Climbing the Trees to See the Clouds: Using Random Forests to Identify Clouds in Satellite Images

Team Cumulonimbus
Members: Alison Reynolds (alison.reynolds@duke.edu) & Jackie Du (jacquelyn.du@duke.edu)

Section 1: Data Collection and Exploration

Part A

The paper by Yu et al. describes a novel algorithm to determine cloudiness versus non-cloudiness in satellite photographs taken in the Arctic region. This approach aims to detect non-cloudiness in pictures by using three new calculated features - correlation of MISR images of the same scene, the standard deviation of pixel values across a given image, and a normalized difference angular index (NDAI) which enumerates the variation in images of a given scene due to changes in angles. These features are then used to build an enhanced linear correlation matching (ELCM) algorithm that classifies pixels as clear or cloudy, which are then used to build a Bayesian probabilistic model of cloudiness using QDA for the areas that are ambiguously cloudy. Altogether, the results from these two steps detect cloud presence for each image.

This study uses pictures taken by the Multiangle Imaging SpectroRadiometer (MISR) which has nine cameras that capture a different angle in four spectra bands. The satellite travels the polar regions in 233 geographically distinct paths over a 16 day cycle. Within each path, there are 180 blocks numbered sequentially starting from north to south, and each MISR pixel covers 275 square meters of Earth. This particular study uses data from 10 MISR orbits of path 26, which contains a wide variety of surface features. Additionally, only the data from the red radiation measurements were used for each pixel as images from this wavelength had the highest resolution and there was little variation in data across the different radiation levels.

Compared to previous algorithmic methods, this ELCM algorithm has a 91.80% agreement rate with the expert labels, which is 8-12% higher than other algorithms. The significance of this new method is that it guarantees total coverage of the image by including a probabilistic model of partly cloudy regions. Moreover, this method provides increased accuracy in cloud detection because by identifying if a particular pixel in a picture is clear (rather than cloudy), it circumvents the variation in clouds (especially low cloud coverage in the polar regions) that have proven to be obstacles for prior methods. Overall, improving the performance of cloud detection will aid in climate change studies, as cloud coverage is related to increasing surface air temperatures and amounts of atmospheric carbon dioxide.

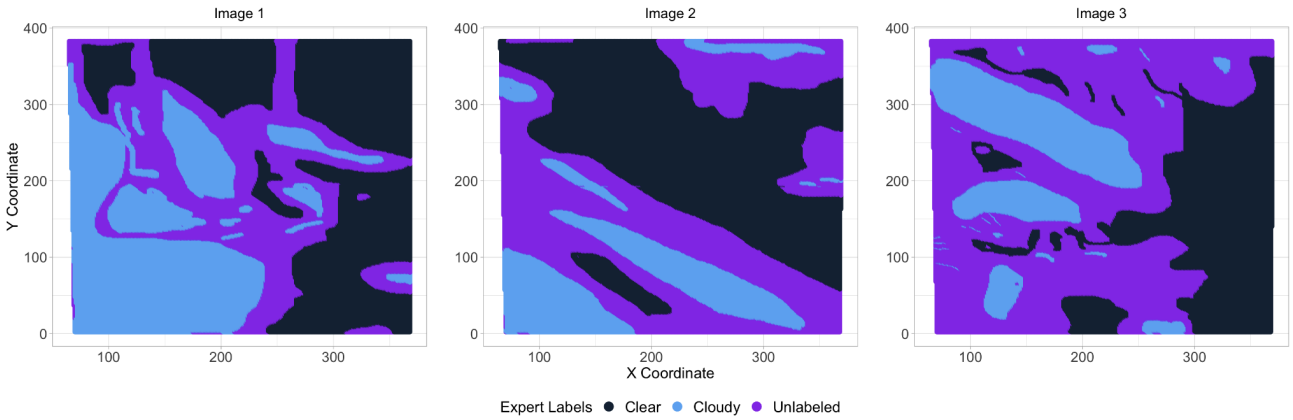
Part B

Fig. 1 *- Summary Table of Class Labels*

Type	Overall	Image 1	Image 2	Image 3
Clear	37%	37%	44%	29%
Cloudy	23%	34%	18%	18%
Unlabeled	40%	29%	38%	52%

The proportion of cloudy and not cloudy across each image highly varies. For cloudy, this varies from 29% to 44%, and for non-cloudy, from 18% to 34%. Even for the same amount of non-cloudy labels in an image, there are differing percentages of cloudy, so there does not seem to be a high correlation between these two percentages. The unlabeled class occurs on average for about 40%. The unlabeled proportion varies from 29% to 52% across the three images.

Fig. 2 *- Spatial representation of labels for each image*



From the plots of each of the images, we see that there are distinct regions that have the same label, either cloudy or clear. This makes sense as clouds tend to form masses that span over a large area of pixels, so there exists a stickiness to the distribution of pixels labeled as cloudy. Therefore, we cannot assume that pixels in this dataset are distributed independently since knowing that one pixel is cloudy would suggest that nearby pixels have a relatively higher likelihood of cloudiness.

Part C

Pairwise Relationships

Looking first at the relationship between the five different angles using Fig. 3 on the next page, the correlations between them range from 0.548 to 0.971. As expected, the angles in the forward direction that are closer next to each other tend to be more correlated, and those that are farther away tend to be less correlated. For the one angle in the aft direction, it seems to be most aligned with angle AF and incrementally less correlated with angle BF to DF.

For the calculated features, NDAI and SD are the only two calculated features that are strongly correlated with each other at 0.631. In terms of the angles' relationship with the calculated features, angle AF and angle AN tend to be the most correlated with each of the features; each of the three features tend to be less correlated as the angle moves towards the DF direction. Among the three calculated predictors, only CORR has a correlation above 0.5 for angle AF and AN - this suggests that there are either more cloud-free conditions or low altitude clouds as the angle increases for these two viewing directions. Otherwise, SD and NDAI only appear to be weakly correlated to the angles.

Besides simply looking at correlation, it appears that the variance of points increases as NDAI increases. Conversely, variance of other predictors slightly decreases as SD increases. This relationship also holds between CORR and the two angles in which it is strongly correlated with. The correlation of these features is discussed more in the assumptions section of Section 3.

Relationships with Expert Labels

Finally, looking at the labels vs features -- for the calculated features (i.e. NDAI, SD, CORR), having a cloudy label tends to have a higher right distribution. For NDAI and CORR, the distribution is bimodal, and the values with the cloudy label covers only and almost all of the right peak.

Fig. 3 *- Correlation matrix for all predictors*

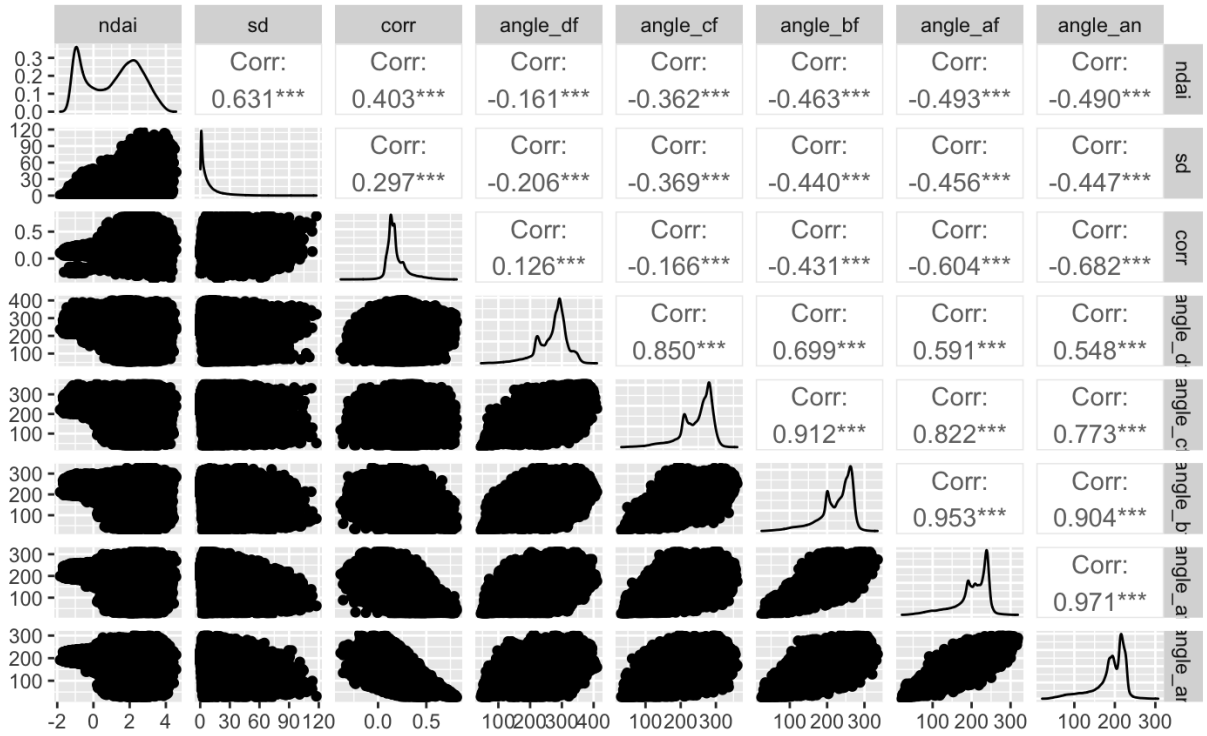
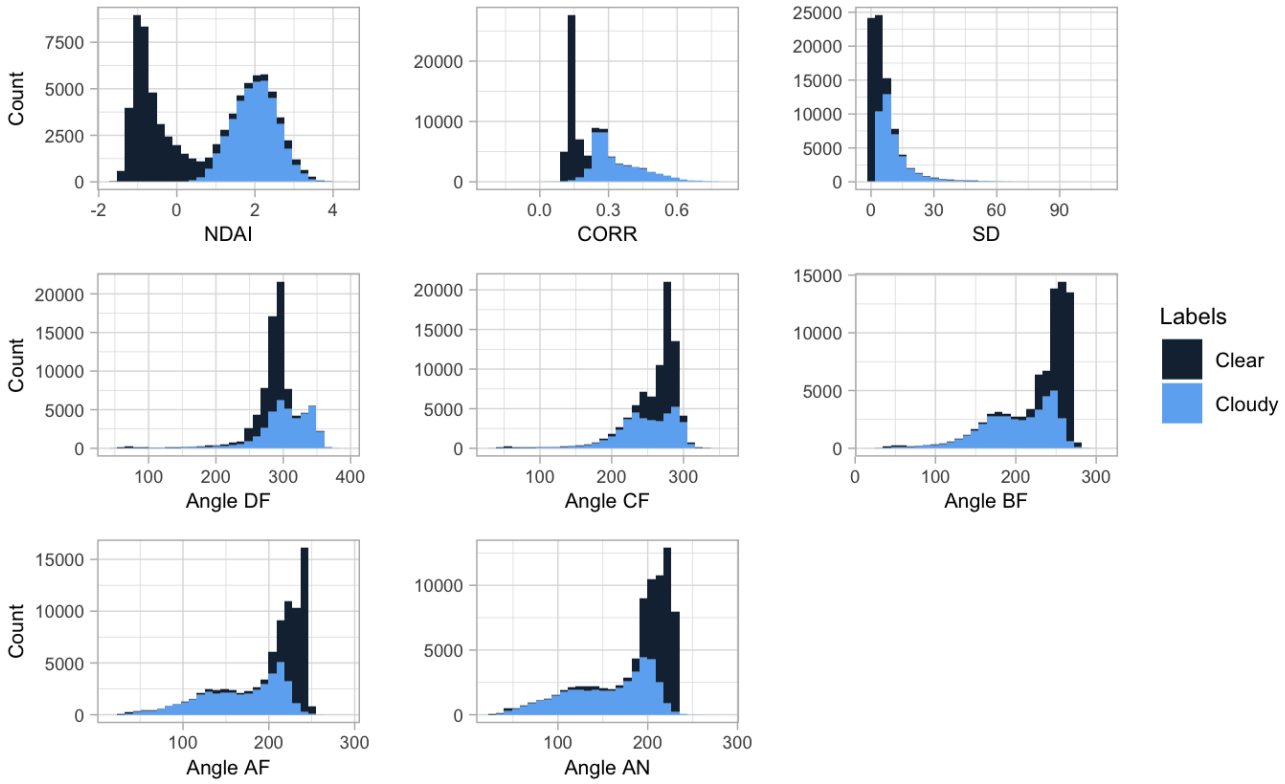


Fig. 4 *- Histograms of predictors grouped by expert labels*



The distribution for SD peaks at zero with a high right tail; the non-cloudy label has a very high frequency at zero whereas the cloudy label is distributed much more to the right. With respect to the angles, only for angle DF is the cloudy label distributed more to the right, while for the other angles the cloudy label is distributed to the left. For these four other angles, the non-cloudy label has a much more pronounced peak whereas the distribution of the cloudy label tends to have a relatively more spread out distribution. Examining these distributions informs our feature selection in Section 2C.

Section 2: Preparation

Part A

Splitting the data into training, validation, and testing sets is a more complicated task when looking at pixels of image data because assumptions on independence are no longer appropriate. It is not accurate to treat the pixel data as independent and identically distributed because the clouds take up many adjacent pixels, so knowing that one pixel has a cloud in it provides some information on the cloudiness of the surrounding pixels. This paper takes two different approaches to split the data while preserving some of the spatial relationships: (1) treating each of the three images as one split, and (2) breaking each image into blocks and dividing those into each set.

In the first method, one image is treated as the training set, one image is treated as the validation set, and one image is treated as the test set. This process is done randomly by shuffling the images and then picking out the train, validation, and test image. Splitting by this method preserves the spatial dependency in the entire image; however, it loses some interpretability if there are major differences between images. For instance, if one image has different color saturation or lighting, those values may be taken into account by the model which will impact the predictions on the other images. This method will be referred to as the image split for the remainder of the analysis.

In the second method, each image is broken up into blocks by splitting the range of the x and y axes into subsections and dividing the image space by those intervals. Then a fraction of the blocks from each image is placed in the training set, a fraction is placed into the validation set, and the rest are placed in the test set. This method partially preserves the spatial dependency of the pixels; the pixels closest to the center of the block are in the same set as their neighbors, but the pixels on the boundaries lose some of their adjacent pixels. However, this method ensures that each set has pixels from every image which makes the overall model more robust. This method also allows more control over the size of each setting, which means that the training set can be larger than the validation and testing sets giving the models more data to fit on. Additionally, the number of total blocks can be changed, balancing the desire for larger blocks to preserve more spatial dependency with the desire for more blocks to ensure that each set has a mixture of cloudy and cloud-free data. This method will be referred to as the block split for the remainder of the analysis.

Part B

The first model run was a trivial classifier, which set all labels to cloud-free to provide a baseline accuracy level for the subsequent models. For the image split, the validation accuracy was 71.13% and the test accuracy was 52.20%. For the block split, the validation accuracy was 84.24% and the test accuracy was 62.84%. This model will have high accuracy when the data is mostly cloud-free, and it's important to note that for both splits the test data has a lower cloud-free proportion than the validation data. This will

impact the performance of the models because the validation data will not be as accurate a proxy for the test set.

Part C

The best features for this model are the variables that are highly correlated with the expert labels and have clear splits between the cloudy and cloud-free data. NDAI and CORR are quantitatively strong predictors based on their correlation with the expert labels; their correlation values are 0.617 and 0.444 which are all much higher than the correlation with the other features. This will be most applicable for linear models. Additionally, SD is also selected as a best feature because visually the range of SD values between cloudy and cloud-free data is relatively more separable than the other predictors. This will be useful particularly in tree based models.

Predictor	Correlation
NDAI	0.6169
SD	0.2954
CORR	0.4441
Angle DF	0.0066
Angle CF	-0.2083
Angle BF	-0.3379
Angle AF	-0.3897
Angle AN	-0.3894

Fig. 5 - *Correlations between predictors and expert label*

Section 3: Modeling

Part A

Assumptions

The methods attempted for analysis were Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Random Forest, and Boosted Trees. Logistic regression models the log-odds of the response variable as a linear function of the predictors. Similar to a linear regression model, this method assumes that there is no multicollinearity in the data and that the observations are independent of each other. Neither of these assumptions is completely met by the data. The adjacent zenith angle measurements tend to be highly correlated with values as high as 0.971 and 0.953. However, since there are eight predictors and this issue only occurs with a subset of them, this should not have a major impact on the analysis though it is certainly a concern moving forward. Additionally, as discussed in Section 2, the data points have a spatial dependency that we are trying to preserve, so they are not independent of each other. Even so, each training set has at least 100,000 data points spread across the three images so this should not have a strong impact on analysis.

LDA models the data from each class of the response variable as normally distributed with a covariance matrix that is shared across the classes. This method makes some quite strong assumptions, primarily that the underlying data is normally distributed and that the variance is the same for the two groups. Based on the histograms of the variables, the variances seem to be approximately equal for each class, though the cloud free data tends to have slightly higher variance. The data does not seem to be normally distributed as it is not symmetric, however it is unimodal for most variables and there are enough data points that the models should still function effectively. QDA has similar assumptions, except that it allows the classes to have unequal variance which may model the higher variance for cloud free data better than LDA.

Random forest uses bootstrap samples to make many different decision trees, using a random sample of the variables at each step to split the data along a variable in order to maximize node purity. This decorrelates the trees and reduces the variance. Random forests do not require any assumptions on the underlying data other than that the training data is representative of the population. Boosted trees learn the predictive function slowly, fitting subsequent shallow trees on the same dataset. Just as in random forest, boosted trees only require the training data to be representative of the population.

Results

Each of the models was trained using the train and validation data using 10-fold cross validation. In terms of standardization, only the data for the logistic regression model was centered and rescaled. In addition, the logistic regression, random forest, and boosted trees models were tuned for its regularization parameters; the parameters for each of these models in which the mean accuracy across folds is maximized are shown below in Figure 7.

Fig. 7 - *Optimal regularization parameters based on accuracy*

Model	Parameter	Image Split	Block Split
Logistic Regression	Penalty	0.00033	0.0001
Random Forest	Min Try / Min Nodes	4/25	5/37
Boosted Trees	Min Try / Min Nodes	4/10	6/3

The following accuracy table as well as the metrics in Part B and C are shown for the models with the tuned parameters wherever applicable. Using the accuracy loss function, there is little variation in estimates across folds for each model. Additionally, for each model methodology, the block split method consistently performs better than the image split method, but the difference is very small when validating on the training data, <0.01 difference between each except for QDA in which the block split does significantly better. In general, this trend makes sense because the image split method tends to overfit the training data, as it is only training for one image as opposed to blocks from all three images, and therefore performs worse when it comes to the validation accuracies. This trend also holds when we look at the test accuracy between the different data split methods for our model in Section 4.

From Figure 8, we can conclude that the random forest model with block data split is performing the best out of the eight different model results. Further analysis in Part B and C corroborate this ranking, so for these reasons, we will select the random forest model to run further diagnostics and tuning in Section 4 before predicting on our test data set.

Fig. 8 - *Validation accuracy estimates by model and data splitting method*

Model	Type	Mean Accuracy	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Fold 06	Fold 07	Fold 08	Fold 09	Fold 10
Random Forest	Block Split	0.9696	0.9696	0.9693	0.9698	0.9692	0.9694	0.9691	0.9705	0.9700	0.9700	0.9698
Random Forest	Image Split	0.9627	0.9615	0.9637	0.9614	0.9621	0.9626	0.9646	0.9627	0.9614	0.9634	0.9640
Boosted Trees	Block Split	0.9553	0.9553	0.9540	0.9553	0.9554	0.9549	0.9551	0.9556	0.9545	0.9561	0.9564
Boosted Trees	Image Split	0.9512	0.9508	0.9523	0.9497	0.9495	0.9506	0.9527	0.9523	0.9492	0.9526	0.9523
QDA	Block Split	0.8812	0.8826	0.8813	0.8845	0.8815	0.8803	0.8787	0.8819	0.8817	0.8783	0.8809
LDA	Image Split	0.8727	0.8737	0.8719	0.8723	0.8670	0.8720	0.8712	0.8763	0.8763	0.8713	0.8745
LDA	Block Split	0.8727	0.8746	0.8721	0.8722	0.8689	0.8744	0.8790	0.8762	0.8666	0.8762	0.8666
QDA	Image Split	0.8487	0.8508	0.8502	0.8477	0.8498	0.8459	0.8503	0.8534	0.8477	0.8405	0.8503
Logistic Regression	Block Split	0.8015	0.7999	0.8024	0.8013	0.8003	0.8026	0.8006	0.8009	0.8009	0.8014	0.8045
Logistic Regression	Image Split	0.7977	0.7972	0.7986	0.7946	0.7963	0.7947	0.7974	0.8010	0.7983	0.7983	0.8001

Part B

Below is the graph of the ROC curves for both types of data splitting. The curves shown for each model are for the model with the best tuned parameters. We chose a cutoff value of 0.5 by looking at the false positive rate of our best performing model, which for the block split random forest was approximately 0.49. Like we see in the accuracy estimates table, the random forest and boosted trees model are

predicting cloudiness markedly better than the others at the cutoff value. Interestingly, LDA performs considerably worse at the block split than the other models.

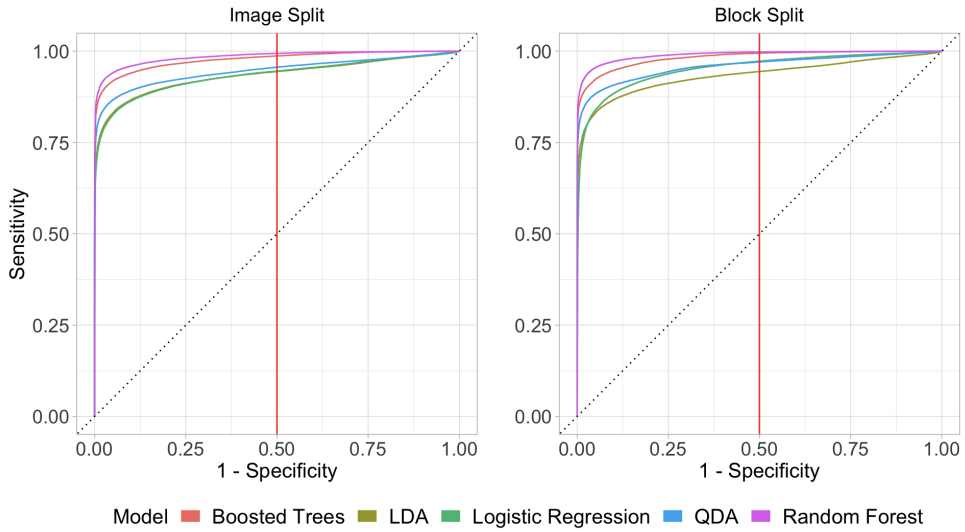


***Fig. 9** - ROC curves for model by data splitting method with cutoff value = 0.5*

Part C

Additional metrics that we analyzed were precision estimates, confusion matrices, and F1 scores. The precision estimate is the proportion of true positives over all positive predictions. Similar to accuracy estimates, each model is more precise in its classification for the block data split vs the image split, and the random forest ranks best out of all models. We do see that random forest and boosted trees models have lower precision than accuracy, whereas logistic, LDA, and QDA are either similar or doing better in precision. This suggests that random forest and boosted trees have more false positives than the other models, probably because logistic regression, LDA, and QDA have a higher rate of false negatives (as seen in figure 12 below).

Model	Image Split	Block Split
Random Forest	0.8968	0.9264
QDA	0.8395	0.8835
Boosted Trees	0.8574	0.8826
Logistic Regression	0.7988	0.8548
LDA	0.7933	0.7934

***Fig. 10** - Precision estimates by model and data splitting method*

The F1 metric is another way of measuring a model's accuracy, in which it balances precision and recall, which is the proportion of true positives over all points that should be labeled as positive. All models have a high F1 score similar to that of precision, except for QDA which has a lower F1 score than precision. This suggests that it has more false negatives that pull the F1 score down relative to its precision.

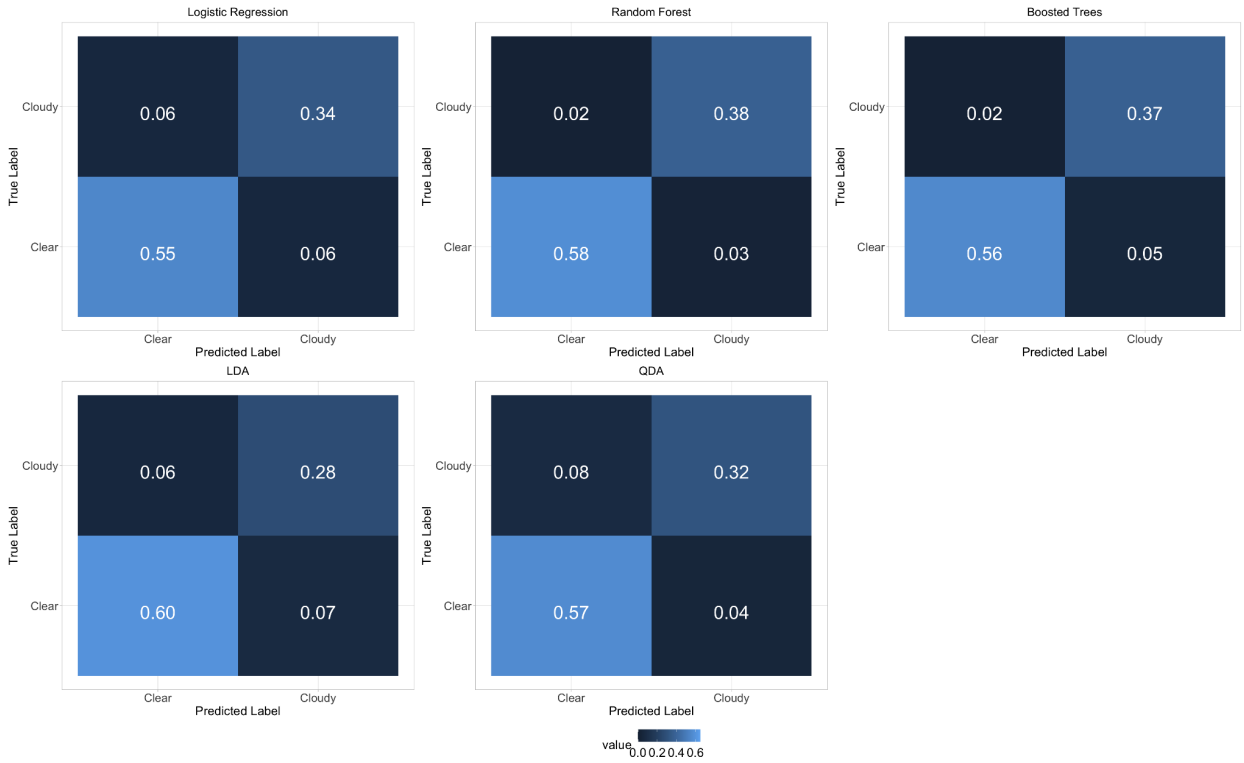
Model	Image Split	Block Split
Random Forest	0.9177	0.9412
Boosted Trees	0.8983	0.9176
Logistic Regression	0.7980	0.8546
QDA	0.7462	0.8413
LDA	0.8125	0.8125

***Fig. 11 -** Table of F1 metrics for each model*

Additionally, the confusion matrix shows the breakdown between model classification and the true labels. The off-diagonals are the true positives and true negatives, and the diagonal entries show the false positives and false negatives. Examining the confusion matrix for each model reveals that most of the

variation across the models occurs for false negatives, or when the model falsely classifies a point as clear when it is actually cloudy. These results follow our intuition on the performance of the models, since the majority of the data has clear data points and so the models tend to prefer to classify points as clear. The below figure shows the confusion matrices for the model with the block image split method. These results show the same trend as previously discussed, where random forest and boosted outperform logistic, LDA, and QDA models.

Fig. 12 - *Confusion matrix plots for block split method*



Section 4: Diagnostics

Part A

The best classification model based on earlier testing was the random forest, which is a bagged machine learning model that averages the results of many different decision trees. It is a flexible model and as discussed earlier, requires no strong assumptions about the underlying data or the form of the relationship with the response value. A decision tree considers all variables in the training data and splits the data into two groups based on the value that minimizes the node impurity in the resulting groups where the predicted class for each group is the majority class of the points in that group (Figure 13).

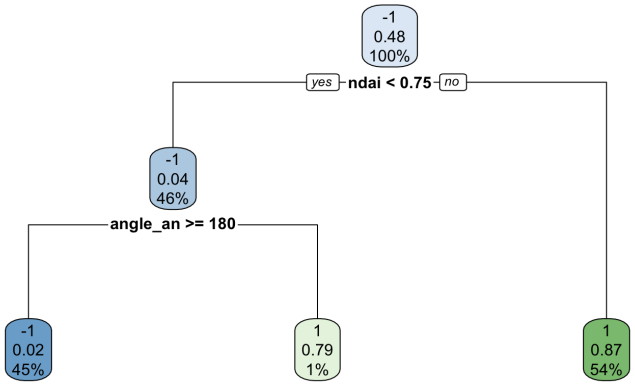


Fig. 13 - *Decision tree based on the block train data*

This continues recursively with each group until the groups are sufficiently small without being overfit. A random forest builds many decision trees and averages the predictions. For each tree, a random subset of that data is chosen and at each split in the tree, a random subset of the variables is available to be split on to decorrelate the trees. This prevents the model from building the same tree many times.

Random forests have three main hyperparameters: number of variables available for splits, minimum node size, and the number of trees. Increasing the number of variables available for splits will decrease bias because the "best" variable for a split will be more likely to be in the subset of variables allowed, but it will also increase the correlation between the trees which can increase bias. Increasing the minimum node size will lead to shallower trees which reduces the computational complexity, but also makes the model less flexible. Increasing the number of trees will decrease the variance of the model, but does have diminishing returns after a certain point. To limit computation requirements, we first varied the number of variables available for splits and minimum node size with the number of trees set to 50, then varied the number of trees separately using the best values for variables to split on and node size. The initial tests were run using the block split data because we expect it to be more accurate since the training, validation, and test sets each include data from all three images while also preserving spatial dependency. To compare results, all models were trained on the training set and tested on the validation set.

Fig. 14 - *Heatmap of random forest parameter tests (block split data)*

As Figure 14 shows, many of the hyperparameter values resulted in similar accuracies. To favor a more restricted model, we chose four for the number of variables available to split on and 25 for the minimum node size, resulting in a validation accuracy of 91.42%. Then we increased the number of trees from 50 to 100, 250, and 500; however, we found that increasing the number of trees had barely any impact on the prediction accuracy, so we decided to stick with 50 trees for the final model.

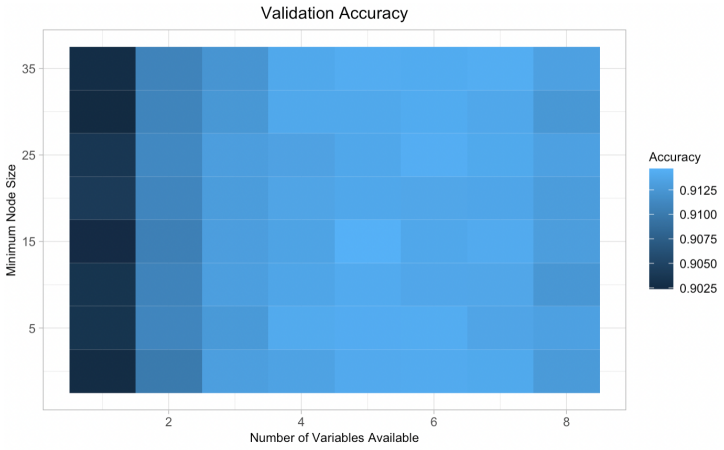
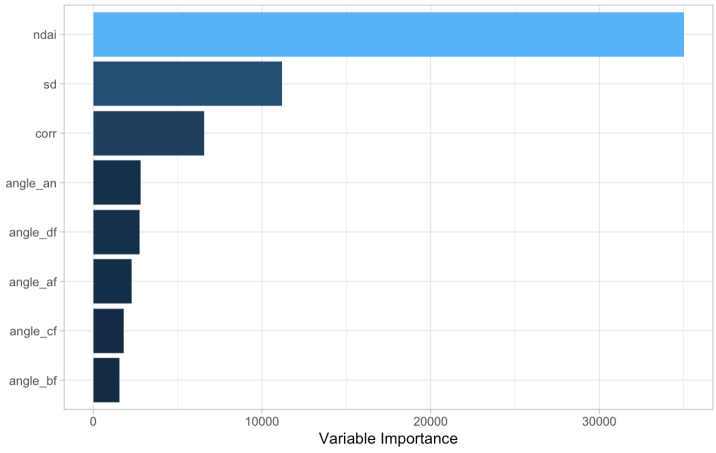


Fig. 15 - *Variable importances (block split data)*



The test accuracy for the final model was 95.78%. The most important variable by far was NDAI, with the SD and CORR as the next most important variables (Figure 15). This lines up with our best features in Section 2C. NDAI had the highest correlation with the expert labels out of all the variables and even though SD had a lower correlation with the expert labels than CORR, its range was more separable based on class which was advantageous for tree splits.

Figure 16 shows the probability predictions from the random forest model on each image. These predictions should be taken with a grain of salt because each image includes training, validation, and test data; but it's an interesting visualization to see not only how the model predicts on the cloudy and cloud-free data, but also the predictions on the unlabeled data. Most of the lighter points in the image with probability values closer to 0.5 are the unlabeled points which concurs with the experts' difficulty in labeling them, however a lot of the unlabeled points have probabilities close to 0 or 1 based on the values of the variables which could improve on the expert labels.

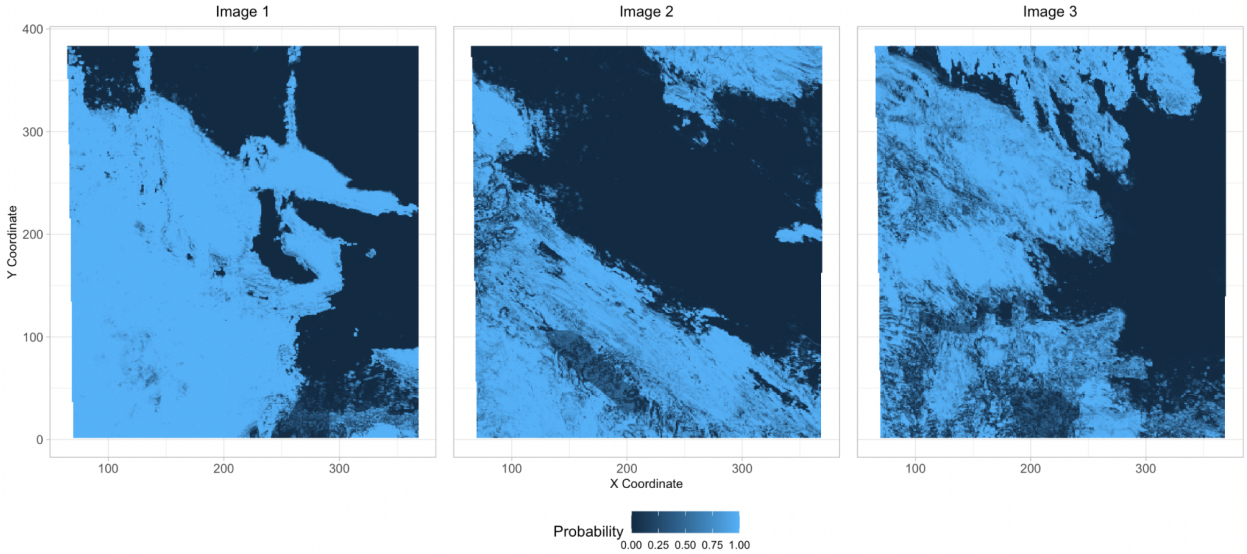


Fig. 16 - *Probability predictions from final random forest model (block split data), value of 1 is cloudy*

Part B

Figure 17 shows the misclassified points in each image. The more transparent points are the correctly classified points while the more opaque ones are the misclassified points. The gray grid overlaid on the plot shows the blocks that were used to split the data into train, validation, and test sets. Most of the misclassified points are labeled as cloudy when they should be clear and the majority of misclassified points are near the boundaries of the blocks. Both of these outcomes were expected based on the design of the model. Based on the ROC curve, the random forest model has much higher sensitivity than specificity, so it classifies almost all of the cloudy points correctly but has many false positives. Additionally, as described in Section 2A, when the data is broken up into blocks the spatially dependencies in the interior of the blocks are preserved but not as well at the boundaries. The points at the boundaries lose some of their neighbors, which decreases the predictive performance at those points.

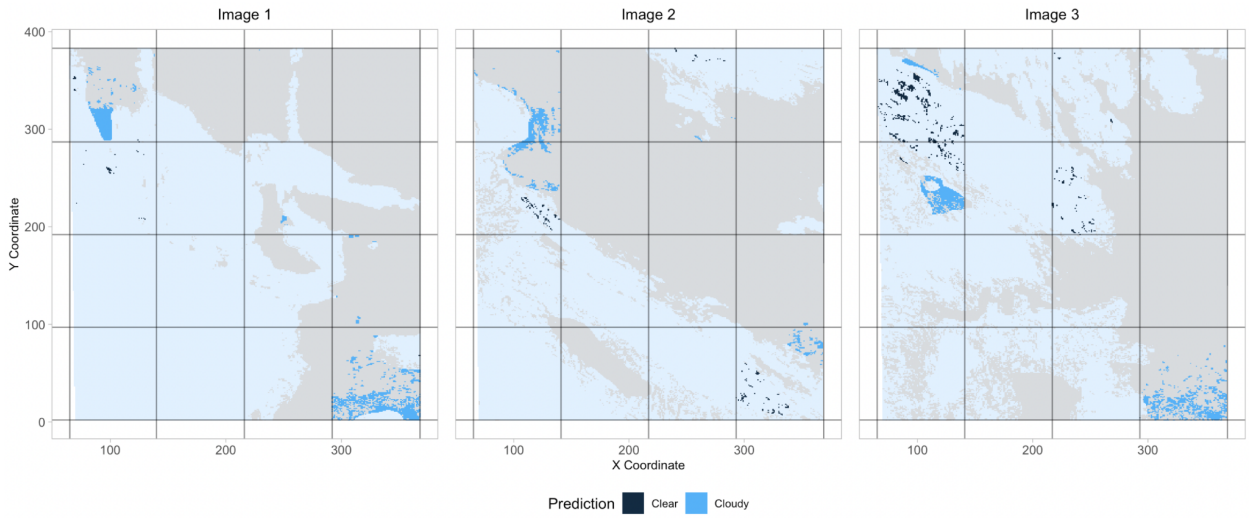


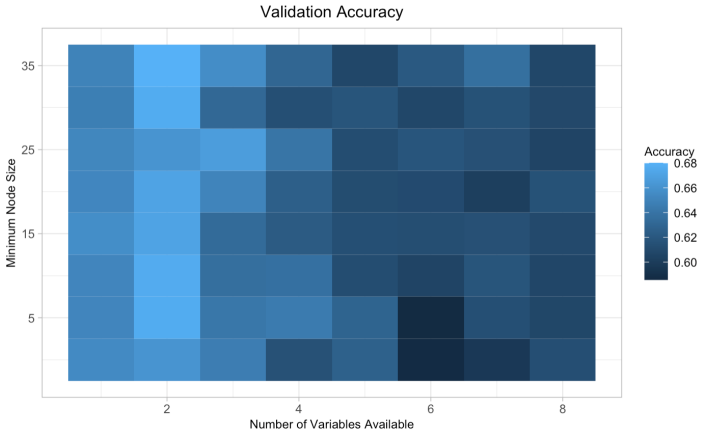
Fig. 17 - *Final model predictions with misclassified points more opaque and overlay of block split grid*

Part C

Based on the errors from the classifier, there are a few ways we could modify the modeling process to improve the results. One possible reason for the low specificity of the model is the class imbalance between cloudy and cloud-free data. There are significantly more cloud-free points, which makes it difficult for the function to predict on those values. To mitigate this, we could augment the data by training a generative model on the data such as LDA or QDA and use them to generate new data, subsetting the results to only increase the training set for cloudy data. Since QDA performed better in initial testing, it would likely be the better model to generate data from. Then, the new data that fits the distribution for cloudy data could be added to the training dataset to counteract the class imbalance. To address the misclassifications near the boundaries of the block split, we could fit several different models that each use different shapes and sizes of blocks and then take the majority vote of the predictions. This would ensure that data points are not always at the boundaries of the blocks and preserve more spatial dependency. However, we expect this model to perform well on new data without expert labels because of the high validation and testing accuracy. Additionally, the probability models show the uncertainty of the predictions which can quantify the confidence on future predictions.

Part D

Fig. 18 - *Heatmap of random forest parameter tests (image split)*



We repeated the process of choosing the hyperparameters and building a final model with the image split data. However, the validation error for all the models was much lower with this data split because the models overfit to the training image and thus predicted poorly on the validation image. Just as with the block split, increasing the number of trees from 50 did not have a significant effect on the model so we set the number of trees at 50. Based on the validation error from tests on the number of variables available and the minimum

node size, we chose the values 2 and 35, respectively, resulting in a validation accuracy of 66.10% (Figure 18). The variable importance values followed the same trend and the test accuracy for the model was 74.54%. Not only is the accuracy much lower for this model, but the validation set is not as good of a proxy for the test error with a difference of 8.44% (compared to 4.36% for the block split).

For this model, there were no misclassified points for the training image as the model overfit completely to the training data. For the validation and test images, the misclassified points were spread throughout the image and occurred primarily on cloud-free data. However, overall this splitting method is less valid than the block split because it does not account for differences between images. Splitting based on image would be the ideal method to preserve spatial dependency, however to be accurate it requires many images in each of the sets to dilute the differences between images.

Part E

Out of all the models, the random forest trained on the block split data performed the best with a validation accuracy of 91.42% and a test accuracy of 95.78%. This result is not particularly surprising, as random forest is one of the best "off-the-shelf" classifiers and the block split method is more generalizable because the training set includes data from all three images. However, the model struggles to predict values at the edges of the blocks and has lower specificity in part due to the class imbalance between the data. This could be mitigated by augmenting the data using a generative model and using multiple block splitting arrangements that essentially smooths out the difficulties seen at the boundaries of using just one block split. Improving this model even further would significantly help researchers using satellite imagery to detect where the image is blocked by clouds.

Section 5: Acknowledgement

When working on this project, we started by reading the paper and doing some basic EDA. Then we created the two data splits for the train, validation, and test sets and predicted on them using the trivial classifier. We used the EDA to choose the best features and began designing the CVmaster function. After testing many different packages, we decided to use the tidymodel framework in the CVmaster function and began adding more metrics to the function. Once it became clear that random forest would perform the best, we worked through the diagnostics section to choose the best hyperparameters and build the final model using the block split data. Then we used visualizations to analyze the final model and repeated the process with the image split data, and then concluded with our results.

For this project, we used "An Introduction to Statistical Learning with Applications in R" and "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" to understand the structure and assumptions for each model. Jackie did EDA, wrote Section 1, wrote the block split function, rewrote the CVmaster function using tidymodels, and did the results and visualizations for Section 3. Alison designed the splitting methods, did the trivial classification, chose the best features, wrote the initial versions of CVmaster, wrote about the model assumptions for Section 3, and did all of Section 4.

Our zip folder contains four items: a *README.md* that contains documentation on the expected output of our code; a *CVmaster.R* file that contains one function that fits various models and returns model metrics; a *main_processing.Rmd* file that houses the bulk of our code and creates all tables and graphs included in this writeup, and finally a *sta521_proj2_final.doc* that contains the raw Word document.