EI320A(3) 深度學習使用 Python

Instructors

Tipajin Thaipisutikul (t.greentip@gmail.com)

Prof. Huang-Chia Shih (hcshih@Saturn.yzu.edu.tw)

Course Syllabus

Evaluation Criteria

Tasks	Percentage
In Class Hands-on	60
Project Proposal Presentation	10
Project Final Presentation	30
Bonus (In Class Participation)	5
Total	110/100

Week	Date	Content	Note	Total
1	2/26	Welcome to the course	Download & Install Anaconda Homework (1)	1
2	3/5	Crash Course of Python, Numpy, Pandas, and Matplotlib	In class hands-on (4)	5
3	3/12	Get to know about Data ML:Classification Models	In class hands-on (5)	10
4	3/19	ML:Regression Models	In class hands-on (5)	15
5	3/26	ML:Clustering /Apriori Models	In class hands-on (5)	20
6	4/2	Holiday		
7	4/9	Introduction to Deep Learning (ANN)	In class hands-on (5)	25
8	4/16	Convolutional Neural Network (CNN)	In class hands-on (5)	30
9	4/23	Convolutional Neural Network (CNN)	In class hands-on (5)	35
10	4/30	Recurrent Neural Network (RNN)	In class hands-on (5)	40
11	5/7	Recurrent Neural Network (RNN)	In class hands-on (5)	45
12	5/14	Project Proposal Presentation	Proposal Presentation (10)	55
13	5/21	Time series with DNN, CNN, RNN	In class hands-on (5)	60
14	5/28	Attention Neural Network	In class hands-on (5)	65
15	6/4	Generative Adversarial Network (GAN)	In class hands-on (5)	70
16	6/11	Reinforcement Learning (RL)	In class hands-on (5)	75
17	6/18	Final Project Presentation	Final Presentation (30)	105

Bonus: 5 For class participation.

Preface

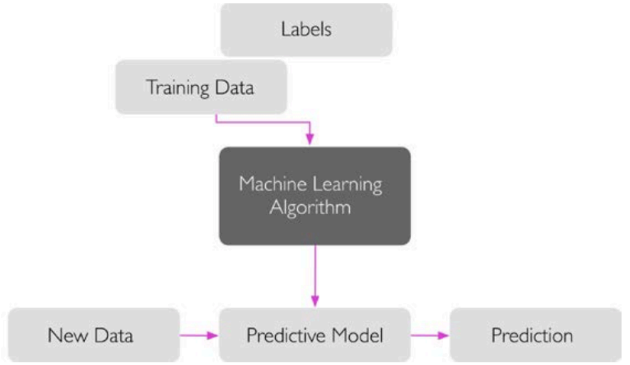
ARTIFICIAL INTELLIGENCE

Science thae enables computers to mimic human imtelligence.

Subfields: Machine Learning, robotics, and computer vision

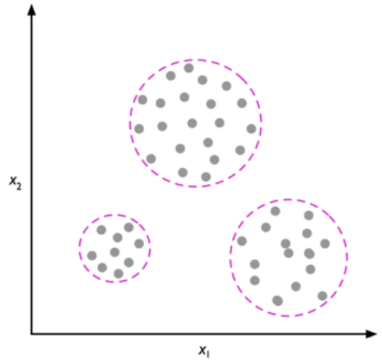
MACHINE LEARNING

Subset of AI that enable maches to improve at tasks with experience

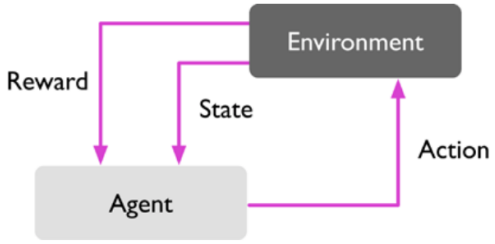


The three different types of machine learning

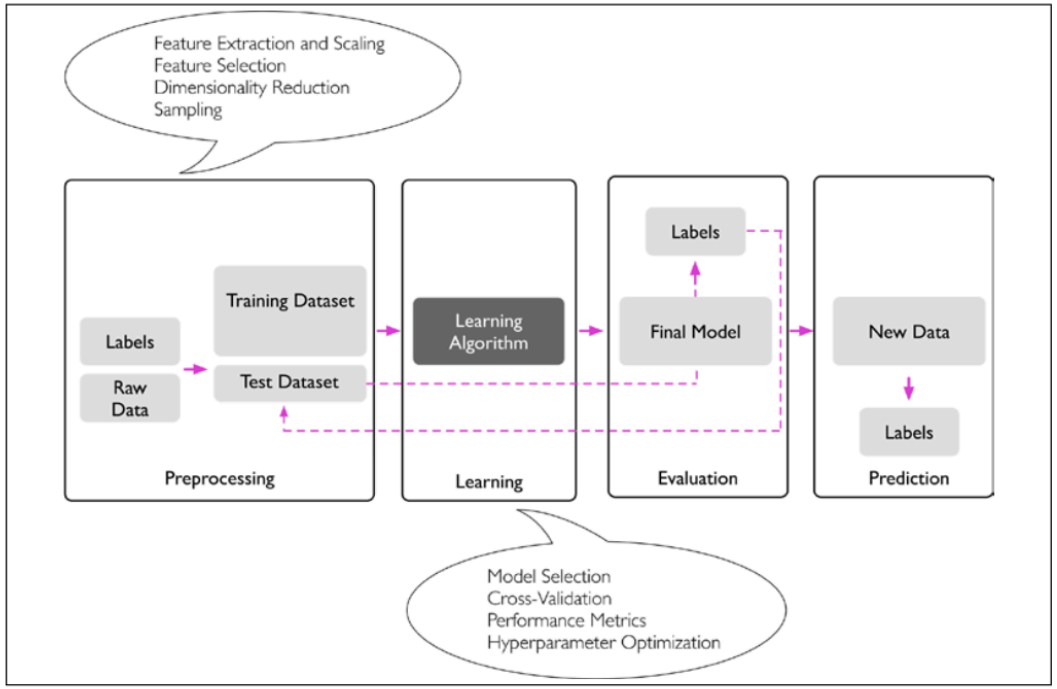
Supervised Learning	> Labeled data > Direct feedback > Predict outcome/future
Unsupervised Learning	> No labels > No feedback > Find hidden structure in data
Reinforcement Learning	> Decision process > Reward system > Learn series of actions



- Machine learning uses algorithms to parse data, learn from that data, and make informed decisions based on what it has learned
- Deep learning structures algorithms in layers to create an artificial "neural network" that can learn and make intelligent decisions on its own
- Deep learning is a subfield of machine learning. While both fall under the broad category of artificial intelligence, deep learning is usually what's behind the most human-like artificial intelligence



A Roadmap for building ML Systems



A predictive modeling machine learning project can be broken down into **6 top-level tasks:**

1. **Define Problem**: Investigate and characterize the problem in order to better understand the goals of the project.
2. **Analyze Data**: Use **descriptive statistics** and **visualization** to better understand the data you have available.
3. **Prepare Data**: Use **data transforms** in order to better expose the structure of the prediction problem to modeling algorithms.
4. **Evaluate Algorithms:** Design a test harness to evaluate a number of standard algorithms on the data and select the top few to investigate further.
5. **Improve Results:** Use algorithm tuning and ensemble methods to get the most out of well-performing algorithms on your data.
6. **Present Results:** Finalize the model, make predictions and present results.

Classification Methods Continue…

	ML Lesson
	Lesson 1: Python Ecosystem for Machine Learning. Lesson 2: Python and SciPy Crash Course. (Numpy, Pandas, Matplotlib)
Define Problem & Analyze Data	Lesson 3: Understand Data With Descriptive Statistics. Lesson 4: Understand Data With Visualization.
Prepare Data	Lesson 5: Pre-Process Data.
Evaluate Algorithms	Lesson 6: Resampling Methods. Lesson 7: Algorithm Evaluation Metrics. Lesson 8: Spot-Check Classification Algorithms. Lesson 9: Spot-Check Regression Algorithms. Lesson 10: Model Selection.
Improve Results	Lesson 11: Ensemble Methods. Lesson 12: Algorithm Parameter Tuning.
Present Results	Lesson 13: Model Finalization.

1. Logistic Regression Classification

Linear Regression:

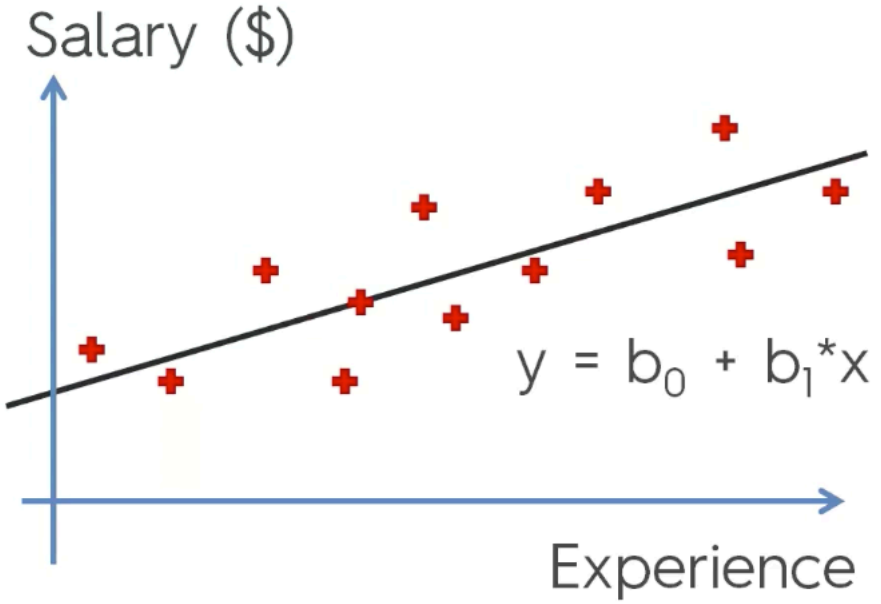
- **Simple:**

$$y = b_0 + b_1 * x$$

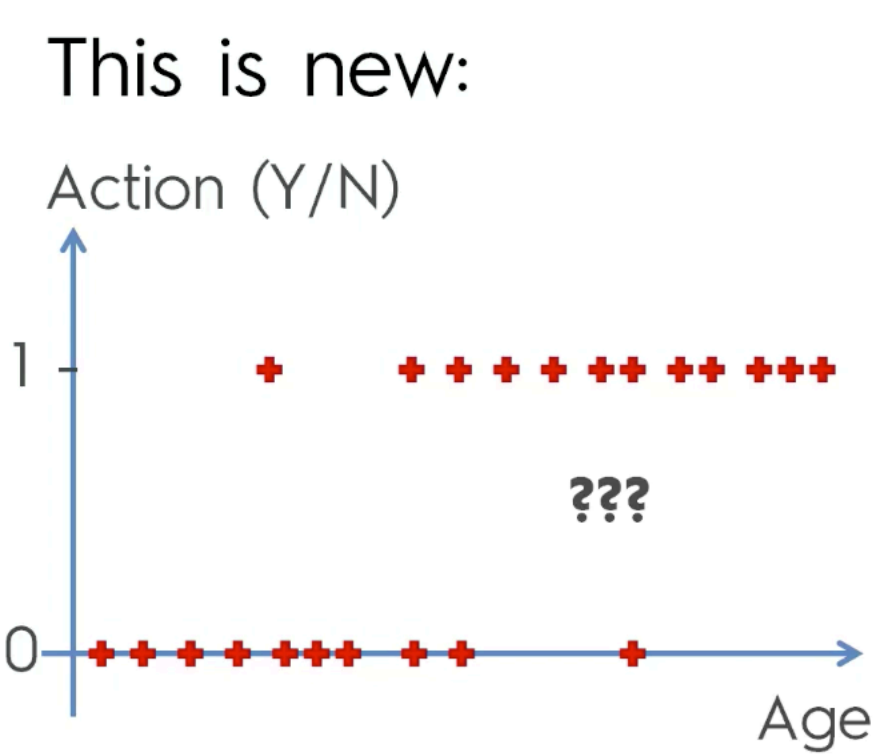
- **Multiple:**

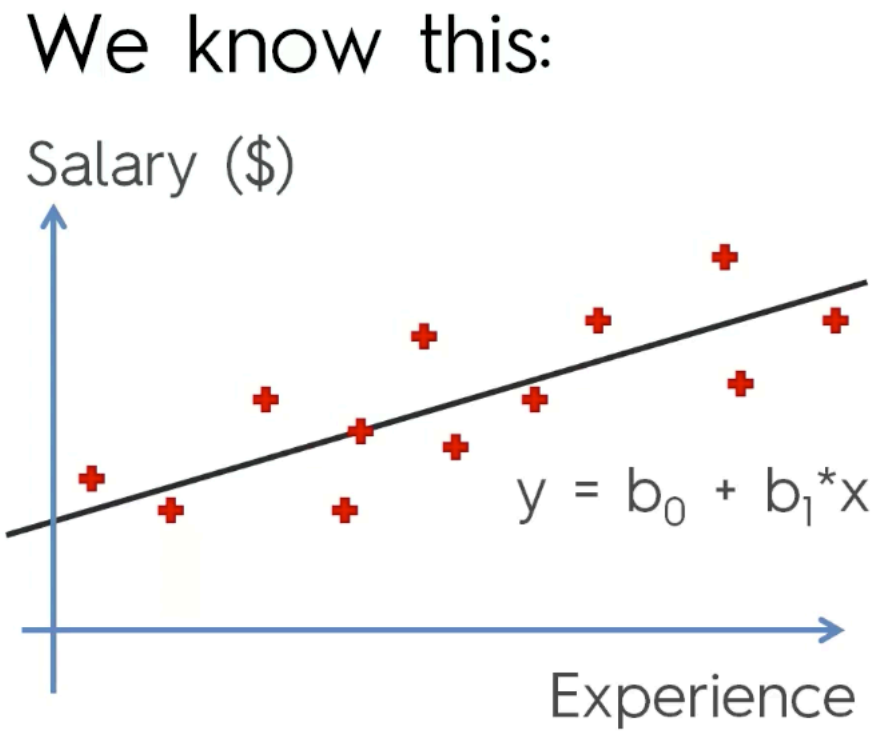
$$y = b_0 + b_1 * x_1 + \ldots + b_n * x_n$$

We know this:

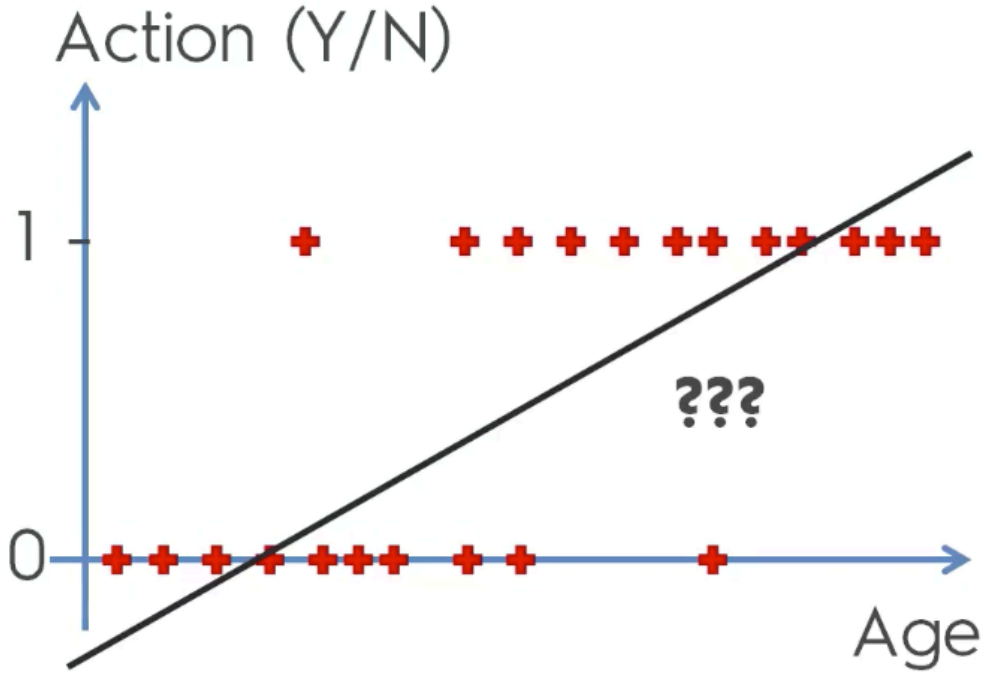


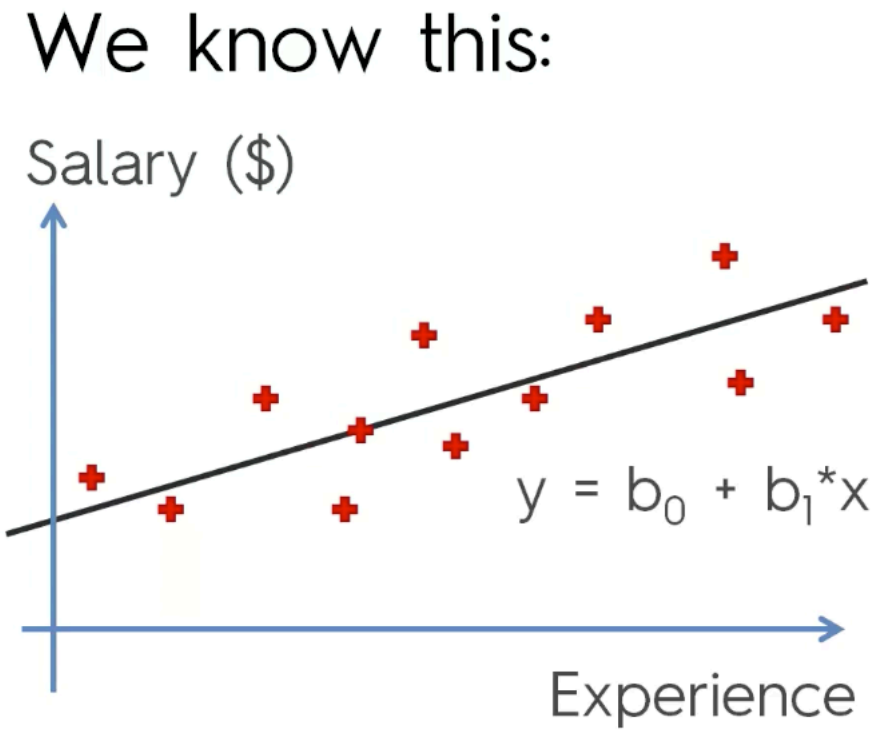
1. Logistic Regression Classification



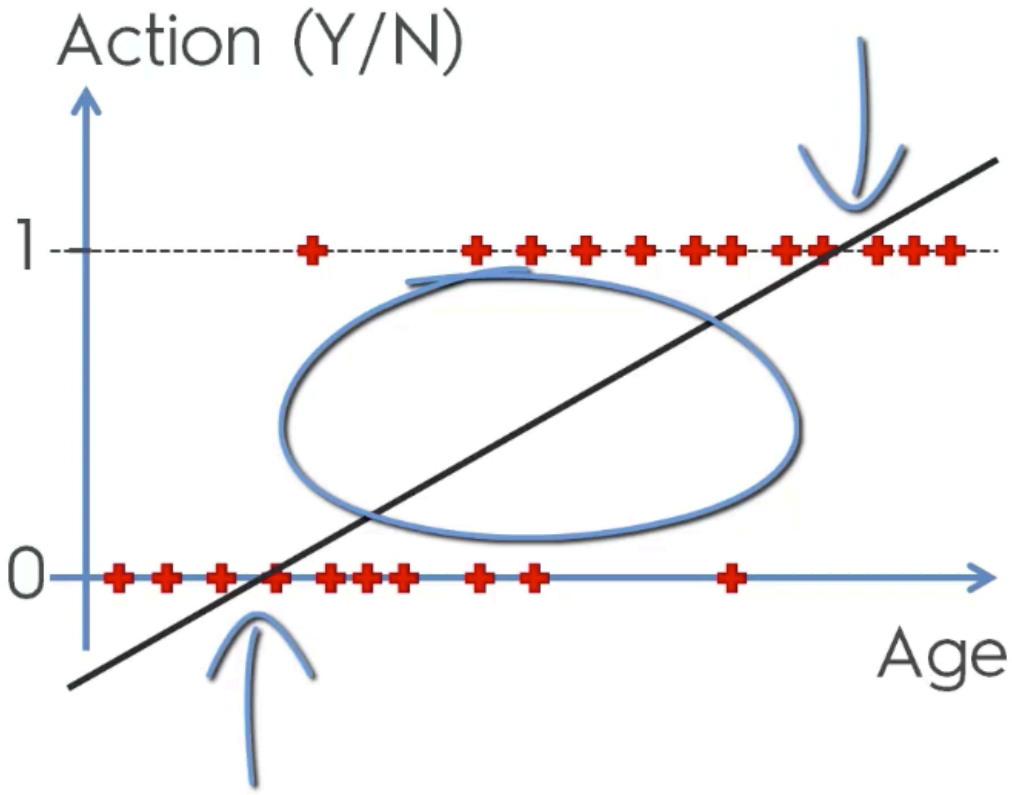


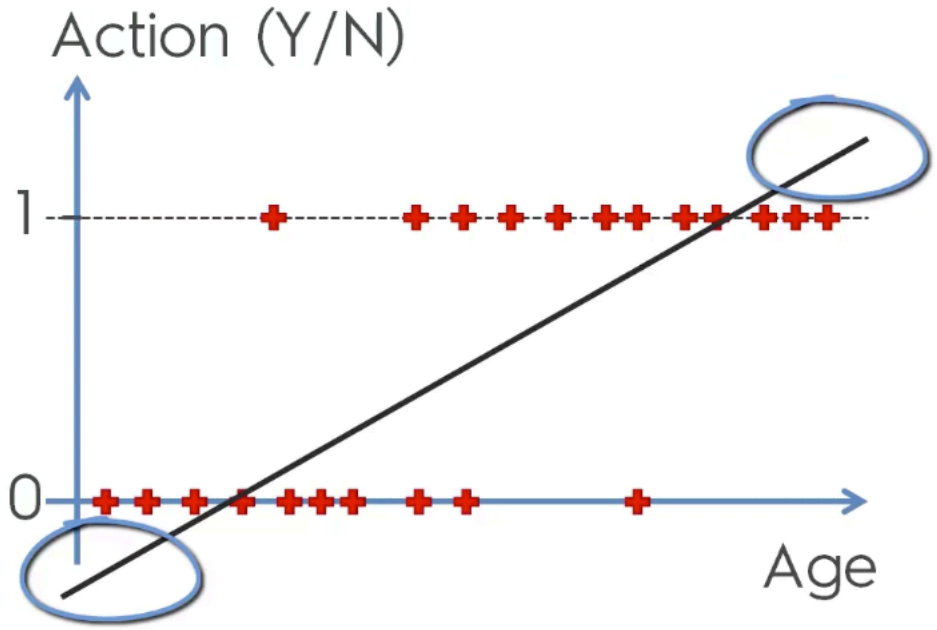
1. Logistic Regression Classification



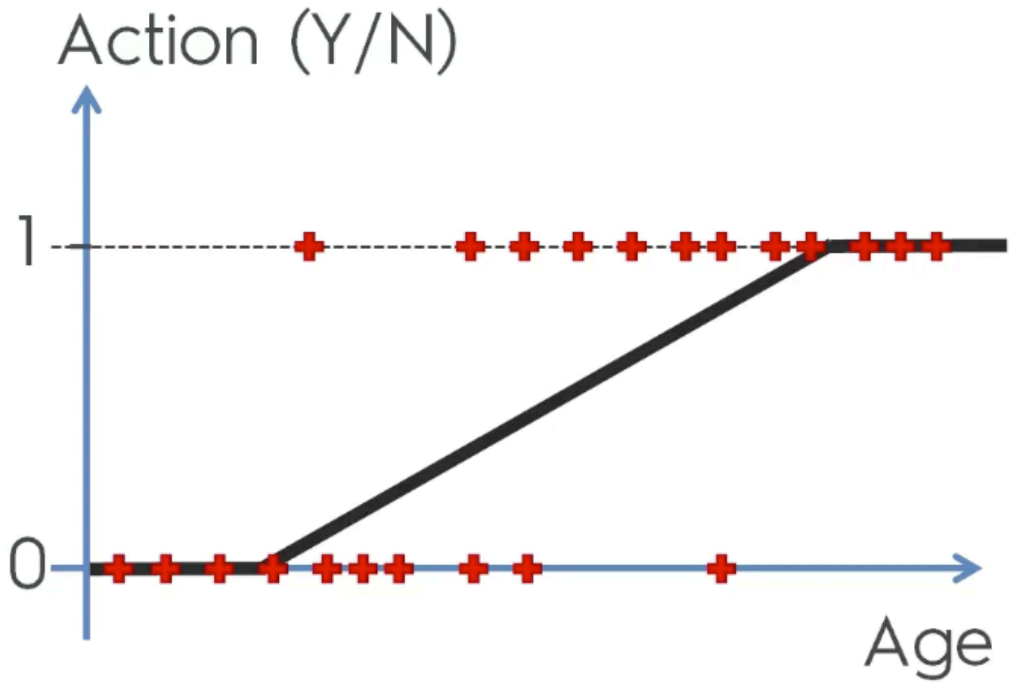


1. Logistic Regression Classification





1. Logistic Regression Classification



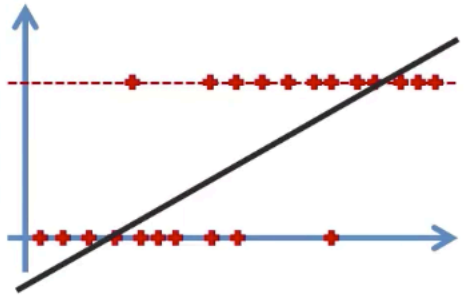
1. Logistic Regression Classification

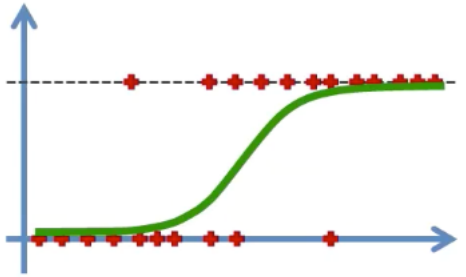
$$y = b_0 + b_1 * x$$

Sigmoid Function

$$p = \frac{1}{1 + e^{-y}}$$

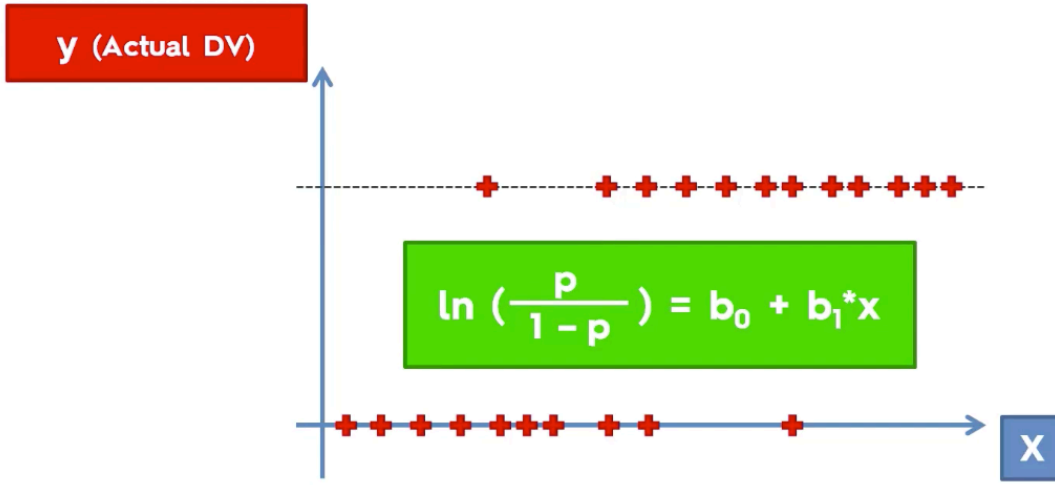
$$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 * x$$

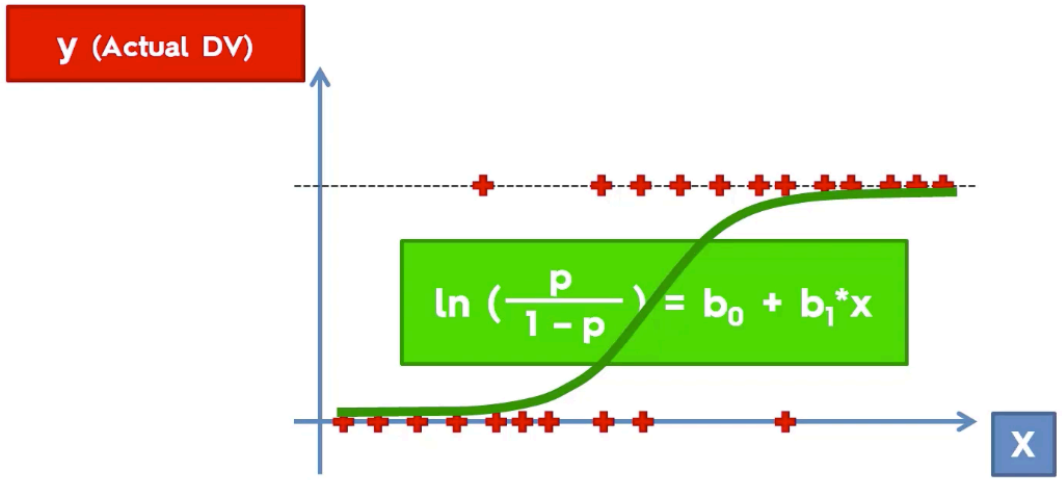




1. Logistic Regression Classification

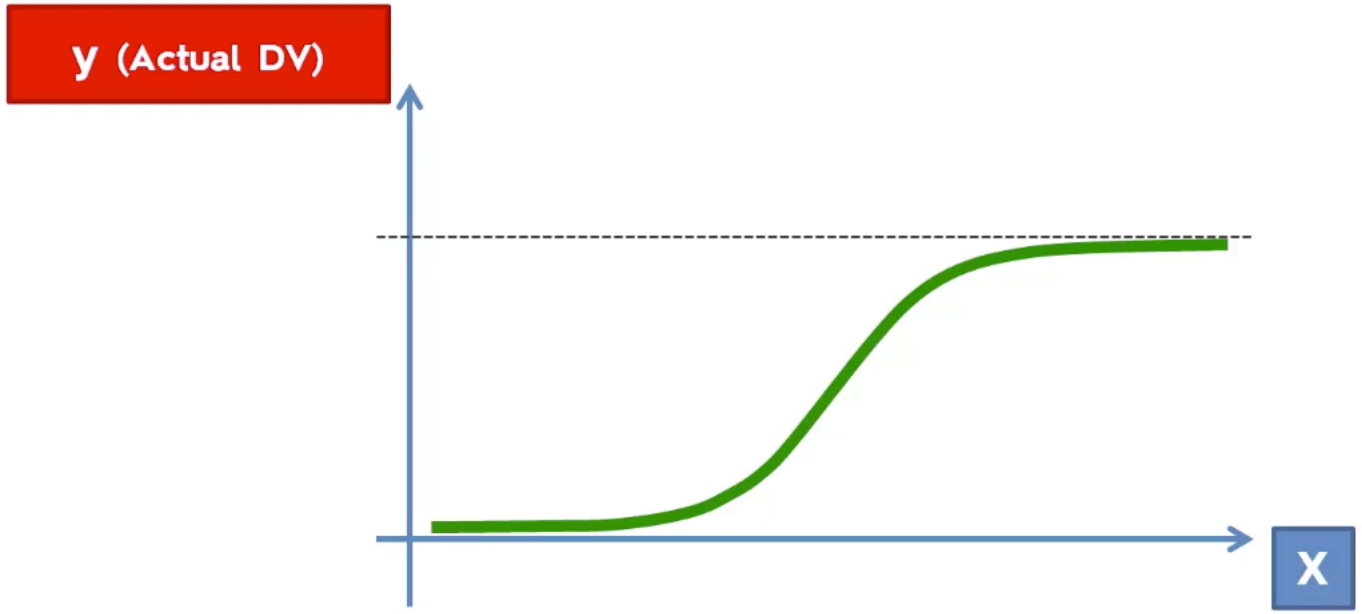
DV = Dependent Variable



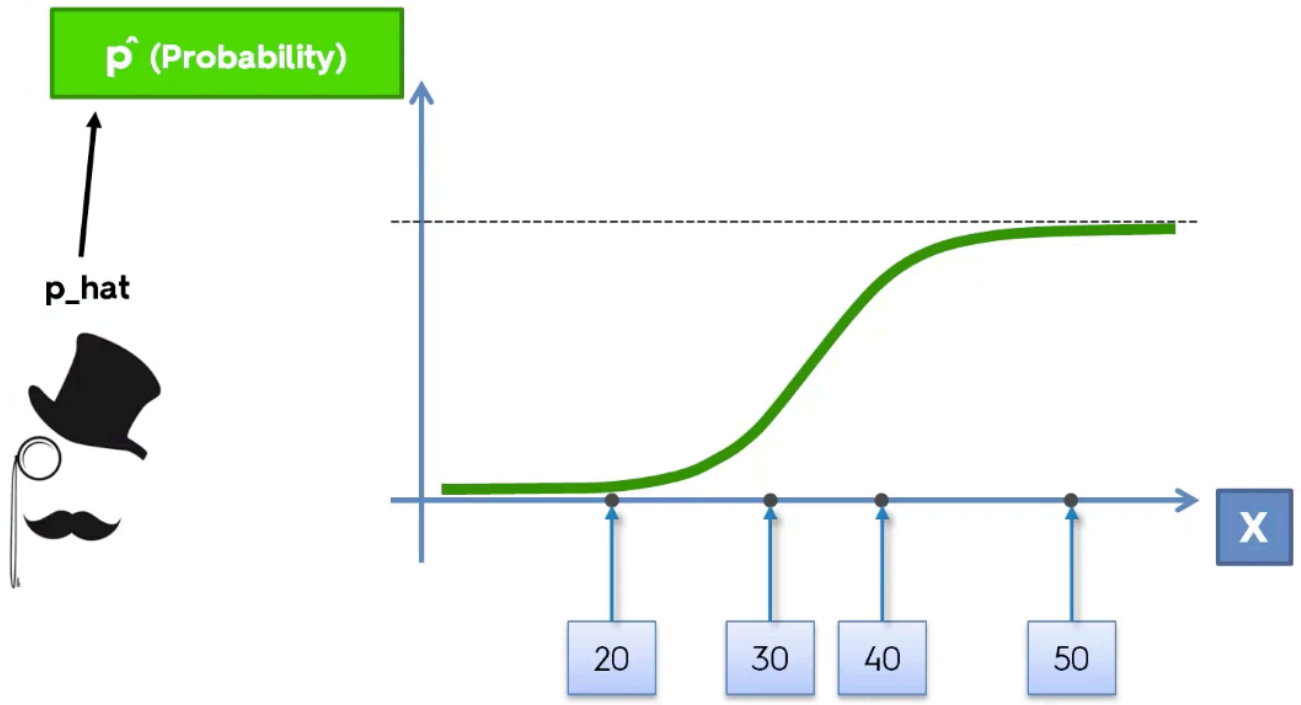


1. Logistic Regression Classification

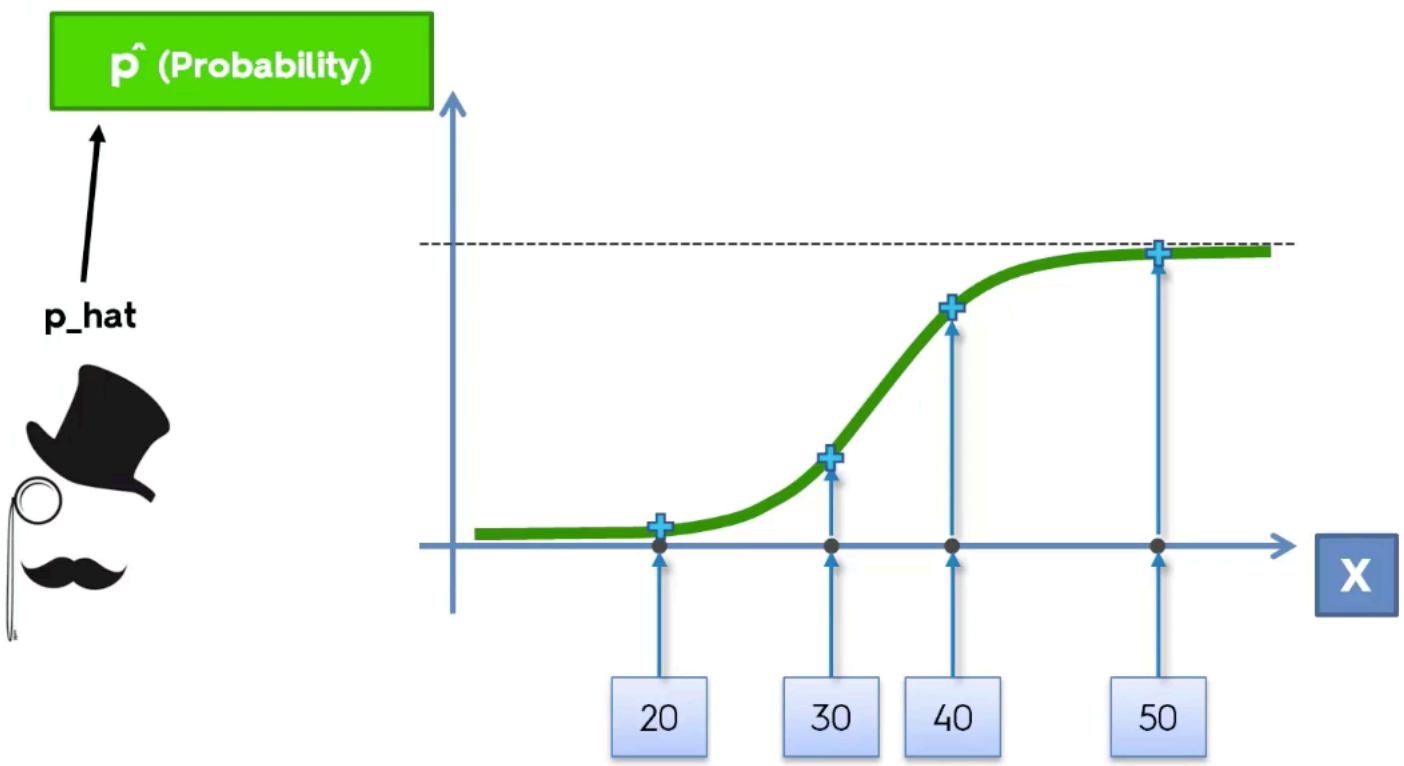
Get the pre-trained model After Training



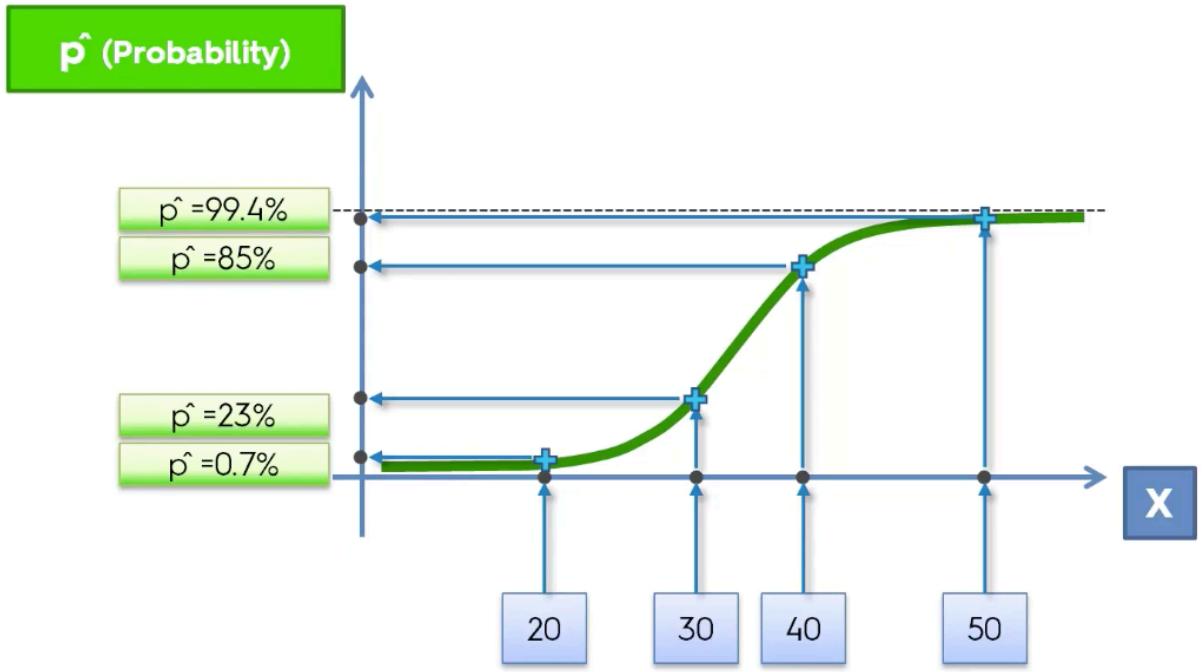
1. Logistic Regression Classification

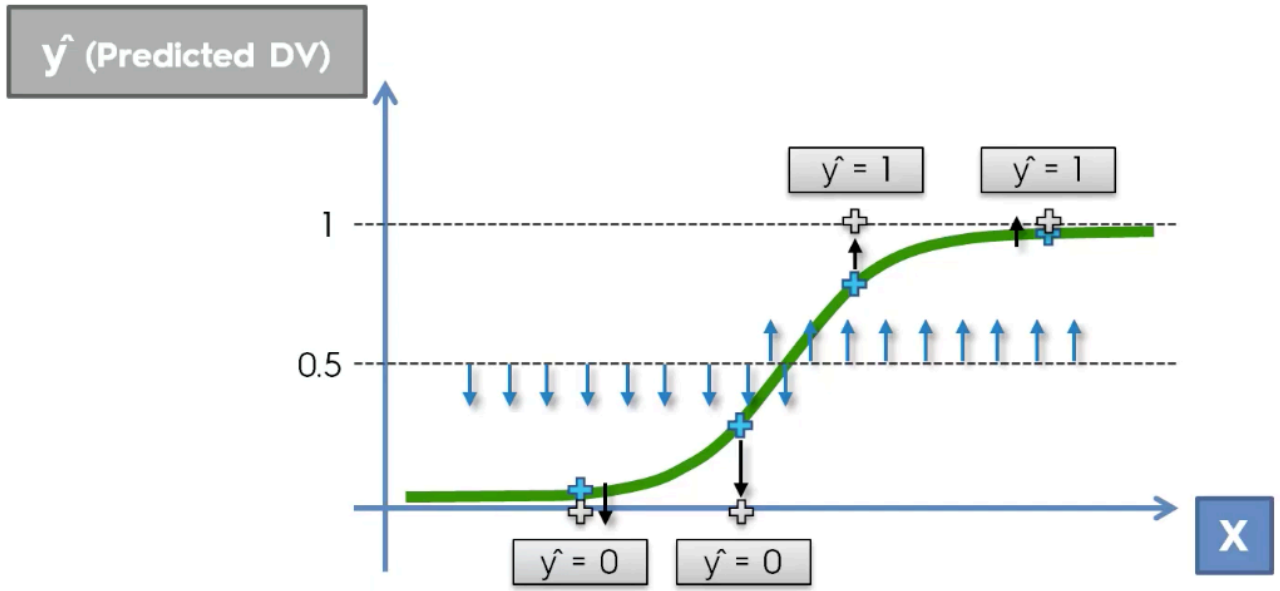


1. Logistic Regression Classification

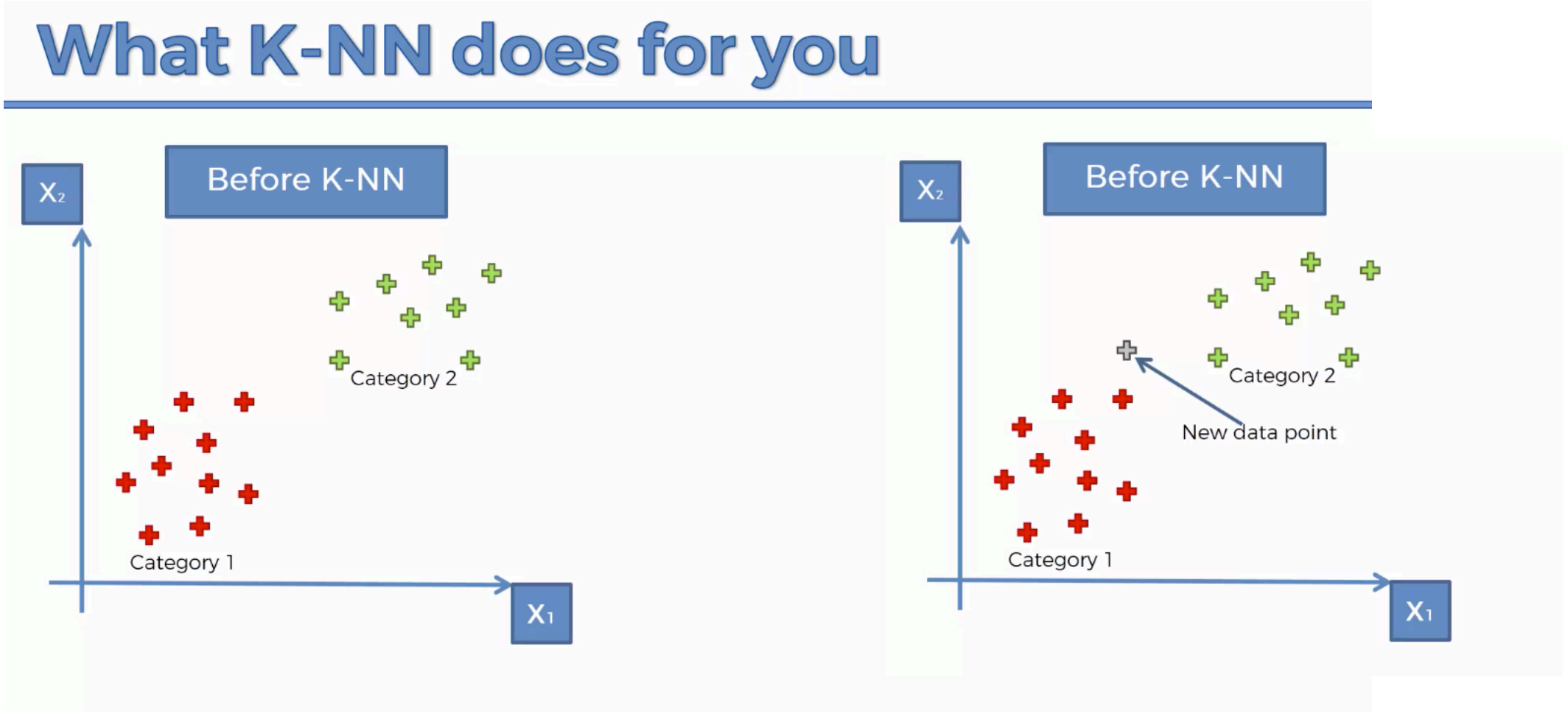


1. Logistic Regression Classification

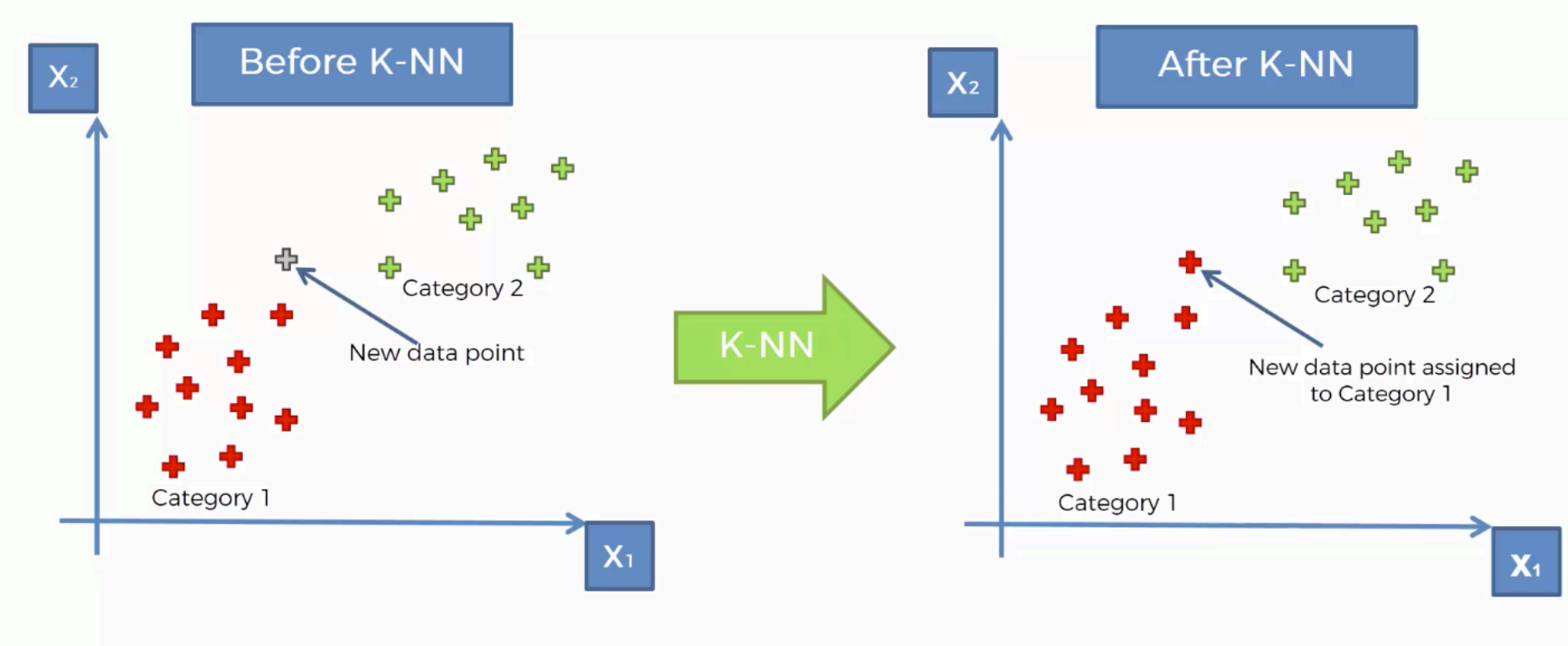




2. K-Nearest Neighbors (K-NN)



2. K-Nearest Neighbors (K-NN)



2. K-Nearest Neighbors (K-NN)

How did it do that ?

STEP 1: Choose the number K of neighbors

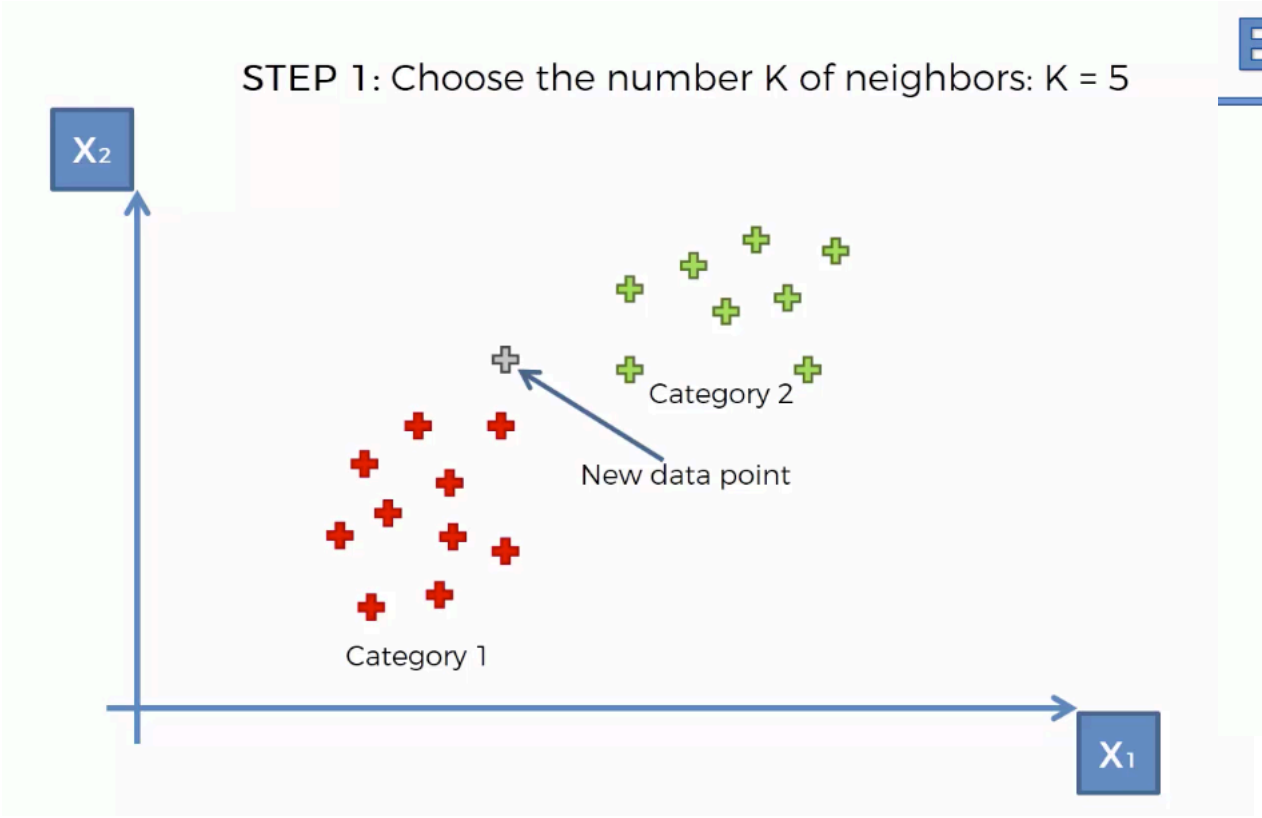
STEP 2: Take the K nearest neighbors of the new data point, according to the Euclidean distance

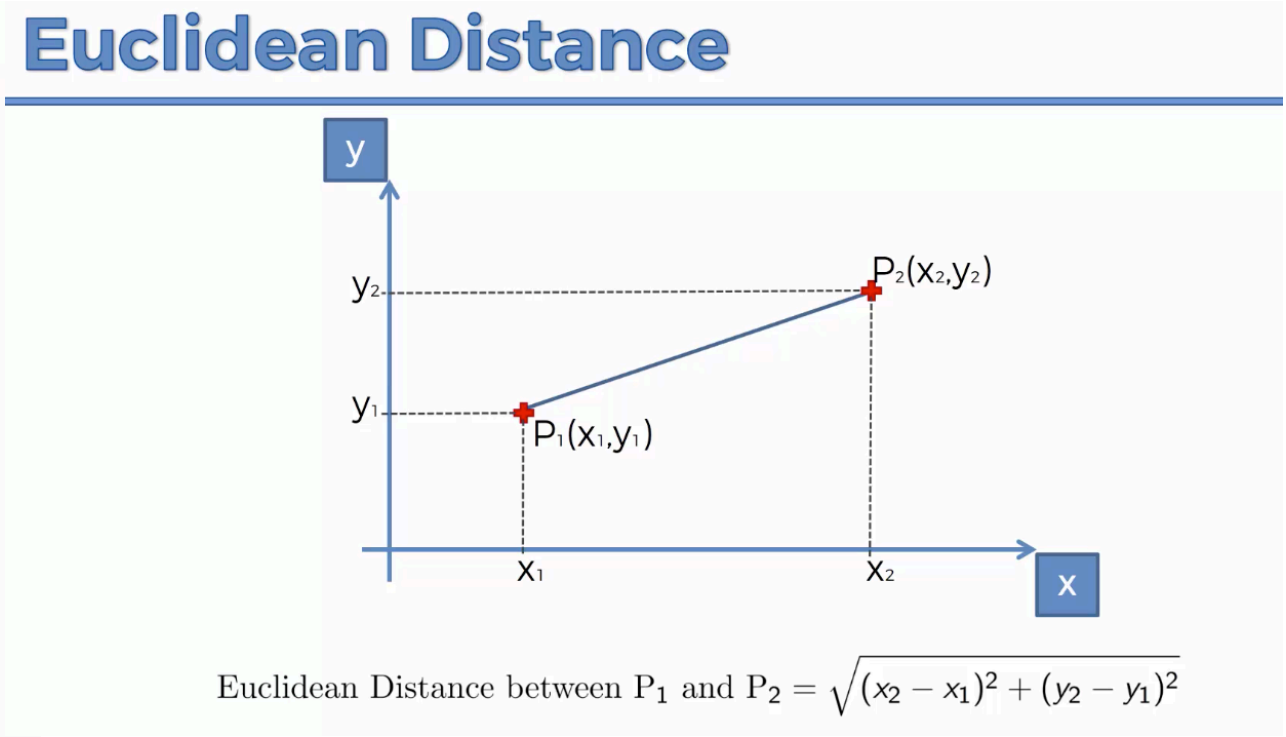
STEP 3: Among these K neighbors, count the number of data points in each category

STEP 4: Assign the new data point to the category where you counted the most neighbors

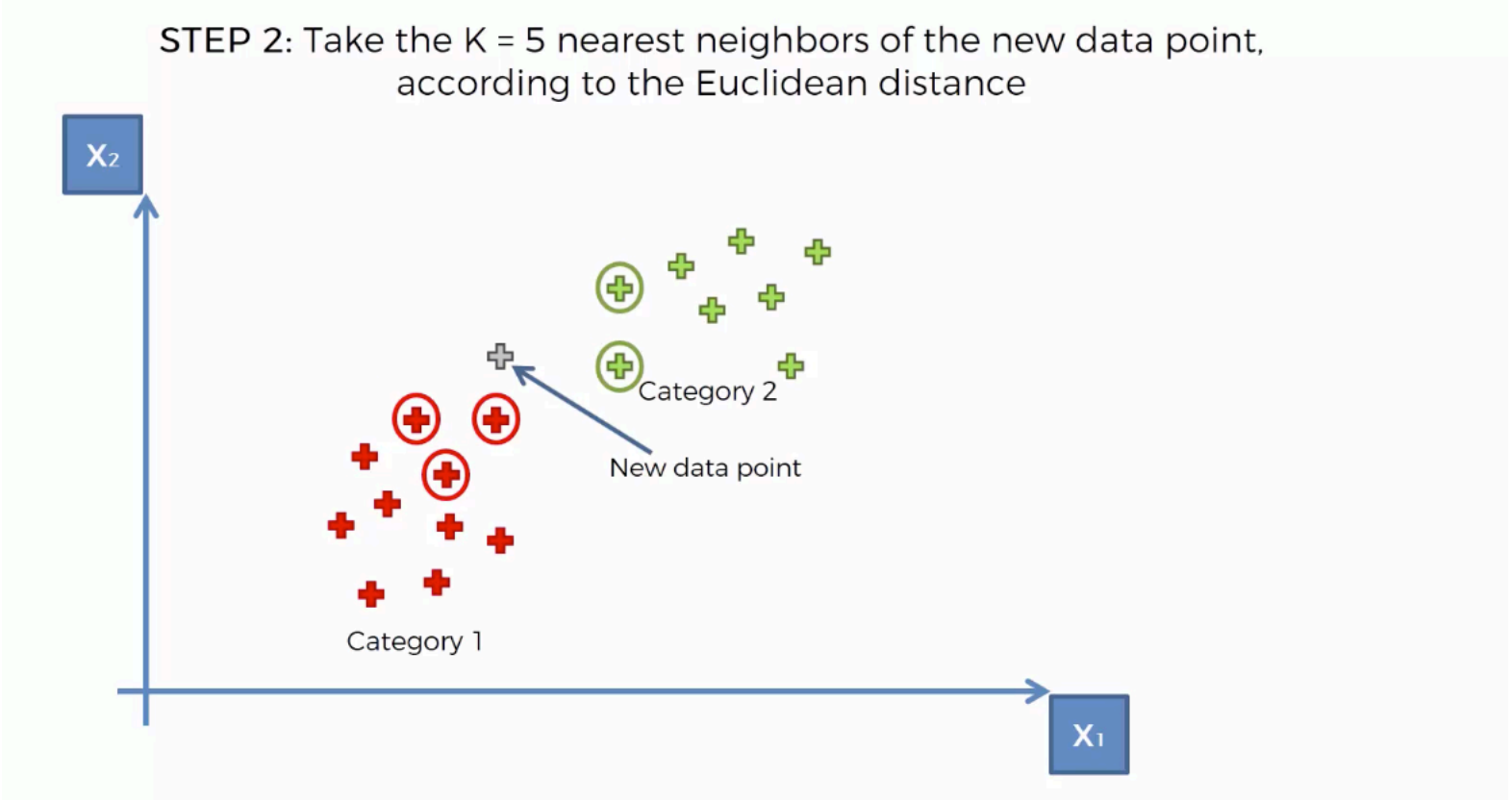
Your Model is Ready

2. K-Nearest Neighbors (K-NN)

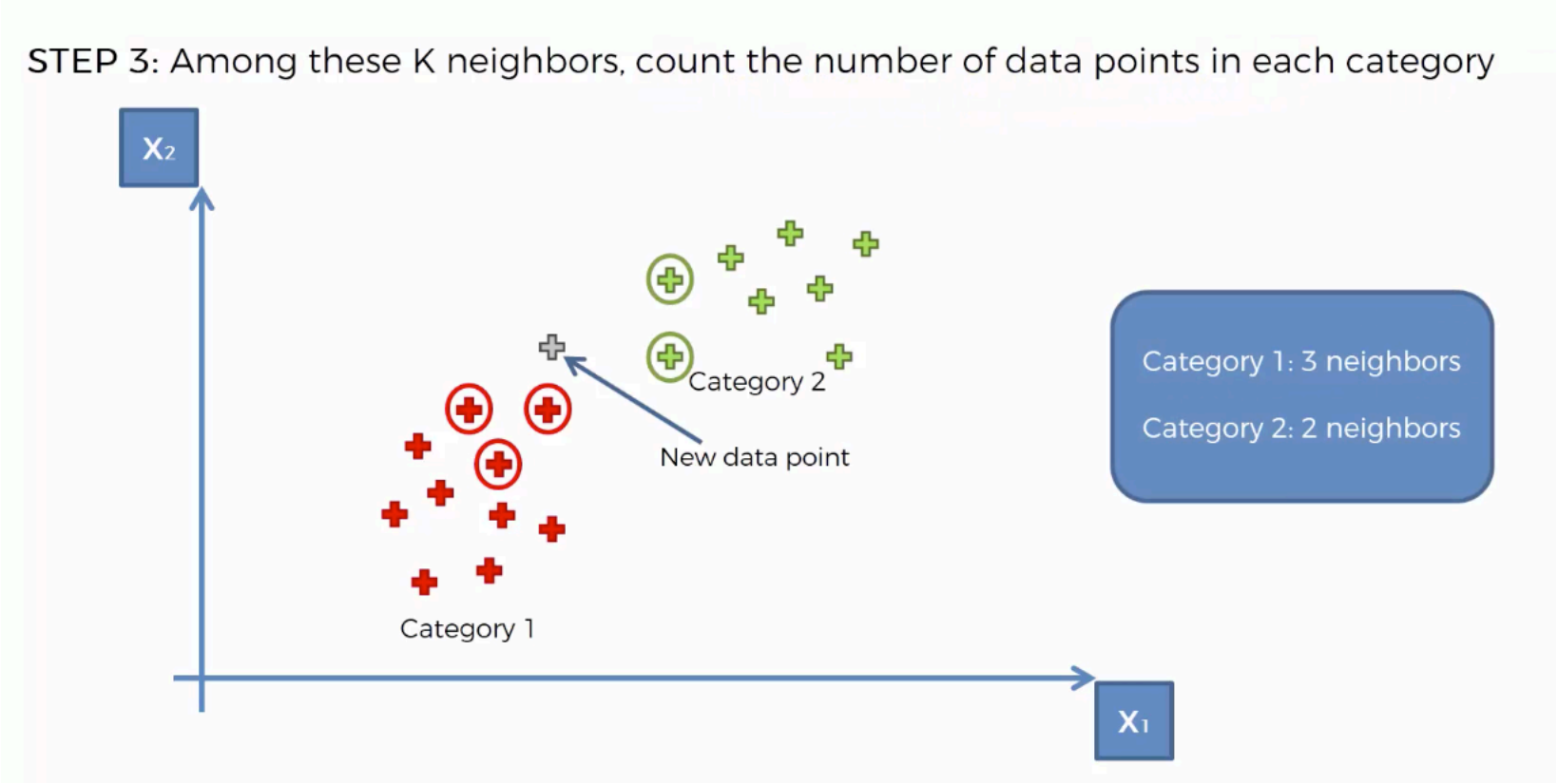




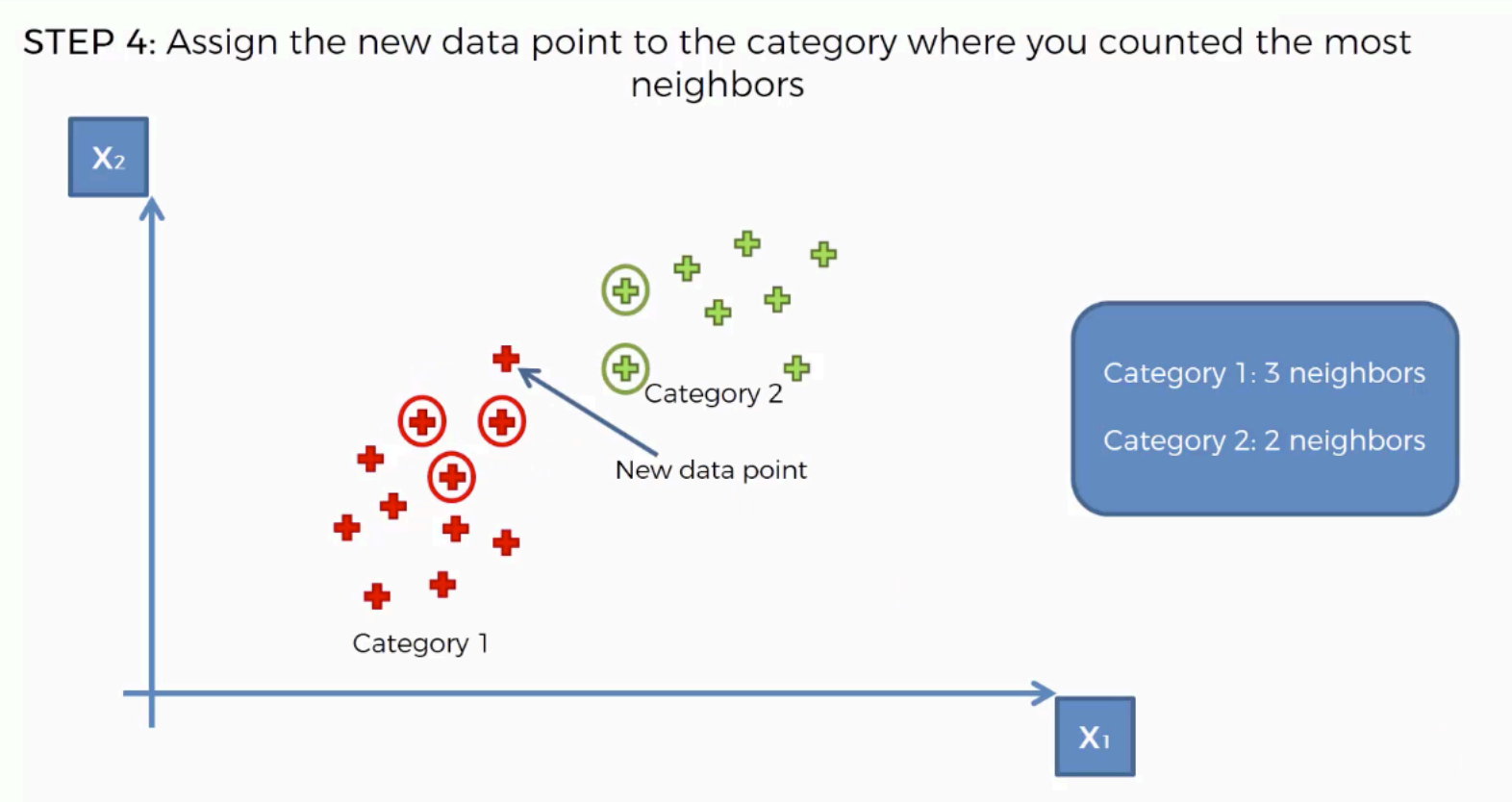
2. K-Nearest Neighbors (K-NN)



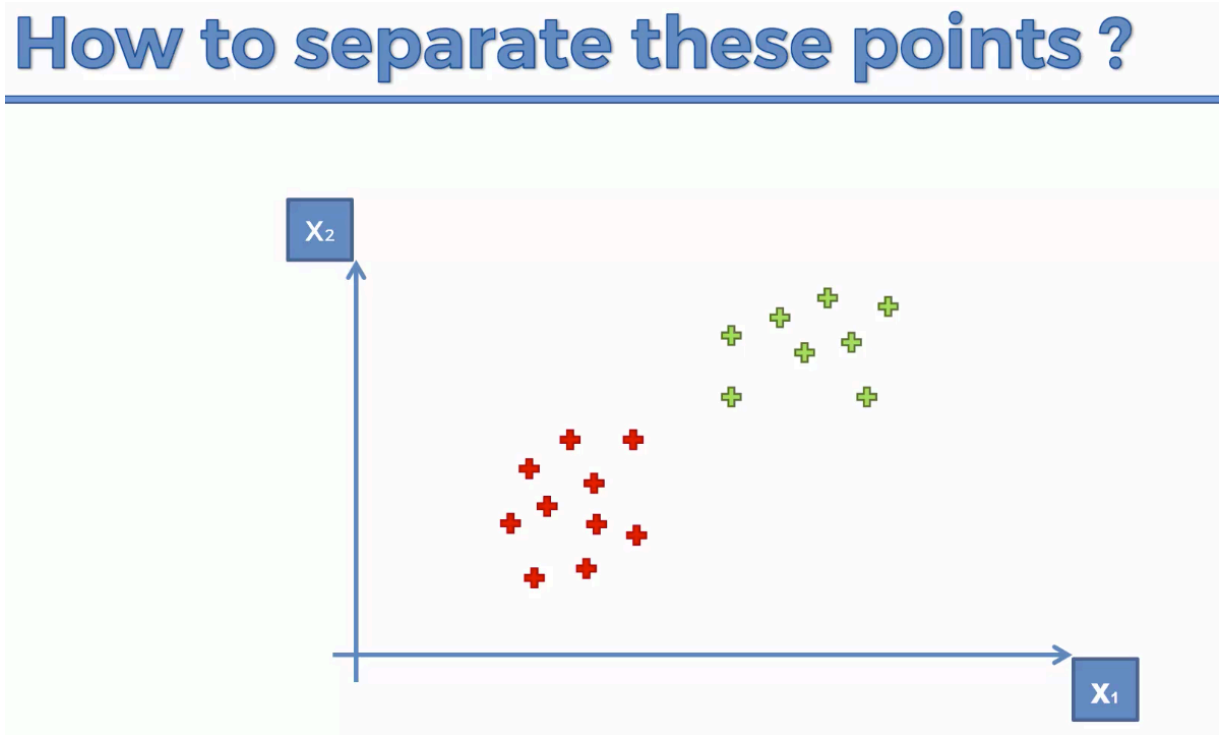
2. K-Nearest Neighbors (K-NN)

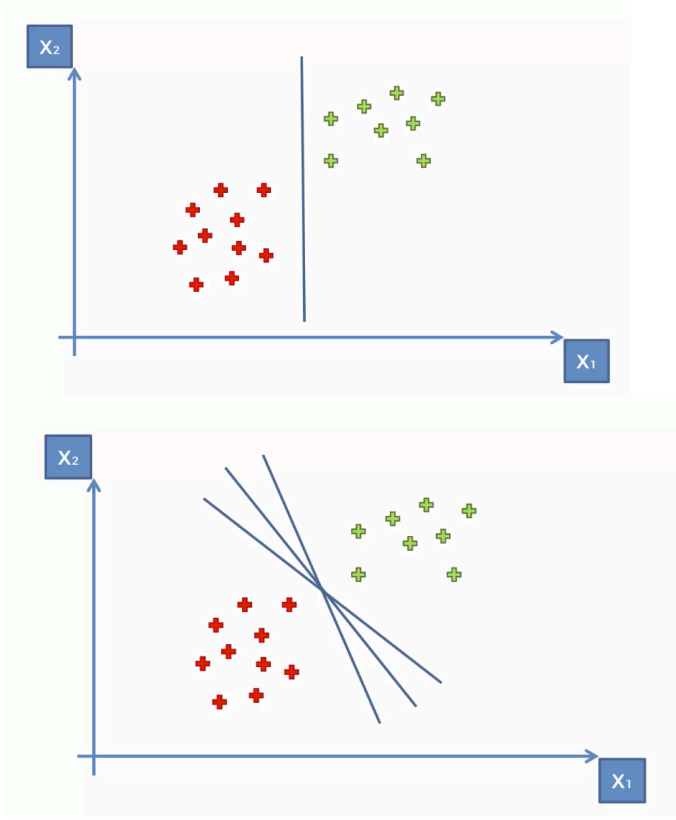


2. K-Nearest Neighbors (K-NN)



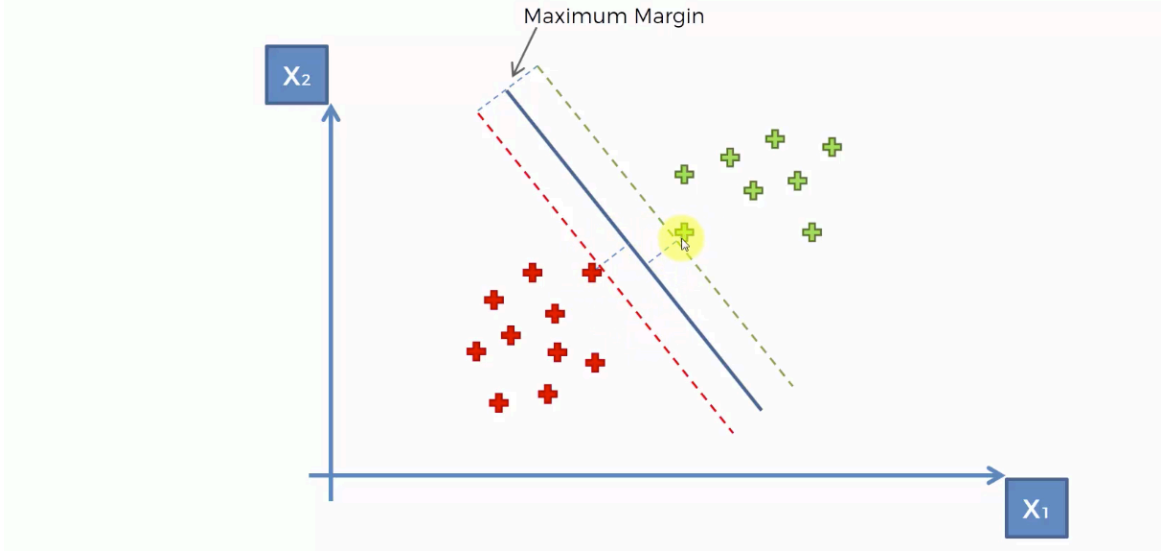
3. Support Vector Machine (SVM)



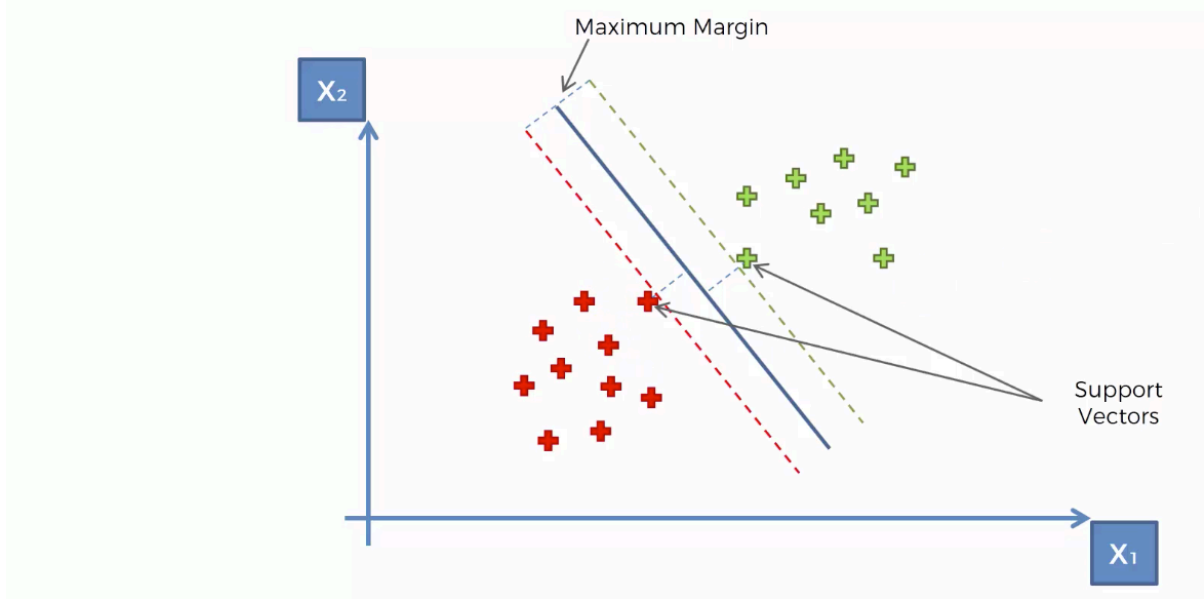


3. Support Vector Machine (SVM)

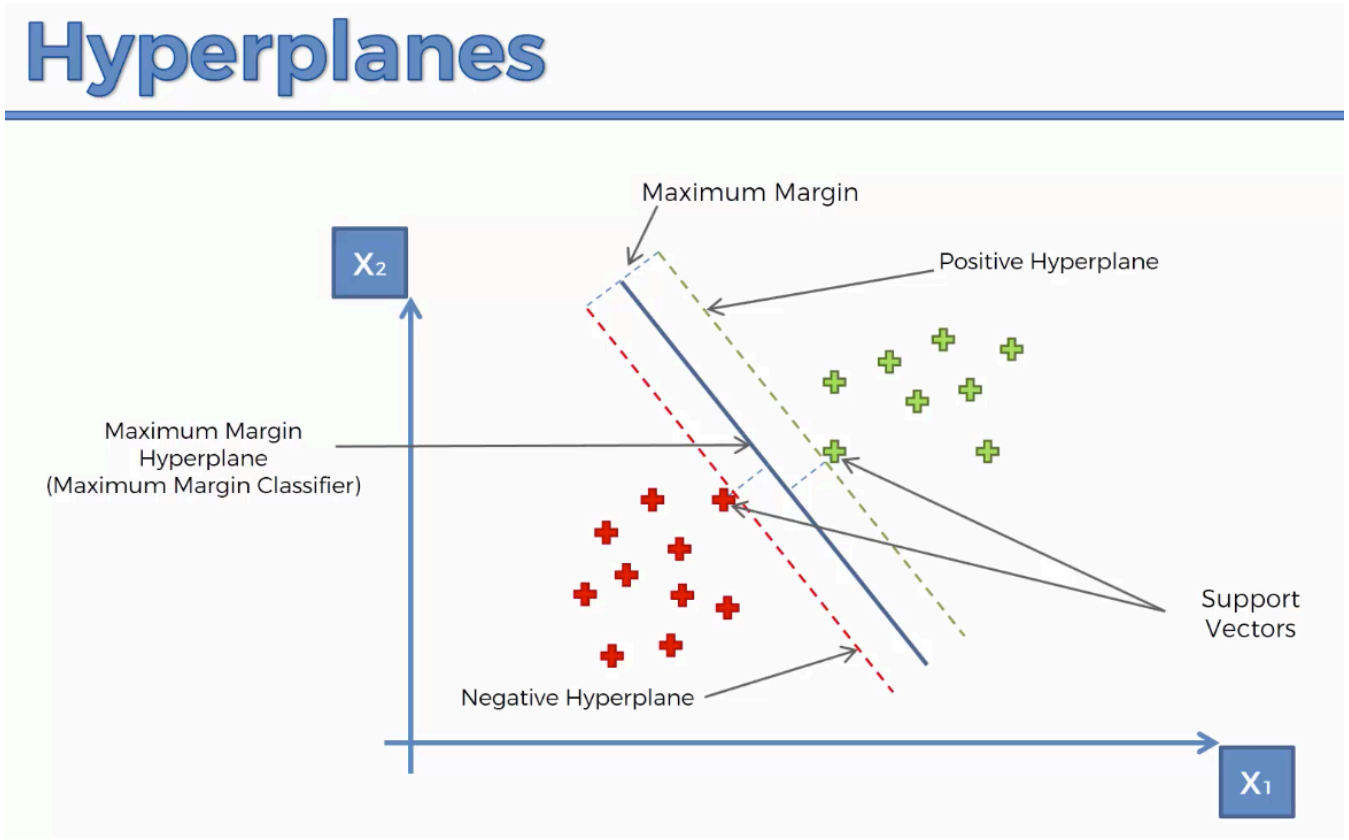
Maximum Margin



Support Vectors

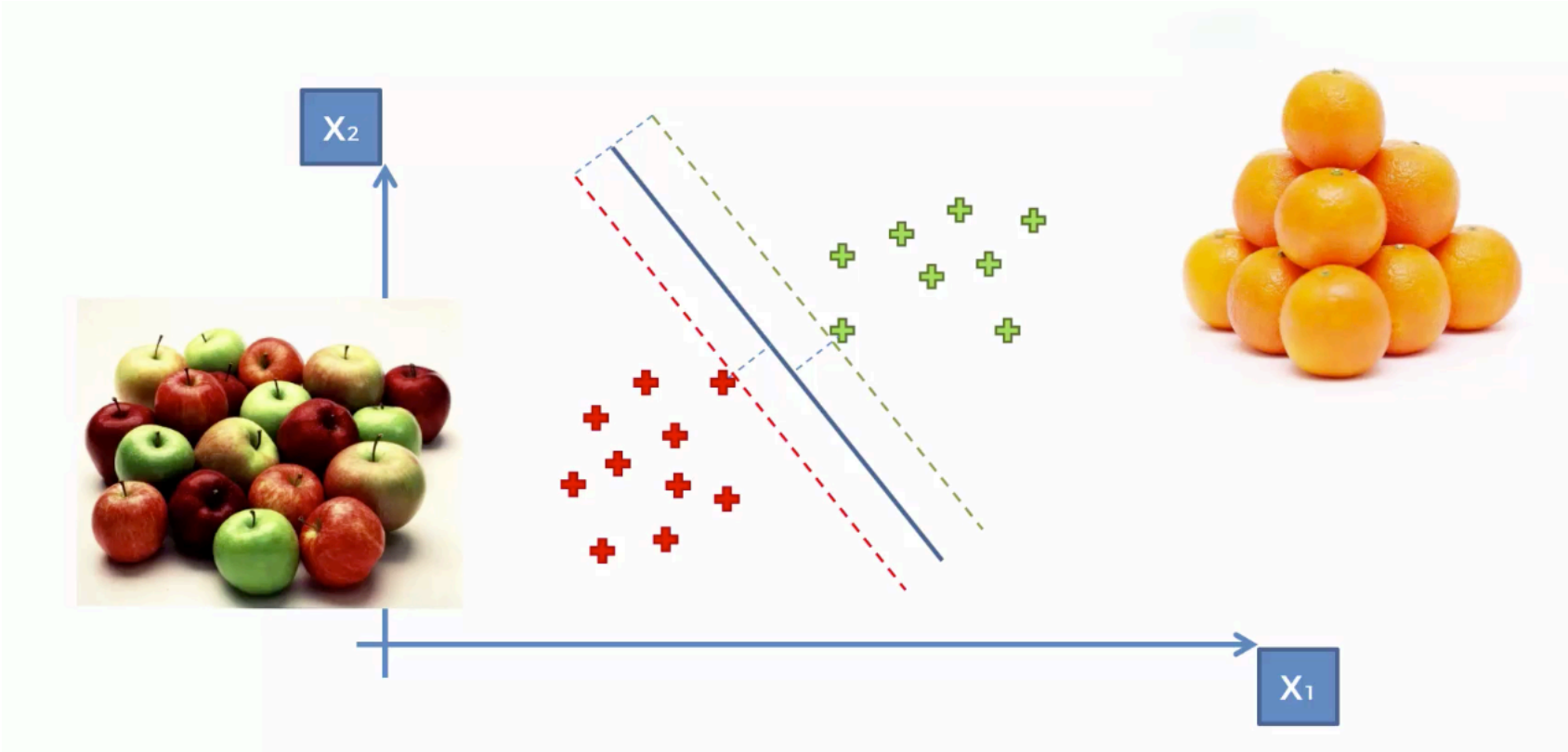


3. Support Vector Machine (SVM)

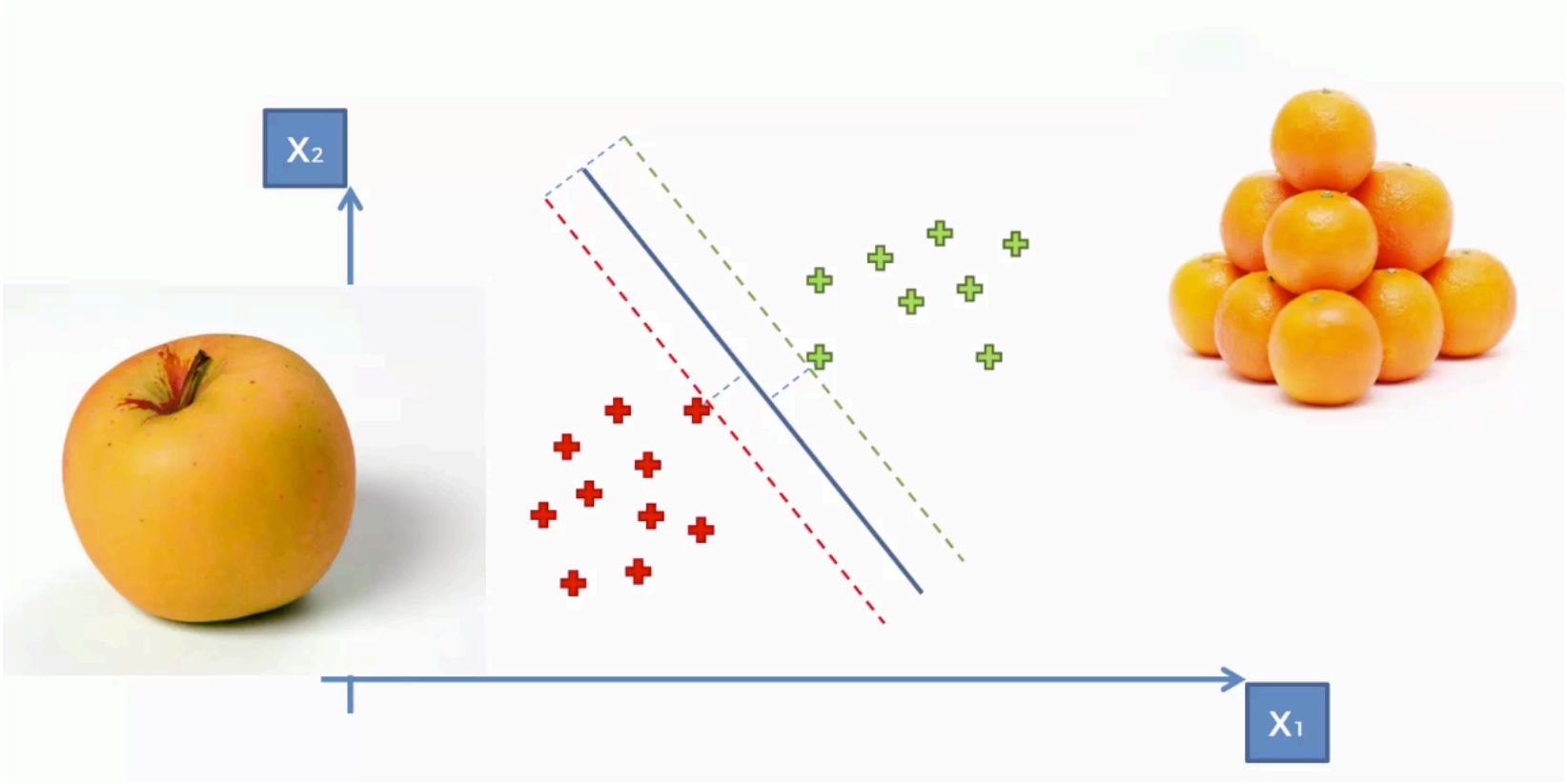


3. Support Vector Machine (SVM)

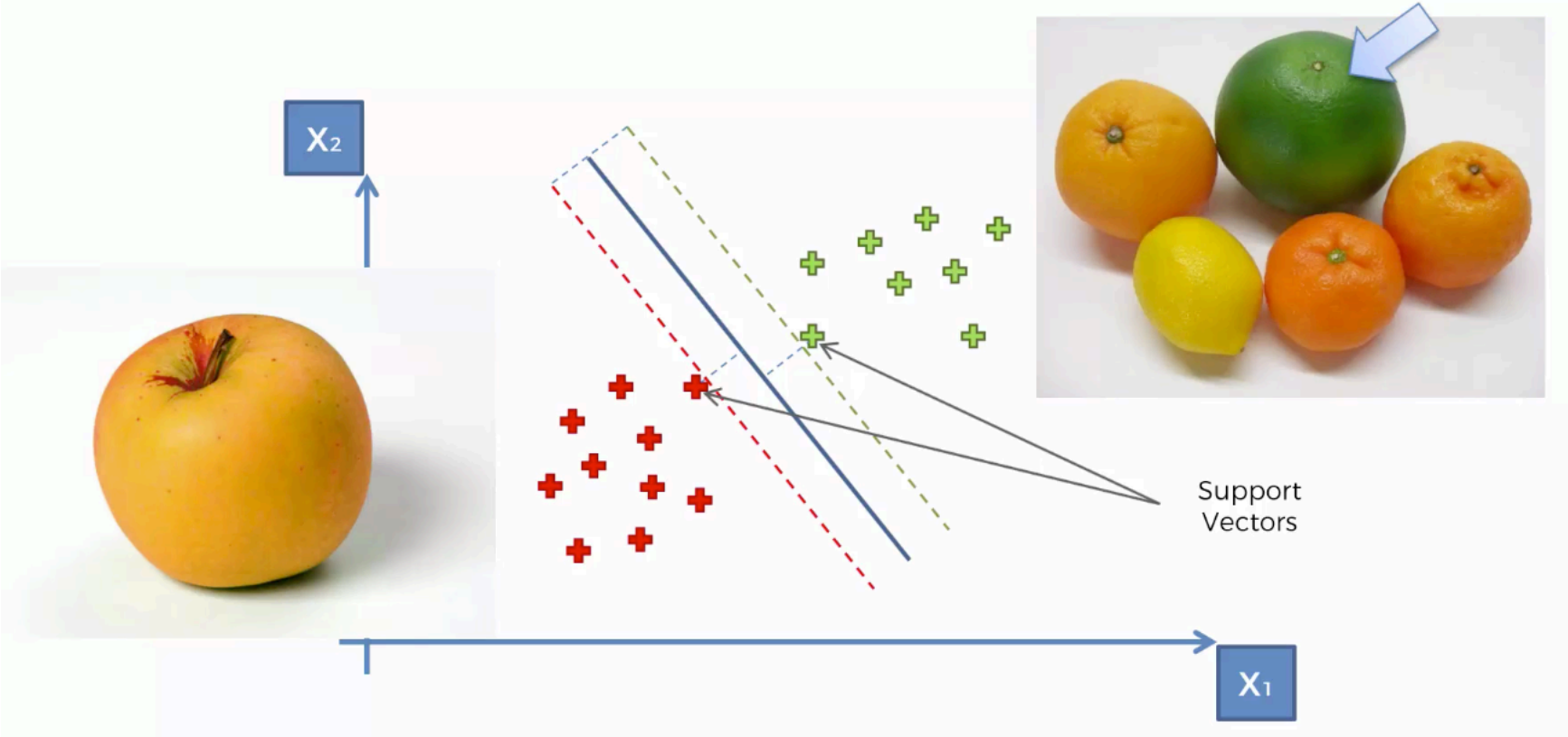
3. Support Vector Machine (SVM)



3. Support Vector Machine (SVM)



3. Support Vector Machine (SVM)



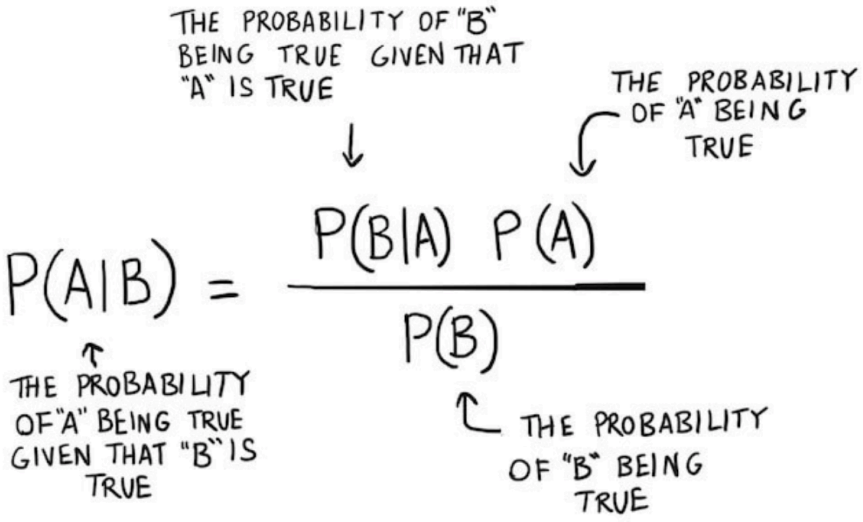
4. Naïve Bayes

A Brief look on Bayes Theorem :

Bayes Theorem helps us to find the probability of a hypothesis given our prior knowledge.

As per wikipedia,In probability theory and statistics, **Bayes' theorem** (alternatively **Bayes' law** or **Bayes' rule**, also written as **Bayes's theorem**) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

Lets look at the equation for Bayes Theorem,



Where,

- **P(A|B)** is the probability of hypothesis A given the data B. This is called the **posterior probability.**
- **P(B|A)** is the probability of data B given that the hypothesis A was true.
- **P(A)** is the probability of hypothesis A being true (regardless of the data). This is called the **prior probability of A**.
- **P(B)** is the probability of the data (regardless of the hypothesis).

How Naive Bayes algorithm works?

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

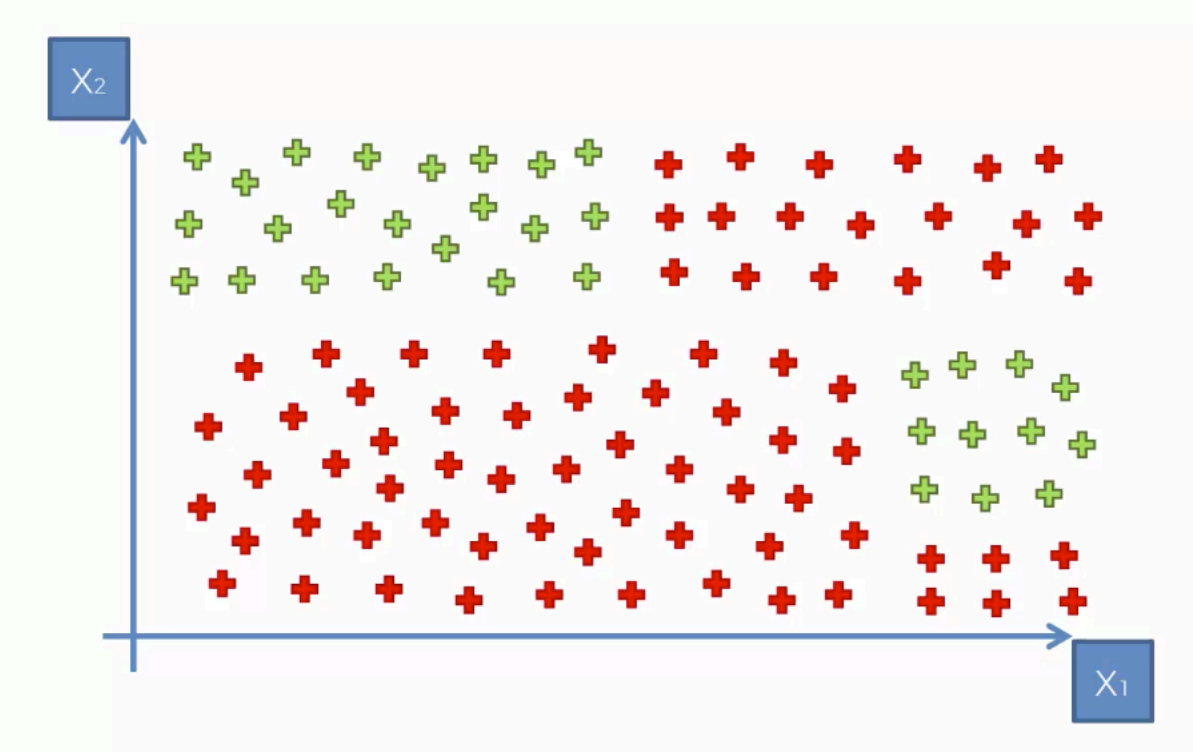
P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

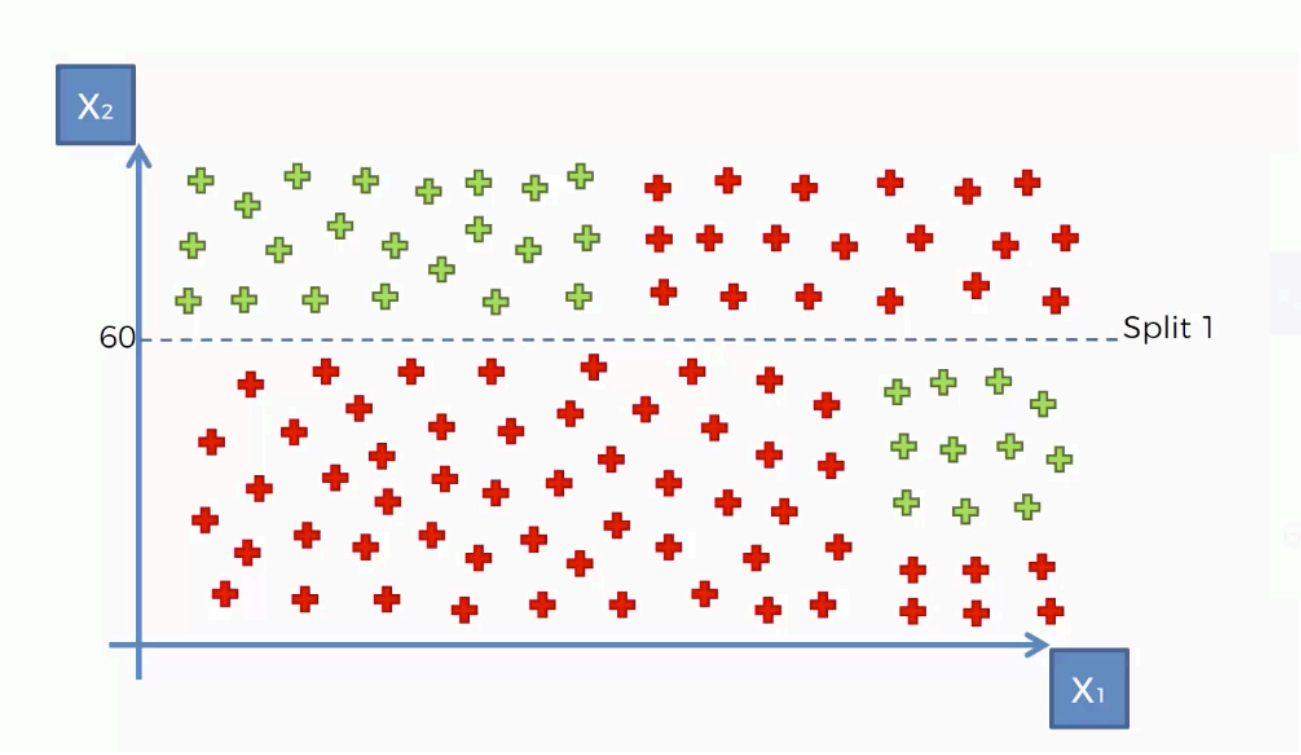
Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

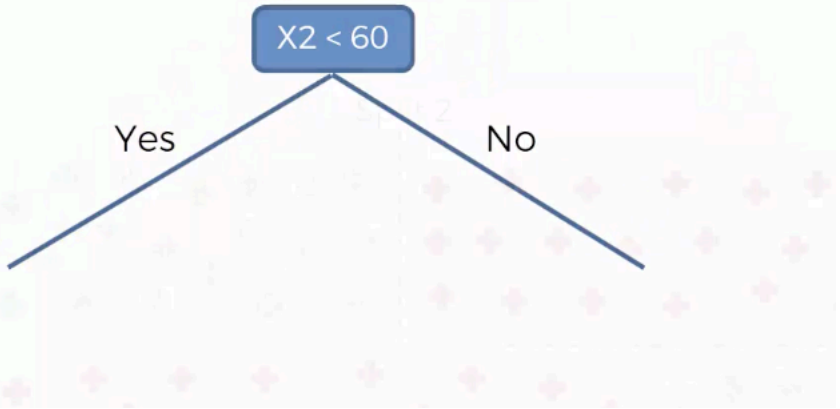
Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

5. Decision Tree Classification

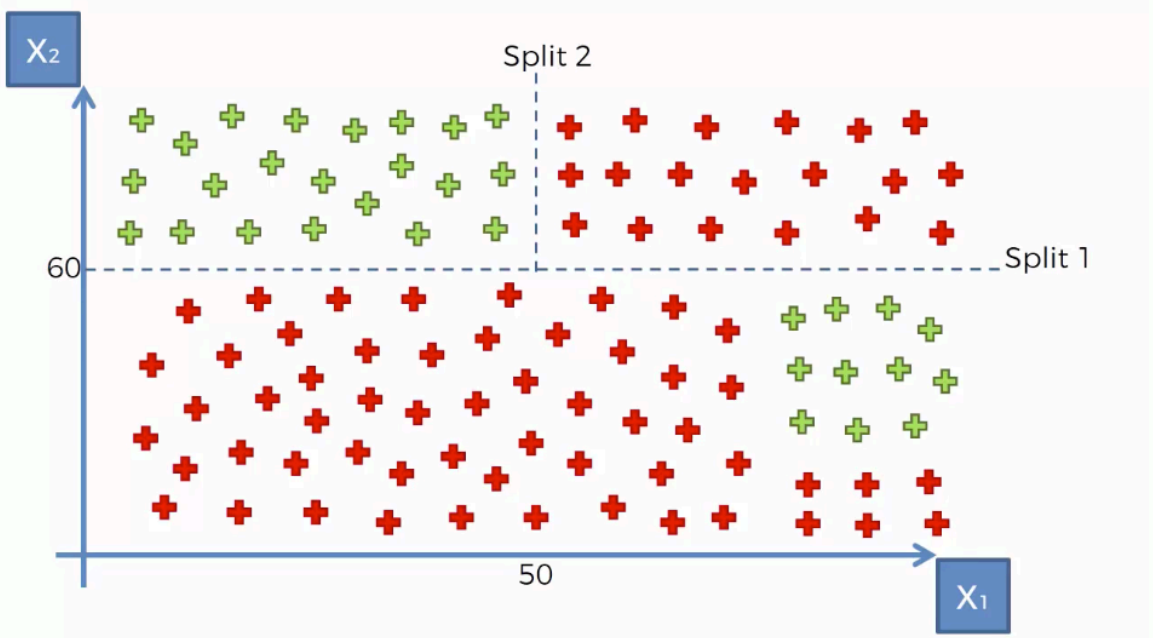


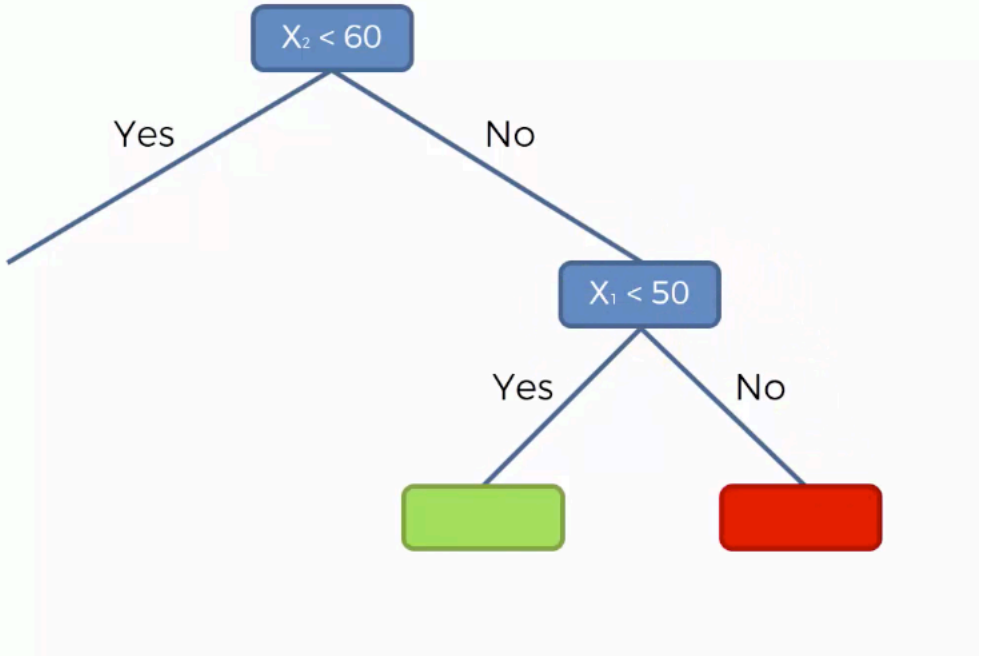
5. Decision Tree Classification



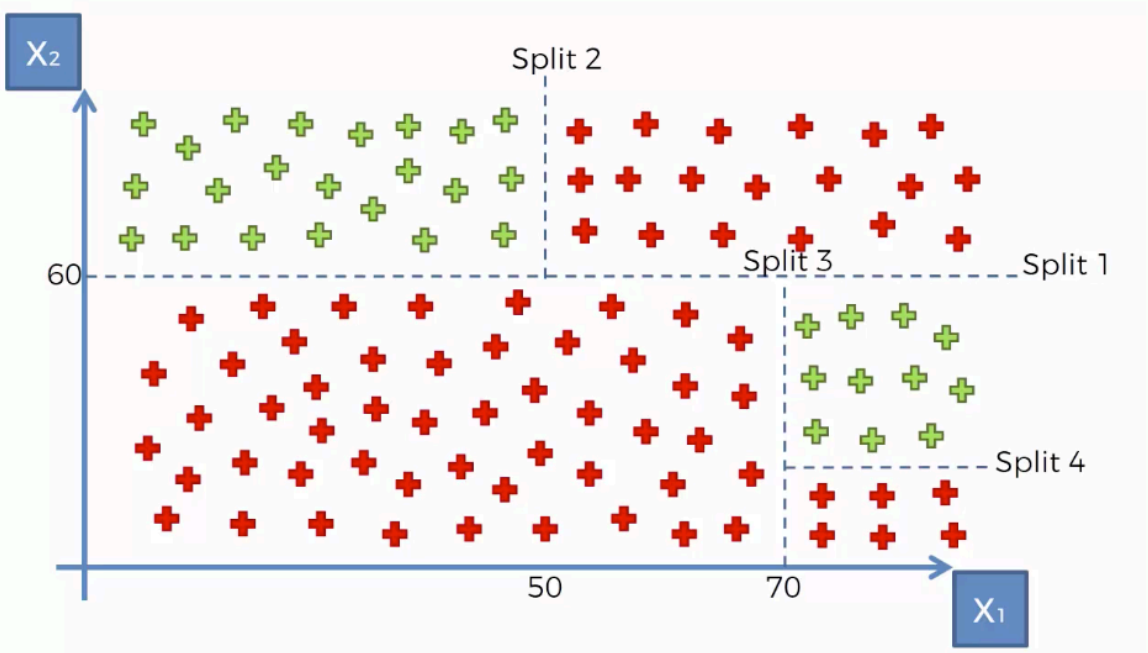


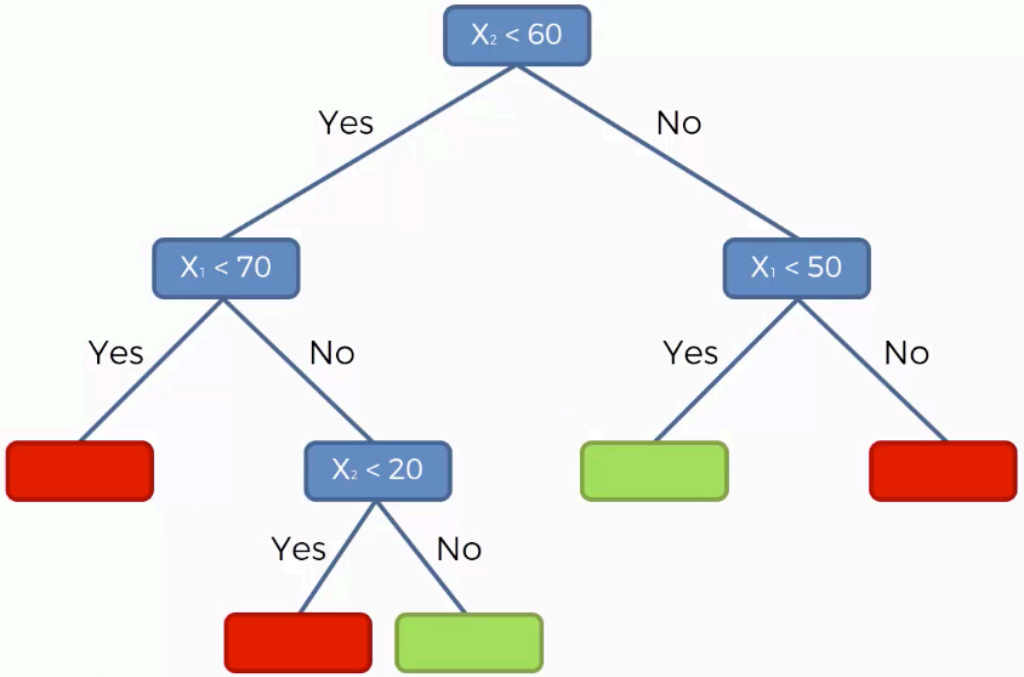
5. Decision Tree Classification





5. Decision Tree Classification





6. Random Forest Classification

Ensemble Learning

STEP 1: Pick at random K data points from the Training set.

STEP 2: Build the Decision Tree associated to these K data points.

STEP 3: Choose the number Ntree of trees you want to build and repeat STEPS 1 & 2

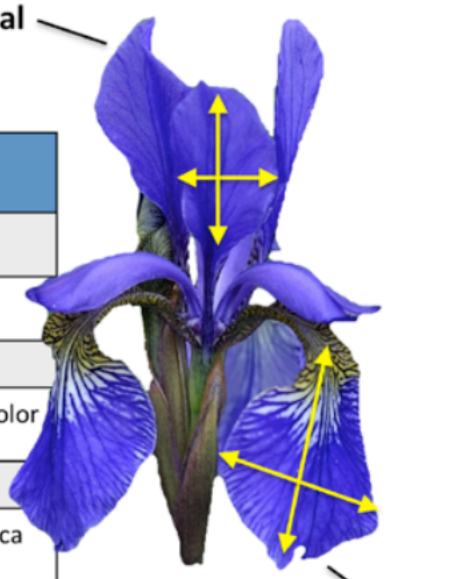
STEP 4: For a new data point, make each one of your Ntree trees predict the category to which the data points belongs, and assign the new data point to the category that wins the majority vote.

Classification Model	Pros	Cons
Logistic Regression	Probabilistic approach, gives informations about statistical significance of features	The Logistic Regression Assumptions
K-NN	Simple to understand, fast and efficient	Need to choose the number of neighbours k
SVM	Performant, not biased by outliers, not sensitive to overfitting	Not appropriate for non linear problems, not the best choice for large number of features
Kernel SVM	High performance on nonlinear problems, not biased by outliers, not sensitive to overfitting	Not the best choice for large number of features, more complex
Naive Bayes	Efficient, not biased by outliers, works on nonlinear problems, probabilistic approach	Based on the assumption that features have same statistical relevance
Decision Tree Classification	Interpretability, no need for feature scaling, works on both linear / nonlinear problems	Poor results on too small datasets, overfitting can easily occur
Random Forest Classification	Powerful and accurate, good performance on many problems, including non linear	No interpretability, overfitting can easily occur, need to choose the number of trees

Classification Example 4: Iris Species

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

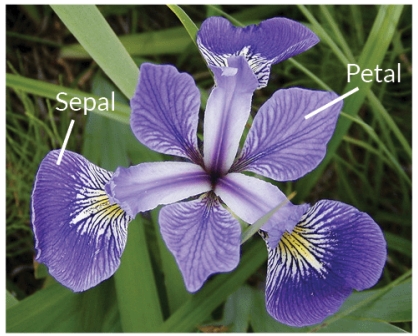


Features
(attributes, measurements, dimensions)

Class labels
(targets)

Row: samples, records, instances, observations
Column: attributes, features, measurements, dimensions

More than 2 output classes (3 classes) →
multi-class classification problem

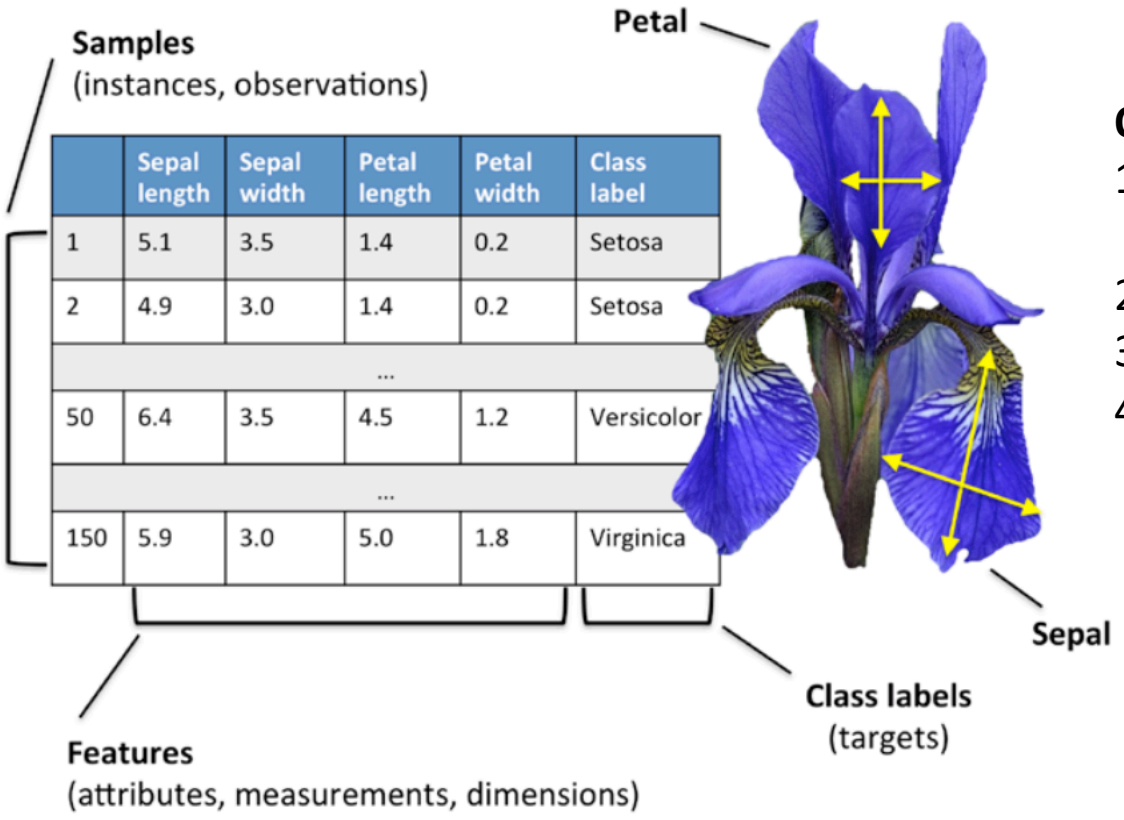


Iris Versicolor

Iris Setosa

Iris Virginica

Classification Problem: Iris Species (# Lab 1)



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Questions:

1. What is the classification machine Learning method should we use for predicting the iris species?
2. How is the Accuracy?
3. How is the Confusion matrix?
4. How is the Precision/Recall/F-Measure?

Machine Learning Algorithm Performance Metrics

Confusion Matrix (Precision, Recall, F-Measure)

The confusion matrix is a handy presentation of the accuracy of a model with two or more classes. The table presents predictions on the x-axis and true outcomes on the y-axis. The cells of the table are the number of predictions made by a machine learning algorithm. For example, a machine learning algorithm can predict 0 or 1 and each prediction may actually have been a 0 or 1. Predictions for 0 that were actually 0 appear in the cell for prediction = 0 and actual = 0, whereas predictions for 0 that were actually 1 appear in the cell for prediction = 0 and actual = 1. And so on.

		predicted condition	
	total population	prediction positive	prediction negative
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**

Precision = TruePositives / (TruePositives + FalsePositives)

Recall = TruePositives / (TruePositives + FalseNegatives)

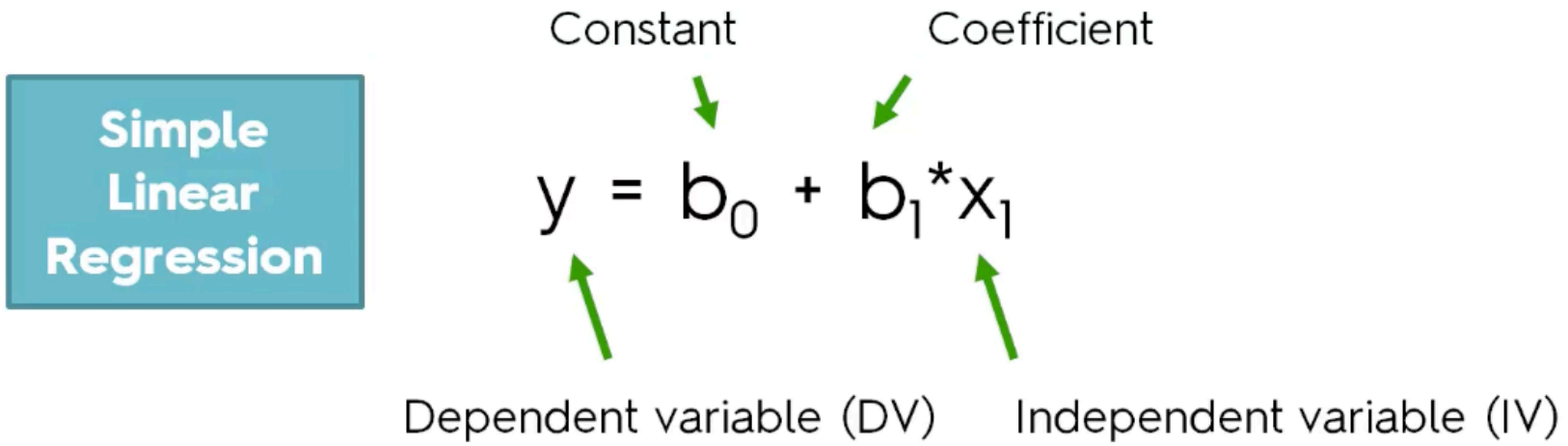
F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Regression Methods

	ML Lesson
	Lesson 1: Python Ecosystem for Machine Learning. Lesson 2: Python and SciPy Crash Course. (Numpy, Pandas, Matplotlib)
Define Problem & Analyze Data	Lesson 3: Understand Data With Descriptive Statistics. Lesson 4: Understand Data With Visualization.
Prepare Data	Lesson 5: Pre-Process Data.
Evaluate Algorithms	Lesson 6: Resampling Methods. Lesson 7: Algorithm Evaluation Metrics. Lesson 8: Spot-Check Classification Algorithms. Lesson 9: Spot-Check Regression Algorithms. Lesson 10: Model Selection.
Improve Results	Lesson 11: Ensemble Methods. Lesson 12: Algorithm Parameter Tuning.
Present Results	Lesson 13: Model Finalization.

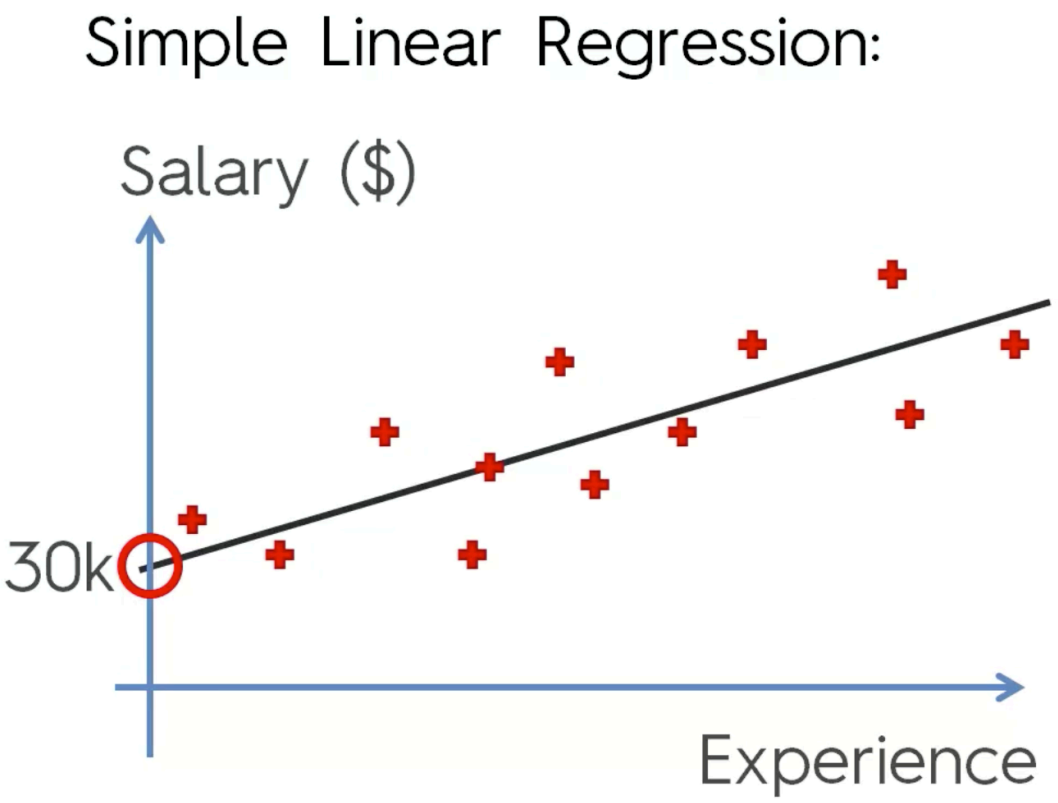
Regression Algorithms

Simple Linear Regression



Regression Algorithms

Simple Linear Regression

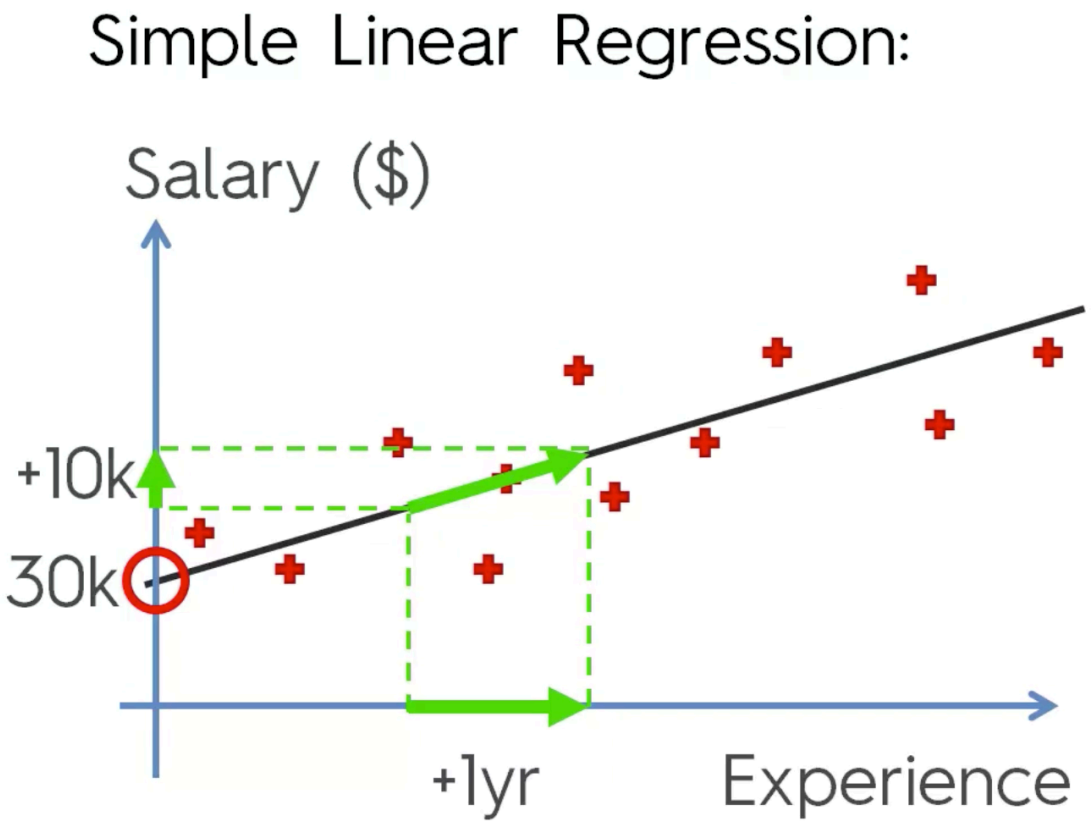


$$y = b_0 + b_1 * x$$

Salary = b_0 + b_1 *Experience

Regression Algorithms

Simple Linear Regression



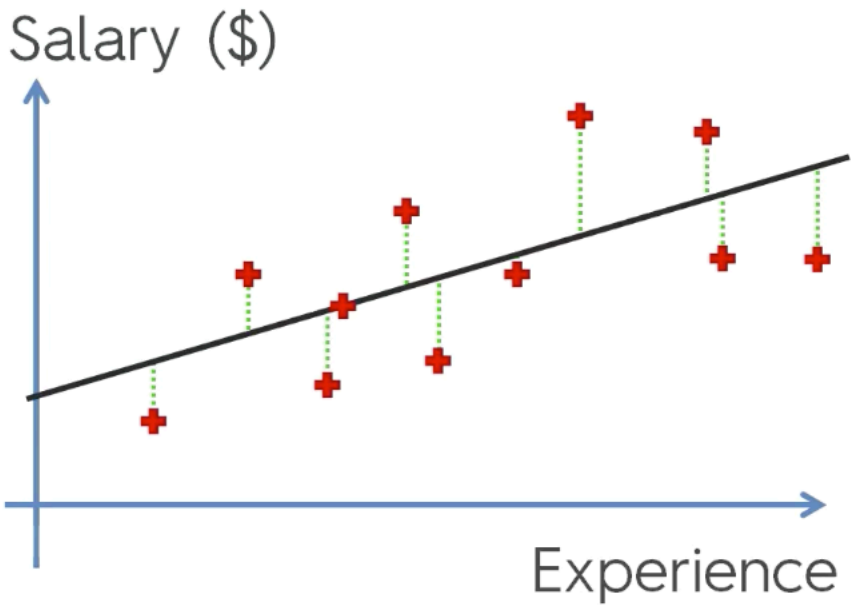
$$y = b_0 + b_1 * x$$

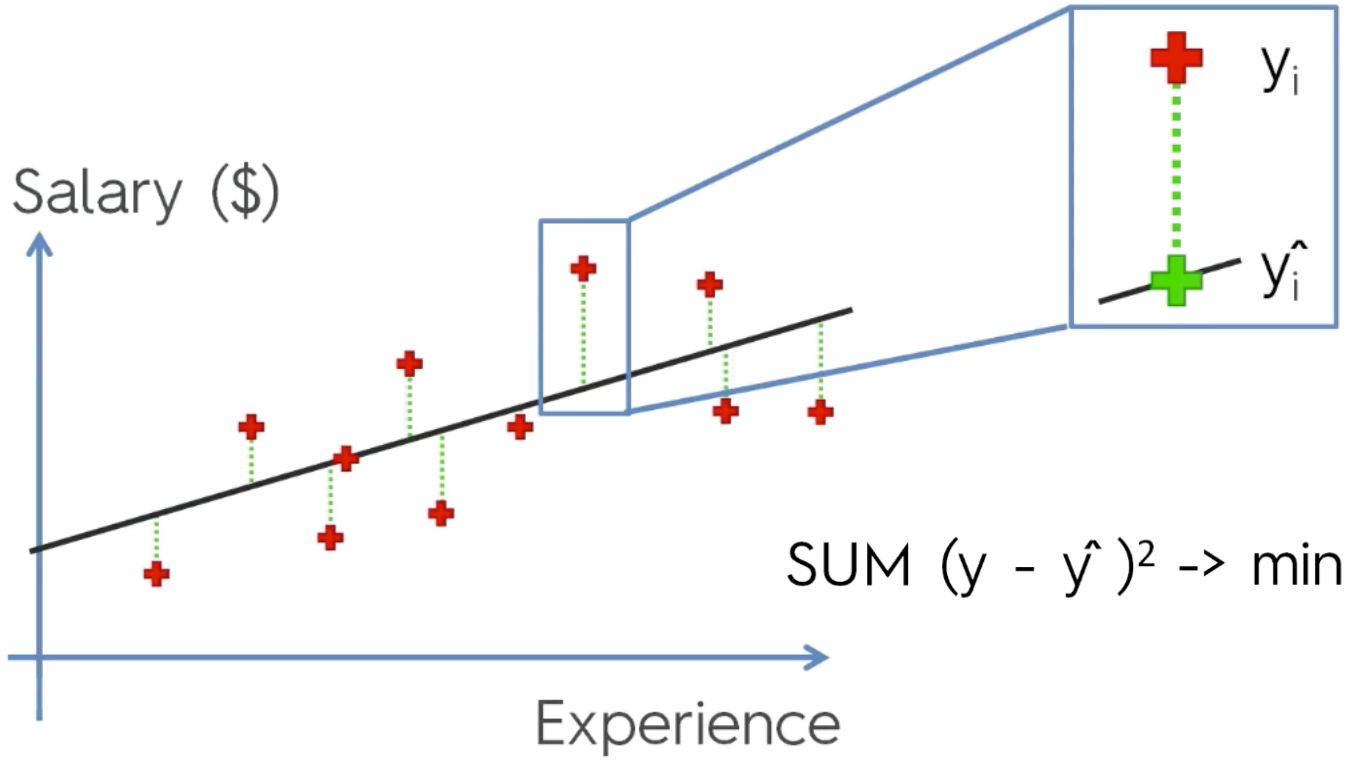
Salary = b_0 + b_1 *Experience

Evaluating Regression Models Performance

Ordinary Least Squares

Simple Linear Regression:



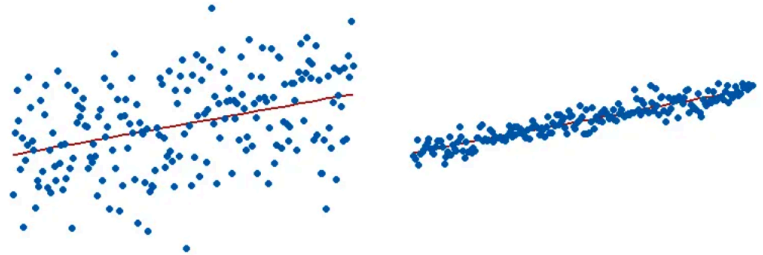


Regression Metrics

1. Mean Absolute Error
2. Mean Squared Error
3. R^2

Visual Representation of R-squared

To visually demonstrate how R-squared values represent the scatter around the regression line, you can plot the fitted values by observed values.



The R-squared for the regression model on the left is 15%, and for the model on the right it is 85%. When a regression model accounts for more of the variance, the data points are closer to the regression line. In practice, you'll never see a regression model with an R^2 of 100%. In that case, the fitted values equal the data values and, consequently, all of the observations fall exactly on the regression line.

Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

Mean absolute error. Image by the author.

MAE does not penalize large errors.

Mean Squared Error (MSE)

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

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

Mean squared error. Image by the author.

MSE penalizes large errors.

Demo House Price Prediction