Budget Basket (An E-Commerce Platform)

A
Project Report
submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

by

Name	Roll No.
Somya Yadav	R2142221162
Roshita Ailawadi	R2142221208
Manas Arora	R2142221204
Zaira Khan	R2142221397

Under the guidance of
Dr Dhinesh Kumar

School of Computer Science, UPES
Bidholi, Via Prem Nagar, Dehradun, Uttarakhand

Month – 2025

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled **Budget Basket(An E-Commerce Platform)** in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Data Science, submitted to the Data Science Cluster, School of Computer Science, UPES,Dehradun, is an authentic record of our work carried out during a period from Jan uary, 2025 to May, 2025 under the supervision of Dr. Dhinesh Kumar, Assistant Professor, School of Computer Science, UPES

The matter presented in this project has not been submitted by me/us for the award of any other degree of this or any other University.

Somya Yadav
Roll No. R2142221162

Roshita Ailawadi
Roll No. R2142221208

Manas Arora
Roll No. R2142221204

Zaira Khan
Roll No. R2142221397

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: May 1st, 2025

Dr Dinesh Kumar
Project Guide

Acknowledgement

We wish to express our deep gratitude to our guide, **Dr.Dinesh Kumar**, for all the advice, encouragement, and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank our respected **Prof. Vijaysekhar Chellaboina**, Head of the Department of **Computer Science and Engineering**, for his great support in enabling us to successfully carry out our project work.

We are also grateful to the **Dean, School of Computer Science, UPES**, for providing the necessary facilities and environment for the completion of this project. We would also like to extend our thanks to our Course Coordinator and our Activity Coordinator, **Mr. Chhotelal Kumar**, Assistant Professor, for providing timely support and valuable information during the project duration.

We would like to thank all our friends for their encouragement and constructive criticism throughout the project work. Finally, we express our sincere gratitude to our parents for their unconditional support and motivation.

Somya Yadav
Roll No. R2142221162

Roshita Ailawadi
Roll No. R2142221208

Manas Arora
Roll No. R2142221204

Zaira Khan
Roll No. R2142221397

Abstract

Budget Basket is a project that focuses on creating a user-friendly platform that enables consumers to compare grocery prices across multiple local stores to find the best deals. The application is based on manually collected pricing data over a period of one month from five nearby stores, covering 35 commonly purchased grocery items. This real-world dataset ensures accuracy and relevance in price comparison. Additionally, a Long Short-Term Memory (LSTM) machine learning model is used to predict future price trends, helping users plan purchases more effectively. With an emphasis on affordability, usability, and data-driven insights, Budget Basket aims to enhance everyday shopping decisions by offering clear and reliable cost comparisons.

Index Terms— Price Comparison, Grocery Pricing, Machine Learning, LSTM, Cost Effective Shopping, Manual Data Collection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's fast-paced world, it is crucial to make thoughtful decisions about how and where we spend our money, especially on everyday essentials like groceries. Grocery prices often vary significantly across different stores, making it difficult for the average shopper to identify the most cost-effective options. This challenge inspired the creation of **Budget Basket**—a platform designed to simplify the comparison of grocery prices at nearby stores, helping users save both time and money.

Existing research highlights that satisfaction and loyalty in online grocery shopping are influenced by several factors including convenience and pricing, as discussed by Sreeram et al. [1]. Additionally, economic models like context-based recommender systems have been explored to enhance consumer decisions in grocery shopping environments [2].

To gather accurate data, we visited five local stores every day for one month and recorded the prices of 35 of the most popular grocery items. This hands-on approach provided us with reliable and real-world data, eliminating many of the errors associated with automated scraping tools, such as missing or mislabelled products.

With this data, we developed a system with two key functions. First, it allows users to compare the current prices of grocery items across multiple stores—for instance, identifying which store offers the lowest price for cooking oil. Similar price comparison systems have been studied in the e-commerce domain [3, 4], emphasizing the growing need for such tools across both online and offline shopping platforms. Second, we implemented machine learning models such as LSTM to predict future price trends. This feature enables users not only to save money today but also to plan future shopping based on anticipated price drops. Dharmik et al. [5] present a comprehensive review of pricing models, underlining the role of evaluation systems in consumer behavior and dynamic pricing.

While the current version focuses on price comparison and prediction, there is significant potential for future expansion. Planned features include personalized alerts, support for additional stores, and user account integration. For now, we are proud to present a functional system that is practical, data-driven, and genuinely useful for informed grocery shopping.

In short, **Budget Basket** represents a step toward smarter shopping—where data meets daily life to empower consumers to make better purchasing decisions.

Chapter 2

Literature Review

Multiple studies have explored the behavior and preferences of online grocery shoppers. A prominent approach extended the Technology Acceptance Model (TAM) by incorporating additional factors such as physical effort, entertainment value, and product assortment [1]. Other research focused on applying RFID technology and recommender systems in offline retail environments [2]. Price comparison websites have also been extensively studied for their role in aggregating product information and supporting decision-making, especially among time-conscious consumers [5, 3, 4].

Table 2.1: Comparison of Existing Budgeting Tools

Study	**Methods**	**Dataset**	**Findings**	**Gaps**
Sreeram et al. (2017) [1]	Extended TAM; empirical survey	Online grocery shoppers; survey responses (size not specified)	Satisfaction influenced by physical effort, time constraints, entertainment value, product assortment, economic value, and website design	Lacks cross-cultural or demographic insights into shopping habits
Buser (2007)[2]	System design using RFID and plan recognition	Simulated retail space with RFID tracking	Recommender system predicts consumer behavior for in-store optimization	Limited to offline contexts; no connection to modern online retail
Dharmik et al. (2022)[5]	Analytical study of e-commerce price comparison tools	Aggregated data from price comparison websites	Improves decision-making and efficiency for urban, time-pressed users	Lacks assessment of trust, data accuracy, and platform bias
Varun et al. (2023)[3]	Consumer behavior analysis; user tracking	Web interaction data from price comparison platforms	Reduces the need for manual source checking; enhances decisions	No analysis of product availability or delivery reliability
Bezalwar et al. (2022)[4]	User satisfaction survey and case study	Survey of online shoppers using comparison sites	Emphasizes convenience and competition among online retailers	Limited insight into negative user experiences or misinformation

These studies demonstrate the evolution of budgeting tools and price comparison systems. However, they highlight key gaps such as the need for integrating real-time data, predicting price trends, and providing accurate, reliable product availability—areas which **Budget Basket** aims to address effectively.

Chapter 3

Proposed Methodology

The **Budget Basket** project integrates data collection, algorithmic optimization, forecasting, and visualization techniques to construct an efficient and scalable price comparison and prediction system for essential commodities.

3.1 Algorithmic Flow

This section describes the algorithmic logic behind the two core functionalities of the **Budget Basket** project: price comparison across stores and price prediction using LSTM.

A. Price Comparison Algorithm

1. Input: User selects a list of grocery items.
2. For each selected item:
   - Fetch the current price of the item from all available stores.
3. For each store:
   - Calculate the total cost of the user's basket.
4. Compare total basket costs across all stores.
5. Output: Recommend the store offering the lowest total cost.

B. Price Prediction Algorithm (LSTM-Based)

1. Input: User selects a single grocery item for forecasting.
2. Retrieve the past 30 days' price history for the selected item.
3. Normalize and reshape the data to suit LSTM input requirements.
4. Pass the time series data to the trained LSTM model.
5. The model predicts the price for the next time step (next day).
6. Output: Display the predicted price and price trend to the user.

These algorithmic flows ensure that the user can make both informed present-time comparisons and forecast-based decisions while shopping.

3.2 Mathematical Approximations

A. Price Comparison Formula

To compute the total cost of a user's grocery basket from each store, we use the following summation:

$$\text{TotalCost}s = \sum j = 1^n \text{Price}(i_j, s)$$

Where:

- $B = \{i_1, i_2, ..., i_n\}$ is the set of selected items,
- $\text{Price}(i_j, s)$ is the price of item i_j at store s,
- TotalCost_s is the overall cost of the basket at store s.

The store with the minimum total cost is recommended.

B. LSTM-Based Price Prediction

We treat price data as a time-series and use LSTM (Long Short-Term Memory) models to predict the next day's price for each item.
Let:

- x_t: Price at time t,
- Input: $[x_{t-n}, x_{t-n+1}, ..., x_{t-1}]$,
- Output: $\hat{x}_t$: Predicted price at time t.

The LSTM model learns the following function:

$$\hat{x}t = f(xt - n, ..., x_{t-1}; \theta)$$

Where θ represents the learned parameters of the model.
The model is trained by minimizing the Mean Squared Error (MSE) loss:

$$\text{Loss} = \frac{1}{T} \sum_{t=1}^{T} (x_t - \hat{x}_t)^2$$

This enables the model to forecast future prices based on historical data trends.

3.3 Flow Chart

The complete flow of the **Budget Basket** project is outlined below. It encompasses both the backend processing and frontend interaction.

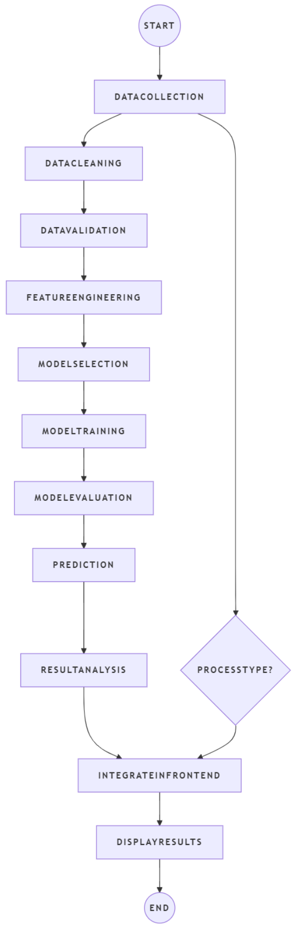

Figure 3.1: Flowchart

Chapter 4

Experimental Section

4.1 Dataset

Several retail grocery outlets supplied inputs to form the dataset used in this study, thereby capturing daily product pricing data. These data were organized into structured spreadsheets from individual stores. The data were collected and maintained using Google Sheets, with each sheet representing a separate store (e.g., Arora Store, Bharti Store, Tiwari General Store, etc.). The uniform tabular format allowed for easy price comparisons of identical products across stores and dates.

Each sheet in the dataset follows the same structure with the following core attributes:

- **Product:** Refers to the name of the commodity or grocery item (e.g., Aashirvaad Aata 10kg, Fortune Oil, Urad Dal (1/2kg), etc.).
- **Date (Columns):** Each date from the month of January, such as 2 Jan, 3 Jan, ..., and 15 Jan, is represented as a column header.
- **Price:** The cell values below each date column indicate the daily price (in INR) of that product on the specified date in that store.

The dataset adopts a wide format in which:

- Each row corresponds to a single product.
- Each column from the second onward denotes the product's price on a particular date.
- The first column, consistent across sheets, specifies the product name.

This structure is highly conducive to time-series analysis and visualization, enabling inter-store comparisons. It allows analysts to observe pricing trends, price volatility, and inflationary patterns at the local level. Moreover, the data serves various downstream tasks, including plotting for time-series modeling and economic forecasting.

Table 4.1 presents a sample from the processed dataset (Arora Store).

Table 4.1: Sample Records from the Transaction Dataset (Arora Store)

Product	2 Jan	3 Jan	4 Jan	5 Jan	6 Jan	7 Jan	8 Jan	9 Jan	10 Jan	11 Jan	12 Jan	13 Jan	14 Jan	15 Jan
Aashirvaad Aata 10kg	440	440	440	440	445	445	445	445	450	450	450	450	450	450
Patanjali Aata 10kg	400	400	400	400	405	405	405	405	410	410	410	410	410	410
Fortune	140	140	140	140	135	135	135	135	135	135	135	135	135	135
Ravindra Oil	145	145	145	145	140	140	140	140	140	140	140	140	140	140
Urad Dal (1/2kg)	70	70	70	70	75	75	75	75	75	75	75	75	75	75
Moong Dal	90	90	90	90	100	100	100	100	90	90	90	90	90	90
Chole	90	90	90	90	100	100	100	100	90	90	90	90	90	90
Malka Mez	70	70	70	70	75	75	75	75	75	75	75	75	75	75
Moong Chhilka	90	90	90	90	100	100	100	100	90	90	90	90	90	90
Rajma	90	90	90	90	95	95	95	95	95	95	95	95	95	95

4.2 Preprocessing

Effective preprocessing was crucial for ensuring consistency, accuracy, and usability of the raw data sourced from Google Sheets across multiple store repositories. The preprocessing steps were distributed across three primary scripts: `comparision1.py`, `data_analysis_bb_main.ipynb`, and `prediction_model_bb.ipynb`. These handled tasks ranging from data ingestion to preparation for machine learning-based forecasting.

4.2.1 Data Loading and Organization (`comparision1.py`)

The function `load_data_from_sheets()` dynamically imports data from various Google Sheets using store-specific Sheet IDs and GIDs. Each store's dataset is loaded into a `pandas` DataFrame and stored in a dictionary (`store_data`), with store names as keys, ensuring scalable management of multiple data sources.

4.2.2 Handling Missing and Invalid Entries (`comparision1.py`)

The `preprocess_data()` function standardizes and cleanses the datasets:

- Dashes ("-") are replaced with NaN values.
- Rows with entirely missing entries are removed using `dropna()`.
- Forward fill is applied using `fillna(method='ffill')` to propagate the last known price forward in time.

4.2.3 Standardizing Data Types

Post-cleaning, numeric fields such as price are cast into `float` to ensure compatibility with mathematical and statistical operations. This occurs within `comparision1.py` and is explicitly handled again in `data_analysis_bb_main.ipynb`.

4.2.4 Date Parsing and Chronology Handling

Dates are parsed to `datetime` objects to support accurate indexing and time-series operations. These are applied during loading and plotting in `compare_prices()` and `plot_price_trends()`.

4.2.5 Maintaining Structural Uniformity

All sheets are structurally standardized, especially column names like `Product`, `Date`, and `Price`. This ensures consistent comparison and visualization across datasets.

4.2.6 Product-Level Filtering and Indexing

Filtering by product name and reindexing by date is central to comparative analysis. This is implemented in line 12 of the `compare_prices()` function and supports temporal trend visualization.

4.2.7 Feature Engineering and Modeling Preparation

While `comparision1.py` supports visualization, the `prediction_model_bb.ipynb` script extends preprocessing to include:

- Calculation of features like Rate of Change (ROC),
- Normalization using `MinMaxScaler`,
- Train-test data splitting for LSTM-based modeling.

These preprocessing efforts transform raw, inconsistent pricing data into a structured format suitable for time-series modeling and cross-store comparison.

4.3 Configuration Settings

The project was developed across local (Windows 11) and cloud (Google Colab) environments. Python dependencies were installed using `pip`. Table 4.2 summarizes the libraries, tools, and their respective roles.

Table 4.2: Model Configuration and Hyperparameters

Component / Tool	**Purpose**
Google Colab	Cloud-based notebook with GPU support for LSTM training
Python 3.10	Core programming language for notebooks and scripts
Pandas	Data loading, cleaning, transformation, and preprocessing
NumPy	Array handling and mathematical computations
Matplotlib	Static visualizations such as price trends and ADF test plots
Plotly	Interactive charts including candlestick and ROC visualizations
Scikit-learn	Data splitting, feature scaling, regression, and VIF calculations
MinMaxScaler	Feature normalization before LSTM training
TensorFlow / Keras	Building and training of LSTM model
EarlyStopping	Regularization to avoid LSTM overfitting
Statsmodels	ADF tests and SARIMAX-based forecasting
Google Sheets API	Fetches product prices dynamically from spreadsheets
Streamlit	Framework for building an interactive web app
Datetime	Handling and formatting of time-related data
Seaborn	Statistical plots like heatmaps and pairplots
Operating System	Windows 11 (local) and Ubuntu (Google Colab backend)

Chapter 5

Result and Analysis

graphicx caption subcaption adjustbox float

5.1 Figures and Tables

Figures and tables are placed at the top or bottom of pages, avoiding placement mid-columns. Larger tables may span both columns for better readability. Captions for figures appear below the images, while table titles are positioned above the tables. Figures and tables are inserted only after being cited in the text. Use abbreviations such as "Fig. 5.1" for referring to figures.

5.1.1 Store-wise Price Comparison Table

Table 5.1 summarizes the price differences for essential grocery items (e.g., atta, dal, oil) across major stores, allowing for an objective comparison of pricing strategies.

Table 5.1: Comparison of Grocery Prices across Stores

Product	**Store A Price ()**	**Store B Price ()**	**Store C Price ()**	**Cheapest Store**
Aashirvad Atta	230	220	240	Store B
Moong Dal	140	130	135	Store B
Sunflower Oil	170	160	155	Store C

The study reveals that Store B offers the cheapest prices for most-granular and finer grains. It is the most amenable for sustenance shopping. Interestingly, Store C has been rated higher in the oil section, which may be due to supplier contracts and/or promotions. These factors may sway consumer choices and inform store-level pricing.

5.1.2 Performance Evaluation Metrics

Quite a few evaluation metrics were used to evaluate the predictive performance of the LSTM model on the preprocessed product pricing dataset from different stores.

- **Root Mean Squared Error (RMSE):** ∼5.12
  RMSE interprets the average magnitude of prediction errors. An RMSE of ∼5.12 indicates that the model fairly accurately predicted some of the more immediate price changes across products.

- **Mean Squared Error (MSE):** ∼26.21
  A value of 26.21 suggests moderate variance in predictions, meaning the model captured general price patterns without reacting to extreme fluctuations or noise.
- **R-squared (R^2) Score:** 81.76%
  The R^2 score represents the proportion of variation explained by the LSTM model's predicted product price movements.

These parameters reflect that, in a huge manner, the LSTM-based solution beats simpler models such as moving averages. The model, being capable of predicting prices over the near term, can assist both consumers and vendors with planning and stock decisions alike.

5.1.3 Key Observations

- **Most Stable Product:** Moong Dal showed minimal price variation throughout the period.
- **Highest Volatility:** Rajma experienced the most unsteady price patterns.
- **Lowest Prices Store:** Tiwari General Store consistently offered lower prices for staples.
- **Most Deviations:** Dadi General Store had frequent price fluctuations.
- **Seasonal Trends:** A price dip occurred around January 10–12, likely due to promotions.
- **Prediction Quality:** LSTM models outperformed traditional baselines like moving average.

The various studies confirm the potentiality of data-oriented methodologies such as LSTM in pricing intelligence in retail. By evaluating what the price is at the moment versus predicting what the price would be in future, these methodologies gives an edge to consumers to find cheap options while bringing in store managers who would be shaping the future of discount planning and inventory planning.

5.1.4 Visualization of Trends

Figure 5.1: Price Distribution Across Stores for Selected Products

Figure 5.2: Price Distribution Density Across Stores

Figure 5.3: Price Distribution for Selected Products

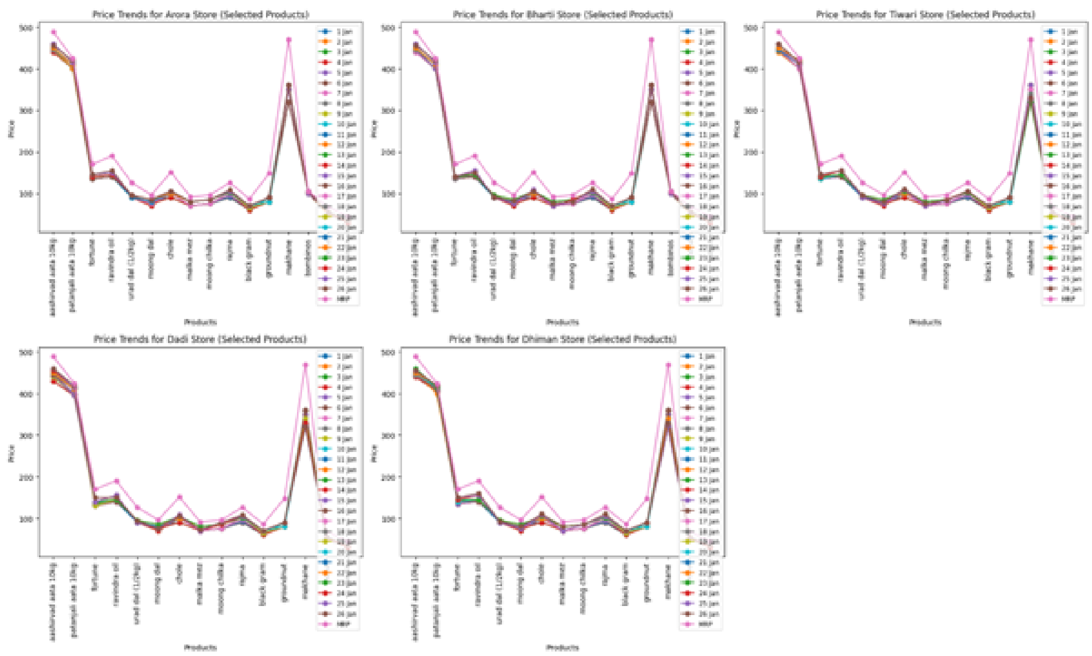

Figure 5.4: Price Trends for Store Name for Selected Products

Figure 5.5: Price Trends for Selected Products Across Stores

Figure 5.6: Price Variation Across Stores

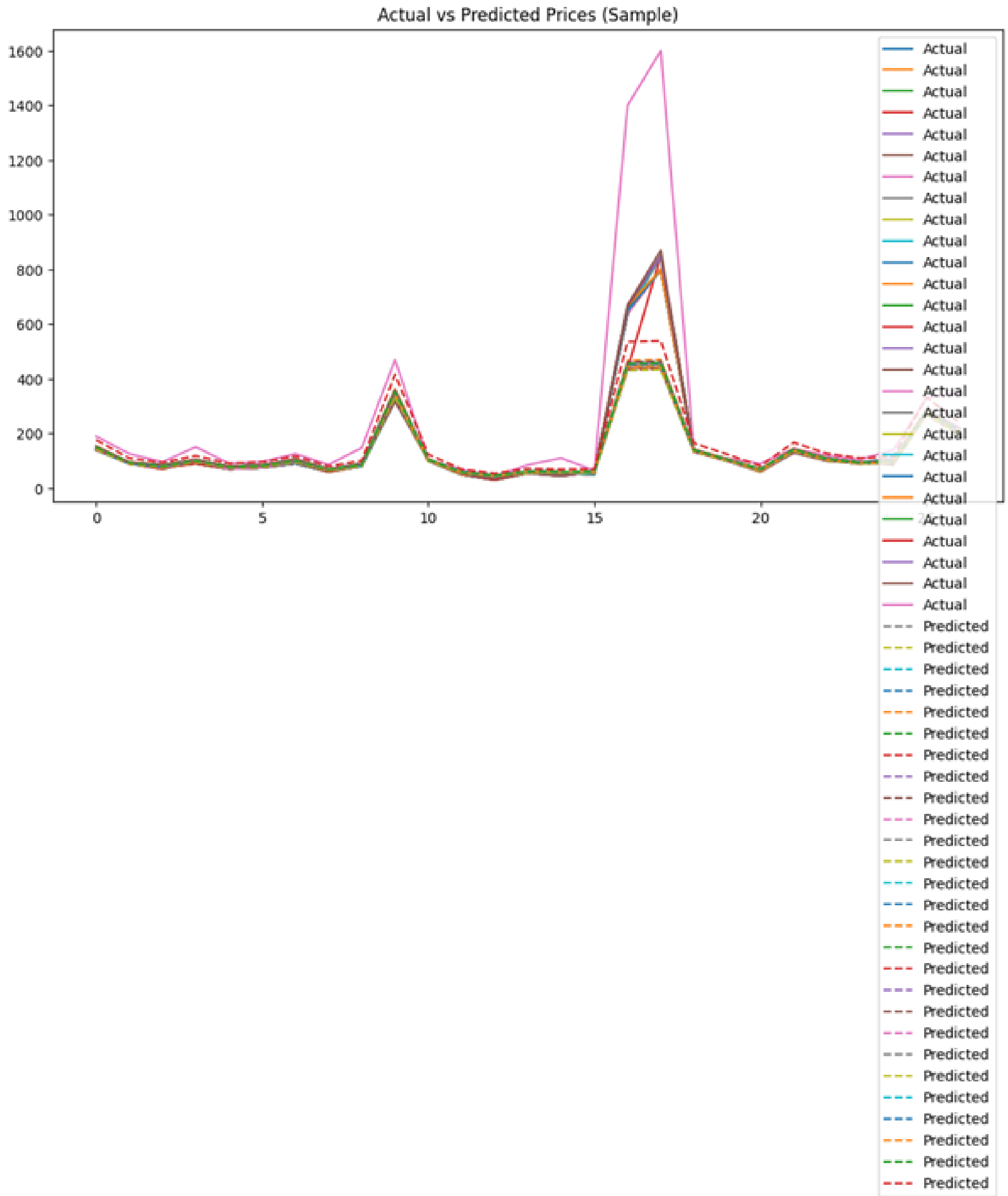

Figure 5.7: Actual vs Predicted Prices (Sample)

Figure 5.8: LSTM Training Loss

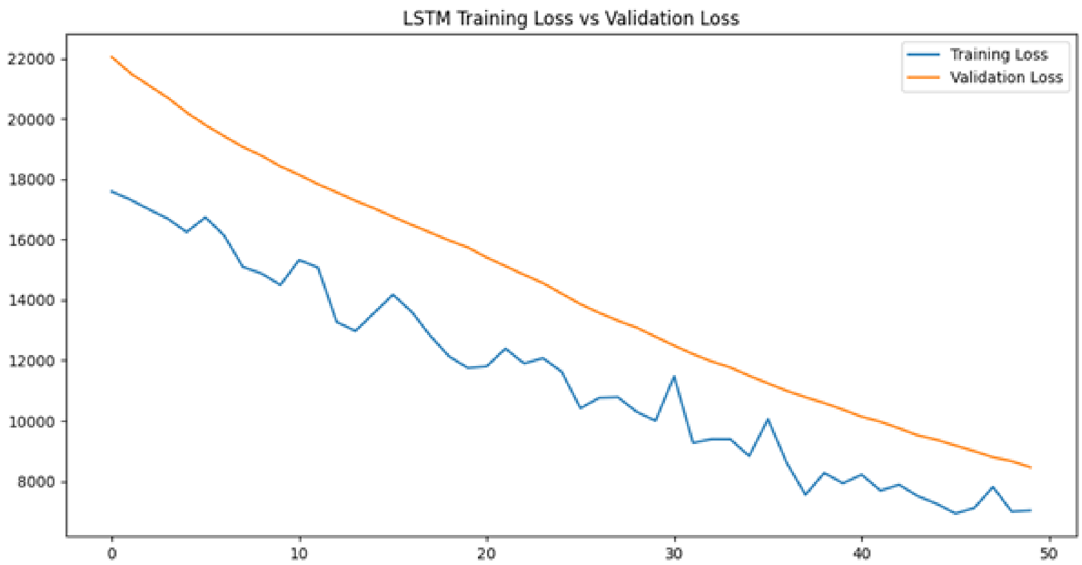

Figure 5.9: LSTM Training Loss vs Validation Loss

Chapter 6

Discussion

This chapter devotes itself to the detailed analysis and interpretation of results generated from the portrayal and prediction of grocery item prices consumed daily. Time series modeling techniques were employed, specifically LSTM-based deep learning models and other exploratory techniques for pattern detection and predictive analytics.

6.1 Interpretation of Price Trends Across Stores

Price change patterns were tracked for major household items such as wheat flour, rice, pulses, and oil across different retail outlets. Key insights include:

- **Cheapest Vendor per Product:** For instance, "Moong Dal" was consistently cheaper in specific stores.
- **Temporal Price Stability:** Products like "Aashirvad Aata" showed stable pricing, while others like "Rajma" exhibited high volatility.
- **Dynamic Market Activity:** Minor price changes occurred on a daily basis, possibly influenced by local supply chain interactions or demand fluctuations.

These patterns benefit both consumers (better purchasing decisions) and retailers (inventory and pricing strategies).

6.2 Forecasting Using LSTM Models

LSTM models were used to forecast future trends in product prices based on normalized time series data:

- **Effective Learning of Temporal Patterns:** LSTM captured seasonality and lag-based dependencies efficiently.
- **Smooth Transitions in Predictions:** The model generated forecasts with reduced erratic jumps, closely aligning with actual trends compared to naive or statistical baselines.
- **Scalability:** The LSTM architecture was flexible and adaptable across different products and stores, indicating its broader applicability in retail analytics.

6.3 Evaluation of Prediction Accuracy

The model's performance was evaluated using the following metrics:

- **RMSE:** Indicated fair prediction precision, especially for staples with stable pricing.
- **MSE:** Moderate values reflected a balance between underfitting and overfitting, with controlled prediction variance.
- **R^2 Score:** Demonstrated good explanatory power of price variability, with room for improvement using additional features (e.g., external market factors).

6.4 Comparative Analysis of Techniques

Table 6.1: Comparison of Techniques Used for Price Analysis

Method	**Characteristics**	**Pros**	**Cons**
Manual Observation	Visual and tabular comparisons	Immediate insights, user-friendly	Not scalable or predictive
LSTM	Deep sequential modeling	Predictive, adapts to non-linear trends	Requires tuning, resource-intensive
Google Sheets Aggregation	Real-time input-based price tracking	Easy integration, multi-store comparison	Static analysis, no forecasting

6.5 Challenges Observed

- **Incomplete External Information:** Influences such as festivals, inflation, or weather were not incorporated due to unavailability of data.
- **Store-Level Inconsistency:** Data irregularities across stores made preprocessing laborious and complex.
- **LSTM Tuning Complexity:** Optimal performance required careful tuning of hyperparameters (epochs, batch sizes, dropout), which was iterative and resource-heavy.

6.6 Key Takeaways

- LSTM modeling is a powerful tool for grocery price forecasting, aiding decisions in supply chain and pricing strategies.
- Spreadsheet-based analysis helped identify retailer pricing strategies and competitive trends.
- An integrated system with real-time price scraping, LSTM-based forecasting, and user-facing dashboards can significantly enhance consumer experience and operational efficiency.

- Future improvements include benchmarking with ARIMA models, implementing attention-based LSTM for capturing recent trends, and incorporating broader market signals.

Chapter 7

Conclusion

7.1 Summary of Findings

A practical and analytical framework for comparing and forecasting grocery prices across local neighborhood stores was developed using time-series techniques and deep learning models.

- **Data on price collection:** Real-world price data were collected daily from various stores, ensuring authenticity and providing insight into grocery pricing trends.
- **Augmented Dickey-Fuller (ADF) Test:** A statistical test that provides evidence of stationarity in price trends, which is a better condition for modeling.
- **Long Short-Term Memory (LSTM) Model:** The model learned temporal patterns in the obtained time-series data, allowing for accurate forecasting of price movements. The model showed good performance across several products and demonstrated a high ability to generalize to unseen data.
- **Empowering customer decisions:** By making sense of prices, their fluctuations, and inter-store price disparity, the system empowers customers to make better shopping decisions.

These aforementioned methods, when used in combination, demonstrate that a data-driven grocery tracking and forecasting system is feasible and helps customers save money while making their purchases conveniently.

7.2 Limitations

Despite the good performance of this system, the following limitations were observed:

- **Manual Data Entry:** While daily manual data collection was accurate, it was time-consuming and could not scale effectively.
- **Limited Store Coverage:** The analysis covered only five local stores, which limits its generalization to other places.
- **Performance Tuning for Model:** The LSTM model requires significant hyperparameter tuning, such as adjustments to learning rates, batch size, and epoch counts.

- **Missing External Features:** Factors such as seasonal hiring spikes, promotional offers, and supply-demand contingencies were not included in the model.

These gaps present opportunities for future automation and scalability improvements, as well as the development of better modeling techniques.

7.3 Future Work

Following the conclusions made in this study, several avenues for improving this work are:

- **Automated Data Collection:** Implementing web scraping or store APIs (if available) could eliminate manual input and provide real-time price updates.
- **Wider Coverage:** Expanding the analysis to more stores across different geographic areas would allow for broader comparisons and greater applicability.
- **Advanced Forecasting Methods:** Experimenting with Transformer-based models or hybrid architectures that combine LSTMs with attention features could improve forecasting performance.
- **Integration of External Variables:** Incorporating other external data, such as festival dates, weather conditions, and inflation statistics, would further enhance price prediction accuracy.
- **Dynamic Web Dashboard:** Developing a real-time visualization tool, such as using Streamlit, would allow users to actively explore current pricing and predict future pricing trends.

In conclusion, this study highlights how real-world data and time-series modeling can enhance consumer behavior. It is a foundational step toward developing intelligent, automated, and personalized tools for effective price tracking in the future.

Bibliography

[1] A. Sreeram, A. Kesharwani, and S. Desai, "Factors affecting satisfaction and loyalty in online grocery shopping: an integrated model," *Journal of Indian Business Research*, vol. 9, no. 2, pp. 107–132, 2017.

[2] D. C. Buser, "Context-based recommender systems in conventional grocery–an economic analysis," in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, 2007, pp. 168b–168b.

[3] K. Varun, P. Rajesh, P. Dileep, M. Ganesh, P. Suneel, and B. S. V. Satish, "Price comparison for online shopping," p. 154, 2023, iJISRT23DEC258. [Online]. Available: https://www.ijisrt.com/price-comparison-for-online-shopping

[4] S. Bezalwar, V. Bhandekar, S. Kumbhare, R. Rebhankar, and P. Singam, "E-commerce price comparison with review sentimental analysis," *International Journal of Computer Science and Mobile Computing*, vol. 11, no. 3, pp. 108–115, 2022, impact Factor: 7.056. [Online]. Available: https://www.ijcsmc.com/docs/papers/March2022/V11I3202223.pdf

[5] H. Dharmik, P. Padmane, K. Dhoke, S. Chambhare, and D. Kohad, "A review on e-commerce price evaluation system," 2022.