QTM 347: Machine Learning

Lecture 1: Preliminaries in machine learning

Ruoxuan Xiong

Lecture plan

- Preliminaries in machine learning
    - Parametric and nonparametric methods
    - Training/test data and training/test MSE

Supervised and unsupervised machine learning

- **Supervised machine learning** (main focus of this course)
  - **Data**: $(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$
    - X_i: predictors
    - Y_i: response
  - **Task**: Fit a model that relates response to predictors
    - E.g., linear regression or logistic regression model from your regression analysis class
    - You will learn many more in this course

- **Unsupervised machine learning**
  - **Data**: $X_1, X_2, \cdots, X_n$
  - **Task**: Understand the relationships between variables/observations

Supervised machine learning

- **Illustrative example**: Prediction of housing values in suburbs of Boston

- **Training dataset**: given a training dataset that contains n samples

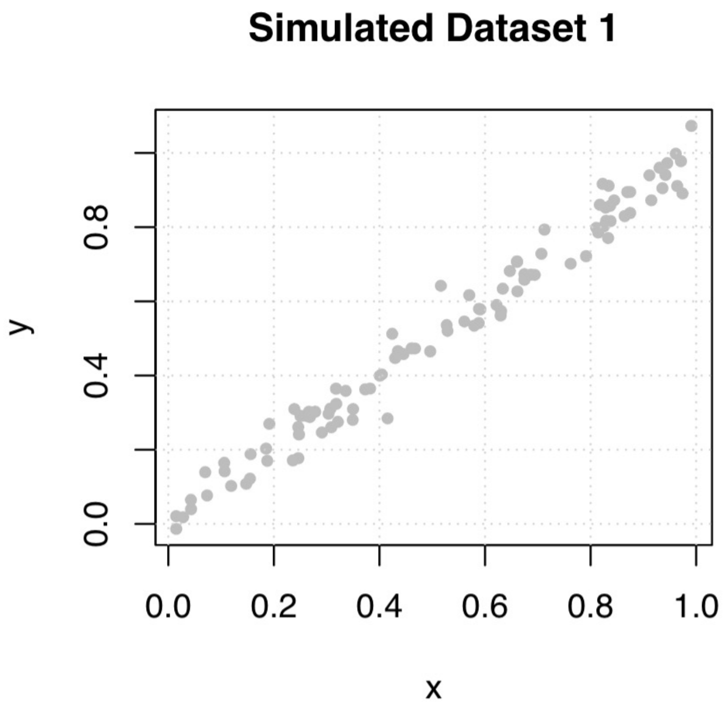
$$(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)$$

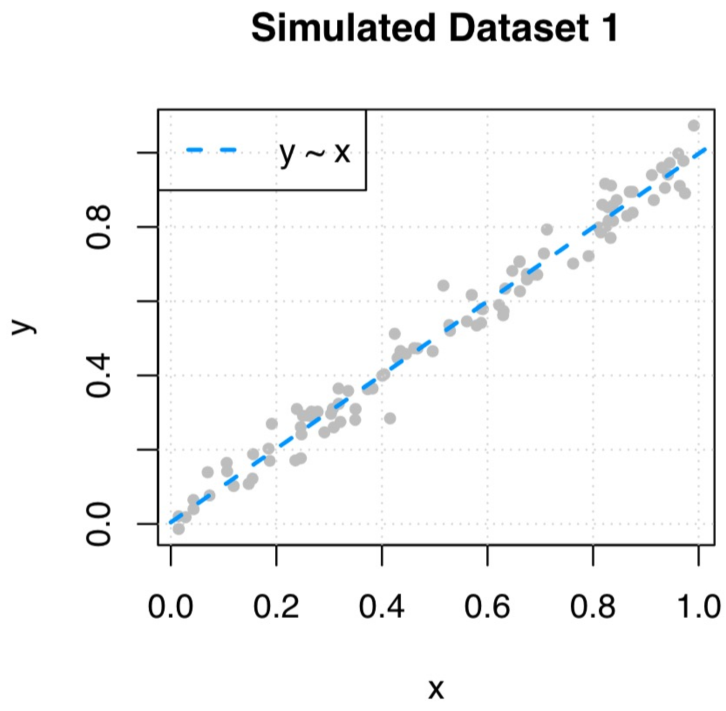
  - X_i is a feature vector
  - Y_i is a label

- Supervised machine learning finds a function f that maps X to Y
  - $Y = f(X) + \varepsilon$, where ε has mean 0
  - f can be quite general, but is **unknown**

Supervised machine learning: How do we estimate f?

- Supervised machine learning finds a function f that maps X to Y
- We may first look at the scatterplot for the exploratory analysis

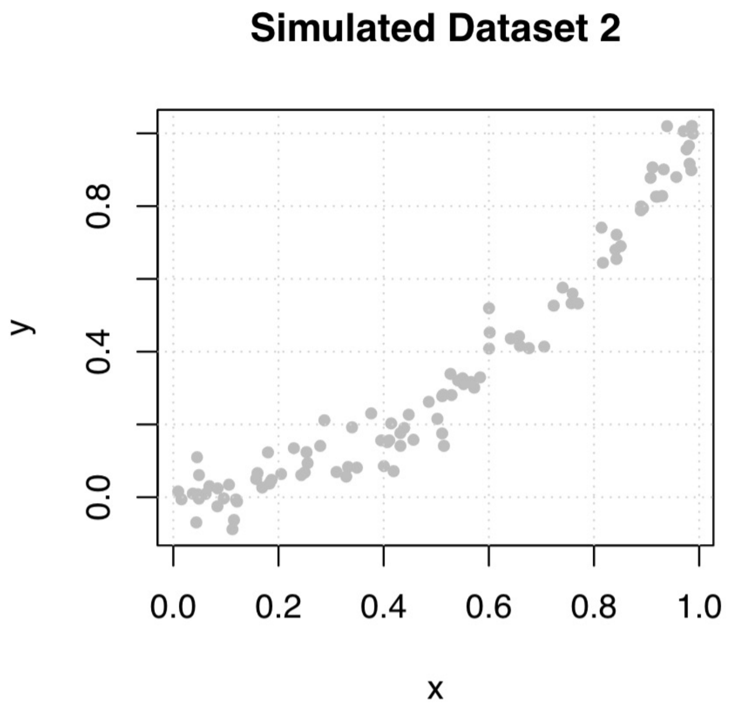


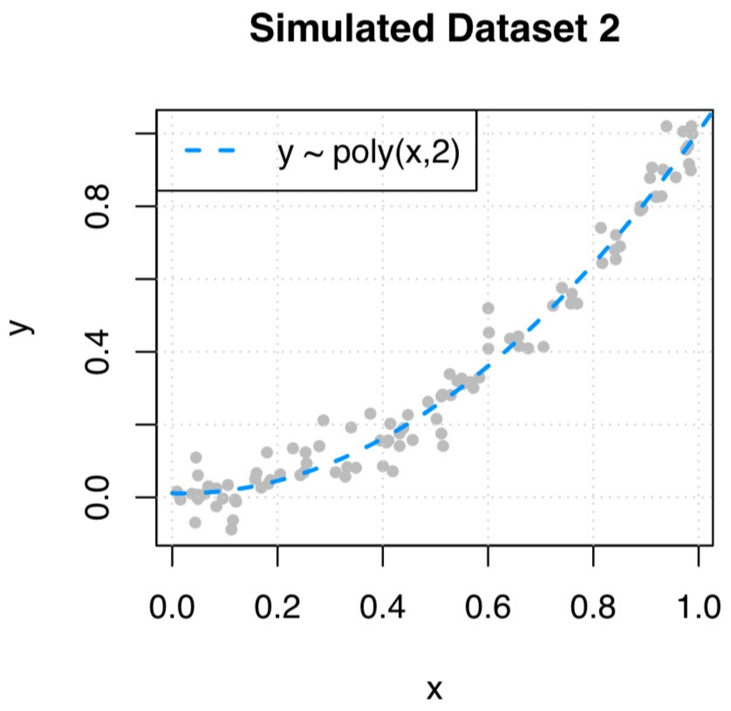


$$Y = \beta_0 + X \cdot \beta_1 + \varepsilon$$

Supervised machine learning: How do we estimate f?

- Supervised machine learning finds a function f that maps X to Y
- We may first look at the scatterplot for the exploratory analysis





$$Y = \beta_0 + X \cdot \beta_1 + X^2 \cdot \beta_2 + \varepsilon$$

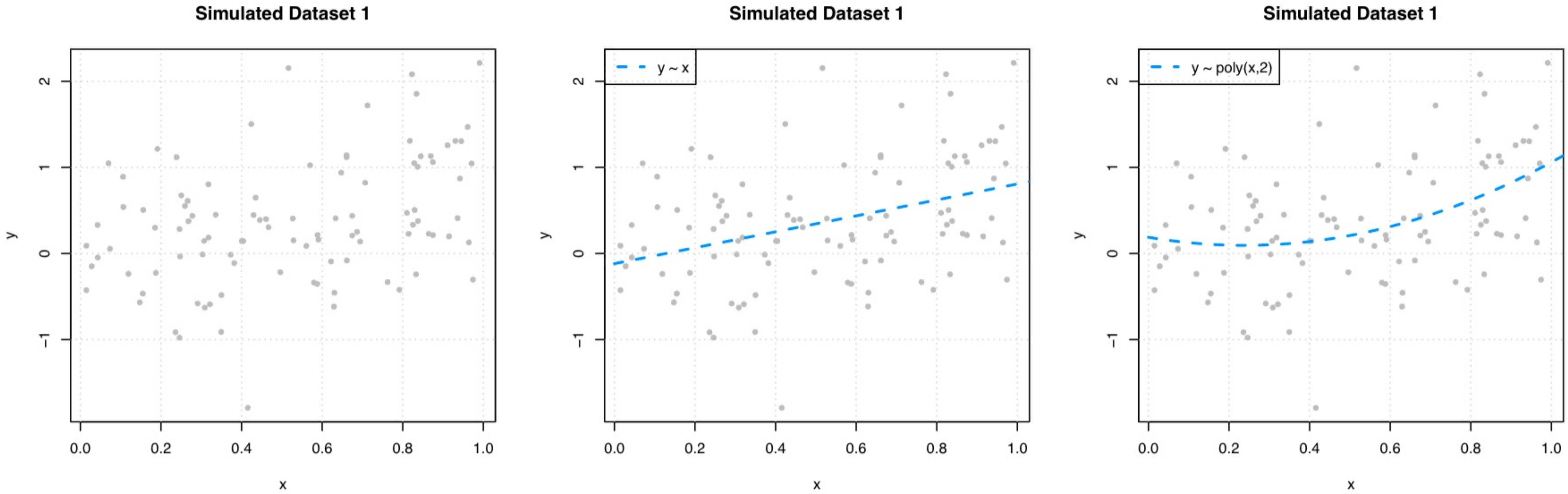
Parametric methods

- We assume that f takes a specific form. For example,
  - $Y = \beta_0 + X \cdot \beta_1 + \varepsilon$
  - $Y = \beta_0 + X \cdot \beta_1 + X^2 \cdot \beta_2 + \varepsilon$

- We use the training data, (X_1, Y_1), (X_2, Y_2), $\cdots$, (X_n, Y_n), to *fit* the parameters

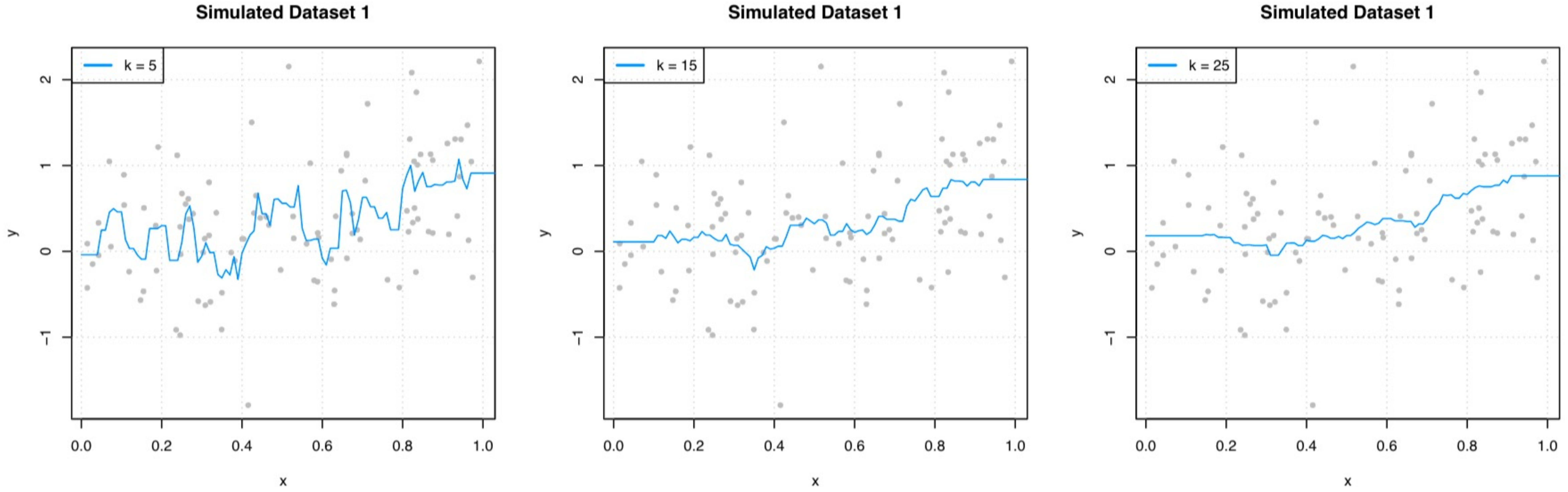
A more complicated case…

- From the scatterplot, which f should we choose?



Nonparametric methods

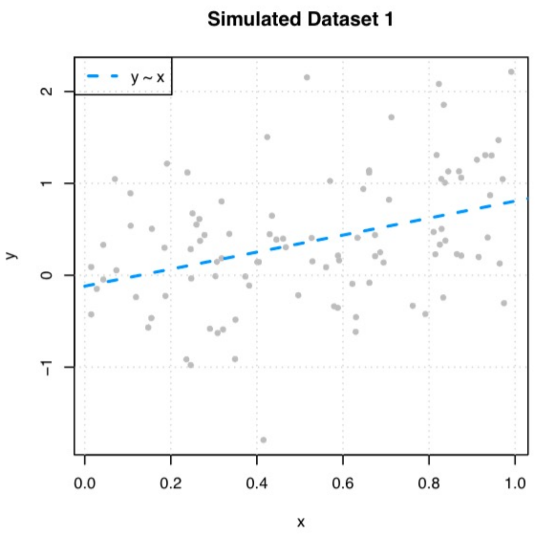
- We don't make any assumptions on the form of f, but we restrict how "rough" the function can be
  - For example, k-nearest neighbors (KNN)

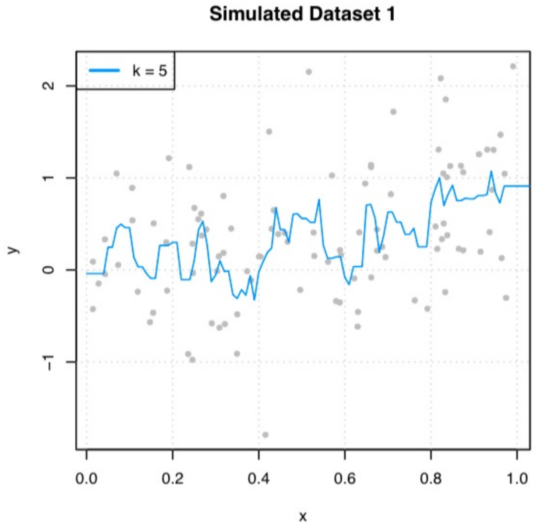


Roughest | Smoothest

Parametric vs nonparametric methods

- **Parametric methods** are often simpler to interpret, but strongly rely on assumptions and can be less flexible to capture complex data patterns
- **Nonparametric methods** rely on fewer assumptions, are flexible and suitable for large datasets





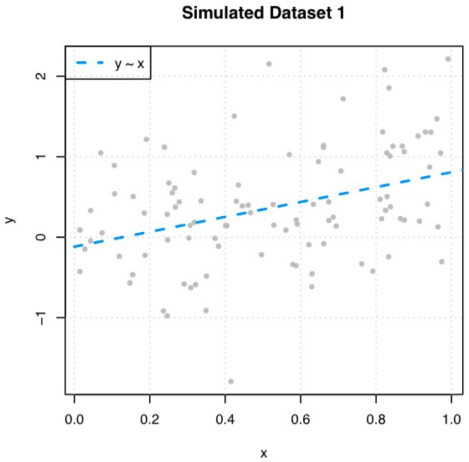
In practice, which model we should use?

- Linear model, quadratic model, nonparametric model, or some other model?

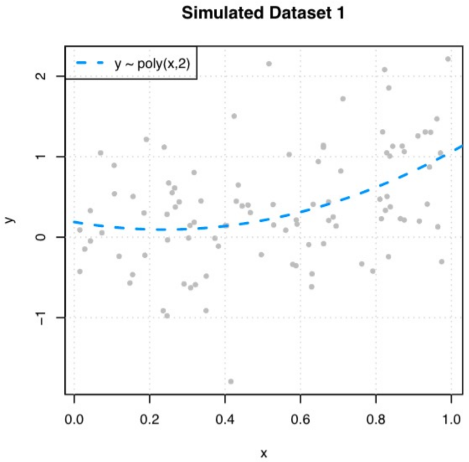
- We need an evaluation metric…

- From the regression analysis class, we could use R^2 (goodness of fit)
  - $R^2 = 1 - \frac{\sum_i (Y_i - \hat{Y}_i)^2}{\sum_i (Y_i - \bar{Y})^2}$
  - $\hat{Y}_i$ is the fitted Y_i and $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$
    - In linear regression, $\hat{Y}_i = \hat{\beta}_0 + X_i \cdot \hat{\beta}_1$
    - For a more general fitted function $\hat{f}$ (e.g., quadratic), $\hat{Y}_i = \hat{f}(X_i)$
  - Interpretation of R^2: Fraction of the variance of Y_i captured by $\hat{f}(X_i)$. The larger the R^2, the better $\hat{Y}_i$ fits Y_i
  - *Quiz: Can R^2 be less than zero? Can R^2 be larger than one?*

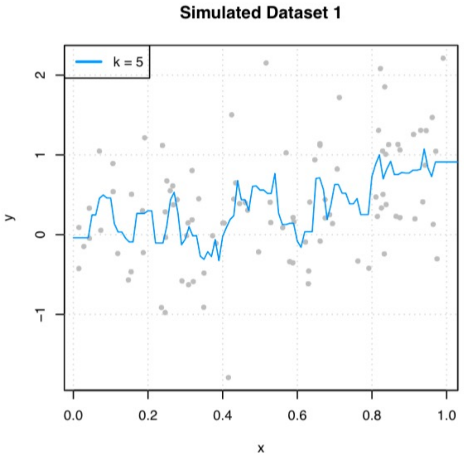
Example of R^2



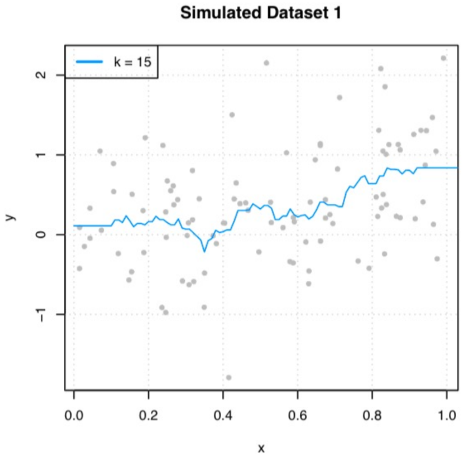
$R^2 = 0.138$



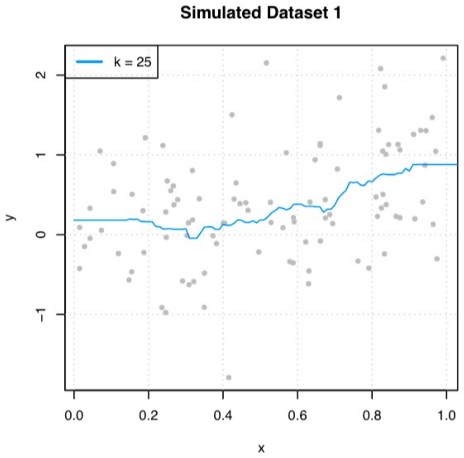
$R^2 = 0.166$



$R^2 = 0.305$



$R^2 = 0.191$

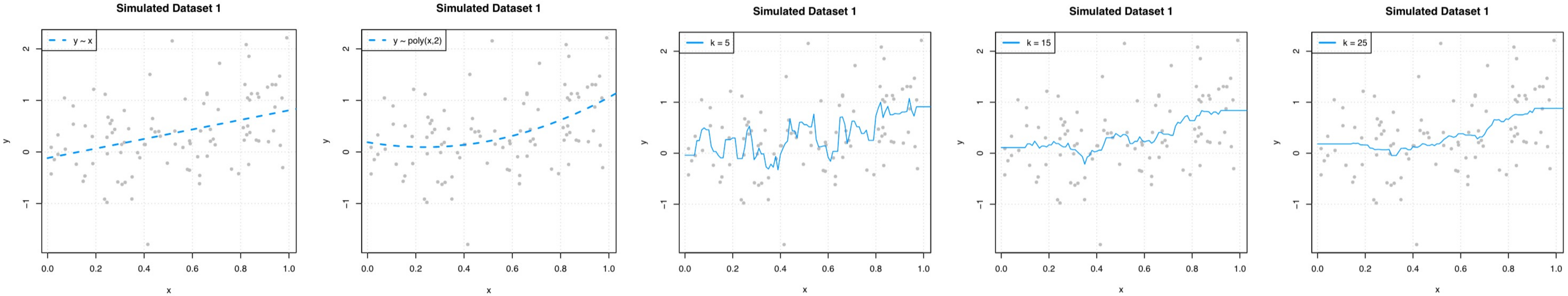


$R^2 = 0.158$

Mean-squared error (MSE)

- MSE and RMSE are commonly used in machine learning
  - $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}(X_i))^2$
  - $\text{MSE} \geq 0$
  - If $\hat{f}(X_i)$ is very close to Y_i for all i, then MSE would be small
  - $R^2 = 1 - \frac{\sum_i (Y_i - \hat{Y}_i)^2}{\sum_i (Y_i - \bar{Y})^2}$ is the standardized version of MSE
  - Root Mean-Squared Error (RMSE) is $\sqrt{\text{MSE}}$

MSE and RMSE



$MSE = 0.439$	$MSE = 0.425$	$MSE = 0.354$	$MSE = 0.412$	$MSE = 0.429$
$RMSE = 0.663$	$RMSE = 0.652$	$RMSE = 0.595$	$RMSE = 0.642$	$RMSE = 0.655$

MSE, RMSE and R^2

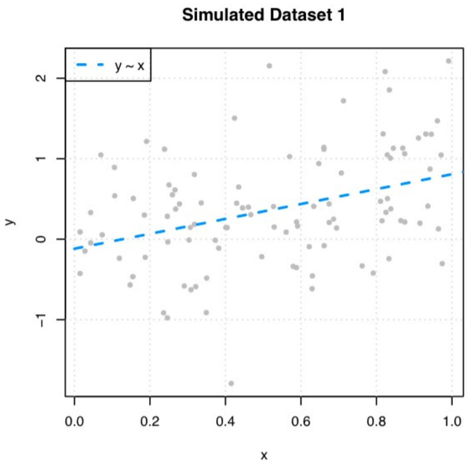
- Given the training data,(X_1, Y_1), (X_2, Y_2), $\cdots$, (X_n, Y_n), there is a one-to-one mapping between MSE, RMSE and R^2

- When is each metric used?
- **MSE**: (a) used in model training because it is mathematically simpler and differentiable; (b) used in theoretical analysis

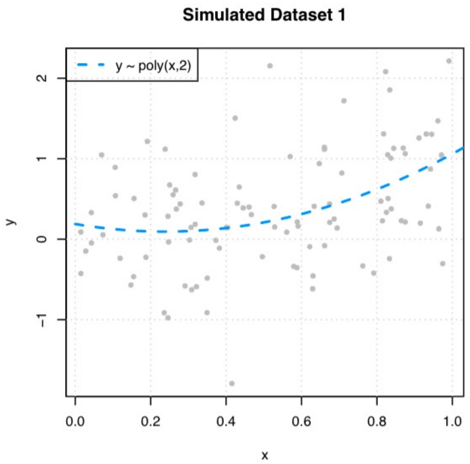
- **RMSE**: Used in performance reporting because it reflects error in original data scale

- $\boldsymbol{R^2}$: A scale-independent metric, used when audience is familiar with "percentage of variance explained"

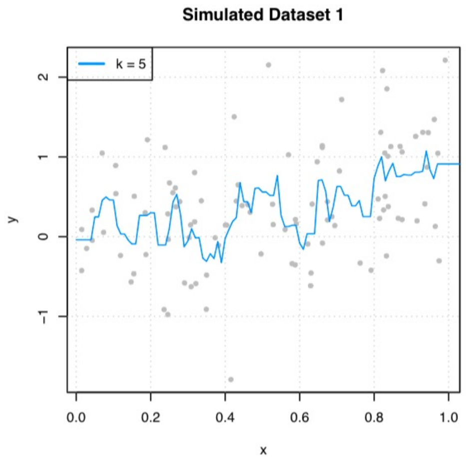
One-to-one mapping between MSE, RMSE and R^2



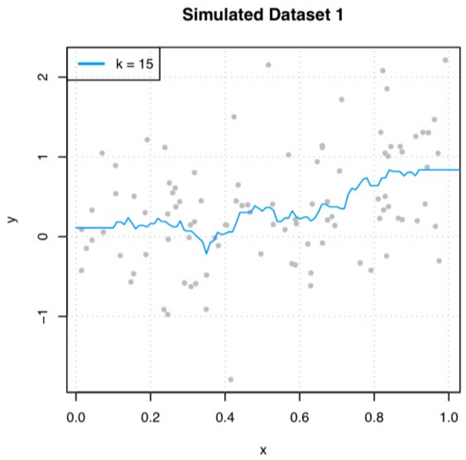
$$\text{MSE} = 0.439$$
$$\text{RMSE} = 0.663$$
$$R^2 = 0.138$$



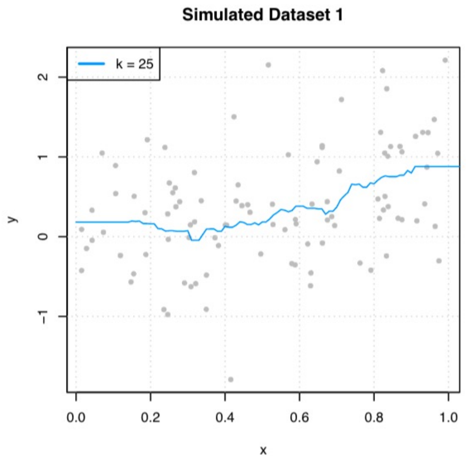
$$\text{MSE} = 0.425$$
$$\text{RMSE} = 0.652$$
$$R^2 = 0.166$$



$$\text{MSE} = 0.354$$
$$\text{RMSE} = 0.595$$
$$R^2 = 0.305$$



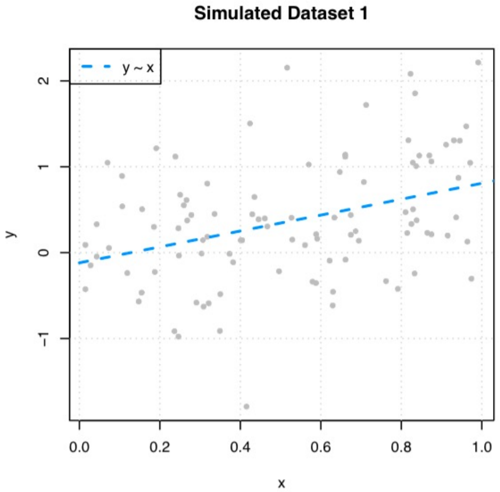
$$\text{MSE} = 0.412$$
$$\text{RMSE} = 0.642$$
$$R^2 = 0.191$$



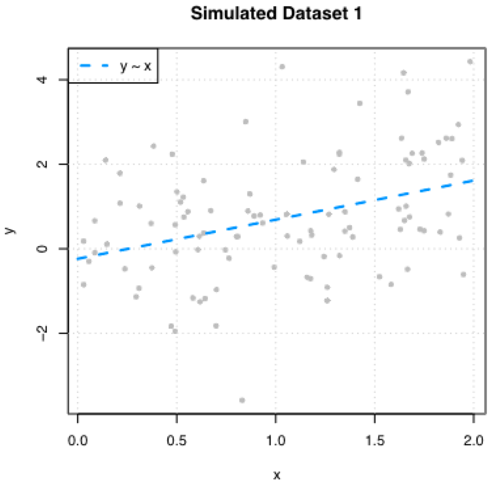
$$\text{MSE} = 0.429$$
$$\text{RMSE} = 0.655$$
$$R^2 = 0.158$$

Scaling of MSE, RMSE and R^2

If we multiply x_i and y_i by 2



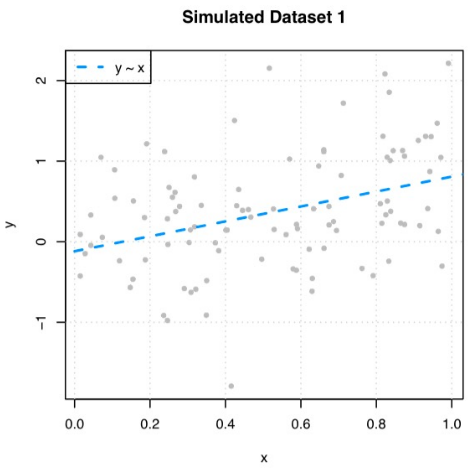
$$\text{MSE} = 0.439$$
$$\text{RMSE} = 0.663$$
$$R^2 = 0.138$$



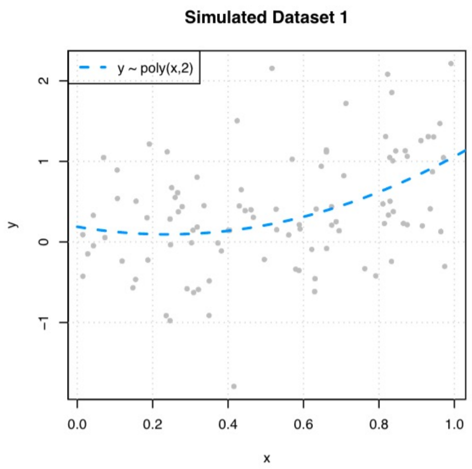
$$\text{MSE} = 1.756$$
$$\text{RMSE} = 1.325$$
$$R^2 = 0.138$$

Which model to use for prediction?

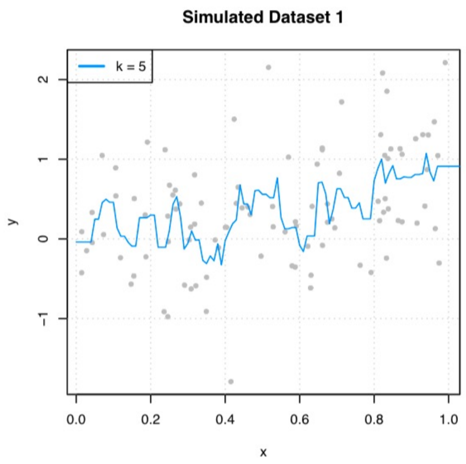
- Suppose we have m new units
  - Their predictors are $X_1', X_2', \cdots, X_m'$
  - We want to predict the outcome of these m units
  - *Quiz: Which model should we use? Shall we choose the one with minimum MSE?*



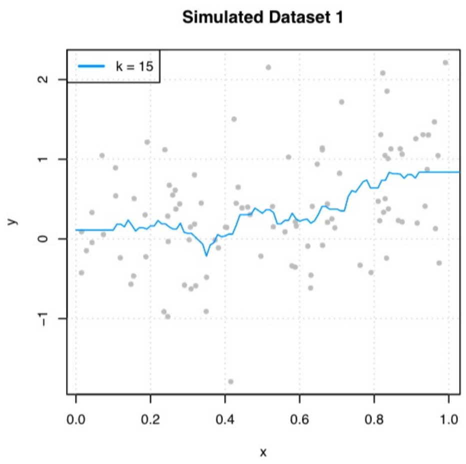
$\mathrm{MSE} = 0.439$
$\mathrm{RMSE} = 0.663$
$R^2 = 0.138$



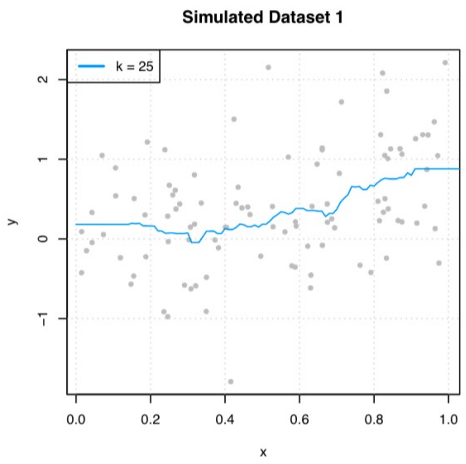
$\mathrm{MSE} = 0.425$
$\mathrm{RMSE} = 0.652$
$R^2 = 0.166$



$\mathrm{MSE} = 0.354$
$\mathrm{RMSE} = 0.595$
$R^2 = 0.305$



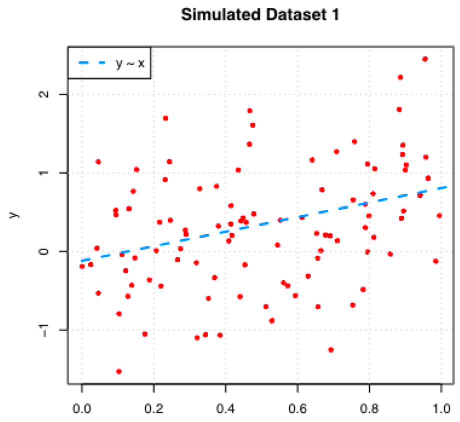
$\mathrm{MSE} = 0.412$
$\mathrm{RMSE} = 0.642$
$R^2 = 0.191$



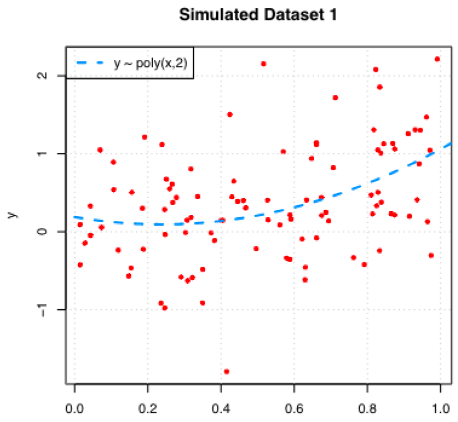
$\mathrm{MSE} = 0.429$
$\mathrm{RMSE} = 0.655$
$R^2 = 0.158$

Which model to use for prediction?

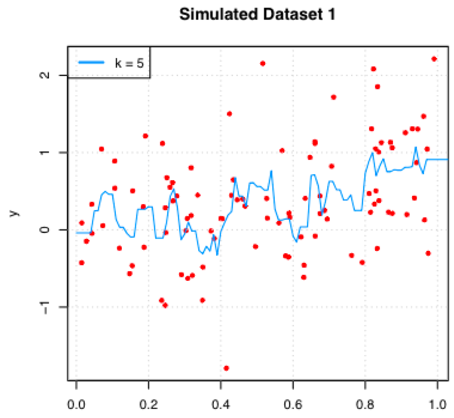
- Suppose we have m new units
  - Their predictors are $X_1', X_2', \cdots, X_m'$
  - The fitted outcomes are $\hat{Y}_1', \hat{Y}_2', \cdots, \hat{Y}_m'$
- Suppose we are clairvoyants, and know the true outcome $Y_1', Y_2', \cdots, Y_m'$
  - We can calculate $\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(Y_i' - \hat{f}(X_i'))^2$



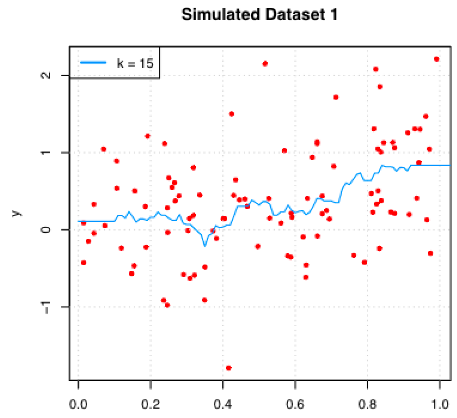
MSE = 0.533



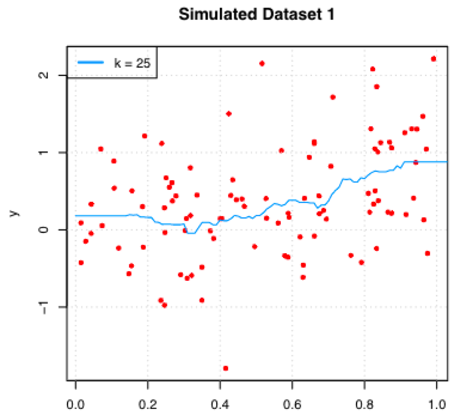
MSE = 0.518



MSE = 0.564



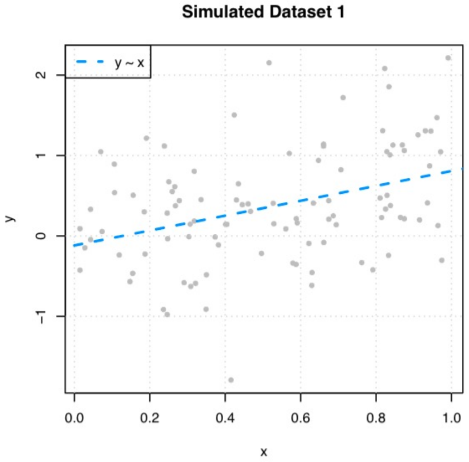
MSE = 0.507



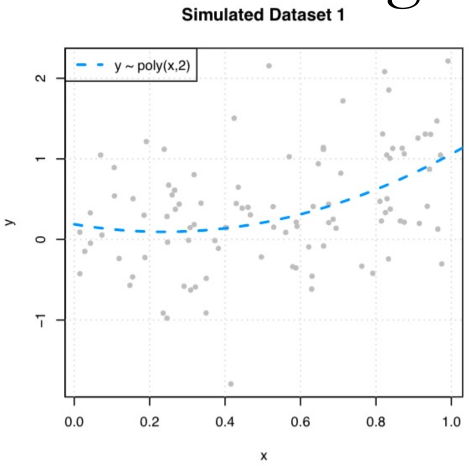
MSE = 0.524

A low training MSE does not imply a low test MSE…

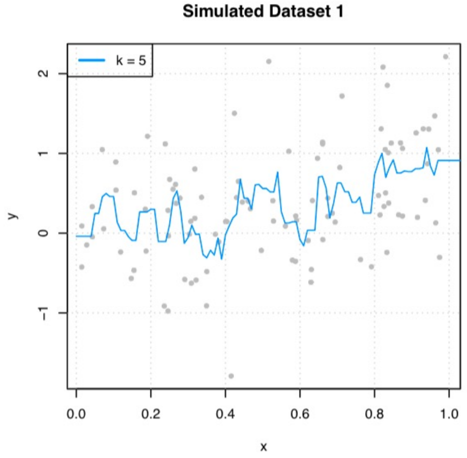
- This is the main challenge in machine learning



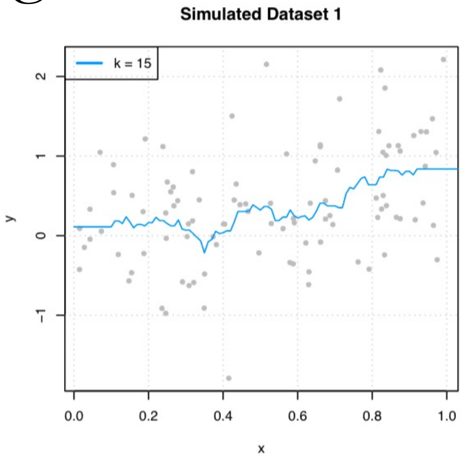
Training MSE = 0.439



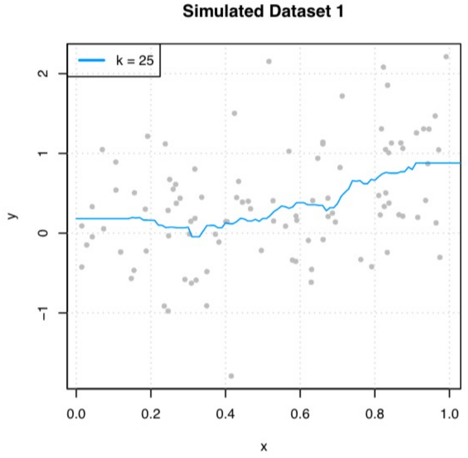
Training MSE = 0.425



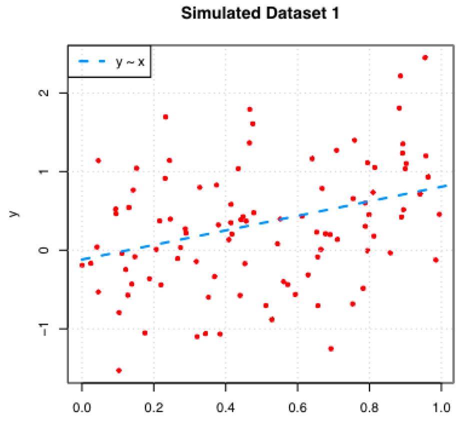
Training MSE = 0.354



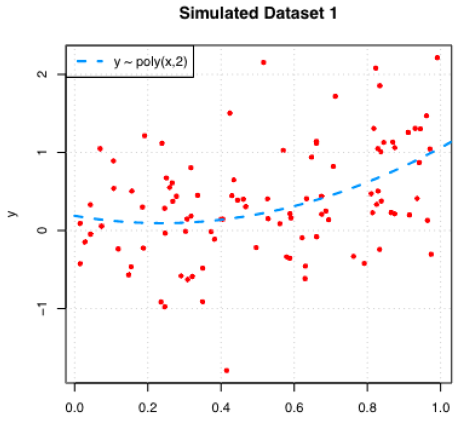
Training MSE = 0.412



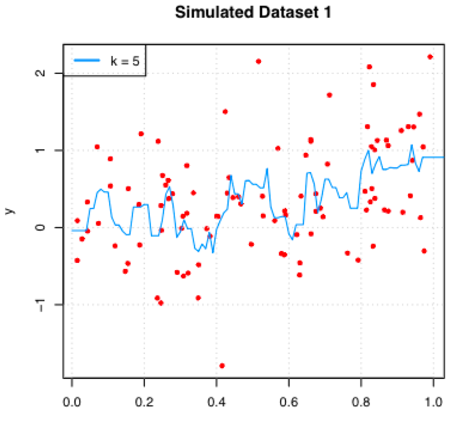
Training MSE = 0.429



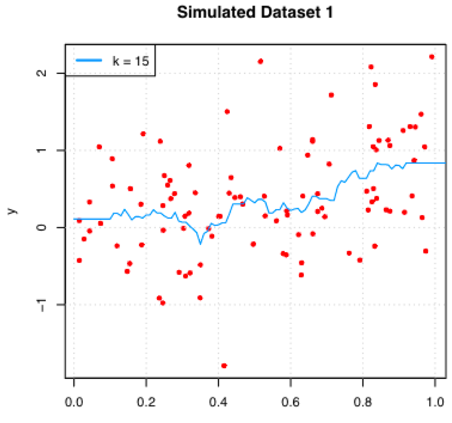
Test MSE = 0.533



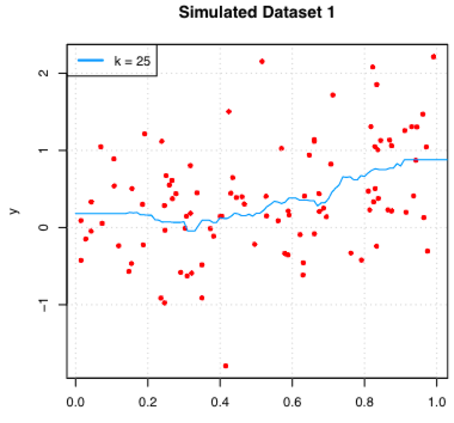
Test MSE = 0.518



Test MSE = 0.564



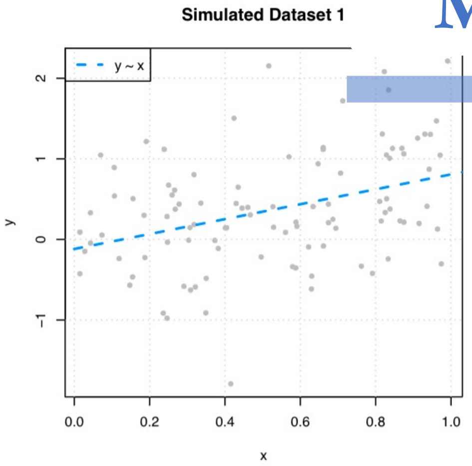
Test MSE = 0.507



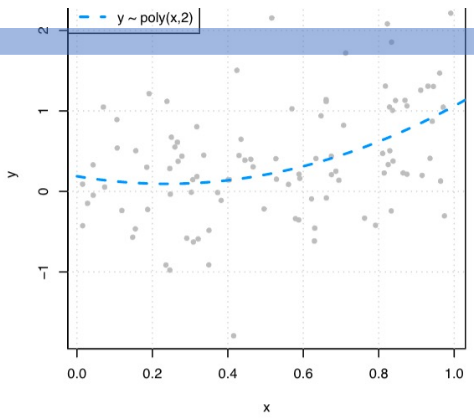
Test MSE = 0.524

MSE varies with model flexibility

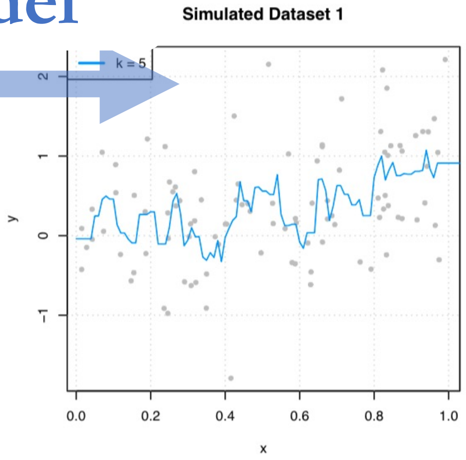
More flexible model



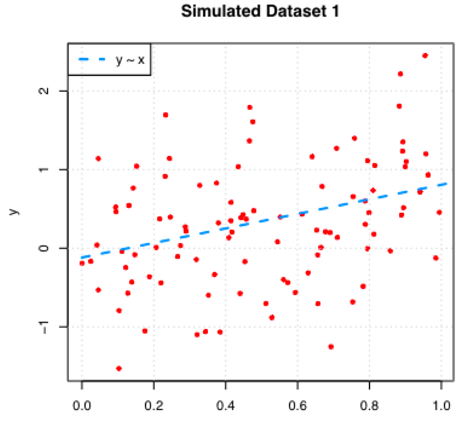
Training MSE = 0.439



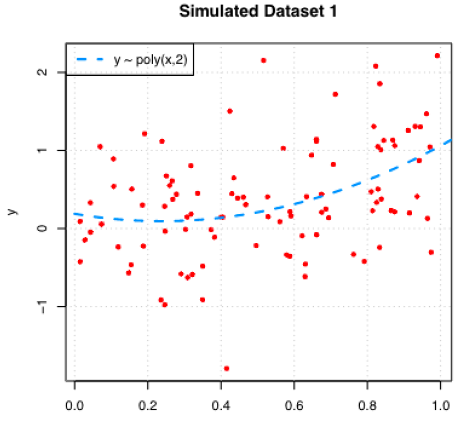
Training MSE = 0.425



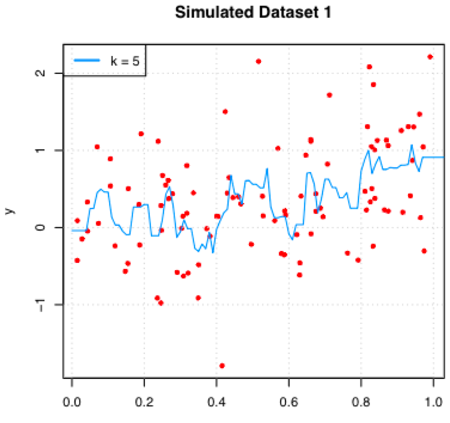
Training MSE = 0.354



Test MSE = 0.533



Test MSE = 0.518



Test MSE = 0.564