Time Series Analysis and Forecasting

Induni Sandapiumi Nawarathna Pitiyage - 906451
University of Milan Bicocca, CdLM Data Science
A.Y. 2023/2024

---

1. Introduction

This project aims to apply linear and machine learning models to the given time series to compare their predictive performance and to provide a forecast for all days from 2015-04-01 to 2015-11-07. The models that have been incorporated in this project are: ARIMA, UCM and machine learning models.

2. Dataset and Data Pre-processing

2.1 Dataset

The provided dataset for the analysis contains total 3009 entries with 3 columns. The data have been recorded from 2007-01-04 to 2015-03-31 along with the day of the week in textual format and the final column named as, '*avg_days*' which represent as a floating-point number on the average number of days needed to close the requests that were closed that day.

2.2 Data Pre-processing

The provided dataset is composed with missing values only for '*avg_days*' attribute. The total number of missing entries for '*avg_days*' attribute is 202 and among these missing entries majority of the missing values are observed on Sundays. Since the number of missing values are comparatively low with respect to the original data, forward fill imputation (is also known as Last Observation Carried Forward or LOCF) technique is used to handle missing data in time series. The *figure 1*, shows the original time series from 2007-01-04 to 2015-03-31 with forward fill imputation.
To focus on the most relevant and stable periods and to improve model performance, original time series is trimmed. As depicted in *figure 1*, early transition periods (from 2007-01-04 to 2008-12-31) in the time series does not reflect with the latter part of the time series. Therefore, the original time series is trimmed from 2007-01-04 to 2008-12-31.

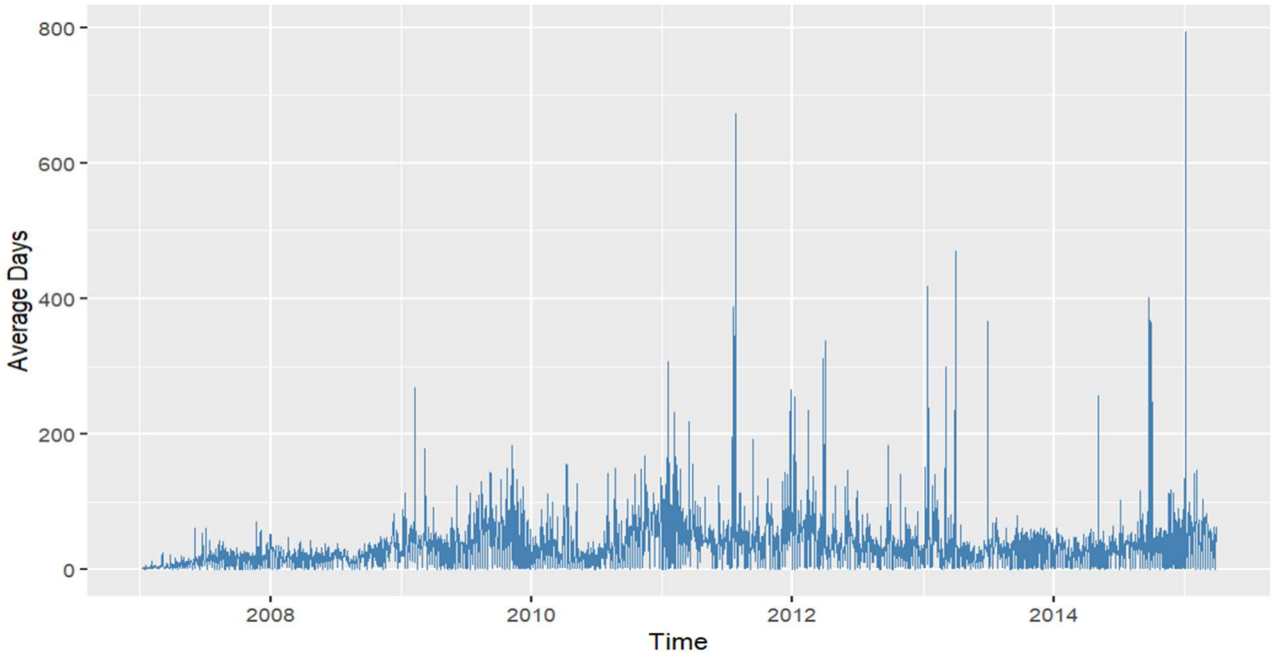

Figure 1: Original time series with Forward Fill (LOCF) Imputation.

The trimmed dataset is partitioned into 2 sets, where initial partition from 2009-01-01 to 2014-03-31 is composed with the data required for model training and the other partition from 2014-04-01 to 2015-03-31 for testing purposes. *Figure 2*, shows the partition of data into training and testing purposes.

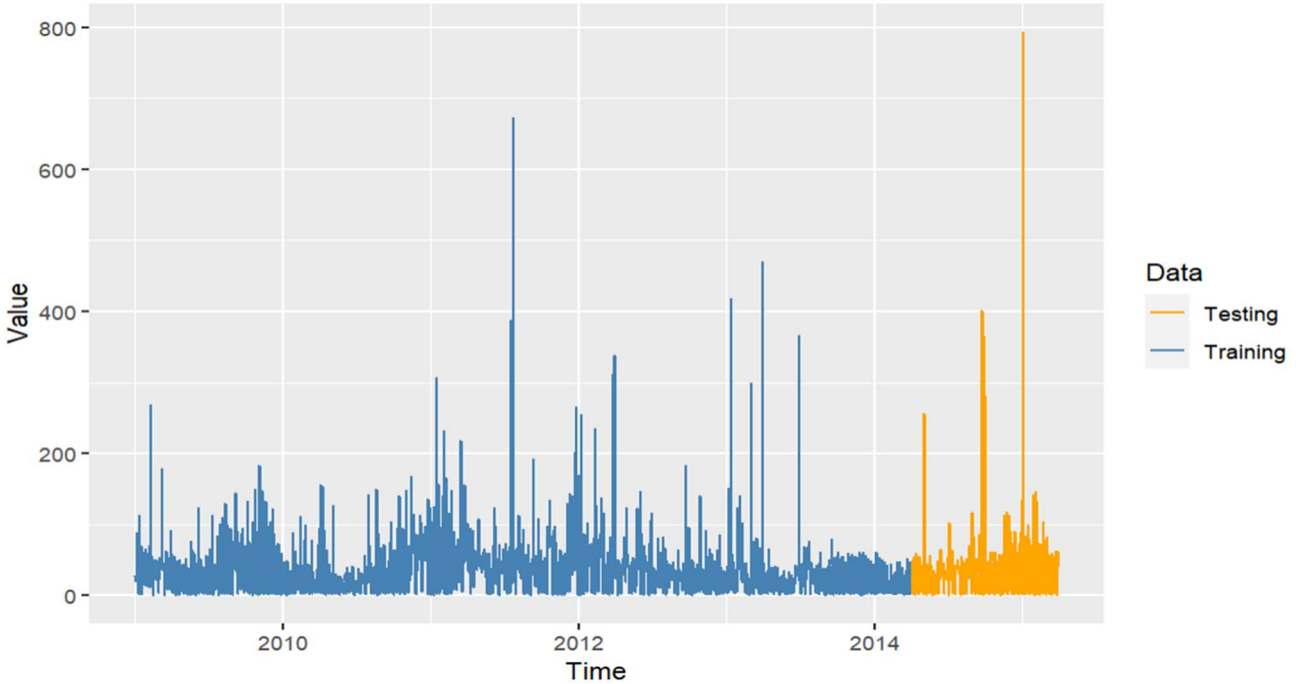

Figure 2: Trimmed time series with training and testing Data

As shown in the *figure 2*, both training and test data are composed with outliers. To identify the outliers exists in both datasets, the 95th percentile is calculated. The values exceeding the threshold is considered as outliers and these outliers are replaced using

the Last Observation Carried Forward (LOFC) technique. *Figure 3* shows the outlier correction trimmed time series.

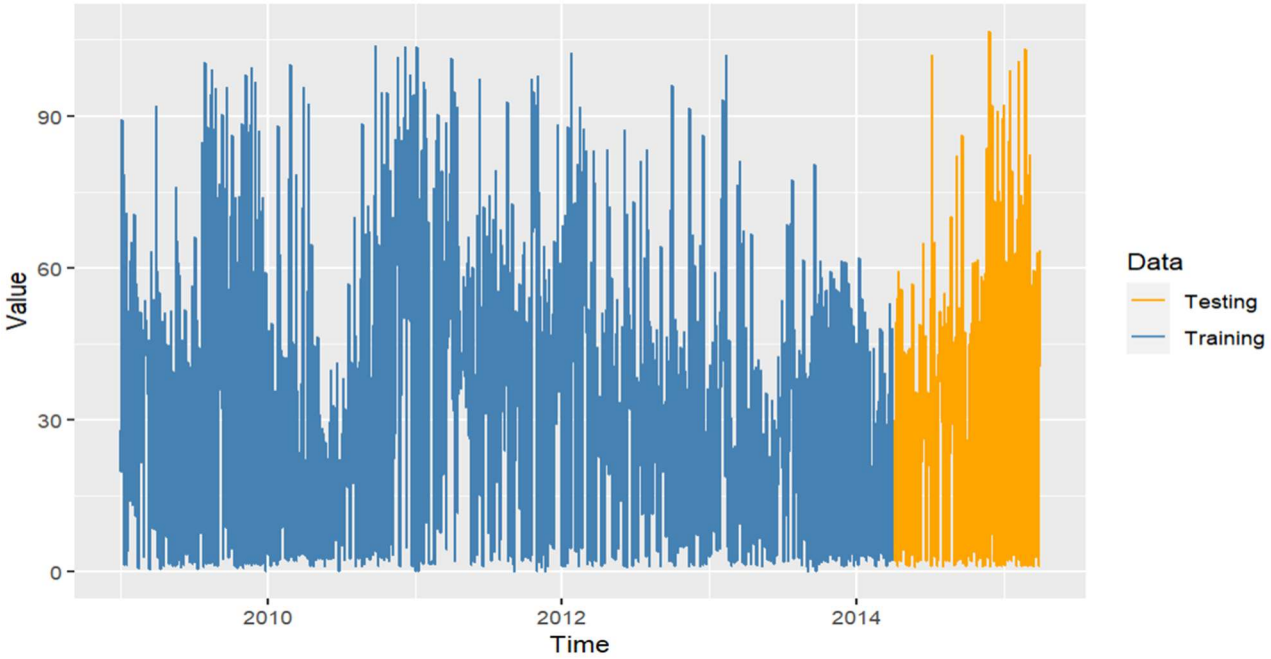

Figure 3: Outlier Correction in trimmed time series data using LOFC technique.

3. Exploratory Data Analysis

To understand the underlying structure and the behaviour of the time series data exploratory data analysis is conducted for training dataset. *Figure 4* shows the decomposition of the time series.

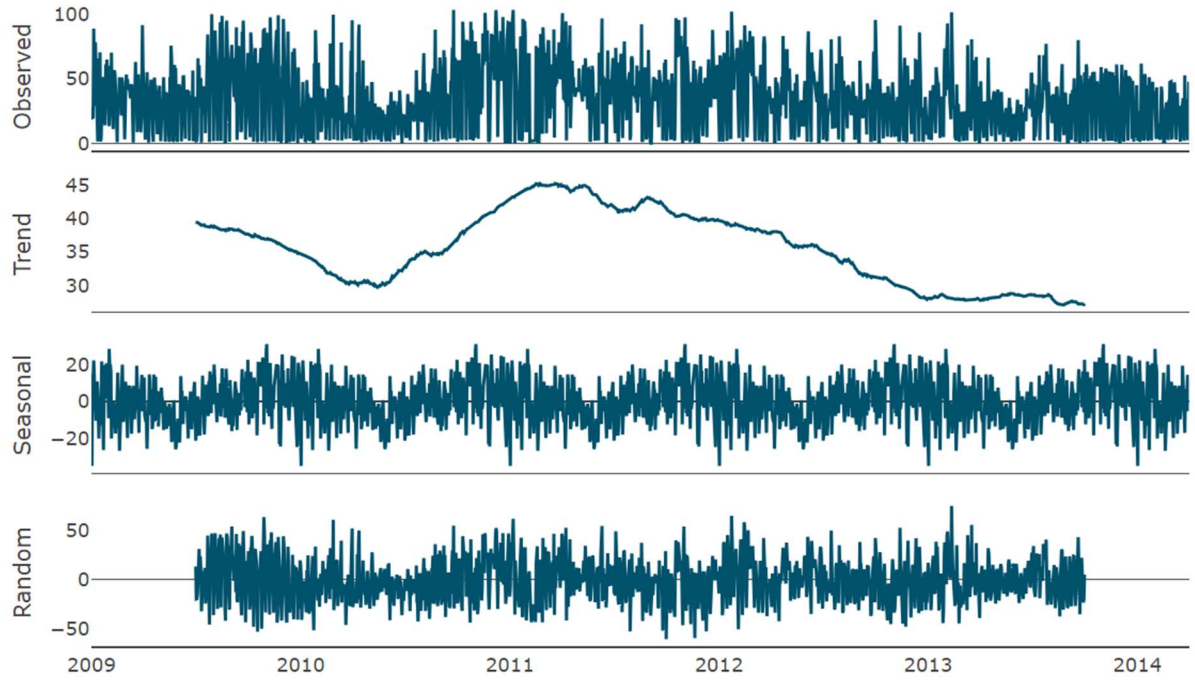

Figure 4: Decomposed Time Series

As shown in the *figure 4*, the observed time series shows the original time series data which composed with trend, seasonal and random(noise) components. The trend component in the time series shows non-monotonic trend as at the beginning of the time series shows a downward trend followed by an upward trend starting from the year 2010 till year 2011 and then time series experienced a decline after year 2011. The seasonal component in the time series captures periodic pattern that occur in regular time intervals. The random(noise) component shows a small noise.

3. ARIMA

After identifying unusual observations and understanding patterns exists in the time series data, it is necessary to observe the variance that exists in time series data. When modelling ARIMA, it is crucial to understand the underlying time series is stationary, meaning constant mean and variance. Following *figure 5*, along with the *figure 6* shows the rolling mean and rolling variance with a window size of 30.

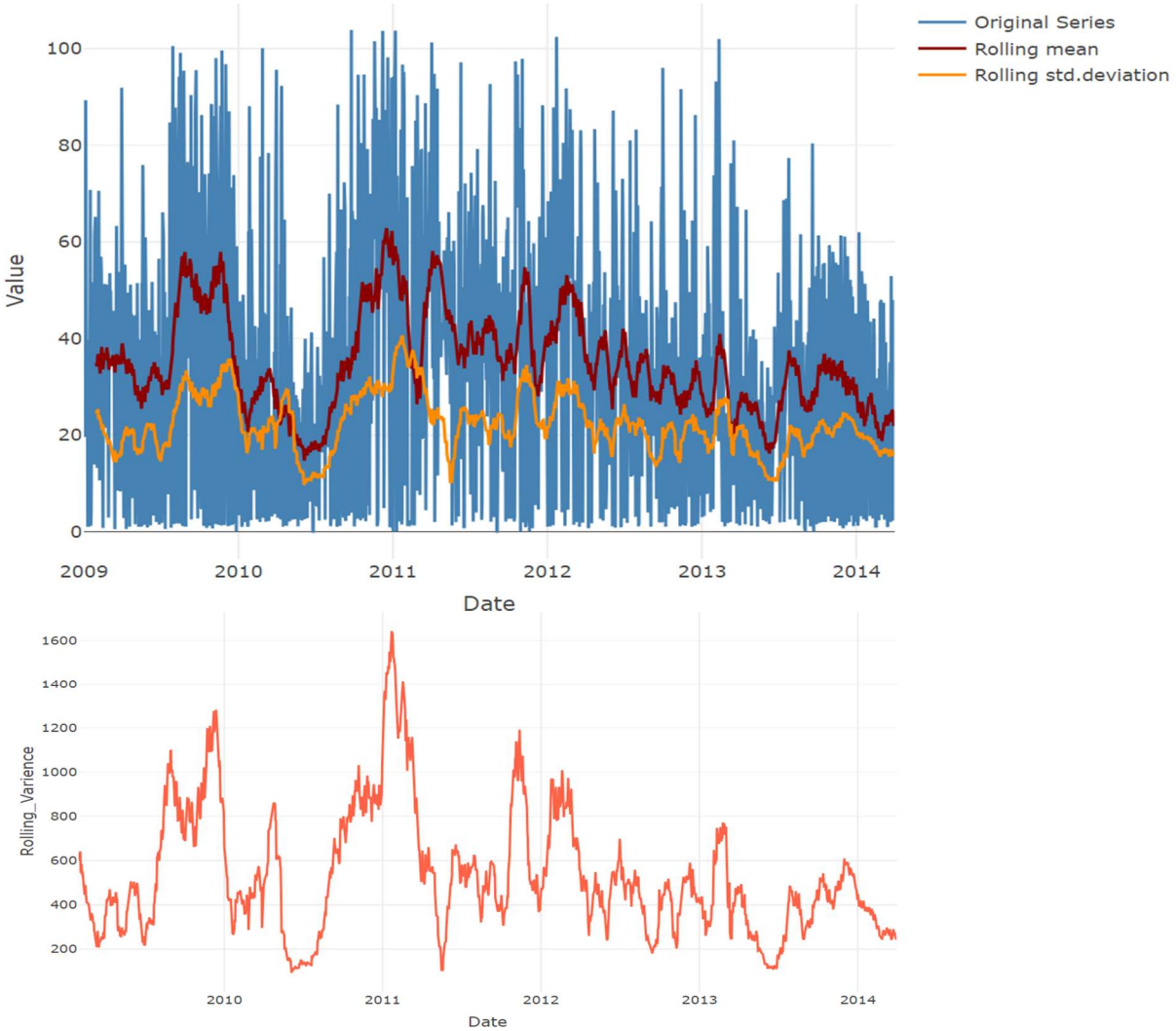

Figure 5: Rolling mean, Rolling Standard Deviation and Rolling Variance of the time series with window size of 30.

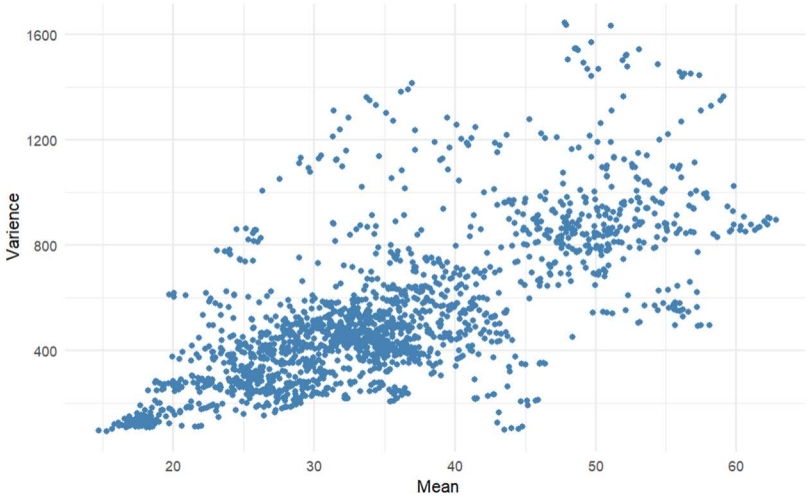

Figure 6: Rolling mean vs Rolling variance of the time series.

As shown in the *figure 6*, variability and the mean of the time series shows a upward trend over time, suggesting that the mean and variability is not constant and it is changing over time. Therefore, to stabilize the variance, Box-Cox transformation is used. This transformation is parameterized by λ, and it is adjusted to find the best transformation for stabilizing the variance.

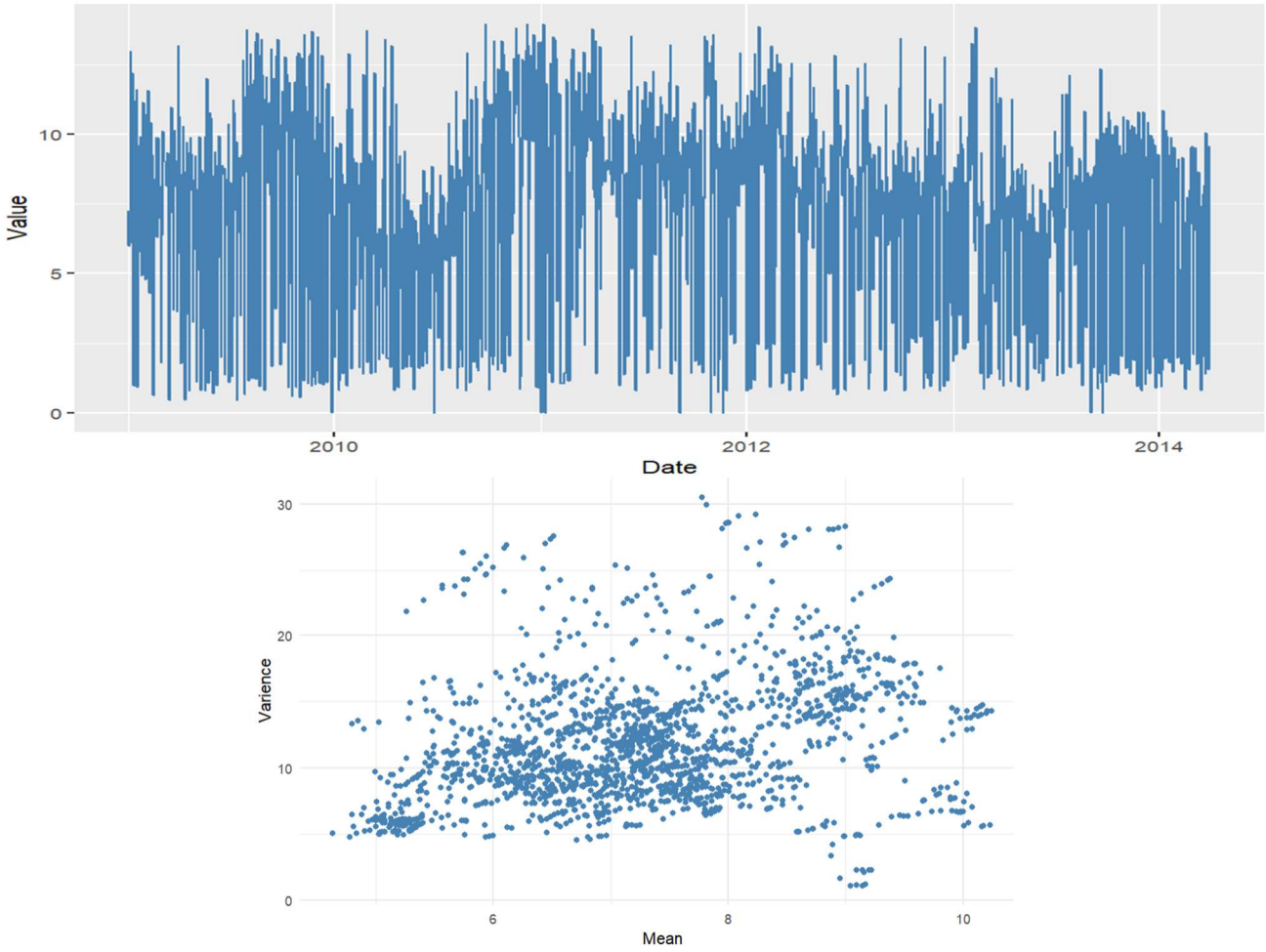

Figure 7: Box-Cox transformed time series and the scatter plot of rolling mean vs rolling variance of the Box-Cox transformed time series.

Before applying Box-Cox transformation, a constant term is added to time series data to ensure that the series has no 0 or negative terms and to make it suitable for the Box-Cox transformation. *Figure 7* shows the time series after applying the Box-Cox transformation and the plot of rolling mean vs rolling variance of the Box-Cox transformed time series. The optimal value for the best transformation (λ) is 0.409 that is used for stabilizing the variance. The scatter plot in *figure 7* shows no obvious relationship, it suggests that the variance of the series is relatively constant indicating homoscedasticity.

To further understand about the repeating patterns, trends within the data such as the seasonality and to check stationarity of the time series data that obtained after the application of Box-Cox transformation, ACF (Auto-correlation function) and PACF (Partial Auto-correlation function) plots have been incorporated.

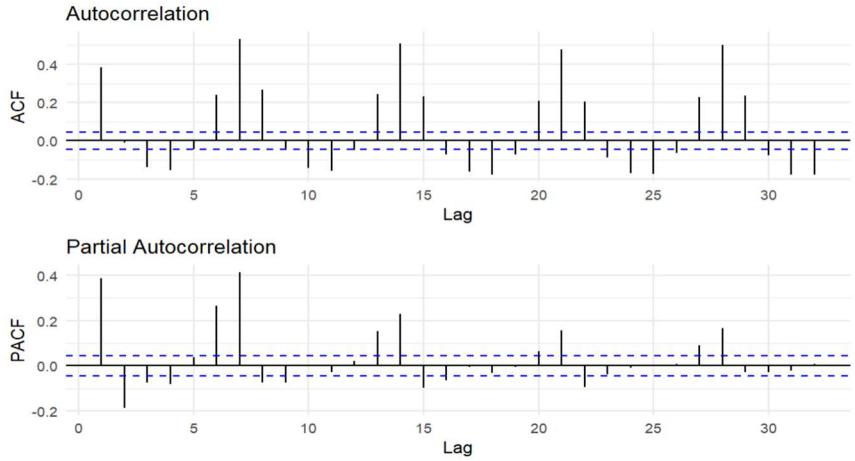

Figure 8: ACF (Auto-correlation function) and PACF (Partial Auto-correlation function) plots for Box-Cox transformed time series.

From the *figure 8*, it is evident that the time series is not stationary as it persists with higher lags and shows significant spikes at regular lags 7, 14, 21 and 28 suggesting that the seasonality is present weekly in the data. Therefore, to remove the seasonality from the time series data, seasonal differencing is performed with a lag 7.

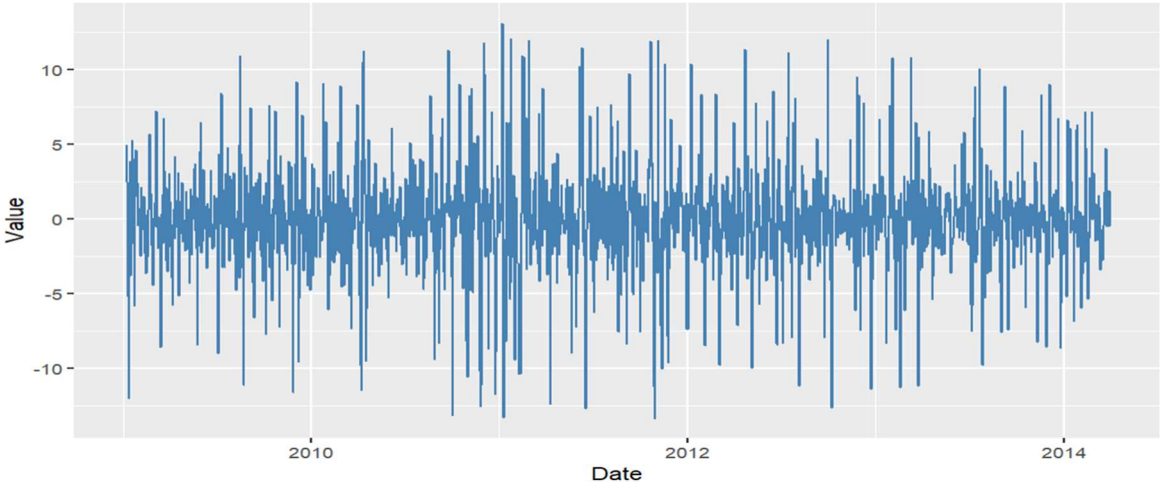

Figure 9: Seasonality Differenced time series with lag 7.

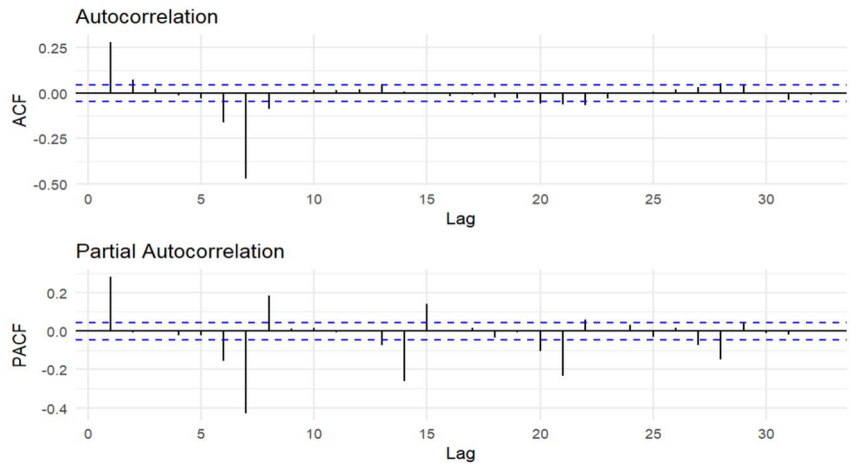

Figure 10: ACF (Auto-correlation function) and PACF (Partial Auto-correlation function) plots for seasonality differenced time series with lag 7

Figure 10 shows the ACF (Auto-correlation Function) plot has a rapid decay autocorrelation suggesting the time series is stationary. Furthermore, the plots in *figure 10* have been incorporated to identify the most suitable ARIMA model for the time series. To analyse non-seasonal part of the ARIMA model (p and q parameters), the first 7 lags in ACF and PACF have been observed. According to PACF plot the number of significant lags for p are equal to 1 while according to ACF plot number of significant lags for q are equal to 4. To analyse the seasonal part of the ARIMA model (P and Q parameters), the lags at 7,14, 21…. in ACF and PACF have been observed. The number of significant lags for P is 0 and for Q is 3. Since we did not difference non-seasonal part of the ARIMA model, d is equal to 0 while for seasonal part of the ARIMA model D is equal to 1. The final ARIMA model deduce from the ACF/PACF plot is ARIMA $(1,0,4)(0,1,3)_7$. The *table 1*, shows the comparison MAE of training and test set of different ARIMA models.

Table 1: Summarized MAE of different ARIMA Models.

Model Description	**Model**	**Training set MAE**	**Test set MAE**
1.Model deduce from the ACF/PACF plots	ARIMA $(1,0,4)(0,1,3)_7$	13.28	15.40
2. Auto ARIMA Model	ARIMA(5,1,1) with drift	16.18	22.47
3.ARIMA with sinusoidal components as external regressors.	ARIMA $(1,0,4)(0,1,3)_7$	20.18	16.80
4.ARIMA with Italian holidays as external regressors.	ARIMA $(1,0,4)(0,1,3)_7$	20.71	17.04

As depicted in *table 1*, out of all the models the model that deduce from the ACF/PACF plot shows the least values for MAE. To model 4th ARIMA model, all the main Italian bank holidays are considered. Since model 1 has the lowest MAE, *figure 11* and *figure 12* shows the associated residuals for the first model and ARIMA forecast of the time series on the test set.

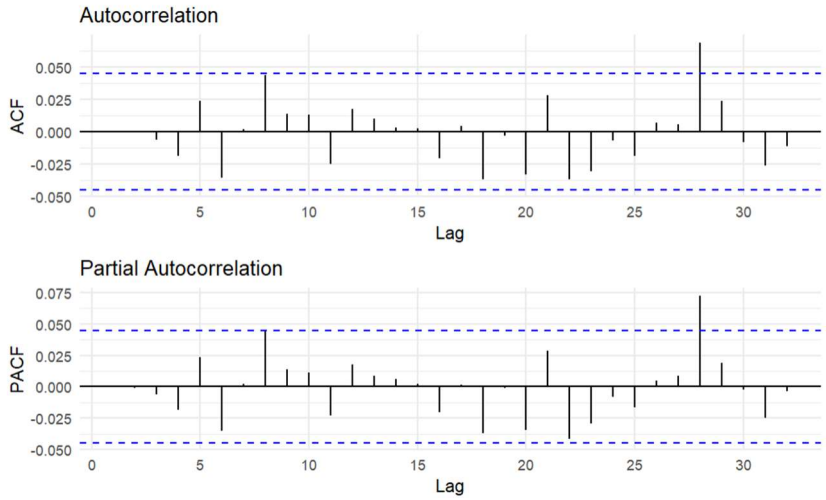

Figure 11: ACF (Auto-correlation function) and PACF (Partial Auto-correlation function) plots for first model's residuals.

According to the *figure 11*, it shows that there is not much significant autocorrelation at specific lags. However, to further assure the presence of autocorrelation in the residuals, Ljung-Box test is performed. The result of the Ljung-Box test provided are:

```
##
##  Box-Ljung test
##
## data:  residuals_mod_1
## X-squared = 39.461, df = 36, p-value = 0.3179
```

The results suggests that there is no evidence to reject the null hypothesis, meaning that there is no significant autocorrelation in the residuals of the first ARIMA model. This indicates that the first ARIMA model has likely captured the underlying structure of the time series data.

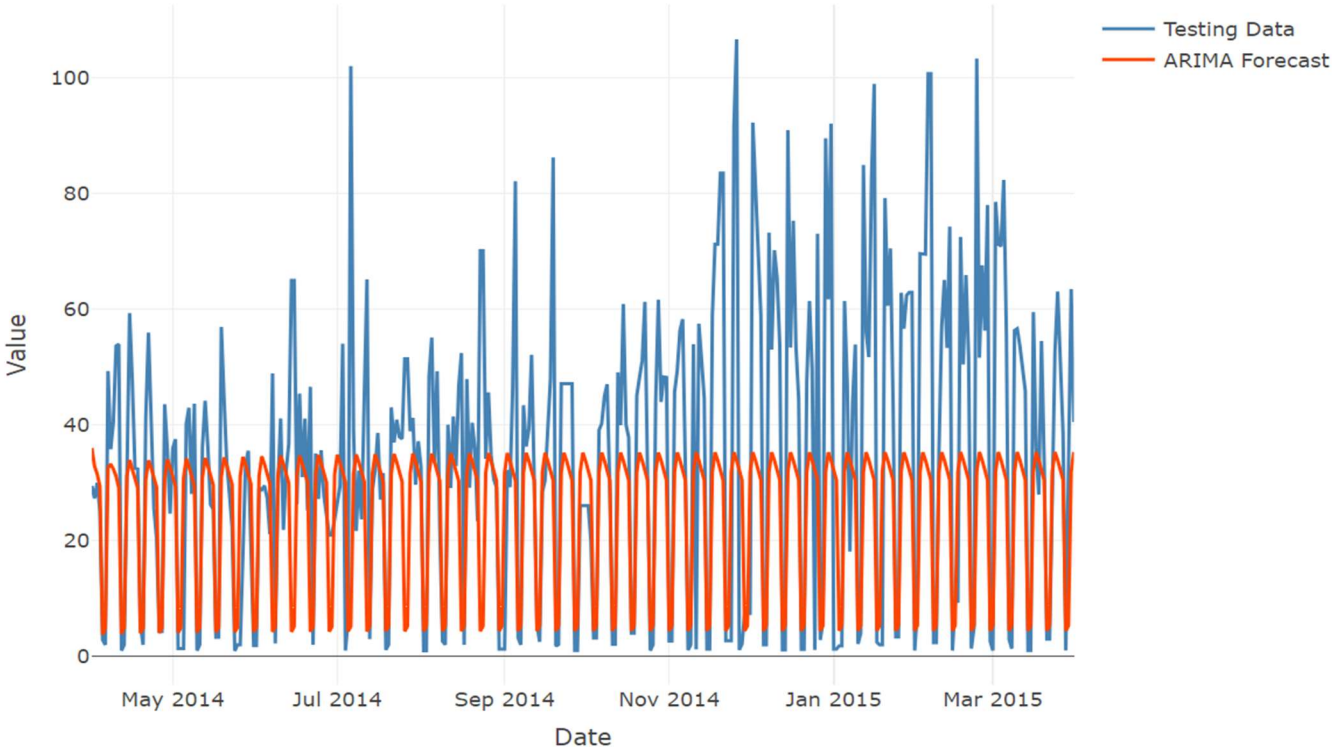

Figure 12: ARIMA Forecast of model 1 on the Test Data

4.UCM (Unobservant Component Model)

The UCM (Unobservant Component Model) is a type of time series model that used to decompose a series into various components such as trend, seasonality and irregular and noise components.

Table 2 shows the summarized MAE values of the UCMs that incorporated to analyse training and testing data. The first UCM allows to capture only the seasonal pattern in training data. The log-variance of the training data is used to set the initial values for the state disturbance for seasonal component and for the trend component.

The second UCM allows to capture both the underlying trend and the seasonal component in the time series data. The initial values for the model captures the log-variances of the training data of the state disturbance of level, slope and the seasonal component.

The third UCM model combines a random walk trend (1st-order trend), weakly seasonality (modelled as dummy variables) and yearly seasonality (modelled with trigonometric function). A set of initial values are used along with an update function that is used during the optimization process.

The fourth UCM model combines second-order polynomial trend, weakly seasonality and yearly seasonality. The initial values that set in this model is similar to the third model and an update function is used to adjust the variances during the optimization process.

Table 2: Summarized MAE values for different UCM Models.

Model Description	Training set MAE	Testing set MAE
1.Seasonal UCM with a weakly period (7 days).	24.78	16.10
2.State-Space model with local linear trend and weekly seasonality component.	**26.61**	**15.48**
3.State-Space model that combines random walk trend, weakly seasonality and yearly seasonality.	26.58	17.07
4. State-Space model that combines second-order polynomial trend, weakly seasonality and yearly seasonality.	26.58	17.07

Out of all the models summarized in the *Table 2*, the second model is relatively best performed model for the testing data. Therefore, this model is selected to forecast for all days from 2015-04-01 to 2015-11-07. *Figure 13,* shows second UCM model forecast of the time series on the test set.

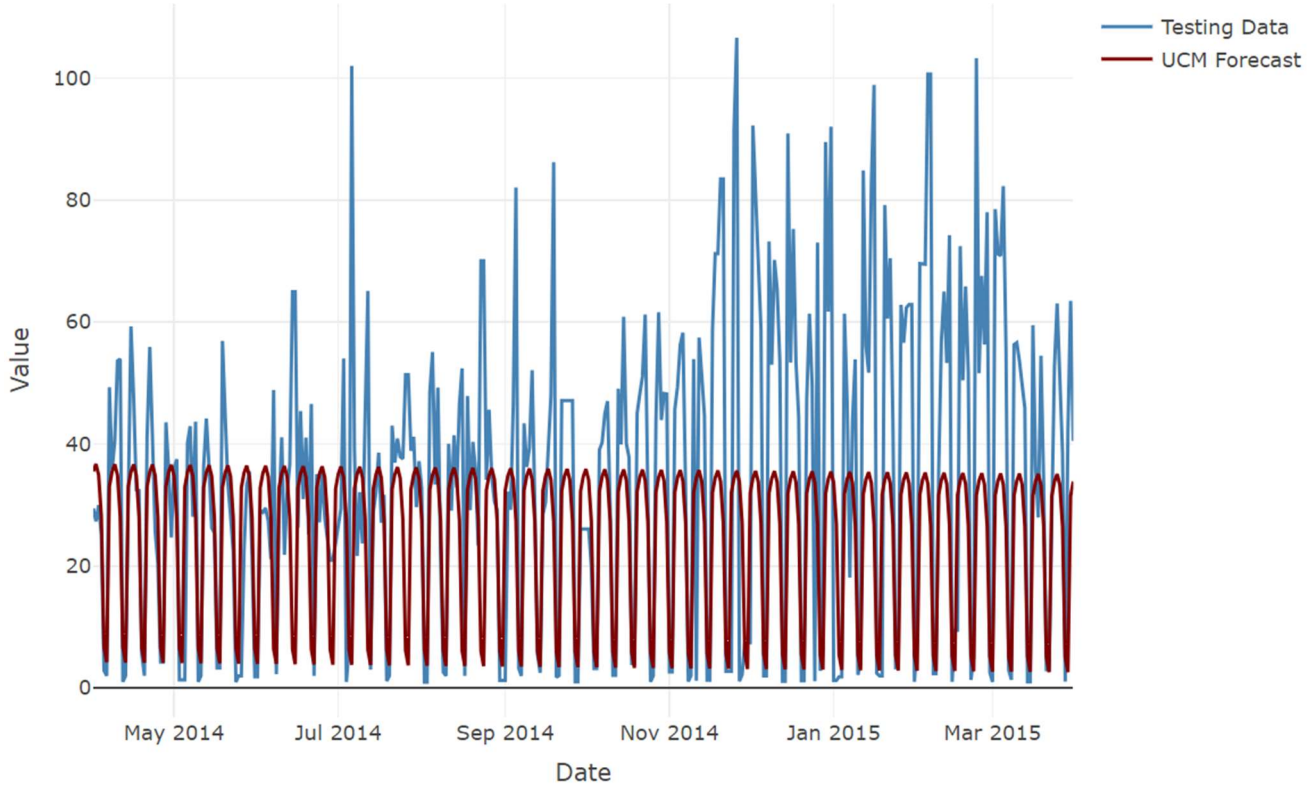

Figure 13: UCM Forecast of the 2nd model on Test data

5.Machine Learning Models

There are several machine learning algorithms and methods exists in analysing and forecasting time series data. Among them these machine learning models are implemented in this project: Random Forest, k-NN(K-Nearest Neighbours) and eXtreme Gradient Boost (XGBoost) algorithm.

The first Random Forecast model depicted in *Table 3*, is built by adding 13 lagged features to the training dataset and the model consists with 100 decision trees. The second Random Forest model is constructed similar to the 1st model with 13 lagged features and 100 decision trees along with 10-fold cross-validation for training Random Forest model. *Figure 14* shows the Random Forest forecast on test data and their MAE, MAPE and RMSE values are summarized in *Table 3*.

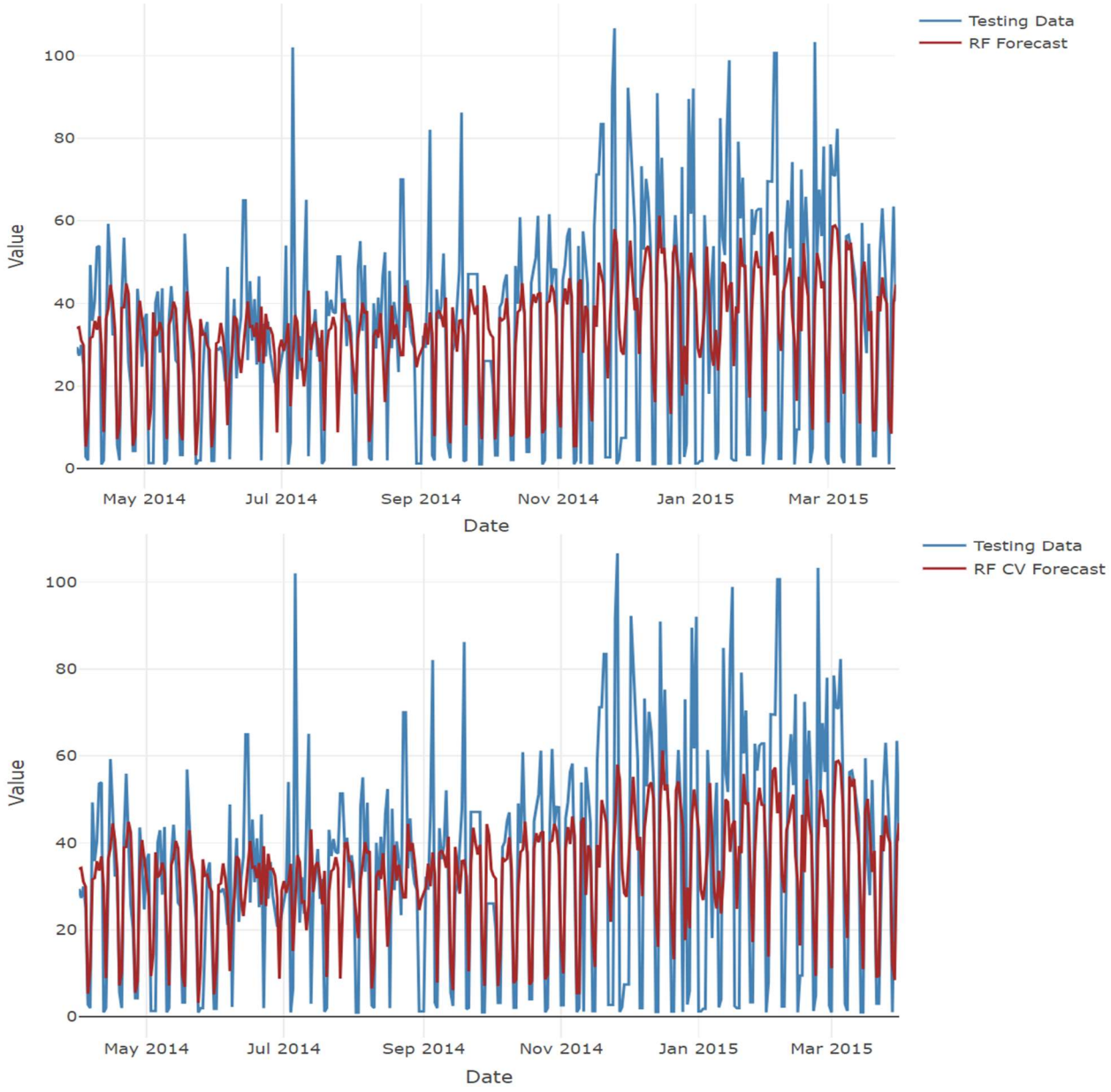

Figure 14: Random Forest Forecast on test data (Model 1) and Random Forest Forecast implemented using Cross Validation on test data (Model 2).

Model 3 and 4 depicted in *Table 3* summarizes the results for MAE, MAPE and RMSE that built by incorporating k-Nearest Neighbours(k-NN) algorithm. Both models consist with 13 lags that used as autoregressive variables in training k-NN model. The 3rd model uses a recursive strategy with 3 nearest neighbours. The 4th model is also encompassed with recursive strategy along with the transformed training samples using additive method and used different k parameters. *Figures 16* shows the k-NN forecast on test data for model 3 and 4.

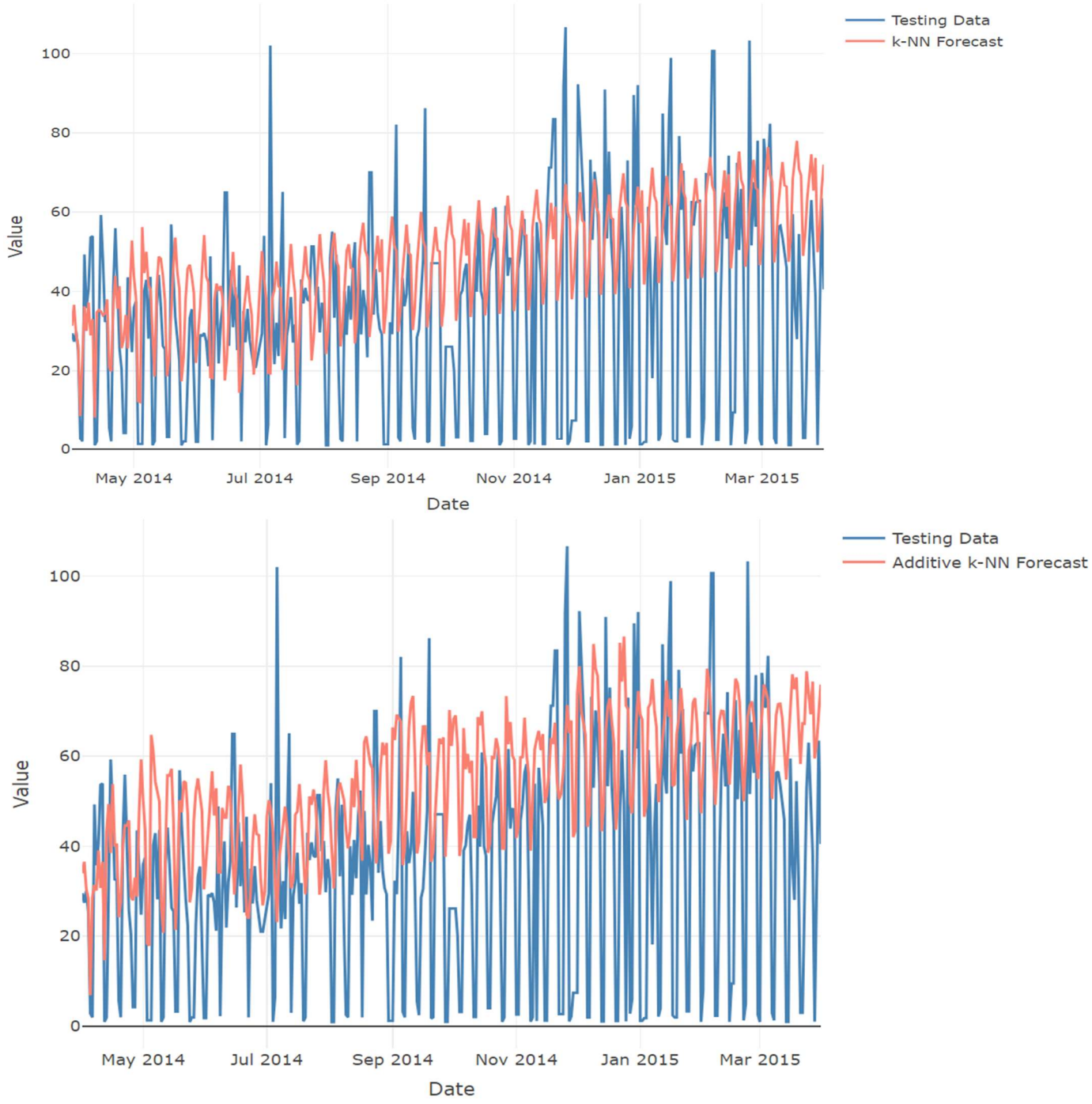

Figure 15: k-NN Forecast on test data (model 4) and k-NN model trained using additive technique Forecast on test data (model 5).

Final models in *Table 3* (model 5 and 6) are implemented using XGBoost algorithm. The dataset that is used in both models are also composed with 13 lagged features. The 5th model is trained with 100 boosting iterations. The 6th model is also using same number of boosting iterations along with 10-fold cross-validation. *Figure 16* shows the XGBoost forecast on test data for model 5 and 6.

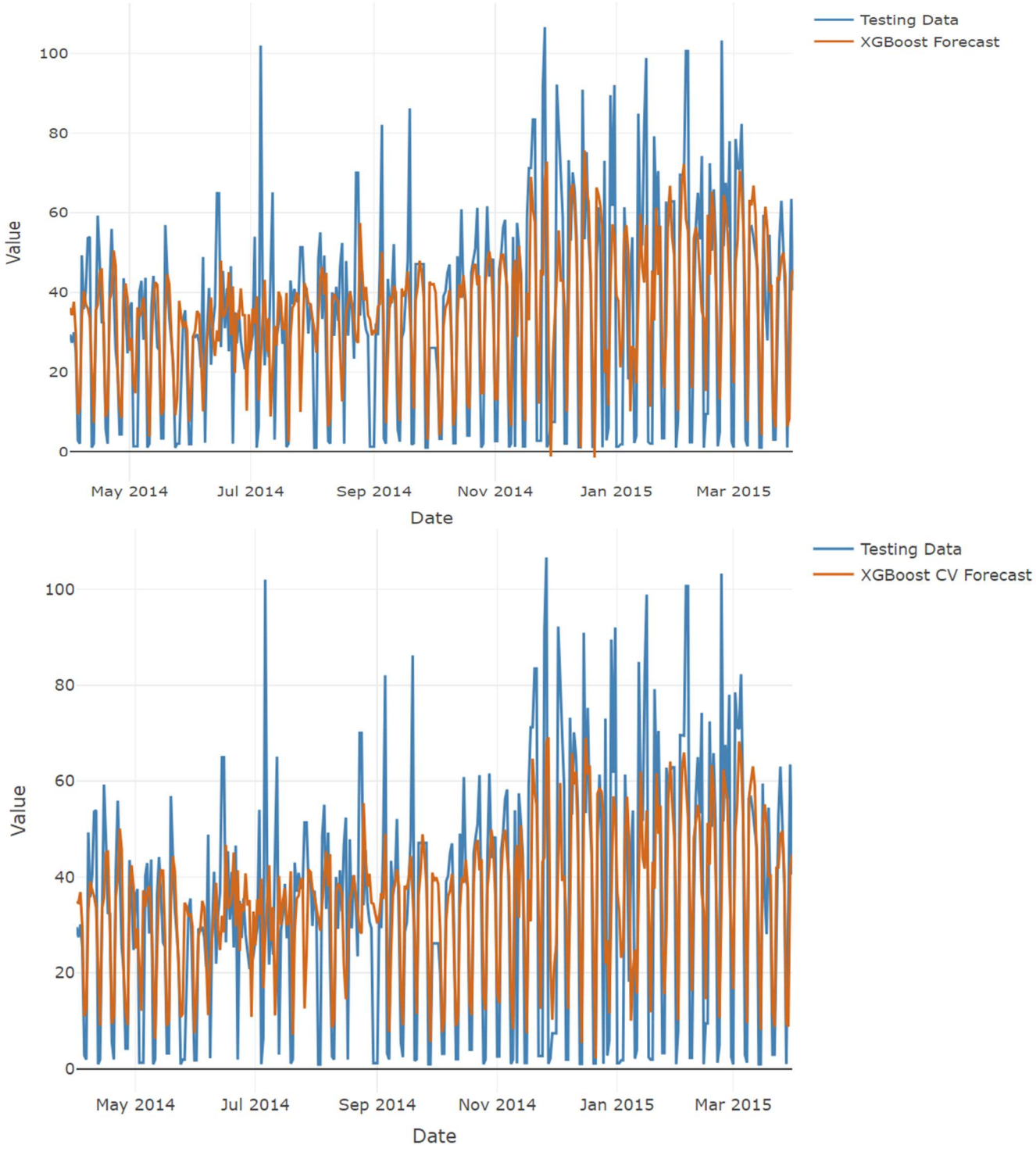

Figure 16: XGBoost Forecast on test data (model 5) and XGBoost implemented using cross-validation Forecast on test data.

Table 3: Summarized MAE, MAPE and RMSE of different Machine Learning Models.

Machine Learning Model	Model Description	Testing set MAE	Testing set MAPE	Testing set RMSE
Random Forest	1.Random Forest model with the lag values.	15.19	333.442	20.27
	2. Random Forest model with the lag values implemented using cross-validation method.	15.54	332.346	20.44
k-Nearest Neighbours(k-NN)	3. Recursive KNN model with lag values.	20.35	602.617	25.57
	4. Recursive KNN model with lag values and transformed training samples.	24.50	720.306	29.66
eXtreme Gradient Boost (XGBoost)	5.XGBoost model with the lag values.	14.67	328.902	20.00
	6. XGBoost model with the lag values implemented using cross-validation method.	14.66	335.733	19.68

Out of all the machine learning models summarized in *Table 3*, eXtrme Gradient Boost (XGBoost) is the best performed model on the testing data. Therefore, this model is selected to forecast for all days from 2015-04-01 to 2015-11-07.

6.Conclusion

The primary objective of this project is to develop linear and machine learning models to the given time series to compare their predictive performance and to provide a forecast for all the days from 2015-04-01 to 2015-11-07. The challenges encountered in building the models were handling missing values with appropriate imputation techniques, identification of outliers and making datasets with relevant time series format to apply different models. *Table 4* shows the best performed models that is selected under ARIMA, UCM and machine learning algorithms.

Table 4: Summarizing the best performed models.

Model	Model Description	Testing set MAE
ARIMA	ARIMA $(1,0,4)(0,1,3)_7$	15.40
UCM	State-Space model with local linear trend and weekly seasonality component.	15.48
Machine Learning Algorithm	eXtreme Gradient Boost (XGBoost)	14.66

Out of all the models that summarized in *Table 4*, machine learning eXtreme Gradient Boost (XGBoost) algorithm shows the best performance for the test set.