Machine Learning: Processes

Akitaka Matsuo
Essex IADS

Machine learning steps

1. Define the problem
2. Split the data to train and test set
3. Data wrangling
4. Build the model (including cross-validation)
5. Evaluate the model
6. Apply the model (prediction)

Define the problem

University of Essex

- What kind of the insight do you want to get from the data
  - In supervised learning, the ultimate goal is to construct a model that closely approximates the generation of the outputs.
- Mathematically, it can be formulated as:

$$Y \sim f(X)$$

- $f(X)$ is the function to generate the outcome Y

Motivation for learning *f*(X)?

- We want to estimate $f(X)$, but what for?
  - **Inference**: Want to explain the relationship between X and Y.
    - Find key predictors
    - Effect of key predictors on outcome
    - Functional form
  - **Prediction**: Get the output from $f(X)$ as close as Y

$$\hat{Y} = \hat{f}(X)$$

The fundamental question then to ask is “Do we need to explain the effect of X on Y?”

- Sometimes it doesn’t matter (e.g. spam filter, weather forecasting)
- But often times it does matter in social science

Model estimation

There are two essentially different strategies:

- Parametric methods
  - Steps
    1. Decide functional form
    2. Estimate the parameter
- Non-parametric methods
  - make the estimated f as close to the data points as possible
  - e.g. KNN Regression

Example: KNN Regression

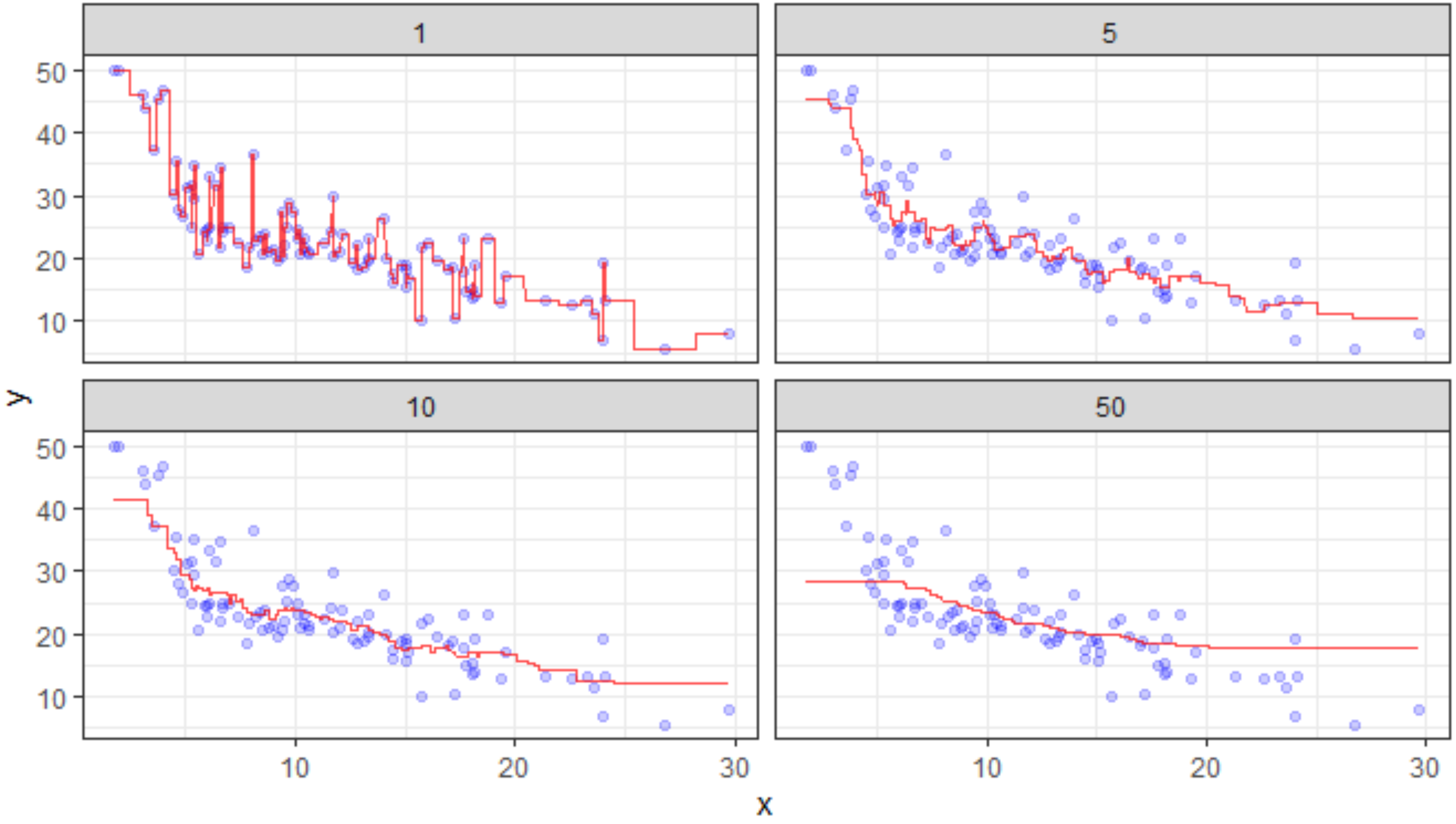
– KNN: K Nearest Neighbor

– KNN regression: Predict based on the mean of y values nearest k data points.

  - small k: prediction is very sensitive to local values
  - large k: prediciton is not so sensitive

See next slides for various k values

Example: KNN Regression



KNN regression

- From the figure, we learned k matters
- But, how to objectively evaluate the model
- For continuous output model it is typically Mean Squared Error (MSE) or Root MSE :

$$MSE = \frac{1}{n}\sum_{i}(y_i - \hat{f}(x_i))^2$$

- If the model is flexible, MSE gets smaller.
  - But is it always good?
  - **No!**

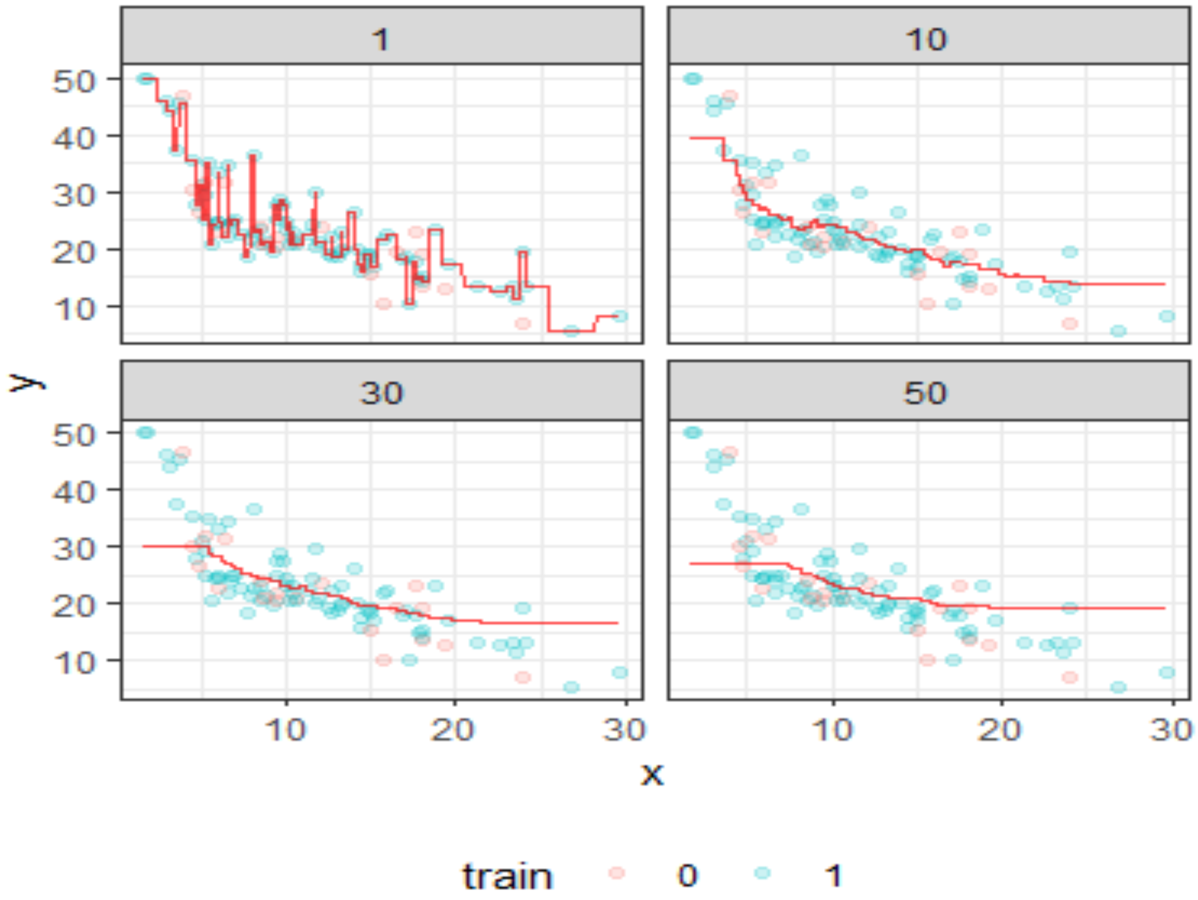
Train-test split

University of Essex

- If the model gets more flexible, the MSE for the data used for estimating the data will gets smaller
- But this does not guarantee the accuracy when the model applied to the new data.
- Here I split the data into Train set and Test set
  - **Train set**: use for estimating the model
  - **Test set**: use for evaluating the model

KNN Example again with Train-Test split

- For smaller k,
  - train-MSE is smaller
  - but not the case for test-MSE



Bias-Variance tradeff

- The previous example illustrates the issue of how flexible the model should be.
- This problem can be formulated in the following equation:

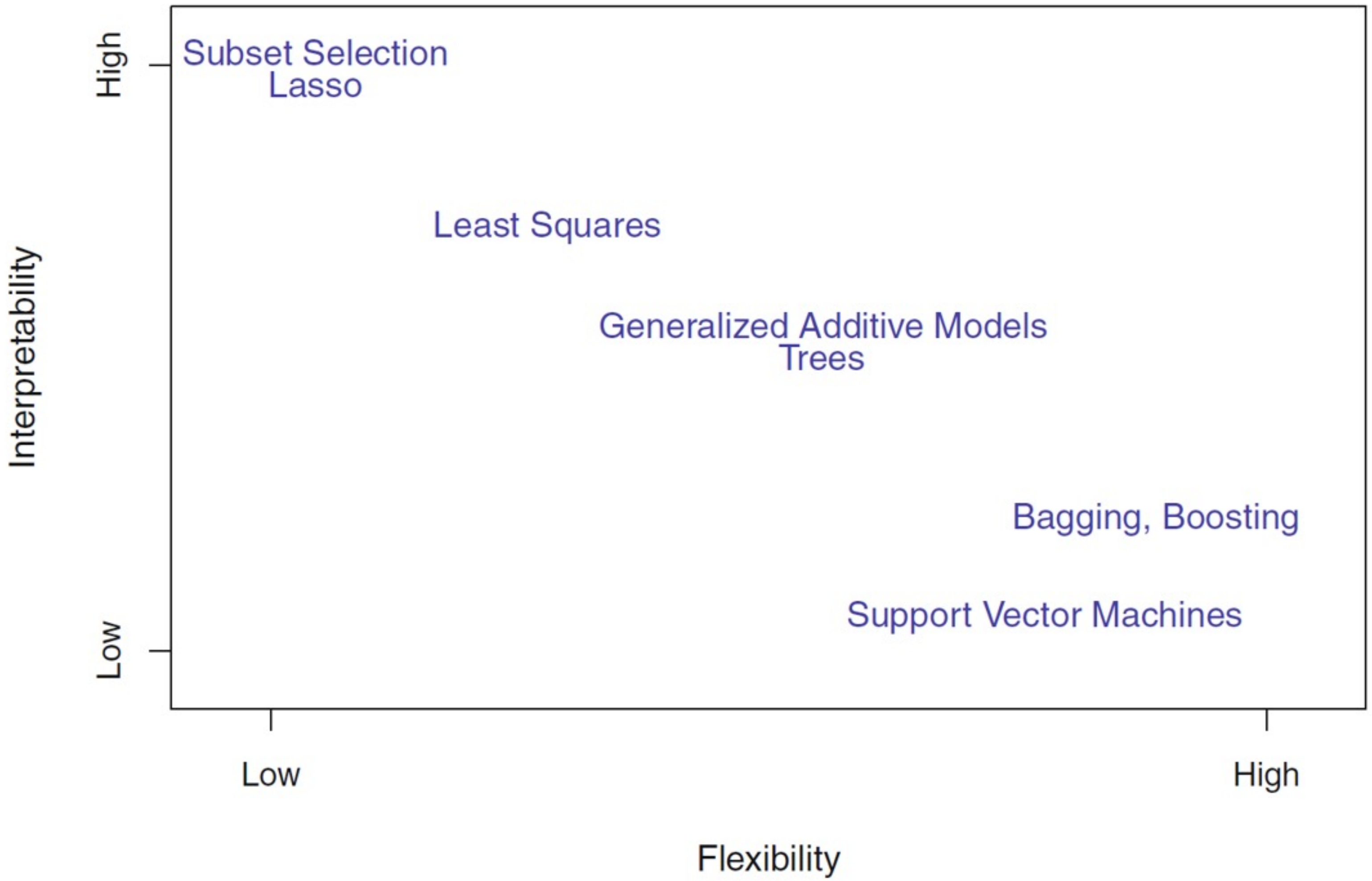
$$E(y_0 - \hat{f}(x_0))^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

- $\mathrm{Var}(\hat{f}(x_0))$: Variance of a model, how much estimates change across different splits of train-test
  - Increases with model complexity/flexibility
- $[\mathrm{Bias}(\hat{f}(x_0))]^2$: Caused by the oversimplification of the model

- $\mathrm{Var}(\epsilon)$: Irreducible error

$Var[\hat{f}(X)]$	$Bias[\hat{f}(X)]$	Implication
High	*Low*	Overfitting
High	*High*	Underfitting
Low	*High*	Underfitting
Low	*Low*	Optimal

Interpretability-Accuracy tradeoff



Data wrangling

- We have already seen enough for how to wrangle the data in Python
- Sometimes, further data work has to be conducted after train-test split.
  - For data wrangling, as well as model fitting and evaluations, we can use **scikit-learn** package
- If you manipulate the data, you have to do it first with train data, then apply the same procedure with test set
  - normalizing
  - create dummy variable using the distribution of data (e.g. 1st quartile)

Build the model

- Basically, it means you estimate the model using the train set
- Sometime, you need to tune **parameters** (e.g. k in KNN regression)
  - Parameters are tuned through multiple trials of different candidate values
  - The best values are usually done through cross-validation in training set

Evaluating the model

- Models are evaluated with some measure, which is usually a function of errors
  - For **continuous outcomes**, errors are difference between the prediction and actual values
    - Measures:
      - Mean absolute error (MAE)
      - Mean squared error (MSE)
      - Root mean squared error (RMSE)
  - For **categorical outcomes,** errors are the misclassification (see next)
- The evaluation is conducted with *test data*