

# ECS 171: Machine Learning

Summer 2023

Edwin Solares

[easolares@ucdavis.edu](mailto:easolares@ucdavis.edu)

Linear Regression

# What is Machine Learning: Recap

@Ilias Tagkopoulos

## Recap

Step 1. Get enough data!



Dataset

Step 2. Do all of the data samples have **labels**?

$$\begin{bmatrix} x_{11} & \cdots & x_{1m} & y_1 \\ \vdots & & \vdots & \vdots \\ x_{n1} & \cdots & x_{nm} & y_n \end{bmatrix}$$

Yes

No

$$\begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$$

Step 3: The task is to predict a continuous variable, assign a new sample to a class, or perform an optimal action?

**Supervised Learning**

**Reinforcement Learning**

**Unsupervised Learning**

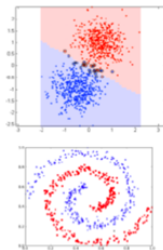
Assign to a class

Predict a continuous variable

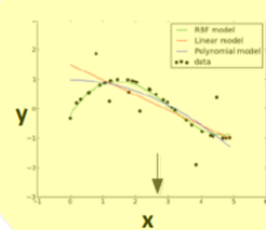
Perform an optimal action

- Bayesian Classification (Naïve Bayes)
- Linear Discriminant Analysis
- Artificial Neural Networks
- Decision Trees
- Support Vector Machines

**CLASSIFICATION**



**REGRESSION**



Linear, polynomial, logistic, ...

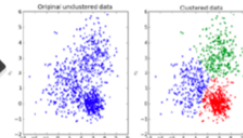
**REINFORCEMENT LEARNING (\*)**



Markov Decision Process (MDP), POMDP, Q-learning, ...

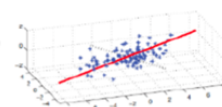
Step 3: The task is to cluster data together, find latent factors or complete missing data?

**Clustering**



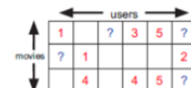
- K-means
- Hierarchical clustering
- SOM

**Dimensionality Reduction**



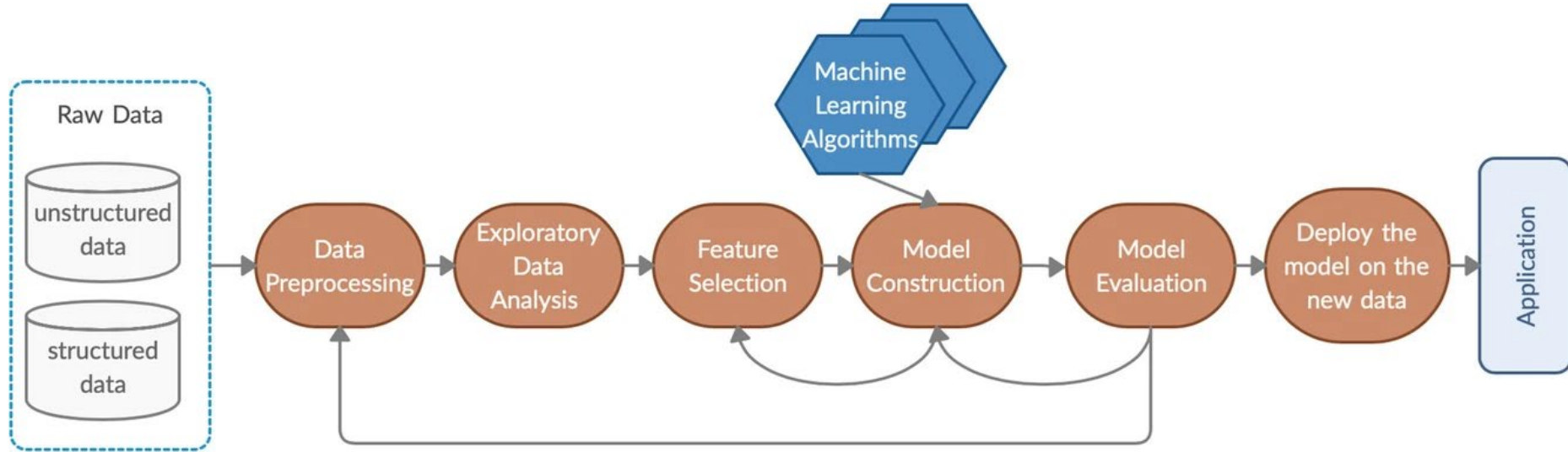
- PCA
- ICA

**Missing Data**



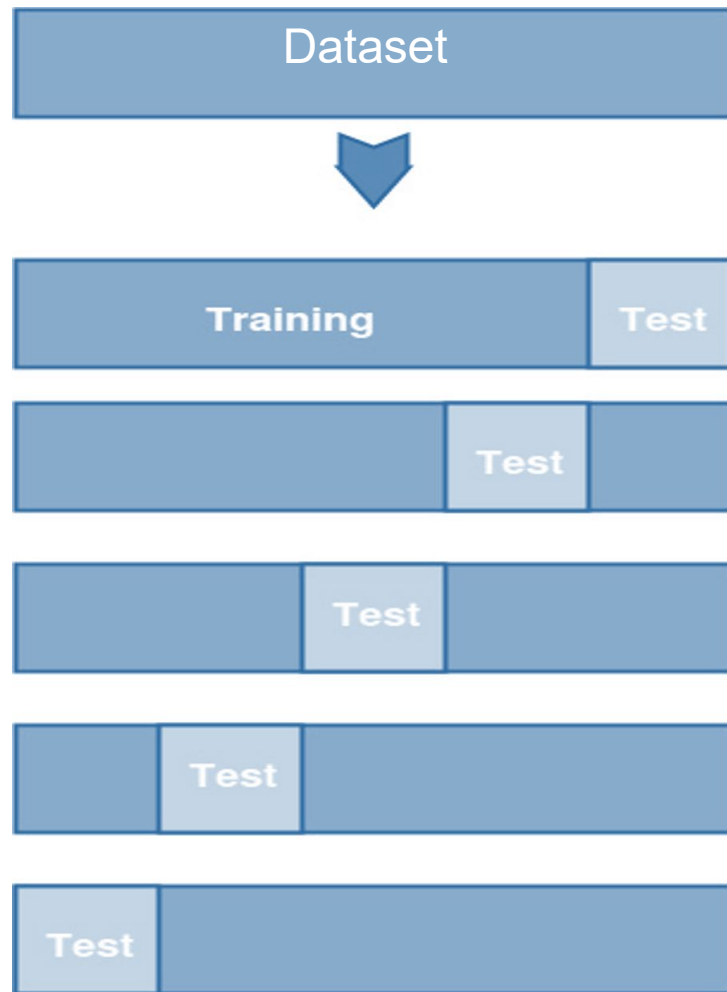
- Collaborative filtering
- Market Basket analysis

# 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 ( $\mathbf{X}$ ) includes
  - advertising budget (dollar value)
- Output  $\mathbf{y}$  i.e., the class attribute
  - sales in thousands of units

Find an approximate  $\mathbf{y}$   
We will call  $\hat{\mathbf{y}}$ .

Model maps  $\mathbf{f}(\mathbf{X}) = \hat{\mathbf{y}} \rightarrow \mathbf{y}$

For all seen  $\mathbf{X}$  and **unseen  $\mathbf{X}$**

Linear regression: find a linear relationship between  $\mathbf{X}$  (input) and  $\mathbf{y}$  (output).

Goal: find  $\mathbf{f}(\mathbf{X}) = \hat{\mathbf{y}} \rightarrow \mathbf{y}$

Advertisement budget  
(**independent variable**)  $\mathbf{X}$

Output sales  
(**dependent variable**)  $\mathbf{y}$

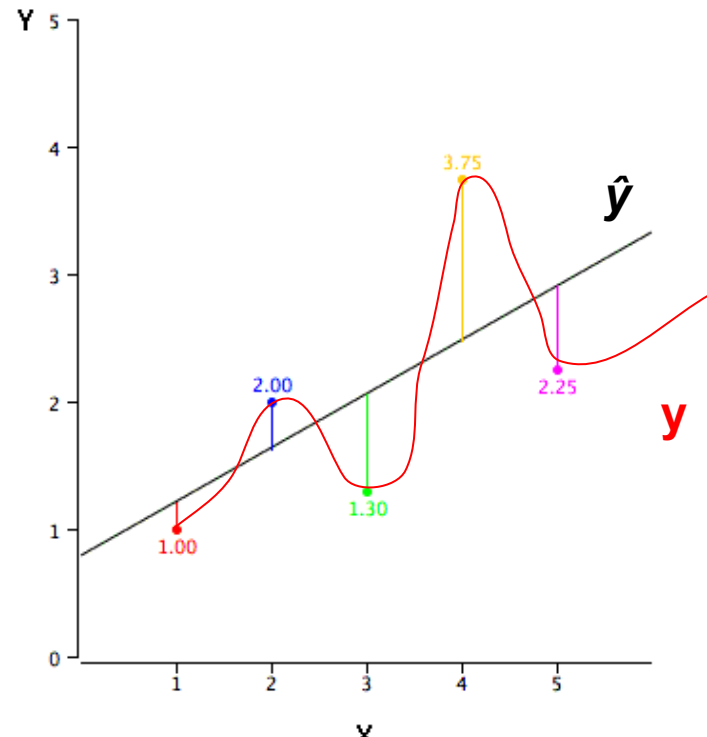
# Linear Regression Model

Supervised learning

Popular statistical learning method

Predicts a quantitative response  $\mathbf{y}$  from predictive attribute  $\mathbf{X}$

Linear relationship between  $\mathbf{X}$  and  $\mathbf{y}$



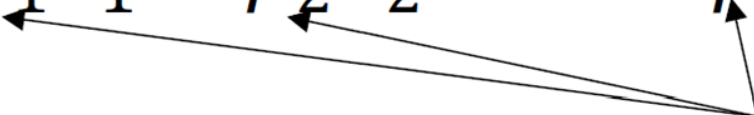
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$



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**:

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

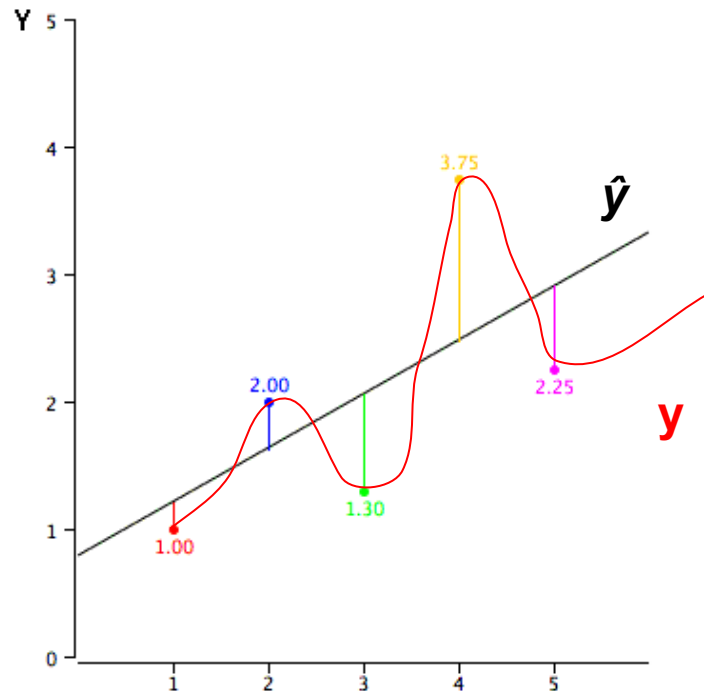
Observed value

Predicted value

Minimize the  
Loss!

Method 1: Ordinary Least Squares (OLS)

Method 2: Gradient Descent (GD)



# Tabular Data $\rightarrow$ Matrix $\rightarrow$ Formula

$m$ -by- $n$  matrix

$a_{i,j}$

$n$  columns

$j$  changes

$m$   
rows

$i$   
changes

$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$\cdot$	$\cdot$	$\cdot$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$\cdot$	$\cdot$	$\cdot$
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

For  $m$  rows,  $i$ th row in  $m$  rows

For  $n$  columns,  $j$ th column in  $n$  columns

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



# Linear Regression: Formulation

Given a dataset/matrix **M** with *m* observations:

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

Note: matrix **M** is of size *m* x *n* (rows x cols)

*For i = 1*

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

Weights

$$w = \Theta = \beta$$

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

# 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

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

$4 \times 1$   $1 \times 4$

# Transpose

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

$4 \times 1$   $1 \times 4$

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

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

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

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

# Linear Regression Formulation

For  $i = 1$

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

Our weights or tuning parameters

$$w = \Theta = \beta$$

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

**Goal:** In  $j$ , Find some  $w_j$  for  $f(w_j, x_j) \mid f(w_j, x_j) \rightarrow y_j$  For all  $n$

Find an approximate  $y$

We will call  $\hat{y}$ .

Model maps  $f(X) = \hat{y} \rightarrow y$

For all seen  $X$  and **unseen  $X$**

# Linear Formulation

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

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

$$\text{For } i = 1, \sum_{j=0}^n w_{1,j}x_{1,j} = w_1^T x_1 = f(w_1 x_1)$$
$$W = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_n \end{bmatrix}$$

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

Find an approximate  $y$

We will call  $\hat{y}$ .

Model maps  $f(X) = \hat{y} \rightarrow y$

For all seen  $X$  and unseen  $X$

# Residual Error & Estimating $y$

Find an approximate  $y$

We will call  $\hat{y}$ .

Model maps  $f(X) = \hat{y} \rightarrow y$

For all seen  $X$  and **unseen  $X$**

We can **adjust** our  $w$  values to **approximate**  $y$  using  $\hat{y}$

**Goal:** Find some  $w$  for  $f(w, x) \mid f(w, x) \rightarrow y$

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



## Residual Error & Estimating $y$

Find an approximate  $y$

We will call  $\hat{y}$ .

Model maps  $f(X) = \hat{y} \rightarrow y$

For all seen  $X$  and **unseen  $X$**

We can **adjust** our  $w$  values to **approximate**  $y$  using  $\hat{y}$

**Goal:** Find some  $w$  for  $f(w, x) \mid f(w, x) \rightarrow y$

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

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

$$\hat{y}_i = w_i^T x_i = f(w_i x_i)$$



## Residual Error & Estimating $\mathbf{y}$

Find an approximate  $\mathbf{y}$

We will call  $\hat{\mathbf{y}}$ .

Model maps  $\mathbf{f}(\mathbf{X}) \hat{\mathbf{y}} \rightarrow \mathbf{y}$

For all seen  $\mathbf{X}$  and **unseen  $\mathbf{X}$**

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

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

$$\epsilon_i = y_i - \hat{y}_i \text{ since } \hat{y}_i = \mathbf{w}^T \mathbf{x}_i$$

Residual Sum of Squares

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

## Residual Error & Estimating $y$

Find an approximate  $y$

We will call  $\hat{y}$ .

Model maps  $f(\mathbf{X}) = \hat{y} \rightarrow y$

For all seen  $\mathbf{X}$  and **unseen  $\mathbf{X}$**

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

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

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

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  $\mathbf{y}$

We will call  $\hat{\mathbf{y}}$ .

Model maps  $\mathbf{f}(\mathbf{X}) = \hat{\mathbf{y}} \rightarrow \mathbf{y}$

For all seen  $\mathbf{X}$  and **unseen  $\mathbf{X}$**

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