GROCERY SALES FORECASTING

A Project Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

YEPURI GOWTHAMI	**: 209T1A0588**
IJJIGANI SIVA KISHORE KUMAR	**: 209T1A0536**
KONDRAJU VEERA KRISHNAM RAJU	**: 209T1A0547**
NALAMALA DEEPTHI REDDY	**: 209T1A0561**
BODDU NARESH	**: 209T1A0509**

Under the Esteemed Guidance of

MS. CH. SUNEETHA, M. Tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VIKAS GROUP OF INSTITUTIONS

Nunna, Vijayawada-521212, Andhra Pradesh 2020-2024ISO 9001:2015 Certified

(Affiliated to JNTU Kakinada, Approved by AICTE)

2020-2024

VIKAS GROUP OF INSTUTIONS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINERING

CERTIFICATE

This is to certify that of the IV B. Tech II Sem (CSE) has satisfactorily completed the dissertation work for Major project entitled "GROCERY SALES FORECASTING" being submitted by

YEPURI GOWTHAMI	**: 209T1A0588**
IJJIGANI SIVA KISHORE KUMAR	**: 209T1A0536**
KONDRAJU VEERA KRISHNAM RAJU	**: 209T1A0547**
NALAMALA DEEPTHI REDDY	**: 209T1A0561**
BODDU NARESH	**: 209T1A0509**

In partial fulfilment for the award of the Degree of bachelor of technology in computer science and engineering to the Jawaharlal Nehru Technological University, Kakinada is a record if bonafied work carried out under my guidance and supervision.

Internal Guide	Head of the Department
CH. SUNEETHA, M. Tech	**B. SURESH, M. Tech (Ph D)**
Assistant professor	Associate professor

External Examiner

ACKNOWLEDGEMENT

We thank my chairperson Sri N. NARSI REDDY for providing the necessary infrastructure required for my project.

We thankful to Secretary Sri N. SATYANARAYANA REDDY for providing us excellent facilities in the college without which I would not have succeeded.

We thankful to our principal Dr.P.S. SRINIVASU ~~RAO~~ for fostering an excellent environment in our college and helping us to all points for achieving our task.

We grateful to B. SURESH Head of the Department of Computer Science and Engineering for his valuable guidance, which helped me to bring how this project successfully. His wise approach made me to learn the minute details of the subject. His matured and patient guidance paved a way for completing my project with sense of satisfaction and pleasure.

We very much thankful to Ms. CH. SUNEETHA for his valuable guidance, which helped me to bring out this project successfully.

Finally, we thank all the faculty of COMPUTER SCIENCE AND ENGINEERING DEPARTMENT and Library of VIKAS GROUP OF INSTUTIONS for imparting knowledge to me throughout my course.

YEPURI GOWTHAMI	**: 209T1A0588**
IJJIGANI SIVA KISHORE KUMAR	**: 209T1A0536**
KONDRAJU VEERA KRISHNAM RAJU	**: 209T1A0547**
NALAMALA DEEPTHI REDDY	**: 209T1A0561**
BODDU NARESH	**: 209T1A0509**

DECLARATION

We hereby declare that the dissertation entitled “GROCERY SALES FORECASTING” submitted for the bachelor of technology in computer science and engineering in our original work. The dissertation and results embodied in this project report has not been submitted to any other University or Institute for the award of any Degree, Associate ship or any other similar titles.

YEPURI GOWTHAMI	**: 209T1A0588**
IJJIGANI SIVA KISHORE KUMAR	**: 209T1A0536**
KONDRAJU VEERA KRISHNAM RAJU	**: 209T1A0547**
NALAMALA DEEPTHI REDDY	**: 209T1A0561**
BODDU NARESH	**: 209T1A0509**

Place: Nunna

Date:

INDEX

ABSTRACT

Predicting grocery sales is crucial for optimizing inventory management, ensuring product availability, and enhancing overall business profitability. This study proposes a data-driven approach to grocery sales prediction, leveraging advanced machine learning techniques. The aim is to develop accurate and reliable models that can forecast future sales based on historical data and relevant features. The aim of this project is to forecast more accurate product sales for the Ecuadorian supermarket chain based on certain feat.

LIST OF FIGURES

List Of Tables

CHAPTER-1

INTRODUCTION

1.1 ABOUT PROJECT:

In today's highly competitive retail landscape, accurate sales forecasting is paramount for the success of any grocery store. It enables efficient inventory management, resource allocation, and strategic decision-making. With the advent of big data analytics and machine learning techniques, retailers now have powerful tools at their disposal to predict future sales with greater precision than ever before.

This project aims to develop a robust sales forecasting model specifically tailored to the grocery industry. By leveraging historical sales data, alongside external factors such as seasonality, promotions, and economic indicators, we seek to build a predictive model that can anticipate future sales volumes accurately.

The goal of this project is to develop a robust sales forecasting system tailored specifically for grocery retailers. Leveraging advanced data analytics techniques, including machine learning algorithms and statistical models, we aim to provide accurate predictions of future sales volumes based on historical sales data and relevant external factors.

1.2 USE CASES:

- **Optimizing Inventory Management**: By accurately predicting future sales, grocery stores can optimize their inventory levels. This ensures that they have sufficient stock of high-demand items while minimizing excess inventory of slow-moving products, thereby reducing storage costs and minimizing wastage.

- **Promotional Planning:** Understanding sales patterns can help in planning promotions more effectively. For instance, if the model predicts a surge in sales of a particular product during a certain time period, the store can plan promotions or discounts to capitalize on this trend and drive additional sales.

- **Staff Scheduling**: Sales forecasts can also inform staff scheduling decisions. During peak hours or days with expected high sales, the store can schedule more staff to handle increased customer demand efficiently, leading to improved customer satisfaction and reduced wait times.

- **Supply Chain Management**: Accurate sales forecasts enable better coordination with suppliers and distributors. By sharing sales predictions with suppliers, stores can ensure timely replenishment of stock, reducing instances of stockouts and ensuring consistent product availability for customers.

- **Space Planning and Layout Optimization**: Sales forecasts can guide decisions regarding store layout and product placement. Products that are predicted to have higher sales can be given more prominent placement within the store, potentially leading to increased visibility and sales.

- **Seasonal Demand Forecasting**: Understanding seasonal variations in sales can help stores anticipate changes in demand for certain products. This enables them to adjust their inventory levels and marketing strategies accordingly, ensuring they can meet customer demand during peak seasons.

- **New Product Introductions:** Sales forecasting can assist in the successful launch of new products. By analyzing historical sales data and market trends, stores can predict the potential demand for new products, allowing them to make informed decisions regarding inventory levels and promotional strategies.

- **Budgeting and Financial Planning**: Accurate sales forecasts are essential for budgeting and financial planning purposes. Stores can use these forecasts to estimate future revenue and allocate resources effectively, ensuring they have the necessary funds to support their operations and growth initiatives.

- **Risk Management:** Identifying potential risks, such as unexpected fluctuations in sales or supply chain disruptions, is critical for grocery stores. Sales forecasting models can help in identifying and mitigating these risks by providing insights into potential scenarios and enabling proactive decision-making.

- **Customer Relationship Management**: Understanding sales patterns can also provide insights into customer behavior and preferences. Stores can use this information to tailor their marketing efforts, personalize promotions, and enhance the overall shopping experience for customers, ultimately fostering customer loyalty and retention.

1.3 EXISTING SYSTEM:

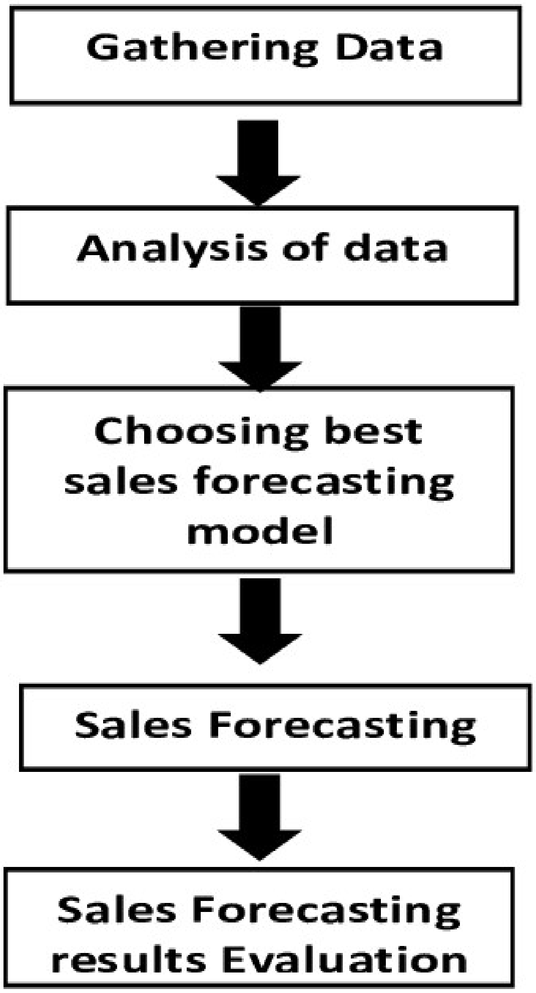

Gathering Data
Analysis of data
Choosing best sales forecasting model
Sales Forecasting
Sales Forecasting results Evaluation

Fig 1.1 Existing System

In traditional grocery sales prediction systems, forecasting is often conducted using simplistic methods such as moving averages or linear regression, which may not capture the complexity of sales patterns accurately. These approaches typically rely on historical sales data without considering other influential factors such as promotions, holidays, or external factors like weather conditions.

Data collection in traditional systems may be limited to basic sales figures without incorporating additional relevant features that could significantly impact sales patterns. Feature engineering in traditional systems is often rudimentary, if existent at all, lacking the sophistication needed to extract meaningful insights from the data.

Machine learning models used in traditional systems may be limited to basic algorithms like linear regression or simple time-series methods, which may not adequately capture the non-linear and dynamic nature of grocery sales data.

Validation and testing in traditional systems may rely on simple metrics or may not be comprehensive enough to assess the true performance of the models.Ensemble methods and continuous monitoring are often not integrated into traditional systems, limiting their ability to adapt to changing sales patterns and environmental factors.

Overall, the existing system for grocery sales prediction may lack the sophistication and accuracy required for effective inventory management and business optimization in today's competitive retail landscape. There is a clear need for a more data-driven approach leveraging advanced machine learning techniques to improve prediction accuracy and enhance business profitability.

1.4 DRAWBACKS OF EXISTING SYSTEM:

- **Limited Accuracy:** Traditional methods like moving averages or simple linear regression often lack the ability to capture the complex patterns inherent in grocery sales data. As a result, the accuracy of predictions made by these methods may be limited, leading to suboptimal inventory management decisions.

- **Inadequate Data Utilization:** The existing system may primarily rely on historical sales data without incorporating other relevant factors such as promotional events, holidays, or external factors like weather conditions. This limited scope of data utilization can result in incomplete or inaccurate predictions.

- **Lack of Advanced Techniques**: The traditional system may not leverage advanced machine learning techniques capable of capturing non-linear relationships and dynamic patterns in sales data. Without these techniques, the models may fail to provide accurate forecasts, especially in scenarios with complex sales dynamics

- **Poor Feature Engineering:** Feature engineering, if performed at all, may be rudimentary in the existing system. This can result in the failure to extract meaningful insights from the data, leading to less informative models and inferior predictions.

- **Suboptimal Model Selection and Tuning:** The choice of machine learning models and the process of hyperparameter tuning may not be systematically performed in the existing system. This can lead to suboptimal model performance and reduced predictive accuracy.

- **Limited Validation and Testing**: The validation and testing procedures in the existing system may be insufficient to assess the true performance of the models. Lack of comprehensive evaluation metrics and testing on separate datasets may result in overfitting and poor generalization to unseen data.

- **Absence of Ensemble Methods and Continuous Monitoring**: Ensemble methods, which can combine predictions from multiple models to improve accuracy, and continuous monitoring of model performance may not be integrated into the existing system. As a result, the system may lack adaptability to changing sales patterns and environmental factors over time.

1.5 PROPOSED SYSTEM:

The proposed system for grocery sales prediction aims to address the limitations of the existing system by leveraging advanced machine learning techniques and comprehensive data analysis. Here are the key components of the proposed system:

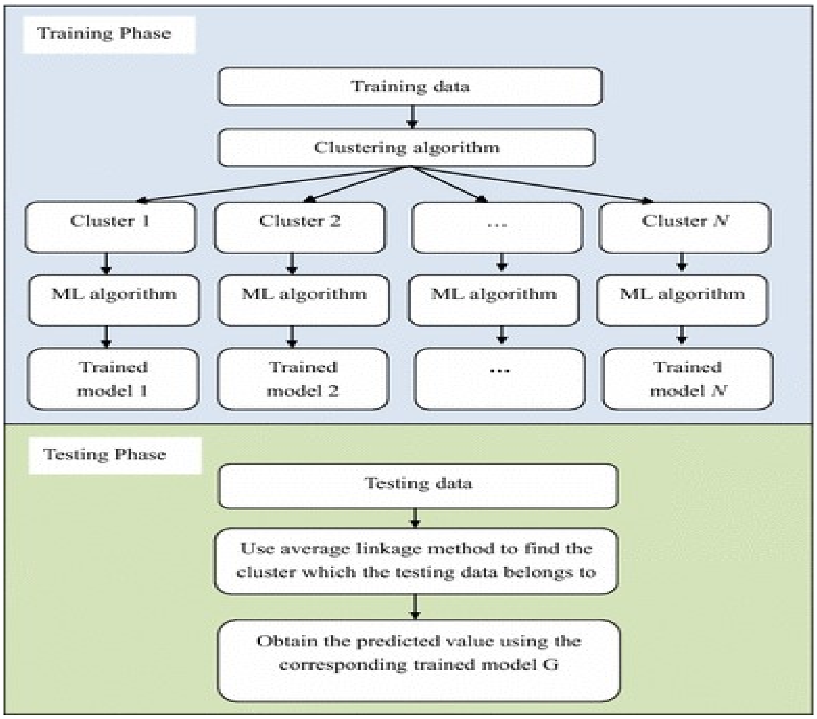


Fig 1.2 Proposed System

- **Comprehensive Data Collection**: The proposed system will collect a wide range of data, including historical sales figures, promotional events, holidays, weather data, economic indicators, and any other relevant factors that may influence grocery sales. This extensive data collection ensures that the models have access to all pertinent information for accurate predictions.

- **Advanced Feature Engineering**: Sophisticated feature engineering techniques will be applied to preprocess the collected data and extract meaningful features. This may involve handling missing values, encoding categorical variables, creating lag features to capture temporal dependencies, and incorporating domain knowledge to engineer new features that can improve prediction accuracy.

- **Machine Learning Model Selection**: Various advanced machine learning algorithms suitable for time-series forecasting will be evaluated for their effectiveness in predicting grocery sales. This may include models such as ARIMA, SARIMA, LSTM, XGBoost, or hybrid approaches that combine multiple models for improved performance.

- **Hyperparameter Tuning and Model Optimization**: Systematic hyperparameter tuning and model optimization techniques will be employed to fine-tune the selected machine learning models. This process ensures that the models are optimized for performance and can effectively capture the underlying patterns in the data.

- **Comprehensive Validation and Testing**: The performance of the developed models will be rigorously evaluated using appropriate evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or Mean Absolute Percentage Error (MAPE). The models will be validated on separate training, validation, and testing datasets to ensure robustness and generalization to unseen data.

- **Ensemble Methods Integration**: Ensemble methods such as stacking, boosting, or bagging will be explored to further improve prediction accuracy. By combining predictions from multiple models, ensemble methods can effectively capture diverse patterns in the data and enhance overall forecasting performance.

- **Continuous Monitoring and Model Updating**: The proposed system will include mechanisms for continuous monitoring of model performance in real-time production environments.

1.6 ADVANTAGES OF PROPOSED SYSTEM:

- **Improved Prediction Accuracy**: By leveraging advanced machine learning techniques and comprehensive data analysis, the proposed system can provide more accurate forecasts of grocery sales. Advanced models and sophisticated feature engineering allow for better capturing of complex patterns and dynamics in the data, leading to more precise predictions.

- **Better Utilization of Data**: The proposed system collects and utilizes a wide range of data sources, including historical sales figures, promotional events, holidays, weather data, and economic indicators. By incorporating all relevant factors that influence sales, the system can provide more comprehensive and informed predictions.

- **Enhanced Adaptability**: Advanced machine learning models and ensemble methods used in the proposed system offer greater adaptability to changing sales patterns and environmental factors. Continuous monitoring and regular model updates ensure that the predictions remain accurate and reliable over time, even as conditions evolve.

- **Optimized Inventory Management**: Accurate sales predictions provided by the proposed system enable businesses to optimize their inventory management processes. By anticipating demand more effectively, businesses can reduce overstocking or stockouts, minimize inventory holding costs, and improve overall operational efficiency.

- **Improved Decision Making**: The accurate forecasts generated by the proposed system empower businesses to make data-driven decisions regarding pricing, promotions, and resource allocation. By having better insights into future sales trends, businesses can allocate resources more efficiently and maximize profitability.

- **Competitive Advantage**: By adopting advanced machine learning techniques and leveraging comprehensive data analysis, businesses can gain a competitive advantage in the retail industry. The ability to accurately forecast sales and adapt to changing market conditions can differentiate businesses from competitors and drive growth and success.

- **Cost Savings**: Optimized inventory management and improved decision-making enabled by the proposed system can result in cost savings for businesses. By implementing cost-saving strategies, you ensure that resources are allocated efficiently. This means investing in tools, technologies, and personnel where they are most needed and where they can have the greatest impact on improving sales forecasting accuracy.

CHAPTER-2
SYSTEM ANALYSIS

2.1 SOFTWARE REQUIREMENTS:

One of the most difficult tasks in that, the selection of software. Once system requirements known i.e., determining whether a particular software package fits the requirements.

- Operating System : Windows Family.
- Version : Python 3.9.1
- Programming Language : Python.
- Development IDE : Jupyter lab

2.2 HARDWARE REQUIREMENTS:

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

- Processor : intel i5
- Speed : 1.1 GHz (min)
- RAM : 16GB RAM
- Hard Disk : 512

2.3 PROJECT LIFE CYCLE:

Forecasting grocery sales involves predicting future sales volumes based on historical data, external factors like seasonality, promotions, and other variables that influence consumer behaviour. Here's a generalized project lifecycle for grocery sales forecasting:

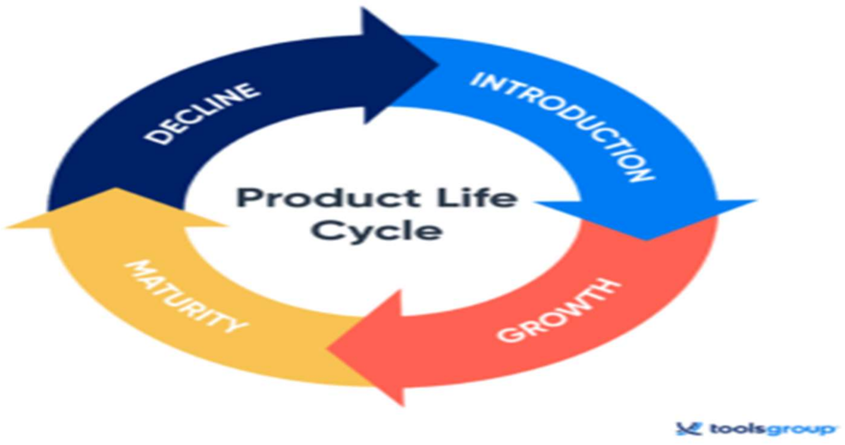


Fig 2.1 Project Life Cycle

Here's a generalized project lifecycle for grocery sales forecasting:

1. **Understanding Business Objectives:**
   - Define the specific objectives of the grocery sales forecasting project. This could include optimizing inventory management, improving supply chain efficiency, or maximizing sales revenue.

2. **Data Collection:**
   - Gather historical sales data from various sources such as point-of-sale systems, transaction records, and customer databases.
   - Collect external data such as weather patterns, holidays, and economic indicators that may impact sales.

3. **Data Preprocessing:**
    - Clean the collected data by handling missing values, outliers, and inconsistencies.
    - Aggregate the data to the desired level of granularity (e.g., daily, weekly, monthly).
    - Create additional features such as seasonality indicators, holiday flags, and promotional event variables.

4. **Exploratory Data Analysis (EDA):**
    - Analyse the historical sales data to identify trends, patterns, and seasonality.
    - Visualize the data using charts and graphs to gain insights into sales behavior over time.
    - Conduct correlation analysis to understand the relationships between sales and other variables.

5. **Model Selection:**
    - Choose appropriate forecasting models based on the characteristics of the data and the business requirements. Common models include time series models (e.g., ARIMA, SARIMA), machine learning algorithms (e.g., regression, neural networks), and ensemble methods.
    - Consider the trade-offs between model complexity, interpretability, and accuracy.

6. **Model Training:**
    - Split the historical data into training and validation sets.
    - Train the selected forecasting models using the training data.
    - Tune model hyperparameters to optimize performance using techniques like grid search or random search.

7. **Model Evaluation:**
    - Evaluate the performance of the trained models using the validation set.
    - Measure forecast accuracy using metrics such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), or Root Mean Squared Error (RMSE).
    - Compare the performance of different models and select the best-performing one.

8. **Forecasting:**
    - Use the trained model to generate forecasts for future sales volumes.

- Incorporate external factors and events into the forecasting process to improve accuracy.
- Generate point estimates as well as prediction intervals to quantify uncertainty.

9. Model Deployment:

- Deploy the trained forecasting model into production or the operational environment.
- Integrate the model with existing systems or tools for generating and disseminating forecasts.
- Set up automated pipelines for updating forecasts regularly as new data becomes available.

10. Monitoring and Maintenance:

- Monitor forecast accuracy and performance metrics over time.
- Continuously retrain the model using updated data to adapt to changing patterns and trends.
- Identify and address any issues or discrepancies between forecasted and actual sales.

11. Documentation and Reporting:

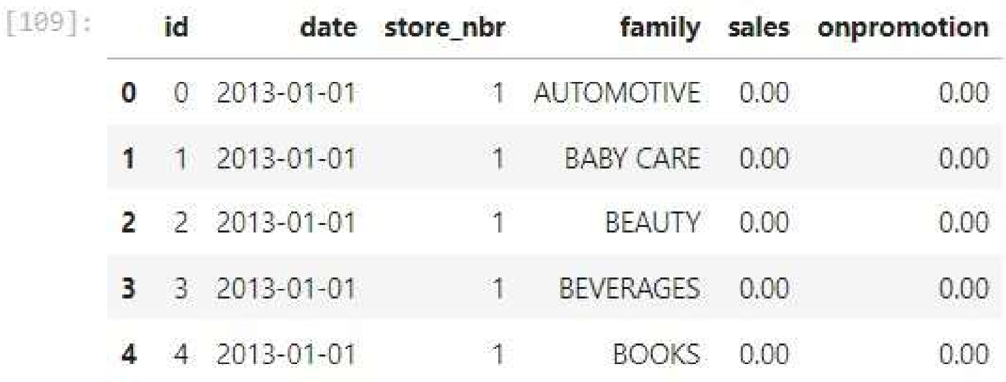
- Document the entire forecasting process, including data sources, preprocessing steps, model selection criteria, and evaluation results.
- Prepare reports or dashboards to communicate forecasted sales volumes and insights to stakeholders.
- Provide explanations for forecast variances and recommendations for improving future forecasts.

12. Feedback Loop:

- Collect feedback from stakeholders and end-users regarding the accuracy and usefulness of the forecasts.
- Incorporate feedback into future iterations of the forecasting model to enhance performance and address specific business needs.

2.4 DATA COLLECTION:

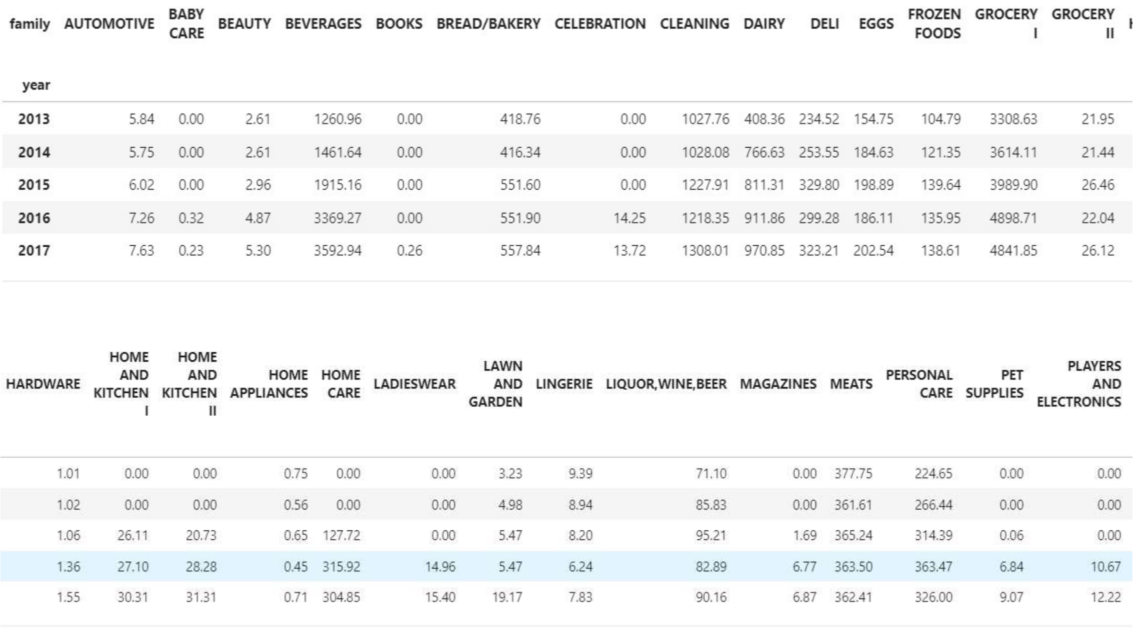
- Collecting data for a Kaggle project involves sourcing datasets from publicly available repositories or contributing your own datasets to the Kaggle platform.
- Start by exploring the Kaggle Datasets platform (https://www.kaggle.com/datasets). Use relevant keywords such as "grocery sales," "retail," or "supermarket" to search for existing datasets related to grocery sales forecasting or retail analytics.



[109]:

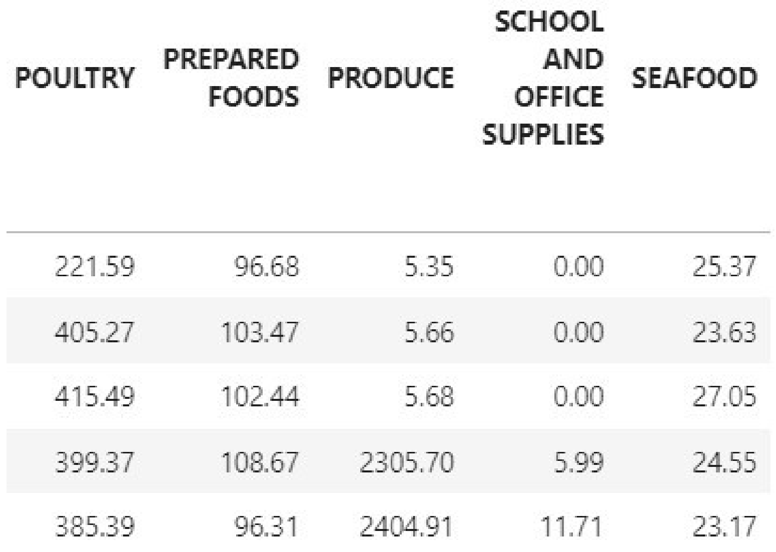
	id	date	store_nbr	family	sales	onpromotion
0	0	2013-01-01	1	AUTOMOTIVE	0.00	0.00
1	1	2013-01-01	1	BABY CARE	0.00	0.00
2	2	2013-01-01	1	BEAUTY	0.00	0.00
3	3	2013-01-01	1	BEVERAGES	0.00	0.00
4	4	2013-01-01	1	BOOKS	0.00	0.00

Table:2.1 CSV File



family	AUTOMOTIVE	BABY CARE	BEAUTY	BEVERAGES	BOOKS	BREAD/BAKERY	CELEBRATION	CLEANING	DAIRY	DELI	EGGS	FROZEN FOODS	GROCERY I	GROCERY II
year														
2013	5.84	0.00	2.61	1260.96	0.00	418.76	0.00	1027.76	408.36	234.52	154.75	104.79	3308.63	21.95
2014	5.75	0.00	2.61	1461.64	0.00	416.34	0.00	1028.08	766.63	253.55	184.63	121.35	3614.11	21.44
2015	6.02	0.00	2.96	1915.16	0.00	551.60	0.00	1227.91	811.31	329.80	198.89	139.64	3989.90	26.46
2016	7.26	0.32	4.87	3369.27	0.00	551.90	14.25	1218.35	911.86	299.28	186.11	135.95	4898.71	22.04
2017	7.63	0.23	5.30	3592.94	0.26	557.84	13.72	1308.01	970.85	323.21	202.54	138.61	4841.85	26.12

HARDWARE	HOME AND KITCHEN I	HOME AND KITCHEN II	HOME APPLIANCES	HOME CARE	LADIESWEAR	LAWN AND GARDEN	LINGERIE	LIQUOR,WINE,BEER	MAGAZINES	MEATS	PERSONAL CARE	PET SUPPLIES	PLAYERS AND ELECTRONICS
1.01	0.00	0.00	0.75	0.00	0.00	3.23	9.39	71.10	0.00	377.75	224.65	0.00	0.00
1.02	0.00	0.00	0.56	0.00	0.00	4.98	8.94	85.83	0.00	361.61	266.44	0.00	0.00
1.06	26.11	20.73	0.65	127.72	0.00	5.47	8.20	95.21	1.69	365.24	314.39	0.06	0.00
1.36	27.10	28.28	0.45	315.92	14.96	5.47	6.24	82.89	6.77	363.50	363.47	6.84	10.67
1.55	30.31	31.31	0.71	304.85	15.40	19.17	7.83	90.16	6.87	362.41	326.00	9.07	12.22



POULTRY	PREPARED FOODS	PRODUCE	SCHOOL AND OFFICE SUPPLIES	SEAFOOD
221.59	96.68	5.35	0.00	25.37
405.27	103.47	5.66	0.00	23.63
415.49	102.44	5.68	0.00	27.05
399.37	108.67	2305.70	5.99	24.55
385.39	96.31	2404.91	11.71	23.17

Table:2.2 Grocery

2.5 DATA PREPROCESSING:

Preprocessing techniques play a crucial role in preparing data for grocery sales forecasting projects.

1. **Handling Missing Values**:
   - Identify missing values in the dataset and decide on appropriate strategies for handling them. Options include imputation techniques such as mean, median, mode imputation, forward or backward filling, or using advanced methods like K-nearest neighbours (KNN) imputation or predictive modeling-based imputation.

2. **Outlier Detection and Treatment**:
   - Detect outliers in the sales data that may skew the forecasting model. Techniques such as Z-score, IQR (Interquartile Range), or visual inspection (box plots, histograms) can be used. Decide whether to remove outliers, cap them, or transform them using techniques like visualization.

3. **Time Series Decomposition**:
    - Decompose the time series data into its constituent components (trend, seasonality, and residual) using methods such as additive or multiplicative decomposition. This helps in understanding the underlying patterns and extracting features that can improve forecasting accuracy.

4. **Feature Engineering**:
    - Create additional features from existing ones that may capture relevant information impacting grocery sales. For example, derive lag features (previous sales data), rolling statistics (moving averages), day of the week, month, seasonality indicators, holiday indicators, or features representing promotions or marketing campaigns.

5. **Normalization/Scaling**:
    - Normalize or scale the features to a similar scale to ensure that all features contribute equally to the model training process. Techniques such as Min-Max scaling or Standardization (Z-score normalization) can be applied.

6. **Handling Categorical Variables**:
    - Encode categorical variables into numerical representations suitable for machine learning models. Techniques include one-hot encoding, label encoding, or target encoding, depending on the nature of the categorical variables and the chosen machine learning algorithm.

7. **Time Series Stationarity**:
    - Check for stationarity in the time series data, as many forecasting models assume stationary data. Techniques such as Dickey-Fuller test or visual inspection of rolling statistics can be used. If the data is non-stationary, apply differencing or transformations (e.g., log transformation) to make it stationary.

8. **Handling Seasonality and Trends**:
    - Account for seasonality and trends present in the sales data by applying techniques such as seasonal differencing, detrending, or using seasonal

decomposition methods like STL (Seasonal and Trend decomposition using Loess).

9. **Feature Selection**:
   - Select the most relevant features for forecasting grocery sales using techniques like correlation analysis, feature importance ranking, or dimensionality reduction methods such as PCA (Principal Component Analysis).

10. **Data Splitting**:
    - Split the dataset into training, validation, and testing sets. Ensure that the temporal order is maintained to mimic real-world scenarios, where the model is trained on past data and evaluated on future data.

2.6 MODULE DESCRIPTION:

NUMPY:

- NumPy, which stands for Numerical Python, is a powerful open-source library in Python that is used for numerical and mathematical operations.
- In a grocery sales forecasting project, the NumPy library can be used for various tasks such as data manipulation, statistical analysis, and mathematical computations.
- You can use NumPy to load data from various sources like CSV files or databases into NumPy arrays using functions like **NumPy. loadtxt ()** or **NumPy. genfromtxt ()**. These arrays can efficiently store and manipulate large datasets.
- Installation: pip install NumPy
- Access: import NumPy as np

PANDAS:

- The pandas library is extensively used in grocery sales forecasting projects due to its powerful data manipulation and analysis capabilities, particularly for tabular data.
- Pandas provides convenient functions like **pd. read_csv ()** and **pd. read_excel ()** to load data from CSV files, Excel spreadsheets, databases, or other sources into Data Frame objects. These Data Frames serve as the primary data structure for data manipulation and analysis in pandas.

- The primary two data structures in Pandas are Series and Data Frame.
- Installation: pip install pandas
- Access: import pandas as pd

MATPLOTLIB:

- Matplotlib is a powerful library in Python commonly used for creating static, interactive, and animated visualizations in data analysis and presentation.
- In a grocery sales forecasting project, Matplotlib can be utilized for various visualization tasks to help understand the data, analyze trends, and present forecasts.
- Installation: pip install matplotlib
- Access: import matplotlib as plt.

SCIKIT-LEARN:

- The purpose of scikit-learn, often abbreviated as SK learn, is to provide a simple and efficient tool for data mining and data analysis tasks, particularly in machine learning.
- It is built on top of other popular Python libraries such as NumPy, SciPy, and matplotlib. Scikit-learn provides a simple and efficient tool for data mining, data analysis, and machine learning tasks.
- Installation: pip install scikit-learn.
- Access: import scikit-learn as SK learn.

SEABORN:

- Seaborn can be utilized in a grocery sales forecasting project to visualize and analyze various aspects of the data, aiding in understanding patterns, trends, and relationships that can be valuable for forecasting.
- Seaborn can be employed to create visualizations like histograms, box plots, and violin plots to explore the distribution of sales data, identify outliers, and understand the central tendency and spread of different product sales.
- Installation: pip install seaborn.
- Access: import seaborn as sns.

PYPLOT:

- In a grocery sales forecasting project, matplotlib's pyplot module can be incredibly useful for visualizing various aspects of the data, model results, and forecasted trends.
- Line plots: Matplotlib can be used to create line plots to visualize the historical sales data over time. This allows analysts to observe trends, seasonal patterns, and potential outliers.
- Scatter plots: Scatter plots can be employed to explore the relationship between time and sales volume, potentially highlighting any underlying patterns or correlations.

CHAPTER-3

SYSTEM DESIGN

3.1 BLOCK DIAGRAM:

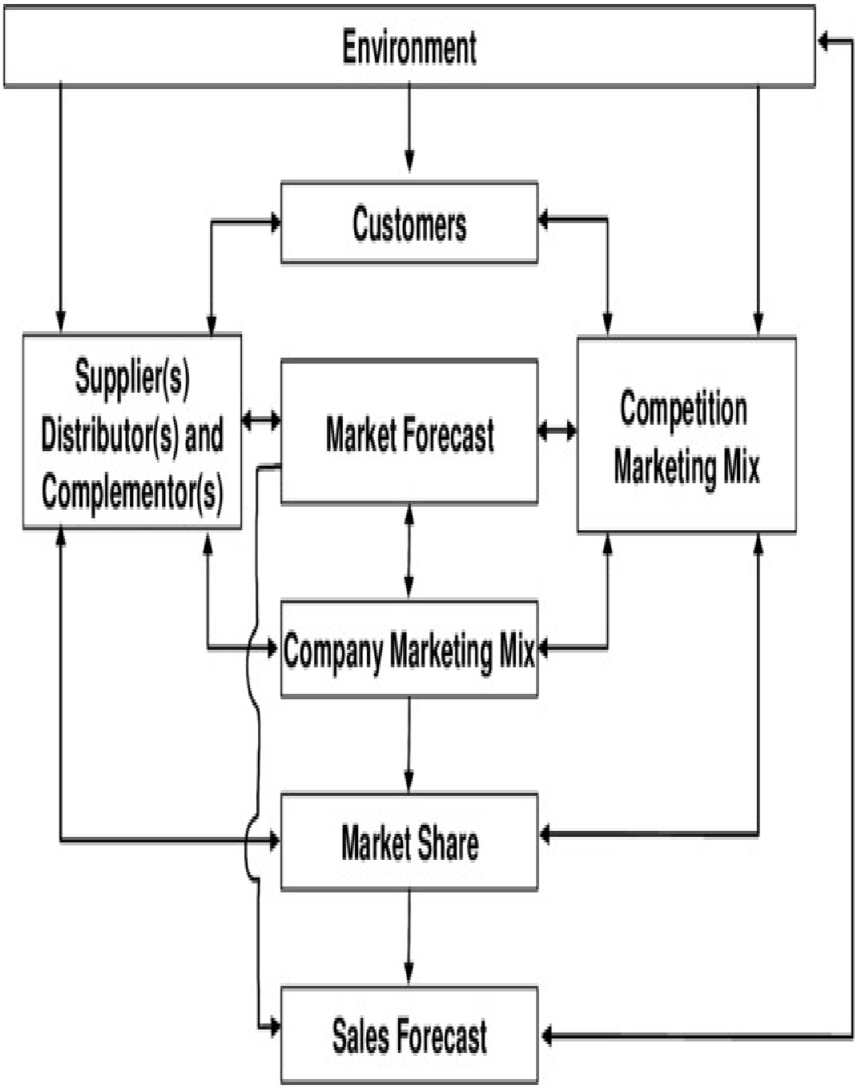


Fig 3.1 Block Diagram

3.2 UML DIAGRAM:

The use case diagram is a technique used in the development of a software or system to capture the functional requirements of a system. The use case diagram is used to construct behavioural things in a model, since the use case diagrams can explain the interactions that occur between the users and the system itself. A use case diagram can define functionality and software features from user perspective.

The design of sales forecasting application that adopt RFM concept is done by model the use case diagram.

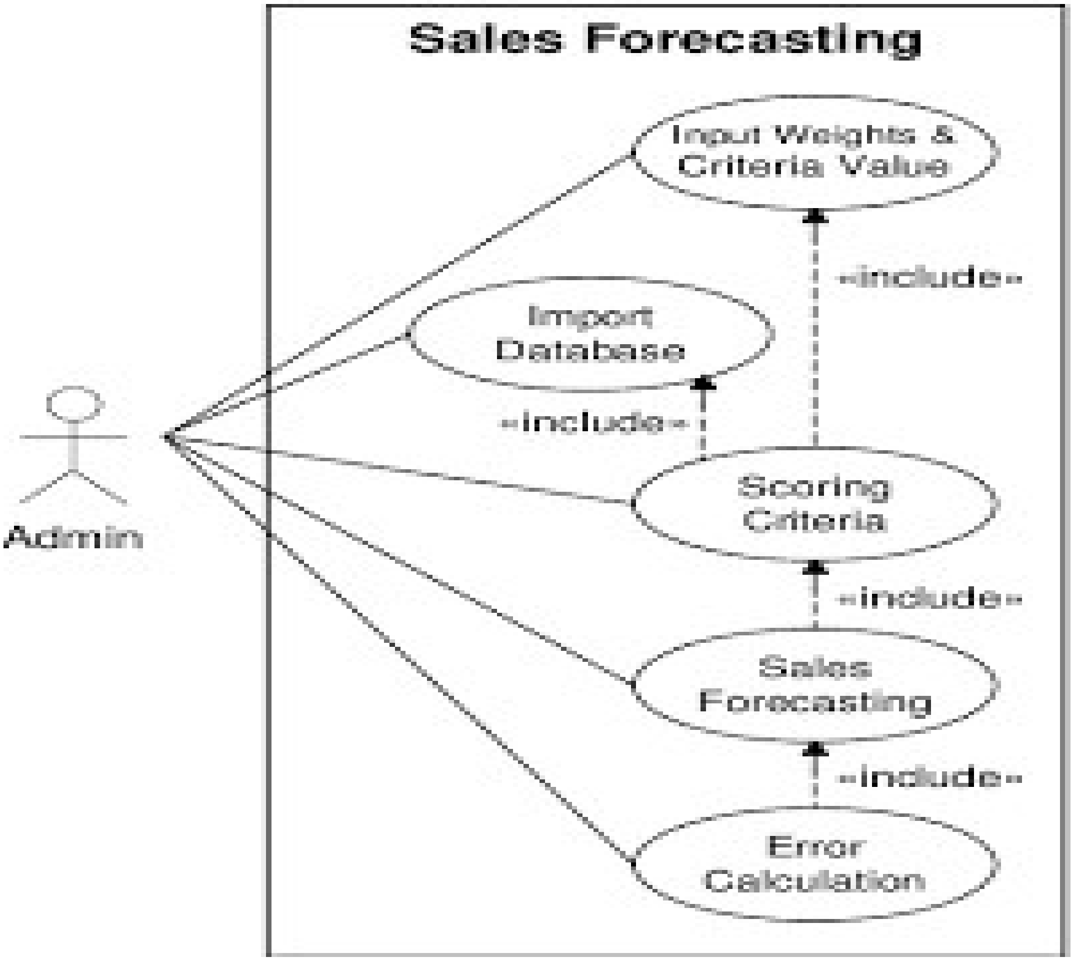


Fig 3.2 UML Diagram for Grocery Sales Forecasting

3.3 CLASS DIAGRAM:

- A “Class Diagram” shows a set of classes, interfaces and collaborations and their relationships. These diagrams are most common diagram in model object oriented systems. Class diagrams are the backbone of almost every object – oriented methods, including UML. They describe the static structure of a system . An object Class describes a group of objects with similar properties (attributes), common behaviour (operations), common relationships to the other objects, and common semantics. The Abbreviation Class is often used instead of Object Class. Project manager in a class have same attributes and behaviour patterns. Most objects derive their individuality from differences in their attribute values and relationships to other objects. Different classes identified in the system are the Data Identification, and Data Operation.
- Class diagram plays a major role inhume design. They represent Static Structure of the System Basic Class Diagram Symbols and Notations: Classes represent an abstraction of entities with the common characteristics. Associations represent the relationships between classes.

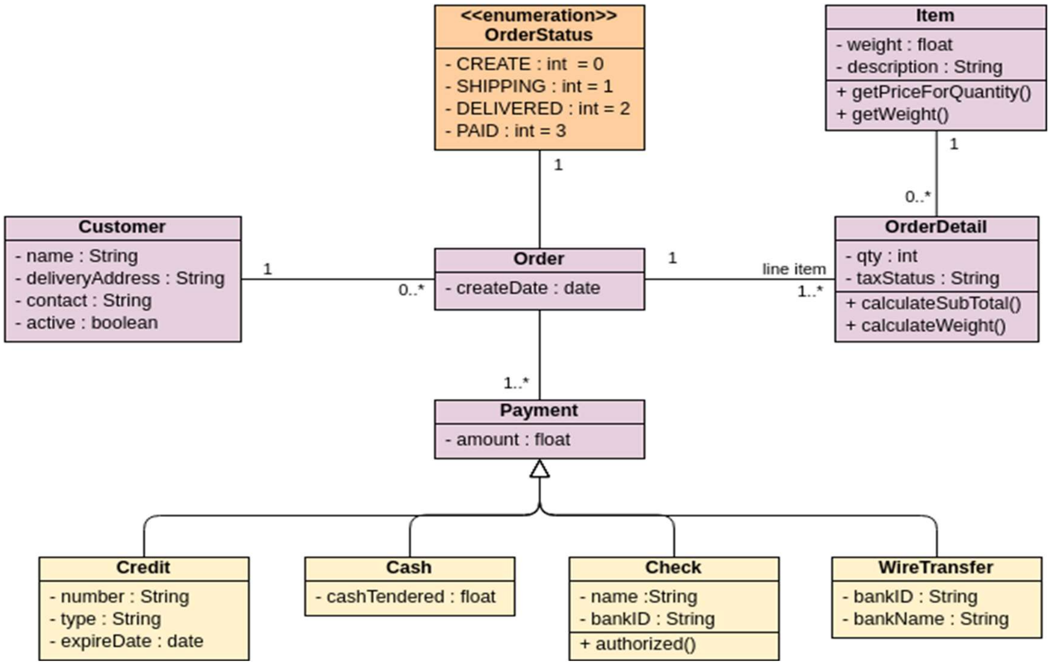


Fig 3.3 Class Diagram

In a grocery sales forecasting project, a class diagram can be a valuable tool for visualizing the static structure of the system, including the key classes, their attributes, methods, and relationships. Here's how a class diagram might be used in such a project:

Identifying Key Classes:

- The first step in creating a class diagram is identifying the main classes in the system. For a grocery sales forecasting project, this could include classes such as:
- Product: Represents individual products sold in the grocery store.
- Order: Represents customer orders placed for products.
- Inventory: Represents the inventory of products available in the store.
- SalesForecast: Represents the forecasted sales data, which could be generated by the system.
- Customer: Represents customers who place orders.
- Supplier: Represents suppliers who provide products to the store.

Defining Attributes and Methods:

- Once the main classes have been identified, you would define the attributes (properties) and methods (behaviors) associated with each class. For example:
- Product class might have attributes like name, price, quantity Available, etc., and methods like update Price() or adjust Quantity().
- Order class might have attributes like order ID, customer ID, order Date, etc., and methods like calculate Total() or generate Invoice().
- Inventory class might have attributes like product ID, quantity In Stock, reorder Threshold, etc., and methods like update Inventory() or check Availability().
- Sales Forecast class might have attributes like forecast Date, forecasted Sales Data, etc., and methods like generate Forecast().

Establishing Relationships:

- Class diagrams also show how classes are related to each other. In a grocery sales forecasting project, you might have relationships like:
- An Order is associated with one or more Products.
- Inventory keeps track of the available quantity of Products.

- Sales Forecast might be generated based on historical sales data from Orders and Products.
- Customers place Orders.
- Suppliers provide Products to the Inventory.

Navigating Multiplicity and Associations:

- Multiplicity indicates how many instances of one class are associated with instances of another class. For example, in an Order class, there might be a "1 to Many" relationship with Product, indicating that one order can contain multiple products. Similarly, in a grocery sales forecasting system, one product can have multiple orders associated with it.

Refining the Design:

- As the project evolves, the class diagram may need to be refined or updated to reflect changes in requirements or design decisions. For example, if new features are added to the system, new classes may need to be introduced, or existing classes may need to be modified.

Overall, the class diagram serves as a blueprint for developers to understand the structure of the system and how its various components interact with each other, aiding in the development and maintenance of the grocery sales forecasting project.

3.4 USE CASE DIAGRAM:

Use case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for model the dynamic aspects of systems). use case diagram are central to model the behaviour of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.

A use case diagram in the Unified Model Language (UML) is a type of behaviour diagram defined by and created from a Use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.

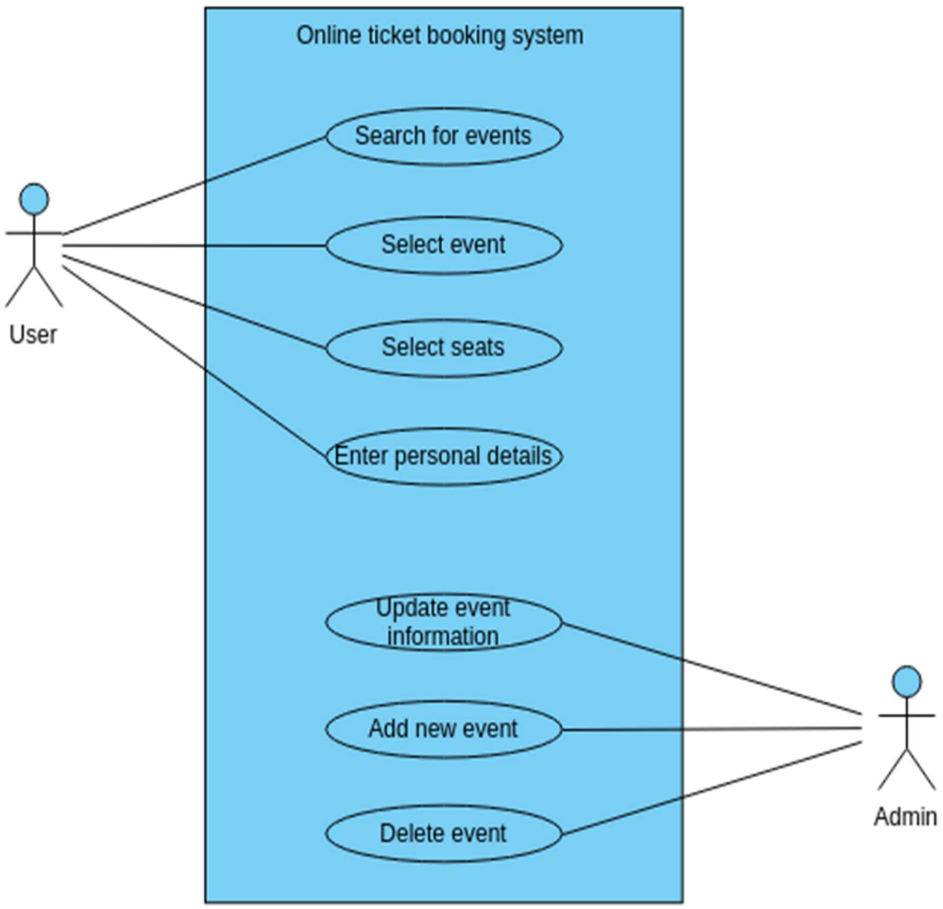


Fig 3.4 Use Case Diagram

1.Actors:

- **Admin**: Manages the system, views reports, and configures settings.
- **Data Analyst**: Analyz sales data and generates forecasts.

Fig 3.5 Representation of Actor

2. **Use Cases**:
    - **View Sales Data**: Both Admin and Data Analyst can view historical sales data.
    - **Generate Forecast**: Data Analyst generates sales forecasts based on historical data.
    - **Configure Forecast Parameters**: Admin configures forecasting parameters (e.g., time period, confidence level).
    - **View Forecast Report**: Both Admin and Data Analyst can view generated sales forecasts.

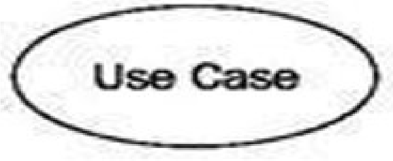


Fig 3.6 Representation of Use case

3. **Relationships**:
    - **Admin** interacts with all use cases.
    - **Data Analyst** interacts mainly with “Generate Forecast” and “View Forecast Report.”

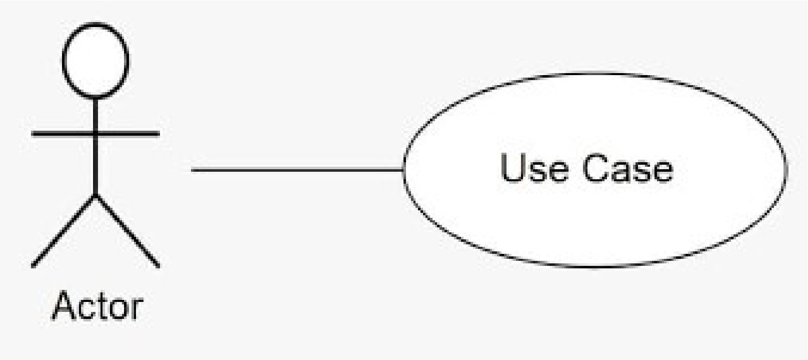


Fig 3.7 Relation to Actor and Use case

3.5 ACTIVITY DIAGRAM:

- Activity diagram describe the workflow behaviour of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed.
- Activity diagrams can show activities that are conditional or parallel. Activity diagram should be used in conjunction with other model techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for analyze a use case by describing what actions need to take place and when they should occur describing a complicated sequential algorithm and model applications with parallel processes.

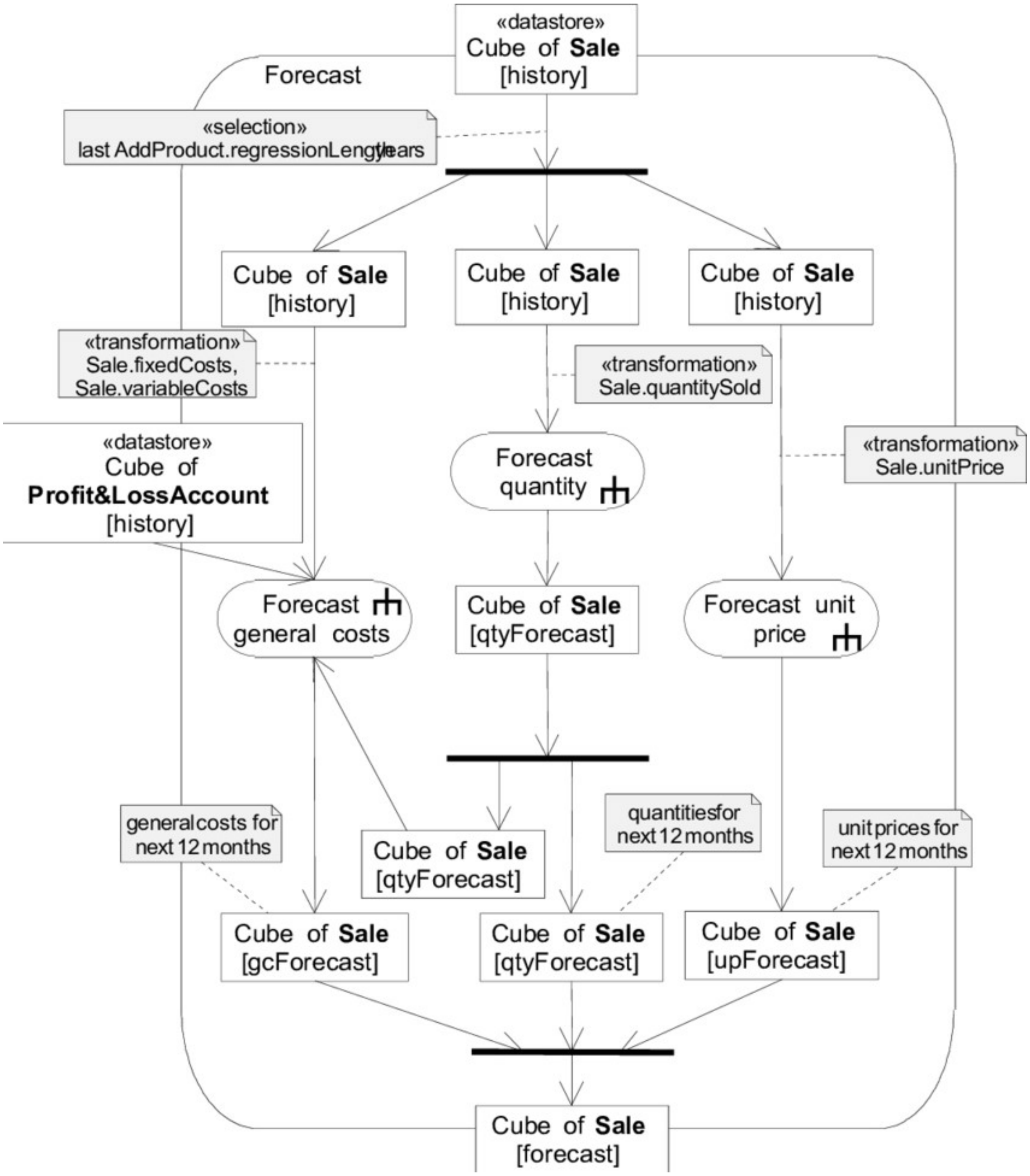


Fig : 3.8 Activity Diagram

3.6 SEQUENCE DIAGRAM:

Sequence diagram is an interaction diagram which is focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of a system. Sequence diagrams belong to a group of UML diagrams called Interaction Diagrams.

Sequence diagrams describe how objects interact over the course of time through an exchange of messages. A single sequence diagram often represents the flow of events for a single use case.

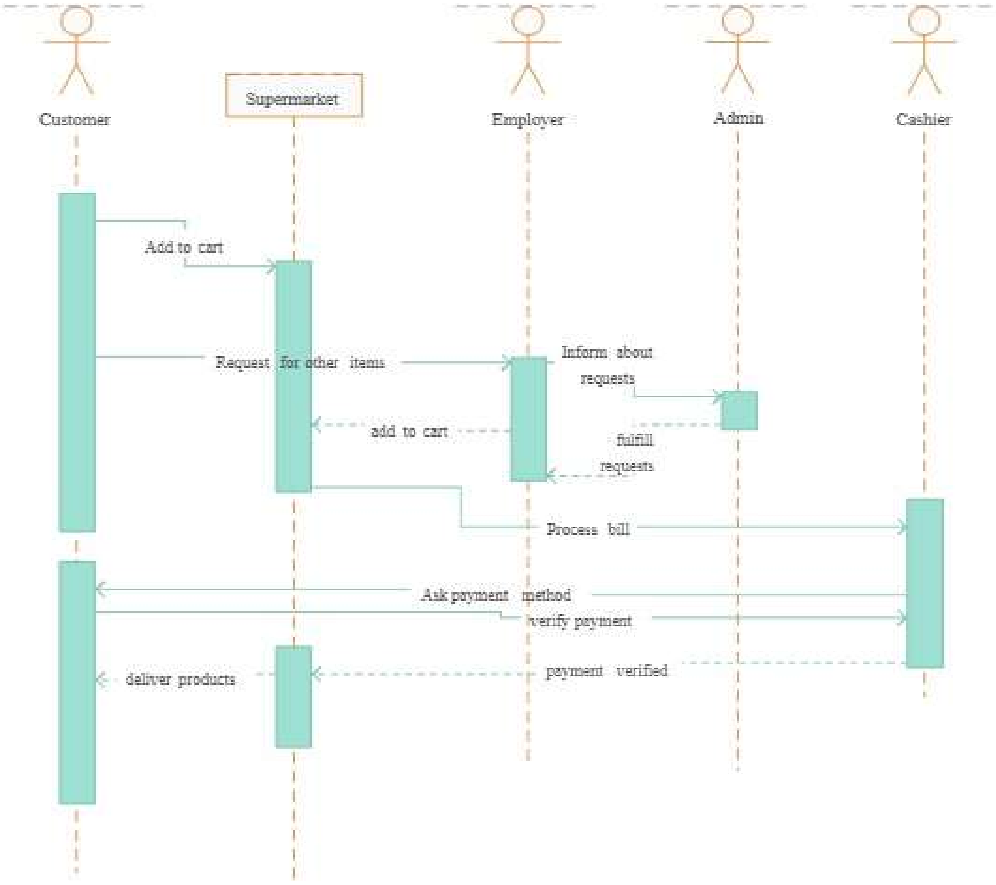


Fig: 3.9 Sequence Diagram

Sequence diagrams are valuable for understanding the interaction between different components or objects within a system over time. In a grocery sales forecasting project, sequence diagrams can be particularly useful for visualizing the flow of messages or actions between various elements of the system during processes such as order placement, inventory management, or sales forecasting. Here's how sequence diagrams can be beneficial for such a project:

Process Visualization:

- Sequence diagrams allow you to visually represent the flow of processes within the system. For example, you can illustrate the sequence of steps involved in placing an order, from the customer selecting products to checkout, payment processing, and order confirmation. This helps stakeholders, including developers and business analysts, to understand the order of operations and identify potential bottlenecks or areas for optimization.

Interaction Modeling:

- Sequence diagrams depict how objects or components interact with each other to accomplish a specific task. In a grocery sales forecasting system, this could involve interactions between components such as the user interface, inventory management system, sales forecasting algorithm, and database. By visualizing these interactions, you can ensure that the system components are communicating effectively and fulfilling their respective roles.

Error Handling:

- Sequence diagrams can also illustrate how the system handles errors or exceptions during the execution of processes. For instance, if a product is out of stock when a customer tries to place an order, the sequence diagram can show how the system detects this condition, notifies the user, and provides alternative options (e.g., backordering or suggesting similar products). Understanding error handling mechanisms is crucial for ensuring the reliability and robustness of the system.

Integration Points:

- In a complex system like a grocery sales forecasting platform, there may be multiple integration points with external systems or services, such as payment gateways, inventory management systems, or third-party APIs. Sequence diagrams help visualize the flow of data and control between the system and these external entities, facilitating the design and implementation of seamless integrations.

Performance Analysis:

- By examining the sequence of interactions between system components, you can identify potential performance bottlenecks or areas where optimization is needed. For instance, if a particular process takes longer than expected due to database queries or network latency, the sequence diagram can highlight these dependencies and guide performance tuning efforts.

Overall, sequence diagrams provide a comprehensive view of how different components of a grocery sales forecasting system interact with each other to execute processes, handle errors, integrate with external systems, and deliver value to users.

3.7 COMPONENT DIAGRAM:

Component diagrams are highly beneficial in understanding the physical components of a system and their relationships, dependencies, and interactions. In a grocery sales forecasting project, a component diagram can provide a clear overview of the various software and hardware components involved, helping stakeholders understand the system's architecture and aiding in development, deployment, and maintenance. Here's how component diagrams can be used in such a project:

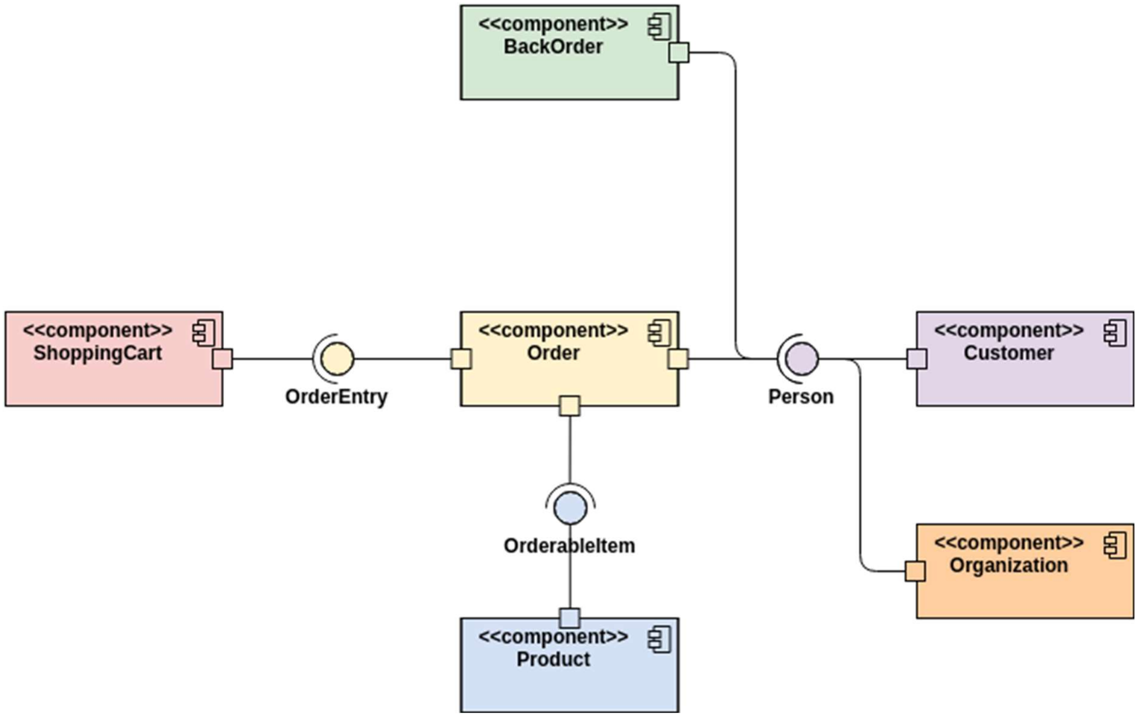


Fig 3.10 Component Diagram

Component Identification:

- The first step in creating a component diagram for a grocery sales forecasting project is identifying the main components of the system. This could include software components such as user interfaces, databases, forecasting algorithms, inventory management systems, and external APIs, as well as hardware components such as servers, databases, and networking infrastructure.

Component Relationships:

Component diagrams show how different components of the system are related to each other. For example:

- The user interface component may depend on backend components for data retrieval and processing.
- The forecasting algorithm component may depend on historical sales data stored in the database.

- The inventory management system component may interact with external APIs for real-time inventory updates or supplier information.

Interfaces and Dependencies:

Component diagrams highlight the interfaces between components and the dependencies between them. This includes communication protocols, data formats, and service contracts used for interaction.

- The user interface component may communicate with backend components via RESTful APIs or WebSocket protocols.
- The database component may depend on a specific database management system (DBMS) such as MySQL or MongoDB.
- The forecasting algorithm component may require certain input data formats and produce output data in a specified format for consumption by other components.

Deployment Considerations:

Component diagrams can also incorporate deployment considerations, indicating how software components are distributed across hardware nodes or servers. This includes details such as:

- Which components run on which servers or virtual machines.
- How components are scaled or replicated for high availability and performance.
- Load balancing configurations for distributing incoming requests across multiple instances of a component.

Technology Stack Visualization:

- Component diagrams provide a concise visualization of the technology stack used in the system, including programming languages, frameworks, libraries, and infrastructure components. This helps stakeholders understand the technological choices made during system design and development.

Overall, component diagrams serve as a valuable tool for system architects, developers, and other stakeholders involved in the grocery sales forecasting project, providing a clear and structured view of the system's architecture, components, relationships, and deployment considerations.

CHAPTER-4
IMPLEMENTATION

4.1 TECHNOLOGY USED

PYTHON:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. Python, the reference implementation of Python, is opens source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non- profit Python Software Foundation. Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system.

Van Rossum remains Python's principal author. His continuing central role in Python's development is reflected in the title given to him by the Python community: Benevolent Dictator For Life (BDFL). On the origins of Python, Van Rossum wrote in 1996:

In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to

Unix/ hackers . I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus)."

Guido van Rossum Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed.

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward- compatible with previous versions. However, many of its major features have been back ported to the Python 2.6.x and 2.7.x version series, and releases of Python 3 include the utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to2020 out of concern that a large body of existing code could not easily be forward-ported to Python3. Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and Python 3.7.0b1(PEP 540) adds a new "UTF-8 Mode" (and overrides POSIX locale). In January 2017, Google announced work on a Python 2.7 to go compiler to improve performance under concurrent workloads.

Readability and Simplicity: Python's syntax is clean, easy to read, and resembles English, making it accessible for beginners and experienced programmers alike. This simplicity leads to faster development and easier maintenance of projects.

Scalability: While Python is often praised for its ease of use and simplicity, it is also scalable and capable of handling projects of all sizes. Whether you're building a small script or a large-scale application, Python can accommodate your needs.

Data Science and Machine Learning: Python's rich ecosystem of data science libraries, such as Pandas, NumPy, SciPy, and scikit-learn, along with its dominance in machine learning frameworks like TensorFlow and PyTorch, make it the language of choice for data analysis, machine learning, and artificial intelligence projects.

4.2 SAMPLE CODE:

```
# Read CSV files into pandas DataFrames
train = pd.read_csv(r"C:\Users\kondr\Downloads\data\train.csv")
test = pd.read_csv(r"C:\Users\kondr\Downloads\data\test.csv")
stores = pd.read_csv(r"C:\Users\kondr\Downloads\data\stores.csv")
transactions =
pd.read_csv(r"C:\Users\kondr\Downloads\data\transactions.csv").sort_values(["store_nbr",
"date"])
# Convert date columns to datetime format
train["date"] = pd.to_datetime(train.date)
test["date"] = pd.to_datetime(test.date)
transactions["date"] = pd.to_datetime(transactions.date)
# Convert data types for specific columns
train.onpromotion = train.onpromotion.astype("float16")
train.sales = train.sales.astype("float32")
stores.cluster = stores.cluster.astype("int8")
# Merge sales data with transaction data
temp = pd.merge(train.groupby(["date", "store_nbr"]).sales.sum().reset_index(), transactions,
how="left")
# Calculate Spearman correlation between total sales and transactions
print("Spearman Correlation between Total Sales and Transactions:
{:,.4f}".format(temp.corr("spearman").sales.loc["transactions"]))
# Visualize transactions over time
px.line(transactions.sort_values(["store_nbr", "date"]), x='date', y='transactions',
color='store_nbr',title="Transactions")
# Visualize transactions over time with box plot
a = transactions.copy()
a["year"] = a.date.dt.year
a["month"] = a.date.dt.month
px.box(a, x="year", y="transactions", color="month", title="Transactions")
# Visualize average transactions by day of the week over the years
a = transactions.copy()
a["year"] = a.date.dt.year
```

```
a["dayofweek"] = a.date.dt.dayofweek + 1
a = a.groupby(["year", "dayofweek"]).transactions.mean().reset_index()
px.line(a, x="dayofweek", y="transactions", color="year", title="Transactions")
# Read and preprocess oil data
oil = pd.read_csv(r"C:\Users\kondr\Downloads\data\oil.csv")
oil["date"] = pd.to_datetime(oil.date)
oil = oil.set_index("date").dcoilwtico.resample("D").sum().reset_index()
oil["dcoilwtico"] = np.where(oil["dcoilwtico"] == 0, np.nan, oil["dcoilwtico"])
oil["dcoilwtico_interpolated"] = oil.dcoilwtico.interpolate()
# Visualize daily oil price
p = oil.melt(id_vars=['date']+list(oil.keys()[5:]), var_name='Legend')
px.line(p.sort_values(["Legend", "date"], ascending=[False, True]), x='date', y='value',
color='Legend',title="Daily Oil Price")
# Merge transaction and oil data
temp = pd.merge(temp, oil, how="left")
# Calculate correlation between daily oil prices and sales/transactions
print("Correlation with Daily Oil Prices")
print(temp.drop(["store_nbr", "dcoilwtico"],
axis=1).corr("spearman").dcoilwtico_interpolated.loc[["sales", "transactions"]], "\n")
# Visualize relationship between daily oil price and sales/transactions
fig, axes = plt.subplots(1, 2, figsize=(15, 5))
temp.plot.scatter(x="dcoilwtico_interpolated", y="transactions", ax=axes[0])
temp.plot.scatter(x="dcoilwtico_interpolated", y="sales", ax=axes[1], color="r")
axes[0].set_title('Daily oil price & Transactions', fontsize=15)
axes[1].set_title('Daily Oil Price & Sales', fontsize=15)
# Calculate and visualize correlation among stores using heatmap
a = train[["store_nbr", "sales"]]
a["ind"] = 1
a["ind"] = a.groupby("store_nbr").ind.cumsum().values
a = pd.pivot(a, index="ind", columns="store_nbr", values="sales").corr()
mask = np.triu(a.corr())
plt.figure(figsize=(20, 20))
sns.heatmap(a, annot=True, fmt='.1f', cmap='coolwarm', square=True, mask=mask,
linewidths=1, cbar=False)
```

```
plt.title("Correlations among stores", fontsize=20)
# Filter out irrelevant data points from the train dataset
print(train.shape)
train = train[~((train.store_nbr == 52) & (train.date < "2017-04-20"))]
train = train[~((train.store_nbr == 22) & (train.date < "2015-10-09"))]
# Similar filtering for other store numbers and dates
# Perform anti-join to remove zero sales data points
c = train.groupby(["store_nbr", "family"]).sales.sum().reset_index().sort_values(["family
```

4.3 REGRESSION:

1.Predective Analysis:

Regression techniques can be utilized to build models that predict future sales based on historical data and other relevant features. In this project, the sales data is likely the target variable, and various features such as date-related information, store attributes, promotions, holidays, etc., can serve as predictors. Here are some regression techniques that can be applied:

- **Linear Regression:**

This is a straightforward approach where sales are predicted as a linear combination of predictor variables. It assumes a linear relationship between the predictors and the target variable.

- **Ridge Regression and Lasso Regression:**

These are regularized versions of linear regression that can handle multicollinearity and prevent overfitting by penalizing large coefficients.

- **Gradient Boosting Regressor:**

This is an ensemble technique that builds multiple decision trees sequentially, where each tree corrects the errors of the previous one. It's often effective in capturing complex nonlinear relationships between features and the target variable.

- **Random Forest Regressor:**

Another ensemble method that builds multiple decision trees and averages their predictions. It's robust to outliers and noise in the data.

2. Feature Importance Analysis:

Regression models can provide insights into which features have the most significant impact on sales prediction. This can help in understanding the drivers of sales and making informed decisions. Techniques like coefficients in linear regression or feature importance in tree-based models can be used for this purpose.

3.Assumption Checking:

Linear regression specifically, it's essential to check the assumptions of the model, such as linearity, normality of residuals, homoscedasticity, and independence of errors. Diagnostic plots and statistical tests can be employed to assess whether these assumptions hold.

4.Model Evaluation:

Once regression models are trained, they need to be evaluated to ensure their effectiveness in predicting sales accurately. Metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), or R-squared can be used to assess model performance.

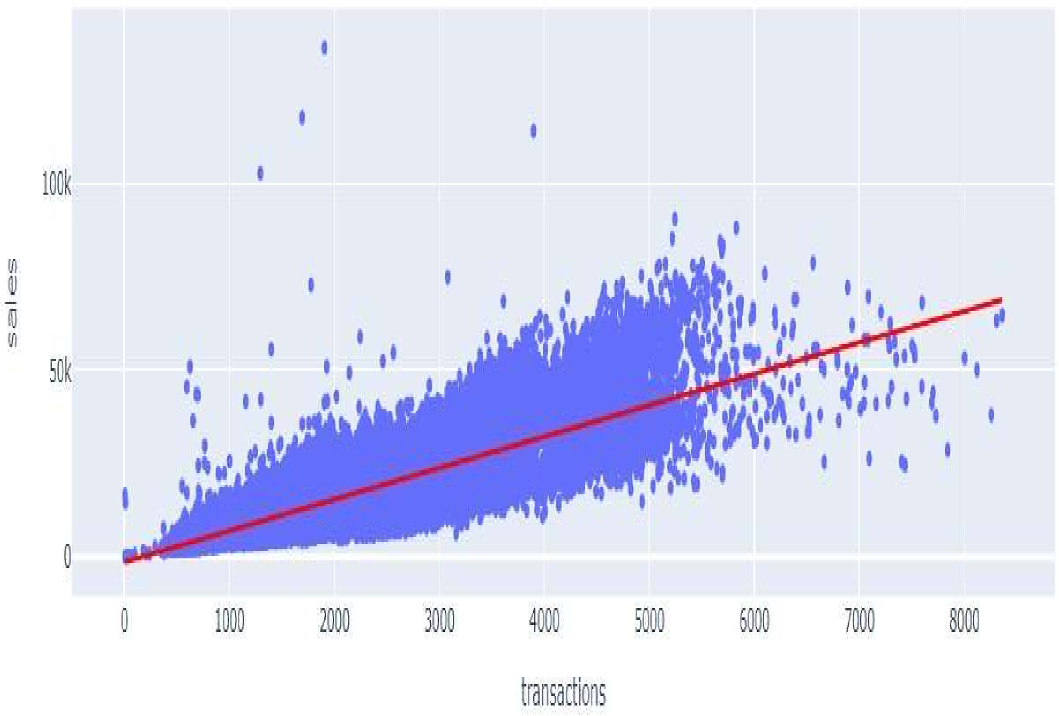


Fig 4.1 Regression

4.4 CORRELATION:

Spearman correlation is used to assess the relationship between different variables, particularly sales and other factors such as transactions, oil prices, and various holiday events. Here's a detailed explanation of its usage in the project:

- **Assessing Correlation between Sales and Transactions:**

The project calculates the Spearman correlation between total sales and transactions. This correlation helps to understand if there's any monotonic relationship between sales and the number of transactions across different stores and dates.

- **Correlation with Daily Oil Prices**:

Daily oil prices are an external factor that could potentially influence sales. By computing the Spearman correlation between daily oil prices and sales, the project aims to determine whether there's any association between them.

- **Correlation between Sales and Oil Prices for Different Product Families:**

The project further investigates the correlation between sales and oil prices for different product families. It calculates the Spearman correlation individually for each product family to understand if the relationship between sales and oil prices varies across different types of products.

- **Assessing Correlations among Stores:**

The project analyses correlations among different stores based on their sales data. This can help identify similarities or differences in sales patterns between different stores. The Spearman correlation is likely used to compute pairwise correlations among stores.

- **A/B Testing:**

A/B testing is performed to compare different groups based on certain categorical variables (such as events, holidays) and their impact on sales. While A/B testing typically focuses on comparing means or proportions between groups, correlation might not be directly involved in this part of the analysis.

- **Seasonal Analysis:**

Spearman correlation can be used to assess the relationship between sales and seasonal factors such as holidays, weather conditions, or special events. By ranking the sales data and the seasonal factors, you can determine whether there is a monotonic relationship between them and identify which factors have the strongest influence on sales during different seasons.

4.5 CLUSTERING:

Clustering can be used in grocery sales forecasting projects to segment customers based on their purchasing behaviour, preferences, and other relevant characteristics. Here's how clustering can be applied in such projects:

Customer Segmentation: Clustering techniques can help identify different segments of customers based on their purchasing patterns. For example, clustering algorithms like k-means clustering or hierarchical clustering can group customers into clusters based on factors such as the types of products they purchase, the frequency of purchases, and the total amount spent. These customer segments can then be used to tailor marketing strategies, promotions, and product offerings to better meet the needs of each segment.

Product Segmentation: Clustering can also be used to group products based on their sales patterns. By clustering products together based on factors such as sales volume, seasonality, and price elasticity, retailers can gain insights into which products are similar in terms of sales behaviour. This information can be used for inventory management, assortment planning, and pricing strategies.

Demand Forecasting: Clustering can be integrated into demand forecasting models to improve the accuracy of sales predictions. By clustering similar products or customers together, forecasting models can capture more nuanced patterns in sales data and make more accurate predictions. For example, clustering techniques can be used to identify groups of products or customers that exhibit similar demand patterns over time, allowing retailers to better anticipate future sales trends.

Market Basket Analysis: Clustering can be combined with association rule mining techniques to perform market basket analysis, which identifies patterns of co-occurring products in customer transactions. By clustering transactions or customer baskets based on the items purchased, retailers can identify frequently co-purchased items and uncover insights into cross-selling opportunities, promotional strategies, and product placement strategies.

Anomaly Detection: Clustering can also be used for anomaly detection to identify unusual or unexpected patterns in sales data. By clustering sales data into normal and anomalous clusters, retailers can detect outliers, anomalies, or irregularities in sales patterns that may indicate issues such as fraud, stockouts, or pricing errors.

Overall, clustering techniques can enhance grocery sales forecasting projects by enabling retailers to segment customers and products effectively, improve demand forecasting accuracy, uncover hidden patterns in sales data, and detect anomalies or irregularities in sales patterns.

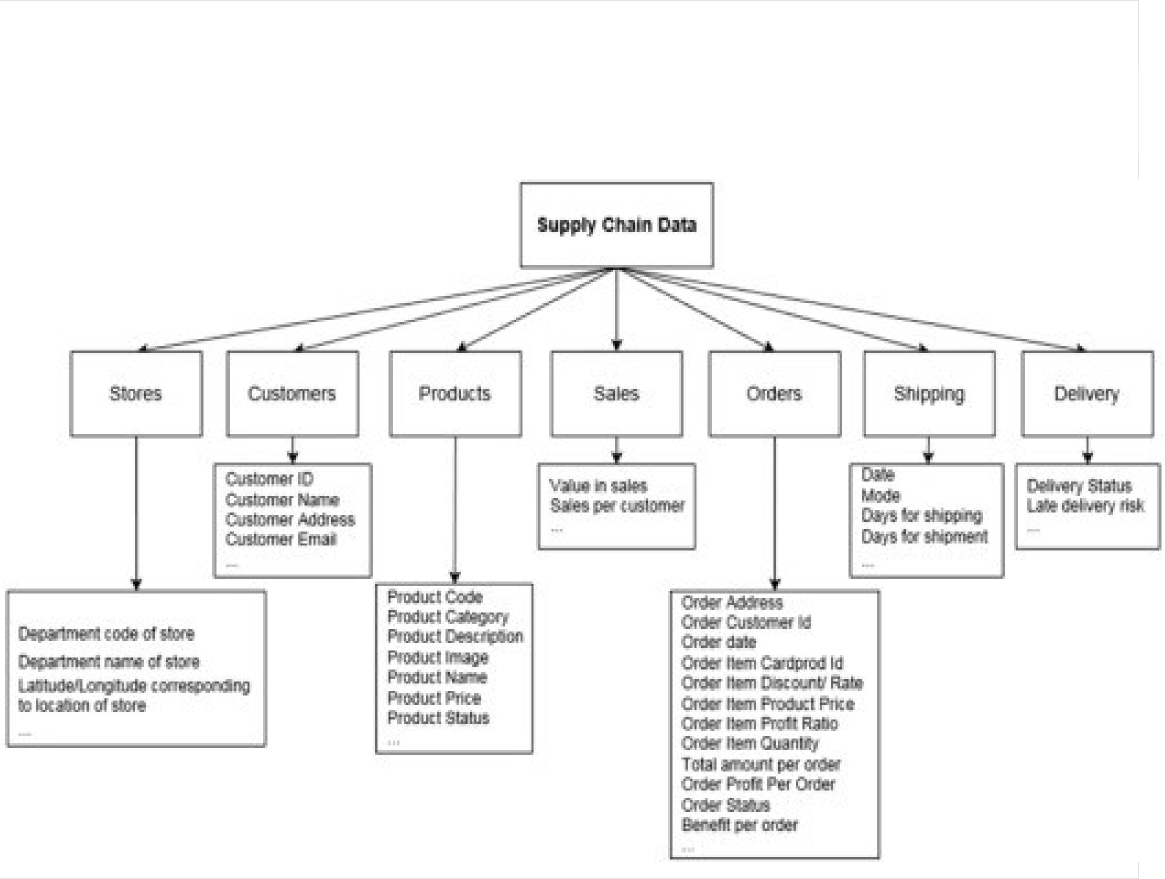


Fig 4.2 Clustering

CHAPTER-5

TESTING

5.1 TESTING PRINCIPLES:

When it comes to testing principles for a grocery sales forecasting project, it's essential to ensure accuracy, reliability, and adaptability of the forecasting model. Here are some key principles you can follow:

1. **Data Quality Testing**:
    - Verify the integrity and completeness of the data used for forecasting.
    - Check for missing values, outliers, and inconsistencies.
    - Ensure data is correctly formatted and standardized.

2. **Model Validation**:
    - Validate the forecasting model against historical data to ensure it accurately predicts past sales.
    - Use techniques like cross-validation to assess the model's performance across different subsets of the data.
    - Compare the forecasted values against actual sales to measure accuracy.

3. **Robustness Testing**:
    - Test the model's performance under different conditions, such as varying time periods, product categories, or geographic regions.
    - Assess how the model handles changes in data patterns, seasonality, and external factors (e.g., holidays, promotions, economic conditions).

4. **Sensitivity Analysis**:
    - Conduct sensitivity analysis to evaluate how changes in input variables impact the forecasted sales.
    - Identify which factors have the most significant influence on sales forecasts.

5. **Forecasting Horizon Testing**:
    - Evaluate the model's performance for different forecasting horizons (e.g., daily, weekly, monthly).
    - Determine the appropriate level of granularity for forecasting based on business needs and data availability.

6. **Benchmarking**:
    - Compare the performance of the forecasting model against alternative methods (e.g., statistical models, machine learning algorithms).
    - Identify the most accurate and efficient approach for grocery sales forecasting.

7. **Real-time Testing**:
    - Implement mechanisms to continuously monitor and evaluate the model's performance in real-time.
    - Update forecasts regularly based on new data and assess the model's ability to adapt to changes.

8. **User Acceptance Testing**:
    - Involve stakeholders in testing to ensure the forecasted results meet their expectations and requirements.
    - Gather feedback from end-users to identify areas for improvement and refinement.

9. **Documentation and Reporting**:
    - Document the testing process, including methodologies, assumptions, and results.
    - Provide clear and concise reports summarizing the model's performance and any identified issues or recommendations for improvement.

5.2 Tests Used in This Project:

In a grocery sales forecasting project, various tests and methodologies can be employed to assess the accuracy and reliability of the forecasting model. Here are some common tests used in such projects:

1. **Mean Absolute Error (MAE):**

- MAE measures the average absolute difference between actual and forecasted sales values.
- It provides a simple and intuitive measure of forecast accuracy.

2. **Mean Absolute Percentage Error (MAPE):**

- MAPE calculates the percentage error between actual and forecasted sales values.
- It helps assess the relative accuracy of the forecasts, especially when comparing across different products or time periods.

3. **Root Mean Square Error (RMSE):**

- RMSE calculates the square root of the average squared differences between actual and forecasted values.
- It penalizes large errors more heavily than MAE and provides a measure of the model's predictive performance.

4. **Forecast Bias:**

- Forecast bias measures the tendency of the forecasting model to systematically overestimate or underestimate sales.
- It helps identify any systematic errors in the forecasting process that need to be corrected.

5. **Tracking Signal:**

- The tracking signal assesses whether the forecast errors are within acceptable bounds over time.
- It helps detect if the forecasting model is consistently over or under forecasting, indicating the need for adjustment.

6. **Forecast Error Decomposition**:

- Decomposing forecast errors into components such as trend, seasonality, and random variation can provide insights into the sources of forecast inaccuracies.
- Techniques like decomposition analysis (e.g., using methods like seasonal decomposition of time series) can be employed for this purpose.

7. **Back testing:**

- Back testing involves testing the forecasting model's performance on historical data that was not used during model development.
- It helps validate the model's ability to generalize to new data and assess its predictive accuracy under different scenarios.

8. **Cross-Validation:**

- Cross-validation techniques, such as k-fold cross-validation, partition the data into multiple subsets for training and testing.
- They help assess the model's performance across different data subsets and reduce the risk of overfitting.

9. **Rolling Forecast Origin:**

- Rolling forecast origin involves iteratively retraining the forecasting model with updated data and evaluating its performance on a rolling basis.
- It simulates real-world forecasting scenarios where the model needs to adapt to changing data over time.

10. **Outlier Detection:**

- Outlier detection techniques identify and handle anomalous data points that could distort the forecasting results.
- Methods like statistical tests, clustering, or machine learning algorithms can be used to identify outliers.

CHAPTER-6
RESULTS

The final results of a grocery sales forecasting project can vary depending on the specific objectives, the quality of data and features, the chosen modelling approach, and the evaluation metrics used.

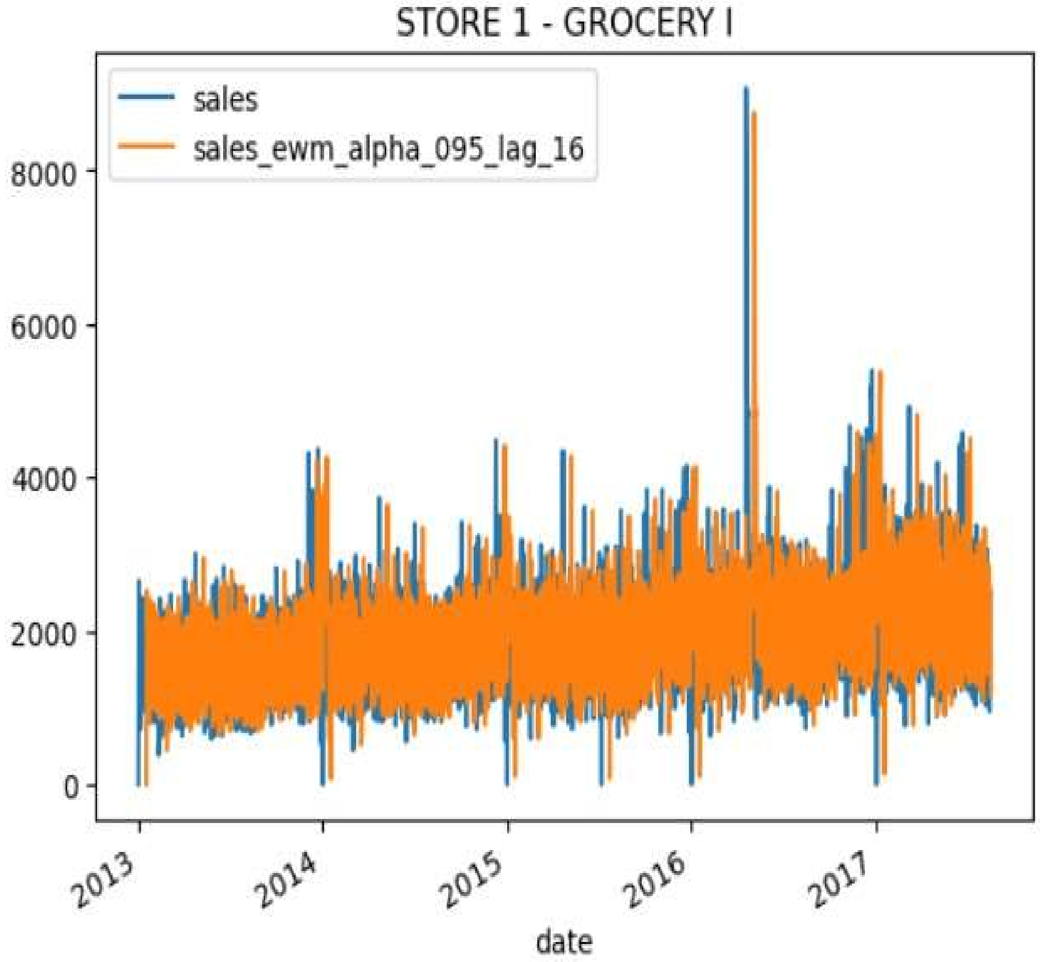


Fig 6.1 Grocery graph

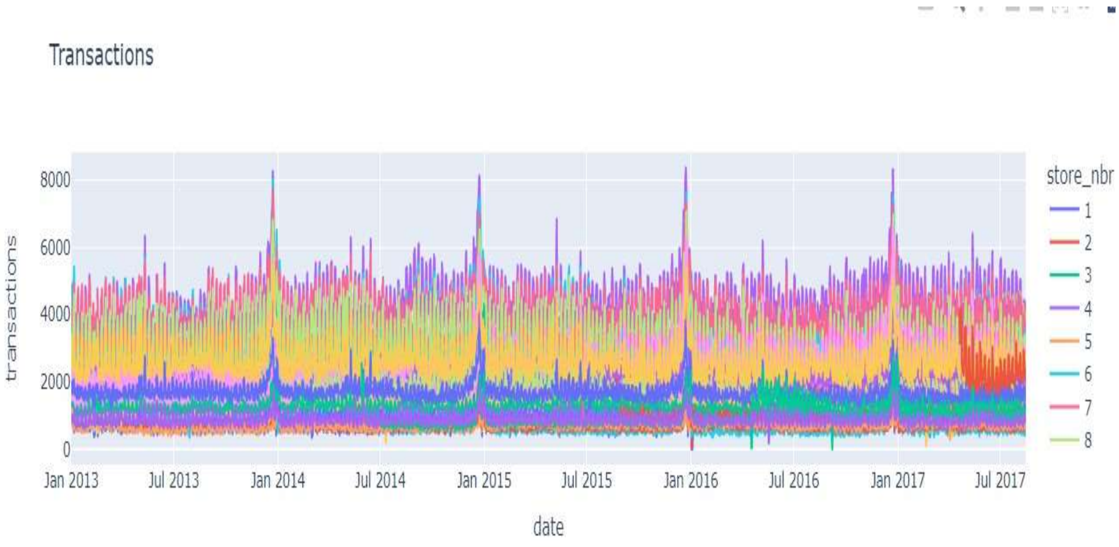
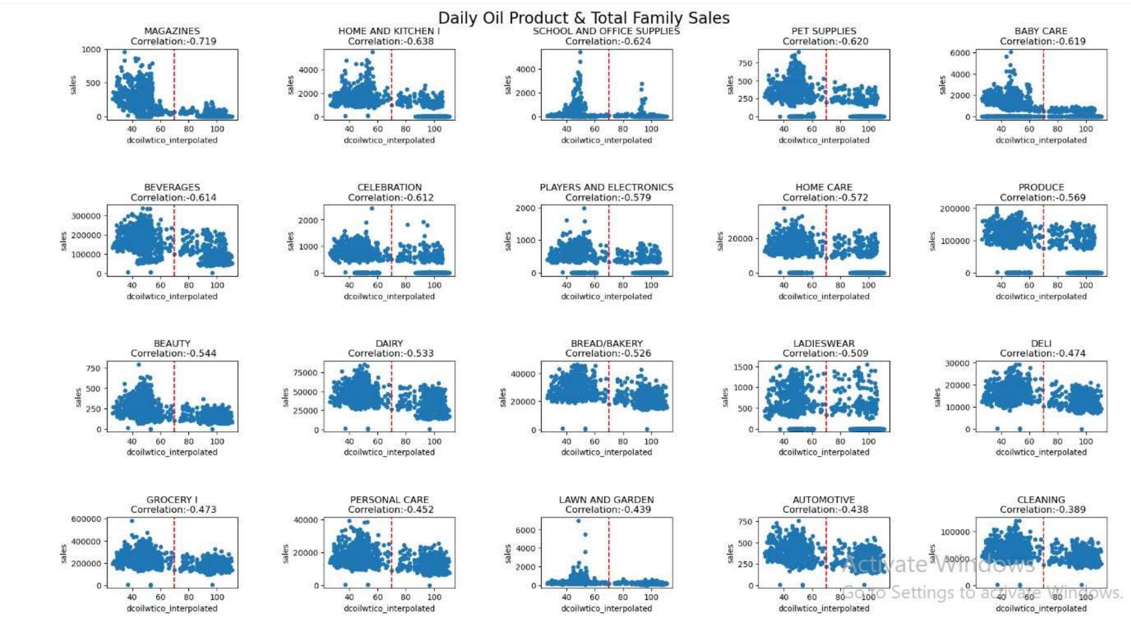


Fig 6.2 Clustering Transaction



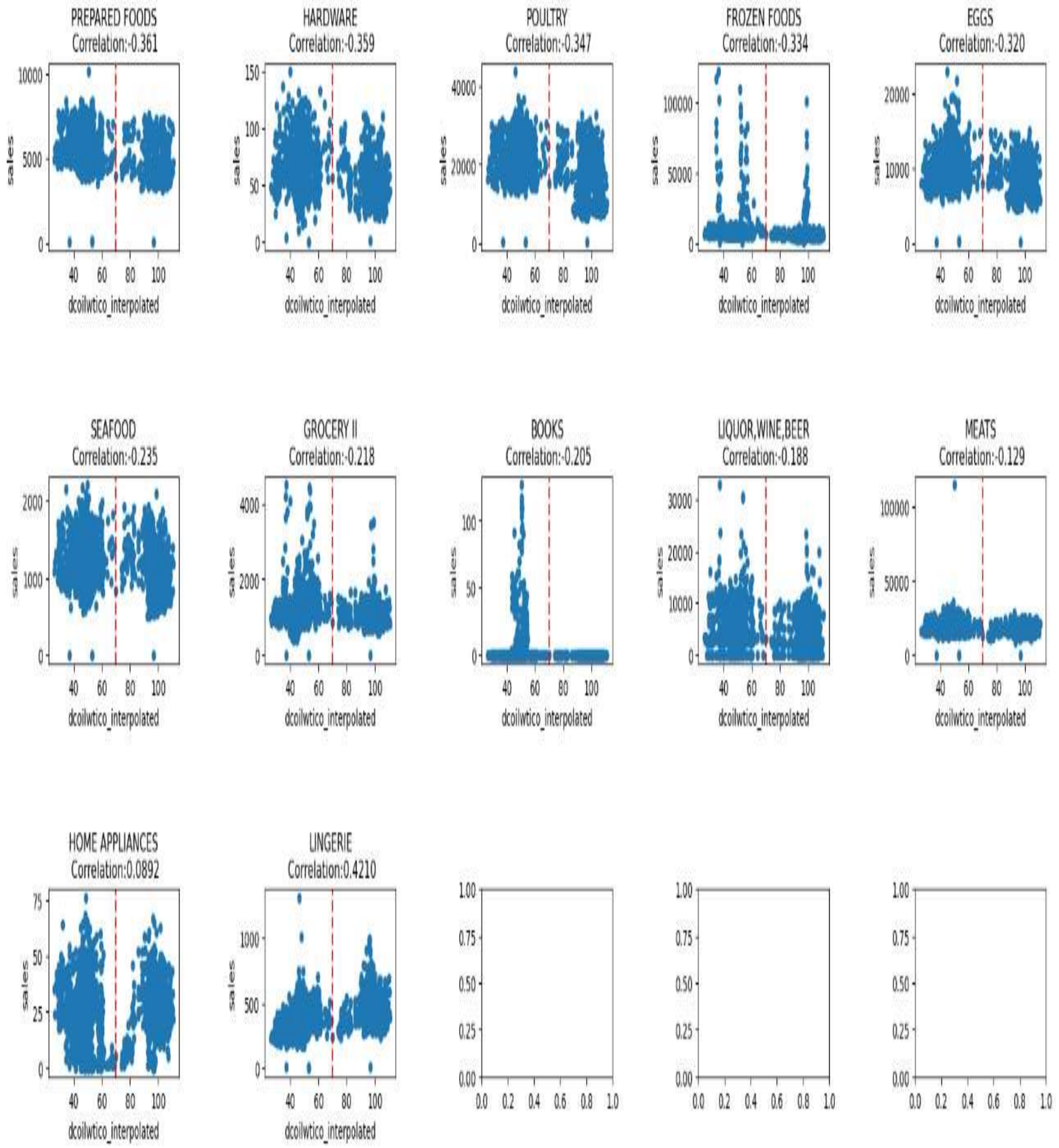


Fig 6.3 Oil Details

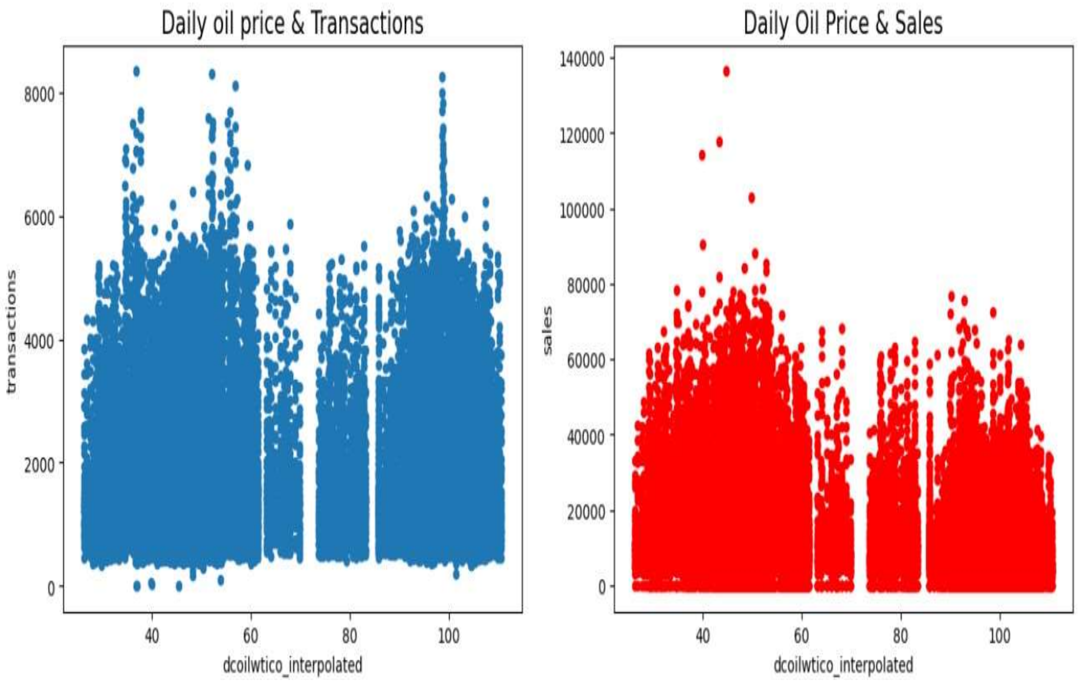


Fig 6.4 Oil Transaction Sales

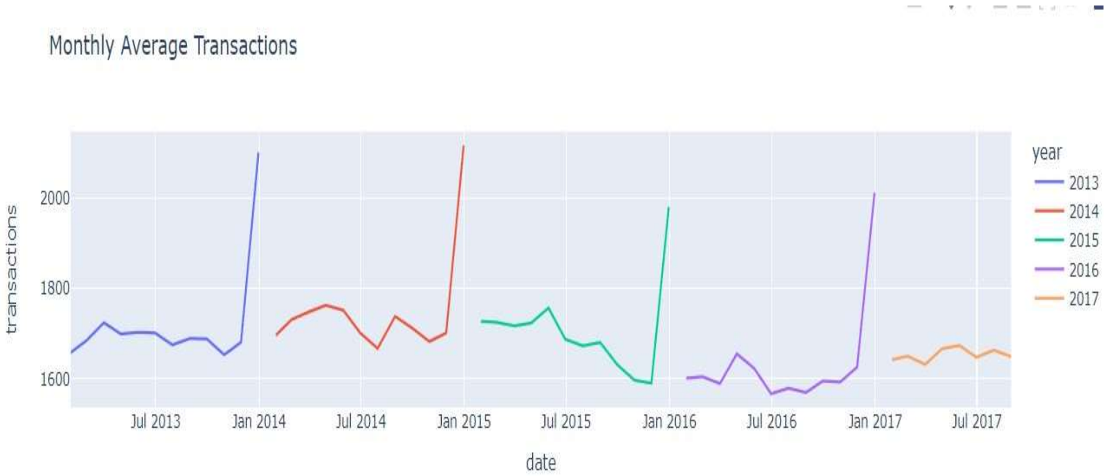


Fig 6.5 Average Transaction

CHAPTER -7

CONCLIUSION

In conclusion, developing accurate and reliable models for grocery sales forecasting is essential for optimizing inventory management, ensuring product availability, and ultimately maximizing business profitability. Through a data-driven approach, leveraging advanced machine learning techniques, businesses can harness historical sales data and relevant features to predict future sales with precision.

Key steps in this process include data collection from various sources, thorough data preprocessing to handle missing values, outliers, and seasonality, feature selection to identify impactful predictors, and model selection and training using suitable algorithms like ARIMA, LSTM, or XGBoost. Additionally, deploying the trained models into production systems allows for real-time forecasting, integration with inventory management systems, and automated decision-making processes.

Continuous monitoring and optimization of the forecasting models enable adaptation to changing market dynamics and ensure sustained accuracy and reliability. By following these steps and utilizing preprocessing techniques tailored for grocery sales forecasting, businesses can gain valuable insights into consumer behaviour, optimize inventory levels, and enhance overall operational efficiency and profitability in the competitive retail landscape.

CHAPTER-8
FUTURE SCOPE AND ENHANCEMENT

The future scope for a grocery sales forecasting project using advanced machine learning techniques is vast and promising. Here are some potential directions for further development and application:

1. **Integration of Real-Time Data:** Incorporating real-time data streams such as weather conditions, economic indicators, social media trends, and local events can enhance the accuracy of sales predictions. Integrating these dynamic factors into the forecasting models can provide more timely and precise insights for inventory management and marketing strategies.

2. **Enhanced Feature Engineering:** Continuously refining and expanding the set of features used in the models can improve prediction accuracy. Exploring new data sources and experimenting with different feature combinations, transformations, and interactions can uncover additional patterns and insights relevant to grocery sales forecasting.

3. **Model Ensemble Techniques:** Employing ensemble learning methods, such as bagging, boosting, or stacking, can further boost prediction performance by combining multiple models' predictions. Ensemble techniques can help mitigate individual model biases and variability, leading to more robust and reliable forecasts.

4. **Spatial and Temporal Analysis:** Considering spatial and temporal factors such as geographical location, seasonality, and time trends can enhance the granularity and specificity of sales predictions. Developing models that account for regional variations in consumer behaviour and market dynamics can tailor forecasting strategies to specific locations and demographics.

5. **Dynamic Pricing and Promotion Optimization:** Integrating sales forecasts with pricing and promotion optimization algorithms can enable retailers to dynamically adjust pricing strategies and promotional campaigns in response to predicted demand fluctuations. This proactive approach can help maximize revenue and profitability while maintaining competitive pricing.

6. **Customer Segmentation and Personalization:** Leveraging customer data for segmentation and personalized recommendations can enhance the relevance and effectiveness of sales forecasts. Developing models that differentiate between various customer segments and predict individual purchasing behaviour can enable targeted marketing efforts and tailored product offerings.

7. **Supply Chain Optimization:** Extending sales forecasting to encompass supply chain optimization can streamline inventory management, reduce stockouts and overstock situations, and minimize logistical costs. Integrating forecasting models with inventory replenishment systems and distribution networks can improve operational efficiency and resilience.

8. **Adoption of Advanced AI Techniques**: Exploring cutting-edge AI techniques such as deep learning, reinforcement learning, and neural architecture search can unlock new capabilities and insights for grocery sales forecasting. These advanced approaches can handle complex data patterns and interactions, leading to more accurate predictions and actionable insights.

9. **Deployment of Cloud-Based Solutions**: Leveraging cloud computing platforms and scalable infrastructure can facilitate the deployment and scalability of grocery sales forecasting models. Cloud-based solutions offer flexibility, cost-effectiveness, and accessibility, enabling retailers to efficiently manage and scale their forecasting systems as needed.

10. **Continuous Model Evaluation and Refinement**: Implementing robust validation frameworks and performance metrics to continuously evaluate and refine forecasting models is essential for maintaining their effectiveness over time. Monitoring model performance, identifying areas for improvement, and iterating on model design and training processes can ensure that the forecasting system remains reliable and adaptive in dynamic retail environments.

CHAPTER-9

REFERENCES

1. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice (2nd ed.). OTexts. [Online Book] Available at: https://otexts.com/fpp2/

2. Chen, C. T., & Zhang, C. Y. (2017). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences, 441-442, 148-167.

3. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 1135-1144.

4. Zheng, X., Padmanabhan, B., & Bao, Y. (2018). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS '18), 690-700.

5. Hossain, M. S., Muhammad, G., & Muhammad, Z. (2019). An ensemble approach for demand forecasting in retail using machine learning. Expert Systems with Applications, 132, 67-79.