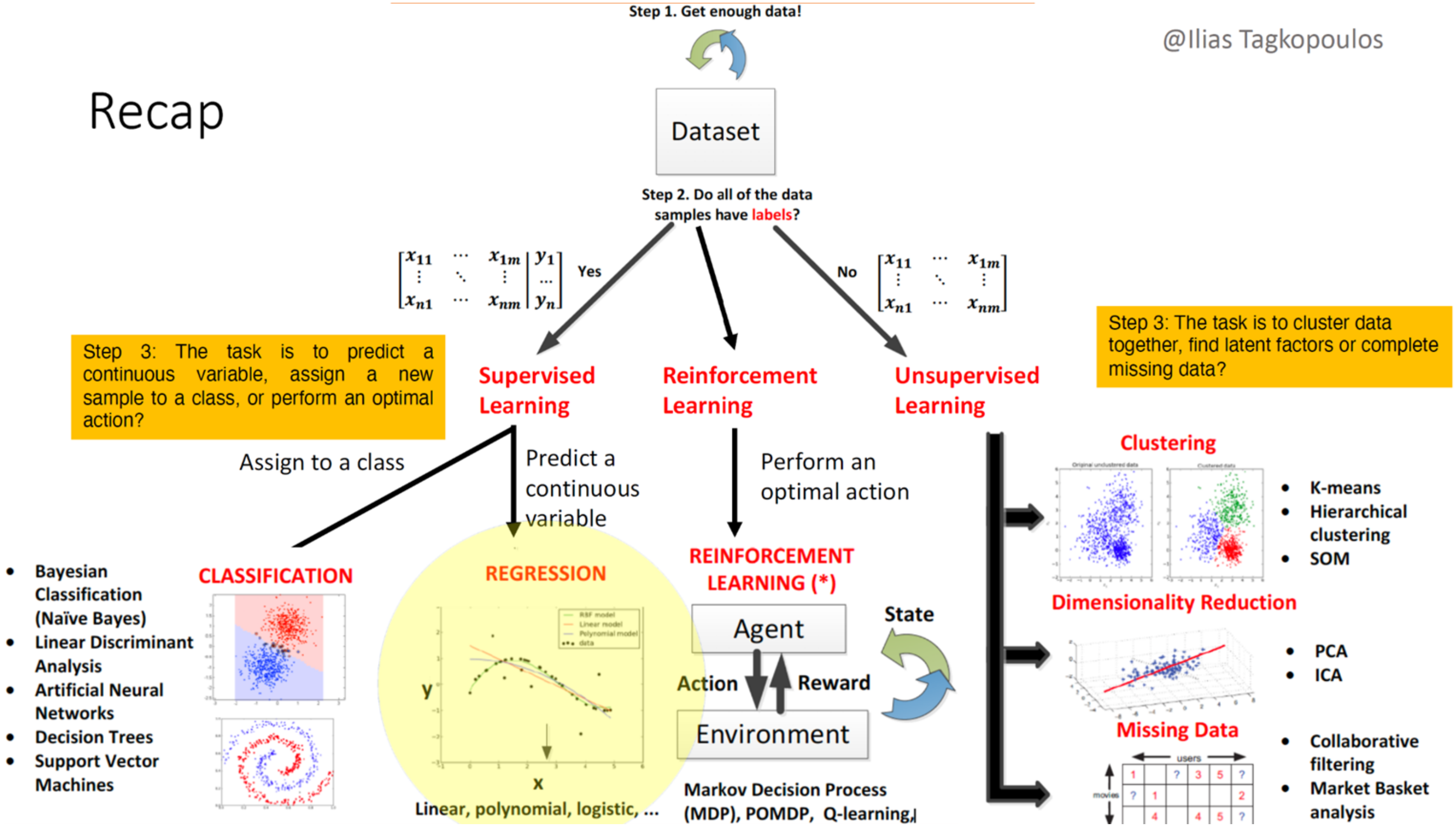
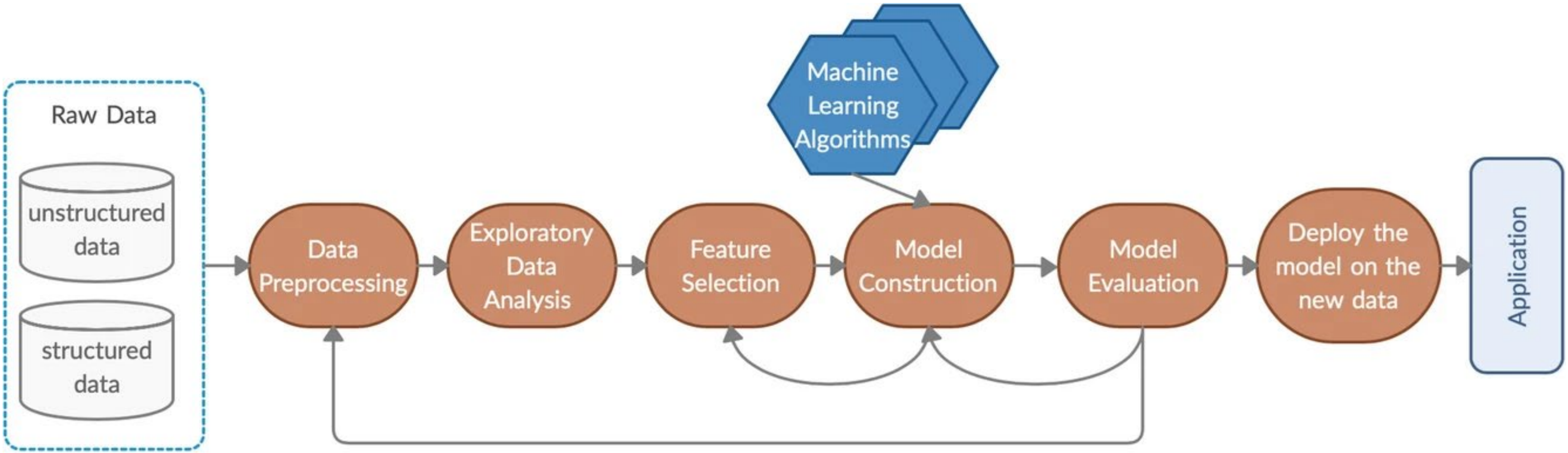
ECS 171: Machine Learning

Summer 2023
Edwin Solares
easolares@ucdavis.edu
Linear Regression

What is Machine Learning: Recap



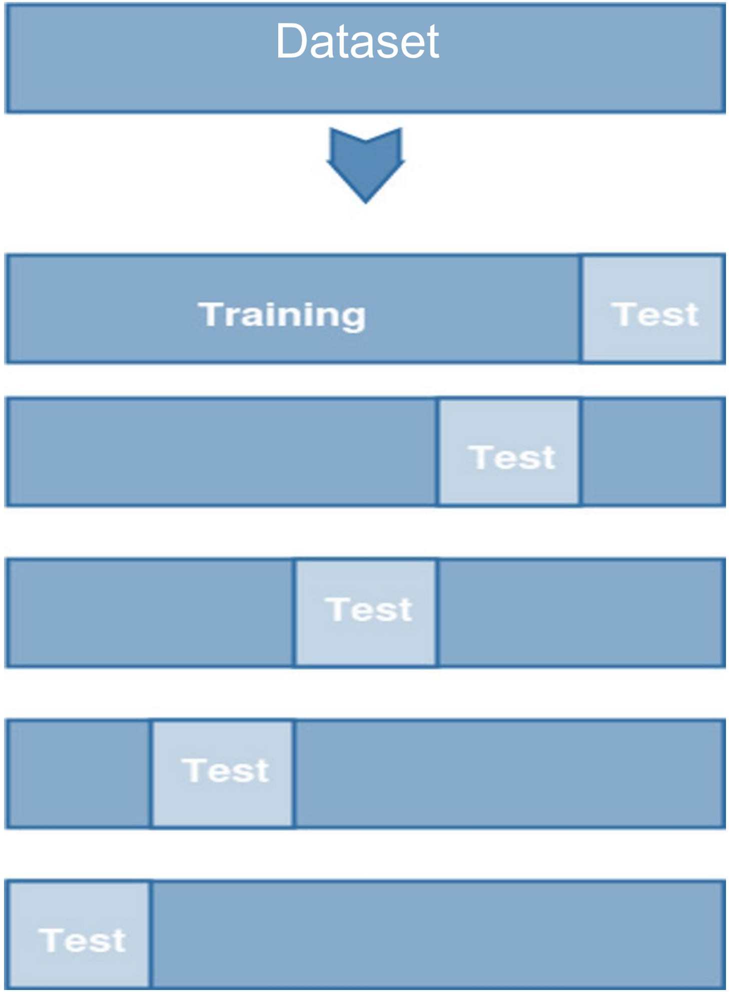
Machine Learning Pipeline



Cross Validation

```
from sklearn import cross_validation

# value of K is 5.
data =
cross_validation.KFold(len(train_set)
, n_folds=5, indices=False)
```



Regression Problem Example?

Predicting sales for a particular product

Data set Description

- Attribute(s) of the data set (**X**) includes
  - advertising budget (dollar value)
- Output **y** i.e., the class attribute
  - sales in thousands of units

Linear regression: find a linear relationship between ***X*** (input) and y (output).

Goal: find ***f(X)* = *ŷ*** → ***y***

Advertisement budget (**independent variable**) ***X***

Output sales (**dependent variable**) ***y***

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X)* = *ŷ*** → ***y***
For all seen ***X*** and unseen ***X***

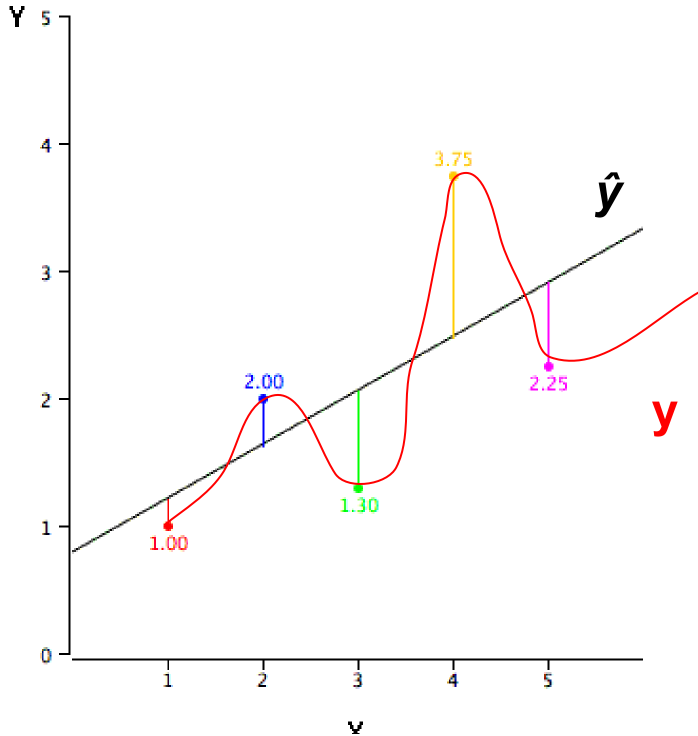
Linear Regression Model

Supervised learning

Popular statistical learning method

Predicts a quantitative response **y** from predictive attribute **X**

Linear relationship between **X** and **y**



$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

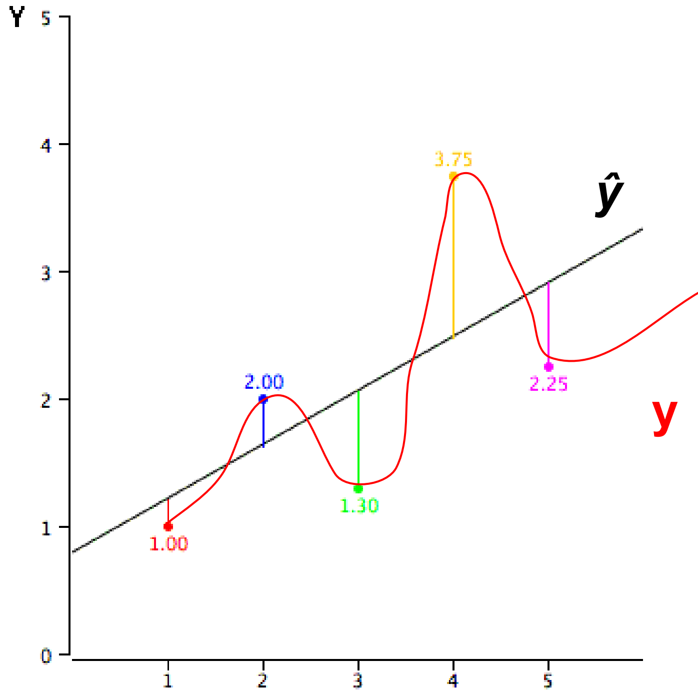
Output

intercept

model coefficients (model parameters)

Cost Function

When **training** the model, the goal is to **minimize** the **error** and **update** the model **coefficients** to achieve the **best fit** line.



Error is the **difference between predicted value** (Y) generated by the model and the **class attribute value**.

Cost function L is used to **measure the error**:

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

Observed value   Predicted value

Minimize the Loss!

Method 1: Ordinary Least Squares (OLS)

Method 2: Gradient Descent (GD)

Tabular Data → Matrix → Formula

m-by-n matrix

$a_{i,j}$ n columns j changes

m rows

i changes

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

For m rows, ith row in m rows

For n columns, jth column in n columns

$a_{i,j} = x_{i,j}$

Linear Regression: Formulation

Given a dataset/matrix **M** with $\boldsymbol{m}$ observations:

M = { (x_1, y_1) | 1 ≤ i ≤ m } Where x_i = {$x_{i,1}$, $x_{i,2}$, … , $x_{i,\mathrm{n}}$,}; n # of attributes

Note: matrix **M** is of size $\boldsymbol{m}$ x $\boldsymbol{n}$ (rows x cols)

$$For\ i = 1$$

$$w_{1,0}x_{1,0} + w_{1,1}x_{1,1} + w_{1,2}x_{1,2} + \ldots + w_{1,n}x_{1,n} = \sum_{j=0}^{n} w_{1,j}x_{1,j}$$

Weights

$$w = \Theta = \beta$$

$$M = \left\{ \begin{pmatrix} 1 & x_{1,1} & \ldots & x_{1,n} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_{m,1} & \ldots & x_{m,n} \end{pmatrix} \begin{pmatrix} y_1 \\ \ldots \\ y_n \end{pmatrix} \right\}$$

Linear Regression: Visualization

m-by-*n* matrix

n columns — *j* changes

y

m rows — *i* changes

Description	Guests	Seat class	Customer ID	Fare	Age	Title	Success
Braund, Mr. Owen Harris; 22	1	3	1	7.25	22	Mr	0
Cumings, Mrs. John Bradley ...	1	1	2	71.3	38	Mrs	1
Heikkinen, Miss. Laina; 26	0	3	3	7.92	26	Miss	1
Futrelle, Mrs. Jacques Heath...	1	1	4	53.1	35	Mrs	1
Allen, Mr. William Henry...	0	3	5	8.05	35	Mr	0
Moran, Mr. James;	0	3	6	8.46	0	Mr	0
McCarthy, Mr. Timothy J; 54	0	1	7	51.9	54	Mr	0

Transpose

$$A = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad A^T = [a\ b\ c\ d]$$

4x1

1 x 4

Transpose

$$A = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \qquad A^T = [a\ b\ c\ d]$$

1 x 4

4x1

$$(A + B)^T = A^T + B^T$$

$$(AB)^T = B^T A^T$$

$$\boxed{A^T . A = \sum_i a_i^2}$$

$$(cA)^T = cA^T$$

Linear Regression Formulation

$For\ i = 1$

$$w_{1,0}x_{1,0} + w_{1,1}x_{1,1} + w_{1,2}x_{1,2} + \ldots + w_{1,n}x_{1,n} = \sum_{j=0}^{n} w_{1,j}x_{1,j}$$

Our weights or tuning parameters

$$w = \Theta = \beta$$

We can **adjust** our $\boldsymbol{w_j}$ values to **approximate** $\boldsymbol{y_j}$ using $\boldsymbol{\hat{y}_j}$

Goal: In $\boldsymbol{j}$, Find some $\boldsymbol{w_j}$ for $\boldsymbol{f(w_j,x_j)}$ | $\boldsymbol{f(w_j,x_j)} \rightarrow \boldsymbol{y_j}$ For all $\boldsymbol{n}$

Find an approximate $\boldsymbol{y}$
We will call $\boldsymbol{\hat{y}}$.
Model maps $\boldsymbol{f(X) = \hat{y}} \rightarrow \boldsymbol{y}$
For all seen $\boldsymbol{X}$ and unseen $\boldsymbol{X}$

Linear Formulation

$$A^T . A = \sum_i a_i^2$$

$$w_{1,0}x_{1,0} + w_{1,1}x_{1,1} + w_{1,2}x_{1,2} + \dots + w_{1,n}x_{1,n} = \sum_{j=0}^{n} w_{1,j}x_{1,j}$$

$$For\ i = 1, \sum_{j=0}^{n} w_{1,j}x_{1,j} = w_1^T x_1 = f(w_1 x_1)$$

$$W = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_n \end{bmatrix}$$

$$w_i^T x_i = f(w_i x_i)$$

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X)* = *ŷ*** → ***y***
For all seen ***X*** and unseen ***X***

Residual Error & Estimating *y*

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X)* = *ŷ*** → ***y***
For all seen ***X*** and unseen ***X***

We can **adjust** our ***w*** values to **approximate *y*** using ***ŷ***

Goal: Find some ***w*** for ***f(w,x)* | *f(w,x)*** → ***y***

$$\sum_{j=0}^{n} w_{1,j} x_{1,j}$$

Residual Error & Estimating *y*

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X)* = *ŷ*** → ***y***
For all seen ***X*** and $\boxed{\text{unseen } \boldsymbol{X}}$

We can **adjust** our ***w*** values to **approximate *y*** using ***ŷ***

Goal: Find some ***w*** for ***f(w,x)* | *f(w,x)*** → ***y***

$$\sum_{j=0}^{n} w_{1,j} x_{1,j} + \epsilon_i = w_{1,0} x_{1,0} + w_{1,1} x_{1,1} + \ldots + w_{1,n} x_{1,n} + \epsilon_i$$

$$y_i = \hat{y}_i + \epsilon_i \qquad \hat{y}_i = w_i^T x_i = f(w_i x_i)$$

Residual Error & Estimating *y*

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X) ŷ*** → ***y***
For all seen ***X*** and $\boxed{\text{unseen } \boldsymbol{X}}$

$$A^T . A = \sum_i a_i^2$$

$$y_i = \hat{y}_i + \epsilon_i$$

$$\epsilon_i = y_i - \hat{y}_i \ \ since\ \hat{y}_i = w^T x_i$$

Residual Sum of Squares

$$RSS = \epsilon^T \epsilon$$

Residual Error & Estimating *y*

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X)* = *ŷ*** → ***y***
For all seen ***X*** and unseen ***X***

$$A^T . A = \sum_i a_i^2$$

$$y_i = \hat{y}_i + \epsilon_i$$

$$\epsilon_i = y_i - \hat{y}_i \ \ since \ \ \hat{y}_i = w^T x_i$$

Residual Sum of Squares

$$RSS = \epsilon^T \epsilon = \sum_{i=1}^{m} (\epsilon_i)^2 = \sum_{i=1}^{m} (y_i - w\ x_i)^2$$

Observations - Predictions

Minimize on RSS

Find an approximate ***y***
We will call ***ŷ***.
Model maps ***f(X)*** **=** ***ŷ*** → ***y***
For all seen ***X*** and $\boxed{\text{unseen } \boldsymbol{X}}$

Residual Sum of Squares

$$RSS = \epsilon^T \epsilon = \sum_{i=1}^{m} (\epsilon_i)^2 = \sum_{i=1}^{m} (y_i - w\, x_i)^2$$

Observations - Predictions

$$RSS_t := min(RSS_{t-1}),\ where\ w\ is\ changed\ at\ each\ time\ step\ t$$

How to minimize the RSS?

1. Ordinary Least Squares (OLS) : Method 1 – Analytical approach
2. Gradient Descent (GD) : Method 2 – Numerical approach