



3099704: AI for Digital Health

## Regression

Prof. Peerapon Vateekul, Ph.D.

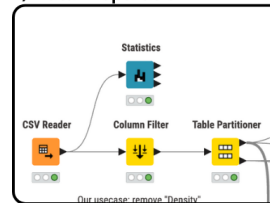
[Peerapon.v@chula.ac.th](mailto:Peerapon.v@chula.ac.th)



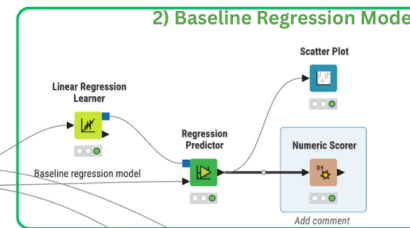
# Outlines

- Supervised Learning (recap)
- Linear Regression
- Handle Categorical Variables
- Feature Selection
- Regression Performance

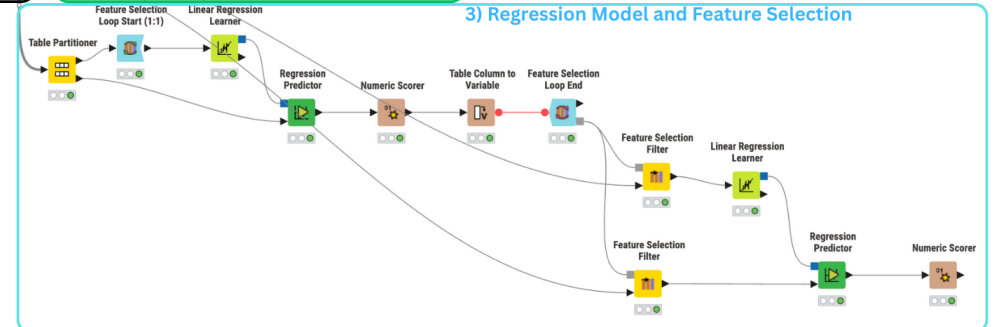
1) Data Preparation



2) Baseline Regression Model



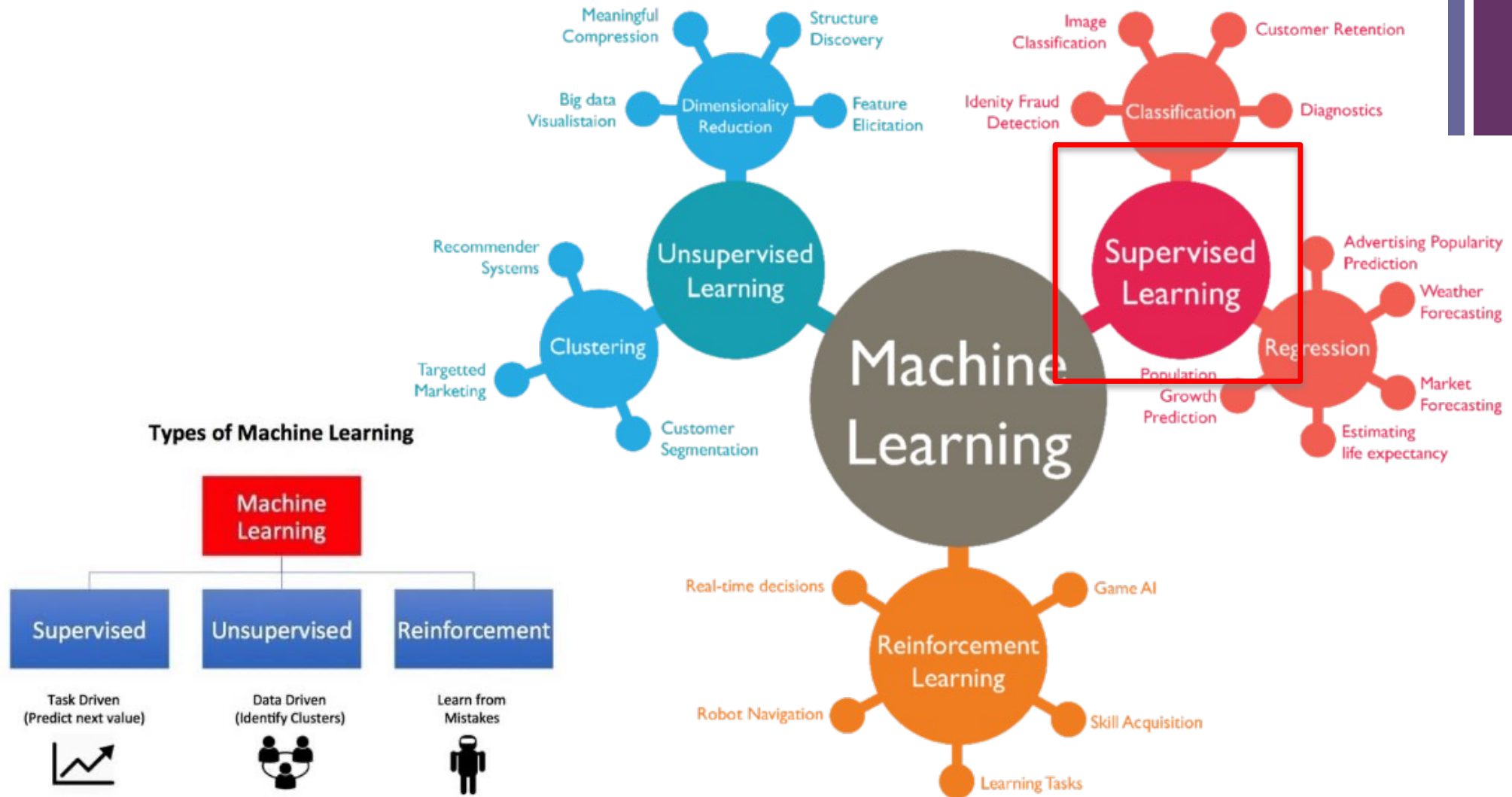
3) Regression Model and Feature Selection





# Supervised Learning (recap)

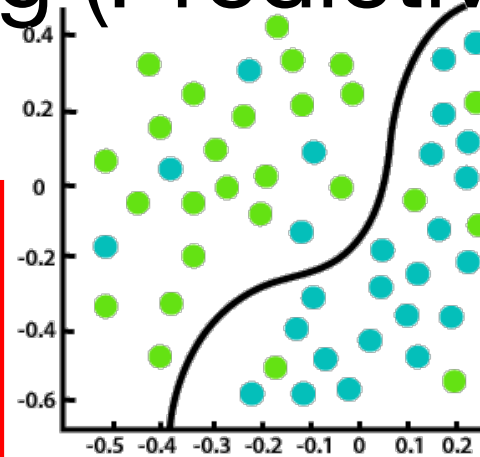
# + Machine Learning



# Supervised Learning (Predictive Task)

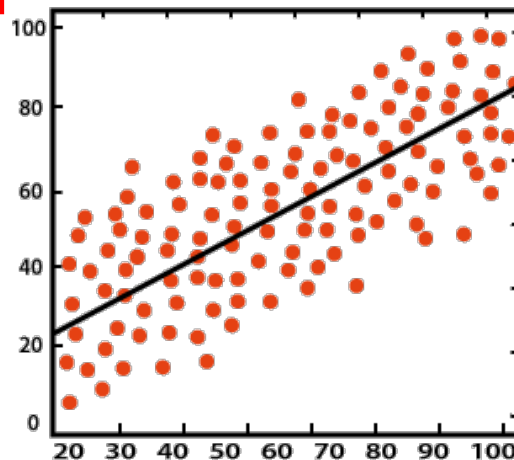
inputs				target
Age	Temp	Gender	Smell	Covid
25	39.0	Female	No	Yes
35	38.9	Female	No	Yes
32	36.5	Male	Yes	No

- Goal: To learn a prediction model mapping from inputs to output.
- Data without label (answer) is meaningless!
- Label should be provided by experts!



- Target is categorical variable.
- Example
- Covid diagnosis (yes/no)
- Disease diagnosis from gait information:
  - 1) Normal,
  - 2) Sick/Knee OA
  - 3) Sick/Parkinson

Classification



Regression

- Target is numeric variable.
- Example
- PD's state diagnosis from movement data.
- Glucose level prediction from breath particles.



# There are two main processes: Train/Test

## 1) Training Phase: Model Construction

Training Data



Age	Income	inputs		target
		Gender	Province	Purchase
25	25,000	Female	Bangkok	Yes
35	50,000	Female	Nontaburi	Yes
32	35,000	Male	Bangkok	No

## 2) Testing Phase: Model Evaluation, Model Assessment Also called “prediction, inference, scoring”

Testing Data




Age	Income	Gender	Province	Purchase
25	25,000	Female	Bangkok	?

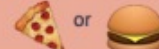


# Prediction Algorithms


- Decision Tree
- (Logistic) Regression
- Neural Networks (NN)
- kNN
- Support Vector Machine
- Deep Learning


### BASIC REGRESSION


**LINEAR** `linear_model.LinearRegression()`  
Lots of numerical data 


**LOGISTIC** `linear_model.LogisticRegression()`  
Target variable is categorical 


### CLASSIFICATION


**NEURAL NET** `neural_network.MLPClassifier()`  
Complex relationships. Prone to overfitting  
Basically magic. 

**K-NN** `neighbors.KNeighborsClassifier()`  
Group membership based on proximity 

**DECISION TREE** `tree.DecisionTreeClassifier()`  
If/then/else. Non-contiguous data  
Can also be regression 

**RANDOM FOREST** `ensemble.RandomForestClassifier()`  
Find best split randomly  
Can also be regression 

**SVM** `svm.SVC()` `svm.LinearSVC()`  
Maximum margin classifier. Fundamental  
Data Science algorithm 

**NAIVE BAYES** `GaussianNB()` `MultinomialNB()` `BernoulliNB()`  
Updating knowledge step by step with new info 



# Linear Regression

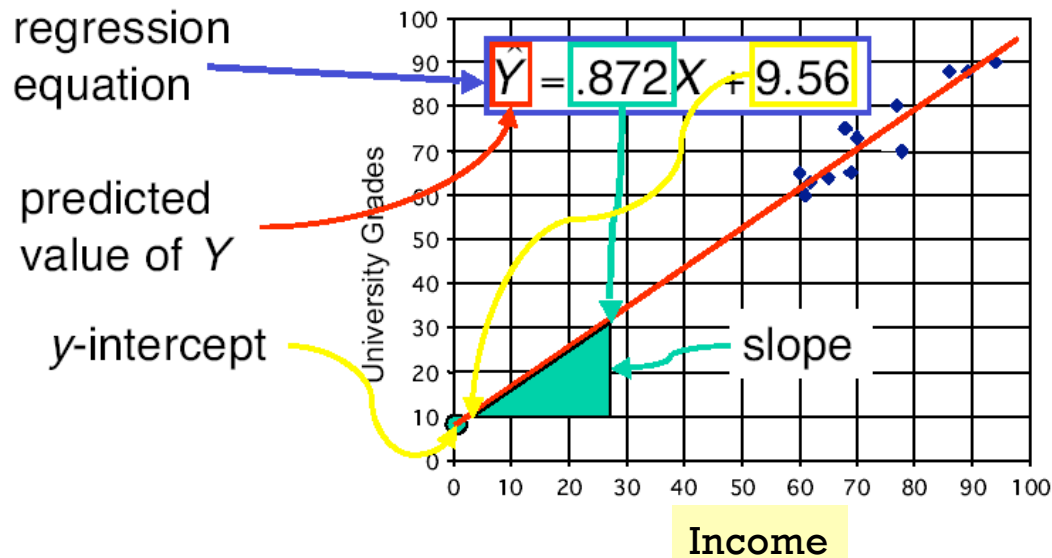




# Linear Regression

Spending

Relation between High School and University Grades



weight, coefficient

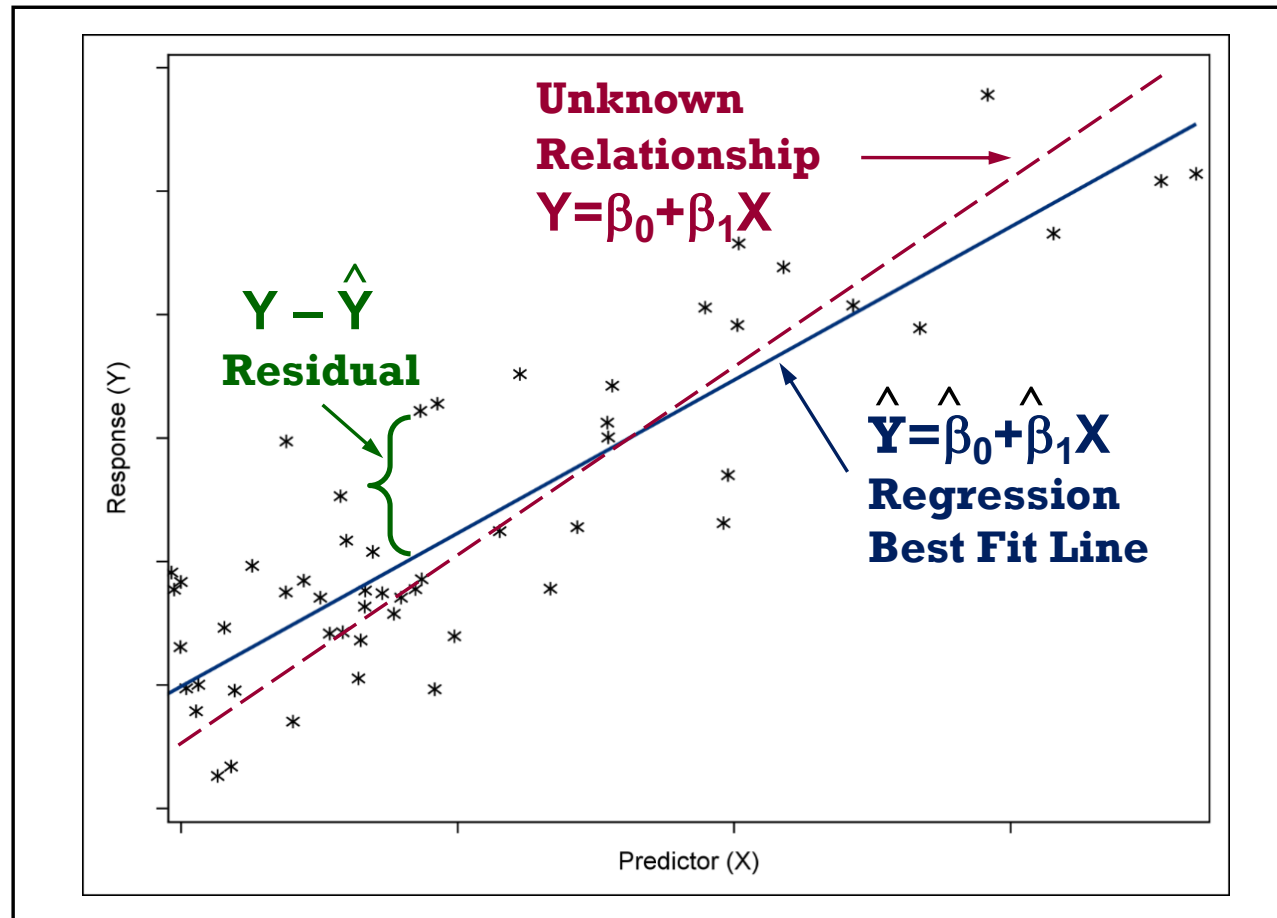
$$\hat{y} = \hat{w}_0 + \hat{w}_1 x_1 + \hat{w}_2 x_2$$

target      intercept      input

- The least square method aims to minimize the following term

$$\sum_{\text{training data}} (y_i - \hat{y}_i)^2$$

# Ordinary Least Squares (OLS) Regression



# + Multiple Linear Regression Model Matrix Multiplication Approach

	inputs	target
Age	Income in k\$	Spending
25	25	400
35	50	500
32	35	550

$$Y = \beta_0(1) + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

$$[Y] = [X][\beta]$$

$$[\beta] = [X]^{-1}[Y]$$

General least-squares solution:

$$\beta = (X^T X)^{-1} X^T y$$

$$y = \begin{bmatrix} 400 \\ 500 \\ 550 \end{bmatrix} \quad X = \begin{bmatrix} 1 & 25 & 25 \\ 1 & 35 & 50 \\ 1 & 32 & 35 \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

In this toy example,  $X$  is  $3 \times 3$  and invertible, so it fits **exactly** and you can also use:

$$\beta = X^{-1}y$$

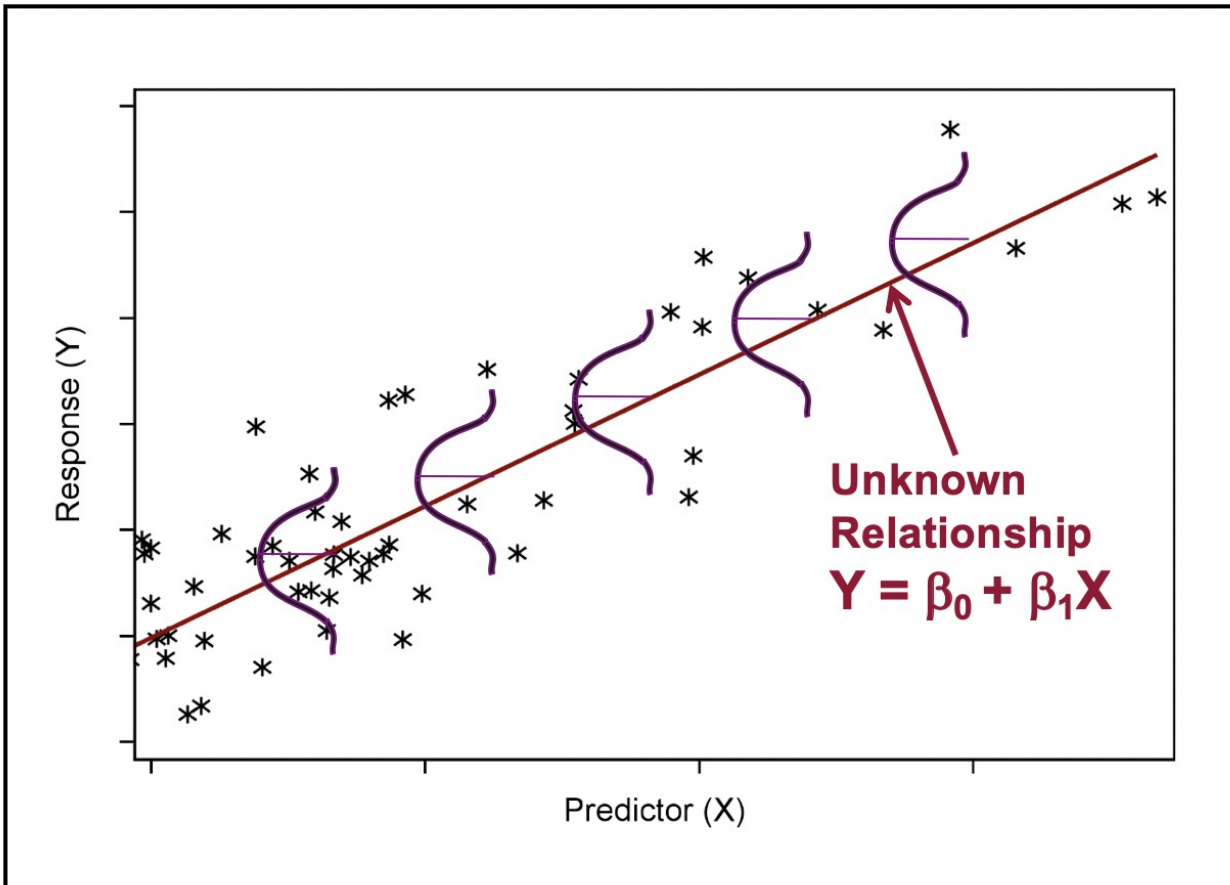
$$\beta = \begin{bmatrix} -250 \\ \frac{110}{3} \\ -\frac{32}{3} \end{bmatrix} \approx \begin{bmatrix} -250 \\ 36.6667 \\ -10.6667 \end{bmatrix}$$

$$\widehat{Spending} = -250 + 36.6667(\text{Age}) - 10.6667(\text{Income in k\$})$$



# Linear Regression Assumption

$$\text{Spending} = 500 + 10 * \text{Income10K} + 2 * \text{Age}$$



- Linear relationship between  $(y, x_i)$ .  
[Pearson correlation]
- Error is normal distributed. [remove outliers, log-transformation]
- Error has equal variance (homoscedasticity) [remove outliers, log-transformation]
- Errors are independent from each other. [design new data correction]



# Linear Regression Limitation

13

- Manage missing value
- Handle outliers (skewness)
- Handle categorical variables (dummy codes)
- Variable selection:
  - Wrapper (all combinations)
  - Forward, Backward, Stepwise
- Accounted for nonlinearities



## Remarks:

- Require good data preparation
- Variable selection

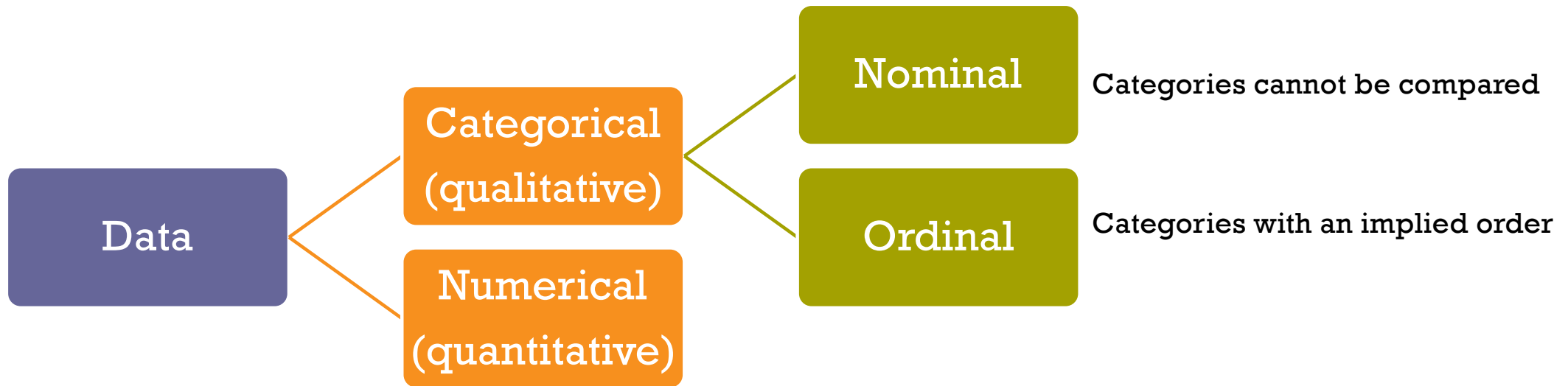


# Handle Categorical Variables

Dummy coding, OneHotEncoder

## + Terminology: Kinds of data (recap)

15





# Ordinal: Recode

Grade	GradeNum
A	4.00
B+	3.50
B	3.00
C+	2.50
C	2.00
D+	1.50
D	1.00
F	0.00

Size	SizeNum
XL	5
L	4
M	3
S	2
XS	1





# Categorical

- Dummy coding = (n-1) dummy codes

Branch	BranchNum	D_BKK	B_Patum	D_Non	D_BKK	B_Patum	
BKK	1	1	0	0	1	0	
Patumtani	2	0	1	0	0	1	
Nontaburi	3	0	0	1	0	0	reference



# Feature Selection



# Feature Selection Approach

- Sequential Feature Selection:
  - Sequential Feature Selection
  - Backward Feature Elimination
- Best Subset Selection (create all combinations & pick the best one)

# + Sequential Feature Selection

Start with no features, then add the feature that improves performance the most at each step.

## Step 0: No features

- Model: intercept only
- RMSE = 120

## Step 1: Try adding one feature at a time

Feature added	RMSE
Age	80
Income	60 ✓
Education	95
Gender	110

➡ Select Income

## Step 2: Try adding one more feature (given Income)

Features	RMSE
Income + Age	45 ✓
Income + Education	58
Income + Gender	62

➡ Add Age

## Step 3: Try adding another feature

Features	RMSE
Income + Age + Education	46
Income + Age + Gender	47

➡ No improvement → STOP

Goal: Predict *Spending*

Candidate features:

Age, Income, Education, Gender

20

## ✓ Final selected features

code

Income, Age

# + Backward Feature Elimination

Backward elimination starts with all features and iteratively removes the least useful one based on performance impact.

**Goal:** Predict *Spending*

**Candidate features:**

Age, Income, Education, Gender

**Step 0 — Start with ALL features**

code

Age, Income, Education, Gender

Baseline: RMSE = 60

**Step 1 — Try removing ONE feature at a time**

Feature removed	Remaining features	Val RMSE
Age	Income, Education, Gender	95 ✖
Income	Age, Education, Gender	110 ✖
Education	Age, Income, Gender	62 ✖
Gender	Age, Income, Education	58 ✔ (better)

✔ **Decision:** remove **Gender** (RMSE improves: 60 → 58)

**Step 2 — Current set**

code

Age, Income, Education

Baseline: RMSE = 58

Try removing one:

Feature removed	Remaining features	Val RMSE
Age	Income, Education	88 ✖
Income	Age, Education	105 ✖
Education	Age, Income	57 ✔ (better)

✔ **Decision:** remove **Education** (58 → 57)

**Step 3 — Current set**

code

Age, Income

Baseline: RMSE = 57

Try removing one:

Feature removed	Remaining features	Val RMSE
Age	Income	80 ✖
Income	Age	92 ✖

🛑 **Decision:** STOP

✔ **Final selected features**

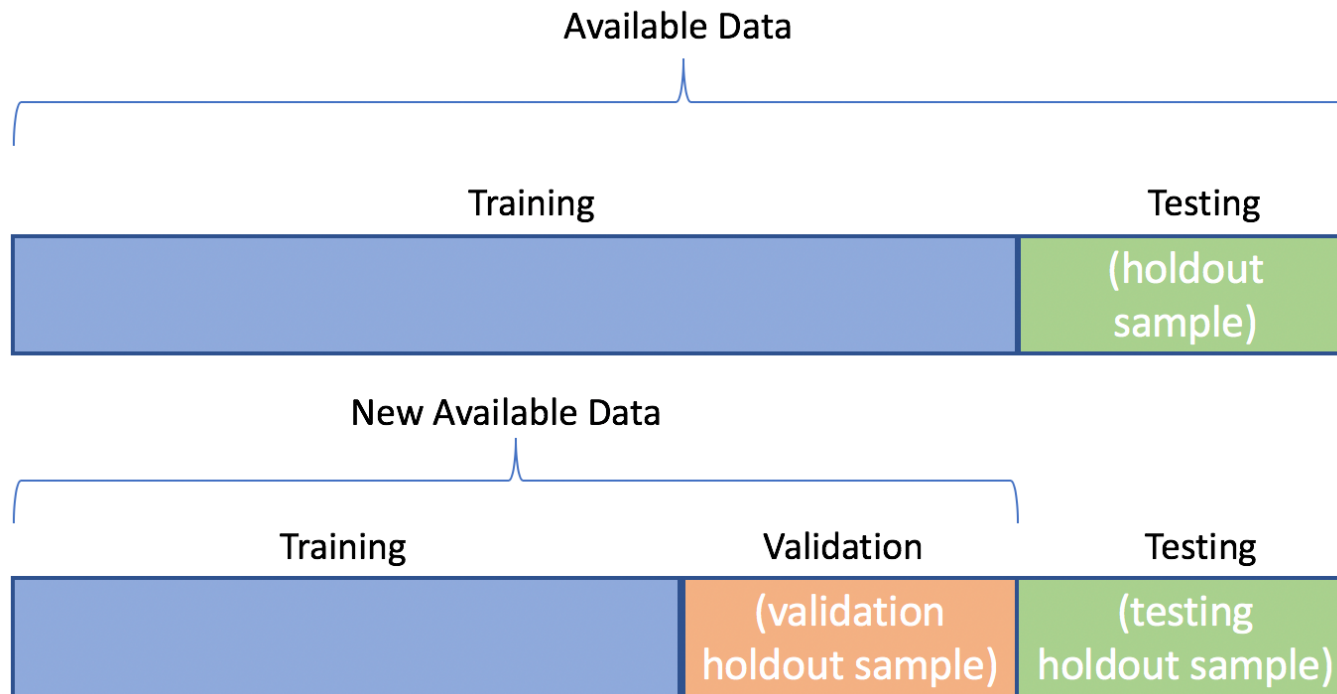
code

Income, Age



# + Train (Validation) & Test

- Training data = Textbook
- Validation data = Exercise
- Testing data = Final exam





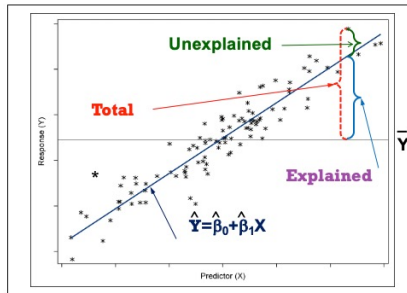
## Regression Performance

### Regression

'mean_absolute_error'	<code>sklearn.metrics.mean_absolute_error</code>
'mean_squared_error'	<code>sklearn.metrics.mean_squared_error</code>
'r2'	<code>sklearn.metrics.r2_score</code>



# Coefficient of Determination



## Regression

'mean\_absolute\_error' `sklearn.metrics.mean_absolute_error`

'mean\_squared\_error' `sklearn.metrics.mean_squared_error`

'r2' `sklearn.metrics.r2_score`

$$R^2 = \frac{SSM}{SST} = 1 - \frac{SSE}{SST}$$

id	chol (x)	bp (y)	predict	error	squred error (SE)	guess	(y - y_bar )	squred total (ST)
1	437	194	196.1897	(2.1897)	4.7948	143.4286	50.5714	2,557.4694
2	264	121	141.4179	(20.4179)	416.8906	143.4286	(22.4286)	503.0408
3	249	131	136.6689	(5.6689)	32.1364	143.4286	(12.4286)	154.4694
4	297	159	151.8657	7.1343	50.8982	143.4286	15.5714	242.4694
5	243	123	134.7693	(11.7693)	138.5164	143.4286	(20.4286)	417.3265
6	272	161	143.9507	17.0493	290.6786	143.4286	17.5714	308.7551
7	161	115	108.8081	6.1919	38.3396	143.4286	(28.4286)	808.1837
average	274.7143	143.4286		SSE	972.2548		SST	4,991.7143
				MSE	138.8935			
				<b>RMSE</b>	<b>11.7853</b>			
	<b>R^2</b>	<b>1 - (SSE/SST)</b>	<b>0.8052</b>					



Thank you & any questions