Big Mart

Sales Prediction

Name: Padmani Gayatri Ashokbhai

Date: 07/02/2023

*"AI is probably the most important thing humanity as ever worked on.
I think of it as something more profound than electricity or fire."*

~ Sunder Pichai, CEO of Google

Abstract

Now days shopping malls and Big Marts keep the track of their sales data of each and every individual item for predicting future demand of the customer and update the inventory management as well. The sales forecast is based on Big Mart sales for various outlets to adjust the business model to expected outcomes. The resulting data can then be used to prediction potential sales volumes for retailers such as Big Mart through various machine learning methods. The estimate of the system proposed should take account of price tag, outlet and outlet location. A number of networks use the various machine- learning algorithms, such as linear regression and decision tree algorithms, and XGBoost regressor, which offers an efficient prevision of Big Mart sales based on gradient. At last, hyperparameter tuning is used to help you to choose relevant hyperparameters that make the algorithm Shine and produce the highest accuracy.

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

Introduction

Every item is tracked for its shopping canters and Big Marts in order to anticipate a future demand of the customer and also improve the management of its inventory. Big Mart is an immense network of shops virtually all over the world. Trends in Big Mart are very relevant and data scientists evaluate those trends per product and store in order to create potential centres. Using the machine to forecast the transactions of Big Mart helps data scientists to test the various patterns by store and product to achieve the correct results. Many companies rely heavily on the knowledge base and need market patterns to be forecasted. Each shopping canter or store endeavours to give the individual and present moment proprietor to draw in more clients relying upon the day, with the goal that the business volume for everything can be evaluated for organization stock administration, logistics and transportation administration, and so forth. To address the issue of deals expectation of things dependent on client's future requests in various BigMarts across different areas diverse Machine Learning algorithms like Linear Regression, Random Forest, Decision Tree, Ridge Regression, XGBoost are utilized for gauging of deals volume. Deals foresee the outcome as deals rely upon the sort of store, populace around the store, a city wherein the store is located, i.e., it is possible that it is in an urban zone or country. Population statistics around the store also affect sales, and the capacity of the store and many more things should be considered. Because every business has strong demand, sales forecasts play a significant part in a retail centre. A stronger prediction is always helpful in developing and enhancing corporate market strategies, which also help to increase awareness of the market.

Problem Statement

To find out what role's certain properties of an item play and how they affect their sales big understanding Big Mart sales. In order to help Big Mart, achieve this goal, a predictive model can be built to find out for every store, the key factor that can increases their sales and what changes could be made to the product or store's characteristics.

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-sales-prediction-datasets

Relevant articles Link:

- https://www.analyticsvidhya.com/blog/2016/02/bigmart-sales-solution-top-20/
- https://www.researchgate.net/publication/340252000_A_Comparative_Study_of_Big_Mart_Sales_Prediction
- https://medium.com/analytics-vidhya/bigmart-dataset-sales-prediction-c1f1cdca9af1

Benchmarking

(Fawcett, Tom and Foster J. Provost) The method of identifying suspicious behaviour using an automated prototype is described in this study. For the purpose of completing this acceptable prototype, many machine learning methods were used. Here, data mining and constructive induction approaches are used to uncover the disparity in cell phone owners' behaviour.

(Demchenko et al.) To forecast sales, a generic linear method, a decision tree approach, and a decent gradient approach were employed. The original data set evaluated included a large number of entries, but the final data set utilized for analysis was significantly less than the original since it included non-usable data, duplicate entries, and unimportant sales data.

(Ragg et al.) Many vendors would profit from the forecast of a single transaction rate, as shown in this study, which implies the knowledge collected may be useful for the design of a set-up that would predict a large number of results. The neural network technique is used to make the prediction. They used Bayesian learning to acquire insights in this situation.

(Armstrong J) Three modules, hive, R programming, and tableau, were used to forecast sales. By looking at the store's past, you may have a better knowledge of the income and make changes to the objective to make it more successful. To achieve the findings, key values are retrieved inside the diagram to decrease 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.

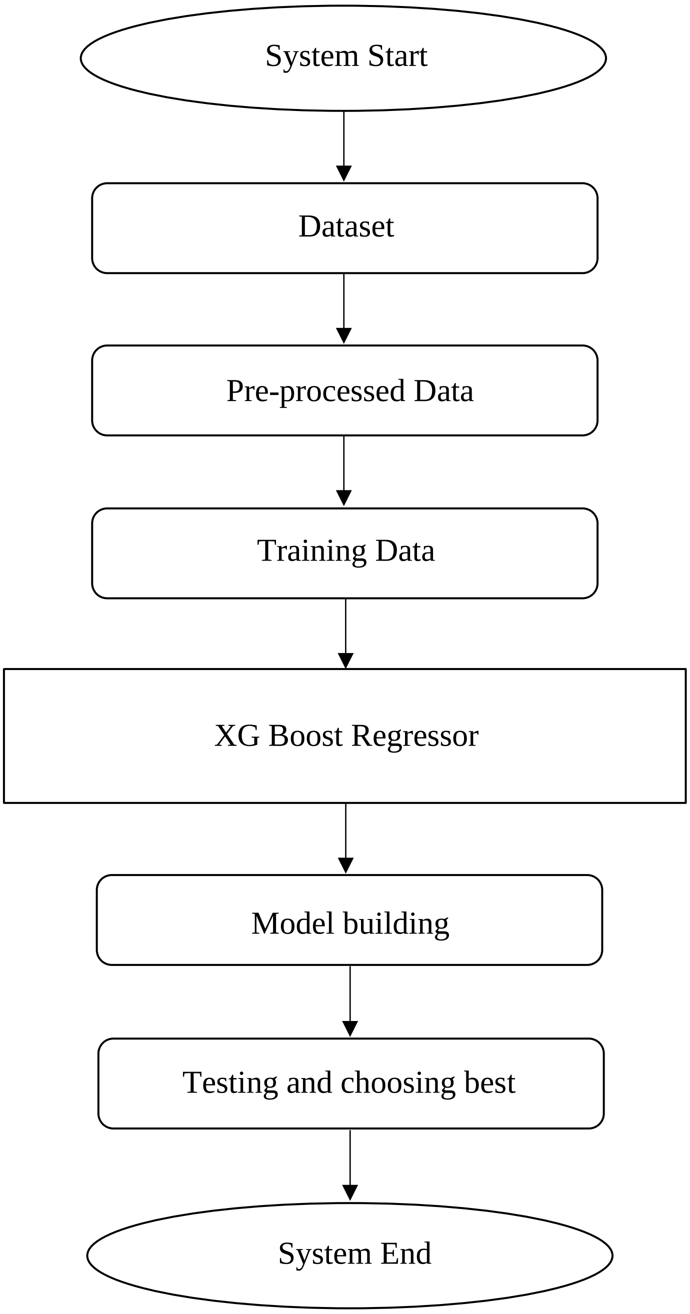
Before processing to top techniques that help with sales forecast, check out some of the free courses on sales management, sales conversion and many more on great learning academy.

Concept Generation

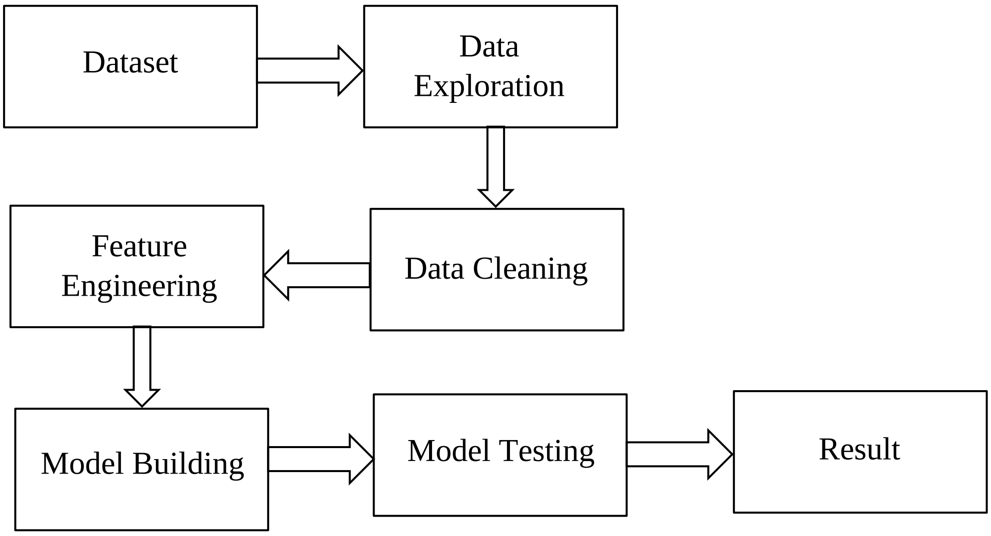
This product requires the tool of machine learning models to be written from scratch in order to suit our needs. Tweaking these models for our use is less daunting than coding it up from scratch. A well-trained model can either be repurposed or built. But building a model with the resources and data 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

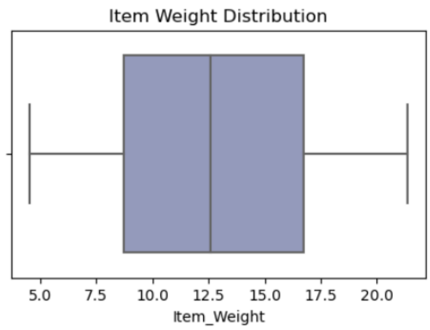
Some Basic Visualizations on Real World or Augmented Data:

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())
```

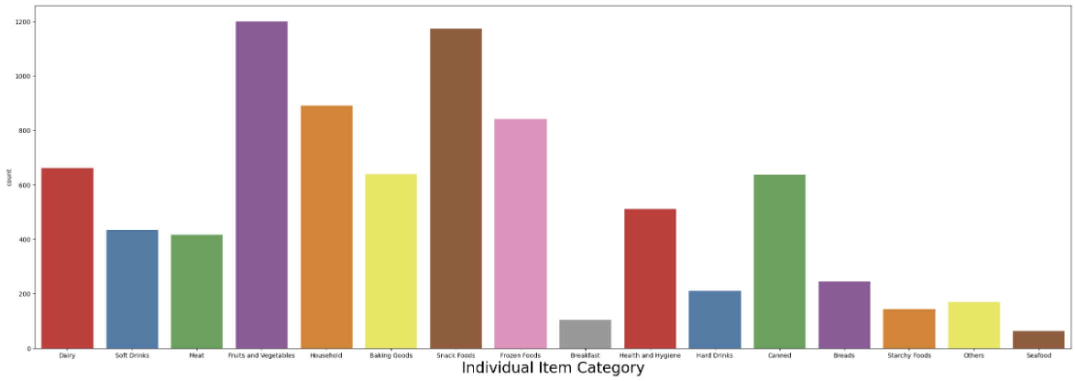
```
Missing Values :  4016
Missing values after filling: 0
```

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

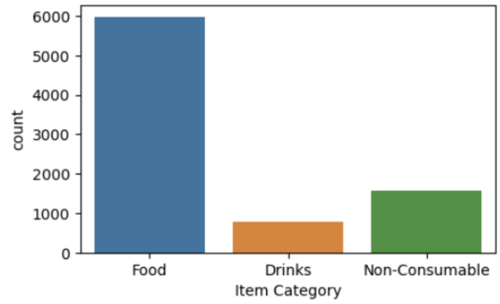
Out[11]:
```
Text(0.5, 1.0, 'Item Weight Distribution')
```



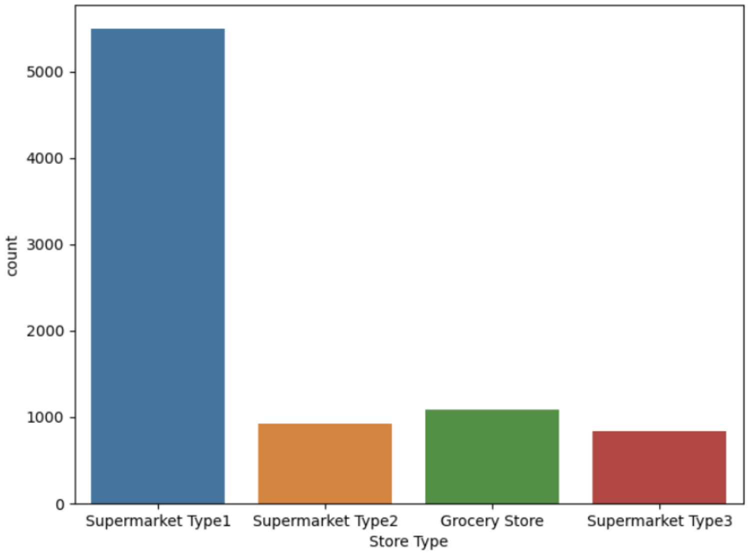
In [27]:
```
# Countplot for indiviual Item Category

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]:
```python
# 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]:
```python
# 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]:

Outlet_Type	Item_Outlet_Sales
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
```

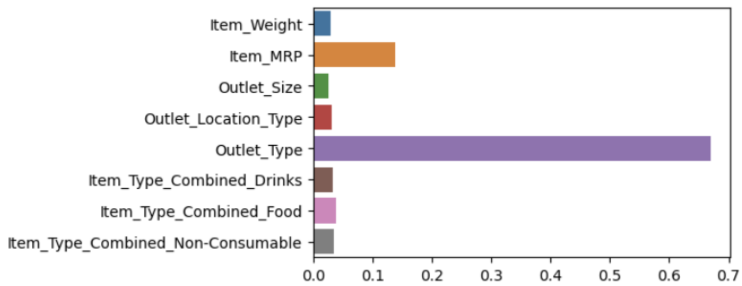
In [67]:
```
cross_val(XGBRegressor(),X, y, 5)
```

```
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/gayatripadmani/Big-Mart-Sales-Prediction

Conclusion

In this project, basics of machine learning and the associated data processing and modelling algorithms have been described, followed by their application for the task of sales prediction in Big Mart shopping canters at different locations. On implementation, the prediction results show the correlation among different attributes considered and how a particular location of medium size recorded the highest sales, suggesting that other shopping locations should follow similar patterns for improved sales.

Multiple instances parameters and various factors can be used to make this sales prediction more innovative and successful. Accuracy, which plays a key role in prediction-based systems, can be significantly increased as the number of parameters used are increased. Also, a look into how the sub-models work can lead to increase in productivity of system.

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)