**Context-Aware Inventory Management using Predictive Analytics**

**Project Overview**

**Objective:**

To develop a smart inventory management system that uses **AI-powered predictive analytics** to:

* Forecast product demand.
* Identify missing or low-stock products based on **season**, **weather**, and **festive trends**.
* Recommend stocking actions to prevent lost sales and overstocking.

**Problem Statement**

Retailers often struggle with:

* **Understocking** during high-demand periods (e.g., festivals, winter).
* **Overstocking** after low-demand periods.
* Failing to align inventory with **contextual factors** like weather and festivals.

Traditional systems lack the intelligence to **adapt inventory planning** to real-world external trends.

**Solution Summary**

Build a **context-aware AI system** that:

* Uses **time-series demand forecasting** for each product.
* Tracks **weather, season**, and **upcoming festivals**.
* Suggests **inventory restocking** for relevant product categories (e.g., winter clothes in winter, festive lights during Diwali).

**Tech Stack**

| **Layer** | **Tools/Tech Used** |
| --- | --- |
| Frontend | React / Angular (for dashboard UI) |
| Backend API | Python (Flask / FastAPI) |
| ML Models | Prophet, XGBoost, LSTM (optional) |
| Data Storage | MySQL / PostgreSQL / MongoDB |
| External Data | Weather APIs (e.g., OpenWeatherMap) |
| Hosting | Docker + AWS/GCP (optional for PoC phase) |

**Dataset Structure**

**1. inventory\_data.csv**

| **Product\_ID** | **Product\_Name** | **Category** | **Stock\_Level** | **Store\_Location** |
| --- | --- | --- | --- | --- |
| C101 | Cotton T-Shirt | Summer Wear | 80 | Delhi |
| W201 | Winter Jacket | Winter Wear | 0 | Delhi |
| F301 | LED Lights | Festive Decor | 5 | Delhi |
| F302 | Sweets Box | Festive Special | 0 | Delhi |

**2. weather\_forecast.csv**

| **Date** | **Location** | **Season** | **Avg\_Temperature** | **Weather\_Condition** |
| --- | --- | --- | --- | --- |
| 2025-11-05 | Delhi | Winter | 10°C | Clear |
| 2025-11-06 | Delhi | Winter | 8°C | Cloudy |

**3. festive\_calendar.csv**

| **Date** | **Festival** | **Regions** | **Festive\_Categories** |
| --- | --- | --- | --- |
| 2025-11-05 | Diwali | All India | Festive Decor, Sweets Box |
| 2025-12-25 | Christmas | Urban India | Winter Wear, Gifts |
| 2026-01-01 | New Year | All India | Gifts, Party Supplies |

**📦 Recommendation Example (Output JSON)**

{

"location": "Delhi",

"date": "2025-11-05",

"season": "Winter",

"festival": "Diwali",

"recommendations": [

{

"product\_name": "Winter Jacket",

"category": "Winter Wear",

"reason": "Winter season, stock is 0"

},

{

"product\_name": "LED Lights",

"category": "Festive Decor",

"reason": "Diwali festival, stock is below threshold"

},

{

"product\_name": "Sweets Box",

"category": "Festive Special",

"reason": "Diwali festival, stock is 0"

}

]

}

**Optional Dashboard Features**

* **Inventory Health Monitor**: View product stock vs forecasted demand.
* **Upcoming Trends**: Weather + festival alerts.
* **Restocking Suggestions**: Click to generate supplier order sheets.

**Success Criteria**

| **Goal** | **Metric** |
| --- | --- |
| Accurate Forecasting | >85% MAPE on test data |
| Useful Recommendations | >90% relevance on test cases |
| Usability | Simple dashboard or API output |

**🚀 Future Scope**

* Real-time integration with **weather APIs** and **holiday calendars**.
* ML model auto-tuning based on recent demand shifts.
* **Procurement automation**: Generate supplier orders automatically.
* Integrate with POS systems for continuous learning.

**📂 Deliverables**

1. 📁 Sample Datasets: Inventory, Weather, Festive Calendar (CSV)
2. 🧪 Python Codebase: Forecasting, Recommendation Engine
3. 📊 Dashboard or CLI: Outputs actionable recommendations
4. 📄 Documentation + ReadMe

**Roadmap for Inventory Management PoC with Predictive Analytics**

|  |  |  |
| --- | --- | --- |
| Week | Tasks & Milestones | Details |
| Week 1 | **Requirement Analysis & Data Collection** | - Define scope and objectives clearly.- Gather sample data (inventory, weather, festivals).- Identify external APIs (weather, festival calendar). |
| Week 2 | **Data Preparation & Exploration** | - Clean and preprocess datasets.- Explore historical sales, weather, and festive patterns.- Create mappings for season-product categories and festive-product categories. |
| Week 3 | **Demand Forecasting Model Development** | - Choose forecasting models (e.g., Prophet, XGBoost).- Train models on historical sales data.- Validate forecasting accuracy. |
| Week 4 | **Contextual Intelligence Integration** | - Integrate weather and festival data.- Implement logic to map weather/season and festivals to product demand categories.- Develop inventory check mechanism. |
| Week 5 | **Recommendation Engine Development** | - Build the system to generate restocking suggestions based on forecasts and current stock.- Define thresholds for alerts (e.g., low stock). |
| Week 6 | **Backend API & Data Pipeline Setup** | - Develop backend APIs to serve forecasts and recommendations.- Automate data ingestion from external sources.- Set up database/storage for inventory and forecasts. |
| Week 7 | **Frontend / Dashboard Development (Optional)** | - Create a simple UI/dashboard to visualize inventory status, upcoming festivals, and alerts.- Implement features like filter by location/date. |
| Week 8 | **Testing, Evaluation & Optimization** | - Test end-to-end pipeline.- Collect feedback from potential users.- Optimize models and system logic based on feedback.- Prepare documentation. |
| Week 9 | **Deployment & Final Review** | - Deploy PoC on cloud or local server.- Final demo and presentation preparation.- Plan next steps for scaling or production use. |

**⏳ Total Estimated Time: 8-9 weeks**