Wyjaśnienia modelu

In [ ]:

```
import dalex as dx
import pickle
import os

import pandas as pd
import numpy as np
from scipy import stats

from sklearn.pipeline import Pipeline
import xgboost as xgb
import plotly

plotly.offline.init_notebook_mode()
```

Przygotowanie danych dla modelu

In [ ]:

```
# a directory with all gz_files like 'like 'avg_from_2020_10_01_00_00_00_to_2020_10_01_23_59_00.gz'dirpath = 'data'
dirpath = 'data'
df = pd.concat([pd.read_csv(os.path.join(dirpath, fname))
                for fname in os.listdir(dirpath)], ignore_index=True)

df['czas'] = df['czas'].str[:19]
df['czas'] = pd.to_datetime(df['czas'], format='%Y-%m-%d %H:%M:%S')

col_df = pd.read_excel('opis_zmiennych.xlsx')
col_df['opis'] = col_df['opis'] + ' ' + col_df['Jednostka']
col_df.drop(columns=['Jednostka'], inplace=True)

names_dict = col_df.set_index('Tagname').to_dict()['opis']
names_dict =  {k.lower(): v for k, v in names_dict.items()}

df.rename(columns=names_dict, inplace=True)

temp = pd.read_csv('temp_zuz.csv', delimiter=';')
temp.rename(columns={'Czas': 'czas'}, inplace=True)
```

```
temp['czas'] = pd.to_datetime(temp['czas'])

merged = pd.merge(df, temp, how='left', on='czas')
df = merged.copy()
df.set_index(['czas'], inplace=True)
df = df[~df.index.duplicated()]
df = df.asfreq('T')
```

In [ ]:

```
def last_h(df, n_hours):
    temp = df['temp_zuz'].copy()
    for n in range(1, n_hours+1):
        label = 'temp_last' + '_' + str(n)
        df[label] = temp.fillna(method='ffill').shift(periods=1+60*(n-1))
```

In [ ]:

```
last_h(df, 4)

cols = list(df.columns)
cols.remove('temp_zuz')

df = df.dropna(subset=cols)

col_names = ['REG NADAWY KONCENTRATU LIW1 Mg/h', 'REG NADAWY KONCENTRATU LIW2 Mg/h',
       'REG KONCENTRAT PRAZONY LIW3 Mg/h', 'REG PYL ZWROT LIW4 Mg/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ7 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ8 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ9 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ10 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ11 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ12 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ13 m3/h',
       'WODA CHŁODZĄCA DO KOLEKTOR KZ15 m3/h',
       'SUMARYCZNA MOC CIEPLNA ODEBRANA - CAŁKOWITA MW',
       'WODA POWROTNA KOLEKTORA KZ7 °C', 'WODA POWROTNA KOLEKTORA KZ8 °C',
       'WODA POWROTNA KOLEKTORA KZ9 °C']

new_df = df.copy()

new_df = pd.concat([new_df, new_df[col_names].rename(columns=lambda col_name: f'{col_name}_avg_00-15').rolling(window=15).mean()],
new_df = pd.concat([new_df, new_df[col_names].shift(periods=15, freq='min').rename(columns=lambda col_name: f'{col_name}_avg_15-30
new_df = pd.concat([new_df, new_df[col_names].shift(periods=30, freq='min').rename(columns=lambda col_name: f'{col_name}_avg_30-45
new_df = pd.concat([new_df, new_df[col_names].shift(periods=45, freq='min').rename(columns=lambda col_name: f'{col_name}_avg_45-60
```

```
threshold = 0.165

correlated_cols = new_df.columns[new_df.corr()['temp_zuz'].abs() > threshold].tolist()
corr_df = new_df[correlated_cols].copy()

corr_df['temp_zuz'] = corr_df['temp_zuz'].interpolate()
df = corr_df.dropna()

df = df[(np.abs(stats.zscore(df)) < 3).all(axis=1)]

df['minute'] = df.index.minute.values
df['minute'] = np.where(df['minute'] == 0, 60, df['minute'])
```

Wczytanie wytrenowanego modelu

In [ ]:
```
model = pickle.load(open('model.sav', 'rb'))
```

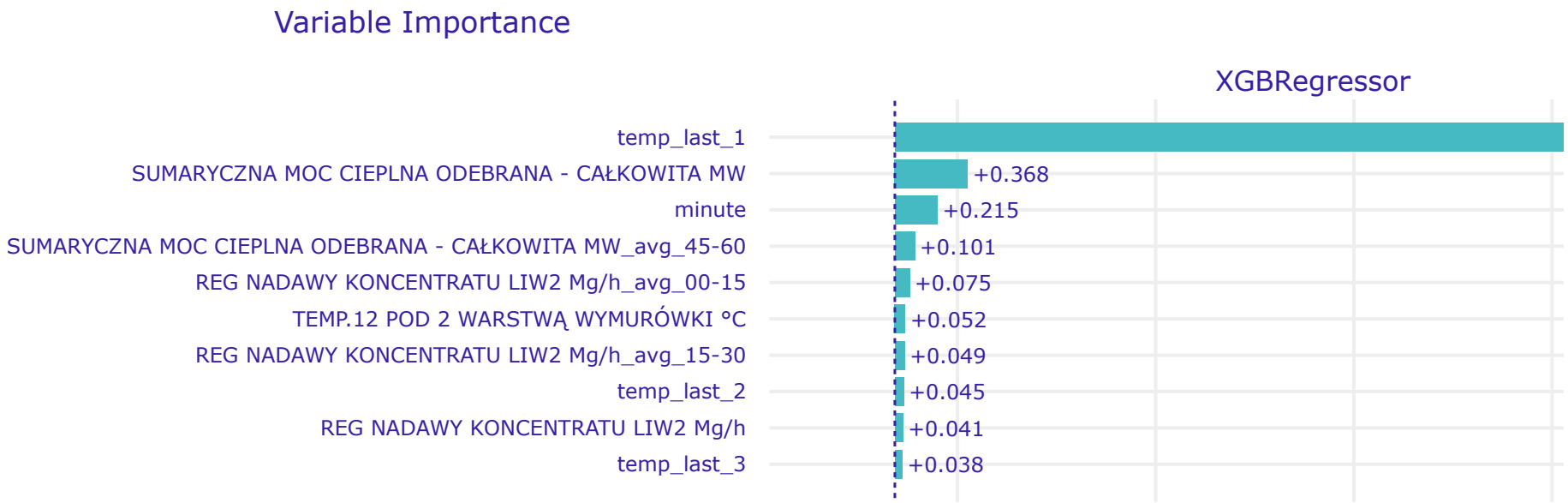
In [ ]:
```
# we create model explainer object
explainer = dx.Explainer(model, df.drop(columns=['temp_zuz']), df['temp_zuz'])
```

```
Preparation of a new explainer is initiated

  -> data              : 639786 rows 67 cols
  -> target variable   : Parameter 'y' was a pandas.Series. Converted to a numpy.ndarray.
  -> target variable   : 639786 values
  -> model_class       : xgboost.sklearn.XGBRegressor (default)
  -> label             : Not specified, model's class short name will be used. (default)
  -> predict function  : <function yhat_default at 0x000002580FFD6040> will be used (default)
D:\Programy\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning:

X does not have valid feature names, but StandardScaler was fitted with feature names

  -> predict function  : Accepts pandas.DataFrame and numpy.ndarray.
  -> predicted values  : min = 1.27e+03, mean = 1.3e+03, max = 1.33e+03
  -> model type        : regression will be used (default)
  -> residual function : difference between y and yhat (default)
  -> residuals         : min = -29.8, mean = 0.0346, max = 22.8
  -> model_info        : package sklearn
```

```
A new explainer has been created!
```

Analiza kontrybucji zmiennych

In [ ]:
```
permutatational_variable_importance = explainer.model_parts()
permutatational_variable_importance.plot()
```



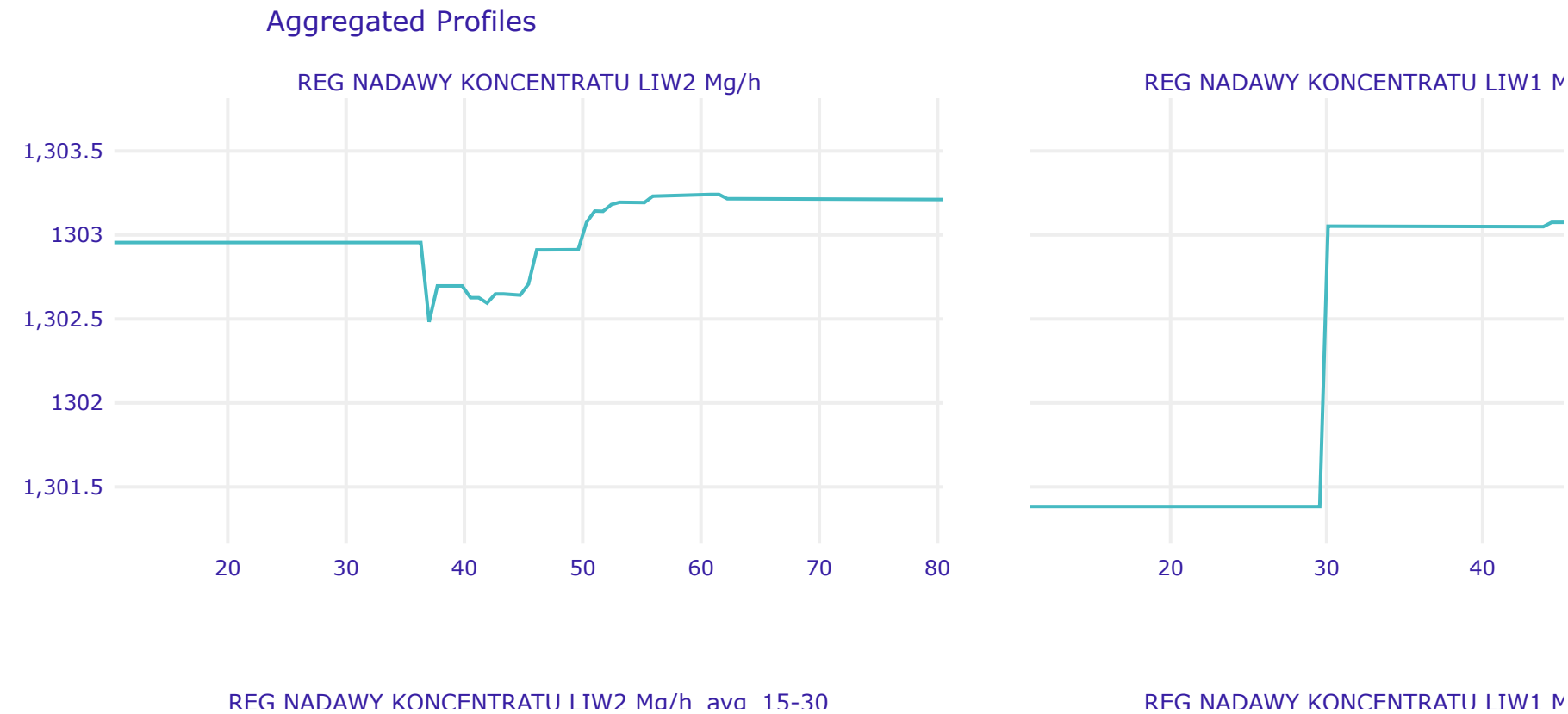
Widzimy, że wyuczony przez nas model, tak jak oczekiwaliśmy, dużą uwagę zwraca na dane z ostatniego pomiaru temperatury dla czasu w którym dokonujemy predykcji. Duży wkład do predykcji modelu mają również wartości m.in. sumarycznych strat ciepła w procesie oraz uśrenionych wartości wrzucanej do pieca nadawy koncentratu miedzi. Z przeprowadzoncyh przez nas analiz wynikało, że te zmienne mogą mieć duży wpływ na modelowanie temperatury.

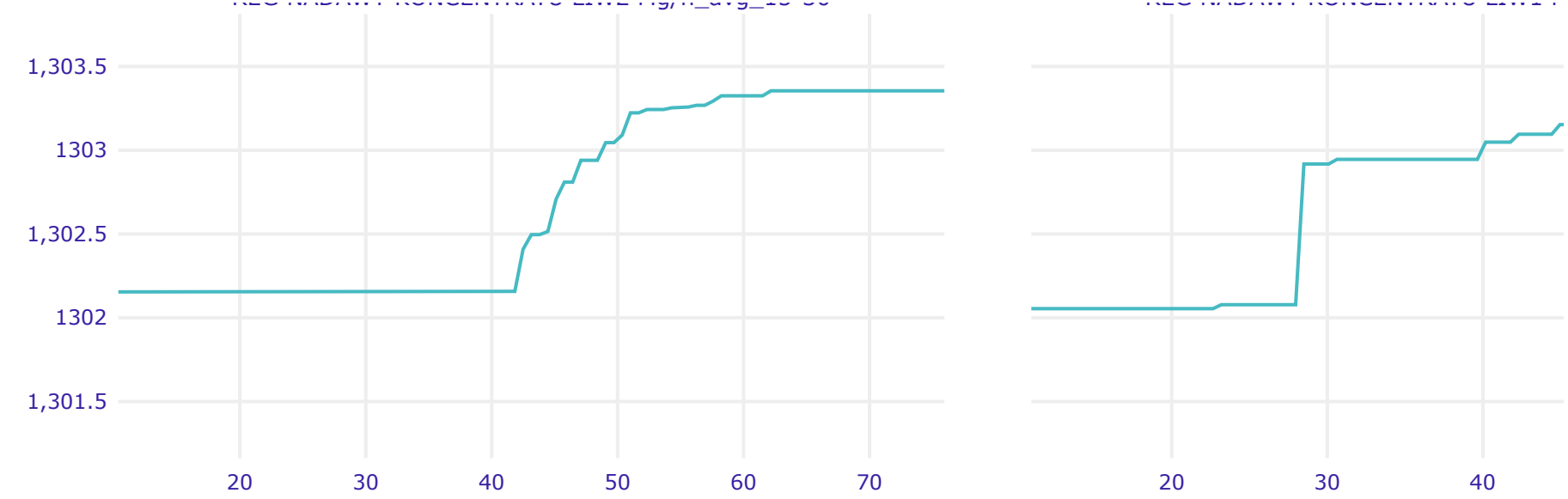
Analiza profili Ceteris Paribus

In [ ]:

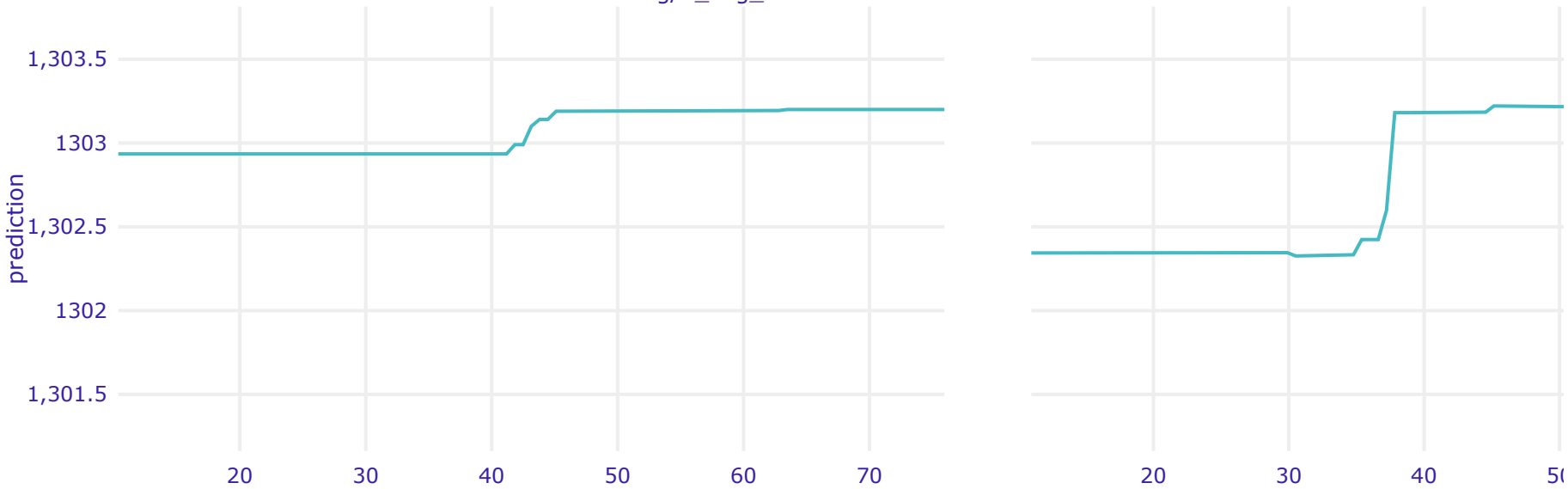
```
partial_dependence_profiles = explainer.model_profile(variables=[
        'REG NADAWY KONCENTRATU LIW2 Mg/h',
        'REG NADAWY KONCENTRATU LIW1 Mg/h_avg_15-30',
        'REG NADAWY KONCENTRATU LIW2 Mg/h_avg_15-30',
        'REG NADAWY KONCENTRATU LIW1 Mg/h_avg_30-45',
        'REG NADAWY KONCENTRATU LIW2 Mg/h_avg_30-45',
        'REG NADAWY KONCENTRATU LIW1 Mg/h_avg_45-60',
        'REG NADAWY KONCENTRATU LIW2 Mg/h_avg_45-60',
        'SUMARYCZNA MOC CIEPLNA ODEBRANA - CAŁKOWITA MW_avg_15-30',
        'SUMARYCZNA MOC CIEPLNA ODEBRANA - CAŁKOWITA MW_avg_30-45',
        'SUMARYCZNA MOC CIEPLNA ODEBRANA - CAŁKOWITA MW_avg_45-60'

    ], type='partial')

partial_dependence_profiles.plot()
```

```
Calculating ceteris paribus: 100%|████████| 10/10 [00:01<00:00,  7.60it/s]
```

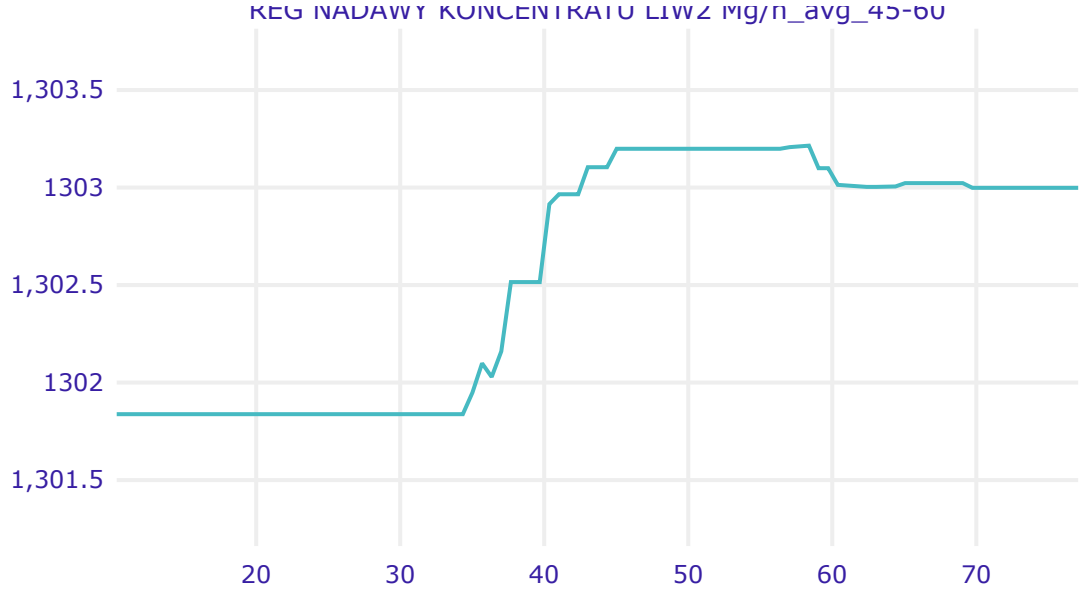


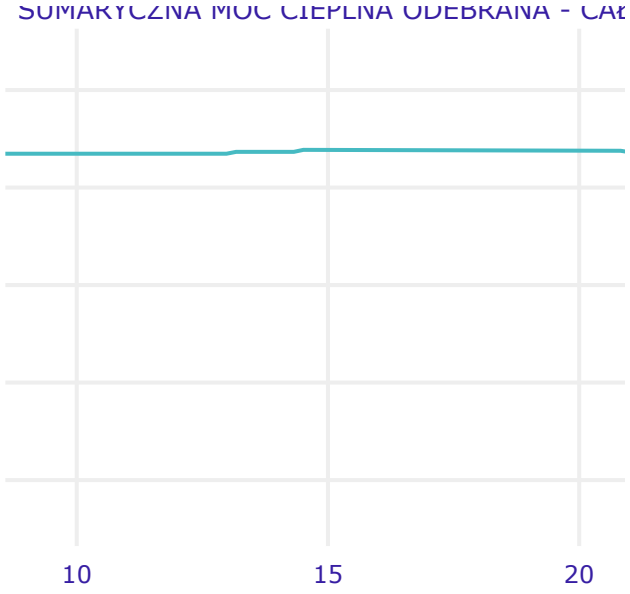


REG NADAWY KONCENTRATU LIW2 Mg/h_avg_30-45



REG NADAWY KONCENTRATU LIW1 M





SUMARYCZNA MOC CIEPLNA ODEBRANA - CAŁKOWITA MW_avg_30-45

SUMARYCZNA MOC CIEPLNA ODEBRANA - CAł

Na powyższych wykresach przedstawione zostały profile wybranych zmiennych, przedstawiające w jaki sposób przy manipulowaniu powyższymi zmiennymi zmieniać się będzie temperatura wenątrz pieca. Widzimy między innymi zgodną z rzeczywistością zależność zwiększania nadawy koncentratu miedzi a wzrostem temperatury pieca, czy proporcjonalność temperatury do sumarycznej mocy ciepła odbieranej w procesie.

Przykładowe wyjaśnienie przykładowych predykcji

In [ ]:
```
# for i in range(5):
#     break_down = explainer.predict_parts(df[df.index.minute == 0].sample().drop(columns=['temp_zuz']), type='break_down')
#     break_down.plot()
```

In [ ]:
```
# shap = explainer.predict_parts(df[df.index.minute == 0].sample().drop(columns=['temp_zuz']), type='shap')
# shap.plot()
```

Na powyższych wykresach przedstawione zostały jakie cechy w jaki sposób wpłynęły na predykcję temperatury dla losowych próbek.