




# Essential ML 101

☰ Tags	ML
➤ Class	
☑ Finished Yet?	☑
➤ Knowledge	 <u>The Seventh Sprint: Machine Learning</u>

-pdf:

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d6df573b-d5ac-42a5-8baf-d2592e782b4c/ML\\_Crash\\_Course.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d6df573b-d5ac-42a5-8baf-d2592e782b4c/ML_Crash_Course.pdf)

\*ข้อมูลเพิ่มเติม : <https://www.notion.so/cf0acfa15ac540029aec565e9d39f984?v=d01679b413e846d19d668c0d1bbe2e52&pvs=4>

## Lesson 1: What is ML?

Field of study that gives computers the ability to learn without being explicitly programmed  
-Arthur Samuel (1959)

-ML ทำให้คอมพิวเตอร์สามารถเรียนรู้จากประสบการณ์ได้

-ML อยู่ทุกที่

-คอร์สนี้จะโฟกัสที่ Classical ML (Linear Regression, Logistic Regression, Regularized Regression, Decision Tree, KNN (K-Nearest Neighbors), Simple Neural Networks)

## Lesson 2: Model = Algorithm + Data

-Algorithm + Data = Model

-Algorithm Keywords: rules, instructions, procedure, steps

-Algorithm Example:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. KNN

-Model = f(Data) [function = algorithm]

-R is a great example:

```
#lm = linear regression  
#Apply function to data to create model  
model = lm(mpg ~ hp + wt + am, data = mtcars)
```

-บางโมเดลเขียนเป็นสมการได้ (Parametric) บางโมเดลก็เป็น Non-Parametric เช่น Rule-based (เขียน if-else เพื่อ classify data เราเป็นส่วน ๆ)

-Linear regression algorithm จะค้นหาค่า  $b_0$  และ  $b_1$  (เรียกว่าค่า weight) ที่ทำให้สมการมี Error ต่ำที่สุด

-ขั้นตอนการทำงานของ Linear Regression:

1. สุ่มค่า  $b_0$ ,  $b_1$  ขึ้นมา

2. ปรับค่า  $b_0$ ,  $b_1$  ตาม learning rate (step ในการปรับค่า weight เช่น 0.001 เป็นต้น ปรับเปลี่ยนได้)
  3. คำนวณค่า Error (ถ้าค่า Error ลดลง = มาถูกทาง ทำต่อไป แต่ถ้าค่า Error เพิ่มขึ้น = มาผิดทาง เปลี่ยนทิศทางการปรับค่า)
  4. ถ้า Error ลดต่ำสุดแล้วให้จบการทำงาน (Convergence)
- 

## Lesson 3: Types of ML

-ML จะมีหลัก ๆ 3 ประเภทคือ

1. Supervised Learning (Predict Data)
  2. Unsupervised Learning (Summarize Data)
  3. Reinforcement Learning (Bot Training)
- 

## Lesson 4: Supervised Learning

-Supervised Learning (Predictive model/Predictive analytics): Learn from labeled data to make a prediction

-ใช้ตัวแปรต้น (x) ตัวหนึ่งหรือหลาย ๆ ตัวเพื่อทำนายตัวแปรตาม (y)

-ตัวแปรตามที่มีอยู่ก่อนเรียกว่า labeled data (เรียนรู้ x และ y จากข้อมูลในอดีต พอเจอ x ในอนาคต ก็จะสามารถทำนาย y ในอนาคตได้)

-มี 2 ประเภทย่อย ๆ คือ

1. Classification (Category) label มีลักษณะเป็น Discrete มีค่าที่แน่นอน (finite) เช่น เป็น 0 หรือ 1 เท่านั้น
  2. Regression (Numeric) label มีลักษณะเป็น Continuous จะมีค่าไหนก็ได้ (infinite)
- 

## Lesson 5: Unsupervised Learning

-Unsupervised Learning (Summarize model): Group and summarize data

-Algorithm ที่ได้รับความนิยมที่สุดคือ K-Means Clustering (Kmeans)

-Kmeans มีหน้าที่หลักในการหากลุ่ม segments หรือ clusters ใน dataset ด้วยการคำนวณระยะห่าง (default: Euclidean distance) ของทุก ๆ data points

-การใช้งานอื่น ๆ จะมี Clustering (การจัดกลุ่มลูกค้าหรือกลุ่มเป้าหมาย), Association (Market basket analysis สมมติว่าถ้าลูกค้าหยิบนมกับไข่ขึ้นมา แล้วมีแนวโน้มว่าจะหยิบไก่ทอดขึ้นมาด้วยหรือไม่ ใช้ในการทำนายแนวโน้มการซื้อของผู้กันหรือร่วมกันของลูกค้า), Dimension Reduction (ลดจำนวน column ของ dataset ให้อยู่ในระดับที่เราจัดการได้ เรียกว่า component ซึ่งนำไปใช้ในการทำ ML Model ต่อไปได้)

-Data Science = Interdisciplinary (ความรู้แบบบูรณาการ ผสมหลายศาสตร์เข้าด้วยกัน เช่น ในกรณีของ Data Science คือ สถิติ + วิทยาศาสตร์คอมพิวเตอร์ + โปรแกรมมิ่ง)

---

## Lesson 6: Reinforcement Learning

-Reinforcement Learning ใกล้เคียงกับคำว่า AI มากที่สุด

-Parameter ต่าง ๆ ใน Reinforcement Learning:

1. Agent
2. Environment
3. Actions
4. Reward/ Penalty
5. Observation

-Agent จะสำรวจ Environment และเลือกทุก ๆ Action ที่จะทำ

-ทุก ๆ Action ที่เลือก จะมี Reward หรือ Penalty เกิดขึ้นมา

-เป้าหมายของ Agent คือการ Maximize reward หรือ Minimize risk ในระยะยาว เช่น AlphaGo หรือ AI ในบอร์ดเกมต่าง ๆ เป็นต้น

---

## Lesson 7: Basic ML Workflow

-ML Workflow ที่เรียบง่ายที่สุดคือ

1. Split Data (Train - Test)
2. Train Model

### 3. Score Model

### 4. Evaluate Model

-สัดส่วนการ Train - Test ไม่มีกฎตายตัว

-เป้าหมายของการสร้าง ML Model คือการนำโมเดลไปใช้กับข้อมูลใหม่ที่ยังไม่เคยเห็นมาก่อนได้  
นี่คือเหตุผลที่ต้องทำ Test Model (เหมือนเรียนแล้วไปสอบ)

---

## Lesson 8: Regression Example

-ใน R มี caret ที่สามารถใช้ Train regression model ได้

-caret ย่อมาจาก [C]lassification [A]nd [RE]gression [T]ree

-Split → Train → Test → Evaluate

-หน้าที่หลักของ Data Analyst คือการรับ Business requirements และเลือกใช้ Algorithms ให้เหมาะสมกับงานนั้นๆ เพื่อช่วยแก้ปัญหาให้กับธุรกิจ

```
#Load caret
library(caret)

#Load dataset
mtcars

#Create function to split data
train_test_split <- function(data) {
  set.seed(42)
  n <- nrow(data)
  id <- sample(n, size=0.8 * n)
  train_data <- data[id, ]
  test_data <- data[-id, ]
  return(list(train_data, test_data))
}

#Split dataset by using created function
split_data <- train_test_split(mtcars)

#Train model
lm_model <- train(mpg ~ hp,
  data = split_data[[1]],
  method = "lm")

#Score and evaluate model
p <- predict(lm_model, newdata = split_data[[2]])

error <- split_data[[2]]$mpg - p
rmse <- sqrt(mean(error ** 2))
```

\*No one size fits all solution (ตอนทำงานจริง แต่ละ Business problems จะมีความแตกต่างกันไป ดังนั้น Algorithms ที่เคยทำงานได้ดีกับปัญหา Churn problem อาจจะทำไม่ได้กับปัญหาแบบ Product recommendation)

-โอเดียสำคัญของ ML Projects คือการทำ Experimentation (การทดลอง)

-โมเดลจะทำงานได้ดีหรือไม่ขึ้นอยู่กับหลายปัจจัย เช่น Data quality, Sample size, Bias, Model tuning, Performance metric เป็นต้น

---

## Lesson 9: Classification Example

-ก่อนจะสร้างโมเดล ต้องเปลี่ยน column ที่เราต้องการทำนายเป็น factor

-glm = Generalized linear model

-ใช้ Accuracy วัดผลแทน RMSE

-Overfitting คือการที่ Algorithm ของเราเรียนรู้ Training data ดีเกินไป จนไม่สามารถนำไปประยุกต์หรือทำนายข้อมูลใหม่ ๆ ได้ เช่น Test Accuracy จะมีค่าต่ำลงเยอะ หรือ Test Error มีค่าสูงขึ้นมาก

```
#Load caret
library(caret)

#Load clean data
data("mtcars")

#Prepare data
mtcars$am <- factor(mtcars$am, levels=c(0,1), labels=c("Auto", "Manual"))

#Split data
split_data <- train_test_split(mtcars)

#Train model
glm_model <- train(am ~ mpg, #Classification
                  data = split_data[[1]],
                  method = "glm")

#Score and evaluate model
p <- predict(glm_model, newdata=split_data[[2]])

#Use accuracy as a metric to evaluate model
acc <- mean(p == split_data[[2]]$am)
```

# Lesson 10: Course Summary

- ML algorithm learns from data
  - Supervised learning learns from labeled data
  - Unsupervised learning learns from unlabeled data
  - Basic ML workflow: Split Data → Train → Score → Evaluate
  - Use caret to build models in R
-