Data Analytics Module

Capstone Project Intro

Car Price Prediction EDA

Table of Contents

Aim

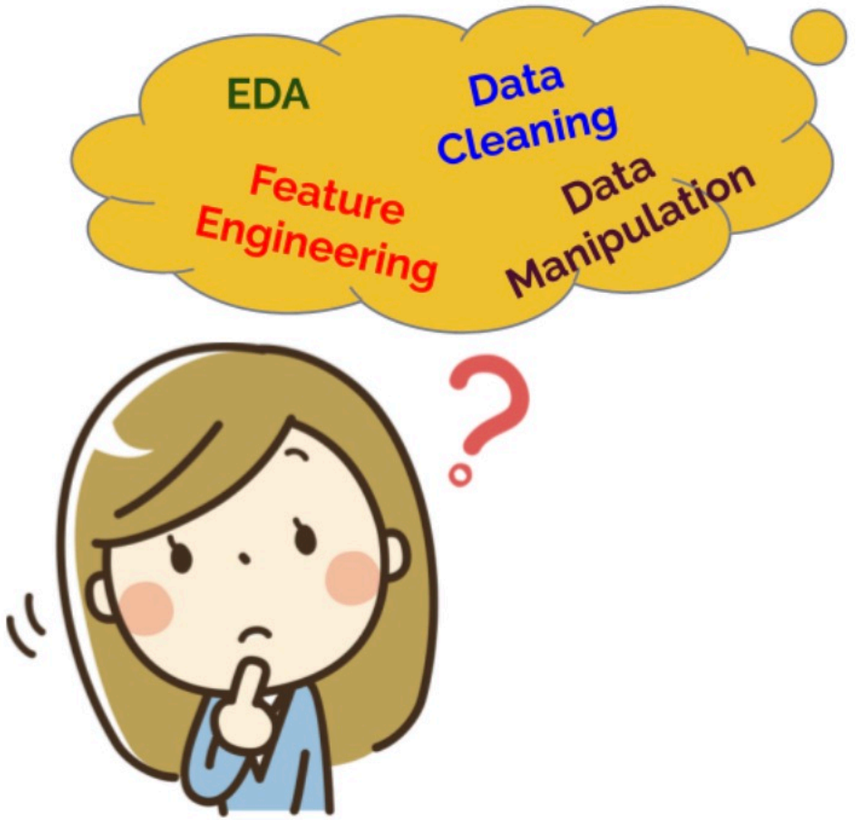
- To get the dataset ready to provide an appropriate input to ML model predicting car prices by applying Exploratory Data Analysis (EDA) process.

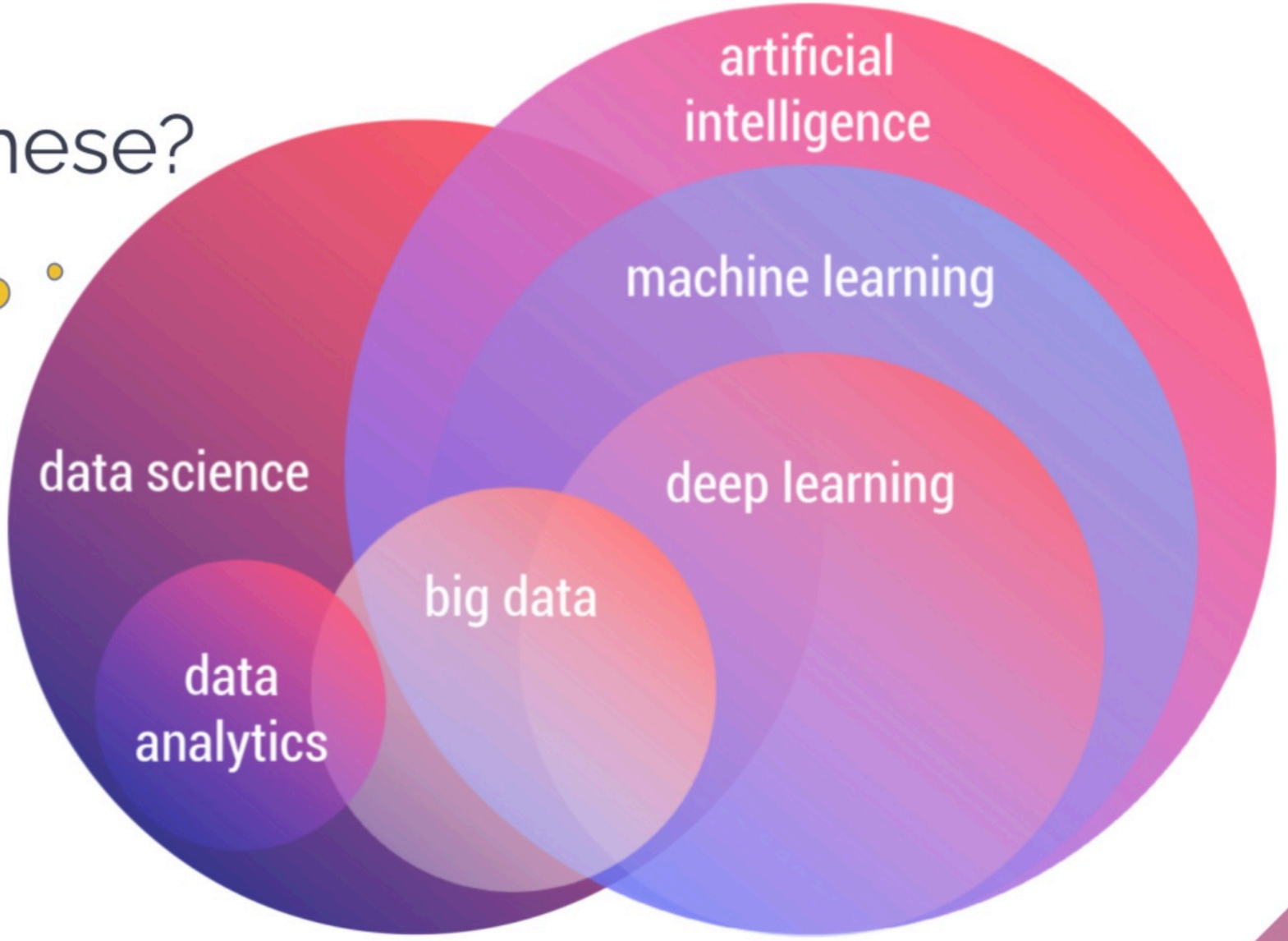
Goals

- To ensure that all our students complete all projects.
- To increase soft skill abilities within the scope of project management (self-study, group work, time planning, task sharing, etc.).

Big Picture

- Where am I?
- Why will I learn these?



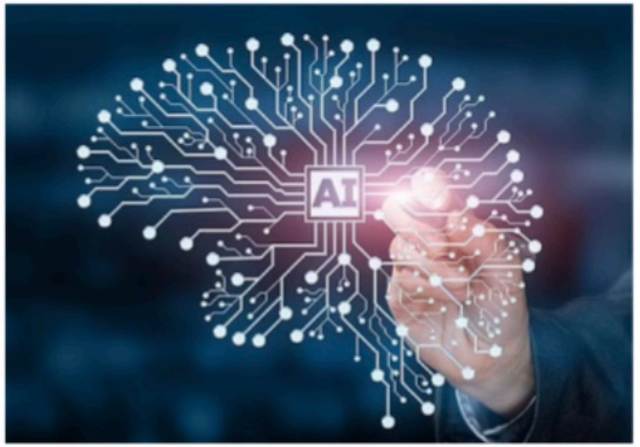


Big Picture

Data Analytics

- Excel/Google Spreadsheets
- SQL
- BI Tools (Tableau, Power BI)
- Python ...

Artificial Intelligence



- Modelling
  - Prediction/Forecasting
    - Regression
    - Classification
    - Clustering...

Big Picture

Artificial Intelligence

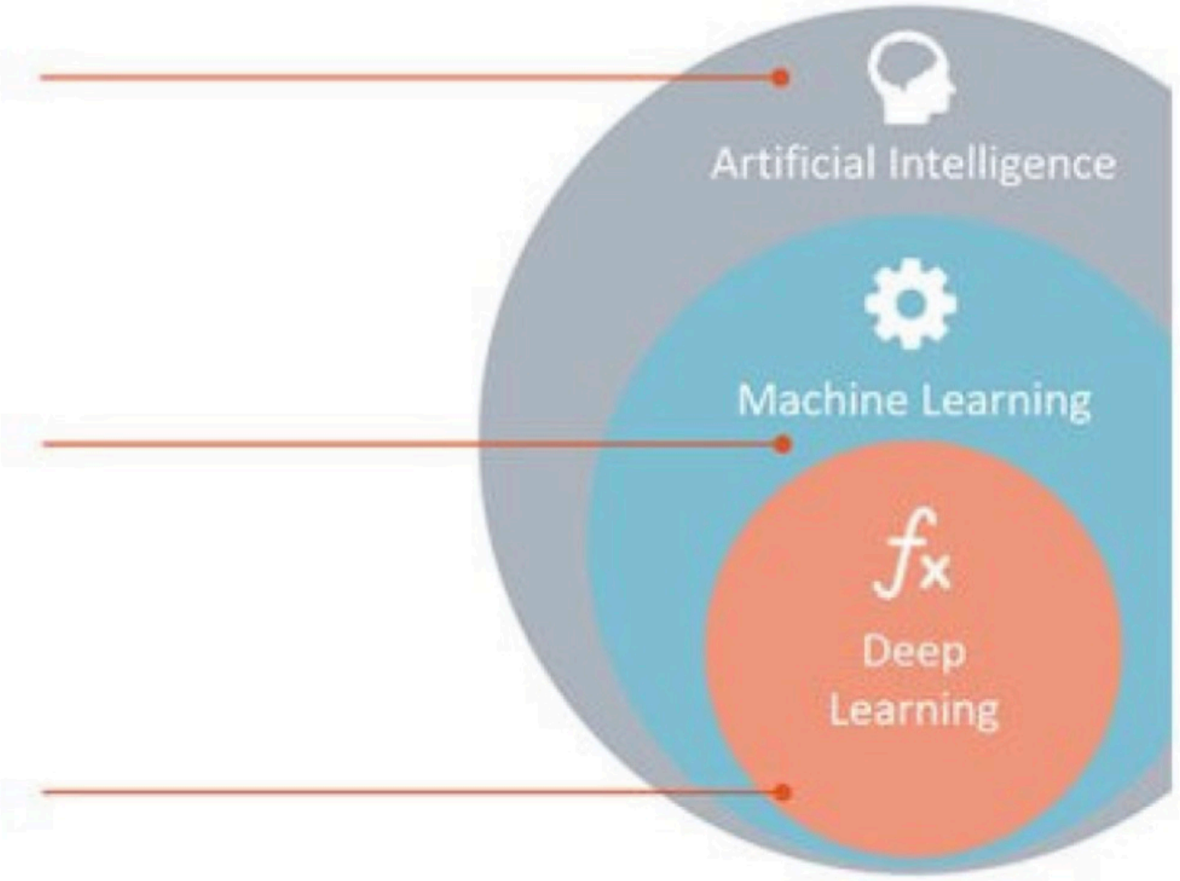
Any technique which enables computers to mimic human behavior.

Machine Learning

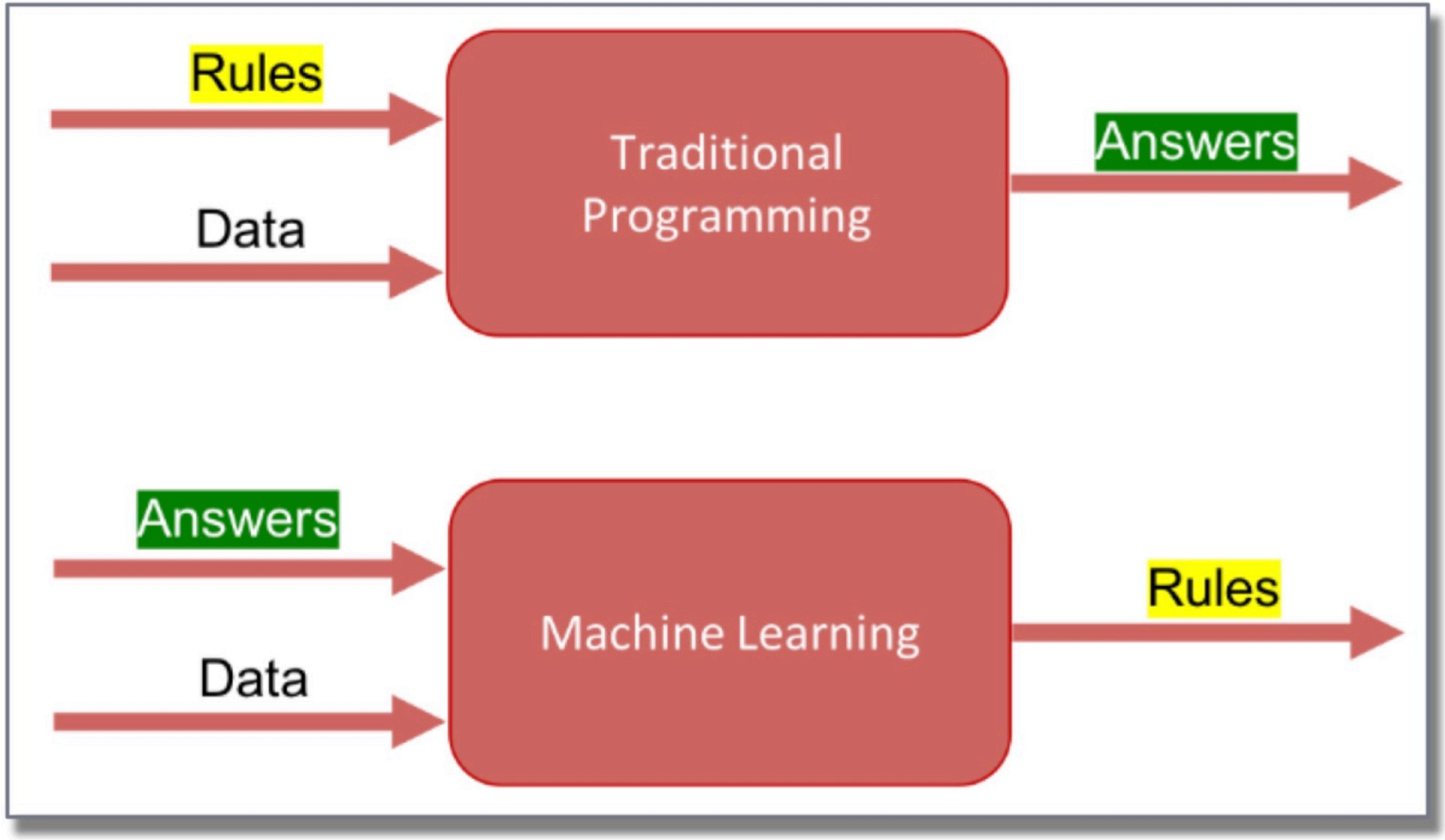
Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

Deep Learning

Subset of ML which make the computation of multi-layer neural networks feasible.



Big Picture


Rules
Data
Traditional Programming
Answers
Answers
Data
Machine Learning
Rules

Big Picture

Independent Variables , "X"

Dependent Variables

Target

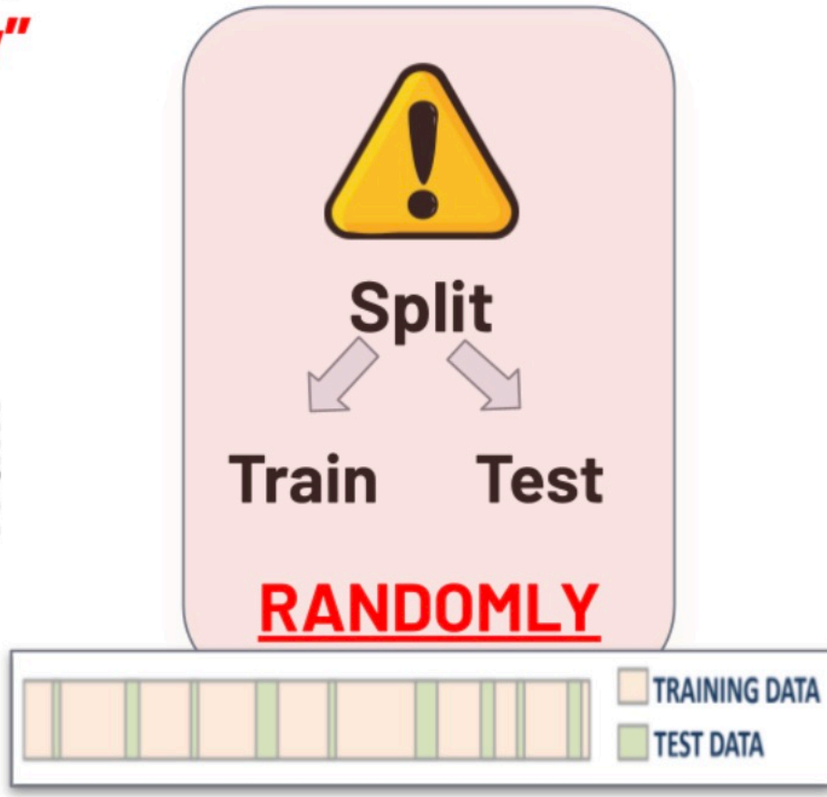
"y"

Features | **Label**

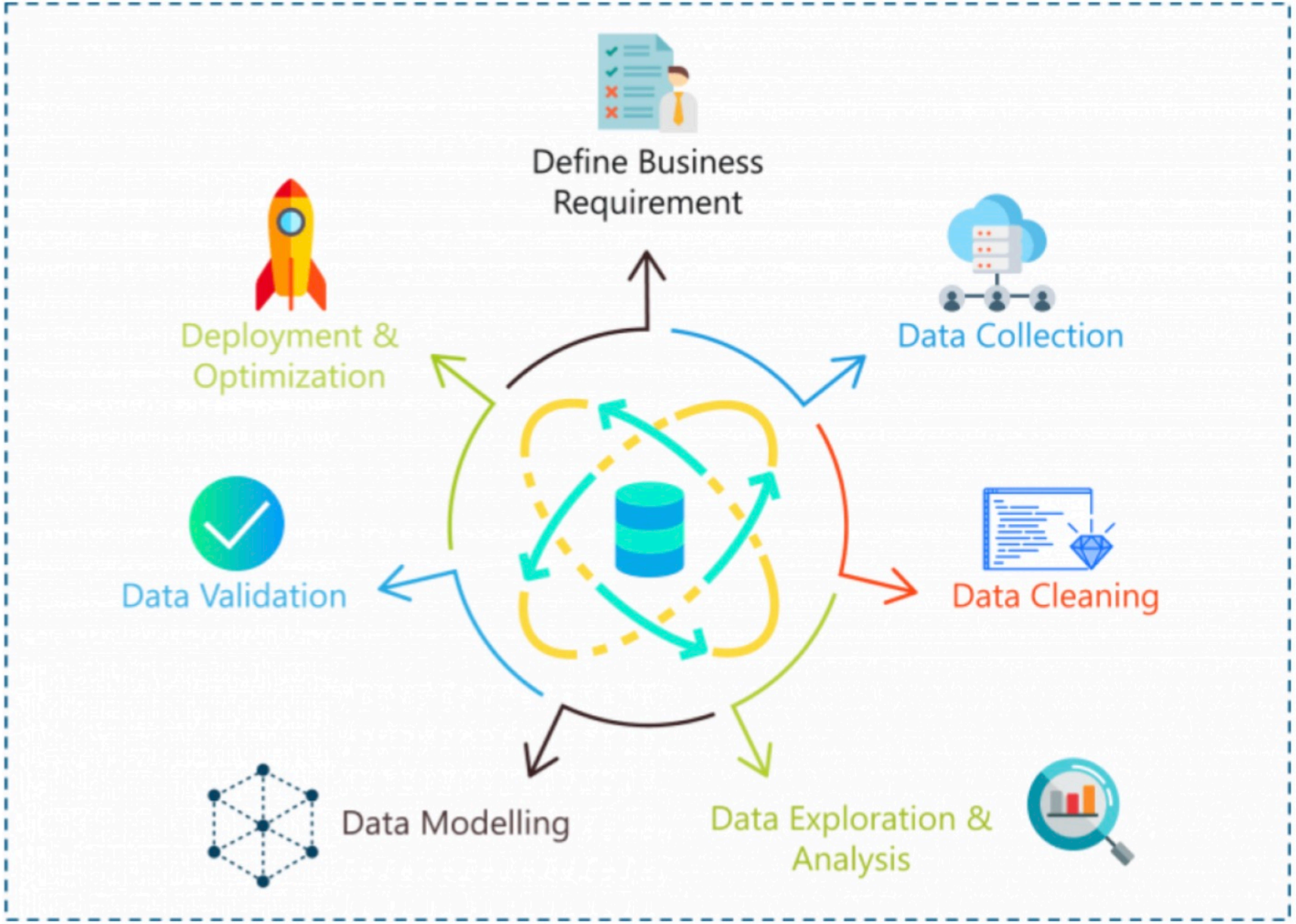
Position	Experience	Skill	Country	City	Salary ($)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

Train

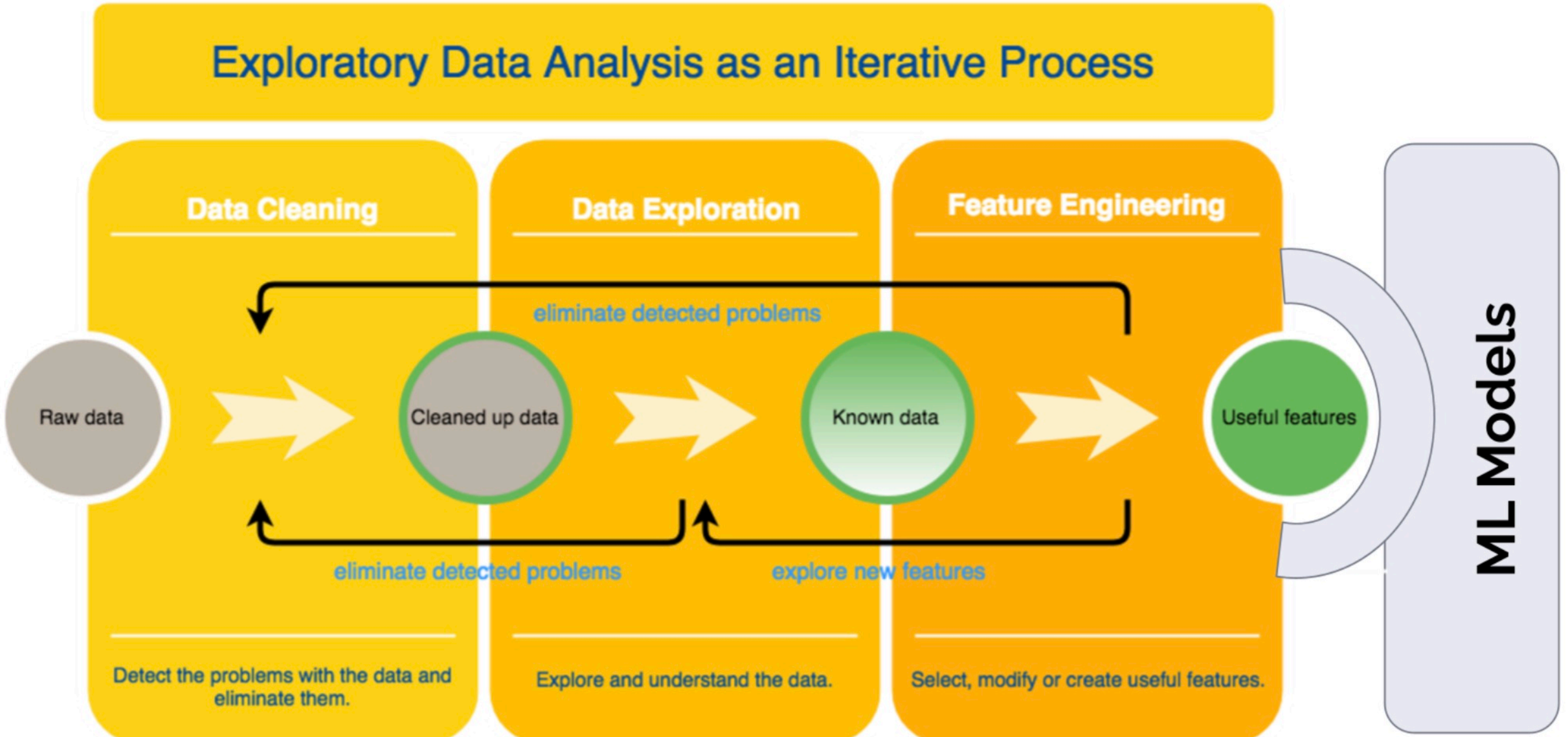
Test



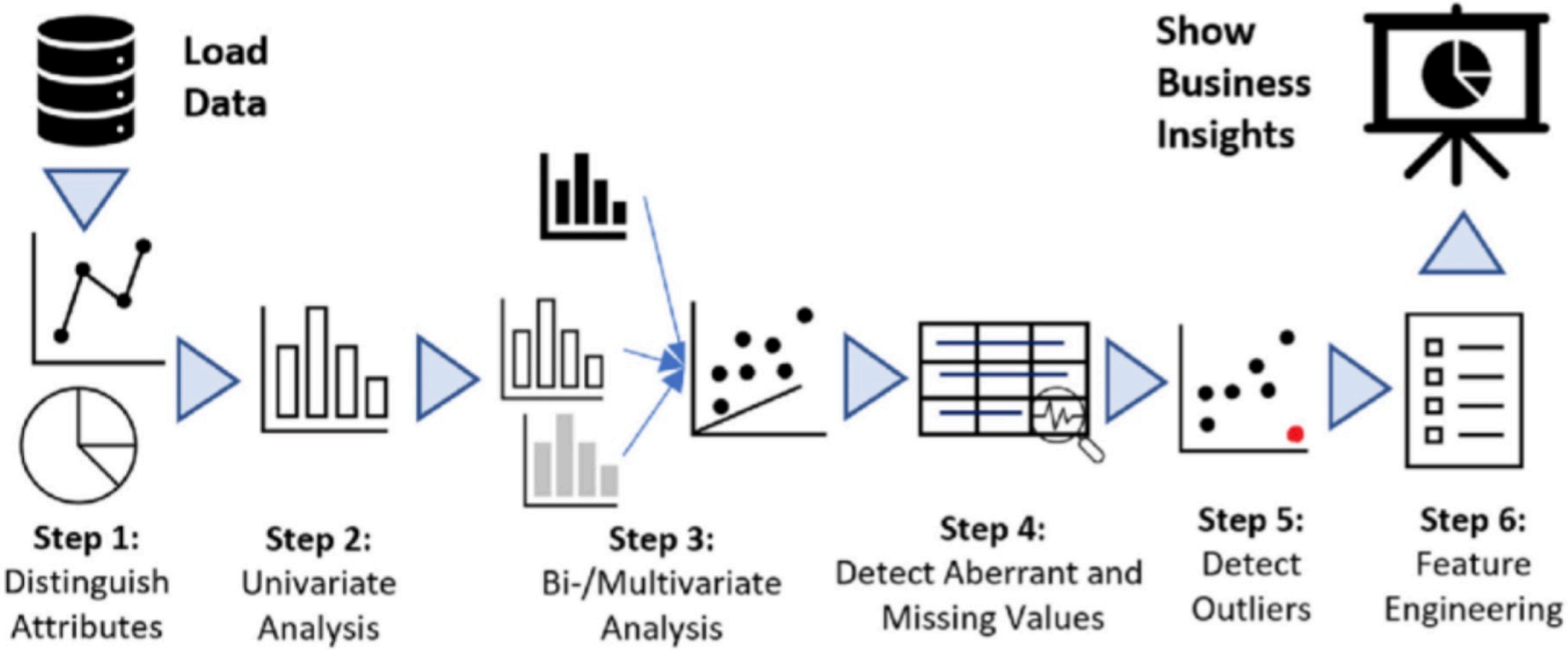
Big Picture


Define Business Requirement
Data Collection
Data Cleaning
Data Exploration & Analysis
Data Modelling
Data Validation
Deployment & Optimization

Big Picture


Exploratory Data Analysis as an Iterative Process
Data Cleaning
Data Exploration
Feature Engineering
eliminate detected problems
Raw data
Cleaned up data
Known data
Useful features
ML Models
eliminate detected problems
explore new features
Detect the problems with the data and eliminate them.
Explore and understand the data.
Select, modify or create useful features.

Big Picture


Load Data
Show Business Insights
Step 1: Distinguish Attributes
Step 2: Univariate Analysis
Step 3: Bi-/Multivariate Analysis
Step 4: Detect Aberrant and Missing Values
Step 5: Detect Outliers
Step 6: Feature Engineering

Description

- A ``.json`` file containing a dataset consisting of **15919 rows and 54 columns** is provided.
- This dataset, scraped from the online car trading company in 2019, contains many features of **9 different car models**.
- The features (variables) of this dataset are **too messy and distorted**.

What is expected of you?

- Read the "**.json**" file and assign the dataset into a "**DataFrame**" using "pandas".
- Implement all aspects of the "**EDA process**" to the dataset.
  - Fix corrupted **data formats**
  - Handle with **missing values and outliers**
    - **Domain knowledge** (automobiles) is important
    - Always use the **internet** to do the research that you need (Domain Knowledge)
    - Think carefully to decide whether a data is **outliers or not**
  - **Drop the columns/rows** you determined unnecessary as a result of your analysis
  - Use **visualization tools** while doing all these processes

What is expected of you?

```
df.head(3).T
```

	0	1
url	https://www.autoscout24.com//offers/audi-a1-sp...	https://www.autoscout24.com//offers/audi-a1-1-...
make_model	Audi A1	Audi A1
short_description	Sportback 1.4 TDI S-tronic Xenon Navi Klima	1.8 TFSI sport
body_type	Sedans	Sedans
price	15770	14500
vat	VAT deductible	Price negotiable
km	56,013 km	80,000 km
registration	01/2016	03/2017
prev_owner	2 previous owners	None
kW	NaN	NaN
hp	66 kW	141 kW
Type	[, Used, , Diesel (Particulate Filter)]	[, Used, , Gasoline]
Previous Owners	\n2\n	NaN
Next Inspection	['\n06/2021\n, \n99 g CO2/km (comb)\n]	NaN
Inspection new	['\nYes\n, \nEuro 6\n]	NaN
Warranty	['\n, \n, \n4 (Green)\n]	NaN
Full Service	['\n, \n]	NaN
Non-smoking Vehicle	['\n, \n]	NaN
null	[]	[]
Make	\nAudi\n	\nAudi\n
Model	['\n, A1, \n]	['\n, A1, \n]
Offer Number	['\nLR-062483\n]	NaN
First Registration	['\n, 2016, \n]	['\n, 2017, \n]
Body Color	['\n, Black, \n]	['\n, Red, \n]

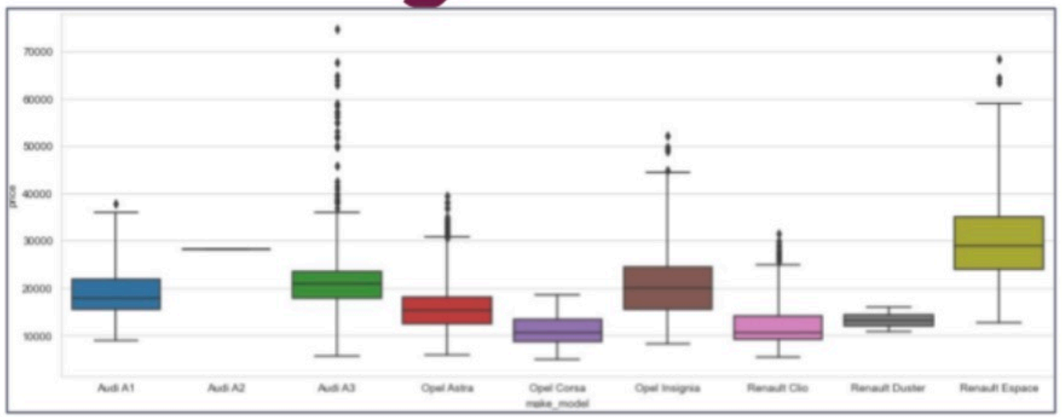
```
df.head(3).T
```

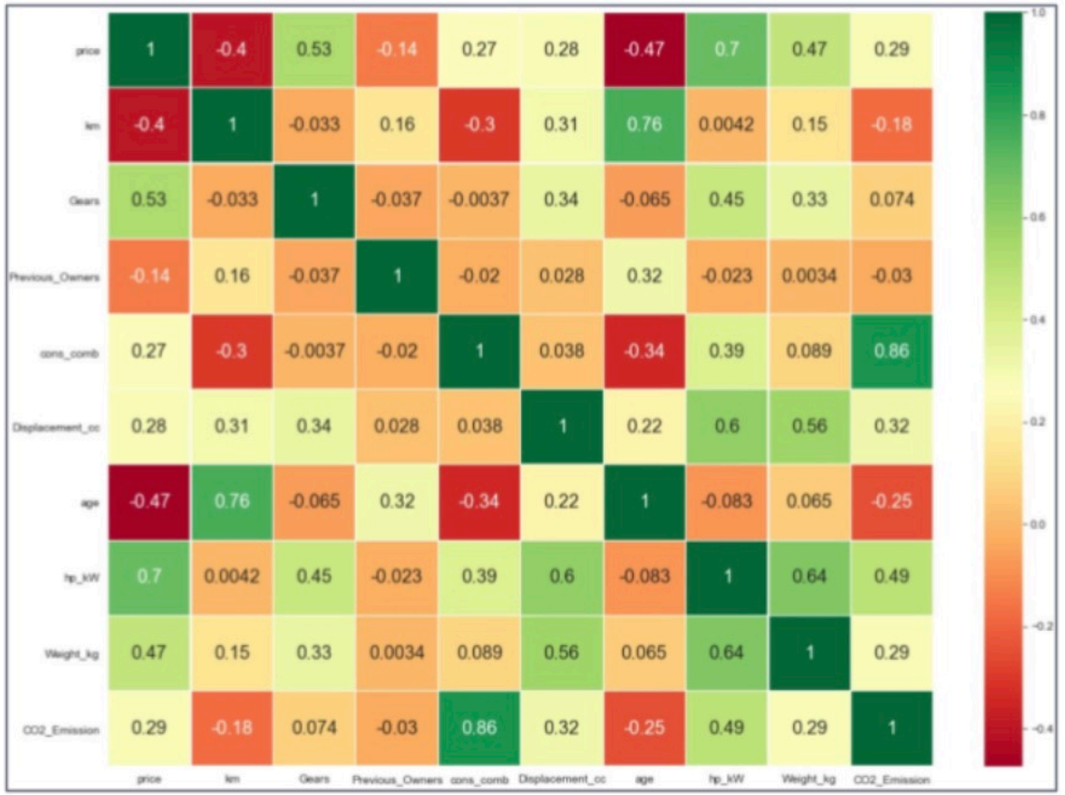
	0	1
make_model	Audi A1	Audi A1
body_type	Sedans	Sedans
price	15770	14500
vat	VAT deductible	Price negotiable
km	56013.000	80000.000
Type	Used	Used
Fuel	Diesel	Benzine
Gears	7.000	7.000
Comfort_Convenience	Air conditioning,Armrest,Automatic climate con...	Air conditioning,Automatic climate control,Hil...
Entertainment_Media	Bluetooth,Hands-free equipment,On-board comput...	Bluetooth,Hands-free equipment,On-board comput...
Extras	Alloy wheels,Catalytic Converter,Voice Control	Alloy wheels,Sport seats,Sport suspension,Voic...
Safety_Security	ABS,Central door lock,Daytime running lights,D...	ABS,Central door lock,Central door lock with r...
age	3.000	2.000
Previous_Owners	2.000	1.000
hp_kW	66.000	141.000
Inspection_new	1	0
Paint_Type	Metallic	Metallic
Upholstery_type	Cloth	Cloth
Nr_of_Doors	5.000	3.000
Nr_of_Seats	5.000	4.000
Gearing_Type	Automatic	Automatic
Displacement_cc	1422.000	1798.000
Weight_kg	1220.000	1255.000
Drive_chain	front	front
cons_comb	3.800	5.600
CO2_Emission	99.000	129.000

What is expected of you?

```
df.head(3).T
```

	0	1
make_model	Audi A1	Audi A1
body_type	Sedans	Sedans
price	15770	14500
vat	VAT deductible	Price negotiable
km	56013.000	80000.000
Type	Used	Used
Fuel	Diesel	Benzine
Gears	7.000	7.000
Comfort_Convenience	Air conditioning,Armrest,Automatic climate con...	Air conditioning,Automatic climate control,Hil...
Entertainment_Media	Bluetooth,Hands-free equipment,On-board comput...	Bluetooth,Hands-free equipment,On-board comput...
Extras	Alloy wheels,Catalytic Converter,Voice Control	Alloy wheels,Sport seats,Sport suspension,Voic...
Safety_Security	ABS,Central door lock,Daytime running lights,D...	ABS,Central door lock,Central door lock with r...
age	3.000	2.000
Previous_Owners	2.000	1.000
hp_kW	66.000	141.000
Inspection_new	1	0
Paint_Type	Metallic	Metallic
Upholstery_type	Cloth	Cloth
Nr_of_Doors	5.000	3.000
Nr_of_Seats	5.000	4.000
Gearing_Type	Automatic	Automatic
Displacement_cc	1422.000	1798.000
Weight_kg	1220.000	1255.000
Drive_chain	front	front
cons_comb	3.800	5.600
CO2_Emission	99.000	129.000





What is expected of you?

```
df.head(3).T
```

	0	1
make_model	Audi A1	Audi A1
body_type	Sedans	Sedans
price	15770	14500
vat	VAT deductible	Price negotiable
km	56013.000	80000.000
Type	Used	Used
Fuel	Diesel	Benzine
Gears	7.000	7.000
Comfort_Convenience	Air conditioning,Armrest,Automatic climate con...	Air conditioning,Automatic climate control,Hil...
Entertainment_Media	Bluetooth,Hands-free equipment,On-board comput...	Bluetooth,Hands-free equipment,On-board comput...
Extras	Alloy wheels,Catalytic Converter,Voice Control	Alloy wheels,Sport seats,Sport suspension,Voic...
Safety_Security	ABS,Central door lock,Daytime running lights,D...	ABS,Central door lock,Central door lock with r...
age	3.000	2.000
Previous_Owners	2.000	1.000
hp_kW	66.000	141.000
Inspection_new	1	0
Paint_Type	Metallic	Metallic
Upholstery_type	Cloth	Cloth
Nr_of_Doors	5.000	3.000
Nr_of_Seats	5.000	4.000
Gearing_Type	Automatic	Automatic
Displacement_cc	1422.000	1798.000
Weight_kg	1220.000	1255.000
Drive_chain	front	front
cons_comb	3.800	5.600
CO2_Emission	99.000	129.000

```
df_final.head().T
```

	0	1	2	3	4
price	15770.000	14500.000	14640.000	14500.000	16790.000
km	56013.000	80000.000	83450.000	73000.000	16200.000
Gears	7.000	7.000	7.000	6.000	7.000
age	3.000	2.000	3.000	3.000	3.000
Previous_Owners	2.000	1.000	1.000	1.000	1.000
hp_kW	66.000	141.000	85.000	66.000	66.000
Inspection_new	1.000	0.000	0.000	0.000	1.000
Displacement_cc	1422.000	1798.000	1598.000	1422.000	1422.000
Weight_kg	1220.000	1255.000	1135.000	1195.000	1135.000
cons_comb	3.800	5.600	3.800	3.800	4.100
cc_Air conditioning	1.000	1.000	1.000	0.000	1.000
cc_Air suspension	0.000	0.000	0.000	1.000	0.000
cc_Armrest	1.000	0.000	0.000	1.000	1.000
cc_Automatic climate control	1.000	1.000	0.000	0.000	1.000
cc_Auxiliary heating	0.000	0.000	0.000	1.000	0.000
cc_Cruise control	1.000	0.000	1.000	0.000	0.000
cc_Electric Starter	0.000	0.000	0.000	0.000	0.000
cc_Electric tailgate	0.000	0.000	0.000	0.000	0.000
cc_Electrical side mirrors	1.000	0.000	1.000	1.000	1.000
cc_Electrically adjustable seats	0.000	0.000	0.000	0.000	0.000
cc_Electrically heated windshield	0.000	0.000	0.000	0.000	0.000
cc_Heads-up display	0.000	0.000	0.000	1.000	0.000
cc_Heated steering wheel	0.000	0.000	0.000	0.000	0.000
cc_Hill Holder	1.000	1.000	1.000	1.000	1.000
cc_Keyless central door lock	0.000	0.000	0.000	0.000	0.000

Need to Study

- str.method
- contains()
- extract()
- get_dummies()
- add_prefix()
- sample()
- to_numeric()
- isin()
- apply()
- replace()
- split()
- join()
- regex
- def
- lambda

Assumptions

- Assume the year you are currently in is 2019

Hints

- Domain Knowledge is one of the most important things to evaluate your data.

- You have to evaluate each column by target label.

Hints

- Check the column names.

  (You can change the column names to something more useful.)

- Check the percentage of null values for each column.

  (You can drop columns having more than %... null value.)

- Check the value_counts of each column, evaluate them and take notes about what you'll do.

  (drop, similarity between columns, how to clean, define the pattern etc.)

Hints

- How to exclude each value of columns from list.

```
\n1\n                                                                                  8101
NaN                                                                                    6640
\n2\n                                                                                   766
\n0\n                                                                                   163
\n3\n                                                                                    17
                                                                                        ...
[\n1\n, \n96 g CO2/km (comb)\n]                                                          1
[\n1\n, \n181 g CO2/km (comb)\n]                                                         1
[\n1\n, \n, 6 l/100 km (comb), \n, 8 l/100 km (city), \n, 4.9 l/100 km (country), \n]    1
[\n1\n, \n, 6.7 l/100 km (comb), \n, 8.6 l/100 km (city), \n, 5.6 l/100 km (country), \n] 1
[\n1\n, \n102 g CO2/km (comb)\n]                                                         1
Name: Previous Owners, Length: 103, dtype: int64
```

```python
df["Previous_Owners"] = [item[0] if type(item) == list else item for item in df["Previous Owners"]]
df["Previous_Owners"]
```

...

```python
df["Previous_Owners2"] = df["Previous Owners"].apply(lambda item: item[0] if type(item) == list else item)
df["Previous_Owners2"]
```

Hints

- You can create functions to clean values

```
df["Fuel"].value_counts(dropna=False)
```

```
Diesel (Particulate Filter)          4315
Super 95                             4100
Gasoline                             3175
Diesel                               2984
Regular                               503
Super E10 95                          402
Super 95 (Particulate Filter)         268
```

```
benzine = ["Gasoline", "Super 95","Regular","Super E10 95","Super Plus 98","Super Plus E10 98", "Others"]
lpg = ["LPG","Liquid petroleum gas", "CNG", "Biogas", "Domestic gas H"]
def fueltype(x):
    if x in benzine:
        return "Benzine"
    elif x in lpg:
        return "LPG/CNG"
    else:
        return x
df["Fuel"] = df.Fuel.apply(fueltype)
```

Hints

- Relatively hard-to-handle columns
  - Consumption

    To create separate columns, define the patterns for each consumption type. Then evaluate which one is enough to ML Model.

```
NaN                                                                                    1906
[[3.9 l/100 km (comb)], [4.1 l/100 km (city)], [3.7 l/100 km (country)]]                304
[[4.2 l/100 km (comb)], [5 l/100 km (city)], [3.7 l/100 km (country)]]                  276
[[5.4 l/100 km (comb)], [6.8 l/100 km (city)], [4.5 l/100 km (country)]]                257
[[3.8 l/100 km (comb)], [4.3 l/100 km (city)], [3.5 l/100 km (country)]]                253
                                                                                       ...
[[3.6 l/100 km (comb)], [], [4.4 l/100 km (country)]]                                    1
[\n, 4.8 l/100 km (comb), \n, 5.6 l/100 km (city), \n, 4.3 l/100 km (country), \n]       1
[[7.6 l/100 km (comb)], [], []]                                                          1
[[5.6 l/100 km (comb)], [7.6 l/100 km (city)], [4.4 l/100 km (country)]]                 1
[\n, 4.7 l/100 km (comb), \n, \n, \n]                                                    1
Name: Consumption, Length: 882, dtype: int64
```

Hints

- Relatively hard-to-handle columns
  - Comfort_Convenience
  - Entertainment_Media
  - Extras
  - Safety_Security

How can missing values in these columns be filled?

```
NaN                                                                    1374
[Bluetooth, Hands-free equipment, On-board computer, Radio, USB]       1282
[Bluetooth, Hands-free equipment, MP3, On-board computer, Radio, USB]   982
[Bluetooth, CD player, Hands-free equipment, MP3, On-board computer, Radio, USB]   783
[On-board computer, Radio]                                              487
Name: Entertainment_Media, dtype: int64
```

```
df["Entertainment_Media"] = [",".join(item) if type(item) == list else item for item in df["Entertainment_Media"]]
```

Hints

- How to examine columns to fill missing values

```
df.groupby("age").km.describe()
```

```
df.groupby(['make_model', 'age']).km.describe()
```

```
df.groupby(['make_model',"body_type", 'age']).price.describe()
```

Hints

- How to fill missing values by groups

Example-1

```
#Step-1
#df["body_type"].fillna(df["body_type"].mode()[0])

#Step-2
#df.loc[df["make_model"]=="Audi A1", "body_type"].fillna(df[df["make_model"]=="Audi A1"]["body_type"].mode()[0])
```

```
#Step-3
for group in list(df["make_model"].unique()):
    cond = df["make_model"]==group
    mode = list(df[cond]["body_type"].mode())
    if mode != []:
        df.loc[cond, "body_type"] = df.loc[cond, "body_type"].fillna(df[cond]["body_type"].mode()[0])
    else:
        df.loc[cond, "body_type"] = df.loc[cond, "body_type"].fillna(df["body_type"].mode()[0])
```

You can generalize this loop to create your own function

Hints

Example-2

- How to fill missing values by groups

```
#Step-1
#df.["Previous_Owners"].fillna(method="ffill")

#Step-2
#df.loc[df["age"]==0, "Previous_Owners"].fillna(method="ffill")
```

```
#Step-3
for group in list(df["age"].unique()):
    cond = df["age"]==group
    df.loc[cond, "Previous_Owners"] = df.loc[cond, "Previous_Owners"].fillna(method="ffill").fillna(method="bfill")
df["Previous_Owners"] = df["Previous_Owners"].fillna(method="ffill").fillna(method="bfill")
```

You can generalize this loop to create your own function

Hints

- How to fill missing values by groups

Example-3

```
# Step-1

# df.["Paint_Type"].fillna(method="ffill")

# Step-2
# df.loc[df["make_model"]=="Audi A1", "Paint_Type"].fillna(method="ffill")

# Step-3
# for group in list(df["make_model"].unique()):
#     cond = df["make_model"]==group
#     df.loc[cond, "Paint_Type"] = df.loc[cond, "Paint_Type"].fillna(method="ffill").fillna(method="bfill")
# df["Paint_Type"] = df["Paint_Type"].fillna(method="ffill").fillna(method="bfill")
```

```
# Step-4

for group1 in df["make_model"].unique():
    for group2 in list(df["body_type"].unique()):
        cond2 = (df["make_model"]==group1) & (df["body_type"]==group2)
        df.loc[cond2, "Paint_Type"] = df.loc[cond2, "Paint_Type"].fillna(method="ffill").fillna(method="bfill")

for group1 in list(df["make_model"].unique()):
    cond1 = df["make_model"]==group1
    df.loc[cond1, "Paint_Type"] = df.loc[cond1, "Paint_Type"].fillna(method="ffill").fillna(method="bfill")

df["Paint_Type"] = df["Paint_Type"].fillna(method="ffill").fillna(method="bfill")
```

Hints

- Dummy Operation

(The get_dummies function has 2 different uses)

```
pd.get_dummies(df)
```

need to be research

```
df["col_name"].str.get_dummies(sep = " , ")
```

Hints

- pd.factorize()
- count()
- map()
- cat.codes

- LabelEncoder()
- OneHotEncoder()

Capstone Project Period

Capstone Project Period Duration

03 November - 12 November 2022

Self Study + Group Study → Project Solutions

1 Breakout Room Sessions

07 November 2022 (Monday)

2 Project Solution Sessions

10 November 2022 (Thursday)
12 November 2022 (Saturday)