Forecasting the Minimum Cost of Monthly Survival During the Syrian Civil War

Aneesh Dahiya*, Jaco Fuchs*, Christoph Glanzer*, Julia Ortheden*

Motivation and Summary

It is becoming more and more of a common practice for aid organizations and NGOs to deliver cash-based assistance to war-town countries such as Syria. On average, it takes three to six weeks to deliver aid to the local population. It is therefore crucial to forecast the minimum cost of monthly survival (SMEB, "survival minimum expenditure basket") to ensure that the aid delivered is neither too high nor too low. The goal of this project is to develop a model to forecast the SMEB price.

When predicting a single month into the future, the baseline model which simply uses the value from the previous month as a prediction for the next month outperforms all of the models which we have tested. We believe that this is due to the high volatility of the SMEB price. For a forecast of two or more months into the future, we present a variant of an `ARIMA` model which outperforms the baseline model.

This project is joint work with IMPACT Initiatives as part of the "Hack4Good" project of the Analytics Club at ETH Zürich.

Structure of the Data

A file in the .xlsx format containing the monthly price development of various products in all governorates, districts and subdistricts from February 2017 to August 2019 is provided. Unfortunately, around 60% of the data is missing.

From this data the SMEB prices in Syrian Pounds are extracted on all available levels of regional granularity. The decision to use the SMEB price in Syrian pounds was taken together with the other teams since the currency exchange rate between the local currency and USD does not have to be taken into account.

To handle the missing data, we use the following quick-fix: If a missing data point lies in between given data, linear interpolation is used and otherwise, the data point is excluded. We have chosen this simple approach as a different team was given the task of finding more sophisticated approaches towards solving the imputation problem.

Model Exploration

We have focused on implementing three types of models to generate predictions for future SMEB prices: Linear (auto-)regression, time-series based models and neural networks. In order to compare models with each other we have used a simple common evaluation metric based on a time-series adapted cross validation procedure: A model is trained on all datapoints up to February 2018. Then, the next month (March 2018) is predicted. Next, the model is retrained on all previous datapoints, including the true value of March 2018, and once again the next month (April 2018) is predicted. This procedure is repeated consecutively. Finally, the relative mean squared error between all predicted and true values is calculated and averaged. Table 1 shows a quick overview for the district level.[1]

*Authors are listed in alphabetical order as they contributed equally.

[1] We want to emphasize that these values were generated using the SMEB prices without water as this data is subject to a lesser number of missing values. In the code provided, we use the SMEB prices with water in Syrian

Model	Rel. MSE
Baseline Model	1,79
Linear Momentum	4,34
Savitzky-Golay extrapolation	3,67
ElasticNetCV	5,97
RANSACRegressor	6,01
`LOESS+ARMA(p,q)`	3,34
Lasso	2,66
RidgeCV	2,87
Local `LOESS` + `ARMA(p,q)`	2,65
Holt-Winters	2,07

Table 1: MSE values for various models on a district level.

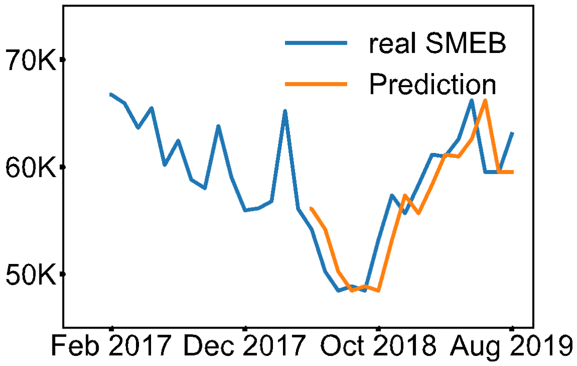

Figure 1: Due to the high volatility of the SMEB price, the baseline model serves as a good predictor for the prediction of one month into the future.

Time Series Models

Since there was not enough data to observe seasonal effects with sufficient evidence, we fit a trend without any seasonal behavior. The models which performed best were a combination between `LOESS` (using a linear trend) and `ARMA`(p, q), and a non-seasonal Holt-Winters model. The latter, however, is a very simple exponential smoothing approach which is closely related to the baseline model.

Linear Models

Regarding linear models, we have taken three different approaches.

- The first type of model is autoregressive and similar to `ARMA(p,q)`: When predicting the next month, the previous months are used as input features. The window length is optimized via cross validation on the time series at each instance of time. The difference between this model and `ARMA(p,q)` is that in the latter, we first fit a trend and then an autoregressive, moving average model (`ARMA`) to the remainder while in the former, we assume that the time series is already a stationary process and fit an autoregressive model directly.

- The second type of linear model assumes that one of the products in the SMEB sets a *trend* for the other products in the SMEB. Thus, the trendsetter's price or its last relative change is included as an additional input feature. Unfortunately, we could not identify any product as a trendsetter with sufficient evidence.

- The third type of linear model assumes that some districts are trendsetters for the SMEB price. We have identified two districts which apparently set the trend for the others, as

pounds, which is why the results may differ.

including their SMEB price as input feature leads to a slight improvement on the common evaluation metric.

To further investigate whether this is a random effect or not, we conducted the following experiment: (1) We randomly generate artificial SMEB prices. (2) We create a linear (autoregressive) model, where the "real" district data is used together with the randomly generated data as input features. (3) The model is then separately trained on all real districts as responses (one by one) using `LASSO`. (4) Subsequently, the weight which the model puts on the real features is analyzed in comparison to the weight of the artificial features. As the graph below indicates, the model seems to prefer the artificially created districts as predictors over the real district data. This implies that the observed trendsetting effect of the two districts likely comes from a random effect.

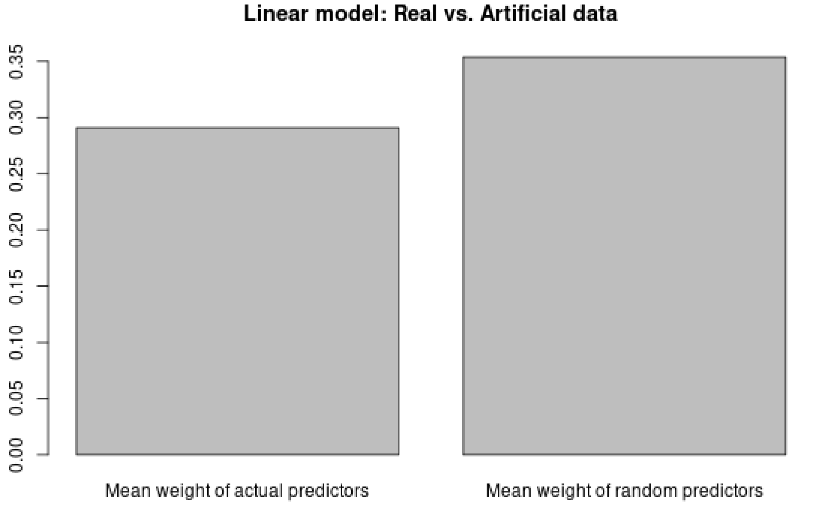

Figure 2: An experiment indicates that the predictive power of two districts as "trendsetters" which we have observed is likely a random effect.

Neural Networks

Unfortunately, most likely due to the low amount of available training points, the performance of neural network approaches was rather poor when compared to the other models.

Predicting multiple months into the future

Based on our testing data, we have observed that when predicting multiple steps into the future, our `LOESS`+`ARMA`(p, q) model outperforms the baseline model.[2] As described previously, the model first fits a linear trend using the `LOESS` method. Then, an `ARMA`(p, q) model is fitted to the remainder, with an automated choice of hyper-parameters p and q. The reason we believe that this model outperforms the baseline model for multiple steps into the future is the following: While there exists a clear trend, the price data is still highly volatile. Starting at a prediction of at least 2 months into the future, the trend prediction kicks in and the trend becomes more significant when compared to the seemingly random volatility of the prices. Since our model uses `LOESS` to fit a trend, it performs better after a few months. However, as the trend itself changes over time, the prediction worsenes again significantly when predicting more than 4 months into the future. Figure 3 shows a prediction of 4 months into the future.

Conclusion

Due to the high volatility of the data, predicting the SMEB price one month into the future is difficult and the best results are obtained by simply predicting the previous month's SMEB

[2] For a prediction of $2-4$ months into the future.

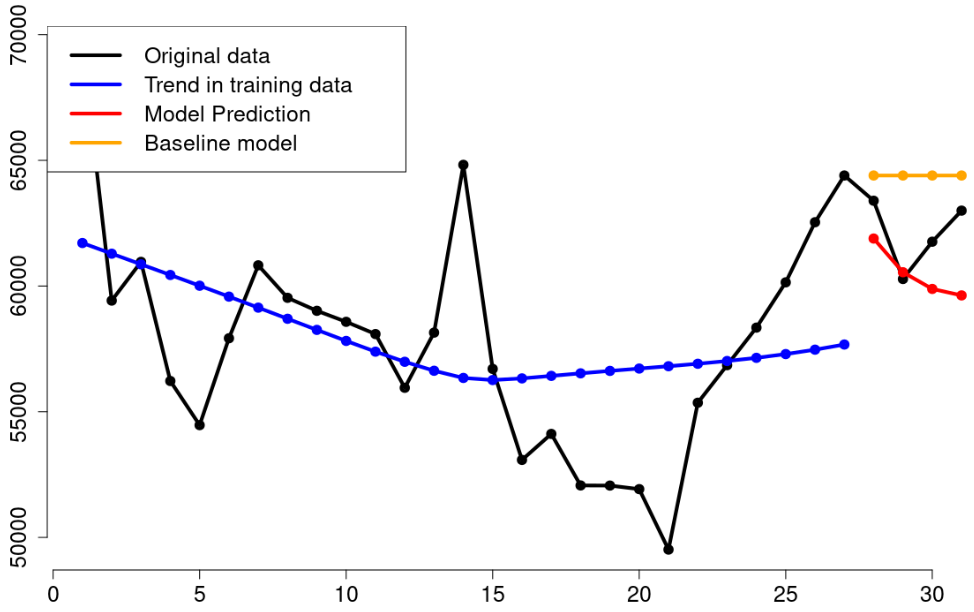

Figure 3: A prediction of four months into the future using the LOESS+ARMA(p, q)-model. Here, the model successfully picked up on the decreasing price in the first two predicted months. The baseline model is added for comparison.

price. This outperforms all other models which we have tried to fit to the data. For predicting multiple months into the future, certain time series models are able to pick up on the trend of the price and outperform the baseline model.

Future Recommendations

We suggest to look up external events which could explain the sudden peaks and otherwise seemly random volatility of the SMEB price. Including such features into the model could significantly improve the predictive performance.

Code

Aside from the data-cleaning script which we have used, we provide the script which performs the experiment described in Figure 2. Furthermore, we have added the code to evaluate the predictive power of the LOESS+ARMA(p, q) model as well as a function which allows to make custom predictions with this model. We refer to the README-files for more information on how to run the code. A note on the data which we use: The current implementation of the code uses a file which was given to us by the Hack4Good imputation team. Details regarding this file can be found in the README files.

Acknowledgments

We want to thank the Hack4Good organizational team and IMPACT Initiatives for their efforts and for making this project possible. Furthermore, we want to thank our mentor Anastasia Pentina from the Swiss Data Science Center for her help and advise regarding our project.