# Task 1 – 20MIP10033 – Chandan Thota

**Data Preprocessing Stage 1**

1. We are going to load the train dataset, check the shape and data types present in it
2. We are going to set the "date" column as index for this dataset
3. We are checking if any nulls are present in the dataset are not
4. We are going to check the percentage of nulls if present in the dataset

| | Column | Null Count | Null Percentage |
|---|---|---|---|
| **0** | ID | 0 | 0.000 |
| **1** | Item Id | 2 | 0.002 |
| **2** | Item Name | 1832 | 1.805 |
| **3** | ad_spend | 24187 | 23.832 |
| **4** | anarix_id | 0 | 0.000 |
| **5** | units | 17898 | 17.635 |
| **6** | unit_price | 0 | 0.000 |

5. We can see that the nulls exist in columns "Item Id", "Item Name", "ad_spend" and "units". So are going to 1st handle the nulls present in "Item Id", "Item Name" and "ad_spend". We would later look into "units" since it is our target variable
6. We are going to remove the 2 null rows present in the column "Item Id", since the whole row as nulls values.
7. We are going to fill the column "Item Name" by mapping the nulls with column "Item Id", since there is one to one mapping present between them. We are going 1st sort the dataset and then apply forward imputing.

| | Column | Null Count | Null Percentage |
|---|---|---|---|
| **0** | ID | 0 | 0.000 |
| **1** | Item Id | 0 | 0.000 |
| **2** | Item Name | 3 | 0.003 |
| **3** | ad_spend | 24187 | 23.832 |
| **4** | anarix_id | 0 | 0.000 |
| **5** | units | 17896 | 17.634 |
| **6** | unit_price | 0 | 0.000 |

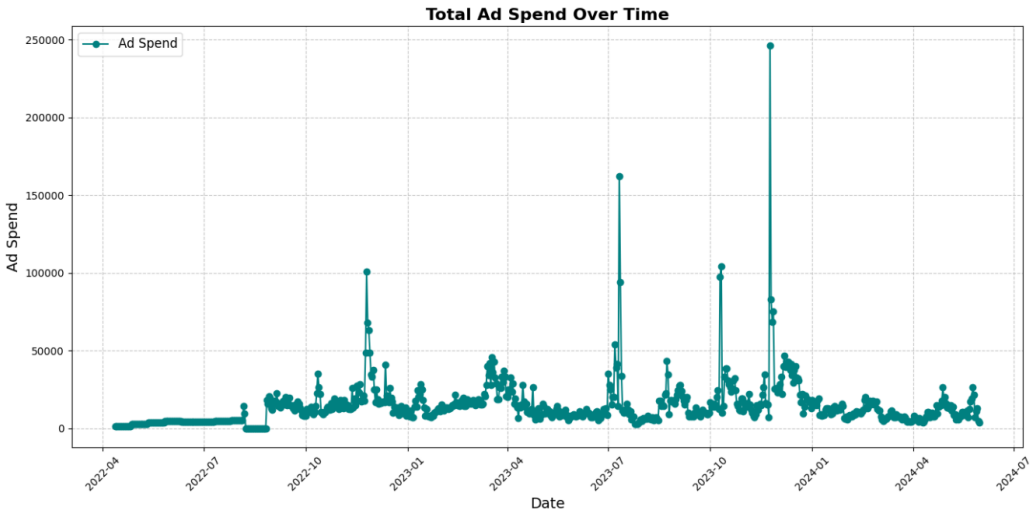
8. We still have 3 nulls present which are off "ASIN_BLANK". Removing this rows from the dataset

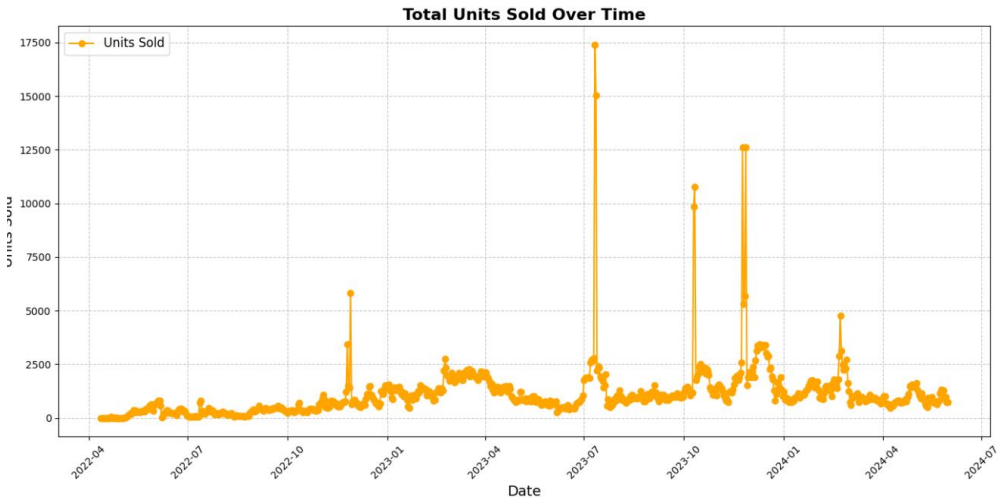
| date | ID | Item Id | Item Name | ad_spend | anarix_id | units | unit_price |
|---|---|---|---|---|---|---|---|
| **2023-09-25** | 2023-09-25_ASIN_BLANK | ASIN_BLANK | NaN | 0.0 | NAPQUEEN | NaN | 0.0 |
| **2023-10-10** | 2023-10-10_ASIN_BLANK | ASIN_BLANK | NaN | 0.0 | NAPQUEEN | NaN | 0.0 |
| **2023-11-02** | 2023-11-02_ASIN_BLANK | ASIN_BLANK | NaN | 0.0 | NAPQUEEN | NaN | 0.0 |

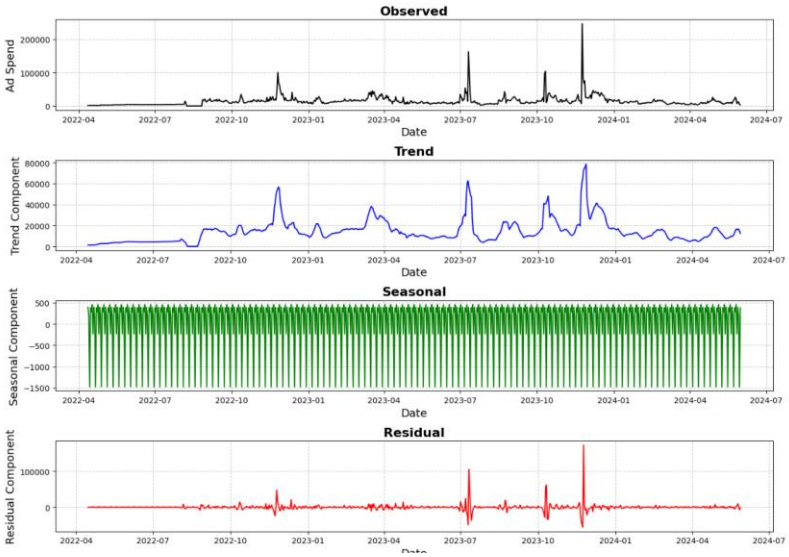
9. We are now going to fill the nulls present in column “ad_spend” using forward and backward imputing.

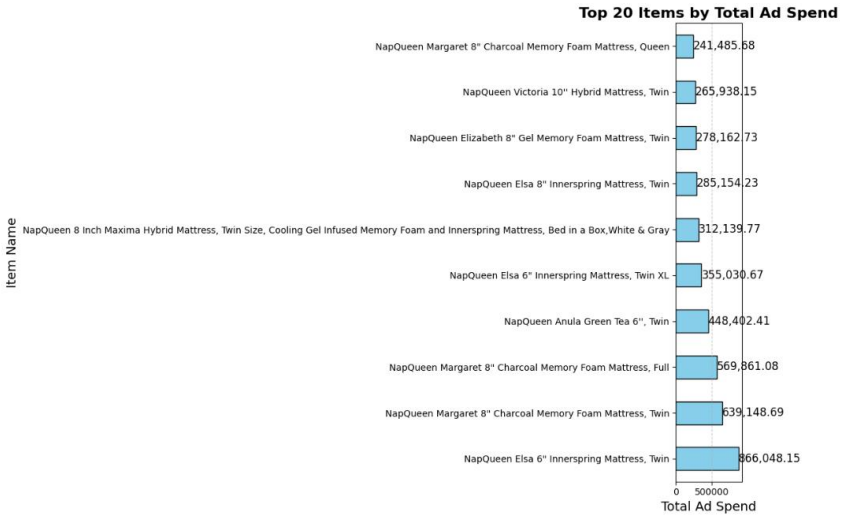
| | Column | Null Count | Null Percentage |
|---|---|---|---|
| **0** | ID | 0 | 0.000 |
| **1** | Item Id | 0 | 0.000 |
| **2** | Item Name | 0 | 0.000 |
| **3** | ad_spend | 0 | 0.000 |
| **4** | anarix_id | 0 | 0.000 |
| **5** | units | 17893 | 17.631 |
| **6** | unit_price | 0 | 0.000 |

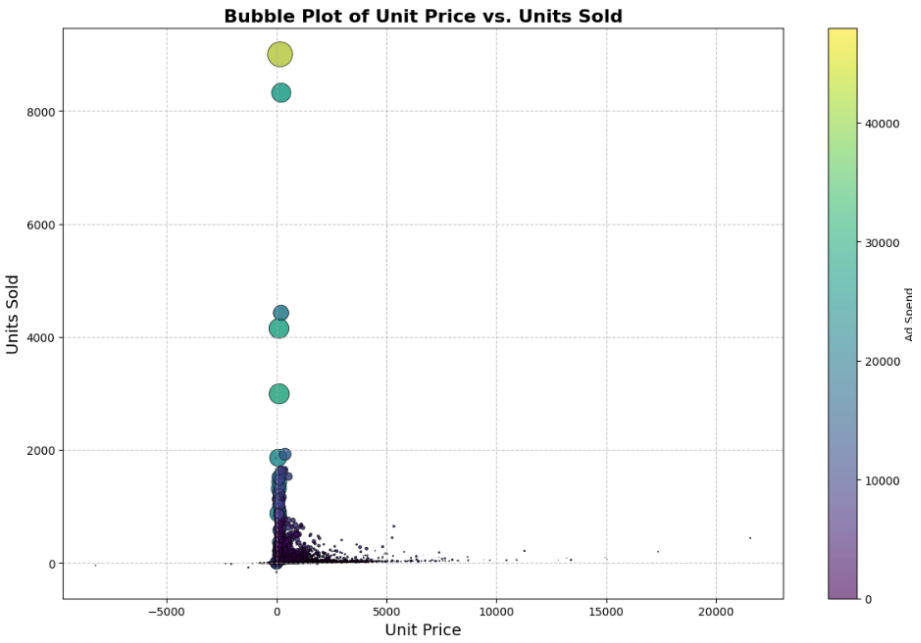
## EDA



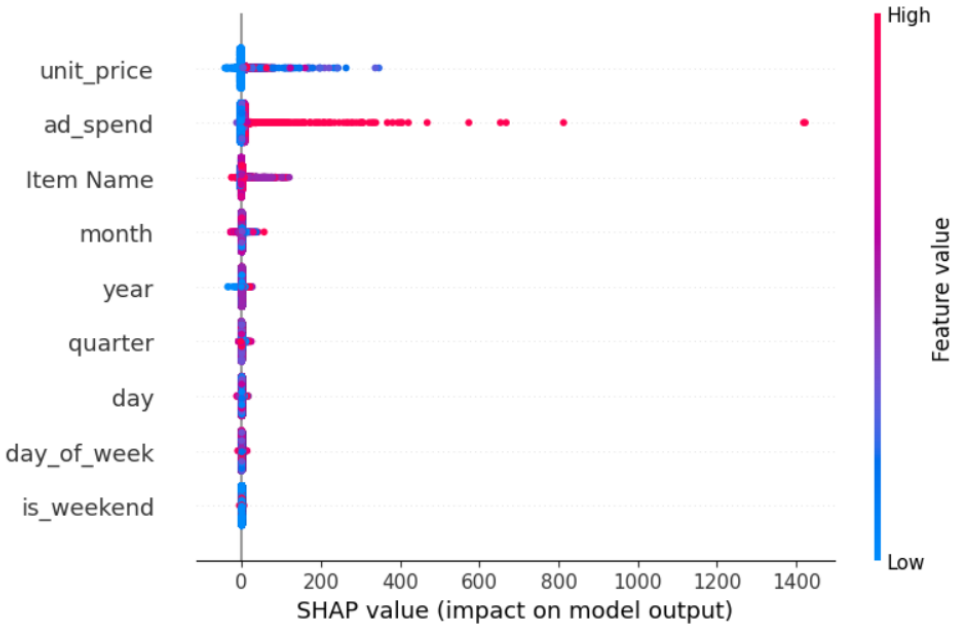

Observed
Ad Spend
Date
Trend
Trend Component
Date
Seasonal
Seasonal Component
Date
Residual
Residual Component

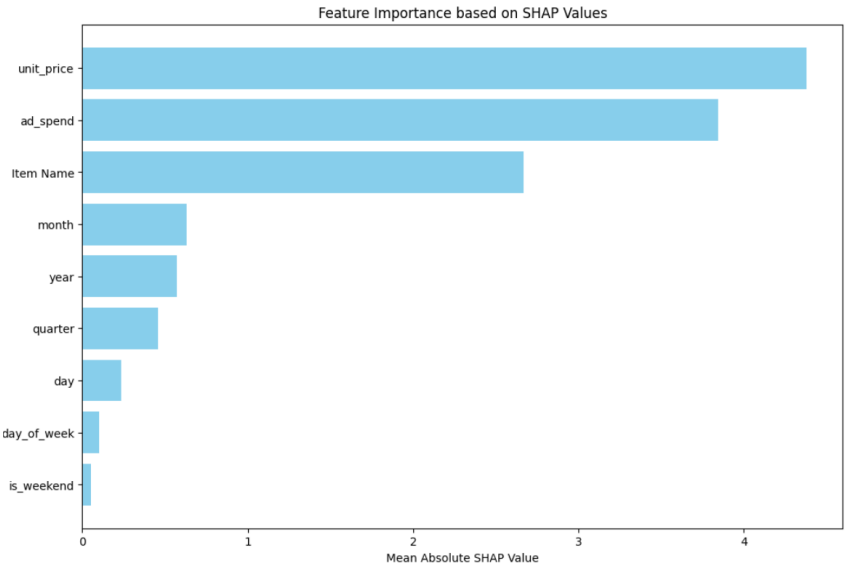

Top 20 Items by Total Ad Spend
NapQueen Margaret 8" Charcoal Memory Foam Mattress, Queen
241,485.68
NapQueen Victoria 10" Hybrid Mattress, Twin
265,938.15
NapQueen Elizabeth 8" Gel Memory Foam Mattress, Twin
278,162.73
NapQueen Elsa 8" Innerspring Mattress, Twin
285,154.23
NapQueen 8 Inch Maxima Hybrid Mattress, Twin Size, Cooling Gel Infused Memory Foam and Innerspring Mattress, Bed in a Box,White & Gray
312,139.77
NapQueen Elsa 6" Innerspring Mattress, Twin XL
355,030.67
NapQueen Anula Green Tea 6", Twin
448,402.41
NapQueen Margaret 8" Charcoal Memory Foam Mattress, Full
569,861.08
NapQueen Margaret 8" Charcoal Memory Foam Mattress, Twin
639,148.69
NapQueen Elsa 6" Innerspring Mattress, Twin
866,048.15
Item Name
Total Ad Spend


Bubble Plot of Unit Price vs. Units Sold
Units Sold
Unit Price
Ad Spend

**Feature Engineering**

10. We are going to create 2 dataframes one contains the values of column “units” and another with null values in column “units”. [known_df & unknown_df]
11. We are going to create few more features using the column “date” like “year”, “month”, “day”, “day_of_week”, “is_weekend”, “quarter”.
12. Now we are going to drop the unnecessary columns like 'ID', 'units', 'Item Id' and 'anarix_id'.
13. We are going to split the known_df into train and val datasets
14. We are going to train a ensemble learning model called random forest.
15. We are going to use the SHAP [SHapley Additive exPlanations] library to get the feature importance




16. We can see that the columns “unit_price”, “ad_spend”, “Item Name” and “month” have more importance in predicting the units.

**Data Preprocessing Stage 2**

17. Now we are going to predict the nulls in the column “nulls” using the trained random forest model [predictive imputing]
18. We are going to add these null values to the original train dataset

39]:

| | Column | Null Count | Null Percentage |
|---|---|---|---|
| 0 | ID | 0 | 0.0 |
| 1 | Item Id | 0 | 0.0 |
| 2 | Item Name | 0 | 0.0 |
| 3 | ad_spend | 0 | 0.0 |
| 4 | anarix_id | 0 | 0.0 |
| 5 | units | 0 | 0.0 |
| 6 | unit_price | 0 | 0.0 |

19. Now we will convert the categorical columns to numeric values using one hot encoding
20. We will also normalization the numerical columns present using standard scaler
21. We are going do the above operations on both train and test dataset and also set the column “date” as index.
22. We are also going to do the memory optimization to decrease the size of the dataframe for fast training of models.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 101485 entries, 2022-04-12 to 2024-05-3
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Item Name    101485 non-null  int32
 1   ad_spend     101485 non-null  float64
 2   unit_price   101485 non-null  float64
 3   year         101485 non-null  int32
 4   month        101485 non-null  int32
 5   day          101485 non-null  int64
 6   day_of_week  101485 non-null  int32
 7   is_weekend   101485 non-null  int64
 8   quarter      101485 non-null  int32
 9   units        101485 non-null  float64
dtypes: float64(3), int32(5), int64(2)
memory usage: 6.6 MB
None
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2833 entries, 2024-07-01 to 2024-07-28
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Item Name    2833 non-null   int32
 1   ad_spend     1382 non-null   float64
 2   unit_price   2833 non-null   float64
 3   year         2833 non-null   int32
 4   month        2833 non-null   int32
 5   day          2833 non-null   int64
 6   day_of_week  2833 non-null   int32
 7   is_weekend   2833 non-null   int64
 8   quarter      2833 non-null   int32
dtypes: float64(2), int32(5), int64(2)
memory usage: 166.0 KB
None
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 101485 entries, 2022-04-12 to 2024-05-31
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Item Name    101485 non-null  int32
 1   ad_spend     101485 non-null  float32
 2   unit_price   101485 non-null  float32
 3   year         101485 non-null  int32
 4   month        101485 non-null  int32
 5   day          101485 non-null  int8
 6   day_of_week  101485 non-null  int32
 7   is_weekend   101485 non-null  int8
 8   quarter      101485 non-null  int32
 9   units        101485 non-null  float32
dtypes: float32(3), int32(5), int8(2)
memory usage: 4.1 MB
None
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2833 entries, 2024-07-01 to 2024-07-28
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Item Name    2833 non-null   int32
 1   ad_spend     1382 non-null   float32
 2   unit_price   2833 non-null   float32
 3   year         2833 non-null   int32
 4   month        2833 non-null   int32
 5   day          2833 non-null   int8
 6   day_of_week  2833 non-null   int32
 7   is_weekend   2833 non-null   int8
 8   quarter      2833 non-null   int32
dtypes: float32(2), int32(5), int8(2)
memory usage: 105.1 KB
None
```

**Model Training**

23. We are going build and test 3 models which are as follows: -
    - Light Gradient-Boosting Machine
    - XGBoost
    - Long Short-Term Memory
24. We are train these models using the train data which is split into 80%

25. Later the model is validated using the val data which is 20% using MSE
26. So, the results are as follows: -

| | Model | MSE |
|---|---|---|
| 0 | LGBM | 3476.089708 |
| 1 | XGBoost | 3227.409424 |
| 2 | LSTM | 6162.423340 |

27. From the results we can clearly see that XGBoost model are more accuracy than other models.
28. So XGBoost model is used to predict the unseen test data
29. Later the predictions from the best model are saved to a .csv file.
30. GitHub Link: - NapQueen--Anarix--VIT-Assignment