**Machine Learning**

# **Understanding Data**

7 questions you must ask yourself.

1. How big data is?
2. How does data look like?
3. What are data types of columns?
4. Are there any missing values?
5. How does data look mathematically?
6. Are there any duplicate values?
7. How is the correlation between columns?

# **EDA**

Exploratory Data Analysis

Asking question again and again and finding their answers.

## **Univariant Analysis**

### **Categorical Variable**

1. Count Plot, how much value were there in each
2. For percentage use Pie chart

### **Numerical Variable**

1. **Histogram**, to understand the distribution of the data
2. **Displot**, tell the probability of the number, it shows curve + histogram. On y-axis probability and on x-axis values. Also called PDF (Probability Density Functions). Also shows data is skewed or not.
3. **Boxplot**, shows the outliers and quartile of the data

## **Bivariant Analysis**

1. **Scatterplot** (Numerical - Numerical), relationship between columns to understand a relationship or trend, which would be impossible to see in almost any other form.
2. **Barplot** (Numerical – Categorical)
3. **Boxplot** (Numerical – Categorical)
4. **Displot** (Numerical – Categorical) you can check the probability of for example, survived not survived with age column.
5. **HeadMap** (Categorical – Categorical)
6. **ClutsterMap** (Categorical – Categorical), shows which value is closer to each other, it create a dendrogram for relationship.
7. **Pairplot,** shows the scatter plot of each numerical column with others.
8. **Lineplot**, if on x-axis time-based data.
9. **PivotTable** draw table of two columns, on the basis of index.

## **Profiling**

Is the python library which automatically do some of above works. And return HTML interactive page.

# **Feature Engineering**

Preparing dataset for machine learning model

## **Feature Transformer**

If we think a column will not give us well performance, we transform it in some other form by changing it.

### **Missing Values**

Either you fill missing values or remove.

### **Handling Categorical Values:**

By converting the categorical values into numerical

### **Outlier Detection**

### **Feature Scaling**

Converting the feature (columns) numerical values into given range.

Is last step before giving dataset to the model.

#### Standardization

Also called z-score normalization.

Create new column with given column example, age has some value, subtract each value of age from its mean and divide it on the standard deviation. This will give us new column, new columns means will be = to zero and standard deviation will be = to 1.

In terms of mean your centering, and in terms of scaling you are scaling down the large values, or small values to scale up but to the 1.

Datapoints remains same to actual values, but data become centered and scaled.

Draw displot of actual data and scaled data you will understand.

Outliers behave same with or without standardization.

**We must do** standardization for these algorithm

* K-means (Euclidian distance)
* KNN (Euclidian distance)
* PCA (we need to control variance and do mean centering)
* ANN
* Gradient Decent

**We do not need scaling** for decision tree, random forest, XG boost, GB boost.

#### Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information.

Normalization is the process of translating data into the range [0, 1] (or any other range) or simply transforming data onto the unit sphere.

## **Feature Construction**

Creating new column with given columns.

## **Feature Selection**

Select the main feature from the dataset and gives these features to model so model can perform better.

## **Feature Extraction**

Converting given input features into completely new features. We do feature reduction as well as feature creation.