Bias-Variance Trade-Off & Polynomial Regression Session-3

SUMMARY of PREVIOUS CLASS

- Residuals
  - Sum and mean of the Residuals are always Zero
  - Residuals are normally distributed for Suitable Linear Regression.
- Regression Error Metrics
  - MAE, MSE, RMSE...
- Scikit-learn Library and ML
  - 5 Steps: Import, Split Data, Model Building and Fit, Prediction, Evaluation

SUMMARY of PREVIOUS CLASS

Model Building

1- Import Library

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

[18] ✓ 0.1s

2- Data Preparation / Train-Test Split

```
df = pd.read_csv("Advertising.csv")
X = df.drop(columns ="sales") #df[["TV", "radio", "newspaper"]]
y = df["sales"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state =42)
```

[19] ✓ 0.1s

3- Model Building and Fitting

```
model = LinearRegression()
model.fit(X_train, y_train)
```

4- Prediction

```
y_pred = model.predict(X_test)
```

[21] ✓ 0.5s Python

5- Evaluation

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
R2_score = r2_score(y_test, y_pred)
print("Model Performance Evaluation:")
print("--------------------------")
print(f"R2_score \t: {R2_score}")
print(f"MAE \t\t: {mae}")
print(f"MSE \t\t: {mse}")
print(f"RMSE \t\t: {rmse}")
```

[22] ✓ 1.1s Python

```
Model Performance Evaluation:
--------------------------
R2_score        : 0.8609466508230368
MAE             : 1.5116692224549086
MSE             : 3.796797236715219
RMSE            : 1.9485372043446385
```

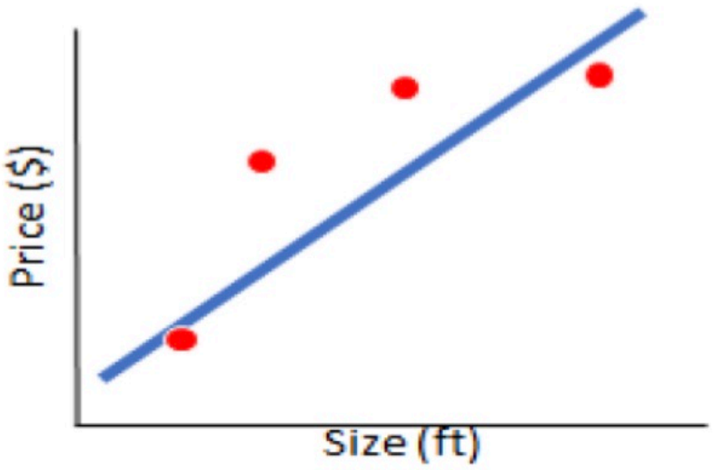
Introduction to Bias-Variance Trade-Off

Underfitting and Overfitting Problems

Training Error vs. Validation Error (Test Error)

Polynomial Regression

Introduction to Bias-Variance Trade-Off

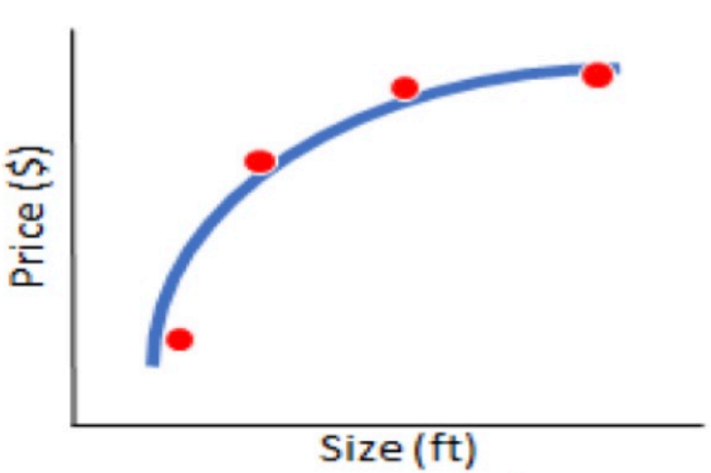


Models with **too few** parameters may not fit the data well **(high bias)**,

but the scores in the test set are the same as in the train set. **(low variance)**

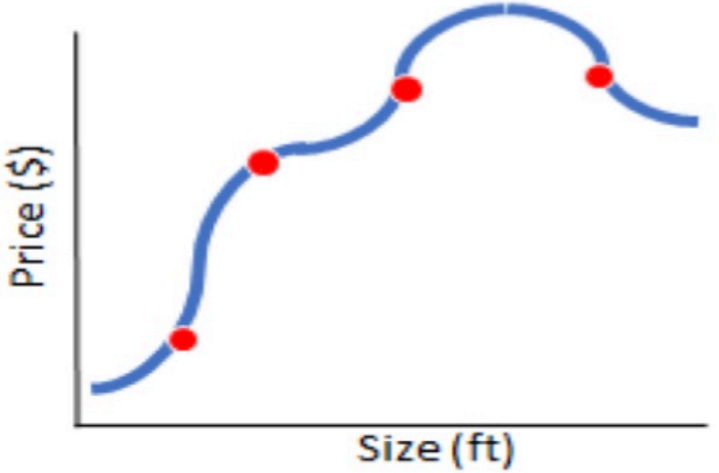
Simple

UNDERFIT



Just Right

Ideally, we would like to choose a model that both accurately **captures the regularities** in the training data, but also **generalizes well** to unseen data. **(low bias & low variance)**



Models with **too many** parameters may fit the ***training data*** well **(low bias)**,

but the scores on the test set are much lower than on the train set **(high variance)**

Complex

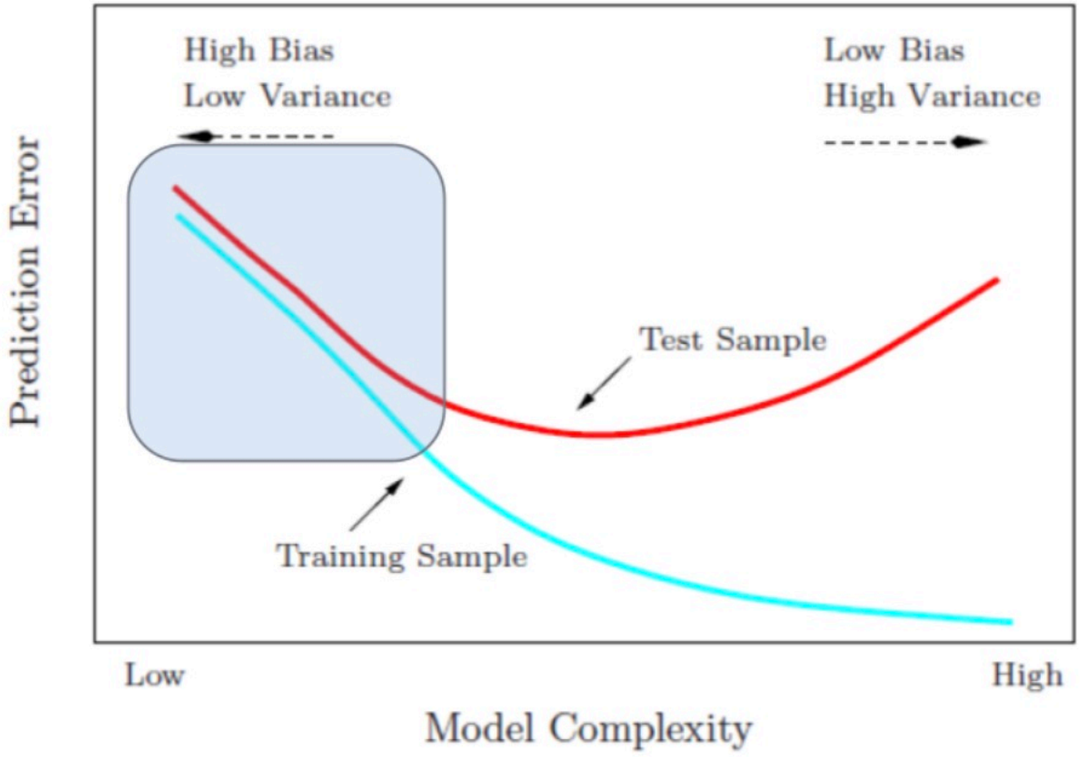
OVERFIT

Underfitting and Overfitting Problems

HOW TO RECOGNIZE ?

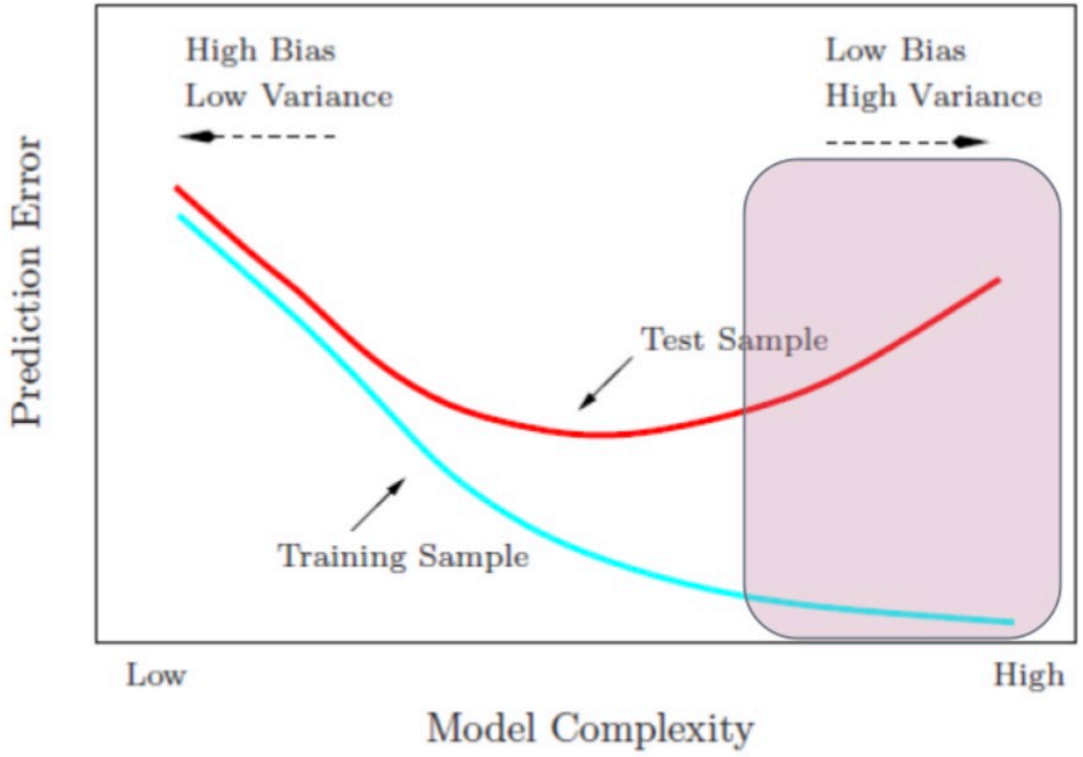
UNDERFITTING

High training error and **high** test error

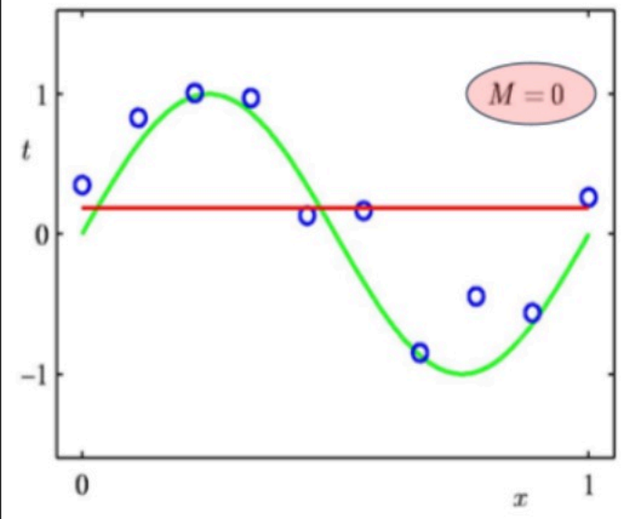


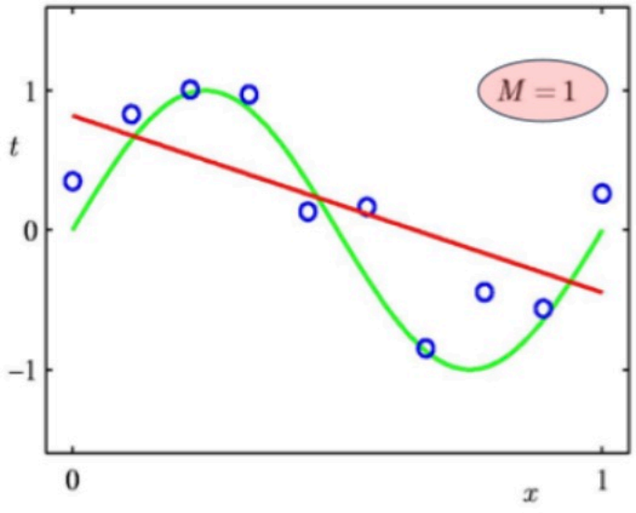
OVERFITTING

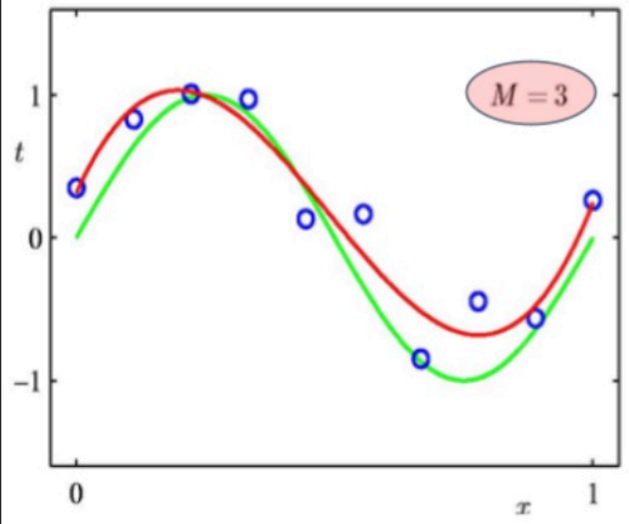
Low training error and **high** test error

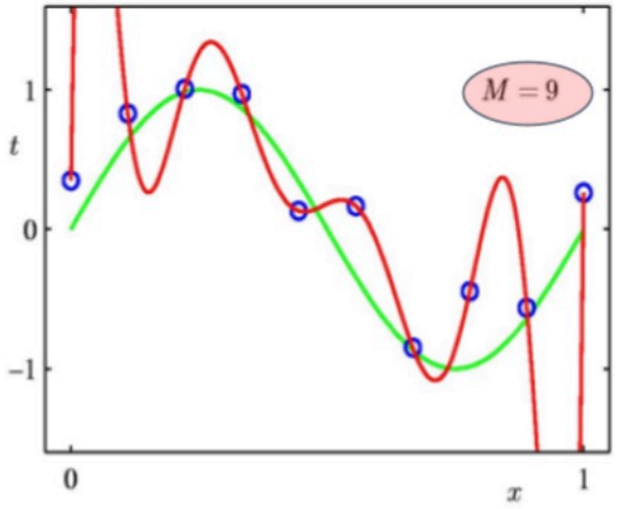


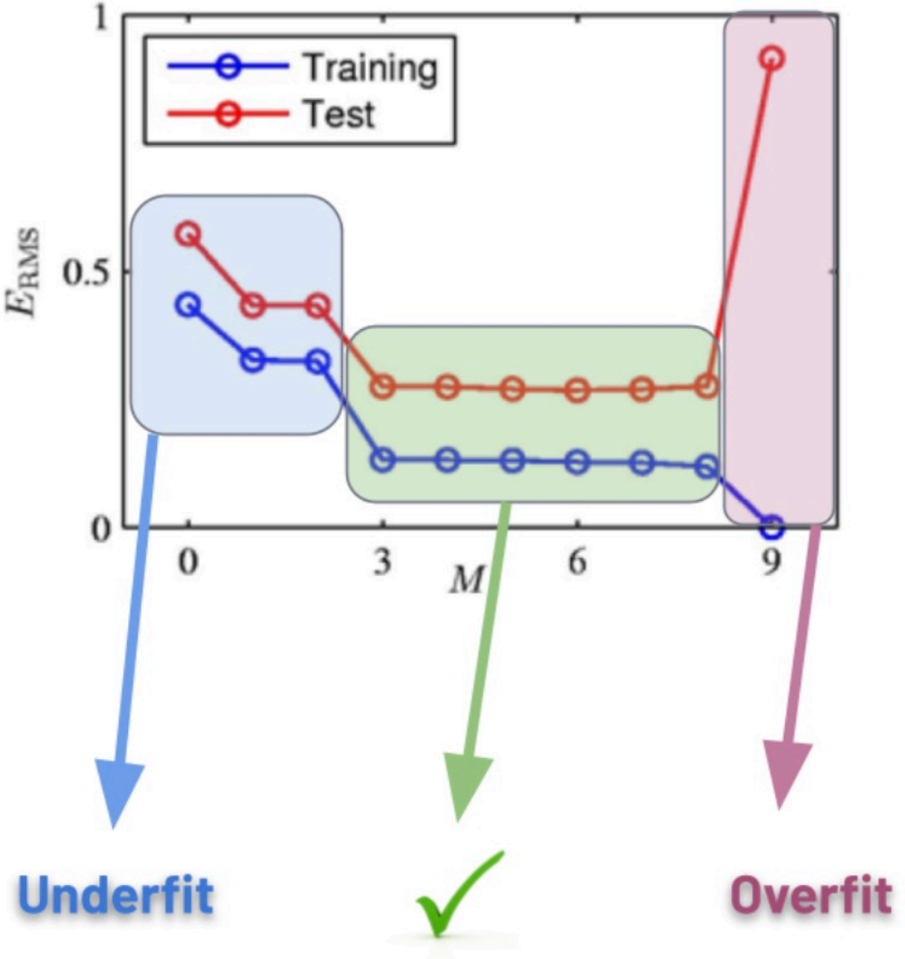
Underfitting and Overfitting Problems



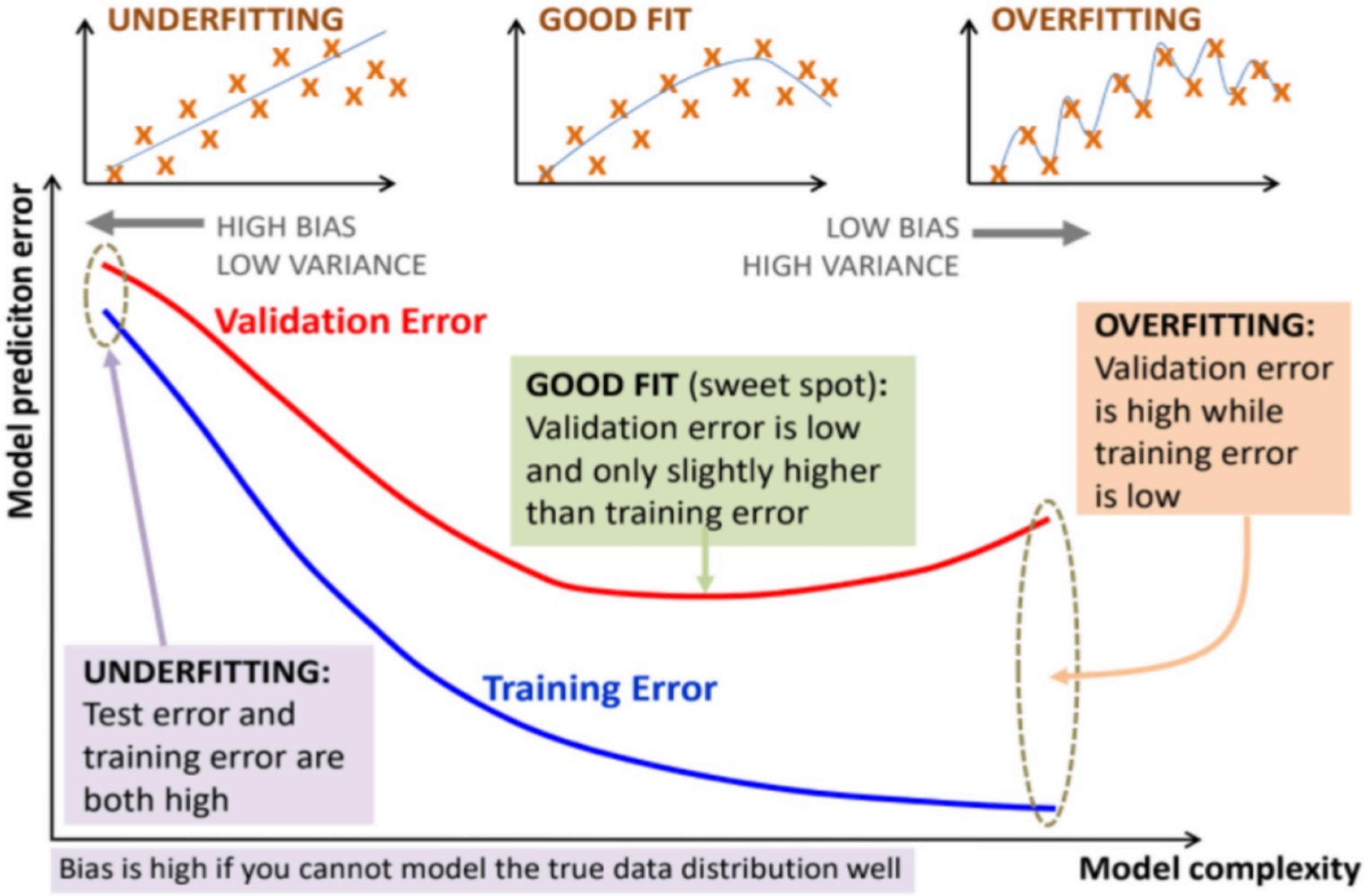








Training Error vs. Validation Error (Test Error)

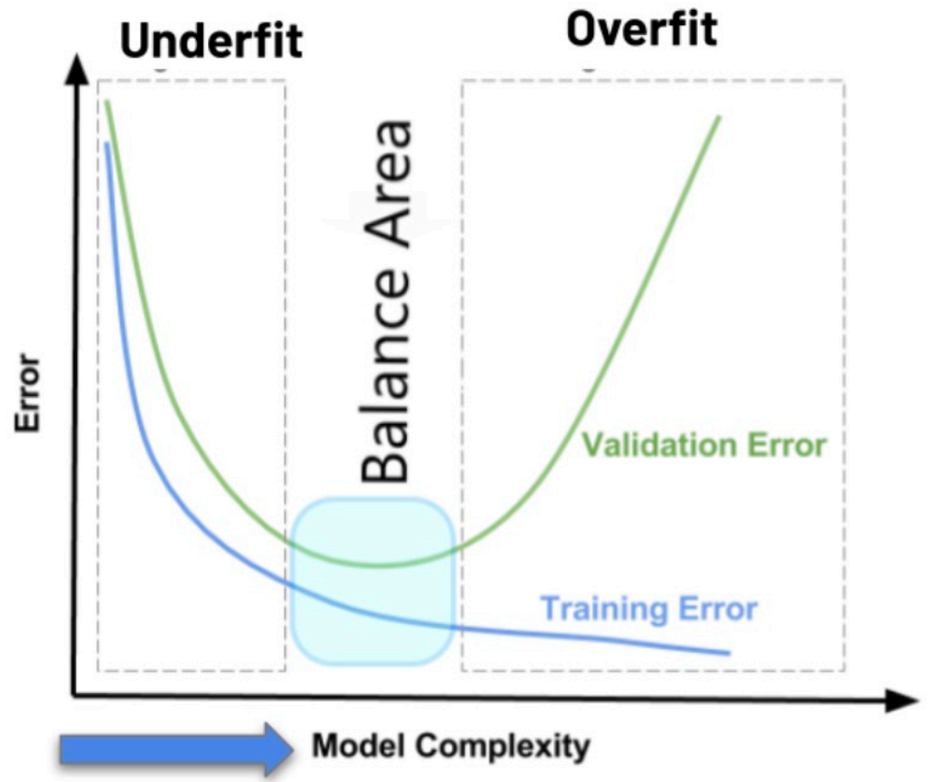


Underfitting and Overfitting Problems

HOW TO DEAL WITH

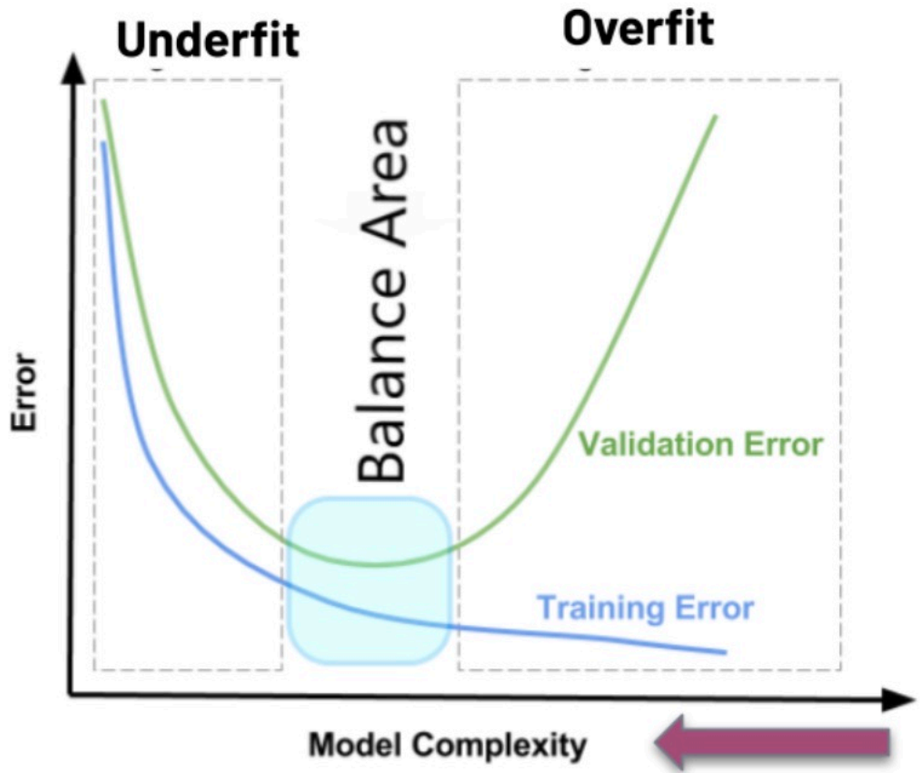
UNDERFITTING ?

- Find a **more complex** model



OVERFITTING ?

- Decrease the number of parameters
- More training data / **Cross Validation**
- **Regularization (Lasso&Ridge)**



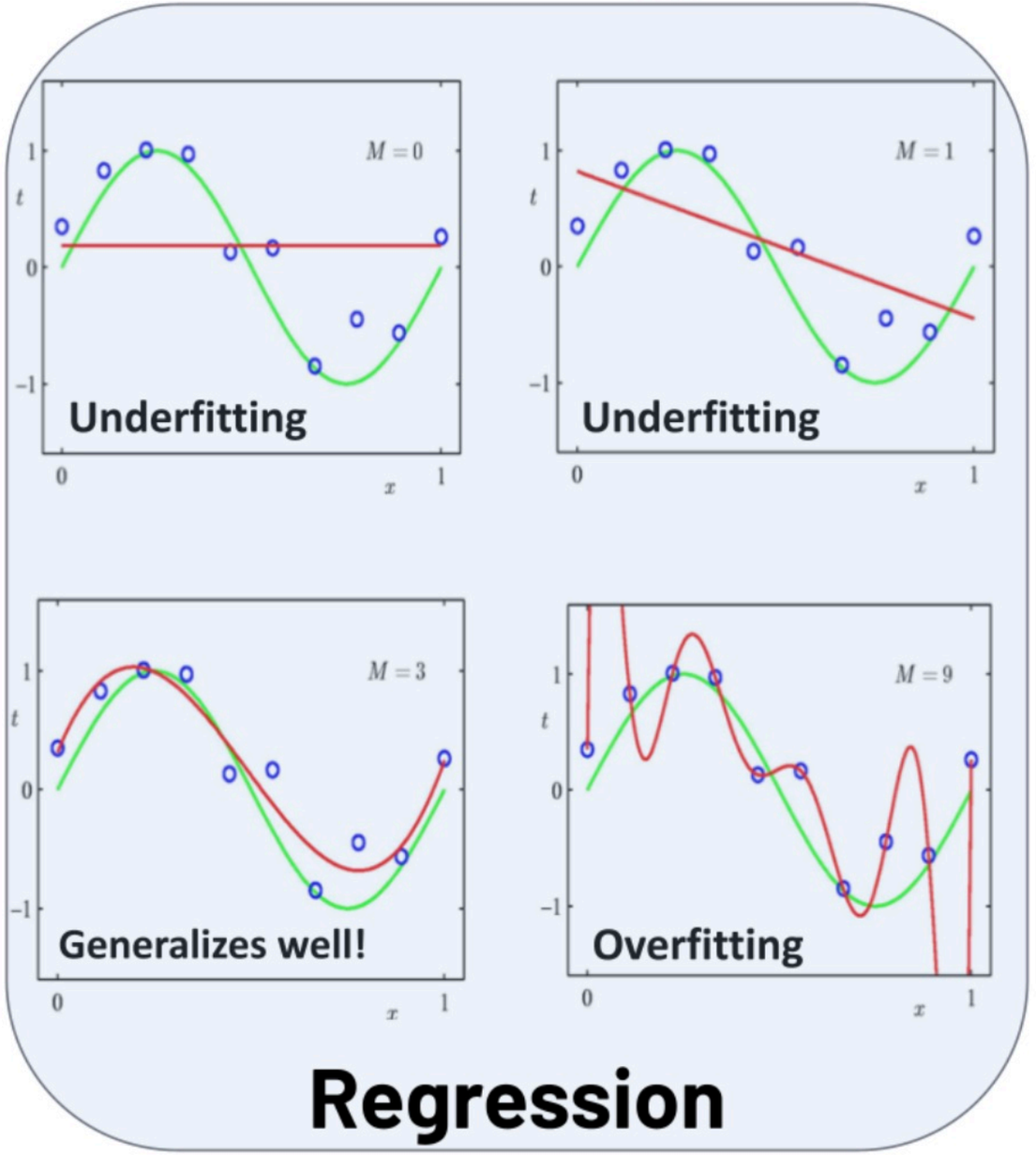
Underfitting and Overfitting Problems

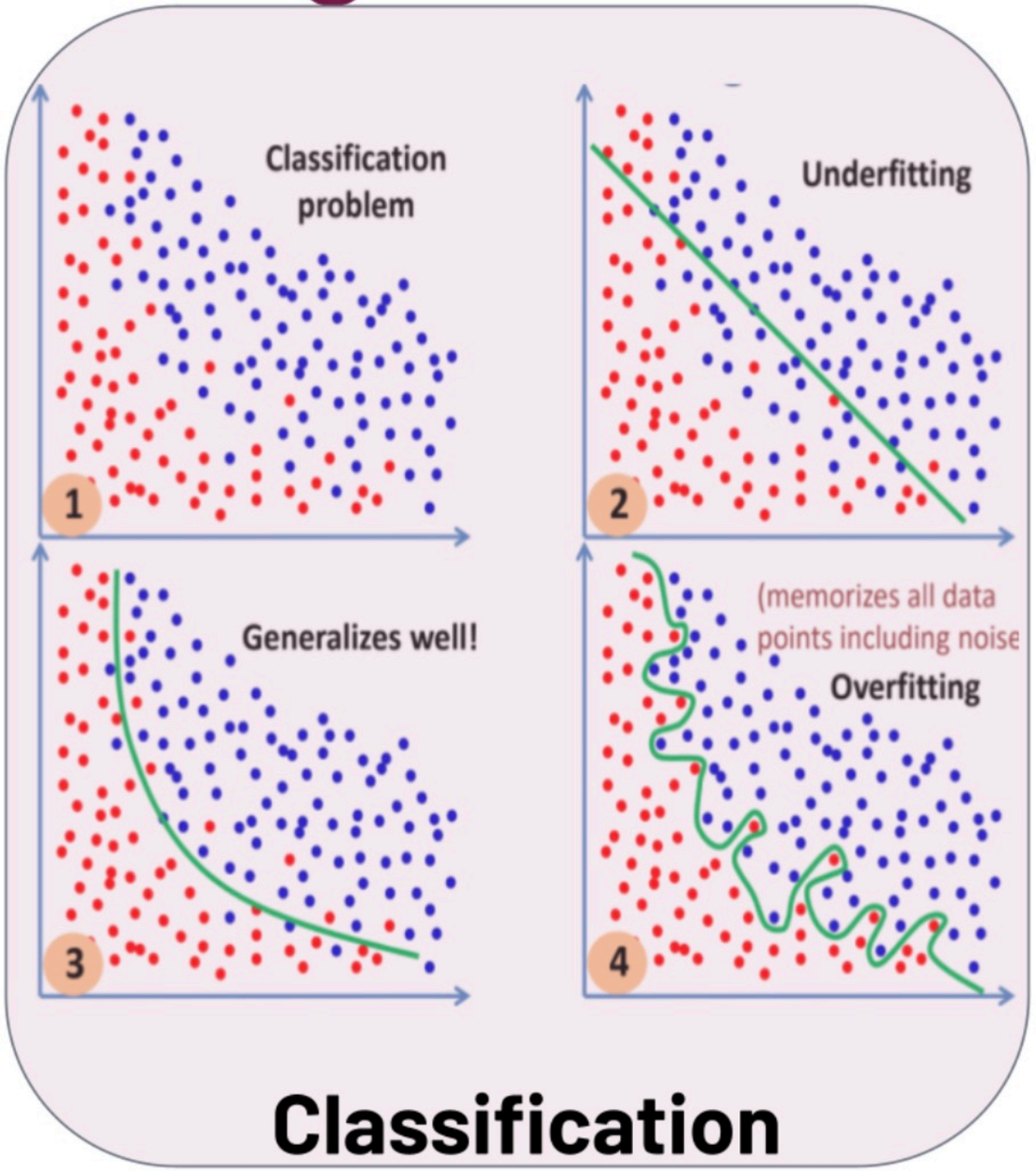
Summary

	MODEL				
	Bias	**Variance**	**Complexity**	**Flexibility**	**Generalizability**
Underfitting: An overly simple model	High	Low	Low	Low	High
Overfitting: Modeling the noise as well	Low	High	High	High	Low

Ref: https://cambridgecoding.wordpress.com/2016/03/24/misleading-modelling-overfitting-cross-validation-and-the-bias-variance-trade-off/

Underfitting and Overfitting Problems



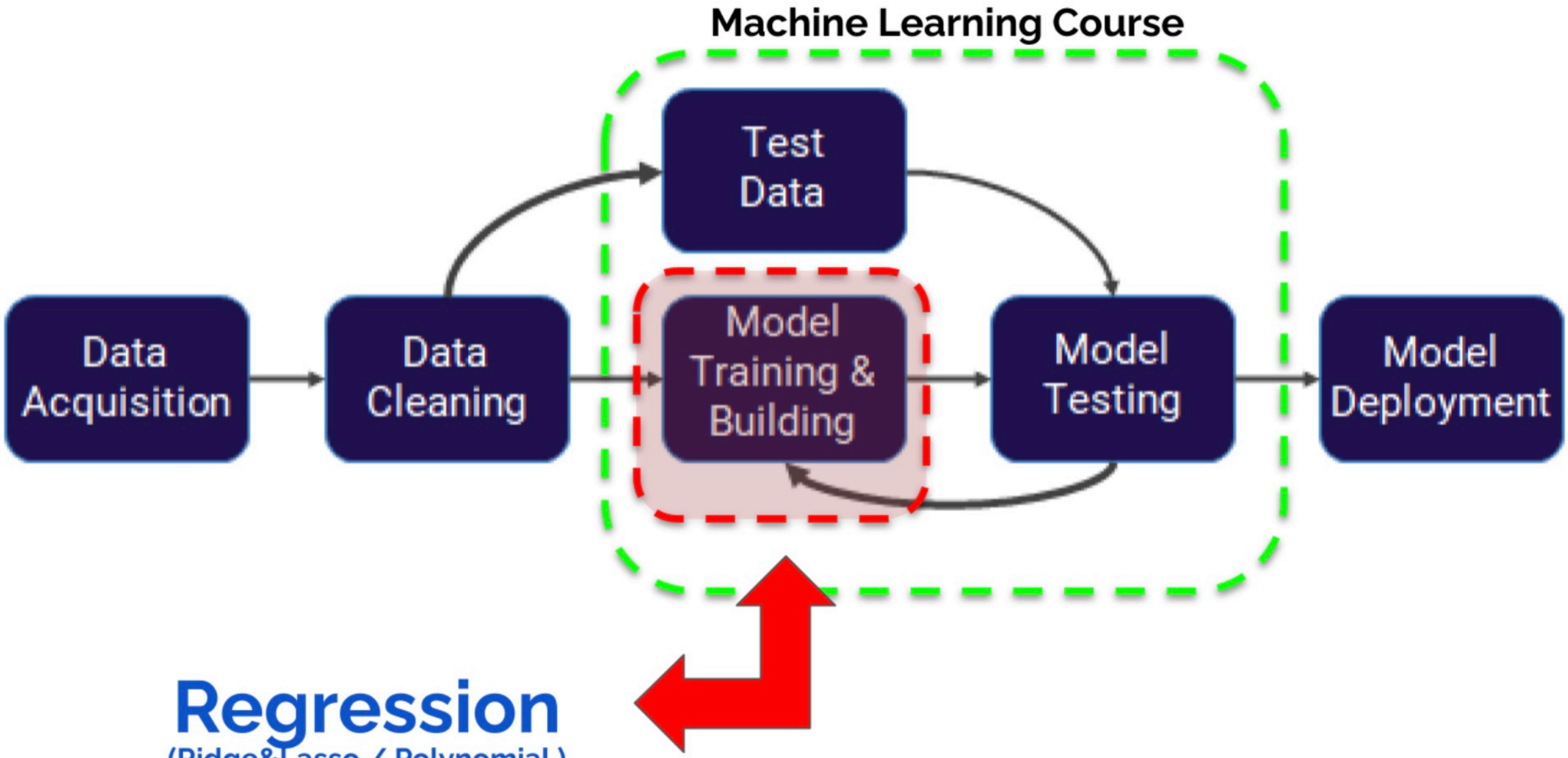


Regression (cont.)

Session-3

Polynomial Regression

Where are we?


Machine Learning Course
Test Data
Data Acquisition
Data Cleaning
Model Training & Building
Model Testing
Model Deployment
Regression
(Ridge&Lasso / Polynomial)

Polynomial Regression

Types of Regression Models

Simple Linear Regression

A linear regression model that estimates the relationship between ***one independent*** variable and ***one dependent*** variable using a straight line.

$$Y = \theta_0 + \theta_1 x$$

Multi Linear Regression

A linear regression model that estimates the relationship between ***several independent*** variables (features) and ***one dependent*** variable.

Polynomial Regression

A special case of multiple linear regression. It also works with non linear relationship. The relationship between the independent variable x and dependent variable y is modeled as an **nth degree polynomial** in x.

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2$$

Advantages:

- *Give info about the relevance of the features*
- *Works good irrespective of data size*

Disadvantages:

Assumptions of linear regression

Advantages:

Works good on nonlinear probs.

Disadvantages:

Need to choose right polynomial degree for good bias/variance trade off.

Sources : www.geeksforgeeks.org & Towards DS (Animesh Agarwal)

Polynomial Regression

Linear regression: $Y = \beta_0 + \beta_1 X$

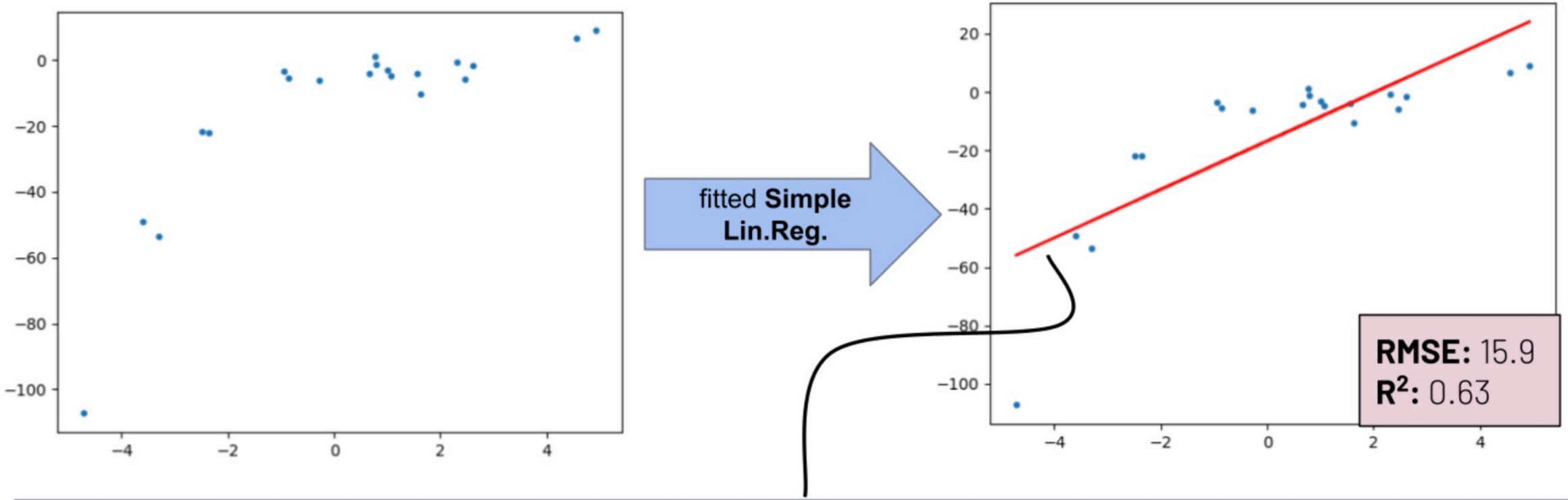
Linear regression: $Y = \beta_0 + \beta_1 X^2$

Nonlinear regression: $Y = \beta_1 X^{\beta_2}$

Nonlinear regression: - $Y = \beta_0 + (\beta_2 - \beta_1) e^{-\beta_3 X^{\beta_4}}$

Nonlinear regression: $Y = \beta_1 cos(X_1 + \beta_2) + \beta_3 cos(2X_2 + \beta_4)$

Polynomial Regression

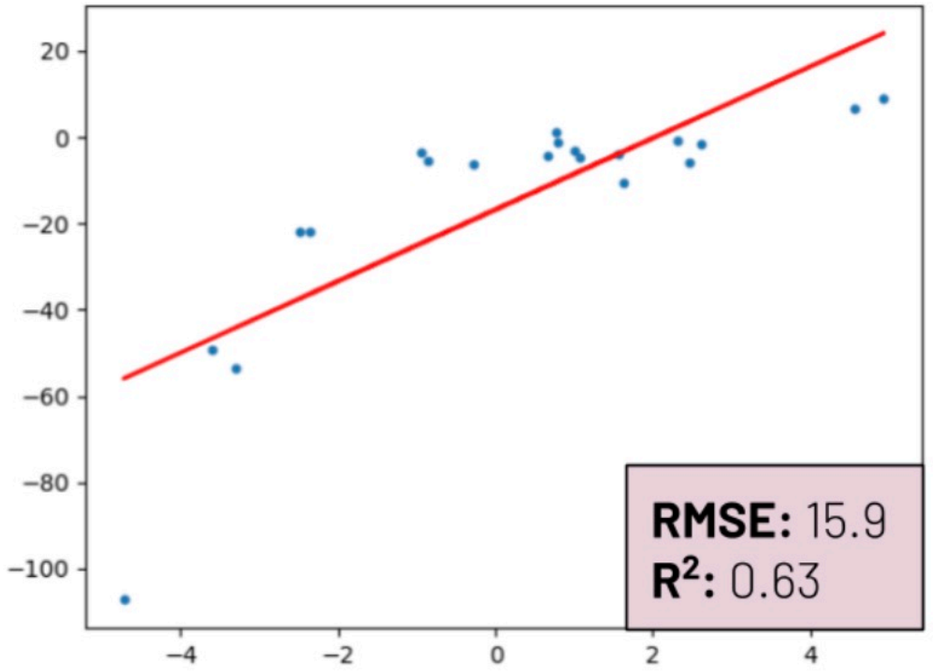


The regression line is **unable to capture the patterns** in the data. This is an example of **under-fitting**.

To overcome under-fitting, we need to **increase the complexity** of the model.

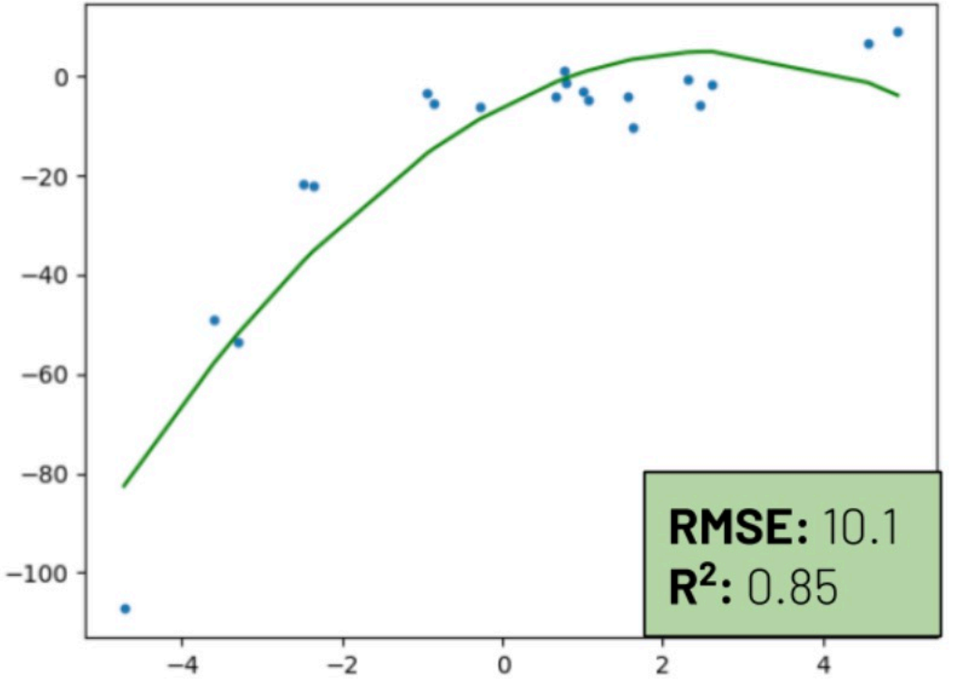
Polynomial Regression

To generate a higher order equation we can **add powers of the original features as new features.**



$$Y = \theta_0 + \theta_1 x$$

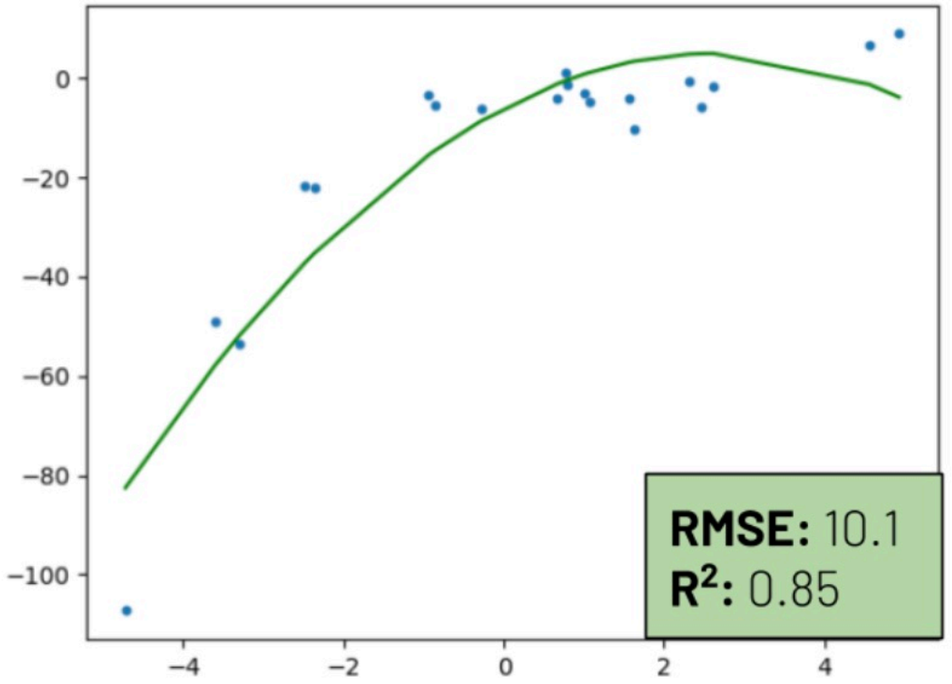
added **powers of features**



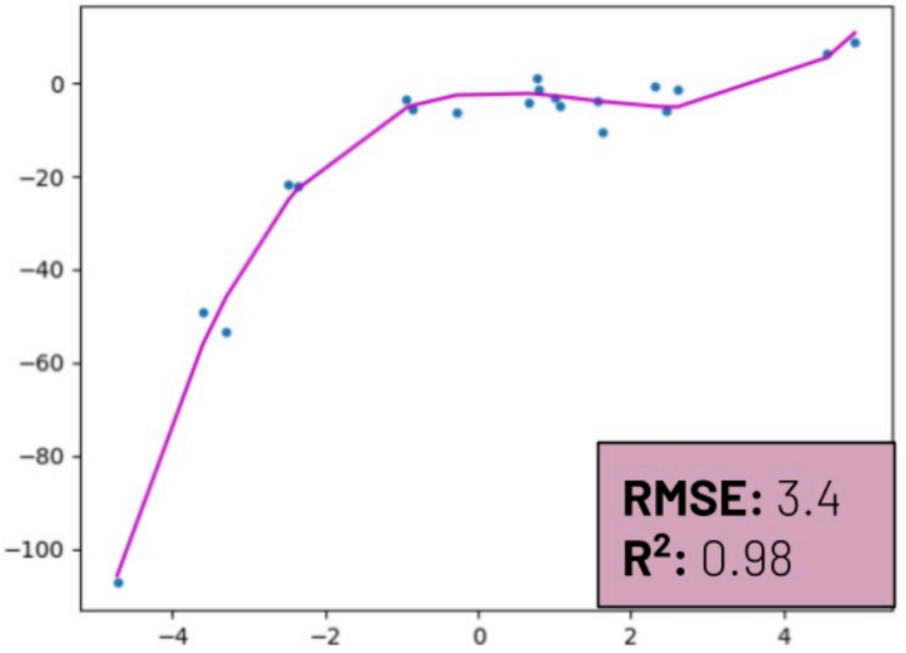
$$Y = \theta_0 + \theta_1 x + \theta_2 x^2$$

Polynomial Regression

If we try to **fit a cubic curve (degree=3)** to the dataset, we can see that it is **more successfully generalized** than the quadratic (degree=2) and the linear plots (degree=1).

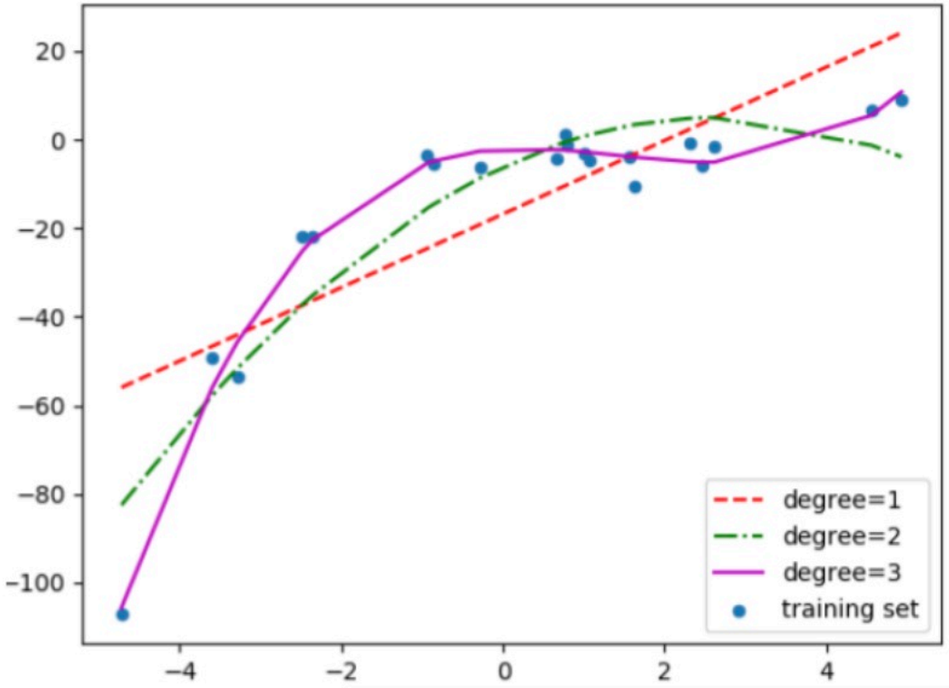


fit a cubic curve (degree=3)

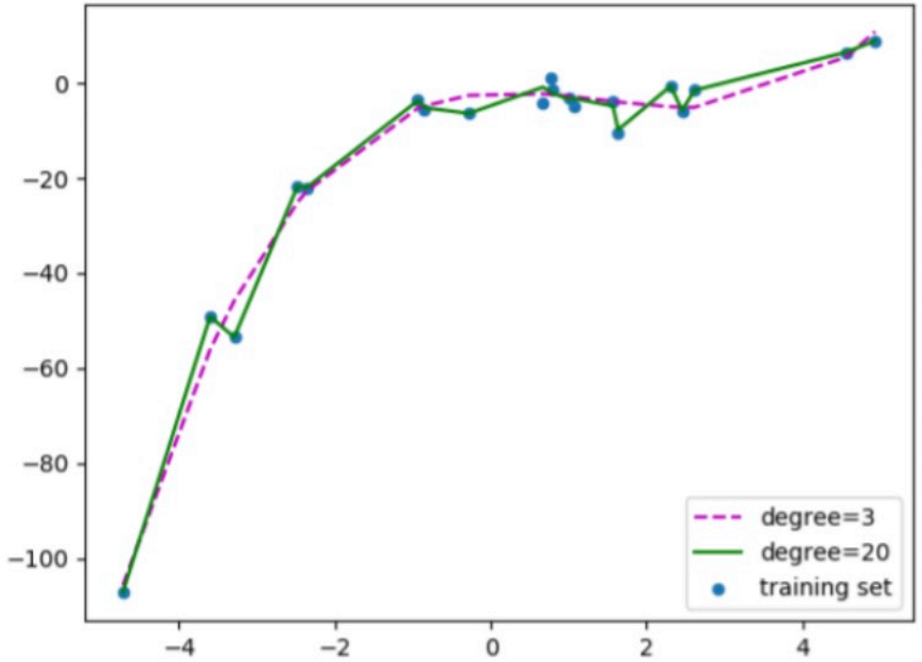


Polynomial Regression

If we further **increase the degree to 20**, we can see that the curve memorize the data points. This is an example of **overfitting**.



degree 1,2,3 to 20



How do we choose an optimal degree?

Bias Variance Trade-off