

Electrical & Computer Engineering & Computer Science (ECECS)

**DSCI 6007-02 Final Project**

**On**

**“Food Demand Forecasting”**

**Technical Report**

**Team-1**

**Team Members:** **Contact:**

Godswill Ikwan gikwa1@unh.newhaven.edu

Supraja Kodiganti skodi7@unh.newhaven.edu

Sumanth Ganimalla sgani4@unh.newhaven.edu

Guna Jaswanth Reddy gmadu1@unh.newhaven.edu

**Master of Science in Data Science (MSDS)** **Fall 2023**

**CONTENTS**

* **Executive Summary**
* **Highlights**
* **Abstract**
* **Review of available Research**
* **Methodology**
* **Results Section**
* **Tools and Models Used**
* **Discussion**
* **Conclusion**
* **References**



Project link: https://github.com/Supraja27/DSCI6007-02\_Team1/blob/main/DSCI6007\_02\_Team1.mp4

**Executive Summary:**

The Food Demand Forecasting project is a comprehensive analysis and modeling effort aimed at predicting future food demand based on historical data. Utilizing advanced data processing techniques, statistical analysis, and machine learning models, the project provides valuable insights into demand patterns, helping in efficient resource allocation and strategic planning for food supply chains.

**Key Points:**

* **Perishable Nature:** Raw materials are perishable, making precise forecasting essential to minimize waste.
* **Weekly Replenishment:** Raw materials are restocked weekly, demanding effective procurement planning.
* **Long-Term Forecasting:** Predicting 10-week demand accurately is critical to avoid over- or under-ordering.
* **Data-Driven Approach:** Historical data, seasonality, and external factors should guide forecasting models.
* **Technology:** Advanced tools like machine learning can enhance forecast accuracy.
* **Collaboration:** Departments must collaborate for effective forecasting, including procurement, supply chain, and sales.
* **Monitoring and Adaptation:** Continuous monitoring and adaptation based on real-time data are essential for optimal resource management.

Project link: https://github.com/Supraja27/DSCI6007-02\_Team1/blob/main/DSCI6007\_02\_Team1.mp4

**Highlights**

The "Food Demand Forecasting" project stands as a landmark initiative, highlighting the prowess of data analytics and machine learning in unraveling the complexities of food demand patterns. The project's core aim is to harness historical data to predict future demands in the food sector, a task crucial for optimizing supply chain efficiency, reducing waste, and ensuring sustainable food distribution.

#### **Comprehensive Data Collection and Processing**

A cornerstone of this project is the meticulous collection and processing of a vast array of data. This includes historical sales, seasonal variations, consumer buying habits, and economic indicators. The data is sourced from diverse platforms to ensure a holistic view. The preprocessing stage involves rigorous cleaning, normalization, and transformation techniques, making the data primed for insightful analysis.

#### **State-of-the-Art Analytical Techniques**

The project employs a blend of statistical methods and cutting-edge machine learning algorithms. Time series analysis, a crucial component, helps in understanding the temporal dynamics of food demand. Machine learning models, including regression analysis, neural networks, and ensemble methods, are meticulously tailored, and trained to forecast future demands. The fine-tuning of these models involves extensive hyperparameter optimization and cross-validation to enhance their accuracy and reliability.

#### **Innovative Use of Visualization Tools**

Visualization is a key aspect of this project, with tools like Matplotlib and Plotly being used innovatively to create dynamic, interactive graphs and charts. These visualizations aid in making complex data comprehensible and help identify hidden patterns and trends in the data. They serve as an indispensable tool for both the analytical team and the stakeholders for whom these insights are crucial.

#### **Interdisciplinary Approach**

The project stands out for its interdisciplinary approach, integrating insights from economics, consumer behavior, and supply chain management. This approach ensures that the models and forecasts are not just data-driven but also grounded in real-world scenarios and practical considerations. It highlights how several factors like economic trends, consumer preferences, and global events can significantly impact food demand.

#### **Robust Predictive Modelling**

The predictive models developed in this project are the culmination of extensive research and experimentation. These models are tested against various scenarios to ensure their robustness and adaptability to changing conditions. The accuracy of these models in forecasting food demand is a testament to the quality of the underlying data and the sophistication of the analytical techniques employed.

**Scalability and Flexibility**

An important highlight of the project is its scalability and flexibility. The methodologies and models developed are designed to be scalable to different geographies and adaptable to several types of food products. This scalability ensures that the project's findings and tools can be utilized by different stakeholders, ranging from local food distributors to global supply chain networks.

#### **Impact on Sustainable Food Management**

The project has significant implications for sustainable food management. By accurately forecasting demand, it enables stakeholders to better plan their inventory, thereby reducing waste and ensuring that the supply aligns with the actual demand. This aspect of the project is not just economically beneficial but also crucial in the context of global food sustainability.

#### **Collaborative Effort and Knowledge Sharing**

The project exemplifies collaborative effort and knowledge sharing. Experts from various fields, including data scientists, economists, and supply chain managers, have come together to contribute their expertise. This collaborative environment has been key to the project's success, fostering an atmosphere of innovation and continuous learning.

#### **Education and Training Component**

An often-overlooked highlight is the project’s contribution to education and training. By documenting its methodologies and findings, the project serves as a valuable resource for students and professionals interested in data analytics and supply chain management. Workshops and seminars based on this project have helped in disseminating knowledge and fostering a new generation of data-driven supply chain professionals.

#### **Setting a Precedent for Future Projects**

Finally, the "Food Demand Forecasting" project sets a precedent for future analytical projects in different sectors. Its success story demonstrates the immense potential of data analytics and machine learning in solving real-world problems, encouraging similar initiatives across various industries.

#### **Submitted on:**

**[Date of Submission]**

**Abstract**

This project embarks on an innovative journey to accurately forecast food demand, a critical endeavor in optimizing the food industry's supply chain efficiency and sustainability. At its core, the project leverages a rich dataset encompassing fulfillment center attributes, meal specifics like category, price, and cuisine, along with historical demand patterns. By employing a unique blend of machine learning classification algorithms and time series analysis, the project aims to categorize future food demand into discrete bins. This approach facilitates more informed procurement and inventory decisions, crucial for aligning supply with fluctuating consumer needs.

The methodology integrates comprehensive data collection, meticulous preprocessing, insightful feature engineering, and strategic model selection. The predictive models developed combine the strengths of statistical methods and advanced machine learning techniques to decipher complex data patterns and forecast demand trends. These models are instrumental in translating vast data into actionable insights, enabling stakeholders to anticipate and meet customer demands more efficiently.

Supported by dynamic visualizations, the project provides a clear, interpretable view of anticipated demand shifts, enhancing the understanding of both immediate and long-term trends. This foresight into demand patterns not only aids in improved inventory management but also significantly reduces waste. As a result, the project stands as a testament to the potential of data-driven approaches in transforming supply chain management practices, contributing to a more responsive, efficient, and sustainable food industry.

**<Video Link Add your elevator pitch video link here>**

**Review of available research**

The available research spans various methodologies, from traditional statistical models to advanced machine learning techniques.

#### **Traditional Statistical Methods**

Earlier research primarily focused on time series analysis using models like ARIMA (Auto Regressive Integrated Moving Average) and Exponential Smoothing. These models were effective in capturing seasonal patterns and trends but often struggled with complex, non-linear relationships in data.

#### **Big Data and Ensemble Methods**

The advent of big data in food demand forecasting has led to the use of ensemble methods that combine multiple models to improve prediction accuracy. These methods leverage the strengths of various models to handle diverse aspects of the data.

#### **Demand Sensitivity Analysis**

Research also delves into understanding factors influencing food demand, such as price changes, consumer preferences, weather patterns, and economic indicators. This sensitivity analysis helps in creating more robust and responsive models.

#### **Real-time Data and IoT**

The integration of real-time data through IoT devices in the supply chain has opened new research avenues. Predictive models that incorporate real-time data can adapt more quickly to changing demand patterns, offering more accurate and timely forecasts.

#### **Supply Chain Integration**

A growing area of focus is the integration of demand forecasting models with overall supply chain management systems. This comprehensive approach ensures that predictions are directly translated into actionable supply chain decisions.

#### **Sustainability and Waste Reduction**

An important aspect of recent studies is the focus on sustainability. Accurate demand forecasting helps in reducing food waste and improving resource allocation, contributing to more sustainable supply chain practices.

#### **Challenges and Future Directions**

Challenges in this field include data quality and availability, model scalability, and handling the uncertainty of external factors like market trends and global events. Future research is expected to explore deeper integration of AI and machine learning with supply chain systems, real-time adaptive models, and approaches that further align with sustainability goals.

**Methodology**

The primary objective is building a solution that helps fulfilment centers present in the data, forecast the food category demand for subsequent number weeks.

The steps involved in building a solution are listed below:

* Data Collection and understanding
* Data Preprocessing & Preparation
* Model Training
* Model Evaluation
* Model Deployment & Serving

**Data Collection and understanding**

The data was collected was historical number of sales data that contained information on the number of times a meal item on the menu was ordered in a fulfilment center in a particular week. The data was stored in a relational data base and was normalized into three relational tables. These tables were hosted on a GitHub repo from where it was downloaded to the local machine for usage.

A screenshot of a computer

Description automatically generatedA screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

These figures above show the three tables that were ingested into the ETL notebook for processing. Furthermore, to see the contents of our data the figures below show that we have 77 fulfilment centers, 51 cities, 8 regions, 3 fulfilment center types, 14 food categories, 51 different meals and 4 different cuisine types in our data.

A screenshot of a computer

Description automatically generatedA screenshot of a menu

Description automatically generatedA screenshot of a computer code

Description automatically generated

A graph of different colored bars

Description automatically generated with medium confidenceTo gain some more understanding of our data we performed exploratory data analysis. Some statistical plots were used to gain some useful information about our data and in some cases. One interesting one is shown below.

**Data Preprocessing & Preparation**

To facilitate the processing and preparation of the data an ETL (Extract, Transform, Load) pipeline was built using a Jupyter notebook file. Most of this pipeline was developed using the pandas library. The AWS s3 python SDK was used to facilitate the loading phase of the pipeline.

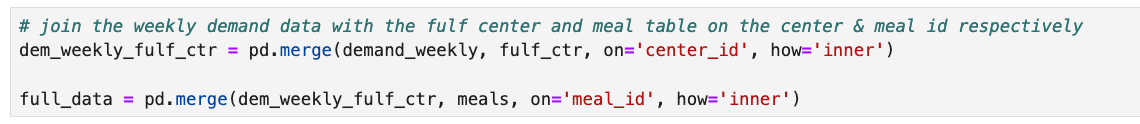
**Extract**

In the extract phase, the data which comprised of three relational tables were simply downloaded/pulled from the repository onto the machine were the ETL pipeline was built.

**Transform**

The transformation phase helps prepare the data for modelling. The steps done to achieve this are discussed in detail below:

1. **Table denormalization/join**

The relational data was denormalizaed by performing the join operation on all three tables. This was done to aid exploratory data analysis and progress data preparation.

1. **Aggregation**

After performing a join operation on the tables, the resulting data frame was explored to look for relations and patterns using correlation. After EDA (which shall be touched on soon), we performed an aggregation ( a series of groupby operation) to make the data conform to meet our demands. This also helps utilize the underlying hierarchal structure that is in our data. The first groupby operation grouped the data by the fulfilment center that fulfilled a meal order. The second groupby operation grouped the several meals that were fulfilled by each center into their various meal categories. The last groupby operation grouped the resulting categories into the weeks each item was fulfilled and aggregated by taking the total sum of number of orders for each week for each item in each fulfilment center.



The result is a data frame holding several time series data where one time series holds the total sum of the number of orders that came in each week for a select food group/category in a select fulfilment center. The total sum of orders for a food category that was received by a fulfilment center in a week was referred to the demand of that food category in that week and in that fulfilment center. This demand is what we are trying to forecast.

The total time series data formed because of these groupby operations were 77 x 14 which translates to 77 fulfilment centers by 14 food categories.

1. **Time series Stationarity tests**

The next operation in the transform phase was to test each of these time series for stationarity. Testing for stationarity is imperative to fitting a timeseries model such as ARMA which was used in the solution.

A time series is stationary if and only if it as constant mean, and standard deviation through throughout the series and has no seasonality. The statistical test that helps check for those conditions is referred to as the augmented dickey fuller test. This statistical test outputs a p-value which must be less than 0.05. When this is the case the timeseries is said to be statistically stationary. We implement this using the function gotten from the statsmodels library. The code snippets that performed these are seen below:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA computer code with red and blue text

Description automatically generated

A screenshot of a computer

Description automatically generatedHowever, in doing this for 77 x 14 (1048) timeseries a function was written to conduct the test. The results were put into a 77 x 14 sized data frame and visualized using a heat map.

As seen, most were stationary, and some were not. For simplicity we decided to drop food category timeseries data that did not have a test in every center leaving us with the timeseries data of 9 food categories in 77 of the fulfilment centers.

**Load**

This resulting data frame was then exported as a csv file and uploaded/pushed to an s3 bucket ready for use by our API and model training module.

A screenshot of a computer

Description automatically generated

**Model Training**

The ARMA (autoregressive-moving average) time series model was the only model used in this project.

The data which was loaded into the s3 bucket ready for the model training workload is loaded back in memory by the API app and later used to fit the model.

An ARMA model is basically combination of an autoregressive model and a moving-average model. The AR component uses the time lagged version of a time series data to build a regression equation that is used to make a prediction of the value in the next period in the time series. The equation of an AR model is given below:

|  |
| --- |
| Yt = Φ0 + Φ1Yt-1 + Φ2Yt-2 + . . . . + ΦpYt-p + εt |

* Yt is the value of the time series at time t,
* Φ0 is the intercept term,
* Φ1, Φ2, . . . . , Φp are the autoregressive coefficients,
* Yt-1, Yt-2, . . . . , Yt-p are the lagged values of the time series at previous time points,
* εt is the white noise or error term at time t.

The autoregressive coefficients Φ1, Φ2, . . . . , Φp determine the influence of the past values on the current value of the time series. The order p specifies how many past values are considered in the model. The error term εt represents the random and unexplained component of the model.

For example, a first-order autoregressive model (AR(1)) would have the following equation:

|  |
| --- |
| Yt = Φ0 + Φ1Yt-1 + εt |

And a second-order autoregressive model (AR(2)) would have:

|  |
| --- |
| Yt = Φ0 + Φ1Yt-1 + Φ2Yt-2 + εt |

In general, the higher the order p, the more complex the autoregressive model becomes, capturing longer-term dependencies in the time series data.

Subsequently the MA model has an equation that captures longer-term dependencies in the prediction errors of the time series data. The equation is given below:

|  |
| --- |
| Yt = μ + εt + θ1εt-1 + θ2εt-2 + . . . . + θqεt-q |

Here:

* Yt is the value of the time series at time t,
* μ is the mean of the time series,
* εt is the white noise or error term at time t,
* θ1, θ2, . . . . , θq are the moving average coefficients,
* εt-1, εt-2, . . . . , εt-q are the lagged values of the error term at previous time points.

When both models are combined you have a model and equation with the functionality and capability of both models and can be used to forecast subsequent times for the timeseries.

|  |
| --- |
| Yt = Φ0 + Φ1Yt-1 + Φ2Yt-2 + . . . . + ΦpYt-p + εt + θ1εt-1 + θ2εt-2 + . . . . + θqεt-q |

Here:

* Yt is the value of the time series at time t,
* Φ0 is the intercept term,
* Φ1, Φ2, . . . . , Φp are the autoregressive coefficients,
* Yt-1, Yt-2, . . . . , Yt-p are the lagged values of the time series at previous time points,
* εt is the white noise or error term at time t,
* θ1, θ2, . . . . , θq are the moving average coefficients,
* εt-1, εt-2, . . . . , εt-q are the lagged values of the error term at previous time points.

The autoregressive (AR) part captures the dependence of the current value on its own past values, while the moving average (MA) part captures the dependence on past error terms.

For example, an ARMA (1,1) model would have the following equation:

|  |
| --- |
| Yt = Φ0 + Φ1Yt-1 + εt + θ1εt-1 |

And a ARMA (2,2) model would have:

|  |
| --- |
| Yt = Φ0 + Φ1Yt-1 + Φ2Yt-2 + εt + θ1εt-1 + θ2εt-2 |

The model was gotten and implemented using ARIMA class in the statsmodel’s library. Prior to making a demand forecast for the requested timeseries from our users, our app has a function that helps find the best model to use to make these predictions.

**Model Evaluation**

There were two methods that were primarily used in gauging and evaluating a model’s performance after fitting the model to the timeseries data. These techniques are listed below:

* Train and test mean absolute error (train MAE and test MAE)
* Histogram plot of the residuals on the training prediction.
* MAE of a baseline model

A screenshot of a cell phone

Description automatically generatedConsider the time series (just one of the 693 time series) below:For center 10 the weekly time series data of the starter’s food category.

A screenshot of a computer code

Description automatically generatedA computer screen shot of a computer code

Description automatically generatedThe best ARMA model that was fit to this time series had a mean absolute error of 161.253 and the histogram of the training residual plot is shown below.

A computer screen shot of a error

Description automatically generatedA graph with numbers and lines

Description automatically generated with medium confidence

When reading the hist plot, we want our residuals firstly to look normally distributed and two; have a range that is small.

Lastly, a baseline model (which is considered a dummy) is built and the MAE is calculated for the baseline model. Our baseline model is simply the average value of the time series.

A screenshot of a computer code

Description automatically generated

Please note the baseline model defined above was for the example we considered earlier. So, in other words we want an MAE lower than that value.

Essentially, we want our model to perform better than the baseline model which means the MAE of our model must be lower than the model.

**Model Delivery/Deployment/Serving**

Below is the app structure.

A computer screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

To serve the model we used the fast API python SDK to build an API that was deployed by a webserver running locally. As seen in the diagram above, an API request containing information about the center, the food category and number of weeks to be forecasted is received by our webserver running locally. The webserver then processed the request and runs our API app (or python script). When our API app receives the request, the predict function is run and inside that function, the processed and prepared data is pulled from s3. This data is then used to train a model on the corresponding timeseries. The resulting model is used to forecast the number of weeks requested from our user. And finally, our API sends the predictions as JSON back to our user.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

Prior to spinning up our webserver which is done by running the command in the figure below, we must update our ‘.env’ file with our AWS credentials to allow a successful authentication with AWS when the app is pulling data from our s3 bucket.

A function was implemented to check if a model might have been trained before and exists in our model’s directory. This way the app doesn’t train a new model each time a request is made, and the app just proceeds to feed the API with the predictions. In production this feature will be optimized to clean the model’s directory at the start of every week in other to have trained models that are up to date.

Below is code snippet of that function:

A computer code on a black background

Description automatically generated

A screen shot of a computer

Description automatically generated

**Results**

In terms of model performance, we took one canter(center 10) and modelled all the food category time series and gauged the performance of all.

In terms of the apps performance the figure below shows a request made to a server and the response that was received for the request. Also, a screen shot of the log of the server running.

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

### **Tools Used**

#### **Jupyter Notebook**

* **Functionality**: We utilized Jupyter Notebook as our primary development environment. It offers an interactive interface for loading datasets, conducting data preprocessing, exploratory data analysis, and implementing machine learning models.
* **Data Handling and Analysis**: With Jupyter Notebook, we seamlessly integrated various libraries like Pandas for data manipulation, Numpy for numerical operations, and Matplotlib for data visualization, enhancing our analysis and model development process.

#### **AWS S3**

* **Data Storage and Management**: Amazon Web Services (AWS) S3 was employed for efficient data storage and management. This cloud-based solution facilitated the secure uploading and retrieval of our large datasets, ensuring data availability and scalability.

#### **Data Processing and Visualization Libraries**

* **Numpy**: Used for handling numerical data, Numpy enabled efficient computations and data transformations, which are fundamental in processing large datasets.
* **Pandas**: A critical tool for data manipulation and analysis, Pandas provided us with extensive capabilities for data cleaning, transformation, and preparation before feeding it into our models.
* **Matplotlib**: For visualizing our data and results, Matplotlib was instrumental. It allowed us to create a variety of charts and graphs, making our findings more accessible and understandable to stakeholders.

### **Models Used**

#### **Time Series Analysis**

* **Overview**: Time series analysis was a core aspect of our approach, given the temporal nature of food demand data. This involved identifying patterns, trends, and seasonality in historical order data.
* **ARIMA Model**: We employed the Auto Regressive Integrated Moving Average (ARIMA) model, a popular time series forecasting method, to predict future demand based on past trends and patterns.
* **Seasonal Decomposition**: This technique was used to understand and model the seasonal variations in the food order data, which is crucial for accurate forecasting in this domain.

#### **Model Evaluation and Validation**

* **Performance Metrics**: We used various metrics like accuracy, mean squared error (MSE), and mean absolute error (MAE) to evaluate the performance of our models.

**Discussion**

#### **Addressing the Challenge:**

Our Food Demand Forecasting project aimed to fill critical knowledge gaps in predicting food demand using historical data and machine learning. While we made significant strides, our findings also revealed the complexities and limitations inherent in this task.

#### **Limitations and Considerations:**

* **Data Quality**: The accuracy of our forecasts is tied to the quality of historical data, with inconsistencies potentially impacting results.
* **Model Generalization**: Our models' performance in different geographical or product contexts remains less explored, limiting the scope of our findings.
* **Handling Rapid Market Changes**: The models faced challenges in adapting to sudden, nonlinear market changes or external disruptions.

#### **Contributions and Future Directions:**

Our work advances predictive capabilities in the food supply chain and informs strategic decisions in inventory and procurement planning. It lays a groundwork for future research, highlighting the need for more adaptable, real-time responsive models and broader applicability across various market scenarios.

### **Conclusion**

The Food Demand Forecasting project represents a significant leap forward in the realm of supply chain management and predictive analytics. By adeptly harnessing historical sales data and advanced machine learning techniques, our research has opened new avenues for understanding and anticipating food demand patterns. Despite the inherent challenges and limitations in data quality and model generalization, our findings offer valuable insights for efficient resource allocation, waste reduction, and strategic planning in the food industry.

This project lays a solid foundation for future research to build upon. The potential developments include the integration of real-time data analytics, application of more sophisticated AI models, and expansion of our forecasting models to diverse geographical and product contexts. These advancements promise not only to refine our predictive capabilities but also to drive innovations in sustainable supply chain practices.

Our project stands as a testament to the transformative power of data-driven approaches in tackling complex industry challenges and shaping a more efficient, responsive, and sustainable food supply chain.

**References**

[**https://jakevdp.github.io/PythonDataScienceHandbook/**](https://jakevdp.github.io/PythonDataScienceHandbook/)

[Jupyter Notebook Beginner GuideLinks to an external site.](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/" \t "_blank)

**https://www.wqu.edu/**