Project 2 Cloud Data

Dajie Sun (SID: 3032161913)
Jiahong Xia (SID: 3033398357)

1. Data Collection and Exploration

(a) **Summary of the Paper**

Purpose:

The purpose of the study is to develop algorithms that could classify image pixels with a cloud from pixels without the cloud. The data based on is the image data collected by the Multiangle Imaging SpectroRadiometer (MISR) onboard the National Aeronautics and Space Administration (NASA) Terra satellite launched in 1999. The MISR sensor comprises nine cameras, with each camera viewing Earth scenes at a different angle in four spectral bands (blue, green, red, and near-infrared). So for each pixel, the data is 36 dimensions. The data used in this study were collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and the Baffin Bay, span approximately 144 days from April 28 through September 19, 2002.

The author extracted three physically useful features out of the massive data set to differentiate surface pixels from cloudy pixels. The three features are: (1) the linear correlation of radiation measurements from different MISR view directions (CORR), (2) the standard deviation of MISR nadir red radiation measurements within a small region (SD), (3) a normalized difference angular index (NDAI).

Method:

Based on the three features mentioned above, the author developed 2 algorithms: (1) Enhanced Linear Correlation Matching Algorithm (ELCM), (2) Probability Prediction by Training QDA on ELCM (ELCM-QDA). The first algorithm can give a binary prediction result for each pixel, cloud or non-cloud. The second algorithm gives the probability of each possible classification.

Conclusion:

The three features contain sufficient information to separate clouds from ice- and snow-covered surfaces. And the ELCM algorithm based on the three features is more accurate and provides better spatial coverage than the existing MISR operational algorithms for cloud detection in the Arctic.

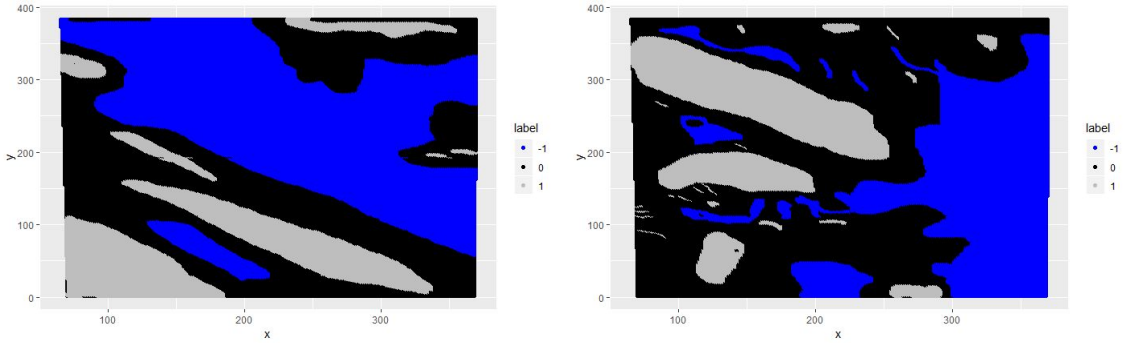
Potential Impact:

(1) In this work, statisticians are directly involved in the nuts and bolts of data processing.
(2) This research is that it demonstrates the power of statistical thinking, and also the ability of statistics to contribute solutions to modern scientific problems.

(b) Summary of the Data

Expert Label	Percentage (%)
-1	36.77552
0	39.78950
+1	23.43499

From the figure below, there are patterns in the region. There are blocks of cloud and no cloud in the map. **So, the i.i.d assumption for the samples is NOT good for this data set**.

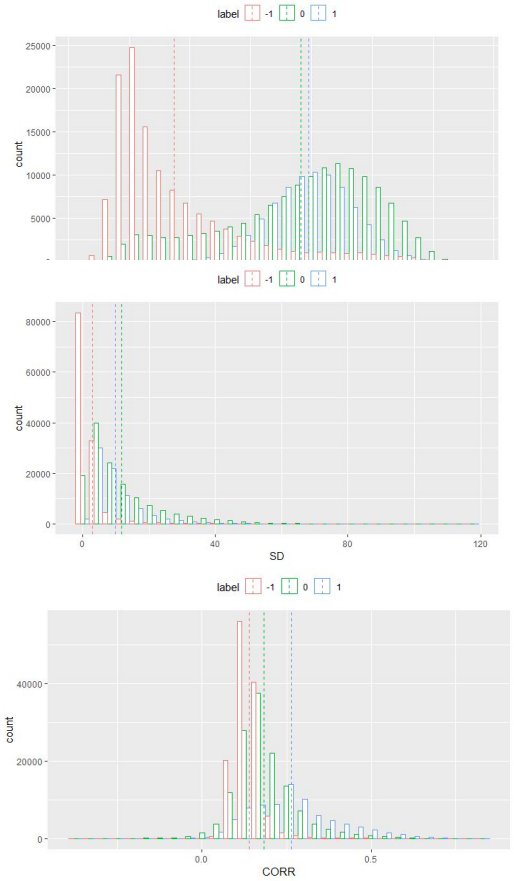


(c) Differences Between the Two Classes

The correlation matrix is below.

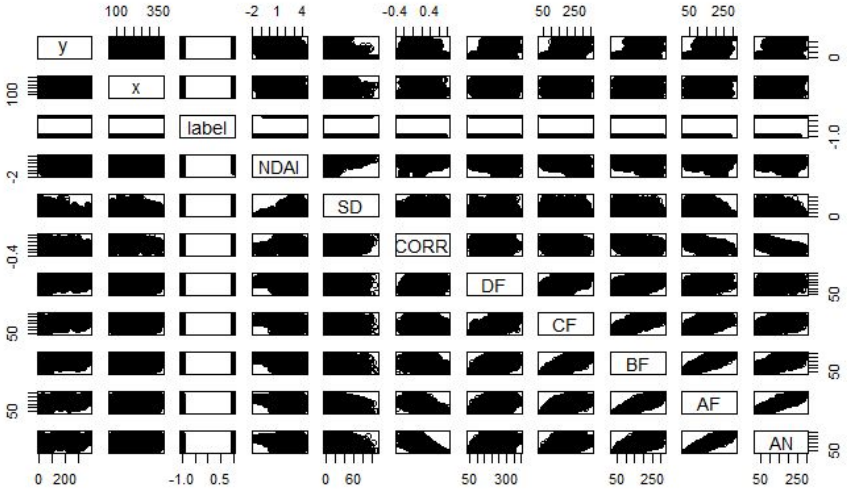
	y	x	label	NDAI	SD	CORR	DF	CF	BF	AF	AN
y	1	-0.04046	-0.28049	-0.42889	-0.31487	-0.35469	0.367421	0.503607	0.576624	0.599007	0.590364
x	-0.04046	1	-0.57119	-0.59859	-0.33724	-0.45665	-0.02904	0.222762	0.327012	0.376997	0.401983
label	-0.28049	-0.57119	1	0.75841	0.436026	0.551004	0.010787	-0.28276	-0.44766	-0.50732	-0.5046
NDAI	-0.42889	-0.59859	0.75841	1	0.647447	0.535021	-0.164	-0.43847	-0.57101	-0.61199	-0.60853
SD	-0.31487	-0.33724	0.436026	0.647447	1	0.407306	-0.19657	-0.40703	-0.49124	-0.51433	-0.50688
CORR	-0.35469	-0.45665	0.551004	0.535021	0.407306	1	0.147762	-0.22909	-0.51821	-0.68402	-0.74608
DF	0.367421	-0.02904	0.010787	-0.164	-0.19657	0.147762	1	0.850304	0.670345	0.537794	0.489264
CF	0.503607	0.222762	-0.28276	-0.43847	-0.40703	-0.22909	0.850304	1	0.918958	0.825947	0.77952
BF	0.576624	0.327012	-0.44766	-0.57101	-0.49124	-0.51821	0.670344	0.918958	1	0.962479	0.92556
AF	0.599007	0.376997	-0.50732	-0.61199	-0.51433	-0.68402	0.537794	0.825947	0.962479	1	0.981917
AN	0.590364	0.401983	-0.5046	-0.60853	-0.50688	-0.74608	0.489264	0.77952	0.92556	0.981917	1

Feature AF is correlated with CF, BF, AN.



We plot the distribution for NDAI, SD and with CORR on left hand side, we could see that, **cloudy region** are likely to have a **higher** NDAI, CORR and SD.

We could also see that the label is highly correlated with NDAI, CORR, and are negatively correlated with x coordinate, AN and AF in the plot below.



2. Preparation

(a)Split the entire data in two ways

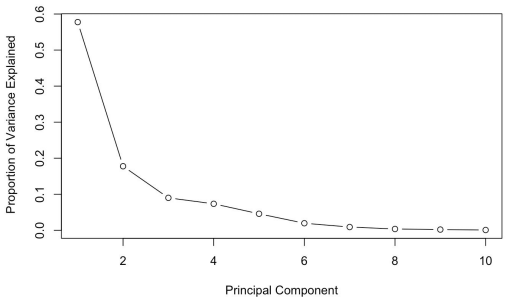
The first method I used is extract 60%, 20%, 20% of the data from image1, image2 and image3 and combine it into our training set, validation set, test set accordingly. We choose 20% test set since we want to make sure our testset size is large enough.

The second method I used is separate the data by label, one group for label equal to 1, and the other group is for label equal to -1. For each group, I also extract 80%, 10%, 10% of the data from image1, image2 and image3 and combine it into our training set, validation set, test set accordingly. Finally, I combined two groups' training set, validation set and test set together, and finalize the training set, validation set and test set.

(b)Report the accuracy of a trivial classifier

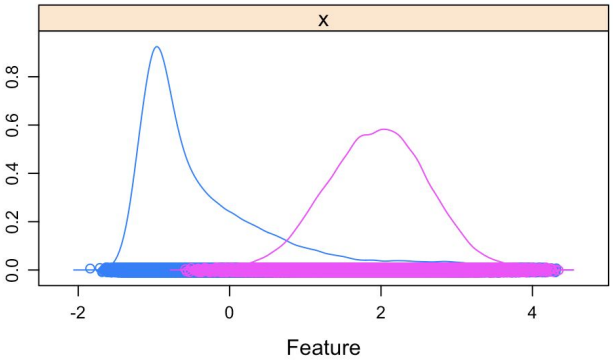
The accuracy of a trivial classifier which sets all labels to -1 (cloud-free) on the validation set is 0.6093382 and 0.6122125 on the test set. If there is a large portion of cloud-free area, this classifier have high average accuracy.

(c) Select Three of the "best" Features by Quantitative and Visual Justification

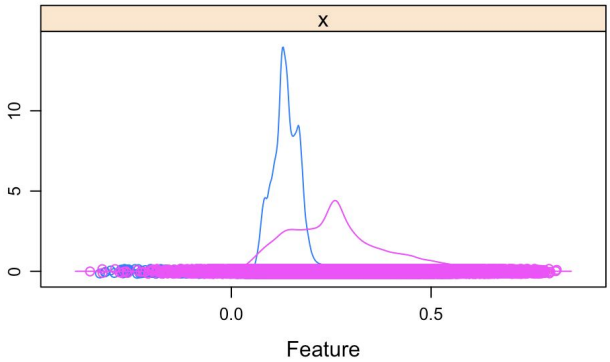


Quantitative Justification:
When we performed principal components analysis, we noticed that the curve is going down. There are 10 PCs, and since the first 3 PC can explain 85% of the variability, it's good enough so we'd better keep the first 3 PCs, which is NDAI, CORR and SD.

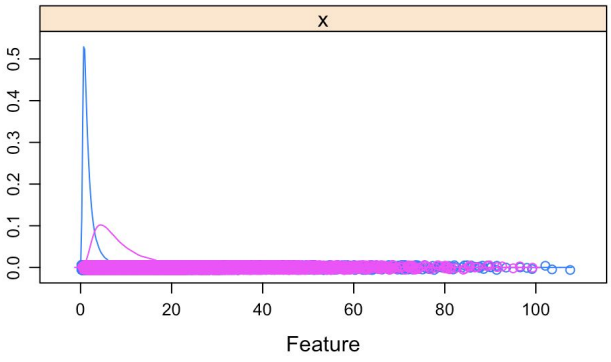
Visual Justification:



X = NDAI



X=CORR



X = SD

We then used featureplot for the first three components(NDAI, CORR and SD), there has a clear threshold between two distributions, which mean these components can better classify different labels.

(d)Generic Cross Validation (CV) Function

```
library(caret)
library(e1071)
# 2 (d)
CVgeneric <- function(data=TrainSet, formula= "label ~ NDAI + SD + CORR",method="glm",family="binomial",
cvfold=10, lossfunction="sensitivity", cutoff=0.5){
  currFormula <- as.formula(formula)
  folds <- createFolds(data$label, k = cvfold)
  conf_matrix=list()
  for (i in 1:cvfold){
    train_cv=TrainSet[-folds[[i]],]
    valid_cv=TrainSet[folds[[i]],]
    if (method == "svm"){
      trainsubIndex <- sample(seq_len(nrow(train_cv)), size = 10000) # actual training data for svm
      training_cv_sub <- train_cv[trainsubIndex, ]
      validsubIndex <- sample(seq_len(nrow(valid_cv)), size = 2000) # actual validation data for svm
      valid_cv_sub=valid_cv[validsubIndex,]
      mod_fit <- svm(currFormula, data = training_cv_sub, type = 'C-classification',kernel = 'linear',
probability = TRUE)
      prob <- predict(mod_fit, type="prob", newdata=valid_cv_sub, probability = TRUE)
      a=attr(prob, "probabilities")
      if (colnames(a)[1]=="-1"){
        b=a[,2]
      }else {
        b=a[,1]
      }
      p=factor((b>=cutoff)*1+(-1)*(b< cutoff))
      conf_matrix[[i]]=confusionMatrix(p,valid_cv_sub$label,positive = levels(valid_cv_sub$label)[2])
    }else{
      mod_fit <- train(currFormula,  data=train_cv, method=method, family=family)
      pp=predict(mod_fit, newdata=valid_cv, type="prob")
      p=factor((pp$`1`>=cutoff)*1+(-1)*(pp$`1`< cutoff))
      conf_matrix[[i]]=confusionMatrix(p,valid_cv$label,positive = levels(valid_cv$label)[2])
    }

  }
  return(conf_matrix)
}

cv_res <- CVgeneric(TrainSet, method = "svm")
cv_res[[1]]$table # get the confusion matrix
cv_res[[2]]$overall[['Accuracy']] # get the overall accuracy.
```

3 Modeling

(a)Accuracies Across Folds and the Test Accuracy Report

For split method 1 (cutoff=0.5)

Model/ Accuracy	Cross Validation										Test set
	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	
glm	0.89001	0.896027	0.894104	0.895378	0.889298	0.88985	0.892342	0.893864	0.891132	0.890812	0.890972
lda	0.896427	0.894505	0.900913	0.899063	0.899223	0.891532	0.895698	0.898029	0.897228	0.901626	0.896042
qda	0.897621	0.895466	0.897541	0.895546	0.899223	0.89682	0.900593	0.89738	0.895066	0.895466	0.895946
svm	0.8905	0.901	0.899	0.8905	0.8975	0.8835	0.893	0.9025	0.8865	0.8975	0.884

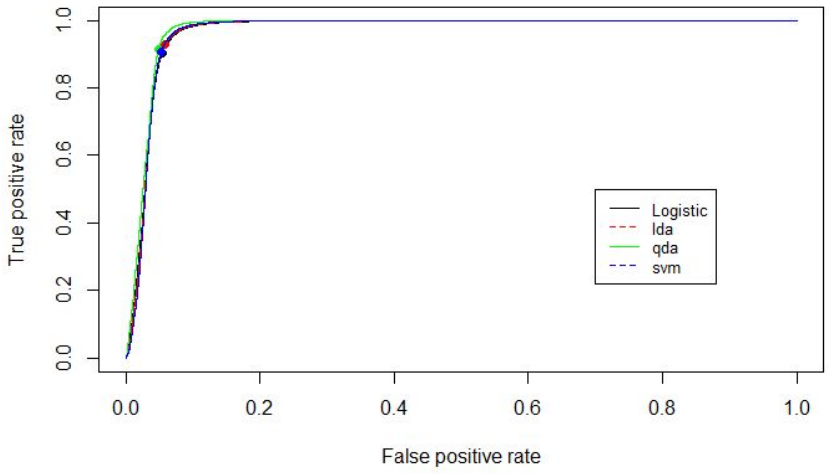
For split method 2 (cutoff=0.5)

Model/ Accuracy	Cross Validation										Test set
	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	
glm	0.890892	0.900585	0.896347	0.890731	0.890171	0.891461	0.890972	0.894577	0.892422	0.892743	0.893327
lda	0.896267	0.895146	0.900585	0.89859	0.897132	0.898262	0.899792	0.895298	0.898983	0.896988	0.898181
qda	0.897941	0.89706	0.895466	0.900913	0.898662	0.89827	0.898021	0.892182	0.898342	0.898983	0.898541
svm	0.892	0.889	0.8945	0.8955	0.898	0.899	0.8835	0.901	0.9035	0.8935	0.9085

It shows that data split method 2 is obviously better than method 1, that is method 2 reduce the variance due to random by keep the ratio of class 1 and class 2 equal after splitting.

Besides, qda model is the best of the four model since qda has the highest accuracy rate.

For svm model at this section, we found that svm is super time consuming for this dataset. It will at least take a few hours in our pc. In order to make it easier, we just sampled 10000 data points out of each fold to represent this fold and do the analysis on the 10000 points subnet, we think this method is reasonable.



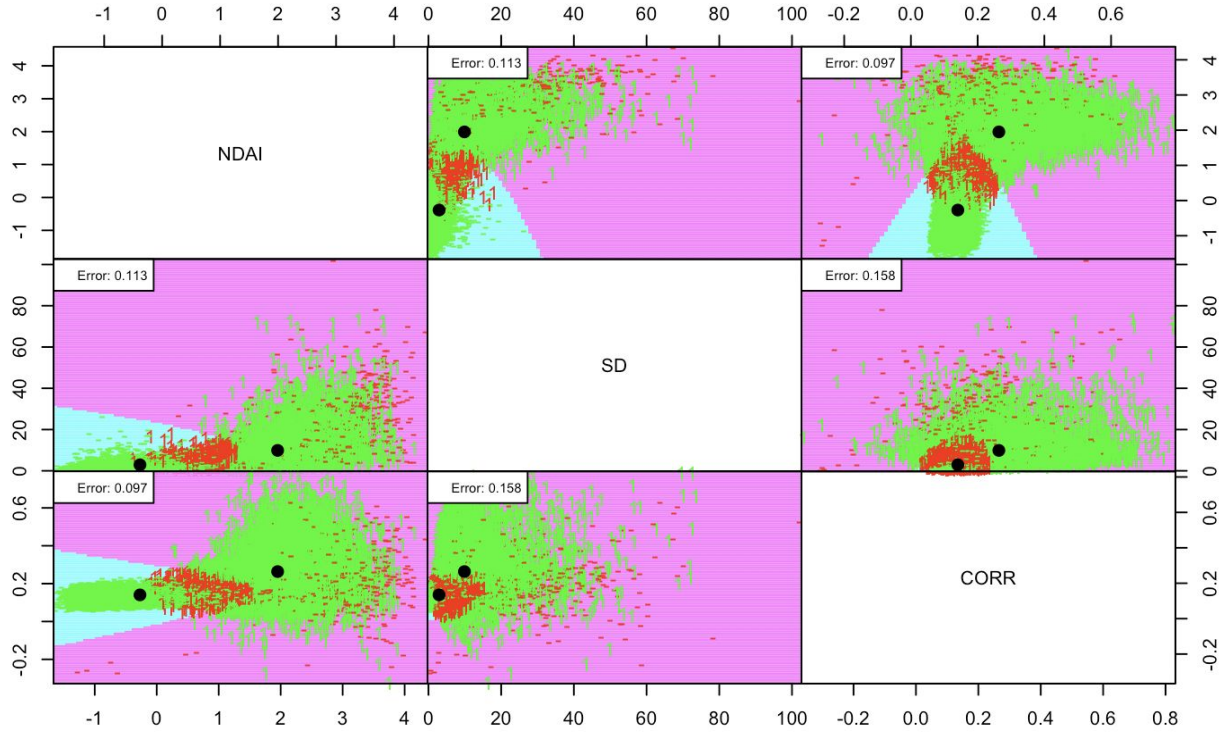
(b)

We could see from the plot above, the green line(qda) is the best, obviously, it has the biggest AUC. Since 0.5 is our default value, label == -1 & label ==1 has the same priority.

4 Diagnostics.

(a) In-Depth Analysis of a Good Classification

We used qda as our the classifier, so we use the function *partimat* in package "klaR", partimat provides a multiple figure array which shows the classification of observations based on classification methods for every combination of two variables. Moreover, the classification borders are displayed and the apparent error rates are given in each title.



qda

In the qda plot above, We could see that the error rate in SD~CORR map is relatively higher than other factors (and the means of the two class are very close to each other), that is because these two features are not independent, so, it will be kind of difficult to classify them using the two feathers. To reduce the features' collinear effect, we should use *Random Forest* Algorithm.

(b)Patterns for the Misclassification Errors

We tried to separate two situation, one is for label ==1 & predict ==-1, and the other is label ==-1 & predict ==1. However, there's no intuitive difference between this two situation. Therefore, we combine this two situation together.

We choose a subset of the training set with size equal to 20000 as our training set in order to increase our efficiency, and it makes sense.

Quantitative method:

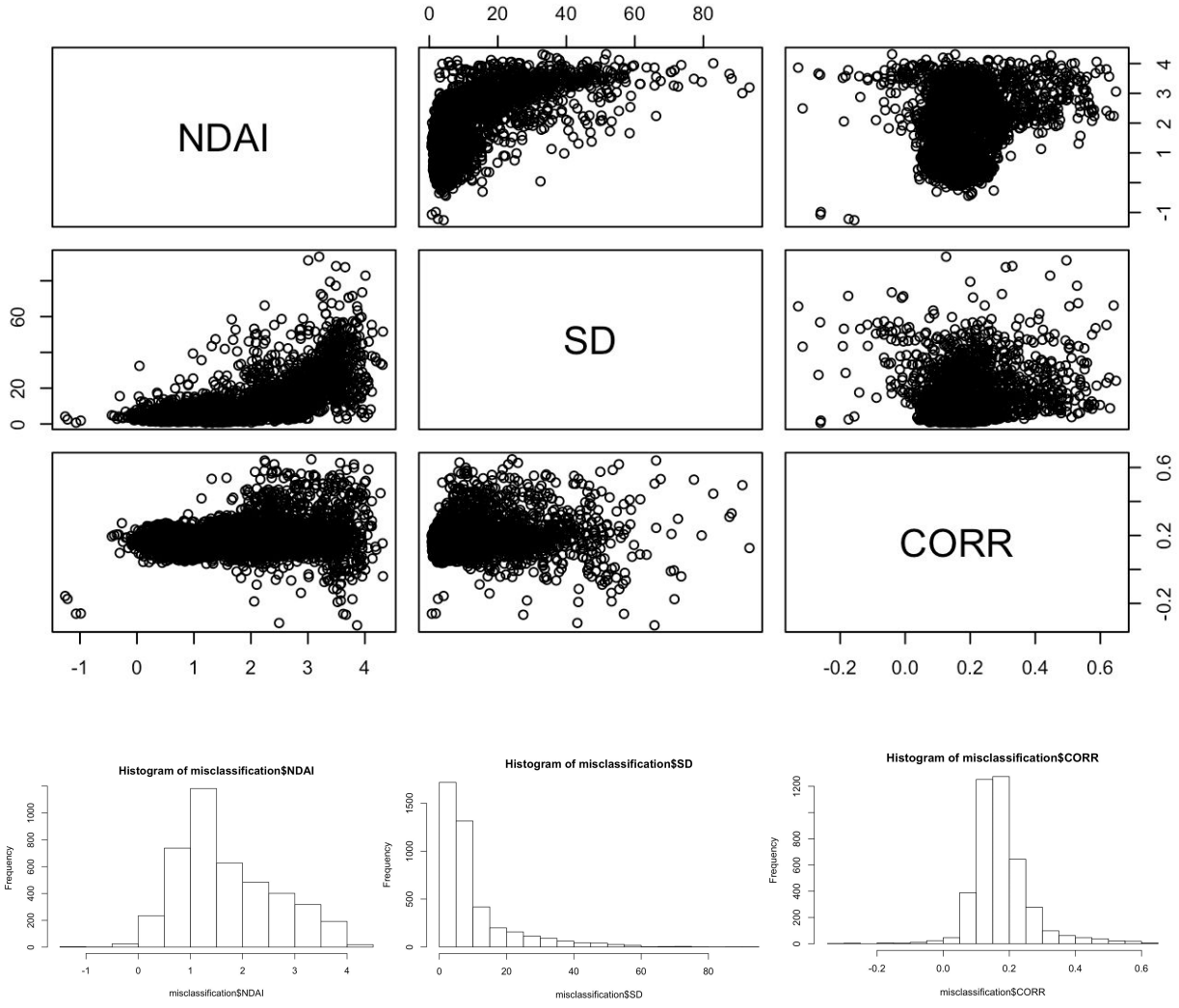
Number of misclassification: 4222

% of misclassification: 10.145%

Range of misclassification

NDAI: [-1.259984 4.315708] SD: [0.6916609 93.4643860] CORR: [-0.3290062 0.6481404]

visual methods

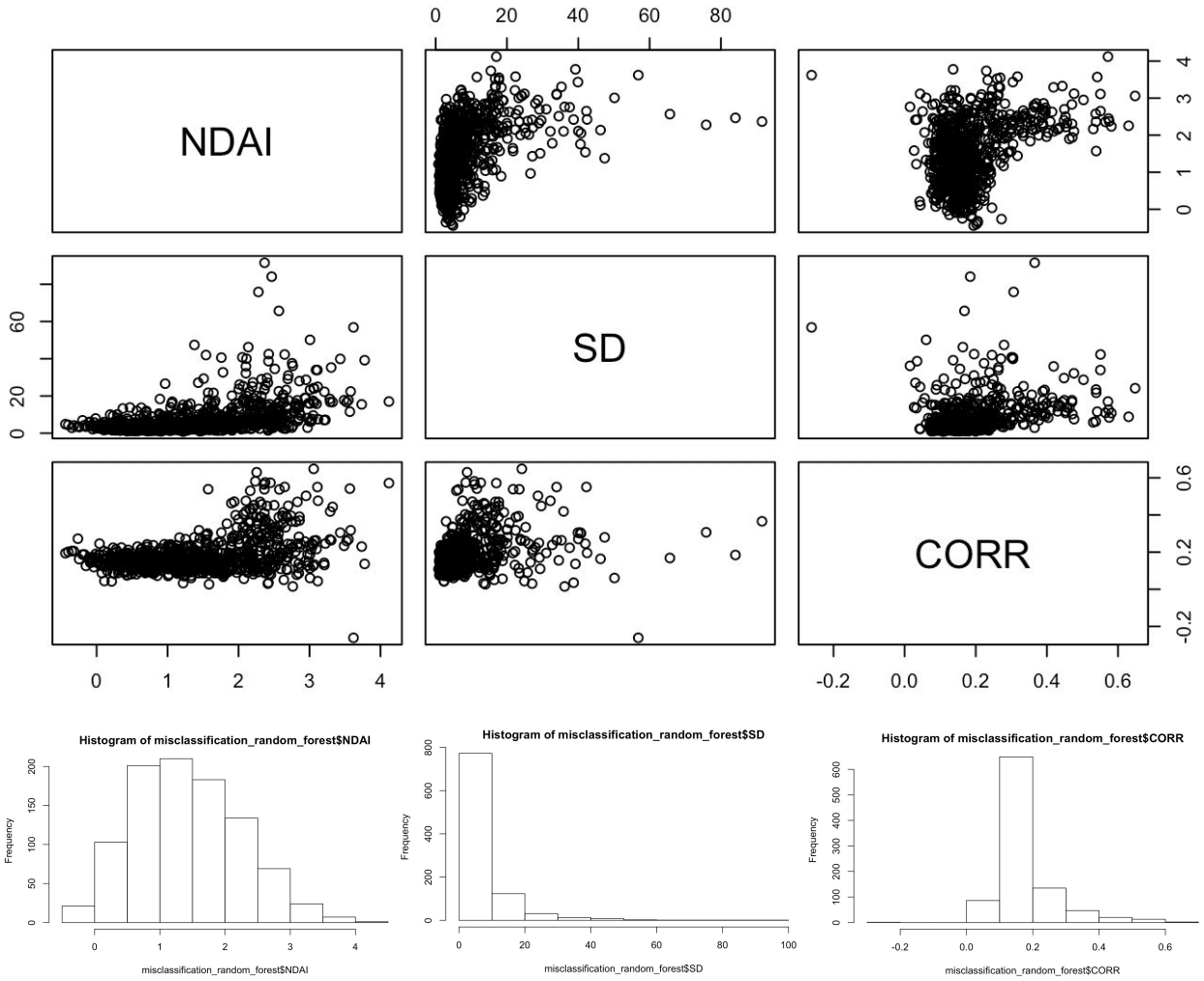


The majority misclassification overlap range are as below:

NDAI: [0, 4] SD: [0, 15] CORR: [0.05, 0.35].

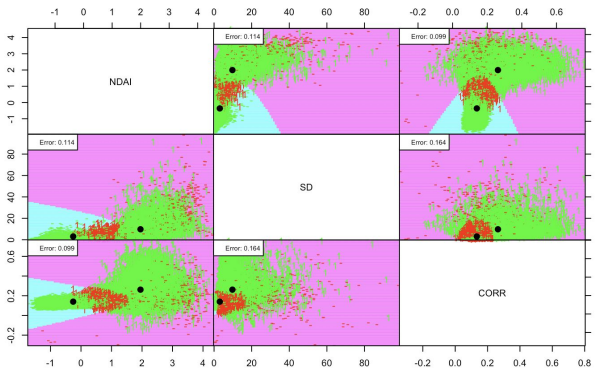
(c)Better Classifier

Based on parts 4(a) and 4(b), since the features are NOT independent, and these dependence will increase the variance, so we'd better using random forest classifier with all the features can do a better job. **When using qda, we have 4222 misclassification, however, using random forest classifier, we just have 547 misclassification, which improves 87%.**

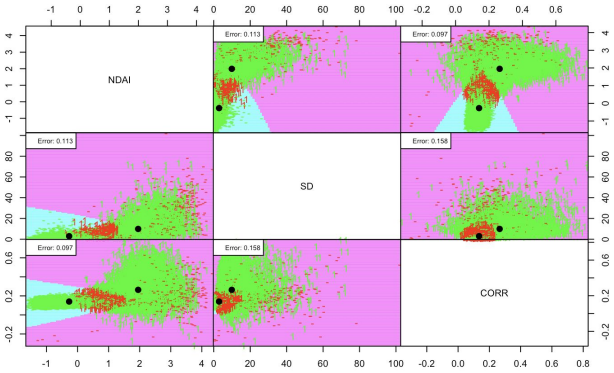


As you can see from the histplot, the frequency for each feature are reduced a huge amounts. Also, the majority misclassification overlap range is narrower than using the qda method.

(d)Modify Splitting Method



Method 1 for QDA



Method 2 for QDA

Method	Error(NADI & SD)	Error(NADI & CORR)	Error(CORR & SD)	Misclassification Rate
Method 1	0.114	0.099	0.164	10.41%
Method 2	0.113	0.097	0.158	10.15%

QDA Method

Method	Number of Misclassifications	Misclassification Rate
Method 1	566	1.36%
Method 2	547	1.31%

Random Forest

As the change of the way of splitting the data, no matter we use QDA of Random Forest, the second method performs better.

(e) Conclusion

Splitting method is very important, a good split method could reduce the variance due to spatial correlation, because these pixels are not i.i.d. We also found that the three features are not independent features, they are correlated, and to reduce the correlation effect, we could introduce *Random Forest* Algorithm, which could improve our predict accuracy significantly.

5. Reproducibility

https://github.com/JiahongXia/Cloud-Data

5. Acknowledgment

Jiahong Xia prepared the Data collection, exploration and preparation parts. Dajie Sun was responsible for modeling and diagnostics parts. We worked together especially for the diagnostics parts since this part is open-ended, we brainstorm together and finally resulting in a creative way. We used the resource from piazza (Thanks for Yuansi and Razz) and the method of evaluating logistic regression models(Please see Resources below).

Resources:

https://www.r-bloggers.com/evaluating-logistic-regression-models/
https://guides.github.com
https://www.textbook.ds100.org/ch/17/classification_log_reg.html
https://www.textbook.ds100.org/ch/17/classification_sensitivity_specificity.html