[AI6123] Project 1: Forecasting wwwusage data

Reinelle Jan Bugnot
G2304329L
bugn0001@e.ntu.edu.sg

***Abstract*—The wwwusage time series data consist of the number of users connected to the internet through a server. The data are collected at a time interval of one minute and there are 100 observations. The task is to fit an appropriate ARIMA model for this data.**

I. Introduction

An Autoregressive Integrated Moving Average (ARIMA) model is a statistical analysis model used for time series data to either better understand the data set or to predict future trends. It's based on the idea that the information in the past values of the time series can alone be used to predict the future values [1].

The ARIMA model is composed of three parts:

1) **Autoregression (AR)**: This refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
2) **Integrated (I)**: This represents the differencing of raw observations to allow the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values.
3) **Moving average (MA)**: This incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each component in ARIMA functions as a parameter with a standard notation. For ARIMA models, a standard notation would be ARIMA with p, d, and q, where integer values substitute for the parameters to indicate the type of ARIMA model used. The parameters can be defined as:

- **p**: the number of lag observations in the model, also known as the lag order.
- **d**: the number of times the raw observations are differenced; also known as the degree of differencing.
- **q**: the size of the moving average window, also known as the order of the moving average.

This project aims to fit an appropriate ARIMA model on the wwwusage data as shown in Fig. 1. The data consists of 100 observations collected at 1 minute intervals, representing the number of users connected to the internet through a server. To facilitate proper evaluation of the model, I did an 80-20 train-test split, where the first 80 observations is used to fit the ARIMA model, and the last 20 observation is used to evaluate model performance.

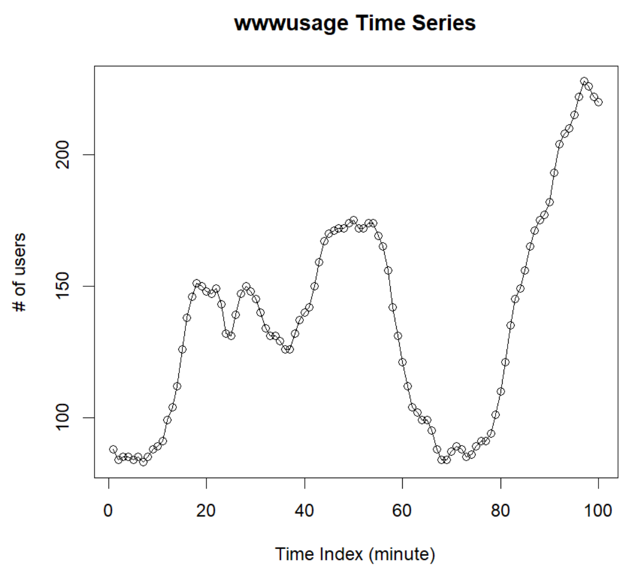

Fig. 1. wwwusage data

II. Fitting an ARIMA model

In order to determine the appropriate parameter values for our ARIMA model, we must first ensure that our time series data is stationary in order to have any meaningful insights from the ACF and PACF plots. This can be easily checked using the Augmented Dickey-Fuller (ADF) test. The Augmented Dickey-Fuller (ADF) test is a type of statistical test called a unit root test. The purpose of this test is to determine whether a time series is stationary or not. The null hypothesis of the ADF test is that the time series is non-stationary. So, if the p-value is less than the significance level (0.05 is commonly used), then we can reject the null hypothesis and infer that the time series is indeed stationary [2].

```
	Augmented Dickey-Fuller Test

data:  X_train
Dickey-Fuller = -2.6278, Lag order = 4,
p-value = 0.3188
alternative hypothesis: stationary
```

In our case, the p-value is 0.3188, which is greater than 0.05. Therefore, we fail to reject the null hypothesis, suggesting that the time series is non-stationary. To convert a non-stationary data into stationary, we can apply first-order differencing to remove temporal dependence (which includes trends and seasonality), and check if the first-order differenced data is stationary. Differencing is performed by subtracting the previous observation from the current observation in the time series. This transformation creates a new time series where the values are the differences between consecutive values. This can easily be done in R using the `diff()` function. The output of this process is shown in Fig. 2.

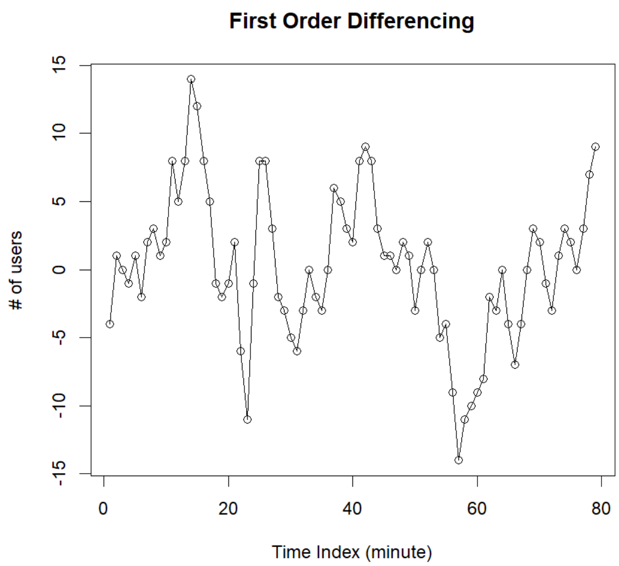

Fig. 2. Times Series after 1st order Differencing

Afterwards, we can apply the ADF test to check for stationarity. The results of which is as follows:

```
Augmented Dickey-Fuller Test

data:  z1
Dickey-Fuller = -2.3855, Lag order = 4,
p-value = 0.418
alternative hypothesis: stationary
```

We can see from the results that the p-value (0.418) is still above 0.05, which means we fail to reject the null hypothesis and conclude that the first-order differenced data is still non-stationary. By applying another round of differencing, we get the second-order differenced data, shown in Fig. 3.

And, applying ADF test again, we can see that the p-value (0.01) is now below 0.05. This means we can reject the null hypothesis of the ADF test and conclude that the 2nd order differenced data is stationary.

```
Augmented Dickey-Fuller Test

data:  z2
Dickey-Fuller = -4.3929, Lag order = 4,
p-value = 0.01
```

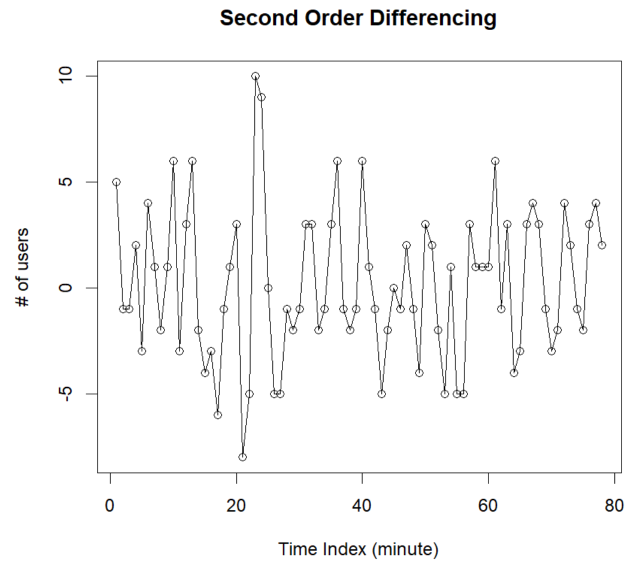

Fig. 3. Times Series after 2nd order Differencing

```
alternative hypothesis: stationary
```

Now that we have a stationary time series, we can apply PACF and ACF (shown in Fig. 4 and 5) to decide the order of the autoregressive (AR) and moving average (MA) terms for our ARIMA model, respectively.

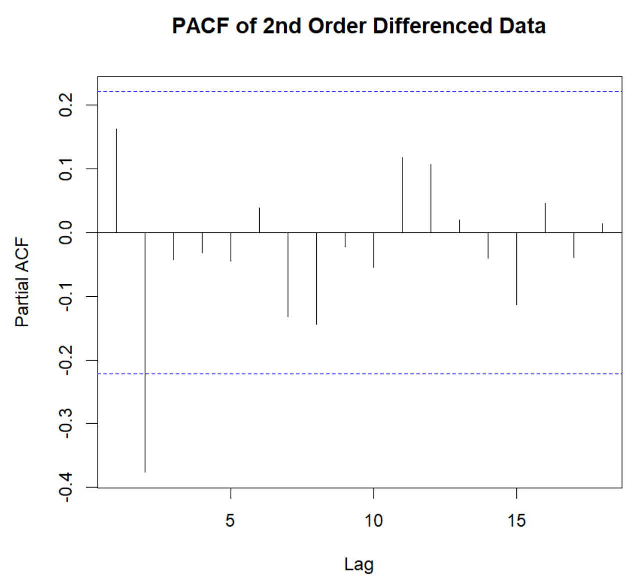

Fig. 4. PACF of the 2nd order Differenced Data

We can observe from the PACF plot a sharp (negative) cut-off at lag equal to 2 before the correlation values fall below the threshold indicated by the broken blue lines. The same is observed from the ACF plot at lag equal to 2. Hence, we can set $p = 2$ and $q = 2$ as the orders of the AR and MA terms

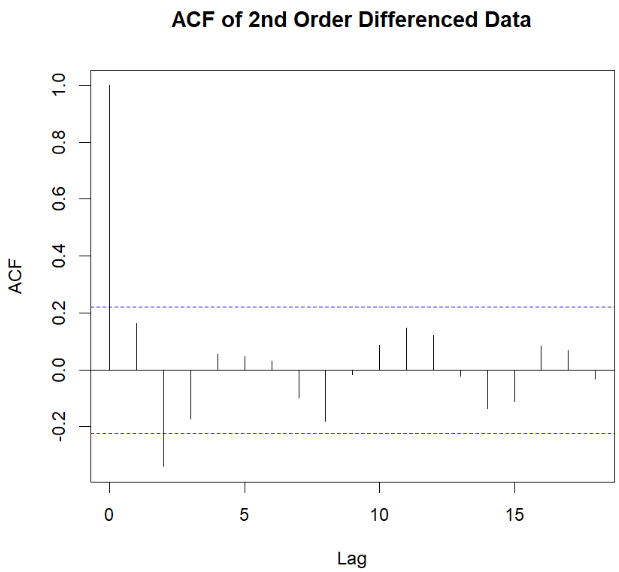

Fig. 5. ACF of the 2nd order Differenced Data

in our ARIMA model. After fitting an ARIMA(2,2,2) model and using the first 80 observations of the wwwusage data as input, we can get a quick forecast of the next 20 observations as shown in Fig. 6.

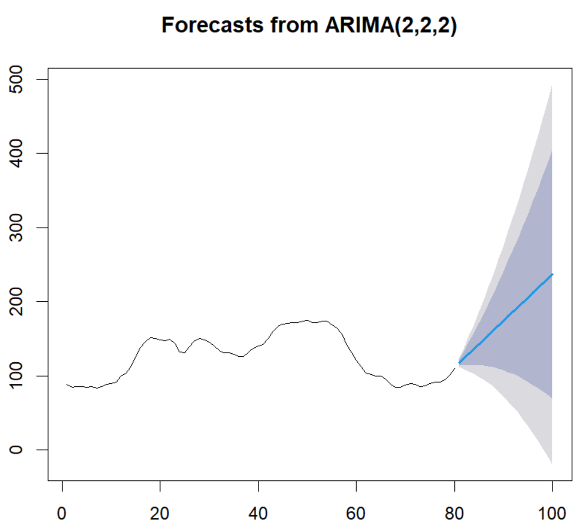

Fig. 6. Forecasts from ARIMA(2,2,2)

One way to check the goodness-of-fit of our ARIMA model is to inspect the *residuals*. The residuals of a time series model are the differences between the observed values and the predicted values from the model. If a model is a good fit for the data, these residuals should ideally resemble white noise. Using the `checkresiduals()` function in R, we get the following outputs shown in Fig. 7 and 8. This functions performs several operations that allows us to inspect for any patterns that may show signs of temporal dependence in the residuals; which implies that we were not able to capture all the meaningful information using our ARIMA model.

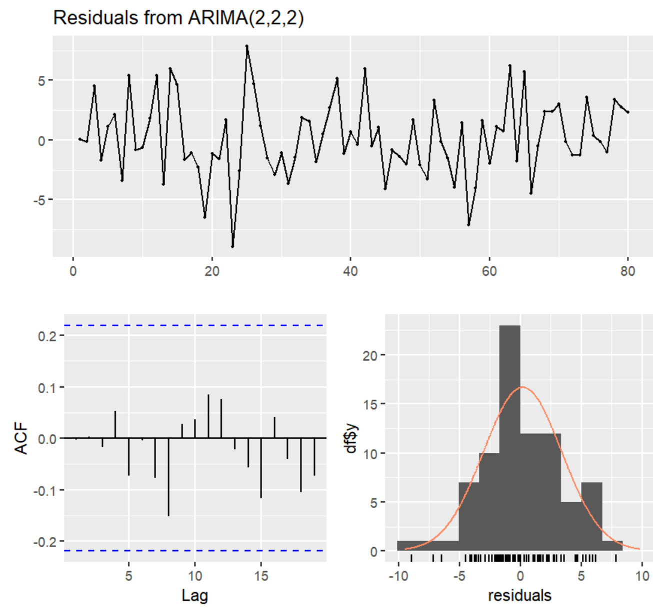

Fig. 7. Residuals from ARIMA(2,2,2)

From the ACF plot in Fig. 7, we can see that there is no significant autocorrelation (indicated by spikes above the threshold other than zero). We can also see from the histogram that the residuals are normally distributed.

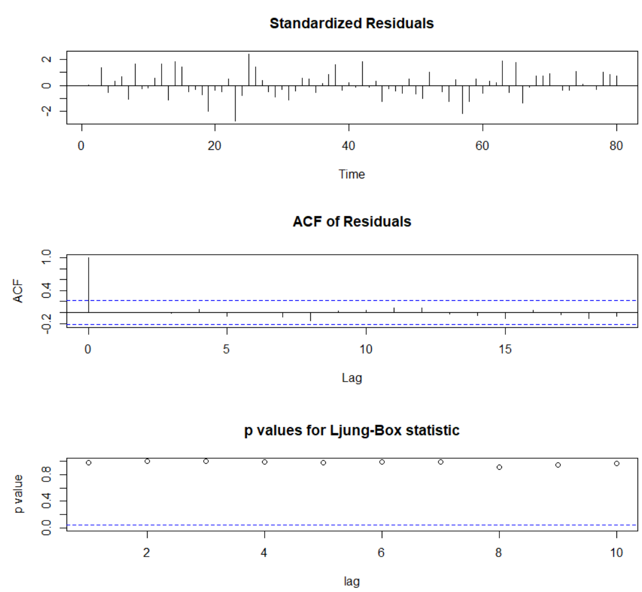

Fig. 8. Ljung-Box Statistic

The `checkresiduals()` function also performs a Ljung-Box statistical test for checking if the residuals resemble

TABLE I
AIC VALUES RESULTING FROM BRUTE-FORCE SEARCH (D=2)

	p=0	p=1	p=2	p=3	p=4	p=5	p=6	p=7	p=8
q=0	421.945	421.787	411.581	413.440	415.363	417.126	418.873	418.802	419.203
q=1	418.317	418.723	413.408	411.590	417.154	418.727	420.501	419.600	421.190
q=2	414.877	**411.511**	415.225	413.537	415.308	417.183	419.172	419.929	423.019
q=3	412.124	413.752	415.742	414.459	415.202	415.034	417.601	418.464	423.810
q=4	413.815	415.749	415.617	416.413	NA	417.010	418.978	419.696	421.311
q=5	415.632	414.421	416.337	415.665	417.516	416.333	416.759	418.472	420.448
q=6	416.358	416.211	418.053	418.305	NA	416.455	418.444	421.234	422.441
q=7	417.509	418.128	417.463	419.493	418.901	418.444	420.445	422.365	421.327
q=8	415.835	417.656	419.393	419.721	419.802	420.444	422.443	424.444	423.190

TABLE II
ARIMA MODEL BENCHMARKS

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
model	**47.41917**	**54.99199**	**47.41917**	**22.81397**	**23.79050**	**7.52685**	**0.87408**
model_brute_force	51.59900	59.48786	51.59900	25.19947	25.91833	8.19032	0.87167
model_auto	55.99344	64.53449	55.99344	27 .46907	28 .11760	8 .88785	0 .87101

white noise. A p-value less than 0.05 indicates that the residuals are NOT white noise [3]. From Fig. 8, we can see that the p-values are clearly above this significant level.

Hence, from the outputs shown in Fig. 7 and 8, we can conclude that the residuals resemble white noise; and therefore our ARIMA(2,2,2) model was able to capture all meaningful information in the time series data it was fitted on.

Plotting the forecast values side-by-side with the actual values, we can see that our forecast is very close to the actual values, indicating a good fit.

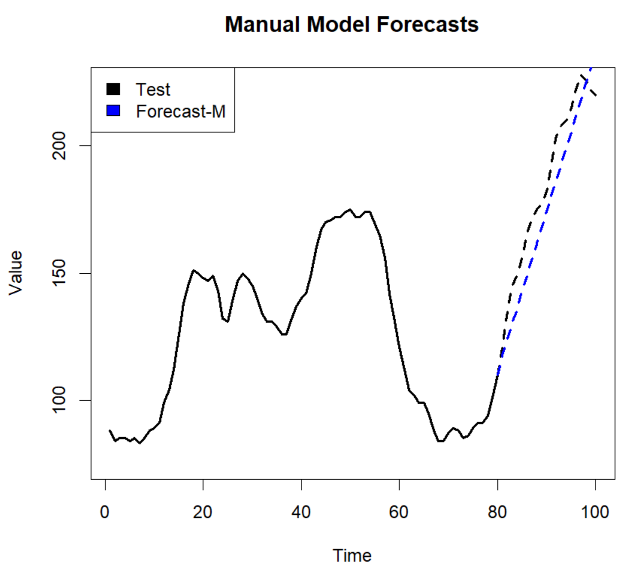

Fig. 9. Forecasted Values vs Actual Values

III. BENCHMARKS

In order to quantitatively evaluate the performance of the fitted model, we will perform simple benchmarks across several standard evaluation metrics; namely, MSE, RMSE, MAE, MPE, MAPE, MASE, and ACF1. The fitted ARIMA(2,2,2) model will be benchmarked against the models generated using a brute-force approach and the auto-arima function.

A. Brute Force Search

To perform brute-force search, we iterate over all the possible combinations of p and q values across a specified range (in this case, from 0 to 8). Here, we set the $d = 2$ since we already know that the time series becomes stationary after 2nd order differencing; hence, we only need to check all 81 combinations of p and q values ranging from 0 to 8. For each p, d = 2, and q combination, we fit an ARIMA model and measure how good the fit is using the Akaika Information Criterion (AIC) metric. The AIC provides a balance between the goodness of fit and the complexity of the model. When comparing multiple models, the one with the lowest AIC is considered the best [4]. The output of this process is presented in Table I.

From the table, we can see that the lowest AIC value of **411.511** is obtained when p = 1 and q = 2, suggesting an ARIMA(1,2,2) model. It is also important to note from the same table, we see that our original ARIMA(2,2,2) model produced an AIC value of **415.225**.

From the brute-force approach, we will therefore fit an ARIMA(1,2,2) model.

B. Auto-Arima

Auto-ARIMA is a function used in time series analysis that automatically identifies the most optimal parameters for an ARIMA model. In order to find the best model, Auto-ARIMA optimizes for a given information criterion (the default is set to AICc or the Corrected Akaike Information Criterion) and returns the ARIMA model which minimizes this value [5].

In order to implement auto-arima in R, we simply need to run the `auto.arima()` function from the forecast package in R with our training series as input. The model selected using the auto-arima method is ARIMA(1,1,1).

C. Benchmark Results

Fig. 10 shows the forecast trends of all 3 models; namely, the ARIMA(2,2,2) model from manual fitting, the ARIMA(1,2,2) model using brute force search based on AIC at d=2, and the ARIMA(1,1,1) model from the auto-arima model using the AICc information criterion. We can see from the figure that the model generated using manual techniques, i.e., identifying stationarity using the ADF test, and selecting the appropriate p and q parameters by inspecting the ACF and PACF plots of the 2nd order differenced data, generated a forecast trend that resembles the testing data the closest.

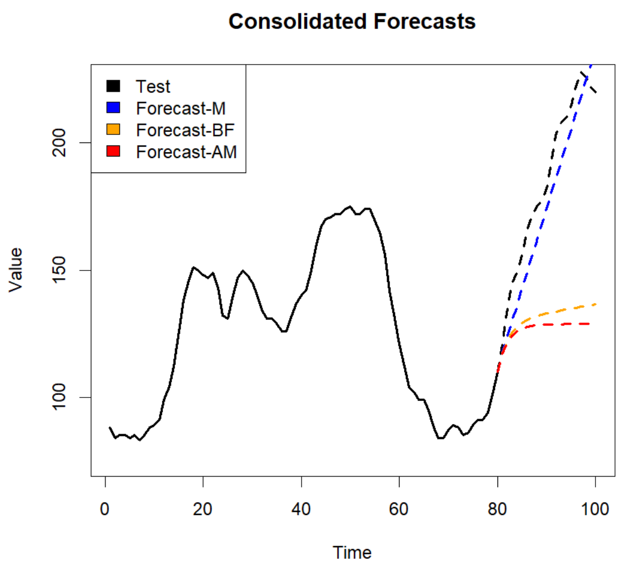

Fig. 10. Benchmark Results. Forecast-M represents the forecasts of the manually fitted model. Forecast-BF represents the forecasts from the model obtained using brute-force search. Forecast-AM represents the forecasts from the model generated using auto-arima

We can further quantify the result by testing the three models across several selected evaluation metrics; namely, MSE, RMSE, MAE, MPE, MAPE, MASE, and ACF1, as shown in Table II. From the table, we can see that the ARIMA(2,2,2) model outperforms the brute-force and auto-arima models across all selected benchmarks. This evaluation provides a robust measure of the predictive performance of each model. The Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) all measure the average magnitude of the errors in a set of predictions, without considering their direction. Meanwhile, the Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Scaled Error (MASE) are scale-independent measures that express the forecast errors as a percentage, providing a relative measure of the prediction accuracy. Lastly, the Autocorrelation Function at lag 1 (ACF1) measures the correlation between the series' observations and the lagged observations.

The ARIMA(2,2,2) model's superior performance across these metrics indicates its ability to capture the underlying dynamics of the time series data. The benchmark results suggest that the manual process of identifying stationarity, and selecting the appropriate p and q parameters, has resulted in a more accurate and reliable model. However, it's important to note that while the ARIMA(2,2,2) model outperforms the other models in this instance, the optimal model can vary depending on the specific characteristics of the time series data.

IV. Conclusion

In conclusion, the ARIMA(2,2,2) model, as determined through manual techniques, has demonstrated the most accurate forecasting ability for the wwwusage dataset, as evidenced by its superior performance across all selected benchmark metrics. This underscores the importance of careful model selection and validation in time series forecasting. Future work could explore the potential benefits of incorporating additional explanatory variables or applying more complex models, such as SARIMA or ARIMAX, to further enhance forecasting accuracy.

References

[1] A. Hayes, "Autoregressive integrated moving average (ARIMA) prediction model," Investopedia, https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp (accessed Mar. 9, 2024).
[2] S. Prabhakaran, "Augmented Dickey Fuller Test (ADF Test) – Must Read Guide," Machine Learning Plus, 2023. [Online].
[3] Z. Statology, "Ljung-Box Test: Definition + Example," Statology, 2020. [Online]
[4] A. Zajic, "What Is Akaike Information Criterion (AIC)?," Built In, 29 Nov. 2022. [Online]
[5] S. Prabhakaran, "ARIMA Model – Complete Guide to Time Series Forecasting in Python," Machine Learning Plus, 2023. [Online].