**Group 10: Smart Water B**

Abhay Jhala, Dhruvi Darji, Iustin Toader, Kelsie Miller & Rachael O’Shea

Department of Decision Sciences, Drexel University

BSAN 460-001: Business Analytics Senior Project

Dr. Chris Gaffney

December 7, 2022

**Problem Background, Description, and Goal**

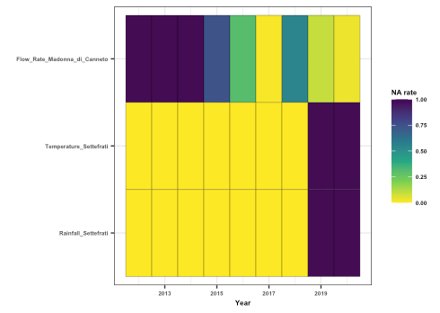
Acea Group is one of the leading Italian multi-utility operators. They manage and develop water and electricity networks and environmental services, supplying nearly nine million inhabitants. The problem is that Acea Group struggles with forecasting the water level in different waterbodies. The issue to be addressed is that water bodies need to be preserved to handle daily consumption. It is important to predict the most efficient water availability so that water is available for drinking use. The goal of our project is to create models that can predict the flow rate in the Madonna and Lupa water springs.

**Data Description**

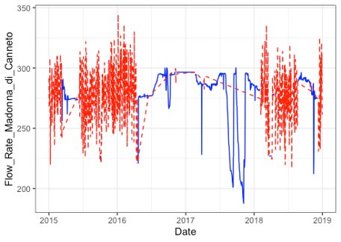
The data provided contains four different types of waterbodies including water springs, lakes, aquifers, and rivers. Looking further at water springs, there is data on three different springs: Amiata, Lupa, and Madonna. We focused on water spring Madonna and applied methods to Lupa as well to see how the methods performed on other springs. Each of these datasets are completely independent from each other, with nearly 14,000+ observations across the three springs alone. The feature that our models are going to be forecasting is the flow rate.

**Problems Faced and Steps Taken**

One of the biggest issues faced with the data was the high prevalence of missing values. What’s more, we originally planned to use an external dataset for imputing temperature and rainfall data, but the data was not in a compatible format for our analysis. As such, we decided to exclude this step from our analysis.



For the Madonna water spring, 100% of values were missing for flow rate in years 2012, 2013, and 2014, so these years were dropped. 100% of the values were missing for the temperature and rainfall variables in the years 2019 and 2020. Imputation was not suitable for our predictor variables, so we dropped these two years from the dataset. We imputed all remaining missing values using the Amelia package, adapted for time-series. The graph below illustrates the data before and after the missing data was imputed. The blue line represents the original flow rate values, and the red line shows the imputed flow rate values.



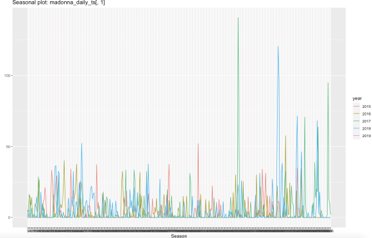
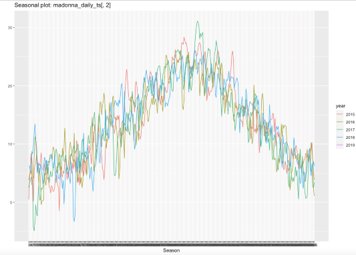
**Method, Results, & Out of Sample Analysis (ARIMA)**

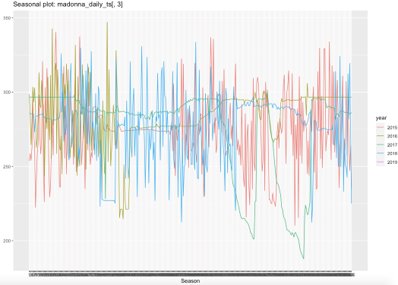
In order to gauge the true performance of our models, as well as to accommodate for various levels of comparison and interests of users, we have decided to expand our analysis on 3 levels of data aggregation: daily, weekly, and monthly. Although the paper will only discuss daily and weekly findings, primarily due to the very similar values of weekly and monthly data, the accompanying code contains all 3 levels of aggregation for each model implemented.

During the preliminary stages of our project, we used an 80/20 training to testing split. However, we soon realized that this corresponded to an unrealistic forecasting time-horizon, especially given the difficulty of predicting weather data. We saw an RMSE of around 15 for the training data and around 22 for the testing data in both univariate and multivariate ARIMA. To combat this issue, we adapted our approach to only forecast 90-days ahead. We have similarly adapted our weekly and monthly aggregation models to maintain the same ratio corresponding to a 90-day split – 12 weeks and 3 months. One final step for preprocessing was converting our data to a time-series format, adapting each aggregation to its respective frequency setting: 365 for daily, 52 for weekly, 12 for monthly.

The first method we focused on was Autoregressive Integrated Moving Average (ARIMA). ARIMA is a time series forecasting model which will be used to predict the water flow rates in the future. It is a class of models that forecasts using its own past values including lag values and lagged forecast errors. The Auto Regressive term is used for lag values to forecast where the dependent variable depends on past values of itself. The Integrated term reflects the differencing steps required to make our data stationary, so instead of predicting the time series itself, we will predict the differences of the series. Lastly, the Moving Average term uses lagged forecast errors to forecast results and the model works by analyzing how wrong you were in predicting values for previous time periods to make a better estimate for the current time period. Time Series data is considered stationary if it contains constant mean, constant variance, and covariance that is independent of time. ARIMA consists of three parameters: p, q, and d. P is the number of autoregressive terms, which is the number of lags Y used as predictors. D is the number of non-seasonal differences needed for stationarity. Q is the number of lagged forecast errors.

Another important factor to look at in our model is the seasonality of the data. The charts below show the seasonality of the temperature and rainfall variables. These charts show the variable’s values over a year of time. Each line on the chart represents a different year in the data. A consistent pattern seen between these lines means that there is seasonality in the data. When plotting temperature against seasons in different years (left), seasonality is seen for temperature. Although seasonality can be spotted in our rainfall regressor (right), it is not nearly as pronounced as the temperature’s.





However, when looking at our outcome variable (flow rate), there is no trend that each year follows. Seasonality is not carrying over between our regressors and target variables, which is likely what leads to similar results for univariate and multivariate ARIMA models.

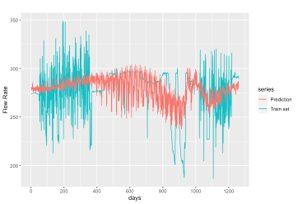
We started off by building both a univariate and multivariate ARIMA model using daily data for the Madonna water spring. The univariate model only considers past flow rates to predict future flow rates, while the multivariate model also considers the regressors. The RMSE of the univariate model on the training set was 16.05 and the RMSE of the univariate model on the testing set was 15.39. The RMSE of the multivariate model on the training set was 16.03 and the RMSE of the multivariate model on the testing set was 15.64. Multivariate ARIMA had a slightly better RMSE in the training set compared to the univariate model, but they are both extremely similar. The univariate model had a better RMSE for out of sample data.

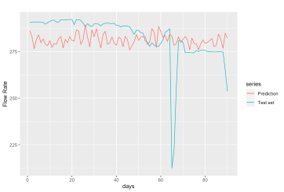
We also built both a univariate and multivariate ARIMA model using weekly data for the Madonna water spring. The RMSE of the univariate model on the training set was 13.37 and the RMSE of the univariate model on the testing set was 7.78. The RMSE of the multivariate model on the training set was 13.32 and the RMSE of the multivariate model on the testing set was 9.25. Similarly to the daily observations, the multivariate model had a better RMSE in the training set, but both were very similar. The univariate model had a better RMSE for the out of sample data. Overall, we found that the weekly aggregations had better RMSEs., which should not be surprising given the lower volume of predicted values.

**Method, Results, & Out of Sample Analysis (LSTM)**

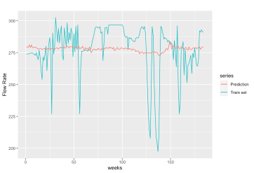
The second method that we focused on was Long Short-Term Memory (LSTM). LSTM is a type of recurrent neural network that is widely used for time series forecasting. An LSTM network helps to overcome gradient problems and makes it possible to capture long-term dependencies. It also can recognize patterns over the length of the series.

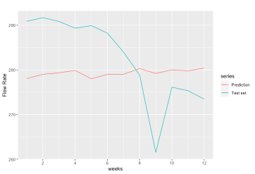
We created an LSTM model built on daily data from the Madonna water spring. First, we added a delta value equal to the absolute minimum value encountered in the target dataset to avoid negative square roots. Next, the data was normalized by centering, scaling, and removing outliers. The transformations specification was saved to reinvert predicted values and compare. For the specifications of the model, we used a batch size of 30 days, which must be divisible with our desired test length of 90 days while maintaining the suggested ratio. The weekly models have been adapted to a batch size of 4 and test length of 12 weeks. Our dependent variable is flow rate and feature is 90-day lag value of flow rate. The number of time steps is 1 and the LSTM layer is 50 units (also known as neurons). Dense layer connects the neurons of each layer to the next and this was 1 unit. We used 90 epochs, equal to the length of our testing set. The loss function was mean squared error, and the optimizer was Keras’ native “adam”.



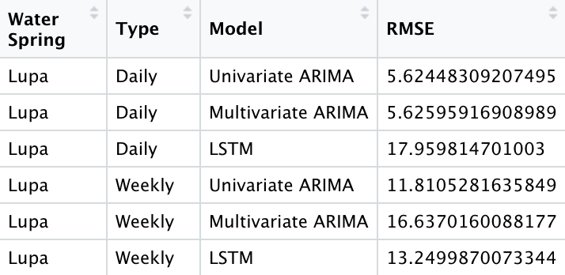


Our LSTM results for daily data in the Madonna water spring yielded an RMSE of 23.45 in the training set (top) and 12.94 in the testing set (bottom). The graph for the testing set shows one large outlier. However, when looking at this value in the context of the entire data set, it does not fall under the requirements for removal. Our LSTM results for weekly data in the Madonna water spring yielded an RMSE of 19.45 in the training set and 9.99 in the testing set.

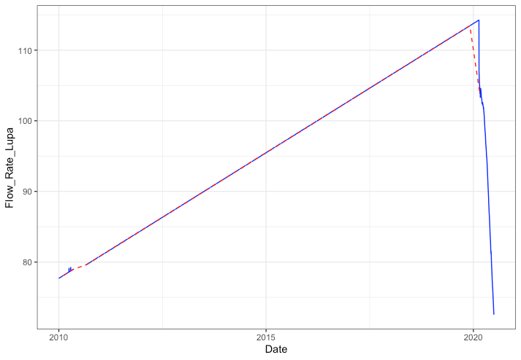




The chart below shows an overall summary of our testing results. From this table, we can see that LSTM generally performed better than ARIMA on daily data. However, ARIMA outperforms LSTM when it comes to weekly data.

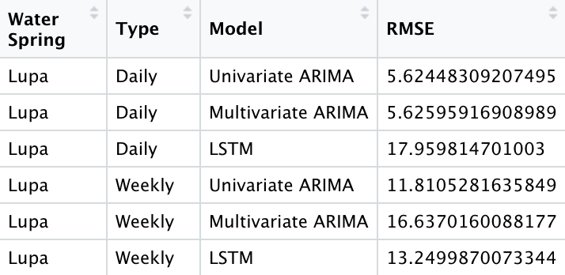


**Lupa: How do our methods transfer to another water spring?**



For our analysis of Lupa, we replicated all the steps took for our earlier Madonna model. The only difference is the changing of our forecasting horizon to 30 days in order to account for the sudden drop in flow rate described in the next paragraph. Our motivation for this sizable decrease was to give our model more training data to adapt to the shift in flow rate dynamics. We also decided to use time-series linear interpolation instead of Amelia in order to account for the linearity of our flow rate.

The Lupa water spring showed a much different behavior than Madonna. The flow rate for this water spring followed a linear pattern, with a sudden drop at the start of 2020. This linear pattern indicates that this is a man-made water spring with a monitored flow rate, which means our previous methods of ARIMA and LSTM don’t make as much sense for this data. Still, we wanted to test the adaptability of our models to unforeseen changes in data patterns. The ARIMA and LSTM RMSE values were better because a linear data set is much easier to predict.



**Project Conclusions**

All in all, we believe that the LSTM approach has the most potential for future implementations. When it comes to the most granular level of data, we can clearly see that it outperforms more generalized models such as ARIMA. This should come as no surprise, since Neural Networks typically scale with large amounts of data, and daily aggregations truly start to illustrate the benefit of implementing more complex frameworks of analysis.

With that being said, the complexity and power of LSTM Neural Networks could not account for the increase in performance generated by higher levels of aggregation and lower number of predicted values. The weekly aggregation provided the best results out of the three and univariate ARIMA was better suited for this purpose as illustrated by the summary statistics.

**Recommendations/Next Steps**

For our next steps, we would want to enhance our LSTM models through hyperparameter tuning and expand by creating an LSTM model for the Amiata water spring, as well as the other different types of waterbodies. It would also be a good idea to dedicate more time towards integrating external data sources that can help provide more accurate flow rate forecasting using temperature and rainfall data.

**Sources**

Our model implementations have been inspired by code from the following Kaggle notebooks:

<https://www.kaggle.com/code/marcomarchetti/acea-smart-water-baseline-models-comparison/notebook#5-LSTM>

<https://www.kaggle.com/code/vlarmet/a-super-learner-for-water-availability-forecasting>