Cape Peninsula
University of Technology
creating futures

AUTOTRADER DAILY LISTING COUNTS

A Technical Report
Presented to

AutoTrader

by

Zintlanu Bozana
STUDENT NO. 220067481

As an Assessment for the Module
MATHEMATICAL SCIENCES PROJECT 4 (MSP470S)
Within the Qualification
ADVANCED DIPLOMA IN MATHEMATICAL SCIENCES

ACADEMIC SUPERVISOR: Dr. Milaine Sergine Seuneu Tchamga

Cape Peninsula University of Technology

October 2024

DECLARATION

I, Zintlanu Bozana, declare that the contents of this research report represent my own work. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Click on the box to confirm your agreement: ☒

Date of Declaration: 01 October 2024

ACKNOWLEDGEMENTS

Special thanks to my academic supervisor Dr. Milaine Sergine Seuneu Tchamga for the guidance and supervision from the beginning of the research till the end. Additionally, a huge appreciation to the AutoTrader company for the provision of the datasets implemented on this study and a huge gratitude to the department of Mathematics and Physics for granting me the opportunity and permission to conduct this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AR - Autoregressive

SARIMA - Seasonal Autoregressive Integrated Moving Average

SARIMAX - Seasonal Autoregressive Integrated Moving Average Exogenous

MSE - Mean Squared Error

RMSE - Root Mean Squared Error

ARMA - Autoregressive Moving Average

ARIMA - Autoregressive Integrated Moving Average

ACF - Autocorrelation Function

PACF – Partial Autocorrelation Function

GLOSSARY OF TERMS

Daily Listing Count - refers to the total number of new listings added to a website or marketplace on a given day.

Model - refers to a mathematical or computational representation of a system.

Time series - refers to a sequence of data points recorded in chronological order typically spaced at intervals such as day, year, month etc.

EXECUTIVE SUMMARY

The research problem of the study is to develop a machine learning model or statistical model that is capable of predicting the daily listing counts in the website. Thus, this study developed a model to predict AutoTrader's daily listing counts at least seven days in advance, enabling accurate revenue and budget forecasting. In contrast, the type of a database implemented in this study is a secondary database which was provided by AutoTrader. Furthermore, python 3 software (jupyter notebook) was used to process, execute, visualize data and generate outcomes. In attempt to select the best fit times series model for forecasting of the daily listing counts for Autotrader, it was discovered that the data is non-stational. Seasonal autoregressive integrated moving average model gave the most accurate predictions compared to other time series models as it has the lowest root mean squared error. Additionally, the best fit model is independent of the presence of exogenous variables, this implies that adding exogenous variables (like holidays, promotions etc) to the existing data does not improve the daily listing count forecasts. Probably, as the time goes on and more data is captured, adding exogenous might have positive impact on the predictions as the model is regularly retrained on the new data.

1. INTRODUCTION

1.1 Background

AutoTrader was founded on the 16th of April in 1992 as a print business specialising in magazine publishing (AutoTrader, n.d). The world wide web is the most popular platform for locating items, customers, and sellers globally and has taken the word by a storm (Copeland &Le Blanc, 2013:8). As a result, AutoTrader has successfully transitioned into a digital marketplace and has created an impressive website where consumers can search for cars (Torku, 2022:3). It essential for a company to have an idea of how the daily listing counts will be in the next upcoming days. Additionally, knowing the daily listing counts in advance will enable the stakeholders to make meaningful business decisions, marketing planning, allocate resources based on the model insights and take preventive measures before the damage. Therefore, AutoTrader desires to develop a machine learning or a statistical model that is capable of forecasting daily listing counts for at least seven days in advance.

Time series dataset is either stational or non-stational and select the best time series model is dependent on the attributes of the data. For example, if the dataset is stational, then models to be considered are AR, ARMA and ARIMA. Alternatively, if the dataset is non-stational then best models will be SARIMA and SARIMAX. Time series of a real data is typically non-stational (Zhu, 2014:1859). Subsequently, our dataset is a time series data, therefore this study will explore all time series model and observe which model exhibit a better fit for the data.

1.2 Business Problem

AutoTrader is an online marketplace for the automotive industry, enabling the company to find new or used cars. Additionally, the main goal of Autotrader is to be able to have an ultimate solution for selling new cars and certified used cars for consumers (Copeland &Le Blanc, 2013:8). AutoTrader wants to predict the daily count listings at least for the next 7 days in advance. The provided dataset that has two variables, the date and the daily listing count. The study requires an adoption of machine learning technique or a statistical model that will assist in predicting the daily listing counts. In contrast, the company pronounces that it is crucial to know the number of listings in their website in advance enabling them to forecast revenue and budget consistently. Daily listing count estimation is significant for market trend prediction, customer behavior analysis, and system resource allocation. As an example, in the automotive industry, companies need the daily listing count to allocate resources for hosting the millions of vehicles advertised on their websites. Furthermore, if the company does not use the relative data to accurately estimate the listing counts, the revenue will not be consistent because they will be no preventative measures taken to prevent drop in sales revenue or instability when encountered.

1.3 Chosen Solution(s)

1.3.1. Autoregressive (AR) Model

a. Objectives

- Provide accurate daily forecasts for at least 7 days in advance to aid in planning and resource allocation.
- Business decisions that could be improved from forecasting the daily listing counts; inventory level (or management), marketing strategies, improve customer service etc.
- Check if there is a linear relation or correlation between the date and the number of daily listings.
- Automate process to predict & replenish daily listing counts.

b. Deliverables

- Code and documentation: the AR model is delivered with comprehensive code in Python along with data preprocessing, training and prediction aspects of the model.
- Train an AR model, which will predict the listing count in each day.
- Deliver a graph or a table presentation representing the predictions in an accessible way for stakeholders to consult.

- It will use the MSE (mean squared error) and RMSE (root mean squared error) metric for model evaluation.

c. Benefits

- Improved Operations: By forecasting listing counts, business units will be able to better prepare for expected volume which could impact staffing, infrastructure load and marketing efforts.
- Reduced expenses: By using forecasting to more tightly couple your resources with anticipated demand, you can control the bottom-line costs associated with server capacity and marketing.
- Increased income: targeted marketing can convert better, and more activity surges will mean more money.

1.3.2. Seasonal Autoregressive Moving Average (SARIMA)

a. Objectives

- Construct a model that will forecast the daily listing counts for at least seven days in advance with the focus of seasonality and trend analysis.
- Provide actionable business insights from the model outcomes that will assist in resource allocation, marketing strategies and in decision-making etc.

b. Deliverables

- The project will be conducted in Python (Jupyter notebook).
- The data preprocessing will be done on python which includes cleaning od data and variable conversions.
- Time series plots, ACF/PACF plots.
- Develop a trained SARIMA model that will predict the daily listing counts using parameter tuning (p, d, q) and seasonal (P, D, Q) parameters for the best model fit.
- Provide a visual or tabular of predicted daily listing counts along with key metrics like, MSE and RMSE.

c. **Benefits**

- SARIMA model can handle seasonality and trend in the dataset (Hossein et al., 2021).
- The daily listing count prediction will increase the improved marketing strategies.
- The model outcome will enable the business to take data-driven decisions from model insights.
- Regulatory predictions enable the company to identify the peak periods and monitor their business strategies during a specific period.

1.3.3. Seasonal Autoregressive Moving Average with Exogenous variables (SARIMAX) Model

a. **Objectives**

- Forecast daily listing count for at least 7 days in advance which will assist in planning, marketing strategies of the business and resource allocations.
- Provide reliable forecasts that will support in formative business decision-making.
- Identify and model seasonal effects of the data using SARIMAX model.

b. **Deliverables**

- Software to be implemented in this study is Python.
- Data processing and variable conversions will be performed in Python.
- Develop a trained SARIMAX model that can predict the daily listing counts with high level of accuracy.
- Provide a graph or a tabular that will elucidate the forecasted daily listing counts.
- Evaluate the model using MSE or RMSE metrices.

c. **Benefits**

- Regularly forecasted daily listing counts will improve that marketing strategy and reduce waste of resources and assist the company with optimization based on the anticipated fluctuations.

- SARIMAX is capable of handling seasonality in the dataset.
- SARIMAX model considers exogenous variables like public holidays, promotions that contribute or affect daily listing counts that are other models do not consider.
- The optimized resource allocation will assist in curbing the cost expenses in association with over and under preparation.

2. METHODOLOGY

2.1. Data Source

The dataset implemented in this study is a secondary data which is a time series data and is provided by AutoTrader. The dataset represents daily listing counts from 2021-03-22 to 2024-08-18.

2.2. Data Description

Table 1: Variable description

Variable	Type of a variable	Unit of measure
Daily Listing Count	Dependent	Units
Date	Independent	Date format

2.3. Data Preparation

- The excel file containing the dataset was imported to python3 (Jupyter script).
- Inspected the dataset for missing values.
- Transform the date variable into an index column.
- Assignment of new variables in the dataset like exogenous variables might be required if the model does not perform well with the current variables. e.g. holidays, day of the week, promotions etc.
- Quantification of exogenous variables from non-numeric to numeric data will also be required.
- Deploy Dickey Fuller approach to test for stationarity.

2.4. Methods

2.4.1. Autoregressive (AR) Model

In this section we will cover Autoregressive analysis on daily listing count data by predicting the daily listing counts for at least seven days ahead precisely using AR model. The term autoregression describes a regression of the variable against itself. An autoregression is run against a set of lagged values of order p. The model is expressed as AR(p). Where:

- AR- an autoregressive component that considers previous values for prediction.

Formula:

$$y_t = c + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} \ + \cdots + \emptyset_p y_{t-p} \ + \varepsilon_t \ldots \ldots \ldots . . (1)$$

Where c is a constant, $\emptyset_1, \emptyset_2, \ldots \ldots, \emptyset_p$ are lag coefficients to order p, ε_t is a white noise and y_t represents dependent variable at time period t.

Model Assumptions

- An Autoregressive model assumes that the output variable depends linearly on its own previous values.
- The data must be stationary, this implies that the value of the dependent variable is constant and do not change over time.
- Lag coefficients are usually less than 1, lag coefficients refers to the measure of the influence of past values on the current values.

2.4.2. Seasonal Autoregressive Integrated Moving Average (SARIMA) Model

This section will cover autoregressive analysis on daily listing count data by predicting the daily listing counts for at least seven days in advance precisely using SARIMA model. The SARIMA model can be expressed as follows SARIMA(p, d, q)(P, D, Q)m (Nontapa et al., 2021:1343). Where:

- S-seasonal component that determines the seasonality period e.g. monthly or weekly etc.
- AR- an autoregressive component that considers previous values for prediction.
- I-is an integration component for differencing where necessary.
- MA-Moving average component is responsible for forecasting previous errors.

p, d and q are ARIMA parameters.

- SARIMA accepts additional components P, D and Q which represent the seasonal regression, differencing and moving average coefficients, and m represents the number of data points (rows) in each seasonal cycle.

Formula:

$$\emptyset_p(B)\mathit{\Phi}_P(B)^m(1-B)^d(1-B)^D y_t = \theta_q(B)\mathit{\Theta}_Q(B^m)\varepsilon_t \ldots\ldots\ldots(1)$$

$$\left(1-\sum_{i=1}^{p}\emptyset_i\, B^i\right)\left(1-\sum_{i=1}^{P}\mathit{\Phi}_k B^{km}\right)z_t = \left(1-\sum_{j=1}^{q}\theta_j B^j\right)\left(1-\sum_{j=1}^{Q}\mathit{\Theta}_l B^{lm}\right)\varepsilon_t \ldots\ldots\ldots(2)$$

Where $\emptyset_p(B)$ represents a polynomial in the backshift operator B of order p, $\mathit{\Phi}_P(B)^m$ denotes seasonal component of the autoregressive polynomial, $\mathit{\Theta}_Q(B^m)$ represents the seasonal component of the moving average polynomial. Additionally, B is the backshift operator, m represents the seasonal period, d and D are integers representing the degrees of non-seasonal and seasonal differencing respectively. y_t represents the times series variable at time t and ε_t denotes the transformed time series variable, often obtained after differencing.

Model Assumptions

- The SARIMA model assumes that the time series data should be stationary.
- The model assumes that the seasonal pattern is consistent and repeats over fixed intervals.

- The underlying data is assumed to be linear and follows a statistical distribution, often the normal distribution.
- The residuals (errors) from the model should be independent and normally distributed.

2.4.3. Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX) Model

In this section we will cover Autoregressive analysis on daily listing count data by predicting the daily listing counts for at least seven days ahead precisely using SARIMAX model. SARIMAX model is also a SARIMA model but with Exogenous variables denoted SARIMAX(p, d, q)(P, D, Q)s (Nontapa et al., 2021:1344). Where:

- S-seasonal component that determines the seasonality period e.g. monthly or weekly etc.
- AR-an autoregressive component that considers previous values for prediction.
- I-is an integration component for differencing where necessary.
- MA- moving average is the component used to predict errors while comparing previous actual outputs with its forecasted output.
- X-exogenous variable component for external variables like holidays that affect the daily listing counts.

Formula:

The exogenous variables can be modelled using Multiple Linear Regression (MLR) model which can be denoted as follows:

$y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \cdots + \beta_k X_{k,t} + \omega_t \ldots \ldots \ldots (1)$,

where β_0 is the constant, β_1,…, β_k are parameter coefficients, $X_{1,t}$…,$X_{k,t}$ are observations of k exogenous variables that corresponding with the dependent variable y_t and ω_t is a stochastic residual.

$\omega_t = \frac{\theta_q(B)\Theta_Q(B^S)}{\emptyset_p(B)\Phi_p(B^S)(1-B)^d(1-B^S)(1-B^S)^D}\varepsilon_t \ldots \ldots \ldots (2).$

The general SARIMAX equation is acquired by substituting with equation (2) into equation (1).

$$y_t = \beta_0 + \sum_{i=1}^{k} \beta_1 X_{i,t} + \frac{\theta_q(B)\Theta_Q(B^S)}{\emptyset_p(B)\Phi_p(B^S)(1-B)^d(1-B^S)(1-B^S)^D}\varepsilon_t \dots\dots\dots (3)$$

Model Assumptions

- The data must be stationary and linear.
- If the external variables are included, they should be relevant and have an impact on the time series modelled.
- Normal distribution of the residual errors.
- The model assumes that the seasonal component is consistent and can be captured by seasonal components.

3. RESULTS AND DISCUSSION

The daily listing counts results of AutoTrader using AR, SARIMA and SAXIMAX models to forecast daily listing counts for at least seven days in advance and selecting the best fit model are shown below:

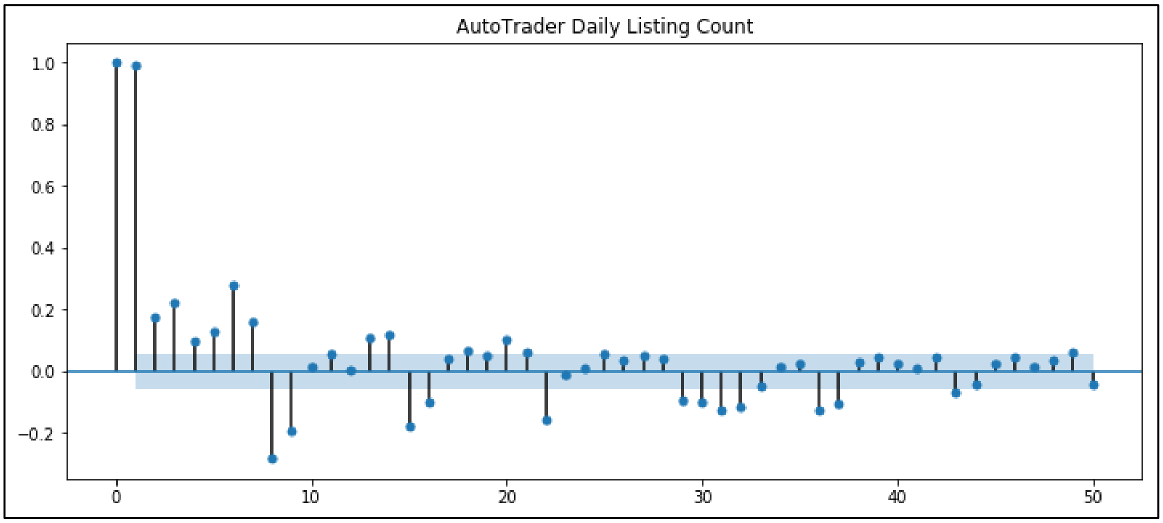


Figure 1: Partial Autocorrelation Function (PACF)

An AR(2) model is suggested by the PACF plot. The PACF plot shows a strong correlation at lag 1, indicating that the daily listing count is highly correlated with the previous day's count. There are also significant correlations at lags 7 and 14, suggesting a weekly pattern.

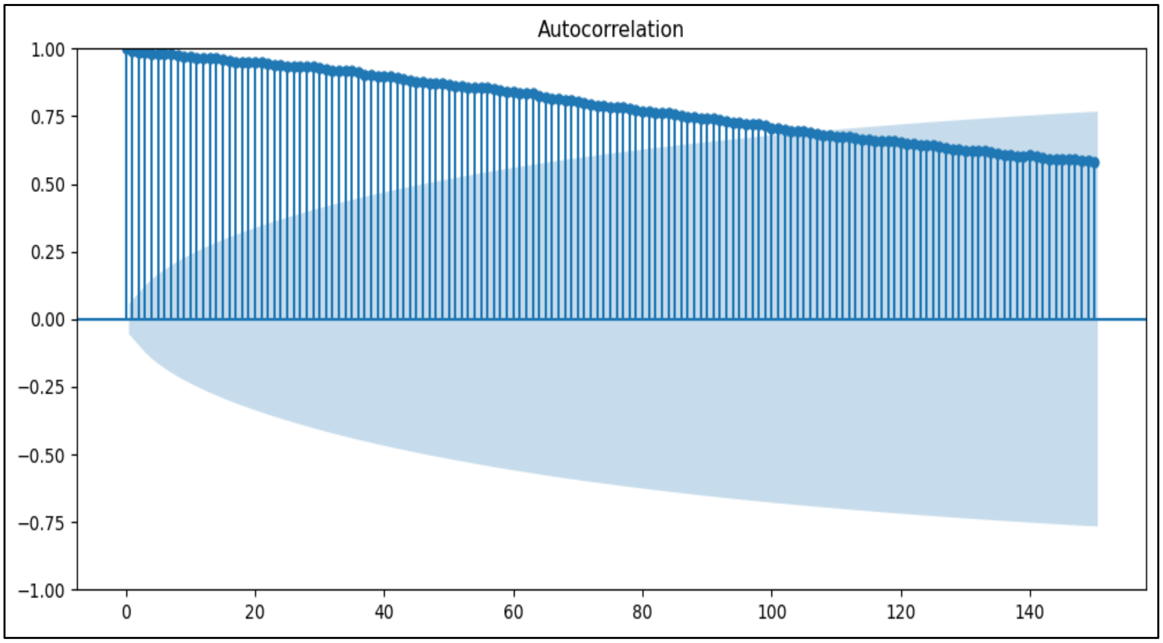


Figure 2: Autocorrelation Function (ACF)

The ACF shows a positive autocorrelation that decays gradually, suggesting a process with temporal dependence, possibly AR or MA or ARMA. Further analysis is needed to specify the model. The ACF plot shows a strong positive autocorrelation at lag 1, indicating a strong correlation between the current value and the previous value. The autocorrelation gradually decreases as the lag increases, suggesting that the correlation between values weakens as they become further apart in time. The ACF plot also shows a significant autocorrelation at lag 7, which could indicate a seasonal pattern with a period of 7. The ACF plot suggests that the data is likely to be autoregressive, meaning that past values can be used to predict future values. The strong autocorrelation at lag 1 suggests that an AR(1) model might be appropriate for modelling the data.

Table 2: Autoregressive Model Summary

const	272.195347
L1.Daily Listing Count	0.594662
L2.Daily Listing Count	0.173074
L3.Daily Listing Count	0.122495
L4.Daily Listing Count	0.019843
L5.Daily Listing Count	-0.037807
L6.Daily Listing Count	0.067427
L7.Daily Listing Count	0.241128
L8.Daily Listing Count	-0.080521
L9.Daily Listing Count	-0.127415
L10.Daily Listing Count	-0.041512
L11.Daily Listing Count	0.004926
L12.Daily Listing Count	-0.032168
L13.Daily Listing Count	0.023845
L14.Daily Listing Count	0.189930
L15.Daily Listing Count	-0.067066
L16.Daily Listing Count	-0.116575
L17.Daily Listing Count	-0.032649
L18.Daily Listing Count	-0.004541
L19.Daily Listing Count	-0.009758
L20.Daily Listing Count	0.085408

L21.Daily Listing Count	**0.185171**
L22.Daily Listing Count	**-0.161019**

The table above represents the Autoregressive summary of the daily listing count data. According to the results an Autoregressive model with lag 22 is best fit model.

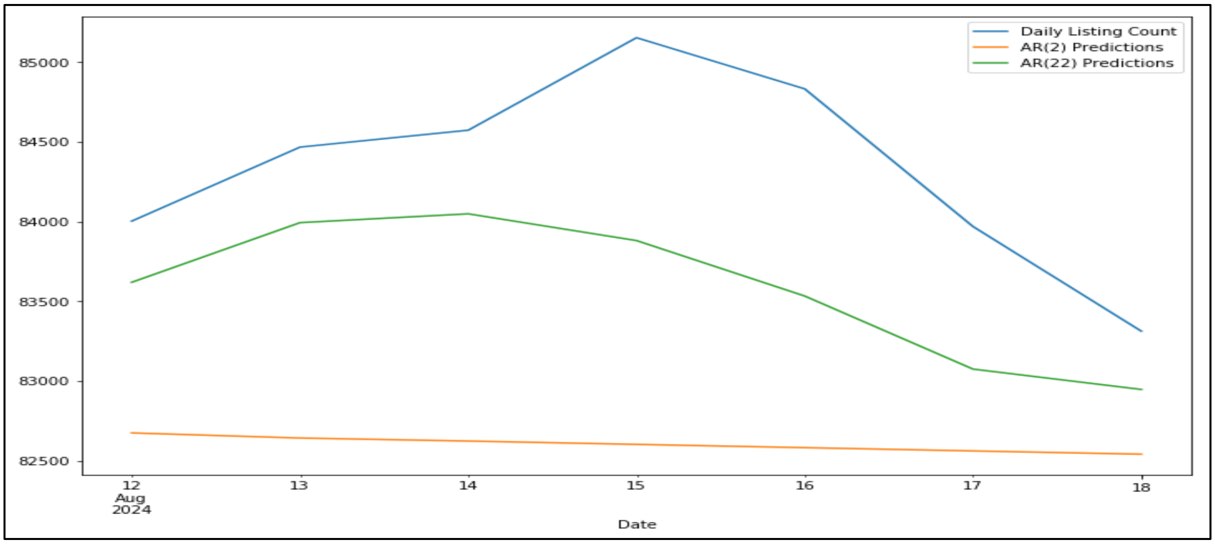


Figure 3: Autoregressive Model: Actual vs Predicted Values

The graph in figure 3 shows the Autoregressive model for the actual daily listing counts, AR(2) model predictions and AR(22) model predictions of daily listing counts from 12 August 2024 to 18 August 2024 using the testing set. AR(22) predictions are generally higher than the AR(2) predictions. This implies that AR(22) gives the best daily lusting count predictions.

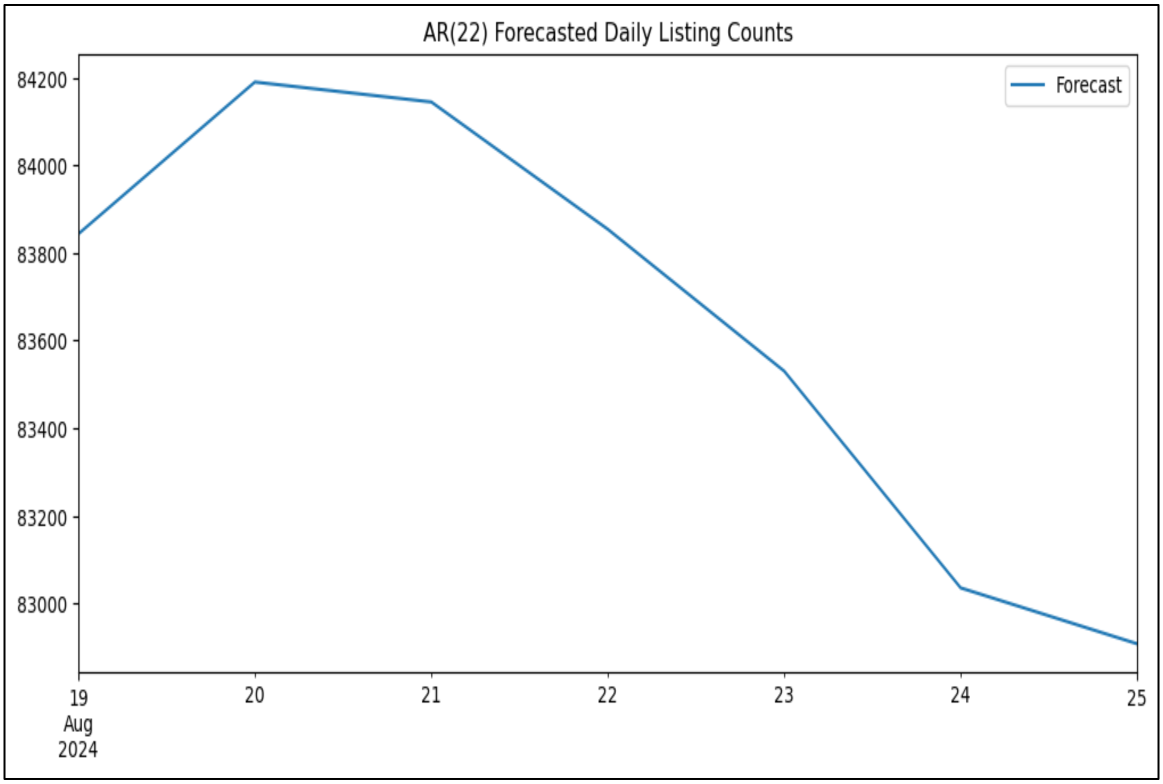


Figure 4: AR(22) Forecasted Daily Counts

The plot represents the forecasted daily listing counts on mode AR(22) of autoregressive model for seven days from the last day on the dataset.

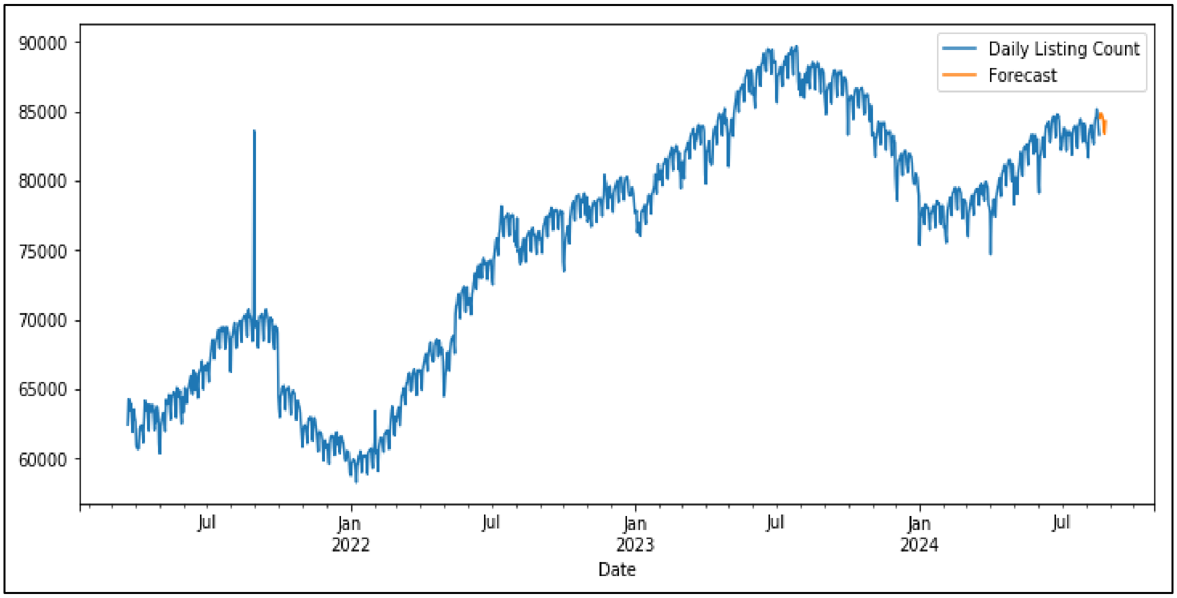


Figure 5: AR Daily Listing Counts and Forecasted Daily Listing Counts

The graph displays a generally increasing trend in daily listing counts from 22 March 2021 to 18 August 2024, characterized by significant short-term fluctuations and a prominent peek in late 2021. The forecast line approximates the overall upward trend but does not reflect the short-term volatility from 12 August 2024 to 18 August 2024.

```
Augmented Dickey-Fuller Test:

ADF Test Statistic      -1.185285

P-value                 0.679893

# Lags used             21.000000

# Observations          1224.000000

Weak evidence against the null hypothesis

Fail to reject the null hypothesis

Data has unit root and is non-stationary
```

Figure 6: Dickey-Fuller Test

The Dickey-Fuller test results above indicate that the data is non-stationary. The p-value is greater than 0.05 significance level.

Table 3: SARIMA Model Summary

SARIMAX Results			
Dep. Variable:	Daily Listing Count	**No. Observations:**	1239
Model:	SARIMAX(3, 0, 1)x(2, 0, [], 7)	**Log Likelihood**	-10080.858
Date:	Fri, 27 Sep 2024	**AIC**	20175.716
Time:	01:04:59	**BIC**	20211.570
Sample:	03-23-2021	**HQIC**	20189.200
	- 08-12-2024		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ar.L1	-0.3237	0.018	-17.733	0.000	-0.360	-0.288
ar.L2	0.9996	0.012	84.081	0.000	0.976	1.023
ar.L3	0.3241	0.007	45.404	0.000	0.310	0.338
ma.L1	0.9999	0.019	53.629	0.000	0.963	1.036
ar.S.L7	0.3498	0.010	33.443	0.000	0.329	0.370

ar.S.L14	0.3053	0.008	36.818	0.000	0.289	0.322
sigma2	7.067e+05	2.71e-08	2.6e+13	0.000	7.07e+05	7.07e+05

Ljung-Box (L1) (Q):	2.88	**Jarque-Bera (JB):**	323020.47
Prob(Q):	0.09	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.36	**Skew:**	1.78
Prob(H) (two-sided):	0.00	**Kurtosis:**	82.02

Table 3 displays SARIMA model summary, according to auto_arima, SARIMAX(3, 0, 1)x(2, 0, [], 7) is the best fit model for the differentiated data which is a stationary data. This implies that ARIMA(3,0,1) is an ARIMA model with three autoregressive terms (AR) and one moving average term (MA). The '0' indicates no differencing. Seasonal AR(2) means the seasonal part of the model has two autoregressive terms (lags at 7 and 14 days), with no seasonality differencing or moving average terms. There the seasonality is weekly as indicated by 7-period lag. AIC = 20175.716 & BIC = 20211.570: These are measures of model fit, where lower values indicate better fit. The BIC is slightly higher than the AIC, as it penalizes model complexity more. Ljung-Box (L1) (Q) = 2.88, p = 0.09: The p-value (0.09) is slightly above the conventional significance level (0.05), suggesting that the residuals are not significantly autocorrelated. The model seems to capture most of the structure in the data, though the result is marginal. Jarque-Bera (JB) = 323020.47, p = 0.00: This large JB statistic with a near-zero p-value indicates the residuals are not normally distributed. This is a common issue in time series models, and while normality is not strictly required for forecasting, it suggests that there may be outliers or skewness in the data. Heteroskedasticity (H) = 0.36, p = 0.00: The p-value indicates evidence of heteroskedasticity, meaning the variance of the residual's changes over time. This suggests that the model might perform better if you address this issue, perhaps by using a model that accounts for changing variance (e.g., a GARCH model). Skewness = 1.78 and Kurtosis = 82.02: These statistics indicate that the residuals are highly skewed (right skewed) and have extreme kurtosis (high number of outliers). This could be important for interpreting the uncertainty around your forecasts, as extreme values might have a bigger influence.

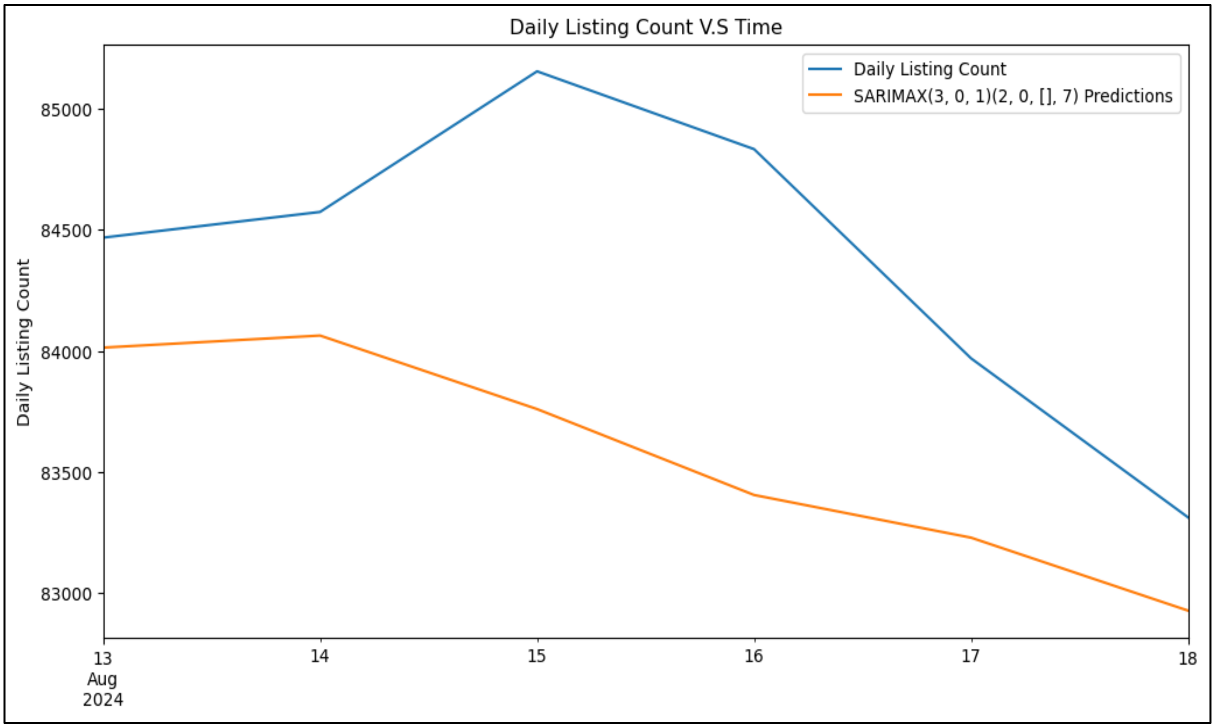


Figure 7: SARIMA Model Actual Values vs Predicted

The graph shows the daily listing count and its SARIMA predictions over 7 days. Th actual daily listing count shows an initial increase followed by a decrease, while the SARIMA predictions show a consistent decrease. The predictions do not accurately reflect the initial increase in daily listing count.

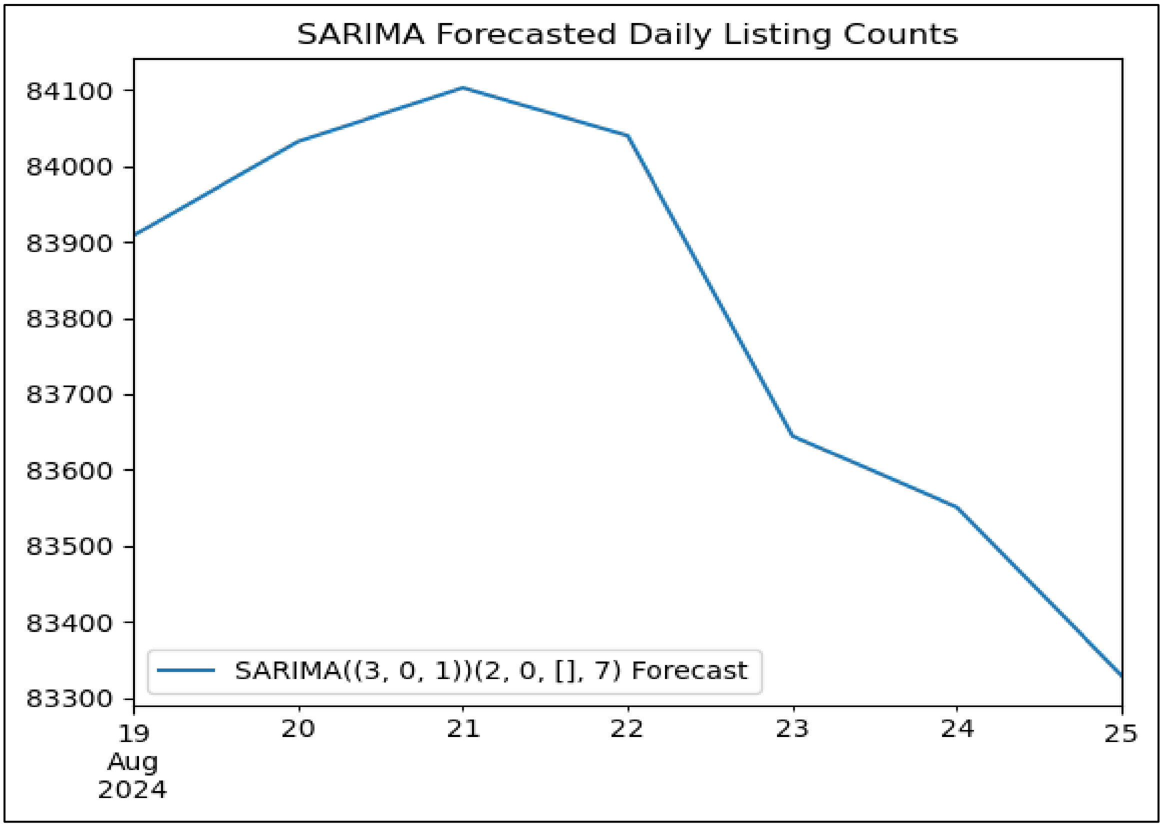


Figure 8: SARIMA Forecasted Daily Listing Counts

The graph above shows the forecasted daily listing counts using seasonal autoregressive integrated moving average (SARIMA) model. The forecasts are for seven days from 19 August 2024 to 25 August 2024, from the last day on the data.

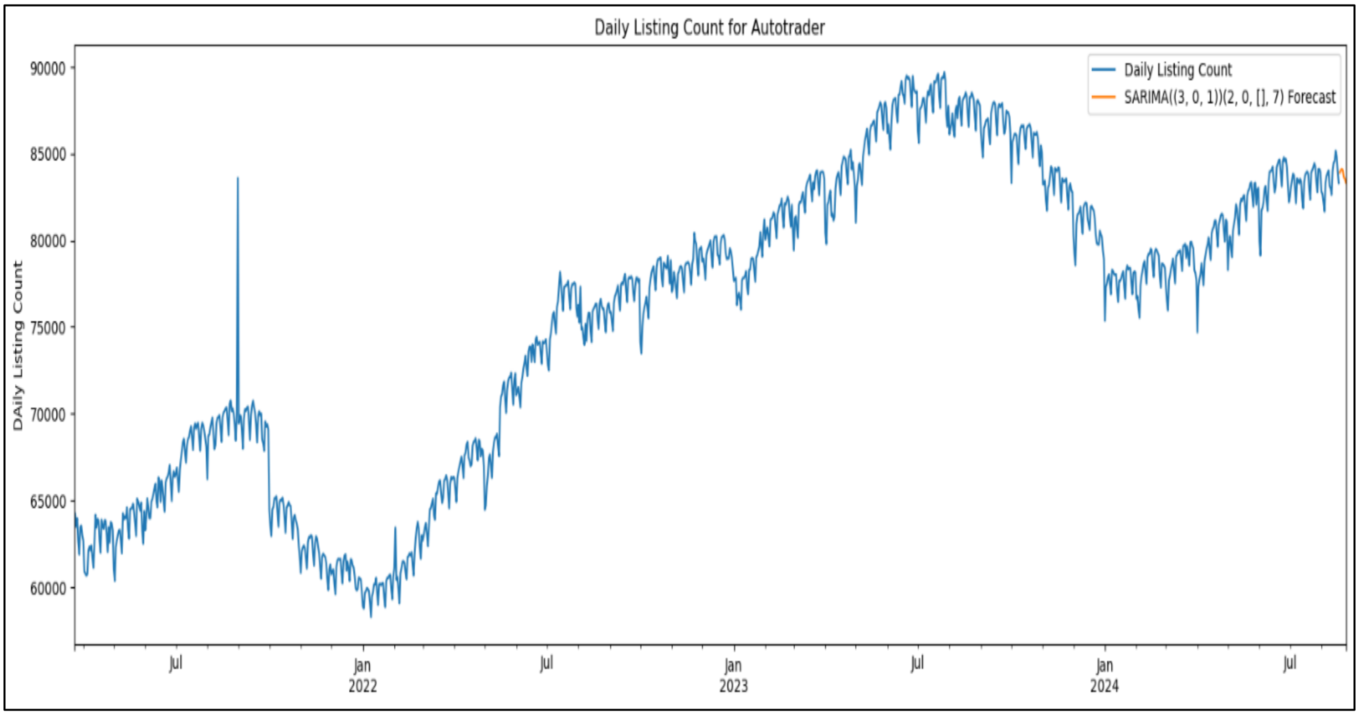


Figure 9: Actual Daily Listing Counts and SARIMA Predictions

The graph shows an upward trend in daily listing counts for AutoTrader from January 2024 to the present. There is a weekly seasonality with peaks occurring on certain days of the week. The forecast based on SARIMA((3, 0, 1))(2, 0, [], 7) model, predicts an increase in daily listing count for the next 7 days.

Table 4: SARIMAX Model Summary

SARIMAX Results

Dep. Variable:	Daily Listing Count	**No. Observations:**	1239
Model:	SARIMAX(3, 0, 1)x(2, 0, [], 7)	**Log Likelihood**	-13899.061
Date:	Fri, 27 Sep 2024	**AIC**	27814.122

Time:	13:13:27	**BIC**	27855.099
Sample:	03-22-2021	**HQIC**	27829.533
	- 08-11-2024		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
Holiday	7.384e+04	867.872	85.081	0.000	7.21e+04	7.55e+04
ar.L1	0.9380	0.332	2.828	0.005	0.288	1.588
ar.L2	-0.1832	0.069	-2.670	0.008	-0.318	-0.049
ar.L3	0.0448	0.024	1.862	0.063	-0.002	0.092
ma.L1	-0.7350	0.334	-2.201	0.028	-1.389	-0.081
ar.S.L7	0.4894	0.015	33.151	0.000	0.460	0.518
ar.S.L14	0.4764	0.015	32.718	0.000	0.448	0.505
sigma2	2.749e+08	0.098	2.8e+09	0.000	2.75e+08	2.75e+08

Ljung-Box (L1) (Q):	0.13	**Jarque-Bera (JB):**	5310.17
Prob(Q):	0.72	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.92	**Skew:**	-1.77

Prob(H) (two-sided):	0.37	**Kurtosis:**	12.51

Table 4 shows SARIMAX model summary, according to auto_arima, SARIMAX(3, 0, 1)x(2, 0, [], 7) is the best fit model for the differentiated data which is a stationary data. This implies that ARIMA(3,0,1) is an ARIMA model with three autoregressive terms (AR) and one moving average term (MA). The ‘0’ indicates no differencing. Seasonal AR(2) means the seasonal part of the model has two autoregressive terms (lags at 7 and 14 days), with no seasonality differencing or moving average termsLog Likelihood: -13899.061, the higher (less negative), the better the model fit. AIC: 27814.122 (Akaike Information Criterion), used for model selection. Lower AIC values indicate a better fit, though model simplicity must be balanced. BIC: 27855.099 (Bayesian Information Criterion), penalizes complexity more heavily than AIC. HQIC: 27829.533 (Hannan-Quinn Information Criterion), another model selection criterion with different penalization for complexity. Ljung-Box (L1): 0.13 (Prob(Q): 0.72), indicating no significant autocorrelation in residuals at lag 1, meaning the model captures the data patterns well. Jarque-Bera (JB): 5310.17 (Prob(JB): 0.00), with a p-value of 0, indicating that the residuals are not normally distributed. This could suggest outliers or skewness. Heteroskedasticity (H): 0.92 (Prob(H): 0.37), indicating no significant heteroskedasticity, meaning the residuals have constant variance over time. Skew: -1.77, which suggests a left skew in the residuals. Kurtosis: 12.51, indicating heavy tails, meaning the data has more extreme values than a normal distribution.

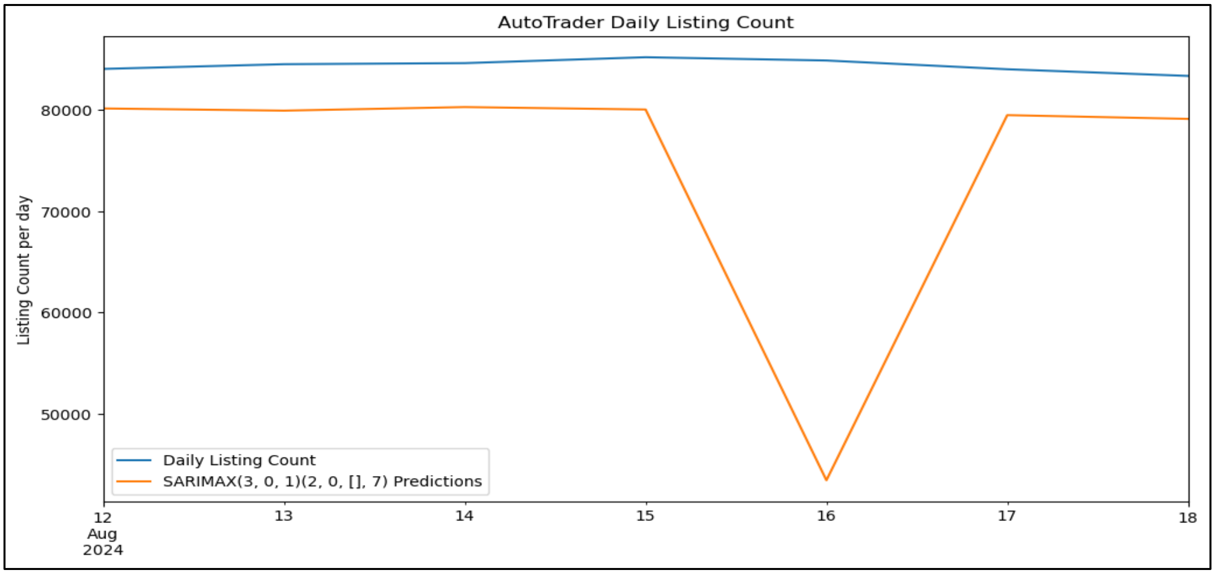


Figure 10: SARIMAX Model Actual vs Predicted

The graph shows a relatively stable daily listing count until a sharp drop around 16th of August, followed by a slight recovery when the exogenous variable holiday is added to the data. The SARIMAX prediction model captures the general trend but shows discrepancies in the exact values compared to the actual daily listing count.

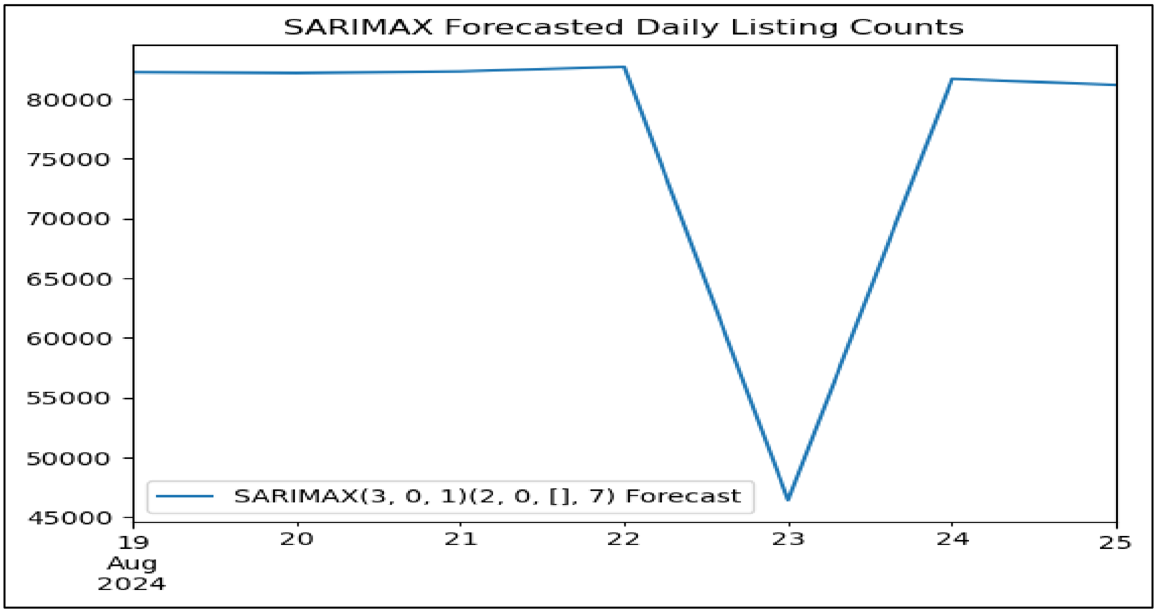


Figure 11: SARIMAX Forecasted Daily Counts

The graph above shows the forecasted daily listing counts using seasonal autoregressive integrated moving average with exogenous(holiday) variable (SARIMAX) model. The forecasts are for seven days from 19 August 2024 to 25 August 2024, from the last day on the data.

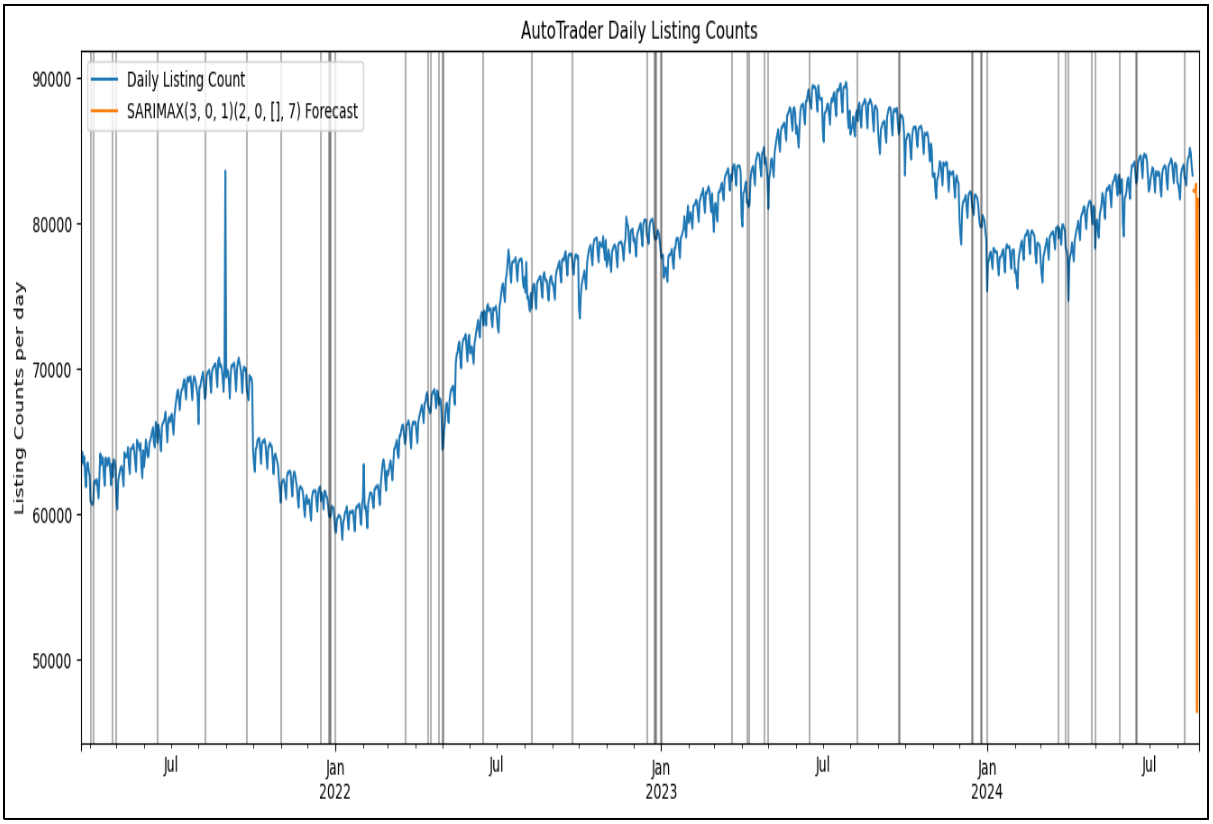


Figure 12: Actual Daily Listing Counts and SARIMAX Daily Listing Counts

The graph displays fluctuating AutoTrader daily listing counts from March 2021 to August 2024, with peaks and troughs indicating periods of high and low listing activity. A forest generally follows the overall trend but does not perfectly predict the daily listing counts foe the next 7 days. The vertical lines on the graph represent days where there are holidays.

Model	Mean	RMSE
AR (22)	84384.5	1146.971224257485
SARIMA	84384.5	925.2970755
SARIMAX	84384.5	16164.50488

Table 5: Model Evaluation

The model was evaluated by means of calculating the mean and the root mean squared error for all three tome series models. The comparison was done between the mean and the root mean squared error for all three tome series models and in order to determine the best fit model, we find the model with the lowest RMSE. Additionally, for a model to be considered to have good approximations it must have a RMSE that is less than a mean. Furthermore, according to the results in the evaluation table, SARIMA exhibits best fit.

4. CONCLUSION AND RECOMMENDATIONS

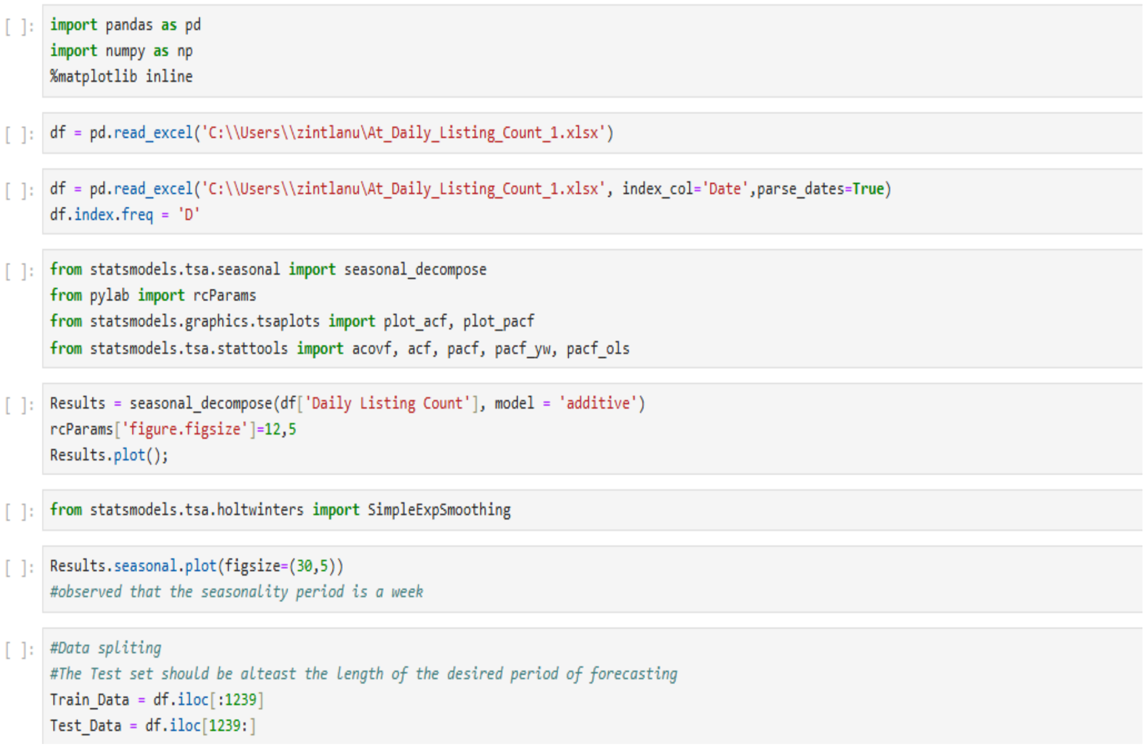
Time series forecasting has been proven to be one of the most demanding subjects during the last few decades since it has many applications in financial, economic and scientific modelling. In contrast, the main objective of this research is to develop a model that will predict the AutoTrader daily listing counts for at least seven days in advance which in turn will allow the company to take formative decisions, improve business marketing strategies and manage resource allocations. This study considered three time series models namely: AR, SARIMA and SARIMAX models. However, there is still a lot of time series models that can be explored when developing a model to model a time series dataset but depending on the structure of the data. In order to select the right time series model, one has to understand the type of the database whether it is seasonal or non-seasonal and stational or not. Subsequently, to check for stationarity, this study made use of seasonality plots. However, sometimes it is hard to tell that the data is seasonal or not by a naked eye by virtue of the pattern may not be vivid enough on the visualization plots. Thus, this study implemented a Dickey-Fuller test which provides quantitative analysis based on the stationarity of the data. Additionally, according to Dickey-Fuller test the dataset is non-stationary.

Since the data possesses nonstationary data, then the study considered SARIMA and SARIMAX models. Using the auto_arima library to select the best fit model from, SARIMAX((3,0,1),(2,0,[],7)) model without exogenous variables exhibits a better fit compared to the AR and SARIMA models according to auto-arima. From the daily listing count predictions of the three models, we observed that SARIMA model gives the better predictions compared to other time series models. Additionally, the model has been evaluated by means of calculating mean and RMSE. SARIMA’s RMSE is 925.2970755 which is less than the mean is 84384.5, this implies that the model is indeed the best fit since it has the lowest RMSE compared to AR(22) and SARIMAX.

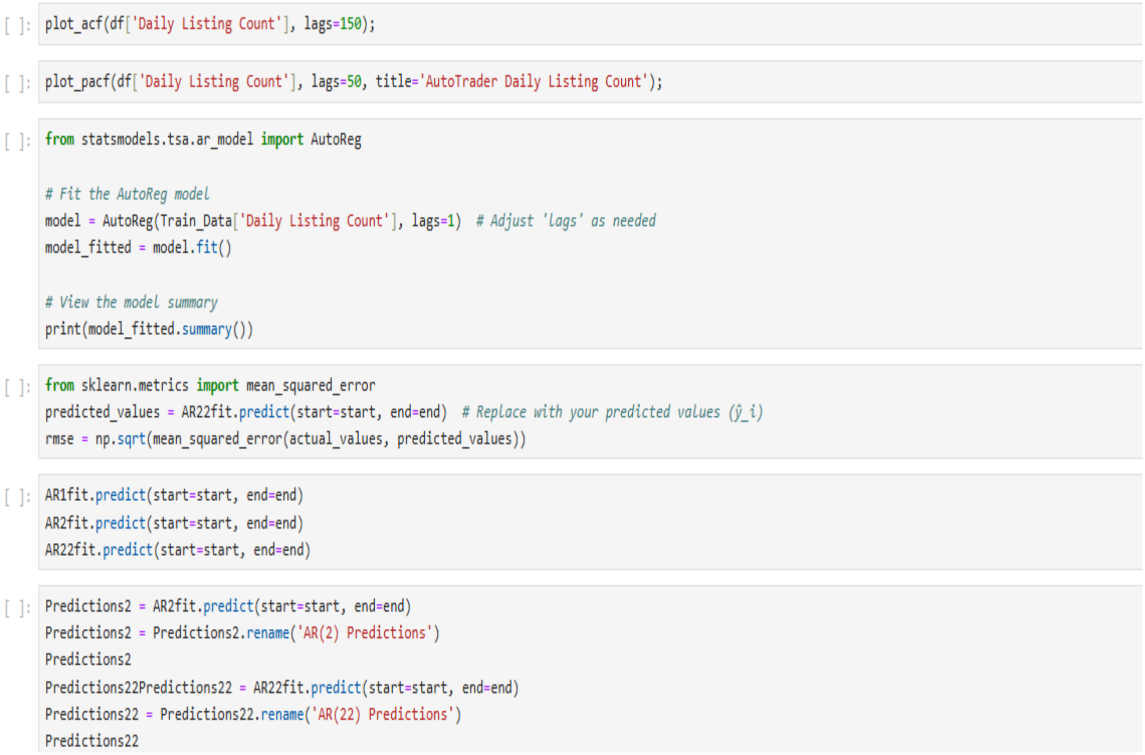
As a result, this study highly recommends the company to implement SARIMA model among AR(22) and SARIMAX, since the SARIMA model was found to be the best fit, then AutoTrader should direct its focus on the use of the SARIMA model. It was identified through the auto_arima library, and it fares well in handling the non-stationarity of the data. Secondly, even though the selected best-fitting SARIMA model was without exogenous variables, there is a need to go ahead and explore whether the addition of relevant external factors may impact this. Examples include: Some economic indicators, like interest rates, inflation, public holidays that could be potential disruptors in listings etc. AutoTrader needs to test whether these can further assist in increasing model accuracy by adding them as exogenous variables to the SARIMAX model. Furthermore, the company should continuously check the data for seasonality. As it is often hard to identify seasonality from visual plots, statistical tests-Dickey-Fuller-could be carried out in order to capture it. This would allow the model to be precise in case of changes in market patterns over time.

Additionally, AutoTrader must reevaluate the model periodically, and the auto_arima method was useful for selecting the best model, but usually periodic retraining and reassessment are very important. Market conditions change, data characteristics change, and user behavior changes, which at some time may require changes in model parameters or the introduction of different models altogether. Lastly, resource allocation and marketing strategy: The model would let AutoTrader make informative business decisions on resource allocations-for instance, staffing or inventory management-and enhance business strategies, targeting campaigns in periods of high or low expected listings.

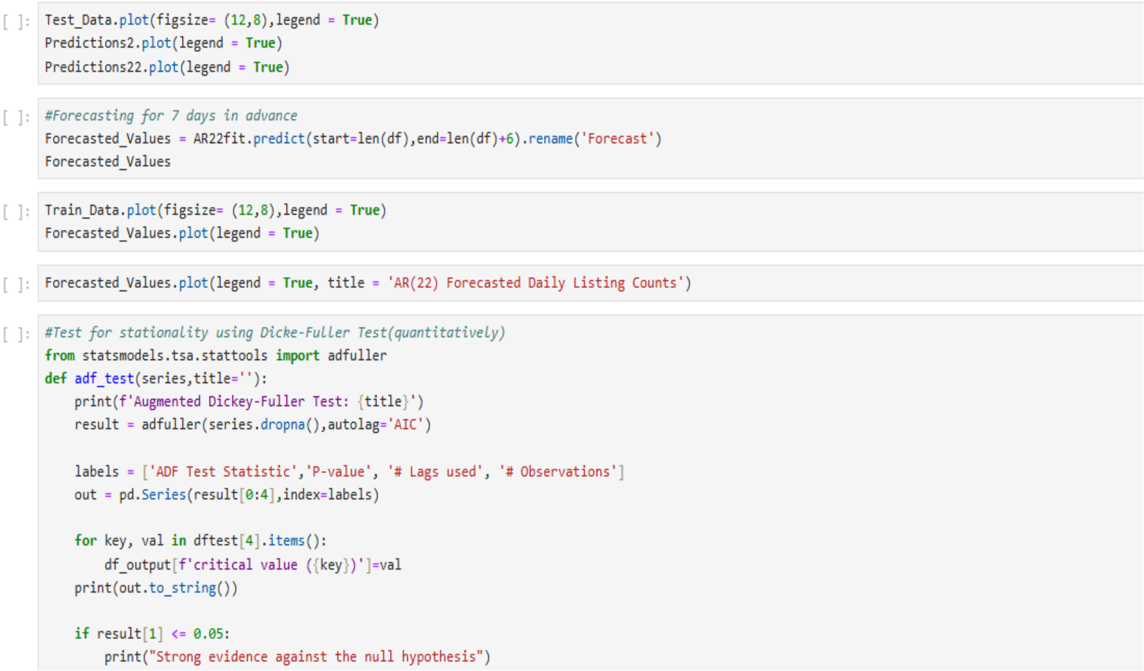
APPENDIX A. PROJECT CODES



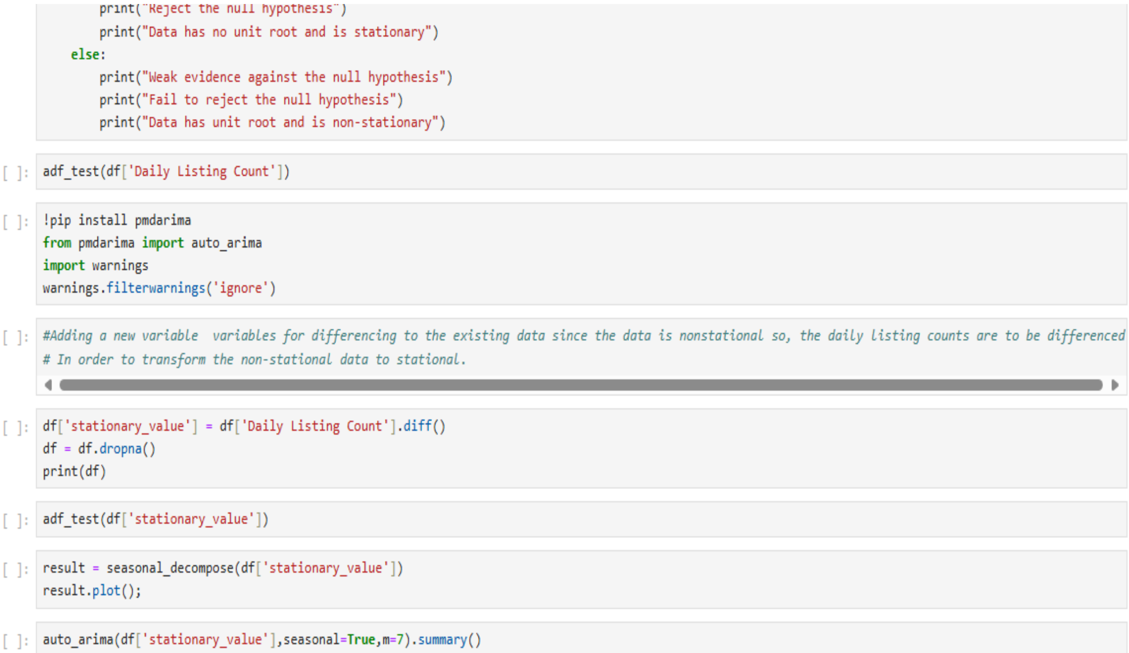
```
[ ]: import pandas as pd
     import numpy as np
     %matplotlib inline

[ ]: df = pd.read_excel('C:\\Users\\zintlanu\At_Daily_Listing_Count_1.xlsx')

[ ]: df = pd.read_excel('C:\\Users\\zintlanu\At_Daily_Listing_Count_1.xlsx', index_col='Date',parse_dates=True)
     df.index.freq = 'D'

[ ]: from statsmodels.tsa.seasonal import seasonal_decompose
     from pylab import rcParams
     from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
     from statsmodels.tsa.stattools import acovf, acf, pacf, pacf_yw, pacf_ols

[ ]: Results = seasonal_decompose(df['Daily Listing Count'], model = 'additive')
     rcParams['figure.figsize']=12,5
     Results.plot();

[ ]: from statsmodels.tsa.holtwinters import SimpleExpSmoothing

[ ]: Results.seasonal.plot(figsize=(30,5))
     #observed that the seasonality period is a week

[ ]: #Data spliting
     #The Test set should be alteast the length of the desired period of forecasting
     Train_Data = df.iloc[:1239]
     Test_Data = df.iloc[1239:]
```



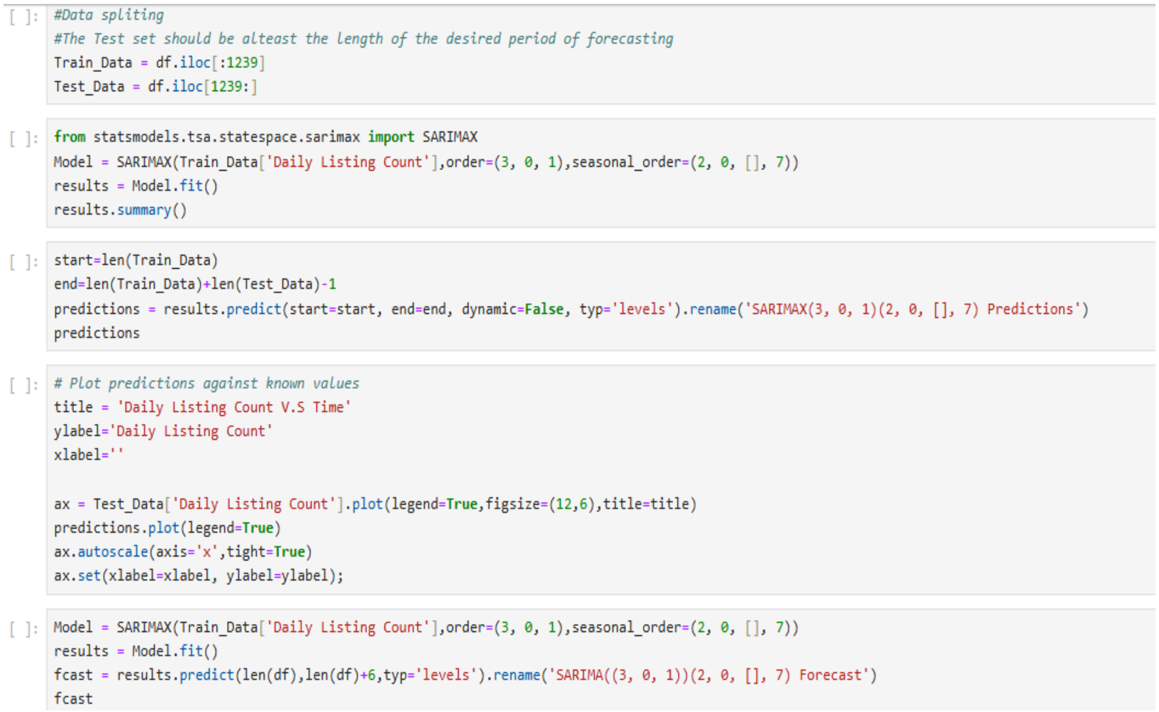
```
[ ]: plot_acf(df['Daily Listing Count'], lags=150);

[ ]: plot_pacf(df['Daily Listing Count'], lags=50, title='AutoTrader Daily Listing Count');

[ ]: from statsmodels.tsa.ar_model import AutoReg

     # Fit the AutoReg model
     model = AutoReg(Train_Data['Daily Listing Count'], lags=1)  # Adjust 'lags' as needed
     model_fitted = model.fit()

     # View the model summary
     print(model_fitted.summary())

[ ]: from sklearn.metrics import mean_squared_error
     predicted_values = AR22fit.predict(start=start, end=end)  # Replace with your predicted values (ŷ_i)
     rmse = np.sqrt(mean_squared_error(actual_values, predicted_values))

[ ]: AR1fit.predict(start=start, end=end)
     AR2fit.predict(start=start, end=end)
     AR22fit.predict(start=start, end=end)

[ ]: Predictions2 = AR2fit.predict(start=start, end=end)
     Predictions2 = Predictions2.rename('AR(2) Predictions')
     Predictions2
     Predictions22Predictions22 = AR22fit.predict(start=start, end=end)
     Predictions22 = Predictions22.rename('AR(22) Predictions')
     Predictions22
```



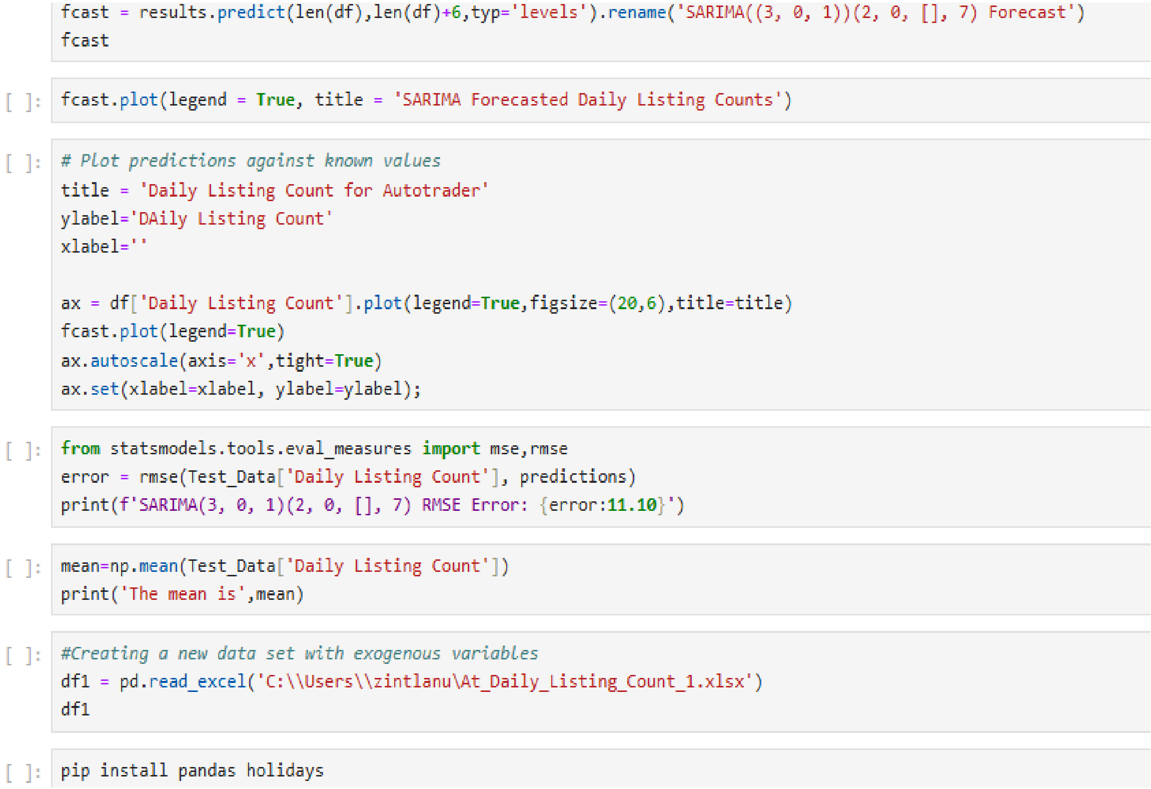
```
[ ]: Test_Data.plot(figsize= (12,8),legend = True)
     Predictions2.plot(legend = True)
     Predictions22.plot(legend = True)

[ ]: #Forecasting for 7 days in advance
     Forecasted_Values = AR22fit.predict(start=len(df),end=len(df)+6).rename('Forecast')
     Forecasted_Values

[ ]: Train_Data.plot(figsize= (12,8),legend = True)
     Forecasted_Values.plot(legend = True)

[ ]: Forecasted_Values.plot(legend = True, title = 'AR(22) Forecasted Daily Listing Counts')

[ ]: #Test for stationality using Dicke-Fuller Test(quantitatively)
     from statsmodels.tsa.stattools import adfuller
     def adf_test(series,title=''):
         print(f'Augmented Dickey-Fuller Test: {title}')
         result = adfuller(series.dropna(),autolag='AIC')

         labels = ['ADF Test Statistic','P-value', '# Lags used', '# Observations']
         out = pd.Series(result[0:4],index=labels)

         for key, val in dftest[4].items():
             df_output[f'critical value ({key})']=val
         print(out.to_string())

         if result[1] <= 0.05:
             print("Strong evidence against the null hypothesis")
```



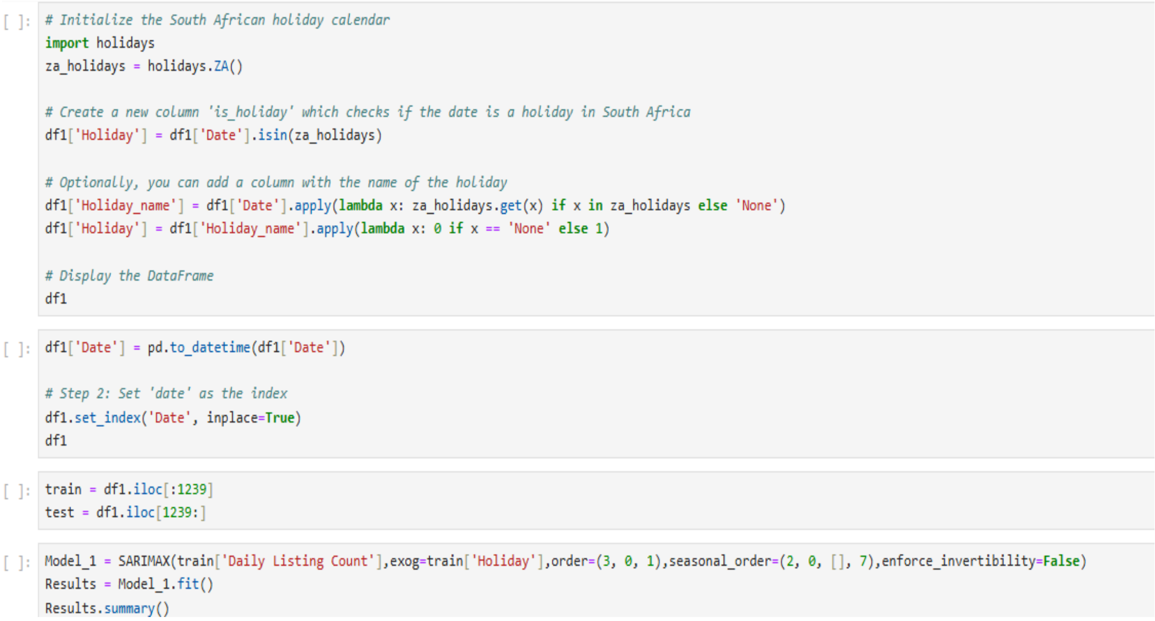
```
             print("Reject the null hypothesis")
             print("Data has no unit root and is stationary")
         else:
             print("Weak evidence against the null hypothesis")
             print("Fail to reject the null hypothesis")
             print("Data has unit root and is non-stationary")

[ ]: adf_test(df['Daily Listing Count'])

[ ]: !pip install pmdarima
     from pmdarima import auto_arima
     import warnings
     warnings.filterwarnings('ignore')

[ ]: #Adding a new variable  variables for differencing to the existing data since the data is nonstational so, the daily listing counts are to be differenced
     # In order to transform the non-stational data to stational.

[ ]: df['stationary_value'] = df['Daily Listing Count'].diff()
     df = df.dropna()
     print(df)

[ ]: adf_test(df['stationary_value'])

[ ]: result = seasonal_decompose(df['stationary_value'])
     result.plot();

[ ]: auto_arima(df['stationary_value'],seasonal=True,m=7).summary()
```



```
[ ]: #Data spliting
     #The Test set should be alteast the length of the desired period of forecasting
     Train_Data = df.iloc[:1239]
     Test_Data = df.iloc[1239:]

[ ]: from statsmodels.tsa.statespace.sarimax import SARIMAX
     Model = SARIMAX(Train_Data['Daily Listing Count'],order=(3, 0, 1),seasonal_order=(2, 0, [], 7))
     results = Model.fit()
     results.summary()

[ ]: start=len(Train_Data)
     end=len(Train_Data)+len(Test_Data)-1
     predictions = results.predict(start=start, end=end, dynamic=False, typ='levels').rename('SARIMAX(3, 0, 1)(2, 0, [], 7) Predictions')
     predictions

[ ]: # Plot predictions against known values
     title = 'Daily Listing Count V.S Time'
     ylabel='Daily Listing Count'
     xlabel=''

     ax = Test_Data['Daily Listing Count'].plot(legend=True,figsize=(12,6),title=title)
     predictions.plot(legend=True)
     ax.autoscale(axis='x',tight=True)
     ax.set(xlabel=xlabel, ylabel=ylabel);

[ ]: Model = SARIMAX(Train_Data['Daily Listing Count'],order=(3, 0, 1),seasonal_order=(2, 0, [], 7))
     results = Model.fit()
     fcast = results.predict(len(df),len(df)+6,typ='levels').rename('SARIMA((3, 0, 1))(2, 0, [], 7) Forecast')
     fcast
```



```
     fcast = results.predict(len(df),len(df)+6,typ='levels').rename('SARIMA((3, 0, 1))(2, 0, [], 7) Forecast')
     fcast

[ ]: fcast.plot(legend = True, title = 'SARIMA Forecasted Daily Listing Counts')

[ ]: # Plot predictions against known values
     title = 'Daily Listing Count for Autotrader'
     ylabel='DAily Listing Count'
     xlabel=''

     ax = df['Daily Listing Count'].plot(legend=True,figsize=(20,6),title=title)
     fcast.plot(legend=True)
     ax.autoscale(axis='x',tight=True)
     ax.set(xlabel=xlabel, ylabel=ylabel);

[ ]: from statsmodels.tools.eval_measures import mse,rmse
     error = rmse(Test_Data['Daily Listing Count'], predictions)
     print(f'SARIMA(3, 0, 1)(2, 0, [], 7) RMSE Error: {error:11.10}')

[ ]: mean=np.mean(Test_Data['Daily Listing Count'])
     print('The mean is',mean)

[ ]: #Creating a new data set with exogenous variables
     df1 = pd.read_excel('C:\\Users\\zintlanu\At_Daily_Listing_Count_1.xlsx')
     df1

[ ]: pip install pandas holidays
```

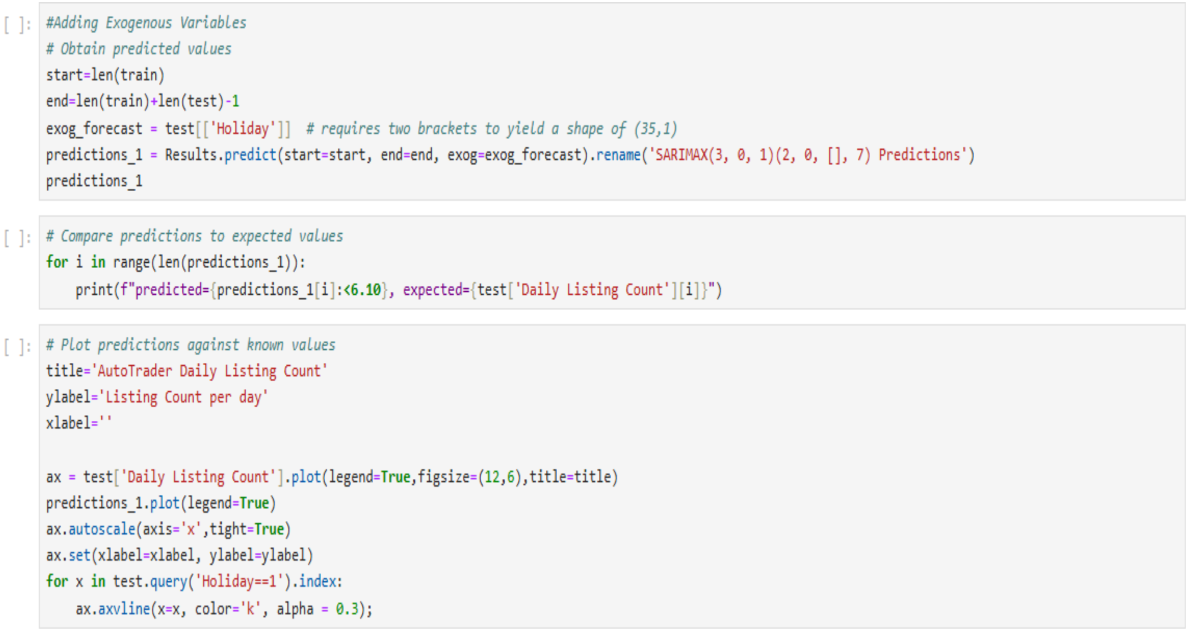


```
# Initialize the South African holiday calendar
import holidays
za_holidays = holidays.ZA()

# Create a new column 'is_holiday' which checks if the date is a holiday in South Africa
df1['Holiday'] = df1['Date'].isin(za_holidays)

# Optionally, you can add a column with the name of the holiday
df1['Holiday_name'] = df1['Date'].apply(lambda x: za_holidays.get(x) if x in za_holidays else 'None')
df1['Holiday'] = df1['Holiday_name'].apply(lambda x: 0 if x == 'None' else 1)

# Display the DataFrame
df1
```

```
df1['Date'] = pd.to_datetime(df1['Date'])

# Step 2: Set 'date' as the index
df1.set_index('Date', inplace=True)
df1
```

```
train = df1.iloc[:1239]
test = df1.iloc[1239:]
```

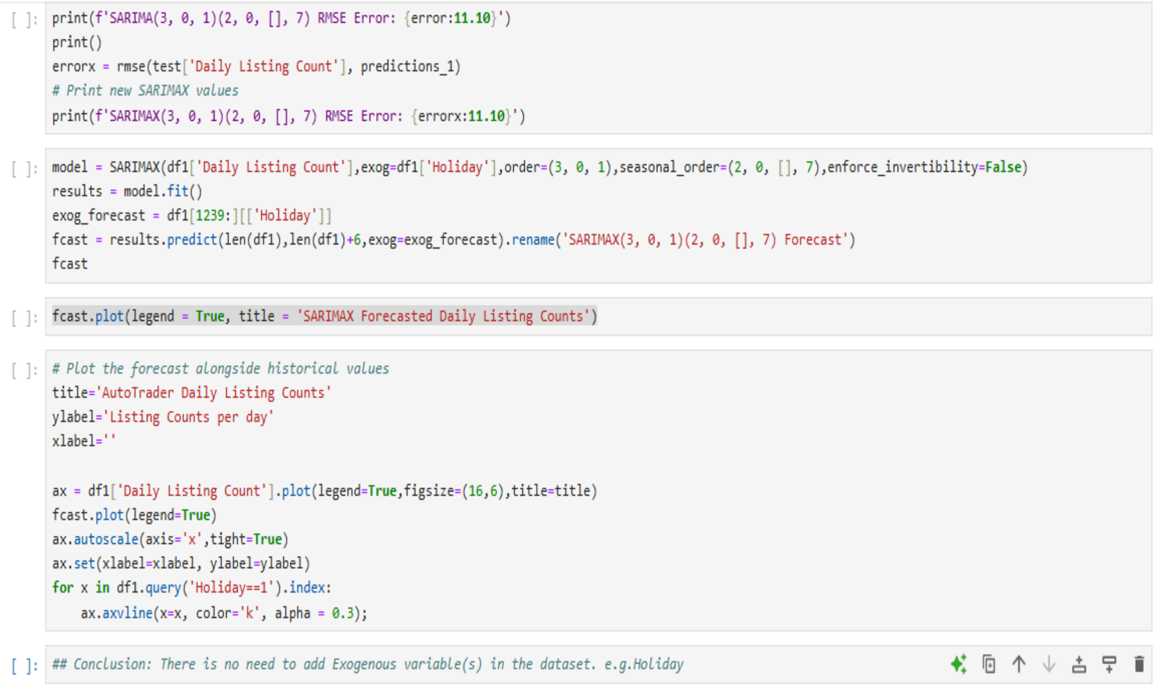
```
Model_1 = SARIMAX(train['Daily Listing Count'],exog=train['Holiday'],order=(3, 0, 1),seasonal_order=(2, 0, [], 7),enforce_invertibility=False)
Results = Model_1.fit()
Results.summary()
```



```
#Adding Exogenous Variables
# Obtain predicted values
start=len(train)
end=len(train)+len(test)-1
exog_forecast = test[['Holiday']]  # requires two brackets to yield a shape of (35,1)
predictions_1 = Results.predict(start=start, end=end, exog=exog_forecast).rename('SARIMAX(3, 0, 1)(2, 0, [], 7) Predictions')
predictions_1
```

```
# Compare predictions to expected values
for i in range(len(predictions_1)):
    print(f"predicted={predictions_1[i]:<6.10}, expected={test['Daily Listing Count'][i]}")
```

```
# Plot predictions against known values
title='AutoTrader Daily Listing Count'
ylabel='Listing Count per day'
xlabel=''

ax = test['Daily Listing Count'].plot(legend=True,figsize=(12,6),title=title)
predictions_1.plot(legend=True)
ax.autoscale(axis='x',tight=True)
ax.set(xlabel=xlabel, ylabel=ylabel)
for x in test.query('Holiday==1').index:
    ax.axvline(x=x, color='k', alpha = 0.3);
```



```
[ ]: print(f'SARIMA(3, 0, 1)(2, 0, [], 7) RMSE Error: {error:11.10}')
     print()
     errorx = rmse(test['Daily Listing Count'], predictions_1)
     # Print new SARIMAX values
     print(f'SARIMAX(3, 0, 1)(2, 0, [], 7) RMSE Error: {errorx:11.10}')
```

```
[ ]: model = SARIMAX(df1['Daily Listing Count'],exog=df1['Holiday'],order=(3, 0, 1),seasonal_order=(2, 0, [], 7),enforce_invertibility=False)
     results = model.fit()
     exog_forecast = df1[1239:][['Holiday']]
     fcast = results.predict(len(df1),len(df1)+6,exog=exog_forecast).rename('SARIMAX(3, 0, 1)(2, 0, [], 7) Forecast')
     fcast
```

```
[ ]: fcast.plot(legend = True, title = 'SARIMAX Forecasted Daily Listing Counts')
```

```
[ ]: # Plot the forecast alongside historical values
     title='AutoTrader Daily Listing Counts'
     ylabel='Listing Counts per day'
     xlabel=''

     ax = df1['Daily Listing Count'].plot(legend=True,figsize=(16,6),title=title)
     fcast.plot(legend=True)
     ax.autoscale(axis='x',tight=True)
     ax.set(xlabel=xlabel, ylabel=ylabel)
     for x in df1.query('Holiday==1').index:
         ax.axvline(x=x, color='k', alpha = 0.3);
```

```
[ ]: ## Conclusion: There is no need to add Exogenous variable(s) in the dataset. e.g.Holiday
```

REFERENCES

AutoTrader. n.d. AutoTrader is South Africa's most trusted motoring marketplace. https://www.autotrader.co.za [29 September 2024].

Copeland, R.D. & Le Blanc, L.A., Online Vehicle Auctions: Ebay, Autotrader, Craigslist and Beyond.

Nontapa, C., Kesamoon, C., Kaewhawong, N. & Intrapaiboon, P., 2021. A New Hybrid Forecasting Using Decomposition Method with SARIMAX Model and Artificial Neural Network. International Journal of Mathematics and Computer Science, 16(4), pp.1341-1354.

Shadkam, A., 2020. Using SARIMAX to forecast electricity demand and consumption in university buildings (Doctoral dissertation, University of British Columbia).

Torku, S., 2022. A web-based application for the sale of used cars which predicts market values.