CS/ENGR M148 L5:
Model Selection and Cross Validation

Sandra Batista

Administrative News

This week in discussion section:

Lab on regression and cross validation

Project Data Check-in: Your team will need to demonstrate a regression model on your project data.

Projects

1. Projects will be graded on how well they demonstrate mastery of the methods taught in class and discussions.
2. You may choose your own data set or a data set supported by the course staff.
3. Team contract 5% - This week during discussion. A sample contract will be made available. Team contracts can be updated until 11:59 pm PT on 10/11/24
4. Project discussion check-ins: 30%, 6x5%
5. Final project code: 25%
6. Final project report: 40%

Join our slido for the week…

https://app.sli.do/event/fCYNaz1LPznfsUF8YhGeqo

Today's Learning Objectives

Students will be able to:

- Review: Error metrics
- Review: Multiple Linear Regression
- Solve Ordinary Least Squares
- Apply **model selection** and **cross validation** for **overfitting**

The Modeling Process

1. Choose a model	*How should we represent the world?*
2. Choose a loss function	How do we quantify prediction error?
3. Fit the model	How do we choose the best parameters of our model given our data?
4. Evaluate model performance	How do we evaluate whether this process gave rise to a good model?

- Regression: A process for modeling the relationship between variables of interest

Least Squares Loss Function

L2 loss function or squared loss:

$$\frac{1}{n}\sum_{i=1}^{n}(\text{observed response}_i - \text{predicted response}_i)^2.$$

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2.$$

[Yu, Barter 2024]

Measuring Error

How do we measure error?

root mean squared error rMSE

mean absolute error, MAE

median absolute deviation, MAD

Correlation

- The covariance of two random variables *X* and *Y* is in units that are a product of those of *X* and *Y*. To obtain a dimensionless number, the covariance can be divided by the product of the standard deviation of *X* and the standard deviation of *Y*. This is called the ***correlation coefficient***:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}.$$

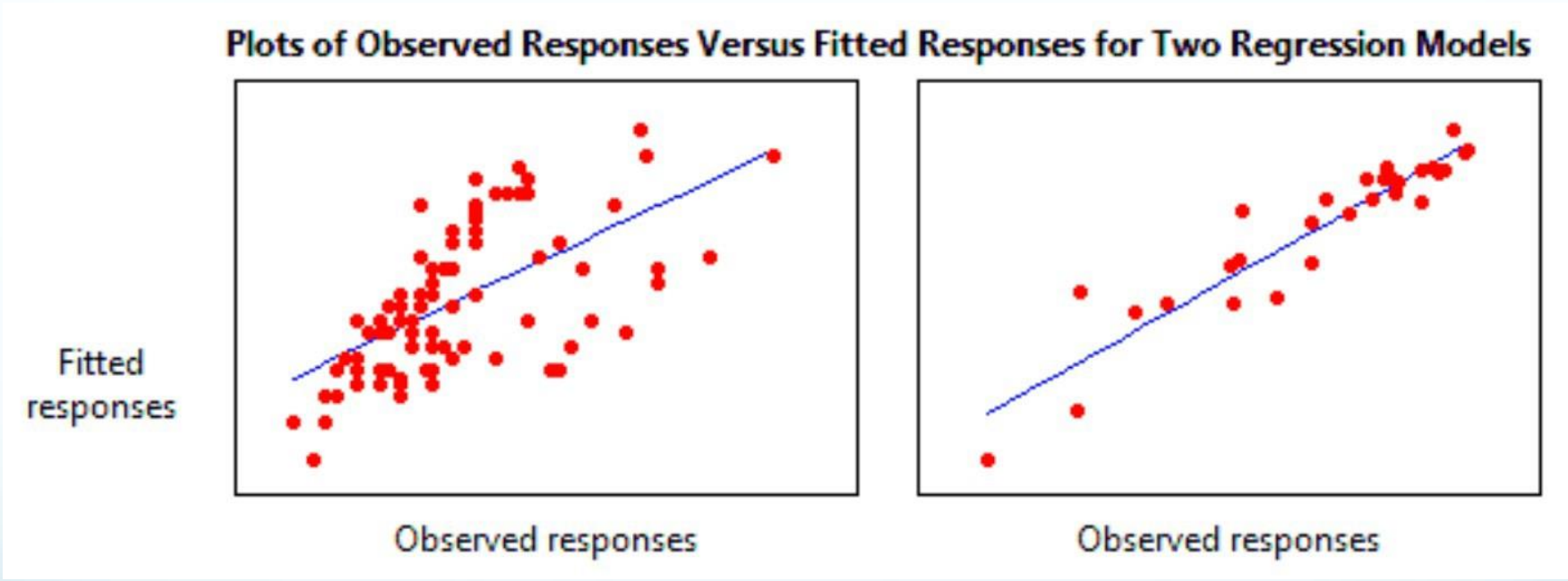
Other names include Pearson's product-moment correlation coefficient, Pearson's coefficient, or Pearson's correlation.

R-squared

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$

- *This will be 0 if model is as good as mean*
- *This will be 1 if model is perfect*
- *This can be negative if the model is worst than the average. This can happen when we evaluate the model on the test set.*

R-squared



$R^7 = 0.38$

$R^7 = 0.87$

Today's Learning Objectives

Students will be able to:

- ✓ Review: Error metrics
- ✗ Review: Multiple Linear Regression
- ✗ Solve Ordinary Least Squares
- ✗ Apply **model selection** and **cross validation** for **overfitting**

From One Feature to Many Features

Dataset for SLR

x	y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_n	y_n

	FG	PTS
1	1.8	5.3
2	0.4	1.7
3	1.1	3.2
4	6.0	13.9
5	3.4	8.9
...	...	...

Dataset for Multiple Linear Regression

$x_{:,1}$	$x_{:,2}$		$x_{:,p}$	y
x_{11}	x_{12}	$\cdots$	x_{1p}	y_1
x_{21}	x_{22}	$\cdots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\cdots$	x_{np}	y_n

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

Matrix Notation

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \mathbb{X} \theta$$

Design matrix with dimensions n x (p + 1)

The Design Matrix $\mathbb{X}$

We can use linear algebra to represent our predictions of all n data points at once.

One step in this process is to stack all of our input features together into a **design matrix**:

$$\mathbb{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

What do the **rows** and **columns** of the design matrix represent in terms of the observed data?

Field Goals, Assists, 3-Point Attempts

Bias	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
1	0.4	0.8	1.5	1.7
1	1.1	1.9	2.2	3.2
1	6.0	1.6	0.0	13.9
1	3.4	2.2	0.2	8.9
...	...	...	...	...
1	4.0	0.8	0.0	11.5
1	3.1	0.9	0.0	7.8
1	3.6	1.1	0.0	8.9
1	3.4	0.8	0.0	8.5
1	3.8	1.5	0.0	9.4

Example design matrix
708 rows x (3+1) cols

Ordinary Least Squares

The **least squares estimate** $\hat{\theta}$ is the parameter that **minimizes** the objective function $R(\theta)$:

$$R(\theta) = \frac{1}{n}(||\mathbb{Y} - \hat{\mathbb{Y}}||_2)^2$$

How should we interpret the OLS problem?

A. Minimize the mean squared error for the linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$

B. Minimize the **distance** between true and predicted values $\mathbb{Y}$ and $\hat{\mathbb{Y}}$

C. Minimize the **length** of the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$

D. All of the above

E. Something else

Ordinary Least Squares

The **least squares estimate** $\hat{\theta}$ is the parameter that **minimizes** the objective function $R(\theta)$:

$$R(\theta) = \frac{1}{n}(||\mathbb{Y} - \hat{\mathbb{Y}}||_2)^2$$

How should we interpret the OLS problem?

A. Minimize the mean squared error for the linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$

B. Minimize the **distance** between true and predicted values $\mathbb{Y}$ and $\hat{\mathbb{Y}}$

C. Minimize the **length** of the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$ Important for today

D. All of the above

E. Something else

Using multiple predictors

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

$$\frac{1}{n}\sum_{i=1}^{n}(\text{observed price}_i - \text{predicted price}_i)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\text{observed price}_i - (b_0 + b_1\text{area}_i + b_2\text{quality}_i + b_3\text{year}_i + b_4\text{bedrooms}_i))^2.$$

[Yu, Barter 2024]

Using multiple predictors

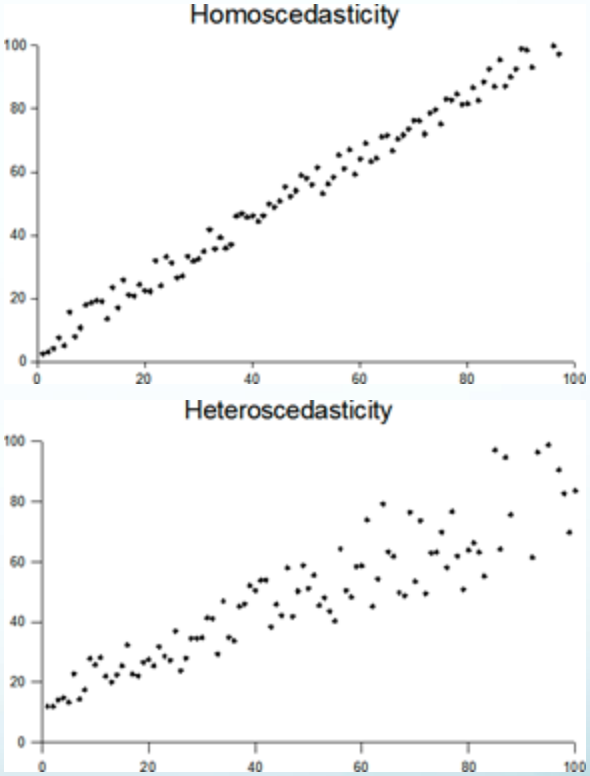
$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

How do we interpret the coefficients?

[Yu, Barter 2024]

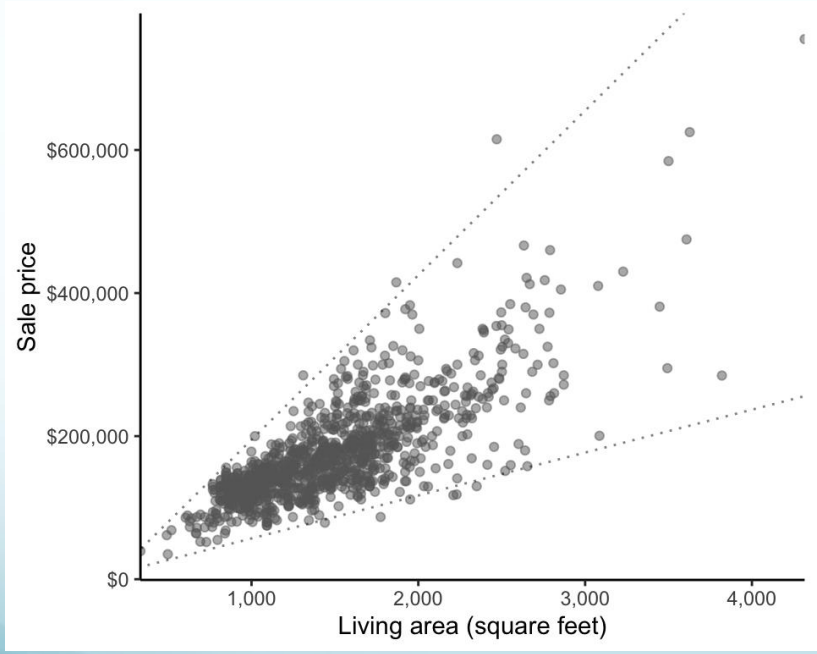
Homoscedasticity and heteroscedasticity

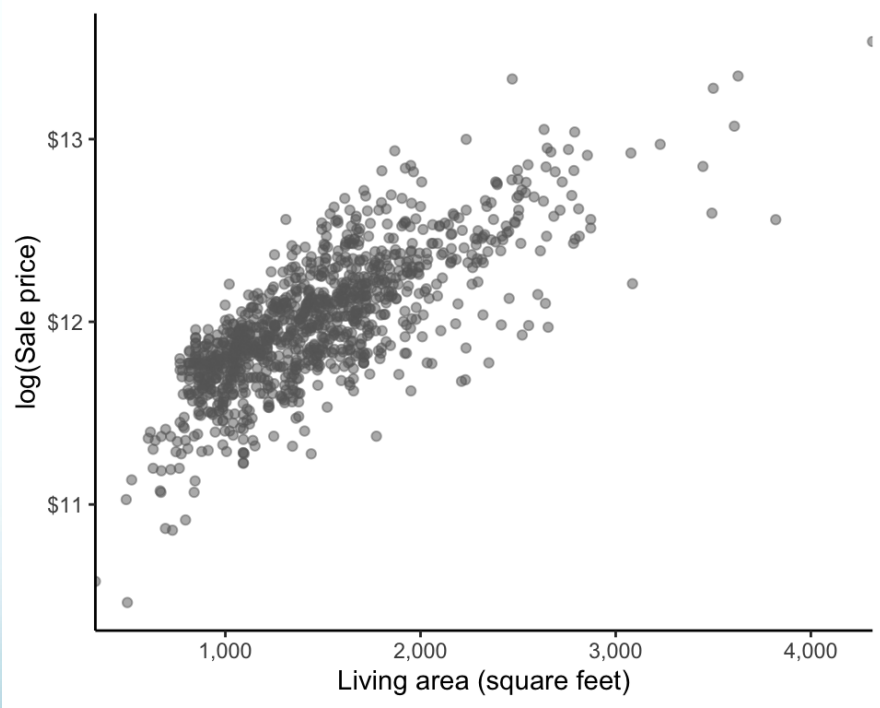
- One of the assumptions for linear regression is *homoscedasticity*.
- Meaning constant variance across all observations, and in particular does not depend on the value of the explanatory variables.
- LS works better when variables have symmetric or Gaussian distributions



heteroscedasticity

- Preprocessing: log transformation of the response variable
- Remember to transform data back after fitting!





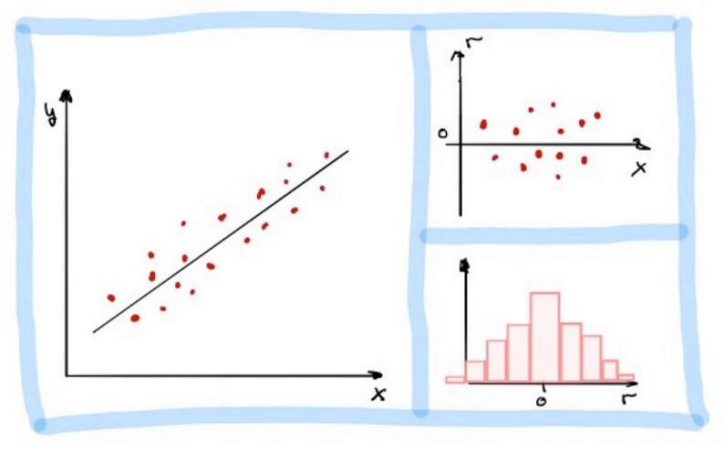
Residual Analysis

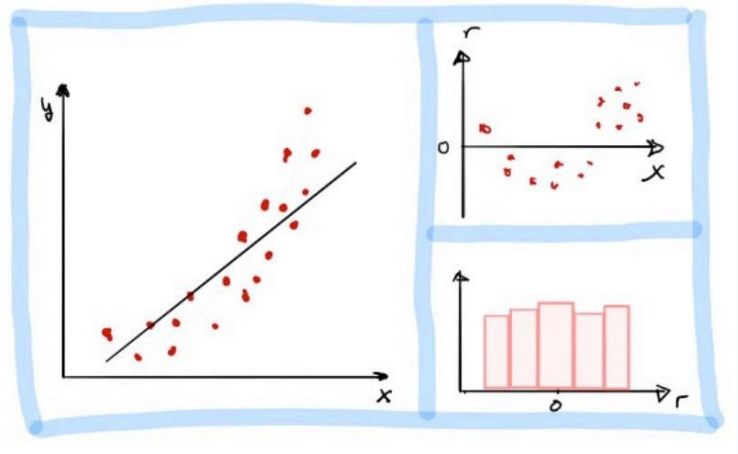
Residuals *are the difference between the predicted and observed values.*

In residual analysis, we typically create two types of plots:

1. *Plot residuals against predictor variable or predicted value in a scatterplot*
2. *Plot histogram of residuals*

Residual Analysis





Linear assumption is correct. There is no obvious relationship between residuals and x. Histogram of residuals is symmetric and normally distributed.

Linear assumption is incorrect. There is an obvious relationship between residuals and x. Histogram of residuals is symmetric but not normally distributed.

Note: For multi-regression, we plot the residuals vs predicted value since there are too many x's and that could wash out the relationship.

Look out for collinearity

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

Collinearity is when predictors are correlated with each other.

[Yu, Barter 2024]

Activity: Demo

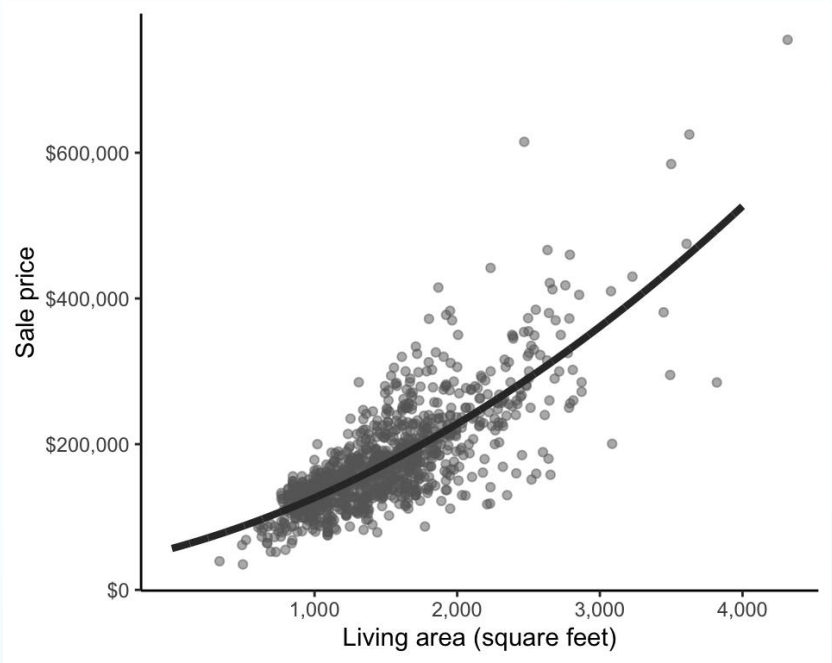
Let's look at error, checking residuals, multiple regression, and collinearity now:

https://colab.research.google.com/drive/1v7VFQhODzza1wNCPcqlQFdusO8Fm2-lm?usp=sharing

[Yu, Barter 2024]

Using polynomial of predictors

This is still a **linear combination of the coefficients and transformed features.**

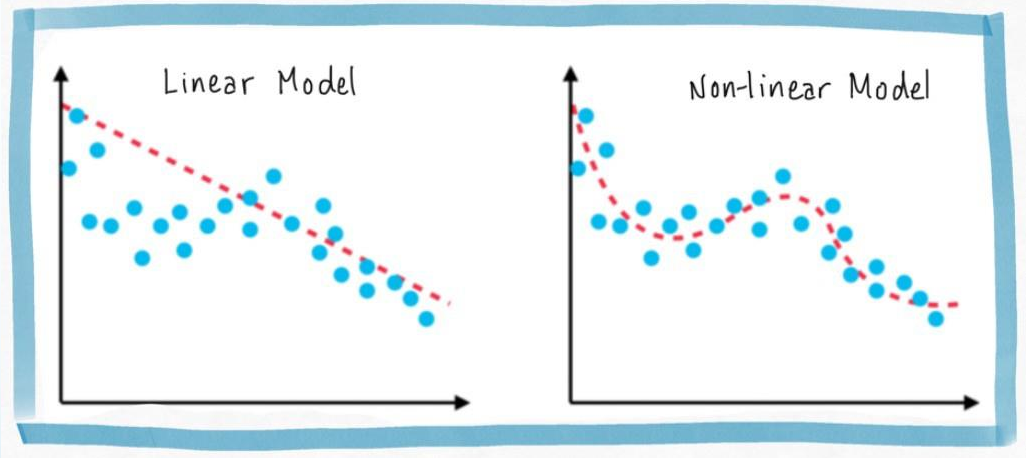


$$\text{predicted price} = b_0 + b_1 \text{area} + b_2 \text{area}^2.$$

[Yu, Barter 2024]

Fitting non-linear data

Multi-linear models can fit large datasets with many predictors. But the relationship between predictor and target isn't always linear.



We want a model:

$$y = f(x)$$

Where f is a non-linear function and β is a vector of the parameters of f.

Polynomial Regression

This looks a lot like multi-linear regression where the predictors are powers of x!

Multi-Regression

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,J} \\ 1 & x_{2,1} & \dots & x_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,J} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix},$$

Poly-Regression

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^M \\ 1 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^M \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}.$$

Model Training

Give a dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, we find the optimal polynomial model:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_M x^M$$

1. We transform the data by adding new predictors:

$$\tilde{x} = [1, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_M]$$

where $\tilde{x}_k = x^k$

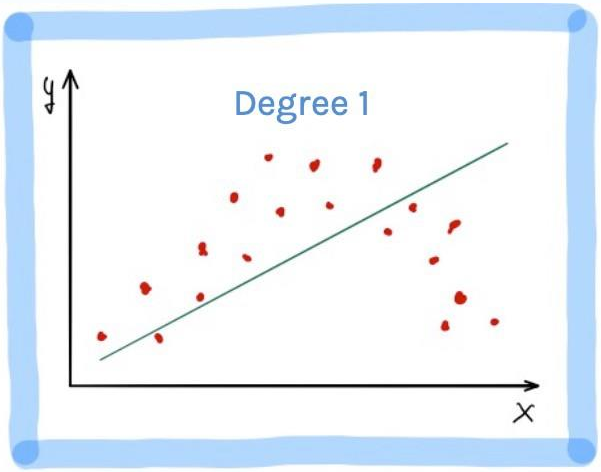
2. Fit the parameters by minimizing the MSE using vector calculus. As in multi-linear regression:

$$\widehat{\boldsymbol{\beta}} = (\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}})^{-1} \tilde{\boldsymbol{X}}^T \boldsymbol{y}$$

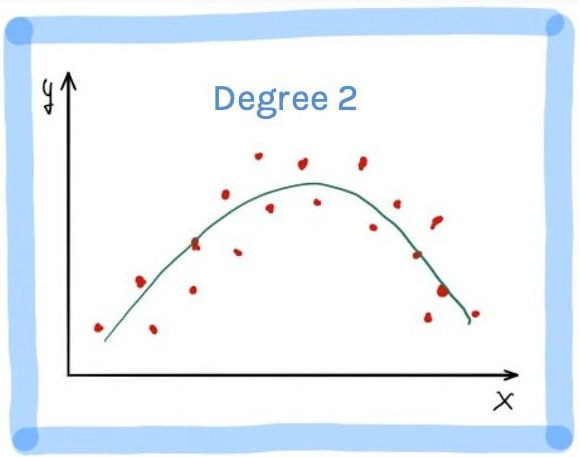
It is a good idea to standardize variables before polynomial regression
Use: sklearn StandardScaler

Polynomial Regression (cont)

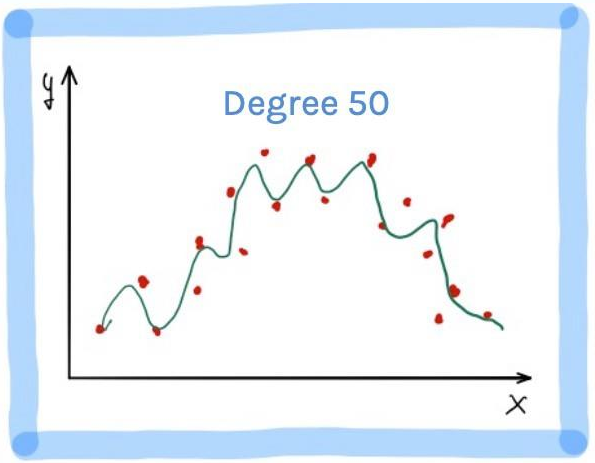
Fitting a polynomial model requires choosing a degree.



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.

Today's Learning Objectives

Students will be able to:

- ✓ Review: Error metrics
- ✓ Review: Multiple Linear Regression
- ✗ Solve Ordinary Least Squares
- ✗ Apply **model selection** and **cross validation** for **overfitting**

Solving the least squares problem

- Method: Linear algebra.

$$y = X\beta + \epsilon.$$

- Find the value β that minimizes $(\|X\beta - Y\|_2)^2$; the minimal β is denoted $\hat{\beta}$.

- Using orthogonality, the solution emerges naturally as the normal equation: $(X^TX)^{-1}X^TY$.

- Overdetermined: more rows than columns – common for OLS
- Underdetermined: more columns than rows – infinitely many or no solutions

Solving the least squares problem

- Method: Linear algebra.

$$y = X\beta + \epsilon.$$

^

- In practice: inversion not often used, but singular value decomposition (SVD) is and is efficient

Solving the least squares problem

- Method: Linear algebra.

$$y = X\beta + \epsilon.$$

^

- In practice: inversion not often used, but singular value decomposition (SVD) is and is efficient

Solving the least squares problem

- Method 1: (Multivariable) calculus.

$$f(m,b) = \sum_{i=1}^{n} (y_i - (mx_i + b))^2.$$

- The task is to minimize *f(m,b)* over the parameters *m* and *b*. The square is helpful because the derivatives of *f* with respect to *m* and *b* are linear.
- Compute the two partial derivatives (with respect to *m* and *b*), set them equal to 0, …
- Gradient descent is used for this when we cannot solve analytically, but we'll come back to this later..

Today's Learning Objectives

Students will be able to:

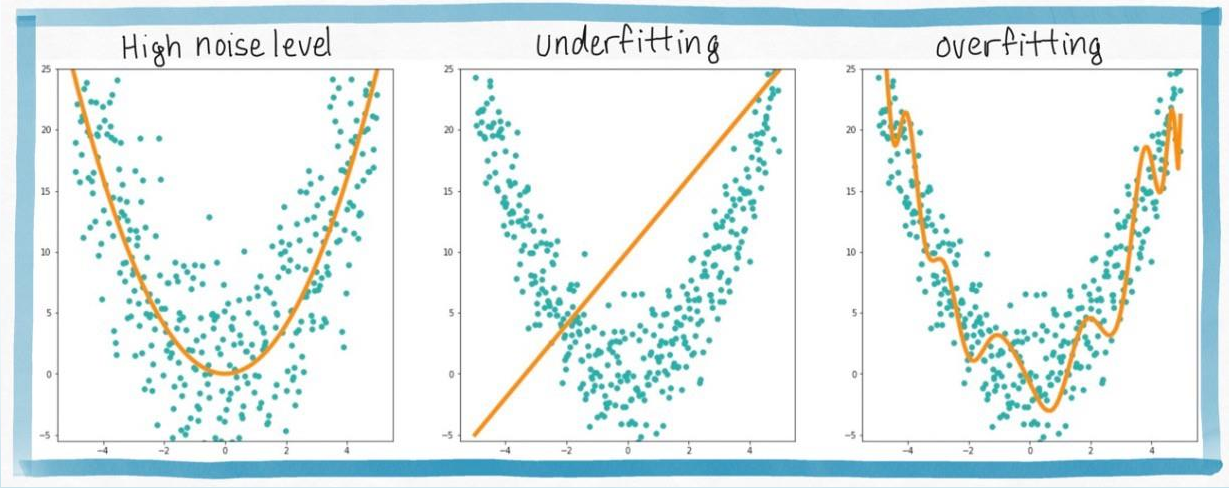
- ✓ Review: Error metrics
- ✓ Review: Multiple Linear Regression
- ✓ Solve Ordinary Least Squares
- ✗ Apply **model selection** and **cross validation** for **overfitting**

Test Error and Generalization

We know to evaluate models on both train and test data because models can do well on training data but do poorly on new data.

When models do well on new data is called generalization.

There are at least three ways a model can have a high test error.



Irreducible and Reducible Errors

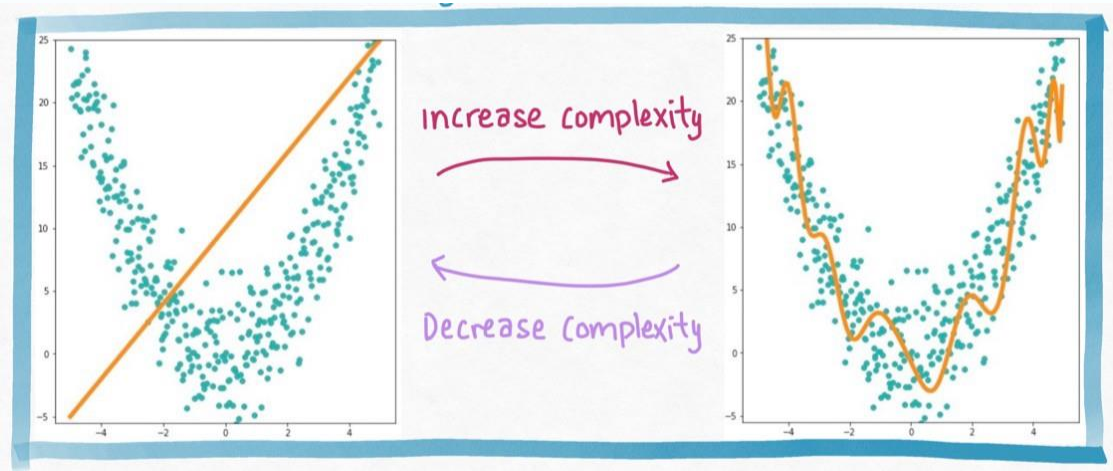
We distinguished the contributions of noise to the generalization error:

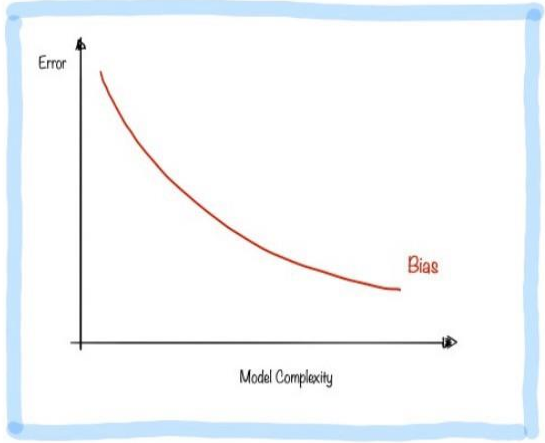
Irreducible error: we can't do anything to decrease error due to noise.

Reducible error: we can decrease error due to overfitting and underfitting by improving the model.

Underfitting and Overfitting

*Reducible error comes from either **underfitting** or **overfitting**. There is a trade-off between the two sources of errors:*

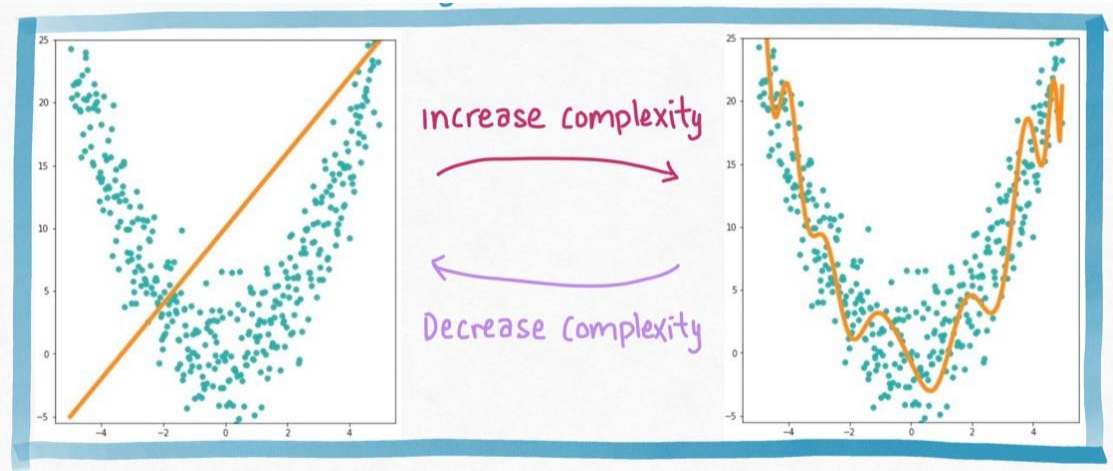


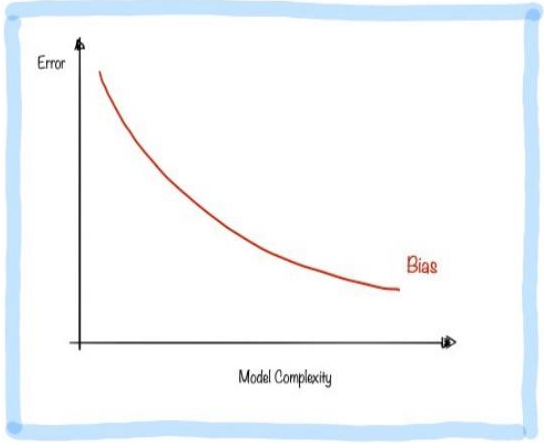


Underfitting and Overfitting

Underfitting *is when a model performs poorly on the training and testing data sets*

Overfitting *is when model performs well on training data set, but poorly on the testing data set.*

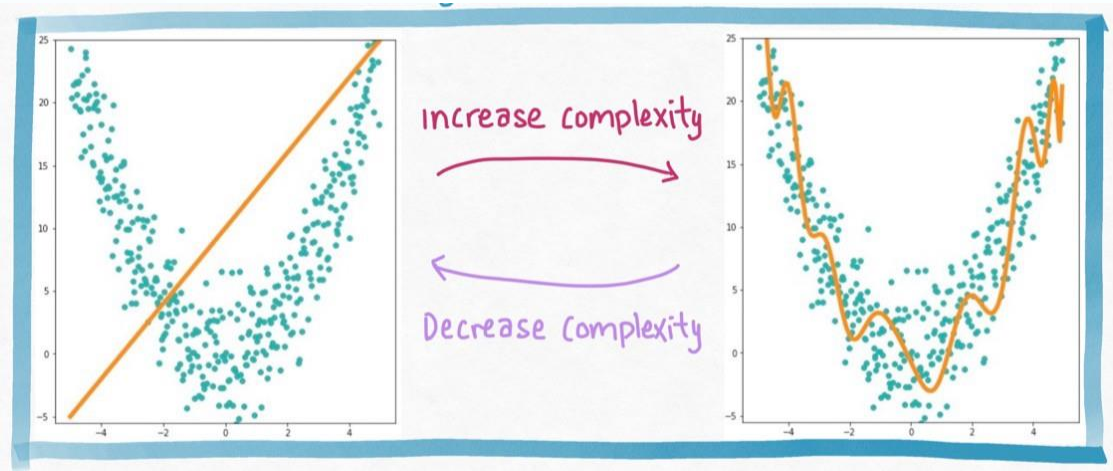


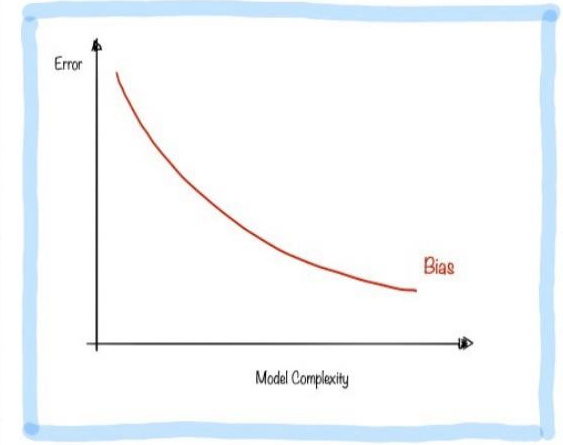


Bias Variance Tradeoff

Underfitting *occurs when there is* ***high bias***

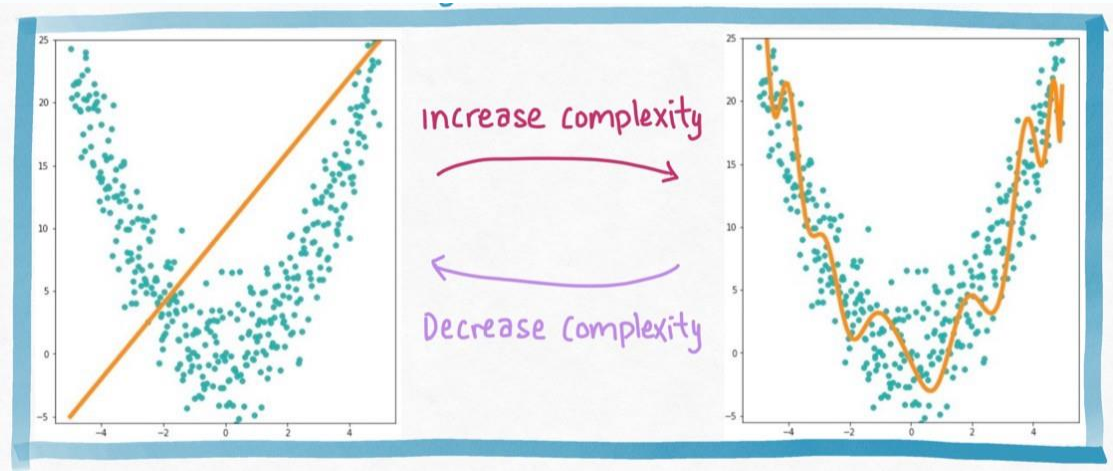
Overfitting *occurs when there is* ***high variance***

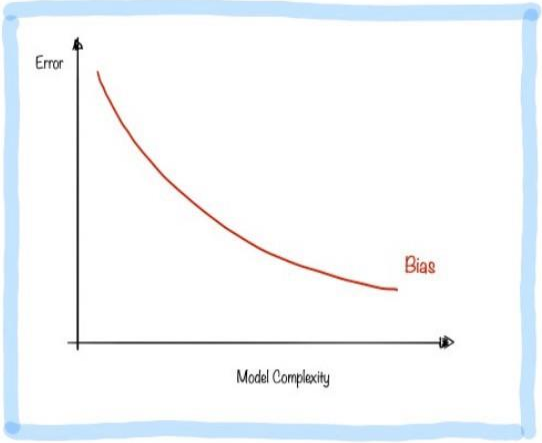




Bias

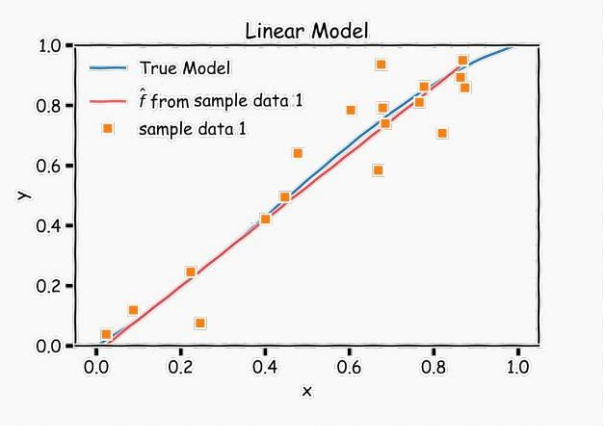
Bias is the distance between expected value of estimator and parameter we want to estimate

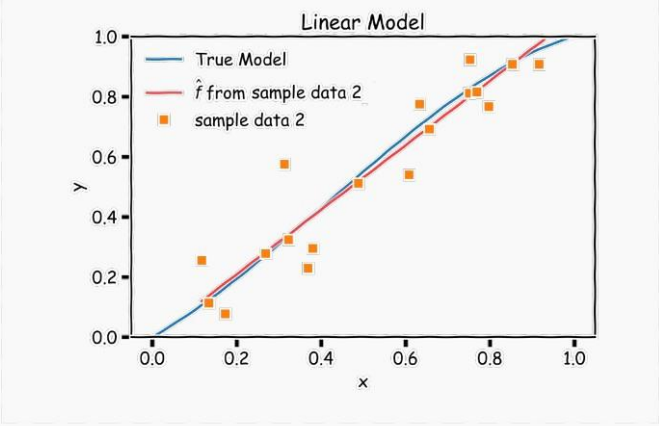


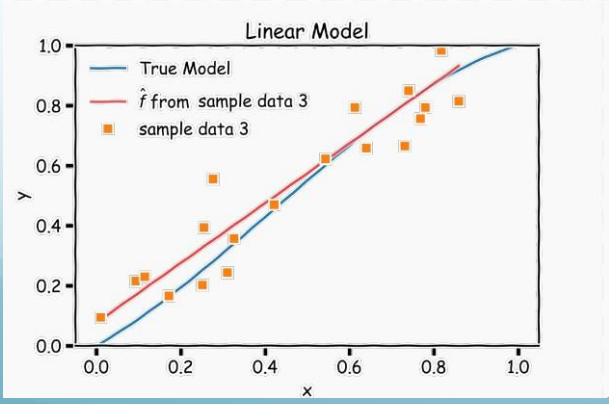


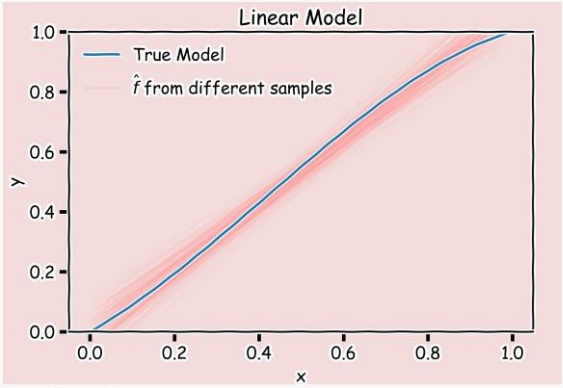
Variance

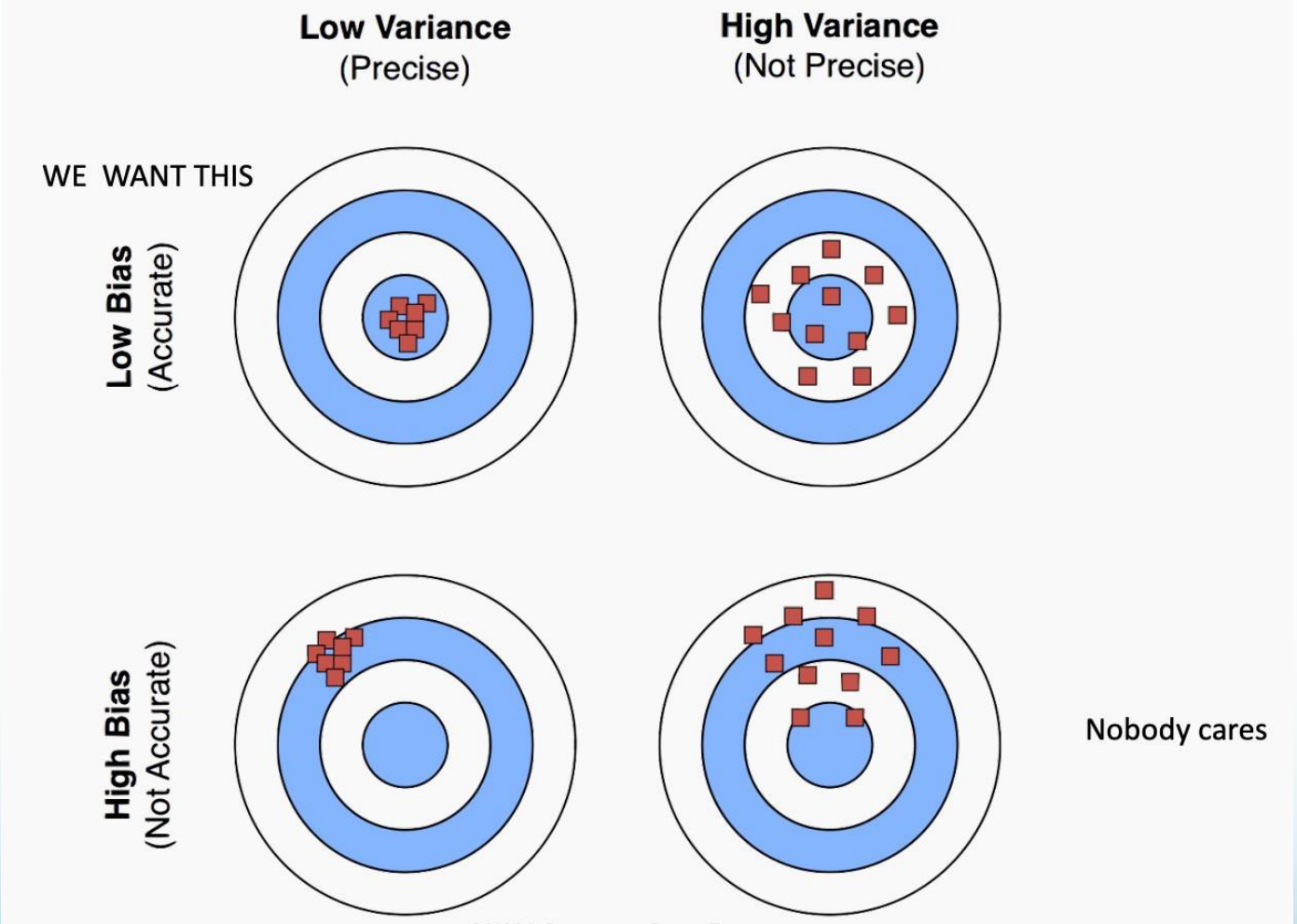
Variance measures how the predictions will vary over training sets.










Low Variance
(Precise)
High Variance
(Not Precise)
WE WANT THIS
Low Bias
(Accurate)
High Bias
(Not Accurate)
Nobody cares

Overfitting

Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

So far, we have seen that overfitting can happen when:

- *Too many parameters*
- *Degree of the polynomial is too large*
- *Too many interaction terms*
- *Number of samples used in training or validating*

Overfitting

Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

Ways to address:

1. *Model selection: Limiting the number of parameters in model*
2. *Using more validation data sets*

Next, we will see other evidence of overfitting, which will point to a way of avoiding overfitting: Ridge and Lasso regressions.

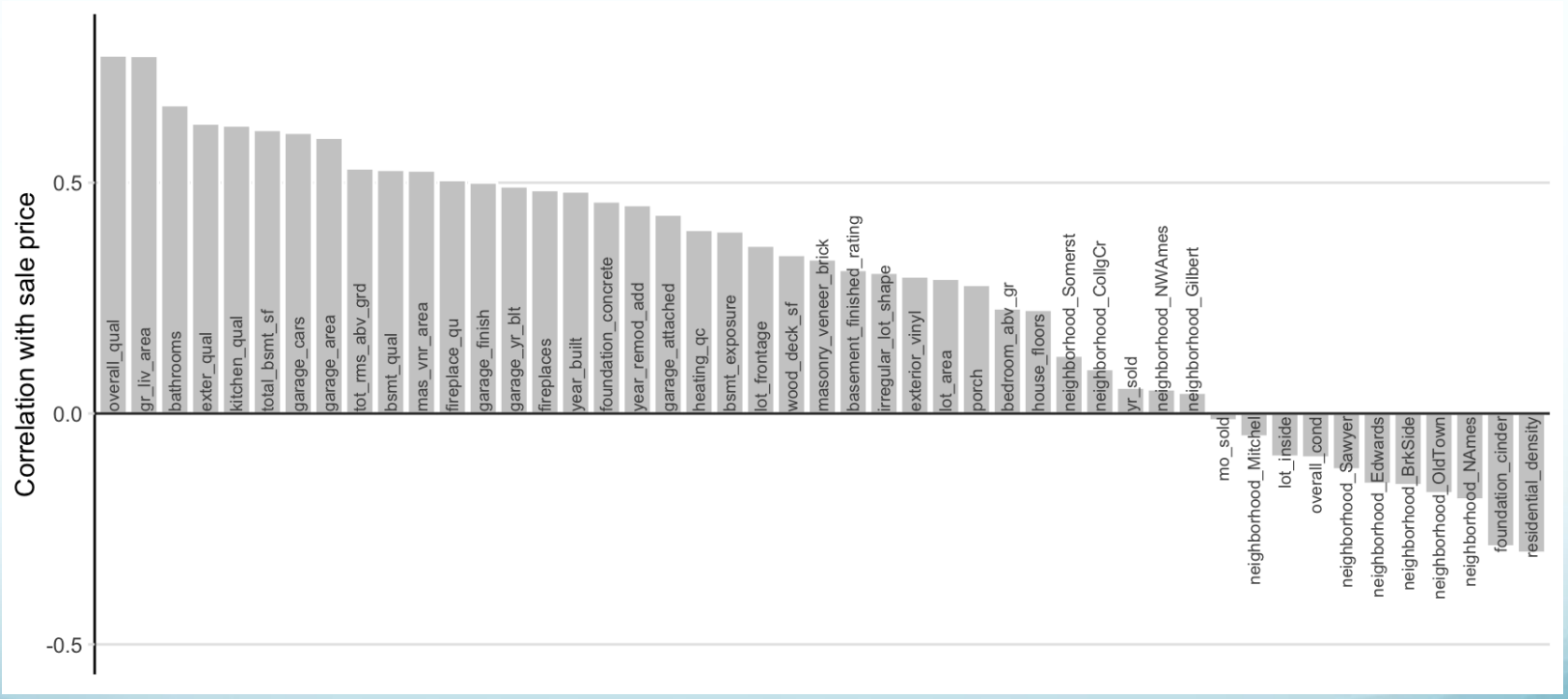
Model Selection

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

Example: 3 predictors *(X_1, X_2, X_3)*

EDA for house prices

Correlation screening: keep only the predictive features that are most correlated with the response



[Yu, Barter 2024]

Stepwise Variable Selection and Cross 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 iteratively building up our set.

1. Start with the empty set P0, construct the null model M0.

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/BIC on training set)

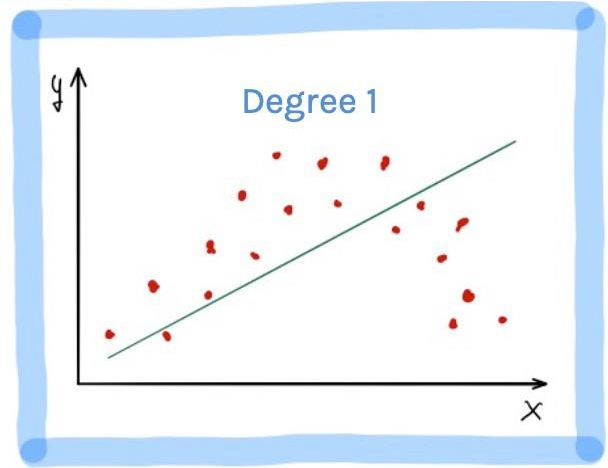
Stepwise Variable Selection Computational Complexity

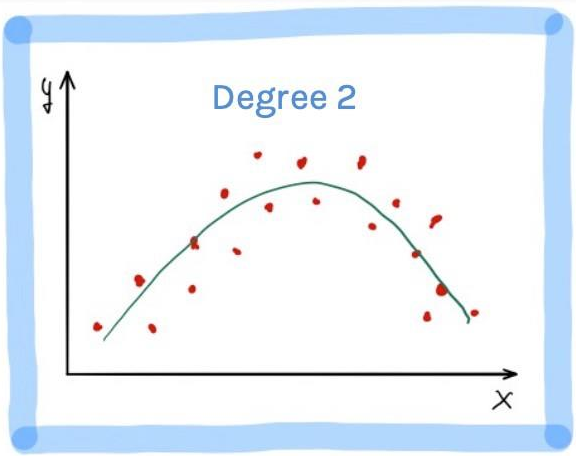
How many models did we evaluate in the worse case?

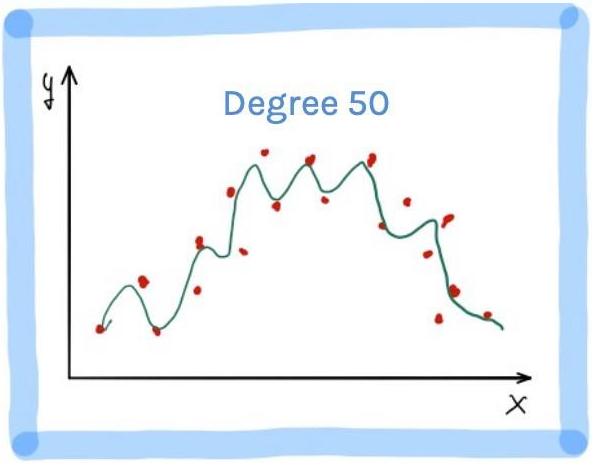
Is this tractable for many features?

Choosing the degree of the polynomial model

Fitting a polynomial model requires choosing a degree.





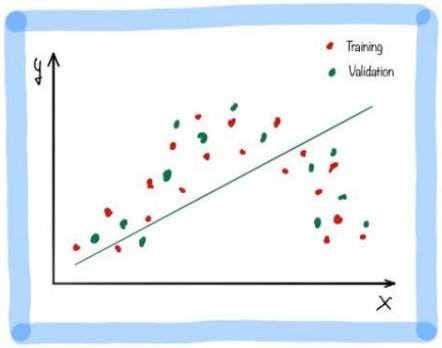


Underfitting: when the degree is too low, the model cannot fit the trend.

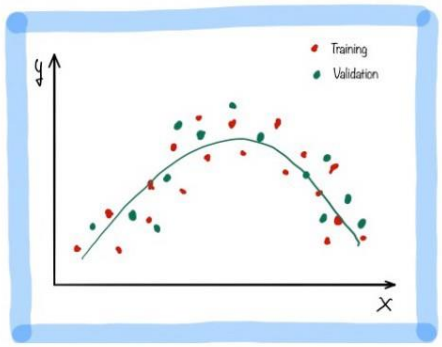
We want a model that fits the trend and ignores the noise.

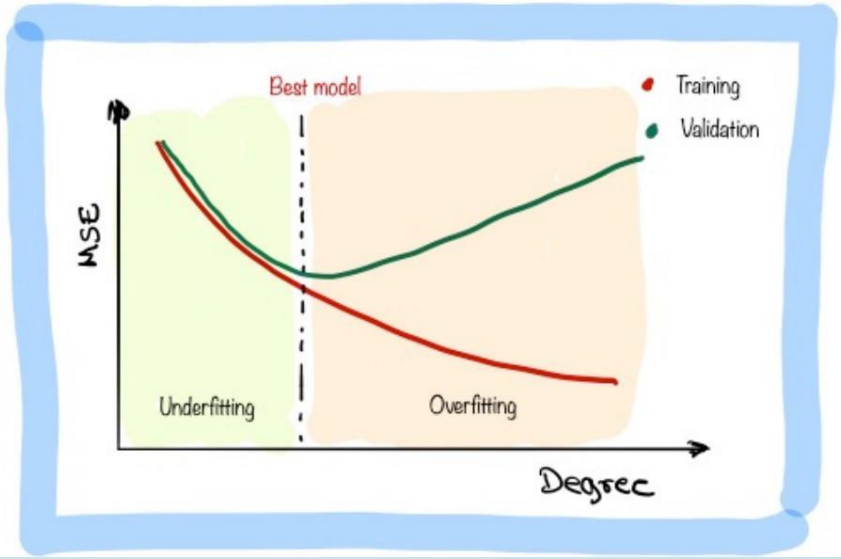
Overfitting: when the degree is too high, the model fits all the noisy data points.

Underfitting: train and validation error is high.

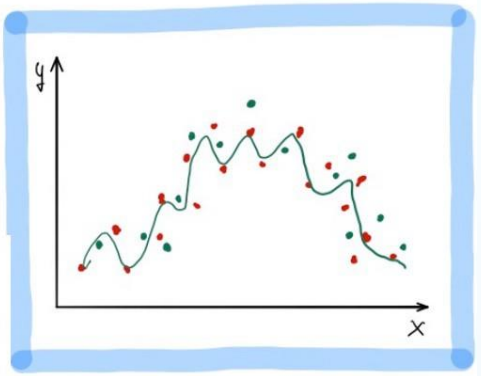


Best model: validation error is minimum.



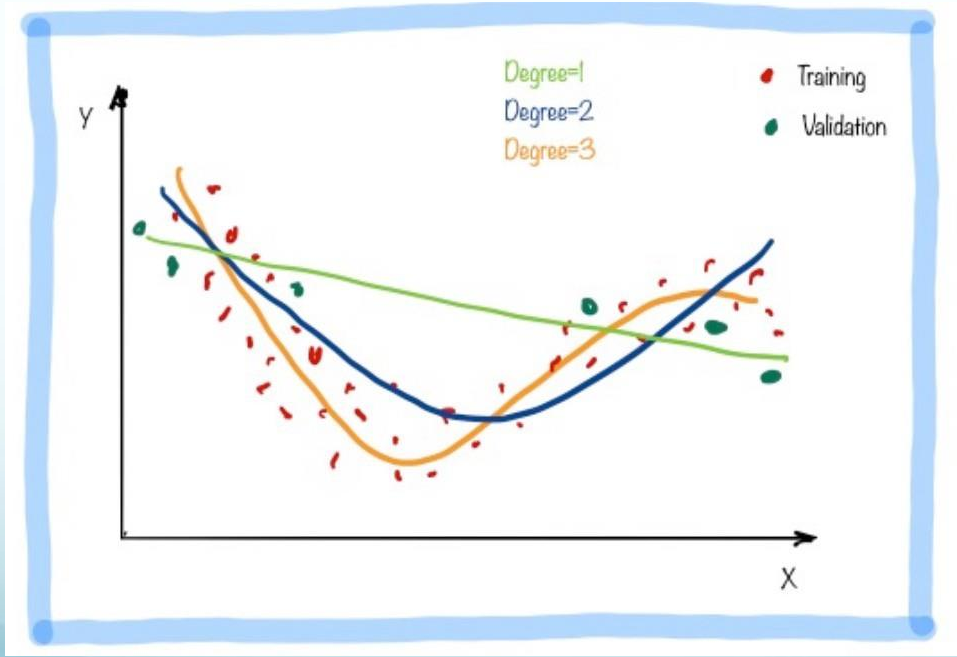


Overfitting: train error is low, validation error is high.



Cross Validation: Motivation

Using a single validation for comparing models- ***there is the possibility of overfitting to the validation set***



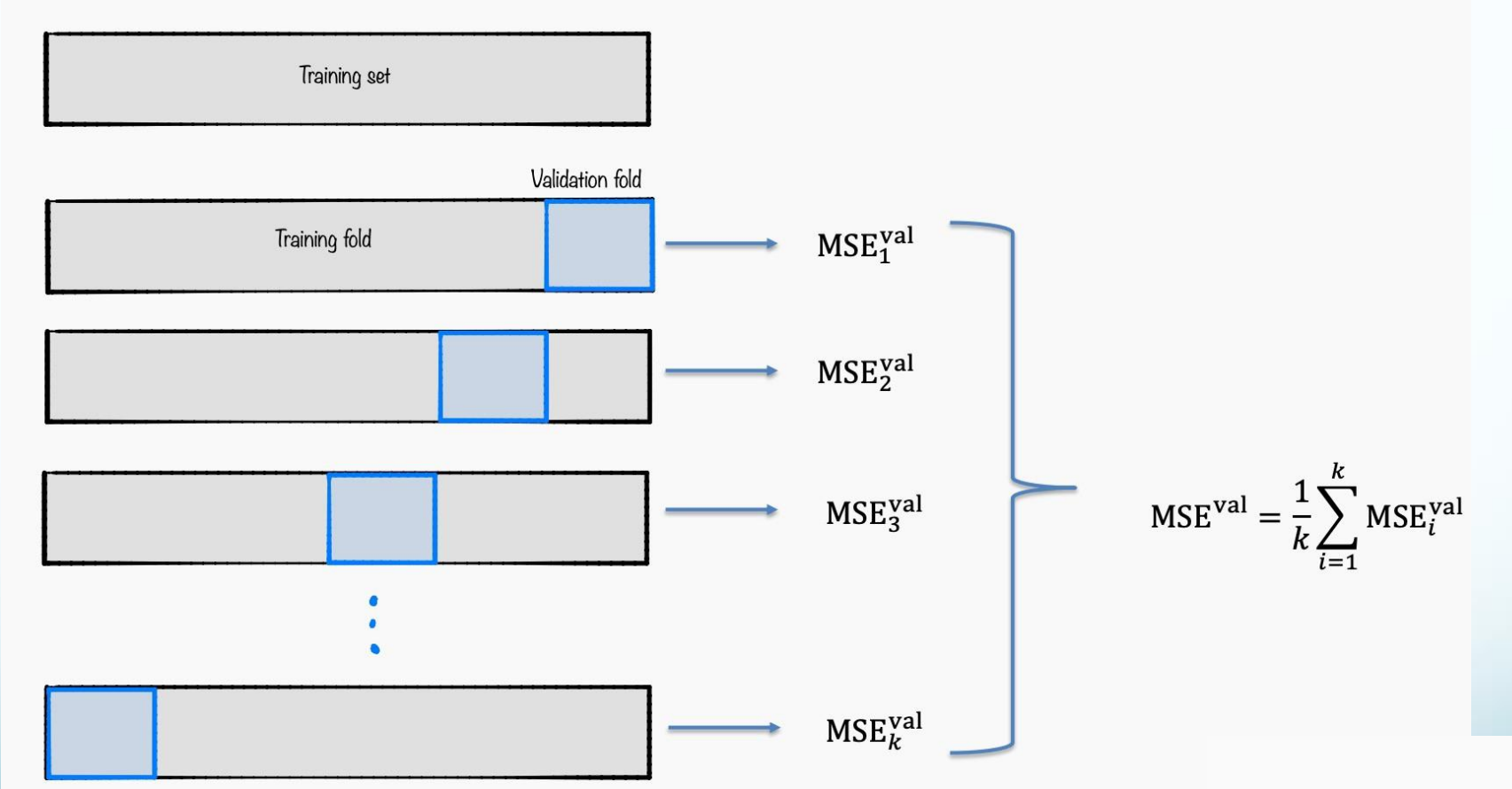
Cross Validation: Motivation

*Let's use **multiple** validation sets and average the validation performance.*

***One approach:** randomly split the training set into training and validation multiple time*

***Problem: R**andomly creating these sets can create the scenario where important features of the data may never appear in random draws.*

Cross Validation



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.

Today's Learning Objectives

Students will be able to:

- ✔ Review: Error metrics
- ✔ Review: Multiple Linear Regression
- ✔ Solve Ordinary Least Squares
- ✔ Apply **model selection** and **cross validation** for **overfitting**

Citations:

Yu, B., & Barter, R. L. (2024). Veridical data science: The practice of responsible data analysis and decision making. The MIT Press.

Shah. C. (2020) A hands-on introduction to data science. Cambridge University Press.

Data 100, Fall 2024, UC Berkeley.

Baharan Mirzasoleiman, UCLA CS M148 Winter 2024 Lecture 4 Notes

Sebatstian Raschka, University of Wisconsin, Stat451, Fall 2020, Lecture 8 Notes