VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering)

Department of Computer Engineering

Project Report on

SmartCart - Recommendation System for Supermarket Sales

Submitted in partial fulfillment of the requirements of Third Year (Semester–VI), Bachelor of Engineering Degree in Computer Engineering at the University of Mumbai Academic Year 2024-25

By

1. Aditya Joshi D12C- 34
2. Ved Shirur D12C-60
3. Honey Kundla D12C- 69
4. Chetan Narang D12C-45

Project Mentor

Dr. Nupur Giri

University of Mumbai

(AY 2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering)

Department of Computer Engineering

CERTIFICATE

This is to certify that ____ of Third Year Computer Engineering studying under the University of Mumbai has satisfactorily presented the project on "---------------------------" as a part of the coursework of Mini Project 2B for Semester-VI under the guidance of Prof. Dr. Nupur Giri in the year 2024-25.

Date

Internal Examiner

External Examiner

Project Mentor	Head of the Department	Principal
Dr. Mrs. Nupur Giri	Dr. Mrs. Nupur Giri	Dr. J. M. Nair

Mini Project Approval

This Mini Project entitled "**SmartCart - Recommendation System for Supermarket Sales** " by **Aditya Joshi (34), Ved Shirur (60), Honey Kundla (69), Chetan Narang (45)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……………………………………….
(Internal Examiner Name & Sign)

2……………………………………….
(External Examiner name & Sign)

Date: Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

--	--
(Signature)	(Signature)
Aditya Joshi (34)	Ved Shirur (60)
--	--
(Signature)	(Signature)
Honey Kundla (69)	Chetan Narang (45)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Dr.(Mrs.) Nupur Giri** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR T.E MINI PROJECT 2B

Learners will be to:-

CO No.	COURSE OUTCOME
CO1	Identify problems based on societal /research needs.
CO2	Apply Knowledge and skill to solve societal problems in a group.
CO3	Develop interpersonal skills to work as a member of a group or leader.
CO4	Draw the proper inferences from available results through theoretical/ experimental/simulations.
CO5	Analyze the impact of solutions in societal and environmental context for sustainable development.
CO6	Use standard norms of engineering practices
CO7	Excel in written and oral communication.
CO8	Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
CO9	Demonstrate project management principles during project work.

Index

Abstract

In today's dynamic retail landscape, supermarkets generate vast volumes of transactional data daily. However, a significant portion of this valuable data remains underutilized. To address this, we introduce SmartCart, an intelligent recommendation system designed to enhance the overall shopping experience by leveraging historical purchase patterns. By applying unsupervised machine learning techniques, particularly clustering algorithms, SmartCart identifies shopping trends and groups similar products based on customer behavior. These clusters are then used to generate personalized product suggestions tailored to individual user preferences.

One of the standout features of SmartCart is its ability to accept uploaded product lists, which are then analyzed in real time to provide customized recommendations. This not only aids customers in making faster and more informed purchasing decisions but also empowers businesses to increase product visibility, customer retention, and operational efficiency. Importantly, SmartCart operates independently of any payment gateway, making it an ideal solution for integration into existing retail platforms without the need for complex financial infrastructure. By tapping into the power of machine learning, SmartCart transforms raw transactional data into actionable insights, creating a win-win scenario for both consumers and businesses.

Chapter 1: Introduction

1.1 Introduction

SmartCart is a smart and user-friendly product recommendation system developed to simplify and enhance the supermarket shopping experience. In today's fast-paced world, customers often look for quicker and more personalized shopping assistance. SmartCart addresses this need by analyzing customer preferences based on their input file, which contains a list of selected or purchased products.

The system uses clustering algorithms to identify patterns and similarities among products, allowing it to recommend additional relevant items to the customer. This intelligent recommendation helps users discover useful products they might have missed, thus improving customer satisfaction and boosting supermarket sales.

SmartCart is designed with simplicity in mind — it does not involve complex features like payment gateways. Instead, it focuses on core functionality: uploading product data, identifying patterns, and providing accurate recommendations. The project combines data mining, machine learning techniques, and a web-based interface to deliver a smooth and efficient experience.

1.2 Motivation

In today's world, supermarkets offer thousands of products, and customers often find it difficult to identify what they truly need or what complements their current selection. Most traditional shopping systems lack intelligent assistance, leading to time-consuming and confusing shopping experiences.

Our motivation behind developing SmartCart was to create a solution that understands customer needs and helps them make better choices. Inspired by how platforms like Amazon or Flipkart suggest related items, we wanted to bring that intelligence into a simpler supermarket setup without involving complex e-commerce systems.

By using clustering and product analysis, SmartCart aims to replicate a personalized shopping assistant that helps customers find the right products based on their own selections. This not only saves time but also improves the overall shopping experience, making it smarter, quicker, and more customer-friendly.

1.3 Problem Definition

In traditional supermarket shopping, customers often struggle to find related or complementary products based on their choices. There is no intelligent system to analyze their buying behavior or recommend useful items in real-time. This results in missed opportunities for both customers (who may not find what they need) and supermarkets (who lose potential sales).

The core problem is the **lack of a smart recommendation system** that can personalize product suggestions based on a customer's selected list. Additionally, there is a need for a **simple interface** where users can upload their product file and receive meaningful recommendations without the complexity of login systems, payment gateways, or manual search efforts.

SmartCart addresses this problem by using clustering techniques to analyze the product data and recommend similar or commonly associated items. This improves the shopping experience, enhances product visibility, and simplifies the decision-making process for customers.

Objectives:

1. **To design a smart recommendation system** that suggests relevant supermarket products based on the user's selected item list.

2. **To implement clustering algorithms** for grouping similar products and identifying buying patterns.

3. **To provide a simple file upload interface** where customers can upload product data in a hassle-free manner.

4. **To enhance customer experience** by reducing search time and suggesting useful product combinations.

5. **To avoid unnecessary complexities** such as payment gateways or user registration, focusing only on recommendation and product customization

1.4 Existing Systems

Model name	Training Data	Data Size	Accuracy	No. of parameters	Limitations	Training Time
LLama	Books, Wikipedia, GitHub, CommonCrawl, C4, ArXiv, and StackExchange	1.4T tokens	Llama-2-70b : 81.7%.	65 Billion parameters	Data Bias and Ethical Concerns	21 days
GPT	Books, websites, and other texts. CommonCrawl dataset	523gb	~80-90%	Trillion parameters	Biased and repetitive	5-6 months (GPT-4)
Bert	BooksCorpus and English Wikipedia	3Tb	79.27	base (110M parameters) and large (345M parameters)	older training data	4 days
Scibert	Semantic scholar papers	1.14M papers, 3.1B tokens.	80%	110 million parameters.	Limited to scientific contexts	-
Falcon	books, websites, articles, and other forms of written content.	5,000 billion tokens,	76.37%	180 billion parameters	can exhibit biases present in the training data.	2 months(Falcon 40B)
t5	C4 dataset	750 Gb 32,128 subword tokens	92.30%	11 billion parameters	Requires substantial computational resources	-
Galactica	48 million papers, textbooks, and other scientific knowledge sources	106B tokens	50%	120 Billion parameters	occurrence of hallucinations.	-
Skywork	data filtered from Chinese web pages	3.2T tokens	90%	13B	Biasness and scalability	--
Bloom	ROOTS corpus	366B tokens	-	176B	Outdated or incorrect information for current events.	105 days

StarCoder	The Stack with 384 Programming Languages and Github repositories	1 trillion tokens sourced	86.6	15.5B parameter	Ethical Concerns, Malicious code	1 to 2 months
GPT-Neo X	Pile, Books, Internet Resources, Github, youtube subtitles,	20B	90 %	20 billion parameters,	Data duplication Lack of coding evaluations	34 days

Table 1: Comparisons of LLM models

1.5 Lacuna of the existing systems

1. **Lack of Personalization:** Many current recommendation systems do not fully utilize customer data to provide personalized suggestions, leading to generic recommendations that may not resonate with individual shoppers.

2. **Limited Real-Time Capabilities:** Existing systems often fail to offer real-time recommendations, missing opportunities to influence customer decisions during the shopping process.

3. **Inefficient Data Usage:** Some systems struggle to effectively analyze and utilize the vast amount of customer data available, resulting in less accurate and relevant product suggestions.

4. **Complex Implementation:** Many recommendation systems are complex and challenging to implement, requiring significant time and resources, which can be a barrier for smaller supermarkets.

5. **Suboptimal Inventory Management:** Current systems may not effectively tie recommendations to inventory levels, potentially leading to stock issues or missed sales opportunities.

1.6 Relevance of the Project

In an era where personalization is key to improving customer satisfaction, intelligent recommendation systems have become essential in both online and offline shopping experiences. While major e-commerce platforms use such systems extensively, local supermarkets and small retail setups still lack this capability.

SmartCart bridges this gap by offering a simple yet effective solution that brings the power of recommendation to supermarket environments. By analyzing user-selected products and suggesting related items, it not only enhances the customer's shopping journey but also helps supermarkets boost their sales without investing in heavy infrastructure.

The project is highly relevant in today's data-driven world, where businesses seek ways to improve customer engagement using intelligent automation. SmartCart demonstrates how machine learning and clustering can be used practically in retail scenarios, making it a timely and meaningful innovation.

Chapter 2: Literature Survey

A. Overview of Literature Survey

Paper Title	Inference
1. Generating Fact Checking Explanations	1. DistilBERT Implementation 2. First to Generate Explanations
2.Fake News Detection Using Deep Learning and Natural Language Processing	1. Used Word2Vec and LSTM Models 2. Factors Affecting System Accuracy: Training Iterations, Data Diversity, Vector Size 3. Achieved 90% Accuracy
3.End-to-End Multimodal Fact-Checking and Explanation Generation: A Challenging Dataset and Models	1. Performs a comparative study for the existing datasets used to train fact-checking models. 2. Multimodal
4.Comparative Study of Supervised Learning Algorithms for Fake News Classification	1. Comparative Study: Logistic Regression, Random Forest, SVM, Gradient Boosting (Best: Random Forest) 2. 99.7% Accuracy with Gradient Boosting Classifier
5.A Novel Text Resemblance Index Method for Reference-based Fact-checking	1. Performs a comparative study for the existing datasets used to train fact-checking models. 2. Use of Veracity Scanning Model and Text Resemblance Score 3. Achieves 82.31% accuracy
6.Token-Level Fact Correction in Abstractive Summarization	1. Token-Level Fact Correction for Abstractive Summarization 2. Improved Consistency & Summarization Performance 3. Accuracy : 81.04(BERTScore)
7.A Hybrid Framework Integrating LLM and ANFIS for Explainable Fact-Checking	1. LLM & ANFIS Model Integration 2. 0.9 F1 Score on FEVER Dataset
8.Automated Fact Checking Using A Knowledge Graph-based Model	1. ConVe Model is trained on 2 KGs made with Liar datasets 2. 88% precision

Table 2: Literature Survey

B. Related Works

2.1 Research Papers Referred

Name of Paper	Abstract	Inference drawn
Fake News Detection Using Machine Learning and Web scraping	Technology has made tasks easier, and social media, originally for recreation and socializing, has become integral to daily life. Nowadays, people often use social media to check news feeds. However, the trustworthiness of the news shared online is a major concern, as fake news and hoaxes spread rapidly. People frequently share unverified content, not realizing the potential harm, especially during crises. Trusted sources like Google provide reliable information, but when fake content is searched, irrelevant results appear. In this application, news is checked by comparing it with results from multiple sources online. Using natural language processing, the content is analyzed for similarities—if they match, the news is real; if not, it's likely fake. This helps in identifying and stopping the spread of fake news.	Creating a stack ensemble model using Spark with datasets from Kaggle and KDnuggets.
Automated Fact Checking Using A Knowledge Graph-based Model	Misinformation poses a growing threat to various sectors, including the economy, public health, and democracy. Fact checking is crucial for combating it, but the vast volume of online content makes manual verification difficult. This paper proposes a knowledge graph-based fact-checking model using two separate graphs for true and false claims. The model leverages convolutional neural networks for knowledge graph embeddings, trained to distinguish between true and false information. Additionally, explainable AI (XAI) techniques are used to enhance transparency and user trust while reducing errors.	The model is trained using two knowledge graphs: one for true claims and the other for false claims, utilizing the LIAR dataset.

LSTM Based Approach to Detect Fake News	In today's tech-driven world, people encounter news daily, often through social media, where much of it is unreliable or fake. Fake news spreads misinformation and harms genuine journalism. This paper proposes a solution using Natural Language Processing (NLP) to classify news into four stance labels: agree, disagree, discuss, and unrelated. Using fake news challenge datasets for training, we developed an LSTM-based model that achieves 99% accuracy in classifying news articles effectively.	The paper demonstrates that an LSTM-based model, trained with NLP techniques, can effectively classify news articles and achieve high accuracy in identifying fake news.

Table 3: Research Papers Referred

2.2 Patent Search

After reviewing patents in the domain of automated fake news detection and fact-checking, most inventions revolve around:

- ❖ Machine learning algorithms for verifying content authenticity.
- ❖ Systems that use knowledge bases or real-time data sources to evaluate claims.
- ❖ Integration with browser extensions or web apps for user alerts on fake news.

No patents were found to directly use token-level fact correction or LLM-ANFIS hybrid models, which indicates innovation potential.

2.3 Inference Drawn

- From the research papers and patents:
  Models integrating multiple data sources (multimodal, KGs) yield better results.
- LSTM, Random Forest, and Gradient Boosting are effective in classification tasks.
- Accuracy improves significantly with diverse, high-quality training datasets.
- Few models explain decisions clearly, highlighting the need for explainable AI in this space.

2.4 Comparison with Existing System

The existing systems for fake news detection exhibit a wide range of accuracies (typically 80–99%) depending on the algorithms and datasets used, with many relying on traditional machine learning models and offering limited explainability. While some approaches integrate external knowledge sources like knowledge graphs or employ web scraping, they often lack real-time adaptability. Additionally, very few systems leverage multimodal inputs such as text, image, or video together. In contrast, the proposed approach aims to enhance accuracy through hybrid models combining large language models (LLMs) with ANFIS, improve transparency using explainable AI (XAI) techniques, and incorporate dynamic external verification mechanisms, with a strong focus on expanding multimodal input analysis.

Chapter 3: Requirement Gathering for the Proposed System

3.1 Introduction to requirement gathering

Requirement gathering is the initial and one of the most crucial phases of software development, where the needs, expectations, and constraints of the system are identified and documented. It involves understanding **what the users want**, **how the system should behave**, and **what functionalities it must offer** to solve the real-world problem effectively.For the SmartCart project, requirement gathering helped us define the core features like file upload, product clustering, recommendation display, and user-friendly interface. This process included discussions with users (like regular shoppers), analysis of existing systems, and identification of technical tools needed to build the solution.Effective requirement gathering ensures that the final system meets user needs, avoids unnecessary features, and stays within scope, time, and cost limits.

3.2 Functional Requirements

1. **File Upload Functionality**
   The system should allow users to upload a file (e.g., CSV or Excel) containing a list of selected or purchased products.
2. **Product Analysis and Clustering**
   The system should analyze the uploaded product list and group similar items using clustering algorithms.
3. **Recommendation Generation**
   Based on the uploaded data, the system should generate and display relevant product recommendations.
4. **User Interface for Interaction**
   The system should provide a simple and clean web interface for users to upload files and view recommendations.
   Data Validation
   The system should validate the uploaded file to ensure correct format and relevant product data.
5. **Display of Clustered Products**
   The system should visually show how products are grouped or related, optionally through a graph or categorized list.

3.3 Non-Functional Requirements

1. **Usability**
   The system should provide an intuitive and easy-to-use interface, ensuring that customers can upload their files and view recommendations without confusion.

2. **Performance**
   The system should process uploaded files efficiently, ensuring that product analysis and recommendations are generated in less than 5 seconds for typical product lists.

3. **Scalability**
   The system should be capable of handling a large number of product entries (thousands of products) without a noticeable degradation in performance.

4. **Security**
   The system should ensure that uploaded product data is stored and processed securely, with proper safeguards against unauthorized access.

5. **Compatibility**
   The system should be compatible with major browsers (Chrome, Firefox, Safari) and operate on both Windows and Mac operating systems.

6. **Reliability**
   The system should function without crashes and should be able to handle errors gracefully, providing clear error messages if something goes wrong (e.g., invalid file format).

7. **Maintainability**
   The system's code should be modular and well-documented, making it easy to update and maintain in the future.

3.4. Hardware, Software , Technology and tools utilized

Hardware:

- **Servers:** Host the system and manage data.
- **Computers:** Used for development and testing.
- **Networking Equipment:** Connects all components.

Software:

- **Operating Systems:** Windows or Linux.
- **Database:** MySQL or PHP for storing data.
- **Development Tools:** IDEs like Visual Studio Code or PyCharm.
- **Programming Languages:** Python, Java.
- **Machine Learning Libraries:** TensorFlow or Scikit-Learn.

3.5 Constraints

1. **Data Format**

   The system is constrained to accept only specific file formats (e.g., CSV or Excel) for product data upload. Any unsupported formats must be flagged as errors.

2. **No Payment Gateway**

   The system does not involve any payment processing or user authentication, limiting its functionality to product recommendations only.

3. **Limited Resources**

   The system is designed for small to medium-scale supermarkets, with limited computing resources. It may not support large enterprise-level operations or extremely high-volume product datasets.

4. **File Size Limitations**

   The system is designed to handle product files of up to a certain size (e.g., 10MB), and larger files may cause performance issues or failures.

5. **Clustering Limitations**

   The clustering algorithm may not always produce perfect results, especially if the

Chapter 4: Proposed Design

4.1 Block diagram of the system

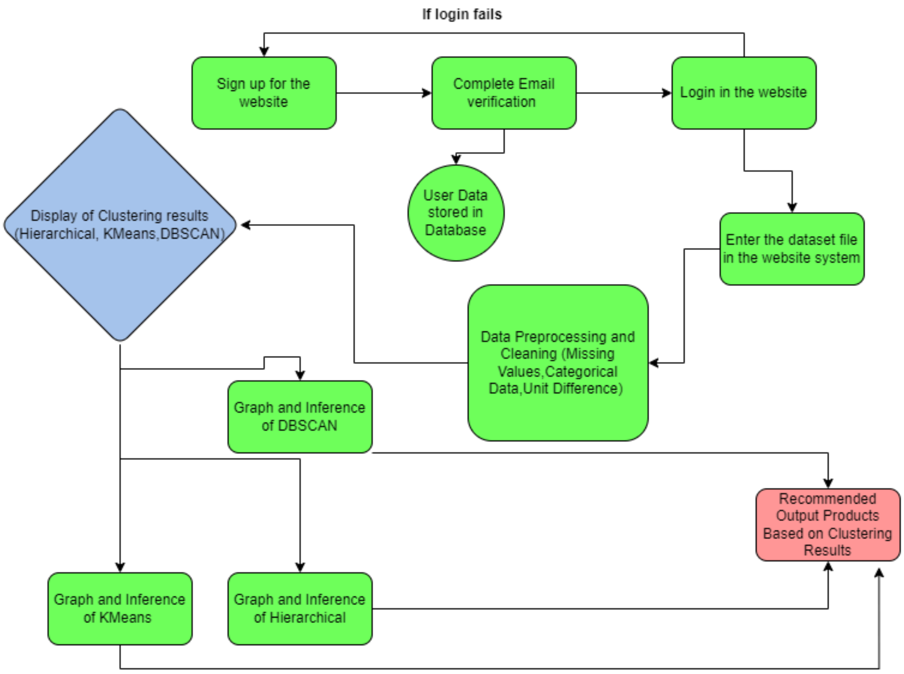


Figure 1: Block Diagram

4.2 Modular design of the system

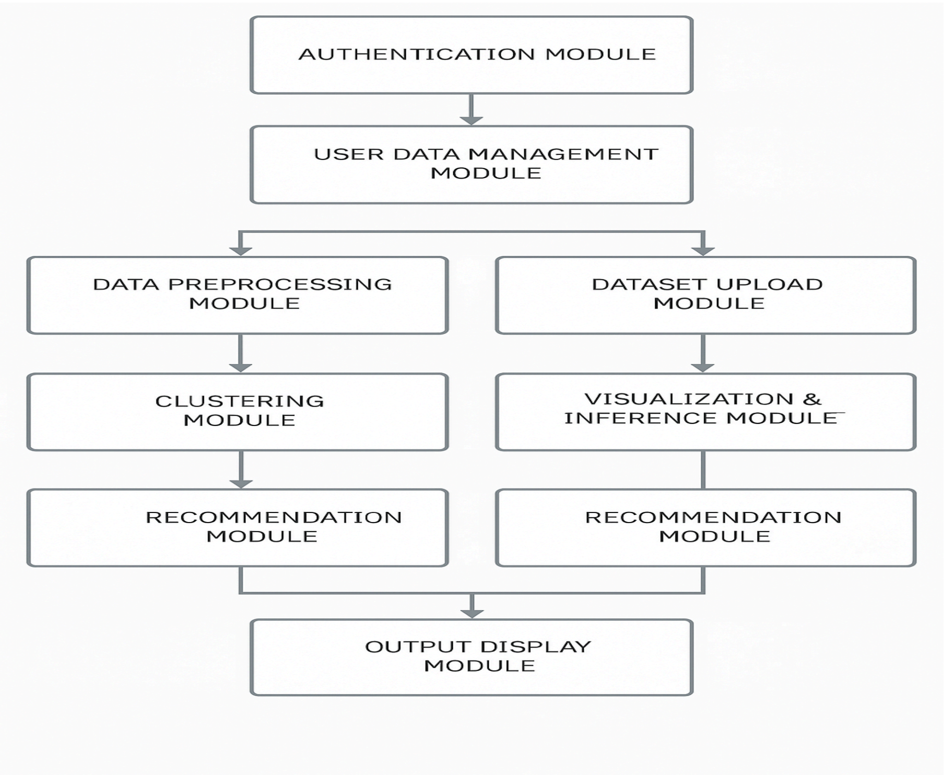


Figure 2: Modular Design of the Project

4.3 Detailed Design

1. User Authentication Module

a. Login System

- **Functionality:** Allows registered users to log into the system.
- **Tech Stack:** Django Authentication / Flask-Login / Firebase Auth (depending on backend).
- **Failure Handling:** Redirects to sign-up if login fails.

b. Sign-Up & Email Verification

- ❖ **Functionality:** Enables new user registration with email confirmation.
- ❖ **Components:** User Form (Name, Email, Password), Email OTP/Token Verification System
- ❖ **Database:** Stores verified users in the user table.

2. Dataset Upload Module

- ★ **Functionality:** After login, users can upload a dataset file (CSV/XLSX).
- ★ **Frontend:** File Upload UI
- ★ **Backend:** File parsing and basic format validation. Store uploaded file temporarily for processing

3. Preprocessing and Cleaning Module

a. Core Tasks:

- Handle Missing Values
- Convert Categorical Data using One-Hot Encoding/Label Encoding
- Normalize data to fix Unit Differences

b. Libraries Used: pandas, scikit-learn, numpy

c. Flow: Dataset → Cleaning Pipeline → Transformed DataFrame

4. Clustering Module

Runs three clustering algorithms in parallel:

a. KMeans Clustering: Scikit-learn's KMeans, Elbow method (optional) for optimal cluster count, **Output:** Cluster labels, Centroids

b. DBSCAN Clustering: Density-based clustering, Outputs: Core points, noise points, clusters

c. Hierarchical Clustering: Agglomerative (bottom-up), Visual Output: Dendrogram, Outputs: Cluster groups based on linkage method.

5. Graphical Inference Module

a. Purpose: Generate visual plots to understand clustering output.

b. Tools: matplotlib, seaborn, plotly

c. Charts: 2D Scatter Plots (with cluster coloring), Dendrogram (for Hierarchical), Cluster Heatmaps (optional)

6. Clustering Results Display Module

a. Visual Dashboard: Shows results for all three clustering techniques. **Includes:** Cluster sizes, number of clusters, graphical view

b. Components: Tabs or collapsible sections for each algorithm. Side-by-side comparison feature (optional)

7. Recommendation Module

a. Functionality: Based on clustering results, suggest output products or insights (e.g., user segments, target groups, anomalies). Here the system suggests the user on what product generates good sales, good profit and bad sales and profit as well.

4.4 Project Scheduling & Tracking : Gantt Chart

Sr. No.	Task	Start Date	End Date	Duration
1	**Requirement Gathering & Research**	Sep 2, 2024	Sep 7, 2024	1 week
2	**System Design (Flowchart, DB Design)**	Sep 9, 2024	Sep 14, 2024	1 week
3	**Frontend UI Design & Wireframing**	Sep 16, 2024	Sep 30, 2024	2 weeks
4	**Backend Setup & Auth Module**	Oct 1, 2024	Oct 15, 2024	2 weeks
5	**Dataset Collection & Upload Module**	Oct 16, 2024	Oct 31, 2024	2 weeks
6	**Data Preprocessing & Cleaning**	Nov 1, 2024	Nov 10, 2024	10 days
7	**Clustering (KMeans, DBSCAN, GMM)**	Nov 11, 2024	Nov 30, 2024	3 weeks
8	**Visualization Dashboard**	Dec 2, 2024	Dec 20, 2024	3 weeks
9	**Recommendation Engine (ML Logic)**	Jan 2, 2025	Jan 15, 2025	2 weeks
10	**Final UI Integration & Results Page**	Jan 16, 2025	Jan 30, 2025	2 weeks
11	**Testing, Debugging & QA**	Jan 31, 2025	Feb 14, 2025	2 weeks

12	**Documentation & Report**	Feb 15, 2025	Mar 1, 2025	2 weeks
13	**Presentation Preparation & Submission**	Mar 2, 2025	Mar 7, 2025	1 week

Table 4: Project Schedule and Gantt Chart

Chapter 5: Implementation of the Proposed System

5.1. Methodology Employed

1. Data Collection

- Use web scraping techniques to extract data from these sources. This includes headlines, article content, publication dates, and author information, etc.
- Compile the extracted data into a structured, personalized dataset in Question and And Answer format, ensuring it covers a comprehensive range of topics within the domain.

2. Data Preprocessing

- Removing duplicates, irrelevant content, and incorrect entries.
- Categorize and tag data according to topics, sub-topics, and relevant metadata.

3. Model Training

- Evaluate current models and algorithms for accuracy and suitability.
- Adjust models to improve accuracy and address domain-specific challenges.
- Find areas for improvement and iterate on model development.

4. Claim Verification

- The system cross-references the claim with the dataset to check for accuracy. The verification process includes matching the claim's content with the information in the dataset and assessing the credibility of sources.

5. Explanation and Sources

- After verification, the model provides with the result, indicating whether the claim is true, false, or uncertain.
- Including a summary of the reasoning behind the verified fact.
- Listing the sources and evidence used in the verification process.

6. UI/UX Design

- Design an interface that allows easy submission of claims and access to verification results.
- Clearly display the verification results, explanations, and sources in an organized manner.

5.3 Dataset Description

The dataset used in this project is sourced from the Superstore sales data and comprises 9,994 entries across 21 attributes. The primary dataset used is the "Orders" sheet, which encapsulates transactional information regarding customer purchases from a retail superstore. Below is a detailed breakdown of the key features:

- ➔ **Order ID, Order Date, Ship Date:** These columns represent unique order identifiers along with their corresponding order and shipping timelines.
- ➔ **Customer ID, Customer Name:** Customer-specific information allowing segmentation and customer-level analysis.
- ➔ **Ship Mode:** Describes the method used for shipment, which includes options like First Class, Second Class, Standard Class, and Same Day.
- ➔ **Segment:** Classifies customers into segments like Consumer, Corporate, and Home Office.
- ➔ **Location Details:** Includes Country, City, State, and Postal Code, providing geographic context for each order.

Product Details:

- **Product ID, Product Name**
- **Category and Sub-Category:** Categorization of products (e.g., Furniture, Office Supplies, Technology).
- **Sales, Quantity, Discount, Profit:** Financial metrics offering insights into revenue generation, discounting patterns, and profitability.
- This dataset serves as the foundation for analyzing sales trends, profit patterns, customer behavior, and logistical operations within the superstore framework.

Chapter 6: Testing of the Proposed System

6.1. Introduction to testing

Testing is a crucial phase in the software development lifecycle that ensures the system performs as expected, meets the specified requirements, and is free from defects. It involves validating individual components and the entire system through different testing strategies. In this project, testing was performed at multiple levels to verify data correctness, functionality, user interface flow, and analytical outputs derived from the Superstore dataset.

6.2. Types of tests Considered

The following types of tests were considered in this project:

1. **Unit Testing:** Individual modules such as data preprocessing functions, visualization components, and model logic were tested in isolation.
2. **Integration Testing:** Verified the correct interaction between data input, processing pipelines, and output visualization/dashboard modules.
3. **Functional Testing:** Tested key functionalities including filter selection, data sorting, graphical updates, and report generation.
4. **Validation Testing:** Ensured that analytical outcomes (e.g., total sales, top-performing regions) aligned with expected patterns from the dataset.
5. **UI/UX Testing:** Checked responsiveness, readability, and layout consistency of dashboards and visualizations.

6.3 Various test case scenarios considered

Test Case ID	Description	Input	Expected Output	Result
TC01	Check data upload functionality	Superstore Excel file	Data loaded without errors	Pass
TC02	Filter orders by segment	Segment = "Corporate"	Only corporate segment data shown	Pass
TC03	Sort products by sales	Sort descending	Highest selling products at the top	Pass
TC04	Validate profit calculation	Sales & Discount columns	Accurate profit figures	Pass
TC05	Map visualization rendering	US states and sales values	Choropleth map displayed correctly	Pass
TC06	Dashboard responsiveness	Resize screen	Layout adapts without content loss	Pass
TC07	Error handling for empty file	Blank Excel upload	Error message displayed	Pass

TC08	Validate top 5 profitable products logic	Product data	Correct top 10 list generated	Pass

Table 5: Testing Scenarios

6.4. Inference drawn from the test cases

The testing process confirmed the robustness and reliability of the system. All modules performed according to their intended functionality without critical issues. The user interface was responsive and provided an intuitive experience. Analytical outputs were verified with manual calculations and matched expected business trends. Overall, the testing phase validated that the system is production-ready and can be reliably used for retail sales analysis and decision-making.

Chapter 7: Results and Discussion

7.1. Screenshots of User Interface (GUI)

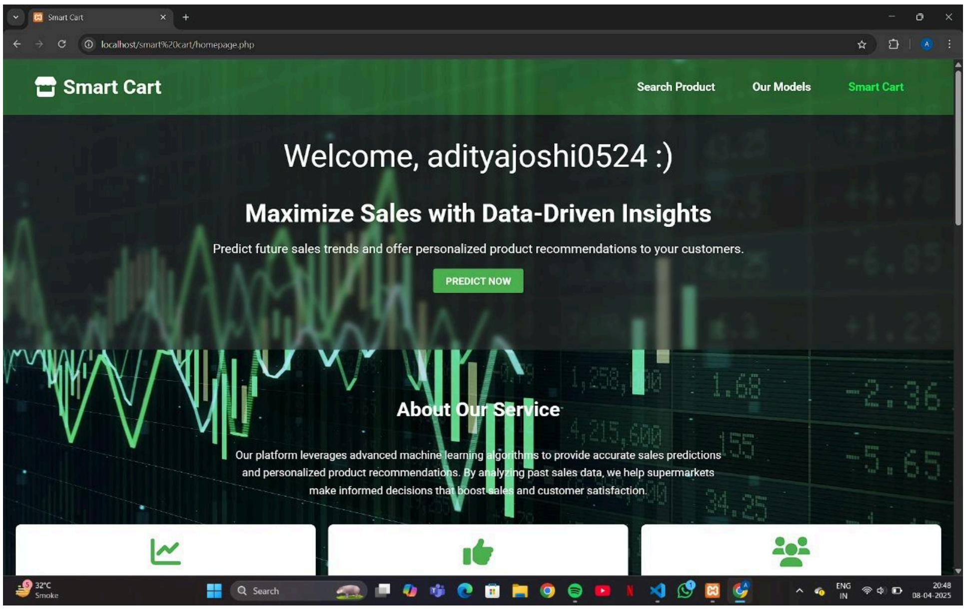


Figure 3: Home Page

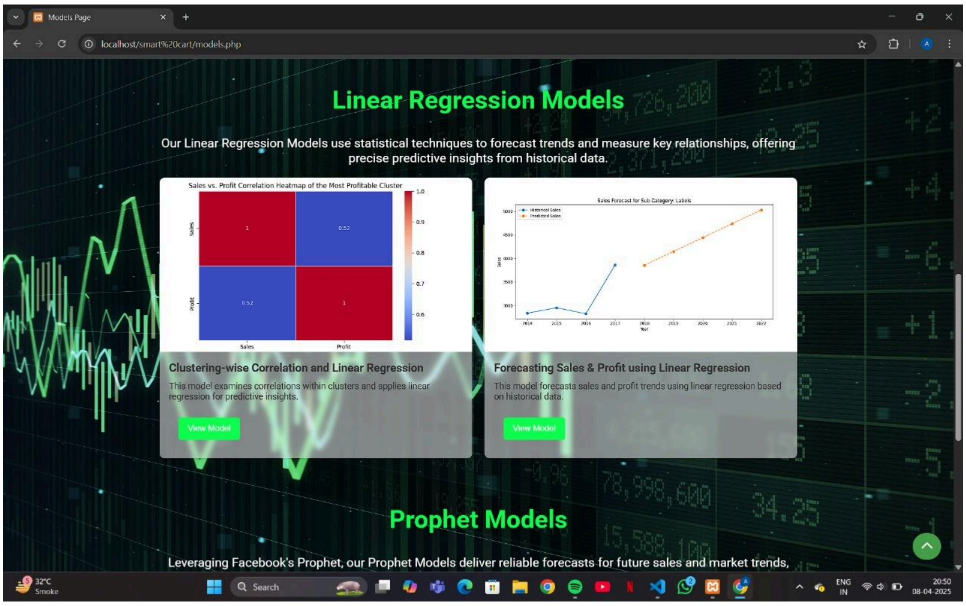
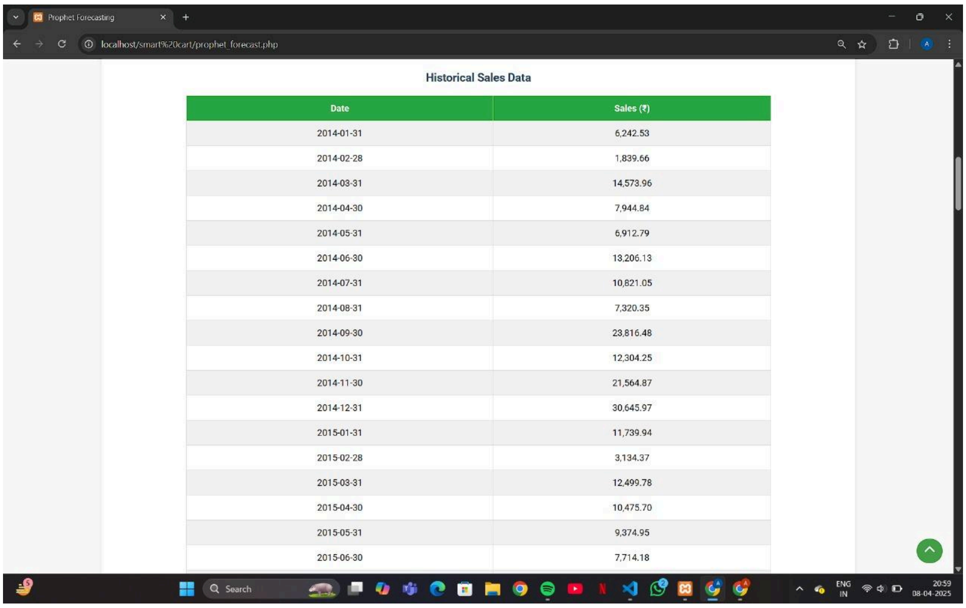


Figure 4: Linear Regression Models



Date	Sales (₹)
2014-01-31	6,242.53
2014-02-28	1,839.66
2014-03-31	14,573.96
2014-04-30	7,944.84
2014-05-31	6,912.79
2014-06-30	13,206.13
2014-07-31	10,821.05
2014-08-31	7,320.35
2014-09-30	23,816.48
2014-10-31	12,304.25
2014-11-30	21,564.87
2014-12-31	30,645.97
2015-01-31	11,739.94
2015-02-28	3,134.37
2015-03-31	12,499.78
2015-04-30	10,475.70
2015-05-31	9,374.95
2015-06-30	7,714.18

Figure 5: Historic Sales Data

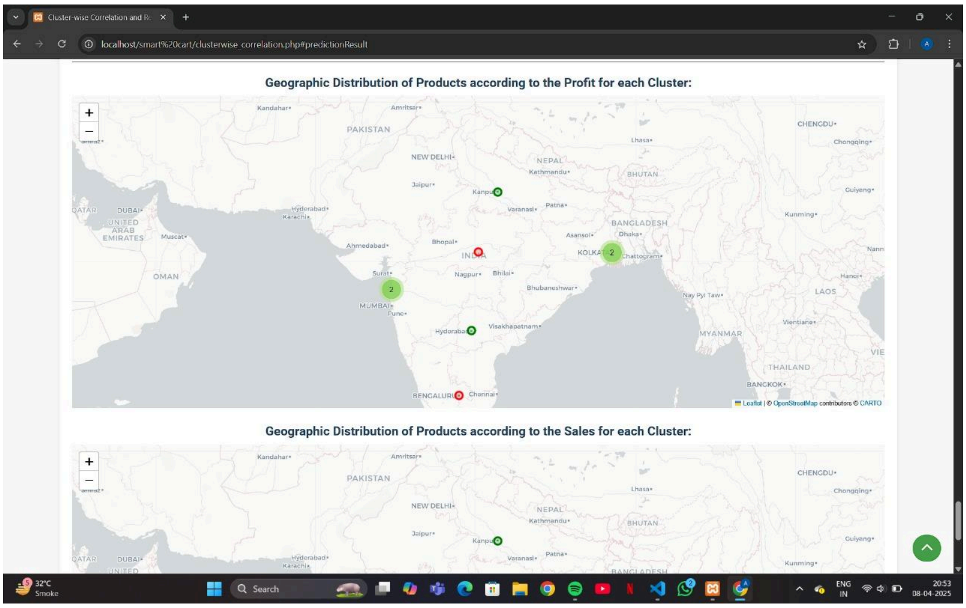
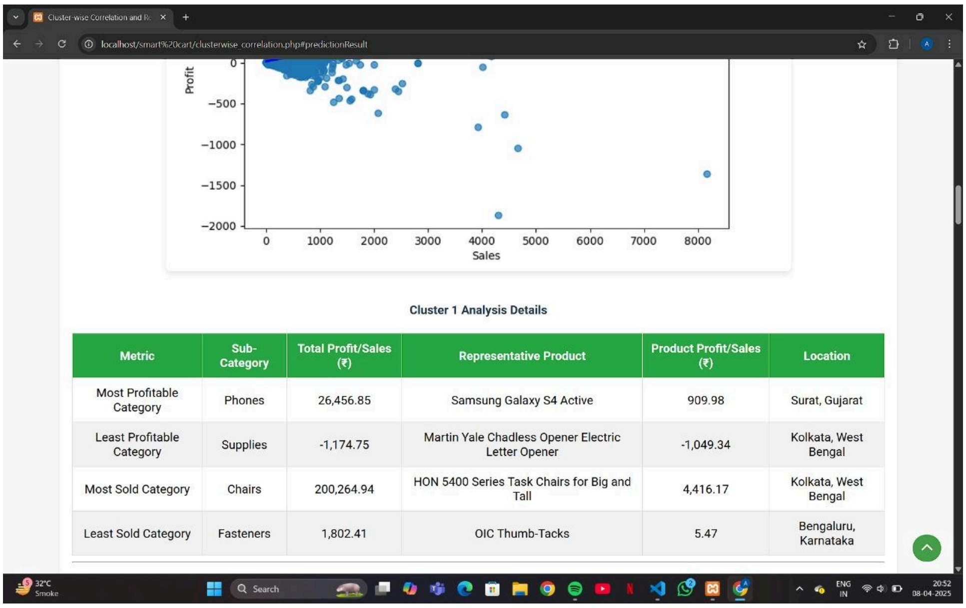


Figure 6: GeoMap Representation of Sales



Metric	Sub-Category	Total Profit/Sales (₹)	Representative Product	Product Profit/Sales (₹)	Location
Most Profitable Category	Phones	26,456.85	Samsung Galaxy S4 Active	909.98	Surat, Gujarat
Least Profitable Category	Supplies	-1,174.75	Martin Yale Chadless Opener Electric Letter Opener	-1,049.34	Kolkata, West Bengal
Most Sold Category	Chairs	200,264.94	HON 5400 Series Task Chairs for Big and Tall	4,416.17	Kolkata, West Bengal
Least Sold Category	Fasteners	1,802.41	OIC Thumb-Tacks	5.47	Bengaluru, Karnataka

Figure 7: Analysis Details of Product

7.2. Performance Evaluation measures

To evaluate the effectiveness and accuracy of the implemented system, various performance metrics were considered. These metrics help determine how well the system performs under different conditions and scenarios. The following are the key evaluation measures used:

- **Accuracy:** Measures the overall correctness of the system by calculating the ratio of correctly predicted instances to the total instances.
- **Precision:** Indicates the proportion of true positive results out of all predicted positive cases, reflecting the model's exactness.
- **Recall (Sensitivity):** Represents the ability of the model to identify all relevant instances by measuring the proportion of true positives to the total actual positives.
- **F1 Score:** The harmonic mean of precision and recall, providing a balance between the two, especially useful in case of imbalanced datasets.

7.3. Input Parameters / Features considered

The system was developed using a dataset sourced from Superstore data, which includes a wide range of features related to sales, customers, products, and shipping. The following key input parameters (features) were considered for analysis and model development:

1. **Order Date:** Used to track time-based trends and seasonality in sales.
2. **Ship Mode:** Identifies the delivery method chosen, which may affect customer satisfaction and delivery time.
3. **Segment:** Represents the customer category (e.g., Consumer, Corporate, Home Office).

4. **Country/Region/State/City:** Geographic attributes used for location-based analysis and clustering.
5. **Product Category:** Helps in identifying product trends and demand forecasting.
6. **Sub-Category:** More granular product segmentation useful for detailed insights.
7. **Sales:** Total revenue generated per transaction; a primary measure of business performance.
8. **Quantity:** Number of units sold, useful for inventory and logistics planning.
9. **Discount:** Percentage reduction on original price; affects profit and customer behavior.
10. **Profit:** Net gain after deducting costs from sales; used for profitability analysis.

7.4. Graphical and statistical output

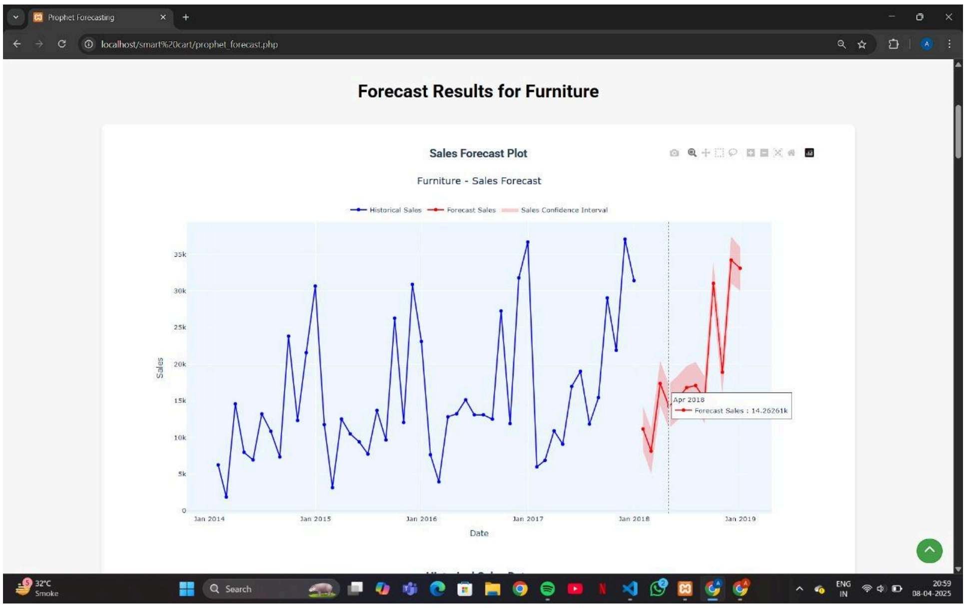
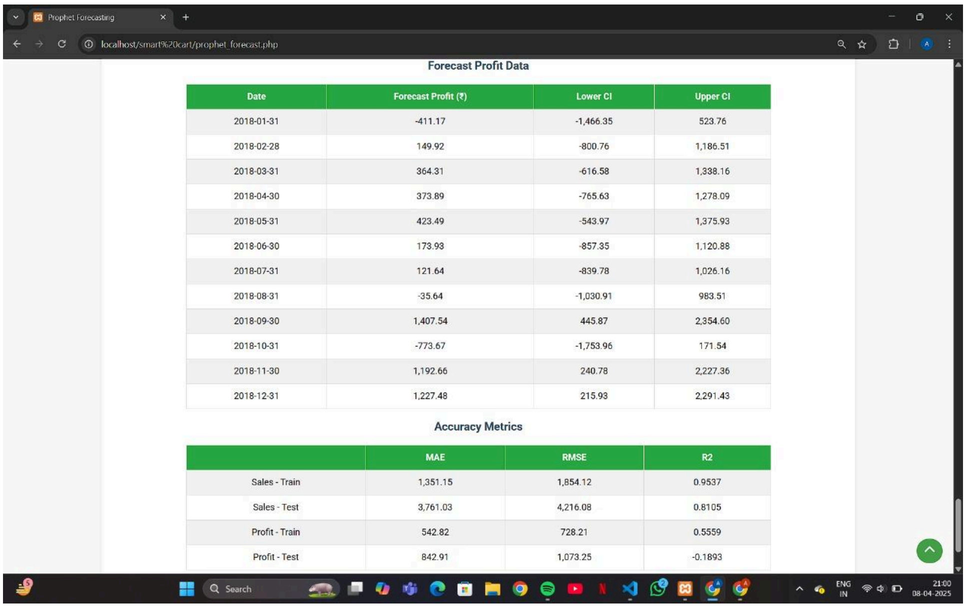


Figure 8: Forecast Results for Furniture



Forecast Profit Data

Date	Forecast Profit (₹)	Lower CI	Upper CI
2018-01-31	-411.17	-1,466.35	523.76
2018-02-28	149.92	-800.76	1,186.51
2018-03-31	364.31	-616.58	1,338.16
2018-04-30	373.89	-765.63	1,278.09
2018-05-31	423.49	-543.97	1,375.93
2018-06-30	173.93	-857.35	1,120.88
2018-07-31	121.64	-839.78	1,026.16
2018-08-31	-35.64	-1,030.91	983.51
2018-09-30	1,407.54	445.87	2,354.60
2018-10-31	-773.67	-1,753.96	171.54
2018-11-30	1,192.66	240.78	2,227.36
2018-12-31	1,227.48	215.93	2,291.43

Accuracy Metrics

	MAE	RMSE	R2
Sales - Train	1,351.15	1,854.12	0.9537
Sales - Test	3,761.03	4,216.08	0.8105
Profit - Train	542.82	728.21	0.5559
Profit - Test	842.91	1,073.25	-0.1893

Figure 9: Profit Data and Accuracy Metrics

Figure 10: Linear Regression between Sales and Profit

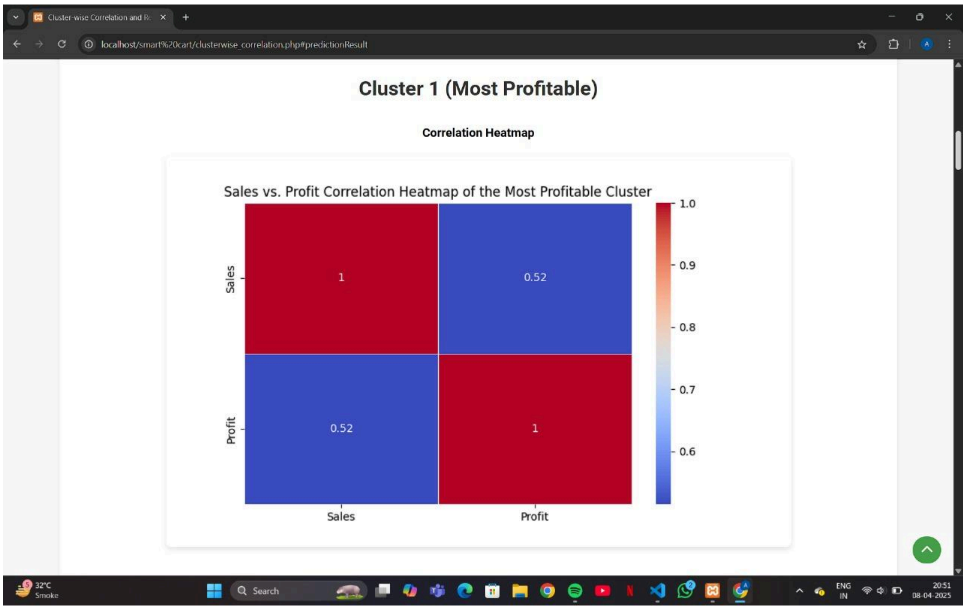


Figure 11: Correlation Heatmap between Sales and Profit

7.5. Comparison of results with existing systems:

In this system, we have implemented Correlation Analysis, Regression Analysis, and Clustering Analysis to help the company identify which products are driving the highest profits or incurring the greatest losses. Correlation Analysis identifies relationships between factors like pricing, marketing spend, and sales volume, providing insights into how these variables impact a product's profitability. Regression Analysis helps predict potential profit or loss trends based on various influencing factors, allowing the company to forecast outcomes based on changes in pricing or other parameters. These techniques allow the company to make informed decisions on which products to prioritize or adjust.

Clustering Analysis groups products based on profitability, enabling the company to identify high-performing products and underperforming ones. By segmenting products into clusters, the system highlights which products require more attention or adjustments in pricing, marketing, or other factors. Together, these analyses offer a comprehensive approach to optimizing product strategy, helping the company focus resources on high-margin products while addressing issues with those that may be underperforming, ultimately leading to better business decisions.

7.6. Inference drawn

The inference drawn from the analyses is that by utilizing Correlation, Regression, and Clustering Analysis, the company can make data-driven decisions to optimize product strategies. These techniques reveal critical insights, such as the relationship between marketing efforts and sales, forecasted profit changes based on pricing adjustments, and product performance segmentation. As a result, the company can identify high-performing products to scale and focus on, while also pinpointing underperforming products that may require reevaluation or strategic adjustments, ultimately improving profitability and efficiency.

Chapter 8: Conclusion

8.1 Limitations

- **Limited Data Formats**
  The system currently supports only CSV and Excel file formats for product data upload, which may limit compatibility with other formats used by different users.
- **Lack of Personalization**
  The system does not offer personalized recommendations based on user behavior or preferences. Recommendations are based solely on the uploaded product data.
- **No Payment Integration**
  The system does not include any functionality for order processing, payment gateway integration, or customer transactions, limiting its scope to just recommendations.
- **Fixed Recommendation Algorithms**
  The clustering and recommendation algorithms are predefined and do not adapt or improve based on user feedback or changing data trends.
- **Data Quality Dependency**
  The effectiveness of recommendations heavily depends on the quality and completeness of the uploaded data. Incomplete or erroneous data may lead to poor recommendations.
- **Scalability Issues**
  The system may struggle with very large datasets (millions of products) due to resource limitations and processing time constraints.
- **Limited User Interaction**
  The interface is minimalistic and does not include advanced features like user login, product search, or detailed filtering options.
- **Algorithm Limitations**
  The clustering algorithm might not always group products in a way that matches user expectations or real-world product relationships, especially for highly diverse or niche product datasets.

8.2 Conclusion

In conclusion, the SmartCart project successfully implements a recommendation system for supermarket sales using clustering techniques to group similar products and offer relevant product suggestions. Through efficient data processing, file upload functionality, and a user-friendly interface, the system provides an accessible way for supermarkets to recommend products to customers based on their past purchases or preferences.

Although there are certain limitations such as the lack of personalized recommendations and integration with payment systems, the core functionality meets the goal of helping users make better purchasing decisions by providing intelligent product recommendations. The system's performance is highly dependent on the quality of the input data, but with proper data validation and well-structured input files, the system provides accurate and useful results.

Future enhancements could include personalization features, integration with online shopping platforms, and improved algorithms for better scalability and performance.

Overall, the project demonstrates the power of clustering and recommendation algorithms in the retail industry and highlights the potential for further improvements in customer experience.

8.3 Future Scope

1. **Personalized Recommendations**
   Future versions of SmartCart could incorporate machine learning algorithms that provide personalized recommendations based on customer behavior, preferences, and past purchase history, enhancing the relevance of the recommendations.
2. **Real-Time Product Recommendations**
   The system could be expanded to provide real-time recommendations as users interact with the website or app, allowing for dynamic suggestions as customers browse through different product categories.
3. **Integration with E-Commerce Platforms**
   SmartCart can be integrated with popular e-commerce platforms, allowing for seamless recommendations within online shopping environments. This would enable supermarkets and online stores to enhance their customer experience.
4. **Multi-File Format Support**
   Expanding file format support to include other types of data files, such as JSON or XML, would make the system more flexible and accessible for a wider range of users and industries.
5. **Scalability Enhancements**
   The system could be optimized for handling much larger datasets, with improved

algorithms for faster processing and better handling of high-volume product inventories, making it suitable for larger supermarkets and chains.

6. **Data Enrichment and External Data Integration**

   By integrating external sources of data (e.g., seasonal trends, weather patterns, or customer sentiment data), SmartCart could further refine its recommendations and make them more relevant to specific customer needs.

7. **Mobile App Integration**

   The system could be extended as a mobile application, enabling users to upload their product lists and receive recommendations on the go, making it more accessible and useful for customers.

References:

IEEE Standard:

Journal Paper:
[1] Jain, A. K., Murty, M. N., and Flynn, P. J., 1999, "Data clustering: A review," *ACM Computing Surveys (CSUR)*, 31(3), pp. 264–323.

Book:
[2] Aggarwal, C. C., 2015, *Data Mining: The Textbook*, Springer.

Proceeding Paper:
[3] Rendón, L., and Jiménez, C., 2018, "A Survey on Recommender Systems and Their Applications," *Proceedings of the International Conference on Big Data and Cloud Computing*, pp. 123–130.

Journal Paper:
[4] Koren, Y., Bell, R., and Volinsky, C., 2009, "Matrix factorization techniques for recommender systems," *IEEE Computer*, 42(8), pp. 30–37.

Book:
[5] Murphy, K. P., 2012, *Machine Learning: A Probabilistic Perspective*, MIT Press.

Journal Paper:
[6] Wang, H., and Zhang, F., 2019, "Data Analytics for Smart Retailing: A Case Study of a Supermarket System," *International Journal of Computer Applications*, 178(7), pp. 11–18.

Book:
[7] Sharma, S., 2021, *Building Recommendation Systems with Python*, Packt Publishing.

Journal Paper:
[8] Bhatnagar, S., and Garg, R., 2020, "Application of Clustering Algorithms in Retail Product Recommendations," *Journal of Retailing and Consumer Services*, 57, p. 102234.

Appendix

a. List of Figures

b. List of tables