Title: Sales Forecasting for Small Basket

Ayan Maity
Batch - 86
Student ID - 3845

Problem Statement

Small Basket is a huge online / mobile application-based grocery retailer in India, founded in 2011. Small Basket is trying to manage its supply chain and delivery partners and would like to accurately forecast the sales for the period starting from January of 2019 to August of 2019. You are also given a few features that were developed by the Business Intelligence team, that might or might not provide additional insights to your analysis.

Problem objective

Our objective is to create a model that can accurately forecast the sales for small basket across several locations and product categories.

Why Sales Forecasting ?

- Improve on shelf availability. Avoid Backorder.
- If the product is out of stock then it leads to loss.
- There is also moderate amount of products that are perishable, their quality decline over time and expire after a certain date.
- There is huge competition. To keep up with the fluctuating customer demand we need robust supply chain management.

Dataset Overview

Dataset	Features
train.csv	Date, locationId, item_id, unit_sales, onpromotion
train_transactions.csv	Date, location_identifier, transactions
items.csv	Item_id, category_of_items, class, perishable
locations.csv	Location_id, city, state, type, cluster
test.csv	Id, date, locationId, item_id, onpromotion

Classifying the problem statement.

- Forecast no of sales for a given item for the dates given in test.csv.
- We are looking at a Time Series data at an item and location level.

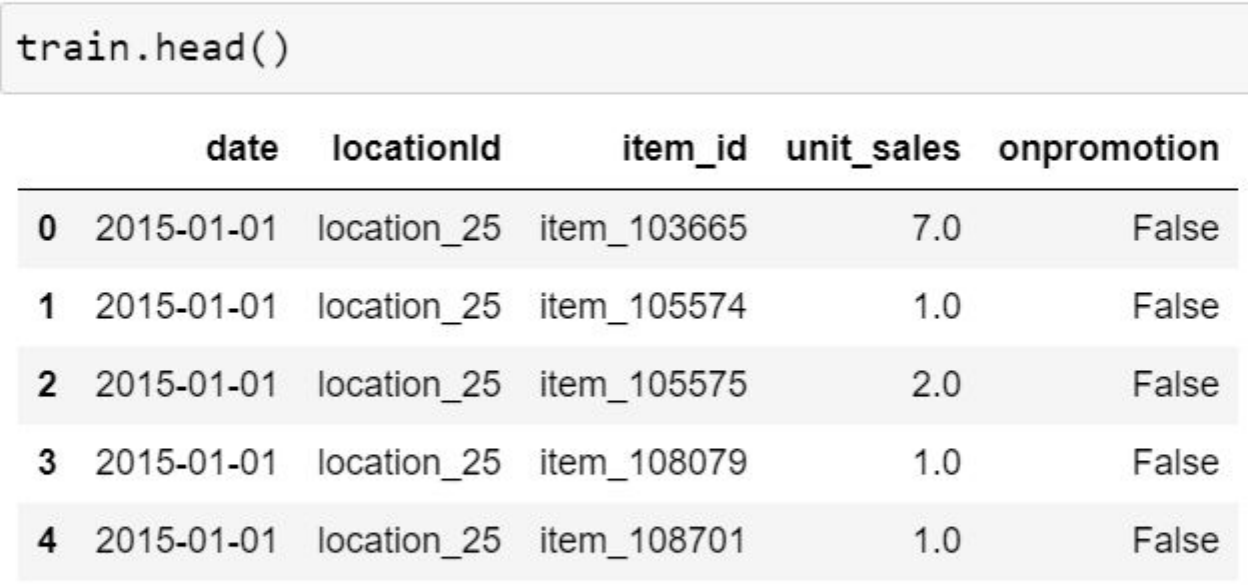
Error Metric

- MAPE (Mean Absolute Percentage Error)

$$\frac{1}{N}\sum_{t=1}^{N}\frac{ABS(Actual_t - Forecast_t)}{Actual_t} * 100\%$$

Exploratory Data Analysis.

train.csv



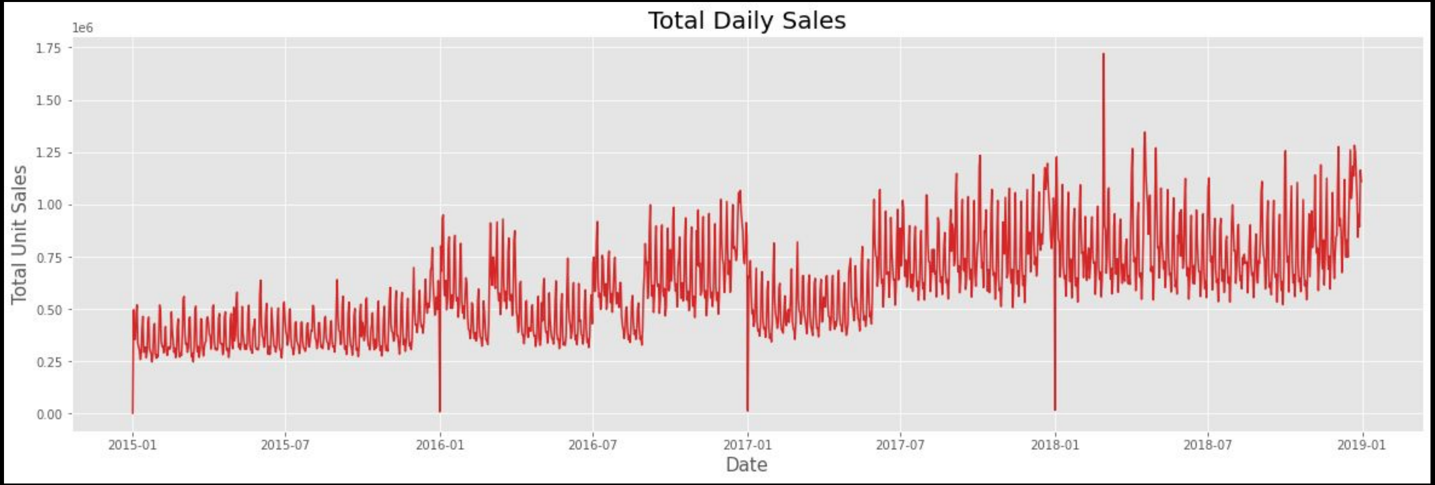
```
train.head()
```

	date	locationId	item_id	unit_sales	onpromotion
0	2015-01-01	location_25	item_103665	7.0	False
1	2015-01-01	location_25	item_105574	1.0	False
2	2015-01-01	location_25	item_105575	2.0	False
3	2015-01-01	location_25	item_108079	1.0	False
4	2015-01-01	location_25	item_108701	1.0	False

- Training data includes unit sales by date, location_id and item_id. An additional onpromotion column is there to indicate that the given product sold was in a promotion or discount.

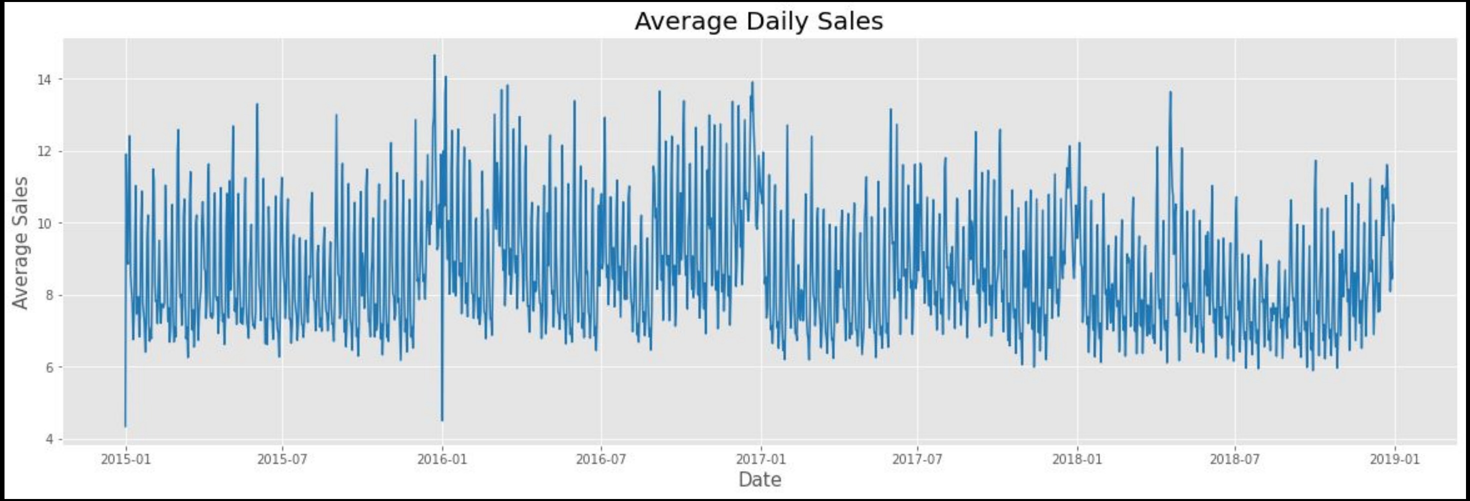
train.csv

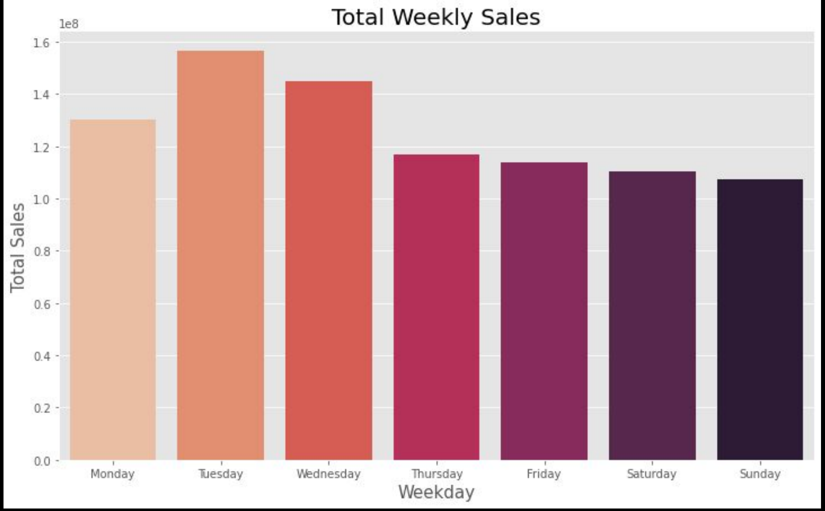
- There are 101688779 rows and 5 columns in the train data.
- The target feature (unit_sales) can be integer (2 bottles of soft drink) or float (200g flour)
- There are negative values in the target feature, which may indicate return of that particular item.
- There are missing values in the onpromotion column.
- We have data for past 4 years starting from 01/01/2015 - 31/12/2018.



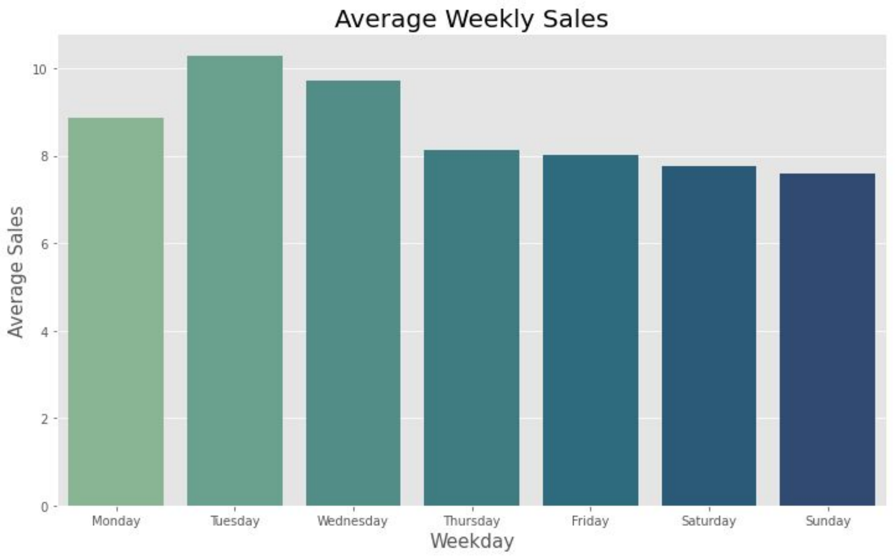
Total Sales over time (daily).

Average Sales over time (Daily)



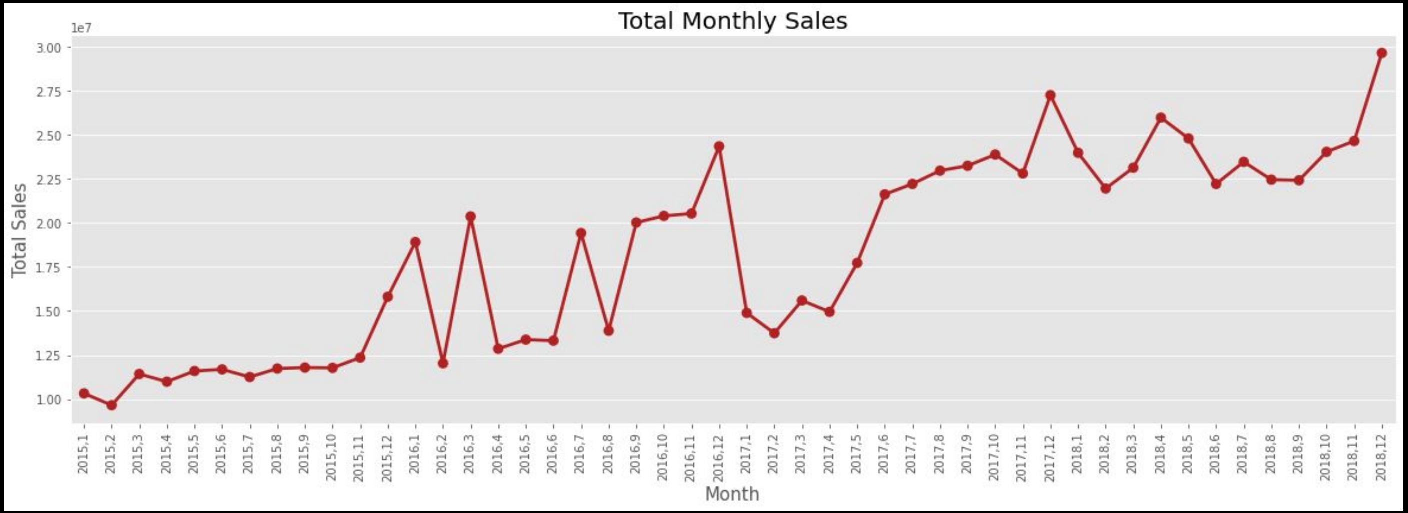


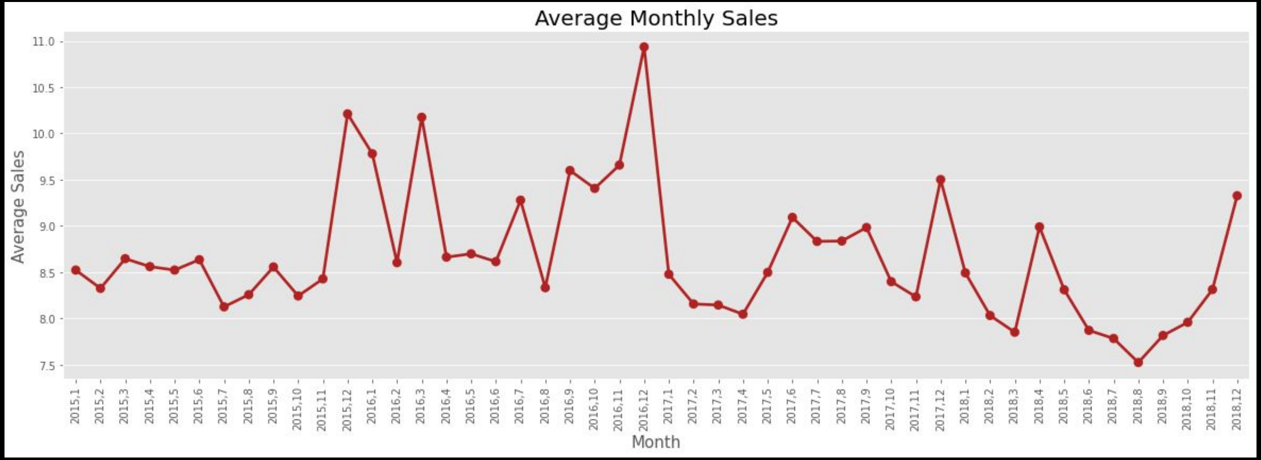
Average sales weekday basis.



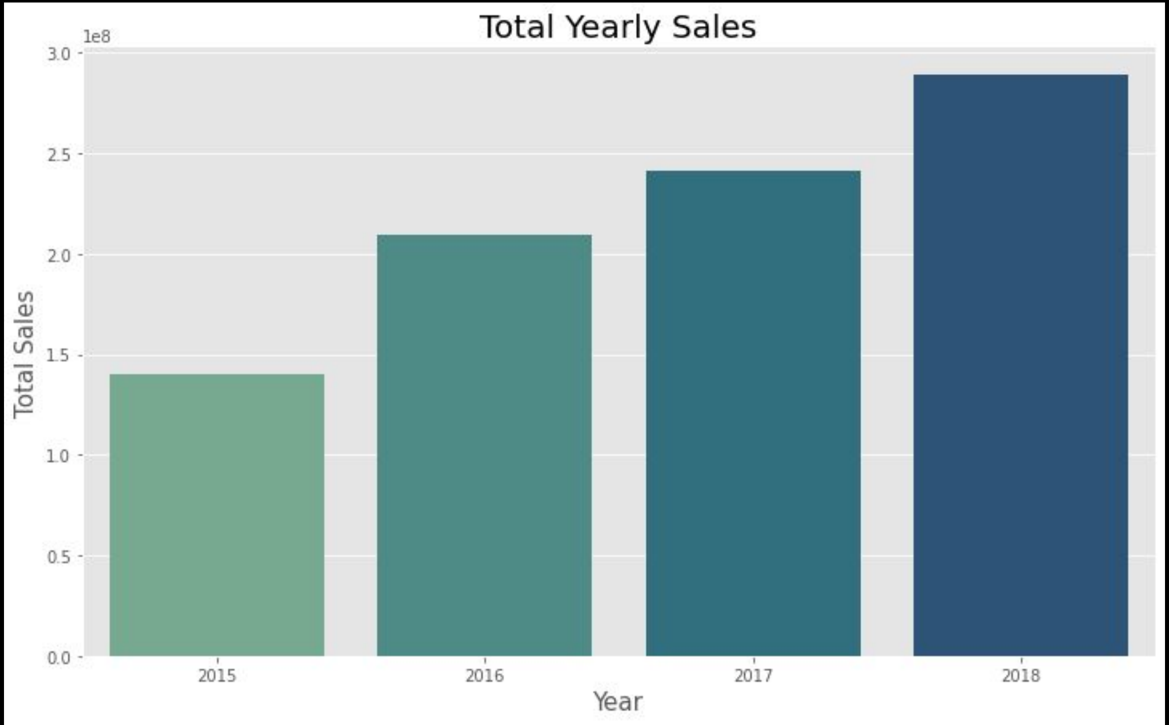
Total sales weekday basis.

Total Sales over time (Monthly).



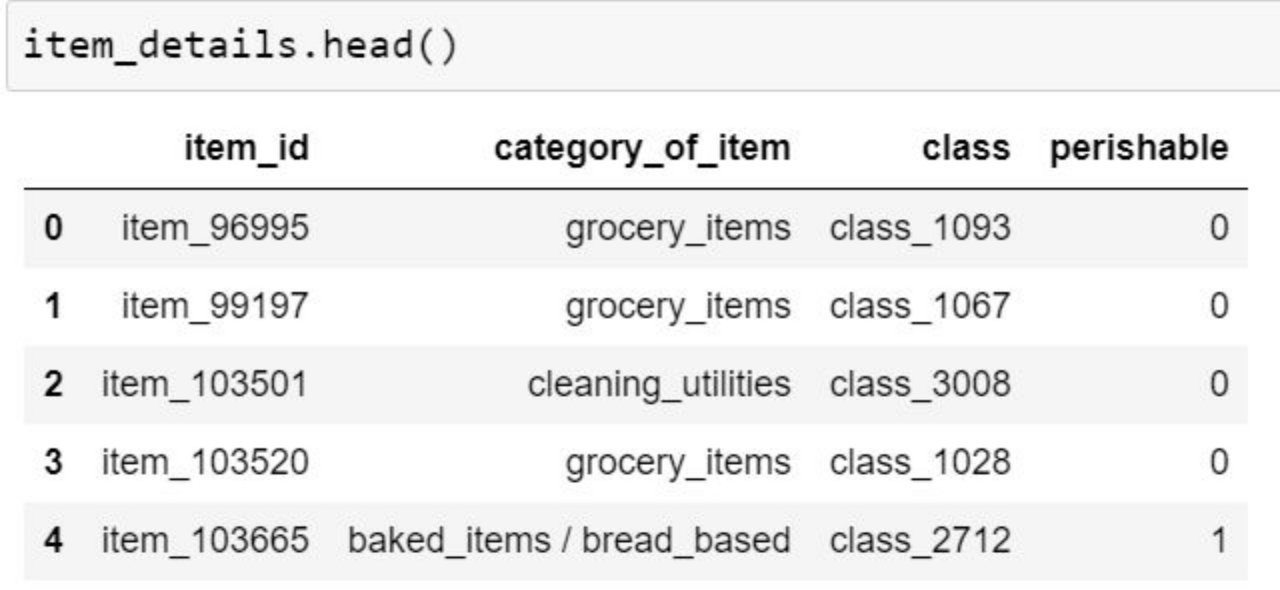


Average Sales over time (Monthly).



Total Sales over time (yearly).

Item Data



```
item_details.head()
```

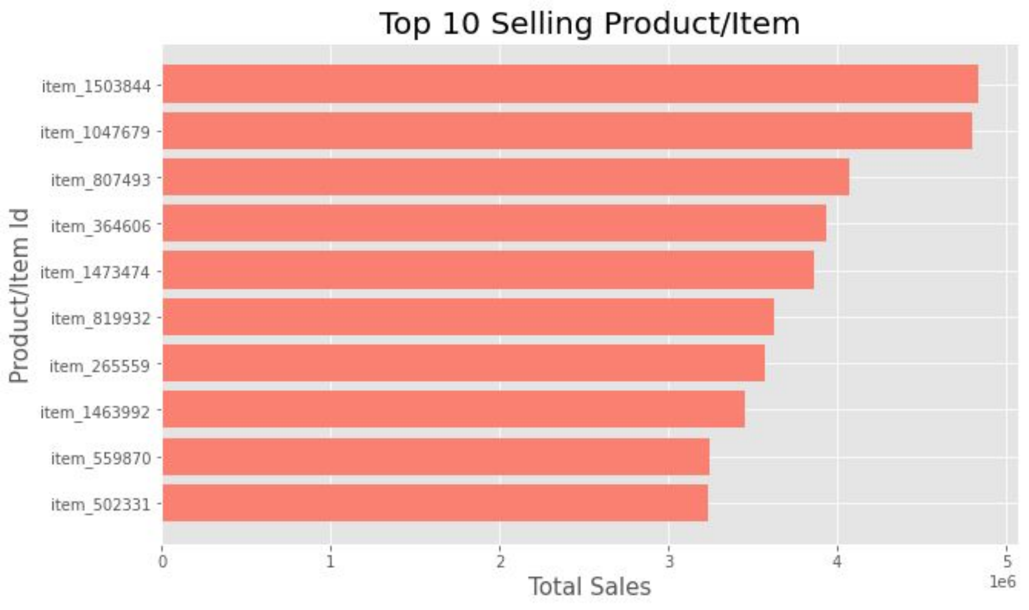
	item_id	category_of_item	class	perishable
0	item_96995	grocery_items	class_1093	0
1	item_99197	grocery_items	class_1067	0
2	item_103501	cleaning_utilities	class_3008	0
3	item_103520	grocery_items	class_1028	0
4	item_103665	baked_items / bread_based	class_2712	1

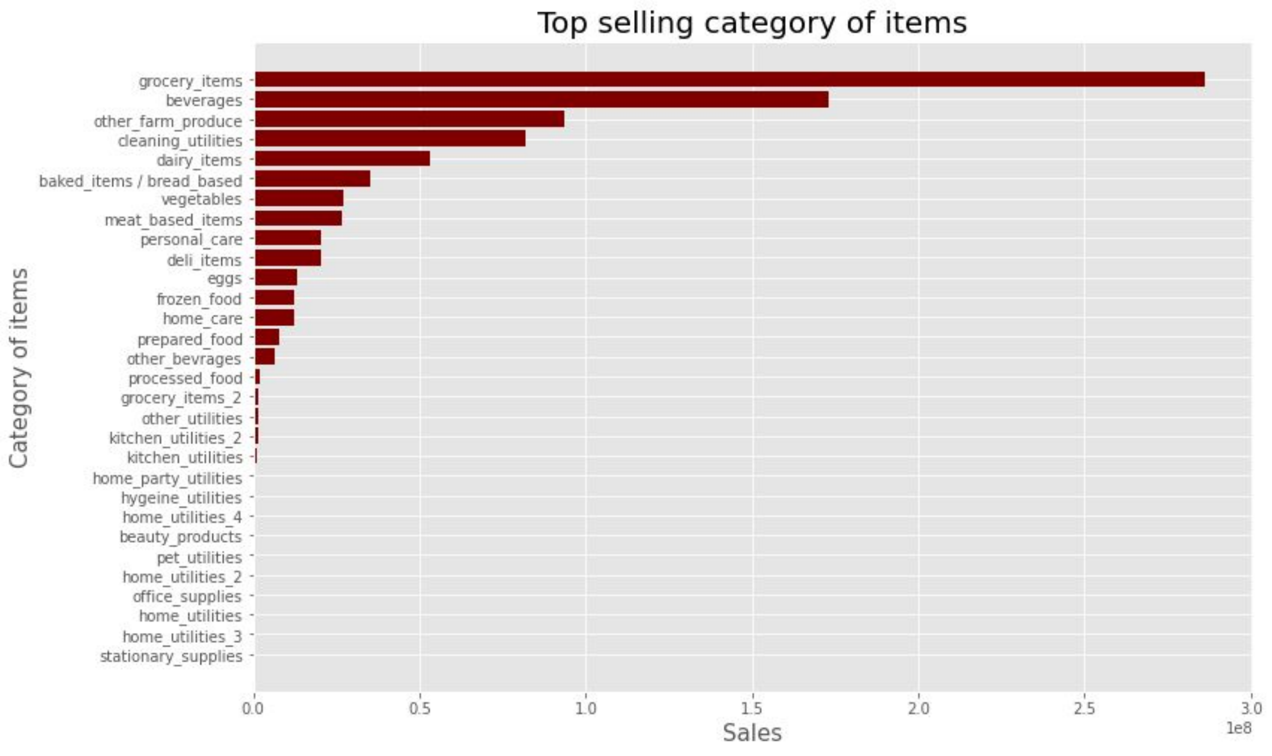
- Item data includes item_id (An identifier of a product), category_of_item (the category to which the product belongs to), class(Another way to classify product), perishable(Whether the item is perishable or not).

Item data

- There are 4100 unique number of products.
- There are 30 unique categories of item.
- There are 337 unique classes.

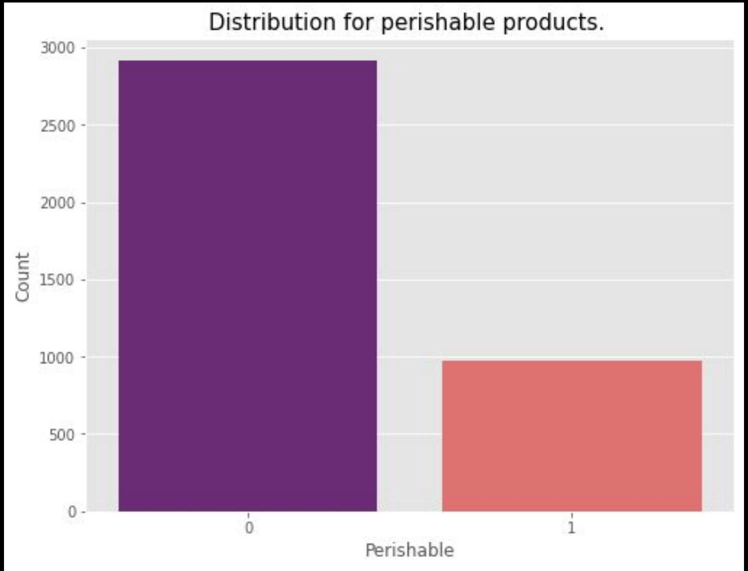
Top 10 selling products.



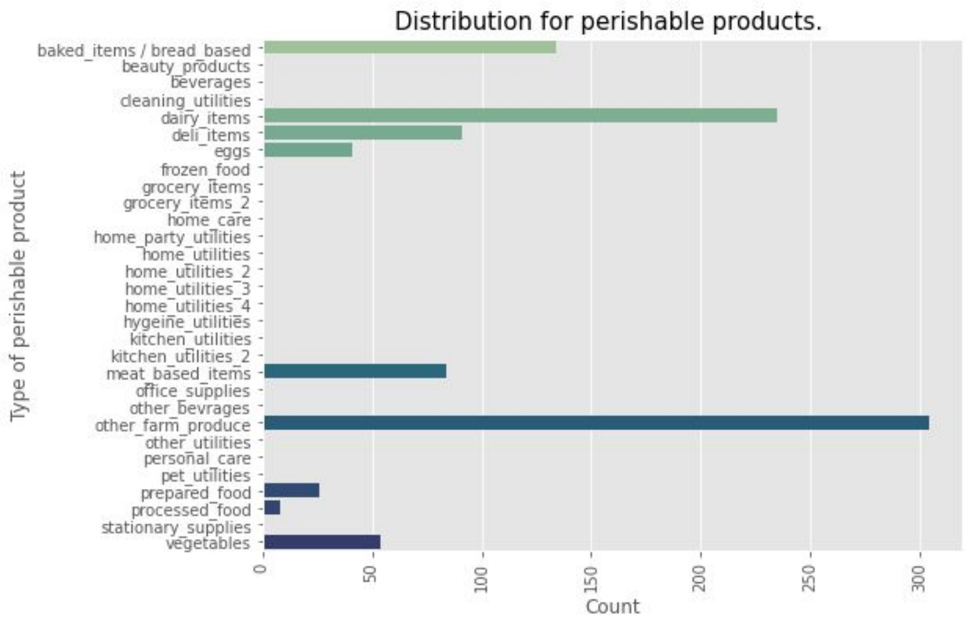


Grocery items , beverages , other farm produce, cleaning utilities, dairy items are the top selling categories.

Top selling categories.



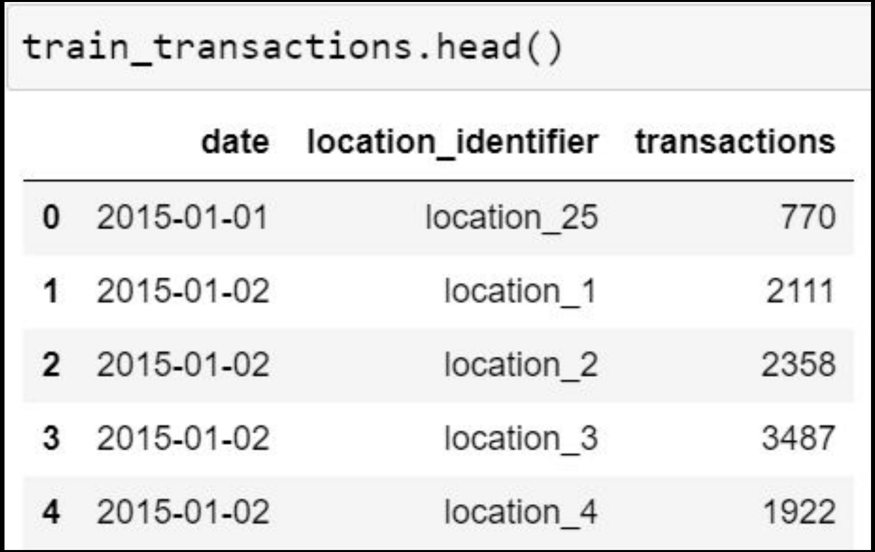
➔ **Majority of the items are non perishable.**



Top 5 Perishable Foods are

- **Other Farm Produce**
- **Dairy Items**
- **Baked Items / Bread Based**
- **Deli Items**
- **Meat Based Items**

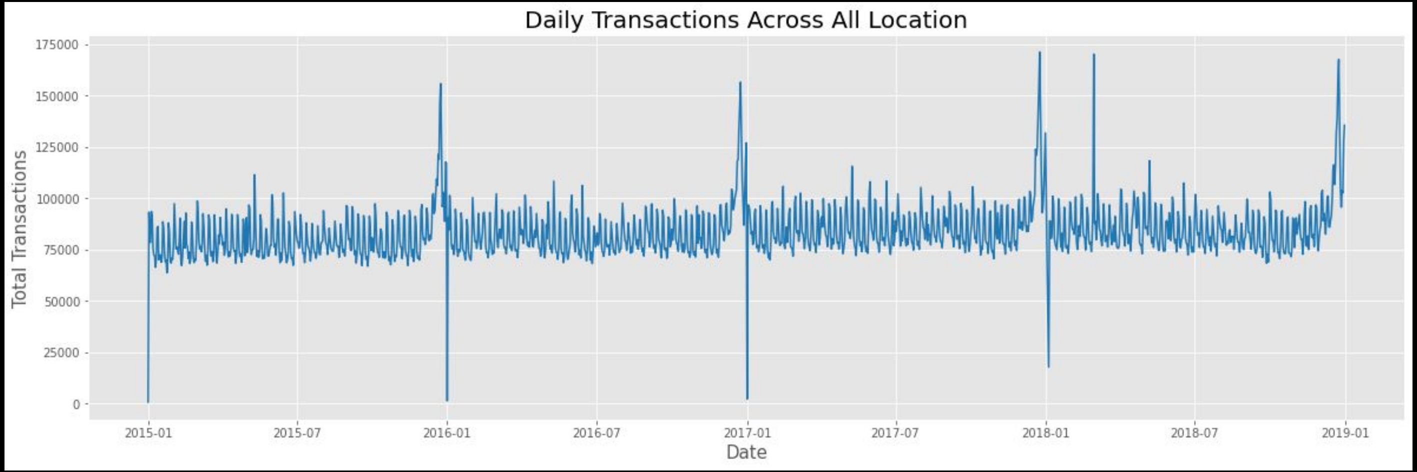
Transaction data



```
train_transactions.head()
```

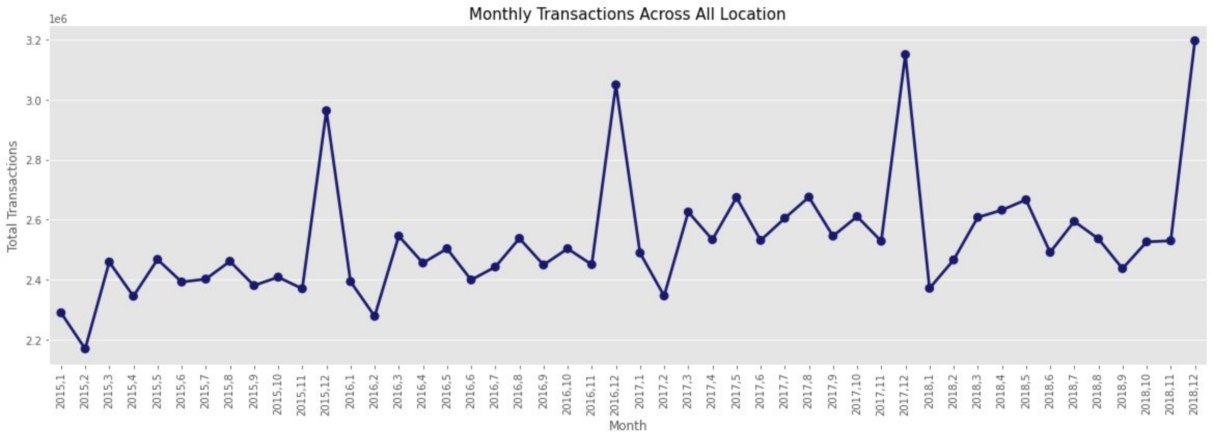
	date	location_identifier	transactions
0	2015-01-01	location_25	770
1	2015-01-02	location_1	2111
2	2015-01-02	location_2	2358
3	2015-01-02	location_3	3487
4	2015-01-02	location_4	1922

- Transaction data includes date, location_identifier (The location from where the transactions were handled) , transactions (The number of transactions handled by the particular location)

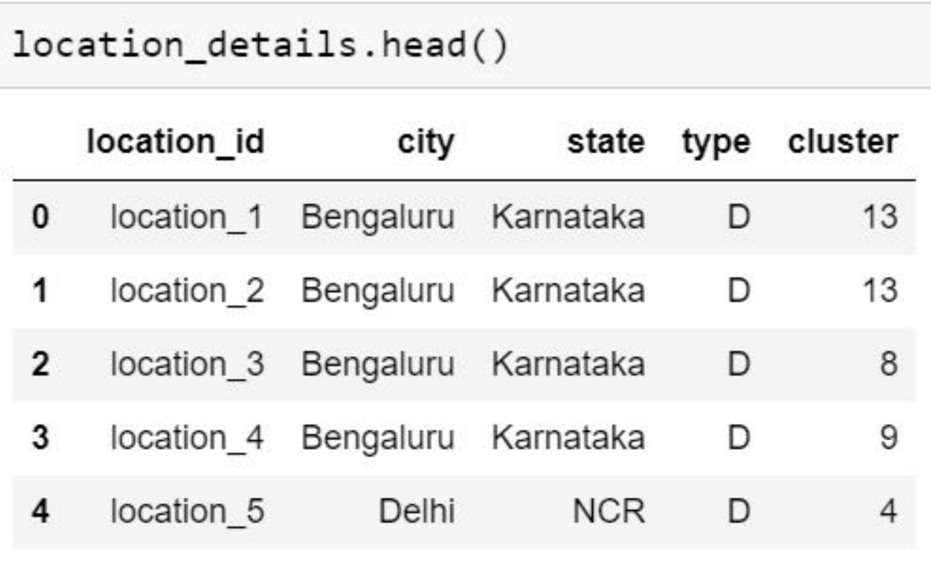


➔ **Daily transactions across all location**

➔ **Monthly transactions across all location**



Location data.



```
location_details.head()
```

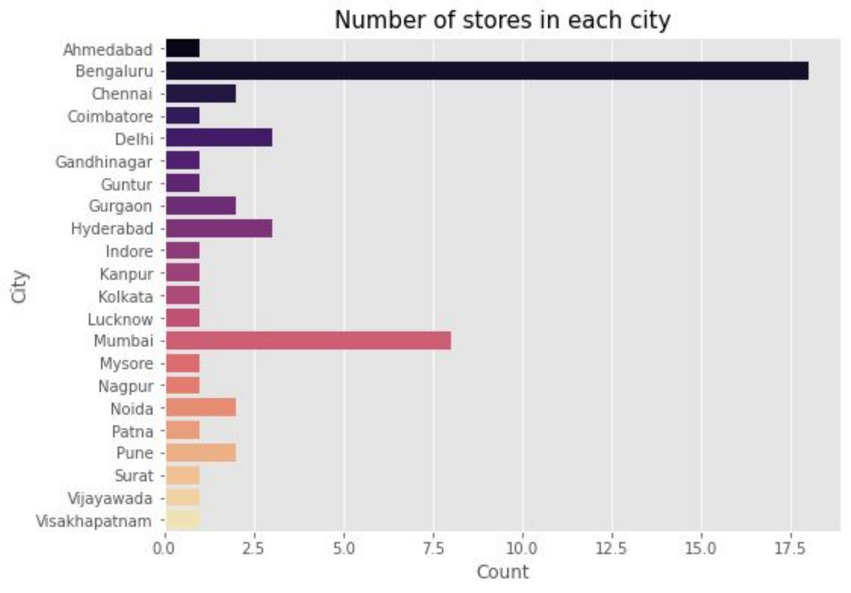
	location_id	city	state	type	cluster
0	location_1	Bengaluru	Karnataka	D	13
1	location_2	Bengaluru	Karnataka	D	13
2	location_3	Bengaluru	Karnataka	D	8
3	location_4	Bengaluru	Karnataka	D	9
4	location_5	Delhi	NCR	D	4

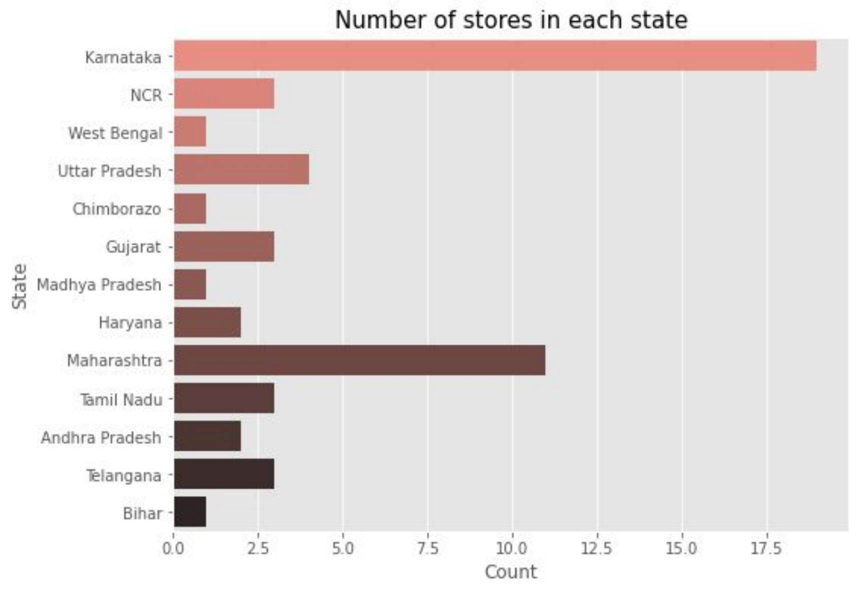
- Location data includes location_id (The location of the store / warehouse),city (The city where the unit is located), state (The state in which the city is located), type (The type of business unit (‘A’, ‘B’, ‘C’, ‘D’, ‘E’)) , cluster (The cluster that the unit belongs to)

Location data

- There are 54 unique stores across all India.
- They operate in 13 states and 22 cities.
- There are 5 different types of business units (‘A’, ‘B’, ‘C’, ‘D’, ‘E’).
- There are 17 different types of clusters. (1-17)

Count of stores in each city ,state.



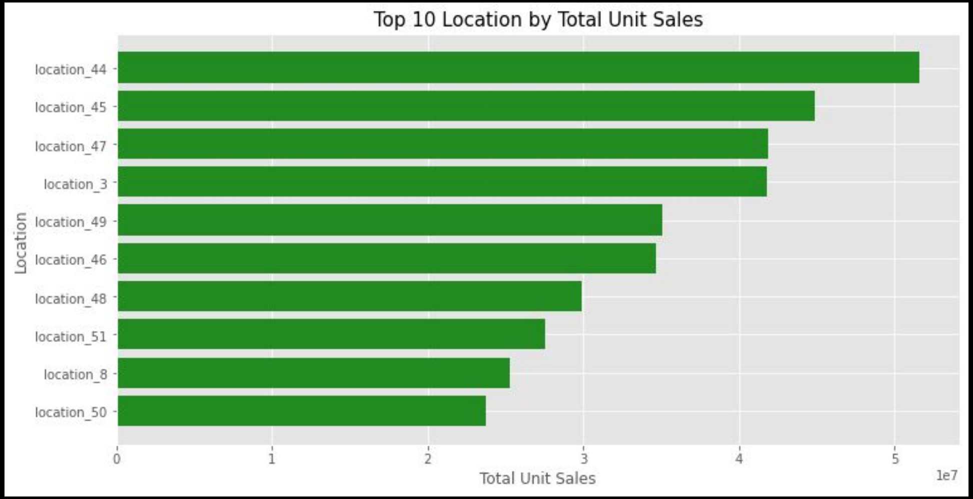


Top cities with majority of the store units are:

- Bengaluru
- Mumbai
- Hyderabad , Delhi

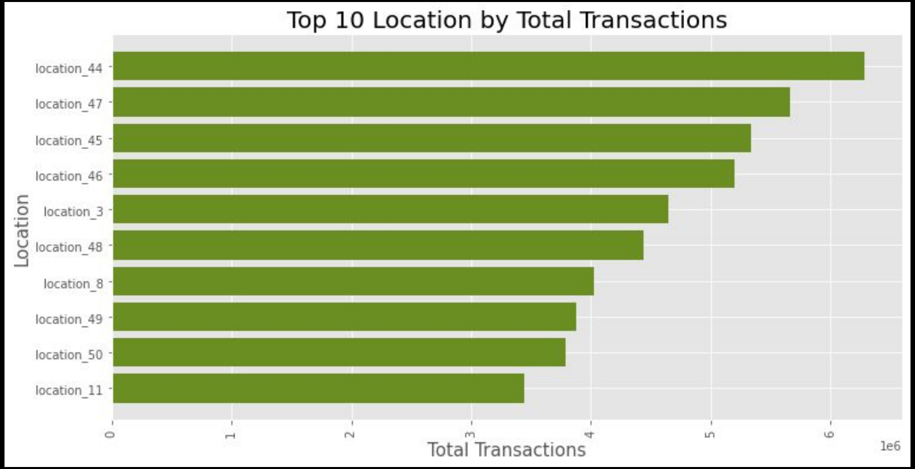
Top States with majority of the store units are:

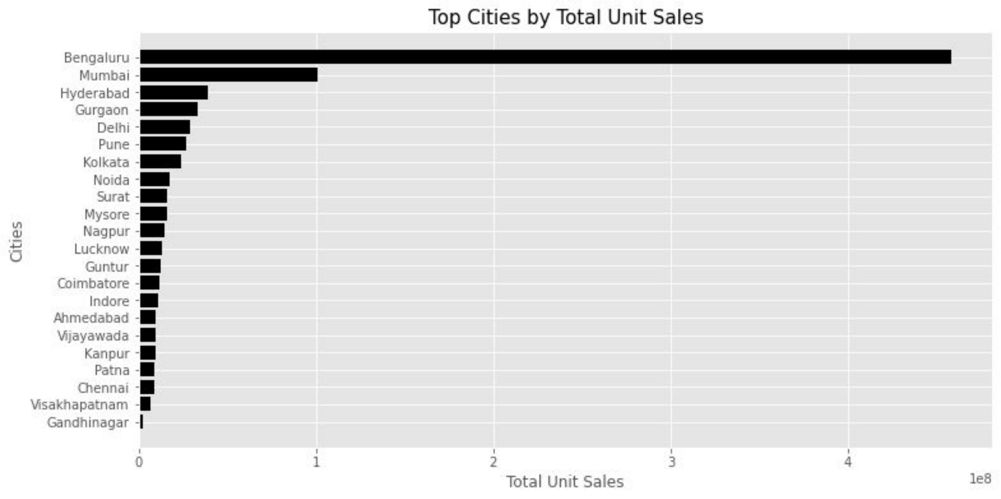
- Karnataka
- Maharashtra
- Uttar Pradesh
- Telangana
- Tamil Nadu

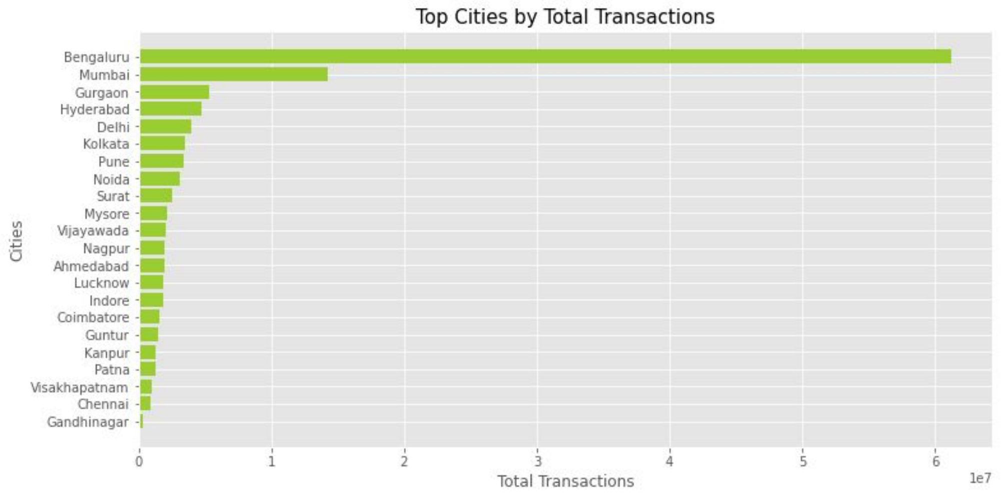


- ➔ **Top 10 location with highest unit sales.**

- ➔ **Top 10 location with highest transactions.**

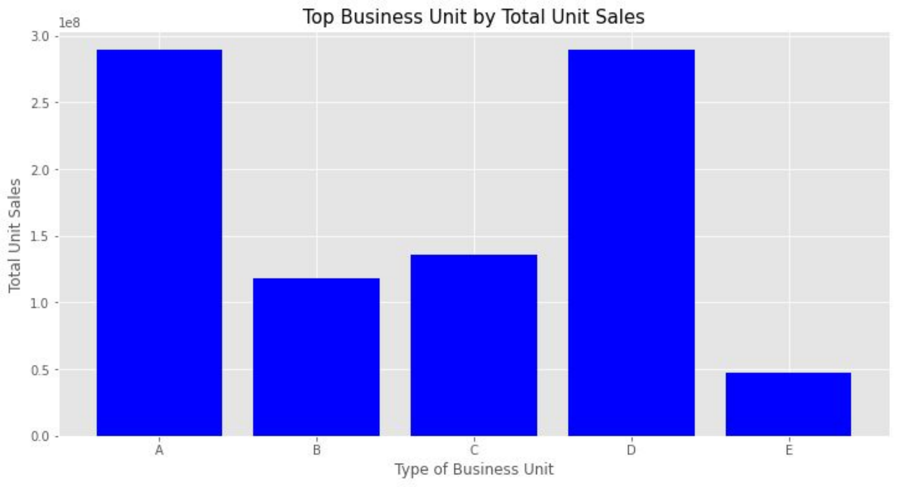






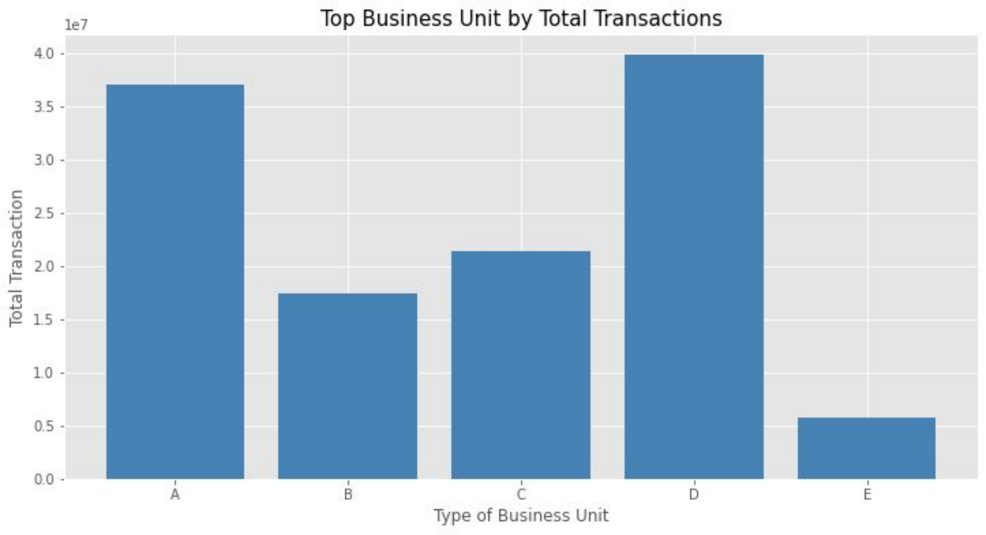
- ➔ **Cities with total unit sales.**

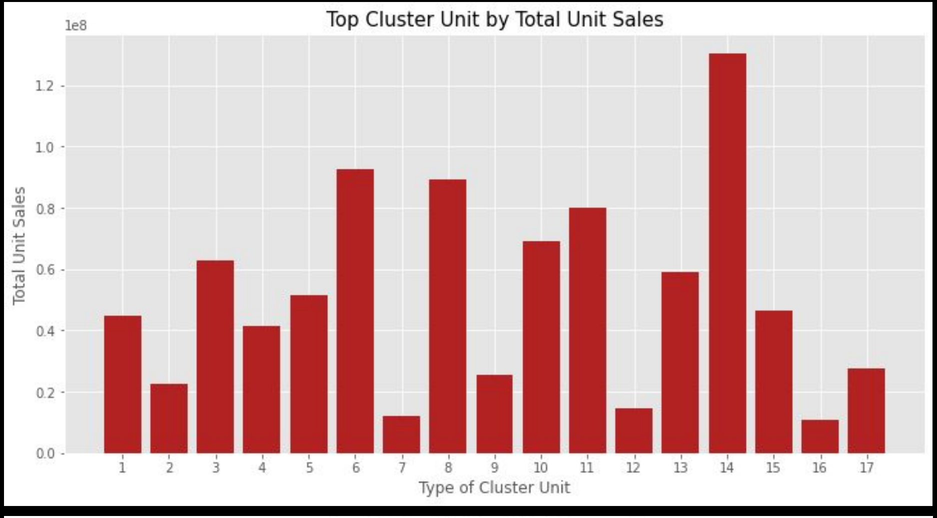
- ➔ **Cities with total transactions.**

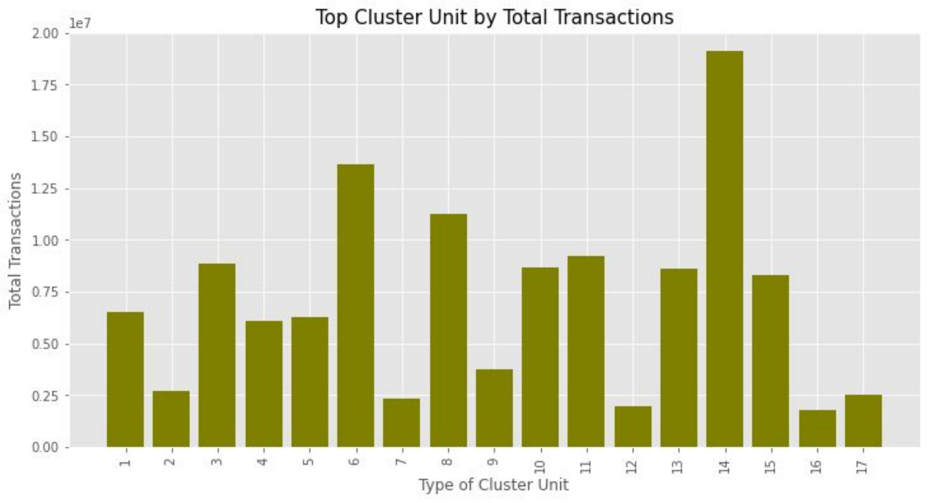


- ➔ **Business units with total unit sales.**

- ➔ **Business units with total transactions.**





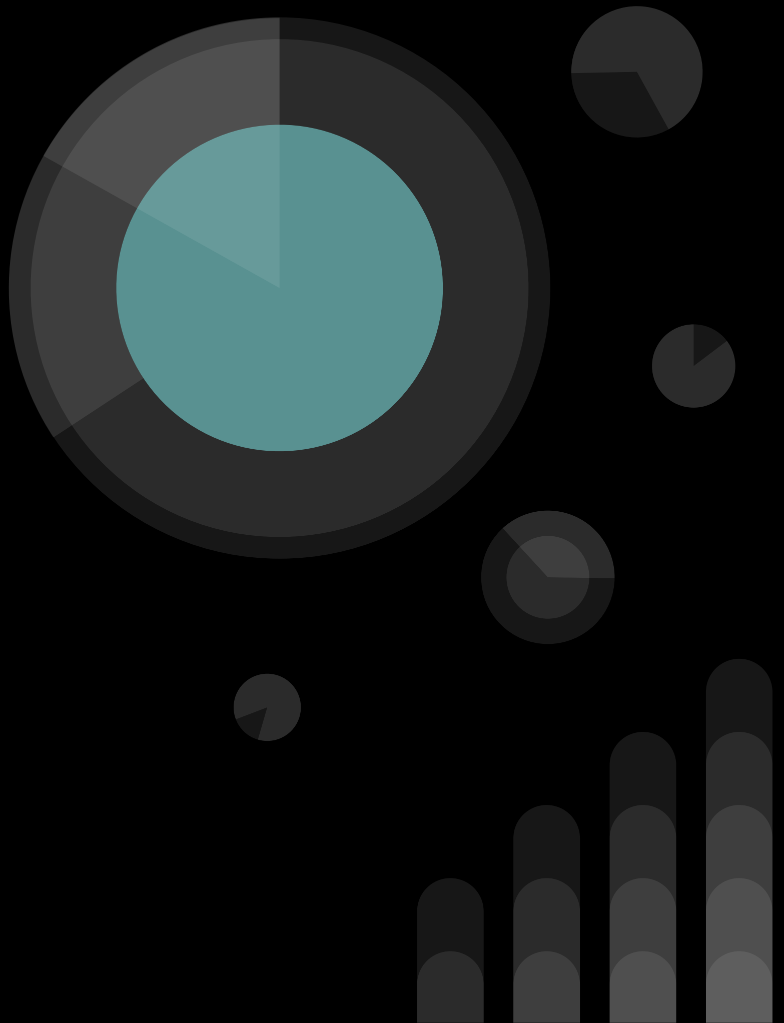


- ➔ **Clusters with total unit sales.**
- ➔ **Clusters with total transactions.**

EDA Final Thoughts.

- There’s an upward trend in year by yearly total unit sales. Which is a good indication that the sales are growing.
- Amount of transactions increases during the year end seasons.
- Even the total number of store units are 54 among them few stores have high amount of sales.
- Bangalore is an important location, both in terms of total unit sales and total transactions.
- Unit “A” and “D” generate most profit for the company.
- Cluster no. belonging to 14,6,8 shows maximum sales and transactions.

Predictive Analytics



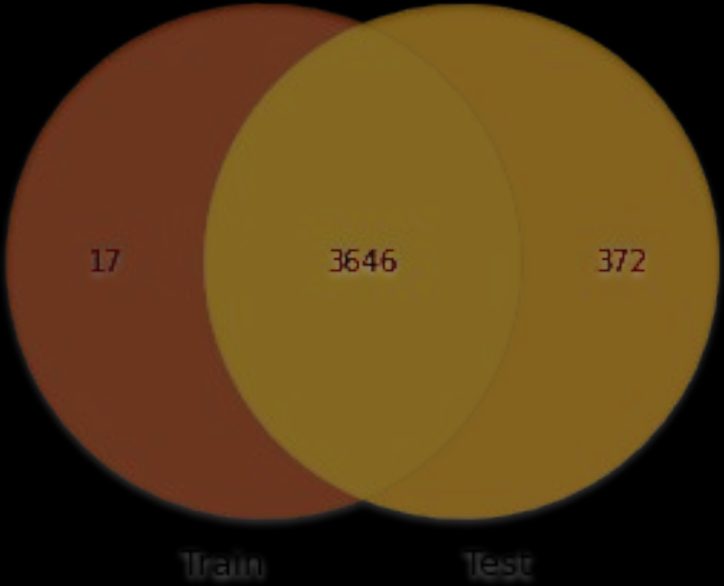
Challenges

1. Since our training data is large in size, its difficult to perform preprocessing activities. So from now on we'll work with only a subset of data. Since our test data contains dates ranging from 01/01/2019-15/08/2019, we'll try to build our model only with the previous year data where date ranging from 01/01/2018-15/08/2018.

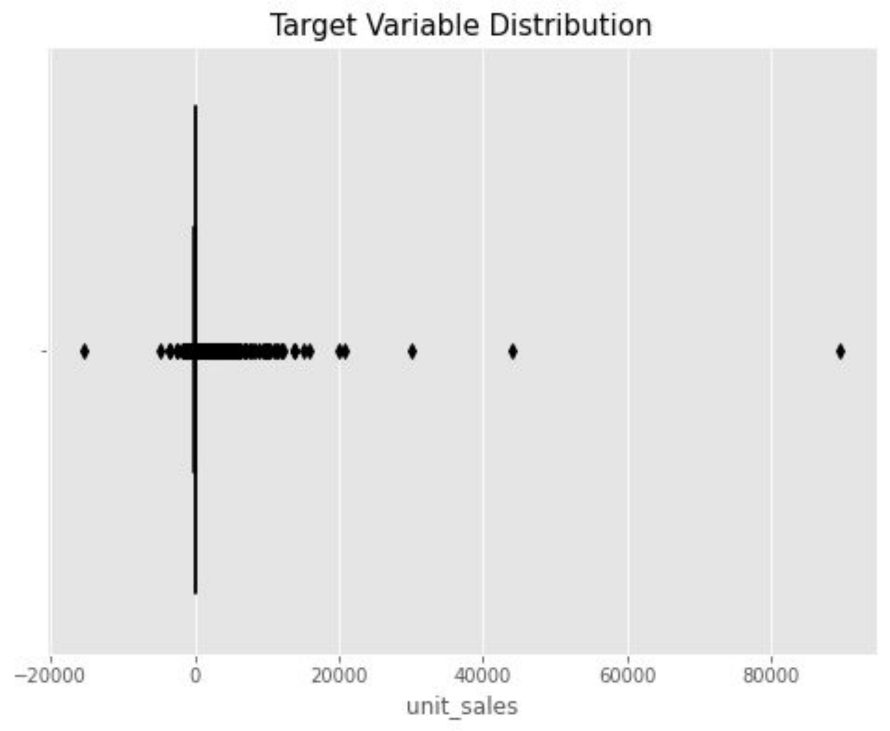
2. Although we have a time series problem, but since there are multiple instances of the same date it's difficult to figure out the time series model/algorithm for this problem. To overcome this issue, we'll try regression method with time features to build our models.

3. 'item_id' is a categorical variable with high cardinality. And There are 4018 distinct items in test. There are 3663 distinct items in train. There are 3646 items common in train and test. There are 372 items only in test data (not in train.) These items represent about 6.01% of test data. Will have to deal with these new items.

Challenges



Item distribution in train and test



Experiment 1

Data Preprocessing:

1. Loading train, item_details,location_details data.
2. Merging item and location data with train using left join.
3. Extracting date,month,week from the date column.
4. Dropping date and the features with high cardinality i.e. item_id, class from the dataframe.
5. Removing negative unit_sales.
6. Label Encoding all the columns except the target variable.
7. Converting all columns into category expect the target variable.
8. Splitting the data into train and validation set.
9. Log transforming the target variable.

Experiment 1

Model Building:

1. Applied Random Forest Regression and Gradient Boost Regression on the prepared dataset.
2. Predicted on the test data.

Results:

Model	Model Performance (Mape)
Random Forest Train (Without Log)	160.17869842229744
Random Forest Test (Without Log)	163.00
Gradient Boost Train	109.97420120515697
Gradient boost Test	119.15

Experiment 2

Data Preprocessing:

1. Loading train, item_details,location_details data.
2. Merging item and location data with train using left join.
3. Extracting date,month,week from the date column.
4. Dropping date and the features with high cardinality i.e. item_id from the dataframe.
5. Removing negative unit_sales.
6. Removed outliers using IQR.
7. Label Encoding all the columns except the target variable.
8. Converting all columns into category expect the target variable.
9. Splitting the data into train and validation set.
10. Log transforming the target variable.

Experiment 2

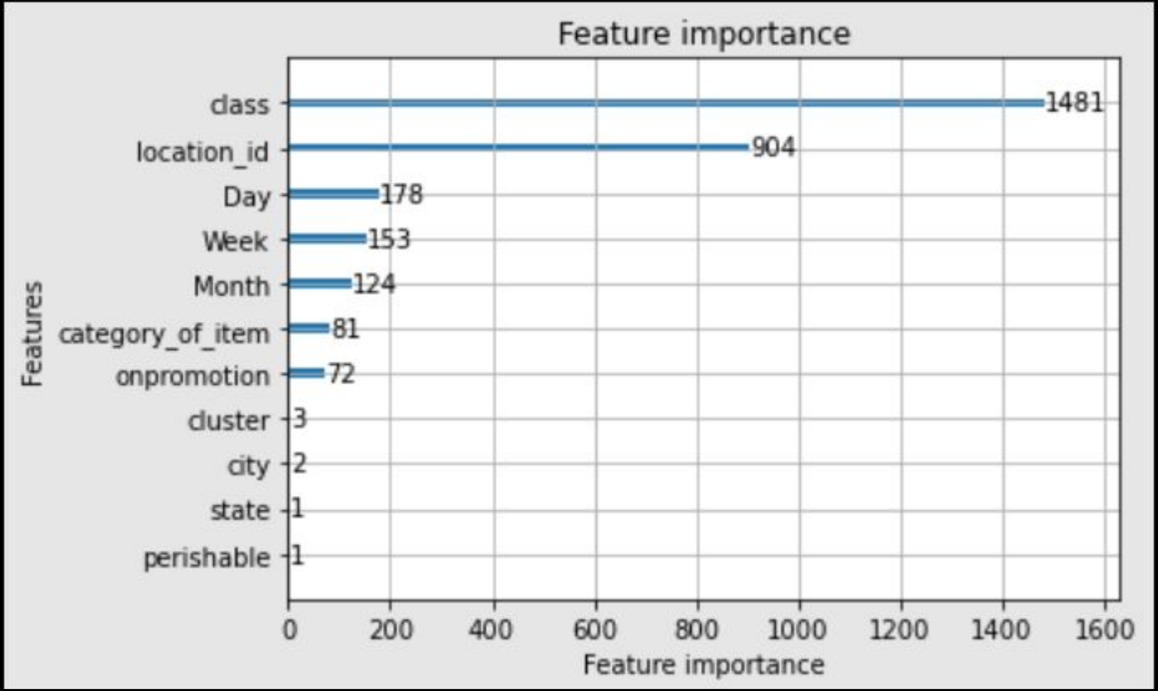
Model Building:

1. Applied LGBM (Light GBM) on the prepared dataset.
2. Predicted on the test data.

Results:

Model	Model Performance (Mape)
LGBM Train	82.58407165058243
LGBM Test	86.12

Experiment 2



Feature Importance for Experiment 2

Experiment 3

Data Preprocessing:

1. Loading train, item_details data.
2. Merging item data with train using left join.
3. Extracting date,month,week from the date column.
4. Dropping date and perishable column from the data.
5. Removing negative unit_sales.
6. Removed outliers using IQR.
7. Converting all columns into category expect the target variable.
8. Splitting the data into train and validation set.
9. Log transforming the target variable.

Experiment 3

Model Building:

1. Applied LGBM (Light GBM) with
   a. boosting_type= 'dart', # rf,goos,gbdt
   b. num_leaves = 31,
   c. objective = 'regression_l1', # l2,mape
   d. max_depth = 8, #3,4,5,7
   e. min_data_in_leaf = 50, #10,20,20,80
   f. learning_rate = 0.01, # 0.1,0.05,0.001
   g. metric = 'l1' # l2,mape
2. Predicted on the test data.

Experiment 3

Results:

Model	Model Performance (Mape)
LGBM Train	56.40810836436047
LGBM Test	60.24

Experiment 4

Data Preprocessing:

1. Loading train, item_details data.
2. Merging item data with train using left join.
3. Extracting date,month,week from the date column.
4. Dropping date and perishable column from the data.
5. Removing negative unit_sales.
6. Removed outliers using IQR.
7. Converting all columns into category expect the target variable.
8. Log transforming the target variable.

Experiment 4

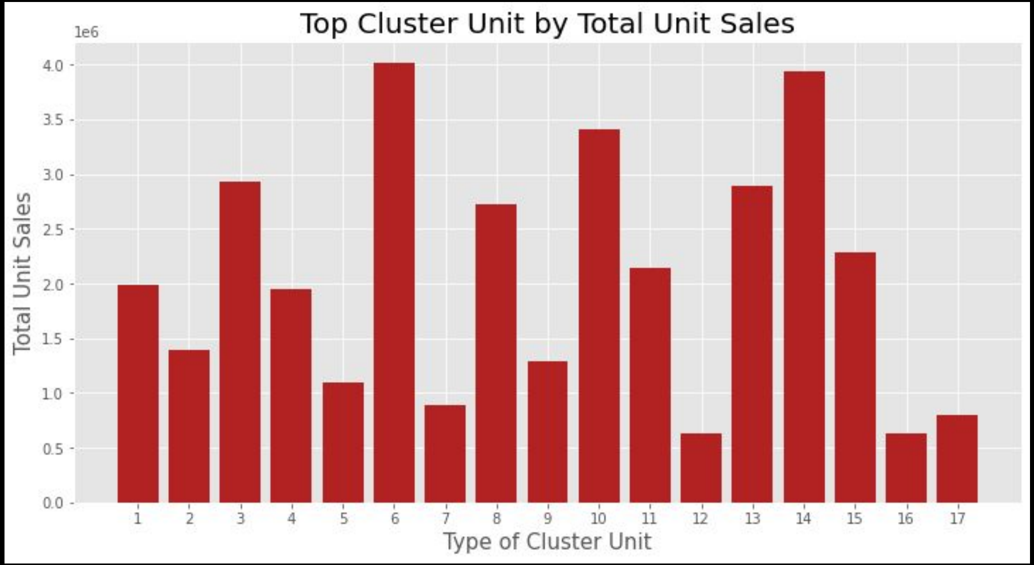
Results:

Model	Model Performance (Mape)
LGBM Train	56.36846
LGBM Test	59.89

Other experiments:

1. Categorical embeddings with single hidden layer. Test Mape - 64.
2. Generated nonlinear features with autoencoders and then added those feature to improve LGBM model.
3. LGBM model with different subset of data.
4. Added new items in the train data along with 0 for unit sales.

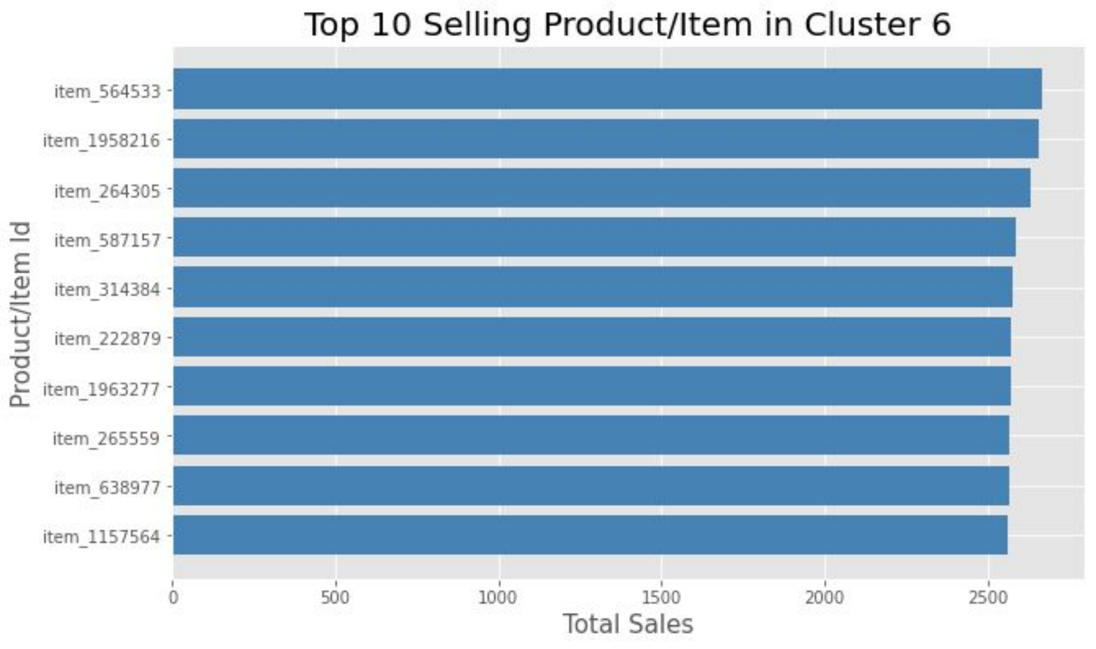
Answering questions from operations team.



Question : Business units belonging to which cluster will see the highest amount of sales in 2019 ?

Answer: Cluster 6

Answering questions from operations team.



Question : What are the top selling items in this cluster.

Answer: These are the top selling products in that cluster.

Future Work

1. Add more layers with categorical embeddings and check model performance.
2. Add time series related features with the training data and check model performance.
3. Handle new items in test data for more explainability.

References:

1. https://medium.com/bigdatarepublic/advanced-pandas-optimize-speed-and-memory-a654b53be6c2
2. https://towardsdatascience.com/understanding-lightgbm-parameters-and-how-to-tune-them-6764e20c6e5b?gi=e8c09744e86d
3. https://www.fast.ai/2018/04/29/categorical-embeddings/
4. https://lightgbm.readthedocs.io/en/latest/Parameters.html
5. https://arxiv.org/abs/1505.01866

Thank You

Ayan Maity