

Task 3 – Modelling

Setup:

First, we need to mount this notebook to our Google Drive folder, in order to access the CSV data file. If you haven't already, watch this video <https://www.youtube.com/watch?v=woHxvbBLarQ> to help you mount your Google Drive folder.

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

We want to use dataframes once again to store and manipulate the data.

```
!pip install pandas
```

Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.3.5)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (1.21.6)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)

```
import pandas as pd
```

Data Loading:

```
path = "/content/drive/MyDrive/Forage - Cognizant AI Program/Task 3/Resources/"

sales_df = pd.read_csv(f"{path}sales.csv")
sales_df.drop(columns=["Unnamed: 0"], inplace=True, errors="ignore")
sales_df.head()
```

	transaction_id	timestamp	product_id	category	customer_type	unit_price	quantity	total	payment_type
0	a1c82654-c52c-45b3-8ce8-4c2a1efee3ed	2022-03-02 09:51:38	3bc6c1ea-0198-46de-9ffd-514aa3328713	fruit	gold	3.99	2	7.98	e-wallet
1	931ad550-09e8-4da6-beaa-8c9d17be9c60	2022-03-06 10:33:59	ad81b46c-bf38-41cf-9b54-5fe7f5eba93e	fruit	standard	3.99	1	3.99	e-wallet
2	ae133534-6f61-4cd6-b6b8-d1c1d8d90aea	2022-03-04 17:20:21	7c55cbd4-f306-4c04-a030-628cbe7867c1	fruit	premium	0.19	2	0.38	e-wallet
3	157cebd9-aafe-475d-8a11-7c8e0f5b76e4	2022-03-02 17:23:58	8bd4a348-1707-403f-8be7-9eddecc883	fruit	gold	0.19	4	0.76	e-wallet
4	a81a6cd3-5ebc-44a2-b26c-aea43e46c514	2022-03-05 14:32:43	7f5e86e6-f0ef-45f6-bf44-27b095c9ad1d	fruit	basic	4.49	2	8.98	debit card

```
stock_df = pd.read_csv(f"{path}sensor_stock_levels.csv")
stock_df.drop(columns=["Unnamed: 0"], inplace=True, errors="ignore")
stock_df.head()
```

	id	timestamp	product_id	estimated_stock_pct
0	422be505-c247-47bd-9831-e0ef87a44d9a	2022-03-07 12:11:02	f658605e-75f3-4fed-a655-c99a3f44427	0.75
1	f2612b26-fc82-49ea-8940-0751fdd4d9ef	2022-03-07 16:39:46	de06083a-f5c0-451d-b2f4-9ab8bb5260d	0.48
2	989a287f-67ee-4478-aa49-ca3a5dace2e	2022-03-01 18:17:43	ce8f3a04-d1a4-43b1-a7c2-fa1b8e7674c8	0.58
3	a78e5683-d247-46ac-9909-1a77bdebe7b2	2022-03-02 14:29:09	c21e3ba9-92a3-4745-92c2-6faef73223f7	0.79
4	08a32247-3f44-4002-85fb-c198a34dd4b2	2022-03-02 13:46:18	7f478817-aa5b-44e9-9059-8845228c9eb0	0.22

```
temp_df = pd.read_csv(f"{path}sensor_storage_temperature.csv")
temp_df.drop(columns=["Unnamed: 0"], inplace=True, errors="ignore")
temp_df.head()
```

	id	timestamp	temperature
0	d1ca1ef8-0eac-42fc-af80-97106efc7b13	2022-03-07 15:55:20	2.96
1	4bba5ec4-0f3a-4f16-826f-8cf9397e9d1a	2022-03-01 09:18:22	1.88
2	3dd7ae7-1e72-4512-812f-b0b5d8a28cf3	2022-03-04 15:12:26	1.78
3	9500357b-c015-424a-817a-7d77b38ef471	2022-03-02 12:30:42	2.18
4	c4b61fec-99c2-4ced-8e5d-4eadd8c9632fa	2022-03-05 09:09:33	1.38

Data Cleaning:

Now that we have our 3 datasets successfully loaded, we need to ensure that the data is clean. Data cleaning can be a very intense task, so for this exercise, we will focus just on ensuring that the correct datatypes are present for each column, and if not, correcting them.

Merge data

Feature Engineering