Universität St. Gallen

Hochschule für Wirtschafts-,

Rechts- und Sozialwissenschaften

Forecasting the Water Level of Lake Zurich

Data Science Fundamentals

Johannes Binswanger & Lyudmila Grigoryeva

Group 13

Magalì Alluisetti, Leo di Luzio, Bryan Kaelin

Table of Contents

1. Introduction

The protection and preservation of our planet's water resources are part of humanity's greatest challenges. As the consequences of climate change ramify steadily and the population grows exponentially, freshwater resources are placed under increasing stress (OECD, 2016, p. 1). Thus, extreme climatic conditions and growing water demands threaten the integrity of lakes, rivers and aquifers. According to the OECD (2016, pp. 10-11), in 2015 the economic losses related to improper water distribution, flooding damage and water insecurity amounted to nearly USD 475 billion. If counteractions were not to be taken, these numbers would continue to increase.

This situation is also particularly critical for Switzerland, which is defined as the "water tower of Europe" (OECD, 2015, p. 1). In fact, 5 percent of Europe's water resources are located on Swiss territory, from which 22 percent is regularly deployed for domestic use (OECD, 2015, p. 2). Water shortages could therefore affect the lives of thousands of people.

The need to monitor and find immediate solutions to cope with this pressing issue made the topic ideal to be tackled with machine learning. More specifically, this project will be focused on the prediction of water levels of Lake Zurich. This, besides being one of the largest lakes in Switzerland, is of particular interest, since it accounts to 70% of the water distributed to the city of Zurich and the 67 surrounding municipalities (Kanton Zürich, 2022).

After gathering the relevant data from various sources ranging from 2012 to 2019, we selected the most important variables and merged the different datasets. After extensive cleaning work, we tested several univariate models such as Naive, Mean, ARIMA and SARIMA. However, in order to model the water level based on the other variables, we also used the multivariate VAR model. In addition, with the purpose of comparing our models with others that did not include any aspects concerning time series, we included the XGBoost model. All models are trained, validated and then used to make predictions for evaluating the models´ performance and selecting the best one.

2. Data pre-processing and data-cleaning

We gathered water-related data on lake of Zurich and its most relevant rivers from the hydrological department of the Swiss Confederation. Furthermore, we retrieved data concerning the suction tension and weather from the official website of the *Amt für Wirtschaft und Natur – Fachstelle Bodenschutz* of the Canton of Zurich and *opendata.swiss*, respectively. We also received measurements about solar radiation and evaporation from *MeteoSchweiz*.

In order to work with the different datasets, we first had to standardize the index of each of them to CET datetime format. During the merging process we selected the relevant variables and dealt with missing values either by dropping or imputing them.

Initially, most of our data had an hourly format that we subsequently transformed into weekly and monthly data. Because the hourly and daily changes in the water level are only incremental, it made sense to down-sample the data into a weekly and monthly format to observe some noticeable variations.

3. Methodology

The observations of the data – namely the water, weather, and soil data – are allocatable to one specific date and the values constitute a series that evolves over time and depends on previous values. Thus, we primarily focused on univariate and multivariate time series models. However, we also made use of one non-time series model in the form of a boosted tree.

When seasonally decomposing the data, we observed a yearly seasonal pattern as well as a yearly periodic trend. For the time series models we decided to split the final year 2019 from the rest of the data as a final, unseen test set for our trained and validated models. We used data from 2017 to the end of 2018 for model validation using the walking forward technique.

3.1 Basic models

In order to have a benchmark to compare our more advanced approaches to, we tested our weekly and monthly datasets with two simple univariate models: naive and mean. While the first method implies that the value of the forecast is equal to the previous observation, the second one sets the forecast as equal to the mean of past data (Hyndman & Athanasopoulos, 2018).

3.2 Univariate models

The time series models from the ARMA family are among the most common ones for time series predictions (Dettling, 2020). We opted for the ARIMA and SARIMA models in order to deal with non-stationary data. Furthermore, we could compare whether the usage of a seasonal component (SARIMA-Model) would result in better predictions.

Initially, we validated our models using a simple single train-test-split, but we abandoned this validation method in favor of the walking forward one. Using this validation technique renders not only more robust models but is also more likely to be used in practice, since models are continuously updated with new data (Brownlee, 2016).

We proceeded by fitting ARIMA and SARIMA models to both monthly and weekly data, using data from 2012 to the end of 2016 for training the model and data from 2017 to 2018 to validate and retrain the model after each single step using the walking forward validation technique. We approximated our preliminary sets of parameters by analyzing the *Partial Autocorrelation Function* (PACF) and the *Augmented Dickey-Fuller-Test* (ADF). We the further optimized our order and seasonal-order parameters *p,d,q* and *P,D,Q,s* by minimizing the error measures MAE, MSE, MAPE as well as the *Akaike Information Criterion* (AIC).

3.3 Multivariate models

While univariate models can only predict one variable based on the historic data, the multivariate approach integrates multiple ones. In our case, we opted for the Vector Autoregressive model (VAR), which permitted us to forecast the water level by considering also the reciprocal influence that the variables in our dataset might have (Hyndman & Athanasopoulos, 2018).

Before carrying out a VAR analysis, we had to understand the nature of our time series. First, we executed the Johansen’s Co-Integration Test, which helped us understand whether our time series had a significant statistical relationship. Secondly, we differenced our data in order to make it stationary, since VAR cannot handle non-stationary time series. Following the same procedure of the univariate models, we separated our data into training, validation and test sets. We fitted the model using the training set and improved its performance by changing the lag order.

We also tried to reduce the number of variables by using the Granger’s causality test, where we determined those with the biggest influence on the water level and dropped the other ones.

3.4 XGBoost model

We also used the *extreme gradient boost model* (XGBoost) to predict the water level of the lake. With this approach, we used our features to predict a target variable. Although this model is not meant for modeling time series problems, it is nevertheless applicable to them.

As we were dealing with a time series problem, we could not use a classical k-fold approach for cross-validation as it ignores the time-series aspects. If we were to use it, the model could look at future values to make predictions, which is in practice not possible.

Our first approach consisted in conducting a single train-test-split to train and validate our model. We then improved our approach using multiple train-test-splits, in total we used five splits to sequentially train and validate the model. We then optimized our model hyperparameters by trying to minimize the mean errors. Finally, we used our test set to assess whether our model performs well on unseen data.

4. Results

When testing the ARIMA and SARIMA models with the final, unseen dataset from 2019 we had two approaches. The first one was to forecast applying a walking forward method and the second one to forecast the entire length of the test set with a frozen model, so no updating after each step. This second approach simulates a long-term forecast where one has no data available to continuously feed the existing model, and as expected this type of forecasting performed significantly worse than applying walking forward. Thus, we are only displaying the test results for the walking forward method in the table below. While for the monthly predictions the SARIMA model has a clear performance advantage compared to the ARIMA model, the ARIMA and SARIMA models perform equally well when forecasting weekly data.

While the VAR model should have helped predicting the water level considering also external variables, the results were not as comparably good as the ones obtained with univariate models. This might be due to rounding error correlated with the inversion of the differenced data. The selection of fewer variables that might have greater influence on the water level did also not lead to any improvement in the error measures.

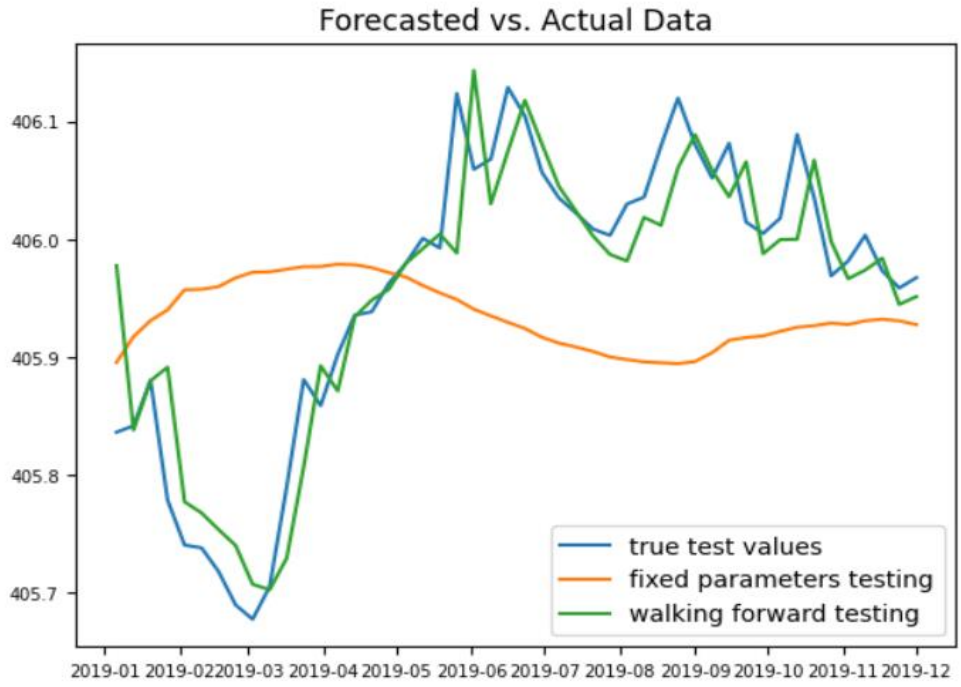
For the testing of the XGBoost model, we predicted our target variable using data from the features of the test set. Our results are shown in table 1. What we conclude from these results is that our model performs largely better than the mean and naive method, moderately better than the VAR but performs slightly worse compared to the ARIMA and SARIMA on weekly data but it beats ARIMA on monthly data. One possible reason could be that the other models were tested using walking-forward validation while it was not the case for XGBoost. Another point is that lags were not used as features in our model which could have potentially improved our results and the model was trained and predicted on non-differentiated data.

Overall, it can be said that for both monthly and weekly prediction all our models performed better than the mean except for the VAR applying the Granger´s causality. When predicting the weekly data, the SARIMA model performs the best, having the lowest error measures. Concerning monthly prediction, the SARIMA model is performing the best, however it should be noted that also the XGBoost model performs quite well compared to the other ones.

Table 1 Our predictions on unseen data

	MAE	**MSE**	**MAPE**	**R-squared**
Weekly Predictions				
Naive	0.0995	0.0153	0.0245	-0.0004
Mean	0.1121	0.0164	0.0276	-0.0682
ARIMA	0.0345	0.0022	0.0085	0.8578
SARIMA	0.0341	0.0023	0.0084	0.8513
VAR	0.0564	0.0056	0.0139	0.6334
VAR with Granger’s causality	0.0665	0.0076	0.0164	0.5071
XGBoost	0.0489	0.0032	0.0121	0.7906
Monthly Predictions				
Naive	0.1230	0.0172	0.0304	-0.2137
Mean	0.1121	0.0151	0.0276	-0.0681
ARIMA	0.0583	0.0057	0.0144	0.5990
SARIMA	0.0385	0.0029	0.0095	0.7923
VAR	0.0569	0.0048	0.0140	0.6631
VAR with Granger’s causality	0.2926	0.0936	0.0721	-5.6068
XGBoost	0.0441	0.0023	0.0109	0.8381

Figure 1: Final Test for SARIMA with weekly data



5. Conclusion

In this project we gather data from multiple stations and sources which was then preprocessed, cleaned and merged into weekly and monthly datasets. We then tackled our biggest challenge, which consisted of forecasting with time series by selecting three different approaches to predict the water level of Lake Zurich. This was done by selecting simple models as a reference point for more sophisticated models such as the ARIMA, SARIMA and VAR. Additionally we also fitted a non-time-series model, namely the XGBoost.

Our best performing models are the univariate models from the ARIMA family, with the SARIMA model performing the best overall. The error measures were worse for the VAR model which however could be due to the inversion of the differenced data. Performing Variable selection for the VAR model even worsened its forecast performance. The XGBoost models' performance comes close to the one of the ARIMA model and is even better in the case of monthly predictions.

An interesting thing that could be done in the future could be to try our models on other lakes to see how they perform. It might be difficult to find the same features in other lakes, but with the univariate models, we don't need them. We might find out that our models are only applicable in that environment.

6. References

- Brownlee, J. (2016). How To Backtest Machine Learning Models for Time Series Forecasting. Retrieved from: https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/
- Dettling, M. (2020). Applied Time Series Analysis. retrieved from: https://ethz.ch/content/dam/ethz/special-interest/math/statistics/sfs/Education/Advanced%20Studies%20in%20Applied%20Statistics/course-material-1921/Zeitreihen/ATSA_Script_v200504.pdf
- Hyndman, R. J. & Athanasopoulos, G. (2018). Forecasting: principles and practice (2nd edition). OTexts: Melbourne. Retrieved from: https://otexts.com/fpp2/index.html
- Kanton Zürich, (2022). Wasserversogung. Retrieved from: https://www.stadt-zuerich.ch/dib/de/index/wasserversorgung.html
- OECD. (2015). *Water resources allocation.* Retrieved from https://www.oecd.org/switzerland/Water-Resources-Allocation-Switzerland.pdf
- OECD. (2016). *Water, growth and finance*. Retrieved from: https://www.oecd.org/environment/resources/Water-Growth-and-Finance-policy-perspectives.pdf