STA 521 Project 2 Cloud Data Report

Yicheng Shen (yicheng.shen@duke.edu) & Yunhong Bao (yunhong.bao@duke.edu)

02 December, 2022

1 Data Collection and Exploration

1.1 Background & Motivation

With the global climates getting increasingly extreme, humans are making the best use of sciences and technologies to understand the environment, especially in the Arctic region. The detection of clouds in satellite images has become an important task, as cloud coverage is closely related to the surface air temperatures and atmospheric carbon dioxide levels. Yet it is a challenging problem since clouds are similar on snow- and ice-covered surfaces. In this study, we are going to examine various classification methods, build reliable models that distinguish the presence of cloud from Arctic satellite images using available features and evaluate our models' performance.

The data is obtained from a study by Yu et al. (2008). This team of researchers collected the data via the camera of Multiangle Imaging SpectroRadiometer (MISR) launched by the NASA. The data are in the forms of image pixels, with each MISR pixel covering a 275 m by 275 m region on the ground. Since standard classification frameworks of clouds were not readily applicable, their goal was also to build operational cloud detection algorithms that can efficiently process the massive MISR data set one data unit at a time without requiring human intervention or expert labeling.

Yu et al. (2008) proposed two algorithms, an enhanced linear correlation matching (ELCM) algorithm based on thresholding the three features with values either fixed or data-adaptive, and an ELCM algorithm with Fisher's quadratic discriminant analysis (ELCM-QDA).

Their results suggest that both proposed algorithms are computationally efficient for operational processing of the massive MISR data sets. The accuracy and coverage of ELCM are better and more informative compared with conventional MISR operational algorithms. This findings provides important implications and foundations for further analysis of MISR data.

1.2 Data Description

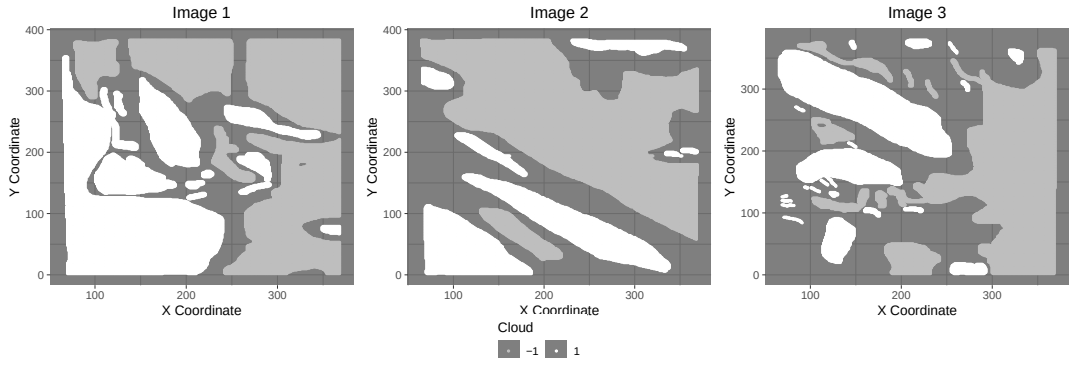

Figure 1: Maps of three images with expert labels. White represents high confidence of cloud; gray represents high confidence of clear; and dark represents unlabeled pixels.

In this study, we primarily focus on three of the MISR images. These three images contain 115110, 115229 and 115217 pixels respectively. However, not all pixels are labeled with confident experts' classification. As shown in in Figure 1, there are considerable portions of images not labeled. We notice the pattern that usually experts have difficulties distinguishing the areas around what they recognize as clouds. It is understandable that the borderlands between cloudy and clear surfaces are more challenging to determine.

We can also observe that labeled clouds are often clustered in chunks. Therefore, adjacent pixels' labels are not independent and instead seem to have very high positive correlations. For example, if a pixel's neighbors are all labeled as clouds, it is highly likely that it is also labeled as part of the cloud.

In Table 1, we present the percentages of expert labels in each image. Since we have no confident expert opinion on unlabeled pixels, they are viewed as missing / unknown values. After removing unlabeled pixels, we have 82148, 70917 and 54996 pixels in each image, with available information (labels and features).

Table 1: Proportions of Cloudy and Clear Surfaces by Expert Labeling

Cloud Labels	Clear Labels	Unknown	Image
0.3411172	0.3725306	0.2863522	1
0.1776549	0.4377891	0.3845560	2
0.1843825	0.2929429	0.5226746	3

In specific, each pixel has eight features. The first three are physically useful features for characterizing the scattering properties of ice and snow covered surfaces: the correlation (`CORR`) of MISR images of the same scene from different MISR viewing directions, the standard deviation (`SD`) of MISR nadir camera pixel values across a scene, and a normalized difference angular index (`NDAI`) that characterizes the changes in a scene with changes in the MISR view direction.

The latter five are five view zenith angles of the cameras. 70.5° (DF), 60.0° (CF), 45.6° (BF), and 26.1° (AF) in the forward direction and 0.0° (AN) in the nadir direction.

1.3 Explortary Data Analysis

Conducting a EDA on the available features gives a preliminary picture of the relationship within potential predictors as well as with the response. First of all, we notice in Figure 2 that there is positive correlation between the features. The correlation is particularly strong and positive among the five radiance angles. For example, `BF`, `AF` and `AN` are very strongly correlated. We also see positive correlations between `NADI`, `CORR` and `SD`.

However, the correlations between the first three features (`NADI`, `CORR` and `SD`) and five radiance angles (`Df`, `CF`, `BF`, `AF`, `AN`) are mostly negative.

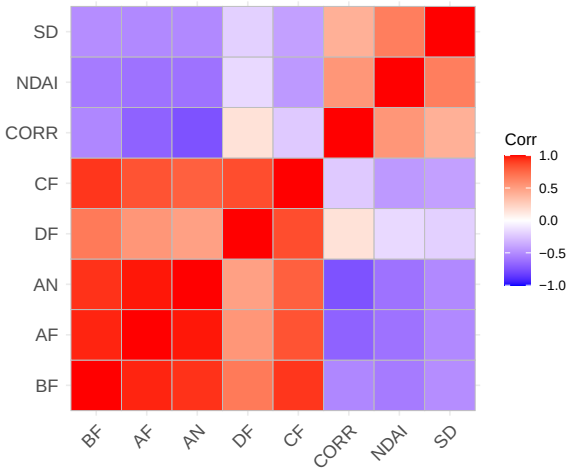

Figure 2: Pair-wise correlations between eight available features.

Figure 3 shows that for pixels labeled as cloud, their `NADI`, `CORR` and `SD` are likely to be higher than those labeled as no cloud. This distinct pattern provides strong support to use these features as predictors of our classification models. It also seems reasonable to have a log transformation of `SD` so that it is in a similar scale as other features.

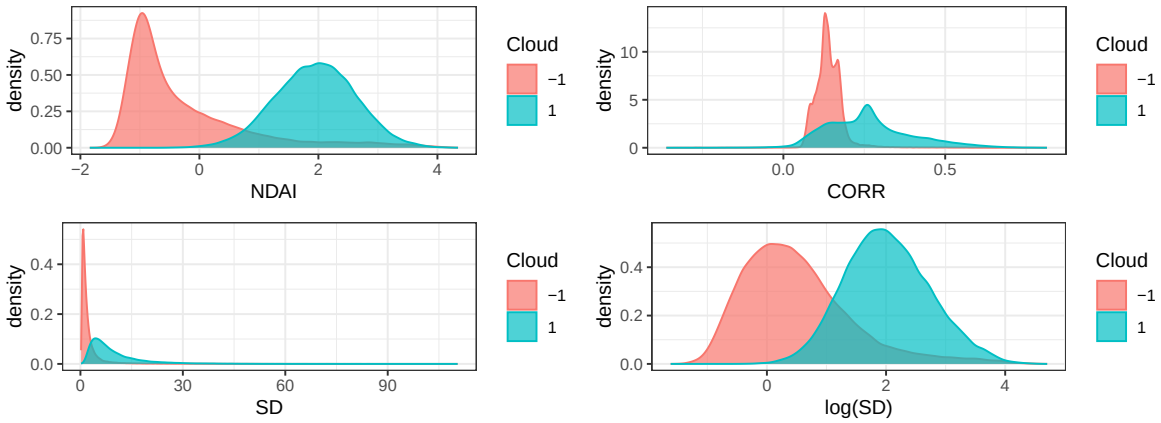

Figure 3: Density distributions of three features that describe cloud and clear pixels.

In terms of other features, we can see that the density distributions of radiance angles of cloud and cloud-free pixels are pretty consistent across angles in Figure 4. The radiance angles of cloud pixels are usually wider and right skewed, whereas the radiance angles of cloud-free pixels are usually higher and distributed in a bimodal shape. The distributions between cloud and cloud-free pixels are not as separable and distinct as the first three features.

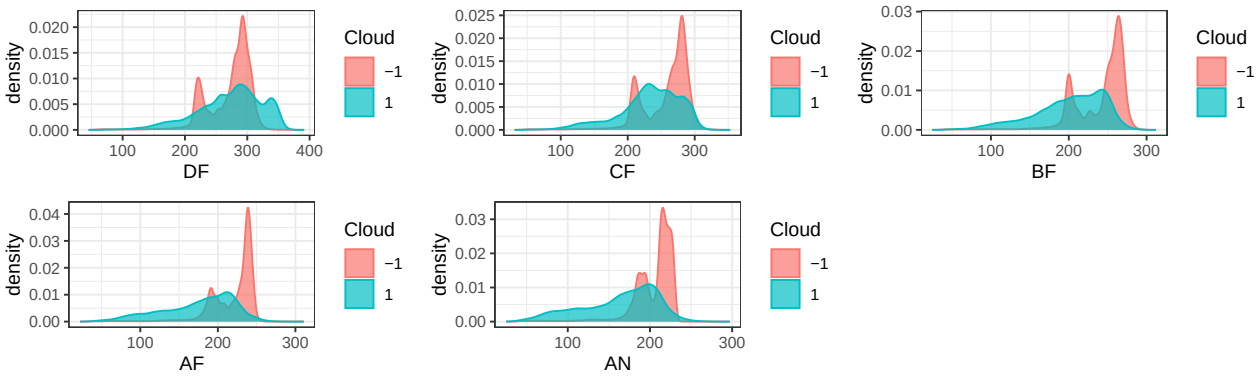

Figure 4: Density distributions of radiance angles from cloud and clear pixels.

2 Preparation

2.1 Data Splitting

Before building models, it is important to prepare the data as training, validation and testing sets so that we can make the best use of the data and evaluate our models. The key idea of our data splitting is to take into account the fact that this data set is not i.i.d. Therefore, we propose the two following ways of dividing data into blocks.

Method 1. Horizontal Cuts: The first method cuts each image horizontally in order to ensure that every resulting block has a reasonable portion of clouds and clear surfaces. Basically, each image is cut into five blocks by evenly dividing Y coordinates (shown in Figure 5), and three of them would be used as training data, the rest two blocks are used as validation and testing respectively. This methods splits the data into

58.59% training data, 19.03% validation data and 22.38% testing data (roughly 3:1:1).

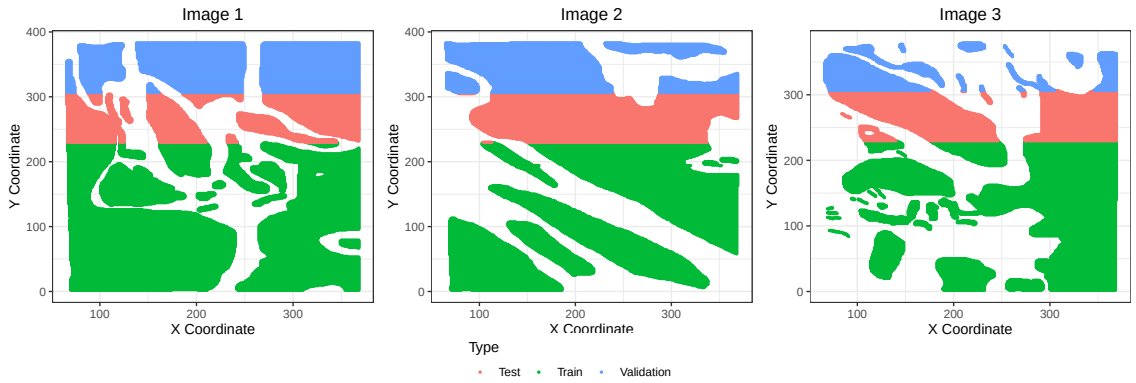

Figure 5: The first data splitting method is to divide each image horizontally.

Method 2. K-means Clusters: The second method of blocked data splitting is to use the K-means algorithm. By selecting a cluster size of five, we can divide each image's datapoints into five distinct groups (according to X-Y coordinates). Again, shown in Figure 6, three of these are used for training data, one is for validation and the last one is for testing. The K-means method splits all datapoints into 60.72% training data, 20.04% validation data and 19.24% testing data.

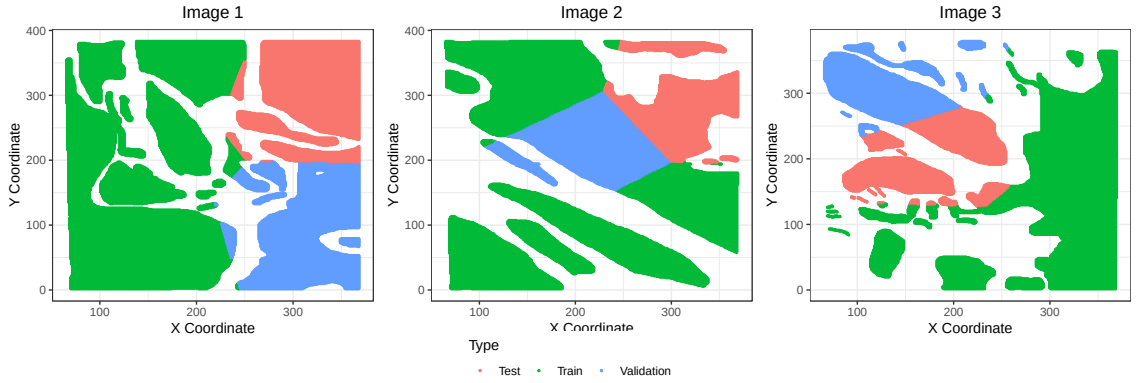

Figure 6: The second data splitting method is to divide based on the K-means algorithm.

The table below describes how much cloud pixels there are in each set after two ways of data splitting. We think they are both reasonable in terms of having both cloud and clear pixels in every subset of data.

Table 2: Proportions of cloud pixels in each set

Method	Type	Cloud Prop	Method.1	Type.1	Cloud Prop.1
Horizontal Cuts	Test	0.3384662	K-means	Test	0.3296281
Horizontal Cuts	Train	0.4778361	K-means	Train	0.4479412
Horizontal Cuts	Validation	0.1760129	K-means	Validation	0.2684817

2.2 Baseline Accuracy

We can examine the accuracy of a trivial classifier which sets all labels to -1 on the validation set and on the test set as shown in the table below. This classifier assumes that the image has no cloud pixels at all. Logically, the accuracy, or the success rate, of this trivial classifier depends entirely on the percentage of

actual cloud free pixels labeled in the data. If the image is mostly cloud free, then labeling all points as -1 would easily achieve a high accuracy.

Table 3: Accuracy of a trivial classifier that sets all labels to -1

Method	Data Type	Accuracy
Horizontal Cuts	Validation	0.8239871
Horizontal Cuts	Test	0.6615338
K-means	Validation	0.7315183
K-means	Test	0.6703719

Here it is shown that labeling all points as -1 can only achieve an accuracy of 66.15% and 67.04% on the test data sets, which shows that to achieve high accuracy in this classification problem is not trivially easy. This also sets a baseline of reference for our subsequent classification methods.

2.3 First order importance

We consider `NADI`, `CORR` and `logSD` (notice that we apply a log transformation on `SD` as justified in section 1.3) as the three most important features when classifying pixels. Firstly, the EDA as shown in Figure 3 and 4 has shown that the density distributions of the first three variables are more distinct between cloud and non-cloud pixels compared with the angular radiance variables.

Then we also examine the pair-wise correlation between the response, `Cloud`, and each of the eight features. Looking at the two bottom rows in Figure 7, we notice that the correlation between `Cloud` and `NADI`, `CORR` and `logSD` are much stronger than the other five features.

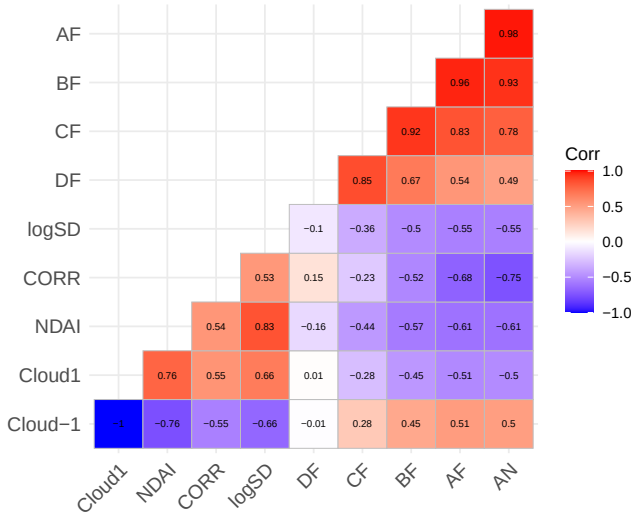

Figure 7: Pair-wise correlations between response (Cloud) and features (eight covaraites).

We also apply PCA on the eight features, which suggests that first three PCs could explain over 95% of the total variance. Among the first three PCs, `NADI`, `CORR` and `logSD` again consistently contribute a lot to the total variance. Thus, we tentatively suggest these three as the best features without model justification.

2.4 Cross Validation Method

After splitting training and test data, we use various classification methods to train models that can classify cloud and non-cloud pixels based on available features. To make the best use of the training data, we employ cross validation as a way to compare models.

As shown in our code appendix and github repository, our `CVmaster` function is able to take a generic classifier, training features, training labels, number of folds K and a loss function as inputs, then perform K-fold cross validation (CV) to assess the classification methods, and finally outputs the K-fold CV loss on the training data.

3 Modeling

In this section, different machine learning algorithms including logistic regression, LDA, QDA, Naive Bayes, and gradient boosting are used to construct the classification model. Hyperparameters are selected base on cross validation accuracies. Performance is assessed through both CV error and prediction accuracy on the test data. A comparison across model fit is also conducted utilizing various performance metrics including AUC and Gini Index.

3.1 Model Fitting

Before fitting the chosen classification algorithms, we first assess different model assumptions and their fitness on the given data set. Logistic regression assumes linearity in the classification log odds. Although such a strong assumption is not satisfied by the data, we can adjust the classification threshold to still achieve a desirable accuracy. LDA assumes predictors in different classes follow normal distributions with the same covariance matrix Σ but different means μ_j, while QDA allows distinct class covariance matrices Σ_j. From Figure 3 and 4, we can observe that covariants are hardly normal. Thus LDA and QDA's assumption of normality is unsatisfied here. Naive Bayes makes the assumption that conditional on the class labels, predictors follow iid normal distributions. Such a assumption is also unsatisfied base on Figure 3 and 4. Random Forest, on the other hand, do not have any model assumptions, and should be able to provide good results.

After assessment of model assumption fitness, different classifications methods are applied to the data set. For Boosting Tree algorithm, hyperparameters including max-depth of each weak learner, number of weak learners, and the learning rate are tuned through K-fold validation. For algorithms that returns the classification probability, we determine the model threshold by consulting the Youden statistics which maximized the sum of specificity and sensitivity. Both CV and test accuracies are displayed in the following table:

Table 4: 10-fold CV Results and Test Accuracy based on Two ways of Data Splitting

classifier	data	fold.1	fold.2	fold.3	fold.4	fold.5	fold.6	fold.7	fold.8	fold.9	fold.10	Cv.mean	Test
Logistic	Horizontal Cut	0.6989	0.7931	0.7025	0.9983	0.7148	0.9810	0.9018	0.9190	0.9984	0.9788	0.8687	0.8705
Logistic	K-means	0.9621	0.9950	0.7803	0.9982	0.7681	0.9983	0.8340	0.6920	0.7516	0.8660	0.8646	0.9065
LDA	Horizontal Cut	0.6830	0.8055	0.7082	0.9981	0.8046	0.9783	0.9012	0.9272	0.9983	0.9780	0.8783	0.8779
LDA	K-means	0.9619	0.9932	0.7916	0.9980	0.8274	0.9964	0.8431	0.6935	0.7418	0.8649	0.8712	0.9089
QDA	Horizontal Cut	0.6891	0.8274	0.6518	0.9989	0.8559	0.9926	0.8818	0.8983	0.9969	0.9863	0.8779	0.8957
QDA	K-means	0.7512	0.6402	0.8823	0.9978	0.8942	0.9990	0.8449	0.9705	0.8197	0.9966	0.8796	0.9031
Naive Bayes	Horizontal Cut	0.5574	0.8075	0.6151	0.9990	0.9580	0.9396	0.8071	0.8654	0.9877	0.9713	0.8508	0.8455
Naive Bayes	K-means	0.9340	0.9204	0.8083	0.9986	0.9618	0.9912	0.7678	0.6008	0.6447	0.8075	0.8435	0.8974
Boosting Trees	Horizontal Cut	0.9958	0.7299	0.9925	0.5908	0.9662	0.9596	0.8636	0.9685	0.9286	0.9981	0.8994	0.9393
Boosting Trees	K-means	0.9573	0.9996	0.9085	0.9852	0.5777	0.9966	0.8511	0.9811	0.8618	0.7360	0.8855	0.9499

As displayed in the table, Boosting Trees algorithm achieve the best CV and Test errors. This result aligns with our expectation as Boosting Tree has the weakest model assumption thus is able to fit data well. LDA, QDA, Naive Bayes, on the other hand, have a lower accuracy since their model assumptions are unsatisfied by the data. Logistic regression has a relatively low CV mean while a higher test accuracy, an explainable result since CV classification threshold is set to 0.5 while final model threshold is adjusted by Youden statistics.

3.2 ROC Curves

(b) Use ROC curves to compare the different methods. Choose a cutoff value and highlight it on the ROC curve. Explain your choice of the cutoff value.

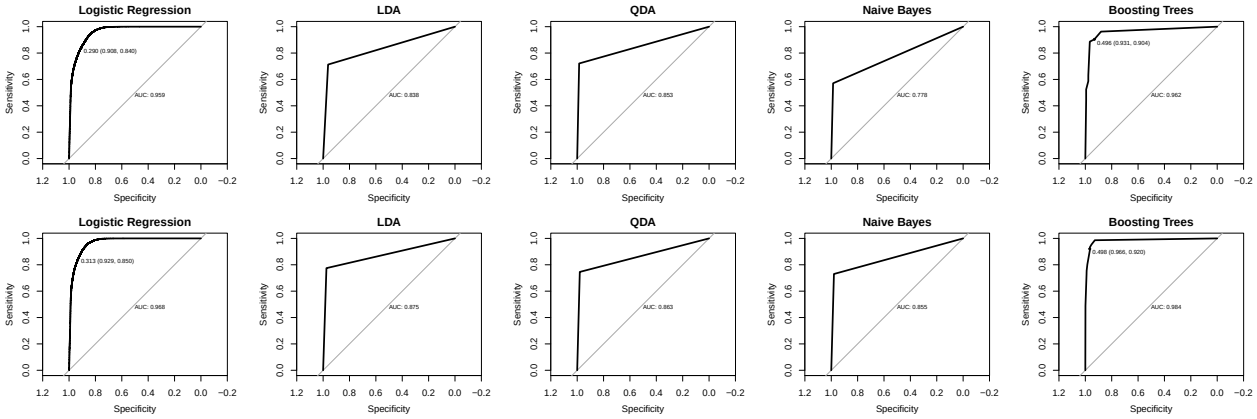

Figure 8: ROC curves on the test data (the first row is horizontal cut data, the second row is K-means splitted data)

3.3 More Model Assessments

(c) (Bonus) Assess the fit using other relevant metrics. Use quantitative measures and show clean and interpretable figures!

4 Diagnostics

4.1 Examining Boosting Trees

(a) Do an in-depth analysis of a good classification model of your choice by showing some diagnostic plots or information related to convergence or parameter estimation.

4.2 Patterns of Missclassification

(b) For your best classification model(s), do you notice any patterns in the misclassification errors? Again, use quantitative and visual methods of analysis. Do you notice problems in particular regions, or in specific ranges of feature values?

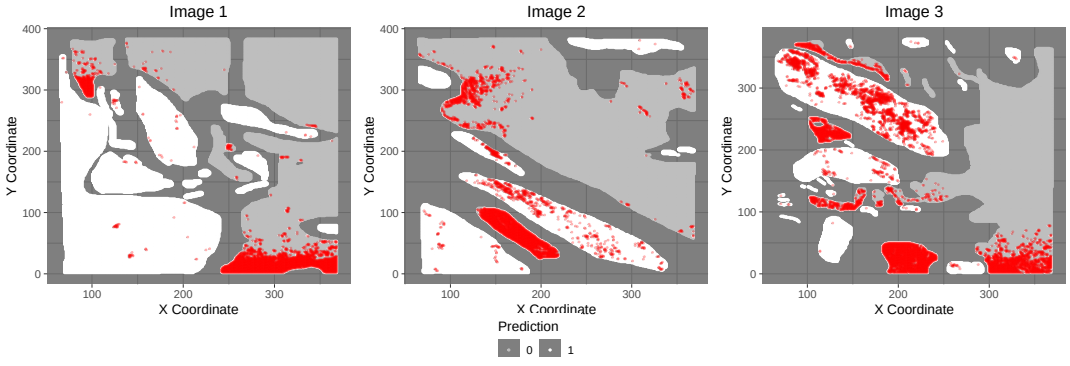

Figure 9: Prediction of cloud pixels using boosting trees, with missclassified points highlighted as red.

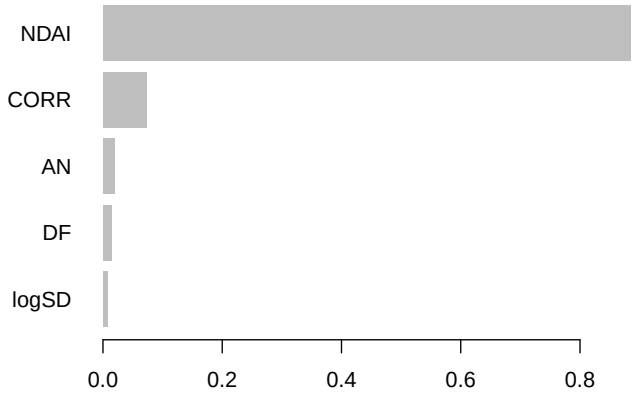

Figure 10: The feature importance plot in boosting trees.

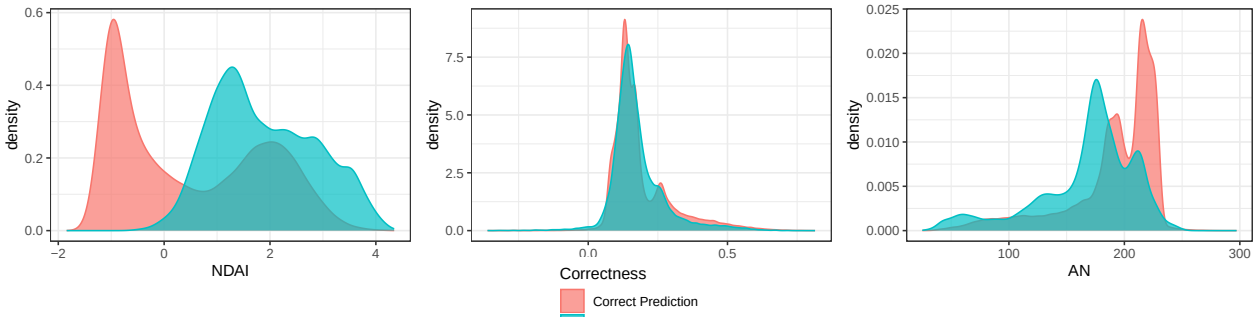

Figure 11: The feature density plot of correctly and wrongly predicted pixels.

4.3 Better Classifier

(c) Based on parts 4(a) and 4(b), can you think of a better classifier? How well do you think your model will work on future data without expert labels?

With low CV and test errors, the proposed XGboost model framework will have a robust performance in future classification of unlabeled data points. Our block data splitting approach provides the functionality of mimicing a new data set. Thus the reported CV and test errors are reliable estimates of model accuracy.

Despite the high model accuracy, there are two potential improvements. One potential improvement is to add L_1 or L_2 regularization terms to reduce model complexity. A simpler model potentially reduces model variance and avoids over-fitting. Moreover, a combination of supervised and unsupervised learning methods might be beneficial. Observing that spatially close data points tend to share the same classification label we propose the following procedure for classification:

Step 1. Utilize pre-trained XGBoost Classifier on the desired data set to get the classification probability p_i for each data point i.

Step 2. Define an upper bound UB and lower bound LB, $UB>LB$, to be the classification thresholds. Classify data point i to 1 if $p_i \geq UB$, to 0 if $p_i \leq LB$.

Step 3. For the remaining data points j, where $LB < p_j < LB$, utilize KNN on geographical coordinates $(X, Y$ in this case) to classify point j to the class of its closet neighbors.

This methods would avoid false classification of data points within a cluster of same-class data and therefore increase the overall accuracy. However, it does not provide any performance improvements over datapoints on the cluster boundary.

4.4 Modefied way of data splitting

(d) Do your results in parts 4(a) and 4(b) change as you modify the way of splitting the data? [what if the data is splitted independently?]

4.5 Conclusion

(e) Write a paragraph for your conclusion.

Overall, there are various methods that we can try to employ on the cloud classification problem. Of the methods discussed in this report, boosting trees yield the best performance on the training and test data.

5 Reproducibility

In addition to a writeup of the above results, please submit a zip file containing everything necessary to reproduce your writeup to Gradescope "PROJ2 code". Specifically, imagine that at some point an error is discovered in the three image files, and a future researcher wants to check whether your results hold up with the new, corrected image files. This researcher should be able to easily re-run all your code and produce all your figures and tables. This zip file should contain:

(i) the raw Latex, Rnw, Qmd or Word used to generate your report,
(ii) your R code (with CVmaster function in a separate R file),
(iii) a README.md file describing, in detail, how to reproduce your paper from scratch (assume researcher has access to the images).

6 Reference

Yu, Bin, Tao Shi, Eugene E Clothiaux, and 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): 584–93.