

Intelligent Business Analytics System

– Maximizing Revenue and Efficiency

MIDTERM REPORT

Course: 4495 - 002

Student Name: Seungyeol Chae

Student ID: 300362271

Team Lead: Seungyeol Chae

GitHub Repository: https://github.com/Ed-chae/W25_4495_S2_SeungyeolC.git

Video Demonstration Link: https://drive.google.com/file/d/17siOSDnCBD0dJqieH4q1ScjYxJS1nMsM/view?usp=share_link OR <https://youtu.be/Asv6-DxklrQ>

Introduction

Background and Context

In today's rapidly evolving business landscape, organizations are struggling with fragmented data systems, lack of predictive modeling, and challenges in integrating customer feedback into operational strategies. Many small-to-medium enterprises (SMEs) lack sophisticated analytics tools that can process large volumes of data efficiently to drive business growth and optimize revenue.

The **Intelligent Business Analytics System** is designed to address these gaps by incorporating **predictive modeling, text analytics, and weather impact analysis** into a unified analytics solution. The system provides actionable insights through an interactive dashboard, helping businesses optimize **inventory management, revenue forecasting, and strategic decision-making**.

Research Questions

The system aims to address the following key questions:

1. **How can predictive modeling optimize inventory management and revenue generation?**
2. **What insights can text analytics provide from customer feedback to improve decision-making?**
3. **How does weather data influence business performance, and how can it be leveraged for strategic planning?**

Existing Knowledge Gaps

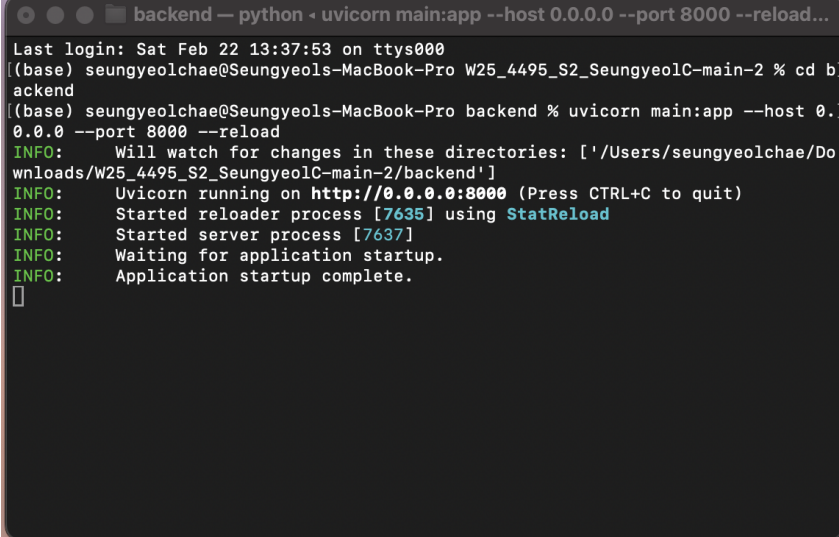
- **Limited Integration of Predictive Analytics with Weather and Text-Based Insights:** Most current models do not combine these diverse data sources into a single predictive framework.
- **Inadequate Data Processing Tools for SMEs:** Many small businesses lack automated tools for efficient data validation, processing, and reporting.
- **Lack of Real-Time Data Visualization:** Business intelligence tools often fail to provide dynamic, real-time insights for decision-making.

Expected Contributions

- **Enhanced Forecasting Accuracy:** By integrating weather and sentiment analysis into predictive models, businesses can improve revenue forecasting.
- **Real-Time Visualization and Insights:** A dashboard displaying dynamic trends, correlations, and recommendations.
- **Actionable Decision Support:** Providing small-to-medium businesses with insights to improve resource allocation and marketing strategies.

How to Run the Program

1. Start the Backend Server



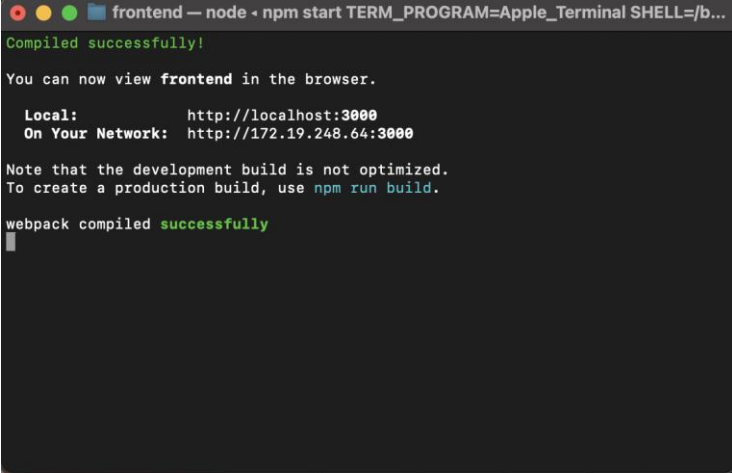
```
backend — python · uvicorn main:app --host 0.0.0.0 --port 8000 --reload...
Last login: Sat Feb 22 13:37:53 on ttys000
(base) seungyeolchae@Seungyeols-MacBook-Pro W25_4495_S2_SeungyeolC-main-2 % cd b[
ackend
(base) seungyeolchae@Seungyeols-MacBook-Pro backend % uvicorn main:app --host 0.
0.0.0 --port 8000 --reload
INFO: Will watch for changes in these directories: ['/Users/seungyeolchae/Do
wnloads/W25_4495_S2_SeungyeolC-main-2/backend']
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO: Started reloader process [7635] using StatReload
INFO: Started server process [7637]
INFO: Waiting for application startup.
INFO: Application startup complete.
█
```

- Open your terminal and navigate to the backend directory.
- Run the following command to start the server:

```
uvicorn main:app --host 0.0.0.0 --port 8000 --reload
```

- You should see a message like the image above

2. Start the Frontend Server



```
frontend — node · npm start TERM_PROGRAM=Apple_Terminal SHELL=/b...
Compiled successfully!

You can now view frontend in the browser.

Local:      http://localhost:3000
On Your Network:  http://172.19.248.64:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
█
```

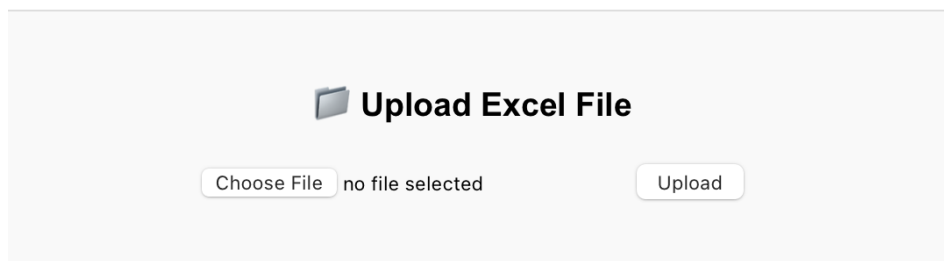
- Open a new terminal and go to the frontend directory.
- Start the frontend server with the command:

npm start

- The frontend should compile successfully and show the message above

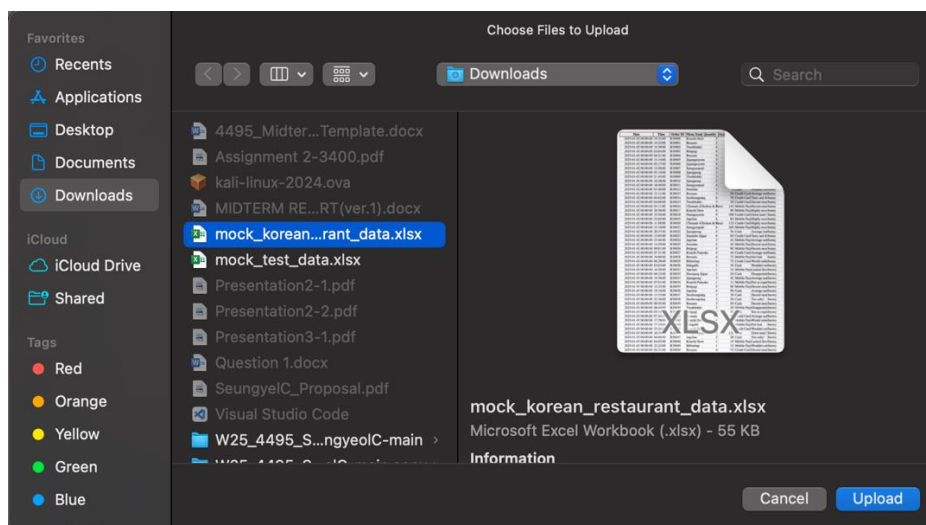
3. Upload Excel File

Intelligent Business Analytics



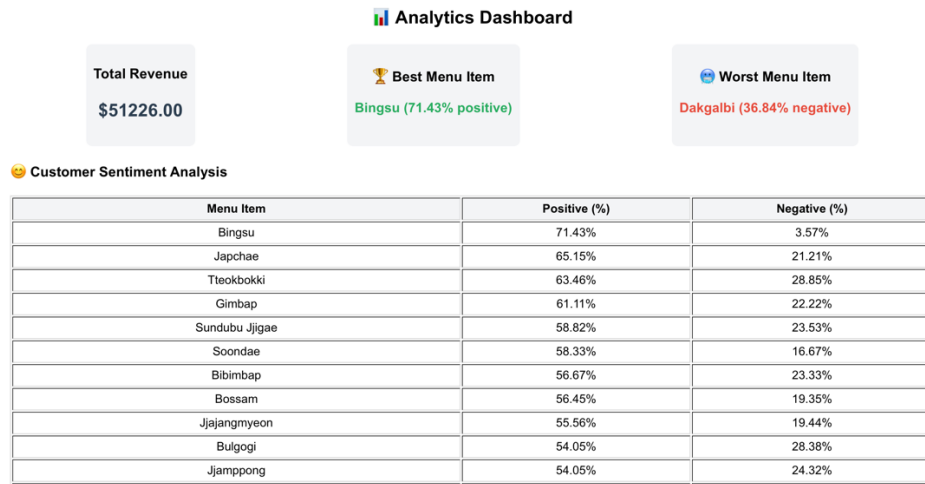
- Open your browser and go to **http://localhost:3000**.
- You will see the upload interface:
- Click on **Choose File** and select the sample data file.

4. Select Sample Data File



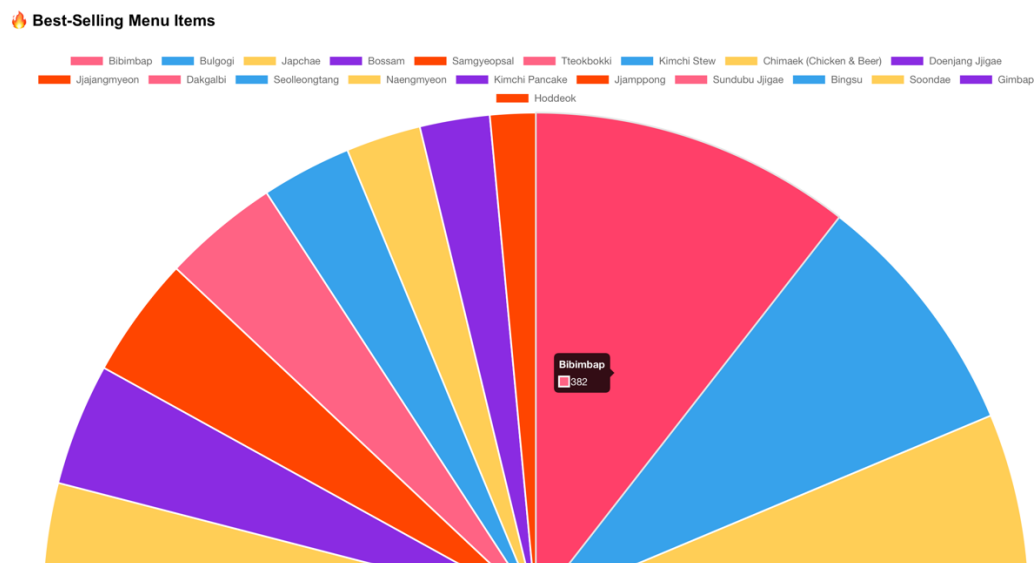
- In the file selection dialog, select **mock_korean_restaurant_data.xlsx**

5. View Analytics Dashboard



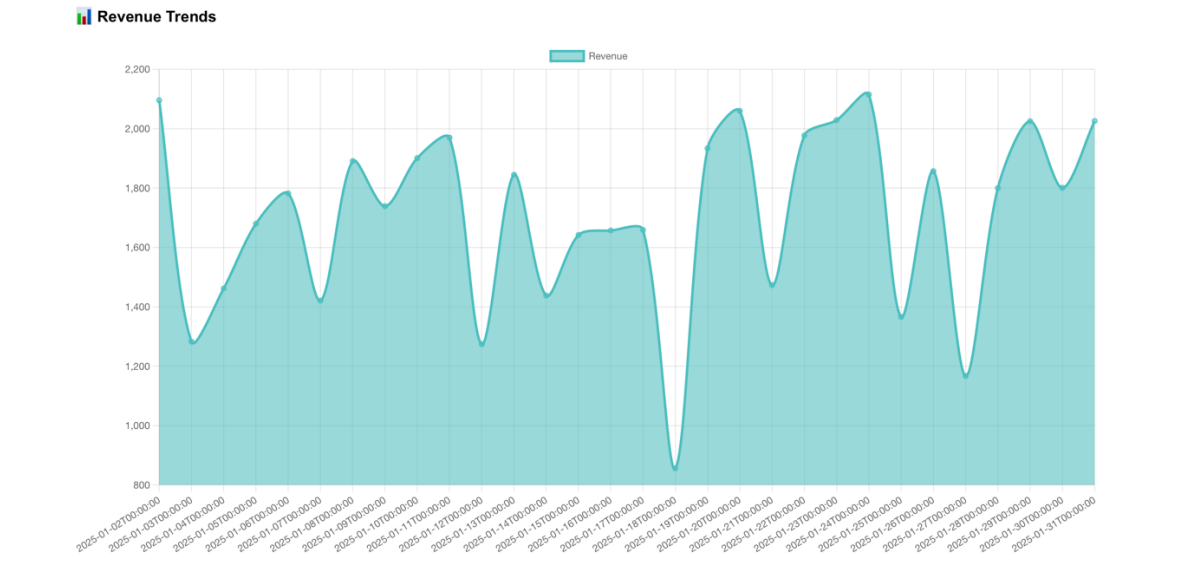
- After uploading, the dashboard will display analytics data:
- You can see **Total Revenue**, **Best Menu Item**, and **Worst Menu Item**.

6. Best-Selling Menu Items Visualization



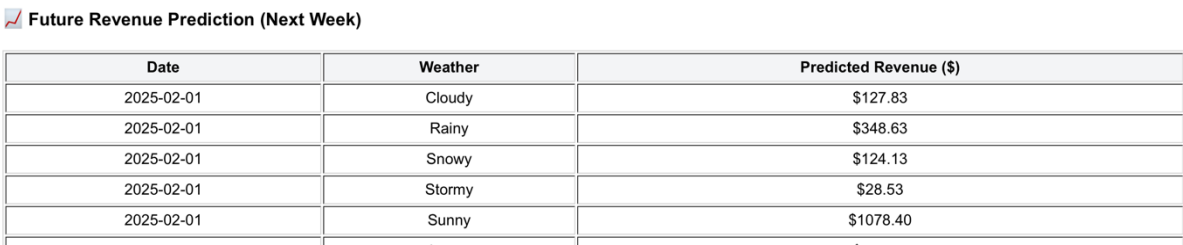
- The **Best-Selling Menu Items** are displayed using a pie chart:
- Hover over the sections to see the quantity of items sold.

7. Analyze Revenue Trends



- The **Revenue Trends** chart shows daily sales data:
- This chart helps identify peak sales periods and trends.

8. Future Revenue Prediction

The table, titled "Future Revenue Prediction (Next Week)", lists predicted revenue for five different weather scenarios on 2025-02-01. The predicted revenue values are \$127.83 for Cloudy, \$348.63 for Rainy, \$124.13 for Snowy, \$28.53 for Stormy, and \$1078.40 for Sunny.

Date	Weather	Predicted Revenue (\$)
2025-02-01	Cloudy	\$127.83
2025-02-01	Rainy	\$348.63
2025-02-01	Snowy	\$124.13
2025-02-01	Stormy	\$28.53
2025-02-01	Sunny	\$1078.40

- The **Future Revenue Prediction** table provides expected revenue based on weather conditions:
- Use this information for planning and inventory management.

Summary of Initially Proposed Research Project

Objectives

The **Intelligent Business Analytics System** was initially proposed to achieve the following:

- **Design a system capable of processing and validating input data from Excel files.**
- **Analyze and predict revenue trends using historical and real-time data.**
- **Explore the impact of weather data on revenue generation.**
- **Perform sentiment analysis on customer feedback to extract actionable insights.**
- **Present findings through a user-friendly, interactive dashboard.**

Methodology

Data Collection

- **Sources:** Real-world Excel files simulating various business scenarios (growth, stability, decline).
- **Inputs:** Customer feedback, weather data, and revenue trends.

Analysis Techniques

- **Revenue and Profit Analysis:** Utilizing **Python's Pandas and NumPy** for data transformation and trend analysis.
- **Weather Impact Analysis:** Correlation modeling to identify trends between sales and weather conditions.

- **Text Analytics:** Employing **NLTK and SpaCy** for sentiment analysis and customer feedback classification.

Technology Stack

Backend Development

- **Programming Language:** Python (chosen for its flexibility in analytics and machine learning).
- **Frameworks:** FastAPI (for API development) and Flask (for microservices).
- **Database Management:** PostgreSQL (for structured data) and MongoDB (for unstructured customer feedback).
- **Data Processing Libraries:** Pandas, NumPy, NLTK, Scikit-learn.
- **Visualization Libraries:** Matplotlib, Seaborn, Plotly.

Frontend Development

- **Framework:** React.js (chosen for interactive dashboard design and state management).
- **Visualization:** Chart.js and Plotly for data-driven insights.

Expected Results

- **Automated Data Validation:** The system will identify errors in input files before processing.
- **Real-Time Data Visualization:** Interactive dashboards for revenue forecasting, customer feedback, and weather analysis.
- **Actionable Insights:** Predictive recommendations for inventory optimization, pricing strategies, and marketing adjustments.

Changes to the Proposal

Adjustments Made

1. Migration to React.js for Frontend Development

- Initially planned for a simpler frontend, but migrated to **React.js** for improved flexibility and maintainability.
- Implemented **Redux** for state management, allowing a seamless data flow across components.

2. Enhanced Sentiment Analysis with NLTK

- Originally planned for basic sentiment scoring but expanded to include **lemmatization, stop-word removal, and sentiment classification** using **NLTK and SpaCy**.

3. Live Weather Data Integration

- Shifted from using static weather data to **real-time weather API** integration for more dynamic forecasting.

4. More Interactive Dashboards

- Improved **Chart.js and Plotly** integration to allow **filtering, dynamic visualization, and enhanced user interaction**.

Project Planning and Timeline

Updated Timeline

Phase	Milestone	Timeline
Research & Setup	Collect datasets, initialize repo	Week 1-2
Data Validation	Implement file validation module	Week 3-4
Backend Development	Implement API, database schema	Week 5-6
Analytics Module	Develop revenue, weather, and sentiment analysis modules	Week 7-8
Visualization	Develop interactive charts & reports	Week 9-10
Testing & Refinement	Debugging, improving model accuracy	Week 11-12
Final Delivery	Finalize project, documentation, and submission	Week 13-14

Project Management Approach

- **Agile Development** with iterative improvements.
- **Kanban Board** to track tasks and milestones.

Implemented Features

1. **GitHub Repository & Code Structuring**

- Implemented a clear project structure with distinct folders for API, services, and utilities.
- Configured CI/CD using **GitHub Actions** for automatic testing.

2. **Backend API Development**

- Developed FastAPI and Flask-based API endpoints for **data processing and analysis**.
- Integrated **PostgreSQL database schema** optimized for query performance.

3. **Sentiment Analysis Implementation**

- Developed an **NLTK-based NLP module** for customer feedback processing.
- Applied **lemmatization and stop-word filtering** to improve sentiment classification accuracy.

4. **Data Visualization Enhancements**

- Used **Chart.js and Plotly** to create interactive charts.
- Improved dashboard UI with **filtering and real-time updates**.

Work Logs

Date	Hours	Description of Work Done
Jan 25, 2025	3	Initialized project repository and structured directories.
Jan 27, 2025	6	Designed system architecture, set up FastAPI, PostgreSQL, and tested Excel upload feature.
Jan 30, 2025	4	Implemented data preprocessing, frontend migration to React.js.
Feb 1, 2025	6	Developed API endpoints, structured backend, and created initial data visualizations.
Feb 3, 2025	5	Refactored API, modified data handling for better results, started sentiment analysis.
Feb 5, 2025	6	Implemented and debugged sentiment analysis using NLTK, optimized data visualization.
Feb 8, 2025	3	Updated README.md with improved structure and content.
Feb 12, 2025	4	Updated backend with improved revenue prediction with using Facebook Prophet
Feb 14, 2025	3	Fixed sentiment analysis, revenue prediction, and Chart.js errors. Improved frontend Dashboard sorting
Feb 21, 2025	5	Implemented advanced insights and predictive analytics in Dashboard.js. Enhanced frontend with improved data visualization, including future revenue predictions, sentiment analysis, and interactive charts. Integrated with the enriched / insights API endpoint for deeper business insights.

Closing and References

Acknowledgments

Special thanks to instructor for valuable feedback and guidance throughout the development process.

References

1. Python Documentation for Pandas and NumPy
 2. NLP Techniques from NLTK and SpaCy
 3. Research on Weather Analytics and Retail Forecasting
 4. Case Studies on Business Intelligence and Predictive Modeling
-