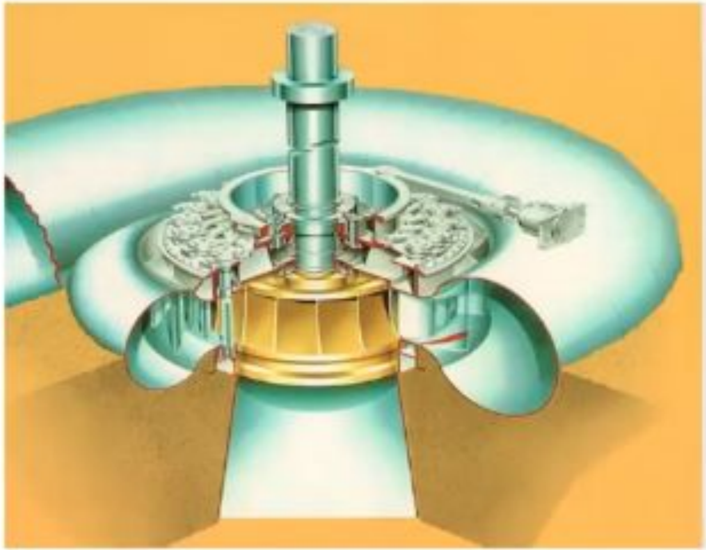
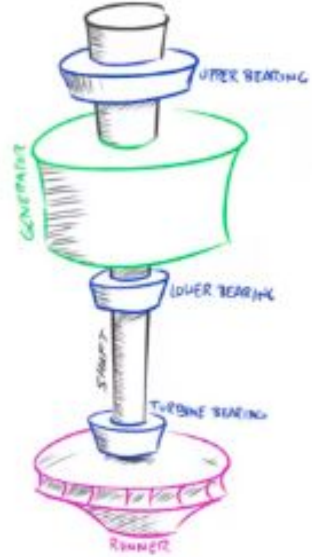
KraftHack 2022

Team Data Liberators: Hayden, Simon, Andris

Introduction

- Predicting the strain rate on 6 different bolts.
- Input data contains features such as:
  - Pressures
  - Guide Vane Openings
  - Seasonality
  - Power
- Another critical aim is to have an estimate for end of lifetime for each bolt

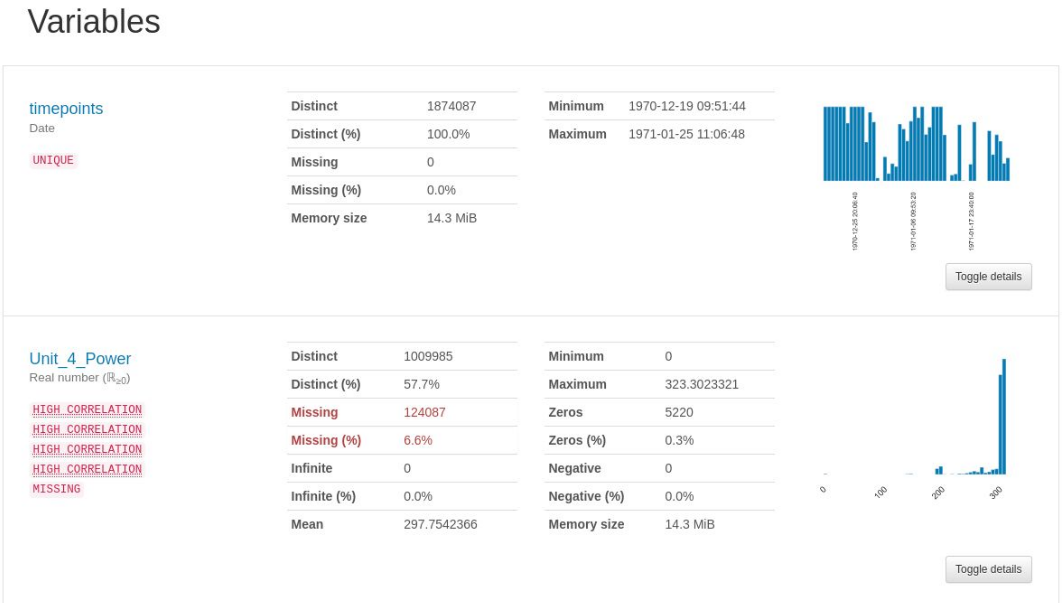




Data exploration

- Preliminary data exploration done with the pandas data profiler.

- Explored many different features, some useful, some useless (no significant improvement to the model or decreased performance)
  - Useless: Power triangle (reactive/active/real power), pressure difference
  - Useful: Seasonal features (day, month, “season” [defined according to our expert “vestlander” Simon]), “Stress deficit” (3000 - permissible stress), number of seconds in given operating condition



Variables

timepoints
Date

UNIQUE

Distinct	1874087	Minimum	1970-12-19 09:51:44
Distinct (%)	100.0%	Maximum	1971-01-25 11:06:48
Missing	0		
Missing (%)	0.0%		
Memory size	14.3 MiB		

Unit_4_Power
Real number ($\mathbb{R}_{\geq 0}$)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	1009985	Minimum	0
Distinct (%)	57.7%	Maximum	323.3023321
Missing	124087	Zeros	5220
Missing (%)	6.6%	Zeros (%)	0.3%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	297.7542366	Memory size	14.3 MiB

Data exploration

- Power can be expressed in terms of an active (real) and reactive (imaginary) component:

$$S = P + jQ$$

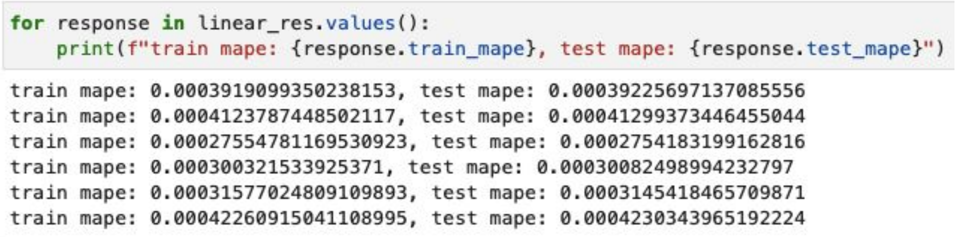
- Tried combinations of determining the remaining power variable through the magnitude, no luck.

$$|S| = \sqrt{P^2 + Q^2}$$

- Permissible stress: f = p * d / (2 * \eta * t) (according to some university slides which I’ve lost the link to XD).
- Max flow rate according to this paper.

Data exploration

- Seasons:
  - Snow: December - March
  - Melting: April - June
  - Dry: July - August
  - Heavy Rain: September - November
- Seasonality features improved results by an order of 5-20x (depending on the bolt)



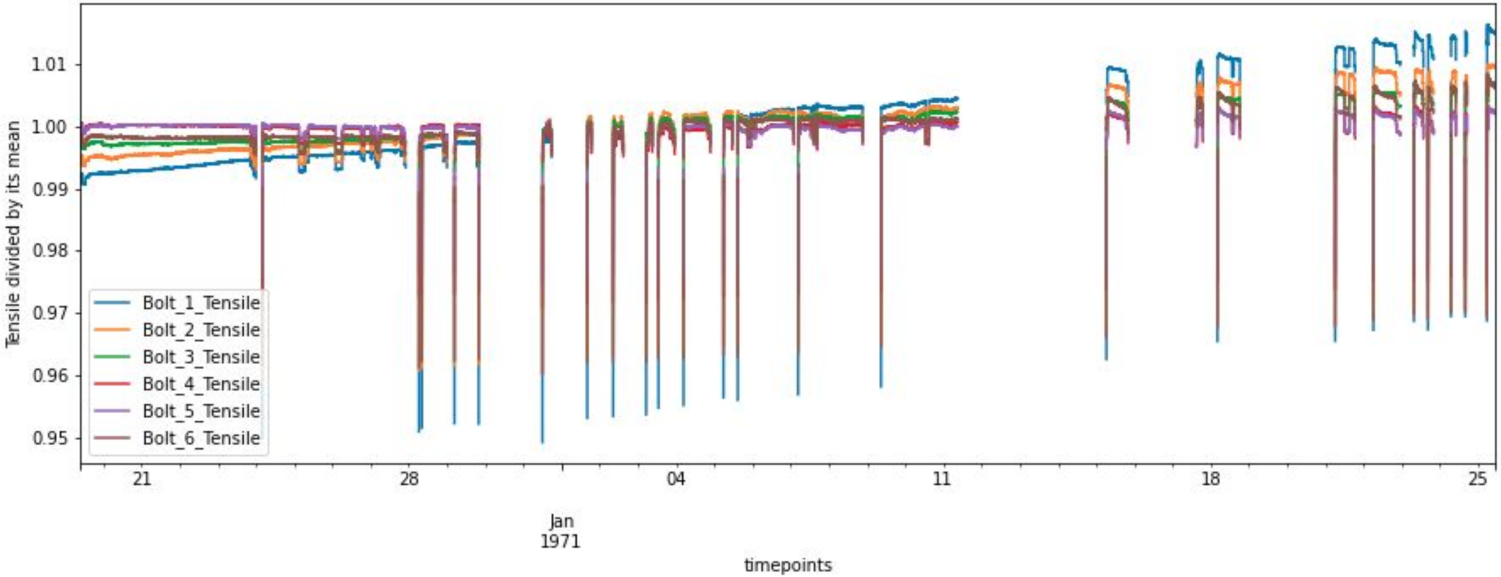
```
for response in linear_res.values():
    print(f"train mape: {response.train_mape}, test mape: {response.test_mape}")
```

```
train mape: 0.0003919099350238153, test mape: 0.0003922569713708556
train mape: 0.0004123787448502117, test mape: 0.0004129937344645504
train mape: 0.00027554781169530923, test mape: 0.0002754183199162816
train mape: 0.000300321533925371, test mape: 0.00030082498994232797
train mape: 0.00031577024809109893, test mape: 0.0003145418465709871
train mape: 0.00042260915041108995, test mape: 0.0004230343965192224
```

Data exploration

- Plotting of trends showed that the strain was increasing considerably over time for several bolts.

- Indicates that time is an important parameter for bolt strain!



Model considerations

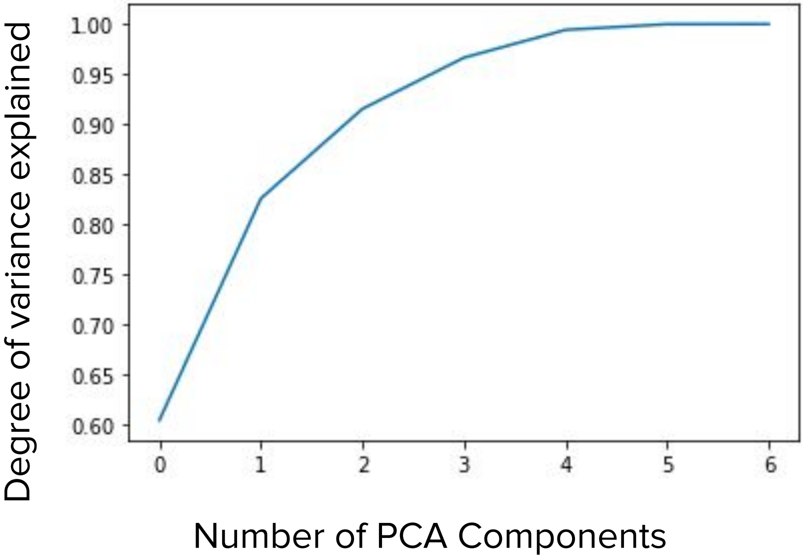
- A **multi-target regression** problem was considered initially but resulted in poor performance.

- Having a **single predictive model for each bolt** achieved good accuracy and allowed us to capture the subtleties of each bolt.

- **Several feature reduction techniques** were considered at an early stage:
    - **PCA** (where number of components was selected according to those that explained the most variance)
    - Hand-crafted feature by looking at variables that showed the highest degree of correlation with target variables (i.e. strain values of each bolt)

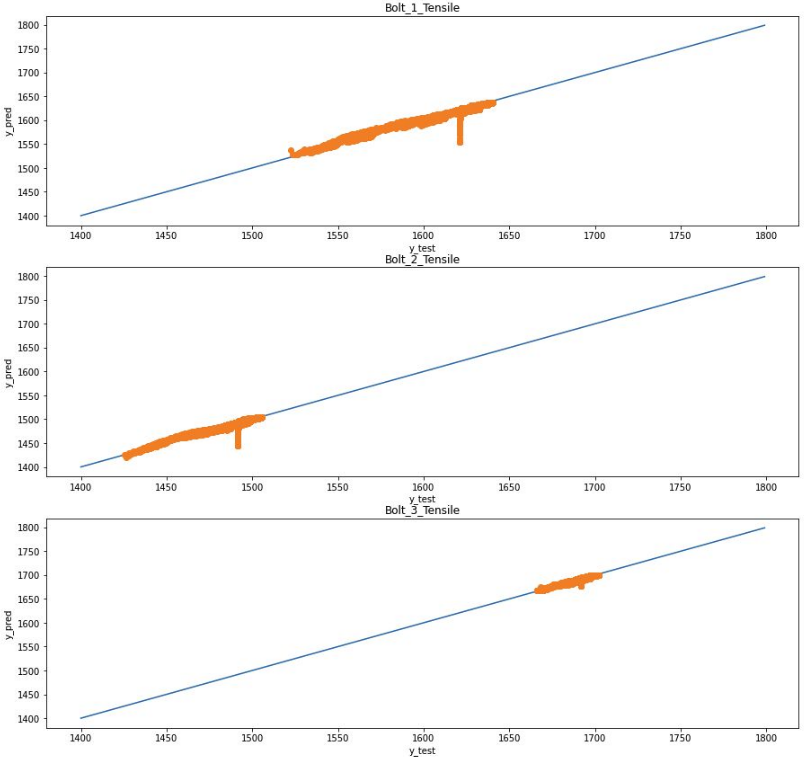
- These feature engineering techniques were then paired with a predictive model to increase model performance.

Here 3 parameters were found to explain over 90% of dataset variance



Test metrics and gauging performance

- Test size: 40%
- Train-test split was not shuffled, to avoid “leaking” information into the training set
- **“Mean Absolute percentage error”** used to gauge performance numerically since this is used as the evaluation metric in the hackathon. Is not heavily influenced by outliers.
- Plot true bolt tension vs. predicted bolt tension as a means of visual evaluation of results



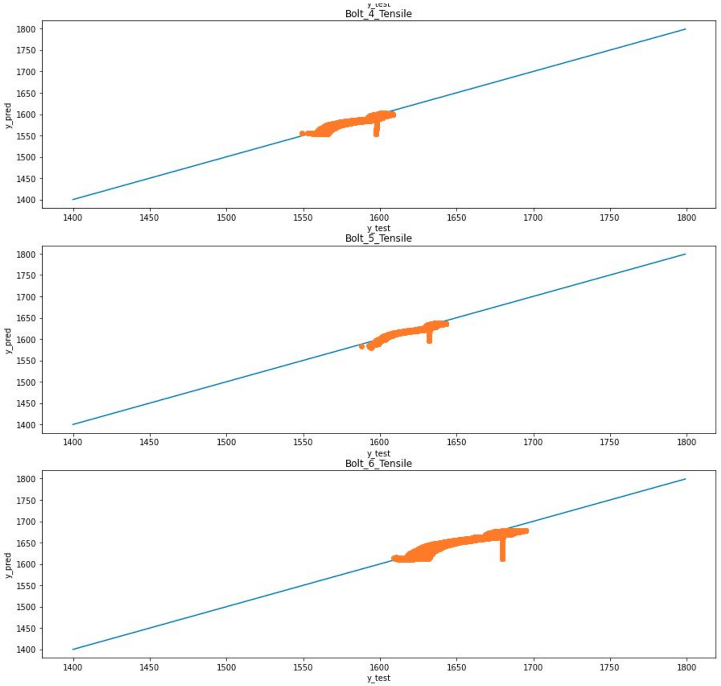
Example plot of true bolt tension vs. predicted bolt tension

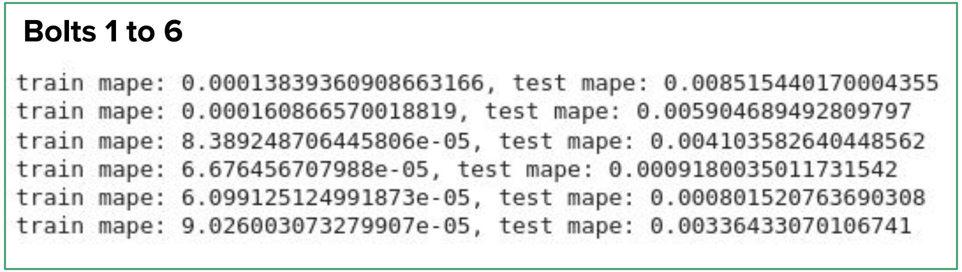
Overall Model Performance

- Good model performance found with Linear Regression + hand-crafted features for all bolts

- A high bias model was chosen also to help ensure robust future predictions of an unknown future and different environments (i.e. different units).

- Model can also be used across different units in Kvilldal as it based on scaled inputs (again the high bias ensures robust predictions).





Bolts 1 to 6

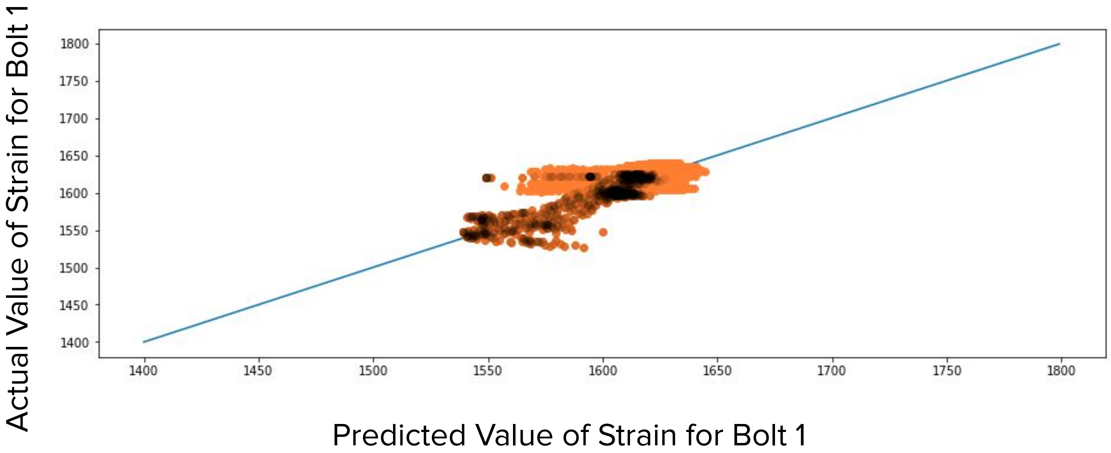
```
train mape: 0.00013839360908663166, test mape: 0.008515440170004355
train mape: 0.000160866570018819, test mape: 0.005904689492809797
train mape: 8.389248706445806e-05, test mape: 0.004103582640448562
train mape: 6.676456707988e-05, test mape: 0.000918003501173l542
train mape: 6.099125124991873e-05, test mape: 0.0008015207636903O8
train mape: 9.026003073279907e-05, test mape: 0.00336433070106741
```

Examining model performance - start-up perods

- Start-up periods had overall significantly poorer predictive performance compared to normal operation.

- We considered having separate models for different modes of operation but this was found to not significantly increase performance.

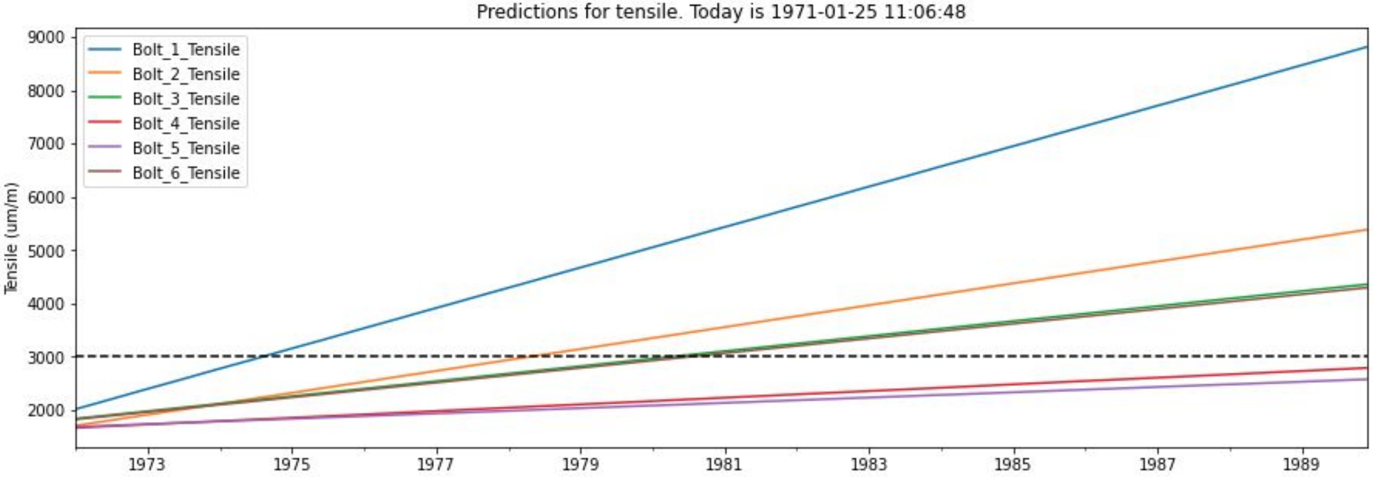
- It was presumed that this might be the case because:
  - There is another factor at play here which is not included in the data set;
  - Model readings for these regions are not reliable



Legend
Orange: Normal operation
Black: Start-up regions

End of Life Prediction

- Using our model, we can predict the end-of-life for each bolt!
- The following bolts will fail in the near future (to the nearest month):
  - Bolt_1_Tensile fails at 1974-08-31 00:00:00
  - Bolt_2_Tensile fails at 1978-05-31 00:00:00
  - Bolt_3_Tensile fails at 1980-04-30 00:00:00
  - Bolt_6_Tensile fails at 1980-07-31 00:00:00



Putting it into production

Things to consider

- Retraining
  - Online retraining when model performance degrades. The onset of automatic retraining could be determined from changepoint detection, bayesian modelling for understanding when physics or dynamics change, or by monitoring model performance vs. actual target values over time.
  - Offline: Local training and upload of pickled model based on predetermined intervals for retraining
  - Online retraining requires a lot more work upfront, while oflline retraining requires manual work on a scheduled basis.
- Fallback when database is not reliable or data not trustworthy
  - When the database is not able to provide data to for predictions this needs to be highlighted in the dashboard to alert the user to not trust results in this region

Transition from local notebook to an online machine learning model:

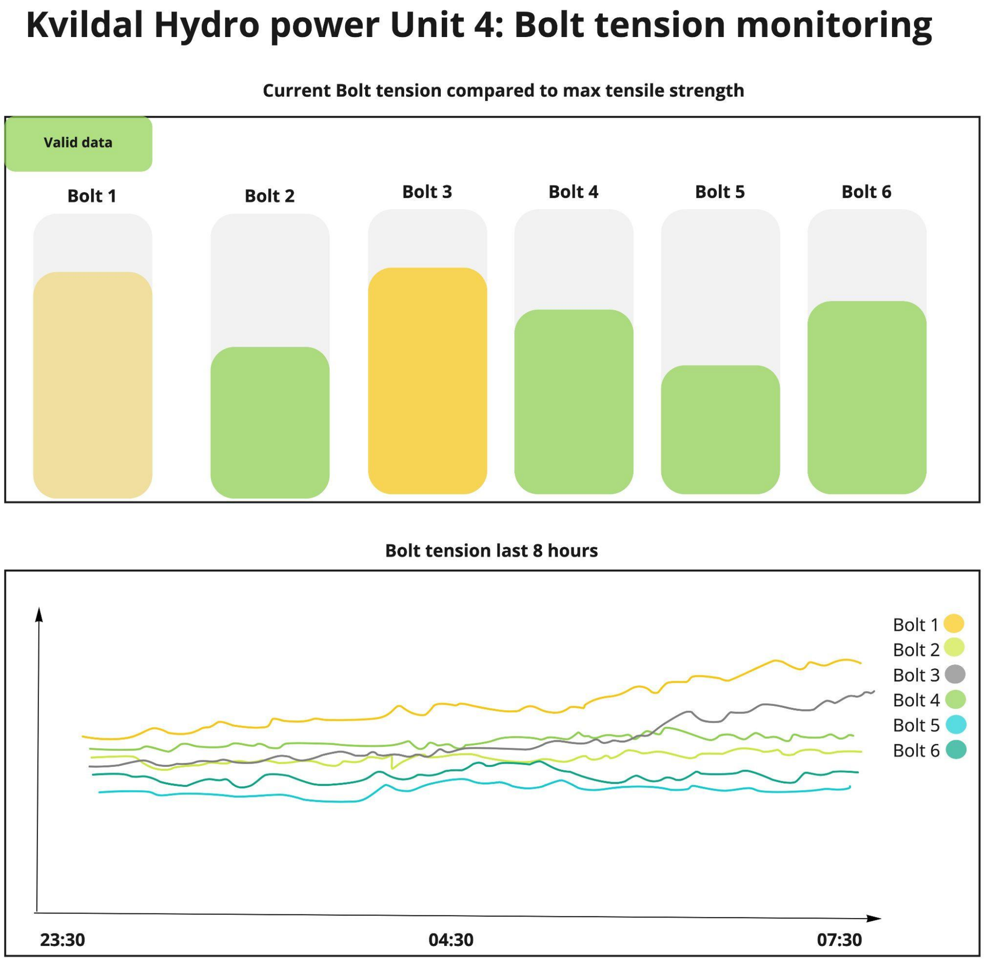
- Rewrite the python code in the notebook to python modules that is checked into github repository for source control. Writing tests to verify that the code works as expected is especially important when putting the model into production.
- Data from a time series database is streamed to the model where the data undergoes the same preprocessing and feature engineering pipeline and predictions are made with the ML model. The predictions will be written back to the time series database such that a dashboard can display the data.
- There are multiple alternatives for hosting the machine learning model: Functions as a Service vendors, e.g. Azure Functions, and a docker container + kubernetes are common methods for doing this.

Mock-up dashboard

Label indicating whether data is valid or not. Important for user to trust the output of the dashboard

Bars indicating current tension in bolts compared to the maximum tensile strength.

Line chart with graph of tension during the last 8 hours. The user would also be able to select another time window


Kvildal Hydro power Unit 4: Bolt tension monitoring
Current Bolt tension compared to max tensile strength
Valid data
Bolt 1
Bolt 2
Bolt 3
Bolt 4
Bolt 5
Bolt 6
Bolt tension last 8 hours
Bolt 1
Bolt 2
Bolt 3
Bolt 4
Bolt 5
Bolt 6
23:30
04:30
07:30

Scalability & Transferability

- Before scaling it is very important that the model and its predictions are validated offline and that Subject Matter Experts approve the results. Otherwise a lot of time will be wasted in scaling and putting the model into production when it is not valuable.

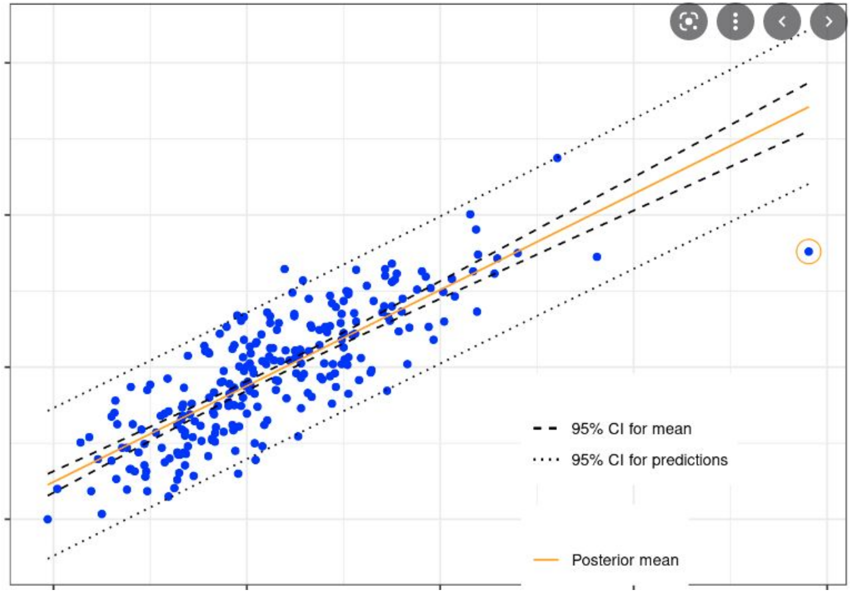
- Considering that the other generators has partly different equipment, both in age and type, it is recommended to have a safety factor on the predicted values for the other generators without tensile measurements for bolts.

Limitations/ Future Work

- Estimates for the future strain on the bolts were based on static values for the turbines. More work would need to be done to validate what turbine input values should be chosen for future prediction.

- The adoption of a Bayesian framework for regression would be able to estimate the 95% credible interval of the strain over time to give flexibility in decision-making.

- This will help select the most conservative time to failure.


95% CI for mean
95% CI for predictions
Posterior mean