STAT 154 Project 2

Alice Meng & Yijun Long

1. Data Collection and Exploration

(a) Summary:

"Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data" is a study on operational cloud detection algorithms applied in the polar regions. Cloud detection in the Arctic is challenging because of the similar remote sensing characteristics of clouds and ice- and snow-covered surfaces. The goal of this study is to derive, train, and compare algorithms that can efficiently process massive imagery data without requiring human intervention. The data used in the study was collected by NASA's Multi-angle Imaging Spectroradiometer (MISR) imagery, which are electromagnetic radiation measurements using nine cameras at nine different angles, each of which views the Earth in four spectral bands (blue, green, red, and near-infrared). Each MISR pixel encompasses a 275m*275m region, yielding tremendous amounts of data. The nine view zenith angles of the cameras are 70.5" (Df), 60.0" (Cf), 45.6" (Bf), and 26.1" (Af) in the forward direction; 0.0" (An) in the nadir direction and 26.1" (Aa), 45.6" (Ba), 60.0" (Ca), and 70.5" (Da) in the aft direction. (The "f" in the letter designation of the cameras represents the "forward" direction, and the "a" represents the "aft" direction.) Besides the features provided by the nine angles, after substantial exploratory data analysis, three other physically useful features are applied in the analysis, which are CORR, an average linear correlation of radiation measurements at different view angles, SD, the standard deviation of MISR nadir camera pixel values across a scene, and NDAI, the ratio between the difference and sum of the mean radiation measurements from the first and fifth angle associated with a particular pixel region. To evaluate the performance of proposed methods, an expert hand-labeled the data to either high confidence cloudy, high confidence clear, or unlabeled. Thus, the expert label variable is included in the dataset used to better train the model. In the end of the paper, the researchers concluded that CORR, SD, and NDAI contains sufficient information to separate clouds from ice- and snow-covered surfaces. Moreover, the ELCM algorithm, which combines classification and clustering frameworks, provides better spatial coverage and is more computationally efficient. The work shown in the paper is significant because, when tackling data-related problems, it is very important for the researchers to choose appropriate statistical methods to the data on a case-by-case basis. In addition, the study demonstrates the power of statistical thinking, and the ability of statistics to contribute solutions to modern scientific problems.

(b) Summarize data & Maps

Summarizing the data, we calculate the percentage of pixels for the different classes among the three image data sets: 36.8% as no loud, 39.8% as unlabeled, and 23.4% as cloud. Then we plot well-labeled maps using x, y coordinates and the expert labels as colors of the region (Figure 1). From the three maps we plot, we can observe a similar pattern. The reason lies in the fact that they are three images of the same location, and so the samples are assumed as non identically independent distributed(iid).

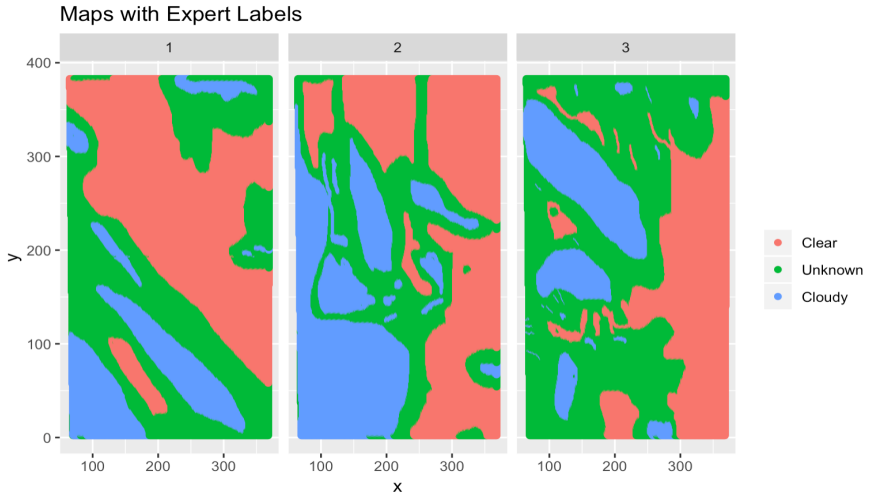


Figure 1: Maps with expert labels as colors of the region

(c) EDA

We explore the data by performing (i) pairwise relationship between the features and (ii) relationship between the expert labels with the individual features. The plots are shown above in Figure 2,3,4,5. In this section, we will provide our findings from these plots, respectively.

Figure 2

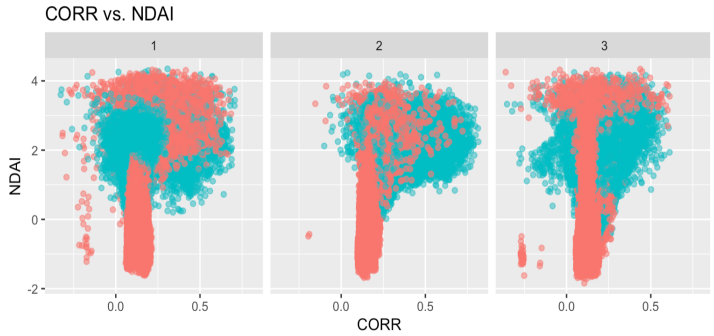


Figure 3

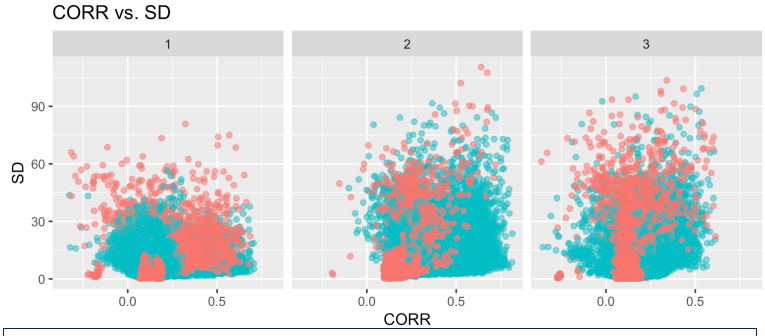


Figure 4

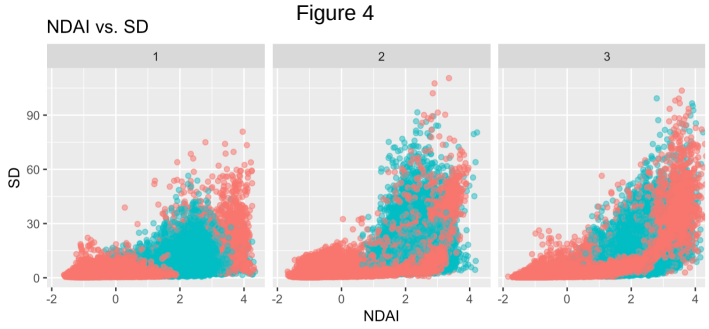


Figure 5

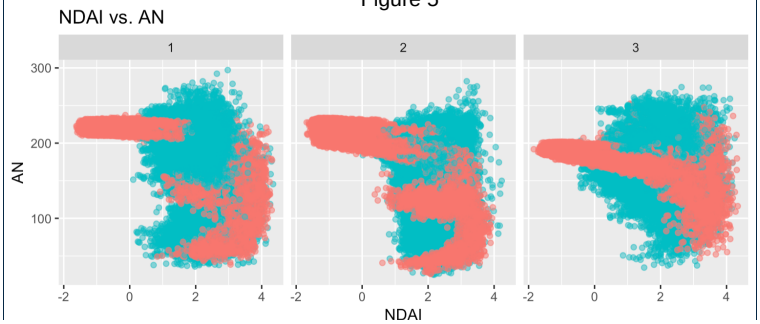


Figure 6

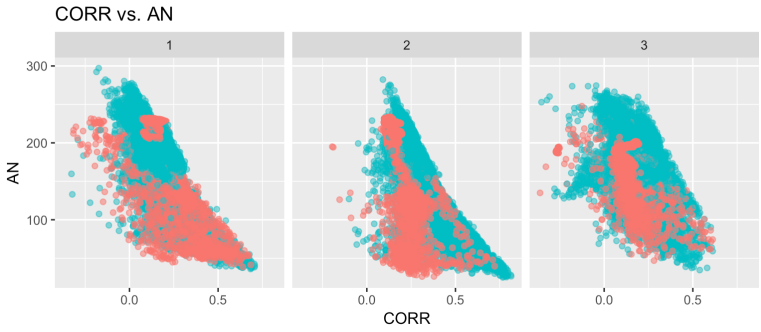
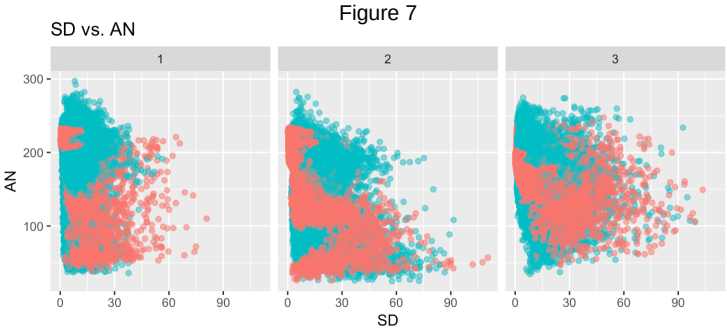


Figure 7



Since we only care about the relationship between the two classes(cloud, no cloud) and the features, for now we remove the the unlabeled observations. From the pairwise plots between the variables NDAI vs. CORR, CORR vs. SD, SD vs. NDAI, we see patterns, which indicate whether there are cloud with specific shapes. We also plotted NDAI vs the radiance variables (e.g. AN) and noticed that there are merely cloud trends that are not seperatable.

Then we plotted the conditional densities of the variables and we found that they all somehow help predict if there are clouds. We ca conclude that higher NDAI indicates higher likelihood of the precence of cloud. It is clear that pixels labelled as clouds have higher SD values Also, higher CORR suggests higher likelihood of the precence of cloud, though not as strongly as NDAI. From the density plots of the radianc features, we see similar distributions with different expert labels for different radiance. For instance, when there is no cloud(expert label == -1) the distributions of AN have peaks around 200. Those conditional densities plots are shown below as Figure 8,9,10,11,12,13,14,15.

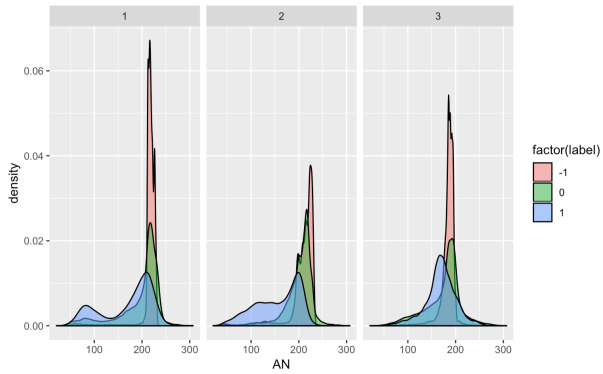


Figure 8

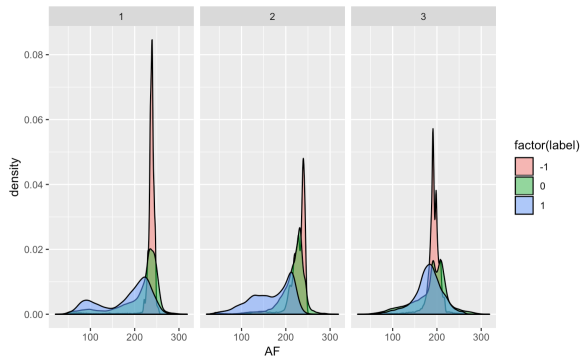


Figure 9

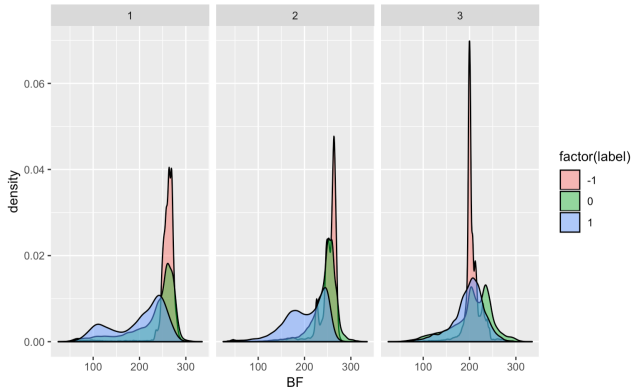


Figure 10

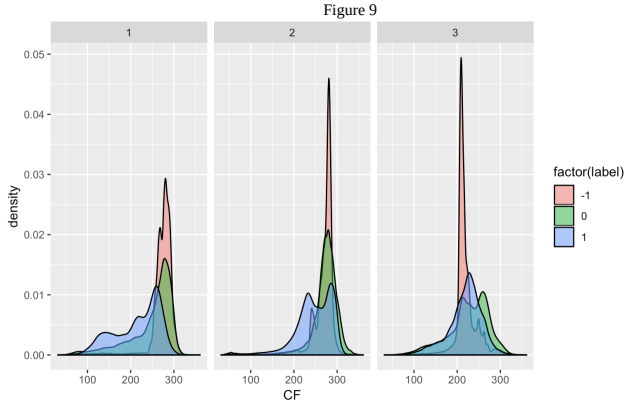


Figure 11

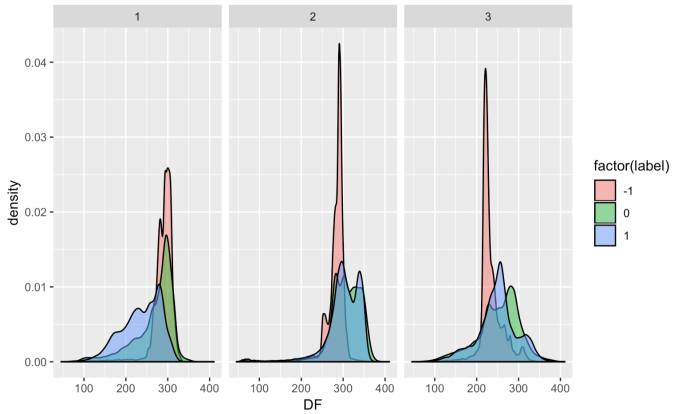


Figure 12

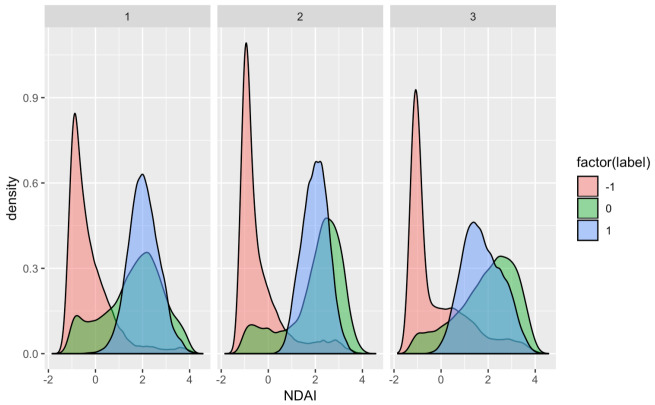


Figure 13

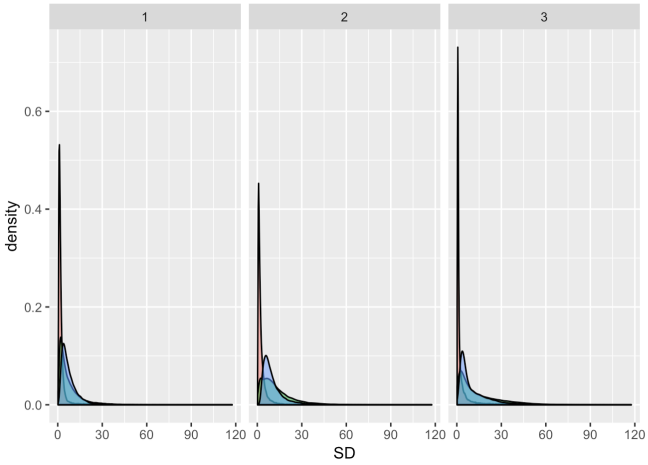


Figure 14

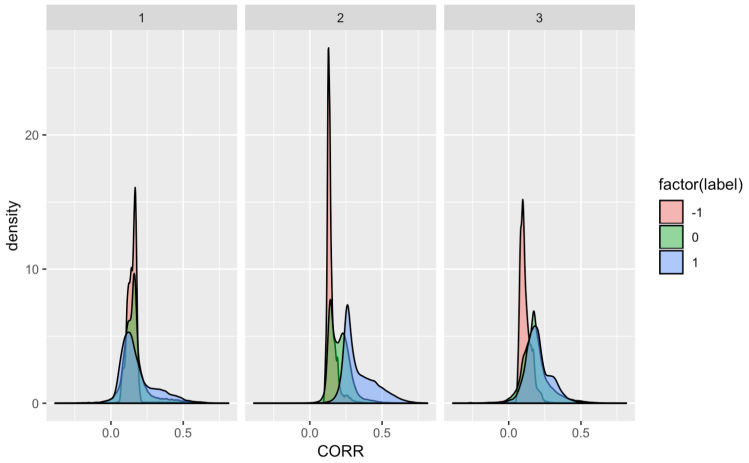


Figure 15

2. Preparations

(a) For our data, the rows are not i.i.d. As such we have to be more careful in choosing the train set, test set, and validation set. We only hae three images, one way to create more observations is to divide each image into k by k smaller images. Doing this, each block can be thought of as a separate image, and we have 3k^2 images. These newly created images are not totally independent; Still, dividing three images into small images should help us in building a more stable model on new images. In our data, we choose k=3, as such there are 27 small images. We choose 15 blocks at random to use as train, and the remaining as test. By splitting this way, we take into account that the data is not iid, unlike random split, which does not take into account of non-iid data.

Another splitting method is to split each image into 2 by 2 smaller images by the mean of x and y. Doing this, each block can be thought of as a separate image, and we have 12 images. Then we choose 6 blocks at random to use as train, 3 blocks as validation set, the remaining 3 as test set. This splitting method takes into account the distributions of x and y and the fact that the data is not iid.

(b) The accuracy of a trivial classifier which sets all labels to -1 on the validation set and on the test set is approximately 36.8%, which serves as a baseline to ensure that the classification problem is not trivial.

(c) Based on the conditional densities and the pairwise correlation plots, we selected CORR, NDAI, and SD as the three predictors because they predict the presence of clouds better than the radiances of angles. The conditional densities generally exhibit less overlap in the two distributions than the conditional densities of the radiances. We see that NDAI has fairly better separation between cloudy and clear in all three plots, which is confirmed in our modeling sections later. Also, it is clear that cloudy pixels have higher SD values. Hence, we expect the two features together can help determine whether a pixel with high NDAI should be labelled as cloudy by using the SD feature. Finally, CORR values seem to be a good separator for image 2, but less so for image 1. The different distributions of CORR values for these images suggest us to cross validate our models across images.

(d) you can find the CVgeneric function in github as a separate R file.

3. Modeling

In order to perform AUC function, we need to make variable label binary. That is, we need to remove the unlabeled observations first.

(a) Assumptions: Some of the models are more of an optimization procedures than a statistical model, namely Support Vector Machine which really has no assumption. We instead check the model assumption for the probabilistic models. For Linear Regression, it is clear that the assumption will not be met for binary reponses. However, later we will see a Least Square Method can still perform very well.

Next, we check the model assumption for Logistic Regression. The log-odd should be linear in each of the inputs. We plot the relationship in Figure 16 below.

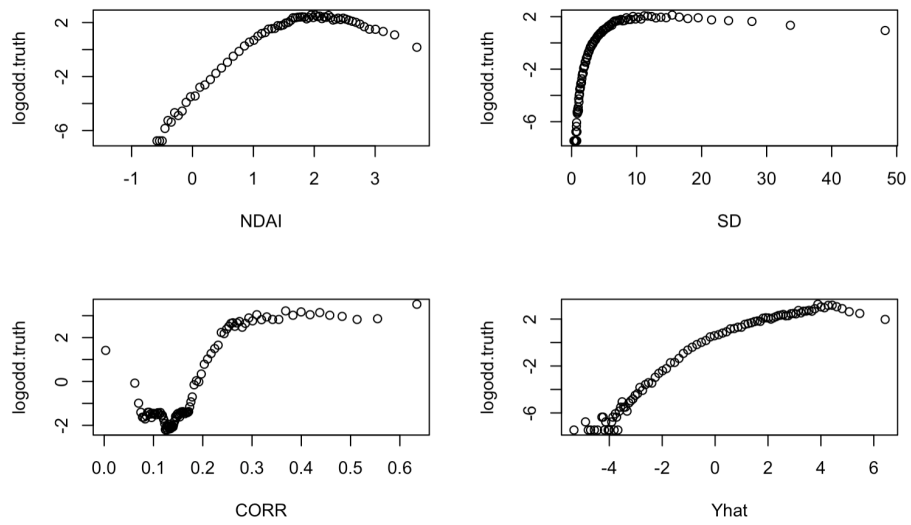


Figure 16: log-odd vs. predictors

Looking at the plots of log-odd versus each predictor, we see that the plot of log-odd with respect to NDAI, SD, and CORR are not linear. So we might want to include quadratic terms. This also explains why we see a higher performance in QDA.

For QDA, we need the predictor variables X to be drawn from a multivariate Gaussion distribution. Looking at the marginal Q-Q plot with respect to the normal quantiles, we see that none of the inputs have a linear Q-Q plot. So again, the assumptions are not met.(Figure 17)

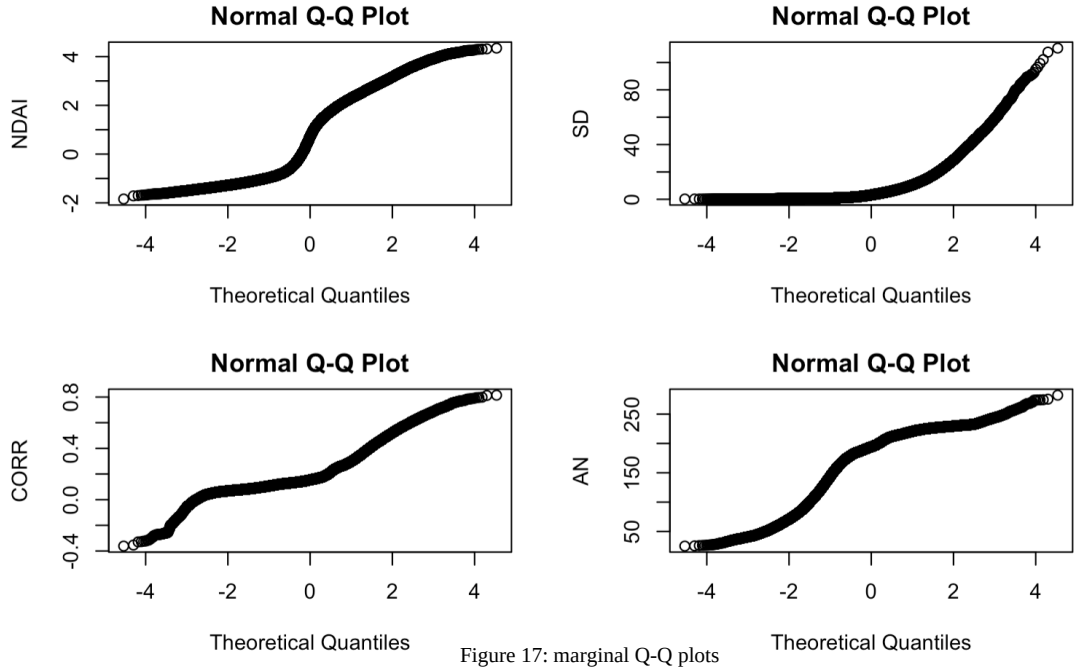


Figure 17: marginal Q-Q plots

1. Linear Regression

We perform linear regression for the training-validation data and set data splitted by two different ways, as described in 2(a). For the first method of splitting(27 folds), the AUC is 0.915 and the test accuracy is 0.867. For the second method, the AUC is 0.933 and the test accuracy is 0.763. The figure 18 shown below is the ROC curve for the first splitting way.

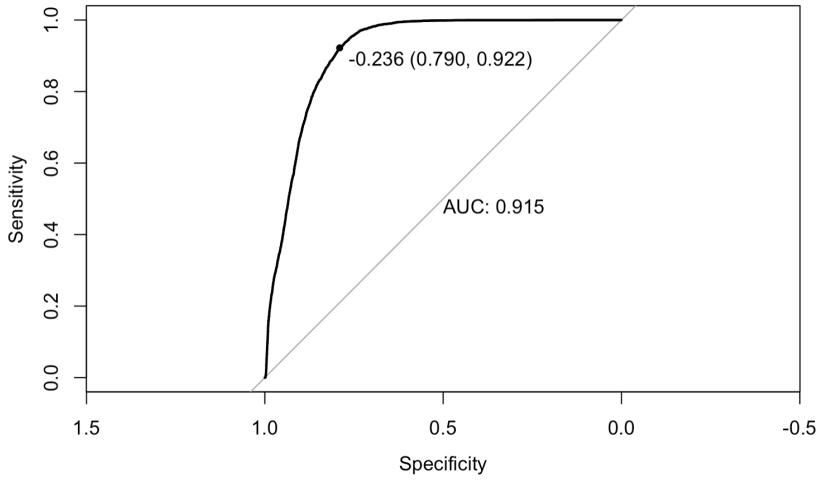


Figure 18: ROC curve for linear regression

Besides the test accuracy, we also have the 10-fold CV accuracies across folds. For splitting method 1, the CV accuracies are 0.6930476, 0.6864236, 0.6973101, 0.6976557, 0.6890732, 0.6952940, 0.6907436, 0.6940092, 0.7015725, 0.6963309. These values are almost all clustering around 0.70. For splitting method 2, the CV accuracies are given by 0.6857725, 0.6776228, 0.6823971, 0.6807210, 0.6776638, 0.6878105, 0.6835435, 0.6865168, 0.6890842, 0.6824204, which are very close to, but a little lower than the first method.

2. Logistic Regression

We perform logistic regression for the training-validation data and set data splitted by two different ways. For the first method, the AUC is 0.913 and the test accuracy is 0.858. For the second splitting, the AUC is 0.9335 and the test accuracy is 0.8839.

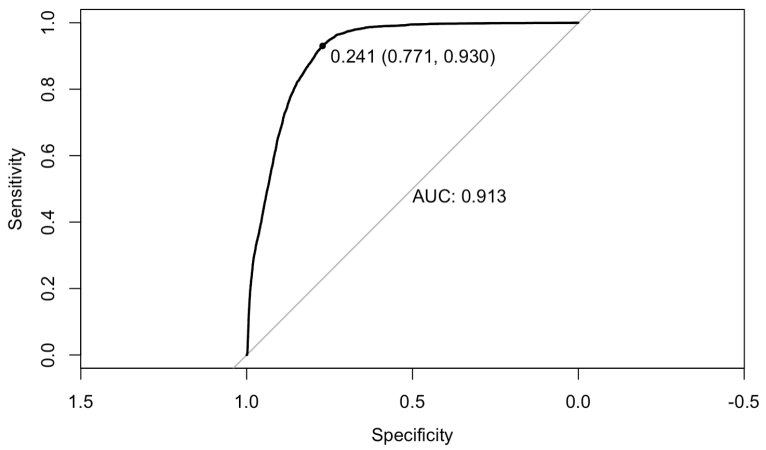


Figure 19: ROC curve for logistic regression

Besides the test accuracy, we also have the 10-fold CV accuracies across folds. For splitting method 1, the CV accuracies are 0.9033466, 0.9023098, 0.9043834, 0.9062266, 0.9052474, 0.9050170, 0.9061114, 0.9033986, 0.9052474, 0.9073786, which are all around 0.90. For splitting method 2, the CV accuracies are 0.8949815, 0.8908211, 0.8901414, 0.8898796, 0.8898159, 0.8908419, 0.8924341, 0.8913445, 0.8995669, 0.8923567, which are on average lower than the first splitting way.

3. Generalized Linear Model

When fitting a generalized linear model, we tried different values of lamba in the model, and got different values of AUCs and accuracies for different values of lambda. The maximum of AUC achieved reaches 0.913 and the maximum accuracy is 0.8572. Using the corresponding lambda value for maximum AUC, we have the ROC curve as below.

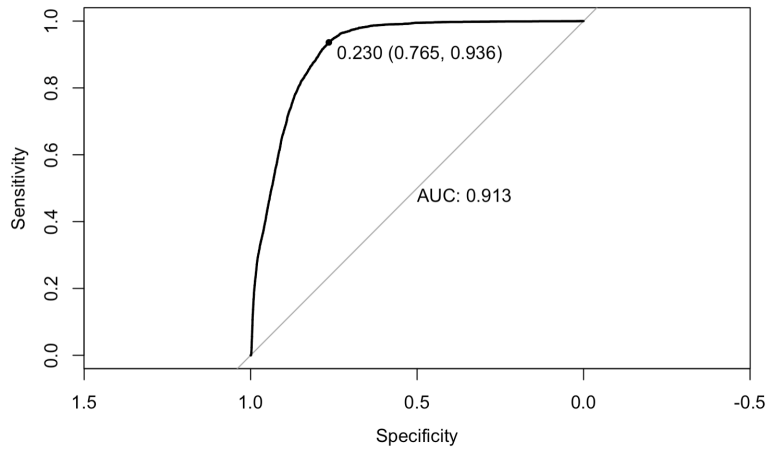


Figure 20: ROC curve for GLM

The CV accuracy averaged over all folds is reported as 0.8607142.

For splitting method 2, the maximum AUC achieved is 0.9335383 which is higher than the first way. And accuracy as 0.8839546.

4. Quadratic Discriminative Analysis

For the first splitting method, the AUC is 0.8726 and the test accuracy is 0.9049732. For the second method, the AUC is 0.8732 and the test accuracy is 0.8758448.

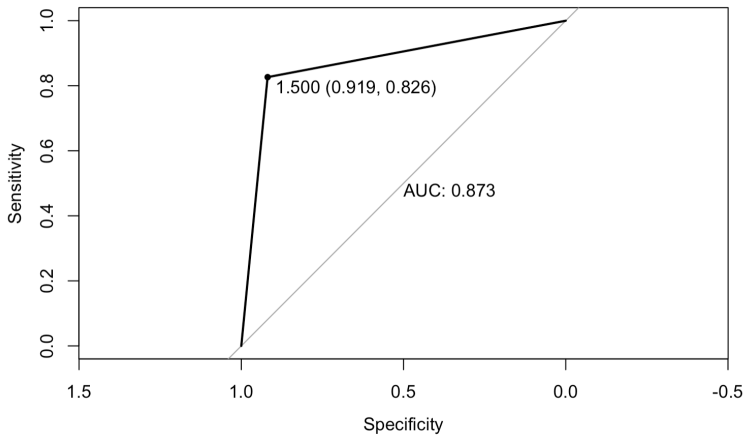


Figure 21: ROC curve for QDA

Using the CVgenerics function we wrote in 2(d), we got the CV accuracies across folds: 0.8946489, 0.8939001, 0.8906169, 0.8930361, 0.8926329, 0.8919417, 0.8926905, 0.8919355, 0.8920569, 0.8949369.

5. Random Forest

For the first splitting method, the AUC is 0.9749 and the test accuracy is 0.9031094. For the second splitting method, the AUC is 0.9558 and the test accuracy is 0.9104978.

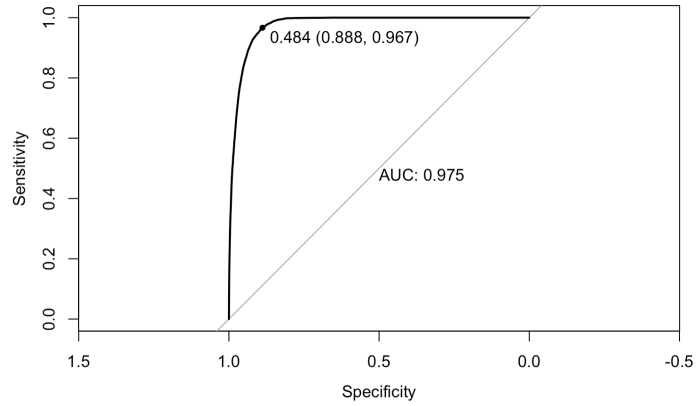


Figure 22: ROC curve for Random Forest

ROC Curve Cutoff Value

When plotting ROC curves, we choose cutoff values for each method according to the maximum value of the sume of sensitivity and specificity. We choose this point as cutoff value because we want both sensitivity and specificity to be equally weighted when evaluating the ROC curve, rather than weighing one more over the other. By choosing the cutoff point, we can better compare the performance of different methods.

Commentary about the models

In total we used 5 different classification methods or models for 2 ways of splitting the data: Linear Regression, Logistic Regression, Generalized Linear Model, Quadratic Discriminant Analysis, and Random Forest. And we see that for the first way of splitting data, random forest has the highest test accuracy of 91% followed by logistic regression of 88% and QDA of 87%. Linear regression has the lowest test accuracy of 80%, probably due to its misassumptions. For the second way of creating folds, random forest still performs best with an 88% test accuracy followed by QDA of 76% and logistic regression of 71%, while linear regression has the lowest value of 65%.

We also did 10-folds cross validation on the data for both the ways of creating folds, it is clear that random forest gives the highest accuracies across folds whereas linear regression has the lowest accuracies across folds. The other CV-results are shown in the boxplots we created for these different models in Figure23.

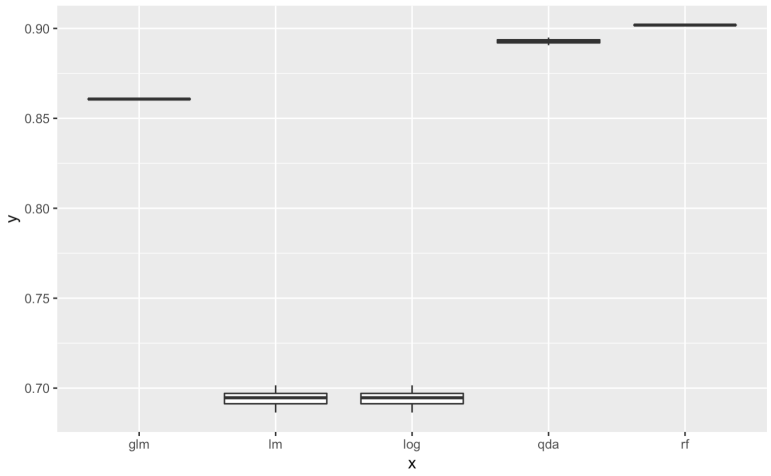


Figure 23: boxplots of CV-results for different models

4. Diagnostics

(a) We know that for a random forest model, when the out-of-bangs error stabilizes (i.e. when our solution converges) as more tress are being trained, the training can be stopped before actually training all the trees. From the plots for convergence analysis(Figure 24, 25, 26, 27) we see that the accuracy stays around 88%, which is pretty high. We conclude that the model is quite stable with respect to addming more trees into the training data.

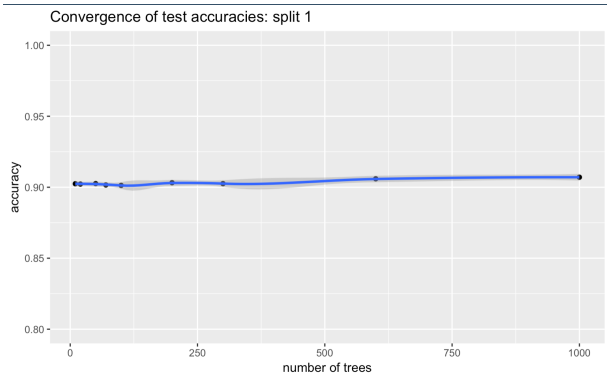


Figure 24

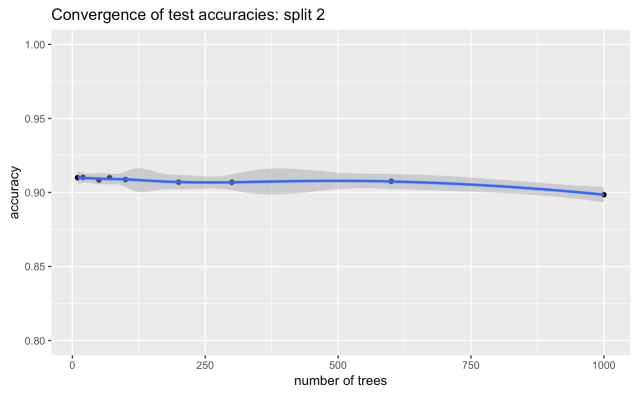


Figure 25

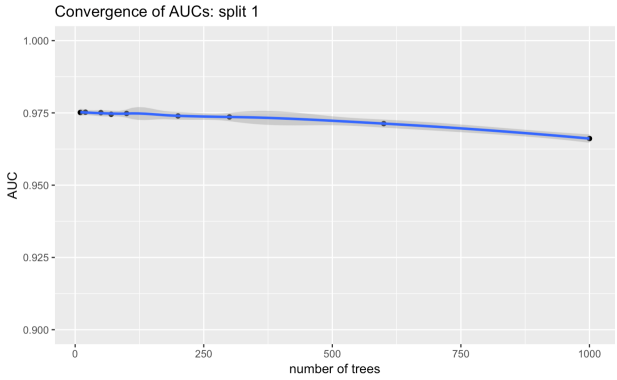


Figure 26

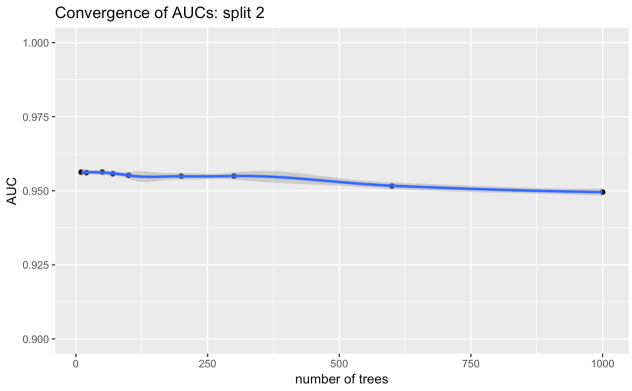


Figure 27

(b) The images below show misclassification errors for model trained on image 1 and 2, and tested on image 3. The other plot show misclassification errors for model trained on image 2 and 3, and tested on image 1(Figure 28,29). We see that this model did not perform as well as the k-fold trained one but plotting their misclassification errors helps us see where the model failed. For image 1, false positives are clustered at (x, y) around (150,50) to (200,100). For image 3, false negatives are centered at (x,y) around (50, 200) to (250,300). It is clear that the false negative rate is higher than the false positive rate, which indicates that the model misclassifies more absence of clouds.

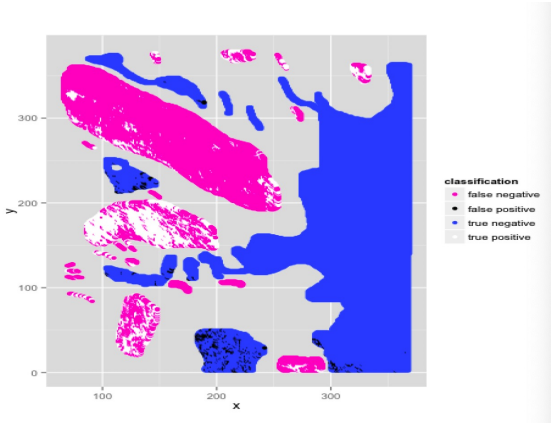


Figure 28: prediction error for image 3-trained on image 1 and 2

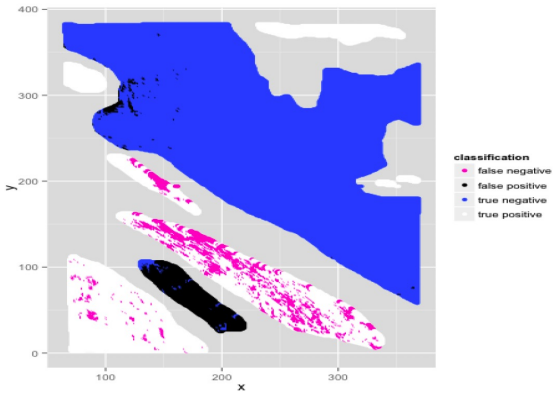


Figure 29: prediction error for image 1-trained on image 2 and 3

(c) Based on 4(a) and 4(b), we want to find a better classifier. We have ran our random forest model using all the features in previous parts. To gather a quantitative measurement of the importance of each feature, we looked at the Gini Importance measure, which is calculated by recording the difference between the Gini measure of a random forest's predictions on a fold and the Gini measure with a particular feature's values randomly shuffled. From Figure 30, it is clear that features with higher mean-decreased Gini values are more important to the model. If a feature were crucial to the forest's trees, then randomly shuffling that feature will drastically decrease its Gini measure. As we can see, NDAI, SD and CORR consistently ranked as the top three in all cross validations. And also AN is the most important radiance feature that we should include in our model.

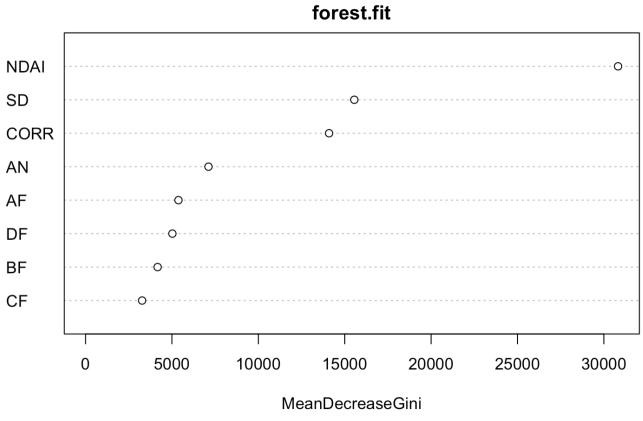


Figure 30: Features' Mean Decreased Gini

For future data without expert labels, our model will give a pretty accurate prediction on the presence of clouds and a relatively less accurate prediction on the absence of clouds. As depicted in the graphs in Figure 28 and 29, we have many more false negatives than false positives with large regions sometimes completely misclassified. But in general, the random forest model with these features will do a great job at predicting clouds.

(d) As we modify the way of splitting the data, the result of the convergence analysis for our random forest model with four features does not change. The plot suggests that the model is still quite stable with the accuracies convergent. Split 1 has an AUC value of 0.968 and an accuracy of 90.5%, whereas split 2 has an AUC value of 0.955 and an accuracy of 90.9%. For all the models we have used, the first way of spitting the data gives a higher accuracy and AUC value. However, for our best random forest model with four features, the results for misclassification errors do not change much. The patterns are pretty similar with more false negatives than false positives. We conclude that our model is better in terms of time saving with more representative features.

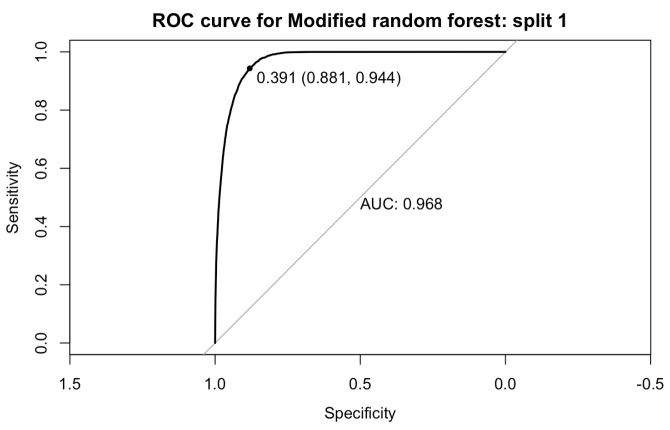


Figure 31

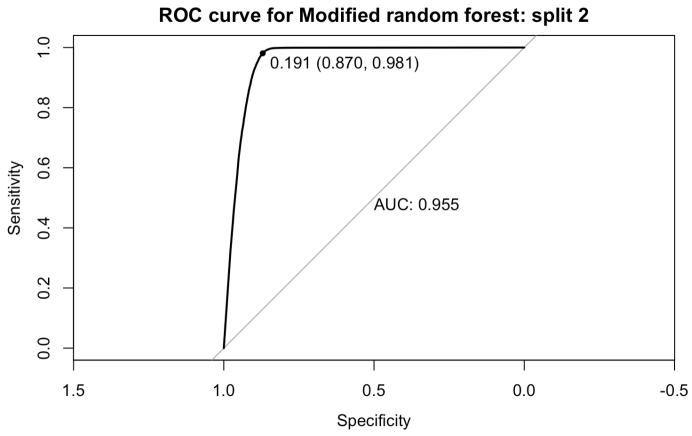


Figure 32

(e) Conclusion:

Linear regression, Logitic regression, Genralized Linear model, QDA and random forest all created reasonable predictive model with pretty high AUC values. On average, random forest had the best performance based on the accuracies and the AUC values across the folds. However, we are not sure whether the models are failing for similar resons. Logistic regression and QDA/LDA have parametric assumptions on the distribution of the classifications, but random forest only assumes no heavy tails. In the absence of significant domain knowledge, coupled with the better ROC performance, we conclude that random forest model gives a better generalization of the data. It also helped confirm the significance of the derived features NDAI, CORR and SD.

Our repository is at https://github.com/wmeng2019/STAT154