Project 2 Cloud Data

Caiyi Deng 3033303379, Winnie Gao 3031884025

May 2, 2019

1 Data Collection and Exploration

a. Summary of Paper

Climate changes has been a popular topic in scientific research. Particularly, the warming Arctic is one of the biggest stories in our times. Global climate models predict that the increasing atmospheric carbon dioxide levels is strongly related to the increasing surface air temperatures in the Arctic, where cloud plays an important role in producing more atmospheric carbon dioxide. In this paper, scientists use the Multiangle Imaging SpectroRadiometer (MISR) imagery to perform a cloud detection to ascertain whether cloud can potentially lead to further warming in the Arctic.

MISR collects a massive amount of data from its nine cameras viewing at a different angel in four spectral. It covers the daylight side of the Earth from the Arctic down to Antarctica in 45 minutes and completes all paths in 16 days of a cycle. Each path is subdivided into blocks, with the block numbers increasing from the North Pole to South Pole, and each complete trip of MISR around the Earth is counted as a unique orbit. However, due to the transmission channel constraints, only the red radiances and all channels from the nadir camera are transmitted at full 275m * 275m resolution. The remaining blue, green and near-infrared radiances from the non-nadir cameras are aggregated to a lower resolution before transmission.

Scientists utilizes correlations in brightness among multiple MISR views of the same scene under cloud-free conditions to model the surface. This new algorithm, enhanced linear correlation matching (ELCM), is based on thresholding three features: the correlation (CORR) of MISR images, the standard deviation (SD) of the MISR nadir camera pixel values, and a normalized difference angular index (NDAI) to create labels for classification. Then, the resulting labels are used in the second algorithm, ELCM-QDA, to produce more informative probability prediction.

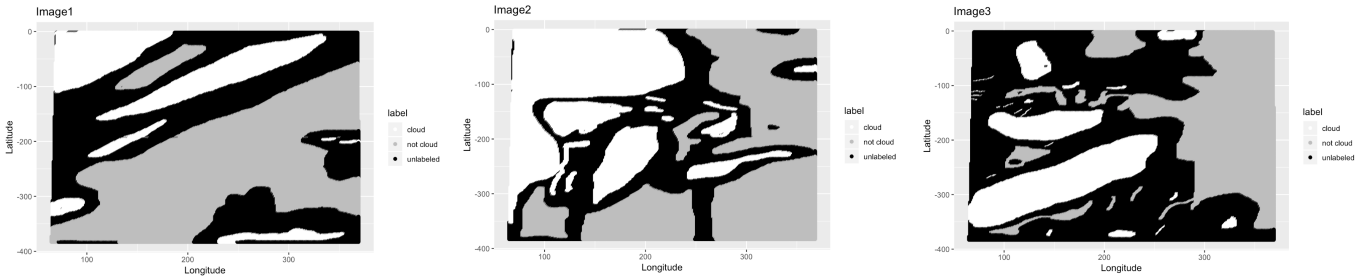
The results suggest that the ELCM algorithm based on the three features outperforms those existing algorithms based only on the radiation measurement, where it provides better spatial coverage for cloud detection in the Arctic. Moreover, the ELCM algorithm combines classification and clustering framework to fit the MISR data processing, which helps improve the computational speed online effectively.

This research not only creates a better algorithm to implement cloud detection but also encourages further study on the changing cloud properties to the warming Arctic. In addition, it demonstrates the significant impact of statistics in successfully solving a modern scientific problem. Statisticians are now directly involved in the data processing and use powerful statistical thinking to help tackle challenges.

b. Summery of Data

In image1 data set, there are 115229 data points. We calculate the percentage for pixels in two different classes: 17.77% pixles are classified as cloud, 43.78% pixels are classified as not cloud and 38.46% pixels are marked as unlabeled. In image2 data set, there are 115110 data points. 34.11% pixels are classified as cloud, 37.25% pixels are classified as not cloud and 28.64% pixels are marked as unlabeled. In image3 data set, there are 115217 data points. 18.44% classified as cloud, 29.29% as not cloud and 52.27% as unlabeled. After we combined three image data sets, there are 345556 data points in total: 23.43% classified as cloud, 36.78% as not cloud and 39.79% as unlabeled.

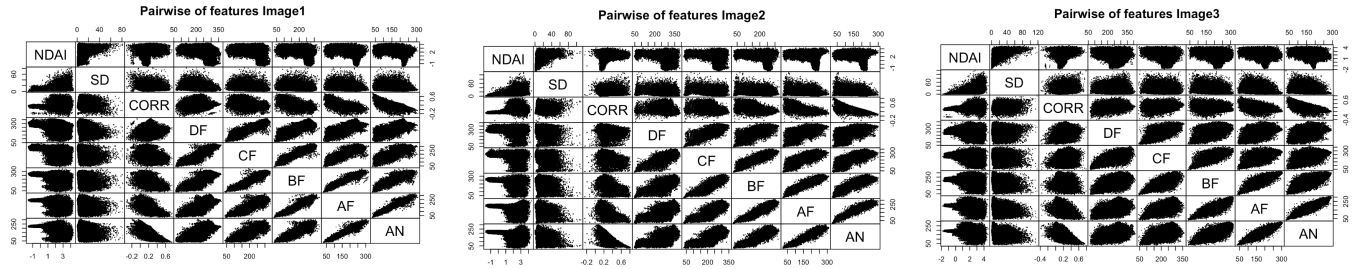
In order to view the pattern or trend for the data points, we plot scatter plots and color each data point according to their labels.



From the three plots, we can see the patterns that the pixels with same labels are more tend to connect to each other, and unmarked pixels stay around those marked as clouds. This pattern contradicts to the assumption in the paper that the samples are independent and identically distributed because the pixels adjcent to pixels marked as cloud are more likely to be marked as cloud.

c. EDA

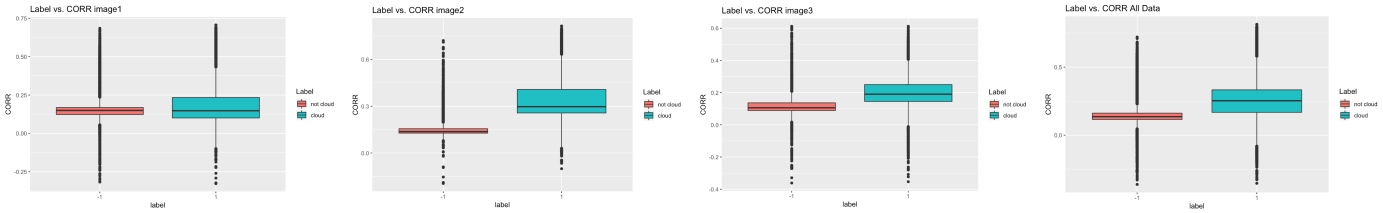
To further explore our data, we also summarize the pairwise relationship between the features themselves.



The three pairwise scatterplots all show that DF, CF, BF, AF and AN shows a linear relationship between each other, especially for AF and AN, AF and BF. However, we observe that data points in the third pairwise scatterplot spread more widely than in the first two pairwise sctterplots. The differences between three graphs can also show that the (linear) relationships decrease over time.

After checking pairwise relationship plots, we study the relationship between each independent featues and their expert labels for each image separately and together.

(relationship between expert label and NDAI)

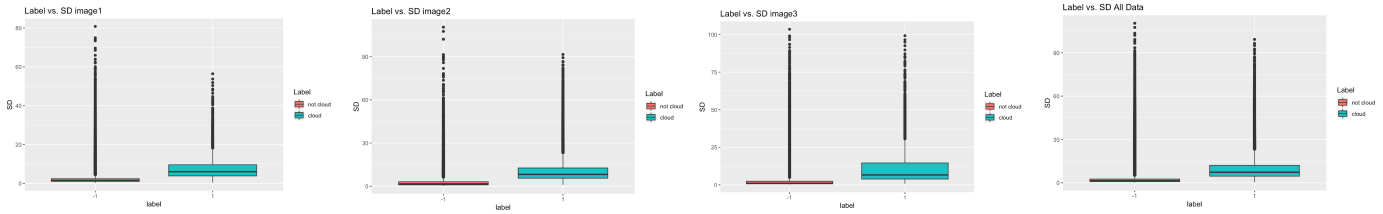


The boxplots for CORR based on different labels show that CORR is on average higher for pixels marked as cloud than as not cloud. CORR also has a higher variance for those marked as cloud.

This conclusion contradicts to the statement in the paper that high correlations over cloud-free or low-cloud areas are expected. This is because high correlations also occur under rare circumstances due to cloud movement. More importantly, recklessly declaring clear for high CORR pixels and cloudy for low CORR pixels will produce errors because of the smoothness of surface terrain and the difference of attitudes of clouds.

Therefore, we should also involve the feature SD to identify surfaces into our investigation. We first plot SD vs. Label independently for each image and all data.

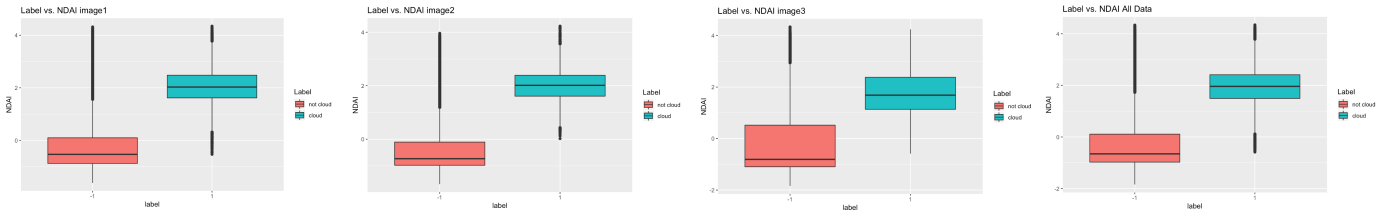
(relationship between expert label and SD)



From the boxplots for SD based on different labels, we can clearly see that the SD is higher for cloud pixels than those cloud-free ones. But the cloud-free pixels spread more widely. It can be explained that SD are usually small for radiation emanating from smooth surfaces.

Finally, the thrid feature NDAI relates to the differences for isoreopic level of surface-leaving radiation between low-attitude clouds and snow-coved surfaces.

(relationship between expert label and NDAI)



From the boxplots for NDAI based on different labels, we can clearly see that the NDAI is higher for cloud pixels than those cloud-free ones. The distrubition for cloud-free pixels are left skewed, while the distribution for pixels marked as cloud are roughly symmetric. The distribution for three different image data are roughly the same as well.

2 Preparation

a. Data Split

Even though the three image data sets represent the cloud dsitributions at different times at the same place, we decide to merge all data into one to have more data for training. We also clear out the unlabeled data here because they are not helpful for classification. Since the data are not i.i.d., we cannot simply split the data by random. To solve this problem, we come up with two different ways to split data into training, validation and test set.

First method: We divide the data into 25 groups by cutting the image horizontally. For example, data with y_coord from 2.0 to 78.2 are in the same group. We randomly select 23 groups from these 25 groups as training/validation data and 2 groups as test data. In the 23 groups, we randomly sampled 18 groups as validation data. Splitting data in this way looks like a one-stage cluster sampling. Every time we sample a group wihout replacement and include all data points in that group. Data in the same groups tend to be correlated to each other, which fits the property of the non-i.i.d. data.

Our Second method is, for each image data set, to divide all the data into 25 groups by cutting the image into 5*5 sub-images. For instance, data with y_coord from 2.0-78.2 and x_coord from 65.0-125.8 are in the same group. Same as the first method, we randomly select 23 groups from these 25 groups as training/validation data and 2 groups as test data. In the 23 groups, we randomly sampled 18 groups as validation data. Splitting the data in this way looks like stratified sampling. Every time we sample a group wihout replacement and include all data points in that group. We can also see this method as combining small pixels into a much larger one in a visual way. This method can guarantee pixels that have high correlations (adjacent pixels) can be sampled at the same time (except those on the boundary). And the random sampling on the 25 groups can make sure that we are not too sticking to the patterns for the training data so that overfitting can be somewhat prevented.

b. Baseline

Setting all labels to -1, the accuracy for test data is 0.520 and that for validation data is 0.270, both of

which are very low. Only if pixels marked as -1 have high frequency in the validation and test set, the trivial classifier will have a high average accuracy.

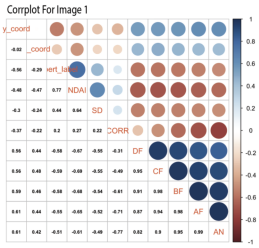
c. First Order Importance

Recall in the EDA part, we notice that five raw features(DF, CF, BF, AF, AN) have high linear relationship between each other. To have more information represented by features, we will choose at most one from these five features.

To find three best features, we first caculate and plot the correlation between each feature and expert label. The magnitudes of the absolute values of the correlations represent how close the relationships are between features and expert labels. Larger correlation value means a better feature. We find that NDAI and CORR are better than others on average.

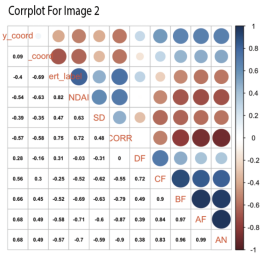
Correlation for Image 1

	[,1]
NDAI	0.7697836
SD	0.4423335
CORR	0.1965444
DF	-0.5845621
CF	-0.5856738
BF	-0.5761878
AF	-0.5485383
AN	-0.5089311



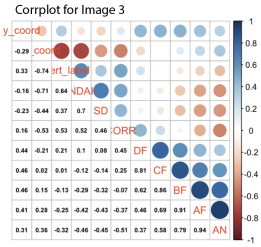
Correlation for Image 2

	[,1]
NDAI	0.8193795
SD	0.4736005
CORR	0.7500088
DF	0.3090899
CF	-0.2534827
BF	-0.5154060
AF	-0.5842979
AN	-0.5726166

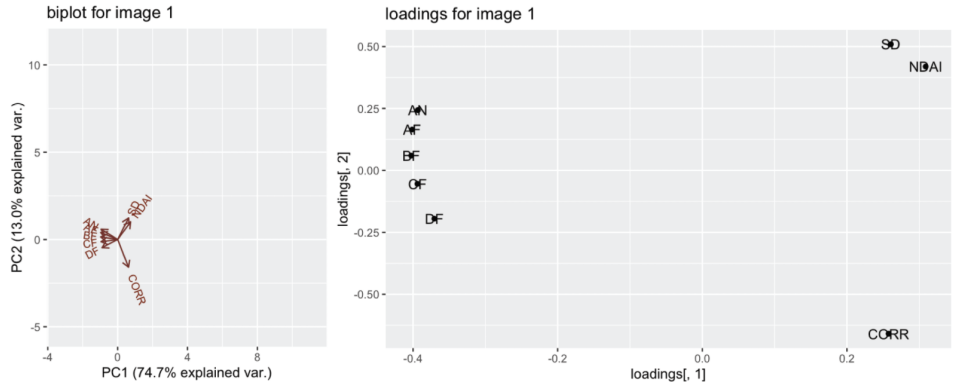


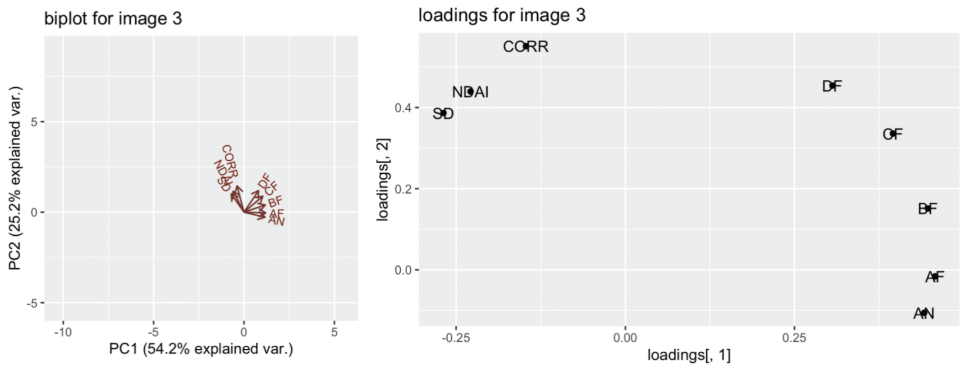
Correlation for Image 3

	[,1]
NDAI	0.49887924
SD	0.23597170
CORR	0.34274487
DF	0.14241384
CF	0.02168176
BF	-0.05722629
AF	-0.12841936
AN	-0.17290429



Then, we plot biplots and loadings on first two PCs to see how much each feature contributes to these PCs. We choose to use two PCs because they capture almost 90% of the data which is enough to represent the entire data set. The loading plots can illustrate the association between features and PCs. The larger the loading of a feature is in a given PC, the more association the feature contains with that PC. After viewing the four loading plots, we observe that SD and DF have larger loadings than other features (expect NDAI and CORR). Considering there are other reasons that is more associated with scientific explanation, we decide to be consistent with the three features chosen in the paper, which are CORR, NDAI and SD.





d. CVgeneric Function

See `https://github.com/WinnieGao/Project2_Git` for more detailed explanation and code.

3 Modeling

a. Different Classification Models

To train the data, we try several different classification methods: Generalized Linear Model, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-nearest Neighbor and Support Vector Machine. For GLM, LDA and QDA, since, in our natural world, most phenomena can be approximated using Gaussion distribution, we assume our data sets are under Gaussian distribution, which means LDA and QDA are feasible. Since Gaussian distribution is exponential family, we can also apply GLM in our case. And there is no specific assumptions for using KNN and SVM.

We use cross-validation with 8 folds to train the models, and get accuracies across all folds and average them. We also apply the model on the test set and get an accuracy on the test set.

For our first splitting method, the cross-validataion averaged accuracies show that LDA, QDA,SVM and KNN all return great performance with an accuracy above 0.91. Among them, LDA performs the best on the validataion data with 0.9286555 accauray and GLM gives the lowest average cross-validation accuarcy. As for the performance on the test set, all models gives lower accurary about 0.7 to 0.8. KNN gives the best accuracy and GLM returns the worst performance on the test set. Considering both the accuracies on the validation and test set, QDA performs relatively better than other models. Hence, we conclude that QDA is the best model for the first spliting data.

Validatation Accuracy on 1st spliting method

	glm <dbl>	lda <dbl>	qda <dbl>	knn <dbl>	svm <dbl>
1	0.8398113	0.9140388	0.9173759	0.8998891	0.9092465
2	0.8555957	0.9135048	0.9173330	0.9058482	0.9132592
3	0.8168549	0.9169997	0.9168187	0.8971260	0.9076886
4	0.8085911	0.9242555	0.9199618	0.9032978	0.9173894
5	0.8137595	0.9310160	0.9214933	0.9083902	0.9286789
6	0.8273810	0.9337363	0.9261355	0.9186813	0.9319231
7	0.8376645	0.9315191	0.9297815	0.9307014	0.9300626
8	0.8966396	0.9641736	0.9779027	0.9607087	0.9568515
average	0.8370372	0.9286555	0.9283503	0.9155803	0.9243875

Test Accuray on 1st spliting method

glm <dbl>	lda <dbl>	qda <dbl>	knn <dbl>	svm <dbl>
0.7036969	0.7767846	0.7899793	0.805667	0.7769062

For the second one, the cross-validation average accuracies demonstrate that LDA have the best performance on validation data. QDA also has a high accuracy on test data, but LDA performs better. GLM has lowest average cross-validation accuracy on validation set and svm performs worst on test data. Considering both average validation set accuracies and test set accuracies, we believe that QDA is also the best model for the second splitting method.

Validation Accuarcy on 2nd splitting method

	glm <dbl>	lda <dbl>	qda <dbl>	knn <dbl>	svm <dbl>
1	0.7767825	0.8990213	0.9047321	0.8906669	0.9003490
2	0.8290906	0.9018136	0.9040138	0.9026772	0.9028212
3	0.8000246	0.8915289	0.8905115	0.8976084	0.8932436
4	0.7853162	0.8939197	0.8973964	0.9009526	0.8948832
5	0.8250315	0.9015435	0.9011351	0.9038857	0.9030449
6	0.8625074	0.9406148	0.9410797	0.9480536	0.9450137
7	0.8657387	0.9409888	0.9516311	0.9452806	0.9414238
8	0.9226541	0.9798456	0.9945033	0.9807617	0.9790603
average	0.8333932	0.9186595	0.9231254	0.9212358	0.9199800

Test Accuarcy on 2nd splitting method

glm <dbl>	lda <dbl>	qda <dbl>	knn <dbl>	svm <dbl>
0.7882208	0.9052102	0.9154809	0.9120039	0.908403

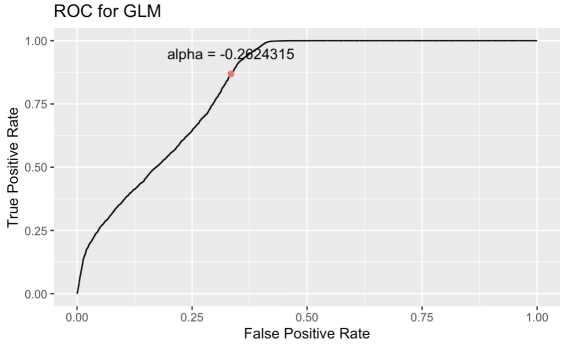
b. ROC curves

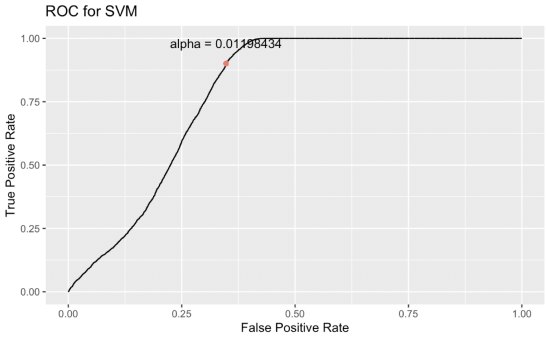
We then plot ROC curves for each classification method. We don't plot a ROC curve for KNN, because the model does not calculate the probabily of being each class. It directly gives the classification result instead, and we don't really think it is worthy changing the hyperparameter K here.

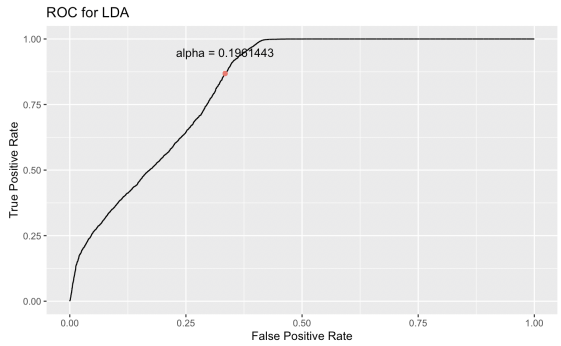
In each model, we choose the cutoff value that has the smallest distance to (0, 1). We do so because we want to maximize true positive rate and minimize false positive rate at the same time. We did not put more weight on true positive rate because, unlike the medical cases, the incorrect classifications of both clouded or cloud-free pixels have same consequence on our result.

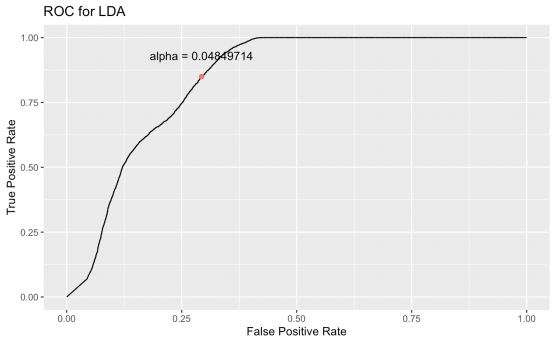
To choose a best method according to the ROC curves, we calculate the distance between the point we marked and (0, 1). The smaller distance indicates a better model.

The ROC curves are drawn below with cutoff value we choose marked for first splitting method:



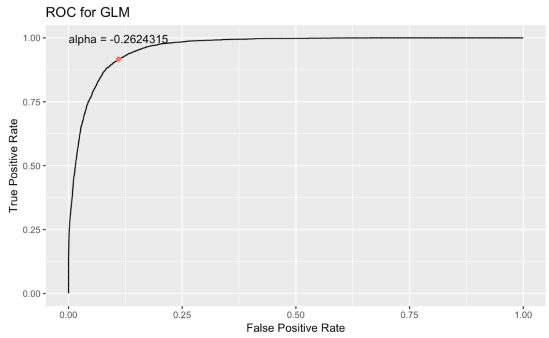


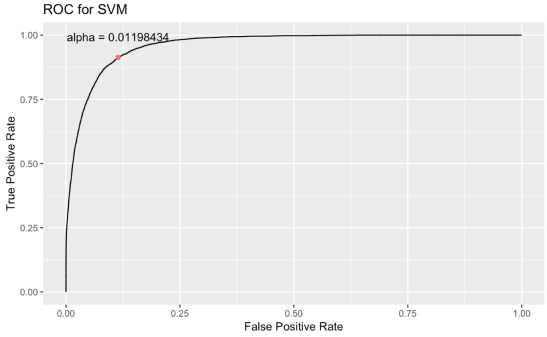


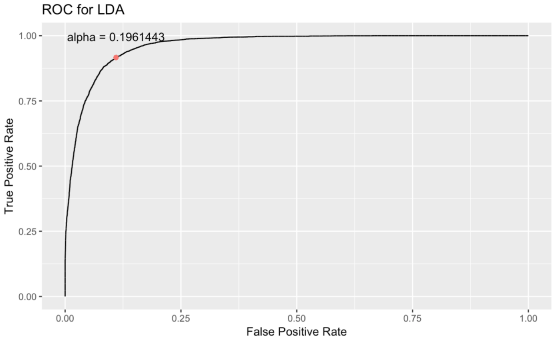


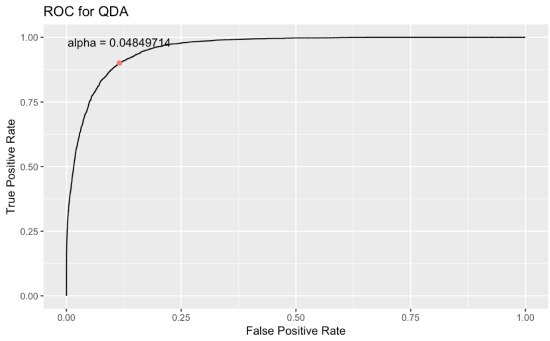
In the first spliting method, GLM and LDA both have the shortest distance 0.1291 to the point (0,1) and the same value 0.8239 for the area under the ROC curve value.SVM returns a slightly larger distance 0.1296 but covers the least area under the ROC curve with value 0.7854. QDA gives the largest distance 0.1637 to the point (0,1) and the largest AUC value 0.8391. Noticed that the ROC curves for four models are not smooth and some curves are slightly convex, which are different from usual ROC curve. This is because the cloud data set cannot be separated with a linear combination of featues. All our models assume that the data are linear separable, which might result in the abnormal shape of the RUC cruve.

Similarly, we analyze the the ROC curves for second spliting method:









In the second spliting method, GLM has a distance of 0.019, SVM has a distance of 0.021, LDA has a distance of 0.019, QDA has a distance of 0.023. The distances of different methods does not change very much. Then, we also calculate the AUC for each classification method. The larger AUC indicate a better model. The area under GLM's ROC curve is 0.9635, the area under SVM's ROC curve is 0.9613, the area under LDA's ROC curve is 0.9635 and the area under QDA's ROC curve is 0.9581. Considering both of these two factors, for the second splitting method, we think GLM and LDA are better than the rest of two.

c. (Bonus) Assessing with New Metric

Having a high accuracy is not enough to evaluate a particular model. Suppose we have a trivial classifier, which is predicting all data points as 1, we may still get a very high accuracy since most data points are marked as 1 in our training and test data set, but it by no means proves this model is good.

Therefore, we come up with two new metrics that can avoid this kind of coincidence: Among all pixels labeled as cloud, what is the proportion of the pixels that have true label as cloud. Simiarly, Among all pixels with true label as cloud, what is the proportion of pixels we predicted as cloud. The first is called "precision"

and the second one is called "recall". Since they defined very similarly and have similar effect, in this report, we only use precision as a new metric to evaluate our model to save time.

The precision for our models are as follows:

fold	GLM	LDA	QDA	KNN	SVM
1	0.8502976	0.8146243	0.8357528	0.8396218	0.8012071
2	0.8641889	0.8158233	0.8270786	0.8251075	0.8120095
3	0.8381238	0.8138386	0.8292501	0.8275254	0.812126
4	0.8606599	0.8357375	0.8475883	0.8376058	0.8313803
5	0.8350298	0.800172	0.8124216	0.8072105	0.7949866
6	0.9209357	0.9205664	0.918663	0.9233081	0.919797
7	0.9989035	0.9131391	0.9247144	0.8924905	0.8962012
8	1	0.8898386	0.9170984	0.8541901	0.8671587
Average	0.8960174	0.8504675	0.8640709	0.8541979	0.8418583
Test Precision	0.9667242	0.9375206	0.9380211	0.936958	0.9361003

We can see that test precisions of our models give different ranking from our test accuracies.

4 Diagnostics

a. In-depth Analysis

According to the analysis above, we think second splitting method is better and QDA is the best classification method in our case. Therefore, we do some in-depth analysis on QDA for the second splitting method.

The means and covariances of each class are shown as follows:

Mean and Covariance matrix for design matrix classified as cloud

```
     NDAI        SD       CORR
1.9844069 10.1232066 0.2670402
```

```
             NDAI          SD        CORR
NDAI  0.43347838   2.4371609 0.01911099
SD    2.43716086  70.3901334 0.20784936
CORR  0.01911099   0.2078494 0.01805218
```

Mean and Covariance matrix for design matrix classified as cloud-free

```
      NDAI        SD       CORR
-0.2722521 3.0813551 0.1405674
```

```
             NDAI          SD        CORR
NDAI  1.20711956   4.51222766 0.018354552
SD    4.51222766  38.10392684 0.098908453
CORR  0.01835455   0.09890845 0.002495529
```

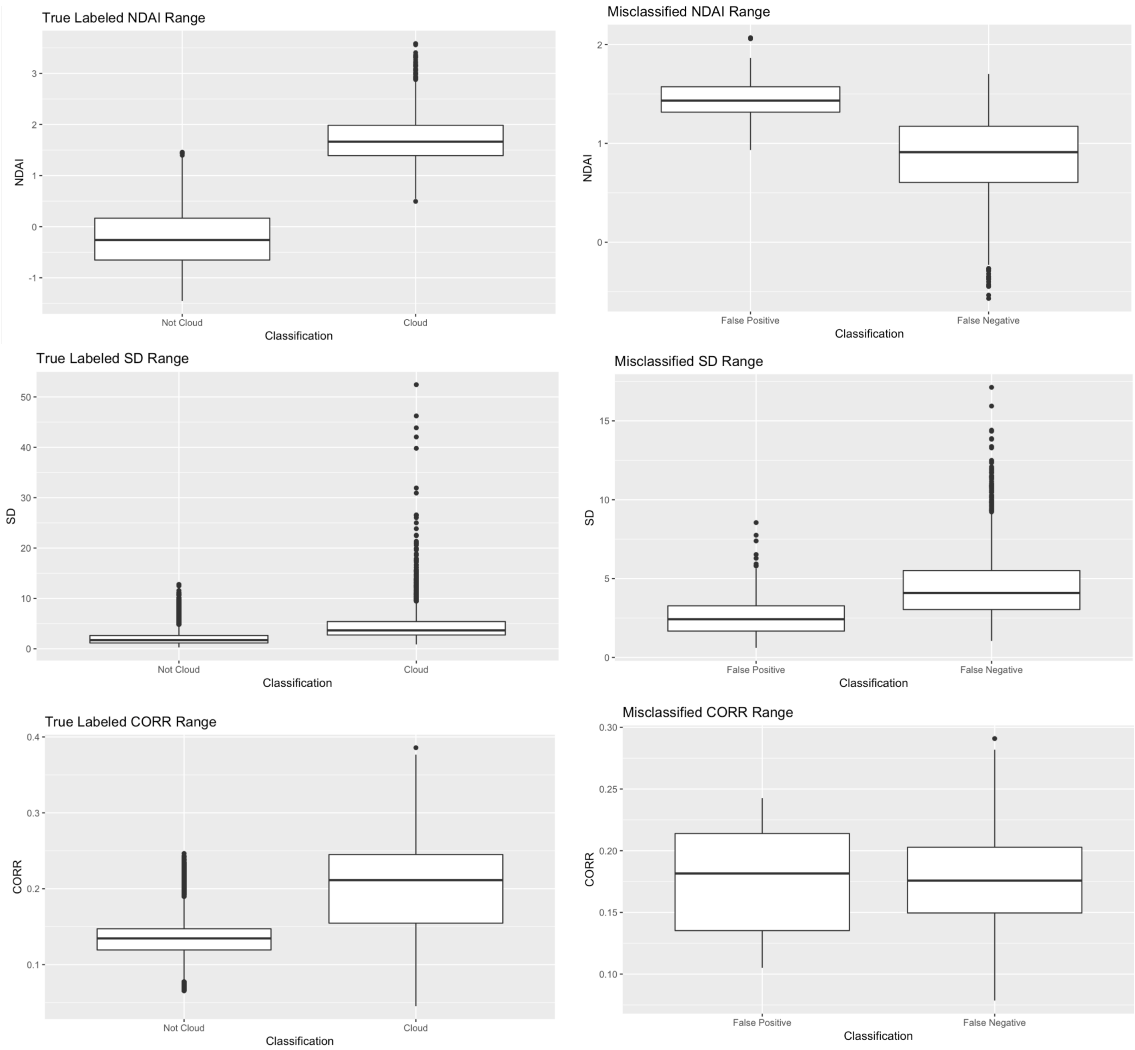
The high variances (covariances between features themselves) means that the corresponding features have a lot of expressiveness. In other words, features with low variances are close to a constant. In our case, we prefer features with larger variances. For the rest covariances (covariances between different features), features with high covariances have a lot of redundancy for each other. Since we hope the features can capture as much as possible, we prefer features with low covariances.

In our covariance matrices, SD has very high variance and relatively low covariance. SD is the best feature of these three. CORR has lowest variance and relatively high covariance. We may want to improve our model by changing this feature to another.

b. Misclassification Analysis

For our best classification model which is QDA, we analyze the misclassification by showing ranges of feature values. We also apply our trained model to three different image data sets and plot the misclassified data point on the original map to check if they are in particular regrions.

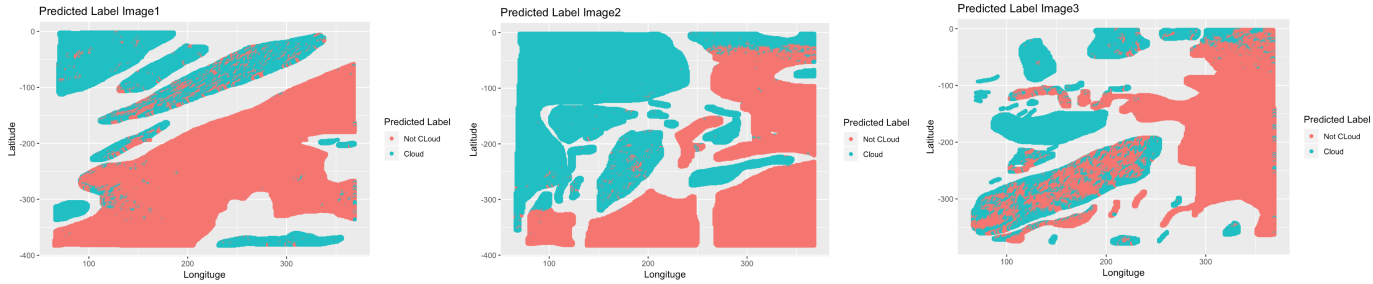
In the test data, we pick all misclassified data points and plot boxplots for each feature based on two different classification errors (False Positive and False Negative).



The plot on the left is the data correctly classified and the plot on the right are those misclassified. Comparing groups of distributions (Not Cloud vs. False Positive & Cloud vs. False Negative) for NDAI, we see they have a different range and are differently distributed, which means the mispredicted data have a partiular pattern on this feature. Thus, we can conclude that our model can easily predict data with cloud as not clouded when their NDAI have high values and more tend to predict data without cloud as clouded when their NDAI have extremely low value.

The data in plots of SD and CORR have quite similar distributions for both groups. So misclassification does not have particular patterns for these two features.

Then, we use our trained QDA model to predict data points in three image data sets (data points with expert label of 0 are excluded). And plot the predicted labels for each image data set.



Comparing these three scatterplots with those in Part1b, we notice that there are several inconsistencies in our plots. The inconsistencies imply the misclassified data points.

We notice that the data at the boundary of clouded and cloud-free pixels are often misclassified. More importantly, we realize that there are several small blocks that are entirely misclassified. For example, in the

first image, the small patch between the upper left two larger patches are classified as cloud while they should have been marked as cloud-free. This is probably because our model is not expressive enough to sperate out the rapid change. In image2, we have significantly fewer misclassified data points. This again verifies that our model is underfitting for image1. The same explanation can also be applied to the third image.

c. Better Classifier

Based on the analysis in the previous two sections, we come up with two ways to improve our classifier. First, as we mentioned above, the three features selected are not expressive enough to train our model. We decide to add a new feature to train our QDA. Recall the PCA figures we plotted in the second part. Among the rest of the five raw features, DF stands out as the best. Therefore, we add DF as a new feature and train our QDA classification model again on our training data. The final accuracy on test data arises from 0.9155 to 0.9251, which is better than the original QDA model. For future data without expert label, we believe our data can work well because our average accuracy over 8 folds cross validation and test accuracy are quite high. Moreover, QDA won't take that much time to train and predict as other classification methods like svm and knn.

Secondly, QDA requires that the data should be Gaussian distribution, but we cannot ensure that the future data are strictly meet this requirement. Therefore, we also tried Random Forest to train our data. Since we don't have a large number of features, Random Forest is suitable in our case.

```
                Type of random forest: classification
                      Number of trees: 25
No. of variables tried at each split: 3

        OOB estimate of  error rate: 9.09%
Confusion matrix:
       -1     1 class.error
-1 104735 10108  0.08801581
1    7102 67417  0.09530455
```

We choose to consider 3 features at each split and 25 trees for each forest. We train our model on train data and get a test accuracy of 0.9224 on test data, which perform better than QDA model with test accuary 0.9155. Moreover,random forest does not have particular assumption on the data set, the accuracy on future data should remain around the same. Hence, we confirm that random forest is a better classifier.

d. Changes after Modification of Splitting Method

The results in parts 4(a) and 4(b) change as we use splitting method 1.

Mean and Covariance matrix for design matrix classified as cloud using splitting method 1

```
     NDAI       SD       CORR
1.9292561 9.7734463 0.2610552
```

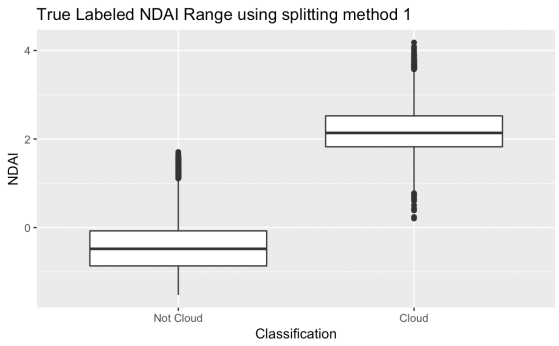
```
             NDAI         SD        CORR
NDAI  0.46201682  2.6965044 0.02185575
SD    2.69650438 70.5814816 0.25087456
CORR  0.02185575  0.2508746 0.01588994
```

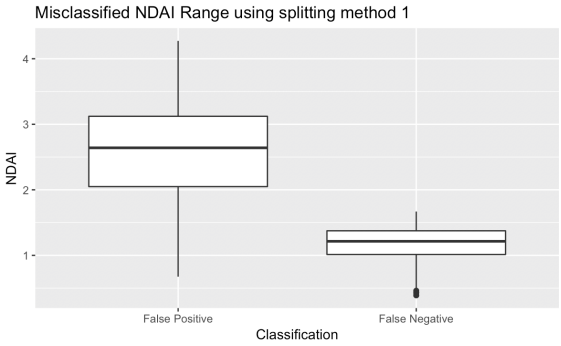
Mean and Covariance matrix for design matrix classified as cloud-free using splitting method 1

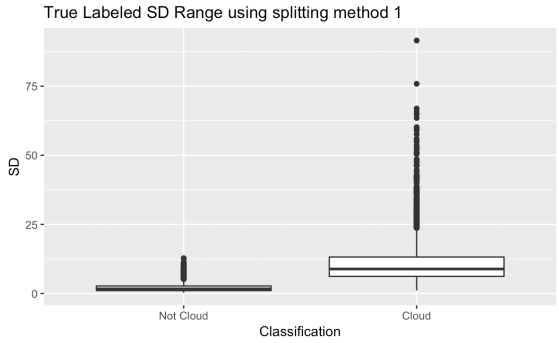
```
      NDAI       SD       CORR
-0.3283963  2.5588087  0.1388433
```

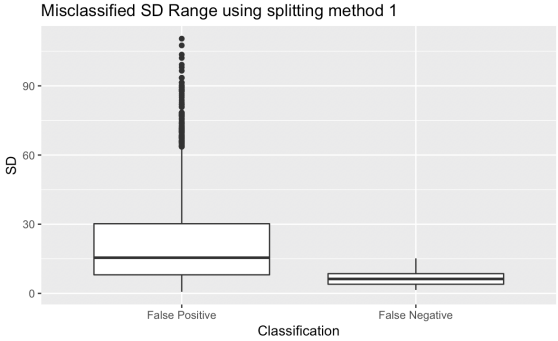
```
             NDAI         SD         CORR
NDAI  0.9643521  2.77374722 0.012458602
SD    2.7737472 20.26466969 0.051271107
CORR  0.0124586  0.05127111 0.002073285
```

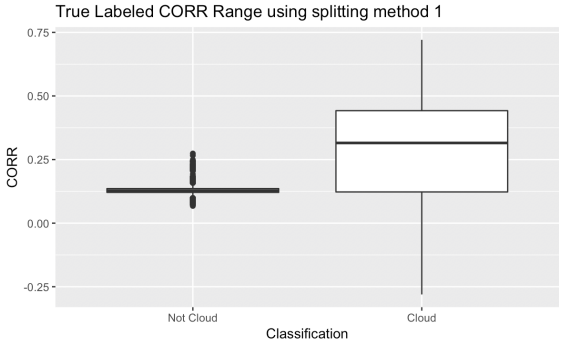
Applying the same feature preference rule we use in 4(b), a good feature would be the one with larger variances and low covariances. In the covariance matrix,

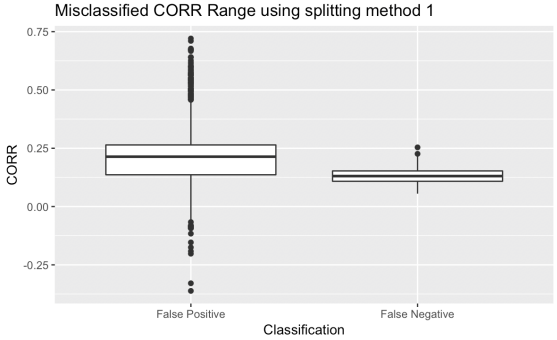












The box plots above show that