CSE 4939W Week 11 (Spring 2025) - Sage Pia

2/10/2025

This week, I researched Grafana. For the front-end display, Grafana has many tools for visualizing different types of data. Our project would mainly revolve around the Time Series display, which allows multiple data ranges to be plotted on a graph. From a backend perspective, Grafana does not seem to perform any direct calculations, thereby required all measurements to be precomputed. Fortunately, Grafana has a wide range of data source options. The Infinity plugin processes CSV, JSON, XML, and other common datatypes. As proof of concept, I built a basic interface through flask. The workflow is as follows:

1) The user uploads a file through Flask
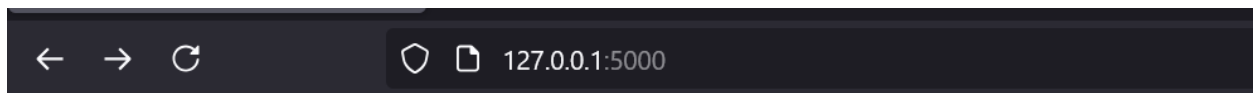
Time Series Analysis Data Upload

Upload compatible data file here (csv, excel file, etc.):

Time Data Field:
[date]
Target Data Field:
[OT]
[Browse...] ETTh1.csv
[Upload File]

Alternatively, submit text as input in csv format:

Time Data Field:
[ ]
Target Data Field:
[ ]
Raw data:
[ ]
[Submit Query]

Data can either be a file or submitted as an inline csv file. For now, only csv is supported. I require a time field for indexing and a target data field to forecast over.
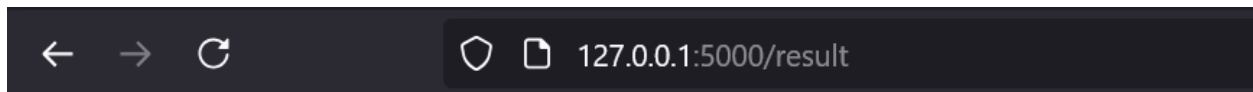
2) The user presses a button to run time series tasks.

← → C   ◯ ▢ 127.0.0.1:5000
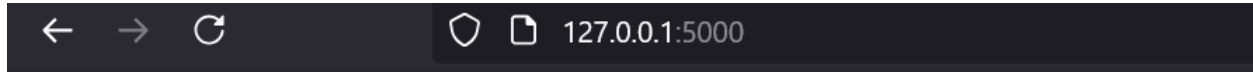
File ETTh1.csv successfully uploaded

Click here to generate time series data:

Generate Time Series Results

For now, this only does zero shot forecasting with the Chronos Tiny Bolt model.

3) A link to Grafana is given.

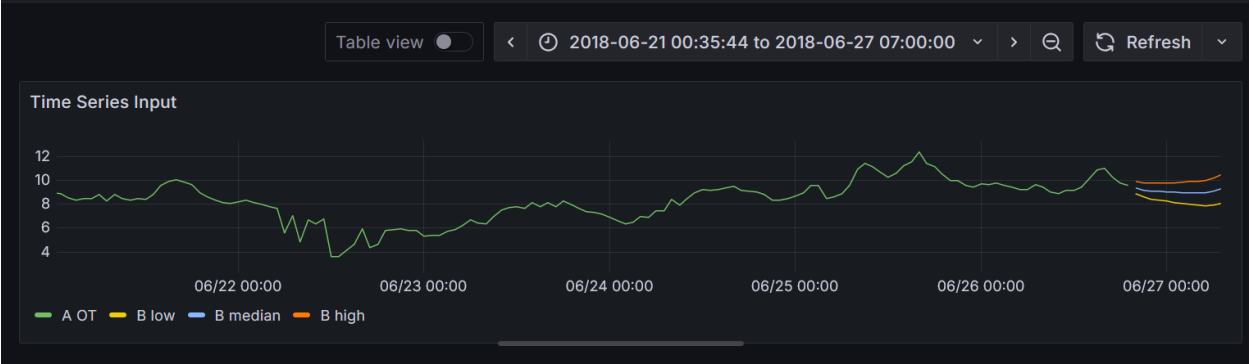← → C   ◯ ▢ 127.0.0.1:5000/result

Click below to view results in Grafana

Link

This links to my local Grafana homepage. I will need to see how to make a Grafana dashboard public.

4) Grafana collects and displays data from my local Flask app.

Time Series Input

Table view ⬤

‹ 🕐 2018-06-21 00:35:44 to 2018-06-27 07:00:00 ⌄ › ⊖ 🔄 Refresh ⌄

12
10
8
6
4

06/22 00:00   06/23 00:00   06/24 00:00   06/25 00:00   06/26 00:00   06/27 00:00

━ A OT   ━ B low   ━ B median   ━ B high

⌄ A   (Infinity)   ⓘ ⧉ 👁 🗑 ⠿

| Type | CSV ⌄ | Parser Try backend parser! | Default ⌄ | Source | URL ⌄ |

| Format | Data Frame ⌄ | 🗎 Help | Github |

| Method | GET ⌄ | URL | http://127.0.0.1:5000/input |

⚙ Headers, Request params

› Parsing options & Result fields Field types and alias

← → C   ○ 🗎 127.0.0.1:5000/input   ☆   ⊘ ⬇ Ⓢ ⓤ

date,HUFL,HULL,MUFL,MULL,LUFL,LULL,OT 2016-07-01
00:00:00,5.827000141143799,2.009000062942505,1.5989999771118164,0.4620000123977661,4.203000068664552,1.3400000333786009,30.5310001373291 2016-07-01
01:00:00,5.692999839782715,2.075999975204468,1.4919999837875366,0.4259999990463257,4.142000198364259,1.371000051498413,27.78700065612793 2016-07-01
02:00:00,5.1570000648498535,1.741000056266785,1.279000043869018,0.354999989271116394,3.776999950408936,1.21800054359436,27.78700065612793 2016-07-01
03:00:00,5.0900001525878915,1.9420000314712524,1.279000043869018,0.3910000026226044,3.806999921798706,1.279000043869018,25.04400062561035 2016-07-01
04:00:00,5.357999801635742,1.9420000314712524,1.4919999837875366,0.4620000123977661,3.868000030517578,1.279000043869018,21.94799954223643 2016-07-01
05:00:00,5.625999927520752,2.1429998874664307,1.5279999971389768,0.5329999923706055,4.051000118255615,1.371000051498413,21.173999786376953 2016-07-01
06:00:00,7.166999816894531,2.9470000267028813,2.131999969482422,0.7820000052452087,5.026000022888184,1.858000400543213,22.79199981689453 2016-07-01
07:00:00,7.434999942779541,3.282000064849853,2.309999942779541,1.031000018119812,5.086999893188477,2.2239999771118164,23.143999099731445 2016-07-01
08:00:00,5.559000152258789,3.013999938964844,2.45199990272522,1.1729999780654907,2.9549999237060547,1.432000410079956,21.66699981689453 2016-07-01
09:00:00,4.554999828338623,2.545000762939453,1.919000029563904,0.8169999718666077,2.680000667572017,1.371000051498413,17.44599145507812 2016-07-01
10:00:00,4.956999778747559,2.545000762939453,1.9900000095367432,0.8529999852180481,2.9549999237060547,1.4919999837875366,19.979000091552734 2016-07-01
11:00:00,5.760000228881836,2.545000762939453,2.20300006866455,0.8529999852180481,3.441999912261963,1.4919999837875366,20.11899948120117 2016-07-01
12:00:00,4.689000129699707,2.545000762939453,1.812000036239624,0.8529999852180481,2.8329999446868896,1.523000019073486,19.20499992370605 2016-07-01
13:00:00,4.689000129699707,2.678999900817871,1.7769999504089355,1.24399995803833,3.107000112335693,1.613999628067017,18.57200050354004 2016-07-01
14:00:00,5.0900001525878915,2.9470000267028813,2.45199990272522,1.350000023841858,2.559000015258789,1.432000410079956,19.55599975585937 2016-07-01
15:00:00,5.0900001525878915,3.148000019073486,2.486999988555908,1.350000023841858,2.5889999866485,1.523000019073486,17.305000305175778 2016-07-01
16:00:00,4.2199997901916495,2.411000133514404,1.705999704360962,0.7820000052452087,2.61899995803833,1.4919999837875366,19.48600006103516 2016-07-01
17:00:00,4.756000041961167,2.344000101089477,1.6349999904632568,0.7110000252723694,3.075999975204468,1.4919999837875366,19.13400077819824 2016-07-01
18:00:00,5.625999927520752,2.880000114440918,2.523000019073486,1.2079999446868896,3.075999975204468,1.4919999837875366,20.68199920654297 2016-07-01
19:00:00,5.492000102996826,3.013999938964844,2.45199990272522,1.2079999446868896,3.015000104904175,1.5529999732971191,18.71199989318848 2016-07-01
20:00:00,5.357999801635742,3.013999938964844,2.45199990272522,1.2079999446868896,2.86299991607666,1.523000019073486,17.868000305175778 2016-07-01
21:00:00,5.0900001525878915,2.9470000267028813,2.38100004196167,1.2079999446868896,2.680000667572017,1.523000019073486,18.009000778198242 2016-07-01