Artificial Intellligence: An Introduction

Wei Wang(王伟)

Engineering Research Centre of Applied Technology on Machine Translation and Artificial Intelligence , Macao Polytechnic University

weiwang@mpu.edu.mo; 匯智樓 (WUI CHI)-5/F, N56

Jan. 25, 2026

Outline

Reflex-based Models

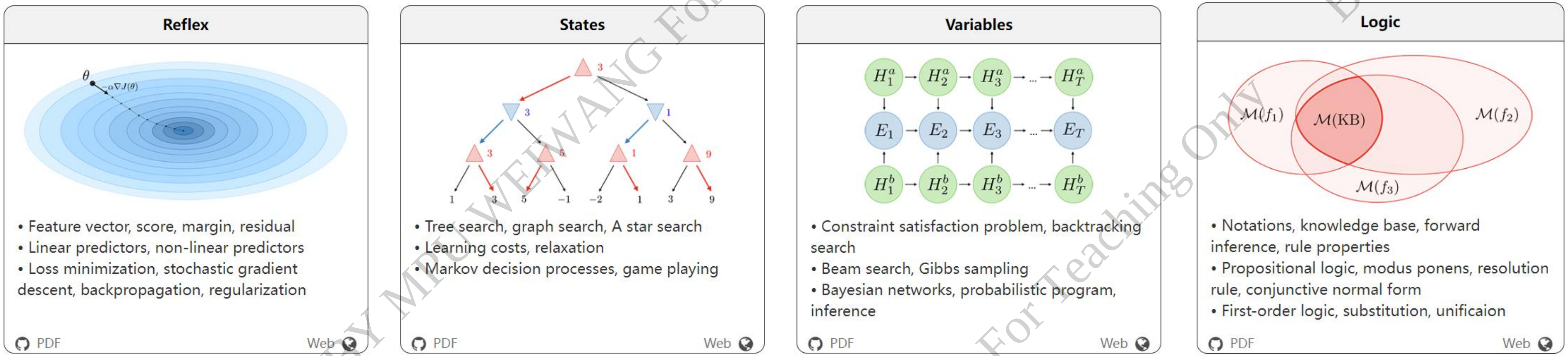
- Supervised Learning

Regression

AI Models Types

AI models refer to different approaches or frameworks that are used to represent and solve problems in the field of AI.

These models provide a structured way to understand and analyze complex systems and make intelligent decisions.



θ
$-\alpha\nabla J(\theta)$
H_1^a H_2^a H_3^a H_T^a
E_1 E_2 E_3 E_T
H_1^b H_2^b H_3^b H_T^b
$\mathcal{M}(f_1)$
$\mathcal{M}(\mathrm{KB})$
$\mathcal{M}(f_2)$
$\mathcal{M}(f_3)$

Reflex

- Feature vector, score, margin, residual
- Linear predictors, non-linear predictors
- Loss minimization, stochastic gradient descent, backpropagation, regularization

PDF Web

States

- Tree search, graph search, A star search
- Learning costs, relaxation
- Markov decision processes, game playing

PDF Web

Variables

- Constraint satisfaction problem, backtracking search
- Beam search, Gibbs sampling
- Bayesian networks, probabilistic program, inference

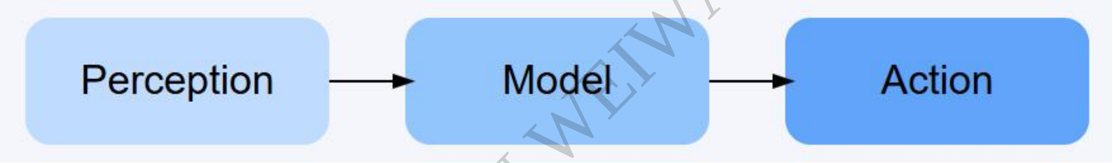
PDF Web

Logic

- Notations, knowledge base, forward inference, rule properties
- Propositional logic, modus ponens, resolution rule, conjunctive normal form
- First-order logic, substitution, unificaion

PDF Web

Source: https://stanford.edu/~shervine/teaching/cs-221/



Perception → Model → Action

AI models define how an agent perceives, reasons, and acts.

Different models suit different environments and tasks.

AI Models Types

1. Logic-based models

- Symbolic representation of classes of objects.
- Deductive Reasoning.
- **Apps**: Question Answering Systems, Natural Language Understanding, Expert system
- **Options**: Propositional Logic , First-Order Logic, Knowledge Base.

2. States-based models

- Solutions are defined as a sequence of steps.
- Model a task as a graph of states and a solution as a path in the graph.
- A state captures all of the relevant information about the past in order to act in the future.
- **Apps**: Navigation and Games.
- **Options**: Tree Search (Breadth-first search, Depth-first search, and Iterative deepening), Graph search (Dynamic programming), Markov decision processes, Game playing

3. Variables-based models (Uncertainty)

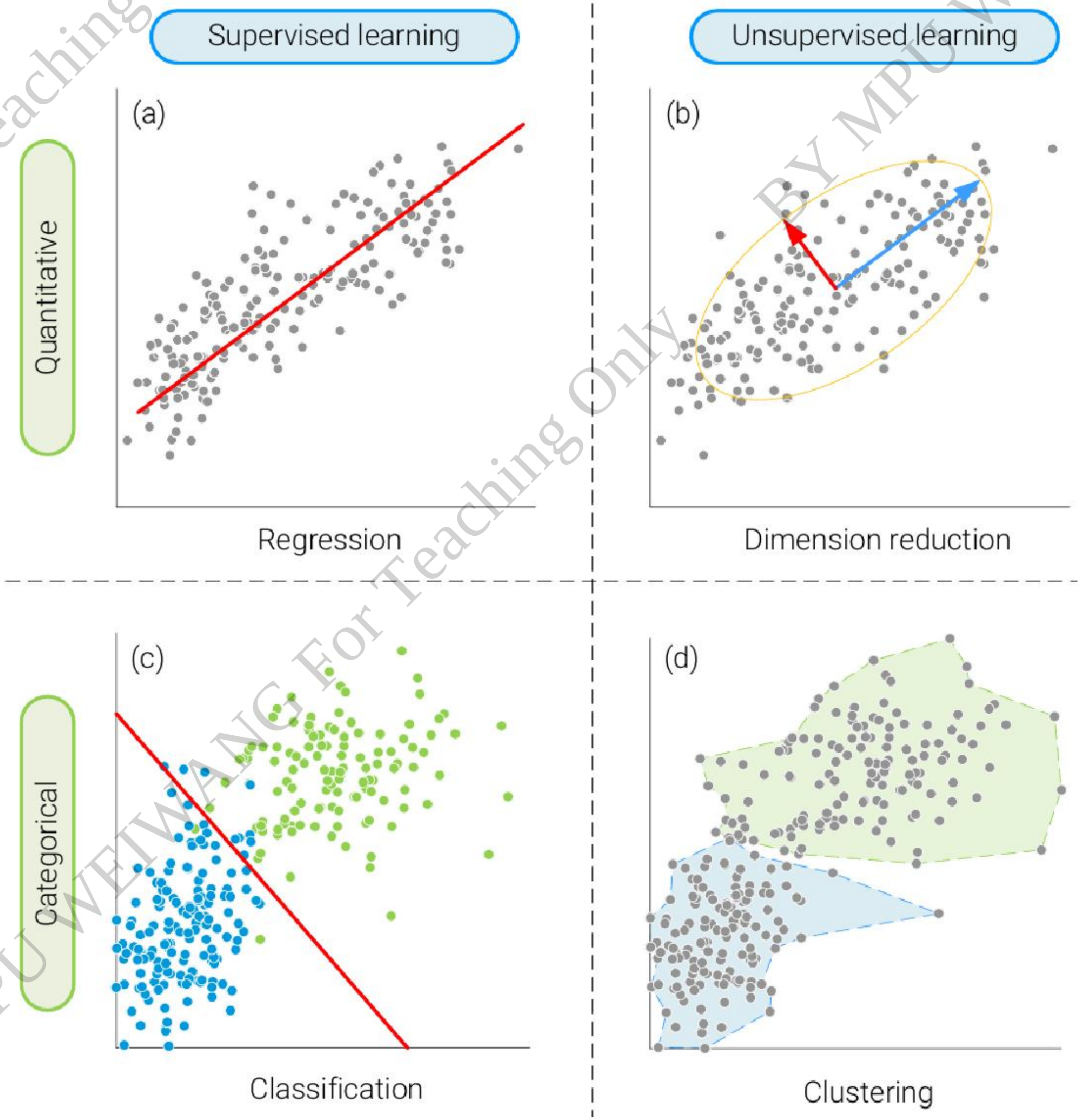
- Solution in an assignment of values for a set of variables.
- **Apps**: Soduko, Speech Recognition, and Face Recognition.
- **Options**: Convolutional Neural Networks, Constraint Satisfaction, Bayesian Networks, Factor Graphs, and Dynamic Ordering.

4. Reflex-based models

- Given a set of <Input, Output> pairs of training data, learn a set of parameters that will map input to output for future data.
- **Apps**: Classification and Regression.
- **Options**: Artificial Neural Networks (ANN), Decision Trees, Support Vector Machines, Regression, Principal Component Analysis, K-Means Clustering, and K-Nearest Neighbor

AI Models Types

Four Main
Machine Learning Methods



AI Models Types

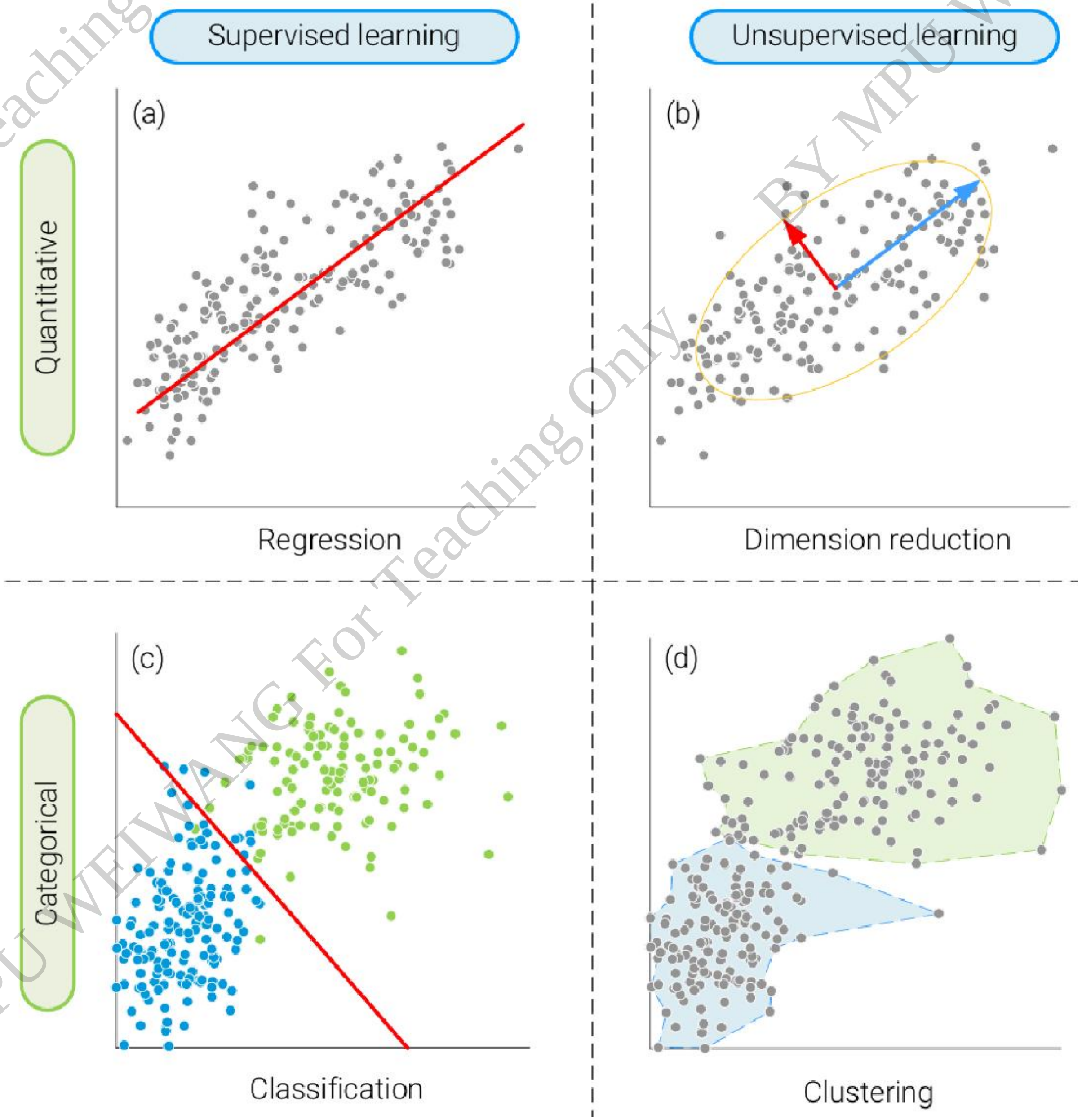
The Big Picture - Learning Paradigms

Supervised Learning: Data comes with Labels (Input X + Output Y)

- → Regression
- → Classification

Unsupervised Learning: Data has No Labels (Input X only)

- → Clustering
- → Dimension Reduction



AI Models Types

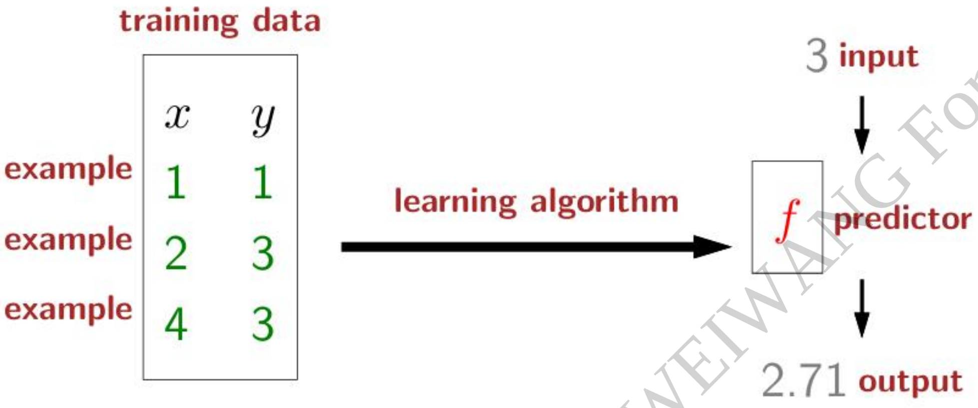
Regression

Definition: Predicting continuous numerical output.

- The Key Question: "How much?" or "How many?"
- Real-world Examples:
  - ◦Predicting House Prices (based on size, location).
  - ◦Forecasting tomorrow's temperature.
  - ◦Stock market price prediction.
- **Common Algorithms**: Linear Regression, Polynomial Regression.
- **Visual**: A scatter plot with a "line of best fit" running through the points.

When the target is a number, we call it **Regression**. Essentially, we are trying to fit a line or curve that best represents the trend of the data

Simple Linear Regression



training data

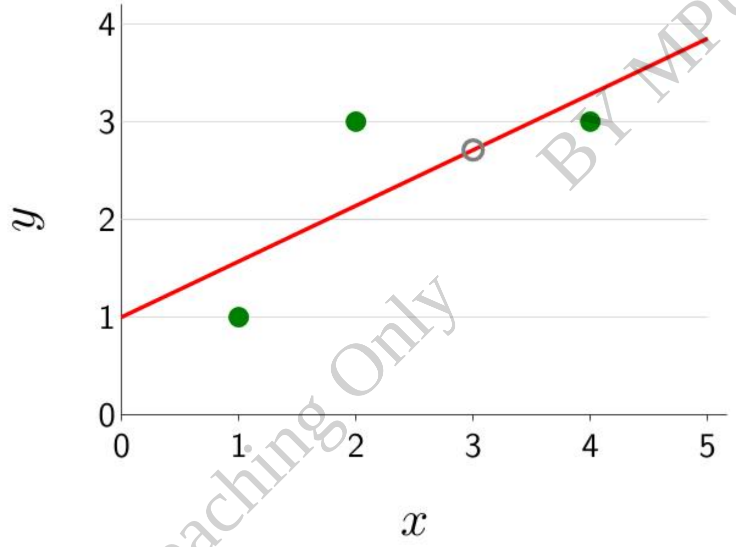
	x	y
example	1	1
example	2	3
example	4	3

learning algorithm

3 input

f predictor

2.71 output



Design decisions:

Which predictors are possible? **hypothesis class**

How good is a predictor? **loss function**

How do we compute the best predictor? **optimization algorithm**

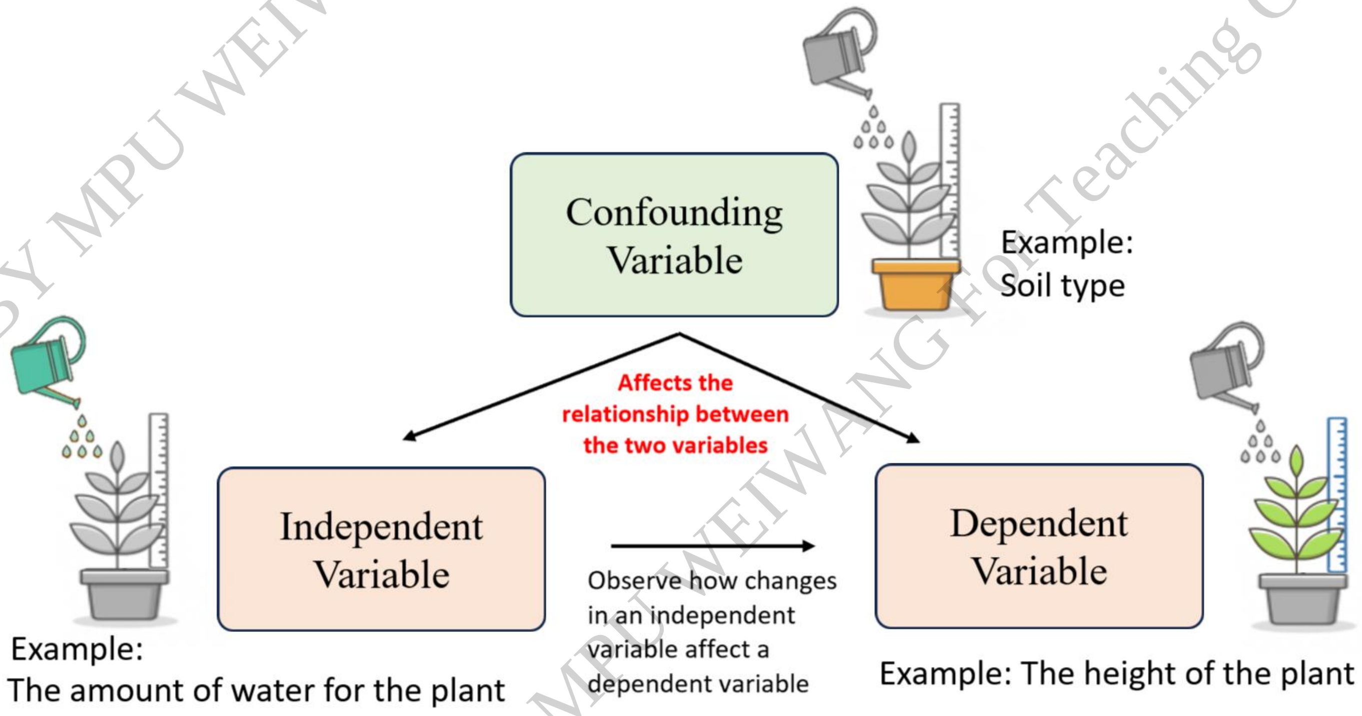
https://www.youtube.com/watch?v=-JTKf-a1JpU

Simple Linear Regression

□ **Independent Variable**: The independent variable is the variable that is manipulated or changed by the researcher in an experiment. It is called "independent" because its value is not influenced by other variables in the study.

□ **Dependent Variable**: The dependent variable is the variable that is measured or observed in an experiment. It is called "dependent" because its value depends on or is influenced by the independent variable.

□ **Confounding Variable**: A confounded variable is an extraneous variable that is related to both the independent and dependent variables, making it difficult to determine the true relationship between them.



Confounding Variable

Example: Soil type

Affects the relationship between the two variables

Independent Variable

Example: The amount of water for the plant

Observe how changes in an independent variable affect a dependent variable

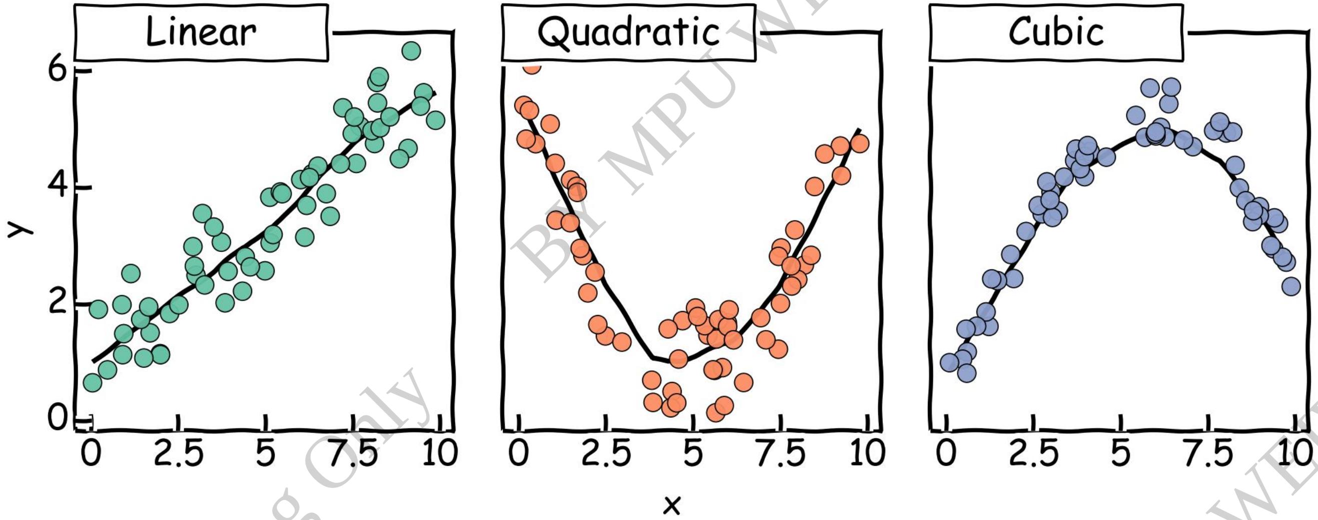
Dependent Variable

Example: The height of the plant

Simple Linear Regression

Regression is a statistical technique used to understand the relationship between one dependent variable and one or more independent variables: Y=f(X_1, X_2,…, X_n)

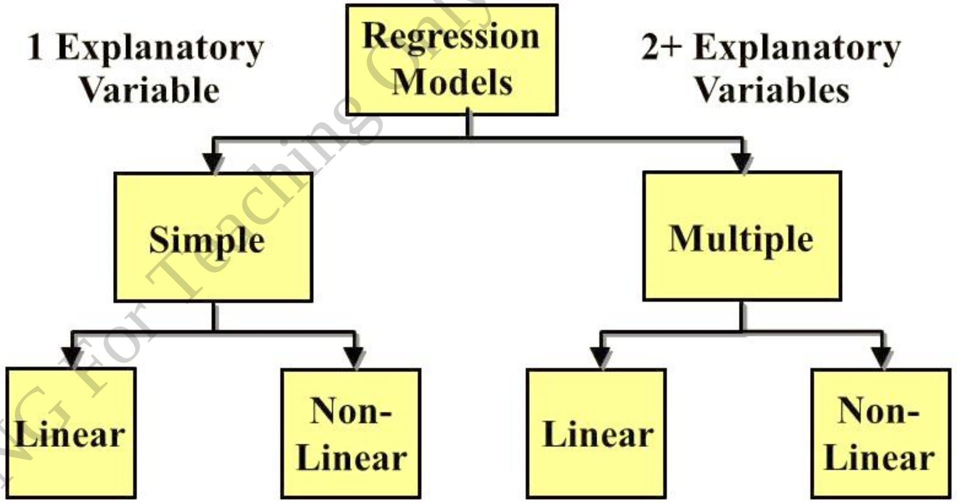
It is commonly employed to predict or estimate the value of the dependent variable based on the values of the independent variables. In regression analysis, the goal is to model the relationship between variables and make predictions or inferences based on that model.



Linear equation in one variable

Quadratic equation of one variable

Cubic equation of one variable



1 Explanatory Variable

Regression Models

2+ Explanatory Variables

Simple

Multiple

Linear

Non-Linear

Linear

Non-Linear

Linear motion represents regular, smooth, and predictable motion;
nonlinear motion often represents irregular, abrupt, and sensitive motion to initial conditions .

Simple Linear Regression

Scenario: Predicting the height of the flower

Data:

- Input (x): the amout of the water.

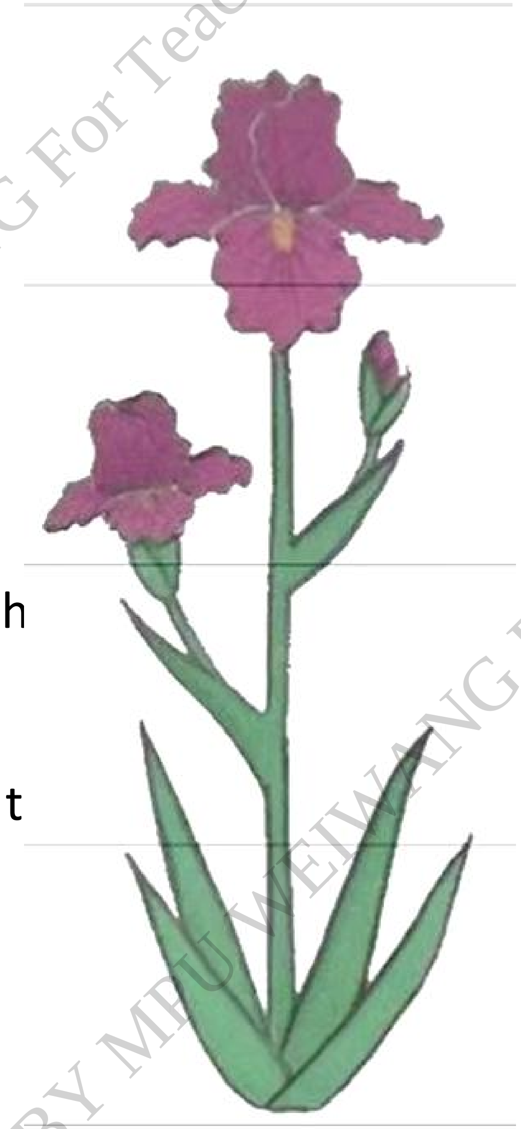
- Output (y): the height of the flower.

Goal: Find a relationship between x and y.

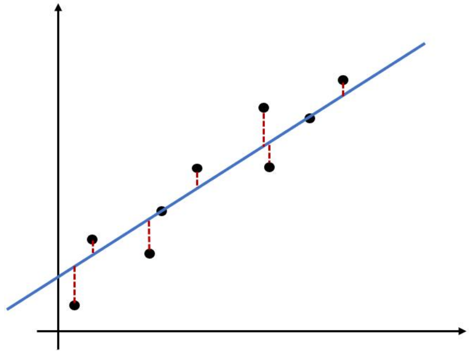
Visual: A scatter plot showing water vs. heigh The points show a general upward trend.

The Model: A straight line that cuts through t data points with minimal error.

Iris Flower



water, x	5	7	12	16	20
growth rate y:	40	120	180	210	240



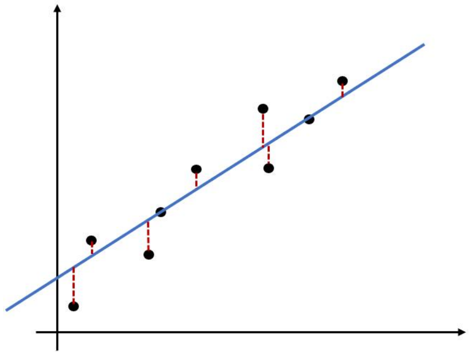
Simple Linear Regression

Mathematical Formulation (The Hypothesis)

Equation: y=wx+b

- Components:
  - ◦y: The prediction (Dependent variable).
  - ◦x: The input feature (Independent variable).
  - ◦w: Weight (Slope/Coefficient, 斜率) - How much does x affect y?
  - ◦b: Bias (Intercept, 截距) - What is the baseline value when x=0?
- The Goal of Learning: To find the optimal values for w and b.

water, x	5	7	12	16	20
growth rate y:	40	120	180	210	240



Simple Linear Regression

Measuring Error - The Cost Function

water, x	5	7	12	16	20
growth rate y:	40	120	180	210	240

- **Question**: How do we know if our line is "good"?

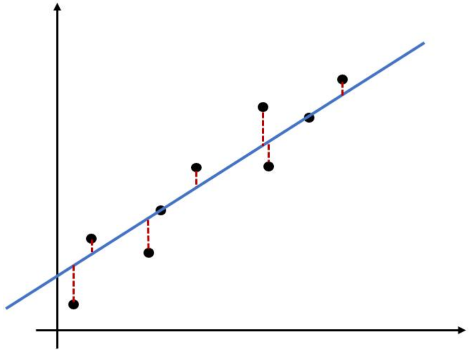
- **Residuals**: The vertical distance between the actual data point and the predicted line.

- **The Metric**: Mean Squared Error (MSE) or Cost Function J(w,b).

- **Formula**:

$$J(w,b) = \frac{1}{n}\sum_{i=1}^{n}(y_{pred}^{(i)} - y_{actual}^{(i)})^2$$

- Why Square?
  - ◦Removes negative signs (errors don't cancel out).
  - ◦Penalizes large errors more heavily.



Simple Linear Regression

Ordinary Least Squares (OLS) 最小二乘法

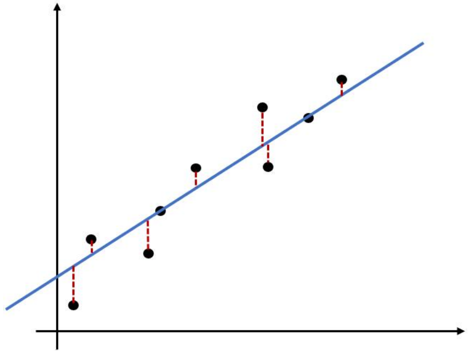
Given a dataset $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ consisting of inputs x and outputs y, where $x_i = (x_{i1}; x_{i2}; ...; x_{id}), y_i \in \mathbb{R}$. The linear regression aims to train a linear model to best fit the output y based on the input x. We aim to learn the relationship to approximate y_i : $y = wx_i + b$. Then, $y \cong y_i$.

The key challenge in linear regression learning lies in determining the parameters w and b to make the fitted output y as close as possible to the true output y_i. In regression tasks, mean squared error is commonly used to measure the loss between predictions and labels. Therefore, the optimization goal in regression tasks is to minimize the mean squared error between the fitted output and the true output.

$$y = wx_i + b$$

$$\Downarrow$$

$$(w^*, b^*) = \arg\min \sum_{i=1}^{m} (y - y_i)^2 = \arg\min \sum_{i=1}^{m} (wx_i + b - y_i)^2$$



Simple Linear Regression

To obtain the minimizing parameters $w*$ and $b*$ for w and b, **based on the mean square loss function, the first derivatives of *w* and *b* can be calculated and equated to zero**. The derivation process for w is as follows:

$$\frac{\partial L(w,b)}{\partial w} = \frac{\partial}{\partial w}\left[\sum_{i=1}^{m}(wx_i + b - y_i)^2\right]$$

$$= \sum_{i=1}^{m}\frac{\partial}{\partial w}[(y_i - wx_i - b)^2]$$

$$= \sum_{i=1}^{m}[2\cdot(y_i - wx_i - b)\cdot(-x_i)]$$

$$= \sum_{i=1}^{m}[2\cdot(wx_i^2 - y_ix_i + bx_i)]$$

$$= 2\cdot\left(w\sum_{i=1}^{m}x_i^2 - \sum_{i=1}^{m}y_ix_i + b\sum_{i=1}^{m}x_i\right)$$

$$\frac{\partial L(w,b)}{\partial b} = \frac{\partial}{\partial b}\left[\sum_{i=1}^{m}(wx_i + b - y_i)^2\right]$$

$$= \sum_{i=1}^{m}\frac{\partial}{\partial b}[(y_i - wx_i - b)^2]$$

$$= \sum_{i=1}^{m}[2\cdot(y_i - wx_i - b)\cdot(-1)]$$

$$= \sum_{i=1}^{m}[2\cdot(-y_i + wx_i + b)]$$

$$= 2\cdot(-\sum_{i=1}^{m}y_i + \sum_{i=1}^{m}wx_i + \sum_{i=1}^{m}b\,)$$

$$= 2\left(mb - \sum_{i=1}^{m}(y_i - wx_i)\right)$$

Simple Linear Regression

The best expressions of w and b can be solved as:

$$\frac{\partial L(w,b)}{\partial b} = 2\left(mb - \sum_{i=1}^{m} (y_i - wx_i)\right)=0$$

$$\frac{\partial L(w,b)}{\partial w} = \frac{\partial}{\partial w}\left[\sum_{i=1}^{m} (wx_i + b - y_i)^2\right]$$

$$= \sum_{i=1}^{m} \frac{\partial}{\partial w}[(y_i - wx_i - b)^2]$$

$$= \sum_{i=1}^{m} [2 \cdot (y_i - wx_i - b) \cdot (-x_i)]$$

$$= \sum_{i=1}^{m} [2 \cdot (y_i - wx_i - \bar{y} + w\bar{x}) \cdot (-x_i)]$$

$$=-2\sum_{i=1}^{m} [\ (y_i - \bar{y}\ + w(\bar{x} - x_i)) \cdot x_i]=0$$

$$b^* = \frac{1}{m}\sum_{i=1}^{m} (y_i - wx_i) = \bar{y} - w\bar{x}$$

$$\sum_{i=1}^{m} (y_i - \bar{y}\ + w(\bar{x} - x_i)) = 0$$

$$\sum_{i=1}^{m} (y_i - \bar{y}) =- w\sum_{i=1}^{m} (\bar{x} - x_i)$$

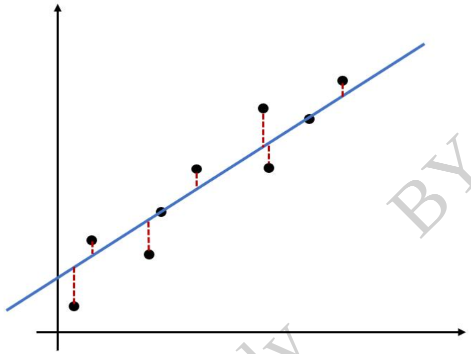
covariance of x and y

$$w^* = \frac{\sum_{i=1}^{m} (y_i - \bar{y})}{\sum_{i=1}^{m} (x_i - \bar{x})} = \frac{\sum_{i=1}^{m} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{m} (x_i - \bar{x})^2}$$

variance of x

Simple Linear Regression

The best expressions of w and b can be solved as:



$$\frac{\partial L(w,b)}{\partial b} = 2\left(mb - \sum_{i=1}^{m} (y_i - wx_i)\right) = 0 \quad \Rightarrow \quad b^* = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i) = \bar{y} - w\bar{x}$$

$$\frac{\partial L(w,b)}{\partial w} = 2 \cdot \left(w\sum_{i=1}^{m} x_i^2 - \sum_{i=1}^{m} y_i x_i + b\sum_{i=1}^{m} x_i\right) = 0$$

$$w^* = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2} = \frac{\sum_{i=1}^{m} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{m} (x_i - \bar{x})^2}$$

numerator 分子
分母denominator

Simple Linear Regression

The effect of water x on the plant growth rate y was measured:

water, x	5	7	12	16	20
growth rate y:	40	120	180	210	240

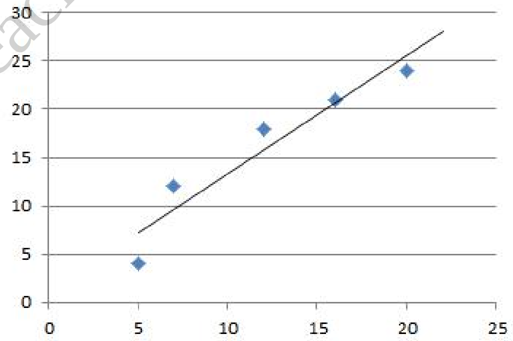


Diagram 3

Find the equation of the regression line.

$$y = wx_i + b$$

$$b^* = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i) = \bar{y} - w\bar{x}$$

$$w^* = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2}$$

$$= \frac{\sum_{i=1}^{m}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

Simple Linear Regression

The effect of water x on the plant growth rate y was measured:

water, x	5	7	12	16	20
growth rate y:	40	120	180	210	240

Find the equation of the regression line.

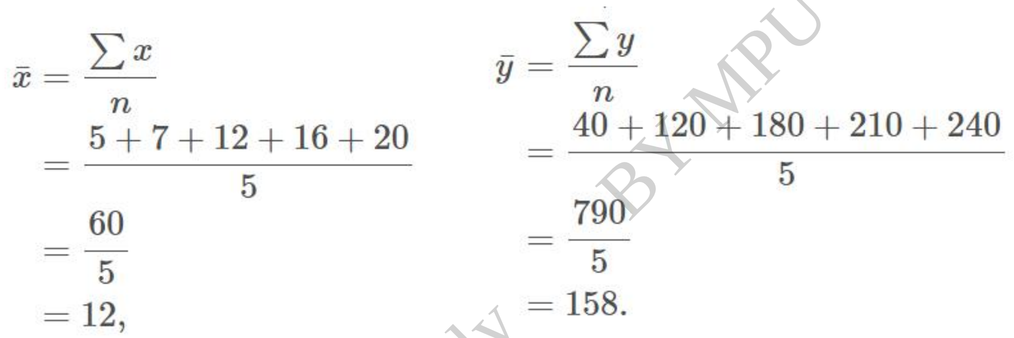
$$y = wx_i + b$$

$$b^* = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i) = \bar{y} - w\bar{x}$$

$$w^* = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2}$$

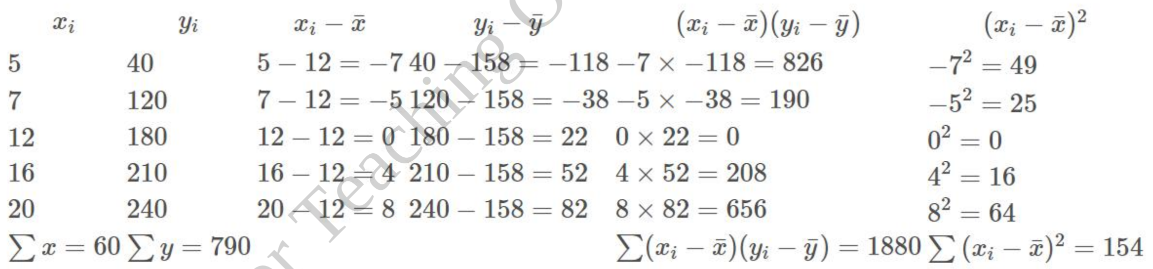
$$= \frac{\sum_{i=1}^{m}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

https://www.youtube.com/watch?v=gPfgB4ew3RY

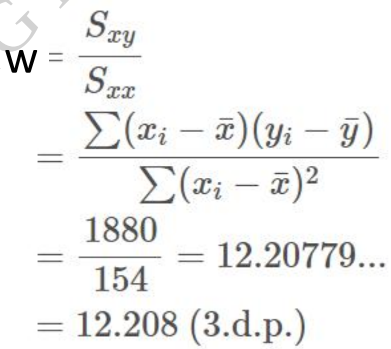


$$\bar{x} = \frac{\sum x}{n} = \frac{5+7+12+16+20}{5} = \frac{60}{5} = 12,$$

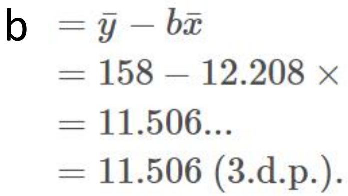
$$\bar{y} = \frac{\sum y}{n} = \frac{40+120+180+210+240}{5} = \frac{790}{5} = 158.$$



x_i	y_i	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
5	40	$5 - 12 = -7$	$40 - 158 = -118$	$-7 \times -118 = 826$	$-7^2 = 49$
7	120	$7 - 12 = -5$	$120 - 158 = -38$	$-5 \times -38 = 190$	$-5^2 = 25$
12	180	$12 - 12 = 0$	$180 - 158 = 22$	$0 \times 22 = 0$	$0^2 = 0$
16	210	$16 - 12 = 4$	$210 - 158 = 52$	$4 \times 52 = 208$	$4^2 = 16$
20	240	$20 - 12 = 8$	$240 - 158 = 82$	$8 \times 82 = 656$	$8^2 = 64$
$\sum x = 60$	$\sum y = 790$			$\sum(x_i - \bar{x})(y_i - \bar{y}) = 1880$	$\sum(x_i - \bar{x})^2 = 154$



$$w = \frac{S_{xy}}{S_{xx}} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{1880}{154} = 12.20779... = 12.208 \text{ (3.d.p.)}$$



$$b = \bar{y} - b\bar{x} = 158 - 12.208 \times 12 = 11.506... = 11.506 \text{ (3.d.p.).}$$

Simple Linear Regression

- **Error-based metrics**, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), quantify the differences between predicted and actual values.

- **Mean Squared Error (MSE) and Root Mean Squared Error (RMSE)**: MSE and RMSE quantify the average squared difference between the predicted values and the actual values. Lower values of MSE and RMSE indicate better model performance, as they suggest that the model's predictions are closer to the true values. RMSE is particularly useful as it has the same unit of measurement as the dependent variable, making it easier to interpret.

$$MSE = \frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

- **Mean Absolute Error (MAE)**: MAE calculates the average absolute difference between the predicted values and the actual values. Like MSE and RMSE, lower values of MAE indicate better model performance. MAE is less sensitive to outliers compared to MSE, making it a suitable metric when outliers are present in the data.

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

Simple Linear Regression

- **Coefficient of Determination metrics**, such as R-squared (R²), assess the proportion of variance explained by the model. R-squared measures the proportion of variance in the dependent variable that is explained by the regression model. It ranges from 0 to 1, with higher values indicating a better fit. R-squared can help assess how well the model captures the variability in the data and how much of the response variable's variation is accounted for by the predictors.

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\Sigma_{i=1}^{N}(y_i - \hat{y_i})^2}{\Sigma_{i=1}^{N}(y_i - \bar{y_i})^2}$$

- **Adjusted R-squared** is a modified form of R-squared that considers the number of predictors in a model. Itis a valuable metric for assessing the goodness of fit of regression models, especially when comparing models with different numbers of predictors. Generally positive, the adjusted R-squared is designed to be lower than the R-squared value, reflecting a balance between model complexity and explanatory power.

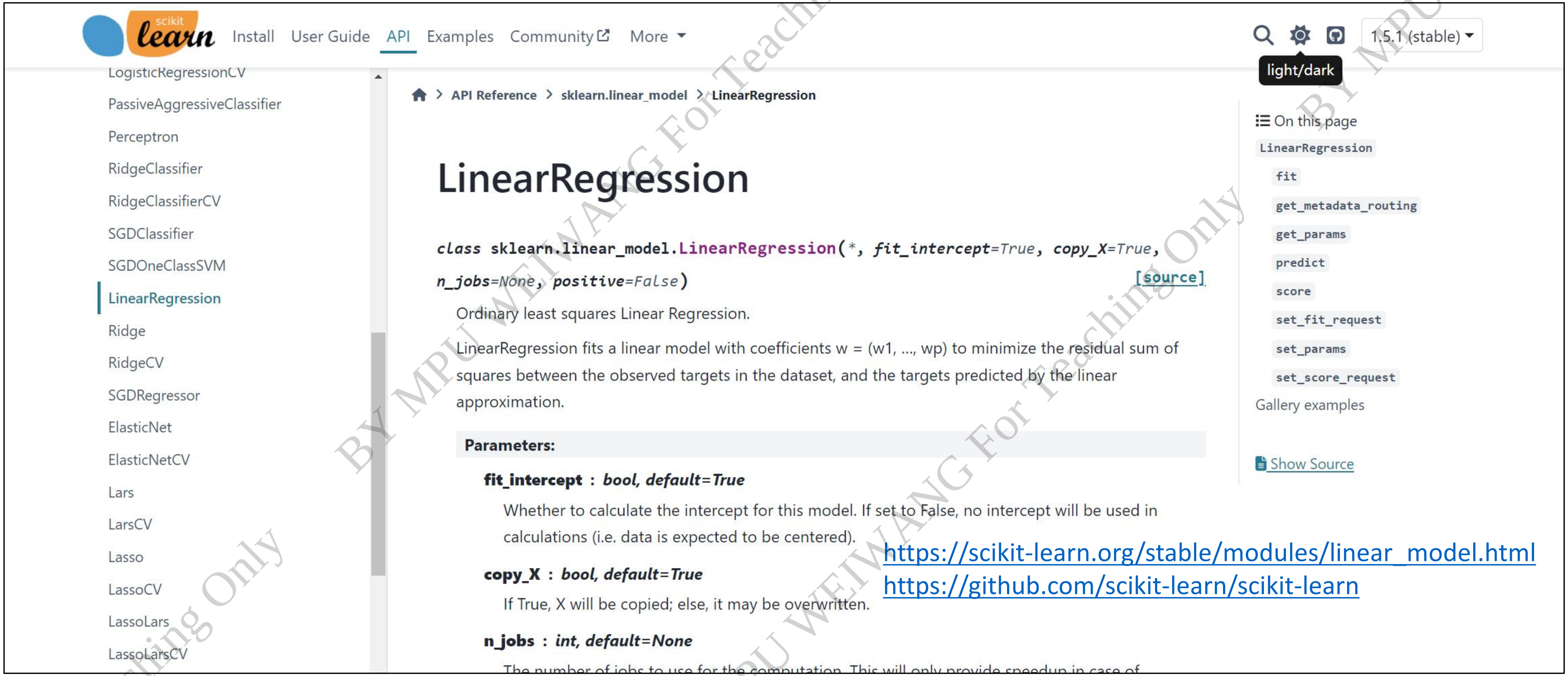
$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

R^2 Sample R-Squared

N Total Sample Size

p Number of independent variable

Simple Linear Regression



API Reference > sklearn.linear_model > LinearRegression

LinearRegression

```
class sklearn.linear_model.LinearRegression(*, fit_intercept=True, copy_X=True,
n_jobs=None, positive=False)
```

[source]

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Parameters:

fit_intercept : ***bool, default=True***

Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).

copy_X : ***bool, default=True***

If True, X will be copied; else, it may be overwritten.

n_jobs : ***int, default=None***

The number of jobs to use for the computation. This will only provide speedup in case of

https://scikit-learn.org/stable/modules/linear_model.html

https://github.com/scikit-learn/scikit-learn

Multiple Linear Regression

Multiple Linear Regression is a statistical method used to model the relationship between a dependent variable (response) and **multiple independent variables** (predictors). It extends the concept of simple linear regression by considering the joint effect of multiple predictors on the outcome.

In multiple linear regression, the relationship between the predictors and the response is assumed to be linear, but the model accounts for the influence of multiple predictors simultaneously. The objective is to **estimate the regression coefficients** that minimize the sum of squared differences between the observed values of the response variable and the predicted values based on the predictors. The multiple linear regression equation can be represented as:

$$\mathrm{y} = w_0 + w_1\mathrm{x}_1 + w_2\,\mathrm{x}_2 + \ldots + w_n{}^{*}\mathrm{x}_n$$

$$\Downarrow$$

$$(w_0,\ w_{1,\ \ldots,}\ w_n) = \arg\min \sum_{i=1}^{m} (y - y_i)^2$$

Multiple Linear Regression

$$y = w_0 + w_1 x_1 + w_2\, x_2 + \dots + w_n{}^{*} x_n$$

⇓

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} x_1^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ x_m^{\mathrm{T}} & 1 \end{pmatrix} \quad + \quad \boldsymbol{y} = (y_1; y_2; \dots; y_m)$$

⇓

The matrix expression for the optimization objective function of parameters is:

$$\widehat{\boldsymbol{w}}^{*} = \arg\min\,(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}})^{\mathrm{T}}\,(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}})$$

⇓

Let $L = (\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}})$. The derivation process for the parameter $\widehat{\boldsymbol{w}}$ is as follows:

$$L = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}\widehat{\boldsymbol{w}} - \widehat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + \widehat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\widehat{\boldsymbol{w}}$$

Multiple Linear Regression

$$L = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + \hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\hat{\boldsymbol{w}}$$

The derivation process for the parameter $\hat{\boldsymbol{w}}$ is as follows:

$$\frac{\partial L}{\partial \hat{\boldsymbol{w}}} = \frac{\partial \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}{\partial \hat{\boldsymbol{w}}} - \frac{\partial \boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}\hat{\boldsymbol{w}}}{\partial \hat{\boldsymbol{w}}} - \frac{\partial \hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}}{\partial \hat{\boldsymbol{w}}} + \frac{\partial \hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\hat{\boldsymbol{w}}}{\partial \hat{\boldsymbol{w}}}$$

$$\frac{\partial \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}^{\mathrm{T}}\boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$$

+

$$\frac{\partial \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}$$

By using the **matrix differentiation rule**, we have:

$$\frac{\partial L}{\partial \hat{\boldsymbol{w}}} = 0 - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\hat{\boldsymbol{w}} = 2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y})$$

When the matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is full rank or positive definite, setting the above equation to 0, the parameters can be solved as:

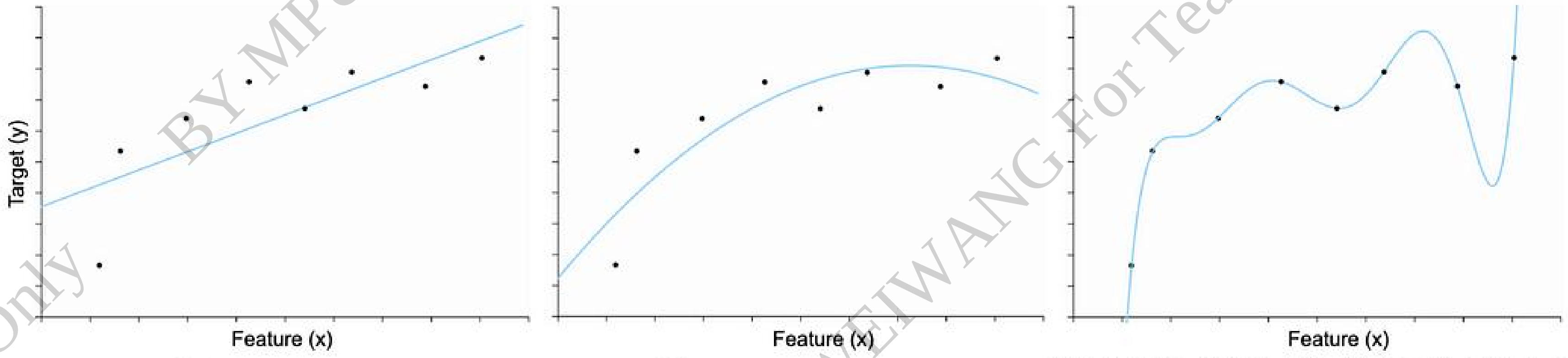
$$\hat{\boldsymbol{w}}^{*} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

Linear Regression

□ **Underfitting:** Underfitting occurs when the model doesn't work well with both training data and testing data (meaning the accuracy of both training & testing datasets is below 50%). A possible solution is applying Data Wrangling (data preprocessing or feature engineering).

□ **Good Fit**: A model is a Good Fit when it works well with both training and testing datasets (meaning the accuracy for both datasets is around 70%–85% in general cases). It means that we almost achieved our goal.

□ **Overfitting :** Overfitting occurs when the model works very well with the training dataset, but on the testing dataset, it fails (meaning that the accuracy of training data is >90% and testing data is < 65%). Since the model works best only on training data and whenever it faces a new situation during testing, it gives wrong results. It is also called a model with *high variance.* A possible solution is to use the correct regression technique, that is **Ridge** or **Lasso**.



$f(x) = y = \beta_1 x + \beta_0$

Simple Linear Regression

High bias => Under-fitted => Very simple

$f(x) = y = \beta_2 x^2 + \beta_1 x + \beta_0$

2nd Order Polynomial

Better or Just Right?

$f(x) = y = \beta_7 x^7 + \beta_6 x^6 + \beta_5 x^5 + \beta_4 x^4 + \beta_3 x^3 + \beta_2 x^2 + \beta_1 x + \beta_0$

7th order polynomial

High variance => Over-fitted =>Very complex

Linear Regression

LASSO Regression

The optimal parameter estimation expression for the linear regression model is as follows:

$$\hat{\boldsymbol{w}}^{*} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

Assuming there are *m* training samples and *n* features,

- □ When ***m>n***, if rank(X)=n, meaning *X* is full rank, then $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is invertible, and the linear regression parameter estimation formula can be directly solved. Typically with *m*>*n*, meaning that the number of samples is greater than the number of features.
- □ If ***m<n***, implying that the number of features is greater than the number of samples, rank(*X*)<*n*, and the matrix *X* is not full rank. In this case, typically with *m*>*n*, meaning that the number of samples is greater than the number of features. is not invertible, and the parameter $\hat{\boldsymbol{w}}^{*}$ in the parameter estimation formula is not estimable.

The LASSO (Least Absolute Shrinkage and Selection Operator) model provides a solution for such unestimable parameters in linear regression by adding an L1 regularization term to the loss function:

$$\boldsymbol{L(w) = (y - wX)^{2} + \lambda||w||_{1}}$$

Where $||\boldsymbol{w}||_1$ 1 represents the 1-norm of the matrix, *λ* is the coefficient of the 1-norm term.

A norm can be viewed as a function representing a concept of length or distance. For vectors or matrices, common norms include the 0-norm, 1-norm, 2-norm, and *p*-norm. The 0-norm of a matrix counts the number of non-zero elements in the matrix. The 1-norm of a matrix is defined as the sum of the absolute values of all elements in the matrix, while the L2-norm of a matrix refers to the square root of the sum of squares of all elements in the matrix.

Linear Regression

LASSO Regression

The above expression is equivalent to adding an L1 regularization term to the original linear regression loss function, where λ is also known as the regularization coefficient. From the perspective of preventing overfitting, the regularization term imposes a penalty on the target parameters, preventing the model from becoming overly complex.

During the optimization process, the presence of the regularization term gradually forces the coefficients of unimportant features to zero, thereby retaining essential features and simplifying the model.

$$L(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{wX})^2 + \lambda||\boldsymbol{w}||_1$$

$$\Downarrow$$

$$\begin{aligned} &\arg_w \ \min\,(\boldsymbol{y} - \boldsymbol{wX})^2 \\ &\text{s.t.} \quad \sum|w_{ij}| < s \end{aligned}$$

The constraint $\sum|w_{ij}| < s$ implies that the sum of the absolute values of all elements in the weight coefficient matrix is less than a specified constant ss. A smaller value of *s* leads to more features having their coefficients compressed to zero.

Linear Regression

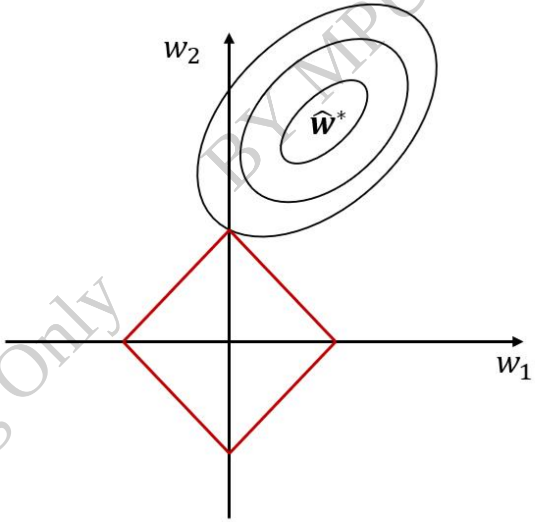
LASSO Regression

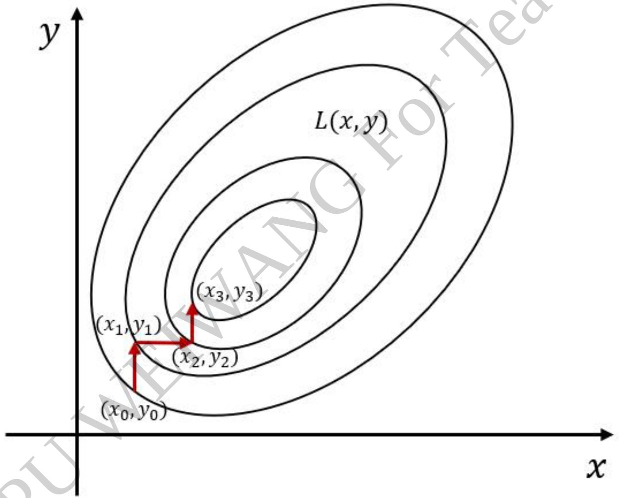
During the optimization process, the presence of the regularization term gradually forces the coefficients of unimportant features to zero, thereby retaining essential features and simplifying the model.

$$\arg_w \min (\boldsymbol{y} - \boldsymbol{wX})^2$$

$$\text{s.t.} \quad \sum |w_{ij}| < s$$

Example: The x-axis and y-axis represent two regression parameters w_1 and w_1 respectively. The red rectangular line represents the L1 regularization constraint of LASSO, $|w_1| + |w_2| \leq s$, while the elliptical region denotes the solution space for the regression parameters. It can be observed that the parameter solution space for LASSO intersects with the y-axis. This intersection implies that the parameter w_1 is being compressed to zero.





https://xavierbourretsicotte.github.io/coordinate_descent.html

Linear Regression

Ridge Regression

Ridge regression is a model that modifies the linear regression loss function by using the **L2-norm** as a penalty term. In Ridge regression, $\lambda||\boldsymbol{w}||_2 = \lambda\sum_{i=1}^{n} w_i^2$ is the L2 regularization term. The term $\lambda||\boldsymbol{w}||_2$ is also known as the L2 regularization term.

The principle behind using the L2-norm for regularization is to **minimize each element of the parameter matrix**, making them approach zero but not exactly zero as with L1 regularization. This approach helps reduce the complexity of the model.

$$L(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{wX})^2 + \lambda||\boldsymbol{w}||_2$$

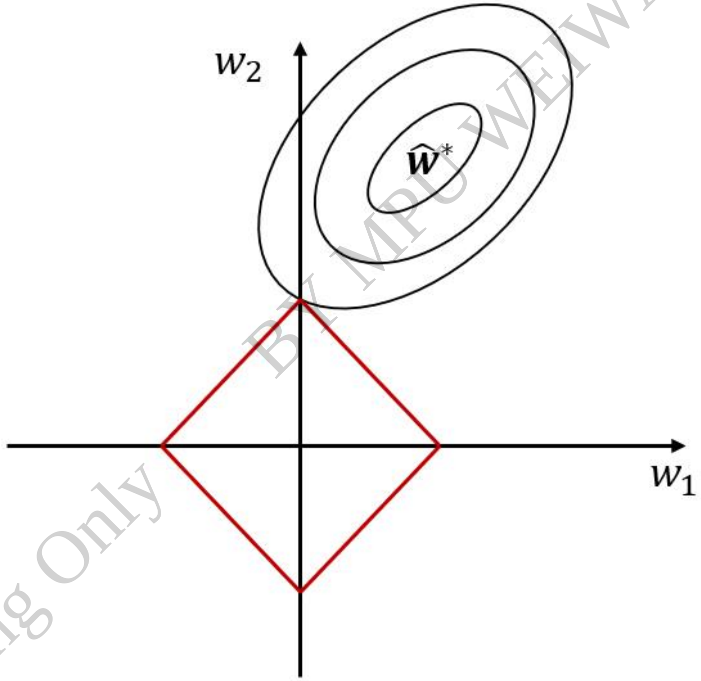
$$\arg_w \ \min\,(\boldsymbol{y} - \boldsymbol{wX})^2$$
$$\text{s.t.} \ \sum w_{ij}^2 < s$$

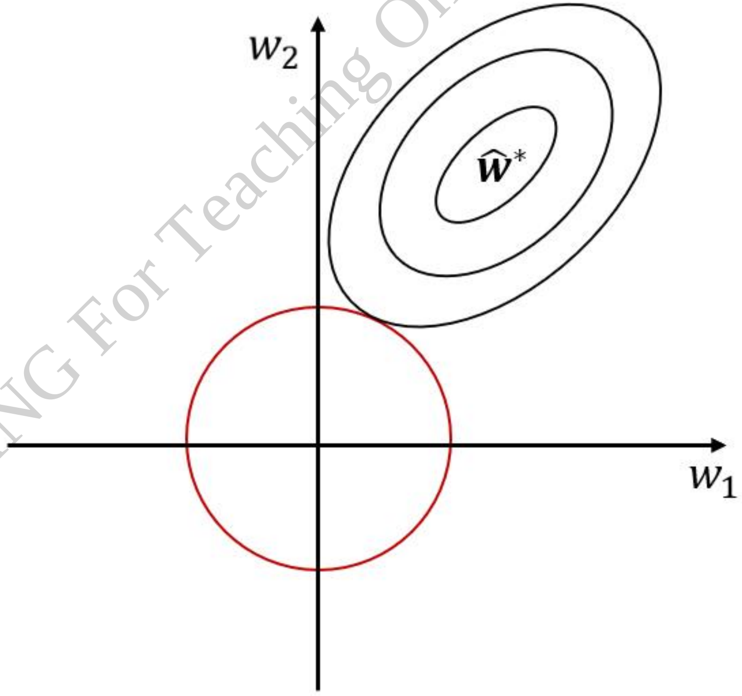
The constraint $\sum w_{ij}^2 < s$implies that the sum of the squares of all elements in the weight coefficient matrix is less than a specified constant *s*.

Linear Regression

Ridge Regression: The x-axis and y-axis represent two regression parameters *w*1 and *w*2 respectively. The red circular region represents the L2 regularization constraint of Ridge regression, $w_1^2 + w_2^2 \leq s$, while the elliptical region denotes the solution space for the regression parameters.

It can be observed that the parameter solution space for LASSO intersects with the y-axis, indicating that certain parameters are being compressed to zero, whereas the parameters in Ridge regression are close to zero but not exactly zero.





Linear Regression

- □ From the perspective of interpretability in linear models, both LASSO and Ridge aim to identify **key factors** influencing the dependent variable while reducing model complexity.
- □ Mathematically, LASSO and Ridge are viewed as compromise solutions in linear regression models where the parameter estimation formulas are not directly solvable, leading to biased estimations.
- □ In the context of the three elements of machine learning, LASSO and Ridge can be seen as machine learning methods that utilize linear regression models with regularization terms, minimize the loss function with regularization terms as the strategy, and employ algorithms such as coordinate descent or gradient descent.

Linear Regression

Stepwise Regression

In statistics, **stepwise regression** is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure. This is **an efficient way to select the most useful explanatory variables**.

$$y = w_0 + w_1 y_1 + w_2\, y_2 + \ldots + w_n{}^* y_n$$

□**Forward selection**, which involves starting with no variables in the model, testing the addition of each variable using a **chosen model fit criterion**, adding the variable (if any) whose inclusion gives the most statistically significant improvement of the fit, and repeating this process until none improves the model to a statistically significant extent.

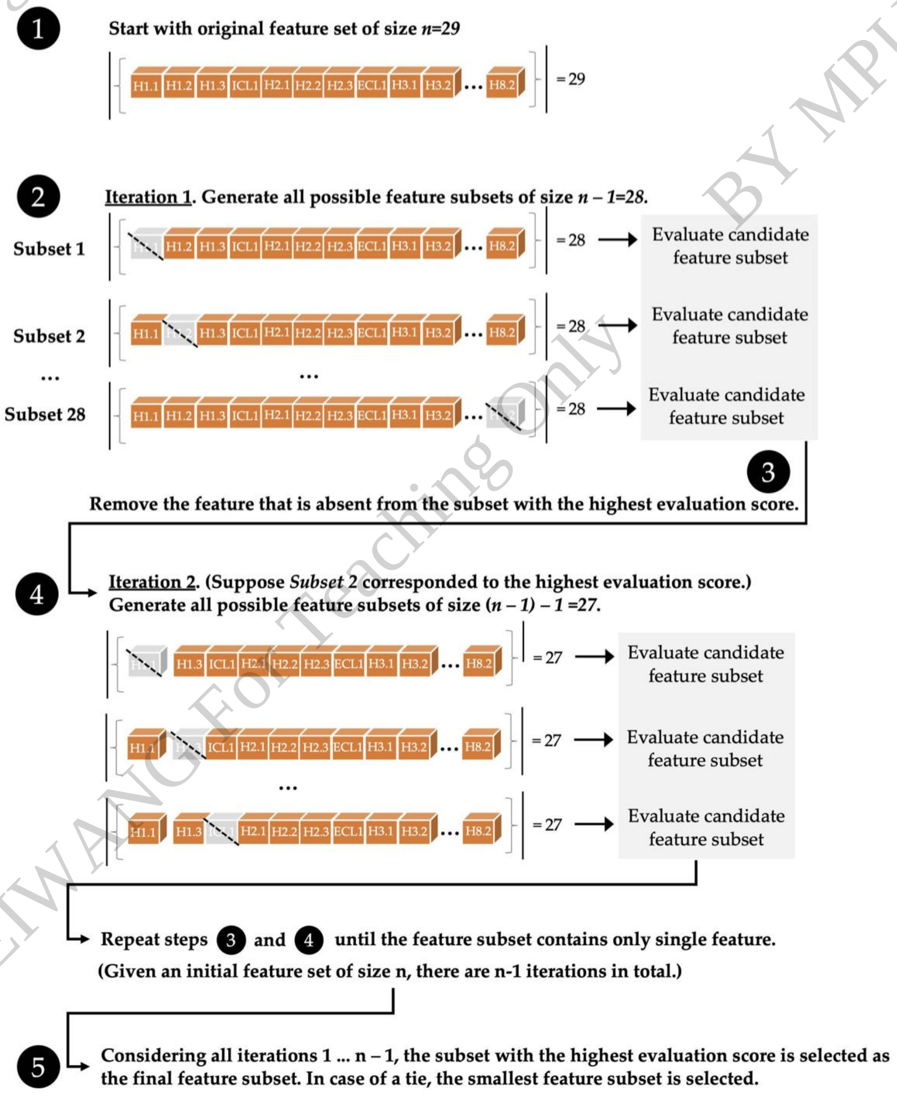
□**Backward elimination**, which involves starting with all candidate variables, testing the deletion of each variable using a chosen model fit criterion, deleting the variable (if any) whose loss gives the most statistically insignificant deterioration of the model fit, and repeating this process until no further variables can be deleted without a statistically significant loss of fit.

□**Bidirectional elimination**, a combination of the above, testing at each step for variables to be included or excluded.

Linear Regression

Sequential backward selection process

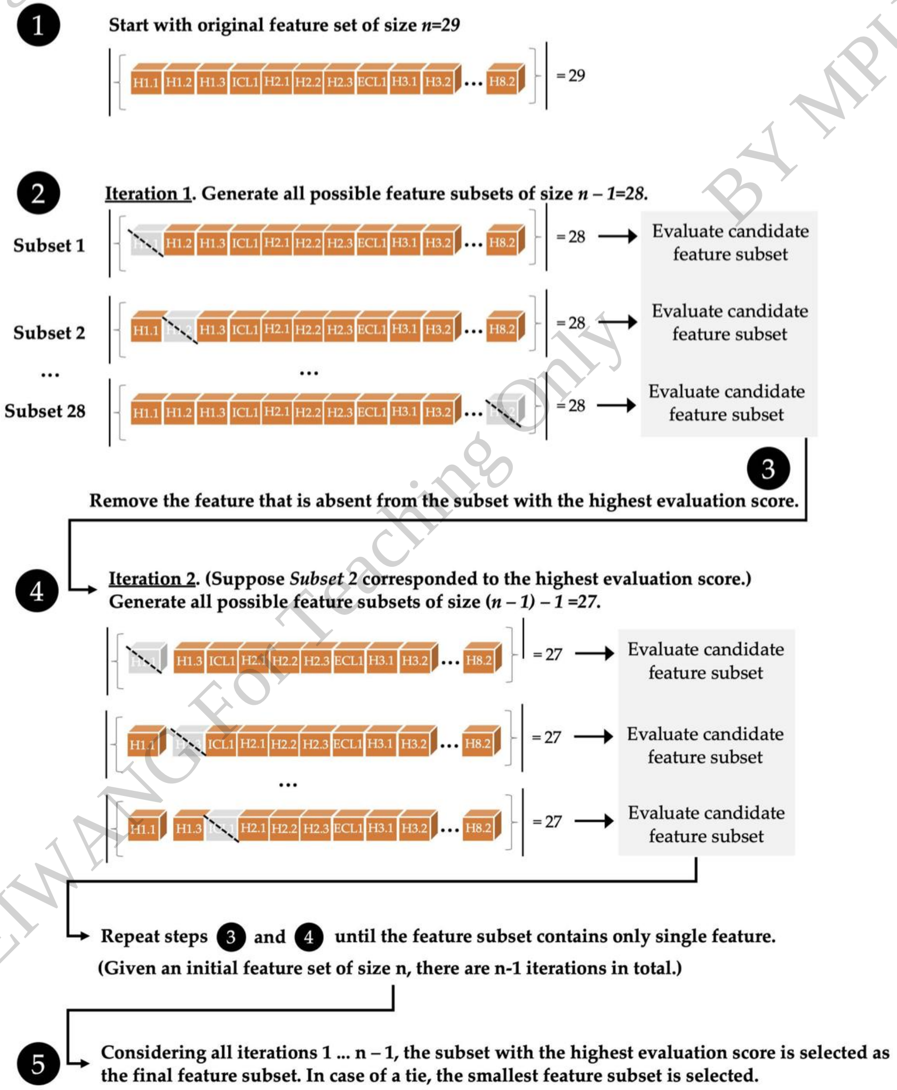
Backward elimination, which involves starting with all candidate variables, testing the deletion of each variable using a chosen model fit criterion, deleting the variable (if any) whose loss gives the most statistically insignificant deterioration of the model fit, and repeating this process until no further variables can be deleted without a statistically significant loss of fit.


1
Start with original feature set of size *n=29*
H1.1 H1.2 H1.3 ICL1 H2.1 H2.2 H2.3 ECL1 H3.1 H3.2 ... H8.2
= 29
2
Iteration 1. Generate all possible feature subsets of size *n – 1=28.*
Subset 1
Subset 2
...
Subset 28
= 28
Evaluate candidate feature subset
3
Remove the feature that is absent from the subset with the highest evaluation score.
4
Iteration 2. (Suppose *Subset 2* corresponded to the highest evaluation score.)
Generate all possible feature subsets of size *(n – 1) – 1 =27.*
= 27
Evaluate candidate feature subset
Repeat steps 3 and 4 until the feature subset contains only single feature.
(Given an initial feature set of size n, there are n-1 iterations in total.)
5
Considering all iterations 1 ... n – 1, the subset with the highest evaluation score is selected as the final feature subset. In case of a tie, the smallest feature subset is selected.

Linear Regression

Sequential backward selection process

Backward elimination, which involves starting with all candidate variables, testing the deletion of each variable using a chosen model fit criterion, deleting the variable (if any) whose loss gives the most statistically insignificant deterioration of the model fit, and repeating this process until no further variables can be deleted without a statistically significant loss of fit.



1 Start with original feature set of size *n=29*

2 Iteration 1. Generate all possible feature subsets of size *n – 1=28.*

3 Remove the feature that is absent from the subset with the highest evaluation score.

4 Iteration 2. (Suppose *Subset 2* corresponded to the highest evaluation score.) Generate all possible feature subsets of size *(n – 1) – 1 =27.*

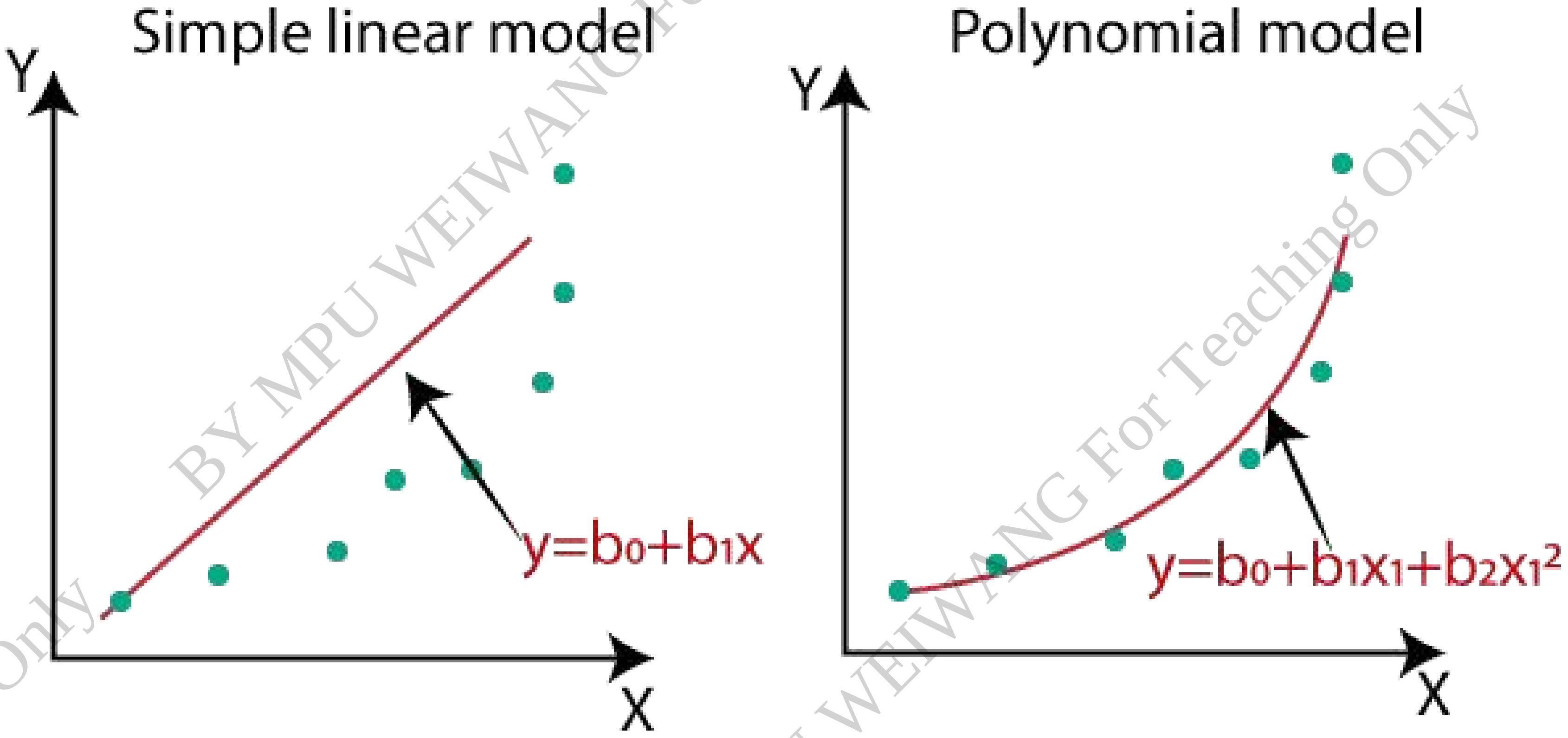
Repeat steps 3 and 4 until the feature subset contains only single feature. (Given an initial feature set of size n, there are n-1 iterations in total.)

5 Considering all iterations 1 ... n – 1, the subset with the highest evaluation score is selected as the final feature subset. In case of a tie, the smallest feature subset is selected.

H1.1 H1.2 H1.3 ICL1 H2.1 H2.2 H2.3 ECL1 H3.1 H3.2 ... H8.2
= 29
Subset 1
Subset 2
...
Subset 28
= 28
= 27
Evaluate candidate feature subset

Nonlinear Regression

Polynomial Regression



Nonlinear Regression: Polynomial Regression

Linear regression limitation

- ◦Assumes a linear relationship between input x and output y
- ◦Many real-world phenomena are nonlinear

Examples of nonlinear relationships

- ◦Growth curves (population, learning curves)
- ◦Physical systems (trajectory, acceleration)
- ◦Economics (cost vs. production)

•Key idea

- ◦Keep the model linear in parameters
- ◦But allow nonlinear relationships in features

•Polynomial Regression

- ◦Extends linear regression by adding polynomial terms

Nonlinear Regression: Polynomial Regression

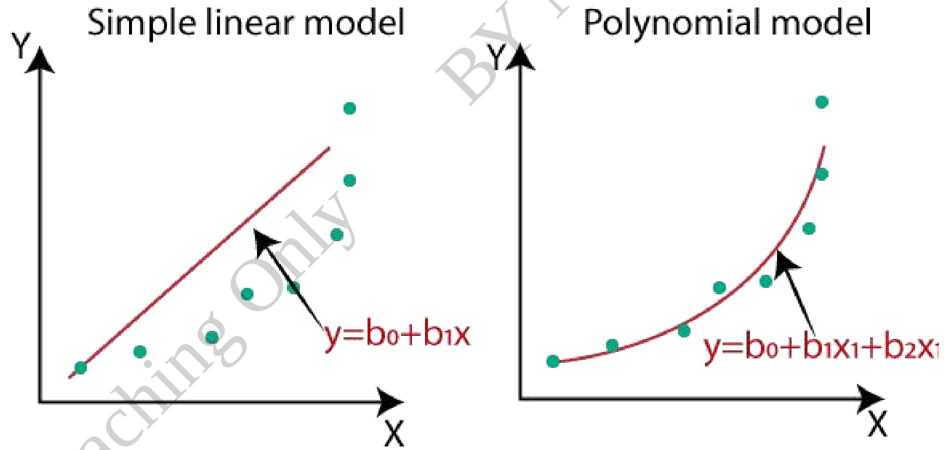
Polynomial regression is a type of regression analysis that models the relationship between the independent variable(s) and the dependent variable as an n-th degree polynomial function. It extends the concept of simple linear regression by introducing higher-order polynomial terms.

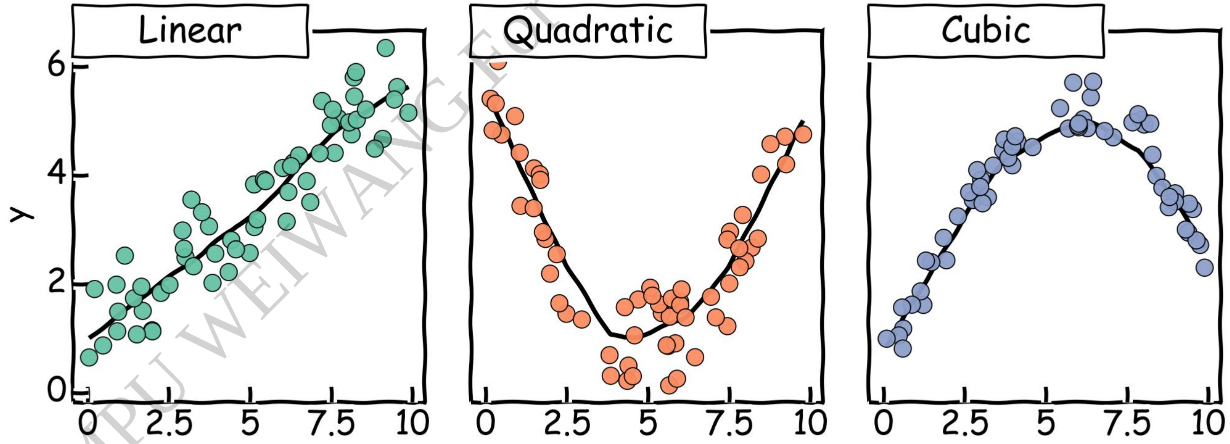
□ **Univariate polynomial regression** involves fitting a polynomial function to a single independent variable (or feature) and a single dependent variable. The model assumes that the relationship between the independent and dependent variables can be approximated by a polynomial equation of a specified degree.

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \ldots + \beta_n X^n$$

□ **Multivariate polynomial regression** extends the concept of polynomial regression to multiple independent variables. It allows for modeling the relationship between a dependent variable and multiple independent variables using a polynomial equation of a specified degree.

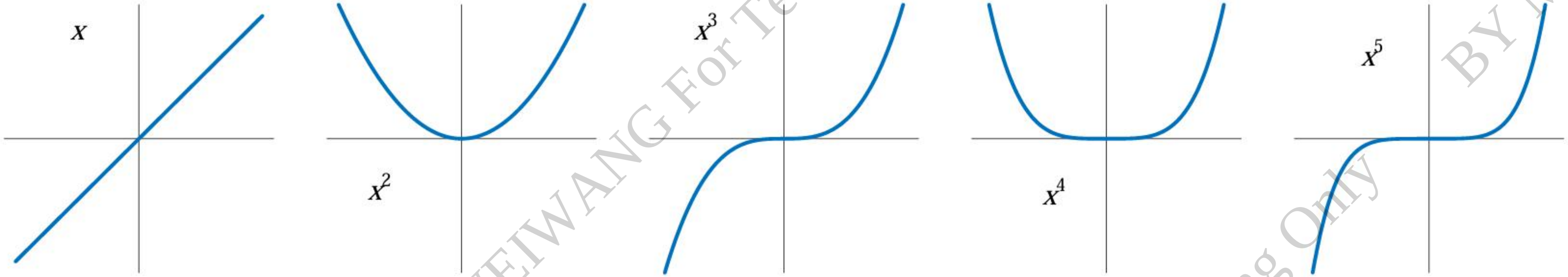
$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \beta_{p+1} X_1^2 + \beta_{p+2} X_1 X_2 + \ldots + \beta_{2n+p} X_p^n$$



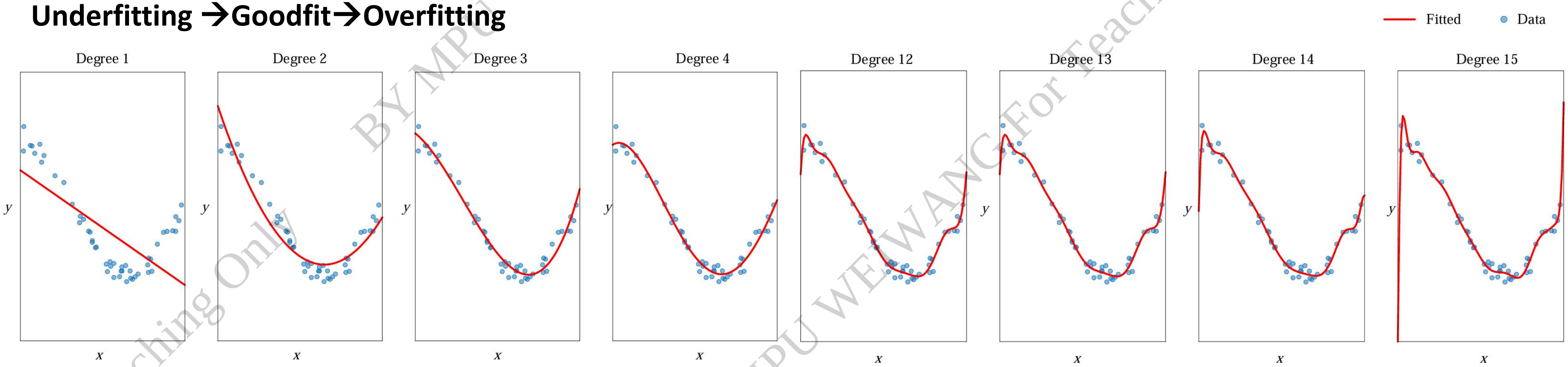


Nonlinear Regression: Polynomial Regression

One-variable function of one to five degrees

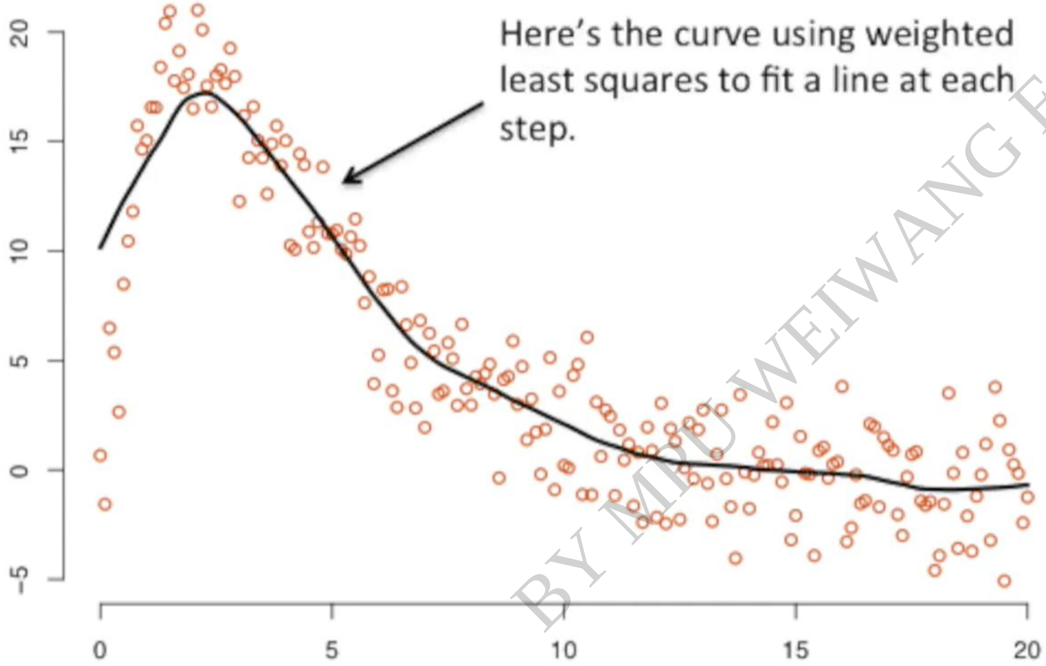


Underfitting →Goodfit→Overfitting



Nonlinear Regression

Nonparametric Regression



Source: https://www.youtube.com/watch?v=Vf7oJ6z2LCc

Parametric regression

- Assumes a fixed functional form
  - ◦Example: linear, polynomial
- Limited flexibility

Key question

- What if we do not know the true form of the relationship?

Nonparametric regression

- Makes minimal assumptions about the data distribution
- Model complexity grows with data

Core idea

- Let the data determine the shape of the regression function

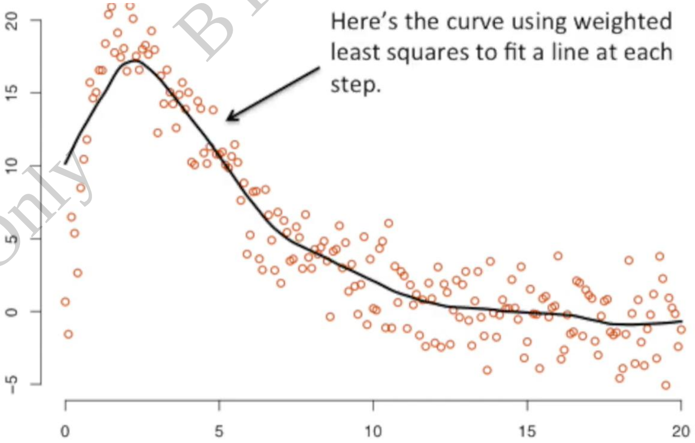
Nonlinear Regression

Nonparametric Regression

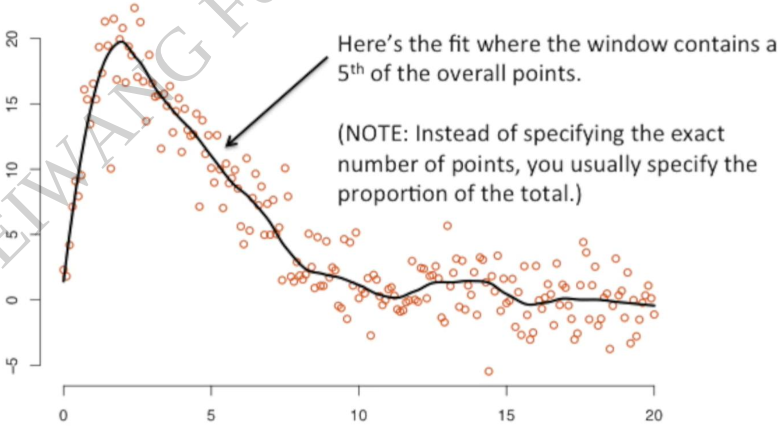
□ Nonparametric regression is a category of regression analysis in which the predictor does not take a predetermined form but is constructed according to information derived from the data.

□ Nonparametric Regression techniques, such as kernel smoothing, splines, or local regression (e.g., LOWESS, LOESS), allow **for capturing complex and flexible nonlinear relationships**. They are particularly useful when the relationship is unknown or cannot be adequately described by a specific parametric equation.

□ LOESS (Locally Weighted Scatterplot Smoothing) regression, also known as LOWESS (Locally Weighted Scatterplot Smoothing) regression, is a non-parametric regression method used to fit a smooth curve to a scatterplot of data points. Unlike traditional regression methods that assume a global relationship between the variables, LOESS regression allows for **local variations and non-linear relationships**.



Here's the curve using weighted least squares to fit a line at each step.



Here's the fit where the window contains a 5th of the overall points.

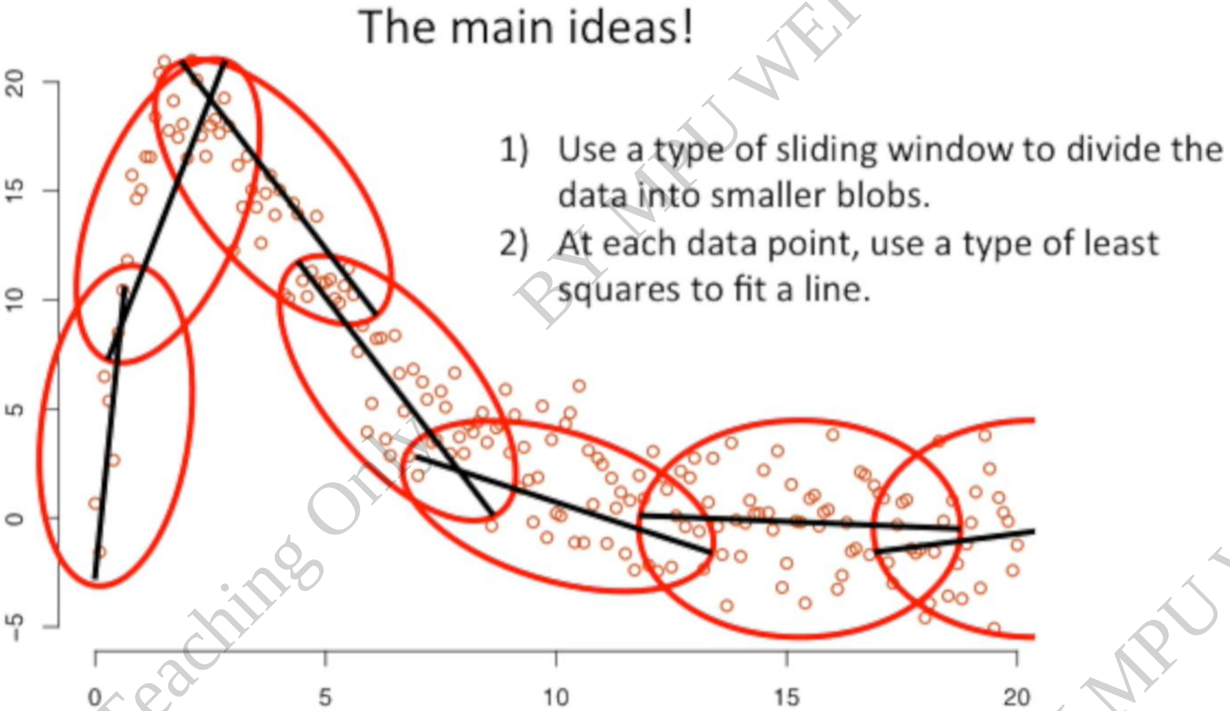
(NOTE: Instead of specifying the exact number of points, you usually specify the proportion of the total.)

Nonlinear Regression

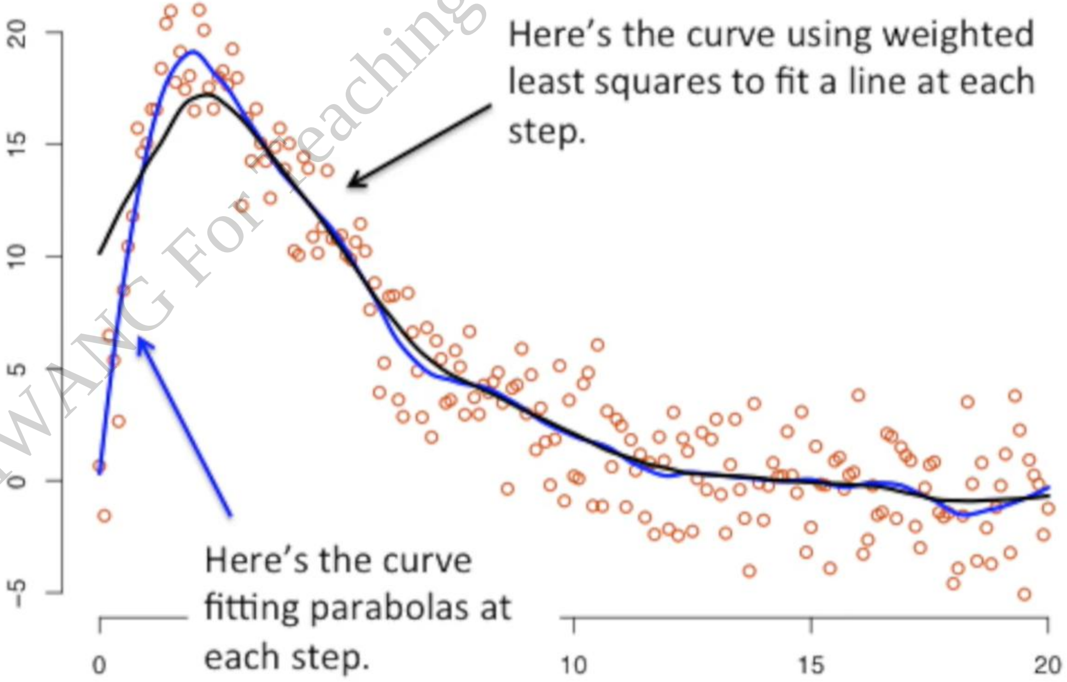
Nonparametric Regression

- □ LOESS regression allows for **local variations and non-linear relationships.** At each point in the range of the data set a low-degree polynomial is fitted to a subset of the data, with explanatory variable values near the point whose response is being estimated. **Parameter: bandwith α.**

$$\mathrm{RSS}_x(A) = \sum_{i=1}^{N} (y_i - A\hat{x}_i)^T w_i(x)(y_i - A\hat{x}_i). \qquad w(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\alpha^2}\right).$$


The main ideas!

1) Use a type of sliding window to divide the data into smaller blobs.
2) At each data point, use a type of least squares to fit a line.


Here's the curve using weighted least squares to fit a line at each step.

Here's the curve fitting parabolas at each step.

https://www.youtube.com/watch?v=Vf7oJ6z2LCc

Why Logistic Regression?

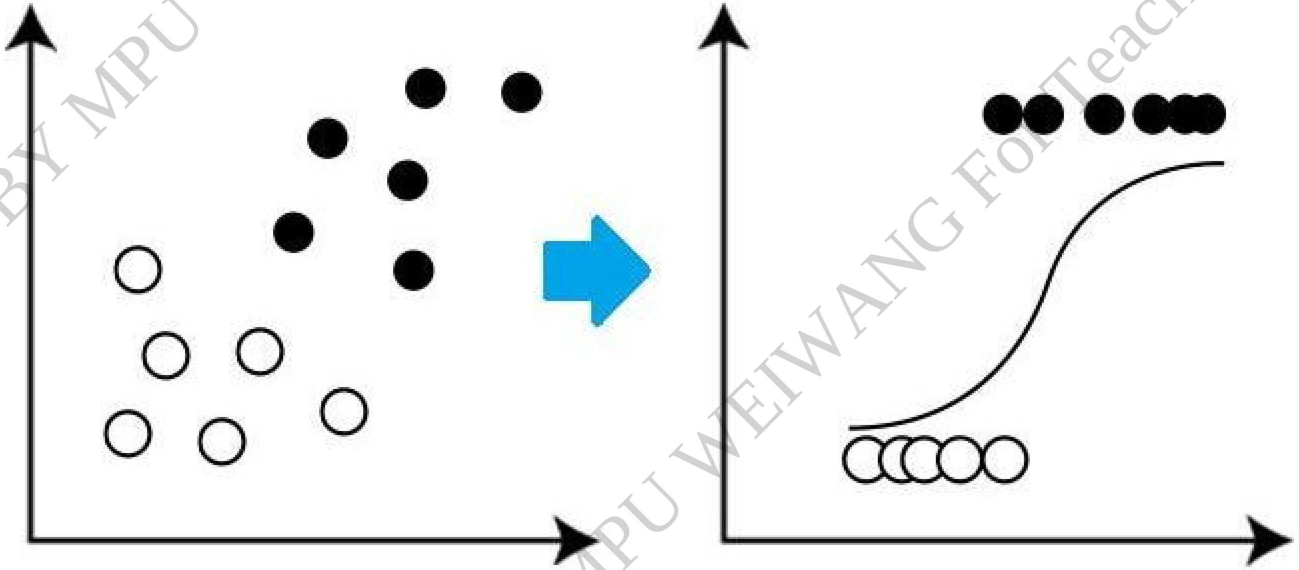
- Many AI problems involve classification, not regression
  - ◦Spam vs. non- spam
  - ◦Disease vs. healthy
  - ◦Fraud vs. normal
- Linear regression is not suitable for classification
  - ◦Outputs are unbounded
  - ◦Cannot represent probabilities
- Logistic regression
  - ◦Models class probabilities
  - ◦One of the most fundamental classification algorithms in AI
- Despite the name:
  - ◦Logistic regression is a classification model

Logistic Regression

- Logistic Regression is a Supervised statistical technique to find the probability of dependent variable(Classes present in the variable).

- Logistic regression uses functions called the ***logit functions***, *that helps* derive a relationship between the dependent variable and independent variables by predicting the probabilities or chances of occurrence.

- The logistic functions (also known as the *sigmoid functions*) convert the probabilities into binary values which could be further used for predictions.



Logistic Regression

Types of Logistic Regression:

- □ **Binary Logistic Regression:**
  The dependent variable has only two 2 possible outcomes/classes.
  Example-Male or Female.

- □ **Multinomial Logistic Regression:**
  The dependent variable has only two 3 or more possible outcomes/classes without ordering.
  Example: Predicting food quality.(Good, Great and Bad).

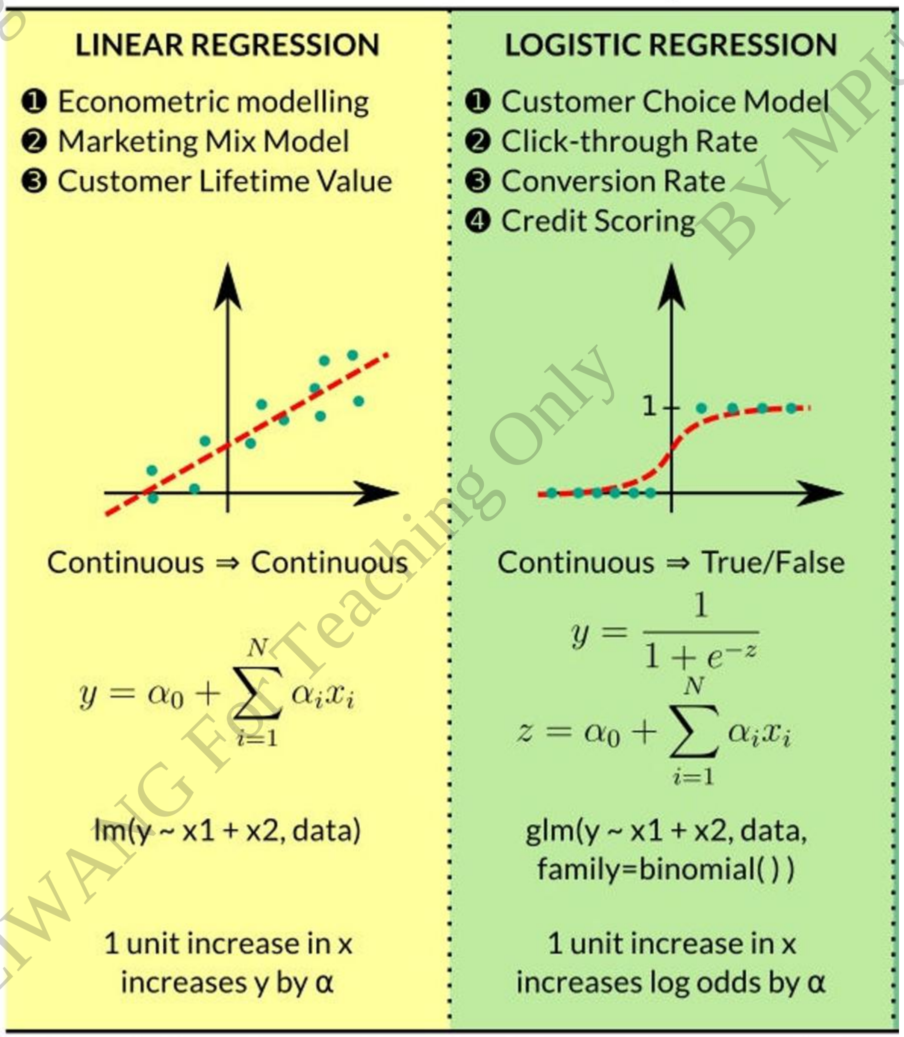
- □ **Ordinal Logistic Regression:**
  The dependent variable has only two 3 or more possible outcomes/classes with ordering.
  Example: Star rating from 1 to 5

Logistic Regression

The logistic model (or logit model) is a statistical model that models the log-odds of an event as a linear combination of one or more independent variables. In regression analysis, logistic regression (or logit regression) estimates the parameters of a logistic model (the coefficients in the linear or non linear combinations).

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

In binary logistic regression there is a single binary dependent variable, coded by an indicator variable, where the two values are labeled "0" and "1", while the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).



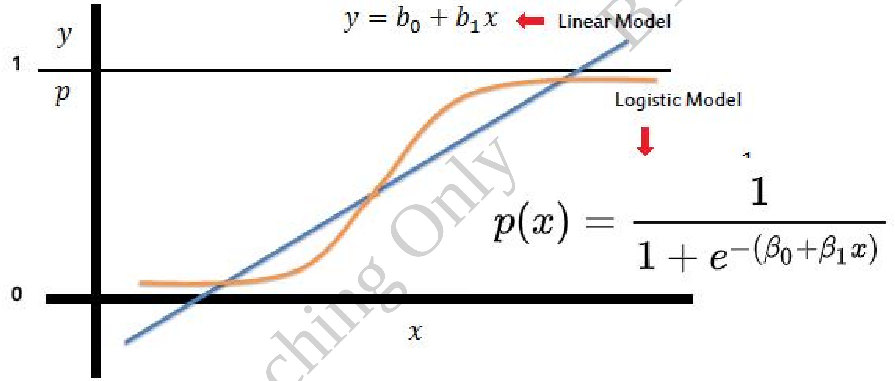
LINEAR REGRESSION	LOGISTIC REGRESSION
❶ Econometric modelling ❷ Marketing Mix Model ❸ Customer Lifetime Value	❶ Customer Choice Model ❷ Click-through Rate ❸ Conversion Rate ❹ Credit Scoring
Continuous ⇒ Continuous	Continuous ⇒ True/False
$y = \alpha_0 + \sum_{i=1}^{N} \alpha_i x_i$	$y = \frac{1}{1 + e^{-z}}$ $z = \alpha_0 + \sum_{i=1}^{N} \alpha_i x_i$
lm(y ~ x1 + x2, data)	glm(y ~ x1 + x2, data, family=binomial( ))
1 unit increase in x increases y by α	1 unit increase in x increases log odds by α

MarketingDistillery.com is a group of practitioners in the area of e-commerce marketing.

Logistic Regression

The Logistic Regression instead for fitting the best fit line, condenses the output of the linear function between 0 and 1.

- □ when **b0+b1X == 0**, then the p will be 0.5,
- □ similarly, **b0+b1X > 0**, then the p will be going towards 1;
- □ **b0+b1X < 0**, then the p will be going towards 0.



$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

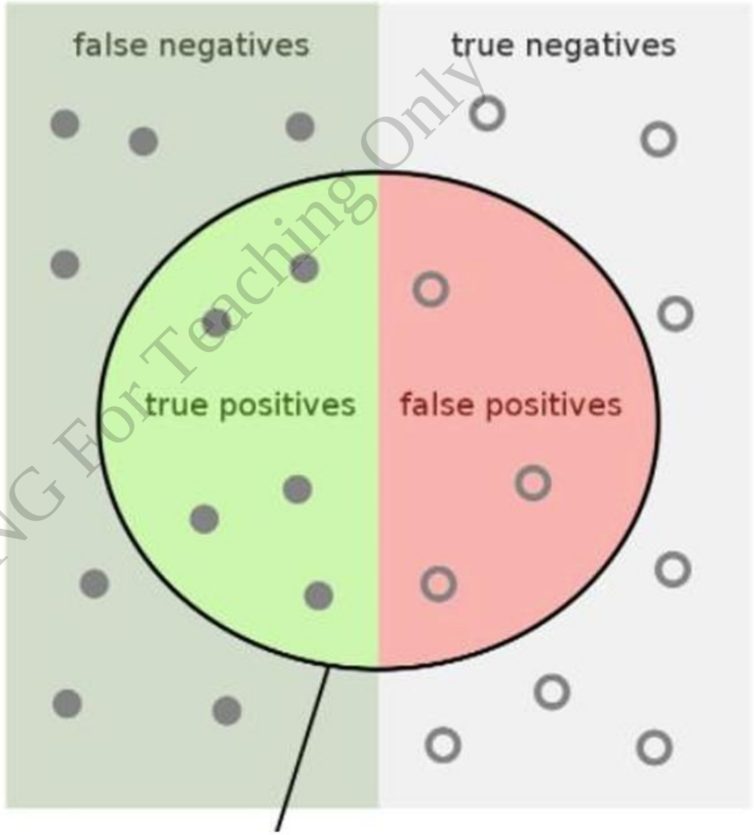
As we Logistic Regression was introduced to tackle the classification problems be it binary classification or multi-class classification problem, ***but why can’t we use Linear Regression?***

- ● Linear Regression predicts continuous variables like price of house, and the output of the Linear Regression can range from negative infinity to positive infinity.
- ● Since, the predicted values is not probability value but a continuous value for the classes, it will be very hard to find the right threshold that can help distinguish between the classes.

Logistic Regression: Evaluation Metrics

Confusion Matrix

- It is important to know how the model make wrong prediction
- In **binary classification**, confusion matrix is a common tool to analyze the predictions



Logistic Regression

Logistic Regression: Evaluation Metrics

TP – True Positive

Definition

The model predicts positive, and the actual class is positive

- Interpretation
  - ◦A correct detection of the positive class
- Example (disease detection)
  - ◦Patient has the disease
  - ◦Model predicts disease

True Negative

Definition

The model predicts negative, and the actual class is negative

- Interpretation
  - ◦A correct rejection
- Example (disease detection)
  - ◦Patient has no disease
  - ◦Model predicts no disease

False Positive

Definition

The model predicts positive, but the actual class is negative

Interpretation

- A false alarm

Example (disease detection)

- Patient has no disease
- Model predicts disease

False Negative

Definition

The model predicts negative, but the actual class is positive

- Interpretation
  - ◦A missed detection
- Example (disease detection)
  - ◦Patient has disease
  - ◦Patient has no disease

Logistic Regression: Evaluation Metrics

Confusion Matrix

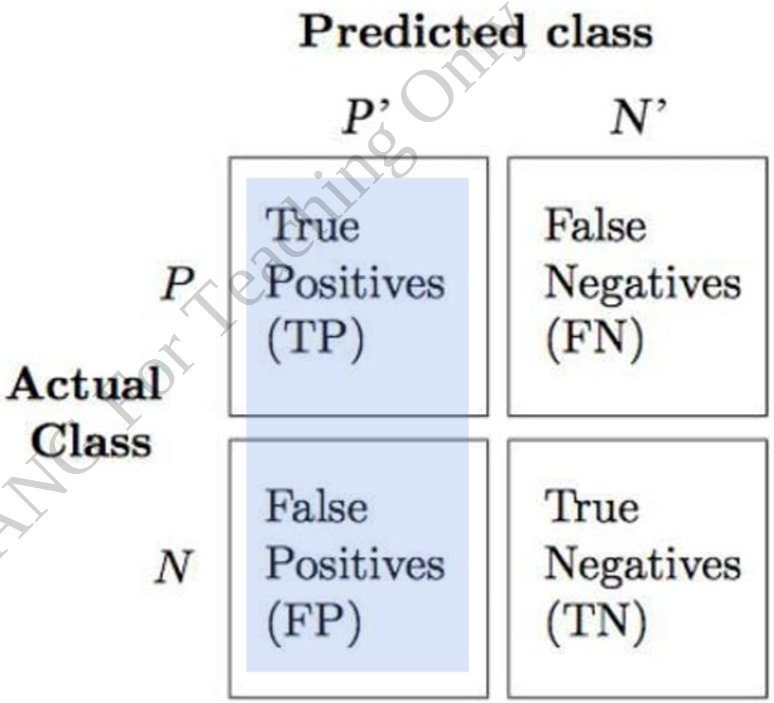
- Precision(PRE) & Recall Rate(REC)

$$\boxed{PRE = \frac{TP}{TP + FP}, \quad \text{(the higher, the better)}}$$

$$REC = \frac{TP}{TP + FN} = TPR. \quad \text{(the higher, the better)}$$

- F-1 Score

$$F_1 = 2\frac{(PRE * REC)}{PRE + REC}, \quad \text{(the higher, the better)}$$



		Predicted class	
		P'	N'
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Logistic Regression

Logistic Regression: Evaluation Metrics

Confusion Matrix

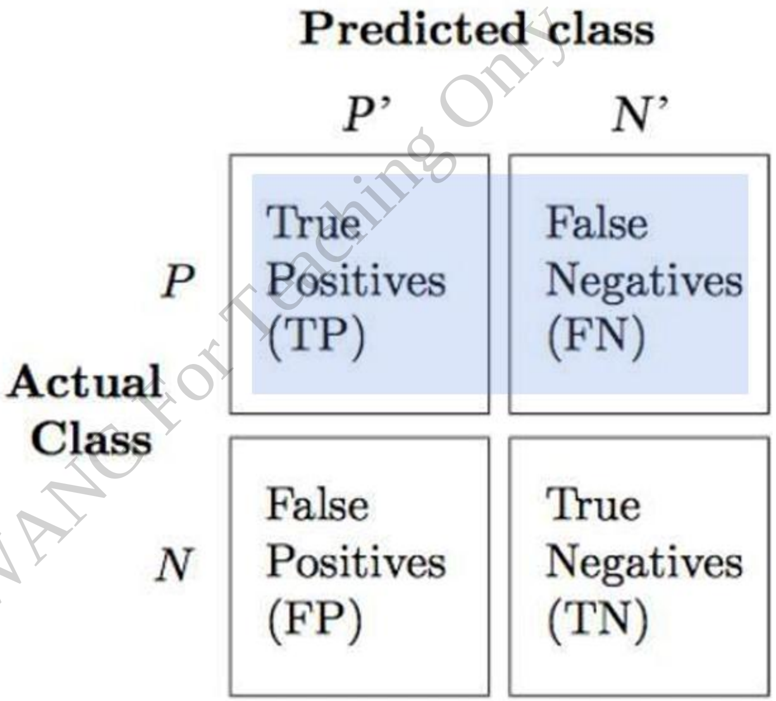
- Precision(PRE) & Recall Rate(REC)

$$PRE = \frac{TP}{TP + FP}, \quad \text{(the higher, the better)}$$

$$\boxed{REC = \frac{TP}{TP + FN} = TPR. \quad \text{(the higher, the better)}}$$

- F-1 Score

$$F_1 = 2\frac{(PRE * REC)}{PRE + REC}, \quad \text{(the higher, the better)}$$



Predicted class

Actual Class		P'	N'
	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Logistic Regression

Logistic Regression: Evaluation Metrics

ROC Curve

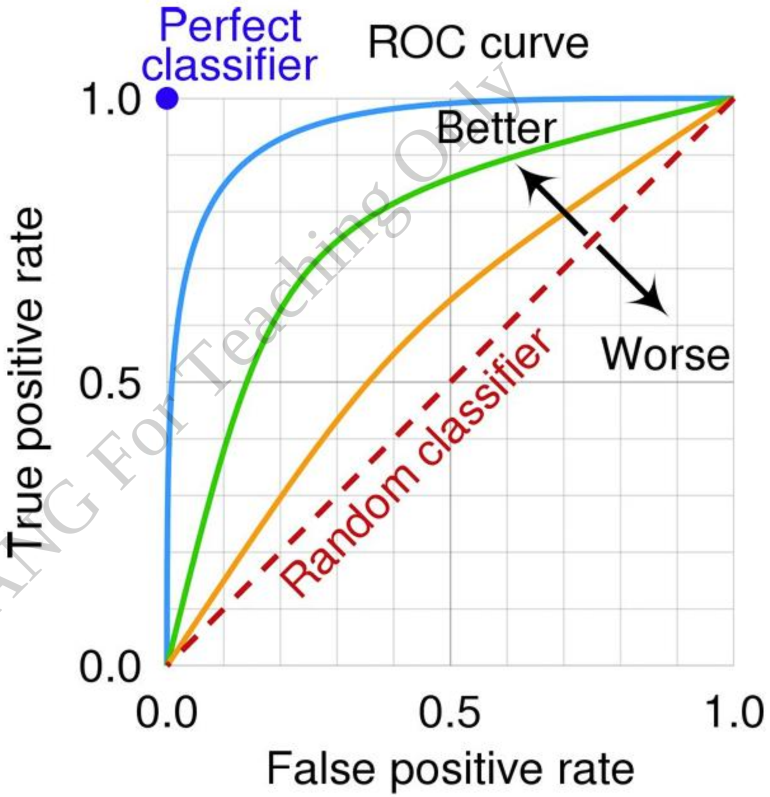
- ROC curve analyze the performance for **every threshold in soft classifiers**
- In X-axis $$FPR = \frac{FP}{FP + TN}$$
- In Y-axis $$TPR = \frac{TP}{TP + FN}$$

1	
1	
0.87	θ
0.64	⇓
⋮	
−0.88	
−0.93	
−1	

Logistic Regression: Evaluation Metrics

ROC Curve

- ROC curve analyze the performance for **every threshold in soft classifiers**
- In X-axis $FPR = \frac{FP}{FP + TN}$
- In Y-axis $TPR = \frac{TP}{TP + FN}$

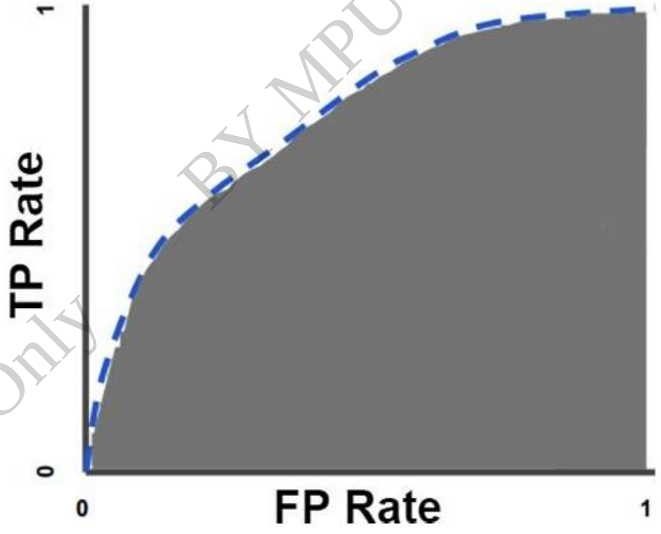


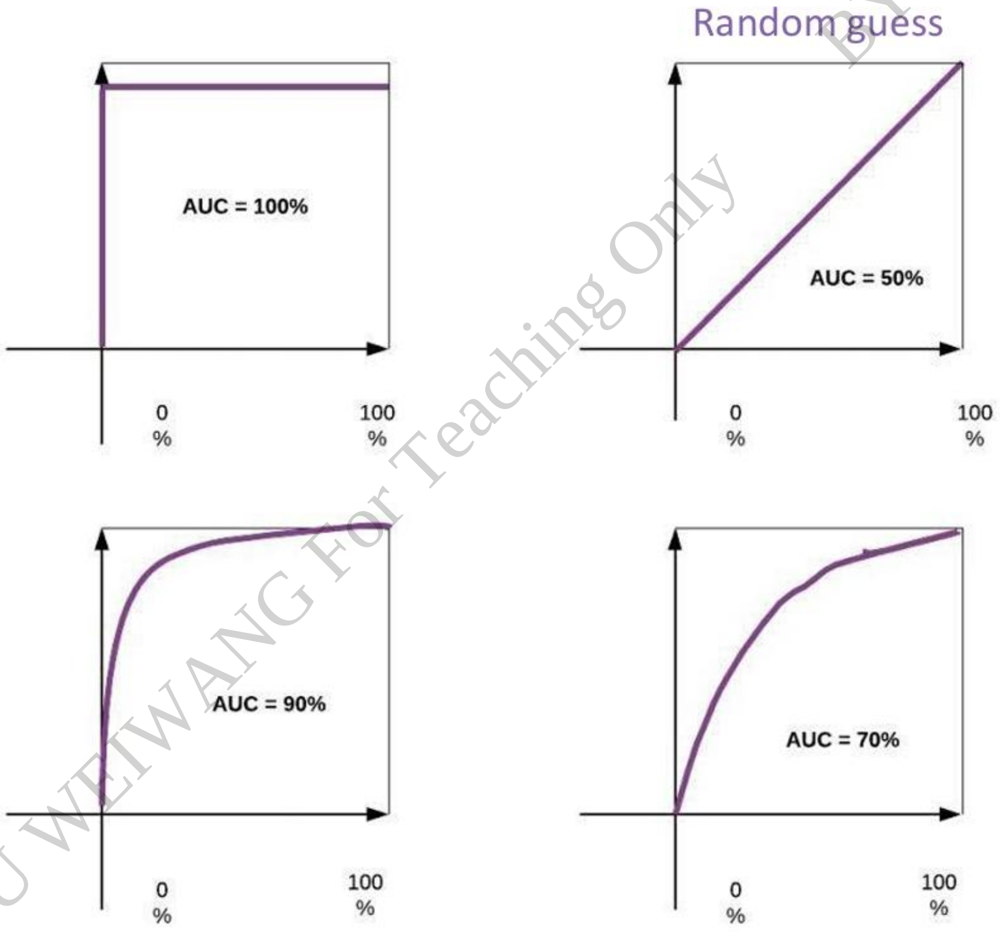
https://www.youtube.com/watch?v=QBVzZBsif20

Logistic Regression: Evaluation Metrics

AUC

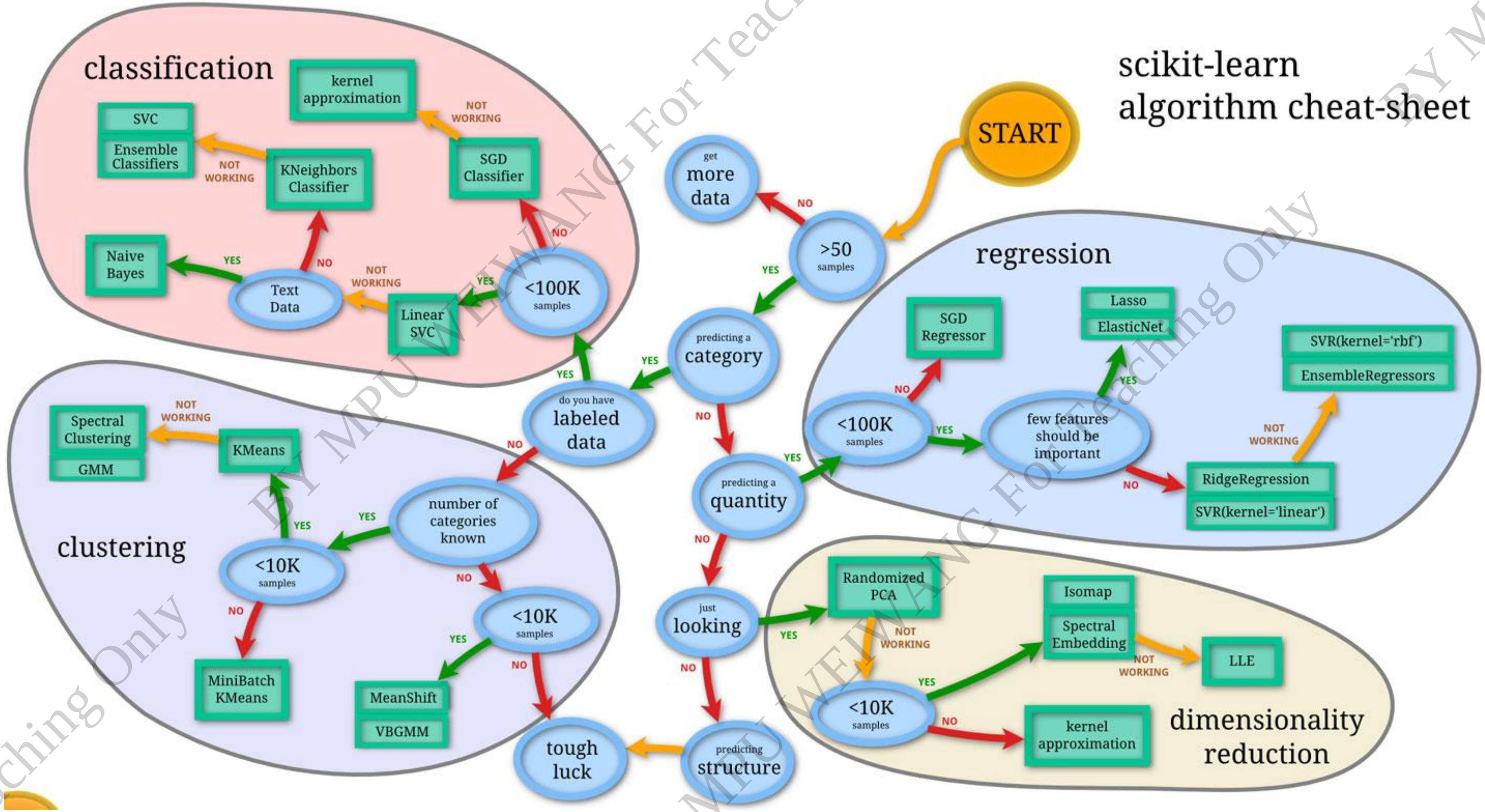
- AUC (Area Under the ROC Curve)
  - ROC can be quantified using AUC



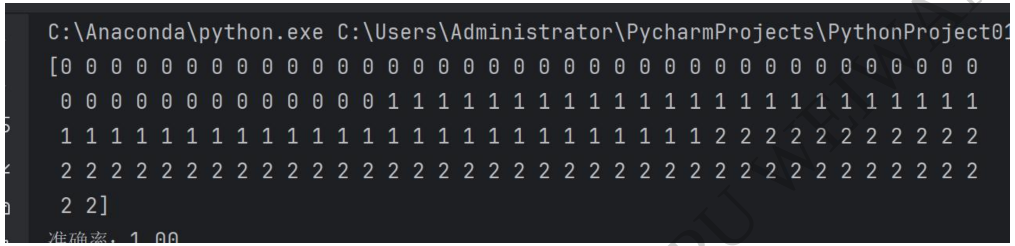


Logistic Regression

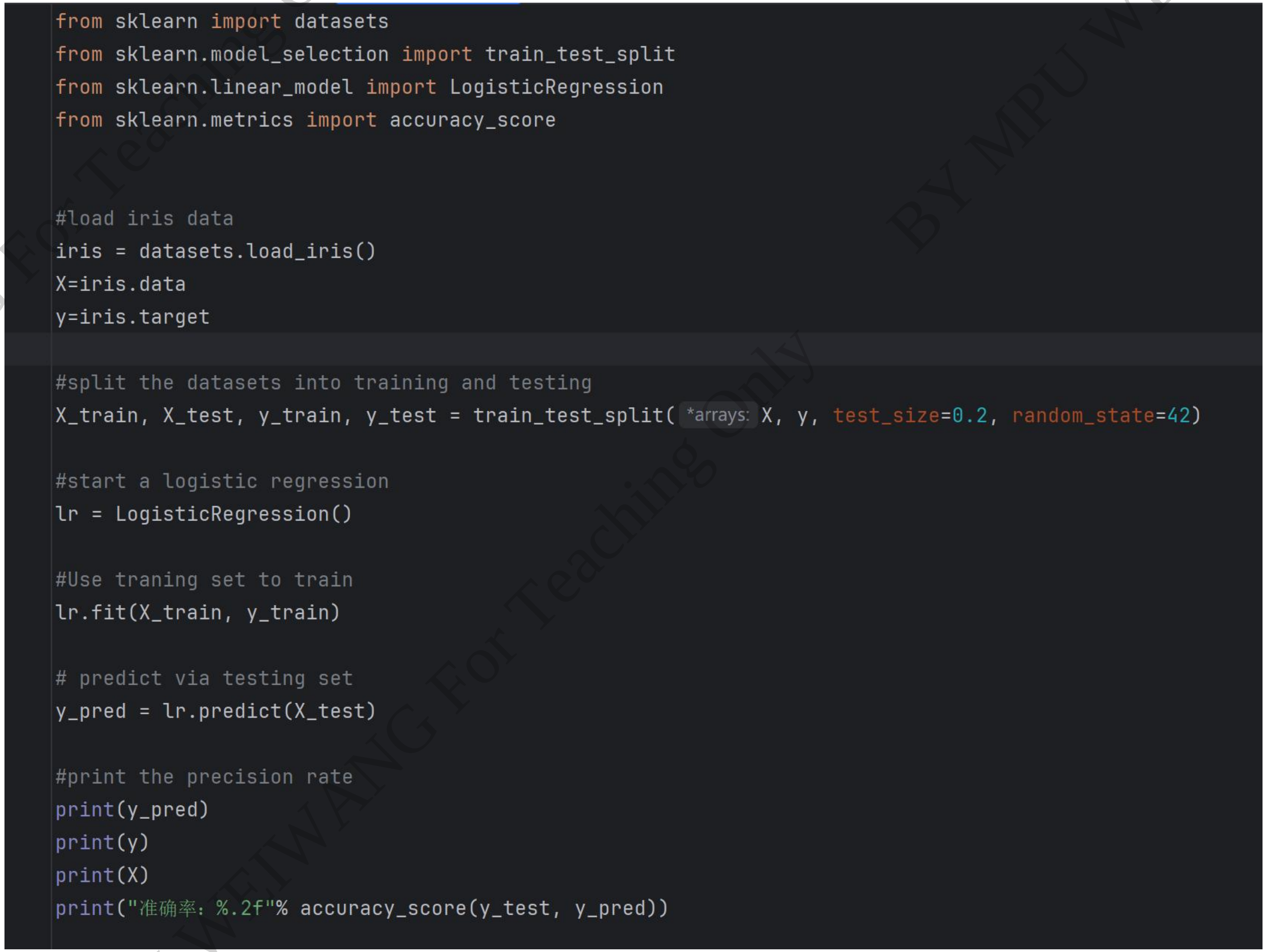
Logistic Regression: Evaluation Metrics



Run the Logistic Regression of Iris



```
C:\Anaconda\python.exe C:\Users\Administrator\PycharmProjects\PythonProject01
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
准确率: 1.00
```



```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score


#load iris data
iris = datasets.load_iris()
X=iris.data
y=iris.target

#split the datasets into training and testing
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)

#start a logistic regression
lr = LogisticRegression()

#Use traning set to train
lr.fit(X_train, y_train)

# predict via testing set
y_pred = lr.predict(X_test)

#print the precision rate
print(y_pred)
print(y)
print(X)
print("准确率: %.2f"% accuracy_score(y_test, y_pred))
```

Anaconda

IDE: Pycharm

Run the Logistic Regression of Iris

https://mirror.tuna.tsinghua.edu.cn/help/anaconda/

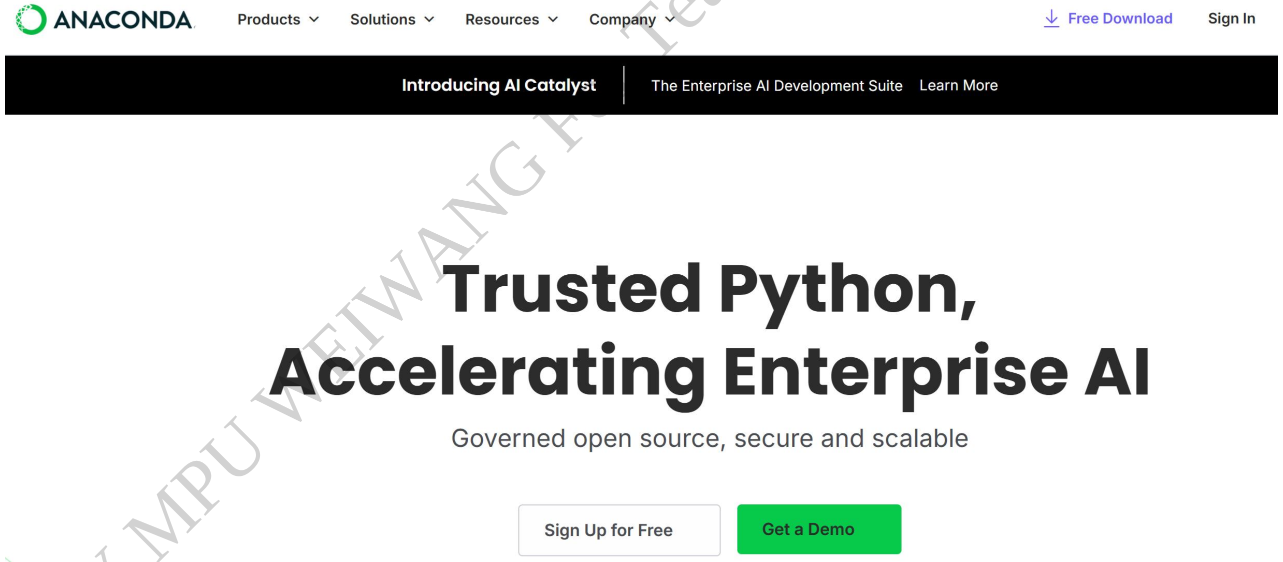
1. What is Anaconda?

Anaconda is a Python data science distribution, essentially an ‘integrated toolkit’. Its core components include:

- **Python interpreter**: The foundational program for directly executing Python code.
- **Hundreds of pre-installed libraries**: Covering common tools for data processing, numerical computation, visualisation, machine learning, and more.

Conda package manager: for installing, updating, and uninstalling libraries, offering greater capabilities than Python's built-in pip (manages non-Python dependencies); Environment management tool: enables creation of multiple independent virtual environments (e.g., one environment using Python 3.8, another using 3.10), preventing dependency conflicts between different projects. 2. How to install Anaconda

1. Download from the official website
2. Download via mirror sites



Trusted Python, Accelerating Enterprise AI

Governed open source, secure and scalable

Anaconda Install

Run the Logistic Regression of Iris

https://www.jetbrains.com/pycharm/

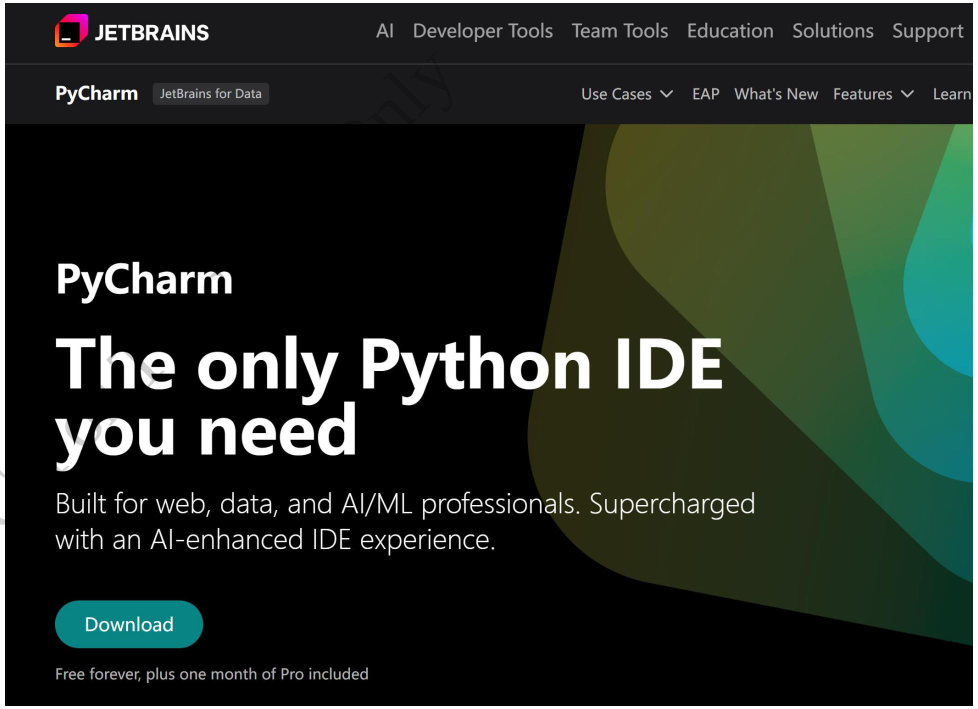
What is PyCharm?

PyCharm is a professional Python integrated development environment (IDE) developed by JetBrains, specifically designed for the Python programming language.
It offers extensive features and tools aimed at enhancing developers' programming efficiency and code quality.

It is widely used across various Python development domains, including web development, data analysis, artificial intelligence, and scientific computing.

It is available in two editions:
Community Edition and **Professional Edition**.

Opt for a **more stable cracked version** during PyCharm installation.



PyCharm

The only Python IDE you need

Built for web, data, and AI/ML professionals. Supercharged with an AI-enhanced IDE experience.

Free forever, plus one month of Pro included

Pycharm Install

Thank you!

Innovating into the Future