Predicting Time Series with ARIMA, UCM and Gradient Boosting

Giovanni Noè – University of Milano Bicocca

Abstract

Time series forecasting aims to predict future values of sequentially ordered data. Accurate forecasts allow organizations to make informed decisions in many different areas. This project focuses on predicting an hourly count of people passing in front of a sensor installed on a public street in Australia. To achieve this, three different models are employed: ARIMA, UCM, and a Gradient Boosting Machine. The results show that the predictions from the models built for this work differ significantly: the Gradient Boosting Machine tends to produce lower forecasts, UCM predictions fall in the middle, and ARIMA generates the highest values.

Table of Contents

Introduction

Time series analysis plays a fundamental role in understanding and forecasting data that evolve over time. By modelling temporal patterns such as trend and seasonalities, time series forecasting makes it possible to anticipate future events and support data-driven decision making. Its applications span a wide range of domains, including finance, transportation, energy management, and public safety.

In this project, the time series under study consists of hourly recordings of the number of people passing through a street in Australia. The dataset covers the period from April 15, 2015, to December 31, 2019. The observations to be predicted correspond to the period from January 1, 2020, to February 29, 2020 and are unknown values.

To achieve this forecasting task, three different models are evaluated: an ARIMA model, an Unobserved Components Model (UCM), and a machine learning approach based on a Gradient Boosting Machine (GBM).

Before producing the forecasts, several preliminary steps are carried out. First, an exploratory data analysis is conducted to identify trends, seasonal patterns, missing values, and potential outliers. Second, dummy variables are created to incorporate the effect of vacation days that may influence the target variable.

1. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in time series forecasting, as it provides an initial understanding of the structure and the main characteristics of the data. Through EDA, it is possible to identify important patterns such as trends and seasonalities, as well as to detect anomalies, missing values, and potential data quality issues.

In this project, the exploratory analysis aims to investigate the presence of long-term trends and daily, weekly, and yearly seasonal patterns, and to identify missing observations, outliers, and other anomalies. In addition, the necessary data cleaning procedures are carried out during this phase.

1.1 Preliminary Steps

Firstly, the dataset is provided with a single time column that contains both the date and the hour. Therefore, the first data cleaning step consists of splitting this variable into two separate columns: one for the date and one for the hour. This transformation facilitates subsequent analysis and allows for a clearer investigation of daily and hourly patterns.

Next, the presence of missing values is examined. The only NA values found correspond to the observations to be predicted at the end of the dataset, and therefore they are expected and do not require any additional treatment at this stage.

By inspecting the zero values in the time series, a full day with zero-only observations is identified, namely September 15, 2017. This behaviour is likely due to an external event, such as a street closure or a sensor malfunction. To avoid introducing a structural break in the series, the hourly values for that day are imputed as the mean of the corresponding hourly values from the previous and the following day. This approach preserves the continuity of the time series while providing reasonable estimates for the missing information.

Lastly, the presence of issues related to Daylight Saving Time was investigated by checking for duplicated timestamps and for days with an anomalous number of observations (i.e. 23 or 25 hours instead of the usual 24). No

such irregularities were found, indicating that the time series is not affected by Daylight Saving Time adjustments.

1.2 Descriptive Statistics

To assess the overall quality of the data and to identify potentially interesting insights, some basic descriptive statistics are computed. The most relevant findings are summarized below:

- The mean value of the series is 101.5, and 50% of the observations lie between 69 and 159, indicating a moderate level of variability around the central tendency.
- The maximum observed value is 906 and occurs at 11 a.m., while the minimum value is 0 and appears at least once at each hour between 9 p.m. and 7 a.m., when pedestrian activity is expected to be very low.
- On average, the peak hour of pedestrian traffic is 4 p.m., with a mean value of 196.70, whereas the lowest average activity is recorded at 2 a.m., with a mean of 8.85.

These insights are useful for evaluating the plausibility of the model predictions, as they provide reference values and expected patterns against which the forecasted results can be compared.

1.3 Trend and Seasonality Analysis

Trends and seasonal patterns are fundamental in time series forecasting. Understanding these patterns is essential for selecting appropriate models and improving predictive accuracy.

In this project, the time series is analysed to identify the trend and daily, weekly, and yearly seasonalities in pedestrian traffic. Visualizations are used to explore the typical behaviour at different times of the day, variations across weekdays, and broader seasonal trends throughout the year.

Trend Analysis

To identify any long-term trend in the time series, the data is first visualized through a time series plot representing the daily average.

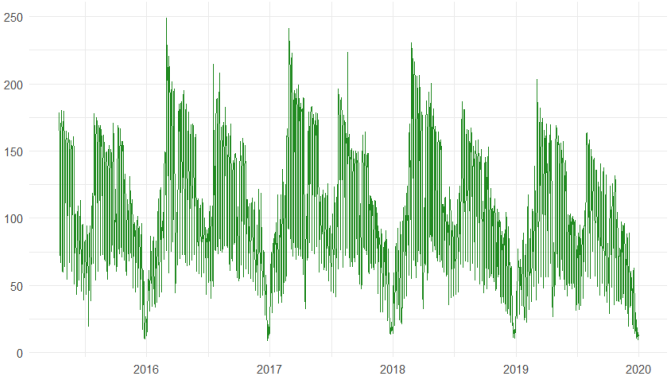

Figure 1 – Time Series Plot (Daily Average)

The time series exhibits a slightly decreasing trend over time, suggesting a gradual reduction in pedestrian activity throughout the years.

Yearly Seasonality

To visually examine the yearly seasonality, a plot of the daily averages for each year is generated, with all years overlaid in the same figure.

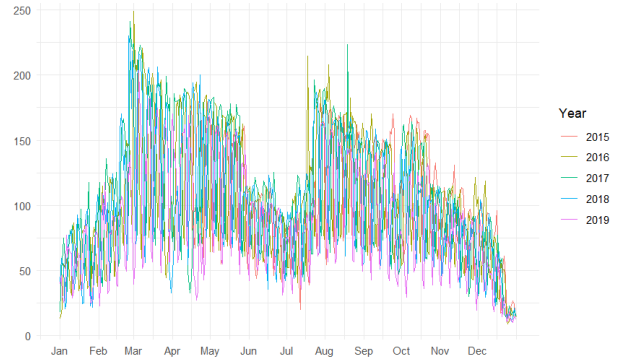

Figure 2 – Time Series Plot (Daily Average for each Year)

The yearly seasonality is evident: the time series shows the largest decrease during the summer vacation period in Australia, from December to the end of January. A second, smaller drop is also observed between June and July, reflecting lower pedestrian activity during the winter months.

Daily and Weekly Seasonality

Daily and weekly seasonalities are analysed using a single plot that shows the average pedestrian count for each hour of the day, separated by day of the week.

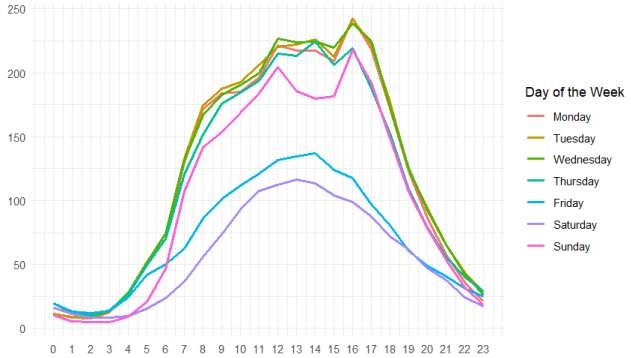

Figure 3 – TS Plot (Average of each Hour for each Day of the Week)

The plot highlights that pedestrian traffic gradually increases from 2-3 a.m., reaches a peak at 4 p.m., and then decreases rapidly after 5 p.m. The weekly seasonality is also apparent, with Fridays and Saturdays exhibiting significantly lower values compared to other days of the week.

2. Models

In this section, the forecasting models used to predict pedestrian traffic are presented. Three different approaches are considered, representing both traditional statistical methods and modern machine learning techniques. First, the ARIMA model captures linear dependencies and autocorrelations in the time series. Second, the Unobserved Components Model (UCM) decomposes the series into trend, seasonal, and other (dummy) components, allowing for a more interpretable structure. Finally, a Gradient Boosting Machine (GBM) is employed to model complex, non-linear relationships and interactions in the data.

Before introducing the individual models, a common advanced preprocessing step is applied: all three models incorporate a dummy variable to account for Australian public holidays. This binary variable takes the value of 1 on the following nationally observed holidays and 0 otherwise:

- Australia Day - January 26
- ANZAC Day - April 25
- Easter Sunday - computed with timeDate::Easter
- Easter Monday - computed with timeDate::EasterMonday
- Christmas Day - December 25
- Boxing Day - December 26

Including this holiday variable allows the models to capture potential deviations in pedestrian traffic associated with these special dates, which often correspond to reduced activity or special patterns in urban areas.

The following subsections describe each model in detail, including their additional regression features, and implementation choices. Also, each model is assessed leaving out the last 30 days of available observations, that represent the month of December, making it possible to evaluate the model both on ordinary days and on vacation days.

2.1 ARIMA Model

The ARIMA (AutoRegressive Integrated Moving Average) model is a widely used statistical approach for time series forecasting. It captures linear dependencies in the data through autoregressive and moving average components while addressing non-stationarity via differencing.

To prepare the time series for ARIMA modelling, it is necessary to make it stationary. The first step is to examine the variance using a Box-Cox plot.

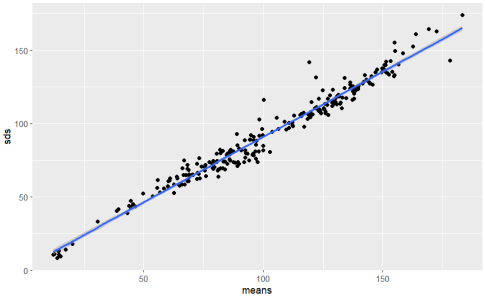

Figure 4 – Original Box-Cox Plot

Figure 4 shows the initial Box-Cox plot, which indicates that the variance is not constant, as evidenced by a strong positive correlation. A log transformation (log(y + 1)) is then attempted, but the variance remains positively correlated with the mean. The optimal Box-Cox transformation found corresponds to a lambda value of 0.15.

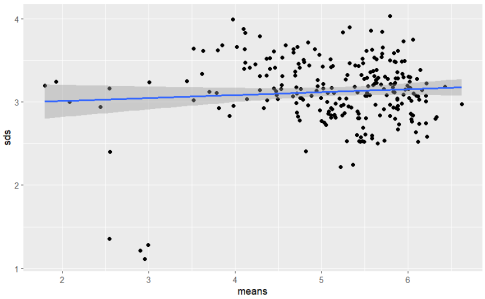

Figure 5 –Box-Cox Plot after Box-Cox Transformation

Figure 5 displays the series after applying the Box-Cox transformation with $\lambda = 0.15$, showing a stabilized variance suitable for modelling.

To account for yearly seasonality, 15 cosine-sine pairs are included as additional regressors. Furthermore, the holiday dummy variable is incorporated to capture the effect of public holidays transforming the model into an ARIMAX.

Instead of building a single model for the entire time series, separate ARIMA models are constructed for each hour of the day. To maintain consistency and reduce complexity, all hourly models use the same ARIMA parameters, which are determined based on the series corresponding to the peak hour (4 p.m.).

To decide the values for the parameters, the following process is applied:

1. Set d=1 to account for the decreasing trend in the series and D=1 for the weekly seasonality.
2. Plot the autocorrelation function (ACF) and partial autocorrelation function (PACF).
3. Identify the most prominent process (autoregressive or moving average, seasonal or not) suggested by the ACF/PACF patterns.
4. Include the corresponding parameter (p for AR, q for MA, P for SAR or Q for SMA) in the model and fit the ARIMA model to the series.
5. Evaluate the residuals and repeat steps 2-4 recursively until the residuals resemble white noise, indicating that the model has captured the underlying structure of the series.

At the beginning of the process, the ACF decreased gradually, while the PACF exhibited peaks at multiples of 7, making the initial choice of parameters challenging. However, the first PACF lag was significantly higher than the others, suggesting an AR(1) component.

Next, a Seasonal MA(1) component with a period of 7 became evident, as the ACF showed a significant value only at lag 7 and the PACF decreased gradually at multiples of 7. Finally, the ACF displayed significant values at the first three lags, while the PACF declined gradually, indicating the need for an MA(3) component.

After incorporating these components, the residuals resembled white noise, confirming that the model adequately captured the series' structure. The final model selected was therefore ARIMA(1, 1, 3)(0, 1, 1) with a period of 7 for the seasonal part.

The time series is then modelled using the selected ARIMA specification, leaving out the last 30 days of available observations. These kept-out observations are used to generate out-of-sample predictions, which are subsequently compared with the true values to evaluate the model's performance using the Mean Absolute Error (MAE).

The following plot shows the true time series alongside the predicted values for the out-of-sample observations for the ARIMA model.

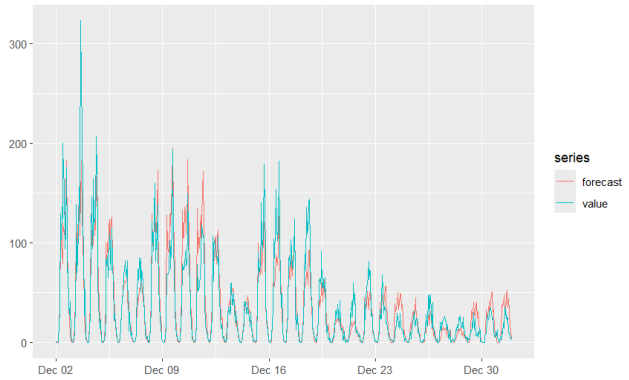

Figure 6 – ARIMA Forecast vs True Value

The model performs well, capturing the main patterns in the withheld data and achieving a Mean Absolute Error (MAE) of 12.02.

Finally, the model was retrained on the full dataset, and forecasts were generated for the unknown observations.

2.2 Unobserved Components Model

The Unobserved Components Model (UCM) is a flexible statistical framework for time series analysis that decomposes a series into interpretable components. Unlike purely autoregressive models, UCM explicitly models these underlying structures, allowing for more insight into the sources of variation in the data. To keep the predictions greater or equal to zero the log(y+1) was used for the UCM.

The components selected for this UCM are as follows:

1. A first-order stochastic trend; a second-order trend was also tested, but the estimated variance of the additional component was effectively zero, indicating that it did not contribute to explaining the series.
2. A seasonal dummy with a period of 7 was used to model the weekly seasonality.
3. A trigonometric seasonal component with a period of 365 and 15 harmonics, included to capture the yearly seasonality; other numbers of harmonics were tested, but 15 provided the best performance.

In addition to the holiday dummy, several other binary variables were created for the UCM based on empirical inspection of the model's initial results.

- Initially, the model was unable to capture the effect of the summer vacation period in Australia, (from the second half of December to the end of January). To address this, a dedicated summer vacation dummy variable was added.
- Subsequently, the model was found to significantly underpredict on Fridays and Saturdays during holidays. To correct this, two additional dummies were introduced, indicating when a holiday falls on a Friday or a Saturday.

As with the ARIMA model, separate UCMs were fitted for each hour of the day and initially evaluated on the same out-of-sample observations.

The plot that follows shows the true values versus the predicted values for these observations for the UCM.

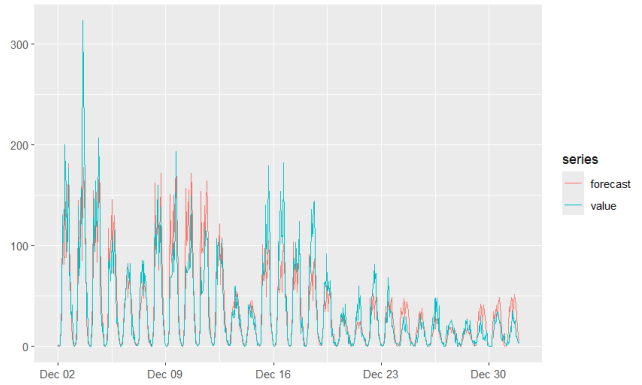

Figure 7 – UCM Forecast vs True Value

The results are very similar to those obtained with ARIMA, with the model achieving a comparable Mean Absolute Error (MAE) of 12.63.

2.3 Gradient Boosting Machine - ML Model

Gradient Boosting Machines (GBMs) are a class of machine learning algorithms that build a strong predictive model by sequentially combining multiple decision trees. Each new tree is trained to correct the errors made by the previous ones, resulting in improved predictive accuracy over a wide range of tasks. Gradient boosting has become a popular choice for regression problems because of its ability to capture complex, non-linear relationships in the data and its flexibility in handling diverse features.

In the context of time series forecasting, GBMs can effectively incorporate engineered features such as lags, calendar variables, and holiday indicators, making them suitable for modelling time series patterns. This flexibility and strong performance on structured/tabular data motivate the inclusion of a Gradient Boosting Machine as the machine learning component of this work.

The features selected to build the GBM model are the following:

- A 24-hour lag, to capture short-term temporal dependence.
- A 168-hour (one week) lag, to model weekly seasonality.
- A 365-day lag, to account for yearly seasonal effects.
- Day of the year, to encode long-term seasonal patterns.
- Day of the week, to capture systematic differences across weekdays.
- Holiday dummy variable.
- Summer vacation dummy variable.

The model hyperparameters were selected through an empirical fine-tuning procedure, using the same out-of-sample observations adopted for ARIMA and UCM as a validation set. The final configuration is:

- distribution = tdist
- n.trees = 1000
- interaction.depth = 5
- bag.fraction = 0.7

Predictions are generated using a recursive forecasting strategy. In this approach, once the model produces a prediction for the next time step, this predicted value is fed

back into the model as an input to forecast subsequent observations. The procedure is repeated iteratively until the entire forecasting horizon is covered. This method is necessary when lagged values of the target variable are used as features and future true values are not available.

With this final configuration, the GBM model performs reasonably well on non-vacation days, but it appears to underestimate the impact of the dummy variables: it tends to keep predictions too high during the vacation period, suggesting that the model does not fully capture the effect of holidays and summer breaks.

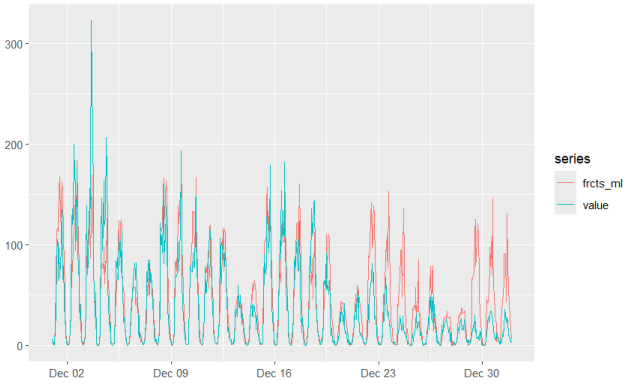

Figure 8 – GBM Forecast vs True Value

Figure 8 shows the predictions and the true value, the MAE on the out-of-sample is 18.36.

Finally, the model is trained on the full dataset, and forecasts for the previously unknown observations are generated using the recursive forecasting strategy.

2.4 Final Considerations About Predictions

Now that forecasts for the unknown values have been obtained from all three models, it is useful to take a first look at their overall behaviour and compare the magnitude and distribution of the predicted values. Simple summary statistics provide an immediate and intuitive way to highlight the main differences among the approaches:

ARIMA produces the highest forecasts, with a mean predicted value of 62.03 and a maximum of 262.92.

UCM provides intermediate predictions, with a mean of 37.27 and a maximum of 195.42.

The ML model generates the lowest forecasts, with a mean of 24.17 and a maximum of 112.72.

These statistics, while not directly informative about the true unknown values, provide useful insights into the behaviour of the models.

Conclusions

In this project, three different approaches for time series forecasting were explored and compared: two traditional models for time series (ARIMA and UCM) and a machine learning method (Gradient Boosting Machine). Each model was designed to capture different aspects of the data and to provide complementary perspectives on the forecasting problem.

The ARIMA model proved to be effective in modelling the linear structure of the series, achieving good predictive performance on the out-of-sample observations with a MAE of 12.02. The UCM achieved very similar results, with a MAE of 12.63, and offered the additional advantage of interpretability by explicitly decomposing the series into trend and seasonal components. The close performance between ARIMA and UCM suggests that both approaches are well suited for time series forecasting.

The Gradient Boosting Machine, while flexible and powerful in principle, showed weaker performance in this application, with a higher MAE of 18.36. Although it performed reasonably well on non-vacation days, it appeared to struggle in properly exploiting the information contained in the dummy variables.

Overall, the comparison shows that traditional and structural time series models (ARIMA and UCM) outperform the machine learning approach in this specific forecasting task.

However, it is important to emphasize that all these results are necessarily partial. The models have been evaluated using only a limited out-of-sample portion of the available data, while the final objective is to predict 1,439 future observations whose true values are unknown. Consequently, the real forecasting performance of the models can only be fully assessed once these values are compared to the predictions. Until then, the conclusions drawn in this work should be interpreted as indicative rather than definitive.

References

Pelagatti, M. M. (2015). Time Series Modelling with Unobserved Components. Chapman & Hall/CRC.

R Core Team (2024). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Wickham H et Al. (2019). “Welcome to the tidyverse.” Journal of Open Source Software.

Jouni Helske (2017). KFAS: Exponential Family State Space Models in R. Journal of Statistical Software.

Ryan JA, Ulrich JM (2024). xts: eXtensible Time Series. R package version 0.14.1

Ridgeway G, Developers G (2024). gbm: Generalized Boosted Regression Models. R package version 2.2.2.

Hyndman R et Al. (2025). forecast: Forecasting functions for time series and linear models. R package version 8.24.0.

OpenAI (2026). ChatGPT, GPT-5.2. Used for language revision in the report.