Travel Recommenders

Anish Dixit
Tanushree Yandra
Ananya Nair
Mithali Shashidhar

Background & Use Case

- Travel Recommenders is a tool that combines weather data with users interests to recommend places to visit
- **Our Use Case:**
  - We want to design a solution that can recommend places or activities taking the effect of weather into account. For instance, recommending a park on a sunny day v/s an indoor cozy cafe on a rainy day.
- **What about it needs a Python library?**
  - Our recommender needs a python library to access to dynamic weather data given a particular location.
  - We would also require to decode a user entered location or the users current location into coordinates.

Python Package Choices

1. **Meteostat Python Library:**

   The Meteostat Python library provides a simple API for accessing open weather and climate data. The historical observations and statistics are collected by Meteostat from different public interfaces, most of which are governmental.

   The developers, Meteostat, are an agency that maintains an open source record of global climate data. Among the data sources are national weather services like the National Oceanic and Atmospheric Administration (NOAA) and Germany's national meteorological service (DWD).

2. **Python-weather Library:**

   A free and asynchronous weather API wrapper made in python, for python.

   Made by a private contributor for use by other python developers in weather data associated projects.

Comparison Table

Features	**Meteostat**	**Python-weather**
Ease of installation	pip install meteostat	pip install python-weather
Addressing requirements of our project	Returns a rich response of parameters like temp data, wind data etc grouped daily, hourly etc, given a specific location. Gives historical data.	Returns limited parameters such as current temperature etc only. No provision for manually entering dates or historical data.
Release date and usage	Released in 2020. 222 stars on github, popular among users.	Released in 2021. 62 stars on github, not very widely used.
Availability of Examples	https://dev.meteostat.net/python/. Very detailed explanation of classes, data types and examples of use cases.	Not very user friendly. Debugging tough due to lack of examples.
Computational Efficiency	Works well for data of multiple years & parameters at once.	Restricted to single location, less features.
Software Bugs	7 open issues on github. Certain destinations not serviced, android SSL errors.	4 open issues on github about system glitches. Local trials gave JSON errors which could not be resolved.

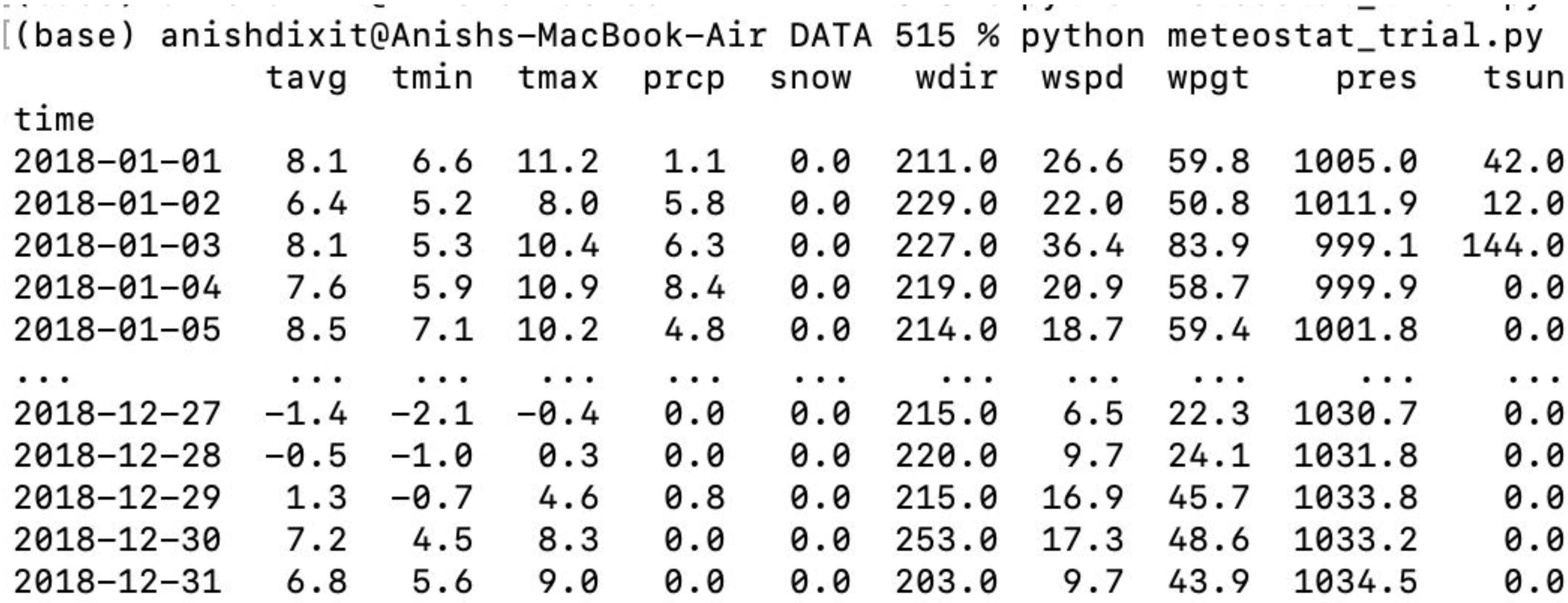
Our Choice

- What did we choose? **Meteostat Python Library**
- Why did we choose to use it?
  - Ease of use: Far outweighs python-weather in terms of writing code to perform an operation.
  - Rich response with parameters relevant to our use case. Historical data also available.
  - Availability of relevant examples and debugging is relatively easy.
  - Comprehensive documentation: Dedicated website to explain classes and data types.
  - Exposes a bulk data interface that provides access to full data dumps of individual weather stations in a csv format
  - Returns data in a compatible Python List and Array format for our service to easily ingest and combine with our other data elements.
  - The existing issues of some locations not being serviced is not applicable to us as we are limiting out study to Washington state. Android SSL issues are also irrelevant as we will not be using Android.

Drawbacks/Remaining Concerns

- **Potential Drawbacks:**
  - Meteostat collects data from several public sources, creating an upstream dependency. For areas with no weather stations, there would be no data.
  - The working of this package is dependent on stable internet access.
- **Will we have to mitigate any drawbacks?**
  - Primary exploration does not reveal any pressing issues, but will get a better idea as we use it more.
- **What will we watch out for?**
  - Granularity and consistency of data being produced.
  - Potential edge cases where data is not returned/returned inaccurate data/ timeouts.
  - Limit on number of API calls.

Package Demonstration



```
[(base) anishdixit@Anishs-MacBook-Air DATA 515 % python meteostat_trial.py
              tavg  tmin  tmax  prcp  snow   wdir  wspd  wpgt    pres   tsun
time
2018-01-01     8.1   6.6  11.2   1.1   0.0  211.0  26.6  59.8  1005.0   42.0
2018-01-02     6.4   5.2   8.0   5.8   0.0  229.0  22.0  50.8  1011.9   12.0
2018-01-03     8.1   5.3  10.4   6.3   0.0  227.0  36.4  83.9   999.1  144.0
2018-01-04     7.6   5.9  10.9   8.4   0.0  219.0  20.9  58.7   999.9    0.0
2018-01-05     8.5   7.1  10.2   4.8   0.0  214.0  18.7  59.4  1001.8    0.0
...            ...   ...   ...   ...   ...    ...   ...   ...     ...    ...
2018-12-27    -1.4  -2.1  -0.4   0.0   0.0  215.0   6.5  22.3  1030.7    0.0
2018-12-28    -0.5  -1.0   0.3   0.0   0.0  220.0   9.7  24.1  1031.8    0.0
2018-12-29     1.3  -0.7   4.6   0.8   0.0  215.0  16.9  45.7  1033.8    0.0
2018-12-30     7.2   4.5   8.3   0.0   0.0  253.0  17.3  48.6  1033.2    0.0
2018-12-31     6.8   5.6   9.0   0.0   0.0  203.0   9.7  43.9  1034.5    0.0
```