DATA DRIVEN PREDICTIVE MODELING FOR PRODUCT LAUNCH SUCCESS

A PROJECT REPORT

Submitted by

AAKAASH KB	**711620243001**
JENIFER ROHINI R	**711620243004**
SHRI DHARSHINI J	**711620243008**

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

KATHIR COLLEGE OF ENGINEERIING

“WISDOM TREE”, NEELAMBUR, COIMBATORE – 641 062

NOV 2023

ANNA UNIVERSITY: CHENNAI 600 025

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report
DATA DRIVEN PREDICTIVE MODELING FOR PRODUCT LAUNCH SUCCESS
is the bonafide work of

AAKAASH KB (711620243001)
JENIFER ROHINI R (711620243004)
SHRI DHARSHINI J (711620243008)
who carried out this project under my supervision.

SIGNATURE
Dr. M.RAJESH BABU, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor and Head
Department of
Artificial Intelligence and Data Science
Kathir College of Engineering
Coimbatore – 641 062

SIGNATURE
Dr. M.RAJESH BABU,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
Professor and Head
Department of
Artificial Intelligence and Data Science
Kathir College of Engineering
Coimbatore – 641 062

Project viva voce held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our immense gratitude to **Thiru E.S. KATHIR, Chairman, Kathir Institutions**, Coimbatore for giving us an opportunity to study in their prestigious institution and to take up the project in Partial Fulfillment of the Regulation for the B.E Program.

We would like to express our deepest gratitude to **Thirumathi LAVANYA KATHIR, Secretary, Kathir Institutions**, Coimbatore for the soul support in our studies.

We would bound to express our gratitude to **Dr. G. DORAISAMY, CEO and Dr. R.UDAIYAKUMAR, M.E.,Ph.D., Principal, Kathir College of Engineering**, Coimbatore for their permission and constant encouragement throughout our course.

It is a great pleasure to express our sincere and wholehearted gratitude to Professor **Dr. Rajesh Babu,M.E.,Ph.D., Head of the Artificial Intelligence and Data Science**, for their constant suggestion and encouragement in the project work and for being supportive throughout the tenture of our project.

We also thank all our Faculty Members and Non Teaching Staff Members of Department of Artificial Intelligence and Data Science and our Lovable Parents and Friends who contribute many suitable ways for achieving final results.

ABSTRACT

Data-driven predictive modeling is a process that involves using statistical algorithms and machine learning techniques to analyze historical data and make predictions about future events or outcomes. The goal is to uncover patterns, relationships, and trends within the data that can be used to forecast future occurrences.

The objective of this research is to use data-driven predictive modelling techniques to improve the accuracy of product launch success prediction. The emergence of big data and data analytics presents a unique chance to examine several elements that impact a product's success or failure in the market. To create prediction models, the project collects historical data from prior product launches, market trends, customer behaviour, and other necessary characteristics.

Here as the main concern goes for data analytics , this follows on with data collection , preprocessing , exploratory data analysis, feature engineering and visualizations. After these processes, the model is build using XGBRegressor and the accuracy of the model is visualized along with the training data.

The project's results have great significance for companies looking to improve their overall financial performance, competitiveness in the market, and strategic planning.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1. Project Overview

Data-driven predictive modelling is a statistical method that makes use of previous and current data to foresee and predict expected future events using machine learning and data mining. It operates by evaluating both recent and past data, then extrapolating its findings onto a model designed to predict future events. Predictive modelling may be approached in two ways: model-driven and data-driven. Algorithmic approaches, often known as data-driven methods, identify an algorithm that generates the output given the input.

Here the model for predicting the product launch success is done using XGBRegressor model which provides almost the required accuracy. Good key performance indicators (KPIs) round out a technology product launch plan by providing concrete ways to measure the launch's success. KPIs provide you with insight into several parts of your launch, such as how you market it and how many sales you make, allowing you to determine what was successful and what needs improvement.

1.2. Objective

The main objective of this project is to develop a model for predicting the product launch success. The key part of this project is to have a better data analysis and feature engineering of the data. The primary objective of this project is to develop a data-driven predictive model to enhance the success rate of product launches. By leveraging historical and real-time data, the model aims to provide actionable insights and predictions that can guide decision-making throughout the product launch lifecycle.

The model is followed by the EDA process. The primary objective of Exploratory Data Analysis (EDA) is to gain a deeper understanding of the structure, patterns, and characteristics of the dataset before applying more complex statistical models. EDA helps identify important features, educates feature engineering, and directs the selection of suitable modelling methodologies through variable exploration. Moreover, it is essential for evaluating the quality of data, identifying mistakes, and facilitating efficient data preparation.

1.3.Importance of the project

Predicting product launch success is crucial for companies as it helps them to identify the potential of their product in the market and make informed decisions. It enables companies to understand the market demand, customer preferences, and competition, which can help them to optimize their product launch strategy and increase their chances of success.

The importance of a data-driven predictive modeling project for product launch success lies in its potential to significantly enhance the efficiency, effectiveness, and overall success of product launches. The key objectives to be dealt with can be given as follows:

- **Risk Reduction:**

It is possible to identify potential risks and difficulties related to a product launch using predictive modelling. Through the examination of past data and market patterns, the model is able to anticipate possible obstacles, allowing proactive approaches to risk reduction.

- **Resource Efficiency**

Effective resource management is essential to a successful product launch. By determining the most important elements

influencing performance and providing guidance for decisions on marketing expenditures, distribution routes, and other resources, predictive modelling aids in the optimisation of resource allocation.

- **Making Decisions Quickly:**

The predictive model's real-time insights give decision-makers access to pertinent data. This makes it possible to react quickly to shifting consumer tastes, rivalry, and market conditions, which enhances the strategy for launching a new product.

- **Customer Satisfaction:**

Understanding customer preferences and predicting their responses allows for the development of products that better meet customer needs. Satisfied customers are more likely to become repeat customers, contributing to long-term success.

- **Measurable success metrics:**

The implementation of predictive modelling to provide quantifiable success measures facilitates an objective evaluation of the efficacy of the product launch plan. This makes learning and constant development for next releases possible.

CHAPTER 2

LITERATURE SURVEY

a) Smirnov, P & Sudakov, Vladimir. (2021). Forecasting new product demand using machine learning. Journal of Physics: Conference Series. 1925. 012033. 10.1088/1742-6596/1925/1/012033.

Machine learning models are computational algorithms or mathematical frameworks designed to enable computers to learn patterns, make predictions, or perform specific tasks without being explicitly programmed for those tasks. These models helps in predicting the success of a product launch. This paper proposes the use of machine learning methods. They have used data about new product demand from the Ozon online store. The input data of the algorithm are characteristics such as the price, name, category and text description of the product. To solve the regression problem, various implementations of the gradient boosting algorithm were used, such as XGBoost, LightGBM, CatBoost. The proposed model satisfies initial conditions because it meets all the requirements – forecasting demand without any marketing research and to work independent of the type of goods and historical data, so the goal of the paper has been achieved. This model can be used by companies' analysts for optimizing sales assortment, planning and logistic optimization. But as the dataset used here is not used in any other model, we cannot have any comparative results. But the model is trained and works efficiently on forecasting.

b) Albora, Giambattista, Luciano Pietronero, Andrea Tacchella, and Andrea Zaccaria. (2017) "Product Progression: a machine learning approach to forecasting industrial upgrading." arXiv preprint arXiv:2105.15018.

This paper argues on out-of-sample forecast exercises should play the role on product progression, and they compare various machine learning models to set the prediction benchmark. The key object to forecast is the activation of new products, and that tree-based

algorithms clearly overperform both the quite strong auto-correlation benchmark and the other supervised algorithms. This paper launches idea to use cross validate and influences the use of random forests and many more type of decision trees. We can infer from this paper that desirable output of a classification task is not only a correct prediction, but also an assessment of the likelihood of the label, i.e, the activation. The likelihood provides a sort of confidence on the prediction. Here the fraction of positive elements as a function of the output (i.e., the scores) of the XGBoost and Random Forest algorithms in the activations prediction task are found helpful in analysing the product success rate.

c) Chavan, Sandeep & Panchal, Simsri & Sawant, Tanvi & Shinde, Janhavi. (2020). Predicting Online Product Sales using Machine Learning. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS040708.

Forecasting the sales are crucial in determining inventory stock levels and accurately estimating the future demand for goods has been an ongoing challenge in industries If goods are not readily available or if goods availability is more than demand overall profit can be compromised. This project is on creating a prediction model using machine learning algorithms for accurately predicting online product sales. This aims to use up-to date data which includes online reviews,online ratings ,online promotional strategies and sentiments and various other parameters for predicting product sales. This uses NLP for classification and Multiple Linear Regression model for training the data. Here more than of using the product price and details, it encompasses views on using the user reviews through online and then tries to have a prediction system. Here it shows the important on dealing with the audience reviews and the use of regression model is inferred from this paper. The regression model is proven good for predicting a continuous variable or outcome based on one or more input features.

d) Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. Data, 4(1), 15.

The main goal of this paper is to consider main approaches and case studies of using machine learning for sales forecasting. The effect of machine-learning generalization has been considered. This paper infers that ales prediction is rather a regression problem than a time series problem. Practice shows that the use of regression approaches can often give better results compared to time series methods. Machine-learning algorithms make it possible to find patterns in the time series. Some of the most popular methods used are tree-based machine-learning algorithms e.g., Random Forest, Gradient Boosting Machine etc., This paper also helps us in having a data pipeline by extracting and analysing the features of the dataset. The effect of machine-learning generalization consists in the fact that a regression algorithm captures the patterns which exist in the whole set of stores or products. This paper concludes that the use of regression approaches for sales forecasting can often give better results compared to time series methods. One of the main assumptions of regression methods is that the patterns in the historical data will be repeated in future. The accuracy on the validation set is an important indicator for choosing an optimal number of iterations of machine-learning algorithms. Thus this paper influences on having a regression model along with the data exploration processes. This tends to produce high accuracy in the prediction of success in product launch.

CHAPTER 3

METHODOLOGY

3.1. Data Collection

A thorough data gathering approach is required in the goal of establishing a data-driven prediction model for product launch success. The first stage is to clearly define project objectives and identify important factors and performance indicators crucial to the success of a product launch. Following that, a wide range of data sources, including internal databases, external reports, market research, consumer feedback channels, and social media platforms, should be reviewed. The data collection strategy must be meticulously prepared, describing the frequency and methods for collecting data and being in sync with the project's timeframe and goals.

The dataset provided here consists of a collection of data about the product sales inlet and outlet of items such as Item_Identifier,Item_Weight,Item_Fat_Content,Item_Visibility,Item_Type,Item_MRP,Outlet_Identifier,Outlet_Establishment_Year,Outlet_Size,Outlet_Location_Type,Outlet_Type and Item_Outlet_Sales. This gives us an overall idea about the product rate and its sales throughout the period.

3.2. Data preprocessing

Data preprocessing is an important stage in data analytics that involves cleaning and converting raw data into an analysis-ready state. This stage is critical for assuring data quality, dealing with missing or inconsistent information, and improving the overall accuracy and efficacy of analytical models.

Data analytics is the process of inspecting, cleaning, transforming, and modeling data with the goal of discovering useful information, drawing conclusions, and supporting decision-making. It involves various techniques and tools to analyze and interpret complex datasets, providing insights that can inform business strategies, scientific research, and other decision-making processes. The various data preprocessing components can be given as follows:

- **Handling missing data:**

  Handling missing data is a critical stage in the data preparation phase, intended to address cases when certain observations or values are missing from the dataset. Missing data can arise for a variety of reasons, including measurement errors, incorrect data input, or purposeful non-response. Failure to handle missing data correctly might result in unequal analysis and inaccurate model predictions.

- **Handling outliers:**

  Outliers are data points that differ dramatically from the bulk of observations in a dataset. Outliers are numbers that are exceptionally high or low in comparison to the rest of the data and may skew the findings of statistical analysis or machine learning models. So, handling these outliers helps in better understanding of the data trends.

- **Encoding Categorical Variables:**

  Categorical variables are values that are basically of characters and not numbers. For computation, these variables are processed into **One Hot Encoding.** [(4)]This creates binary columns for each category in a categorical variable and assign a unique numerical label to each category.

  There are even many more data pre-processing techniques. This would greatly help in optimized model building and visualization of the data.

3.3.Model Building

Model building is a key phase in the process of developing predictive models or analytical solutions to solve a particular problem. After preprocessing and properly dealing with the numerical and categorical values, the model is first passed with training data and is trained to yield the optimal solution. The model being used here is **XGBRegressor.**[1,3,5] It utilizes decision trees as base learners and combines them to create a strong predictive model. The training process involves minimizing a loss function, which measures the difference between the predicted values and the actual target values.

The **cross_validate** function from the sklearn.model_selection module is also used for cross-validating the performance evaluation of the machine learning model. [3] Cross-validation is a technique for evaluating a model's generalization performance by dividing the dataset into different subsets for training and testing.

CHAPTER 4

DATA PREPROCESSING

4.1. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an essential step in the data analysis process that involves examining and visualizing the characteristics of a dataset to gain insights, detect patterns, and understand the underlying structure. EDA aims to provide a preliminary understanding of the data, identify relationships between variables, and guide further analysis. Here on prior of building the model, various steps are carried out in order for accurate analysis:

- **Data Summary:**

A data summary is a brief presentation of significant information and attributes from a dataset. A data summary's objective is to give an overview that helps analysts, stakeholders, or decision-makers to rapidly comprehend the most important aspects of the data.

Many functions are used here in order to understand the features of the dataset. The **info( )** function is used to provide a quick overview of the data frame, showing the data types and the count of non-null values for each column. The **describe( )** function is used to have a descriptive statistics of a data frame, including measures of central tendency, dispersion, and the shape of the distribution.(Appendix).It provides statistics such as mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values. Other functions such as count_values and sum is also used in order to know about statistics of the data.

- **Data Visualization:**

Data visualization is the graphical representation of data to extract insights, patterns, and trends that may not be immediately apparent in raw datasets. Visualizations play a crucial role in exploratory data analysis, communication of findings, and making data-driven decisions. Various types of visualizations can be created based on the

nature of the data and the goals of the analysis. Some of the visualizations used here are **Histogram** and **Box plots.** Histograms display the distribution of numerical data by dividing it into bins and plotting the frequency of each bin. Histograms help identify patterns and outliers. Box plots also known as box and whisker plots Illustrate the distribution of numerical data by displaying key summary statistics such as the median, quartiles, and potential outliers. They both are together presented using the **subplots.** Subplots refer to a way of organizing multiple plots within a single figure. Subplots allows us to display several visualizations side by side or in a grid, making it easier to compare different aspects of the data.

4.2. Data Wrangling and Engineering

Data wrangling, also known as data munging, refers to the process of cleaning, structuring, and enriching raw data into a more usable and understandable form. Data engineering involves the design and construction of systems and architectures for the collection, storage, and processing of large volumes of data. While data wrangling and data engineering have distinct focuses, there is overlap, especially in tasks related to data transformation and cleaning. Both involve preparing data for analysis and ensuring its quality.

Data wrangling is primarily concerned with preparing individual datasets for analysis, while data engineering involves creating the infrastructure and processes to handle large-scale data operations, ensuring data is efficiently collected, stored, and processed. Some of the key components dealt in this process are:

- **Data mapping:**

Data mapping involves establishing a relationship between data elements in different datasets or systems. It helps ensure that data from one source can be correctly translated or transformed to match the format or structure of another source. Here the mapping is carried out

between Item ID to Item Type in order to create a relationship between them and deal with the data structure.(Appendix)

- **Handling missing data:**

For filling out the missing values, we can use the mapping of already existing values or take median value for new entries. We can also use mode for filling out the values.

The data is also standardized that involves transforming the data into a standard scale. The duplicates values are also dropped and sorted in order to provide a clear idea of the data values.

4.3. Dataset Generalization

The dataset is completely generalized by grouping out the data created, the functions that were created to handle the weights and outlets of the product. The standardized data about the fat and the corrected data is also grouped which now returns a single complete data frame.

Now the functions that are used to understand the descriptive summary of the data is used and the data is free of outliers and missing values in its present state.

4.4. Handling Categorical Data

Handling categorical data is a crucial step in data preprocessing, especially when working with machine learning models that require numerical input. Categorical data represents variables that can take on a limited, fixed number of distinct categories or labels. We have used **One – Hot Encoding** here to deal with the characters. One-hot encoding creates binary columns for each category and indicates the presence of a category with a 1 or 0. It is suitable for nominal categorical data with no inherent order.

CHAPTER 5
IMPLEMENTATION

5.1. Module 1: Market Research

Data market research is the systematic collecting, analysis, and interpretation of data industry information. Insights regarding market trends, upcoming technologies, industry leaders, and the entire landscape of data-related goods and services might be included. This module includes a thorough examination of market dynamics, industry trends, and competitive landscapes linked to product launches. Analysing market dynamics, customer preferences, regulatory settings, and competition strategies are all part of dealing with this modeling.

The data is also collected after market research which would provide beneficial for training of the model. The data collection is the process of collecting the relevant and comprehensive data to support the development of predictive models. Here a collection of product data with its value and outlets is chosen as a dataset. The further processing goes on with the data cleaning and transformation to the training level.

5.2. Module 2: Data preprocessing and EDA

Now the collected dataset is preprocessed, cleaned, integrated and transformed into a prediction dataset. First, the data summary is described and the list of missing values and other outliers are known. This is done with using the Exploratory Data analysis (EDA) which is the crucial part in dealing with the data science pipeline. Now the mapping and handling of missing values is carried out. The mapping is done in order to create the relationship between the relevant features. This would help in further processing. (Appendix)Then the missing values are filled with median of the data column or with the mode. This helps in having the data within the leverage and avoiding outliers.

Separate standardizing functions are used in order to convert the data into an understandable form. The conversion of all data into numerical form would help us in easier training and visualization. After

exploring and visualizing the data, the categorical variables are handled with One-Hot Encoding. This created binary columns to mark the existence with 1 and ensure no value as 0. Now the cleaned and standardized dataset is described and the problems have been rectified. This now provides a generalized view of the data.

5.3. Module 3: Feature and Model Selection

The required features is selected and they are represented and visualized separately in order to understand the required dataset. A visualization bar chart is created to view the efficiency and contribution of each field and the higher efficient fields are brought to concern. The model chosen here to train is XGBRegressor[5]. This would help in preparing the data for prediction. The training data would be first given to the model and the testing data is used for prediction.

5.4. Module 4: Model Training and Validation

The model is created initially and their parameters specifies the learning task and the corresponding objective function. In this model, "reg:squarederror" indicates that the model is designed for regression tasks, and the objective is to minimize the mean squared error.[1,4] R-squared is a statistical measure that represents the proportion of the variance in the dependent variable that is explained by the independent variables in a regression model. It ranges from 0 to 1, where 1 indicates a perfect fit.

Root Mean Squared Error (RMSE) is a commonly used metric to evaluate the performance of a regression model, particularly in the context of predictive modeling(Appendix).[7] It is an extension of Mean Squared Error (MSE) and is calculated as the square root of the average squared differences between predicted and actual values.

Now the model is fit with the training data and the model is made to understand the features of the training dataset. Here the sklearn.metrics is a module within the scikit-learn library that provides a variety of functions for evaluating the performance of machine learning models. These functions cover a wide range of metrics for classification, regression, clustering, and multilabel tasks. Now the test data is passed on to the model and the predicted data with the training data is presented using a plot where the prediction overshades the testing data with higher accuracy.

CHAPTER 6
RESULT AND ANALYSIS

Thus, the testing of the model is done and found to produce around the level of aimed accuracy. The goal of the project was to develop a predictive model that would help in predicting the success rate of the product before its launch. The data collected would be of the product prices and their sales details in their respective fields. Now when the new product is launched, the category of the product is chosen and then similar product details, their investments and their success rate over a period of time is also given a look. With this help of prediction model, the new product to be launched is tested and the model predicts and visualizes the success rate of the product. Using XGBRegressor, we have achieved almost 70% accuracy of predicting the success. But the usage of Exploratory Data Analysis and data preprocessing were made accurate in order to maintain the data pipeline and produce optimal results in achieving the success rate of the model. Thus, the data analytics is highly efficient and provides us in yielding an accurate model. The success rate can be given even higher with other highly advanced models but this almost have high prediction rate.

CHAPTER 7
CONCLUSION AND FUTURE WORK

This project on predictive modeling for product launch success has been a comprehensive exploration into leveraging data analytics and predictive modeling to enhance our understanding of the factors influencing product launch outcomes. The project successfully addressed its objectives and produced valuable insights that can inform strategic decision-making in the product launch process. We carefully gathered and preprocessed the data to create a solid dataset that contained key characteristics that helped the product launch succeed. The application of exploratory data analysis (EDA) provided valuable insights into patterns, correlations, and trends that clarified the dynamics of different elements within the ecosystem of product launches.

While the project has yielded valuable insights, there are several avenues for future research and improvement. The future works would be of analyzing and having an enhanced dataset with many more features. The refinements and other hyperparameter tuning should be made more in a refined way. The user feedback data should also be collected in order to make the decision even more prominent to the user. Collaboration with stakeholders and the expansion of the project's scope to encompass diverse product categories could further enhance its impact. Thus, this project could be further developed in increasing its data pipeline contributing to the accuracy of the model.

BIBLIOGRAPHY

1. Salmen, Alexander. (2021). New Product Launch Success: A Literature Review. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis. 69. 151-176. 10.11118/actaun.2021.008.
2. Smirnov, P & Sudakov, Vladimir. (2021). Forecasting new product demand using machine learning. Journal of Physics: Conference Series. 1925. 012033. 10.1088/1742-6596/1925/1/012033.
3. Albora, Giambattista, Luciano Pietronero, Andrea Tacchella, and Andrea Zaccaria. (2017) "Product Progression: a machine learning approach to forecasting industrial upgrading." arXiv preprint arXiv:2105.15018.
4. Chavan, Sandeep & Panchal, Simsri & Sawant, Tanvi & Shinde, Janhavi. (2020). Predicting Online Product Sales using Machine Learning. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS040708.
5. Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. Data, 4(1), 15.
6. Kadam, H., Shevade, R., Ketkar, D., & Rajguru, S. (2018). A forecast for big mart sales based on random forests and multiple linear regression. Int. J. Eng. Dev. Res, 6(4), 41-42.
7. Saha, P. (2019). Performance analysis of the Machine Learning Classifiers to predict the behaviour of the customers, when a new product is launched in the market. vol, 5, 1907-1911.

APPENDIX

Source Code

main.py

#Loading and Importing Dataset

```
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Project/sales_prediction.csv')
df.head()

X = df.drop(columns=["Item_Outlet_Sales"])
Y = df["Item_Outlet_Sales"]
SEED = 42

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3,
  random_state = SEED)

X_train.shape, X_test.shape
X_train.head(3)
Y_train.head(3)
```

Exploratory Data Analysis

```
X_train_c = X_train.copy()
X_train_c.info()
X_train_c.isnull().sum()
num_data = X_train_c.select_dtypes(exclude=["object"])
num_data.head()
num_data.describe()
```

```
num_data.isnull().sum()

import seaborn as sns
import matplotlib.pyplot as plt

def visuzalize_num_features(data_frame, col_name):
  fig, ax = plt.subplots(1, 2, figsize=(12,5))
  sns.histplot(data = X_train_c, x = col_name, ax = ax[0])
  sns.boxplot(data = X_train_c, y = col_name, ax = ax[1])

visuzalize_num_features(X_train_c, "Item_Weight")
visuzalize_num_features(X_train_c, "Item_Visibility")
visuzalize_num_features(X_train_c, "Item_MRP")
visuzalize_num_features(X_train_c, "Outlet_Establishment_Year")
sns.countplot(data = X_train_c, x = "Outlet_Establishment_Year")

cat_features = X_train_c.select_dtypes(include=["object"])
cat_features.head()
cat_features.describe()
cat_features.isnull().sum()
cat_features['Item_Identifier'].value_counts()
cat_features['Item_Fat_Content'].value_counts()
cat_features['Item_Type'].value_counts()
cat_features['Outlet_Identifier'].value_counts()
cat_features['Outlet_Size'].value_counts()
cat_features['Outlet_Location_Type'].value_counts()
cat_features['Outlet_Type'].value_counts()
```

Data Wrangling and Feature Engineering

 # Creating High Level Item Types To Seperate Out Based on First Two Characters

```
X_train_c['Item_Identifier'].str[:2].value_counts()
```

 # Mapping Item ID to Item Type

```
def create_item_type(data_frame):
  data_frame["Item_Type"] = data_frame["Item_Identifier"].str[:2]
  data_frame["Item_Type"] = data_frame["Item_Type"].map({
      'FD' : 'Food',
      'NC' : 'Non_Consumables',
      'DR' : 'Drink'
  })

  return data_frame


X_train_c = create_item_type(X_train_c)
X_train_c.head()
```

 # Handling Missing Data For Item_Weight

```
X_train_c.isnull().sum()

X_train_c[["Item_Identifier",
  "Item_Weight"]].drop_duplicates().sort_values(by=["Item_Identifier"])
X_train_c[["Item_Type",
  "Item_Weight"]].drop_duplicates().sort_values(by=["Item_Type"])
```

 # Filling Missing Values

 # Use Mapping of Already Existing Values or Take Median Value For New Entries

```
ITEM_ID_WEIGHT_PIVOT =
  X_train_c.pivot_table(values="Item_Weight",
  index="Item_Identifier").reset_index()
ITEM_ID_WEIGHT_MAPPING =
  dict(zip(ITEM_ID_WEIGHT_PIVOT["Item_Identifier"],
  ITEM_ID_WEIGHT_PIVOT["Item_Weight"]))
list(ITEM_ID_WEIGHT_MAPPING.items())[:10]
```

```
ITEM_TYPE_WEIGHT_PIVOT =
  X_train_c.pivot_table(values="Item_Weight", index="Item_Type",
  aggfunc="median").reset_index()
ITEM_TYPE_WEIGHT_MAPPING =
  dict(zip(ITEM_TYPE_WEIGHT_PIVOT["Item_Type"],
  ITEM_TYPE_WEIGHT_PIVOT["Item_Weight"]))
list(ITEM_TYPE_WEIGHT_MAPPING.items())[:10]

def handle_weight(data_frame):
  data_frame.loc[:,"Item_Weight"]                                    =
  data_frame.loc[:,"Item_Weight"].fillna(data_frame.loc[:,"Item_Identifi
  er"].map(ITEM_ID_WEIGHT_MAPPING))
  data_frame.loc[:,"Item_Weight"]                                    =
  data_frame.loc[:,"Item_Weight"].fillna(data_frame.loc[:,"Item_Type"].
  map(ITEM_TYPE_WEIGHT_MAPPING))

  return data_frame

X_train_c = handle_weight(X_train_c)
X_train_c.isnull().sum()
```

 # Handling Missing Data For Outlet_Size Using Mode

```
X_train_c.groupby(by=["Outlet_Type", "Outlet_Size"]).size()

import pandas as pd
def mode_func(x):
    if x.isna().all():
        return None  # Return None for all missing values
    mode_result = x.mode()
    if not mode_result.empty:
        return mode_result.iloc[0]
    else:
        return x.value_counts().idxmax()
```

```
OUTLET_TYPE_SIZE_PIVOT =
 X_train_c.groupby("Outlet_Type")["Outlet_Size"].agg(mode_func).res
 et_index()
OUTLET_TYPE_SIZE_MAPPING =
 dict(zip(OUTLET_TYPE_SIZE_PIVOT["Outlet_Type"],
 OUTLET_TYPE_SIZE_PIVOT["Outlet_Size"]))
OUTLET_TYPE_SIZE_MAPPING

def handle_outlet(data_frame):
 data_frame.loc[:,"Outlet_Size"] =
 data_frame.loc[:,"Outlet_Size"].fillna(data_frame.loc[:,"Outlet_Type"]
 .map(OUTLET_TYPE_SIZE_MAPPING))
 return data_frame

X_train_c = handle_outlet(X_train_c)
X_train_c.isnull().sum()
```

 # Standardizing Item_Fat_Content Categories

```
X_train_c["Item_Fat_Content"].value_counts()
def standardize_itemfat(data_frame):
 data_frame["Item_Fat_Content"] =
 data_frame["Item_Fat_Content"].replace({
     "Low Fat" : "Low_Fat",
     "LF" : "Low_Fat",
     "reg" : "Regular",
     "low fat" : "Low_Fat"
 })
 return data_frame

X_train_c = standardize_itemfat(X_train_c)
X_train_c['Item_Fat_Content'].value_counts()
```

 # Correct Item Fat Content For Non Consumables

```
X_train_c.groupby(by=["Item_Type", "Item_Fat_Content"]).size()
```

```
X_train_c.loc[X_train_c["Item_Type"] == 'Non_Consumables',
 'Item_Fat_Content']

def correct_itemfat(data_frame):
  data_frame.loc[data_frame["Item_Type"] == "Non_Consumables",
  "Item_Fat_Content"] = "Non_Edible"
  return data_frame

X_train_c = correct_itemfat(X_train_c)
X_train_c.groupby(by=["Item_Type", "Item_Fat_Content"]).size()
X_train_c.info()
```

Dataset Preparing - Generalization

```
def dataset(data_frame):
  data_frame = create_item_type(data_frame)
  data_frame = handle_weight(data_frame)
  data_frame = handle_outlet(data_frame)
  data_frame = standardize_itemfat(data_frame)
  data_frame = correct_itemfat(data_frame)

  return data_frame

X_train.isnull().sum()
X_train = dataset(X_train)
X_train.isnull().sum()
X_test.isnull().sum()
X_test = dataset(X_test)
X_test.isnull().sum()
```

Handling Categorical Data

```
cat_feats = X_train.select_dtypes(include=["object"])
cat_feats
```

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(handle_unknown="ignore")
ohe.fit(cat_feats)

ohe_feature_names =
  ohe.get_feature_names_out(input_features=cat_feats.columns)
ohe_feature_names

num_feats_train =
  X_train.select_dtypes(exclude=["object"]).reset_index(drop=True)
num_feats_train.head()

cat_feats_train = X_train.select_dtypes(include=["object"])
X_train_cat_ohe =
  pd.DataFrame(ohe.transform(cat_feats_train).toarray(), columns =
  ohe_feature_names)
X_train_cat_ohe.head()

X_train_final = pd.concat([num_feats_train, X_train_cat_ohe], axis = 1)
X_train_final.head()

final_columns = X_train_final.columns.values
final_columns


num_feats_test =
  X_test.select_dtypes(exclude=["object"]).reset_index(drop=True)
cat_feats_test = X_test.select_dtypes(include=["object"])
X_test_cat_ohe = pd.DataFrame(ohe.transform(cat_feats_test).toarray(),
  columns = ohe_feature_names)
X_test_final = pd.concat([num_feats_test, X_test_cat_ohe], axis = 1)
X_test_final.head()
```

Model Creation

```
import xgboost as xgb
from sklearn.model_selection import cross_validate
import numpy as np

cv = 5
model=xgb.XGBRegressor(objective="reg:squarederror",
          random_state=SEED)

cv_results = cross_validate(model, X_train_final, Y_train, cv=cv,
 scoring=("r2","neg_root_mean_squared_error"),)
print("Model :",model)
r2_scores = cv_results["test_r2"]
print("R2 CV Scores :",r2_scores)
print("R2 CV Scores Mean / Stddev :",np.mean(r2_scores), "/",
 np.std(r2_scores))

rmse_scores = cv_results["test_neg_root_mean_squared_error"]
rmse_scores = [-1 * score for score in rmse_scores]
print("RMSE CV Scores :",rmse_scores)
print("RMSE CV Scores Mean / Stddev :",np.mean(rmse_scores), "/",
 np.std(rmse_scores))


X_train_final.shape
X_test_final.shape
model.fit(X_train_final,Y_train)
Y_pred = model.predict(X_test_final)
Y_pred

from sklearn.metrics import r2_score, mean_squared_error
print("R2 Score :",r2_score(Y_test,Y_pred))
print("RMSE Score :",mean_squared_error(Y_test, Y_pred,
 squared=False))
```

```
Y_pred = pd.DataFrame(Y_pred)
Y_pred
Y_test = Y_test.reset_index(drop=True)
#Y_test.drop(["index"],axis=1)
Y_test
result = pd.concat([Y_test, Y_pred], axis=1)
result.rename(columns = {'Item_Outlet_Sales':'Test Value', 0:"Predicted
  Value"}, inplace = True)
result.head()
plt.plot(result['Test Value'], label='Test Value')
plt.plot(result['Predicted Value'], label='Predicted Value')
plt.legend()
plt.show()
```

SAMPLE OUTPUT:

```
R2 Score : 0.5701327508635561
RMSE Score : 1097.2945741485482
```