Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data

Jen Shi, Alina Skripets

Part I

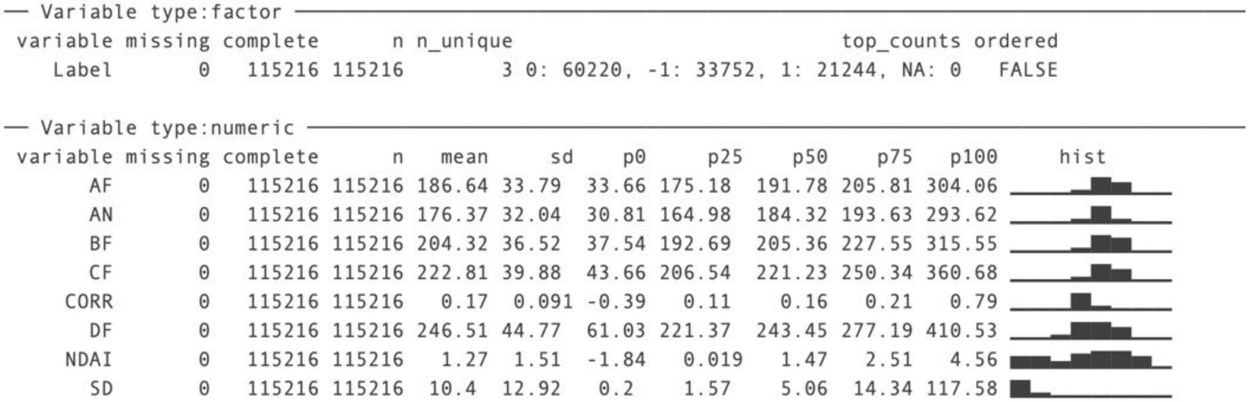
Paper Summary

The study contributes to the quintessential Earth science issue of the century—climate change. Specifically, the research team made considerable progress towards cloud identification over the Arctic which can potentially aid in the study of dependences of surface air temperatures on the levels of carbon dioxide in the atmosphere. A major challenge that researchers faced in the process is the vast volume of data collected by the Multiangle Imaging Spectro Radiometer (MISR) launched by NASA in 1999. Making use of the data from 10 orbits over 16 different angled paths and spanning 144 days from April through September of 2002, the team's exploration was based on 7,114,248 1.1 km resolution pixels with 36 radiation measurements per pixel.

Researchers developed features based on the 275m red radiation measurements and selected three features: CORR, SD and NDAI. High values of CORR allowed the differentiation of cloud-free conditions and indicated the presence of high-altitude clouds. SD helped identify smooth surfaces that could have been falsely identified as cloud due to their low CORR. Finally, NDAI contributes to the distinguishing between ice and snow-covered surfaces form low-altitude clouds. The creation of a set of features that enable the algorithm to fully distinguish pixels with clouds and clear also resulted in a data asset that is separable in low dimensions and facilitated the research process. The authors then employed a three step ELSM algorithm consisting of feature construction, LDA and QDA, to come up with a novel approach for cloud identification approach and reached a new level of efficiency (in terms of sensitivity and specificity*)*. The algorithm focused on the identification of cloud-free pixels instead of the cloudy pixels common in traditional approaches. The result of the study was a significant improvement in the accuracy of identification -- 91.80% versus the highest rate of 83.23% that was previously attainable. In addition, the paper served as a reminder of the intimate connection between statistics and Earth science. The impact that statisticians may have on other areas of science as a result of collaborative efforts cannot be underestimated.

b) *Data summary/Description*

A brief summary of our data by variable is shown below. Several interesting characteristics to note at first glance include: the DF angle seem to have a higher standard deviation and mean compared to the rest of the angles. Additionally, we observe that on average our correlation is slightly positive.



```
— Variable type:factor ——————————————————————————————————————————————————————
 variable missing complete      n n_unique                           top_counts ordered
    Label       0   115216 115216        3 0: 60220, -1: 33752, 1: 21244, NA: 0   FALSE

— Variable type:numeric —————————————————————————————————————————————————————
 variable missing complete      n   mean    sd     p0    p25    p50    p75   p100     hist
       AF       0   115216 115216 186.64 33.79  33.66 175.18  191.78 205.81 304.06
       AN       0   115216 115216 176.37 32.04  30.81 164.98  184.32 193.63 293.62
       BF       0   115216 115216 204.32 36.52  37.54 192.69  205.36 227.55 315.55
       CF       0   115216 115216 222.81 39.88  43.66 206.54  221.23 250.34 360.68
     CORR       0   115216 115216   0.17  0.091 -0.39   0.11    0.16   0.21   0.79
       DF       0   115216 115216 246.51 44.77  61.03 221.37  243.45 277.19 410.53
     NDAI       0   115216 115216   1.27  1.51  -1.84   0.019   1.47   2.51   4.56
       SD       0   115216 115216  10.4  12.92   0.2    1.57    5.06  14.34 117.58
```

Expert labeling was utilized to provide the data for building and testing the model. Based on the data for the three images, the expert had identified and labeled between 50 and 70% of the pixels. Out of the pixels classified for the images 1, 2 and 3, around 70%, 50% and 60% of the pixels were cloud free, respectively (See figures 1, 2, 3 below).

In the absence of computational constraints, it is not reasonable to assume that the pixels are independent and identically distributed. We can see this since intuitively clouds are likely to be surrounded by other patches of clouds and therefore a pixel of cloud is more likely to be surrounded by other pixels of clouds. Similarly, the surface is smooth and clear pixels are likely to be surrounded by other cloud-free pixels.

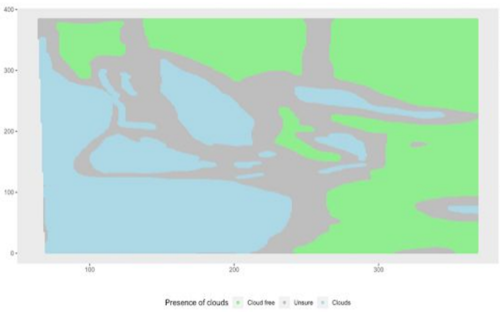


Figure 1

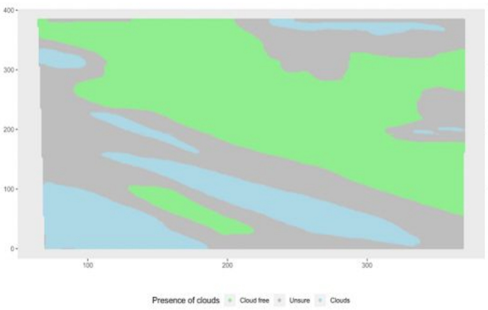


Figure 2

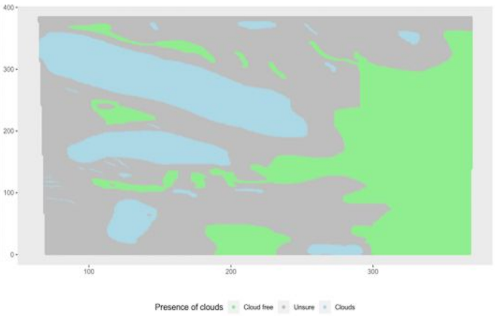


Figure 3

The hypothesis of the non-i.i.d. distribution is further supported by the clustering of clear pixels in certain parts of the image and cloudy pixels in others. This is easy to see from the boxplot of x-y coordinates broken down by cloud status. From Figure A, we note that cloudy pixels tend to be distributed around one value of y-coordinate, while cloudless pixels are distributed in a different region. For example, in A, the clear pixels are located between 300 and 200 on the y-axis, while cloudy pixels are below 125. This mirrors the information in Figure 2(expert labels), where we see a cloud below and land above. If we randomly generate the labels, we are unlikely to obtain an image such as in Figure 2. However, this problem is addressed in the paper by appropriate feature selection. Both CORR and SD consider the smoothness of pixels and attempt to benefit from the information contained in patches of pixels. Treating data as i.i.d. makes sense in light of computational difficulty associated with computing 115216 correlations.

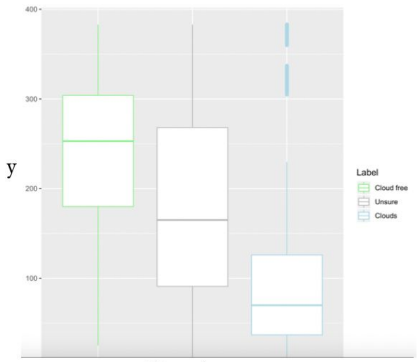


Figure A

EDA

Shown on the right (figure 4) is a correlation plot of our variables. The correlations are indicated by the color scale on the right. Some interesting things we observed include: the angles are perfectly correlated with each other, NDAI has a constant negative correlation with all angles of the cameras, and SD is positively correlated with NDAI while being negatively correlated with all camera angles.

It is reasonably expected that the different angles of the camera would capture images in a synchronized manner. Furthermore, the radiation index is consistently negatively correlated with camera radiance angles perhaps due to the reflectance characteristics of radiation over the arctic region (ice vs cloud vs lower altitude cloud). More expected is the fact that SD has a

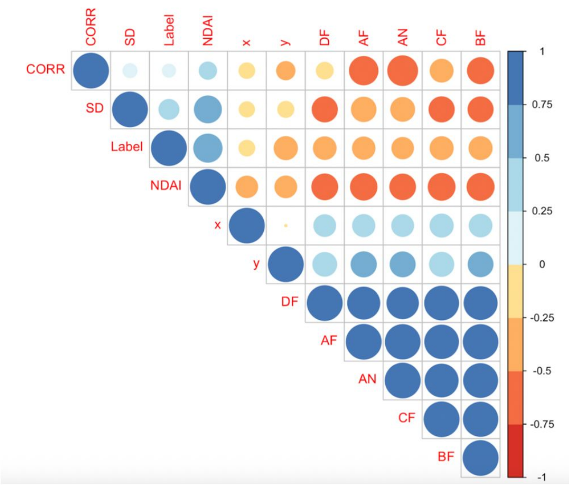
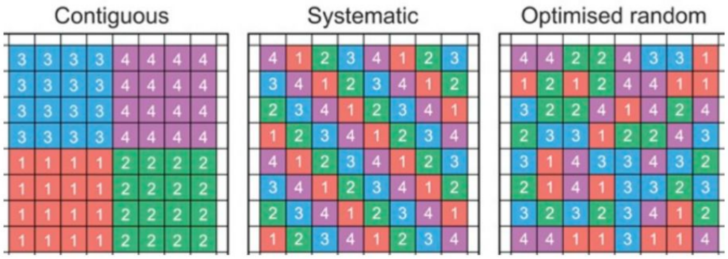


Figure 4

positive relationship with NDAI which can potentially be explained when viewing radiation as a complicating factor.

Part II

Division of Data



Contiguous

3	3	3	3	4	4	4	4
3	3	3	3	4	4	4	4
3	3	3	3	4	4	4	4
3	3	3	3	4	4	4	4
1	1	1	1	2	2	2	2
1	1	1	1	2	2	2	2
1	1	1	1	2	2	2	2
1	1	1	1	2	2	2	2

Systematic

4	1	2	3	4	1	2	3
3	4	1	2	3	4	1	2
2	3	4	1	2	3	4	1
1	2	3	4	1	2	3	4
4	1	2	3	4	1	2	3
3	4	1	2	3	4	1	2
2	3	4	1	2	3	4	1
1	2	3	4	1	2	3	4

Optimised random

4	4	2	2	4	3	3	1
1	2	1	2	4	4	1	1
3	2	2	4	1	4	2	4
2	3	3	1	2	2	4	3
3	1	4	3	3	4	3	2
2	1	4	1	3	3	2	3
3	2	3	2	3	4	1	2
4	4	1	1	3	1	1	4

Figure 5

As shown in the three- panel demonstration (figure 5) below, our three methods are contiguous, systematic and optimized random. The most prominent distinguishing factor among the three models of division is the way we assign identifiers after dividing our data into identical k by k regions. Our general proportionate division of data is as follows: 10% test (hold-out set), 20% validation and 70% training.

The optimized random method consists of a division of our data set into n blocks of k by k regions. We then assign a number to each block and randomly parse the numbers into training, validation and test sets. This method could preserve the patches labeled cloud or clear and accounts for the smoothness of our data. However, this method does introduce some randomness into our samples which may not be totally realistic considering the fact that cloud is more likely to be next to cloud and vice versa. The Contiguous method only randomizes assignment of training, validation and test once and assigns all pixels in the vicinity an identical grouping. A potential major drawback to this type of division will be especially pronounced if cloudy pixels or clear pixels aggregate together in one specified region. It is possible to unintentionally train our model on a vastly cloudy or clear section of our data. Lastly the Systematic method provides a principled assignment of training, test and validation to the data set where the assignment of groups rotates which could potentially mitigate our previous dilemma for aggregating pixels.

Baseline accuracy of a trivial classifier

The accuracy of the trivial classifier is around 62.5%, that is the fraction of pixels correctly classified by setting all labels to clear. At first glance this rate of accuracy is quite high. A trivial classifier generally has high accuracy in unbalanced samples with a non-uniform number of samples from each level. Therefore, when the pixels with clouds are rare and far between, high accuracy for a cloud-free classifier can be expected. However, this result is meaningless when we are interested in identifying the more isolated cloud pixels that exist amongst the non-cloud pixels.

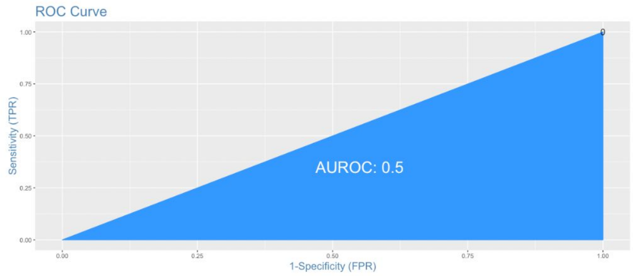


Figure 6

Feature Selection

The criteria that we employed when choosing our three features are as follows. Taking into consideration that most methods for data classification we can utilize rely on our data being separable in low dimensional space, we looked for

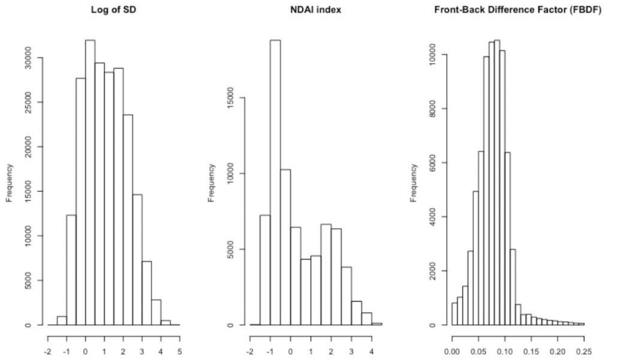


Figure 7

covariates where this would be true. Another important criterion is that our features have to be predictable, computable and stable (PCS-compliance as in lecture).

Our initial data analysis revealed that the pixels were not i.i.d. which prompted us to look for features that can explicitly consider the relationship between pixels and compensate for our computational constraint. Specifically, we identified SD as the first feature to test. This feature is easily obtainable from the data set, as per the method suggested in the paper. Additionally, it doesn't impose computational constraints especially considering that SD of land pixels does not vary over time or location. We took the square root of SD for better separability by spreading out the distribution of this specific feature (see figure 7). The second feature we picked is NDAI. This feature was evidently useful in previous studies and is great for distinguishing cloudy areas and smooth areas from clear areas/ areas with low altitude cloud.

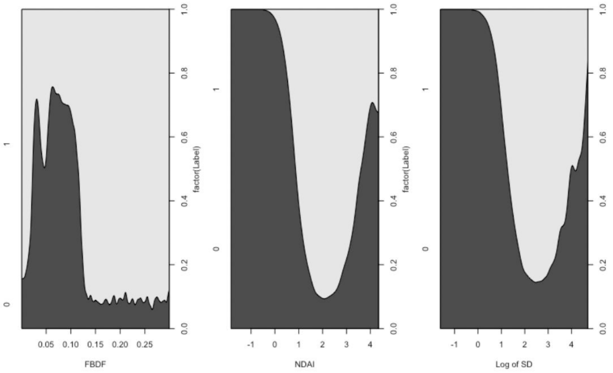


Figure 8

We spent a long time screening for a suitable third feature. Among others, we looked at the SD of SD, noting that clouds tend to have higher variation of SDs across quadrants. We tested some synthetic features that lacked good interpretability like SD of CORR among the neighboring pixels. We looked into various combinations of radiation readings, however, none of the combinations met our stability standard. We did notice that radiation readings were higher for regions with clouds and that the difference between the radiation readings of the front and back (An and Af) cameras is smaller for cloudy regions than for clear regions.

Inspired by the NDAI index, we created our own measure: the Front-Back Difference Factor (FBDF), as calculated according to the following formula: $\frac{An+Af}{Af}$ The feature worked well in terms of separating the data, as shown in the figure 8. Clearly, lower values of FBDF occurs more frequently for clear pixels than for ones with clouds. Additionally, it was stable enough on different data portions and interpretation coincided with our understanding of underlying scientific principles. Additionally, it was stable enough on different data portions and interpretation coincided with our understanding of underlying scientific principles. Figure 8 displays the density of each feature for cloudy and clear pixels. Our analysis shows that log(SD), NDAI and FBDF are effective at separating the data.

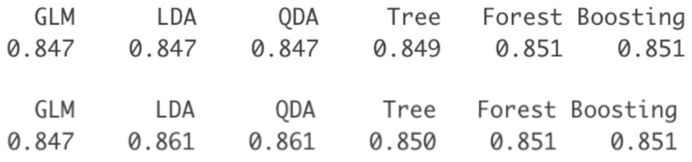
Generic CV Function

To increase the reproducibility of the analysis and streamline the CV process, we introduced the Generic CV function "*CVGeneric*". The user may select between 6 different classification methods: GLM, LDA, QDA, Decision Tree, Random Forest, Boosting. In addition, the function supports several loss function types (and fit assessment criteria): L1, L2, Accuracy, AIC, LogLoss, ZeroOne, AUC, Precision, Sensitivity, Specificity. The function is subsequently used in Part III (a) to provide CV measurements and in Part III (c) to explore other relevant metrics that guided our model selection.

PART III

Classification methods & fit assessment

The logistic regression model assumes the following: that the outcome is a binary variable (in our case cloud or no cloud), 1 vs -1; that the relationship between the logit of the outcome and each predictor variable is linear (logit function is logit(p) = log (p/(1-p)) where p is the probabilities of the outcome; that there is no influential values in the continuous predictors (extreme outliers), and that there is no high intercorrelations among the predictors. We cleaned out the extreme values form the the long tail of FBDF distribution and examined our correlations prior to applying the logistic model. All of the aforementioned assumptions are met. The penalty for missing some of the conditions would be decreased accuracy, so our error rates also serves as validation for the satisfaction of the assumptions. Specifically the GLM-logit model yielded an AIC of 90,150 for us. We chose this specific model after trying to implement other features within GLM but ultimately eliminated them due to high AIC values.



```
  GLM     LDA     QDA    Tree  Forest Boosting
0.847   0.847   0.847   0.849   0.851    0.851

  GLM     LDA     QDA    Tree  Forest Boosting
0.847   0.861   0.861   0.850   0.851    0.851
```

We then performed LDA and QDA and found that LDA performed better than QDA on an accuracy basis. Additionally we fit a decision tree onto our data which yielded a 0.82 rate of correct classification. Furthermore, the random forest method gave us a 0.92 rate of correct classification. We are pleasantly surprised by the low rate of misclassification especially from our tree methods. A summary of the accuracies across folds are provided here (averages). The top line represents the first split and the second summary represents the second split. A detailed version across folds can be found in the appendix (3a).

ROC curve comparison

The ROC curve shows the tradeoff between sensitivity (TPR) and specificity (1-FPR). Classifiers that are closer to the top left corner indicate better performance. We want to avoid any curves that come close to a 45 degree angle which indicates no predictive power whatsoever. Since the ROC curve doesn't depend on class distribution it is a great tool for evaluating classifiers that predicts rare occurrences. The *AUC* values for each of the tested models are as follows: *LDA(0.816), QDA (0.8159), GLM:(0.8205), Tree(0.7947), Forest(0.8193), Boosting (0.8273).* As we can see in the plot below, the red highlighted ROC curve representing Boosting and the blue curve from Forest are the two best performing methods according to the AUC (area under the curve).

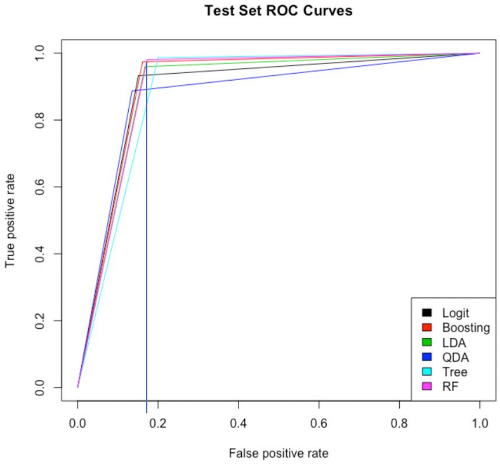
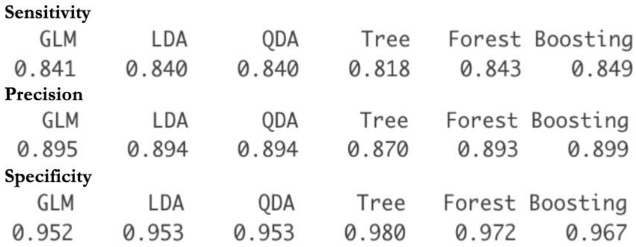


Figure 9

We choose a cut off value of 0.18 which corresponds to the intersection between the Tree ROC curve and that of the Boosting method. The reason for choosing the intersection of Boosting and random forest is that while our objective is to maximize our TPR and 1-FPR, we also note that the AUC for both of these methods are very close. A cutoff value of 0.18would be a compromise between these two methods (average of the two kinks). This intersection and the indication of the cutoff can be found on figure 9 as a blue vertical line.

Other relevant techniques of assessing fit

The summaries on the right shows us the sensitivity, precision and specificity centric methods of assessing fit. Using each of these as the key metric, the corresponding fit of our models are



Sensitivity

```
  GLM     LDA     QDA    Tree  Forest Boosting
0.841   0.840   0.840   0.818   0.843    0.849
```

Precision

```
  GLM     LDA     QDA    Tree  Forest Boosting
0.895   0.894   0.894   0.870   0.893    0.899
```

Specificity

```
  GLM     LDA     QDA    Tree  Forest Boosting
0.952   0.953   0.953   0.980   0.972    0.967
```

as summarized on the left. We see that in the context of sensitivity-specificity trade off Boosting is still the best. However, based on precision alone, Boosting ranks on the lower end with GLM, LDA and QDA leading the ranks. For more precise reporting of the three metrics for each of the 10 folds as well as the average value, see Appendix.

$$Sensitivity = \frac{\#\,Actual\ 1's\ and\ Predicted\ as\ 1's}{\#\ of\ Actual\ 1's}$$

$$Specificity = \frac{\#\,Actual\ 0's\ and\ Predicted\ as\ 0's}{\#\ of\ Actual\ 0's}$$

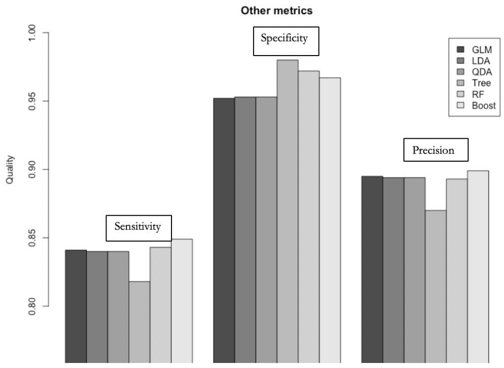


Figure 10

Part IV

Classification Model

We chose the Boosting model for analysis. Since our data set is large, low bias/high variance classifiers start to win out as our training set grows because they have lower asymptotic error. We know that high bias classifiers aren't powerful enough to provide accurate models. After performing a *Kolmogorov-Smirnov statistic* analysis on our Boosting model we plot the effects of our model predictions on the KS plot below. After computing the prediction probability scores from our boosting logit model, we sort these values in descending order of prediction probability scores. We then split our set of values into 10 groups (or ranks) as marked on the horizontal axis such that the group ranked 1 contains the top 10% of values with highest prediction probability scores, group ranked 2 containing the next 10% and so on. The resulting plot below can be interpreted as: for example, by targeting first 40% (rank 4.0) of the set, our model will be able to capture 80.97% of total cloudy pixels, while without the model, you can expect to capture only 40% of clouds.

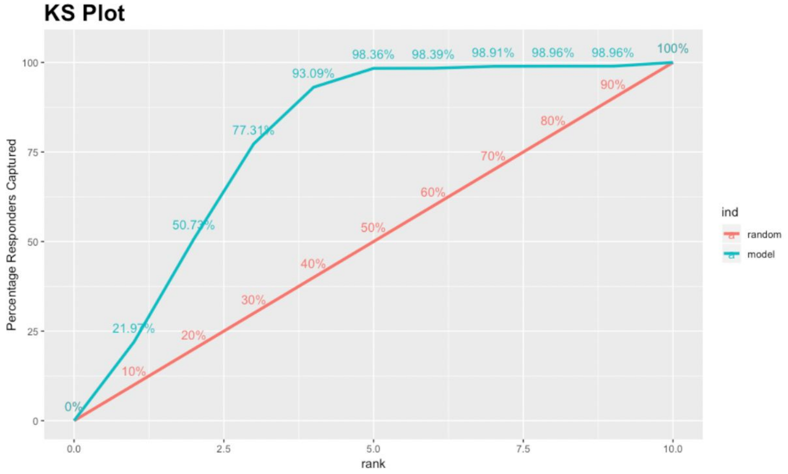
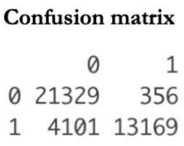


Figure 11

Best Classification Models

In our Boosting classification model, the confusion matrix on the left shows a brief summary of correct classification vs misclassifications of both type 1 and type 2. Here we are interested in examining our boosting model from the perspective of test errors which help guide the selection as well as the tuning of our models and features.



Confusion matrix

```
       0     1
0  21329   356
1   4101 13169
```

We plotted the Boosting test errors in terms of accuracy and zero-one loss against both the number of tree models fitted as well as the value of our shrinkage parameter(s) below.

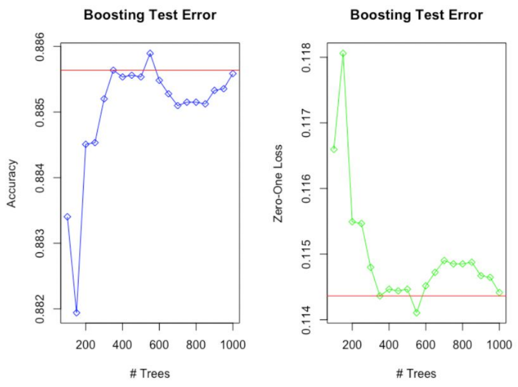


Figure 12

We notice that our accuracy stops increasing substantially once we pass the 300 mark. As indicated by the red fitted line below, we consider this value an appropriate cutoff to effectively preserve accuracy. The same analysis is conducted with respect to zero-one loss. The CV process was repeated to tune the model with respect to shrinkage parameter and interaction depth. Our analysis revealed the optimal values for shrinkage parameter and interaction depth are 0.01 and 8, respectively.

Better Classifier & Future Predictions

Based on our analysis thus far, we have discovered the relative proportionate importance of our chosen variables (features) are as plotted in the pie chart on the right (Figure 13). As we see here, NDAI accounts for a striking proportion of the prediction power of our model.

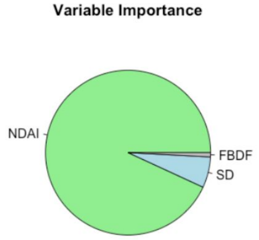


Figure 13

Study of residuals

After fitting the model we carefully examined the misclassified data for patterns and correlations with features. We have discovered that there remains some correlation between misclassified pixels and NDAI. In particular, for lower values of NDAI, the true label of pixels is almost always cloudy. (See Figure 14) Same is true for SD. Lower values of SD indicate cloud. Comparing this with Figure 14, we notice that the pattern in residuals is reverse of that in the original data set, where high SD and high NDAI indicated a cloud. FBDA variable holds little further information. Additionally, there is little spatial correlation in the residuals as seen in the Figure 14 (c and d). The spread of coordinated is independent of true labels now and warrants an assumption that with respect to spatial dependencies, the misclassified pixels are i.i.d.

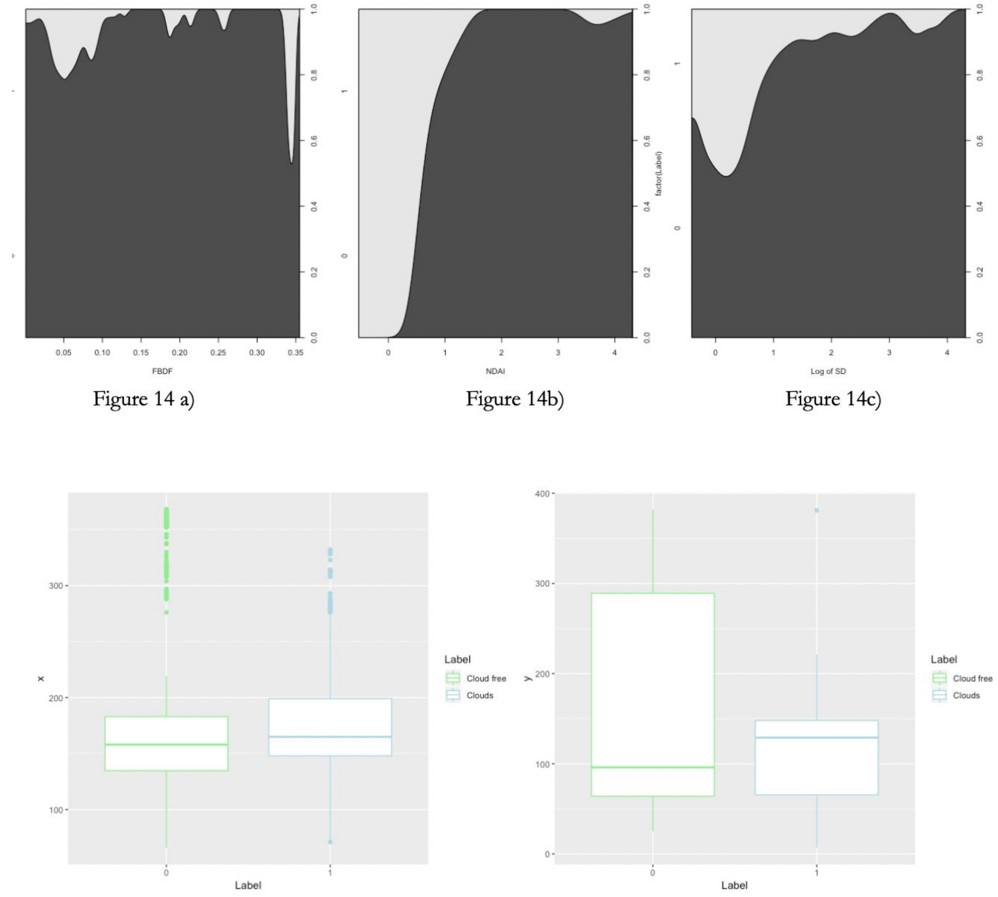


Figure 14 a)

Figure 14b)

Figure 14c)

Figure 14 D)

Figure 14 E)

To conclude our discussion of misclassified pixels, we refer to the Figure 15 on the following page. The pictures in the first column display the true labels assigned by the expert. The images in the second column are colored according to Gradient boosting model classification results. Lastly, the misclassified pixels are highlighted in red in the third columns.

For convenience, the pixels that the expert was unsure about were temporarily et to NA, since they do not aid our understanding of fit accuracy. From the third column images, we observe that our prediction is conservative in nature. Keeping in mind that the identification of clouds over Arctic is important for climate change studies, we favor Type I error of classifying non-cloudy pixel as a pixel with cloud. Our algorithm is tuned to minimize Type I error and notably miss classifies land as cloud more frequently than the reverse.

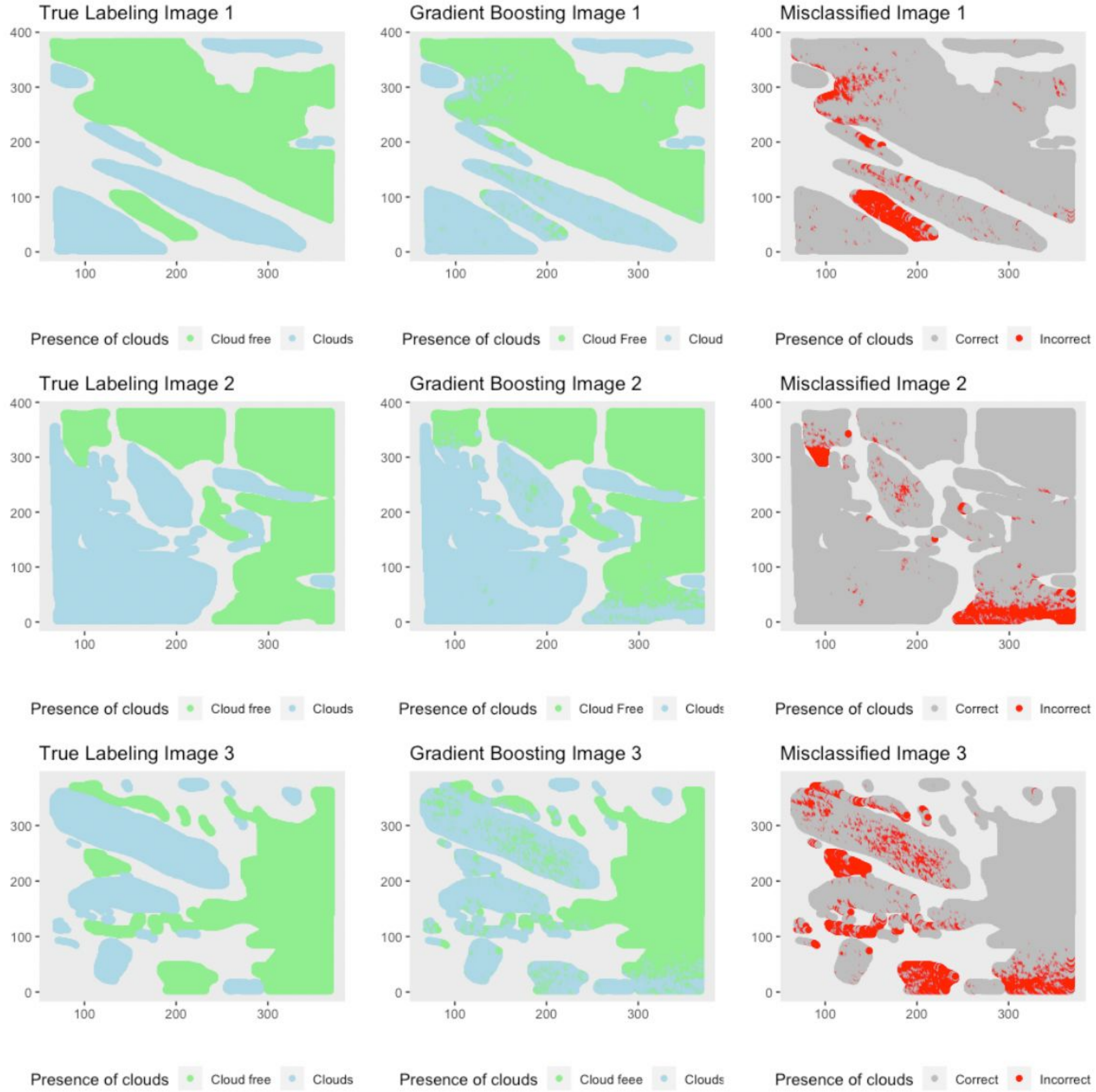


Figure 15

Modification to data splitting

It is definitely possible to imagine better classifiers, especially in absence of computational constraints. For instance, we may attempt to apply Chan-Vese algorithm to redefine the outlines of clouds and land, taking into account our assumption of smooth surface and cloud nature. Studying the misclassified patches, we also note that pixels in the middle of correctly identified surface and pixels in the middle of correctly identified clouds have largely contributed to the error of classification (Figure 16). We believe clouds over Arctic would remain smooth and rarely in small patches based on scientific research published on the subject. Therefore, using Chan-Vese algorithm would allow us to treat the Gradient Boosting results as a noisy image. The algorithm would identify borders and reduce misclassification. The algorithm results (simulated in Python) would look approximately like in figure 16. The entire area inside a Chan-Vese method generated frontier would be labeled according to the label of the majority of the pixels.

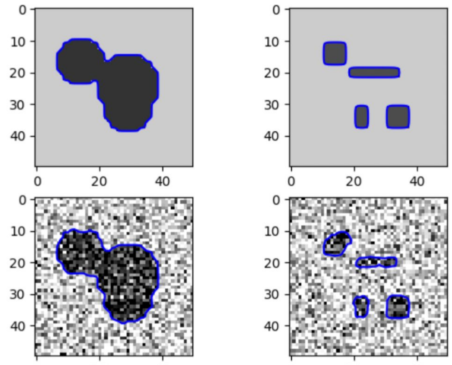


Figure 16

In addition, we recognize that tree-based methods may outperform other methods in samples but underperform in out-of-sample problems. Running a simulation of out-of-sample problem on the test data, however, we found the same accuracy for the Boosting model as we had on the training and validation sample: around 95%. However, GLM model proved almost as effective and could be considered instead.

Additional splits of data may be studied in order to better understand if the Gradient Boosting + Chan-Vese method would yield consistent improvements in accuracy. Namely, one may consider splitting the data in a way that is more conscious of the distribution of positive observations (cloudy pixels) in the sets. Since our data is highly imbalanced and Chan-Vese will have an opportunity to use CV to assess the fit of the model (figure 17).

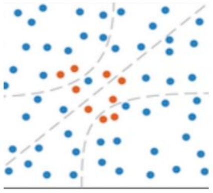


Figure 17

Conclusion

Employing the gradient boosting method we were able to achieve a considerably accurate classification with average accuracy on the test set of 91.7% and specificity of over 96%. In the figure 18 the final gradient boosting classification model predictions are displayed. We performed extensive diagnostic of our model and are confident in its predictive ability. In conclusion, while the ELCM–QDA algorithm presented in the original paper shows better performance compared to our preferred method, considering the time constraint of the project we did our best to make our own intellectual contribution to the work of statisticians on climate change.

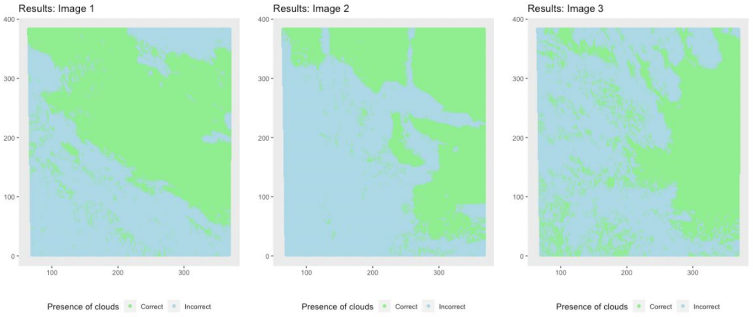
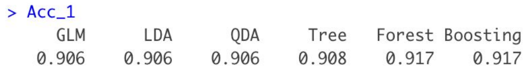


Figure 18

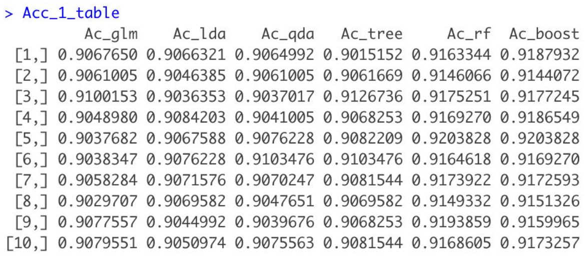
ACKNOWLEDGMENTS

Jen Shi created all the analysis and Alina Skripets wrote the R code. We have used various web resources such as Data Camp and R package manuals. We received no additional outside help or guidance.

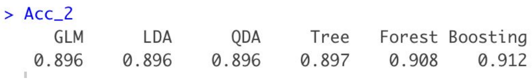
APPENDIX



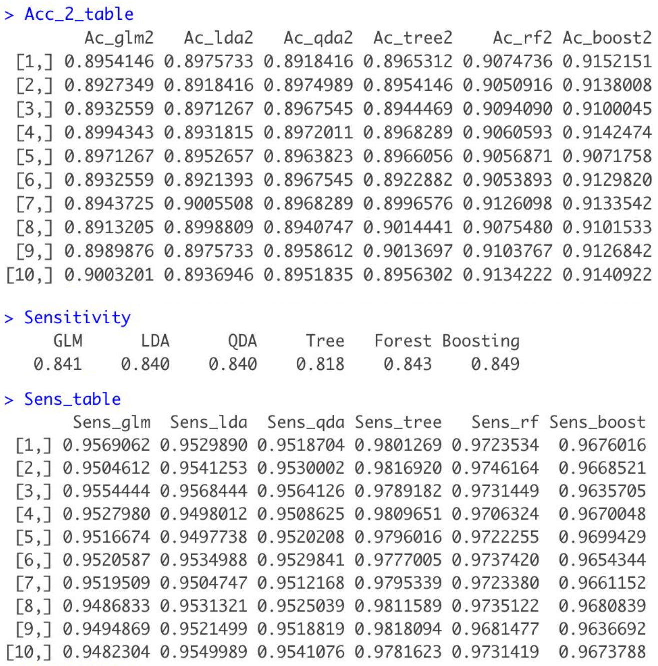
```
> Acc_1
     GLM      LDA      QDA     Tree   Forest Boosting
   0.906    0.906    0.906    0.908    0.917    0.917
```



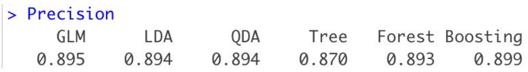
```
> Acc_1_table
        Ac_glm    Ac_lda    Ac_qda   Ac_tree     Ac_rf  Ac_boost
 [1,] 0.9067650 0.9066321 0.9064992 0.9015152 0.9163344 0.9187932
 [2,] 0.9061005 0.9046385 0.9061005 0.9061669 0.9146066 0.9144072
 [3,] 0.9100153 0.9036353 0.9037017 0.9126736 0.9175251 0.9177245
 [4,] 0.9048980 0.9084203 0.9041005 0.9068253 0.9169270 0.9186549
 [5,] 0.9037682 0.9067588 0.9076228 0.9082209 0.9203828 0.9203828
 [6,] 0.9038347 0.9076228 0.9103476 0.9103476 0.9164618 0.9169270
 [7,] 0.9058284 0.9071576 0.9070247 0.9081544 0.9173922 0.9172593
 [8,] 0.9029707 0.9069582 0.9047651 0.9069582 0.9149332 0.9151326
 [9,] 0.9077557 0.9044992 0.9039676 0.9068253 0.9193859 0.9159965
[10,] 0.9079551 0.9050974 0.9075563 0.9081544 0.9168605 0.9173257
```



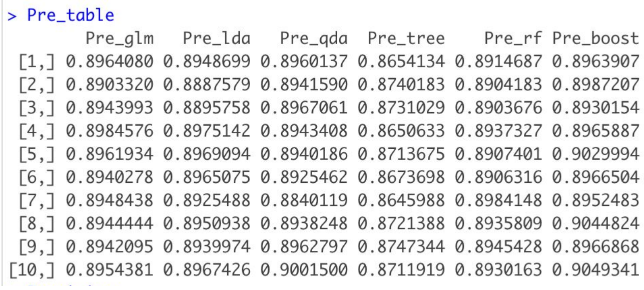
```
> Acc_2
     GLM      LDA      QDA     Tree   Forest Boosting
   0.896    0.896    0.896    0.897    0.908    0.912
```



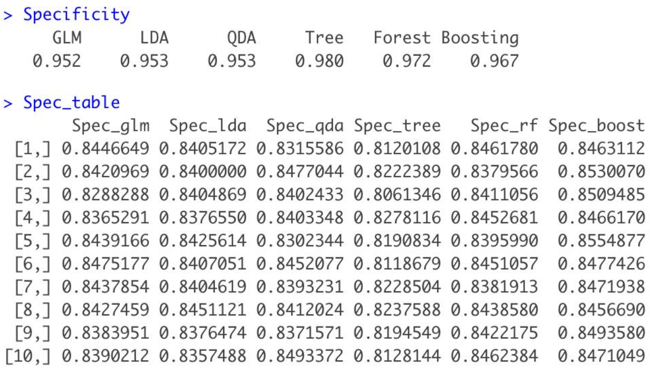
```
> Acc_2_table
        Ac_glm2   Ac_lda2   Ac_qda2  Ac_tree2    Ac_rf2 Ac_boost2
 [1,] 0.8954146 0.8975733 0.8918416 0.8965312 0.9074736 0.9152151
 [2,] 0.8927349 0.8918416 0.8974989 0.8954146 0.9050916 0.9138008
 [3,] 0.8932559 0.8971267 0.8967545 0.8944469 0.9094090 0.9100045
 [4,] 0.8994343 0.8931815 0.8972011 0.8968289 0.9060593 0.9142474
 [5,] 0.8971267 0.8952657 0.8963823 0.8966056 0.9056871 0.9071758
 [6,] 0.8932559 0.8921393 0.8967545 0.8922882 0.9053893 0.9129820
 [7,] 0.8943725 0.9005508 0.8968289 0.8996576 0.9126098 0.9133542
 [8,] 0.8913205 0.8998809 0.8940747 0.9014441 0.9075480 0.9101533
 [9,] 0.8989876 0.8975733 0.8958612 0.9013697 0.9103767 0.9126842
[10,] 0.9003201 0.8936946 0.8951835 0.8956302 0.9134222 0.9140922

> Sensitivity
     GLM      LDA      QDA     Tree   Forest Boosting
   0.841    0.840    0.840    0.818    0.843    0.849

> Sens_table
       Sens_glm  Sens_lda  Sens_qda Sens_tree   Sens_rf Sens_boost
 [1,] 0.9569062 0.9529890 0.9518704 0.9801269 0.9723534  0.9676016
 [2,] 0.9504612 0.9541253 0.9530002 0.9816920 0.9746164  0.9668521
 [3,] 0.9554444 0.9568444 0.9564126 0.9789182 0.9731449  0.9635705
 [4,] 0.9527980 0.9498012 0.9508625 0.9809651 0.9706324  0.9670048
 [5,] 0.9516674 0.9497738 0.9520208 0.9796016 0.9722255  0.9699429
 [6,] 0.9520587 0.9534988 0.9529841 0.9777005 0.9737420  0.9654344
 [7,] 0.9519509 0.9504747 0.9512168 0.9795339 0.9723380  0.9661152
 [8,] 0.9486833 0.9531321 0.9525039 0.9811589 0.9735122  0.9680839
 [9,] 0.9494869 0.9521499 0.9518819 0.9818094 0.9681477  0.9636692
[10,] 0.9482304 0.9549989 0.9541076 0.9781623 0.9731419  0.9673788
```



```
> Precision
     GLM      LDA      QDA     Tree   Forest Boosting
   0.895    0.894    0.894    0.870    0.893    0.899
```



```
> Pre_table
        Pre_glm   Pre_lda   Pre_qda  Pre_tree    Pre_rf Pre_boost
 [1,] 0.8964080 0.8948699 0.8960137 0.8654134 0.8914687 0.8963907
 [2,] 0.8903320 0.8887579 0.8941590 0.8740183 0.8904183 0.8987207
 [3,] 0.8943993 0.8895758 0.8967061 0.8731029 0.8903676 0.8930154
 [4,] 0.8984576 0.8975142 0.8943408 0.8650633 0.8937327 0.8965887
 [5,] 0.8961934 0.8969094 0.8940186 0.8713675 0.8907401 0.9029994
 [6,] 0.8940278 0.8965075 0.8925462 0.8673698 0.8906316 0.8966504
 [7,] 0.8948438 0.8925488 0.8840119 0.8645988 0.8984148 0.8952483
 [8,] 0.8944444 0.8950938 0.8938248 0.8721388 0.8935809 0.9044824
 [9,] 0.8942095 0.8939974 0.8962797 0.8747344 0.8945428 0.8966868
[10,] 0.8954381 0.8967426 0.9001500 0.8711919 0.8930163 0.9049341
```



```
> Specificity
     GLM      LDA      QDA     Tree   Forest Boosting
   0.952    0.953    0.953    0.980    0.972    0.967

> Spec_table
       Spec_glm  Spec_lda  Spec_qda Spec_tree   Spec_rf Spec_boost
 [1,] 0.8446649 0.8405172 0.8315586 0.8120108 0.8461780  0.8463112
 [2,] 0.8420969 0.8400000 0.8477044 0.8222389 0.8379566  0.8530070
 [3,] 0.8288288 0.8404869 0.8402433 0.8061346 0.8411056  0.8509485
 [4,] 0.8365291 0.8376550 0.8403348 0.8278116 0.8452681  0.8466170
 [5,] 0.8439166 0.8425614 0.8302344 0.8190834 0.8395990  0.8554877
 [6,] 0.8475177 0.8407051 0.8452077 0.8118679 0.8451057  0.8477426
 [7,] 0.8437854 0.8404619 0.8393231 0.8228504 0.8381913  0.8471938
 [8,] 0.8427459 0.8451121 0.8412024 0.8237588 0.8438580  0.8456690
 [9,] 0.8383951 0.8376474 0.8371571 0.8194549 0.8422175  0.8493580
[10,] 0.8390212 0.8357488 0.8493372 0.8128144 0.8462384  0.8471049
```