



Babeş-Bolyai University
Faculty of Mathematics and Computer Science

Curs opțional
Modele de inteligență artificială în schimbarea climatică

Exploratory Data Analysis

Data Analysis

and

Exploratory Data Analysis

Classical Data Analysis



Exploratory Data Analysis

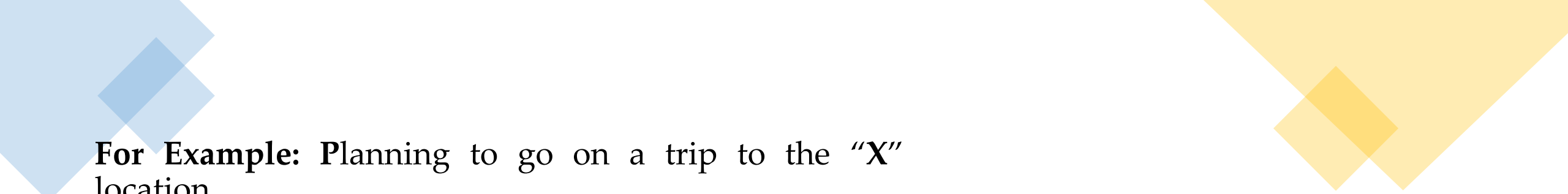


Data Analysis

- is basically using statistics and probability to figure out trends in the data set.
- it helps to sort out the “real” trends from the statistical noise.

Exploratory Data Analysis (EDA)

- is the first step in the data analysis process
- was developed by John Tukey in the 1970s.
- The exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods.
- By the name itself, we can get to know that it is a step in which we need to explore the data set.





For Example: Planning to go on a trip to the “X” location.


- Things to do before taking a decision:
 - we will explore the location on what all places, waterfalls, trekking, beaches, restaurants that location has in Google, Instagram, Facebook, and other social Websites.
 - compute whether it is in your budget or not.
 - check for the time to cover all the places.
 - type of travel method.
- Similarly, when we are trying **to build a machine learning model** we need to be pretty sure whether the **data is making sense or not**.
- The main aim of exploratory data analysis is to obtain confidence in the data such as being ready to engage a machine learning algorithm.

What Is Exploratory Data Analysis?





Need For Exploratory Data Analysis


- Exploratory Data Analysis is a crucial step before we jump to machine learning or modeling the data.
 - By doing this we can get to know whether the selected features are good enough to model, are all the features required, are there any correlations based on which extract some relevant information.
 - Once Exploratory Data Analysis is complete and insights are drawn, its feature can be used for supervised and unsupervised machine learning modeling.
- 

In addition, EDA ...

- In every machine learning workflow, the last step is Reporting or Providing the insights to the Stake Holders and a Data Scientist can explain every line and every decision.
- By completing the **Exploratory Data Analysis** we will have many plots, heat-maps, frequency distribution, graphs, correlation matrix along with the hypothesis by which any individual can understand what the data is all about and what insights we got from exploring the data set.
- There is a saying “**A picture is worth a thousand words**”.
- For data scientist the saying becomes “**A Plot is worth a thousand rows**”
- Comparing ... In the beginning **Trip Example**, we do all the exploration of the selected place based on which we will get the confidence to plan the trip and even share with our friends the insights we got regarding the place so that they can also join.



What Are The Steps In Exploratory Data Analysis In Python?

- There are different steps for performing EDA.
 - The most important steps are:
 - I. Description of data
 - II. Handling missing data
 - III. Handling outliers
 - IV: Understanding relationships and new insights through plots
- 

I. Description of data

- We need to know the different kinds of data and other statistics of our data before we can move on to the other steps.
- A good one is to start with the **describe()** function in python.
- In Pandas, we can apply describe() on a DataFrame which helps in generating descriptive statistics that summarize the central tendency, dispersion, and shape of a dataset's distribution, excluding NaN values.
- The result's index will include count, mean, std, min, max as well as lower, 50 and upper percentiles.
- By default, the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.

```
import pandas as pd
# using a standard dataset
from sklearn.datasets import load_boston

boston = load_boston()
x = boston.data
y = boston.target
columns = boston.feature_names

# creating dataframes
df = pd.DataFrame(boston.data)
df.columns = columns
df.describe()
```

	CRIM	ZN	INDUS	CHAS
count	506.000000	506.000000	506.000000	506.000000
mean	3.593761	11.363636	11.136779	0.069170
std	8.596783	23.322453	6.860353	0.253994
min	0.006320	0.000000	0.460000	0.000000
25%	0.082045	0.000000	5.190000	0.000000
50%	0.256510	0.000000	9.690000	0.000000
75%	3.647423	12.500000	18.100000	0.000000
max	88.976200	100.000000	27.740000	1.000000

II. Handling missing data

- Data in the real-world are rarely clean and homogeneous.
- Data can either be missing during data extraction or collection due to several reasons. Missing values need to be handled carefully because they reduce the quality of any of our performance matrix.
- It can also lead to wrong prediction or classification and can also cause a high bias for any given model being used. There are several options for handling missing values.
- However, the choice of what should be done is largely dependent on the nature of our data and the missing values.
- There are some techniques:
 - **Drop NULL or missing values**
 - **Fill Missing Values**
 - **Predict Missing values with an ML Algorithm**

Drop NULL or missing values

There are several useful functions for detecting, removing, and replacing null values in Pandas DataFrame :

- [isnull\(\)](#) - The `isnull()` method returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False.
- [notnull\(\)](#) - The `notnull()` method returns a DataFrame object where all the values are replaced with a Boolean value True for NOT NULL values, and otherwise False.
- [dropna\(\)](#) - The `dropna()` method removes the rows that contains NULL values. The `dropna()` method returns a new DataFrame object unless the `inplace` parameter is set to True, in that case the `dropna()` method does the removing in the original DataFrame instead.
- [fillna\(\)](#) - The `fillna()` method replaces the NULL values with a specified value. The `fillna()` method returns a new DataFrame object unless the `inplace` parameter is set to True, in that case the `fillna()` method does the replacing in the original DataFrame instead.
- [replace\(\)](#) - The `replace()` method replaces a specified phrase with another specified phrase.
- [interpolate\(\)](#) - Fill NaN values using an interpolation method.

```
df.shape
```

```
(506, 13)
```

```
df=df.dropna()
```

```
df.shape
```

```
# The result indicates that  
# there are no null values in  
# our data set
```

```
(506, 13)
```

Fill Missing Values

The most common method of handling missing values.

- This is a process whereby missing values are replaced with a test statistic like mean, median or mode of the particular feature the missing value belongs to.
- Let's suppose we have a missing value of age in the boston data set. Then the code will fill the missing value with the 30.


```
df['AGE']=df['AGE'].fillna(30)
```

```
df.shape
```

```
(506, 13)
```



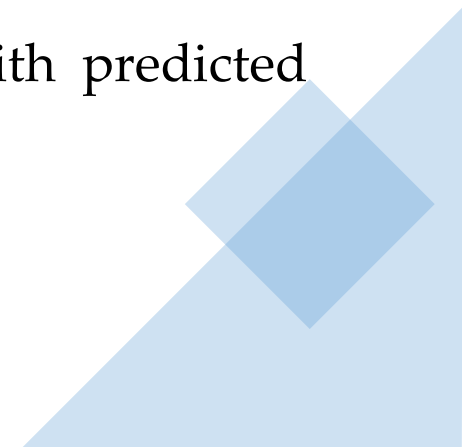
Predict Missing values with an ML Algorithm

- This is by far one of the best and most efficient methods for handling missing data.
 - Depending on the class of data that is missing, one can either use a regression or classification model to predict missing data.
- 




Steps to Follow for Predicting Missing Values

Here, we look at the simple steps required to achieve this.

- Separate the null values from the data frame (df) and create a variable “test data”
 - Drop the null values from the data frame (df) and represent them as ‘train data’
 - Create “x_train” & “y_train” from train data
 - Build the linear regression model
 - Create the x_test from test data
 - Apply the model on x_test of test data to make predictions
 - Replace the missing values with predicted values.
- 



III. Handling outliers

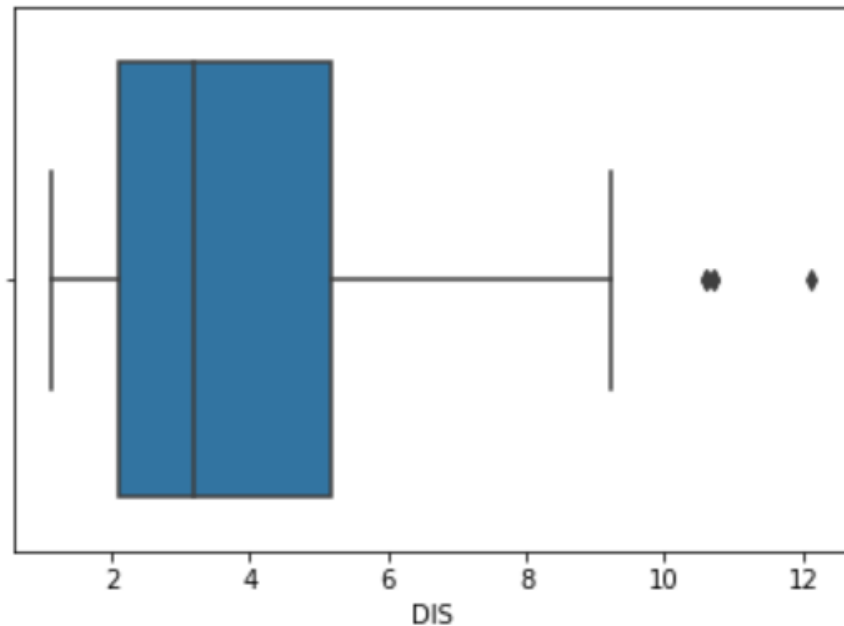
- An outlier is something which is separate or different from the crowd.
 - Outliers can be a result of a mistake during data collection or it can be just an indication of variance in your data.
 - Some of the methods for detecting and handling outliers:
 - BoxPlot
 - Scatterplot
 - Z-score
 - IQR(Inter-Quartile Range)
- 

BoxPlot

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
sns.boxplot(x=df['DIS'])  
plt.show()
```

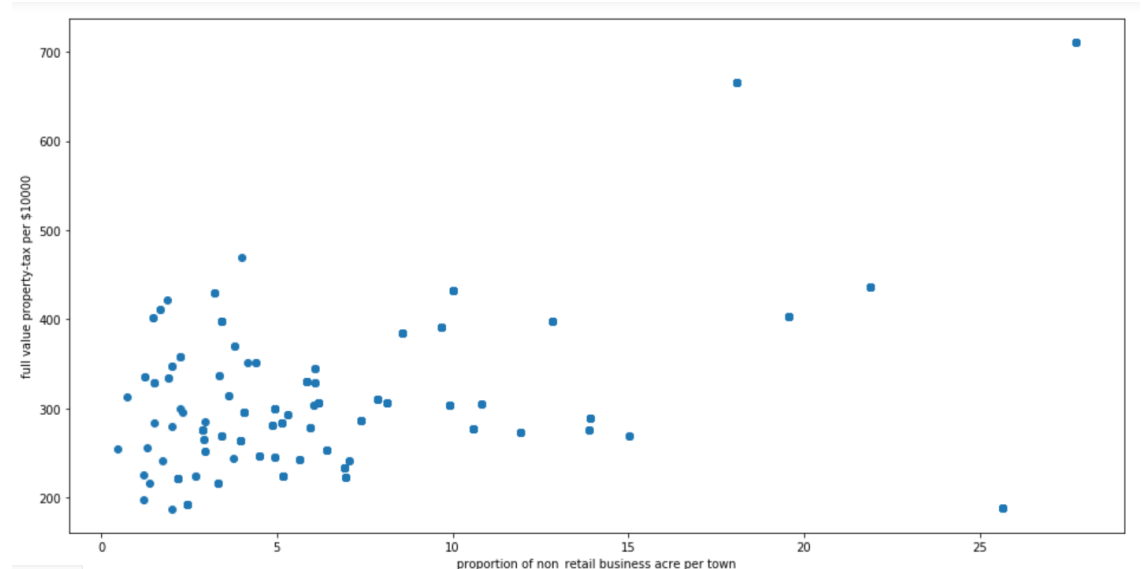


- A box plot is a method for graphically depicting groups of numerical data through their quartiles.
- The box extends from the Q1 to Q3 quartile values of the data, with a line at the median (Q2).
- The whiskers extend from the edges of the box to show the range of the data.
- Outlier points are those past the end of the whiskers.
- Boxplots show robust measures of location and spread as well as providing information about symmetry and outliers.

Scatterplot

- A scatter plot is a mathematical diagram using Cartesian coordinates to display values for two variables for a set of data.
- The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.
- The points that are far from the population can be termed as an outlier.

```
fig, ax=plt.subplots(figsize=(16, 8))  
ax.scatter(df['INDUS'], df['TAX'])  
ax.set_xlabel('proportion of non_retail business acre per town')  
ax.set_ylabel('full value property-tax per $10000')  
plt.show()
```



Z-score

- The Z-score is the signed number of standard deviations by which the value of an observation or data point is above the mean value of what is being observed or measured.
- While calculating the Z-score we re-scale and center the data and look for data points that are too far from zero.
- These data points which are way too far from zero will be treated as the outliers.
- In most of the cases a threshold of 3 or -3 is used i.e if the Z-score value is greater than or less than 3 or -3 respectively, that data point will be identified as outliers.
- We can see from the code that the shape changes, which indicates that our dataset has some outliers.

```
from scipy import stats
```

```
import numpy as np
```

```
z=np.abs(stats.zscore(df))  
print(z)
```

```
[[0.41771335 0.28482986 1.2879095 ... 1.45900038 0.44105193 1.0755623 ]  
 [0.41526932 0.48772236 0.59338101 ... 0.30309415 0.44105193 0.49243937]  
 [0.41527165 0.48772236 0.59338101 ... 0.30309415 0.39642699 1.2087274 ]  
 ...  
 [0.41137448 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.98304761]  
 [0.40568883 0.48772236 0.11573841 ... 1.17646583 0.4032249 0.86530163]  
 [0.41292893 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.66905833]]
```

```
df_outlier_Zscore=df[(z<3).all(axis=1)]  
df_outlier_Zscore.shape
```

```
(415, 13)
```


IQR

- The interquartile range (IQR) is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles.
- $IQR = Q3 - Q1$.

Once we have IQR scores below code will remove all the outliers in our dataset.

```
Q1=df.quantile(0.25)
Q3=df.quantile(0.75)
```

```
IQR = Q3-Q1
print(IQR)
```

```
CRIM      3.565378
ZN        12.500000
INDUS     12.910000
CHAS      0.000000
NOX       0.175000
RM        0.738000
AGE       49.050000
DIS       3.088250
RAD       20.000000
TAX      387.000000
PTRATIO   2.800000
B         20.847500
LSTAT     10.005000
dtype: float64
```

```
df_outlier_IQR=df[~((df<(Q1-1.5*IQR))|(df>(Q3+1.5*IQR))).any(axis=1)]
df_outlier_IQR.shape
```

```
(274, 13)
```

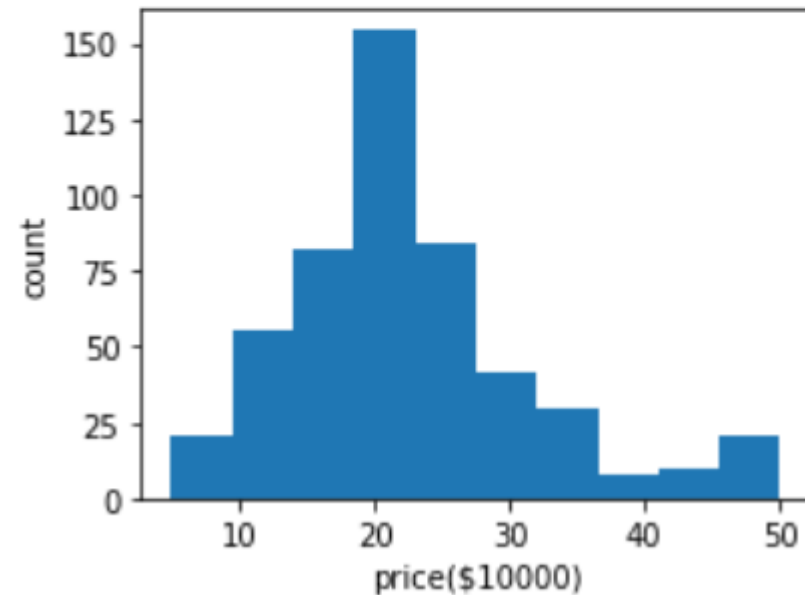
Understanding relationships and new insights through plots: Histogram

We can get many relations in our data by visualizing our dataset.

A histogram is a great tool for quickly assessing a probability distribution that is easy for interpretation by almost any audience.

Python offers a handful of different options for building and plotting histograms.

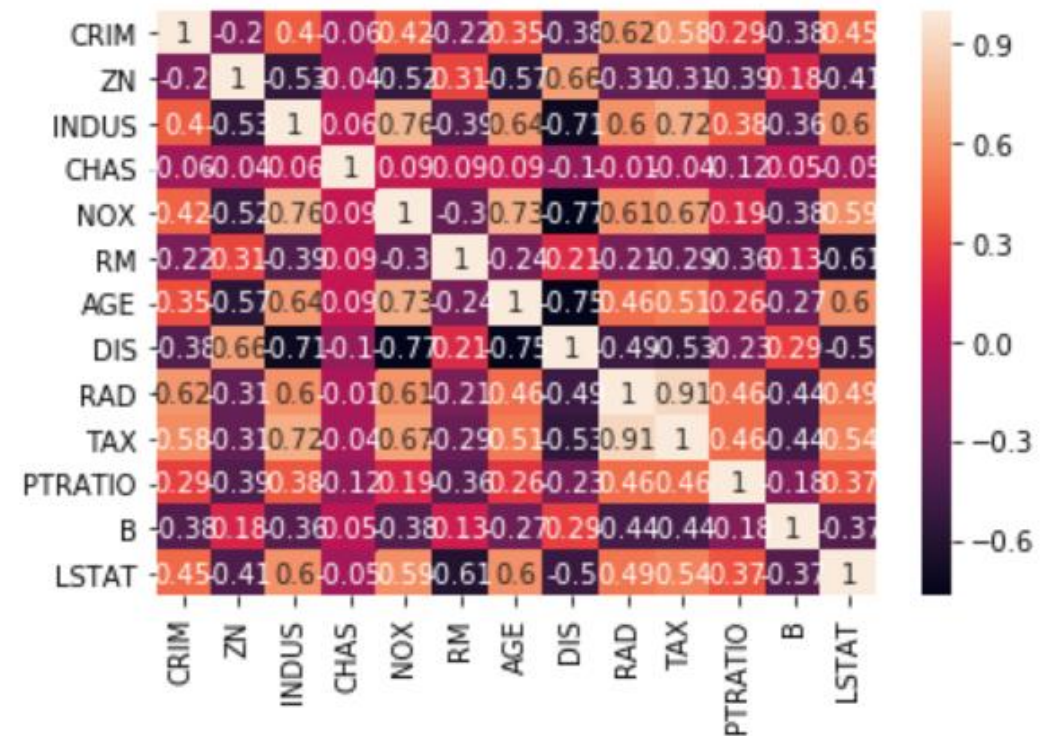
```
plt.figure(figsize=(4,3))  
plt.hist(boston.target)  
plt.xlabel('price($10000)')  
plt.ylabel('count')  
plt.tight_layout()  
plt.show()
```





Understanding relationships and new insights through plots: HeatMaps


- The Heat Map procedure shows the distribution of a quantitative variable over all combinations of 2 categorical factors.
- If one of the 2 factors represents time, then the evolution of the variable can be easily viewed using the map.
- A gradient color scale is used to represent the values of the quantitative variable.
- The correlation between two random variables is a number that runs from -1 through 0 to +1 and indicates a strong inverse relationship, no relationship, and a strong direct relationship, respectively.

```
correlation_matrix=df.corr().round(2)  
sns.heatmap(data=correlation_matrix,annot=True)  
plt.show()
```





The Tools Exploratory Data Analysis

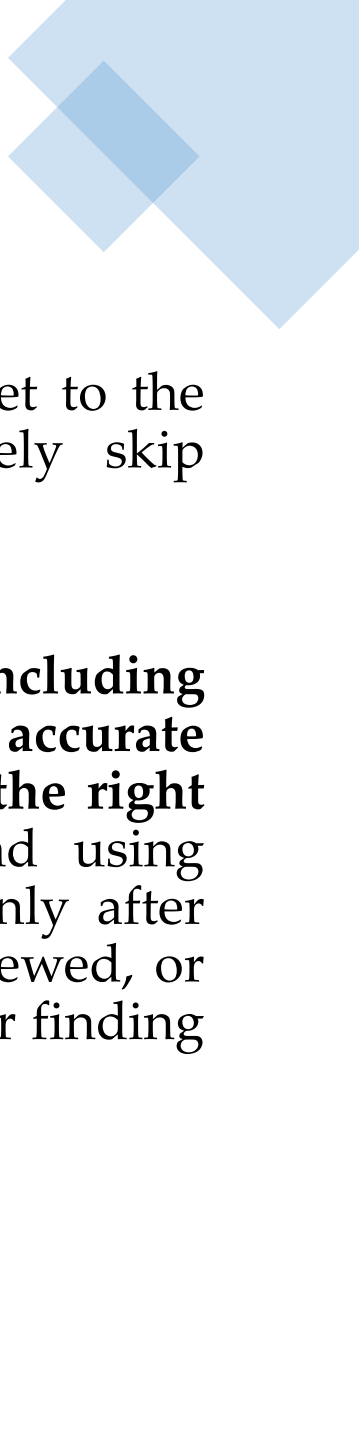
- There are plenty of open-source tools exist which automate the steps of predictive modeling like data cleaning, data visualization.
 - Some of them are also quite popular like Excel, Tableau, Qlikview, Weka and many more apart from the programming.
 - In programming, we can accomplish EDA using Python, R, SAS.
 - Some of the important packages in Python are:
 - Pandas
 - Numpy
 - Matplotlib
 - Seaborn
 - Bokeh
- 



What if no EDA?

Many Data Scientists will be in a hurry to get to the machine learning stage, some either entirely skip exploratory process or do a very minimal job.

This is a mistake with many implications, **including generating inaccurate models, generating accurate models but on the wrong data, not creating the right types of variables in data preparation**, and using resources inefficiently because of realizing only after generating models that perhaps the data is skewed, or has outliers, or has too many missing values, or finding that some values are inconsistent.



The primary purpose of EDA includes

Having covered most of what we need to know to get started with EDA, in order that we don't lose track of what we seek to achieve from all this, let us summarise our goals.



Uncovering simple efficient models with great explanatory power i.e. models which can explain the data with minimum parameters.



Estimating parameters and establish the uncertainty of those estimates



Verifying assumptions and achieving confident conclusions



Identifying outliers and anomalies



Pinpointing the important variables and factors



Maximizing insight into the underlying structure of the data set

Instead of conclusions

EDA:

- visualize the patterns in data from the beginning and to know data
- EDA is ... not preprocessing the data set

In preprocessing the data step the sub-steps are:

- Encoding categorical data
- Standardization
- Normalization
- Feature engineering
- Treatment of imbalanced data set
- Splitting the dataset into the training and test sets

Questions time:

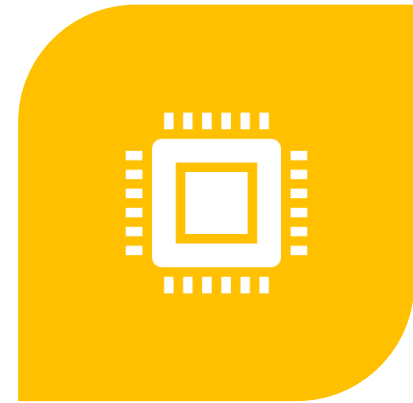
1. What is Exploratory Data Analysis (EDA) ?
2. You will use EDA?



HOW TO ENSURE WE ARE READY TO
USE MACHINE LEARNING
ALGORITHMS IN A PROJECT?



HOW TO CHOOSE THE MOST
SUITABLE ALGORITHMS FOR YOUR
DATA SET?



HOW TO DEFINE THE FEATURE
VARIABLES THAT CAN POTENTIALLY
BE USED FOR MACHINE LEARNING?