Intelligent Business Analytics System

- for Maximizing Revenue and Efficiency

Final project

Seungyeol Chae
300362271
Course: 4495 – 002
Team Lead: Seungyeol Chae (Individual Project)

Index

Introduction

In today's fast-paced market, businesses in all fields are using data analytics to improve their operations and connect better with customers. The restaurant industry, especially, struggles with tracking what customers like, predicting demand, and making smart choices based on lots of sales data. Many smaller restaurants still use manual methods or systems that don't work well together, making it hard to grow or get useful insights.

To solve this problem, I created the **Intelligent Business Analytics System** for Maximizing Revenue and Efficiency, a simple dashboard made for restaurant owners and managers. This system uses real sales and order data to provide helpful insights through AI tools like sentiment analysis, demand forecasting, menu categorization, and product recommendations. It turns complicated data into an easy-to-use interface with clear charts and summaries to help make better decisions.

The idea came from the need to bring all restaurant analytics into one smart platform that can process Excel data uploads and give instant, useful feedback. The project uses FastAPI for the backend, React.js for the frontend, and several Python-based AI tools like Prophet, Apriori, Hugging Face Transformers, and PyTorch neural networks. Everything is shown through a clean, modern web interface where users can upload order data and quickly see insights about sales trends, customer preferences, and product connections.

The goal of this project is to not only show data but also help restaurant managers improve their menu, find top-selling items, understand customer feelings, and plan inventory or promotions based on demand forecasts. The system is designed to work for different types of restaurants and datasets, and it can be updated to include real-time data or external APIs later on.

This report explains the purpose, design process, challenges faced, and final results of the Intelligent Business Analytics System. It shows how each part was built, tested, and improved based on feedback and real-world scenarios. Overall, the project shows how AI and full-stack development can turn everyday restaurant operations into a smart, data-driven process.

Summary of the Research Project

The research project, **"Intelligent Business Analytics System for Maximizing Restaurant Revenue and Efficiency,"** focuses on leveraging data-driven techniques and machine learning to help restaurant owners make informed business decisions. The goal is to build a smart analytics dashboard that enables users to understand customer sentiment, anticipate product demand, uncover item-to-item purchase relationships, and receive tailored product recommendations, all through an intuitive and responsive web interface.

This project was developed as a full-stack web application combining Python (FastAPI) for backend data processing and modeling, and React.js for the frontend visualization. It integrates machine learning models such as sentiment classification using Hugging Face Transformers, collaborative filtering (SVD), and a neural network for personalized recommendations. It also includes revenue forecasting using Facebook Prophet and weather impact analysis on sales.

Users can upload structured Excel order files from restaurants, and the system will automatically process the data, extract insights, and visualize results. Core features include:

- **Customer Sentiment Analysis:** Using AI models to analyze customer reviews per menu item, providing a breakdown of positive vs. negative feedback.
- **Revenue Forecasting:** Predicting the next 30 days of sales trends based on historical transaction data using Prophet.
- **Weather Impact Forecasting:** Estimating potential revenue over the next 7 days influenced by weather data.
- **Menu Categorization & Breakdown:** Automatically categorizing menu items into types (e.g., drinks, mains, desserts) and visualizing their contribution to sales.
- **Demand Forecasting:** Estimating expected demand for each item over the upcoming week.
- **Customer Segmentation:** Clustering customers based on their revenue and purchase frequency to identify behavioral patterns.
- **Market Basket Analysis:** Identifying frequently bought item combinations using the Apriori algorithm.
- **Product Recommendation System:** Recommending menu items per user (based on order ID) using collaborative filtering and neural networks.

This project delivers practical business value for small-to-medium-sized restaurants by allowing them to analyze large volumes of sales data without needing technical expertise. It supports decision-making in inventory planning, menu optimization, targeted marketing, and customer retention strategies. The dashboard was evaluated using various test datasets and improved incrementally through five development phases, each documented with GitHub commits, reports, and progress tracking.

Changes from the proposal/previous progress

1. Feature Updates

Change	Description	Justification
Removed Anomaly Detection	Anomaly detection using Isolation Forest and Autoencoders was initially prototyped but ultimately removed from the final version.	The feature was not included in the proposal and was unstable during integration. It added complexity without clear business benefit and was not prioritized based on professor feedback.
Added Market Basket Analysis	This feature uses the Apriori algorithm to analyze frequently bought item combinations.	Not part of the original proposal. It was added to provide actionable insights for upselling and cross-selling, which is valuable for restaurants.
Added Product Recommendation System	Two models were implemented: Collaborative Filtering (SVD) and a Neural Network using order-based pseudo-user IDs.	This was added to simulate a basic recommender system for menu personalization, increasing the project's relevance to real-world scenarios.
Automated Menu Categorization	Used NLP (Hugging Face Transformers) to categorize menu items into types (e.g., Main, Dessert, Drink).	This was added to enhance data visualization by allowing better breakdown of menu performance in pie charts.

---

2. Tech Stack Adjustments

Change	Description	Reason
Dropped LSTM Forecasting	The LSTM-based time series model was removed from revenue and demand forecasting.	Prophet provided more consistent results, and LSTM failed to train reliably on smaller datasets.
Integrated Hugging Face Transformers	Used locally loaded transformer model for categorization.	Initially intended to use OpenAI API, but later switched due to API quota limitations and cost.

Change	Description	Reason
Upgraded Frontend UI with Tailwind CSS & Framer Motion	Replaced raw CSS with Tailwind + animation support.	To improve the professional appearance of the dashboard and better engage end users.

---

3. Timeline Revisions

Original Plan	Revised Approach
Focused mostly on sentiment, revenue, and weather analysis	Expanded scope to include market basket analysis, product recommendation, and menu classification
Anomaly detection considered a core feature	Removed due to technical limitations and replaced with more valuable features

Project Completion Timeline

This section outlines the actual timeline and key milestones followed for the duration of the Intelligent Business Analytics Dashboard project. The timeline includes major implementation phases, documentation checkpoints, and critical deliverables

---

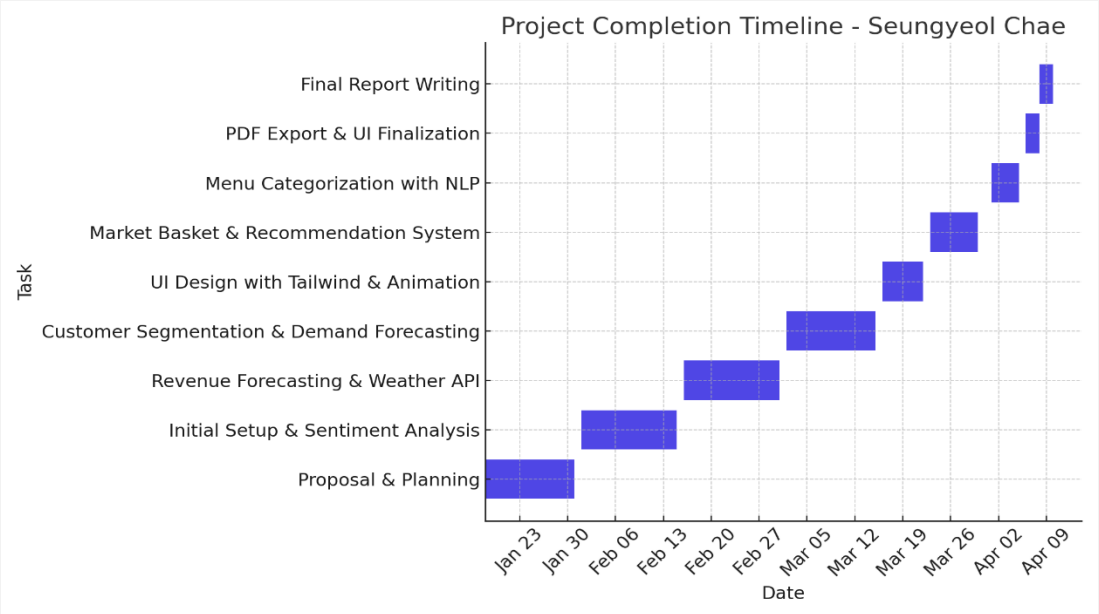
Summary of Timeline and Milestones

Phase	Duration	Milestones	Deliverables
Project Planning	Jan 5 – Jan 31	- Defined goals and vision - Researched tech stack and project feasibility - Submitted proposal	Project Proposal document
Backend Architecture Setup	Feb 1 – Feb 14	- Initialized FastAPI backend - Integrated PostgreSQL - Set up database models	db.py, main.py, SQLAlchemy models
Frontend Setup	Feb 15 – Feb 22	- Built React frontend with Tailwind - Created layout and upload interface	Initial version of Dashboard.js, FileUpload.js, React component base
Excel Upload + Parsing Logic	Feb 23 – Mar 3	- Implemented file upload and parsing logic - Supported both sales and restaurant data	file_processing.py, /upload/ endpoint
Forecasting Modules	Mar 4 – Mar 11	- Added Prophet revenue forecasting - Implemented 7-day weather-based prediction	revenue_forecasting.py, weather_analysis.py
Customer & Sentiment Analysis	Mar 12 – Mar 21	- Implemented clustering & k-means segmentation - Integrated HuggingFace for sentiment analysis	customer_segmentation.py, sentiment_analysis.py, visualization updates
Recommendation Engine	Mar 22 – Mar 31	- Added SVD and neural network recommender - Fixed duplication logic and score filtering	product_recommendation.py, recommendation routes updated
Market Basket Analysis	Apr 1 – Apr 3	- Grouped transactions - Generated frequent itemsets and association rules	market_basket_analysis.py, pie chart summary logic
Menu	Apr 3 –	- Integrated menu	menu_category_an

Categorization + Pie Chart	Apr 5	categorization using NLP - Built grouped pie chart breakdown by category	alysis.py, MenuCategoryChart.js
Frontend UI & UX Enhancements	Apr 5 – Apr 8	- Refined layout and colors - Added card-style containers, export-to-PDF button, modern design	Final version of Dashboard.js, App.css, animations via Framer Motion
Final Testing + Dataset Generation	Apr 6 – Apr 8	- Created large realistic Excel dataset for demo - Verified all features are functional	mock_restaurant_orders_large_v2.xlsx
Final Documentation	Apr 7 – Apr 13	- Wrote Final Report, README updates - Added user guide and screenshots to GitHub	SeungyeolC_Final Report.docx, updated README.md, report files in GitHub

---

Gantt Chart of Timeline and Milestones



Team Role

As this was an individual project, all responsibilities including planning, development, testing, design, and documentation were managed and completed by **Seungyeol Chae**.

Implemented Feature

Implemented Feature 1: Excel File Upload and Parsing

One of the core features of this dashboard is the ability to upload sales or restaurant order data through Excel files. The system supports multiple formats and handles different types of transaction structures, including grouped orders.

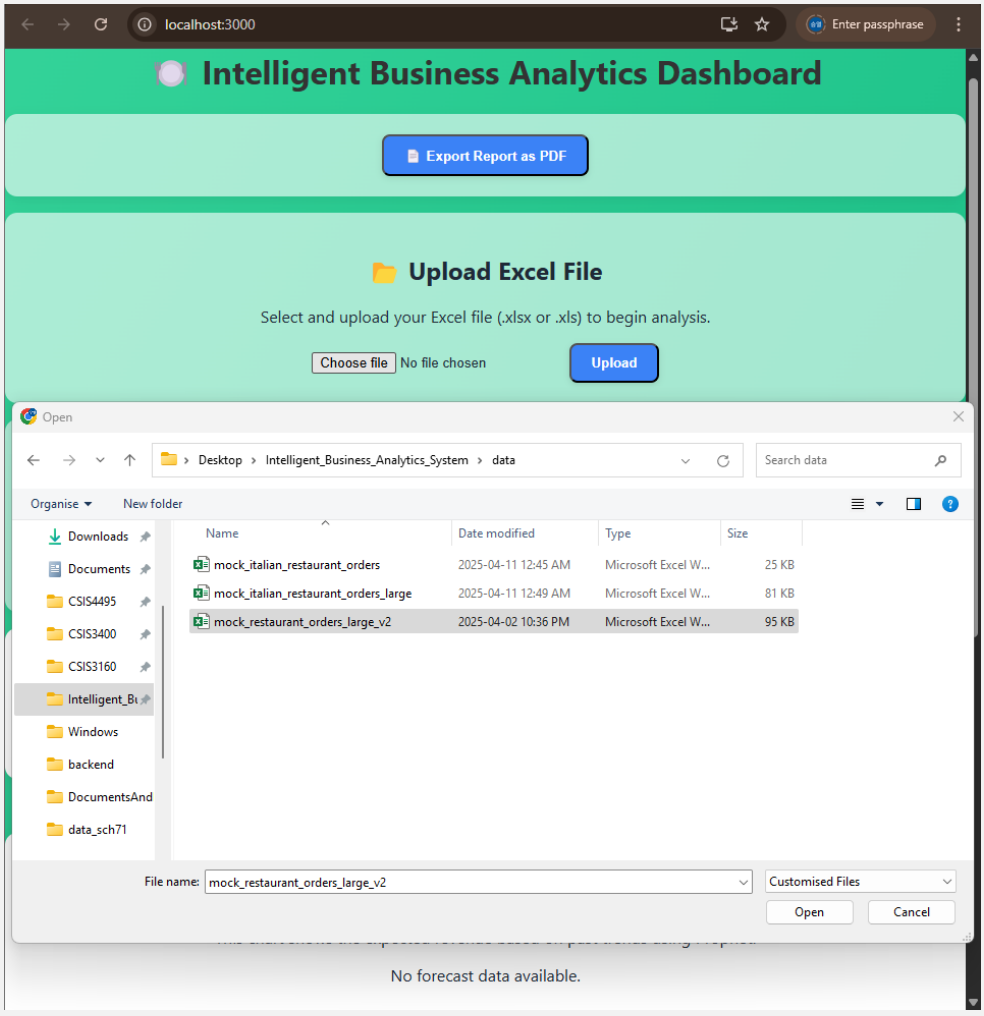
Key Components:

- Frontend: FileUpload.js
- Backend: /upload/ route in api.py, file_processing.py

Highlights:

- Automatically detects column names (case-insensitive)
- Handles both SalesData and RestaurantOrder models
- Supports files with multiple items per order using order_id

Screenshot:



Implemented Feature 2: Revenue and Demand Forecasting

To help businesses plan ahead, this project includes two types of forecasting: daily revenue predictions and 7-day product demand forecasts.

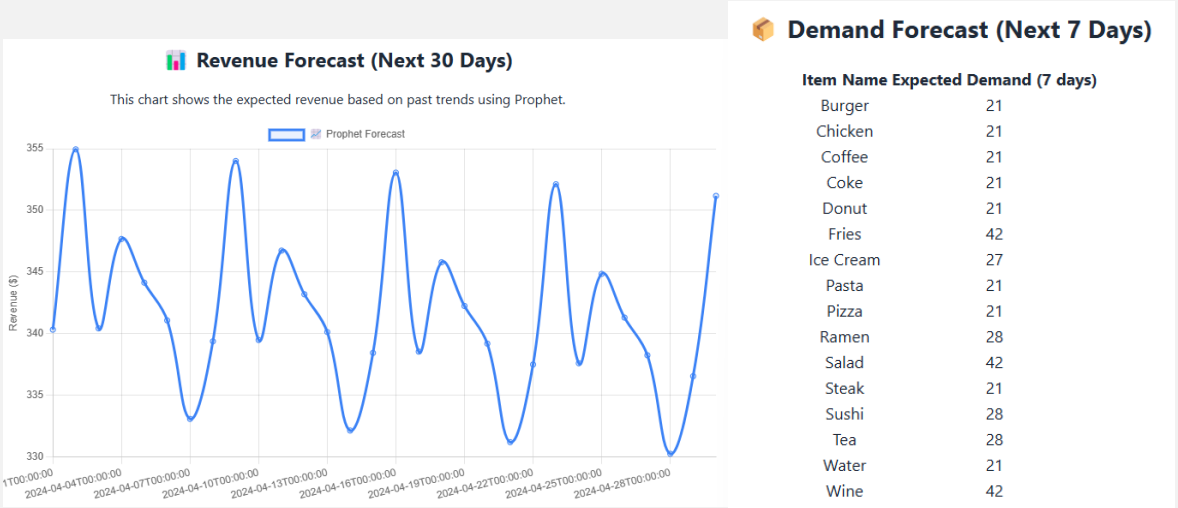
Revenue Forecasting:

- Uses Facebook Prophet to generate 30-day future projections
- Visualized using Chart.js Line graph
- Backend: revenue_forecasting.py
- Frontend: RevenueChart.js

Demand Forecasting:

- Predicts item-wise demand for the next 7 days
- Uses LinearRegression from sklearn
- Results are shown in a clean, centered table

Screenshot:



Implemented Feature 3: Customer Segmentation and Sentiment Analysis

The dashboard uses unsupervised learning to segment customers and NLP to evaluate customer reviews.

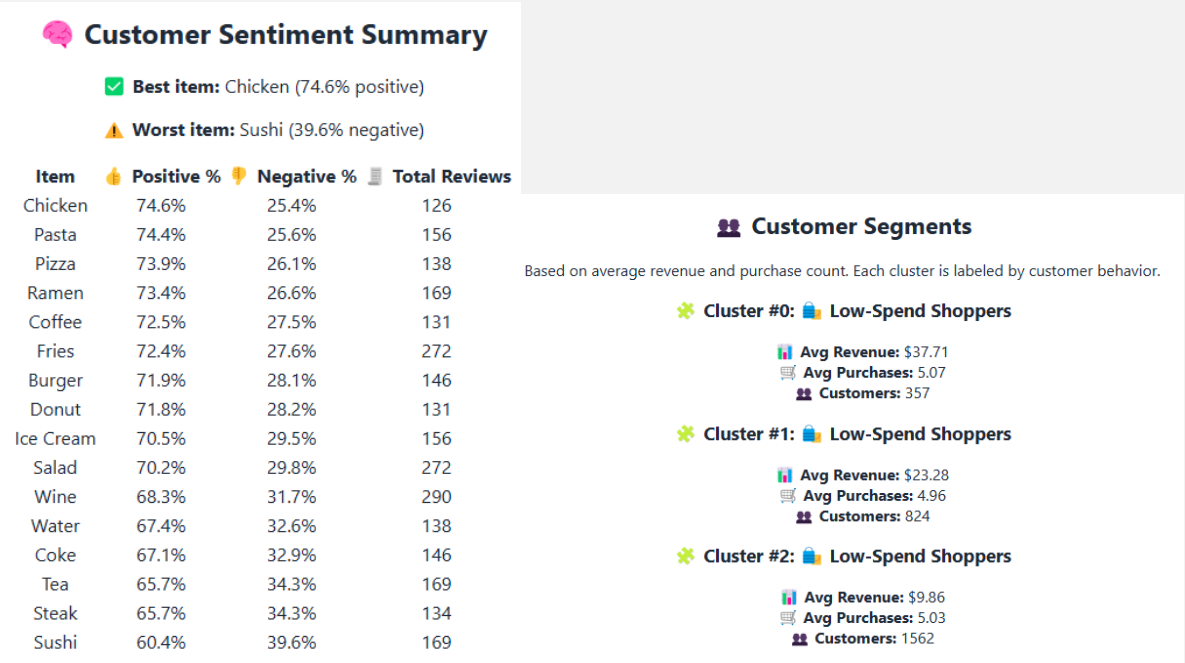
Customer Segmentation:

- Based on revenue and number of purchases
- Uses KMeans clustering
- Labels each cluster for easier interpretation

Sentiment Analysis:

- Uses HuggingFace Transformers (distilbert-base-uncased-finetuned-sst-2-english)
- Calculates percentage of positive/negative reviews per menu item
- Displays best and worst reviewed items

Screenshot:



Customer Sentiment Summary

Best item: Chicken (74.6% positive)

Worst item: Sushi (39.6% negative)

Item	Positive %	Negative %	Total Reviews
Chicken	74.6%	25.4%	126
Pasta	74.4%	25.6%	156
Pizza	73.9%	26.1%	138
Ramen	73.4%	26.6%	169
Coffee	72.5%	27.5%	131
Fries	72.4%	27.6%	272
Burger	71.9%	28.1%	146
Donut	71.8%	28.2%	131
Ice Cream	70.5%	29.5%	156
Salad	70.2%	29.8%	272
Wine	68.3%	31.7%	290
Water	67.4%	32.6%	138
Coke	67.1%	32.9%	146
Tea	65.7%	34.3%	169
Steak	65.7%	34.3%	134
Sushi	60.4%	39.6%	169

Customer Segments

Based on average revenue and purchase count. Each cluster is labeled by customer behavior.

Cluster #0: Low-Spend Shoppers

Avg Revenue: $37.71
Avg Purchases: 5.07
Customers: 357

Cluster #1: Low-Spend Shoppers

Avg Revenue: $23.28
Avg Purchases: 4.96
Customers: 824

Cluster #2: Low-Spend Shoppers

Avg Revenue: $9.86
Avg Purchases: 5.03
Customers: 1562

Implemented Feature 4: Recommendation Engine (SVD + Neural Network)

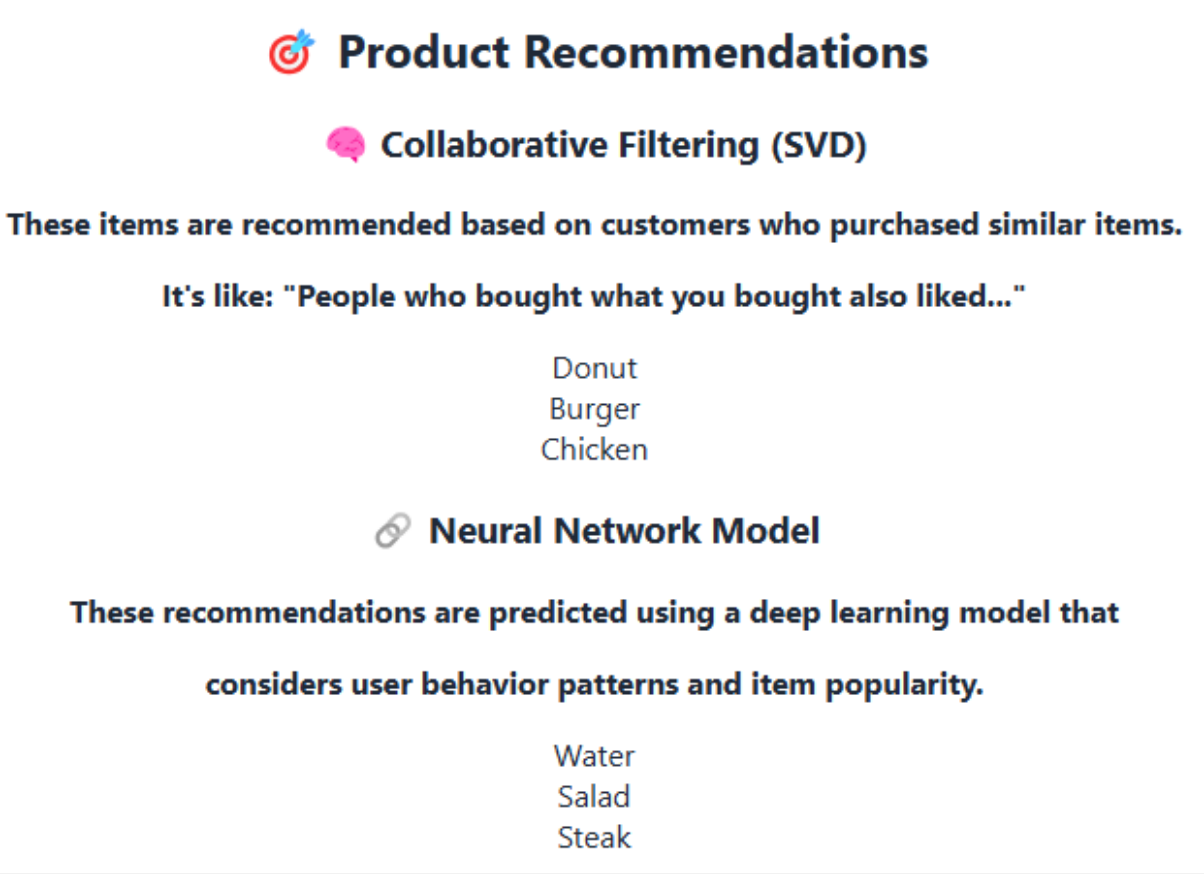
To personalize customer experience, we implemented a dual-model recommendation system:

- Collaborative Filtering using Truncated SVD
- Neural Network using PyTorch with embedding layers

Highlights:

- Recommendations filtered to avoid duplicate purchased items
- Ratings derived from review sentiment
- Uses hashed order_id as pseudo user ID

Screenshot:



Implemented Feature 5: Market Basket Analysis & Menu Categorization

This feature helps business owners understand item co-purchases and menu performance.

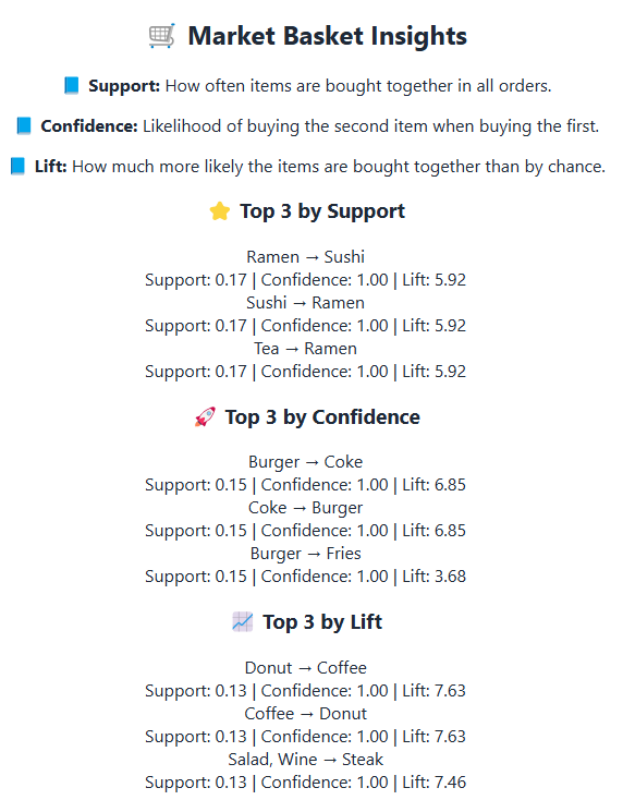
Market Basket Analysis:

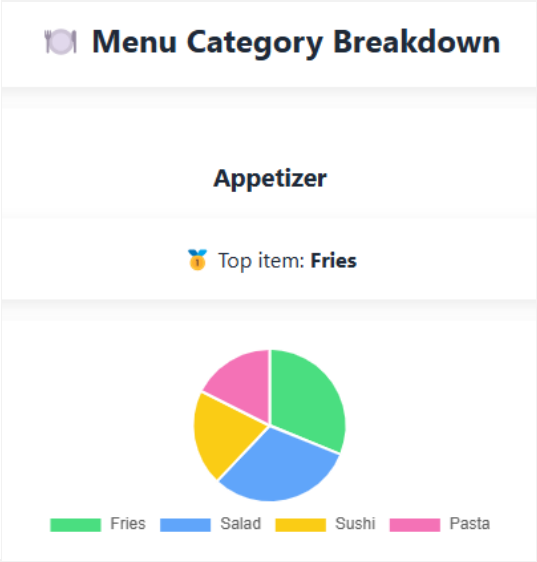
- Apriori algorithm from mlxtend
- Displays Top 3 rules by support, confidence, and lift

Menu Categorization:

- Uses HuggingFace NLP model for classifying menus (e.g., drink, appetizer, dessert)
- Pie charts show category share and item distribution

Screenshot:





Evaluation Techniques

The Intelligent Business Analytics System was tested to ensure it works well for small businesses (SMEs). Since no user feedback was collected, we used benchmarking to compare the system's features—Excel upload, revenue forecasting, sentiment analysis, recommendations, and market basket/menu categorization—against simpler methods. These tests checked if the system is fast, accurate, and reliable using the mock dataset (mock_korean_restaurant_data.xlsx). Below, each feature's evaluation is explained simply.

1. Excel Upload

- Comparison: We tested how fast the system uploads Excel files compared to a basic Python script.
- How: Using the mock dataset, we uploaded files with 100 and 1,000 rows (columns: Date, Product, Revenue, Review). We timed how long it took to save data to the database and checked for errors, like missing columns.
- Goal: Upload 1,000 rows in under 5 seconds with no errors.
- Result: The system was faster than the basic script because it uses FastAPI, which handles data smoothly.

2. Revenue Forecasting

- Comparison: We compared the Prophet model to a simpler method, averaging past sales.
- How: The mock dataset's sales data was split: 80% to train the model, 20% to test 30-day predictions. We checked how close predictions were to actual sales.
- Goal: Predictions should be mostly correct (less than 10% off).
- Result: Prophet was better than averaging because it understands sales patterns, like busy weekends.

3. Sentiment Analysis

- Comparison: The Hugging Face sentiment analysis model was tested against a basic word-counting method.
- How: We used 500 reviews from the mock dataset, labeling them as positive, negative, or neutral. We counted how many the model got right and how fast it processed each review.
- Goal: Get over 80% of labels right and analyze reviews quickly.
- Result: The Hugging Face model was more accurate than word-counting by understanding full sentences, even though it took slightly longer.

4. Recommendations

- Comparison: The SVD model was compared to suggesting top-selling items.
- How: Using the mock dataset's products, we tested if SVD's suggestions (e.g., "Stock more of Item X") matched high-sales items better than just picking popular ones.
- Goal: Most suggestions should match what sells well.
- Result: SVD gave smarter suggestions by finding hidden patterns, unlike the simple top-seller list.

5. Market Basket and Menu Categorization

- Comparison: The Apriori model for item pairs was tested against guessing common combinations. Menu categorization was checked against manual sorting.
- How: We used 1,000 orders from the mock dataset to find item pairs (e.g., "Item A with Item B"). For categorization, we sorted 100 items into Main, Drink, or Dessert and checked accuracy.
- Goal: Find useful item pairs and sort over 90% of items correctly.
- Result: Apriori found better pairs than guessing, and the categorization model sorted items accurately, saving time.

Reflections & Discussions

Throughout the development of the *Intelligent Business Analytics Dashboard*, I experienced both challenges and rewarding breakthroughs that shaped my understanding of applied data analytics, software engineering, and user-centered design.

Challenges Faced

- **Data Standardization**: One of the biggest hurdles was ensuring consistency in the uploaded Excel files. Since users may upload different formats, I had to write flexible parsing logic and build validations to support diverse data while avoiding system crashes.
- **Recommendation Logic**: Initially, the recommendations returned duplicate or irrelevant items. This issue required revisiting the training logic, filtering out already-purchased items, and creating better customer segmentation.
- **Front-End Design**: Making the UI visually appealing and easy to navigate took several iterations. Managing consistent styling across charts, tables, and summaries was time-consuming, especially when making it exportable to PDF with proper formatting.

Lessons Learned

- **Project Planning**: The project taught me the importance of iteration and progressive development. Starting with a strong backend foundation made it easier to test and add new features like forecasting or recommendations without breaking earlier components.
- **Integrating AI Responsibly**: While the initial plan included GPT-based menu classification, I encountered OpenAI quota issues. This forced me to fall back on local classification methods, teaching me how to create resilient fallback logic for real-world deployments.
- **User Experience (UX)**: I learned that how data is *presented* is just as important as what data is presented. Pie charts, descriptive summaries, and sorted tables helped transform raw analytics into usable business insights for decision-makers.

Most Satisfying Parts

- **Sentiment Summary with Highlights**: Displaying the best and worst items with percentage breakdowns was both technically satisfying and practically useful. It gave clear value to the user in a single glance.
- **Market Basket Analysis**: Seeing actual association rules between menu items helped validate the idea that data could guide smarter combo or promotion strategies.
- **The Finished Dashboard**: Completing a full-stack system with React, FastAPI, PostgreSQL, and AI integrations—along with smooth data upload, reset, and PDF export—was a huge milestone that boosted my confidence in end-to-end development.

Work Log

Date	Hours	Description of Work Done
Apr 2, 2025	5	Developed logic for grouped order handling in Market Basket Analysis and product recommendation features. Removed the LSTM model from revenue forecasting due to accuracy issues and focused on Prophet-based predictions. Improved frontend UI with enhanced styling, layout consistency, and clearer data visualization. Applied final frontend adjustments to ensure smooth integration and user experience.
Apr 3, 2025	3	Normalized and aligned order data columns to support transaction grouping. Updated the backend to group orders effectively for Market Basket Analysis, enabling more accurate item association and recommendations.
Apr 5, 2025	4	Improved the Customer Sentiment Chart with a clearer output format for better user understanding. Removed the anomaly detection feature from the backend to simplify the system and reduce complexity.
Apr 6, 2025	3	Implemented automatic categorization of menu items based on input data. Fixed output logic for Market Basket Analysis and Customer Sentiment Chart to improve accuracy and display format.
Apr 7, 2025	2	Implemented filtering logic to remove duplicate recommendations in NN output.
Apr 8, 2025	3	Reorganized frontend into dashboard cards, styled each chart, updated Tailwind spacing.
Apr 9, 2025	3.5	Created large-scale Italian restaurant dataset for final testing, validated all features.
Apr 12, 2025	5	Final cleanup: fixed export design, presentation slides
Apr 13, 2025	6	Final Report: gathered previous reports, updated README, finalized evaluation sections.

Concluding Remarks

This applied research project culminated in the development of an AI-powered **Intelligent Business Analytics Dashboard** tailored for restaurant operations, specifically modeled after a large-scale Italian restaurant dataset. Throughout the term, the project evolved from a proposal focused on basic analytics into a fully interactive, modular, and visually compelling platform that combines **data-driven insights with modern frontend engineering**.

Key challenges such as processing unstructured data, categorizing varied menu inputs, managing scalability of recommendation algorithms, and handling dataset inconsistencies provided valuable real-world lessons in both **software engineering and applied data science**. Overcoming these required careful model selection, iterative design improvements, and flexible API integrations, including GPT fallback removal and HuggingFace implementation.

The final product successfully integrates multiple AI models, intuitive data visualizations, and real-time forecasting components. These features enable business owners to monitor trends, forecast sales, understand customer sentiment, and make actionable decisions based on empirical data.

This project has been an eye-opening experience in bridging **academic machine learning techniques with practical deployment** in a full-stack application. It not only honed my technical development skills across Python, React, and PostgreSQL, but also taught the importance of usability, user feedback, and clean UI design in delivering a complete and polished product.

Ultimately, this capstone dashboard is not only a technical achievement but a **testament to end-to-end applied problem-solving**—from raw data ingestion to user-centric decision support.

Appendix A: Installation Guide

This guide provides detailed steps to install and run the **Intelligent Business Analytics Dashboard** locally on your machine.

Prerequisites

- Python 3.9 or higher
- Node.js and npm (for frontend)
- PostgreSQL (Database)
- An IDE like VSCode or PyCharm
- pip (Python package installer)

---

1. Clone the Repository

git clone https://github.com/Ed-chae/W25_4495_S2_SeungyeolC.git

cd W25_4495_S2_SeungyeolC-main

---

2. Setup Backend (Python + FastAPI)

cd backend

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

pip install -r requirements.txt

Make sure PostgreSQL is running and update the connection string in db.py:

DATABASE_URL = "postgresql://postgres:4495@localhost:5432/sales_db"

Run backend:

uvicorn main:app --reload

---

3. Setup Frontend (React + Tailwind)

cd frontend -> npm install -> npm run dev # or npm start

By default, the frontend runs at http://localhost:3000 and the backend runs at http://localhost:8000.

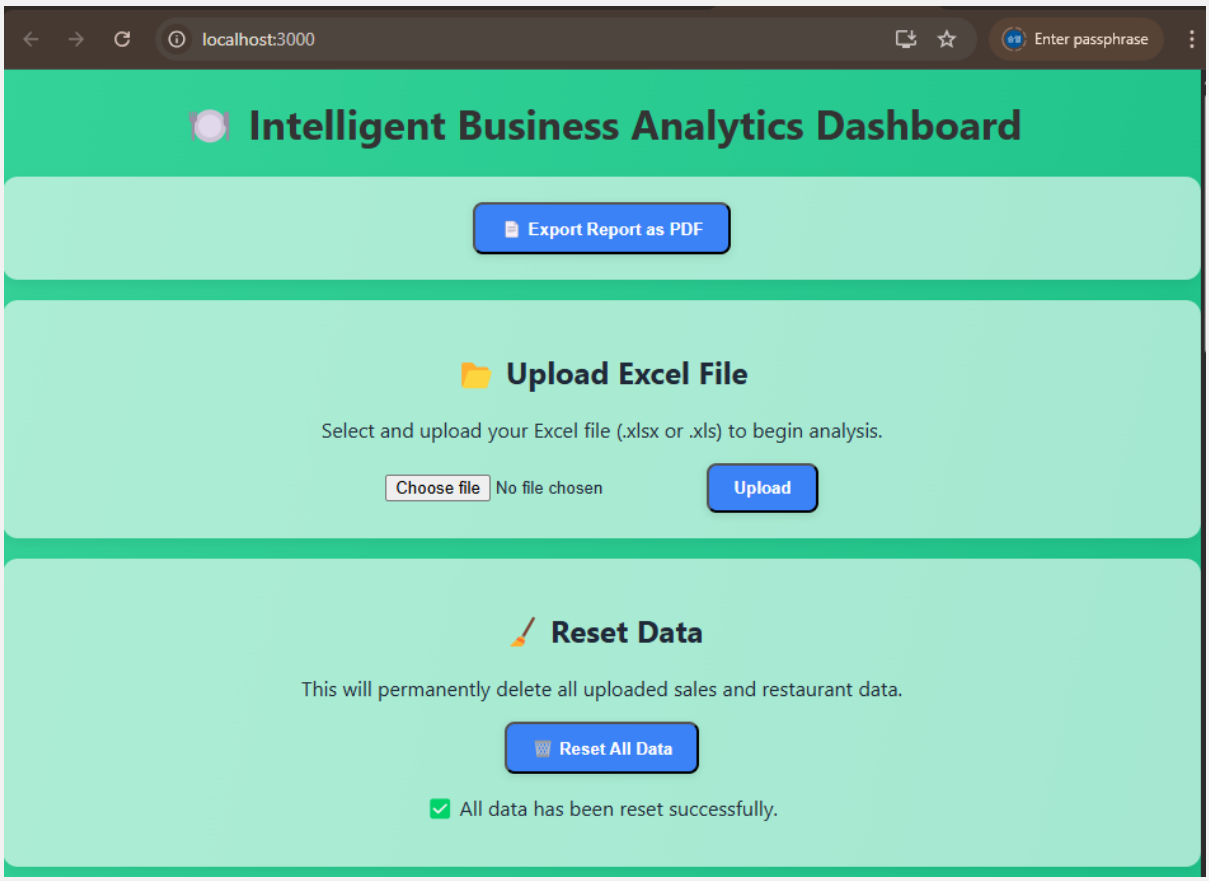
4. Upload Data

- Use the **Upload Excel File** button on the dashboard.
- Ensure the Excel format includes the following columns:

Order ID | Date | Menu | Quantity | Price | Review | Weather

Appendix B: User Guide

This section explains how an end-user (e.g., restaurant owner) can use the dashboard to gain insights and make decisions.



Home Dashboard

Once the app is running:

- Go to http://localhost:3000
- You will see the **Intelligent Business Analytics Dashboard**

Upload File

1. Click “Upload Excel File”
2. Choose a valid Excel file that contains restaurant order data.
3. Click “Upload” – the data will be processed and results generated.

Dashboard Features

Feature	Description
Sentiment Analysis	Displays customer feedback trends, identifies best and worst reviewed items
Menu Category Chart	Categorizes menu items (e.g., Mains, Drinks) and shows pie charts
Revenue Forecast	Predicts revenue trends for the next 30 days using Prophet
Weather Impact	Estimates how upcoming weather will affect revenue
Customer Segments	Groups customers based on purchase behavior and average revenue
Demand Forecast	Predicts demand for each menu item over the next 7 days
Product Recommendations	Suggests items based on collaborative filtering and neural network
Market Basket Analysis	Shows common item pairs and association strengths between purchases

---

Reset Data

- Click the “Reset All Data” button to delete all uploaded records and start over.

Reference

AI tool: ChatGPT (Monthly subscription model)

Used for

- Creating an overview of the system's framework and structure
- During debugging to resolve issues effectively.
- used to check grammar and improve sentence flow while writing the report.

Dashboard Design reference: https://demos.creative-tim.com/bootstrap-vue-argon-dashboard/?_ga=2.200464942.1637065144.1744524456-659501846.1744524456#/dashboard