Data Lab 3: Cloud Prediction

ACMS 40950/60876 (Spring 2026)

Due Monday, March 16, 2026 11:59 PM via GitHub

Background and Goals

One of the most important and powerful concepts in AI/ML is prediction. Given some training data, we often want to use the observed data to predict some responses of interest such that this model generalizes well to future, unseen (test) data. However, as in the unsupervised learning workflow shown in lab 2, there is more to the ML prediction workflow than the prediction model itself. This includes many of the concepts that we have seen already in previous labs (e.g., problem formulation, data collection, data cleaning/preprocessing, and exploratory data analysis), but it also includes other ideas like post-hoc exploratory data analysis and scrutinization of results that come after the prediction modeling.

In this lab, you will build a prediction modeling pipeline to help NASA scientists detect clouds in polar regions based upon image data recorded aboard the NASA satellite Terra. In building this pipeline, you will work through the full data science life cycle — from problem formulation through the prediction modeling through communication of results — as a true data scientist.

Instructions

Imagine that you are collaborating with NASA scientists, who would like to build a prediction model to distinguish cloudy versus non-cloudy regions from satellite images of the Earth's poles. After talking with your collaborators, you learn that cloud detection is particularly difficult in polar regions since both cloudy and icy regions look similar in these satellite images to the human eye. For this reason, the NASA scientists spent a lot of valuable time and energy to manually hand-label pixels from three satellite images. You may find the satellite image data and the "expert labels" on Canvas in the `lab3/` folder. Each of these data files contains the following measurements:

01	y coordinate of the satellite image
02	x coordinate of the satellite image
03	expert label (+1 = cloud, -1 = not cloud, 0 = uncertain)
04	NDAI
05	SD
06	CORR
07	Radiance angle DF
08	Radiance angle CF
09	Radiance angle BF
10	Radiance angle AF
11	Radiance angle AN

At a high level, the radiance angles (AN, AF, BF, CF, DF) are the raw radiances recorded automatically by the MISR sensor aboard the NASA satellite Terra. More information on MISR is available at http://www-misr.jpl.nasa.gov/. The other three satellite image features (NDAI, SD, and CORR) are custom-engineered features, specifically developed for this cloud detection problem and described in the article `yu2008.pdf`.

To help your scientific collaborators, your objective is to use this small expertly-labeled dataset to train a cloud detection model with the end goal of deploying it to distinguish between cloudy versus non-cloudy polar regions on a large number of images (also from the satellite Terra) that do not have these "expert labels."

Exploratory Data Analysis

1. For each of the three satellite images, plot the expert labels for the presence or absence of clouds as an image (i.e. using the provided x- and y-coordinates).
2. Explore and discuss the following relationships, both visually and quantitatively:
   (a) The relationship between different radiances (AN, AF, BF, CF, DF)
   (b) The relationship between different engineered features (CORR, NDAI, and SD)
   (c) The relationship between the radiances and the engineered features
3. Explore and discuss the relationship between the expert cloud labels and each predictor feature (i.e., radiances and engineered features), both visually and quantitatively.

Prediction Modeling

4. Given the expert labels of (+1 = cloud, -1 = not cloud, 0 = uncertain), there are many potential ways to *formulate* your prediction task. For example, one could formulate this as a multi-class classification problem or transform this problem into a binary classification problem. Clearly formulate and define your prediction problem. Justify your choice. (Note: you may choose to revisit or modify this formulation after preliminary prediction modeling.)
5. Perform data splitting. Detail your data splitting scheme *carefully* and justify your choices. Be sure to include sufficient details so that a general reader can reproduce your analysis.
6. Choose at least two classification models to build for your formulated prediction problem. At least one of your classification models needs to require hyperparameter tuning. For each of these models,
   (a) Provide a brief description of the classifier.
   (b) State the assumptions of the classification model (if any), and if applicable, investigate whether the model assumptions are reasonable for the given data.

(c) Fit the model (i) using only the radiance-based features, (ii) using only the engineered features (i.e., NDAI, CORR, and SD), and (iii) using both the radiance-based and the engineered features.

- If hyperparameter tuning is needed, be sure to tune these hyperparameters carefully. Clearly detail how you tuned these hyperparameters, and justify this choice. In addition, document the grid of hyperparameters that you searched over. If there are additional model hyperparameters that were not tuned, please also make sure to document the fixed values for these hyperparameters. (Remember to include sufficient information so that your analysis can be easily reproduced from reading the text.) Finally, document the best tuned hyperparameters and include relevant hyperparameter tuning plots (e.g., the cross-validation error plot).

(d) Assess the fit of the (tuned) models using at least two different evaluation metrics. Discuss why the metrics you've chosen are appropriate for your problem.

7. Identify your best model(s). Justify why you believe this is your best model(s).

8. For your best model(s), perform some post-hoc exploratory data analysis. Using visual and quantitative evidence, do you notice any patterns in the misclassification errors? Do you notice higher rates of misclassification in particular regions, or in specific ranges of feature values?

9. How well do you think your model would perform on other Terra satellite images that the model has not yet seen? Provide a quantitative measure for this generalization performance of your model development pipeline.

10. *For ACMS 60876 students only*: Suppose that the scientists want two things: (1) they want you to build a binary classifier, which labels each pixel as either cloud or not cloud, and (2) they want you to use all of the pixels (cloud, not cloud, *and* uncertain pixels) in the training process. Sketch an analysis plan which meets both requirements. You do not need to implement or carry out your plan. (Hint: ideas from *semi-supervised learning* may be helpful.)

As usual, please carefully document your analysis pipeline, justify the choices you make, and place your work within the domain context whenever possible.

Submission Details

Please push a folder named `lab3/` to your `dsip` GitHub repository **by 11:59 PM on Monday, March 16, 2026**. I will run an *automated* script that pulls from each of your GitHub repositories promptly at 12:00 AM and attempts to reproduce your report so please follow the file structure and names *exactly* with the exception that you may *add* folders as you wish.

The structure of your `lab3/` folder should follow the project structure discussed in class:

```
lab3/
```

```
├── data/
├── R/ (for R users)
├── python/ (for python users)
├── scripts/ (optional)
├── notebooks/
│   ├── lab3.qmd
│   └── lab3.html (or .pdf or other rendered output)
├── renv/ (for R users)
├── renv.lock (for R users)
├── .Rprofile (for R users)
├── environment.yml (for python users)
└── conda-lock.yml (for python users)
```

The `R/` and/or `python/` folders should contain all functions (and only functions, no scripts) necessary to reproduce your report. The `data/` folder should contain the raw data files, but **do not** push the `data/` folder to GitHub. In general, it is not good practice to store data on GitHub due to their restrictions on maximum file size (max: 100MB).

I will attempt to reproduce your report by running 'quarto render' so please be sure to include all necessary code in your repository. Keep in mind that my `data/` folder will only contain the raw data files that were initially provided to you.

A Note on Grading + Rubric

You will be graded on both the quality and reproducibility of your analysis.

A detailed rubric can be found on Canvas.

Please recall the course policy regarding collaboration: Collaboration *of ideas* with the instructor and with classmates is encouraged throughout this course, with the following caveats:

- You must write up the final code and text by yourself.
- If you collaborate or use any resources other than course texts, you must explicitly acknowledge your collaborators (e.g., in writing at the end of your report) and cite the resources you used.