AI6123: REPORT For Project 1

Names: Joseph UWINEZA
Matric No: G2303477F
Program: MSAI

NTU, March 2024

PROJECT 1

Abstract

Time series analysis is a statistical technique used to analyze patterns in data collected over time. It involves studying the sequential order of data points to understand and forecast future trends or behaviors. This field encompasses various methods for modeling, forecasting, and interpreting time-dependent data, often used in fields like economics, finance, signal processing, and environmental science.

In this report, I am going to detail how to appropriately fit the ARIMA model, and make some diagnostics for its adequacy, by Using the wwwusage time dataset which contains non-blank 100 observations. I also appended the full lines of codes used to implement this.

1 Introduction

In the realm of time series analysis, it is commonly understood that the inherent nature of most datasets is non-stationary. Stationarity is a pivotal characteristic of time series data that necessitates the data's statistical properties, such as mean, variance, to remain constant over time. The ability to harness these invariant properties is crucial for developing reliable forecasting models.

This Report commences with a meticulous statistical and graphical examination of the dataset to ascertain its underlying structure. Subsequently, the method of differencing will be employed to transform the non-stationary data into a stationary form, thus rendering it amenable for subsequent analysis. Upon achieving stationarity, the data will be utilized to calibrate Autoregressive Integrated Moving Average (ARIMA) models.

Then I will proceed to assess the adequacy and the predictive prowess of these models, and then diagnostic criteria, namely AIC and BIC, RMSE, ME (to mention few) will be rigorously applied. These metrics will guide the selection of an optimal model that balances model complexity with the goodness of fit, thereby ensuring the robustness of the forecasts derived from this empirical investigation.

2 Preliminary Analysis

2.1 Statistical analysis

Below is the figure 1 which shows the data contained in the dataset given with 100 observations, followed by the dataset summary and then the Mean and variance.

```
  [1]  88  84  85  85  84  85  83  85  88  89  91  99 104 112 126 138 146 151
 [19] 150 148 147 149 143 132 131 139 147 150 148 145 140 134 131 131 129 126
 [37] 126 132 137 140 142 150 159 167 170 171 172 172 174 175 172 172 174 174
 [55] 169 165 156 142 131 121 112 104 102  99  99  95  88  84  84  87  89  88
 [73]  85  86  89  91  91  94 101 110 121 135 145 149 156 165 171 175 177 182
 [91] 193 204 208 210 215 222 228 226 222 220
[1] "************ SUMMARY ************"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   83.0    99.0   138.5   137.1   167.5   228.0
[1] "************ Mean and Variance ************"
[1] " Mean: 137.08 Variance: 1599.95313131313"
```

Figure 1: Show the various wwwusage dataset information.

The time series data has a mean of **137.08**, suggesting the average level around which the values oscillate. With a variance of **1599.953**, the data points exhibit considerable spread, indicating notable fluctuations over time. With such a summary, our data implies to be non-stationary data.

2.2 Visualization Analysis

2.2.1 Initial graph

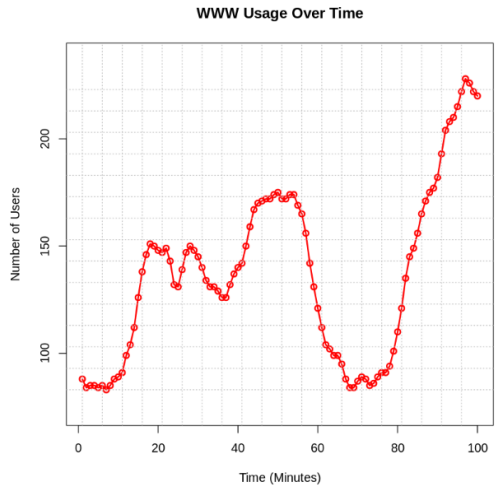

Figure 2: **Shows a time plot of initial data. The number of observations is 100.**

As observed from 2, The Mean of the time series exhibits notable variations, as indicated by the multiple peaks and troughs throughout the graph. Additionally, the fluctuation amplitude changes, reflecting a non-uniform variance. These factors suggest that the data lacks stationarity.

2.2.2 Ploting ACF and PACF

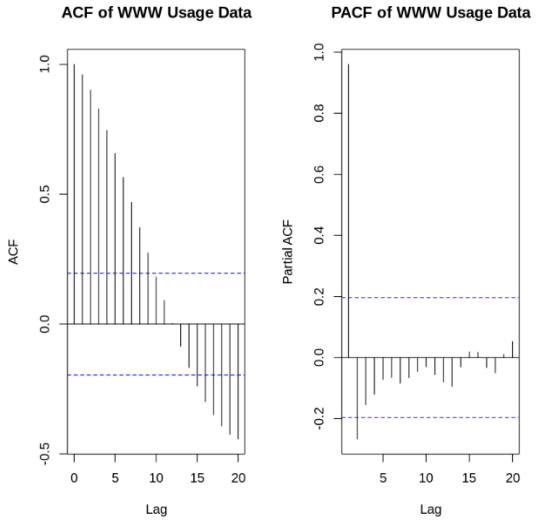

Figure 3: Shows ACF (left) and PACF (Right) of the data

The left of figure 3 shows the ACF for WWW Usage Data, where numerous lags dies down slowly and extend beyond the confidence intervals indicated by blue dotted lines, suggesting notable autocorrelation at those points which implies again that it is not stationary data. The right of figure 3 shows PACF of the non-stationary data behaves orbitally and therefore can not be used to infer the stationarity of the data.

3 One-time Differencing (d=1)

Differencing data is the method used to make that stationary. Stationarity means that the statistical properties of the series—such as its mean, variance, and autocorrelation—are constant over time. Non-stationary data often contain trends or seasonality, which can bias the analysis and lead to unreliable forecasts. Differencing removes these elements, stabilizing the mean of the time series by subtracting the previous observation from the current observation, thus helping to meet the assumption of stationarity for subsequent analyses. $Z_t = X_t - X_{t-1}$

3.1 A time Plotting after d=1

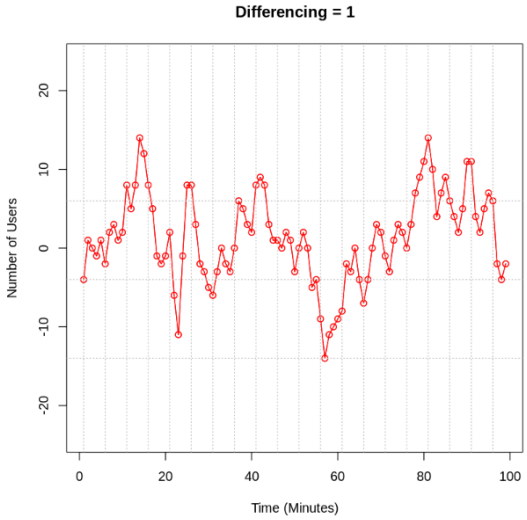

Figure 4: Shows a time plot after one-time differencing (d=1). The number of observations is now 99.

3.2 coefficient checking

We can use ar.yw() Yule-Walker function to estimate the coefficient of AR(p) model.

```
Call:
ar.yw.default(x = differenced_data, max = 5)

Coefficients:
      1        2        3
 1.1060  -0.5957   0.3029

Order selected 3  sigma^2 estimated as  10.32
```

Figure 5: shows that AR has 3 coefficient

3.3 ACF and PACF after d=1

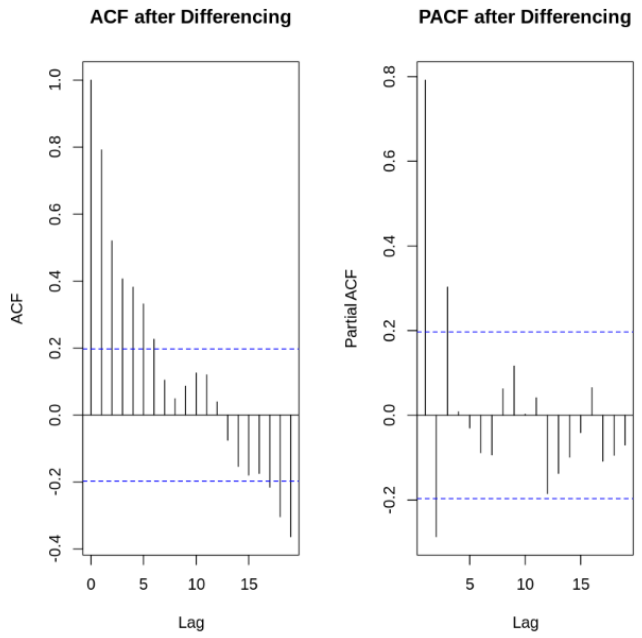

Figure 6: shows ACF (left) dies down slowly and PACF (right) cut-off at time lag=3.

The left Figure 6 reveals the ACF after differencing. it indicates that it cut-off on time lag six. which suggests the MA(6) model. The significant correlations at the initial lags have been reduced, with most of the autocorrelations now falling within the confidence interval.

The right of figure 6 displays the PACF after differencing indicates that it cut-off on time lag three, which suggests AR(3) model. Similar to the ACF plot, the significant partial autocorrelations have largely been removed, with only a few lags showing significant correlations. This suggests the AR(3) model.

3.4 Fitting model

Given the prior analysis, I am going to fit ARIMA(3,1,0), ARIMA(0,1,6) and ARIMA(3,1,6) models, and I will verify the standardized and ACF residuals and Ljung-Box Test to confirm it's adequacy.

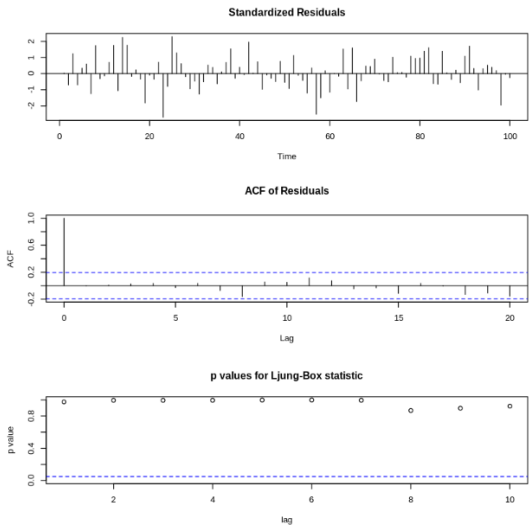

Figure 7: Shows the Standardized, ACF Residuals and Ljung-Box Test of ARIMA(3,1,0) model. The diagnostic plots for the ARIMA(3,1,0) model show standardized residuals that resemble white noise, indicating a good model fit. The ACF of the residuals lies within confidence bounds, suggesting no significant autocorrelation and supporting the model's adequacy. The Ljung-Box test results confirm the residuals are random and independent, further validating the model's suitability for the data.

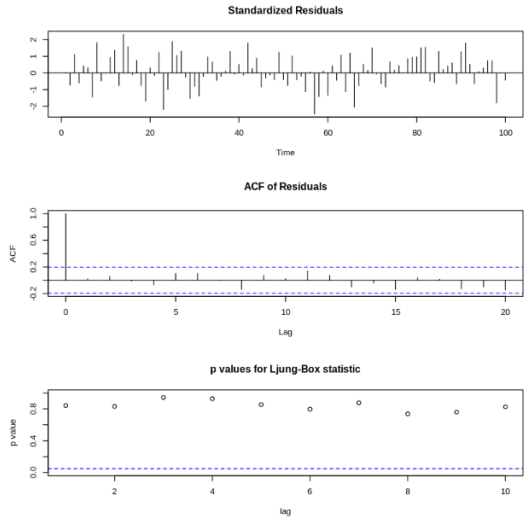

Figure 8: The standardized residuals plot indicates that the residuals are randomly distributed around zero with no obvious patterns, suggesting a good fit. The ACF plot of residuals shows all values within the confidence bounds, implying no significant autocorrelation. The Ljung-Box test p-values are well above the significance level across all lags, indicating the residuals are independent and the model is adequate.

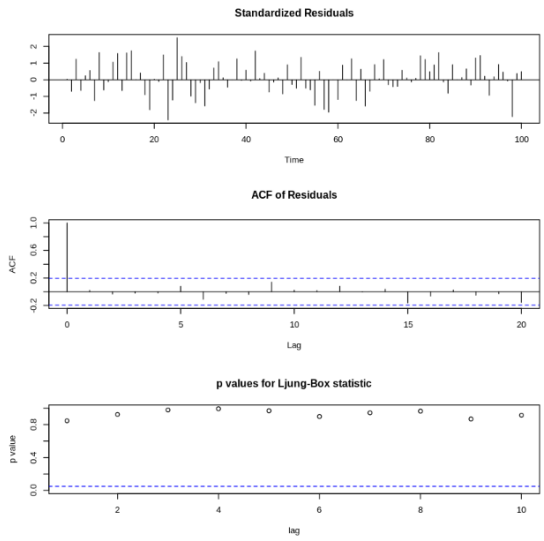

Figure 9: The displayed diagnostics for an ARIMA(3,1,6) model show standardized residuals fluctuating randomly around zero, suggesting a good model fit. The ACF of residuals lies within the confidence intervals, indicating no significant autocorrelation post-modeling. Lastly, the Ljung-Box test p-values are consistently high, reinforcing the absence of autocorrelation in the residuals and suggesting the model's adequacy.

3.5 Fitted data

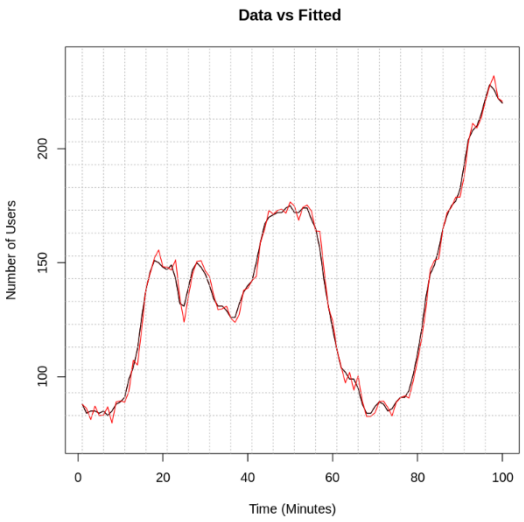

Figure 10: The image shows a time series plot comparing observed data (black line) to fitted values from a statistical ARIMA(3,1,0) (red line) over time, measured in minutes. The fitted values closely track the observed data, capturing the overall trends and fluctuations. The model appears to fit particularly well during the latter part of the series, where there's a sharp increase in the number of users. I simply choose ARIMA(3,1,0) among others for its small AIC values, refers to table 1.

4 Second Differencing (d=2)

We can try out with the second order time differencing:

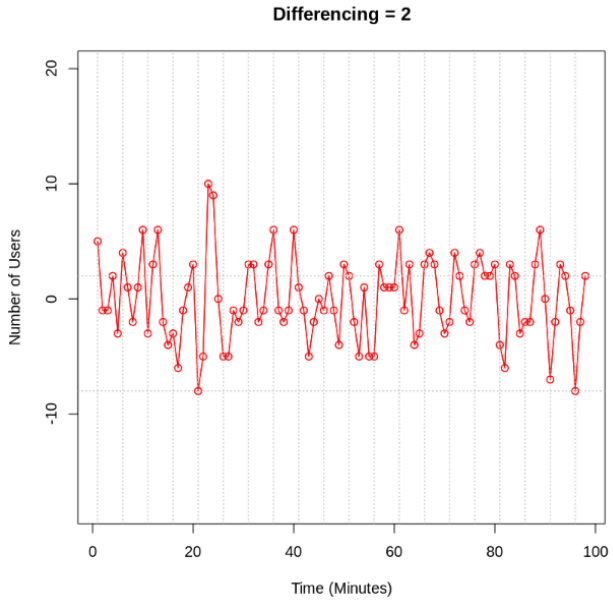

Figure 11: Show a time plot after applying second-order differencing. The number of observations is now 98. Furthermore, the plot exhibits the fluctuations of the differenced data around zero, with no apparent trend or seasonal patterns. The variability seems consistent over time, suggesting that the differencing have helped in stabilizing the mean and detrending the data.

4.1 Checking coefficient

We can use ar.yw() Yule-Walker function to estimate the coefficient of AR(p) model.

```
Call:
ar.yw.default(x = differenced_data_2, max = 5)

Coefficients:
      1        2
 0.2489  -0.4341

Order selected 2  sigma^2 estimated as  10.56
```

Figure 12: shows that AR has 2 coefficient

4.2 ACF and PACF after d=2

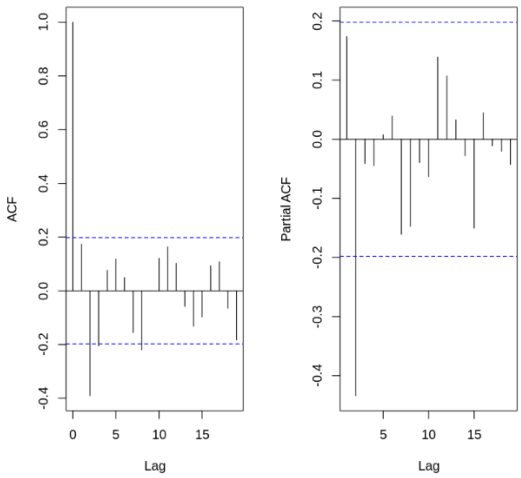

Figure 13: shows ACF (left) dies down at lag 2, suggesting MA(2) and PACF (right) cut-off at time lag 2, suggesting AR(2).

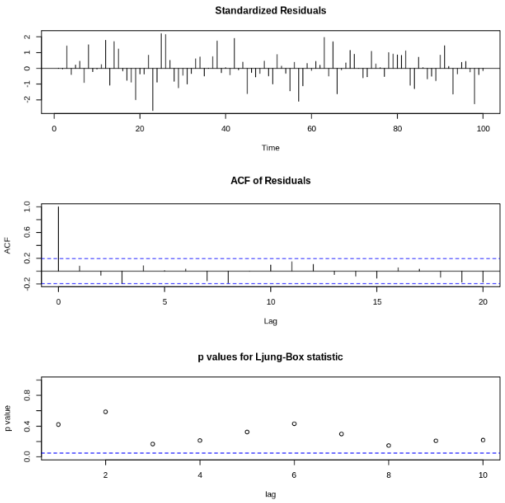

Figure 15: The image shows diagnostic plots for an ARIMA model with standardized residuals appearing as random fluctuations around zero, indicative of a good model fit. The ACF plot of the residuals reveals no significant autocorrelations, as all fall within the confidence interval, implying an appropriate model selection. The Ljung-Box test p-values are well above the typical significance level, confirming the absence of autocorrelation in the residuals and suggesting the model's residuals are independently distributed

4.3 Checking model adquecy

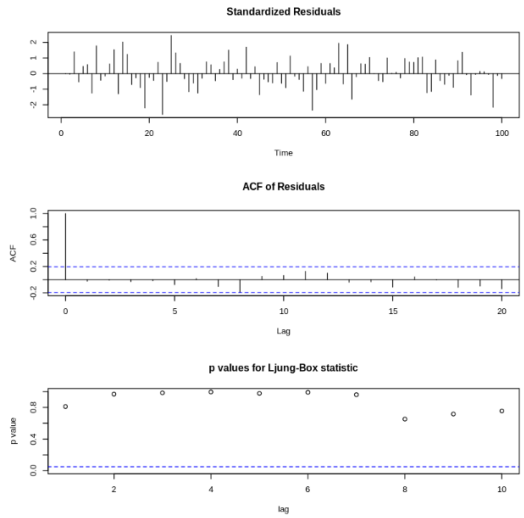

Figure 14: contains diagnostic plots for an ARIMA(2,2,0) model, which include the standardized residuals, the ACF of residuals, and the p-values for the Ljung-Box test. The ARIMA(2,2,0) model's residuals resemble white noise, with no apparent trends or patterns, and the ACF within confidence intervals indicates a good fit with no significant autocorrelation. Ljung-Box test results further confirm the lack of autocorrelation, implying the model's assumptions are well-suited for the data

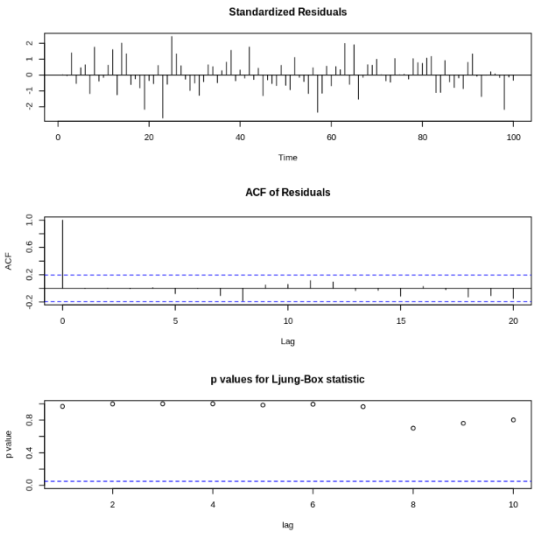

Figure 16: The image depicts diagnostic checks for an ARIMA model, showing standardized residuals that are distributed randomly around the zero line, indicating no obvious model inadequacies. The ACF of residuals plot demonstrates all values within the confidence bounds, suggesting a lack of autocorrelation in the residuals. The Ljung-Box test yields p-values above the common significance threshold, supporting the hypothesis that the residuals are independently distributed, which affirms the model's fit

4.4 Fitted values plot

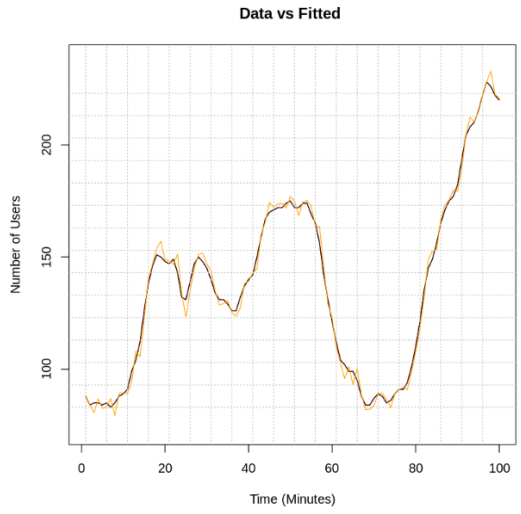

Figure 17: shows a time series plot comparing observed data (black line) to fitted values from a statistical ARIMA(2,2,0) model (orange line) over time, measured in minutes. The fitted values closely track the observed data, capturing the overall trends and fluctuations. The model appears to fit particularly well during the latter part of the series, where there's a sharp increase in the number of users. I simply choose ARIMA(2,2,0) as it has small AIC values, refer to table 1

5 Model performance

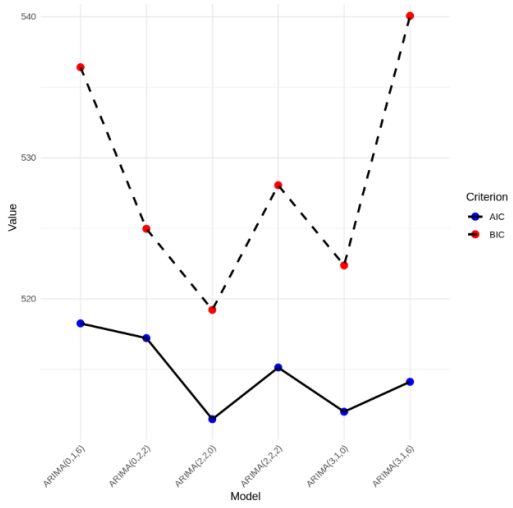

Figure 18: presents a comparison of different ARIMA(p,d,q) models based on AIC and BIC metrics.

Figure 18 Shows that ARIMA(2,2,0) outperforms other models in terms of minimum of both AIC and BIC.

No	Model	AIC	BIC
1	ARIMA(3,1,0)	511.99	522.37
2	ARIMA(0,1,6)	518.25	536.42
3	ARIMA(3,1,6)	514.12	540.07
4	ARIMA(2,2,0)	**511.46**	**519.22**
5	ARIMA(0,2,2)	517.21	524.96
6	ARIMA(2,2,2)	515.13	528.05

Table 1: Shows the performance metric between different models. It also shows that **ARIMA(2,2,0)** has the minimum values of AIC and BIC compared to the other models, which suggests the best model.

However, our analysis must not stop at this point, since an effectively fitting model is characterized not merely by its statistical scores but by its capacity to deliver precise forecasts with the least amount of error. Therefore, we intend to conduct a more thorough examination of the model's suitability for making accurate future forecasts.

6 Forecast

Since according to the *tsdiag* shows all these six (ARIMA(3,1,0), ARIMA(0,1,6), ARIMA(3,1,6), ARIMA(2,2,0), ARIMA(0,2,2), and ARIMA(2,2,2)) models are adequate we can plot it's forecasting for better analysis.

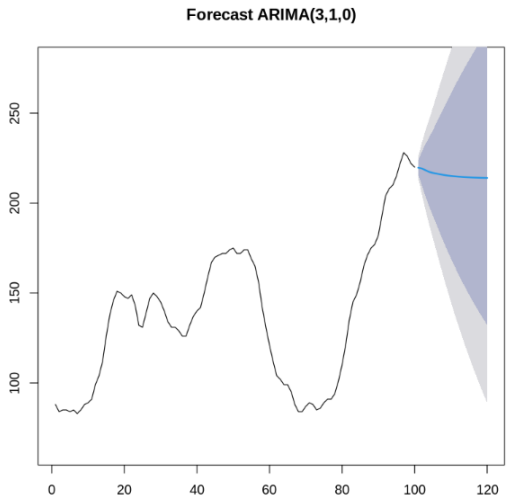

Figure 19

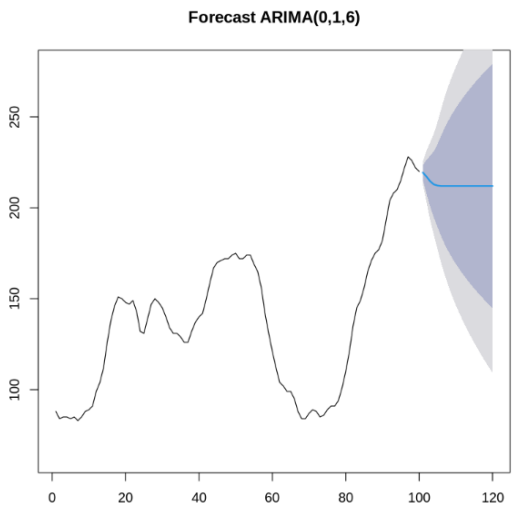

Figure 20

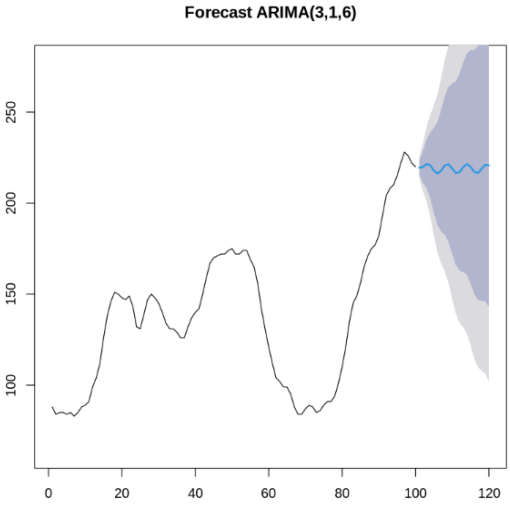

Figure 21

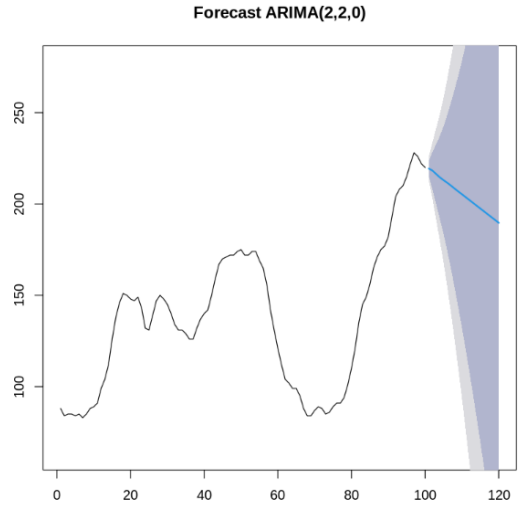

Figure 22

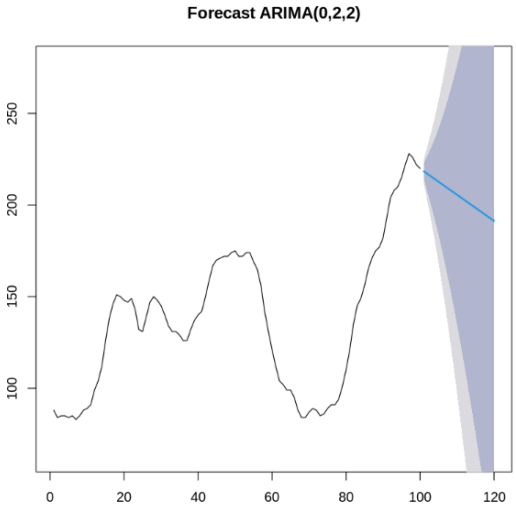

Figure 23

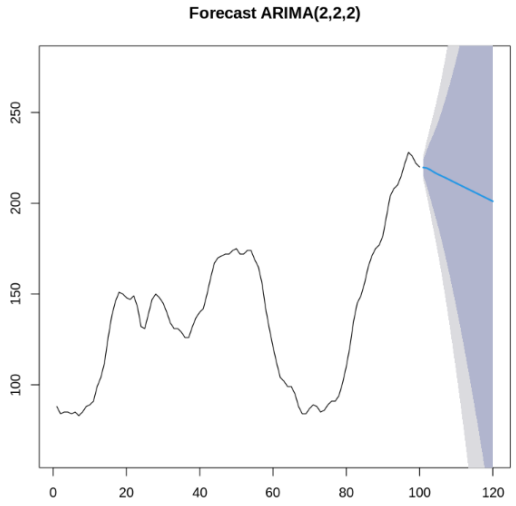

Figure 24

These graphs are used to visualize the model's ability to forecast future values based on historical data. The confidence intervals provide a range where the actual values are expected to fall, with a certain level of confidence.

A model with narrower confidence intervals **ARIMA(3,1,6)**, ARIMA(0,1,6) and ARIMA(3,1,0) respectively are closely to follow the historical data trends indicates a better fit, as it suggests higher precision and more confidence in the predictions.

Conversely, a model with wider confidence intervals ARIMA(0,2,2), ARIMA(2,2,2) and ARIMA(2,2,0) respectively suggest less certainty in its forecasts, potentially indicating a less accurate model.

No	**Model**	**Set**	**ME**	**RMSE**	**MAE**	**MPE**	**MAPE**	**MASE**
1	ARIMA(3,1,0)	Training	0.230588	3.044632	2.367157	0.2748377	1.890528	0.5230995
	ARIMA(3,1,0)	Test	-2.168862	12.071916	10.086919	-1.2910728	4.816833	2.2290289
2	ARIMA(0,1,6)	Training	0.3060598	3.043862	2.460036	0.2847553	1.958335	0.543624
	ARIMA(0,1,6)	Test	1.1818247	12.831987	10.984344	0.2512384	5.183404	2.427344
3	ARIMA(3,1,6)	Training	0.2520268	2.820084	2.223619	0.2579421	1.758860	0.4913802
	ARIMA(3,1,6)	Test	**-4.6524481**	**11.931761**	**9.240339**	**-2.4264750**	**4.461395**	**2.0419499**
4	ARIMA(2,2,0)	Training	0.02797758	3.150308	2.511921	0.2062350	1.994727	0.5550897
	ARIMA(2,2,0)	Test	2.39255695	**14.750798**	13.118737	0.7787425	6.155561	2.8990065
5	ARIMA(0,2,2)	Training	0.03078476	3.246492	2.540705	0.2182086	1.986679	0.5614504
	ARIMA(0,2,2)	Test	2.81446793	14.421516	12.824363	0.9845315	6.007490	2.8339552
6	ARIMA(2,2,2)	Training	0.03788898	3.144716	2.494388	0.2237268	1.977332	0.5512153
	ARIMA(2,2,2)	Test	-0.78477429	13.008919	11.301668	-0.6693249	5.360484	2.4974669

Table 2: Based on the highlighted metrics in the table, the **ARIMA(3,1,6)** model has the lowest RMSE and MAE values in the test set, indicating that it performs best in terms of predictive accuracy among the models listed when considering the test set. In contrast, the model with the highlighted highest RMSE in the test set is the **ARIMA(2,2,0)**, suggesting that it performs the worst in terms of predictive accuracy on the test data. The selection of the "best" model should ideally be based on its performance on out-of-sample data (the test set), which better represents the model's ability to predict new, unseen data. Therefore, considering the given metrics, **ARIMA(3,1,6)** would be considered the best model and **ARIMA(2,2,0)** the worst model.

A Appendix

Below are the lines of codes used to implement, project one.

```
install.packages("forecast")
#Loading the dataset
data <- scan("/content/dataset.txt", skip = 1)
print(data) #Display the content of the dataset.
print("************* SUMMARY ************")
summary(data)
print("************* Mean and Variance ************")
print(paste(" Mean:", mean(data), "Variance:", var(data)))

# Basic statistical exploration
minimum_value <- min(data)
maximum_value <- max(data)
average_value <- mean(data)
# Plotting the initial time series
plot(data, type = "o", main = "WWW Usage Over Time", xlab = "Time (Minutes)", ylab = " Number of Users", ylim = c(minimum_value - 10, maximum_value + 10), col = "red")
lines(data, type = "o", col = "red", lwd = 2)
# Adding a grid
abline(h = seq(from = floor(minimum_value), to = ceiling(maximum_value), by = 10), col = "gray", lty = "dotted")
abline(v = seq(from = 1, to = length(data), by = 5), col = "gray", lty = "dotted")

par(mfrow = c(1, 2))
#Plot for Autocorrelation
acf(data, main = "ACF of WWW Usage Data")
#plot for Partial Autocorrelation
pacf(data, main = "PACF of WWW Usage Data")
par(mfrow = c(1, 1))


# Differencing the series for stationarity
differenced_data <- diff(data)
par(mfrow = c(1, 2))
# ACF and PACF After Differencing Data
acf(differenced_data, main = "ACF after Differencing")
pacf(differenced_data, main = "PACF after Differencing")
par(mfrow = c(1, 1))

# Basic statistical exploration
minimum_value <- min(differenced_data)
maximum_value <- max(differenced_data)
average_value <- mean(differenced_data)
# Plotting the initial time series
plot(differenced_data, type = "o", main = " Differencing = 1 ", xlab = "Time (Minutes)", ylab = "Number of Users", ylim = c( minimum_value - 10, maximum_value + 10), col = "red")
lines(differenced_data, type = "o", col = "red", lwd = 1)
# Adding a grid
abline(h = seq(from = floor(minimum_value), to = ceiling(maximum_value), by = 10), col = "gray", lty = "dotted")
abline(v = seq(from = 1, to = length(data), by = 5), col = "gray", lty = "dotted")

# Fit specific ARIMA models based on prior analysis
arima_model_310 <- arima(data, order = c(3, 1, 0))
print("************* arima_model_310 ************")
tsdiag(arima_model_310)
arima_model_310
arima_model_016 <- arima(data, order = c(0, 1, 6))
print("************* arima_model_016 ************")
tsdiag(arima_model_016)
arima_model_016
arima_model_316 <- arima(data, order = c(3, 1, 6))
print("************* arima_model_316 ************")
tsdiag(arima_model_316)
arima_model_316

# Basic statistical exploration
minimum_value <- min(data)
maximum_value <- max(data)
average_value <- mean(data)
# Plotting the initial time series
plot(data, type = "l", main = " Data vs Fitted ", xlab = "Time (Minutes)", ylab = " Number of Users", ylim = c(minimum_value - 10, maximum_value + 10), col = "red")
lines(data, type = "l", col = "black", lwd = 1)
lines(data-arima_model_310$residuals, type = "l", col = "red", lwd = 1)
# Adding a grid
abline(h = seq(from = floor(minimum_value), to = ceiling(maximum_value), by = 10), col = "gray", lty = "dotted")
abline(v = seq(from = 1, to = length(data), by = 5), col = "gray", lty = "dotted")

#Second order differencing
differenced_data_2 = diff(data, differences = 2)
#checking coefficient
ar.yw(differenced_data_2, max=5)

# Basic statistical exploration
minimum_value <- min(differenced_data_2)
maximum_value <- max(differenced_data_2)
average_value <- mean(differenced_data_2)
# Plotting the initial time series
plot(differenced_data_2, type = "o", main = " Differencing = 2 ", xlab = "Time (Minutes)", ylab = "Number of Users", ylim = c( minimum_value - 10, maximum_value + 10), col = "red")
lines(differenced_data_2, type = "o", col = "red", lwd = 1)
# Adding a grid
abline(h = seq(from = floor(minimum_value), to = ceiling(maximum_value), by = 10), col = "gray", lty = "dotted")
abline(v = seq(from = 1, to = length(data), by = 5), col = "gray", lty = "dotted")

#plotting ACF and PACF of d=2
par(mfrow = c(1, 2))
acf(differenced_data_2)
pacf(differenced_data_2)
par(mfrow = c(1, 1))

arima_model_220 <- arima(data, order = c(2, 2, 0))
```

```r
print("************* arima_model_220 ************")
tsdiag(arima_model_220)
arima_model_220
arima_model_022 <- arima(data, order = c(0, 2, 2))
print("************* arima_model_220 ************")
tsdiag(arima_model_022)
arima_model_022
arima_model_222 <- arima(data, order = c(2, 2, 2))
print("************* arima_model_222 ************")
tsdiag(arima_model_222)
arima_model_222

# Basic statistical exploration
minimum_value <- min(data)
maximum_value <- max(data)
average_value <- mean(data)
# Plotting the initial time series
plot(data, type = "l", main = " Data vs Fitted ", xlab = "Time (Minutes)", ylab = "Number of Users", ylim = c(minimum_value - 10, maximum_value + 10), col = "red")
lines(data, type = "l", col = "black", lwd = 1)
lines(data-arima_model_220$residuals, type = "l", col = "orange", lwd = 1)
# Adding a grid
abline(h = seq(from = floor(minimum_value), to = ceiling(maximum_value), by = 10), col = "gray", lty = "dotted")
abline(v = seq(from = 1, to = length(data), by = 5), col = "gray", lty = "dotted")

# AIC and BIC comparison
aic_values <- c(AIC(arima_model_310), AIC(arima_model_016), AIC(arima_model_316), AIC(arima_model_220), AIC(arima_model_022), AIC(arima_model_222))
bic_values <- c(BIC(arima_model_310), BIC(arima_model_016), BIC(arima_model_316), BIC(arima_model_220), BIC(arima_model_022), BIC(arima_model_222))


# Create a data frame with AIC and BIC values
model_comparison <- data.frame(Model = c("ARIMA(3,1,0)", "ARIMA(0,1,6)", "ARIMA(3,1,6)" ,"ARIMA(2,2,0)", "ARIMA(0,2,2)","ARIMA(2,2,2)"),
                               AIC = aic_values,
                               BIC = bic_values)

# Plotting
library(ggplot2)

ggplot(model_comparison, aes(x = Model)) +
  geom_point(aes(y = AIC, color = "AIC"), position = position_dodge(width = 0.3), size = 3) +
  geom_point(aes(y = BIC, color = "BIC"), position = position_dodge(width = 0.3), size = 3) +
  geom_line(aes(y = AIC, group = 1, linetype = "AIC"), position = position_dodge(width = 0.3), size = 1) +
  geom_line(aes(y = BIC, group = 1, linetype = "BIC"), position = position_dodge(width = 0.3), size = 1) +
  scale_color_manual(values = c("AIC" = "blue", "BIC" = "red")) +
  scale_linetype_manual(values = c("AIC" = "solid", "BIC" = "dashed")) +
  labs(x = "Model", y = "Value", color = "Criterion", linetype = "Criterion") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))



# Forecasting and plotting future values
forecast_310 <- forecast(arima_model_310, h = 20)
forecast_016 <- forecast(arima_model_016, h = 20)
forecast_316 <- forecast(arima_model_316, h = 20)
forecast_220 <- forecast(arima_model_220, h = 20)
forecast_022 <- forecast(arima_model_022, h = 20)
forecast_222 <- forecast(arima_model_222, h = 20)

#Plotting the forecast
plot(forecast_310, main = "Forecast ARIMA (3,1,0)", ylim = c(min(data) - 20, max(data) + 50))
plot(forecast_016, main = "Forecast ARIMA (0,1,6)", ylim = c(min(data) - 20, max(data) + 50))
plot(forecast_316, main = "Forecast ARIMA (3,1,6)", ylim = c(min(data) - 20, max(data) + 50))
plot(forecast_220, main = "Forecast ARIMA (2,2,0)", ylim = c(min(data) - 20, max(data) + 50))
plot(forecast_022, main = "Forecast ARIMA (0,2,2)", ylim = c(min(data) - 20, max(data) + 50))
plot(forecast_222, main = "Forecast ARIMA (2,2,2)", ylim = c(min(data) - 20, max(data) + 50))

# Model accuracy assessment on a test set
test_set <- window(data, start = length(data) - 9)
accuracy(forecast_310, test_set)
accuracy(forecast_016, test_set)
accuracy(forecast_316, test_set)
accuracy(forecast_220, test_set)
accuracy(forecast_022, test_set)
accuracy(forecast_222, test_set)
```

Source Code 1: Full codes implementation