Project 2 Cloud Data Report: The Battle Between Classifiers and Who's The Winner

Evelyn Zou (#SID:3031983215)
Hanting Fei (#SID:3034489420)

1. Data Collection and Exploration

(a) Summary of the Paper:

This paper is a statistical study of the data features and distributions of clouds in the Arctic regions. It aims to analyze the characteristics of clouds to detect warming in the Arctic, which is particularly associated with the cloud coverage. Nonetheless, this research faces the major challenge of identifying clouds from other surfaces accurately, such as ice or snow-covered surfaces that look similar to the clouds. Therefore, the goal of this work is also to develop an operational cloud detection algorithm to process the MISR data without expert labels, the prior information of cloud and cloud-free surfaces.

The data is collected by the Multiangle Imaging SpectroRadiometer(MISR) that detects the red radiation primarily. During the daytime, MISR gathers the data from the 233 distinct path and nine different zenith angles on the Arctic, with each path divided into 180 blocks and repeated on a 16-day cycle. The number of blocks increases as observation moves from south pole to north pole. This study uses the data from 10 orbits of path 26 due to its abundance of the surface features, with each orbit contains six data units.

The analysis of the cloud data is based on two cloud detection algorithms, which have better accuracy and computational complexity than others. Both algorithms are based on three main physical features, including CORR(linear correlation of radiations), SD(standard deviation of radiation), and NDAI(normalized difference angular index), for these features contain the sufficient information for classification. First, the researchers build the ELCM algorithms by setting thresholds on each of the three features. They then apply the resulting ELCM labels to train the QDA and develop the ELCM-QDA algorithm. With the two algorithms, the researches predict the probability of cloudiness efficiently and accurately.

Hence, this study provides a valuable insight into the statistical application to the complex scientific questions, such as the climate studies, as the statistical methods distinguish the clouds from cloud-free regions effectively.

(b) Image 1 Class Table

```
Frequency of Each Class(Label)
   -1    0    1
50446 44312 20471
Percentage of Each Class(Label)
       -1         0         1
0.4377891 0.3845560 0.1776549
```

Image 2 Class Table

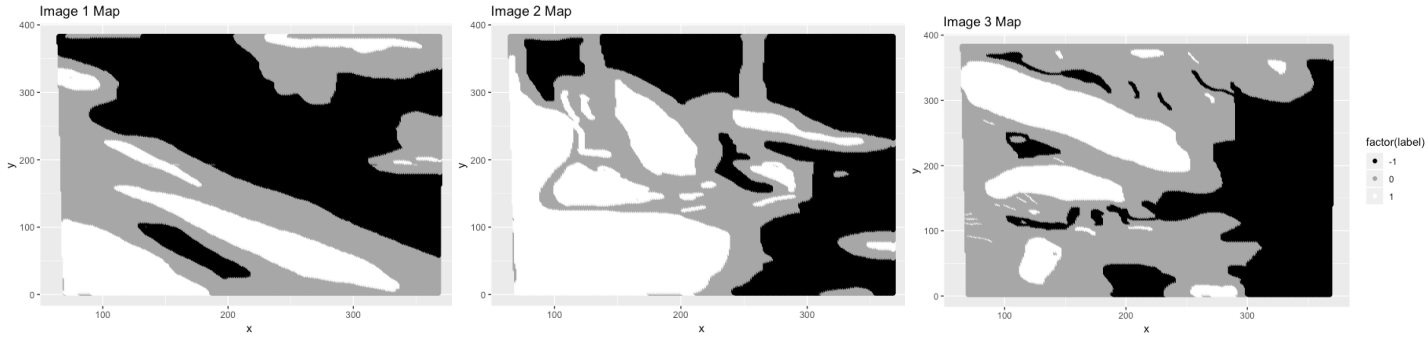
```
Frequency of Class(Label)
   -1    0    1
42882 32962 39266
Percentage of Class(Label)
       -1         0         1
0.3725306 0.2863522 0.3411172
```

Image 3 Class Table

```
Frequency of Class(Labels)
   -1    0    1
33752 60221 21244
Percentage of Class(Labels)
       -1         0         1
0.2929429 0.5226746 0.1843825
```

From the data summary of the three images, we see that all three variables are consistent in the variances and means for each of the features. Therefore, we can perform classification on the three images data.

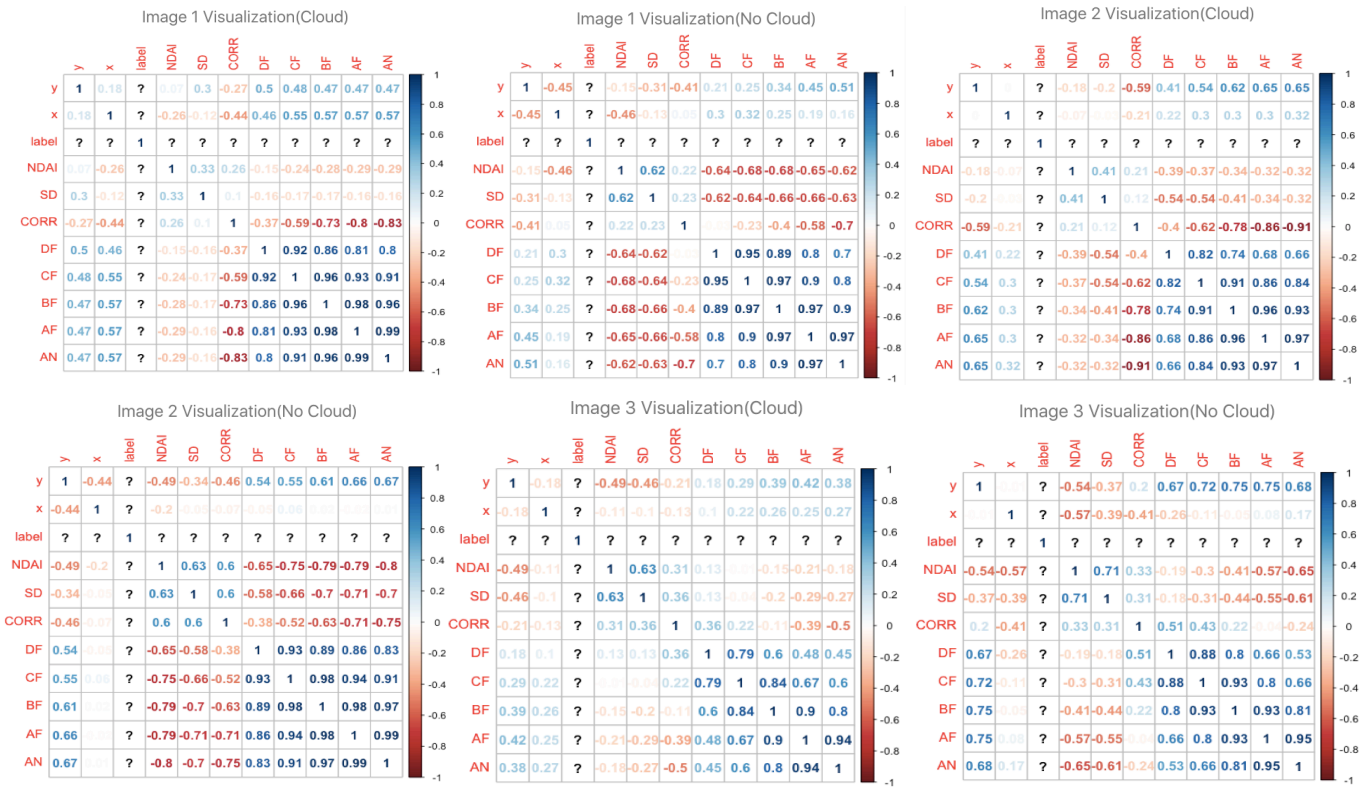
To distinguish clouds from cloud-free regions on the map, we use white color for the clouds, grey for the unlabeled regions, and black for cloud-free areas.



Based on the plots above, we observe that both the clouds and cloud-free regions are distributed randomly as clusters, surrounded by the unlabeled regions. This indicates the difficulty in identifying clouds accurately and no obvious pattern being observed across the maps. Thus, the dataset of **all three images** isn't identically and independently distributed, because the data of each image is collected by MISR from different orbits

with different x-y coordinates. However, we can assume that the clouds follow the i.i.d distribution **for each of the image, whereas the dataset of three images doesn't.**

(c)



For this part, we use the correlation plots to investigate the pairwise relationship between the features themselves, and between the expert labels with the individual features. From the plots of three images, we observe that each of the radiance angles is highly correlated to other radiance angles, whereas the rest of the features are only weakly correlated to each other.

However, for different expert labels(cloud for 1, cloud-zero for -1), the correlations between all radiance angles(AF, AN, BF…) and other features(y, x, NDAI, SD, CORR …) change in the intensity. For example, for image 1, the radiance angle features are strongly correlated to the SD in the cloud-free regions; on the contrary, this correlation gets much weaker in the cloudy regions. This difference can be explained: according to the paper, SD is the standard deviation of radiation measures, and thus the indicator of the surface smoothness. Smoother the surface, less the SD index, and more the radiation reflected and received by MISR. Therefore, SD and the amount of radiation received by MISR are negatively correlated. Since cloud-free regions are smoother than the clouds, this negative correlation grows stronger, as shown in the case of image1.

The change in correlation between radiation and other features can also be explained in a similar manner, especially noticeable for the three features: **NDAI, SD, and CORR.** Hence, we conclude that **the features are also correlated to the expert labels, evident in the change in the strength of correlation.**

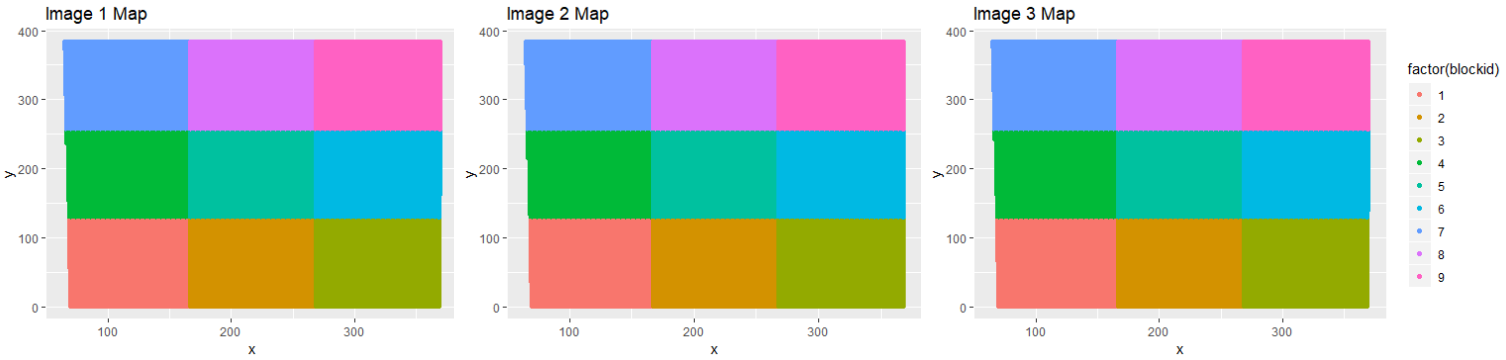
2. Preparation

(a) Splitting Method 1: From 1(b), the data of three image data aren't identically and independently distributed to each other, but the clouds follow i.i.d in individual image. As a result, we randomly split each image to half and combine the three halves from each image to form training set. Next, we repeat this operation and splitting the rest of data into halves to obtain validation and testing set.

Since the unlabeled regions(with expert labels 0) don't help the classification process much, to obtain the binary classification, we remove all the rows with 0 expert labels before sampling and make sure the probabilities of being chosen for every pixel are same. Then, we transform the {-1, 1} label back into {0,1} by replacing all the 0 labels with -1 labels to simplify the classification process.

Splitting Method 2: Because each image consists of three main blocks, we make a k*k grid for each image(k=3 in this case), dividing each image into 9 equal squares. Next, we reconstruct the three datasets through random sampling. We randomly select four squares from each image, and combine the total twelves squares as the training set; likewise, we sample another three squares from each image, and combine the total nine squares as the validation set. The rest of the squares are combined together and used as the testing set.

Different from the first method, we remove all the 0 labels at the end of this process, for it will become hard to visualize the partition if we remove them in the beginning. The value of k (k*k as the dimension of the squares) can vary in different classification methods.



(b) For this part, we design a trivial classifier reporting the label with the highest frequency, and the accuracy of data classification. Given that there're much more -1 labels than the 1 labels(more cloud-free areas than clouds) in the cloud data, the classifier will automatically set and display all labels to -1 with the corresponding accuracies for the test and validation set(using splitting method 1), shown as below:

```
Test Set Classifier Result
   -1
52015 Frequency

$accuracy
[1] 0.6120158 Accuracy
```

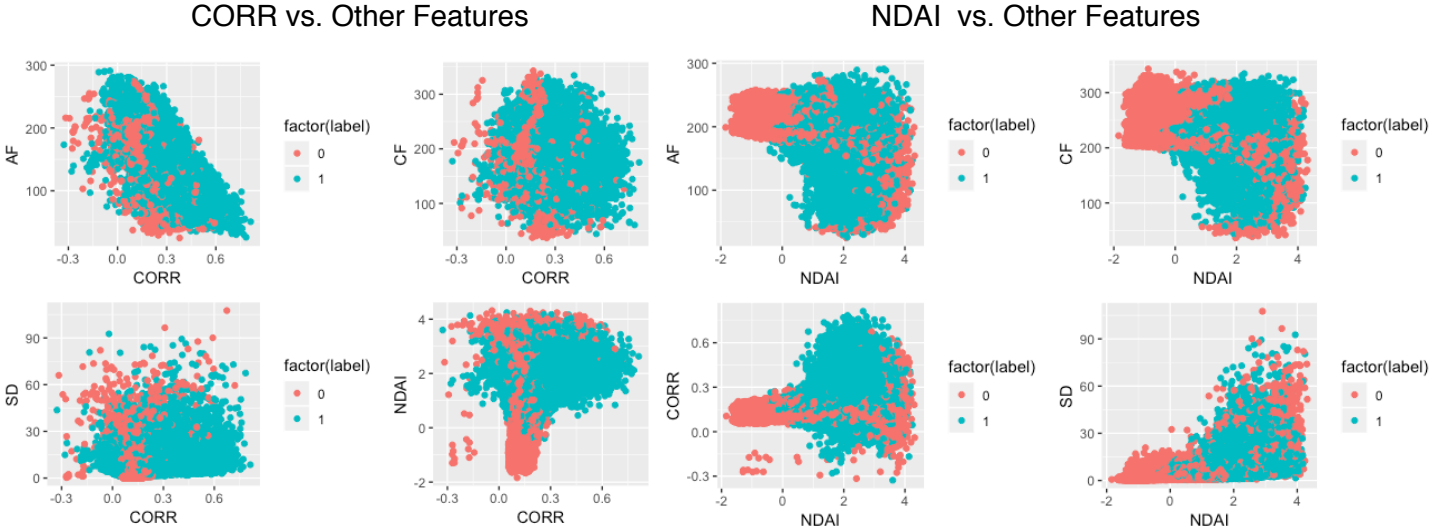
```
Validation Set Classifier Result
   -1
52015 Frequency

$accuracy
[1] 0.6117466 Accuracy
```

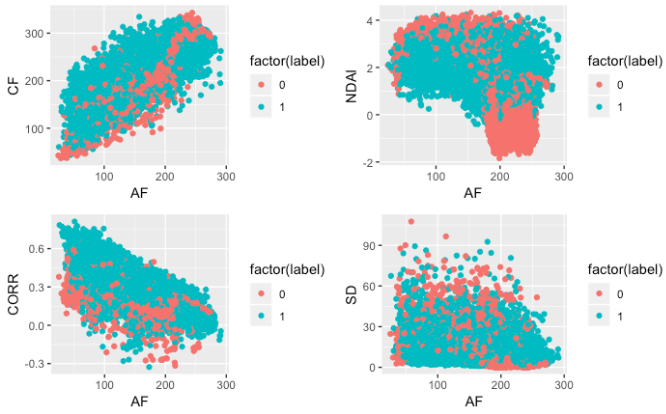
Since this trivial classifier only displays the majority of the labels, it has high accuracy when a particular label is much more than the other one in the data set.

(c) To find the best features, we first scatterplot different features and color them by the two labels below, to find any pattern useful for classification. From the 1(c), because the radiation measurements in different angles(AF, AN, CF …) behave consistent(with low standard deviation, in the same order of magnitude), we only need to use the radiation feature in a single angle for the scatterplot between radiation and other features.

Based on the plots below, we find a vertical trend in the two labels between CORR and other features, which separates the two labels remarkably. We thus conclude that CORR is one of the best features. The same observation holds for SD and NDAI, as they show similar patterns in labels in the scatterplot with each of the other features respectively. Even though the patterns may not be obvious for some features, they still prove the three features(CORR, SD, and NDAI) as the best features.



Radiation Features vs. Other Features

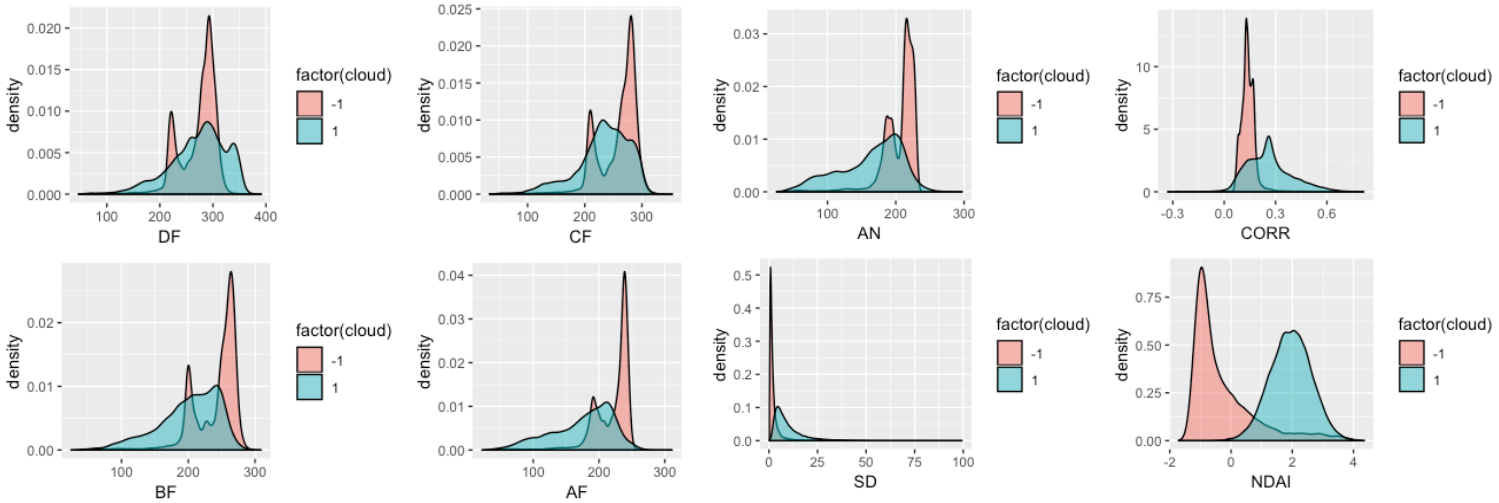


In contrast, the scatterplot between the radiation feature and other features show distinctive patterns in the labels for each of the variables, so the radiations(AF, CF …) are not the desirable features for classification due to the inconsistent patterns.

To further examine the conclusions above, we look at the density plots between different features. It can be seen that all radiation features have similar means and for different labels, as the density distributions under two different labels overlap each other much. On the contrary, NDAI, CORR and SD show disparate distributions for different labels, and each density distribution rarely overlaps with the other. So even though we can consider radiation features as "good features" owing to the different distributions corresponding to different labels, the other three features will still be the best for classification.

Radiation Features

The Three Best Features



(d) Because the CVgeneric function requires different arguments for different classifiers, we use the character string as the inputs of the classifier. We apply a variety of loss functions to fit the classifier best as the output, besides the accuracy. For instance, we use the logarithmic loss for the logistic regression/LDA classifier, the exponential loss for adaboost, and the hinge loss for the SVM.

We refer to some preexisting R functions and packages when writing the classifier models in the CVgeneric function. In the end, the CVgeneric function can process each of the **five different classifiers(LDA, QDA, Logistic Regression, SVM, and Random Forest)**, and perform cross validation across K folds with a specific loss function.

```
[[1]] Misclassification Error for each fold (LDA)
 [1] 0.1473918 0.1450207 0.1473400 0.1449994 0.1475093 0.1448926 0.1471355 0.1458004 0.1464412 0.1463131

[[2]] Loss for CV set (LDA)
[1] 0.1462844

[[1]] Logarithmic Loss for each fold (Logistic Regression)
 [1] 0.3302142 0.3298738 0.3298280 0.3293777 0.3309024 0.3297869 0.3296224 0.3301943 0.3277213 0.3305638

[[2]] Logarithmic Loss for CV set (Logistic Regression)
[1] 0.3298085
```

3. Modeling

(a) For this question, we try out substantial classifiers. We attach the result of accuracies across the folds(for individual fold & mean accuracies), and the test accuracies below for each the classifier.

The first classifier we use is the **Logistic Regression.** It assumes the linear relationship between the logic of the outcome and each independent features. In addition, this model assumes no high correlations between variables. Apparently, the dataset meets some of the assumptions, so the accuracies aren't bad as shown below:

Accuracies across Folds:

```
 [1] 0.8675390 0.8632793 0.8675652 0.8656099 0.8657757 0.8600496 0.8714670 0.8634090 0.8702973 0.8626071
[1] 0.8657599
```

Test Accuracy: `[1] 0.8872825`

The second classifier is the **Linear Discriminant Analysis(LDA)**, which assumes that the features follow the multivariate normal distribution, and the number of features is less than the sample size. Similar to the Logistic Regression, LDA also assumes the equality of covariances among features and thus show similar accuracies. Again, the cloud data satisfies some of assumptions and are classified with high accuracies.

Accuracies across Folds:

```
 [1] 0.8612355 0.8625548 0.8703846 0.8641172 0.8652695 0.8724446 0.8624238 0.8693982 0.8682111
0.8611058
```

Test Accuracy: `[1] 0.8923387`

The third classifier is the **Quadratic Discriminant Analysis(QDA).** Similar to LDA, it assumes that each variable in dataset follows the Gaussian distribution, and the number of features is less than the sample size. The cloud data still satisfies these assumptions and the accuracies are improved in this case.

Accuracies across Folds:

```
 [1] 0.8937701 0.8935648 0.8940952 0.8932996 0.8935391 0.8935477 0.8936922 0.8943604 0.8935477 0.8932825
[1] 0.8936699
```

Test Accuracy: `[1] 0.8904931`

The fourth classifier is the **Random Forest**. This classifier assumes that we obtain the best split of data at each step of generating an individual tree, for which we use the bootstrap sample, instead of the entire dataset. It also assumes that the outputs for individual trees are aggregated by averaging. From the previous parts, we can that the dataset fully meets the assumptions, thus achieve high accuracies:

Accuracies across Folds:

```
 [1] 0.9593841 0.9608767 0.9641791 0.9634692 0.9593841 0.9617409 0.9596983 0.9601697 0.9633121 0.9611910
[1] 0.9613405
```

Test Accuracy: `[1] 0.9681246`

The fifth classifier is the **Generalized Boosted Regression Model(GBM),** which holds assumptions related to the interaction depth. If the interaction depth is set to 1, it assumes a strictly additive model. As we increase the interaction depth to 5 here, this assumption is relaxed and the dataset satisfiers the assumption.

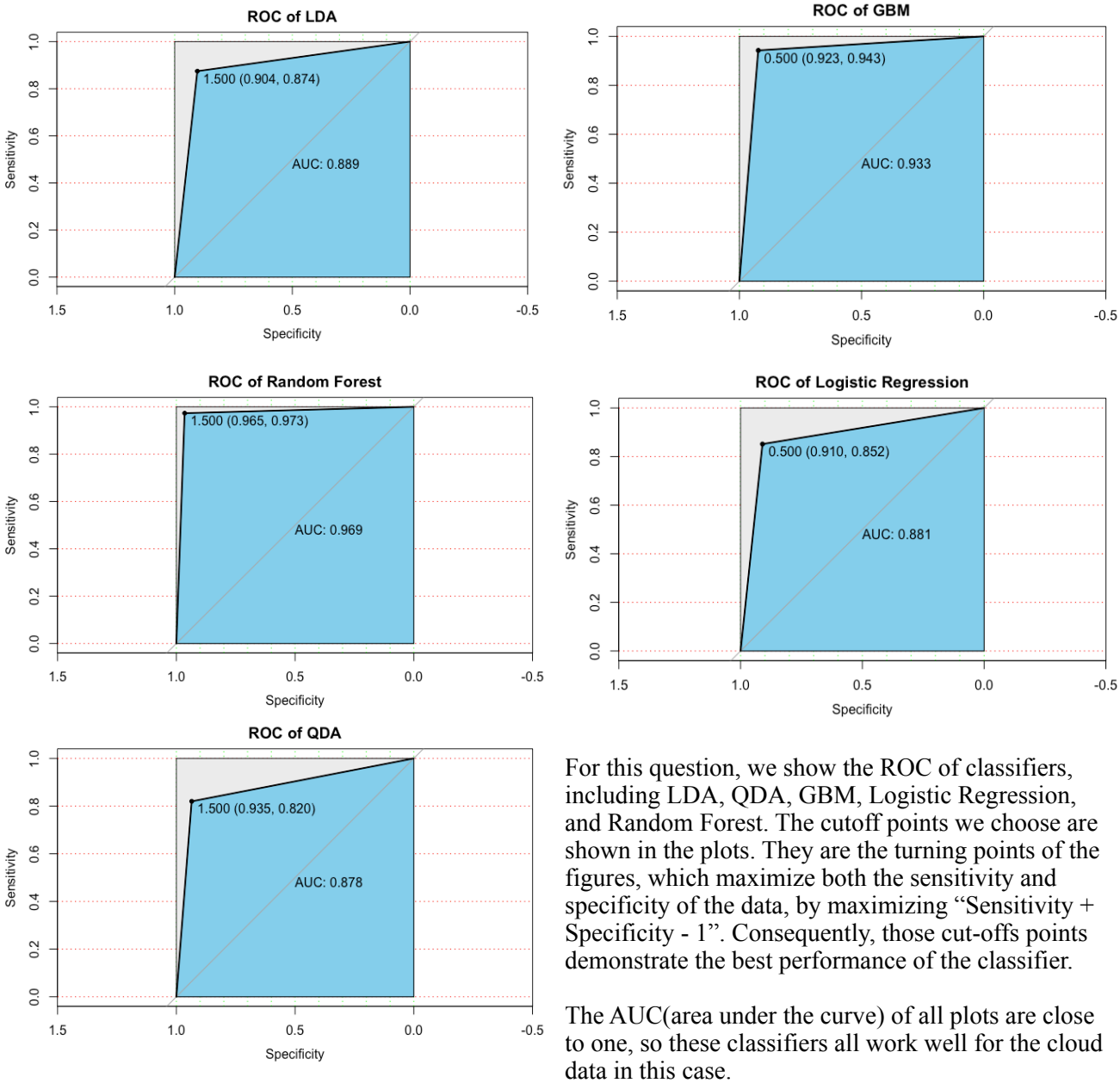
Accuracies across Folds:

```
 [1] 0.9503200 0.9502686 0.9504911 0.9502772 0.9503970 0.9501916 0.9498062 0.9503371 0.9499264 0.9500633
[1] 0.9502078
```

Test Accuracy: `[1] 0.9305008`

We also try some other models like **SVM, Bagging, Tree, Neural Network,** and **Adaboost**, but they run too slowly in R or have relatively low test accuracies in this case, so we choose not to include them.

(b)

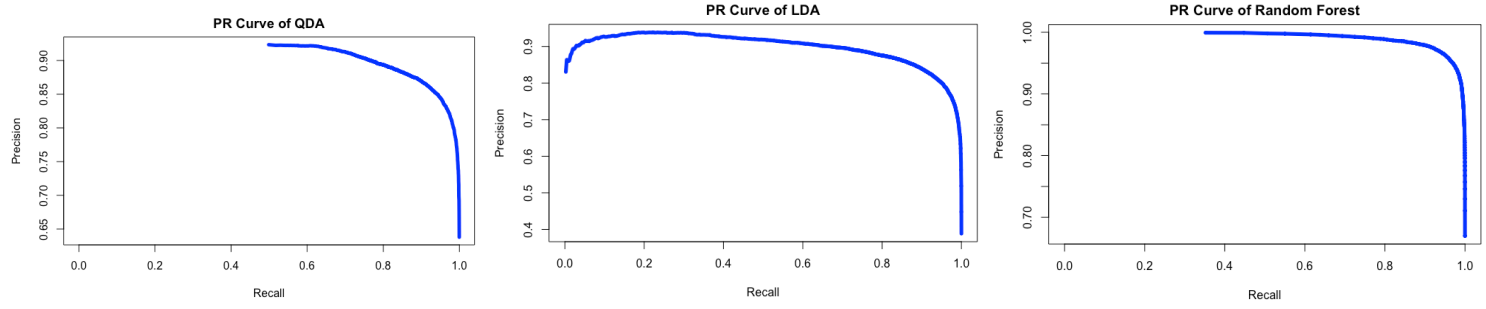


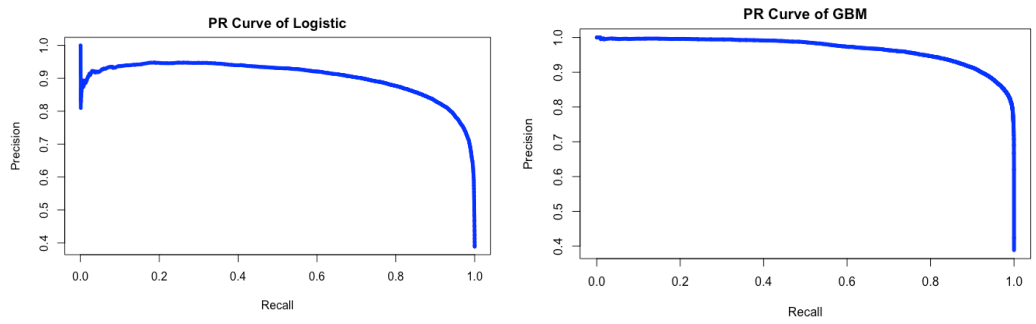
For this question, we show the ROC of classifiers, including LDA, QDA, GBM, Logistic Regression, and Random Forest. The cutoff points we choose are shown in the plots. They are the turning points of the figures, which maximize both the sensitivity and specificity of the data, by maximizing "Sensitivity + Specificity - 1". Consequently, those cut-offs points demonstrate the best performance of the classifier.

The AUC(area under the curve) of all plots are close to one, so these classifiers all work well for the cloud data in this case.

Among the ROC plots, the Random Forest has the threshold that gives the largest sum of specificity and sensitivity among all the classifier. This corresponds to the fact that Random Forest has the highest accuracy and best performance. Therefore, we can determine the classifiers with the highest accuracy by comparing their ROC cut-off values.

(c) We use the PR Curves to assess the fit of different classifiers. Its horizontal axis represents the Recall Ratio, the proportion of label 1 detected from the actual 1 labels. The vertical axis represents the Precision Ratio, the proportion of label 1 detected from all the labels. The shape of the PR Curve is the reverse of that of ROC. As a result, similar to the ROC, the PR Curve has a threshold that maximizes both the Recall and the Precision Ratio, a point shows the best performance of a classifier.





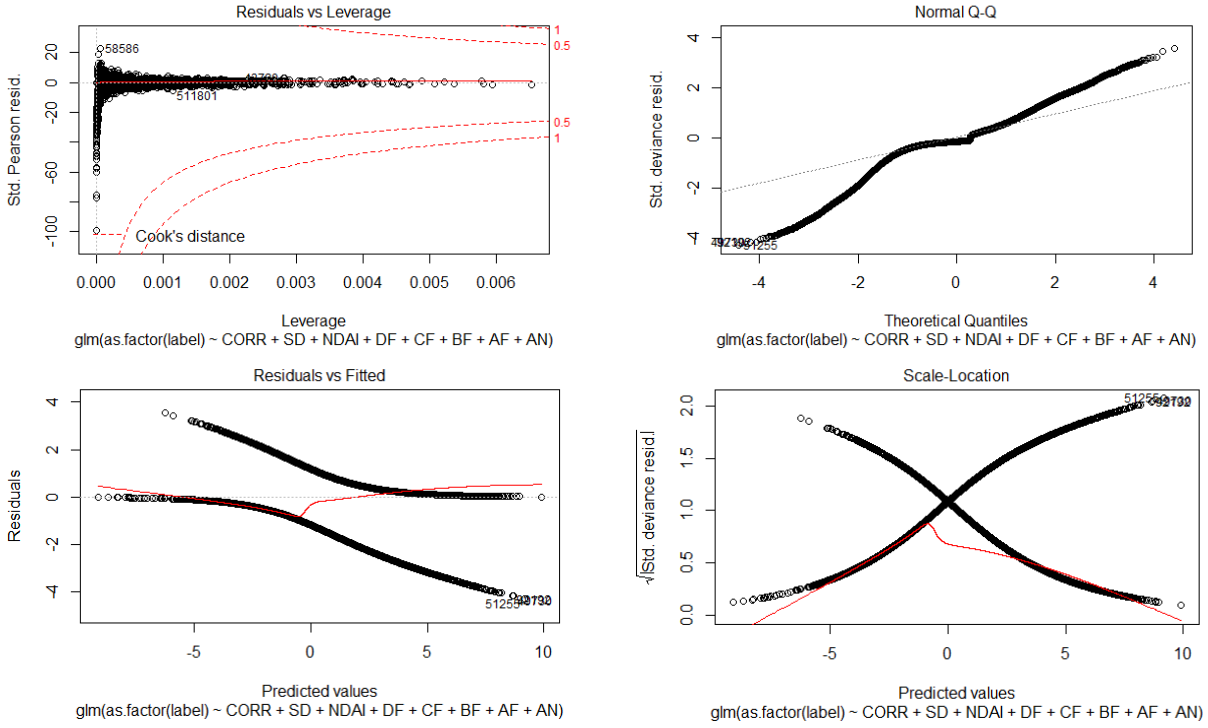
Once again, the Random Forest has the threshold that has the largest sum of Precision and Recall Ratio, among the PR Curves of all other features. By using PR Curve, we again confirm that the Radom Forest Classifier fits the cloud data best.

4. Diagnostics

(a) From 3(a), we already find the Logistic Regression and GBM(Generalized Boosting Regression Methods) to be good classifiers of the cloud data.

For the Logistic Regression, we first have the following diagnostic figures: Scale-Location plot, Normal Q-Q plot, Residual-fitted plot, and Residual-leverage plot. The Scale-Location plot displays the heteroscedasticity of the Logistic Regression, and the Residuals-leverage plot identifies the possible outliers. Both the Normal Q-Q plot and Residual-fitted plot show that the residuals of Logistic Regression aren't normally distributed.

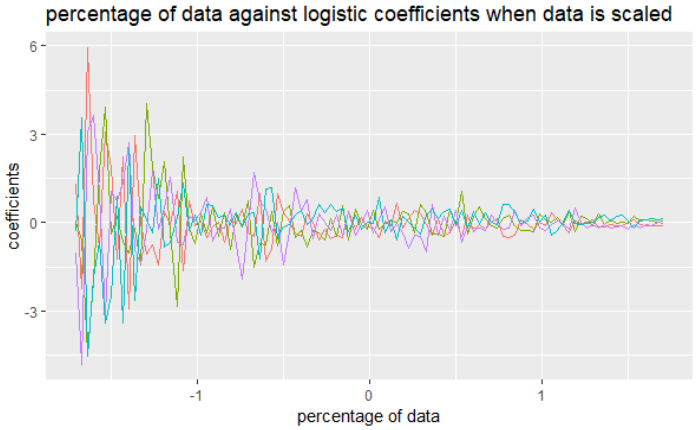
However, the residuals of logistics don't behave in the same way as those of the linear regression, so these four plots might not be helpful in estimating the parameters fairly, and we can give a better evaluation by plotting the coefficients separately.

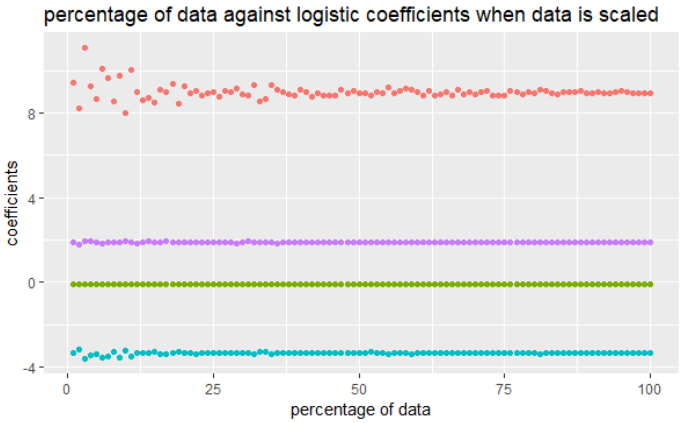


Then, we continue to diagnosing the Logistic Regression further, primarily by visualizing its coefficients(parameter estimations). The left plot below is about the percentage of the data against four different coefficients from the Logistic Regression after we scale the coefficients. We do scaling to make it easier to see the convergence. From this plot, different coefficients eventually converge and stabilize as the percentages of dataset increase.

The right plot shows the individual trend between the four coefficient percentages and percentage of data. It demonstrates that the NDAI coefficients are least stable when the amount of data is small, while the other variables, though not stable either at first, converge faster than NDAI. However, the NDAI has the largest

coefficients and intercept has the second largest, coincidentally. They are just the first and the second least stable coefficients in this plot, so it's likely to think that the larger the coefficients, the slower they converge.



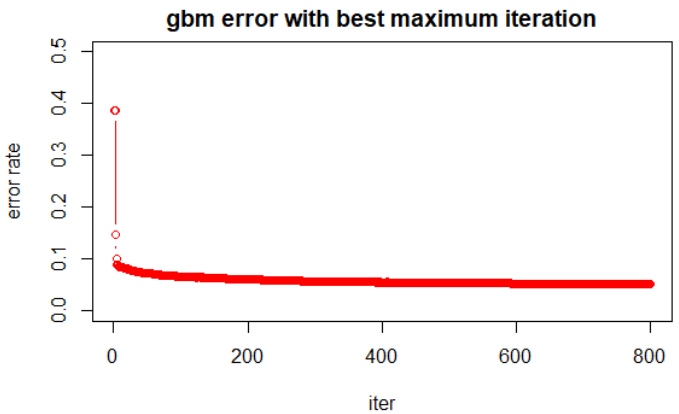


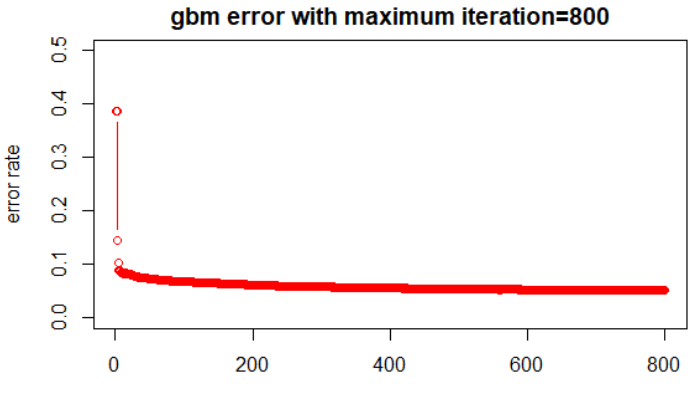
(b) In 3(a), by comparing the loss of different algorithms, we find the Random Forest and GBM(Generalized Boosting Regression Methods) to be the best classifiers of the cloud data.

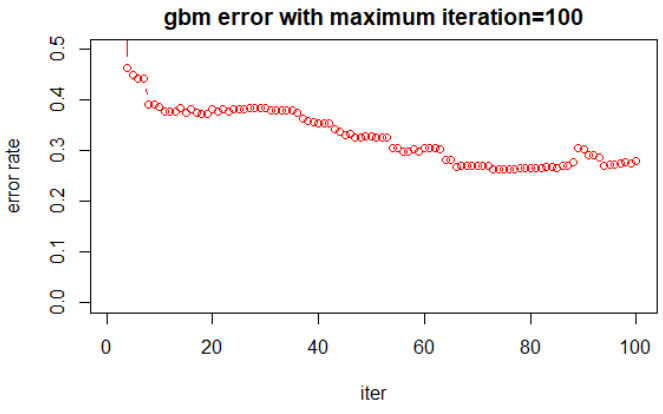
(1) For the GBM classifier, we plot the error rate with respect to the number of trees in classification, which is the iteration time in the gradient descent part of GBM algorithm. As we generate the plot in R, the number of trees(iterations) automatically range from 0 to 715. For the comparison, we then generate another two plots with trees from 0 to 800, and from 0 to 100 respectively.

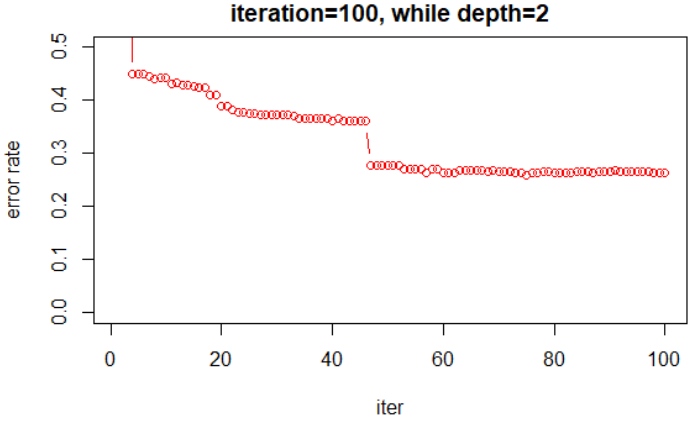
When we compare the three plots, we first find that the error rate for 800 trees is slightly larger than the rate for 715 trees. This is unexpected because usually the more the trees used, the more accurate the results are. A close-up of the plots shows that the tail of the second plot indeed rises slightly. That's to say, there is an over-fitting trend we use too many trees.

Second, we look at the third plot and observe that the variability becomes even intenser for trees from 0 to 100 trees, and no trend of convergence is observed like in other two plots. The error rates in this case are much larger than those in the other two as well. Therefore, the best number of trees(iterations) would be 715, otherwise there will be either overfitting or high variability.



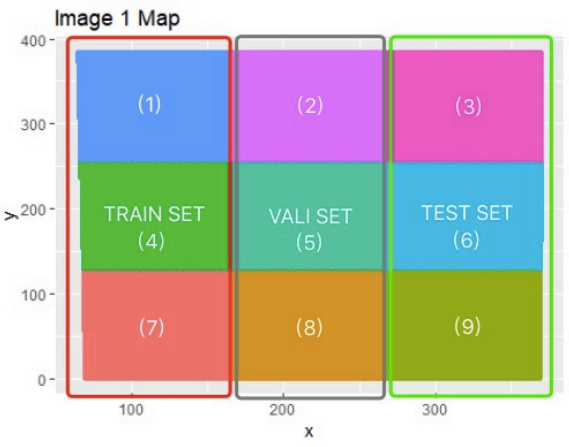




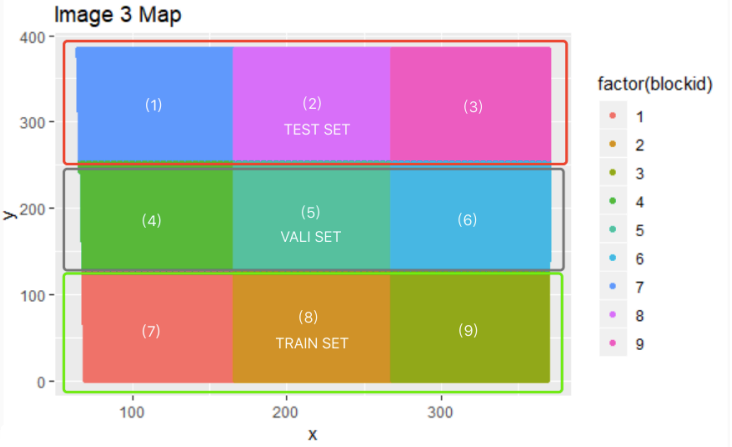


(2) For the Random Forest, we observe that the very middle square from nine squares 1 always gives the smallest error. From part 2(a), we already know the cloud data can be divided into 9 squares by Splitting Method 1. If we only divide the dataset into three main blocks(train set, validation set, and test set), we can either divide it in the vertical direction, or in the horizontal, as illustrated below.

Division Method 1(Vertical)



Division Method 2(Horizontal)



To verify our conclusion, we first train the random forest model using the training and validation set before, using the trees=160 to run faster and the training error rate is 0.0049, then compute the loss(misclassification error rate) of the three blocks following both division methods. To be more specific, we compute the loss of the assigned bock in each of the images, and calculate the mean loss as the loss indicator of the whole block. The results are displayed below. Interestingly, we find that the loss of the middle block(block 5) always leads to the smallest loss among the three, both in Division 1 and 2. This result is confirmed after we compute and compare the loss of the individual nine square as well. Thus, we conclude that the very middle square, which is the intersection of the two blocks with the least error, also gives the least error. This indicates that the MISR may collect data better at the center of the orbit.

Loss of Division Method 1(Vertical)

```
[1] 0.018775145 0.005545156 0.026957213
[1] 0.0170925
[1] 0.002524928 0.005644757 0.004073853
[1] 0.004081179
[1] 0.013860207 0.001868137 0.023664587
[1] 0.01313098
```

Train Loss: image1, image2, image3
Mean Loss of the Training Block
Validation Loss: image1, image2, image3
Mean Loss of the Validation Block
Test Loss: image1, image2, image3
Mean Loss of the Testing Block

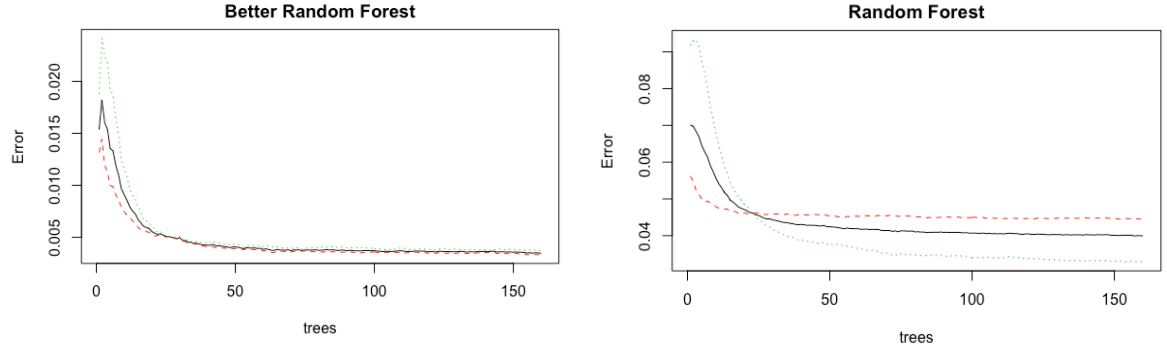
Loss of Division Method 2(Horizontal)

```
[1] 0.030689943 0.008218131 0.030512560
[1] 0.02314021
[1] 0.004240440 0.003720292 0.007993006
[1] 0.005317913
[1] 0.0025338964 0.0007803017 0.0093461969
[1] 0.004220132
```

Train Loss: image1, image2, image3
Mean Loss of the Training Block
Validation Loss: image1, image2, image3
Mean Loss of the Validation Block
Test Loss: image1, image2, image3
Mean Loss of the Testing Block

(c) Since we've tried many classifiers in 3(a), and the best classifier Random Forest reaches 96% test accuracy, we will look for and apply better features, instead of finding a new classification method.

Commonly, the more features used in classifier, the more residuals can be explained by the features unless there is an overfitting trend. Because the Random Forest classifier accounts for the correlation between different variables, overfitting doesn't occur frequently, so we try to find better features from the original features. We try increasing the power of the original features, as to increase their variability.



After several attempts, we find the better features to be the squares of the original features, except for the radiation features in different angles. To be more specific, the better features are x^2, y^2, $NDAI^2$, $CORR^2$, SD^2, which increase the test accuracy significantly, compared to the original test accuracy in 3(a):

```
[1] 0.9590503   Test Accuracy of Random Forest (from 3(a) )
[1] 0.9961742   Test Accuracy of RF with Better Features: x², y², NDAI², CORR², SD²
[1] 0.9963472   Test Accuracy of RF with Even Better Features: x², y²
```

However, if we further increase the powers of each features, such as using the cubic of features, the test accuracy remains constant. Accordingly, the Random Forest classifier reaches its limit of explaining the correlation between the features, and the Test Accuracy improves significantly only with the squares of the features.

We then plot the Error Rate of Random Forest against the number of trees, to see the effects of better features. In the plots, the dashed red line refers to the error rate of -1 labels, the dotted green line is the error rate of 1 labels, and the black line is the out of bag error, the error of samples without bootstrap. The better the classifier is, the faster the convergence of these three variables. Accordingly, in the case of Random Forest with better features, the convergence occurs around 50 trees. In contrast, the convergence of the original classifier doesn't occur at all even for 150 trees. As a result, the Random Forest with better features win over, because its error rate decreases fast as the number of trees increases without overfitting, and the parameters also stabilize fast.

(d) For this part, we use the Splitting Method 2 and again perform the analysis of the best classifiers, Random Forest and GBM.

(1) Under Splitting Method 2, the results for Random Forest change a little bit but the middle square(from 4(b)) still has the smallest misclassification error, consistent with the results from the previous parts. Because the train set in this case partly overlaps with the test set, there're some zero losses in each of the Division Methods. Nevertheless, this doesn't change the mean losses and the results, which provide strong evidence that the middle squares of the data has the least loss and thus the best classification, for each of the three images. Therefore, the conclusions for Random Forest method from previous part don't change under different splitting methods.

Loss of Division Method 1(Vertical)

```
[1] 0.09155118 0.02066189 0.06139886   Train Loss: image1, image2, image3
[1] 0.05787064                         Mean Loss of the Training Block
[1] 0.00000000 0.09818741 0.00000000   Validation Loss: image1, image2, image3
[1] 0.03272914                         Mean Loss of the Validation Block
[1] 0.002700556 0.001047979 0.031273779  Test Loss: image1, image2, image3
[1] 0.0116741                          Mean Loss of the Testing Block
```

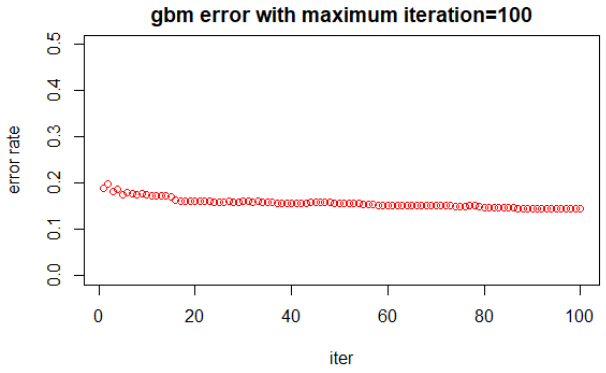
Loss of Division Method 2(Horizontal)

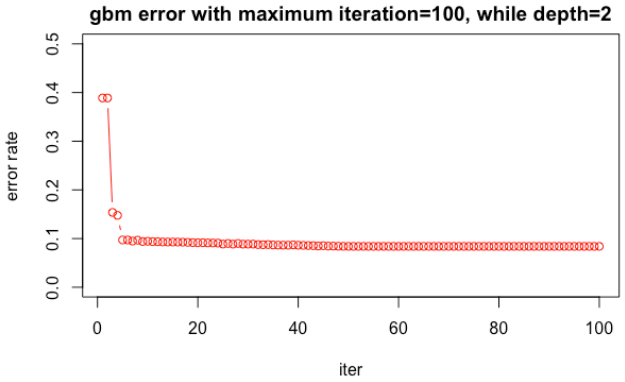
```
[1] 0.00000000 0.11830286 0.05893174   Train Loss: image1, image2, image3
[1] 0.0590782                          Mean Loss of the Training Block
[1] 0.0548653474 0.0008307448 0.0000000000  Validation Loss: image1, image2, image3
[1] 0.01856536                         Mean Loss of the Validation Block
[1] 0.02849522 0.00000000 0.01765871   Test Loss: image1, image2, image3
[1] 0.01538464                         Mean Loss of the Testing Block
```

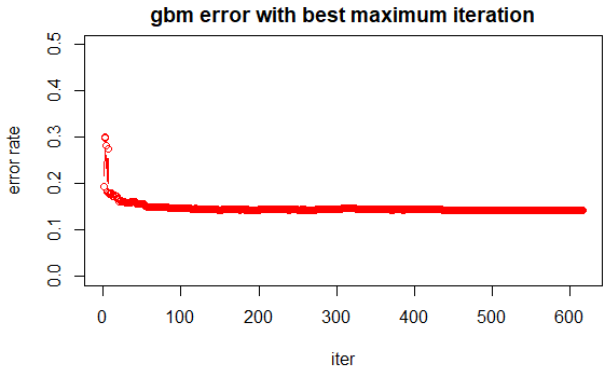
(2) For the GBM, the conclusions become different under Splitting Method 2. From 4(b), we conclude that from the plots with different numbers of trees, the rate of convergence varies, and the fastest convergence corresponds to the best number of trees(iterations), otherwise there will be issues like overfitting or high variability of the error rate.

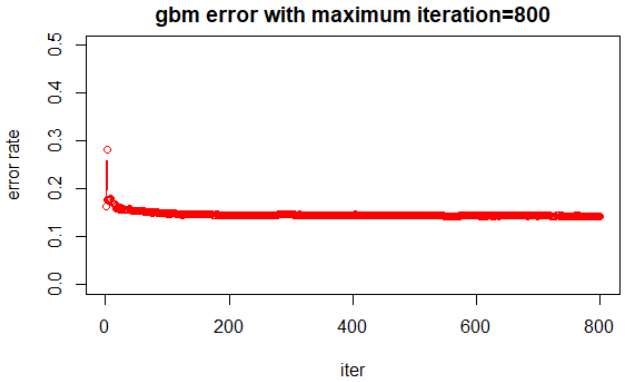
Again, we make plots about the number of trees(iterations) and the error rate. This time, the plots all look the same for different numbers of iterations, and the error rates now all converge fast and have lower variability, despite the increase in error rates. Although we already know that the overfitting occurs only for more than 715 trees, the convergences now are much faster than those in 4(b). Also, the variabilities are much lower under Splitting Method 2 than those from the previous parts. From 4(b), the variability is pretty intense and no convergence occurs for trees from 0 to 100 trees; but now the error rates converge for 100 trees.

As for the best number of trees, we find when we use splitting method 2, we only need 615 trees to reach the minimum error and the computational cost decreases. Hence, if we want a lower error rate, we can choose the train set more randomly using the Splitting Method 1, but if we seek for decreasing the cost for computation, we should use the Splitting Method 2.









(e) In this project, we explore the properties of the cloud data by looking into their correlations and plots. The other main goal is to find classifiers that fit the data best and identify the labels accurately. This can help us effectively distinguish the cloud-free region from the clouds, by classifying the labels of data.

First, we plot the cloud data and observe if there's any pattern associated with a certain distribution. From the figures, we conclude that the cloud data isn't identically and independently distributed across the three images, but the cloud itself follows i.i.d. in each of the image. Next, we develop two different methods of splitting that doesn't violate the assumptions before.

We then find the three physical features to be the best predictors, due to their high associations with each labels. After knowing the characteristics of the data, we fit a variety of classification methods and select the best ones based on their algorithms and performances. After trying methods including Neural Network, QDA and Adaboost, we gather their results and do a series of in-depth analysis, such as parameter convergence. In the end, we find the Random Forest to be the best predictor of cloud data, and the GBM to be the second best in this case. They not only have the highest test accuracy(95.9%) among all, but also avoid issues like overfitting.

Next, we continue improving the Random Forest method by finding better features. We try increasing the power of each features and find their squares to be the better features: x^2, y^2, $NDAI^2$, $CORR^2$, and SD^2. These better features help improve the test accuracies up to 99.6%, so the classification works pretty well.

Nonetheless, this doesn't suggest that the Random Forest is a perfect classifier. It still has limited test accuracy and feature improvement, as shown in 4(c). The reason for its accurate fit to the cloud data is that the dataset already meets the assumption for Random Forest method. Therefore, it is possible that this algorithm may not work well for other types of dataset.

To summarize, there's no perfect algorithm that can do classification with 100% test accuracy. All we can do is by following the PQRS principle, to make sure that the model is in accordance with the population and question in the domain, with the representativeness of samples and scrutiny of the process.

5. Contributions and Credits:

Contributions:

All parts of the project shall be the collective effort by Evelyn Zou and Hanting Fei, who both contribute to the computations, visualization and explanations for each of the questions.

Sources of References:

1. "Introduction to Random Forest"
https://www.datascience.com/resources/notebooks/random-forest-intro

2. "Random Forests by Leo Breiman and Adele Cutler"
https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

3. "Gradient Boosting Machines"
http://uc-r.github.io/gbm_regression

4. "Gradient Boosting from scratch"
https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

5. "A Beginner's Guide to Neural Networks and Deep Learning"
https://skymind.ai/wiki/neural-network

6. "The Relationship Between Precision-Recall and ROC Curves"
https://www.biostat.wisc.edu/~page/rocpr.pdf

7. **"Precision-Recall"**
https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

Github Link of Project: https://github.com/zoulanlan/STAT-154-Project-2