STAT 154 Project 2
Ranzhi Xue 3031883713; Zhanyuan Zhang 3033208206

Report on Arctic Cloud Classifier Exploration

Team Classifighter

1. Data Collection and Exploration

a) Summary of Research Paper

Purpose of the Study: The goal 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 (expert labeling). By developing the new Arctic cloud detection algorithms based on existing technology, the study aims to achieve higher accuracy despite the similar remote sensing characteristics of clouds and ice- and snow-covered surfaces.
Approach: The study took an innovative approach by searching for cloud-free surface image pixels. It exploits correlations in brightnesses among multiple views of the same scene inherent under unobstructed, cloud-free conditions. The algorithms are based on three physically useful features: for characterizing the scattering properties of ice- and snow-covered surfaces the correlation (CORR) of MISR images of the same scene from different MISR viewing directions, the standard deviation (SDAn) of MISR nadir camera pixel values across a scene, and a normalized difference angular index (NDAI) that characterizes the changes in a scene with changes in the MISR view direction. The algorithm includes two steps: ELCM and ELCM-QDA. ELCM first classifies the pixels as cloud and clear, but for partly cloudy cases, results from the ELCM algorithm were used to train QDA to provide probability labels.
The Data and Collection Method: In terms of data, the data used in this study include 57 data units with 7,114,248 pixels and 36 radiation features in total were studied. Expert label for 71.5% of the data were also included to evaluate the performance of our proposed methods and existing MISR operational algorithms. The data were collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and Baffin Bay over a span of 144 days. MISR collects data from all paths on a repeat cycle of 16 days, at a rate of 3.3 megabits per second on average, with a peak rate of 9.0 megabits/second.
Conclusions: The study concluded that CORR, SD, and NDAI contain sufficient information to separate clouds from ice- and snow-covered surfaces. ELCM and ELCM–QDA based on these features provide the best performance to date among all available operational algorithms using MISR data.
Potential Impact: First, it promotes the participation of statisticians in data processing and analysis of weather and climate studies. Second, it demonstrates the power of statistical thinking, and also the ability of statistics to contribute solutions to complex modern scientific problems. By improving understanding of visible and infrared radiation through the atmosphere, the research has meaningful implications on response of clouds to changes in Arctic climate and their feedback on it, more accurate global climate model simulations, and eventually how changing cloud properties may enhance or ameliorate any initial changes in the Arctic brought about by increasing concentrations of atmospheric carbon dioxide.

b) Data Summary and Plots

First, we take a look at the 3 images separately, For image 1, we can see that about 17.77% of the pixels are labeled as "cloud", about 43.78% are "no cloud", and 38.46% are "unlabeled". For image 2, about 34.11% of the pixels are "cloud", 37.25% are "no cloud", and 28.64% are "unlabeled". Lastly, for image

Table 1: Percentage of Pixels by Class

	Image 1	Image 2	Image 3	Combined
Cloud (1)	17.77%	34.11%	18.44%	23.43%
No Cloud (-1)	43.78%	37.25%	29.29%	36.78%
Unlabeled (0)	38.46%	28.64%	52.27%	39.79%

3, about 18.44% are "cloud", 29.29% are "no cloud", and 52.27% are "unlabeled". From the data, we can see that image 1 has clearly more "no cloud" pixels compared to "cloud" pixels, while the other two images are more balanced in the proportion of cloud versus clear pixels. The three images all have a fair amount of unlabeled pixels (from about one third to above), with image 3 having the most unlabeled pixels (more than half). These results are illustrated in Table 1 and aligns with Figure 4 in the research report.

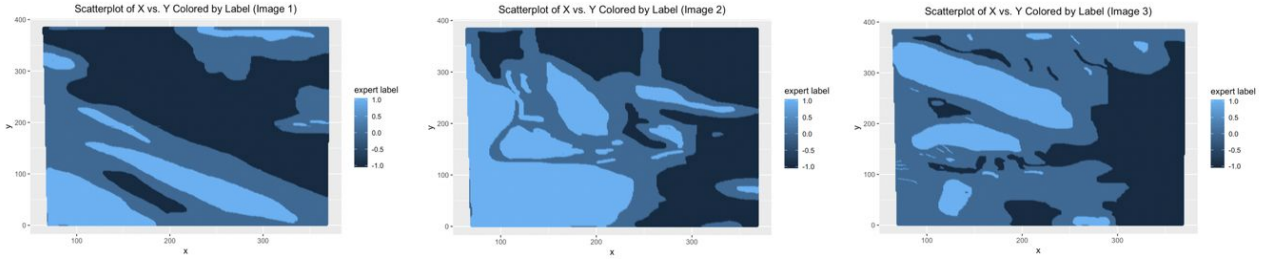

Figure 1: Scatterplots of Pixel Coordinates Colored by Expert Labels

To better understand the spread and dependence of the data, we plotted 3 graphs of x versus y colored by expert label for the 3 images. In the maps above, light blue represents cloud pixels; medium blue represents unlabeled pixels; and dark blue represents clear pixels. From the maps, we can see that there's spatial dependence of the pixels by class as appeared in all three images. The cloudy and clear pixels tend to cluster together instead of spreading out randomly. Thus, the samples would not be justified to have an iid assumption in the dataset. This has implications on how we proceed to split the data for further analysis and model training, which will be illustrated in later sections.

c) Exploratory Data Analysis

In this section, we will explore the relationship of features among themselves and with expert labels. We focus on radiance angles (DF, CF, BF, AF, AN), NDAI, SD, and CORR because the x, y coordinates are locations specific to each image, which are not as informative for analysis as the other features. First, we take a look at the pairwise relationship between features themselves.

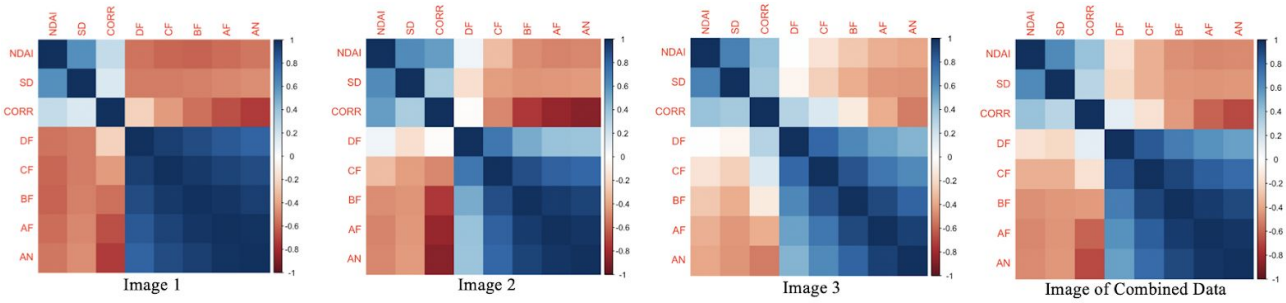

Figure 2: Correlograms of Feature Correlations

After qualitative and quantitative exploration with plots and correlation computed, we see from Figure 2 that overall image 1 has the strongest correlation among features, followed by image 2, and then image 3. What is consistent among all images is that radiance angles (DF, CF, BF, AF, AN) generally have strong correlations among themselves compared to the correlations among NDAI, SD, and CORR themselves. In

other words, the three features proposed by the research report are relatively more independent with each other compared to the raw features. More specifically, for image 1, there's a moderate correlation between NDAI and different radiance angles (DF, CF, BF, AF, AN). CORR has a moderately strong correlation with BF, AF, AN, but weak correlation with NDAI or SD. The radiance angles (DF, CF, BF, AF, AN) all have a strong correlation with each other. For image 2, there's a moderate correlation between NDAI and SD, CORR, and some radiance angles (BF, AF, AN). CORR has a strong correlation with BF, AF, and AN. Most of the radiance angles (CF, BF, AF, AN) have a strong correlation with each other. For image 3, there's a moderate correlation between NDAI and SD or CORR. The correlation between these features and the radiance angles is weaker than among the radiance angles themselves.

	NDAI	SD	CORR	DF	CF	BF	AF	AN
cor_image1	0.6591295	0.3324615	0.1448060	-0.427776963	-0.43994562	-0.43874824	-0.4153346	-0.3838326
cor_image2	0.6825384	0.3509872	0.6922682	0.260879825	-0.21740910	-0.45947552	-0.5258646	-0.5167218
cor_image3	0.4988792	0.2359717	0.3427449	0.142413840	0.02168176	-0.05722629	-0.1284194	-0.1729043
cor_combined	0.6169346	0.2954477	0.4440592	0.006550085	-0.20827917	-0.33794850	-0.3897410	-0.3893588

Table 2: Correlation of Features with Expert Labels

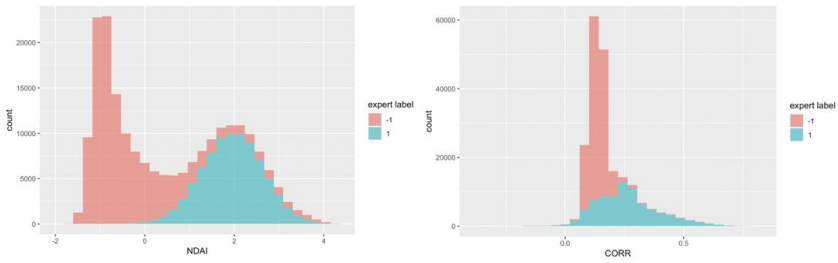

Figure 3: Histogram of Features by Expert Labels

Next, we look at the relationship between expert labels and individual features. From Table 2, we can see that the highlighted cells for each image are the top 3 features with the strongest correlation with expert labels. Across images, NDAI and CORR have the strongest correlation with expert labels, followed by AF. We also plotted histograms after removing unlabeled data to see if there are any differences between the two classes based on the features. By observing the graphs, we noticed there is clearer separation in data distribution by class based on NDAI and CORR (Figure 3) than the radiance features. It helps us understand how features contribute to differences in classes (classification).

2. Preparation

a) Data Splitting

Before we splitted the data, we first conducted data cleaning by removing x, y coordinates because they are location coordinates specific to each image and might not be helpful in modeling to predict the class of future data. We also removed unlabeled data for training models using expert labels as the true classes. Note that for later modeling purposes, after removing unlabeled data, we changed the label of "clear" from -1 to 0. Therefore, in the following sections, **class 0 represents "clear" pixels (originally class -1).**

We assume that the future data are represented as an unlabeled image. Since randomly sampling assume that pixels are independent and identically distributed, it fails to capture the spatial dependency among the pixels (i.e. pixels around an cloud pixel are more likely to be cloud). To take the spatial dependency into

account and make sure the validation and testing data are closer to future data, we suggested the following two ways:

1) Sampling from blocks: Divide each raw image into a 10-by-10 grid so that each image is divided into 100 blocks. For each image, sample 10 blocks for validation and another 10 blocks for testing. The rest of the blocks are training data. Figure 4 visualizes pixels for validation and testing on the original images.
Reasoning: The validation and testing blocks using this method can more closely mimic future data, and thus may give more accurate diagnostic statistics like validation or testing error rate.

2) Sampling after blurring: For a given image, blur it by replacing pixels to super-pixels. To do this, slide a 3-by-3 window through the image with stride 3. Taking average across the features and the majority vote as the class label among the 9 pixels in the window generates a super-pixel. The location of a super-pixel is the same as the location of the center-pixel. After generating super-pixels, randomly samples 10% for validation, 10% for testing, and the rest for training. Figure 5 shows blurred images.
Reasoning: This method smoothes the data, reducing the noise by the systematic error such as the variance of measurement and possibly some wrong-labeled pixels along boundaries of cloud and cloud-free pixels. Furthermore, since 9 surrounding pixels generate one super-pixel, spatial dependency in a blurred image is weaker than in raw data. Reducing the number of data further reduces this dependency.

After finishing the splitting on each image, we merged the training data from each image to form the final training set, and did the same thing to the validation sets and testing sets from each image. The assumption behind doing so was that the images are independent of each other and so combining them won't change the distribution of data in the validation and testing set too much.

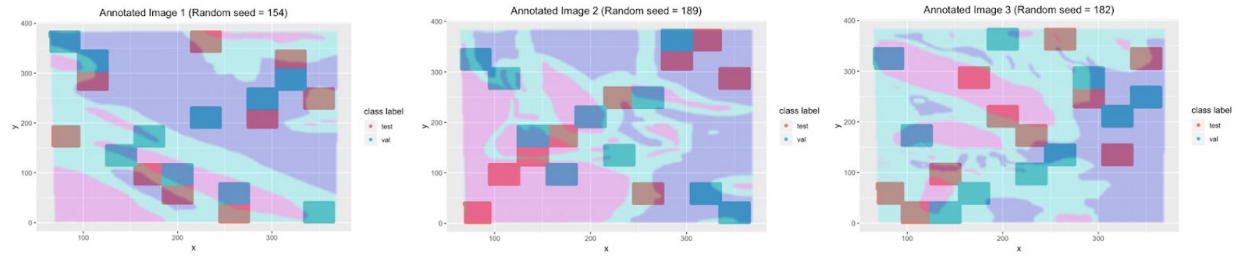

Figure 4: Image Visualization of Splitting by Blocks

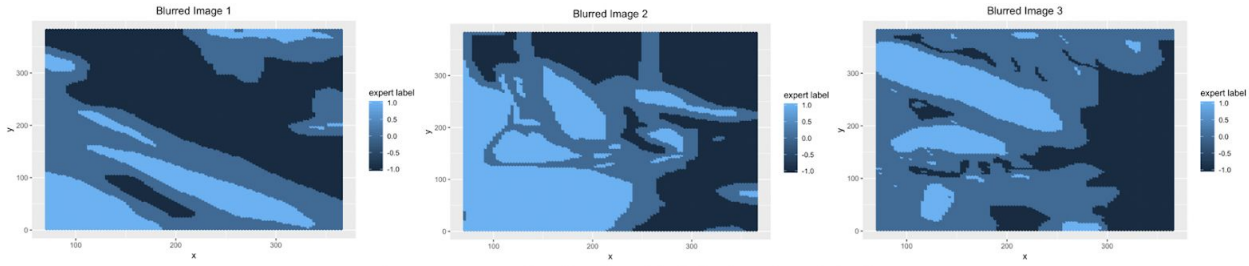

Figure 5: Image Visualization of Splitting by Blurred Pixels

b) Baseline - Trivial Classifier

For a trivial classier that sets all labels as cloud-free, the validation accuracy is 0.5568391, and test accuracy is 0.5620155. Since the baseline model blindly classifies all the pixel to be unclouded, it gives high average accuracy when predicting pictures in clear days or cloud-free locations.

c) First Order Importance - Best Features Selection

To select the best 3 features, we look at: a) Relationship between feature and label (using correlation and mean difference); b) how well each feature predicts the labels as a classifier (using training accuracy); and c) how much new information an additional feature can provide (through the correlation among features). For all of the above comparisons, we used standardized training data.

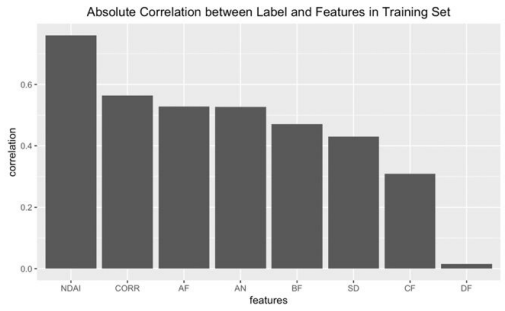

Figure 6

Feature <fctr>	Absolute Mean Difference (Standardized) <dbl>
NDAI	1.56384013
CORR	1.16049959
AF	1.08721365
AN	1.08391120
BF	0.96919280
SD	0.88451767
CF	0.63625456
DF	0.03074319

Table 3: Absolute Mean Difference by Class for Features in Standardized Training Data

a) We first calculated the correlation between different features and the expert labels. Figure 6 ranked the correlations from highest to lowest, and the 3 features with the highest correlations to labels are NDAI, CORR and AF. We also looked at the distribution of features by class (similar to Figure 3), from which we further computed the absolute mean differences between different classes of each feature. From Table 3, we can see the top 3 features in terms of mean difference aligns with the results using correlation, which are NDAI, CORR, and AF.

accuracy_NDAI	accuracy_CORR	accuracy_SD	accuracy_AN	accuracy_AF	accuracy_BF	accuracy_CF	accuracy_DF
0.9023481	0.8507133	0.840452	0.7838988	0.7682589	0.7365729	0.67941	0.5851167

Table 4: Training Accuracy of Each Feature as a Classifier

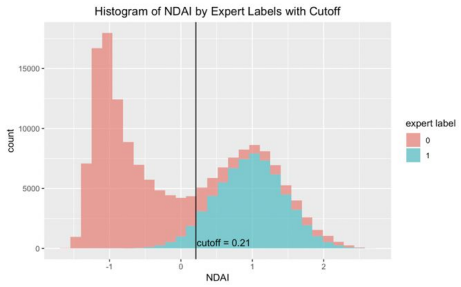

Figure 7.1

b) Next, we looked at how well each feature works as a single classifier by finding a cutoff value in the distribution of feature by class that minimizes (training) error. Table 4 shows that NDAI, CORR, and SD have the highest accuracy. Figure 7.1 provides a visualization of how the classifier and cutoff value looks like by using NDAI as an example.

c) Lastly, we looked at the correlation among features themselves. From Figure 7.2, we can see that overall the three features (NDAI, SD, CORR) chosen by the research report have lower correlation among each other than the radiance angles. This indicates that each of the three features (NDAI, SD, CORR) can contribute more new information than the randiance angles to the model.

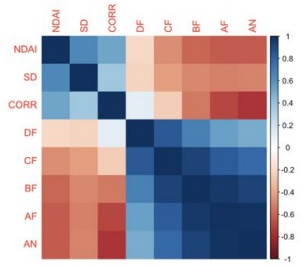

Figure 7.2: Correlogram of Standard Training Data Feature Correlations

Based on the criteria explored above, we decide to choose NDAI, CORR and SD as the 3 best features, because they work the best when models are based on them individually as classifiers, which indicates their capability to predict labels. Besides, they also have lower correlation with each other, so more information can be provided using them compared to the

radiance angles. Though SD does have a lower correlation and absolute mean difference than some of the features, but the small difference can be balanced by other criteria. **d) See Github**

3. Modeling

a) We tried LDA, QDA, Logistic Regression and Random Forest.

LDA and QDA assume the data from one class follow Gaussian distribution, and they estimate the mean and covariance matrix from the training data. LDA assumes the covariance matrix of different classes are the same, while QDA assumes that they are different. This is the reason why LDA is better when dealing with overfitting. In the EDA section, we found that many variables (like NDAI, SD, CORR) have bimodal distribution for cloud class yet most of the variables of uncloud data have normal-like distribution. This implies that the assumption of Gaussian may not be satisfied here.

Logistic regression assumes that each data point follows some Bernoulli distribution, and logistic tries to model the parameter of this distribution. Furthermore, since logistic regression is equivalent to modeling the linear relation between log-odd and features, it prefers no collinearity among the variables. In EDA section, we found that NDAI, SD, CORR have some collinearity, so do the DF, CD, BF, AF and AN. Therefore, the assumption of logistic regression is not satisfied in this case.

Random forest is an ensemble model of multiple decision trees. Since it uses a bootstrapped sample to build a decision tree it requires the sampled data is representative. Since the two sampling methods take care of the spatial dependency, this assumption is satisfied.

To pick the best threshold for logistic regression, we apply cross-validation on thresholds 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75. Turned out that 0.4 gave the best average cross-validation accuracy. To pick the best hyperparameters for the random forest, we applied grid-search on the following hyperparameters: number of sampled features: [5, 6, 7, 8], minimum size of a node: [5, 50, 500], number of trees: [10, 20, 30]. Turned out that 7 sampled features, minimum size of 500 and 10 trees gave the best average cross-validation accuracy.

The following models were trained on data split by the method **sampling from blocks**.
Table 5 shows different models' cross-validation accuracy across folds. We can see that random forest gives the highest CV accuracy across folds.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
LDA	0.8967158	0.8977510	0.8984992	0.8958845	0.8997542	0.8987607	0.8971865	0.8963498	0.9001151	0.8977041
QDA	0.8968727	0.8916898	0.8966112	0.8968835	0.9015896	0.8961874	0.8944148	0.8951992	0.8950478	0.8954498
Logistics Regression	0.9020551	0.9027821	0.9032056	0.9024161	0.9031431	0.9036762	0.9001098	0.8971811	0.9026303	0.8998013
Random Forest	0.9684656	0.9684133	0.9700868	0.9658491	0.9679967	0.9663180	0.9699854	0.9680489	0.9694577	0.9698269

Table 5: CV Accuracy Across Folds for Different Models (Split by Blocks)

lda.acc	qda.acc	logit.acc	rf.acc
0.8568015	0.8449081	0.8630668	0.8967824

Table 7: Test Accuracy on Method 1 (Split by Blocks)

Table 7 shows the test accuracy of different models. We can see that random forest also gives the highest test accuracy.

The following models were trained on data split by the method **sampling after blurring**. Table 6 shows different models' cross-validation accuracy across folds. We can see that random forest gives the highest

CV accuracy across folds.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
LDA	0.8895945	0.8895945	0.9037616	0.9120664	0.9022960	0.8905181	0.9027846	0.9061584	0.9085575	0.9046921
QDA	0.9057157	0.9071359	0.9008789	0.9031785	0.9042033	0.9076698	0.8973607	0.9101562	0.9081583	0.8992665
Logistic Regression	0.9145508	0.9115347	0.9046921	0.9081134	0.9066015	0.8973607	0.8910068	0.9130435	0.9198828	0.9096680
Random Forest	0.9477539	0.9599218	0.9608993	0.9609375	0.9521251	0.9628543	0.9540567	0.9516129	0.9560117	0.9545455

Table 6: CV Accuracy Across Folds for Different Models (Split by Blurring)

lda.acc	qda.acc	logit.acc	rf.acc
0.8994266	0.8989854	0.9056021	0.9280988

Table 8: Test Accuracy on Method 2 (Split by Blurring)

Table 8 shows the test accuracy of different models. We can see that random forest also gives the highest test accuracy.

Comparing the two sampling method, we found that even though Method 1 gave higher cross-validation accuracy in overall, Method 2 gives higher testing accuracy. This suggests that Method 1 is more likely to introduce overfitting.

b) ROC Curves

For computing cutoff values, we plotted ROC curves based on validation set; and for comparing models, we plotted ROC curves and computed area under the curve.

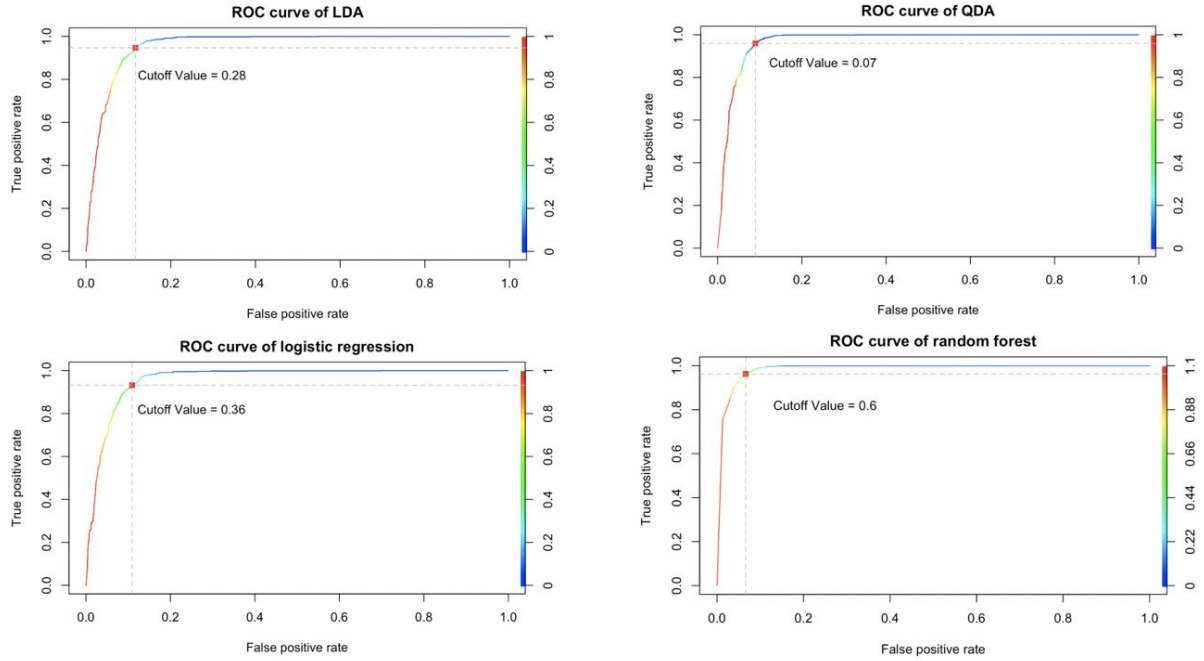

Figure 8.1: ROC Curves for Different Models with Cutoff Values (on Validation Set)

Figure 8.1 shows the ROC curves for different methods with cutoff values highlighted and labeled on each plot. Because the most ideal cutoff point will be at the top left corner where true positive rate (TP) is 1 and false positive rate (FP) is 0, for each case, we want to find the cutoff point closest to this "ideal point". Therefore, we computed the distance of the corresponding point on the ROC curve (defined at the intersection of the corresponding TP rate and FP rate) at each threshold value. The cutoff value will thus be the point where the distance is minimized.

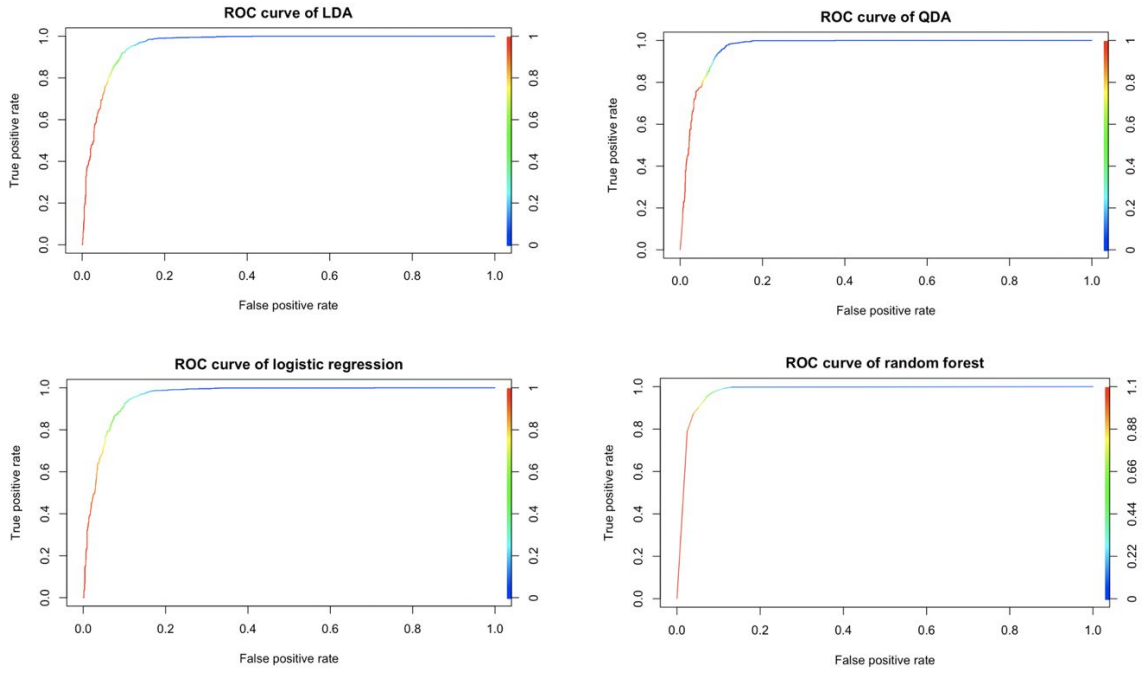

Figure 8.2: ROC Curves for Different Models with Cutoff Values (on Test Set)

AUC_lda	AUC_qda	AUC_logit	AUC_rf
0.9608413	0.968061	0.9610437	0.9776802

Table 9: Area Under the Curve for Different Models

Figure 8.2 shows ROC curves on test set. For comparing different models, we compared the area under the curve (AUC), which is an aggregate measure of performance across all possible classification thresholds. From Table 9, we see that random forest has the highest AUC, and thus the best performance according to this metric.

c) Assess the Fit Using Other Metrics

	LDA	QDA	Logistic Regression	Random Forest
Sensitivity	0.8802139	0.8556150	0.8641711	0.9593583
Specificity	0.9129129	0.9294294	0.9241742	0.9271772
Pos Pred Value	0.8764643	0.8948546	0.8888889	0.9024145
Neg Pred Value	0.9156627	0.9016752	0.9064801	0.9701493
Precision	0.8764643	0.8948546	0.8888889	0.9024145
Recall	0.8802139	0.8556150	0.8641711	0.9593583
F1	0.8783351	0.8747950	0.8763557	0.9300156
Prevalence	0.4124393	0.4124393	0.4124393	0.4124393
Detection Rate	0.3630348	0.3528893	0.3564182	0.3956771
Detection Prevalence	0.4142038	0.3943538	0.4009704	0.4384649
Balanced Accuracy	0.8965634	0.8925222	0.8941726	0.9432677

Table 10: Key Statistics Related to Confusion Matrix for Different Models

Metric 1: Confusion Matrix Related Statistics

Besides directly looking at only TP and FP, we also look at more statistics that can be computed from the confusion matrix. Table 10 shows a list of related statistics for different methods. Random Forest outperforms the other 3 methods on all of them. To give a deeper analysis on a few key statistics: 1) Recall is a measure of a classifier's completeness. By having the highest Recall among all models, random forest has the lowest false negative rate. 2) F1 score conveys the balance between precision and recall. It is helpful when there is an uneven class distribution. Since during EDA, we saw that some images, especially image 1 has an uneven class distribution (clearly more no-cloud than cloud pixels), F1 score can thus complement ROC curves in such cases to give a more balanced analysis of the models.

Metric 2: AIC

AIC_lda	AIC_qda	AIC_logistic
123955.9	220087.2	10049.89

Table 11: AIC for Different Models

Akaike Information Criterion (AIC) estimates the relative amount of information lost by a given model. It deals with the trade-off between the goodness of fit of the model and the simplicity of the model. In other words, AIC deals with both the risk of overfitting and the risk of underfitting. As shown in Table 11, we computed the AIC for each model and discovered that logistic regression has the smallest loss of data among all, which indicates a good balance between model fit and simplicity.

Metric 3: BIC

BIC_lda	BIC_qda	BIC_logistic
124580.7	220712	10121.23

Table 12: BIC for Different Models

Bayesian Information Criterion (BIC) also deals with the risk of overfitting like AIC, but with a stronger penalty term. As shown in Table 12, we computed the BIC for each model and conforming to our finding in AIC, logistics regression has the lowest BIC among all, which indicates a better balance in risks of overfitting and underfitting.
(Note that for both AIC and BIC, since they are usually applied to regression models, the limitation of Metrics 2&3 is that it's hard to compute for the random forest model for a comprehensive comparison.)

4. Diagnostics

For this part, we focus on analyzing the random forest model containing 10 trees, with 7 sampled features and minimal node size of 50 in each tree. Cross-validation gave these optimal hyperparameters. The computation of part (a) and (b) were based on sampling method 2 (sampling after blurring).

a) Analysis of Chosen Classifier

To test the robustness of the random forest model, we trained the random forest on different size of data and observed the tendency of accuracy on the testing data. The accuracy converged after the training size was greater than 17,000 (see the Figure 9 left), so we bootstrapped 17,000 samples 100 times to see how robust the random forest model could be and the plot at Figure 9 right.

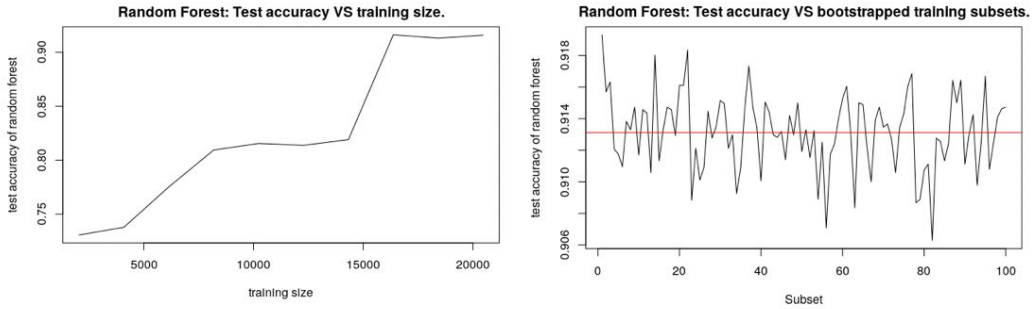

Figure 9: Diagnostic Plots for Random Forest (Split by Blurring)

The average testing accuracy is 91.31%. The testing accuracy of bootstrapping has small variance (5.834684e-6), which implies that the model is robust to the training samples.

b) Patterns of Misclassification

We first computed the confusion matrix, which shows that the model misclassified around 9% of the "not cloud" samples and around 7% of the "cloud" samples.

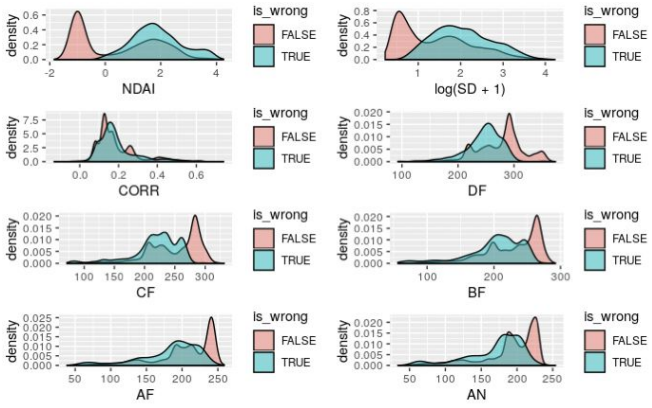

Figure 10: Distribution of Features (Split by Blurring)

Inspecting the distribution of each variable of correctly classified samples and misclassified samples (see Figure 10), we found that for those correctly classified samples, most of their variables present bimodal distributions. Meanwhile, given a variable of misclassified sample (like NDAI, CF or AN), it usually concentrates on one of the two peaks. These density plots tell us that the random forest model tend to misclassified those samples having relatively higher NDAI, or lower DF, CF, BF, AF, and AN.

label	NDAI	SD	CORR	DF	CF	BF	AF	AN
Min. :0.0000	Min. :-1.8420	Min. : 0.2498	Min. :-0.1314	Min. : 92.72	Min. : 71.45	Min. : 59.51	Min. : 45.87	Min. : 33.75
1st Qu.:0.0000	1st Qu.:-0.9720	1st Qu.: 0.8903	1st Qu.: 0.1240	1st Qu.:249.96	1st Qu.:221.34	1st Qu.:199.02	1st Qu.:190.45	1st Qu.:183.35
Median :0.0000	Median : 0.3386	Median : 2.4400	Median : 0.1602	Median :287.21	Median :264.47	Median :239.52	Median :213.64	Median :199.61
Mean :0.4458	Mean : 0.4455	Mean : 4.9568	Mean : 0.1914	Mean :277.57	Mean :251.63	Mean :228.04	Mean :204.59	Mean :191.41
3rd Qu.:1.0000	3rd Qu.: 1.7635	3rd Qu.: 6.1950	3rd Qu.: 0.2427	3rd Qu.:300.56	3rd Qu.:283.54	3rd Qu.:263.37	3rd Qu.:239.72	3rd Qu.:221.89
Max. :1.0000	Max. : 4.2543	Max. :67.3436	Max. : 0.7395	Max. :374.36	Max. :331.29	Max. :292.44	Max. :258.77	Max. :253.91

Table 13: Summary of Features – Correct (Split by Blurring)

label	NDAI	SD	CORR	DF	CF	BF	AF	AN
Min. :0.0000	Min. :-0.2698	Min. : 0.6372	Min. :-0.1304	Min. : 97.46	Min. : 74.79	Min. : 56.29	Min. : 38.31	Min. : 43.69
1st Qu.:0.0000	1st Qu.: 1.0612	1st Qu.: 3.3782	1st Qu.: 0.1223	1st Qu.:233.47	1st Qu.:207.16	1st Qu.:189.56	1st Qu.:172.78	1st Qu.:162.26
Median :0.0000	Median : 1.6883	Median : 5.8117	Median : 0.1585	Median :252.30	Median :227.35	Median :211.00	Median :195.85	Median :184.21
Mean :0.3543	Mean : 1.7106	Mean : 8.2824	Mean : 0.1797	Mean :247.01	Mean :220.93	Mean :204.16	Mean :186.99	Mean :175.02
3rd Qu.:1.0000	3rd Qu.: 2.2508	3rd Qu.:10.3301	3rd Qu.: 0.2038	3rd Qu.:266.49	3rd Qu.:245.61	3rd Qu.:233.13	3rd Qu.:215.91	3rd Qu.:201.60
Max. :1.0000	Max. : 4.0343	Max. :51.8394	Max. : 0.6299	Max. :352.63	Max. :319.54	Max. :269.65	Max. :247.26	Max. :241.31

Table 14: Summary of Features – Wrong (Split by Blurring)

Inspecting the summary of variables in Table 13 and 14, we found that some of the variables behave dramatically differently between correctly classified samples and misclassified samples: for examples, samples with NDAI ranging -0.9728 ~ 1.7635 are more likely to be correctly classified than samples with NDAI ranging 1.6883 ~ 2.2508; samples with DF ranging 249.96 ~ 300.56 are more likely to be correctly classified than samples with DF ranging 233.47 ~ 266.49 etc.. This pattern implies that in the prediction on future data, we can give more weight to those samples whose variables are within certain ranges to increase the accuracy and robustness of the prediction.

c) Finding a Better Classifier

One way of getting a better classifier is adding more features. Here we tried to add the first PC from PCA to the data matrix to see if the testing accuracy increased. Again, we inspected the testing accuracy converged after the size of training is more than 16,000 (see Figure 11 left), so we bootstrap 16,000 samples 100 times to test the robustness of the model (see Figure 11 right).

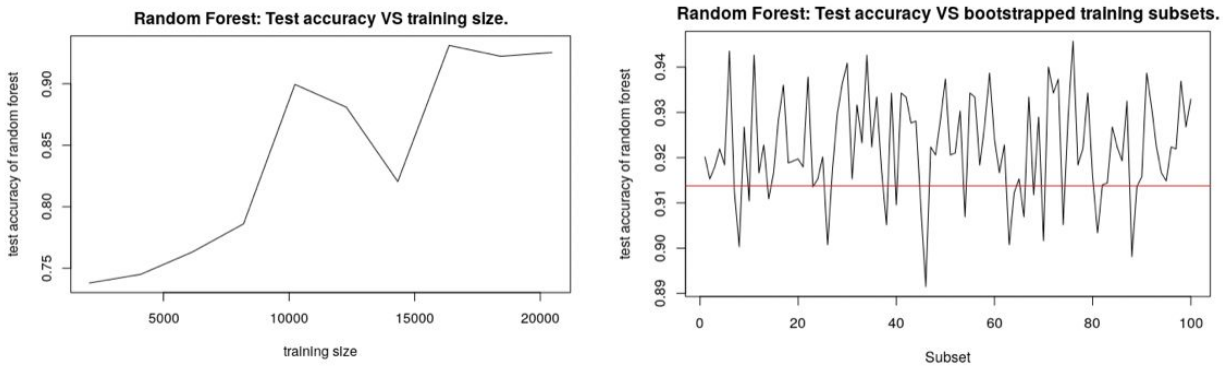

Figure 11: Diagnostic Plots for Random Forest (Split by Blurring with PCA)

The average of testing accuracy over 100 times of bootstrapping is 91.93%. Overall, adding new features helps increases the accuracy of the model a little bit. Regrading the robustness of models using two sets of features, even though more features leads to larger variance, both of them have small variance, 5.834684e-6 for using the original features and 0.0001312791 for using additional PC features. Even though the PC features add little accuracy to the classifiers, the bootstrapping shows that the model is robust.

d) Comparison with Parts 4(a) and 4(b) by Modifying Data Splitting

By using sampling method 1 (sampling from blocks), the random forest misclassified 10% of uncloud samples and 4% of cloud samples. Comparing with using method 2 (blurring), the model by method 1 is relatively easier to misclassify uncloud samples.

We found that even though method 1 produce more training data for the model, its testing accuracy needs more data to converge (see Figure X left). Since it testing accuracy converged after training size was larger than 150,000, we took 150,000 samples 100 times to see how much the testing accuracy would variate (see Figure X right). The average testing accuracy is 91.96% with variance 2.129837e-6. The model trained on data split by blocks is also robust to the training data.

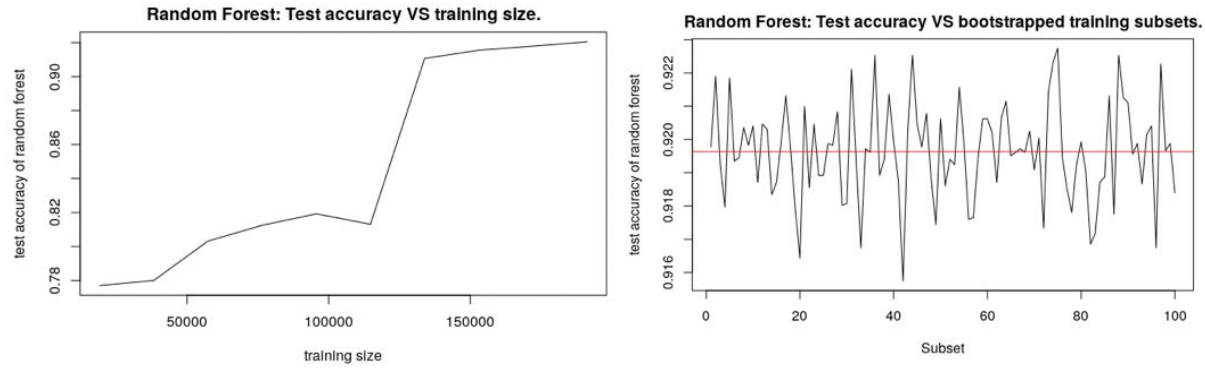

Figure 12: Diagnostic Plots for Random Forest (Split by Blocks)

Comparing the following summaries of features with the one in part (b), we found that the distribution of correctly classified samples and misclassified samples did not change too much.

label	NDAI	SD	CORR	DF	CF	BF	AF	AN
Min. :0.0000	Min. :-1.8420	Min. : 0.2498	Min. :-0.1314	Min. : 92.72	Min. : 71.45	Min. : 59.51	Min. : 45.87	Min. : 33.75
1st Qu.:0.0000	1st Qu.:-0.9697	1st Qu.: 0.8935	1st Qu.: 0.1238	1st Qu.:249.82	1st Qu.:221.56	1st Qu.:199.14	1st Qu.:190.59	1st Qu.:184.02
Median :0.0000	Median : 0.3556	Median : 2.4550	Median : 0.1593	Median :287.00	Median :263.96	Median :239.41	Median :213.77	Median :199.77
Mean :0.4529	Mean : 0.4389	Mean : 4.9226	Mean : 0.1902	Mean :277.39	Mean :251.58	Mean :228.15	Mean :204.84	Mean :191.70
3rd Qu.:1.0000	3rd Qu.: 1.7536	3rd Qu.: 6.1501	3rd Qu.: 0.2404	3rd Qu.:300.35	3rd Qu.:283.49	3rd Qu.:263.32	3rd Qu.:239.67	3rd Qu.:221.83
Max. :1.0000	Max. : 4.2543	Max. :67.3436	Max. : 0.7395	Max. :374.36	Max. :331.29	Max. :292.44	Max. :258.77	Max. :253.91

Table 15: Summary of Features – Correct (Split by Blocks)

label	NDAI	SD	CORR	DF	CF	BF	AF	AN
Min. :0.000	Min. :-0.2575	Min. : 1.085	Min. :-0.1304	Min. :105.4	Min. : 74.79	Min. : 56.29	Min. : 38.31	Min. : 43.69
1st Qu.:0.000	1st Qu.: 1.2753	1st Qu.: 3.698	1st Qu.: 0.1291	1st Qu.:230.5	1st Qu.:204.45	1st Qu.:185.24	1st Qu.:163.29	1st Qu.:151.53
Median :0.000	Median : 1.7970	Median : 6.246	Median : 0.1663	Median :251.5	Median :224.72	Median :207.16	Median :191.47	Median :179.91
Mean :0.264	Mean : 1.8878	Mean : 8.946	Mean : 0.1928	Mean :246.7	Mean :219.08	Mean :200.93	Mean :182.60	Mean :170.37
3rd Qu.:1.000	3rd Qu.: 2.4010	3rd Qu.:11.428	3rd Qu.: 0.2195	3rd Qu.:267.6	3rd Qu.:243.88	3rd Qu.:229.99	3rd Qu.:212.15	3rd Qu.:198.23
Max. :1.000	Max. : 4.1202	Max. :53.069	Max. : 0.6690	Max. :355.2	Max. :319.54	Max. :269.65	Max. :247.26	Max. :248.01

Table 16: Summary of Features – Wrong (Split by Blocks)

e) Conclusion

Regardless of the data splitting methods, random forest models are robust to the training data and give around 91~92% testing accuracy. There exists some patterns in the distribution of some features such as NDAI, DF, CF, AF, BF. These patterns can become references of weights on data to improve the quality of prediction in the future. Since all the models we tried (LDA, QDA, logistic regression, and random forest) have decent testing accuracy, combining these four models and using majority vote to generate predicted class may further improve the model. Some boosting method like Adaboost can possibly improve the model as well, which can be explored in future analysis.

5. Reproducibility, Acknowledgement and References

Github Repo Link:
https://github.com/zhanyuanucb/stat-154-project

Teamwork: Zhanyuan took care of coding for Section 2, 3(a)(b) and 4, write-up for parts of Section 2, 3, and Section 4, as well as Github Repository. Ranzhi took care of coding for Section 1 and 3(c), write-up for Section 1 and parts in Section 2 and 3, as well as formatting.

References and Outside Sources:
For this project, we want to acknowledge the help from both GSI's office hours, the help from friends in CS majors, and discussions with 2 other teams. Below is a list of references we have used:
https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/
https://www.r-bloggers.com/a-small-introduction-to-the-rocr-package/
https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
https://drsimonj.svbtle.com/quick-plot-of-all-variables