JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A study on the impact of demand forecasting accuracy on supply chain performance

Safuwan Ashraf
Student
Amity University

Abstract : This study investigates the critical link between demand forecasting accuracy and supply chain performance in the pharmaceutical industry, specifically within a single pharmacy setting. The research employs a comparative analysis to evaluate the effectiveness of various forecasting techniques for pharmaceutical sales data. By identifying the most accurate approach, the study aims to optimize inventory management and resource allocation within the pharmacy's supply chain. Furthermore, the research delves into proposing strategies and best practices specifically tailored to enhance demand forecasting accuracy in this context. These recommendations hold the potential to streamline the pharmacy's supply chain, minimize stockouts and overstocking situations, ultimately leading to improved efficiency and customer satisfaction.

I. INTRODUCTION

In today's dynamic and competitive business landscape, the success of any organization hinges on its ability to efficiently manage its supply chain. This intricate network of activities from procurement of raw materials to the delivery of finished goods faces constant challenges in optimizing resource allocation and meeting customer needs. At the heart of this optimization lies demand forecasting, the crucial practice of predicting future customer demand for products and services.

This research delves into the intricate relationship between demand forecasting accuracy and its subsequent impact on supply chain performance. By quantifying the influence of precise forecasts, exploring the challenges associated with achieving high accuracy, and investigating the potential of emerging technologies, this study aims to equip businesses with the knowledge and tools necessary to elevate their supply chain efficiency.

The Importance of Demand Forecasting

Demand forecasting serves as the cornerstone of supply chain planning. Accurate forecasts enable businesses to:

- Optimize inventory management: Precise predictions minimize the risk of stockouts, where in-demand products are unavailable due to insufficient stock. Conversely, accurate forecasts prevent overstocking, a costly scenario that ties up valuable capital in unsold inventory and incurs additional storage expenses.
- Reduce lead times: Lead time refers to the duration between placing an order and receiving the product. Accurate forecasts allow for informed procurement and production planning, leading to shorter lead times and faster product delivery, ultimately enhancing customer satisfaction.
- Improve resource allocation: Precise forecasts of future demand empower businesses to allocate resources effectively. This includes allocating production capacity, scheduling labour, and managing transportation needs, ensuring resources are available to meet anticipated demand fluctuations.
- Mitigate risks: The ability to anticipate future demand allows businesses to identify and mitigate potential disruptions. For instance, if a forecast predicts a surge in demand, the company can take proactive steps to secure additional raw materials or ramp up production capacity to avoid stockouts.

The Impact of Forecasting Accuracy on Supply Chain Performance

The accuracy of demand forecasts directly impacts various aspects of supply chain performance. This research will investigate the following key areas:

- Inventory Levels: Inaccurate forecasts can lead to either stockouts or overstocking. This study will quantify the impact of forecasting accuracy on inventory levels, aiming to establish a clear correlation between precise forecasts and optimal inventory management strategies.
- Lead Times: The research will explore how improved forecasting accuracy translates to shorter lead times. By analysing historical data and forecasting performance, we can determine the extent to which accurate predictions enable better planning and lead time reduction.
- Delivery Performance: On-time delivery is essential for customer satisfaction. This study will investigate the relationship between forecasting accuracy and delivery performance. When forecasts are precise, companies can allocate resources to meet demand fluctuations, leading to more timely deliveries.
- Cost Efficiency: Inaccurate forecasts can have a significant financial impact. The research will explore how improved forecasting accuracy contributes to cost reductions throughout the supply chain. By minimizing stockouts, optimizing inventory levels, and reducing lead times, businesses can achieve greater cost efficiency.

Challenges in Achieving High Forecasting Accuracy

While the benefits of accurate forecasting are undeniable, achieving high accuracy presents numerous challenges. This study will address the following obstacles:

- Market Dynamics: Consumer preferences can be unpredictable and susceptible to external factors like economic fluctuations, sudden trends, or competitor actions. These market dynamics can significantly impact demand, making it challenging to develop consistently accurate forecasts.
- Limited Data Availability: The quality and quantity of historical data can significantly influence the accuracy of forecasting models. Limited data availability, particularly for new products or volatile markets, can hinder the development of robust forecasting models.
- Data Quality: Even with sufficient data volume, inconsistencies, errors, or missing values within the data can significantly impact the accuracy of forecasts. This research will explore data cleaning and preprocessing techniques to ensure data quality and its subsequent impact on forecasting performance.

The Role of Emerging Technologies

The landscape of demand forecasting is constantly evolving with the emergence of new technologies. This research will explore the potential of these technologies in enhancing forecasting accuracy:

- Machine Learning (ML): ML algorithms have the capability to analyse vast datasets and identify complex patterns in customer behaviour. This study will investigate how ML models can learn from historical data and external factors to generate more accurate and dynamic forecasts.
- Big Data Analytics: Modern businesses generate vast amounts of data from various sources like sales transactions, social media, and customer interactions. Big data analytics tools can help process and analyse this data, providing valuable insights that can be used to improve forecasting models.

II. REVIEW OF LITERATURE

Demand Forecasting Methods

Qualitative Methods: Leveraging Expertise

Qualitative methods rely on experience and judgment. Some common approaches include:

1. Personal Experience: For small businesses with stable demand and straightforward supply chains, managers can leverage their past experiences to estimate future needs. For instance, a pizza shop owner might use past sales records to forecast demand for the upcoming weekend.

2. Delphi Technique: This method involves a panel of experts who anonymously forecast future demand. Each expert's prediction is shared with the group for review and possible revision. This iterative process continues until a consensus is reached.

3. Market Research: Here, surveys and interviews are used to gauge customer preferences and predict future demand. This is particularly valuable for new product launches. However, careful questionnaire design is essential to avoid biased data.

4. Scenario Planning: By considering various potential future scenarios (from best-case to worst-case), this method helps businesses prepare for fluctuations in demand

Quantitative Methods: Data-Driven Insights

Quantitative methods utilize data to predict demand. They consider factors like:

1. Time series analysis—this forecasting method uses historical sales data to predict future demand. It is reasonable to believe that future demand follows similar trends or patterns of previous sales. This is usually true when there is not much volatility or turbulence in market conditions.

2. Causal Relationships: These methods identify factors influencing demand, such as weather or holidays. For example, a spike in summer temperatures might indicate increased demand for ice cream.

3. Machine Learning: Advanced algorithms like support vector machines and random forests can analyze complex data patterns and generate more accurate forecasts compared to traditional methods.

4. Simulation models can combine qualitative and quantitative methods. By simulating various "what-if" scenarios, managers can assess the impact of potential supply and demand fluctuations. This helps them prepare for unforeseen events, such as economic downturns.

The optimal forecasting method depends on the specific demand pattern and desired level of accuracy. Managers should select methods that best fit their needs. Remember, no forecast is perfect. Generally, short-term forecasts tend to be more accurate than long-term ones, and aggregate data forecasts are more precise than individual item forecasts. **(Supply Chain Analytics, Kurt Y. Liu, 2022)**

Relationship between Demand Forecasting Accuracy and Supply Chain Performance:

Numerous studies have investigated the impact of demand forecasting accuracy on supply chain performance. These studies have employed various methodologies, including empirical analyses, case studies, and simulation models, to understand the complex dynamics between forecasting accuracy and supply chain outcomes.

Empirical Evidence:

A study by Chen et al. (2017) examined the effect of demand forecasting accuracy on inventory performance in the retail industry. The findings revealed a significant positive correlation between forecast accuracy and inventory turnover rates, indicating that more accurate forecasts led to better inventory management practices and improved supply chain efficiency.

Case Studies:

Several case studies have provided insights into the practical implications of demand forecasting accuracy on supply chain operations. For instance, a case study conducted by Smith et al. (2018) analyzed the supply chain of a multinational consumer goods company and found that improving forecast accuracy by 10% led to a 5% reduction in inventory holding costs and a 3% increase in on-time deliveries.

Simulation Models:

Simulation modeling techniques have been widely utilized to assess the impact of demand forecasting accuracy on supply chain performance under different scenarios and conditions. For example, a simulation study by Wang and Li (2020) investigated the effects of forecast accuracy levels on supply chain responsiveness and found that higher accuracy levels resulted in shorter order lead times and improved customer service levels**.**

Similarly, research conducted by Li et al. (2019) focused on the impact of demand forecasting accuracy on production scheduling in manufacturing environments. The study demonstrated that higher forecast accuracy resulted in reduced production lead times, lower work-in-progress inventory levels, and increased production throughput, thereby enhancing overall supply chain performance.

Traditional Time Series Forecasting Methods

Time series analysis leverages historical data to predict future trends. This exploration helps identify recurring patterns like seasonality (e.g., higher sales during holidays) or trends (e.g., gradual increase in demand over time). By recognizing these patterns, we can use them to make informed predictions about future demand. Now, we look at some of the popular time series forecasting methods:

1. Moving Average:

This is a simple and intuitive method that calculates an average demand over a predefined period (e.g., past 3 months, past year).

2. Exponential Smoothing:

This method addresses some limitations of moving averages. It assigns a higher weight to more recent data points, giving greater influence to current trends. Exponential smoothing uses a smoothing constant (alpha) to determine the weight given to past data:

- Higher alpha: Places more emphasis on recent data, reacting faster to changes but potentially becoming more susceptible to noise.
- Lower alpha: Leverages historical data to a greater extent, providing a smoother forecast but potentially missing recent trends.

3. ARIMA (Autoregressive Integrated Moving Average):

This is a more sophisticated statistical model that analyzes historical data to identify patterns and trends. It incorporates three components:

- Autoregressive (AR): Considers the impact of past demand values on future demand.
- Integrated (I): Transforms the data (often by differencing) to achieve stationarity, meaning the data's mean and variance remain constant over time.
- Moving Average (MA): Accounts for the influence of past forecast errors on future predictions.

(Supply Chain Analytics, Kurt Y. Liu, 2022)

Prophet Method:

4. Facebook Prophet is an open-source forecasting tool developed by Facebook's Core Data Science team. It's specifically designed to handle time series data with several key strengths:

- Handles Seasonality: Prophet excels at modeling seasonality in data, including weekly, yearly, and even custom holiday effects. This makes it well-suited for forecasting scenarios where trends and seasonal patterns are evident.
- Interpretability: Prophet provides interpretable results. It breaks down the forecast into components like trend, seasonality, and holidays, allowing you to understand the factors influencing the predictions.
- Ease of Use: Prophet is known for its user-friendliness, and it requires minimal data pre-processing and also offers a simple interface for specifying model parameters. This makes it accessible to a broader range of users, even those without extensive forecasting experience.

(Forecasting at scale, Taylor, S. J., & Letham, B., YouTube, 2018, February 21)

III. OBJECTIVES OF THE STUDY

- Compare the effectiveness of different forecasting methods for sales data of pharmaceutical products in a single pharmacy setting.
- Proposing strategies and best practices for enhancing demand forecasting accuracy to optimize supply chain performance.

IV. RESEARCH METHODOLOGY

4.1. Research Design:

Descriptive research design aims to describe the characteristics of a population or phenomenon. It provides a detailed picture of a situation, often through observing, surveying, or analyzing existing data. Unlike other research designs that focus on cause-and-effect relationships, descriptive research paints a clear picture of "what is" without exploring "why it is."

Here's a breakdown of some key features of descriptive research:

- Focus on Description: The primary goal is to describe the current state of a variable or phenomenon. It answers questions like "what are the characteristics of X?" or "how frequent is Y?"
- Variety of Data Collection Methods: Descriptive research draws on a broad range of data collection methods. Surveys, interviews, observations, document analysis, and archival research are all commonly used tools.
- Quantitative and Qualitative Data: Descriptive research can utilize both quantitative data (numerical) and qualitative data (descriptive, non-numerical). Surveys with closed-ended questions generate numerical data, while open-ended questions or interview transcripts provide qualitative insights.
- Cross-sectional Studies: Typically, descriptive research employs cross-sectional studies, where data is collected at a single point in time. Imagine surveying customer satisfaction at a restaurant on a specific day.
- Subtypes of Descriptive Research: There are various subtypes of descriptive research each with specific purposes:

- Observational Research: Observing and recording behavior in a natural setting. (e.g., observing customer interaction patterns in a store)
- Survey Research: Distributing questionnaires or conducting interviews to collect data from a sample population. (e.g., surveying customer satisfaction with a new product)
- Case Studies: Deep dive into a single individual, group, or event. (e.g., studying the customer journey of a loyal customer)
- Content Analysis: Examining existing data sources like documents, records, or social media posts to understand trends or themes. (e.g., analyzing customer reviews to identify product strengths and weaknesses)

Benefits of Descriptive Research:

- Provides a Baseline Understanding: Descriptive research offers a foundational understanding of a topic or population, laying the groundwork for further investigation.
- Identifies Patterns and Trends: It allows researchers to identify patterns, trends, and relationships within the data which can be valuable for decision-making.
- Cost-Effective and Efficient: Compared to other research designs, descriptive research methods can be relatively inexpensive and time-efficient to conduct.

Limitations of Descriptive Research:

- Limited Causality: Descriptive research doesn't establish cause-and-effect relationships. It identifies what exists but not why it exists.
- Sample Bias: If the sample population is not representative of the larger group, the results may not be generalizable.
- Focus on the Present: Primarily focused on the current state, it doesn't provide insights into future trends or past events.

4.2. Data Collection Method:

Secondary Data:

Secondary data collection involves using data that has already been collected by someone else for a different purpose. This data can come from a variety of sources, both published and unpublished. Here's a breakdown of the different types:

Published Sources:

- Books and Journals: Academic publications, industry reports, and government documents are rich sources of secondary data. They offer reliable and well-documented information on various topics.
- Government Databases: Government agencies collect and maintain extensive data on demographics, economics, health, and more. These databases provide readily accessible data for various research questions.
- Statistical Yearbooks and Reports: International organizations like the World Bank and national statistical agencies publish comprehensive reports with valuable data sets.
- Newspapers and Online News Sources: News articles can be used to understand public opinion, track current events, and gather historical data.

Unpublished Sources:

- Internal Reports and Documents: Organizations often have internal reports, surveys, and customer data that can be valuable for research within the organization's domain.
- Dissertations and Theses: Graduate student research can provide in-depth data and analysis on specific topics.

- Personal Archives and Historical Documents: Historical records, personal diaries, and letters can offer valuable insights into past events and social trends.

Benefits of Using Secondary Data:

There are several advantages to using secondary data collection methods:

- Cost-Effective: It's significantly cheaper than primary data collection, eliminating the need for designing surveys, recruiting participants, or conducting fieldwork.
- Time-Efficient: Existing data allows researchers to bypass the time required for data collection, allowing them to focus on analysis and interpretation.
- Large Data Sets: Secondary data sources often provide vast amounts of information, offering a broader perspective than data collected from a smaller sample.
- Longitudinal Studies: Secondary data sources may provide historical data, enabling researchers to conduct longitudinal studies that analyze trends over time.
- Data Comparability: Using pre-existing data sets allows for easier comparison with other studies that used the same source.

Tips for Effective Secondary Data Collection:

Here are some tips to ensure successful secondary data collection:

- Clearly Define Research Objectives: Having a clear understanding of your research question will help you identify the most relevant secondary data sources.
- Thorough Search Strategy: Utilize online databases, library resources, and search engines to locate relevant data sets.
- Critical Evaluation: Carefully assess the data source for its credibility, accuracy, and potential biases. Check the data collection methodology and publication date.
- Data Documentation: Document the source of the data, including the author, date of publication, and any access restrictions.
- Data Transformation: If necessary, transform the data into a format suitable for your analysis. This might involve cleaning, coding, or aggregating data points.

4.3. Data Collection Tool:

Secondary data resides in a diverse landscape, encompassing both published and unpublished sources. Here's a breakdown of the key categories and tools to navigate them effectively:

- Published Sources:
  - Academic Databases: Online academic databases like Google Scholar, JSTOR, and EBSCOhost provide access to a vast collection of scholarly articles, journals, and books. These offer in-depth research on various topics, often accompanied by valuable datasets within the publications.
  - Government Databases: Government agencies collect and maintain extensive data on everything from demographics and health to economics and education. Many offer user-friendly interfaces and downloadable datasets through their official websites (e.g., U.S. Census Bureau, World Bank Open Data).
  - Statistical Yearbooks and Reports: Reputable organizations like the United Nations, World Bank, and International Monetary Fund publish comprehensive yearbooks and reports containing a wealth of statistical data

on various countries and global trends. These reports are readily available online and offer valuable insights for researchers.

- News & Media Archives: News articles and media archives can be a source of secondary data for understanding public opinion, tracking current events, and gathering historical data. Tools like Factiva and LexisNexis provide access to a vast archive of news content from various sources.

- Unpublished Sources:
    - Organizational Data Archives: Organizations often maintain internal data repositories containing reports, surveys, and customer data. Researchers can inquire internally for access to relevant data sets that may be valuable for research within the organization's domain.
    - University Libraries and Repositories: University libraries and repositories often maintain collections of dissertations, theses, and research reports from graduate students. These can be valuable sources of in-depth data and analysis on specific topics.
    - Historical Societies and Archives: Historical societies, museums, and national archives can house valuable historical documents, personal diaries, and letters. These sources offer unique insights into past events, social trends, and cultural practices.

- Tools for Efficient Search and Retrieval

Beyond the data sources themselves, several tools can facilitate efficient search and retrieval of secondary data:

- Search Engines and Specialized Databases: General search engines like Google Scholar or specialized databases tailored to specific fields (e.g., PubMed for medical literature) can be powerful tools for locating relevant information. Utilizing advanced search techniques like Boolean operators can further refine your search results.
- Data Catalogs and Portals: Many organizations and government agencies maintain data catalogs or portals that list and describe the data sets they hold. These can be helpful for identifying relevant data sets and understanding their contents.
- Data Sharing Platforms: Online platforms like Dataverse or Figshare are dedicated to sharing research data. These platforms can be valuable resources for discovering and accessing datasets shared by other researchers.

Strategies for Selecting and Evaluating Secondary Data

While secondary data offers a wealth of information, careful selection and evaluation are crucial. Here are some key strategies to ensure its effectiveness:

- Relevance: Align the data source with your research question. Does it answer your research needs or provide a foundation for further investigation?
- Quality: Critically evaluate the data's credibility, accuracy, and potential biases. Check the data collection methodology, publication date, and source organization's reputation.
- Documentation: Ensure the data source is properly documented, including information about the author, date of collection, methodology, and any limitations.
- Accessibility: Consider any access restrictions or fees associated with obtaining the data.
- Data Format: Assess if the data format (e.g., spreadsheets, databases) is compatible with your analysis tools and requires any transformation or cleaning.

4.4. Data Analysis Tool:

Python:

Python has emerged as a dominant force in the data analysis world, offering a compelling blend of power, flexibility, and ease of use. Here's why Python shines in this domain:

- Readability and Simplicity: Python's syntax is known for its clarity, resembling natural language. This makes it easier to learn and write code, even for those without extensive programming experience. Beginners can quickly grasp the fundamentals and start analyzing data, while experienced programmers appreciate its conciseness.
- Extensive Data Analysis Libraries: Python boasts a rich ecosystem of libraries specifically designed for data manipulation, analysis, and visualization. Popular examples include:
    - NumPy: Provides powerful multi-dimensional array structures and mathematical functions for efficient numerical computations.
    - Pandas: Offers high-performance data structures like DataFrames and Series, ideal for handling tabular data, performing data cleaning and transformations, and facilitating time series analysis.
    - Matplotlib and Seaborn: Create a wide variety of static, animated, and interactive visualizations to explore and communicate insights from your data.

- Open-Source and Community-Driven: Being open-source, Python is free to use and modify. This fosters a vibrant community that has developed a vast collection of libraries, frameworks, and tutorials, providing ongoing support and innovation for data analysis tasks.
- Machine Learning Integration: The lines between data analysis and machine learning are blurring. Python seamlessly integrates with popular machine learning libraries like Scikit-learn and TensorFlow, allowing you to build and deploy predictive models directly within your data analysis workflow.
- Versatility and Scalability: Python's versatility extends beyond data analysis. It's a general-purpose language, enabling you to automate data collection tasks, build data pipelines, and create web applications to present your findings. As your data needs grow, Python scales well, handling large datasets and complex analyses efficiently.

4.5. Data Source:

Kaggle:

Kaggle is a web platform that serves as a hub for data science enthusiasts. Launched in 2010, it has become a breeding ground for collaboration, learning, and competition in the field.

Beyond its popular data science competitions, Kaggle shines as a rich repository of datasets across various domains. This vast collection empowers data scientists to explore, experiment, and refine their skills.

Kaggle's dataset library boasts several advantages:

- Diversity: The platform offers datasets encompassing a wide range of fields, including finance, healthcare, marketing, and social sciences. Users can find datasets for specific research interests or explore uncharted territories, fostering innovation and discovery.
- Quality: Many datasets on Kaggle are meticulously curated and cleaned by the community, ensuring data integrity and facilitating smooth analysis. Users can also find raw datasets, providing the opportunity for data cleaning and preprocessing practice.
- Accessibility: Datasets are readily downloadable in various formats (CSV, JSON, etc.), allowing users to seamlessly integrate them into their preferred data analysis tools.

- Community-Driven: Kaggle leverages the power of its user base. Uploading datasets is encouraged, constantly expanding the available resources and fostering knowledge sharing within the data science community.

About Dataset:

The dataset is based on over 600,000 pharmacy sales transactions collected between 2014 and 2019. It includes information like the date, time, specific drug brand sold, and quantity sold. This data comes from pharmacy point-of-sale systems.

The data focuses on 57 specific drugs categorized under different classifications within the Anatomical Therapeutic Chemical (ATC) system. These categories include:

- M01AB - Anti-inflammatory and antirheumatic products, non-steroids, Acetic acid derivatives and related substances
- M01AE - Anti-inflammatory and antirheumatic products, non-steroids, Propionic acid derivatives
- N02BA - Other analgesics and antipyretics, Salicylic acid and derivatives
- N02BE/B - Other analgesics and antipyretics, Pyrazolones and Anilides
- N05B - Psycholeptics drugs, Anxiolytic drugs
- N05C - Psycholeptics drugs, Hypnotics, and sedatives drugs
- R03 - Drugs for obstructive airway diseases
- R06 - Antihistamines for systemic use.

The original sales data has been transformed to represent hourly, daily, weekly, and monthly sales volumes. Additionally, the data has been pre-processed to remove outliers, fix missing values, and ensure its quality for analysis.

The objective of the research was to validate different methods and approaches related to sales time series data preparation, analysis and forecasting. The goal was to use these methods to develop recommendations for sales and marketing strategies. These strategies would be based on trends, seasonal patterns, and forecasts for eight different groups of pharmaceutical products.

For each method short-term forecasting was used:

- Short-term forecasting: This approach focused on making predictions for the near future, often using a technique called "rolling forecast" where the model is continuously updated with new data.

The research followed a standard process for time series forecasting, which has three main steps:

- Data preparation: This involved cleaning the sales data, defining how to create new features, and transforming the data into a format suitable for forecasting.
- Data exploration: Here, the researchers analyzed the data to understand its characteristics, such as trends, seasonality, and potential issues.
- Forecasting: This is where the researchers used different methods to predict future sales.

Here's a breakdown of the data preparation steps:

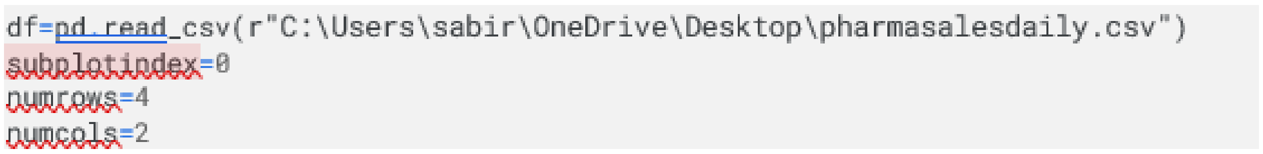
- Cleaning and feature engineering: The researchers cleaned the sales data and decided how to create new features from the existing data.
- Transformation to hourly data: All data was converted into hourly time series, summarizing the total sales for each defined product category (anti-inflammatory, analgesics, etc.) within each hour.
- Anomaly and outlier detection: The researchers identified unusual data points (anomalies) and outliers (extreme values) in consultation with pharmacy staff.
- Missing data imputation: Missing data points were filled in using appropriate methods, such as replacing them with average values or using more sophisticated techniques.
- Rescaling and storage: Finally, the data was rescaled to weekly time series and stored for further analysis.

Time series analysis was used for two main purposes:

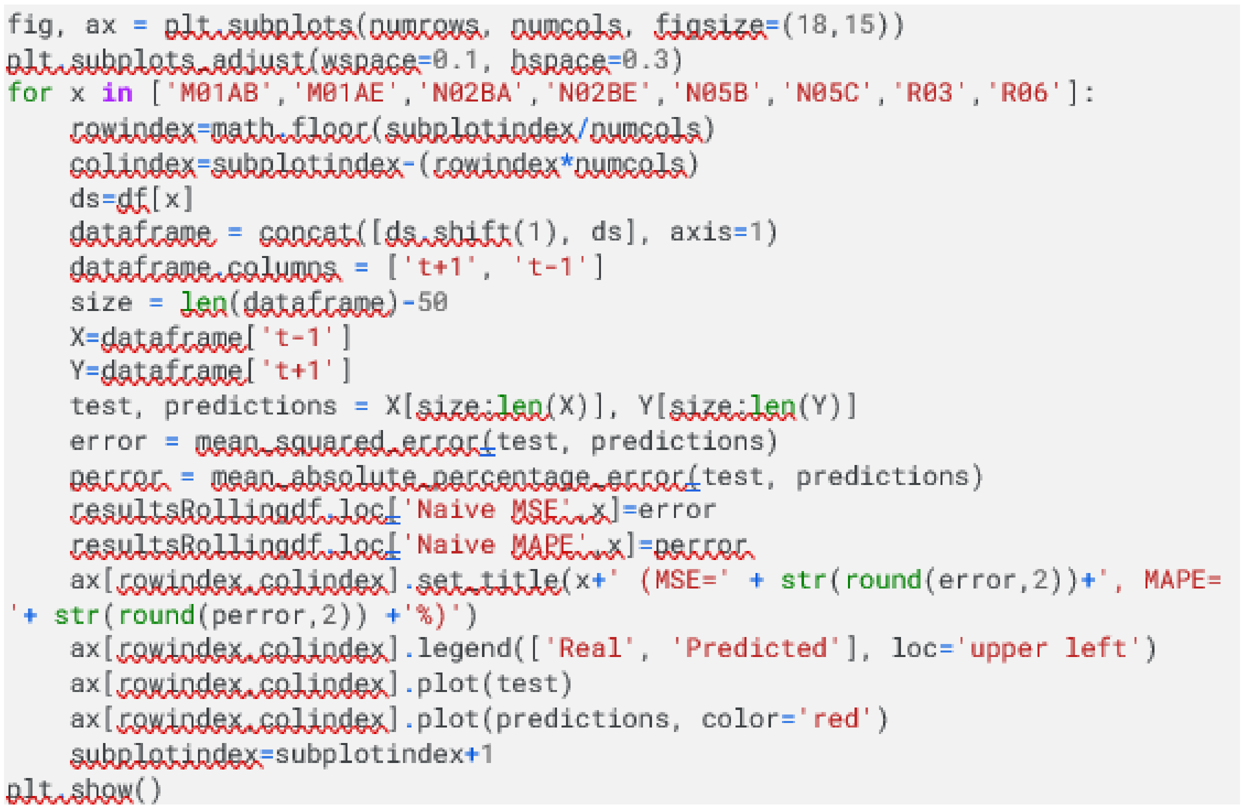
- Identifying sales and marketing opportunities: They analyzed the data at different frequencies (annual, weekly, daily) to find patterns and trends that could be used to improve sales and marketing strategies. This might involve uncovering peak sales periods for certain products or identifying days of the week with lower sales that could benefit from targeted promotions.
- Setting up forecasting models: They analyzed the characteristics of each product group's time series data, focusing on stationarity (whether the data fluctuates around a constant level), autocorrelation (how past sales values are related to current sales), and predictability. This information was used to choose appropriate parameters for the forecasting methods (ARIMA, Prophet, LSTM) that would be applied later.

Naïve forecasting

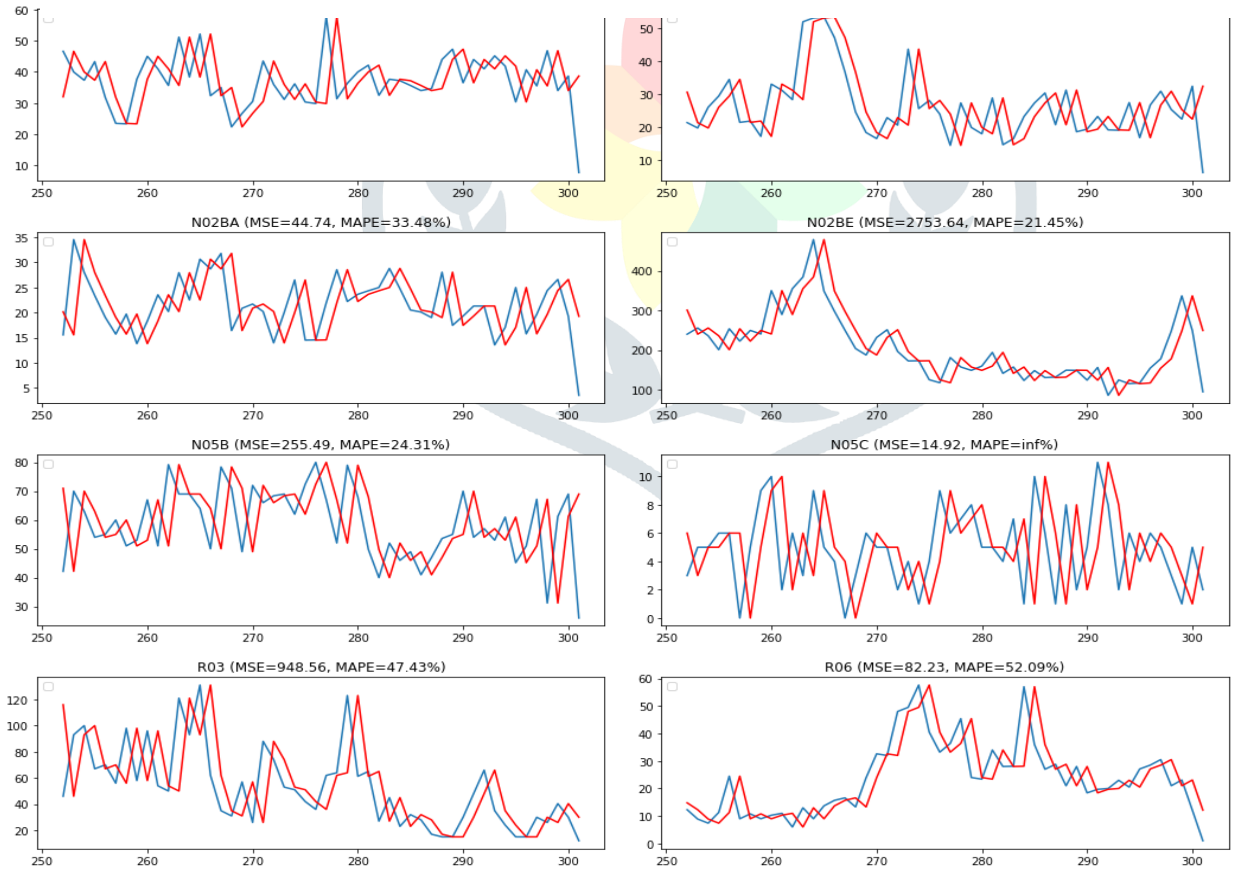
Code:



```
df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")
subplotindex=0
numrows=4
numcols=2
```

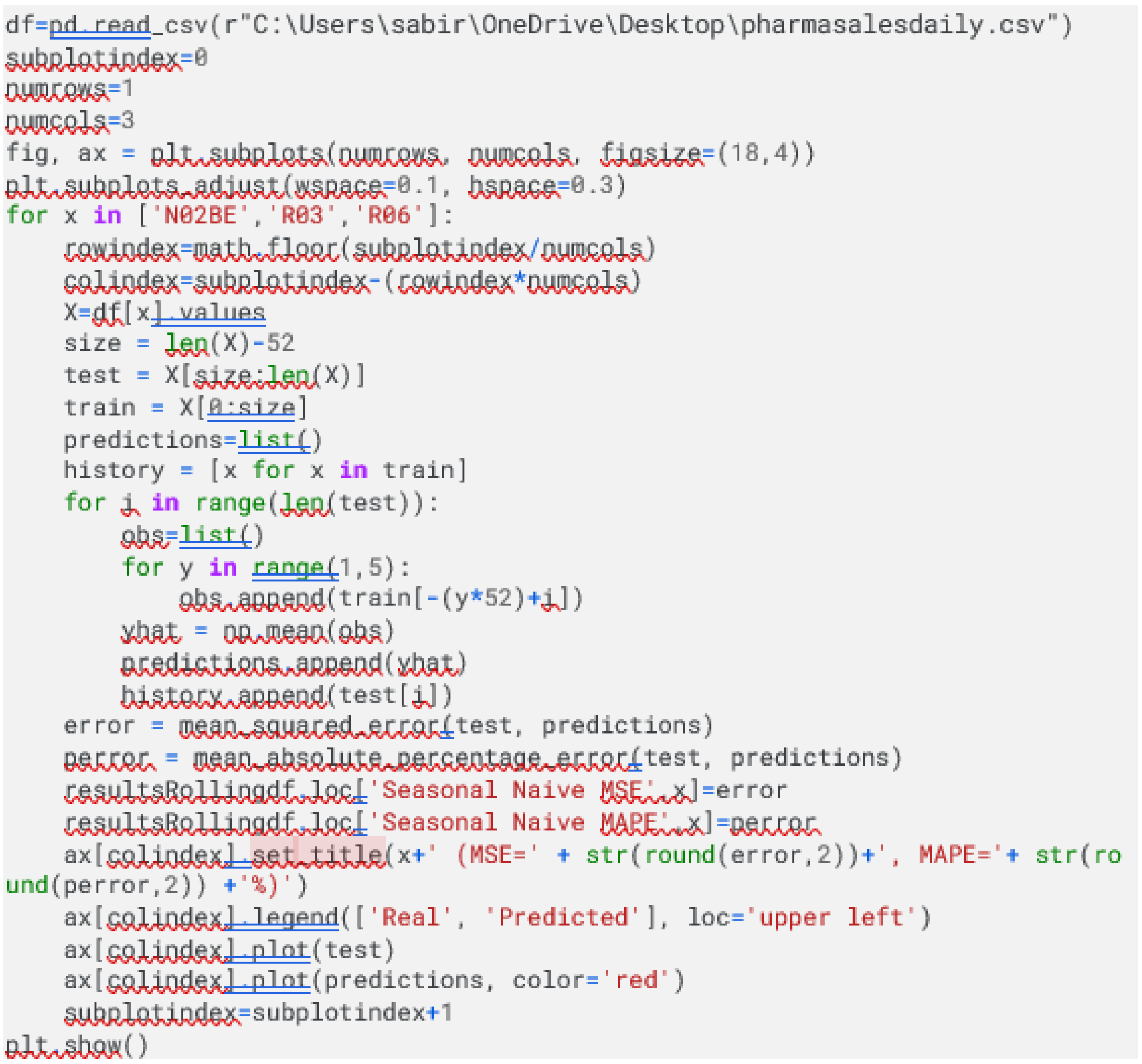


```
fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))
plt.subplots_adjust(wspace=0.1, hspace=0.3)
for x in ['M01AB','M01AE','N02BA','N02BE','N05B','N05C','R03','R06']:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
    ds=df[x]
    dataframe = concat([ds.shift(1), ds], axis=1)
    dataframe.columns = ['t+1', 't-1']
    size = len(dataframe)-50
    X=dataframe['t-1']
    Y=dataframe['t+1']
    test, predictions = X[size:len(X)], Y[size:len(Y)]
    error = mean_squared_error(test, predictions)
    perror = mean_absolute_percentage_error(test, predictions)
    resultsRollingdf.loc['Naive MSE',x]=error
    resultsRollingdf.loc['Naive MAPE',x]=perror
    ax[rowindex,colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE=
'+ str(round(perror,2)) +'%)')
    ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[rowindex,colindex].plot(test)
    ax[rowindex,colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```

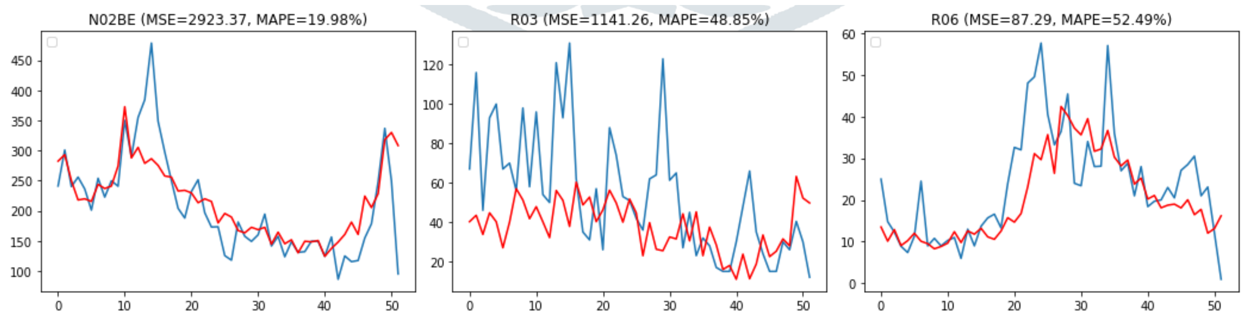


Seasonal Naïve forecasting

Code:



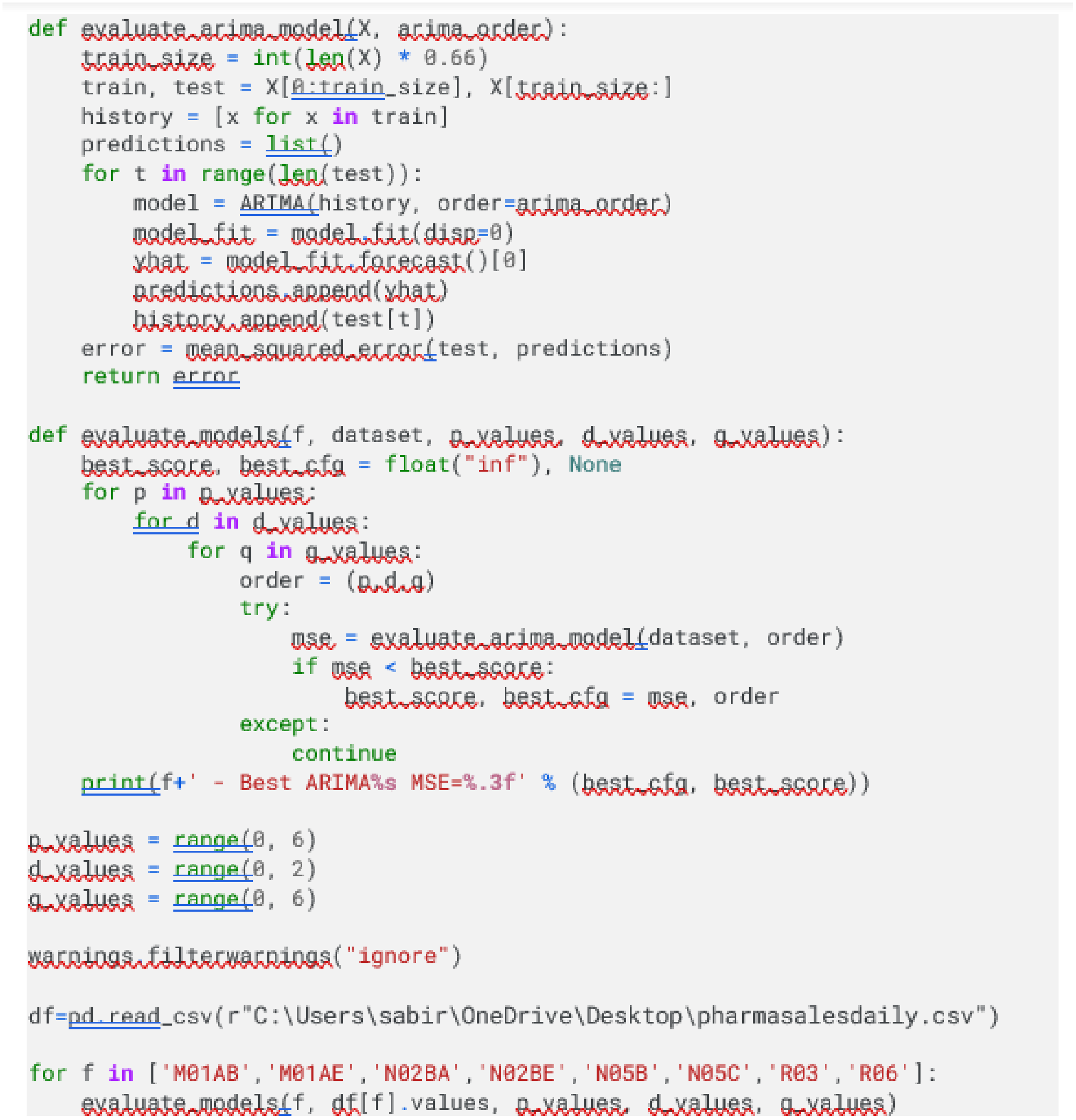
```
df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")
subplotindex=0
numrows=1
numcols=3
fig, ax = plt.subplots(numrows, numcols, figsize=(18,4))
plt.subplots_adjust(wspace=0.1, hspace=0.3)
for x in ['N02BE','R03','R06']:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
    X=df[x].values
    size = len(X)-52
    test = X[size:len(X)]
    train = X[0:size]
    predictions=list()
    history = [x for x in train]
    for i in range(len(test)):
        obs=list()
        for y in range(1,5):
            obs.append(train[-(y*52)+i])
        yhat = np.mean(obs)
        predictions.append(yhat)
        history.append(test[i])
    error = mean_squared_error(test, predictions)
    perror = mean_absolute_percentage_error(test, predictions)
    resultsRollingdf.loc['Seasonal Naive MSE',x]=error
    resultsRollingdf.loc['Seasonal Naive MAPE',x]=perror
    ax[colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE='+ str(ro
und(perror,2)) +'%)')
    ax[colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[colindex].plot(test)
    ax[colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```



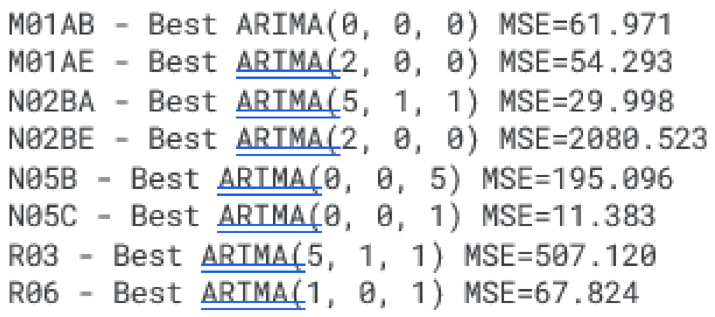
ARIMA

Grid search optimization for short-term forecast

Code:



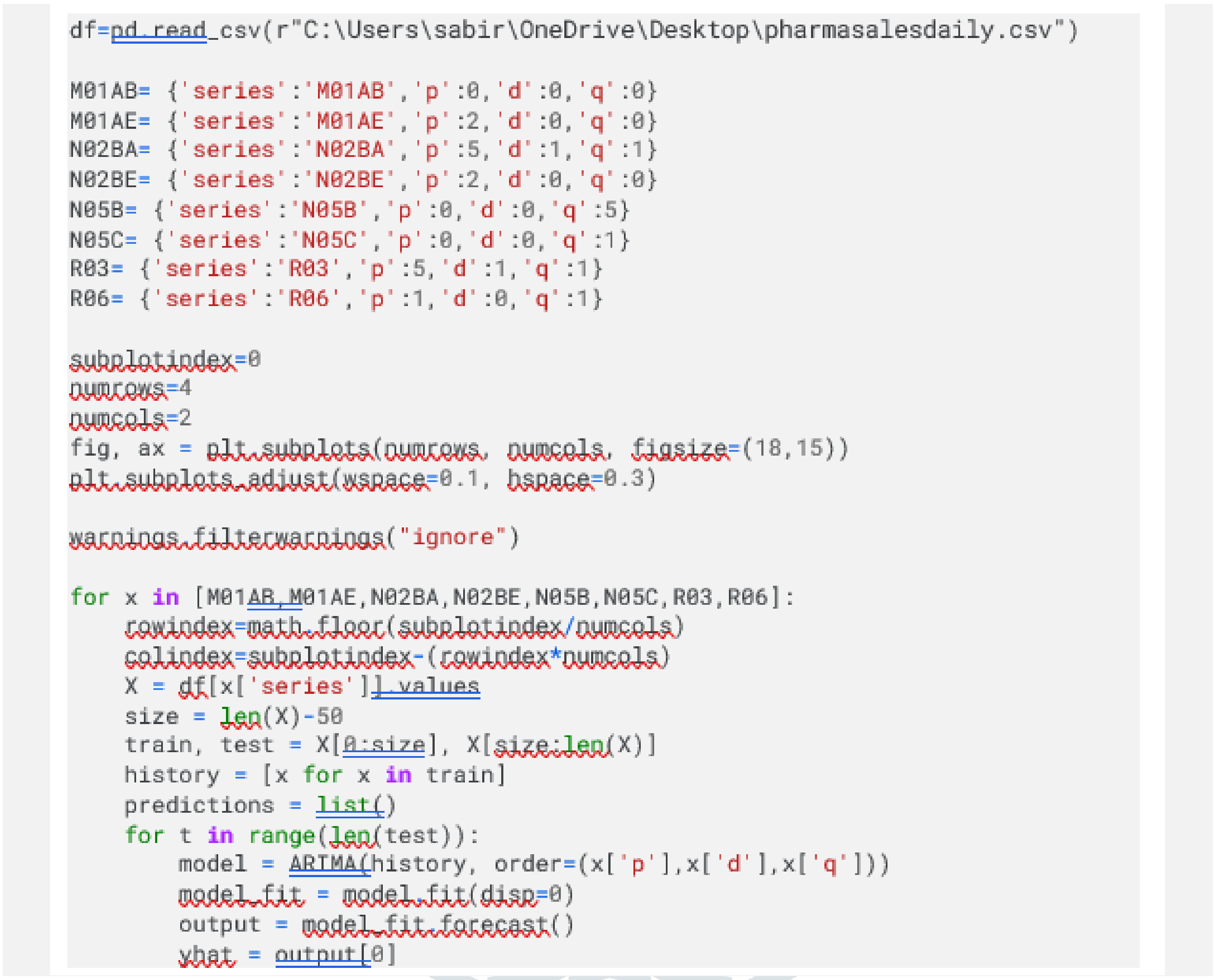
```
def evaluate_arima_model(X, arima_order):
    train_size = int(len(X) * 0.66)
    train, test = X[0:train_size], X[train_size:]
    history = [x for x in train]
    predictions = list()
    for t in range(len(test)):
        model = ARIMA(history, order=arima_order)
        model_fit = model.fit(disp=0)
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        history.append(test[t])
    error = mean_squared_error(test, predictions)
    return error

def evaluate_models(f, dataset, p_values, d_values, q_values):
    best_score, best_cfg = float("inf"), None
    for p in p_values:
        for d in d_values:
            for q in q_values:
                order = (p,d,q)
                try:
                    mse = evaluate_arima_model(dataset, order)
                    if mse < best_score:
                        best_score, best_cfg = mse, order
                except:
                    continue
    print(f+' - Best ARIMA%s MSE=%.3f' % (best_cfg, best_score))

p_values = range(0, 6)
d_values = range(0, 2)
q_values = range(0, 6)

warnings.filterwarnings("ignore")

df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")

for f in ['M01AB','M01AE','N02BA','N02BE','N05B','N05C','R03','R06']:
    evaluate_models(f, df[f].values, p_values, d_values, q_values)
```



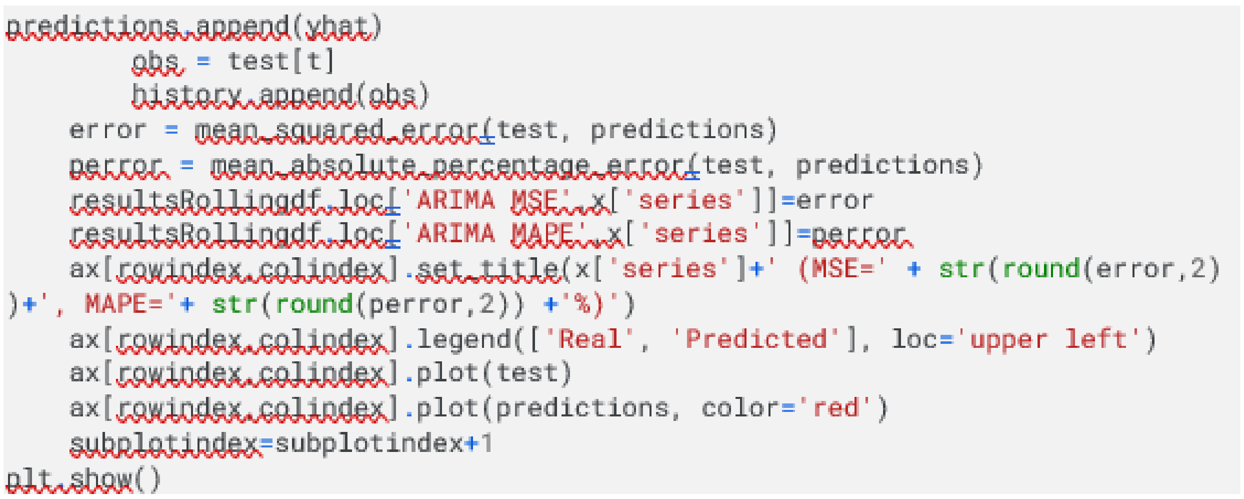
```
M01AB - Best ARIMA(0, 0, 0) MSE=61.971
M01AE - Best ARIMA(2, 0, 0) MSE=54.293
N02BA - Best ARIMA(5, 1, 1) MSE=29.998
N02BE - Best ARIMA(2, 0, 0) MSE=2080.523
N05B - Best ARIMA(0, 0, 5) MSE=195.096
N05C - Best ARIMA(0, 0, 1) MSE=11.383
R03 - Best ARIMA(5, 1, 1) MSE=507.120
R06 - Best ARIMA(1, 0, 1) MSE=67.824
```

Short-term forecasting with ARIMA

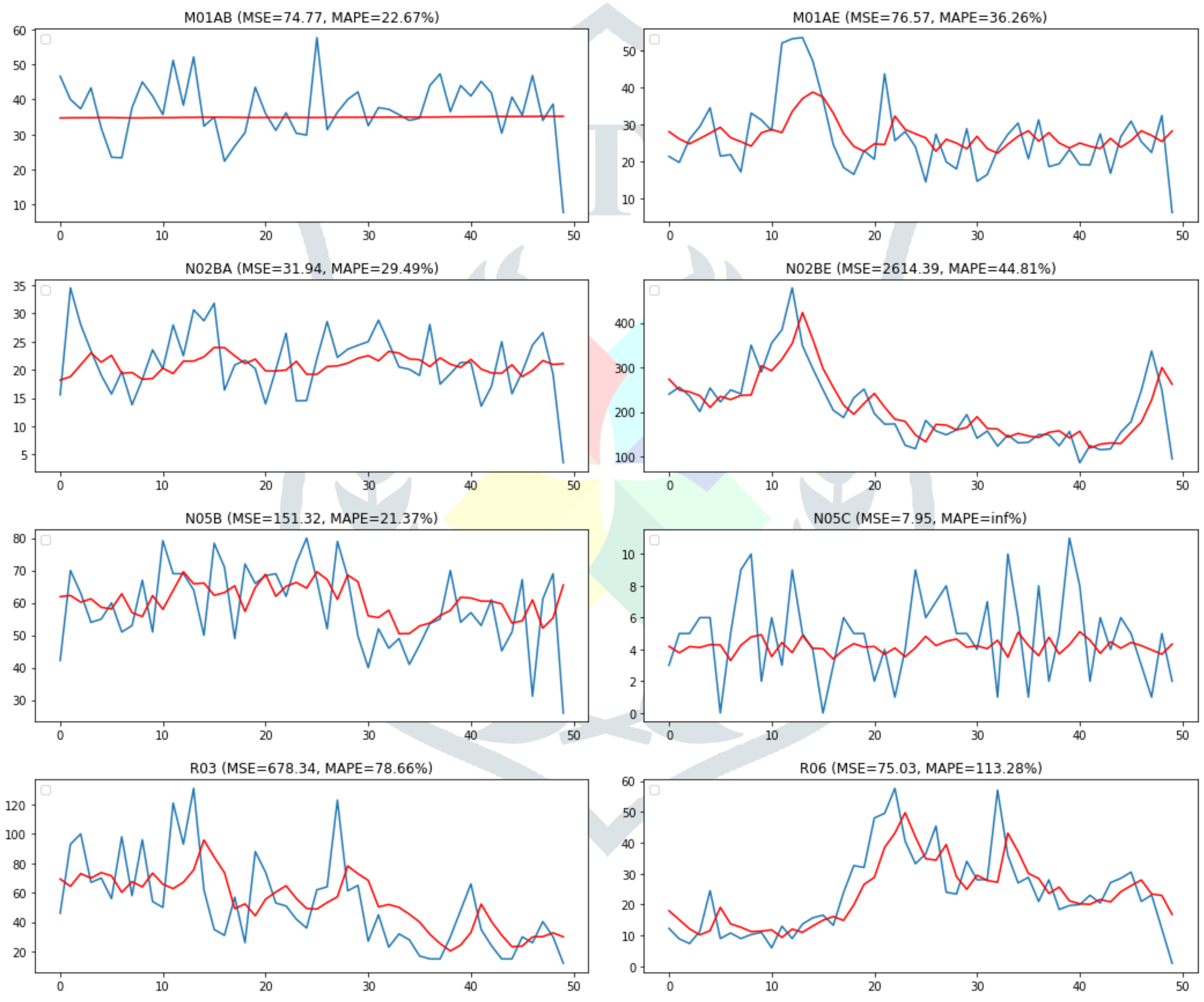
Code:



```
df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")

M01AB= {'series':'M01AB','p':0,'d':0,'q':0}
M01AE= {'series':'M01AE','p':2,'d':0,'q':0}
N02BA= {'series':'N02BA','p':5,'d':1,'q':1}
N02BE= {'series':'N02BE','p':2,'d':0,'q':0}
N05B= {'series':'N05B','p':0,'d':0,'q':5}
N05C= {'series':'N05C','p':0,'d':0,'q':1}
R03= {'series':'R03','p':5,'d':1,'q':1}
R06= {'series':'R06','p':1,'d':0,'q':1}

subplotindex=0
numrows=4
numcols=2
fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))
plt.subplots_adjust(wspace=0.1, hspace=0.3)

warnings.filterwarnings("ignore")

for x in [M01AB,M01AE,N02BA,N02BE,N05B,N05C,R03,R06]:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
    X = df[x['series']].values
    size = len(X)-50
    train, test = X[0:size], X[size:len(X)]
    history = [x for x in train]
    predictions = list()
    for t in range(len(test)):
        model = ARIMA(history, order=(x['p'],x['d'],x['q']))
        model_fit = model.fit(disp=0)
        output = model_fit.forecast()
        yhat = output[0]
```

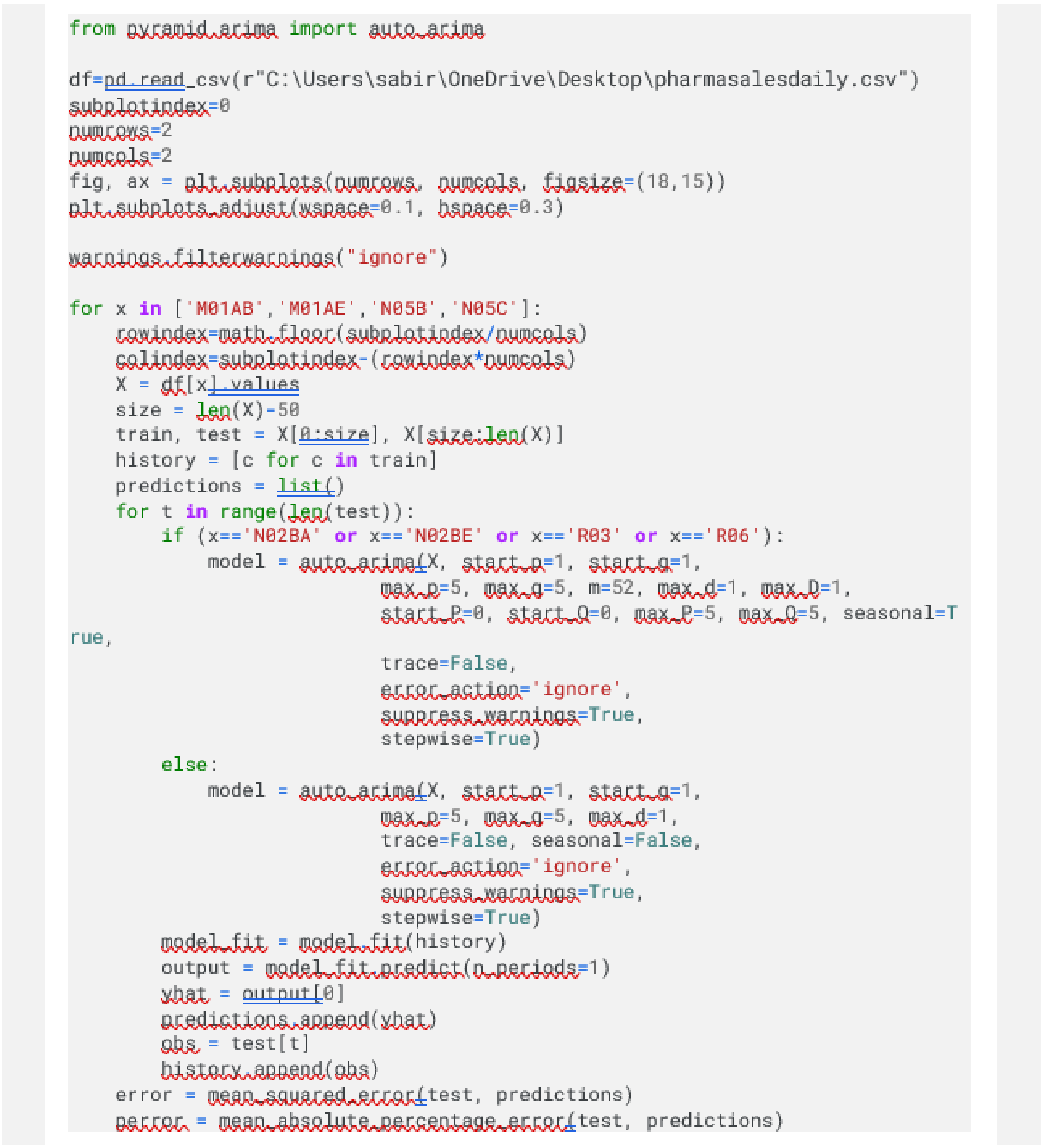


```
predictions.append(yhat)
        obs = test[t]
        history.append(obs)
    error = mean_squared_error(test, predictions)
    perror = mean_absolute_percentage_error(test, predictions)
    resultsRollingdf.loc['ARIMA MSE',x['series']]=error
    resultsRollingdf.loc['ARIMA MAPE',x['series']]=perror
    ax[rowindex,colindex].set_title(x['series']+' (MSE=' + str(round(error,2)
)+', MAPE='+ str(round(perror,2)) +'%)')
    ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[rowindex,colindex].plot(test)
    ax[rowindex,colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```

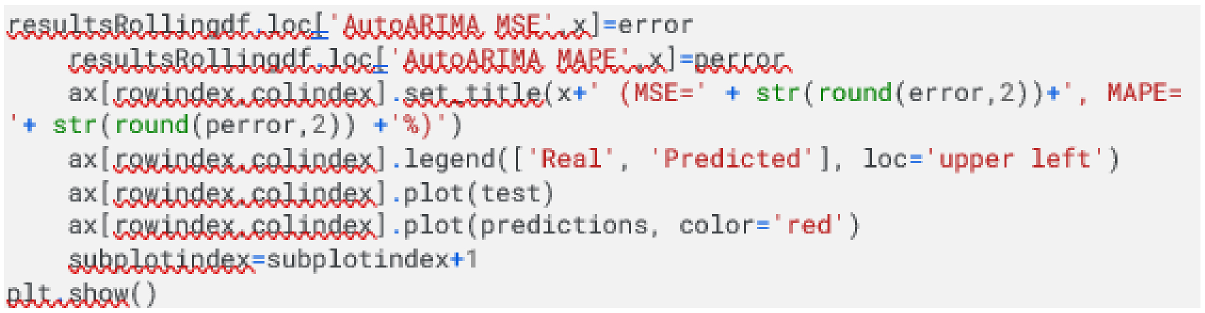


Short-term forecasting with Auto-ARIMA model

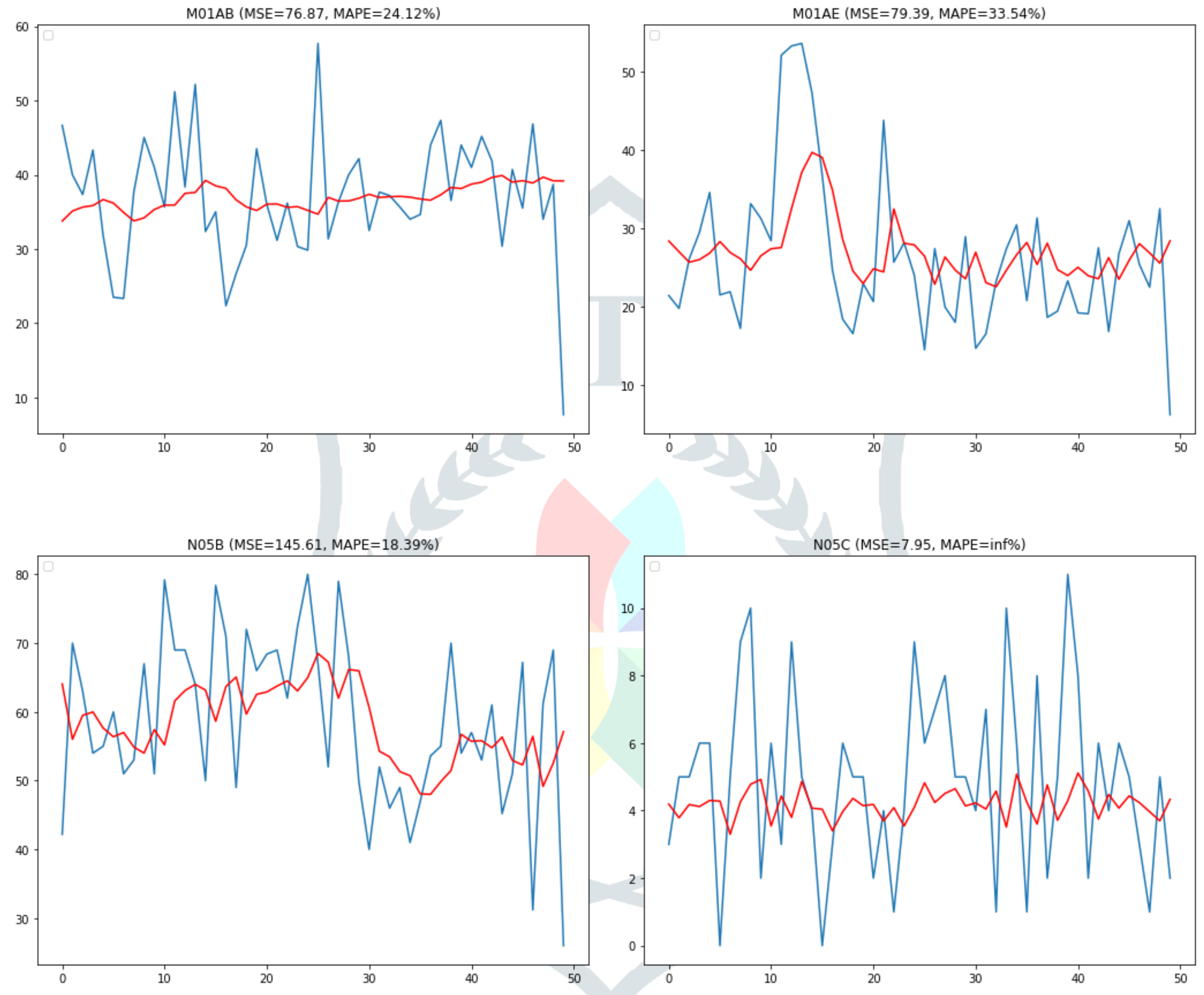
Code:



```
from pyramid.arima import auto_arima

df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")
subplotindex=0
numrows=2
numcols=2
fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))
plt.subplots_adjust(wspace=0.1, hspace=0.3)

warnings.filterwarnings("ignore")

for x in ['M01AB','M01AE','N05B','N05C']:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
    X = df[x].values
    size = len(X)-50
    train, test = X[0:size], X[size:len(X)]
    history = [c for c in train]
    predictions = list()
    for t in range(len(test)):
        if (x=='N02BA' or x=='N02BE' or x=='R03' or x=='R06'):
            model = auto_arima(X, start_p=1, start_q=1,
                               max_p=5, max_q=5, m=52, max_d=1, max_D=1,
                               start_P=0, start_Q=0, max_P=5, max_Q=5, seasonal=T
rue,
                               trace=False,
                               error_action='ignore',
                               suppress_warnings=True,
                               stepwise=True)
        else:
            model = auto_arima(X, start_p=1, start_q=1,
                               max_p=5, max_q=5, max_d=1,
                               trace=False, seasonal=False,
                               error_action='ignore',
                               suppress_warnings=True,
                               stepwise=True)
        model_fit = model.fit(history)
        output = model_fit.predict(n_periods=1)
        yhat = output[0]
        predictions.append(yhat)
        obs = test[t]
        history.append(obs)
    error = mean_squared_error(test, predictions)
    perror = mean_absolute_percentage_error(test, predictions)
```



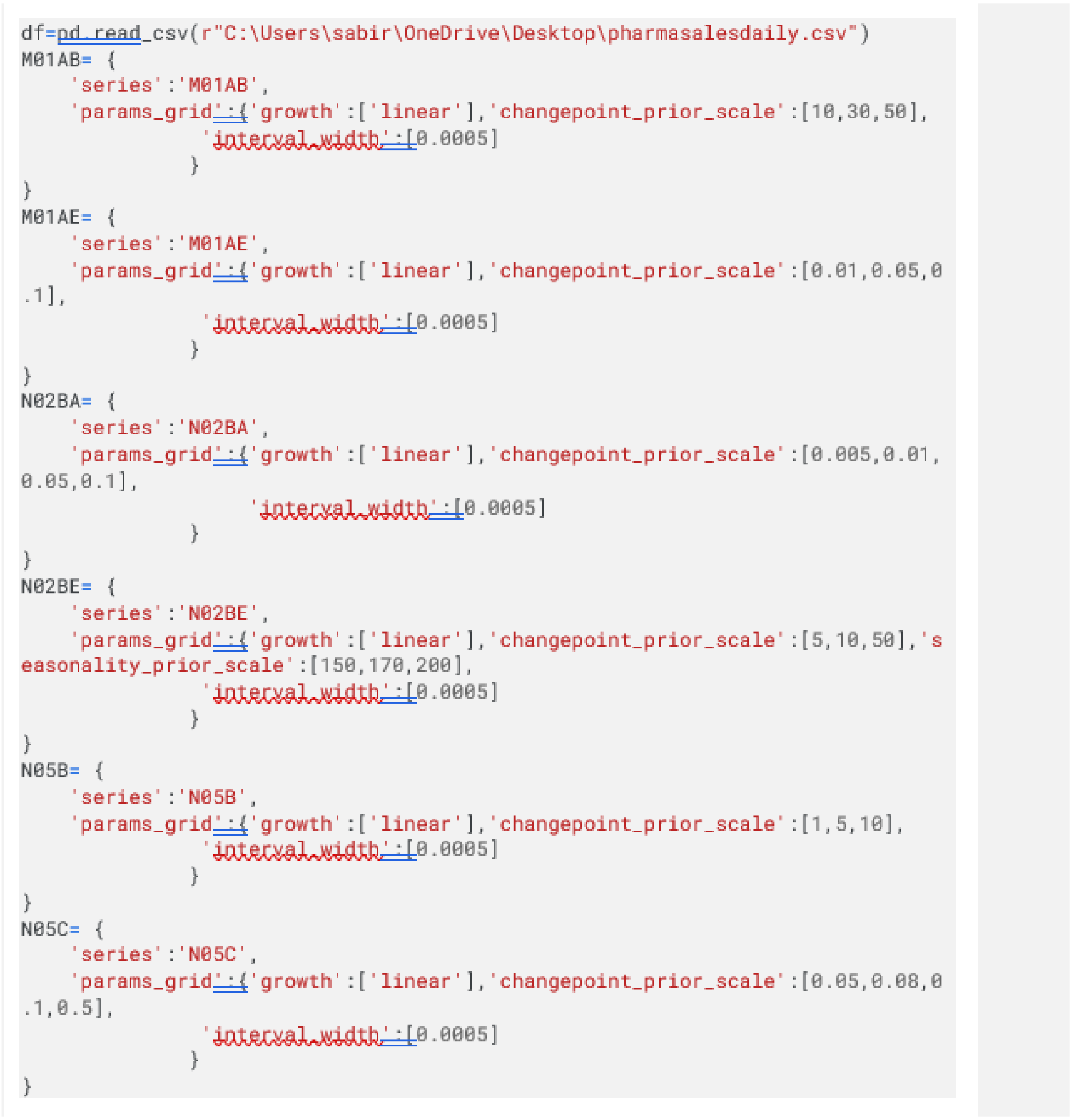
```
resultsRollingdf.loc['AutoARIMA MSE',x]=error
    resultsRollingdf.loc['AutoARIMA MAPE',x]=perror
    ax[rowindex,colindex].set_title(x+' (MSE=' + str(round(error,2))+', MAPE=
'+ str(round(perror,2)) +'%)')
    ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[rowindex,colindex].plot(test)
    ax[rowindex,colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```



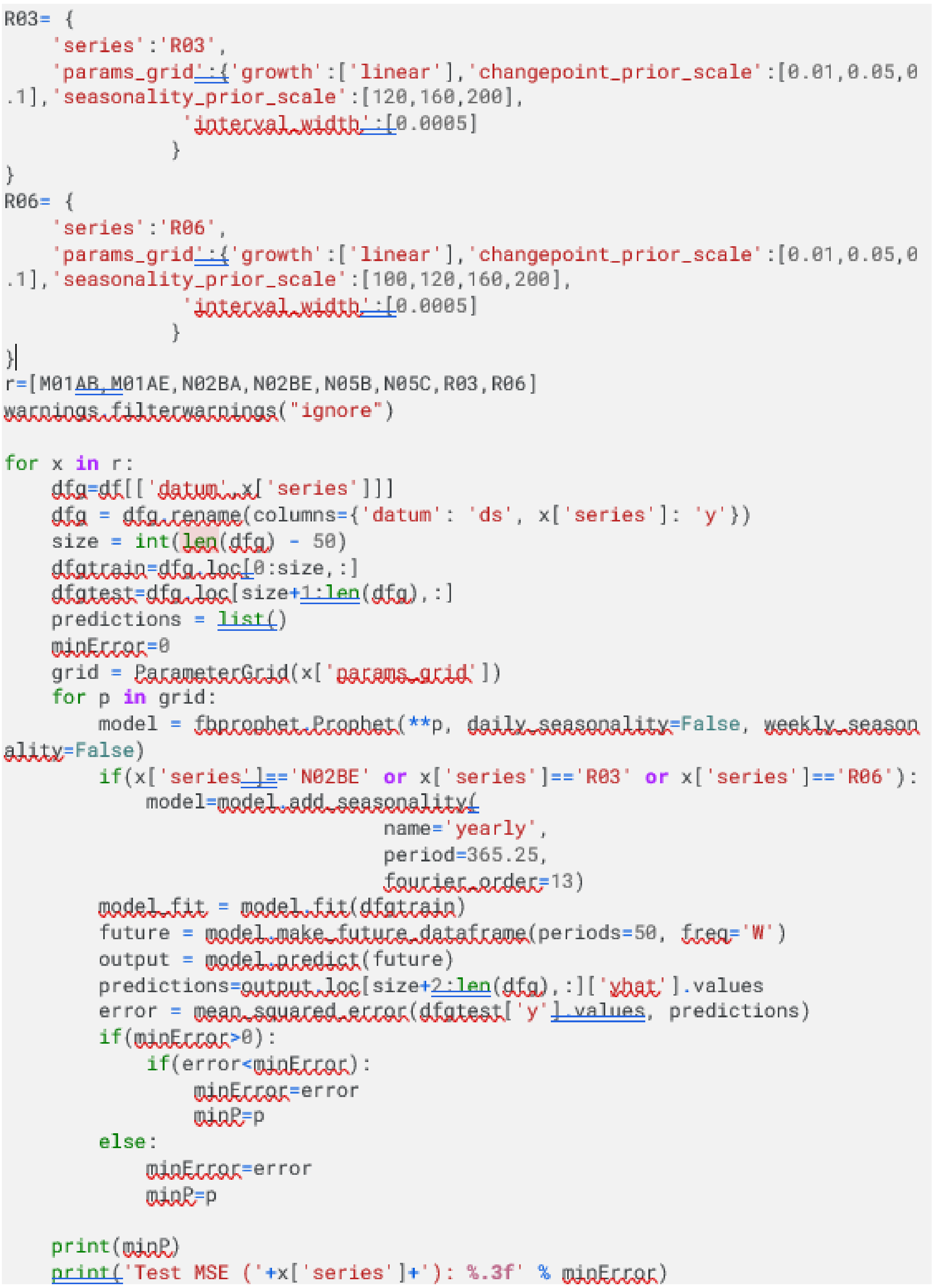
Prophet forecasting

Grid-search optimization of Prophet hyper-parameters

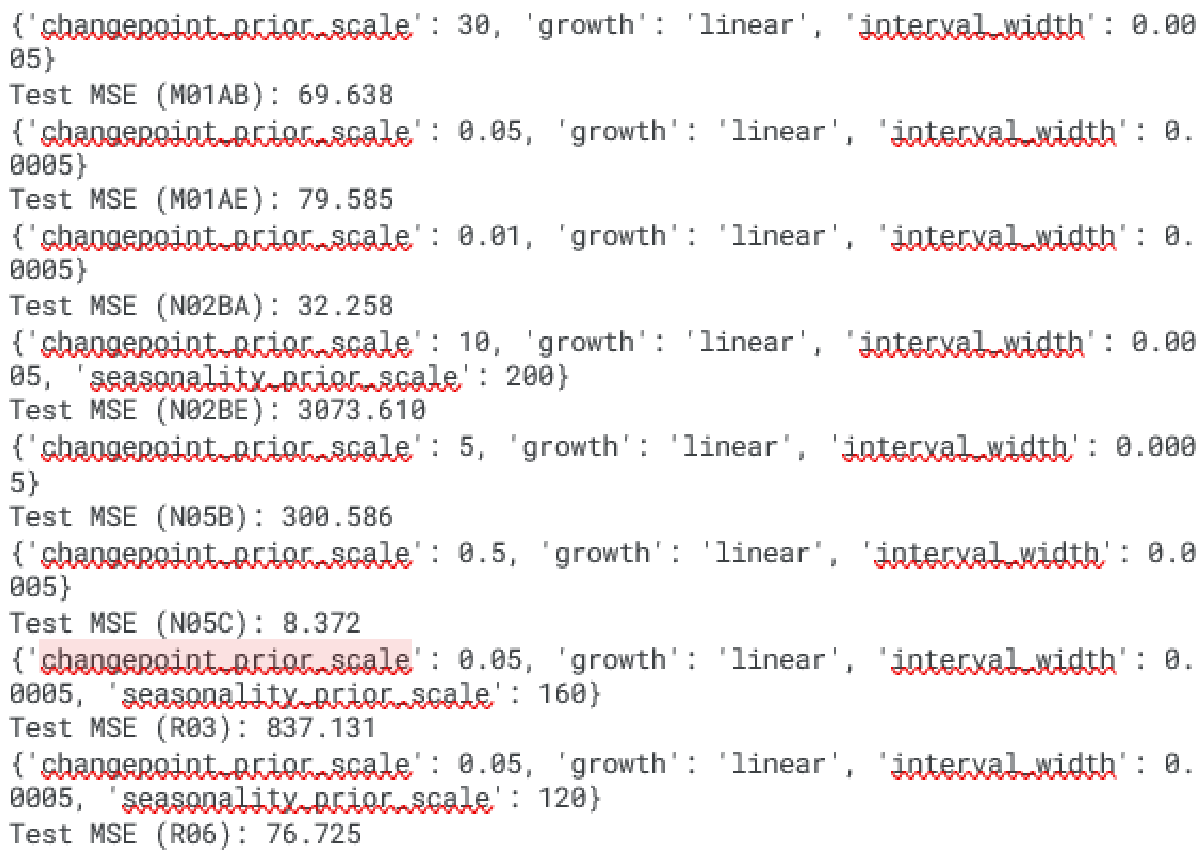
Code:



```
df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")
M01AB= {
    'series':'M01AB',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[10,30,50],
                 'interval_width':[0.0005]
              }
}
M01AE= {
    'series':'M01AE',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[0.01,0.05,0
.1],
                 'interval_width':[0.0005]
              }
}
N02BA= {
    'series':'N02BA',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[0.005,0.01,
0.05,0.1],
                     'interval_width':[0.0005]
              }
}
N02BE= {
    'series':'N02BE',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[5,10,50],'s
easonality_prior_scale':[150,170,200],
                 'interval_width':[0.0005]
              }
}
N05B= {
    'series':'N05B',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[1,5,10],
                 'interval_width':[0.0005]
              }
}
N05C= {
    'series':'N05C',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[0.05,0.08,0
.1,0.5],
                 'interval_width':[0.0005]
              }
}
```



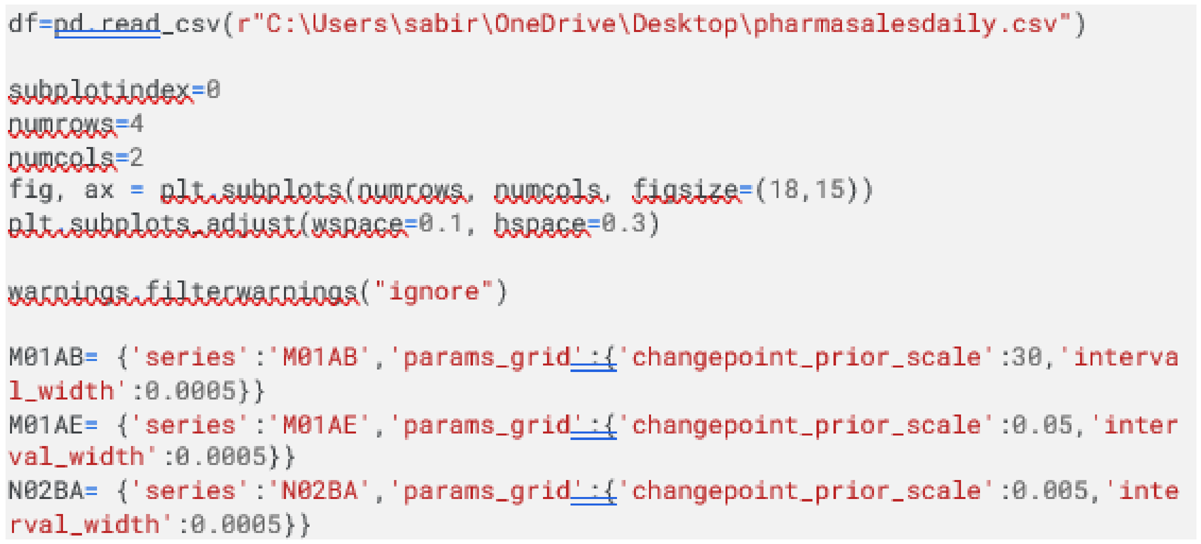
```
R03= {
    'series':'R03',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[0.01,0.05,0
.1],'seasonality_prior_scale':[120,160,200],
               'interval_width':[0.0005]
              }
}
R06= {
    'series':'R06',
    'params_grid':{'growth':['linear'],'changepoint_prior_scale':[0.01,0.05,0
.1],'seasonality_prior_scale':[100,120,160,200],
               'interval_width':[0.0005]
              }
}
r=[M01AB,M01AE,N02BA,N02BE,N05B,N05C,R03,R06]
warnings.filterwarnings("ignore")

for x in r:
    dfg=df[['datum',x['series']]]
    dfg = dfg.rename(columns={'datum': 'ds', x['series']: 'y'})
    size = int(len(dfg) - 50)
    dfgtrain=dfg.loc[0:size,:]
    dfgtest=dfg.loc[size+1:len(dfg),:]
    predictions = list()
    minError=0
    grid = ParameterGrid(x['params_grid'])
    for p in grid:
        model = fbprophet.Prophet(**p, daily_seasonality=False, weekly_season
ality=False)
        if(x['series']=='N02BE' or x['series']=='R03' or x['series']=='R06'):
            model=model.add_seasonality(
                                    name='yearly',
                                    period=365.25,
                                    fourier_order=13)
        model_fit = model.fit(dfgtrain)
        future = model.make_future_dataframe(periods=50, freq='W')
        output = model.predict(future)
        predictions=output.loc[size+2:len(dfg),:]['yhat'].values
        error = mean_squared_error(dfgtest['y'].values, predictions)
        if(minError>0):
            if(error<minError):
                minError=error
                minP=p
        else:
            minError=error
            minP=p

    print(minP)
    print('Test MSE ('+x['series']+'): %.3f' % minError)
```



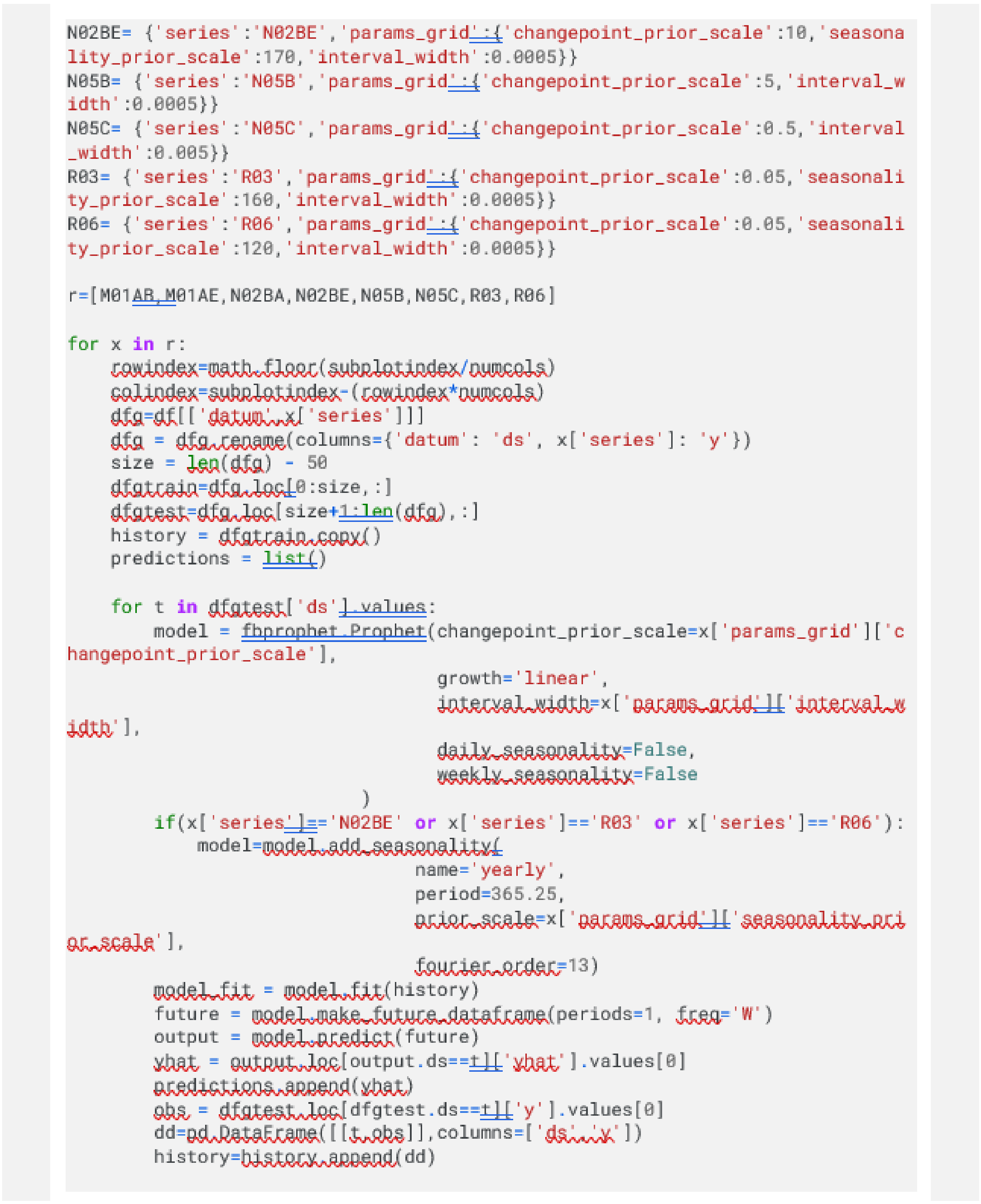
```
{'changepoint_prior_scale': 30, 'growth': 'linear', 'interval_width': 0.00
05}
Test MSE (M01AB): 69.638
{'changepoint_prior_scale': 0.05, 'growth': 'linear', 'interval_width': 0.
0005}
Test MSE (M01AE): 79.585
{'changepoint_prior_scale': 0.01, 'growth': 'linear', 'interval_width': 0.
0005}
Test MSE (N02BA): 32.258
{'changepoint_prior_scale': 10, 'growth': 'linear', 'interval_width': 0.00
05, 'seasonality_prior_scale': 200}
Test MSE (N02BE): 3073.610
{'changepoint_prior_scale': 5, 'growth': 'linear', 'interval_width': 0.000
5}
Test MSE (N05B): 300.586
{'changepoint_prior_scale': 0.5, 'growth': 'linear', 'interval_width': 0.0
005}
Test MSE (N05C): 8.372
{'changepoint_prior_scale': 0.05, 'growth': 'linear', 'interval_width': 0.
0005, 'seasonality_prior_scale': 160}
Test MSE (R03): 837.131
{'changepoint_prior_scale': 0.05, 'growth': 'linear', 'interval_width': 0.
0005, 'seasonality_prior_scale': 120}
Test MSE (R06): 76.725
```

Short-term forecasts with Prophet

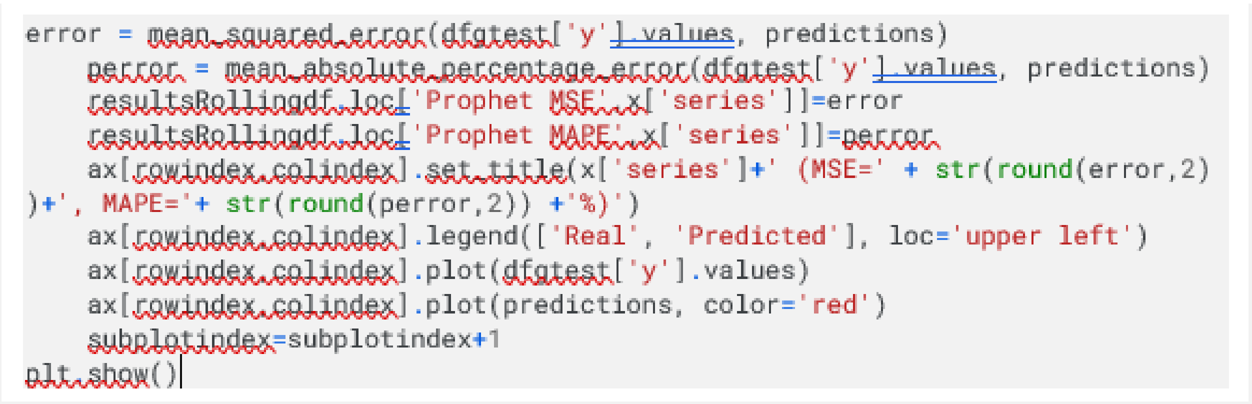
Code:



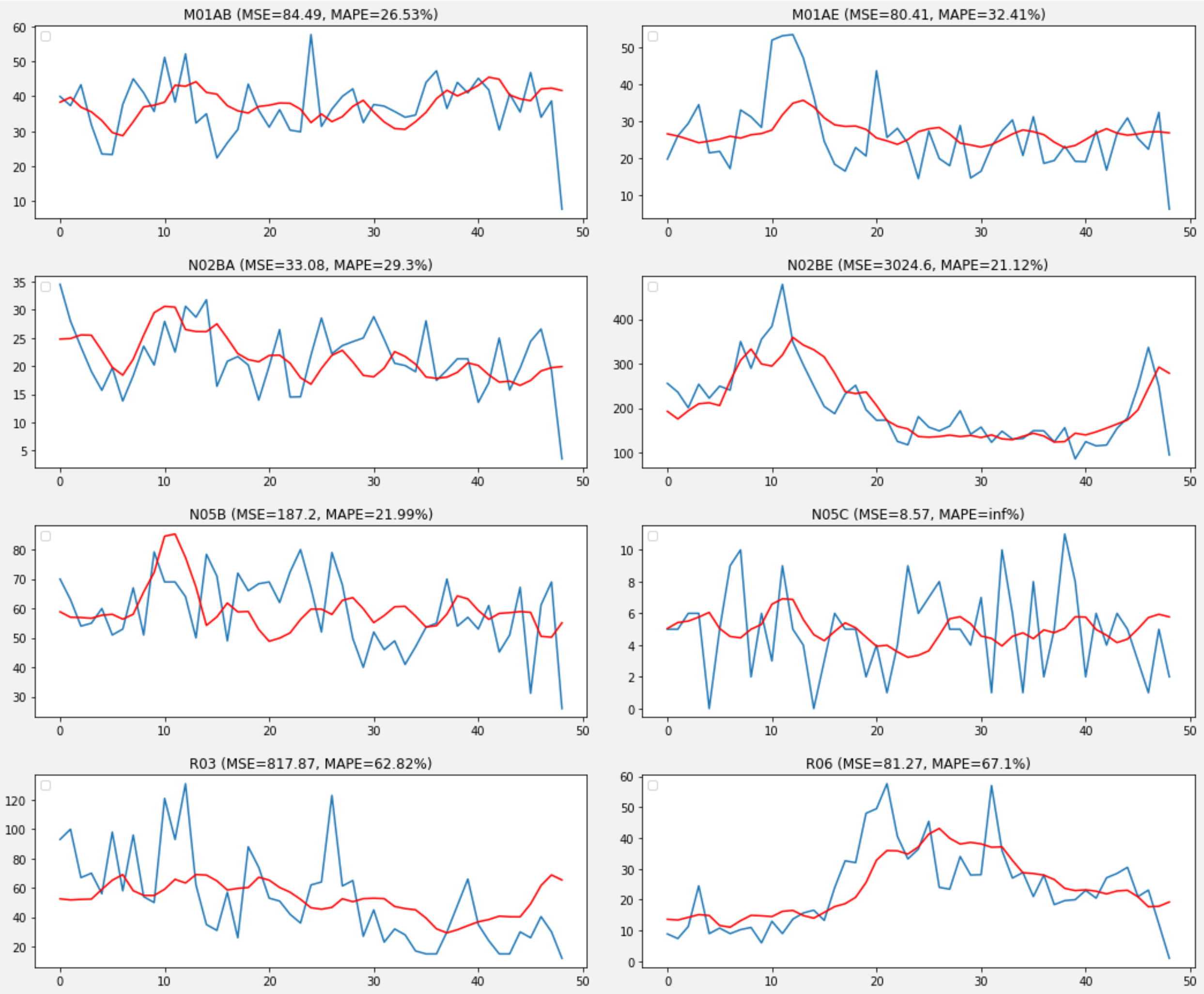
```
df=pd.read_csv(r"C:\Users\sabir\OneDrive\Desktop\pharmasalesdaily.csv")

subplotindex=0
numrows=4
numcols=2
fig, ax = plt.subplots(numrows, numcols, figsize=(18,15))
plt.subplots_adjust(wspace=0.1, hspace=0.3)

warnings.filterwarnings("ignore")

M01AB= {'series':'M01AB','params_grid':{'changepoint_prior_scale':30,'interva
l_width':0.0005}}
M01AE= {'series':'M01AE','params_grid':{'changepoint_prior_scale':0.05,'inter
val_width':0.0005}}
N02BA= {'series':'N02BA','params_grid':{'changepoint_prior_scale':0.005,'inte
rval_width':0.0005}}
```



```
N02BE= {'series':'N02BE','params_grid':{'changepoint_prior_scale':10,'seasona
lity_prior_scale':170,'interval_width':0.0005}}
N05B= {'series':'N05B','params_grid':{'changepoint_prior_scale':5,'interval_w
idth':0.0005}}
N05C= {'series':'N05C','params_grid':{'changepoint_prior_scale':0.5,'interval
_width':0.005}}
R03= {'series':'R03','params_grid':{'changepoint_prior_scale':0.05,'seasonali
ty_prior_scale':160,'interval_width':0.0005}}
R06= {'series':'R06','params_grid':{'changepoint_prior_scale':0.05,'seasonali
ty_prior_scale':120,'interval_width':0.0005}}

r=[M01AB,M01AE,N02BA,N02BE,N05B,N05C,R03,R06]

for x in r:
    rowindex=math.floor(subplotindex/numcols)
    colindex=subplotindex-(rowindex*numcols)
    dfg=df[['datum',x['series']]]
    dfg = dfg.rename(columns={'datum': 'ds', x['series']: 'y'})
    size = len(dfg) - 50
    dfgtrain=dfg.loc[0:size,:]
    dfgtest=dfg.loc[size+1:len(dfg),:]
    history = dfgtrain.copy()
    predictions = list()

    for t in dfgtest['ds'].values:
        model = fbprophet.Prophet(changepoint_prior_scale=x['params_grid']['c
hangepoint_prior_scale'],
                                  growth='linear',
                                  interval_width=x['params_grid']['interval_w
idth'],
                                  daily_seasonality=False,
                                  weekly_seasonality=False
                           )
        if(x['series']=='N02BE' or x['series']=='R03' or x['series']=='R06'):
            model=model.add_seasonality(
                                name='yearly',
                                period=365.25,
                                prior_scale=x['params_grid']['seasonality_pri
or_scale'],
                                fourier_order=13)
        model_fit = model.fit(history)
        future = model.make_future_dataframe(periods=1, freq='W')
        output = model.predict(future)
        yhat = output.loc[output.ds==t]['yhat'].values[0]
        predictions.append(yhat)
        obs = dfgtest.loc[dfgtest.ds==t]['y'].values[0]
        dd=pd.DataFrame([[t,obs]],columns=['ds','y'])
        history=history.append(dd)
```



```
error = mean_squared_error(dfgtest['y'].values, predictions)
    perror = mean_absolute_percentage_error(dfgtest['y'].values, predictions)
    resultsRollingdf.loc['Prophet MSE',x['series']]=error
    resultsRollingdf.loc['Prophet MAPE',x['series']]=perror
    ax[rowindex,colindex].set_title(x['series']+' (MSE=' + str(round(error,2)
)+', MAPE='+ str(round(perror,2)) +'%)')
    ax[rowindex,colindex].legend(['Real', 'Predicted'], loc='upper left')
    ax[rowindex,colindex].plot(dfgtest['y'].values)
    ax[rowindex,colindex].plot(predictions, color='red')
    subplotindex=subplotindex+1
plt.show()
```



Results of Descriptive Statics of Study Variables

Code:

```
from IPython.display import display, HTML
display(HTML(resultsRollingdf.to_html()))
```

	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06
Naive MSE	116.014866	93.875126	44.741150	2753.643864	255.485600	14.920000	948.560347	82.228700
Naive MAPE	28.765667	36.202952	33.475848	21.451644	24.308687	inf	47.432638	52.094536
Seasonal Naive MSE	0.000000	0.000000	0.000000	2923.370345	0.000000	0.000000	1141.258975	87.288057
Seasonal Naive MAPE	0.000000	0.000000	0.000000	19.976880	0.000000	0.000000	48.845563	52.491895
ARIMA MSE	74.768445	76.572892	31.942950	2614.392094	151.322517	7.949828	678.336899	75.030664
ARIMA MAPE	22.672768	36.261902	29.489328	44.809616	21.372122	inf	78.656105	113.282567
AutoARIMA MSE	76.868212	79.386964	0.000000	0.000000	145.608244	7.949828	0.000000	0.000000
AutoARIMA MAPE	24.118497	33.544141	0.000000	0.000000	18.394689	inf	0.000000	0.000000
Prophet MSE	84.485525	80.414991	33.075161	3024.603235	187.198474	8.574808	817.871777	81.267203
Prophet MAPE	26.529136	32.405102	29.301263	21.122458	21.985744	inf	62.824113	

V. RESULTS AND DISCUSSION

For short-term forecast, ARIMA and (Auto-ARIMA for series with seasonal character) outperforms Prophet and is considered as a best candidate for rolling (short-term) sales forecasting.

Time series analysis identified daily, weekly, and annual seasonality patterns in sales data. This information can be leveraged to implement targeted sales and marketing campaigns during peak demand periods. However, product categories N05B and N05C did not exhibit significant regularities, requiring alternative strategies for these products.

Enhancing demand forecasting accuracy is crucial for optimizing supply chain performance and minimizing costs associated with inventory management, production, and distribution. Here are some strategies and best practices to achieve this:

- Data Quality and Integration:
  Ensure data accuracy, consistency, and completeness across all relevant sources, including historical sales data, market trends, and customer feedback.
  Integrate data from various internal and external sources, such as ERP systems, CRM platforms, point-of-sale data, and market research reports, to gain a comprehensive understanding of demand drivers.
- Advanced Analytics and Predictive Modelling:
  Leverage advanced analytics techniques, such as machine learning algorithms and predictive modeling, to analyze large datasets and identify complex demand patterns.
  Use sophisticated forecasting models that can capture both short-term fluctuations and long-term trends in demand, considering factors such as seasonality, promotions, and external market conditions.

- Collaborative Forecasting: - Foster collaboration and information sharing among stakeholders, including sales teams, marketing departments, suppliers, and distributors, to gather diverse perspectives and insights into demand drivers. - Implement collaborative forecasting processes that involve cross-functional teams in demand planning activities, ensuring alignment between sales forecasts, production schedules, and inventory levels.

- Continuous Improvement and Adaptation:
  Establish a culture of continuous improvement by regularly evaluating forecasting accuracy, identifying areas for enhancement, and implementing corrective actions. Monitor and adjust forecasting models in response to changes in market conditions, customer behaviour, and supply chain dynamics, ensuring that forecasts remain relevant and reliable over time.

- Demand Sensing and Real-Time Data Analytics:
  Adopt demand sensing techniques and real-time data analytics to capture sudden changes and emerging trends in demand, enabling more responsive supply chain decision-making. Leverage technologies such as IoT sensors, RFID tracking, and social media monitoring to gather real-time insights into customer preferences, market sentiment, and demand signals.

- Scenario Planning and Risk Management:
  Conduct scenario planning exercises to assess the potential impact of various demand scenarios, such as demand spikes, supply disruptions, or shifts in consumer behaviour.
  Develop contingency plans and risk mitigation strategies to address uncertainties and minimize the adverse effects of demand variability on supply chain performance.

- Cross-Functional Alignment and Communication:
  Foster alignment and communication between different functions within the organization, including sales, marketing, operations, and finance, to ensure a shared understanding of demand forecasting objectives and priorities.
  Encourage open dialogue and information sharing to facilitate collaboration and coordination across the supply chain network, enabling timely responses to changes in demand and supply dynamics.

VI. SUGGESTIONS

- Emphasize the potential for significant improvement in sales forecasting accuracy by incorporating additional relevant data sources. (e.g., weather, economic indicators, promotions)
- Recommend investigating additional accuracy metrics beyond those used in this study.
- Briefly mention a few relevant metrics to consider, such as, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), MAPE (Mean Absolute Percentage Error).
- Optimizing hyperparameters for LSTM models using techniques like grid search or Bayesian optimization could potentially lead to further improvements in forecasting accuracy.
- Machine Learning Integration: Propose research that explores integrating machine learning techniques with traditional forecasting methods to potentially enhance forecast accuracy and capture complex supply chain dynamics.

VII. CONCLUSION

By analyzing the sales data over time (time series analysis) and generating forecasts, we were able to provide valuable insights for the pharmacy. This analysis revealed patterns in how sales fluctuate throughout the year (daily, weekly, and annual seasonality). This information can be used to the pharmacy's advantage by strategically planning sales and marketing campaigns during periods of high demand.

Potential areas for future research could include, investigating the effectiveness of these methods on a larger scale (multiple pharmacies), exploring the incorporation of additional explanatory variables (e.g., weather data, competitor promotions) into multivariate forecasting models, experimenting with advanced deep learning architectures for potentially higher forecasting accuracy.

By implementing the strategies given in the results section, organizations can enhance demand forecasting accuracy, optimize inventory levels, reduce costs, and improve overall supply chain performance. Continuous investment in technology, talent development, and process improvement is essential to sustain and enhance forecasting capabilities over time.

VIII. REFERENCES

[1] Forecasting at scale, Taylor, S. J., & Letham, B., YouTube, 2018
[2] Supply Chain Analytics, Kurt Y. Liu, 2022
[3] Supply Chain Management, Seventh Edition, Sunil Chopra, 2019
[4] Demand Forecasting Best Practices, Nicolas Vandeput, 2023
[5] Supply Chain Foundations: Buy It, Make It, Move It, Eddie Davila, 2022
[6] Effective Demand Forecasting in Health Supply Chains: Emerging Trend, Enablers, and Blockers, Subramanian, L, 2021
[7] Business forecasting, 7thed, Hanke, J.E.; Wichern,D.E.; Reitsch, A.G., 2001
[8] Forecasting: Methods and Applications, Makridakis, S.;Wheelwright, S.C.;Hyndman,R.J. 1998
[9] Supply chain strategies to manage volatile demand, SCM Pulse, Gangadharan, R., 2007
[10] Balancing demand and supply with effective sales and operations planning, Supply Chain Market, 2018,
[11] Demand planning, how it can help improve SCM, Michigan State University, 2019
[12] The difference between MSE and MAPE, Stack Exchange, Hyndman, R., 2011
[13] Demand planning, TechTarget, Rouse, M., 2010