

Project Presentation

“Niyojan: Demand Forecasting System”

AI-powered Retail Demand Forecasting using LSTM

Rohit Khatri - MITU22BTCS0659

Mahesh Swami - MITU22BTCS0414

Siddharth Suryawanshi - MITU22BTCS0823

Satyanarayan Mohapatro - MITU22BTCS0751

Guided by
Prof. Bhushan Bhokse

Prof. Dr. Ganesh Pathak (HOD, Computer Science & Engineering)



Department of Computer Science & Engineering, MITSOC, Loni Kalbhor

Outline

- Problem Statement & Introduction
- Modules Proposed in Phase - I
- Status of Modules Completed in Phase - I
- Modules Proposed in Phase - II
- Project Development Plan
- Status of Paper Publication
- Conclusion



1. Introduction

Motivation :

- Grocery retail faces fluctuating consumer demand driven by price, promotions, holidays, and local events
- Overstocking ties up capital and increases wastage; understocking causes stock-outs and lost sales
- Manual/spreadsheet forecasting is error-prone, slow, and lacks adaptability to market changes

Solution :

- Niyojan uses LSTM-based deep learning for time series demand forecasting with advanced pattern recognition
- Automated analytics pipeline: FastAPI backend + React dashboard for real-time insights
- Integrates seasonal patterns, price elasticity, and promotion effects for accurate predictions



NIYOJAN
DEMAND FORECASTING SYSTEM



Department of Computer Science & Engineering, MITSOC, Loni Kalbhor

Problem Statement

Problem :

- Retailers struggle with accurately predicting grocery demand due to complex patterns and variables
- Price fluctuations, promotions, and seasonal variations create non-linear demand signals
- Traditional forecasting methods fail to capture complex relationships between variables
- Forecasting errors lead to inventory imbalances, affecting profitability and customer satisfaction

Objectives :

- Build LSTM-based forecasting model capable of capturing temporal dependencies in retail data
- Implement FastAPI backend for efficient model serving and data processing
- Design React dashboard for intuitive visualization and interaction
- Enable automated reporting and alerts for inventory management
- Target accuracy: **MAPE \leq 8%**



2. Modules Proposed in Phase - I

System Layers



Data Layer

CSV ingestion → SQLite storage with data versioning



Model Layer

TensorFlow LSTM training & serving artifacts



Backend Layer

FastAPI endpoints for predict, alerts, reports



Frontend Layer

React.js UI (charts, tables, filters, exports)



Automation Layer

Email alerts + PDF/Excel generated reports



Data Collection

Historical sales data ingestion from CSV and storage in SQLite



Preprocessing

Data cleaning, normalization, and feature engineering



LSTM Training

Deep learning model training with validation



Forecast Generation

6-week ahead predictions with confidence intervals



API Deployment

FastAPI endpoints for model serving



Dashboard

React visualization and user interface



3. Status of Modules Proposed in Phase - I

Model Implementation

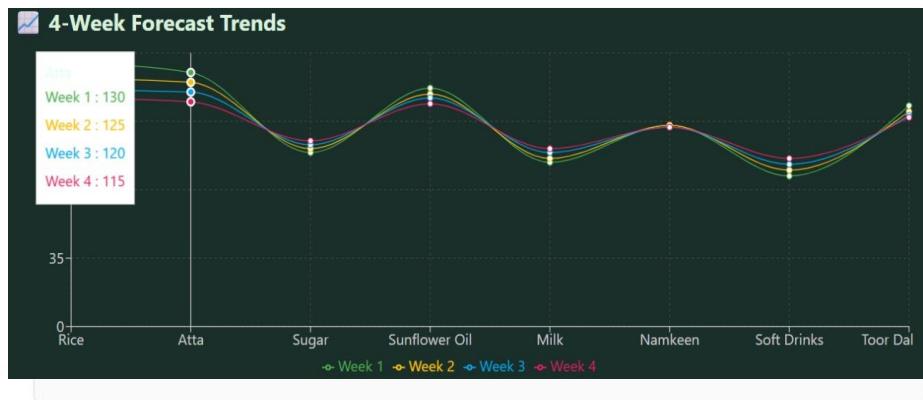
- 🕒 Look-back period: 12 weeks of historical data
- 📅 Forecast horizon: 6 weeks ahead prediction
- 📊 Loss function: Mean Squared Error (MSE)
- ⚙️ Optimizer: Adam with learning rate 0.001
- ✖ Dropout rate: 0.2 for regularization
- ⌚ Early stopping: patience 10 epochs

Backend (FastAPI)

- 🔗 REST endpoints: /predict, /alerts, /reports/{store_id}
- ↔ JSON input/output with CORS support
- ☰ Async background tasks for report generation

Frontend (React.js)

- 📊 Data visualization: Recharts library
- 💻 Tabular data: AG-Grid with sorting/filtering
- ⬇️ Export functionality: CSV/PDF reports
- 🔔 Real-time alert notifications via web sockets



Results & Benefits

Model Performance

6.8%

MAPE

11.5

RMSE

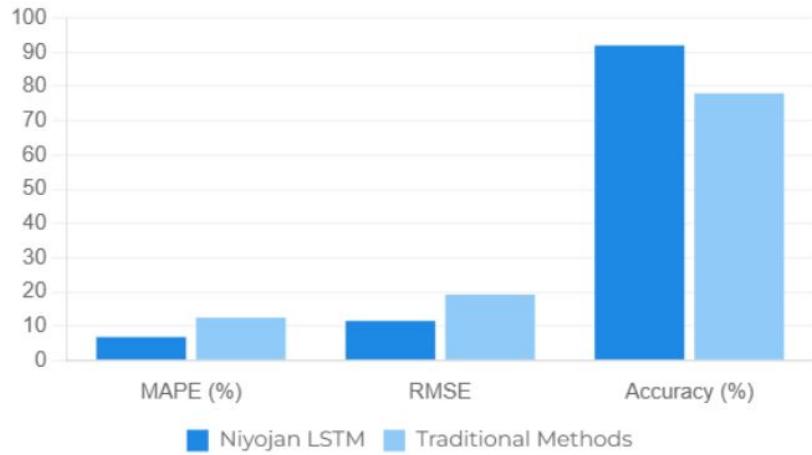
0.92

R²

Business Outcomes :

- 30% reduction in stock-out occurrences across product categories
- 18% inventory value released, improving cash flow and reducing holding costs
- Real-time forecasting delivered in under 2 seconds per request
- Increased forecasting accuracy during peak seasonal periods

Performance Comparison



Stock-Out Reduction

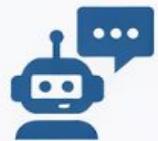


4. Modules Proposed in Phase - II



Cloud deployment (AWS/Azure) for scalability

- Deploy on AWS / Azure cloud infrastructure (with auto-scaling capabilities)



Virtual Assistant Chatbot of Niyojan

- Chatbot to provide interactive Q&A, access forecasts, generate reports



IoT sensor integration for real-time signals

- Integrate IoT sensors to capture real-time inventory, sales & environmental data



Explainable AI insights (feature attributions)

- Provide understandable reasons behind forecasts, help users interpret model decisions

5. Conclusion

- Phase 2 of Niyojan focuses on transforming the system into a **scalable, intelligent, and user-friendly platform** through cloud deployment.
- The **virtual assistant chatbot** enhances user interaction by enabling natural language access to forecasts, reports, and insights.
- **IoT sensor integration** allows real-time data ingestion, improving forecast accuracy and responsiveness.
- **Explainable AI insights** ensure transparency by clearly communicating the factors influencing demand predictions, building trust and supporting better decision-making.

"Transforming Retail Intelligence through Predictive Analytics"



Department of Computer Science & Engineering, MITSOC, Loni Kalbhor

Thank You

Team Niyojan
MIT School of Computing, MIT ADT University, Pune



Department of Computer Science & Engineering, MITSOC, Loni Kalbhor