greatlearning
Power Ahead

Exploratory Data Analysis (deep dive)

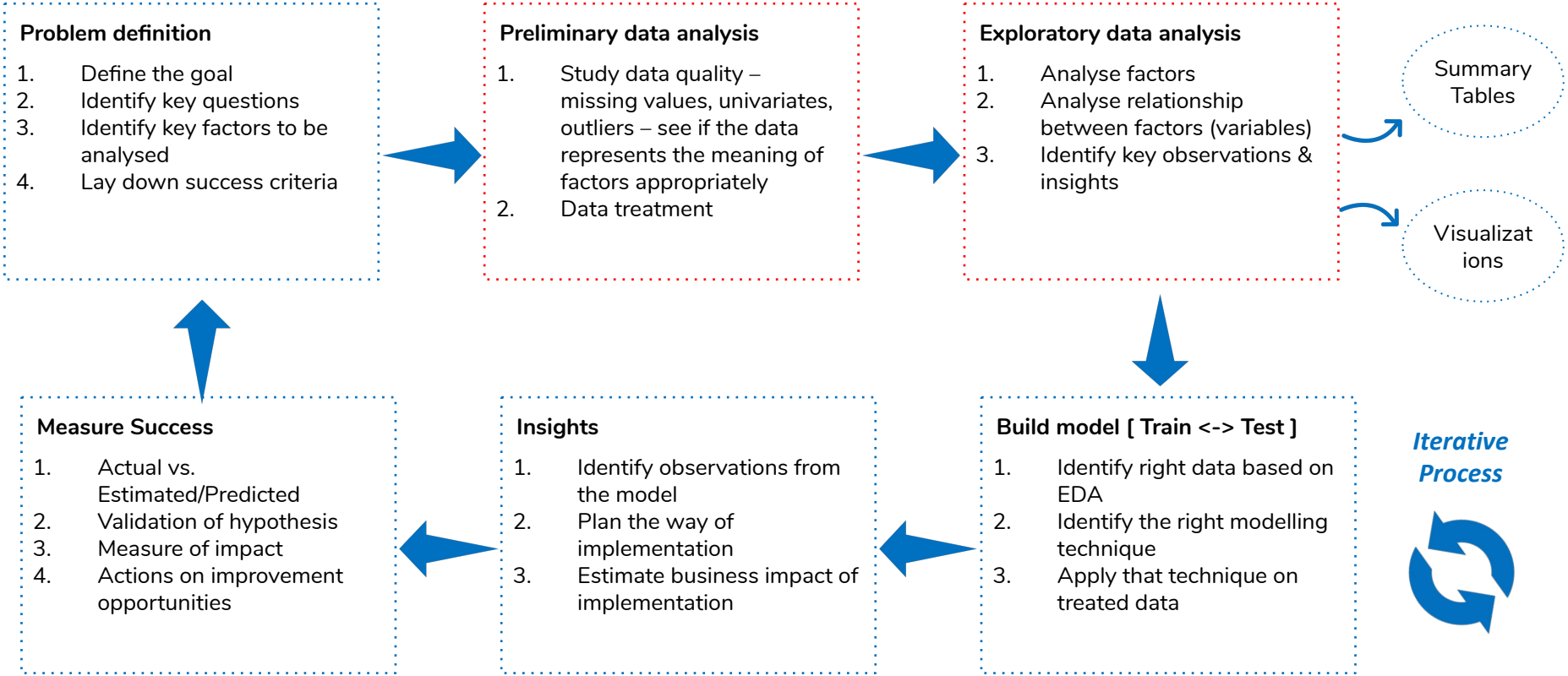
Agenda

1. Data Science Life Cycle
2. Univariate and Multivariate Analysis
3. Data Preprocessing
4. Encoding Categorical data
5. Missing values treatment
6. Working with Outliers
7. Pandas profiling

Pop Quiz

1. What do you understand by Data Science Life Cycle?
2. What is univariate and Multivariate analysis?
3. What do you understand by the term ‘Data Preprocessing’?
4. How do you handle categorical data?
5. What are the ways to handle missing values in the data?
6. How do you define outliers?

Data Science Life Cycle



Univariate and Multivariate Analysis

Univariate Analysis: Univariate analysis refers to the analysis of a single variable. It is a simplest form of analysis that summarizes and find patterns in the data. **Examples:** Frequency distribution, averages, measure of dispersion etc.

You have several options for describing data with univariate data:

- Frequency distribution tables
- Bar charts
- Histograms
- Frequency polygons

Multivariate Analysis: Multivariate analysis is used to study the interaction between more than one variable. Examples: Correlation, Regression analysis etc.

You have several options for describing data with multivariate data:

- Scatter plot
- Pair plot
- Heatmap

Data Preprocessing

Data preprocessing refers to the process of preparing the raw data into a structured format to build and train machine learning models.

Below are the important steps of Data Preprocessing. Remember, not all these steps are applicable for each problem, it is highly dependent on the data we are working with.

- Encoding Categorical data
- Missing value treatment
- Normalization and Scaling
- Working with outliers

Encoding categorical data

Sometimes, there is a need to convert categorical (non-numeric) data into numerical data to build a machine learning model. There are two ways to handle categorical data:

Label encoding - In this technique each categorical variable is assigned to a unique integer.

Country	Age	Salary
India	44	32000
US	34	33400
Japan	43	45000
US	23	23000
Japan	23	67000

Country	Age	Salary
0	44	32000
2	34	33400
1	43	45000
2	23	23000
1	23	67000

The problem with this technique is that it creates the ranks for variables. For e.g. here India < Japan < US. This affects the model interpretation

Encoding categorical data

One hot encoding - One hot encoding is a representation of categorical variables as binary values. It creates additional features based on the number of unique labels in the categorical features.

Country	Age	Salary
India	44	32000
US	34	33400
Japan	43	45000
US	23	23000
Japan	23	67000

Country.India	Country.Japan	Country.US	Age	Salary
1	0	0	44	32000
0	0	1	34	33400
0	1	0	43	45000
0	0	1	23	23000
0	1	0	23	67000

The problem with this technique is that if the categories are more, it can lead to multi-dimensional data.

Imputing Missing values

We often face the problem of missing values while handling the raw data. In order to make a good model, we need to clean all the missing values.

There are some function used to identify the missing values in the dataset.

- .isnull() - returns the dataframe of boolean values which are true for null values
- .notnull() - returns the dataframe of boolean values which are false for null values

Example:

Age	Name	Income	No. of cars
21	Adam	5000	-
25	Bill	-	5
23	Carey	3000	3
22	David	4000	-
24	Easter	-	1
26	Frank	6000	3

- In this data, we have 2 missing values in income column and 2 missing values in no. of cars column
- In order to do analysis, we need to clean the data.

Dealing with missing values

There are following ways by which we can deal with missing data:

1. Drop data
    a. **Drop the whole row -** If a row has lot of columns missing in the data, then we remove the whole row because it will not give us good results.
    b. **Drop the whole column -** If a column has lot of missing data, then we remove the whole column because it will not help us get good results.
2. Replace data
    a. **Replace with mean -** If we have continuous variable where we have missing values then we can replace those values with the mean of that column.
    b. **Replace with frequency -** If we have categorical variable where we have missing values then we can replace those values with the max frequency value of that column.
    c. **Replace with some other functions -** Sometimes we use functions min, max, kNNImputer (uses kNN algorithm) etc. to replace the missing values depending on the dataset.

Example: In the previous dataset, we can replace the missing values in income column by 4500 (avg. value) and in no. of cars column by 3 (max frequency).

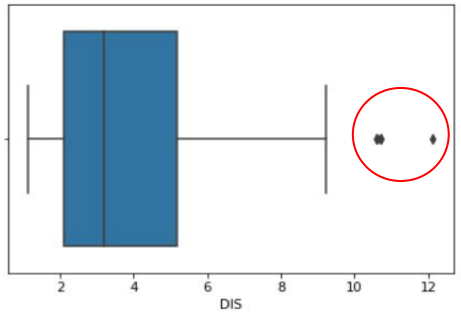
Working with Outliers

An outlier is an observation which is very distant from the other observations. In order to make a good prediction model, sometimes we have to work with the outliers.

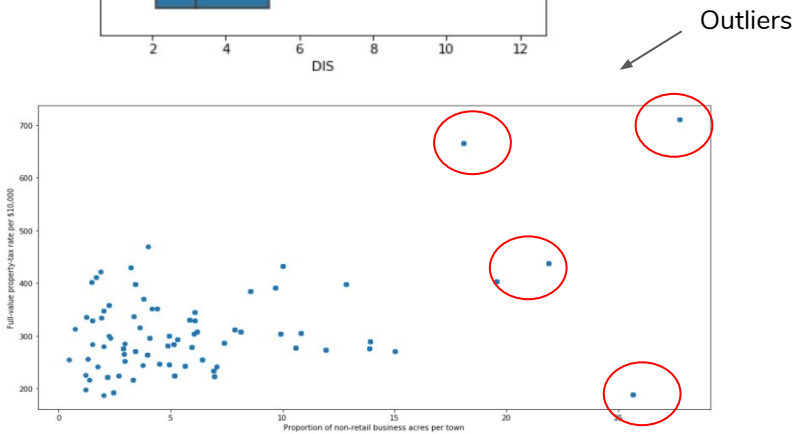
How to detect outliers

There are several ways through which we can detect outliers in the data:

1. **Box plot:** We can visualize the outliers through box plot.



2. **Scatter plot:** We can also check outliers through scatter plot.



3. **Z- Score:** We can check the z-score in order to detect outliers
4. **IQR:** It also helps detecting the outliers in the data.

Working with Outliers

How to deal with outliers?

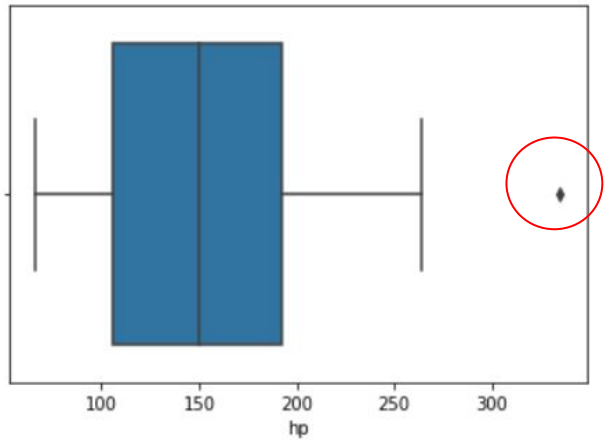
We can - remove outliers / cap them / leave them untreated. This is subjective to the business problem we are trying to solve.

There are following methods by which you can remove outliers.

1. **Z-score -** By choosing a particular threshold for z-value, we can remove the outliers greater than that threshold value.
2. **IQR -** Same as z-score, we can also set a threshold for IQR and remove the outliers.

- Here in this dataset, if we look at the hp column, row no. 30 is an outlier which we have to drop
- The lower and upper whisker values of the box plot are -23.75, 322.25 resp.
- Now we can remove the value greater than upper whisker value and our outlier will get removed.

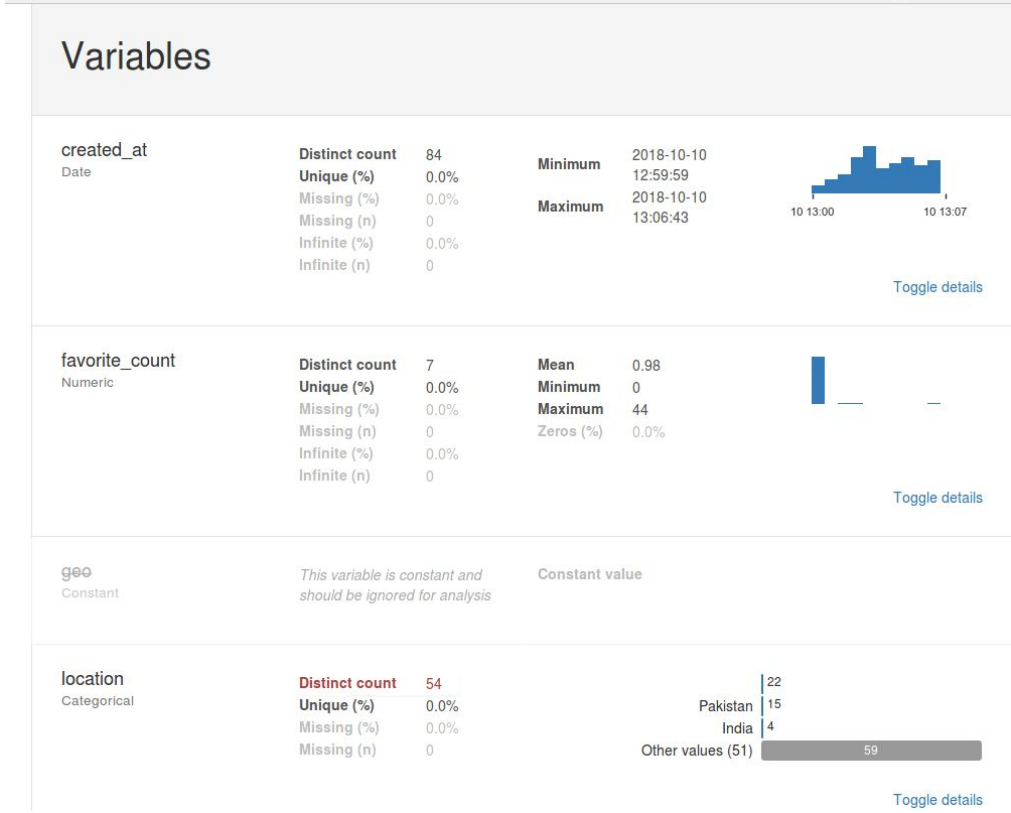
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
20	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
21	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
22	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
23	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
26	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
27	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
30	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2



Pandas Profiling

- Pandas profiling is an open source python module which helps us do quick exploratory data analysis with a few lines of code.
- It saves all the work of visualizing and checking distribution of each variable.
- It generates a report with all the information available.
- The only problem with pandas profiling is working with large datasets, it takes a lot of time to generate the report.

Sample Report



greatlearning
Power Ahead

Happy Learning !