POLITECNICO DI TORINO

ICT IN SMART MOBILITY

LABRATORY REPORTS - GROUP 12

Report 2 - Predictions using ARIMA models

Authors
Arman MOHAMMADIGILANI s301000
Sara GHOLAMHOSSEINZADEH s301689
Nikoo ARJANG s300262

Supervisors
Marco MELLIA
Luca VASSIO

ACADEMIC YEAR 2022/23

1 Introduction

For this laboratory, the January 2018 rental time series for Berlin, Firenze, and Toronto are examined. Using Autoregressive Integrated Moving Average (ARIMA) models with proper hyperparameter tuning, the objective is to provide accurate rental price predictions. N, the number of past samples used for training, p, the number of lag observations in the Auto Regressive model, d, the number of differentiation steps required to make the model stationary, and q, the number of lag observations in the Moving Average model, are the hyperparameters utilized and analyzed in the model.

1.1 Building time series

Due to the absence of data for a few days in December 2017 as well as trends and fluctuating levels towards the end of the month, the data from January 2018 were instead used for all subsequent activities. After filtering for outliers, data from bookings lasting between 3 minutes and 3 hours, and only bookings in which the car moved, were retained.

1.2 Fitting missing data

After re-indexing the time series data frames using Datetime indexes of all January 2018 hours, it seems that just a few hours of data are missing, which is preferable. The approach utilized to fit the missing data consisted of filling in the gaps with the number of rentals from the three prior days at the same hour. The selection of this method was determined by its periodic nature. In all other cases, the subsequent days were taken into account.

1.3 Stationary check

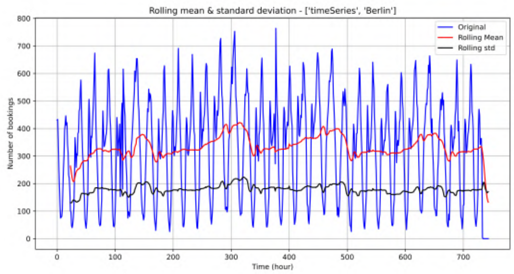
To determine whether or not to use differencing, it is necessary to check if the time series is stationary, i.e., whether its statistical features remain constant throughout time. The rolling mean and rolling standard deviation are shown with a 1-week frame to determine stationarity.

Figure 1 demonstrates that, while the results indicate that the rolling mean and standard deviation vary over time, their variance is not significant. Consequently, they are considered constant. This means that all time series are stationary, therefore there is no need for differentiation. This implies that the ARIMA model's hyperparameter d has the value 0 Thus, the ARIMA$(p, d = 0, q)$ model could be described more simply as ARMA(p, q).

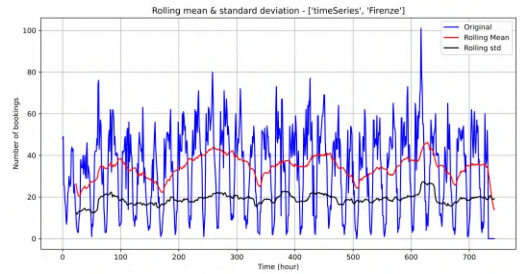
1.4 ACF and PACF computation

The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of a stationary time series are helpful for estimating the number of observations, p, to include in the model and the order of the moving average, q. These values were calculated for the time series of the three analyzed cities and are shown in Figure 2. It displays the ACF, PACF, and autocorrelation for the first 50 hours of data for the number of rentals time series.

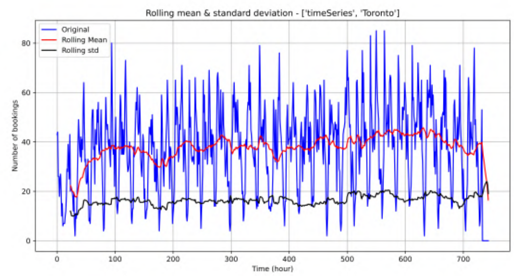
The ACF plots demonstrate a periodicity. There is a significant correlation between the rentals every 24 hours at the same time. This is fair given the cyclical nature of rentals.



(a) Berlin



(b) Firenze



(c) Toronto

Figure 1: Stationary check.

In the instance of the PACF, the curves in each graph seem to cut off at a certain location. However, the remaining numbers are not entirely insignificant.

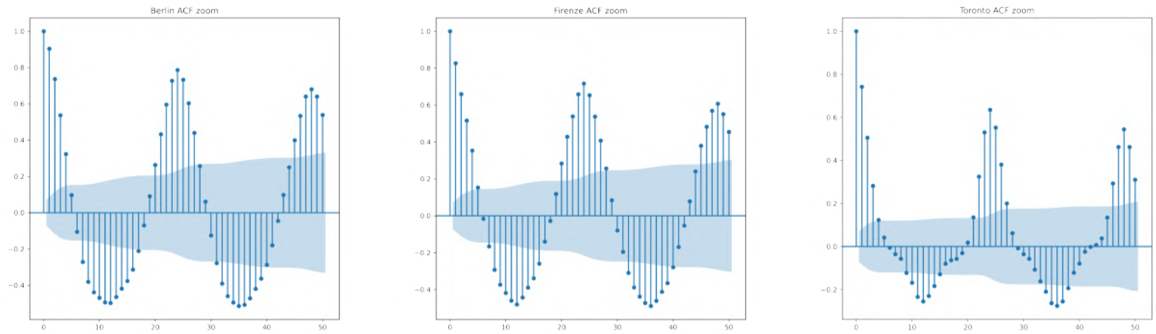
The PACF is used to determine the starting value of the hyper-parameter p. Before the curve reaches the blue zone, the number of peaks with a strong correlation is counted. Specifically, examining Figure 3a for Berlin reveals that the PACF becomes insignificant after two lags. Consequently, $p = 4$ has been selected as the initial draft value. The same reasoning was applied to Firenze and Toronto, hence $p = 1$ for Firenze and $p = 1$ for Toronto have been determined. Instead, it is more difficult to determine the hyper-parameter q from the curves. In all situations, this option is randomly assigned to the value 3.

Obviously, this technique can only be used as a first attempt at producing a reasonable ARMA model, and not the best one; hence, more analysis is necessary to appropriately adjust p and q.

1.5 Model training

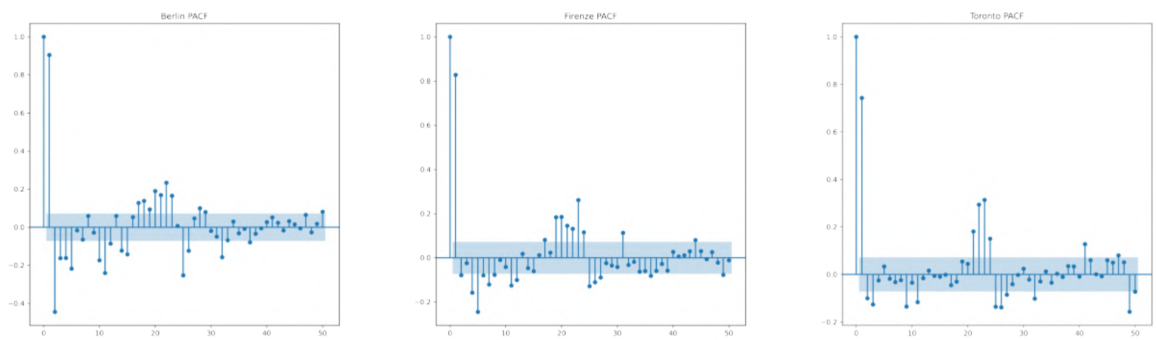
It has been determined how many samples from the past must be utilized for training and how many for testing. The walk-forward validation using either the sliding window approach or the expanding window approach has been selected. Specifically, the previous procedure included one week of training followed by one week of testing.

For walk-forward validation, a window size with a maximum time length has been chosen, and the strategy for the first window must be optimized. The model should then be trained and fitted using all available data inside the given window or preceding it. The model outputs for all accessible data for the subsequent time step should be produced once the model has been constructed and parameters defined for the given period. Now, the window must be adjusted so that all of the data through that window can be utilized for fitting and the data for the subsequent time-step could be used for testing.



(a) Berlin (b) Firenze (c) Toronto

Figure 2: ACF for different cities.



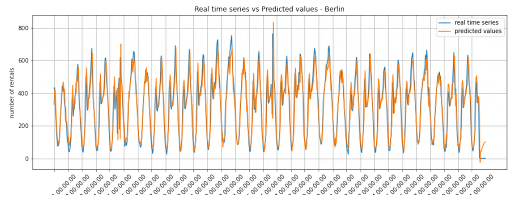
(a) Berlin (b) Firenze (c) Toronto

Figure 3: PACF for different cities.

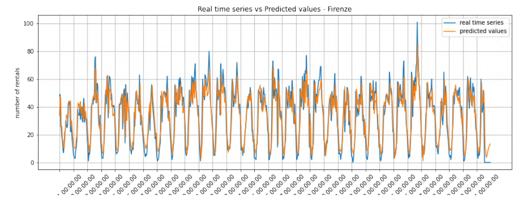
1.6 ARIMA Model training and error computation

Using the previously obtained values for N and (p, d, q), the ARIMA model has been trained and the error in the prediction has been calculated. Figure 4 shows a comparison between the original dataset and the prediction, from which it can be seen that the model yields an acceptable outcome. The training of the ARIMA model is done over 4 weeks of data. N, the number of samples for training, is then set as $N = 672$. The expanding window strategy is used, which means that test data from the past are included as part of the training set after being tested.

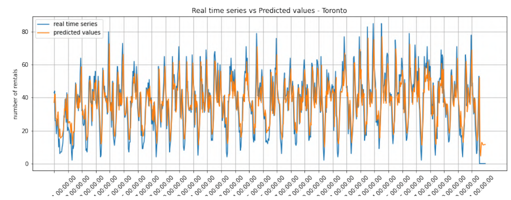
Figure 4 depicts the outcomes of the just-described method, which use the ARIMA$(3, 0, 2)$ model as a starting point for simplicity and because it is preferable to utilize low hyper-parameter values, which correlate to more complicated models.



(a) ARIMA(3,0,2) - Berlin



(b) ARIMA(3,0,2) - Firenze

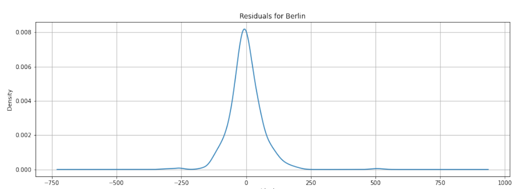


(c) ARIMA(3,0,2) - Toronto

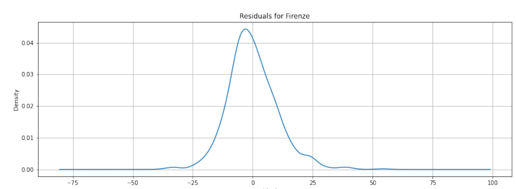
Figure 4: Predictions with initial guess for parameters p and q.

The three graphs demonstrate that the prediction curves attempt to replicate the original ones. However, there is still considerable mistake, particularly in the peaks. Comparing the cities, it is evident that Berlin's predictions seem to be the most accurate, since the orange curve deviates least from the original time series.

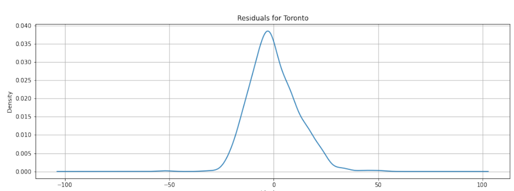
In addition, Figure 6 displays the distribution of the residuals, which shows a Gaussian behavior with a mean value of 0.



(a) Berlin



(b) Firenze



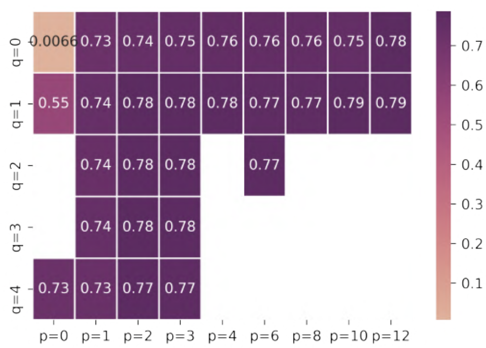
(c) Toronto

Figure 5: Distribution of residuals for different cities.

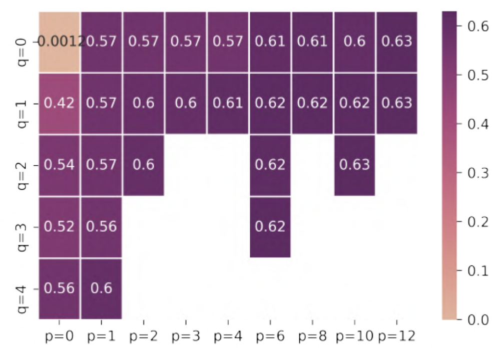
1.7 Parameters tuning

In this section, the performance of the ARIMA model will be evaluated by varying and modifying several parameters, such as p, q, the model's proper, and the size of the training dataset, N.

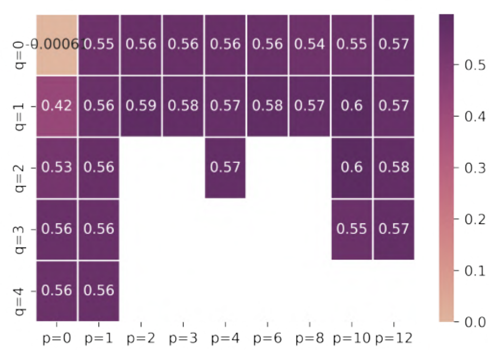
The subsequent step focused on determining the effect of altering hyper-parameters p and q while maintaining the training set, test set, and learning approach constant. A grid search was conducted by allowing q to range between 0 and 3 and p to take on the values $[0, 1, 2, 3, 4, 6, 8, 10, 12]$. As illustrated in Figure 6, the Mean Squared Error (MSE) was calculated and given for each of the resulting models.



(a) Berlin



(b) Firenze



(c) Toronto

Figure 6: Parameter tuning using heatmap for different cities.

Performing a grid search on parameters p and q, maintaining $d = 0$ since time series is stationary. For specific parameters, the calculation of the ARIMA model fails. White boxes represent missing elements due to problems with each specific combination, such as not locating the best r^2 score or best MSE in comparison to the previous q fitting in the ARIMA model. Increasing the amount of p the calculation begins requiring a lot of time, without any significant improvement in the quality of the prediction.

The optimal parameters, determined by maximizing the r^2 score and minimizing the mean squared error, are $p = 12$ and $q = 1$ for Berlin, $p = 10$ and $q = 2$ for Firenze, and $p = 10$ and $q = 1$

for Toronto. For testing reasons, it was chosen to adopt the best parameters acquired by the grid search, but in a real-world application, it might be advantageous to maintain p as low as feasible in order to save computation time without severely compromising the model's performance.

In Berlin, the MSE is 3303.25, and it cannot attain a r^2 value larger than 0.78 using the grid search on this dataset. The best MSE in Firenze is 149.17. Using this dataset, the same grid search identified a set of parameters that enables to get a r^2 value of 0.63. In addition, the best MSE in Toronto is 96.67 and r^2 equals 0.55. When calculating the relative error between the number of rentals and the expected value for these cities, Toronto, which has the biggest number of rentals, has the lowest error. Firenze also has the lowest reported number of rentals and the greatest relative error value.

With the best model's hyper-parameters in hand, the next step was to evaluate the model's accuracy by modifying the learning approach and the size of the training window. The test was conducted with three different training window sizes, while the test size remained constant at the same number of data.

The MSE provided by the two learning strategies is comparable for the city of Firenze. Also, the least error is acquired while training for one week, while the largest error is obtained when training for two weeks. The MSE provided by the two learning strategies for Toronto is similar. The minimum error is acquired with a training duration of three weeks, while the largest error is obtained with a training duration of one week. The behavior as a whole demonstrates that the error rate decreases as the number of training data points increases.