DAIMLER TEST

Subham Agrawal

AE13B063

Data Cleaning

- In the given dataset, we have x1 and x2 outflows with date stamps.
- Dates missing in the dataset are filled with ‘Nan’ for both X1 and X2. So that it gets easier to recognize seasonal effects.
- Predictive model used is robust to missing data, shifts in the trend, and large outliers. Thus, there is no need of anymore data peprocessing/cleaning.

Identification of Patterns

- Both time series in dataset are **not stationary**. There is a linear trend & non-constant variance in x1, and linear trend in x2.
- Stationary means that statistical structure of the series is independent of time. It allows preserving model stability i.e. the model in which parameters and structures are stable in time.
- Dataset is transformed to stationary by removing trends. Once the model has been constructed, we can account for trends separately, by adding the trends component wise.

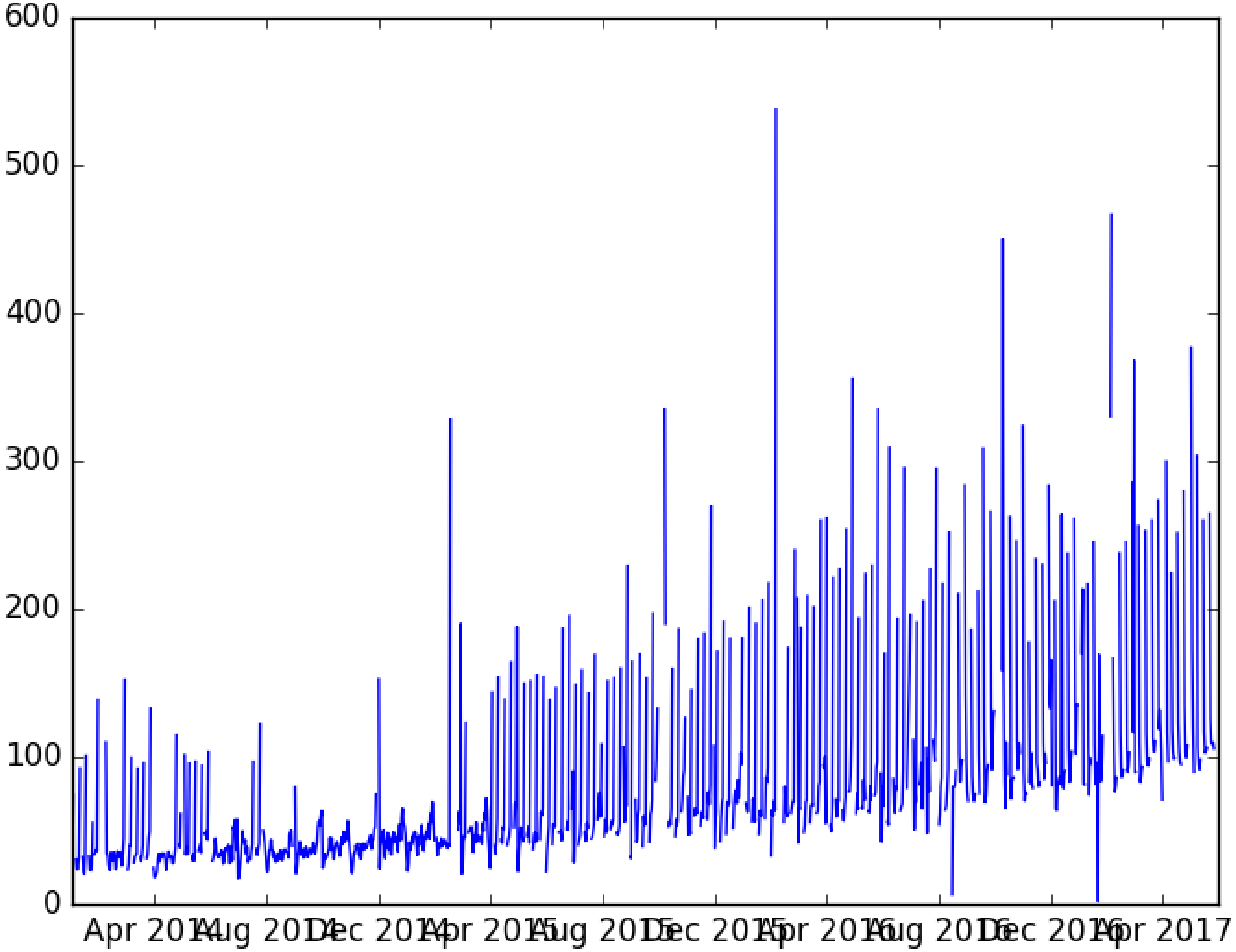

Fig. Plot of X1 before transformation

NON-STATIONARY

1) Increasing trend
2) Non-constant Variance

- For X1 took the logarithm of the series in order to stabilize the variance. Then, differenced the data to remove trend.

$$y(t)=x(t)-x(t-1)$$

- Then existence of stationarity was confirmed using Augmented Dickey-Fuller test.

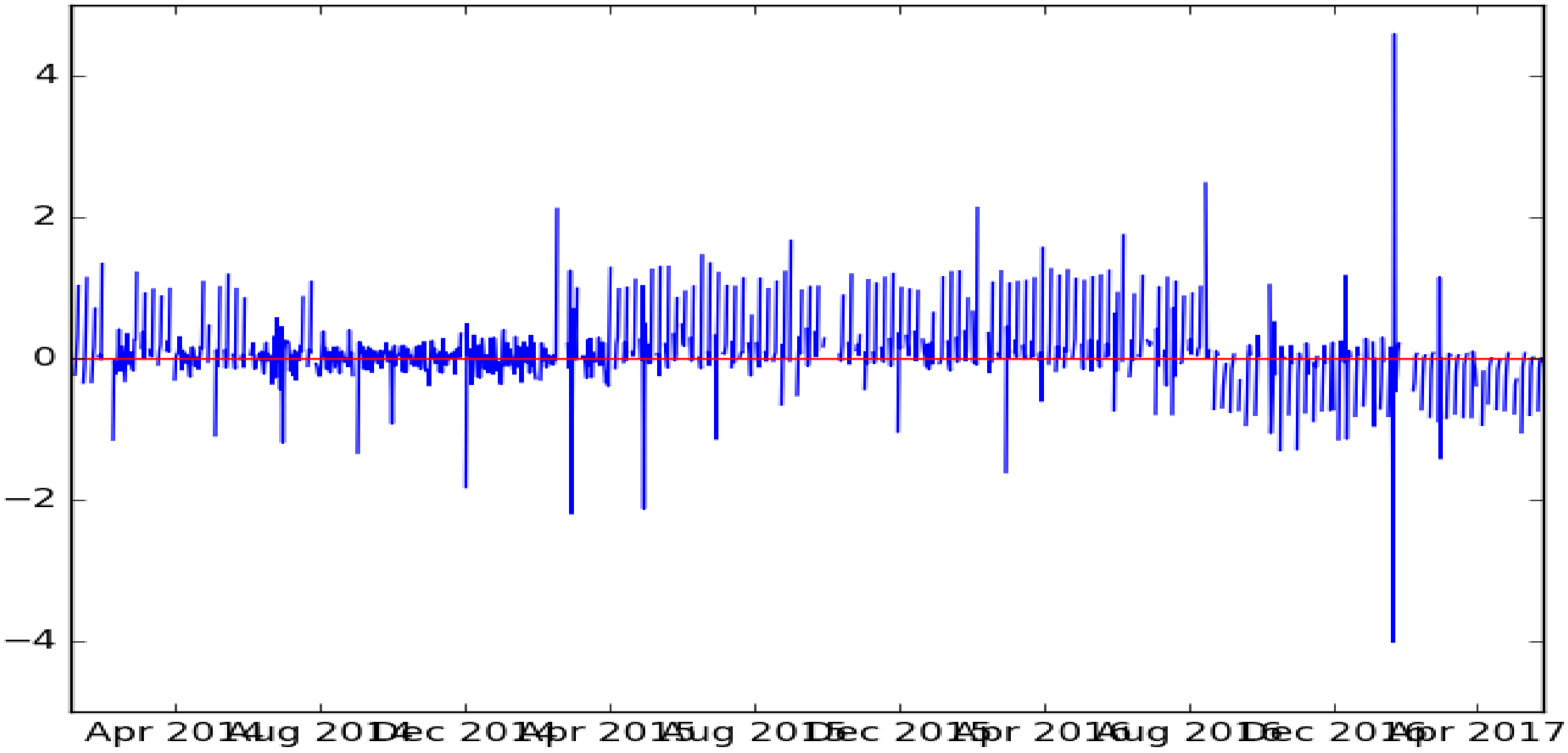

Fig. Plot of X1 after transformation

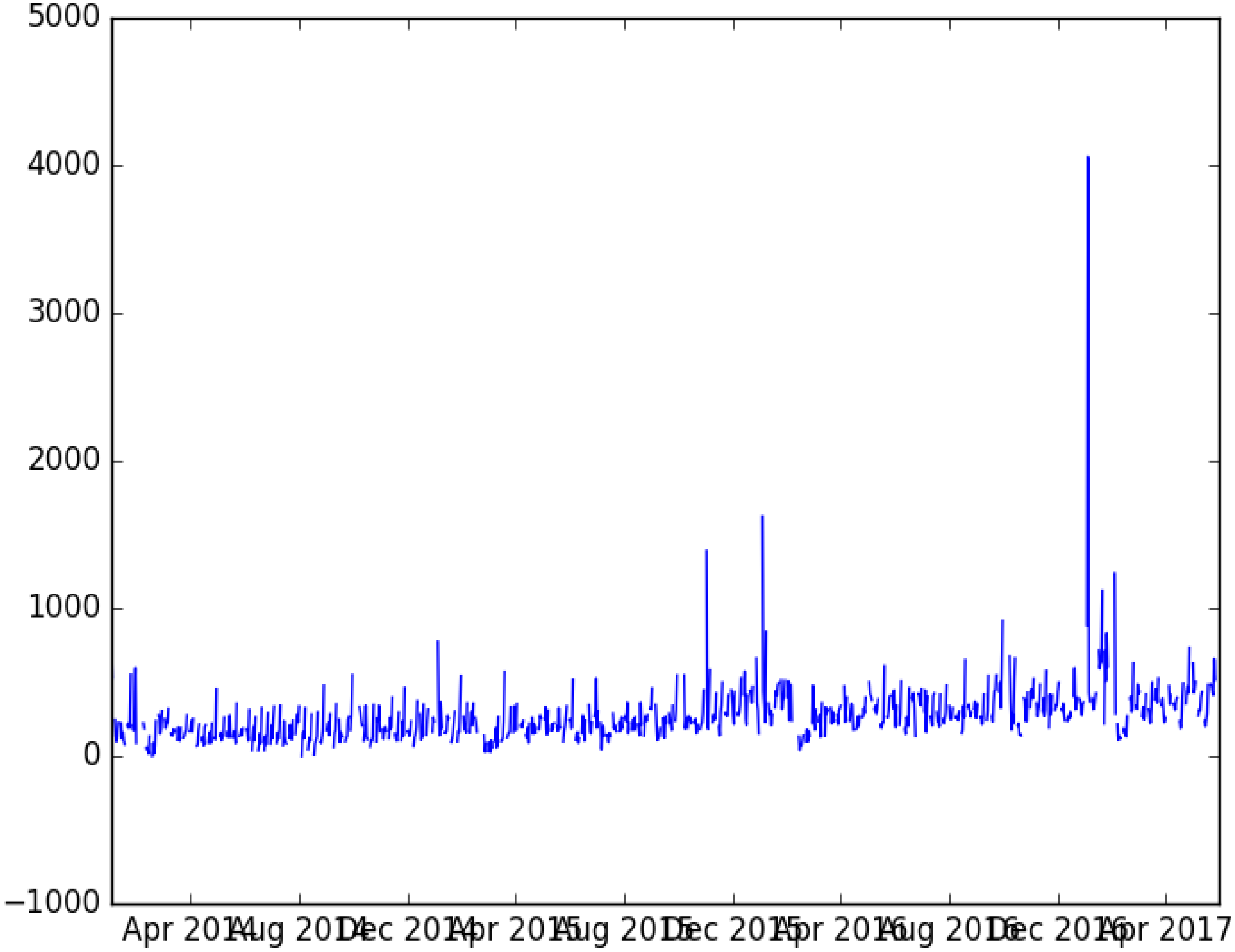

Fig. Plot of X2 before transformation

NON-STATIONARY
(Increasing trend)

- Differencing ‘X2’ removes the trend. Then the existence of stationarity was confirmed using Augmented Dickey-Fuller test.

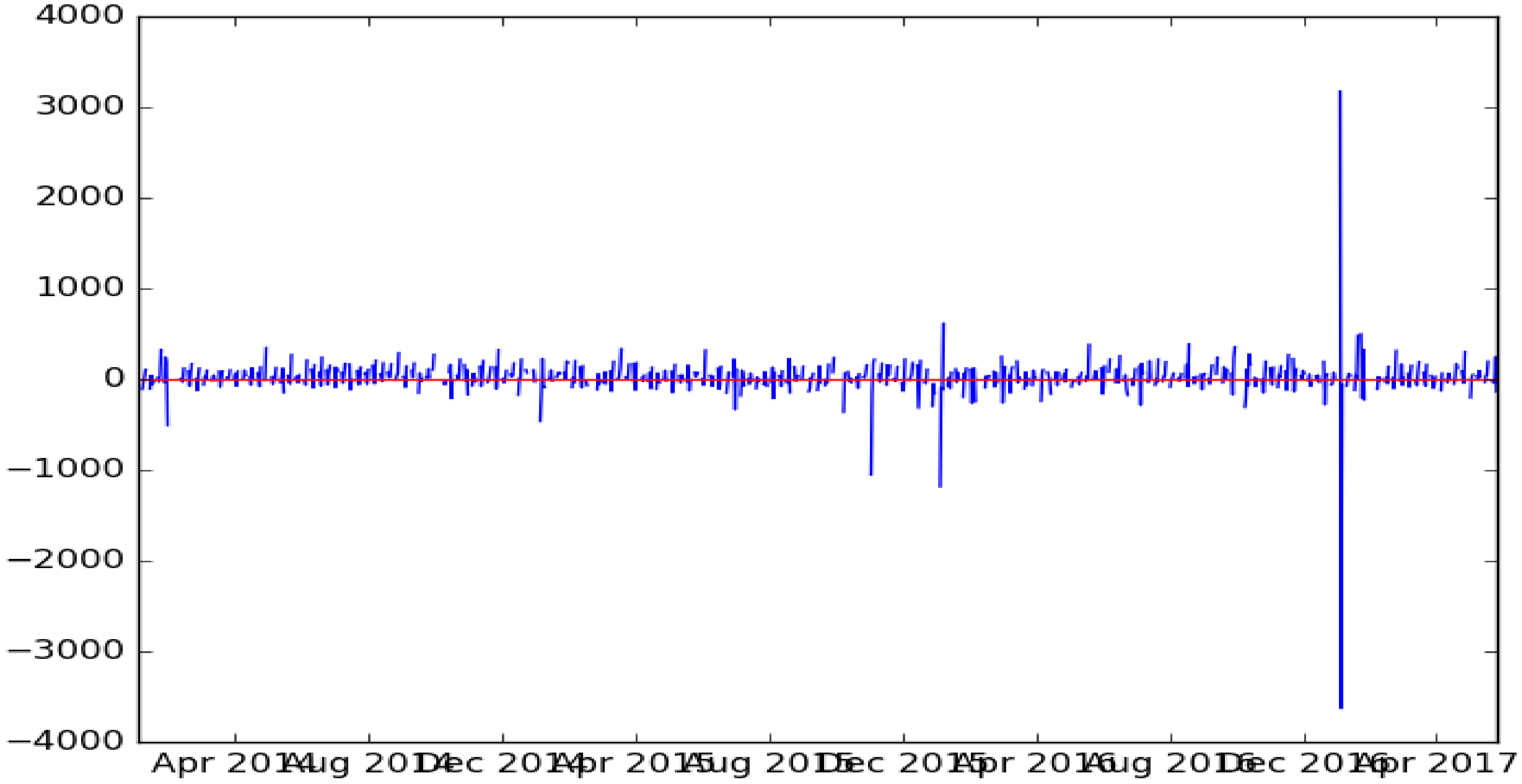

Fig. Plot of X2 after transformation

Identification of Patterns

- Autocorrelation plots are a commonly used tool for checking randomness in a dataset. The randomness is ascertained by computing autocorrelations for data values at different time lags. Since we have missing values in the dataset, autocorrelation couldn’t be plotted using standard libraries in Python.
- So covariance matrix is calculated by forming a data frame with different time lags. Above process was repeated for dataset including weekends/non-business days and excluding weekends. Co-variance matrix gave an idea of seasonality in both time series.

1. Important values from co-variance matrix for X1
    a. Excluding weekends:
        i. t-5 : 0.63 -> **WEEKLY** SEASONALITY (damps slowly)
    b. Including weekends:
        i. t-7 : 0.53 -> **WEEKLY** SEASONALITY (damps slowly)
        ii. t-366: 0.315-> **YEARLY** SEASONALITY

Hence, concluded to use dataset **including weekends** for X1 because we would miss yearly seasonality otherwise.

2. Important values from co-variance matrix for X2
    a. Excluding weekends:
        i. t-1 : 0.5 (sudden damp)
        ii. t-262 : 0.287 -> **YEARLY** SEASONALITY
    b. Including weekends:
        i. t-1: 0.49 (sudden damp)
        ii. t-31: -0.39 -> **MONTHLY** SEASONALITY
        iii. t-366: 0.513-> **YEARLY** SEASONALITY
        iv. t-**429**: 0.44 -> SEASONAL

Hence, concluded to use dataset **including weekends** because we don’t miss seasonality presents in time series.

Forecast Method

- Firstly modelled seasonal ARIMA for both outflows, which didn’t produce good results. Because both time series have multiple seasonal periods which can’t be captured in the seasonal ARIMA model.
- Next, **decomposition procedures** was used because it estimates seasonal effects that can be used to create and present seasonally adjusted values. A seasonally adjusted value removes the seasonal effect from a value so that trends can be seen more clearly.
- In this case, **additive decomposition** is used because the seasonal variation is relatively constant over time.

Additive: Trend + Seasonal+ Random

Forecast Method

- **Prophet** is a method/model used for forecasting time series data provided by Facebook. It is based on an additive model where trends are fit with different seasonality. It works best with daily periodicity data and is robust to missing data and large outliers. Because we have trends, seasonality, missing values and outliers; it is used to model.
- Seasonality (except weekly) is modelled with the help of Fourier series with different time periods.

  Weekly seasonal component is modelled using dummy variables.
- After decomposition of seasonality and trends in the data, prophet uses many different forecasting techniques (ARIMA, exponential smoothing, etc.) to model the white noise. It gives us forecast after adding up all the components.

1. **Model for X1**

- Decomposed trend, weekly and yearly seasonality.
- Weekly pattern was observed to change significantly with time. There was a shift in weekly pattern in the year 2016.
- Weekly seasonality is most prominent (0.53) from co-variance matrix. Thus, only **last one year data** is used to get predictions i.e. 2016-07-01 to 2017-07-31.

2. **Model for X2**

- Decomposed trend, weekly, monthly and yearly seasonality to model.
- Another seasonality with time period equals to 429 was also decomposed because it has high value in covariance matrix obtained before.
- In this case, we use the **entire data** because yearly seasonality is most prominent among all others.

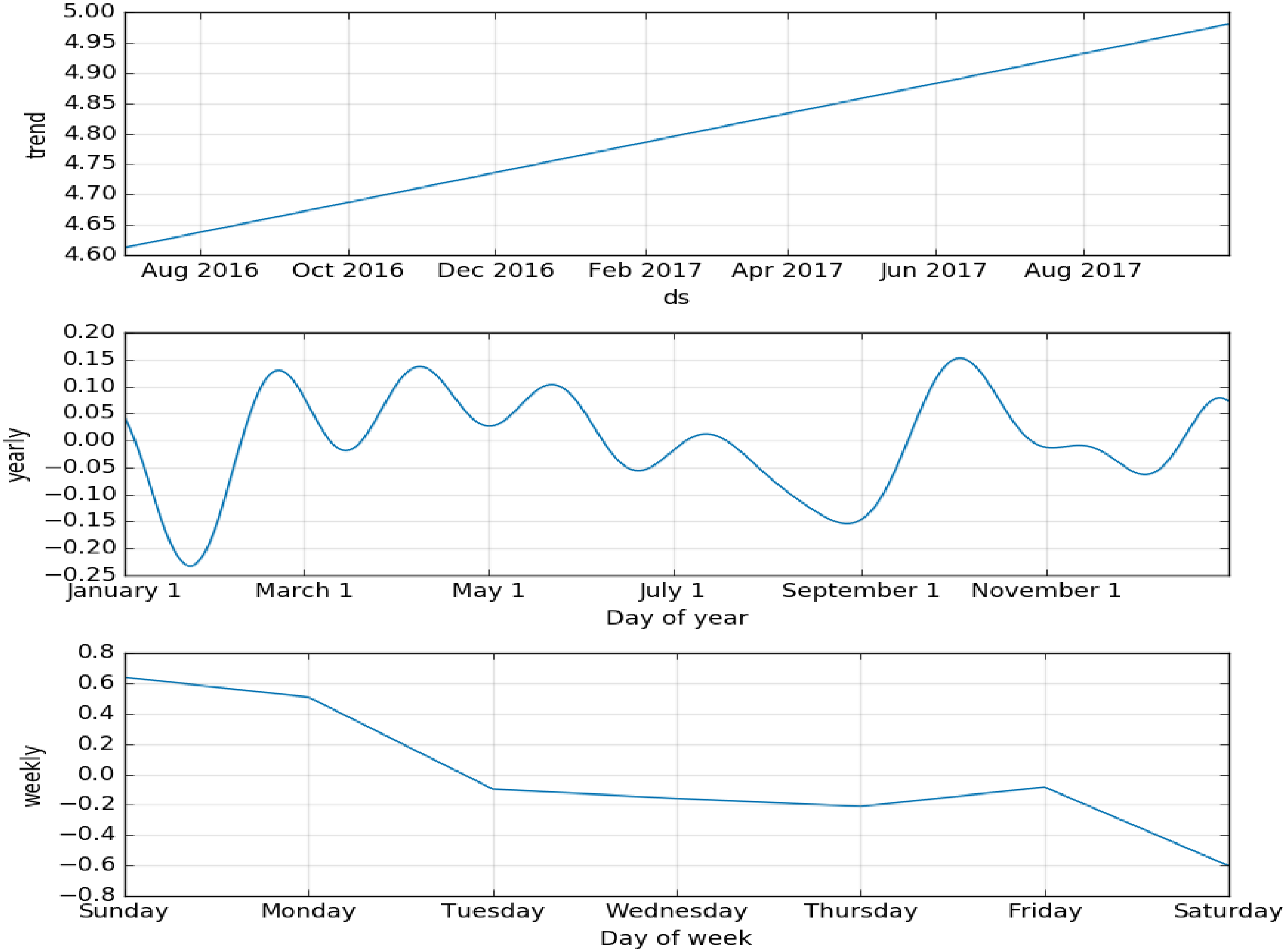

Fig. **Decomposition of X1**

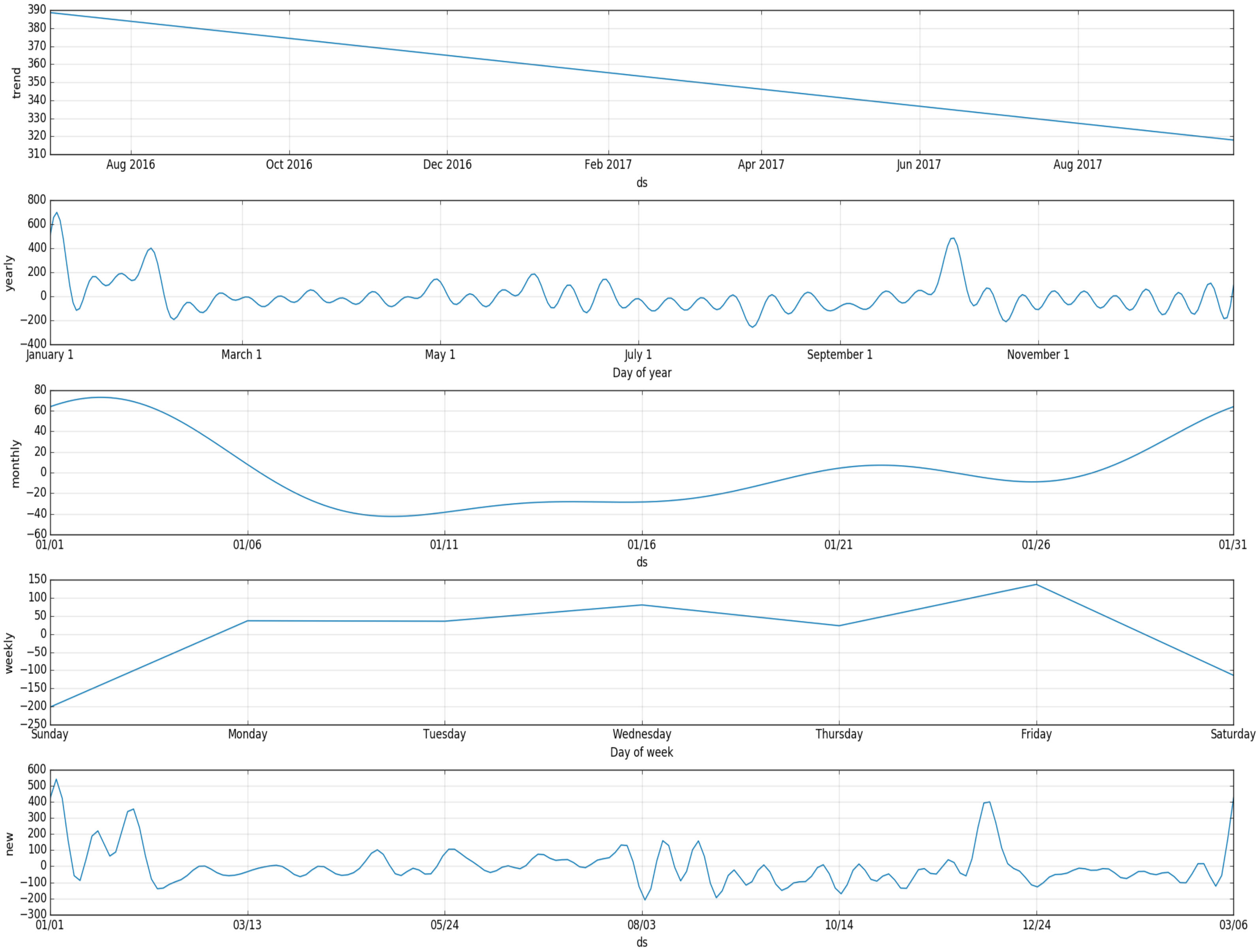

Fig. **Decomposition of X2**

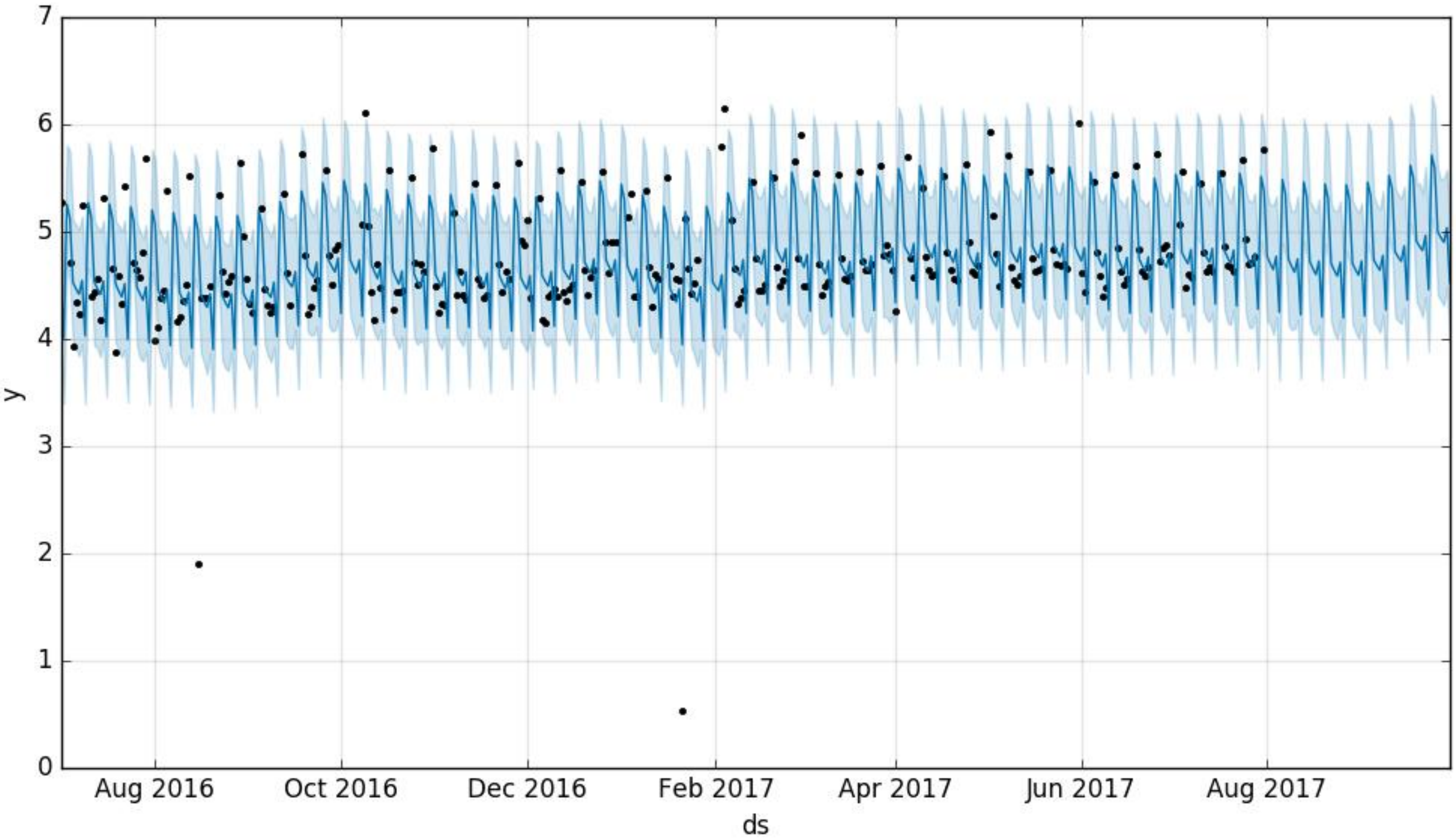

Fig. **Modelling of X1**

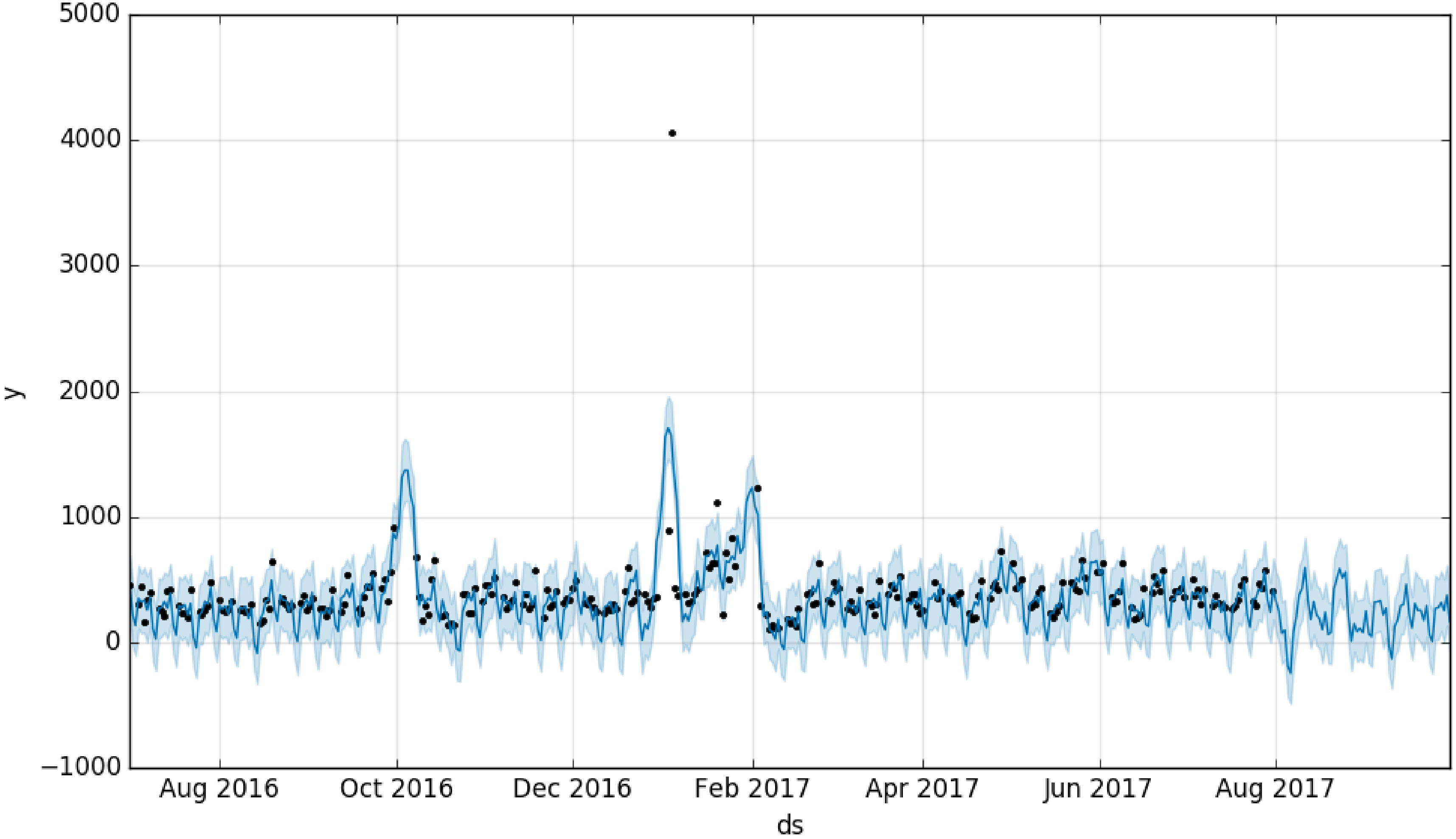

Fig. **Modelling of X2**

Scalability

- This approach to the problem can be used for similar issues/problems.
- Different tests on the dataset which are performed manually here can be done using a pre-processing pipeline in a software form.
- A similar pipeline can be formed for predictive modelling or already built ones like FBProphet can also be used for same.
- Scalability depends on the similarity in the problems. If the problems are very similar, it is easy to build an API with completely automated forecasts that can be tuned to obtain best results.