

Machine Learning

Prepared by Dr. Yasser Abdelhamid

Lecture 4





OUTLINE

- **Multivariate regression**
- **Data wrangling**
- **Feature scaling**
- **Normalization**
 - Standardization (Z-Score Normalization)
 - Mean normalization
 - Min Max Normalization
- **Dealing with non-numerical data**
 - One-Hot encoding
- **Multicollinearity**
- **Training and testing**
 - K-Fold cross validation
- **Ploynomial regression**

MULTIVARIATE REGRESSION

- ❖ Normally, the target variable may depend on more than one feature.
- ❖ example: House price may depend on number of rooms, age, floor, in addition to house area.

LINEAR REGRESSION WITH SINGLE VARIABLE

Size (feet ²)	Price (\$1000)
x	y
2104	460
1416	232
1534	315
852	178
...	...

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

MULTIVARIATE REGRESSION

Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

NOTATION

Notation:

- n = number of features
- $x^{(i)}$ = input (features) of i^{th} training example.
- $x_j^{(i)}$ = value of feature j in i^{th} training example.

HYPOTHESIS FUNCTION

❖ **Single variable/feature :**

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

❖ **Multiple variables/features:**

For n features

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$$

or more comfortably

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$$

where x_0 is always 1

HYPOTHESIS FUNCTION

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$$

θ and x are two multiplied vectors.

$$h_{\theta}(x) = \theta^T x$$

GRADIENT DESCENT FOR LINEAR REGRESSION WITH MULTIPLE FEATURES

Hypothesis: $h_{\theta}(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \dots, \theta_n$

Cost function:

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \dots, \theta_n)$$

}

(simultaneously update for every $j = 0, \dots, n$)

GRADIENT DESCENT WITH $N = 1$

Repeat {

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)}$$

(simultaneously update θ_0, θ_1)

}

GRADIENT DESCENT WITH $N \geq 1$

New algorithm ($n \geq 1$):

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for
 $j = 0, \dots, n$)

}

EXAMPLE FOR TWO FEATURES

New algorithm ($n \geq 1$):

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for
 $j = 0, \dots, n$)

}

$$\begin{aligned} \theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)} \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_1^{(i)} \\ \theta_2 &:= \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_2^{(i)} \end{aligned}$$

DATA WRANGLING

- ❖ Data wrangling is the process of **cleaning, structuring and enriching raw data** into a desired format for better decision making in less time.

GRADIENT DESCENT CONVERGENCE

Gradient descent need data to be within near/ same **scale** to reach **convergence faster**.

FEATURE SCALING

- ❖ Machine learning algorithms deal with features as **numbers** only.
- ❖ We have to work on all features with the **same scale**.
- ❖ It is **better to deal with small values** rather than large values.

FEATURE SCALING

❖ Sometimes the scale of data representing **features vary in scale**, for example:

X1	3	5	6	4	2
X2	25000	38000	36000	28000	24000

FEATURE SCALING

- ❖ All features are represented in **magnitude** and **unit** which are different :
 - age is represented in **magnitude(integer)** and **unit(years)**
 - weight is represented in **magnitude(real number)** and **unit(Kg)**

NORMALIZATION

Normalization is the process of **scaling down** feature values to be **between 0 and 1**.

STANDARDIZATION (Z-SCORE NORMALIZATION)

Standardization is the process of scaling feature data **based on the standard normal distribution**, where the **mean is 0** and **standard deviation is 1**.

Z-SCORE NORMALIZATION

❖ Scaling data according to the standard normal distribution:

$$Z = \frac{X - \mu}{\sigma}$$

❖ Where:

- x is the current attribute value
- μ is the **mean** of the population, and
- σ is the **standard deviation**

EXAMPLE

i	Area	Bedrooms	z-Score(Area)	z-Score(Bedrooms)
1	900	2	-0.563734521	-0.866025404
2	1300	3	1.240215946	0.866025404
3	1100	3	0.338240713	0.866025404
4	800	2	-1.014722138	-0.866025404

Mean	1025	2.5
St.Dev.	221.7355783	0.577350269

MIN MAX NORMALIZATION

❖ Scaling values of features to be within 0 and 1.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

EXAMPLE

i	Area	Bedrooms	Min-Max Norm(Area)	Min-Max Norm(Bedrooms)
1	900	2	0.2	0
2	1300	3	1	1
3	1100	3	0.6	1
4	800	2	0	0

Min	800	2
Max	1300	3

DEALING WITH NON-NUMERICAL DATA

- ❖ Machine learning algorithms deal with **numeric data**.
- ❖ Sometimes dataset have **text features**.
- ❖ This data has to be **encoded** in a form that is suitable for ML algorithms.

NON-NUMERICAL CATEGORICAL VARIABLES

Ordinal variables:

- **Has implicit numeric order.**
 - excellent, very good, good, pass
 - first floor, second floor,...

NON-NUMERICAL CATEGORICAL VARIABLES

Nominal variables:

- **No numeric ordering.**
 - male, female
 - cairo, giza, ...
 - red, green, blue

NON-NUMERICAL CATEGORICAL VARIABLES SIMPLE ENCODING

- ❖ Simple encoding is to give a numerical code for each categorical value.
- ❖ Simple numerical encoding (giving a numerical value for each label) may not work for nominal variables.

ONE HOT ENCODING

- ❖ Create a separate column for each of the distinct labels of the nominal variable.
- ❖ Put one if the field value matches the column label,
- ❖ Put 0 otherwise.

EXAMPLE

Gender	Height	Weight	Dummy variables	
			Male	Female
Male	170	77	1	0
Female	156	60	0	1
Female	170	88	0	1

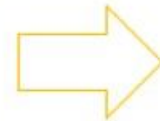
MULTICOLLINEARITY

- Multicollinearity occurs when independent variables in a regression model are correlated.
- This correlation is a problem because independent variables should be independent of each other.
- If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results.

MULTICOLLINEARITY

- To avoid Multicollinearity problem, we **remove one of these variables.**

Height	Weight	Male	Female
170	77	1	0
156	60	0	1
170	88	0	1



Height	Weight	Male
170	77	1
156	60	0
170	88	0

TRAINING AND TESTING DATASETS

- ❖ In order to make sure that the model that we build is generalizing well (i.e. is able to predict results for unseen instances) We **split dataset into two sets**.
- ❖ The first is for **training** and the second is for **testing**.

TRAINING AND TESTING DATASETS

- ❖ Usually we take:
 - **80% for training** and
 - **20% for testing.**
- ❖ This method is usually used when there is a **sufficient amount of data.**

K-FOLD CROSS VALIDATION

❖ Divide dataset into k subsets.

$$\{d_1, d_2, \dots, d_k\}$$

For d in $\{d_1, d_2, \dots, d_k\}$

set d as a test set.

use the rest as training set.

Calculate error

K-FOLD CROSS VALIDATION

- ❖ Calculate the overall error as the average of all subsets.
- ❖ K-Fold cross validation method is a good choice when data is not sufficient.

POLYNOMIAL FUNCTION

❖ A polynomial function is a mathematical function that consists of one or more terms, each of which is a **constant multiplied by a variable raised to a non-negative integer power**.

❖ The general form of a polynomial function is:

$$f(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_kx^k$$

❖ Here, $f(x)$ represents the polynomial function, x is the variable, $a_0, a_1, \dots, a_k$ are the coefficients of the polynomial, and k is the highest power or degree of the polynomial.

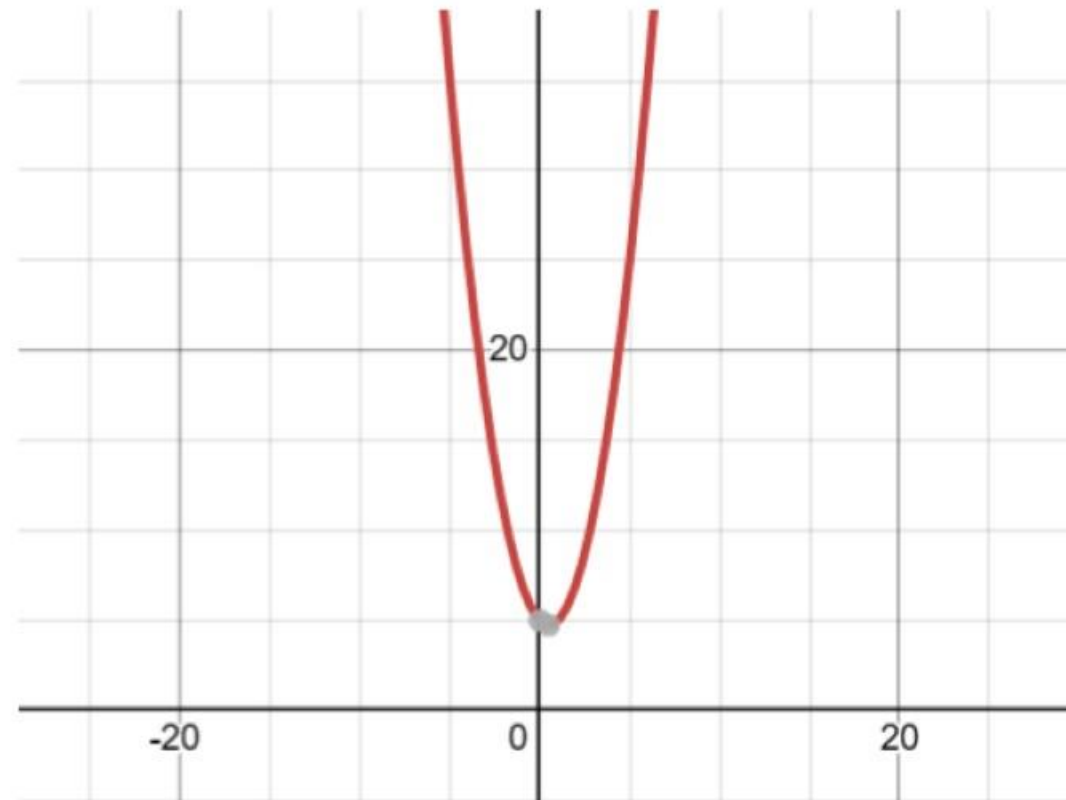
POLYNOMIAL FUNCTION EXAMPLES

Polynomial Functions

Degree	Name	Example
0	Constant	5
1	Linear	$2x + 3$
2	Quadratic	$x^2 - x + 5$
3	Cubic	$3x^3 + 23$
4	Quartic	$x^4 + 2x^3 - 3$
5	Quintic	$x^5 + 2x^3 - .5x + 3$

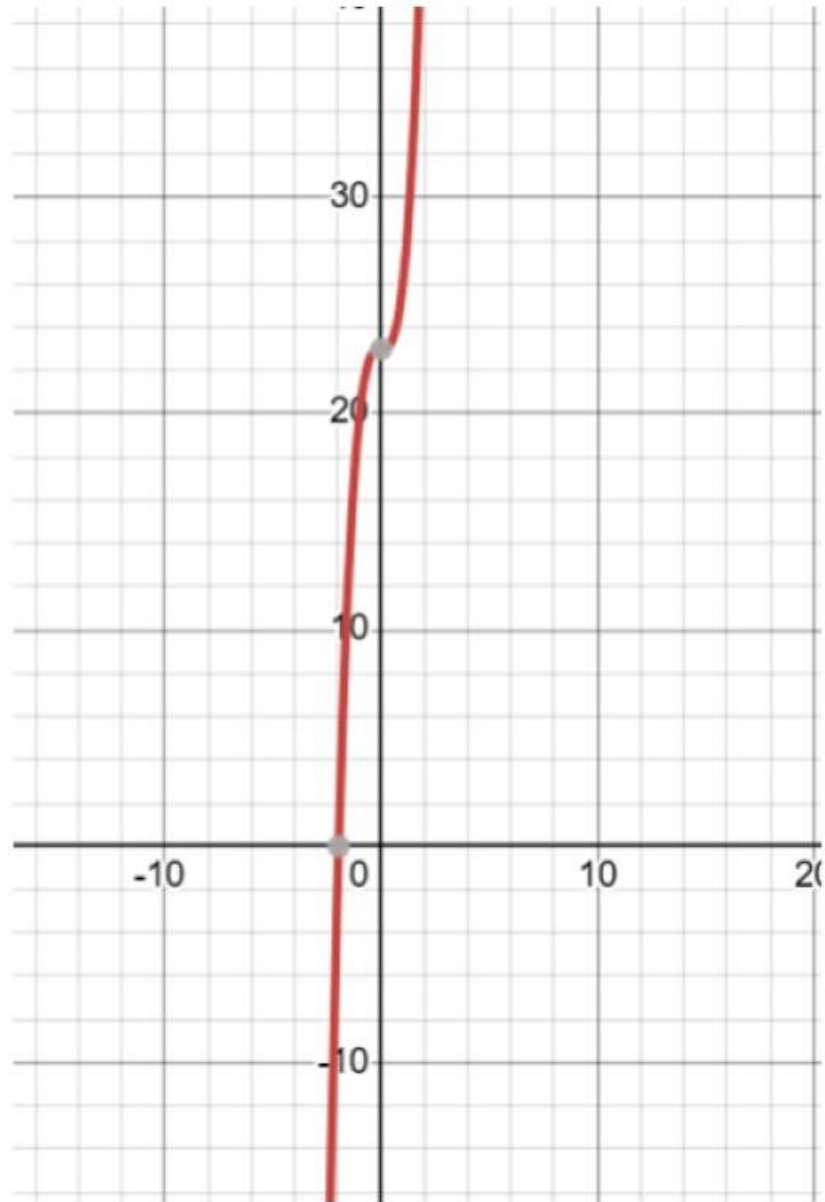
QUADRATIC

$$x^2 - x + 5$$



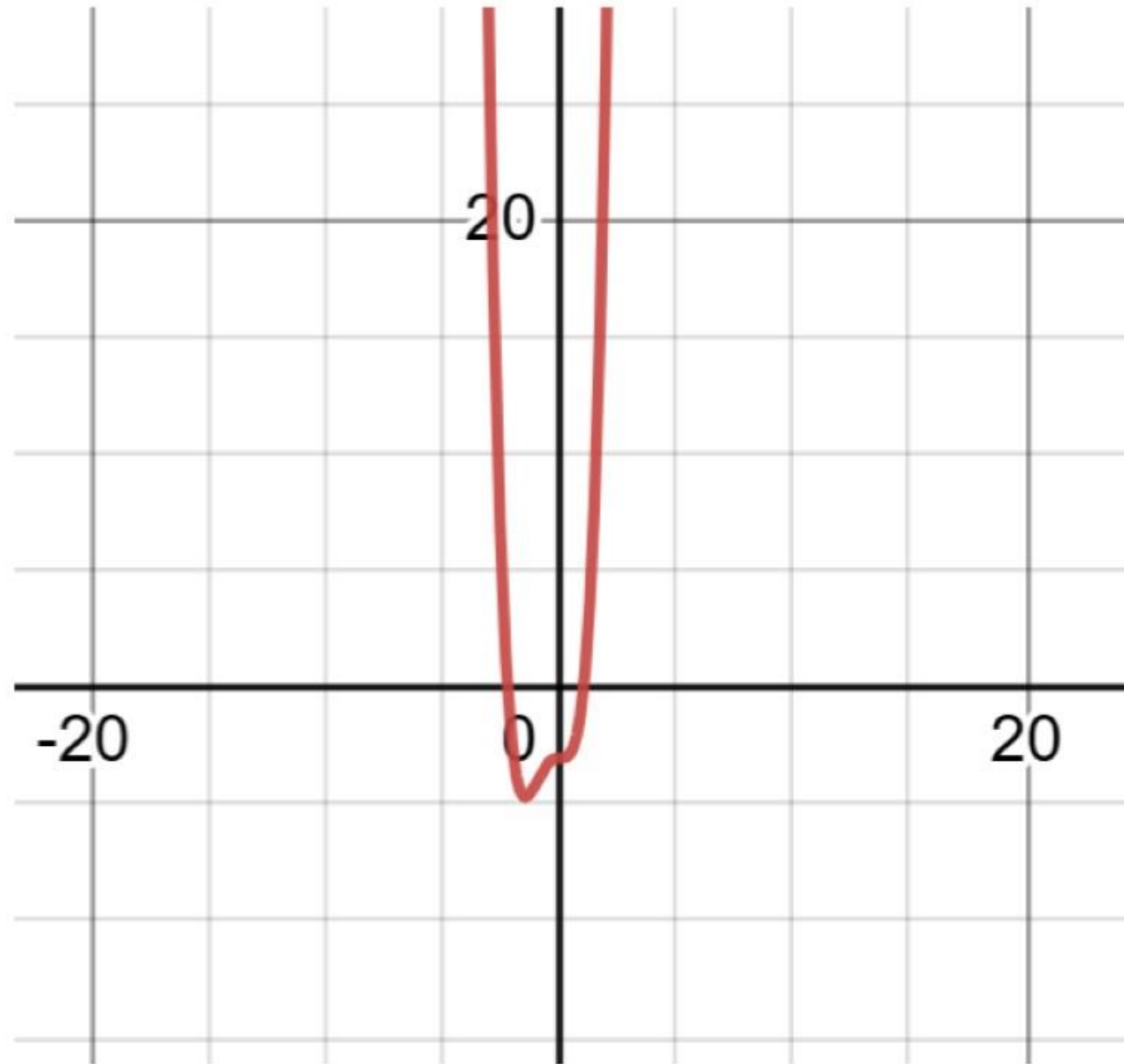
CUBIC

$$3x^3 + 23$$



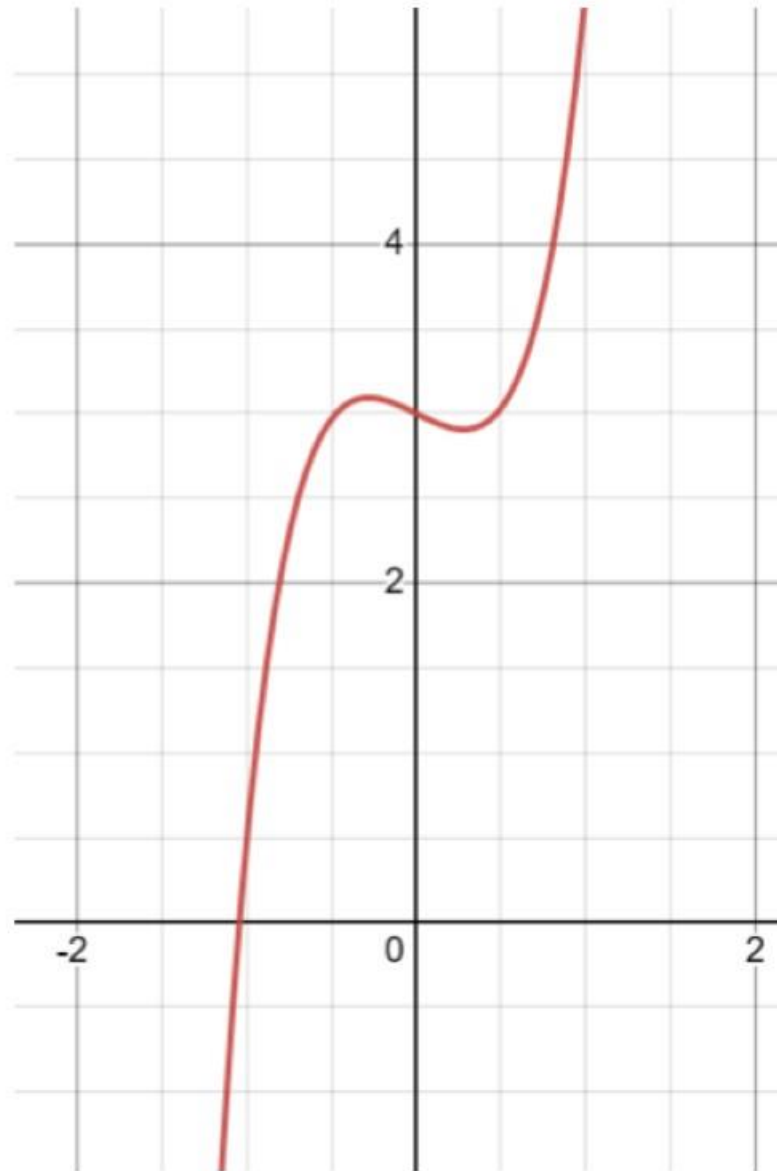
QUARTIC

$$x^4 + 2x^3 - 3$$



QUINTIC

$$x^5 + 2x^3 - 0.5x + 3$$



POLYNOMIAL REGRESSION

- ❖ Non-linear relationships can be modeled using polynomial functions.
- ❖ We can model relationships using k^{th} degree polynomials in the following form:

$$h_{\theta}(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \dots + \theta_k x^k$$

EXAMPLE:

$$h_{\theta}(\text{age}) = \theta_0 + \theta_1 \text{age} + \theta_2 (\text{age})^2 + \theta_3 (\text{age})^3 + \dots$$

By replacing age , age^2 , age^3 ... with x_1, x_2, x_3 ...

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots$$

which is similar to linear regression.



SUMMARY(1)

- ◉ **Multivariate regression**
- ◉ **Data wrangling**
- ◉ **Feature scaling**
- ◉ **Normalization**
 - ◉ **Standardization (Z-Score Normalization)**
 - ◉ **Mean normalization**
 - ◉ **Min Max Normalization**



SUMMARY(2)

- **Dealing with non-numerical data**
 - One-Hot encoding
- **Multicollinearity**
- **Training and testing**
 - K-Fold cross validation
- **Polynomial regression**

