Weather Forecasting and MOS Correction with Python and Scikit-Learn

Charlie Maalouf - maalouf.charles@gmail.com
Python in the Geosciences, University of Washington

Weather For(now)(back)casting

Many industries depends on weather forecasting (as well as nowcasting and back-casting) for day to day core operations

- **maritime**: directing shipping routes
- **energy**: assessing land resources for asset management
- **roads**: managing salt dispensing operations for icing
- **aviation**: directing and operating flights

Features and NWP

Most of the industries operate on 2 discrete horizons

- **x-days (weeks) ahead**: set energy schedule x-days before generation date
- **real-time**: re-optimize decisions at the time of (or few hours before) realization

For example, ISO’s schedule energe 1 to 3 days ahead of generation time, then deals with imbalances at real time.

Both of these problems rely on the availability of predictive features: **observations** and **nwp forecasts**

NWP Model Output

The center of any forecasting system is a collection of **NWP model output** and **observations**

NWP output tend to have structural errors that may be problematic to some industries

Selling wind energy in advance that was forecasted erroneously may be very costly

Global models: tend to have biases and lack local features of weather systems

Local models: tend to capture local features but at the expense of stronger biases

MOS Correction

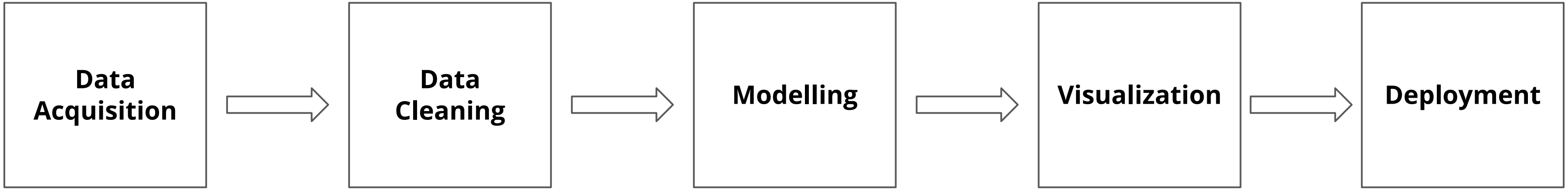
Use output of NWP model as an input to a statistical model to improve forecasting accuracy

A cheaper way to up/downscale model and to project onto new coordinates as opposed to WRF

Try to model observations as a function of model output and other variables:

- Correcting for timing errors require often to compute moving averages of NWP output
- Adding measured observations help often in short horizons
- Including time information may help capturing seasonal and other local recurrent weather features
- Deriving features from model output such as pressure gradients

Weather With python™



unidata

GRIB-API

pandas

Dato

xray

Spark

scikit learn

Spark

Dato

matplotlib

Cartopy

Seaborn

django

Flask
web development,
one drop at a time

The OSS Python stack (… that I’ve used in weather)

Siphon: collection of Python utilities for downloading data from Unidata data technologies (THREDDS)

Pandas: data structures and data analysis tools for the Python programming language

SFrame: Scalable tabular (**SFrame**, **SArray**) and graph (**SGraph**) data-structures built for out-of-core data analysis

Xray: N-dimensional variants of the core pandas data structures

Scikit-Learn: Simple and efficient tools for data mining and data analysis in Python

Demo!

Why Python?

Almost anything you need is available somewhere

- Stop rewriting code
- Whatever is missing → contribute
- Don’t be afraid of giving up legacy code

Great Community: when in doubt, ask!

Flexible language: data acquisition - data cleaning - machine learning - visualization - deployment

Very performant***************

Questions!

Charlie Maalouf - maalouf.charles@gmail.com