Big Mart Sales Prediction

Name: Akhilesh Kumar Shah

@Feynn Lab ML Intern

Date: 29/06/2024

Abstract

Nowadays, shopping malls and big marts meticulously track their sales data for each individual item to predict future customer demand and update inventory management accordingly. Sales forecasts, based on data from various Big Mart outlets, allow businesses to adjust their models to align with expected outcomes. This data can be utilized to predict potential sales volumes for retailers such as Big Mart through various machine learning methods. The proposed system should consider factors such as price, outlet, and outlet location. Several machine learning algorithms, including linear regression, decision tree algorithms, and XGBoost regressor, provide efficient sales predictions based on gradient boosting. Finally, hyperparameter tuning is employed to select the most relevant hyperparameters, enhancing the algorithm's performance and accuracy.

Keywords: Machine Learning, Sales Prediction, Big Mart, Random Forest, Linear Regression

Introduction

Every item in shopping centers and big marts is tracked to anticipate future customer demand and improve inventory management. Big Mart, a vast network of stores worldwide, relies heavily on analyzing trends to predict potential sales. Data scientists evaluate these trends by product and store to create accurate forecasts. Using machine learning to predict Big Mart transactions helps test various patterns to achieve precise results.

Many companies depend on this knowledge base and need to forecast market patterns. Each shopping center or store aims to attract more customers daily, which helps evaluate sales volumes for inventory management, logistics, and transportation. To address sales prediction based on customer demand across different Big Marts, various machine learning algorithms like Linear Regression, Random Forest, Decision Tree, Ridge Regression, and XGBoost are used.

Sales predictions consider factors such as store type, population around the store, the city where the store is located (urban or rural), and other aspects like store capacity. Accurate sales forecasts are crucial for retail centers because they help develop and enhance business strategies, increasing market awareness.

Problem Statement

To understand how certain properties of an item affect their sales and gain a comprehensive understanding of Big Mart sales, a predictive model can be built. This model will help identify the key factors that influence sales at each store and suggest changes to the product or store characteristics to increase sales. This approach will enable Big Mart to optimize their strategies for better sales outcomes.

Market/Business/Customer Need Assessment

Price analysis is the study of the prices of products and services on the market to improve the profitability of e-commerce itself. It allows to know and understand ow prices affect the growth of certain businesses and its influence on the sales volume. From this knowledge, companies can apply appropriate price optimization to increase their profits. Price analysis can be carried out with an automated pricing tool that collects the data of greatest interest to the company. we explain its benefits and what you should consider when performing price analysis.

As a starting point, you should know that price analysis can be applied both routinely, to evaluate the profitability of your pricing strategy periodically, and at certain key moments for e-commerce. Among these moments are the evaluation of new product ideas, the launch of new products and services, or the adjustment of the positioning strategy of a product against those of the computation.

Target Specifications and Characterization

- Increasing annual sales and profit
- Increasing customer numbers
- Increasing upsells and cross-sells
- Improving customer retention
- Increasing conversion rates
- Increasing sales rep productivity
- Cutting the time sales reps spend on non-sales tasks

Enhancing your sales processes and sales activities.

External Search

I used the online dataset from Kaggle:

- Data set Link: https://www.kaggle.com/datasets/shivan118/big-mart-salesprediction-datasets

Relevant articles Link:

- https://www.analyticsvidhya.com/blog/2016/02/bigmart-sales-solutiontop-20/
- https://www.researchgate.net/publication/340252000_A_Comparative_Study_of_Big_Mart_Sales_Prediction
- https://medium.com/analytics-vidhya/bigmart-dataset-sales-predictionc1f1cdca9af1

Benchmarking

(Fawcett, Tom and Foster J. Provost) This study describes the method of identifying suspicious behavior using an automated prototype. To develop this prototype, various machine learning methods were employed. Data mining and constructive induction approaches were used to uncover the disparities in cell phone owners' behavior.

(Demchenko et al.) To forecast sales, a generic linear method, a decision tree approach, and a gradient boosting method were used. The initial data set contained a large number of entries, but the final data set used for analysis was significantly reduced after removing non-usable data, duplicate entries, and irrelevant sales data.

(Ragg et al.) This study shows that many vendors would benefit from forecasting a single transaction rate, suggesting that the collected knowledge could be useful for designing a system that predicts multiple outcomes. A neural network technique was used for prediction, and Bayesian learning provided additional insights.

(Armstrong J) Three modules—Hive, R programming, and Tableau—were used to forecast sales. By analyzing the store's past data, a better understanding of revenue can be achieved, allowing for more effective adjustments to objectives. Key values are extracted within the diagram to reduce all intermediate values by lowering the intermediate key feature.

Applicable Regulations

The patents mentioned above might claim the technology used if the algorithms are not developed and optimised individually and for our requirements. Using a pre-existing model is off the table if it incurs a patent claim.

- Must provide access to the third-party websites to audit and monitor the authenticity and behaviour of the service.
- Enabling open-source, academic and research community to audit the algorithms and research on the efficacy of the product.
- Laws controlling data collection: Some websites might have a policy against collecting customer data in form of reviews and ratings.
- Must be responsible with the scraped data: it is quintessential to protect the privacy and intention with which the data was extracted.

Applicable Constraint

- Continuous data collection and maintenance
- Lack of technical knowledge for the user
- Taking care of rarely bought products

Business Model

Sales Prediction is vital for any company's success. Sales forecast provides insight into how much revenue the concerned organization will generate. In our uncertain times, forecasting revenue has become an even more challenging job with distributed timelines and business and entire growth strategies in shambles. Sales assumptions are paramount in mapping and planning ahead and really affect the organization. Predicating revenue is not easy, but it is also very important to make strategic decisions for predictable revenue.

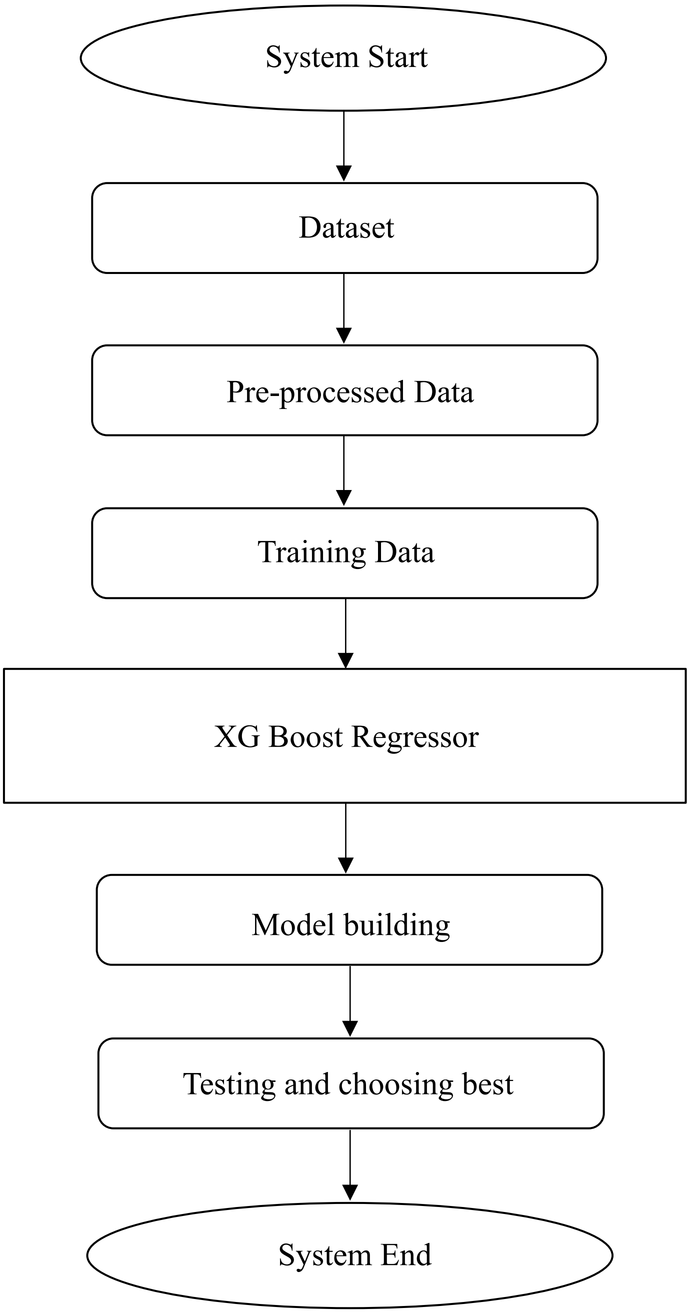
Before processing to top techniques that help with sales forecast, check out some of the free courses on sales management, sales conversion and many more on great learning academy.

Concept Generation

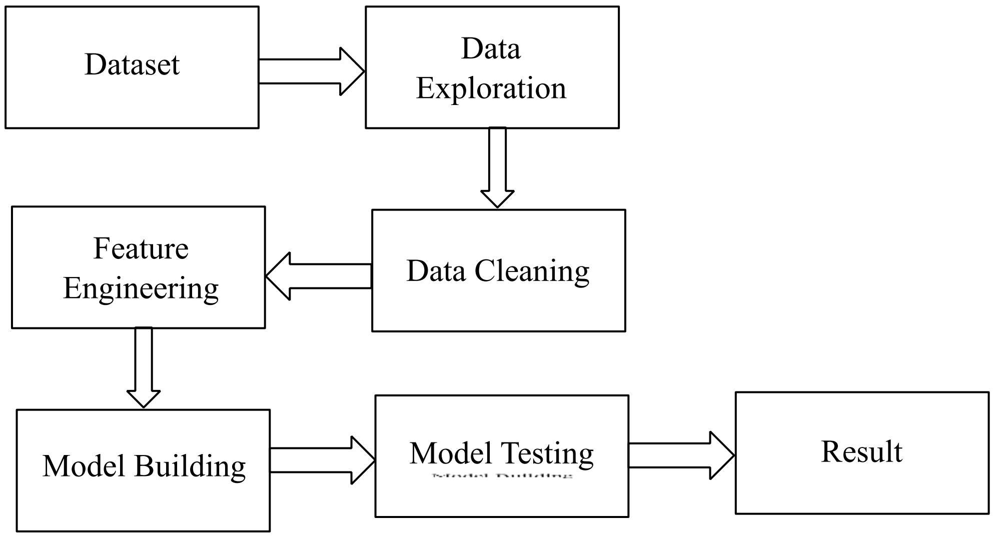
This product requires the tool of machine learning models to be written from scratch in order to suit our needs. Tweaking these models for our use is less daunting than coding it up from scratch. A well-trained model can either be repurposed or built. But building a model with the resources and data we have is dilatory but possible. The customer might want to spend the least amount of time giving input data. This accuracy will take a little effort to nail, because it's imprudent to purely on Classic Machine Learning algorithm.

Final Product Prototype

System Architecture:



Proposed System:



Product Details

How does it work?

- To predict the future sales from data of the previous year's using Machine Learning Techniques.
- To conclude the best model which is more efficient and gives fast and accurate result by using XG Boost Regressor.
- To find out key factors that can increase their sales and what changes could be made to the productor store's characteristics.

Data Source:

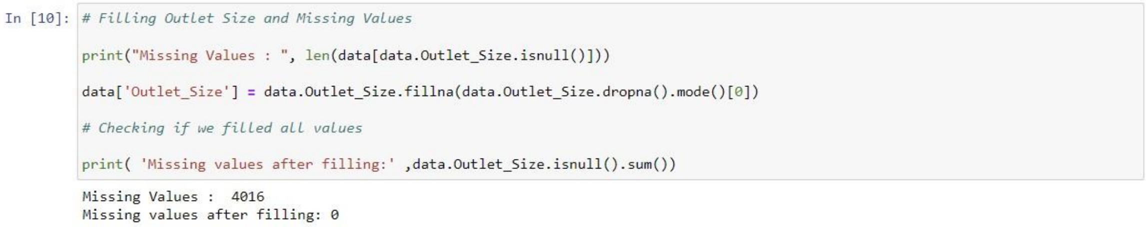
https://www.kaggle.com/datasets/shivan118/big-mart-sales-prediction-datasets

Algorithm needed:

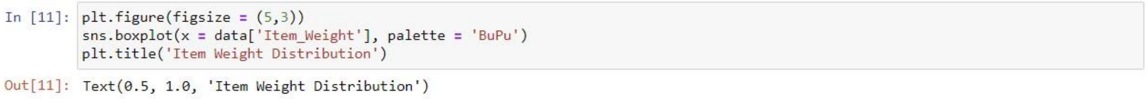
- Linear Regression
- Decision Tree
- Random Forest
- XGBoost

Code Implementation

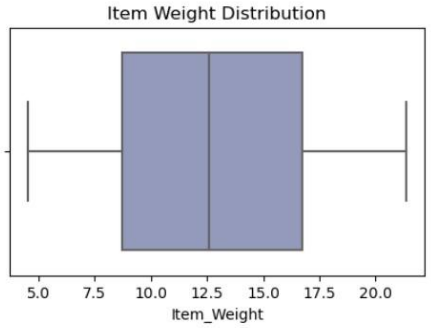
Some Basic Visualizations on Real World or Augmented Data:

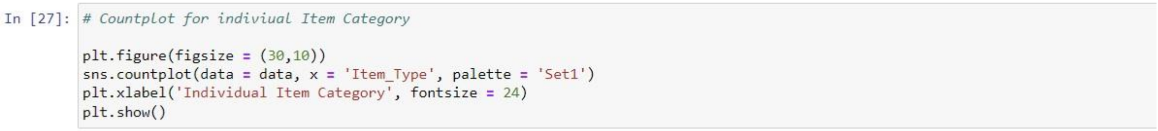


```
In [10]: # Filling Outlet Size and Missing Values

         print("Missing Values : ", len(data[data.Outlet_Size.isnull()]))

         data['Outlet_Size'] = data.Outlet_Size.fillna(data.Outlet_Size.dropna().mode()[0])

         # Checking if we filled all values

         print( 'Missing values after filling:' ,data.Outlet_Size.isnull().sum())

         Missing Values :  4016
         Missing values after filling: 0
```

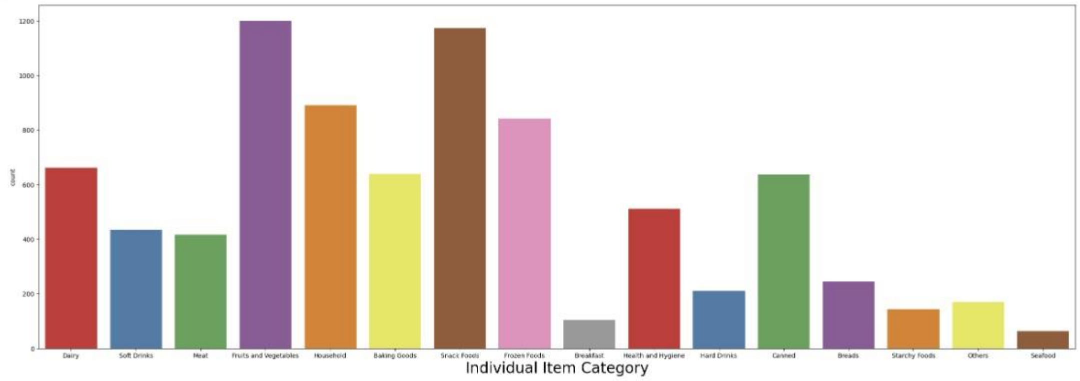


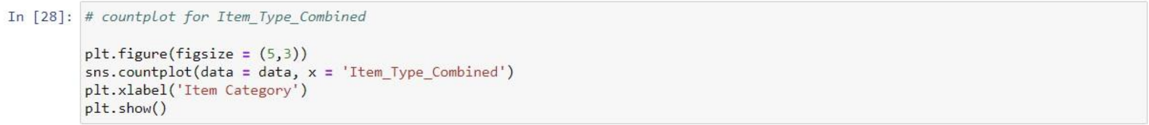
```
In [11]: plt.figure(figsize = (5,3))
         sns.boxplot(x = data['Item_Weight'], palette = 'BuPu')
         plt.title('Item Weight Distribution')

Out[11]: Text(0.5, 1.0, 'Item Weight Distribution')
```



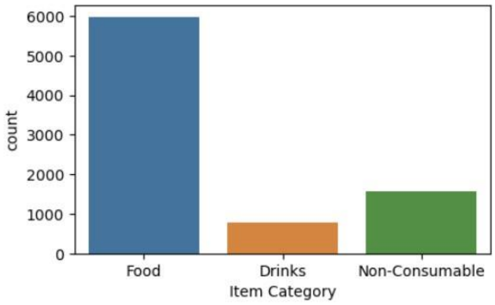


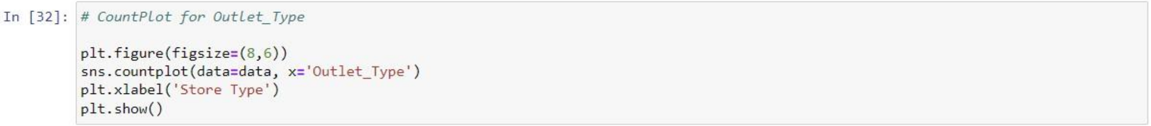
```
In [27]: # Countplot for indiviual Item Category

         plt.figure(figsize = (30,10))
         sns.countplot(data = data, x = 'Item_Type', palette = 'Set1')
         plt.xlabel('Individual Item Category', fontsize = 24)
         plt.show()
```



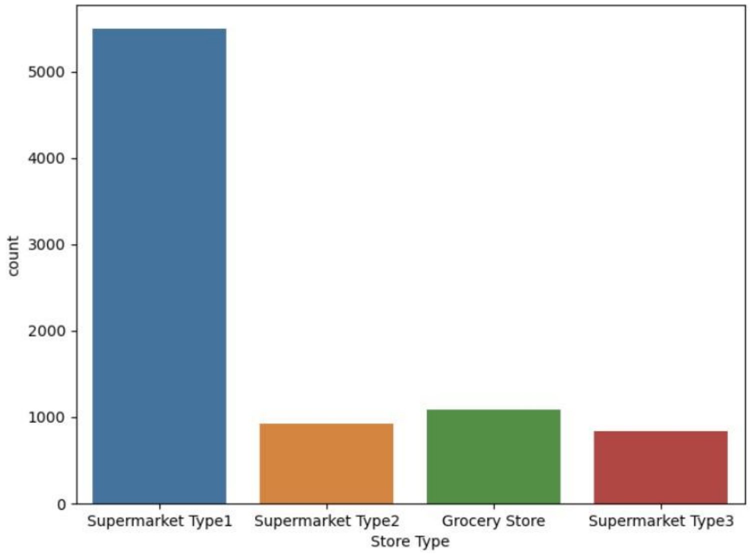


```
In [28]: # countplot for Item_Type_Combined

         plt.figure(figsize = (5,3))
         sns.countplot(data = data, x = 'Item_Type_Combined')
         plt.xlabel('Item Category')
         plt.show()
```

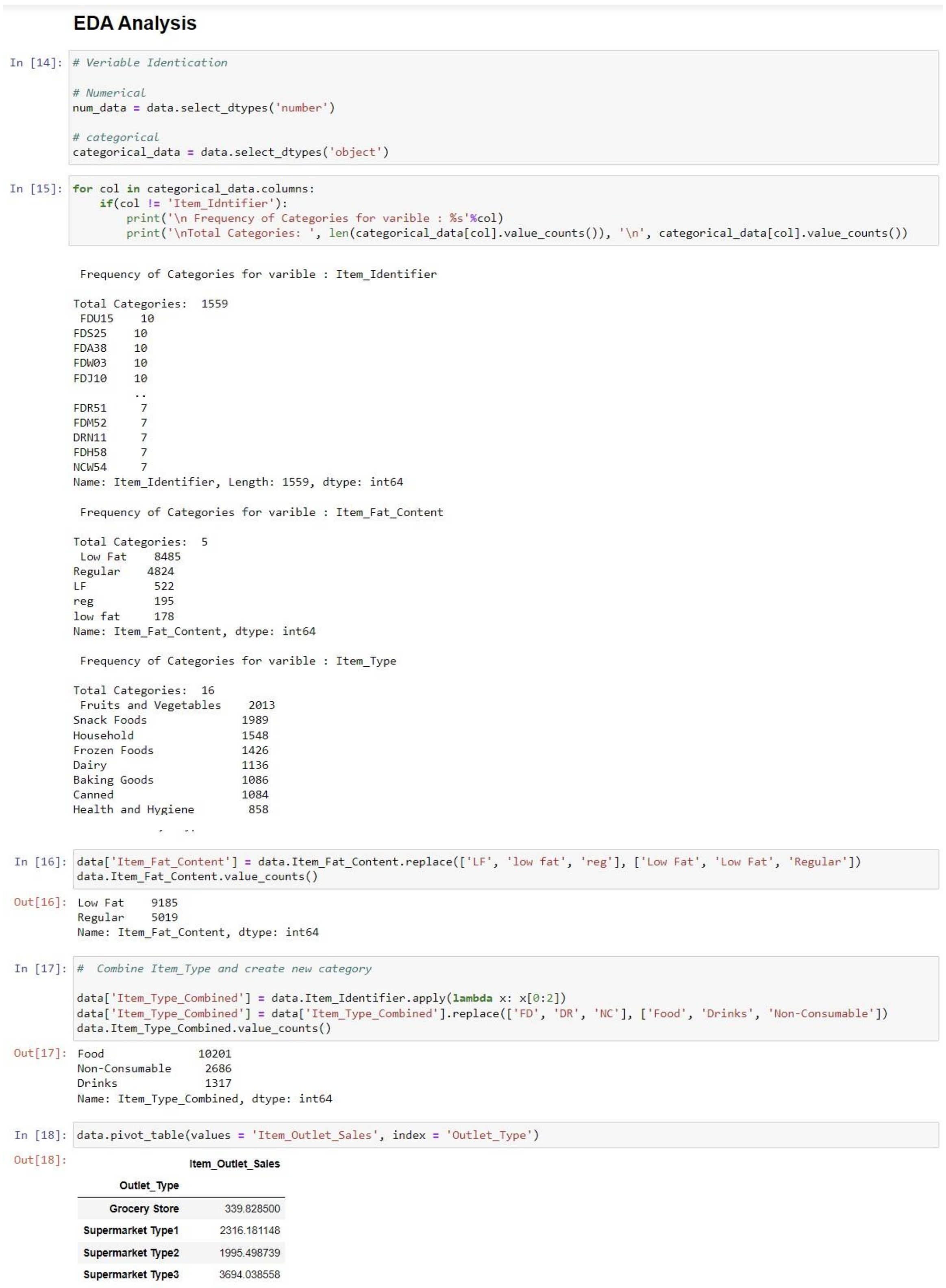




```
In [32]: # CountPlot for Outlet_Type

         plt.figure(figsize=(8,6))
         sns.countplot(data=data, x='Outlet_Type')
         plt.xlabel('Store Type')
         plt.show()
```



Simple EDA:



EDA Analysis

```
In [14]: # Veriable Identication

         # Numerical
         num_data = data.select_dtypes('number')

         # categorical
         categorical_data = data.select_dtypes('object')
```

```
In [15]: for col in categorical_data.columns:
             if(col != 'Item_Idntifier'):
                 print('\n Frequency of Categories for varible : %s'%col)
                 print('\nTotal Categories: ', len(categorical_data[col].value_counts()), '\n', categorical_data[col].value_counts())
```

```
 Frequency of Categories for varible : Item_Identifier

Total Categories:  1559
 FDU15    10
FDS25    10
FDA38    10
FDW03    10
FDJ10    10
         ..
FDR51     7
FDM52     7
DRN11     7
FDH58     7
NCW54     7
Name: Item_Identifier, Length: 1559, dtype: int64

 Frequency of Categories for varible : Item_Fat_Content

Total Categories:  5
 Low Fat    8485
Regular    4824
LF          522
reg         195
low fat     178
Name: Item_Fat_Content, dtype: int64

 Frequency of Categories for varible : Item_Type

Total Categories:  16
 Fruits and Vegetables    2013
Snack Foods              1989
Household                1548
Frozen Foods             1426
Dairy                    1136
Baking Goods             1086
Canned                   1084
Health and Hygiene        858
```

```
In [16]: data['Item_Fat_Content'] = data.Item_Fat_Content.replace(['LF', 'low fat', 'reg'], ['Low Fat', 'Low Fat', 'Regular'])
         data.Item_Fat_Content.value_counts()

Out[16]: Low Fat    9185
         Regular    5019
         Name: Item_Fat_Content, dtype: int64
```

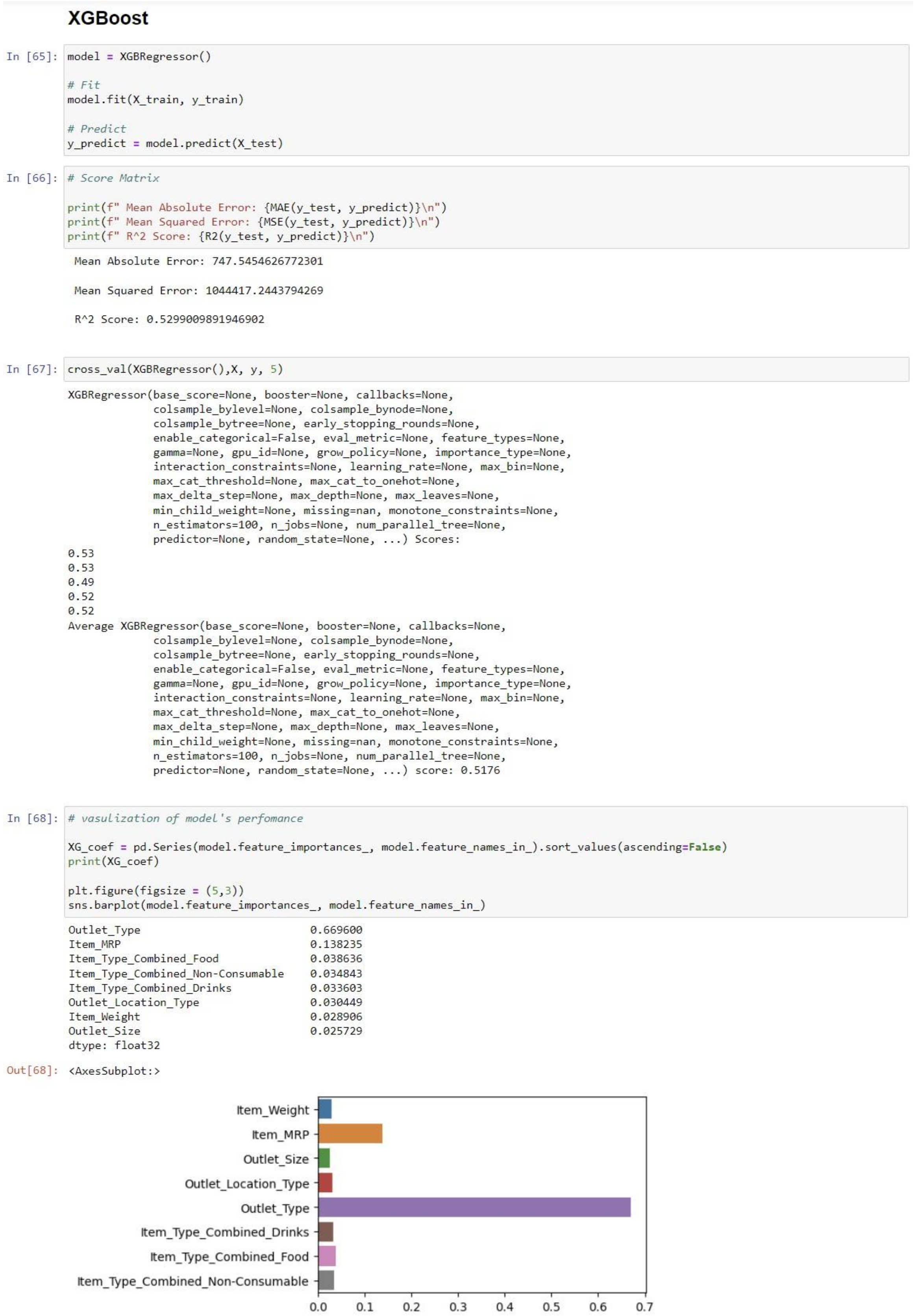
```
In [17]: #  Combine Item_Type and create new category

         data['Item_Type_Combined'] = data.Item_Identifier.apply(lambda x: x[0:2])
         data['Item_Type_Combined'] = data['Item_Type_Combined'].replace(['FD', 'DR', 'NC'], ['Food', 'Drinks', 'Non-Consumable'])
         data.Item_Type_Combined.value_counts()

Out[17]: Food              10201
         Non-Consumable     2686
         Drinks             1317
         Name: Item_Type_Combined, dtype: int64
```

```
In [18]: data.pivot_table(values = 'Item_Outlet_Sales', index = 'Outlet_Type')
```

Out[18]:

	Item_Outlet_Sales
Outlet_Type	
Grocery Store	339.828500
Supermarket Type1	2316.181148
Supermarket Type2	1995.498739
Supermarket Type3	3694.038558

ML Model:



XGBoost

```
In [65]: model = XGBRegressor()

         # Fit
         model.fit(X_train, y_train)

         # Predict
         y_predict = model.predict(X_test)
```

```
In [66]: # Score Matrix

         print(f" Mean Absolute Error: {MAE(y_test, y_predict)}\n")
         print(f" Mean Squared Error: {MSE(y_test, y_predict)}\n")
         print(f" R^2 Score: {R2(y_test, y_predict)}\n")
```

```
 Mean Absolute Error: 747.5454626772301

 Mean Squared Error: 1044417.2443794269

 R^2 Score: 0.5299009891946902
```

```
In [67]: cross_val(XGBRegressor(),X, y, 5)
```

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...) Scores:
0.53
0.53
0.49
0.52
0.52
Average XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...) score: 0.5176
```

```
In [68]: # vasulization of model's perfomance

         XG_coef = pd.Series(model.feature_importances_, model.feature_names_in_).sort_values(ascending=False)
         print(XG_coef)

         plt.figure(figsize = (5,3))
         sns.barplot(model.feature_importances_, model.feature_names_in_)
```

```
Outlet_Type                         0.669600
Item_MRP                            0.138235
Item_Type_Combined_Food             0.038636
Item_Type_Combined_Non-Consumable   0.034843
Item_Type_Combined_Drinks           0.033603
Outlet_Location_Type                0.030449
Item_Weight                         0.028906
Outlet_Size                         0.025729
dtype: float32
```

```
Out[68]: <AxesSubplot:>
```

GitHub Link: https://github.com/Akhushah/Feynn_Labs_Internship_2024

Conclusion

In this project, the basics of machine learning and the associated data processing and modeling algorithms are described, followed by their application for sales prediction in Big Mart shopping centers at various locations. Upon implementation, the prediction results demonstrate the correlation among different attributes and highlight how a particular medium-sized location recorded the highest sales. This suggests that other shopping locations could improve sales by following similar patterns.

Utilizing multiple parameters and various factors can make sales prediction more innovative and successful. Accuracy, which is crucial in prediction-based systems, can be significantly enhanced by increasing the number of parameters used. Additionally, examining how the sub-models operate can further increase the system's productivity.

Reference

- Smola, A., & Vishwanathan, S. V. N. (2008). Introduction to machine learning. Cambridge University, UK, 32, 34.

- Kumari Punam, Rajendra Pamula, Praphula Kumar Jain (2018), A Two-Leval Statistical Model for Big Mart Sales Prediction. (https://ieeexplore.ieee.org/document/8675060)

- Gopal Behere, Neeta Nain (2019). Grid Search Optimization (GSO) Based Future Sales Prediction for Big Mart. 2019 International Conference on Signal-Image Technology & Internet-Based Systems (SITIS).

- Das, P., Chaudhury, S.: Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data (2007)
- Kadam, H., Shevade, R., Ketkar, P. and Rajguru.: “A Forecast for Big Mart Sales Based on Random Forests and Multiple Linear Regression.” (2018).
- Pavan Chatradi, Meghana, Avinash Chakravarthy V, Sai Mythri Kalavala, Mrs.Neetha KS (2020), Improvizing Big Market Sales Prediction, Volume 12 Issue 4 (https://www.xajzkjdx.cn/gallery/423-april2020.pdf)