Predicting the White: Cloud or Ice-covered Land?

Chenxing Wu (3034269515)
Yanting Pan (3034299376)

April 26 2019

1 Data Collection and Exploration

1.1 Summary on Paper

With the issue of climate change, change in distribution of clouds above the Arctic can lead to sensitivity towards increasing atmospheric carbon dioxide levels, which is one of the culprits for global warming. As a result, close monitoring of the change in cloud levels in the Arctic is crucial for assessing the intensity of climate change. However, identifying white clouds from the icy, showy, white surface of the Arctic with satellite images has always been a challenge.

The data was collected from the Multiangle Imaging SpectroRadiometer (MISR) on NASA's Terra satellite that launched in 1999. MISR has 9 cameras with different zenith angles, and the angles with angular radiance data collected in this project are: 70.5°(Df), 60.0°(Cf), 45.6°(Bf), and 26.1°(Af) in the forward direction as well as 0.0°(An) in the nadir direction.

The data set for this project consists of three satellite images stored in space-separated txt files. Each row of the image file represents a pixel of the satellite image, with x and y coordinates indicating the position of this pixel. The other columns of the data set includes an expert label (if this pixel belongs to a cloud or not, or undetermined), CORR (the correlation of the images from the same scene with different viewing directions), SD (the standard deviation of a pixel value from the same scene with different viewing directions), NDAI (Normalized Difference Angular Index, measures changes in a scene with changes in the MISR view direction), and the angular radiance values from the 5 camera angles mentioned above.

The paper presented the Enhanced Linear Correlation Matching algorithm (ELCM) that labels the pixel as clear (-1) when $SD_{An} < threshold_{SD}$ or $CORR > threshold_{CORR}$ and $NDAI < threshold_{NDAI}$. The model provided 91.8% accuracy when compared to expert label and can be used for QDA modeling training to predict probability of partially cloudy areas.

Aside from providing NASA with a reliable method that fills the blank of classifying clouds over the Arctic region, this study also shows how statisticians can join in the active data-collecting and problem-solving process and to others who are not familiar with statistical disciplines, the power of statistical thinking and methods in terms of solving scientific problems.

1.2 Summary on Data

Table 1 shows that the proportions of not cloud, cloud and unlabeled pixels are 0.37, 0.23 and 0.40 respectively across the whole three images.

Separately, non-cloud pixels have the highest proportion in image 1 and image 2. The proportion of non-cloud is always higher than that of cloud. In image 1, the difference between the proportions of non-cloud and cloud pixels is largest among three images. High CORR values to some extent

	proportion
not cloud	0.367755154012664
cloud	0.234349859357094
unlabeled	0.397894986630242

	image1	image2	image3
not cloud	0.43778909823048	0.372145900771507	0.292912374489061
cloud	0.177838589175571	0.34111719225089	0.184553904960473
unlabeled	0.384596023156305	0.286086254632563	0.52267460530998

Figure 1: Proportion of Pixels for Different Classes

implies clear weather. However, the box plot 2 shows image 1 has low CORR values. The reason is that image 1 has smaller SD values, which means smooth surface, and smooth cloud-free terrain surface could be classified as cloudy when computing CORR values. Moreover, in image 2, the proportion of could pixels is highest compared to other images. From the box plot, we can see that image 2 has the larger NDAI values, which coincides with the evidence of presence of clouds.

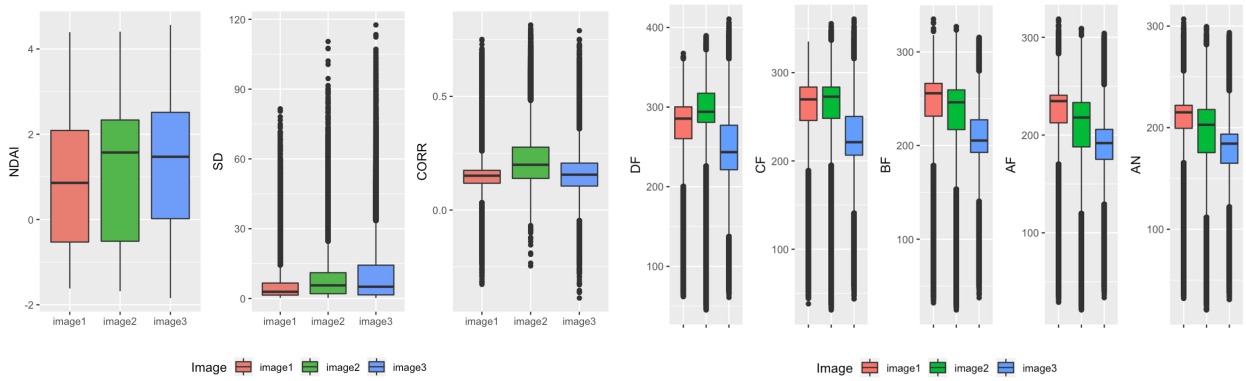

Figure 2: Box Plot

From figure 3, we can see that same label appears in proximity with each other, which indicates the samples are not identical and independent. Therefore, we cannot randomly split data into training, validation and test set. The details about data splitting is discussed in section 2.1.

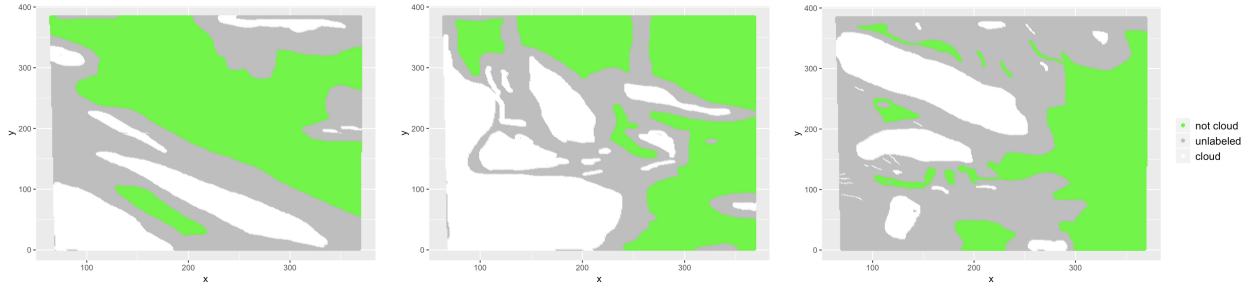

Figure 3: Image of Expert Labels

1.3 EDA

We use Pearson correlation coefficient to measure the linear correlation. For absolute values of correlation coefficients, 0-0.19 is regarded as very weak, 0.2-0.39 as weak, 0.40-0.59 as moderate,

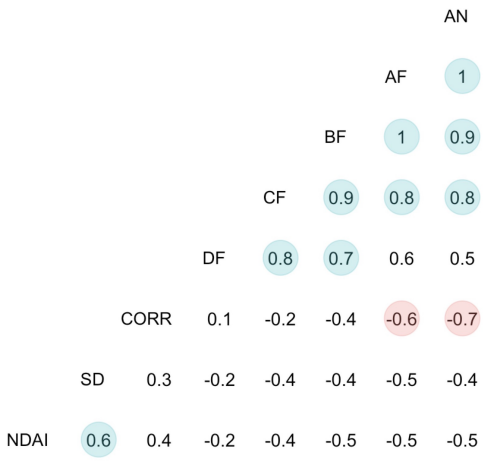

Figure 4: Pearson Correlation Coefficient between Features

NDAI	CORR	SD	DF	CF	BF	AN	AF
0.616934624	0.444059231	0.295447745	0.006550085	-0.208279170	-0.337948500	-0.389358825	-0.389741017

Figure 5: Pearson Correlation Coefficient between Response and Features

0.6-0.79 as strong and 0.8-1 as very strong correlation.

In figure 4, we color the pairs that have strong or very strong correlation. Specifically, AN has strong or very strong correlation with other 4 features, AF, BF, CF, and CORR.

Figure 5 shows the Pearson correlation coefficient values between label and features. With large absolute values, NDAI, CORR and AF might be the important features.

2 Preparation

2.1 Data Splitting

Considering the spatial properties of the data, as data points that are spatially close to each other tends to have the same labels, we need to propose some data splitting method that takes into account of the non-i.i.d (independent and identically distributed) nature of the data and try to keep the shape in whole.

The first method would be a horizontal slice. Depending on the percentage of data split into test, validation, and training set, we apply the same percentage on the y-coordinate and slices the data. Figure 6 on the left shows how the data in image1 is sliced into 20% test, 10% validation, and 70% train. By splitting the data with this method, we reserved the integrity of the image. The individual test, validation, and train data set after splitting are shown in Figure 7.

The second method would be slices radiated from the center of the image, much like slicing a pizza (Figure 6 right). Assume the slices provided 20% test, 10% validation, and 70% train data. We first find the center of the image, then draw a line linking the center and the upper left corner of the data. This would be the place for the first slice. Then, we start from the upper left corner, walk along the perimeter of the image clockwise, until the distance reached 20% the length of the perimeter. This spot will then be connected to the center for another slice. The region that is now sliced out would be the test data. Lastly, we continue to walk along the perimeter until another 10% distance is covered. Align this spot with the center for another slice, and the new region that

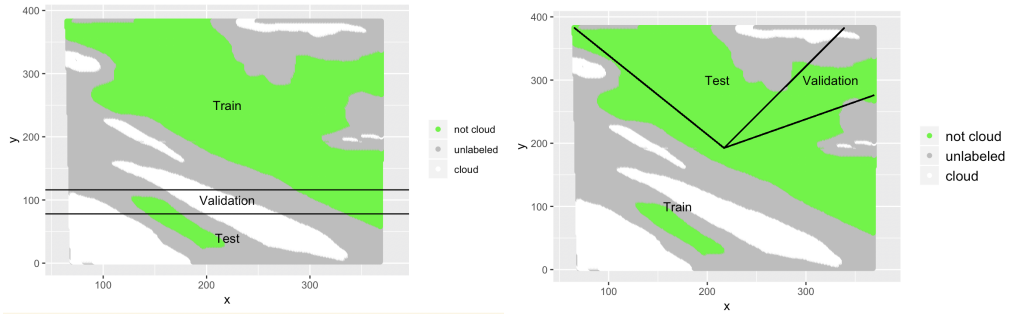

Figure 6: Left: Method 1; Right: Method 2

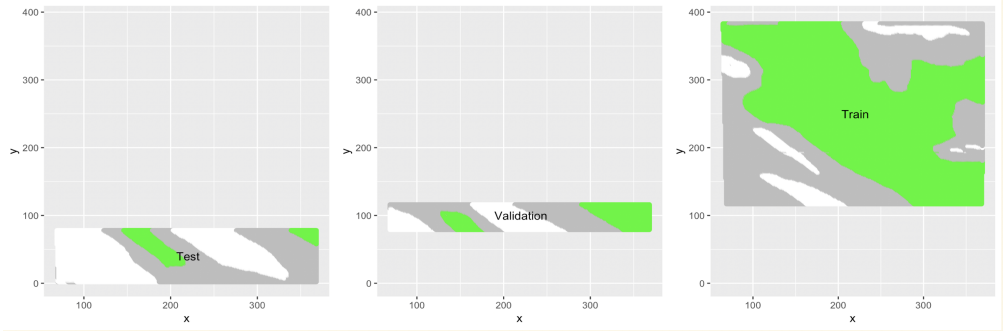

Figure 7: Data Split with Method 1

is sliced out would be the validation data. The piece left would be the training data.

The important assumption to make method 2 work is that the data is roughly square shaped. By connecting points on the perimeter with the center point, we create small triangles with areas proportional to the length it took on the perimeter, since their heights are the same (since the radius remains the same in squares). With this way, we can slice out the test, validation, and train data with the exact percentage of 20%, 10%, and 70%, as shown in Figure 8.

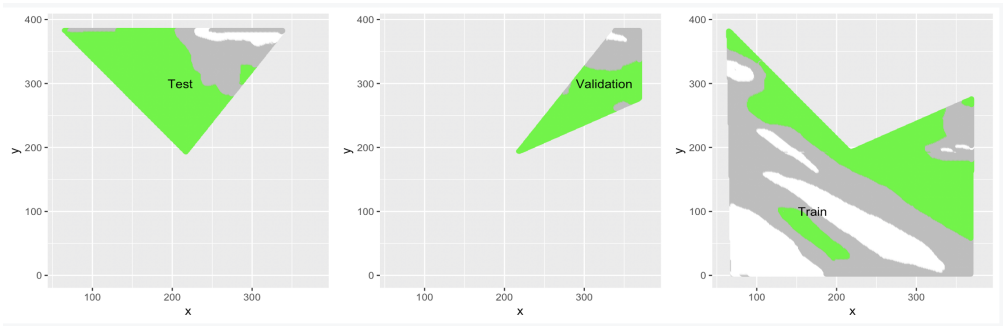

Figure 8: Data Split with Method 2

2.2 Trivial Classifier

For a trivial classifier which sets all labels to -1, its accuracy is shown in Figure 9, this would be the baseline for later analysis. Since the trivial classifier sets everything to -1, it will have a high average accuracy when the expert labels are mostly -1, which means the data is skewed.

	Test Acc 1	Val Acc 1	Test Acc 2	Val Acc 2
image1	0.0811316551570506	0.258947368421053	0.71420948315292	0.730692711159342
image2	0.312040952592922	0.280877192982456	0.535880601572639	0.738482507950931
image3	0.337077649995542	0.296754385964912	0.0581643543223052	0.457587267136914

Figure 9: Accuracy of a Trivial Classifier

2.3 Best Features

We calculated the Gini Index on our data to see dividing on which variable can provide most information (the smaller gini index the better). Three of the "best" features are NDAI, CORR and AN with Gini Index 0.17968, 0.2841701 and 0.3499206. On the left hand side of figure 10, we can see that with cut off value 0.76, NDAI has good performance to separate label 1 while CORR and AN perform well to split label -1 with cut off values 0.76 and 167.15 respectively.

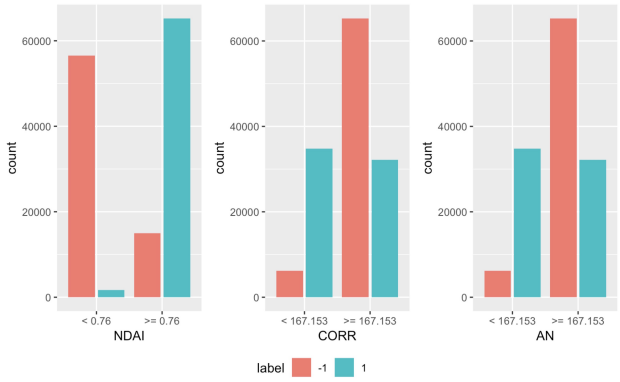

Figure 10: Important Features

2.4 Cross Validation

For each method, we implemented a general cross-validation function to do k-fold on the data with the k provided by the user, as well as the modeling function, training features, training labels, loss function, and a test data. The function trains different models according to each fold's training data, and uses its validation data to determine the best final model. The accuracy of each CV and on the test data is outputted for further review. Technically the test data set should not appear in a CV function, but for the sake of later model comparisons, we still require an input for test data and only to see how the best model across CVs performs.

Figure 11 shows the plots of the training data (removed of label 0) for CV after it has been divided into 10 folds using Method 1, the horizontal slice. For each round of cross validation, one fold of data is used a validation data, and the rest are combined to provide training data. All the three images are split separately but combined together to create validation and training data.

Similarly, Figure 12 shows the plots of the training data (removed of label 0) for CV after it has been divided into 10 folds using Method 2, the "pizza" slice. Three images are also processed separately and combined later to make the full validation and training data.

The code for CV can be seen at:
https://github.com/estherwu211/stat154_project2/blob/master/CVgeneric.Rmd

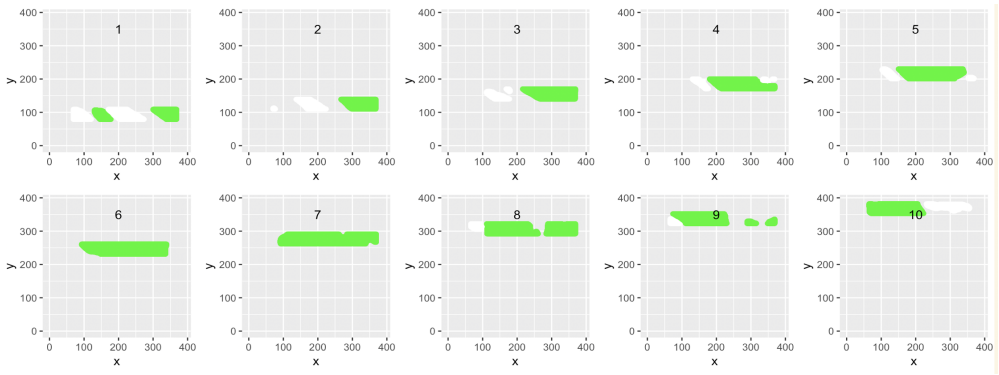

Figure 11: Validation at Each Fold for Method 1

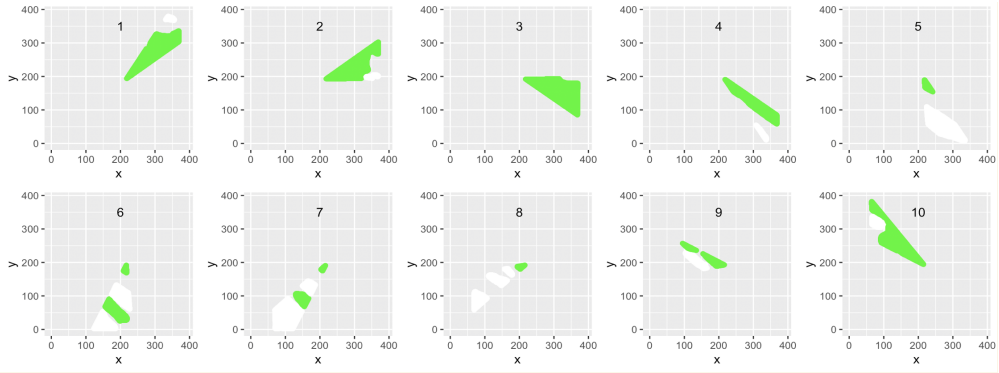

Figure 12: Validation at Each Fold for Method 2

3 Modeling

3.1 Logistic Regression

A logistic regression is trained as a simple and basic model for classification, mainly serving purpose as a baseline model, in addition to the baseline we calculated in 2.2. Since logistic regression separate values with a linear line, it might not work as well as other methods that we will talk about later. All accuracy across folds for two dividing methods are shown in Figure 13. Overall, logistic regression performs better than baseline. We also used ROC curves to find the optimal cutoff value for this model, and the accuracy after the updated cutoff value is shown in the lower two rows of Figure 13. The ROC curve and the chosen cutoff value by Youden index is shown in Figure 14. The new cutoff value is 0.02 for method 1 and 0.97 for method 2 in terms of probability.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
CV acc Method 1	0.534303789693508	0.322594410070408	0.381725584182145	0.423995996496935	0.363395552270067	0.389086069210293	0.382434154630416	0.531664292757752	0.856357927786499	0.965490144718748
CV acc Method 2	0.229613733905579	0.326158940397351	0.340702392612285	0.202269421006692	0.465139231582658	0.729409685401202	0.887838370195284	0.800414364640884	0.823645126389299	0.46953427790838
CV acc Method 1 New Cutoff	0.534303789693508	0.322594410070408	0.381725584182145	0.423933441761541	0.363025934824124	0.37888198757764	0.336183092608326	0.369288922975786	0.677066980638409	0.966406915067776
CV acc Method 2 New Cutoff	0.813417664332505	0.632023071993164	0.670257917074763	0.443642711667152	0.412900951709552	0.621185342288206	0.754234903483595	0.393905386740331	0.519720664896233	0.40676027701887

Figure 13: Accuracy of Logistic Regression

We can see that the new cutoff provides a slightly better accuracy when the data is split by method 2, mainly because the shape of each "slice" in method 2 makes it hard for logistic regression to divide it with its normal cutoff, and a specialized cutoff will be helpful in this situation.

Eventually, the test accuracy with method 1 is 0.6216 and 0.8702 for method 2.

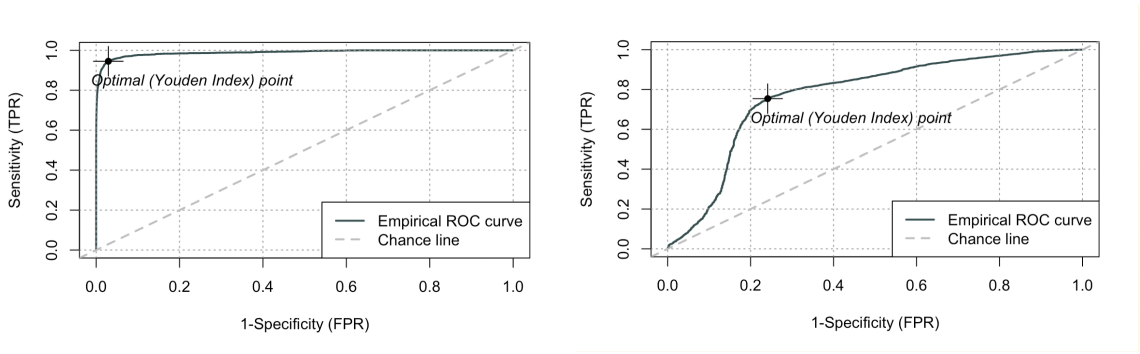

Figure 14: ROC of Logistic Regression (Left: Method 1; Right: Method 2)

3.2 Support Vector Machine

SVM works well with unstructured and semi-structured data like images. In this case, we find that SVM has similar performance with logistic regression. The CV accuracy of method 1 and method 2 with or without new cutoff are close to what of logistics classification. The optimal cutoff values method 1 and method 2 are 0.006530352 and 0.9856693 respectively, with test accuracy of 0.6837 and 0.8103.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
CV acc Method 1	0.534303789693508	0.322594410070408	0.381785500299581	0.423995996496935	0.379720322799236	0.49234693877551	0.468298640611725	0.596526788649063	0.877223966509681	0.954881802108572
CV acc Method 2	0.429353964309916	0.422505874813074	0.409635744601464	0.219319173697992	0.43729291505111	0.722929185813597	0.89791209409646	0.887776243093923	0.849021343562506	0.46464197217104
CV acc Method 1 New Cutoff	0.534303789693508	0.322594410070408	0.381725584182145	0.423995996496935	0.363025934824124	0.38082298136646	0.338678844519966	0.362441108798072	0.613095238095238	0.95350664658503
CV acc Method 2 New Cutoff	0.968997063474136	0.893024994659261	0.878503801128679	0.672563281931917	0.415227352837504	0.612937433722163	0.786200686589003	0.390193370165746	0.514507721058326	0.402821017853739

Figure 15: Accuracy of SVM

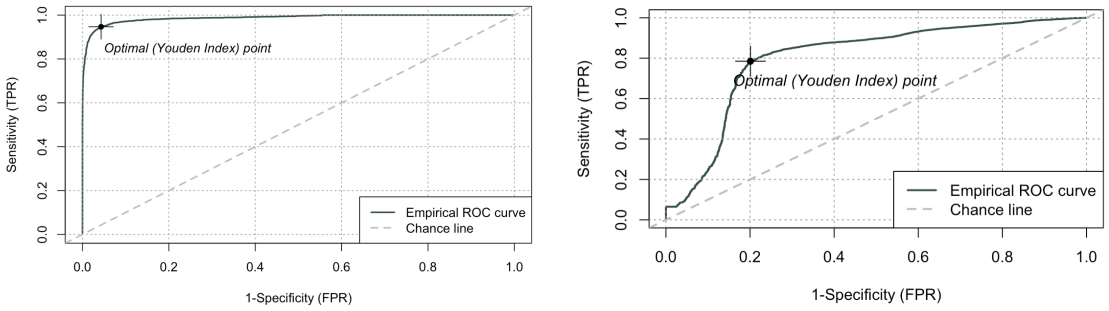

Figure 16: ROC of SVM (Left: Method 1; Right: Method 2)

3.3 Quadratic Discriminant Analysis

From logistic regression, we know that a model might perform better if the lines that separate each cluster is non-linear. As a result, we ran a QDA model on the data. The cutoff values are also updated using a ROC curve shown in Figure 18, which are 0.0139 and 0.9995 for method 1 and method 2 respectively. The resulting accuracy is shown in Figure 17.

Similar to logistic regression, we can see that the new cutoff only improved accuracy for method 2, and method 1 generally has better CV accuracy. However, QDA did show a better result than

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
CV acc Method 1	0.556292611018979	0.746106251333476	0.795686039544637	0.856812210684349	0.842666173843405	0.887034161490683	0.961501699235344	0.952339213323107	0.926412872841444	0.974526881016305
CV acc Method 2	0.141630901287554	0.166577654347362	0.176717503847768	0.195286587139948	0.36559746210786	0.627253446447508	0.761044515729641	0.730058701657459	0.854726074554933	0.765169324607663
CV acc Method 1 New Cutoff	0.559056569006818	0.748595405732167	0.800239664469742	0.866946077818091	0.852153021622621	0.895962732919255	0.96946686491079	0.946477484386984	0.911891679748823	0.966996267435008
CV acc Method 2 New Cutoff	0.254574203749718	0.284875026703696	0.274567417564479	0.301600232761129	0.352767007402185	0.554907505596795	0.723563509482807	0.61766229281768	0.772007475164749	0.789694389732512

Figure 17: Accuracy of QDA

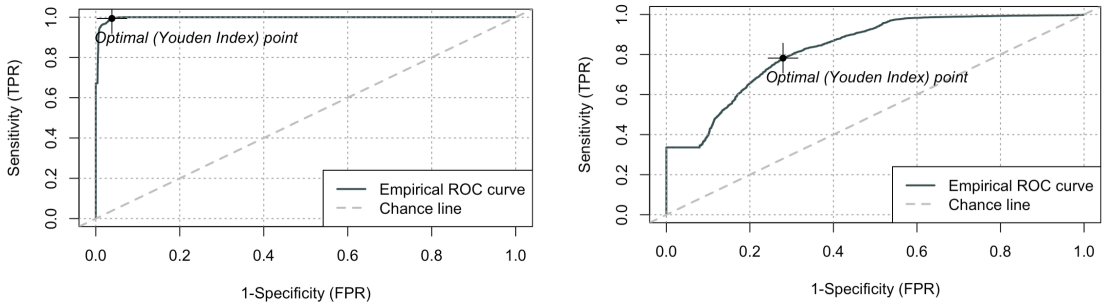

Figure 18: ROC of QDA (Left: Method 1; Right: Method 2)

logistic regression, which means that we are right about the dividing lines being non-linear. In the case of QDA, method 2 cutoff works better as it preserves the shape of the label chunks better.

The resulting test accuracy is 0.5045 and 0.903 for method 1 and 2 respectively. Surprisingly, QDA works better on method 2 with test data.

3.4 Random Forest

After all the parametric methods above, we have decided to run a method that introduces more variance for less bias, which is random forest, hoping that it can provide better accuracy and better results in method 2, as it doesn't divide data by lines. A ROC curve is also ran to choose for a better cutoff value as shown in Figure 20, with accuracy shown in Figure 19. The best cutoff value is 0.104 for method 1 and 0.85 for method 2.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
CV acc Method 1	0.745470179964376	0.653367470307944	0.762492510485321	0.822532215688728	0.789502864535206	0.814607364685004	0.844573067119796	0.796318615098061	0.909209837781266	0.967913037784035
CV acc Method 2	0.840919358482042	0.582461012604144	0.652674781959797	0.311027058481234	0.331194924215721	0.721750913161306	0.910462040632562	0.940866712707182	0.890134749680338	0.668657475061948
CV acc Method 1 New Cutoff	0.595725078312143	0.404309793044591	0.606890353505093	0.632053046415614	0.555658227068318	0.448147737355812	0.380841121495327	0.253314342062014	0.169871794871795	0.954750834915854
CV acc Method 2 New Cutoff	0.910492432798735	0.6105533005768	0.67063103400028	0.400931044515566	0.340289037715897	0.696712619300106	0.560695593449266	0.759582182320442	0.85315235566047	0.635872672977953

Figure 19: Accuracy of Random Forest

We can see the accuracy is generally better in random forest models compared to logistic, SVM and QDA. This could be due to the fact that random forest creates various decision trees to decrease bias. Similarly, the new cutoff value also helps with method 2's accuracy, mainly because of its unusual shape.

Random forest received a test accuracy of 0.6313 for method 1 and 0.9343 for method 2.

3.5 ROC Comparison

For each ROC plot, a cutoff value is chosen by Youden Index, or Youden's J Statistics. It has the following definition:

$J = sensitivity + specificity - 1 = TruePositiveRate - FalsePositiveRate$

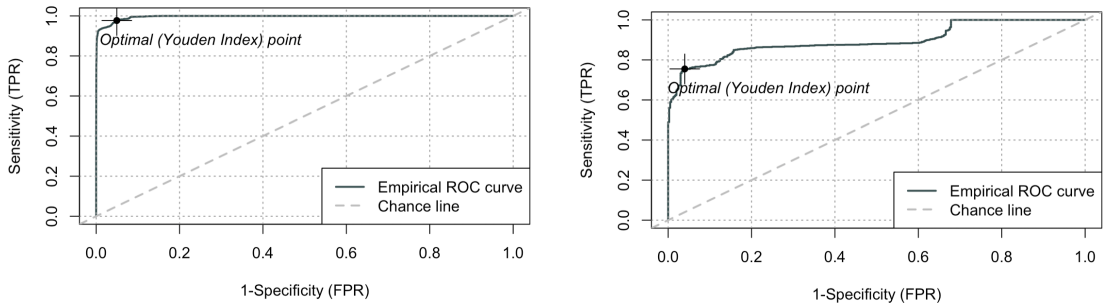

Figure 20: ROC of Random Forest (Left: Method 1; Right: Method 2)

From the plots for ROC shown previously, we can see that generally, method 1 has a nice ROC curve and AUC compared to method 2. This could be because the data is split in a "weirder" way in method 2. Within ROC plots for method 2, the ROC plot for random forest looks the best.

The cutoff point with maximum Youden Index is chosen as the new cutoff value for each model. With such new cutoff, we re-predict the labels of different models on validation set and training set. The results are shown in 3.1 - 3.4 section.

3.6 Another Metric

Figure 21 shows the accuracy in line plots. From the plot, we can see that the random forest model of the 8th fold using method 2 has the highest CV accuracy rate and test accuracy rate. Also, the plots show that the accuracy has a greater change after the cutoff value is updated on the models trained on data split by method 2 compared to method 1.

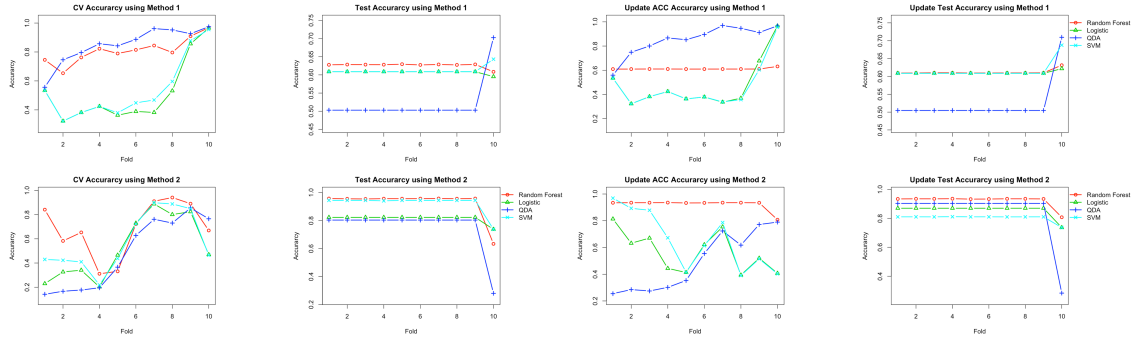

Figure 21: Comparison of Accuracy between Models

4 Diagnostics

4.1 Analysis on Random Forest

Printing the model shows an OOB estimate of error rate 2.84%. The training model fits the training data well. The class error of classify non-cloud is only 0.06. Compared to cloud, the model performs better on classifying non-clouds. The CV error and test error are 0.06 and 0.04 respectively. The variance importance plot indicates what variables had the greatest impact in the classification

model. Three of "best" features are BF, NDAI, and DF, which are different from the conclusion we have from section 3.3. Since we do not have enough data on clouds (most of the data are labeled as non-cloud), the random forest model is unstable. Figure 22 also plots one of the trees using 5 variables in construction. It turned out to be 11 terminal nodes with misclassification error rate: 0.0493.

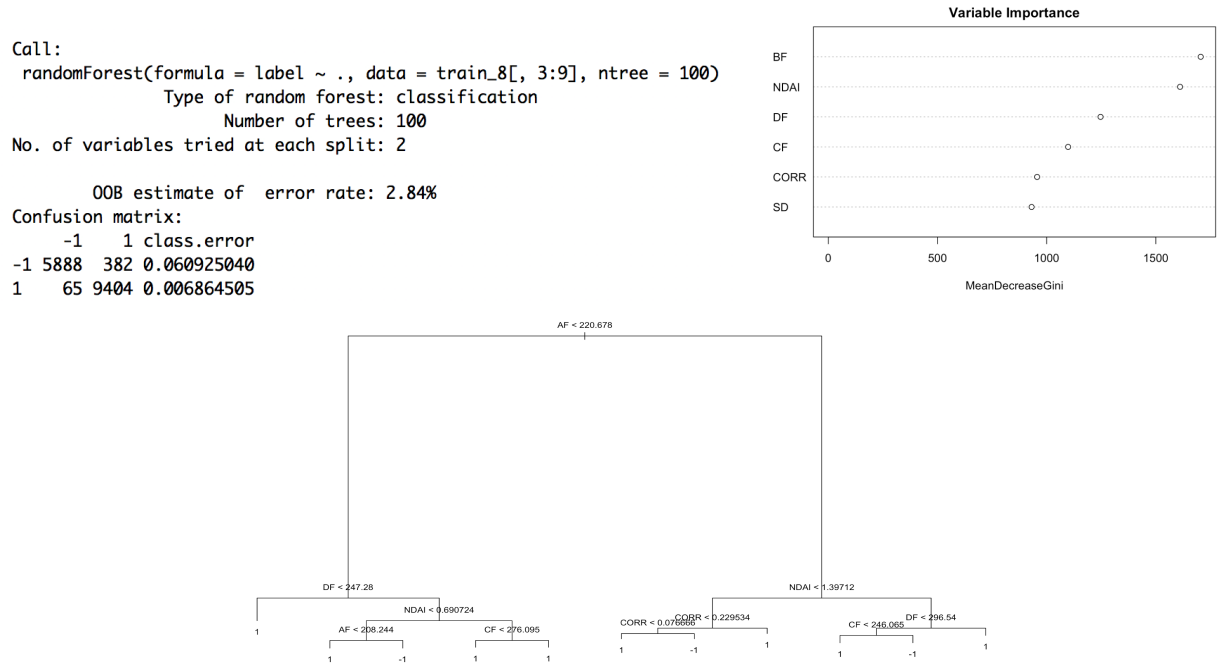

Figure 22: Random Forest of the 8th fold

4.2 Misclassification

With the best classification model, we took its prediction result on test data, and computed a confusion matrix in Figure 23 left. From the confusion matrix, we can see that the model does a good job in predicting land, with most of its error in predicting clouds. In fact, its error in predicting clouds is almost 25%!. This could be that unlike the similar, flat, icy land, clouds come in various shapes and forms, making it harder to predict.

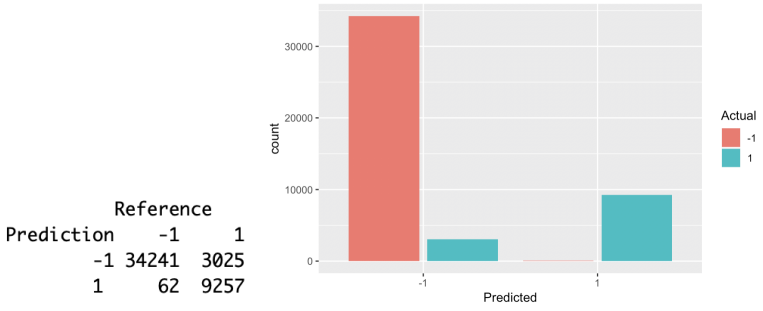

Figure 23: Left: Confusion Matrix, Right: Histogram

We further plotted out the location of classification error to see if it is easier to misclassify

specific places compared to others. From the lower row of plots in Figure 24, we can see that the misclassification errors mainly occur in the edge areas. This could be that at the edge of clouds, they become more transparent and harder to distinguish from land.

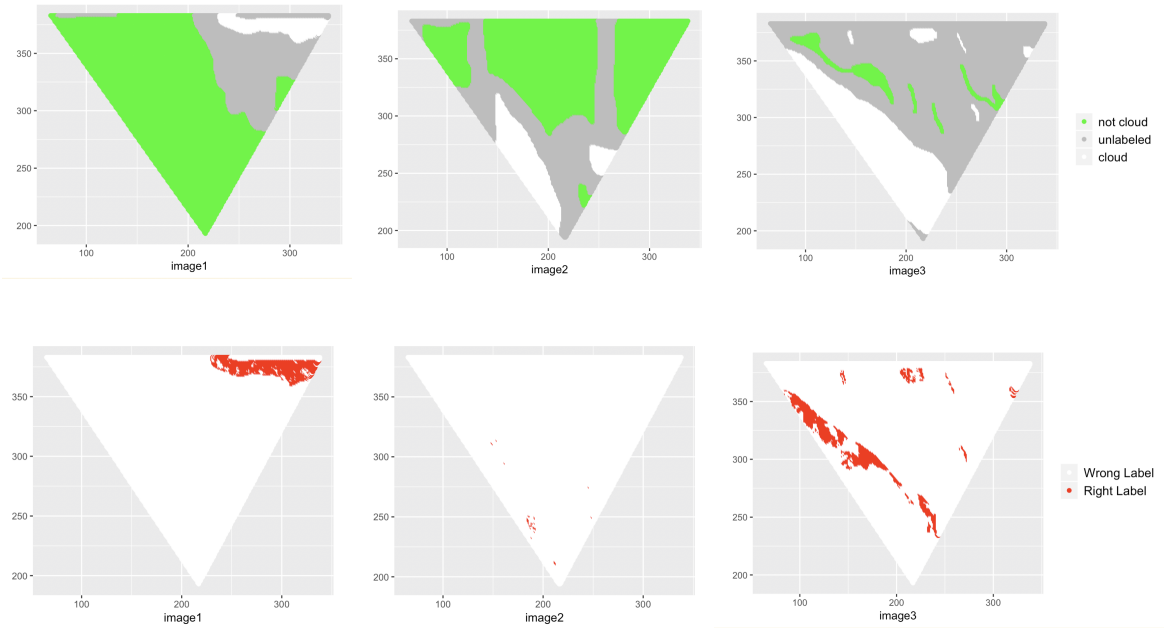

Figure 24: Location of Misclassification

4.3 Better Classifier

We find that non-parameter model, random forest, has better performance than parameter models, such as logistics, QDA, and SVM. K-nearest neighbors algorithm might be a better classifier because non-cloud and cloud appears in proximity with each other. We also need more data. We currently only have data from 3 images and the data is also unbalanced with high proportion of non-cloud pixels and low proportion of could pixels. Considering the classification performance mentioned in this paper, the new model might also suffer from the same problem: it can better classify the non-cloud labels than the cloud labels.

4.4 Change in Data Splitting

Figure 21 shows that using method 1 to split data gives lower CV and test accuracy. The reason might be that splitting data through method 2 ensures continuity in space. The long bars provided by method 1 splits data that have similar y coordinates but different x coordinates, while method 2 splits data that have both similar x and y coordinates together. Other methods that also has this property are to split into squares or rectangles.

When inspecting the confusion matrix and the resulting histogram from using the same model on method 1 data (Figure 25), we can see that this time, the model predicts most of the data to be clouds, which leads to a high accuracy in cloud prediction but low accuracy in land prediction. This is a further indicator that method 1 is not the optimal way for splitting the data.

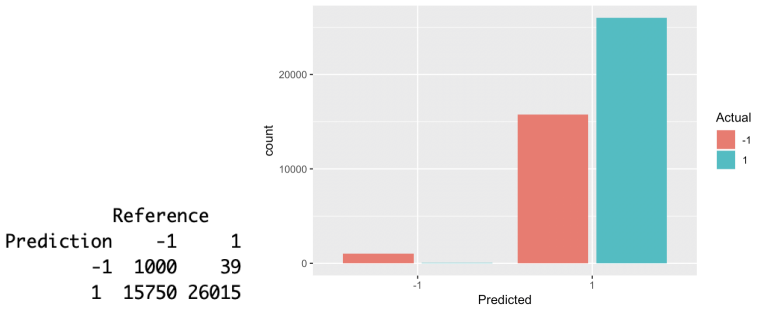

Figure 25: Left: Confusion Matrix, Right: Histogram

4.5 Conclusion

The four classification methods perform well in classifying the non-cloud labels compared to cloud labels. Among the four models, random forest provides the best result, but still suffer from the same problem. Having more unskewed data to train random forest or switch to KNN might provide better results. In terms of misclassification error, it usually occurs on the edge of cloud area.

5 Reproducibility

The full code (including R code and latex code) can be seen in this github repository:
https://github.com/estherwu211/stat154_project2
The README.md file provides details on how to run the code with new image data.

6 Acknowledgment

Work is divided equally among authors, and both authors participated in each step of the analysis.