AI in Mathematics
Lecture 2
Classic ML. Part 1.

Bar-Ilan University
Nebius Academy | Stevens Institute of Technology

March 25, 2025

About This Course

~~1 week: Intro~~

2 weeks: Classic ML

2 weeks: Deep Learning in Mathematics

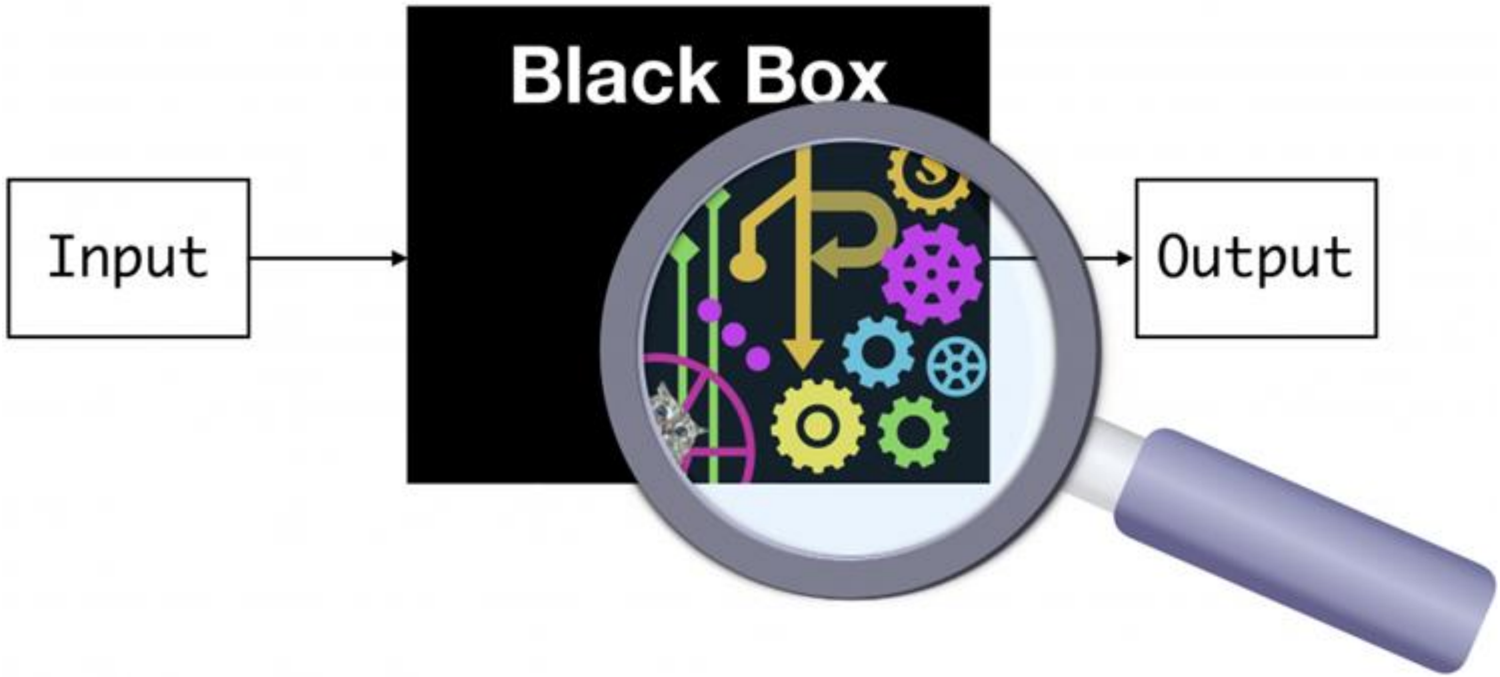
3 weeks: Math as an NLP problem (LLMs etc.)

3 weeks: Reinforcement Learning (RL) in Math

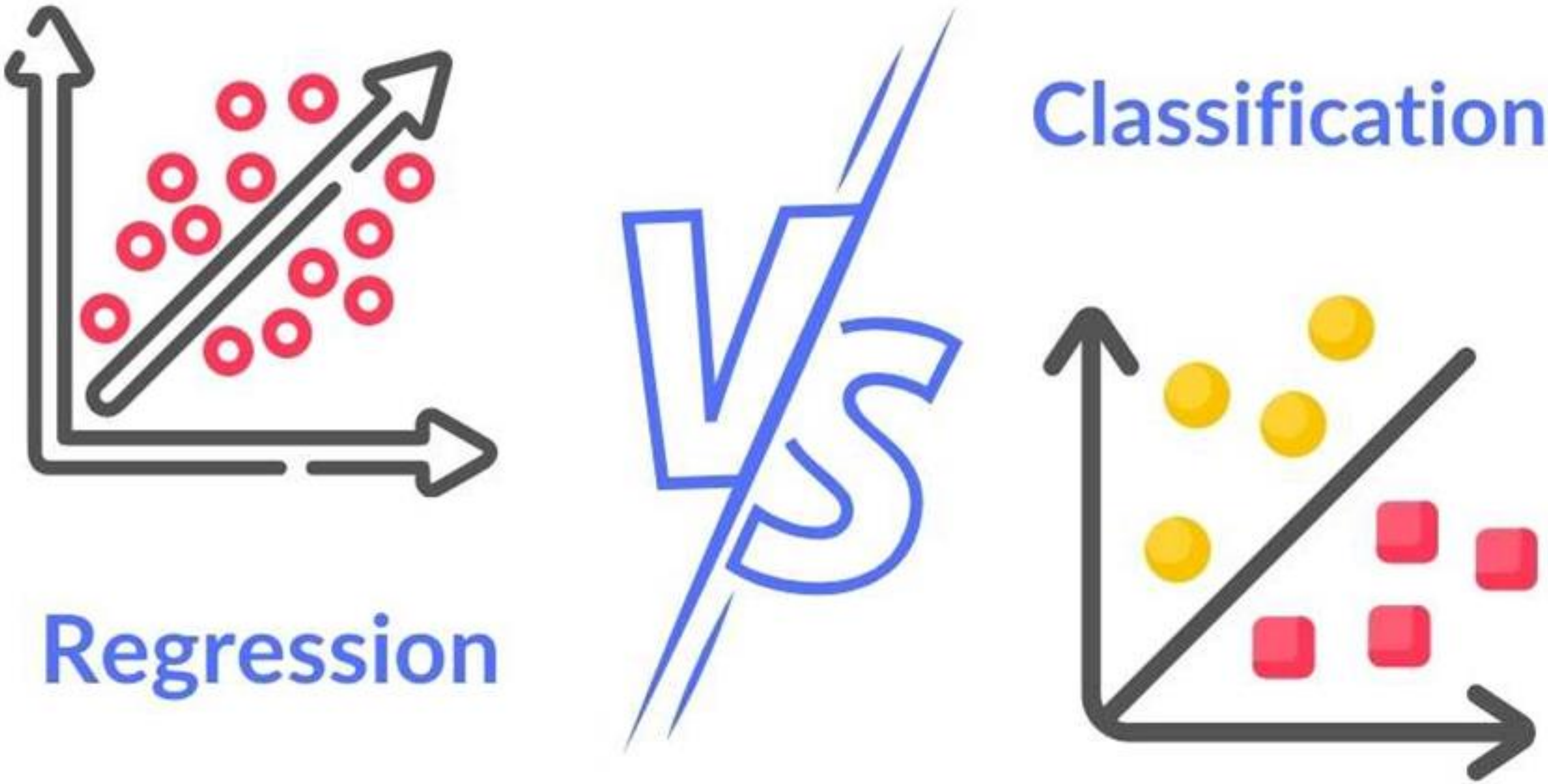
1 week: Advanced AI topics

1 week: Project Presentations

Machine Learning

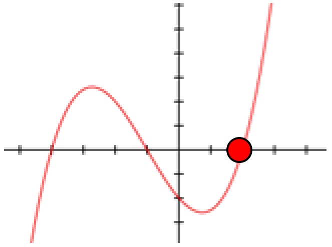


Regression and Classification

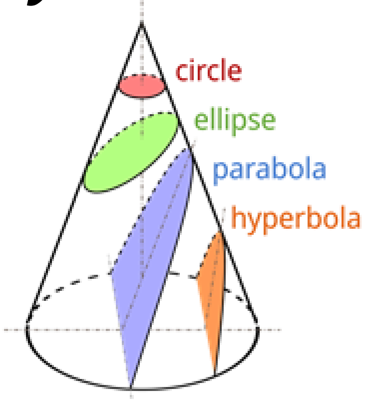


Regression and Classification

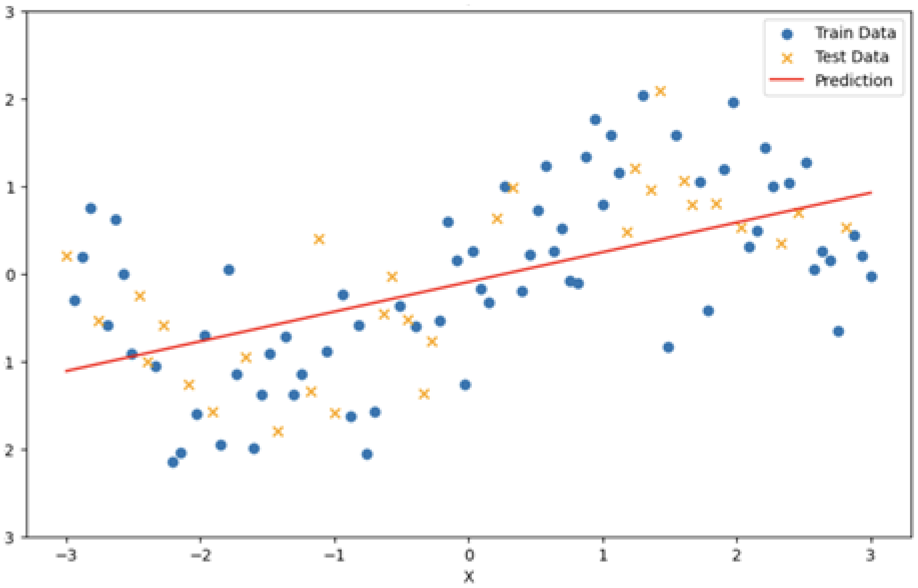
Regression task: What is the largest root of polynomial $Ax^3 + Bx^2 + Cx + D = 0$?



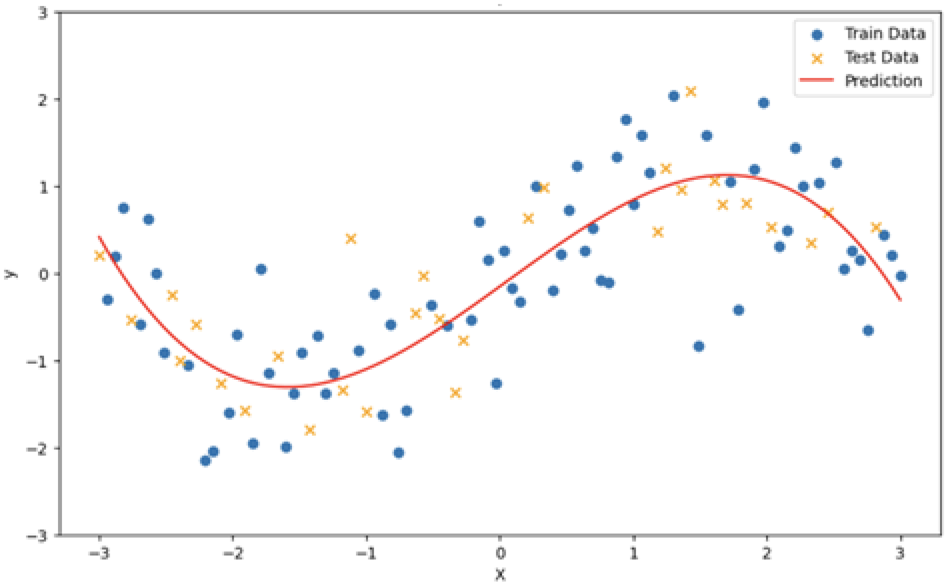
Classification task: What type of quadratic curve is defined by equation $Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$?



Regression



Train Data
Test Data
Prediction
X



Train Data
Test Data
Prediction
X
y

Formal setting

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n},$$

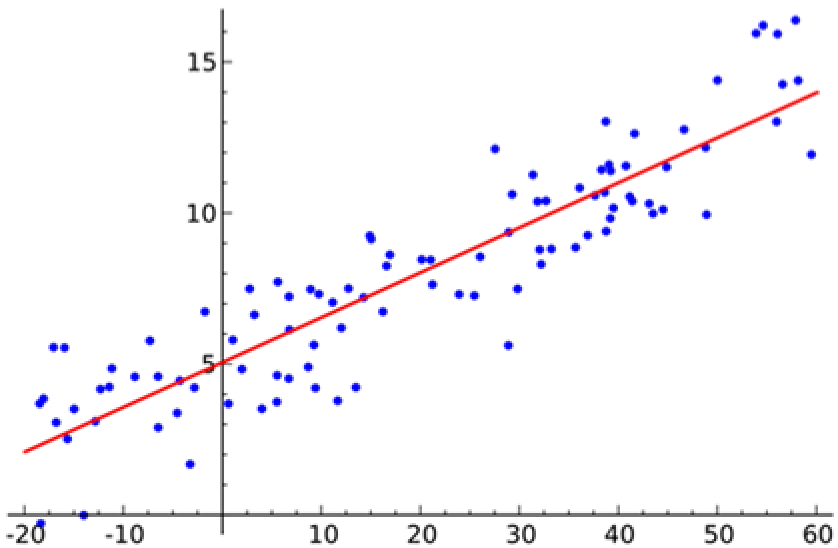
Each row is a data point, consisting of n features

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^{m}.$$

Each value is a label of a data point in X

We want to construct T: $\mathbb{R}^{m \times n} \rightarrow \mathbb{R}^m$ such that T is taken from a **simple enough** class of functions and $T(X)$ approximates y **good enough**

Linear Regression



Regression task:

Find w, such that

$$\frac{1}{m}\|Xw - y\| \to \min.$$

Norm $\|\cdot\|$ can be any norm, the most popular one is MSE (L_2norm):

$$L(w) = \frac{1}{m}\sum_{i=1}^{m}(X_i w - y_i)^2.$$

For MSE exist exact (closed form) solution of this optimization problem: $w = (X^T X)^{-1} X^T y$.

Linear regression

Example:

$X_1 = (1, 1), y_1 = 5$
$X_2 = (1, 0), y_2 = 1$
$X_3 = (0, 1), y_3 = 1$

Let:
$w = (w_1, w_2)$

$$L(w) = \frac{1}{3}\|Xw - y\|_2^2 = \frac{1}{3}\sum (X_i w - y_i)^2 =$$

$$= (w_1 + w_2 - 5)^2 + (w_2 - 1)^2 + (w_1 - 1)^2$$

What is an optimal w?

Linear regression

Let's transform solution:

$$\nabla_{\mathrm{w}} L(w) = 0 \Leftrightarrow$$
$$X^T(Xw - y) = 0 \Leftrightarrow$$
$$X^T X w = X^T y \Leftrightarrow$$
$$w = (X^T X)^{-1} X^T y$$

Since

$$\nabla_{\mathrm{w}} \frac{1}{m} \sum_{i=1}^{m} (X_i w - y_i)^2 = \frac{1}{m} \sum_{i=1}^{n} X^T \cdot 2(X_i w - y_i)$$

Linear regression

$$\frac{\partial}{\partial w_1}((w_1 + w_2 - 5)^2 + (w_2 - 1)^2 + (w_1 - 1)^2)$$
$$= 4w_1 + 2w_2 - 12 = 0$$

$$\frac{\partial}{\partial w_2}((w_1 + w_2 - 5)^2 + (w_2 - 1)^2 + (w_1 - 1)^2$$
$$= 2w_1 + 4w_2 - 12 = 0$$

We can derive that $(w_1, w_2) = (2, 2)$ is a solution.

Linear regression

But if we want to add an **intercept (bias term)** term and minimize $\|Xw + w_0 \; - y\|$?

Example transforms:

$$X_1 = (1, 1, 1), y_1 = 5$$
$$X_2 = (1, 1, 0), y_1 = 1$$
$$X_3 = (1, 0, 1), y_1 = 1$$

Constant feature

Original features

Add w_0 to the feature vector:

$$w = (w_0, w_1, w_2)$$

Linear regression

What if we want to predict label $y = 5x^2 - 2x + 4$?

$$X_1 = (1), y_1 = 7$$
$$X_2 = (2), y_2 = 20$$
$$X_3 = (3), y_3 = 43$$

…

If we just train linear regression, we can not obtain this dependency.

Adding features

But we can train linear regression with additional features – **polynomial** regression.

Original feature

Feature x^2

Constant feature

$$X_1 = (1, 1, 1), y_1 = 7$$
$$X_2 = (2, 4, 1), y_2 = 20$$
$$X_3 = (3, 9, 1), y_3 = 43$$

It’s tempting to think

Evaluating Model Performance

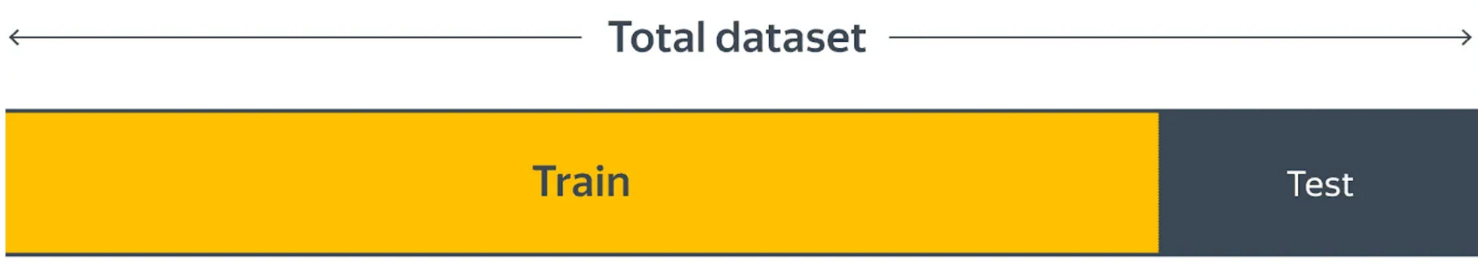
Use unseen samples to evaluate model performance.

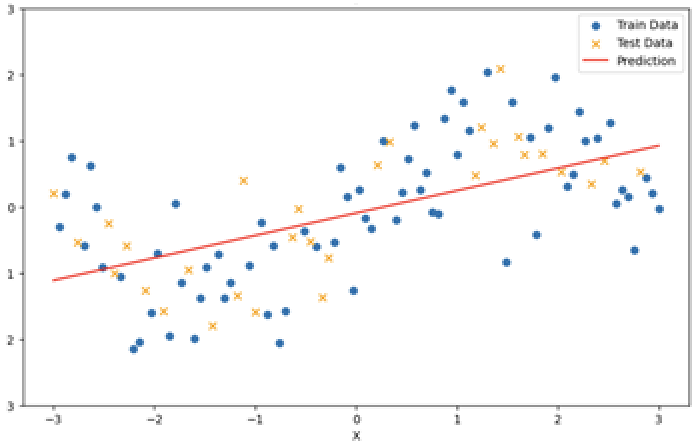
In linear regression, compute the optimal $\boldsymbol{w}$ using only a subset of the data.

Approach:

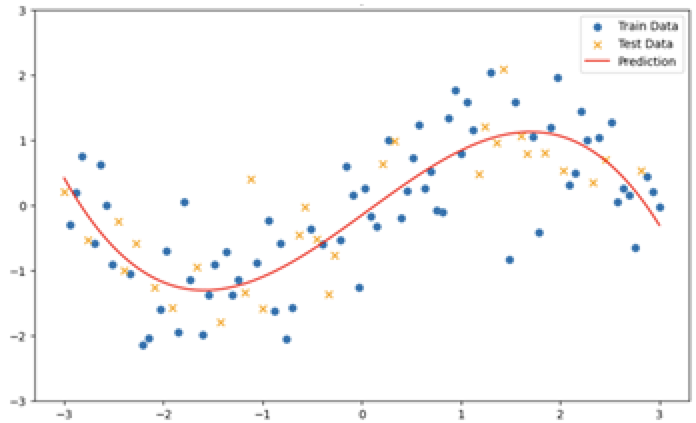
Split the dataset (X, y) into:

- **Training set:** $(\boldsymbol{X_{train}}, \boldsymbol{y_{train}})$— used to learn the model
- **Test set**: $(\boldsymbol{X_{test}}, \boldsymbol{y_{test}})$— used to evaluate performance
- Sometimes, a separate **validation set** $(\boldsymbol{X_{val}}, \boldsymbol{y_{val}})$ is used for tuning hyperparameters.

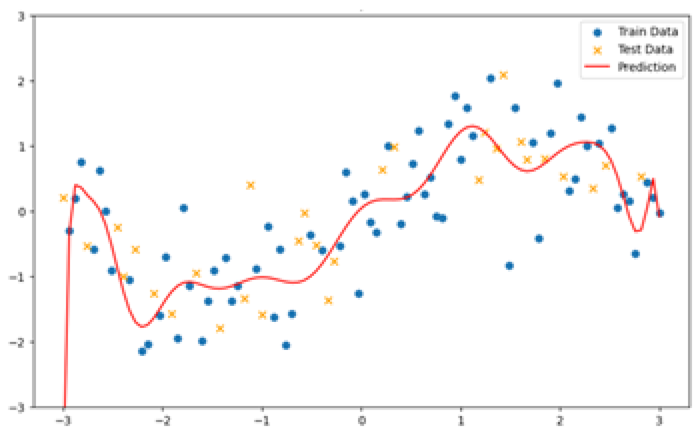




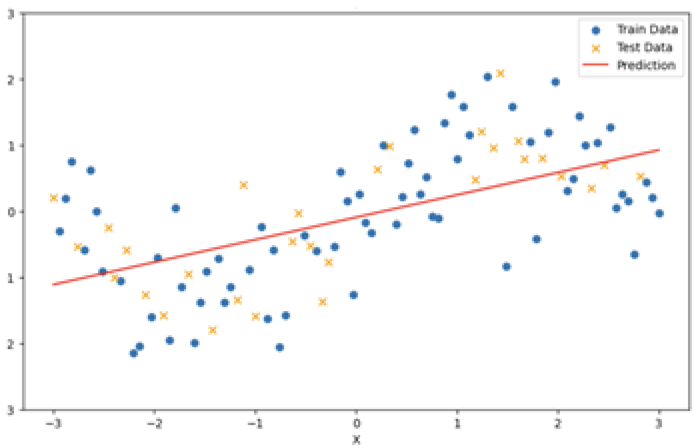
Regression on original dataset



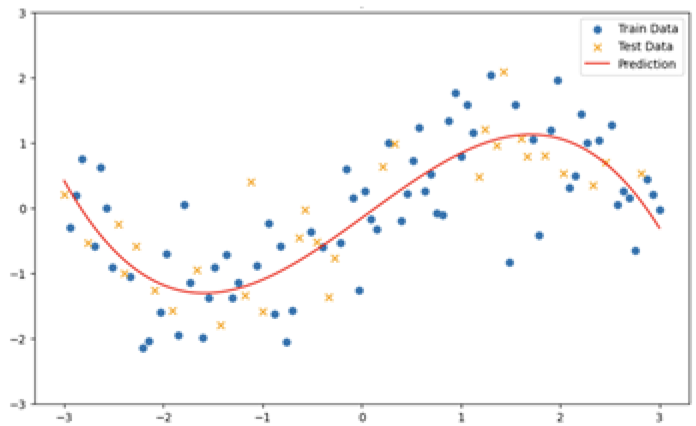
Added powers ≤3



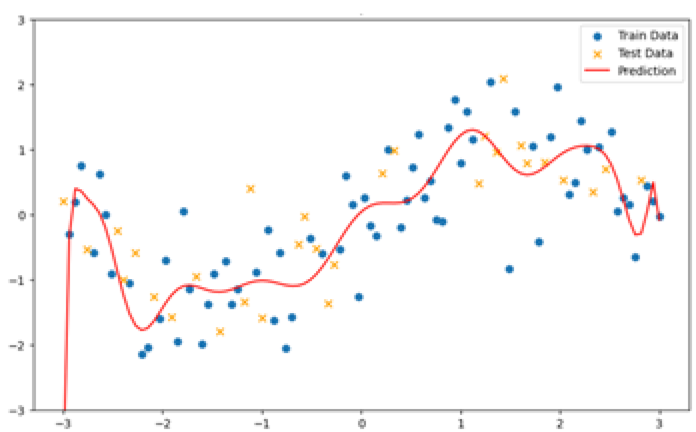
Added powers ≤18



Underfitted

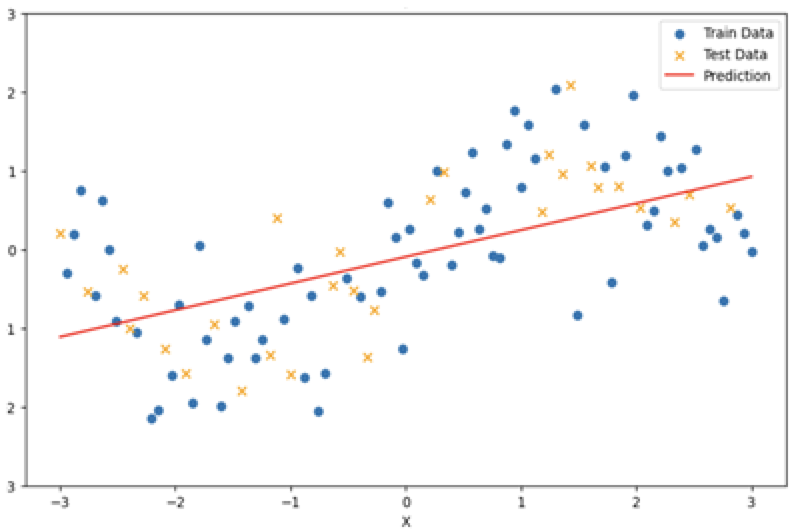


Properly fitted



Overfitted

Underfitted

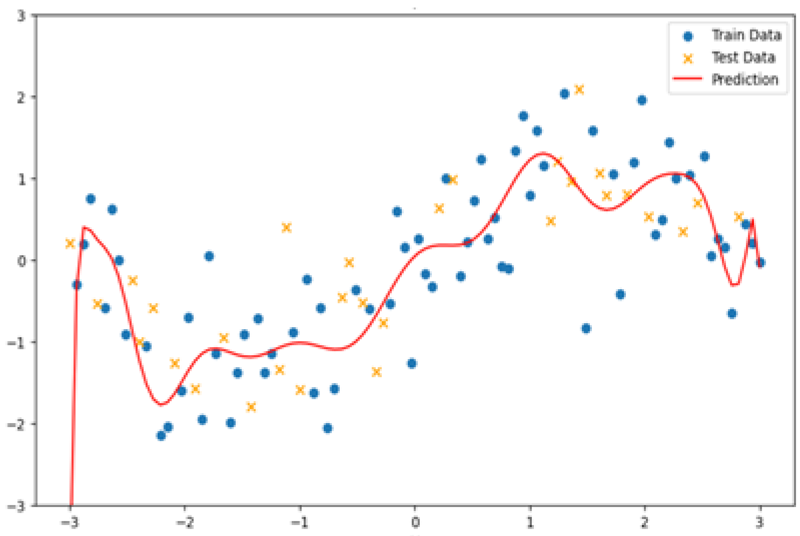


Main indicator:

Doesn’t capture the **pattern**
Huge error on training dataset

Solutions:
- Increase model complexity
- Add more features
- Reduce regularization*
- Improve optimization technique

Overfitted



Main indicator:

Learns **noise patterns.**
Difference in error between train and test datasets.

Solutions:

- Reduce model complexity
- Use regularization*
- Increase training data
- Select better features
- Improve optimization technique

Multicollinearity

Suppose $X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$ and $y = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

What w minimizes $\|Xw - y\|$?

Multicollinearity

Suppose $X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$ and $y = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

What w minimizes $\|Xw - y\|$?

$$\boldsymbol{w = (2, x)\ \forall x \in \mathbb{R}.}$$

$$X^T X = \begin{pmatrix} 3 & 0 \\ 0 & 0 \end{pmatrix}.\ w = \color{red}{\boldsymbol{(X^T X)^{-1}}} X^T y$$

Multicollinearity

Even if $X = \begin{pmatrix} 1 & \varepsilon \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$ and $y = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

We have a problem: $X^T X = \begin{pmatrix} 3 & \varepsilon \\ \varepsilon & \varepsilon^2 \end{pmatrix}$ is close to degenerate.

$$(X^T X)^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -\frac{1}{\varepsilon} \\ -\frac{1}{\varepsilon} & \frac{3}{\varepsilon^2} \end{pmatrix}$$

$w = (X^T X)^{-1} X^T y$ can be *sensitive* to y.

Multicollinearity

How to fix an almost degenerate matrix:

$$w = (X^T X + \lambda I)^{-1} X^T y$$

We can prove that this corresponds to the following task:

$$||Xw - y|| \ + \lambda ||w||_2^2 \to \min.$$

where $||w||_2^2 = w_1^2 + \cdots + w_n^2$

Regularization

Instead of minimizing $\|Xw - y\|$ let's minimize

$\mathcal{L}(w) = \frac{1}{m}\|Xw - y\| + f(w)$,

commonly we use $f(w) = \lambda\|w\|_p^p$.

$f(w) \; = \; \lambda\|w\|_2^2$ – **Ridge (L2)** regularization.

Ridge solution: $w \; = (X^\top X + \lambda I)^{-1} X^\top y$

$f(w) \; = \; \lambda\|w\|_1$– **Lasso (L1)** regularization.

Lasso doesn't have a closed form solution.

Compare: L1 vs L2