MACHINE LEARNING BASED WEATHER PREDICTION

Robin Shindelman CS

CS 332 – Applied Data Science | shindelr@oregonstate.edu

Table of Contents

Introduction *2*

Why weather? **2**

Research questions **3**

Gathering Data *4*

Direct Data Download **4**

Atmospheric Data from ERA5 4
Avalanche Data for CO 4
General Marine Data 5
Urban Air Quality Data 5
Curated Rain Occurrence Data 5

Data Gathering Using an API **6**

Atmospheric Data for Colorado, Front Range 6

Data Cleaning *7*

ERA5 Meteorological Data **7**

Before Cleaning 7
After Cleaning 9

CO Avalanche Data **11**

Before Cleaning 11
After Cleaning 12

Urban AQI Data **15**

Before Cleaning 15
After Cleaning 17

Unsupervised Learning with k-means Clustering *18*

Formatting the data **18**

Visualization in 3D **20**

Applying k-means: **21**

k-means Conclusions **22**

Supervised Learning with Decision Trees *23*

Data Formatting **23**

Label Engineering **24**

Test-Train Split **24**

Training Visualization 24

Applying Decision Trees **26**

Confusion Matrix 26
Tree Visualization 27

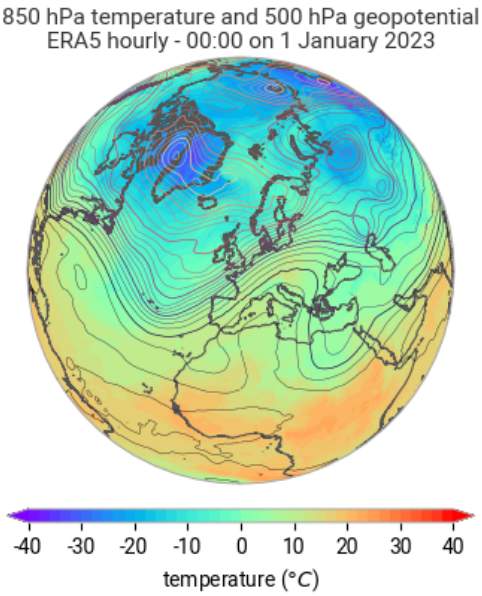
Conclusion **27**

References *29*

Introduction

Why weather?

The weather, on any given day, is important for almost every industry and activity a person can think of. From simple recreational activities like skiing all the way to the hyper complex organization of global shipping lanes, weather plays an integral role. Weather can reach us even in places where it physically cannot travel. Wind speed and solar predictions affect prices on the stock market based on renewable energy prediction. Even infrastructure as wide and ubiquitous as a state's electricity grid can be brought to a grinding halt by the power of heat domes or hurricanes.



1. Beautiful chart depicting temperature, geopotential, and isobars over the Arctic, Europe, and North Africa.

Precisely because of the wide reaching impacts of weather, large quantities of accessible data from many sources exist for use in modeling and research. This huge amount of information makes the topic ideal for an aspiring data scientist. Many large models from around the globe are in production, with many more under development. These studies can be used as inspiration and guides for those of us beginning to dip our toes in the field.

The impetus for this project comes from multiple places. As a lifelong surfer, skier, and general outdoor enthusiast, the weather has always played an important role in my favorite activities. I've been poring over marine forecasts to predict swell at my favorite beaches since I was 12. Sometime later, when I found myself exploring alpinism and backcountry skiing, I added avalanche reports and mountain weather to the list of charts that needed understanding. Now, after nearly two decades of this self-forecasting, I've found myself a part of an OSU CEOAS lab where the understanding of how swell might propagate to our field site in the Caribbean is of great importance for data collection.

The true prediction of weather is beyond the scope of this course and probably an undergraduate degree in general. However, this seems to be a perfect opportunity to explore atmospheric data in a structured way. A primary goal for this project is to gain some understanding of how processes in the atmosphere are measured, as well as how non-weather specific events can be related to the weather in a way that is apparent through the data. The latter of these goals is most obviously illustrated my work with the urban air quality dataset at the end of this report.

Research questions

Below are some initial research questions to guide this study.

1. How can data about pressure and wind speed be used to predict swell height and direction at specific beaches?
    a. Data on pressure, windspeed, and Lat/Long can be used from the ERA5 hourly dataset for atmospheric information. Buoy data from NDBC can be used for dominant swell height and direction labels at specific locations.
2. How can data about precipitation and wind direction be used to predict avalanche risk in specific locations?
    a. Data on precipitation and wind direction from the ERA5 dataset can be used for atmospheric information, while labels for avalanche size and severity can be sourced from the CAIC accident repository (pretty sad dataset of avalanche accidents reported to the CAIC).
3. Can patterns in meteorological data be related to avalanche frequency?
    a. Perhaps by combining ERA5 and CAIC data, I can try to cluster the data to identify patterns in the all the variables.
4. Can patterns in meteorological data be related to power outages?
    a. Similar to question 3, but I can use the EIA hourly electric grid monitoring dataset to correlate weather patterns to severe drops in subregion demand that could be related to power outages.
5. What weather patterns (pressure, wind speed, temperature, etc.) in the Atlantic tend to produce swell heights of 3ft or more near Puerto Rico?
    a. Weather data from ERA5 can be used to predict for continuous swell height labels of greater than 3ft at the NDBC buoy near Puerto Rico.
6. Have northerly wind speeds along the Oregon Coast increased over time?
    a. This question can be accomplished by comparing windspeed trends overtime either from the NDBC databases for offshore windspeeds, or from the ERA5 database using a combination of their u- and v-component variables.
7. Are there recognizable patterns relating humidity and atmospheric pressure to rainfall?
    a. Using the ERA5 dataset, we can relate the relative humidity, fraction of cloud cover, and geopotential measurements to predict on specific rain water content.
8. What windspeeds, wind directions, and atmospheric pressure measurements are associated with high measurements of vertical velocity (low pressure storm systems) in Oregon?
    a. Using the ERA5 dataset I can use the u- and v-wind component variables to find wind direction and velocity, geopotential for pressure, and relate them to vertical velocity and divergence.
9. Looking at historical flood events, what meteorological patterns are most associated with flooding?
    a. I can explore the ERA5 dataset, comparing spikes in specific rain water content to historic flood events.
10. Looking at historical events of massive waves (greater than 40ft) at Mavericks, CA, what was the weather like in the Gulf of Alaska? Can specific weather patterns be attributed to large waves at specific spots?
    a. This is a very exciting question which I've always wondered about. I'd like to define a region of Lat/Long coordinates to use from the ERA5 dataset and try to correlate

storm events there with buoy readings in known big wave spots along the West Coast of North America.

Gathering Data

Direct Data Download

Atmospheric Data from ERA5

https://object-store.os-api.cci2.ecmwf.int/cci2-prod-cache/43b6f072cb65230a2e1c2354af9a06c7.grib

The ERA5 dataset from the European Center for Medium Range Weather Forecasts (ECMWF) is an incredible trove of information on global meteorological data. The link above will bring the user to a download for hourly pressure levels of 800, 900, and 1000 for an area spanning the NE coast of Asia in the winter of 2024. Variables include latitude, longitude, vertical velocity, temperature, u- and v- wind components, and more. I'm hoping to use this dataset, and derivations of it, as the backbone for weather analyses in this project.

				number	step	valid_time	d	z	crwc	t	u	v	w	vo
time	isobaricInhPa	latitude	longitude											
2024-01-01	1000.0	70.0	117.00	0	0 days	2024-01-01	-0.000043	780.67627	0.0	239.068787	0.486252	-0.047791	0.006964	0.000040
			117.25	0	0 days	2024-01-01	-0.000049	774.92627	0.0	239.455505	0.174728	0.265686	-0.004023	0.000047
			117.50	0	0 days	2024-01-01	-0.000045	771.05127	0.0	239.664490	-0.154373	0.594788	-0.002802	0.000045
			117.75	0	0 days	2024-01-01	-0.000034	769.67627	0.0	239.719177	-0.427811	0.873108	0.010870	0.000035
			118.00	0	0 days	2024-01-01	-0.000019	771.05127	0.0	239.635193	-0.601639	1.056702	0.028448	0.000020

Avalanche Data for CO

https://avalanche.state.co.us/sites/default/files/2023-11/CAIC_Accident_Data_2023.xlsx

The Colorado Avalanche Institute keeps a running dataset of avalanche accidents reported to them. The dataset includes columns describing Lat/Long, D Size, month, year, day, and more. This is a simple Excel spreadsheet. The data here is labeled, where the size of the avalanche is the predictable item. The dataset primarily qualitative with just a few quantitative variables.

NOTE: This dataset includes information about avalanche fatalities and injury, and so may be sensitive to some readers. I do not plan to use those columns for my analyses.

	AvyYear	YYYY	MM	DD	Location	Trigger	D Size	Setting	State	lat	lon	PrimaryActivity	TravelMod	Killed	Description	Date
2	2023	2023	7	2	Split Mountain, southwest of Big Pine			BC	CA	37.021	-118.422	Hiker	Foot	1	3 hikers caught, 2 injured, 1 killed	7/2/2023
3	2023	2023	6	14	Hurd Peak, southwest of Bishop	N	1.5	BC	CA	37.143	-118.566	Backcountry Tourer	Ski	1	2 backcountry tourers caught, 1 killed	6/14/2023
4	2023	2023	5	5	West Ridge of the Moose's Tooth			BC	AK	62.969	-150.613	Climber	Foot	2	2 climbers caught, killed	5/5/2023
5	2023	2023	5	4	Denali National Park, near Jenny Creek			BC	AK	63.720	-148.984	Backcountry Tourer	Ski	1	1 backcountry skier caught and killed	5/4/2023
6	2023	2023	4	29	Bald Mountain, southeast of Breckenridge	AS	2	BC	CO	39.446	-105.960	Backcountry Tourer	Ski	1	1 backcountry tourer caught, buried, and killed	4/29/2023
7	2023	2023	4	17	Big Cottonwood Creek, east of Sandy	U		TN	UT	40.604	-111.583	Resident	Foot	1	1 resident caught, partially buried-critical, and killed	4/17/2023
8	2023	2023	3	27	Pole Canyon, Oquirrh Moutains	AM	4	BC	UT	40.369	-112.192	Snowmobiler	Snowmobi	1	1 snowmobiler caught, buried, and killed	3/27/2023
9	2023	2023	3	22	Trinity Lakes, east of Boise			BC	ID	43.624	-115.429	Snowmobiler	Snowmobi	1	1 snowmobiler caught, buried, and killed	3/22/2023
10	2023	2023	3	19	Maroon Bowl, north of Highland Peak	AS	3	BC	CO	39.135	-106.877	Sidecountry Rider	Ski	1	1 sidecountry rider caught, partially buried-critical, and killed	3/19/2023
11	2023	2023	3	17	Rapid Creek, southwest of Marble	AS	3	BC	CO	39.048	-107.258	Backcountry Tourer	Ski	1	3 backcountry tourers caught, 2 injured, 1 buried and killed	3/17/2023
12	2023	2023	3	16	Purgatory Creek, north of Durango	N		TN	CO	37.629	-107.814	Resident	Foot	1	3 residents caught, buried, and 1 killed	3/16/2023
13	2023	2023	3	15	Paulina Peak, south of Bend	AR	2	BC	OR	43.692	-121.255	Backcountry Tourer	Snowboarc	1	1 backcountry snowboarder killed	3/15/2023
14	2023	2023	3	12	Red Mountain, northeast of Lowman			BC	ID	44.254	-115.406	Snowmobiler	Snowmobi	1	1 snowmobiler caught and killed	3/12/2023
15	2023	2023	3	12	Observation Peak, Stanley Lake Creek Drainage	AM	2	BC	ID	44.172	-115.106	Snowmobiler	Snowmobi	1	1 snowmobiler caught, buried, and killed	3/12/2023
16	2023	2023	3	9	Upper Weber Canyon, southwest of Windy Ridge	AS		BC	UT	40.838	-111.021	Mechanized Guided Client	Ski	1	2 skiers caught and fully buried, 1 injured and 1 killed	3/9/2023
17	2023	2023	3	2	Black Crater, west of Sisters	AS	2	BC	OR	44.268	-121.738	Backcountry Tourer	Ski	1	1 backcountry skier caught, partially buried-critical, and killed	3/2/2023
18	2023	2023	2	25	South of Vallecito Reservoir	AS	2	BC	CO	37.369	-107.572	Backcountry Tourer	Ski	2	2 backcountry skiers caught, buried, and killed	2/25/2023

General Marine Data

https://www.ndbc.noaa.gov/data/historical/stdmet/46050h1995.txt.gz

The National Data Buoy Center holds a massive repository of maritime information. The data you can collect from NDBC is dependent on the type of station you choose to query via their API. One can find spectral swell information, meteorological information, and tidal data. An example dataset from the Stonewall Bank buoy off Newport, OR in 1995 is shown below. This dataset can be read in as a text file or CSV and is purely quantitative.

46050h1995.txt

YY	MM	DD	hh	WD	WSPD	GST	WVHT	DPD	APD	MWD	BAR	ATMP	WTMP	DEWP	VIS
95	01	01	00	999	99.0	99.0	02.70	11.10	06.80	234	1016.0	08.4	999.0	999.0	23.4
95	01	01	01	999	99.0	99.0	02.50	10.00	06.50	216	1015.8	08.3	999.0	999.0	21.6
95	01	01	02	999	99.0	99.0	02.50	10.00	06.50	209	1015.5	08.6	999.0	999.0	20.9
95	01	01	03	999	99.0	99.0	02.20	10.00	06.10	203	1015.0	08.6	999.0	999.0	20.3
95	01	01	04	999	99.0	99.0	02.40	12.50	06.00	263	1015.1	08.3	999.0	999.0	26.3
95	01	01	05	999	99.0	99.0	02.10	14.30	05.90	279	1014.7	07.9	999.0	999.0	27.9
95	01	01	06	999	99.0	99.0	02.30	14.30	06.00	266	1014.2	07.9	999.0	999.0	26.6
95	01	01	07	999	99.0	99.0	02.60	10.00	06.50	214	1014.2	07.9	999.0	999.0	21.4
95	01	01	08	999	99.0	99.0	02.50	11.10	06.40	226	1014.3	08.3	999.0	999.0	22.6
95	01	01	09	999	99.0	99.0	02.40	12.50	06.40	257	1014.3	08.3	999.0	999.0	25.7

Urban Air Quality Data

https://www.kaggle.com/datasets/abdullah0a/urban-air-quality-and-health-impact-dataset

This dataset focused on urban air quality provides real measurements of 10 different cities' AQI for the month of September in 2024. Columns span the gamut from meteorological data such as windspeed, precipitation, and temperature; to human dimensions such as the day of the week, health risk score, and whether it's the weekend or not.

Curated Rain Occurrence Data

https://www.kaggle.com/datasets/zeeshier/weather-forecast-dataset

This dataset is focused on predicting whether or not rain occurred on a given day, based on a variety of factors such as cloud cover, pressure, or humidity. The majority of the data is quantitative with just a single qualitative column of "rain" or "no rain" which is to be used as labels for ML.

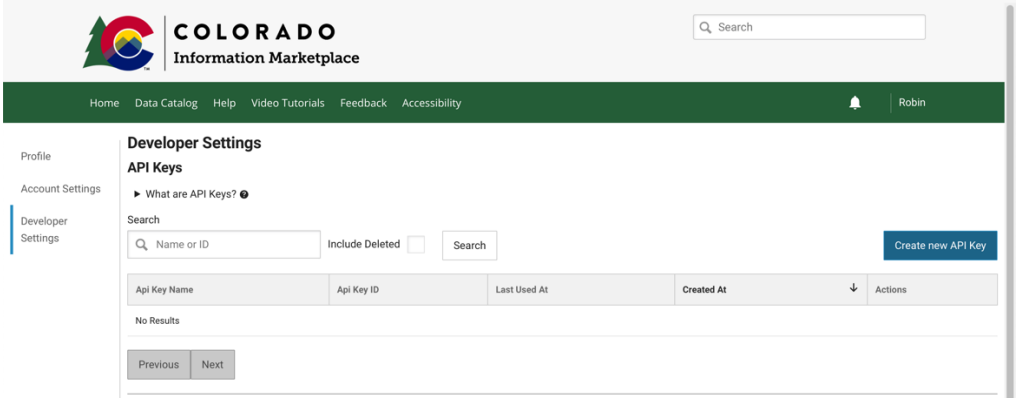
```
Temperature,Humidity,Wind_Speed,Cloud_Cover,Pressure,Rain
23.720337598183118,89.59264065174611,7.335604391040214,50.501693832913155,1032.378758690279,rain
27.879734159310487,46.48970403534824,5.952483593282764,4.990052927536981,992.6141895121403,no rain
25.069084401791095,83.07284289257146,1.3719918180799207,14.855783939243427,1007.2316201172738,no rain
23.622079574922424,74.36775792086564,7.050550632784658,67.25528206034686,982.6320127095369,rain
20.591369983472617,96.85882241307947,4.6439209259534975,47.67644427890656,980.8251417426507,no rain
26.147352826666403,48.21726041042326,15.258546760433369,59.76627863227763,1049.738751014439,no rain
20.939680281567313,40.79944443606025,2.2325659971252265,45.827508421021676,1014.1737660638806,no rain
32.29432501955199,51.848471069719,2.873620839084101,92.5514972525134,1006.0417334780344,no rain
34.09156901252573,48.05711447385431,5.570206017562278,82.52487276907972,993.7320466194785,no rain
19.58603797064444,82.9782932331459,5.7605365867061025,98.0144498972855,1036.5034571561982,rain
29.793125952066614,81.31765126927587,16.92609873392226,93.92329417135517,1029.4026903671104,no rain
23.22237299382261,76.87794305987336,15.825673129009745,72.86978986682412,980.1089338550743,rain
24.201114027348307,45.14653801209415,11.572712664104507,5.25304210213815,1033.9858671128165,no rain
```

Data Gathering Using an API

Atmospheric Data for Colorado, Front Range

This dataset is real time quantitative meteorological data from a weather station in Boulder, CO. I plan to use this dataset to provide more information for the avalanche data from the Colorado Avalanche Information Center.
Socrata API documentation: https://dev.socrata.com/docs
The steps I took to acquire this data are as follows:

1. Sign up for an account and generate an API token.

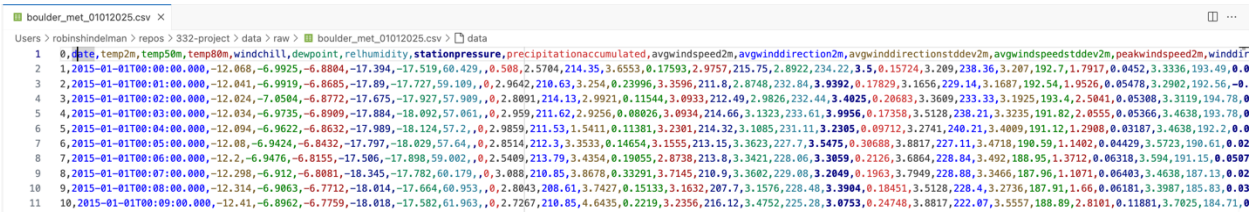


key: 9CRMr027a7dUWzFtpnFTukjLd

2. Using Socrata's API/SDK documentation and examples, I wrote a quick function to communicate with the CO Information Marketplace API.

```python
def boulder_weather_station(domain: str, token: str) -> pd.DataFrame:
    """
    API Documentation:
        https://dev.socrata.com/foundry/data.colorado.gov/pfjr-vhp3

    SoQL example: where=date > 2014-12-31T00:00:00.00
    """
    client = Socrata(domain, token)
    results = client.get("pfjr-vhp3",
                         content_type='csv',
                         limit=10000,
                         where="date >= '2015-01-01T00:00:00' and date < '2015-01-02T00:00:00'")
    return pd.DataFrame.from_records(results)
```

3. Socrata's API uses their own query language named SoQL. Using their library, users can pass SQL-like commands to the HTTP GET request as query parameters. Using this filtering method, I requested all the available atmospheric weather measurements at the Boulder station on January 1st, 2015. A small portion of the resulting csv can be seen below:



Example Query: https://data.colorado.gov/pfjr-vhp3?$$app_token=9CRMr027a7dUWzFtpnFTukjLd&content_type=csv&limit=100&$where=date>='2015-01-01T00:00:00'%20and%20date<'2015-01-02T00:00:00'

Please note that this typed out version will probably not yield an actual result since I had to use Socrata's proprietary API to get the correct data.

Data Cleaning

ERA5 Meteorological Data

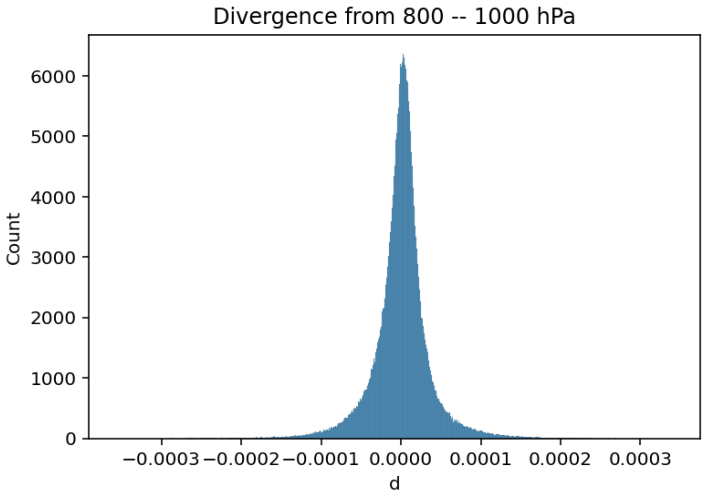
Before Cleaning

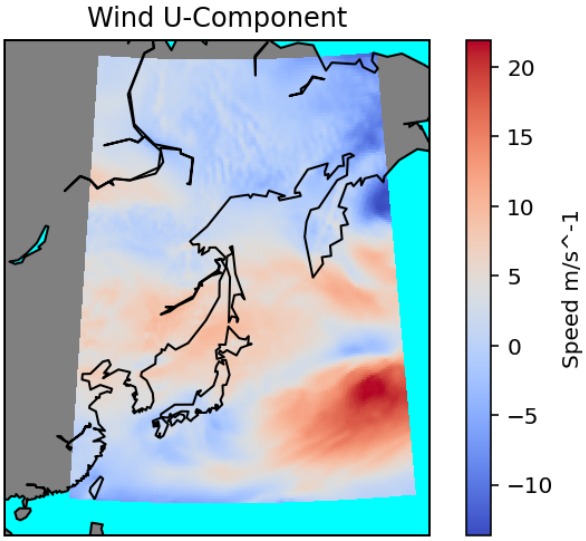
Example Data

```
In [22]: df.head()
Out[22]:
         time  isobaricInhPa  latitude  ...         v         w         vo
0  2024-01-01         1000.0      70.0  ... -0.047791  0.006964  0.000040
1  2024-01-01         1000.0      70.0  ...  0.265686 -0.004023  0.000047
2  2024-01-01         1000.0      70.0  ...  0.594788 -0.002802  0.000045
3  2024-01-01         1000.0      70.0  ...  0.873108  0.010870  0.000035
4  2024-01-01         1000.0      70.0  ...  1.056702  0.028448  0.000020

[5 rows x 15 columns]
```

Visualizations of variables

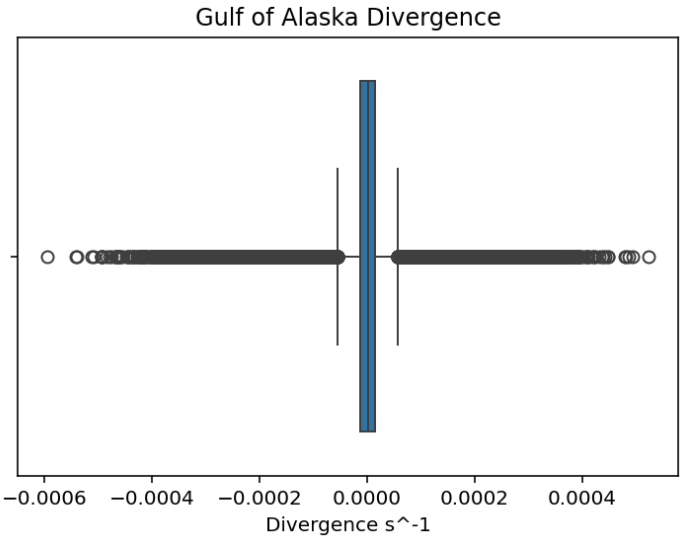
The majority of the data in this data set is incredibly normal in shape. For instance, the measurements of divergence, which is the horizontal flow of air from a single point, shows little to no skewing and no outliers. This indicates that the majority of air mass over East Asia was extremely stable on January 1st, 2024.





This map visualizes the U-component, horizontal velocity, of the windspeed for the area of interest. I heavily leaned on the materials in the matplotlib basemap documentation to create this chart.

A boxplot of divergence data over the Gulf of Alaska and US West coast appears to show a possibly significant outlier on the left side. This variable also shows some left skewing and quite a few data points falling outside of the interquartile range.



After Cleaning

The ERA5 data began in .GRIB formatting, but was wrangled into a Pandas dataframe and eventually a clean .csv for future analysis.

A few of the data types were loaded into the dataframe incorrectly. The "time" variable was converted from an object to a datetime, and "isobaricInhPa" was converted from a float to an integer. Additionally, some unnecessary columns were removed: "step", "number", and "valid_time" all provide unneeded metadata for the analysis.

```
time              object
isobaricInhPa    float64
latitude         float64
longitude        float64
number             int64
step              object
valid_time        object
d                float64
q                float64
crwc             float64
cswc             float64
t                float64
u                float64
v                float64
dtype: object
```

```
time             datetime64[ns]
isobaricInhPa             int64
latitude                float64
longitude               float64
d                       float64
q                       float64
crwc                    float64
cswc                    float64
t                       float64
u                       float64
v                       float64
dtype: object
```

Out of the 966,875 rows in the data set, there were 0 missing values. Due to this, no operations were conducted on the dataset to correct missing values.

```
RangeIndex: 966875 entries, 0 to 966874
Data columns (total 11 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   time           966875 non-null  datetime64[ns]
 1   isobaricInhPa  966875 non-null  int64
 2   latitude       966875 non-null  float64
 3   longitude      966875 non-null  float64
 4   d              966875 non-null  float64
 5   q              966875 non-null  float64
 6   crwc           966875 non-null  float64
 7   cswc           966875 non-null  float64
 8   t              966875 non-null  float64
 9   u              966875 non-null  float64
 10  v              966875 non-null  float64
dtypes: datetime64[ns](1), float64(9), int64(1)
memory usage: 81.1 MB
```

The variable, "t", which denotes temperature, was being reported in Kelvin. This metric was particularly difficult to interpret, so it was converted to Fahrenheit using the formula:

$$F = (K - 273.15) * 1.8 + 32$$

This resulted in the following statistics for the temperature variable:

```
count    966875.000000
mean        271.912019
std          12.698879
min         230.954940
25%         265.058135
50%         273.336700
75%         281.240420
max         300.213650
Name: t, dtype: float64
```

```
count    966875.000000
mean         29.771635
std          22.857981
min         -43.951108
25%          17.434643
50%          32.336060
75%          46.562756
max          80.714570
Name: t, dtype: float64
```

To remove the outliers discovered in the Gulf of Alaska divergence variable, a routine Pandas cleaning procedure was applied. A Boolean mask filtered out all rows showing a divergence below -0.0005 s^{-1} or above 0.0005 s^{-1}. This removed ~60 rows.

```
count    9.668750e+05
mean     5.085720e-08
std      3.767190e-05
min     -5.939205e-04
25%     -1.279777e-05
50%      1.756241e-06
75%      1.498265e-05
max      5.227116e-04
Name: d, dtype: float64
```



```
                  d
count  9.668690e+05
mean   5.309996e-08
std    3.764831e-05
min   -4.940219e-04
25%   -1.279777e-05
50%    1.756241e-06
75%    1.498265e-05
max    4.939251e-04
```

To finish the cleaning process, this data set was reduced further into purely quantitative variables by removing the timestamp, longitude, and latitude variables from the data frame. The pressure variable was also removed. This smaller data frame was then normalized via the Min-Max method resulting in the following data frame:

	d	q	crwc	cswc	t	u	v
0	2.372311e-06	0.003365	0.0	0.000045	32.680580	-1.099014	2.905258
1	-5.781185e-07	0.003388	0.0	0.000049	32.476694	-0.929092	2.958969
2	-5.763723e-06	0.003402	0.0	0.000053	32.395820	-0.861710	3.075180
3	-1.556869e-05	0.003423	0.0	0.000057	32.395820	-0.954483	3.277328
4	-2.781744e-05	0.003461	0.0	0.000063	32.381780	-1.282608	3.582993

	d	q	crwc	cswc	t	u	v
0	0.534010	0.209805	0.0	0.013216	0.614698	0.444417	0.517974
1	0.531368	0.211202	0.0	0.014173	0.613062	0.446380	0.518559
2	0.526724	0.212124	0.0	0.015410	0.612413	0.447158	0.519826
3	0.517943	0.213402	0.0	0.016768	0.612413	0.446087	0.522029
4	0.506974	0.215751	0.0	0.018509	0.612301	0.442297	0.525361

CO Avalanche Data

Before Cleaning

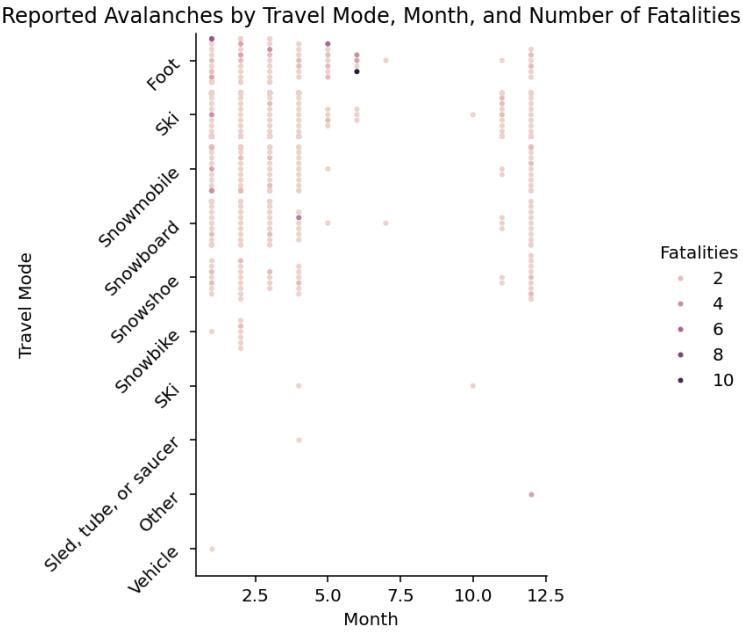
Example Data

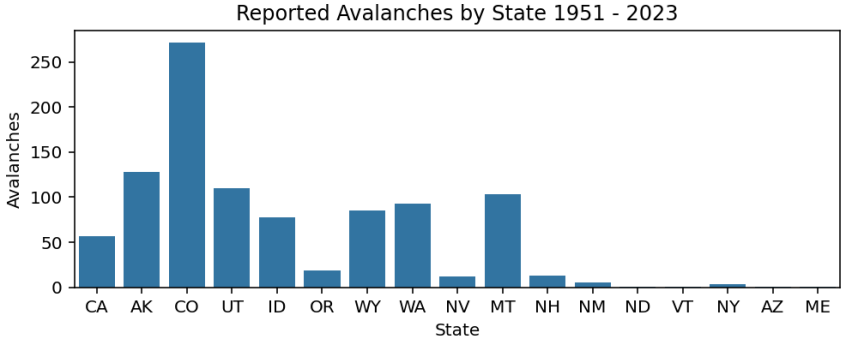
	AvyYear	YYYY	MM	...	lon	PrimaryActivity	TravelMode
0	2023	2023	7	...	-118.422300	Hiker	Foot
1	2023	2023	6	...	-118.566151	Backcountry Tourer	Ski
2	2023	2023	5	...	-150.613333	Climber	Foot
3	2023	2023	5	...	-148.984013	Backcountry Tourer	Ski
4	2023	2023	4	...	-105.959700	Backcountry Tourer	Ski
..	...	...	..	...	...	...	...
976	1952	1952	1	...	0.000000	Inbounds Rider	Ski
977	1952	1952	1	...	0.000000	Resident	Foot
978	1952	1952	1	...	0.000000	Miner	NaN
979	1952	1951	12	...	0.000000	Motorist	NaN
980	1951	1951	4	...	0.000000	Motorist	NaN

Visualizations of variables

The raw Colorado Avalanche data set has 981 total rows. As the beeswarm plot to the right shows, there are some possible outliers in the "Fatalities" column. One incident shows 10 reported fatalities where all people involved were traveling by foot. This extremely large number warrants investigation.

There also appears to have been fatal avalanches as late in the year as July or June. These avalanches seem to have occurred during a time of year when very little snow would be present and are possible outliers as well.

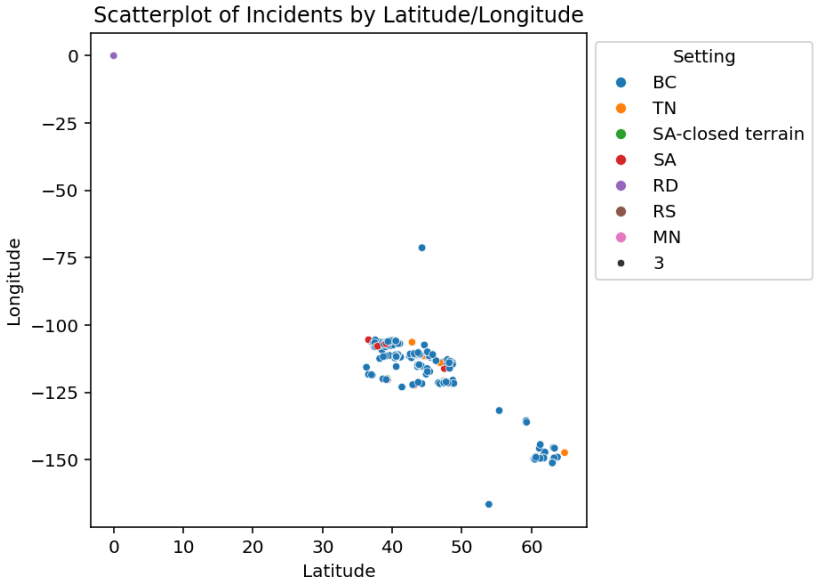




The majority of avalanches reported in this dataset have taken place in states with the most prominent mountains. However, it is possible that Colorado has gained a disproportionate number of reports since this dataset was sourced from the Colorado Avalanche Institute.

Exploration of the locations and settings of the data revealed some suspicious points. At least one of the avalanches reported appears to have occurred at 0, 0 latitude and longitude. This is most likely an error as all the states in the continental US are between 30 and 50 degrees North, while Alaska is between 50 and 80 degrees North.

There is also at least one setting reported as "3". This is an error as well, as the "Setting" variable should be acronyms for the type of recreation being done.



After Cleaning

The CO Avalanche Institute data was gathered as an Excel spreadsheet. To begin EDA and the cleaning process, this spreadsheet was read into a Pandas dataframe. After being fully cleaned, the dataset was output into a .csv.

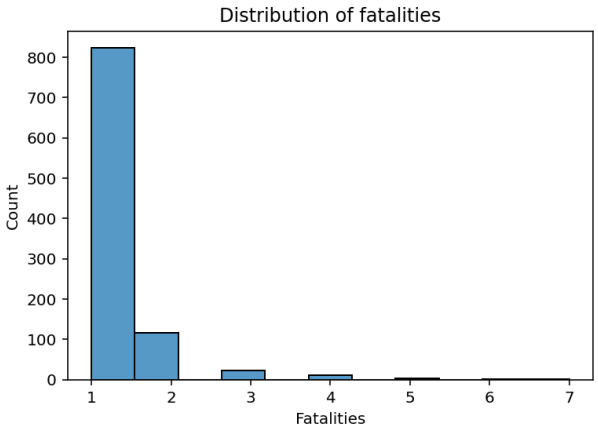
Not all of the data was read in to the dataframe correctly. "Setting", "State", "PrimaryActivity", "TravelMode", and "Trigger" were all converted to categories.

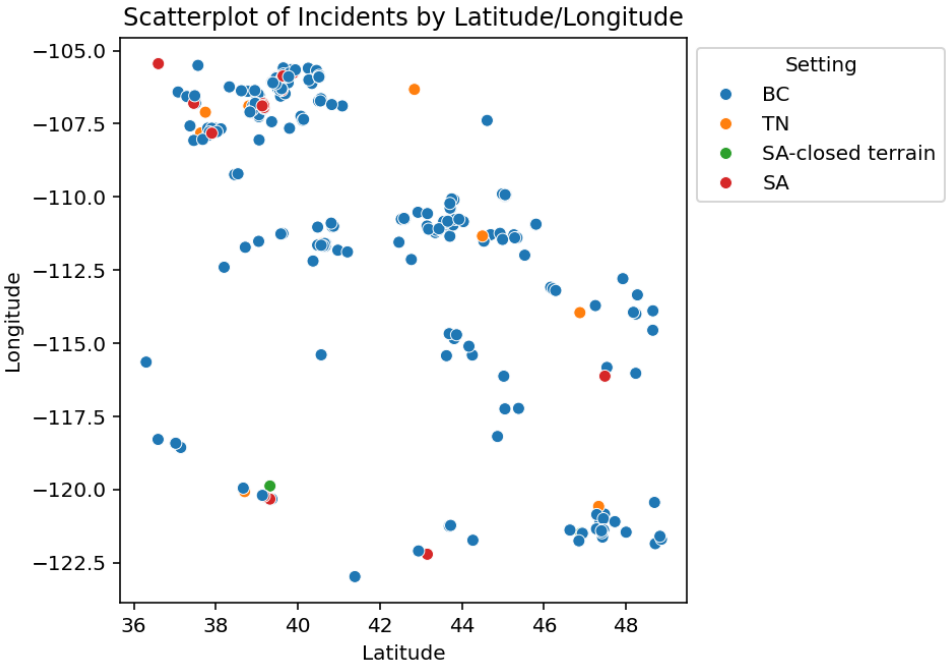
```
AvyYear                 int64
YYYY                    int64
MM                      int64
DD                      int64
Location               object
Trigger                object
D Size                float64
Setting                object
State                  object
lat                   float64
lon                   float64
PrimaryActivity        object
TravelMode             object
Description            object
Date           datetime64[ns]
Fatalities              int64
dtype: object
```

```
AvyYear                 int64
YYYY                    int64
MM                      int64
DD                      int64
Location               object
Trigger              category
D Size                float64
Setting              category
State                category
lat                   float64
lon                   float64
PrimaryActivity      category
TravelMode           category
Description            object
Date           datetime64[ns]
Fatalities              int64
dtype: object
```

Multiple steps were needed to clean the outliers present in this dataset.

First, the one record reporting an avalanche which caught a party of 10 was removed using a Boolean mask. This resulted in a much cleaner distribution of the variable fatalities. A slight rightward skew is still present.





Removing the records listed as 0 degrees of both latitude and longitude dropped more than 300 rows. Further investigation revealed that another outlier was a report from Alaska. As the only sample from the state, that record was removed since it is part of a different population of data. Similarly, a group of 3 records from New Hampshire were removed for the same reason. This process created a much more balanced looking scatterplot of locations all bounded within 20 degrees of one another, reducing clumping.

After dropping unneeded columns and cleaning outliers, there were still a large number of missing values. Since the total length of the dataset was 981 rows, the dataframe was split to create one with "Trigger" and "D Size", one with "TravelMode", and one without any of those three variables.

```
YYYY                0
MM                  0
DD                  0
Location            0
Trigger           876
D Size            897
Setting             0
State               0
lat                 2
lon                 2
PrimaryActivity     0
TravelMode        314
Date                0
Fatalities          0
dtype: int64
```

Original dataset null counts:

Dataframe with "TravelMode" and no null values:

```
Index: 272 entries, 1 to 545
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   YYYY            272 non-null    int64
 1   MM              272 non-null    int64
 2   DD              272 non-null    int64
 3   Location        272 non-null    object
 4   Setting         272 non-null    category
 5   State           272 non-null    category
 6   lat             272 non-null    float64
 7   lon             272 non-null    float64
 8   PrimaryActivity  272 non-null    category
 9   TravelMode      272 non-null    category
 10  Date            272 non-null    datetime64[ns]
 11  Fatalities      272 non-null    int64
dtypes: category(4), datetime64[ns](1), float64(2), int64(4), object(1)
```

Dataframe with "Trigger" and "D Size" and no null values:

```
Index: 80 entries, 1 to 109
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   YYYY            80 non-null     int64
 1   MM              80 non-null     int64
 2   DD              80 non-null     int64
 3   Location        80 non-null     object
 4   Setting         80 non-null     category
 5   State           80 non-null     category
 6   lat             80 non-null     float64
 7   lon             80 non-null     float64
 8   PrimaryActivity  80 non-null     category
 9   Trigger         80 non-null     category
 10  D Size          80 non-null     float64
 11  Date            80 non-null     datetime64[ns]
 12  Fatalities      80 non-null     int64
dtypes: category(4), datetime64[ns](1), float64(3), int64(4), object(1)
```

Dataframe containing all other variables excluding "Trigger", "D Size", and "TravelMode":

```
Index: 286 entries, 1 to 545
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   YYYY             286 non-null    int64
 1   MM               286 non-null    int64
 2   DD               286 non-null    int64
 3   Location         286 non-null    object
 4   Setting          286 non-null    category
 5   State            286 non-null    category
 6   lat              286 non-null    float64
 7   lon              286 non-null    float64
 8   PrimaryActivity  286 non-null    category
 9   Date             286 non-null    datetime64[ns]
 10  Fatalities       286 non-null    int64
dtypes: category(3), datetime64[ns](1), float64(2), int64(4), object(1)
```

Each of the three dataframes were greatly reduced from the original 981 rows. All three datasets now only contain high quality data free of outliers, erroneous records, and null values. Importantly, the new dataset containing both "Trigger" and "D Size" information is from 2018 – 2023, while the other two datasets span from 1999 – 2023.

The new datasets were written to a new folder as separate and clean .csv files.

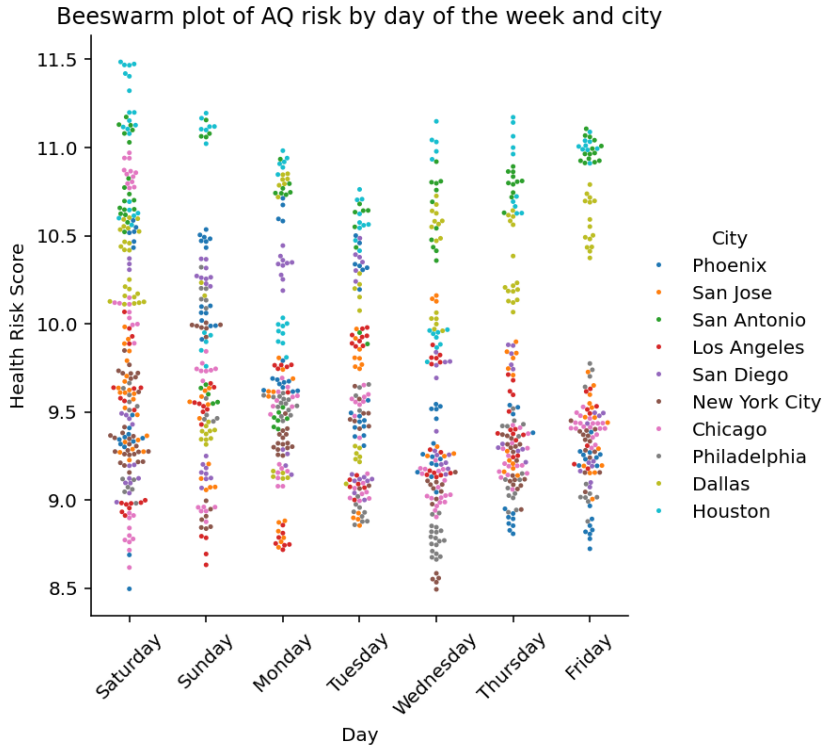
Urban AQI Data

Before Cleaning

Example Data

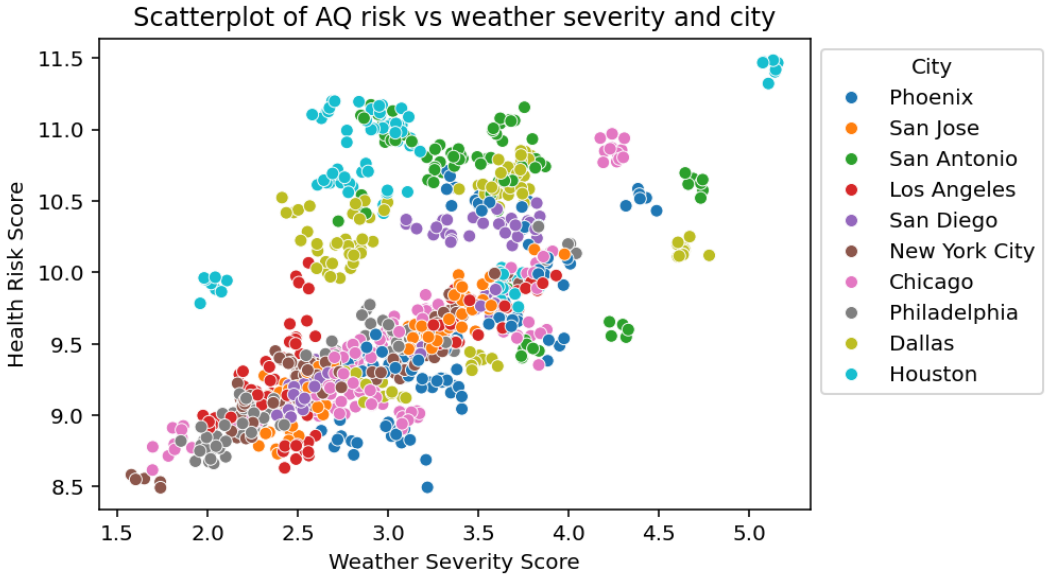
```
       datetime  datetimeEpoch  ...  Is_Weekend  Health_Risk_Score
0    2024-09-07   1.725692e+09  ...        True          10.522170
1    2024-09-08   1.725779e+09  ...        True          10.062332
2    2024-09-09   1.725865e+09  ...       False           9.673387
3    2024-09-10   1.725952e+09  ...       False           9.411519
4    2024-09-11   1.726038e+09  ...       False           9.515179
..          ...            ...  ...         ...                ...
995  2024-09-18   1.726633e+09  ...       False           8.750142
996  2024-09-17   1.726550e+09  ...       False           9.118198
997  2024-09-12   1.726122e+09  ...       False           9.880093
998  2024-09-14   1.726284e+09  ...        True           9.561602
999  2024-09-18   1.726618e+09  ...       False          10.978044

[1000 rows x 46 columns]
```

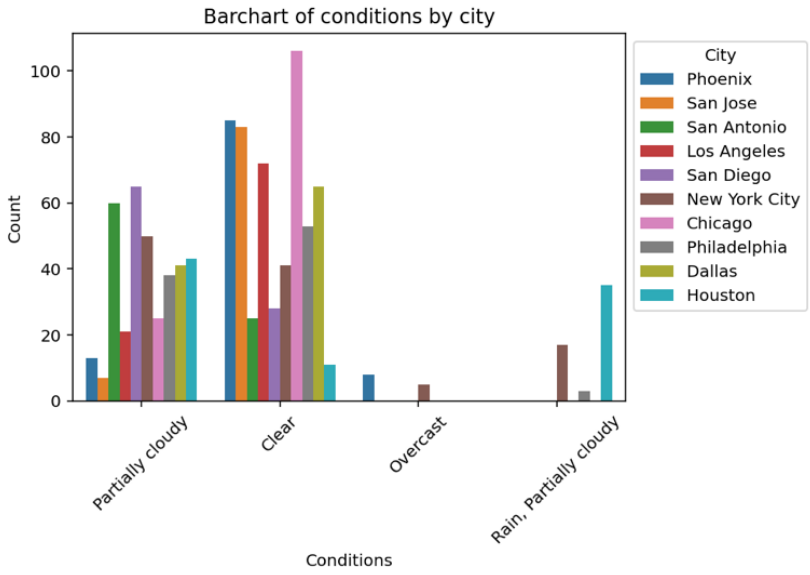
Visualizing Variables



Initial EDA indicates an interesting trend between air pollution and the day of the week. Across the board, the health risk score seems to be lower for most cities on the weekdays. Air quality decreases most on Saturdays in general. This plot does not reveal any outliers or erroneous data points, but it's quite informative for the structure of the data.

Further exploration of air quality, while taking the city in question into account, revealed an interesting linear trend in conjunction with the severity of the day's weather. The scatter plot shows two fairly strong clumps of data. Nearly all the cities in the upper section are located Texas! Only Houston exhibits data approaching an outlier level of significance. More investigation is needed to be sure.





A bar chart of general sky conditions and the cities they occur in most often is shown to the left. Much this plot is fairly sparse, with most cities only reporting either Partially Cloudy or Clear. This could indicate that the data is not very representative of the cities in question. If all these locations always have this nice of weather in September, how can the analysis derive insight on the impact of poor weather on the AQI?

After Cleaning

The Urban AQI dataset was downloaded as a .csv. Initial EDA seemed to indicate that the dataset is quite clean already, as is often the case when data is gathered from Kaggle. After some simple cleaning operations, the final, ready-for-analysis data will be stored as a new .csv.

A majority of the data types for this dataset were read in correctly by Pandas. The "datetime" variable was changed to a datetime type. "City", "conditions", and "Day_of_Week" were all converted to categories. Lastly, "Month" was converted to an integer.

datetime	object
tempmax	float64
tempmin	float64
temp	float64
feelslikemax	float64
feelslikemin	float64
feelslike	float64
dew	float64
humidity	float64
precip	float64
precipprob	float64
precipcover	float64
windgust	float64
windspeed	float64
winddir	float64
pressure	float64
cloudcover	float64
visibility	float64
solarradiation	float64
solarenergy	float64
uvindex	float64
severerisk	float64
sunrise	object
sunset	object
conditions	object
City	object
Temp_Range	float64
Heat_Index	float64
Severity_Score	float64
Month	float64
Day_of_Week	object
Is_Weekend	bool
Health_Risk_Score	float64
dtype: object	

datetime	datetime64[ns]
tempmax	float64
tempmin	float64
temp	float64
feelslikemax	float64
feelslikemin	float64
feelslike	float64
dew	float64
humidity	float64
precip	float64
precipprob	float64
precipcover	float64
windgust	float64
windspeed	float64
winddir	float64
pressure	float64
cloudcover	float64
visibility	float64
solarradiation	float64
solarenergy	float64
uvindex	float64
severerisk	float64
sunrise	object
sunset	object
conditions	category
City	category
Temp_Range	float64
Heat_Index	float64
Severity_Score	float64
Month	int64
Day_of_Week	category
Is_Weekend	bool
Health_Risk_Score	float64
dtype: object	

```
datetime            0
tempmax             0
tempmin             0
temp                0
feelslikemax        0
feelslikemin        0
feelslike           0
dew                 0
humidity            0
precip              0
precipprob          0
precipcover         0
windgust            0
windspeed           0
winddir             0
pressure            0
cloudcover          0
visibility          0
solarradiation      0
solarenergy         0
uvindex             0
severerisk          0
sunrise             0
sunset              0
conditions          0
City                0
Temp_Range          0
Heat_Index          0
Severity_Score      0
Month               0
Day_of_Week         0
Is_Weekend          0
Health_Risk_Score   0
dtype: int64
```

Once the columns, 'datetimeEpoch', 'sunriseEpoch', 'sunsetEpoch', 'moonphase', 'description', 'icon', 'stations', 'source', 'Condition_Code', 'preciptype', 'snow', 'snowdepth', 'Season' were dropped, not a single null value remained in the dataset. These columns were dropped because they either contained all 0's, as in the case of the "snowdepth" variable, or because they seemed unnecessary for analysis.

Working with all 25 quantitative columns to discover outliers was a challenge. In the end, the 1st and 3rd quartiles, and the interquartile range for each column was calculated. From there, a Boolean mask using the quartile outlier formula was used to uncover columns with at least one record that contained a possible outlier.

$$low\ outliers = Q1 - 1.5 * IQR$$
$$high\ outliers = Q3 + 1.5 * IQR$$

Using a loop, each of the suspicious variables was inspected with a boxplot. From this exploration, only two variables had outliers of any significance: "dew" and "windspeed". Both records were cleaned using a Boolean mask.

Lastly, the clean dataset was output to a .csv in a new location separate from the raw data.

Unsupervised Learning with k-means Clustering

Formatting the data

The dataset chosen for the k-means model is the ERA5 dataset for the Gulf of Alaska. This dataset was cleaned and formatted into a K-means friendly mode in the section, ERA5 Meteorological Data. Prior to this process, the dataset consisted of multiple variables which were not needed for clustering. These columns included such information as timestamps, latitude, and longitude. The dataset also included a variable for air pressure, this column can act somewhat like a label for the data and so was removed for this analysis.

In order for k-means to cluster effectively, the columns discussed above were dropped from the dataframe. From there, the data is in a purely quantitative format. However, there was a large discrepancy between datapoints. This can be a problem for ML models that use geometric distance formulas as part of their algorithm. To avoid this pitfall, the remaining data was normalized using SKLearn's MinMaxScaler class.

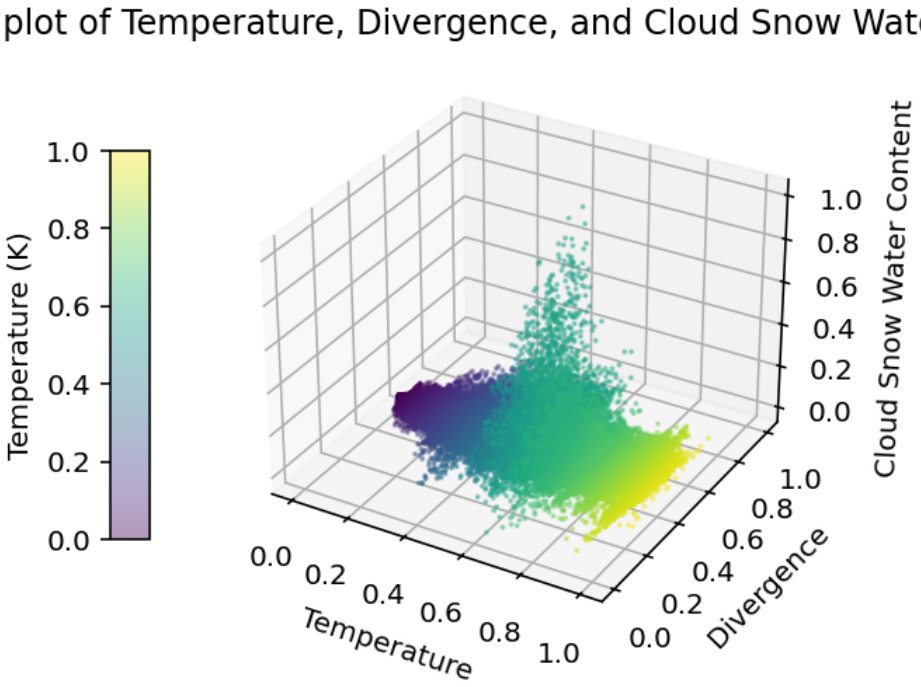
Before Formatting:

```
In [18]: before
Out[18]:
                       time  isobaricInhPa  ...          u         v
0                2024-01-12           1000  ...  -1.099014  2.905258
1                2024-01-12           1000  ...  -0.929092  2.958969
2                2024-01-12           1000  ...  -0.861710  3.075180
3                2024-01-12           1000  ...  -0.954483  3.277328
4                2024-01-12           1000  ...  -1.282608  3.582993
...                     ...            ...  ...        ...       ...
966870  2024-01-12 23:00:00            600  ...  12.761536 -2.962769
966871  2024-01-12 23:00:00            600  ...  12.406067 -2.665894
966872  2024-01-12 23:00:00            600  ...  11.988098 -2.482300
966873  2024-01-12 23:00:00            600  ...  11.458801 -2.335815
966874  2024-01-12 23:00:00            600  ...  10.867004 -2.238159

[966875 rows x 11 columns]
```

After Formatting and Normalizing:

```
In [19]: df
Out[19]:
              d         q  crwc      cswc         t         u         v
0      0.534010  0.209805   0.0  0.013216  0.614698  0.444417  0.517974
1      0.531368  0.211202   0.0  0.014173  0.613062  0.446380  0.518559
2      0.526724  0.212124   0.0  0.015410  0.612413  0.447158  0.519826
3      0.517943  0.213402   0.0  0.016768  0.612413  0.446087  0.522029
4      0.506974  0.215751   0.0  0.018509  0.612301  0.442297  0.525361
...         ...       ...   ...       ...       ...       ...       ...
966870 0.519222  0.135532   0.0  0.000000  0.644787  0.604501  0.454016
966871 0.521184  0.137762   0.0  0.000000  0.645351  0.600396  0.457252
966872 0.521611  0.142236   0.0  0.000000  0.645506  0.595568  0.459253
966873 0.519650  0.145818   0.0  0.000000  0.645549  0.589455  0.460849
966874 0.521331  0.151913   0.0  0.000000  0.645675  0.582620  0.461914

[966875 rows x 7 columns]
```

Visualization in 3D



This scatterplot shows Temperature in Kelvin, Divergence in s^{-1}, and Specific Snow Water Content in $\frac{kg}{kg^{-1}}$. All data in this dataset is instantaneous and has been normalized using the MinMaxScaler. Divergence measures the horizontal speed at which a mass of air is moving away from a single point in space. The metric can be used to understand where geographically and low or high pressure system may be forming. The Specific Snow Water Content variable describes the mass of ice crystals produced from a cloud.

Examining this plot shows an interesting and somewhat unexpected trend. The amount of ice crystals in clouds does not perfectly increase as the temperature drops. Instead, it would appear that as temperature decreases, snow water content increases to a point; it then begins to drop again. Perhaps, at particularly cold temperatures, clouds themselves do not readily form. As divergence increases, a small increase in snow water content can be observed. This could be due to stormy low pressure conditions being correlated with larger snow events.

Applying k-means:

```
-------------------- KMeans, k=2 --------------------
Cluster centers:
[[5.32752375e-01 9.02030139e-02 3.59599564e-05 5.35579039e-03
  5.14527623e-01 5.06333597e-01 4.85389479e-01]
 [5.30282844e-01 4.17503766e-01 1.30368615e-02 3.97081146e-03
  7.45580127e-01 4.75852322e-01 5.33176577e-01]]
Labels: [0 0 0 ... 0 0 0]
Predicting on fake point: [1]
-----------------------------------------------------
```

```
-------------------- KMeans, k=3 --------------------
Cluster centers:
[[5.31880712e-01 1.40188348e-01 1.13836449e-03 8.03956947e-03
  6.09351828e-01 5.02368249e-01 5.19035840e-01]
 [5.30250763e-01 4.71872023e-01 1.54098194e-02 5.27048254e-04
  7.77622254e-01 4.66093238e-01 5.25951296e-01]
 [5.34148896e-01 3.19360278e-02 4.82215269e-10 1.23314379e-03
  3.09853980e-01 5.15566707e-01 4.19750734e-01]]
Labels: [0 0 0 ... 0 0 0]
Predicting on fake point: [1]
-----------------------------------------------------
```

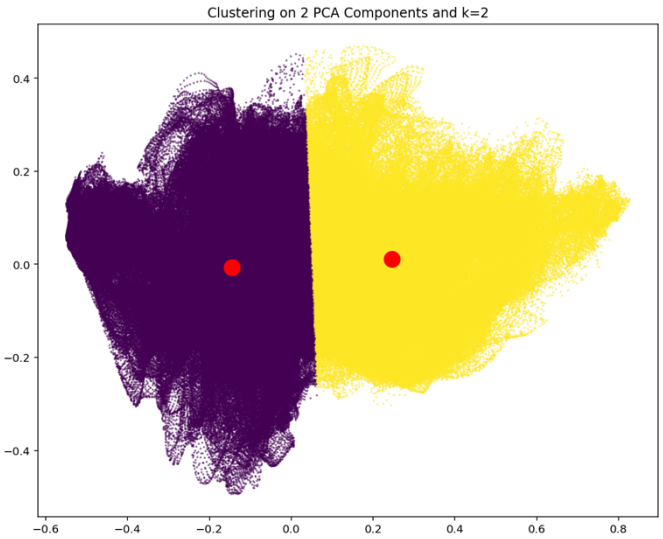
```
-------------------- KMeans, k=4 --------------------
Cluster centers:
[[ 5.30292142e-01  4.83181854e-01  1.49026700e-02  1.50847795e-04
   7.84653992e-01  4.61811607e-01  5.22442961e-01]
 [ 5.34127222e-01  2.98220258e-02 -9.61036806e-16  1.08970051e-03
   2.94631828e-01  5.13855294e-01  4.17181514e-01]
 [ 5.29547320e-01  1.84102517e-01  3.21793778e-03  1.51225559e-02
   5.56749175e-01  5.23149163e-01  6.15381198e-01]
 [ 5.33957174e-01  1.07121859e-01  3.98081009e-04  1.33413929e-03
   6.51258754e-01  4.86762390e-01  4.32772400e-01]]
Labels: [2 2 2 ... 3 3 3]
Predicting on fake point: [0]
-----------------------------------------------------
```

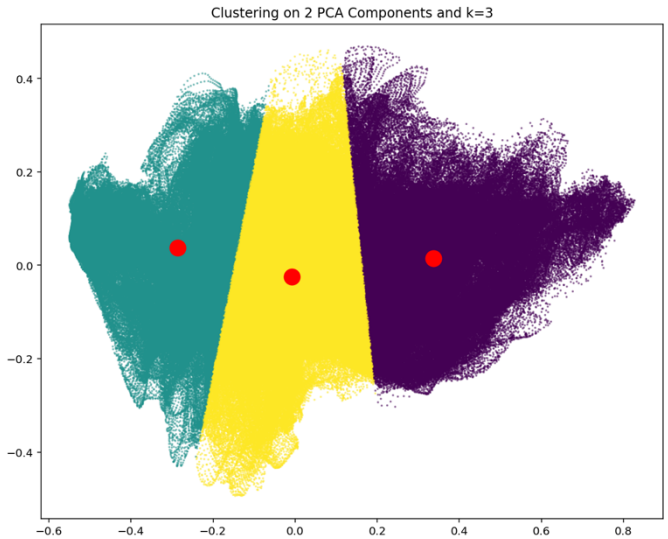
Applying k-means, at $k = [2, 3, 4]$, to this dataset results in the above output. The false point created to use for a prediction was [0.482371, 0.673214, 0.023453, 0.512987, 0.618472, 0.473912, 0.398745]. This is simply a series of random numbers on the same scale as the rest of the data. The results shown here are extremely difficult to interpret. With 7 columns, all on the same scale between 0 and 1, any pattern is far from easy to see. The fake point created for the purpose of predicting using the model always seems to lie on the low end of cluster assignments.

k-means Conclusions

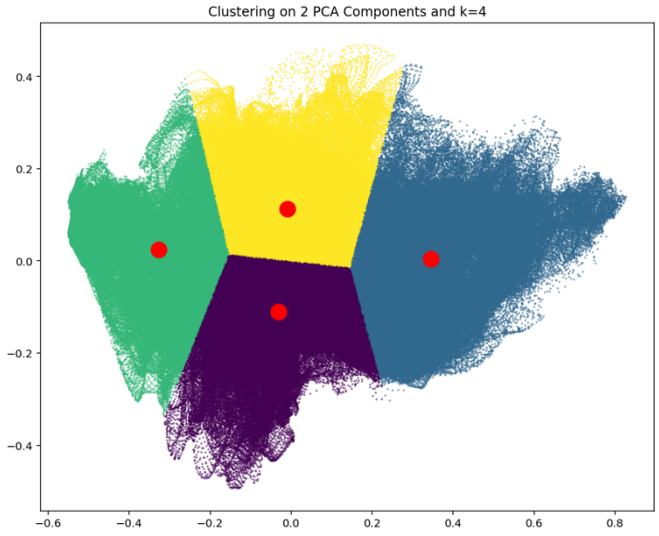
To find a more interpretable of visualization of how this data clustered, the process of Principal Component Analysis was used to reduce the number of dimensions in the data from 7 to 2. This process works by analyzing the dataset and discovering those variables which create the most variability. The technique iteratively and greedily removes columns that provide the least spread in the data points.

Once the data is reduced in dimensionality, k-means can be run once more and the resulting clusters are easily plotted in 2D:





One of the primary conditions for k-means clustering is that separate groups exist within the data in roughly spherical shapes. Knowing this, the reader may be able to see that this particular data is unsuitable for k-means clustering. While the dimensionally reduced dataset does result in a wonderfully circular object, there are no obvious delineations between groups of points. Therefore, running a clustering algorithm on this data serves only to partition out the single group into k blocks.



The failing of the this process was probably not due to the k-means model or data preparation process, but rather because of the data itself. Since this data represents atmospheric conditions in the Gulf of Alaska on January 1st 2024, it may make sense that the data show a rather uniform distribution. For instance, if this process were to be applied to a mixed dataset from the Gulf of Alaska and also from Mumbai, India, perhaps the end result with $k = 2$ would show a pair of clusters, one from each geographical location.

Even without a desirable outcome, however, the process of applying k-means to the ERA5 data was a valuable experience in data wrangling, visualization, and analysis.

Supervised Learning with Decision Trees

Data Formatting

The Urban AQI data cleaned in the previous section was chosen for this demonstration. Before decision trees can be implemented, this particular dataset must undergo some formatting.

	datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	...	Heat_Index	Severity_Score	Month	Day_of_Week	Is_Weekend	Health_Risk_Score
0	2024-09-07	106.100000	91.000000	98.500000	104.000000	88.100000	...	95.918703	4.430000	9	Saturday	True	10.522170
1	2024-09-08	103.900000	87.000000	95.400000	100.500000	84.700000	...	92.281316	3.880000	9	Sunday	True	10.062332
2	2024-09-09	105.000000	83.900000	94.700000	99.900000	81.600000	...	90.599165	3.630000	9	Monday	False	9.673387
3	2024-09-10	106.100000	81.200000	93.900000	100.600000	79.500000	...	89.638811	2.851200	9	Tuesday	False	9.411519
4	2024-09-11	106.100000	82.100000	94.000000	101.000000	80.000000	...	89.760414	3.390800	9	Wednesday	False	9.515179
..	...	...	...	...	...	...	...	...	...	...	...	...	...
993	2024-09-18	76.060546	64.359387	69.002142	77.673823	63.510920	...	71.837558	1.957318	9	Wednesday	False	8.750142
994	2024-09-17	68.409198	65.939319	66.567410	68.956722	64.805635	...	72.463491	2.537413	9	Tuesday	False	9.118198
995	2024-09-12	69.756690	65.286919	65.919492	68.158536	63.662942	...	67.560060	3.595470	9	Thursday	False	9.880093
996	2024-09-14	77.106797	61.481724	68.106569	76.426959	60.901526	...	67.930437	3.498942	9	Saturday	True	9.561602
997	2024-09-18	90.923080	79.296868	81.636991	94.180423	78.071851	...	86.802712	3.040020	9	Wednesday	False	10.978044

[998 rows x 33 columns]

As can be seen in the above image, this dataset contains a variety of non-quantitative variables. In order for SciKit-Learn's Decision Tree model to function, the fitted dataset must contain only quantitative data. As per that requirement, the following variables were removed from the dataset: 'datetime', 'sunrise', 'sunset', 'conditions', 'City', 'Day_of_Week', and 'Is_Weekend'.

```
RangeIndex: 998 entries, 0 to 997
Data columns (total 33 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   datetime           998 non-null    object
 1   tempmax            998 non-null    float64
 2   tempmin            998 non-null    float64
 3   temp               998 non-null    float64
 4   feelslikemax       998 non-null    float64
 5   feelslikemin       998 non-null    float64
 6   feelslike          998 non-null    float64
 7   dew                998 non-null    float64
 8   humidity           998 non-null    float64
 9   precip             998 non-null    float64
 10  precipprob         998 non-null    float64
 11  precipcover        998 non-null    float64
 12  windgust           998 non-null    float64
 13  windspeed          998 non-null    float64
 14  winddir            998 non-null    float64
 15  pressure           998 non-null    float64
 16  cloudcover         998 non-null    float64
 17  visibility         998 non-null    float64
 18  solarradiation     998 non-null    float64
 19  solarenergy        998 non-null    float64
 20  uvindex            998 non-null    float64
 21  severerisk         998 non-null    float64
 22  sunrise            998 non-null    object
 23  sunset             998 non-null    object
 24  conditions         998 non-null    object
 25  City               998 non-null    object
 26  Temp_Range         998 non-null    float64
 27  Heat_Index         998 non-null    float64
 28  Severity_Score     998 non-null    float64
 29  Month              998 non-null    int64
 30  Day_of_Week        998 non-null    object
 31  Is_Weekend         998 non-null    bool
 32  Health_Risk_Score  998 non-null    float64
dtypes: bool(1), float64(25), int64(1), object(6)
memory usage: 250.6+ KB
```



```
RangeIndex: 998 entries, 0 to 997
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   tempmax            998 non-null    float64
 1   tempmin            998 non-null    float64
 2   temp               998 non-null    float64
 3   feelslikemax       998 non-null    float64
 4   feelslikemin       998 non-null    float64
 5   feelslike          998 non-null    float64
 6   dew                998 non-null    float64
 7   humidity           998 non-null    float64
 8   precip             998 non-null    float64
 9   precipprob         998 non-null    float64
 10  precipcover        998 non-null    float64
 11  windgust           998 non-null    float64
 12  windspeed          998 non-null    float64
 13  winddir            998 non-null    float64
 14  pressure           998 non-null    float64
 15  cloudcover         998 non-null    float64
 16  visibility         998 non-null    float64
 17  solarradiation     998 non-null    float64
 18  solarenergy        998 non-null    float64
 19  uvindex            998 non-null    float64
 20  severerisk         998 non-null    float64
 21  Temp_Range         998 non-null    float64
 22  Heat_Index         998 non-null    float64
 23  Severity_Score     998 non-null    float64
 24  Month              998 non-null    int64
 25  Health_Risk_Score  998 non-null    float64
dtypes: float64(25), int64(1)
memory usage: 202.8 KB
```

Label Engineering

Since Urban AQI has already been stringently cleaned by analyzing and removing outliers, correcting erroneous data, and dropping missing value, the last step before working with the model is to engineer a feature label. In this case, the variable, 'Health_Risk_Score', is intended to be a continuous label. Since Decision Trees is a classifier, this label set must be discretized.

The resulting column is named 'health_score_label' and can be seen on the far right of the dataset below. The maximum value was 11.49 and the minimum value was 8.49. This range was binned into 5 discrete labels (Minimal, Low, Medium, Hight, Severe).

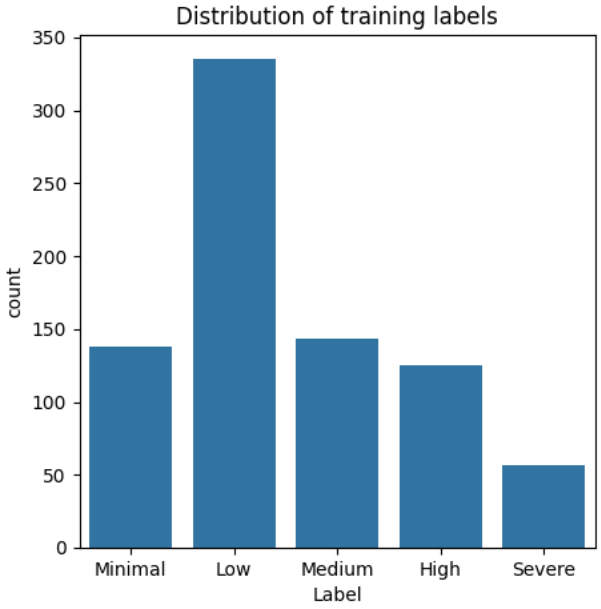
	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	...	severerisk	Temp_Range	Heat_Index	Severity_Score	Month	health_score_label
0	106.100000	91.000000	98.500000	104.000000	88.100000	95.900000	...	10.000000	15.100000	95.918703	4.430000	9	High
1	103.900000	87.000000	95.400000	100.500000	84.700000	92.300000	...	10.000000	16.900000	92.281316	3.880000	9	Medium
2	105.000000	83.900000	94.700000	99.900000	81.600000	90.600000	...	10.000000	21.100000	90.599165	3.630000	9	Low
3	106.100000	81.200000	93.900000	100.600000	79.500000	89.800000	...	10.000000	24.900000	89.638811	2.851200	9	Low
4	106.100000	82.100000	94.000000	101.000000	80.000000	90.000000	...	10.000000	24.000000	89.760414	3.390800	9	Low
..	...	...	...	...	...	...	...	...	...	...	...	...	...
993	76.060546	64.359387	69.002142	77.673823	63.510920	67.003338	...	10.563084	12.886665	71.837558	1.957318	9	Minimal
994	68.409198	65.939319	66.567410	68.956722	64.805635	65.992526	...	9.838767	2.613629	72.463491	2.537413	9	Low
995	69.756690	65.286919	65.919492	68.158536	63.662942	67.313322	...	10.502440	4.598936	67.560060	3.595470	9	Medium
996	77.106797	61.481724	68.106569	76.426959	60.901526	68.094309	...	9.847929	15.477717	67.930437	3.498942	9	Low
997	90.923080	79.296868	81.636991	94.180423	78.071851	84.987113	...	30.395643	11.017871	86.802712	3.040020	9	Severe

[998 rows x 26 columns]

Test-Train Split

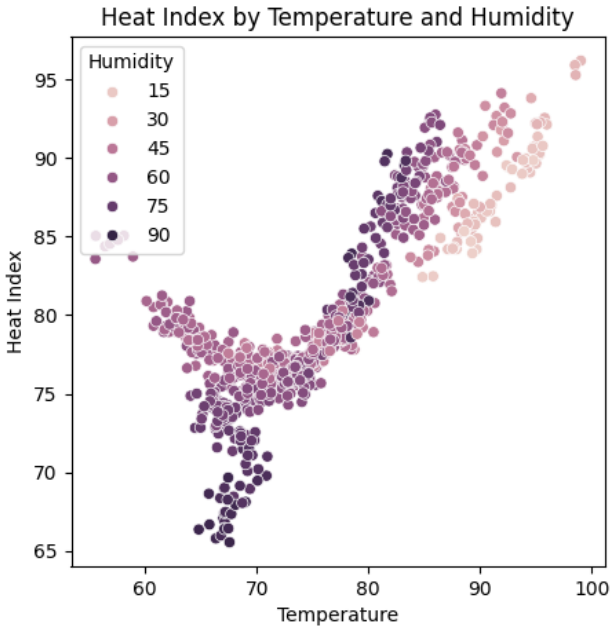
In order to validate the efficacy of a model, it's common for the data to split into a training set and a validation set. Here, the Urban AQI dataset was divided along an 80-20 split, 80% of data partitioned for training, and the remaining 20% for testing.

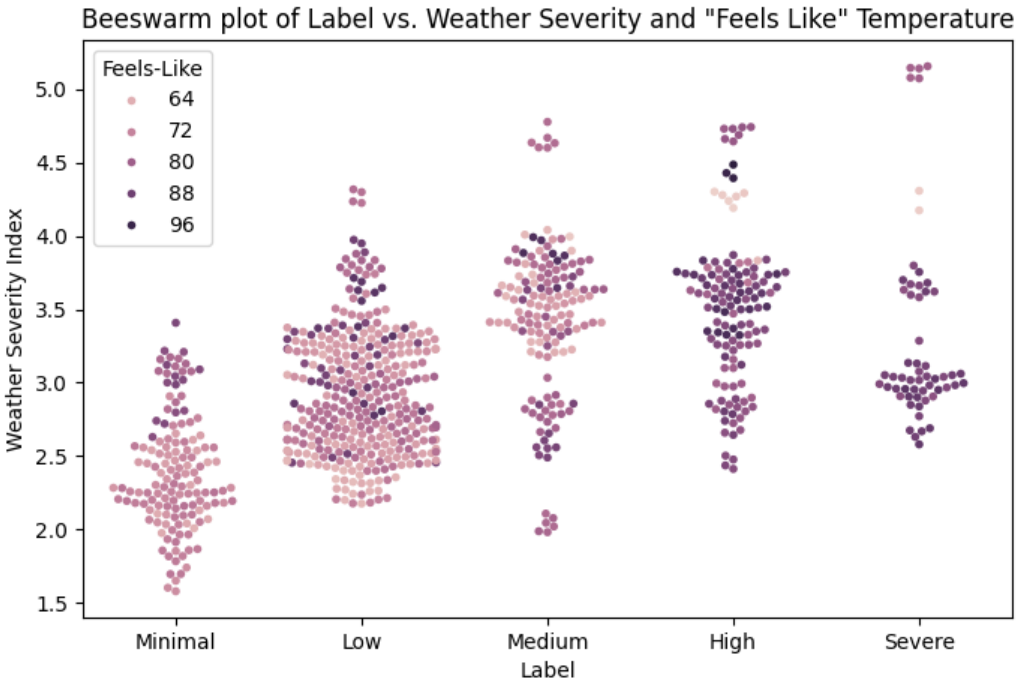
Training Visualization



The bar chart to the left shows the distribution of labels in the training data. A strong right skew is evident and a single mode centered over the Low label can be seen. Out of 798 labels, the Low label occurs 335 times. This ratio of 0.42 is indicative of an imbalance in the training data. Further steps to correct this imbalance should be taken.

The scatterplot to the right is an exploratory plot examining the relationship between temperature, humidity, and heat index in the training dataset. Unexpectedly, there appears to be a non-linear relationship between temperature and heat index. A sharp curve is quite obvious in the temperature range between 60 and 80 degrees Fahrenheit. From there, the relationship is strongly linear. Looking at this plot, and taking into account humidity, it's possible that there is a tipping point where the weather's humidity outweighs the effect of temperature on heat index.





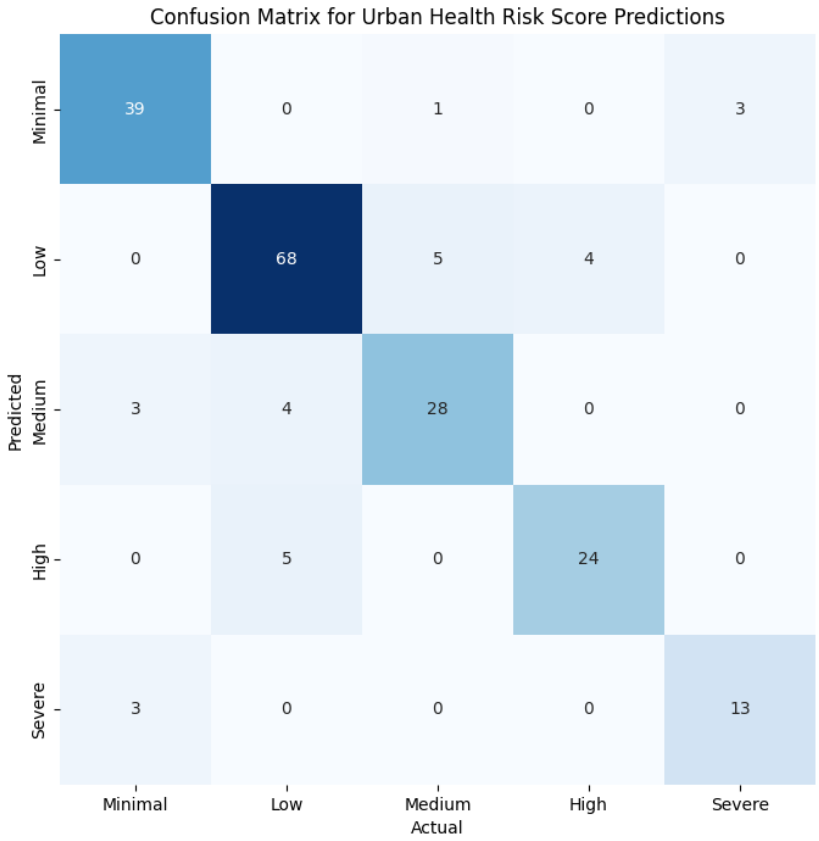
The above beeswarm plot examines the distribution of labels in the training set, taking into account the effects of the ambient "feels-like" temperature, and the weather severity index. It's useful to examine labels vs. severity since this dataset has nearly 25 columns of metrics detailing the instantaneous weather conditions for each observation. The weather severity column takes all of these details into account and collates them into a single metric. This plot then shows an interesting relationship between a city's heath score and the severity of the weather. It is evident that the majority of high and severe health risk days occur on days where the "feels-like" temperature is high.

Applying Decision Trees

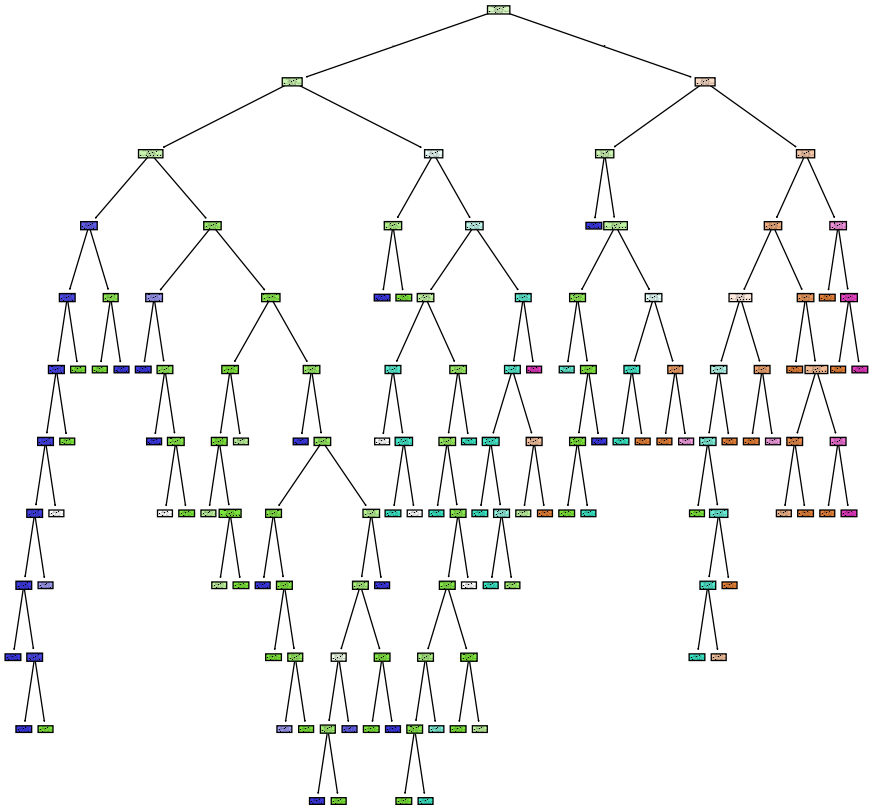
Walking the Decision Tree through the validation process, an eventual validation accuracy of 86% was settled upon. This is a fairly strong score, and could be indicative of overfitting. It's worth noting that the usefulness of hold-out testing is limited. Without testing the model on multiple sets of data, such as with K-fold cross-validation, it can be difficult to get a sense of model performance.

In this case, the decision tree was limited to a maximum depth of 15, features to 14, and the minimum number of samples to split a node to 8. On a dataset with 26 variables, these settings could possibly limit overfitting while still optimizing for predictive capabilities.

Confusion Matrix



Tree Visualization



Conclusion

Decision Trees are an impressive model, highly flexible, and very commonly used in many contexts. A key aspect of the model is its transparency. Once a decision tree has been fit to the data, a new prediction can be traced from top to bottom. For instance, if a mortgage lending company turns down an application for a loan based on the output of a Decision Tree, the reasons why are readily available. The company can trace the applications route through the model's decision making process and point to key nodes with a significant impact on the eventual denial. This is quite a powerful feature of the model and can help to uncover bias or flaws in the prediction.

In the case of predicting a city's air quality health score from a large variety of meteorological factors, a Decision Tree does quite well. Some brief insights can be gleaned from the diagram above. In example, the majority of Low and Minimal scoring days depend on a heat index below 80.292, wind gusts less than 14.74 mph, and a general weather severity score under 2.31. On the other hand, a city is likely to have a Severe or High scoring day with a heat index above 80.292 and a dew point less than 70.63.

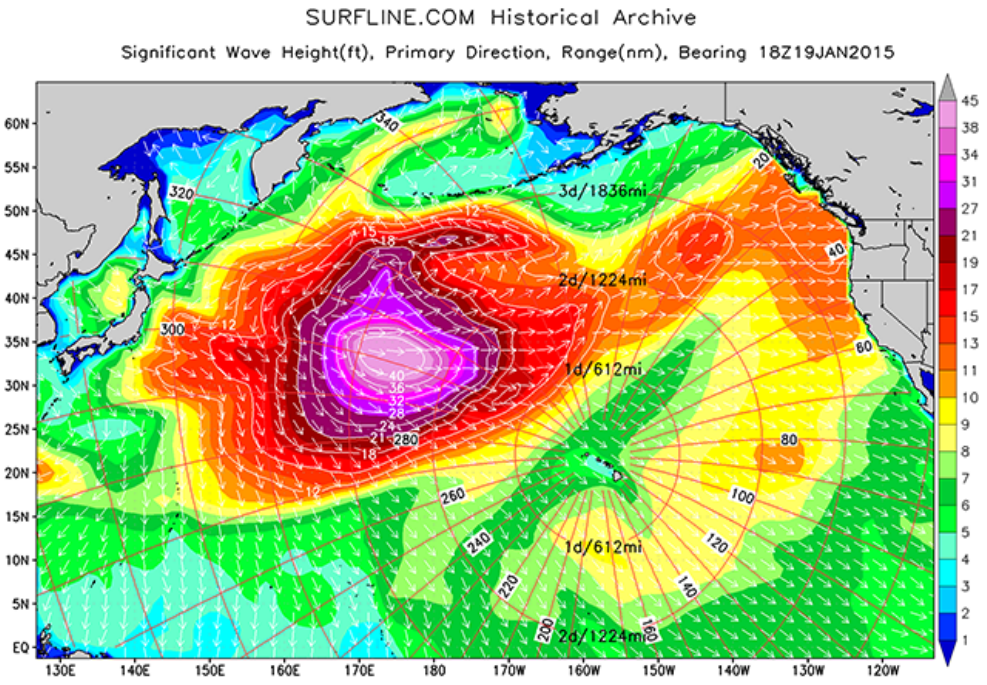
Final Conclusions

The weather is a fascinating and wide domain to explore. This project has covered the gamut from building supervised machine learning models with an urban air quality dataset to cluster analysis on a small subsection of the massive ERA5 dataset over the Gulf of Alaska. This diverse breadth of information and applications is what makes the study of the atmosphere so exciting.

By seeking to understand how weather behaves on our planet, we can better learn to coexist with the ever changing natural processes all around us. For instance, as humanity slowly transitions to renewables as a primary energy source, our entire energy grid will become reliant on the weather that generates our electricity. Supply chains represent another extreme vulnerability to the weather. Without strong knowledge of the currents and storms at sea, much of the food and technology that our culture subsists on could be interrupted while shipping.

With so much of our society completely dependent on the weather, it's easy to see why there is such a vast plethora of data informing the subject. The Earth is a seriously complex system encompassing and containing nearly all the other systems that humanity has studied. As technology and computational techniques progress, so too does our ability to understand and predict these large systems. Massive versions of the machine learning models discussed in this paper are currently being trained on datasets that dwarf the toy data used for this project.

Though much atmospheric data is used for activities critical to the functioning of our country, many models simply exist for our convenience and pleasure. With these models, I, an avid surfer, can try to choose the best time of day to catch some waves. With a similar approach, the Northwest Avalanche Center can assess slide risk for those athletes adventuring in the mountains. Regardless of the use case, the data explored here in this project represents the very tip of the tip of the iceberg of knowledge that can be applied to everyday life in a modern culture.



2. Swell chart depicting a very large low depression generating extra-large surf in the NE Pacific. Source: www.surfline.com

To Top

References

1. Price, I., Sanchez-Gonzalez, A., Alet, F. et al. *Probabilistic weather forecasting with machine learning.* Nature 637, 84–90 (2025). https://doi.org/10.1038/s41586-024-08252-9