Streaming Data Management and Time Series Analysis Project

Luca Porcelli

Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy

Abstract. Time series analysis is a branch of statistics essential for understanding structures and patterns in sequentially observed data. This study focuses on predicting the average number of days required to close requests, using a dataset that covers the period from January 4, 2007, to March 31, 2015. The main objective is to forecast the average days to close requests for the period from April 1, 2015, to November 7, 2015. To achieve this goal, three distinct approaches were employed: an AutoRegressive Integrated Moving Average (ARIMA) model, an Unobserved Components (UCM) Model, and a Machine Learning (ML) model. The process involved building predictive models, optimizing parameters through grid search, and evaluating performance using Mean Absolute Error (MAE). The initial dataset was pre-processed to handle null values and outliers, ensuring temporal continuity and stationarity through tests such as ADF and KPSS. A logarithmic transformation was applied to reduce variance and manage outliers.

Keywords: Time Series Analysis · Forecasting · ARIMA · UCM · Machine Learning

1 Introduction

Time series analysis is a branch of statistics that deals with examining sequences of observed data in chronological order. Time series are wherever in the real world, finding applications in various fields such as economics, meteorology, medicine, and engineering. The main goal of time series analysis is to understand the structures and patterns present in the data to make accurate future forecasts.

In this project, time series analysis is applied to a dataset that includes three fields: date, day of the week, and average days to close requests. The period considered ranges from January 4, 2007, to March 31, 2015. The purpose of the project is to provide forecasts for the days from April 1, 2015, to November 7, 2015, using three distinct approaches: an ARIMA model, a UCM, and a Machine Learning model.

The main objectives of the project are as follows:

- **Construction of Predictive Models:** Create three sequences of forecasts using the ARIMA, UCM, and ML models. Each model will be done exploration and experimentation to determine the optimal configuration.

- **Performance Evaluation:** Evaluate the models' performance through a validation process, using Mean Absolute Error (MAE) as the primary metric to compare forecast accuracy.
- **Forecasting:** Perform predictions for future data without the presence of test data to evaluate them.

Time series analysis allows for understanding the dynamic behavior of data over time. Accurate time series forecasting can lead to significant improvements in planning and decision-making, reducing uncertainty and optimizing resources. In the context of this project, providing accurate forecasts of the average days to close requests can have practical implications for operations management and customer satisfaction.

2 Data and Experimental Setting

2.1 Datasets and Pre-processing

The dataset consists of three columns:

- **date**: represents the date in the format *yyyy-mm-dd.*
- **weekday**: a string containing the corresponding day of the week for each date.
- **ave_days**: average number of days required to close requests that were closed on that day.

The first step undertaken was the analysis of the trend of values over time. Figure 1 shows the overall trend of the series on a weekly and monthly basis. This allowed us to observe a generally increasing trend from 2008 to 2012, followed by a decreasing trend from 2012 to 2014, with a recovery in 2015. One surprising observation from these initial graphs is the presence of high outliers that deviate from the mean. Using a weekly plot, fewer outliers are evident compared to a daily plot, which would certainly show many more. Lastly, there appears to have been significant variability in the data in 2015.

The first step in the pre-processing was to ensure the temporal continuity of the series by checking for the absence of null values within the data. In the **ave_days** column, it was observed that there were 202 null values. Subsequently, it was investigated which day of the week typically had null values:

- **Monday**: 4.45%
- **Tuesday**: 0.23%
- **Wednesday**: 0.00%
- **Thursday**: 0.93%
- **Friday**: 0.93%
- **Saturday**: 1.40%
- **Sunday**: 38.84%

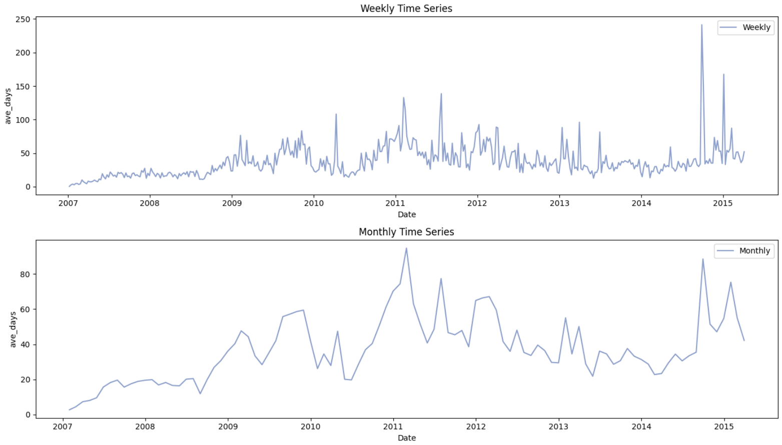

Fig. 1. Time series with weekly and monthly granularity

It can be noted that Sunday has more than 38% of missing values. This observation can be attributed to the fact that most companies do not operate on Sundays. To address this issue, the decision was made to impute missing values on a weekly basis using the weekly mean, rather than the overall mean. Additionally, duplicates within the historical series were checked and found to be absent.

Regarding outliers, a more detailed analysis was conducted using a Box Plot (Figure 2). As suspected, numerous outliers can be observed. The boxplot appears highly compressed, with Q1 around 1 and Q3 around 100. However, values of **ave_days** extend up to almost 800.

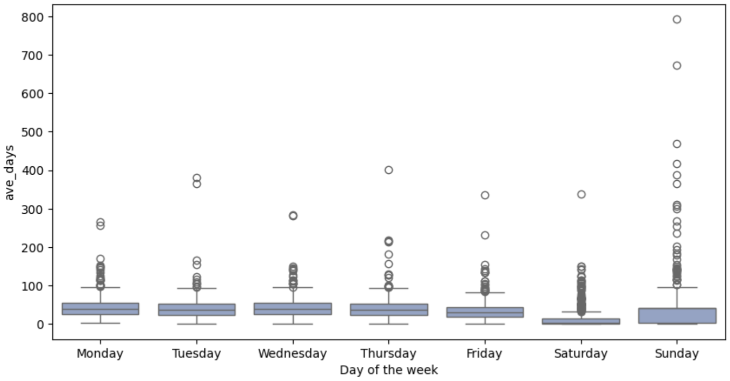

Fig. 2. Boxplot outliers

The outliers represent 3.39% of the entire dataset. To address the outlier issue, it was decided to set a maximum threshold for the **ave_days** value at 100. This decision was based on the assumption that the maximum average number of days required to close requests would be 100. By setting all outlier points (a total of 102) to 100, no points are removed, thus maintaining the temporal continuity of the dataset.

2.2 Stationarity Analysis

After completing the preprocessing steps, we conducted the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test to evaluate the stationarity of the time series. Stationarity implies that statistical properties of the time series such as mean and variance remain constant over time. Checking for stationarity is crucial because accurate forecasting requires working with stationary data; non-stationarity can lead to inaccurate forecasts.

The ADF test checks if the time series contains a unit root, which would indicate non-stationarity. If the null hypothesis of a unit root is rejected, we can conclude the series is stationary. In our case, the ADF test produced a p-value of 0.0049, rejecting the null hypothesis of non-stationarity. This suggests the time series is stationary. On the other hand, the KPSS test checks the null hypothesis that a time series is stationary around a level or trend. The KPSS test returned a p-value of 0.01, confirming the null hypothesis of stationarity. Both tests indicate the time series is stationary.

In addition to the ADF and KPSS tests, we analyzed the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots for further confirmation. The ACF and PACF plots show significant correlations at lags of 7, 14, and 21 in the ACF plot, while the PACF plot highlights specific correlations at lag 7. These results clearly indicate the presence of a weekly seasonal pattern in the data.

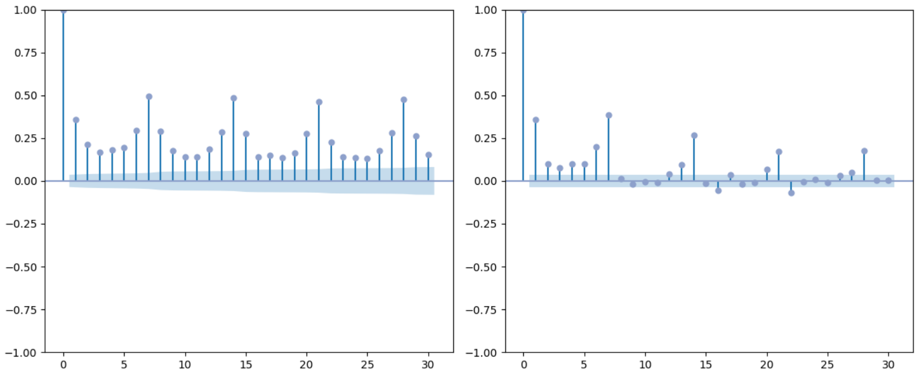

Fig. 3. Plots of the autocorrelation function (ACF) and partial autocorrelation function (PACF)

2.3 Transformation

Although the stationarity of the time series was demonstrated, a decision was made to apply a logarithmic transformation. This choice is motivated by the fact that logarithmic transformation helps reduce the variance of the time series, thereby improving the handling of outliers, which has been reduced to 100. Without this transformation, outliers would have significantly increased the variability. After applying the logarithmic transformation, the ADF and KPSS tests showed identical values to those obtained with the original time series. However, the logarithmic transformation did not resolve the issue of seasonality. Therefore, it may be advantageous to include a seasonal component in predictive models to enhance forecasting accuracy.

Figure 4 displays the time series after applying the logarithmic transformation with weekly granularity. Unlike Figure 1, there are no longer any noticeable spikes, and the overall variability of the series is reduced.

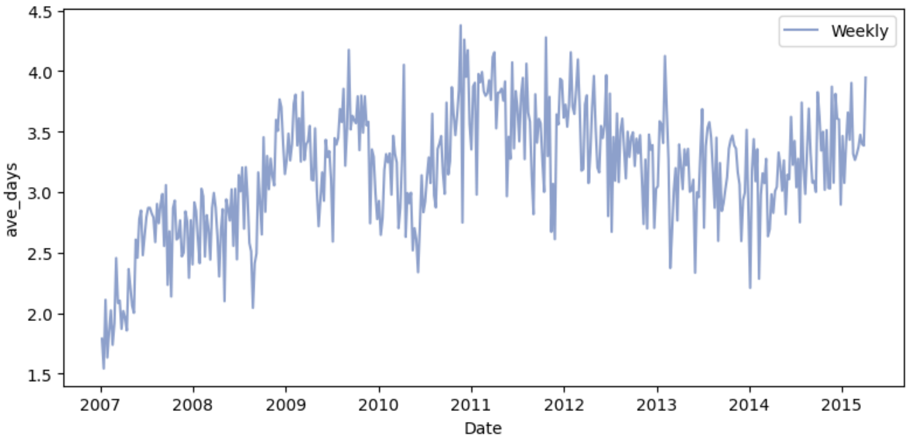

Fig. 4. Time series with logarithmic transformation and weekly granularity

3 Models

Before proceeding with the construction of the models, it was decided to create three types of datasets. The first dataset contains the already available data and will be used for all three model families. For the machine learning models, two additional datasets were created with additional information regarding holidays. The second dataset includes an additional column that takes the value 0 except on holidays, where the value will be 1. The third dataset, instead, creates a separate column for each holiday, assigning the value 1 only on the specific day of the corresponding holiday. The holidays considered are all those in Italy that

are on the same day every year: *New Year's Day*, *Epiphany*, *Liberation Day*, *Labour Day*, *Republic Day*, *mid-August*, *All Saints' Day*, *Immaculate*, *Christmas Day*, and *St. Stephen's Day.*
Regarding the dataset division, it was split as follows:

- Train: from 2007-01-04 to 2014-08-01
- Test: from 2014-08-02 to 2015-03-31

This was done to maintain the same forecasting horizon between 2015-04-01 and 2015-11-07, which is the project's objective.

3.1 SARIMA

The SARIMA (Seasonal AutoRegressive Integrated Moving Average) model represents an evolution of the ARIMA model designed to capture seasonal variations in time series data. It is particularly effective when the data exhibit regular cycles over time, meaning recurring behaviors at regular intervals. Its theoretical formulation combines auto-regressive (AR), integrated (I), and moving average (MA) components, integrated with seasonal components (SAR, SI, SMA), to capture both short-term dynamics and long-term seasonal patterns. In formal notation, a SARIMA model is often denoted as:

$$\text{SARIMA}(p, d, q)(P, D, Q)_s$$

where (P, D, Q) represent the seasonal components and (p, d, q) represent the non-seasonal components. The parameter d indicates non-seasonal differences (denoted as Δ), while D indicates seasonal differences (denoted as Δ_s). The letters p and q indicate the order of the AR and MA components respectively, both for non-seasonal components and seasonal components P and Q.

To optimize the model's performance, an optimal parameter search for the SARIMA model was implemented using a grid search technique. Initially, a list of parameters to test was defined, including non-seasonal components (p, d, q) and seasonal components (P, D, Q) with a seasonal period of 7 days. All possible combinations of these parameters ranging from 0 to 3 were generated. During the grid search, each parameter combination was iterated over and a SARIMA model was trained. The mean absolute error (MAE) between the model's predictions and the actual values on the test set was calculated. The parameters that produced the lowest MAE were tracked. The results obtained are:

- The best parameters found are: $[1, 1, 0, 1, 1, 2, 7]$
- Lowest MAE obtained on the test set is: 16.97

In addition to providing a numerical result, Figure 5 illustrates the predictions obtained on the test data, comparing them with the actual values.

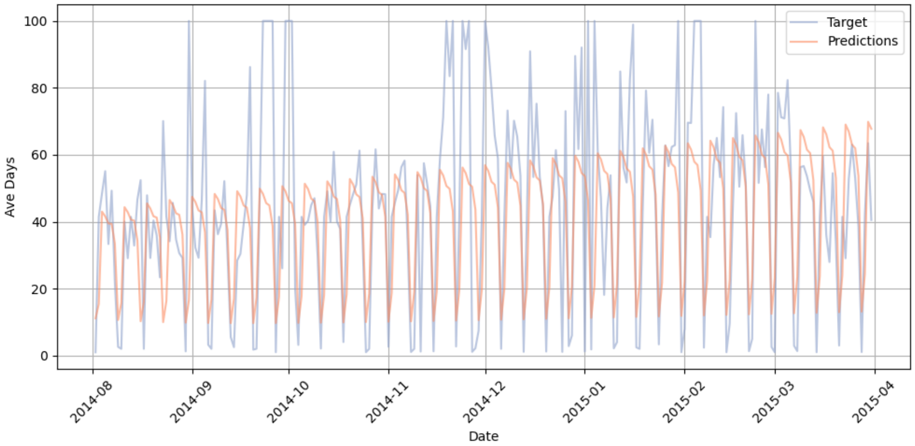

Fig. 5. Comparison of Real Values and Forecasts of Average Days (July 2014 - April 2015)

It can be stated that the model performs reasonably well in capturing the overall trend, following its trajectory. It is observed that it tends to rise correctly, but has difficulty predicting extreme values, which is however quite common in time series forecasting when there is high variability.

3.2 UCM

In models with unobserved components (**UCM**), a time series is considered as the sum of several components that are not directly observable. In its typical and most extended form, a UCM is given by the sum of **trend**, **cycle**, **seasonality**, and **white noise**:

$$Y_t = \mu_t + \psi_t + \gamma_t + \varepsilon_t$$

This allows for optimal forecasts to be constructed for the series and for each component, along with the classic measures of statistical inference uncertainty (standard errors, confidence intervals, prediction error distribution).

To optimize the model's performance, a process was implemented to evaluate different models of unobserved components. Specific parameters were defined for each component of the model, including levels, trends, and seasonality, considering the inclusion of stochastic components for each. Subsequently, forecasts were made on the test set, and the Mean Absolute Error (MAE) between the obtained predictions and the actual values in the test set was calculated, choosing the parameters that minimize the MAE. The obtained results are as follows:

- The best parameters found are: ['smooth trend', True, 7, False, False, False]
- Lowest MAE obtained on the test set is: 16.94

In addition to providing numerical results, Figure 6 illustrates the trend of predictions obtained on the test data, comparing them with actual values.

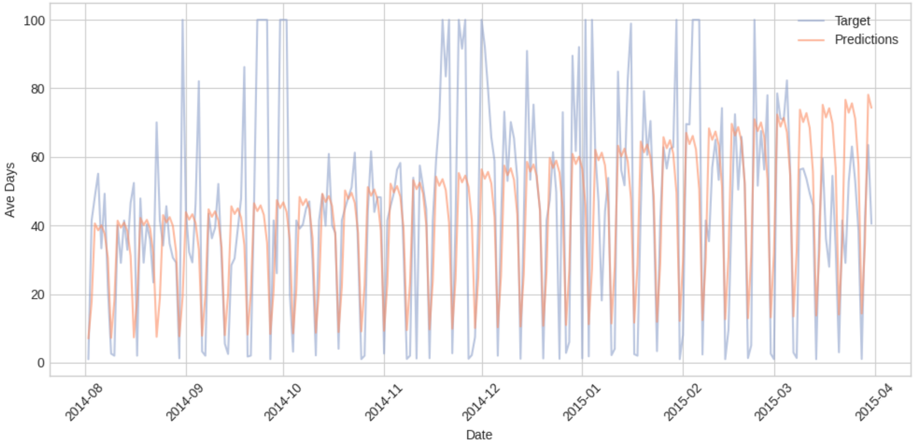

Fig. 6. Comparison of Real Values and Forecasts of Average Days (July 2014 - April 2015)

For the UCM model, it can be said that it also adequately represents the overall trend, following its direction. The only significant difference is that toward the end, the values obtained are higher compared to the forecasts of the SARIMA model.

3.3 ML

Machine learning (ML) models are primarily used for time series prediction. An alternative application is the reconstruction of noisy signals, but only if they have already been observed. Prediction remains the primary use. Many algorithms, such as tree-based methods, KNN, and some neural networks, do not forecast beyond the training data range but handle seasonality well. This limitation can be overcome by making the time series additive and stationary in variance. The objective is to estimate the function $f(x)$, minimizing mean absolute error. ML models are non-parametric, thus independent of functional form and error distribution. In this case, three types of models were chosen for use and comparison:

- **XGBRegressor:** A model based on gradient boosting. It works by combining many small decision trees, gradually improving predictions with each new tree. This approach effectively addresses complex regression problems while maintaining high efficiency and the ability to handle missing data.

- **Random Forest:** A model based on an ensemble of decision trees trained on different subsets of the dataset. Each tree provides a prediction, and their average constitutes the final result. This method enhances robustness and reduces the risk of overfitting, making it ideal for working with complex and noisy data.
- **SVR (Support Vector Regressor):** A regression version of SVM adapted to find a hyperplane that approximates the data as closely as possible, tolerating small errors. SVR is effective in handling nonlinear relationships, thanks to its capability to use kernels, making it useful in high-dimensional contexts.

For the machine learning models, as well as for the SARIMA and UCM models, a grid search approach was applied. This method allowed testing various parameter combinations to identify those producing the lowest MAE results, thereby optimizing model performance. The parameters tested for the models were as follows:
XGBRegressor:

- Learning rate: [0.01, 0.1, 0.2]
- Max depth: [1, 3, 5, 7]
- Min child weight: [1, 3, 5, 7]
- Gamma: [0.1, 0.2]
- Colsample bytree: [0.5, 0.7, 0.8]
- N estimators: [100, 200, 500]

Random Forest:

- N estimators: [100, 200, 300, 400, 500]
- Max depth: [10, 30, 50, 70, 90, 110]
- Min samples split: [2, 5, 10]
- Min samples leaf: [1, 2, 4, 6, 8]

SVR:

- C: [1, 10, 100]
- Epsilon: [0.1, 0.2, 0.3]
- Kernel: ['linear', 'poly', 'rbf']

As anticipated, to determine the best model in this case, the evaluation approach for the machine learning models was conducted on three datasets described earlier. The results obtained are visible in Table 1.

Despite the increase in information, significant performance improvement was not achieved. However, the best result was obtained with dataset 3 using a random forest. This dataset, which includes each column for every holiday, will be used for forecasting, although the difference compared to the others was minimal.

Additionally, for the ML model, over the numerical result, the trend of predictions obtained on the test data compared to the actual values is provided in Figure 7.

Table 1. Comparing the results of the three models on each dataset:

	Dataset 1	**Dataset 2**	**Dataset 3**
XGBRegressor	22.44	21.80	22.13
Random Forest	21.97	21.94	**21.78**
SVR	22.85	22.74	22.80

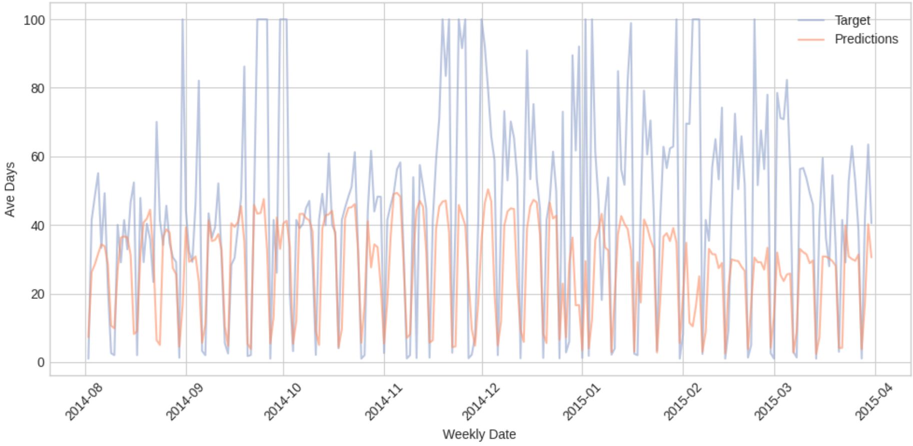

Fig. 7. Comparison of Real Values and Forecasts of Average Days (July 2014 - April 2015)

For the machine learning models (as with the others), the result of the best model is shown, which in this case is the result of the Random Forest with the added holiday variables. Specifically, unlike the other models, it is observed that the trend does not tend to rise but remains fairly constant, except towards the end where it tends to decline. This behavior does not follow the real data, which is why we have a higher MAE compared to the SARIMA and UCM models.

4 Forecast

In the context of time series analysis, forecasting plays a central role. Temporal forecasting, or forecasting, involves accurately estimating future values of a dataset based on patterns identified in its historical data. In our study, the main objective is to accurately anticipate the trend of "ave_days" for the upcoming period. The main part of this project involved selecting the most suitable models from three model families. Currently, forecasting is being conducted for the period from April 1, 2015, to November 7, 2015. Figure 8 below illustrates the forecasting of the ARIMA model. Using data from the first three observed months, the model predicts the remaining months to assess the trend up to November 2015. The results indicate a declining trend over time, unlike the previous months.

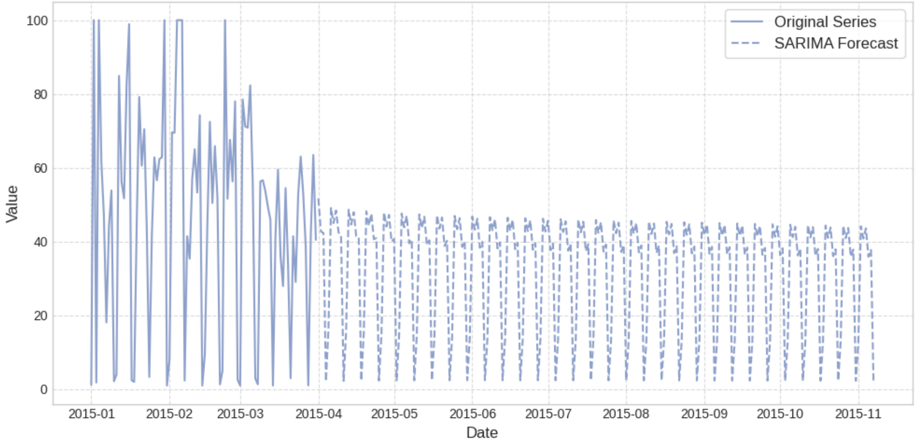

Fig. 8. Forecast SARIMA (April 2015 - November 2015)

In Figure 9, dedicated to the same year 2015, the forecast shows more stable results compared to the ARIMA model, without significant increases or decreases highlighted.

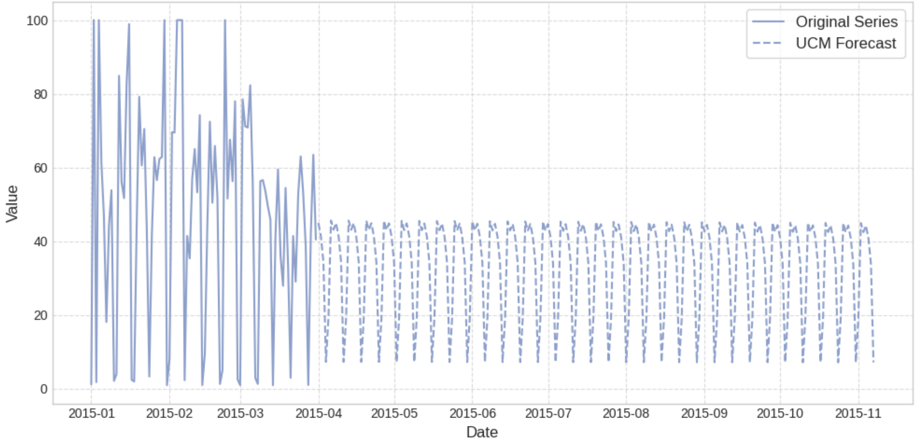

Fig. 9. Forecast UCM (April 2015 - November 2015)

Finally, in Figure 10 related to the machine learning model, an initially stable trend is observed, with values ranging between 10 and 35 during the summer, followed by a slight increase towards September.

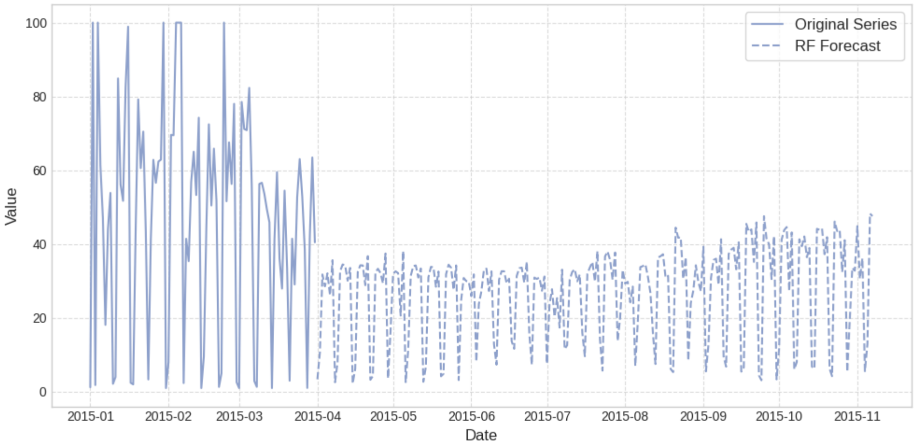

Fig. 10. Forecast RF (April 2015 - November 2015)

5 Conclusion

This study explored and compared various approaches for the analysis and forecasting of the time series "ave_days," using SARIMA, UCM, and machine learning models. The main objective was to determine which model could provide the most accurate forecasts for the period from April 2015 to November 2015. Among the models examined, SARIMA stood out for its ability to capture seasonal variations and short-term fluctuations present in the historical data of "ave_days." It predicted a slight decreasing trend in the forecast period, consistent with patterns observed in the previous data.

On the other hand, the UCM approach demonstrated greater stability in forecasts without showing significant variations during the considered period.

Machine learning models did not significantly outperform traditional models. Despite including holidays as additional variables in the dataset, improvements in forecasting performance were limited compared to SARIMA and UCM models.

Based on the results of the comparative analysis, the UCM model emerges as the most reliable choice for forecasting the series during the period considered.

Figure 11 illustrates the comparison of the three forecasting models. A decreasing trend is evident in both SARIMA and UCM models, which show very similar and nearly overlapping results. In contrast, the Random Forest model shows different results compared to the other models, with a potential increase in the series towards the end of the considered period.

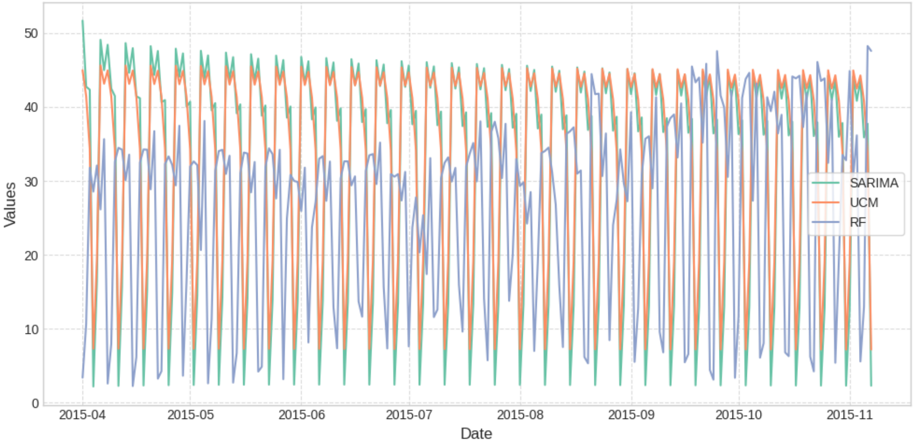

Fig. 11. Forecast with the three model (April 2015 - November 2015)