Machine Learning: Core Concepts

Welcome to the fascinating world where computers learn from experience, just like you do! Let's discover how machines can make smart predictions.

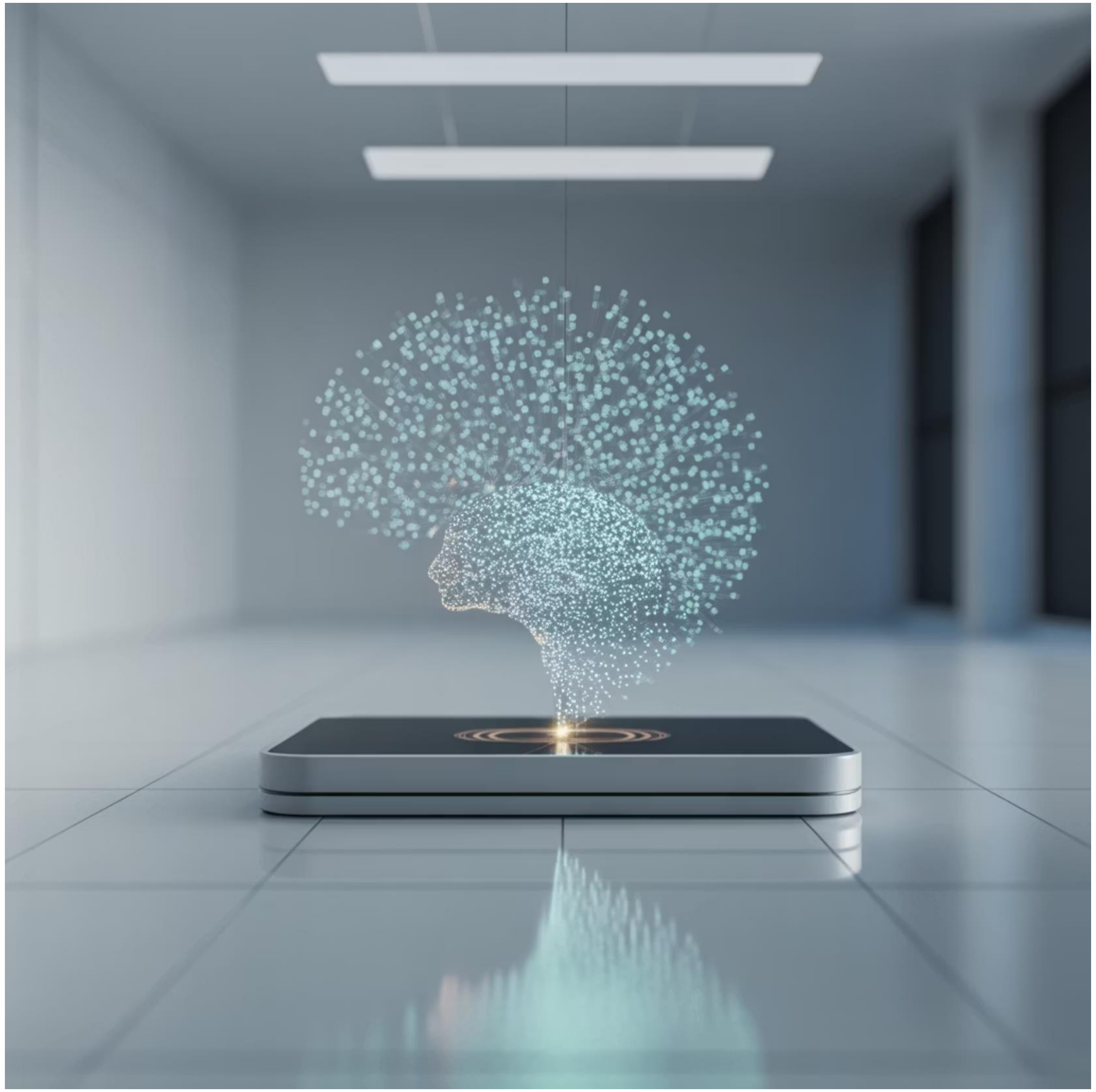
Siva Jasthi, Ph.D.
Python ML (Python for Machine Learning)
www.learnandhelp.com

What is Machine Learning?

Machine Learning (ML) is how computers learn from data instead of being told exactly what to do step-by-step.

It's like teaching a computer by showing examples, so it can make predictions or decisions on its own—similar to how you learn to recognize patterns!

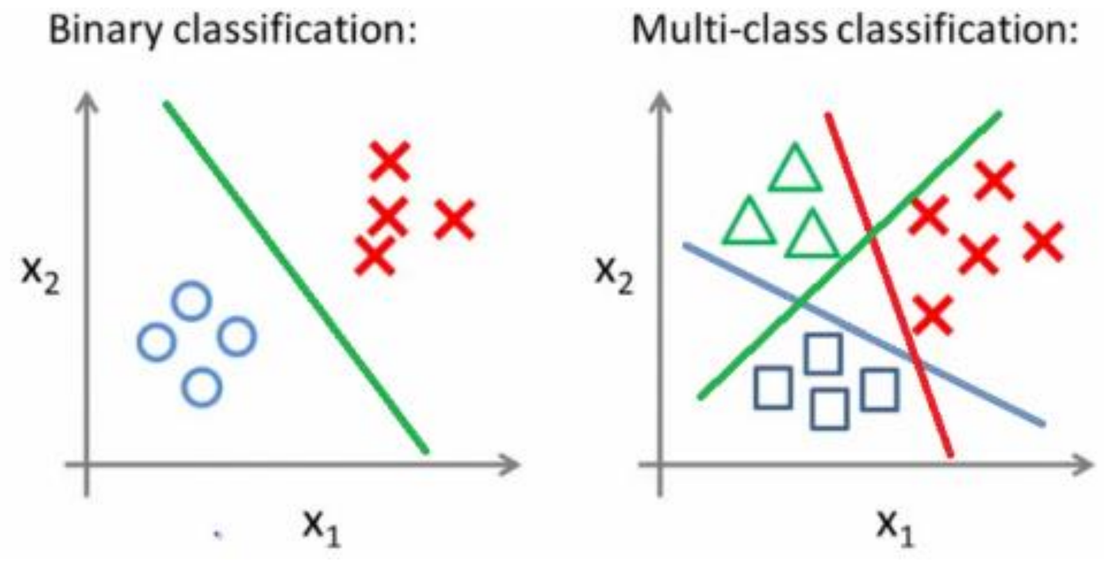
Instead of programming every rule, we let the computer discover patterns from examples.



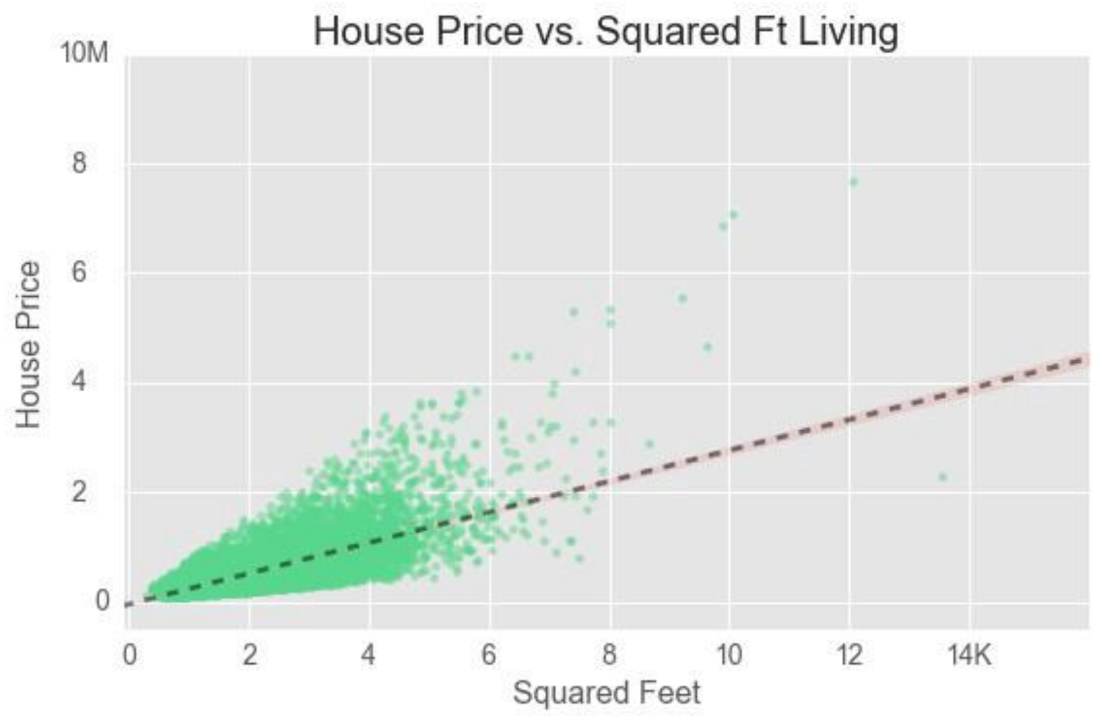
Regression, Classification, and Clustering

https://www.youtube.com/watch?v=rirAaZzjaoA

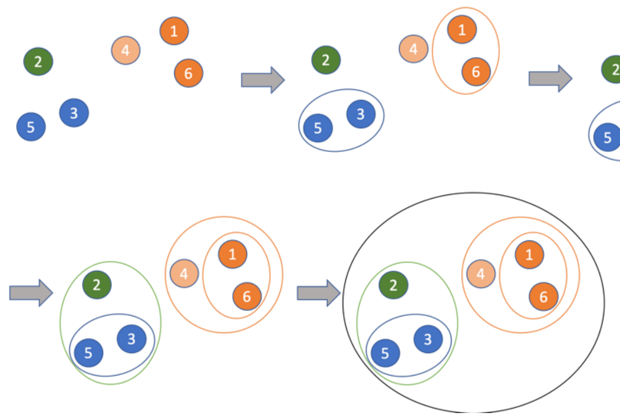
Regression vs Classification vs Regression



Classification



Regression



Clustering

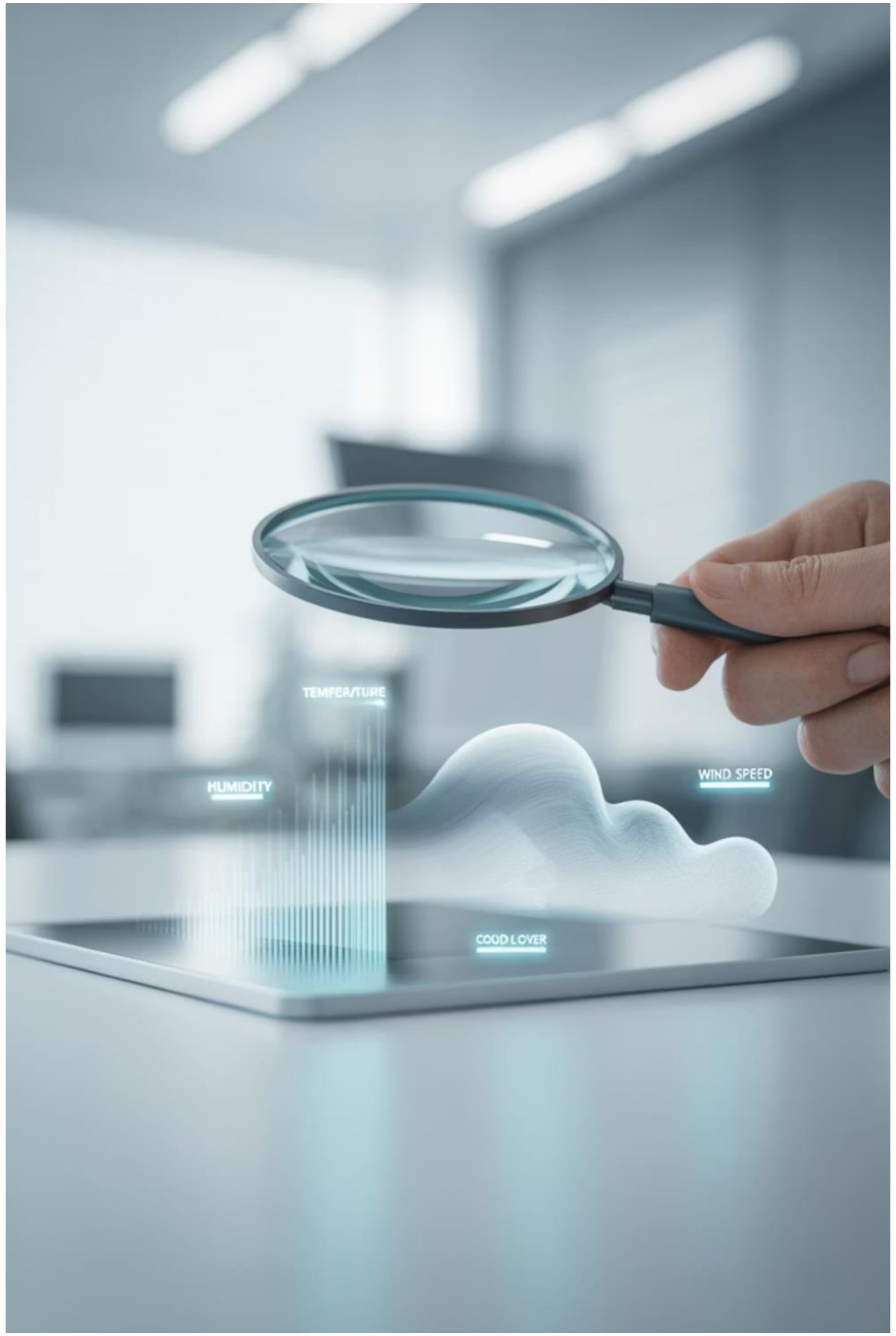
The Goal: Predicting a Target

The Mission

ML models try to predict something called a **target**—the answer we're looking for.

Real Example

Predicting if it will rain tomorrow based on today's weather data and patterns.



What Are Features?

Features are the **clues** or pieces of information the model uses to make predictions. Think of them as ingredients in a recipe!

🌡️ Temperature

How hot or cold it is

💧 Humidity

Amount of moisture in the air

🌬️ Wind Speed

How fast the wind is blowing

☁️ Cloud Cover

How cloudy the sky is

What Is a Target?

The target is the **result or outcome** the model tries to guess—it's what we want to predict!

Every ML model is built to predict its target based on the features it observes.

1

Features (Inputs)

Temperature, humidity, wind

2

Model Processing

Computer analyzes patterns

3

Target (Output)

"Rain" or "No Rain"

Feature Engineering: Picking the Right Clues

Not all features help the model predict accurately. **Feature engineering** means choosing the most useful features—like a detective picking the best clues!

✅ Helpful Feature

Using "cloud cover" to predict rain—clouds are directly related to rainfall!

❌ Unhelpful Feature

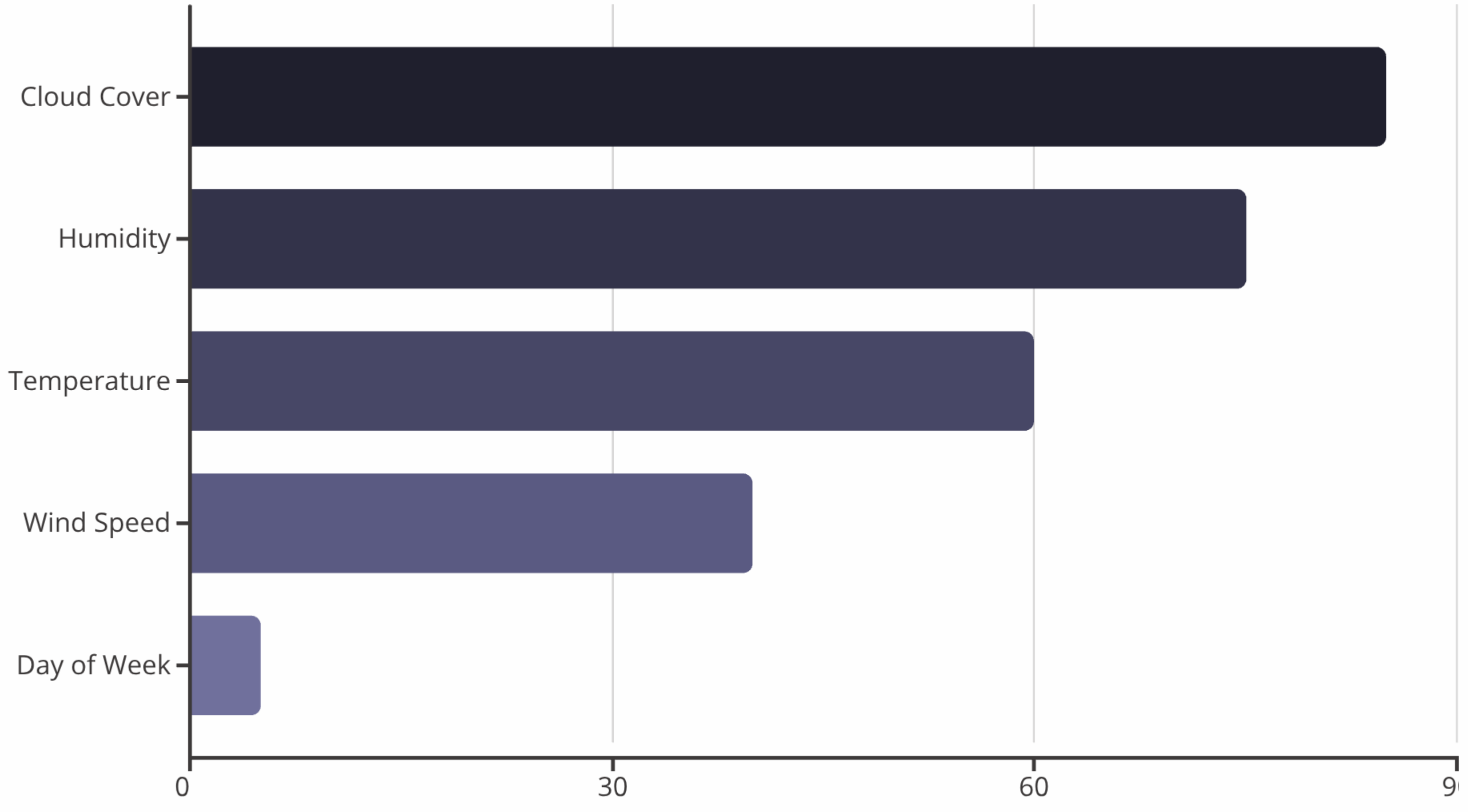
Using "day of the week" to predict rain—Monday doesn't cause rain!

Feature Weights: How Important Are the Clues?

Each feature has a **weight** showing how much it influences the prediction.

Bigger weight = more important feature in making accurate predictions.

The model learns these weights automatically from data, giving more importance to features that help predict the target.

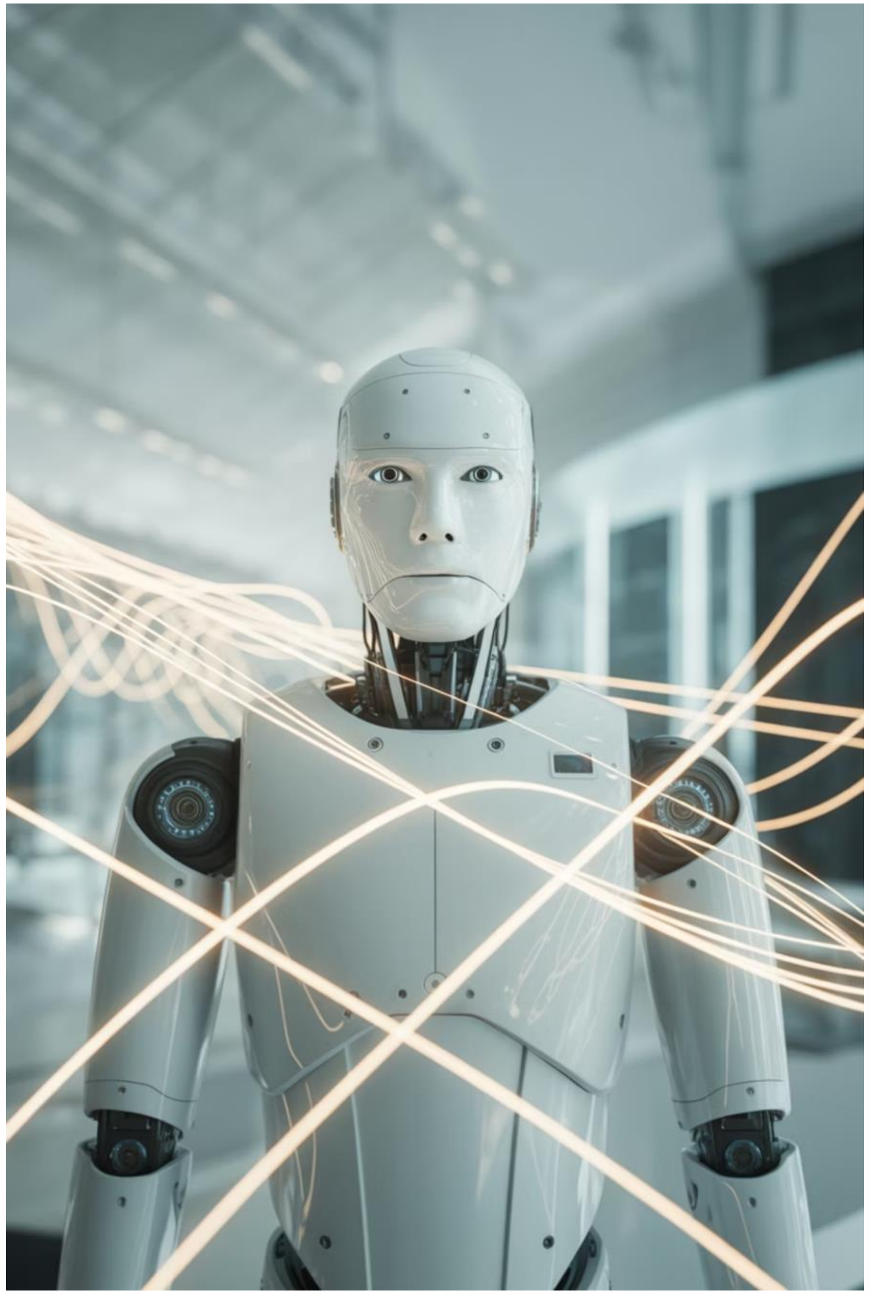


Features and Weights

-Nose , very high weight
-Mouth, very high weight
-Ears, very high weight
-Skull shape, very high weight
-Mouth, very high weight
-Facial hair, high weight
-Eyes, low weight
-Forehead, low weight
-Chest, low weight

Computing the Weights of the Features is a big part of what Machine Learning is!

- https://www.linkedin.com/pulse/introduction-shallow-machine-learning-ayman-mahmoud/



Understanding Variance: When Models Get Too Confused

What is Variance?

Variance means the model's predictions change dramatically with different data sets.

The Problem

High variance means the model learns **noise** (random fluctuations) instead of just true patterns.

The Result

The model makes poor predictions on new data because it memorized quirks instead of learning real patterns.

Understanding Bias: When Models Are Too Simple

Bias means the model is too simple and misses important patterns in the data.

High bias happens when the model makes strong assumptions that prevent it from learning the complexity of real-world data.

Result: The model makes consistent mistakes because it doesn't capture enough detail from the data.

Underfitting: When the Model Is Too Simple

Underfitting happens when the model can't capture the true patterns because it's too basic or simplistic.

01

Weak Example

Trying to predict rain using only the day of the week—this ignores all weather conditions!

02

The Problem

The model is too simple to understand the real relationships in the data.

03

Poor Performance

Predictions are inaccurate on both training data AND new data.

Overfitting: When the Model Is Too Complex

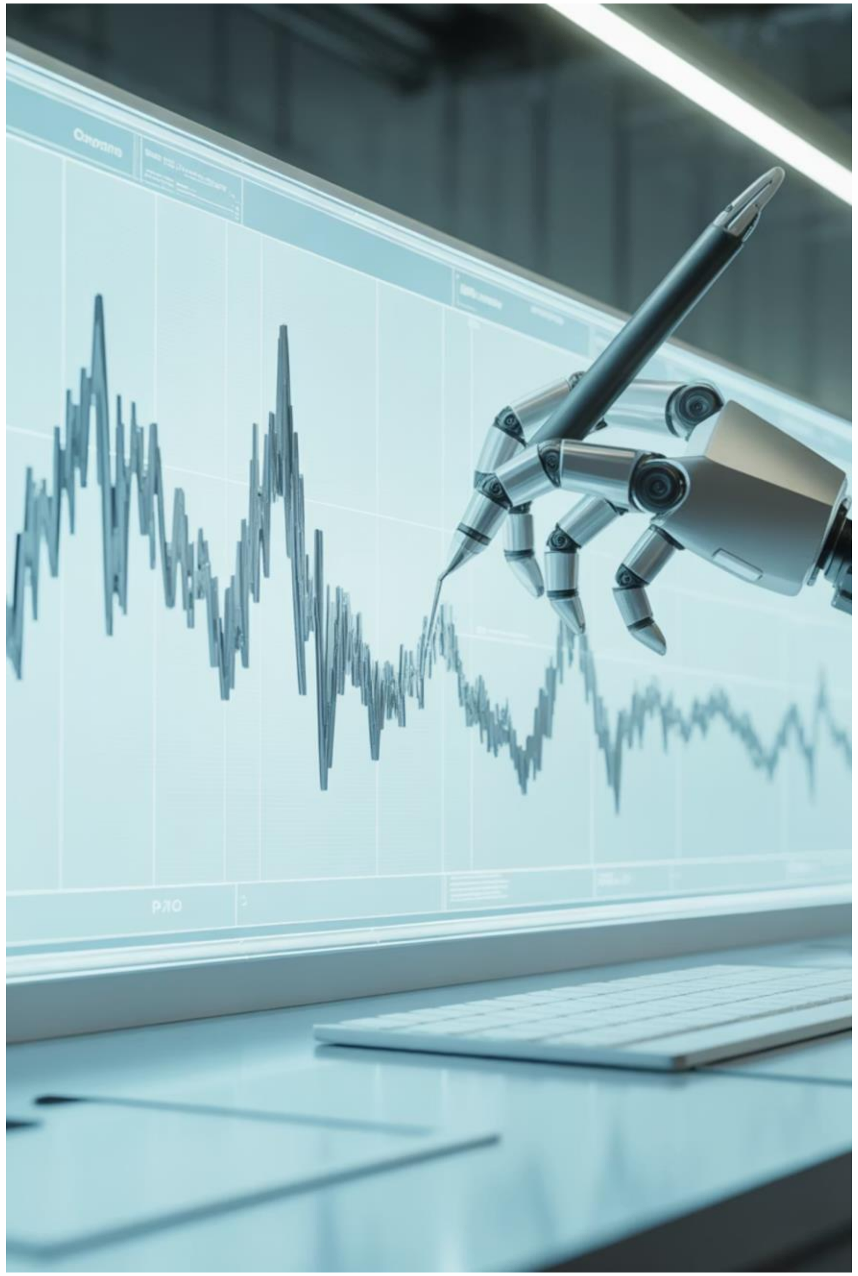
Overfitting happens when the model learns the training data *too well*, including all the noise and random variations.

Training Phase

The model performs **perfectly** on training data—memorizing every example including errors.

Testing Phase

The model performs **poorly** on new data because it learned specific quirks instead of general patterns.



Finding the Optimal Fit: Just Right! 🎯

The best model achieves the perfect balance between **bias** and **variance**.

It learns enough patterns to make accurate predictions without memorizing noise or being too simplistic.

This sweet spot is called **optimal fitting**—like Goldilocks finding the porridge that's just right!

Not Too Complex

Avoids high variance

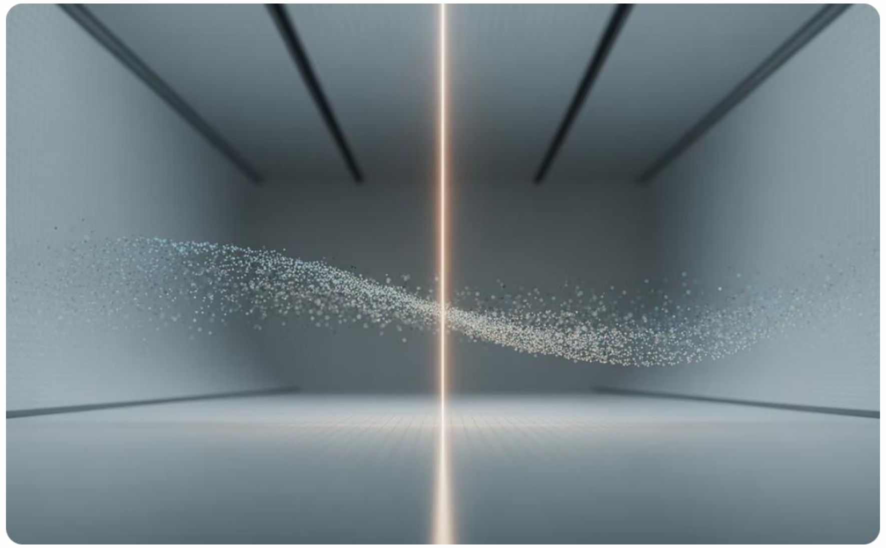
Not Too Simple

Avoids high bias

Just Right

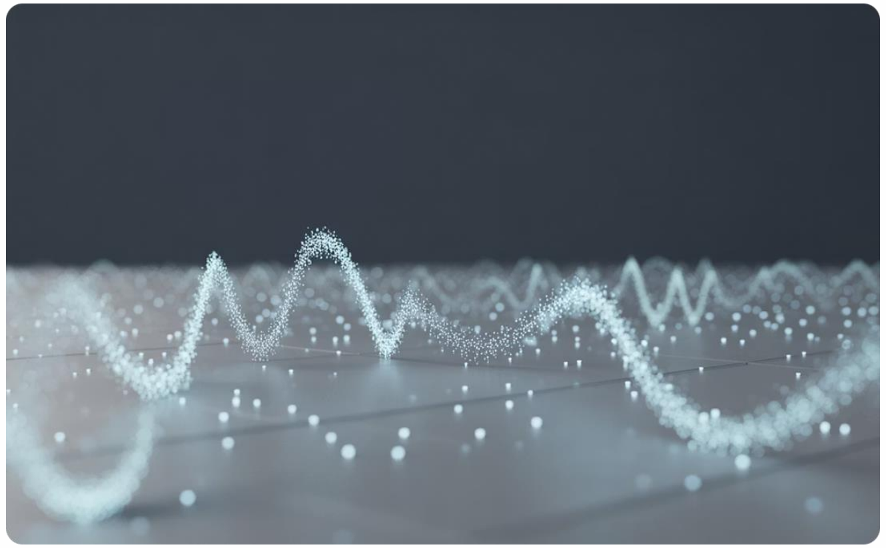
Optimal fitting achieved

Visual Story: Underfitting vs Overfitting vs Optimal Fit



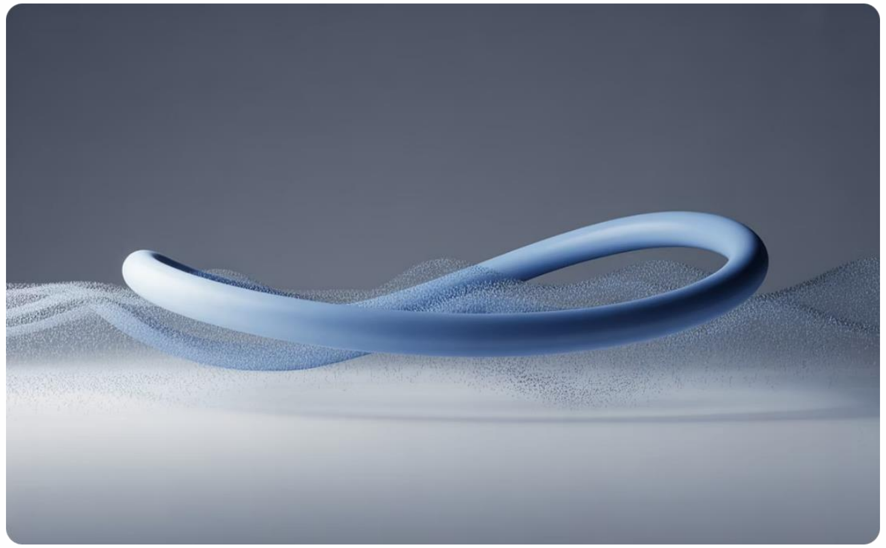
Underfitting

Line is too flat and misses most data points—too simple to capture patterns.



Overfitting

Line zigzags through every single point—memorizes noise instead of learning patterns.



Optimal Fit

Smooth curve captures the overall trend—learns real patterns, ignores noise.

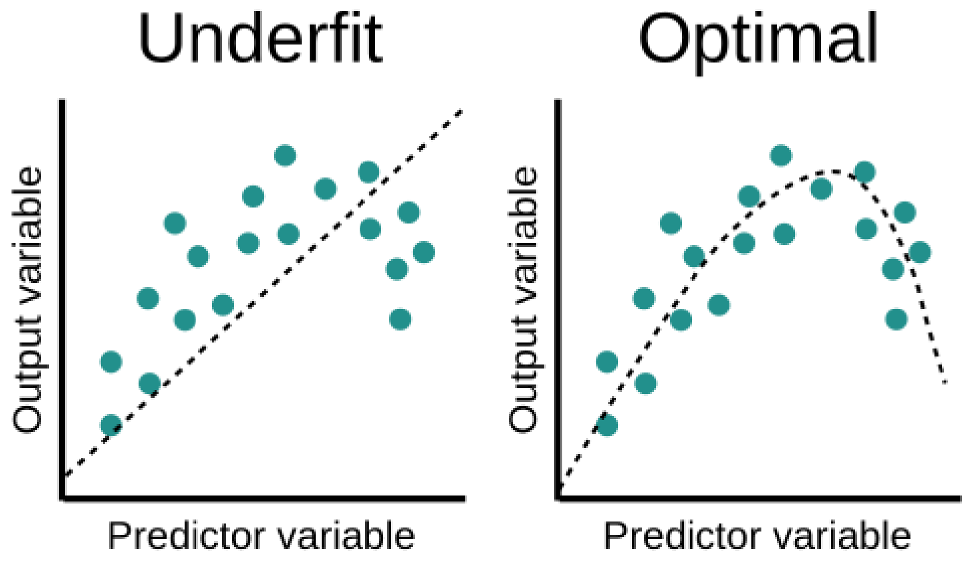
Visual Story: Underfitting vs Overfitting vs Optimal Fit

Underfitting

When a machine learning model is said to be "underfitting", it means that our model fails to produce good results because of an oversimplified model.

Such a model can neither model the training data nor generalize over new data.

When such a situation occurs, we say that the model has a "high bias".



Techniques to address underfitting problem:

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

- https://jashrathod.github.io/2021-09-30-underfitting-overfitting-and-regularization/

Overfitting

Overfitting is the opposite case of underfitting.

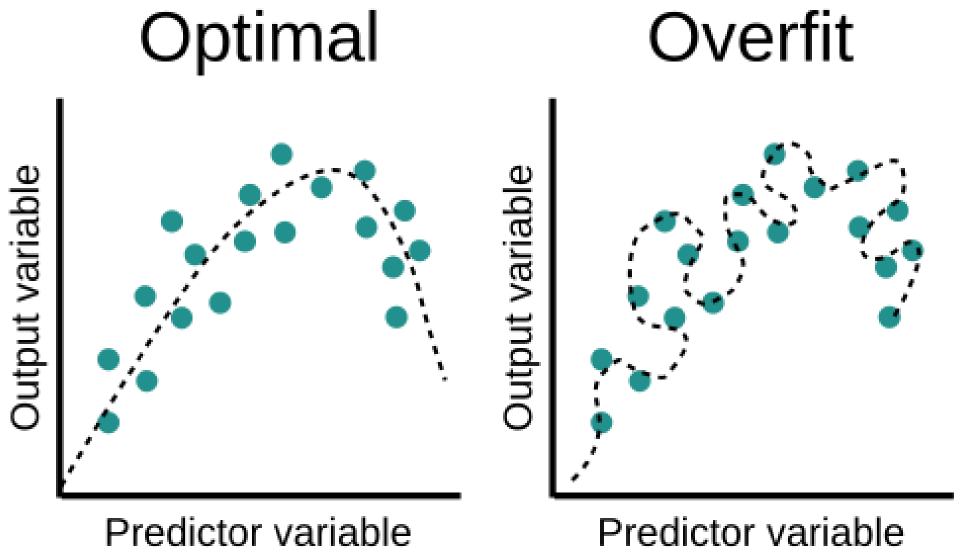
Here, our model produces good results on training data but performs poorly on testing data.

This happens because our model fits the training data so well that it leaves very little or no room for generalization over new data.

When overfitting occurs, we say that the model has “high variance”.

How to address Overfitting?

1. Cross-Validation
2. Train with more data
3. Remove features
4. Early stopping
5. Regularization
6. Ensembling



- https://jashrathod.github.io/2021-09-30-underfitting-overfitting-and-regularization/

Variance and Bias

In machine learning, variance and bias are two important concepts that are related to the accuracy of a model's predictions.

Bias refers to the tendency of a model to consistently underestimate or overestimate the true values of the target variable.

A model with high bias is said to be underfitting the data, meaning it is not complex enough to capture the patterns in the data.

https://www.youtube.com/watch?v=EuBBz3bI-aA

Variance and Bias (Contd.)

Variance refers to the tendency of a model to be highly sensitive to small fluctuations in the training data, which can result in overfitting.

A model with high variance is said to be overfitting the data, meaning it is too complex and has learned the noise or idiosyncrasies in the training data instead of the underlying patterns.

The goal in machine learning is to achieve a balance between bias and variance, which is often referred to as the bias-variance tradeoff.

https://www.youtube.com/watch?v=EuBBz3bI-aA

Why Does This Matter?

Understanding machine learning basics empowers you to create better models and solve real-world problems that impact millions of people!

Weather Forecasts

Predicting rain, storms, and temperature helps people plan their days safely.

Voice Assistants

Understanding speech and answering questions using pattern recognition.

Smart Games

Creating opponents that learn from your playing style and adapt.

Healthcare

Helping doctors diagnose diseases earlier and more accurately.

Cost Function

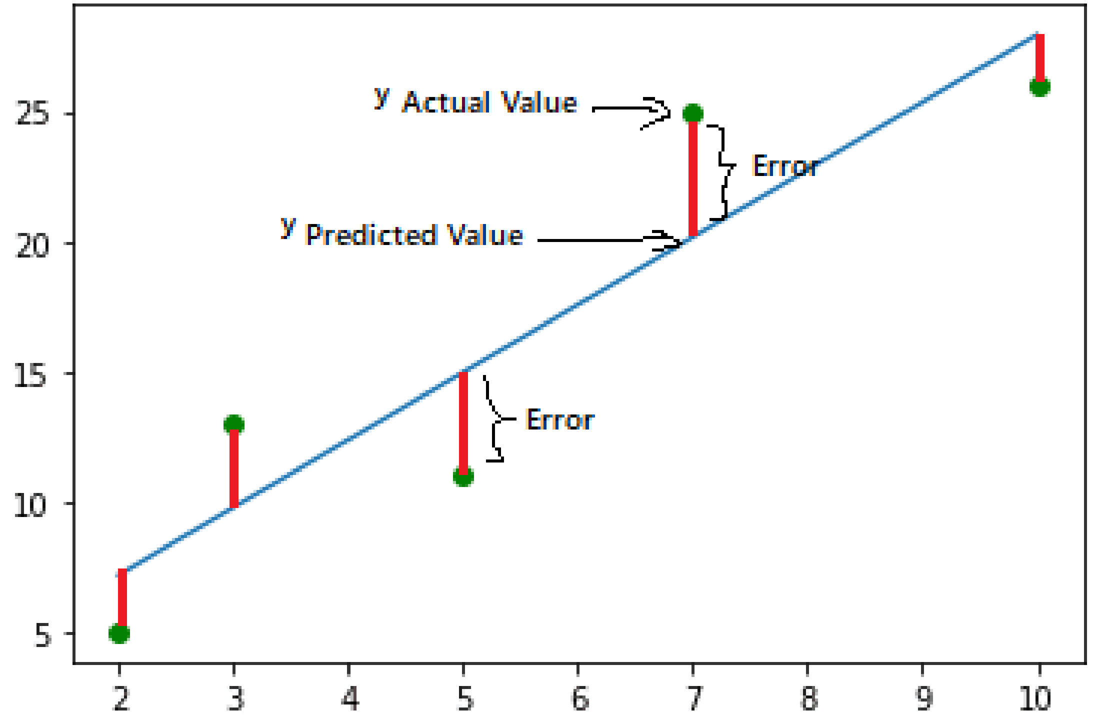
Definition: Cost Function is a function that determines how well a Machine Learning model performs for a given set of data.

Different models have different formulations for calculating the cost function.

It is easy to understand the ‘cost function’ in the case of Linear Regression.

Error: Difference between Actual and Predicted Values
Cost Functions: Mea of Sum of Squared Errors

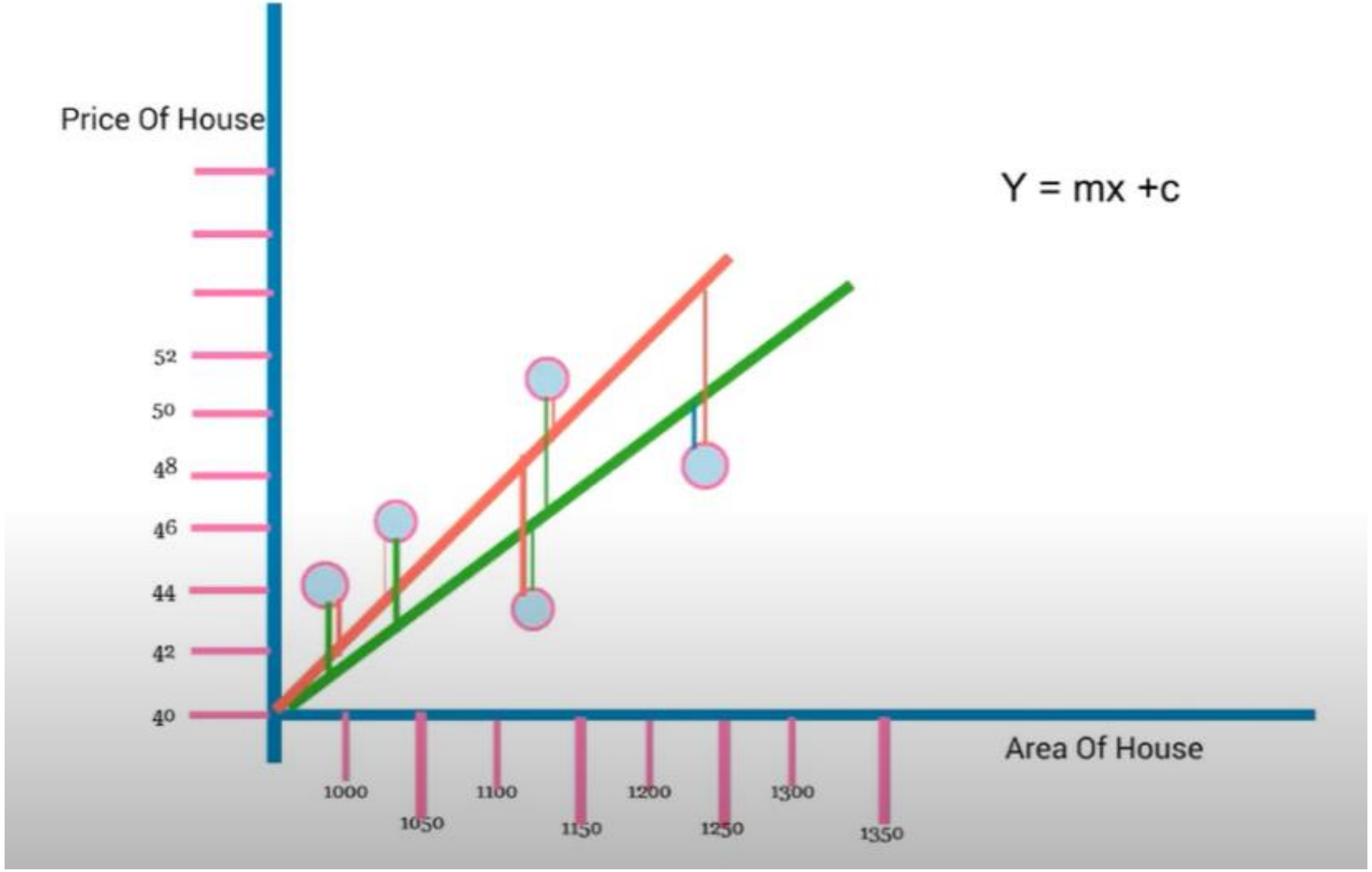
$$Cost\ func \rightarrow \frac{1}{n}\left(\sum_{i=1}^{n}(actual_y - predict_y)^2\right)$$



- Ref

Cost Function: Watch this video

https://www.youtube.com/watch?v=w9FhLszh3O0



Which model
(red line or green line)
is better?

Quick Recap 📝

1. ## Predict Targets with Features

ML models predict targets (outcomes) using features (input clues) to make smart decisions.

2. ## Engineering and Weights

Feature engineering selects the best clues, while weights determine each feature's importance.

3. ## Balance Bias and Variance

Too simple (high bias) or too complex (high variance) both hurt model accuracy.

4. ## Find the Sweet Spot

Avoid underfitting and overfitting by finding the optimal fit—just the right complexity!

Your Turn: Imagine Your Own ML Model! 🚀

1

Choose Your Target

What would you want to predict? Movie ratings? Sports scores? Your favorite food?

2

Pick Your Features

What clues would help you predict it? Think about which information matters most!

3

Design Wisely

How would you make sure your model learns real patterns without overfitting or underfitting?

Let's start exploring the exciting world of machine learning together! The future is full of possibilities, and you have the power to create intelligent systems that make a difference. 🌟