STAT 215A, Lab 4: Predicting Cloud Cover in the Arctic

Mark Oussoren, Sahil Saxena, Hyunsuk Kim, Florica Constantine

11/12/2021

1 Introduction

In this report, we develop classifiers to detect the presence of clouds versus ice in arctic imagery. This problem is motivated by the study of global warming: clouds act as a natural barrier against increasing surface temperatures and hence the melting of ice in the arctic (Shi et al. 2007). Our goal in what follows is to develop, evaluate, and compare several classifiers or models for predicting cloud cover.

2 Exploratory Data Analysis

2.1 Data

Our data set consists of an ensemble of three images taken from the Multi-angle Imaging SpectroRadiometer (MISR) over the arctic, northern Greenland, and Baffin Bay. Within each of these images, there are roughly 115,000 pixels at 275-m resolution per image. Of these 115,000 pixels, approximately 70,000 are definitively labeled. From (Shi et al. 2007), we have that "Expert Labels are only given to those pixels of which the expert is highly confident based on his knowledge."

The data set includes the following features:

- label: Expert label of pixel - either +1 (cloud), 0 (unsure/unlabeled), and -1 (no cloud).
- y_coord: y coordinate of the image - an integer between 2 and 383. From (Shi et al. 2007), $i \in \{1, \ldots, 384\}$.
- x_coord: x coordinate of the image - an integer between 65 and 369. From (Shi et al. 2007), $j \in \{1, \ldots, 512\}$.
- NDAI: normalized difference angular index - computed as the difference of average radiation levels at angles DF (60° Zenith) and AN (0° Zenith). The idea here is that surface-leaving radiation is more isotropic from non-cloud surfaces than low altitude clouds. This serves as a good classifier, as it means radiation disperses more evenly to all angles from non-cloud surfaces, so that an increase in NDAI is a proxy for more uneven radiation levels and the presence of clouds.
- SD: Standard deviation of the 64 radiation measurements for the 5th angle associated with location (i, j). Clouds are generally not smooth, so the rationale for this feature is that larger standard deviations are more indicative of the presence of clouds.
- CORR: Average linear correlation of radiation measurements at different view angles - valued in $[-1, 1]$. Higher CORR implies low or no clouds, while low correlation is generally associated with high clouds. This feature, if used, has to be coupled with several of the other features in order to classify clouds from non-cloud pixels.
- DF, CF, BF, AF, and AN: Radiance angle of the camera, valued in $[70.5°, 60.0°, 45.6°, 26.1°, 0.0°]$, respectively.

2.2 NDAI

The Normalized Differential Angular Index has the functional form

$$NDAI_{ij} = \frac{\overline{I^1_{i,j}} - \overline{I^2_{i,j}}}{\overline{I^1_{i,j}} + \overline{I^2_{i,j}}},$$

where $\overline{I^1_{i,j}}$ is the average radiance sampled from the spectroradiometer in the Df orientation and $\overline{I^2_{i,j}}$ is the average radiance sampled in the An orientation. Both are measures of radiance and thus are positively valued. Thus, the NDAI ratio should be bounded from -1 to 1. Empirically however, we see that the histogram in Figure 1 has a large bulk of the data outside the expected range.

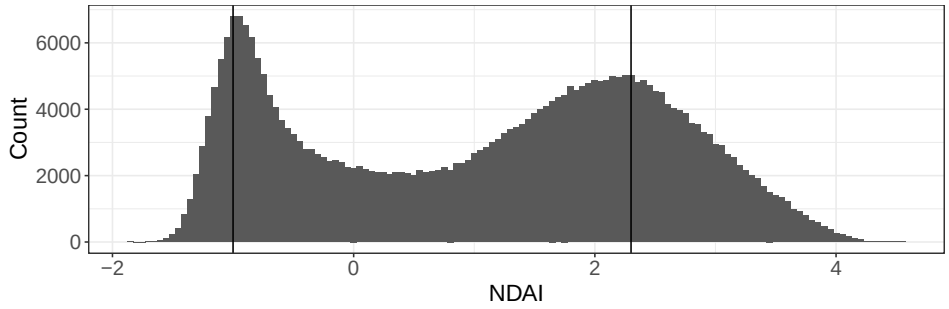

Figure 1: Distribution of NDAI Values

The data outside the -1 to 1 range constitutes a vast majority of the pixels that were labeled by experts as clouds, so it is infeasible to throw out these data points. It is worth noting that this is also how the authors in (Shi et al. 2007) proceeded with their analysis.

2.3 Spatial Distributions of Features

In Figure 2, we look at the expert labels for the data.

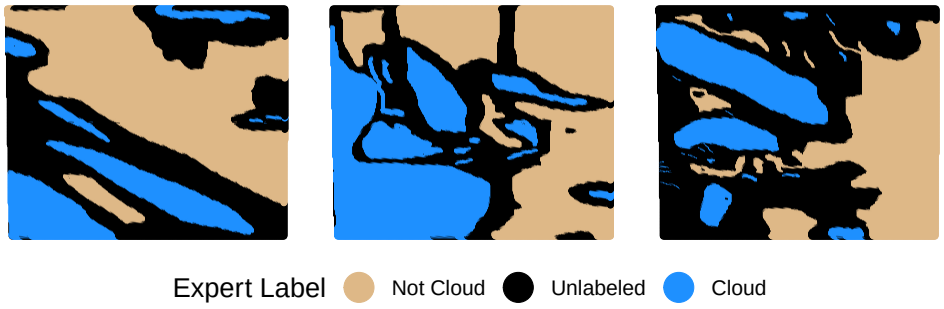

Figure 2: Expert Labeled MISR Data

Next, in Figure 3, we see that low NDAI visually correlates with non-cloud regions, and higher NDAI visually correlates with cloud or unlabeled regions. Intuitively, note that light in the visible spectrum will not be scattered preferentially on incidence with the surface of the earth (especially on a white background, like the arctic images in our dataset), which would lead to an NDAI of around 0. While we cannot be certain this fits with our data given the suspicious units, it may be off by a scaling factor.

In contrast to the light scattered on the ground, incident light on the clouds in the visible spectrum does not scatter isotropically. This leads to a much greater difference in DF and AN and hence in our $\overline{I^1_{i,j}}$ and $\overline{I^2_{i,j}}$ terms, resulting in a larger NDAI. Moreover, this difference is why the SD column is a useful metric, as it represents σ_{An}, which is described in the paper as a quantification of the ground smoothness. This is why lower confidence pixels, i.e., those with a higher SD, appear to track the outline of the cloud segments or show

the clouds themselves. For the CORR feature, we can see that high values of CORR are generally associated with cloud-free regions (but occasionally, as mentioned in (Shi et al. 2007), could also be associated with a low altitude cloud).

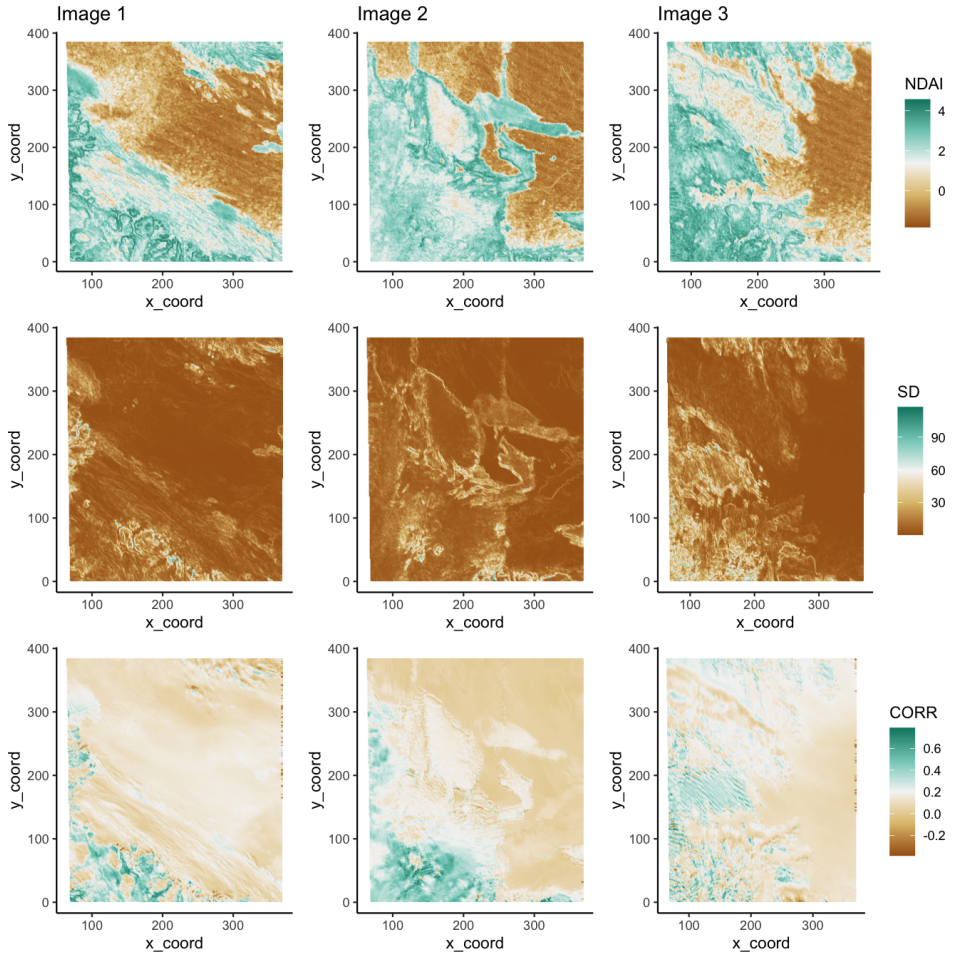

Figure 3: MISR Data colored by NDAI, SD, and CORR respectively

2.4 Kernel Density Plots and Boxplots of Features

Figures 4 and 5 compare the distribution of the features proposed in the paper for the different classes of points (cloud, unlabeled, not cloud). Notably, we see what look like a large number of outliers and that NDAI appears to have the most visible separation of clusters.

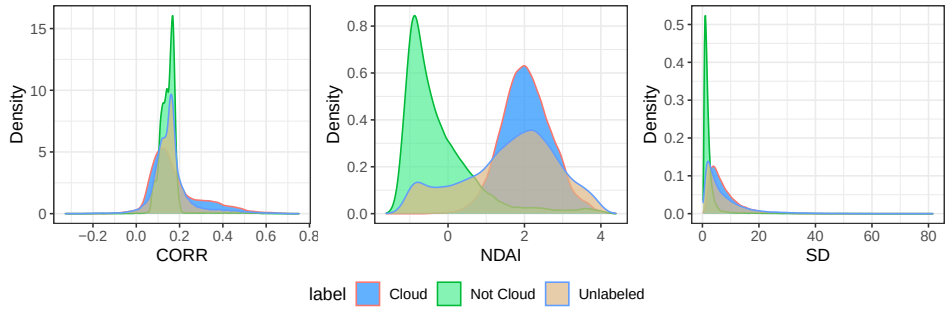

Figure 4: Kernel Density Plots of NDAI, SD, and CORR

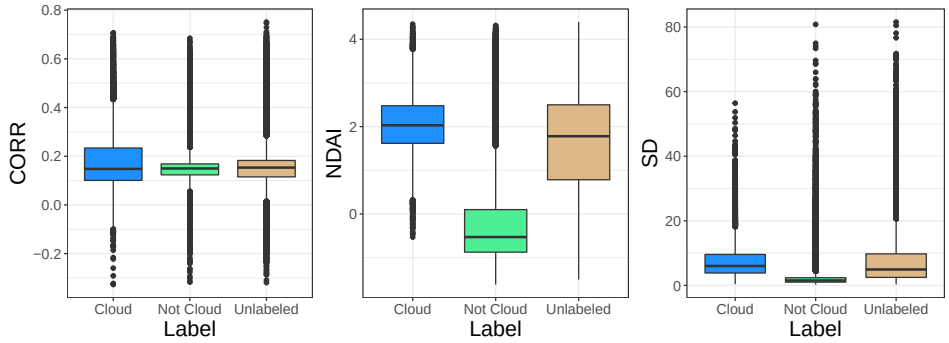

Figure 5: Box Plots of NDAI, SD, and CORR

2.5 Contour plots of Features

In Figure 6 we compared the entropy between the distributions and see which features maximize the entropy between the cloud and no cloud distributions. We see that NDAI maximizes the entropy. For the remaining features, we consider a multivariate setting and see which lead to separate clusters, along with the NDAI. We see that the other features do not lead to great separation.

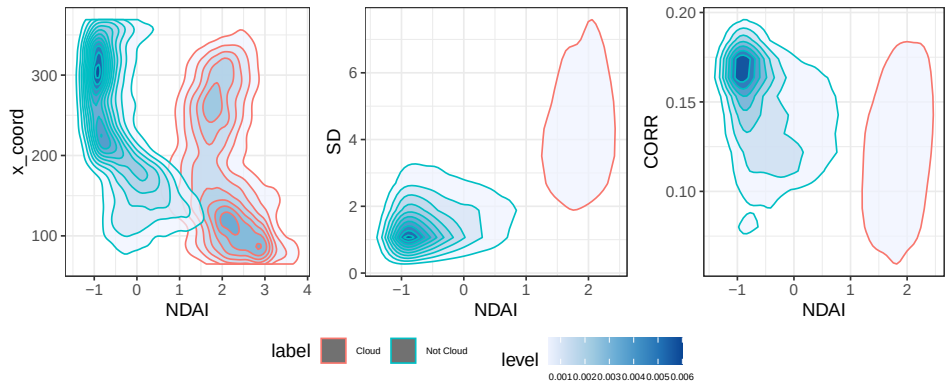

Figure 6: 2D Contour Plots with NDAI

2.6 Radiances

In Figure 7, we present the class-conditional densities for each radiance feature from Image 1 (the other images have similar results). We see that there is a lot of overlap between the classes, so we expect the radiances to not be particularly predictive on their own.

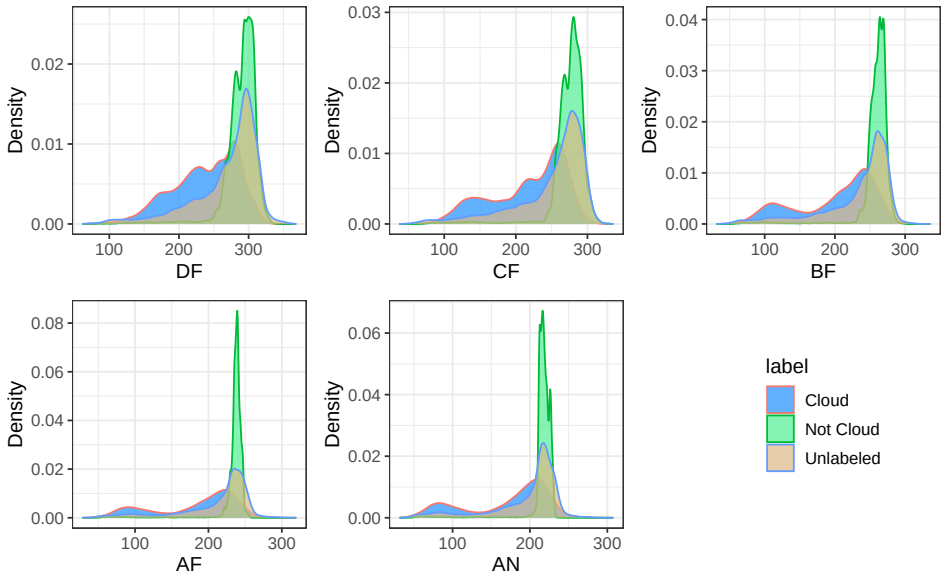

Figure 7: Kernel Density Plots for Radiances

2.6.1 Relationship between radiances

In Figure 8, we present the Spearman correlation between each of the radiances. We see that adjacent pairs (also physically adjacent in terms of angles) are highly correlated, leading to colinearity in the data that can affect models used for prediction.

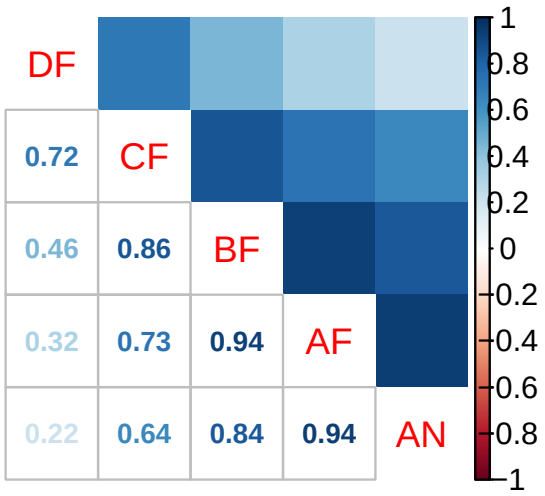

Figure 8: Correlation between radiances

3 Classifiers

We next developed prediction models to try and determine the "Cloud" from "Not Cloud" pixels. In order to do this, we first removed the "Unlabeled" pixels from all three images. Next, we needed to decide on how to split up the data into a training data set and a testing data set. This was done in two ways.

1. For our training set, we randomly selected two-thirds of the pixels across all three images for training and saved the remaining one-third of the pixels for testing. We ensured that the ratio of class labels in both the training and testing data set were the same as those in the original data. The rationale

behind this overall approach is to potentially capture more variability in the data, as all three images were taken from different locations. This could be the cause of changes in visualizations of the sky against the ground, and may allow us to make better predictions for any future data sets. All model visualizations will be shown on this data split.

2. For our training set we use image 1 and image 2 and test on image 3. As the distributions of the first and second images do not quite align with that of the third, this method will give us a better sense of a model's robustness to data that is different from what has previously been seen.

The classifiers we chose to develop on the data were k-Nearest Neighbors (kNN), a Random Forest (RF), Quadratic Discriminant Analysis (QDA), and ℓ_2-penalized Logistic Regression. For the models that needed parameter tuning, we performed 10-fold cross-validation on the training set to select the parameter that minimized the mean cross validation error. After selection of the parameter, we ran the model on the full training data set and tested it on our testing data set.

When selecting features from the data to include in our model we first start with a small number of predictors based off our EDA to aid in classification—in particular NDAI, SD, and CORR. These features are all functions of the radiances (the $1st$ and $5th$ Zenith angles) that were shown to capture particular information about the clouds and their relationship to the landscape in (Shi et al. 2007) and also in our EDA section. We also fit models using NDAI, SD, CORR, and all 5 radiance measures as our data is not high dimensional and this would enable us to see if any of the radiances themselves will aid in the classification task, possibly when combined with other features. From our EDA section, we suspect any improvement in performance not to be dramatic one. We also choose to exclude the x- and y-coordinates from our models as they denote the locations of the pixels in the image which is not a feature that arises from the real physical landscape and clouds of interest.

In order to determine how well each individual model for a classifier performed, we calculate the AUC along with the classification accuracy on the test set.

3.1 k-Nearest Neighbors (kNN)

A benefit of the k-Nearest Neighbors algorithm is it is a nonparametric approach and so no assumptions about the data have to be made explicit a priori. Additionally, kNN assumes that similar things exist close to one another, which is the relationship seen in our data as cloud pixels are more likely to be closer to other cloud pixels and vice versa. The main setback of this approach, however, is finding the optimal choice of k. While somewhat arbitrary, we can tune and select k through careful cross validation. We selected $k = 41$ for our first model using three features (NDAI, CORR, and SD) and $k = 3$ for our model using the same three features plus the five radiances.

The left plot in Figure 9, shows the ROC curves for using three features versus using all features in our kNN - both reveal very strong capability of classifying clouds from no clouds. The right plot in Figure 9, shows that the convex hull of cloud and no cloud predictions appear to be relatively distinct (there is some overlap for high SD and NDAI).

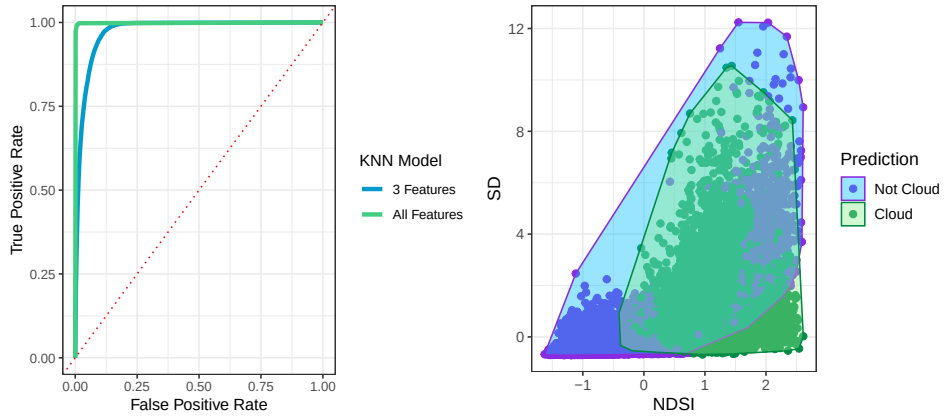

Figure 9: ROC Curve and Convex Hull of Predictions

Additionally, we can look at traces of the decision boundary for our classifier. As seen below, for the most part the model generalizes well, but interpretability of the boundary and results is not straightforward. We can see that the model has no difficulty with the extreme NDAI values (below -1 and above 1), but has some trouble for points with NDAI between 0 and 1.

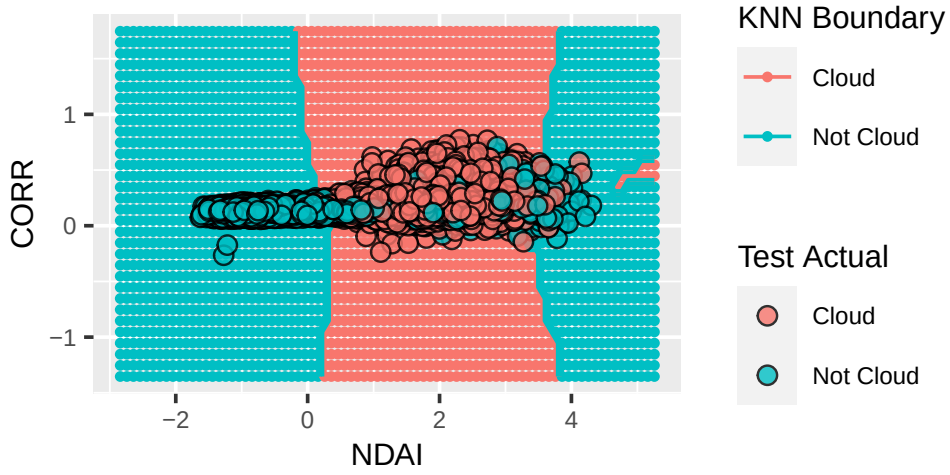

Figure 10: Decision Boundary for kNN

3.2 Quadratic Discriminant Analysis (QDA)

QDA (Quadratic Discriminant Analysis) models each class density as a multivariate Gaussian. As stated in (Shi et al. 2007), compared to other classificiation methods, QDA is more robust to errrors in the training labels because it models the joint distribution $P(s, x)$ (as opposed to other models like ℓ_2-penalized Logistic regression which only models the conditional probability $P(s|x)$, where s denotes the class and x the sample). An additional benefit to QDA is it is computationally efficient as it involves only estimating the vector of means and covariance matrix. We decided to use QDA over another related classifier such as Linear Discriminant Analysis, as it offers more flexibility in the boundaries between classes which must be linear in LDA. It also allows the covariances of matrix associated to each class to be different to one another, as opposed to LDA where they must be the same. Referring back to Figure 4 and Figure 7, we can see that the class distributions have different variances, making QDA a more natural choice. Next, we make QQ plots in Figure 11 to check how well the normality assumption holds. We can see that the "Cloud" pixels in general look to follow a normal distribution, except sometimes at the tails. However, the "Not Cloud" pixels do not follow a normal distribution at all.

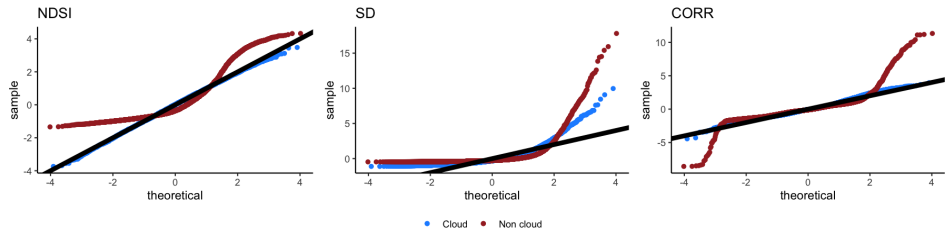

Figure 11: Normal QQ plots for the NDAI, CORR, SD features in the training set.

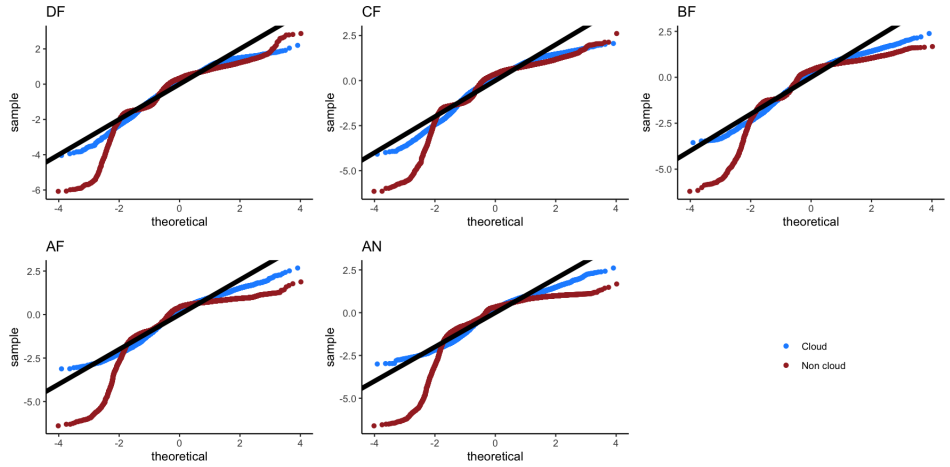

Figure 12: Normal QQ plots for the radiance features in the training set.

In Figure 13, we see an ROC plot and a plot of the confusion matrix, showing the classification results of the QDA model trained on the NDAI, CORR, SD.

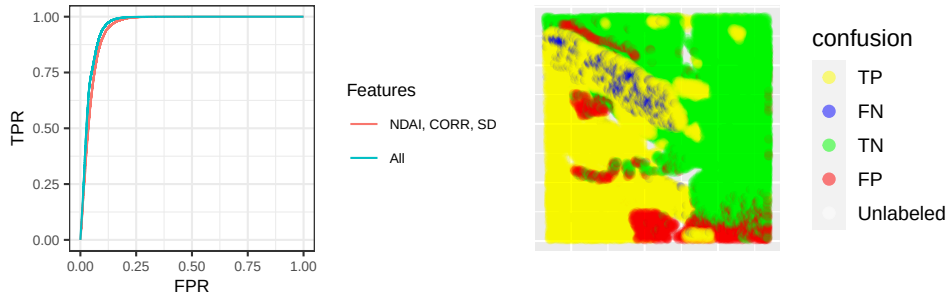

Figure 13: ROC Curve and Plot of Confusion Matrix

Points were either classified or misclassified based on their likelihood ratio with a threshold of 0.129. For a false positive rate (points classified as cloud, but actually non-cloud) of 8.17%, we see a false negative rate (points classified as non-cloud, but actually cloud) of 1.55%.

3.3 Random Forest

When it comes to building a model in order to predict a binary outcome, we can think of a decision boundary by which we separate our prediction. This can be generalized into a decision tree with several nodes at the bottom. Moreover, we can think of an ensemble of decision trees to obtain more stable results. In this regard, it is natural to think of Random Forest (RF), a non-linear statistical model which can be seen as an ensemble of decision trees. For each decision tree, RF bootstraps the data and generates a new data set for the tree. Then, RF randomly chooses a feature to be used as the classification criterion, and computes a threshold

value that best separates the data. In this paper, two hyperparameters were cross-validated: the number of variables to use in classification (mtry), and minimum size of a terminal node (nodesize). As we have eight features, cross-validation was conducted on 1 to 8 for mtry, and on 1, 10, 100, and 1000 for nodesize. The *randomForest* package in *R* was used to cross-validate and eventually build the model.

Accuracy was easy to calculate as a random forest gives exactly 1 or -1 as a predicted value. We can also compute an AUC through looking at the proportion of votes given to a pixel for being a cloud or not a cloud. In total, 32 cross-validated models showed similar accuracy. Among them, the model with 4 as mtry and 10 as nodesize showed the best average accuracy. Additionally, importance analysis showed that variables with high mean decrease accuracy are NDAI (100.5), SD (32.8), CORR (31.57). However, when the random forest was trained by image 1 and image 2 then tested on image 3, it showed 0.641 accuracy and 0.536 AUC. This may be attributable to the distinct distribution of data in image 3 and possible overfitting of the random forest model. In Figures 14 and 15, we see the misclassified points in each of the two different data splits.

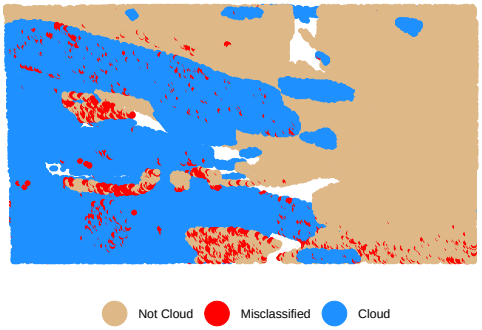

Figure 14: 2/3 Split Missclassified Points

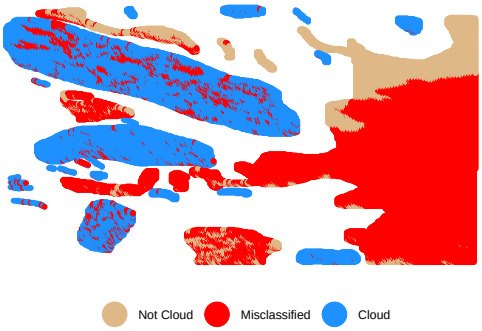

Figure 15: Image Split Missclassified Points

3.4 L2-penalized logistic regression

An ℓ_2-penalized logistic regression model is a penalized generalized linear model wherein the log of the odds ratio is a linear function of the features. That is, the log of the odds ratio is a continuous response that we fit with ordinary least squares. The ℓ_2 penalty acts as a regularizer and acts to improve the conditioning of the moment matrix: in this setting, we know that some of our features are functions of others, so we expect a high degree of colinearity. Using a penalized model will produce more stable solutions than not. This model assumes that the samples are independent; since our data are spatial, they clearly are not. We expect that neighboring pixels are not independent (we may reason that knowing that a pixel is part of a cloud increases the likelihood that its neighbors are also part of a cloud). Moreover, this model assumes that the log of the odds ratio is a linear function of the features We have no way to know a priori whether the odds ratio behaves as assumed and our features exhibit high degrees of colinearity; as we are only interested in

prediction and not inference, we will not interpret the coefficients of our logistic regression or how individual features affect the response. In Figure 16, on the left we see the ROC curves associated to each model. Each have extremely similar performance as the curves are almost entirely overlapping. In Figure 16, on the right we see that a low λ value corresponds to the best cross-validated accuracy. This means that we need very little regularization in our model.

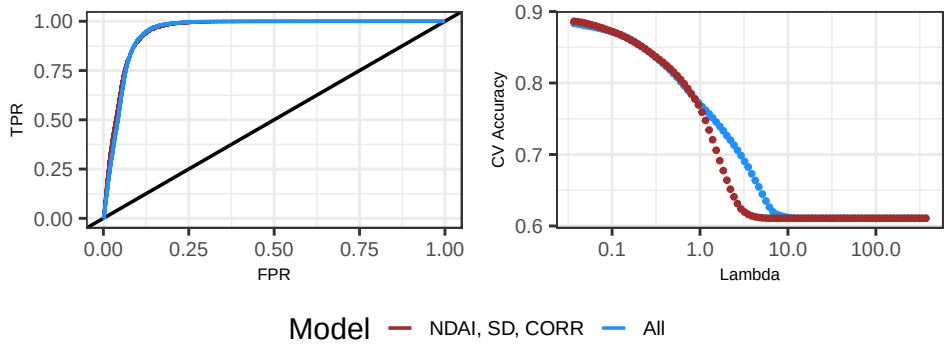

Figure 16: ROC Curve and CV Accuracy v. Lambda

3.5 Final Model Selection

The following tables summarize the performance of 4 classifiers on the test set, which was separated from the training set before classifiers were developed.

Results for training on 2/3rds of the data randomly chosen and testing on the remaining 1/3 of data.

Table 1: AUC and accuracy of 4 classifiers when only 3 variables (NDAI, CORR, SD) were used.

	KNN	QDA	RF	Ridge
AUC	0.974	0.954	0.971	0.950
Accuracy	0.921	0.906	0.917	0.904

Table 2: AUC and accuracy of 4 classifiers when all variables except x, y coordinates were used.

	KNN	QDA	RF	Ridge
AUC	0.998	0.963	0.993	0.953
Accuracy	0.993	0.917	0.960	0.901

Results for training on images 1 and 2 and testing on image 3.

Table 3: AUC and accuracy of 4 classifiers when only 3 variables (NDAI, CORR, SD) were used.

	KNN	QDA	RF	Ridge
AUC	0.916	0.888	0.877	0.899
Accuracy	0.853	0.822	0.819	0.842

Table 4: AUC and accuracy of 4 classifiers when all variables except x, y coordinates were used.

	KNN	QDA	RF	Ridge
AUC	0.829	0.907	0.536	0.898
Accuracy	0.814	0.827	0.641	0.764

From the 4 classifiers above, we choose the QDA classifier as our final classifier. Note that kNN had the best overall performance, but was too computationally intensive for us to run as our classifier when performing stability analysis in the limited time frame we had. The random forest model produced vastly different results depending on the training/test data split chosen and as such was unstable. The ℓ_2-regularized regression performance metric point estimates were slightly lower the QDA but likely not significantly so. However, QDA seemed a more natural choice of model for our data given our reasoning in above sections and is much more computationally efficient and is parameter free (besides a threshold).

4 Post-Hoc EDA

4.1 Classification

Now, we restore and include unlabeled data points to see whether our final model performs continuously on data points. Figure 17 shows that our final model predicts unlabeled data points continuously according to the expert labels. This shows that unlabeled points alongside clouds tend to be classified as clouds and those alongside not-cloud regions tend to be classified as not cloud, as one would hope to see. This suggests that our classifier also performs in a sensible way when classifying unlabeled points, as we do not see discontinuous predictions, where in a narrow unlabeled region between cloud and not cloud we predict alternating sequences of (cloud, not cloud, cloud, not cloud, etc.).

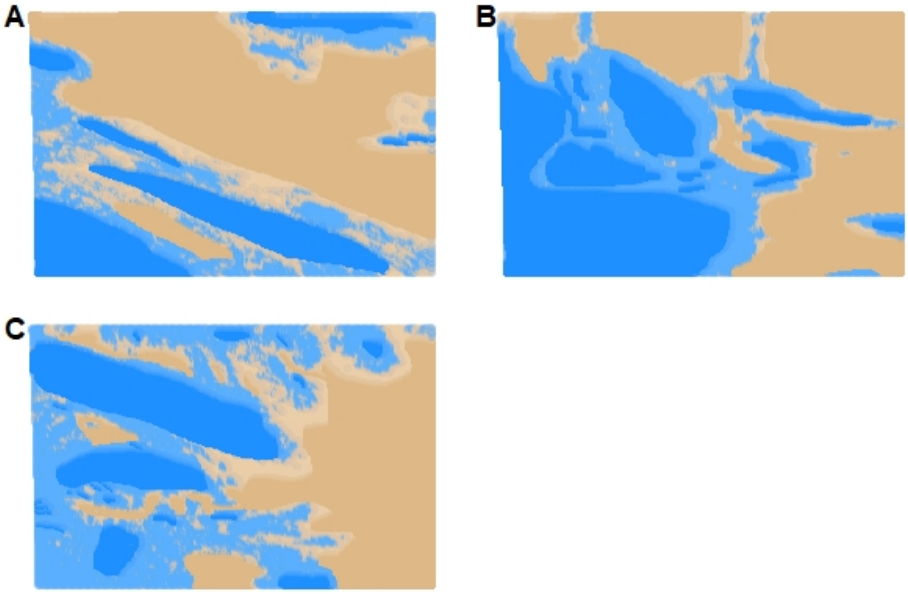

Figure 17: Prediction of QDA final model for unlabeled data for (A) image1 (B) image2 (C) image3. Expert labels are shown with vivid color, and predicted labels for unlabeled data are shown with faded color. Blue and brown indicates cloud and not cloud, respectively.

4.2 Stability Analysis

Next, to ensure our final QDA model is stable, we performed a stability analysis with perturbed data. Since our model development is based on a random split of train set and test set, we repeated the same process with 400 random splits. From the experiment, we find the 95% bootstrapped confidence intervals of the AUC and accuracy on the test set are (0.9676, 0.9701) and (0.8767, 0.8810), respectively. These intervals are narrow, and lead us to believe that we can expect similar values of the AUC and accuracy when we apply our final model on future data.

5 Conclusions

We have analyzed image data and developed classifiers that predict whether a pixel lies within a cloud or outside. We developed and compared the performance of several methods (penalized regression, random forest, k-NN, and QDA) before choosing QDA as our final model due to its high AUC and stability in performance across the two different ways we trained and tested models. We have tested our final classifier and found that it is stable: rerunning on various subsets of the data produces similar results. Moreover, we have found that our final model performs extremely well, with a high AUC and accuracy. Hence, we anticipate that our results and model will be useful for analyzing future data, especially as the confidence intervals are narrow.

5.1 Division of Labor

Mark did the exploratory data analysis, implemented a k-NN classifier, and offered numerous figure edits. Sahil did the exploratory data analysis and implemented the QDA classifier. Hyunsuk implemented the random forest classifier, performed post-hoc analysis, and added in tables of model performances to the paper. Florica coordinated the project, edited others results as they came in, implemented the ℓ_2-penalized logistic regression classifier, did numerous figure edits, and edited and wrote much of the final report.

References

Shi, Tao, Eugene E Clothiaux, Bin Yu, Amy J Braverman, and David N Groff. 2007. "Detection of Daytime Arctic Clouds Using MISR and MODIS Data." *Remote Sensing of Environment* 107 (1-2): 172–84.