Optimizing Data for Supervised Learning

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

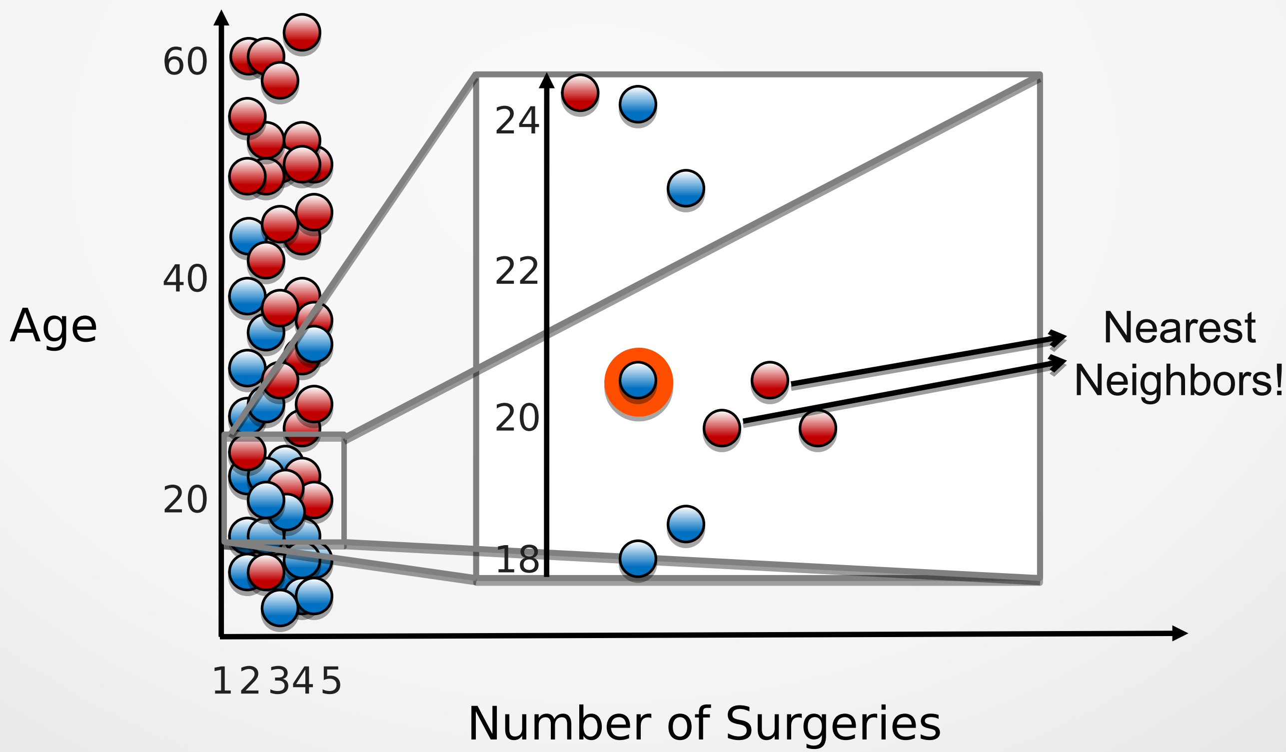
Learning Objectives

After completing this lecture, you will be able to:-

- Appropriately scale your data
- Evaluate and design for a proper generalization/fit
- Split your data for training and testing
- Perform cross validation for model selection
- Transform your features appropriately prior to the training process

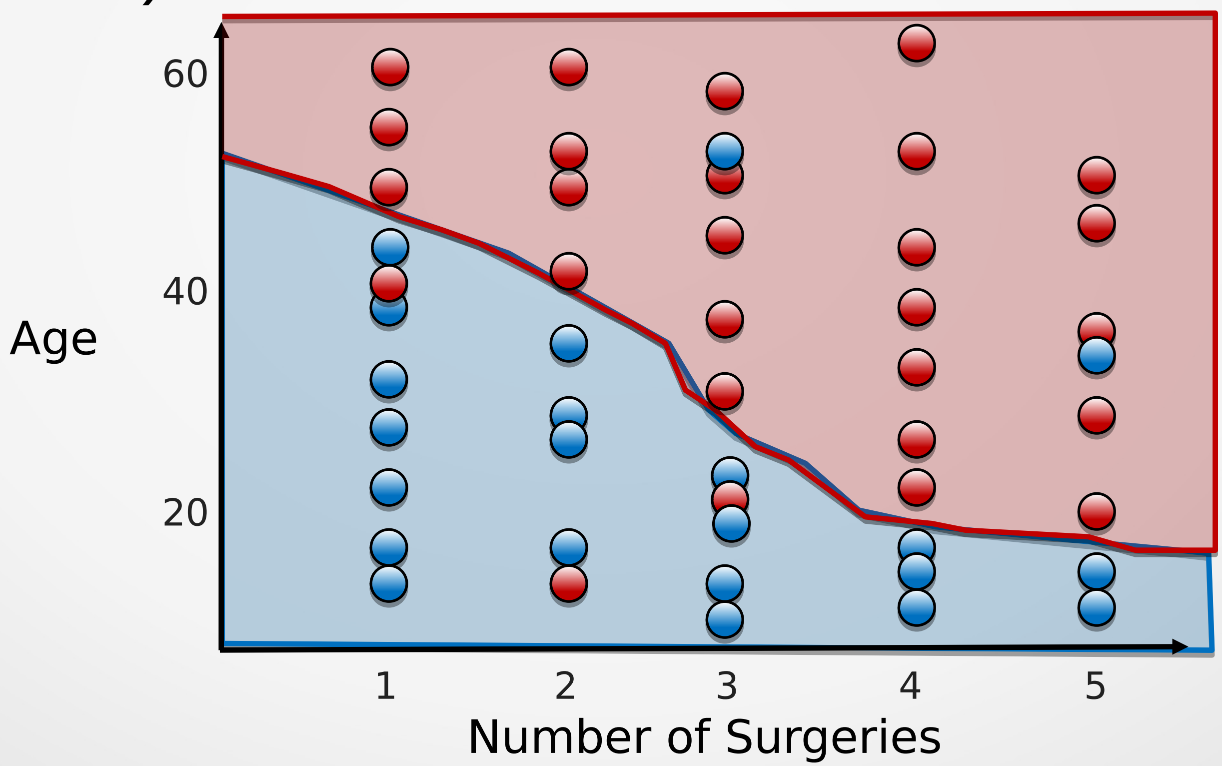
The Effect of Scale

- How would the 1-to-1 scale shown below affect distance measurement for KNN?



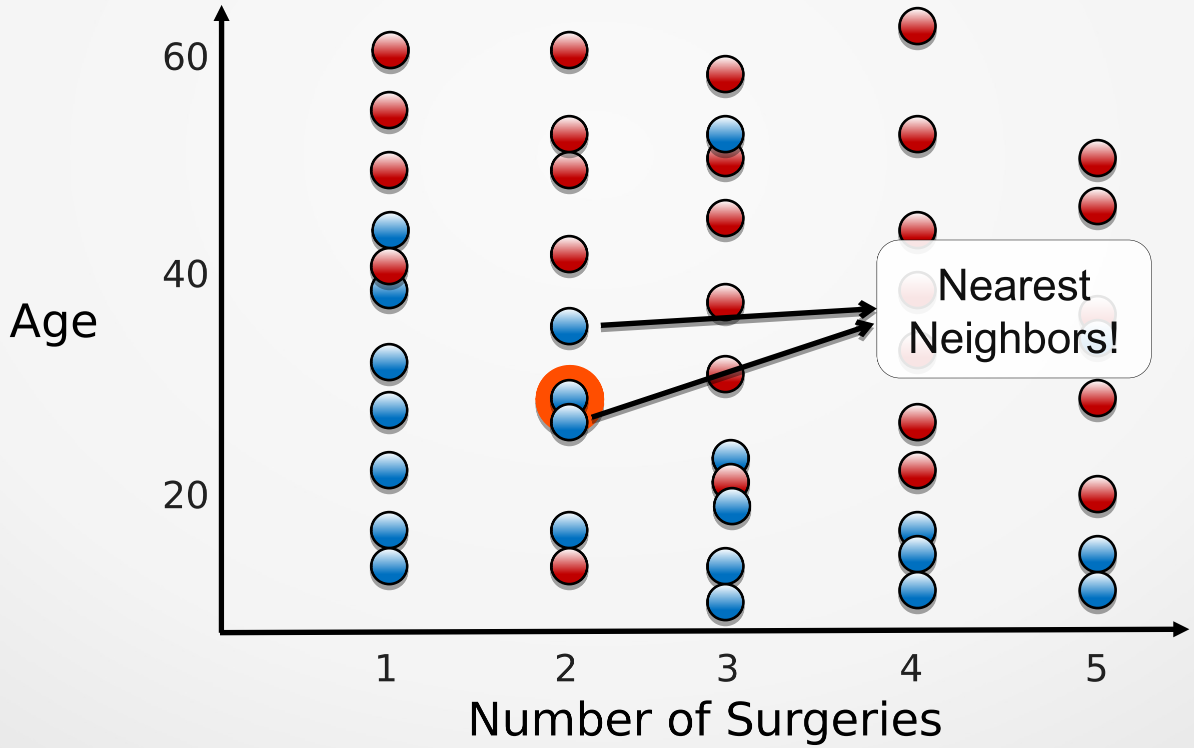
The Effect of Scale

- By scaling features relative to each other, we can get a more accurate representation (and better learning performance!)



The Effect of Scale

- Now where are the nearests neighbor for the previously examined point?



Feature Scaling Methods

Scikit-learn has four feature scalers:-

- **StandardScaler**: Remove the mean and scale to unit variance
- **MinMaxScaler**: Scales to range of [0, 1] (configurable)
- **MaxAbsScaler**: Scales to maximum absolute value
- **RobustScaler**: Removes the median and divides by interquartile range (reduces influence of outliers, no fixed range)

Feature Scaling Syntax

- Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

- Create an instance of the class

```
StdSc = StandardScaler()
```

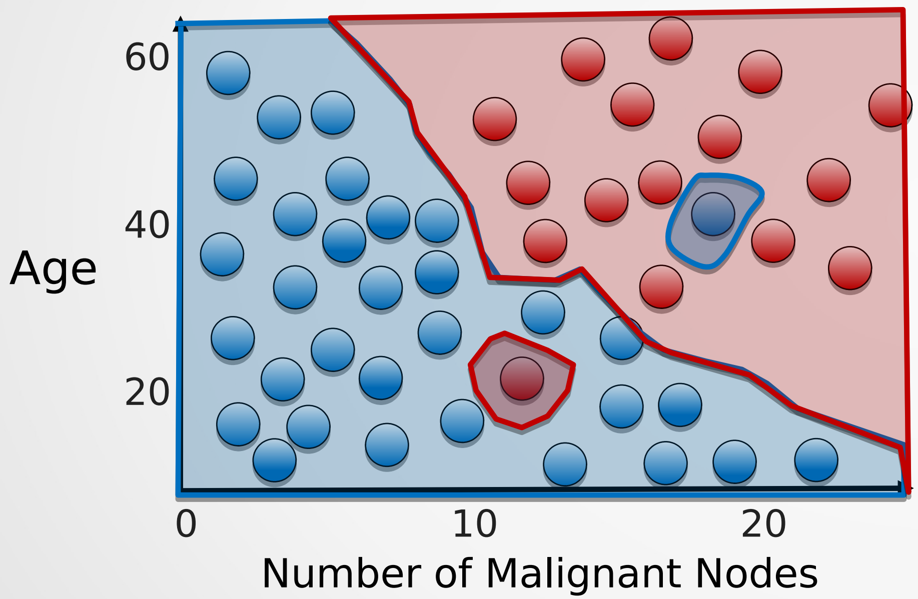
- Fit the scaling parameters and then transform the data

```
StdSc = StdSc.fit(x_data)
x_scaled = StdSc.transform(x_data)
```

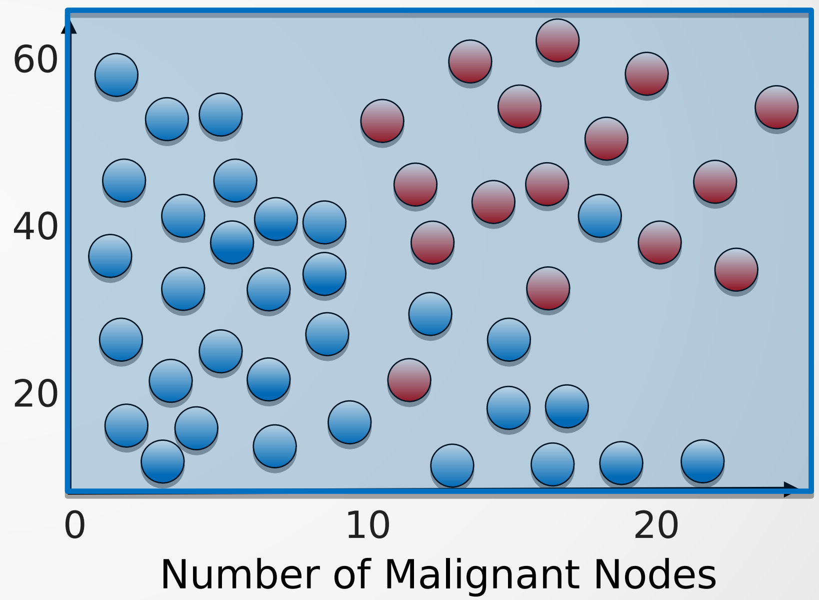
Similar syntax can be used for the other scalers

Generalizing and Proper Fit

We already know that the value of “K” affects the decision boundary for KNN



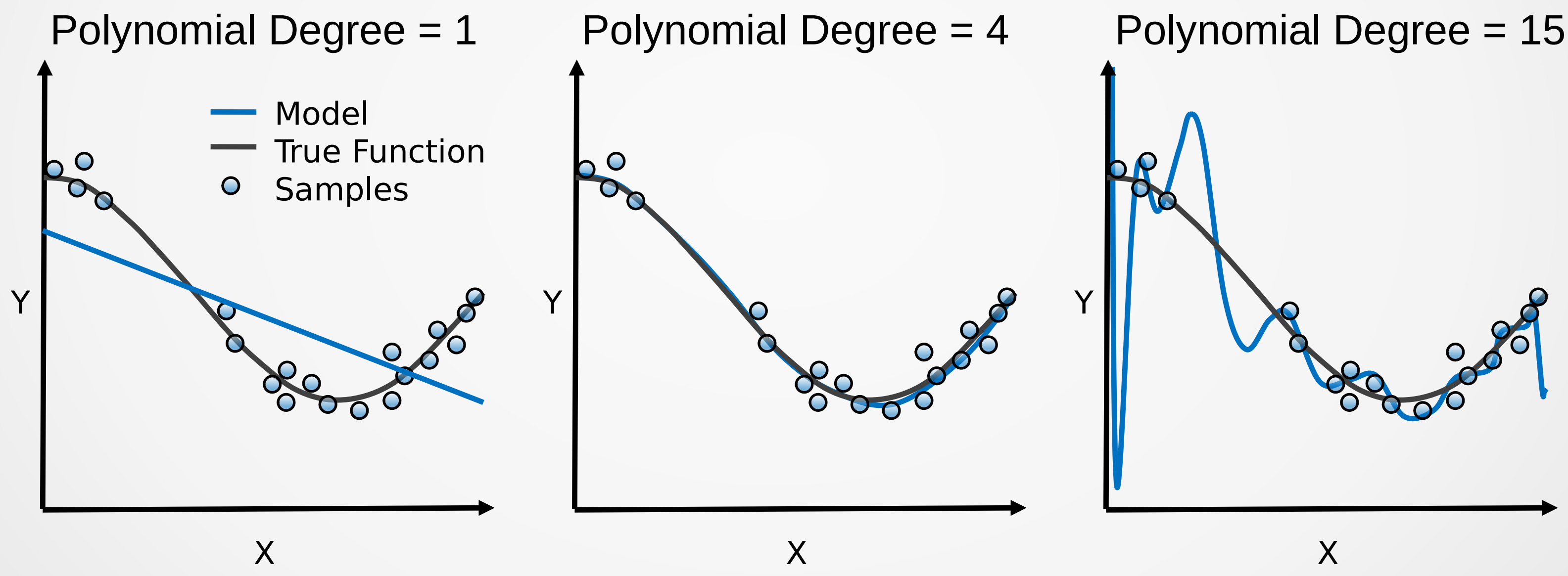
K = 1



K >= 33

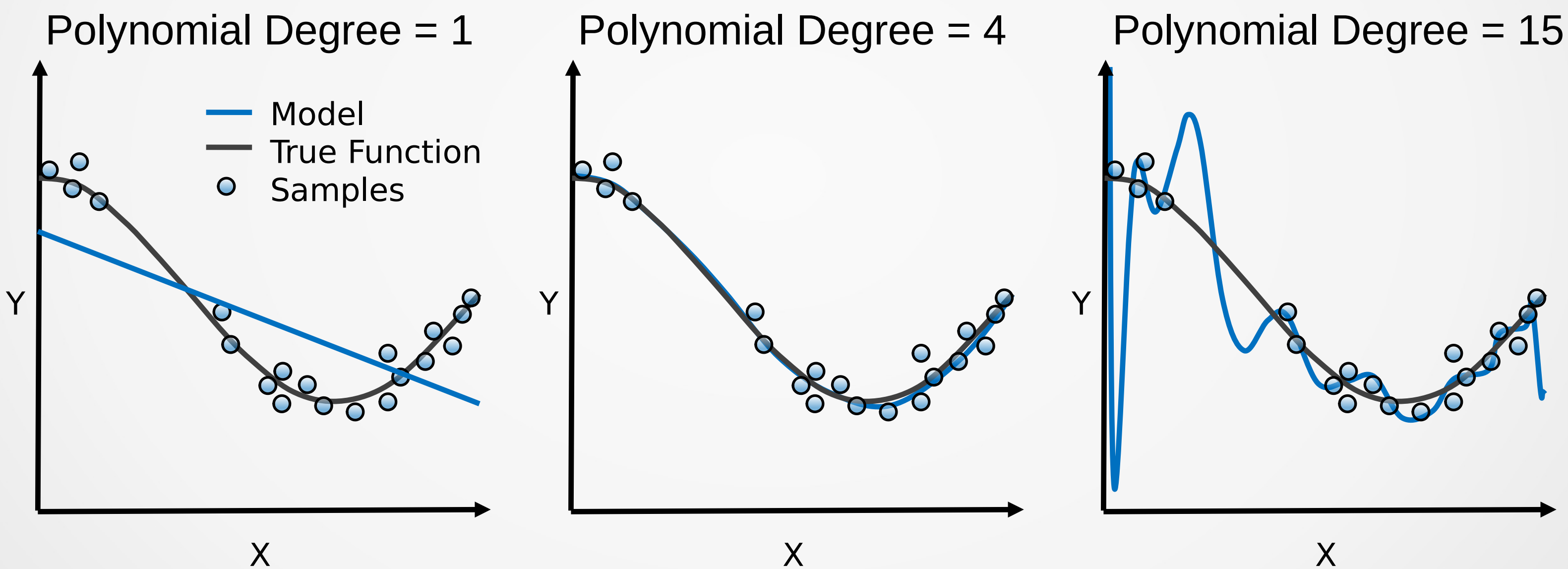
Generalizing and Proper Fit

Similarly, choosing the polynomial degree (complexity) when doing polynomial regression affects the output



Generalizing and Proper Fit

How well does each model generalize?



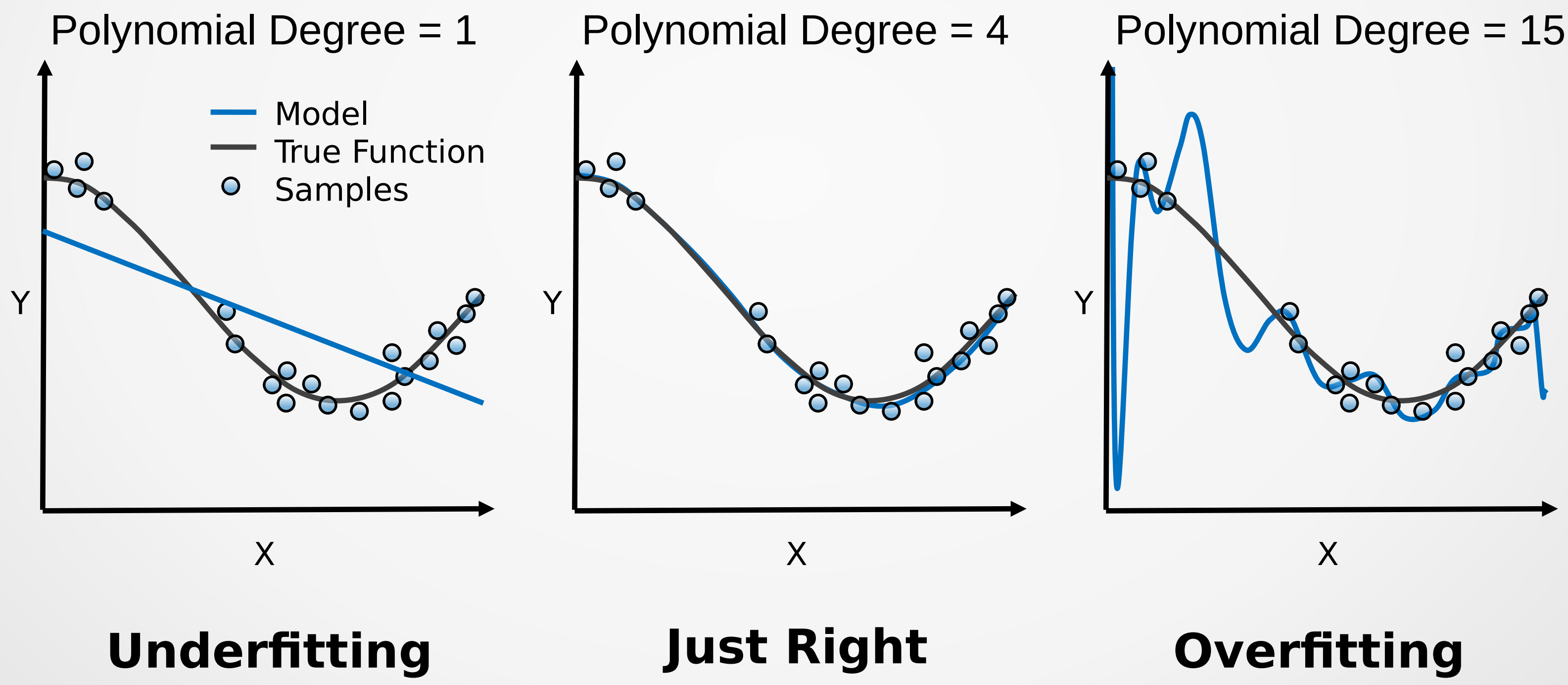
Poor at Training
Poor at Predicting

Just Right

Good at Training
Poor at Predicting

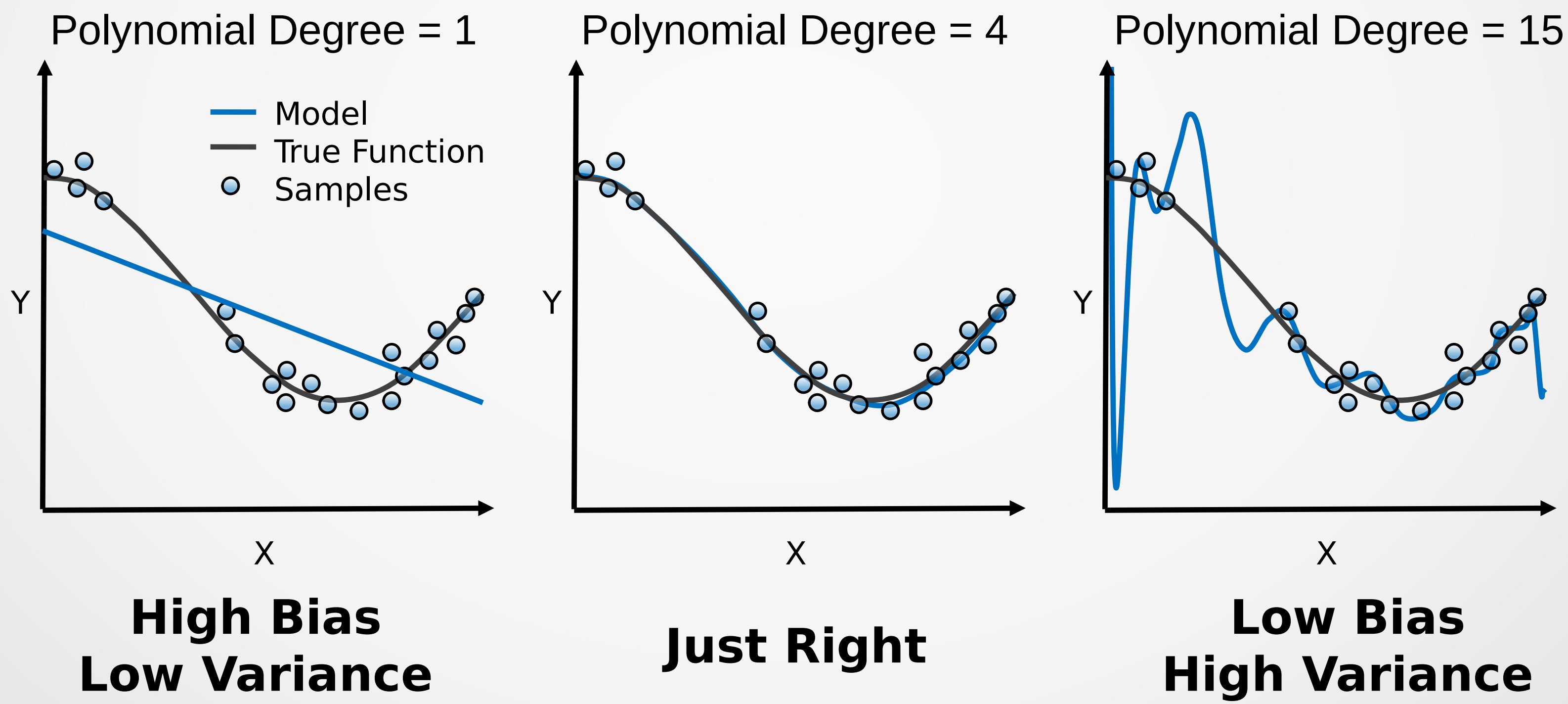
Generalizing and Proper Fit

How well does each model fit (underfit vs overfit)?



Generalizing and Proper Fit

How much trade-off between bias and variance?



Training and Test Splits

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

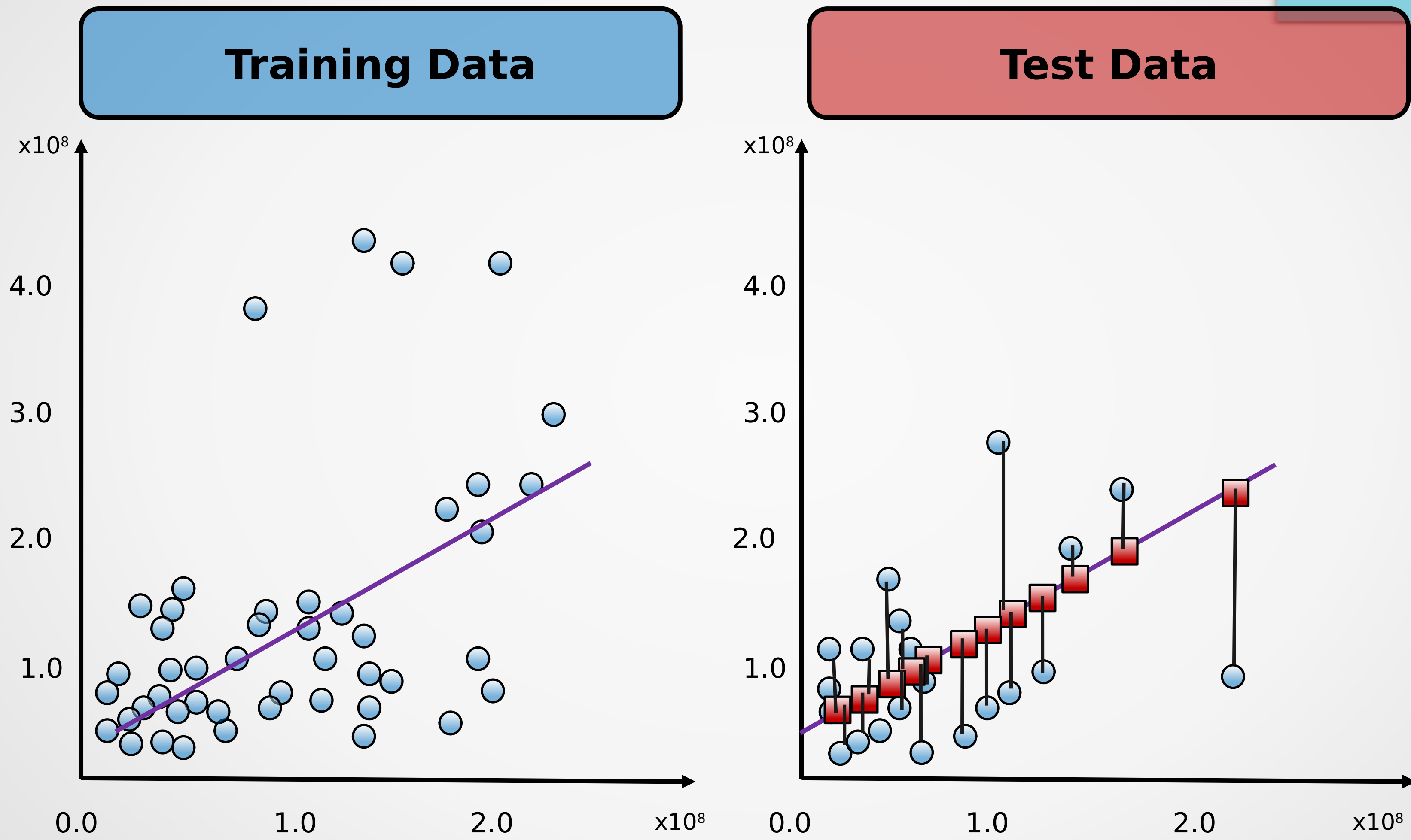
Training Data

Test Data

Training and Test Splits

- Training Data
  - Used to fit/optimize the model
- Test Data
  - Used to measure performance
    - Predict label with fitted/trained model
    - Compare with actual value
    - Measure error

Training and Test Splits

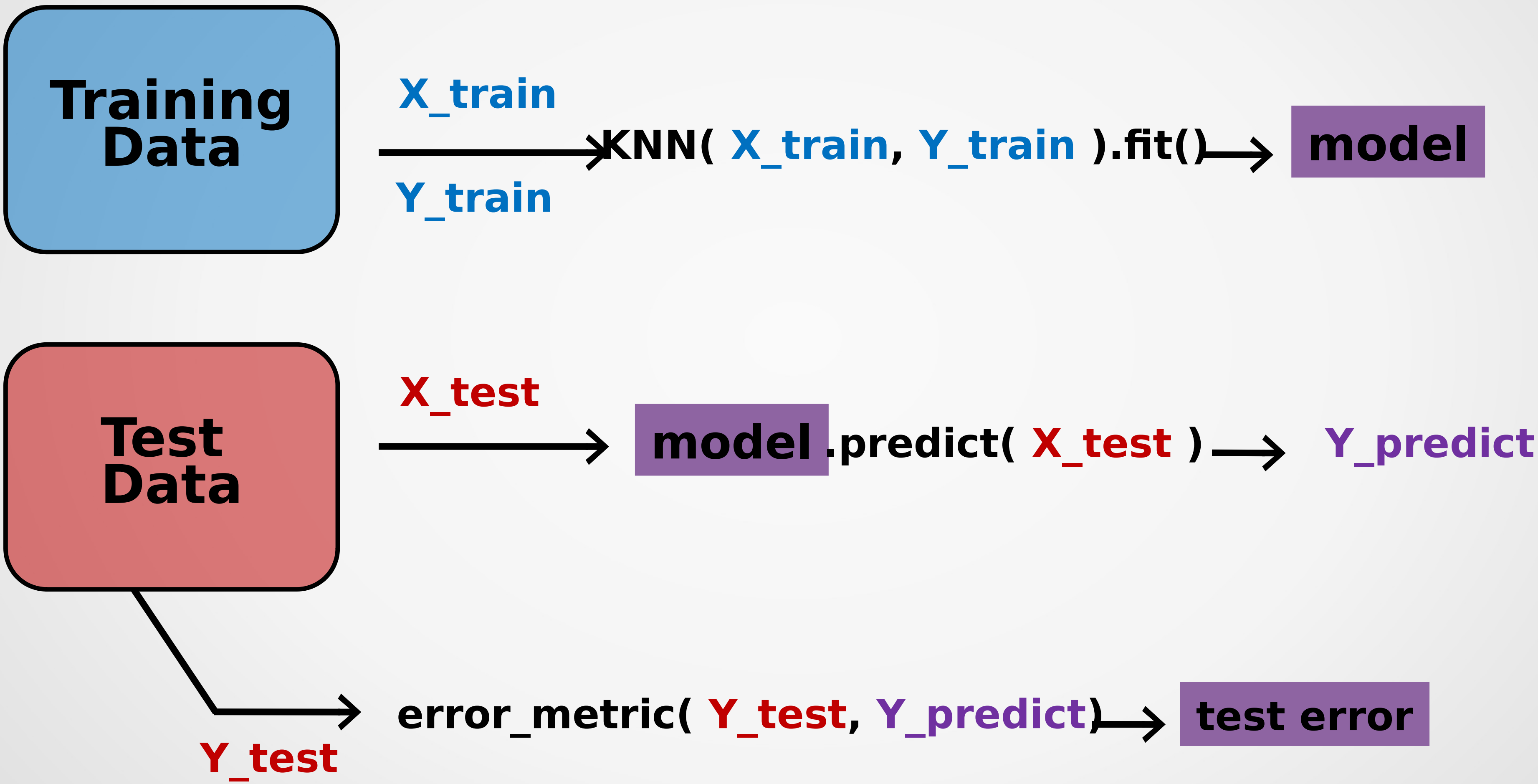


Fit the model

Make predictions

Measure error

Training and Test Splits



Training and Test Splits Syntax

- Import the train and test split function

```
from sklearn.model_selection import train_test_split
```

- Split the data and put 30% into the test set

```
train, test = train_test_split(data, test_size=0.3)
```

Cross Validation

- Using Test sets validates the model's against unseen/new input
- Performance on the Test set should reflect performance 'in-the-wild'
- How well will the Test set (e.g. 30%) generalize to the entire sample?
- Train/Test splits actually gets the best model for **that particular** Test set

Cross Validation

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Training Data 1

Validation Data 1

Cross Validation

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Training Data 2

Validation Data 2

Training Data 2

Cross Validation

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Training Data 3

Validation Data 3

Training Data 3

Cross Validation

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Validation Data 4

Training Data 4

Cross Validation

Training Split	Training Split	Training Split	Test Split

+

Training Split	Training Split	Test Split	Training Split

+

Training Split	Test Split	Training Split	Training Split

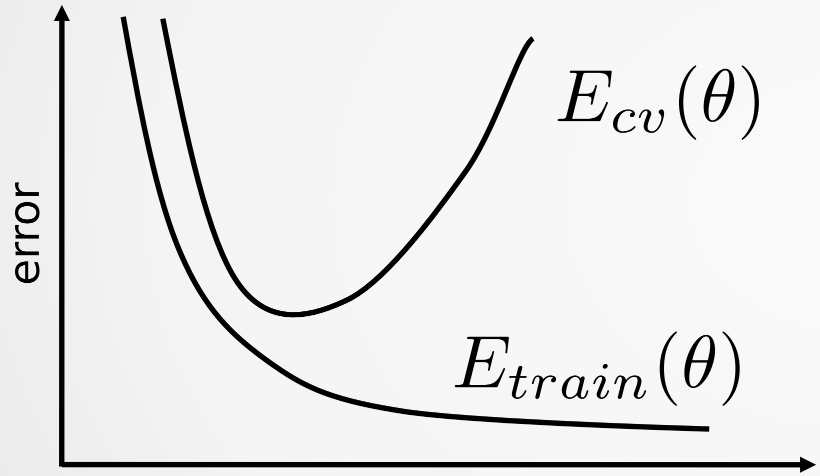
+

Test Split	Training Split	Training Split	Training Split

Average cross validation results.

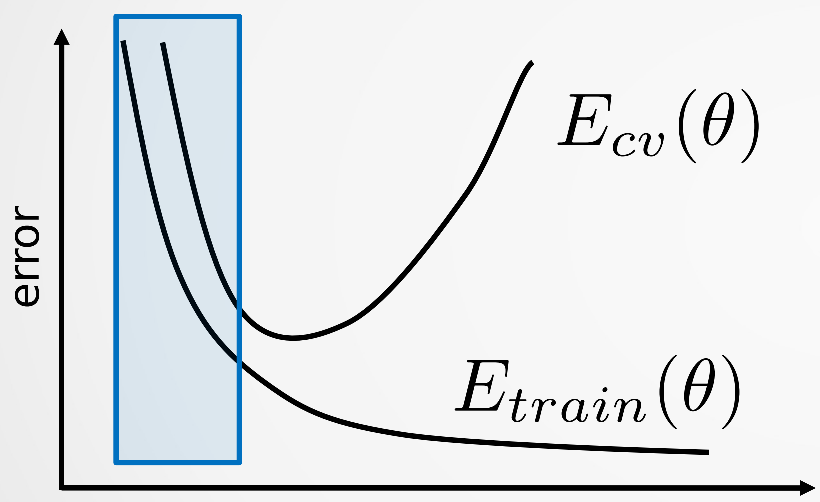
Cross Validation

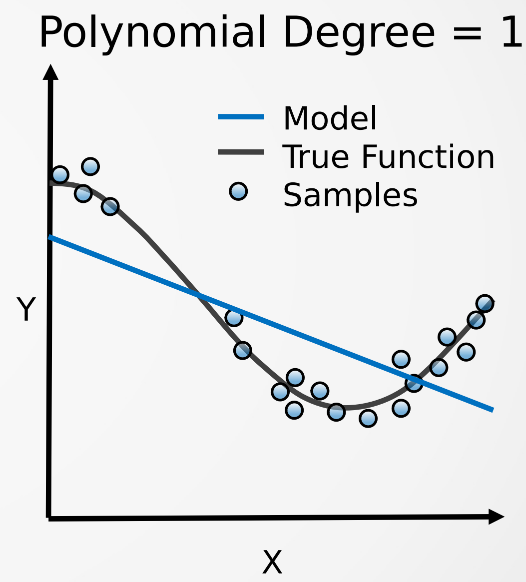
Let's compare training error with cross validation error as model complexity increases



Cross Validation

Let’s compare training error with cross validation error

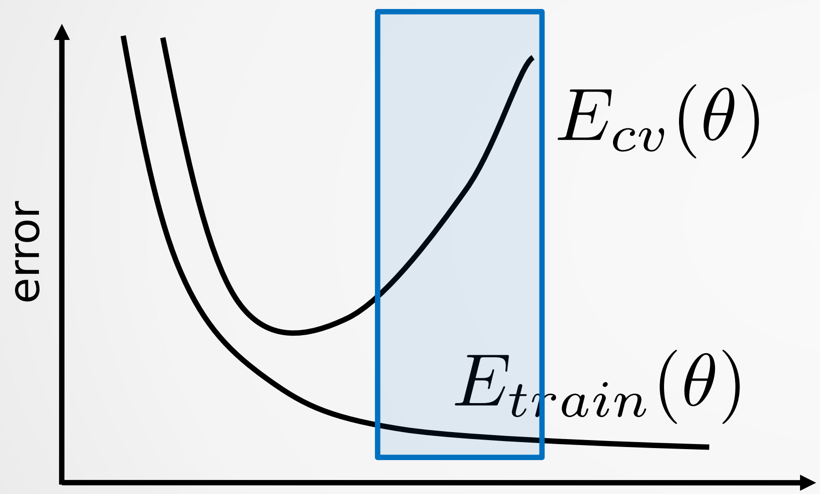


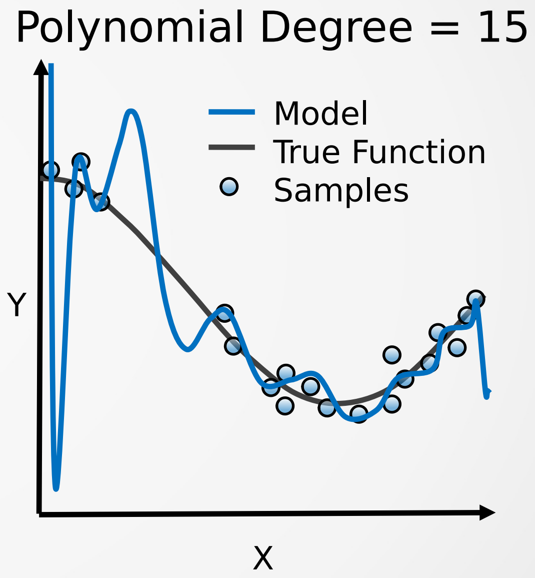


Underfitting: training and cross validation error are high

Cross Validation

Let's compare training error with cross validation error

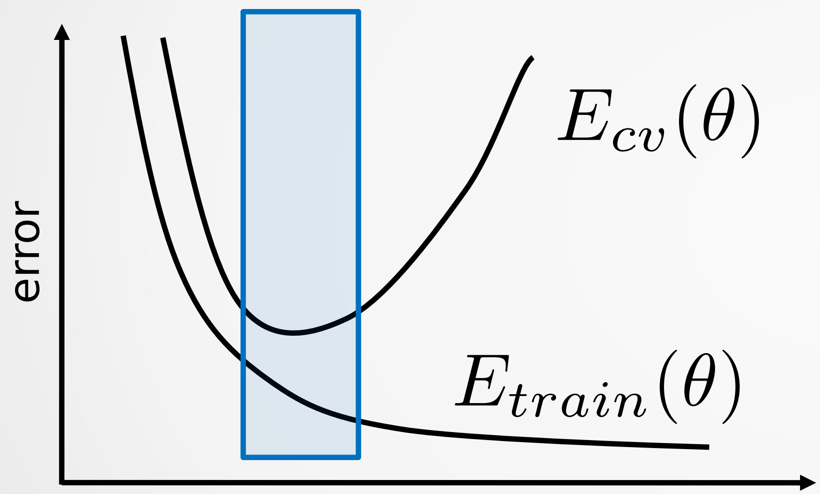


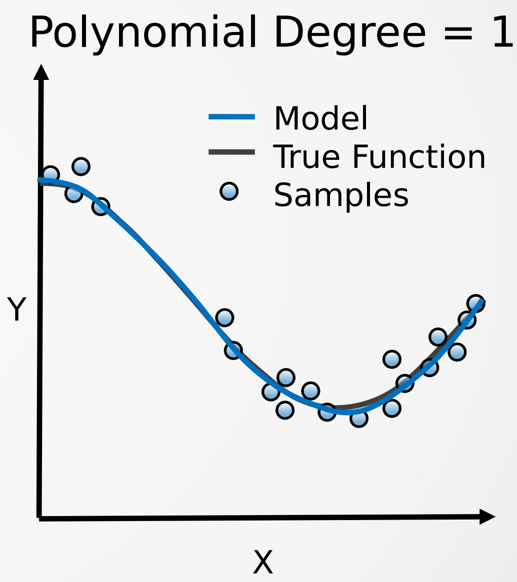


Overfitting: training error is low, cross validation error is high

Cross Validation

Let's compare training error with cross validation error





Just right: training and cross validation error are low

Cross Validation Syntax

- Import the train and test split function

```
from sklearn.model_selection import cross_val_score
```

- Perform cross-validation with a given model

```
cross_val = cross_val_score(KNN, x_data, y_data, cv=4,
        scoring='neg_mean_squared_error')
```

- Other methods for cross validation

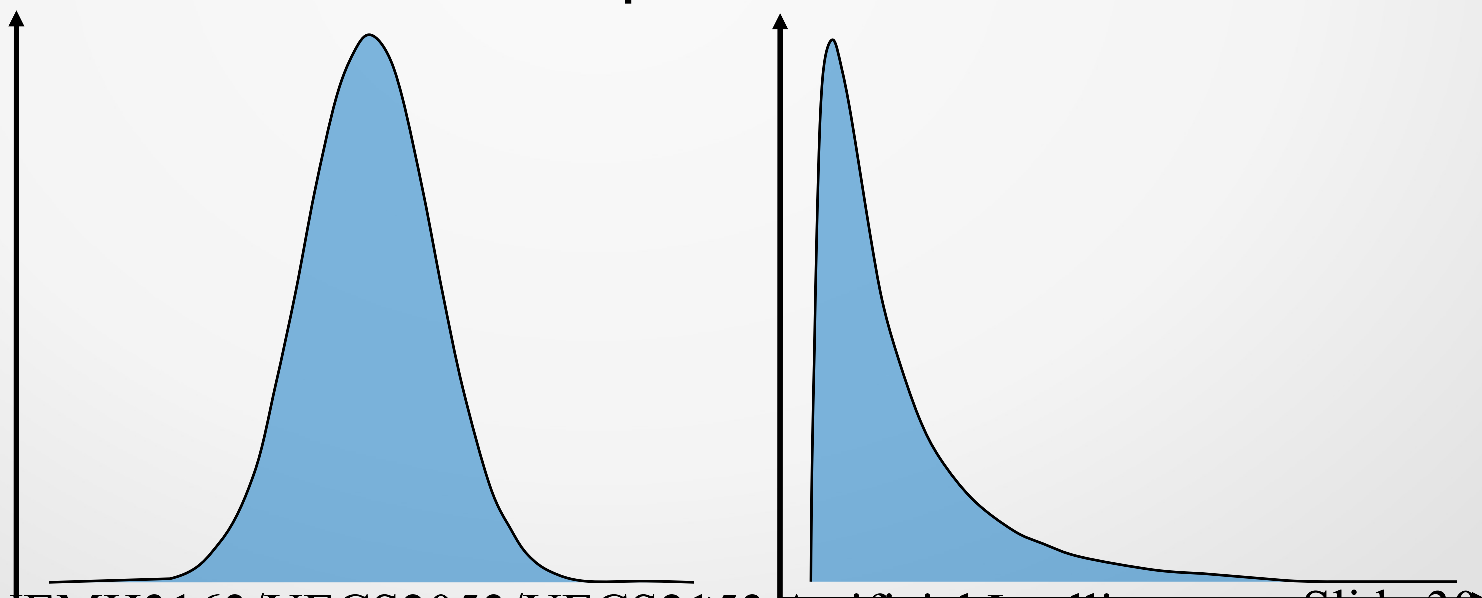
```
from sklearn.model_selection import (BaseCrossValidator,
        GridSearchCV, GroupKFold, Kfold, LeaveOneGroupOut,
        LeaveOneOut, LeavePGroupsOut, LeavePOut,
        RandomizedSearchCV, RepeatedKFold, StratifiedKFold,
        RepeatedStratifiedKFold)
```

Feature Transformation

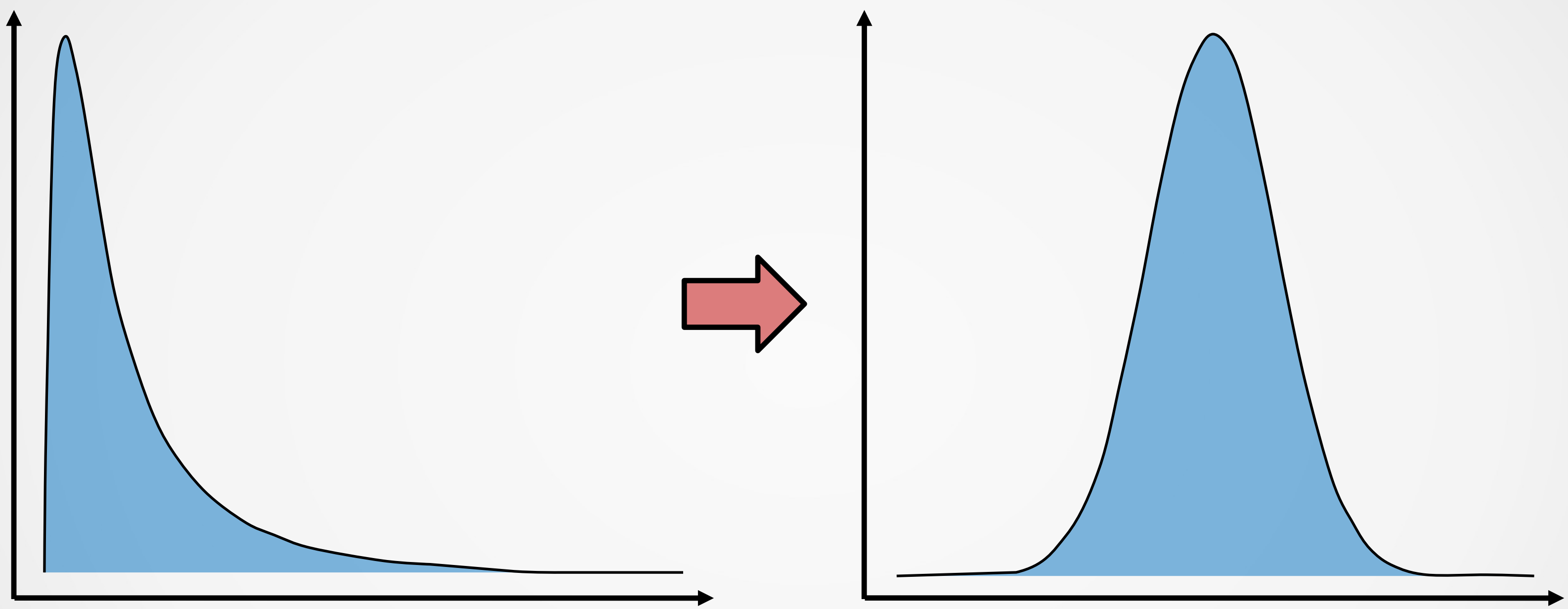
- Any distance-based algorithm (linear regression, KNN etc.) is sensitive to feature scale
- The most obvious effect is from variables with very different ranges
- Transforming the range (through scaling) means the algorithm will find a better solution
- What other feature transformations are useful?

Data Distribution Transformation

- Linear regression models assume residuals are normally distributed
- Features and predicted data often exhibit some level of skew
- Data transformations can help minimize this issue



Data Distribution Transformation



```python
from numpy import log, log1p
from scipy.stats import boxcox
```

Transforming Various Feature Types

Feature Type	Transformation
Continuous: numerical values	Standard Scaling, Min-Max Scaling
Nominal: categorical, unordered features (hair color, country)	One-hot encoding (0, 1)
Ordinal: categorical, ordered features (movie ratings, t-shirt size)	Ordinal encoding (0, 1, 2, 3)

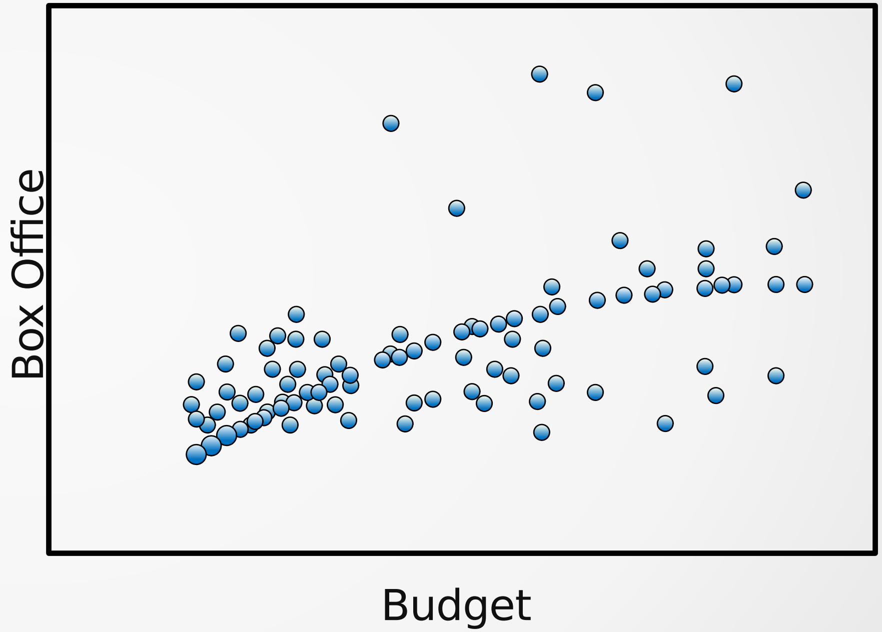
```
from sklearn.preprocessing import (
    LabelEncoder, LabelBinarizer,
    OneHotEncoder)
```

```
from sklearn.feature_extraction import (
    DictVectorizer)
from pandas import get_dummies
```

Adding Polynomial Features

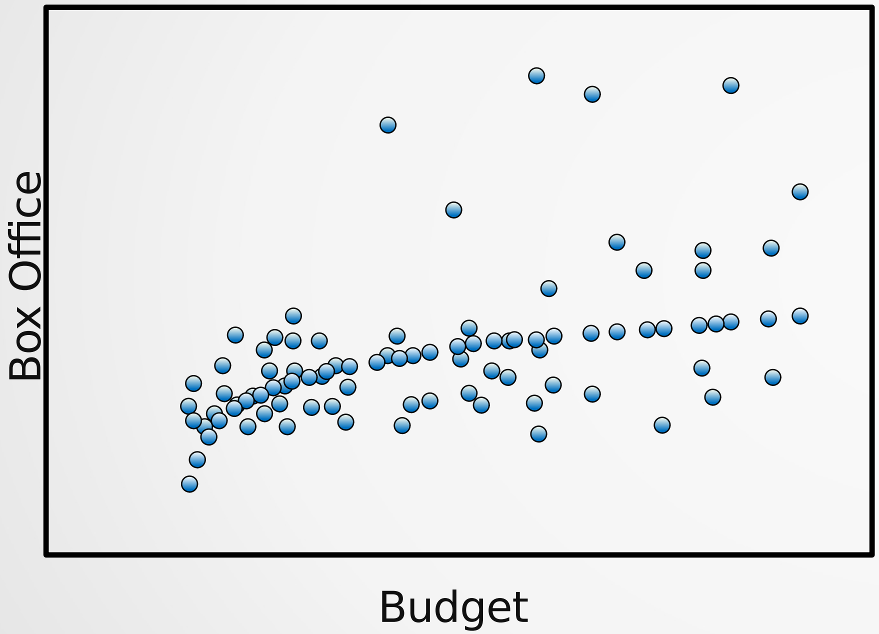
- Higher order features of data can be captured by adding polynomial features
- Still "linear regression" because the equation being solved by the algorithm is a linear combination of features

$$y_\beta(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

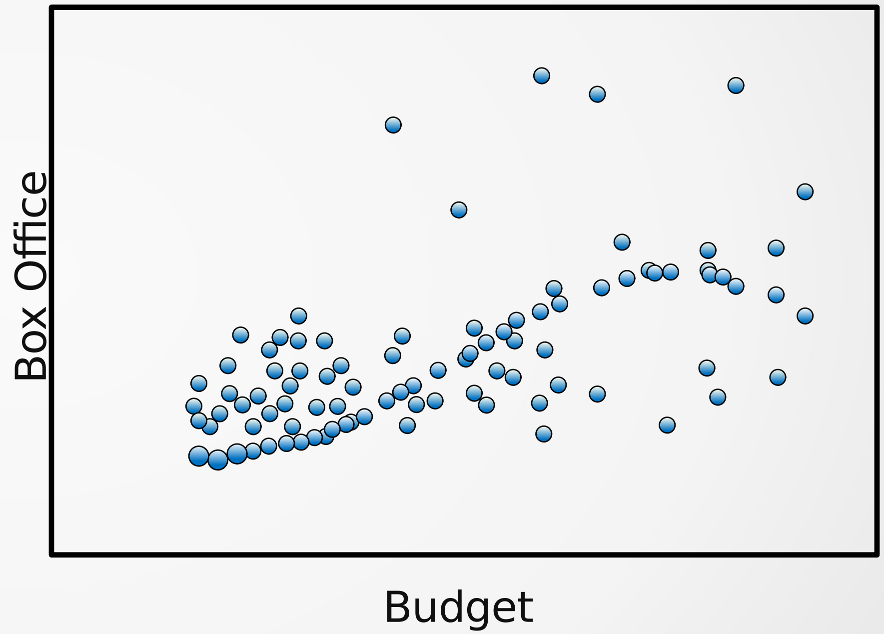


Adding Polynomial Features

$$y_\beta(x) = \beta_0 + \beta_1 \log(x)$$



$$y_\beta(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$



Adding Polynomial Features

- Can also include variable interactions

$$y_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

- How to choose the “correct” functional form?
  - Check relationship between variables or between variables and outcome

Polynomial Features Syntax

- Import the class containing the transformation method

```
from sklearn.preprocessing import PolynomialFeatures
```

- Create an instance of the class

```
polyFeat = PolynomialFeatures(degree=2)
```

- Create the polynomial features and then transform the data

```
polyFeat = polyFeat.fit(x_data)
x_poly = polyFeat.transform(x_data)
```

End of Lecture

Many thanks to Intel Software for providing a variety of resources for this lecture series