Pattern Mining on EEG data: Detecting Eye State from a 13 Channel labeled EEG time series data set

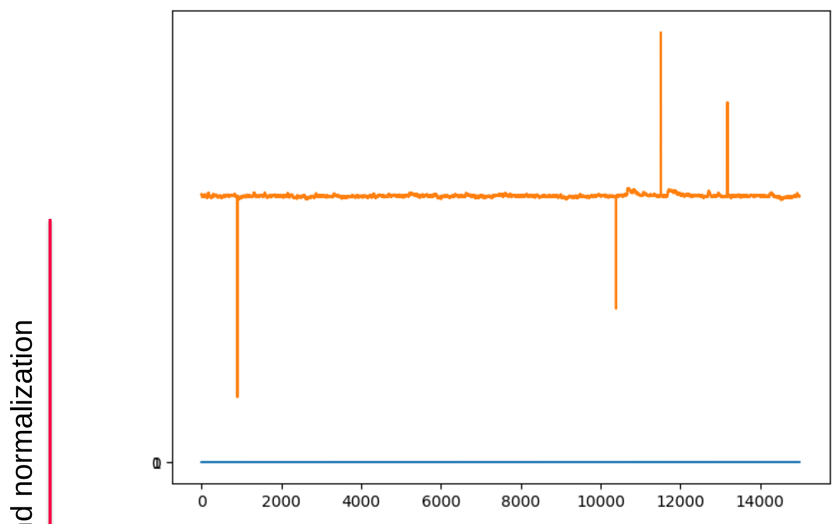
Matthias Friedrich
matthias.friedrich@student.tugraz.at

Data set

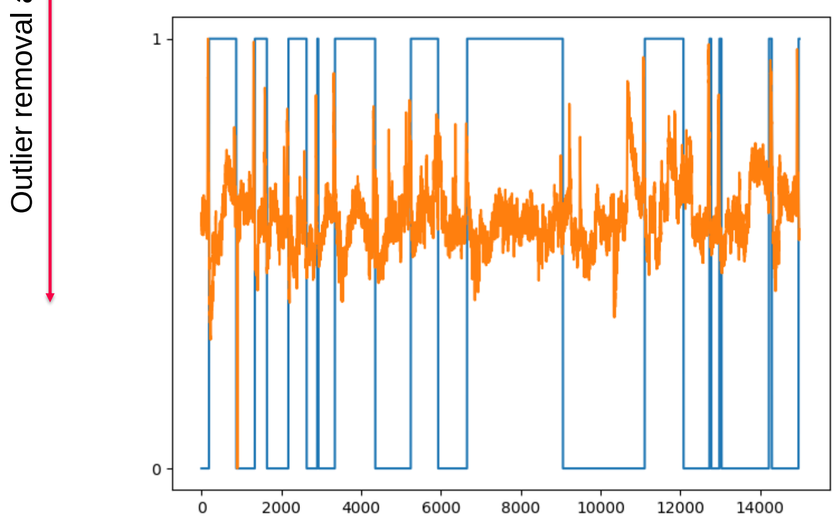
- EEG data from a single meassurement over 120 seconds
- 14980 data points per channel
- 13 Channels
- Labeled data: „Eyes closed/open“
- Raw data with outliers

Feature engineering

- Outlier removal: median deviation method
  - Calculate median and standard deviation of data set
  - If sample more than k * standard deviation away from median -> remove it
- Normalization
  - Samples from different channels vary greatly in value(voltage)
  - Thus, each channel was normalized by its own
  - Min-Max normalization resulting in values from 0 – 1



EEG channel 1 raw data



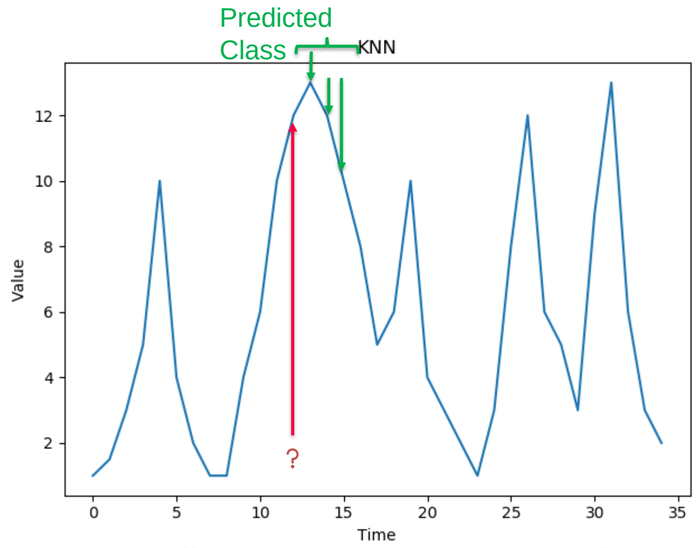
EEG channel 1 after outlier removal and normalization

The naive approach: KNN

- K-Nearest-Neighbors(KNN)
- Look at K known samples and use their dominant class as prediction
- Yields 97% prediction accuracy, if data is „tweaked in our favor“
- Shuffle whole dataset and take 10% as test set

```
traX, tesX, tray, tesy = train_test_split(X, y, test_size=0.1, shuffle=True)
```

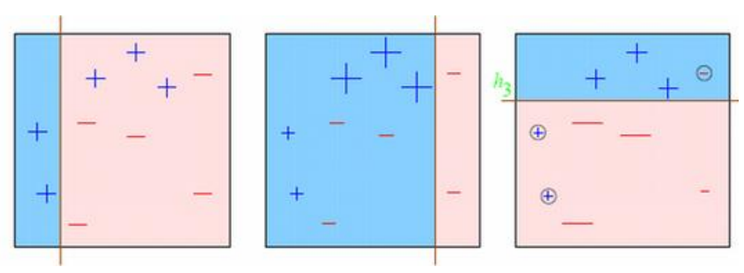
- Commonly used technique for extracting train/test data
- Problem with this methology:
  - Shuffling the timeseries allows the classifier to „look into future“
- Accuracy without shuffling data: 50% !



KNN with shuffled time series data: the classifier can ‚look into the future‘

Gradient Boost

- Ensemble method
- Combination of decision trees
- Sequential ordering -> one classifier „learns from mistakes“ of previous
- Wrong observations get higher weight in next iteration
- Produced 74% prediction accuracy



Boosting: Errors get higher weight in next iteration
Image source: https://www.analyticsvidhya.com/ [1]

Evaluation

- K-Fold Validation
  - Split data set into K sub sets
  - Take one subset as test set, others as training set
  - Iterate over K, each time using a different subset as test set
  - Use mean accuracy over all K runs as overall accuracy
- Used K was 5
  - This is quite low, but made sense for the small data set
- Accuracy of last sub set was significantly lower than mean
  - Probably due to significantly different structure of data set in the last division

Conclusion

- Accuracy of 74% accepable but not great
- Test set needs to be chosen carefully to produce valid result
- Working with such a small data set makes validation even harder

Sources

[1] https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/