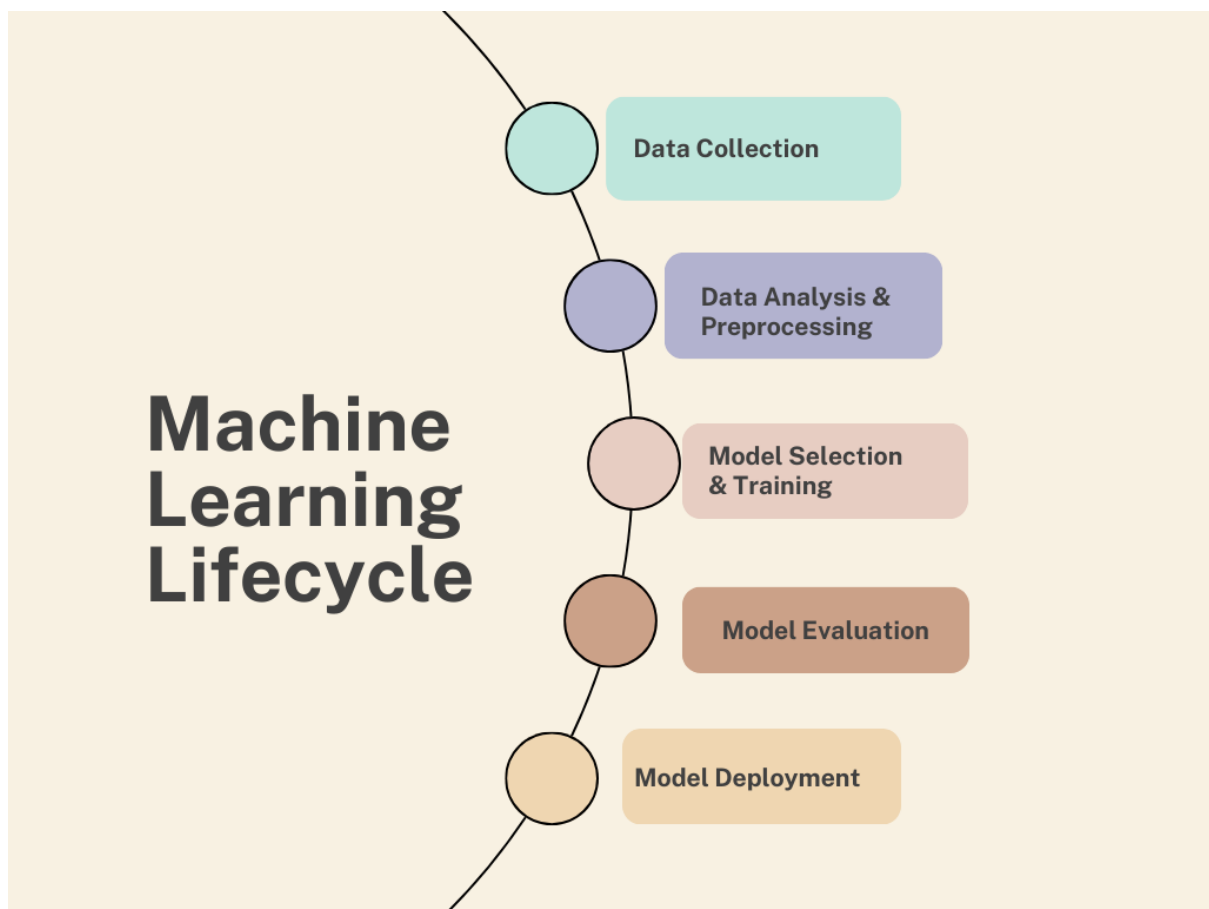# Data Visualization

## Machine Learning Lifecyle:



**Machine Learning Lifecyle** typically consists of four main stages: data collection, data analysis & preprocessing, model selection & training, and model deployment.

**Data Collection:**

- The first step in any machine learning project is to gather relevant data. This data can come from various sources, such as websites, APIs, sensors … In our case we get already gathered data provided from Kaggle for example.

**Data Analysis & Preprocessing:**

- Once you have the data, the next step is to analyze and preprocess it. Data analysis involves exploring the dataset to understand its structure, characteristics, and patterns.

- Data preprocessing is the process of cleaning, transforming, and preparing the data for model training. This may involve handling missing values, encoding categorical variables, removing outliers and splitting the data into training and test sets.

**Model Selection & Training: (we will discover this in the next sessions)**

- In this stage, you select an appropriate machine learning algorithm based on the problem you're trying to solve and the characteristics of your data.

- After splitting your data into training and test sets and use the training data to train the model. The model learns to make predictions or classifications based on the patterns in the training data.

**Model Deployment:**

- Once you have a trained machine learning model that performs well, the final step is to deploy it into a production environment. Model deployment can take different forms, depending on the application, such as integrating the model into a web application, mobile app, or IoT device.

# Machine Learning Types:
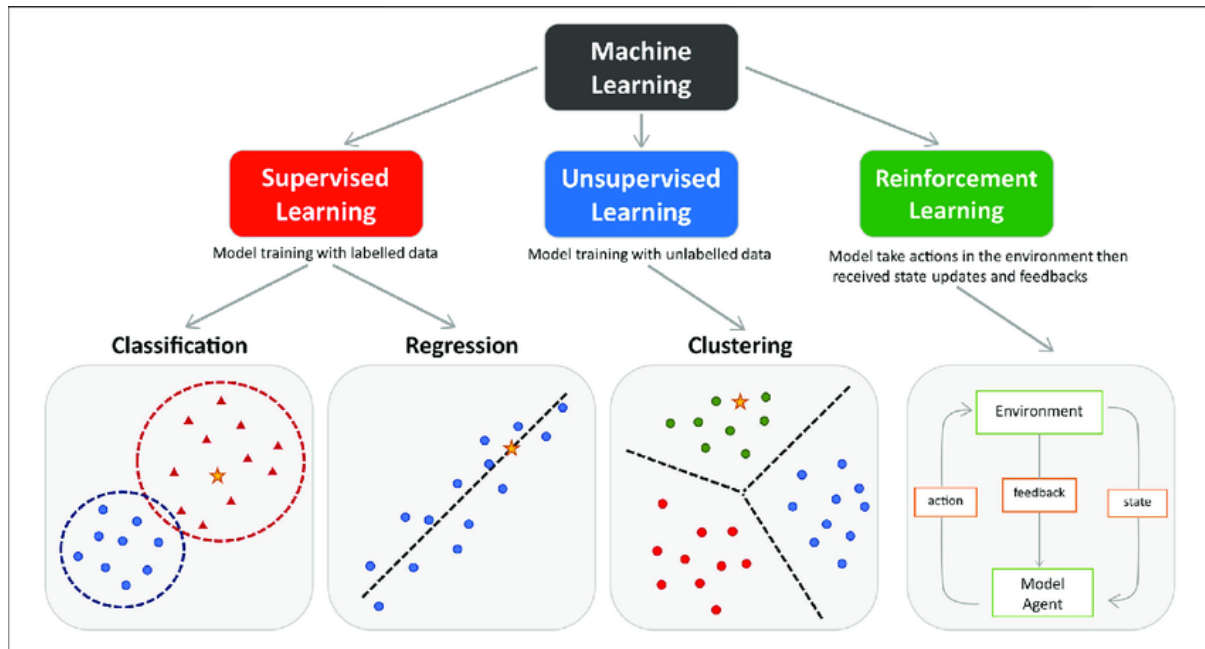
# Supervised Vs Unsupervised Learning, Explained

## Supervised

| $X_1$ | $X_2$ | $X_p$ | Y |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

Target

## Un-Supervised

| $X_1$ | $X_2$ | $X_p$ | Y |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

No Target

## Regression Data

| $X_1$ | $X_2$ | $X_3$ | $X_p$ | Y |
|---|---|---|---|---|
| | | | | 5.2 |
| | | | | 1.3 |
| | | | | 23.0 |
| | | | | 7.4 |

Numeric Target

## Classification Data

| $X_1$ | $X_2$ | $X_3$ | $X_p$ | Y |
|---|---|---|---|---|
| | | | | cat |
| | | | | dog |
| | | | | cat |
| | | | | cat |

Categorical "Labels"

# Tabular Data:
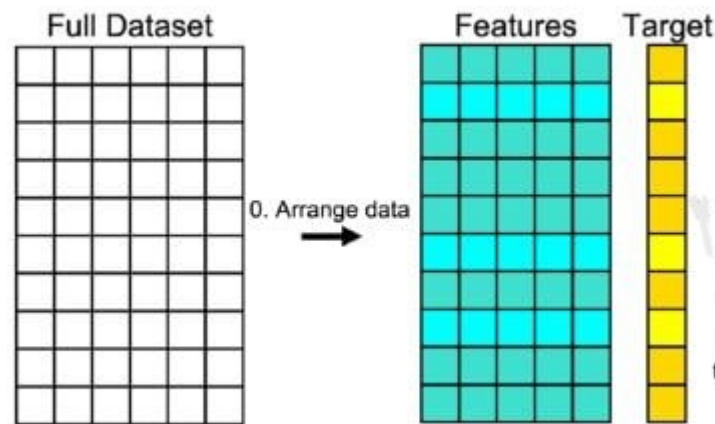


For Machine Learning we commonly use tabular and structured data.

This how the  tabular data going to look like:

- The dataset is separated into features and a target .

- The features are the characteristics or attributes of the data that the model uses to make predictions about the target variable.

This was for the supervised learning. For the unsupervised learning the data doesn't have a target.

**The cycle of the data:**
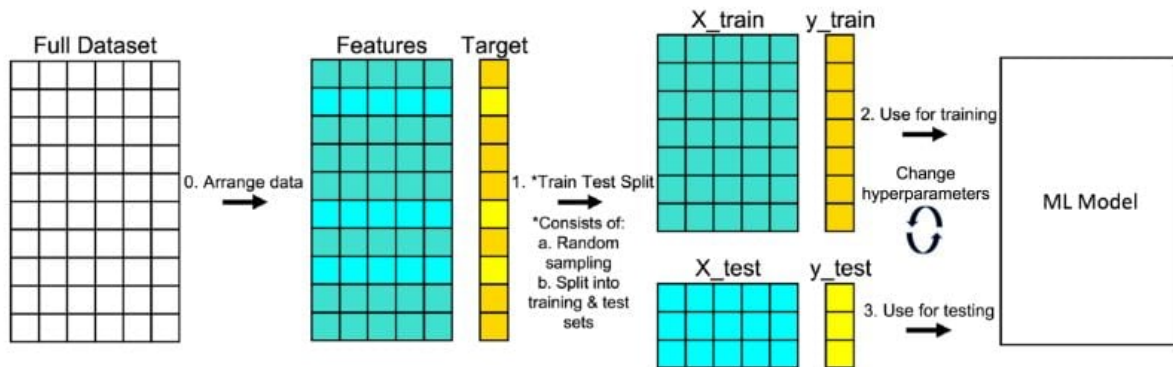


After extracting the target and the features, we going to extract the training and testing data (data split).

# Missing Values:

- How do we get missing values: **Missing values can occur for various reasons, including data entry errors, incomplete data collection.**

- How to see missing values with code: **These values are usually denoted as "NaN" (Not-a-Number) in numerical datasets or as "null" in dataset.**

- Impact on Machine Learning Models:

1. **It may lead to incorrect prediction.**

2. **Reduced Model Performance.**

3. **Many ML models cannot handle missing values, so they will return errors.**

```
df
```

|   | column_a | column_b | column_c | column_d |
|---|---------|---------|---------|---------|
| 0 | 1.0 | 1.2 | a | True |
| 1 | 2.0 | 1.4 | ? | True |
| 2 | 4.0 | NaN | c | NaN |
| 3 | 4.0 | 6.2 | d | None |
| 4 | NaN | NaN | -- | False |
| 5 | NaN | 1.1 | NaN | True |
| 6 | 6.0 | 4.3 | d | False |

**Missing values imputation:** Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located.

# Outliers:

- An outlier is a data point in a dataset that significantly differs from the majority of the other data points.

- Outliers can be caused by various factors, including errors in data collection, natural variation, or extreme events.
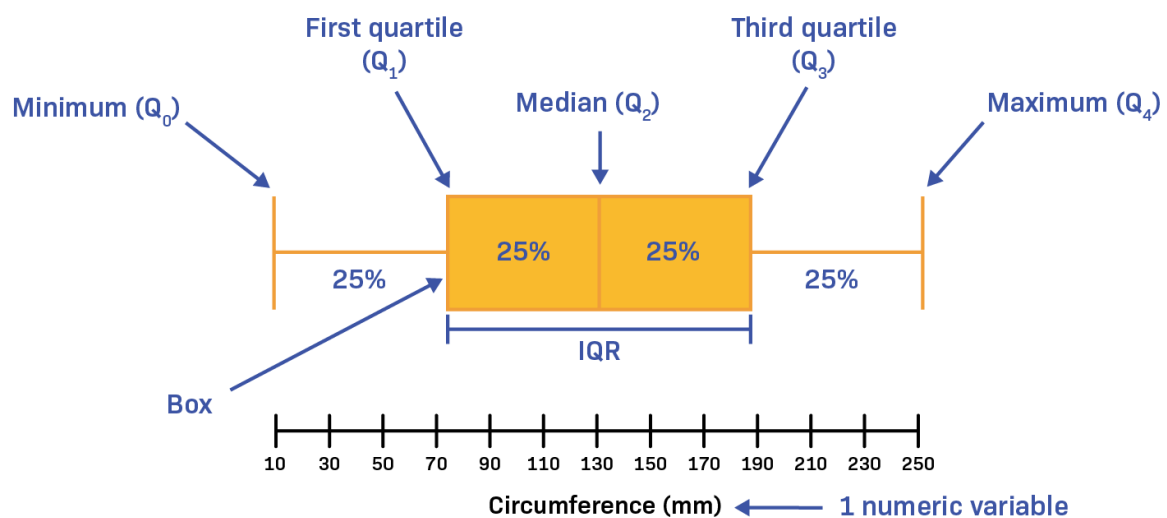
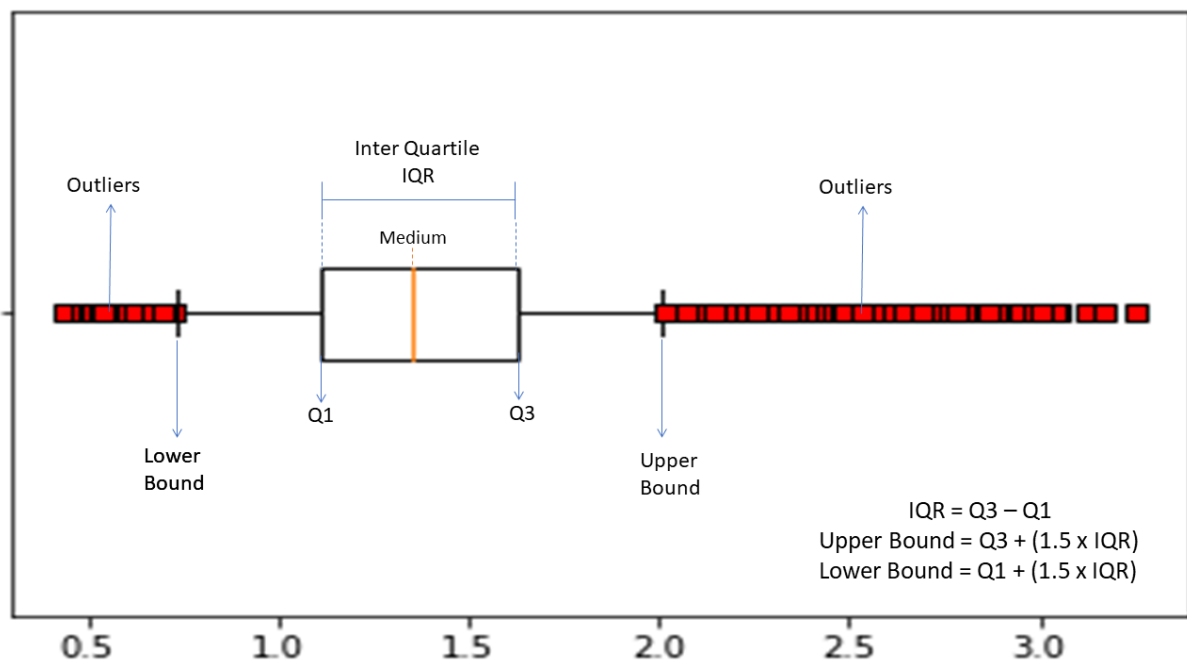Example:
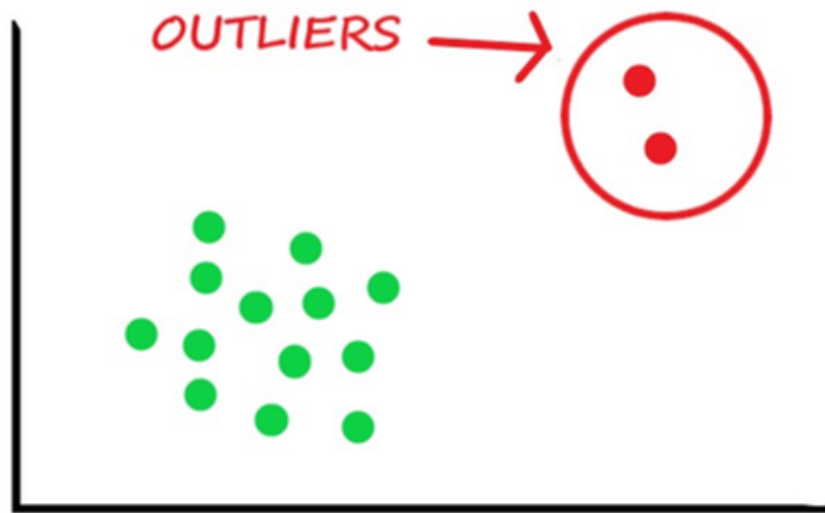
# Boxplots:

- Boxplots and identifying outliers:

Boxplot:

1. First quartile (***Q1):  25% of the data values are lower than this point.***

2. Median (***Q2): the middle value of the dataset, so that 50% of the data values are lower than this point.***

3. Third quartile (***Q3): 75% of the data values are lower than this point.***

4. IQR: Interquartile range

Boxplot and outliers visualization:



# Correlation Explanation:

## Correlation Formula:

**ρ (X, Y) = cov(X, Y) / (σ(X) * σ(Y))**

- **ρ (X, Y):** This is the Pearson correlation coefficient between the variables X and Y. It quantifies the strength and direction of the linear relationship between these two variables. The value of ρ can range from -1 to 1.

- **cov(X, Y):** The covariance between X and Y. Covariance is a measure of how two variables change together. A positive covariance indicates that when one variable increases, the other tends to increase as well, and vice versa for negative covariance. However, the magnitude of covariance is not standardized, so it may be challenging to compare across different datasets.

- **σ(X) and σ(Y):** The standard deviations of the variables X and Y, respectively. The standard deviation measures the degree of dispersion or variability in the values of a variable. A high standard deviation means that the values are spread out, while a low standard deviation indicates that the values are close to the mean.



Positive Correlation    Negative Correlation    No Correlation
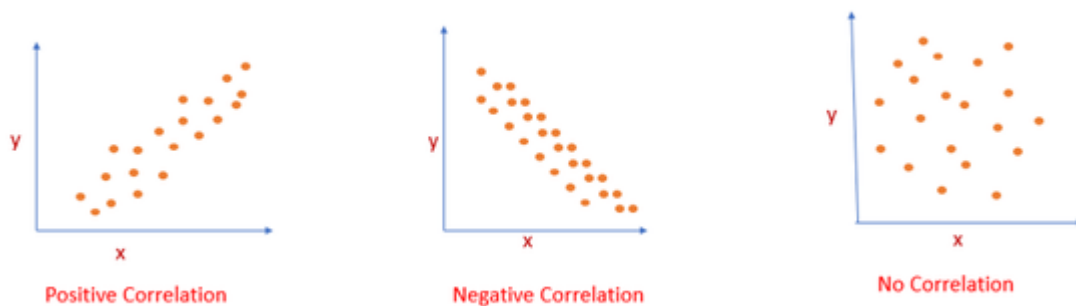
## Correlation Helps Selecting Important Features:

Analyzing the correlation between features and the target variable is a critical step in feature selection and model building in machine learning. Features that have a strong positive or negative correlation with the target variable are typically **considered more important for predicting the target** and are more likely **to be included in the model**. features with little to no correlation with the target may be less informative and can potentially be excluded from the model to simplify it.

# Normal Distribution:

Many natural phenomena, like human height, test scores, or measurement errors, follow a normal distribution. It's a common statistical concept that helps us understand and analyze data patterns in various fields, from science to finance.

## Why it's useful in Machine Learning:

**Predictive Power:** In many cases, data that follows a normal distribution is easier to work with. Machine learning models often make predictions based on patterns they find in data. Normal distributions help these models find those patterns more easily.

**Assumptions:** Some machine learning methods assume that the data they work with is normally distributed. It's like giving the model a hint about how the data behaves. For example, linear regression assumes that the relationship between variables is linear, and the errors follow a normal distribution. This assumption simplifies the math and can make the model work better.