Project 2 Cloud Detection Data Report

Team TBD: Camilla Yu, yy330@duke.edu & Shuo Wang, sw532@duke.edu

Acknowledgement

Camilla and Shuo contributed equally to the project. We discussed EDA, how to separate images, classification model choices, cross validation (tune parameter) as well as diagnostics together. At first, we checked correlations between variables and plotted images to have a general sense about the data. Then, two separating methods were used to solve for the problem that points are not iid. Before fitting the classification methods, we tested the accuracy for trivial classifier as the base model as well as checked assumptions for classification models we would use later. Shuo worked majorly on building CVmaster, while Camilla focused on analyzing the output as well as diagnostic criteria. Finally, we worked together on fitting classification models, write-up and Readme file. Special thanks to the help from Professor Chen, Jiongran Wang and Google.

1. Data Collection and Exploration

a. Summary

To predict the dependences of surface air temperature and atmospheric carbon dioxide level in Arctic, accurate Arctic-wide measurements, especially like cloud coverage are required. So, the study proposes Arctic cloud detection algorithms using Multiangle Imaging Spectro Radiometer (MISR) imagery to identify cloud-free surface pixels.

Data were collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and Baffin Bays. The path 26 was chosen for studying because of the richness of its surface features. Six data units from each orbit are included in the study. The data used for training classification models are a collection of features (x coordinate, y coordinate, expert label, NDAI, SD, CORR, DF, CF, BF, AF, AN) from three images. And these images are three data units collected from MISR blocks 20-22 over three consecutive orbits. Expert labels were used as true class to evaluate the performance of different cloud detection algorithms.

Electromagnetic radiation measurements were captured by nine cameras at nine different angles and only angles of DF, CF, BF, AF, AN are available in our dataset. AN in the nadir direction (0 degree and vertically downward) is the best angle to capture the whole picture of clouds because it is less influenced by atmospheric scattering and surface topographic effects. NDAI was calculated as the ratio between the difference of the mean radiance measurements and the sum of them by using some specific pixel regions. CORR was calculated as the average linear correlation between radiance measurements and SD is the standard deviation within different groups of angles for MISR. And specific formulas for calculating these three features are clearly explained in the paper written by Shi. et al..

In general, ELCM and ELCM-QDA have the best performance in both accuracy and coverage, and they are consistent with the expert labels as well as more informative. This study will enable scientific community to study the relationship between changing cloud properties in Arctic and atmospheric carbon dioxide, which is significant for analyzing global climatic models in the future.

b. Plots of Images

Figure 1 shows three unlabeled images plotted using AN radiance because AN is the best angle to capture the whole picture of clouds and could give us plots that look like satellite pictures. From the plot, we could easily identify there are at least two classes using human eyes, but it’s hard to use only AN value to classify them since two classes have similar overlapped range of AN values, which suggests us to use more features for classification models.

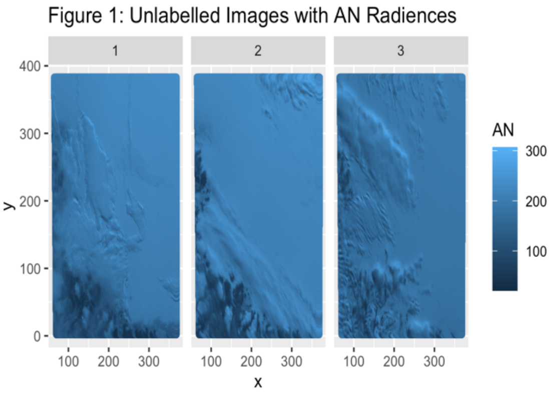


Figure 1: Unlabelled Images with AN Radiences

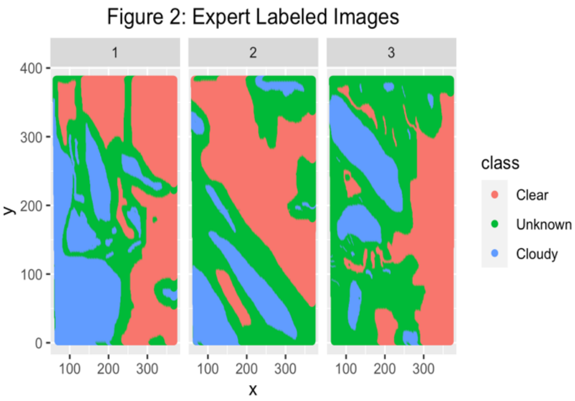


Figure 2: Expert Labeled Images

Compared to figure 1, figure 2 shows three expertly labeled images. For the first image, 37.25% pixels are clear, 28.63% pixels are unknown, and 34.11% pixels are cloudy. Similarly, for the second image, 43.78% pixels are clear, 38.46% pixels are unknown, and 17.77% pixels are cloudy. Lastly for the third image, 29.29% pixels are clear, 52.27% pixels are unknown, and 18.44% pixels are cloudy. In general, three images have different proportions for three classes, but they are not far apart. And from the plot, points from the same class tend to cluster in similar regions, which indicates that points are not identically independently distributed, and we can’t random choose points to train model since one point contains information about other points which are close to it. And the boundaries for different classes seem not to be linear or easily identified.

c. Visual and Quantitative EDA

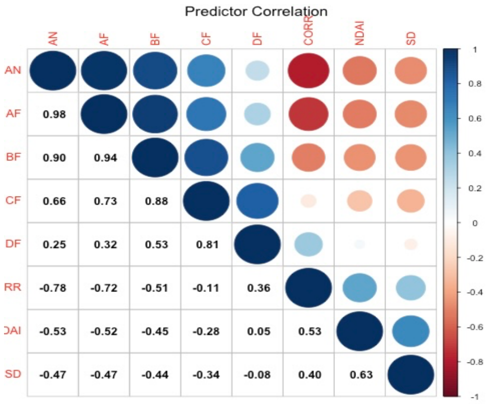
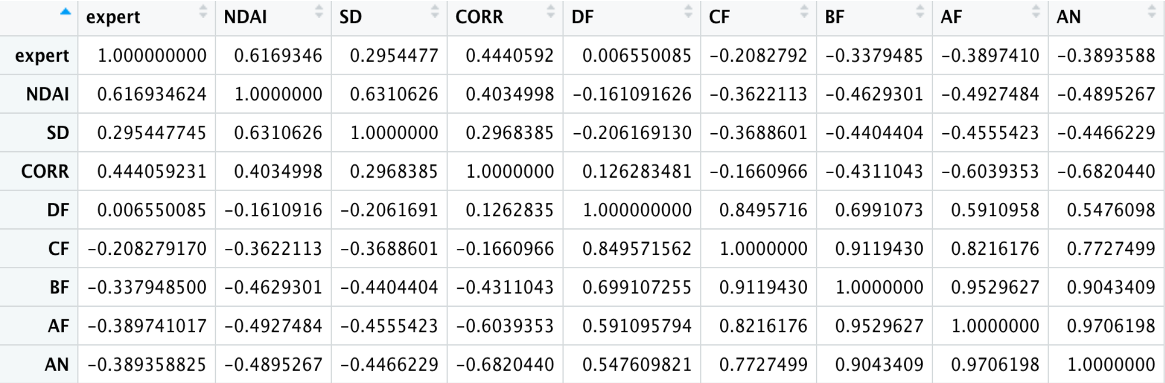


Figure 18 *Feature Correlation*

Table 1 and figure 18 show the correlation between features themselves as well as the correlation between the expert labels and individual features. For radiance features, except for DF, they are all strongly correlated with each other with correlation higher than 0.7 since the only difference between these radiance features is the angle when taking the picture. For NDAI, SD and CORR, they seem to have weaker correlation with each other, but have relatively stronger correlation with radiance features, which is reasonable since NDAI SD, and CORR are calculated based on radiance features. NDAI, CORR, AF and AN have relatively high correlation with expert labels, which might suggest they play important roles in classifying classes.



	expert	NDAI	SD	CORR	DF	CF	BF	AF	AN
expert	1.000000000	0.6169346	0.2954477	0.4440592	0.006550085	-0.2082792	-0.3379485	-0.3897410	-0.3893588
NDAI	0.616934624	1.0000000	0.6310626	0.4034998	-0.161091626	-0.3622113	-0.4629301	-0.4927484	-0.4895267
SD	0.295447745	0.6310626	1.0000000	0.2968385	-0.206169130	-0.3688601	-0.4404404	-0.4555423	-0.4466229
CORR	0.444059231	0.4034998	0.2968385	1.0000000	0.126283481	-0.1660966	-0.4311043	-0.6039353	-0.6820440
DF	0.006550085	-0.1610916	-0.2061691	0.1262835	1.000000000	0.8495716	0.6991073	0.5910958	0.5476098
CF	-0.208279170	-0.3622113	-0.3688601	-0.1660966	0.849571562	1.0000000	0.9119430	0.8216176	0.7727499
BF	-0.337948500	-0.4629301	-0.4404404	-0.4311043	0.699107255	0.9119430	1.0000000	0.9529627	0.9043409
AF	-0.389741017	-0.4927484	-0.4555423	-0.6039353	0.591095794	0.8216176	0.9529627	1.0000000	0.9706198
AN	-0.389358825	-0.4895267	-0.4466229	-0.6820440	0.547609821	0.7727499	0.9043409	0.9706198	1.0000000

Table 1 *Correlation Table Between Features*

Figure 3 and figure 4 show visual NDAI and SD values for clear and cloudy classes in according to x and y for image 1. From figure 3, we could see that points in the clear class seem to have lower NDAI values, while points in the cloudy class have relatively higher NDAI values. And the clear difference between two classes in terms of NDAI values suggests NDAI could be a pretty good classifier. From figure 4, points from two classes seem to have similar range and pattern for SD values, which demonstrates based only on SD values, the classification model will perform badly.

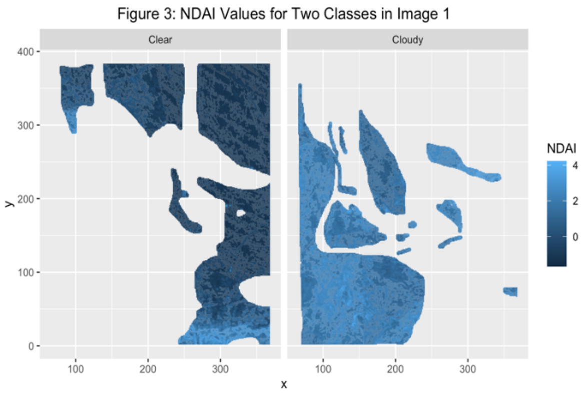


Figure 3: NDAI Values for Two Classes in Image 1

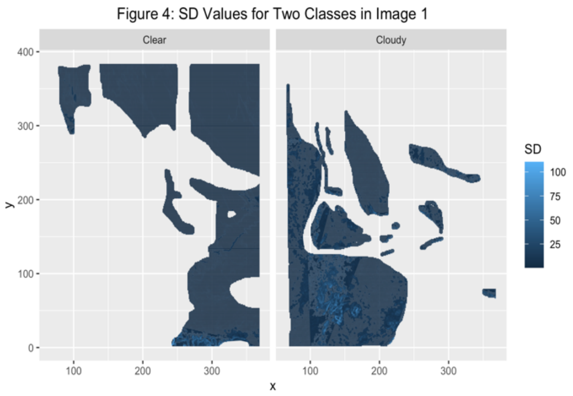


Figure 4: SD Values for Two Classes in Image 1

Figure 5 and figure 6 show visual CORR and DF values for two classes for image 2. For both two plots, two classes share similar patterns and ranges for CORR and DF values, which suggests they might not be strong classifiers.

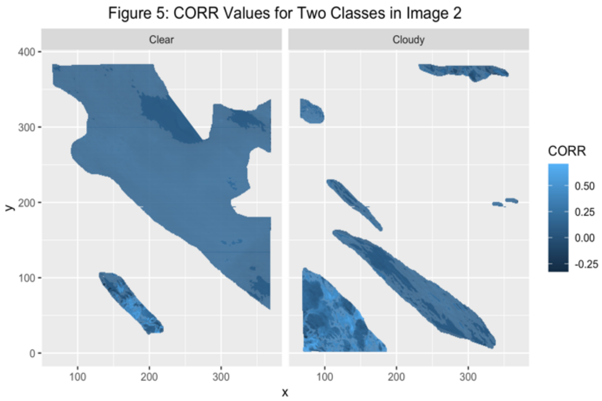


Figure 5: CORR Values for Two Classes in Image 2

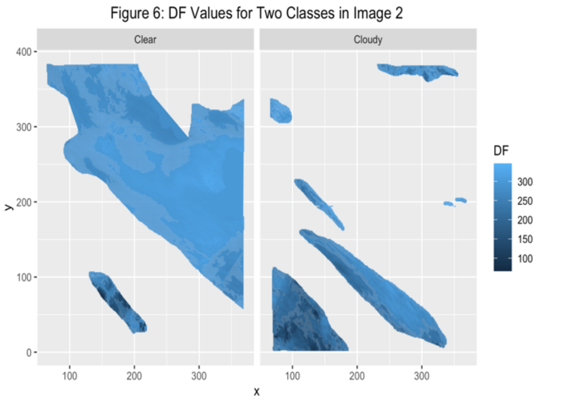


Figure 6: DF Values for Two Classes in Image 2

Figure 7 and figure 8 show visual CF and AF values for two classes for image 3. For both two plots, points in cloudy class have slightly smaller values of CF and AF since points in cloudy class have darker colors in the images. At the same time, points in cloudy class have specific pattern for the change of CF and AF values (change pattern for the color), while it is not the case for points in clear class. So, it might suggest they are informative for predicting the class.

Figure 7: CF Values for Two Classes in Image 3

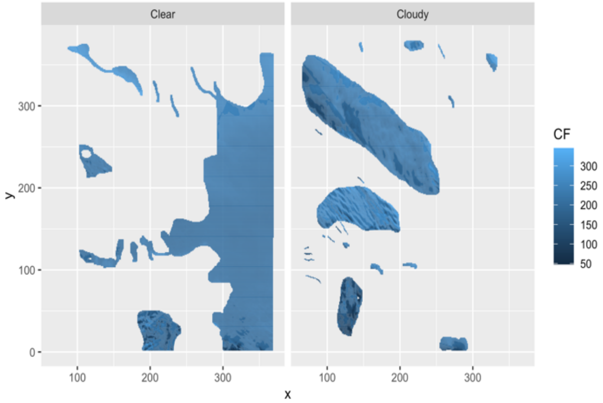
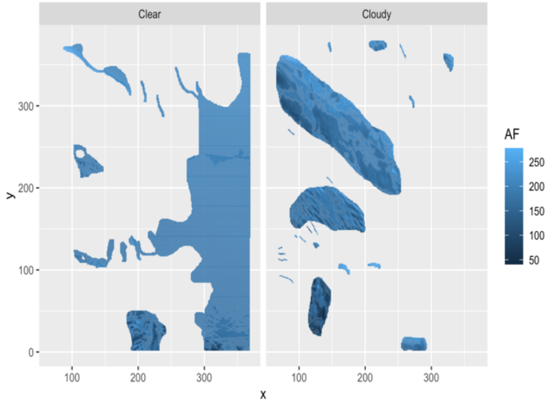


Figure 8: AF Values for Two Classes in Image 3



Because of the limited length for the paper, we didn't show graphs for all features for each image. Graphs above visually give the idea that good classifiers have different range or pattern of values for each class so that we could use them to predict the class for each point.

2. Preparation

a. Split the Entire Data

Since points are not identically independently distributed, it does not make sense to random pick points as training, validation and test dataset. Instead, we separated images into different blocks combining many points in the same regions as points contain information about nearby points.

For the first method, before separating blocks, we combined three images together as our complete dataset, and then separated it into 10 blocks in according to y axis by using the step size of the range of y divided by 10. We selected test dataset and validation dataset by randomly picking two indexes for blocks and used the whole block as test or validation. After selecting the test and validation, the remaining blocks were used as train dataset (1 block for test, 1 block for validation and remaining 8 blocks for train). The reason for us to separate images into 10 blocks is we want to have enough folds later for cross validation but at the same time we don't want to cut images into too small blocks and each block will have higher correlation. In addition, we didn't separate images in according to both x and y axis because we would like our blocks to include points from different classes as much as possible so that when we use trained model to predict the test data, the accuracy would be higher.

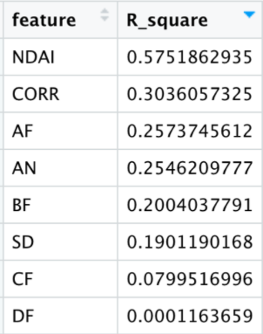
For the second method, we separated each image into 10 blocks in according to y axis. And we randomly pick 2 blocks from each image as our test dataset and validation dataset, and then combined them together as complete test dataset and validation dataset. All the remaining blocks were combined as our train dataset (total: 10x3 blocks, validation: 1x3 blocks, test: 1x3 blocks, train: 8x3 blocks). The advantage for the second method is that test dataset as well as validation dataset could include blocks in different regions in according to y axis, while for the first method, test dataset as well as validation dataset used three blocks (one big block) from three images in the exact same regions since we separated blocks using combined three images. This method could better show the accuracy and performance for trained model since we used more generalized blocks as validation dataset and test dataset.

b. Trivial Classifier

Trivial classifier was used as base classification model by setting all labels to -1 (cloud-free) on the validation set and on the test set. For the first method, the accuracy of the trivial classifier on the validation set is 0.3162 and accuracy on the test set is 0.3715. For the second method, the accuracy of the trivial classifier on the validation set is 0.2699 and accuracy on the test set is 0.552. When the validation dataset and test dataset include many cloud-free points, this trivial classifier has high accuracy. Or when we use the most frequentist class appeared in the validation dataset and test dataset as our trivial classifier, the accuracy for the classifier will be high.

c. Three 'Best' Features

Table 2 *R Squared Table for Features*



feature	R_square
NDAI	0.5751862935
CORR	0.3036057325
AF	0.2573745612
AN	0.2546209777
BF	0.2004037791
SD	0.1901190168
CF	0.0799516996
DF	0.0001163659

Before building classification models, we used simple linear regression model to glimpse what are the best three features to predict expert labels. By fitting features to the model one by one, r squared was used to see how much variation in expert labels the feature could explain in general. From table 2, NDAI, CORR and AF have relatively higher r squared values than the rest of features (r square for AN is really close to AF, it's also reasonable to include AN as three "best" features), which means they could explain more about expert labels than others.

Density plots and figures with expert labels were also used for finding three of the "best" features. And we found that NDAI, CORR and AF have best performance in predicting expert labels than the rest of features, which is consistent with what we found in table 2 as well as in table 1 (these three features also have relatively high correlations with expert labels). Because of limited length for the paper, we will only show the relative figures below.

Figure 9 shows the density plot for NDAI values with respect to two classes for three images. And we could see that points in two classes have different ranges and densities for NDAI values, which suggests that NDAI might be able to classify two classes since density plots for two classes don't have much area overlapped with each other. At the same time, figure 3 shows that points in clear area tend to have lower NDAI values while points in cloudy area tend to have higher NDAI values. So, both two plots could give us evidence that NDAI is a good classifier.

Figure 9: Density Plot for NDAI with respect to Expert Labels

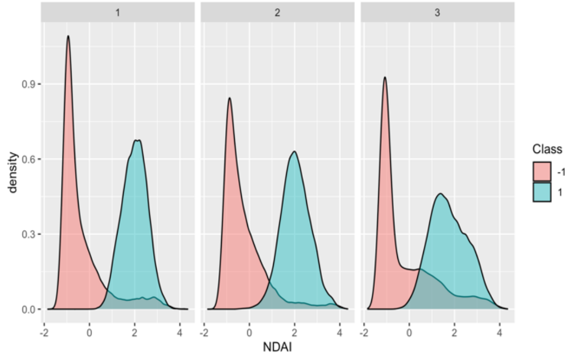


Figure 3: NDAI Values for Two Classes in Image 1

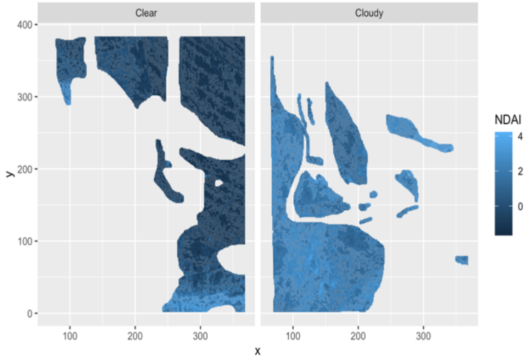


Figure 10 shows the density plot for AF values with respect to two classes for three images. Even though overlapped area for two density plots is larger than what we found in figure 9, they are still relatively small compared to other features, which suggests that AF is relatively a good classifier compared to others. In addition, figure 8 demonstrates there is specific pattern and range of AF values for points in cloudy area, which also suggests AF might be able to classify two classes.

Figure 10: Density Plot for AF with respect to Expert Labels

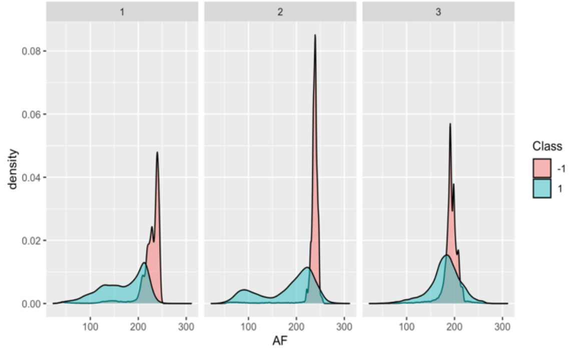
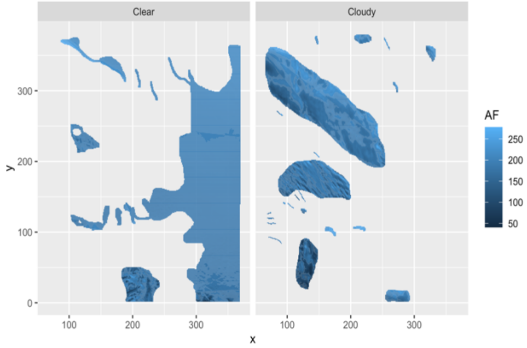


Figure 8: AF Values for Two Classes in Image 3



To show the difference between good classifiers and bad classifiers, we picked SD as our example. Figure 11 shows the density plot for SD values with respect to two classes for three images. And most of the area for density plots are overlapped with each other for two classes, at the same time, SD values with highest density for two classes are so close to each other (peak values for SD in the plot), which might show that the range as well as the pattern for SD values is similar for classes and it won't be able to classify them. What's more, figure 4 also shows that SD values for both clear region and cloudy region are similar and we couldn't use only SD values as the classifier to identify two classes.

Figure 11: Density Plot for SD with respect to Expert Labels

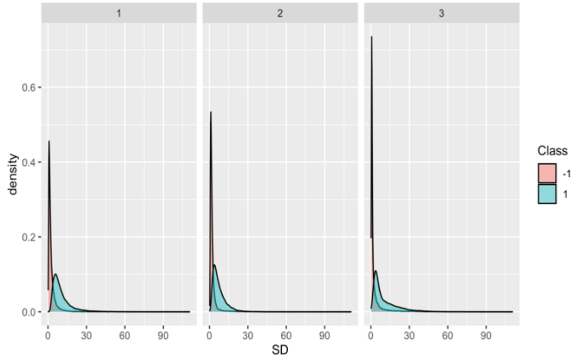
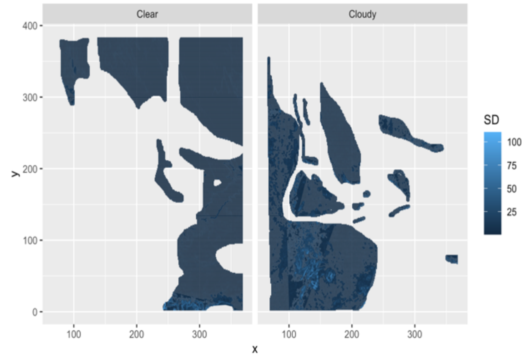


Figure 4: SD Values for Two Classes in Image 1



d. CVmaster

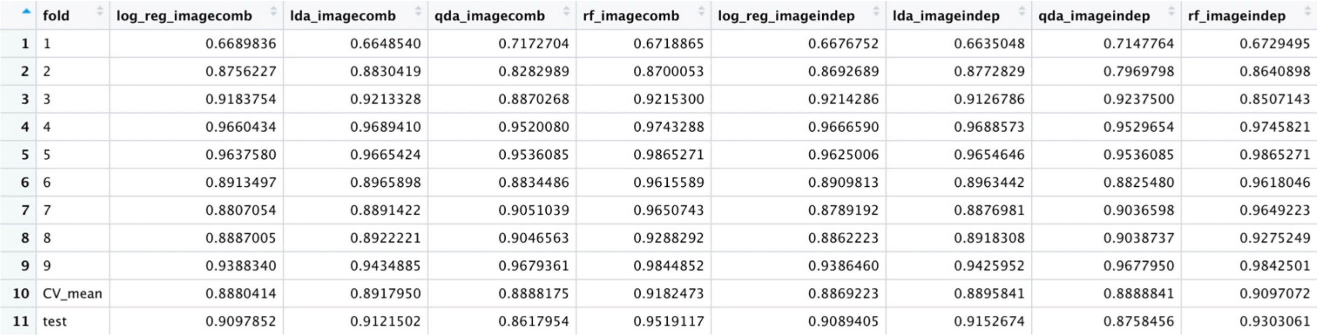
Specific code is included in CVmaster.r document and the details are explained in the README.md file.

3. Modeling

a. Classification Methods: Logistic Regression, LDA, QDA, Random Forest

We tried logistic regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Random Forest as our classification methods. Before assessing the fit for these models, there are some assumptions for each of them. For logistic regression model, at first, it requires the observations to be independent of each other, which is suitable for our blocks (the reason for us to separate image as blocks is individual points are not independent from each other). Secondly, it requires there to be little or no multicollinearity among the independent variables, which might be the concern for us since from table 1, features are correlated with each other. Thirdly, we need to make sure the relationship between independent variables and log odds is linear, which was not met for features (we plotted a graph using log odds as y-axis and feature values as x axis one by one to check linearity, but none of them has linear relationship with log odds). And For LDA, we assume that the joint distribution for features given labels is Gaussian distribution and we used multivariate normality test in the QuantPsyc library to check and found out that it’s unreasonable to assume it. What’s more, LDA requires features given labels to have common covariance matrix, which is not reasonable to assume (we used correlation matrix distance formula to check). For QDA, besides the Gaussian distribution assumption, there is no requirement for common covariance matrix. For random forest, there is no strong assumption, which is quite flexible.

Table 3 shows accuracies across folds and test accuracies for models we mentioned above in two different separating ways (in part 2a). From the table, random forest in two separating ways has highest CV accuracy means, while for the rest of the models, CV accuracy means are pretty close to each other, which is roughly 0.88. And random forest also has the highest test accuracies in two separating ways, followed by LDA and logistic regression. In addition, fold 1 has the lowest accuracies across all methods and we checked its features’ ranges as well as the regions included in fold 1, which will be examined in the 4th part. Accuracies for the remaining folds are similar. Furthermore, accuracies are not significantly different for two separating ways, which makes sense since we separated blocks in according to y axis, and the most important classifier NDAI has similar pattern and range across y axis so that there won’t be much difference between two separating ways (randomly pick blocks across y axis will not have much effect on predicting the class in this case). We also used CV to tune parameter for random forest, which will be explained in the 4th section.



	fold	log_reg_imagecomb	lda_imagecomb	qda_imagecomb	rf_imagecomb	log_reg_imageindep	lda_imageindep	qda_imageindep	rf_imageindep
1	1	0.6689836	0.6648540	0.7172704	0.6718865	0.6676752	0.6635048	0.7147764	0.6729495
2	2	0.8756227	0.8830419	0.8282989	0.8700053	0.8692689	0.8772829	0.7969798	0.8640898
3	3	0.9183754	0.9213328	0.8870268	0.9215300	0.9214286	0.9126786	0.9237500	0.8507143
4	4	0.9660434	0.9689410	0.9520080	0.9743288	0.9666590	0.9688573	0.9529654	0.9745821
5	5	0.9637580	0.9665424	0.9536085	0.9865271	0.9625006	0.9654646	0.9536085	0.9865271
6	6	0.8913497	0.8965898	0.8834486	0.9615589	0.8909813	0.8963442	0.8825480	0.9618046
7	7	0.8807054	0.8891422	0.9051039	0.9650743	0.8789192	0.8876981	0.9036598	0.9649223
8	8	0.8887005	0.8922221	0.9046563	0.9288292	0.8862223	0.8918308	0.9038737	0.9275249
9	9	0.9388340	0.9434885	0.9679361	0.9844852	0.9386460	0.9425952	0.9677950	0.9842501
10	CV_mean	0.8880414	0.8917950	0.8888175	0.9182473	0.8869223	0.8895841	0.8888841	0.9097072
11	test	0.9097852	0.9121502	0.8617954	0.9519117	0.9089405	0.9152674	0.8758456	0.9303061

Table 3 *CV-results and Test Accuracies for Different Models in Two Separating Ways*

b. ROC Curves

Figure 12 and figure 13 show ROC curves for all 4 methods in according to different separating methods on the training dataset. From two plots, random forest has the best performance, whose ROC curve is the closet one to the best theoretical ROC curve, followed by QDA. Since we care more about classifying cloudy area (true positive) correctly, the cutoff value with highest true positive rate is chose as the most optimal cutoff. Cutoff values for random forest, logistic regression and LDA are similar (roughly 0.39), while the cutoff value for QDA is pretty low (roughly 0.039), which might be related to imbalanced dataset.

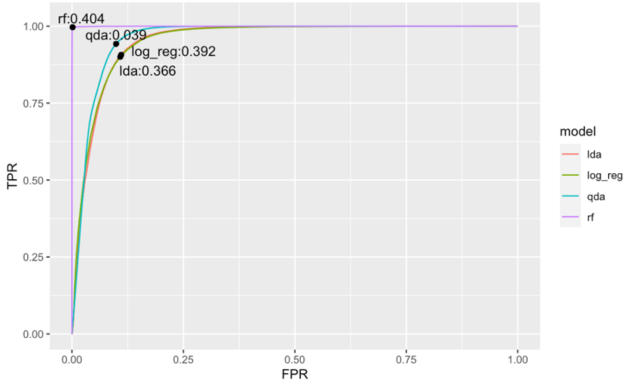


Figure 12 *ROC Curve for Training Data (First Separating Method)*

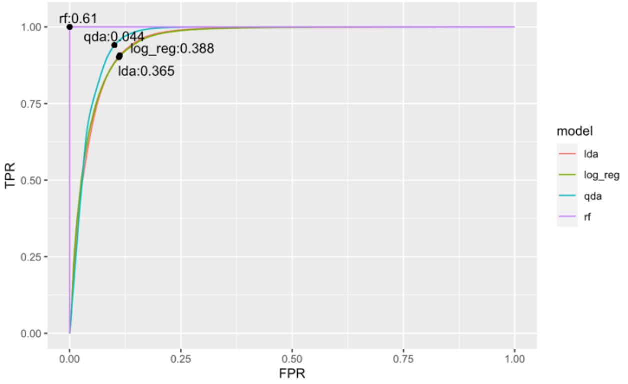


Figure 13 *ROC Curve for Training Data (Second Separating Method)*

Figure 14 and figure 15 show ROC curves for all 4 methods in according to different separating methods on the test dataset. Performance on the test dataset is slightly worse than it on the training dataset, and the difference between two separating methods is subtle. In addition, except for random forest, ROC curves on the test dataset are more similar for the rest of the methods compared to it on the training dataset.

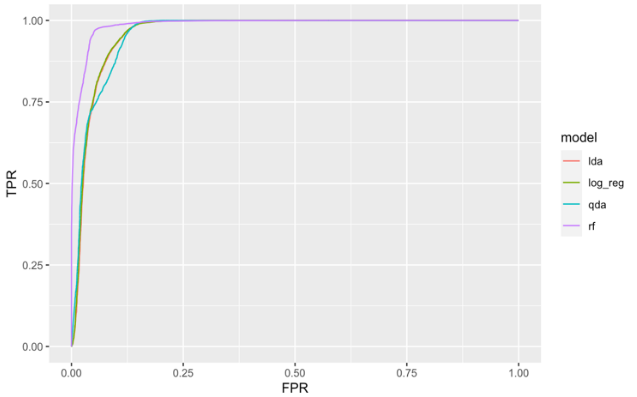


Figure 14 *ROC Curve for Test Data (First Separating Method)*

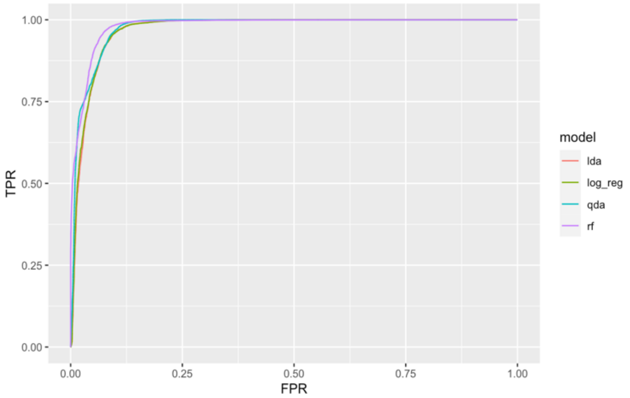


Figure 15 *ROC Curve for Test Data (Second Separating Method)*

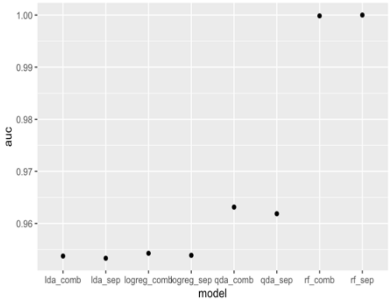
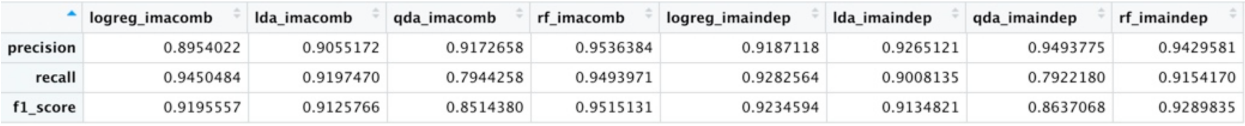


Figure 16 *AUC for All Methods*

c. AUC, Precision, Recall and F1 Score

Figure 16 shows area under the curve (AUC) on the test data for all four methods in two separating ways. From the plot, random forest has the highest values for two separating methods, which indicates random forest has the best performance, followed by QDA, and the rest of methods have similar AUC values. Table 4 shows the precision, recall rate and F1 score for all the methods. Depending on different emphasis of the study, best model could be identified using different criteria. When we want False-Negative rate to be as low as possible, precision could be low, but recall should be high. When we want False-Positive rate to be as low as possible, model with high precision should be chose. Furthermore, F1 score is the harmonic mean of precision and recall and it takes both false positive and false negatives into account so that it performs well on an imbalanced dataset. From table 4, random forest using the first separating method and logistic regression using the first separating method are the best two models when we care more about False-Negatives, while QDA using the second separating method and random forest using the first separating method are the best two models when we care more about False-Positives. And when we care both False-Negatives and False-Positives, random forest using the first method is the best model, whose F1 score is the highest (roughly 0.95).



	logreg_imacomb	lda_imacomb	qda_imacomb	rf_imacomb	logreg_imaindep	lda_imaindep	qda_imaindep	rf_imaindep
precision	0.8954022	0.9055172	0.9172658	0.9536384	0.9187118	0.9265121	0.9493775	0.9429581
recall	0.9450484	0.9197470	0.7944258	0.9493971	0.9282564	0.9008135	0.7922180	0.9154170
f1_score	0.9195557	0.9125766	0.8514380	0.9515131	0.9234594	0.9134821	0.8637068	0.9289835

Table 4 *Precision, Recall and F1 Score*

Figure 17 shows predicted labeled images with black color on wrongly predicted regions for four methods. From the plots, we could see that random forest has the smallest wrongly predicted regions for all three images, which is consistent with what we found in previous sections (random forest has the best performance). QDA has slightly less wrongly predicted region compared to LDA and logistic regression. What's more, wrongly predicted regions for logistic regression, LDA and QDA are similar. The reason might be for the corresponding regions, we have less cloudy points to train, which results in high misclassification error rate for the specific cloudy regions.

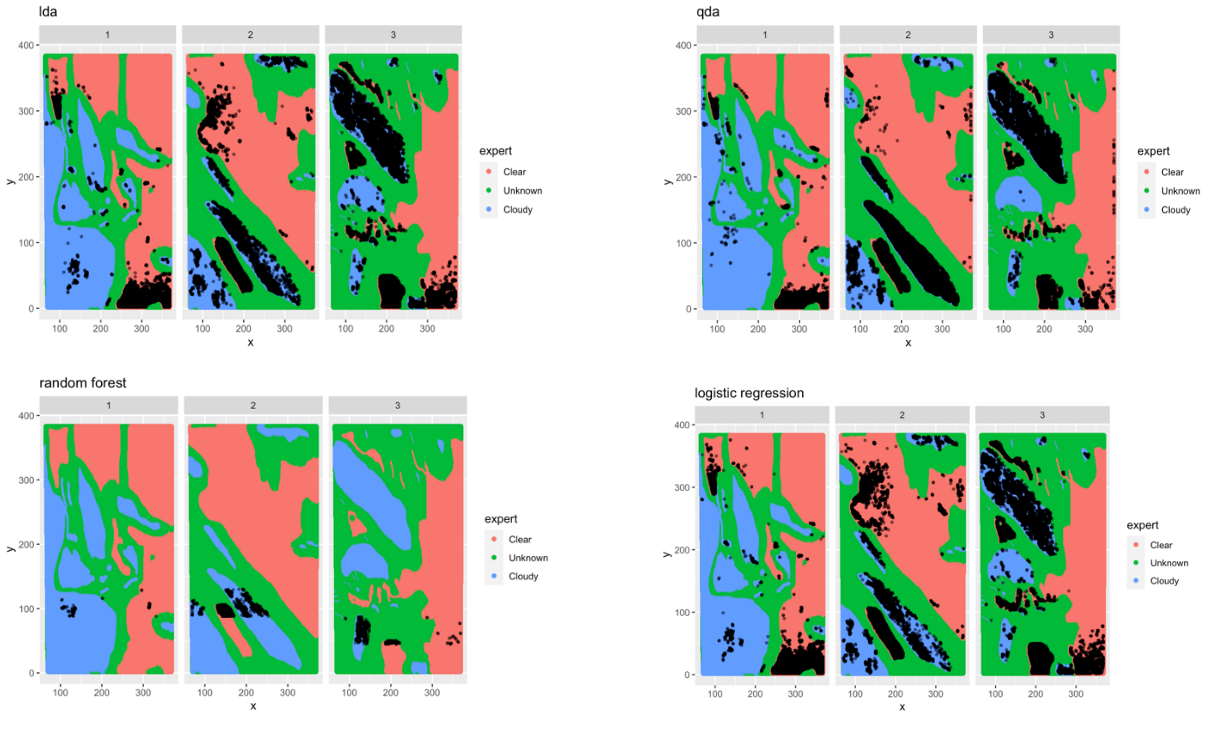


Figure 17 *Predicted Labeled Area*

4. Diagnostics

a. Diagnostics for Random Forest

We can show the performance of a good classifier, random forest, from the perspective of optimization perspective, model complexity perspective, and interpretation perspective. The classifier is trained and examined using the second separating method, which means the image is treated separately.

In terms of optimization perspective, Figure 19 shows that with the number of trees increasing, both out of bag error and error rate for clear and cloudy regions decrease, and finally converge to a stable value. The out of bag error and the error rate of clear region respectively converge to 0.037 and 0.039, and respectively converges at 40 and 100 trees. The error rate of cloudy region converges to an even smaller value 0.031, and achieves convergence at about 200 trees. The result is encouraging because we are expected to predict cloudy regions correctly. From this figure, we can also conclude that the convergence is achieved well at about 300 trees and thus we can stop the algorithm earlier.

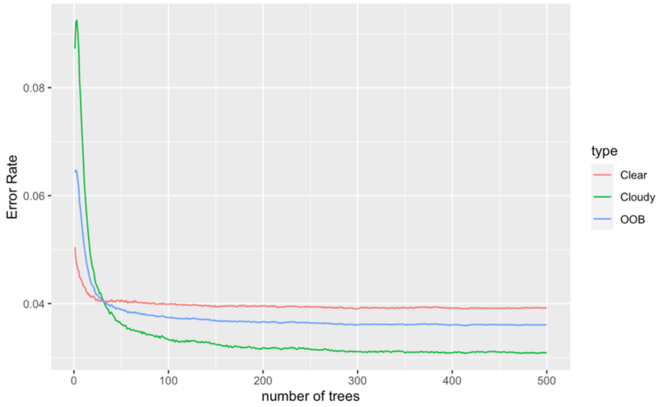


Figure 19 *Out of Bag Error and Error Rate for Clear and Cloud region of different number of trees*

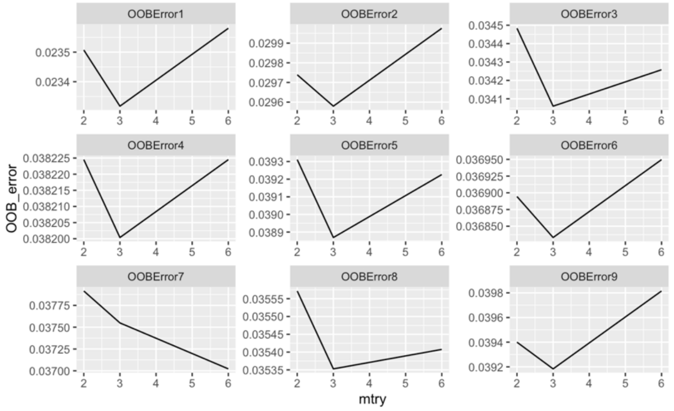


Figure 20 *Out of Bag Error for different number of features used in each split with 9 folds cross validation. Each image denotes one fold.*

In terms of model complexity and bias-variance trade-off, we obtain out of bag error across different mtry parameter (which denotes the number of features used in each split), using 9 folds cross validation, as shown in Figure 20. In 8 out of 9 folds, 3 mtry has the least OOB Error. The mtry parameter roughly represents the model complexity, when it's too small, bias will be large and model is underfitting, so that OOB error will be large. When it's too large, variance will be large and model is overfitting, and OOB error will also be large. In this case, we choose 3 mtry with smallest OOB error as our hyperparameter to train our final random forest.

As for model interpretation, Figure 21 is the variable importance plot. In terms of both accuracy decreasing and Gini index decreasing, NDAI is always the most important feature, followed by SD and CORR. The other features are comparatively weak in terms of classification. It's consistent with the analysis in Section1 c.

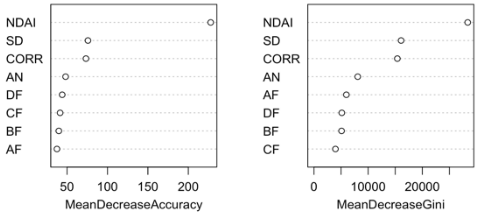


Figure 21 *Feature Importance regarding Accuracy, Gini Index*

b. Pattern in Misclassification

As shown in Figure 17 in Section3 c., when predicting the whole three images using random forest, the misclassification points cluster in the region with y coordinate ranging from 70 to 120, and more cloudy area than clear one. From the original image, we can see that in this region, there are many “unknown” areas. As we train random forest after dropping “unknown” points, we do not have enough training samples in this block, and thus the misclassification rate is high. Besides, the label of Cloudy and Clear in this region is unbalanced, which might lead to misclassification in prediction.

We also examine feature values in high misclassification area by calculating summary statistics separately about misclassification area and correctly classified area. When predicting whole images, the most differentiable feature is NDAI: for wrong area, NDAI ranges from -0.47 to 4.03 with mean 2.07 and median 2.04, but for correctly classified area, it ranges from -1.84 to 4.56 with mean 1.08 and median 1.34, much smaller than that of wrong areas. When using CV, we also notice that accuracy of using fold one as validation set is significantly lower than others. The value of features shows that median of NDAI in fold 1 is 2.08 versus -0.1 in others. Mean of SD in fold 1 is 10.92 versus 4.76 in others. Both are significantly higher in fold 1. As these two are our most important feature for prediction, it is not surprising these areas are misclassified with such different values.

c. Better Classifier

First, a better classifier can take advantages of regularization techniques to avoid overfitting. We can add L1 or L2 penalty on LDA, QDA and logistic regression, and the regularization constant can be tuned using CV. We can also use early stopping with cross validation, which means stopping when the performance on validation set decreases. For random forest, we can use this technique at an appropriate number of trees. Regularization technique will help to improve performance on future data without expert labels, because of its merits on avoiding overfitting and controlling bias-variance trade-off. As our random forest works well on test dataset, we suppose that it will work well on future data in this spatial location, but it can work better if we use more regularization techniques. However, if spatial location is changed, the classifier might perform not very well, since all our images are taken at the same location, and the classifiers might only capture some characteristic special to this location.

Second, a totally new classifier can be constructed using neural networks. Neural network classifier can identify complex patterns of data and build different kinds of non-linear boundaries. Neural network can classify using different types of data, including tabular data, images, texts, and sounds, while random forest might not be appropriate for some types. Neural network is more flexible, but whether it is better than other classifiers still depends on a specific problem. Regularization technique including dropout can be used with neural network to avoid overfitting. If the hyperparameters are chosen appropriately, neural network may work well on future data.

d. Influence of Splitting Data

We repeated the procedure in a. and b. using the first separating method, which treats three images as a whole. The results do no change much, so we won’t show the resulting figures one more. It is reasonable since we split the data into blocks according to its y coordinate, and the only difference of two methods is whether to treat three images as a whole or independently. It means testing data might be slightly different in these two methods but training data are not changed dramatically (one block only contains one out of ten of our data). For the first method, testing data are composed of features in different time in the same spatial location (different image is taken in the same location but at different time). For the second method, testing data are composed of features in both different time and different spatial location. We examine our testing data, the first two images coincidently have same testing blocks, while the third has a different one. It means that the difference in two methods is even smaller since the first two images still have same testing blocks. As shown in Table 3, classifiers only work not very well on the first two folds, which has a specific range of y coordinate. We find out that none of these badly predicted areas are in our testing data. We also find out that features are not significantly different across y, other than these two badly predicted areas in cross validation, so the classifiers will not perform dramatically differently using almost same training dataset and almost same characteristics. We can conclude that results will not change too much in a. and b. with different splitting methods, where the random forest still converges at about 200 trees, the best mtry is still 3, the important features are sill NDAI, SD and CORR, and the worst predicted area is still the first two folds in CV.

If we use a totally different splitting method, the result might change a lot. For example, if we split the data according to both x and y coordinates, it might be worse than only using y coordinate. It might because of imbalanced labels across x coordinate, that is also why we do not choose this splitting method.

e. Conclusion

In summary, the best classification model within logistic regression, LDA, QDA and random forest in this cloud identification problem is the most flexible one: random forest with hyperparameter mtry (number of variables using in each split) equals to 3. It is best in terms of accuracy, AUC, precision and other diagnostic statistics, but it is the least computationally efficient one. The two ways of splitting data, which treat images as a combination or independent part, but both divide them into blocks according to y, does not affect the result too much.

In terms of summary statistics for classifiers, the accuracy of testing dataset for random forest is above 93%, while that for others range from 86% to 91%. The training AUC (area under ROC curve) of random forest is 1, a lot higher than other classifiers, while its testing AUC is slightly higher than others and still the best. The precision, recall and f1 score of random forest are all up to 0.95, and higher than all other classifiers, while that if QDA are only around 0.83. It means that the true boundary of cloudy and clear regions may not be linear and simply quadratic in this case, and data does not fit the assumption of other classifiers. It is also confirmed by the fact that LDA and logistic regression perform similarly in this case.

However, the best model should be chosen with different criteria specifically in different problems, since different researches emphasize on different aspects. Since we emphasize on identifying cloudy areas correctly, random forest with highest precision and other good performance should be chosen as the best classifier.

References

Shi, Tao, et al. “Daytime Arctic Cloud Detection Based on Multi-angle Satellite Data with Case Studies.” Journal of the American Statistical Association 103.482 (2008): 584-593.