Stat 154 Project2: Cloud Data

Andy Liu 26295516

Jade Wang 26349361

GitHub Link https://github.com/liushaoyusz/stat154_project2

1. Data Collection and Exploration (30 pts)

(a) Summary of the paper

The purpose of this study is 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. The data used in this study were collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and Baffin Bay. By excluding inefficient data units, the data investigated included 57 data units with 7,114,248 1.1-km resolution pixels with 36 radiation measurements for each pixel.

The authors perform two algorithms in this study, both of which are based on three physically useful features for identifying surface pixels, and thus obtain cloudy pixels by exclusion, with each pixel treated as independent of the others. In general, there are three steps: (1) construct three features based on EDA and domain knowledge; (2) build an ELCM cloud detection algorithm by setting thresholds (fixed or data-adaptive) on each feature, and apply ELCM to each data unit to produce the first cloud detection product; (3) predict probability of cloudiness, the second cloud detection product, for the partly cloudy data units by training Fisher's QDA on the labels produced by the ELCM algorithm.

In conclusion, by comparing the results against the expert labels, the authors find that the ELCM algorithm is more accurate and provides better spatial coverage than the existing MISR operational algorithms for cloud detection in the Arctic. By efficiently combining classification and clustering frameworks, ELCM is suitable for real-time, operational MISR data processing. Significantly, the study has two effects which go beyond of technical development and implementation of statistical methods. First, three features found in the study can identify clear (cloud-free) regions with a classifier no more sophisticated than QDA but better performance. And now statisticians are directly involved in the nuts and bolts of data processing. Furthermore, the study also demonstrates the power of statistical thinking and the ability of statistics to contribute solutions to modern scientific problems.

(b) Summarize the data

Image1: there are 43.78% pixels are in "1" (cloud) class; 38.46% pixels are in "0" (unlabeled) class; 17.77% are in "-1" (not cloud) class.

Image2: there are 37.25% pixels are in "1" (cloud) class; 28.64% pixels are in "0" (unlabeled) class; 34.11% are in "-1" (not cloud) class.

Image3: there are 41.09% pixels are in "1" (cloud) class; 73.31% pixels are in "0" (unlabeled) class; 25.86% are in "-1" (not cloud) class.

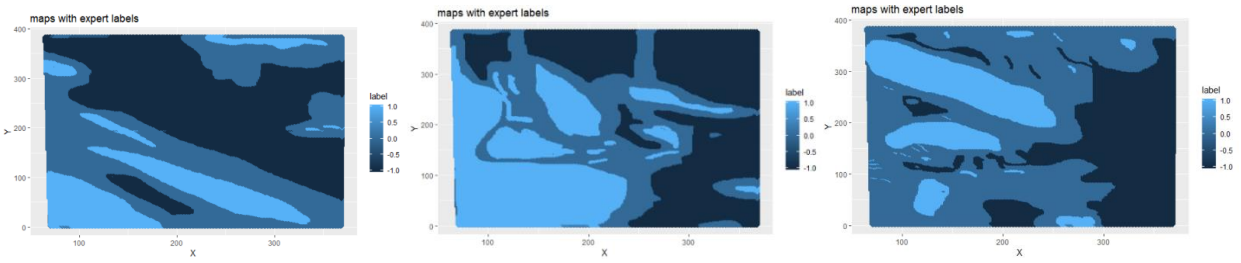
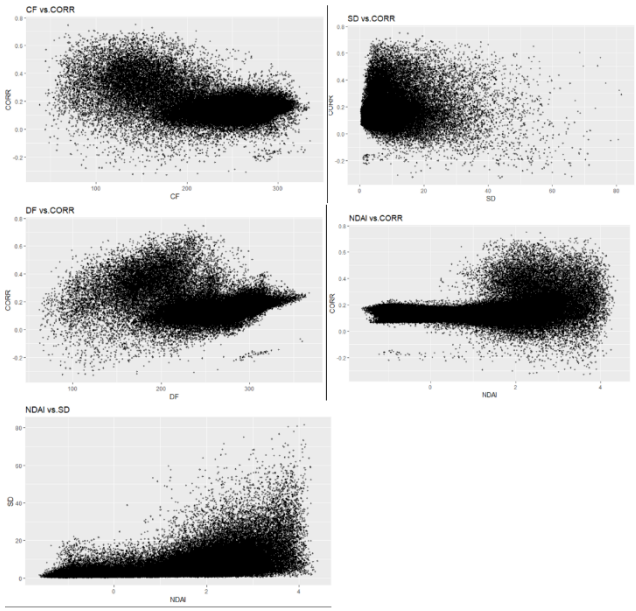


Image 1 *Image 2* *Image 3*

By plotting the maps for three datasets (shown above), we find that, in all three maps, expert label "1" (Cloud) usually do not border "-1" (not cloud). In other words, pixels with "-1" (not cloud) and with "1" form their own groups. Hence, an i.i.d assumption of the samples is not ideal, since the label of a particular pixel is correlated to its neighboring pixels.

(c) Perform a visual and quantitative EDA

For image 1 data:



We draw scatter plots for comparisons between features.

CORR vs. SD: a negative relationship; the variation in CORR increases as SD increases in observations.

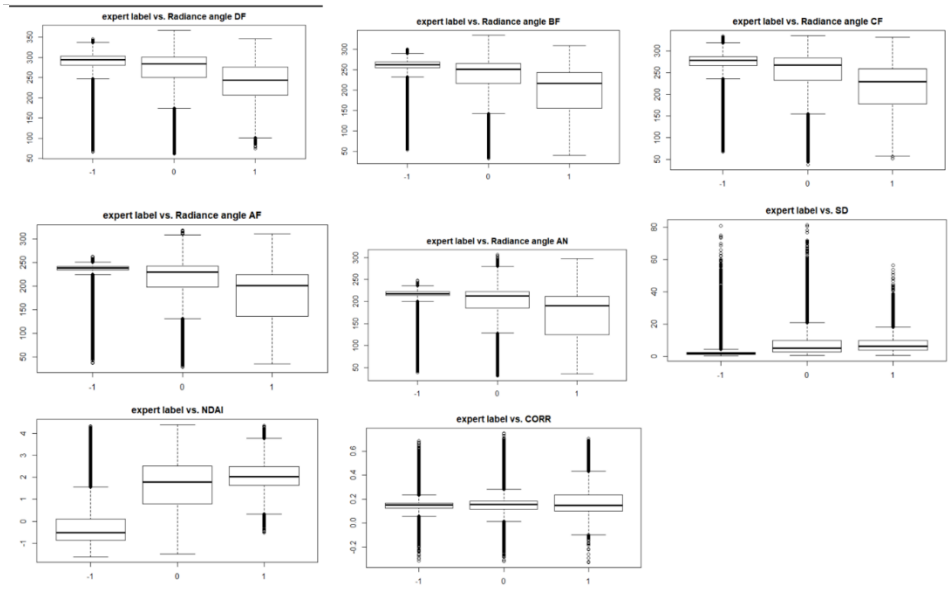
SD vs. NDAI: a positive correlation; the points become more dispersed as SD increases in observations.

NDAI vs. CORR: horizontal when NADI is smaller than roughly 1.5; no significant correlation when NADI is greater than 1.

DF vs. CORR: no significant correlation when DF is smaller than 200; a general positive trend when DF is greater than 200.

CF vs. CORR: no significant correlation, but the points are more concentrated when CF is greater than 150.

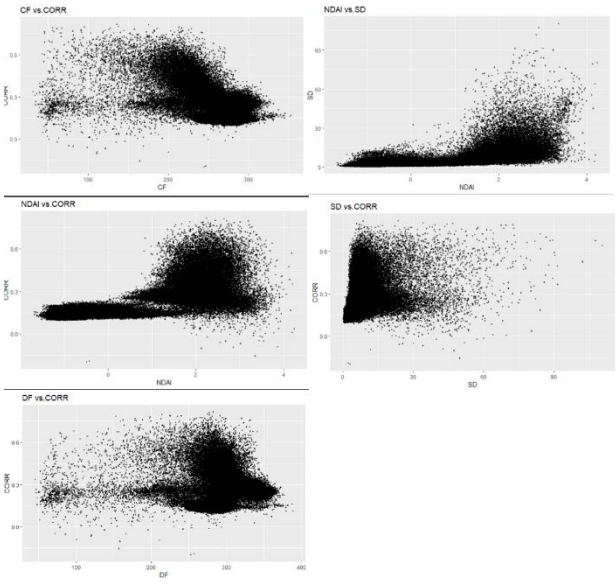
We draw the boxplots for comparisons between expert labels and individual feature.



The two classes, no cloud and cloudy, have differences in radiance angle DF, BF, CF, AF and AN. From the boxplot, no cloud has significantly higher mean and less variance than places with cloud.

Also, the differences between two classes are based on NDAI and CORR. In terms of NDAI, observations with no cloud has significantly lower mean and similar variances, compared to the cloudy class. In terms of CORR, no cloud class has similar mean but smaller variance than cloudy class. In terms of SD, two classes have very close mean, except that cloudy class has a slightly higher variance.

<u>For image2 data:</u>



We draw scatter plots for comparisons between features.

<u>CORR vs. SD</u>: the data points are condensed as a triangle shape on the left; the variation in CORR increases as SD increases in observations.

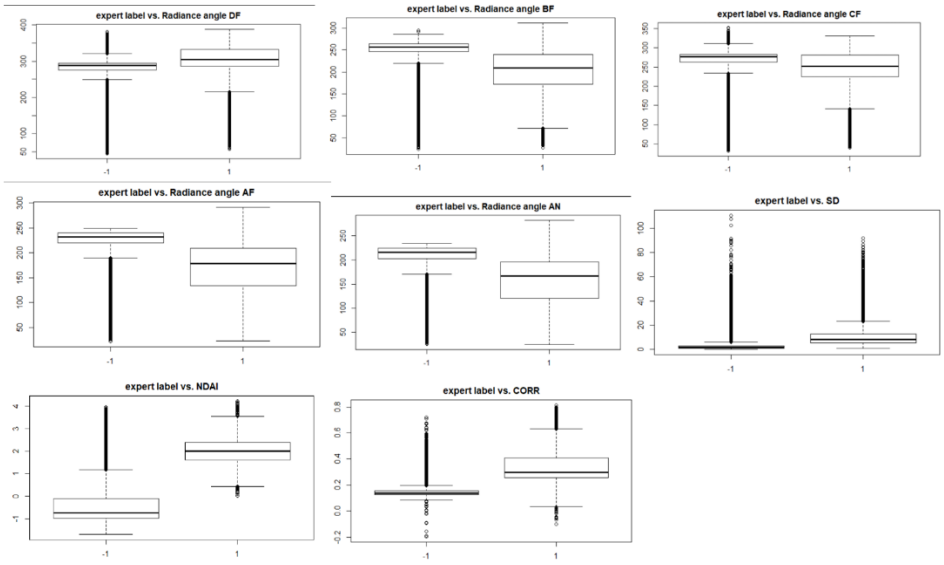
<u>SD vs. NDAI</u>: a positive correlation when NADI is greater than 1; the points become more dispersed as SD increases.

<u>NDAI vs. CORR</u>: horizontal when NADI is smaller than 1; no significant correlation when NADI is greater than 1. We notice a split of two groups.

<u>DF vs. CORR</u>: no significant correlation; data points become more condensed when DF is greater than 250.

<u>CF vs. CORR</u>: no significant correlation, but the points are more concentrated when CF is greater than 200.

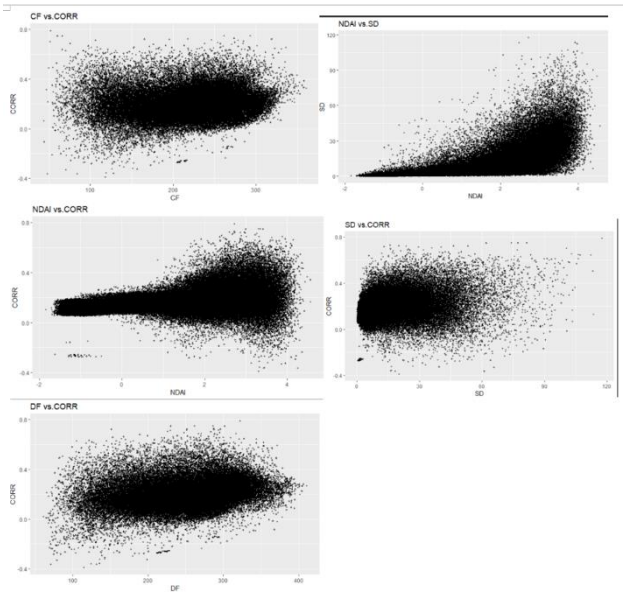
We draw the boxplots for comparisons between expert labels and individual feature.



The two classes (cloud vs. no cloud) have differences in radiance angle DF, BF, CF, AF and AN. From the boxplot, no cloud has significantly higher radiance, less variance in radiance on average than places with cloud.

Also, the difference between two classes are based on NDAI and CORR. In terms of NDAI, observations with no cloud has significantly lower mean and similar variances, compared to the cloudy class. In terms of CORR, no cloud class has similar mean but smaller variance than cloudy class. In terms of SD, two classes have very close mean, except that cloudy class has a slightly higher variance.

<u>For image3 data:</u>



We draw scatter plots for comparisons between features.

<u>CORR vs. SD</u>: no significant correlation; data condensed at left; the variation in CORR increases as SD increases in observations.

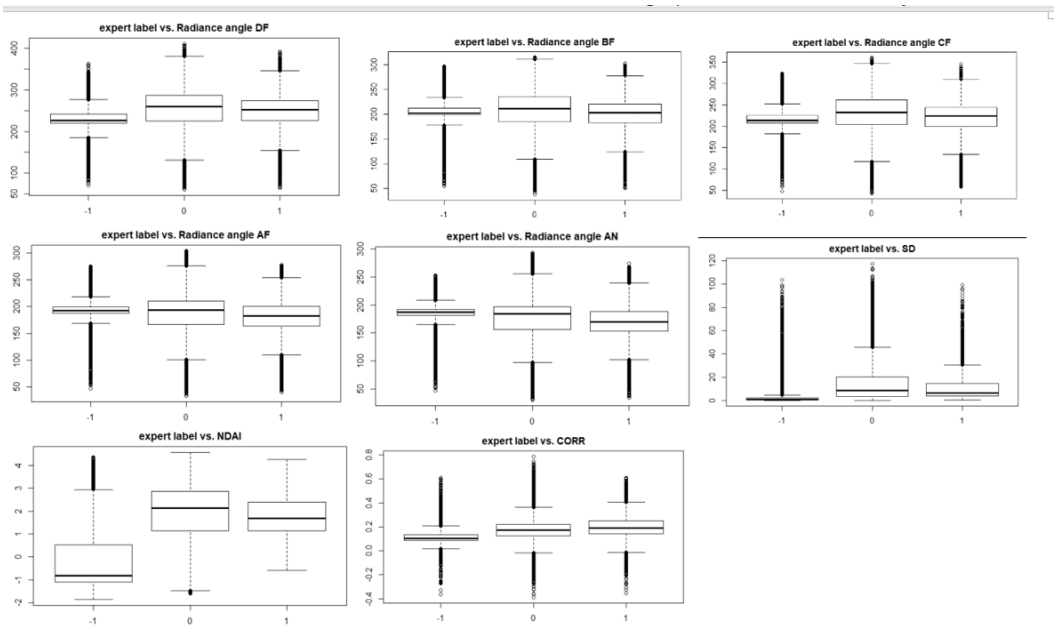
<u>SD vs. NDAI</u>: a positive correlation; the points become more diverse as SD increases in observations.

<u>NDAI vs. CORR</u>: horizontal in general; data points become more diverse as NADI increases.

<u>DF vs. CORR</u>: no significant correlation; points are more concentrated at right.

<u>CF vs. CORR</u>: no significant correlation; points are more diverse at left.

We draw the boxplots for comparisons between expert labels and individual feature.



For all features except NDAI, no cloud class has a similar mean but a smaller variance than cloudy class. In terms of NADI, we notice that no cloud class has a much lower mean than cloudy class, but their variances are pretty similar.

Q2 Preparation

(a) Split the entire data in two non-trivial ways

First, we split data in each image to 9 blocks based on x and y values. Then each image gives us 9 blocks. In total, we have 27 blocks from all three images. We randomly select 9 blocks for training, 9 for validation and 9 for test. Because the data has spatial correlation, it is potentially problematic to split the data in fully randomized way. The way that we do can expectedly mitigate this problem, because observations with similar values of x and y are spatially correlated, and they also tend to be grouped together in blocks.

Second, we use random split method. We combine points in three images all together, and randomly select training, validation and test sets from the entire dataset.

(b) Report the accuracy of a trivial classifier.

<u>For the first split method (the blocking):</u>

We find the trivial classifier has 0.5689587 accuracy on the validation set, and 0.5837874 on the test set.

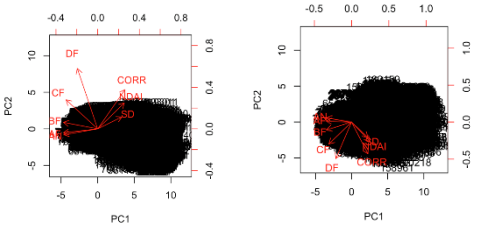
<u>For the second split method (the random split):</u>

We find the trivial classifier has 0.6122158 accuracy on the validation set, and 0.6102836 on the test set.

When on average the pixels have high prevalence of no cloud (encoded as -1), then the trivial classifier will have high average accuracy because it will always assign pixels to no cloud (encoded as -1).

(c) First order importance

We look at the dataset except the test set.



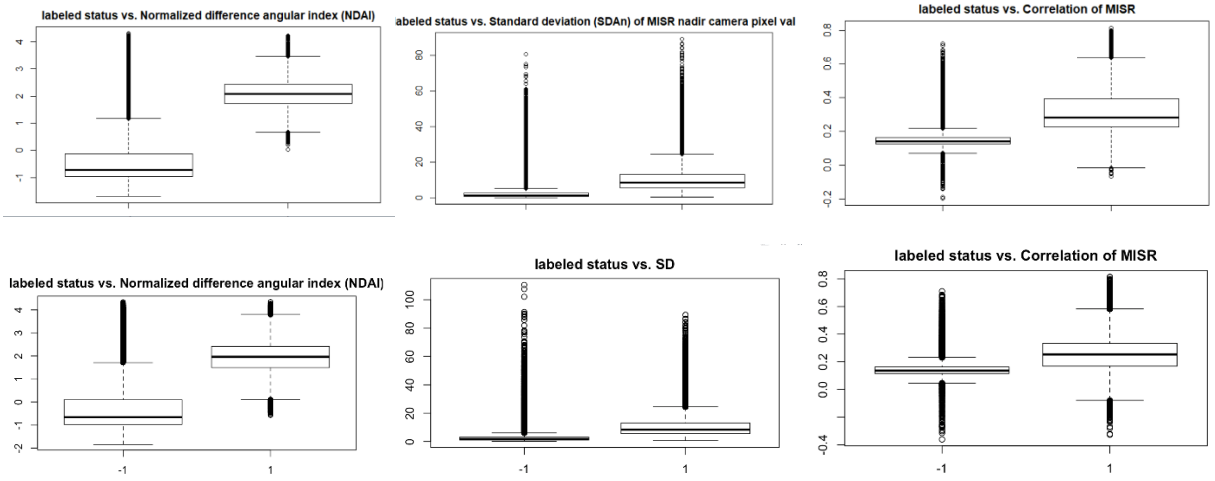
(1) First, we perform a principal component analysis (PCA) for both split methods, and we find that features are grouped into two sets. Here are the biplots for both methods (Left: 1st method; Right: 2nd method):

(2) Furthermore, we calculate the correlation coefficients of each feature with the label.

For both split methods, we find that the coefficients of NDAI, SD, and CORR are among the top. Specifically, NDAI: 0.81 (1st method), 0.76 (2nd method); SD: 0.50 (1st method), 0.43(2nd method); CORR: 0.67(1st method), 0.55(2nd method).

Based on PCA and correlation analysis, we choose NDAI, SD, and CORR as the best three features.

Then, we do boxplots of these features (NDAI, SD, and CORR) with labeled status to see if there is significant difference of these features between labeled classes (-1 vs. 1). The boxplots reported confirm that there are significant differences between two classes in terms of NDAI, SD, and CORR. Here are the boxplots for both split methods (top: 1st method; bottom: 2nd method):



(d) Write a generic CV function

(See Github)

Q3 Modeling

(a) Try and access the fit of several classification methods

First of all, we merged the training and validation set to fit the CV model. We use 5-fold Cross validation and tried QDA, LDA, Logistic and KNN (with k = 5) methods, in which we find that KNN provides the best accuracy both on the training set and on the test set.

Here are the assumptions for the four classification methods we used:

KNN assumptions: it does not make any assumptions of the underlying data distribution. It is a non-parametric "lazy" classification method.

LDA assumptions: The LDA classifier assumes that predictors come from normal distribution with class-specific mean vector and a common variance. From the output using "scaling" for the LDA class, we find it is not the case. Different predictors have different covariance.

QDA assumptions: it holds the same assumptions as LDA except that the covariance matrix that is not common to all K classes.

Logistic regression assumptions: little or no multicollinearity among independent variables; observations are independent. Perfect multicollinearity makes estimation impossible, while strong multicollinearity makes estimates imprecise. This assumption probably is satisfied because CORR, SD and NDAI do not have strong multicollinearity and they measure different aspects. Another assumption of independent observations is likely to be violated, because spatial correlation makes nearby observations to be strongly correlated with each other.

For four classification methods, we summarize the accuracies for each fold, the average accuracy across folds, and test accuracy as below:

	KNN	QDA	LDA	Logit
fold 1 error rate	0.94766413	0.94505495	0.93344715	0.93211237
fold 2 error rate	0.94797293	0.94055564	0.937046	0.9345336
fold 3 error rate	0.94881266	0.94459103	0.93742046	0.92944062
fold 4 error rate	0.9479435	0.94403228	0.93493512	0.93313674
fold 5 error rate	0.94887316	0.94368908	0.9374515	0.93400385
average accuracy rate across folds	0.9482533	0.9435846	0.93606	0.9326454
test accuracy	0.9396008	0.9369976	0.9271244	0.9115053

(1st method: blocking split)

fold 1 error rate	0.9120828	0.8962186	0.8982373	0.8955697
fold 2 error rate	0.9088713	0.8973721	0.8996467	0.8933708
fold 3 error rate	0.907826	0.8968314	0.8958942	0.8924375
fold 4 error rate	0.9107098	0.8967954	0.8992106	0.8911359
fold 5 error rate	0.9084388	0.8978084	0.8963267	0.8924696
average accuracy rate across folds	0.9095857	0.8970052	0.8978631	0.8929967
test accuracy	0.9083818	0.8960968	0.8978559	0.8912953

(2nd method: random split)

Based on the CV results above, we find all four classification methods tend to have higher accuracies (of both train and test test) by the 1st split method than by the 2nd method. Also, we notice that the error rate in each fold for all these classification methods are rather stable.

Among the four classification methods we used (QDA, LDA, logistic and KNN), we find that KNN method provides the best accuracies and has the highest average accuracy across folds, as well as the highest for the test accuracy

(b) Use ROC curve to compare different methods

receiver operating characteristic curve (ROC) is a commonly used summary for assessing the tradeoff between sensitivity and specificity.

The horizontal distance from the curve to the right measures specificity (1− false positive rate); and the vertical distance from the curve to the x axis measures sensitivity (1-false negative rate).

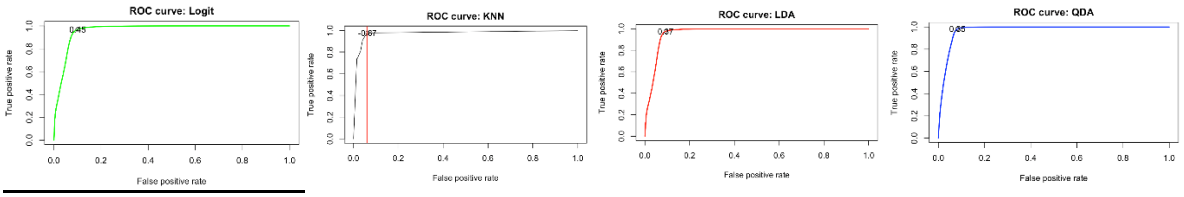
To deal with the trade-off between these two, we apply a function in the code to compute the cutoff point that maximizes the sum of sensitivity and specificity.

This could be good in this case because we take the importance of both traits to account. The plots show the optimal cutoff points in each graph for KNN, QDA, LDA, and Logit.

For the 1st split method:

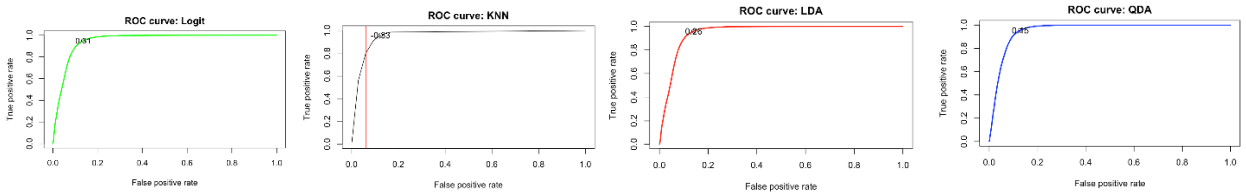
The cutoff values for 4 classification methods:

logit=0.45; KNN= -0.67; LDA=0.37; QDA=0.35.



For the 2nd split method:

logit=0.308; KNN= -0.333; LDA=0.256; QDA=0.152



(c) [Bonus] Assess the fit using other relevant matrices

We use three alternative methods 1) Confusion matrix 2) Cohen's Kappa 3) The area below ROC curve.

First, the confusion matrix method can effectively show true positives, true negatives, false negatives, and false positives. We include the confusion matrix for logit, KNN, LDA, and QDA (in order) as below:

Reference		
Prediction	0	1
0	40718	2350
1	2564	28508

logit

Reference		
Prediction	-1	1
-1	41523	2723
1	1759	28135

KNN

Reference		
Prediction	-1	1
-1	40823	2944
1	2459	27914

LDA

Reference		
Prediction	-1	1
-1	40902	2291
1	2380	28567

QDA

Furthermore, the confusion Matrix in caret package also outputs Cohen's Kappa:

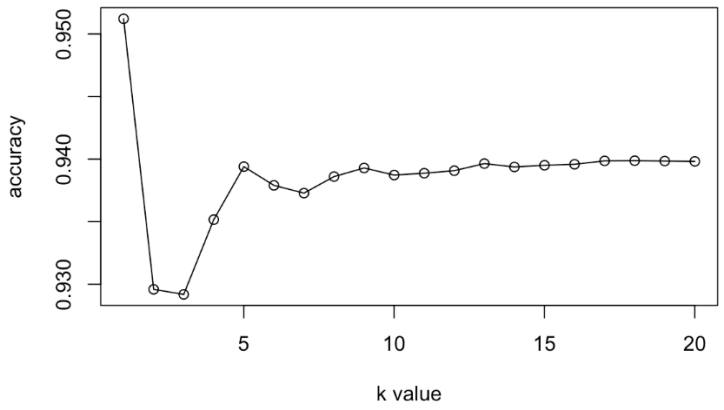
$$Kappa = \frac{Observed\ Accuracy - Expected\ Accuracy}{1 - Expected\ Accuracy}$$

We calculate Kappa for KNN: 0.875, QDA model: 0.8855; LDA: 0.8703 and logit: 0.8583. We find that QDA has a slightly better Kappa than KNN, but KNN is quite good compared to LDA and Logit.

Last but not least, the area under the ROC curve can be used as an evaluation metric to compare the efficacy of the models, and hence we calculate the area under the ROC curve for each model. The area under KNN model is 0.9742579, logit model is 0.9590257, LDA model is 0.9547143, and QDA is 0.9625022. Therefore, we conclude that KNN model has the highest value in this metric. [Because of space limitation, we only report the analysis using the 1st splitting method for this question.]

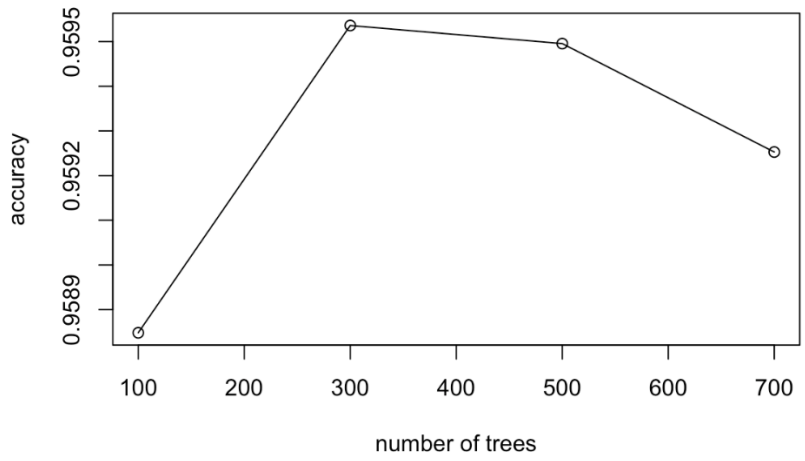
Q4 Diagnostics

(a) An in-depth analysis of a good classification method with plots



The plot above is the test accuracy rate when we use the KNN method. We pick KNN here, because it provides the best outcomes in the previous tests. We find that the accuracy rates converge to around 0.940 when k > 5, while small number of nearby pixels (small k) would give us unstable results of accuracy**.**

In the plot below, we try to find the relationship between accuracy and the number of trees when using the classification method of random tree. We find that random tree method actually gives us a higher accuracy than KNN method does, and therefore we think random tree can be a better potential alternative of KNN, which we would discuss further in 4(c).



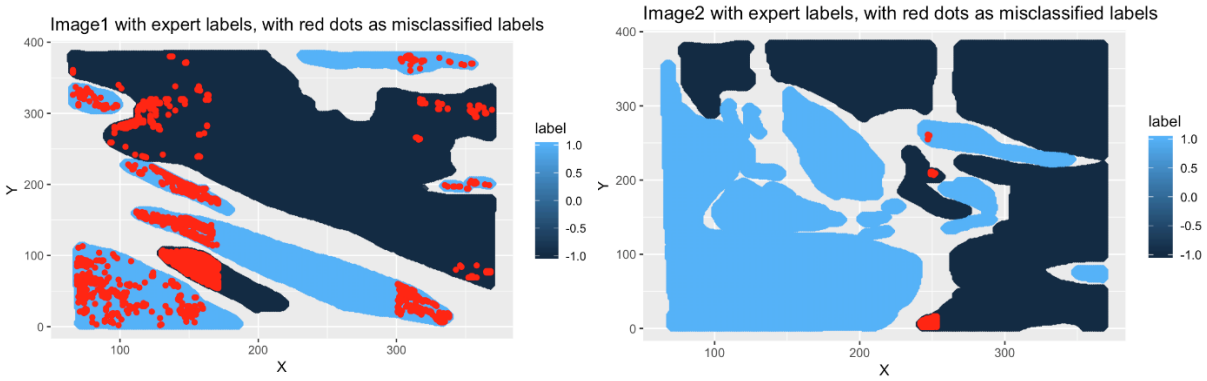
(b) Any errors/problems? Use quantitative and visual methods of analysis. Do you notice problems in particular regions, or in specific ranges of feature values?

We focus on the misclassified labels to see if they have some systematic patterns.

The previous best method we find is KNN, but we find that Random forest gives an even better result, which would be further discussed in 4(c). Also, from the confusion matrix on the training set, we find it has higher rate to classify actual -1 (no could) as 1 (cloud) than classifying 1(cloud) to -1 (no cloud). Then, we try to figure out where these errored labels concentrate spatially. We map the misclassified points to the

image maps we created in question 1 below. From the plots we find misclassified points usually locate on the edge of cloud or no cloud regions. The classification errors especially locate in regions where cloud and non-cloud regions locate very close to each other. Besides discussing the spatial patterns of misclassification, we also report the particular ranges of feature values that misclassification occurs here:

Training set has NDAI in the range: -1.6811455,4.3460255; misclassified labels have NDAI in this specific range: -0.52865756, 4.1982489. Misclassified labels have SD in this specific range: 0.7903856, 88.28141; Training set has SD in the range:0.2506054, 110.46764. Misclassified labels have CORR in this specific range: -0.32712963, 0.7207008; Training set has CORR in the range: -0.32712963, 0.8143999.



(c) Any better classification? Can work how well without expert labels?

There are at least two ways to make it better:

1) Use another (better) classification method. We try using random forest method, and such method gives us higher accuracies on training set and test set. For the 1st split method (blocking), the training accuracy is about 0.976, and test accuracy is 0.959, the highest among all methods that we tried. For future data without expert labels, we could use current images as the training set to train our model. If the future unlabeled data are similar to the images we have, then we could get fairly good predictions. We report the Kappa value as 0.916, detection rate as 0.5695, and balanced accuracy as 0.9563. We include a Confusion matrix (for test set) for random forest as below:

	Reference	
Prediction	-1	1
-1	42222	1943
1	1060	28915

2) Use better features. These measurements of features that the authors used in the paper may not be the best in all cases. For example, CORR is average linear correlation of radiation measurements at different view angles, defined at a 2.2-km spatial resolution, but with a spacing of 1.1 km, using the 275-m data. The feature NDAI also uses the average radiation measurements, which are over a 4 × 4 group of 275-m resolution red radiation measurements. However, the features can be more effective if we change the spatial resolution and spacing. For example, we may change the spacing 1.1km of CORR to 0.8km, which could provide a more detailed correlation measurement, and thus the it is more effective especially on the boundary points.

(d) Do (a) (b) results change?

We apply the random forest method using the 2nd split method (random split). We find the accuracy of training set is 0.919 and for test set is 0.9167. The results show that random forest is still the best among all classification methods that we tried.

(e) Conclusion

Based on previous results, we find that random forest provides us the best model to explore and classify cloud detection, using either random split or blocking split methods. Among the rest of classification methods that we tried in this project, KNN is slightly better than the other three ones (logit, LDA & QDA). We guess the reason behind is that KNN is a non-parametric method which does not make assumptions about data distribution, and thus KNN is more suitable for cases with spatial correlation. However, all these classification methods have limitations and problems, especially when we approach to the boundaries of the dataset, where no cloud (label=0) and cloud (label=1) locate spatially close together. Still, we can always increase the accuracies by using a more suitable and effective classification method, like random forest. Besides that, another potential improvement of our prediction is to use better features, but it may require much more domain knowledge and more detailed and precise measurements.