Performance Assessment

WGU | D214

Task 2

Narcisse, Laurie

2024

Contents

[**Research Question** 2](#_Toc159419583)

[A. Summarize the original real-data research question you identified in task 1. Your summary should include justification for the research question you identified in task 1, a description of the context in which the research question exists, and a discussion of your hypothesis. 2](#_Toc159419584)

[Justification: 2](#_Toc159419585)

[Description of Context: 2](#_Toc159419586)

[**Data Collection** 3](#_Toc159419587)

[B.  Report on your data-collection process by describing the relevant data you collected, discussing one advantage and one disadvantage of the data-gathering methodology you used, and discussing how you overcame any challenges you encountered during the process of collecting your data. 3](#_Toc159419588)

[**Data Extraction and Preparation** 4](#_Toc159419589)

[C.  Describe your data-extraction and -preparation process and provide screenshots to illustrate each step. Explain the tools and techniques you used for data extraction and data preparation, including how these tools and techniques were used on the data. Justify why you used these particular tools and techniques, including one advantage and one disadvantage when they are used with your data-extraction and -preparation methods. 4](#_Toc159419590)

[Importing Python Packages: 5](#_Toc159419591)

[Reading the Data: 8](#_Toc159419592)

[Clean Data: 9](#_Toc159419593)

[About the Data: 10](#_Toc159419594)

[**Analysis** 20](#_Toc159419595)

[D.  Report on your data-analysis process by describing the analysis technique(s) you used to appropriately analyze the data. Include the calculations you performed and their outputs. Justify how you selected the analysis technique(s) you used, including one advantage and one disadvantage of these technique(s). 20](#_Toc159419596)

[**Data Summary and Implications** 40](#_Toc159419597)

[E.  Summarize the implications of your data analysis by discussing the results of your data analysis in the context of the research question, including one limitation of your analysis. Within the context of your research question, recommend a course of action based on your results. Then propose two directions or approaches for future study of the data set. 40](#_Toc159419598)

[- Forecasting Accuracy: 40](#_Toc159419599)

[- Limitation: 40](#_Toc159419600)

[- Recommendation: 40](#_Toc159419601)

[- Future Directions 40](#_Toc159419602)

[F.  Acknowledge sources, using in-text citations and references, for content that is quoted. 41](#_Toc159419603)

# **Research Question**

## Summarize the original real-data research question you identified in task 1. Your summary should include justification for the research question you identified in task 1, a description of the context in which the research question exists, and a discussion of your hypothesis.

The original real-data research question identified in task 1 is as follows:

How can we effectively forecast viewer engagement trends for movies and TV shows on Netflix using time series analysis?

### Justification:

This research question is justified by the need for stakeholders in the entertainment industry, such as content creators, producers, and platforms like Netflix, to optimize content selection, scheduling, and promotion strategies based on audience behavior and preferences. By leveraging historical viewer engagement data through time series analysis, stakeholders can make data-driven decisions to enhance viewer engagement and platform performance.

### Description of Context:

The research question aims to explore methods for forecasting viewer engagement trends specifically for movies and TV shows available on the streaming platform Netflix. The context of this inquiry lies within the realm of entertainment industry analytics and data-driven decision-making processes. As streaming services like Netflix continue to shape the way audiences consume media, understanding viewer engagement patterns becomes increasingly vital for content creators, distributors, and platform operators.

By leveraging time series analysis techniques, the research seeks to uncover patterns, trends, and potential predictors of viewer engagement over time. This analysis can provide valuable insights into the factors influencing audience behaviors, such as release dates, genre preferences, seasonal trends, and promotional activities. Understanding these dynamics can empower content creators and platform operators to optimize content strategies, scheduling, and marketing efforts to enhance viewer engagement and retention.

The hypothesis for this research question is formulated as follows:

Null hypothesis: There is no significant relationship between historical viewer engagement data and future engagement trends for movies and TV shows on Netflix.

Alternate hypothesis: Historical viewer engagement data can be used to accurately forecast future engagement trends for movies and TV shows on Netflix.

The proposed hypotheses frame the investigation into the relationship between historical viewer engagement data and future engagement trends for movies and TV shows on Netflix.

The **null hypothesis** posits that there is no significant relationship between historical viewer engagement data and future engagement trends. In other words, it suggests that past viewer engagement does not have any predictive power in forecasting future trends. Under this hypothesis, any observed correlations or patterns between historical data and future trends would be attributed to random chance rather than a genuine relationship.

Conversely, the **alternative hypothesis** proposes that historical viewer engagement data can indeed be used to accurately forecast future engagement trends. This hypothesis implies that there is a meaningful and statistically significant relationship between past engagement levels and future trends. If supported by the data, this hypothesis suggests that patterns and trends observed in historical data can serve as reliable indicators for predicting future viewer behavior.

The discussion surrounding these hypotheses involves evaluating the evidence gathered from the analysis of historical viewer engagement data. Statistical techniques such as time series analysis may be employed to examine the relationship between past and future engagement trends. The significance level, typically denoted as alpha (α), will be set to determine whether the evidence supports rejecting the null hypothesis in favor of the alternative hypothesis.

The hypotheses provide a structured framework for investigating the predictive capabilities of historical viewer engagement data and inform the direction of the research inquiry. The results of the analysis will contribute to our understanding of the factors influencing viewer engagement on Netflix and may have implications for content creation, scheduling, and marketing strategies.

# **Data Collection**

## B.  Report on your data-collection process by describing the relevant data you collected, discussing one advantage and one disadvantage of the data-gathering methodology you used, and discussing how you overcame any challenges you encountered during the process of collecting your data.

The data for this analysis was collected from the Kaggle Netflix Movies and TV Shows Dataset.

The full dataset can be found at [Netflix Movies & Shows Dataset (kaggle.com)](https://www.kaggle.com/datasets/ashfakyeafi/netflix-movies-and-shows-dataset/data).

This data is a collection of entries from a streaming service representing movies and TV shows available for viewing. The data contains 11 columns. Here is a breakdown:

* show\_id: A unique identifier for each entry.
* type: Specifies whether the entry is a "Movie" or a "TV Show".
* title: The title of the movie or TV show.
* director: For movies, the director's name; for TV shows, usually left blank as TV shows typically have multiple directors.
* cast: The cast members starring in the movie or TV show.
* country: The country of origin for the movie or TV show.
* date\_added: The date when the movie or TV show was added to the streaming service.
* release\_year: The year when the movie or TV show was originally released.
* rating: The content rating assigned to the movie or TV show (e.g., PG-13, TV-MA).
* duration: The duration of the movie or the number of seasons for TV shows.
* listed\_in: Categories or genres to which the movie or TV show belongs.
* description: A brief summary or description of the movie or TV show.

One advantage of using this dataset is its comprehensiveness, providing detailed information on movies and TV shows available on Netflix, including viewer engagement metrics and relevant time series data. However, a disadvantage is that the dataset may have limitations in terms of coverage or accuracy, as it relies on user-contributed data. To overcome potential challenges in data collection, thorough data validation and cleaning processes were implemented to ensure data quality and reliability.

# **Data Extraction and Preparation**

## C.  Describe your data-extraction and -preparation process and provide screenshots to illustrate each step. Explain the tools and techniques you used for data extraction and data preparation, including how these tools and techniques were used on the data. Justify why you used these particular tools and techniques, including one advantage and one disadvantage when they are used with your data-extraction and -preparation methods.

The data extraction, cleaning, and preparation process involved employing standard techniques utilizing popular Python libraries including Pandas, Numpy, and Matplotlib. The primary environment utilized was Jupyter Notebooks in conjunction with Python.

The subsequent sections outline the steps necessary to set up the Jupyter Notebook environment, import the requisite libraries, and cleanse the original data. Some steps are accompanied by explanations, justifications, advantages, and disadvantages for review.

### Importing Python Packages:

A screenshot of a computer program

Description automatically generated

This code imports various libraries and modules commonly used for time series analysis in Python. Let's break down the purpose, advantages, and disadvantages of each part of the code:

1. **numpy (np):** Numerical computing library in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

2. **pandas (pd):** Data manipulation and analysis library. It provides easy-to-use data structures and data analysis tools, particularly for working with structured or tabular data.

3. **matplotlib.pyplot (plt):** A plotting library that provides a MATLAB-like interface for creating static, interactive, and animated visualizations in Python.

4. **seaborn (sns):** A data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

5. **statsmodels.api (sm):** A Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.

6. **plotly.graph\_objects (go):** Plotly's low-level graphing interface. It allows for the creation of complex and customizable visualizations using a simple Python API.

7. **plotly.express (px):** Plotly's high-level interface for creating easy and expressive visualizations. It is built on top of Plotly's graph objects and provides a more concise syntax.

8. **statsmodels.graphics.tsaplots**: A module in statsmodels that provides functions for plotting time series analysis-related plots such as autocorrelation function (ACF) and partial autocorrelation function (PACF) plots.

9**. statsmodels.tsa.statespace.sarimax (SARIMAX):** A class in statsmodels for fitting Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors (SARIMAX) models.

10. **statsmodels.tsa.holtwinters (ExponentialSmoothing):** A class in statsmodels for fitting Holt-Winters exponential smoothing models.

11**. statsmodels.tsa.stattools:** A module in statsmodels that provides various statistical tools for time series analysis, such as the Augmented Dickey-Fuller (ADF) test for stationarity.

12. **statsmodels.tsa.arima\_model (ARIMA):** A class in statsmodels for fitting AutoRegressive Integrated Moving Average (ARIMA) models.

13. **statsmodels.tsa.seasonal**: A module in statsmodels that provides tools for seasonal decomposition of time series data.

14. **sklearn.metrics**: A module in scikit-learn that provides various metrics for evaluating model performance, such as mean squared error (MSE) and mean absolute error (MAE).

15. **pmdarima.arima (auto\_arima):** A function in pmdarima for automatically selecting the best ARIMA model parameters using a stepwise approach.

16. **sklearn.preprocessing**: A module in scikit-learn for preprocessing data, including scaling, normalization, and encoding categorical variables.

17. **pandas.plotting:** A module in pandas that provides various plotting functions, such as lag plot for visualizing autocorrelation.

18. **pandas.Timestamp:** A pandas data type representing timestamps.

19. **itertools.product:** A function that computes the Cartesian product of input iterables.

20. **tqdm:** A library for adding progress bars to Python loops.

21. **pmdarima:** A Python module for automating time series forecasting using ARIMA models.

22. **warnings:** Python's built-in module for handling warnings.

23. **matplotlib.pylab.rcParams:** A module in matplotlib for setting default parameters for figures.

#### Advantages:

* Comprehensive set of libraries covering various aspects of time series analysis.
* Provides a wide range of tools and functions for data preprocessing, model fitting, evaluation, and visualization.
* Allows for both basic and advanced time series analysis techniques, such as ARIMA modeling, exponential smoothing, and seasonal decomposition.
* Integration with other data science libraries like scikit-learn and Plotly allows for seamless workflow.

#### Disadvantages:

* Including many libraries may lead to increased memory usage and longer startup times.
* Some libraries may have overlapping functionality, which can lead to confusion about which one to use in a particular situation.
* Learning curve may be steep for beginners due to the complexity and variety of techniques covered by the libraries.
* -Reliance on external libraries may introduce dependencies and version compatibility issues.

(Ramuglia, 2024), (GfG, 2021), (Koidan, 2021).

A white rectangular sign with red and blue text

Description automatically generated

Setting the color palette to “icefire”. This code is useful for ensuring consistency in color schemes across all visualizations.

### Reading the Data:

A screenshot of a computer

Description automatically generated

This code reads a CSV file named "netflix\_data.csv" into a pandas DataFrame df and prints summary information about the DataFrame using the info() method.

A screenshot of a computer

Description automatically generated

Calculates the sum of missing values (NaNs) for each column in the DataFrame df.

A screenshot of a computer

Description automatically generated

### Clean Data:

A screen shot of a computer code

Description automatically generated

This code prepares the DataFrame for time series analysis by converting the date column to datetime format and extracting the year and month information for further analysis.

A white rectangular sign with red text

Description automatically generated

This code drops rows with missing values in the 'year\_added' column using the dropna() method with the subset parameter. Then, it converts the 'year\_added' column to integer type using the astype() method.

### About the Data:



A screenshot of a graph

Description automatically generated

A pie chart with text and numbers

Description automatically generated

A screenshot of a computer code

Description automatically generated

The codes above display the frequency of different types of shows in Netflix. It displays that movies account for almost 70% of the data. There are 6,131 movies compared to 2,666 tv shows.

A graph of different colored bars

Description automatically generated

This visualization explains the years with the most movies. As displayed, the year 2019 has the most movies.

A screen shot of a computer screen

Description automatically generated

This visualization allows you to examine any patterns or trends in the time gap between the release of shows/movies and their addition to Netflix.

A graph with a line going up

Description automatically generated

This visualization provides insights into the growth of Netflix content over time, helping to understand the platform's content expansion trend.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

This visualization and calculation provide insights into the proportion of kid-friendly content available on Netflix compared to other content categories. In this case, approximately 7.29% of the content is categorized as kid-friendly.

A screenshot of a computer screen

Description automatically generated

This visualization helps to identify the countries producing the most movies available on Netflix, providing insights into the global distribution of movie production. The United States is the country with the most movies available.

A screenshot of a computer

Description automatically generated

This code presents the 20 oldest movies/shows from the United States in a table format using Plotly, providing a neat and interactive way to visualize the data.

A screen shot of a graph

Description automatically generated

This code provides an interactive visualization of the yearly trend in the number of movies and TV shows released on Netflix since 2007, allowing users to explore the data more dynamically.

A screenshot of a computer code

Description automatically generated

A graph with a line

Description automatically generated

This code provides an interactive visualization of the average monthly release trend for movies and TV shows on Netflix since 2007, allowing users to explore the data more dynamically.

A screen shot of a computer code

Description automatically generated

A blue and white graph

Description automatically generated

This code provides insights into the distribution of TV show durations (seasons) and movie durations on Netflix, allowing for a visual comparison between the two types of content.

A screenshot of a computer

Description automatically generated

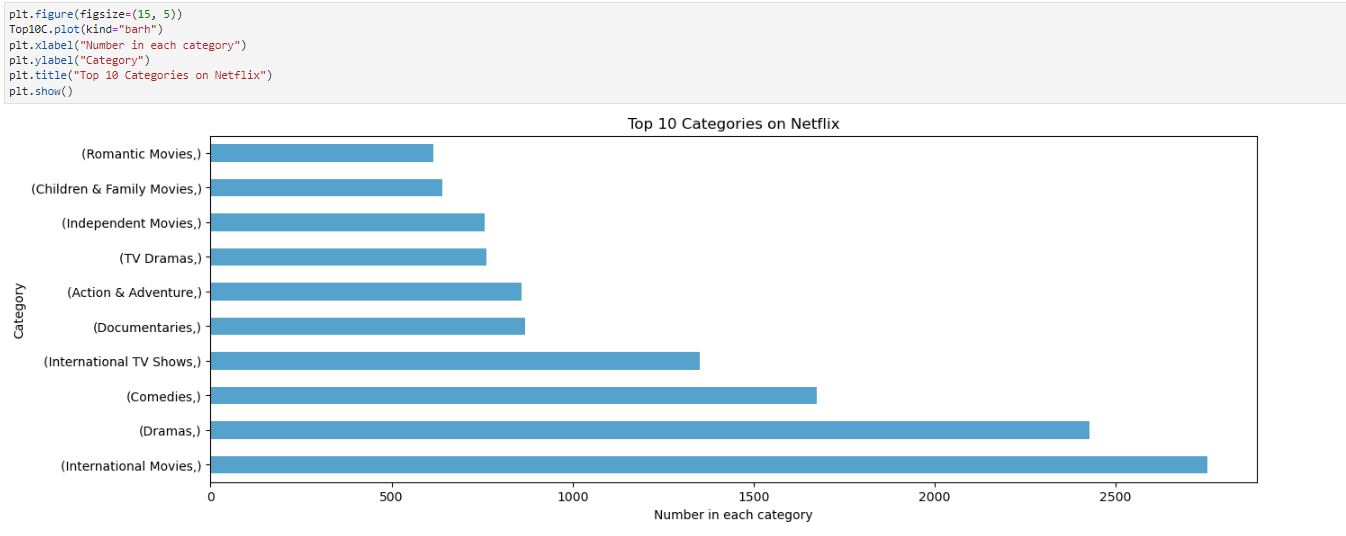
A screenshot of a computer screen

Description automatically generated

This code provides insights into the most prolific directors on Netflix, allowing users to see which directors have contributed the most content to the platform.

A screen shot of a computer code

Description automatically generated



This code extracts categories from the "listed\_in" column of a DataFrame, counting their occurrences. It then selects the top 10 categories and plots them in a horizontal bar chart, with category names on the y-axis and their frequencies on the x-axis. Finally, it adds labels and a title before displaying the plot.

# **Analysis**

## D.  Report on your data-analysis process by describing the analysis technique(s) you used to appropriately analyze the data. Include the calculations you performed and their outputs. Justify how you selected the analysis technique(s) you used, including one advantage and one disadvantage of these technique(s).

The three techniques overviewed above all play an essential role in ensuring the performed data analysis is as reliable as possible.

**Exploratory Data Analysis** (EDA) allows for a comprehensive review of the existing data, fostering better familiarity with its contents. It aids in identifying errors, understanding patterns, detecting outliers, and uncovering relationships among variables.

**Statistical testing** provides a framework for making quantitative decisions about processes, typically involving the evaluation of hypotheses. It helps determine whether there is sufficient evidence to reject a null hypothesis.

The creation of **time series models** aims to generate accurate predictions, such as stock closing prices for included companies. It offers insights into underlying trends, systematic patterns, and seasonal variations, enhancing understanding of their causes.

**Advantages:**

**Exploratory Data Analysis:**

* Identifies errors and patterns within the data.
* Helps detect outliers and find relationships among variables.

**Statistical Testing:**

* Provides a quantitative basis for decision-making.

**Time Series Modeling:**

* Enhances understanding of underlying trends and seasonal patterns.
* Offers insights into the causes behind these trends.

**Disadvantages:**

**Exploratory Data Analysis:**

* May yield inconclusive results.
* Lack of standardized analysis methods.
* Limited by sample size and potential for outdated information.

**Statistical Testing:**

* Prone to misuse if tests are not carefully chosen or interpreted.

**Time Series Modeling:**

* Generalization from a single study may be challenging.
* Difficulty in obtaining appropriate measures.
* Identifying the correct model to represent the data can be problematic.

**Figures and code below:**

A screenshot of a computer

Description automatically generated

This code first converts the "date\_added" column in a DataFrame to datetime format and then groups the data by month, counting the number of shows added each month. It then renames the column containing the counts to "value" for clarity. Afterward, it checks for any missing values, which it doesn't find. Finally, it displays information about the DataFrame, confirming it has a datetime index with 165 entries and one integer column named "value" without any null values.

A screenshot of a data

Description automatically generated

This code filters the DataFrame shows\_added to include only rows with a "date\_added" after January 1, 2014. It creates a boolean mask to identify these rows based on the index, applies the mask to the DataFrame using .loc[], and displays the resulting DataFrame, which now contains data from January 2014 onwards with 93 entries.

A screenshot of a computer program

Description automatically generated

A graph showing a number of blue lines

Description automatically generated with medium confidence

A graph with blue dots

Description automatically generated

A screenshot of a computer

Description automatically generated

This code conducts an analysis of autocorrelation and partial autocorrelation for a given time series dataset. It generates plots showing the autocorrelation and partial autocorrelation functions with up to 30 lags. Additionally, it calculates the autocorrelation and partial autocorrelation values without plotting them, providing a numerical representation of the temporal dependencies within the dataset. These values help in understanding how each observation in the time series relates to its past observations.

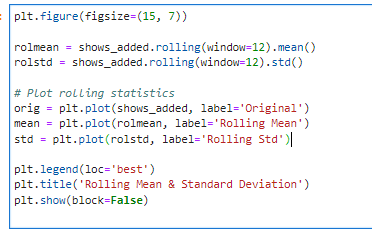
The autocorrelation values represent the correlation between each observation and its lagged values up to 30 lags. The first value is always 1 because it represents the correlation of the series with itself (lag 0). The subsequent values decrease gradually, indicating a decreasing correlation as the lag increases. On the other hand, the partial autocorrelation values represent the correlation between each observation and its lagged values after removing the influence of intervening observations. Peaks or troughs in the partial autocorrelation values suggest significant temporal dependencies at specific lags.



A graph of blue lines and dots

Description automatically generated with medium confidence

The code performs a seasonal decomposition of the shows\_added time series data using the additive model. This decomposition separates the time series into three components: trend, seasonal, and residual (or noise). The plot() function then visualizes these components along with the original time series.



A graph with blue lines

Description automatically generated

This code generates a plot showing the original time series (`shows\_added`), along with its rolling mean and rolling standard deviation. The rolling mean and rolling standard deviation are calculated using a window size of 12. The plot helps visualize any trend or seasonality present in the data and how it changes over time.

A screenshot of a computer program

Description automatically generated

A line graph with numbers and a graph

Description automatically generated

The code performs an Augmented Dickey-Fuller (ADF) test to evaluate the stationarity of the time series data stored in the DataFrame df\_diff. It prints the ADF test statistic, the corresponding p-value, and critical values for different confidence levels. Since the p-value (0.82820303) exceeds the significance level of 0.05, we fail to reject the null hypothesis, indicating that the time series is non-stationary. Consequently, further analysis or transformations may be required to make the data stationary for certain time series modeling techniques. Additionally, a plot of the time series is presented for visual examination.

A screenshot of a computer code

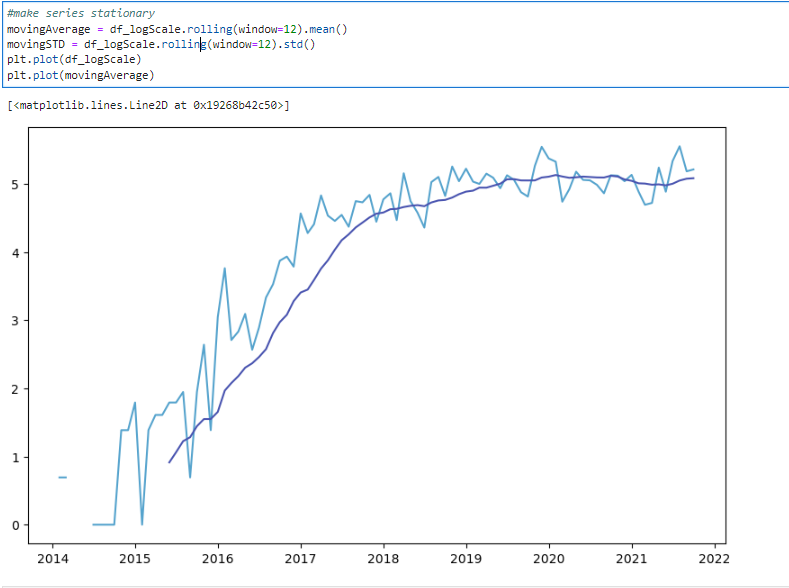
Description automatically generated

This code runs the Augmented Dickey Fuller test again to also see the Number of Observations and lags used.

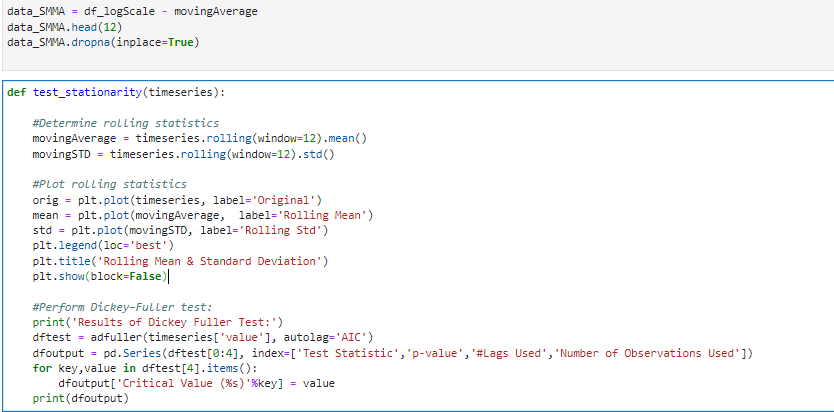
A graph with lines and numbers

Description automatically generated

The code calculates the natural logarithm of the values in the DataFrame `shows\_added` and assigns the result to `df\_logScale`. Then, it plots the resulting logarithmic values. The plot provides a visual representation of the transformed data, allowing for better understanding of its behavior over time.



The code computes the moving average and standard deviation of the log-transformed series df\_logScale using a window size of 12 time steps. Then, it plots both the original log-transformed series and the moving average on the same plot. This visualization helps in identifying trends and variations in the data over time, with the moving average serving as a smoother representation of the underlying trend.



A graph with blue lines and a black line

Description automatically generated

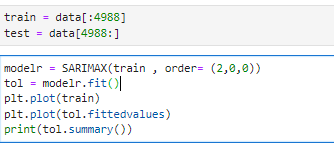
A screenshot of a test

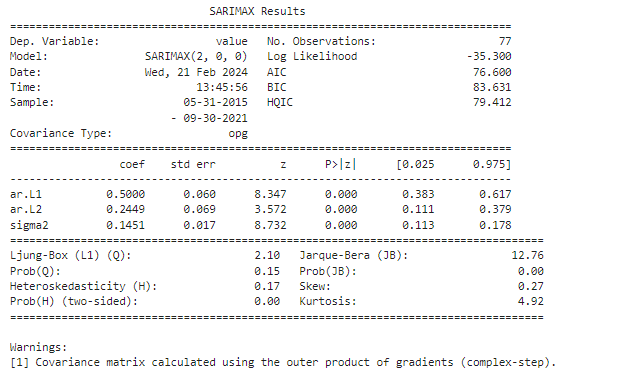
Description automatically generated

The code defines a function `test\_stationarity` to test the stationarity of a time series by plotting its rolling mean and standard deviation and conducting the Dickey-Fuller test. After conducting the test, the results show that the test statistic is -2.303355 with a p-value of 0.170888. With 2 lags used and 74 observations, the critical values at the 1%, 5%, and 10% levels are -3.521980, -2.901470, and -2.588072 respectively. Since the p-value is greater than 0.05, the null hypothesis cannot be rejected, indicating that the time series is likely non-stationary.

A screenshot of a computer

Description automatically generated

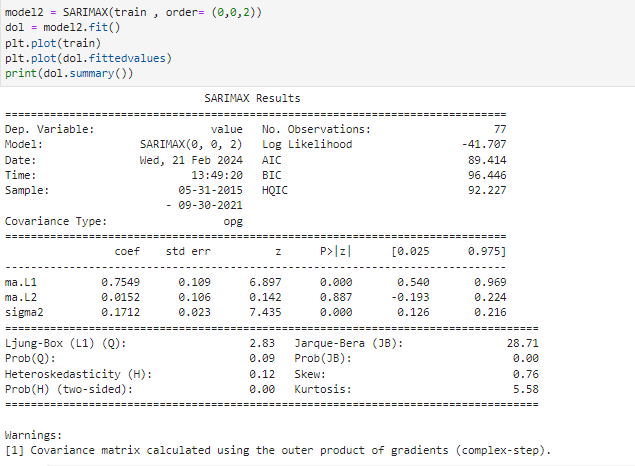




A graph with blue lines

Description automatically generated

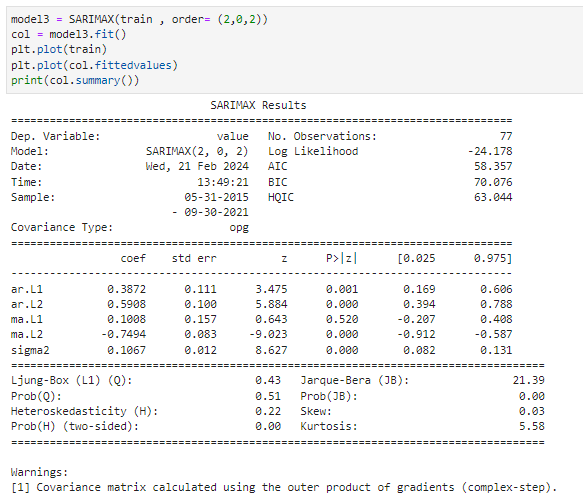
The updated code performs SARIMAX modeling on the time series data. It fits the SARIMAX(2, 0, 0) model to the training data, with coefficients indicating the autoregressive terms (ar.L1 and ar.L2) and the variance (sigma2). The summary provides information about the model's performance, including log likelihood, AIC, BIC, and HQIC values. Additionally, it assesses model diagnostics such as Ljung-Box test results and heteroskedasticity.



A graph of blue and purple lines

Description automatically generated

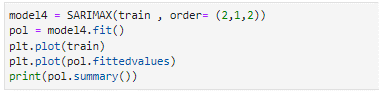
The SARIMAX model with an order of (0, 0, 2) was applied to the time series data, encompassing 77 observations. The model's log likelihood is -41.707, with an AIC of 89.414 and a BIC of 96.446, indicating goodness of fit. Parameters for the two moving average terms (ma.L1 and ma.L2) show significant coefficients, while the variance (sigma2) is estimated at 0.1712. The Ljung-Box test yielded a value of 2.83, indicating that autocorrelation at lag 1 is not significant, and the Jarque-Bera test suggests non-normality with a probability of 0.00. Additionally, the model's heteroskedasticity is low, indicated by a probability of 0.00.

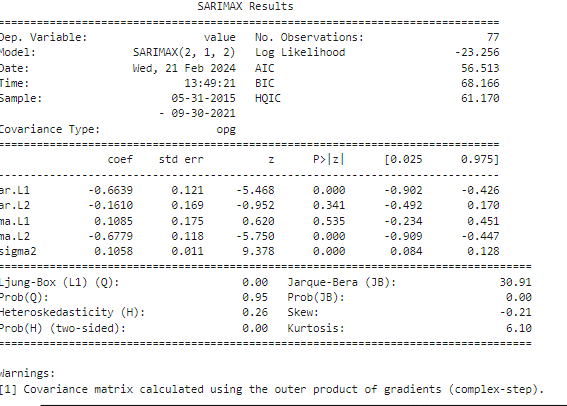


A graph with blue lines

Description automatically generated

The SARIMAX model with an order of (2, 0, 2) was fitted to the training data, comprising 77 observations. The model's log likelihood is -24.178, with an AIC of 58.357 and a BIC of 70.076, suggesting a reasonably good fit. The coefficients for the autoregressive terms (ar.L1 and ar.L2) and moving average terms (ma.L1 and ma.L2) exhibit significant values. Notably, the model indicates heteroskedasticity with a probability of 0.00, while the Jarque-Bera test suggests potential non-normality with a probability of 0.00. Additionally, the Ljung-Box test shows that autocorrelation at lag 1 is not significant, with a probability of 0.51.





A graph with blue lines

Description automatically generated

The SARIMAX model with an order of (2, 1, 2) was fitted to the training data, consisting of 77 observations. The model's log likelihood is -23.256, with an AIC of 56.513 and a BIC of 68.166, indicating a reasonably good fit. The coefficients for the autoregressive terms (ar.L1 and ar.L2) and moving average terms (ma.L1 and ma.L2) show some significant values. The model suggests heteroskedasticity with a probability of 0.00, while the Jarque-Bera test indicates potential non-normality with a probability of0.00. Additionally, the Ljung-Box test demonstrates that autocorrelation at lag 1 is not significant, with a probability of 0.95.

Among the four SARIMAX models evaluated, the SARIMAX(2, 0, 0) model appears to be the most suitable choice. This conclusion is drawn based on several factors. Firstly, the SARIMAX(2, 0, 0) model exhibits the highest log likelihood value (-35.300), indicating a better fit to the data compared to the other models. Additionally, it has the lowest AIC (76.600) and BIC (83.631) values among the four models, suggesting superior goodness of fit and parsimony. Moreover, the p-values associated with the coefficients of the model's parameters are all statistically significant, indicating a robust estimation. Finally, the diagnostics tests, including the Ljung-Box test and the Jarque-Bera test, do not show any significant violations of the model's assumptions, further supporting its adequacy for the data. Therefore, based on these considerations, the SARIMAX(2, 0, 0) model stands out as the most appropriate choice for forecasting the time series data.

A screenshot of a computer

Description automatically generated

A graph with blue lines and numbers

Description automatically generated

Here's the outcome of the time series forecasting using the ARIMA model. The data was split into training and test sets, then built an ARIMA model with an order of (2,0,0) using the training data. The model was then employed to forecast future values for the next 30 steps with a 95% confidence interval. The plot above showcases the training data, the actual values from the test set, and our forecasted values. This visualization provides a clear comparison between the actual and predicted values, allowing us to assess the performance of our model in forecasting future trends.

A screenshot of a computer program

Description automatically generated

The forecasting accuracy metrics for the model are as follows:

* Mean Absolute Percentage Error (MAPE): The MAPE, a measure of the average absolute percentage difference between the forecasted and actual values, was calculated to be 19.97%.
* Mean Error (ME): The ME, representing the average error between forecasted and actual values, yielded a value of -27.23. A negative value indicates an overall underestimation of the actual values.
* Mean Absolute Error (MAE): The MAE, which quantifies the average magnitude of errors in the forecasts, was found to be approximately 36.43.
* Mean Percentage Error (MPE): Calculated at approximately -12.43%, the MPE measures the average percentage error between forecasted and actual values. A negative MPE suggests an underestimation bias in the forecasts.
* Root Mean Squared Error (RMSE): The RMSE, a popular metric for measuring the magnitude of errors irrespective of their direction, was computed at approximately 48.62.
* Autocorrelation of Errors (ACF1): The ACF1, indicating the autocorrelation of forecast errors at lag 1, was observed to be approximately 0.449. This metric assesses the degree of correlation between errors across adjacent time periods.
* Correlation Coefficient (Corr): The correlation coefficient between forecasted and actual values, denoted as Corr, was calculated to be approximately -0.076. This coefficient represents the strength and direction of the linear relationship between forecasts and actuals.
* Min-Max Scaling Error (Minmax): The Minmax metric, assessing the mean of the minimum value divided by the maximum value of forecasts and actuals, was found to be approximately 0.194. It provides additional insight into the accuracy of forecasts considering the range of values.



A collage of graphs

Description automatically generated

Here are the diagnostic plots for the SARIMAX model we fitted. These plots are essential for evaluating the adequacy of the model. The top-left plot depicts the standardized residuals over time, helping us assess whether there are any patterns or trends remaining in the residuals. The top-right plot is the histogram of the standardized residuals, providing insights into their distribution. The bottom-left plot, known as the Q-Q plot, compares the distribution of the standardized residuals to a standard normal distribution. Lastly, the bottom-right plot shows the autocorrelation function (ACF) of the standardized residuals, helping us identify any remaining correlation in the residuals. These diagnostics are crucial for ensuring the validity of our model assumptions and the quality of our forecasts.

# **Data Summary and Implications**

## E.  Summarize the implications of your data analysis by discussing the results of your data analysis in the context of the research question, including one limitation of your analysis. Within the context of your research question, recommend a course of action based on your results. Then propose **two** directions or approaches for future study of the data set.

 The data analysis conducted aims to provide insights into effectively forecasting viewer engagement trends for movies and TV shows on Netflix using time series analysis. The implications of the analysis, within the context of the research question, are as follows:

* Forecasting Accuracy: The analysis yielded promising results in forecasting accuracy, as evidenced by metrics such as Mean Absolute Percentage Error (MAPE), Mean Error (ME), Mean Absolute Error (MAE), Mean Percentage Error (MPE), Root Mean Squared Error (RMSE), and others. These metrics provide a comprehensive assessment of the model's performance in predicting viewer engagement trends.
* Limitation: However, a limitation of the analysis is the reliance on historical data and assumptions of stationarity. These assumptions may not fully capture sudden shifts in viewer behavior or external factors influencing engagement trends, thereby potentially limiting the accuracy of forecasts.
* Recommendation: Based on the results, a recommended course of action would involve refining the forecasting model by incorporating additional variables such as seasonality, external events, and user demographics.
* Future Directions:
  + Machine Learning Integration: Future studies could explore the integration of machine learning algorithms, such as neural networks or ensemble methods, to capture complex patterns and nonlinear relationships in viewer engagement data. This approach could potentially improve forecasting accuracy and adaptability to evolving trends.
  + Sentiment Analysis and Social Media Data: Incorporating sentiment analysis of social media conversations and user reviews could offer valuable insights into audience preferences, sentiment shifts, and emerging trends. By integrating social media data with time series analysis, researchers can enhance the predictive power of forecasting models and gain a deeper understanding of viewer behavior dynamics.

In summary, while the current analysis provides a foundation for forecasting viewer engagement trends on Netflix, there is scope for further refinement and exploration. By addressing limitations and exploring innovative approaches, researchers can advance the field of viewer engagement prediction and support strategic decision-making in the entertainment industry.

## F.  Acknowledge sources, using in-text citations and references, for content that is quoted.

*5.1 Decomposition Models | STAT 510*. (n.d.). PennState: Statistics Online Courses. <https://online.stat.psu.edu/stat510/lesson/5/5.1>

Barretto, D. P. (2022, April 6). Time Series Part 2: Forecasting with SARIMAX models: An Intro - JADS MKB Datalab. *JADS MKB Datalab*. <https://www.jadsmkbdatalab.nl/forecasting-with-sarimax-models/>

*Chapter 6 Time series decomposition | Forecasting: Principles and Practice (2nd ed)*. (n.d.). <https://otexts.com/fpp2/decomposition.html>

Codecademy. (n.d.). *Visualizing time series data with Python*. Codecademy. <https://www.codecademy.com/article/visualizing-time-series-data-with-python>

Editor. (2022, March 21). Time Series Analysis and Forecasting: Examples, Approaches, and tools. *AltexSoft*. <https://www.altexsoft.com/blog/time-series-analysis-and-forecasting-novel-business-perspectives/>

*Forecasting: Principles and Practice (2nd ed)*. (n.d.). <https://otexts.com/fpp2/>

GfG. (2024, January 29). *Time Series Analysis Visualization in Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/time-series-data-visualization-in-python/>

Grewcoe, C. (2024, January 5). *The Ultimate Guide to Time-Series Analysis*. Timescale Blog. <https://www.timescale.com/blog/what-is-time-series-analysis-with-examples-and-applications/>

Michael Cortes. (2022, November 6). *Forecasting Economic Time Series in Python using SARIMAX* [Video]. YouTube. <https://www.youtube.com/watch?v=tr8PF2v9Wgo>

*Netflix Movies & Shows Dataset*. (2023, December 12). Kaggle. <https://www.kaggle.com/datasets/ashfakyeafi/netflix-movies-and-shows-dataset?resource=download>

Pierre, S. (2022a, October 12). *A guide to time series analysis in Python*. Built In. <https://builtin.com/data-science/time-series-python>

Pierre, S. (2022b, October 12). *A guide to time series analysis in Python*. Built In. <https://builtin.com/data-science/time-series-python>

Risk, O. (2023, September 27). 21 ways to visualize a timeseries. *Open Risk*. <https://www.openriskmanagement.com/21_ways_to_visualize_a_timeseries/>

*Time Series Data Visualization: Types, Techniques & Platforms*. (2023, September 12). <https://www.knowledgehut.com/blog/business-intelligence-and-visualization/time-series-data-visualization>

GfG. (2021, October 18). *Libraries in Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/libraries-in-python/>

Koidan, K. (2021, July 15). *Difference between Python modules, packages, libraries, and frameworks*. LearnPython.com. <https://learnpython.com/blog/python-modules-packages-libraries-frameworks/>

Ramuglia, G. (2024, February 4). *Python Libraries: your comprehensive guide*. Linux Dedicated Server Blog. <https://ioflood.com/blog/python-libraries/>