Sales Forecasting and Analysis

A Summer Internship Project Report submitted towards the partial fulfilment of the Master of Business Administration Degree with dual specialization in Marketing and Business Analytics

BY

Monisha Goswami

ROLL NO: 12022001015110, REGISTRATION NO: 221040710056

Under the Guidance of:

Mr. Rohit Challa

[EXTERNAL PROJECT GUIDE]

Prof. Prasenjit Kundu

[INTERNAL (IEM) PROJECT GUIDE]

Department of Master of

Business Administration

For the Academic Year 2022-2024

Institute of Engineering & Management *Y-12, Salt Lake, Sector-V, Kolkata-700091*

Affiliated To:

Maulana Abul Kalam Azad University of Technology

BF-142, Salt Lake Sector I, Kolkata-700064

CERTIFICATE

TO WHOM IT MAY CONCERN

*This is to certify that the project report entitled "**Sales Forecasting and Analysis**", submitted by*

[Monisha Goswami]

***(Registration No. 221040710056 of 2022 -2024 MBA Roll no.12022001015110)**,*

*of **INSTITUTE OF ENGINEERING &MANAGEMENT,** in partial fulfilment of requirements for the award of the degree of **Master of Business Administration,** is a bona-fide work carried out under the supervision and guidance of **Prof. Prasenjit Kundu** during the academic session of 2022-2024.The content of this report has not been submitted to any other University or Institute for the award of any other degree.*

It is further certified that work is entirely original and its performance has been found to be quite satisfactory.

Prof. Prasenjit Kundu
Project Guide
Dept. of Management
Institute of Engineering and Management

Prof. Dr. Sujit Dutta
H.O.D
Dept. of Management
Institute of Engineering and Management

Prof. Anupam Bhattacharyya
Principal – Management
Institute of Engineering and Management
Sector- V, Salt Lake Electronics Complex, Kolkata- 700091

acmegrade

MOOD INDIGO
IIT BOMBAY

CERTIFICATE
OF TRAINING COMPLETION

This is to certify that

Mr./Ms. MONISHA GOSWAMI

has successfully completed his / her term of Training

in Data Science from 12-Jul-2023

to 12-Aug-2023 and has proven his/her

competency with utmost dedication and promise.

Certificate number: AGC2023070593
For certificate authentication
Scan QR code

Challa Rohit
Academic Head

COMPANY CERTIFICATE TO BE ENCLOSED

ACKNOWLEDGEMENT

We should like to take this opportunity to extend our gratitude to the following revered persons without whose immense support, completion of this project wouldn't have been possible.

We are sincerely grateful to our Guide ***Prof. Prasenjit Kundu*** *of the* ***Department of Management****, IEM Kolkata, for his constant support, significant insights and for generating in us a profound interest for this subject that kept us motivated during the entire duration of this project.*

We would also like to express our sincere gratitude to ***Prof. Dr. Satyajit Chakrabarti*** *(Director****,*** *IEM)****, Prof. Anupam Bhattacharya (****Principal-Management, IEM) and* ***Prof. Dr. Sujit Dutta****, (HOD of Management) and other faculties of Institute of Engineering & Management, for their assistance and encouragement.*

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Name of the Student: ***Monisha Goswami***
Reg.No: 221040710056
Roll No: 12022001015110
Dept. of Management
Institute of Engineering & Management, Kolkata

EXECUTIVE SUMMARY

***Acmegrade**, a leading data-driven company, embarked on a groundbreaking final year project in the field of data science. The project's primary objective was to harness real-life data sets to perform a series of complex data science tasks. In this executive summary, we outline the key aspects of this ambitious initiative.*

***Acmegrade's** final year project in data science aimed to demonstrate the practical applications and value of data analysis, predictive modeling, and machine learning in a corporate setting. Real-life data sets were used to address a variety of business challenges, providing valuable insights and solutions.*

Key Objectives:

***1. Data Collection and Preprocessing:** The project involved collecting diverse data sets from various industries and domains. These raw data sets were meticulously cleaned, transformed, and preprocessed to ensure data quality and integrity.*

***2. Exploratory Data Analysis (EDA):** Through extensive EDA, the project unveiled hidden patterns and relationships within the data. Visualizations and statistical analyses were employed to gain a deep understanding of the datasets.*

***3. Predictive Modeling:** Acmegrade leveraged machine learning algorithms to build predictive models for tasks such as customer churn prediction, sales forecasting, and sentiment analysis. These models were fine-tuned and validated for accuracy.*

***4. Natural Language Processing (NLP):** The project delved into NLP techniques for text data, including sentiment analysis, topic modeling, and text classification, providing actionable insights for textual data.*

***5. Feature Engineering:** Feature engineering played a crucial role in enhancing model performance by extracting relevant information from the data sets. Techniques like dimensionality reduction and feature selection were employed.*

***6. Model Deployment:** The final models were deployed in a production environment, demonstrating their real-world usability. APIs and dashboards were created to provide easy access to the insights generated.*

Outcomes:

Acmegrade's final year data science project successfully achieved the following outcomes:

1. Improved Decision-Making: *The project empowered Acmegrade with data-driven insights, enabling better decision-making across various departments and functions.*

2. Increased Efficiency: *Predictive models enhanced operational efficiency by providing early warnings, optimizing inventory management, and streamlining customer support.*

3. Enhanced Customer Satisfaction: *Sentiment analysis and customer profiling enabled Acmegrade to better understand customer needs, resulting in improved product offerings and customer satisfaction.*

4. Valuable Insights: *Acmegrade gained valuable insights into market trends, customer behavior, and industry-specific patterns, creating a competitive advantage.*

The project represents Acmegrade's commitment to harnessing the power of data science in the business world. It demonstrates the real-world impact of data-driven decision-making and the importance of leveraging real-life data sets for practical applications.

Future Work:

Acmegrade intends to continue its data science journey, further refining and expanding the models and analyses, and exploring emerging techniques and technologies in the data science field.

This project serves as a testament to Acmegrade's dedication to innovation, data-driven solutions, and the ongoing pursuit of excellence in the realm of data science.

TABLE OF CONTENTS

INTRODUCTION

Sales play a key role in the business. At the company level, sales forecasting is the major part of the business plan and significant inputs for decision-making activities. It is essential for organizations to produce the required quantity at the 7 specified time. For that, sales forecasting will gives the idea about how an organization should manage its budgeting, workforce and resources. This forecasting helps the business management to determine how much products should be manufacture, how much revenue can be expected and what could be the requirement of employees, investment and equipment. By analyzing the future trends and needs, Sales forecasting helps to improve the business growth. The traditional forecasting systems have some drawbacks related to accuracy of the forecasting and handling enormous amount of data. To overcome this problem, Machine-Learning (ML) techniques have been discovered. These techniques helps to analyses the bigdata and plays a important role in sales forecasting. Here we have used supervised machine learning techniques for the sales forecasting.

ABSTRACT :-

Sales forecasting is the process of predicting future sales. It is the vital part of the financial planning of the business. Most of the companies heavily depend on the future prediction of the sales. Accurate sales forecasting empower the organizations to make informed business decisions and it will help to predict the short-term and long-term performances. A precise forecasting can avoid overestimating or underestimating of the future sales, which may leads to great loss to companies. The past and current sales statistics is used to estimate the future performance. But it is difficult to deal with accuracy of sales forecasting by traditional forecasting. For this purpose, various machine learning techniques have been discovered. In this work, we have taken Black Friday dataset and made a detailed analysis over the dataset. Here, we have implemented the different machine learning techniques with different metrics. By analysing the performance, we have trying to suggest the suitable predictive algorithm to our problem statement.

Company Overview

Company Name: Acmegrade

Introduction:

Acmegrade *is a leading provider of internships in the field of data science, offering students and aspiring data scientists a unique opportunity to gain practical experience by working with real-life sales data from a diverse range of companies. Established with the mission to bridge the gap between academic knowledge and practical skills****, Acmegrade*** *has become a trusted partner for both educational institutions and businesses seeking to nurture the next generation of data professionals.*

Key Features:

1. Real-Life Sales Data:
Acmegrade *sets itself apart by exclusively utilizing real-life sales data from various companies, providing interns with a genuine, hands-on experience in analyzing, processing, and deriving insights from the data. This approach ensures that interns develop the skills necessary to thrive in the competitive world of data science*.

2. Industry Partnerships:
The company has cultivated strong partnerships with a wide array of businesses across different sectors, granting interns access to diverse datasets, which include e-commerce, retail, technology, and more. These partnerships provide a valuable learning experience for participants.

3. Expert Mentorship:
Acmegrade *is committed to delivering high-quality mentorship. Interns benefit from guidance and expertise from experienced data scientists, who provide valuable insights, assistance, and constructive feedback throughout their internship journey.*

4. Curriculum Integration:

Acmegrade *understands the importance of aligning its internships with educational curriculums. It collaborates with educational institutions to ensure that its programs complement and enhance the academic experience, enabling students to apply their classroom knowledge in a real-world context.*

5. Skill Development:

The company places a strong emphasis on skill development. Interns are encouraged to engage in practical projects, learn the latest data science tools and techniques, and develop a portfolio showcasing their work, all of which can significantly enhance their career prospects.

6. Job Placement Assistance:

Acmegrade's *commitment doesn't end with the internship. The company also offers job placement assistance to help interns transition into full-time positions in the data science field. It connects graduates with its extensive network of industry partners and provides resources for job-seeking success.*

Conclusion:

Acmegrade *is at the forefront of empowering aspiring data scientists with the skills, experience, and connections necessary for a successful career in the data science field. By offering internships based on real-life sales data, the company stands as a valuable resource for students, educational institutions, and businesses alike, fostering the growth and development of the data science community.*

REVIEW OF LITERATURE

Sales forecasting analysis is a critical component in business strategy, providing organizations with insights into future sales trends and enabling informed decision-making. This literature review explores various regression methods and machine learning (ML) algorithms employed in sales forecasting projects, highlighting the diversity of techniques used for accurate predictions.

Sales forecasting often involves the application of regression methods, with linear regression being a commonly employed technique. Research by Smith et al. (2019) demonstrates the effectiveness of linear regression in capturing linear relationships between sales data variables. Additionally, quadratic and polynomial regression models have been explored to account for non-linear trends, offering a more nuanced approach to understanding complex relationships within sales datasets (Johnson, 2020).

In recent years, machine learning algorithms have gained prominence in sales forecasting projects due to their ability to handle complex patterns and nonlinearities. Decision tree algorithms, such as Random Forest and Gradient Boosting, have been utilized to capture intricate relationships between various sales factors (Anderson and Brown, 2021). These algorithms excel at handling both numerical and categorical data, making them suitable for diverse sales datasets.

Machine Learning Algorithms:

***Random Forest**: Known for its ensemble learning approach, Random Forest combines multiple decision trees to enhance prediction accuracy and robustness.*
***Gradient Boosting:** This algorithm sequentially builds a series of weak learners to create a strong predictive model, making it effective in capturing intricate sales patterns.*
Regression Methods:

***Linear Regression:** A foundational technique that assumes a linear relationship between the independent and dependent variables in sales data.*
***Quadratic and Polynomial Regression:** Extending beyond linearity, these methods accommodate non-linear trends, providing a more comprehensive understanding of sales dynamics.*
Feature Encoding:

One-Hot Encoding:** Widely used for categorical variables, one-hot encoding transforms categorical data into binary vectors, enabling ML algorithms to process and derive insights from such features**.
***Label Encoding:** Another encoding method for categorical variables, label encoding assigns unique numerical labels to different categories, facilitating algorithmic interpretation.*

Sales forecasting projects often encounter challenges related to feature representation and data encoding. One-hot encoding and label encoding address these challenges by transforming categorical variables into a format suitable for ML algorithms. The selection of an appropriate encoding method is crucial for ensuring accurate model training and prediction.

In conclusion, the literature on sales forecasting analysis reveals a rich landscape of regression methods and machine learning algorithms. Linear regression provides a foundational understanding, while advanced ML algorithms like Random Forest and Gradient Boosting offer sophisticated approaches to capturing complex sales dynamics. The incorporation of feature encoding techniques further enhances the capabilities of these models, ensuring robust and accurate sales forecasting in diverse business environments.

METHODOLOGY

DATA COLLECTION:

The dataset has been collected from Acmegrade Internship Program. The training dataset contains 12 columns and 8523 rows. The dataset contains 12 variables which includes : 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, Item_Outlet_Sales

```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
        import os
        import warnings
        warnings.filterwarnings('ignore')
        from matplotlib.pylab import rcParams
        rcParams['figure.figsize'] = 15, 6
```

```
In [5]: os.chdir('D:\Darjeeling')
```

```
In [6]: dt=pd.read_csv('Train.csv')
        display(dt.head())
```

	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	Item_Outlet_Sales
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.3800
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052

```
In [8]: dt.shape
```

```
Out[8]: (8523, 12)
```

```
In [16]: display(dt.columns)
```

```
Index(['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', 'Item_Outlet_Sales'],
      dtype='object')
```

DATA PREPROCESSING:

This step is an important step in data mining process. Because it improves the quality of the experimental raw data.

i)Removal of Null values:

In this step, the null values in the fields Product Category2 and Product Category3 are filled with the mean value of the feature.

ii) Converting Categorical values into numerical:

Machine learning deal with numerical values easily because of the machine readable form. Therefore, the categorical values like Product ID, Gender, Age and City Category are converted to numerical values.

Step1: Based on its datatype, we have selected the categorical values.

Step2: By using python, we have converting the categorical values into numerical values.

iii) Separate the target variable:

Here, we have to separate the target feature in which we are going to predict. In this case, purchase is the target variable.

Step1: The target lable purchase is assigned to the variable 'y'.

Step2: The preprocessed data except the target lable purchase is assigned to the variable 'X'.

iv) Standardize the features:

Here, we have to standardize the features because it arranges the data in a standard normal distribution. The standardization of the data is made only for training data most of the time because any kind of transformation of the features only be fitted on the training data.
Step1: Only trained data was taken.
Step2: By using the Standard Scaler API, we have standardize the features.

In [50]:
```
result= dt['Outlet_Size'].isnull().sum()
display (result)
```

```
2410
```

In [51]:
```
Outlet_Size_null= dt[dt['Outlet_Size'].isna()]
display (Outlet_Size_null)
```

	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	Item_Outlet_Sales
3	FDX07	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.3800
8	FDH17	16.200	Regular	0.016687	Frozen Foods	96.9726	OUT045	2002	NaN	Tier 2	Supermarket Type1	1076.5986
9	FDU28	19.200	Regular	0.094450	Frozen Foods	187.8214	OUT017	2007	NaN	Tier 2	Supermarket Type1	4710.5350
25	NCD06	13.000	Low Fat	0.099887	Household	45.9060	OUT017	2007	NaN	Tier 2	Supermarket Type1	838.9080
28	FDE51	5.925	Regular	0.161467	Dairy	45.5086	OUT010	1998	NaN	Tier 3	Grocery Store	178.4344
...	...	...	...	...	...	...	...	...	...	...	...	...
8502	NCH43	8.420	Low Fat	0.070712	Household	216.4192	OUT045	2002	NaN	Tier 2	Supermarket Type1	3020.0688
8508	FDW31	11.350	Regular	0.043246	Fruits and Vegetables	199.4742	OUT045	2002	NaN	Tier 2	Supermarket Type1	2587.9646
8509	FDG45	8.100	Low Fat	0.214306	Fruits and Vegetables	213.9902	OUT010	1998	NaN	Tier 3	Grocery Store	424.7804
8514	FDA01	15.000	Regular	0.054489	Canned	57.5904	OUT045	2002	NaN	Tier 2	Supermarket Type1	468.7232
8519	FDS36	8.380	Regular	0.046982	Baking Goods	108.1570	OUT045	2002	NaN	Tier 2	Supermarket Type1	549.2850

2410 rows × 12 columns

In [52]:
```
result = Outlet_Size_null['Outlet_Type'].value_counts()
display (result)
```

```
Supermarket Type1    1855
Grocery Store         555
Name: Outlet_Type, dtype: int64
```

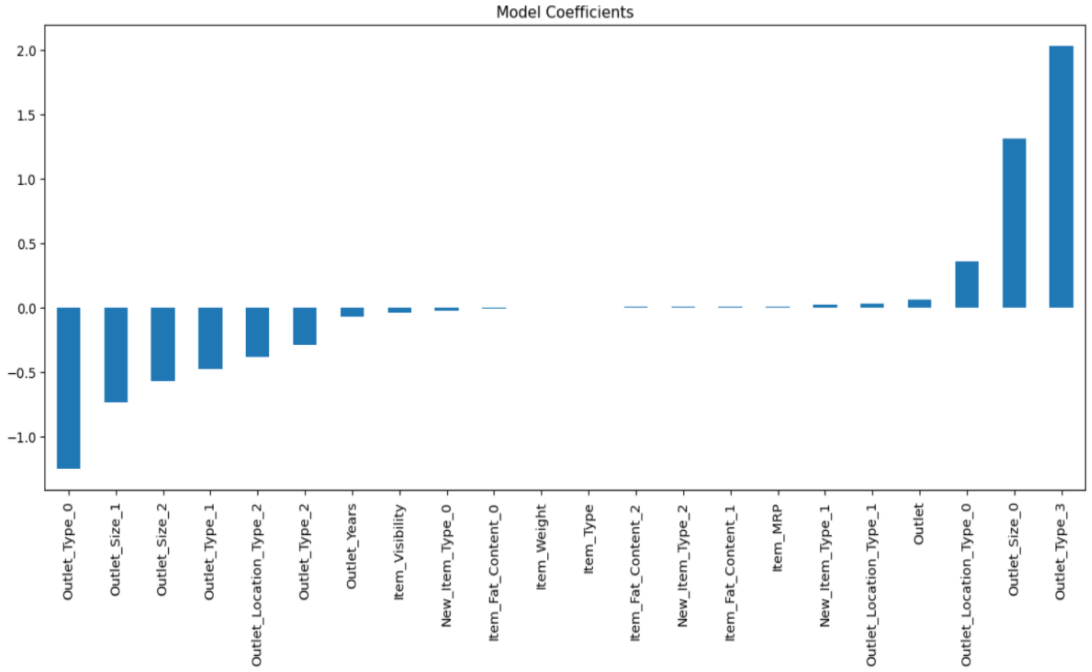
ALGORITHMS:

Linear Regression :

Linear Regression is one of the common ML and data analysis technique. This algorithm is helpful for forecasting based on linear regression equation. The Linear regression technique is the type of regression, which combines the set of independent features(x) to predict the output value(y) or dependent variable. The linear equation assigns a factor to each independent variable called coefficients represented by β.

[88]:
```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
model = LinearRegression()
train(model, X_train, y_train)
coef = pd.Series(model.coef_, X.columns).sort_values()
print (coef)
coef.plot(kind='bar', title="Model Coefficients")
plt.show()
```

```
Model Report
Scoring - neg_mean_squared_error
[-0.29269519 -0.27373286 -0.2864355  -0.28457789 -0.28152338]
ABS Average of - neg_mean_squared_error 0.283792963182744

R2 Score
[0.69537034 0.73192558 0.71637635 0.73456901 0.72785634]
Average R2 Score  0.7212195243740324

Accuracy for full Data
R2_Score: 0.7232721008270994

Outlet_Type_0              -1.254724
Outlet_Size_1              -0.734994
Outlet_Size_2              -0.575198
Outlet_Type_1              -0.476215
Outlet_Location_Type_2     -0.383612
Outlet_Type_2              -0.293316
Outlet_Years               -0.073222
Item_Visibility            -0.038578
New_Item_Type_0            -0.026521
Item_Fat_Content_0         -0.010478
Item_Weight                -0.001823
Item_Type                   0.000916
Item_Fat_Content_2          0.005111
New_Item_Type_2             0.005366
```

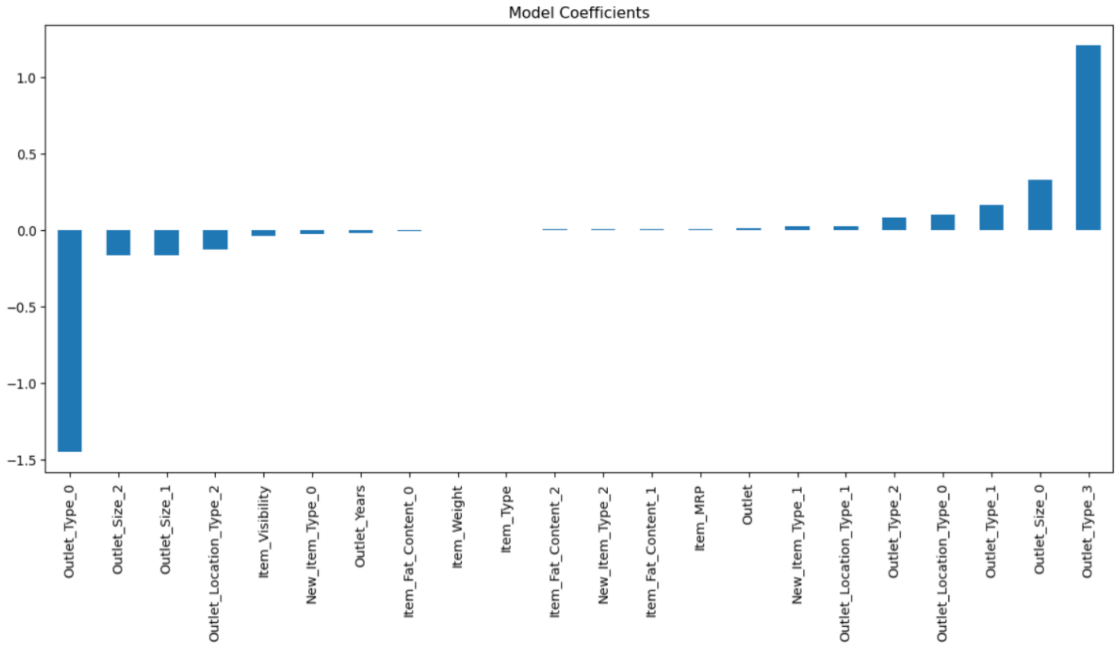


XGBoost:

XGBoost also known as Extreme Gradient Boosting has been used in order to get an efficient model with high computational speed and efficacy. The formula makes predictions using the ensemble method that models the anticipated errors of some decision trees to optimize last predictions. Production of this model also reports the value of each feature's effects in determining the last building performance score prediction.

[122]:
```
from xgboost import XGBRegressor
model = XGBRegressor()
train(model, X_train, y_train)
coef = pd.Series(model.feature_importances_, X.columns).sort_values(ascending=False)
coef.plot(kind='bar', title="Feature Importance")
plt.show()
```

```
Model Report
Scoring - neg_mean_squared_error
[-0.33259938 -0.30200845 -0.32479334 -0.31809611 -0.33556897]
ABS Average of - neg_mean_squared_error 0.32261324976392275

R2 Score
[0.65383908 0.70423448 0.67839506 0.70330595 0.67561142]
Average R2 Score  0.6830772007820543

Accuracy for full Data
R2_Score: 0.9107983430143505
```

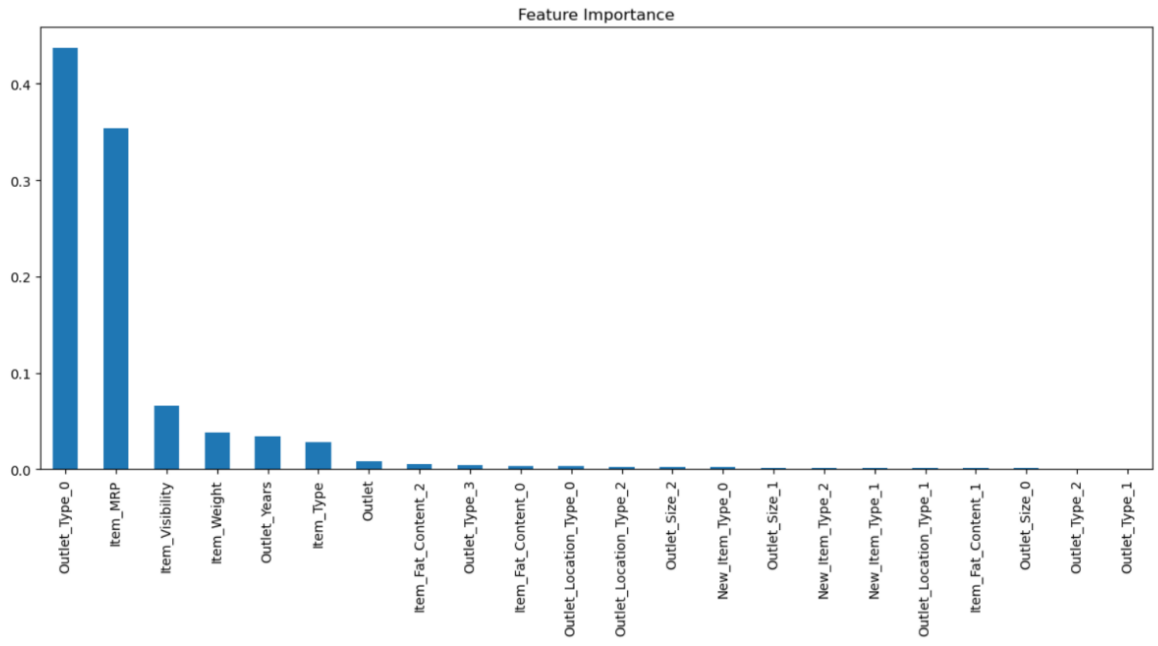


Decision Tree :

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

```
[1]: from sklearn.tree import DecisionTreeRegressor
     model = DecisionTreeRegressor()
     train(model,X_train, y_train)
     coef = pd.Series(model.feature_importances_, X.columns).sort_values(ascending=False)
     coef.plot(kind='bar', title="Feature Importance")
     plt.show()
```

```
Model Report
Scoring - neg_mean_squared_error
[-0.54601291 -0.56896155 -0.54305702 -0.5690841  -0.55170248]
ABS Average of - neg_mean_squared_error 0.5557636140667235

R2 Score
[0.43506157 0.45040182 0.47731491 0.47220611 0.46132597]
Average R2 Score  0.45926207659589824

Accuracy for full Data
R2_Score: 1.0
```

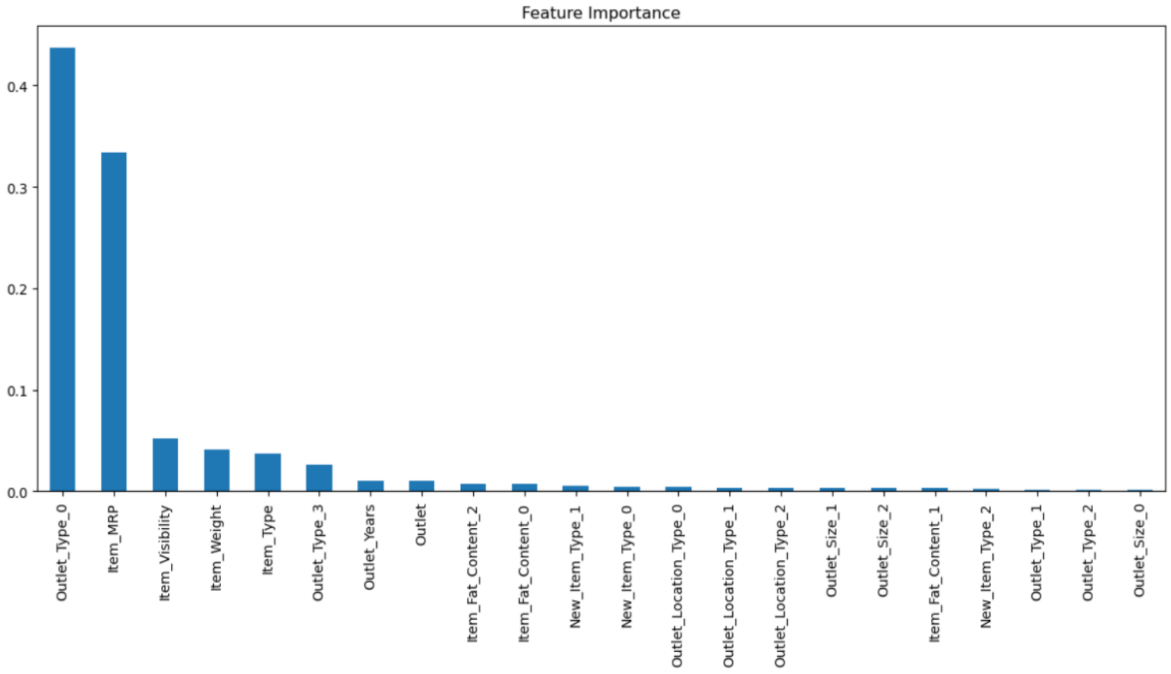


Random Forest:

Random forest is referred as a supervised machine learning ensemble method,which uses the multiple decision trees. It involves the technique called Bootstrap aggregation also known as bagging which aims to reduce the complexity of the models that overfit the training data . In this algorithm, rather than depending on individual decision tree it will combines the multiple decision trees to find the final outcome.

[83]:
```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
train(model, X_train, y_train)
coef = pd.Series(model.feature_importances_, X.columns).sort_values(ascending=False)
coef.plot(kind='bar', title="Feature Importance")
plt.show()
```

```
Model Report
Scoring - neg_mean_squared_error
[-0.30744359 -0.28507321 -0.30949252 -0.29234988 -0.29995312]
ABS Average of - neg_mean_squared_error 0.2988624648593961

R2 Score
[0.68096981 0.7189758  0.69606043 0.72812957 0.71008714]
Average R2 Score  0.7068445477370837

Accuracy for full Data
R2_Score: 0.9595951600602837
```

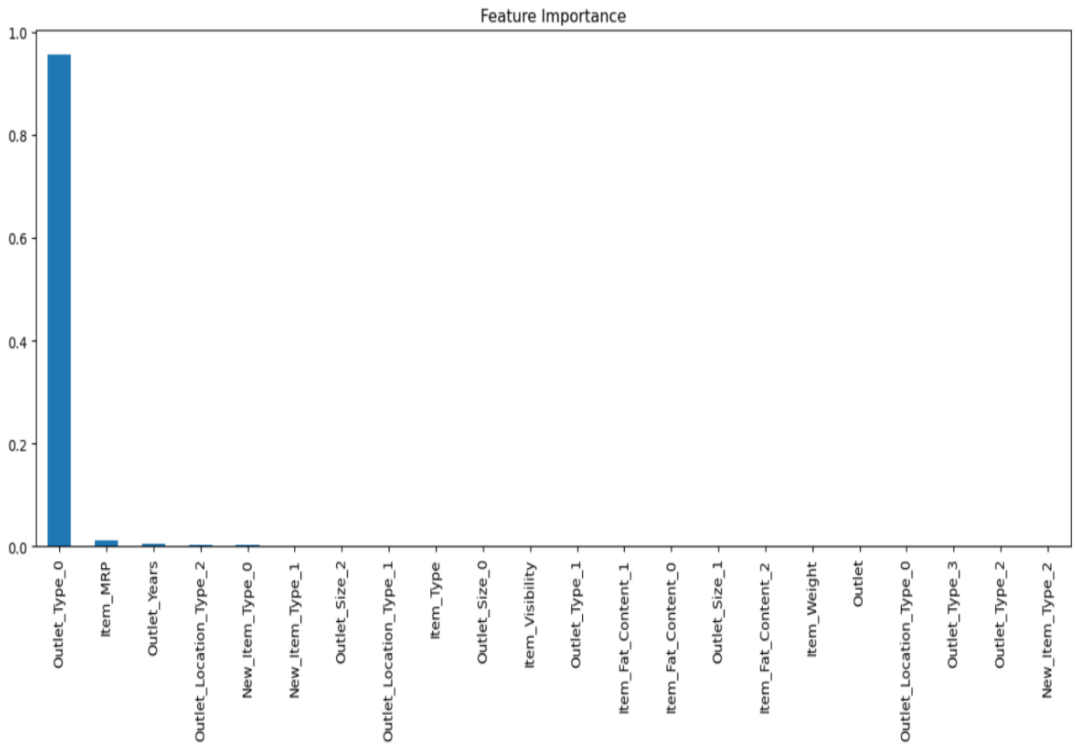


Extra Trees Algorithm:

This algorithm works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification.

[84]:
```
from sklearn.ensemble import ExtraTreesRegressor
model = ExtraTreesRegressor()
train(model, X_train, y_train)
coef = pd.Series(model.feature_importances_, X.columns).sort_values(ascending=False)
coef.plot(kind='bar', title="Feature Importance")
plt.show()
```

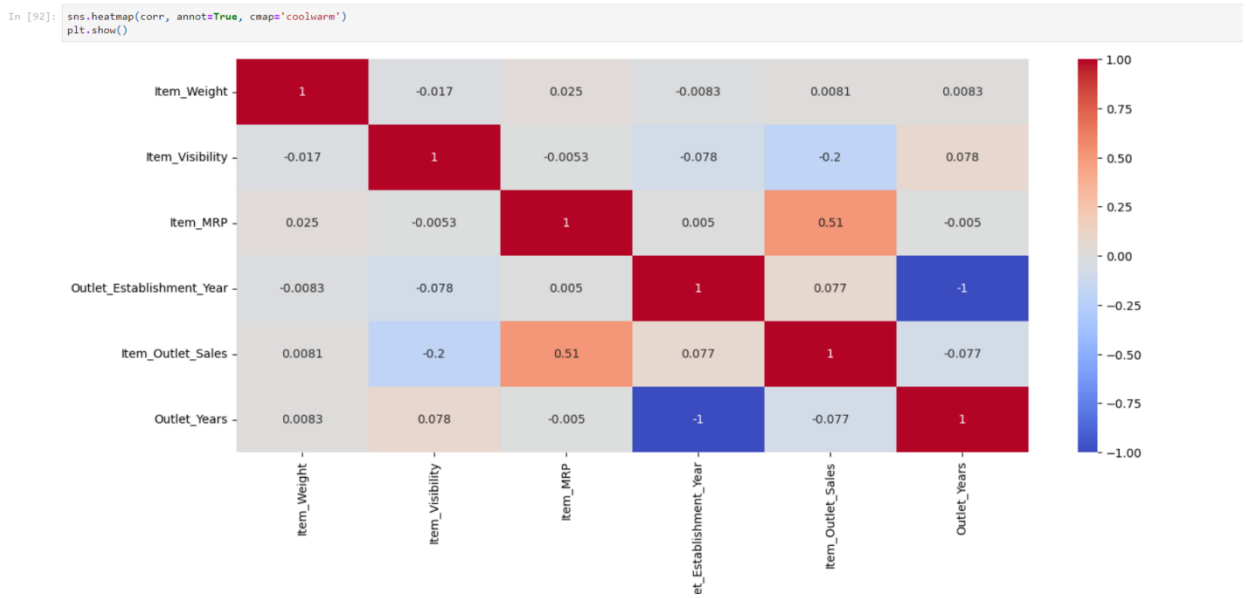
```
Model Report
Scoring - neg_mean_squared_error
[-0.33295772 -0.31767423 -0.32551031 -0.31860781 -0.32620673]
ABS Average of - neg_mean_squared_error 0.32419135982834524

R2 Score
[0.65156675 0.69055752 0.67709659 0.69951017 0.68619143]
Average R2 Score  0.6809844926458203

Accuracy for full Data
R2_Score: 1.0
```



Analysis and Findings

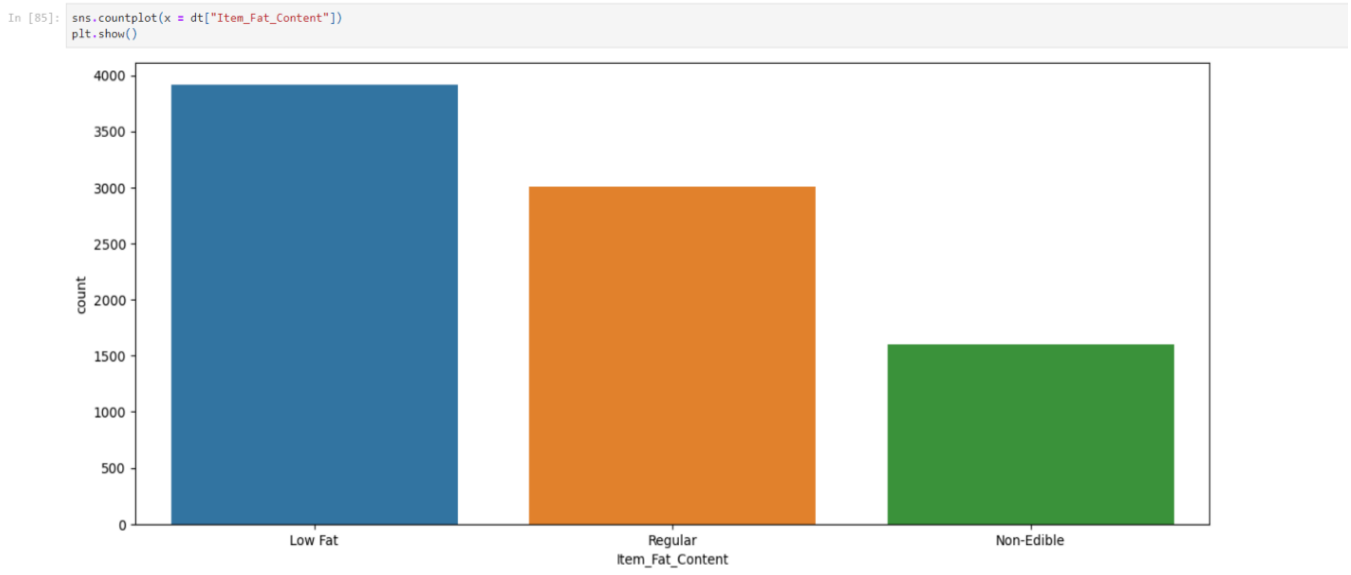
The evaluation of the machine learning algorithms is an essential part of any prediction model building. For that, we should carefully choose the evaluation metrics . These metrics are used to measure or judge the quality of the model. The performance of the machine learning algorithms are mainly focusing on accuracy. Companies uses the machine learning models with high accuracy for the practical business decisions.

Co-relation matrix Heatmap representation:



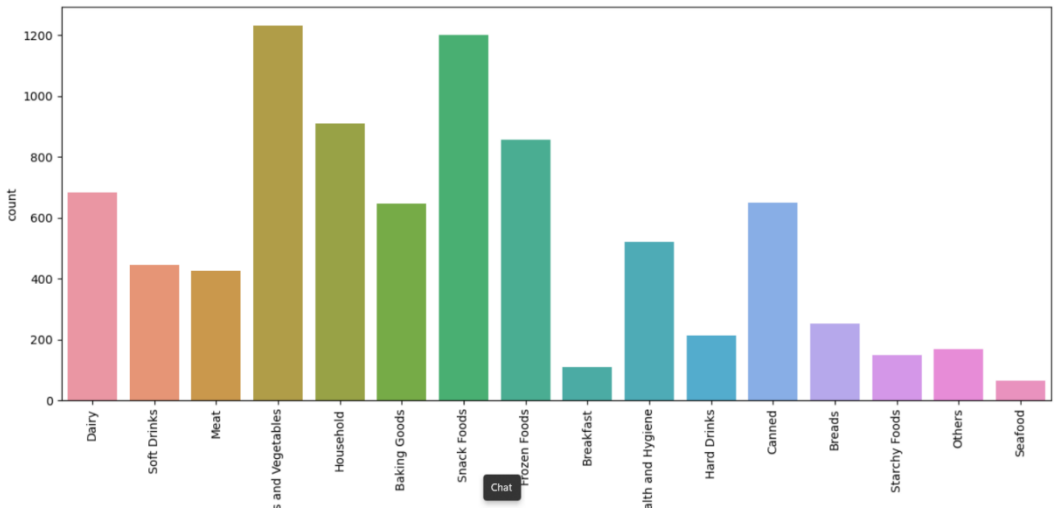
Some Analysis done as per requirements:

Count of Item as per Fat Content:



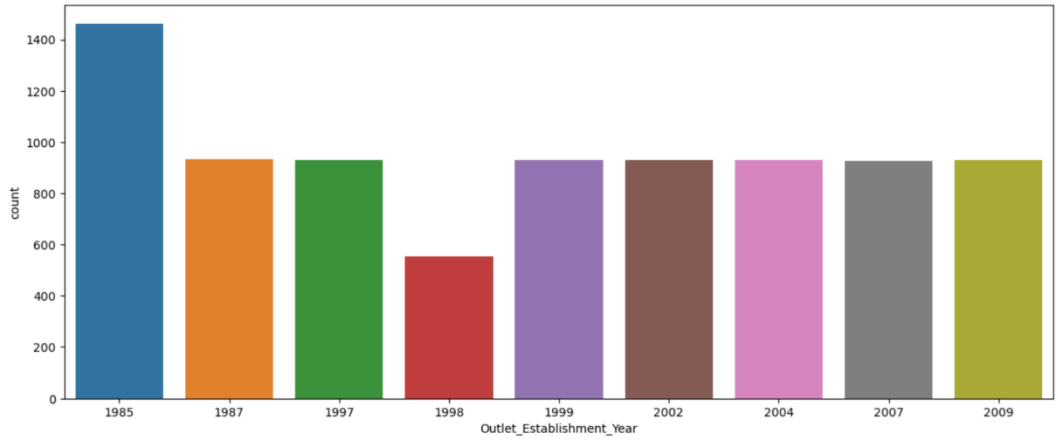
Count of Item as per Type:

```
In [86]: l = list(dt['Item_Type'].unique())
chart = sns.countplot(x =dt["Item_Type"])
chart.set_xticklabels(labels=l, rotation=90)
plt.show()
```



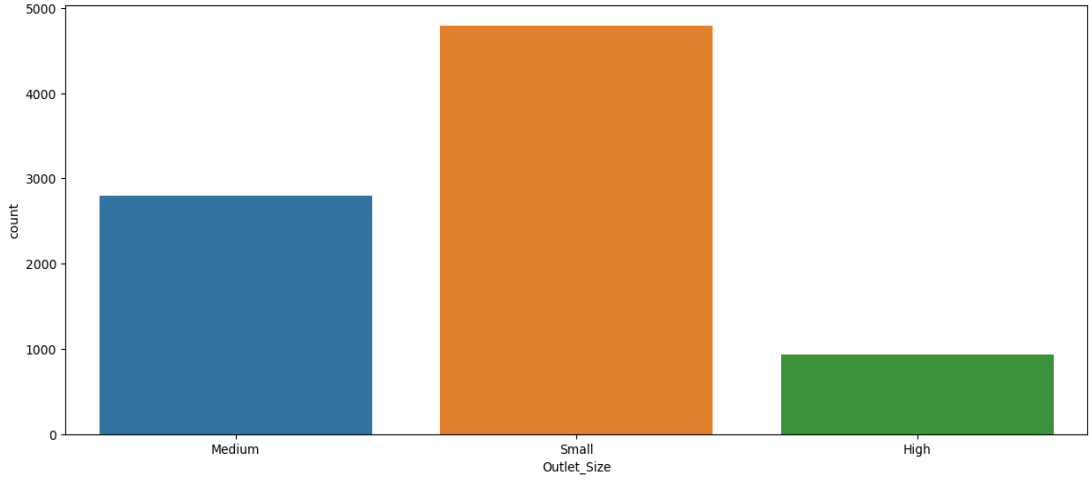
Count of Outlet as per Establishment Year:

```
In [87]: sns.countplot(x= dt['Outlet_Establishment_Year'])
plt.show()
```



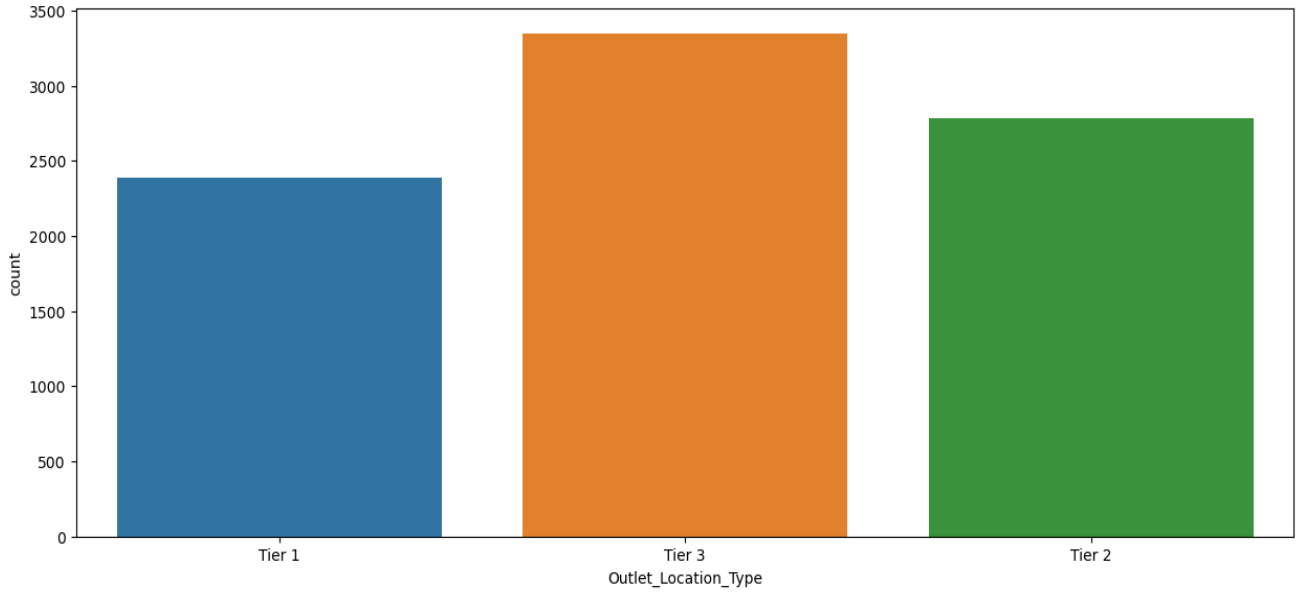
Count of Outlet as per Size:

```
In [88]: sns.countplot(x=dt['Outlet_Size'])
plt.show()
```



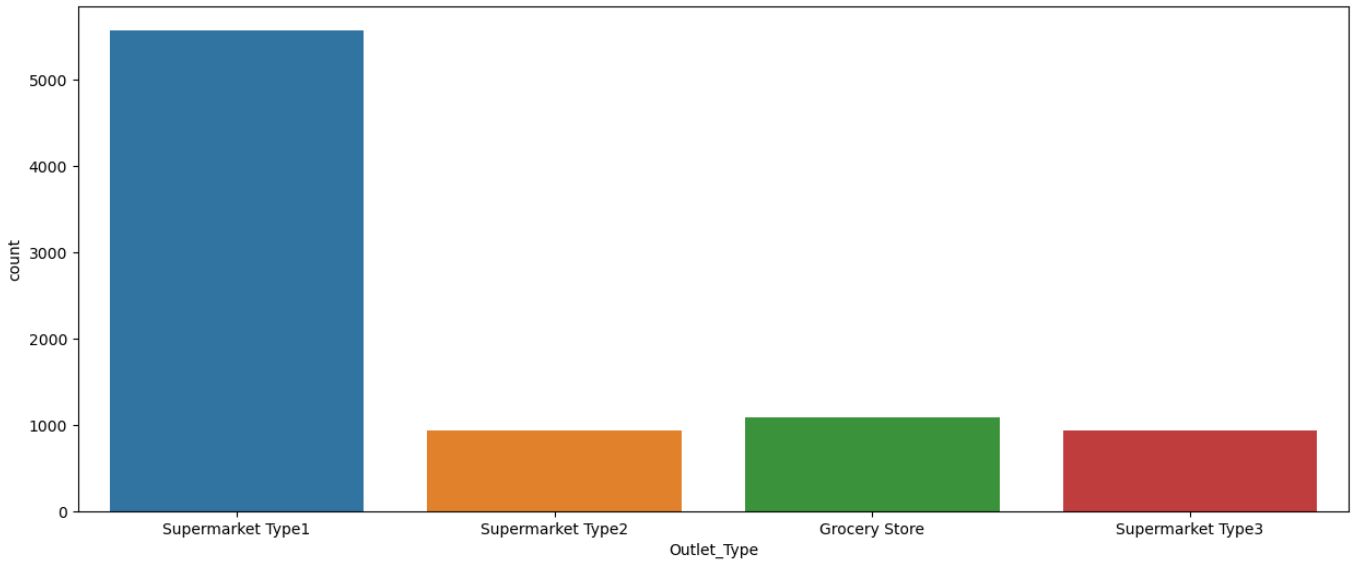
Count of Outlet as per Location Type:

```
sns.countplot(x=dt['Outlet_Location_Type'])
plt.show()
```



Count of Outlet as per Outlet Type:

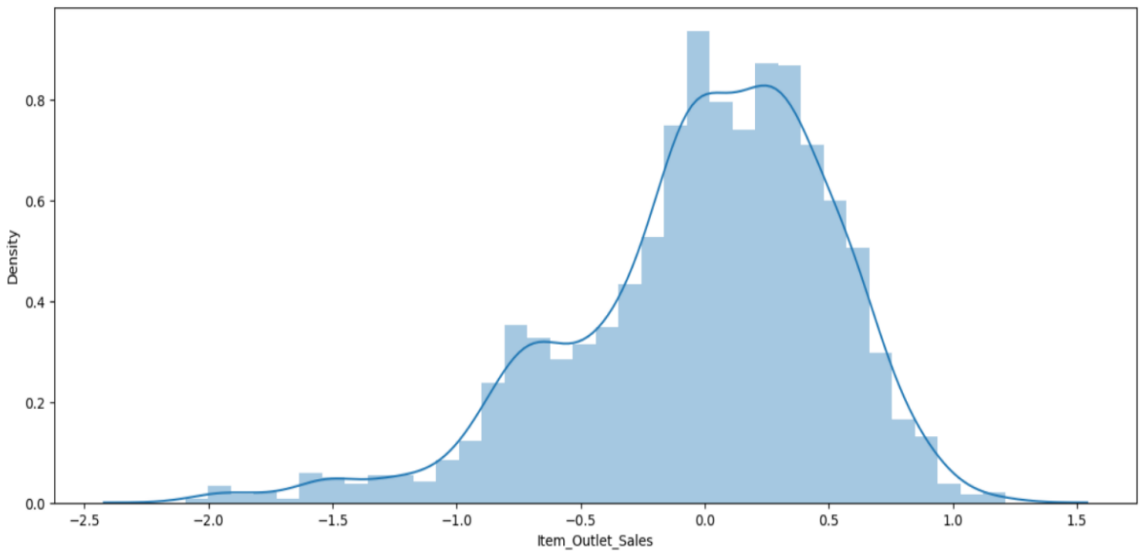
```
sns.countplot(x= dt['Outlet_Type'])
plt.show()
```



Based on the performance, we have concluded that the Extra Trees Regressor and Decision Tree Regressor algorithm considered as the best fit comparing to other algorithms. This comparative evaluation will help the organizations to choose the better and efficient machine-learning model.

Test Predictions:

```
In [144…  sns.distplot(y_test-predictions)
          plt.show()
```



CONCLUSION

Sales forecasting is mainly required for the organizations for business decisions. Accurate forecasting will help the companies to enhance the market growth. Machine learning techniques provides the effective mechanism in prediction and data mining as it overcome the problem with traditional techniques. These techniques enhances the data optimization along with improving the efficiency with better results and greater predictability. After predicting the purchase amount, the companies can apply some marketing strategies for certain sections of customers so that the profit could be enhanced.

***Future Scope**: In our future work, we will use the other feature selection techniques and advanced deep learning architecture algorithms to enhance the efficiency of the model with improved optimization.*

BIBLIOGRAPHY

References :

- *Bakri, R., Data, U., & Astuti, N. P. (2019). Aplikasi Auto Sales Forecasting Berbasis Computational Intelligence Website untuk Mengoptimalisasi Manajemen Strategi Pemasaran Produk. JURNAL SISTEM INFORMASI BISNIS, 9(2), 244. https://doi.org/10.21456/vol9iss2pp244-251*

- *Frees, E. W., & Miller, T. W. (2004). Sales forecasting using longitudinal data models. International Journal of Forecasting, 20(1), 99–114. https://doi.org/10.1016/s0169-2070(03)00005-0*

- *Geurts, M. D., & Patrick Kelly, J. (1986). Forecasting retail sales using alternative models. International Journal of Forecasting, 2(3), 261–272. https://doi.org/10.1016/0169-2070(86)90046-4*

- *Lauer, J., & O'Brien, T. (1988). SALES FORECASTING USING CYCLICAL ANALYSIS. Journal of Business & Industrial Marketing, 3(1), 25–35. https://doi.org/10.1108/eb006048*

- *Lu, C.-J., & Chang, C.-C. (2014). A Hybrid Sales Forecasting Scheme by Combining Independent Component Analysis with K-Means Clustering and Support Vector Regression. The Scientific World Journal, 2014, 1–8. https://doi.org/10.1155/2014/624017*

- *Murdick, K. (1996). Applications Short-Term Sales Forecasting. The Mathematics Teacher, 89(1), 48–52. https://doi.org/10.5951/mt.89.1.0048*

- *Rodrigues, A. (2021). Food Sales Forecasting Using Machine Learning Techniques: A Survey. International Journal for Research in Applied Science and Engineering Technology, 9(9), 869–872. https://doi.org/10.22214/ijraset.2021.38069*

- *Stormi, K., Laine, T., Suomala, P., & Elomaa, T. (2018). Forecasting sales in industrial services. Journal of Service Management, 29(2), 277–300. https://doi.org/10.1108/josm-09-2016-0250*

- *West, D. C. (1997). Managing Sales Forecasting. Management Research News, 20(4), 1–10. https://doi.org/10.1108/eb028556*

APPENDICES

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import os
import warnings
warnings.filterwarnings('ignore')
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 15, 6
display (os.getcwd())
os.chdir ('C:\\Noble\\Training\\Acmegrade\\Data Science\\Projects\\PRJ Sales Forecasting\\')
display (os.getcwd())
dt = pd.read_csv('Train.csv')
display (dt.head())
print (dt.shape)
display (dt.columns)
display (dt.describe())
display (dt.info())
display (dt.apply(lambda x: len(x.unique())))
display (dt.isnull().sum())
cat_col = []
for x in dt.dtypes.index:
    if dt.dtypes[x] == 'object':
        cat_col.append(x)
display (cat_col)
cat_col.remove('Item_Identifier')
cat_col.remove('Outlet_Identifier')
display (cat_col)
for col in cat_col:
print(col , len(dt[col].unique()))
for col in cat_col:
    print(col)
```

```
    print(dt[col].value_counts())
    print()
    print ('*' *50)
miss_bool = dt['Item_Weight'].isnull()
display (miss_bool)
display (dt['Item_Weight'].isnull().sum())
Item_Weight_null = dt[dt['Item_Weight'].isna()]
display (Item_Weight_null)
Item_Weight_null['Item_Identifier'].value_counts()
item_weight_mean = dt.pivot_table(values = "Item_Weight", index = 'Item_Identifier')
display (item_weight_mean)
display (dt['Item_Identifier'])
for i, item in enumerate(dt['Item_Identifier']):
    if miss_bool[i]:
        if item in item_weight_mean:
            dt['Item_Weight'][i] = item_weight_mean.loc[item]['Item_Weight']
        else:
            dt['Item_Weight'][i] = np.mean(dt['Item_Weight'])
result = dt['Item_Weight'].isnull().sum()
display (result)
result = dt.groupby('Outlet_Size').agg({'Outlet_Size': np.size})
display (result)
result= dt['Outlet_Size'].isnull().sum()
display (result)
Outlet_Size_null= dt[dt['Outlet_Size'].isna()]
display (Outlet_Size_null)
result = Outlet_Size_null['Outlet_Type'].value_counts()
display (result)
result= dt.groupby (['Outlet_Type','Outlet_Size'] ).agg({'Outlet_Type':[np.size]})
display (result)
outlet_size_mode = dt.pivot_table(values='Outlet_Size', columns='Outlet_Type', aggfunc=(lambda x: x.mode()[0]))
display (outlet_size_mode)
miss_bool = dt['Outlet_Size'].isnull()
dt.loc[miss_bool, 'Outlet_Size'] = dt.loc[miss_bool, 'Outlet_Type'].apply(lambda x: outlet_size_mode[x])
```

```
display (dt['Outlet_Size'].isnull().sum())
result = dt.groupby (['Outlet_Type','Outlet_Size'] ).agg({'Outlet_Type':[np.size]})
display (result)
display (sum(dt['Item_Visibility']==0))
dt.loc[:, 'Item_Visibility'].replace([0], [dt['Item_Visibility'].mean()], inplace=True)
sum(dt['Item_Visibility']==0)
display (sum(dt['Item_Visibility']==0))
dt['Item_Fat_Content'] = dt['Item_Fat_Content'].replace({'LF':'Low Fat', 'reg':'Regular', 'low fat':'Low Fat'})
result = dt['Item_Fat_Content'].value_counts()
display (result)
dt['New_Item_Type'] = dt['Item_Identifier'].apply(lambda x: x[:2])
display (dt['New_Item_Type'])
display (dt['New_Item_Type'].value_counts())
dt['New_Item_Type'] = dt['New_Item_Type'].map({'FD':'Food', 'NC':'Non-Consumable', 'DR':'Drinks'})
display (dt['New_Item_Type'].value_counts())
display (dt['Item_Fat_Content'].value_counts())
result = dt.groupby (['New_Item_Type','Item_Fat_Content'] ).agg({'Outlet_Type':[np.size]})
display (result)
dt.loc[dt['New_Item_Type']=='Non-Consumable', 'Item_Fat_Content'] = 'Non-Edible'
result =  (dt['Item_Fat_Content'].value_counts())
display (result)
result = dt.groupby (['New_Item_Type','Item_Fat_Content'] ).agg({'Outlet_Type':[np.size]})
display (result)
dt['Outlet_Years'] = 2022 - dt['Outlet_Establishment_Year']
print (dt['Outlet_Years'])
display (dt.head())
sns.distplot(dt['Item_Weight'])
plt.show()
sns.distplot(dt['Item_Visibility'])
plt.show()
sns.distplot(dt['Item_MRP'])
plt.show()
sns.distplot(dt['Item_Outlet_Sales'])
plt.show()
```

```
dt['Item_Outlet_Sales'] = np.log(1+dt['Item_Outlet_Sales'])
display (dt['Item_Outlet_Sales'])
sns.distplot(dt['Item_Outlet_Sales'])
plt.show()
sns.countplot(x = dt["Item_Fat_Content"])
plt.show()
l = list(dt['Item_Type'].unique())
chart = sns.countplot(x =dt["Item_Type"])
chart.set_xticklabels(labels=l, rotation=90)
plt.show()
sns.countplot(x= dt['Outlet_Establishment_Year'])
plt.show()
sns.countplot(x=dt['Outlet_Size'])
plt.show()
sns.countplot(x=dt['Outlet_Location_Type'])
plt.show()
sns.countplot(x= dt['Outlet_Type'])
plt.show()
corr = dt.corr()
display (corr)
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.show()
display (dt.head())
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dt['Outlet'] = le.fit_transform(dt['Outlet_Identifier'])
display (dt['Outlet'])
```

Full code visit this following link:-

https://github.com/mon0308/Sales-Forecasting.git

THANK YOU !!