Data Forecasting for Studo Newspost Impressions

Laurenz Lazarus, Valentin Slawicek

Dataset & Data Aggregation

Approach

- Data aggregation
- Data preprocessing
- Data filtering
- Test out various forecasting methods:
  - Time series prediction using weka framework:
    - Gaussian Processes Model
    - Holt-Winters
  - Time series prediction by self-made regression
    - Naive: Average regression
    - Average regression + sinus regression
    - Average regression + 2x sinus regression
- Evaluation of different methods with Mean Square Error

Data Preprocessing
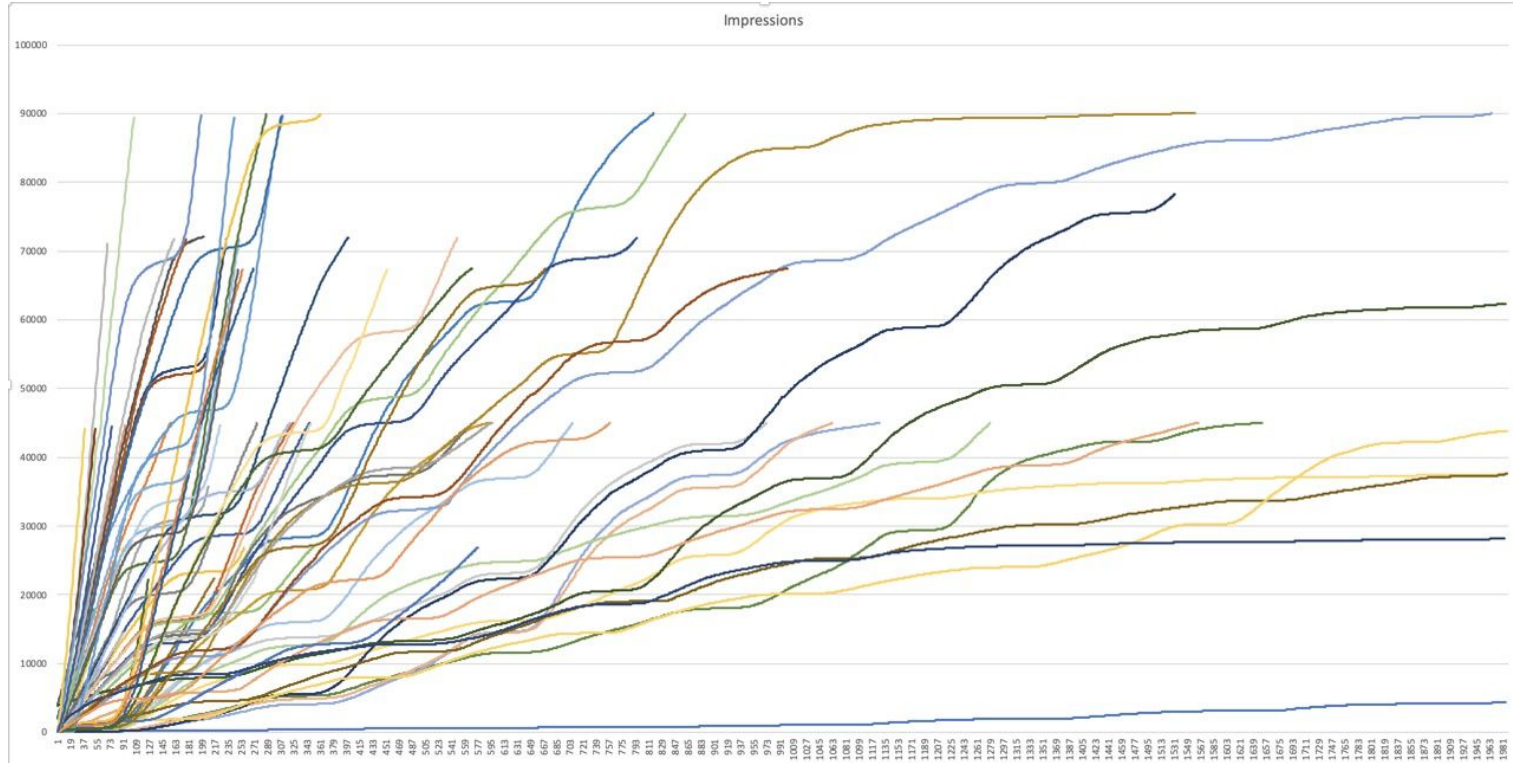
Differential & Moving Average Filtering

Window = 20
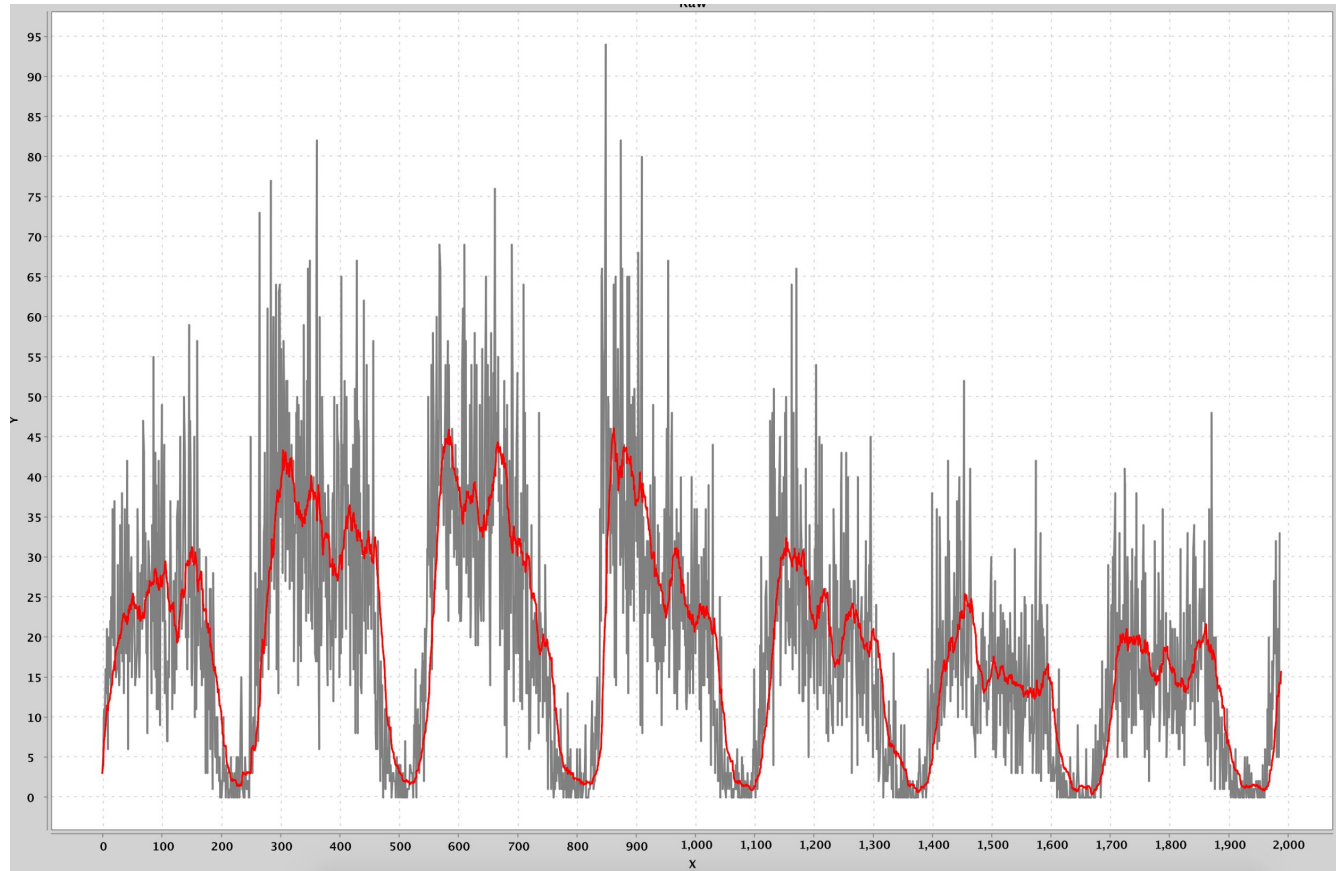
Weka framework: first learnings

2000 data points are too much...

Further data preprocessing (data point reduction) is needed

```
/Library/Java/JavaVirtualMachines/openjdk-11.0.1.jdk/Contents/Home/bin/java ...
Connected to the target VM, address: '127.0.0.1:65151', transport: 'socket'
Transforming input data...
Juni 11, 2019 4:11:19 NACHM. com.github.fommil.jni.JniLoader liberalLoad
INFO: successfully loaded /var/folders/j5/lf9m8by140n0v27q23w6qslw0000gn/T/jniloader14589504639813674045netlib-native_system-osx-x86_64.jnilib
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
    at weka.filters.unsupervised.attribute.Copy.input(Copy.java:242)
    at weka.filters.Filter.useFilter(Filter.java:705)
    at weka.filters.supervised.attribute.TSLagMaker.createLags(TSLagMaker.java:1687)
    at weka.filters.supervised.attribute.TSLagMaker.getTransformedData(TSLagMaker.java:2649)
    at weka.filters.supervised.attribute.TSLagMaker.getTransformedData(TSLagMaker.java:2585)
    at weka.classifiers.timeseries.WekaForecaster.buildForecaster(WekaForecaster.java:1092)
    at MainKt.createWekaForecast(main.kt:76)
    at MainKt.main(main.kt:14)
    at MainKt.main(main.kt)
Disconnected from the target VM, address: '127.0.0.1:65151', transport: 'socket'

Process finished with exit code 1
```
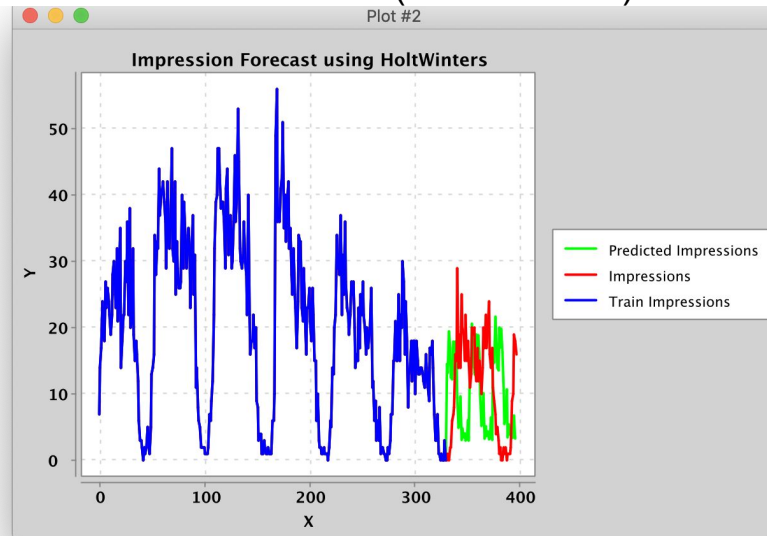
Gaussian-Processes vs. Holt-Winters with Dataset 1

G.P. delivers unexpected results, H.W. doesn't look too bad for a first attempt.
Lets try to simplify/reduce the data.
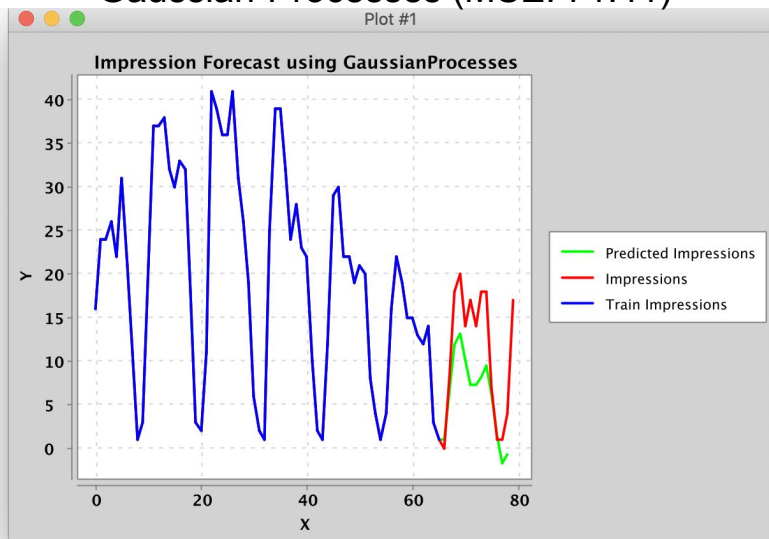

Gaussian-Processes (MSE: 2.85*10^7)
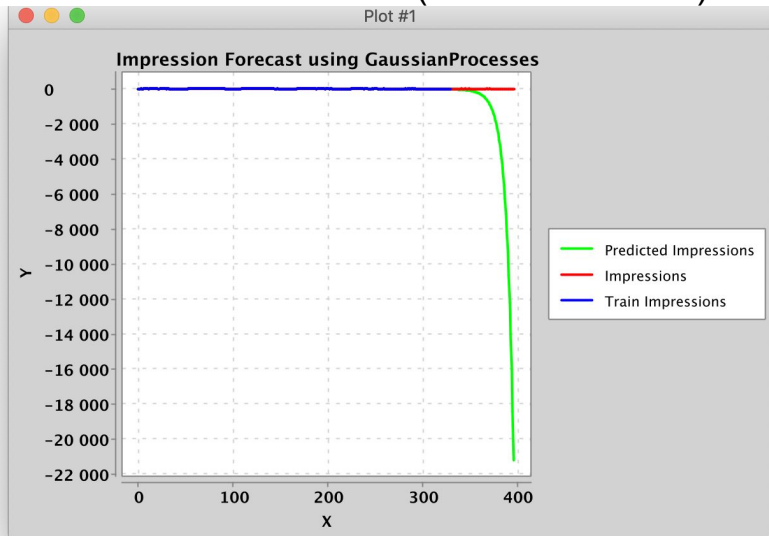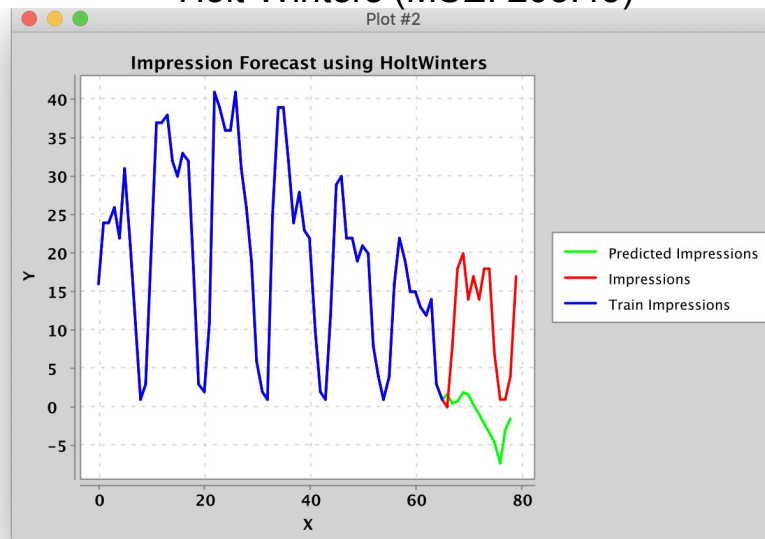

Holt-Winters (MSE: 124.10)

Gaussian-Processes vs. Holt-Winters with processed Dataset 1

G.P. looks a lot better now but H.W. looks worse.
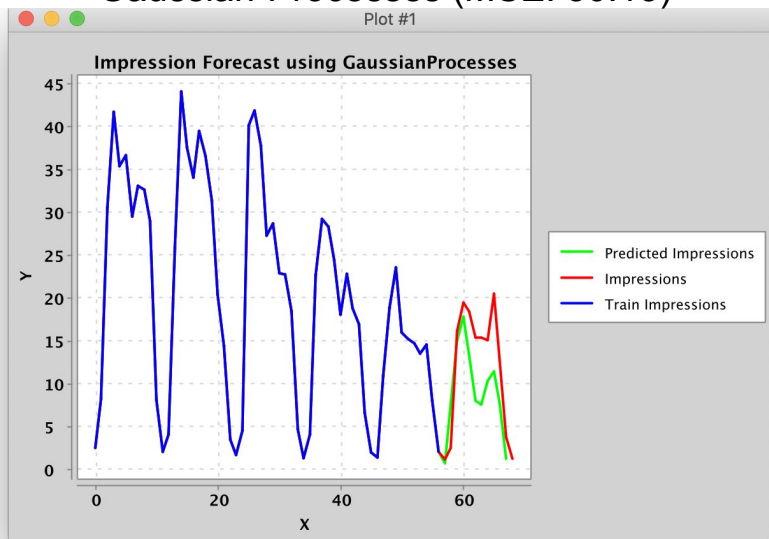

Gaussian-Processes (MSE: 71.41)
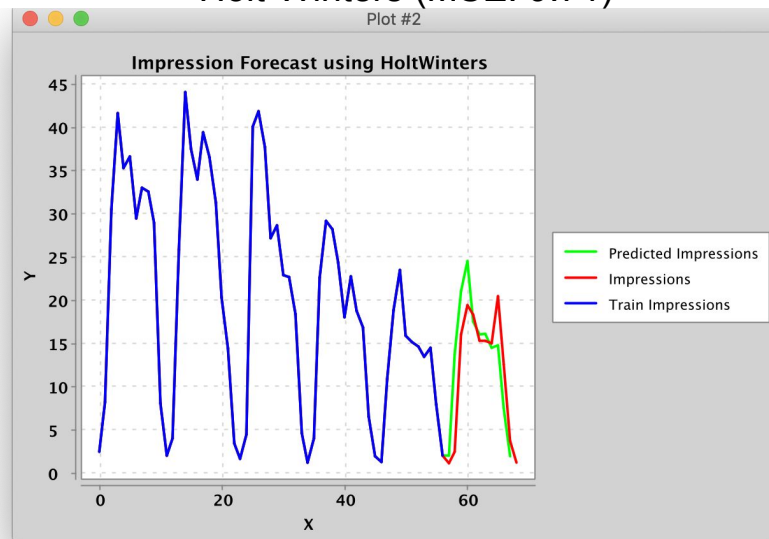
Holt-Winters (MSE: 203.43)

Gaussian-Processes vs. Holt-Winters with final processed Dataset 1

Dataset was modified so it is easier to determine the correct cycle length of a season. This is crucial for the model of H.W.
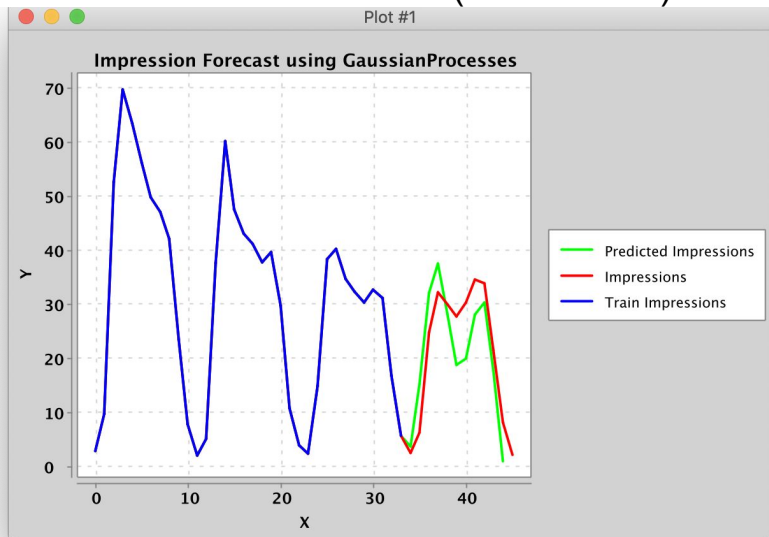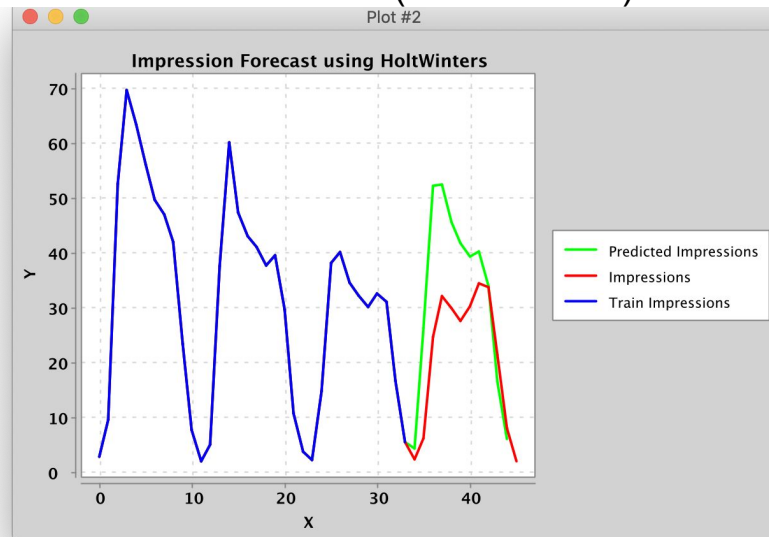
Gaussian-Processes vs. Holt-Winters with final processed Dataset 2

Same parameters (except of Holt-Winters-season-cycle-length of course),
Different dataset

Gaussian-Processes (MSE: 64.35)
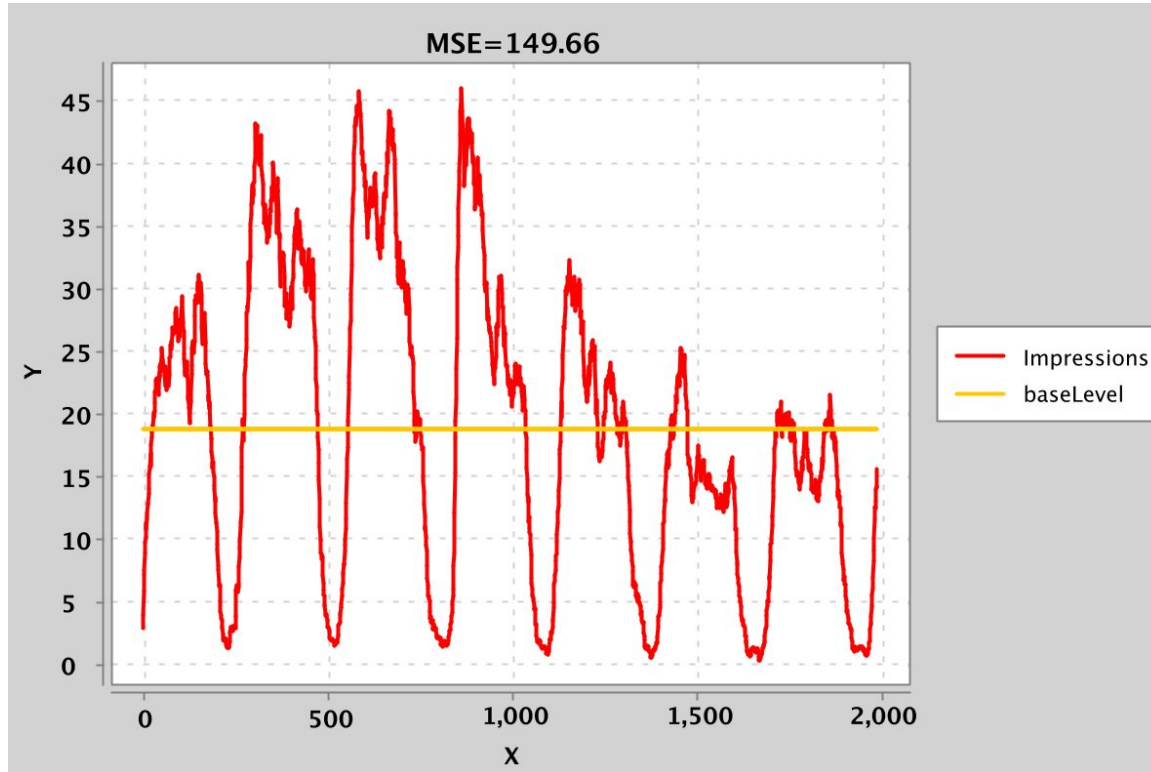
Holt-Winters (MSE: 153.06)
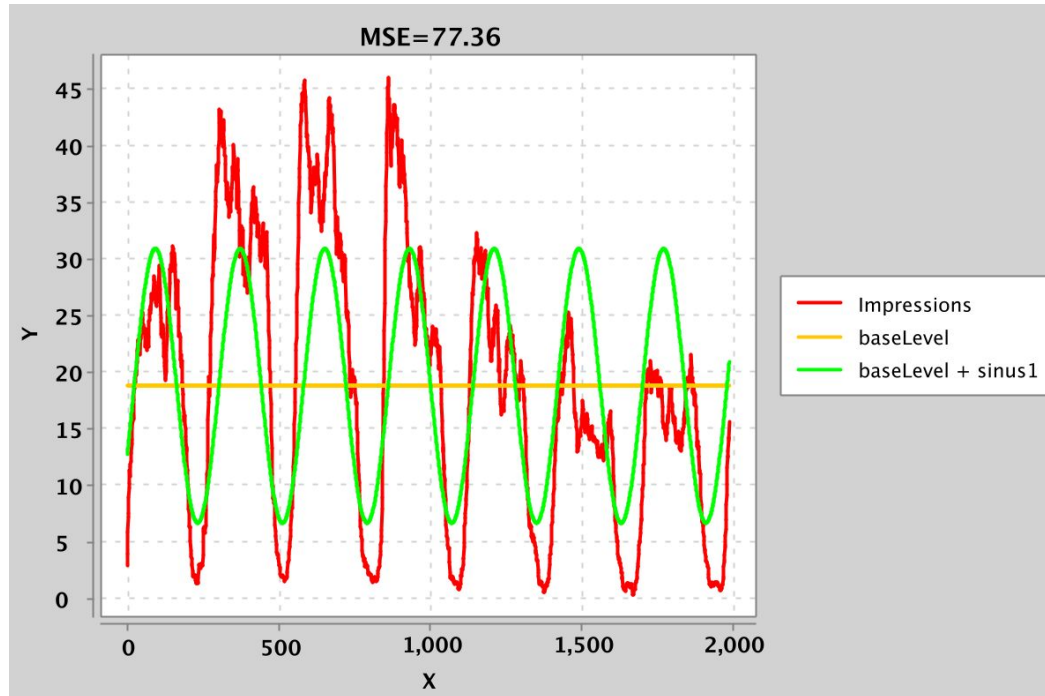
Regression

No framework

Data size not a limit

Full understanding of limitations

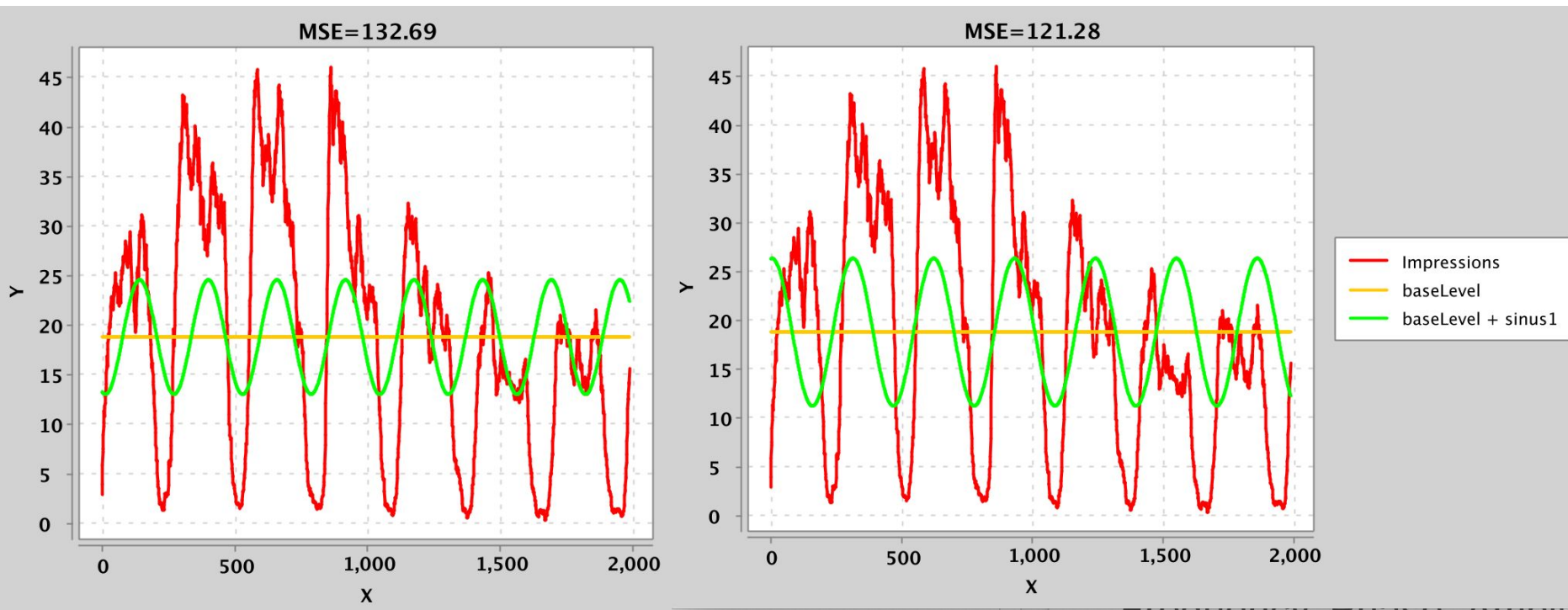# Method: BaseLevel Regression

Method: BaseLevel + Sinus Regression

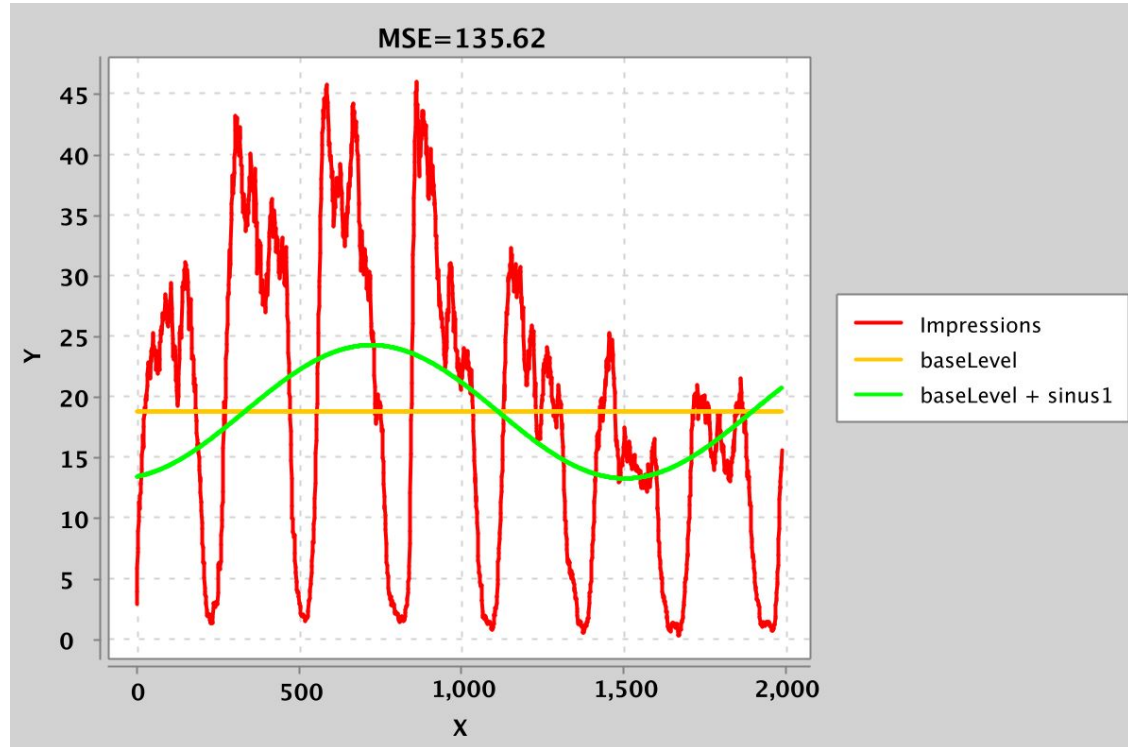Frequency, Phase, Amplitude

Method: BaseLevel + Sinus Regression
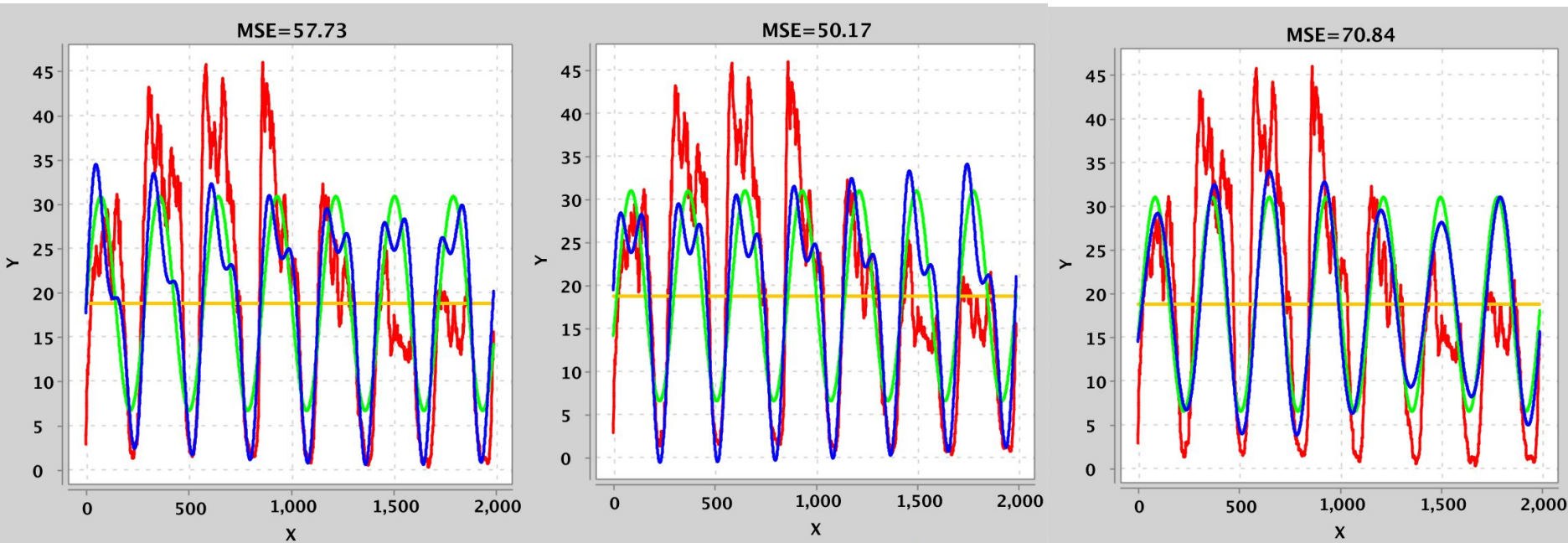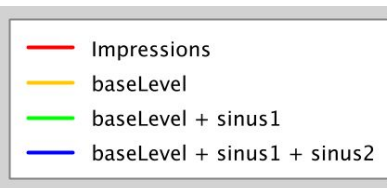
Without amplitude hint → Local minima

Method: BaseLevel + Sinus Regression

Without frequency hint
→ Local minimum

Method: BaseLevel + Sinus1 + Sinus2 Regression

2x Random Sinus

Conclusion

- No silver bullet
- Weka Framework: Gaussian-Processes and Holt-Winters
    - More data isn't always a good thing
    - G.P. can't deal with too large/complex datasets but if you keep that in mind it delivers useable results
    - Hold-Winters needs exact seasonal parameters, which are often hard to find
    - Both are not easy to understand and it can be a hard time to find errors
    - MSE/day Gaussian-Processes: {**30**, **64**}   Holt-Winters: {**10**, **153**}
- Regression
    - Regression works without careful data preprocessing, without parameter tuning, data size is not a problem and can't drift away when predicting multiple days
    - MSE/day BaseLevel: **150**
    - MSE/day BaseLevel + Sinus: {**77**, **121**, **133**, **136**}
    - MSE/day BaseLevel + 2x Sinus: {**50**, **58**, **71**}
    - → A lot of local minima → run multiple times to get a good result