Quantifying Lost Sales

A Data-Driven Approach to Mitigating Out-of-Stock Costs in the Retail Sector

Shriyan Singh **Dirk Hoffmann**

Department of Computer Science,
Stellenbosch University, South Africa

Demo 1 Presentation

May 28, 2024

Table of Contents

Project Background

- Retail companies heavily depend on effective inventory management
- Out-of-stock situations are a major challenge, leading to lost sales and decreased customer loyalty
- Need for a data-driven framework to quantify lost sales and optimize inventory decisions

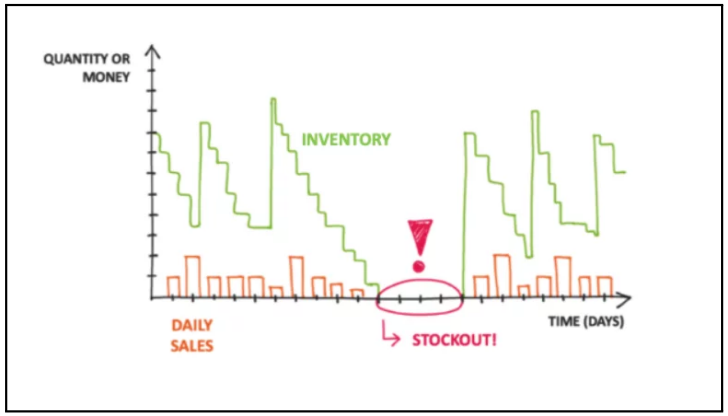

Figure: Graph illustrating impact of out-of-stock

Project Objectives

1. Quantify lost sales across different product within diffrent categories
   - Develop a robust methodology to measure the opportunity cost associated with out-of-stock situations
   - Estimate potential sales if out-of-stock products were available
2. Create an optimization framework to balance availability and waste costs
   - Identify products, stores, and periods where maximizing availability outweighs potential waste costs, and vice versa
3. Build a front-end dashboard system to display results and output

Literature Survey

- Extensive research on the impact of out-of-stock (OOS) situations on consumer behavior and sales in the retail industry
- OOS situations can lead to a 4% loss in retail sales on average (Corsten & Gruen, 2003)
- When faced with an OOS situation:
  - 31% of consumers will buy a different item
  - 26% will delay their purchase
  - 9% will buy from a competitor

Quantifying Lost Sales

- Various methods have been used to estimate lost sales due to OOS situations
- One common approach is to compare sales during an OOS period with sales during a comparable in-stock period (Gruen & Corsten, 2007)
- Another approach is to survey customers to gather data on intended purchases and their behavior when faced with an OOS situation (Woensel et al., 2007)

Managing Inventory Levels

- Researchers have explored techniques to manage inventory levels and minimize the likelihood of OOS situations
- The newsvendor model is a popular method for determining the optimal order quantity to balance the costs of overstocking and understocking (Silver, Pyke, & Peterson, 1998)
- Other optimization techniques, such as dynamic programming and simulation-based methods, have been used to address complex inventory problems (Zipkin, 2000; Axsäter, 2006)

Figure: The newsvendor model, also known as the newsboy problem, is analogous to a newspaper vendor deciding how many copies to stock daily, considering uncertain demand and the risk of unsold copies becoming worthless.

Limitations and Research Gaps

- Many studies focus on analysis at the aggregate level, potentially missing important details about the complex dynamics of OOS situations at the store or product level
- There is a need for more advanced data-driven methods and machine learning techniques to predict customer behavior and optimize inventory decisions in real-time
- The proposed research aims to fill these gaps by utilizing extensive data sources and developing a comprehensive, data-driven framework

Industry Context and Challenges

- Retailers face unique challenges in managing inventory and dealing with OOS situations
- Intense competition, narrow profit margins, and the need to efficiently manage inventory costs
- Highly volatile consumer preferences and demand patterns influenced by seasonality, promotions, and changing trends
- Complexity of managing a diverse range of products across multiple retail locations (Agrawal & Smith, 2013)
- The proposed research aims to provide a flexible, data-driven framework tailored to the specifics of the retail environment

Data Description

This project utilizes three main datasets:

- Sales data
- Article data
- Inventory movements data

These datasets form the foundation for quantifying lost sales, building predictive models, and optimizing inventory decisions.

Sales Data

The sales data contains information about individual sales transactions at stores. It includes the following attributes:

- `date_key` (int64): The date of the sales transaction (format: YYYYMMDD)
- `site_code` (int64): The unique identifier of the store where the sale occurred
- `unique_ticket_id` (object): A unique identifier for each sales transaction
- `ticket_end_time` (object): The timestamp of when the sales transaction was completed
- `article_code` (int64): The unique identifier of the article (product) sold
- `sales_qty_alternate_uom` (float64): The quantity of the article sold in an alternate unit of measure
- `sales_qty_base_uom` (float64): The quantity of the article sold in the base unit of measure
- has 15 Million data points!

Inventory Movements Data

The inventory movements data captures the changes in inventory levels for each article at different stores over time. It includes the following attributes:

- `unique_key` (object): A unique identifier for each inventory movement record
- `posting_date` (int64): The date of the inventory movement (format: YYYY-MM-DD)
- `entry_timestamp` (object): The timestamp of when the inventory movement was recorded
- `article_code` (int64): The unique identifier of the article involved in the inventory movement
- `movement_type` (object): The type of inventory movement (e.g., receipt, issue, transfer)
- `quantity_base_uom` (float64): The quantity of the inventory movement in the base unit of measure
- `stock_level_before_posting` (float64): The stock level of the article before the inventory movement
- has 1 Million data points

Inventory Movements vs Sales Data

Main differences between the data sets:

- Sales has 15 Million data points while movements has only 1 Million
- Sales includes exclusively sales data with no stock level data; while movements keeps the stock level and different kinds of stock level changes: Sales, Shrinkage, Donations, etc.
- Sales ranges from January 2022 - April 2024
- Movements ranges from July 2023 - April 2024
- While both are valuable data sets, the sheer volume of data in Sales dataset makes it a strong contender to train complex models on

Article Data

The article data provides detailed information about each article (product) sold at stores. It includes the following attributes:

- `article_code` (int64): The unique identifier of the article
- `article_desc` (object): A description of the article
- `base_uom` (object): The base unit of measure for the article
- `super_dept_no` (int64): The number of the super department to which the article belongs
- `dept_no` (int64): The number of the department to which the article belongs
- `category_no` (int64): The number of the category to which the article belongs
- `merchandise_category_no` (int64): The number of the merchandise category to which the article belongs
- `sub_category_no` (float64): The number of the sub-category to which the article belongs

Data Preprocessing

During data preprocessing, the datasets will be:

- Checked for missing values and handled by removing records with missing critical information and imputing or assigning defaults for non-critical fields
- Duplicate records will be removed to ensure data integrity
- Outliers and anomalies will be analyzed and treated to prevent skewed analysis
- The data will be validated against business rules to ensure consistency and accuracy

Data Exploration and Preprocessing

- Exploratory Data Analysis (EDA) techniques to identify patterns, trends, and relationships
- Peculiar Findings
  - Missing categories from sales data:
    - Bakery Sales
    - Beef
    - Poultry
    - 19 other categories!
- Data visualizations to showcase sales trends, product distributions, inventory movements, and correlations between variables

NARROWING DOWN THE PROBLEM

- WHY WE NEED TO?
- ON WHAT BASIS
- EXAMPLES TO FOLLOW

GOOD: Condiments, Oil and Spices

article_desc	category_desc	merchandise_category_desc	zero_sales_hours	average_sales_per_hour
OIL SUNFLOWER RITEBRAND 750ML BOT	Condiments, Oils and Spices	Edible Oil	69	2.10378
OIL SUNFLOWER SUNFOIL 2L	Condiments, Oils and Spices	Edible Oil	26	5.88879
OIL SUNFLOWER RITEBRAND 4L BOT	Condiments, Oils and Spices	Edible Oil	20	2.27249
OIL SUNFLOWER RITEBRAND 2L BOT	Condiments, Oils and Spices	Edible Oil	19	3.5113
SOYA MINCE TOP CLASS 500/400G, BF ONION	Condiments, Oils and Spices	Meal Solutions	18	1.30945
OIL COOKING BLENDED CROWN 500ML BOT	Condiments, Oils and Spices	Edible Oil	18	2.06192
SPCE REF PORTUGUES CHIC ROBERTSONS 75G	Condiments, Oils and Spices	Spices, Seasonings	15	1.33293
SPICE SPICE MECCA 25G, CLOVES WHOLE	Condiments, Oils and Spices	Spices, Seasonings	13	1.15851
SOUP KNORROX 400G BAG, MUTTON	Condiments, Oils and Spices	Soups and Stock	13	1.20983
SOYA MINCE TOP CLASS 500/400G, MUTTON	Condiments, Oils and Spices	Meal Solutions	12	1.33634
OIL OLIVE EXT VIRGIN SANTA BIANCA 250ML	Condiments, Oils and Spices	Edible Oil	11	1.01529
SAUCE SWEET CHILLI WELLINGTONS 700ML	Condiments, Oils and Spices	Sauces, Marinades an	10	1.63155
SAUCE TOMATO ALL GOLD 700ML BOTTLE	Condiments, Oils and Spices	Sauces, Marinades an	10	2.99958
SPCE ROBERTSONS 7G, BBQ	Condiments, Oils and Spices	Spices, Seasonings	10	2.95337
SPCE PEPPER WHITE ROBERTSONS 7G	Condiments, Oils and Spices	Spices, Seasonings	9	3.16342
SOYA MINCE TOP CLASS 500/400G, CHAKALAKA	Condiments, Oils and Spices	Meal Solutions	9	1.17836
OIL COOKING BLENDED CROWN 2L	Condiments, Oils and Spices	Edible Oil	9	8.09699
OIL COOKING BLENDED CROWN 750ML	Condiments, Oils and Spices	Edible Oil	9	5.2644
MARINADE SMOKEY BBQ CHAMPIONSHIP 750ML	Condiments, Oils and Spices	Sauces, Marinades an	9	1.22817
OIL SUNFLOWER SUNFOIL 750ML	Condiments, Oils and Spices	Edible Oil	9	9.81372

Figure: Average sales per hour alongside zero sales hours for oils.

GOOD: Carbonated, Cordials, Juices

article_desc	category_desc	merchandise_category_desc	zero_sales_hours	average_sales_per_hour
WATER STILL AQUELLE 5L	Carbonated, Cordials, Juices	Water	54	1.76192
SOFT DRNK PEPSI 2L, REG	Carbonated, Cordials, Juices	CSD Bottles	39	1.37447
WATER STILL EAST HIGHLANDS 5L	Carbonated, Cordials, Juices	Water	29	1.71879
SOFT DRNK COCA COLA 1.5L NRB, NO SUGAR	Carbonated, Cordials, Juices	CSD Bottles	27	7.0538
SOFT DRNK ORIGINAL COCA COLA 1.5L RB	Carbonated, Cordials, Juices	CSD Bottles	26	1.40704
WATER STILL THIRSTI 5L	Carbonated, Cordials, Juices	Water	25	1.267
SOFT DRNK ORIGINAL COCA COLA 300ML NRB	Carbonated, Cordials, Juices	CSD Bottles	23	7.90196
SOFT DRINK NO CAFEINE COCA COLA 1.5L NRB	Carbonated, Cordials, Juices	CSD Bottles	22	3.21444
SOFT DRNK COCA COLA 1.5L NRB, ORG TASTE	Carbonated, Cordials, Juices	CSD Bottles	22	10.567
SQUASH CONC OROS 5L, ORIG	Carbonated, Cordials, Juices	Cordials and Juices	19	1.27616
ENERGY DRINK POWER PLAY 440ML, ORIG	Carbonated, Cordials, Juices	CSD Cans	15	2.16823
WATER STILL REFILL PURE LIFESTYLE 5L	Carbonated, Cordials, Juices	Water	15	8.84872
WATER STILL REFILL PURE LIFESTYLE 2L	Carbonated, Cordials, Juices	Water	15	1.44754
SOFT DRINK JIVE X 2L BOTTLE, CAPE APPL	Carbonated, Cordials, Juices	CSD Bottles	14	1.62756
SOFT DRNK JIVE 2L, GDILLA	Carbonated, Cordials, Juices	CSD Bottles	14	1.91025
WATER STILL REFILL PURE LIFESTYLE 1L	Carbonated, Cordials, Juices	Water	12	1.81342
SOFT DRNK ZIP 2L, COLA ORIG	Carbonated, Cordials, Juices	CSD Bottles	12	1.95371
ENERGY DRINK MOVE 500ML, ORIG	Carbonated, Cordials, Juices	CSD Cans	12	1.49169
ENERGY DRINK SCORE 500ML, PASSIONFRT	Carbonated, Cordials, Juices	CSD Cans	12	1.92084
SOFT DRINK JIVE X 2L BOTTLE, GING BEER	Carbonated, Cordials, Juices	CSD Bottles	12	1.8118

Figure: Average sales per hour alongside zero sales hours for carbanted drinks.

BAD: DIY Items

	article_desc	category_desc	merchandise_category_desc	zero_sales_hours	average_sales_per_hour
15	TAPE PACKAGING QUALITY 48MMX50M	DIY	Adhesives and filler	12	1.22733
27	IRONING BOARD MESH QUALITY	DIY	Laundry	11	0.844156
114	PADLOCK BRASS QUALITY 32MM	DIY	Security	10	1.01053
2	TAPE BUFF QUALITY 48MMX50M	DIY	Adhesives and filler	9	1.22335
12	PAINT SPRAY 300ML BRILIANT HEAT BLK	DIY	Paint	9	1.05495
46	PARAFFIN HANDYMANS 5L	DIY	Paint Products	7	1.33495
10	PADLOCK BRASS QUALITY 50MM	DIY	Security	6	0.78125
528	SCREWDRIVER SLOTTED TACTIX 3X75MM	DIY	Hand Tools	6	0.761905
135	SPIRITS METHYLATED HANDYMANS 750ML	DIY	Paint Products	5	1.03571
52	SEALANT ALCOLIN CLEAR 300ML	DIY	Adhesives and filler	5	1.2
167	PADLOCK IRON QUALITY 50MM	DIY	Security	5	1.08125
258	C-TRACK DBL LIZZY RUFFLETTE 2.0M DBL	DIY	Curtains and Blinds	5	1
98	GLUE CONTACT GENKEM 250ML	DIY	Adhesives and filler	5	1.04348
198	HEAD LAMP W/BASE QUALITY	DIY	Stoves and Accessori	5	1.36496
28	IRONING BOARD STANDARD QUALITY 1S	DIY	Laundry	4	1.02235
122	DRYER CLOTHES 18M QUALITY	DIY	Laundry	4	1.19474
44	GLUE CLEAR BOSTIK 25MLE	DIY	Adhesives and filler	4	1.06564
48	LOCK MORTICE PRO-SERIES 2 LEVER	DIY	Security	4	1.06306
402	POLISHER SET BONNET PRO TOOLS 2PC	DIY	Power Tools	4	1.24138
301	C-TRACK DBL LIZZY RUFFLETTE 1.5M	DIY	Curtains and Blinds	4	1.07407

Figure: Average sales per hour alongside zero sales hours for DIY items.

Interesting categories:

- Confectionery and Snacks
- Eggs
- Rice, Grains and Pasta
- Refrigerated Dairy
- Vegetables

SPREAD 25% FAT OLE 1KG (DAILY)

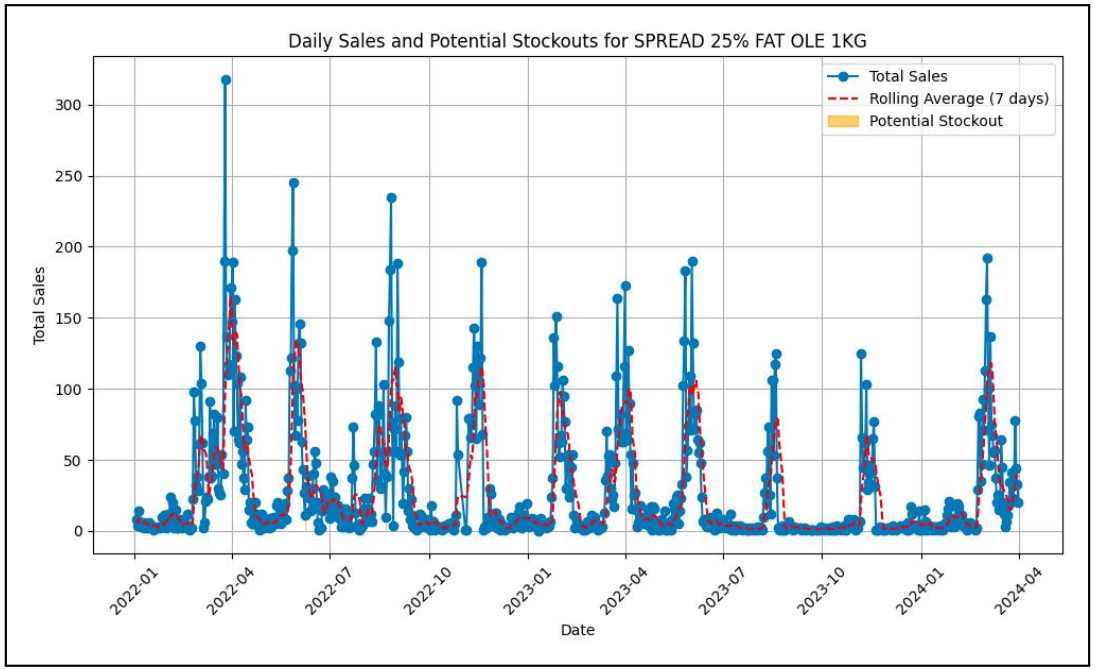

Figure: Daily sales and potential stock-outs for SPREAD 25% FAT OLE 1KG

SPREAD 25% FAT OLE 1KG (HOURLY)

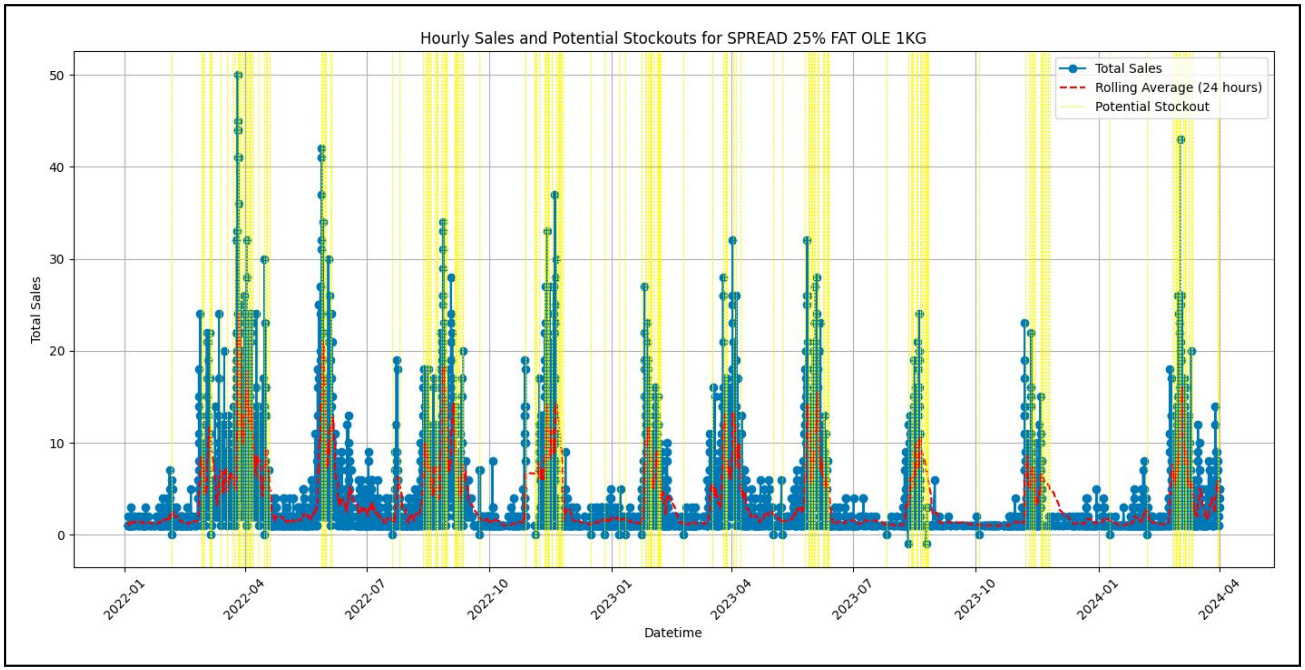

Figure: Hourly sales and potential stock-outs for SPREAD 25% FAT OLE 1KG

COCA-COLA 1.5L (DAILY)

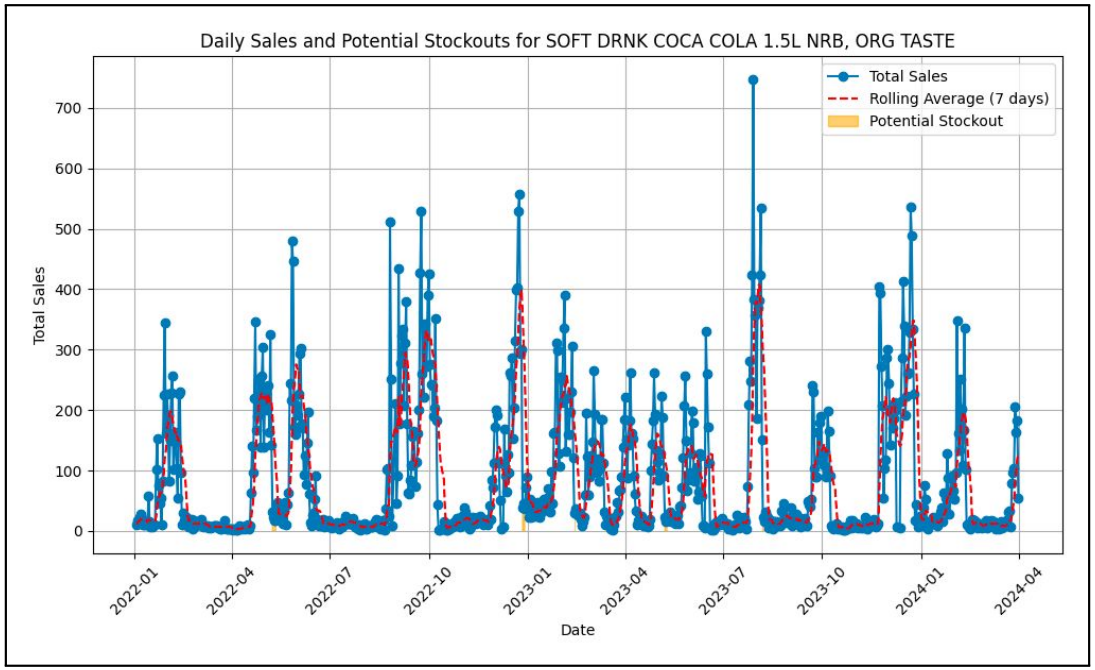

Figure: Daily sales and potential stock-outs for COCA-COLA 1.5L

COCA-COLA 1.5L (HOURLY)

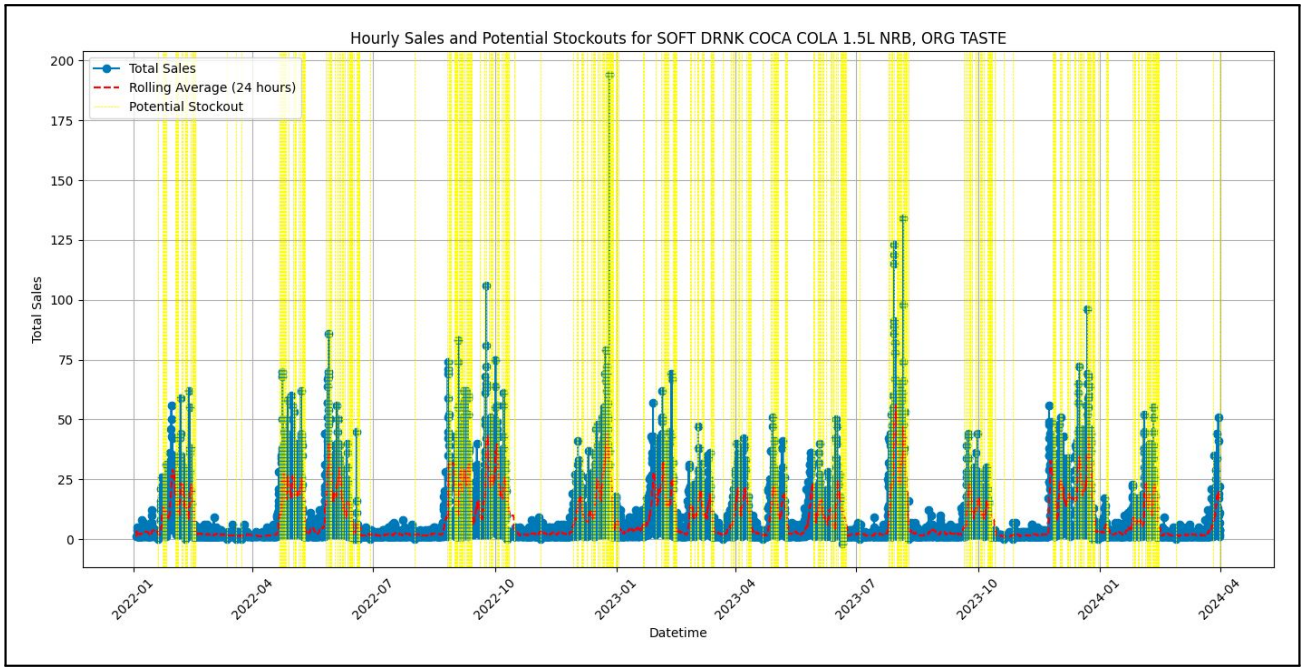

Figure: Hourly sales and potential stock-outs for for COCA-COLA 1.5L

Quantifying Lost Sales

- Multi-step approach to quantify lost sales due to out-of-stock situations
- Examine inventory movement data to identify instances of stock levels falling below zero or becoming negative
- Map relevant sales transactions to these instances
- Utilize time series forecasting models (e.g., ARIMA, Prophet) to estimate expected sales quantity for out-of-stock products
- Calculate lost sales as the difference between expected and actual sales quantities during the out-of-stock period

Introduction to ARIMA Models

ARIMA (AutoRegressive Integrated Moving Average) models:

- General class of models for forecasting time series
- Can handle series made stationary by differencing
- Combine autoregressive (AR) and moving average (MA) components

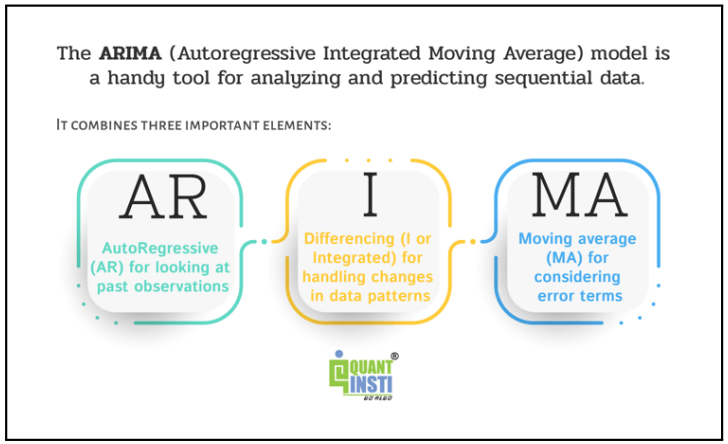

Figure: Illustration of how ARIMA gets it's name

ARIMA Model Structure

ARIMA(p,d,q) model:

- p = number of autoregressive terms
- d = number of nonseasonal differences for stationarity
- q = number of lagged forecast error terms

$$\hat{y}t = \mu + \phi_1 yt - 1 + \cdots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \cdots - \theta_q e_{t-q}$$

where y_t is the dth difference of the time series Y_t

Identifying ARIMA Model Components

1. Difference series until stationary (d)
2. Examine autocorrelation of stationary series
   - Positive autocorrelation suggests AR terms (p)
   - Negative autocorrelation suggests MA terms (q)
3. Estimate model, diagnose residuals, iterate

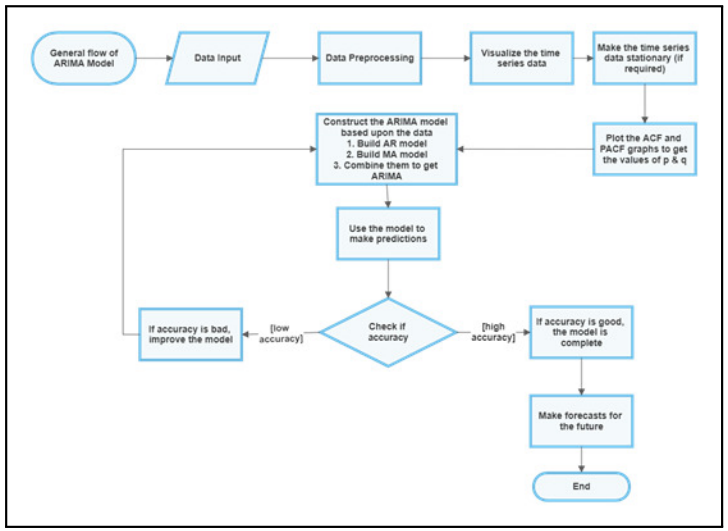

Figure: Flow chart showing the process of determining d, p, and q values.

Advantages of ARIMA Models

- Very general and flexible class of models
- Handle non-stationary series with trends
- Allow both autoregressive and moving average components
- Well-established model identification and estimation procedures

Predictive Model for Customer Purchase Behavior

- Machine learning techniques (e.g., gradient boosting, decision trees, logistic regression) to predict sales potential and customer purchase patterns
- Prepare dataset by combining relevant attributes from inventory movement, sales, and article data
- Split dataset into training and testing sets
- Train models on the training set and optimize performance using hyperparameter tuning
- Evaluate trained models on the testing set using appropriate evaluation metrics

Optimization Framework for Inventory Decisions

- Optimization framework to balance waste costs and availability while making optimal inventory decisions
- Consider variables such as lead times, ordering costs, holding costs, and product demand
- Formulate the optimization problem as a multi-objective optimization with the goal of minimizing waste costs and lost sales
- Apply optimization techniques such as simulated annealing, genetic algorithms, and linear programming to determine optimal inventory levels

Evaluation and Validation

- Evaluate the predictive model's performance using metrics like accuracy, precision, recall, and F1-score
- Employ cross-validation techniques, such as k-fold cross-validation, to ensure model robustness and generalizability
- Assess the optimization framework's performance using metrics like reduced lost sales, increased inventory turnover, and cost savings
- Conduct back-testing to validate the optimization results against historical sales and inventory data

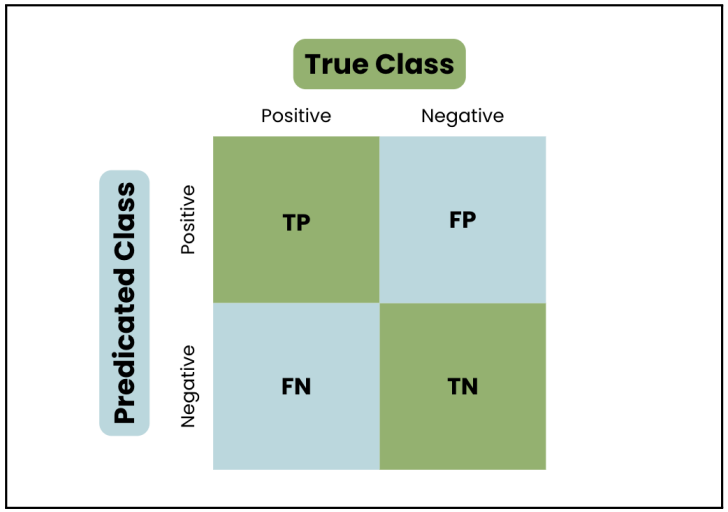

Figure: Illustration of a confusion matrix.

Data Science and Computer Science Integration

- Data science component focuses on quantifying lost sales, developing predictive models, and applying machine learning techniques to optimize inventory decisions.
- Computer science aspect involves building a fully functional web application that incorporates the data science models and provides a user-friendly interface for stakeholders.
- Integration of data science and computer science enables the creation of a reliable, scalable, and accessible solution.

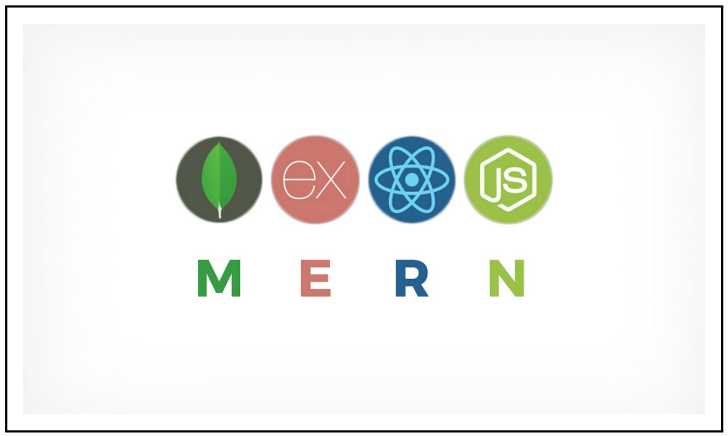

Figure: An illustration of M.E.R.N

Methodology flow diagram

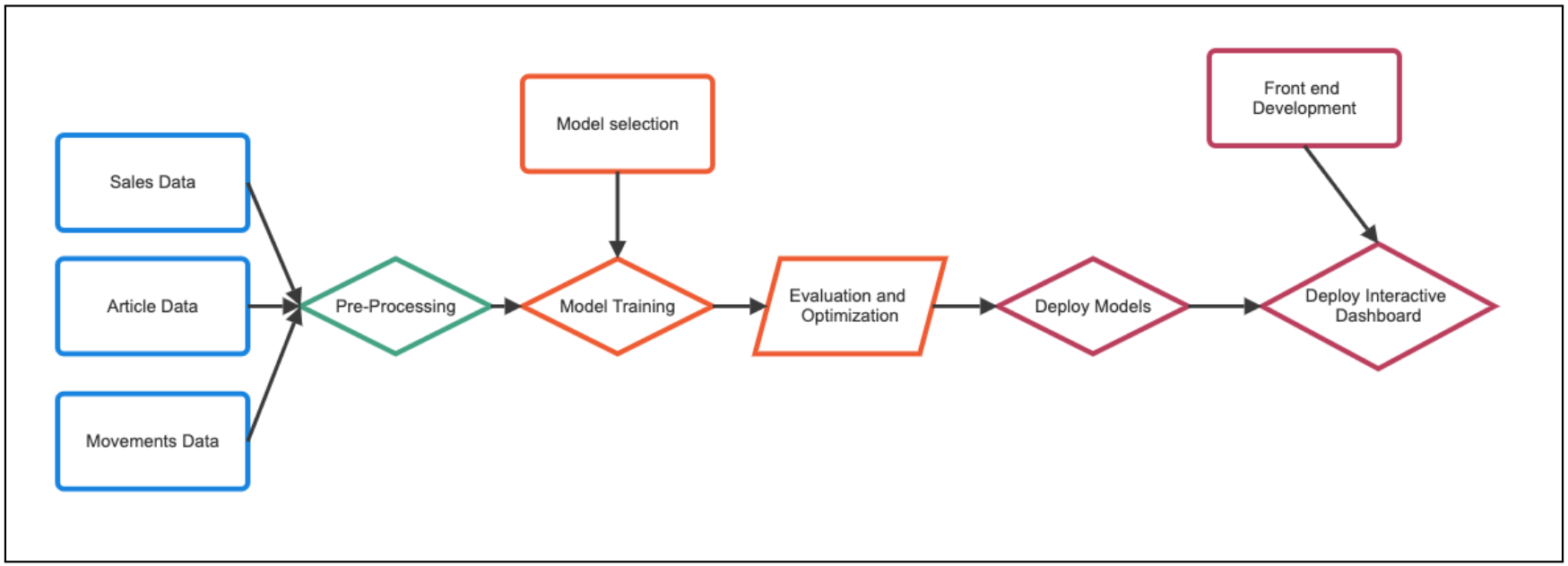

Figure: Flow diagram of project progression

Phase 1: Model Development

- Duration: 17 June to 1 July
- Undertaken by: Shriyan and Dirk
- Tasks:
  - Predict customer purchase behavior
  - Perform feature engineering and preprocessing techniques
  - Evaluate models using appropriate evaluation metrics
  - Carry out hyperparameter tuning

Phase 2: Optimization Framework

- Duration: 1 July to 15 July
- Handled by: Shriyan
- Tasks:
  - Define the objective function and constraints based on business requirements
  - Implement the optimization framework using suitable techniques
  - Test the optimization framework

Phase 3: Demo 2

- Duration: 22 July to 26 July
- Involved: Shriyan and Dirk

Phase 4: Sensitivity Analysis and Iterative Refinement

- Duration: 26 July to 6 September
- Managed by: Dirk
- Tasks:
  - Assess the robustness of the optimization framework
  - Evaluate the impact of varying input parameters
  - Analyze the results and draw insights
  - Refine the models and optimization framework based on evaluation results and sensitivity analysis
  - Make necessary adjustments to improve performance and align with business objectives
  - Re-evaluate the refined models and framework to ensure their effectiveness

Phase 5: Deploy the Web App

- Duration: 6 September to 25 October
- Carried out by: Shriyan and Dirk
- Tasks:
  - Develop the front-end user interface using HTML, CSS, and JavaScript
  - Implement the back-end functionality using Flask and Node.js
  - Integrate the data science models into the back-end for real-time predictions and optimization
  - Implement user authentication and authorization to ensure data security
  - Deploy the web application on a secure and scalable cloud platform like AWS or Azure