Retail Seasonal Demand Forecasting System

Overview

This project is a **Data Science / Machine Learning system** designed to forecast retail demand at the **product category level** using historical sales data. It helps retail businesses optimize inventory, reduce stockouts, and improve supply chain planning.

Problem Statement

Retail businesses experience significant fluctuations in demand due to:

- Seasonal trends
- Promotions
- Changing customer behavior

Without accurate forecasting, businesses face:

- Overstocking → Increased holding costs
- Stockouts → Lost sales
- Poor supply chain planning

Objectives

Primary Goals

- Forecast category-level demand using historical data
- Capture seasonality and trends using ARIMA/SARIMAX
- Provide REST API for forecast retrieval
- Build an interactive dashboard for visualization
- Enable configurable forecasting horizon

Stretch Goals

- Auto-ARIMA for automatic parameter tuning
- Multi-seasonality handling
- External regressors (price, promotions, holidays)
- Confidence intervals for forecasts
- Real-time data streaming
- Model retraining pipeline

Target Users

- Retail planners
- Inventory managers
- Data analysts
- Supply chain teams

Tech Stack

Frontend

- React
- Tailwind CSS
- Recharts

Backend

- FastAPI (Python)

Machine Learning

- statsmodels (ARIMA / SARIMAX)
- pandas
- numpy

Database

- PostgreSQL

Deployment

- Docker
- AWS EC2

System Architecture

Components:

- **Frontend (React)** → User interaction & visualization
- **Backend API (FastAPI)** → Handles requests & responses
- **Model Service** → Training & forecasting
- **Database (PostgreSQL)** → Stores processed data & models

Features:

Core Features

- Upload retail dataset (CSV)
- Data preprocessing and validation
- Category-level aggregation
- ARIMA-based forecasting
- REST API endpoints
- Interactive dashboard

ML-Specific Features

- Train model per category
- Save trained model parameters
- Forecast configurable horizon
- Seasonal decomposition
- Model evaluation metrics (sMAPE, RMSE)

Dataset

Example Dataset: UCI Online Retail Dataset

Fields

- InvoiceNo
- StockCode
- Description
- Quantity
- InvoiceDate
- UnitPrice
- CustomerID
- Country

Derived Features

- Product Category
- Sales = Quantity × UnitPrice
- Time index (daily / weekly / monthly)

Data Pipeline

1. **Data Ingestion**
    - Upload CSV or connect to database
    - Schema validation
2. **Data Cleaning**
    - Remove negative quantities (returns)
    - Handle missing values
    - Convert timestamps
3. **Aggregation**
    - Group by category and time interval
4. **Feature Engineering**
    - Rolling averages
    - Lag features
    - Seasonal decomposition
5. **Storage**
    - Store processed data in PostgreSQL

Machine Learning Approach

Model: **ARIMA / SARIMAX**

Captures:

- Trend (differencing)
- Seasonality
- Autocorrelation

Workflow

1. Stationarity check (ADF test)
2. Differencing
3. Parameter selection (p, d, q, P, D, Q, s)
4. Model fitting per category
5. Forecast generation

Explainability

- Historical vs predicted plots
- Confidence intervals
- Model parameters
- Residual diagnostics
- Error metrics (sMAPE, RMSE)

Non-Functional Requirements

- API response time < 500ms
- Scalable architecture
- Data consistency
- Error handling
- Secure endpoints

Constraints & Assumptions

- Sufficient historical data is available
- Categories are predefined or mapped
- Seasonality is stable
- ARIMA works best for linear patterns

Success Metrics

- sMAPE ≤ 20%
- RMSE ≤ 10–20%
- Forecast accuracy ≥ 80–90%
- API latency ≤ 500 ms
- User engagement

Risks

- Poor data quality
- Concept drift (changing demand patterns)
- Sparse data for some categories

- Overfitting

Setup Instructions

Prerequisites

- Python 3.9+
- Node.js
- PostgreSQL
- Docker (optional)

Backend Setup

```
cd backend

pip install -r requirements.txt

uvicorn main:app --reload
```

Frontend Setup

```
cd frontend

npm install

npm run dev
```

Docker

```
docker-compose up --build
```

Future Improvements

- Auto-ARIMA implementation
- Real-time data ingestion
- Advanced models (Prophet, LSTM)
- Multi-category forecasting
- Cloud scaling

License

This project is for academic purposes.