Project Report

Ruizhi Yuan

November 2022

Contents

1 Introduction

In the Arctic, increasing carbon dioxide level is highly correlated with surface air temperature. As the Arctic warms, clouds can potentially lead to further warming and thus increasing the amount of atmospheric carbon dioxide. So detecting cloud-covered surfaces is important in predicting global climate and regulating the climate and temperature in the Arctic. Due to the similarity between infrared electromagnetic radiation emanating from clouds and snow- and ice-covered surfaces, the existing Multiangle Imaging Spectro Radiometer (MISR) cloud detection algorithms (L2TC, SDCM, and ASCM) were not useful in polar regions. Therefore, to increase the effectiveness of cloud detection, Shi et al. (2008) developed two new operational Arctic cloud detection algorithms which combined classification and clustering schemes. The purpose of the paper is to identify the cloud-free surface pixels in the imagery which enables us to understand the relationship between cloud cover and Arctic climate change.

The data was collected from 10 MISR orbits of path 26 (rich in its surface features) over the Arctic, northern Greenland, and Baffin Bay and span approximately 144 days (a daylight season in the Arctic), including 6 data units from each orbit. 3 of the 60 data units were excluded because the surface is open water that does not use the algorithm. 71.5% of the data was hand-labeled by an expert conservatively, with white for cloudy, gray for clear, and black for unlabeled. This paper uses these labels to evaluate the performance of different cloud detection algorithms.

The conclusion of the paper is that through the use of three physical features — CORR, SD, and NDAI constructed by the authors, the ELCM algorithm is more accurate and provides better spatial coverage than the existing MISR operational algorithms for cloud detection in the Arctic. The ELCM algorithm combines classification and clustering frameworks in a way that makes it suitable for real-time, operational MISR data processing, and the results can be used to train QDA to provide probability labels for partly cloudy scenes. This paper also found that these three features provided sufficient separability and stability to separate clear (cloud-free) regions from clouds providing performance comparable to that of much more sophisticated classifiers. These studies will finally solve one of the biggest puzzles in science and demonstrate the power of statistical thinking and also the ability of statistics to contribute solutions to modern scientific problems.

2 Exploratory Data Analysis

To learn the distribution of three expert labels ("No Cloud = -1", "Cloud = 1", "Unlabeled = 0") in three images and in total, I first build a table to show the percentage of pixels for different classes. As can be seen from the table 1, the percentages of the three expert labels in total are not equal proportions. Image 1 has similar percentages of "No Cloud" and "Cloud" pixels, Image 2 has a huge percentage of "No Cloud" pixels, which is 43.78%, and image 3 has slightly more than half of the pixels labeled as "Unlabeled". I also calculated the number of rows in data over three expert labels, which were 127080, and 80981,137495 respectively.

Label	Image1	Image2	Image3	Total	Rows
No cloud	37.25%	43.78%	29.29%	36.78%	127080
Unlabeled	28.64%	38.46%	52.27%	39.79%	137495
Cloud	34.11%	17.77%	18.44%	23.43%	80981

Table 1: Percentages of Pixels for the Different Images

I also draw well-labeled beautiful maps using x, and y coordinates with light yellow for "Cloud", medium yellow for "No Cloud", and dark yellow for "Unlabeled". In figure 1, I can see that the unlabeled areas separate the cloud areas and no cloud areas, which shows it has strong space dependence, and clouds tend to form in distinct clumps. Since the fact that a pixel in the dataset is cloud suggests that nearby pixels have a relatively high probability of being cloud, therefore, I cannot assume that the pixels in this dataset are independent and identically distributed (i.i.d).

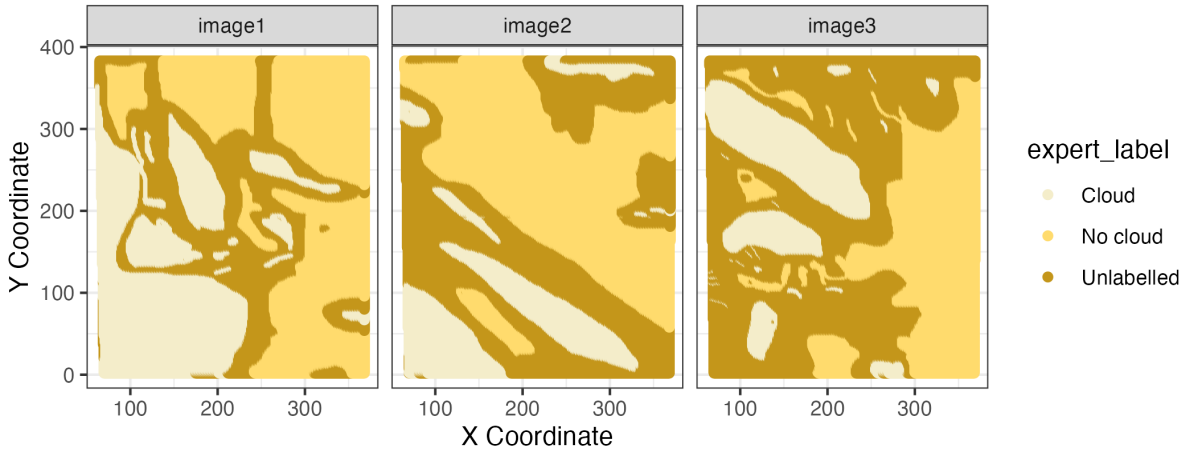


Figure 1: Well-Labeled Beautiful Map

To get the relationship between predictors and features, I first draw a correlation plot on 8 features and expert label to explore the pairwise relationship between these features and expert label. I can see from figure 2 that NDAI, CORR, and SD are positively correlated with expert label, radiance angle DF is almost not correlated with expert label, and other radiance angle features are negatively correlated with expert label. Moreover, I observe high positive correlations between radiance angle features because these features measure the solar spectral reflectances at different angles. I also find that CORR, SD, and NDAI are positively correlated.

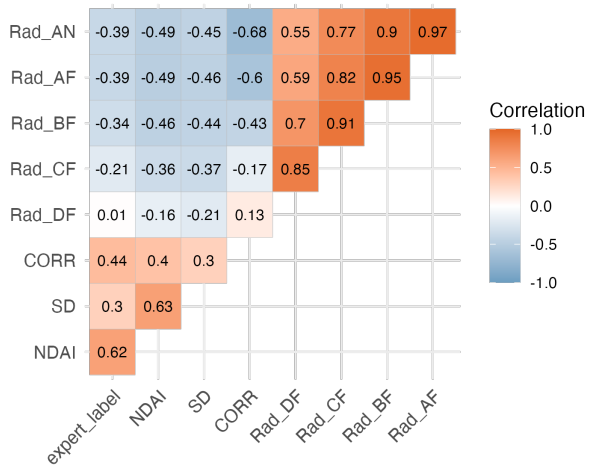


Figure 2: Pairwise Correlation Plot

To get the relationship between the expert labels with the individual features, I then draw the density plots of 8 features over the expert label (Cloud and No Cloud). From feature SD's plot, I find obvious right-skewed distribution. So I further conduct a log transformation on SD to make data conform to normality. As can be seen in figure 3, I can observe clear differences in mean values of cloud and no cloud class in feature CORR, log(SD), and NDAI. Except for Rad_DF, "no cloud" shows higher mean values than the cloud in radiance angle features. In addition, the distributions of radiance angle features, CORR, and NDAI in no cloud are tighter than that in cloud.

In section 3, I will build models with all 8 features(using log(SD) rather than SD)) because I think deleting some features will lead to accuracy reduction.

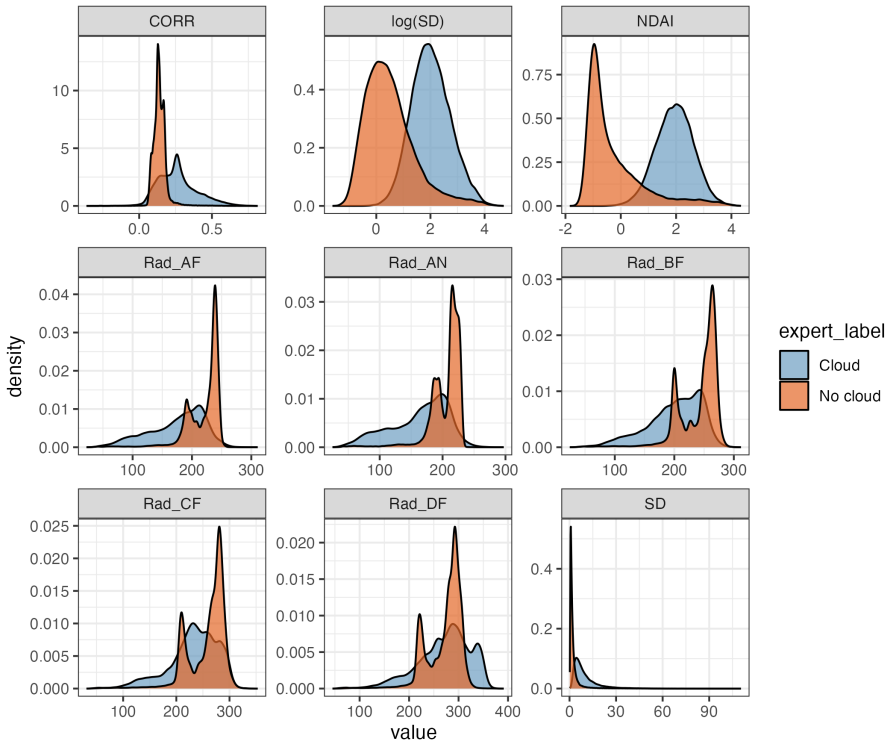


Figure 3: Relationship between the Expert Labels with the Individual Features

3 Data Preparation

3.1 Data Split

Based on my observation in 2, the pixels have spacial dependencies, which means a cloud pixel in the dataset suggests that nearby pixels have a relatively high probability of being cloud. So it is not appropriate to use traditional data splitting methods that treat pixels i.i.d. in this study. Block cross-validation and k-means consider the spacial information and pixels dependency. So I choose to split the dataset into 10 blocks for block splitting method and 10 clusters for k-means method because I will use 10 folds cross-validation in the modeling section.

In this section, I apply one cutting-block method and k-means to split the entire dataset (imagem1.txt, imagem2.txt, imagem3.txt) into three sets: training, validation and test. I first randomly split 20% of the data as test set. Then I use horizontal lines to cut the images into several block. For method 1, I don't cut x coordinate and cut y coordinate into 10 equal length and make 2 random blocks as the validation set and 8 as the training set; I call this method horizontal block splitting method. For method 2, I perform k-means clustering on the image and set the number of clusters as 10. In summary, the proportion of training: test: validation set is 64%: 20%: 16%. In the section 3, I merge training and validation set as a training set to do cross-validation.

3.2 Baseline model

In this section, I report the accuracy of a trivial classifier. I set all expert labels to -1("No Cloud") as a baseline model which is done on both the validation set and the test set using three data splitting methods from section 3.1. Table 2 shows the accuracy of this baseline model. I observe the test accuracy of horizontal block is up to 67.18%, which might due to the test set choose blocks whose expert labels are almost -1("No Cloud"). Other accuracies fall in the range of 50% - 65%. I speculate that the baseline model will have high average accuracy when the data is imbalanced. When the data's expert labels are almost all -1, the baseline model will predict very perfectly. So the model will have high accuracy.

Splitting Method	Set	Accuracy
Horizontal Block	Validation	59.45%
Horizontal Block	Test	67.18%
K-Means	Validation	64.06%
K-Means	Test	60.32%

Table 2: Accuracy of a trivial classifier.

3.3 First Order Importance

In this section, I use both visualization and quantitative justification to find out the most important features. From density plots in figure 3, mean values of CORR, log(SD), and NADI are different over cloud and no cloud. These three features might be highly important during the modeling. In contrast, randiance angle features have similar distribution over cloud and no cloud, I speculate that these 5 features might have equal importance. I further test these features' importance quantitatively.

For classification, ROC curve analysis is conducted on each features For this two-class problem, a series of cutoffs is applied to the feature data to predict the cloud and no cloud. The sensitivity and specificity are computed for each cutoff and the ROC curve is computed. The trapezoidal rule is used to compute the area under the ROC curve. This area is used as the measure of feature importance. The feature importance is shown in figure 4, two methods of cutting block are corresponding to Horizontal Block and K-Means; I can observe that all methods have shown similar result. NADI, log(SD) and CORR show significant great feature importance compared to radiance angle features. This result prove my speculation that these three features might be highly important.

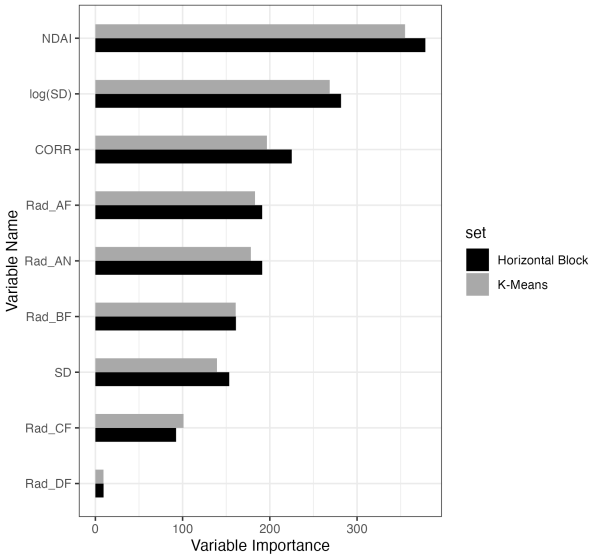


Figure 4: Feature Importance

3.4 Block (Spatial) cross validation

In this section, in order to use my special splitting algorithm for training, I write a generic Block (spatial) cross validation function named **CVmaster** . It takes a generic classifier (Logistic Regression, Random Forest etc.), method of splitting the data(Horizontal Block, Kmeans), training features, tuning parameters, training labels, number of folds K and a loss function (Accuracy, Precision etc.) as inputs and outputs the K-fold CV loss on the training set. Notice that the output is a list of all the metrics I use if I do not input any loss function. Then I apply this function in section 4 to do cross validation modeling and tune parameters so I can find out best parameters of my models using the Accuracy metric. My CVmaster function can also output other metrics such as Precision, F1 score, which I use to access model performance on test set.

4 Methods and Modeling

4.1 Try 5 Classifiers

I now try to fit different classification models and assess the fitted models using different metrics and methods. Although I find out three "best" features, in order to obtain the best predictive performance, I choose to retain these 8 features. For the next three parts, I expect to try Logistic Regression and four other methods - LDA, QDA, Decision Tree, and Random Forest to make classification. I develop these five classifiers because of the large number of pixels and limited computing resources, they are easy to train, penalize, and tuning parameters. I choose Random Forest because it is one of the best off-the-shelf methods. The training and test data I use are created by two methods in data splitting section and I also merge training and validation set to fit models. For each classifier, I do 10-fold cross validation in the two separate scenarios, data split using horizontal block and k_means. I choose to use 10 folds as the choice of k is usually 5 or 10.

I now talk about whether or not this case satisfies the various methods' assumptions as follows:

Logistic Regression: y_i, the response expert label, follows Bernoulli distribution, also it conditionally independent given the predictors. They cannot have strong multicolllinearity. Because I know the response has spatial dependency, it can not satisfied the assumption if it exists multicolllinearity. However, I will try this method because it will give us a model that can be easily interpreted

Linear Discriminant Analysis: The model assumes that the data within each class has a common covariance matrix and normal distribution. From the section 2, I can see the distribution and find that the normality assumption that has constant variance is not satisfied. I choose it because it is a simple and efficient algorithm that can be trained quickly, even on large datasets.

Quadratic Discriminant Analysis: The model assumes the data within each class has normal distribution, and each covariance matrix is separate and different. From the section1, the normality assumption is not satisfied. I choose it because it can achieve high classification accuracy on linearly and non-linearly separable data.

Decision Tree: decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. I choose it because they are easy to understand and interpret, and are robust to outliers and can handle missing data.

Random Forest: Random forest is a supervised learning algorithm that is used for classification and regression. It is an ensemble learning method, meaning that it combines the predictions of multiple individual models to produce a more accurate and stable prediction. The "forest" in random forest refers to the multiple decision trees that are trained and used in the model. Each tree in the forest is trained on a different subset of the training data, and the final prediction is made by averaging the predictions of all the trees. This creates a more robust and accurate model than using a single decision tree. I choose to use it because 1.They can handle large datasets with high dimensionality, making them suitable for complex problems with many features; 2.They are highly accurate and robust, and can perform well on both classification and regression tasks; 3. They can be used for feature selection, identifying the most important variables for making predictions.

From figure 3, I can see that the assumption of LDA and QDA is not effective because both of the cloud data and no cloud data do not follow the normal distribution. For Logistic Regression, the features have strong multicollinearity as shown in figure 2. Although the assumptions are not satisfied in these 3 methods, I can still use them to make prediction and do further analysis.

For Logistic Regression, Random Forest, and Decision Tree, I tune hyper parameters to find out the best parameter. These parameters are chosen based on a number of ad-hoc tests that convinced us that the best parameter is located within the range I choose. Next I carefully explain the choice of parameters.

For λ parameter in Logistic Regression, which controls the amount of regularization applied to the model, I choose range c(0,0.0005, seq(0.001,0.01,0.001)) to tune parameter.

For **mtry** parameter in Random Forest, which means the number of variables to randomly sample as candidates at each split. Since the dataset only contains 8 features, I choose the range (1, 8, by=1) for mtry.

For **CP** in Decision Tree, which means complexity parameter and is used to control the size of the decision tree and to select the optimal tree size, I experiment with a wide range of c(0,0.0005, seq(0.001,0.03,0.001)) values.

I show the results of Logistic Regression, Random Forest and Decision Tree using their best parameter in table 3 and table 4 for both the ways of creating folds, and the test set accuracies are shown in table 5 and table 6. I can clearly see that when using k_means split method, **Random Forest** and **Decision Tree** obtain the same highest average

Classifiers	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
Logistic Rgression ($\lambda = 0.009$)	0.683	0.885	0.924	0.953	0.997	0.960	0.909	0.855	0.866	0.949	0.898
Random Forest (mtry = 6)	0.761	0.884	0.920	0.939	0.998	0.970	0.975	0.961	0.930	0.983	**0.932**
Decision Tree (cp = 0.004)	0.717	0.872	0.928	0.962	0.997	0.967	0.940	0.932	0.907	0.968	0.919
QDA	0.741	0.896	0.926	0.955	0.994	0.961	0.940	0.936	0.896	0.962	0.920
LDA	0.678	0.884	0.924	0.944	0.995	0.962	0.915	0.864	0.873	0.947	0.899

Table 3: Accuracies across Folds and the Average using Horizontal Block Splitting Method

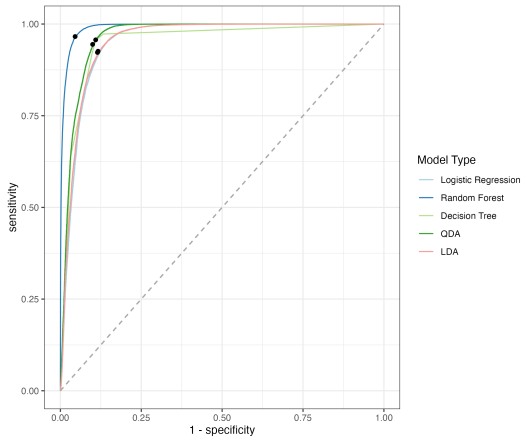
Classifiers	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
Logistic Rgression ($\lambda = 0.01$)	0.852	0.826	0.952	0.947	0.948	0.998	0.963	0.991	0.839	0.904	0.922
Random Forest (mtry = 6)	0.859	0.846	0.911	0.958	0.925	0.998	0.939	0.988	0.943	0.979	**0.935**
Decision Tree (cp = 5e-4)	0.871	0.824	0.931	0.944	0.942	0.997	0.963	0.991	0.919	0.970	**0.935**
QDA	0.875	0.824	0.959	0.967	0.955	0.996	0.951	0.995	0.924	0.896	0.934
LDA	0.864	0.826	0.953	0.935	0.941	0.997	0.953	0.992	0.842	0.844	0.915

Table 4: Accuracies across Folds and the Average using K-Means Splitting Method

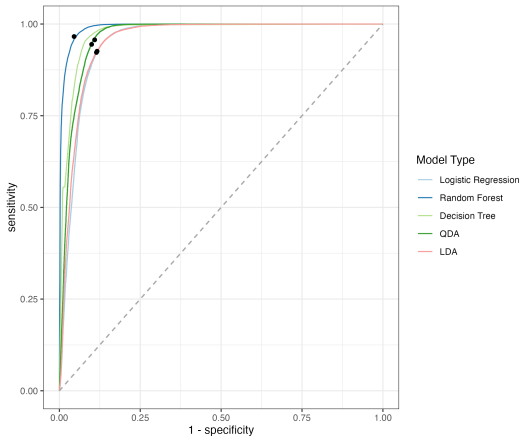
accuracy across 10 folds, which is **0.935**. Also, **Random Forest** obtains the highest test accuracy, which is 0.954. When using horizontal block splitting method, **Random Forest** obtains both the highest average accuracy across 10 folds and the highest test accuracy, which are **0.932** and **0.954** respectively. Therefore, I conclude that Random Forest performs best using two data splitting methods.

4.2 Results

I further compare the different models using the ROC curves. To get the curves, I first fit every model using training set for both the ways of creating folds. Figure 5 shows the ROC curves for the different methods with one probability threshold highlighted in black.



(a) ROC Curves for Different Classifiers in Horizontal Block Splitting Data.



(b) ROC Curves for Different Classifiers in K-Means Splitting Data.

Figure 5: ROC Curves

When making predictions on the test set, I need a threshold to determine the prediction results. I obtained the threshold values for the five classification methods by plotting the ROC of the five methods on the test set. In ROC curve graph (fighure 5, I can clearly see that the LDA curve and the Logistic Regression curve overlap, and some of the cutoff points overlap so that they look like a single line. At the same time, I can see from table 5 and table 6

that the random forest has the highest AUC value in both the horizon block split and the k-means split, indicating that it is the best model here in terms of its overall ability to distinguish between positive and negative classes.

Classifier	Test Accuracy	Precision	Recall	F1_Score	AUC	Threshold
Logistic Rgression ($\lambda = 0.009$)	0.893	0.97	0.855	0.907	0.952	0.265
Random Forest(mtry = 6)	**0.953**	0.982	0.947	0.964	**0.991**	0.401
Decision Tree (cp = 0.004)	0.916	0.969	0.891	0.929	0.95	0.432
QDA	0.916	0.973	0.887	0.928	0.965	0.019
LDA	0.896	0.957	0.868	0.911	0.954	0.213

Table 5: Metrics for Classifiers on Test Set using Horizontal Block Splitting Method

Classifier	Test Accuracy	Precision	Recall	F1_Score	AUC	Threshold
Logistic Rgression ($\lambda = 0.01$)	0.897	0.970	0.857	0.910	0.952	0.298
Random Forest(mtry = 2)	**0.954**	0.983	0.941	0.961	**0.991**	0.435
Decision Tree (cp = 5e-4)	0.933	0.969	0.920	0.944	0.975	0.351
QDA	0.918	0.978	0.885	0.929	0.967	0.030
LDA	0.898	0.965	0.864	0.912	0.925	0.234

Table 6: Metrics for Classifiers on Test Set using K-Means Splitting Method

To get the cutoff values, I then select the point on the ROC curve closest to the top left corner (sensitivity =1, 1-specificity =0) because it has the greatest true positive rate and the lowest false positive rate. I use Euclidien distance to get cutoff point and further the threshold value. The resultes of all thresholds are shown in table 5 and table 6. Compared with other 4 classifiers, random forest has the biggest threshold in k-means splitting method, the threshold values is 0.435 .Threshold values in random forest are 0.401 and 0.435 for two splitting methods..

I also choose Precision, Recall, F1 score, test AUC, and ROC cutoff value (Threshold) to show model performance in table 5 and table 6. In terms of those predicting positives, precision refers to how correct my model is. Recall provides an indication of missed positive predictions. F1 combines the precision and recall of a classifier into a single metric by taking their harmonic meanl. In F1 score, across two test splitting sets, random forest performs better than all the other classifiers. Decision Tree and QDA also perform well under both methods.

5 Insights

5.1 General insights

From the previous cross-validation, test accuracy, and other relevant metrics I can get random forest is the best model for this case. Although some parameterized methods did very well, the assumption of normality and independence make parameterized methods less attractive. I cannot directly evaluate the model convergence of model because the random forest is a method with non-parameter. Comparing two splitting methods to get the 8-feature importance graph (figure 6), I can see that the 3 features defined in the authors' article occupy the top three positions. As I predicted in section 2, NADI is the most important feature. Log(sd) is the third most important in the horizontal block splitting but becomes the second most important feature in the k-means splitting. Also, the small difference between the log(SD) and CORR rankings is not enough to indicate unfavorable convergence of my model. Generally, the feature importance ranking is the same across the different splitting methods. This indicates that the differences in segmentation do not affect the model's predictive power. The importance of all features is essentially concentrated in a similar distribution.

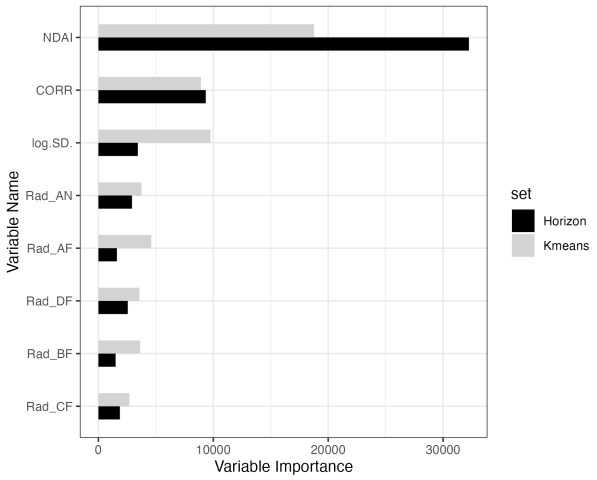


Figure 6: Feature importance using Random Forest

Then I evaluate the the predicted result of two different split methods and I can see whether the predictions converge or not. From the confusion matrix (table 8), the expert labels are shown in the rows and the predicted labels in the columns. In horizontal block splitting data, the overall accuracy is 0.953, sensitivity is 0.983, and specificity is 0.935. In k-means splitting data, the overall accuracy is 0.954, sensitivity is 0.975, and specificity is 0.940. This means that the model is making more mistakes when classifying Not Cloud pixels.

	True Cloud	True Not Cloud
Predict Cloud	23862	2496
Predict Not Cloud	424	35627

Table 7: Confusion Matrix using Horizon Block Split.

	True Cloud	True Not Cloud
Predict Cloud	23679	2281
Predict Not Cloud	607	35842

Table 8: Confusion Matrix using K-Means Split.

5.2 Patterns in the Misclassification

Firstly, I use two Random Forest models to predict all data and label the correct classification and misclassification by comparing expert label with prediction labels. To quantitatively explore the pattern in the misclassification errors, I draw boxplots for both ways of splitting data over 8 features. I can find that these two boxplot sets are similar in figure 7 and figure 8, so I analyze one of them in the following part for conciseness.

I can see obvious differences in mean values. The mean value of NDAI in misclassification is larger than the correct classification, while mean values tend to be smaller in misclassification for other features. In addition, the boxplots of 5 radiance angle features are similar, which indicates their positive correlation and corresponds to my findings in the EDA part. As for outliers, I find that the outliers of misclassification are concentrated near -1 in NDAI, so I speculate that small values of NDIA might lead to misclassifying.

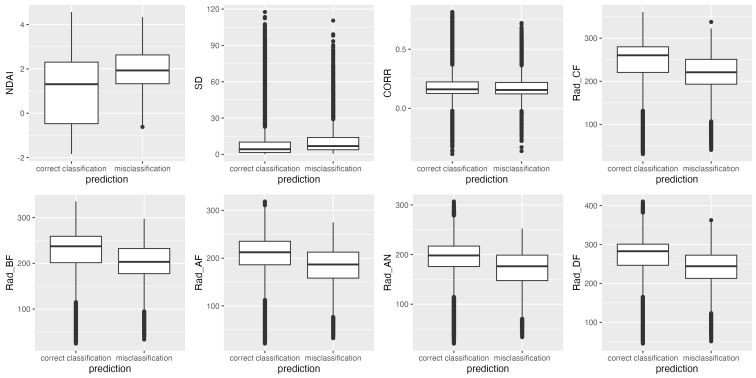


Figure 7: Boxplots of 8 Features Using Random Forest in Horizontal Block Split

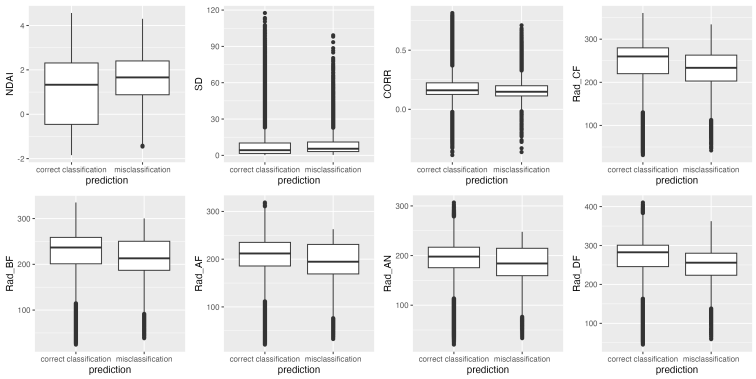


Figure 8: Boxplots of 8 Features Using Random Forest in Block Split

I use pixel plots of three images to visually study the patterns in the misclassification errors. Figure 9 and figure 10 show the misclassified points in the original image, making it easier for us to find the patterns of misclassified point locations. Comparing the original images, I can see that most of the misclassified data points are close to each other and should be classified as no cloud when they are clear. In contrast, these points are mainly distributed near the boundaries or small areas surrounded by unlabeled pixels. This is all based on the roc curve, the sensitivity of the random forest model is much higher than the specificity, so it correctly classifies almost all no cloud points, but there are also many false positives. In addition, when the data is decomposed into blocks or clusterings, the spatial dependencies within the blocks or clusterings are preserved but not well preserved on the boundaries. The points at the boundary lose some neighbors, which reduces the predictability of these points.

Of course, there are some similarities in the misclassification patterns of the two different splitting methods. For image 1, the misclassified points are mainly concentrated in the cloud-free area in the lower right corner. For image 2, misclassified points were mainly in the lower left corner of the unlabeled area. But in k-means split, the misclassified points also concentrate on top right corner. For image 3, the misclassified points of the two methods are mainly concentrated at the bottom of the cloud free area. Still, in the k-means split, misclassification points were a small part of the cloud free area in the upper right corner. In summary, k-means split shows more misclassified data points than horizontal block split.

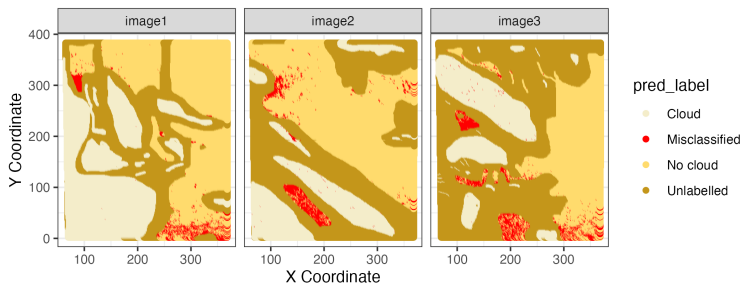


Figure 9: Misclassification Prediction using Random Forest in Horizontal Block Split

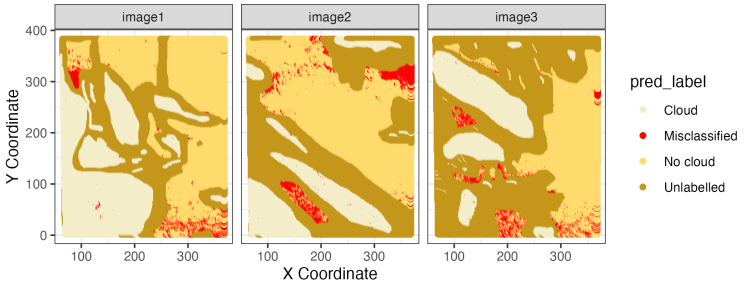


Figure 10: Misclassification Prediction using Random Forest in K-Means Split

5.3 Conclusion

In summary, given three images taken by the MISR camera, my goal is to build a classifier that can distinguish between pixels with and without clouds. I use two methods to split the dataset, the first one is horizontal block split, the second one is k-means. I use five different classification algorithms with tuning parameters to fit my data, namely logistic regression, random forest, decision tree, LDA and QDA.

In this project, I obtained the final model to create a best classifier using random forest and achieve high classification accuracy. I measure my model performance by ROC curve, precision, recall, F1 score, and AUC. To get the cutoff (Threshold), I select the point on the ROC curve closest to the top left corner I strongly recommend using random forest because it performs best in every metric I use and it is able to learn features well even if the features are highly correlated. In addition, I examined the error patterns of the model through diagnostics and error analysis, and provided some ideas for possible improvements.

Bibliography

Shi, Tao, Bin Yu, 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. https://doi.org/10.1198/016214507000001283.