Data Analytics Report and Executive Summary

Gooden, Nina S. | Student ID #: 009823504

Western Governors University

Dr. William Sewell

December 23, 2022

Abstract

The contribution of this study to the field of Data Analytics and the MSDA program is to create a time series model which will predict product sales within the county. Montgomery County of Maryland collects data on product suppliers, as well as volume of sales for tax purposes. This study will create a forecast that the county can use to better understand the underlying costs, trends, or systemic sales patterns over time (Hayes, 2022). This study will utilize autoregressive integrated moving average (ARIMA) to describe autocorrelations in the data, while also using exploratory data techniques to derive general insights about the dataset. Lastly, the researcher will package their findings in a multimedia presentation for presentation. It is the researcher’s hypothesis that there will be a clear prediction derived from this dataset.

Keywords: Time Series Analysis, ARIMA analysis, forecast, sales prediction, master's degree, degree capstone

Data Analytics Report and Executive Summary

Time series data for business sales tends to contain seasonal fluctuations, cycles, and shifts. The objective of building a time series model is to “extrapolate the dynamic pattern in the data for forecasting future observations, to estimate the effect of known exogenous interventions, and to detect unsuspected interventions” (Tiao, 2015) according to the needs of the entity the model is created for. Through the course of this project, the researcher will seek to create a forecast model for the number of sales in the next quarter for the combined efforts of warehouse and retail for Montgomery County of Maryland. Support for the alternative hypothesis is found in *Time series analysis: validating effect of changes* with the statement “time series hypothesis testing talks about how we identify whether different time periods have significantly different observation” (Tunggawan, 2018). Creating time series analyses in professional settings allows company leaders to make informed decisions on procurement, production, and publishing. Researchers are confident this will be the case for this model as well.

# Research Question

The standing research question for this analysis is as follows: can an Arima time series model be created from the dataset in order to accurately predict next quarter sales for Montgomery County?

## Hypothesis

The null hypothesis of this evaluation is that a time series model with accuracy > 70% cannot be made from the Warehouse and Retail Sales dataset.

The alternative hypothesis of this evaluation is that a time series model with accuracy > 70% can be made from the Warehouse and Retail Sales dataset.

The project will seek to create a forecast model for the number of sales in the next quarter for the combined efforts of warehouse and retail distribution points for Montgomery County of Maryland. Support for the alternative hypothesis is found in Time series analysis: validating effect of changes with the statement “time series hypothesis testing talks about how we identify whether different time periods have significantly different observation.” (Tunggawan, 2018)

# Data Governance

## Data Analysis Goals and Objective

The primary goal of this analysis is to identify patterns in sales data to build a forecast model for the providing entity. The given dataset covers retail and warehouse sales for a period of time between 2017 and 2020. Functionally, the county in question was largely impacted by economic fluctuations during this time period, which has made forecasting challenging. Researchers believe that by creating this model, the county will be better equipped to measure the overall impact of market downturn, plan for future market adjustments, and compare linear predictions to actual performance.

### **Data Collection**

The data needed to be collected for the question is the publicly available information provided by the Montgomery Country of Maryland local government ESB Service *Warehouse and Retail Sales - Catalog.* (n.d.). The data was last updated December 9, 2022 and is updated monthly. There are 307,646 records in the data set and 9 columns.

An advantage of this data collection methodology is that the data is located on a trusted and well-governed website that allows public access. This supports reproduction of the analysis, in addition to continuality of the study.

A disadvantage of using this data collection methodology is the disconnect between research analysts and data gatherers. Any missing data must be handled without further explanation about the circumstances around those hurdles.

#### Data Structure***.***

The initial data set is made available through the data.gov website. The data set includes the following variables of year, month, supplier, item code, item description, item type, retail sales, retail transfers, and warehouse sales. The initial predictor variables are broken down in (Figure 1).

#### The updated data set used in this model was cleaned and truncated to answer the research question. That cleaned data set can be found as (Figure 2).

#### Data Accessibility***.***

Montgomery County of Maryland has made this information publicly available through the data.gov website. The dataset is updated regularly and has been consistently managed since 2017, meaning the data available is healthy and of sufficient size. Forecasting a time series is fundamental for business planning, procurement, and production. Ensuring data is healthy is necessary, as any errors in forecasts will ripple down any business context (Prabhakaran, 2022). There will be no information in the final evaluation that would make the companies listed in these transactions identifiable.

##### Data Limitations

The information available through public forums does not include price data or revenue information. As such, the time series evaluation will need to focus on number of sales, rather than their value.

##### Data Delimitations

There are no delimitations for the cleaned dataset.

### **Data Gathering**

Data will be downloaded from a publicly available CSV file from the data.gov website which shows data for all warehouse and retail sales from Montgomery Country of Maryland. Any entries that have missing inputs for total sales will me imputed through data averages. Missing data in the item\_type field will be manually evaluated and inputted, which is the ideal substitution method for easily located, low-cost data (Mesidor, 2021). Records of missing date information will be removed, as time series models work with the complete data and require impute of the missing values prior to modeling (Kumar, 2022). No other columns will be evaluated in this analysis. The data quality is remarkably high, as it has been collected by local government and maintained for five years. Data will be made stationary if it is not initially and correlation and auto correlation will be plotted visually. Overall data sparsity is < 5%.

## Data Analytics Tools and Techniques

A succinct overview of the steps proposed to accomplish and measure this modeling goal has been developed thusly:

1. Exploratory visualizations will be created for cleaned data to gather insights into which item type contributes what percentage of sales.
2. Initial visualization will be created using pyplot.
3. Stationarity will be evaluated and corrected, if necessary, by evaluating primary statistical method t-test values, using autocorrelation function (ACF) and partial autocorrelation function (PACF) to find the p and q values for ARIMA (Zvornicanin, 2022).
4. If t-statistic is higher than the critical value, the researchers will continue the evaluation.
5. A portion of the data will be segmented for testing and assessment of the model.
6. The ARIMA model will be constructed, with coefficient and p-values evaluated to measure accuracy of model.
7. Model accuracy will be measured against withheld data, as well as mean square root evaluation.
8. Data will be exported in order to create a multimedia presentation in Microsoft PowerPoint.

The auto\_arima and ARIMA functions will be used to create forecast data, though it is important to note that ARIMA has difficult prediction turning points. (Understanding ARIMA Models for Machine Learning. (n.d.)). The available data ends in 2020, which was a mark of a global turning point and thus the prediction model will likely exceed the reality of data.

# Data Extraction and Preparation

## Environment construction

For research to be duplicatable, efforts must be made to ensure the tools and settings utilized fit with community standards. To align with these efforts, the analysis began in Jupyter Notebook, via the text editing capabilities of Visual Studio Code. Packages and libraries were imported into the environment as follows:

1. pandas as pd, used for DataFrame components and data manipulation
2. numpy as np, utilized for array work and base Python interactions
3. matplotlib.pyplot as plt, and seaborn as sns used for data visualizations
4. sklearn used for complex mathematical components, as well as testing model accuracy
5. statsmodels.api as sm, used as the primary interface for the ARIMA modeling and data preparation
6. datetime, use for data preparation

### Justification of Tools and Techniques**.**

Python will be used for the duration of the project, as it allows for flexibility through packages and is well documented and familiar to the researchers. According to (Manokhin, 2022), “when it comes to forecasting and time series in 2022, Python is the no-brainer choice.” While SAS may also be effective with time series analysis, “the cost of it proves to be a burden” (Mittal, 2022), rendering it not nearly as supported by community. In addition, (Alam, 2021) states that ARIMA—is “arguably the most popular and widely used statistical technique for forecasting.” These sentiments justify a solid foundation in both for time series analysis that is accurate, reproducible, and scalable.

One disadvantage of utilizing Python for this analysis is runtime. The “flexible data-types in Python contribute toward its high memory consumption,” (The Closure Library Authors, 2019) therefore choosing each package required time, planning, and weighted consideration.

## Data Management

### **Cleaning**

After importing the dataset, exploratory functions were launched in order to investigate the data further. Columns unnecessary for the evaluation were dropped (Figure 3).

A cursory evaluation of the data conducted before the implementation of code incorrectly assessed that there was no missing data in the dataset. As such, it was necessary to correct this lack of information before the ARIMA model could be created.

The .fillna() method was used for a single manual input from a missing value in the ‘ITEM TYPE’ column (Figure 4).

One of the key characteristics of a time series analysis is the use of dates as index markers for growth and attrition. Later in the evaluation, the researchers realized there were date values missing. As such, in future passes of this evaluation, the data was prepared by creating a list of the missing values and using .append() to add them to ‘MONTH’ and ‘YEAR’ columns manually.

This .fillna() method was used again to fill values in the numerical columns ‘RETAIL SALES’ and ‘WAREHOUSE SALES’, with the mean column values (Figure 5).

After renaming all columns to no longer be in caps case, the researchers generated a ‘total\_sales’ column from the ‘retail\_sales’ and ‘warehouse\_sales’ columns. This was necessary to create a single target variable for the later analysis (Figure 6).

In order to continue the conversion of the ‘month’ and ‘year’ variables to the necessary datetime object for ARIMA, a ‘day’ column was created with a default value of 1. After which, the to\_datetime() function was utilized to create a single, date column for the time series analysis. The DataFrame was then sorted by the new ‘date’ column (Figure 7).

The resulting file was exported for later use, as it would be heavily manipulated for exploratory visualization.

Columns deemed unnecessary were dropped. Visualizations for the product types available in the dataset were created to compare the contribution of each product. (Figure 8) Visualization notes that there was a considerable cost for certain products, and so a second visualization (Figure 9) was created without those items. Beer had a disproportionately high contribution to earnings. Due to this observation, researchers decided that there was no additional analysis needed to identify which “type” of sale would be evaluated by the model. Removing Beer in any iteration would result in too few data points.

The exported data was uploaded again and assessed for continuity. Different columns were dropped, leaving only the sales data and the dates. The date data was set at the index for the DataFrame and entries that shared a date were combined with .sum().round(2) and organized in ascending order.

The research noted that once all the duplicates were removed, there were only 43 entries left in the DataFrame, found in (Figure 10). This meant that the model would be more sensitive to each fluctuation and would result in large variances.

#### Advantages of Cleaning Method

The aforementioned methods of cleaning and imputing missing values are easy, quick, and visually versatile. The .fillna() function, combined with the .sum() and .round() functions are readily available with base statistical Python packages, meaning they are also well documented and supported in the Python community.

#### Disadvantages of Cleaning Method

In order to manually impute the data as accomplished in this task, the researcher spent a considerable amount of time identifying missing values. Using an automated method, such as the resample() function would have eliminated the need, but would also have required the researcher to make further adjustments to this particular dataset.

### **Stationarity Analysis**

A visualization of the initial dataset—properly cleaned—was created (Figure 11). As the data was not visually stationary, an Augmented Dickey-Fuller unit root test (.adfuller()) was required. The results, found in Table 1, confirmed that the null hypothesis was not rejectable based on the current data.

To transform the dataset to stationary researchers used differencing to subtract each point from the previous value in the primary column. Differencing is a popular and widely used data transform for making time series data stationary. (Brownlee, 2020)

After completing this step, the data was visualized again in order to conduct an initial measurement of success. A graphic was produced (Figure 12) as the result of that visualization. After the differencing was complete, a second Augmented Dickey-Fuller unit root test (.adfuller()) was required. The results of that test can be found in Table 2. Researchers observed that the absolute value of the ADF Statistic value was greater than the critical value. In addition, the p-Value was below 0.05, confirming that the data was now stationary and the null hypothesis was rejectable. (Chaudhary, 2021)

### **Identifying AR and MA parameters**

In order to run the autocorrelation analysis (acf()) and the partial autocorrelation analysis (pacf()), researchers used the train\_test\_split function on the differenced data to split the data into two groups. The researchers opted to use the recommended 4:1 ratio for said split.

The acf() function was applied to the train dataset in order to correlate the points of the dataset with a lagged version of itself. The autocorrelation function begins at lag 0 and results in a correlation of 1. (Figure 13) Visualization of the resulting graph was produced.

The pacf() function was then applied to the train dataset in order to correlate the points of the dataset by each successive lagged term. The correlation between observed points is considered as correlated to observations at other time junctions. (Figure 14) Visualization of the resulting graph produced.

Finally, the data was exported for posterity.

#### Advantages of Stationary Method

Using the Augmented Dickey Fuller test is a common method for measuring stationary within a given time series. It has been thoroughly vetted and supported in the Python community. Differencing is also one of the most widely used methods for adjusting stationary in a dataset.

#### Disadvantages of Stationary Method

While the ADF test is capable of finding pertinent values, even with zero values within the dataset, the function requires an appropriately sized data set in order to be used effectively. Due to this fact, the researcher was not able to combine like-date values before this step. Differencing, while useful as a linear series representation, is not well supported with truly random data. With evaluations such as this—data heavily impacted by global impact—differencing suffers to identify whether or not “the persistence model (using the observation at the previous time step as what will happen in the next time step) provides the best source of reliable predictions” (Flovik, 2021).

# Model Identification and Analysis

## Pre-modeling Assessment

### **Initial Visualizations**

Two visualizations were created to measure the potential performance of the model before the model was created. The seasonal decomposition is a useful and quick tool created to look at the trend, seasonality, and noise present in the dataset. This visualization was produced (Figure 15). The researchers paid close attention to the Seasonality and Residuals lines especially. The resulting flat visualization was what was expected.

The second visualization created to review the state of the data was the spectral density analysis. This visual, (Figure 16) is a “frequency domain representation of a time series that is directly related to the autocovariance time domain representation.” (*12.1 Estimating the Spectral Density | STAT 510*, n.d.). This analysis helps to determine the frequency of the sales variables. This graph further exhibits no evidence of seasonal trends.

### **Model Creation**

The auto\_arima function was used to test different arguments for the potential ARIMA model quickly. This function was selected as it is easily verifiable and allows the researcher to compare a multitude of variables in a short amount of time. This was done by performing a stepwise search for the minimum Akaike's Information Criterion (AIC) possible. The returned best model values were ARIMA(2,0,0)(0,0,0)[0] intercept.

A second, independent ARIMA model was created with the given order parameters in order to confirm the assertion from auto\_arima. When run, the SARIMAX Results were within acceptable ranges, with p\_Values that were not zero, but largely less than .05. Model and results recorded (Figure 17).

With the model created, the data was split again, and the updated test and train models were filtered into the .predict() function in order to produce prediction information on the train and test values. This is documented (Figure 18). A visualization of the resulting prediction was created (Figure 19).

#### Advantages of ARIMA model

The ARIMA model is easy to read and only requires previous data and time series information to create. The method works well for short-term forecasting and can be updated regularly with relative ease. Once the modeling environment is created, it can be duplicated and rerun without a complete reevaluation of all necessary metrics—making it a handy forecast tool for junior analyst projects.

#### Disadvantages of ARIMA model

As the researchers note in their visualization, the ARIMA model alone cannot be used to adjust for market unpredictability. The model makes it “difficult to predict turning points,” (*Understanding ARIMA Models for Machine Learning*, n.d.-b) in data and market fluctuations. As such, models must be rerun regularly and used largely for comparison, rather than as a goal definition.

### **Research Question Prediction**

The research question was whether or not a time series model could be used to predict the next quarter’s sales data for each item type in the dataset. While there was not enough support to itemize the prediction, a time series model was created for overall sales. (Figure 20) Insights as to the code used in the creation of that prediction are available, as well as a visualization of the properties of the prediction (Figure 21).

# Data Summary and Implications

Though the overall trend in the data is steadily increasing, there is a considerable downturn approaching, according to the model (Pierre, 2021)**.** The implication of this downward trend is that the county will suffer from sales loss at the beginning of the year. A sound recommendation is that the county spend time leading up to this point, putting in place processes and spending limits to lessen the impact of the revenue loss.

As previously mentioned, a glaring limitation of this model is the timing at which the data was gathered. As ARIMA models struggle with adjusting for turning points in market need, there will likely continue to be disparities between the predicted model and the current sales model.

It is the researcher’s recommendation that as data continues to be gathered, the models be split into pre- and post-downturn evaluations and compared thusly. In doing so, the information will not only reflect overall growth (or loss) within the market, but also growth compared to the current economic truth that has impacted the county’s sales numbers. By evaluating the sales data as two separate entities—when there is enough data to support this shift—the county will have more accurate forecasts for decisionmakers to plan with.

Furthermore, in order to better analysis data in the future, it would be beneficial for the county to record data on sales per day, rather than per month. It would also be beneficial for the county to record gains and losses for the missing dates so that the standard deviation information could be replaced with reality. It is the researcher’s recommendation that this evaluation be run on a quarterly basis to continue to reap the benefits of forecasting.

**References**

12.1 Estimating the Spectral Density | STAT 510. (n.d.). PennState: Statistics Online Courses. https://online.stat.psu.edu/stat510/lesson/12/12.1

Alam, M. (2021, December 15). Comparing the performance of forecasting models: Holt-Winters vs ARIMA. Medium. https://towardsdatascience.com/comparing-the-performance-of-forecasting-models-holt-winters-vs-arima-e226af99205f

Brownlee, J. (2020, June 23). How to Remove Trends and Seasonality with a Difference Transform in Python. Machine Learning Mastery. https://machinelearningmastery.com/remove-trends-seasonality-difference-transform-python/

Chaudhary, M. (2021, December 14). Why is Augmented Dickey–Fuller test (ADF Test) so important in Time Series Analysis. Medium. https://medium.com/@cmukesh8688/why-is-augmented-dickey-fuller-test-adf-test-so-important-in-time-series-analysis-6fc97c6be2f0

Flovik, V., PhD. (2021, April 28). How (not) to use Machine Learning for time series forecasting: Avoiding the pitfalls. https://www.linkedin.com/pulse/how-use-machine-learning-time-series-forecasting-vegard-flovik-phd/

Hayes, A. (2022, June 13). What Is a Time Series and How Is It Used to Analyze Data? Investopedia. https://www.investopedia.com/terms/t/timeseries.asp

Kumar, S. (2022, April 30). 4 Techniques to Handle Missing values in Time Series Data. Medium. https://towardsdatascience.com/4-techniques-to-handle-missing-values-in-time-series-data-c3568589b5a8

Manokhin, V., PhD. (2022, December 16). Python vs R for time-series forecasting, MBA, CQF. Medium. https://valeman.medium.com/python-vs-r-for-time-series-forecasting-395390432598

Mesidor, F. (2021, December 11). A basic guide to time series analysis - Towards Data Science. Medium. https://towardsdatascience.com/a-basic-guide-into-time-series-analysis-2ad1979c7438

Mittal, S. (2022, June 26). The Best Machine Learning Tool: Python vs R vs SAS. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. https://www.analytixlabs.co.in/blog/the-best-machine-learning-tools-python-vs-r-vs-sas/

Pierre, S. (2021, July 16). A Guide to Time Series Analysis in Python | Built In. A Guide to Time Series Analysis in Python | Built In. https://builtin.com/data-science/time-series-python

Prabhakaran, S. (2022, September 3). ARIMA Model – Complete Guide to Time Series Forecasting in Python. Machine Learning Plus. https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

Sai Ram, K. (2020, February 28). Time Series Analysis and Forecasting: Novel Business Perspectives. AltexSoft. https://www.altexsoft.com/blog/business/time-series-analysis-and-forecasting-novel-business-perspectives/

The Closure Library Authors. (2019, July 24). R Vs Python – The most trending debate of aspiring Data Scientists. https://data-flair.training/blogs/r-vs-python/

Tiao, G. (2015, April 10). Time Series: ARIMA Methods. International Encyclopedia of the Social & Behavioral Sciences (Second Edition). https://www.sciencedirect.com/topics/social-sciences/time-series

Tunggawan, E. (2018, October 27). Time series analysis: validating effect of changes. https://elvyna.github.io/2018/time-series-hypothesis-testing/

Understanding ARIMA Models for Machine Learning. (n.d.). Capital One. https://www.capitalone.com/tech/machine-learning/understanding-arima-models/

*Warehouse and Retail Sales - Catalog*. (n.d.). https://catalog.data.gov/dataset/warehouse and-retail-sales

Zaugg, C. (2018, April 23). Data Redaction: What it is, why you need it, and how to start. Reveal. https://opin.com/intro-to-redaction/

Zvornicanin, E. (2022, November 8). Choosing the best q and p from ACF and PACF plots in ARMA-type modeling. Baeldung. https://www.baeldung.com/cs/acf-pacf-plots-arma-modeling

**Tables**

Table 1

Initial Augmented Dickey-Fuller unit root test

| df\_test = ts.adfuller(df[‘total\_sales’] | |
| --- | --- |
| ADF Statistic | -1.706677 |
| p-Value | 0.427676 |
| Lags | 1.000000 |
| Observations | 41.000000 |
| Critical Values (1%) | -3.600983 |
| Critical Values (5%) | -2.935135 |
| Critical Values (10%) | -2.605963 |

Note: A statistical significance test that aids in the testing of the null and alternative hypothesis. Results in the creation of test statistics, as well as p-Values which can be utilized to inference as to whether a given dataset is stationary or not.

Table 2

Augmented Dickey-Fuller unit root test post-differencing

| df\_test = ts.adfuller(df[‘total\_sales’] | |
| --- | --- |
| ADF Statistic | -4.594161 |
| p-Value | 0.000132 |
| Lags | 2.000000 |
| Observations | 39.000000 |
| Critical Values (1%) | -3.610400 |
| Critical Values (5%) | -2.939109 |
| Critical Values (10%) | -2.608063 |

Note: A statistical significance test that aids in the testing of the null and alternative hypothesis. Results in the creation of test statistics, as well as p-Values which can be utilized to inference as to whether a given dataset is stationary or not.

Table 3

Augmented Dickey-Fuller unit root test post-differencing

| stepwise\_fit = auto\_arima(df['total\_sales'], trace = True, suppress\_warnings = True) |
| --- |
| Performing stepwise search to minimize aic |
| ARIMA(2,0,2)(0,0,0)[0] intercept : AIC=1091.016, Time=0.05 sec |
| ARIMA(0,0,0)(0,0,0)[0] intercept : AIC=1111.779, Time=0.01 sec |
| ARIMA(1,0,0)(0,0,0)[0] intercept : AIC=1094.260, Time=0.02 sec |
| ARIMA(0,0,1)(0,0,0)[0] intercept : AIC=1105.847, Time=0.02 sec |
| ARIMA(0,0,0)(0,0,0)[0] : AIC=1231.374, Time=0.01 sec |
| ARIMA(1,0,2)(0,0,0)[0] intercept : AIC=1091.692, Time=0.03 sec |
| ARIMA(2,0,1)(0,0,0)[0] intercept : AIC=1088.974, Time=0.04 sec |
| ARIMA(1,0,1)(0,0,0)[0] intercept : AIC=1093.148, Time=0.03 sec |
| ARIMA(2,0,0)(0,0,0)[0] intercept : AIC=1088.892, Time=0.03 sec |
| ARIMA(3,0,0)(0,0,0)[0] intercept : AIC=1088.994, Time=0.04 sec |
| ARIMA(3,0,1)(0,0,0)[0] intercept : AIC=1090.845, Time=0.05 sec |
| ARIMA(2,0,0)(0,0,0)[0] : AIC=inf, Time=0.02 sec |
| Best model: ARIMA(2,0,0)(0,0,0)[0] intercept |
| Total fit time: 0.360 seconds |

Note: Used to test multiple parameters for an ARIMA model, in order to find the lowest Akaike’s Information Criterion (AIC) possible. The resulting best model returns the ideal order parameters for ARIMA, as well as a second set of information that measures seasonality and trends.

**Figures**

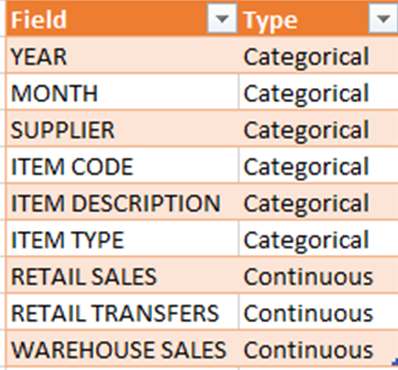


Figure 1. organized list of the available columns and the corresponding types for the original dataset.

Graphical user interface, text, application

Description automatically generated

Figure 2. organized list of the available columns and the corresponding types for the updated dataset.

Graphical user interface, text

Description automatically generated

Figure 3. Dropped unneeded columns.

Text

Description automatically generated

Figure 4. Identify records missing ITEM\_TYPE and use a world wide web search to identify and fill the missing value.

Text

Description automatically generated

Figure 5. Append missing data values in the dataset manually. Fill all missing components with the .fillna() method.

Text

Description automatically generated

Figure 6. Rename columns. Create ‘total\_sales’ column by combining ‘retail\_sales’ and ‘warehouse\_sales.’

Text

Description automatically generated

Figure 7. Rename columns. Create ‘total\_sales’ column by combining ‘retail\_sales’ and ‘warehouse\_sales.’

Chart, waterfall chart

Description automatically generated

Figure 8. Plotted in seaborn as a visualization of all products. Visualization is disturbed by the large number of negative value items in the dataset.

Chart, bar chart

Description automatically generated Figure 9. Adjusted product visualization removes negative-performing products to evaluate those that contribute to total sales.

Text

Description automatically generated

Figure 10. Printed info() before and after grouping to show that there were 307,663 records before combination and only 43 after.

Chart, histogram

Description automatically generatedFigure 11. Visualization of DataFrame after all variables have been cleaned and prepared.

Chart, line chart, histogram

Description automatically generatedFigure 12. Visualization of DataFrame after differencing has been implemented.

Chart, line chart

Description automatically generatedFigure 13. Visualization of ACF function on training data.

Chart, line chart

Description automatically generatedFigure 14. Visualization of ACF function on training data, black line added to mark 0.05 as p-Value.

A picture containing line chart

Description automatically generated

Figure 15. Visualization of the seasonal\_decompose() function. The resulting object contains original, trend, seasonality, and residuals data for researchers to evaluate.

Chart, line chart

Description automatically generated

Figure 16. Power spectral density visual, created to mark power, strength, and frequency of data.

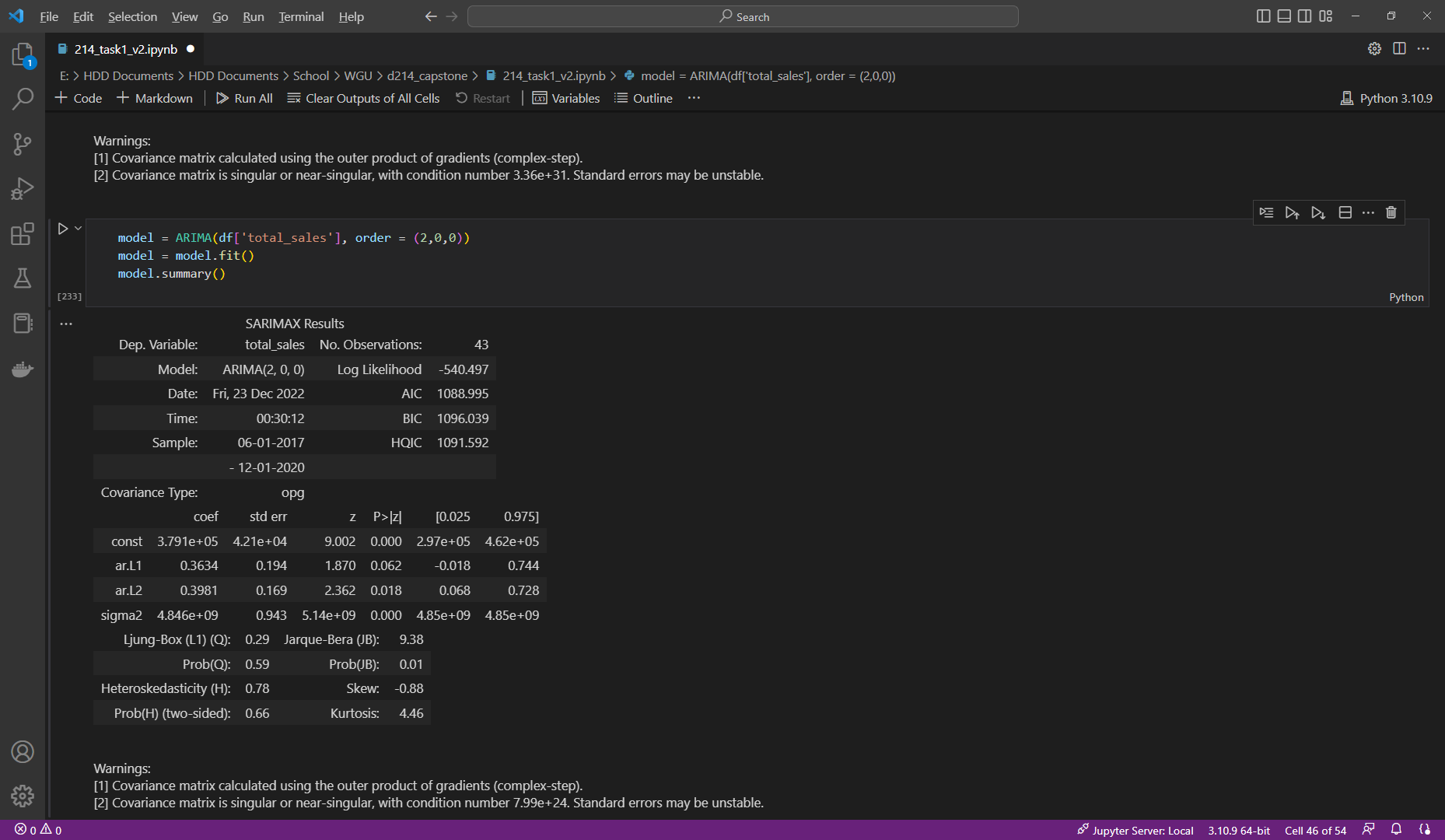


Figure 17. SARIMAX Results from independent model creation using defined parameters from auto\_arima.

A screenshot of a computer

Description automatically generated with medium confidence

*Figure 18.* Test and train variables updated for predictions.

Chart, line chart

Description automatically generated

*Figure 19.* Visualization of model prediction, compared to test data for total sales.

Graphical user interface, text

Description automatically generated

*Figure 20.* Code used to generate the prediction model, fit to extend for 3 months after the dataset concludes.

Chart, line chart

Description automatically generated

*Figure 21.* Model fitted to training data—organized by month.