Krafthack 2022

Broentech Solutions

Team and Responsibilities

Stian Broen - Frontend

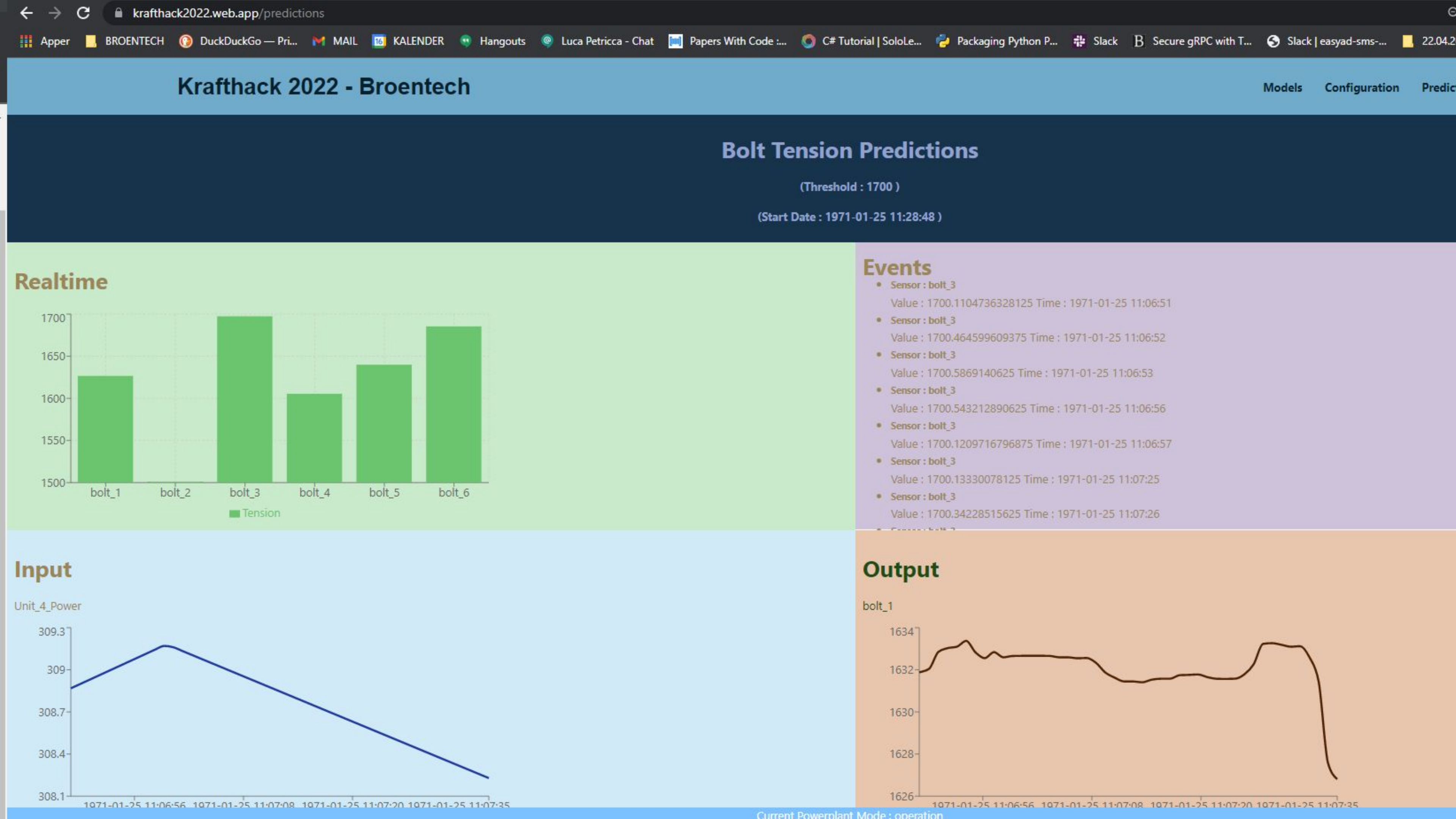
Luca Petricca - Backend

Duo Zhang - Prediction Models

Frontend

- React
- Redux
- Axios
- Socket-IO + REST

https://krafthack2022.web.app/



Krafthack 2022 - Broentech

Bolt Tension Predictions

(Threshold : 1700)

(Start Date : 1971-01-25 11:28:48)

Realtime

Events

- Sensor : bolt_3
  Value : 1700.1104736328125 Time : 1971-01-25 11:06:51
- Sensor : bolt_3
  Value : 1700.464599609375 Time : 1971-01-25 11:06:52
- Sensor : bolt_3
  Value : 1700.5869140625 Time : 1971-01-25 11:06:53
- Sensor : bolt_3
  Value : 1700.543212890625 Time : 1971-01-25 11:06:56
- Sensor : bolt_3
  Value : 1700.1209716796875 Time : 1971-01-25 11:06:57
- Sensor : bolt_3
  Value : 1700.13330078125 Time : 1971-01-25 11:07:25
- Sensor : bolt_3
  Value : 1700.34228515625 Time : 1971-01-25 11:07:26

Input

Unit_4_Power

Output

bolt_1

Current Powerplant Mode : operation

Backend

- Python
- AioHTTP (REST + Socket-IO)
- Deployed using Cloud Run
- Authentication using Firebase

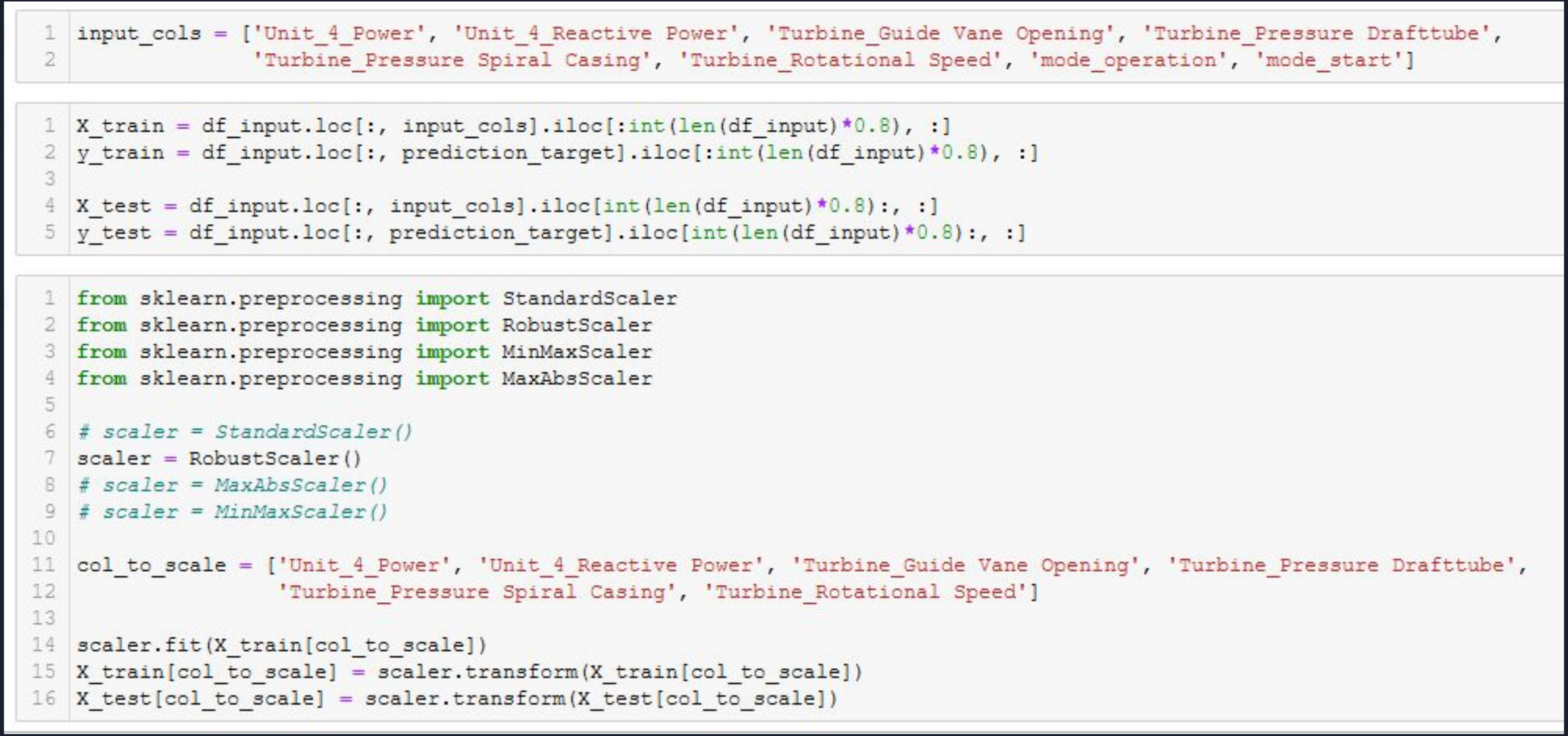
https://krafthack2022.web.app/

Model

- Keeping things easy (little time)
- Considered :
  - Support Vector Regression (SVG)
  - GAN (Generative Adversarial Network)
  - XG-Boost (the chosen one)
- XG-Boost training finished in reasonable time

https://krafthack2022.web.app/

Data Prep



```python
input_cols = ['Unit_4_Power', 'Unit_4_Reactive Power', 'Turbine_Guide Vane Opening', 'Turbine_Pressure Drafttube',
              'Turbine_Pressure Spiral Casing', 'Turbine_Rotational Speed', 'mode_operation', 'mode_start']
```

```python
X_train = df_input.loc[:, input_cols].iloc[:int(len(df_input)*0.8), :]
y_train = df_input.loc[:, prediction_target].iloc[:int(len(df_input)*0.8), :]

X_test = df_input.loc[:, input_cols].iloc[int(len(df_input)*0.8):, :]
y_test = df_input.loc[:, prediction_target].iloc[int(len(df_input)*0.8):, :]
```

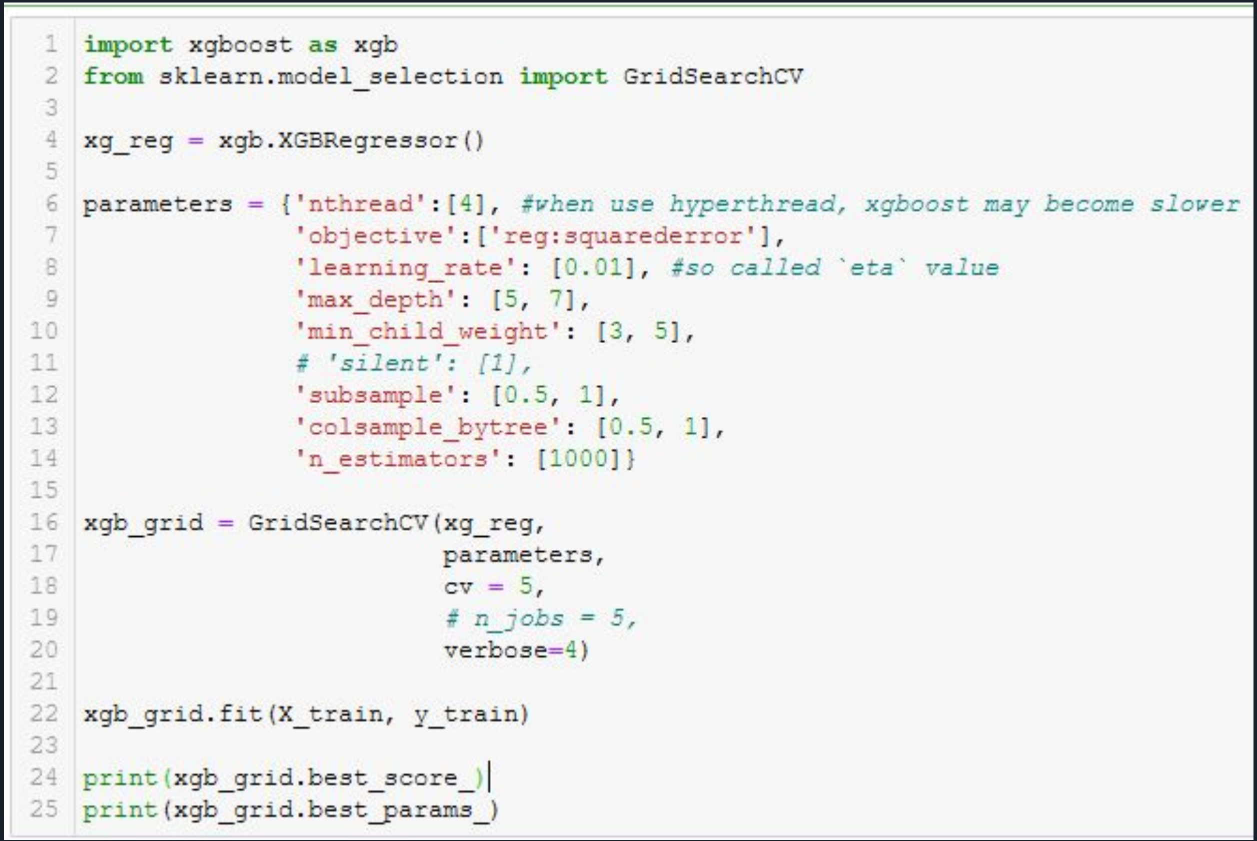
```python
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import RobustScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import MaxAbsScaler

# scaler = StandardScaler()
scaler = RobustScaler()
# scaler = MaxAbsScaler()
# scaler = MinMaxScaler()

col_to_scale = ['Unit_4_Power', 'Unit_4_Reactive Power', 'Turbine_Guide Vane Opening', 'Turbine_Pressure Drafttube',
                'Turbine_Pressure Spiral Casing', 'Turbine_Rotational Speed']

scaler.fit(X_train[col_to_scale])
X_train[col_to_scale] = scaler.transform(X_train[col_to_scale])
X_test[col_to_scale] = scaler.transform(X_test[col_to_scale])
```

Training the Model



```python
import xgboost as xgb
from sklearn.model_selection import GridSearchCV

xg_reg = xgb.XGBRegressor()

parameters = {'nthread':[4], #when use hyperthread, xgboost may become slower
              'objective':['reg:squarederror'],
              'learning_rate': [0.01], #so called `eta` value
              'max_depth': [5, 7],
              'min_child_weight': [3, 5],
              # 'silent': [1],
              'subsample': [0.5, 1],
              'colsample_bytree': [0.5, 1],
              'n_estimators': [1000]}

xgb_grid = GridSearchCV(xg_reg,
                        parameters,
                        cv = 5,
                        # n_jobs = 5,
                        verbose=4)

xgb_grid.fit(X_train, y_train)

print(xgb_grid.best_score_)
print(xgb_grid.best_params_)
```

Thanks =)

App : **https://krafthack2022.web.app/**

Repo : **https://github.com/Broentech/krafthack2022**