DSC 385 - Project 3 Prediction Modeling

Overview

The goal of this project is to build and evaluate prediction models for predicting outdoor particulate matter air pollution concentrations using satellite imagery data. With growth of large satellite constellations being able to image the Earth on a near daily basis, there is the potential to determine air pollution concentrations (and hence, population exposures) anywhere in the world. However, our ability to do that rests on our ability to predict ground-level air pollution concentrations from the satellite imagery data. If a good model could be developed to predict ground-level concentrations, it might eliminate the need to make costly deployments of expensive monitoring equipment. Furthermore, areas with fewer resources dedicated to environmental monitoring would be able to obtain the same quality of information as other areas.

In order to train a prediction model, we have two sources of training data. The first is a time series of ground-level particulate matter (PM) concentration data taken at a single monitor located in Yuma, Arizona (USA). These data are measured in units of $\mu g/m^3$ and will serve as the outcome that we are trying to predict. The following shows the first few rows of the PM data recorded as the ground-level monitor.

```
# A tibble: 1,217 x 2
   date       pmFRM
   <date>     <dbl>
 1 2017-01-25  5.91
 2 2017-02-08  7.93
 3 2017-02-12  3.91
 4 2017-02-13  3.93
 5 2017-02-15  5.75
 6 2017-02-19  3.75
 7 2017-03-04  6.22
 8 2017-03-15 11.3
 9 2017-03-21  8.61
10 2017-03-23  6.91
# i 1,207 more rows
```

We also have a sequence of satellite images taken on the same days as the PM data were measured. The satellite images cover a 24x24 pixel area around the monitor location (you can think of the monitor as being in the center of the image). Each image is taken in four separate color bands—blue (`band1`), green (`band2`), red (`band3`), and near-infrared (`band4`). Because each of the color bands measures light at different wavelengths, each color band captures different features of the area being photographed and may contain different information about the air quality of that area. All of the images have been cropped to contain the exact same area of land, so while the images may appear to change over time, they are all taken in the same location.

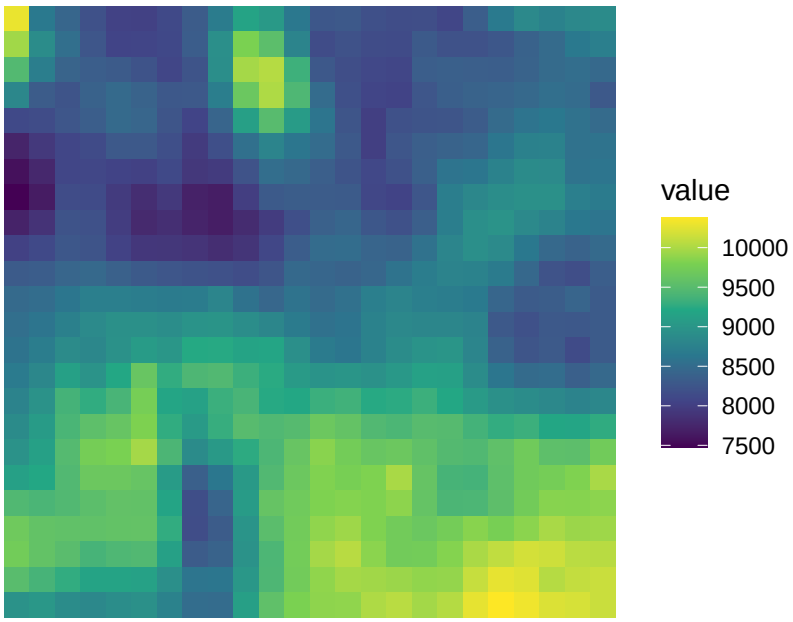
A single "image" is actually a 24x24x4 array consisting of a total of 2,304 numeric values. For this project, the data have been flattened so that they are in a data frame format and not a matrix format.

The following shows the first few rows of the satellite image data.

```
# A tibble: 2,803,968 x 4
   date       pixel    band  value
   <date>     <chr>    <chr> <dbl>
 1 2017-01-25 pixel001 band1  6185
 2 2017-01-25 pixel001 band2  5949
 3 2017-01-25 pixel001 band3  5631
 4 2017-01-25 pixel001 band4  4190
 5 2017-01-25 pixel002 band1  6280
 6 2017-01-25 pixel002 band2  6037
 7 2017-01-25 pixel002 band3  5683
 8 2017-01-25 pixel002 band4  4235
 9 2017-01-25 pixel003 band1  6330
10 2017-01-25 pixel003 band2  6087
# i 2,803,958 more rows
```

The numeric values in the `value` column are not exactly interpretable on their own, except to say that higher values represent more intensely reflected light (i.e. brighter) from the ground.

Below is an example of one of the images. This one was taken on March 23, 2017 and is in the blue color band (`band1`).



The basic idea of this project is to see if we can use characteristics of the images to predict ground-level PM from one day to the next. Because each image represents a large number of pixel values, we need the assistance of machine learning algorithms to determine if there is a predictive relationship there.

The general principle that we are trying to take advantage of here is particles in the atmosphere will tend to scatter light in all directions (including back up into space where the satellite is). So if there are more particles in the air, light will be scattered more; whereas if the area is totally clear, then light will reach the ground and then be reflected back. So at the simplest level, if an image is "blurry", then it is likely there is a high level of PM air pollution. But if the image is "clear", then there is likely a low level of PM air pollution. However, there isn't really a strict definition of what is a blurry or clear image, and furthermore, there are other factors that can make an image blurry or clear (such as clouds) and we don't have data on those factors.

Lecture Note

The data for the project are similar to the data discussed in Lecture 9.3 "Introducdtion to Tidymodels, Part 2" and Lecture 9.5 "Tidymodels Demo Part 2(a)" so you may want to review those lectures before starting this project. The dataset discussed in those lectures is bigger than the one being used here. For this project, I have reduced the size of the dataset to make it more manageable.

Getting the Data

The data can be obtained from the following GitHub repository:

`https://github.com/Principles-of-Data-Science/Project3`

You can pull the GitHub repository into RStudio as follows:

1. Click on File > New Project…
2. Click on "Version Control"
3. Click on "Git"
4. Paste in the URL **https://github.com/Principles-of-Data-Science/Project3** into the "Repository URL" field
5. Type in the "Project directory name" if needed
6. Set the directory if you don't want to use the default
7. Click "Create Project"

The Report

The text of your report will provide a narrative structure around your code and outputs with R Quarto. Answers without supporting code will not receive credit and outputs without comments will not receive credit either. Write full sentences to describe your findings. All code contained in your final project document must work correctly (render early, render often)!

The report template provided in the GitHub repository contains prompts/questions that you will need to answer. Please follow the prompts in the template and answer all of the questions there.

Formatting

- Create the report using R Quarto knitted to a PDF file, with headers for each section and each question answered;
- Include comments describing your R code;
- Include any references (datasets, context), if needed.
- Please do not print out very large objects that require multiple pages; only print out what is needed to explain your reasoning for a question.

- It is extremely important that you **select pages** when submitting on Gradescope (see more below). Points will be taken off if you do not select the appropriate pages for each question in the Gradescope outline.

Submission of the Report to Gradescope

This project report will be submitted on Gradescope for grading. Gradescope is a tool that enables the teaching staff to efficiently grade assignments like this one according to a defined rubric. **You will not be submitting this project on Canvas**. Anything submitted to Canvas will be ignored.

If you have never submitted anything to the Gradescope web site, please watch this video demonstration of how to do so.

To submit your project report, please follow these steps:

1. First render your project report into a PDF file. This can be done by either rendering directly to PDF in RStudio or by rendering to an HTML file and then "Printing" to a PDF file. Either way, **you must have a PDF file to submit to Gradescope**.
2. Go to the course Canvas page and click on the "Gradescope" link on the navigation bar on the left hand side.
3. When the Gradescope page loads, click on the assignment titled "Project 3: Prediction Modeling".
4. You should be prompted with a window allowing you to submit a PDF file of your assignment.
5. After uploading your PDF file, you will be prompted to select pages of your PDF file that correspond to questions in the "Question Outline". Please make sure to do this carefully, as it essential for allowing us to grade your project efficiently.
6. After selecting the pages, submit the assignment.

Late Policy

As per the Syllabus, projects will not be accepted late. There are no exceptions; please do not contact the instructor or TA to request an exception.