1 Introduction

In this homework you develop a timeseries prediction model for surface temperature data from NOAA. This Climate data is a subset of data from Kaggle. The Kaggle data is from NOAA. Feel free to explore the links above to learn more about the dataset.

1.1 Accessing the data

- Download the starter code and slides from the class website.
- Copy to server using your active directory username and password (replace pgerstoft with your username):

  `scp -r JupyterNotebookHW pgerstoft@dsmlp-login.ucsd.edu:`
- The data used can be downloaded from ECE228NOAA data. Only needed if running locally.

Note: If running locally, you will need to use a Python or Conda environment that has all the packages and Conda installed in order for Basemap (maps) to work. It is possible to get results without Basemap. Please contact TAs or Emma immediately if you need help with doing this.

Extra: How to access BigQuery (not required)
Look at "BigQuery Tutorial.ipynb". You can test your queries on the online console *https://console.cloud.google.com/bigquery*. To do this, you must login to your Gmail/Google account, but it is free.

1.2 Visualizing and loading the data

Run script 0A. Read and understand the code.
Once the data is loaded as dataframe, here is a short description of columns.

`stn`: unique station identifier, string
`slp`: sea level pressure, station-adjusted, float
`wdsp`: mean sustained daily wind speed, float
`mxpsd`: max sustained daily wind speed, float
`max`: maximum daily temperature, float
`min`: minimum daily temperature, float
`prcp`: mean daily precipitation (snow, rain, sleet, etc.), float

2 Problem 1

timeseries_prediction_TEMP.iypnb predicts temperature with random forest for a weather station located in Canada using the Pandas package. It uses lags to create time–dependent features from the observations. The selected station has more than 10 years' consecutive data from Jan 1st, 2008. The first 80% of the data (approximately 8 years) is used to train a random forest. The remaining 20% data (about 2 years) is used to test the performance.

A. Predict the daily temperature for a new station using random forest.
B. Plot the **training** *and* **test** *predictions (with axis labels and legend). Very briefly, discuss the possible downsides of the model and suggest improvements.*

3 Problem 2

timeseries_prediction_Wind.iypnb and *timeseries_prediction-Precip.iypnb* predict wind and temperature with random forest for the same station and procedure as in Problem 1.

A. Predict windspeed, max windspeed, precipitation, or another non–temperature variable using random forest as in Problem 1.
B. Based on your suggestions in Problem 1, try to improve your predictions for random forest. Briefly state what you did. Why do you think it did or didn't improve the results?

4 Problem 3

Try one of the following:

1. Implement an another ML model for timeseries prediction,
2. Use common timeseries preprocessing to improve the ML predictions,
3. Predict the El Nino 3.4 index (play with code in *Optional_HW3*),
4. Pick an equatorial station in the Western Pacific and the Eastern Pacific or Atlantic (opposite sides). Plot and correlate the two timeseries,
5. Pick two geographically close stations. Train an ML model to predict weather at one station based on weather at the other station,
6. Choose your topic **except** timeseries prediction with random forest.

Explain your approach. Include plots with axis labels and proper legend. Write a brief description of what you observe and why machine learning could help answer your question.

5 Submission

`Due date`: **10 May**
`Computer device`: write which hardware was use (Jupyterhub, laptop, google cloud, ...)
`Format`: Write–up in pdf or Jupyter notebook (notebook must be your original writing). Less than 2 pages preferred, with figures. Submit your files with the following naming convention –
`PID_lastname_firstname.pdf` or `PID_lastname_firstname.ipynb`.
`Grading`: 10% of total
`Upload location`: Dropbox link