Credit Card Transaction Fraud Detection

0.1 Background

BigMart is a leading retail store chain that operates in various cities. The management at BigMart is interested in understanding the sales patterns across different stores and optimizing their inventory and marketing strategies accordingly. To achieve this goal, they have collected historical sales data for a set of products from different stores.

0.2 Problem Statement

The retail industry, characterized by its dynamic nature and extensive product offerings, faces the challenge of optimizing sales forecasting to maximize revenue and streamline inventory management. BigMart seeks to enhance its sales prediction accuracy through the implementation of advanced machine learning techniques.

0.3 Objective

The objective of this project is to develop robust regression models capable of accurately forecasting the sales of various products across different BigMart outlets. Leveraging historical sales data spanning multiple outlets and product categories, the aim is to construct predictive models that capture the intricate relationships between key factors influencing sales, such as product visibility, store size, location, product price, and seasonal variations.

0.3 Solution

This is a supervised learning regression problem, where the target variable is the sales figure for each product in each store. We will explore various machine learning techniques to build predictive models, evaluate their performance using appropriate metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2 score and Adjusted R2 Score. After examining the metrics we will then select the best-performing model for deployment.

Our process will start with Exploratory Data Analysis (EDA) with a goal to gain better insight about our data set and eventually develop a model using machine learning methodology.

1. Exploratory Data Analysis

I have obtained the data set and now it is time to perform an Exploratory data Analysis (EDA) to gain insight about the dataset and prepare the data for modeling purposes.

1.1 Dataset

“Train.csv” is the historical dataset containing sales data of Bigmart across multiple chains. The dataset contains sales information of outlets established from the year 1985 to the year 2009.

1.2 Observation

- The dataset contains 8523 observation
- The Dataset does not contain 3878 null values and no duplicate values
- The dataset contains a total of 12 columns out of which we have a single dependent variable labeled “Item_Outlet_Sales”
- The Item_Outlet_Sales feature contains continuous data , which represents the outlet sales of the respective item in the observation identified by the “Item_Identifier” column.

1.3 Linear Regression Assumption

A major goal of this project is to check how well suitable the data is to run a linear regression model examining the assumptions of linear regression during the exploratory data analysis phase. The assumptions to check are mentioned below:

1. Linear relationship
2. Multivariate Normality
3. Multicollearity
4. No Heteroscedasticity

1.4 Univariate Analysis

1.4.1 Distribution of Features

The following diagram shows the distribution of all our categorical data in our data set using countplot from seaborn. The plot covers the following features that are present in the dataset 'Outlet_Type','Item_Fat_Content', 'Outlet_Size','Item_Type', 'Outlet_Location_Type', 'Outlet_Establishment_Year'

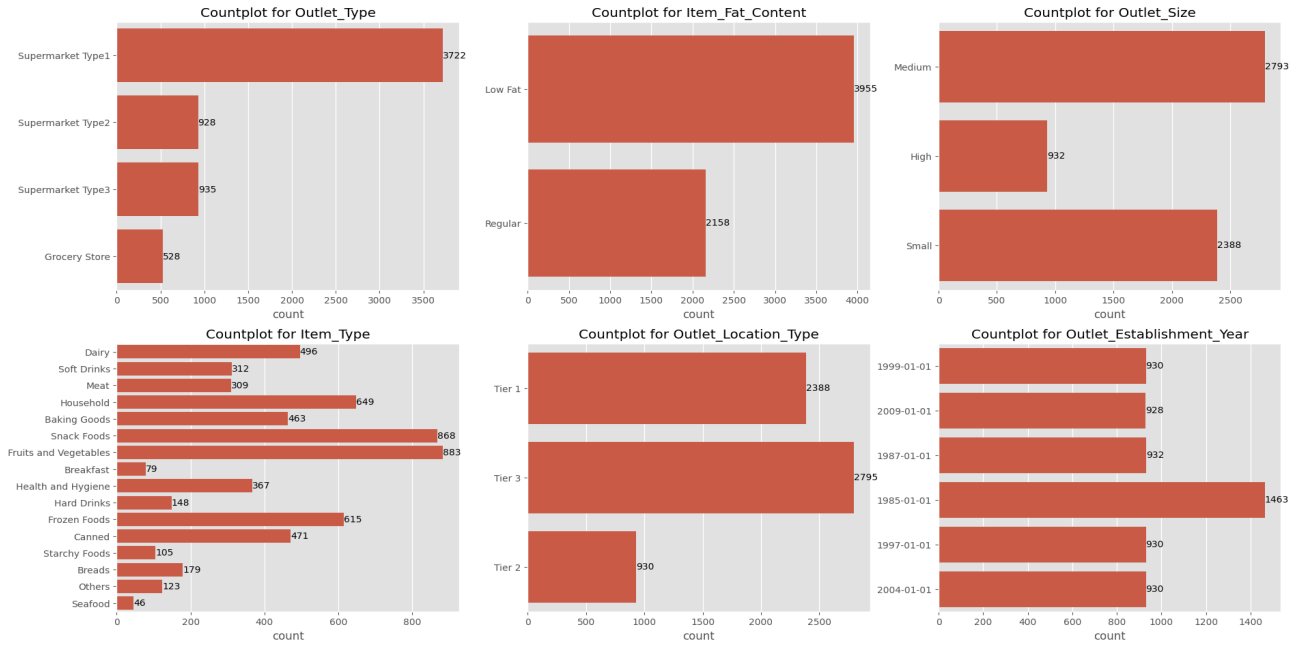

Figure 1. Countplot of Categorical Features

We can see that our categorical features have slight imbalance but most of the values are distributed to an acceptable degree for our purpose.

1.4.2 Distribution of Numeric Features

The following diagram shows the distribution of the numeric features in the data set i.e "Item_Weight", "Item_Visibility", "Item_MRP", "Item_Outlet_Sales".

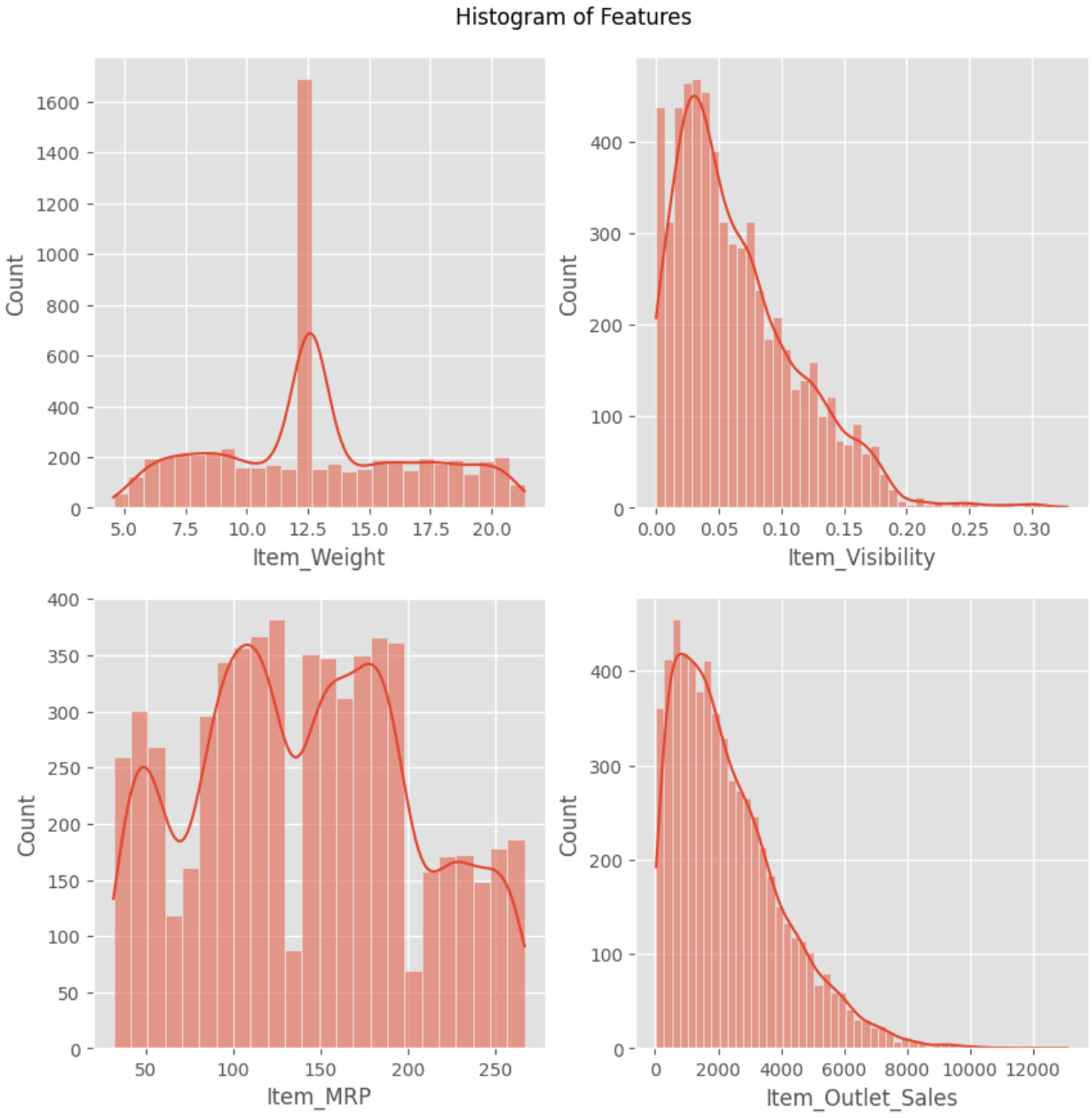

Figure 2. Histogram of Numeric Features

We observe that the distribution of the data is not normal. Item visibility and outlet sales exhibit right-skewed tails, indicating a concentration of lower values with few higher values. The distribution of Item_Weight is peaked in the middle, likely resulting from the imputation of missing values using the median. Conversely, Item_MRP demonstrates a thin-tailed distribution. The diagram of QQ-plot below also confirms these observations.

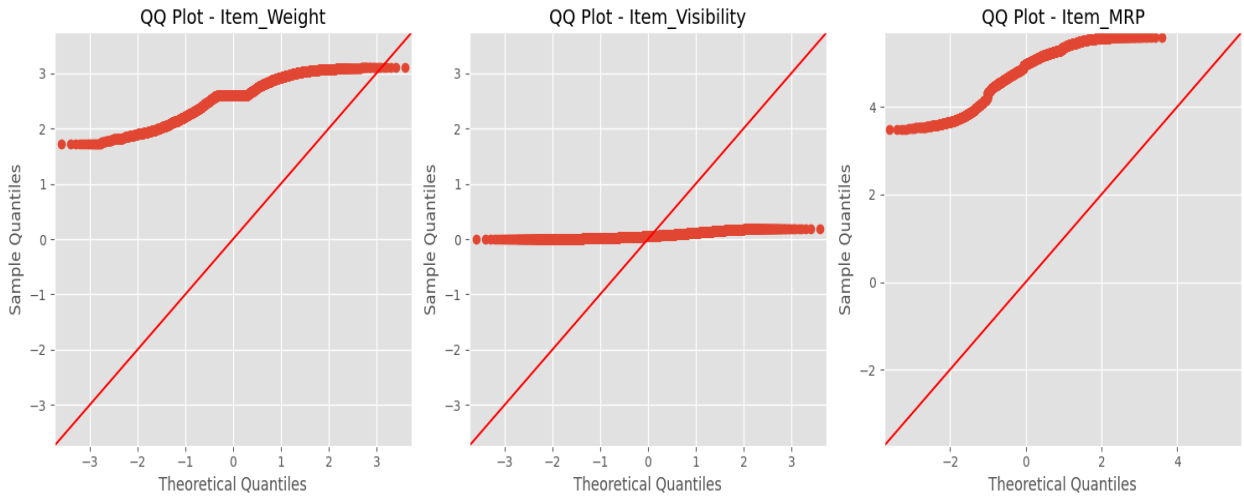

Figure 3. QQ-Plot of Numeric Features

1.4.3 Boxplot of Numeric Feature

The following diagram shows the median, interquartile range and outliers of the numeric features in the dataset i.e "Item_Weight", "Item_Visibility", "Item_MRP", "Item_Outlet_Sales". The figure makes it clear that “Item_Visibility” has a great deal of outliers. We will be

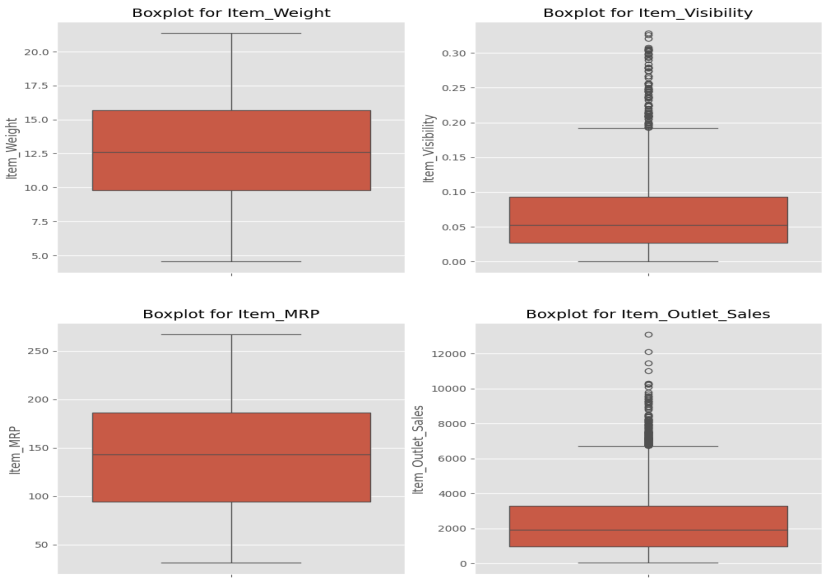

Figure 4. Boxplot for numerical features

1.5 Bivariate Analysis

1.5.1. Categorical Feature vs Outlet Sales Boxplot

The following diagram shows a boxplot of all our categorical data in our dataset versus our outlet sales using boxplot from seaborn. We can examine how the changes in the categorical feature values have an impact on the median, interquartile range, and outlier concentration. This can give us an idea about which categorical features might have a greater impact on our target feature.

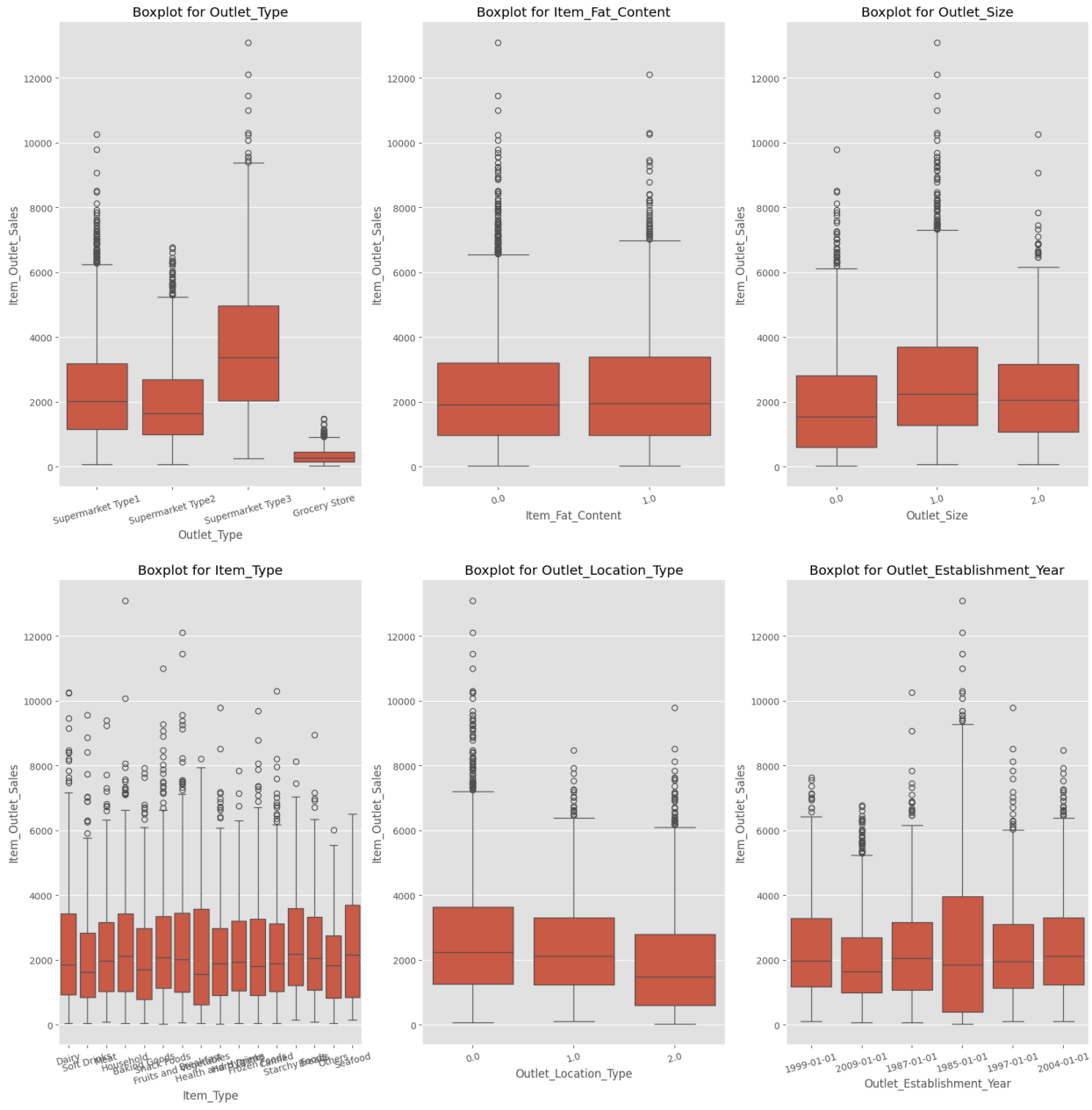

Figure 5. Box Plot for Categorical Features Versus Outlet Sales

1.6. Multivariate Analysis

1.6.1 Pairplot for numeric features

The diagram below presents a pairplot showcasing all the numerical variables in our dataset. This visualization allows us to assess the linearity of our dataset. The pairplot uses outlet type as the hue since this categorical feature displayed a good deal of impact on the target feature.

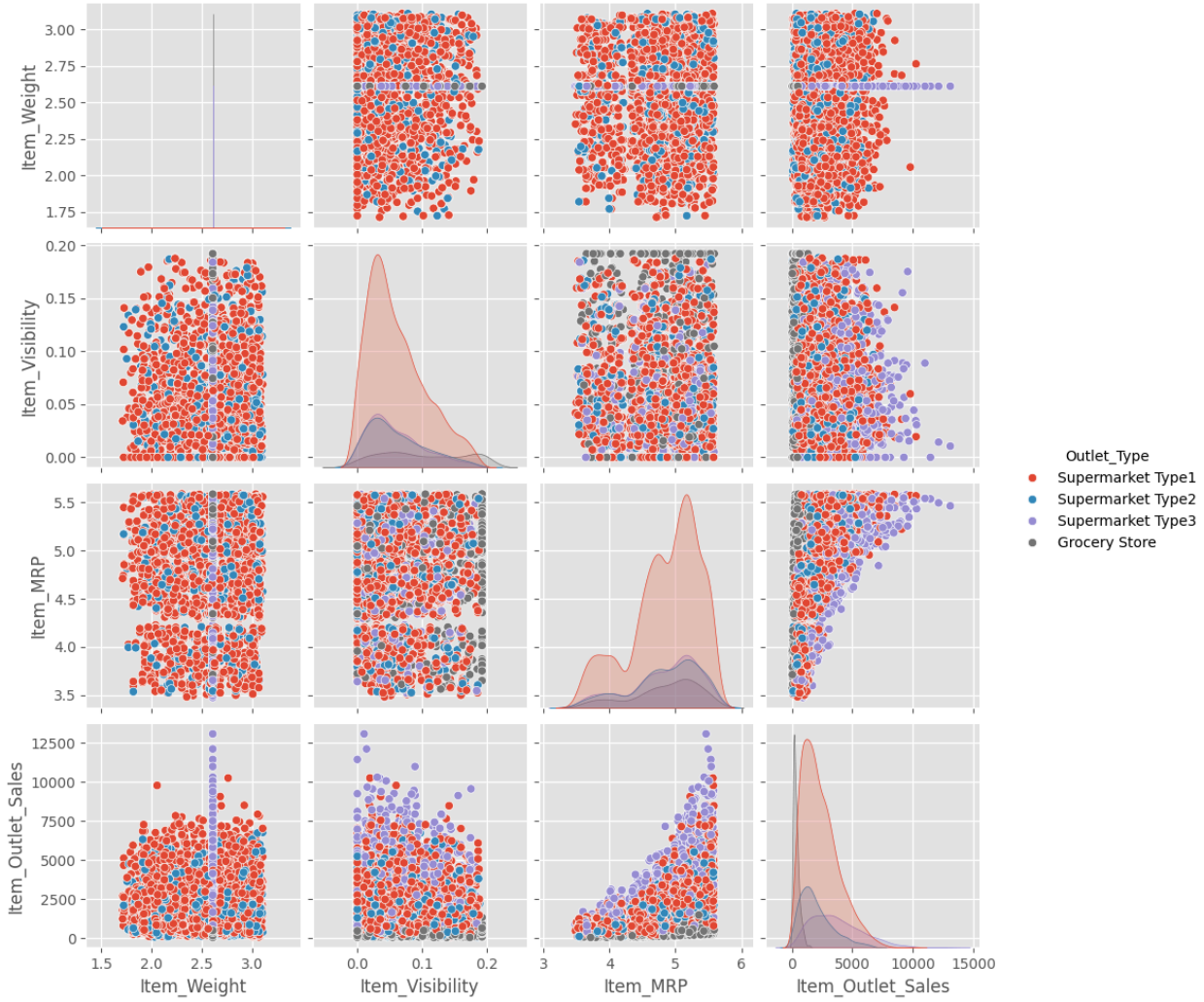

Figure 6. Pairplot for numerical Features

1.6.2 Feature Correlation

The following diagram shows a correlation heatmap which helps us determine the multicollinearity in our dataset. It also gives us an idea about the correlation between our dependent and independent features.

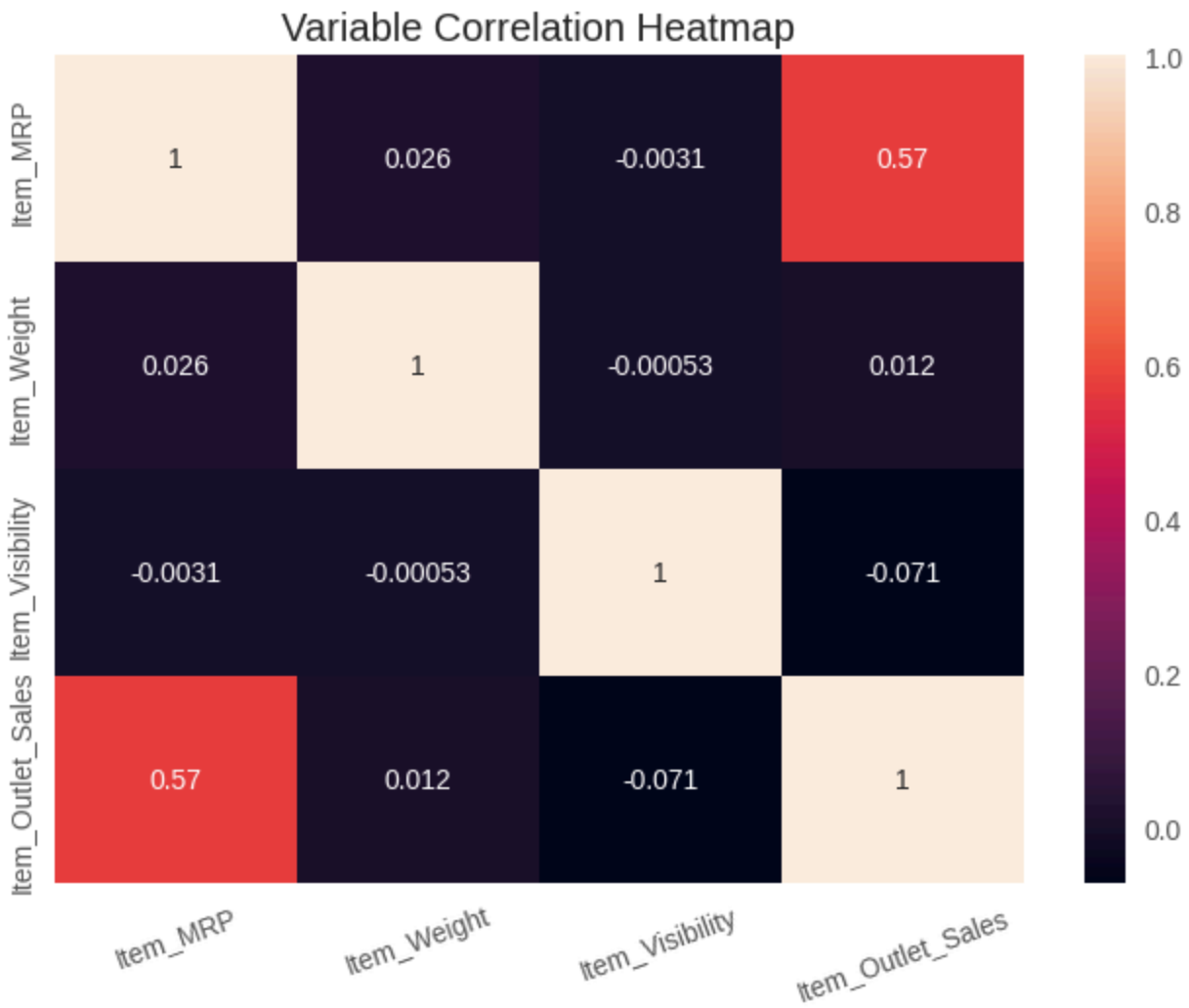

Figure 7. Correlation Heatmap

2. Feature Engineering and Data Cleaning

The subsequent action are taken in the following steps:

- Checking dataset for null and duplicate values. Filling missing values in ‘Item_Weight’ with median of the feature
- Fixing “Item_Fat_Content” as same values are recorded with different notations
- Ordinal Encoded our ordinal features i.e. 'Outlet_Size', 'Outlet_Location_Type' and 'Item_Fat_Content'
- Capped outliers in “Item_Visibility” using Interquartile range
- Normalized our numeric features using function transformers

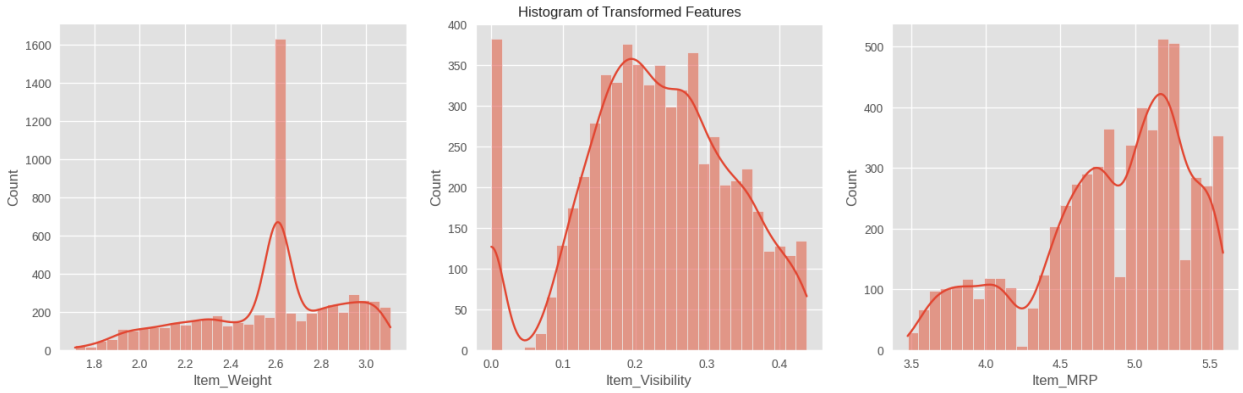

Figure 8. Histogram for normalized numerical Features

- Grouped all values except the top three frequent ones as "other" and one-hot encoded 'Item_Type' feature

3. Approach

I prepare the data from the EDA for machine learning model development by creating a train and test split by 70% and 30% respectively.

3.1 Machine Learning Approach

In this approach, we use three machine learning algorithms:

- Linear Regression
- Pycaret
- Decision Tree Model
- Random Forest Model
- Xgboost Model
- CatBoosting
- Stacking with Light Gradient Boosting Machine (LGBM)
- Blending with LGBM

3.1.1 Linear Regression

Results & Metrics : The following diagram shows the Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2-Score and Adjusted R2-Score.

Metrics Explained

Term Explanation

- **MAE**: The Mean Absolute Error represents the average of the absolute differences between the values predicted by the model (ŷ) and the actual values(y).

$$MAE = \frac{\sum_{i=1}^{n} |yi - \hat{y}i|}{n}$$

where,
MAE = mean absolute error
yi = prediction
ŷi = true value
n = Total number of data points

- **MSE :** Mean Absolute Error (MAE) measures the average discrepancy between the predicted values and the actual values but instead of taking the absolute differences, MSE computes the square of the differences between predicted and actual values.

$$MSE = \frac{\sum_{i=1}^{n} (yi - \hat{y}i)^2}{n}$$

- **RMSE :** Root Mean Squared Error (RMSE) is derived from Mean Squared Error (MSE) by taking the square root of the average squared differences between predicted and actual values. RMSE is advantageous because it represents the typical magnitude of the errors in the same units as the target variable, making it easier to interpret.

$$\mathrm{R}MSE = \sqrt{\frac{\sum_{i=1}^{n} (yi - \hat{y}i)^2}{n}}$$

- **R2 Score :** The R2 (R-squared) score, also known as the coefficient of determination, is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables in a regression model. In simpler terms, it assesses the goodness-of-fit of a regression model by comparing the variance of the model's predictions to the variance of the actual data.

$$R^2\ Score = 1 - \frac{RSS}{TSS}$$

where,
R^2 = Coefficient of determination
RSS = Sum of Square of residuals
TSS = Total Sum of Square

- **Adjusted R2 Score :** The Adjusted R2 Score is a modification of the coefficient of determination (R2 score) that adjusts for the number of predictors in a regression model.

While R2 measures the proportion of the variance in the dependent variable explained by the independent variables, the adjusted R2 takes into account the number of predictors and penalizes the addition of unnecessary predictors that do not significantly improve the model's fit.

$$Adjusted\ R^2 = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

where ,
R2 = The coefficient of determination
n = The Number of Observations
k = The Number of Predictors (independent variables) in the Model

```
MAE : 1058.6969328964144
MSE : 1944194.087416401
RMSE : 1394.3436045022765
R2 Score : 0.3862755022899875
Adjusted R2 Score : 0.38290893894544553
```

Model Performance:

- **MAE :** The MAE for this model is 1058.69 which means on average, the model's predictions are off by approximately 1058.69 when compared to the actual values of the target variable.
- **MSE :** The MSE for this model is 1944194.08 which means on average, the squared differences between the predicted and actual values of the target variable amount to approximately 1944194.08.
- **RMSE :** The RMSE for this model is 1394.34 suggests that, on average, the differences between the predicted and actual values of the target variable amount to approximately 1394.34 units.
- **R2 Score :** The R2 score for this model is 0.386 which means that approximately 38% of the variance in the dependent variable is explained by the independent variables included in the model. The remaining 62% of the variance is unexplained and may be attributed to other factors not included in the model or to random noise.
- **Adjusted R2 Score :** The adjusted R2 score for the model is 0.382 is almost the same as the R2 score but it penalizes the addition of unnecessary predictors that do not significantly improve the model's fit.

Homoscedasticity

The following diagram plots the scatter plots of the residuals to check if the nature of the error is homoscedastic or heteroscedastic. As we can see in the diagram below, the errors move farther from the regression line as the magnitude of x (predicted values) increases. So we can see that our model is plagued by heteroscedasticity.

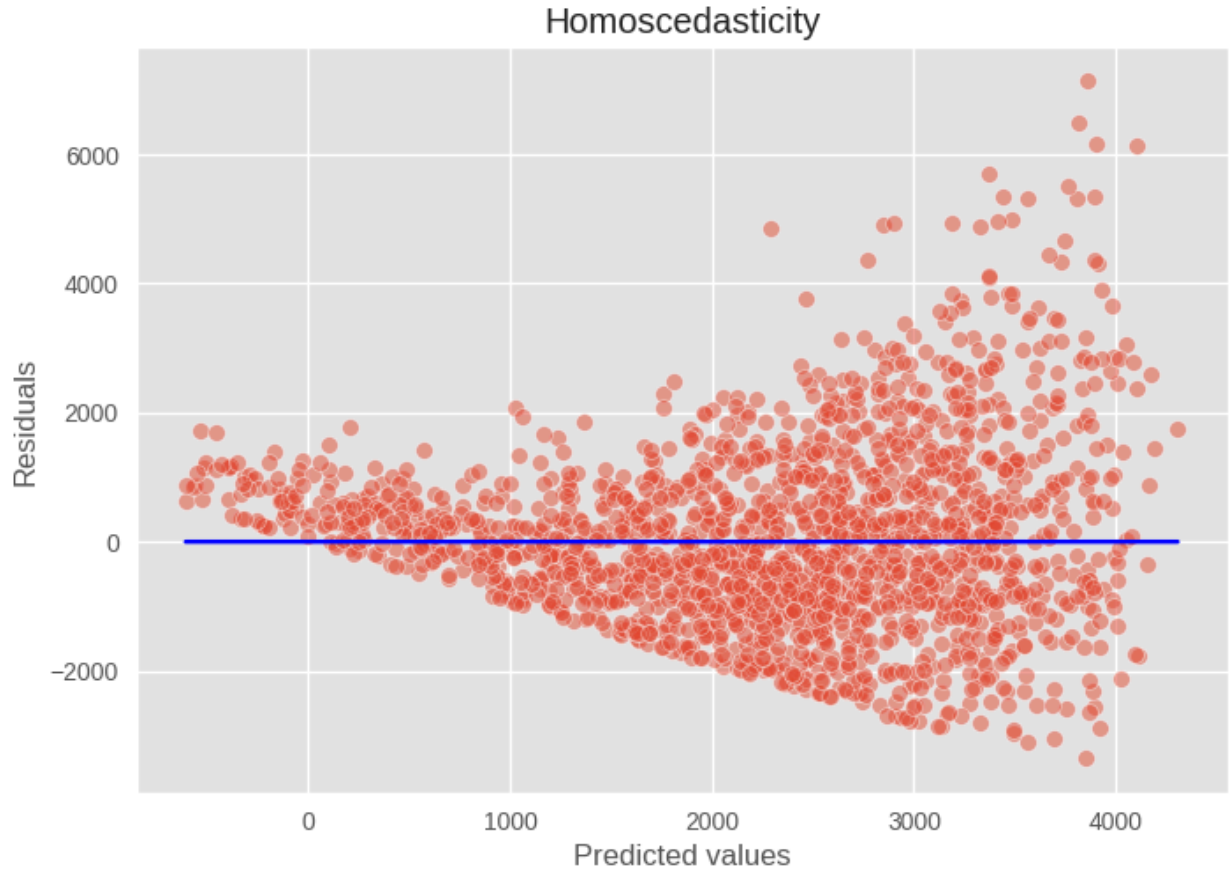

Figure 9. Homoscedasticity Check

3.1.2 Running Pycaret

Since our dataset did not satisfy the assumptions of linear regression, I used Pycaret to test for the best fit model. As seen in the diagram below the metrics for Light Gradient Boosting Machine (LGBM) is the best.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	838.7587	1404373.8479	1183.8681	0.5309	0.5920	0.6259	
gbr	Gradient Boosting Regressor	866.4004	1407607.6288	1185.5331	0.5302	0.6460	0.7407	
rf	Random Forest Regressor	851.8827	1450058.6446	1203.4219	0.5150	0.5838	0.6208	
et	Extra Trees Regressor	887.0168	1575200.7142	1253.5296	0.4730	0.5994	0.6380	
xgboost	Extreme Gradient Boosting	905.9642	1630662.6250	1276.1396	0.4539	0.6345	0.6524	
knn	K Neighbors Regressor	960.9033	1771268.3375	1329.5954	0.4081	0.6908	0.7763	
ada	AdaBoost Regressor	1052.7523	1807080.7222	1343.7401	0.3956	0.8515	1.3467	
llar	Lasso Least Angle Regression	1045.5596	1906621.0125	1379.6338	0.3647	0.8587	1.1353	
ridge	Ridge Regression	1045.9785	1906768.8125	1379.6895	0.3647	0.8610	1.1357	
lasso	Lasso Regression	1045.5600	1906621.1500	1379.6339	0.3647	0.8587	1.1353	
lr	Linear Regression	1046.2225	1906697.3500	1379.6636	0.3647	0.8622	1.1367	
br	Bayesian Ridge	1045.8778	1907334.6625	1379.8947	0.3645	0.8631	1.1350	
lar	Least Angle Regression	1046.6376	1907958.9875	1380.1407	0.3642	0.8614	1.1366	
huber	Huber Regressor	1032.0841	1935874.0464	1390.0319	0.3553	0.8398	1.0663	
par	Passive Aggressive Regressor	1042.1218	2026797.6897	1422.3396	0.3247	0.8261	1.0548	
omp	Orthogonal Matching Pursuit	1058.7836	2037638.3375	1426.0871	0.3212	0.9320	1.2142	
en	Elastic Net	1182.4139	2344815.2500	1529.9960	0.2188	0.9121	1.5059	
dt	Decision Tree Regressor	1157.7945	2711359.1976	1645.2895	0.0938	0.7772	0.7422	
dummy	Dummy Regressor	1367.4982	3005910.1000	1732.6507	-0.0021	1.0139	1.9082	

Figure 10. Pycaret best fit model

3.1.3 Decision Tree Regressor

The following are the model performance metrics for the Decision Tree Regressor model:

```
MAE : 1115.5620582333697
MSE : 2523099.1197276604
RMSE : 1588.4266176716067
R2 Score : 0.20353232737931726
Adjusted R2 Score : 0.19916333301496902
```

3.1.4 Random Forest Regressor

The following are the model performance metrics for the Random Forest Regressor model:

```
MAE : 823.4308400643403
MSE : 1376800.764089129
RMSE : 1173.371537105417
R2 Score : 0.5653847715840792
Adjusted R2 Score : 0.5630007056026425
```

3.1.5 XGBoost

The following are the model performance metrics for the XGBoost model:

```
MAE : 858.8957547025992
MSE : 1398689.8415097597
RMSE : 1182.6621840194941
R2 Score : 0.5584750380691684
Adjusted R2 Score : 0.5560530689965912
```

3.1.6 CatBoost

The following are the model performance metrics for the CatBoost model:

```
MAE : 821.1539247580611
MSE : 1366253.9853378544
RMSE : 1168.8686775416024
R2 Score : 0.5687140773017949
Adjusted R2 Score : 0.5663482741054251
```

3.1.7 Stacking with Light Gradient Boosting Machine (LGBM)

The following are the model performance metrics for the Meta-model trained using Light Gradient Boosting Machine (LGBM) :

```
MAE : 828.5726722873355
MSE : 1405903.3064548166
RMAE : 28.784938288753295
R2 Score : 0.5561979608067656
Adjusted R2 Score : 0.5537635009099294
```

3.1.8 Blending with Light Gradient Boosting Machine (LGBM)

The blending method uses CatBoost, XGBoost and Random Forest Regressor as the base model used to train the meta model. The meta model is trained using Light Gradient Boosting Machine and the model performance metrics for the Meta-model is shown below:

```
MAE : 875.1956184220188
MSE : 1526182.814465849
RMSE : 1235.3877182754607
R2 Score : 0.5182292821050554
Adjusted R2 Score : 0.515586546406235
```