Project 2 Daytime Arctic Cloud Detection
Keqin Cao 3032137720
Caroline Wu 3031798745

I.Data Collection and Exploration

1a Brief Summary

The purpose of the paper, *Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data with Case Studies*, is proposing a new approach to daytime arctic cloud classification for ice- and snow- covered surfaces using Multiangle Imaging SpectroRadiometer (MISR). Specifically, using three physically useful features, the correlation (CORR) of MISR images of the same scene from different MISR viewing directions, the standard deviation (SD) of MISR nadir-camera pixel values across a scene, and a normalized difference angular index (NDAI). In this new approach, researchers search for cloud-free instead of cloudy ice- and snow-covered pixels. This change in perspective is useful as it is challenging to differentiate between snow-covered surfaces from cloud from the satellites' view. The data used in this study consists of 10 MISR orbits of path 26 across different regions in the Arctic with a rich variety of surface features, especially the ones the current MISR operational algorithm not able to accurately classify. The classifying algorithm is the following: 1) identifying the three features; 2) set fixed or data-adaptive thresholds and apply enhanced linear correlation matching (ELCM) to each unit and produce the first cloud detection product; 3) predict the probability of cloudiness and the second cloud detection product by training for quadratic discriminant analysis (QDA) on ELCM. The result of this new approach of a combination of classification and clustering is promising that it out-performed traditional MISR and offline SVM in accuracy and coverage with a relatively simple algorithm (QDA at most). On a larger scale, this approach can be potentially deployed in other kinds of cloud classification tasks for future studies on atmospheric carbon dioxide. In this analysis, however, we only take three of the images taken by the MISR with 10 features and expert labels of cloudiness to find the best classification algorithm for future images. The study is important in the classification and properties of clouds in the Arctic because influx of data statistics plays a key role in many scientific problems like this study. This introduce of high-level statistics into science problems has given rise to new solutions and strategies. As both statistics and science advance, both will go hand in hand to develop reliable scientific conclusions and models.

1b Summarize Data

For this analysis, we are given three images recorded by the MISR sensor with expert labels on cloudiness along with ten other features. Here are the three images along with the percentage of pixels for different classes of cloudiness (1 = cloud, -1 = no cloud, 0 = unlabeled). The three images (figure 1) clearly show a different pattern of cloudiness, and the histogram shows that the proportion of each label for each image varies. Image 2 has the most label 1, the cloud classification and Image 3 has the most label 0 which means it contains the most unwanted data.

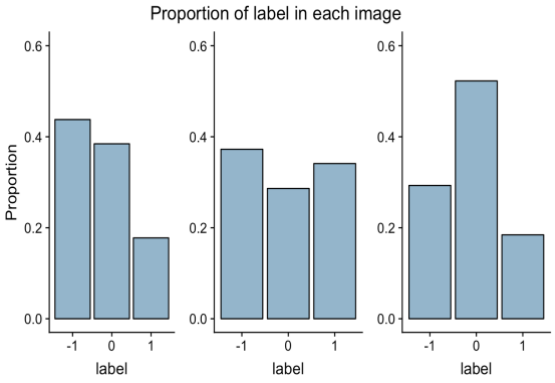
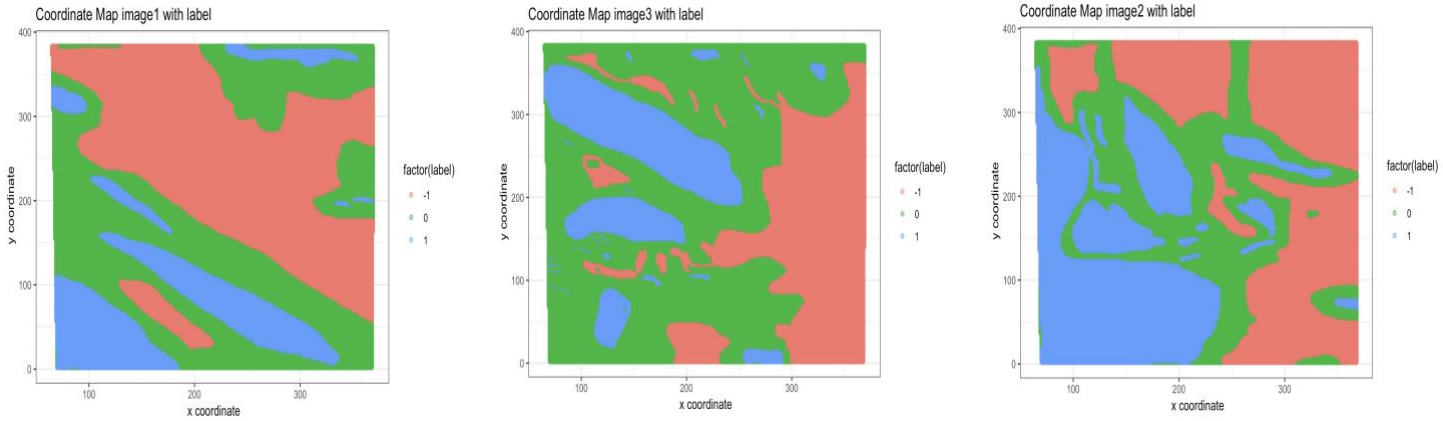


Figure 1

For all three data sets, there are no missing data, which means that each pixel point is accounted for the 11 features. The i.i.d assumption, however, is clearly violated that the cloud appears in large and small patches. This means that if one pixel is classified as cloudy, the pixels surrounding it has a higher probability than random to be cloudy as well as this is a scanned image from the sky. The geographic position causes each pixel to be correlated with the other. It is essential, then, to properly do the data preparation in responding to the i.i.d violate before any further analysis.



1c Visual Quant EDA

The pair-wise relationship between feature is shown in the following correlation plot (Figure 2). The five angular features, most certainly, are highly correlated with one another. Other features are moderately related to each other in either a positive or a negative way but there's no clear pattern of any relationship. To further understand these features, we plot the relationship between an individual feature and expert labels with overlaying histograms (figure 3). The classification patterns are similar among CF, BF, AF, AN, and DF angels but diverge significantly among CORR, SD, and NDAI. This pattern is notable since the article uses these three features to develop a new cloud classification algorithm. The distribution looks fairly normal with some bimodal peaks. For the overlaying histogram for CORR, data with labels -1 are clustered around the mean while data with labels 1 are more spread out in overall distribution. Comparatively speaking, for NDAI, the data with true labels -1 and 1 do not overlap. This suggests a potential good feature in a binary classification problem which we will take into consideration while fitting the model.

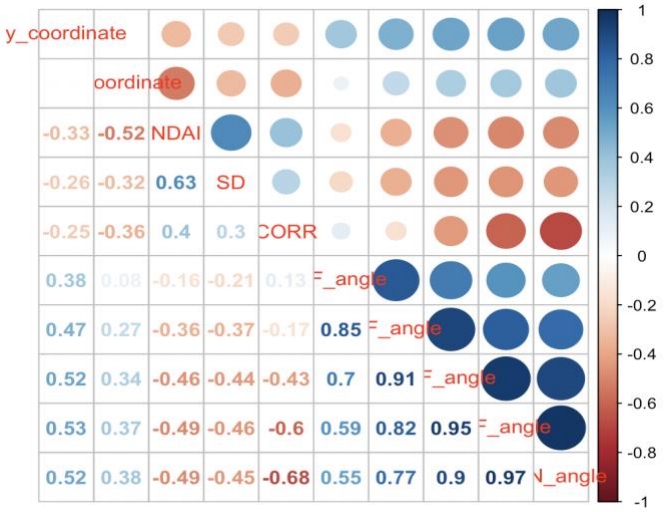


Figure 2

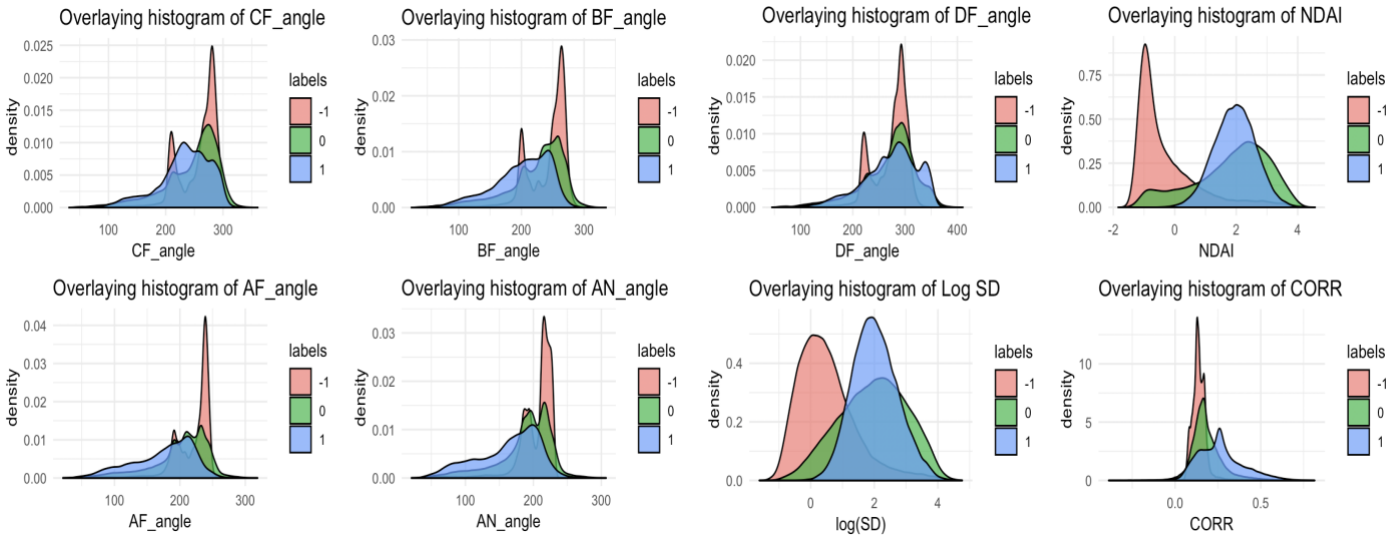


Figure 3

II.Preparation

2a Data Split

To compensate for the i.i.d violation, we suggest some methods as following, all with an 80-20 train-test split. The first method is to sample by blocks, which are divided according to XY coordinates. We picked the block size by balancing a low correlation amongst nearby block but ensure that smaller pieces of the cloud are represented when sampling. The second method is to sample by the outcome. As the histogram of % pixels for each class shows, the pattern for each image is quite different. Taken that into consideration, we keep the overall class distribution of the original data by sampling by each label, -1, 1, 0, to create a balanced split without the i.i.d assumption. In later analysis, we will employ both the data splitting methods described above. Another possible way of splitting the data is to circle the contour and find the boundaries of each pattern of the cloud/non-cloud and then sample within the boundaries accordingly. This shares similar characteristics as sampling by labels, but it will ensure each cloud gets sampled in the training phase.

2b Baseline

To establish a baseline for our analysis, we first arbitrarily set all labels to -1 on the validation set and test set and throw out any points with expert label 0. The accuracy of this classification for the validation set is 0.6115 and for the test set is 0.6103. This classification will have a high accuracy when the train and validation data after the initial data split has high proportion of -1. It indicates that our data split did not effectively compensate for the i.i.d violated observed in the EDA. If this were to happen, we need to refine our data splitting method.

2c First order importance

To visually establish the first order importance of three “best” features, we plot a group of boxplots, comparing the distribution of each feature between label -1 and 1.

The "better" a feature is, the more variation in distribution one should see. In figure 4, we observe the three features with the most variation in boxplot distribution: NDAI, CORR, AF_Angle. Quantitatively, we construct linear models, fitting each feature on at a time and compare their r-squared value. The table below shows the r^2 value descending. Interestingly, both visual and quantitative EDA gives consistent choices of the three most important features. We are omitting the x_coord and y_coord features here because we know that the three images are from MISR blocks 20-22 over three consecutive orbits. We will not know how the future data comes in so geographic features does not provide the first order importance here.

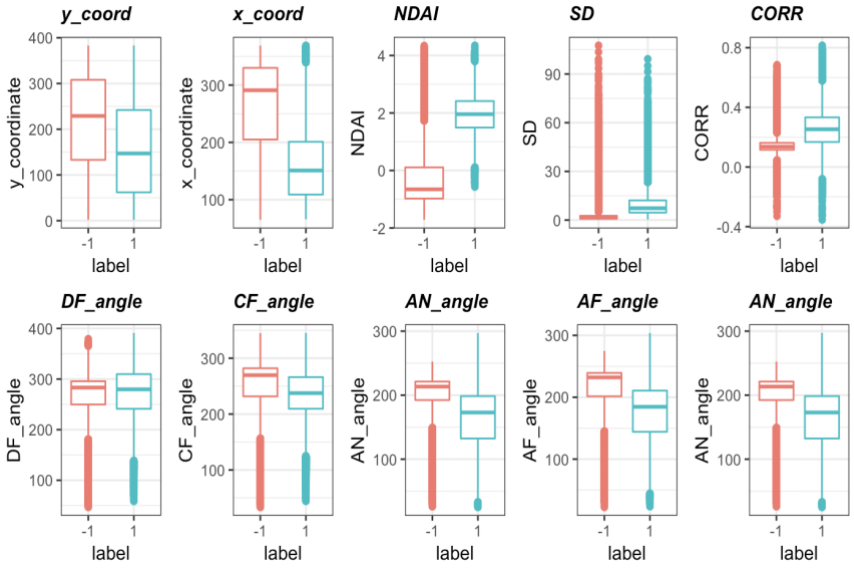


Figure 4

Feature	NDAI	x_coord	CORR	AF_Angle	AN_Angle	SD	CF_Angle	y_coord	DF_Angle
r^2	0.575	0.326	0.304	0.258	0.255	0.191	0.080	0.079	0.0001

2d CVgeneric function

CV generic function is detailed described in the R script. The CVgeneric in R that takes a generic classifier, training features, training labels, number of folds K, training data, a loss function, a pre-defined, seed as inputs and outputs the K-fold CV loss on the training set. There are four generic classifiers: logistic regression model, LDA, decision tree and random forest. Since the project is an image classification problem, the metrics we use are accuracy, precision, and recall.

III.Modeling

Before fitting the image data into any model, we decide that we do not standardize our variables since otherwise, the angle data does not make sense. We try the same model on both scaled and non-scaled data, and the model does not improve accuracy after scaling. We start by fitting relatively simple models with more assumptions on the distribution of training data and no hyperparameters. For each model, we plot a ROC curve to show the cutoff for classification along with how well this model fits the dataset.

Method 1: LDA

LDA assumes normally distributed data, features are statistically independent, and identical covariance matrix in every class. Although we seek to reduce some violations of LDA assumptions when splitting the data, the identical covariance assumption is still slightly violated.

The cutoff and performance tradeoff graph show that the optimal cutoff is at 0.37, which means that if the measurement is less than the cutoff, the predicted condition is negative (no cloud); otherwise, the predicted condition is positive (cloud). At this cutoff value, the model achieves the highest average true positive rate with lowest false positive rate.

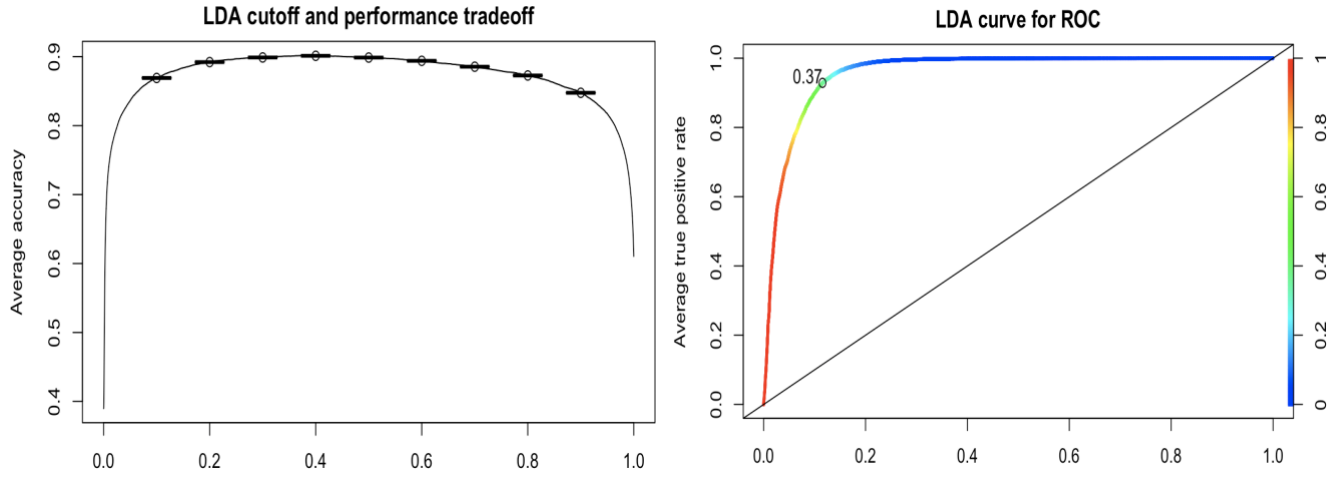


Figure 5

Method 2: Logistic regression

Unlike most linear models, the logistic equation does not assume linearity, normality or homoscedasticity. A binary logistic regression requires the dependent variable to be binary, in our case, it is 1 and -1. For the independence requirement, we compensate for this violation in the data split for both data splitting methods to minimize the dependency of adjacent pixels. However, we slightly violate the low multicollinearity among features, due to the high correlation among angle features. Nevertheless, those features are not the most important as we have shown in the previous section. The last assumption of a logistic model is a large sample size, which the image dataset clearly satisfies. Logistic Regression does not have any hyperparameter. The cutoff and performance trade-off graph show that the optimal cutoff is at 0.39, which means that if the measurement is less than the cutoff, the predicted condition is negative (no cloud); otherwise, the predicted condition is positive (cloud). At this cutoff value, the model achieves the highest average true positive rate with lowest false positive rate.

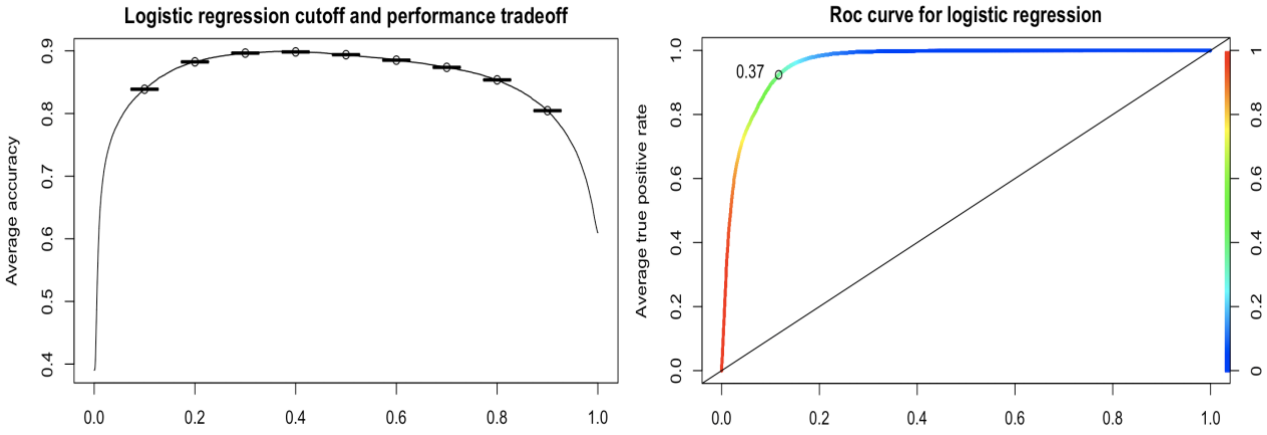


Figure 6

Method 3: Decision Tree

The decision tree method is a nonparametric classifying algorithm, which makes no assumption on the distribution or error of the dataset. Rather, it is constructed based on the observed data and important features. For our dataset, we set the maximum depth as 10 since we only have 10 features to prevent overfitting. There are two stopping criteria for decision tree model: when maximum depth is reached or the minimum of tuples in a leaf node is reached. The stopping criteria are set to prevent overfitting. In our case, the depth stops at 3, which means that it only takes 3 depths for the model to fully converge. There is no need to further tune the hyperparameter as the train error and validation error do not differ. This ensures that there is no overfitting problem. In later sections, we will discuss more in-depth on how the decision tree fits well with our dataset in terms of model assumption and overfitting prevention. The cutoff and performance tradeoff graph show that the optimal cutoff is at 0.8, which means that if the measurement is less than the cutoff, the predicted condition is negative (no cloud); otherwise, the predicted condition is positive (cloud). At this cutoff value, the model achieves the highest average true positive rate with lowest false positive rate. The ROC curve (figure 7) is not as smooth as the LDA or logistic model which shows the decision tree makes less hesitation in classifying the labels.

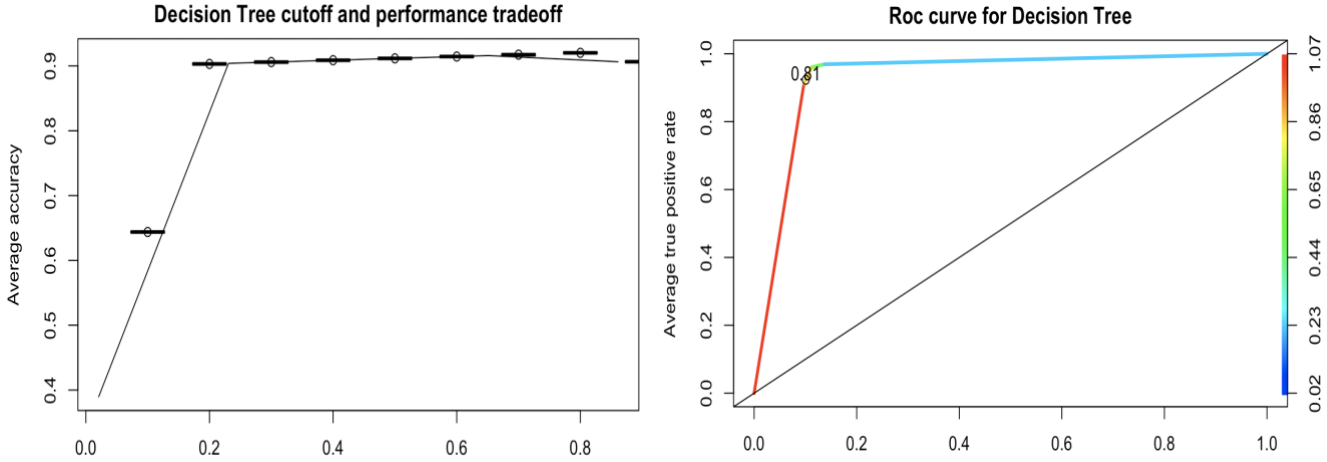


Figure 7

Method 4: Random Forest:

Taken into accounts of the high accuracy of the decision tree model, we decide to fit a random forest model which is constructed by multiple decision trees. Similarly, the random forest model does not have assumptions on the dataset. What differentiates random forest from decision tree is that it does not run on all features at once but to randomly selected a given number of features defined by the user by bootstrapping. This parameter is defined as ntry (number of features selected at each split) and we tune this hyperparameter by plotting out (figure 8) the out-of-bag (OOB)

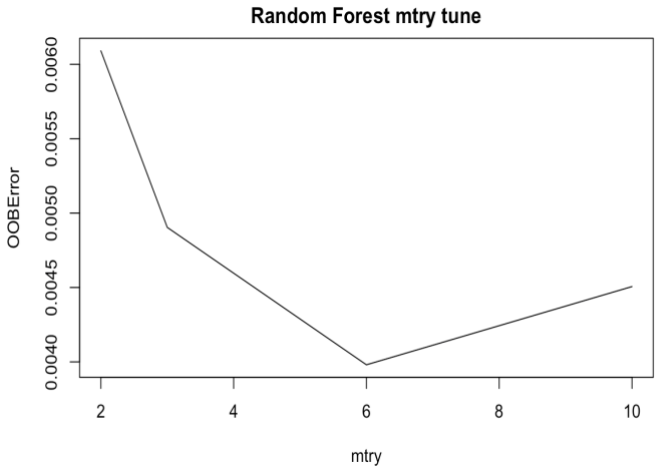


Figure 8

error, an unbiased estimator of the test error. When the number of features is six, the model reaches the lowest OOB error. For ntree, we use the default at 50 since we do not find any justification to increase or decrease model complexity. Node size is simply the number of features we have, and we set importance to true since some features are more important than other features. The cutoff and performance tradeoff graph show that the optimal cutoff is at 0.66, which means that if the measurement is less than the cutoff, the predicted condition is negative (no cloud); otherwise, the predicted condition is positive (cloud). At this cutoff value, the model achieves the highest average true positive rate with lowest false positive rate. The right-angled-line shows that the random forest is the best model choice among all above.

To compare and contrast each model, we first use the CVgeneric function to evaluate the CV across folds for both data splitting methods.

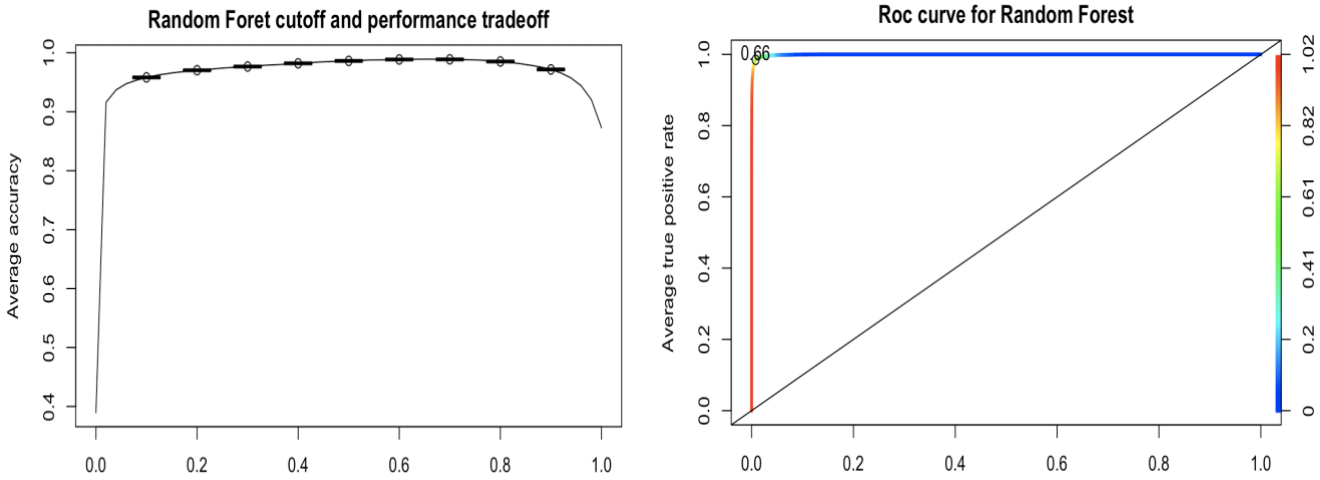


Figure 9

CVgeneric	LDA	Logistic Regression	Decision Tree	Random Forest
Split Method 1 (By Block)				
Fold 1	0.853	0.883	0.920	0.985
Fold 2	0.861	0.885	0.925	0.986
Fold 3	0.857	0.885	0.923	0.985
Fold 4	0.854	0.885	0.921	0.984
Fold 5	0.857	0.886	0.922	0.985
Average	0.856	0.885	0.922	0.985
Split Method 2 (By Label)				
Fold 1	0.853	0.883	0.920	0.987
Fold 2	0.860	0.886	0.924	0.987
Fold 3	0.856	0.887	0.921	0.987
Fold 4	0.857	0.884	0.923	0.986
Fold 5	0.861	0.885	0.924	0.988
Average	0.857	0.885	0.923	0.987

We then use four more metrics Precision (proportion of positive identifications that was actually correct), Recall (proportion of actual positives that we correctly identified), F1 Score (a weighted average of the precision and recall), and Sensitivity (the proportion of actual positive cases that got predicted as positive e) to further explore the relationship between true positives, false negative and false positive.

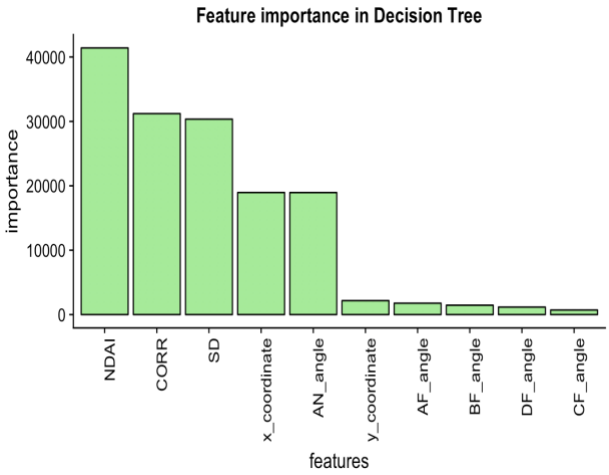
Metric/Model	LDA	Logistic Reg	Decision Tree	Random Forest
Recall	0.856	0.884	0.857	0.972
Precision	0.888	0.809	0.959	0.992
F1 Score	0.914	0.916	0.934	0.988
Sensitivity	0.919	0.928	0.973	0.994

IV.Diagnostics

4a 4b Good model, Best model

From the chart comparing the accuracy, precision, recall, F1 score, and sensitivity, we decide that decision tree is a good model and random forest is the best model not only due to the satisfying metrics but also the model assumptions, applicability to new datasets and robustness. First of all, neither model has restrictive assumptions on the dataset, which is rather practical in the cloud dataset since we cannot foresee the future data.

For decision tree, we mentioned in previous section that it prevents overfitting by setting two stopping thresholds. Note that in this section, we modify the model by excluding x and y coordinates as features. X and Y coordinates are not effective predictors as they are both relative positions of the cloud. We show both decision trees in figure 11 with and without the X, Y coordinates for comparison. The accuracy for new model is 0.904. Decision tree classification is based on the Gini index as splitting criteria. Even though the cross-validation accuracy showed that the there is no overfitting in the training set, we want to double check if we have overfitting the data by using pruning here, a method used in a decision tree to balance the cost complexity and model outcome. We have the complexity parameter, CP, (figure 10) which is used to control the size of the decision tree and to select the optimal tree size. The trend is decreasing monotonically which shows that there is no need to prune the tree. In most cases, we need to set mini-leaf and max depth to prevent overfitting. If we plot the decision tree, however, there are only 3 features that are considered important by the algorithm.



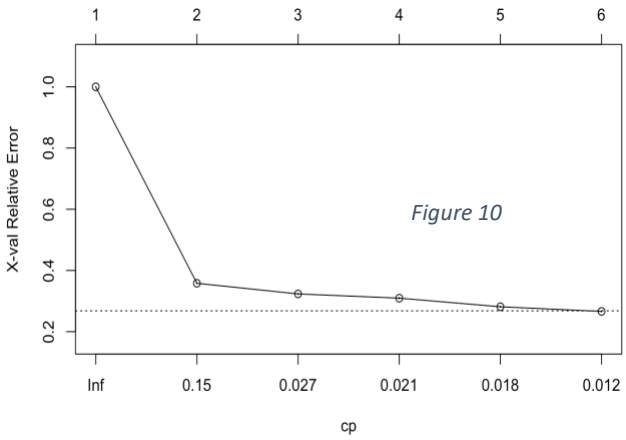


Figure 10

Therefore, it confirms that there is no need to set extra parameter or use pruning which reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.

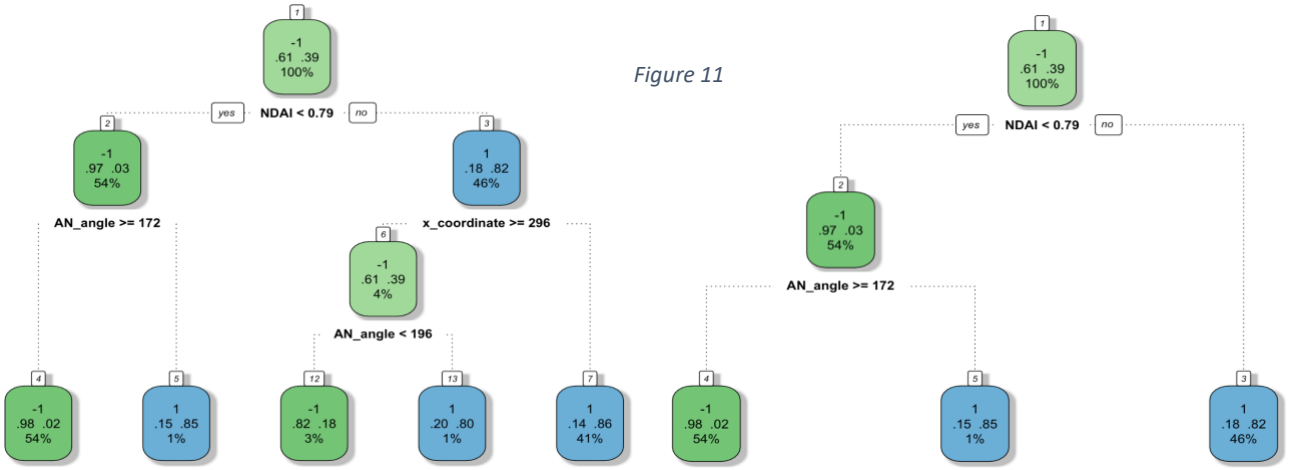
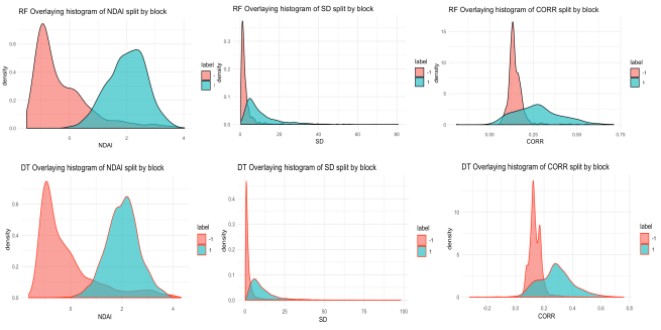
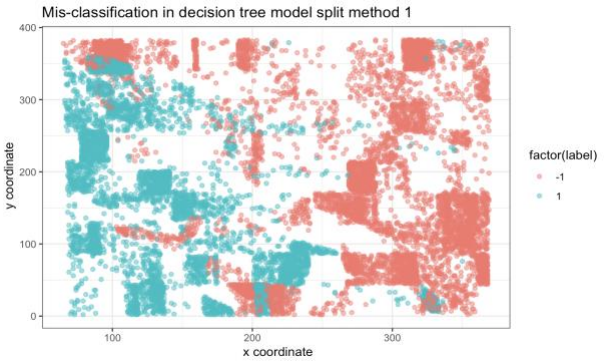


Figure 11

Random Forest is the best model since has the highest measurements in all metrics, nevertheless hard to visualize each tree in the random forest and understand how they are like and interact with each other. Therefore, the most effective illustration of model selection in our case is to show both random forest and decision tree for conceptual and visual clarity.





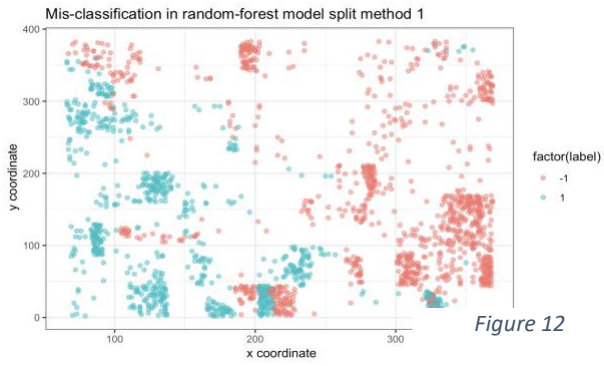
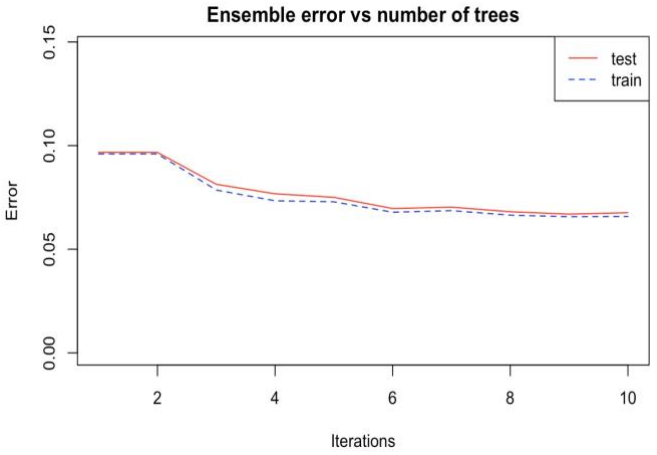


Figure 12

Lastly, we examine the misclassification in both decision tree and random forest (figure 12) and observe a similar systematic pattern across two models: misclassification on label -1 is clustered on the upper right region while on label 1 is clustered on the lower left region. The misclassified labels are more densely distributed in decision tree than random forest, which is consistent with our analysis of model metrics. From the overlaying histogram of features for misclassified labels, we see that both models have the same pattern and the histogram aligned with overall distributions.

4c Better classifier -- Adaboost

To improve our classification, we use AdaBoost which emphasizes on previous mistakes to adjust weights on further iterations. In this case, our optimal cutoff is 0.57 and the accuracy with x_coordintate and y_coordinate as features is 0.955, without is 0.997. For example, if I run the model ten times and have the following vector: 2.655310 2.449690 2.381222 2.204961 2.152319 2.122641 2.037177 2.001023 2.039123 1.950600, which is the weight on each iteration of model. Since decision tree is a good classifier, we want to use the ensemble method to further improve our model. We plot the previously calculated error evolution of an AdaBoost, for the training and test data as the ensemble size grows. But we do see when the iterations (ntrees) = 6 it has the lowest error, which corresponds to the random forest hyperparameter, ntry = 6.
Intuitively, the margin for an observation is related to the certainty of its classification. It is calculated as the difference between the support of the correct class and the maximum support of an incorrect class. This is useful to see how fast bagging or boosting reduce the error of the ensemble. Nevertheless, one disadvantage of this model is due to its sequential training that the run time is relatively long which can be potentially problematic for larger datasets.
Since none of the decision tree, random forest, and AdaBoost model requires prior knowledge on the data distribution, we would expect the classification accuracy still holds. However, if we do not have expert labels, it's inevitable that the systematic misclassification still exists.



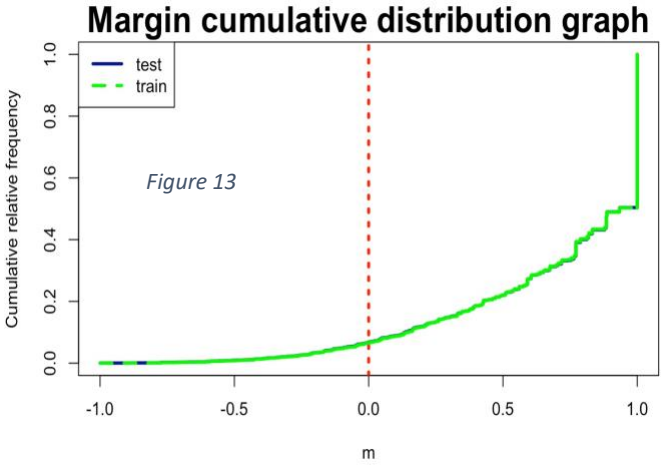


Figure 13

4d Systematic misclassification

Interestingly, our misclassification distribution varies for our two data splitting methods. While the first method has blocks of misclassified chucks and the second method has stratified misclassification chunks. The difference in pattern is presumably caused by how the original dataset is split that method 1 is sampling by blocks and method 2 is sampling by strata. Thus, the error pattern would be consistent with the sampling method. Another consistency across two data splitting method is that there is more misclassification in decision trees than random forest.

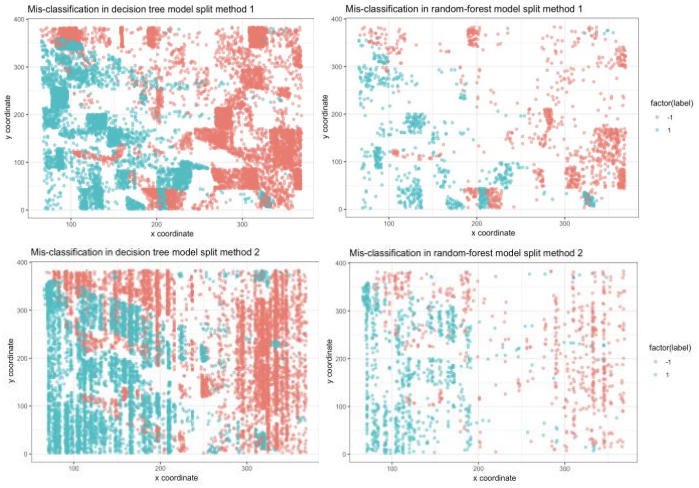


Figure 14

4e summary

Comparing our models above, we observe an increase in classification accuracy as well as in other metrics when model complexity increases. However, the misclassification plots indicate a consistent pattern of error regardless of the models but instead associate closely with the data splitting method. Additionally, we notice a significant runtime increase as model complexity increase. As we first attempt to recreate some of the models mentioned in the paper, we are not able to fit a SVM model as it is too computationally expensive. When we have more images in the future, run time is an important factor in choosing a better classification model. Specifically, with our three images, the increase in accuracy is not as significant as the increase in runtime. More complex models also have difficulty in visualization. For example, we are able to plot the classification process for decision tree but for random forest, it is impossible to show this process as there are too many trees and nodes. Another factor of consideration is the model assumptions since the cloud image data clearly violates the i.i.d condition required by many models and we are not able to standardize the data due to the angle features. With that in mind, model that has few restrictions on the data distribution will be a better fit. Even with careful model selection, the first step of data splitting lays the foundation of further analysis. In our project, we employ two methods, sampling by block and sampling by outcome. Especially for sampling by outcome, the expert labels are essential. In the paper, there is no instruction how to deal with label 0. Our analysis throws out all label 0 and establish the baseline for classification at 0.61 using a trivial method. If we were to apply our classification algorithm to more images, it is essential that we select a reasonable pool of images to establish the "true cloudiness." Lastly, we reconsider whether to fit the data with all features in the diagnosis section, particularly x and y coordinates. Our analysis of feature importance is consistent with the paper that CORR, SD, NDAI are generally more important in cloudiness classification. In conclusion, our analysis and the original paper prove that effective statistical approach can provide a new perspective on natural science studies. Furthermore, this new approach is not necessarily computationally expensive as demonstrated in both the paper and our project.

Brief contribution:

Keqin Cao: Takes care all Code
Caroline Wu: Write up all the analysis

Together:
Discuss all ideas together

We work together throughout the entire project and we are missing redwoods.
Github Link: https://github.com/sunsgneckq/Stats_154_image_classification