

# DA Capstone (AutoScout) Intro\_new

C013 EU Auto Scout Project\_Info Session

Training Clarusway

Pear Deck - March 6, 2023 at 8:01PM

## Part 1 - Summary

Use this space to summarize your thoughts on the lesson

## Part 2 - Responses

Slide 1



Use this space to take notes:

## Slide 2

# ▶ Car Price Prediction EDA

CLARUSWAY®  
WAY TO REINVENT YOURSELF



Use this space to take notes:

## Slide 3

### ▶ Table of Contents

- ▶ Aim & Goals
- ▶ Big Picture
- ▶ Description
- ▶ What is expected of you?
- ▶ Need to Study
- ▶ Assumptions
- ▶ Hints



3

Use this space to take notes:

## Slide 4

### Your Response

You Chose

## Slide 4

I've started working on the project and examined the dataset?

True

False

Students choose an option

Pear Deck Interactive Slide  
Do not remove this bar

## Your Response

- **False**

Other Choices

- True

Use this space to take notes:

## Slide 5

### ► 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.).

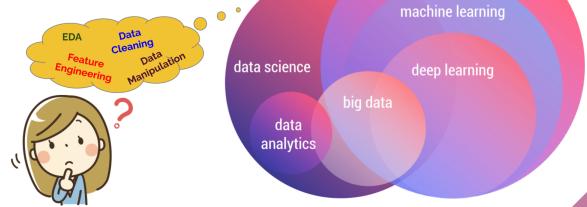


Use this space to take notes:

## Slide 6

### ► Big Picture

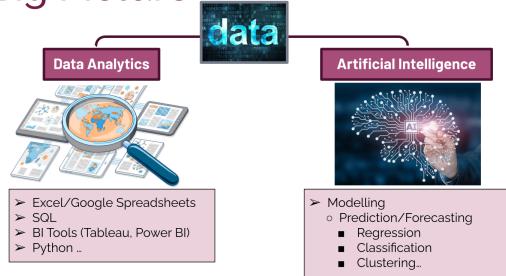
- ▶ Where am I?
- ▶ Why will I learn these?



Use this space to take notes:

## Slide 7

### ► Big Picture



Use this space to take notes:

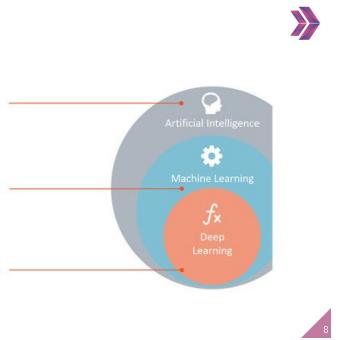
## Slide 8

### ► 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.

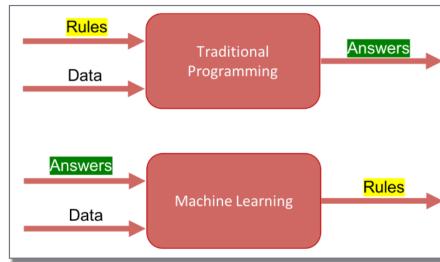


8

Use this space to take notes:

## Slide 9

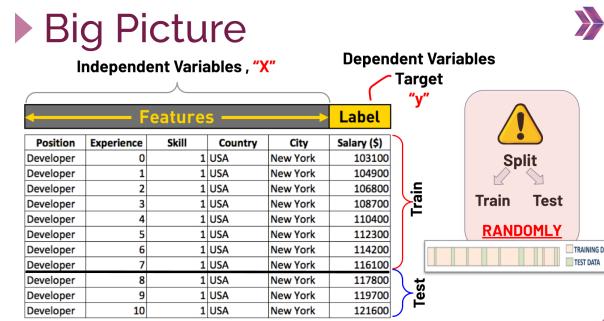
### ► Big Picture



9

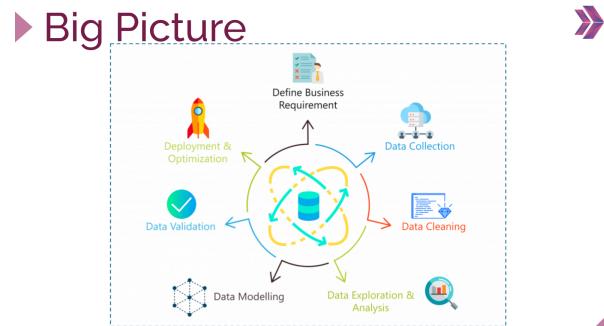
Use this space to take notes:

## Slide 10



Use this space to take notes:

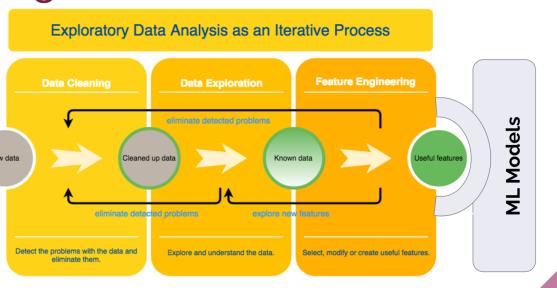
## Slide 11



Use this space to take notes:

## Slide 12

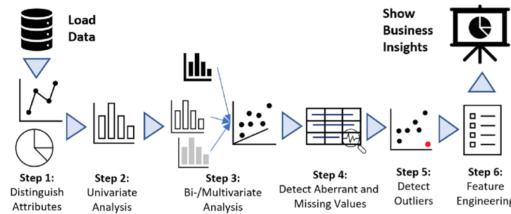
### ► Big Picture



Use this space to take notes:

## Slide 13

### ► Big Picture



13

Use this space to take notes:

## Slide 14

### ► Description ➤

- ▶ A ``.json`` file containing a dataset consisting of **29480 rows and 58 columns** is provided.
- ▶ This dataset, scraped from the online car trading company in 2023, contains many features of **various number of car models**.
- ▶ The features (variables) of this dataset are **too messy and distorted**.

14

Use this space to take notes:

## Slide 15

### ► 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

15

Use this space to take notes:

Slide 16

## ► What is expected of you? ►

Af-Header(1..1)		Af-Header(1..1)	
url	https://www.autodealersCA.com/used/442	url	https://www.autodealersCA.com/used/442
make	Audi	make	Audi
model	A4	model	A4
share_descryption	North Park 10-12 E. 10th Street, Jensen Beach, FL 34950	share_descryption	1.8 TFSI sport
body_type	Sedan	body_type	Sedan
year	2019	year	2019
mil	VAT deductible	mil	Price negotiable
km	56,913 km	km	85,000 km
regeneration_date	03/03/2017	regeneration_date	03/03/2017
prev_mileage	2,000 miles	prev_mileage	2,000 miles
kw	N/A	kw	N/A
hp	141 kW	hp	141 kW
l	U-Select - Diesel (Performance)	l	U-Select - Diesel (Performance)
Previous Owners	1	Previous Owners	1
Last Inspection	[2020-08-14] 60 day(s) Children (passenger)	Last Inspection	n/a
Inspection Status	[Vehicle is valid]	Inspection Status	n/a
Warranty	[n..v..]	Warranty	n/a
Full Service	[n..v..]	Full Service	n/a
Non-existing service	[n..v..]	Non-existing service	n/a
af	8	af	8
Make	Volkswagen	Make	Volkswagen
Model	EA, A4, A4	Model	EA, A4, A4
Offer_Guarantee	[n..2019..n]	Offer_Guarantee	[n..2017..n]
Five Registries	[n..2016..n]	Five Registries	[n..2017..n]
Body Color	[n..Black..n]	Body Color	[n..Red..n]

Use this space to take notes:

Slide 17

## ► What is expected of you? ➤

The figure displays the performance of various machine learning models on the 'adult' dataset. The top section is a box plot showing the distribution of AUC values for different models. The bottom section is a heatmap comparing the AUC values for all pairs of models across four datasets: 'adult', 'adult\_wage', 'adult\_census', and 'adult\_census\_wage'. The color scale indicates the AUC value, ranging from 0.5 (red) to 1.0 (green).

Use this space to take notes:

Slide 18

## ► What is expected of you? ➤

df[1].head(1)	df[2].head(1)				
	0	1	2	3	4
make, model	Audi A1				
body_type	Sedan	Sedan	Sedan	Sedan	Sedan
year	2014	2014	2014	2014	2014
vat	VAT included	Price negotiable	Price negotiable	Price negotiable	Price negotiable
kms	560130000	66000000	66000000	66000000	66000000
Type	Fuel	Diesel	Diesel	Diesel	Diesel
Fuel	Diesel	Diesel	Diesel	Diesel	Diesel
Gears	6	7	7	7	7
Country	Germany	Germany	Germany	Germany	Germany
Condition	All conditioning	All automatic	All automatic	All automatic	All automatic
Entertainment_Media	Bluetooth hands free equipment included				
Exterior	All wheel drive	Catalytic Converter Control	Catalytic Converter Control	Catalytic Converter Control	Catalytic Converter Control
Safety_Security	ABS control disc lock	Daytime running lights	Daytime running lights	Daytime running lights	Daytime running lights
Previous_Owner	2	1	1	1	1
No_MM	66.000	141.000	141.000	141.000	141.000
Impression	+	+	+	+	+
Power_Type	Metallic	Metallic	Metallic	Metallic	Metallic
Upkeep_km	50000	50000	50000	50000	50000
No_Mileage	6.000	6.000	6.000	6.000	6.000
Gearing_Type	Automatic	Automatic	Automatic	Automatic	Automatic
Dimension	4320x1780x1430	4320x1780x1430	4320x1780x1430	4320x1780x1430	4320x1780x1430
Weight_kg	1200.000	1200.000	1200.000	1200.000	1200.000
Drive_chain	Front	Front	Front	Front	Front
Length_mm	4320	4320	4320	4320	4320
CO2_Emission	99.000	120.000	120.000	120.000	120.000

18

Use this space to take notes:

Slide 19

## ► Need to Study

- ▶ str.method
  - ▶ contains()
  - ▶ extract()
  - ▶ get\_dummies()
  - ▶ add\_prefix()
  - ▶ sample()
  - ▶ to\_numeric()
  - ▶ isin()
  - ▶ apply()
  - ▶ replace()
  - ▶ split()
  - ▶ join()
  - ▶ regex
  - ▶ def
  - ▶ lambda

1

Use this space to take notes:

## Slide 20

### ► Assumptions



- ▶ Assume the year you are currently in is 2022

20

Use this space to take notes:

## Slide 21

### ► Hints



- ▶ Domain Knowledge is one of the most important things to evaluate your data.
- ▶ You have to evaluate each column by target label.

21

Use this space to take notes:

## Slide 22

### ► 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.)

22

Use this space to take notes:

## Slide 23

### ► Hints



- ▶ How to exclude each value of columns from list.

```
\n1\n\nNaN\nNaN\nNaN\nNaN\nNaN\n\n[\\n\\n, \\n96 g CO2/km (comb)\\n]\n[\\n\\n, \\n181 g CO2/km (comb)\\n]\n[\\n\\n, \\n182 g CO2/km (comb)\\n], 8 1/100 km (city), \\n, 4.9 1/100 km (country), \\n]\n[\\n\\n, \\n182 g CO2/km (comb)\\n], 6.7 1/100 km (city), \\n, 8.6 1/100 km (country), \\n]\n[\\n\\n, \\n182 g CO2/km (comb)\\n]\nName: Previous Owners, Length: 103, dtype: int64
```

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

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

23

Use this space to take notes:

## Slide 24

### 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)
```



24

Use this space to take notes:

## Slide 25

### 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.

```
Nah:
[[3.9 1/100 km (comb)], [4.1 1/100 km (city)], [3.7 1/100 km (country)]]           1986
[[4.2 1/100 km (comb)], [5 1/100 km (city)], [3.7 1/100 km (country)]]             304
[[4.4 1/100 km (comb)], [6.8 1/100 km (city)], [4.5 1/100 km (country)]]            276
[[3.8 1/100 km (comb)], [4.3 1/100 km (city)], [3.5 1/100 km (country)]]            257
[[3.6 1/100 km (comb)], [], [4.4 1/100 km (country)]]                                ...
[[\n, 4.8 1/100 km (comb), \n, 5.6 1/100 km (city), \n, 4.3 1/100 km (country), \n]       1
[[7.6 1/100 km (comb)], [], []]                                                       1
[[5.6 1/100 km (comb)], [7.6 1/100 km (city)], [4.4 1/100 km (country)]]            1
[\n, 4.7 1/100 km (comb), \n, \n, \n]                                                 1
Name: Consumption, Length: 882, dtype: int64
```



25

Use this space to take notes:

## Slide 26

### ► 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

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

28

Use this space to take notes:

## Slide 27

### ► 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()
```

27

Use this space to take notes:

## Slide 28

### ► 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 len(mode) > 1:  
        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 in your analysis

28

Use this space to take notes:

## Slide 29

### ► Hints

- ▶ How to fill missing values by groups

Example-2

```
#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

29

Use this space to take notes:

## Slide 30

### ► 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()):
        cond1 = df[“make_model”]==group1
        cond2 = df[“body_type”]==group2
        df.loc[cond1, “Paint_Type”] = df.loc[cond1, “Paint_Type”].fillna(method=“ffill”).fillna(method=“bfill”)
        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”)
```

30

Use this space to take notes:

## Slide 31

### ► 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 = “ ”)

31

Use this space to take notes:

## Slide 32

### ► Hints



- ▶ pd.factorize()
- ▶ count()
- ▶ map()
- ▶ cat.codes
- ▶ LabelEncoder()
- ▶ OneHotEncoder()

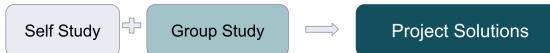
Use this space to take notes:

32

## Slide 33

### ► Capstone Project Period

*Capstone Project Period Duration*  
06 March - 11 March 2023



**Project Solution Sessions**

11 March 2023 (Saturday)

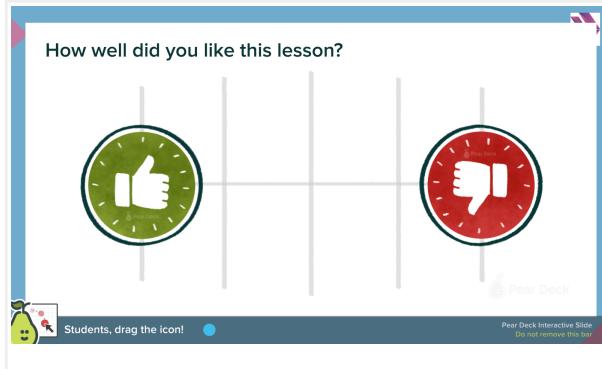
33

Use this space to take notes:

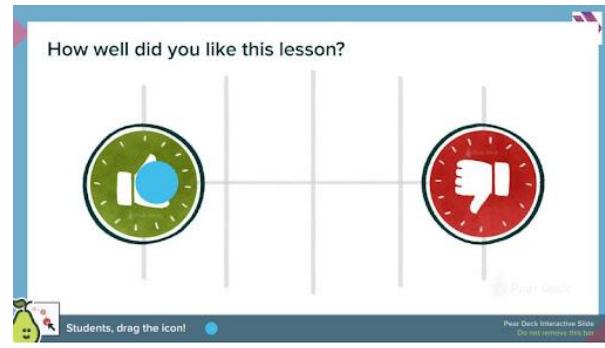
## Slide 34

## Your Response

## Slide 34



## Your Response



Use this space to take notes:

## Slide 35

# THANKS!

**Any questions?**

You can find us at:

- ▶ #questions-answers@Slack



Use this space to take notes: