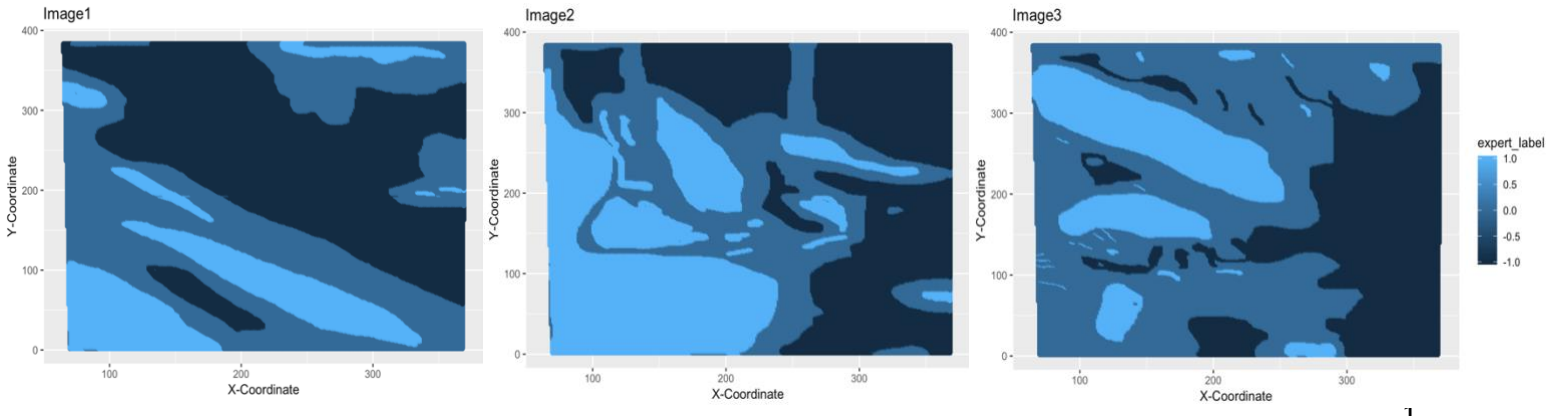
Project 2 Write Up

Angela Gao (3031902238), Chenyu Yang (3034503993)

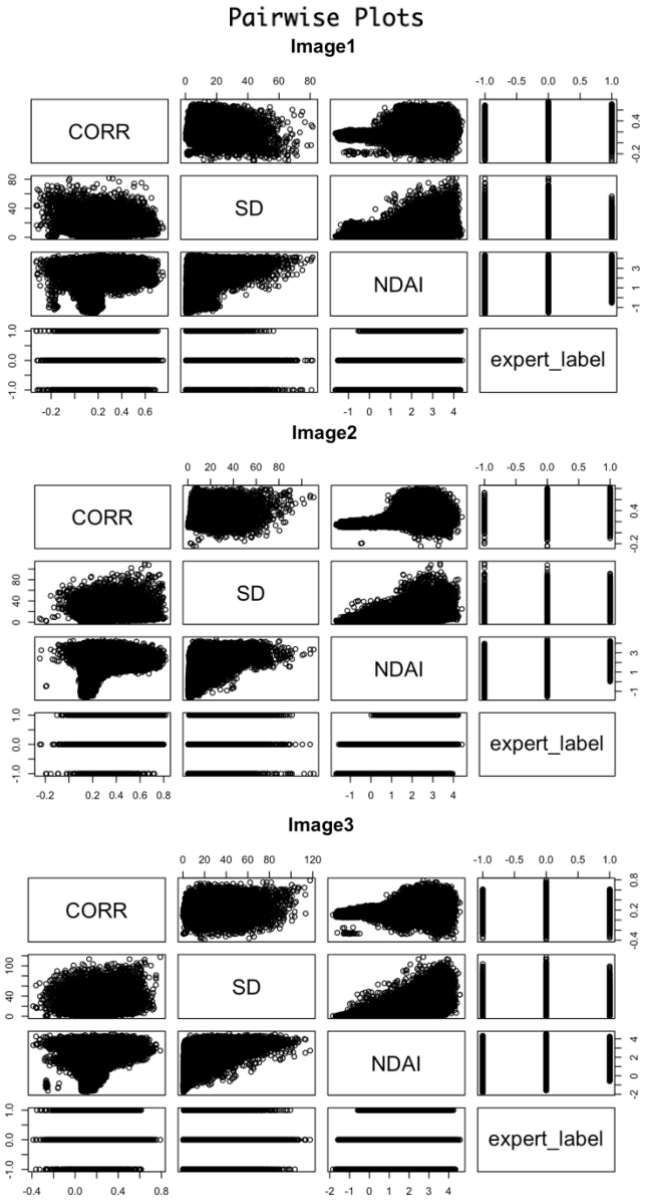
1a) A current scientific area of interest and study is the predicted effect of increasing carbon dioxide levels on global surface air temperatures, especially in the Arctic. In the study of this problem, measuring cloud coverage is especially important as clouds help regulate sensitivity to increasing surface air temperatures. However, cloud coverage properties are difficult to measure because they have similar properties to ice and snow covered surfaces, which leads to problems in cloud detection. The introduction of the MISR has helped with data collection as the sensor has nine cameras viewing the Earth at 9 different angles in four spectral bands. However the current cloud detection algorithm does not work well over polar regions. In addition, the data set collected from the MISR is massive, which is an obstacle to analysis. The purpose of this work was to build operational cloud detection algorithms to efficiently process the MISR data set to study this problem. The data used in the study was collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and Baffin Bay and spans from April 28 through September 19, 2002. 6 data units from each orbit were used in this study, but 3 of the 60 were excluded because the surfaces were open water, leading to a dataset of 57 data units with 7,114,248 1.1-km resolutions pixels with 36 radiation measurements per pixel. The team proposed two cloud detection algorithms based on 3 features: the correlation of MISR images, the standard deviation of MISR nadir camera pixel values, and normalized difference angular index. The proposed algorithms were an enhanced linear correlation matching (ELCM) algorithm, which set thresholds on each feature and applied ELCM to each data unit, and ELCM-QDA algorithm, which predicted probability of cloudiness by training Fisher's QDA on the labels produced through ELCM. The results showed that the ELCM algorithm had agreement rates with expert labels at or above 90% for most cases, which is a significant increase from other algorithms such as SDCM. THE ELCM-QDA did not increase agreement rates much from ELCM, but provided probability labels. The major errors resulting from the ELCM algorithm were due to low correlation as a result of poor terrain data registration, which occurred at sharp elevation changes and led to systematic areas in the algorithm. To test separability of the three features, QDA, logistic regression and logistic regression with L1 penalization were trained on half of the expert labels and tested on the other half from a data unit. To test stability, QDA and logit trained with expert labels from one orbit were applied to a different one. Both tests showed that detectors based on features outperformed those based on radiation measurements. This study has demonstrated that the three features used in our algorithms are sufficient to separate clouds from ice and snow covered surfaces and that the ELCM algorithm provides is more accurate than the existing MISR algorithm for cloud detection. In addition, we can now get probability labels for partly cloudy pixels by training QDA on the ELCM results. This study is significant for both statistics and science. It demonstrates the power of statistics and its potential to contribute to complex scientific problems and has improved our understanding about cloud responses to changes in arctic climate, which can lead to more accurate global climate model simulations and more studies on the feedback of changing cloud properties on atmospheric carbon dioxide.

1b) Using expert labels, approximately 43.78% of the pixels were classified as -1, 38.46% as 0, and 17.77% as 1. In other words, 43.78% of the pixels were not cloudy, 17.77% were cloudy, and 38.46% were not classified. The figure below shows the distribution of pixels by expert label as x- and y- coordinates change.



Pixel Classification for X- and Y- Coordinates of 3 Images

somewhat of a trend for the expert labels based on the x- and y- coordinates, the pixels are most likely not i.i.d, as the trends shows that they are not independent or identically distributed, but are dependent on some other variable or feature.

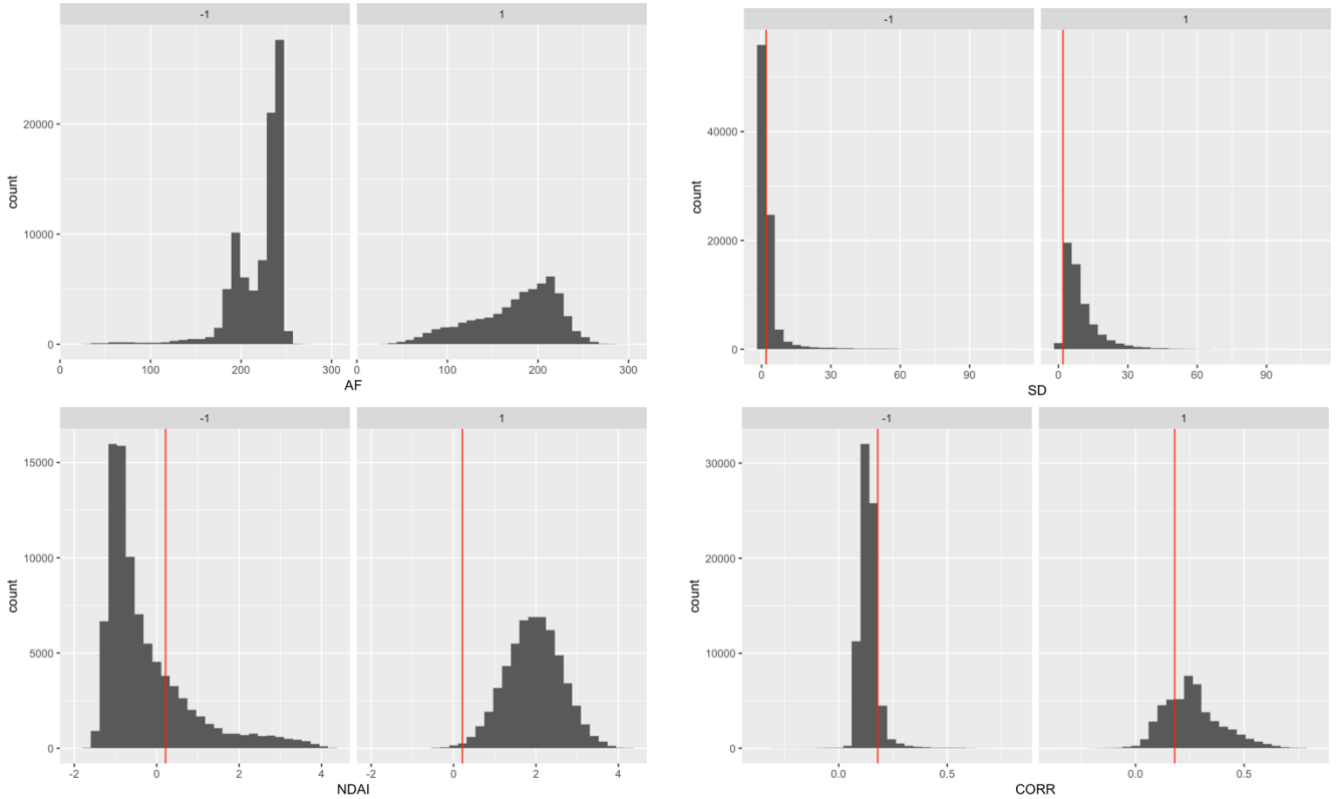


1c) The plots to the left show the pairwise plots for the 3 features and the expert labels. From this plots, we see that correlation and standard deviation do not have any discernible relationship. For small values of NDAI, correlation and standard deviation only take on values close to 0. As NDAI increases, correlation expands to a multitude of values, and the upper bound of standard deviation increases almost linearly. The plots of expert labels do not show us much about data units classified as 0 or -1. However, we see that the range of values for the features taken on by pixels classified as 1 is smaller than the range for 0 or -1. For NDAI, the lower bound is close to -1 and for standard deviation, the upper bound is close to 60. This shows that there may be a significant difference in the features of the cloudy pixels in comparison with the pixels that are not cloudy and the pixels that were not classified. We also performed quantitative EDA on the images to tease apart the differences between the cloudy and not cloudy pixels. We found that the mean and median of NDAI and SD are significantly smaller for cloudy pixels. Correlation is also smaller on average, but that does not hold true for each individual image. Radiances were similar for both classes in the different images, but on average, the cloudy pixels have higher radiances. We also took a look at correlations between the different variables and found that NDAI is correlated ($r>0.5$) with expert label, x, standard deviation, correlation, and radiances. Expert label is also correlated with the x-coordinate and correlation. We also see that the radiance angles AF and AN are correlated with the expert label and features.

2a) We decided to split our data two different ways: 60-20-20 and 70-15-15 for the training, test, and validation sets respectively. Because our sample has a large amount of observations, it was not necessary to train the data on a very large portion of the dataset. However, given that the dataset is not i.i.d. and the outliers or biases could affect the algorithm being trained, we thought it would be safer to still use a relatively larger dataset to train on, which is why we chose the ratios that we did. We also addressed the issue of a non-i.i.d. dataset by drawing the data for each set independently, creating independent training, test, and validation datasets. After analyzing the errors for both split ratios, we decided to go with the 70-15-15 ratio as the CV errors were slightly smaller. The outputs reported for the remainder of this write-up, unless otherwise specified, will be using 70-15-15.

2b) The accuracy of a trivial classifier that sets all labels to -1 is 61.33% on the validation set and 61.23% on the test set. This classifier will have high average accuracy only in scenarios where there are relatively no clouds in the dataset. This step shows that the problem we are about to tackle is not trivial and sets a benchmark for the approximate proportion of clear pixels in our dataset.

2c) From the article, we see that separability and stability are important properties for the features. It is easy to justify this as the data should separate clear and cloudy pixels well to ensure greater prediction accuracy and should be stable for different locations at different times so the training set that we train our models on do not lead to biased algorithms. Given the 2 properties that we set out to achieve, we knew that we could rule out the x- and y- coordinates. X- and y- coordinates do not provide a clear boundary of differentiating clear and cloudy pixels based on a certain threshold and as we saw in part 1b, there may be a trend of expert label changing with the x- coordinate, which shows that the x- coordinate is not a stable feature. We also confirmed this by looking at the correlations, which show around a -0.6 correlation between the x- coordinate and the expert labels. For the remaining features of the radiance angles, NDAI, SD, and CORR, we looked at the correlations and graphed histograms of the features in the training data as a whole and separated by expert label. What we found was that for radiance angles, it was very difficult to decide on a threshold value that could be used to predict whether the expert label should be classified as -1 or 1 as the distributions for both expert labels were similar and the means for both distributions would fall on one side of the threshold. An example of this is shown in the AF angle graph below. However, for NDAI, SD, and CORR, the expert label histogram distributions showed skews toward different sides separability, which was an important property we were looking for. If you compare the NDAI, SD, and CORR graphs with the AF angle graph, you see the difference in distributions that is not present in the radiance angle, but is in the other features. In addition, looking at the correlations showed that NDAI and CORR had the highest correlation with expert label, which shows that they may have the largest effect on expert label. Using this criteria, we concluded that NDAI, SD, and CORR were the "best" features to work with. We also determined possible threshold values and plotted them on the plots. Our thresholds for SD, CORR, and NDAI were 2, 0.18, and 0.215 respectively.



2d) We wrote a generic CV function in R that takes a classifier, such as 'glm', training features, such as 'NDAI', training labels, such as 'expert_labels', number of folds K, and a loss function as in

puts and outputs the CV loss across each K fold and average CV loss across all K folds. The function works for different classification methods: we used it for GLM, LDA, QDA, SVM, and Decision Trees. The function can be found in the "CVgeneric Function" file in Github or in our R code.

3a) We tried using Logistic Regression, LDA, QDA, SVM, and Decision Trees in classifying the expert labels. Common assumptions for classification methods are a large sample size, independent observations, and lack of multicollinearity. The large sample size assumption is satisfied as the amount of data that is collected from the MISR is quite large. In addition, we can assume multicollinearity as our chosen features got rid of any independent variables with very high correlation. The radiance angles, especially AF and AN, had correlations above 0.9, which may have been an issue. The assumption of i.i.d. samples is still an issue, but our approach to splitting the training, test, and validation sets as outlined in 2a were done specifically to combat this issue. Some other assumptions for specific methods were normally distributed data for LDA and QDA, and identical covariance matrices for the classes of LDA. Both of these assumptions are likely not satisfied. Using the CVGeneric function that we created and K = 5 folds, we were able to report the CV errors across each of the 5 folds and the average of the CV errors for the 5 folds for Logistic Regression, LDA, QDA, and Decision Trees. We also found the test errors for SVM and KNN and repeated the process for the other split ratio.

60-20-20 split:

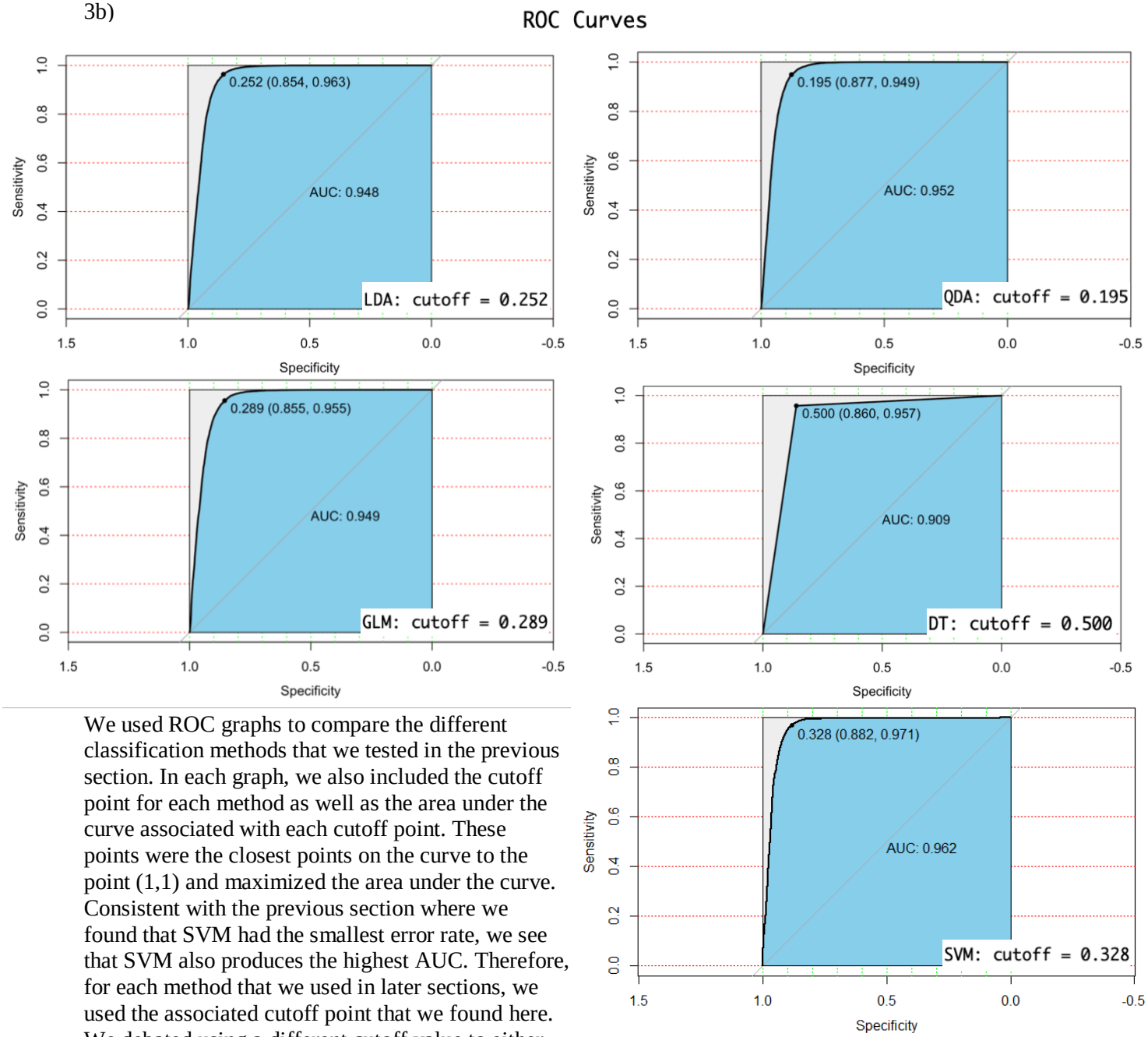
GLM:	CV errors	0.1092520	0.10701793	0.1085611	0.1076900	0.1054703
	Avg. CV error	0.1076306				
	Test Accuracy	0.8933				
LDA:	CV errors	0.1030940	0.1029438	0.1031841	0.1006909	0.1033675
	Avg. CV error	0.1026561				
	Test Accuracy	0.8977218				
QDA:	CV errors	0.1016221	0.1049564	0.1024932	0.1054971	0.1009342
	Avg. CV error	0.1031006				
	Test Accuracy	0.8961598				
Decision Trees:	CV errors	0.09963953	0.10099129	0.10009012	0.09678582	0.09621797
	Avg. CV error	0.09874495				
	Test Accuracy	0.8995482				
SVM:	Test Error	0.08074335				
KNN:	Test Error	0.6682447				

70-15-15 split:

GLM:	CV errors	0.1087614	0.1074892	0.1076336	0.1064179	0.1087079
	Avg. CV error	0.107802				
	Test Accuracy	0.8941651				
LDA:	CV errors	0.1048882	0.1033898	0.1019791	0.1018377	0.1034210
	Avg. CV error	0.1031031				
	Test Accuracy	0.8998045				
QDA:	CV errors	0.1044076	0.1020893	0.1019508	0.1043822	0.1051173
	Avg. CV error	0.1035894				
	Test Accuracy	0.8989394				
Decision Trees:	CV errors	0.09880976	0.09731136	0.10149845	0.09587221	0.09807747
	Avg. CV error	0.09831385				
	Test Accuracy	0.9016309				
SVM:	Test Error	0.08061519				
KNN:	Test Error	0.6702554				

The results of our cross validation and error testing using the different methods showed us that SVM is the most accurate classification method, yielding test errors of 0.08061519 and 0.08074335 for the two split ratios. KNN had the highest error rate, which is most likely the result of using K = 1, as the code too long to run in R. Decision Trees had the second lowest error rate, which is probably because there are less assumptions on probabilistic distribution and other data characteristics that we need for LDA, QDA, and logistic regression. This shows the importance of exploring the data and analyzing whether assumptions are satisfied when choosing to use certain classification methods.

3b)



We used ROC graphs to compare the different classification methods that we tested in the previous section. In each graph, we also included the cutoff point for each method as well as the area under the curve associated with each cutoff point. These points were the closest points on the curve to the point (1,1) and maximized the area under the curve. Consistent with the previous section where we found that SVM had the smallest error rate, we see that SVM also produces the highest AUC. Therefore, for each method that we used in later sections, we used the associated cutoff point that we found here. We debated using a different cutoff value to either get a higher true positive rate (TPR) or lower false positive rate (FPR), but given the nature of our problem, there is no motivation to do either as neither are harmful to our predictions.

3c) From analyzing the errors, we determined that Decision Trees and SVM were both relatively good algorithms for classification. However, an interesting thing that we also noticed was that the ROC Curve for Decision Trees results in a pretty low AUC value. This shows that there may be an issue with the type of classification errors that result from the algorithm and that there may be a very high false positive rate (FPR). Therefore, to analyze the types of errors being made in each algorithm, we decided to run confusion matrices as well. The confusion matrices are printed below.

LDA:

	Reference	
Prediction	0	1
0	17151	1260
1	1989	10810

QDA:

	Reference	
Prediction	0	1
0	17426	1469
1	1714	10601

GLM:

	Reference	
Prediction	0	1
0	17302	1508
1	1838	10562

SVM:

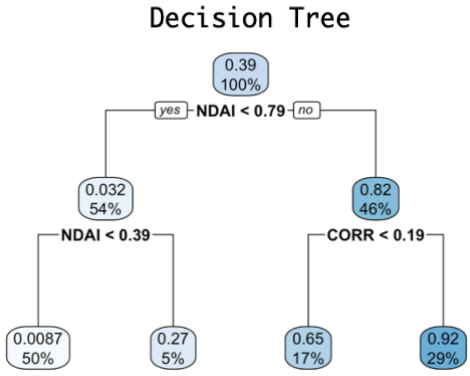
	Reference	
Prediction	0	1
0	17258	682
1	1882	11388

DT:

	Reference	
Prediction	0	1
0	16463	518
1	2677	11552

This showed us that while LDA, QDA, and GLM have higher error rates, they have close to equal false positive and false negative rates so there is not really an issue with misclassifying a certain class. With SVM and Decision Trees, we see that it is much more likely to misclassify a clear pixel as cloudy than to misclassify a cloudy pixel as clear. However, SVM does slightly better than Decision Trees. Moving forward, we decided to keep QDA, SVM, and DT in our analysis. We kept SVM because it provides the method of lowest error, we kept Decision Trees because it provides the second lowest error and so we could analyze more in depth why the false positive rate was so high, and we kept QDA because it provides the best method for roughly equal false positive and false negative error rates.

4a) For this part, we are going to do an in depth analysis of Decision Trees. The reason for choosing this algorithm out of our remaining 3 was because SVM was too computationally complex for R to carry out such in depth analysis and QDA did not have prediction accuracies as high as Decision Trees. In addition, given that the error rate was so low, but there was an unequal distribution of misclassifications, we thought that maybe we could figure out the what was causing the misclassification so that we could improve the algorithm while maintaining a low error rate. We first ran the Decision Trees algorithm on the data and looked at the splits. The splits are mapped out on the decision tree figure to the right along with the mean probability and MSE of being classified as 1 (cloudy). The thresholds for node 1 were NDAI < 0.79, SD < 3.08, and CORR < 0.197. The thresholds for node 2 were NDAI < 0.389, SD < 3.004, CORR < 0.21. The thresholds for node 3 were CORR < 0.189, NDAI < 1.18, and SD < 2.737. We can make a few conclusions from just looking at this decision tree. First, NDAI is the most important feature and primarily used in the algorithm to split the data. In addition, the features, especially NDAI, are much more confident when trying to predict not cloudy pixels. For Node 4 and Node 5, almost all of the pixels split into those nodes would end up being classified as 0. For Node 6, some of the pixels would end up being classified as 0 and some would end up being classified as 1. When it comes to cloudy pixels, NDAI becomes less significant in prediction and the remaining features do not provide a clear means of distinguishing between cloudy and not cloudy. As we found in 3c), the decision tree algorithm does in fact mostly misclassify a 0 (clear) as 1 (cloudy). Therefore, to lower the FPR, it could be beneficial to analyze why NDAI is so useful for clear pixels, but less able to separate cloudy pixels. In addition, maybe we could try to find a different feature that is more useful in predicting cloudy pixels. We wanted to see the distribution of each feature at each individual node in order to analyze why perhaps NDAI was not a good predictor for cloudy pixels or why SD had low feature importance. Therefore, we extracted the observations from each of the nodes and plotted the features for each of the nodes from 4 to 7. The boxplots are shown below. From these plots, we see that NDAI has the largest range of values, which is most likely why it is the most important feature. However, the

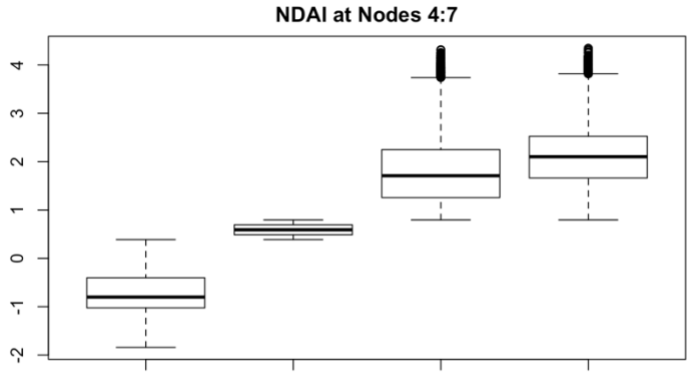


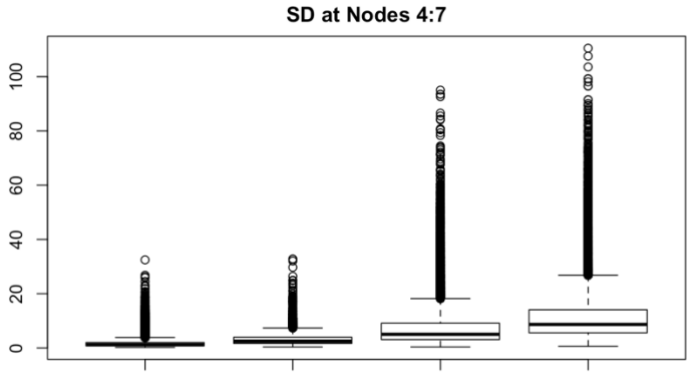
Node number 4: 72100 observations
mean=0.008668516, MSE=0.008593373

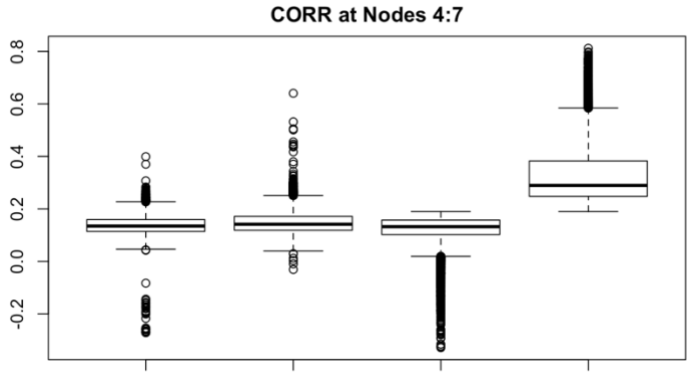
Node number 5: 7203 observations
mean=0.2676663, MSE=0.196021

Node number 6: 24821 observations
mean=0.648604, MSE=0.2279168

Node number 7: 41518 observations
mean=0.9190953, MSE=0.0743591



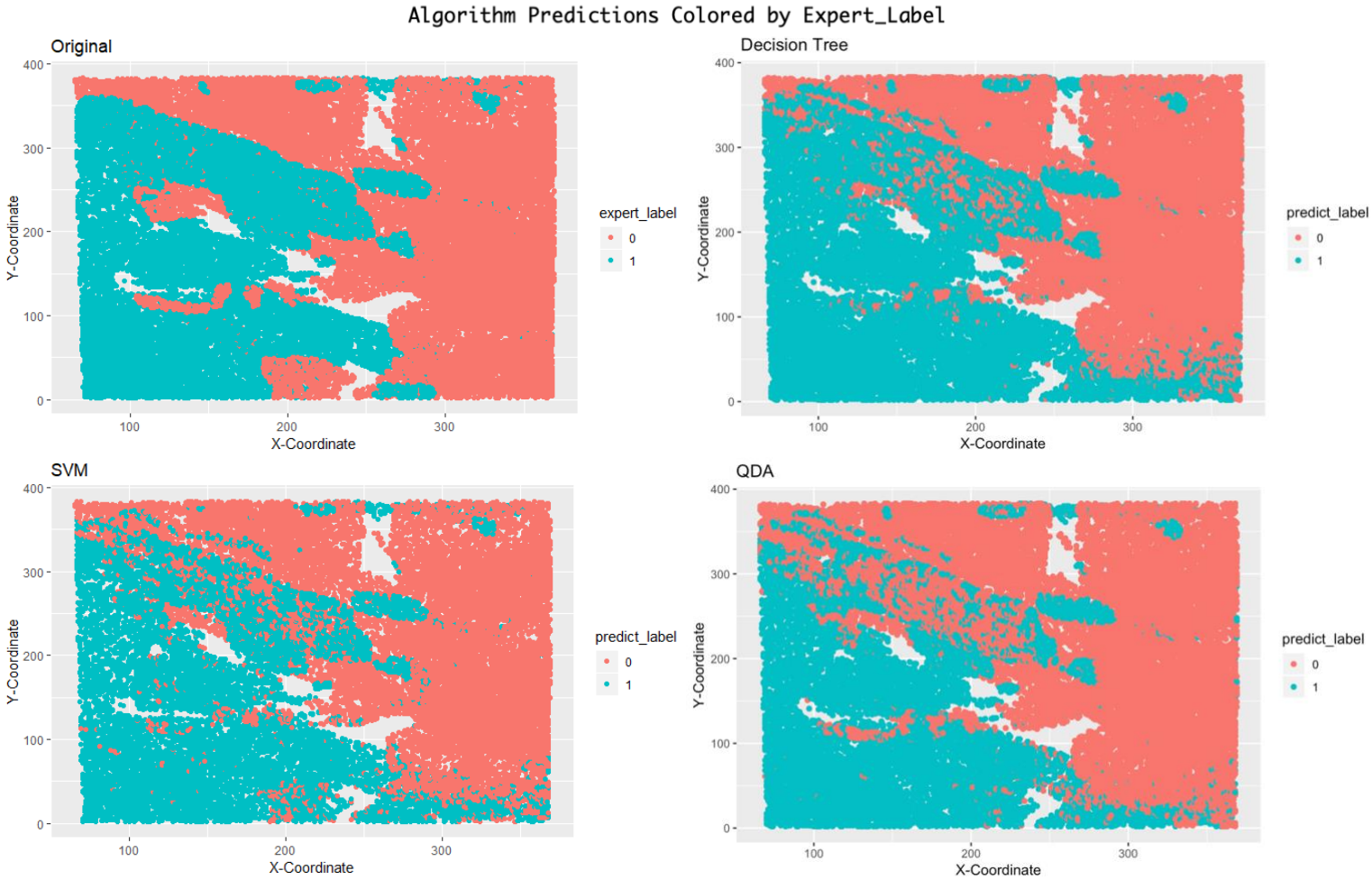




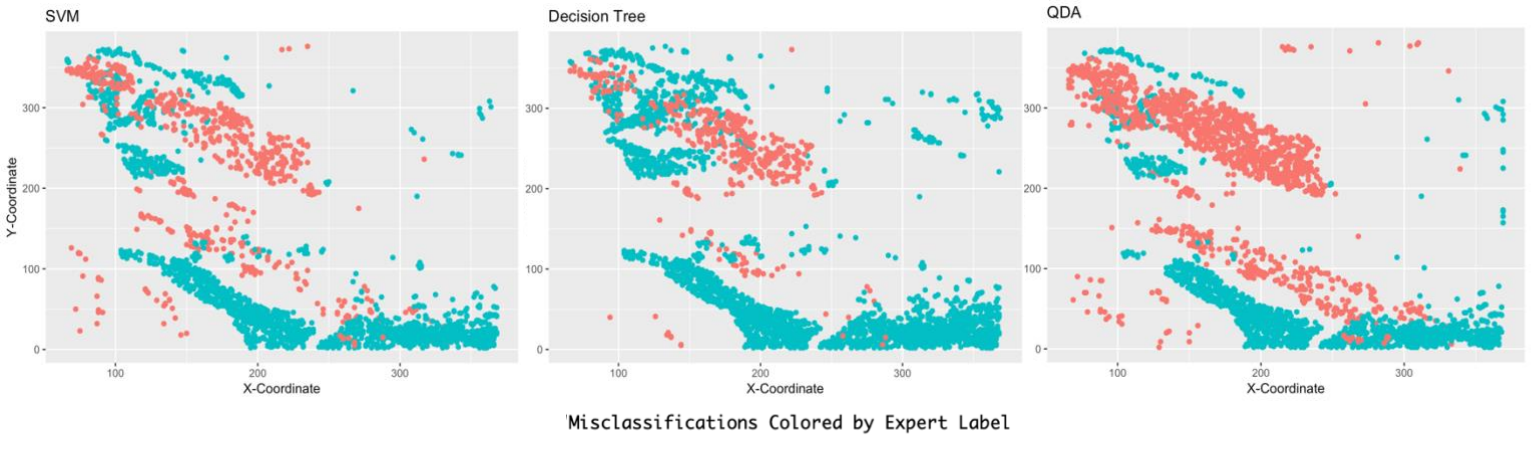
change in mean/range of NDAI values for each node decreases as the probability of being classified as a 1 increases. This leads to 2 possible explanations. First, NDAI only heavily impacts clear pixels. However, given that we already found a high FPR for this algorithm, the more likely explanation is that the range of NDAI values varies widely for clear pixels based on some other factor and then narrows for cloudy pixels to around 2. Therefore, the range of NDAI values a data point that is actually 0 (not cloudy) takes on is actually pretty wide and those with high NDAI values are mistakenly being classified as 1. This shows that NDAI and expert_label may not have a easily differentiable relationship and that NDAI can take on many values for a clear pixel, but only high values for a cloudy pixel. So in order to lower our FPR, we would need to reduce the weight of NDAI or low values of NDAI on the prediction process. We see that SD has a very small range, with the outlier tails causing the primary change in classification. However, we see a distinct increase in means for nodes 6 and 7. Correlation shows an interesting pattern. The 6th node shows a lot of outliers below the mean where the 5th node shows a lot of outliers above the mean. This shows that the algorithm may not completely understand or classify by CORR correctly, or maybe those are the outliers created by separating by NDAI first. We also see that the mean is only significantly different for the 7th node, which explains why CORR is most important for pixels with a very high probability of being cloudy. We can conclude that while NDAI is the most useful feature, NDAI has a pattern that is difficult to separate by expert_label and may be causing the high FPR. The problems from NDAI may be affecting CORR, but CORR is still helpful in correctly predicting cloudy pixels.

We also used the splits to estimate the values of the features. Given that all pixels in Nodes 4 and 5 are classified as 0, the threshold for the values is most likely between the split thresholds in Node 1 and Node 3. This leads to likely thresholds of NDAI between 0.79 and 1.18, CORR between 0.189 and 0.197, and SD between 2.737 and 3.08. We can compare this to the thresholds that we approximated visually in part 2a, which were SD < 2, CORR < 0.18, and NDAI < 0.215, which are comparable. If we take into account the issues brought up with the feature boxplots, the NDAI threshold may be even lower and CORR may change as well. That would bring us even closer to the visual approximation. This analysis showed us that Decision Trees definitely have an issue classifying cloudy pixels and that perhaps this algorithm is not the best at classifying based on NDAI.

4b) We wanted to learn more about the misclassification errors for each of the three methods that we chose to keep. From part 3a, we saw in the confusion matrices that Decision Trees and SVM were both very good at predicting the expert labels. However, they both also had high FPR and low FNR. QDA was not as accurate as SVM or Decision Trees, but the FPR and FNR were roughly equivalent. With that in mind, we created plots of the distributions of predictions compared to the actual distribution of expert labels in our data to see



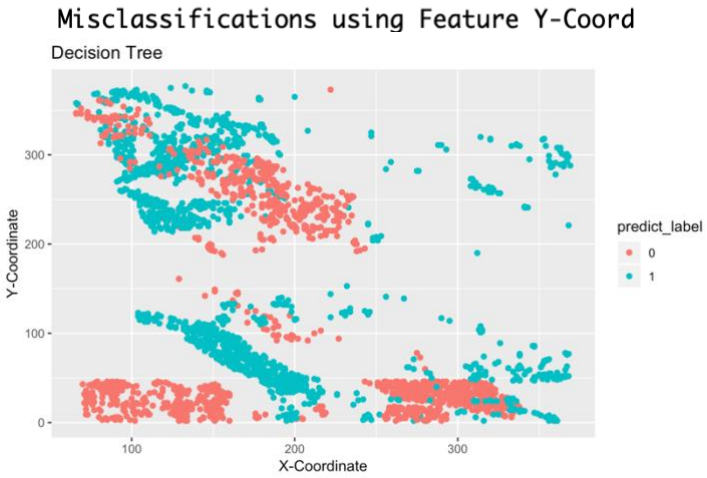
if there was a specific region or area of the data that was more prone to misclassification error. First, looking at Decision Tree, it seems that data with small y- coordinates were almost always misclassified as 1. In addition, clusters of 0's within an area of 1's was commonly misclassified. This is seen in the areas around the coordinate points (150, 225) and (150, 100). We also see that the boundaries between 0 and 1 are commonly misclassified. Looking at SVM, there is less of an error at low y-coordinate values. However there is still pretty substantial misclassification at points with low y- and high x- coordinates. In addition, the boundaries are still misclassified. QDA shows the least amount of misclassification at random areas, but greatest misclassification at the boundaries. We then graphed just the misclassified points, which are shown below. We conclude that all of the methods misclassified as 1 at low y-coordinates and misclassified as 0 in the oval area centered around (150, 250). All the algorithms showed roughly the same amount of misclassification at the low y- coordinates. However, QDA showed much more misclassification in the oval area, which led to equal FPR and FNR for QDA. In addition, Decision Tree misclassified as 1 in the area surround the previously stated oval area, contributing to a slightly higher FPR than SVM.



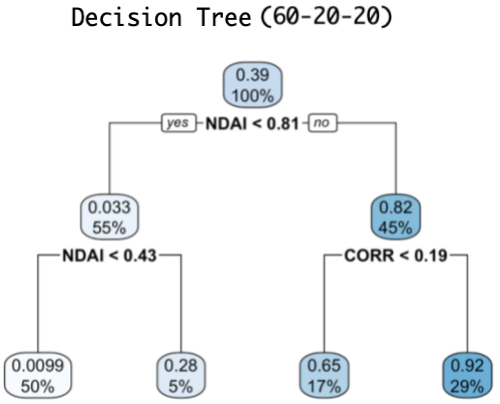
4c) The previous section showed us that there seems to be a relationship between misclassification and certain locations (x- and y- coordinates). Therefore, as a better classifier, it would make sense to utilize the raw information for location in our classification. In addition, as we saw in part 4a), NDAI seems to have a trend that is attributing to misclassification of 0 as 1 and we saw that most of those misclassifications were at low levels of y. Therefore, I added the y-coordinates to the 3 features that I was training on and ran Decision Trees. I found that the accuracy for these features was 91.24%, which is about 1.5% higher for the 70-15-15 split than just running on NDAI, SD, and CORR. An improvement of using the y- coordinate as a feature is best shown in the confusion table and plot of misclassification points. The confusion table not only shows a reduction in number of misclassified points, but the issue of significantly more false positives than false negatives has also been fixed. This shows that there is in fact some kind of trend within at least one of the features (NDAI, SD, CORR) that makes it difficult to classify the data at low y-coordinate values. If we look at the plot, we see that there are much less blue colored points congregated in the lower quarter of the plot. However, this does bring up the issue as to why we now have more false negatives in the same area of the plot. For the purpose of improving our algorithm and addressing a key problem that we faced, using the features of NDAI, SD, CORR, and the y-coordinate does help. However, there still seems to be a problem. For future analyses, it may be helpful to create a new classifier that scales NDAI on the y-coordinate or interacts the two variables together. It may also be helpful to focus on the boundaries where most of the misclassifications occur and try to tease apart the differences between the classes. That way, we can figure out which features are the most important and how they can interact to best predict what classification a data point will be. In addition, as we found that SVM actually predicts with the highest accuracy, it would be helpful to do further in depth analysis using SVM. For this project and using R, the computational time was too long, so Decision Trees was a more efficient choice. For future data without expert labels, we think that our model will still work well. We would need to change some of our methods as they are binary classification models. However, is we can utilize the same methods and output probability predictions, we would just need to establish a clear cutoff threshold to classify as cloudy or not cloudy. For example, with logistic regression, the result of running regression is probabilities. If we use the model that we already fit to the training data with expert labels, it would be very simple to run the model on new data and generate probabilities. We can then use the cutoff values from the ROC curves we have already plotted for each method as the threshold of classification.

Confusion Matrix using Feature Y-Coord

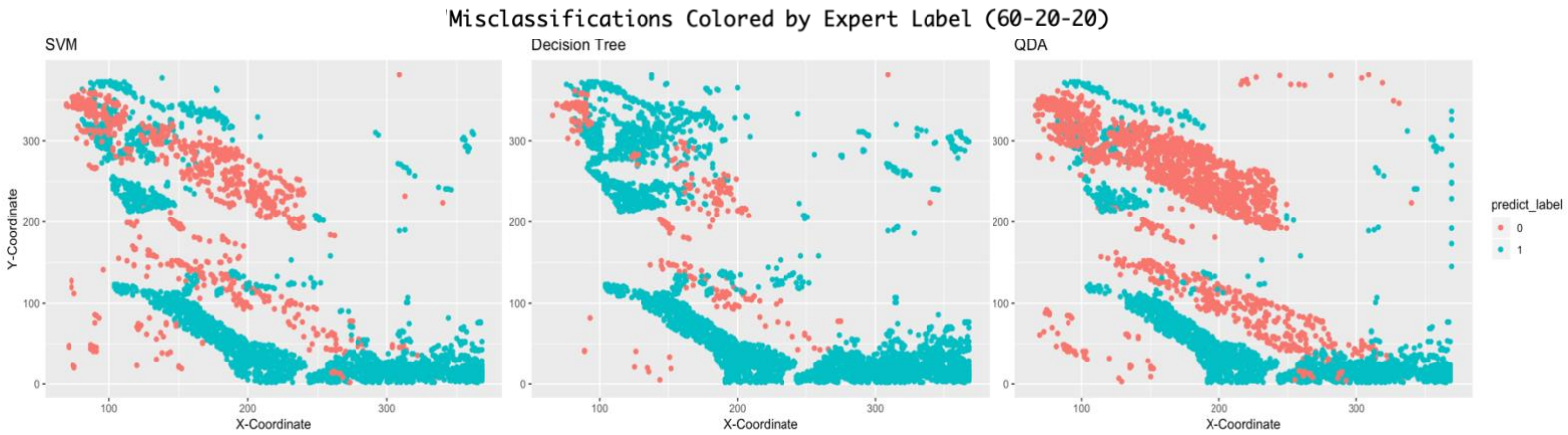
Prediction	Reference 0	1
0	17641	1234
1	1499	10836



4d) We reran the code for parts 4a) and found that splitting by 60-20-20, our other proposed splitting ratio, the split does not change significantly. If we look at the Decision Tree for this split, we see that NDAI and CORR are still the primary split features. In addition, the threshold values of the split change by less than 0.04. It may be useful to note that the NDAI threshold increased as did the minimum mean probability. We also plotted the individual features for each node, but they looked unchanged from the 70-15-15 split, so we won't include them again in this report. Changing the split ratio from 70-15-15 to 60-20-20 caused little variation in the results of our algorithm, so the conclusions that we drew in part a) regarding Decision Trees likely hold for both splits.



We reran the code for part 4b) and found some differences when we changed our split ratio to 60-20-20. Below are the confusion matrices and plots of the misclassified data points for our 3 algorithms.



Confusion Matrices (60-20-20)

SVM:

Prediction	Reference 0	1
0	23001	924
1	2481	15207

DT:

Prediction	Reference 0	1
0	16722	285
1	3463	15394

QDA:

Prediction	Reference 0	1
0	23247	2043
1	2235	14088

What we find is that the misclassifications follow the same trends as before regarding coordinates. However, an interesting new trend is that for QDA, both FPR and FNR increased by about 500 data points. For SVM, FPR increased by 500 while FNR only increased by 250. And for Decision Trees, FPR increased by about 800 while FNR decreased by about 250. This provides some insight about the split ratio that is used. As we saw in part 3a), the average CV error and test accuracy did not change that much between the two split methods – this is confirmed as the changes of about 500 data point s are a very small percentage of the data set we used. However, there appears to be a significant change in the type of error that is created. QDA maintains a roughly equal FPR and FNR. However, FPR increases at a higher rate than FNR for both SVM and Decision Trees. This shows that the ratio you choose to split your data with may not affect the overall accuracy of the algorithm. However, it may change the type of errors you end up with. In this case, it does not matter too much to us what kind of error we get as we just want to maximize accuracy. But for other situations where a false positive or false negative may be harmful, it would be important to choose a split and algorithm that would minimize the harmful error.

4e) This project has presented us with two different classification methods that could potentially be extremely accurate in classifying cloud data collected from the MISR satellites: Decision Trees and SVM. We have shown that while Logistic Regression, LDA, and QDA are good methods of prediction, they require assumptions such as normality or independence that we are not be able to satisfy with our data. We have also shown that CORR, SD, and NDAI are good features for the algorithms that we used and are sufficient in separating between cloudy and clear (snow/ice covered) pixels. However, examination of the features themselves have also showed that there may be some relationship between x- and y- coordinates that cause misclassification at specific areas such as low y values, which may also be a factor causing different false positive and false negative rates in our SVM and Decision Tree algorithms. We confirmed this by showing that using the y- coordinates as a feature decreased the amount of false positives in our data when using Decision Trees. However, there is still further analysis that can be done and even questions that pop up as a result of our work here. Some possible future projects could delve into SVM: how SVM utilizes the three features in the algorithm or if using the raw y-coordinates in some way as a feature improve FPR. In addition, we could do more work on picking even better features and finding ways to scale or condition some

of our current 3 features on the y-coordinates and see if that improves the FPR for our 2 best algorithms. Overall, this has been sufficient in proposing a good algorithm to classify the datasets, but still more can be done to increase prediction accuracy.

5) Github Repository Link: https://github.com/ElvisYang1998/project2

Acknowledgements: We both worked extremely hard on this project and carried equal weights. Much of the misclassification graphs, ROC curves, and CVgeneric was done by Chenyu and more of the decision tree analysis, confusion matrices, and writing was done by Angela. However as a whole, we both contributed to each part of this project in some way. The way we proceeded with this project was to first read and take ideas from the article and then when it came time to prepare and model the data ourselves, it was a procedure of trying many different classification methods and using what we thought was best from there. In addition, a lot of sections built on the work from previous sections, so with the more that we did, the better we understood the data and the algorithms that we were using in order to better our methods and improve accuracy.