Demand Forecasting using ARIMA

Demand Forecasting Using Time Series Analysis to Optimize Inventory

Performance Assessment 2

D214 – Data Analytics Graduate Capstone

Jessica Hosey

MSDA, College of Information Technology

Western Governors University

February 8th, 2025

A. Research Question

Research Question: "To what extent can product demand be predicted using time series analysis of *historical sales data?"*

E-commerce has exploded ever since the 2019 COVID-19 pandemic, where small businesses have been able to reach customers without a physical brick-and-mortar store. Along with the help of social media, companies have been able to generate traffic to their websites and have increased their earnings ever since. According to the International Trade Administration, there has been approximately a 20% increase in sales in the following 2 years after 2019. This has affected traditional businesses; however, more people are purchasing goods online as it is more convenient and does not require shoes or standing in line at the checkout. People are more willing to wait for products via shipping than ever before. As someone who hates dealing with slow walkers, I prefer shopping online to a traditional store. Ultimately, e-commerce is here to stay and will continue to grow, and we need to build effective machine learning prediction models to help these companies plan for the flow of product demand.

This research project will use a dataset with transactional data from December 2010 to December 2011 to practice building such a model. This dataset is based on a United Kingdom e-commerce business and contains various products to predict future demand using ARIMA time-series modeling. As stated in my previous submission, "predictions on inventory would help drive the business decisions about inventory management, marketing campaigns, and procurement (Hosey 2025)."

The hypotheses for this project are as follows:

> *Null Hypothesis* **-** *Product demand cannot be predicted with 90% accuracy (Hosey 2025).*
>
> *Alternate Hypothesis* **-** *Product demand can be predicted with 90% accuracy (Hosey 2025).*

These hypotheses will allow us to set a standard for the resulting prediction models to measure their performance effectively. If the model's results provide a Mean Absolute Percentage Error greater than 10%, this will indicate that we will accept the null hypothesis and reject the alternative hypothesis. If the results provide a Mean Absolute Percentage Error of less than 10%, we reject the null hypothesis and accept the alternative hypothesis.

B. Data Collection

The primary data source for this project is historical transactional data that was not collected but downloaded from the UCI Machine Learning Repository. This dataset for a UK-based e-commerce business contains the following columns: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country. This dataset is housed on the UCI website, where several datasets are available to the public and are licensed by Creative Commons Attribution 4.0 International. This license allows UCI to share the dataset as long as the user credits the dataset owner.

This dataset contains 282,959 transactions that occurred between December 2010 to December 2011. An advantage of this collection method is that we do not need to contact the store owner to use the dataset as it is under that license, and it is large enough to complete the analysis with over 200k transactions. A disadvantage to data collection methods is that we do not know the history of this business, what it specializes in, or the name of it. This information could give us insight into the company's mission, goals, and values. Additionally, the dataset only contains one year of sales, and ideally, we would instead use multiple years of sales to predict product demand. Therefore, there is a chance that this project will fail, and the product demand will not be successfully predicted. Since there are over 200k transactions, we will continue to complete the analysis regardless, as this challenge will not hinder our ability to forecast product demand.

C. Data Extraction and Preparation

This report is being submitted as a Jupyter notebook, which means all my code is found below and is annotated to give context to what is happening at each step of the data extraction and preparation process. The comments before my code allow the reader to see what is happening to the data and which steps are necessary to clean, transform, and extract information from the original dataset in preparation for ARIMA time series analysis.

Here are all the tools and techniques used for my data extraction and preparation process:

- Python programming language – using the Anaconda Cloud environment
- Jupyter Notebook – Also housed in the Anaconda Cloud environment
- Python Libraries:
    - Pandas
    - NumPy
    - MatPlotLib – pyplot
    - Seaborn
    - Sklearn – train_test_split, mean_squared_error
    - Statsmodels – adfuller, seasonal_decompose, plot_acf, plot_pacf, ARIMA
    - Scipy – signal

Python programming language has several benefits and some disadvantages. One benefit is learning and understanding the basic commands for basic computing tasks is easy. It is easy because Python can download/import several different libraries to help users complete their projects. Between learning Python and R, I felt more comfortable coding in Python. I felt confused when learning R. Python was an easy choice to continue learning, and I do not think I will ever try R again. One main disadvantage of Python is that it can be slower than other languages (like R) when computing execution (Joy 2025).

Anaconda Cloud is a robust environment manager that we can use to save many different projects. It automatically saves your progress, making installing and uninstalling Python packages easy. One disadvantage noticed while using TensorFlow is that it occasionally gets stuck on some applications when downloading and installing/uninstalling some packages. However, this project had no issues with the Anaconda Cloud environment.

Jupyter Notebook is an easy-to-use tool for projects using code. It has several benefits, such as allowing users to see all inputs and outputs on the same screen, duplicate cells (to compare outputs on two ways of transforming data), and moving cells up and down the coding field. These are some of the benefits, and there are more! One disadvantage of Jupyter Notebook is it can be slow in running code. This dataset had 200k rows of data; while executing some steps in my code, I noticed it took the Jupyter Notebook a bit longer to load the dataset and compute more complex tasks, such as forecasting many models.

Pandas is a Python library that allows users to load a dataset into a data frame and complete complex operations on that table. This makes manipulating and viewing the contents of the table very easy. Occasionally, syntax is frustrating when completing more complex tasks as it requires the data to be a data frame, not a series. This was one issue we had while completing this project, and it was easily solved.

NumPy is not a library we use directly; however, MatPlotLib (another Python library) uses NumPy to perform many mathematical calculations on the dataset and create visualizations. This project used NumPy to calculate the mean and create graph trend lines. One major disadvantage of using NumPy is that it establishes null values when using pmdarima to generate auto_arima (the time series tool used to forecast product demand in this project). Due to the creation of null values, this ends up affecting the auto_arima function from working.

MatPlotLib was used to generate all the graphs in this project within the Jupyter Notebook. MatPlotLib makes it easy to create simple to complex graphs that are easy to read. The disadvantage to MatPlotLib is that it can be hard to know how to format specific titles and labels of diagrams. For example, on my decomposition graphs, the axis titles are all jumbled, and you cannot see the actual axis numbers as it does not automatically shift the dates to a horizontal format for easy reading. Also, MatPlotLib does not automatically format graphs (computed in the same cell) into a side-by-side format. Each graph is significant, and one is placed on top of another. Therefore, half of my report requires you to scroll through all the graphs to reach the product demand projections. Another irritating feature of MatPlotLib is that you must tell the computer to show the graph, or it will not appear as an output. Why would you write out all the code for a graph but not want to see it? When scripting for a visualization when scripting for a visualization that it appears as an output without specifically stating to show it.

D. Analysis

The primary technique used in this project was time series analysis, specifically the ARIMA (AutoRegressive Integrated Moving Average) forecasting model. In addition to ARIMA, a few other techniques were used to organize and prepare the data for time series analysis. After loading and cleaning the dataset, the product data needed to be aggregated by daily product demand for the top ten products sold by the e-commerce business. Visualizations of demand trends were created using MatPlotLib and Seaborn. We tested for stationarity on product data once the data was clean and organized. The Augmented Dickey-Fuller test was performed on the top ten products to ensure the data was stationary. To complete the ARIMA models, the product data must be stationary. One of the products, Product 4 – 47566, was not stationary and clearly showed a trend on the Daily Demand graph (graph on the left below).

After completing a stationarity check, the data for each product was split into testing and training datasets before moving on to differencing (again, only one product required this step). Once those data frames were created, Product 4 was differenced to make the data stationary (results after differencing for Product 4 on the right above).

Once stationarity was good for all products, we created seasonal decomposition diagrams for each product using statsmodel. This separated the time series data into trend, seasonality, and residual components. Then, the Autocorrelation (ACF) and Partial Autocorrelation (PACF) Functions plots were used to determine the appropriate AR and MA terms for the ARIMA models. In addition to these diagrams, auto_arima was used to select and automate the ARIMA model selection process. The ACF and PACF graphs are additional support to ensure the auto_arima function selects the correct parameters. The pmdarima auto_arima function quickly analyzes the data provided and chooses the model with the lowest AIC value (Zajic 2022). The following are all the models and AIC values that the auto_arima function chose for each product:

- Product 1 – ARIMA (0, 0, 0) AIC = 3097.223 - *poor performance*
- Product. 2 – ARIMA (1, 1, 2) AIC = 2213.843 - moderate performance
- Product 3 – ARIMA (0, 0, 1) AIC = 2698.300 - *poor performance*

- Product 4 – ARIMA (5, 0, 1) AIC = 2321.269 - moderate performance
- Product 5 – ARIMA (0, 1, 2) AIC = 2330.833 - moderate performance
- Product 6 – ARIMA (0, 0, 0) AIC = 2799.106 - *poor performance*
- Product 7 – ARIMA (1, 1, 1) AIC = 1940.366 - strong performance
- Product 8 – ARIMA (1, 0, 0) AIC = 2481.825 - *poor performance*
- Product 9 – ARIMA (1, 0, 0) AIC = 1958.129 - strong performance
- Product 10 – ARIMA (2, 1, 1) AIC = 2132.085 - moderate performance

After getting the best model for each of the top ten products, we fit the model into the dataset. We calculated the Mean Absolute Error (MAE) to evaluate each model's performance. The lower the MAE value, the better the model's accuracy, and a higher MAE indicates prediction errors. The MAE value indicates how far off, in units, the predictions are compared to the actual sales. The following values are the MAE for each product forecasting (ARIMA) model:

- Product 1 - 131.25 units – *High volatility or model inaccuracy*
- Product 2 - 34.16 units – moderately accurate
- Product 3 - 101.06 units – *unstable demand or model inaccuracy*
- Product 4 - 44.75 units – moderate accuracy but might be too high for business needs
- Product 5 - 42.63 units – moderate accuracy
- Product 6 - 102.34 units – *unstable demand or model inaccuracy*
- Product 7 - 17.20 units – high model accuracy
- Product 8 - 64.52 units – moderate to high accuracy; the model might need to be adjusted
- Product 9 - 18.04 units – high model accuracy
- Product 10 - 27.16 units – low to moderate accuracy

As shown above, several MAE values are very high; for example, the models for products 1, 3, and 6 are above 100. This means that these models have a higher chance of making incorrect predictions. The high AIC values could be due to overfitting the model, or it could be due to not having enough data. This dataset is only for one year of store operation and, therefore, can influence how the models predict future data points. Product 2, 7, 9, and 10 models are very low and indicate that they will make more accurate predictions.

The ADF test was chosen because it is a standard statistical test for deciding whether a time series is stationary. Stationary data is needed to use the ARIMA time series modeling. Therefore, we must ensure our data is prepared before using ARIMA, or the model's prediction ability will be inaccurate. The disadvantage to using Adfuller is that it's sensitive to the size of your dataset (Jones 2024). Smaller datasets can provide unreliable results, leading to incorrect ARIMA model predictions.

As for using pmdarima's auto_arima function, it is best for automating the selection of the ARIMA model's p, q, and d parameters without manually calculating each parameter individually. It automatically tries many different models and hyper-tunes them to find the best model that fits the data. This function is a time saver and requires less work for the user. However, the auto_arima function is expensive in terms of dataset size. The larger the dataset, the more time it takes the computer to search for the best ARIMA model (Investopedia 2024). For example, a few product datasets took longer to use auto_arima than others. This was not an issue for this project's timeline but could be for another business project with a much larger dataset. If more data is added to the original dataset (recommended below), it will take much longer for auto_arima to find the best model to use for forecasting.

The mean absolute error was chosen for evaluating our forecast models because it uses the same units as the original data and provides an understandable forecast measure for non-technical people. A disadvantage to using this evaluation tool is that unlike Mean Squared Error (MSE), MAE does not rank significant errors versus small ones (Acharya 2023). Therefore, small and large errors affect the MAE, which is not always ideal for volatile data.

E. Data Summary and Implications

Our focus for this project was forecasting next month's product demand for the top 10 products using historical sales data for a UK-based e-commerce store. The ARIMA time series model tool was selected due to its ability to capture the trends and seasonality of each top 10 product. The models were then evaluated using Mean Absolute Error to measure the effectiveness of the forecasts.

The research question for this project aimed to determine the extent to which product demand can be predicted using time series analysis of historical sales data. The ARIMA models provided reasonable forecasts for all top 10 products that the business sold between December 2010 and December 2011. However, the accuracy of the models can be influenced by the lack of data provided (only one year). Even though there is a large number of historical transaction data, this data is from a single year. Typically, having more than one year of transactional data to forecast product demand would be beneficial.

One limitation is the short time that the dataset contained. Time series models perform better with larger datasets with more than one year of data. Therefore, our results are not as reliable as a forecast based on multiple years of transactional data. Based on this information, we recommend collecting more transactional data to analyze with the ARIMA models.

Two future study recommendations:

1. Refine ARIMA models for products 1, 3, and 6 as they were the lowest performing models in terms of AIC value. This might allow us to use the models to help predict product demand once we add more data to our store's transaction history.
2. Additional Data: Future studies on this topic must contain more than three years of transactional data. This will increase the accuracy and reliability of the models' forecast results. The business would greatly benefit from more accurate product demand forecasts based on at least three years of transactional data versus one year of data.

F. Sources

Acharya, N. (2023, August 30). Choosing between mean squared error (MSE) and mean absolute error (MAE) in regression: A deep dive. Medium. Retrieved February 12, 2025, from https://medium.com/@nirajan.acharya777/choosing-between-mean-squared-error-mse-and-mean-absolute-error-mae-in-regression-a-deep-dive-c16b4eeee603

DataCamp. (2024, December 29). ARIMA for time series forecasting: A complete guide. DataCamp. Retrieved February 8, 2025, from https://www.datacamp.com/tutorial/arima

Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts. https://otexts.com/fpp3/

International Trade Administration. (n.d.). Impact of COVID pandemic on eCommerce. U.S. Department of Commerce. Retrieved February 8, 2025, from https://www.trade.gov/impact-covid-pandemic-ecommerce

Investopedia. (2024, July 31). Autoregressive integrated moving average (ARIMA). Retrieved February 12, 2025, from https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp

Jones, K. (2024, December 18). *Dickey-Fuller test for stationarity in time series with Python*. Medium. Retrieved February 12, 2025, from https://medium.com/@kylejones_47003/dickey-fuller-test-for-stationarity-in-time-series-with-python-4e4bf1953eed

Joy, A. (2025). 5 main disadvantages of Python programming language. Pythonista Planet. Retrieved February 9, 2025, from https://pythonistaplanet.com/disadvantages-of-python/

Zajic, A. (2022, November 29). *What is AIC? Understanding Akaike's Information Criterion*. Retrieved February 12, 2025, from https://builtin.com/data-science/what-is-aic