EDA on Retail Price Dataset

IE 509 Project Presentation
Prof. Narayan Rangaraj, IEOR
By: **Nisarg Jain**
23N0454

About Dataset

I have taken this dataset from Kaggle, Retail Price Optimization (kaggle.com) .

Datasets contains monthly sales of different product categories along with unit prices of ours compared with 3 other competitors. Along with various other features like Product Score, Customer Demand, Freight Price. As of now it has no null values

Description of Features of Dataset

i) product_id ii) product_category_name iii) month_year iv) qty

v) total_price vi) freight_price vii) unit_price viii) product_name_length

ix) product_description_length x) product_photos_qty xi) product_weight_g

xii) product_score xiii) customers xiv) weekday, weekend, holiday, month, year

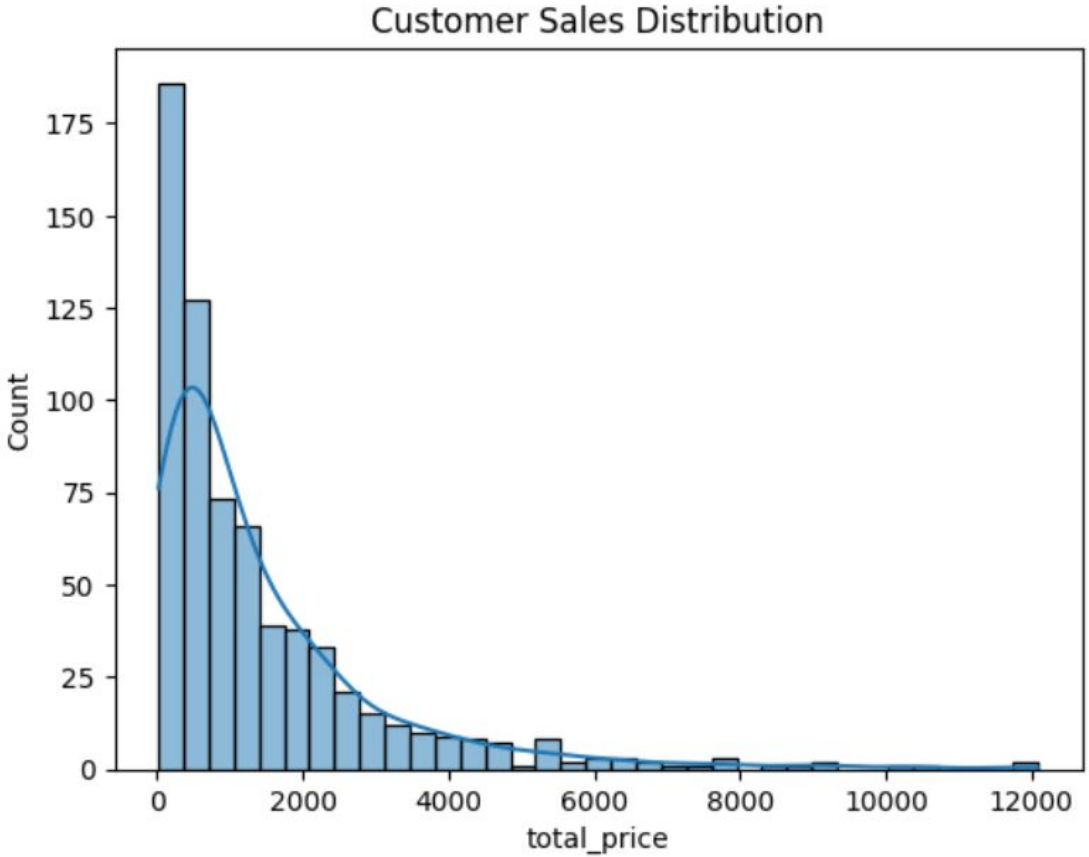
xv) volume xvi) 'comp_1', 'ps1', 'fp1' xvii) 'comp_2', 'ps2', 'fp2',

xviii) 'comp_3', 'ps3', 'fp3' xix) lag_price

Product Categories

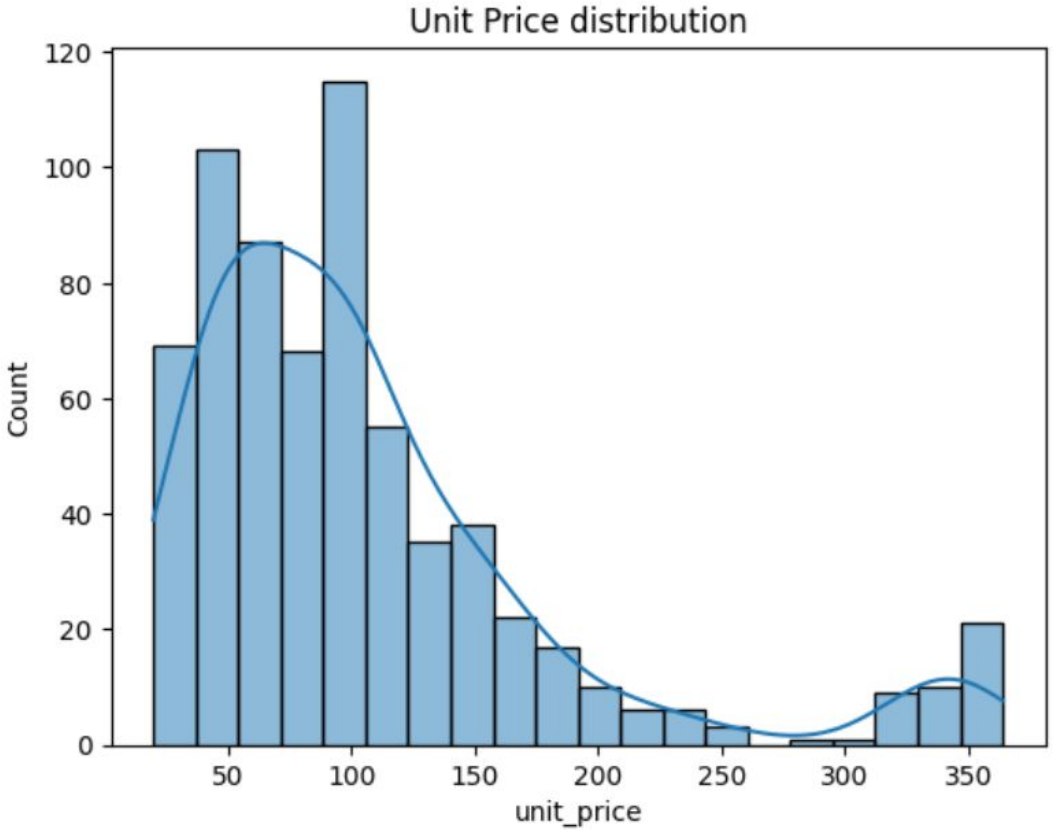
Unique List

```
['bed_bath_table',
 'garden_tools',
 'consoles_games',
 'health_beauty',
 'cool_stuff',
 'perfumery',
 'computers_accessories',
 'watches_gifts',
 'furniture_decor']
```



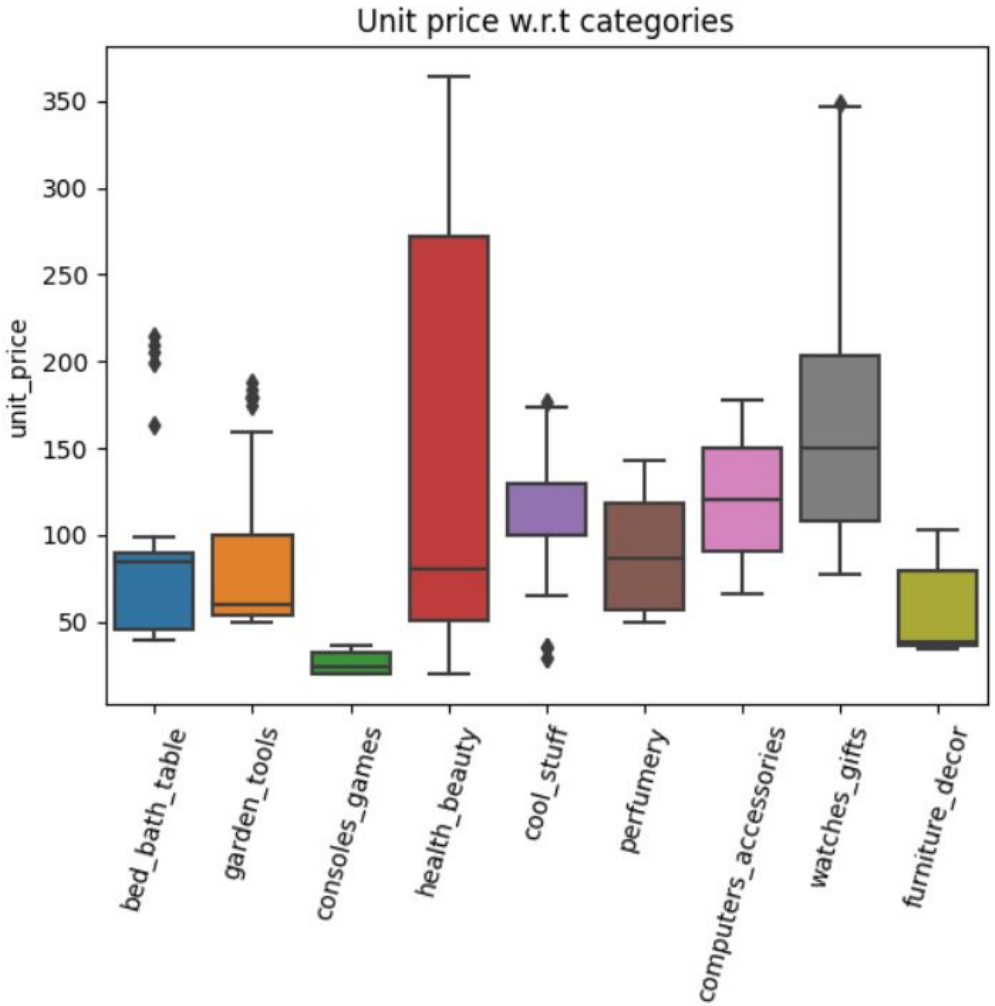
Distribution of Customer Spend on different months.

Plot shows low amount of extremes and most data concentrated around 0-2000$. Follows Gamma like distribution.



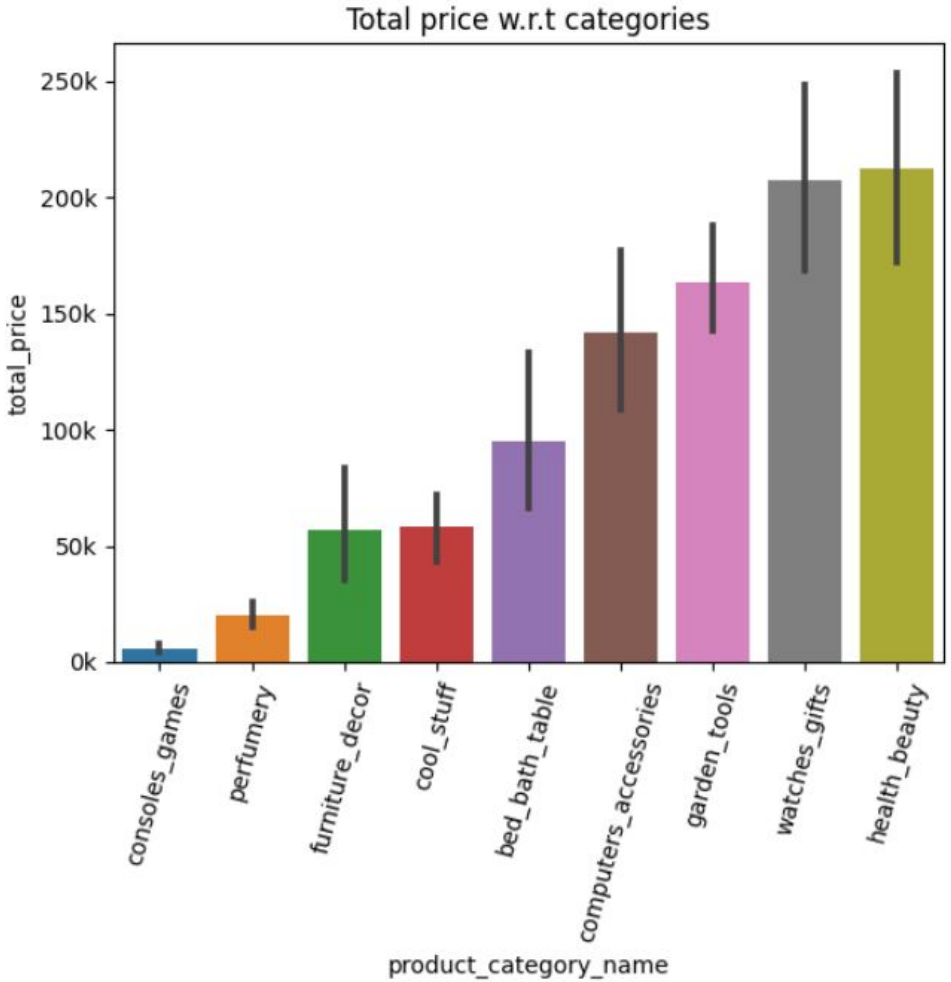
Distribution of Unit Price of different products across months

Plot reveals that most are around 0-200 range, but no mid range exists, after mid there are more extreme prices >=300.



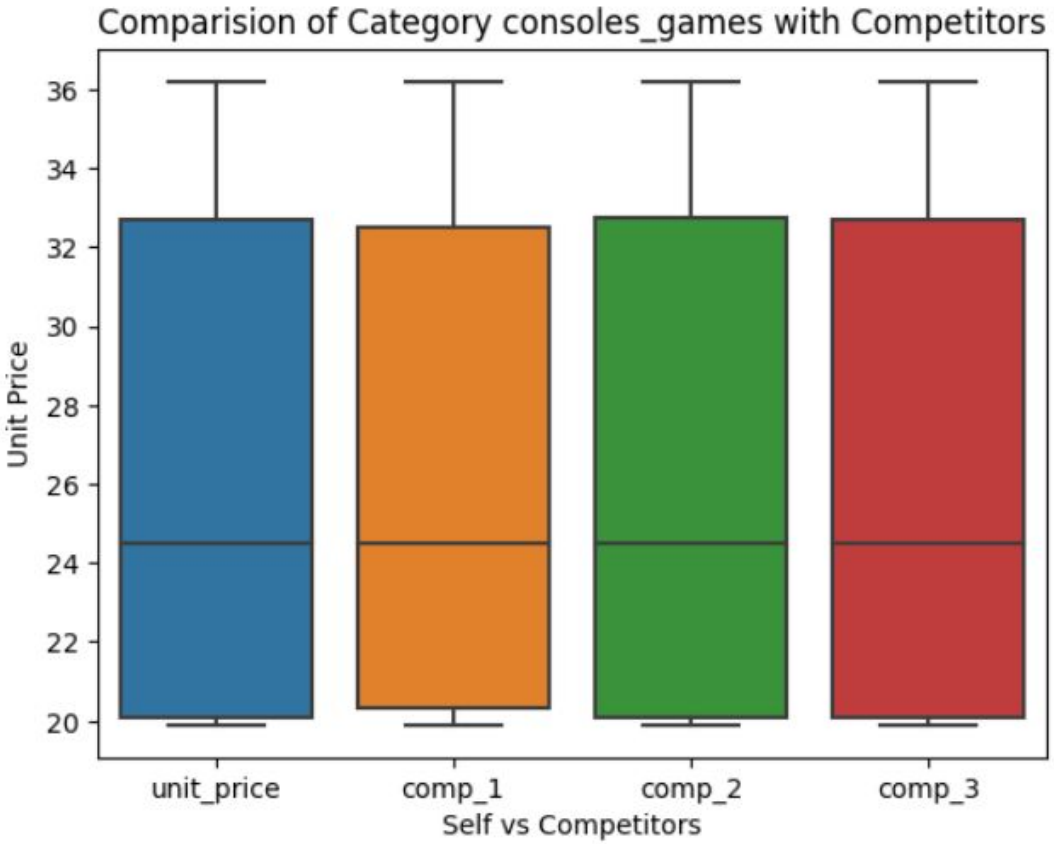
Box Plot of Unit Prices of our store across Categories Outlier Viz

We note few things, first see the variation is low in console prices, whereas we have highest variations in health and beauty . Outliers lie in bed bath products along with garden and cool-stuff. Luxury product like watches have higher extreme prices.



Total Price Sales across Categories for our store

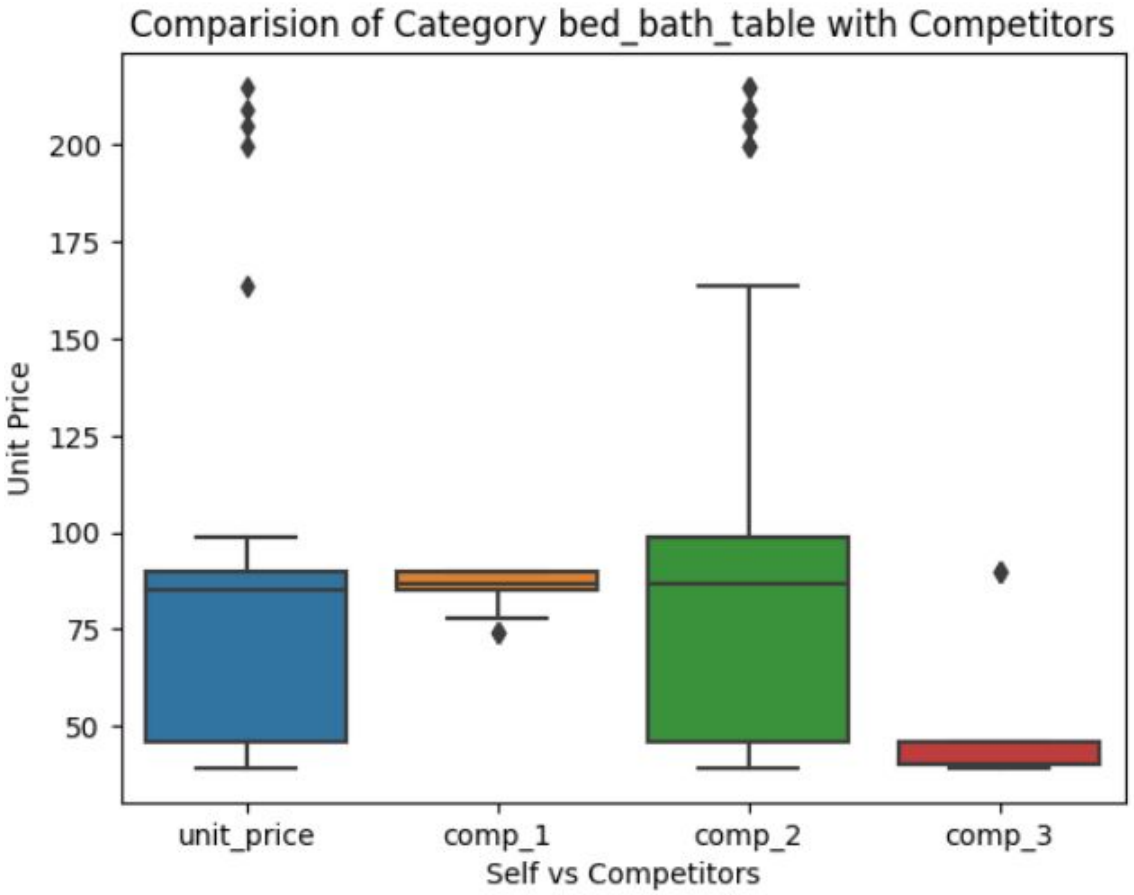
Note that our most total price sales are from health and beauty as well as watches. This is from the fact that unit price had high variations and we sold some product at way higher prices (these being luxury category).



Competitor Analysis on Unit Prices

Console Games

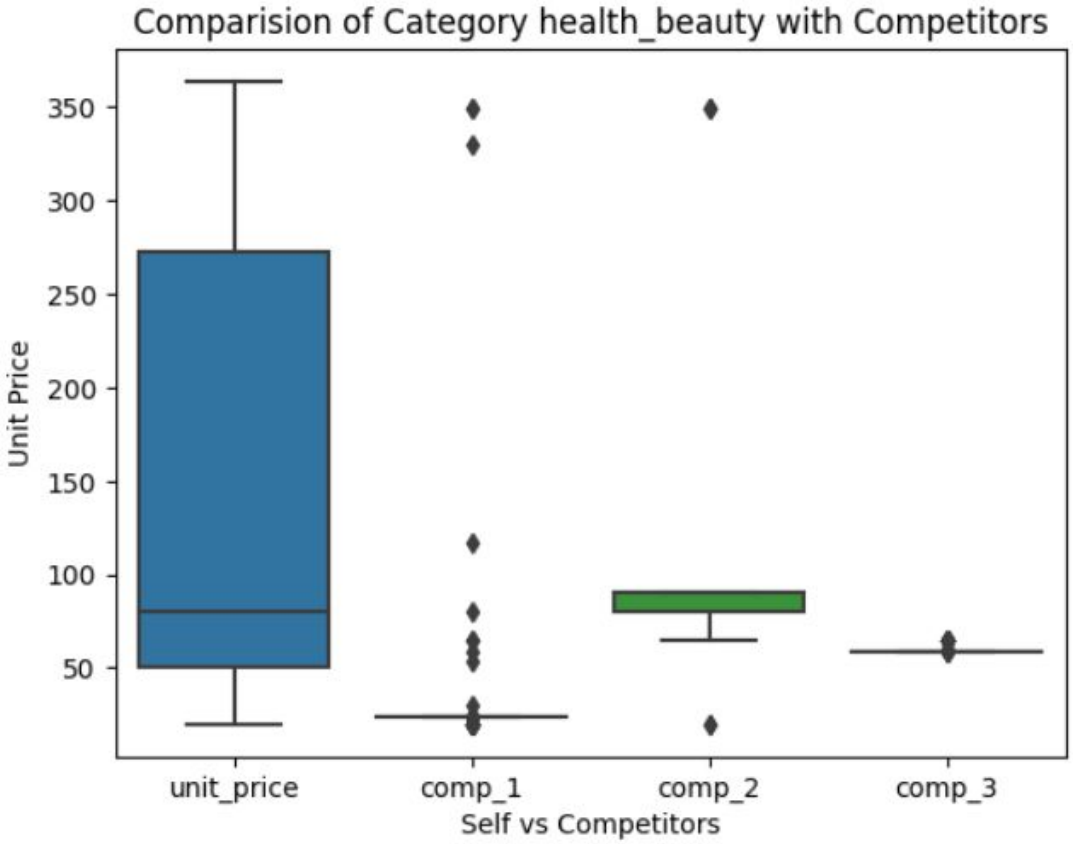
We plot few categories, where I feel we need to see the difference. Here, console_games being fixed priced items, we see that all competitors have to sell at the same price and if there is discount then all of them probably give the same discount.



Competitor Analysis on Unit Prices

Bed Bath Product

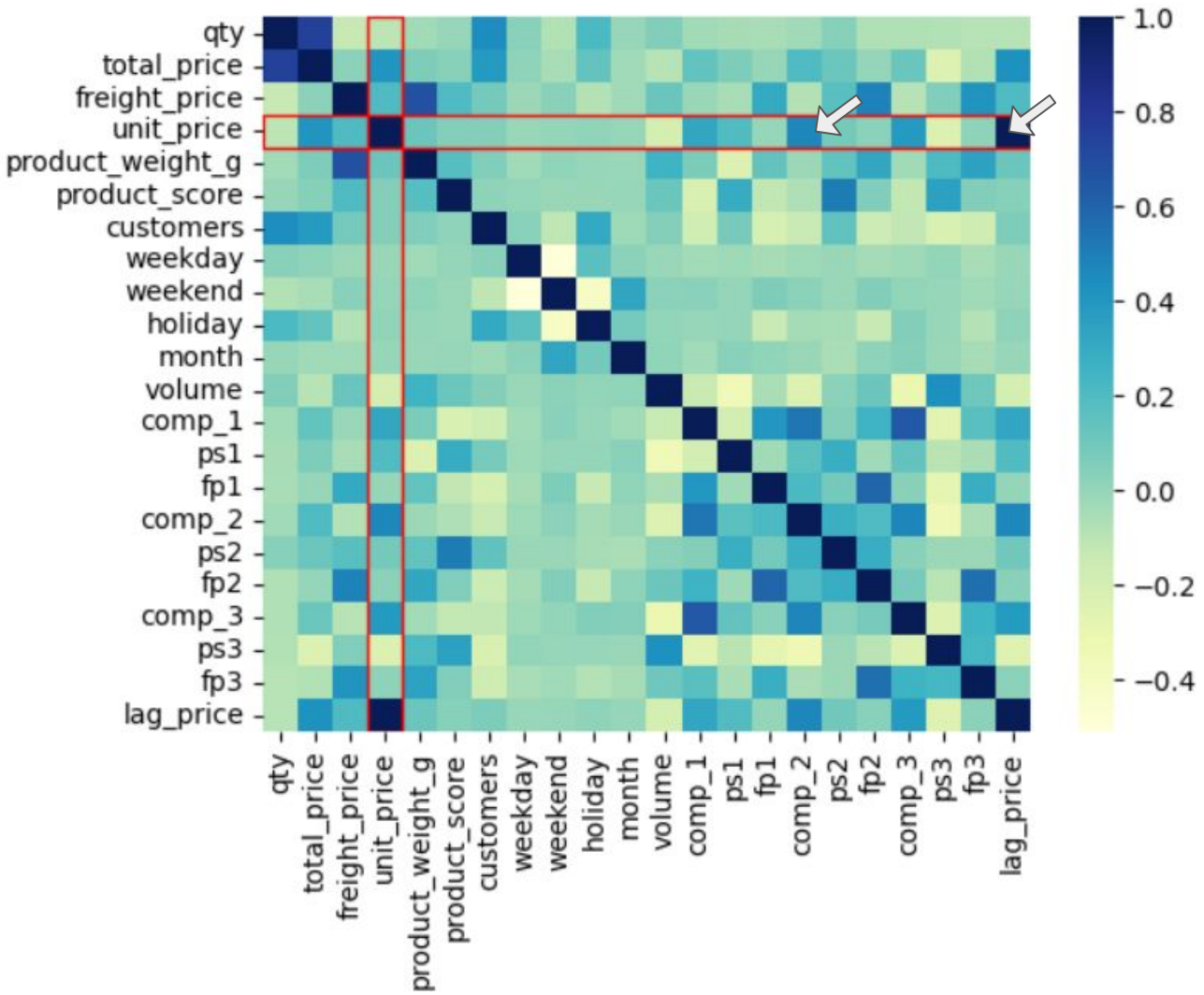
Comp1 and Comp3 sell on fixed prices, where as we try to sell max at comp1's price and sometimes give discount. Our prices vary similar to comp_2.



Competitor Analysis on Unit Prices

Health and Beauty

This is a category where we have varied range compared to all other competitors. Our median remains same as comp_2 but our range is higher and compared to all. Comp_1 and comp_3 engage very less in this category.



HeatMap of Feature Correlations.

Higher dependence on comp_2 unit price.
Lag Price is price difference of previous unit price vs current. Hence 100% correlation, as far as prediction is concerned this is redundant column, as one would not know lag price until unless we would know unit price, which is the target variable.

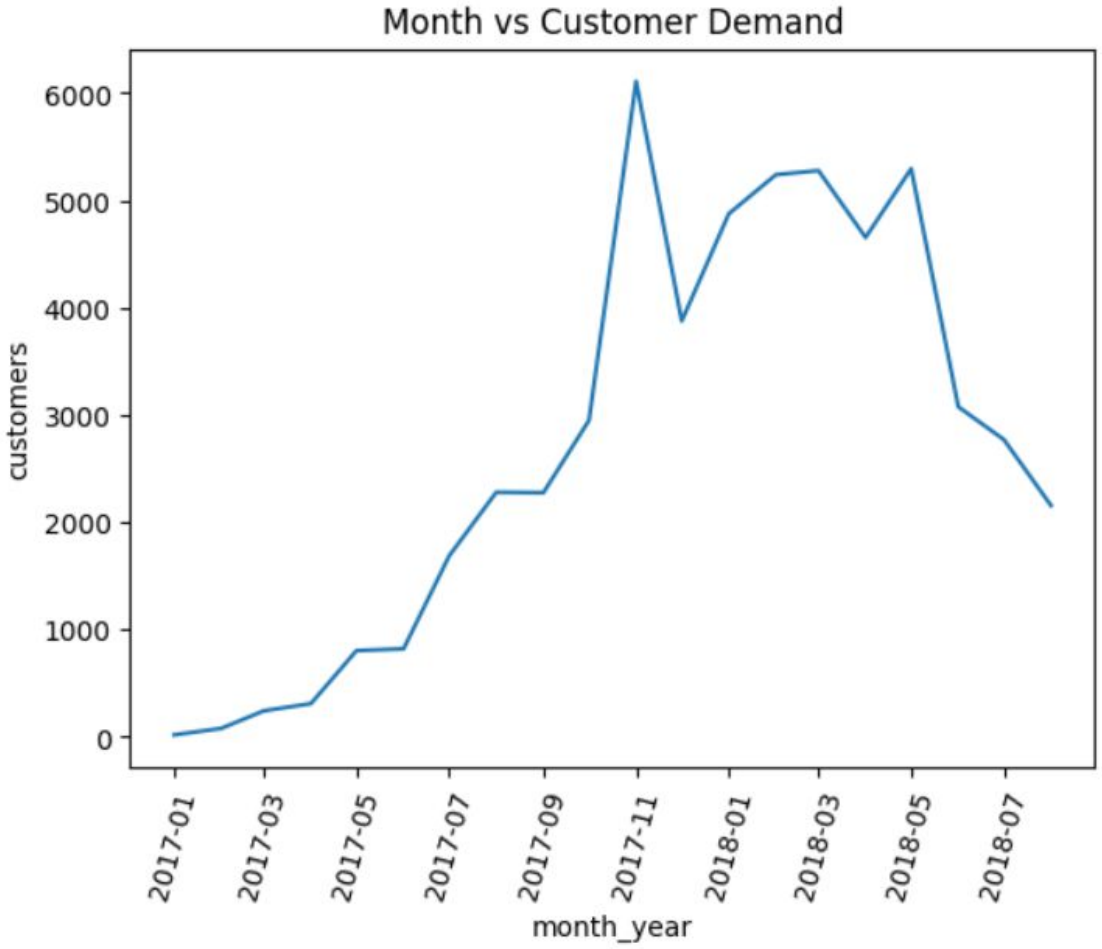
Monthly Revenue Analysis

Data Grouped by each month for all categories

	month_year	unit_price	product_id	total_price	freight_price	qty	weekday	weekend	customers	comp_1	comp_2	comp_3
0	2017-01-01	207.445000	health5	2864.19	33.961250	9	22.0	9.0	18	207.445000	207.445000	64.99
2	2017-02-01	99.990000	bed2	3584.11	217.847838	35	20.0	8.0	78	89.900000	99.990000	89.90
4	2017-03-01	99.990000	bed2	10204.38	282.314965	101	23.0	8.0	242	89.900000	89.990000	99.99
6	2017-04-01	96.656667	bed2	11524.62	335.440132	121	20.0	10.0	309	89.900000	96.656667	89.90
8	2017-05-01	92.445000	bed1	21843.33	393.828633	222	23.0	8.0	803	59.900000	89.990000	64.99
10	2017-06-01	99.990000	bed1	23245.24	498.717980	233	22.0	8.0	820	89.900000	89.990000	89.00
12	2017-07-01	89.900000	bed1	41049.89	617.072993	403	21.0	10.0	1686	75.000000	89.990000	59.90

RegPlot of Total Price Sales vs Monthly Customer Demands

Note that as customers demands is high in a month so is our total sales. The bands around the line represent 95% confidence interval for the point in line.



Demand Analysis

Increase in Demand in year 2017 but then started to decrease after 2018.

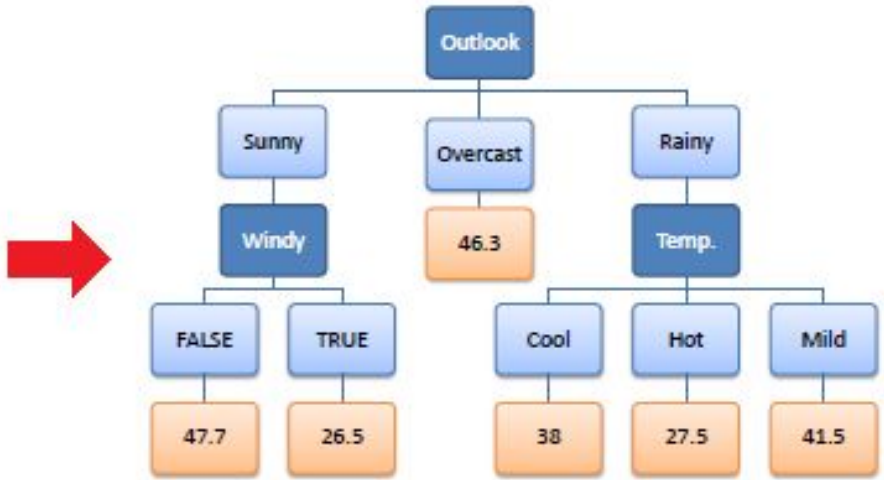
Feature Selection

Selected relevant columns to use in our prediction model for unit prices.
Decision made based on Heatmap and EDA.

	product_id	comp_1	comp_2	comp_3	fp1	fp2	fp3	product_score	unit_price	freight_price	customers
0	bed1	89.9	215.000000	45.95	15.011897	8.760000	15.100000	4.0	45.950000	15.100000	57
1	bed1	89.9	209.000000	45.95	14.769216	21.322000	12.933333	4.0	45.950000	12.933333	61
2	bed1	89.9	205.000000	45.95	13.993833	22.195932	14.840000	4.0	45.950000	14.840000	123
3	bed1	89.9	199.509804	45.95	14.656757	19.412885	14.287500	4.0	45.950000	14.287500	90
4	bed1	89.9	163.398710	45.95	18.776522	24.324687	15.100000	4.0	45.950000	15.100000	54

Model: Random Forest Regressor

Predictors				Target
Outlook	Temp	Humidity	Windy	Hours Played
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



```python
from sklearn.model_selection import train_test_split
# sklearn.model_selection.train_test_split(*arrays, test_size=None, train_size=None, random_state=None, shuffle=True, stratify=None)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state=1, stratify = strat)

X_train.drop(['product_id'], axis = 1, inplace = True)
X_test.drop(['product_id'], axis = 1, inplace = True)
```

```python
model = RandomForestRegressor(n_estimators=50, random_state=40)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Took Train Test Split of 80-20 % and used stratification on categories for even split.

Results

```
print(f"R2 score: {r2_score(y_test, y_pred)}")
```

```
R2 score: 0.8714278471307245
```

```
print(f"Mean Absolute Error: {mean_absolute_error(y_test, y_pred)}")
```

```
Mean Absolute Error: 17.36959764406324
```

Somehow our model underestimates unit prices of higher side.

Feature Importance

```
perm = PermutationImportance(model, random_state=1).fit(X_train, y_train)
eli5.show_weights(perm, feature_names = X.columns.tolist())
```

Weight	Feature
0.7322 ± 0.1460	freight_price
0.6591 ± 0.1040	comp_2
0.4720 ± 0.0824	product_score
0.1594 ± 0.0305	comp_3
0.1417 ± 0.0103	comp_1
0.0658 ± 0.0096	fp2
0.0268 ± 0.0114	fp1
0.0152 ± 0.0048	fp3
0.0126 ± 0.0023	customers

Tools Used

- For Analysis : Pandas, Numpy
- For Visualization: Matplotlib, Seaborn
- For Model Building: Sklearn, eli5

Future Work

- Demand Forecasting for different products, to predict change in unit prices.
- Creating a new feature, where we calculate demand price elasticity for each product type.

Thank You!