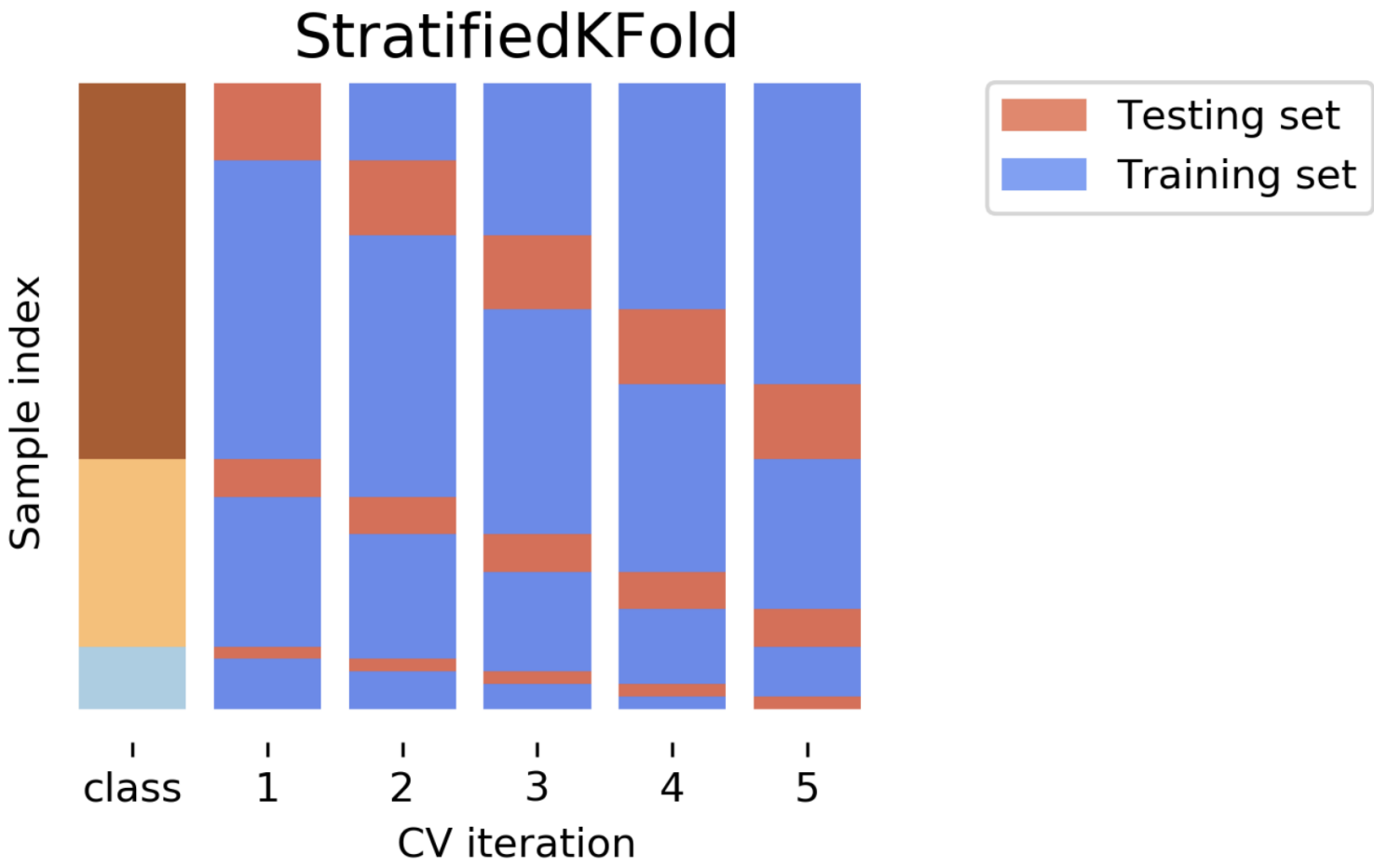


Reminder!

- Cross-validation does not return a model
- When calling cross_val_score, multiple models are built internally, but the purpose of cross-validation is only to evaluate how well a given algorithm will generalize when trained on a specific dataset

Cross-Validation Strategies


KFold
Testing set
Training set
Sample index
class
1
2
3
4
5
CV iteration



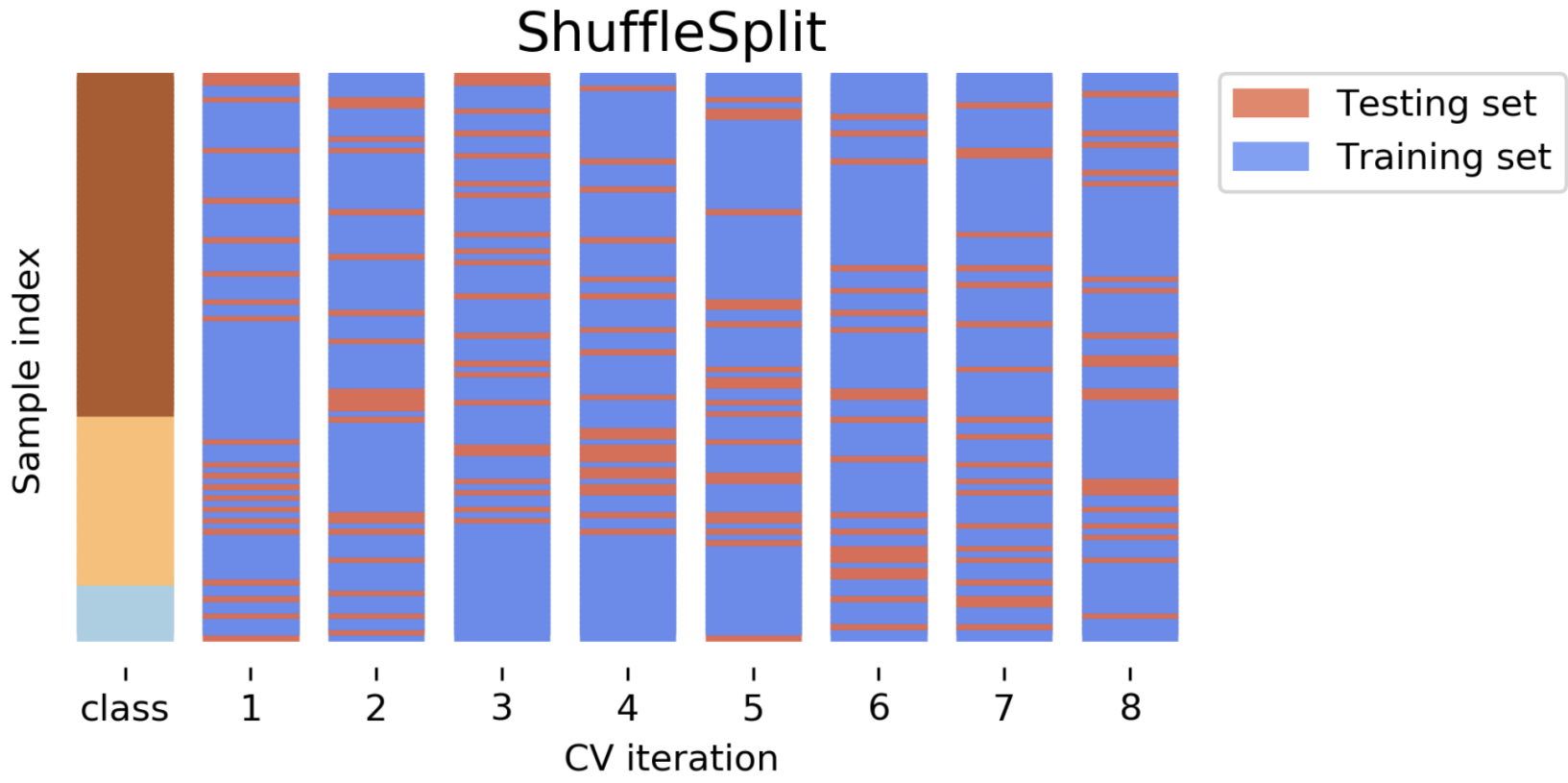
Stratified: Ensure relative class frequencies in each fold reflect relative class frequencies on the whole dataset.

Why stratifying?

- Assume a (multi-class) classification problem. What happens if a class not stratified by chance?

Repeated K-Fold and Leave-One-Out

- Leave-One-Out: K-Fold (number of folds equals number of observations)
  - Drawback: High variance, takes a long time
- Better: Shuffle Split (Monte Carlo)
  - Repeatedly sample a test set with replacement
- Even Better: Repeated K-Fold
  - Apply K-Fold or Stratified K-Fold multiple times with shuffled data



Number of iterations and test set size independent

Defaults in scikit-learn

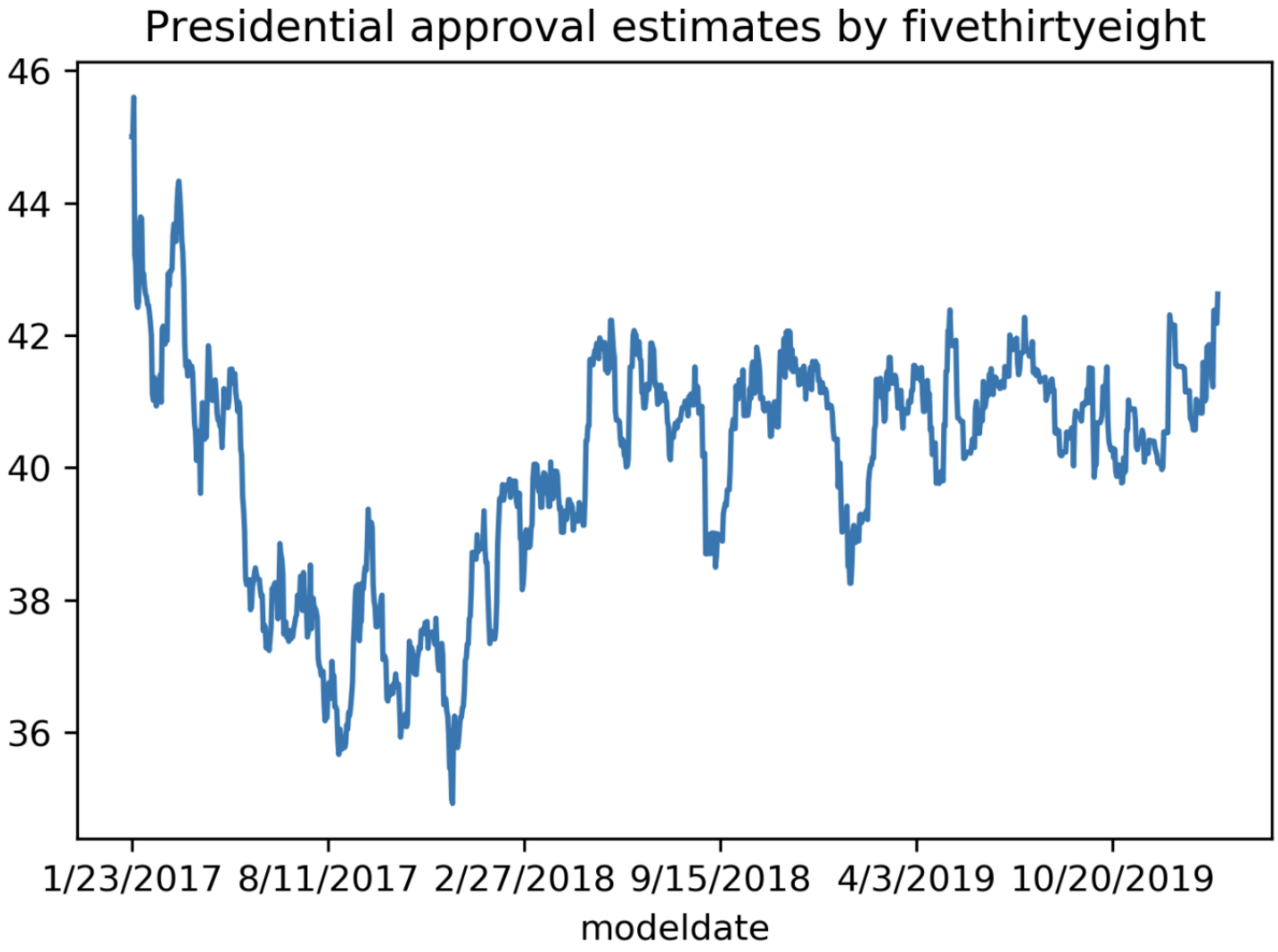
- 5-fold in 0.22 (used to be 3 fold)
- For classification cross-validation is stratified
- train_test_split has stratify option: train_test_split(X, y, stratify=y)
- No shuffle by default!

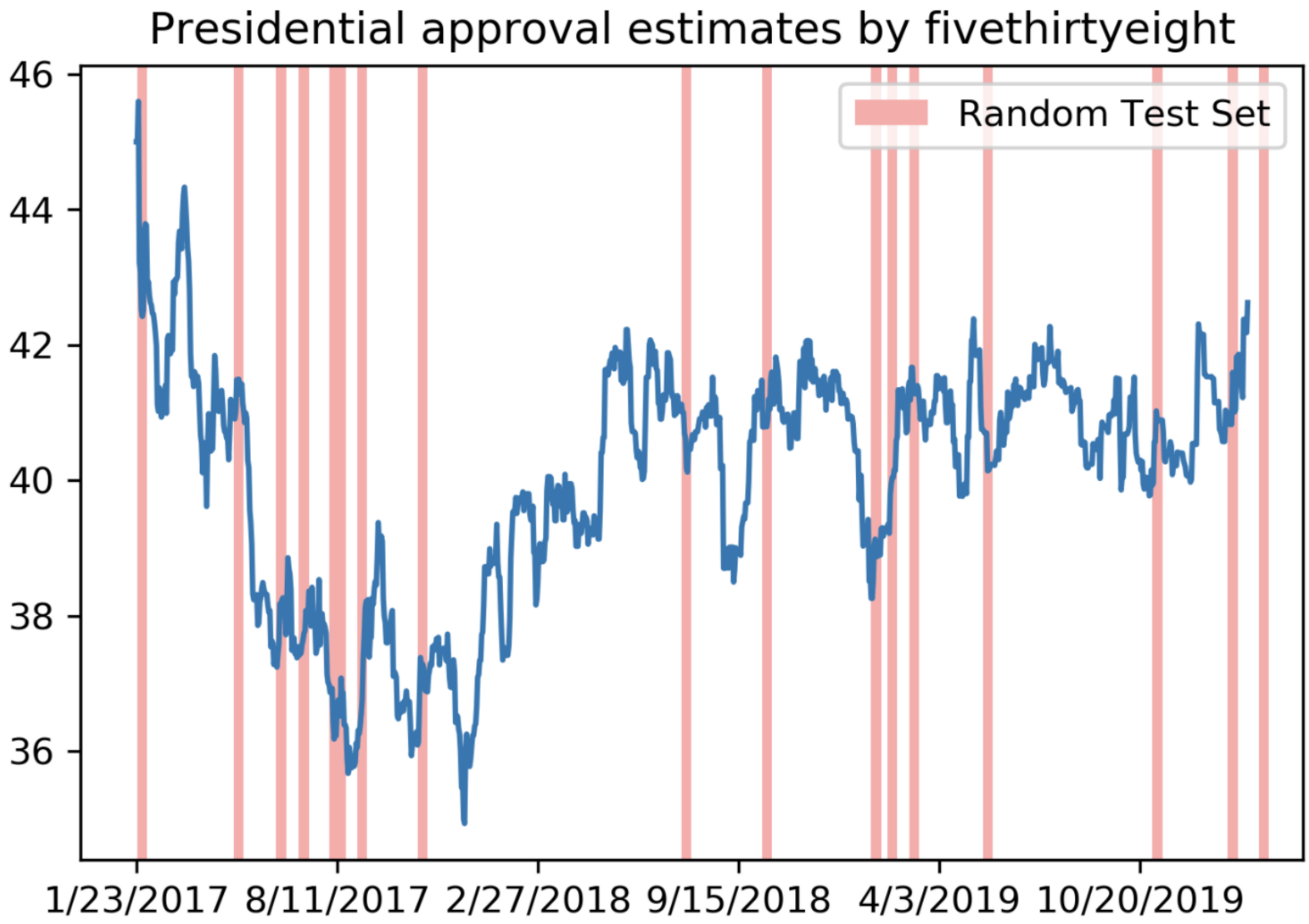
Cross-Validation for correlated data

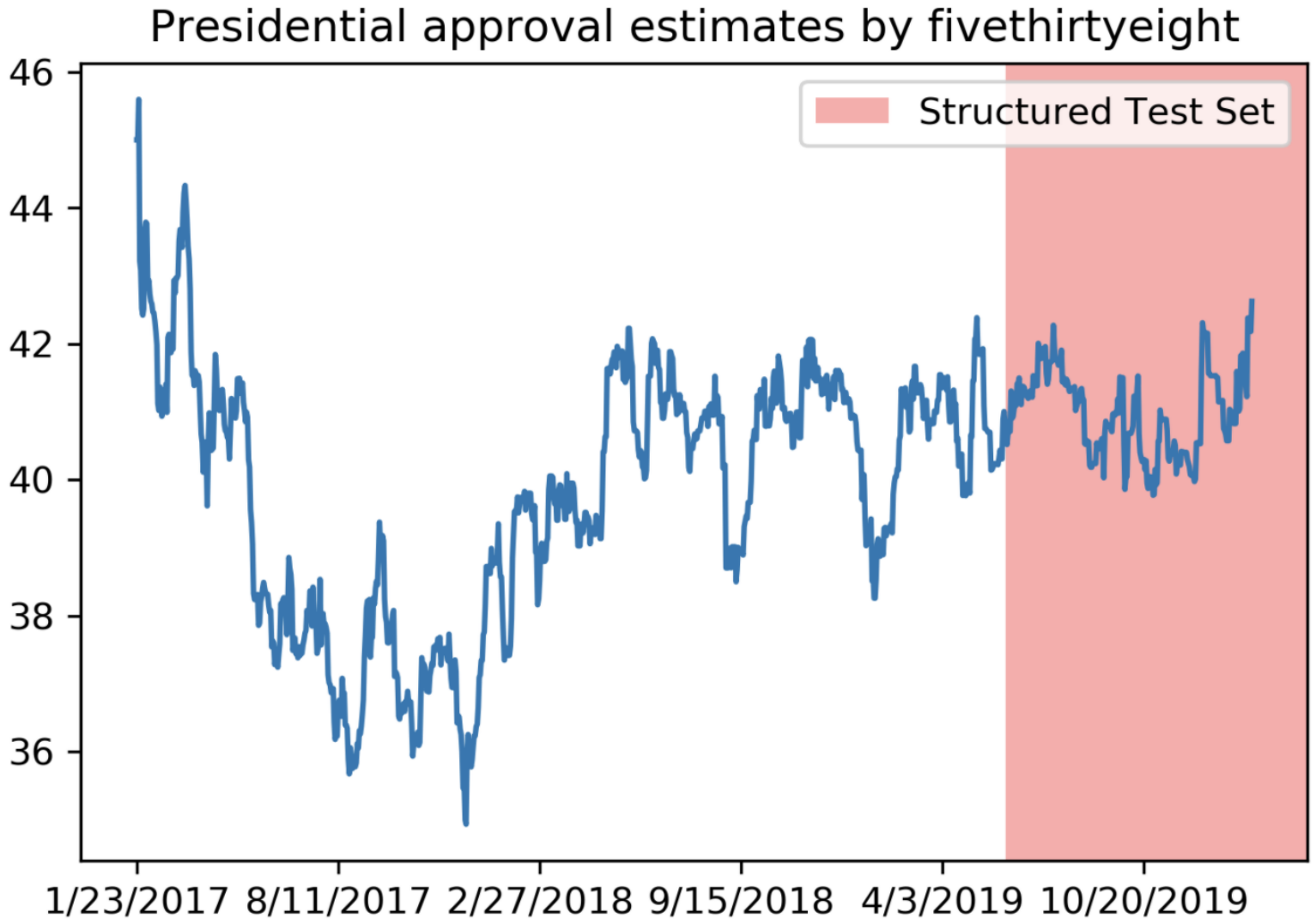
Cross-Validation non-iid data

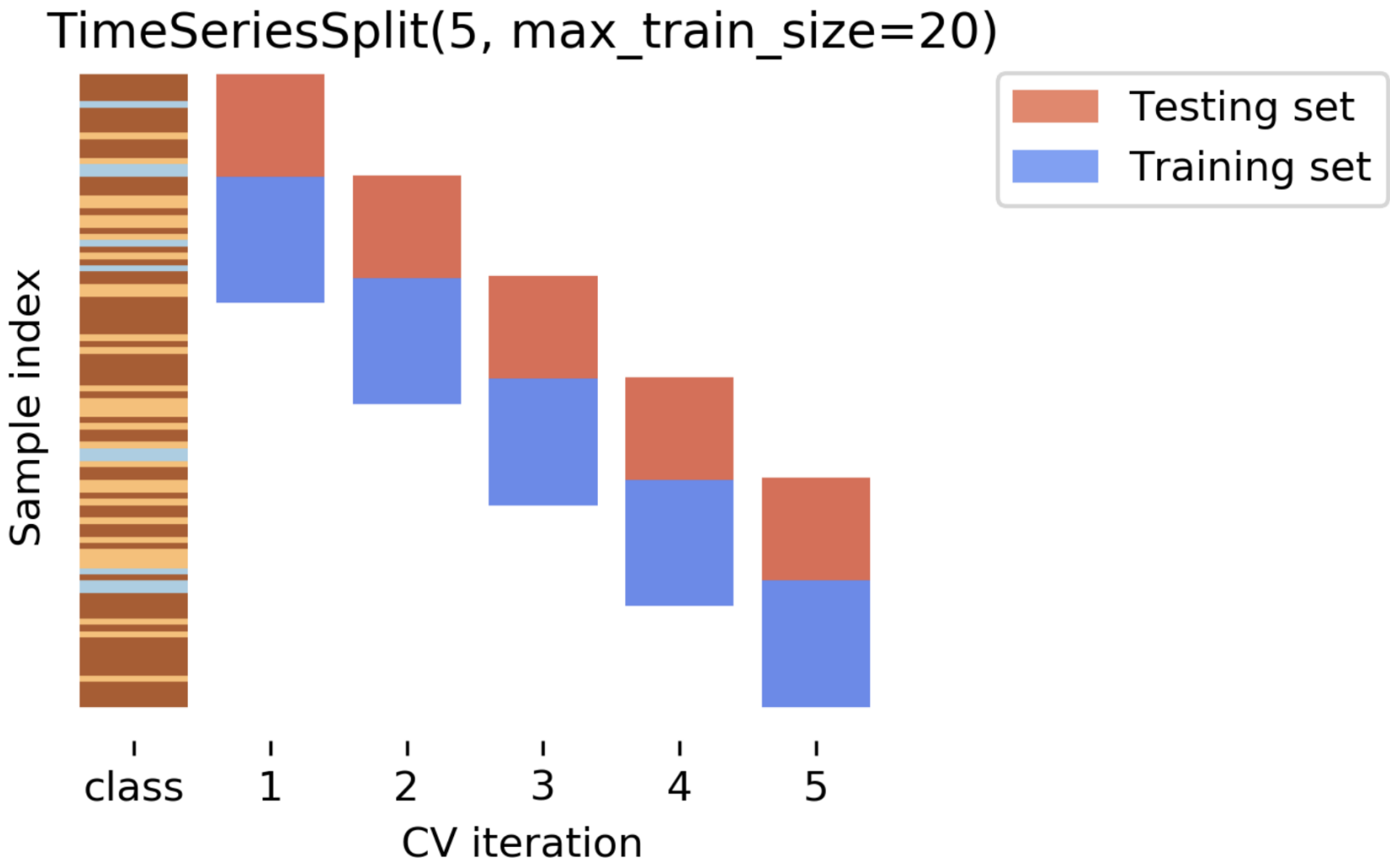
Know your data and setting

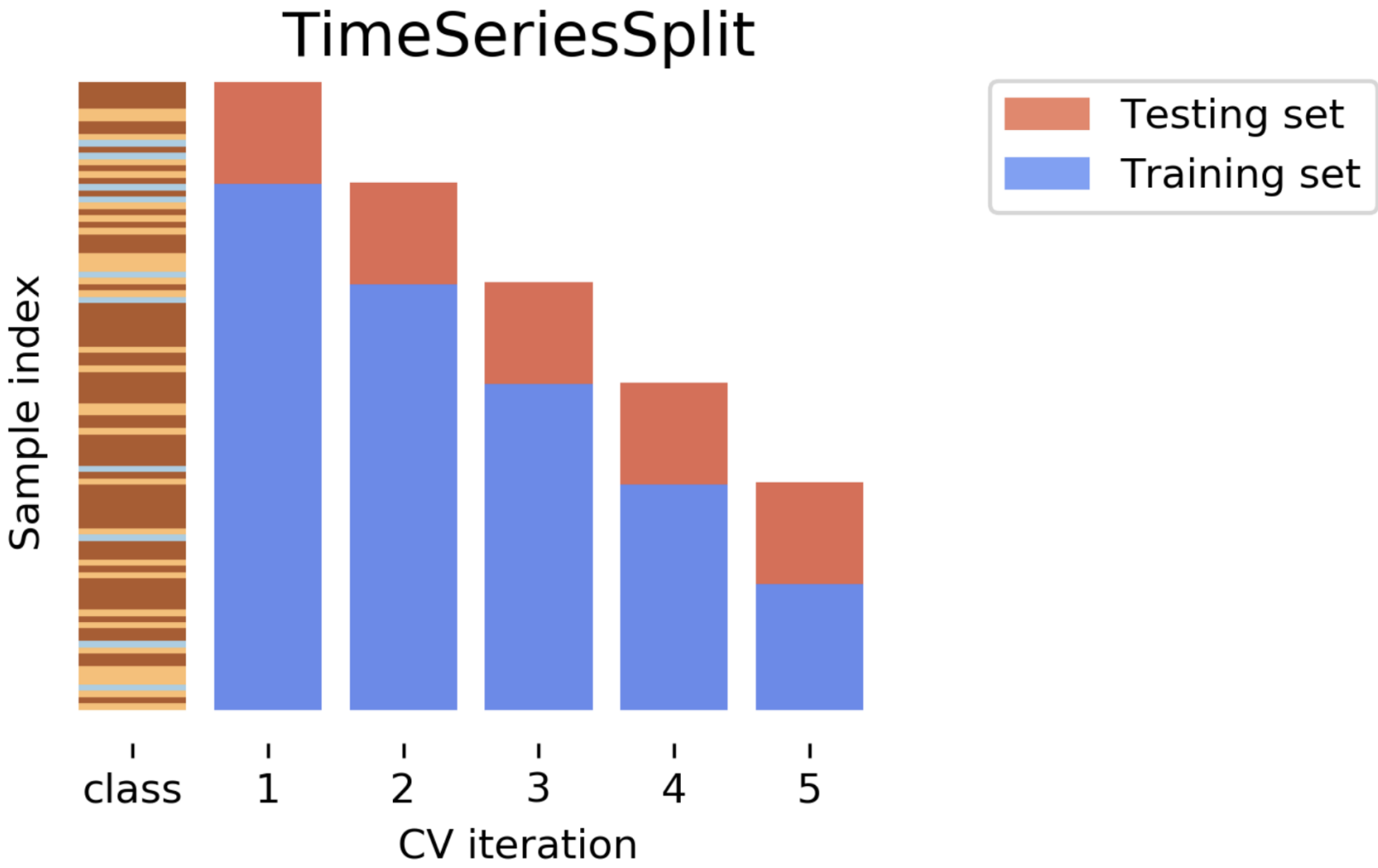
- Time-series data have a time component
- Geographical data have a spatial component
- Grouped data
- DON'T SHUFFLE if dealing with any of the above. Why?


Presidential approval estimates by fivethirtyeight
46
44
42
40
38
36
1/23/2017
8/11/2017
2/27/2018
9/15/2018
4/3/2019
10/20/2019
modeldate


Presidential approval estimates by fivethirtyeight
Random Test Set
46
44
42
40
38
36
1/23/2017
8/11/2017
2/27/2018
9/15/2018
4/3/2019
10/20/2019


Presidential approval estimates by fivethirtyeight
Structured Test Set
46
44
42
40
38
36
1/23/2017
8/11/2017
2/27/2018
9/15/2018
4/3/2019
10/20/2019


TimeSeriesSplit(5, max_train_size=20)
Testing set
Training set
Sample index
class
1
2
3
4
5
CV iteration


TimeSeriesSplit
Testing set
Training set
Sample index
class
1
2
3
4
5
CV iteration

Model Interpretation and Feature Selection

Model Interpretation

Black-box

- No inference
- No causality
- Still useful

Types of explanations

Explain model globally

- How does the output depend on the input?
- Often: some form of marginals

Explain model locally

- Why did it classify this point this way?
- Explanation could look like a "global" one but be different for each point
- What is the minimum change to classify it differently?

explain model $\neq$ explain data

Explain model globally

- Model inspection only tells you about the model
- The model might not accurately reflect the data

Features important to the model

Naive:

- coef_ for linear models
- feature_importances_ for tree-based models

Use with care!

Linear Model coefficients

- Relative importance only meaningful after scaling
- Correlation among features might make coefficients completely uninterpretable
- $L1$ regularization will pick one at random from a correlated group (try this in the regularization notebook)!
- Any penalty will invalidate usual interpretation of linear coefficients