FBEM FORECASTING REPORT

ABSTRACT

The Food Bank of Eastern Michigan (FBEM) requires supply chain planning tools to enhance efficiency. A mathematical modeling tool will forecast operational volume to improve food distribution to agencies serving those in need.

Osama Karim Kidwai

Larry Navarre

Nimalan Manoharan

Prakash Pole

Contents

Introduction

The Food Bank of Eastern Michigan (FBEM), headquartered in Flint, Michigan, is a vital organization addressing food insecurity across 23 counties. Partnering with food pantries, soup kitchens, shelters, and schools, FBEM distributes donated food to communities in need, leveraging resources from individuals, businesses, manufacturers, and government programs to provide millions of meals annually. Our project aims to analyze the donations received by FBEM and their distribution across its service area, focusing on key aspects such as identifying counties that receive more food and understanding the reasons behind these disparities. Additionally, the analysis examines donation patterns, supply chain efficiency in delivering goods on time, and the potential for excessive food delivery leading to waste. To address these challenges, FBEM requires advanced supply chain planning tools to enhance operational efficiency. As part of this project, we are developing a mathematical modeling tool to forecast operational volumes, enabling FBEM to optimize food distribution to agencies and reduce waste. This tool will provide actionable insights to help FBEM improve the balance and efficiency of its food distribution network, ensuring better service to the communities it supports.

Data Overview

The dataset for this project was sourced directly from the Food Bank of Eastern Michigan (FBEM) and organized to align with the analytical goals. The data spans from 2019 to 2022 and includes 380,047 rows, with the following key columns:

- **DateKey**: Represents the date of food distribution, used for analyzing temporal trends and forecasting future patterns.
- **Agency**: Lists the partnered agencies through which FBEM delivers food, enabling an understanding of the distribution network.
- **County**: Indicates the counties receiving food, allowing for a geographic analysis of food distribution.
- **Item Gross Weight**: Specifies the total weight of delivered items in pounds, serving as a measure of distribution volume.
- **UNC Package Type**: Describes the type of package donated, such as loose items or bulk boxes, which aids in supply chain analysis.
- **UNC Product Category**: Categorizes where the products are donated from or purchased (e.g., retail, wholesale, government programs).
- **UNC Product Type**: Represents unique categories of food, such as rice, fresh fruits, vegetables, etc., offering granular insights into food distribution.

The dataset provides a comprehensive overview of FBEM's operations, facilitating an in-depth analysis of donation trends, distribution efficiency, and county-level disparities. Additionally, the data was used to forecast food distribution volumes for 2023, aiding in supply chain planning and optimization.

Methodology

Data Organization

The data was obtained from the Food Bank of Eastern Michigan (FBEM) and organized into the following columns:

- **DateKey**: Contains the date in both year-month and year-week formats for monthly and weekly forecasts.
- **Agency**: Partnered agencies responsible for delivering the food.
- **County**: The counties where food was delivered.
- **Item Gross Weight**: Total weight of items delivered, measured in pounds.
- **UNC Package Type**: The type of package donated, e.g., loose items or bulk boxes.
- **UNC Product Category**: Specifies the source or nature of the donation, e.g., purchased or donated.
- **UNC Product Type**: Unique food categories, such as rice or fresh vegetables.

The dataset spans from **2019 to 2022**, and a forecast for **2023** was generated. The total dataset contained **380,047 rows**.

After organizing the data, a new column was created to classify each product type as **FTE (Food to Encourage)** or **non-FTE**, based on government guidelines prioritizing basic and healthy foods. This categorization allows management to focus on these essential items when analyzing distribution patterns.

Pivot Table and Model Setup in Excel

To ensure user-friendly analysis for FBEM management, **pivot tables** were created for different levels of analysis:

1. **FTE**: Summarizing the delivery weight for encouraged foods.
2. **Yearly**: A yearly breakdown of delivery trends.
3. **Monthly and Weekly**: Used for detailed time-series forecasting.

The pivot tables allow users to drag, drop, and filter by specific attributes, such as year, product type, or FTE classification. While **Excel** was used to meet the organization's current capabilities, a Python-based forecasting model was also developed for future use.

How the Forecasting Model Operates

The forecasting model in Excel relies on **Winters' Seasonal Method**, chosen for its ability to handle both trend and seasonality. The following steps explain how the model operates:

1. **Data Input**:
    - Users select a category (e.g., a specific product type or all products) from the pivot table.
    - The pivot table provides aggregated data, such as the total weight delivered per time period.
2. **Forecast Table Setup**:
   The model uses the following columns:
    - **Delivered**: Actual weight delivered.
    - **Level (Lt)**: Represents the base level of demand.
    - **Trend (Tt)**: Captures the rate of increase or decrease in demand over time.
    - **Seasonal Factor (St)**: Adjusts for recurring patterns (e.g., holiday surges).
    - **Forecast (Ft)**: Predicted weight for the next time period.
3. **Winters' Formula**:
   The forecast is calculated using the following equations:

- **Level Equation**:

$$l_t = \alpha \frac{y_t}{s_{t-p}} + (1-\alpha)(l_{t-1} + b_{t-1})$$

- **Trend Equation**:

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

- **Seasonal Equation**:

$$s_t = \gamma \frac{y_t}{l_t} + (1-\gamma)s_{t-p}$$

- **Forecast Equation**:

$$F_{t+h} = (l_t + hb_t) \cdot s_{t+h-p(k)}$$

Where:

- y_t: Actual value at time t.
- l_t: Level (smoothed average) at time t.
- b_t: Trend (rate of change) at time t.
- s_t: Seasonal component at time t.
- F_{t+h}: Forecast for h steps ahead.
- p: Number of periods in a season.
- k: Number of full seasons ahead, calculated as $k = \lfloor h/p \rfloor$.
- α, β, γ: Smoothing constants for level, trend, and seasonality, respectively, where $0 < \alpha, \beta, \gamma < 1$.

4. **Optimization Using Solver**:
   - The model includes **Mean Squared Error (MSE)** to evaluate forecast accuracy.
   - Users run Excel's **Solver** tool to minimize the MSE by adjusting the smoothing factors (α,β,γ\alpha, \beta, \gammaα,β,γ) and recalibrating the **Level, Trend, and Seasonal Factors**.
   - **GRG Non-Linear Solver** is used for optimization, ensuring the most accurate forecast by iteratively refining the components.
5. **Visualization**:
   - A line graph is automatically updated, displaying both **actual deliveries** and the **forecasted values** over time.
   - This visual comparison allows management to assess forecast accuracy and identify trends.

Advantages of Winters' Method for FBEM

- **Handles Complex Patterns**: Incorporates both trend and seasonality, essential for predicting food distribution needs.
- **Adaptable and Scalable**: Can adjust dynamically as new data is added, keeping forecasts up-to-date.
- **User-Friendly in Excel**: Provides a familiar interface for FBEM staff while offering powerful forecasting capabilities.

Results

1. Gross Weight Distributed Annually

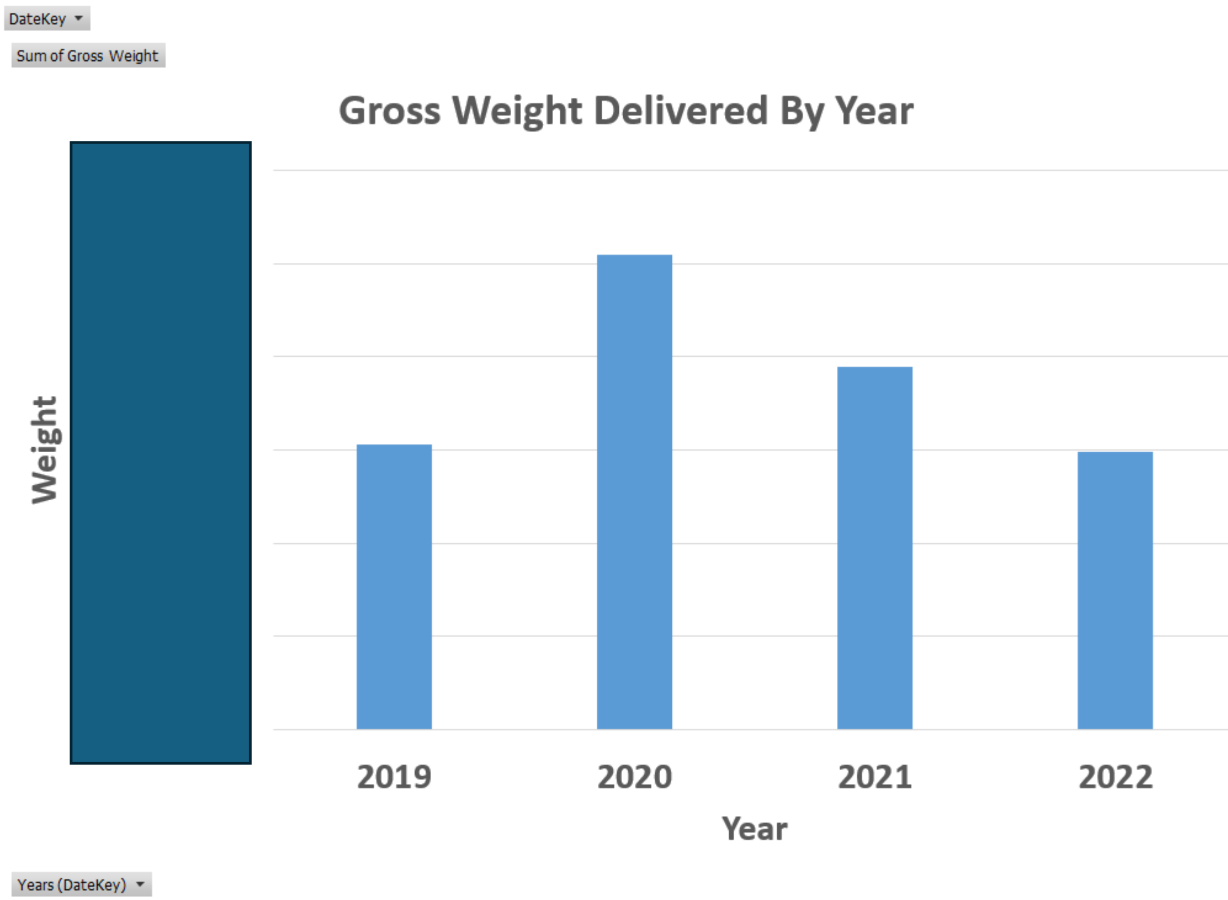


Figure 1: Total Weight of Food Donations (2019–2022)

The bar graph illustrates the total weight of food donations received by the Food Bank of Eastern Michigan from 2019 to 2022.

- **2019**: The donations reflected a typical year of contributions.
- **2020**: A dramatic rise is observed, with donations peaking at. This increase is primarily attributed to the COVID-19 pandemic. Lockdowns led to a surge in donations from grocery stores and other mass donors needing to offload unsold inventory. Also, US government programs to support people who are out of work were funded by donations.
- **2021**: As the pandemic's effects began to stabilize, donations decreased but remained above pre-pandemic levels, signifying the continued recovery phase.
- **2022**: Donations normalized further, returning to 2019 levels.

This trend highlights how external events, such as the pandemic, can significantly impact donation volumes. It also underscores the importance of adaptable supply chain planning to manage such fluctuations effectively.

2. Food Categorization by UNC Product type

UNC Product type which are FTE	
UNC Product Type	**FTE**
21 - Pasta: Macaroni, Spaghetti, Noodles	FTE Products
06 - Complete Meal/Entree, Soup	Non FTE products
Unknown	Non FTE products
28 - Fresh Fruits/Vegetables	FTE Products
10 - Fruit: Canned and Frozen	FTE Products
04 - Bread/Bakery: Bread, Biscuits, Rolls, Batter, Tortillas, Pie Crusts	Non FTE products
16 - Mixed and Assorted Food	FTE Products
15 - Meat/Fish/Poultry	FTE Products
07 - Dairy: Yogurt, Cheese, Milk, Butter, Sour Cream, Ice Cream	FTE Products
01 - Assorted Non-Food: Household Goods, Toys, Books, Clothing	Non FTE products
25 - Snack Food/Cookies: Candy, Crackers, Marshmallows	Non FTE products
03 - Beverage: Coffee, Tea, Soda, Drinks	FTE Products
27 - Vegetables: Canned and Frozen	FTE Products
26 - Spice/Condiment/Sauce: Herbs, Salt, Sugar, Mixes, Bread Crumbs, Vinegar, Extracts, N	Non FTE products
23 - Protein - Non-Meat: Peanut Butter, Beans, Eggs, Pork & Beans, Nuts	FTE Products
05 - Cereal: Hot and Cold	Non FTE products
24 - Rice	FTE Products

Figure 2: Food Categorization by UNC Product Type

The table snippet provides a categorization of unique **UNC Product Types**, indicating whether each item qualifies as **Food to Encourage (FTE)** or not.

- **FTE Items**: These are products identified as essential and nutritious, aligning with the food bank's goal to promote healthy food distribution. Examples may include fresh produce, grains, and dairy products.
- **Non-FTE Items**: These include products that are not prioritized for distribution due to their lower nutritional value or other factors, such as snack foods or processed items.

This categorization is a vital step in aligning the food bank's operations with federal and organizational requirements. It also enables management to focus on ensuring a higher proportion of FTE items in their supply chain, promoting healthier food access for the communities served.

3. Gross Weight Delivered By FTE Category and year

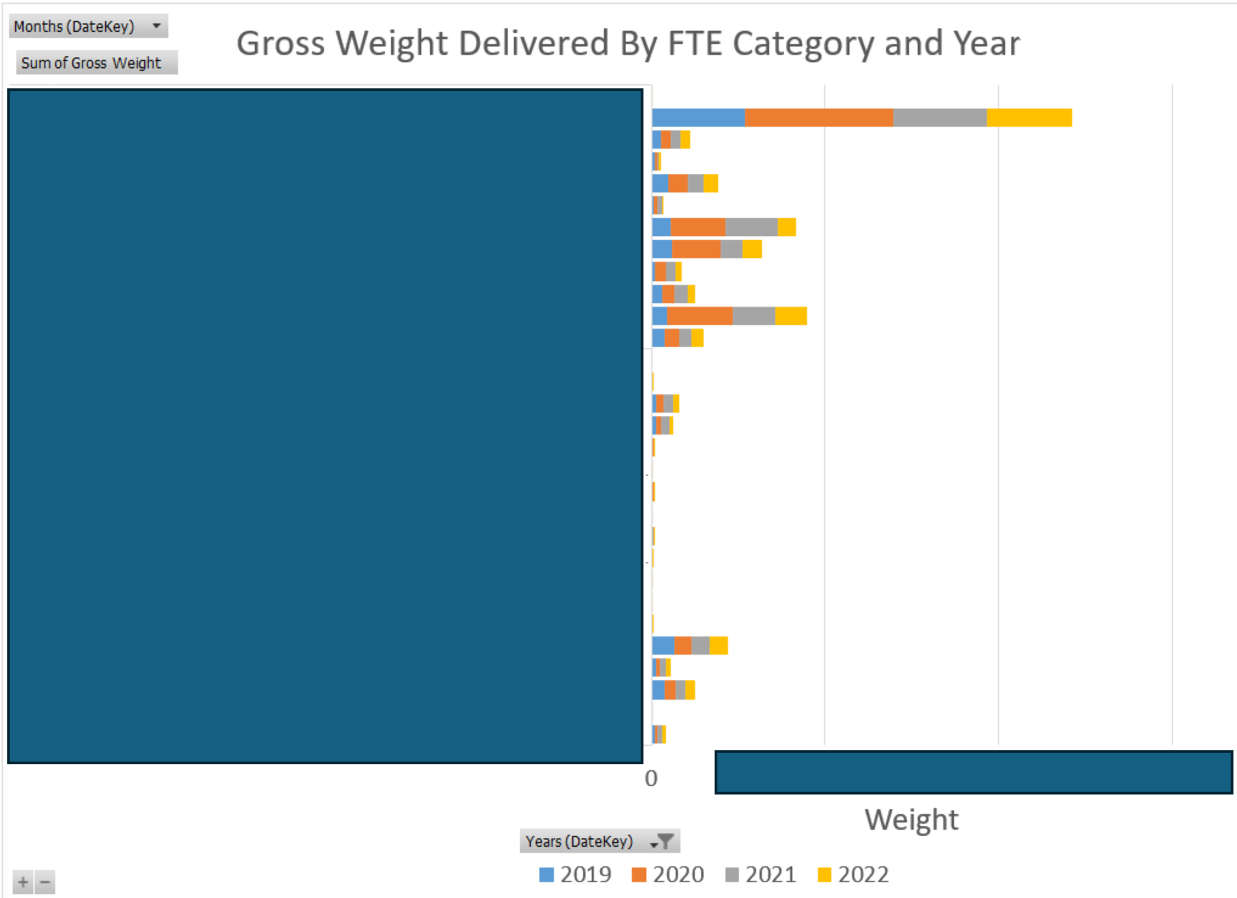


Figure 3: Stacked Bar Graph of FTE vs. Non-FTE Food Donations

This horizontal stacked bar graph presents the distribution of **Food to Encourage (FTE)** and **Non-FTE** items, with contributions filtered by year.

- **Graph Structure**:
  - The **upper stack** represents the total weight of FTE items donated.
  - The **lower stack** represents Non-FTE items.
  - Filters allow users to analyze donations by year and drill down into specific **UNC Product Types**, offering a customizable view of the data.
- **Insights**:
  - The majority of donations each year consist of FTE items, reflecting the food bank's commitment to promoting healthier food distribution.

- The flexibility of this pivot graph allows users to identify trends, such as whether a particular year saw an unusual spike or decline in FTE or Non-FTE categories.

This visualization highlights the importance of categorization in understanding the balance of essential versus non-essential food items, enabling the food bank to adjust procurement and distribution strategies accordingly.

4. The Forecasting Model Overview

Model

Level factor	Trend factor	Seasonality factor	Forecast	Error (Pounds)	Mean Squared Error
Alpha	Beta	Gamma	Intercept	Slope	Mean Squared error
0.902428986	0	0	[redacted]		3.11839E+12

Sequence	Delivered	Level Lt	Trend Tt	Seasonal Factor St	Forecast Ft	
	[redacted]					
1						342,260,516,929
2						17,539,037,291
3						31,676,415,902
4						1,551,889,984,865
5						297,006,857,961
6						479,802,618,989
7						33,829,977,331
8						341,582,325,479
9						28,757,572,466
10						14,116,266,488
11						266,684,947,690
12						78,257,900,189
13						2,046,272,898,241
14						7,768,353,360
15						58,766,897,524
16						532,806,112,222
17						51,018,239,045
18						111,195,385,112
19						22,878,007,474
20						49,057,206,190
21						28,841,441,901
22						304,224,059,083
23						27,031,125,715
24						180,718,117,075

Figure 4: Monthly Forecasting Model Snippet

Model

			Level factor	Trend factor	Seasonality factor	Forecast	Error (Pounds)	Mean Squared Error
			Alpha	Beta	Gamma	Intercept	Slope	Mean Squared error
			0.174752999	0	0.271754408	[redacted]		73,678,957,198
			553037	1287				
Weeks	Sequence	Delivered	Level L_t	Trend T_t	Seasonal Factor S_t	Forecast F_t		
2019-01	1	[redacted]					3,986,038,750	
2019-02	2						29,730,548,073	
2019-03	3						15,789,109,183	
2019-04	4						492,234,594	
2019-05	5						127,660	
2019-06	6						980,952,596	
2019-07	7						7,043,706,850	
2019-08	8						46,025,468	
2019-09	9						440,312,494	
2019-10	10						6,133,794,487	
2019-11	11						366,074,641	
2019-12	12						2,845,938,654	
2019-13	13						17,176,711,389	
2019-14	14						72,884,824,499	
2019-15	15						37,420,316,532	
2019-16	16						25,755,636,962	
2019-17	17						31,069,960,117	
2019-18	18						1,954,799,575	
2019-19	19						752,287,493	
2019-20	20						18,977,126,311	
2019-21	21						27,902,926,712	
2019-22	22						61,568,226,735	
2019-23	23						2,897,780,713	
2019-24	24						23,363,360,151	
2019-25	25						18,108,929,769	
2019-26	26						28,298,222,627	
2019-27	27						63,043,087	
2019-28	28						21,980,146,320	
2019-29	29						238,331,760	
2019-30	30						3,044,910,053	
2019-31	31						248,879,013	

Figure 5: Weekly Forecasting Model Snippet

The snippet highlights the Monthly and Weekly forecasting models developed to predict future donation volumes using **Winters' Method**. This model ensures accurate forecasting by accounting for trends and seasonality, offering a robust tool for supply chain planning.

- **Model Components**:
  The forecasting model includes key metrics such as:
  - **Delivered Weight**: The actual weight of donations delivered.
  - **Level (Lt)**: The baseline level of donation volume at the end of the current period.
  - **Trend (Tt)**: The rate of change in donation volumes over time.
  - **Seasonal Factor (St)**: Adjustments for recurring patterns in donation behavior (e.g., holiday seasons).
  - **Forecast (Ft)**: The projected weight of donations for the next period.

The core equation used in the model is:

$$F_{t+1} = (L_t + T_t) \cdot S_{t+1}$$

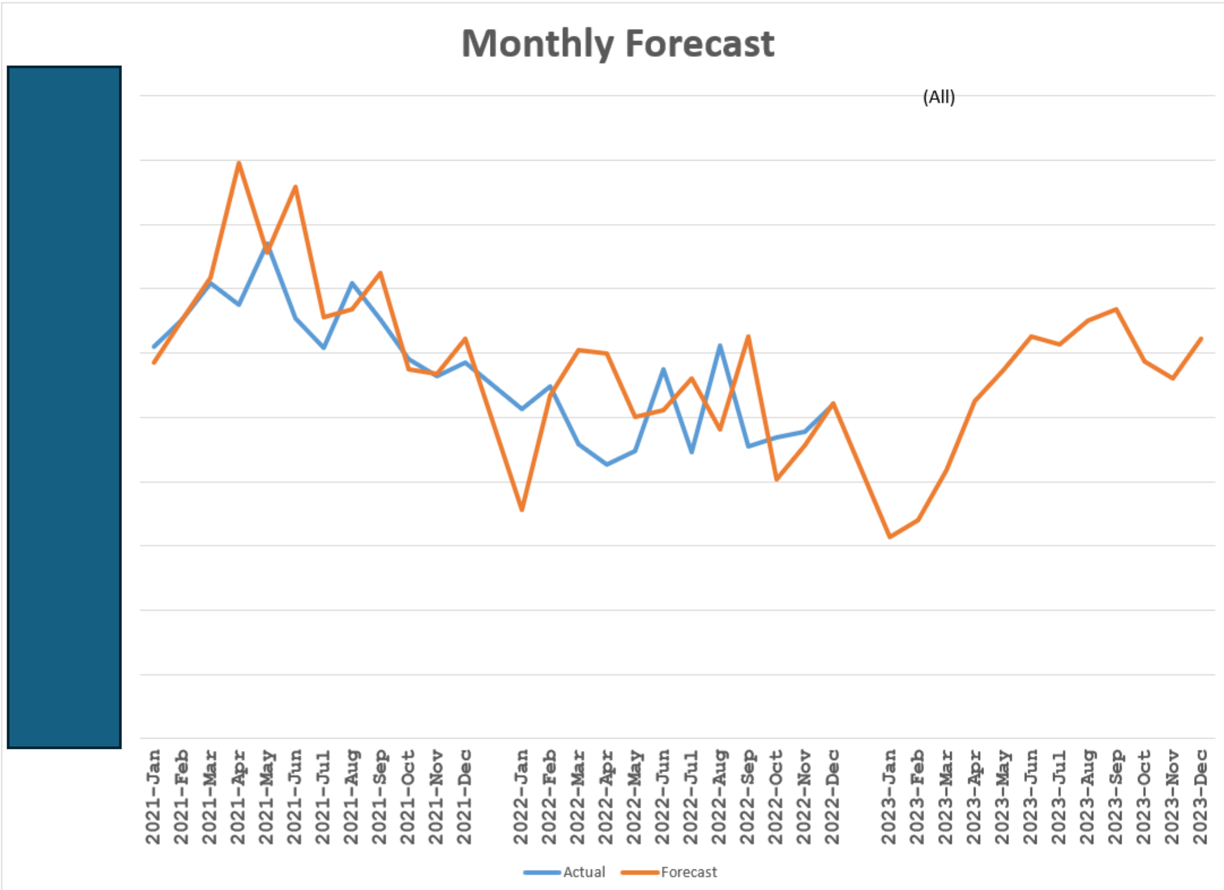
Where:

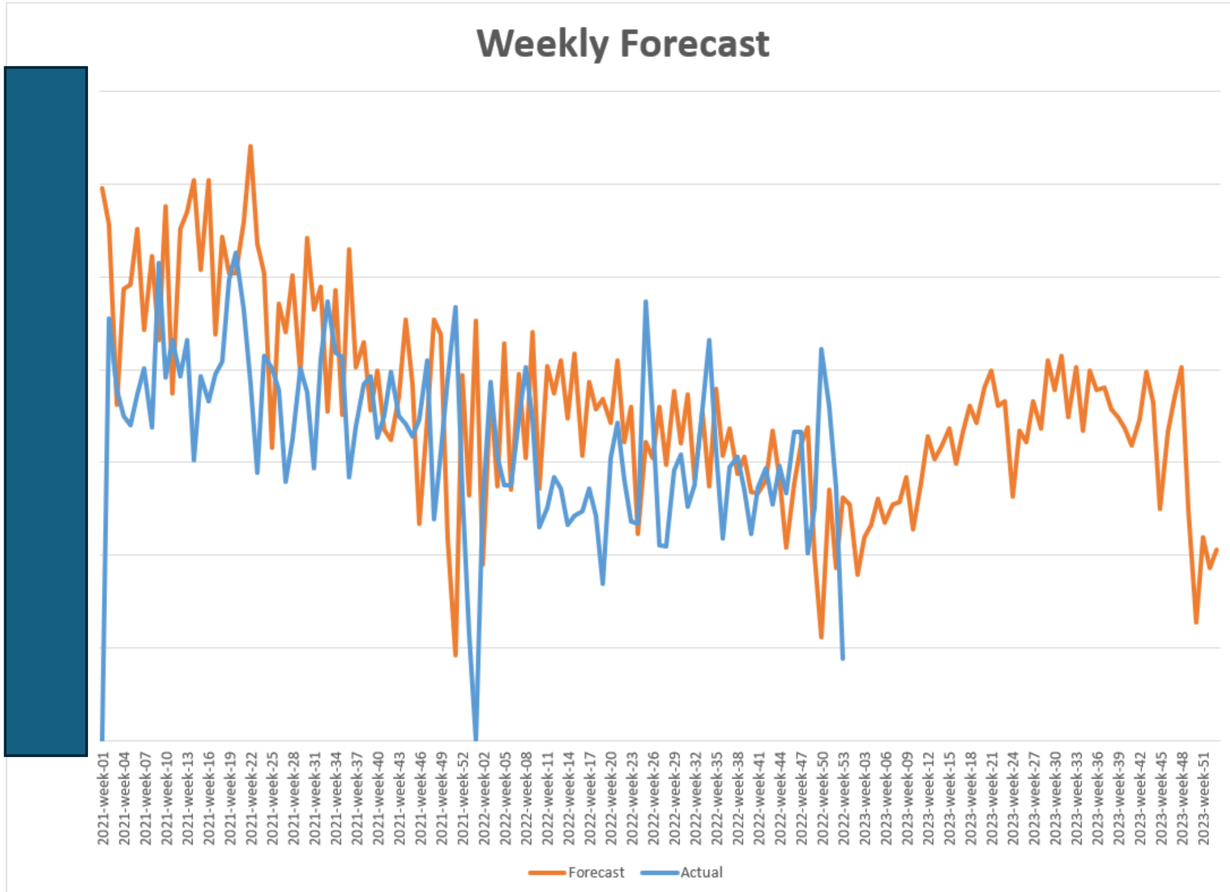
- $L_t = \alpha(D_t/S_{t-p}) + (1-\alpha)(L_{t-1} + T_{t-1})$
- $T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$
- $S_t = \gamma(D_t/L_t) + (1-\gamma)S_{t-p}$

- **Process**:
    1. **Selection**: The user begins by selecting a specific category (or categories) from the pivot table. Filters such as **UNC Product Type** or **FTE categorization** can also be applied for targeted analysis.
    2. **Running Solver**: The solver is used to minimize the **Mean Squared Error (MSE)** by adjusting the parameters α\alphaα, β\betaβ, and γ\gammaγ in Winters' Method. The **GRG Nonlinear** method ensures optimal parameter tuning for accurate forecasts.
    3. **Updating the Winters' Table**: Once the solver runs, the **Winters' Table** updates dynamically, recalculating the level, trend, seasonal factors, and forecast values.
    4. **Visualizing Results**: The forecast is plotted alongside actual delivered weights in a line graph, providing a clear comparison of predicted and actual values. This visualization helps the food bank identify patterns, anomalies, or inefficiencies in their supply chain.

These forecasting models help FBEM improve operational and strategic planning, enabling better decision-making for food distribution. The integration with Excel ensures accessibility for the food bank's management team, while the code provides scalability for future applications.

5. Monthly And Weekly Forecast Graphs





Figures 5 and 6: Monthly and Weekly Forecasting Models

The **Monthly Forecasting Chart** and **Weekly Forecasting Chart** provide detailed insights into donation trends over time, comparing actual delivered weights with forecasted values.

- **Monthly Model** (*Figure 5*):
    - **X-Axis**: Represents **Year-Month** combinations (e.g., 2019-01, 2019-02).
    - **Y-Axis**: Shows the total weight of donations delivered, measured in pounds.
    - The line chart clearly displays the variations in donation volumes across months, highlighting seasonal trends or external factors influencing donations (e.g., holiday seasons or pandemic-related spikes).
- **Weekly Model** (*Figure 6*):
    - **X-Axis**: Represents individual **Weeks** (e.g., Week 1, Week 2).
    - **Y-Axis**: Depicts the weight of donations delivered in pounds.
    - This model provides granular insights into weekly fluctuations, offering a more detailed understanding of short-term trends and operational challenges.

In both models, the **Actual Delivered** and **Forecasted Weights** are plotted as separate lines, enabling a direct comparison of real versus predicted values. These visualizations allow the Food Bank of Eastern Michigan (FBEM) to:

- Identify periods where actual deliveries significantly deviate from forecasts.
- Understand the impact of external factors on donation trends.
- Enhance planning accuracy for food distribution.

The alignment of the forecasted line with the actual delivered line demonstrates the effectiveness of the forecasting models. Both charts serve as valuable tools for FBEM's operational analysis and strategic planning.

These models and graphs provide the Food Bank of Eastern Michigan (FBEM) with powerful tools for in-depth analysis and strategic planning. By utilizing forecasting models and interactive visualizations, FBEM can evaluate donation patterns, understand food distribution across counties, and identify areas for improvement in their supply chain. The integration with Excel ensures that these tools are accessible to the management team for day-to-day operations.

Note: The data labels on the graph are intentionally covered to protect sensitive information, as we cannot share the actual data of the Food Bank of Eastern Michigan (FBEM) for security reasons.

Conclusion

In conclusion, the analysis and forecasting models developed for the Food Bank of Eastern Michigan (FBEM) provide invaluable insights into donation patterns, food distribution, and operational efficiency. By leveraging historical data and predictive tools, FBEM can make data-driven decisions to optimize its supply chain, ensure equitable food distribution across counties, and minimize waste. The integration with Excel ensures accessibility for the management team, while the forecasting models offer scalability for future strategic planning. These tools empower FBEM to better serve its partner agencies and the communities in need, aligning with its mission to alleviate hunger and promote healthier living.

Appendix

Code for Monthly Data Model

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import minimize


# Load the Excel file

file_path = 'FBEM Forecasting Workbook.xlsx'  # Update this with the correct file path

sheet_name = ' Monthly Data Model'


# Load the data and extract relevant columns

data = pd.read_excel(file_path, sheet_name=sheet_name)


# Extract Delivered (Actual) data from the table in "J10:Q74"

forecast_actual_data = data.loc[9:74, ['Unnamed: 11']]

forecast_actual_data.columns = ['Delivered']

forecast_actual_data = forecast_actual_data.dropna()


# Convert delivered data to numeric

delivered = forecast_actual_data['Delivered'].astype(float).values


# Define the Triple Exponential Smoothing function

def triple_exponential_smoothing(params, data, seasons=12):

    alpha, beta, gamma = params

    n = len(data)

    level = np.zeros(n)

    trend = np.zeros(n)

    seasonal = np.zeros(seasons)

    forecast = np.zeros(n)


    # Initialization

    level[0] = data[0]

    trend[0] = data[1] - data[0]

    seasonal[:seasons] = [data[i] / level[0] for i in range(seasons)]

    # Forecast calculation

    for t in range(n):

        if t >= seasons:

            seasonal_term = seasonal[t % seasons]

        else:

            seasonal_term = 1


        if t > 0:

            level[t] = alpha * (data[t] / seasonal_term) + (1 - alpha) * (level[t-1] + trend[t-1])

            trend[t] = beta * (level[t] - level[t-1]) + (1 - beta) * trend[t-1]

            seasonal[t % seasons] = gamma * (data[t] / level[t]) + (1 - gamma) * seasonal[t % seasons]


        forecast[t] = (level[t-1] + trend[t-1]) * seasonal_term if t > 0 else data[0]


    # Compute MSE

    mse = np.mean((data - forecast[:n])**2)

    return mse, forecast
```

```
# Optimization function to minimize MSE

def optimize_smoothing(data):

    # Initial guesses for alpha, beta, gamma

    initial_params = [0.5, 0.5, 0.5]

    bounds = [(0, 1), (0, 1), (0, 1)]


    result = minimize(lambda params: triple_exponential_smoothing(params, data)[0],

                      initial_params, bounds=bounds, method='L-BFGS-B')

    return result.x  # Optimized alpha, beta, gamma


# Run solver to calculate optimal alpha, beta, gamma

optimal_params = optimize_smoothing(delivered)

alpha, beta, gamma = optimal_params


# Generate forecasts using the optimal parameters

mse, forecast = triple_exponential_smoothing(optimal_params, delivered)


# Create MSE Table

mse_table = pd.DataFrame({

    'Level': [alpha],

    'Alpha': [alpha],

    'Beta': [beta],

    'Gamma': [gamma],

    'MSE': [mse]

})

# Plot Delivered vs Forecasted

plt.figure(figsize=(12, 6))

plt.plot(delivered, label='Delivered (Actual)', color='blue', marker='o')

plt.plot(forecast, label='Forecast', color='red', linestyle='--')

plt.title('Delivered vs Forecasted Values')

plt.xlabel('Time')

plt.ylabel('Pounds')

plt.legend()

plt.grid()

plt.show()


# Display the MSE Table

print("MSE Table:")

print(mse_table)
```

Code for Weekly Data Model

```
import pandas as pd

import numpy as np

import matplotlib.pyplot
as plt

from scipy.optimize
import minimize


# Load the Excel file

file_path = 'FBEM
Forecasting
Workbook.xlsx'  # Update
this with the correct file
path

sheet_name = ' Weekly
Data Model'


# Load the data and
extract relevant columns

data =
pd.read_excel(file_path,
sheet_name=sheet_nam
e)


# Extract Delivered
(Actual) data from the
table

forecast_actual_data =
data.loc[9:1000000,
['Unnamed: 11']]

forecast_actual_data.col
umns = ['Delivered']

forecast_actual_data =
forecast_actual_data.dro
pna()


# Convert delivered data
to numeric

delivered =
forecast_actual_data['De
livered'].astype(float).val
ues


# Define the Triple
Exponential Smoothing
function

def
triple_exponential_smoot
hing(params, data,
seasons=12):

    alpha, beta, gamma =
params

    n = len(data)

    level = np.zeros(n)

    trend = np.zeros(n)

    seasonal =
np.zeros(seasons)

    forecast = np.zeros(n)


    # Initialization

    level[0] = data[0]

    trend[0] = data[1] -
data[0]

    seasonal[:seasons] =
[data[i] / level[0] for i in
range(seasons)]


    # Forecast calculation

    for t in range(n):

        if t >= seasons:

            seasonal_term =
seasonal[t % seasons]

        else:

            seasonal_term = 1


        if t > 0:

            level[t] = alpha *
(data[t] / seasonal_term)
+ (1 - alpha) * (level[t-1] +
trend[t-1])

            trend[t] = beta *
(level[t] - level[t-1]) + (1 -
beta) * trend[t-1]

            seasonal[t %
seasons] = gamma *
```

```
(data[t] / level[t]) + (1 - gamma) * seasonal[t % seasons]

        forecast[t] = (level[t-1] + trend[t-1]) * seasonal_term if t > 0 else data[0]

    # Compute MSE
    mse = np.mean((data - forecast[:n])**2)
    return mse, forecast

# Optimization function to minimize MSE
def optimize_smoothing(data):
    # Initial guesses for alpha, beta, gamma
    initial_params = [0.5, 0.5, 0.5]
    bounds = [(0, 1), (0, 1), (0, 1)]

    result = minimize(lambda params: triple_exponential_smoothing(params, data)[0],
                      initial_params, bounds=bounds, method='L-BFGS-B')
    return result.x  # Optimized alpha, beta, gamma

# Run solver to calculate optimal alpha, beta, gamma
optimal_params = optimize_smoothing(delivered)
alpha, beta, gamma = optimal_params

# Generate forecasts using the optimal parameters
mse, forecast = triple_exponential_smoothing(optimal_params, delivered)

# Create MSE Table
mse_table = pd.DataFrame({
    'Level': [alpha],
    'Alpha': [alpha],
    'Beta': [beta],
    'Gamma': [gamma],
    'MSE': [mse]
})

# Plot Delivered vs Forecasted
plt.figure(figsize=(12, 6))
plt.plot(delivered, label='Delivered (Actual)', color='blue', marker='o')
plt.plot(forecast, label='Forecast', color='red', linestyle='--')
plt.title('Delivered vs Forecasted Values')
plt.xlabel('Time')
plt.ylabel('Pounds')
plt.legend()
plt.grid()
plt.show()

# Display the MSE Table
print("MSE Table:")
print(mse_table)
```