Data Collection and Exploration

1.Summary of the paper

The goal of the study [1] is to build an operational cloud detection algorithm on massive MISR(Multiangle imaging spectroradiometer) data with high efficiency and accuracy. Unlike existing MISR algorithms, the proposed algorithms detect cloud-free pixels instead of cloudy or snowy pixels.

The data used in this study were orbits of path 26 MISR data, which has rich features across the Arctic Ocean to Greenland. The most important step is feature engineering to produce representative new features to summarize physical properties of MISR images: CORR as the correlation of the same scene from different directions; SDan as standard deviation of nadir camera pixel values across a scene; DNAI as normalized difference angular index that characterizes the changes in scene with changes in view directions. Two algorithms were invented based on the three new features. Enhanced Linear Correlation Matching (ELCM) algorithm uses CORR and SDan thresholds to label data units sequentially. The second algorithm ELCM-QDA uses labels from the ELCM algorithm to train Fisher's quadratic discriminant analysis to produce probability labels.

The study proposed an efficient cloud detection algorithm on MISR data, and it demonstrated the power of statistical approaches to solve complex scientific problems. The fruit of the research could also contribute to future global climate models.

2. Data Summary

Line plot of x, y axis, colored by label
Note: 1 denotes cloud, 0 denotes unlabelled, -1 denotes not cloud

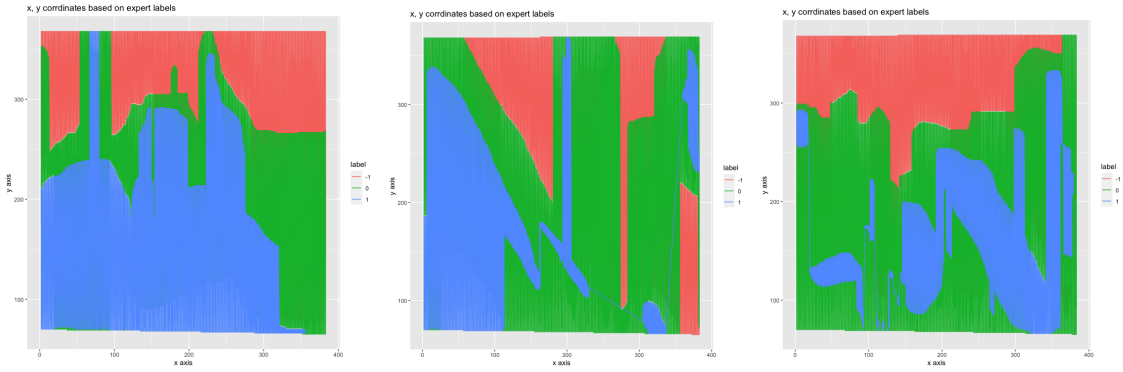


Image 1:

label	count	label_ratio
-1	42882	0.373
0	32962	0.286
1	39266	0.341

Image 2:

Label	count	label_ratio
-1	50446	0.438
0	44312	0.385
1	20471	0.178

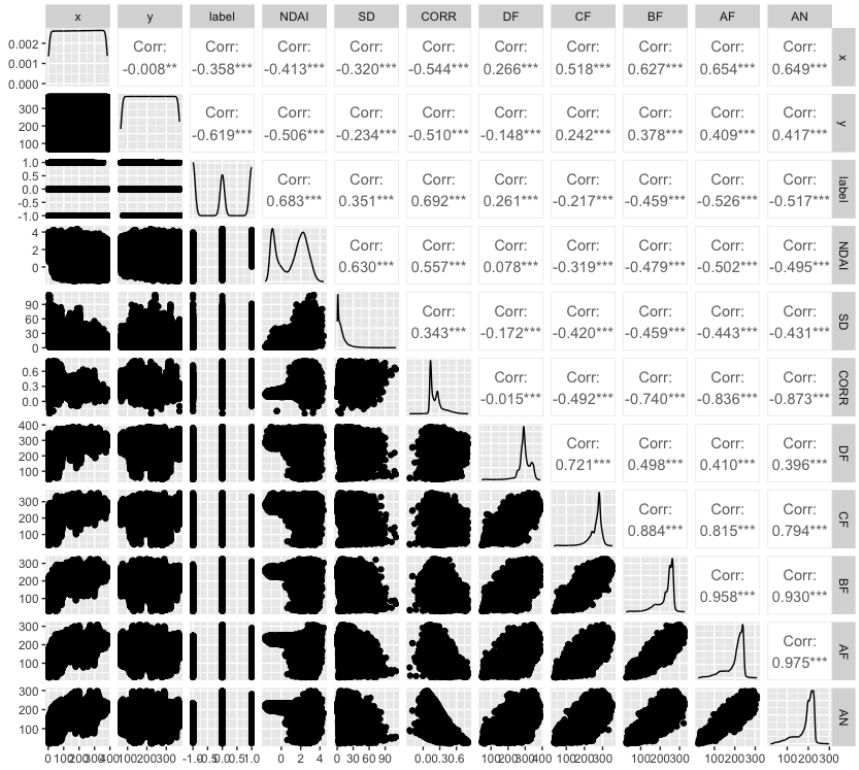
Image 3:

label	count	label_ratio
-1	33752	0.293
0	60221	0.523
1	21244	0.184

The label ratios differ by each image. The majority of labels are not cloud, and the minority is cloud. The big portion of data is unlabeled Based on the plots for the three images above, a clear boundary between cloud (label 1) and not cloud (label -1) is formed by unlabeled data. Given the words from the paper that experts leave uncertain data points with no label because of low confidence, we can assume that cloud and non-cloud data are separable and regions with the same label are mostly connected. Those unlabeled data may also have similar qualities which can not be differentiated from either cloud or non-cloud. Therefore, the x axis and y axis can not be used to separate the data, they may work with other features or project the data to other dimensions.

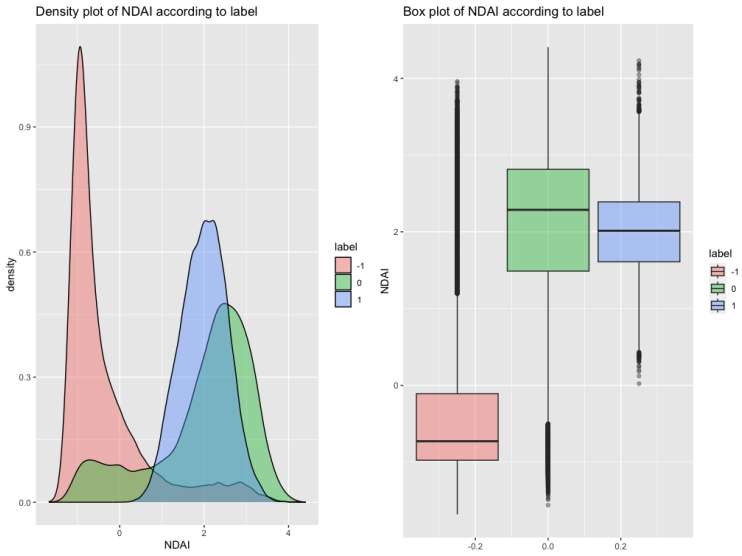
3. Quantitative EDA

Pairwise plot

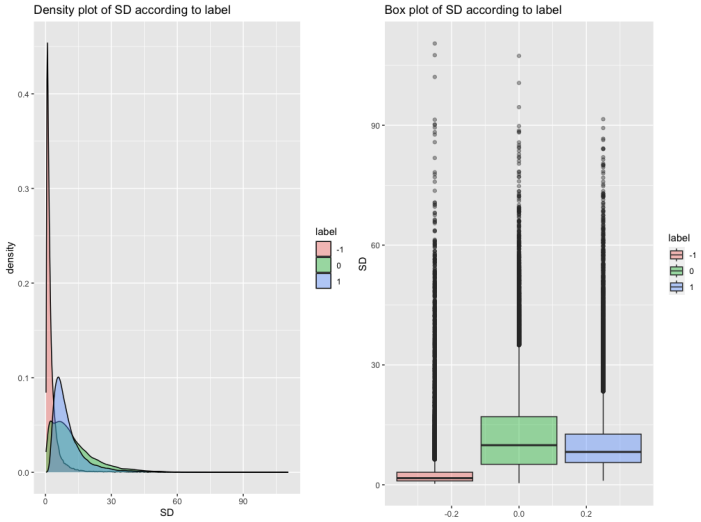


The density plots indicate four of the features (NDAI, SD, CORR and DF) are positively associated with the label. The correlation coefficients are ranked from high to low as CORR, NDAI, SD, DF, CF, BF, AN, AF. The ranking clearly shows the three engineered features have strong associations to the label, they are most likely to be important predictors

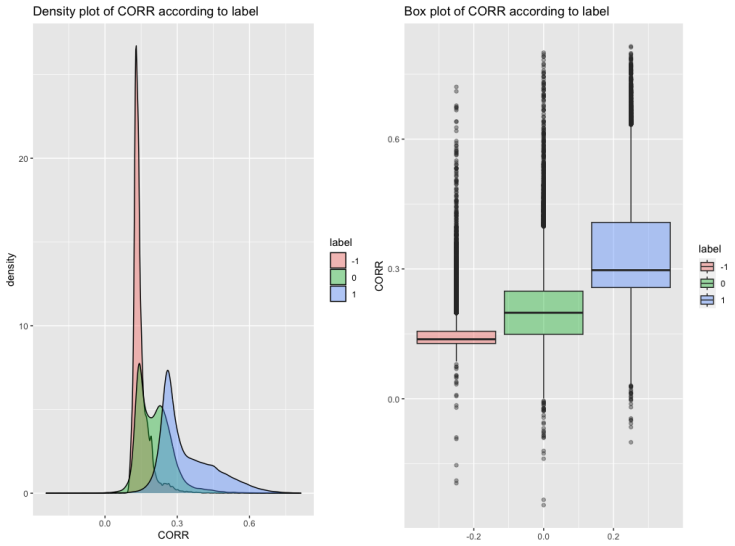
1. NDAI distribution



2. SD distribution



3. CORR distribution



Based on those plots, 1(cloud) and -1(not cloud) are separable with the above three features by density and boxplot. Cloud and unlabeled data are not separable by NDAI and SD. Cloud data intend to have much higher value than not cloud data, and the lower quartile of cloud is not adjacent to the upper quartile of not cloud. This would suggest the NADI, SD and CORR can be used to separate cloud data and not cloud data. Unlabeled data covers partial cloud data, which contradicts my assumption from the last question that unlabeled data could be the decision boundary to separate cloud data and not cloud data.

Preparation

A.Use two non trivial methods to split the data

The standard data preparation process to both splitting methods are:

1. Shuffle the whole dataset by resampling.
2. Remove unlabeled data since we are doing supervised learning with cloud and non-cloud only.
3. Split the data to training, validation and testing index as ratios of 0.8. 0.1, 0.1
4. Filter data by index derived from step 2 and apply standard normalization to feature columns because value range can affect the overall euclidean distance for any dissimilarity based algorithms like KNN and clustering algorithms.

Method 1 - Split the data according to label distribution

After removing unlabeled data, the label distribution of cloud and non-cloud is still imbalanced. Therefore, the first method is to split the data into training, validation and testing sets while

retaining label distribution. In order to retain the label ratio of 0.611 for non-cloud and 0.389 for cloud, the createPartition function was used to split the data with shuffle before every split.

Method 2 -Stratified sampling to reduce variance
Imbalanced label distribution would require a modified loss function for most machine learning algorithms and affect model accuracy. With stratified sampling, the label distribution has been resampled to 0.5 versus 0.5.

The difference between the two methods is the label distribution after resampling. Method 2 change the label ratio to half half instead of retaining the original 60% versus 40%

B. Accuracy of a trivial classifier

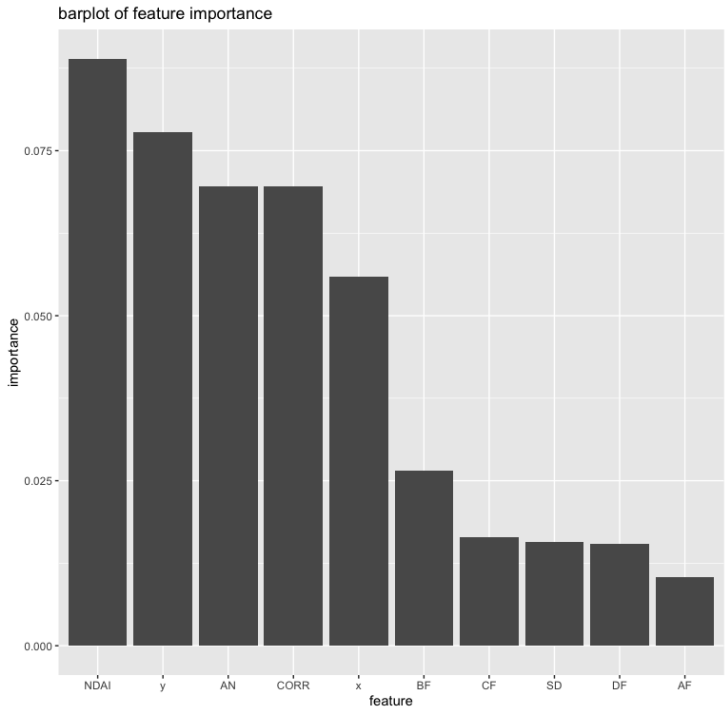
We use KNN as a baseline classifier here due to its simplicity. Before fitting in the KNN model, the data were rescaled by standard normalized function to ensure one vector won't dominate the Euclidean distance with neighbors. After trying a few different numbers of neighbors, the local optimal accuracy of KNN classifiers with k=5 is 0.95.

KNN is a similarity based algorithm, distance between vectors would affect the neighbors. The label of the current data point is decided by the k number of closest neighbors. To approach a high accuracy of a KNN classifier, model errors should be avoided by shuffling the dataset and rescale variables. With a high accuracy of 0.95, the KNN model seems to be a well-performing classifier; it can be used as a baseline classifier.

C. Suggest three of the best features

When the question comes to ranking features based on importance, random forest could be in great use for this problem because it reports decrease of node impurity when the variable has been removed. Features were ranked when splitting to nodes in multiple trees. Based on Gini importance, NDAI, y, AN are ranked the top three features.

Importance ranking barplot



Modeling

A.Cross-validation on 4 classification methods

Statistics of the 4 models

Logistic Regression

Using split according to labels:
The average accuracy over the 10 folds is 0.6109
The average loss over the 10 folds is 0.1645

Using split with stratified method:
The average accuracy over the 10 folds is 0.4962
The average loss over the 10 folds is 0.1002

Stratified splitting has a lower loss but also lower accuracy compared to labels. Overall, logistic regression does not have a satisfactory accuracy since it is below the baseline of KNN above.

LDA (Linear Discriminant Analysis)

Splitting according to labels:
The average accuracy over the 10 folds is 0.8986

The average loss over the 10 folds is 4.5876

Splitting with stratified method:
The average accuracy over the 10 folds is 0.9127
The average loss over the 10 folds is 1.3049

LDA has a significant improvement over logistic regression but it is still below the baseline KNN model. Stratified method performs better than based on labels with lower loss and high accuracy.

QDA (Quadratic Discriminant Analysis)

Splitting according to labels:
The average accuracy over the 10 folds is 0.9045
The average loss over the 10 folds is 11.4405

Splitting with stratified method:
The average accuracy over the 10 folds is 0.9052
The average loss over the 10 folds is 2.23375

For QDA, stratified splitting has a better performance than according to labels with lower loss and higher accuracy.

Naive Bayes

Splitting according to labels:
The average accuracy over the 10 folds is 0.8569
The average loss over the 10 folds is 53.2484

Splitting with stratified method:
The average accuracy over the 10 folds is 0.8560
The average loss over the 10 folds is 8.4542

For Naive Bayes, stratified splitting has a better performance than according to labels with much lower loss and higher accuracy.

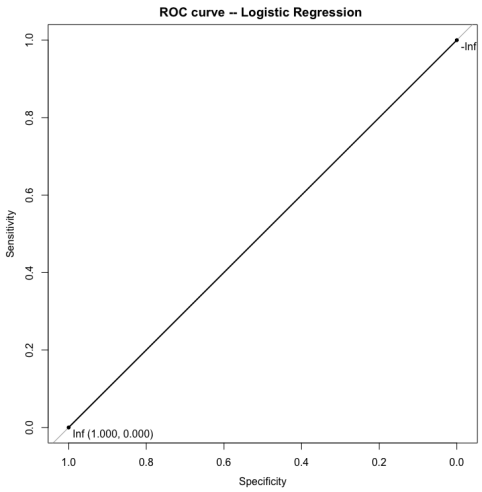
The data was selected by both splitting methods respectively, and the models were fitted using both data separately. The same preprocessing steps were applied to both data to control. Stratified splitting demonstrates better performance.

The linear logistic regression model has the lowest accuracy. There could be some possible reasons like label ratio and cutoff point to assign labels(0.5 here) or the data is not linearly separable as suggested from the first section. Overall, three of the four classification methods

have an accuracy around 90%, except logistic regression. Stratified splitting method produces lower loss than the other method.

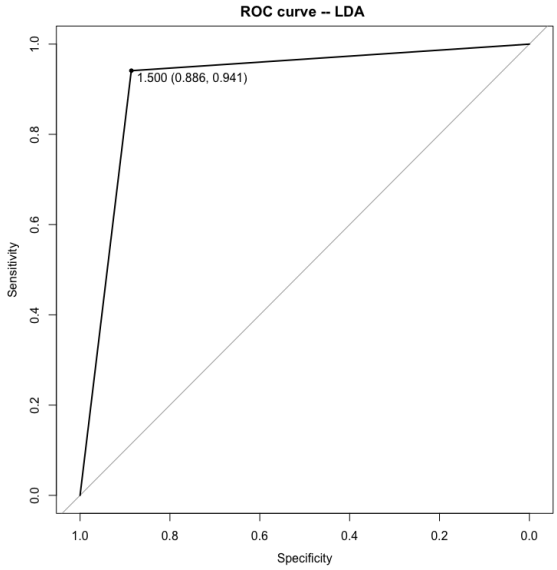
B.Use ROC curves to compare the different methods

Logistic Regression



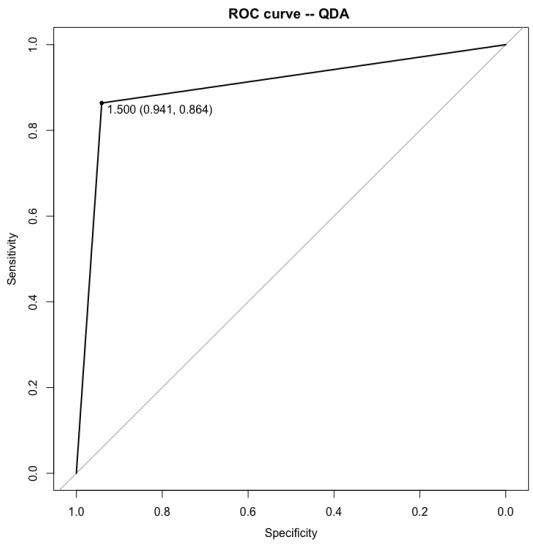
Area under the curve: 0.5

LDA



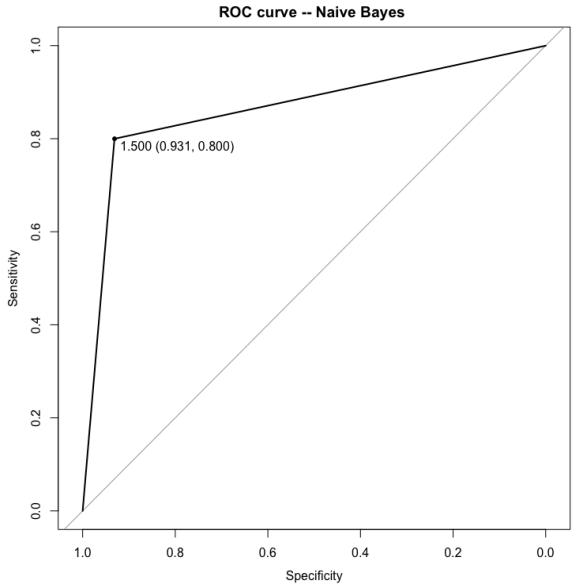
Area under the curve: 0.9134

QDA



Area under the curve: 0.9024

Naive Bayes



Area under the curve: 0.8654

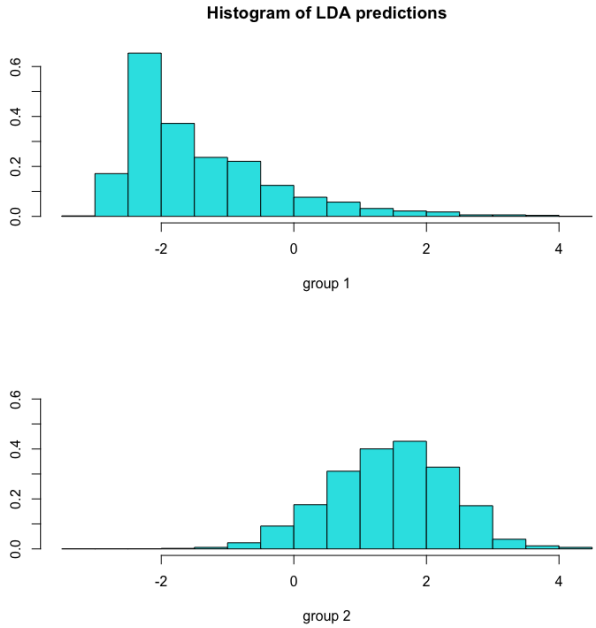
The cutoff value was chosen to maximize (sensitivity + specificity - 1) as a cut-off. Therefore, they were chosen to minimize the left corner of the ROC plot.

4.Diagnostics

A. Analysis on the best model

As the best model from the last section, LDA will be evaluated here.

Stacked histogram on testing dataset



Based on the histogram, cloud and non-cloud do not have much overlap, as a good separability on the model.

Confusion Matrix and Statistics

	Reference	
Prediction	1	2
1	900	116
2	58	926

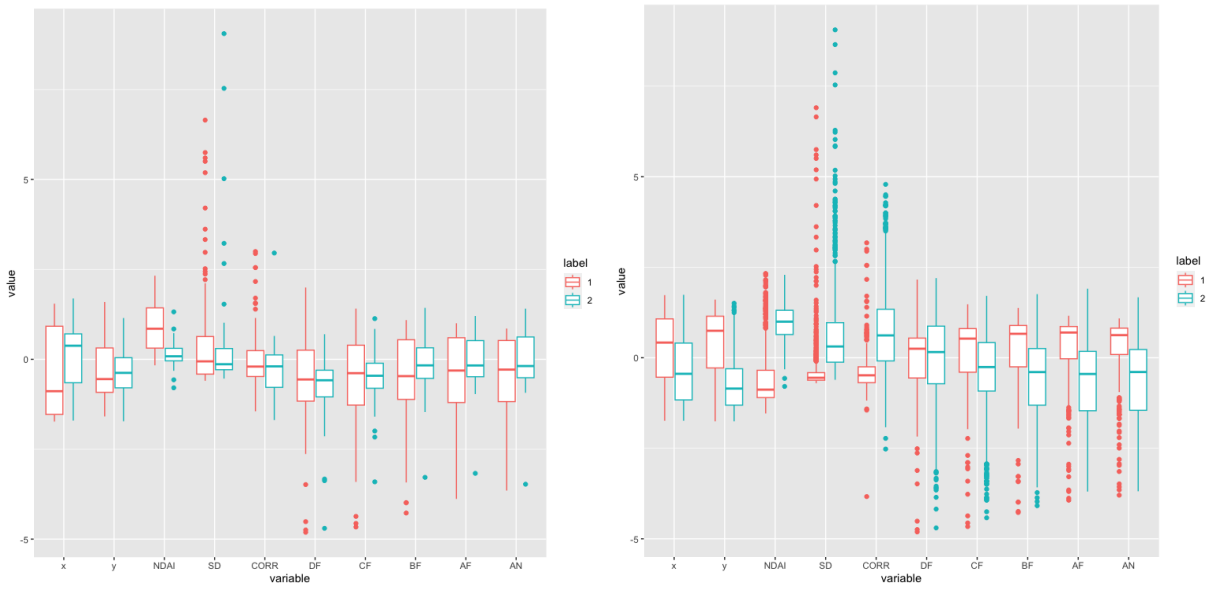
Sensitivity : 0.9395 Which means the model has a high true positive rate
Specificity : 0.8887 Which means the model has a high true negative rate

Unlike some models that have a high true position rate but a lower true negative rate, this model is good at predicting both positive labels and negative labels. In general problems, we intend to predict both labels with high accuracy. Some models can be biased to majority labels as a high TPR(true positive rate) and high FPR (false positive rate), which would lead to the fact the

model has a lower accuracy on minority labels. This LDA model has both high sensitivity and specificity as the ROC curve suggested from the last question.

B. Misclassification analysis

Boxplot of misclassification data(left) and Successfully classified data(right)



Indicated by the boxplot, misclassified data intend to have smaller values on cloud data(1) and small bottom tail of non-cloud data(2).

Since logistic regression doesn't perform well, the labels may not be linearly separable. Besides, Neural networks may also work on this problem since it adds complexity. The models would generally do well on future data without expert labels because the accuracy on testing sets which the model had never seen before was above 90%.

The result changes with different methods to split the data as shown in the modeling section. Stratified splitting resample the data to balance label distribution to 50% cloud versus 50% non-cloud, which improve performance on classification models.

In conclusion, performance of classifiers could vary to different dataset and different problems. For example, logistic regression works better on data which is linearly separable and SVM(support vector machine) is on the opposite way with kernel trick to try different dimensions. Data preparation is a crucial process to remove noises like outliers and feature selection.

References

[1] Shi, Tao & Yu, B. & Clothiaux, Eugene & Braverman, Amy. (2008). Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies. Journal of the American Statistical Association. 103. 584-593. 10.1198/016214507000001283.