ACKNOWLEDGEMENT

Working on the project was very inserting and really enhanced my knowledge.

I would like to express my sincere gratitude to my beloved **Principal Bhagawan Prasad** without whose permission, I would not be able to do my project and for taking keen interest for the students of Diploma in providing useful guidelines and giving all the necessary facilities.

I extend thanks to my Guide **Mr. Gagandeep M N, Lecturer and Project Guide** under Computer Department, Government Polytechnic Bantwal, for her valuable guidance and constant encouragement, which helped me in successfully completing my project.

Finally, I extend thanks to my parents and friends who were directly or indirectly involved in the completion of my project.

Place: BANTWAL ……..

Date: September 01, 2020 Nithesh shettigar

RegNo : 163CS21033

EXECUTIVE SUMMARY

An overview of my internship experience is given in this executive summary, which also highlights the important abilities, information, and successes I acquired while working for the company.

Internship gave me the chance to work in a stimulating and demanding atmosphere while contributing to a variety of projects and engaging with experts in the field. Through practical experience, I was able to learn practical skills that complemented my academic knowledge and a firm awareness of industry practises.

Technical expertise was one of the main things I needed to improve during my internship. I got to work with a variety of software tools and technologies, like Flask as front-end, Jupyter notebook as back-end. By honing my technical abilities and using them in practical situations, this experience helped me to improve my problem-solving abilities.

Throughout the internship, I actively engaged in various projects and successfully contributed to their completion. This included '**AMAZON REVIEW CLASSIFICATION'** and '**SALES PREDICTION'**, which allowed me to apply my knowledge, demonstrate initiative, and deliver results. These achievements not only contributed to the organization's goals but also enhanced my confidence and professional development.

This project report describes about two projects, on which I have worked during internship. The first project is regarding '**AMAZON REVIEW CLASSIFICATION',** which helped to learn how to build a simple sentiment analysis application. The second project is about '**SALES PREDICTION',** in that I have developed a website to see whether the predicting the sales.

INTERNSHIP CERTIFICATE ISSUED BY THE ORGANIZATION

CODELAB

02-05-2024

CERTIFICATE OF COMPLETION

This is to certify that **MR. NITHESH SHETTIGAR**, a bonafide student of **GOVERNMENT POLYTECHNIC, BANTWAL,** (Registration Number 163CS21033) has commendably completed an internship in **"AIRTIFICIAL INTELIGENCE AND MACHINE LEARNING"** at **CodeLab Systems** from 01st January 2024 to 30th April 2024.

We acknowledge and commend **MR. NITHESH SHETTIGAR**, for his significant contribution and commitment throughout the internship period at **CodeLab Systems.** We wish his continued success in all his future endeavors.

Mr. Mohammed Mehroos,
Lead – T & D,
CodeLab Systems,
Mangalore.

CODE LAB SYSTEMS
Ground Floor, Light House Condominium,
Light House Hill Road,
Mangaluru, Karnataka - 575001

codelabsystemsindia@gmail.com
Light House Condominium, Light House Hill Road,Bavutagudda
Mangaluru Karnataka 575001. Ph: +91 7349350390

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ NOTATIONS

SL NO.	ABBREVIATIONS/ NOTATIONS	DESCRIPTION
1.	VS Code	Visual studio code
2.	JS	Java script
3.	HTML	Hypertext Markup language
4.	PD	Pandas
6.	LE	LabelEncoder
6.	OJT	On JOB Training
7.	CSS	Cascading Style sheet
8	CV	TfidVectorizer

CHAPTER – 1
COMPANY PROFILE

COMPANY PROFILE

1.1 Overview of the company

Codelab System is a rapidly growing company in the field of computer application Implementation, solutions and services. Codelab System is a service provider of Web-based Development & Web based Software Development Solutions, Mobile Application Development, Graphic Design and Windows Applications. Codelab Systems is headquartered in Mangalore, with the Business development in UAE, Saudi Arabia and Qatar in a short span of 8+ years, our products as well as services & Solutions have been widely accepted by the global market. Today, Codelab Systems has the experience to undertake any IT development or deployment works on a single point responsibility basis. Our efficient and experienced team is greatest resource Intellect's infrastructure Houses A-team of young and competitive professionals having experience n Web Designing and Software Development who are dedicated to providing high-end Solution to our clients. We develop software and web-based applications with Latest Technologies. For web development projects, we also provide hosting And, domain Facility for customers, so they don't need to bother about that. Our, products and services are user friendly with easy controls and are of Superior specifications. We are always proactive to fulfill client's needs and requirements to the best possible extent of their satisfaction. We manage interactive sessions with clients throughout the Project development.

1.2 VISION AND MISSION OF THE ORGANIZATION

VISION:

To help people and businesses throughout the world realize their full potential.Codelab inspiring vision statement seeks to support people. You can see its intention isn't about business; it's about people and giving those people the services to be their best selves. With this aim, Codelab has numerous initiatives. It's a big supporter of inclusivity, diversity, environmental issues, and corporate responsibility. We are on a journey to be the trusted performance leader that unleashes the potential of data.

MISSION:

"To enable people and businesses throughout the world to realize their full potential and to organize the world's information and make it universally accessible and useful."

Codelab Systems provides customized package to suit the needs of every client and take into consideration the needs and requirements of each client's and plan different ideas to improve client's business strategies. Every customer satisfaction is our business and we pay special attention to each client. To provide best services. The main goal of our company is to provide best and innovative products that will help to drive potential customers to their business.

1.3ORGANIZATION STRUCTURE

An organization is a group of people who work together, like a neighborhood association, a charity, a union, or a corporation. You can use the word organization to refer to group or business, or to the act of forming or establishing something. Organizational structure (OS) is the systematic arrangement of human resources in an organization so as to achieve common business objectives. It outlines the roles and responsibilities of every member of the organization so that work and information flow seamlessly, ensuring the smooth functioning of an organization.

Types of Organizational Structure

- Hierarchical
- Flat
- Flatarchy
- Functional
- Divisional
- Matrix

In a flatarchy, there are little to no levels of management. A company using this structure could have only one manager in between its executive and all other employees. It is called a flatarchy because it is a hybrid of a hierarchy and a flat organization. This type of organizational structure is used more by smaller companies since they have fewer employees, though it can be used in companies of all sizes. While some companies grow out of this organizational structure, others continue to use it. Codelab systems have a Matrix organization structure, where teams report to multiple leaders. The matrix design keeps open communication between teams and can help companies create more innovative products and services. Using this structure prevents teams from needing to realign every time a new project begins.

1.4 ROLES AND RESPONSIBILITIES OF PERSONNEL IN THE ORGANIZATION

We have an expertise team that offer unique solutions. All the members of our team are professional, experienced and have in depth knowledge of the technology. Codelab Systems provides customized package to suit the needs of every client and take into consideration the needs and requirements of each clients and plan different ideas to improve client's business strategies. The

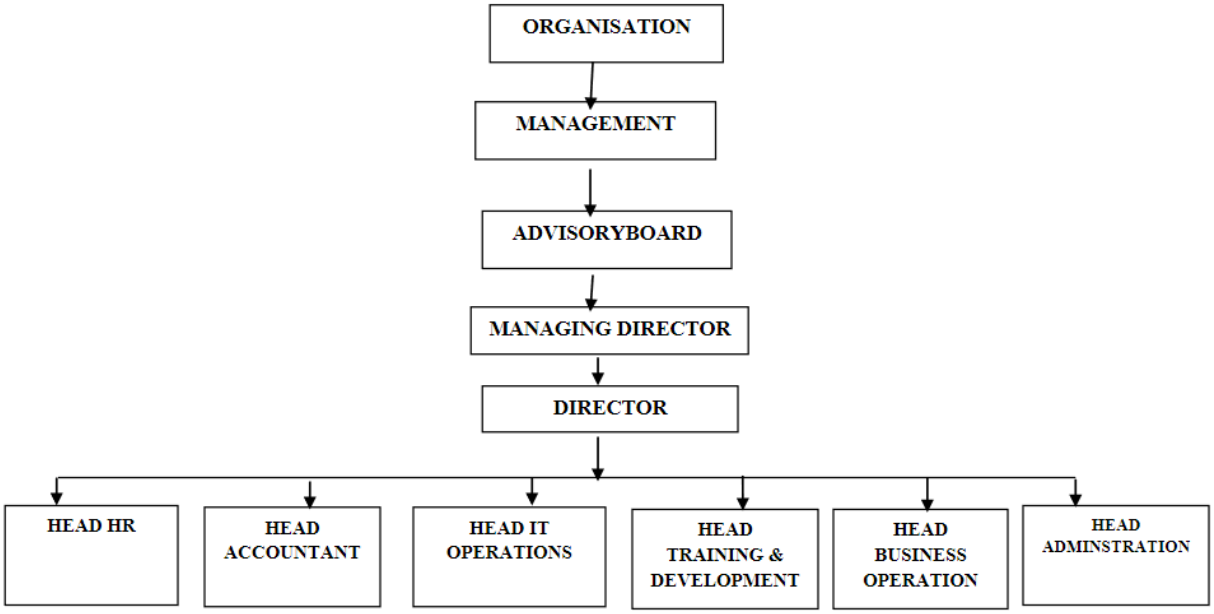


Figure 1.1 Organization Structure

Department of it operation:

Role: Head IT operation

Responsibilities:

• Development of clients line project

• Assigning tasks to the subordinate developers

• Managing client meetings and maintaining a good relationship on stakeholder

Department of Training and Development:

Role: Head of Training and Development

Responsibilities:

• Training to interns and newly joined employees & Work with new projects and domain

Department of HR:

Role: Head of HR

Responsibilities:

• Maintaining Employees data & payroll calculations

• Employee leave management & Interns internship program management.

Department of Account:

Role: Head of Account.

Responsibilities:

• Keep track of daily Account & Maintaining Balance sheet and Income tax procedure

Department of administration:

Role: Head Administration

Responsibilities:

• All the administration work such as file management, print, maintains data of computers and items. Arrangement of training program schedule.

Department of Business operation:

Role: Head Administration

Responsibilities:

• Conducting market research, Contact and approach clients for live projects.

• Communication with new clients and maintaining and managing social

1.5 PRODUCTS AND MARKET PERFORMANCE

- MSS LODGE(INDIA) : MSS LODGE is a budget property located in the beautiful city of Ujire.

- SESCO : SESCO is one of the first enterprises in the electrical equipment sector,
- QACADEMIA : Q-Academia providing wide range of career oriented IT Courses.

SERVICES : We believe in quality services

Web Development	**Promotion**
◎ CMS	◎ SMO/SEO
◎ E-Commerce	◎ RANKING
◎ Web Applications	◎ E-MARKETING
Professional Website	**Graphics Designing**
◎ Re-Design & Solution	◎Banner,Poster& Brochure.
◎ Design & Maintenance	◎ Business Card & Identity Card.
◎ E-Commerce & Forums.	◎ Logo, Letter *Head & Envelope.*
Mobile Application Platforms	**Others**
◎ Android Applications	◎ CSS Conversion
◎ Windows Applications	◎ Old website to new generation.
◎ App Store Optimization	◎Compitable with different Browser.

Table 1.2 Product And Market performance

CHAPTER- 2

ASSESSMENT OF ON JOB TRAINING – 1

CASE-1 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

1.1 INTRODUCTION:

As the commercial site of the world is almost fully undergone in online platform people is trading products through different e-commerce website. And for that reason reviewing products before buying is also a common scenario. Also now a day, customers are more inclined towards the reviews to buy a product. So analyzing the data from those customer reviews to make the data more dynamic is an essential field nowadays. In this age of increasing machine learning based algorithms reading thousands of reviews to understand a product is rather time consuming where we can polarize a review on particular category to understand its popularity among the buyers all over the world.

The objective of this paper is to categorize the positive and negative feedbacks of the customers over different products and build a supervised learning model to polarize large amounts of reviews. A study on amazon last year revealed over 88% of online shoppers trust reviews as much as personal recommendations. Any online item with large amounts of positive reviews provides a powerful comment of the legitimacy of the item. Conversely, books, or any other online item, without reviews puts potential prospects in a state of distrust. Quite simply, more reviews look more convincing. People value the consent and experience of others and the review on a material is the only way to understand others impression on the product. Opinions, collected from users' experiences regarding specific products or topics, straightforwardly influence future customer purchase decisions. Similarly, negative reviews often cause sales loss. For those understanding the feedback of customers and polarizing accordingly over a large amount of data is the goal. There are some similar works done over amazon dataset. In did opinion mining over small set of datasets of Amazon product reviews to understand the polarized attitudes towards the products.

In our model, we used both manual and active learning approach to label our datasets. In the active learning process different classifiers are used to provide accuracy until reaching satisfactory level. After getting satisfactory result we took those labeled datasets and processed it. From the processed dataset we extracted features that are then classified by different classifiers. We used combination of two kinds of approaches to extract features: the bag of words approach and tf-idf & Chi square approach for getting higher accuracy.

1.2 AIM:

The world we see nowadays is becoming more digitalized. In this digitalized world e-commerce is taking the ascendancy by making products available within the reach of customers where the customer doesn't have to go out of their house. As now a day's people are relying on online products so the importance of a review is going higher. For selecting a product, a customer needs to go through thousands of reviews to understand a product. But in this prospering day of machine learning, going through thousands of reviews would be much easier if a model is used to polarize those

reviews and learn from it. We used supervised learning method on a large scale amazon dataset to polarize it and get satisfactory accuracy.

1.3 OBJECTIVES:

- The objective of this research is to develop a supervised learning model that efficiently categorizes customer feedback into positive and negative sentiments for various products. In an era dominated by online commerce, where consumer decisions are heavily influenced by reviews, the need to streamline the analysis of vast amounts of feedback is paramount. By automating the sentiment polarization process, this study aims to provide consumers with valuable insights for informed purchasing decisions and assist businesses in understanding the reception of their products in the market
- To achieve this objective, a combination of manual and active learning approaches will be utilized for labeling datasets. Active learning involves iteratively employing different classifiers to improve accuracy until a satisfactory level is attained. The labeled datasets will then undergo processing to extract relevant features. Feature extraction will utilize two approaches: the bag of words approach and tf-idf & Chi-square approach. These features will serve as input for various classifiers, ensuring accurate sentiment classification.
- Ultimately, the goal is to develop a robust model capable of efficiently categorizing customer feedback, thereby empowering both consumers and businesses in their decision-making processes within the dynamic landscape of online commerce.

1.4 PERPOSE:

- The purpose of this research is to address the pivotal role of online product reviews in influencing consumer behavior and purchasing decisions in the digital marketplace. With the increasing reliance on e-commerce platforms for trading goods, the significance of customer feedback has surged, as shoppers often turn to reviews to guide their buying choices. However, the sheer volume of reviews poses a challenge for effective analysis and interpretation.
- This study seeks to develop a supervised learning model that can automatically categorize customer feedback into positive and negative sentiments for diverse products. By leveraging machine learning algorithms, the aim is to streamline the process of sentiment analysis, making it more efficient and scalable. The ultimate objective is to empower both consumers and businesses with actionable insights derived from large-scale review data.
- Through a combination of manual and active learning approaches, the research endeavors to label datasets accurately, ensuring the training of robust classifiers. Active learning methodologies will be employed iteratively to enhance classification accuracy, culminating in a model capable of effectively polarizing sentiments across a wide array of products. Feature extraction techniques, including the bag of words approach and tf-idf & Chi-square approach, will be utilized to capture the nuanced characteristics of customer feedback and improve classification performance.

- By understanding and categorizing the sentiments expressed in customer reviews, this study aims to provide valuable intelligence to consumers, enabling them to make informed purchasing decisions. Likewise, businesses stand to benefit from insights into the reception of their products in the market, facilitating strategic decision-
- making and product development efforts. Ultimately, the research endeavors to contribute to the optimization of online shopping experiences and the enhancement of consumer satisfaction in the digital age.

1.5 SCOPE:

The scope of this research encompasses the development and implementation of a supervised learning model aimed at categorizing customer feedback into positive and negative sentiments across various products within the realm of e-commerce. Given the widespread reliance on online platforms for trading goods and the growing importance of customer reviews in influencing purchasing decisions, the study focuses on analyzing large volumes of review data to derive actionable insights.

1.6 ADVANTAGES:

- The script preprocesses the raw text data by removing stop words and cleaning the reviews, which helps in improving the quality of the input data for analysis.
- Through bar charts and pie charts, the script provides a visual representation of the distribution of positive and negative sentiments in the dataset, allowing for quick insights into sentiment proportions.
- The word cloud visualizations generated for both positive and negative sentiment categories offer a concise representation of the most frequent words used in reviews, aiding in understanding the key themes and sentiments expressed by customers.
- By using TF-IDF vectorization, the script transforms text data into numerical vectors, capturing the importance of words in reviews relative to the entire corpus. This approach helps in representing text data effectively for machine learning algorithms.
- The logistic regression model is a simple yet effective algorithm for binary classification tasks like sentiment analysis. It offers interpretability and can handle large feature spaces efficiently.
- The script evaluates the performance of the sentiment analysis model using accuracy score, providing a quantitative measure of how well the model performs on unseen data.
- The trained logistic regression model is saved using pickle, allowing for easy deployment and reuse without the need for retraining.
- The script utilizes the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance in the dataset, ensuring better generalization of the model by generating synthetic samples for the minority class.

1.7 ACTIVITY DIAGRAM:

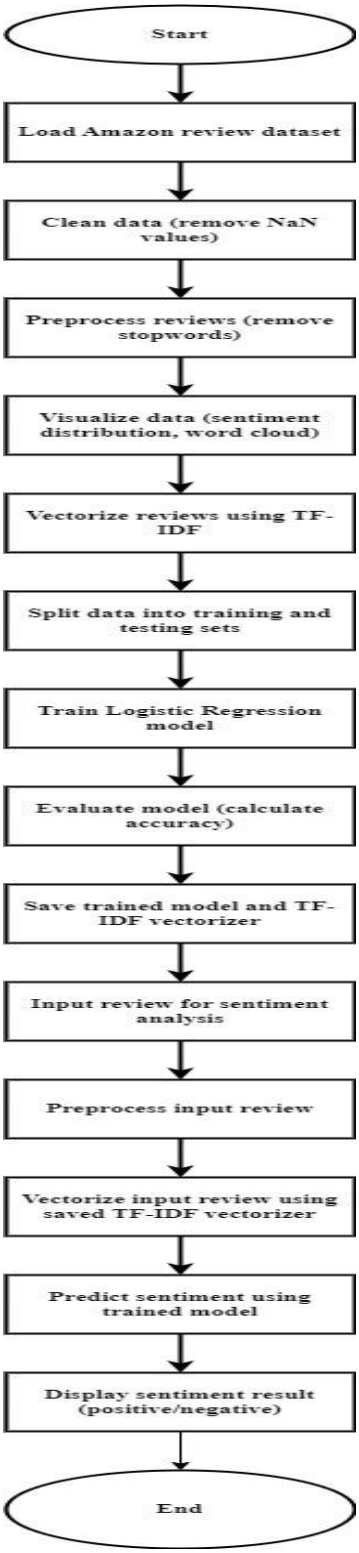


Figure 2 Activity diagram

1.8 MODULE:

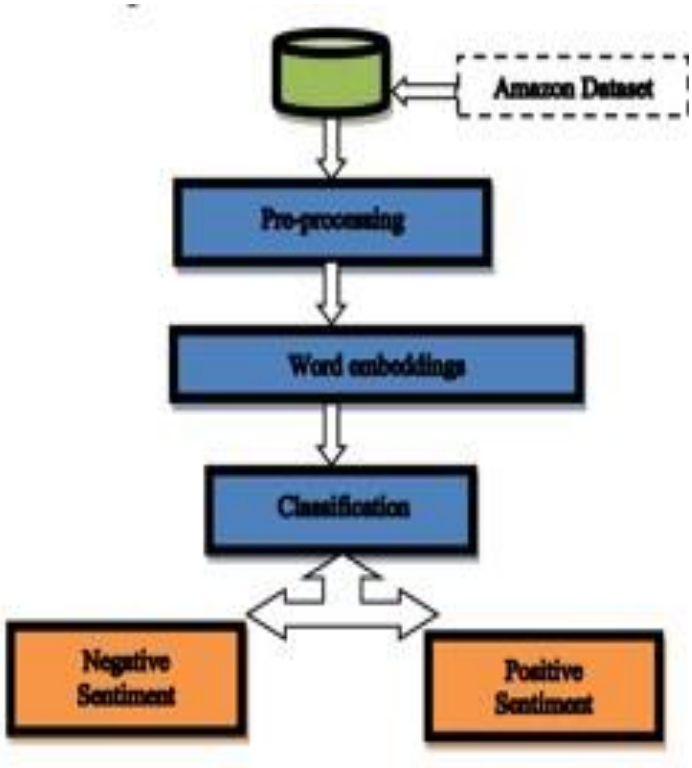


Fig 3 :Module

1.9 REQUERMENT SPECIFICATION

1.9.1 HARDWARE REQUERMENT:

- RAM: 8 GB or higher
- Storage: 256 GB SSD or higher
- Network: Ethernet/Wi-Fi for internet connectivity
- Display: 15-inch monitor or larger
- Processor: Intel Core i5 or equivalent

1.9.2 SOFTWARE REQUERMENT:

- Operating System: Windows 10 or Ubuntu 20.04 LTS
- Web Browser: Google Chrome or Mozilla Firefox
- Integrated Development Environment (IDE): Visual Studio Code, Python 3.8 or higher installed

1.6.1 LANGUAGES USED:

- Back End: HTML,CSS, JavaScript
- Front End:Python,Flask

1.7 CONCLUSION:

In conclusion, as online commerce continues to dominate global trade, the significance of customer reviews in influencing purchasing decisions has never been greater. Analyzing vast amounts of feedback to categorize and polarize sentiments towards products is essential in this landscape. With over 88% of online shoppers trusting reviews as much as personal recommendations, the volume and sentiment of reviews directly impact consumer trust and purchasing behavior. This study aimed to develop a supervised learning model to categorize positive and negative feedback, leveraging both manual and active learning approaches to label datasets. By employing various classifiers and feature extraction techniques, such as the bag of words and tf-idf & Chi-square methods, the model achieved satisfactory accuracy levels. Understanding and effectively polarizing customer feedback is crucial for businesses in maintaining consumer trust and competitiveness in the online marketplace.

ASSESSMENT OF ON JOB TRAINING – 2

CASE-2 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

1.1 INTRODUCTION:

Everyday competitiveness between various shopping centres as and as huge marts is becoming higher intense, violent just because of the quick development of global malls also online shopping. Each market seeks to offer personalized and limited-time deals to attract many clients relying on period of time, so that each item's volume of sales may be estimated for the organization's stock control, transportation and logistical services. The current machine learning algorithm is very advanced and provides methods for predicting or forecasting sales any kind of organization, extremely beneficial to overcome low – priced used for prediction. Always better prediction is helpful, both in developing and improving marketing strategies for the marketplace, which is also particularly helpful.

1.2 AIM:

Currently, supermarket run-centres, Big Marts keep track of each individual item's sales data in order to anticipate potential consumer demand and update inventory management. Anomalies and general trends are often discovered by mining the data warehouse's data store. For retailers like Big Mart, the resulting data can be used to forecast future sales volume using various machine learning techniques like big mart. A predictive model was developed using Xgboost, Linear regression, Polynomial regression, and Ridge regression techniques for forecasting the sales of a business such as Big -Mart, and it was discovered that the model outperforms existing models.

1.3 OBJECTIVE:

- Develop accurate forecasts to aid in strategic decision-making processes, such as inventory management, pricing strategies, and resource allocation.
- Utilize predictive insights to tailor marketing campaigns, promotions, and personalized offers that resonate with target audiences, driving customer engagement and loyalty.
- Streamline logistical operations by anticipating demand fluctuations and ensuring adequate stock levels, thereby minimizing stockouts, overstocking, and associated costs.
- Gain a competitive edge by leveraging advanced analytics to understand market trends, anticipate competitor moves, and capitalize on emerging opportunities swiftly.
- Utilize predictive analytics to anticipate customer preferences, optimize product assortment, and enhance the overall shopping experience, fostering customer satisfaction and retention.

- Reduce expenses associated with excess inventory, transportation, and storage through
  accurate demand forecasting, contributing to improved profitability and sustainability.
- Identify potential risks and uncertainties in the market landscape, enabling proactive risk management strategies to mitigate potential disruptions and adverse impacts on operations.
- Establish a feedback loop to continuously refine and optimize predictive models based on real-time sales data, market feedback, and evolving business dynamics.
- Foster collaboration among different departments, such as marketing, sales, operations, and IT, to leverage predictive analytics insights effectively across the organization.
- Position the organization for long-term growth and adaptability by harnessing data-driven insights to anticipate market shifts, consumer preferences, and technological advancements, enabling agile responses to changing business environments.

1.4 PURPOSE:

The purpose of this statement is to emphasize the heightened competition in the market due to the rapid expansion of global malls and online shopping platforms. This competition drives businesses to offer personalized and time-limited deals to attract customers, necessitating accurate sales forecasting for effective stock management, transportation, and logistical operations. The statement underscores the significance of advanced machine learning algorithms in predicting sales, enabling organizations to overcome challenges posed by low-priced competitors. Ultimately, the goal is to highlight the importance of precise sales predictions in enhancing marketing strategies and achieving success in the marketplace.

1.5 SCOPE:

The scope of this scenario encompasses the dynamic landscape of the global retail market, characterized by the rapid proliferation of both physical and online shopping avenues. It delves into the intense competition among businesses seeking to capture market share by offering personalized and time-limited deals to attract customers. The scope extends to the utilization of advanced machine learning algorithms for accurate sales prediction across various types of organizations, enabling efficient stock control, transportation, and logistical services. Additionally, it addresses the relevance of improved sales prediction in overcoming challenges posed by low-priced competitors and optimizing marketing strategies for sustained marketplace success.

1.6 ADVANTAGES:

- By considering both accuracy and R^2 score, you get a more comprehensive evaluation of the model's performance. Accuracy assesses the correctness of predictions, while the R^2 score measures how well the model's predictions fit the actual data.
- Accuracy focuses on the correctness of individual predictions, making it valuable for assessing the practical utility of the model in real-world scenarios.
- R^2 score, on the other hand, evaluates the goodness of fit of the model to the data. It provides insight into how well the independent variables explain the variability in the dependent variable.
- R^2 score quantifies the proportion of variability in the dependent variable that is predictable from the independent variables. This helps in understanding how much of the variation in the target variable the model captures.
- Accuracy can be influenced by class imbalance, where one class dominates the dataset. In such cases, the R^2 score provides an additional metric that is not affected by class distribution, offering a more balanced evaluation.
- While accuracy alone may not indicate overfitting, the R^2 score can help in identifying if the model is capturing the underlying patterns in the data or if it's just memorizing the training set.
- Visualizations like bar plots comparing accuracy and R^2 scores across different models provide a clear and concise summary of their performance, making it easier to identify the strengths and weaknesses of each model.

1.7 ACTIVITY DIAGRAM:

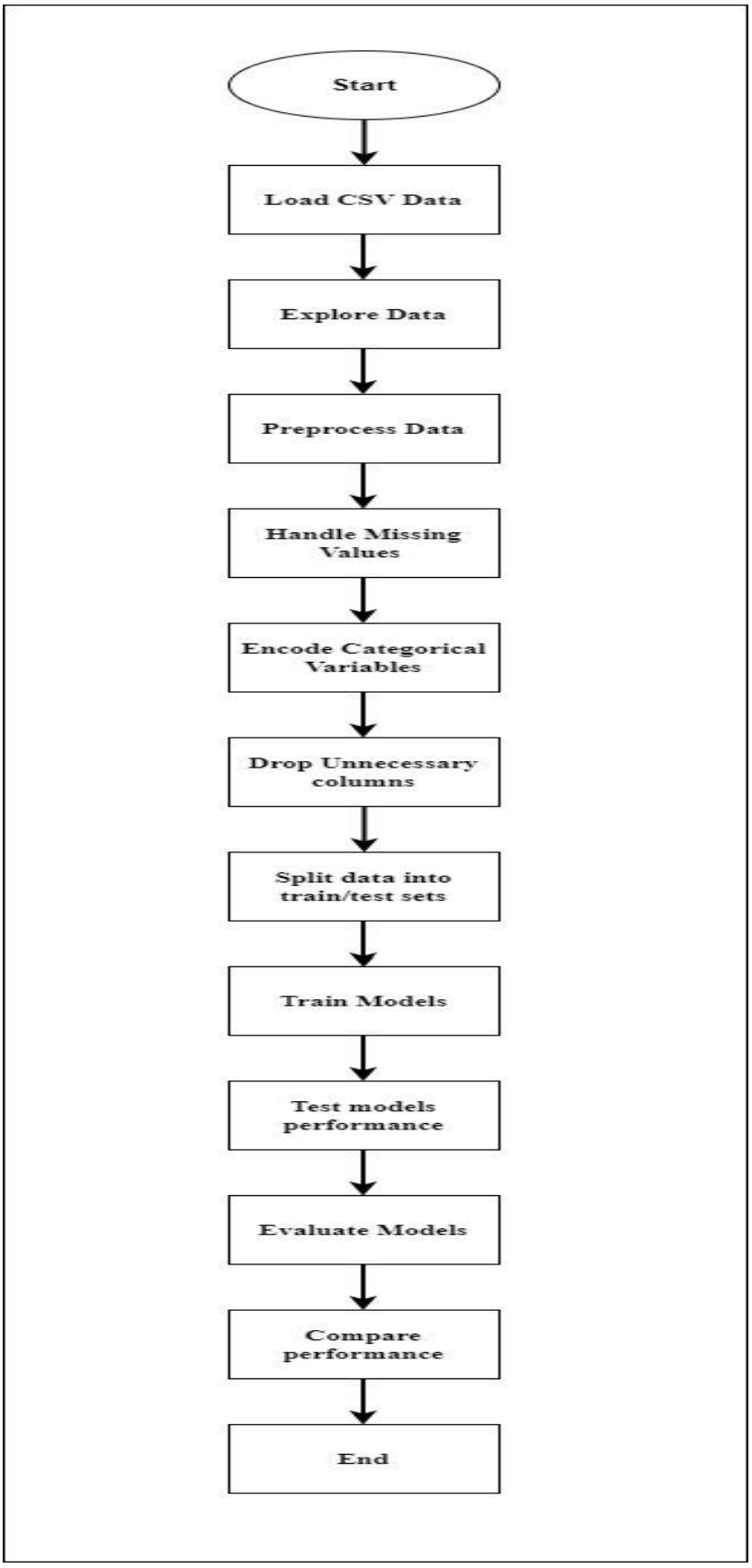


Fig 2.2 :Activity diagram

1.8 MODULE:

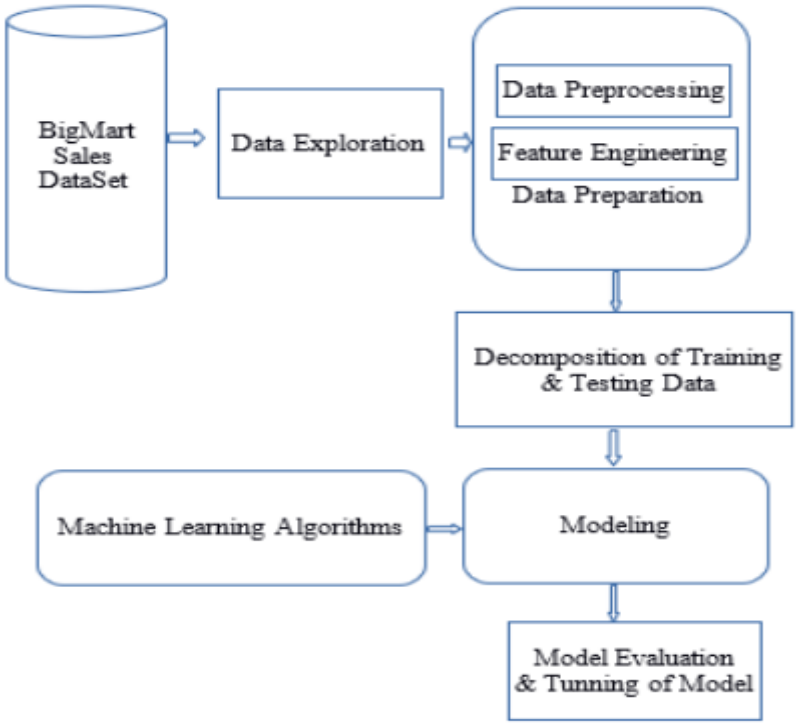


Fig 2.3 :Module diagram

1.10 REQUIREMENT SPECIFICATION:

1.10.1 HARDWARE REQUIREMENTS

- RAM: 8 GB or higher
- Storage: 256 GB SSD or higher
- Network: Ethernet/Wi-Fi for internet connectivity
- Display: 15-inch monitor or larger
- Processor: Intel Core i5 or equivalent

1.10.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 10 or Ubuntu 20.04 LTS
- Web Browser: Google Chrome or Mozilla Firefox
- Integrated Development Environment (IDE): Visual Studio Code, Python 3.8 or higher installed

1.10.3 LANGUAGES USED:

- Front-end: HTML, CSS, JavaScript
- Back-end: Python, Flask

1.11CONCLUSION:

The escalating competition among shopping centers and large retailers is being fueled by the rapid expansion of global malls and online shopping platforms. To stay ahead in this fiercely competitive landscape, businesses are leveraging advanced machine learning algorithms to offer personalized and time-sensitive deals to attract customers. These algorithms not only predict and forecast sales accurately but also assist in stock control, transportation, and logistical services. By continuously improving prediction accuracy, businesses can optimize their marketing strategies and enhance overall marketplace performance. In essence, embracing advanced machine learning techniques is essential for thriving in today's dynamic retail environment.

IMPLEMENTATION

Hypothesis generation with respect to problem statement

1)Item weight: Item weight might effect a sales of the product.
2)Items fat content: Sales of the product may be depends on the items fat content.
3)Item_Visibility: More Item_Visibility of a particular product may be costlier than other products.
5)Item type: Item type could have an effect on the sales.
6)Item MRP: Are the items with more MRP have more item outlet sales.
7)Stores established: Are the stores which have established earlier have more sales.
8)Size of the stores: Size of the stores could have an effect on the item sales at a particular store.
9)Location: Location of the stores might depends on the Item outlet sales.
10)Sales: Are the supermarkets have more sales than others.

Introduction to Dataset

Data We have train (8523) and test (5681) data set, train data set has both input and output variable(s). You need to predict the sales for test data set.

Variable	Definition
Item_Identifier	Unique Product ID
Item_Weight	Weight of product
Item_Fat_Content	Whether the product is low fat or not
Item_visibility	The % of total display area of all products in a store allocated to the particular product
Item_Type	The category to which the product belongs.
Item_MRP	Maximum retail price(list price) of the product.
Outlet_Identifier	Unique store ID.
Outlet_Establishment_Year	The year in which store was established.
Outlet_Size	The size of the store in terms of groud area covered.
Outlet_Location_Type	The type of city in which the store is located.
Outlet_Type	Whether the outlet is just a grocery store or some sort of supermarket
Item_Outlet_Sales	Sales of the product in the particulat store. This is the outcome variable to be predicted.

Loading Packages and Data

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

df_train = pd.read_csv("train.csv")
df_test = pd.read_csv("test.csv")

df_train.head()

df_train.head()
```

OUTPUT:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3

```
df_test.head()
```

OUTPUT:

```
df_test.head()
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type
0	FDW58	20.750	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	Medium	Tier 1
1	FDW14	8.300	reg	0.038428	Dairy	87.3198	OUT017	2007	NaN	Tier 2
2	NCN55	14.600	Low Fat	0.099575	Others	241.7538	OUT010	1998	NaN	Tier 3
3	FDQ58	7.315	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	NaN	Tier 2
4	FDY38	NaN	Regular	0.118599	Dairy	234.2300	OUT027	1985	Medium	Tier 3


```
print('Train Data',df_train.shape)
print('Test Data',df_test.shape)
```

OUTPUT:

Train Data (8523, 12)
Test Data (5681, 11)

Finding some basic information about the features of the data.

df_train.info()

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                6113 non-null   object
 9   Outlet_Location_Type       8523 non-null   object
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

Numerical Features

- Item_Weight
- Item_Visibility
- Item_MRP
- Item_Outlet_Sales(Target Variable)

Categorical Features

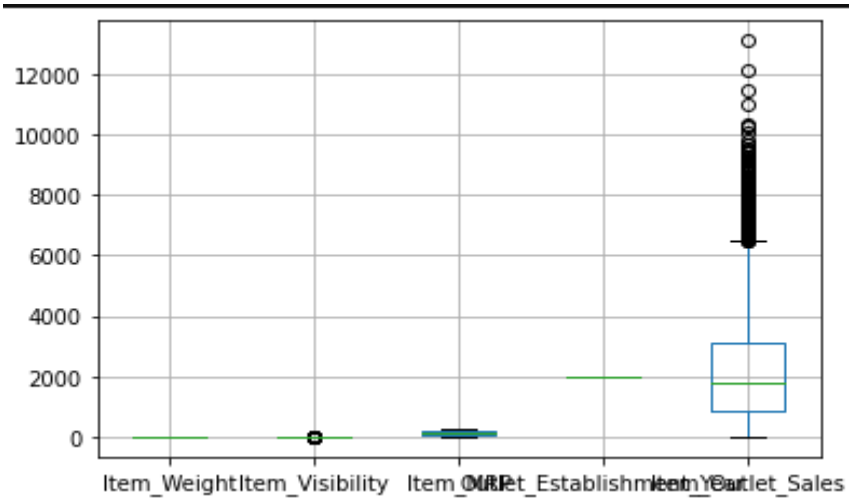
- Item_Identifier
- Item_Fat_Content(Ordinal Feature)
- Item_Type
- Outlet_Itemtifier
- Outlet_Establishment_Year
- Outlet_Size(Ordinal Feature)
- Outlet__Location_Type(Ordinal Feature)
- Ootlet_Type(Ordinal Feature)

Observations:

- There are 4 float type variables, 1 integer type and 7 object type.
- We are considering Item_Establishment_Year as a categorical feature because it contains some fixed value but not converting its data type now will consider later.
- Item_Fat_Content, Outlet_Size, Outelet_Location_Type and Outlet_Type are ordinal features because these values can be arranged in some order.

df_train.boxplot()

OUTPUT:



Univariate Analysis

print('Number of trainings examples:', len(df_train),'\n')
df_train.describe().T.style.background_gradient(cmap='Blues')

OUTPUT:

Number of trainings examples: 8523

```
print('Number of trainings examples:', len(df_train),'\n')
df_train.describe().T.style.background_gradient(cmap='Blues')
```

```
Number of trainings examples: 8523
```

	count	mean	std	min	25%	50%	75%	max
Item_Weight	7060.000000	12.857645	4.643456	4.555000	8.773750	12.600000	16.850000	21.350000
Item_Visibility	8523.000000	0.066132	0.051598	0.000000	0.026989	0.053931	0.094585	0.328391
Item_MRP	8523.000000	140.992782	62.275067	31.290000	93.826500	143.012800	185.643700	266.888400
Outlet_Establishment_Year	8523.000000	1997.831867	8.371760	1985.000000	1987.000000	1999.000000	2004.000000	2009.000000
Item_Outlet_Sales	8523.000000	2181.288914	1706.499616	33.290000	834.247400	1794.331000	3101.296400	13086.964800

List of numerical features:

numerical = df_train.select_dtypes(include = ['int64', 'Int64','float64']).dtypes.index
numerical

OUTPUT:

```
Index(['Item_Weight', 'Item_Visibility', 'Item_MRP',
       'Outlet_Establishment_Year', 'Item_Outlet_Sales'],
      dtype='object')
```

List of categorical features

cat_features = df_train.select_dtypes(include = ['object']).dtypes.index
cat_features

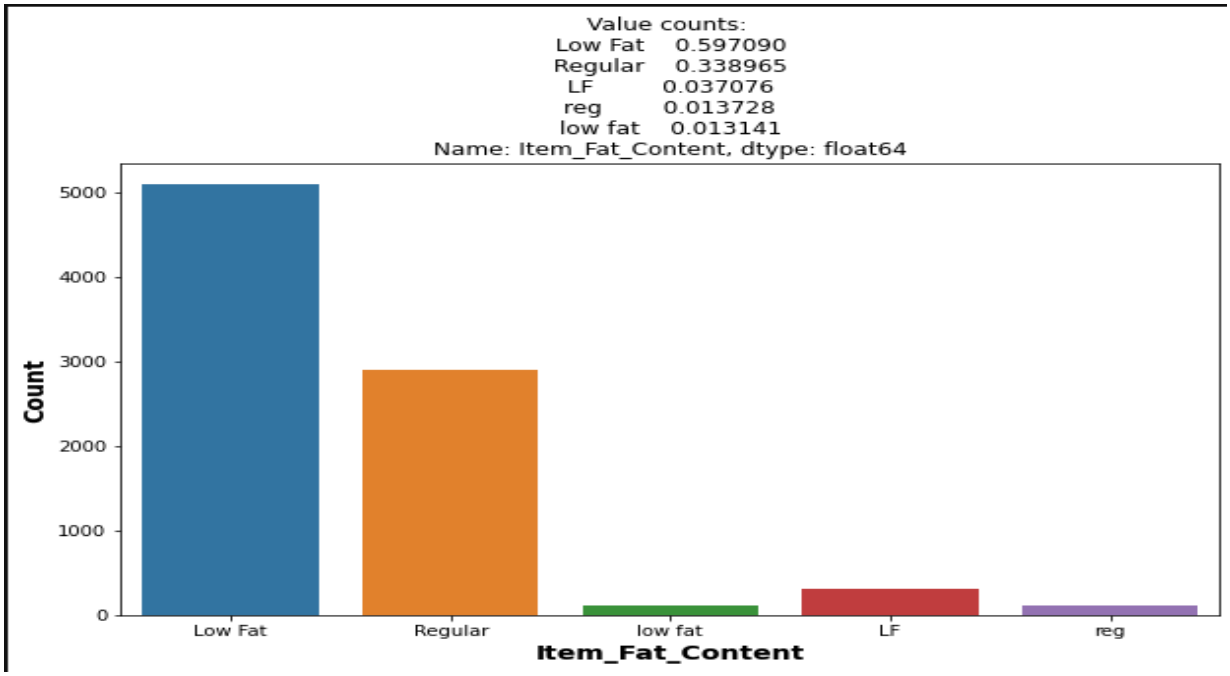
OUTPUT:

```
[10]: Index(['Item_Identifier', 'Item_Fat_Content', 'Item_Type', 'Outlet_Identifier',
       'Outlet_Size', 'Outlet_Location_Type', 'Outlet_Type'],
      dtype='object')
```

```
def UVA_Categorical(data, cat):
  plt.figure(figsize = (10,6))
  sns.countplot(cat, data = data)
  plt.xlabel(cat,fontsize = 14, fontweight = 'bold')
  plt.ylabel('Count',fontsize = 14, fontweight = 'bold')
  plt.title('Value counts: \n{}'.format(df_train[cat].value_counts(normalize = True)))

  # Rotating xticklabels
  if len(data[cat].value_counts()) > 7:
    X = plt.gca().xaxis
    for item in X.get_ticklabels():
      item.set_rotation(90)
  plt.show()
UVA_Categorical(df_train,'Item_Fat_Content')
```

OUTPUT:




```
total_low_fat = 0.597090 + 0.037076 + 0.013141
total_low_fat
```

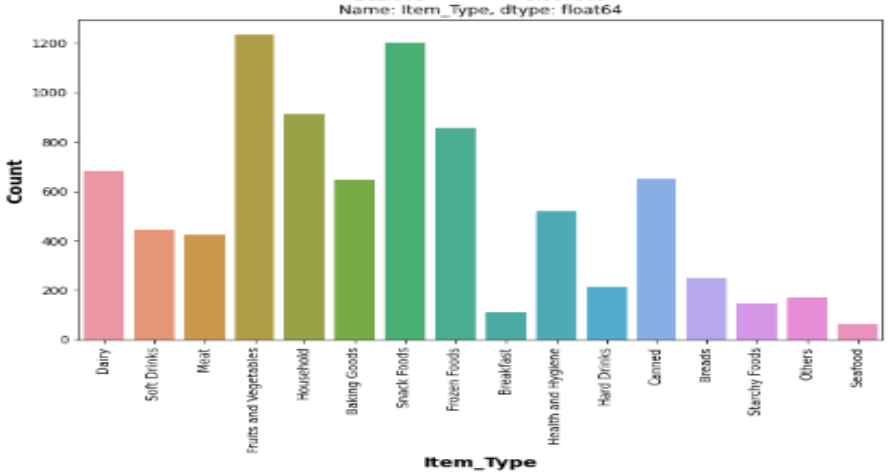
OUTPUT:

0.647307

UVA_Categorical(df_train, 'Item_Type')

OUTPUT:

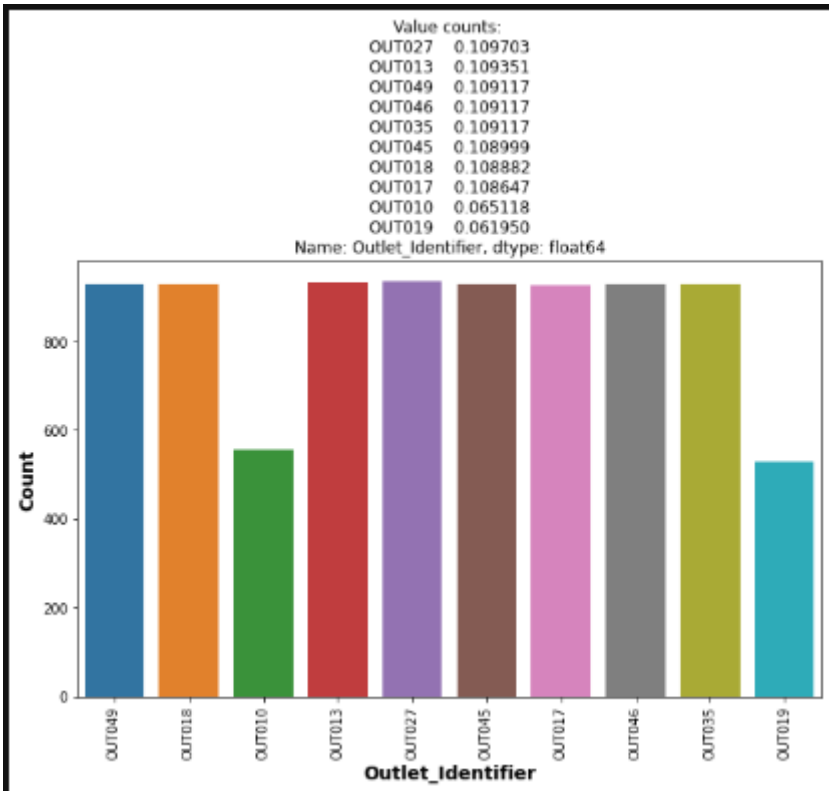
```
Value counts:
Fruits and Vegetables    0.144550
Snack Foods              0.140795
Household                0.106770
Frozen Foods             0.100434
Dairy                    0.080019
Canned                   0.076147
Baking Goods             0.076030
Health and Hygiene       0.061011
Soft Drinks              0.052212
Meat                     0.049865
Breads                   0.029450
Hard Drinks              0.025109
Others                   0.019829
Starchy Foods            0.017365
Breakfast                0.012906
Seafood                  0.007509
Name: Item_Type, dtype: float64
```



UVA_Categorical(df_train, 'Outlet_Identifier')

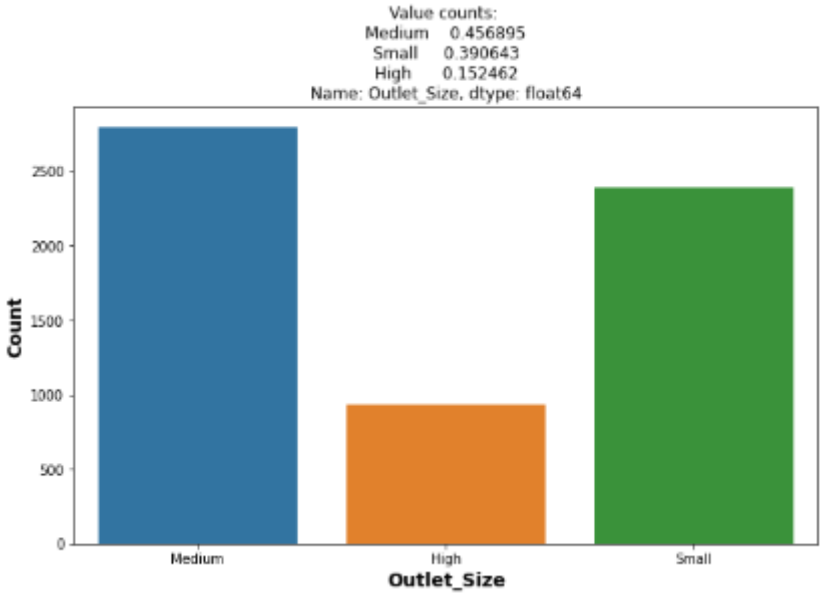
OUTPUT:

```
Value counts:
OUT027    0.109703
OUT013    0.109351
OUT049    0.109117
OUT046    0.109117
OUT035    0.109117
OUT045    0.108999
OUT018    0.108882
OUT017    0.108647
OUT010    0.065118
OUT019    0.061950
Name: Outlet_Identifier, dtype: float64
```



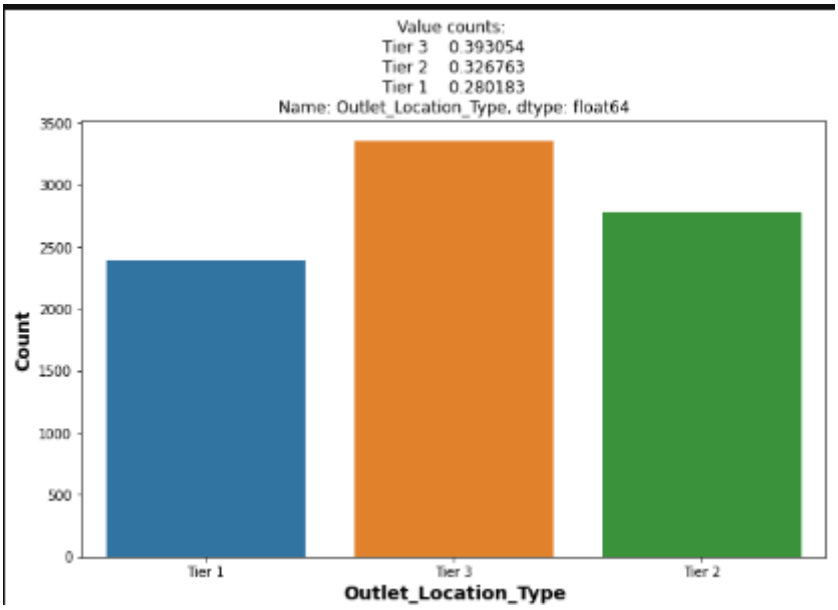
UVA_Categorical(df_train,'Outlet_Size')

OUTPUT:



UVA_Categorical(df_train, 'Outlet_Location_Type')

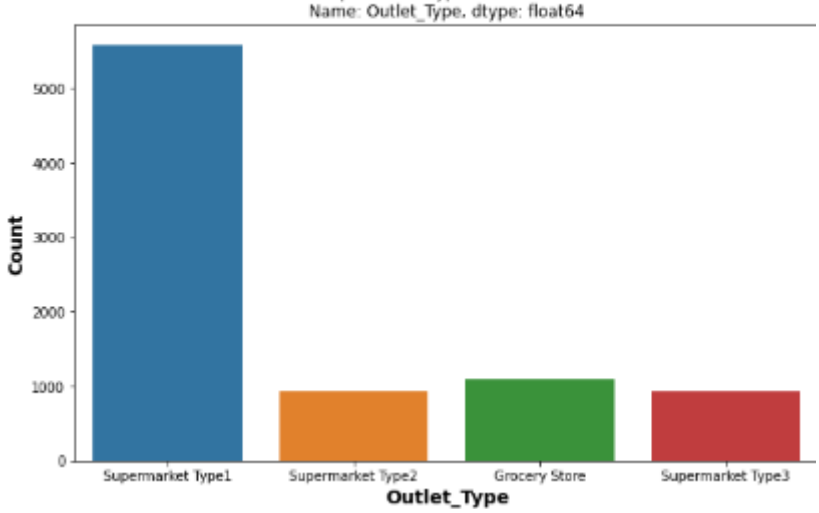
OUTPUT:



UVA_Categorical(df_train, 'Outlet_Type')

OUTPUT:

```
Value counts:
Supermarket Type1    0.654347
Grocery Store        0.127068
Supermarket Type3    0.109703
Supermarket Type2    0.108882
Name: Outlet_Type, dtype: float64
```



Concatinating df_train and df_test data

df_train['source'] = 'train'
df_test['source'] = 'test'
df=pd.concat([df_train,df_test], ignore_index=True)

Missing value treatment

df.info()

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 13 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            14204 non-null  object
 1   Item_Weight                11765 non-null  float64
 2   Item_Fat_Content           14204 non-null  object
 3   Item_Visibility            14204 non-null  float64
 4   Item_Type                  14204 non-null  object
 5   Item_MRP                   14204 non-null  float64
 6   Outlet_Identifier          14204 non-null  object
 7   Outlet_Establishment_Year  14204 non-null  int64
 8   Outlet_Size                10188 non-null  object
 9   Outlet_Location_Type       14204 non-null  object
 10  Outlet_Type                14204 non-null  object
 11  Item_Outlet_Sales          8523 non-null   float64
 12  source                     14204 non-null  object
dtypes: float64(4), int64(1), object(8)
memory usage: 1.4+ MB
```

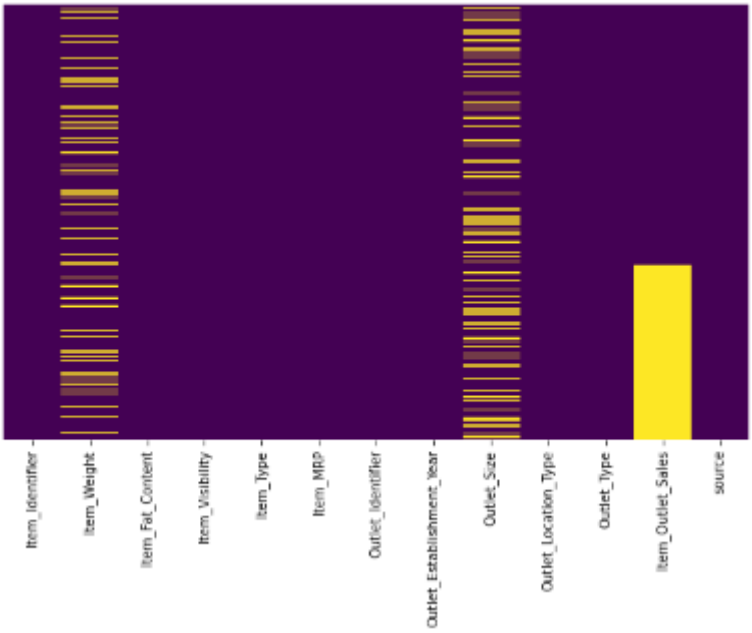
df.isnull().sum()

OUTPUT:

```
Item_Identifier               0
Item_Weight                2439
Item_Fat_Content              0
Item_Visibility               0
Item_Type                     0
Item_MRP                      0
Outlet_Identifier             0
Outlet_Establishment_Year     0
Outlet_Size                4016
Outlet_Location_Type          0
Outlet_Type                   0
Item_Outlet_Sales          5681
source                        0
dtype: int64
```

```
plt.figure(figsize = (10,6))
sns.heatmap(df.isnull(), yticklabels=False,cbar = False,cmap ='viridis')
```

OUTPUT:



Percentage of missing values:

```
def missing_percent():
  miss_item_weight = (df['Item_Weight'].isnull().sum()/len(df))*100
  miss_Outlet_Size = (df['Outlet_Size'].isnull().sum()/len(df))*100

  print('% of missing values in Item_Weight: ' + str(miss_item_weight))
  print('% of missing values in Outlet_Size: ' +str(miss_Outlet_Size))

missing_percent()
```

OUTPUT:

```
% of missing values in Item_Weight: 17.17121937482399
```

```
% of missing values in Outlet_Size: 28.273725711067303
```

Imputing Missing Values

Item_weight is numerical column so we fill it with mean imputation.

```
df['Item_Weight'].fillna(df['Item_Weight'].mean(),inplace=True)
```

Outlet_Size is categorical colums so we fill it with mode imputation. In this case "Medium"

```
df['Outlet_Size'].value_counts()
```

OUTPUT:

```
Medium    4655
Small     3980
High      1553
Name: Outlet_Size, dtype: int64
```

```
df['Outlet_Size'].fillna('Medium', inplace=True)
df.isnull().sum()
```

OUTPUT:

```
Item_Identifier                 0
Item_Weight                     0
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                     0
Outlet_Location_Type            0
Outlet_Type                     0
Item_Outlet_Sales             5681
source                           0
dtype: int64
```

Preprocessing

```
df.Item_Visibility.value_counts
```

OUTPUT:

```
<bound method IndexOpsMixin.value_counts of 0        0.016047
1        0.019278
2        0.016760
3        0.000000
4        0.000000
           ...
14199    0.013496
14200    0.142991
14201    0.073529
14202    0.000000
14203    0.104720
Name: Item_Visibility, Length: 14204, dtype: float64>
```

First replace 0 with nan values

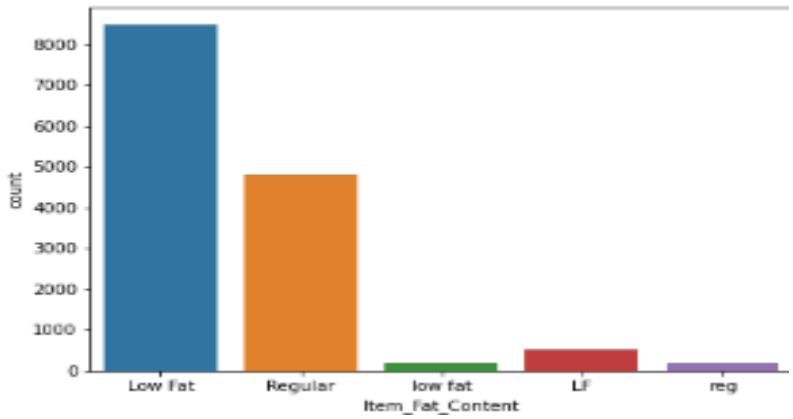
```
df['Item_Visibility'].replace(0.0,value=np.nan,inplace=True)
```

fill nan values with corresponding item identifier mean value

```
df['Item_Visibility']=df['Item_Visibility'].fillna(df.groupby('Item_Identifier')['Item_Visibility'].transform('mean'))

plt.figure(figsize=(7,5))
sns.countplot('Item_Fat_Content',data=df)
```

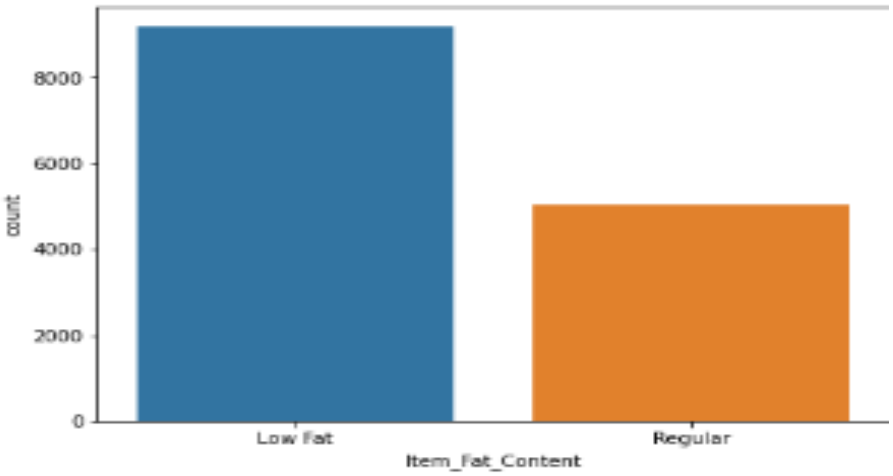
OUTPUT:




```
df['Item_Fat_Content'].replace({'LF':'Low Fat','reg':'Regular','low fat':'Low Fat'},inplace=True)

plt.figure(figsize=(7,5))
sns.countplot('Item_Fat_Content',data=df)
```

OUTPUT:



Storing Data

Store data for future prediction

```
test_pred = df.loc[df['source'] == 'test']
test_pred.drop(['Item_Outlet_Sales','source'],axis=1,inplace=True)
test_pred.head()
```

OUTPUT:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Ty
8523	FDW58	20.750000	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	Medium	Ti
8524	FDW14	8.300000	Regular	0.038428	Dairy	87.3198	OUT017	2007	Medium	Ti
8525	NCN55	14.600000	Low Fat	0.099575	Others	241.7538	OUT010	1998	Medium	Ti
8526	FDQ58	7.315000	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	Medium	Ti
8527	FDY38	12.792854	Regular	0.118599	Dairy	234.2300	OUT027	1985	Medium	Ti

Dealing with categorical variables

Label Encoder We will be converting all categorical variables into numeric types

```
from sklearn.preprocessing import LabelEncoder
categ = ['Item_Fat_Content', 'Item_Type', 'Outlet_Size', 'Outlet_Location_Type',
'Outlet_Type']
le = LabelEncoder()
df[categ] = df[categ].apply(le.fit_transform)

df.head()
```

OUTPUT:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type
0	FDA15	9.30	0	0.016047	4	249.8092	OUT049	1999	1	0
1	DRC01	5.92	1	0.019278	14	48.2692	OUT018	2009	1	2
2	FDN15	17.50	0	0.016760	10	141.6180	OUT049	1999	1	0
3	FDX07	19.20	1	0.022930	6	182.0950	OUT010	1998	1	2
4	NCD19	8.93	0	0.014670	9	53.8614	OUT013	1987	0	2

Droping less important variables ['Item_Identifier','Outlet_Identifier']

```
df.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
df.head()
```

OUTPUT:

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	9.30	0	0.016047	4	249.8092	1999	1	0	1	3735.1380
1	5.92	1	0.019278	14	48.2692	2009	1	2	2	443.4228
2	17.50	0	0.016760	10	141.6180	1999	1	0	1	2097.2700
3	19.20	1	0.022930	6	182.0950	1998	1	2	0	732.3800
4	8.93	0	0.014670	9	53.8614	1987	0	2	1	994.7052

Again splitting the train and test datasets into their original form as they were before

```
train = df.loc[df['source'] == 'train']
test = df.loc[df['source'] == 'test']

train.drop(['source'],axis=1,inplace=True)
test.drop(['Item_Outlet_Sales','source'],axis=1,inplace=True)

train.head()
```

OUTPUT:

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	9.30	0	0.016047	4	249.8092	1999	1	0	1	3735.1380
1	5.92	1	0.019278	14	48.2692	2009	1	2	2	443.4228
2	17.50	0	0.016760	10	141.6180	1999	1	0	1	2097.2700
3	19.20	1	0.022930	6	182.0950	1998	1	2	0	732.3800
4	8.93	0	0.014670	9	53.8614	1987	0	2	1	994.7052

```
test.head()
```

OUTPUT:

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type
8523	20.750000	0	0.007565	13	107.8622	1999	1	0	1
8524	8.300000	1	0.038428	4	87.3198	2007	1	1	1
8525	14.600000	0	0.099575	11	241.7538	1998	1	2	0
8526	7.315000	0	0.015388	13	155.0340	2007	1	1	1
8527	12.792854	1	0.118599	4	234.2300	1985	1	2	3

Spliting the train data into train and test with test size 20%

```
x = train.drop(columns="Item_Outlet_Sales")
y = train["Item_Outlet_Sales"]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y,test_size=0.2, random_state = 0)
```

create empty set to srore accuracies of all modes and later use for comparison

model_comparison = {}

Model Building

1)Linear Regression

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

from sklearn.linear_model import LinearRegression

lr = LinearRegression()

lr.fit(X_train,y_train)

OUTPUT:

```
LinearRegression()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
```

y_pred = lr.predict(X_test)
y_pred

OUTPUT:

```
array([2343.82994339, 2589.8484197 , 1996.40589285, ..., 3861.59171217,
       -343.54388197, 5374.57948081])
```

model_comparison['Linear Regression'] =
[lr.score(X_train,y_train)*100,r2_score(y_test,y_pred)*100]

print("Linear Regression\n\nAccuracy: {}%".format(round(lr.score(X_train,y_train)*100)))
print("r2 score: {}%".format(round(r2_score(y_test,y_pred)*100)))

OUTPUT:

```
Linear Regression

Accuracy: 51%
r2 score: 51%
```

2)Decision Tree Regressor

from sklearn.tree import DecisionTreeRegressor

tree = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
tree.fit(X_train,y_train)

```
y_pred = tree.predict(X_test)

y_pred
```

OUTPUT:

```
array([2517.36467547, 1365.38522314, 2207.87660699, ..., 4476.64235478,
        131.6409046 , 5965.31675472])
```

```
model_comparison['Decision Tree'] =
[tree.score(X_train,y_train)*100,r2_score(y_test,y_pred)*100]

print("Decision Tree\n\nAccuracy: {}%".format(round(tree.score(X_train,y_train)*100)))
print("r2 score: {}%".format(round(r2_score(y_test,y_pred)*100)))
```

OUTPUT:

```
Decision Tree

Accuracy: 62%
r2 score: 58%
```

3) Random Forest Regressor

```
from sklearn.ensemble import RandomForestRegressor

rf =
RandomForestRegressor(n_estimators=400,max_depth=6,min_samples_leaf=100,n_jobs=4)
rf.fit(X_train,y_train)

y_pred = rf.predict(X_test)

y_pred
```

OUTPUT:

```
array([2686.9974087 , 1336.69498214, 2133.38811621, ..., 3999.55694523,
        208.72111188, 5707.16302538])
```

```
model_comparison['Random Forest'] =
[rf.score(X_train,y_train)*100,r2_score(y_test,y_pred)*100]

print("Random Forest\n\nAccuracy: {}%".format(round(rf.score(X_train,y_train)*100)))
print("r2 score: {}%".format(round(r2_score(y_test,y_pred)*100)))
```

OUTPUT:

```
Random Forest

Accuracy: 61%
r2 score: 59%
```

4)XGBoost Regressor

```
from xgboost import XGBRegressor

xgb = XGBRegressor(n_estimators = 100, learning_rate=0.05)
xgb.fit(X_train, y_train)

y_pred = xgb.predict(X_test)

y_pred
```

OUTPUT:

```
array([2610.0571 , 1457.6395 , 2309.8242 , ..., 4400.283  ,  193.92038,
       5855.3677 ], dtype=float32)
```

```
model_comparison['XGBoost Regressor'] =
[xgb.score(X_train,y_train)*100,r2_score(y_test,y_pred)*100]

print("XGBoost Regressor\n\nAccuracy:
{}%".format(round(xgb.score(X_train,y_train)*100)))
print("r2 score: {}%".format(round(r2_score(y_test,y_pred)*100)))
```

OUTPUT:

```
XGBoost Regressor

Accuracy: 69%
r2 score: 59%
```

```
model_comparison_df = pd.DataFrame.from_dict(model_comparison).T
model_comparison_df.columns = ['Accuracy', "r2_score"]
model_comparison_df = model_comparison_df.sort_values('Accuracy', ascending=True)
model_comparison_df.style.background_gradient(cmap='Blues')
```

OUTPUT:

	Accuracy	r2_score
Linear Regression	50.682502	51.110030
Random Forest	60.831590	59.010965
Decision Tree	61.577041	57.792625
XGBoost Regressor	68.569191	58.618578

Test Data

test.head()

OUTPUT:

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type
8523	20.750000	0	0.007565	13	107.8622	1999	1	0	1
8524	8.300000	1	0.038428	4	87.3198	2007	1	1	1
8525	14.600000	0	0.099575	11	241.7538	1998	1	2	0
8526	7.315000	0	0.015388	13	155.0340	2007	1	1	1
8527	12.792854	1	0.118599	4	234.2300	1985	1	2	3

pred = xgb.predict(test)
pred

OUTPUT:

```
array([1556.0006 , 1307.0847 ,  643.95795, ..., 1834.3958 , 3509.5652 ,
       1345.8485 ], dtype=float32)
```

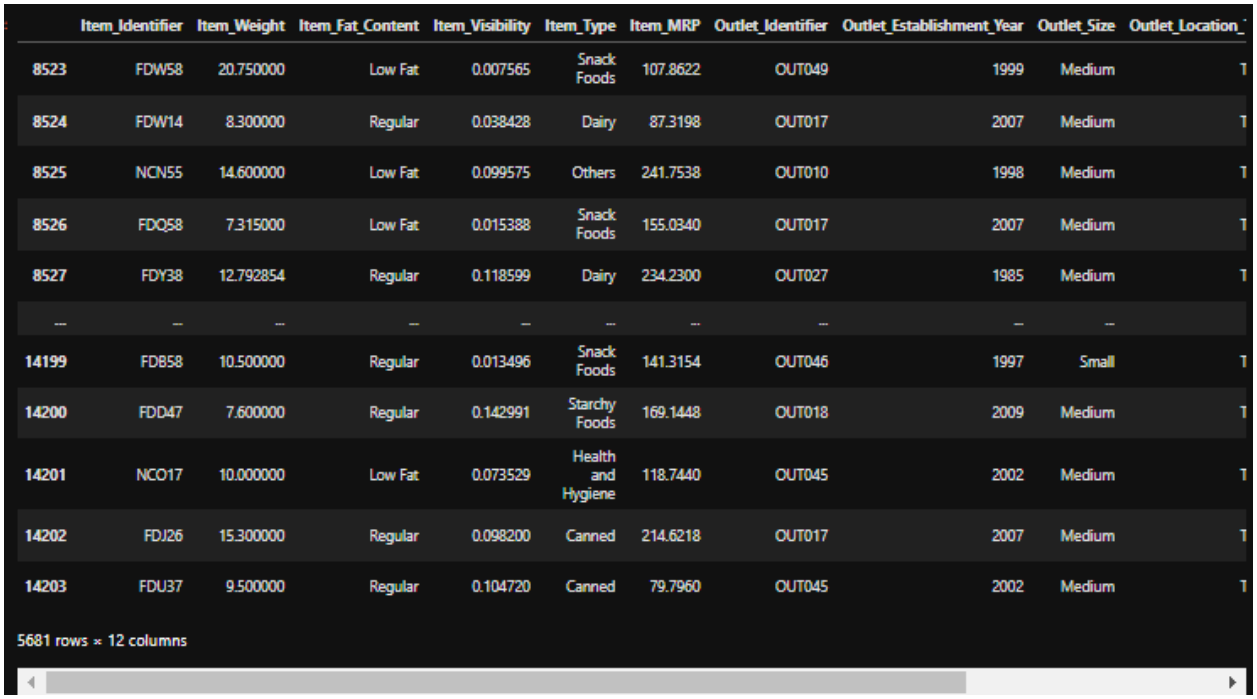
test_pred.head()

OUTPUT:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Ty
8523	FDW58	20.750000	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	Medium	Ti
8524	FDW14	8.300000	Regular	0.038428	Dairy	87.3198	OUT017	2007	Medium	Ti
8525	NCN55	14.600000	Low Fat	0.099575	Others	241.7538	OUT010	1998	Medium	Ti
8526	FDQ58	7.315000	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	Medium	Ti
8527	FDY38	12.792854	Regular	0.118599	Dairy	234.2300	OUT027	1985	Medium	Ti

test_pred["Predicted_Item_Outlet_Sale"] = pred
test_pred

OUTPUT:



	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_
8523	FDW58	20.750000	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	Medium	T
8524	FDW14	8.300000	Regular	0.038428	Dairy	87.3198	OUT017	2007	Medium	T
8525	NCN55	14.600000	Low Fat	0.099575	Others	241.7538	OUT010	1998	Medium	T
8526	FDQ58	7.315000	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	Medium	T
8527	FDY38	12.792854	Regular	0.118599	Dairy	234.2300	OUT027	1985	Medium	T
...	...	...	...	...	...	...	...	...	...	
14199	FDB58	10.500000	Regular	0.013496	Snack Foods	141.3154	OUT046	1997	Small	T
14200	FDD47	7.600000	Regular	0.142991	Starchy Foods	169.1448	OUT018	2009	Medium	T
14201	NCO17	10.000000	Low Fat	0.073529	Health and Hygiene	118.7440	OUT045	2002	Medium	T
14202	FDJ26	15.300000	Regular	0.098200	Canned	214.6218	OUT017	2007	Medium	T
14203	FDU37	9.500000	Regular	0.104720	Canned	79.7960	OUT045	2002	Medium	T

5681 rows × 12 columns

test_pred.to_csv("submission.csv",index=False)

```
ls=[]
for i in test.columns:
    s = float(input(f"Enter the {i}:"))
    ls.append(s)
```

OUTPUT:

```
Enter the Item_Weight:20.75
Enter the Item_Fat_Content:0
Enter the Item_Visibility:0.007565
Enter the Item_Type:13
Enter the Item_MRP:107.8622
Enter the Outlet_Establishment_Year:1999
Enter the Outlet_Size:1
Enter the Outlet_Location_Type:0
Enter the Outlet_Type:1
```

xgb.predict(np.array(ls).reshape(1,-1))

OUTPUT:

array([1556.0006], dtype=float32)

```
ls1=[]
for i in test.columns:
    s = float(input(f"Enter the {i}:"))
    ls1.append(s)
```

OUTPUT:

```
Enter the Item_Weight:8.3
Enter the Item_Fat_Content:1
Enter the Item_Visibility:0.038428
Enter the Item_Type:4
Enter the Item_MRP:87.3198
Enter the Outlet_Establishment_Year:2007
Enter the Outlet_Size:1
Enter the Outlet_Location_Type:1
Enter the Outlet_Type:1
```

xgb.predict(np.array(ls1).reshape(1,-1))

OUTPUT:

array([1307.0847], dtype=float32)

Saving the Model

import joblib
import pickle

joblib.dump(xgb,"BigMart_model.sav")
['BigMart_model.sav']

pickle.dump(xgb,open('BigMart_model.pkl','wb'))

CHAPTER -4
USE CASE-1 AND USE CASE-2

USE CASE-1 DIAGRAM:

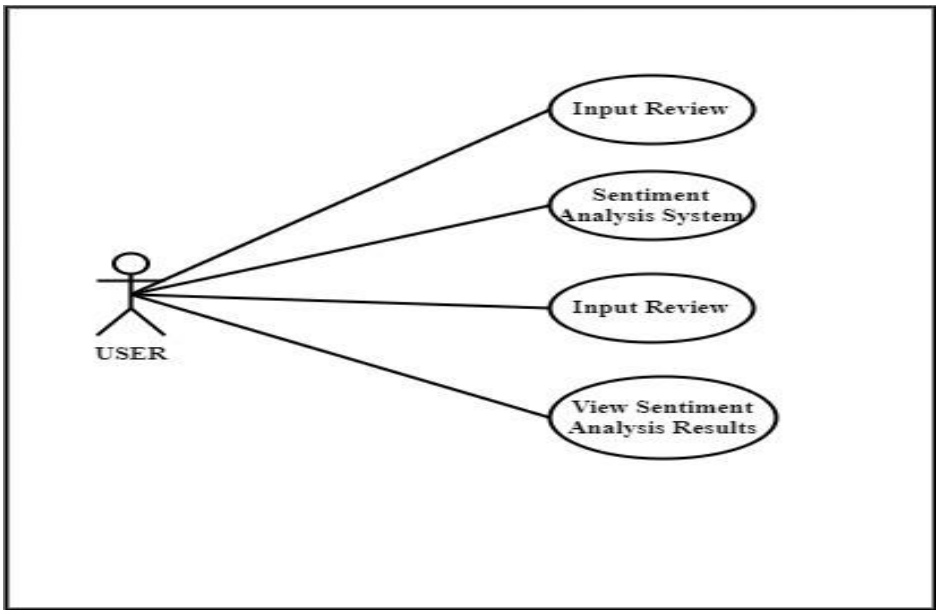


Fig 4.1: USE CASE-1

USE CASE-2 DIAGRAM:

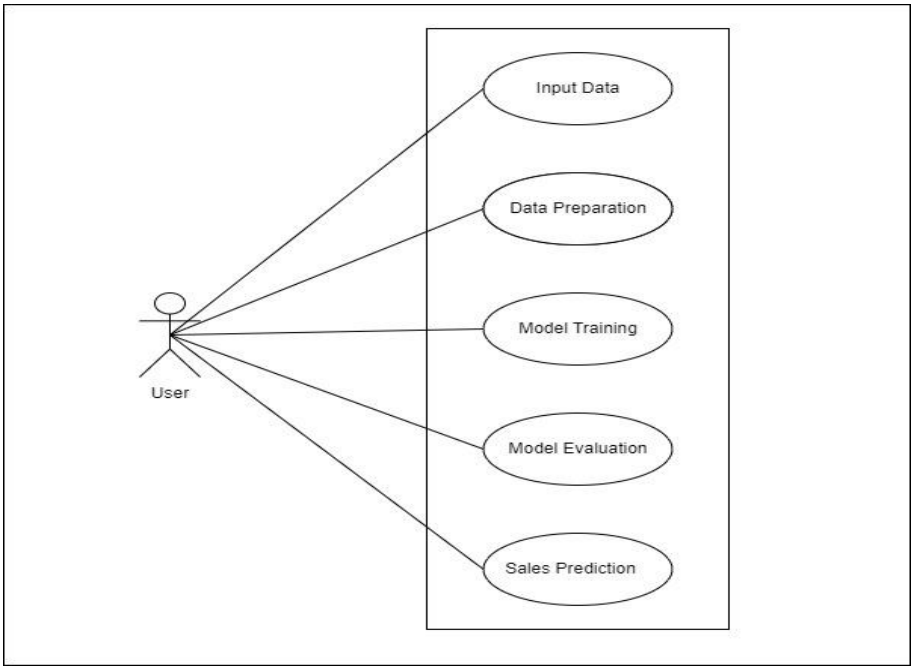


Fig 4.2: USE CASE-2

RESUME

NITHESH SHETTIGAR
S/O Giridhara Shettigar
Kc road kethigudde house, Punjalakatte post, Malady village Belthangady T Q, Dakshina Kannada,
Karnataka-574233
Email Id: nitheshshettigar8@gmail.com
+91 96117 63406

CAREER OBJECTIVE

Enthusiastic and dedicated AI & ML Diploma student seeking opportunities to apply academic knowledge and skills in a real-world environment. Eager to contribute to innovative projects and learn from experienced professionals in the field.

ACADEMIC QUALIFICATION

Course	Institution	Board/ University	Year of Passing	Marks
SSLC	GOVERNMENT JUNIOR COLLEGE PUNJALKATTE	KARNATAKA SECONDARY ENDUCATION EXAMINATION BOARD	2019	64.48%
PUC	GOVERNMENT PU COLLEGE PUNJALKATTE	DEPARTMENT OF PRE-UNIVERSITY EDUCTAION	2021	61.81 %
DIPLOMA in COMPUTER SCIENCE	GOVERNMENT POLYTECNIC BANTWAL	DEPARTMENT OF TECHNICAL EDUCATION	2024	75.80(up to 5th sem)

TECHNICAL SKILLS

Java, python, MS Word, Excel, Database Management, Power point, Basic computer.

PROJECT

Mini-Project – Gold price Prediction

CERTIFICATES

Introduction to Artificial Intelligence from Infosys Spring Board
Data visualization using Python from Infosys Spring Board
Explore Machine Learning using Python from Infosys Spring Board

PERSONAL PROFILE

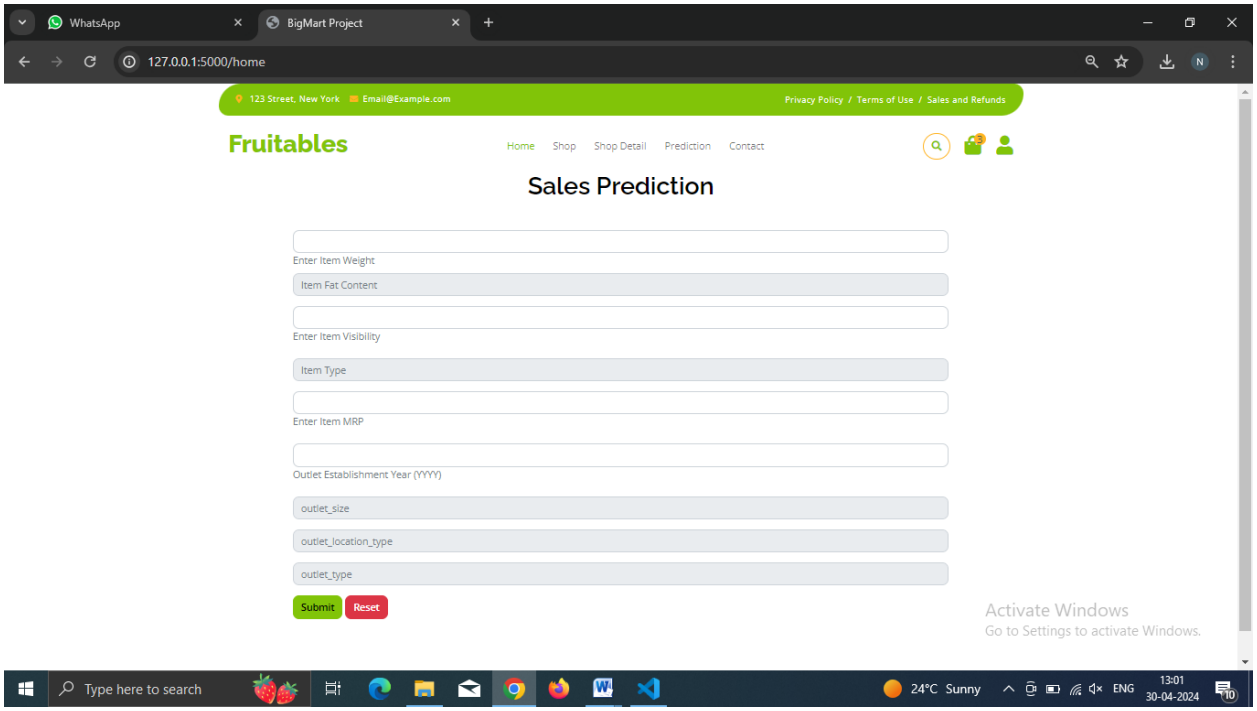
Father's Name : Giridhara Shettigar
Date of Birth : 10.01.2004
Gender : Male
Linguistic Proficiency : Kannada, English, Hindi, Tulu
Nationality : INDIAN
Marital Status : Single
Religion : Hindu
Strength : Good Interpersonal Communication Skills, Sincere and hard working

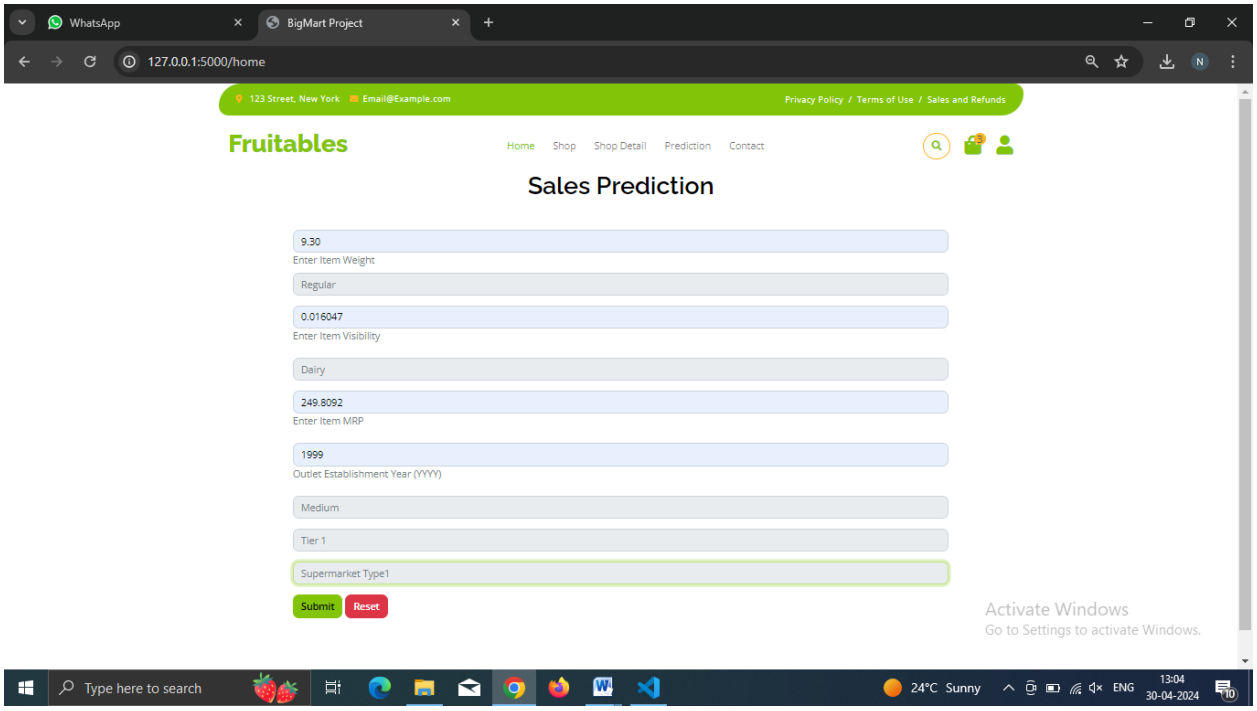
DECLARATION

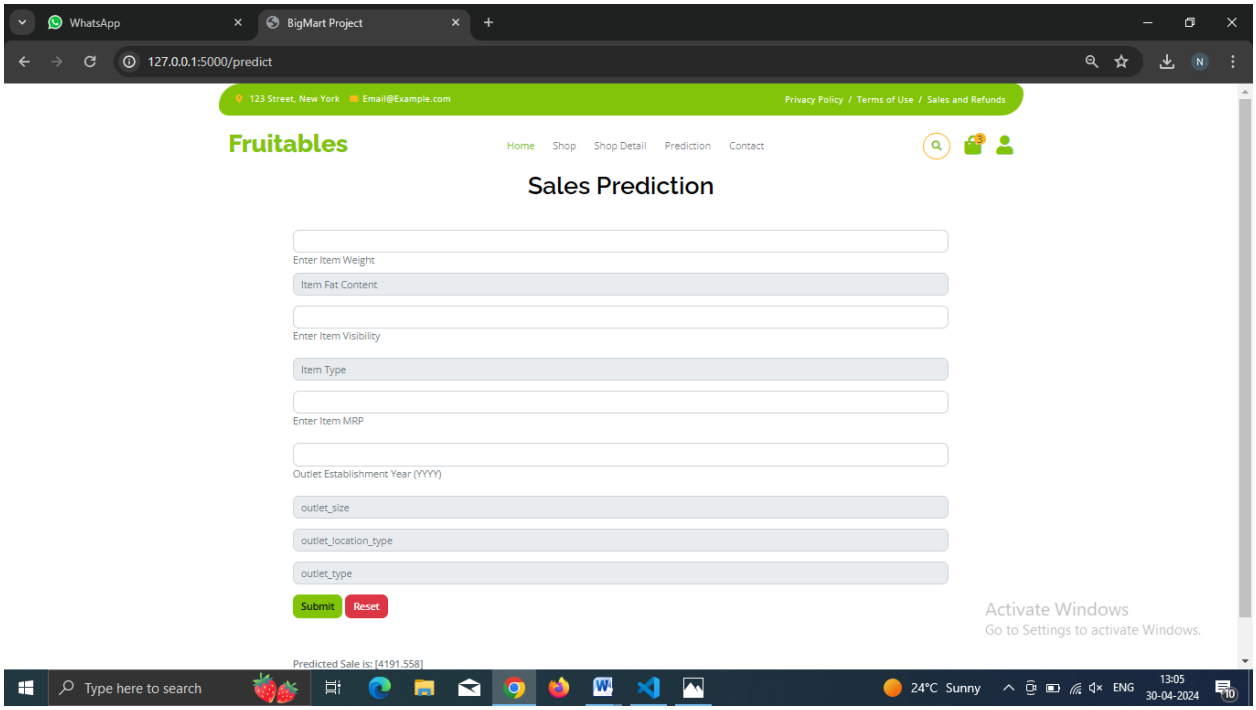
I hereby declare that above given particulars are true to the best of my knowledge.
PLACE : BANTWAL (Nithesh shettigar)
DATE :

PHOTO GALLERY

SNAPSHOT OJT-2






WhatsApp
BigMart Project
127.0.0.1:5000/predict
123 Street, New York
Email@Example.com
Privacy Policy / Terms of Use / Sales and Refunds
Fruitables
Home
Shop
Shop Detail
Prediction
Contact
Sales Prediction
Enter Item Weight
Item Fat Content
Enter Item Visibility
Item Type
Enter Item MRP
Outlet Establishment Year (YYYY)
outlet_size
outlet_location_type
outlet_type
Submit
Reset
Predicted Sale is: [4191.558]
Activate Windows
Go to Settings to activate Windows.
Type here to search
24°C Sunny
ENG
13:05
30-04-2024

FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT

FUTURE SCOPE:

- Explore more feature engineering techniques to create new meaningful features from the existing ones. For example, you could create new features by combining existing ones or by extracting more information from the available data.
- Perform hyperparameter tuning for your models to optimize their performance further. Techniques like grid search or randomized search can be employed to find the best hyperparameters for your models.
- Experiment with ensemble methods such as stacking or blending. These techniques combine predictions from multiple models to improve overall performance.
- Consider using more sophisticated models such as gradient boosting machines (GBM), support vector machines (SVM), or neural networks. These models might capture complex relationships in the data that simpler models like linear regression or decision trees cannot.
- Implement cross-validation techniques to assess the generalization performance of your models more accurately. This will provide a better estimate of how well your models will perform on unseen data.
- Explore techniques for interpreting your models' predictions. Understanding the factors that influence sales predictions can provide valuable insights for decision-making.
- If you haven't already, consider deploying your model into a production environment. This could involve creating APIs for model inference or integrating the model into a larger software system.

FURTHER ENHANCEMENT OF THE PROJECT

- If your dataset contains a timestamp or date column, you can perform time series analysis to capture seasonality, trends, and other temporal patterns in the sales data. Techniques like ARIMA, Prophet, or LSTM networks can be applied for time series forecasting.
- Explore customer segmentation techniques to group customers based on their purchasing behavior, demographics, or other characteristics. This can help tailor marketing strategies and promotions to specific customer segments, ultimately boosting sales.
- Conduct market basket analysis to identify frequently co-occurring items in transactions. This can uncover patterns like which items are often purchased together, leading to better product placement, cross-selling opportunities, and promotional strategies.

- If your dataset includes customer reviews or feedback, perform sentiment analysis to gauge customer sentiment towards products or services. Understanding customer sentiment can guide product development efforts and help improve customer satisfaction and loyalty.
- Implement dynamic pricing strategies based on factors like demand, competitor pricing, and inventory levels. Machine learning models can help optimize pricing decisions in real-time to maximize revenue and profit margins.
- Incorporate geospatial data to analyze sales patterns across different regions or locations. This can provide insights into regional preferences, market saturation, and potential areas for expansion or targeted marketing efforts.
- Create an interactive dashboard using tools like Tableau or Power BI to visualize sales performance metrics, trends, and insights. This enables stakeholders to explore the data dynamically and make data-driven decisions more effectively.
- Conduct A/B testing to evaluate the effectiveness of different marketing campaigns, pricing strategies, or product variations. This experimental approach can help identify the most impactful interventions for driving sales.
- Predict the lifetime value of customers using predictive modeling techniques. Understanding the CLV of customers allows businesses to allocate resources more efficiently towards customer acquisition, retention, and loyalty programs.
- Establish a feedback loop to continuously monitor model performance and gather feedback from stakeholders and end-users. This iterative process ensures that the model remains relevant and effective in addressing evolving business needs.

REFERENCE:

[1] https://ieeexplore.ieee.org/document/9432109
[2] https://ieeexplore.ieee.org/document/9985752
[3] https://ieeexplore.ieee.org/document/10059929