Primer on Machine Learning in Finance

Ajay Verma

January 2026

Warm-up Problem 1

A local grocery chain is building a new central distribution warehouse along a straight highway that connects three of its busiest stores.

- Store A is located at mile marker 10.
- Store B is located at mile marker 20.
- Store C is located at mile marker 30.

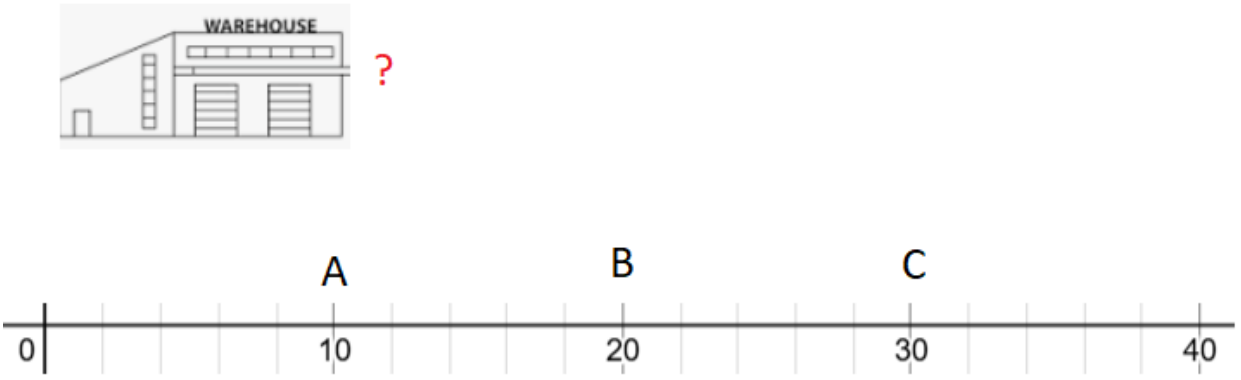


Figure 1: Where to build the warehouse?

The cost of running the warehouse is unique. Due to fuel consumption and vehicle wear-and-tear, the delivery cost to each store is proportional to the *square* of the distance between the warehouse and that store.

Goal: Find the exact mile marker location x of the warehouse that results in the lowest total delivery cost to all three stores combined.

Your Tasks:

1. Find the minimum value of the function
$$f(x) = (x + 3)^2 - 5.$$
2. Write an expression for the total delivery cost $C(x)$ as a function of the warehouse location x.
3. Determine the value of x that minimizes the total cost.

Warm-up Problem 2

An investor wants their portfolio to have a steady annual return of 15%. They are currently invested in three different funds that are performing differently due to market conditions:

- Fund A is returning 4%.
- Fund B is returning 6%.
- Fund C is returning 14%.

Note: In this problem, the idea of *investment stress* is intentionally *over-simplified* for learning purposes. Real investors think about risk using more sophisticated tools, but here we focus on a simple mathematical proxy that captures the intuition that *large deviations from a target feel much worse than small ones.*

The investor's *Stress Level* depends on how far each fund's return is from a chosen target return x. We will loosely refer to this deviation as a *tracking error*. Because large misses are much more stressful than small ones, the stress from each fund is proportional to the *square* of the difference between its return and the target return.

Goal: Find the target return x that minimizes the total stress across all three funds.

Your Tasks:

1. Write an expression for the total stress $C(x)$ as a function of the target return x.
2. Determine the value of x that results in the lowest total stress.

When people talk about Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL), it might sound like they are all the same thing and are frequently used in the same context or even as synonyms. We first need to define clearly what we are talking about when we mention all these terms. How do artificial intelligence, machine learning, and deep learning relate to each other or even are different from each other?

At the highest level, AI refers to the efforts to automate tasks that normally require human intelligence - things like making decisions, recognizing objects, learning or solving problems. Not all AI systems truly learn; some simply follow long lists of rules. Imagine writing a chess program that checks every possible board position and then follows a giant set of "if this happens, then do that" instructions. That program does not actually learn; it just follows commands or rules humans came up with. This is still a form of AI, but limited by humans figuring out explicit rules for solving complex problems. For complex problems, such as image classification or speech recognition, it was not possible for humans to come up with all the possible rules and hence a new approach to solving such problems arose : Machine Learning.

Machine Learning (ML) often categorized as a subfield of AI, goes a step further by allowing computers to improve by learning from data instead of being told every rule. In ML, humans feed the computer examples - such as data points and the correct answers for some of them - and then the computer figures out patterns on its own. It basically comes up with a model which is simply a function of input variables (X) to arrive at possible output values (y) based on values of X. The process of identifying this functional form that best matches y is called 'training'. The process of testing this function on unseen data or values of X is called "validation".

A program is considered to be "learning" if it gets better at a task as it gains more experience. Think of it like practicing math problems: the more examples you work through, the better you become. ML works the same way - computers practice on data, measure their performance, and improve over time. More formally, if a program is working on some task T and we evaluate how well it performs using a performance measure P, then the program is considered to be "learning" if its score on P for the task T increases over time. In other words, as the program receives more experience, its performance on the task gets better according to the measure P.

While ML is closely related to mathematics and statistics, it does not rely as heavily on formal proofs. Instead, it is engineering-oriented and hands-on discipline in which ideas are most often proven empirically (by experiments on real data) rather than proven theoretically.

Deep Learning (DL) is a more advanced form of Machine Learning. Instead of learning from data in one step, DL learns through many layers of processing. Early layers might detect simple patterns (such as lines or edges), while later layers detect more complex features (such as shapes, faces, or even emotions). This approach is inspired by how neurons in the human brain work together. Deep Learning is behind many of the technologies you see today - image recognition in phones, self-driving cars, voice assistants, and scientific tools that analyze large amounts of data. It is essentially ML with more depth, more structure, and the ability to understand very complex patterns.

1. Every deep learning model is machine learning, and every machine learning model is a kind of artificial intelligence — but not every AI system learns, and not every ML model is deep. Draw a venn diagram depicting how AI, ML and DL are related

2. A bank wants to predict a customer's monthly spending using their income and age. Explain in words what the function

$$g(\text{income}, \text{age})$$

represents in this situation. Is $g(\cdot)$ something we choose, or something we learn from data?

3. For each example below, decide whether it is:

   - artificial intelligence (AI) but not machine learning (ML),
   - machine learning (ML) but not deep learning (DL),
   - deep learning (DL).

   (a) A rule-based chess program using hard-coded "if...then" rules.

   (b) A model predicting house prices using house characteristics (size, number of bedrooms, age, distance from school, etc).

   (c) An image recognition system using a large neural network to identify animals.

4. **Patterns and Learning**

   A model is given the following input–output pairs:

$$(1,\ 2), \quad (2,\ 4), \quad (3,\ 6), \quad (4,\ 8).$$

   (a) Find a mathematical rule that maps each input x to its output y.

(b) Using your rule, predict the output when $x = 10$.

(c) Explain how machines can discover such rules. Hint: "learning from data"

5. **Error Measurement (Performance Measure P)**

A learning model predicts the following outputs:

$$\hat{y} = \{3,\ 5,\ 9,\ 12\}.$$

The true outputs are:

$$y = \{2,\ 6,\ 10,\ 11\}.$$

Compute the mean squared error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2.$$

Is a lower MSE better or worse for performance? Explain why.

6. **Linear Regression (Basic ML Model)**

Suppose a model assumes a linear relationship:

$$y = mx + b.$$

Given the data points:

$$(1,\ 3), \quad (2,\ 5), \quad (3,\ 7),$$

Find the values of m and b that best fits the data. Hint: Check how much y changes when x increases by 1.

7. **Layers in Deep Learning**

A simple neural network has the following layer transformations:

$$h_1 = 2x + 1, \qquad h_2 = 3h_1 - 5, \qquad \hat{y} = h_2 + 4.$$

(a) Write $\hat{y}$ directly as a function of x.

8. **Classification Example**

A model must classify inputs into two categories: "cat" (0) or "dog" (1).

Given the model's outputs:

$$\hat{y} = \{0.2,\ 0.8,\ 0.6,\ 0.1\},$$

and the true labels/values:

$$y = \{0,\ 1,\ 1,\ 0\},$$

(a) Convert the predictions into final decisions using the rule:

$$\hat{y}_i > 0.5 \Rightarrow 1, \quad \text{else } 0.$$

(b) Count how many predictions are correct.

9. A program is trained to recognize handwritten digits. On its first test, it correctly identifies 60 out of 100 digits. After being shown more labeled examples, it correctly identifies 85 out of 100 digits. Is the program learning with respect to task T = digit recognition and performance measure P = accuracy? Explain why.

10. A robot tries to navigate a maze. Each attempt is scored from 0 to 100 based on how efficiently it reaches the exit. On three attempts, its scores are:

$$P_1 = 40, \qquad P_2 = 55, \qquad P_3 = 70.$$

Does this indicate that the robot is learning from experience? What evidence suggests improvement?

11. A prediction model estimates house prices. After training on 200 examples, its mean squared error (MSE) is 12,000. After training on 1,000 examples, its MSE drops to 4,500. If the task is T = price prediction and the performance measure is P = MSE , explain why the model is or is not learning from experience.

Machine Learning can be divided into two major categories: **supervised learning** and **unsupervised learning**. These two approaches differ in whether or not the data comes with "labels," meaning the actual correct values/output/outcomes.

Supervised Learning (With Labels)

In supervised learning, we give the computer both:

- the **features** (the inputs), and
- the **labels** (the correct outputs).

The goal is for the model to learn a relationship between the inputs and the outputs. Once the model has learned this pattern, it can be used to predict output for new data where the correct answer is unknown.

Supervised learning includes:

- **Classification**: predicting categories (e.g., cat vs. dog, spam vs. not spam)
- **Regression**: predicting numerical values (e.g., house prices, test scores)

Unsupervised Learning (Without Labels)

In unsupervised learning, the data has **no labels**. The goal is to discover hidden structure or patterns directly from the data. This is often described as "letting the dataset speak for itself," because the model must find relationships without being told the correct answers.

Common unsupervised learning tasks include:

- **Clustering**: grouping similar data points together
- **Dimensionality Reduction**: simplifying complex data while keeping important information

12. **Identifying Supervised vs. Unsupervised Learning**

    You are given the following sets of data:

    (a) A table of students with their hours studied and their exam scores.

    (b) A list of 500 points in the plane with coordinates (x, y) but no additional information.

    (c) A dataset of emails marked as "spam" (1) or "not spam" (0).

    (d) Grouping of customers based on their spending habits.

    (e) Tomorrow's stock return forecast using historical price data.

    For each of the above, state whether a machine learning model would use **supervised** or **unsupervised** learning, and explain your reasoning.

13. **Unsupervised Clustering**

Consider the following set of points in the coordinate plane:

$$(1, 2),\ (2, 1),\ (1, 1),\ (8, 9),\ (9, 8),\ (8, 8).$$

(a) Sketch these points on a coordinate grid.

(b) Based on visual inspection, group the points into two clusters.

Explain why this is an unsupervised learning task.

14. **Predicting Stock Returns**

A dataset contains the daily percentage change of a stock's price (y) and the corresponding change in the overall market index (x):

$$(0.5\%, 0.7\%), \quad (1.0\%, 1.4\%), \quad (1.5\%, 2.1\%).$$

(a) Find the linear relationship between the stock return (output) and the market return (input).

(b) Use your model to predict the stock's return when the market return is 2.5%.

(c) Explain why this is an example of supervised learning.

15. **Loan Default Prediction**

A bank has the following simplified customer data:

Income ($k)	Credit Score	Will Default (1=yes, 0=no)
50	700	0
35	600	1
80	750	0

(a) Identify the inputs (features) and the output (label).

(b) Describe how the bank could use a trained model to evaluate a new customer's risk.

16. **Fraud Detection**

A credit card company monitors a customer's daily spending amounts to detect unusual behavior. Here are the amounts (in dollars) spent over the last six days:

$$40, \quad 35, \quad 50, \quad 42, \quad 39, \quad 600.$$

Important idea: In fraud detection, the computer is not told which transactions are "fraud" or "not fraud." Instead, it looks for values that are *very different* from the customer's usual spending pattern.

(a) Based on the numbers above, identify which amount is most likely an **anomaly** (an amount that does not fit the pattern of the others).

(b) Explain why this situation is an example of **unsupervised learning**. (Hint: Are we given labels such as "fraud" or "not fraud"? Or do we have to find patterns ourselves?)

17. **Bond Price Prediction**

A **bond** is a type of loan, but instead of borrowing money from a bank, a company or government borrows money from investors. When you buy a bond, you are essentially lending money to the issuer.

A bond always includes:

- **Face Value (Principal):** the amount that will be paid back at maturity.
- **Maturity:** the number of years until the bond must be fully repaid.
- **Interest Payments (Coupons):** some bonds pay regular interest; others do not.

The price of a bond often depends on how many years remain until it matures. Suppose we want to build a simple supervised learning model that predicts a bond's price based only on its *time to maturity*.

You are given the following data (in years and dollars):

$$(1,\ 980), \qquad (3,\ 940), \qquad (5,\ 900).$$

Here:

$$x = \text{time to maturity (years)}, \qquad y = \text{bond price (dollars)}.$$

A supervised learning model assumes that the relationship between x and y is approximately linear:

$$y = mx + b.$$

(a) Use intuition to determine whether the m should be positive or negative. (Hint: Do bond prices generally go up or down as maturity increases?)

(b) Find a linear rule that fits the data points above. (Your goal is to estimate m and b by noticing the pattern in how the prices are decreasing.)

(c) Using your linear rule, predict the price of a bond with a maturity of 4 years.

(d) Explain why this is an example of **supervised learning**. (Hint: Are you given "labels" - the actual bond prices?)

Understanding Linear Regression

Linear regression is one of the simplest and most important models in machine learning. It assumes that the output we want to predict (the label) can be written as a **linear combination** of the input features or variables.

The model is written as:

$$\hat{y} = g(X) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n,$$

where:

- $\hat{y}$ is the predicted value.
- n is the number of input features/variables.
- x_j is the j-th input feature/variable.
- θ_k is the k-th model parameter/coefficient.
- The parameter θ_0 is called the **bias term** or **intercept**. It allows the regression line to shift up or down so that it can match the average level of the data. Without this term, every regression line must pass through the origin, which is usually unrealistic. Intuitively, θ_0 plays the same role as the "starting value" in the equation of a line:

  $$y = mx + b,$$

  where b is the vertical intercept. It allows the model to fit data that is not centered around zero.

Each parameter θ_k tells us how strongly that feature contributes to the prediction. The model tries to find the best values of θ so that the predictions match the real data as closely as possible.

To make the model learn, we adjust the values of parameters θ to **minimize the mean squared error (MSE)** between the predictions and the true labels in the training data.

The mean squared error (MSE) objective (also called loss or cost) function is defined as:

$$J(\theta) \;=\; \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - x^{(i)}\theta\right)^2,$$

and the training problem is to solve

$$\min_{\theta}\; J(\theta).$$

where:

- m is the number of training examples,
- $y^{(i)}$ is the actual value for the i-th example,
- $x^{(i)}$ is the row of inputs for the i-th example,
- $x^{(i)}\theta$ is the predicted value for that example.

Intuitively, the model learns by trying different values of the parameters θ and choosing the ones that make the squared differences between the predicted values and the true values as small as possible. Refer to this link for an interactive visualization:

https://claude.ai/public/artifacts/4365666e-216b-4cf1-81e8-4720758afa6d

Problem 1: Finding a Linear Rule

A stock's price (in dollars) depends on the number of days since the beginning of the year. You observe:

$$(1,\ 102), \qquad (2,\ 104), \qquad (3,\ 106).$$

1. Guess a linear rule of the form $\hat{y} = \theta_0 + \theta_1 x$ that fits this pattern.
2. Use your rule to predict the stock's price on day 5.

Problem 2: Linear Regression

A linear model predicts the price of a bond using the rule

$$\hat{y} = 500 + 20x,$$

where x is the number of years to maturity.

Given the following data:

$$(5,\ 620), \qquad (10,\ 690),$$

1. Compute the predicted value $\hat{y}$ for each data point.

2. Compute the squared error $(y - \hat{y})^2$ for each example.

3. Using your results above, compute the mean squared error (MSE).

4. The model parameters are the intercept and slope. Let a general model be

$$\hat{y} = \theta_0 + \theta_1 x$$

We can write the mean squared error (MSE) as a function of θ_0 and θ_1:

$$J(\theta_0, \theta_1) = \frac{1}{2}\left[(620 - (\theta_0 + 5\theta_1))^2 + (690 - (\theta_0 + 10\theta_1))^2\right].$$

(a) By trying reasonable values of θ_0 and θ_1, find parameters that reduce the MSE compared to the original model. Can you minimize the MSE?

(b) Write down your new regression equation using the improved values of θ_0 and θ_1.

Problem 3: Identify Best Fit/Prediction Line

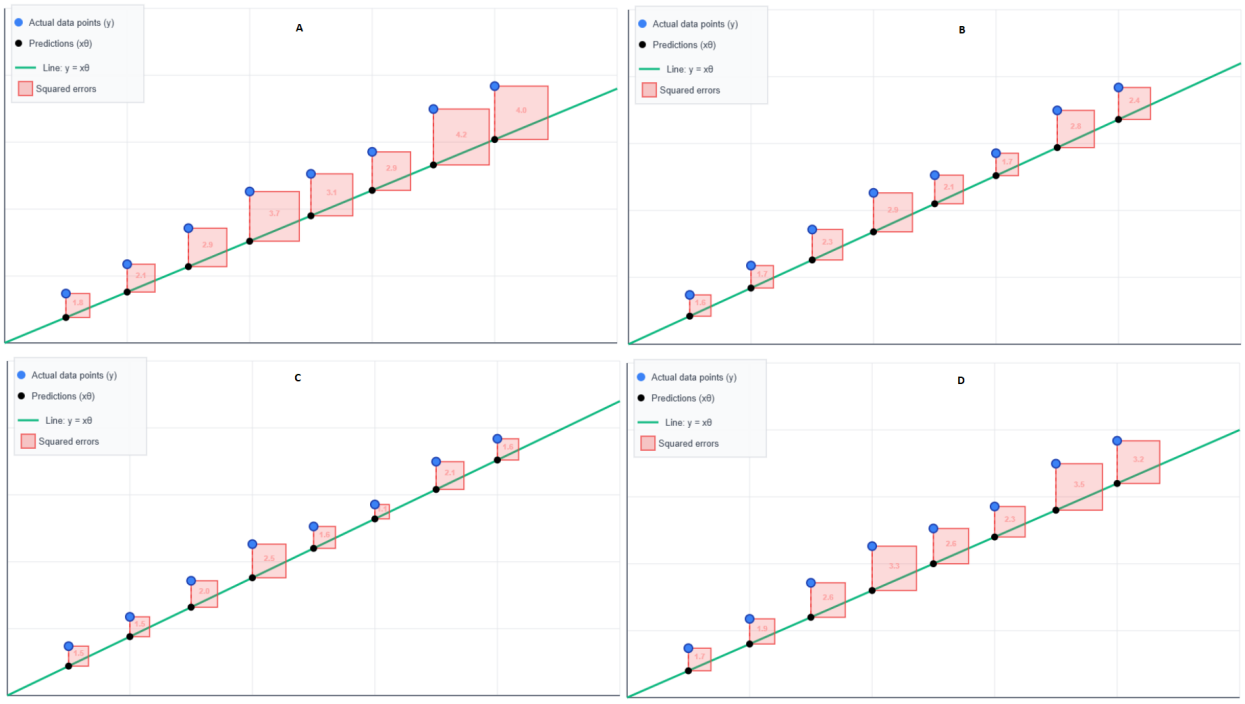


Figure 2: Which figure denotes minimum MSE? Which figure denotes best fit/prediction?

Understanding R^2

The coefficient of determination, written as R^2, measures how well a linear regression model fits the data. It tells us how much of the variation in the labels (the y values) can be explained by the model.

The R^2 value is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{m} \left(y^{(i)} - x^{(i)}\hat{\theta}\right)^2}{\sum_{i=1}^{m} \left(y^{(i)} - \bar{y}\right)^2},$$

where:

- m is the number of training examples,
- $y^{(i)}$ is the actual value,
- $x^{(i)}\hat{\theta}$ is the model's prediction,
- $\bar{y}$ is the average value in the training data.

The denominator measures how much the actual value vary around the average. The numerator measures how much error the model makes.

The R^2 tells us the *proportion of variation in data explained by the model*. For example:

$$R^2 = 0.80 \quad \text{means the model explains 80\% of the variation in the data.}$$

Another way to think about R^2 = 1 - (Model's Error) / (Baseline Error):

- Compare the model to a simple baseline model that predicts the **average** value for every example.
- If the regression model is better than this average-prediction model, R^2 will be positive.
- If the regression model is worse, R^2 may be negative.

Problem: Computing R^2 for a Simple Dataset

A model predicts the following values:

$$\hat{y} = \{7,\ 9,\ 11,\ 13\}$$

and the true labels are:

$$y = \{6,\ 10,\ 14,\ 12\}.$$

1. Compute the squared errors $(y - \hat{y})^2$ for each point.
2. Compute the average label $\bar{y}$.
3. Compute the denominator $\sum(y - \bar{y})^2$.
4. Use the formula to compute R^2.
5. Interpret whether the model explains a lot of the variation.

Problem: Comparing the Model to the Average Predictor

A simple dataset has labels:

$$y = \{8,\ 12,\ 10,\ 14\}.$$

Two models are tested:

- **Model A:** always predicts the average label,
- **Model B:** predicts $\hat{y} = \{7,\ 11,\ 10,\ 12\}$.

1. Compute the average label.
2. Compute the error of Model A and Model B.
3. Which model has the higher R^2?

Training Data, Test Data, In-Sample and Out-of-Sample Data

In linear regression, our goal is to learn a straight-line rule of the form

$$\hat{y} = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

that predicts the label $\hat{y}$ as accurately as possible. To measure how good the model is, we cannot test it only on the data it was built from - we must also test how well it performs on new, unseen data.

To do this, we divide the dataset into two parts: the **training data** and the **test data**. These correspond to what we call **in-sample** and **out-of-sample** performance.

Training Data (In-Sample Data)

The **training data** is the portion of the dataset used to compute the parameters $\theta_0, \theta_1, \ldots, \theta_n$ of the linear regression model.

$$\text{Training (in-sample)}: \quad \text{The model } \textit{sees} \text{ these examples and adjusts } \theta \text{ to fit them.}$$

Because linear regression chooses θ to minimize the in-sample mean squared error, its performance on the training data is usually strong.

Test Data (Out-of-Sample Data)

The **test data** is kept separate during training. The linear regression model is *not* allowed to see these examples while estimating θ.

$$\text{Test (out-of-sample)}: \quad \text{The model predicts } \hat{y} \text{ for data it has never seen before.}$$

We then compute metrics such as the out-of-sample R^2:

$$R^2_{\text{test}} = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}.$$

Unlike the in-sample case, the out-of-sample R^2 *can be negative*. A negative value means:

Linear regression predicts worse than simply using the average of the labels.

This indicates the model does not generalize well and may be **underfitting** (too simple to capture the pattern).

Overfitting vs. Underfitting in Machine Learning

In machine learning, the goal is to build a model that not only fits the training data well, but also *generalizes* to new, unseen data. Two common problems are **underfitting** and **overfitting**.

A model **underfits** when it is too simple to capture the real pattern in the data. This leads to:

- low accuracy on the training data (in-sample),
- low accuracy on the test data (out-of-sample),
- negative or very low R^2 on both datasets.

Underfitting happens when the model is not powerful enough, such as using a straight line to fit a curved trend.

A model **overfits** when it learns the training data *too well*, including outliers or extreme values. This means:

- very high accuracy on the training data,
- poor accuracy on test data,
- a large difference between training R^2 and test R^2.

An overfitted model performs well on data it has already seen, but generalizes badly to new data.

Problem: Identifying Underfitting vs. Overfitting

A model achieves:

$$R^2_{\text{train}} = 0.95, \qquad R^2_{\text{test}} = 0.10.$$

1. Is this model underfitting or overfitting? Explain why.
2. What does the gap between training and test R^2 tell you?

Problem: When Is a Model Too Simple?

A linear model is used to predict a company's revenue based on advertising spending. The training and test R^2 are:

$$R^2_{\text{train}} = 0.12, \qquad R^2_{\text{test}} = 0.09.$$

1. Is the model underfitting or overfitting? Explain your reasoning.

Problem: Understanding Training vs. Test Data

You are given a dataset of 100 students with two features:

$$\text{hours studied}, \qquad \text{attendance rate}.$$

The goal is to predict a student's final exam score using linear regression.

1. Identify which portion of the data should be used for **training** and which for **testing**.

Problem: In-Sample vs. Out-of-Sample Performance

A linear regression model has:

$$R^2_{\text{train}} = 0.75, \qquad R^2_{\text{test}} = -0.20.$$

1. Explain what the training R^2 of 0.75 means in words.
2. Explain why the test R^2 might be negative.
3. Would you trust the model's predictions on new data? Why or why not?

Problem: In-Sample vs. Out-of-Sample Performance

1. A model fits the past 5 years of house prices in your neighborhood perfectly:

   $$R^2_{\text{train}} = 0.99.$$

   However, when it is used to predict prices next year, the errors are very large and

   $$R^2_{\text{test}} \approx 0.$$

   (a) Does this model have strong *predictive power*? Why or why not?

2. Two models are tested on **new** (test) data:

   $$R^2_{\text{test, Model A}} = 0.70, \qquad R^2_{\text{test, Model B}} = 0.20.$$

   Both have similar $R^2_{\text{train}} \approx 0.90$. Which model has more predictive power out-of-sample? Explain your reasoning.

3. A model achieves:

$$R^2_{\text{train}} = 0.98, \qquad R^2_{\text{test}} = -0.10.$$

   (a) What does $R^2_{\text{train}} = 0.98$ tell you about the in-sample fit?

   (b) What does the negative R^2_{test} tell you about performance on new data?

   (c) Why is it dangerous to judge a model only by its training R^2?

4. For a certain prediction problem, two models give the following results:

$$\text{Linear Model: } R^2_{\text{train}} = 0.10, \quad R^2_{\text{test}} = 0.08,$$

$$\text{Complex Model: } R^2_{\text{train}} = 0.95, \quad R^2_{\text{test}} = 0.05.$$

   (a) Which model is likely underfitting the data? Why?

   (b) Which model is likely overfitting the data? Why?

   (c) Which model would you choose, and what might you try next?

5. A very complex model achieves:

$$R^2_{\text{train}} = 0.99, \qquad R^2_{\text{test}} = 0.10.$$

   A simpler, regularized model achieves:

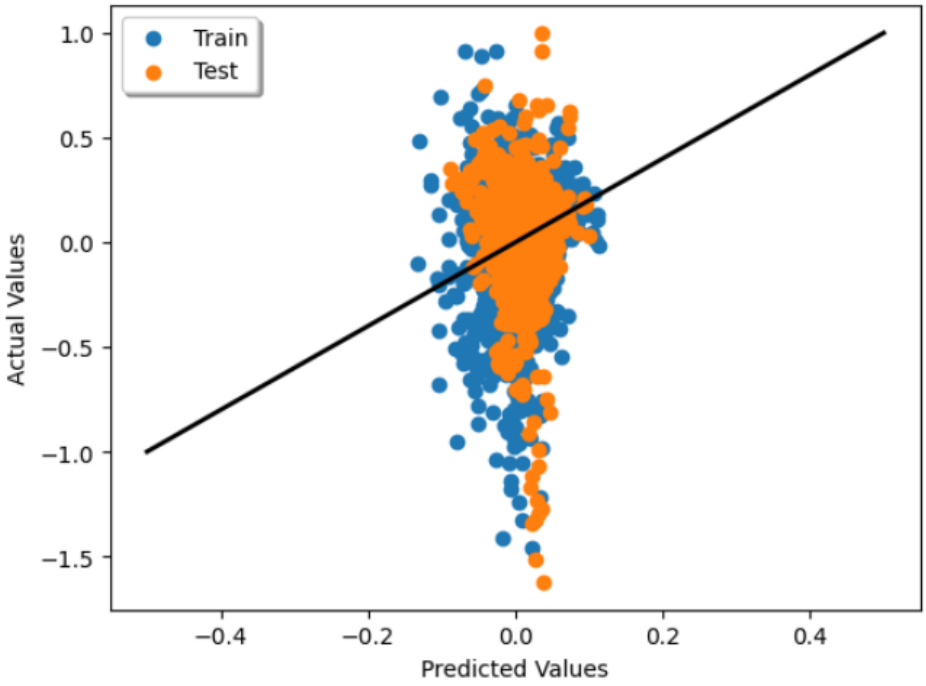
$$R^2_{\text{train}} = 0.90, \qquad R^2_{\text{test}} = 0.25.$$

   (a) Which model fits the training data better?

   (b) Explain why we might prefer the regularized model, even though its training R^2 is lower.

Problem: Predicted vs Actual Plot Interpretation

Below plot shows the relationship between the predicted values (horizontal axis) and the actual values (vertical axis) achieved using linear regression on training and test data. In a model that displays a perfect fit, the coefficient of determination R^2 is equal to 1, and all data points lie on the black line representing the 45° diagonal. In such case, we would have $\hat{y} = y$ for all observations. The model shown here is far from achieving this ideal outcome. a) Is the model overfitting or underfitting? b) Draw a graph showing how overfitting will look like. c) Draw a graph showing how a good fit will look like.



Homework

1. Gradient descent is a widely used optimization algorithm for minimizing a loss function. Watch the YouTube video on *"Mathematics of Gradient Descent"*: https://www.youtube.com/watch?v=jc2IthslyzM
2. Regularization reduces overfitting by constraining the learned parameters θ. Watch the YouTube video on *"Regularization"*: https://www.youtube.com/watch?v=VNrdslReSZU

Useful References

Geron, Aurelien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed., O'Reilly Media, Inc., 2019.

Niven, Ivan. Maxima and Minima Without Calculus. Mathematical Association of America, 1981