Applications of ML

What's the difference between fit and fit_transform in scikit-learn models?

To center the data (make it have zero mean and unit standard error), you subtract the mean and then divide the result by the standard deviation:

$$x' = \frac{x - \mu}{\sigma}$$

You do that on the training set of data. But then you have to apply the same transformation to your testing set (e.g. in cross-validation), or to newly obtained examples before forecast. But you have to use the exact same two parameters μ and σ (values) that you used for centering the training set.

Hence, every sklearn's transform's `fit()` just calculates the parameters (e.g. μ and σ in case of StandardScaler) and saves them as an internal object's state. Afterwards, you can call its `transform()` method to apply the transformation to any particular set of examples.

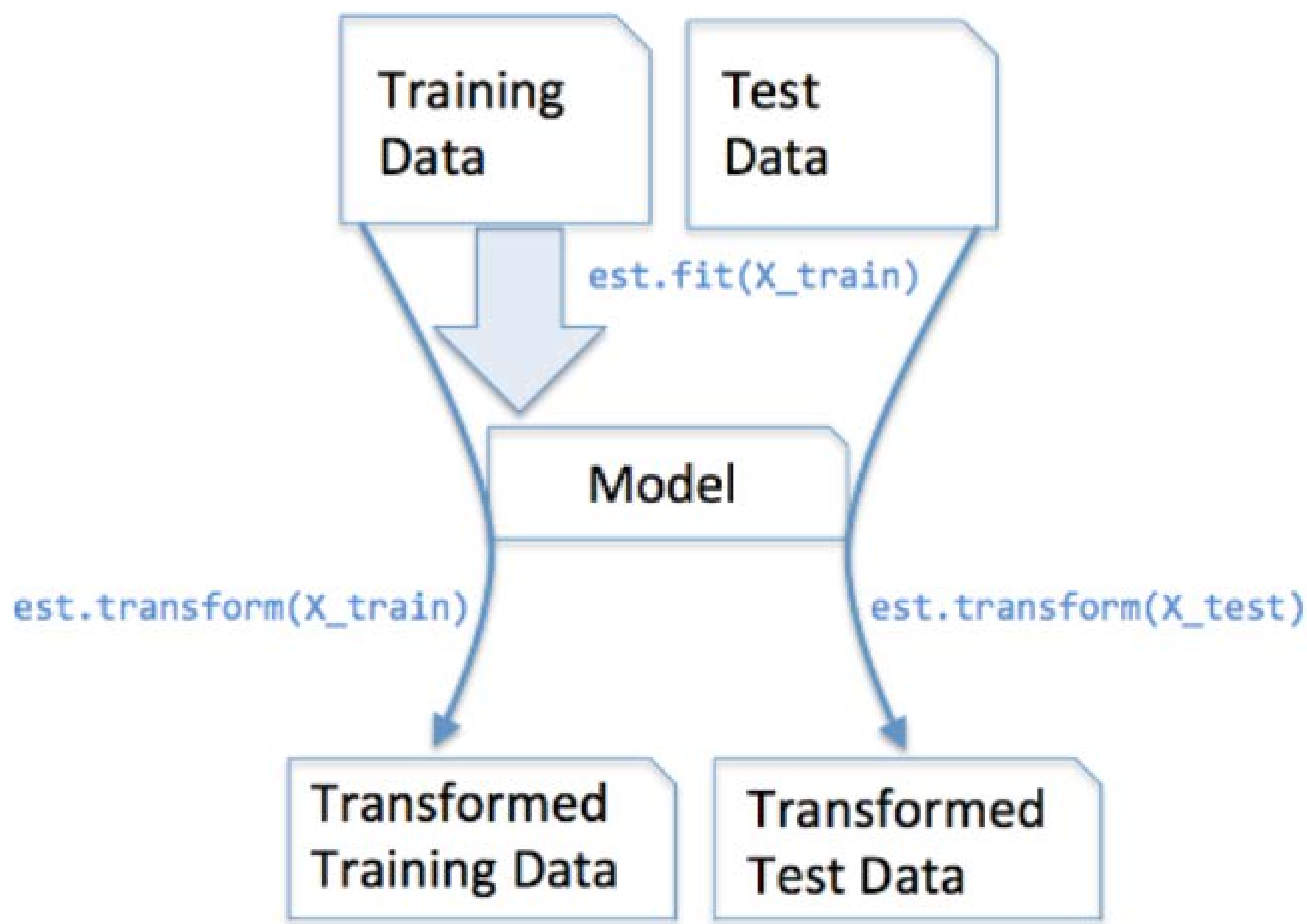
`fit_transform()` joins these two steps and is used for the initial fitting of parameters on the training set x, while also returning the transformed x'. Internally, the transformer object just calls first `fit()` and then `transform()` on the same data.

In **scikit-learn estimator api**, fit, transform, fit_transform

`fit()` : used for generating learning model parameters from training data

`transform()` : parameters generated from `fit()` method,applied upon model to generate transformed data set.

`fit_transform()` : combination of `fit()` and `transform()` api on same data set



Feature Scaling

Standardisation	Normalisation
$x_{\text{stand}} = \dfrac{x - \text{mean}(x)}{\text{standard deviation }(x)}$	$x_{\text{norm}} = \dfrac{x - \min(x)}{\max(x) - \min(x)}$

Regressions

Simple Linear Regression

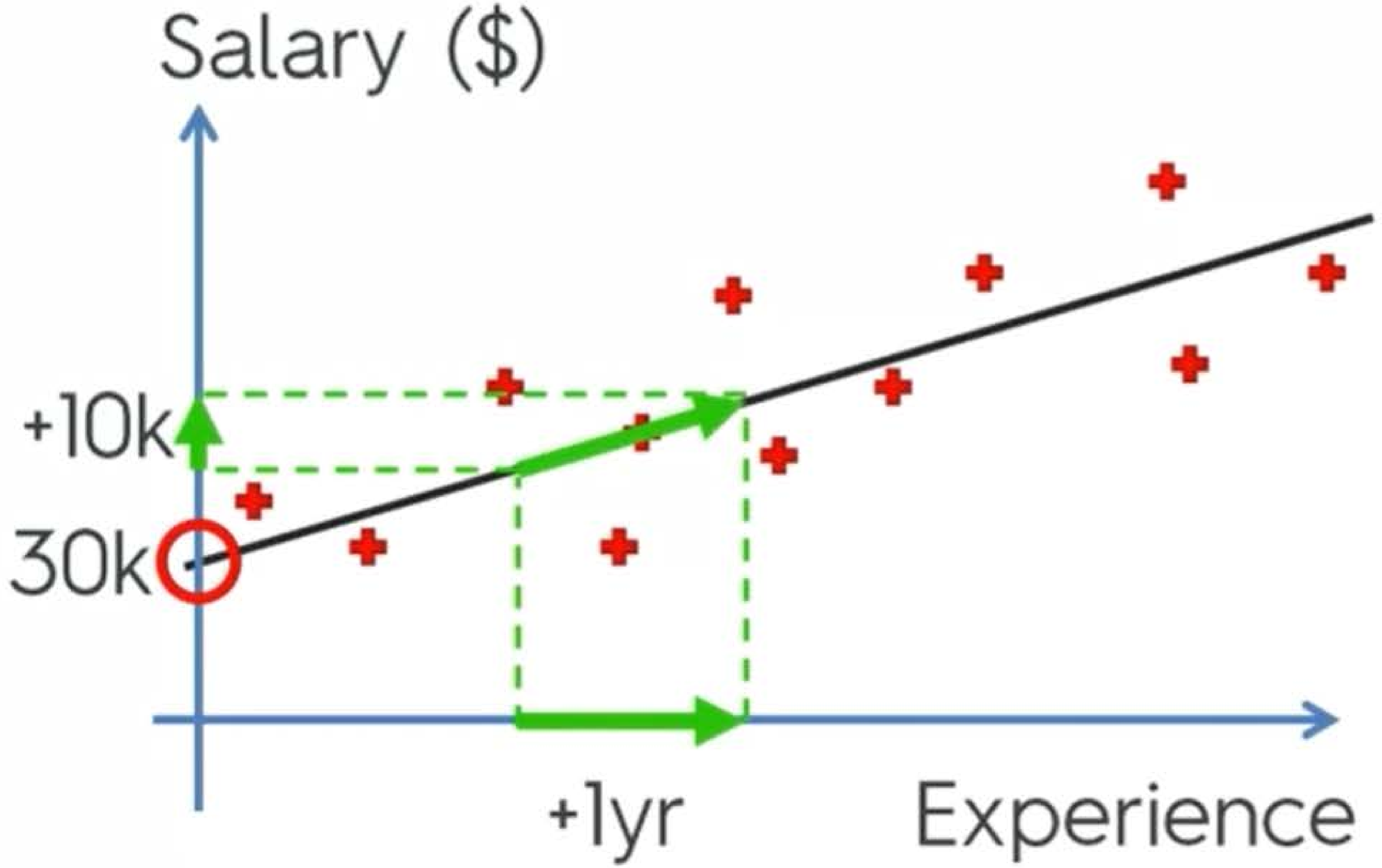
Constant   Coefficient

$$y = b_0 + b_1{}^{*}x_1$$

Dependent variable (DV)   Independent variable (IV)

Regressions

Simple Linear Regression:



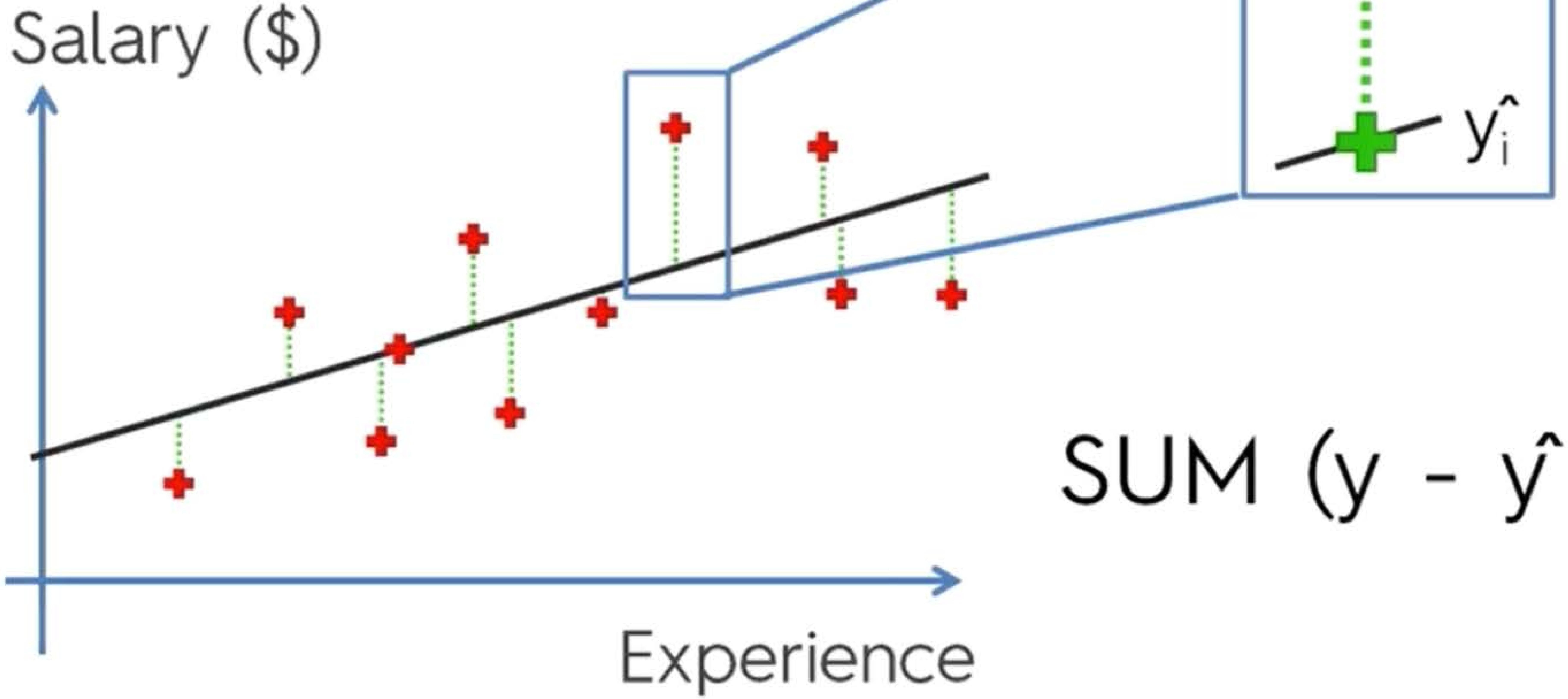
$$y = b_0 + b_1{}^*x$$

$$\text{Salary} = b_0 + b_1{}^*\text{Experience}$$

Simple Linear Regression Intuition - Step 2

Ordinary Least Squares

Simple Linear Regression:



SUM $(y - \hat{y})^2$ -> min

Multiple Linear Regression Intuition - Step 1

Regressions

Simple Linear Regression

$$y = b_0 + b_1 * x_1$$

Multiple Linear Regression

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \dots + b_n * x_n$$

Multiple Linear Regression Intuition - Step 2

A Caveat

Assumptions of a Linear Regression:

1. Linearity
2. Homoscedasticity
3. Multivariate normality
4. Independence of errors
5. Lack of multicollinearity

Multiple Linear Regression Intuition - Step 3

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	151,377.59	443,898.53	California
191,050.39	153,441.51	101,145.55	407,934.54	California
182,901.99	144,372.41	118,671.85	383,199.62	New York
166,187.94	142,107.34	91,391.77	366,168.42	California

Dummy Variables

New York	California
1	0
0	1
0	1
1	0
0	1

$$y = b_0 + b_1{*}x_1 + b_2{*}x_2 + b_3{*}x_3 + b_4{*}D_1$$

Dummy Variable Trap

Profit	R&D Spend	Admin	Marketing	State
192,261.83	165,349.20	136,897.80	471,784.10	New York
191,792.06	162,597.70	[illegible]	[illegible]	[illegible]
191,050.39	153,441.51	[illegible]	[illegible]	[illegible]
182,901.99	144,372.41	[illegible]	[illegible]	[illegible]
166,187.94	142,107.34	[illegible]	[illegible]	[illegible]

$$D_2 = 1 - D_1$$

Dummy Variables

New York	California
1	0
0	1
0	1
1	0
0	1

$$y = b_0 + b_1{*}x_1 + b_2{*}x_2 + b_3{*}x_3 + b_4{*}D_1 + \underline{b_5{*}D_2}$$

Dummy Variable Trap

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State	New York	California
192,261.83	165,349.20	136,897.80	471,784.10	New York	1	0
191,792.06	162,597.70	151,377.59	443,898.53	California	0	1
191,050.39	153,441.51	101,145.55	407,934.54	California	0	1
182,901.99	144,372.41	118,671.85	383,199.62	New York	1	0
166,187.94	142,107.34	91,391.77	366,168.42	California	0	1

$$y = b_0 + b_1{*}x_1 + b_2{*}x_2 + b_3{*}x_3 \quad + b_4{*}D_1 + \underline{b_5{*}D_2}$$

Dummy Variable Trap

Dummy Variables

Profit	R&D Spend	Admin	Marketing	State	New York	California
192,261.83	165,349.20	136,897.80	471,784.10	New York	1	0
191,792.06	162,597.70	151,377.59	443,898.53	California	0	1
191,050.39	153,441.51	101,145.55	407,934.54	California	0	1
182,901.99	144,372.41	118,671.85	383,199.62	New York	1	0
166,187.94	142,107.34	91,391.77	366,168.42	California	0	1

$$y = b_0 + b_1{}^*x_1 + b_2{}^*x_2 + b_3{}^*x_3 \qquad + b_4{}^*D_1 + \cancel{b_5{}^*D_2}$$

Always omit one dummy variable

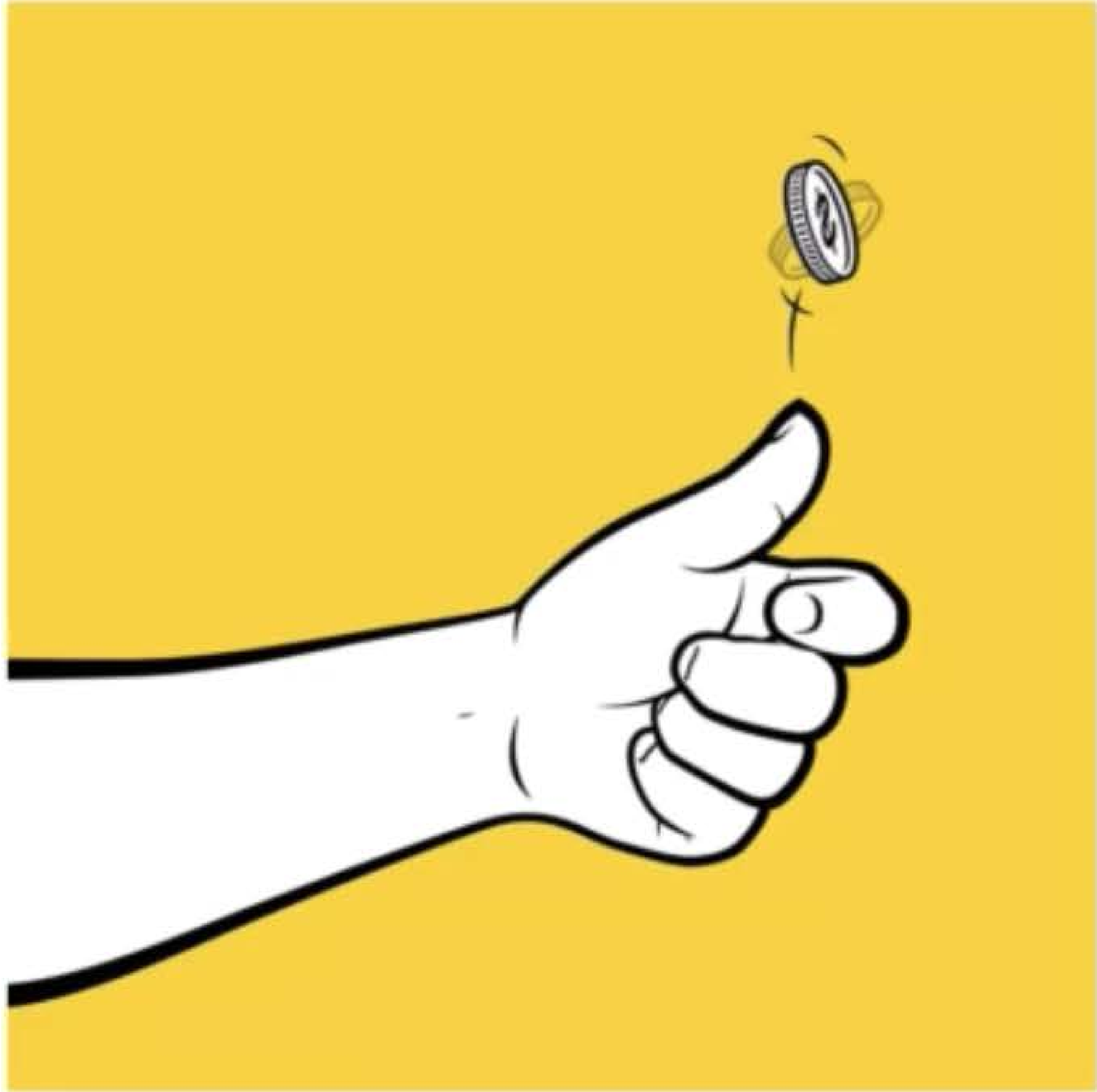
1.75

Statistics for Business Analytics and Data Science A-Z™

Statistical Significance

53. Understanding the P-Value

Statistical Significance



H_0: This is a fair coin
H_1: This is not a fair coin

P-Value

0.5
0.25
0.12
0.06

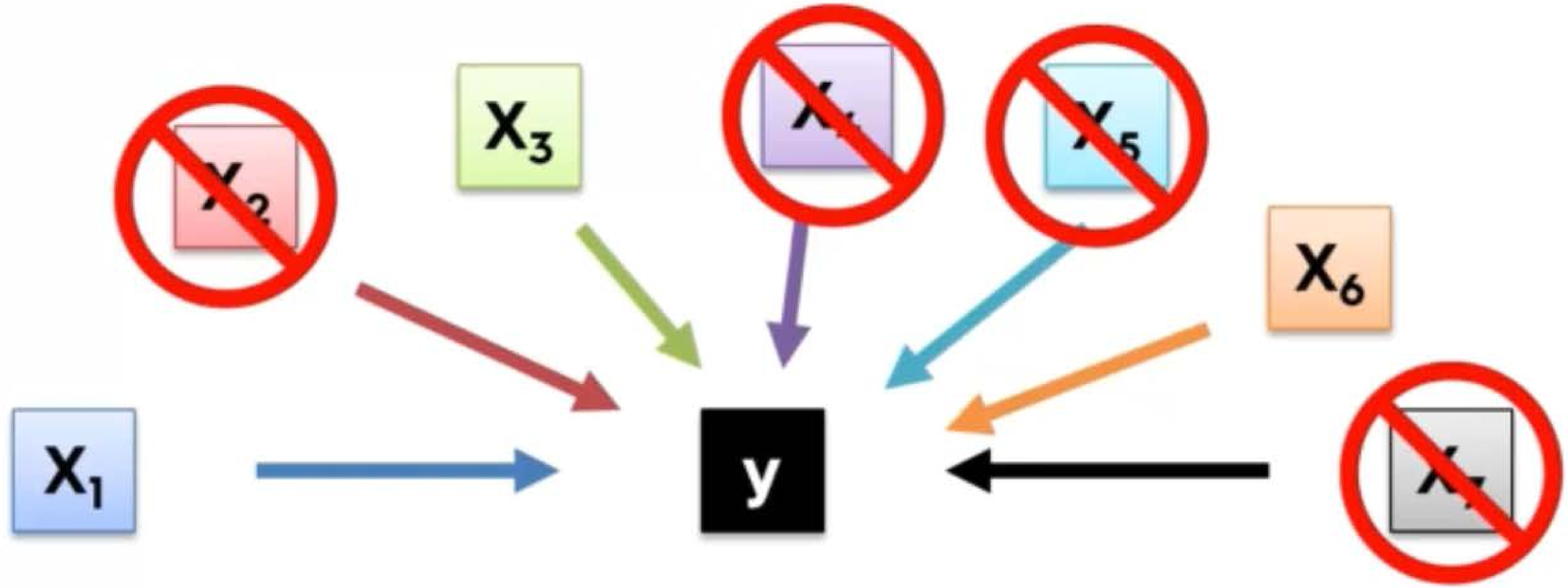
$\alpha = 0.05$

0.03
0.01

Multiple Linear Regression Intuition - Step 5

Building A Model (Step-By-Step)

Building A Model

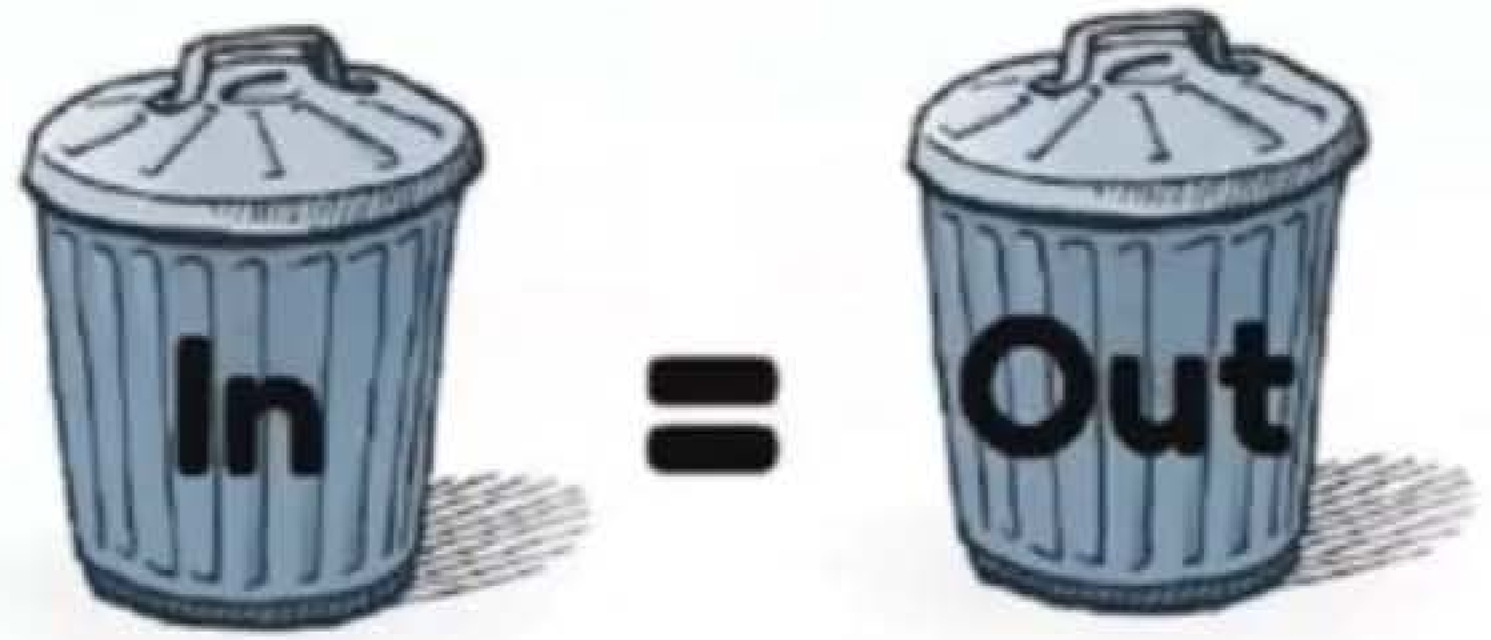


Why?

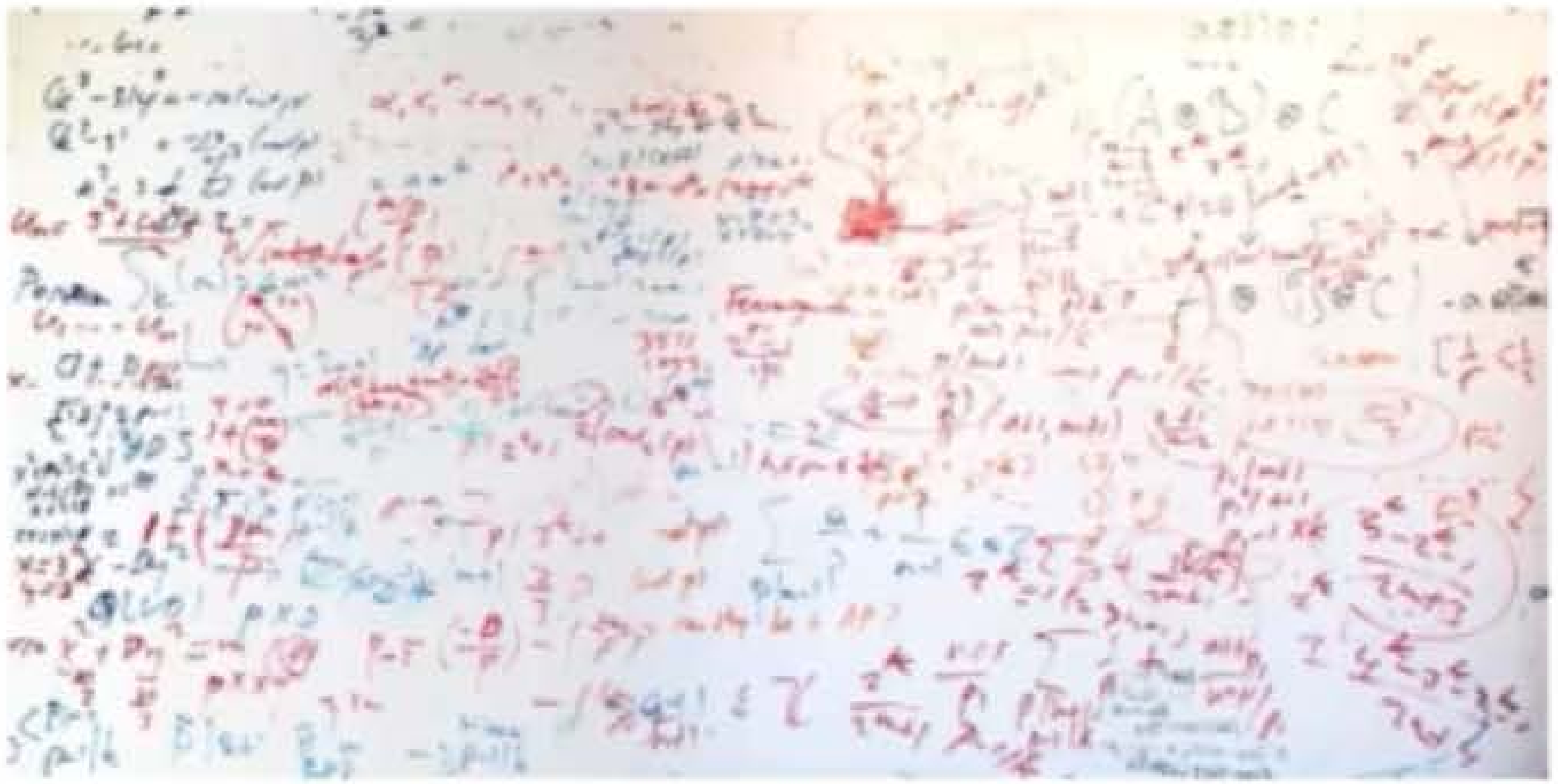
54. Multiple Linear Regression Intuition - Step 5

Building A Model

1)



2)



Playback Rate

1.75x 1:34 / 15:41

Building A Model

5 methods of building models:

1. All-in
2. Backward Elimination
3. Forward Selection
4. Bidirectional Elimination
5. Score Comparison

Stepwise Regression

Building A Model

"All-in" – cases:

- Prior knowledge; OR
- You have to; OR
- Preparing for Backward Elimination

Building A Model

Backward Elimination

STEP 1: Select a significance level to stay in the model (e.g. SL = 0.05)

STEP 2: Fit the full model with all possible predictors

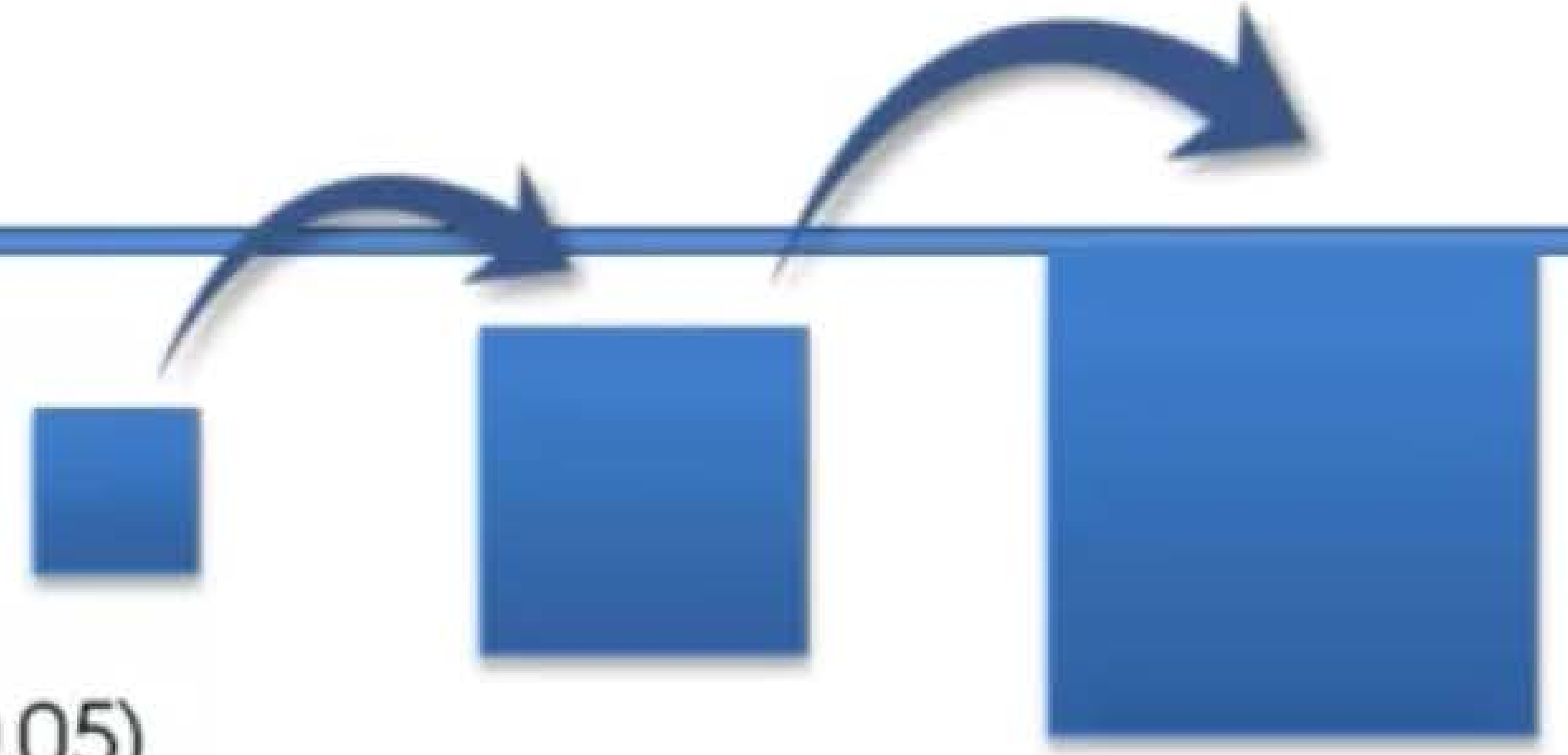
STEP 3: Consider the predictor with the <u>highest</u> P-value. If ~~P > SL,~~ go to STEP 4, otherwise go to FIN

STEP 4: Remove the predictor

STEP 5: Fit model without this variable*

FIN: Your Model Is Ready

Building A Model



Forward Selection

STEP 1: Select a significance level to enter the model (e.g. SL = 0.05)

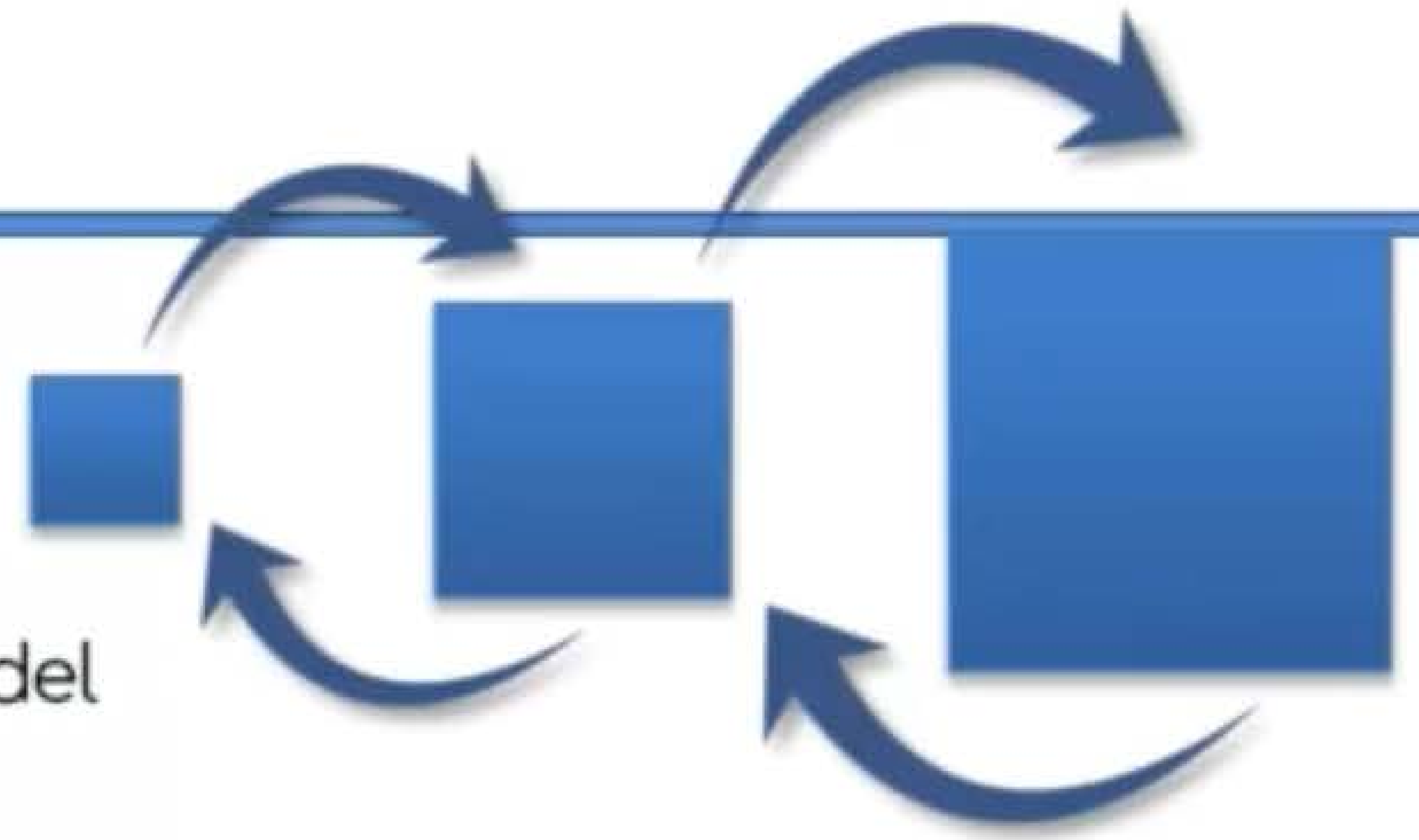
STEP 2: Fit all simple regression models $\mathbf{y} \sim \mathbf{x_n}$ Select the one with the lowest P-value

STEP 3: Keep this variable and fit all possible models with one extra predictor added to the one(s) you already have

STEP 4: Consider the predictor with the <u>lowest</u> P-value. If ~~P < SL~~, go to STEP 3, otherwise go to FIN

FIN: Keep the previous model

Building A Model



Bidirectional Elimination

STEP 1: Select a significance level to enter and to stay in the model
e.g.: SLENTER = 0.05, SLSTAY = 0.05

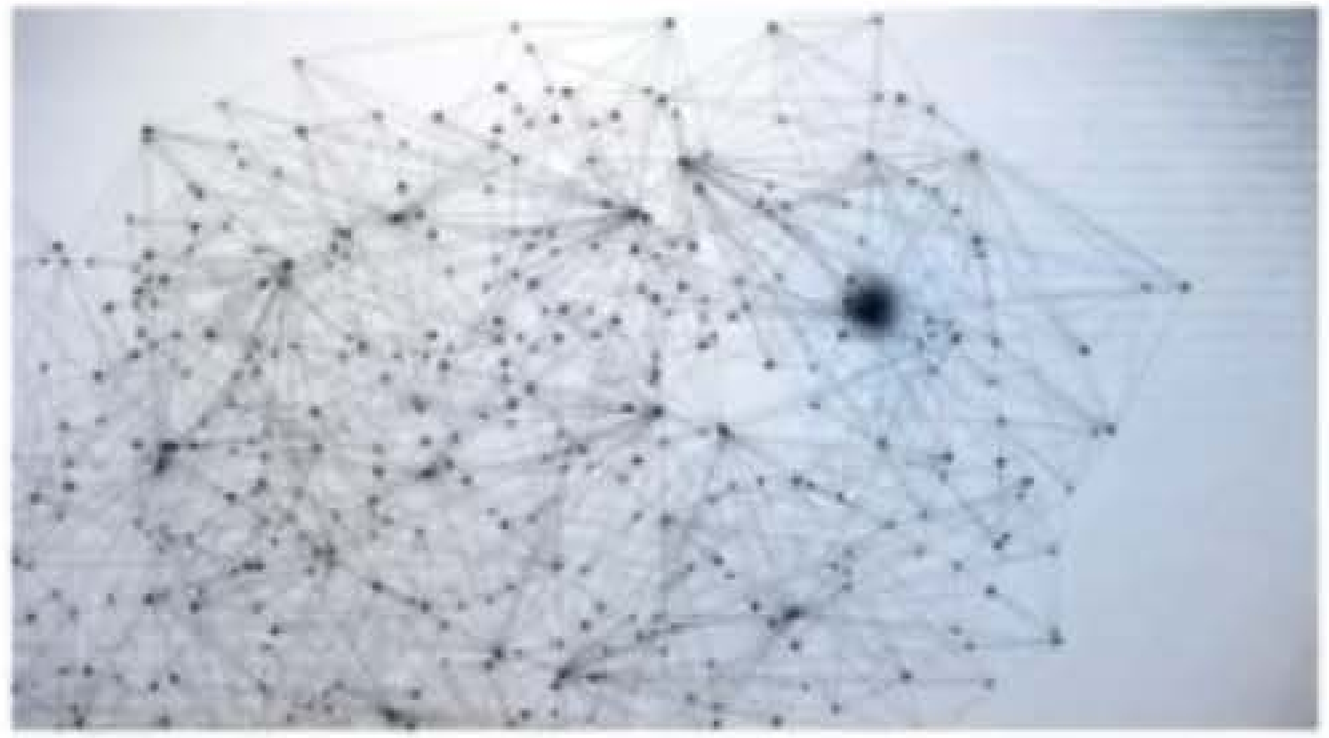
STEP 2: Perform the next step of Forward Selection (new variables must have: P < SLENTER to enter)

STEP 3: Perform ALL steps of Backward Elimination (old variables must have P < SLSTAY to stay)

STEP 4: No new variables can enter and no old variables can exit

FIN: Your Model Is Ready

Building A Model



All Possible Models

STEP 1: Select a criterion of goodness of fit (e.g. Akaike criterion)

STEP 2: Construct All Possible Regression Models: 2^N-1 total combinations

STEP 3: Select the one with the best criterion

FIN: Your Model Is Ready

Example:
10 columns means
1,023 models

Building A Model

5 methods of building models:

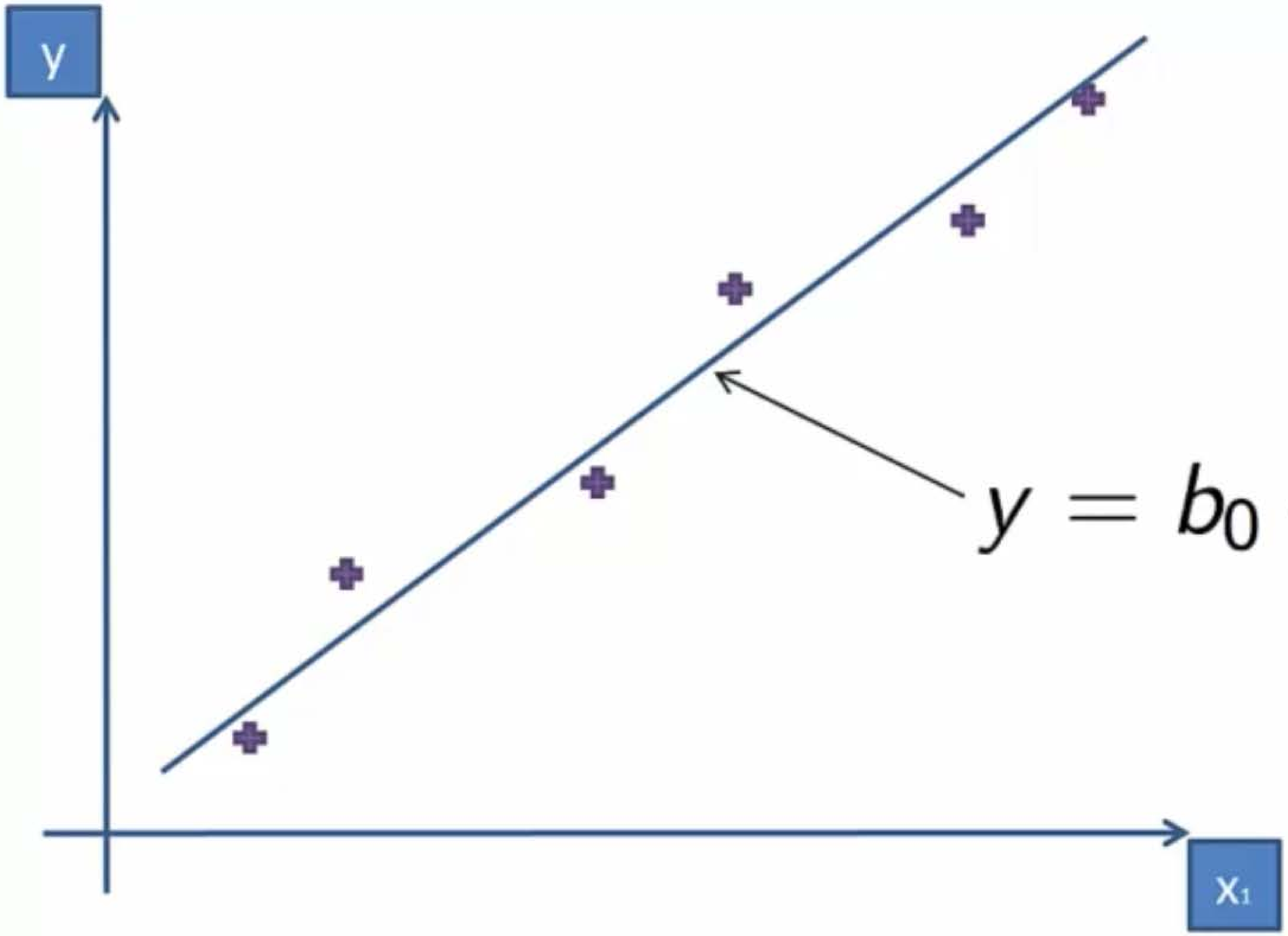
1. All-in
2. Backward Elimination
3. Forward Selection
4. Bidirectional Elimination
5. Score Comparison

Polynomial Regression

Regressions

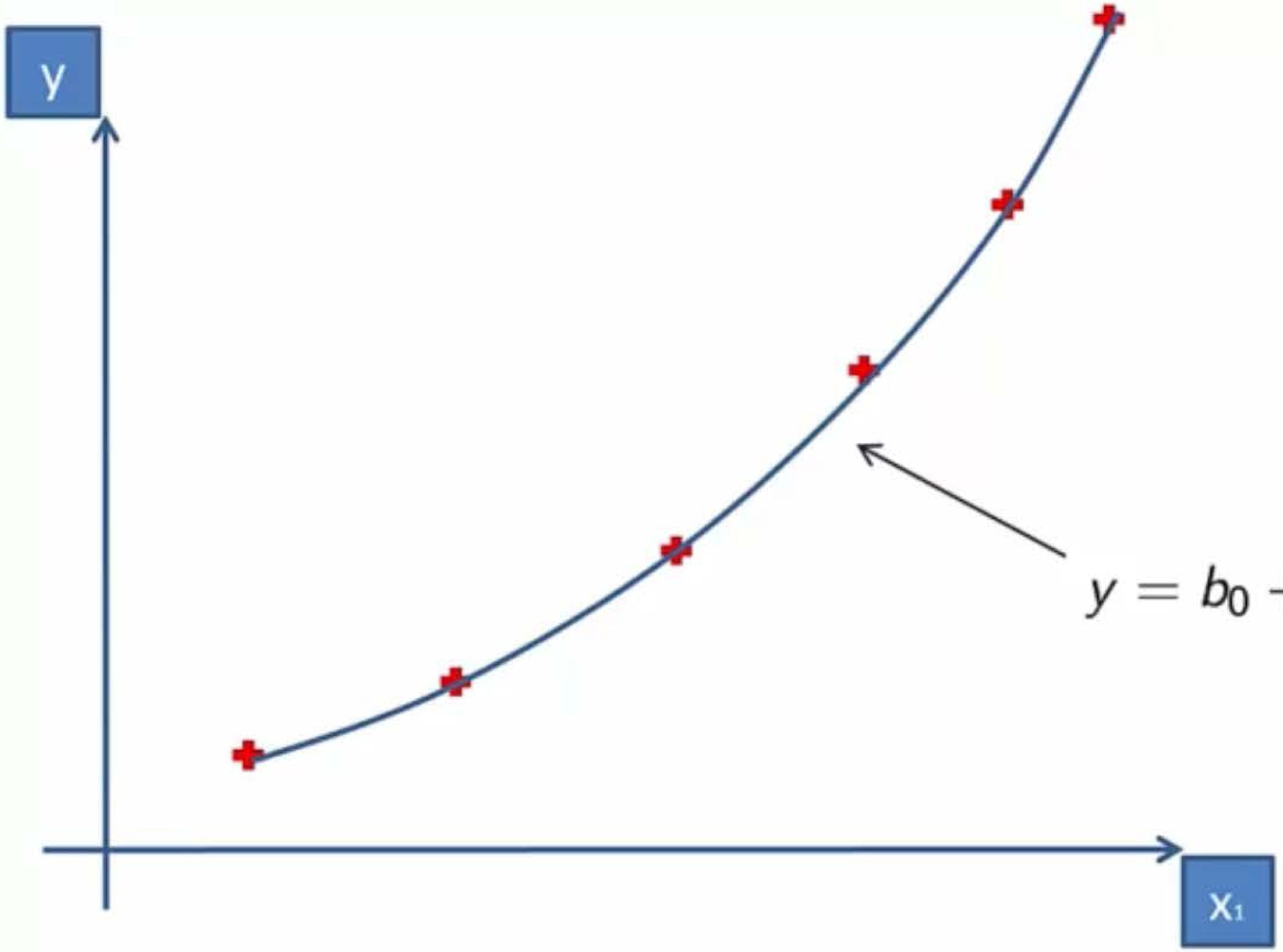
Simple Linear Regression	$y = b_0 + b_1x_1$
Multiple Linear Regression	$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$
Polynomial Linear Regression	$y = b_0 + b_1x_1 + b_2x_1^2 + \ldots + b_nx_1^n$

Simple Linear Regression



$$y = b_0 + b_1 x_1$$

Polinomial Regression



$$y = b_0 + b_1 x_1 \boxed{+ b_2 x_1^2}$$

Polinomial Regression

One Question: Why “Linear”?

Playback Rate

1.75 × 2:51 / 5:08

Polinomial Regression

Polynomial Linear Regression

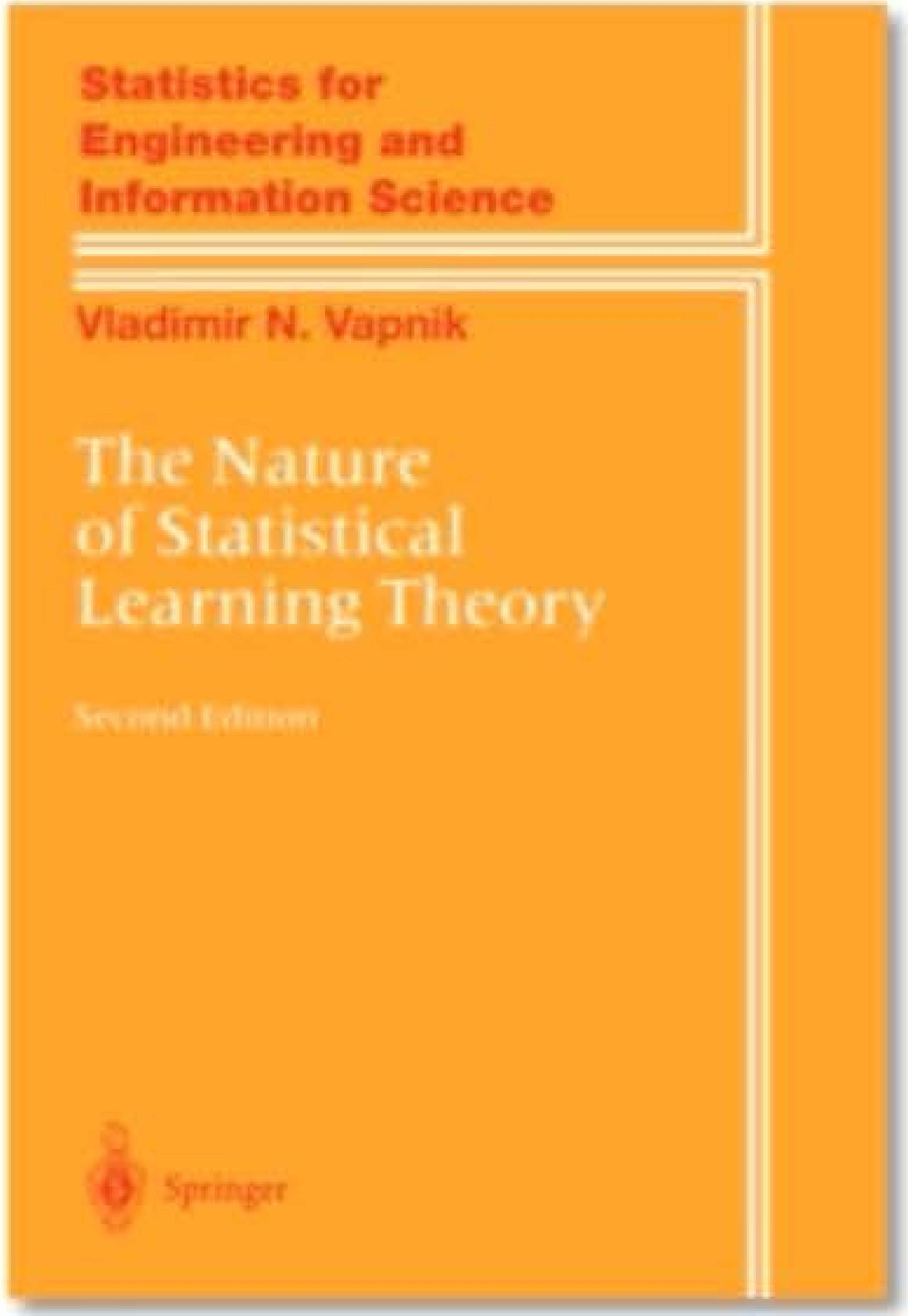
$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

Playback Rate

SVR Intuition

SVR Intuition

Vladimir Vapnik



1992

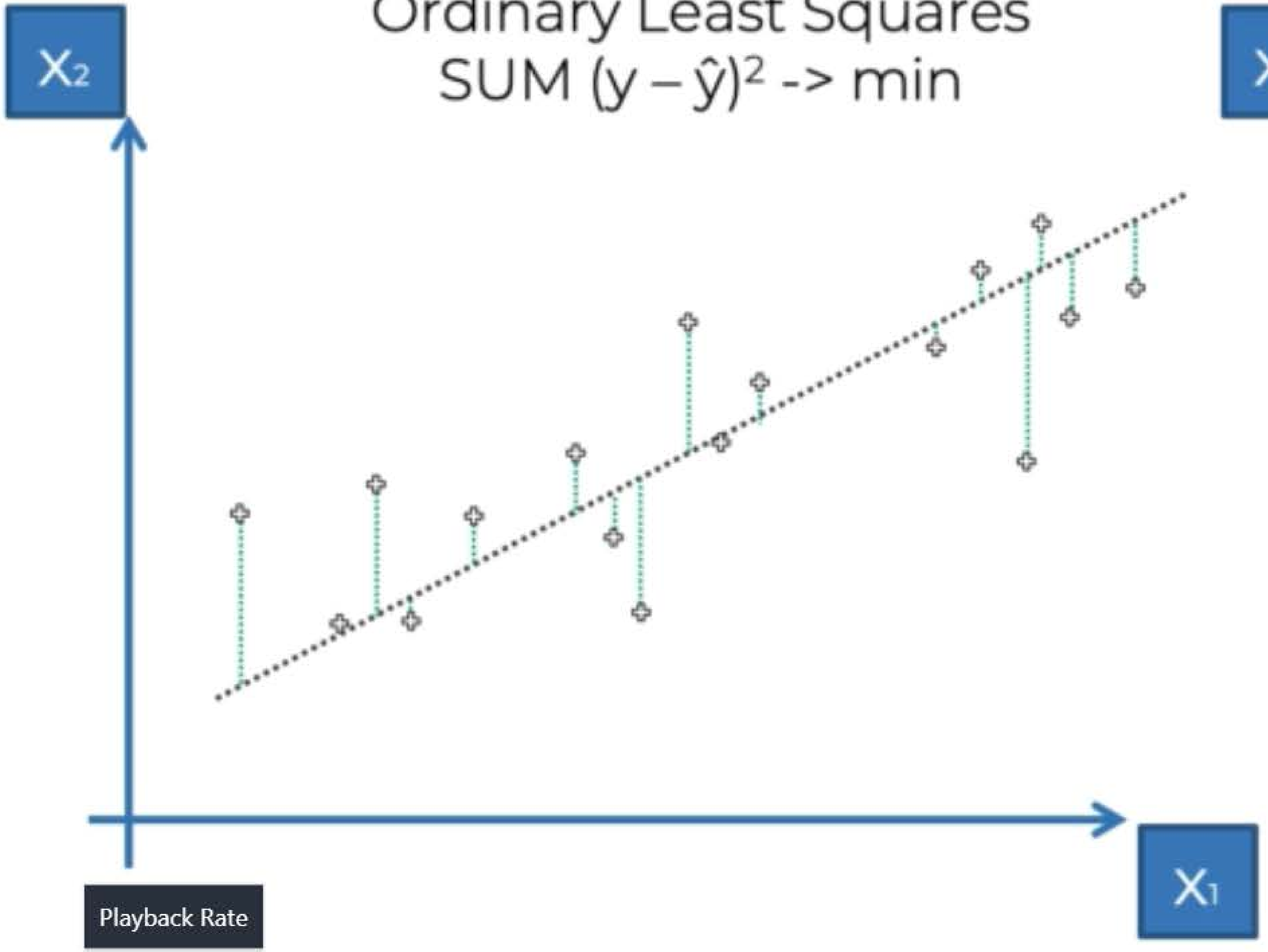
79. SVR Intuition (Updated!)

SVR Intuition

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \rightarrow min$$

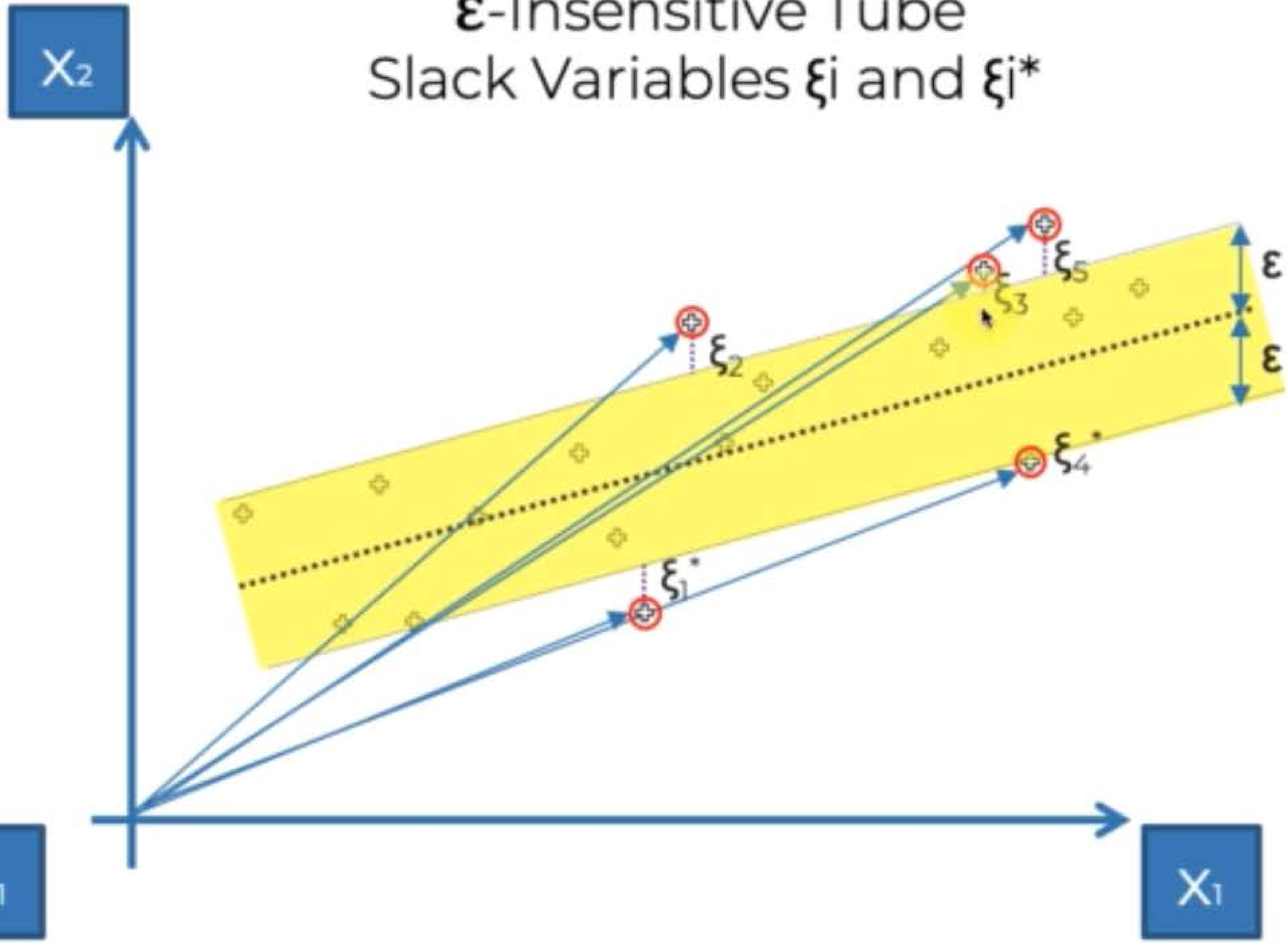
Ordinary Least Squares

SUM $(y - \hat{y})^2$ -> min



ε-Insensitive Tube

Slack Variables ξi and ξi*



Playback Rate

1.75 × 6:47 / 8:09

SVR Intuition

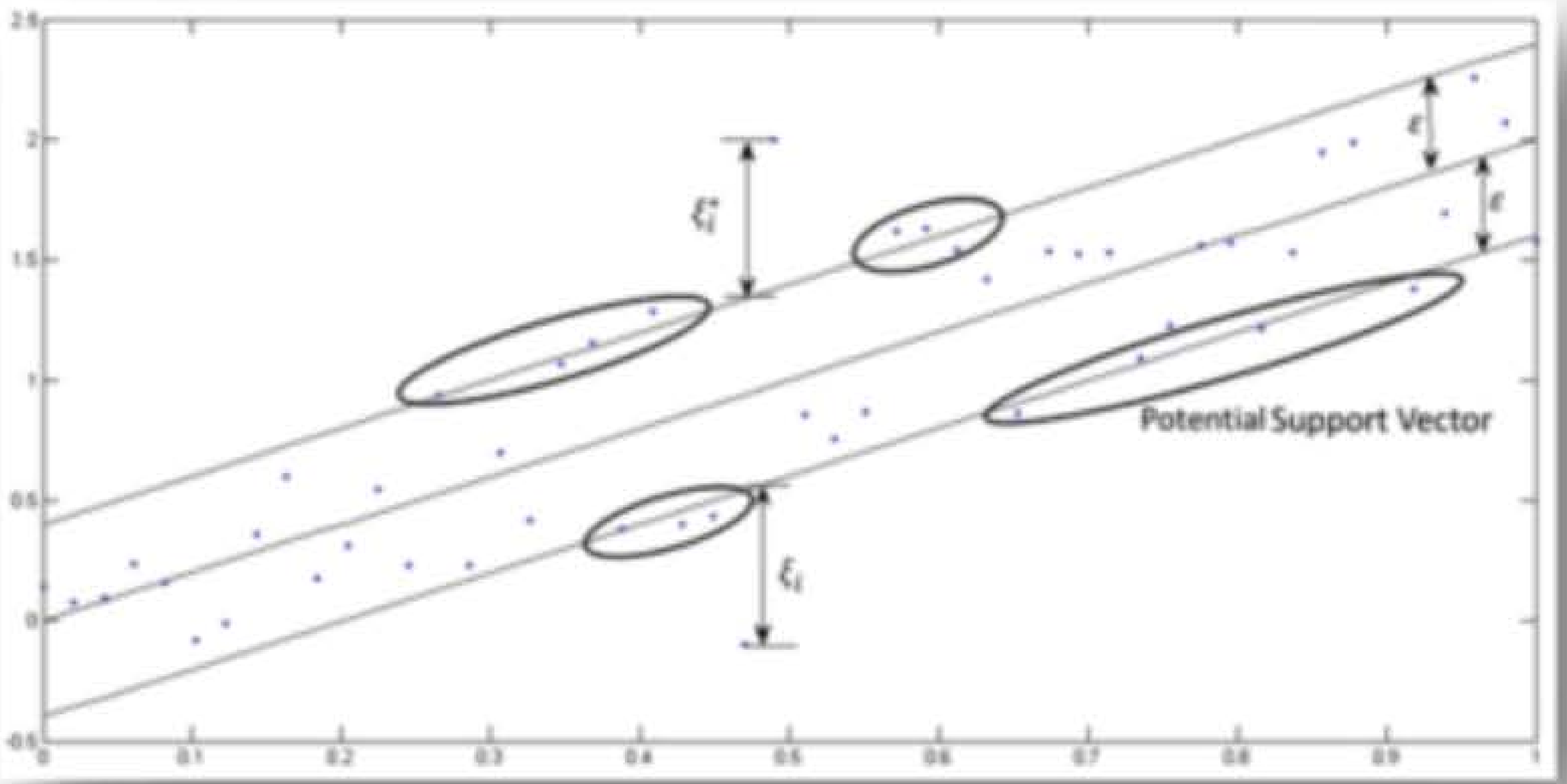
Additional Reading:

Chapter 4 – Support Vector Regression (from: Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers)

By Mariette Awad & Rahul Khanna (2015)

Link:

https://core.ac.uk/download/pdf/81523322.pdf



Heads-up about Non-Linear SVR

Heads-up about Non-Linear SVR

Section on SVM:

- SVM Intuition

Section on Kernel SVM:

- Kernel SVM Intuition
- Mapping to a higher dimension
- The Kernel Trick
- Types of Kernel Functions
- Non-linear Kernel SVR

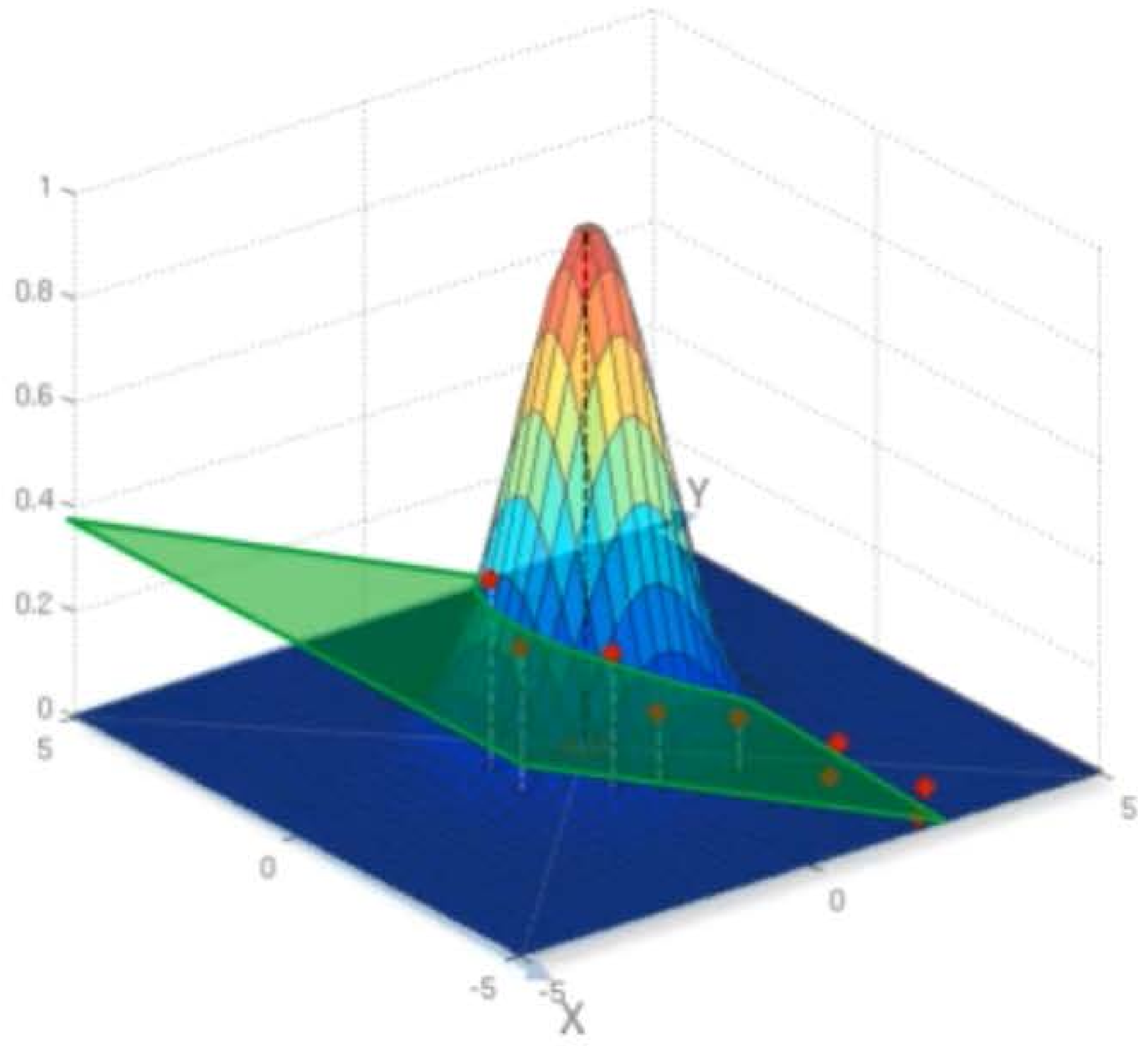


Image source: http://www.cs.toronto.edu/~duvenaud/cookbook/index.html

84. SVR in Python - Step 3

Kernel Functions-Introduction to SVM Kernel & Examples

BY DATAFLAIR TEAM · UPDATED · NOVEMBER 16, 2018

1. Objective

In our previous Machine Learning blog we have discussed about SVM (Support Vector Machine) in Machine Learning. Now we are going to provide you a detailed description of SVM Kernel and Different Kernel Functions and its examples such as linear, nonlinear, polynomial, Gaussian kernel, Radial basis function (RBF), sigmoid etc.

Kernel Functions-Introduction to SVM Kernel & Examples

2. SVM Kernel Functions

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example ***linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.***

Introduce Kernel functions for sequence data, graphs, text, images, as well as vectors. The most used type of kernel function is **RBF.** Because it has localized and finite response along the entire x-axis.

The kernel functions return the inner product between two points in a suitable feature space. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.

3. Kernel Rules

Decision Tree Intuition

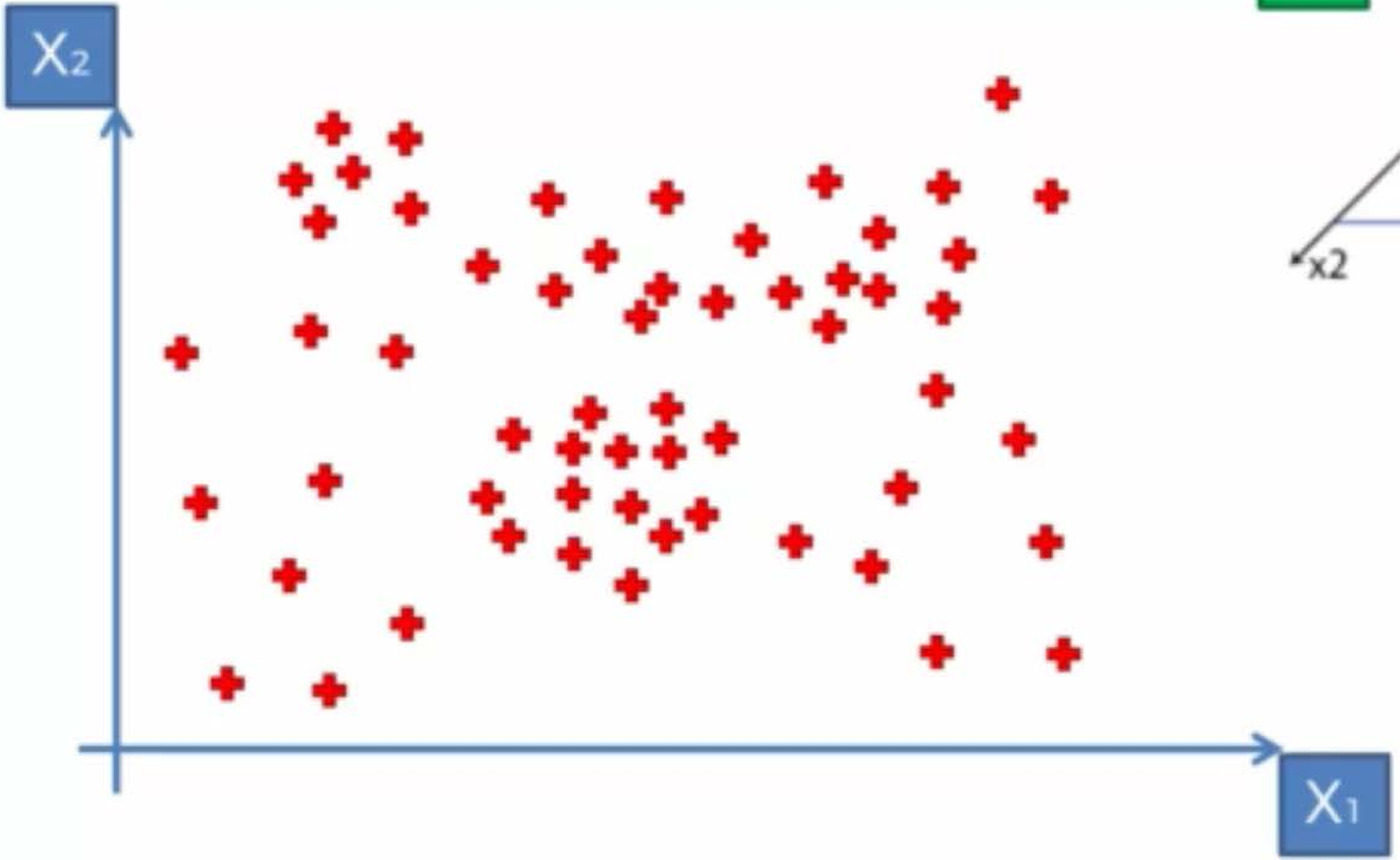
Decision Tree Intuition

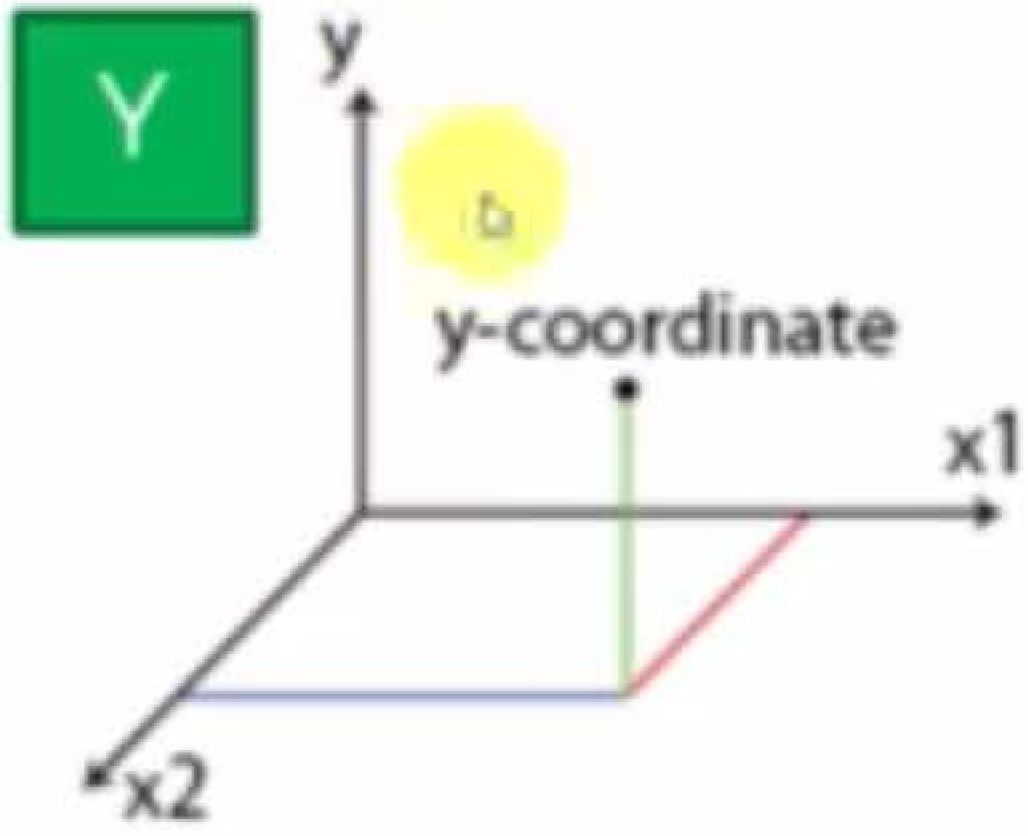
CART

Classification Trees

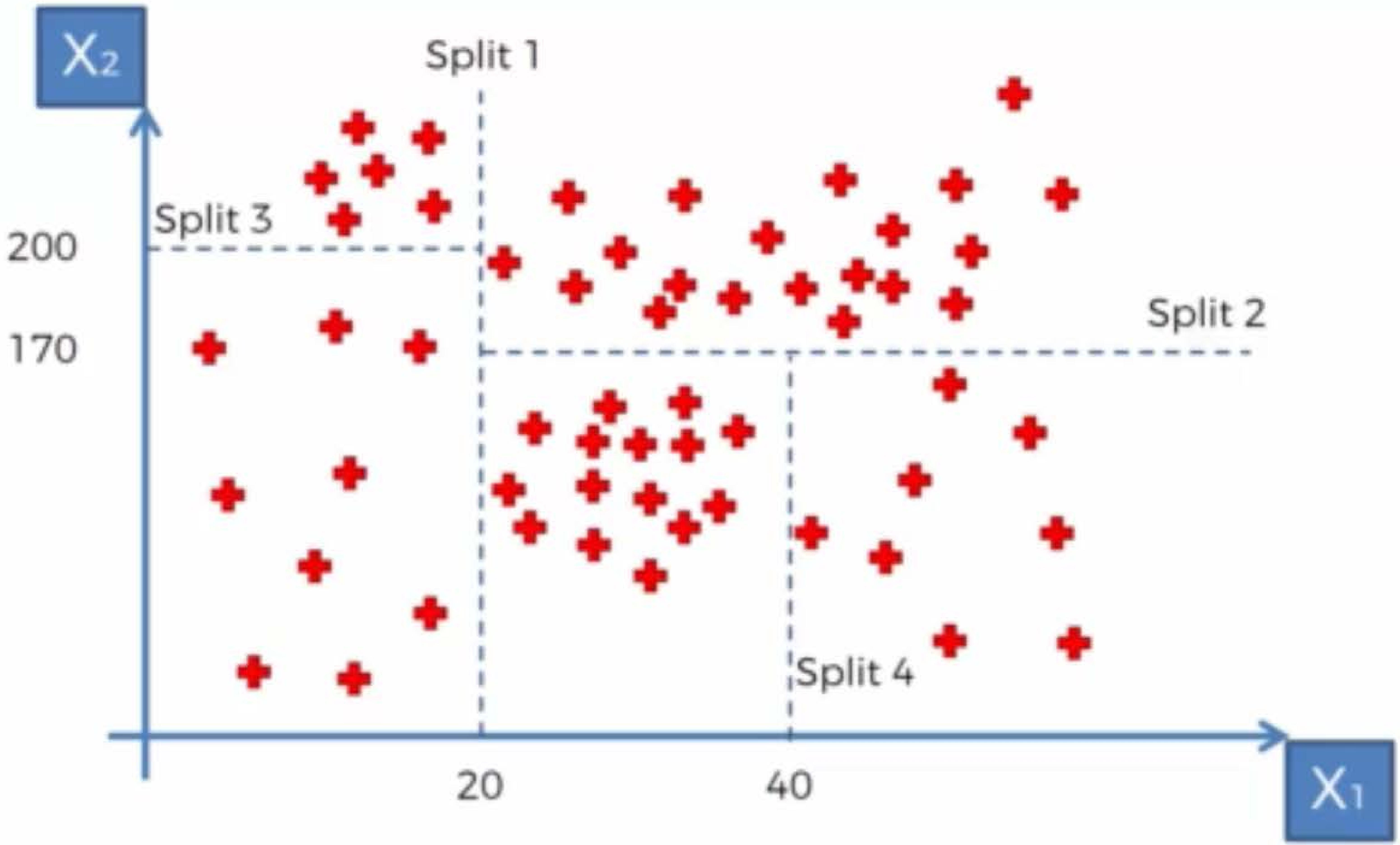
Regression Trees

Decision Tree Intuition





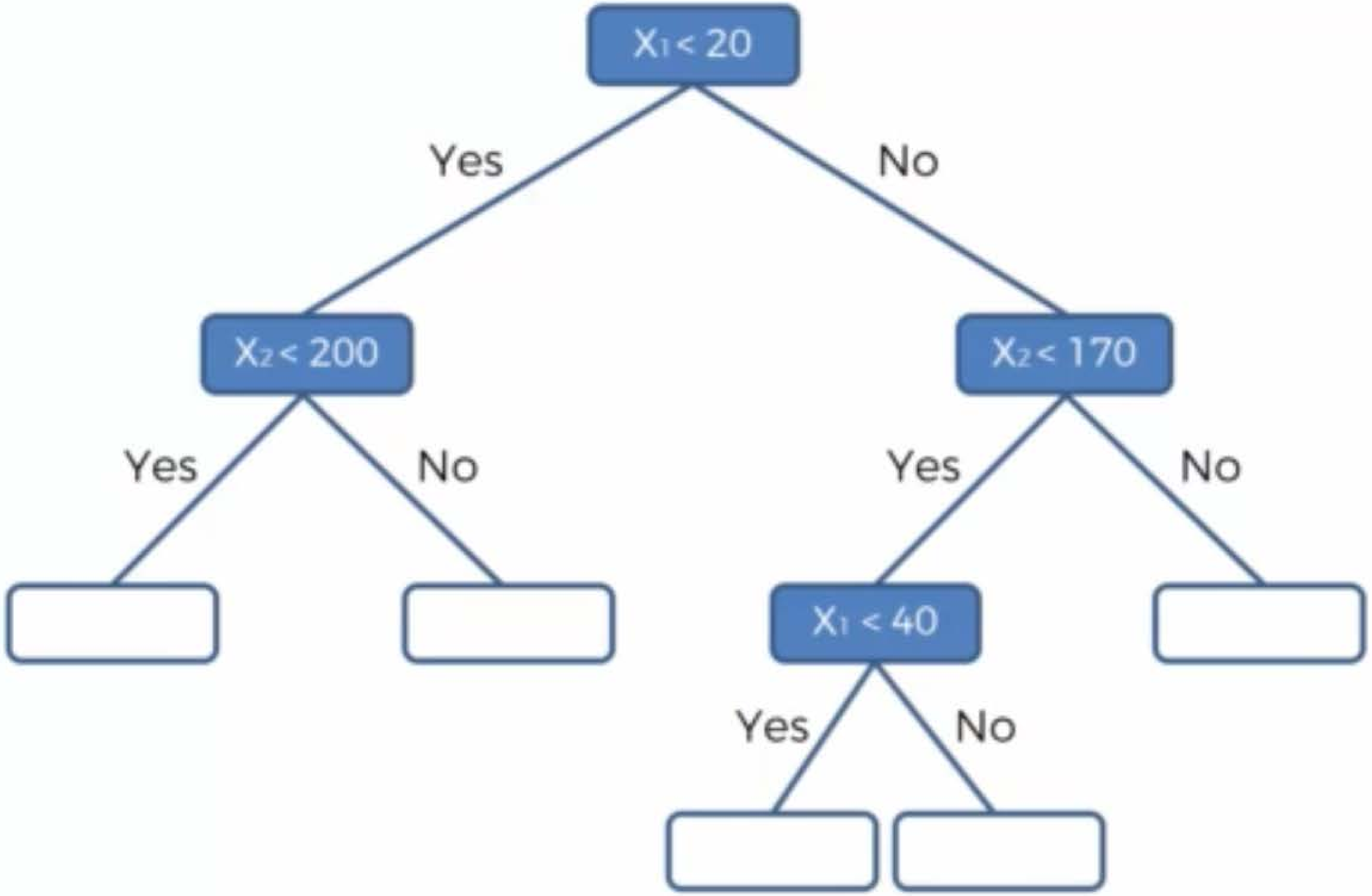
Decision Tree Intuition



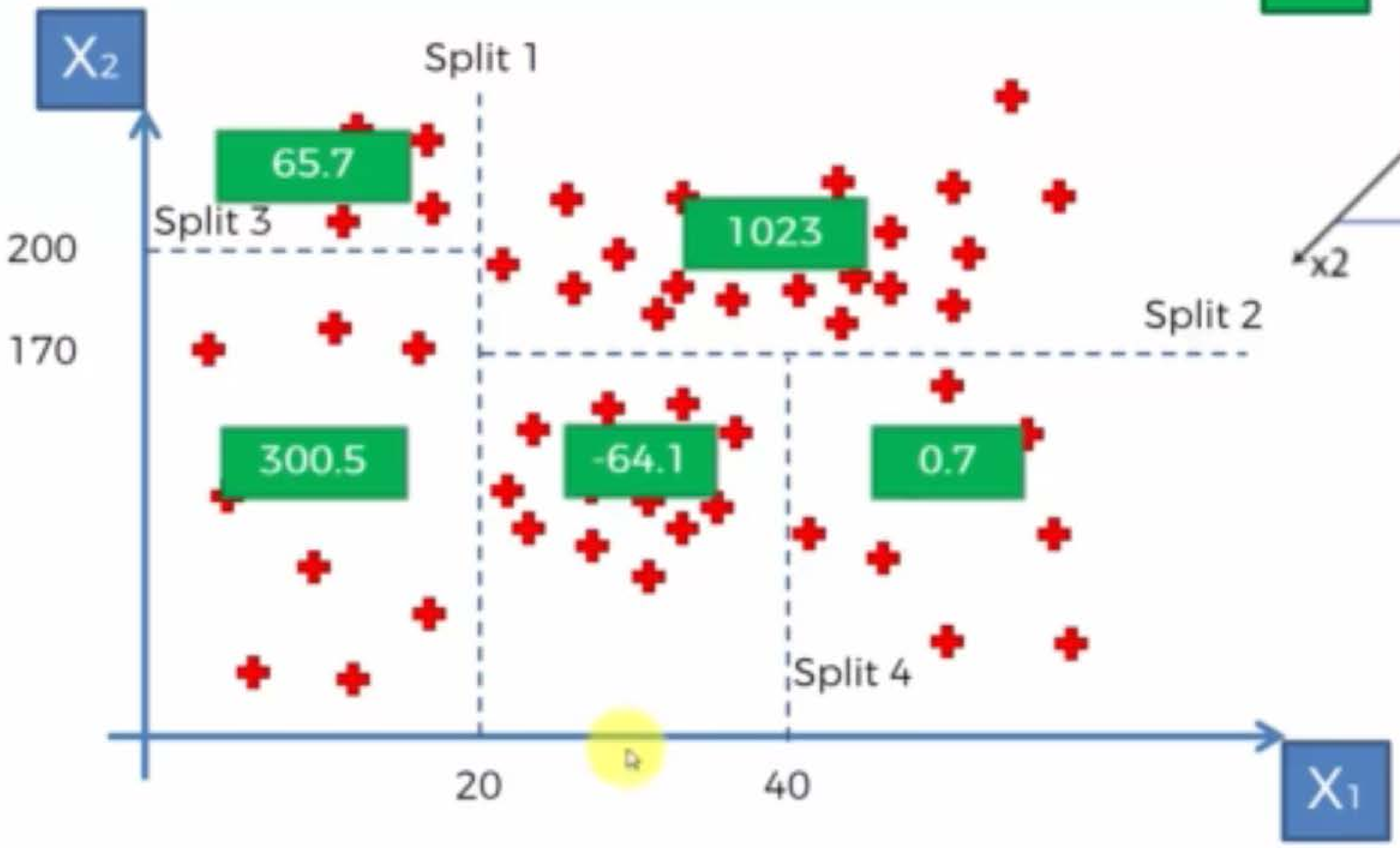
Decision Tree Intuition

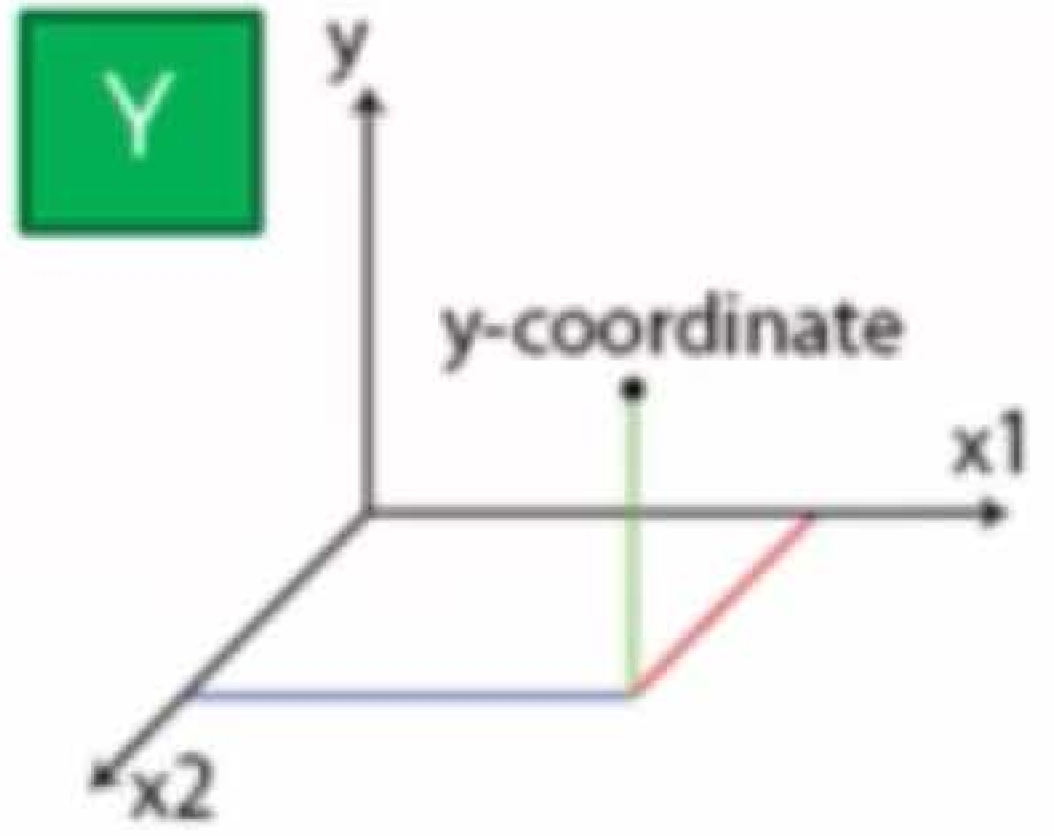
Rewind…

Decision Tree Intuition

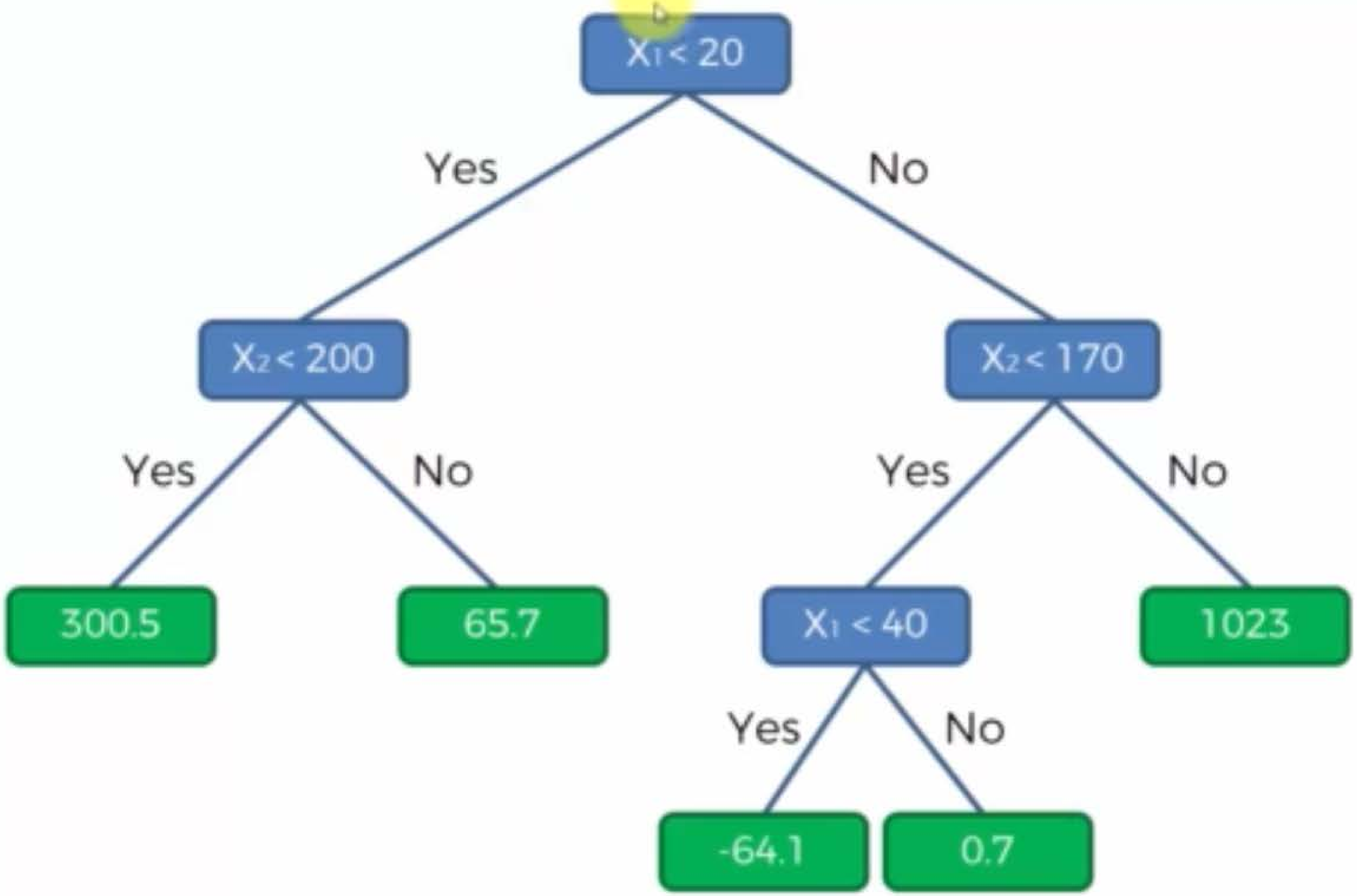


Decision Tree Intuition





Decision Tree Intuition



Random Forest Intuition

Ensemble Learning

Random Forest Intuition

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 value of Y to for the data point in question, and assign the new data point the average across all of the predicted Y values.

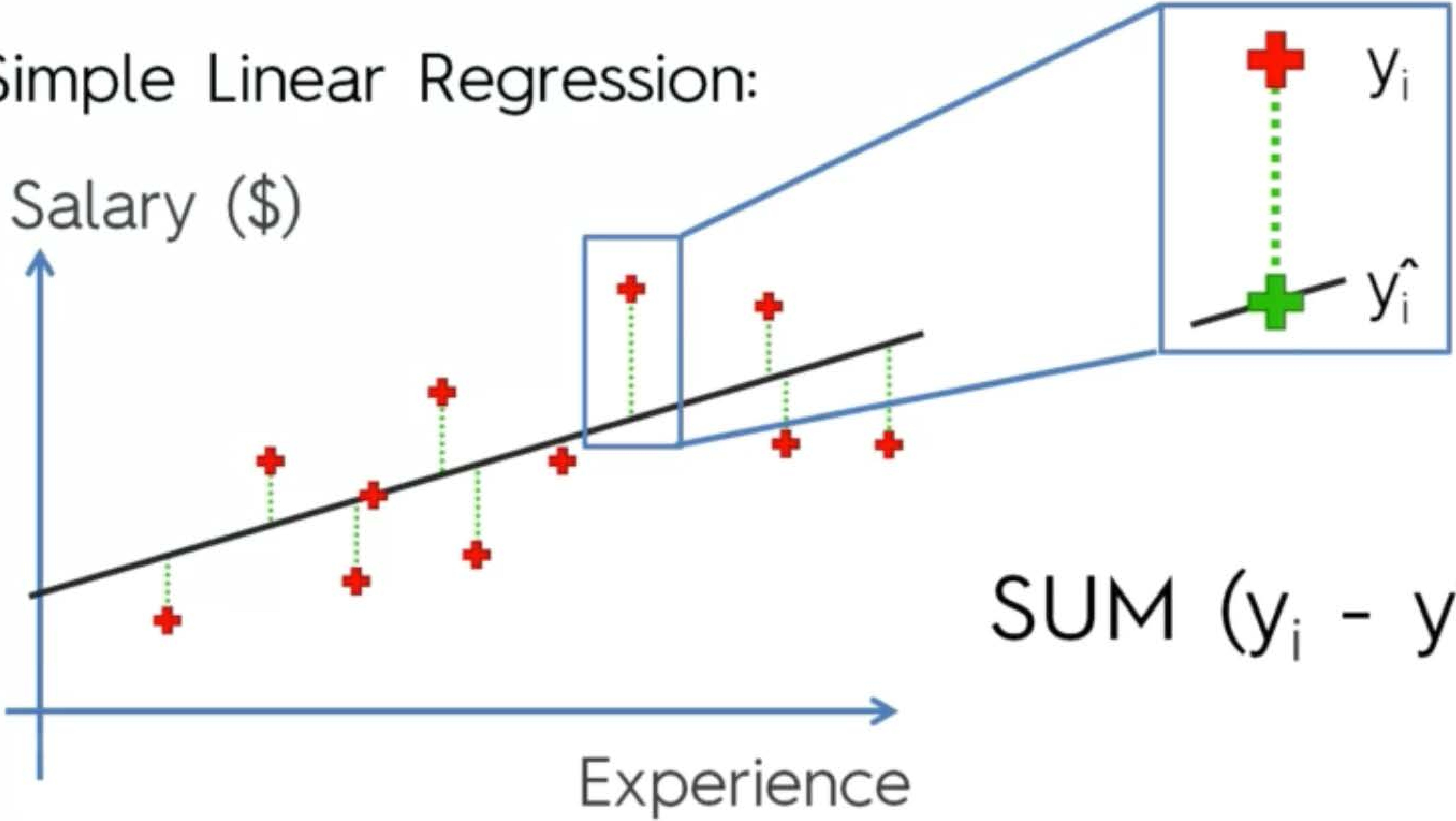
Random Forest Intuition

Random Forest Intuition

R Squared Intuition

R Squared

Simple Linear Regression:

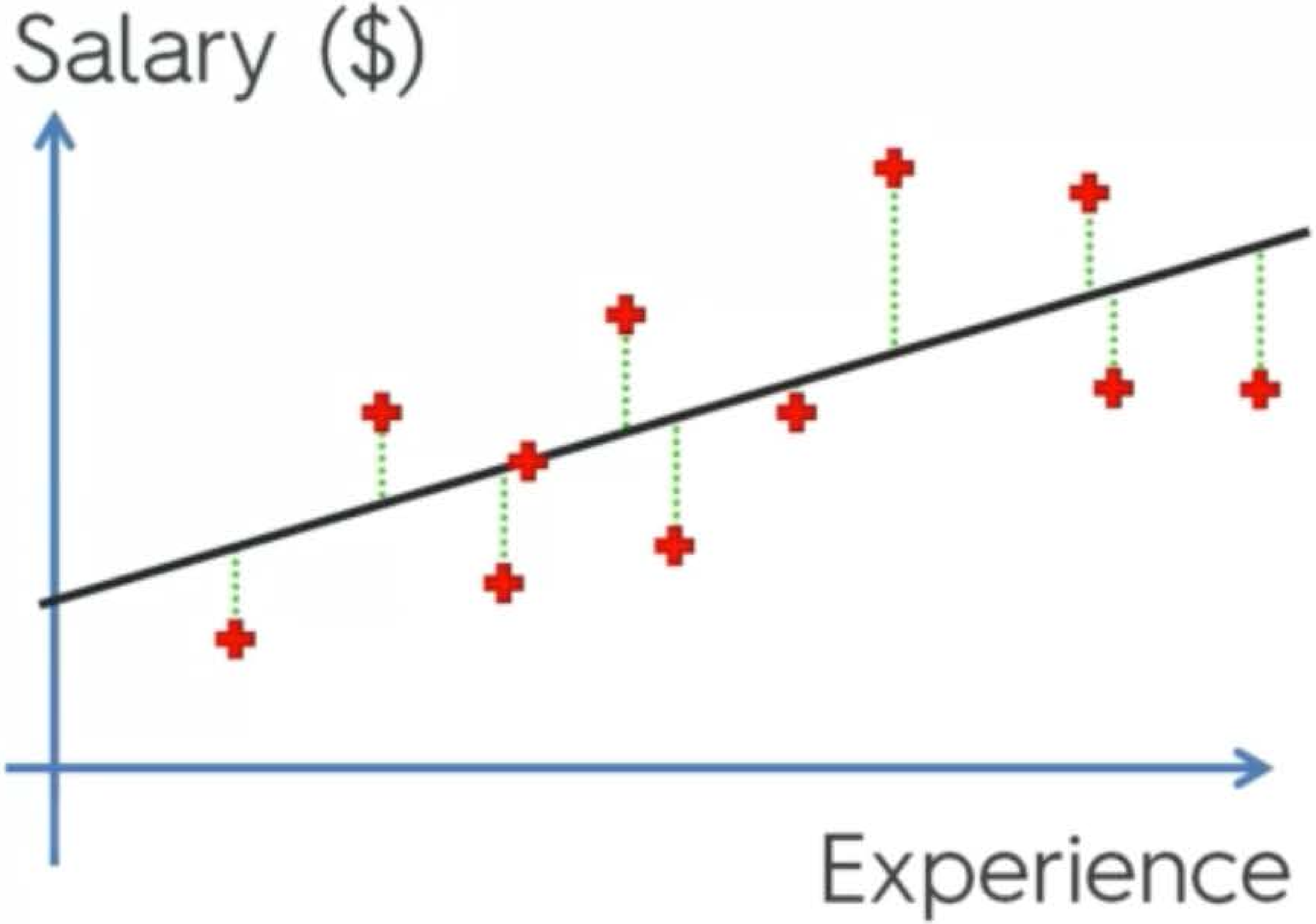


$$\text{SUM } (y_i - \hat{y_i})^2 \text{-> min}$$

R Squared

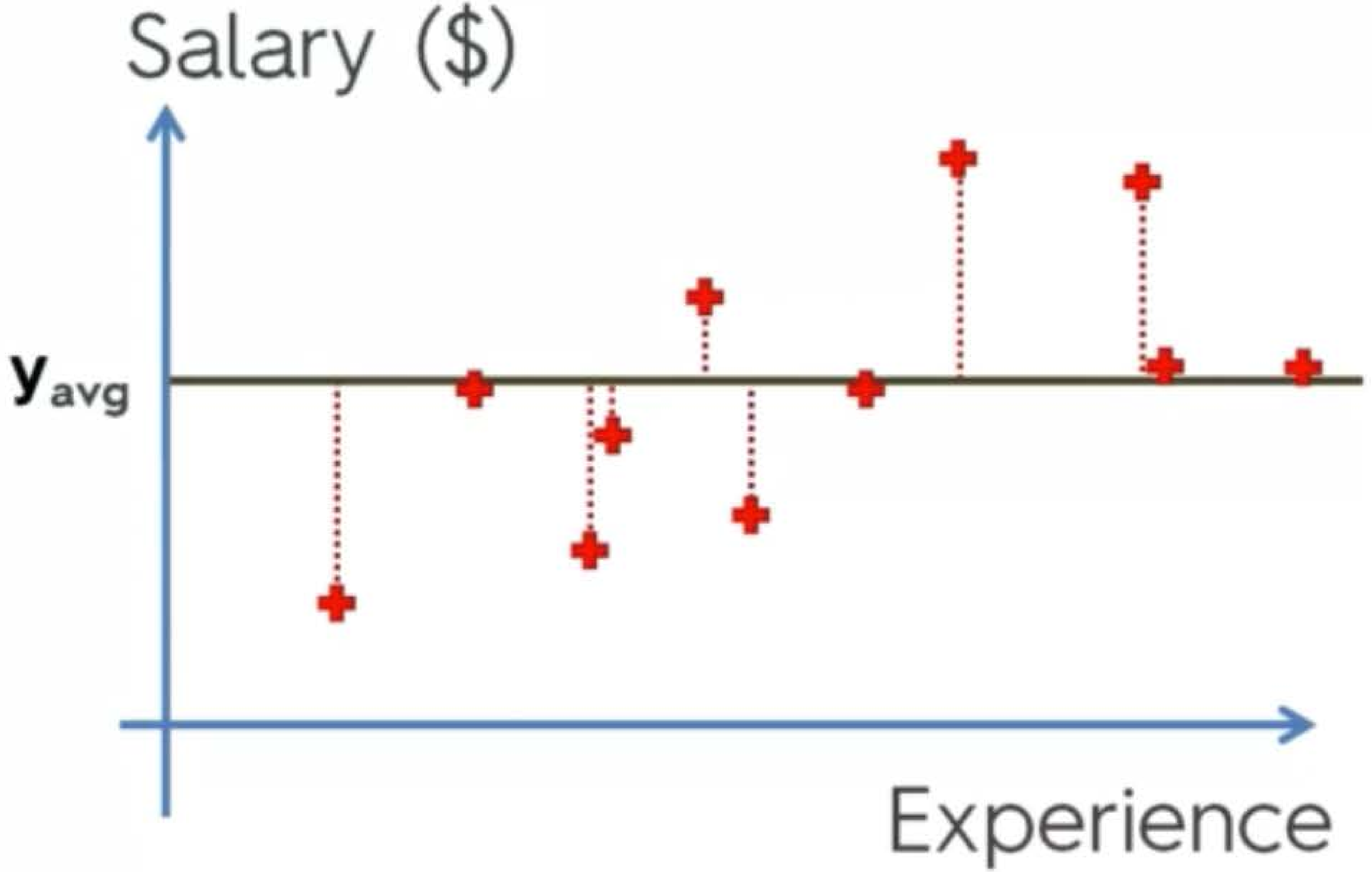
Simple Linear Regression:

$$SS_{res} = SUM\ (y_i - \hat{y}_i)^2$$



R Squared

Simple Linear Regression:



$$SS_{res} = SUM\ (y_i - \hat{y_i})^2$$

$$SS_{tot} = SUM\ (y_i - y_{avg})^2$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Adjusted R^2

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R^2 – Goodness of fit

(greater is better)

$y = b_0 + b_1{*}x_1$

Problem:

$y = b_0 + b_1{*}x_1 + b_2{*}x_2 \leftarrow \boxed{+ b_3{*}x_3}$

SS_{res} -> Min $\longrightarrow$ R^2 will never decrease

Adjusted R^2

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$\text{Adj } R^2 = 1 - (1 - R^2)\frac{n - 1}{n - p - 1}$$

p - number of regressors

n - sample size