Unit 1. Introduction

1. Learning in the Context of Machine Learning

Machine Learning (ML) is a branch of artificial intelligence that focuses on enabling systems to learn and improve automatically through experience without being explicitly programmed. The process can be broken down into:

- 1 **Data Acquisition:** Gathering raw data from various sources such as sensors, logs, or databases.
- 2 **Learning Patterns:** Using algorithms to identify and learn patterns and relationships in the data.
- 3 **Generalization:** Applying learned insights to make predictions or decisions on unseen data.

Types of learning include:

- **Supervised Learning:** Learning from labeled data.
- **Unsupervised Learning:** Discovering patterns in unlabeled data.
- **Reinforcement Learning:** Learning optimal actions through feedback.

2. Three Phases of Performing ML

1. **Data Preparation:**
   - **Data Collection:** Sources can include APIs, web scraping, or manual entry.
   - **Data Cleaning:** Handling missing values, removing duplicates, and correcting errors.
   - **Feature Engineering:** Creating new features, encoding categorical variables, and scaling data.
   - **Data Splitting:** Dividing the dataset into training, validation, and testing sets, typically in a ratio like 70:15:15.
2. **Model Building:**
   - **Algorithm Selection:** Choosing an appropriate algorithm based on the problem (e.g., regression, classification, clustering).
   - **Training:** Feeding the training data into the algorithm to create the model.
   - **Hyperparameter Tuning:** Using techniques like grid search or random search to optimize model performance.
3. **Model Evaluation and Deployment:**
   - **Evaluation Metrics:** Metrics include accuracy, precision, recall, F1-score, and RMSE, depending on the problem type.
   - **Deployment:** Integrating the model into an application for real-time predictions.

    - **Monitoring:** Continuously assessing performance and updating the model if needed.

3. Algorithms and Models in ML

- **Algorithms:** Defined processes or sets of rules for solving specific types of problems.
    - Examples: Gradient Descent, k-Means, Random Forests.
- **Models:** Outputs of algorithms representing learned patterns from data.
    - Examples: Linear regression models, neural network architectures.

4. Logical, Geometric, and Probabilistic Models

1. **Logical Models:**
    - Rely on clear, interpretable rules or logical statements.
    - Examples: Decision Trees, Rule-Based Systems.
    - Best suited for interpretable decision-making tasks.
2. **Geometric Models:**
    - Represent data as geometric objects (e.g., points, lines, planes).
    - Examples: Support Vector Machines (SVM), k-Nearest Neighbors (k-NN).
    - Useful for tasks like pattern recognition or nearest neighbor searches.
3. **Probabilistic Models:**
    - Use probability theory to handle uncertainty and make predictions.
    - Examples: Naïve Bayes, Hidden Markov Models (HMM), Bayesian Networks.
    - Ideal for problems involving uncertainty or noisy data.

5. Underfitting, Overfitting, and Right Models

- **Underfitting:**
    - Occurs when the model is too simplistic to capture underlying data patterns.
    - Indicators: Low training and testing accuracy.
    - Solution: Use more complex algorithms, add features, or increase training time.
- **Overfitting:**
    - Happens when the model learns noise and details in the training data, reducing generalization capability.
    - Indicators: High training accuracy but low testing accuracy.
    - Solution: Use regularization (e.g., L1, L2), simplify the model, or increase the training data size.
- **Right Model:**
    - Balances complexity and generalization.
    - Achieves good performance on both training and unseen data.
    - Methods: Cross-validation, proper hyperparameter tuning.

6. Practical ML Examples

1. **Healthcare:**
   - Predicting disease risks (e.g., diabetes prediction).
   - Personalized treatment using genetic data.
2. **Finance:**
   - Detecting fraudulent transactions.
   - Predicting stock market trends using time-series analysis.
3. **E-commerce:**
   - Building recommendation systems for products.
   - Segmenting customers based on buying behavior.
4. **Autonomous Vehicles:**
   - Object detection and classification.
   - Path planning and control systems.
5. **Natural Language Processing (NLP):**
   - Sentiment analysis for reviews.
   - Machine translation and chatbots.

7. Types of ML Problems

1. **Supervised Learning:**
   - Learning with labeled data.
   - Examples: Regression (predicting continuous values), Classification (categorical prediction).
2. **Unsupervised Learning:**
   - Finding hidden patterns in unlabeled data.
   - Examples: Clustering (e.g., k-Means), Dimensionality Reduction (e.g., PCA).
3. **Semi-Supervised Learning:**
   - Using a mix of labeled and unlabeled data.
   - Example: Using small labeled datasets to guide clustering of larger unlabeled datasets.
4. **Reinforcement Learning:**
   - Agents learn by interacting with the environment and receiving rewards or penalties.
   - Examples: Game-playing AI (e.g., AlphaGo), robotic control systems.

8. Classification of ML Algorithms

1. **Based on Learning Style:**
   - Supervised, Unsupervised, Semi-supervised, Reinforcement Learning.
2. **Based on Functionality:**
   - **Classification:** Predicting discrete categories (e.g., spam detection).
   - **Regression:** Predicting continuous values (e.g., house price prediction).
   - **Clustering:** Grouping similar data points (e.g., customer segmentation).

    - **Dimensionality Reduction:** Simplifying data structure (e.g., PCA).

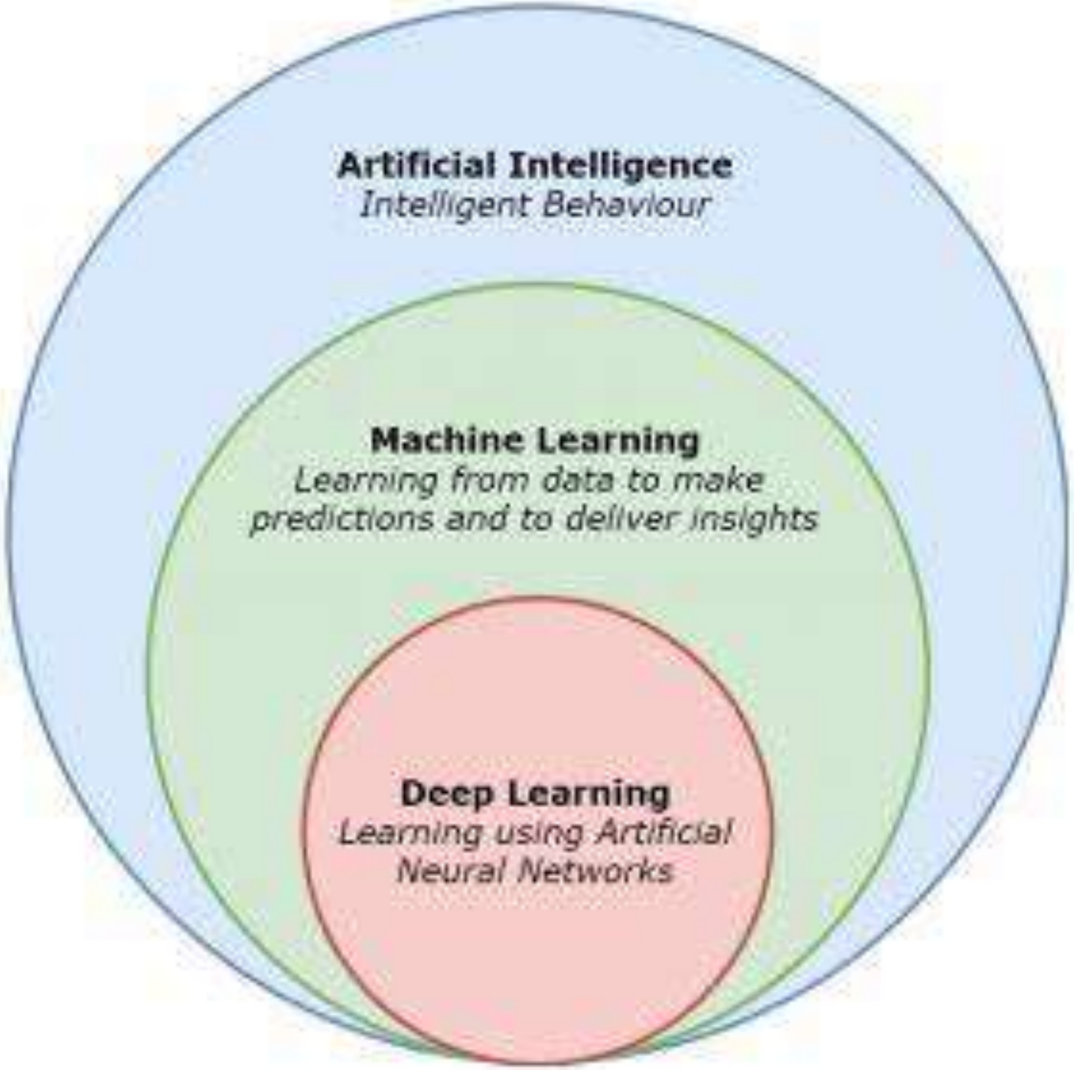
3. **Based on Methodology:**
    - Instance-based: k-Nearest Neighbors (k-NN).
    - Model-based: Linear Regression, Neural Networks.
4. **Based on Output:**
    - Discrete Outputs: Used for classification problems.
    - Continuous Outputs: Used for regression problems.
5. **Based on Data Structure:**
    - Structured Data: Decision Trees, Logistic Regression.
    - Unstructured Data: Convolutional Neural Networks (CNNs) for images, Recurrent Neural Networks (RNNs) for sequences.
6. **Based on Computational Approach:**
    - Batch Learning: Processes entire datasets at once.
    - Online Learning: Processes data incrementally, useful for real-time systems.

Unit 1. Introduction

Contents

- Learning in the context of ML,
- Three phases of performing ML,
- Algorithms and Models in ML,
- Logical, Geometric and Probabilistic models,
- Underfitting, Overfitting and Right models,
- Practical ML examples,
- Types of ML problems,
- Classification of ML algorithms.

Machine Learning



Artificial Intelligence
Intelligent Behaviour

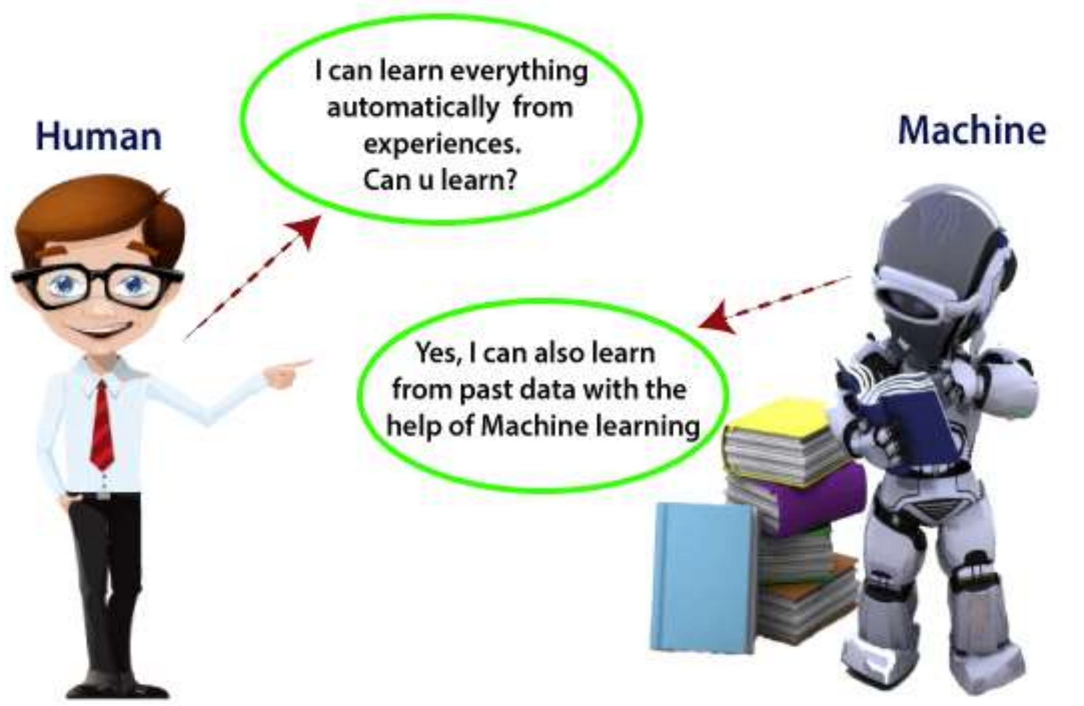
Machine Learning
Learning from data to make predictions and to deliver insights

Deep Learning
Learning using Artificial Neural Networks

What is Machine Learning?

Machine learning (ML) is a branch of artificial intelligence (AI) focused on enabling computers and machines to imitate the way that humans learn,

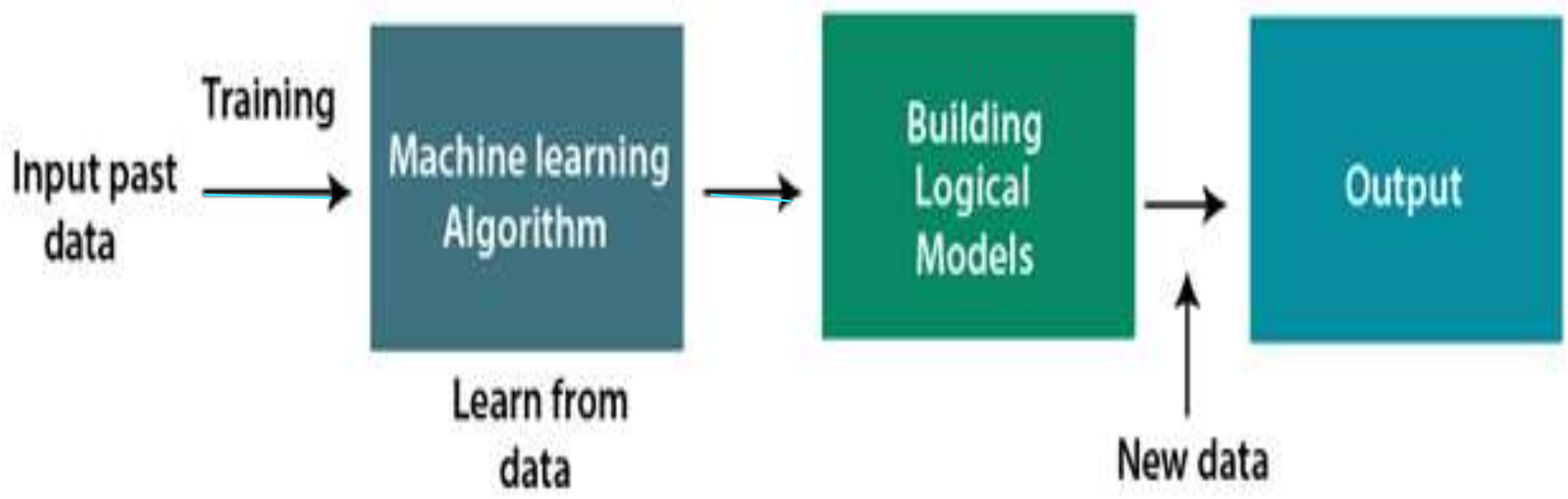
- to perform tasks autonomously,
- to improve their performance and accuracy through experience and exposure to more data.


Human
I can learn everything automatically from experiences. Can u learn?
Machine
Yes, I can also learn from past data with the help of Machine learning

A machine can learn if it can gain more data to improve its performance.

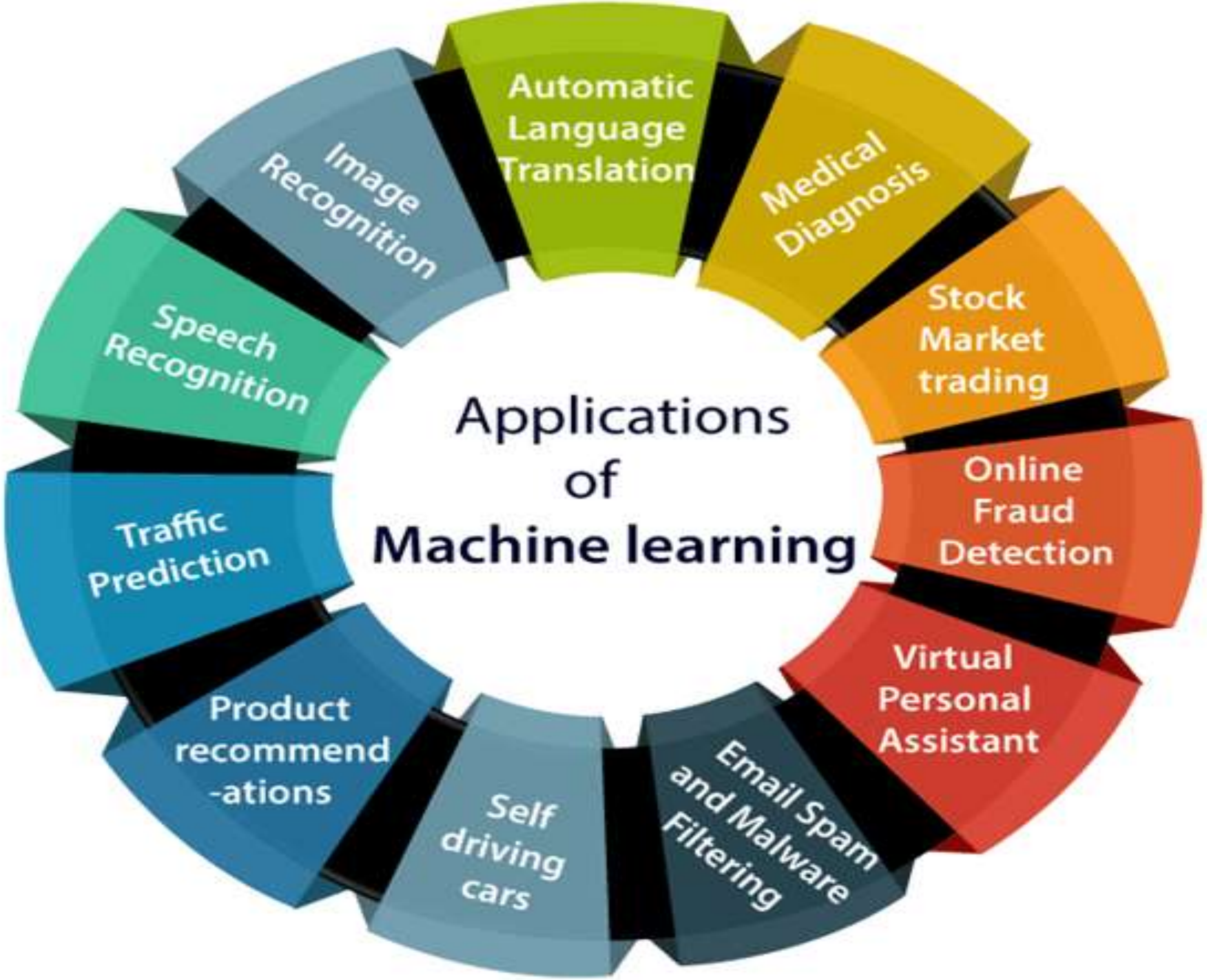
How does Machine Learning work

- A machine learning system builds prediction models, learns from previous data, and predicts the output of new data whenever it receives it.



Machine learning (ML) is a type of Artificial Intelligence (AI) that allows computers to learn and make decisions without being explicitly programmed. It involves feeding data into algorithms that can then identify patterns and make predictions on new data.

Applications of Machine Learning


Applications of Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition

Three phases of performing ML

Phase 1: Discovery

define our problem clearly

Phase 2: Development

Phase 3: Deployment

Phase 1: Discovery

- Define Problem Clearly
  - Are we trying to predict stock prices?
  - Identify different cat breeds from photos?
  - Why is this problem important, and how will solving it make a difference?
- understand who will use our solution
  - Are they tech-savvy teenagers or busy executives?
  - What do they need, and how can our solution make their lives easier?
- reality check
  - can machine learning solve our problem?
  - Is this the right tool, or are we overcomplicating things? Could there be a simpler, more effective way?

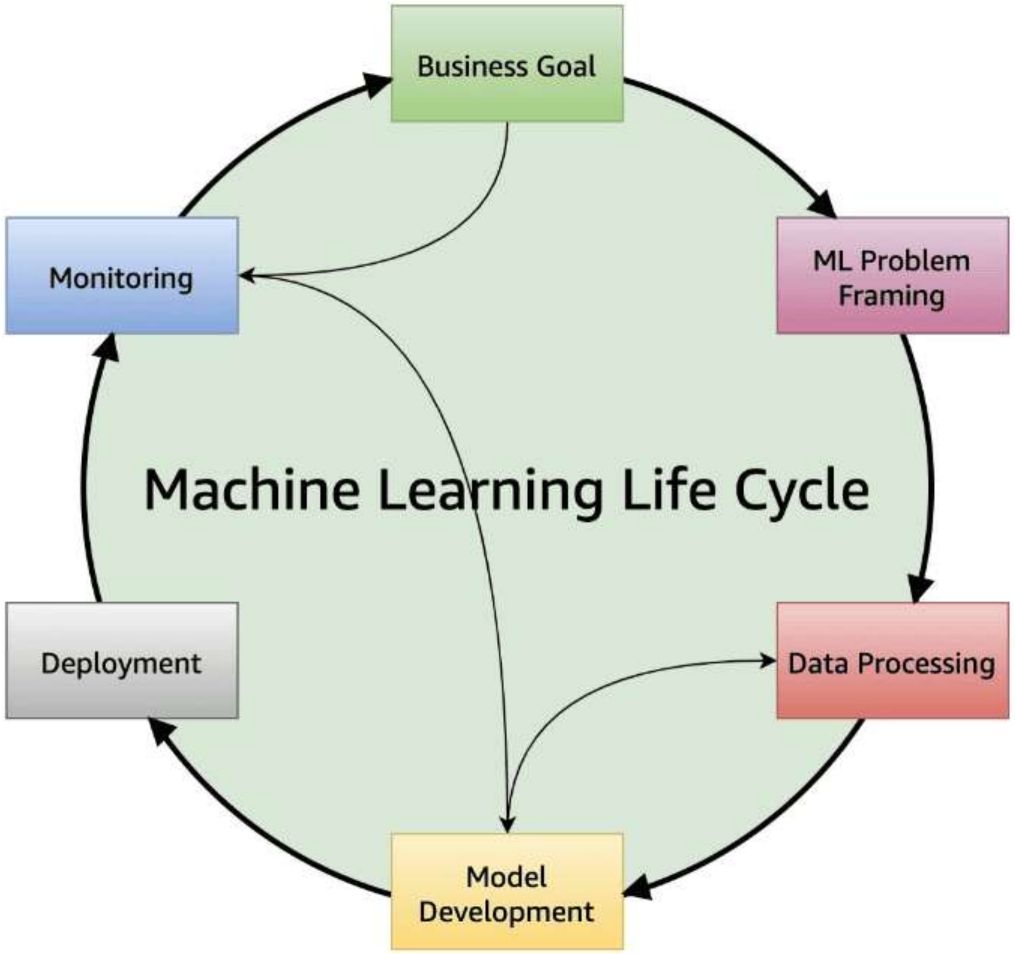
Phase 2: Development

- Data collection
- Feature Engineering
- Building Model(Training)
- Evaluating Model(Testing)

Phase 3: Deployment

- Tracking and Monitoring performance of model for real data?

Machine Learning lifecycle



Machine Learning Lifecycle

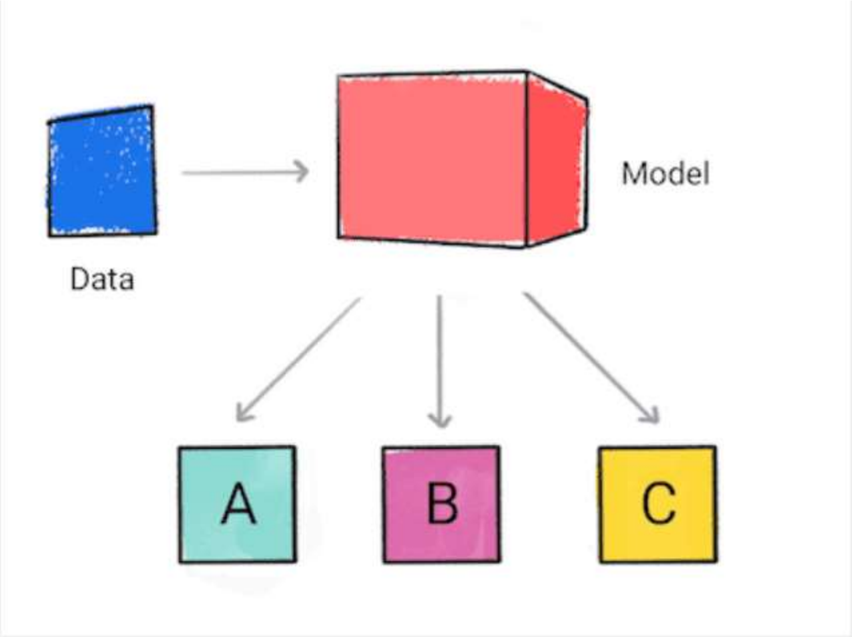
1. Problem Definition
2. Data Collection
3. Data Cleaning and Preprocessing
4. Exploratory Data Analysis (EDA)
5. Feature Engineering and Selection
6. Model Selection
7. Model Training
8. Model Evaluation and Tuning
9. Model Deployment
10. Model Monitoring and Maintenance

Defining the Business Goal

- What **specific problem** are we trying to solve?
- How will the solution impact the business?
- What are the success metrics?

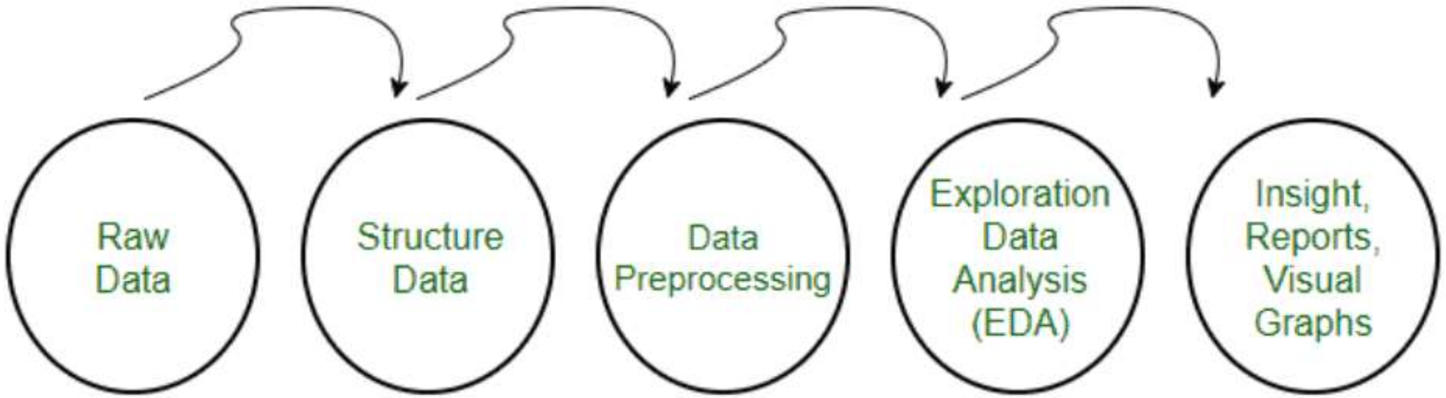
ML Problem Framing

- Defining the machine learning task
- Identifying relevant features and labels
- Setting up an **evaluation framework** for the model



Data Processing

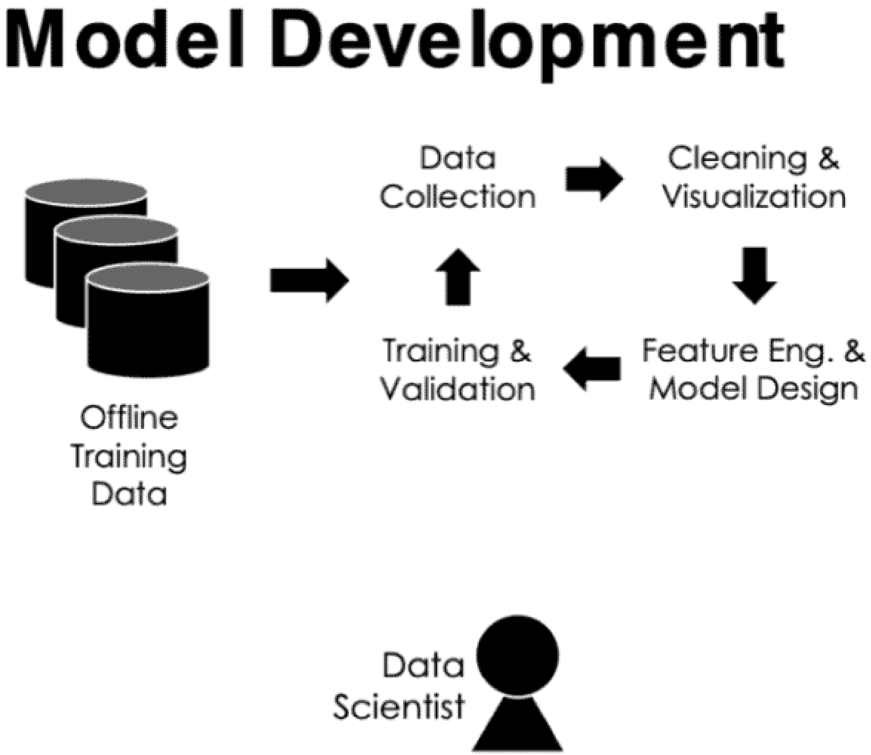
- Data cleaning (handling missing values, removing duplicates)
- Feature engineering (creating new features, selecting important features)
- Data splitting (dividing data into training, validation, and test sets)


Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Data Preprocessing

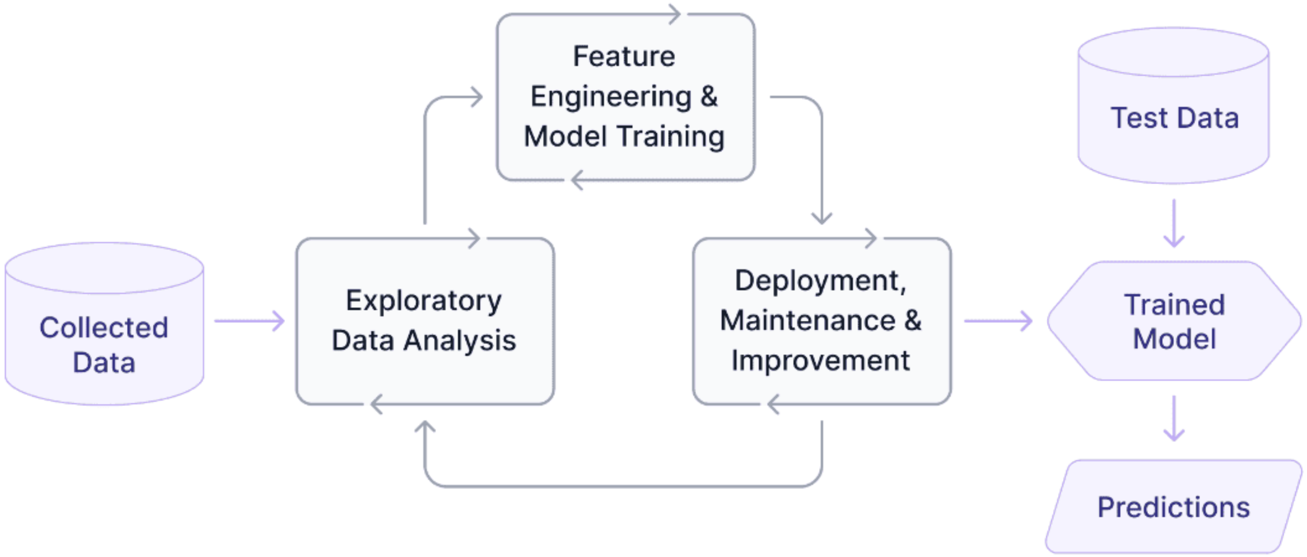
Model Development

- Choosing the right algorithm(s) for the problem
- Training the model on the training data
- Evaluating the model using validation data to tune hyperparameters



Deployment

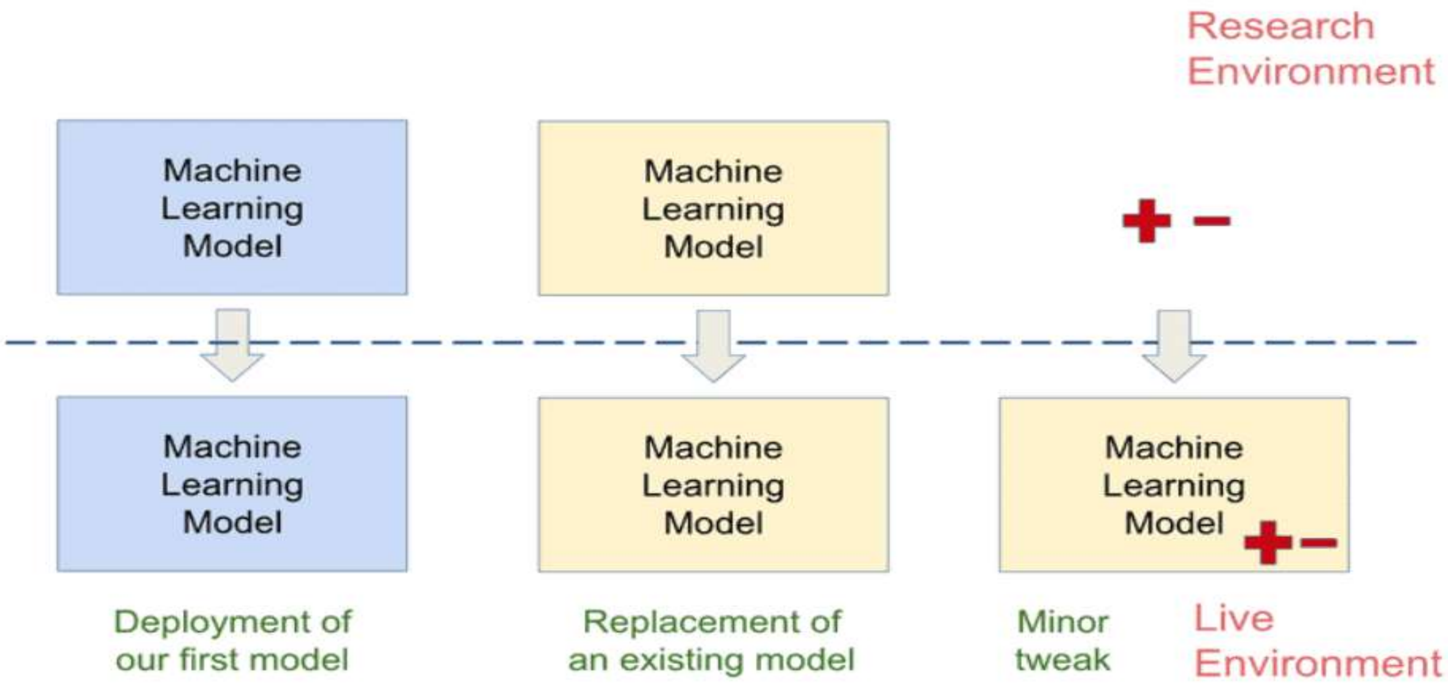
- Model packaging and serving
- Setting up an infrastructure for model inference
- Ensuring security and compliance in the production environment



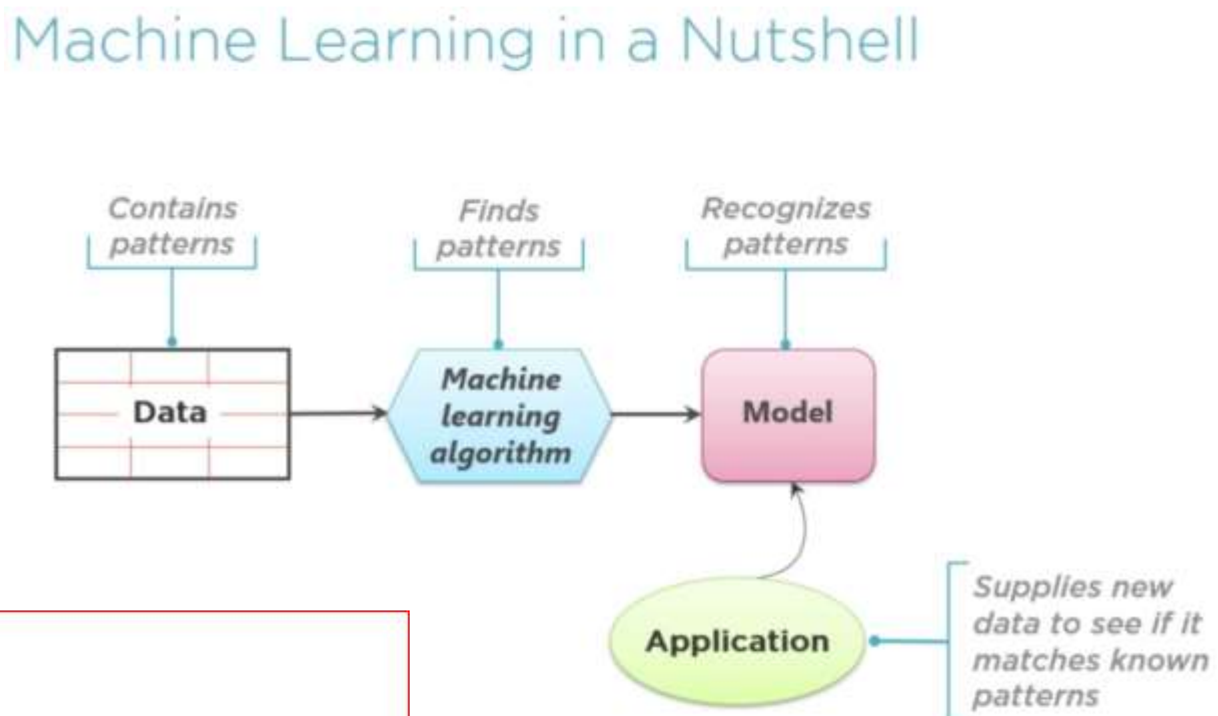
Collected Data
Exploratory Data Analysis
Feature Engineering & Model Training
Deployment, Maintenance & Improvement
Test Data
Trained Model
Predictions

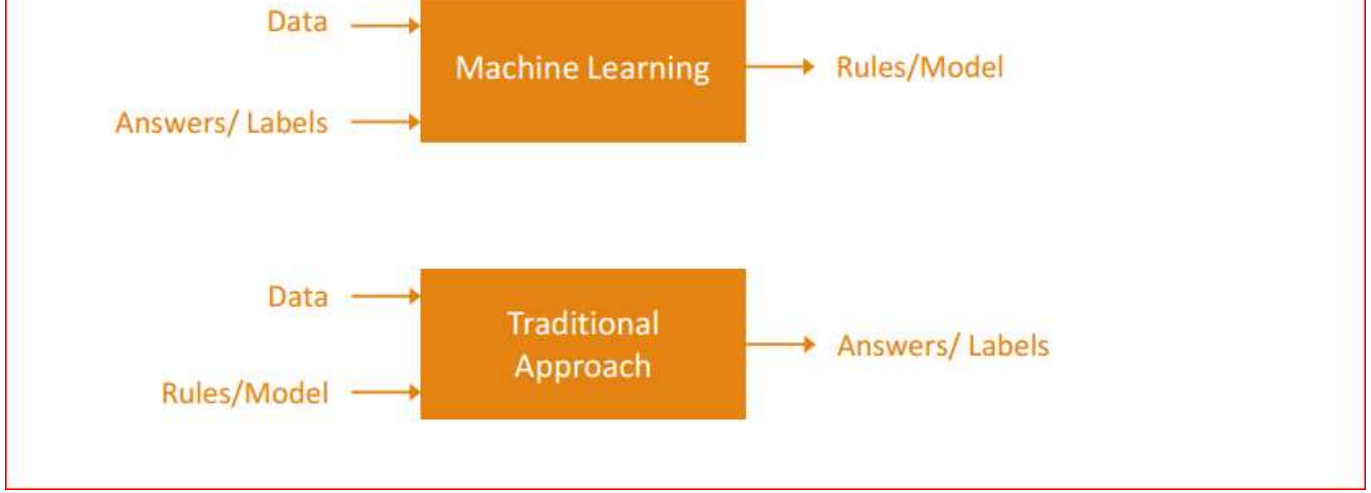
Monitoring

- Setting up monitoring tools to track model performance
- Analyzing feedback and updating the model as needed
- Implementing an automated **retraining pipeline** if required



What is model in ML


Machine Learning in a Nutshell
Contains patterns
Finds patterns
Recognizes patterns
Data
Machine learning algorithm
Model
Application
Supplies new data to see if it matches known patterns



ML Algorithms

- In machine learning (ML), **algorithms** are procedures or techniques used to optimize models, while **models** are mathematical representations that make predictions or decisions based on data.

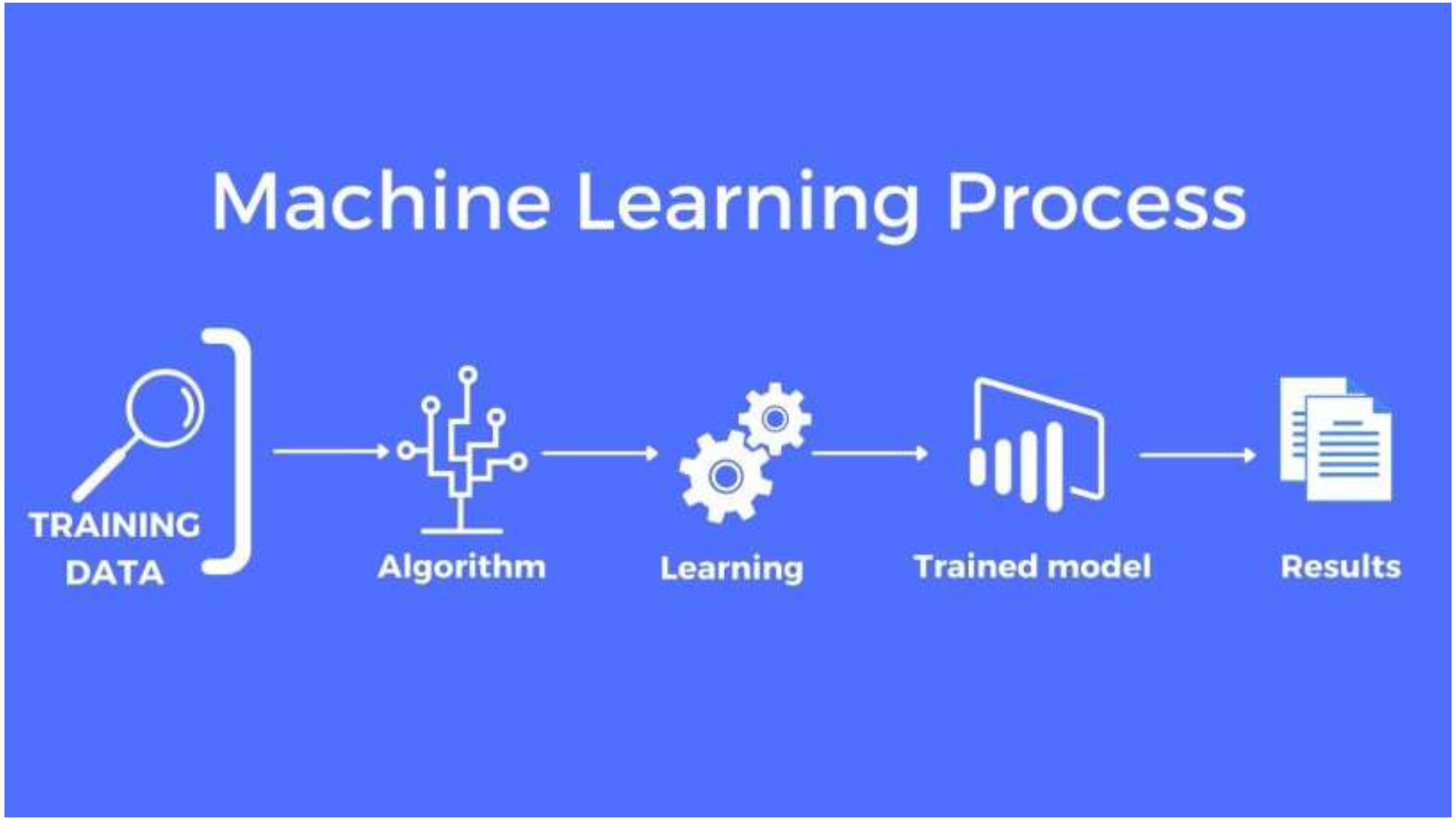
Supervised Learning Algorithms

1. **Linear Regression**
   1. Algorithm for predicting continuous values by minimizing the error in a linear relationship.
2. **Logistic Regression**
   1. Used for binary classification problems; models probabilities using a sigmoid function.
3. **Decision Tree**
   1. Splits data into subsets using feature-based conditions to make predictions.
4. **Random Forest**
   1. Ensemble method that uses multiple decision trees and aggregates their outputs.
5. **Support Vector Machines (SVM)**
   1. Finds the hyperplane that best separates classes in feature space.
6. **K-Nearest Neighbors (KNN)**
   1. Assigns labels based on the majority class among the k closest points.
7. **Gradient Boosting**
   1. Builds an additive model by optimizing a loss function iteratively. Examples: XGBoost, LightGBM.

ML Algorithms

- **Unsupervised Learning Algorithms**
- **K-Means Clustering**
  - Partitions data into k clusters based on centroids.
- **Hierarchical Clustering**
  - Builds a tree of clusters based on distance metrics.
- **Principal Component Analysis (PCA)**
  - Reduces dimensionality by identifying the principal components.
- **DBSCAN (Density-Based Spatial Clustering)**
  - Identifies clusters based on density and noise.
- **Autoencoders**
  - Neural networks that learn efficient representations of input data.

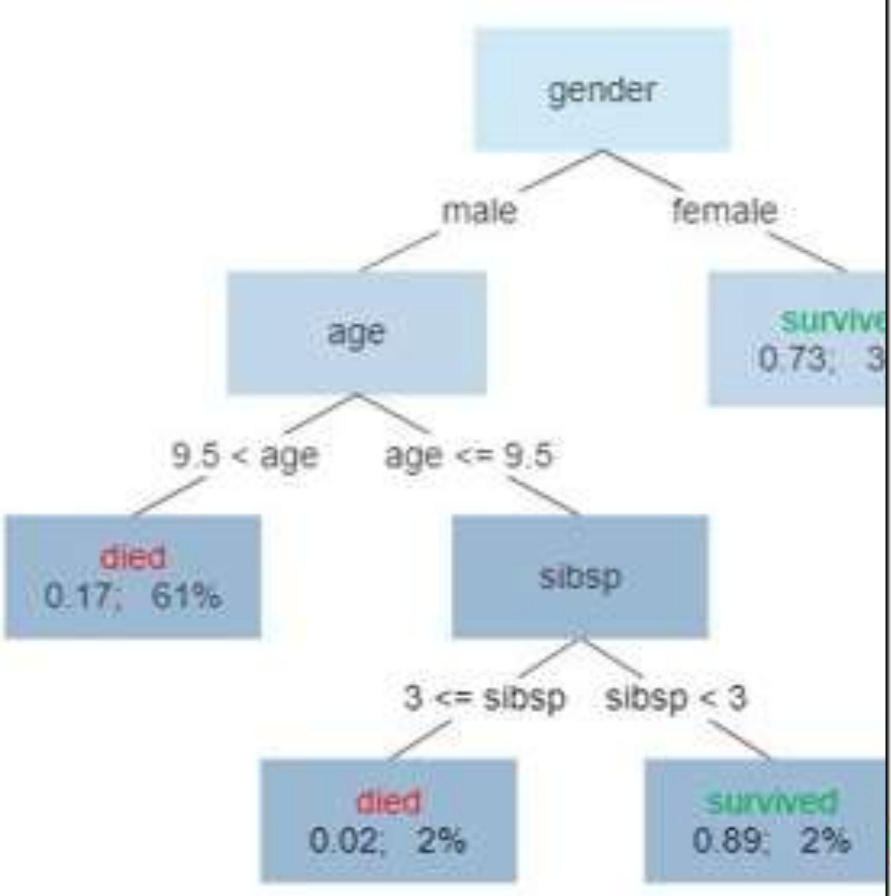
Models in Machine Learning

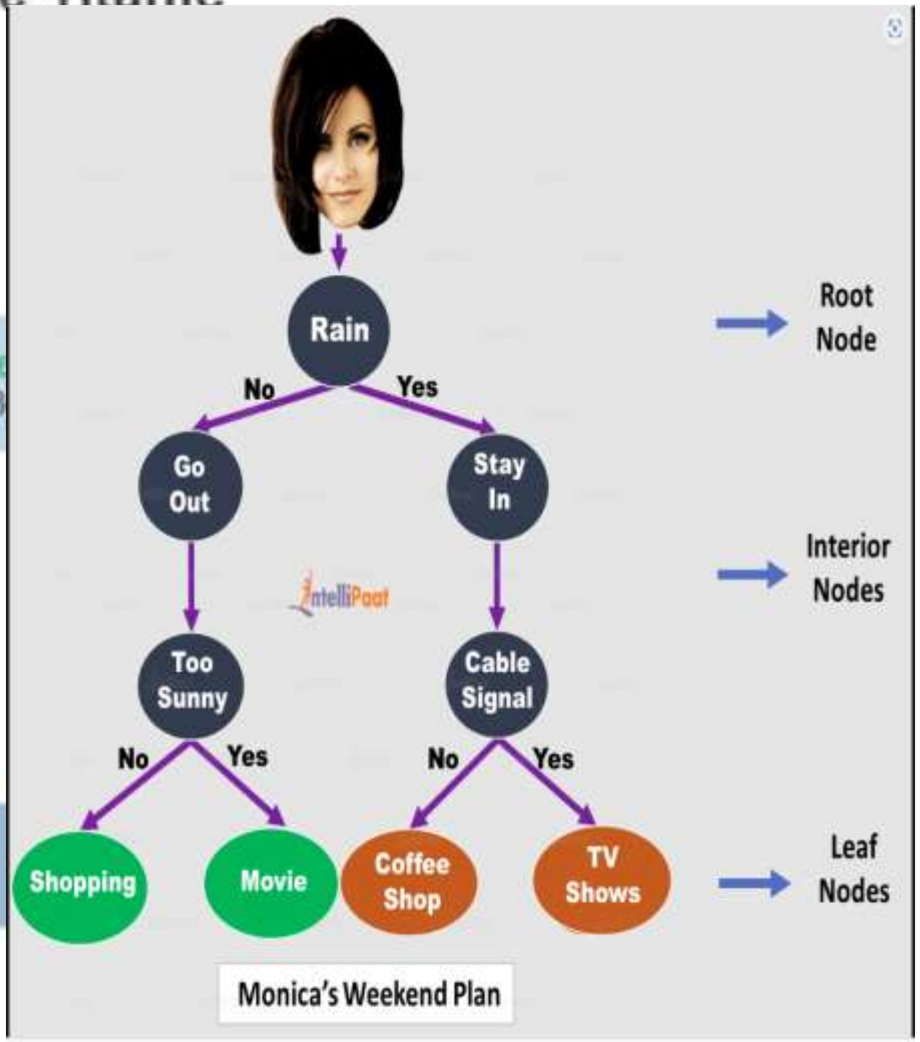


LOGICAL MODELS

- **“Logical”** becaus humans can unde
- E.g *if lottery = 1* structure, which

Survival of passengers on the Titanic





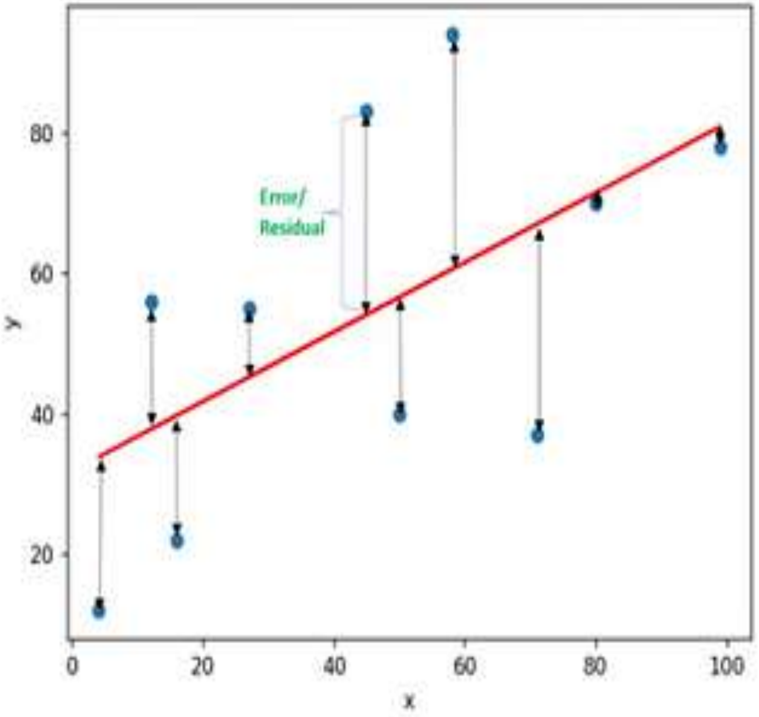
Geometric models/feature learning

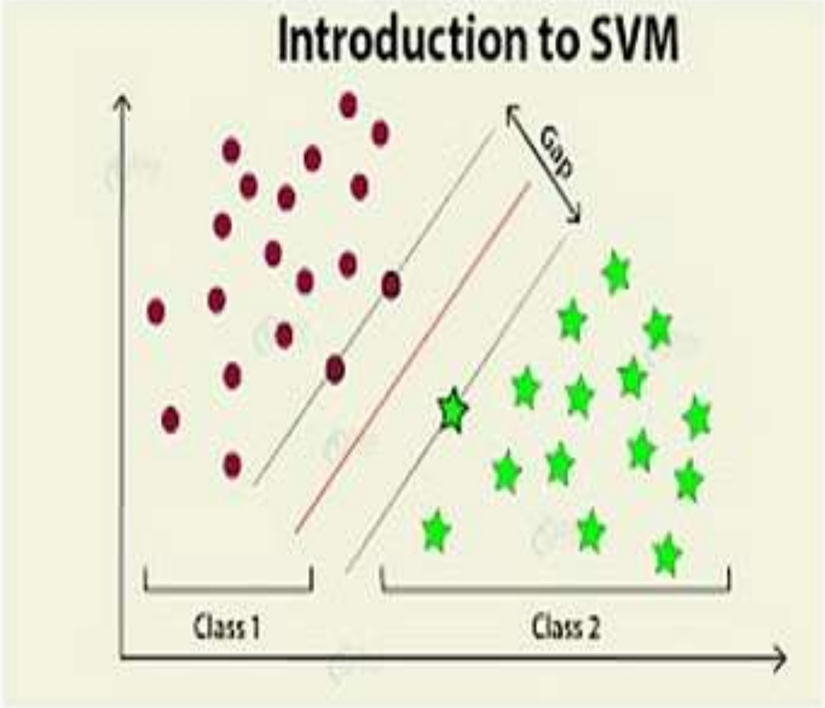
- Machine learning + computer vision to visual tasks.
- Types of Geometric Models

1. **Linear Models**.
2. **Distance based Models**.

Liners Models

- A geometric mo
concepts such a
as **Linear Model**
- Linear models e
estimation and r
- a) Hours spent s
- b) Amount of rai
- c) Electricity usa
- d) Suicide rates



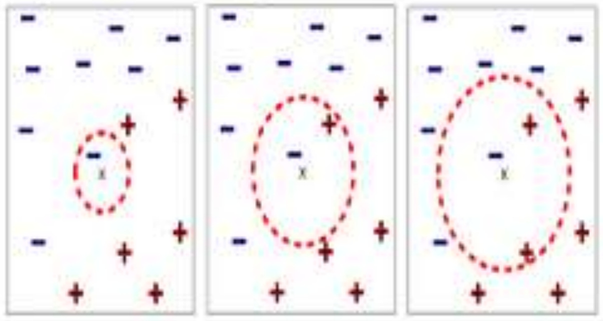


Distance Based Models

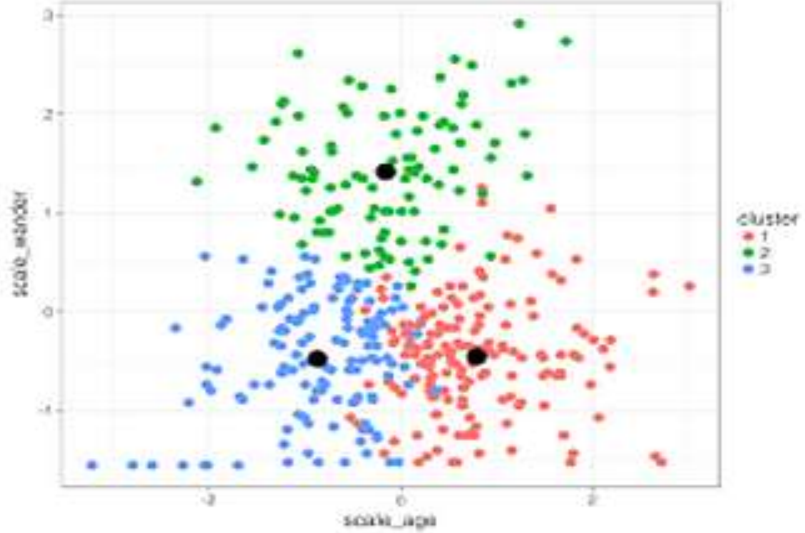
- If the distance between two instances is small then the instances are similar in terms of their feature values, and so nearby instances would be expected to receive the same classification or belong to the same cluster.

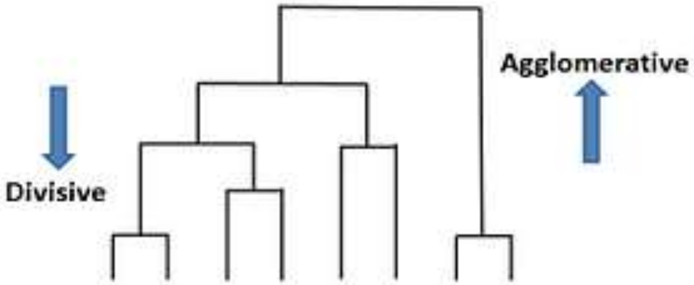
KNN
K means Clustering
Hierarchical Clustering

example of K nearest neighbor.



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor





PROBABILISTIC MODELS

- A probability model/method is based on the theory of probability, or the fact that randomness play a role in predicting future events.

e.g. Baysian classifier

$$\boxed{P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}}$$

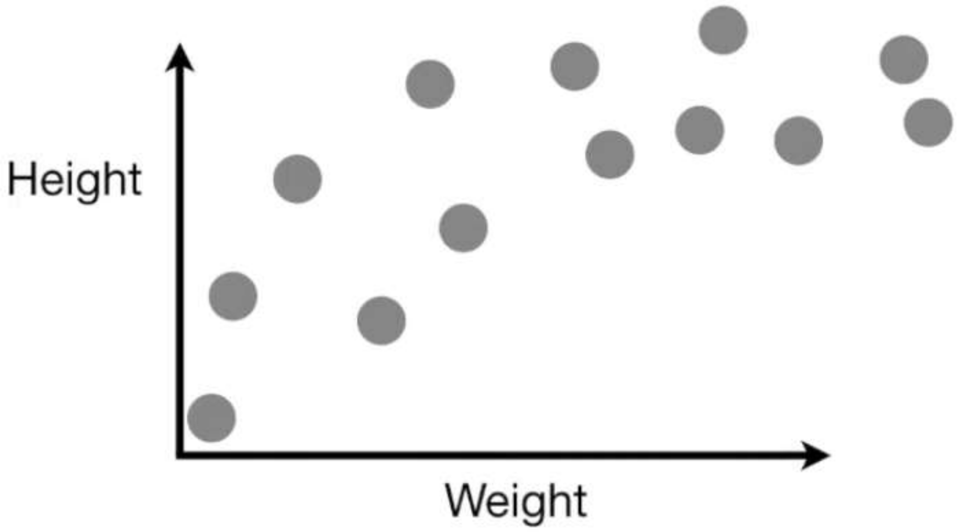
P(Y/X) = Posterior probability (probability of hypothesis is true given the evidence)

P(X/Y) = Likelihood ratio (probability of seeing the evidence if the hypothesis is true)

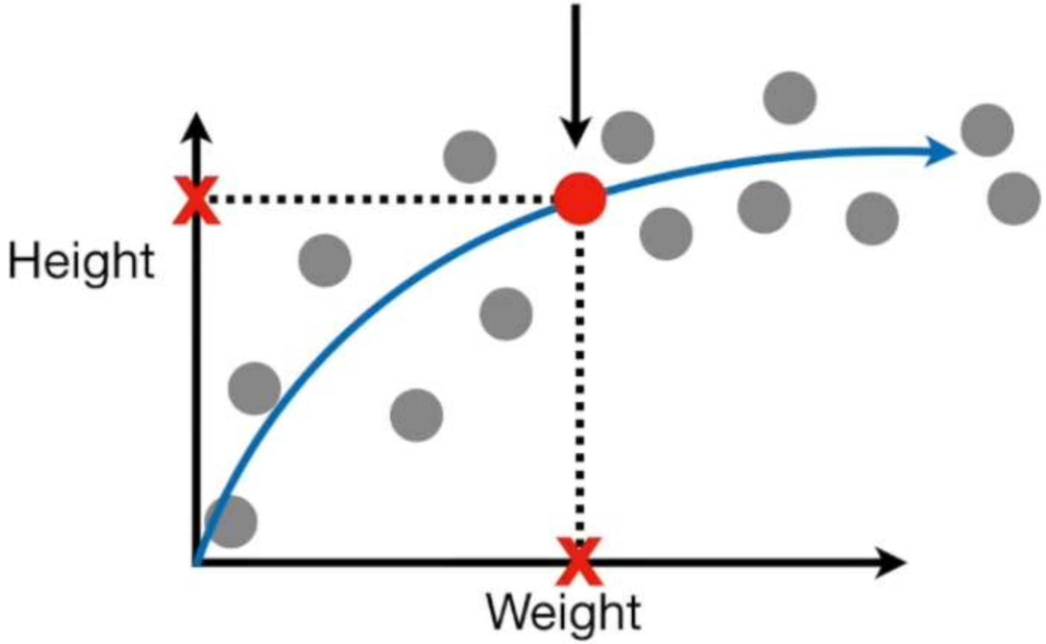
P(Y) = Class Prior probability (probability of hypothesis is true, before any evidence is present)

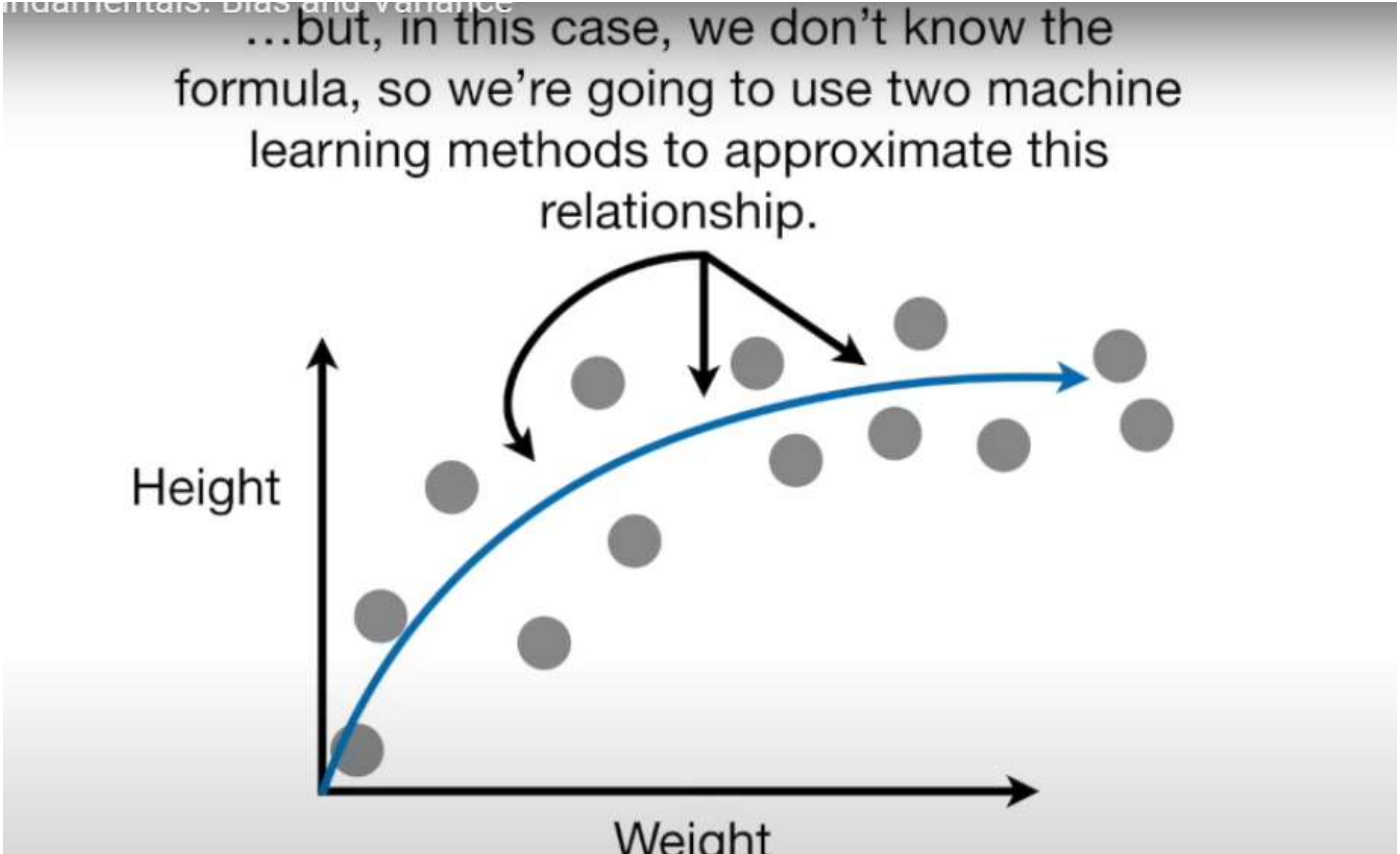
P(X) = Predictor Prior probability (probability of observing the evidence)

Imagine we measured the weight and height of a bunch of mice and plotted the data on a graph…



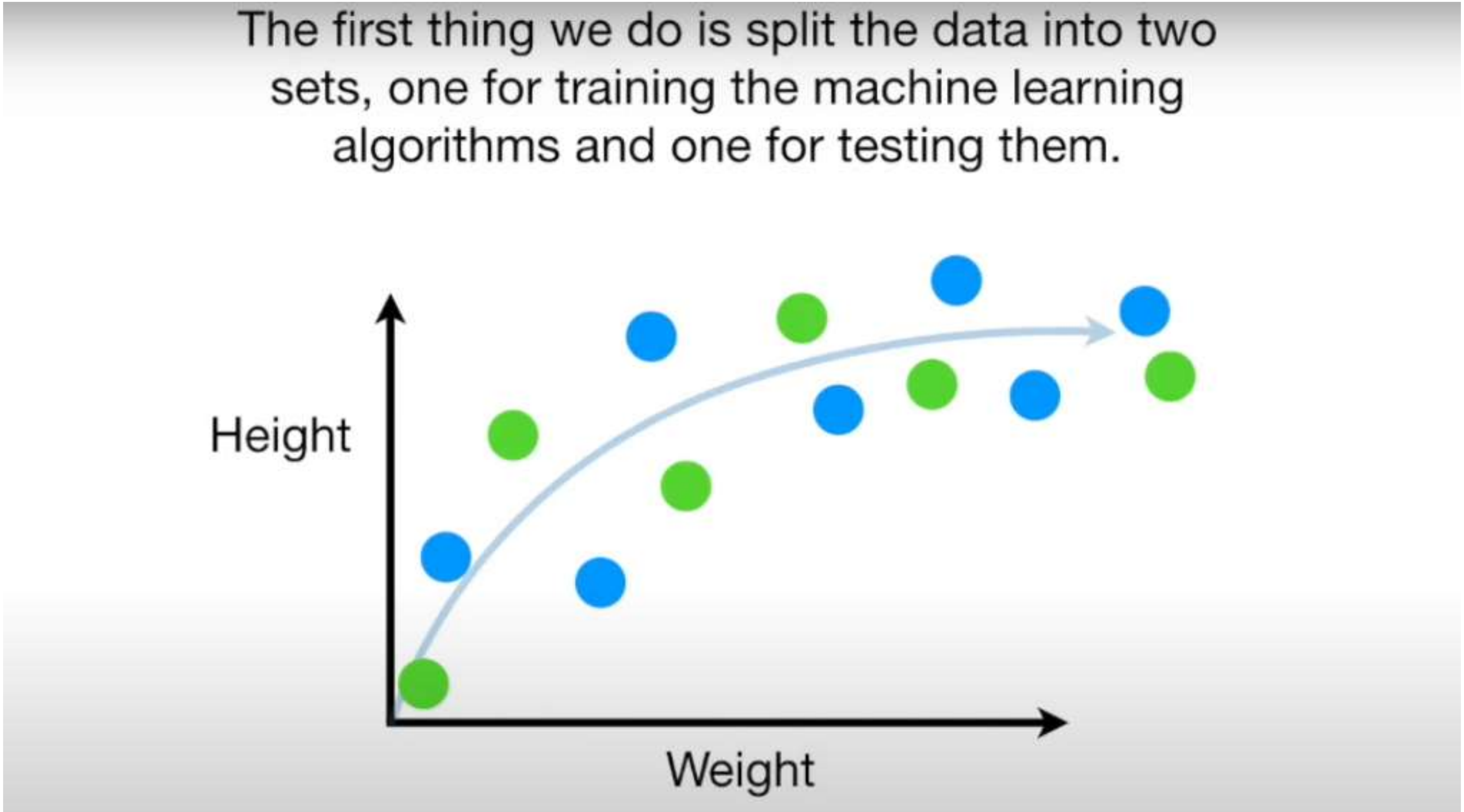
Ideally, we would know the exact mathematical formula that describes the relationship between weight and height...



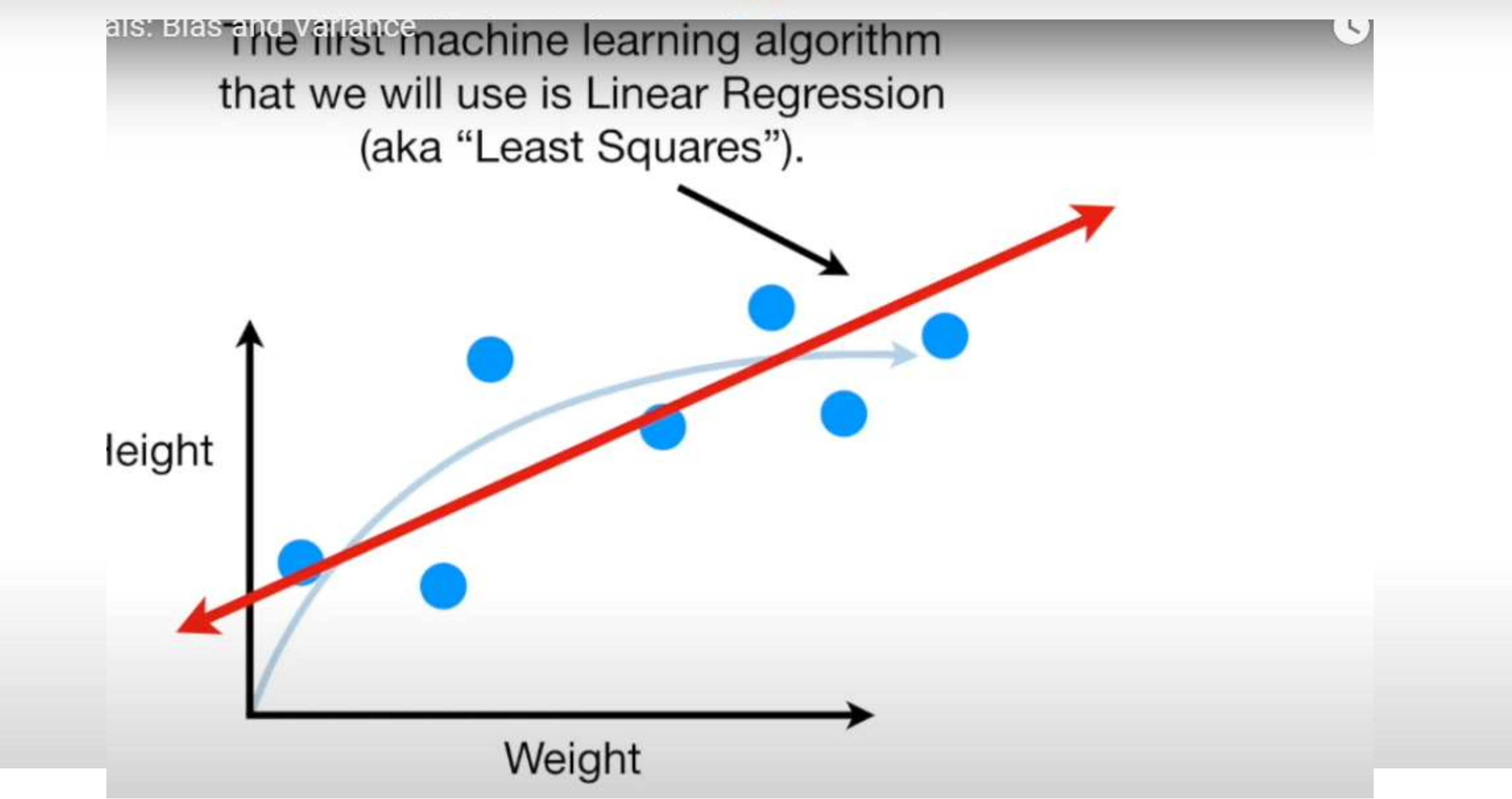


ndamentals: Bias and Variance

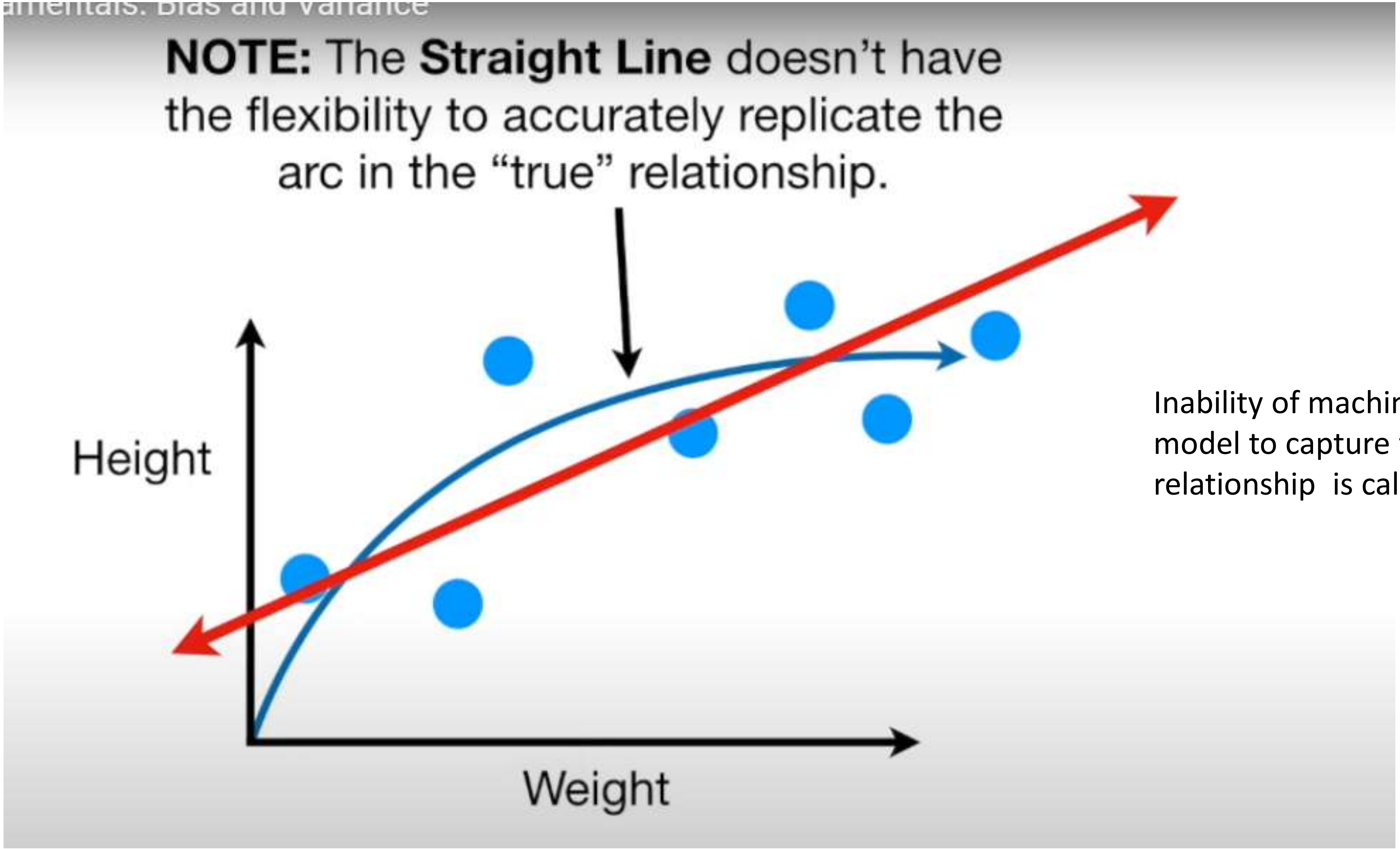
…but, in this case, we don't know the formula, so we're going to use two machine learning methods to approximate this relationship.



The first thing we do is split the data into two sets, one for training the machine learning algorithms and one for testing them.

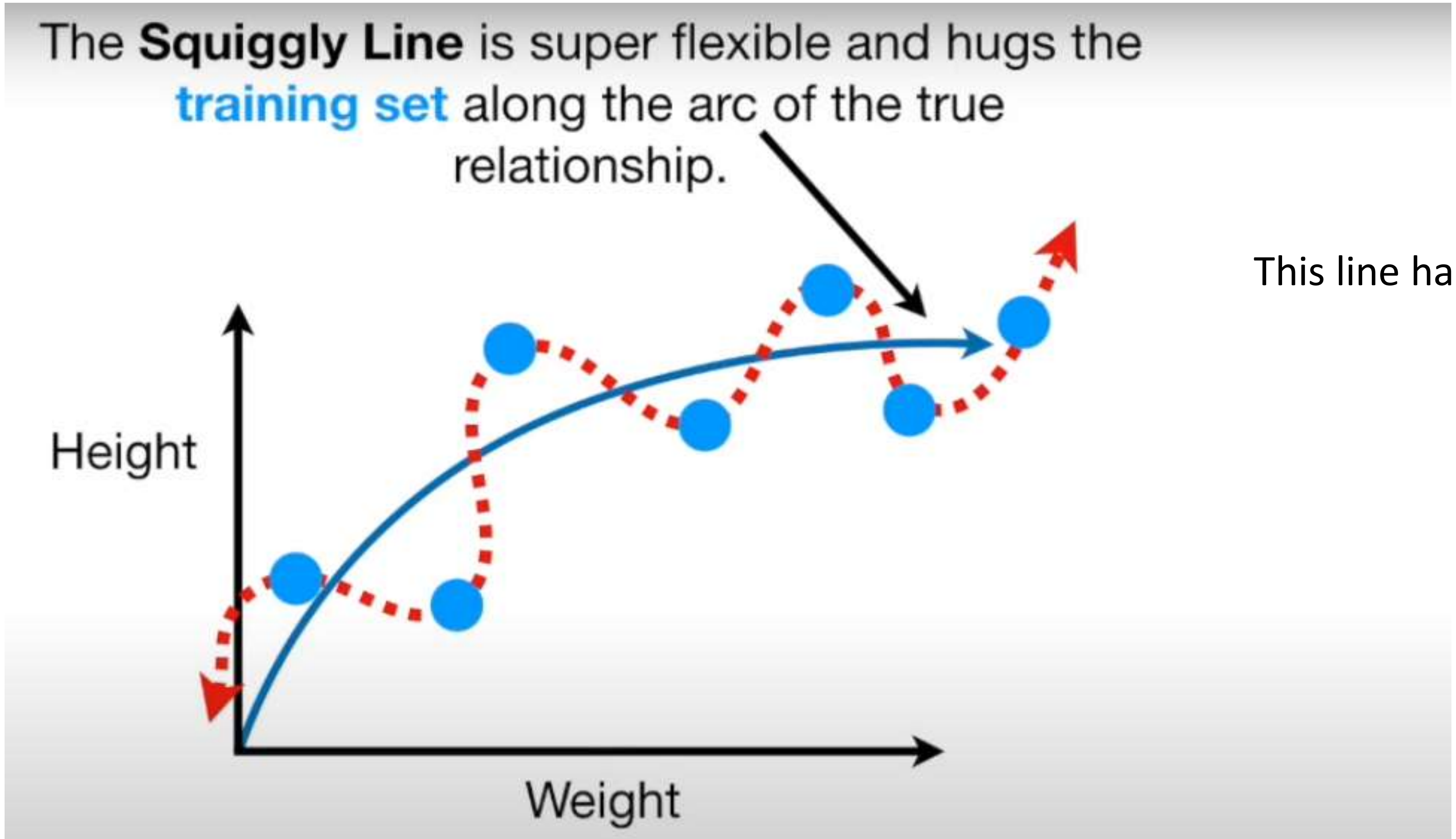


The first machine learning algorithm that we will use is Linear Regression (aka “Least Squares”).



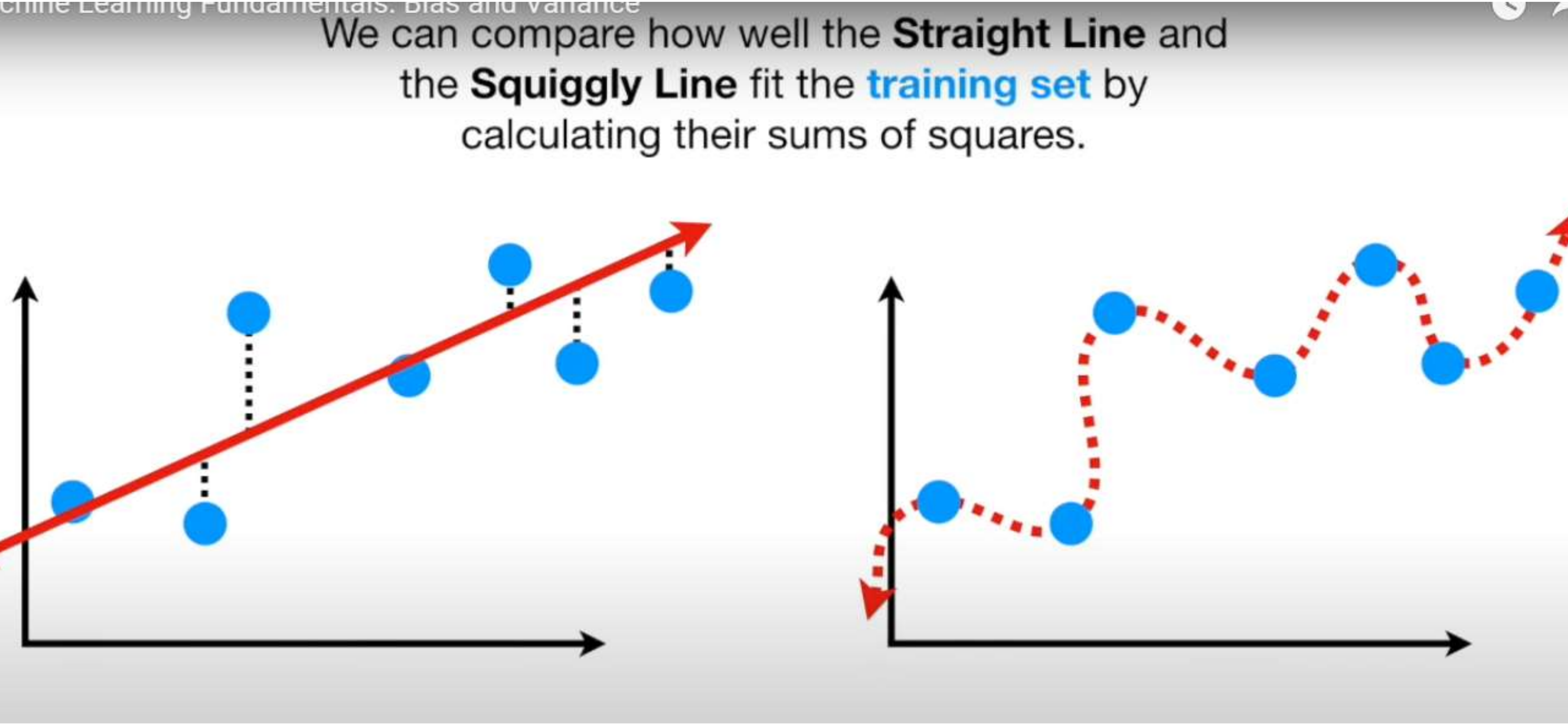
NOTE: The **Straight Line** doesn’t have the flexibility to accurately replicate the arc in the “true” relationship.

Inability of machine learning model to capture the true relationship is called Bias



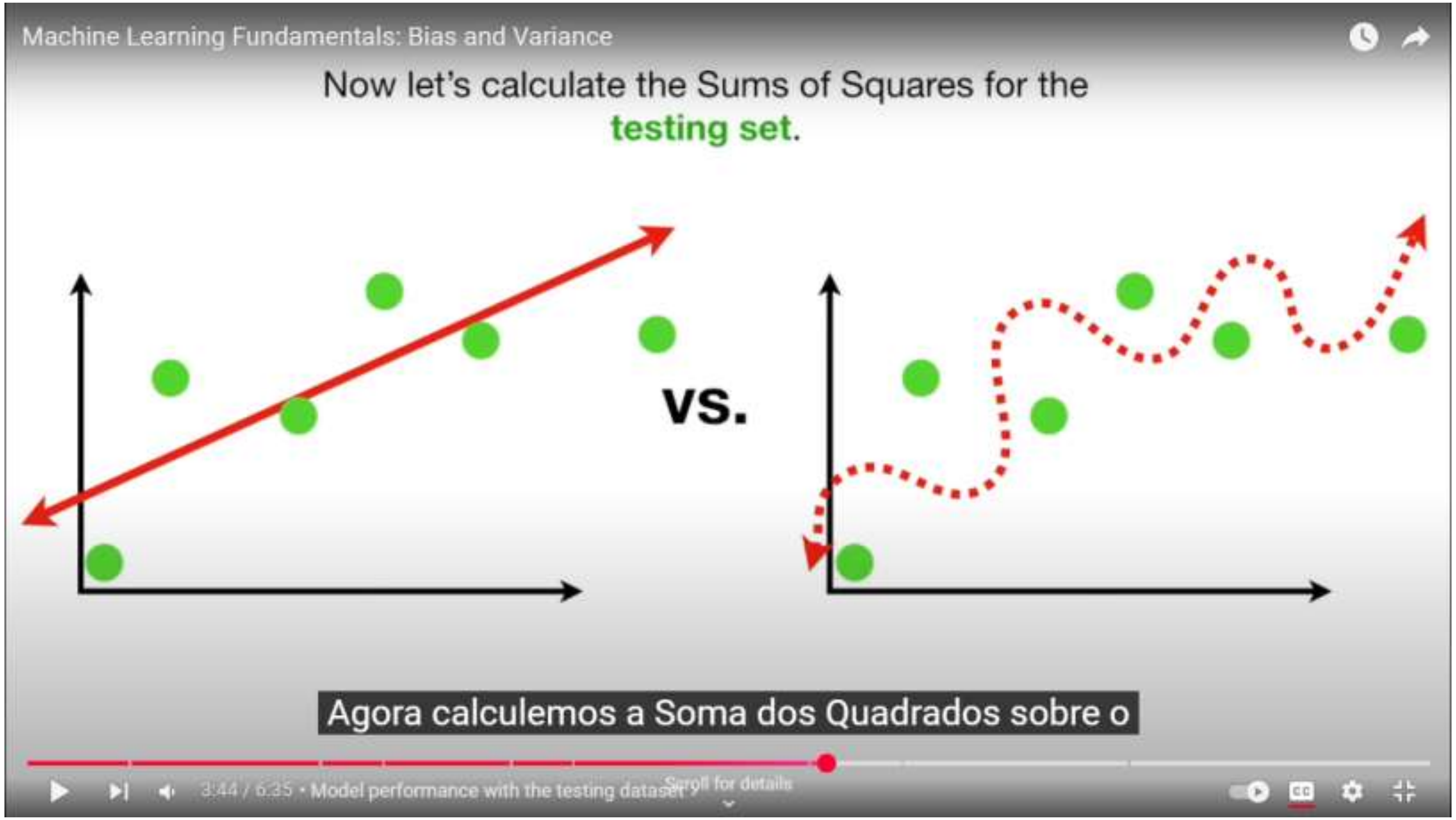
The **Squiggly Line** is super flexible and hugs the training set along the arc of the true relationship.

This line has low bias



chine Learning Fundamentals: Bias and Variance

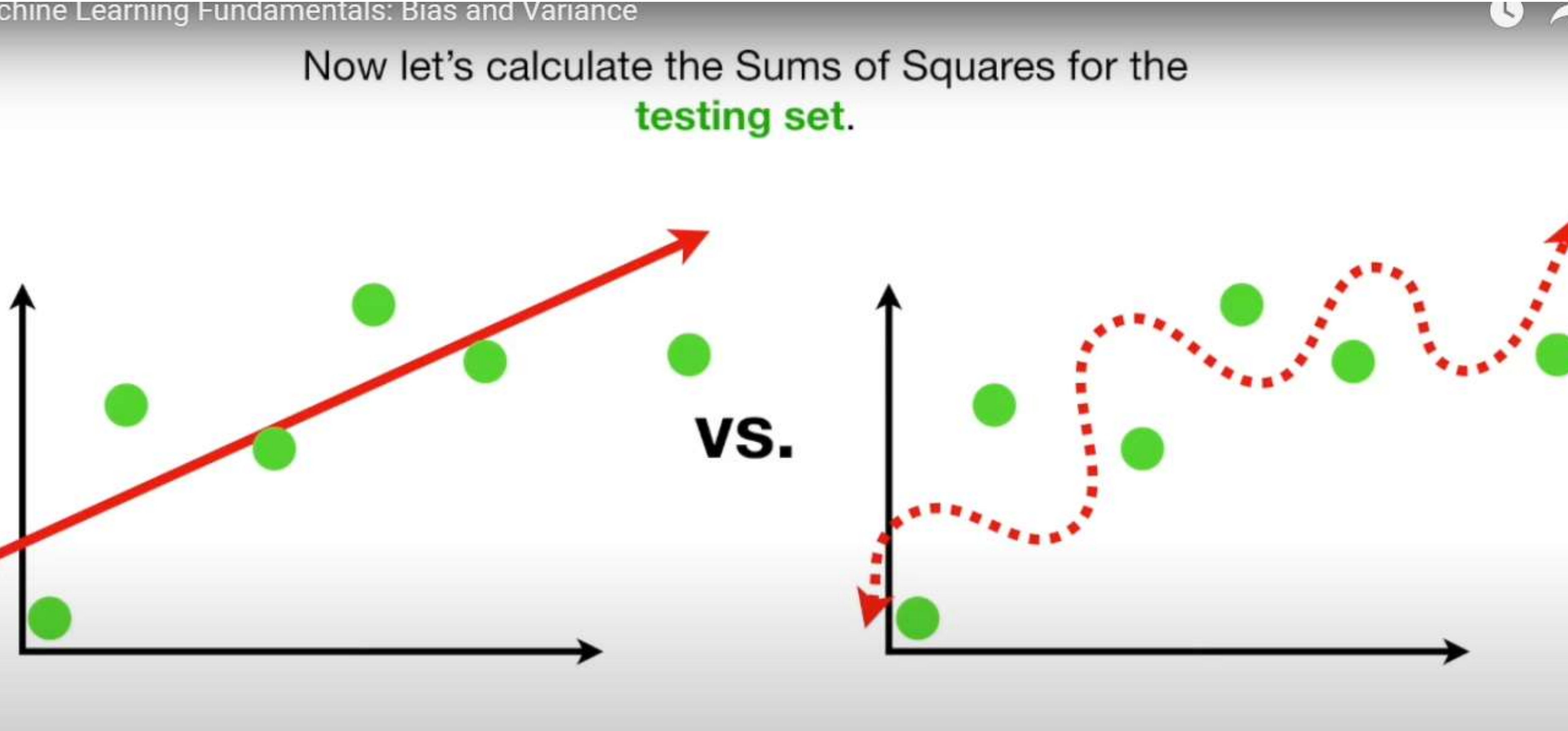
We can compare how well the **Straight Line** and the **Squiggly Line** fit the **training set** by calculating their sums of squares.



Machine Learning Fundamentals: Bias and Variance

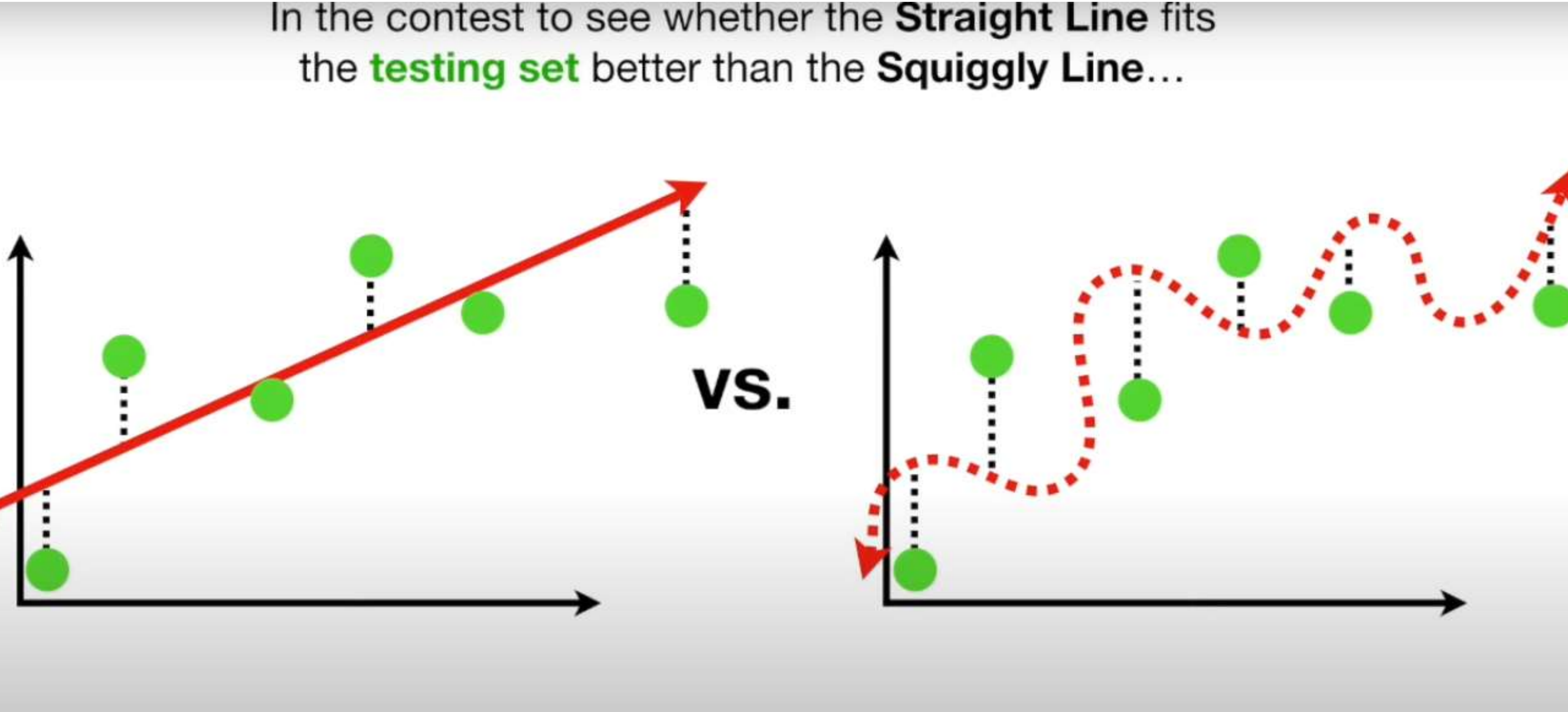
Now let's calculate the Sums of Squares for the **testing set**.

Agora calculemos a Soma dos Quadrados sobre o

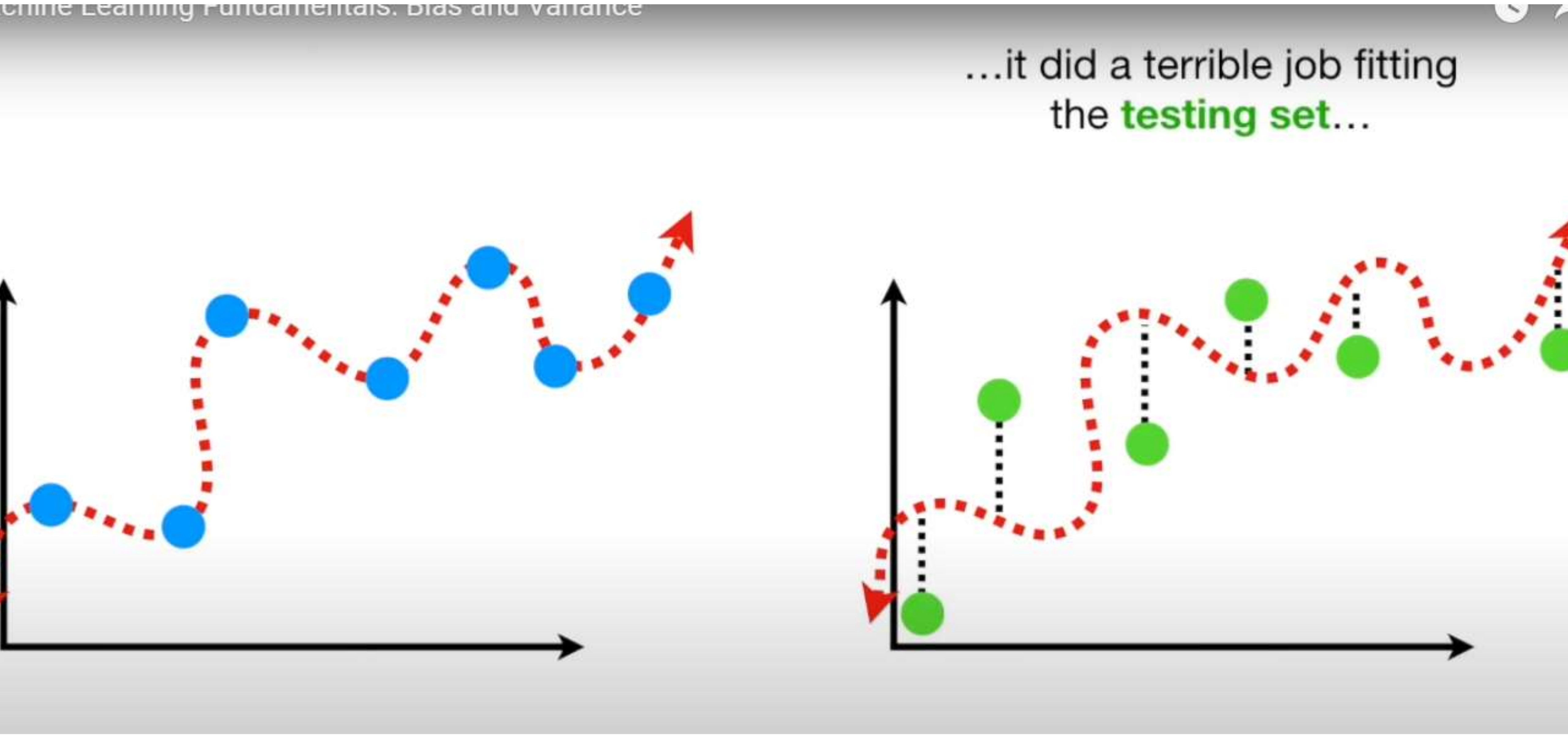


chine Learning Fundamentals: Bias and Variance

Now let's calculate the Sums of Squares for the **testing set**.



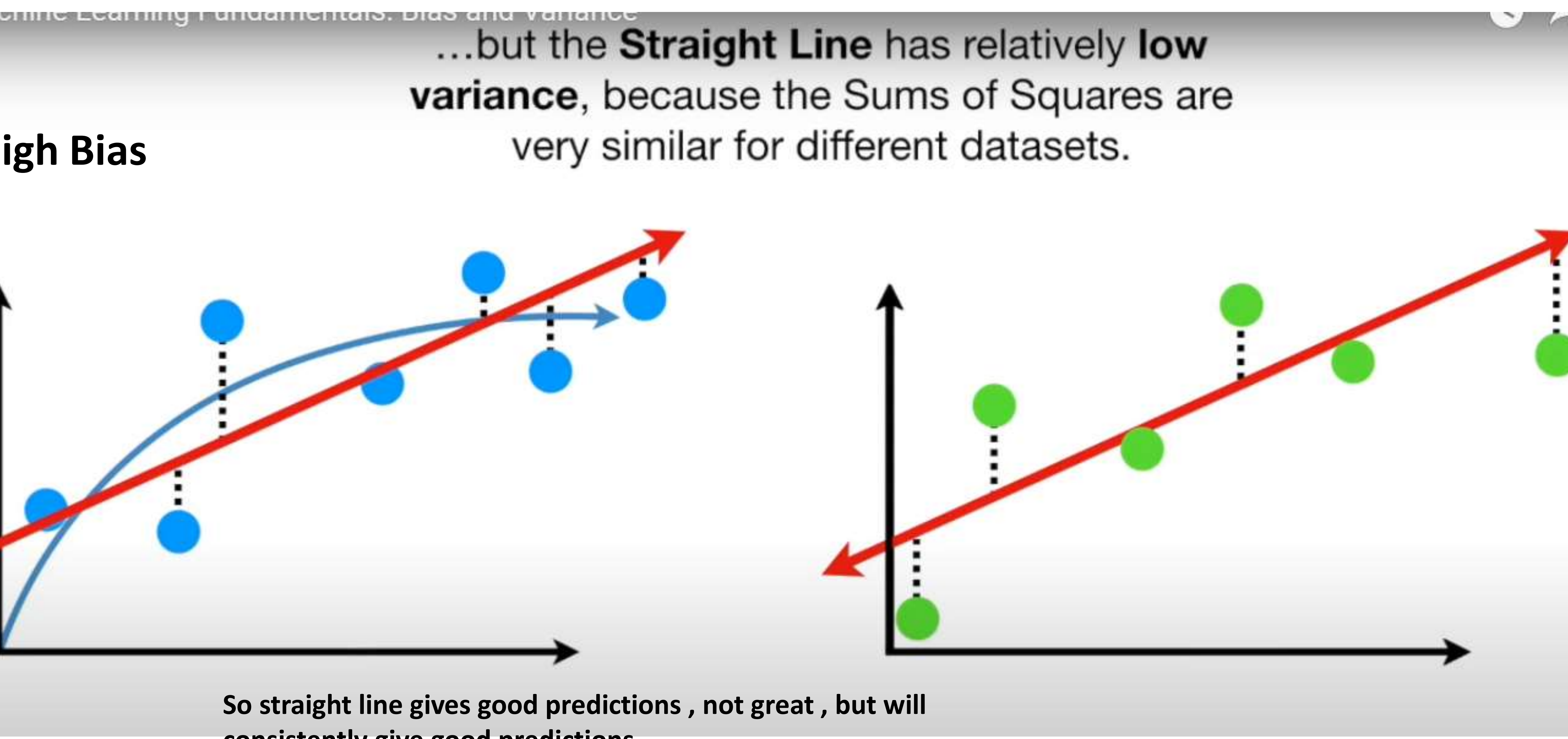
In the contest to see whether the **Straight Line** fits the **testing set** better than the **Squiggly Line**…



chine Learning Fundamentals: Bias and Variance

…it did a terrible job fitting the **testing set**…

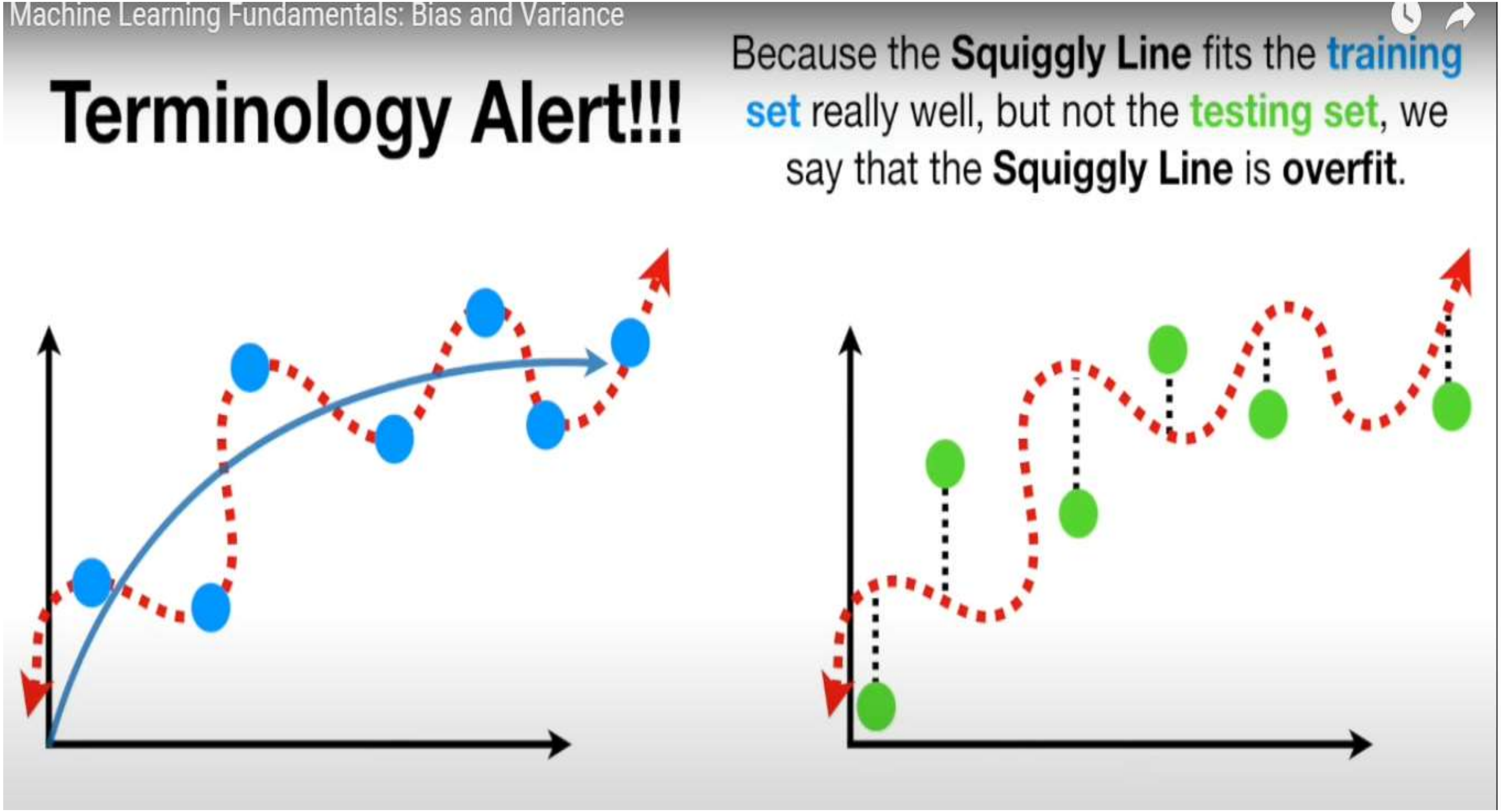
igh Bias



chine Learning Fundamentals: Bias and Variance

…but the **Straight Line** has relatively **low variance**, because the Sums of Squares are very similar for different datasets.

So straight line gives good predictions , not great , but will consistently give good predictions

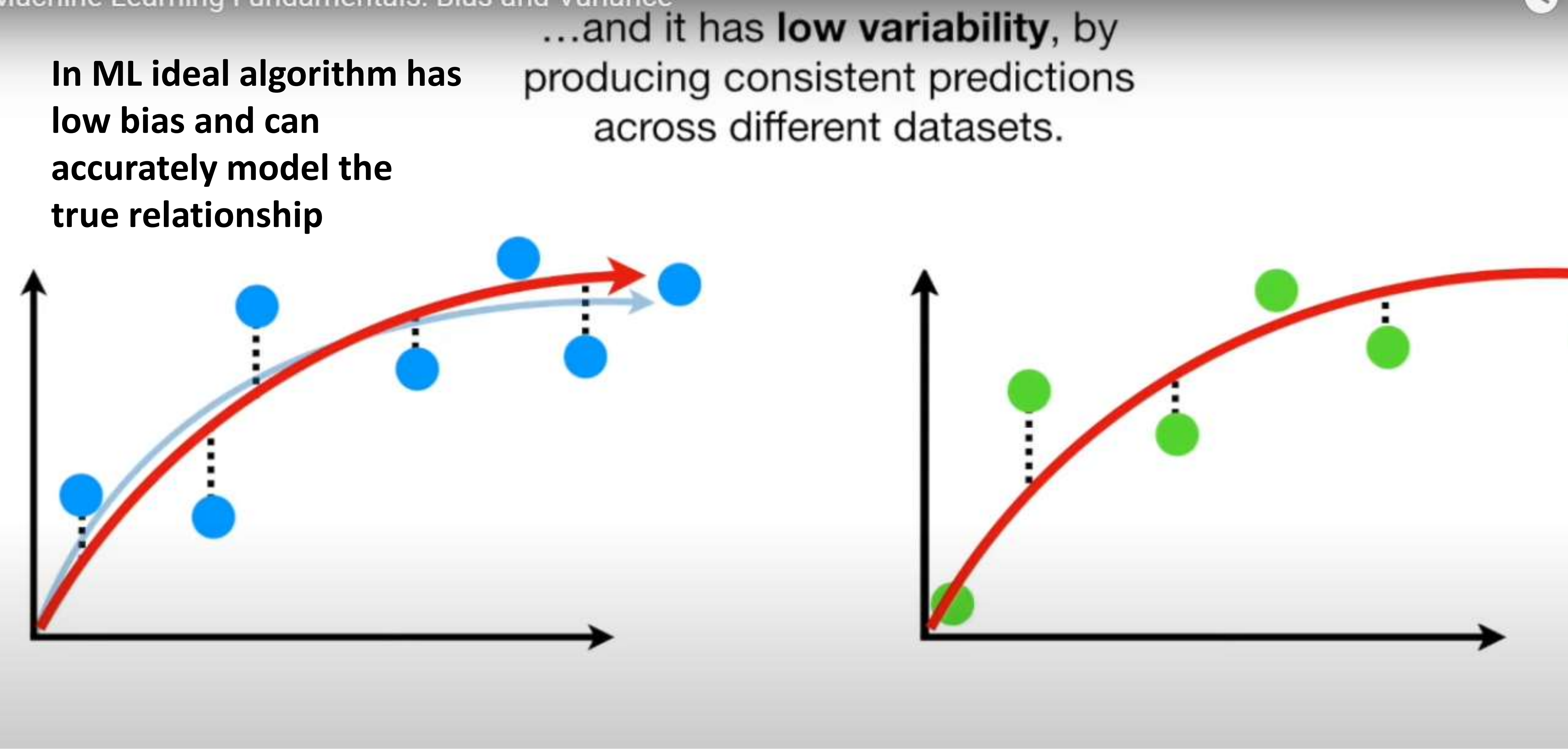


Machine Learning Fundamentals: Bias and Variance

Terminology Alert!!!

Because the **Squiggly Line** fits the **training set** really well, but not the **testing set**, we say that the **Squiggly Line** is **overfit**.

In ML ideal algorithm has low bias and can accurately model the true relationship

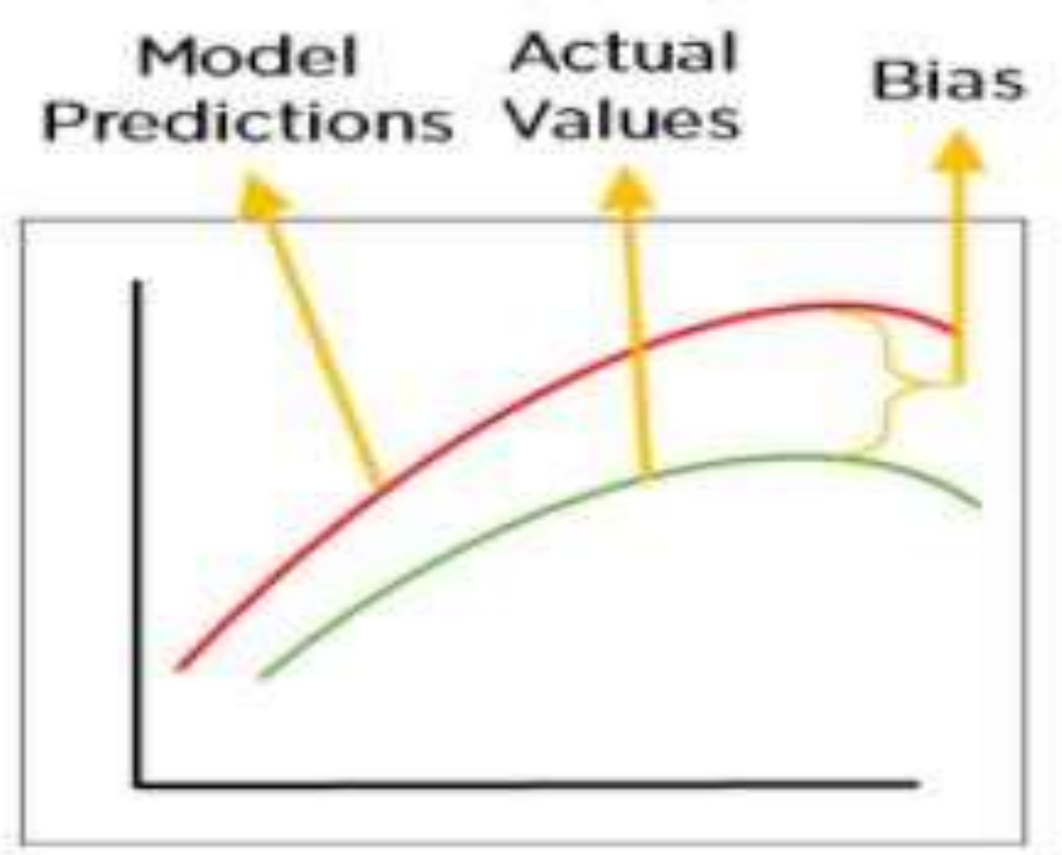


…and it has **low variability**, by producing consistent predictions across different datasets.

What is Bias?

- Bias is the difference between our actual and predicted values. Bias is the simple assumptions that our model makes about our data to be able to predict new data.
-

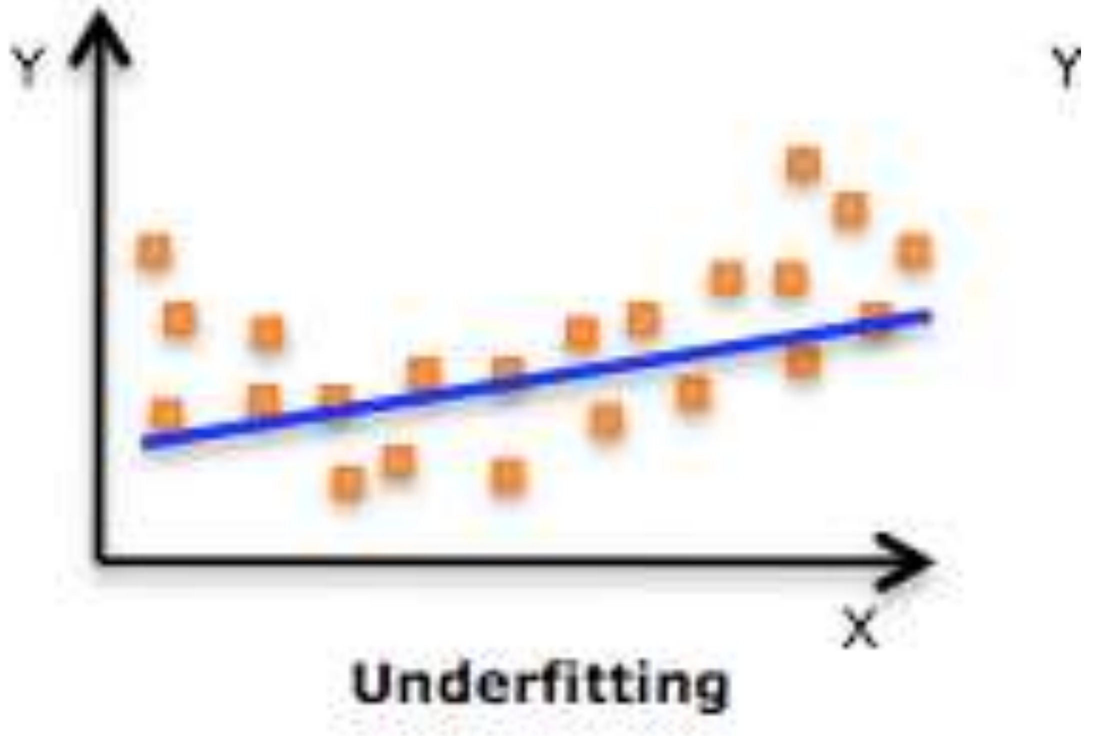
Bias is high, assumptions made by our model are too basic, the model can't capture the important features of our data.



Underfitting

- High bias means that our model hasn't captured patterns in the training data and hence cannot perform well on the testing data too.
- model cannot perform on new data and cannot be sent into production.

the model cannot find patterns in our training set and hence fails for both seen and unseen data, is called Underfitting.



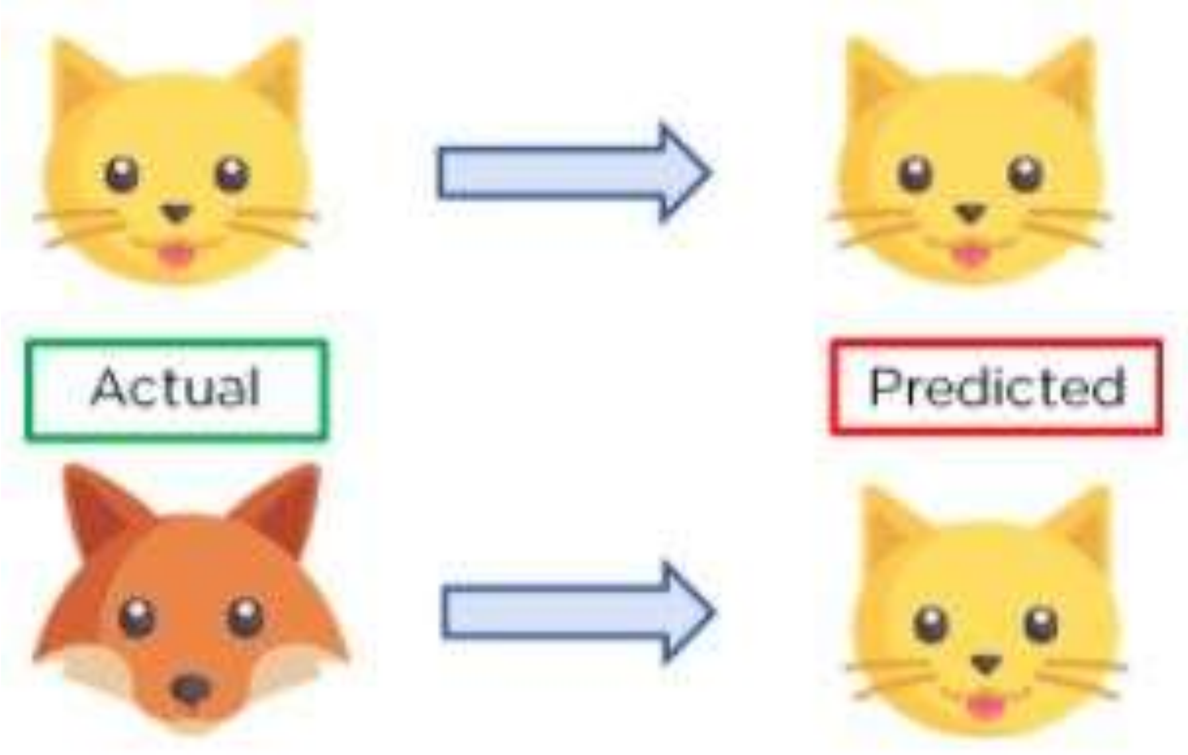
Underfitting

- In a nutshell, Underfitting – High bias and low variance
- Techniques to reduce underfitting :
- Increase model complexity
- Increase number of features, performing feature engineering
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Overfitting

- During training, it allows our model to ‘see’ the data a certain number of times to find patterns in it.
- If it does not work on the data for long enough, it will not find patterns and bias occurs.
- On the other hand, if our model is allowed to view the data too many times, it will learn very well for only that data.
- It will capture most patterns in the data, but it will also learn from the unnecessary data present, or from the noise.

Overfitting



model will perform really well on testing data and get high accuracy but will fail to perform on new, unseen data. New data may not have the exact same features and the model won’t be able to predict it very well. This is called Overfitting.

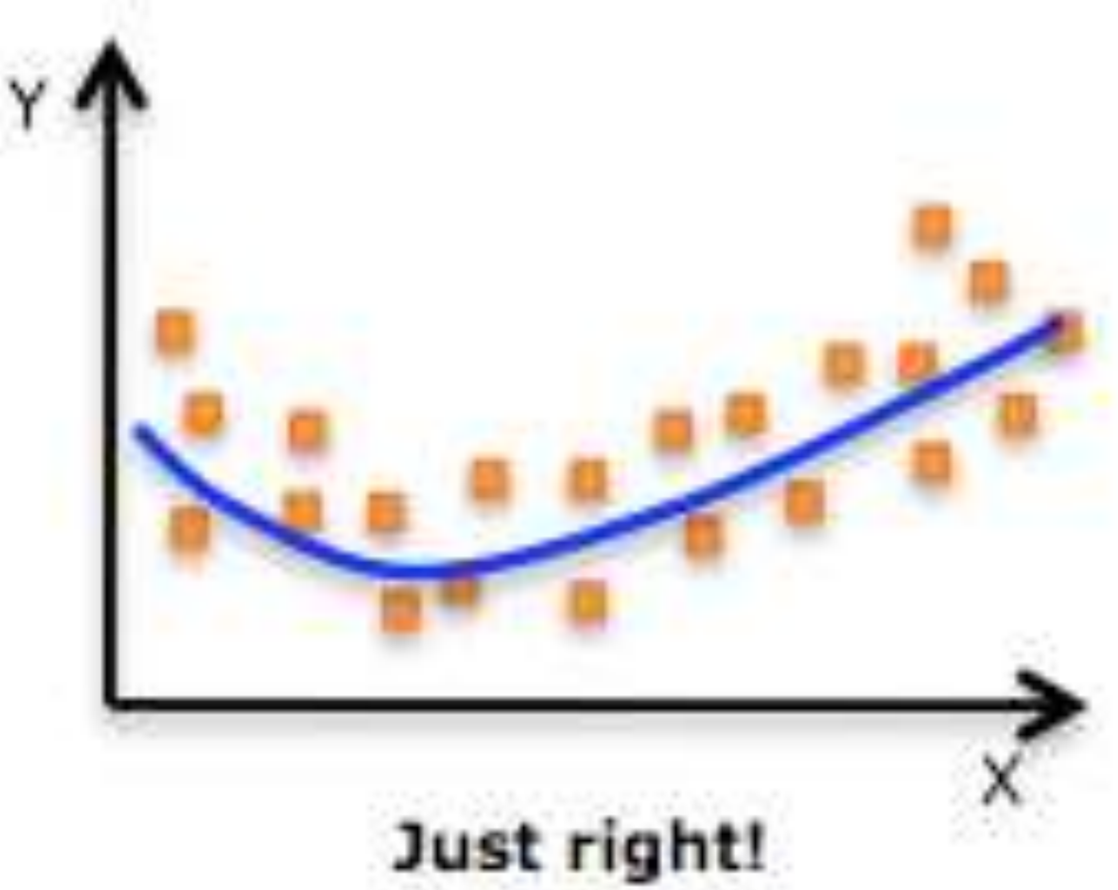
Overfitting

- In a nutshell, **Overfitting** – High variance and low bias
- Techniques to reduce overfitting :
- Increase training data.
- Reduce model complexity.
- Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
- Ridge Regularization and Lasso Regularization
- Use dropout for neural networks to tackle overfitting.

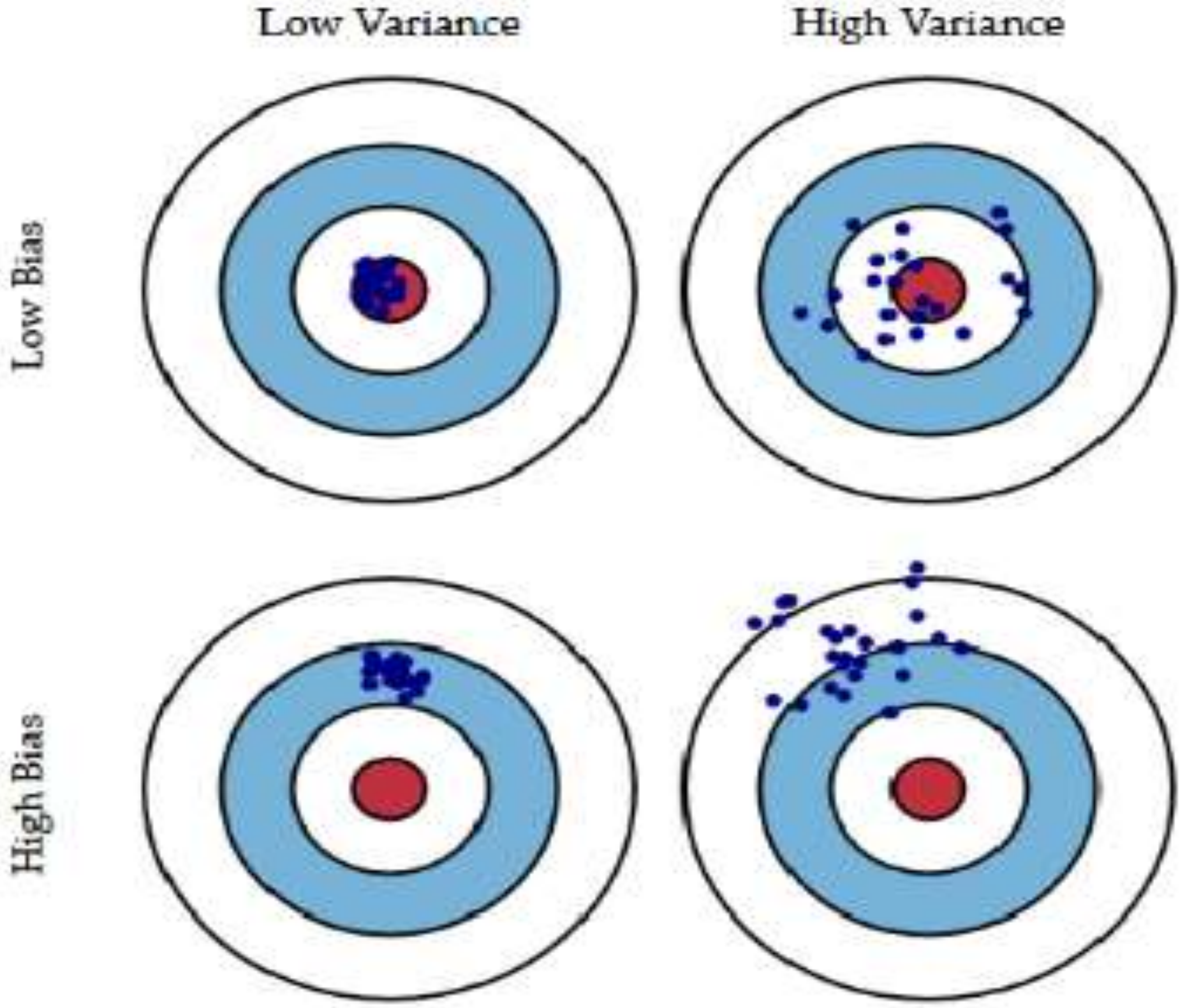
Good Fit/Right Fit

- Ideally, we want to select a model at the sweet spot between underfitting and overfitting.
- look at the performance of a machine learning algorithm over time as it is learning a training data.
- Over time, as the algorithm learns, the error for the model on the training data goes down and so does the error on the test dataset.
- The sweet spot is the point just before the error on the test dataset starts to increase where the model has good skill on both the training dataset and the unseen test dataset.

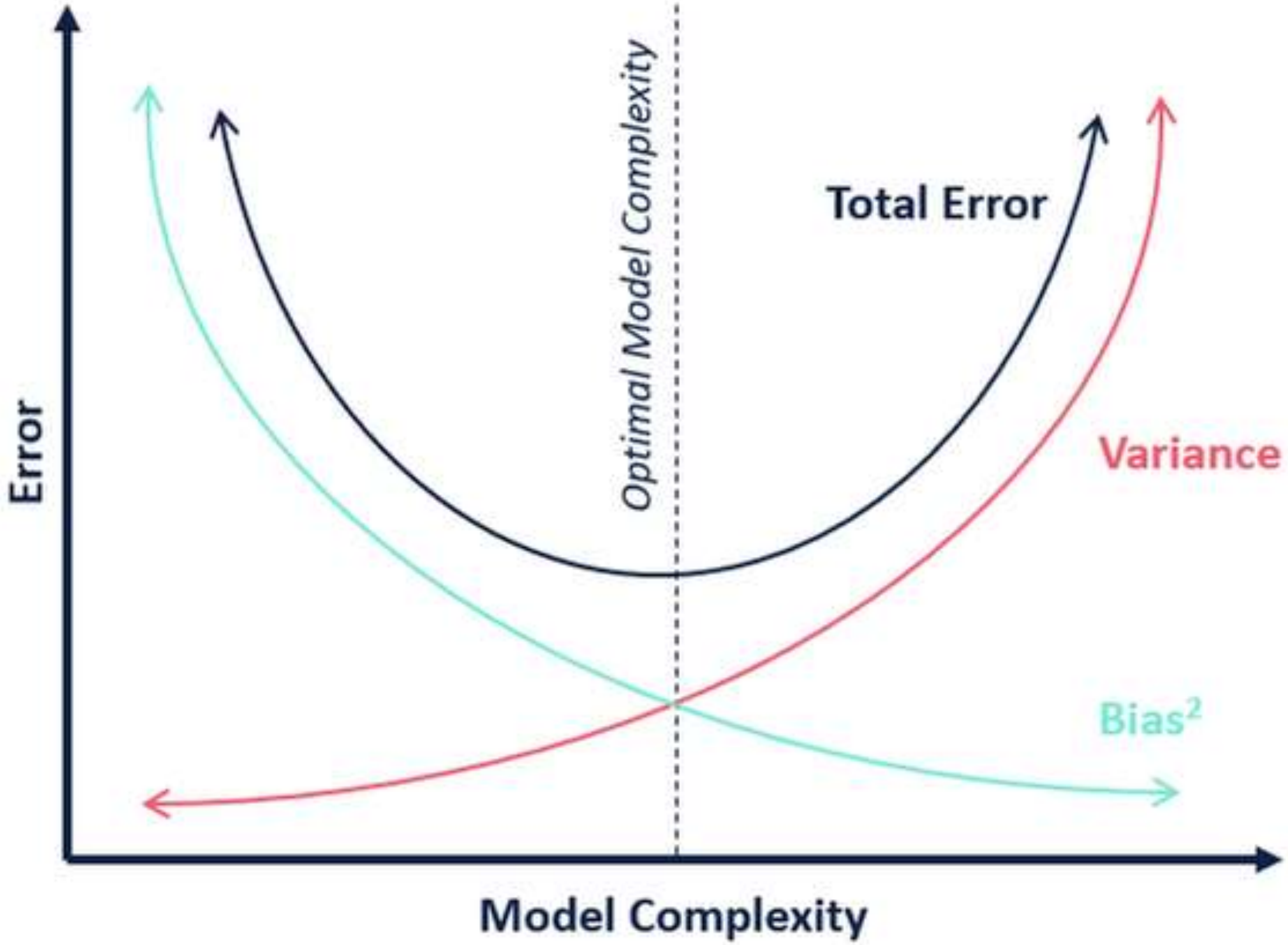
Good Fit/Right Fit



Bias-Variance Tradeoff



Bias-Variance Tradeoff



Types of ML Algorithms

- Separate out emails as spam or non-spam based on labeled data.
- Identify fraudulent transactions in financial systems. Analyze user
- Predict the amount of CO2 emissions produced by vehicles based on engine size, fuel type, and mileage.
- sentiments in social media posts or reviews.
- Predict diseases from patient medical records.
- Estimate future stock prices using historical data, trading volume, and economic indicators.
- Determine the likelihood of loan repayment.

Types of ML Algorithms

- Group customers based on purchasing behavior.
- Train agents to play and excel in games like chess or video games.
- Predict buying patterns using partially labeled customer data.
- Develop driving policies for self-driving cars.
- Discover frequently purchased product combinations.
- Teach robots to navigate through environments like mazes or factories.
- Predict monthly or yearly sales revenue based on factors like advertising spend, product price, and seasonal trends.

Types of ML Algorithms

- Train systems to recognize speech with partially labeled datasets.
- Use features such as square footage, number of bedrooms, location, and age of the house to predict its price.
- Organize web pages into categories with mixed labeled and unlabeled data.