Contributing to Geoscientific Python Through MetPy and Project Pythia

Ana Castaneda Montoya; Climate and Space Sciences and Engineering, University of Michigan; anacast@umich.edu
Drew Camron, Thomas Martin; University Corporation for Atmospheric Research (UCAR), NSF Unidata Program Center, Boulder, USA

Introduction

NSF Unidata is a community dedicated to the sharing of geoscientific data and providing tools for accessing and visualizing this data for educational and research purposes.

During my internship in the summer of 2024 I had the privilege of contributing to some of NSF Unidata's Python projects. My work included developing educational tutorials and adding new functionalities to existing libraries, enhancing the resources available to the community.

PROJECT PYTHIA

Project Pythia is a Python-centered educational and training resource for the geoscience community.

Project Pythia was born from today's scientist's necessity of "high-level technical skills to effectively analyze, manipulate, and make sense of potentially vast volumes of data". [1]

Project Pythia's series of domain-specific tutorials, called "cookbooks", are created from Jupyter Notebooks and provide example workflows.

Machine Learning Cookbook

This summer I authored a new cookbook that serves as an introduction to running machine learning algorithms and complements NSF Unidata's CyberTraining project.

The cookbook presents a basic regression problem with Linear Regression and Decision Tree Regressor models.

It introduces learner to the scikit-learn API through the following sections:

- Exploratory Data Analysis
- Dataset Splitting
- Dataset Scaling
- Training and Evaluation of Model

See cookbook here:

MetPy

"MetPy is [an open-source] collection of tools in Python for reading, visualizing, and performing calculations with weather data". [2]

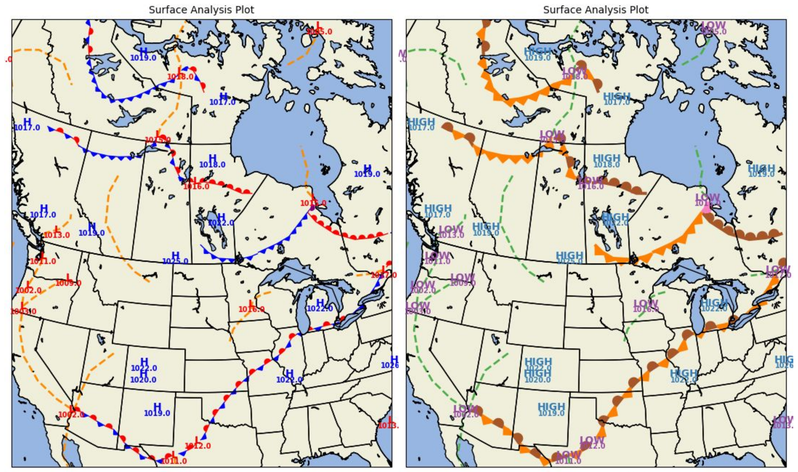

Figure 1. (a) Surface Weather Map plotted using `PlotSurfaceAnalysis()` class with default settings. (b) Same plot with customized colors, labels, marker size and linewidth of fronts.

Plotting WPC Surface Bulletins Made Easy

The Weather Prediction Center's (WPC) surface bulletins are used to plot Surface Weather maps. I wrote and implemented a new class called `PlotSurfaceAnalysis()` (see Fig. 1) that simplifies the process to a few lines of code:

```
df = parse_wpc_surface_bulletin('WPC_file.txt')
```

```
ps = PlotSurfaceAnalysis()
ps.geometry = df['geometry']
ps.feature = df['feature']
ps.strength = df['strength']
```

`PlotSurfaceAnalysis()` is fully customizable. Customizable traits include: colors, labels, label fontsize, markersize, linewidths, linestyles, and label offset.

```
panel = MapPanel()
panel.area = [-120, -80, 30, 70]
panel.projection = 'lcc'
panel.layers = ['lakes', 'land', 'ocean',
                'states', 'coastline', 'borders']
panel.plots = [ps]

pc = PanelContainer()
pc.size = (12,8)
pc.panels = [panel]
pc.show()
```

Complete plotting example:

WPC Surface Bulletin for Southern Hemisphere Issue

- Before plotting a Surface Weather Map, we need to parse the WPC's coded surface bulletin into a Pandas Dataframe.
- MetPy's `parse_wpc_surface_bulletin()` function does exactly that. [3]
- My modifications to this function resolved GitHub issue #3535: WPC Surface Bulletins were being parsed incorrectly for Southern Hemisphere (see Fig. 2).
- After the function was modified, I wrote new tests to ensure its correct performance.

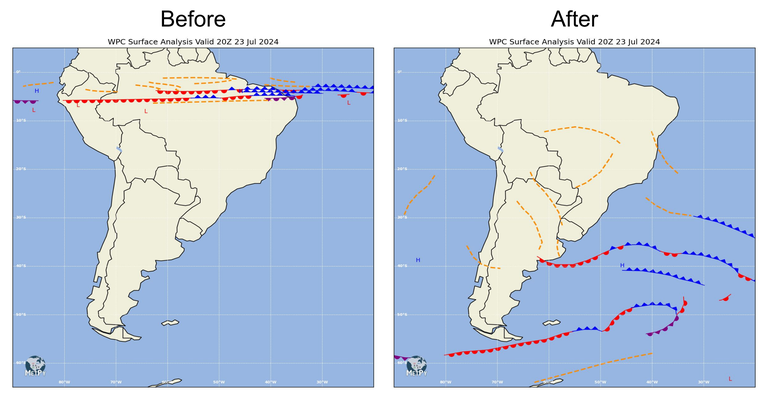

Figure 2. Surface Weather map before and after issue where negative latitudes were being parsed incorrectly was fixed

These new enhancements will be added to MetPy's 1.7.0 milestone release this August/September!

New `train_test_split()` function for xarray

I wrote a new function to create training/testing datasets. Unlike its scikit-learn counterpart, this function:

- Splits the data using the time variable. Specially useful for Climate and Weather data where we deal with time series.
- Creates an optional validation dataset
- Handles xarray datasets

References

[1] https://projectpythia.org/about.html
[2]https://www.unidata.ucar.edu/software/metpy
[3]https://unidata.github.io/MetPy/latest/api/generated/metpy.io.parse_wpc_surface_bulletin.html#metpy.io.parse_wpc_surface_bulletin

This material is based upon work supported by NSF Unidata under award #2403649 from the U. S. National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this material do not necessarily reflect the views of the NSF.