Vez?rl? log megjelen?t?s HoloViews modellel

December 23, 2019

1 Adatvizualizáció, elemzés

Az elfeldolgozott adatokkal egy interaktív vizualizációs megoldással egy hatékony elemzési környezetet alakítunk ki.

Az elemzéshez a Holoviews modult használjuk. A megközelítés lényege, hogy egy wrappert tesz az adatok köré, amely metaadatokkal leírja az adatokat. A mechanizmus lehetséget ad rá, hogy az adatokat annotálva, az adatokra koncetrálva interaktív vizualizációs környezetben kísérletezgessünk.

Ebben a notebookban ezt a környezetet építjük ki. A megközelítés hatékonyságát jól mutatja, hogy nagyon tömör kóddal lehet eljutni interaktív megjelenít komponensekig.

A rendszer mögött a korábban is használt bokeh backend található.

Elször a szükséges importokat hajtjuk végre.

[1]:
```
import pandas as pd
import holoviews as hv
import param
import panel as pn

from holoviews.operation.timeseries import rolling, rolling_outlier_std

hv.extension('bokeh')
```

A korábbi lépésben elfeldolgozott adatokat fileból beolvassuk és elkészítjük a feldolgozáshoz.

[2]:
```
df = pd.read_csv('preprocessed_data_080910.csv', sep=';')
df['datetime'] = pd.to_datetime(df['datetime'])
df['datetime_index'] = pd.to_datetime(df['datetime_index'])
df = df.set_index('datetime_index')
```

Leellenrizzük a betöltött adatokat.

[3]:
```
df
```

[3]:

datetime_index	Value	datetime	Module	IO	NVariable
2019-08-18 11:37:07	0.00	2019-08-18 11:37:07	f	i	kaz
2019-08-18 11:37:07	0.00	2019-08-18 11:37:07	f	i	er_kaz
2019-08-18 11:37:07	0.00	2019-08-18 11:37:07	f	i	pa4
2019-08-18 11:37:07	0.00	2019-08-18 11:37:07	f	i	pa2
2019-08-18 11:37:07	0.00	2019-08-18 11:37:07	f	i	ap1

```
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  i        ap3
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  i        ep1
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  i        ep3
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  i        tp1
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   sziv_kaz
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o    sziv_pf
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   sziv_rad
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   sziv_hmv
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   sziv_jac
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   sziv_kal
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o     pf_spa
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o    hmvcirk
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o      hmvuz
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o  rad_phalo
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   tsz_pfur
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o  rad_shalo
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o    pf_pfur
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o    pf_sfur
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o      pf_tt
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o   rad_vend
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o    pf_vfur
2019-08-18 11:37:07   0.00 2019-08-18 11:37:07      f  o     pf_nap
2019-08-18 11:38:00  23.81 2019-08-18 11:38:00      f  i        pa4
2019-08-18 11:38:00  23.31 2019-08-18 11:38:00      f  i        pa2
2019-08-18 11:38:00  26.50 2019-08-18 11:38:00      f  i        ap1
...                    ...                 ...    ... ..        ...
2019-10-31 23:48:00  20.12 2019-10-31 23:48:00      f  i        ap3
2019-10-31 23:48:00  23.06 2019-10-31 23:48:00      f  i        ep3
2019-10-31 23:49:00  21.93 2019-10-31 23:49:00      f  i        ap1
2019-10-31 23:49:00  20.18 2019-10-31 23:49:00      f  i        ap3
2019-10-31 23:49:00  23.12 2019-10-31 23:49:00      f  i        ep3
2019-10-31 23:50:00  20.93 2019-10-31 23:50:00      f  i        pa4
2019-10-31 23:50:00  23.06 2019-10-31 23:50:00      f  i        ep3
2019-10-31 23:50:00   0.00 2019-10-31 23:50:00      f  o    hmvcirk
2019-10-31 23:50:00   0.00 2019-10-31 23:50:00      f  o   tsz_pfur
2019-10-31 23:50:00   0.00 2019-10-31 23:50:00      f  o   sziv_rad
2019-10-31 23:50:00   0.00 2019-10-31 23:50:00      f  o   sziv_kaz
2019-10-31 23:52:00  20.87 2019-10-31 23:52:00      f  i        pa4
2019-10-31 23:52:00  20.12 2019-10-31 23:52:00      f  i        ap3
2019-10-31 23:52:00   1.00 2019-10-31 23:52:00      f  o      hmvuz
2019-10-31 23:52:00   1.00 2019-10-31 23:52:00      f  o   sziv_hmv
2019-10-31 23:52:00   1.00 2019-10-31 23:52:00      f  o   sziv_kaz
2019-10-31 23:53:00  22.00 2019-10-31 23:53:00      f  i        ap1
2019-10-31 23:53:00  23.12 2019-10-31 23:53:00      f  i        ep3
2019-10-31 23:54:00  21.93 2019-10-31 23:54:00      f  i        ap1
2019-10-31 23:56:00  20.93 2019-10-31 23:56:00      f  i        pa4
2019-10-31 23:56:00  20.18 2019-10-31 23:56:00      f  i        ap3
```

```
2019-10-31 23:56:00  23.06 2019-10-31 23:56:00      f  i        ep3
2019-10-31 23:57:00  22.00 2019-10-31 23:57:00      f  i        ap1
2019-10-31 23:57:00  20.12 2019-10-31 23:57:00      f  i        ap3
2019-10-31 23:58:00  20.87 2019-10-31 23:58:00      f  i        pa4
2019-10-31 23:58:00  21.93 2019-10-31 23:58:00      f  i        ap1
2019-10-31 23:59:00  20.93 2019-10-31 23:59:00      f  i        pa4
2019-10-31 23:59:00   0.00 2019-10-31 23:59:00      f  o      hmvuz
2019-10-31 23:59:00   0.00 2019-10-31 23:59:00      f  o   sziv_hmv
2019-10-31 23:59:00   0.00 2019-10-31 23:59:00      f  o   sziv_kaz

[189957 rows x 5 columns]
```

Mivel a Holoviews komponensek úgy kezelik hatékonyan a beágyazott adatokat, ha az elemzend adatok külön oszlopban, dimenzióban jelennek meg, így egy kis elfeldolgozást kell tennünk.

Az NVariable oszlopban vannak a változó nevek. Ezt leszrve, egyediséget biztosítva kapjuk meg az oszlopneveket. Az új oszlopokat létrehozva egyszeren ezen oszlopokba bemásoljuk a Value oszlop értékét, ami nem más, mint a változó értéke.

[4]:
```
variables = df.NVariable.unique().tolist()
new_df = df
for variable in variables:
    new_df[variable] = df['Value']
```

Ellenrizzük az új dataframe alakját.

[5]:
```
new_df.shape
```

[5]:
```
(189957, 32)
```

1.1 Megjelenítés elemzéshez

A Holoviews modell a háttérben gondoskodik a megfelel adatkezelésrl. Az elemzés támogatásához a megjelenítési területet két oszlopra osztjuk: 1. kontrol blokk, amelyben megjelennek az elemzést támogató kontrollok - A megjelenítend változó típusa (Input vagy Output) - A modul változó neve - Dátum intervallum

```
  (itt lehet szkíteni a megjelenítend adatintervallumot) Hasonló funkció magán a grafikonon is
- Gördülö ablakhossz
  Ez a gördül ablakos jelfeldolgozásokhoz (pl. átlagolás, kiugró értékek keresése) adhat bemen
```

2. Grafikon blokk

   Itt jelenik meg a kiválasztott változó grafikonja. A megjelenítéssel gyors alap elemzések hajthatóak végre, gyorsan áttekinthetek a változók, vizuálisan elemezhetek a jelformák, a hiányzó értékek, a kiugró értékek.

Minden egyes kontrollt definiálunk, mint widgetet, a megfelel típussal és alapértelmezett értékekkel.

[6]:
```
width = 200
w_iotype = pn.widgets.RadioButtonGroup(name='Input/Output', options=['Input',␣
↪'Output'], width=width)
```

```
w_variable = pn.widgets.Select(name='Modul', options=variables, width=width)
w_daterange = pn.widgets.DateRangeSlider(name='Dátum', value=(new_df.index[0],␣
↪new_df.index[len(new_df)-1]), start=new_df.index[0], end=new_df.
↪index[len(new_df)-1], width=width)
w_rolling_window = pn.widgets.IntSlider(name='Gördül ablak ablakhossz',␣
↪value=10, start=1, end=30, width=width)
```

Definiálunk egy függvényt, amely callbackként viselkedik. A widgetek értékei alapján állítjuk be a megfelel értékeket.

Az Input/Output választó értéke alapján dinamikusan tültjük fel a változó listát (legördül liste), illetve a beállított idintervallumon belüli adatokra szkítjük a megjelenítést.

[7]:
```
@pn.depends(iotype=w_iotype, variable=w_variable, daterange=w_daterange,␣
↪rolling_window=w_rolling_window)
def load_variable(iotype, variable, daterange, rolling_window):
    io_type = 'i'
    if iotype == 'Input':
        io_type = 'i'
    else:
        io_type = 'o'

    variables = new_df.loc[new_df['IO'] == io_type].NVariable.unique().tolist()
    w_variable.options = variables
    dfl = new_df.loc[new_df['NVariable'] == variable]
    dfl = dfl.loc[(dfl.index >= daterange[0]) & (dfl.index <= daterange[1])]
    return hv.Curve(dfl, ('datetime', 'Date'), variable).opts(framewise=True)
```

Létrehozzuk a megfelel Holoviews objektumot, ami jelen esetben egy DynamicMap.

[8]:
```
log_dmap = hv.DynamicMap(load_variable)
```

Megadjuk a layoutot, ami jelen esetben egy Row layout, ami két oszlopot tartalmaz:

1. WidgetBox
2. DynamicMap Holoviews objektum

Ekkor meg is jelenik az vizualizáció, a kontrollokkal kísérletezve adat-elelemzés hajtható végre.

[9]:
```
pn.Row(pn.WidgetBox('## Log Explorer', w_iotype, w_variable, w_daterange,␣
↪w_rolling_window), log_dmap.opts(width=600, padding=0.1))
```

[9]:
```
Row
    [0] WidgetBox(css_classes=['widget-box'])
        [0] Markdown(str)
        [1] RadioButtonGroup(name='Input/Output', options=['Input', 'Output'],
value='Input', width=200)
        [2] Select(name='Modul', options=['kaz', 'er_kaz', ...], value='kaz',
width=200)
        [3] DateRangeSlider(end=Timestamp('2019-10-31 2..., name='Dátum',
start=Timestamp('2019-08-18 1..., value=(Timestamp('2019-08-18 11:...,
width=200)
```

```
        [4] IntSlider(end=30, name='Gördül ablak a..., start=1, value=10,
width=200)
    [1] HoloViews(DynamicMap)
```

Az alap megjelenítés mellé érdemes egy olyan megjelenítést tenni, amelyben kísérletezni lehet a simításokkal, alap jelfeldolgozásokkal, illetve a kiugró értékek megjelenítésével.

A kontrollok, a megjelenítés létrehozása megegyezik a korábbival.

[10]:
```
width = 200
w2_iotype = pn.widgets.RadioButtonGroup(name='Input/Output', options=['Input',␣
↪'Output'], width=width)
w2_variable = pn.widgets.Select(name='Modul', options=variables, width=width)
w2_daterange = pn.widgets.DateRangeSlider(name='Dátum', value=(new_df.index[0],␣
↪new_df.index[len(new_df)-1]), start=new_df.index[0], end=new_df.
↪index[len(new_df)-1], width=width)
w2_rolling_window = pn.widgets.IntSlider(name='Gördül ablak ablakhossz',␣
↪value=1000, start=1, end=100000, width=width)
```

A callback függvény kiegészül a görget ablak alapján történ adatfeldolgozással.

[11]:
```
@pn.depends(iotype=w2_iotype, variable=w2_variable, daterange=w2_daterange,␣
↪rolling_window=w2_rolling_window)
def load_variable2(iotype, variable, daterange, rolling_window):
    io_type = 'i'
    if iotype == 'Input':
        io_type = 'i'
    else:
        io_type = 'o'

    variables = new_df.loc[new_df['IO'] == io_type].NVariable.unique().tolist()
    w2_variable.options = variables
    dfl = new_df.loc[new_df['NVariable'] == variable]
    dfl = dfl.loc[(dfl.index >= daterange[0]) & (dfl.index <= daterange[1])]
    #dfl[variable] =  dfl[variable] * rolling_window
    return hv.Curve(dfl, ('datetime', 'Date'), variable).opts(framewise=True)
```

[12]:
```
log2_dmap = hv.DynamicMap(load_variable2)
```

Kiszámítjuk a gördül átlagot a megfelel gördül ablak paraméter felhasználásával.

[13]:
```
# Gördül átlag
smoothed = rolling(log2_dmap, rolling_window=w2_rolling_window.value).opts(
    color='green', marker='circle')
```

Kiszámítjuk a kiugró értékeket, a megfelel gördül ablak paraméter felhasználásával.

[14]:
```
# Kiugró értékek
outliers = rolling_outlier_std(log2_dmap, rolling_window=w2_rolling_window.
↪value).opts(
    color='red', marker='triangle')
```

Megjelenítjük a kontrollokat és a grafikont.

[15]:
```
pn.Row(pn.WidgetBox('## Log Explorer', w2_iotype, w2_variable, w2_daterange,␣
 ↪w2_rolling_window), (log2_dmap * outliers).opts(width=600, padding=0.1))
```

[15]:
```
Row
    [0] WidgetBox(css_classes=['widget-box'])
        [0] Markdown(str)
        [1] RadioButtonGroup(name='Input/Output', options=['Input', 'Output'],
value='Input', width=200)
        [2] Select(name='Modul', options=['kaz', 'er_kaz', ...], value='kaz',
width=200)
        [3] DateRangeSlider(end=Timestamp('2019-10-31 2..., name='Dátum',
start=Timestamp('2019-08-18 1..., value=(Timestamp('2019-08-18 11:...,
width=200)
        [4] IntSlider(end=100000, name='Gördül ablak a..., start=1, value=1000,
width=200)
    [1] HoloViews(DynamicMap)
```