Model documentation and write-up

1. Who are you (mini-bio) and what do you do professionally? If you are on a team, please complete this block for each member of the team.

I work as a Senior Machine Learning Engineer for Uplight, Inc. where I build production machine learning systems that help orchestrate energy assets on the electric grid.

2. What motivated you to compete in this challenge?

Growing up and living in Denver, Colorado I have long been aware of the importance of seasonal streamflow to communities across the Western United States. I wanted to take this opportunity to learn more about the dynamics that shape streamflow from season to season and help give back to the community.

3. If there are any particularly useful snippets of code when producing your communication outputs (calculations, visualizations, etc.) that would be useful to highlight, please copy-paste them here.

For me, some of the more fun data visualizations involve geographic data. The code below uses a USA State Shape file to find the relevant states for a water basin of a stream site and plots them. Next, it plots the boundary line of the basin on top of the state outlines. It then finds the most correlated snotel stations for stream site and plots the location of those stations and it colors them based on the z-score of the snow water equivalent measurement. It also includes a plot using of where the streamflow measurement takes place.

This code was fun to write because it combines several data elements from different sources and tells a story about the relative snow water equivalent measurements for a given stream site and their proximity to the water basin and streamflow measurement point.

```
gdf = gpd.GeoDataFrame(station_deviation,

geometry=gpd.points_from_xy(station_deviation.longitude, station_deviation.latitude))
# USA State Shape File Available at -
https://www2.census.gov/geo/tiger/GENZ2018/shp/cb_2018_us_state_500k.zip us_map =
gpd.read_file(data_dir /
'cb_2018_us_state_500k/cb_2018_us_state_500k.shp')
site_meta = metadata[metadata['site_id'] == site_id]


site_mask = us_map.apply(lambda row: Point(site_meta[['longitude',
```

```
'latitude']].values[0]).within(row['geometry']), axis=1)
site_states = set()
site_states.add(us_map[site_mask]['STUSPS'].values[0])
for idx, rw in corr_stations.iterrows():
station_mask = us_map.apply(lambda row: Point(rw[['longitude',
'latitude']].values).within(row['geometry']), axis=1)
site_states.add(us_map[station_mask]['STUSPS'].values[0])

site_map = us_map[us_map['STUSPS'].isin(site_states)]
site_map = site_map.to_crs("EPSG:3395")
gdf = gdf.set_crs("EPSG:4326")
gdf = gdf.to_crs("EPSG:3395")
meta_gdf = gpd.GeoDataFrame(metadata,
geometry=gpd.points_from_xy(metadata.longitude, metadata.latitude)) meta_gdf =
meta_gdf.set_crs("EPSG:4326")
meta_gdf = meta_gdf.to_crs("EPSG:3395")
drainage_basins = gpd.read_file(data_dir / 'geospatial.gpkg')
drainage_basins = drainage_basins.to_crs("EPSG:3395")

fig, ax = plt.subplots(figsize=(10,8))
site_map.boundary.plot(ax=ax, color='black')
drainage_basins[drainage_basins['site_id'] == site_id].plot(ax=ax, alpha=0.4,
color='black')
drainage_basins[drainage_basins['site_id'] == site_id].boundary.plot(ax=ax,
alpha=0.9, color='black')
cmap = plt.get_cmap('RdYlBu')
pc = gdf[gdf['date'] == issue_date].plot(
ax=ax,
alpha=0.9,
column='snotel_wteq_deviation',
legend_kwds={"shrink": 0.65},
legend=True,
cmap=cmap,
vmin=-3,
vmax=3,

markersize=150,
edgecolors='black'
)
meta_gdf[meta_gdf['site_id'] == site_id].plot(ax=ax, color='yellow',
```

```
edgecolor='black', markersize=250, marker='X', zorder=3, alpha=0.8) plt.title(f'Fig
13: Snotel SWE Deviation (Correlated Stations) on {issue_date}') ax.set_yticks([])
ax.set_xticks([])
for idx, row in site_map.iterrows():
plt.annotate(text=row['NAME'],
xy=(row['geometry'].centroid.xy[0][0],
row['geometry'].centroid.xy[1][0]),
horizontalalignment='center',
alpha=0.7)
plt.savefig(plot_dir / 'fig13.png', bbox_inches='tight')
```

4. Please provide the machine specs and time you used to train your model and to produce the communication outputs.

- CPU (model): 2.6 GHz 6-Core Intel Core i7
- GPU (model or N/A): N/A
- Memory (GB): 16 GB 2667 MHz DDR4
- OS: Mac OS Ventura 13.6.4
- Train duration: 2 hours to generate train features
- Inference duration: Cross-validation trains new models and performs inference for each cv year. It takes a little less than 2 hours to run
- Software used to prepare the forecast summary: Python (matplotlib, pandas, seaborn)

5. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?

When generating the training data the UA Swann datasource https://snowview.arizona.edu/csv/Download/Watersheds/ website goes down temporarily. In these instances I've needed to wait for the website to come back up to run the generate training features code.

6. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

N/A

7. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

I like shap values for explaining which features were particularly impactful for a specific prediction. The final prediction of the model is an ensemble between different models, so I tried to find a way to use a weighted average of the shap values between several models to get a holistic view of which features were most important to the ensemble. I ran into issues combining shap values from different models and maintaining a consistent mapping of how a feature quantitatively changed a given prediction. I had to abandon this approach and instead opted to show the shap values for each model in the ensemble in separate plots.

8. If you were to continue working on your explainability/communication solution for the next year, what methods or techniques might you try in order to build on your work so far? Are there other metrics or visualizations you felt would have been very helpful to have?

If I were to keep working on the explainability solution I would look for visualizations or analytics that could help improve the model by identifying which features or measurement locations are most relevant for specific stream sites. One potential area for exploration is whether certain features are more important for a specific stream site at different times of the year. It may be possible to visualize the data to help inform intuitions along these lines.