Final Report - Prediction Team

Aneesh Dahiya, Jaco Fuchs, Christoph Glanzer, Julia Ortheden

Abstract

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 critical 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 our project is to develop a model to forecast the SMEB price. It turns out that the baseline model which simply uses the value from the previous month as a prediction for the upcoming month outperforms all of the models which we have tested. We believe that this is due to the very high volatility of the SMEB price. For a forecast of several months into the future, we present a variant of an `ARIMA` model. This project is joint work with IMPACT Initiatives as part of the "Hack4Good" project of the Analytics Club at ETH Zürich.

Data Extraction

A file in the .xlsx format with the complete price table for all products, governorates, districts, sub-districts and months is provided. From this data the SMEB prices in Syrian Pounds are extracted on all available levels of regional granularity and stored in the .csv format. The decision to use the SMEB price in Syrian pounds was taken together with the Hack4Good external factors team since the currency exchange rate between the local currency and USD does not have to be taken into account. As a quick-fix to handle the 60% missing values we have linearly interpolated the values if they were in the middle of the time series and excluded them elsewise. For testing, we have used the governorates, districts, resp. sub-districts, for which most of the data was available. For a future analysis, we highly suggest using the fitted values from the Hack4Good imputation team who have analyzed this problem in more detail.

Model Exploration

We have focused on fitting three different types of models for predicting the SMEB values for the next month: linear regression models, time series models and neural networks. For the comparison of the various models, we have defined a common evaluation metric based on a time-series adapted cross validation procedure. In detail, 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 the again the next month (April 2018) is predicted. This procedure is consecutively repeated until August 2019. Finally, the relative mean squared error (MSE) between all predicted and true values is determined and averaged over the given governorates, districts or subdistricts. Table 1 shows a quick overview for the district level.[1] In what follows, we provide a more detailed description of the models which we have used.

Model evaluation

To evaluate the performance of the models we have compared them to a baseline model. The baseline model is defined as the model which uses the value from the previous month as a prediction for the upcoming month, i.e., it predicts zero change. For a prediction of one month into the future, the baseline model outperforms all models, cf. Figure 1.

[1] We want to emphasize that these values were generated using the SMEB prices without water as there are less missing values. In the code we provide, we use the SMEB prices with water in Syrian pounds, which is why the results may differ.

Model	Rel. MSE
Baseline Model	1,79
Linear Momentum	4,34
SGolay Extrap	3,67
ElasticNetCV	5,97
RANSACRegressor	6,01
Loess+ARMA	3,34
Lasso	2,66
RidgeCV	2,87
Local Loess + ARMA	2,65
Holt-Winters	2,07

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

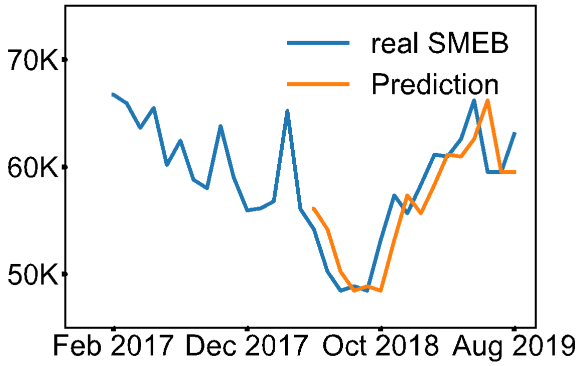

Figure 1: Due to the high volatility of the SMEB price, the baseline model serves as a good predictor for the prediction of one month into the future.

Time Series Models

Since the time series were not sufficiently long to observe seasonal effects, a trend is assumed without seasonal behavior.[2] The models which performed best were a combination between `LOESS` (with a linear trend) and `ARMA`(p, q) and a Holt-Winters model (non-seasonal). 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 tried three different approaches.
The first type of model is an autoregressive model, i.e., for each month the previous months are used as input features. In other words, the time series is split into independent samples which are used for training independently. The window length is optimized via cross validation on the time series at each instance of time. This is closely related to the `ARMA`(p, q) model mentioned previously. The difference between the two is that in the `ARMA`-model, we first fit a trend and then an autoregressive, moving average model (`ARMA`) to the remainder, while in the latter, 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 a product behaves as the trend-setter for the time series. Thus, the product price or its last relative change is included as additional input feature. Unfortunately, we could not identify any product as a trend-setter with sufficient evidence.
The third type of linear model assumes certain interactions between the districts. Here, a subset of the districts is assumed to be setting the trend for the other districts and their previous prices or price changes are included as input features. In fact, a slight improvement on the common evaluation metric could be found for two districts when comparing to the same type

[2] Also, there is no clearly visible seasonal effect in the approx. 2 years of data.

of models which do not include the additional features. The improvement though is considered not significant or even random, which we conclude from the following analysis:
We build a model which allows us to compare the predictive power of the two trendsetting districts to random artificial districts. We use a linear (autoregressive) model, where all real districts are used together with random artificial districts as input features. The model is then separately trained on all real districts as responses (one by one) using `LASSO`. 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 two real districts. This implies that the slight improvement which had been observed can most likely be considered as a random effect.

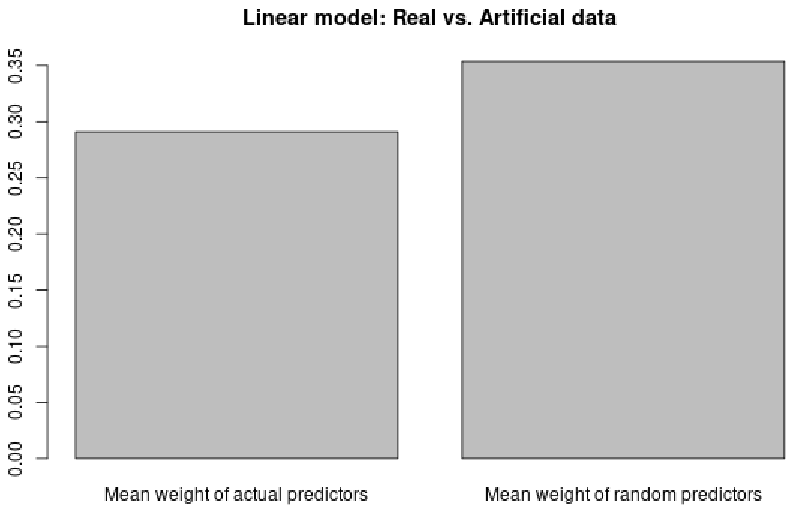

Figure 2: An experiment indicates that the predictive power of two districts as "trend-setters" which we have observed is likely a random effect.

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.[3] 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, resp. 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. After several months, 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

The conclusion we have come to is that due to the high volatility of the data, for a prediction of one month into the future, the "baseline method" which simply predicts the previous value outperforms all other models which we have tried to fit to the data. For predicting multiple

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

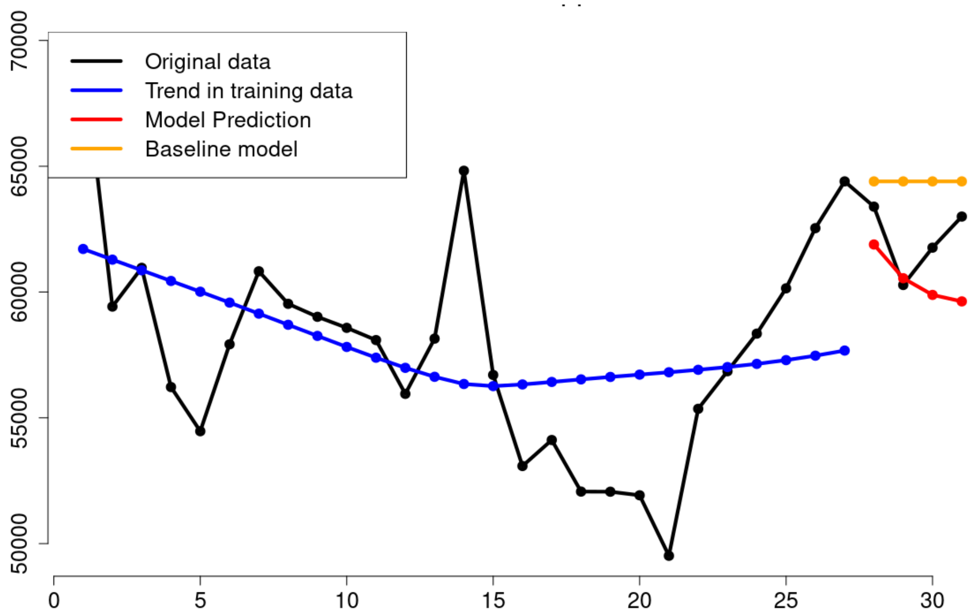

Figure 3: A prediction of four months into the future using the `LOESS`+`ARMA`(p, q)-model. Here, the model successfully picked up the decreasing price in the first two predicted months. The baseline model is added for comparison.

months into the future, certain time series models are able to pick up the trend of the price and outperform the baseline model.

Future Recommendations

We suggest to further look into long-term planning possibilities using the `LOESS`+`ARMA`(p, q) model. It would also be very interesting to look for further correlations between external events and the trend of the SMEB price. Including them into the models would perhaps explain the sudden peaks and otherwise seemingly random trends. Finally, we highly suggest using the results of the Hack4Good imputation team to fill the missing values in the data-cleaning process.

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.

Acknowledgements

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.