Every Cloud has a Clear Lining

Team DB (DataBase): Bailey Joseph (26035452) and Deborah Chang (3033221648)

Overview

Climate change has always been a topic of interest for scientists and for the general public, especially regarding negative effects that may become health concerns and environmental dangers for nature. The study, "Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies" by Tao Shi, Bin Yu, Eugene E. Clothiaux, and Amy J. Braverman, focuses on identifying cloud or clear properties that will help enhance more understanding of the response of clouds to atmospheric carbon dioxide levels. In particular, the Arctic is where carbon dioxide levels will affect temperatures. The goal is to more clearly identify cloud properties in the Arctic, as current methods may not distinctly identify clear and cloudy areas. Examples of issues include the similar scattering properties of clouds and surfaces. MISR has been the current measurement method and has much data to process. With radiance measurements, there seems to be a better potential to differentiate from snow-and ice-covered surfaces. As mentioned, MISR collects data from the same path every 16 days (Shi 2).

Other types of methods are not possible to consider, as they require experts and domain knowledge. Considering time constraints and computational complexities, a new cloud detection algorithm was proposed, that can analyze the MISR data. The algorithm is called enhanced linear correlation matching (ELCM) algorithm. Utilizing that in addition to Fisher's QDA, this method produces a more desirable prediction of cloudy or clear areas.

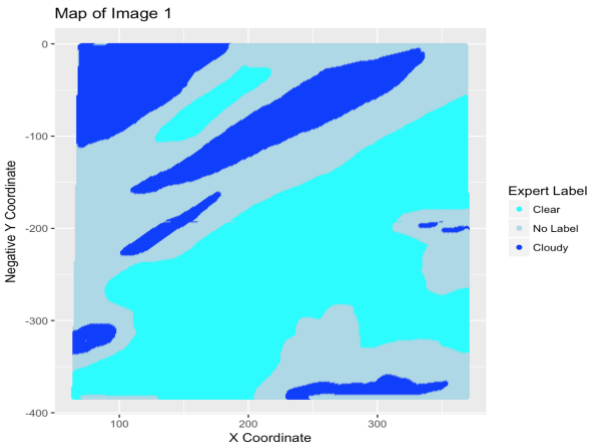
The data is massive, containing orbits from the 26th path through the Arctic. The 10 orbits range about 144 days from April to September (Shi 3). 57 data units were considered – three were excluded to preserve unbiasedness. The method utilized was to identify and build three features and run the proposed ELCM algorithm on the data. To test performance, expert labels were used.

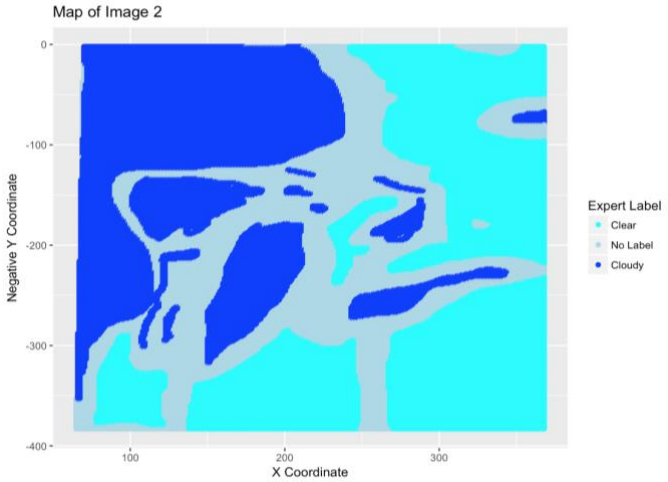
The conclusion derived compared the performance of the proposed algorithm with that of MISR and various others. ELCM performed well, taking into account adaptive thresholding to compare separability. ELCM performed better than ELCM-QDA based on expert labels. The hope is that the ELCM algorithm, which includes statistical thinking application and collaborative atmosphere, climate change models can be more accurate.

Summary of the Data

For each image, we calculate the proportion of pixels for each class label. Image 1 contains 44% of pixels that are clear, 39% that are unlabeled, and 17% that are cloudy. For Image 2, 37% of pixels are clear, 29% are unlabeled, and 34% are cloudy. Image 3 contains 29% of pixels that are clear, 52% that are unlabeled, and 18% that are cloudy. When combined as a whole, 36% of pixels are clear, 40% are unlabeled, and 23% are cloudy.

Maps





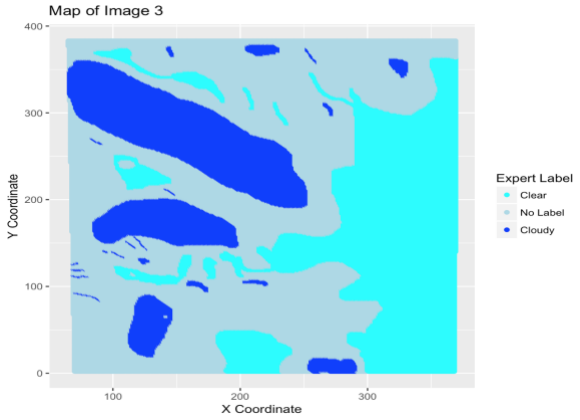
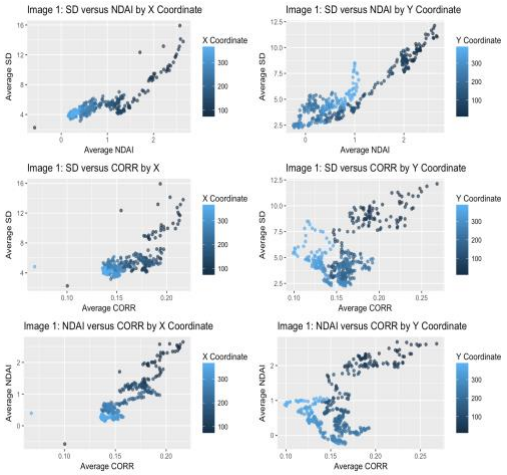


Image 1 contains cloudiness in the center, while Image 2 seems to contain more cloudiness overall. There seem to be more cloudiness on the left side of the images, whereas clear areas are more on the right sides.

An independent and identically distributed assumption may not hold here, as each pixel is not independent of each other - an adjacent pixel of a pixel that is labeled as cloud may be more likely to be cloudy. Pixels close to each other may be dependent. Groups of pixels may be our data units rather than single pixels.

Exploratory Data Analysis

Image 1: Averages Grouped by X and Y Coordinates



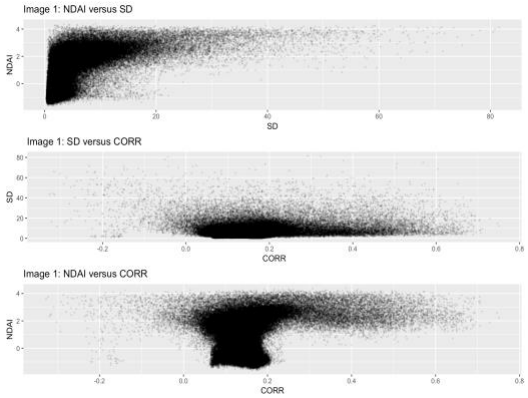


Image 2: Averages Grouped by X and Y Coordinates

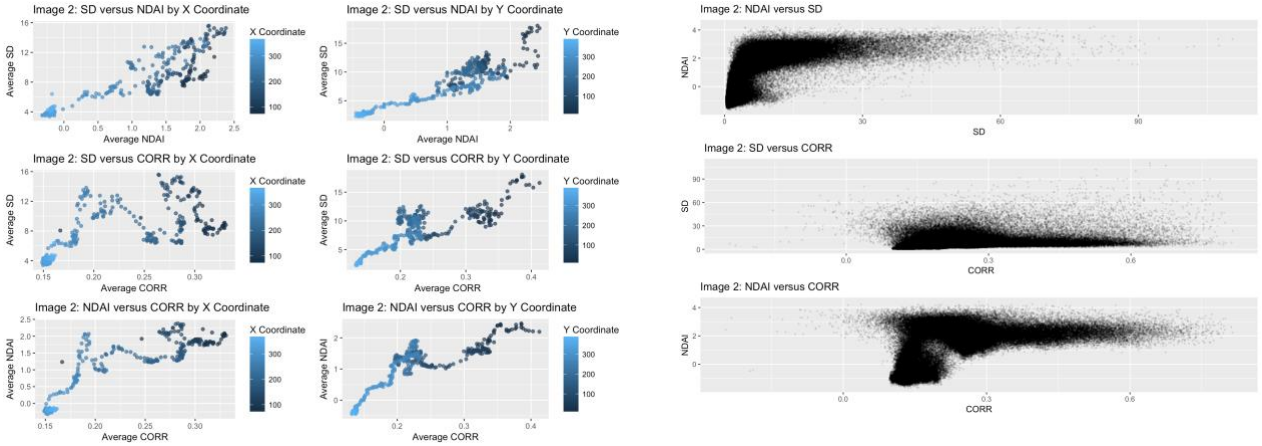
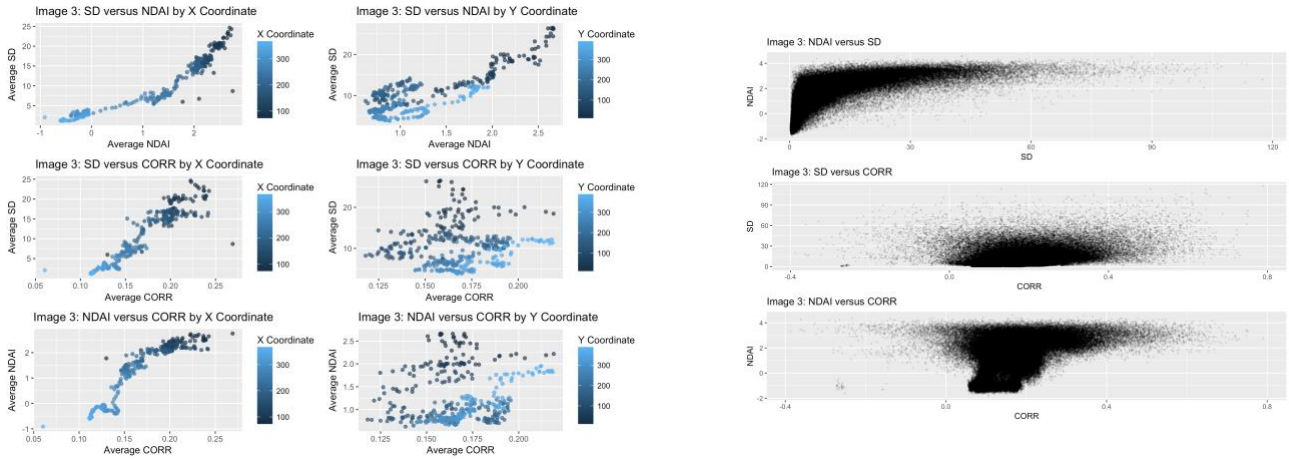


Image 3: Averages Grouped by X and Y Coordinates



Expert Labels and Features

Average Feature Measurements by Expert Labels

Image	Expert.Label	NDAI	SD	CORR
1	-1	-0.25	2.51	0.15
1	0	1.6	7.65	0.16
1	1	2.05	7.5	0.18
2	-1	-0.35	3.14	0.15
2	0	1.97	12.68	0.21
2	1	2	10.61	0.34
3	-1	-0.18	3.48	0.12
3	0	1.9	14.18	0.18
3	1	1.76	10.69	0.2

As shown above, pixels marked by experts as cloudy have lower y coordinates, with the exception of Image 3. The x-coordinates are the lowest for cloudy areas. Regarding features themselves, NDAI is negative for

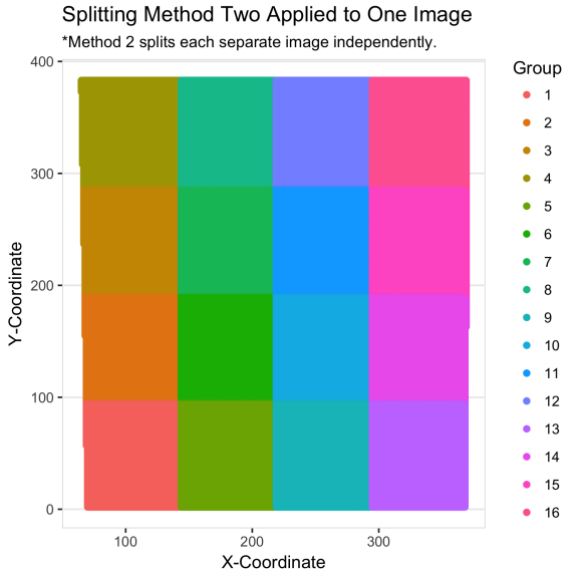
clear areas and SD increases in general for cloudier areas. CORR has smaller absolute differences, but also a smaller scale. As we will show in the next section, it separates the classes fairly well. Radiance measurements overall are lower for cloudier--marked pixels.

Preparation

In order to fairly train and evaluate cloud detection models, the data needs to be split into three groups -- a training set, a validation set, and a testing set. The traditional method of randomly assigning rows to each of the groups will not work in for this data because as we saw in section 1 part b, nearby pixels are highly dependent. When future data comes in, we will always get a whole new image with no expert labels. For this reason, the way we split the data needs to maintain this space dependence, and we suggest two different methods such to ensure that pixels in different sets are not close to each other spatially.

Our first method of splitting the data will be very harsh -- we will simply use one image for training, one image for validation, and one image for testing. We feel that it's important to try this to make sure that the model doesn't overfit -- in this method we will always be predicting on data that is (almost completely) independent from the training data. We will need to introduce slightly more complexity to this method when using it for cross validation, because we'll need to get K equally sized subsets of only 2 images (the training and validation images). To do this, we put the images together completely and then use a K-means algorithm to partition into groups based on the x and y coordinates of the pixels. Each CV holdout set will come from a different spatial region.

We are, however, worried that this method will be too restrictive. The accuracies will be bad if the image chosen for training turned out to be significantly different from the others. Our second method will leverage the fact that we do have three whole images with expert labels. We will divide each image into 16 squares (a grid of 4 rows and 4 columns) and then form our sets by taking a cluster sample of the resulting 48 squares. This number 16 is a somewhat arbitrary choice, but we cannot in good faith use cross validation to tune it. Because of the dependence, it should be the case that more squares leads to better CV/test performance, but this may not be an unbiased estimate of future data that comes with no



labels. We feel that 16 is a good balance because it is subjectively the case that most pixels in the training set will not be close to any in the testing set.

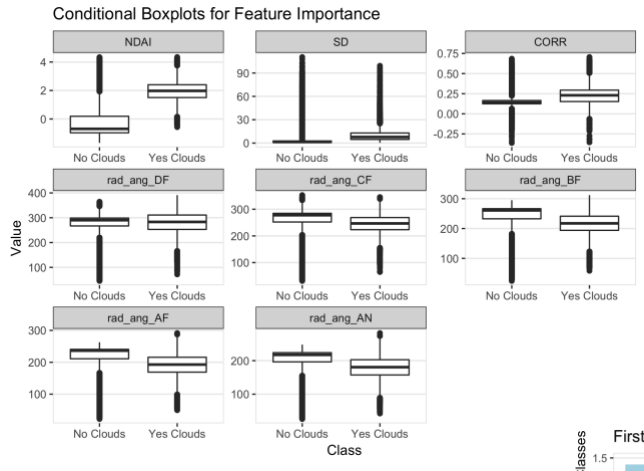
Next, to help us understand the classification problem and ensure there's some kind of class balance, we investigated a trivial classifier that always predicts "not cloudy." We did this on the validation and testing sets under both of our splitting strategies to mirror the way that we will ultimately be evaluating the models. The results are in the table below:

Accuracy of a Trivial Classifier

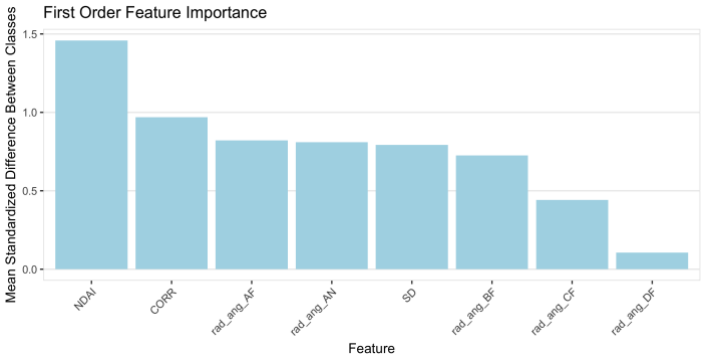
Splitting Method	Type	Baseline Accuracy
1	Test	0.522
1	Validation	0.711
2	Test	0.519
2	Validation	0.687

The overall average baseline accuracy is almost exactly 60%. The trivial classifier will have high average accuracy when the classes to be predicted are not balanced. That is, if we were looking to detect something very rare, we'd get a high accuracy if we never predict its existence. This doesn't mean that the classifier is good, especially if it is important to catch all occurrences.

Before we started modeling, we investigated features in hopes of finding three that seem particularly predictive. First, we looked at conditional boxplots of each predictor for the training set.



While we saw separation between the classes in every variable, we found there to be too many outliers to confidently select only 3 features. To select the final three features, we computed the standard deviation of each of the predictors (combining both classes) and the difference in the means of each predictor in each class. The three features that had the largest standardized difference were NDAI, CORR, and Radiance Angle AF.



Finally, we created a function to streamline our model evaluation through cross validation. We've included all necessary code for this on our GitHub so that the reader can reproduce our results.

Modeling

Applying the two fold methods, we get the CV and test accuracies using four classification methods: logistic regression, LDA, QDA, and KNN.

Logistic Regression: The assumptions are that pixels are independent, there is a binary variable of 0 and 1, and there is linear relationship between logit of the response (cloudy or clear) and the features. The independence assumption may not be satisfied, as pixels that are closer to each other may be dependent. However, our fold methods divide the data in that pixels are sampled from different areas of the image - so we can roughly assume independence between our train and test sets We have binary variable assumption 0,1. We converted expert labels marked clear (-1) to 0 to maintain this assumption.

LDA: The assumptions are that the class conditional probability distributions are normally distributed with different means. The distributions all share the same covariance. The number of cloudy labels equals the number of clear labels. Here, we assume that boundaries are linear and the covariances are all the same for each class. The boundaries seem more quadratic than linear, if we look more closely at the maps of each image. The goal of LDA is to maximize the posterior probability. The class conditional probability distribution may be assumed to be Gaussian. Use MLE to estimate parameters .

QDA: The assumptions are that the class conditional probability distributions are independent Gaussians. Each class has its own covariance and is independent to those of other classes. Again, the distributions may not be independent Gaussians due to pixels not being independent - adjacent pixels are more likely to be similar. However, since our fold methods sample from different parts of the image, we reduce the possibility that we train a classifier that overfits (the training and validation sets versus the test set - they might end up being very similar).

KNN: The assumptions are that there are no explicit assumptions regarding the distribution of the data. This method could be useful for our data, since there is dependence among pixels and the distribution is not clear cut. We tried a grid search for a value of K using cross validation and chose 7 as the optimal value.

Accuracy Tables

Fold Method 1: CV Accuracies Across 7 Folds

Logistic Regression	LDA	QDA	KNN
0.792	0.814	0.865	0.867
0.714	0.752	0.733	0.718
0.969	0.973	0.99	0.738
0.814	0.845	0.796	0.922
0.792	0.797	0.728	0.677
0.999	0.999	0.997	0.932
0.699	0.702	0.688	0.597

Fold Method 1: Overall CV Averages Across 7 Folds

Methods	Average
Logistic Regression	0.825
LDA	0.84
QDA	0.828
KNN	0.779

Fold Method 2: CV Accuracies Across 7 Folds

Logistic Regression	LDA	QDA	KNN
0.888	0.882	0.899	0.919
0.918	0.917	0.952	0.909
0.843	0.844	0.853	0.775
0.856	0.864	0.896	0.698
0.932	0.934	0.946	0.892
0.73	0.758	0.662	0.758
0.772	0.774	0.78	0.809

Fold Method 2: Overall CV Averages Across 7 Folds

Methods	Average
Logistic Regression	0.848
LDA	0.853
QDA	0.855
KNN	0.823

Fold Method 1: Test Accuracy

Methods	Test.Accuracy
Logistic Regression	0.93
LDA	0.936
QDA	0.958
KNN	0.773

Fold Method 2: Test Accuracy

Methods	Test.Accuracy
Logistic Regression	0.92
LDA	0.925
QDA	0.953
KNN	0.925

Using all of the features, QDA had the best performance for both Fold Methods 1 and 2. QDA accounts for the different covariances among class labels, which might have been a factor in its performance. As shown in the Test Accuracy table above for Fold Method 1, KNN had low performance.

We also tried running each classification method using just the three features selected from the previous part, which were NDAI, CORR, and Radiance Angle AF. The test accuracy increased overall among all methods and the CV averages were higher. We decided to proceed with our analysis using just these three features.

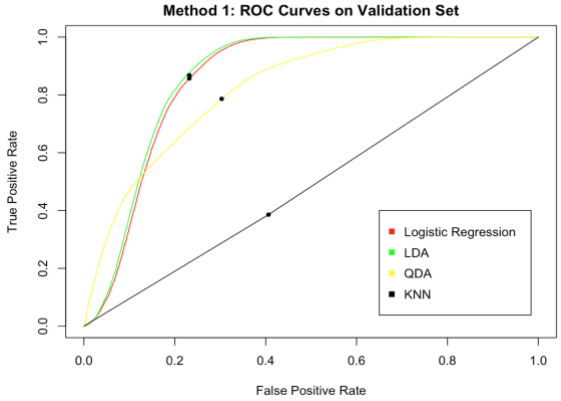
Fold Method 1: Test Accuracy with 3 Features

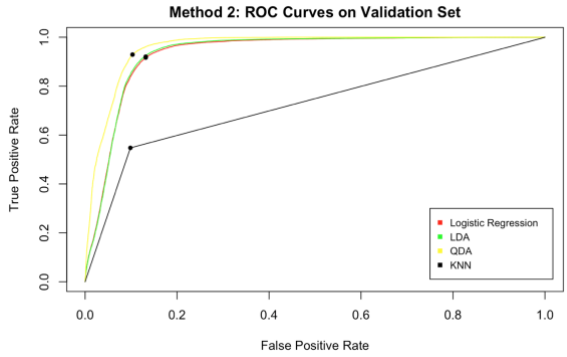
Methods	Test.Accuracy
Logistic Regression	0.929
LDA	0.929
QDA	0.929
KNN	0.919

Fold Method 1: Overall CV Averages Across 7 Folds using 3 Features

Methods	Average
Logistic Regression	0.84
LDA	0.848
QDA	0.849
KNN	0.794

ROC Curves





Method 1: Cutoff Values

Method	Cutoff
Logistic Regression	-0.794
LDA	0.322
QDA	0.304
KNN	1

Method 2: Cutoff Values

Method	Cutoff
Logistic Regression	-1.037
LDA	0.212
QDA	0.201
KNN	1

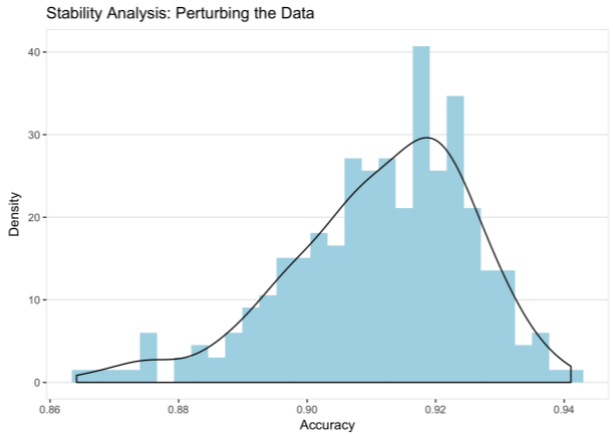
To find the cutoff value, we want the true positive rate to be as close to 1 as possible, and for the false positive rate to be as close to 0 as possible for optimality. We want to account for the best tradeoff possible

from our predictions. Hence, using a function from a tutorial on the ROCR package, we calculate the appropriate sensitivity and specificity values to get our cutoffs for each method. The QDA Model with 3 features improves to 94.6% test accuracy when we use the improved cutoff choice of .3.

Diagnostics.

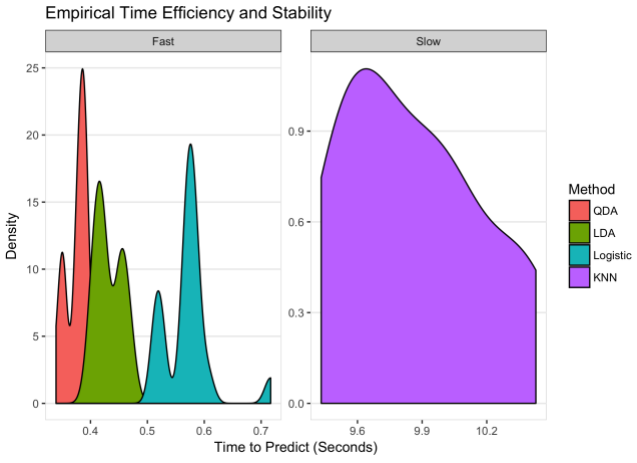
We achieved our highest overall test accuracy by using a QDA model fit with only the three first order conditions from Section 2. In this section, we will do an in depth dive into this model to assess its potential validity as a strategy to use on future real world data.

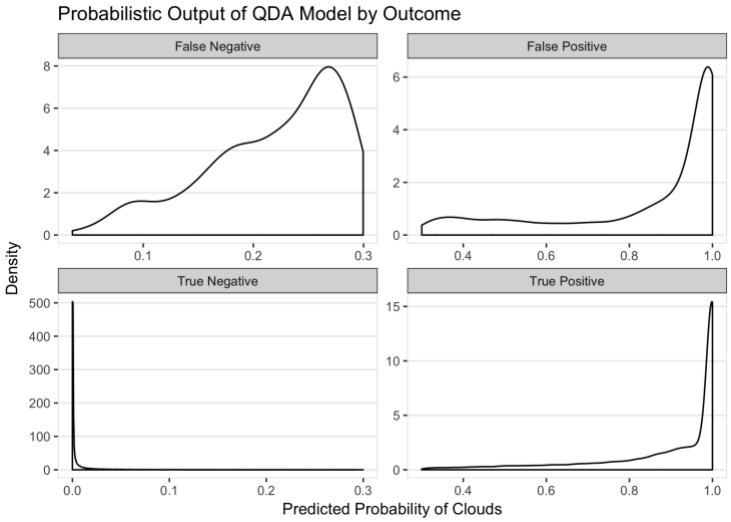
We feel confident that our model is not overfit to the data because with splitting method 1, the test set is a new and completely separate image. This is to imitate the way the model will need to be used in the future. However, we wanted to make sure that our results are stable across different perturbations of the training and validation features. We added independent random normal noise with mean zero and standard deviation equal to the features' standard deviation to each row of the table and recomputed the test accuracy after each iteration. As we can see in the figure below, the model is fairly robust. The average perturbed accuracy is about 91.1%. These results give us confidence in the stability of our model.



Next, we evaluated the time complexity of our model. According to the paper, "The sheer size of MISR data is a very important determinant of the statistical methodologies that are viable in this setting" (Shi et. al. 2008). It is important that our choice of model is sufficiently fast and does not have a slow worst case runtime. We ran a computational experiment to compare our final model with other candidate models. As is evident by the figure below, the QDA model is clearly the best choice on this basis, while LDA and logistic regression would be acceptable, and KNN is unacceptably bad.

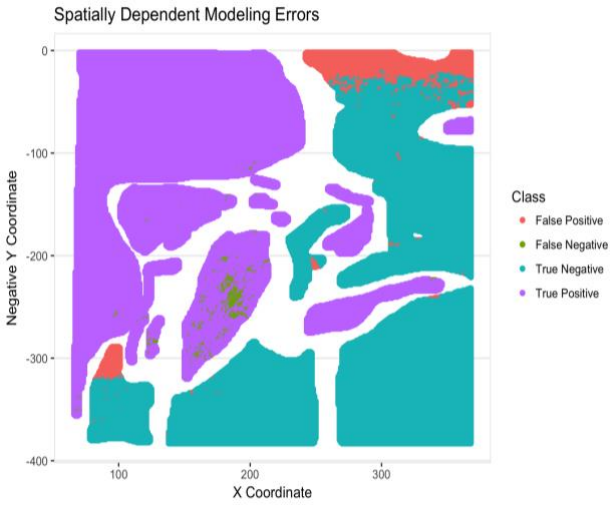
Next, we know that our model is only one part in a larger body of science and that there are many researchers working on the same task. We decided to get a measure of how probable it is that another team could improve the accuracy of this model while maintaining the same basic structure. If this model is to be operationalized and made standard, it would be useful if we can make incremental improvements without a complete structural change. For each outcome (true positive, true negative, false positive, false negative), we look at the density of the model's predicted probabilities.

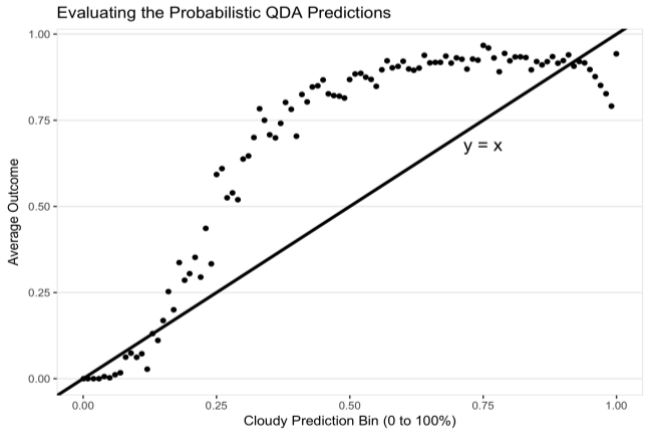




As we can see in the figure on the left, the model is usually very sure when it detects a true negative. Unfortunately, false positives and true positives look very similar, and it will be difficult to marginally improve the model and reduce the false positive rate. However, there is more area for improvement in removing false negatives. The model is usually relatively unsure when it erroneously predicts no clouds. We can verify this by noting the large density in the false negative panel near 30% (our optimal cutoff from part 3) predicted probability of clouds. Future work could improve the model with a new feature that helps separate these false negatives from true negatives.

We know that the assumptions for QDA are not really met with this data. The data are not independent and there's no reason to believe that the features follow a normal distribution. For this reason, it is dangerous to try to use the QDA output probabilities as a "real" probability for each class. For the final step of our in-depth model evaluation, we will demonstrate exactly how these violated assumptions can cause problems. In the plot below and to the right, each point represents many predictions binned together by rounding predictions to the nearest 1%. The y-axis shows the average outcome. That is, the point is located at (.2, .35) shows that among all times the model predicted clouds with probability .2, there really were clouds 35% of the time. If the probabilistic predictions were to be trusted, these points should more or less fall on the identity line. This is clearly not the case, so we should not use this model to represent probability. This does not change the fact that the model achieves incredible accuracy of almost 95% on unseen test data. The model's binary predictions (using a cutoff of 30%) can still be extremely useful.





While the model achieves a test accuracy of 94.6%, we checked whether the errors seem to be randomly

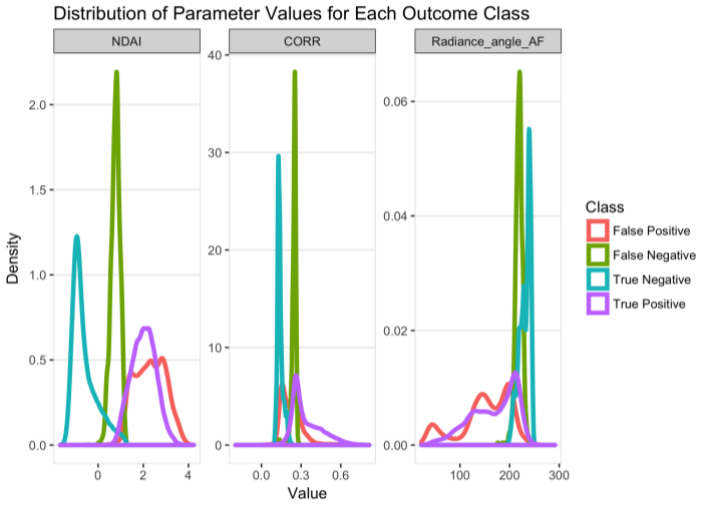
dispersed across the picture area. As we can see in the figure above and to the left, the errors have large spatial dependence. We missed a significant part of the cloud in the middle of the image and incorrectly predicted clouds in the top right and bottom left. There are also some scattered smaller sources of error -- mostly false positives floating near large areas of true negatives. We will propose an interesting method of removing these scattered errors in the next section.

Because the pixels in the test set are roughly split 50/50 between cloudy and non-cloudy, the summary table to the left shows us that we were much more likely to incorrectly identify something that is not a cloud (false positive) than to miss a real cloud (false negative). Whether this is an issue or a benefit depends on the specific application of the model -- for example in medical settings a false negative is a much bigger problem than a false positive. Since cloud detection can be used to a variety of climate science applications, researchers should always choose a model that best fits their own needs.

In Depth Accuracy of 3 Feature QDA

Class	Proportion
False Positive	0.047
False Negative	0.008
True Negative	0.475
True Positive	0.47

Next, we investigated the distribution of parameter values that lead to each class of prediction. The results confirmed what we found above -- false positives and true positives are very confounded while false negatives have some separation from true negatives. Specifically, there is very little overlap in the densities of NDAI and CORR for the true negative and false negative predictions. This very separation is likely what allows us to achieve less than 1% total false negatives but nearly 5% total false positives.

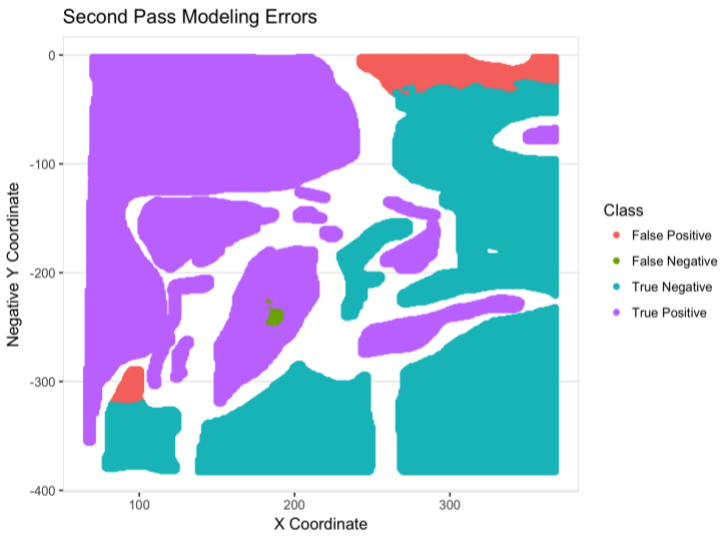


Based on this analysis, we can think of a few ways to potentially improve our model. We propose adding a second, hierarchical, step to our modeling process. After fitting the predictions as we've already done, this new model would go back and examine each prediction with a k nearest neighbor algorithm where the only features are the x and y-coordinates. If all or nearly all of the points physically near a given point do not belong to a cloud, the point is very unlikely to be part of a cloud (and vice versa). For this method, we would choose K to be fairly large so that small local clusters of several points cannot all influence each other.

In implementing this, we would need to be careful of two main issues. First, we would not reflect any updated predictions until after checking every point. This would help us avoid cascading and potentially devastating errors where we change large clusters of correctly labeled points. Second, we would need to make sure that we choose a large cut off before changing a prediction (e.g. almost all of the neighbors would need to disagree). This will ensure that correctly labeled points on the border between a cloudy and non- cloudy region do not get incorrectly changed. This K nearest neighbor step is a sanity check, and should not alter a large number of the predictions.

The benefit of this technique is that it works by exploiting the inherent spatial dependence. This spatial dependence will absolutely be reflected in unlabeled images in future prediction tasks, even if the fitted parameter values change. As long as the method achieves a high accuracy on the first pass, this should be a safe and effective way of cleaning and verifying the predictions.

We implemented this and improved our final accuracy from 94.6% to 95.5%, so we removed about 20% of our errors. As we hoped, the resulting spatial plot shows fewer scattered errors.



We are confident that this model will work well on future data with no expert labels. This was the main reason we used a splitting method that left an entire independent image for our testing set. The pattern of cloudy and clear skies is very different in the test set compared to the training and validation sets, so the commonality must be that the features are predictive of cloudiness.

Our main concern is that these images came all came from a reasonably small period of time. With the rapidly changing climate, we cannot be sure that the learned
model parameter values will stay accurate over long periods of time. With that said, we believe that this would be a problem for any model fit on data from a single period of time. We would recommend that the model is visually checked against future data. It does not need to be checked on a pixel by pixel level, but just that it stays accurate at identifying large areas of clouds and clear skies.

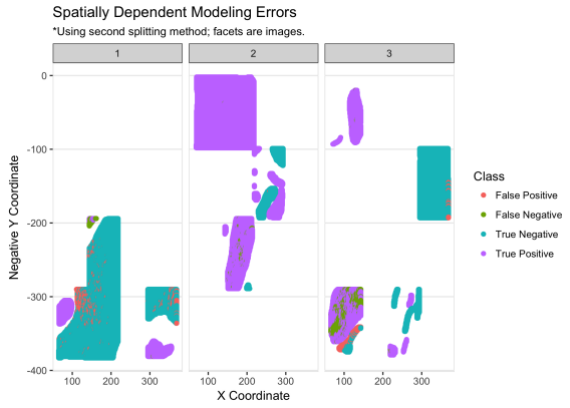
Second Pass Accuracy Breakdown

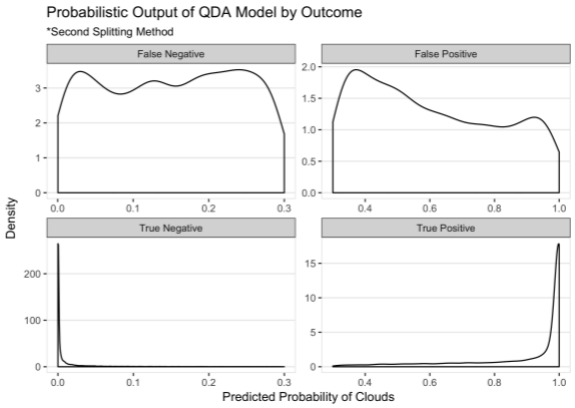
Class	Proportion
False Positive	0.043
False Negative	0.002
True Negative	0.479
True Positive	0.476

We verified that our model is stable to changes in the data splitting method. We were confident that it would be, given that the second splitting method is less conservative than the first. We will only display the parts of the replicated analysis that are significantly different for the second method, but you can easily recreate all of it by running the last part of our R Markdown document included with our GitHub repository.

These plots were generated the same way as the matching ones above, but there is now one important qualitative difference in the results. With the new splitting method, our overall accuracy was similar at about 94%, but the errors are now spread more evenly between false positives and false negatives (instead of being nearly all false positives). We can see from the spatial dependence plot that now that we're testing on subsets of images that were all used to fit the model, there are no longer large areas of completely misclassified points. From the probabilistic output figure below, we can see that there is no longer a clear area of

attack to improve the accuracy of the model. The false negatives and false positives are both nearly uniformly spread across their possible domains.





To summarize all of our results, we believe that a three feature model using NDAI, CORR, and Radiance Angle AF and our first splitting method is the best model to proceed with. We do recommend the second pass with the KNN classifier on the learned labels, as it allows the model to self-correct anomalous predictions. As shown above, this model achieved .2% false negative rate on unseen test data. This means that if scientists were to use our model, (and if the physical forces driving the result have not changed since this data was collected), will be able to correctly identify all real clouds. The next step is to reduce the approximate 4% false positive rate, but we are confident that we have built a highly stable and highly accurate classifier that is ready to be used on future unseen and unlabeled data.

GitHub Repository https://github.com/bmjoseph/Stat_154_Proj_2

Acknowledgements Deborah completed Parts 1 and 3 of the project. She did EDA e.g., mapped out the images, analyzed any possible associations between features and expert labels to grasp a better understanding of the data prior to modeling. Applying what we learned from lectures and the material and Bailey's CV function, she used four classification methods to model the data and see how well each algorithm performed in predicting whether an area was cloudy or clear. Bailey completed Parts 2 and 4 of the project. He came up with the two splitting methods, the three most important features, and wrote the functions in helpers.R (including `CVgeneric`). He went off of Deborah's modeling work from part 3 to do an in depth dive into the results for part 4. Bailey and Deborah worked together to compile our codes and push everything to GitHub.

Resources

Strictlystat. "A Small Introduction to the ROCR Package." *A HopStat and Jump Away*, 22 Dec. 2014, hopstat.wordpress.com/2014/12/19/a-small-introduction-to-the-rocr-package/.

Tao Shi, Bin Yu, Eugene E Clothiaux & Amy J Braverman (2008) Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies, Journal of the American Statistical Association,103:482, 584-593, DOI: 10.1198/016214507000001283