FDC104: Programming for Data Analysis and Scientific Computing

Lecture 10 & 11: Model Evaluation

Lecture overview

Topics

- Model evaluation and refinement techniques
- Overfitting, underfitting and model selection

Activities

- Hand-on lab: Model Evaluation & Refinement

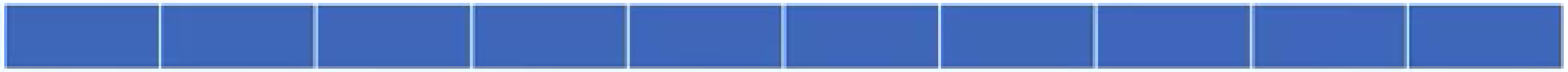
Lecture 10 & 11: Evaluating & Tuning Model

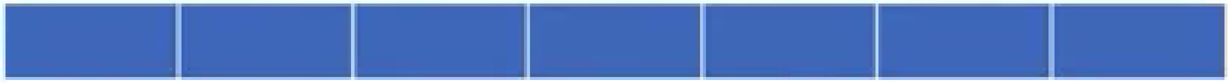
Section 1: Out-of-samples model evaluation

Model Evaluation

- In-sample evaluation tells us how well our model will fit the data used to train it
- Problem?
  - It does not tell us how well the trained model can be used to predict new data
- Solution?
  - In-sample data or train data: train model
  - Out-of-sample evaluation or test set: approximate how the model performs in real world

Training/Testing Sets

Data: 

- Split dataset into:
  - Training set (70%): 
  - Testing set (30%): 

- Build and train the model with the training set
- Use testing set to assess the performance of the model

Function train_test_split()

- Split data into random train and test subsets

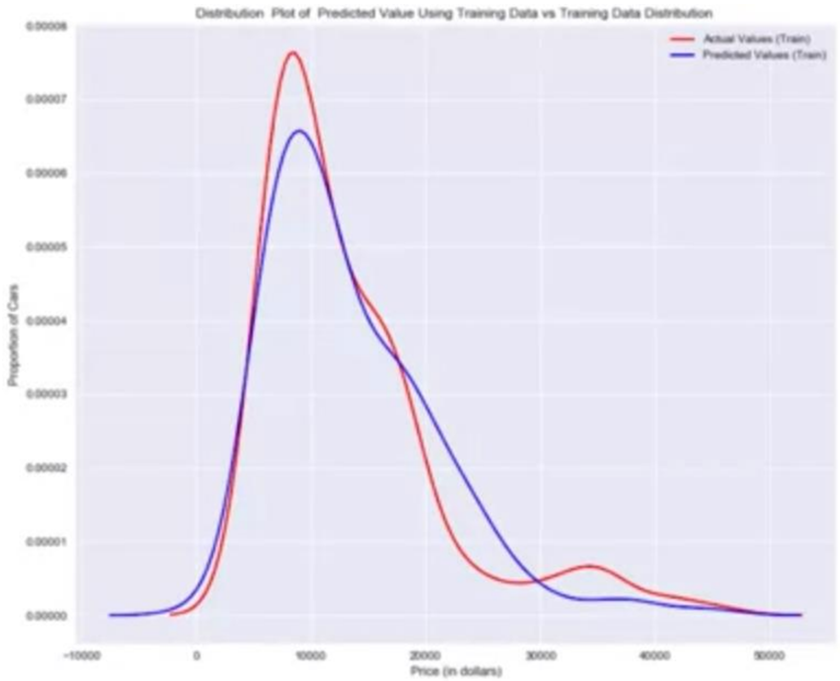
```
from sklearn.model_selection import train_test_split
```

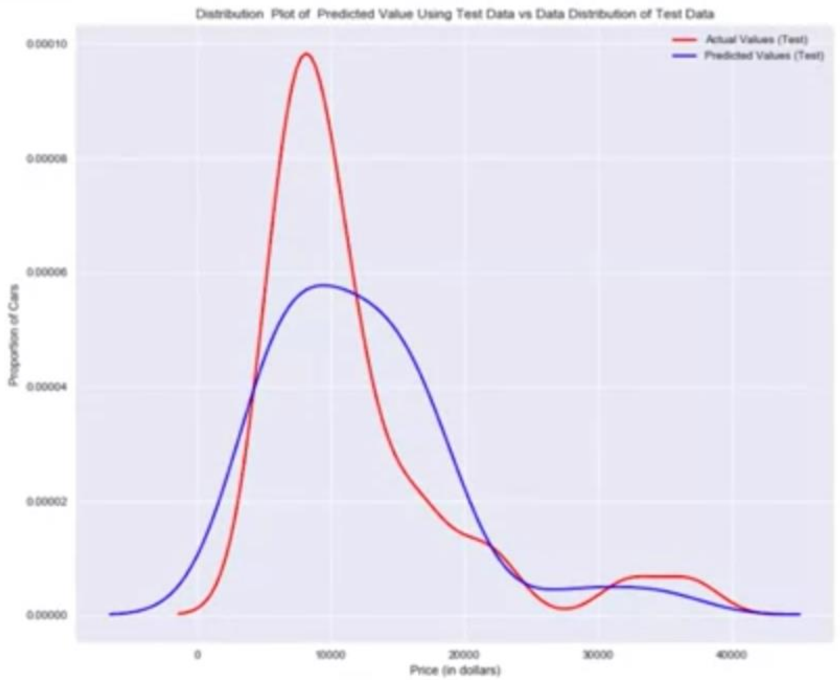
```
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.3, random_state=0)
```

- **x_data**: features or independent variables
- **y_data**: dataset target: df['price']
- **x_train, y_train**: parts of available data as training set
- **x_test, y_test**: parts of available data as testing set
- **test_size**: percentage of the data for testing (here 30%)
- **random_state**: number generator used for random sampling

Generalization Performance

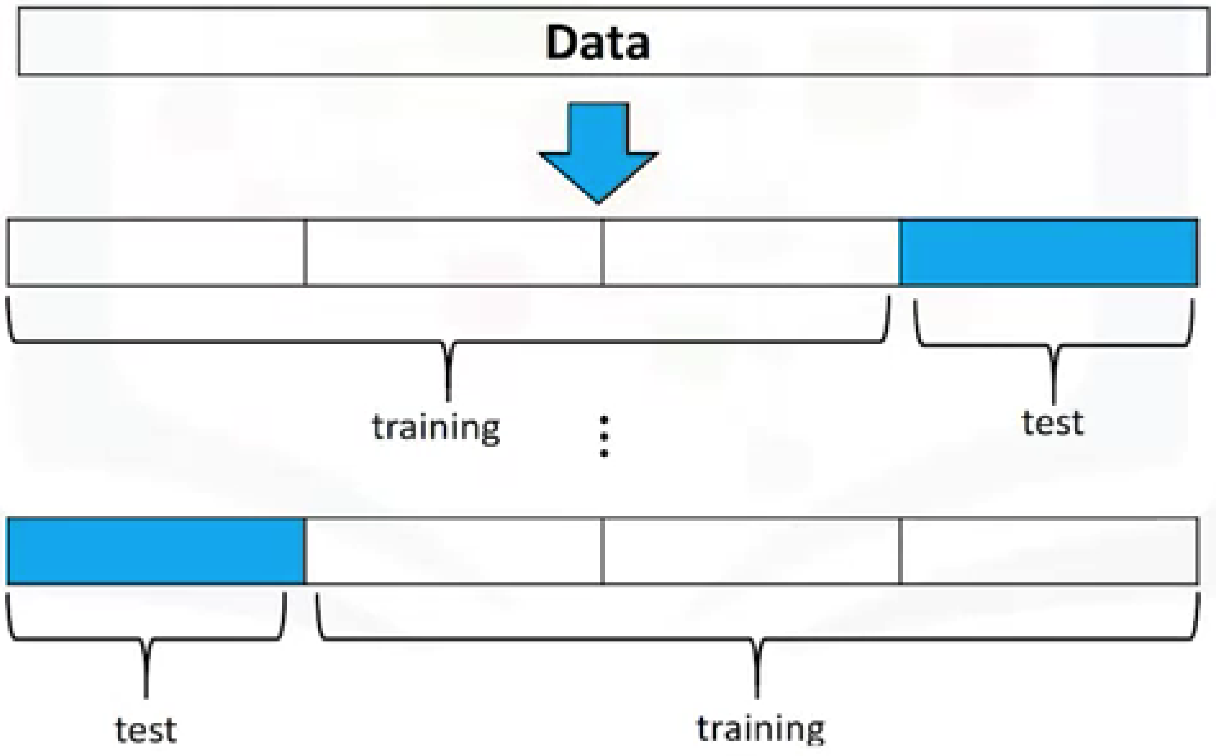
- Generalization error is measure of how well our model does at prediction unseen data
- The error we obtain using our testing set is an approximation of this error





Cross Validation

- Most common out-of-sample evaluation metrics
- More effective use of data (each observation is used for both training and testing)

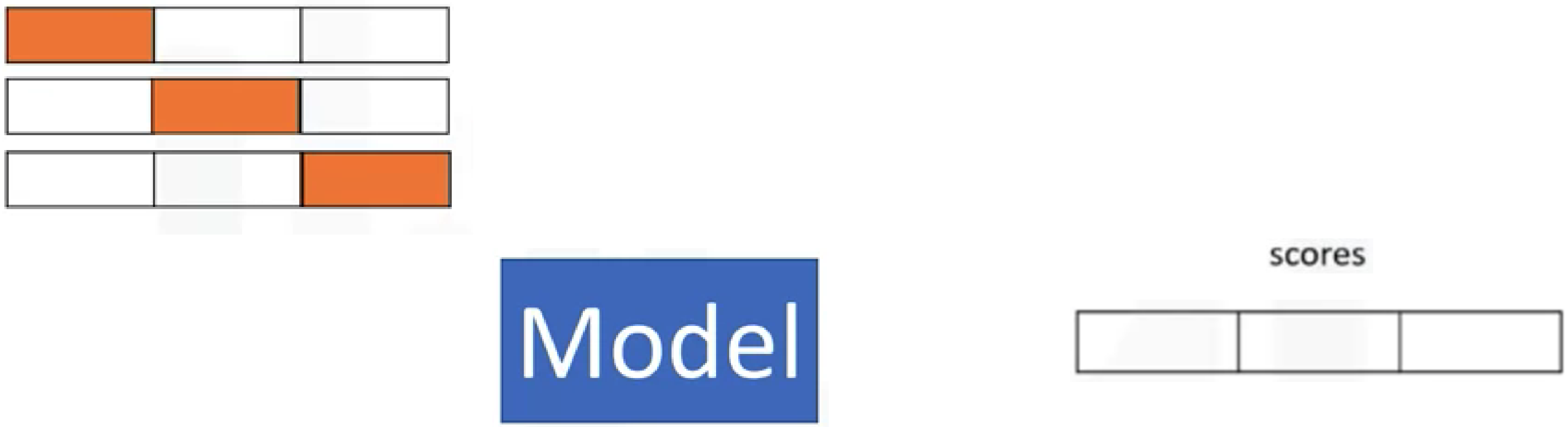


Function cross_val_score()

```
from sklearn.model_selection import cross_val_score

scores= cross_val_score(lr, x_data, y_data, cv=3)

np.mean(scores)
```



Lecture 10 & 11: Evaluating & Tuning Model

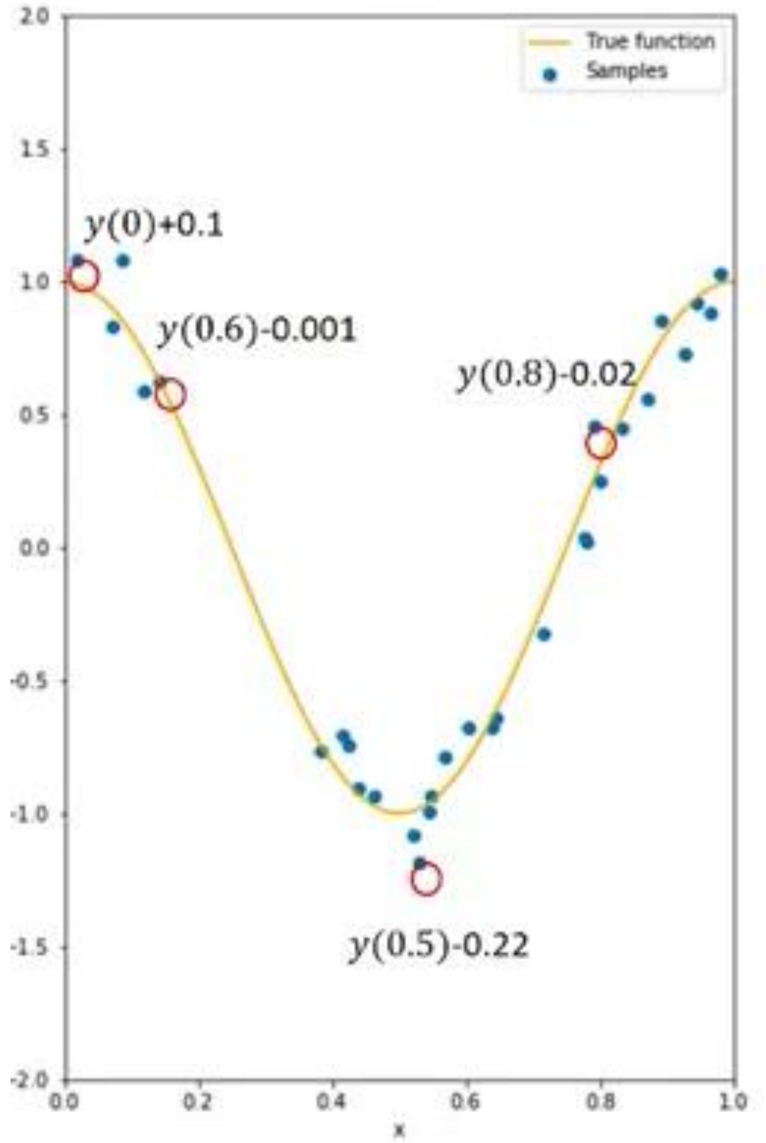
Section 2: Underfitting & Overfitting

DATAPOT
Data Analytics Group

TRƯỜNG ĐẠI HỌC NGOẠI THƯƠNG
Foreign Trade University

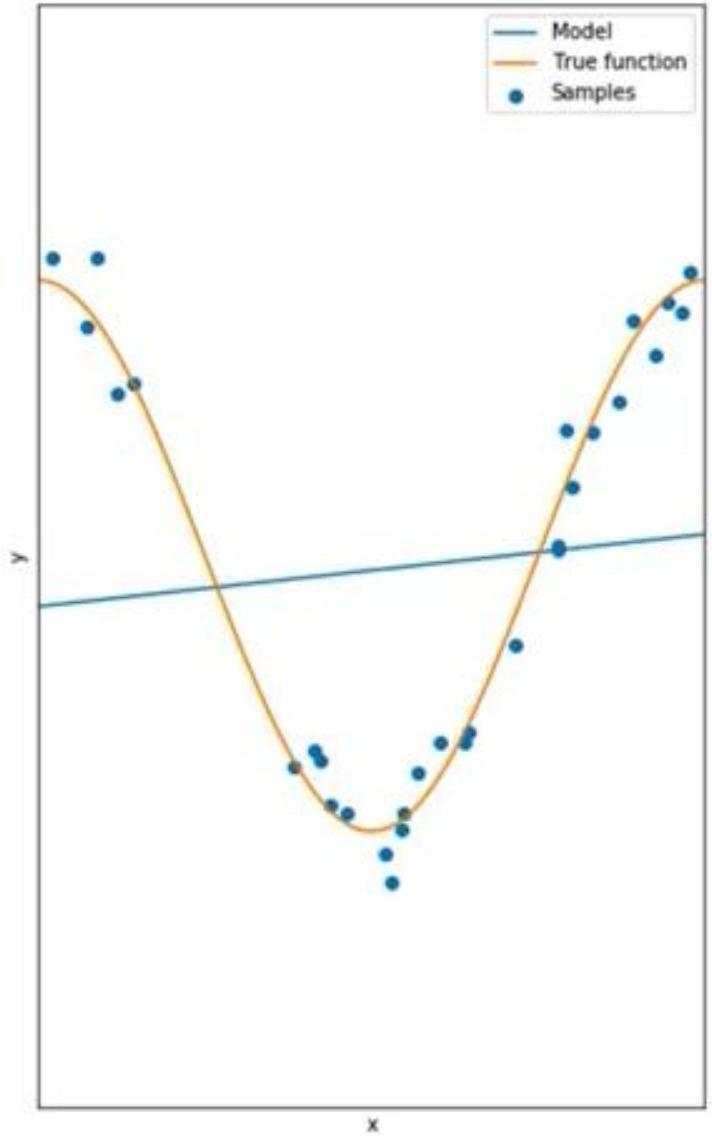
Underfitting & Overfitting

$y(x)$+noise



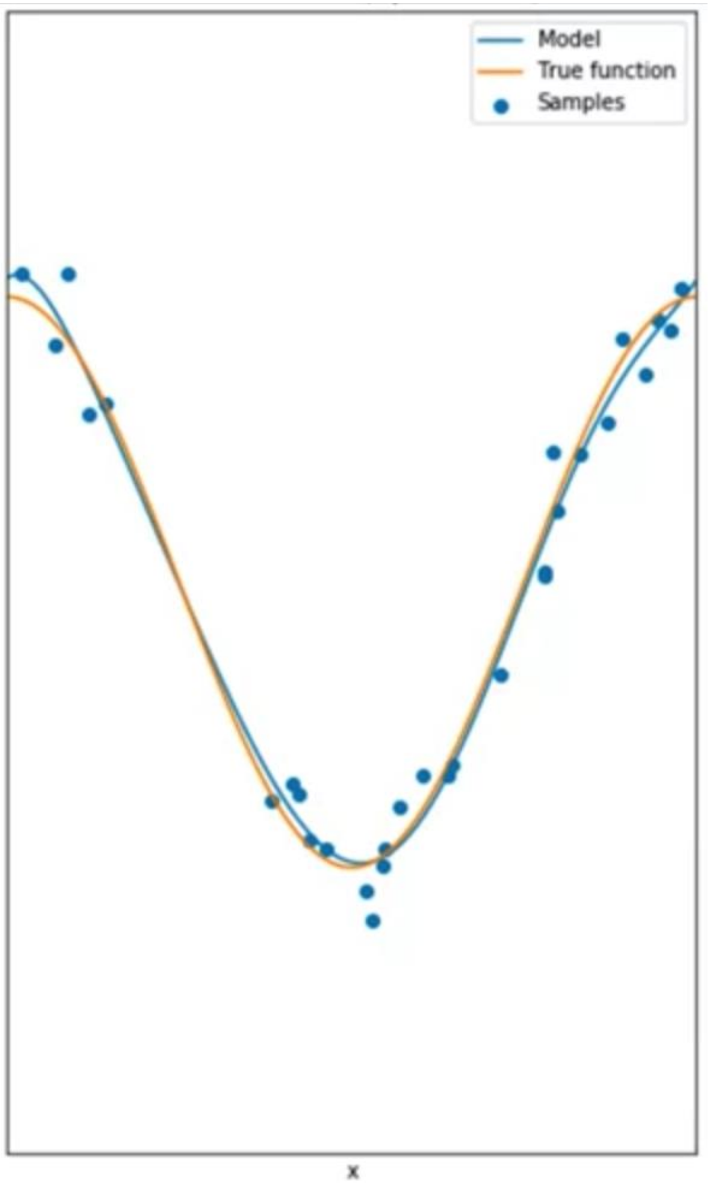
Underfitting & Overfitting

$$y = b_0 + b_1\, x$$



Underfitting & Overfitting

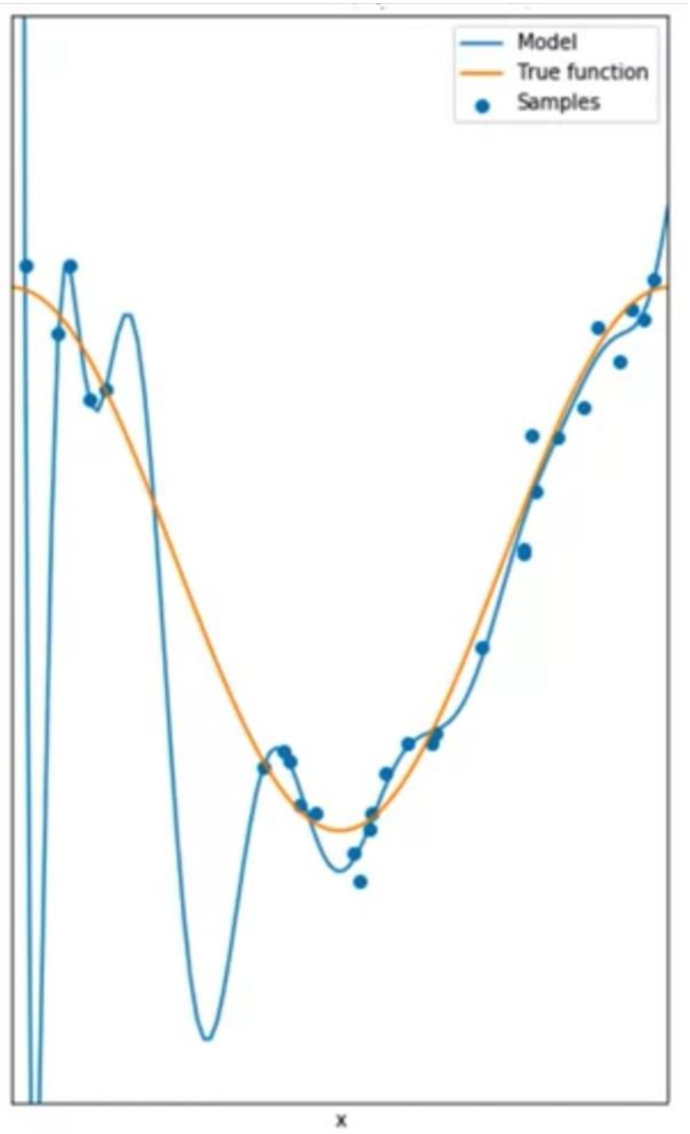
$$\hat{y} = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + b_5 x^5 + b_6 x^6 + b_7 x^7 + b_8 x^8$$



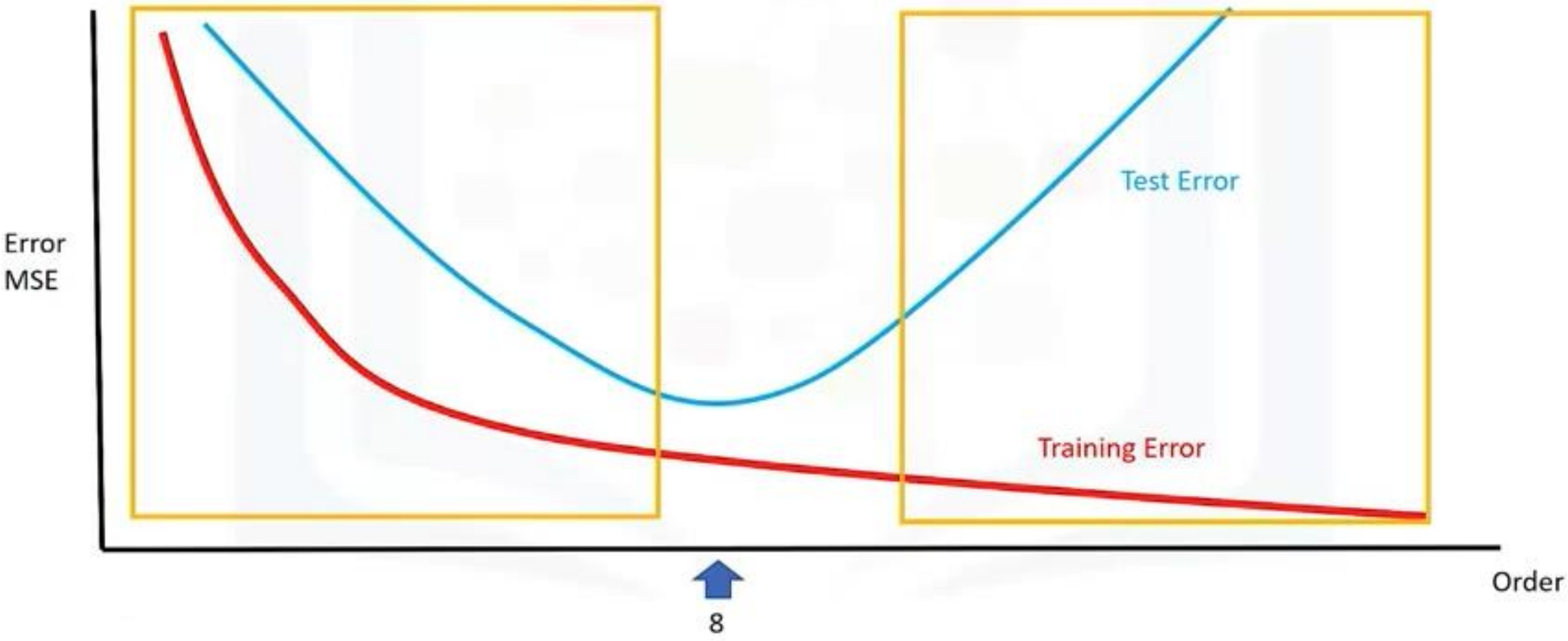
Underfitting & Overfitting

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + b_5 x^5 + b_6 x^6 + b_7 x^7 + b_8 x^8 + ..$$

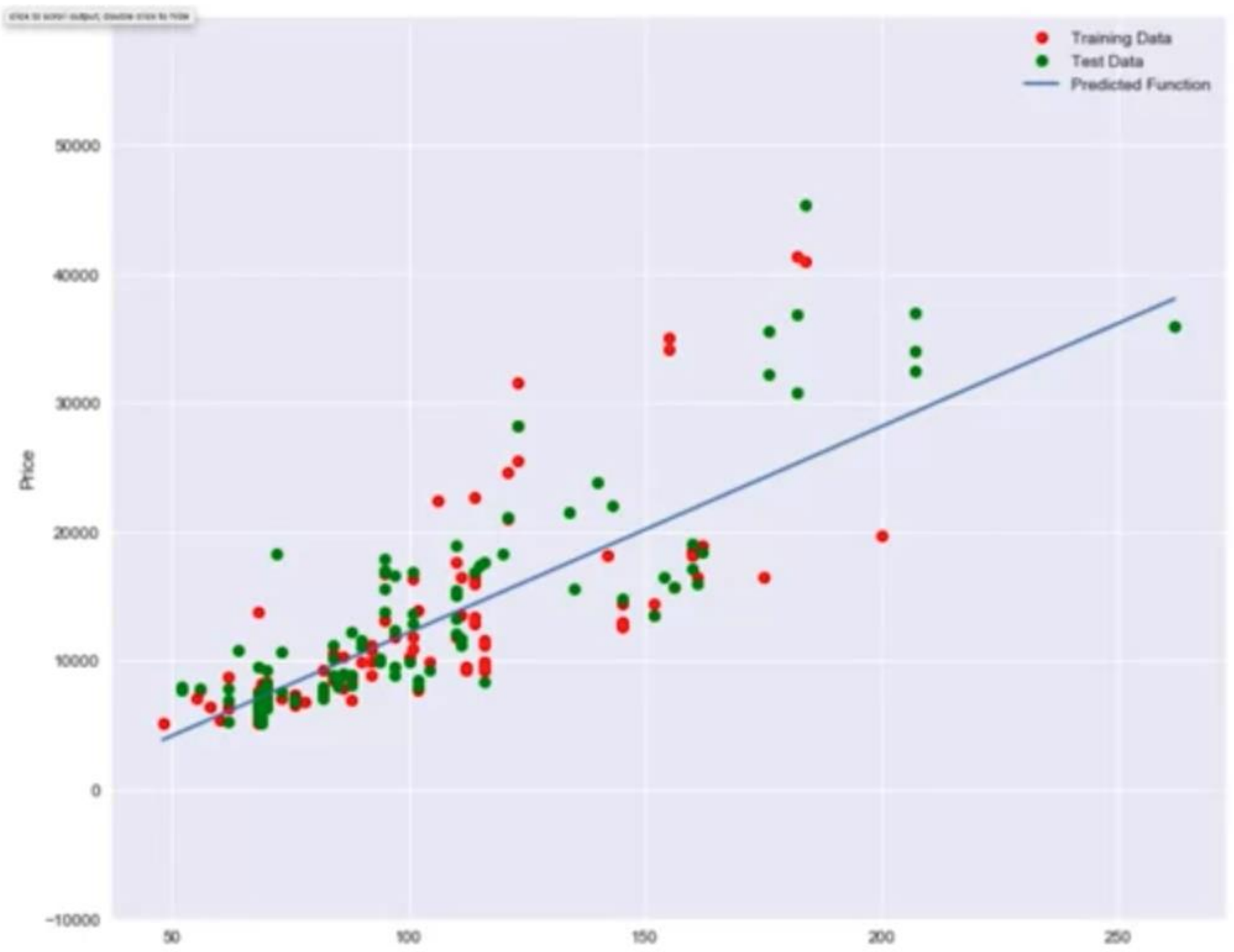
$$+ b_9 x^9 + b_{10} x^{10} + b_{11} x^{11} + b_{12} x^{12} + b_{13} x^{13} + b_{14} x^{14} + b_{15} x^{15} + b_{16} x^{16}$$



Model Selection

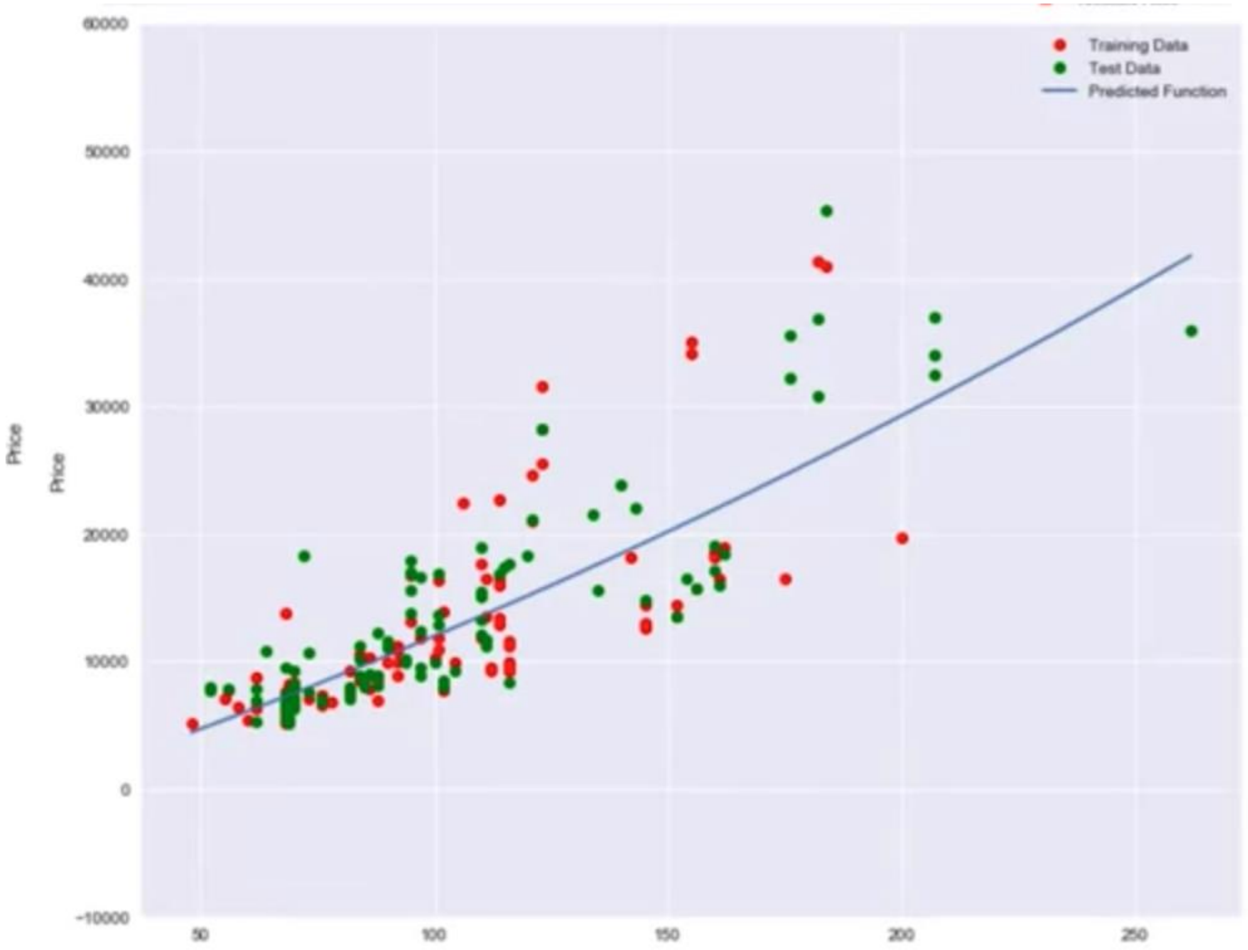


Model Selection: Example



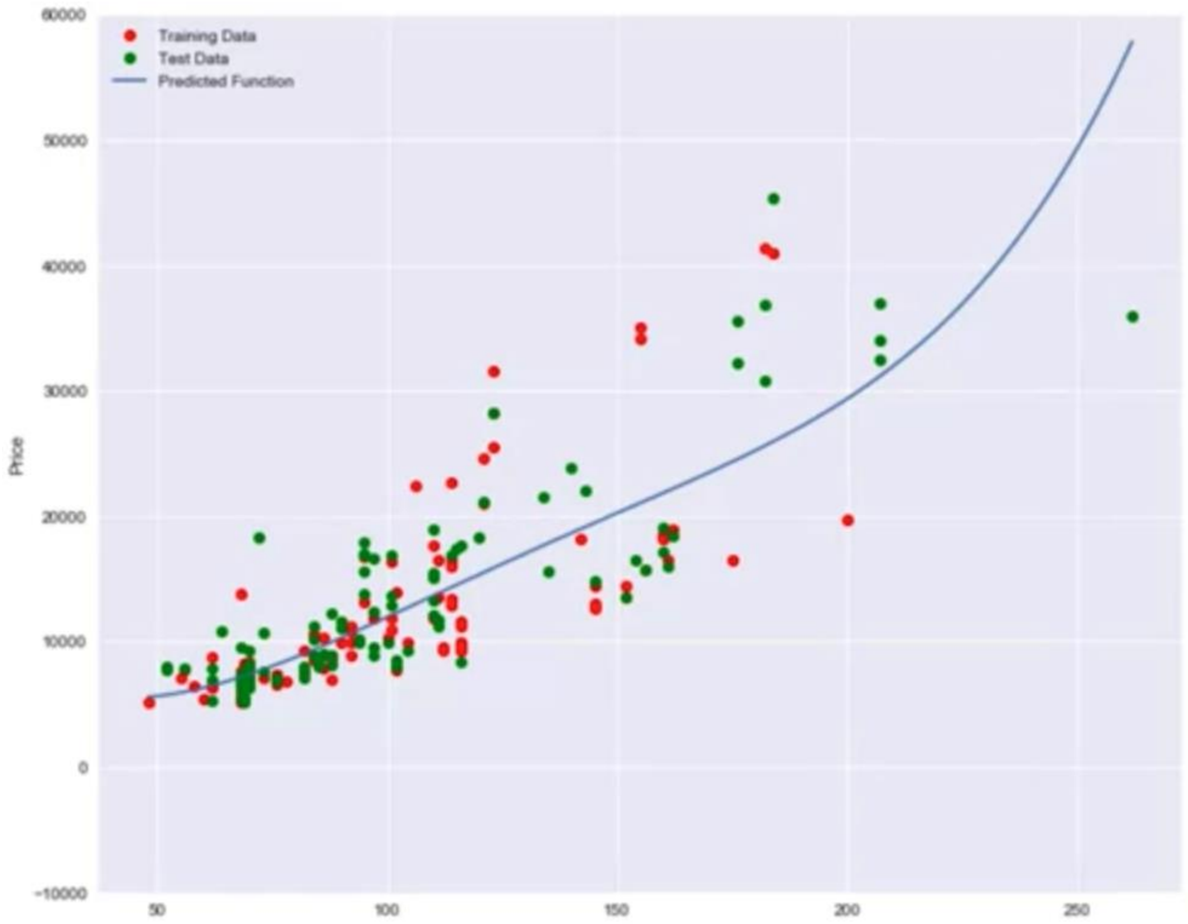
Model Selection: Example

Polynomial Regression with order = 1



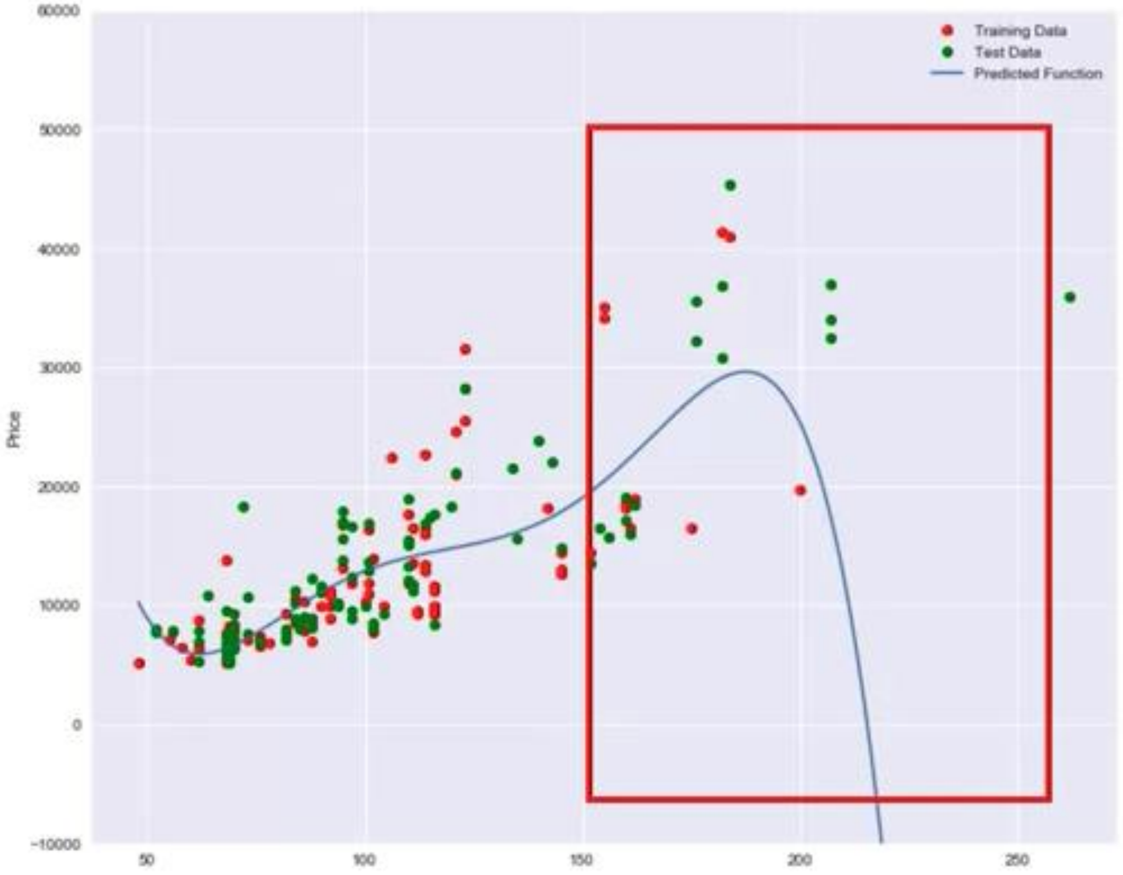
Model Selection: Example

Polynomial Regression with order = 2



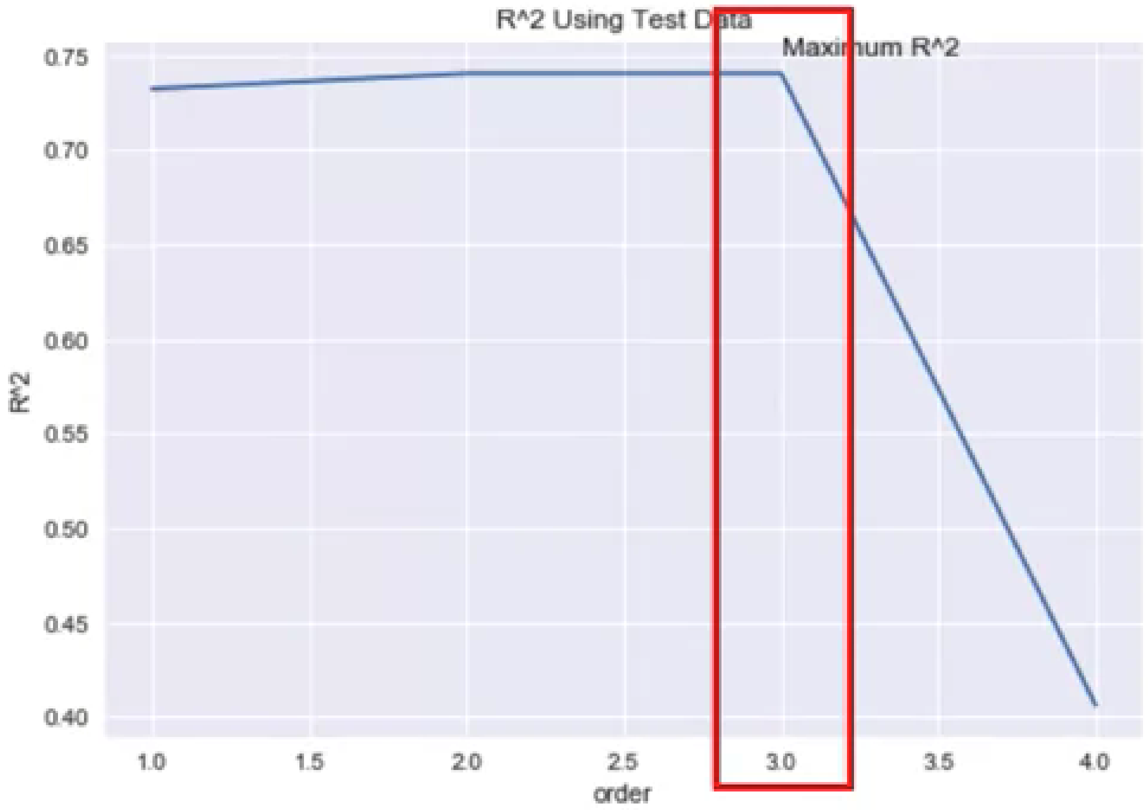
Model Selection: Example

Polynomial Regression with order = 4



Model Selection: Example

Plot R-squared with number of order in Polynomial Regression



How to prevent underfit/orverfit

Reasons for Underfitting

- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high bias
- The size of the training dataset used is not enough
- The model is too simple

Ways to Tackle Underfitting

- Increase the number of features in the dataset
- Increase model complexity
- Reduce noise in the data
- Increase the duration of training the data

Reasons for Overfitting

- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high variance
- The size of the training dataset used is not enough
- The model is too complex

Ways to Tackle Overfitting

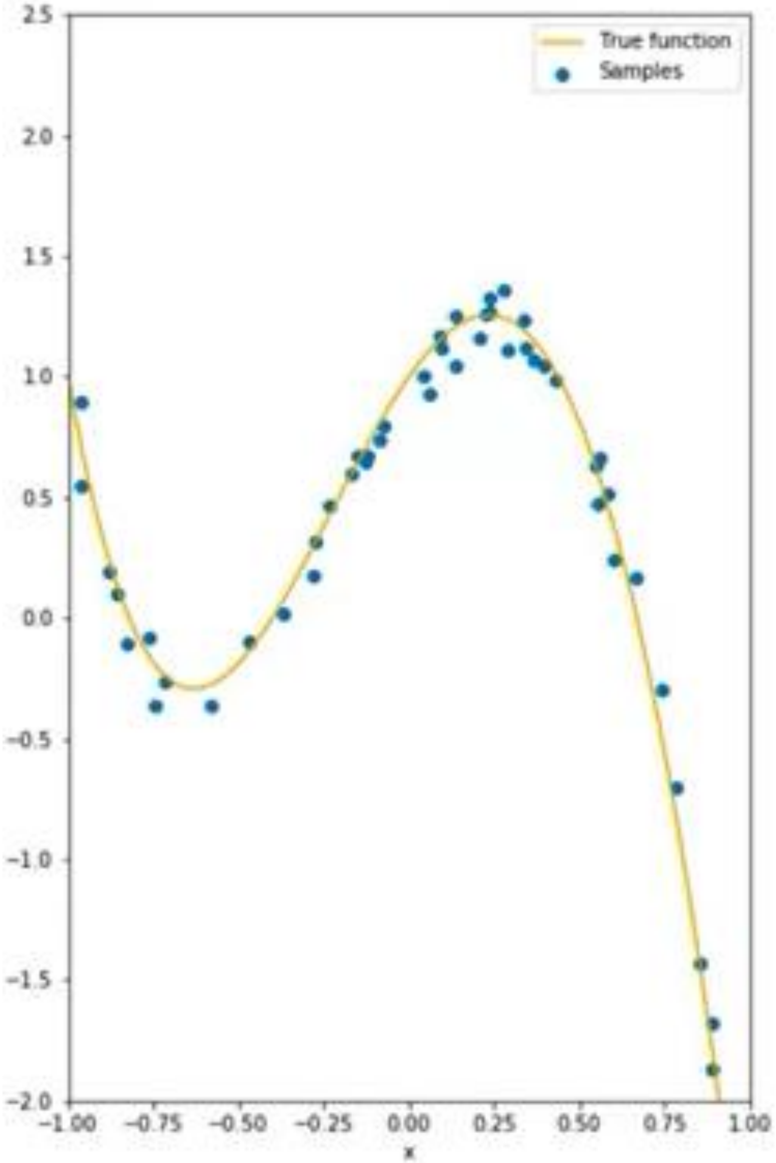
- Using K-fold cross-validation
- Using Regularization techniques such as Lasso and Ridge
- Training model with sufficient data
- Adopting ensembling techniques

Prevent Overfitting: Ridge Regression

- Ridge regression is a regression that is employed in a Multiple regression model when Multicollinearity occurs.
- Multicollinearity is when there is a strong relationship among the independent variables.
- Ridge regression is very common with polynomial regression.
- Ridge regression is good method to avoid over-fitting a regression model

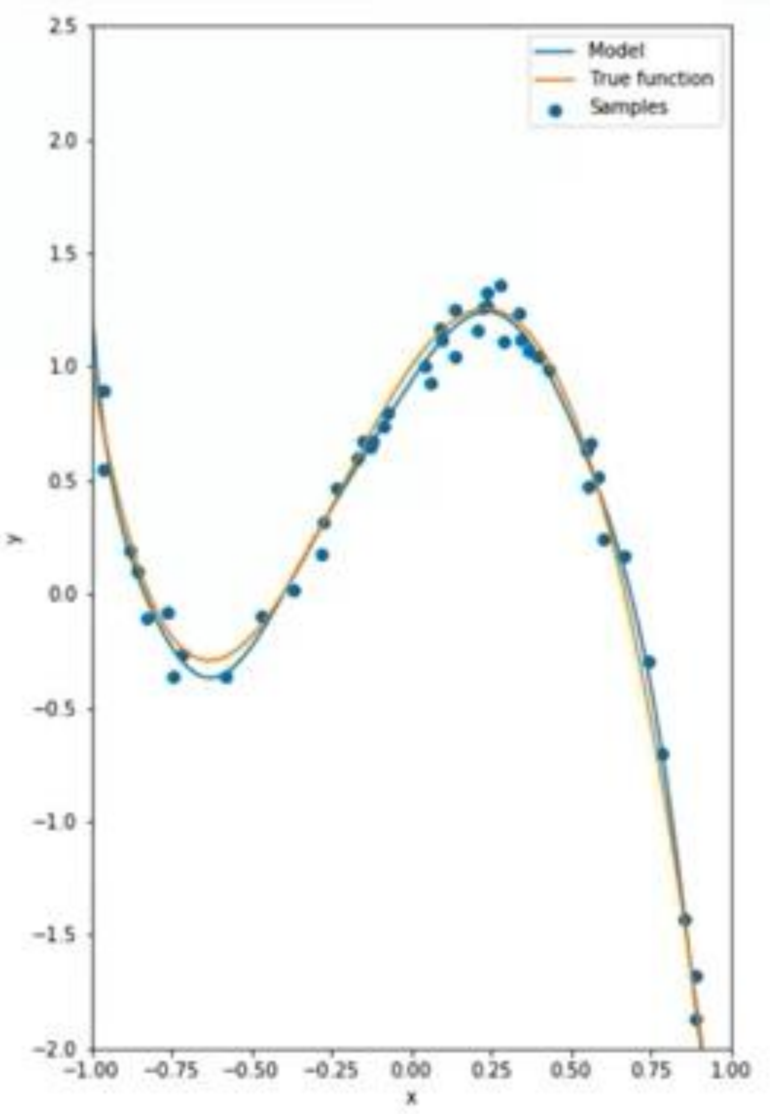
Ridge Regression

$$y = 1 + 2x - 3\,x^2 - 4x^3 + x^4$$



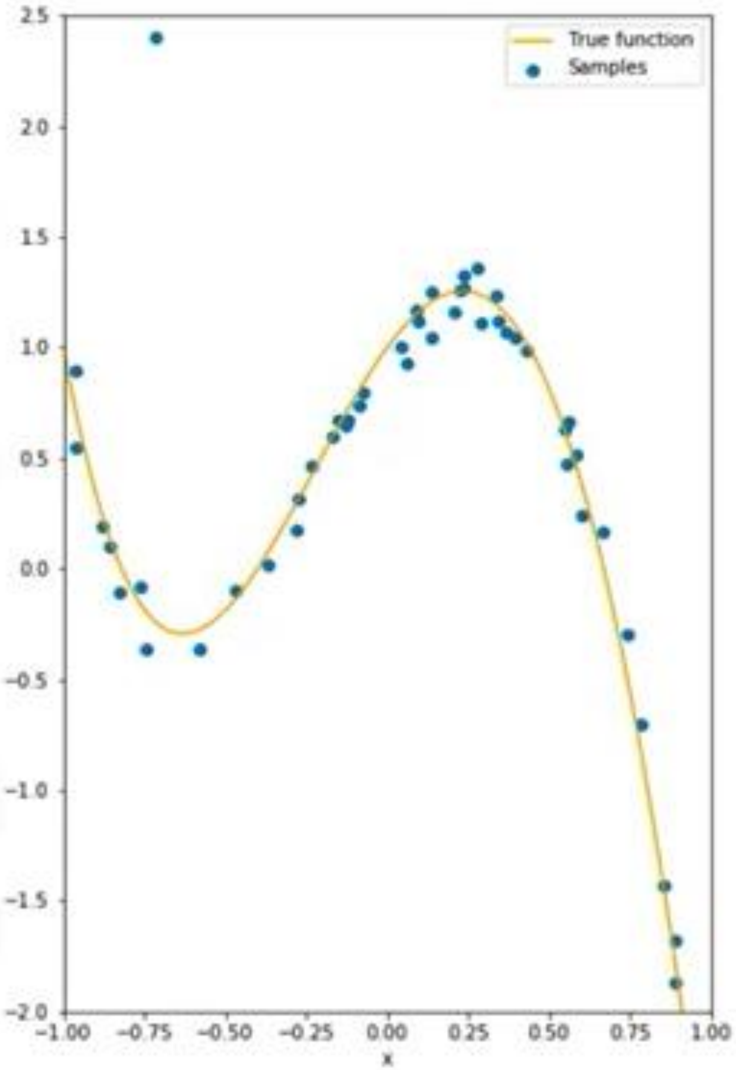
Ridge Regression

- We can use a 10th polynomial regres



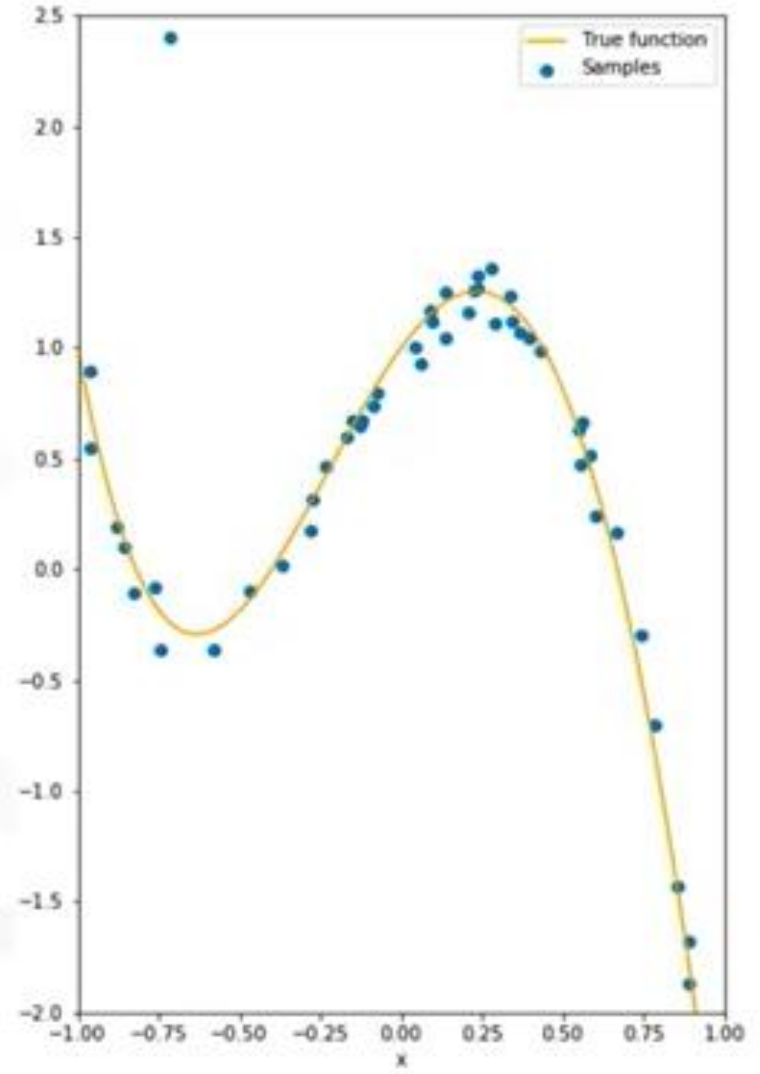
Ridge Regression

- Outlier data points



Ridge Regression

- Outlier data points
- In this case, if we use 10th polynomia ta, the estimated function is incorrect.



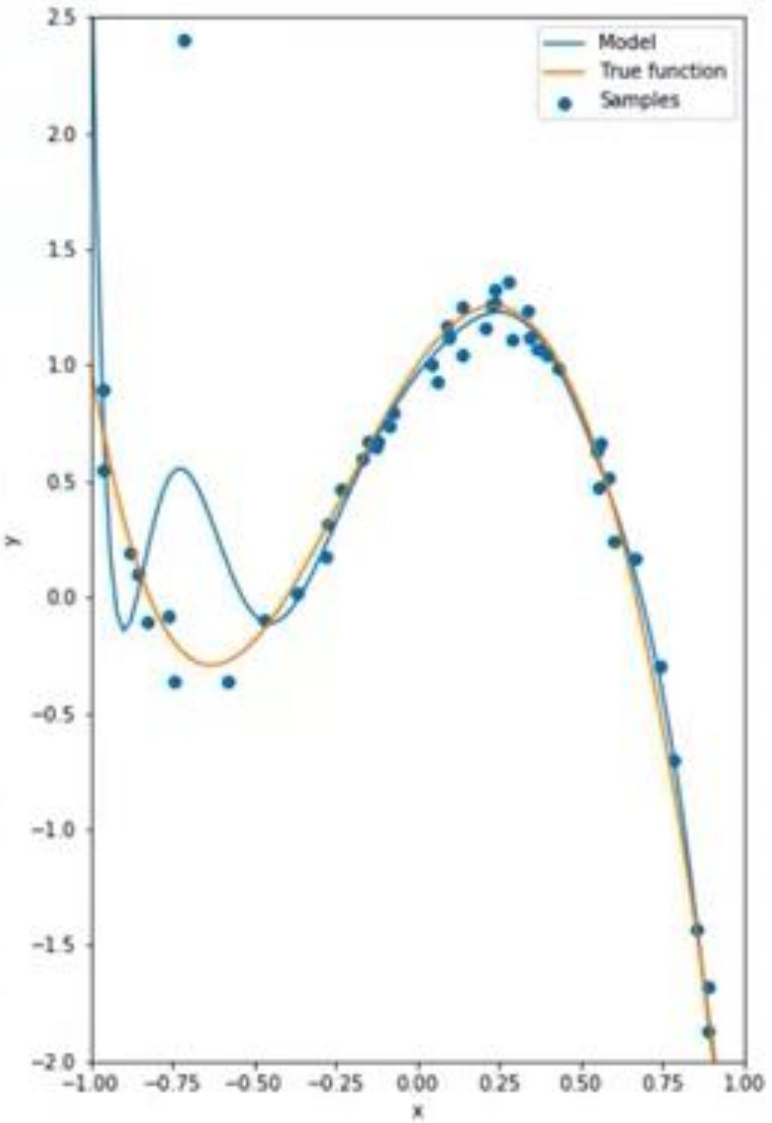
Ridge Regression

$$\hat{y} = 1 + 2\,x - 3\,x^2 - 2x^3 \boxed{- 12\,x^4 - 40x^5 + 80x^6 + 71x^7 - 141x^8 - 38x^9 + 75x^{10}}$$

Alpha	x	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}
0	2	-3	-2	-12	-40	80	71	-141	-38	75
0.001	2	-3	-7	5	4	-6	4	-4	4	6
0.01	1	-2	-5	-0.04	0.15	-1	1	-0.5	0.3	1
1	0.5	-1	-1	-0.614	0.70	-0.38	-0. 56	-0.21	-0.5	-0.1
10	0	-0.5	-0.3	-0.37	-0.30	-0.30	-0.22	-0.22	-0.22	-0.17

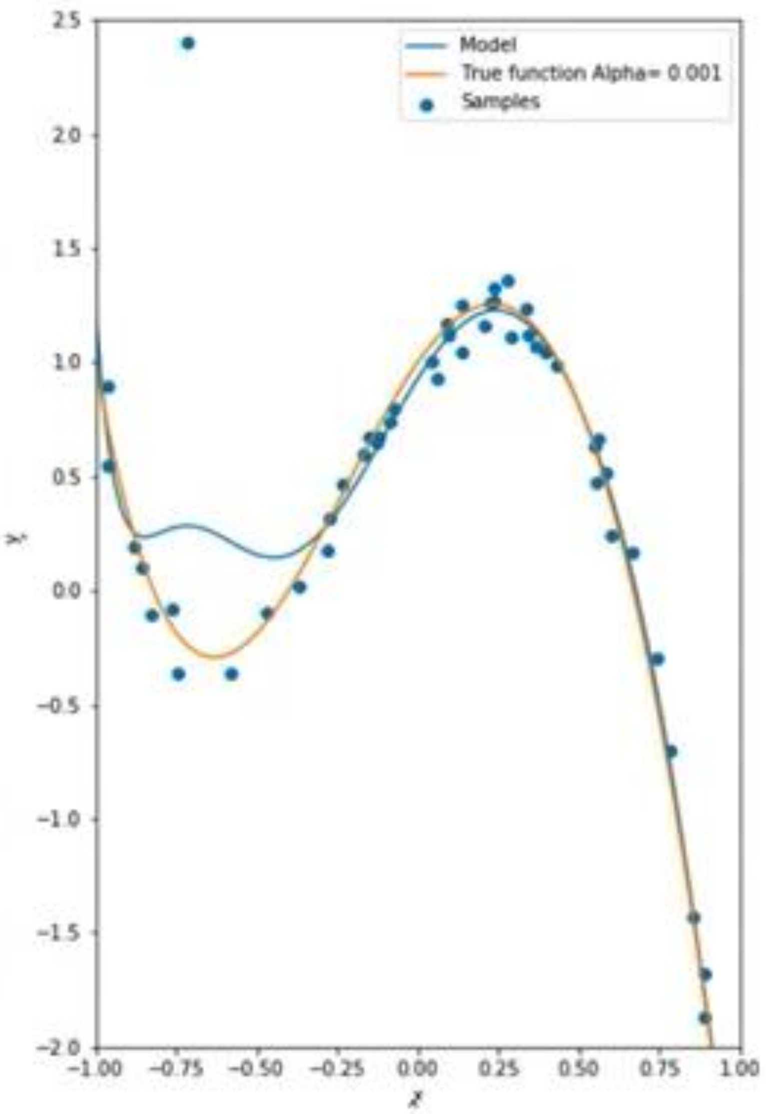
Ridge Regression

alpha
0
0.001
0.01
1
10



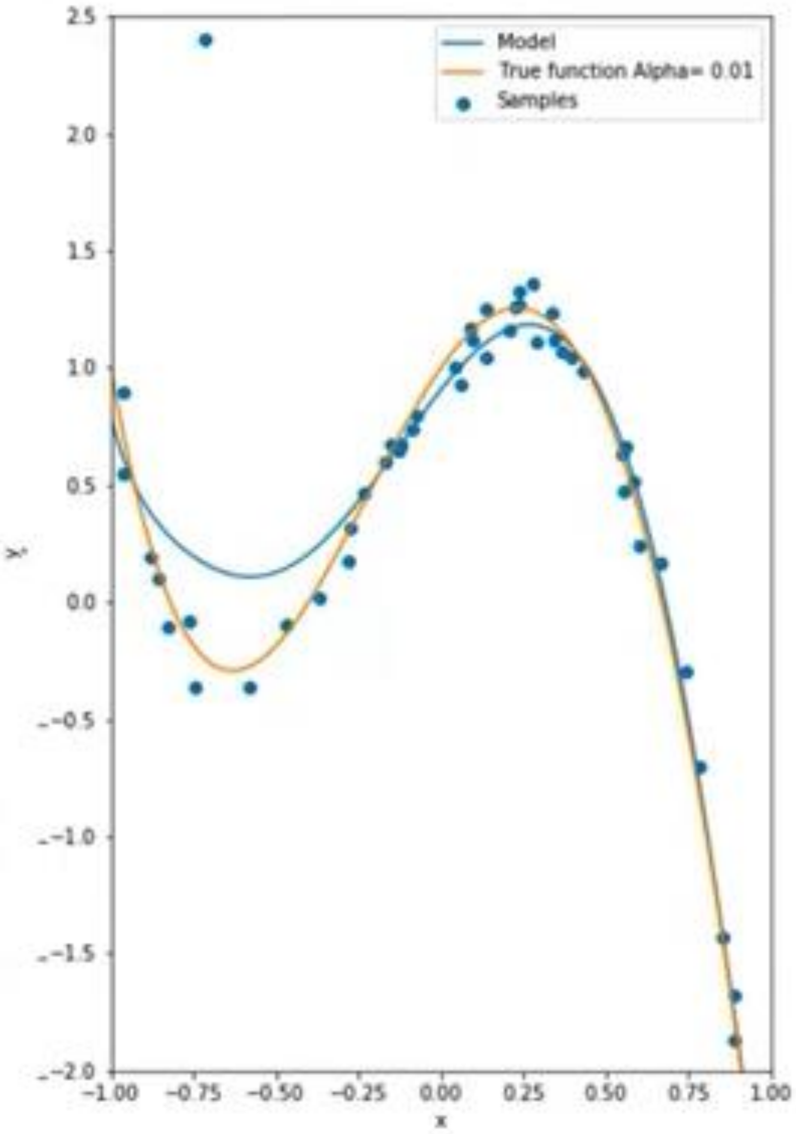
Ridge Regression

alpha
0
0.001
0.01
1
10



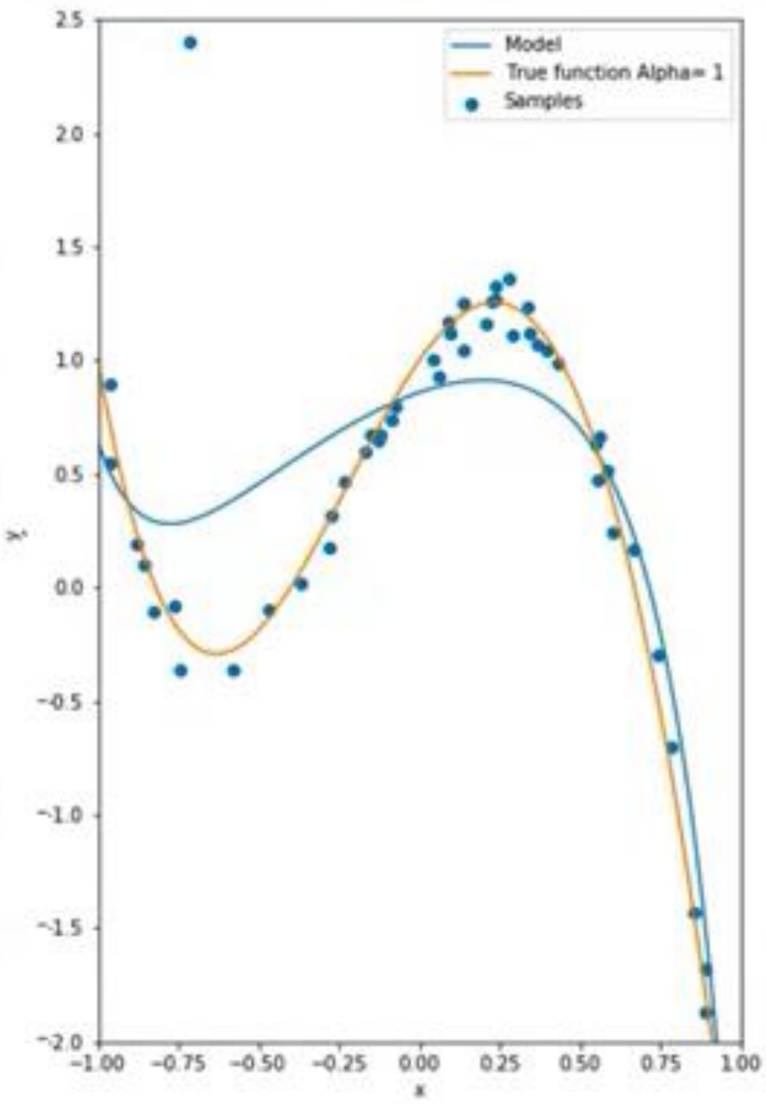
Ridge Regression

alpha
0
0.001
0.01
1
10



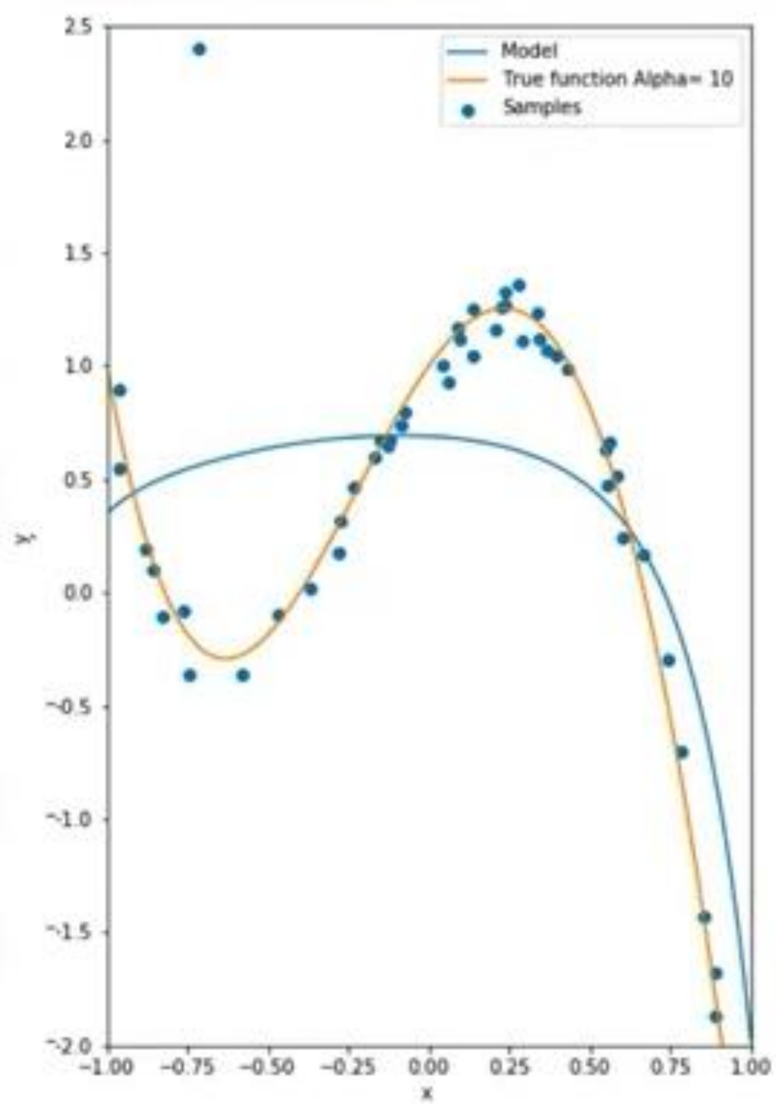
Ridge Regression

alpha
0
0.001
0.01
1
10



Ridge Regression

alpha
0
0.001
0.01
1
10



Ridge Regression in Scikit-learn

```
from sklearn.linear_model import Ridge

RidgeModel=Ridge(alpha=0.1)

RidgeModel.fit(X,y)

Yhat=RidgeModel.predict(X)
```

Activity – Hand-on Lab (~30 mins)

- Model Evaluation & Refinement

Lecture 10 & 11: Evaluating & Tuning Model

Wrap-up

Summary

In summary, in this lecture you learned:

- What are overfit and underfit and how to handle them
- Model evaluation

Thank You !

TRƯỜNG ĐẠI HỌC NGOẠI THƯƠNG
Foreign Trade University