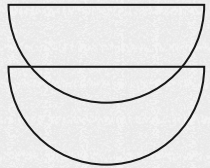
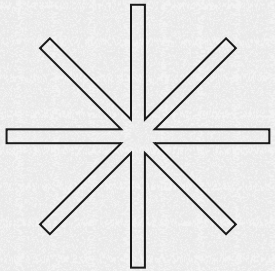
Data Cleaning & Preprocessing

Vanya Valindria, PhD

Outlines

- Data Cleaning in structured data
- Preprocessing for unstructured data
- Importance of Data quality

01

Data Cleaning in Structured Data

Data Cleaning

The process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

Why data cleaning?

Raw data is often noisy, incomplete, and inconsistent, which can negatively impact the performance of the ML model.

Data Cleaning

- Remove duplicates or unwanted observations
- Fixing structural errors
- Data transformation
- Handling missing data
- Managing unwanted outliers

Raw Data

A shop “Ratu Susu” has 35 items sold in the shop.

The product catalogue dataset contains features for predicting the best product to be sold in “Ratu Susu” to get the optimum revenue.

As a data scientist, you have to clean the dataset first before modelling the solution.

Raw Data

```
product_catalogue = pd.read_csv("Product_cat_raw.csv")
```

```
product_catalogue.tail(10)
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
34	Susu Bebelove	2023-02-28	46.521	203	76.538	390.0	597	24509	469	70
35	Susu Greenfields	2022-11-08	64.694	209	108.040	436.0	732	28200	568	43
36	Susu Aptamil	2023-02-09	51.549	2	137.663	567.0	792	45	460	46
37	Susu Neocate	2022-05-17	53.782	118	73.045	557.0	732	24878	460	54
38	Susu Isomil	2022-07-30	41.122	223	74.590	479.0	786	21309	487	48
39	Susu Alula	2022-07-23	30.192	155	119.388	539.0	635	26399	470	29
40	Susu Bebelove	2023-02-28	46.521	203	76.538	390.0	597	24509	469	70
41	Susu Enfamil	2022-03-	56.536	203	124.449	456.0	893	28858	451	55

Data Cleaning

- Remove duplicates or unwanted observations
- Fixing structural errors
- Data transformation
- Handling missing data
- Managing unwanted outliers

Describe the data

Let's see the descriptive structure of the data

```
product_catalog.describe()
```

	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
count	35.000000	35.000000	35.000000	35.000000	35.000000	35.000000	35.000000	35.000000
mean	53.029257	0.183114	103.577429	534.571429	0.702143	24389.771429	0.488286	47.142857
std	11.217603	0.043677	20.073901	93.166608	0.104540	4880.654362	0.084994	10.819761
min	24.470000	0.114000	69.275000	390.000000	0.546000	13882.000000	0.293000	22.000000
25%	48.038000	0.155000	87.363500	457.000000	0.613000	21305.500000	0.448500	41.500000
50%	53.337000	0.181000	104.165000	539.000000	0.693000	24509.000000	0.476000	48.000000
75%	59.644000	0.208500	119.460500	591.000000	0.789000	28163.000000	0.540000	54.500000
max	72.698000	0.298000	137.918000	694.000000	0.938000	35816.000000	0.726000	70.000000

Check for duplicate

Return boolean Series denoting duplicate rows.

```
product_catalogue.duplicated()
```

```
32 False
33 False
34 False
35 True
36 False
37 True
38 True
39 True
40 True
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
24	Susu Nestle	2023-03-01	72.698	182	107.127	540.0	577	31682	384	50
25	Susu Anmum	2023-04-12	35.456	166	114.131	423.0	784	23154	578	53
26	Susu Anlene	2022-05-28	50.458	182	100.210	553.0	450	23803	649	22
27	Susu Ensure	2023-02-15	48.128	159	135.717	432.0	546	30498	293	69
28	Susu Pediasure	2023-06-15	65.328	114	102.538	503.0	819	28276	543	53
29	Susu Greenfields	2022-11-08	64.694	209	108.040	436.0	732	28200	568	43
30	Susu Aptamil	2023-02-09	51.549	120	137.663	567.0	792	16915	436	46
31	Susu Neocate	2022-05-17	53.782	118	73.045	557.0	732	24878	460	54
32	Susu Isomil	2022-07-30	41.122	223	74.590	479.0	786	21309	487	48
33	Susu Alula	2022-07-23	30.192	155	119.388	539.0	635	26399	470	29
34	Susu Bebelove	2023-02-28	46.521	203	76.538	390.0	597	24509	469	70
35	Susu Greenfields	2022-11-08	64.694	209	108.040	436.0	732	28200	568	43
36	Susu Aptamil	2023-02-09	51.549	2	137.663	567.0	792	45	460	46
37	Susu Neocate	2022-05-17	53.782	118	73.045	557.0	732	24878	460	54
38	Susu Isomil	2022-07-30	41.122	223	74.590	479.0	786	21309	487	48
39	Susu Alula	2022-07-23	30.192	155	119.388	539.0	635	26399	470	29

Drop Duplicates

Remove duplicated rows.

```
product_catalogue.drop_duplicates()
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
24	Susu Nestle	2023-03-01	72.698	182	107.127	540.0	577	31682	384	50
25	Susu Anmum	2023-04-12	35.456	166	114.131	423.0	784	23154	578	53
26	Susu Anlene	2022-05-28	50.458	182	100.210	553.0	450	23803	649	22
27	Susu Ensure	2023-02-15	48.128	159	135.717	432.0	546	30498	293	69
28	Susu Pediasure	2023-06-15	65.328	114	102.538	503.0	819	28276	543	53
29	Susu Greenfields	2022-11-08	64.694	209	108.040	436.0	732	28200	568	43
30	Susu Aptamil	2023-02-09	51.549	120	137.663	567.0	792	16915	436	46
31	Susu Neocate	2022-05-17	53.782	118	73.045	557.0	732	24878	460	54
32	Susu Isomil	2022-07-30	41.122	223	74.590	479.0	786	21309	487	48
33	Susu Alula	2022-07-23	30.192	155	119.388	539.0	635	26399	470	29
34	Susu Bebelove	2023-02-28	46.521	203	76.538	390.0	597	24509	469	70
36	Susu Aptamil	2023-02-09	51.549	2	137.663	567.0	792	45	460	46

Drop Irrelevant Data

To remove irrelevant rows or column from our data:

```
product_catalogue.drop()
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114.582	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102.580	458.0	744	20982	532	61
2	ABC	2023-04-11	59.787	200	122.788	NaN	340	21552	579	60

```
product_catalogue.drop(columns='Niche Market')
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Marketing
0	Indomie	2022-08-18	67.641	208	114.582	694.0	551	28407	33
1	Bango	2023-07-23	54.002	262	102.580	458.0	744	20982	61
2	ABC	2023-04-11	59.787	200	122.788	NaN	340	21552	60
3	Sari Roti	2022-07-19	72.409	181	75.303	692.0	764	22722	41

Data Cleaning

- Remove duplicates or unwanted observations
- **Fixing structural errors**
- Data transformation
- Handling missing data
- Managing unwanted outliers

Fixing Structural Error

Check for “typos” and mislabeled classes (mostly in categorical features)

Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Marketing
Indomie	2022-08-18	High	208	114.582	694.0	551	28407	Flyer
Bango	2023-07-23	Medium	262	102.580	458.0	744	20982	*Gogle Ads*
ABC	2023-04-11	Low	200	122.788	NaN	340	21552	Flyer
Sari Roti	2022-07-19	Meedium	181	75.303	692.0	764	22722	Text Ads

Fixing Structural Error

Check for “typos” and mislabeled classes (mostly in categorical features)

Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Marketing
Indomie	2022-08-18	High	208	114.582	694.0	551	28407	Flyer
Bango	2023-07-23	Medium	262	102.580	458.0	744	20982	Google Ads
ABC	2023-04-11	Low	200	122.788	NaN	340	21552	Flyer
Sari Roti	2022-07-19	Medium	181	75.303	692.0	764	22722	Text Ads

Define Data Shape and Type

Check information, shape, and data type for the data frame. You may use these columns to slice and perform analysis and categorize the data.

```
product_catalogue.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44 entries, 0 to 43
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Product Name     44 non-null     object
 1   Datetime         44 non-null     object
 2   Popularity       44 non-null     float64
 3   Profit Margin    44 non-null     int64
 4   Seasonal Demand  44 non-null     float64
 5   Units Sold       40 non-null     float64
 6   Availability     44 non-null     int64
 7   Price            44 non-null     int64
 8   Niche Market     44 non-null     int64
 9   Marketing        44 non-null     int64
dtypes: float64(3), int64(5), object(2)
memory usage: 3.6+ KB
```

Change Data Type

Convert column data type (to string, int, float, etc)

```
product_catalogue['Seasonal Demand'].astype('int')
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102	458.0	744	20982	532	61
2	ABC	2023-04-11	59.787	200	122	NaN	340	21552	579	60
3	Sari Roti	2022-07-19	72.409	181	75	692.0	764	22722	453	41

Data Cleaning

- Remove duplicates or unwanted observations
- Fixing structural errors
- **Handling missing data**
- Data transformation
- Managing unwanted outliers

Check Missing Values

Return dataframe of Boolean values which are True for NaN values.

```
product_catalogue.isnull()
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	True	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	True	False	False	False	False
6	False	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	True	False	False	False	False

Check Missing Values

Return dataframe of Boolean values which are True for NaN values.

```
product_catalogue.isnull()
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114.582	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102.580	458.0	744	20982	532	61
2	ABC	2023-04-11	59.787	200	122.788	NaN	340	21552	579	60
3	Sari Roti	2022-07-19	72.409	181	75.303	692.0	764	22722	453	41
4	Teh Pucuk	2023-07-19	68.676	185	108.047	648.0	938	25087	406	35
5	Pocari Sweat	2022-02-12	40.227	148	86.304	NaN	794	23230	459	55
6	Aqua	2022-01-24	59.501	129	87.676	590.0	609	18125	498	44
7	Mie Sedap	2022-11-21	48.486	115	88.423	413.0	812	90	538	51
8	Teh Botol	2023-04-08	48.968	298	93.769	691.0	568	13882	726	46
9	Kecap Cap Bango	2022-11-18	54.106	175	101.123	473.0	654	28126	496	56
10	Kecap ABC	2022-02-04	51.440	178	76.697	NaN	693	16989	404	56

Handling Missing Values

The most commonly recommended ways of handling missing data is either

- **Dropping the observations**
- **Imputing missing values based on other observations.**

Drop the Missing Values

Drop missing indices

```
product_catalogue.dropna()
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102	458.0	744	20982	532	61
3	Sari Roti	2022-07-19	72.409	181	75	692.0	764	22722	453	41
4	Teh Pucuk	2023-07-19	68.676	185	108	648.0	938	25087	406	35
6	Aqua	2022-01-24	59.501	129	87	590.0	609	18125	498	44
7	Mie Sedap	2022-11-21	48.486	115	88	413.0	812	90	538	51
8	Teh Botol	2023-04-08	48.968	298	93	691.0	568	13882	726	46
9	Kecap Cap Bango	2022-11-18	54.106	175	101	473.0	654	28126	496	56
11	Susu Ultra	2023-03-29	64.543	137	118	594.0	871	19478	465	42

Fill the Missing Values

Replace NaN values with some value

```
product_catalogue.fillna(0)
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102	458.0	744	20982	532	61
2	ABC	2023-04-11	59.787	200	122	0.0	340	21552	579	60
3	Sari Roti	2022-07-19	72.409	181	75	692.0	764	22722	453	41
4	Teh Pucuk	2023-07-19	68.676	185	108	648.0	938	25087	406	35
5	Pocari Sweat	2022-02-12	40.227	148	86	0.0	794	23230	459	55
6	Aqua	2022-01-24	59.501	129	87	590.0	609	18125	498	44
7	Mie Sedap	2022-11-21	48.486	115	88	413.0	812	90	538	51
8	Teh Botol	2023-04-08	48.968	298	93	691.0	568	13882	726	46
9	Kecap Cap Bango	2022-11-18	54.106	175	101	473.0	654	28126	496	56
10	Kecap ABC	2022-02-04	51.440	178	76	0.0	693	16989	404	56

Fill the Missing Values

Interpolate the missing values

```
product_catalogue.interpolate(method ='linear', limit_direction ='forward')
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102	458.0	744	20982	532	61
2	ABC	2023-04-11	59.787	200	122	575.0	340	21552	579	60
3	Sari Roti	2022-07-19	72.409	181	75	692.0	764	22722	453	41
4	Teh Pucuk	2023-07-19	68.676	185	108	648.0	938	25087	406	35
5	Pocari Sweat	2022-02-12	40.227	148	86	619.0	794	23230	459	55
6	Aqua	2022-01-24	59.501	129	87	590.0	609	18125	498	44
7	Mie Sedap	2022-11-21	48.486	115	88	413.0	812	90	538	51
8	Teh Botol	2023-04-08	48.968	298	93	691.0	568	13882	726	46
9	Kecap Cap Bango	2022-11-18	54.106	175	101	473.0	654	28126	496	56
10	Kecap ABC	2022-02-04	51.440	178	76	533.5	693	16989	404	56

Data Cleaning

- Remove duplicates or unwanted observations
- Fixing structural errors
- **Handling missing data**
- **Data transformation**
- Managing unwanted outliers

Data Transformation

Converting the data from one form to another to make it more suitable for analysis.
Data transformation includes:

Scaling

Transforming the values of features to a specific range

Standardization

Transforms the values to have a mean of 0 and a standard deviation of 1.

$$Z = (X - \mu) / \sigma$$

Scaling

Scaling is particularly useful when features have different scales, and certain algorithms are sensitive to the magnitude of the features.

```
from sklearn.preprocessing import MinMaxSca

# initialising the MinMaxScaler
scaler = MinMaxScaler(feature_range=0, 1))

product_catalogue['Profit Margin'] =
scaler.fit_transform(product_catalogue
['Profit Margin'].values.reshape(-1, 1))
```

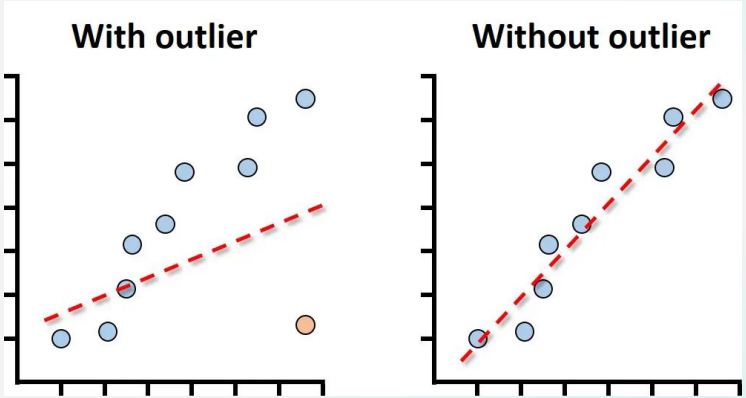
	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand
0	Indomie	2022-08-18	67.641	0.695946	114
1	Bango	2023-07-23	54.002	0.878378	102
2	ABC	2023-04-11	59.787	0.668919	122
3	Sari Roti	2022-07-19	72.409	0.604730	75
4	Teh Pucuk	2023-07-19	68.676	0.618243	108
5	Pocari Sweat	2022-02-12	40.227	0.493243	86
6	Aqua	2022-01-24	59.501	0.429054	87
7	Mie Sedap	2022-11-21	48.486	0.381757	88
8	Teh Botol	2023-04-08	48.968	1.000000	93
9	Kecap Cap Bango	2022-11-18	54.106	0.584459	101
10	Kecap ABC	2022-02-04	51.440	0.594595	76

Data Cleaning

- Remove duplicates or unwanted observations
- Fixing structural errors
- Handling missing data
- Data transformation
- **Managing unwanted outliers**

Outliers Detection

Outlier : data-item that deviates significantly from the rest of the (so-called normal) objects.



Ways to detect outliers:

- Outliers visualization (boxplot, scatter plot, etc)
- Z- Score (standard score)

To understand how far is the data point from the mean

- IQR (Inter Quartile Range)

The most trusted approach used in the research field.

Detecting outliers using IQR

IQR = Quartile 3 - Quartile 1

```
Q1 = np.percentile(product_catalogue['Price'], 25, method='midpoint')
Q3 = np.percentile(product_catalogue['Price'], 75, method='midpoint')
IQR = Q3 - Q1
```

IQR: 6857.5

To define the outlier base value: above and below dataset's normal range

*Upper = Q3 +1.5*IQR*

*Lower = Q1 - 1.5*IQR*

Upper Bound: 38449.25
Above upper bound: 0
Lower Bound: 11019.25
Below lower bound: 2 entries

Units Sold	Availability	Price
694.0	551	28407
458.0	744	20982
NaN	340	21552
692.0	764	22722
648.0	938	25087
NaN	794	23230
590.0	609	18125
413.0	812	90
691.0	568	13882
473.0	654	28126
NaN	693	16989
594.0	871	19478
484.0	626	25260
561.0	617	21302

Removing Outliers

Removing rows of the detected outliers from the dataset using its index

```
# Create arrays of Boolean values indicating the outlier rows
lower_array = np.where(product_catalogue['Price']<=lower)[0]

# Removing the outliers
product_catalogue.drop(index=lower_array, inplace=True)
```

Price
28407
20982
21552
22722
25087
23230
18125
90
13882
28126

Price
28407
20982
21552
22722
25087
23230
18125
13882
28126

pandas-ai

Using generative AI models from OpenAI, **pandas-ai** is a pandas library addition, only need simply a text prompt to analyse

```
!pip install -q pandasai
```

head() function

```
pandas_ai(product_catalogue, "Show the first 5 rows of data in tabular form")
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114.582	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102.580	458.0	744	20982	532	61
2	ABC	2023-04-11	59.787	200	122.788	NaN	340	21552	579	60
3	Sari Roti	2022-07-19	72.409	181	75.303	692.0	764	22722	453	41
4	Teh Pucuk	2023-07-19	68.676	185	108.047	648.0	938	25087	406	35

Data Cleaning

Remove duplicates or unwanted observations

```
pandas_ai(product_catalogue, "Are there any duplicate rows? ")
```

```
'There are duplicate rows.'
```

```
pandas_ai(product_catalogue, "Show the duplicated rows of the data? ")
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
35	Susu Greenfields	2022-11-08	64.694	209	108.040	436.0	732	28200	568	43
37	Susu Neocate	2022-05-17	53.782	118	73.045	557.0	732	24878	460	54
38	Susu Isomil	2022-07-30	41.122	223	74.590	479.0	786	21309	487	48
39	Susu Alula	2022-07-23	30.192	155	119.388	539.0	635	26399	470	29
40	Susu Bebelove	2023-02-28	46.521	203	76.538	390.0	597	24509	469	70
41	Susu Enfamil	2022-03-22	56.536	203	124.449	456.0	893	28858	451	55
42	Susu Frisian Flag	2023-06-29	64.941	219	123.576	390.0	813	26335	516	44
43	Susu Bear Brand	2022-12-18	47.948	174	96.402	529.0	592	24803	523	44

Data Cleaning

Handling missing data

```
pandas_ai(product_catalogue, "Are there any missing values?")
```

4

```
pandas_ai(product_catalogue, "Drop the row with missing values with inplace=True
and return True when done else False ")
```

	Product Name	Datetime	Popularity	Profit Margin	Seasonal Demand	Units Sold	Availability	Price	Niche Market	Marketing
0	Indomie	2022-08-18	67.641	208	114.582	694.0	551	28407	591	33
1	Bango	2023-07-23	54.002	262	102.580	458.0	744	20982	532	61
3	Sari Roti	2022-07-19	72.409	181	75.303	692.0	764	22722	453	41
4	Teh Pucuk	2023-07-19	68.676	185	108.047	648.0	938	25087	406	35
6	Aqua	2022-01-24	59.501	129	87.676	590.0	609	18125	498	44
7	Mie Sedap	2022-11-21	48.486	115	88.423	413.0	812	90	538	51
8	Teh Botol	2023-04-08	48.968	298	93.769	691.0	568	13882	726	46
9	Kecap Cap Bango	2022-11-18	54.106	175	101.123	473.0	654	28126	496	56
11	Susu Ultra	2023-03-29	64.543	137	118.017	594.0	871	19478	465	42

Data Cleaning

Detecting outliers

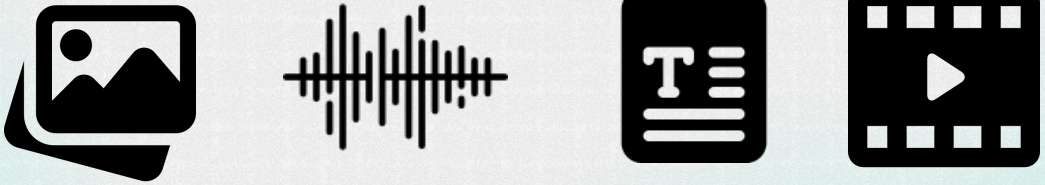
```
pandas_ai(product_catalogue, "Plot a scatterplot for the column 'Price'")
```

02

Unstructured Data Preprocessing

Unstructured Data

Unstructured data is qualitative data and unstructured data pattern is not easily searchable



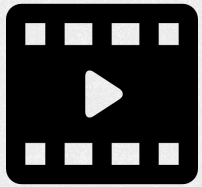
Unstructured Data

Application of AI in unstructured data

- Image classification
- Image segmentation
- Image captioning
- Text-to-image

- Voice cloning
- Text-to-speech
- Speech recognition
- Music classification

- Text summarisation
- Chat-bot
- Sentiment analysis
- Document classification



- Video captioning
- Face-reenactment
- DeepFake video
- Text-to-video

Data Preprocessing

Data preprocessing is the process of transforming the ***raw data*** to a state, amount, structure, and format that the machine learning model can ***digest***

Data preprocessing is an essential before going into the model and depending on how well the data has been preprocessed; the results are seen.

Text Pre-processing

```
!pip install nltk
```

- **Tokenization** Splitting the sentence into words

```
sentence = "The grapes are in the fridge"

words = nltk.word_tokenize(sentence)
```

Output: `['The', 'grapes', 'are', 'in', 'the', 'fridge']`

Text Pre-processing

- **Normalization** Standardizing the text, such as converting all words to lowercase and removing punctuation

```
sentence.lower()
```

- **Stop words removal** Remove words that do not really signify any importance (a, an, the, is, etc.

```
from nltk.corpus import stopwords
sentence = "Generative AI is cool but can be dangerous, too!"
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(sentence)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
Output: ['Generative', 'AI', 'cool', 'dangerous', ',', '!']
```

Text Pre-processing

- **Stemming** Transforming a word to its root form

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
sentence = "We are discussing about machine learning in medical imaging"
```

Output:

```
we
are
discuss
about
machin
learn
in
medic
imag
```

Text Pre-processing

- **Lemmatization**

  Reduces the words to a word existing in the language.

```
# Define the POS: Part of Speech in each words
lemmatized_output = [lemmatizer.lemmatize(w, get_wordnet_pos(w)) for w in nltk.word_tokenize(sentence)]
```

```
sentence = "We are discussing about machine learning in medical imaging"
```

Output:

```
['We', 'be', 'discuss', 'about', 'machine', 'learn', 'in', 'medical', 'image']
```

Audio Pre-processing

- ## Resampling the audio data

In audio data, if there is a discrepancy between sampling rate expected by a model you plan to train/inference. Resample the audio to the model's expected sampling rate.

```
from datasets import Audio
from datasets import load_dataset

minds = load_dataset("PolyAI/minds14", name="en-AU", split="train")
minds = minds.cast_column("audio", Audio(sampling_rate=16_000))
```

Audio Pre-processing

- **Filtering the dataset**

  Depends on the model training/inference requirement, we might need to filter the dataset.

- **Feature extractor**

  Preparing the required input features in the right format for model training

Audio input after preprocessing as log-mel spectrograms

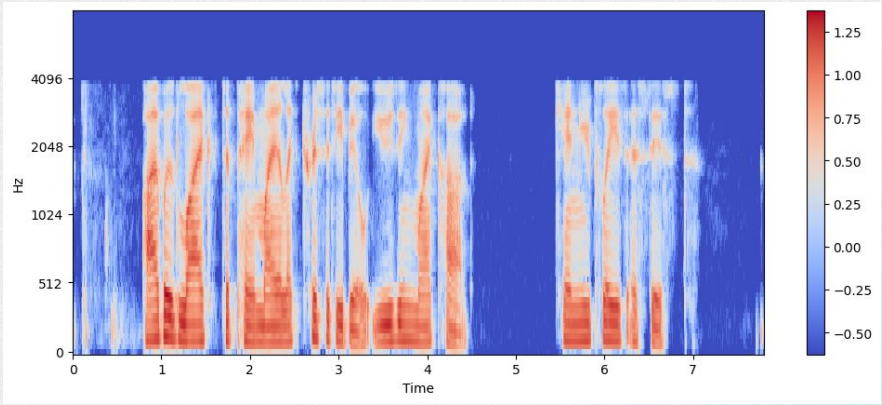


Image Pre-processing

- **Image resizing**

Mostly, the ML models require all input images to be the same size (for example, 256x256 or 512x512 pixels)

```
!pip install --u pillow
from PIL import Image
image =
Image.open('/content/nyoo.png')
new_image = image.resize((256,256))
```

(453, 453)

(256, 256)

Image Pre-processing

- **Image transformation**

  Common image transformations (resizing, intensity normalization, rotating, cropping, flipping, etc.) with pyTorch (torch-vision) before Deep Learning model training/inference.

```
import torchvision.transforms as T
preprocess = T.Compose([
    T.Resize(256),
    T.CenterCrop(224),
    T.ToTensor(),
    )
])
```

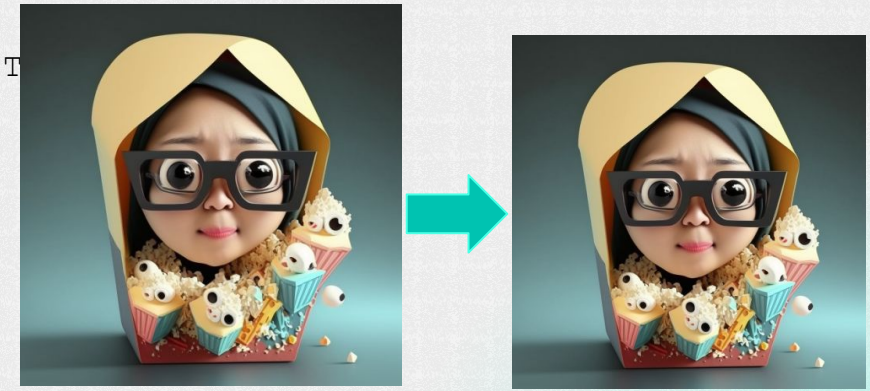


Image Pre-processing

- **Fixing inconsistent image**

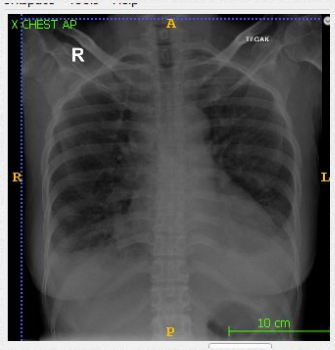
  Inconsistent data points often tend to disturb the model's overall learning, leading to false predictions.

  Mitigation:

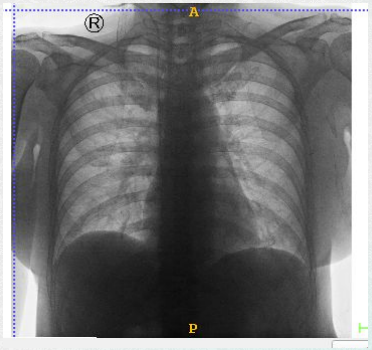
  - Remove inconsistent images
  - If known, convert into standard protocol

(i.e, invert the image intensity of Chest X-Ray)

Normal Chest X-Rays with inconsistencies among samples



Standard protocol



Overexposed

03

Importance of Data quality

Study Case

Output for the product name that will achieve the optimum revenue for shop "Ratu Susu" will be **different** using the **raw** data and **cleaned** data.

```
# Modelling with different df (raw and cleaned data)
df['Revenue'] = df['Units Sold'] * df['Price']
total_revenue = df.groupby('Product Name')['Revenue'].sum()
optimum_product = total_revenue.idxmax()
```

Optimum product with ***raw data*** as input

Optimum product with ***cleaned data*** as input

Importance of Data Cleaning

- Model performance depends on the quality of data
- Avoiding Mistakes
- Improve the reliability of data analysis
- ML algorithms have their own requirements
- Avoiding unnecessary costs

Thanks

IG: @belajar.machinelearning

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, infographics & images by **Freepik** and content by **Swetha Tandri**