**Project 2 Cloud Data**

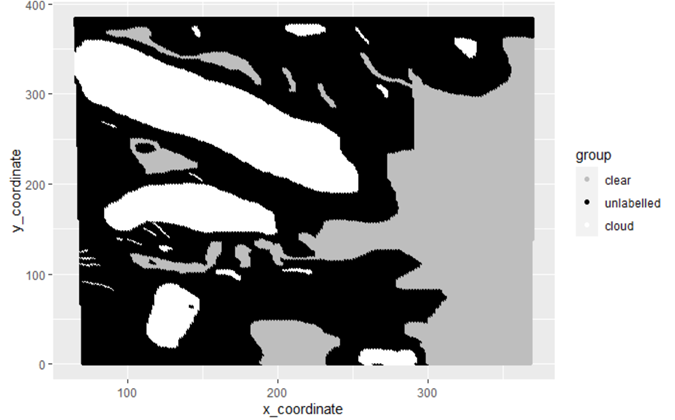
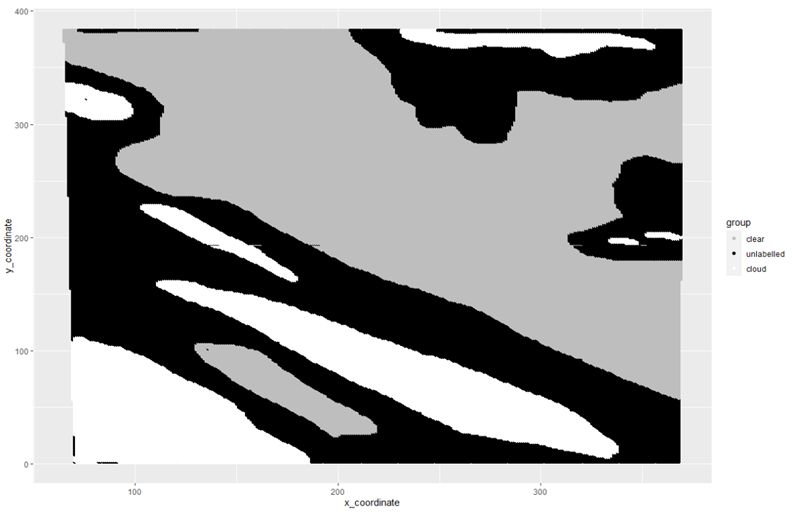
Xuzhang Li and Yang Cheng

gavin.li@duke.edu, yang.cheng@duke.edu

**Data exploration and collection**

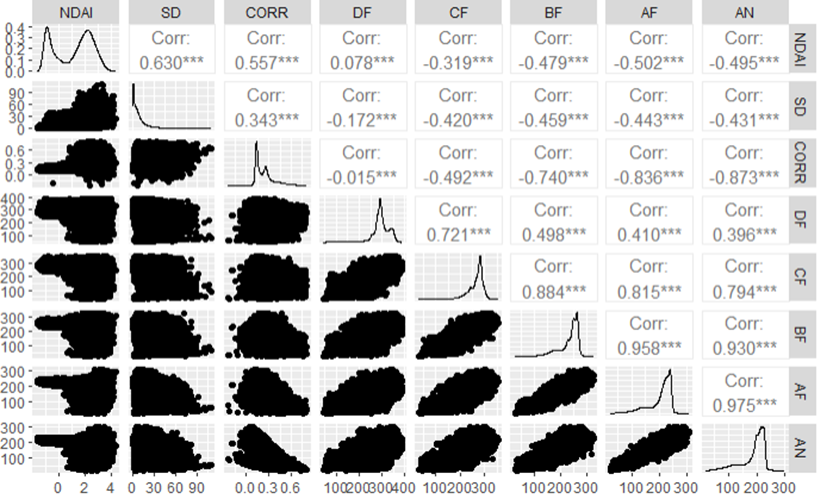
The paper written by Yu et al(2008) discusses a new statistical algorithm they developed to better classify the presence of clouds. Cloud coverage around the Arctic area plays an important role in determining the sensitivity of the Arctic to increasing air temperature. However, past algorithms such as L2TC had trouble in giving a precise prediction of clouds due to the difficulties in some of the cloudy related measurements. As an improvement to the old method, the author of the paper proposes an algorithm that predicts using constructed physical features.

The data in this study comes from 10 Multi Angle Imaging SpectroRadiometer (MISR) orbits of path 26 over the Arctic. The data spans 144 days from April 28 through September 19, 2002. Six data units from each orbit are included in the study. Three out of 60 data units are excluded because the MISR algorithm could already detect cloudy well. The data are collected through MISR cameras from 9 different angles. The data are high-resolution pixels with radiation measurement. The enhanced linear correlation matching (ELCM) algorithm proposed by author construct three features using EDA and domain knowledge. Then they make prediction by setting thresholds for each feature. They also use QDA to predict the probability of cloudiness for partly clouded units. Their algorithm achieved a 91.80% agreement with the labels given by experts and a full coverage rate. Older MISR operational algorithms such as stereo-derived cloud mask (SDCM) and the angular signature cloud mask (ASCM) gives an agreement of 80.00% and 83.23% respectively as well as a lower coverage rate. A simple offline SVM has 80.99% accuracy and full coverage rate. In short, this algorithm outperforms traditional method. This study proposes a new algorithm with higher prediction performance. It helps the scientific community in better cloud detection and helps the study of global climate science. Moreover, it is also significant for statistics community. It shows the potential of how statisticians can contribute to climate studies by helping with design a study. It also demonstrates how statistical thinking can benefit other disciplines. For image1, there are 37.3% of pixels belong to class -1(not cloud), 28.6% of pixels belong to class 0(unlabeled) and 34.1% of pixels belong to class 1 (cloud). For image2, there are 43.8% of pixels belong to class -1(not cloud), 38.5% of pixels belong to class 0 (unlabeled) and 17.8% of pixels belong to class 1(cloud). For image3, there are 29.3% of pixels belong to class -1(not cloud), 52.3% of pixels belong to class 0 (unlabeled) and 18.5% of pixels belong to class 1(cloud).



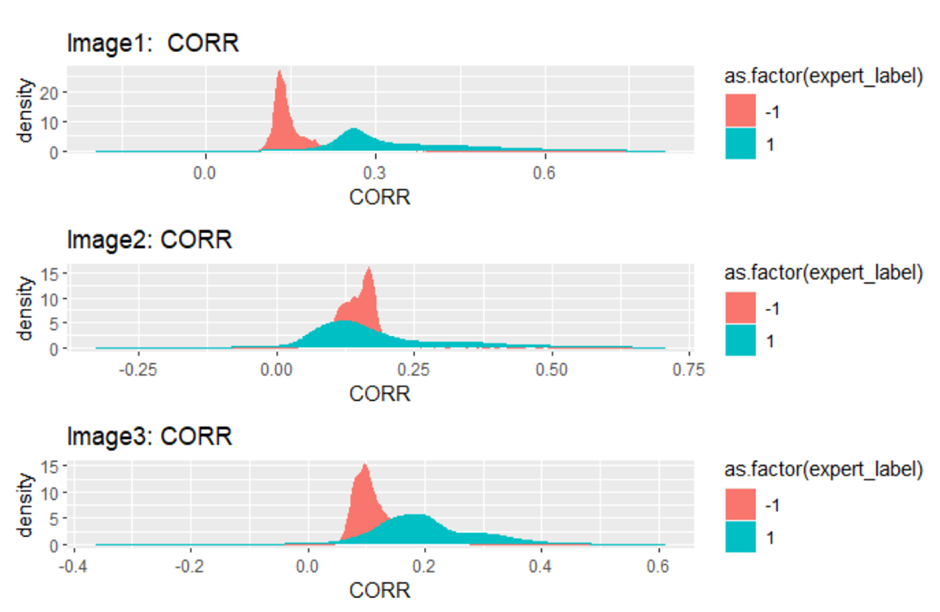
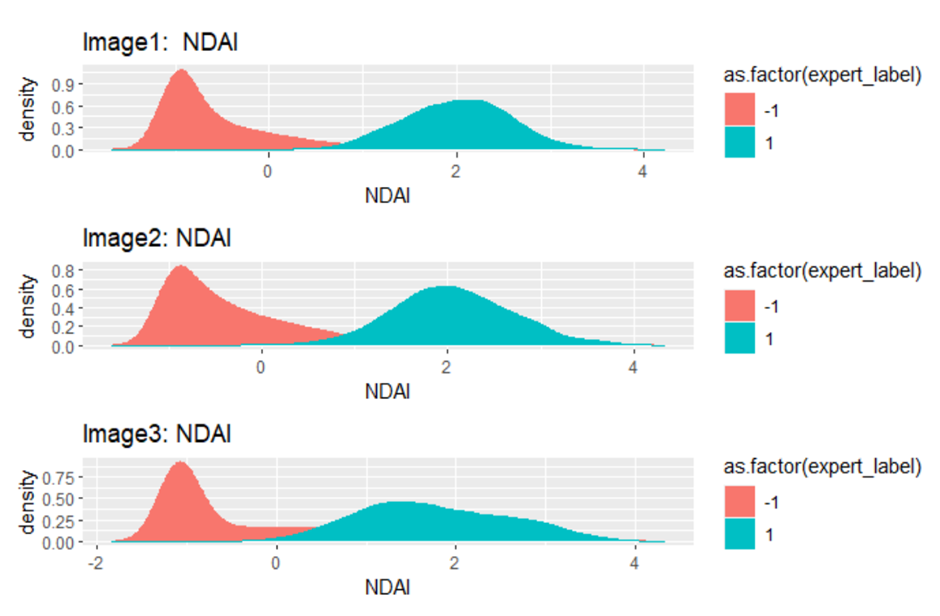
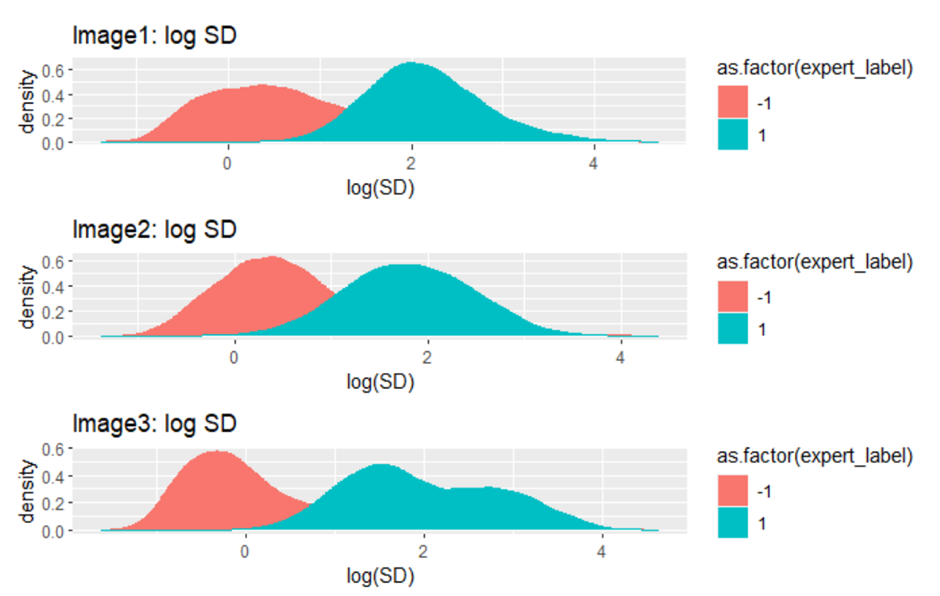
**Figure 1: Maps for image1, image2 and image3**

For all three images, we observe that different colors are presented together in chunk. In other words, it is more likely that cloudy particles are surrounded by other cloudy particles while areas free of cloud come hand in hand. If we make an i.i.d assumption, then each datapoint is completely independent of its neighbor. We should be expecting random dots scattered over the graph. This is clearly not what we observe hence the i.i.d assumption is wrong.



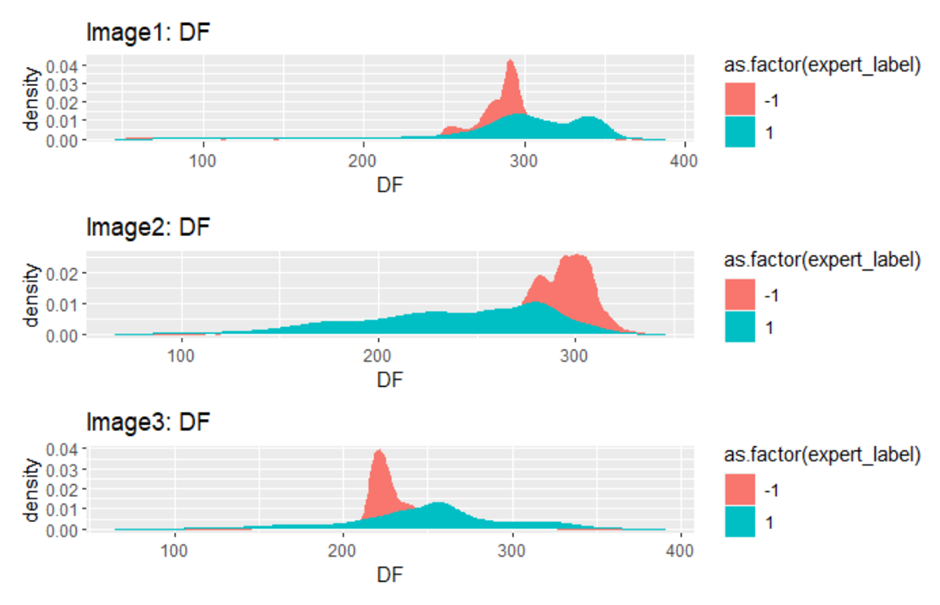
**Figure 2: Pairwise Analysis for image1**

We now do a pairwise analysis of all the features. Due to space issue, we only include the plot of image1. The pairwise relationship for the other two images is similar. We can see from the pairwise plot, the three features of interest NDAI, SD, and CORR all have a pairwise positive correlation. This means when one of the features increase, we expect other features to increase as well. The relationship between those three features and the measurement of radiance angle are negatively correlated. Moreover, the correlation between radiance angles themselves are positive. This makes sense as we should see the angles of different cameras moving in the same direction.



**Figure 3 : Density of three features for different classes**

We now analyze the pariwise relationship between the label and features. We remove the unlabelled data as they won’t be helpful. For the SD feature, we did a log transformation as it has a heavy tail. For all three images, the distribution of log(SD) and NDAI are similar. The distribution of class cloudy has a higher value. This finding is also supported by the paper whhich suggests that they will predict clear if SD is smaller than threshold value or NDAI is smaller than threshold. For CORR, the distribution are not quite similar. This is not suprising as the paper pointed out that high CORR value suggest either clear or presence of some low altitude cloud. This is also the reason why the paper suggests three features for their model since CORR itself is not good enough.



**Figure 4: Density of DF for different classes**

We also did a pairwise plot of density of DF with respect to different labels. Unlike the other three features, the distribution of DF for different images show no similarties. This gives us some idea that radiance angles might not be that helpful in predicting the label of a pixel. We will further explore which feature are important in making prediciton in following sections.

**Preparation**

To split the entire data, including all three data sets (the three images), imagem1.txt, imagem2.txt, and imagem3.txt, into training, validation, and testing sets, it is important to note that simple i.i.d method of separation would not be optimal due to the strong spatial correlation of the data. In order to preserve the spatial correlation of the data, block cv would be efficient as it maintains the spatial structure of the data and the data points in each block would be roughly more similar to each other, and ensure that in most cases a pixel would be in the same block as its close neighbors, so the spatial integrity of the data is preserved.

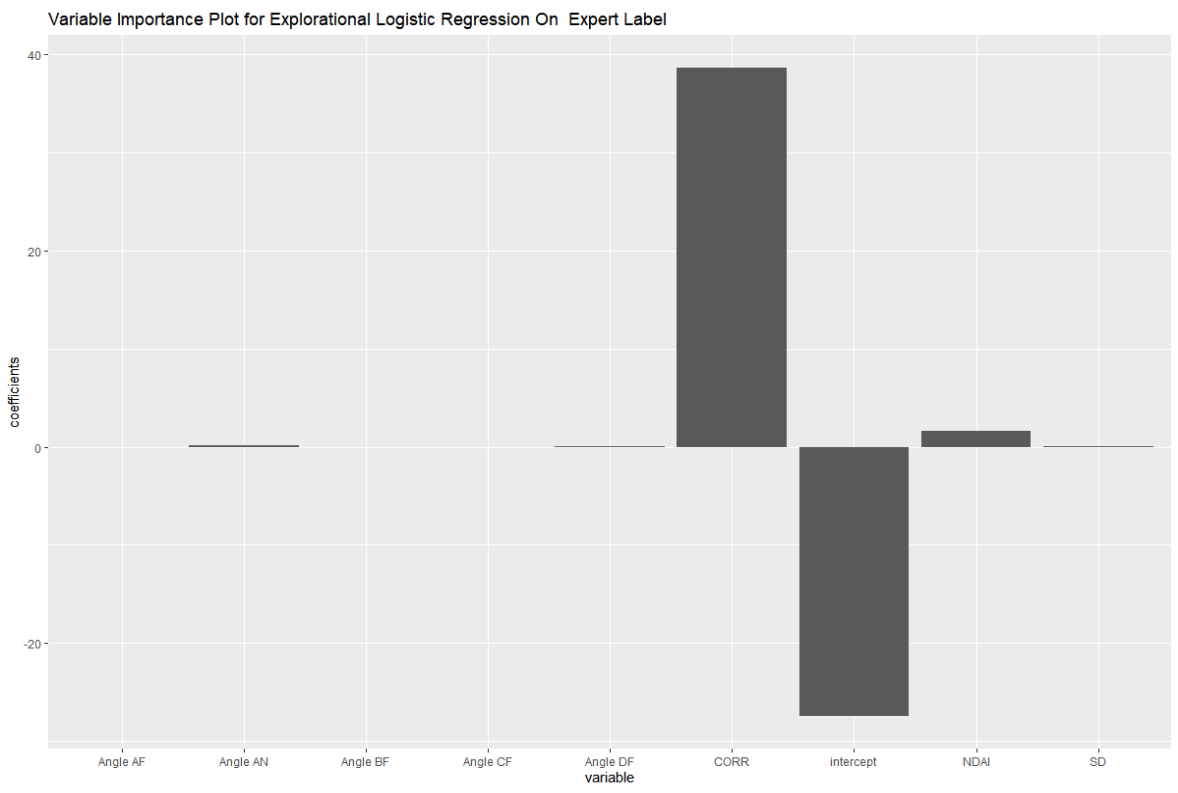
Since the three datasets represent three different images, the most natural and standard way of splitting the data with block cv would be to cut each of the images (data sets) into blocks, and randomly put the blocks into training, validation, and testing sets. Specifically, each of the three images are split into 16 roughly even rectangular blocks, with 12 of the blocks randomly selected to be the training set, 2 of the other 4 blocks randomly selected to be the testing set, and the rest 2 blocks selected to be the testing set. With this random block selection done to each of the three images, there would be a training set containing data from 36 blocks (12 from each of the images), a validation set and a testing set each containing 6 blocks (2 from each of the images), with the separation of blocks completely random.

An alternative method of data splitting would also use the block cv technique as spatial correlation is definitely needed to be preserved in this analysis. Instead of randomly select 12 blocks, 2 blocks, and 2 blocks into the training, validation, and testing set respectively from each of the images, it also makes sense to simply split each image into 64 roughly even blocks, mix up all 192 blocks, and then randomly select 144 blocks as the training set, 24 blocks as the validation set, and 24 blocks as the testing set. This way, the proportion of data in each set would be more random and as it is likely not to be evenly distributed among the images. As the number of blocks were also increased, the randomization was further strengthened and closer to simple i.i.d, while the spatial relation of the data is captured to a lesser extent. Also as the three images might be inherently different in their properties, this methods contains the risk of separating too much of a specific image into a certain set and produce a rather skewed result.

For some of the following analysis, the first method of data separation is used. Also note that data points without an expert label (i.e. with an unlabeled expert model) are removed. They provide no valuable information to the model and cross-validation as classification success and failure rate all depends o the existence of an expert label. Note that this removed 137495 rows out of 345556 rows, so a significant portion of the data was removed due to unlabeled expert labels.

The trivial classifier was introduced which simply sets all classified labels to be -1, which represent cloud-free, on the validation and testing set. The accuracy of this trivial classifier is also trivial in some sense as it simply captures the proportion of -1 expert labels (the proportion of cloud-free expert labels) in the validation set and the testing set. Again, as noted previously, all 137495 data points with unlabeled expert labels were removed as they contribute no valuable information to model construction and cross-validation.

In the case of this trivial classifier, the validation set classification accuracy is 0.632, and the test set classification accuracy is 0.784. When a certain label makes up a large proportion of data, these kinds of trivial classifiers would actually have high average accuracy. In this case, this trivial classifier provides rather moderately high classification accuracy, and serves as a baseline of classification accuracy that the cross-validation trained models need to improve upon. This kind of trivial classifier baseline is important as sometimes even a high 95% model classification rate would be weak if a trivial classifier can produce similar accuracy.



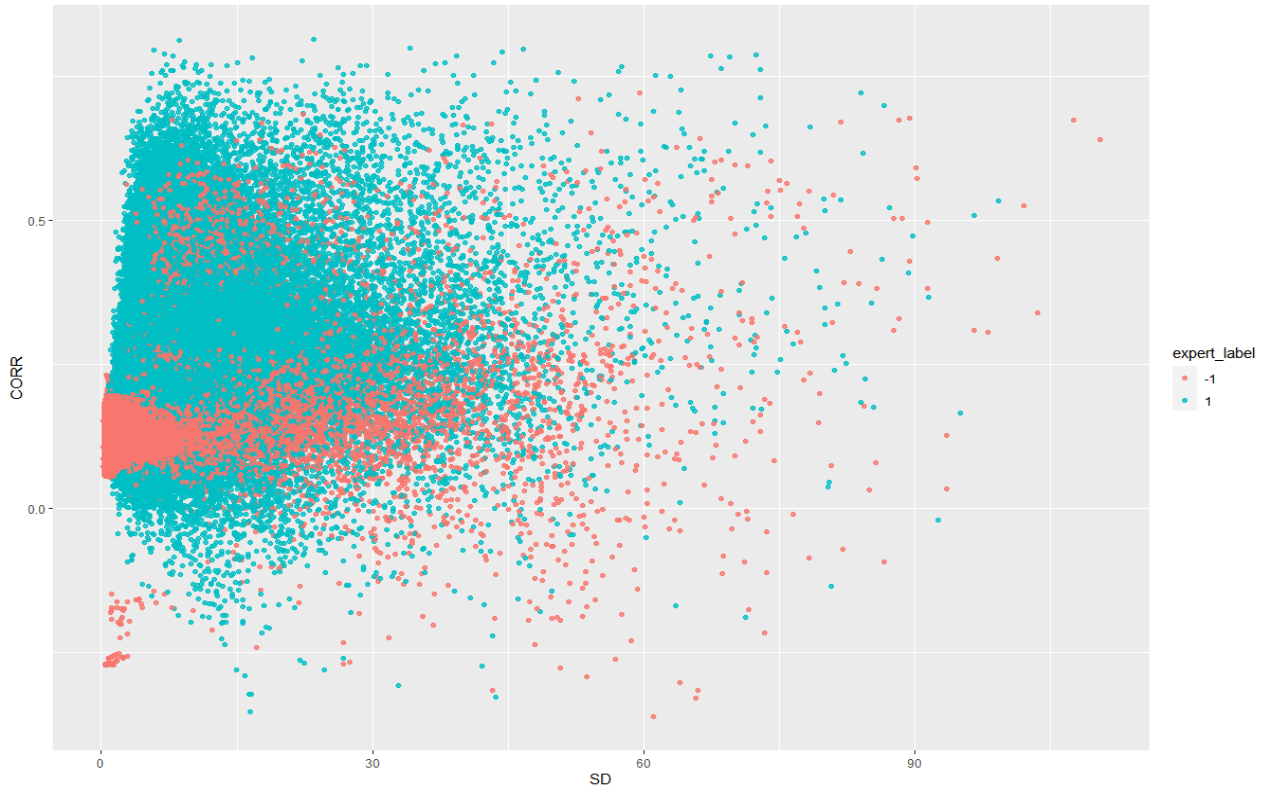
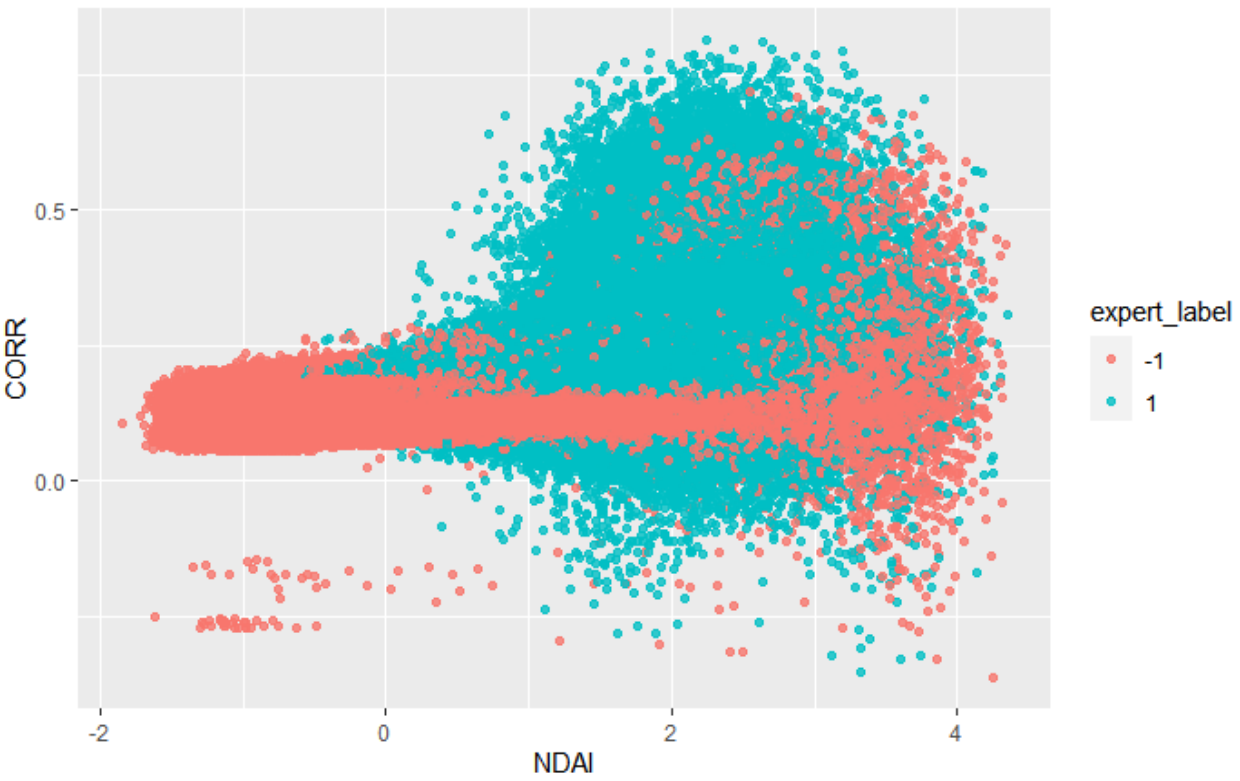
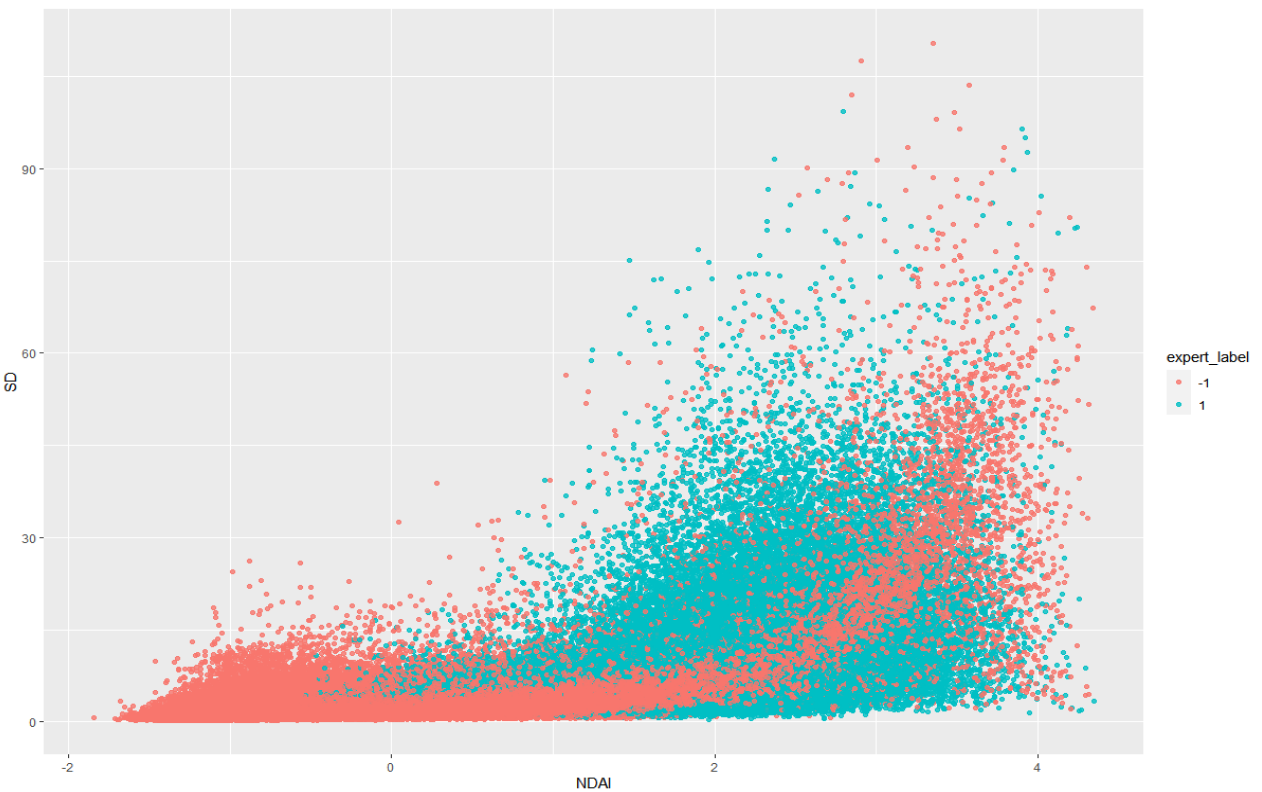
**Figure 5: variable importance plot for naive logistic regression on expert label**

In order to select three of the “best” features, which are the features that can predict the expert label most efficiently and accurately, a simple and naive logistic regression was run that uses all other variables to predict the expert label. A variable importance plot was shown for this logistic regression result, and it is clear that V4 (NDAI) and V6 (CORR) are the two most significant predictor variables for the expert label V3, while the other variables seem much less significant. The variable importance of these two variables are 38.7 and 1.6 while all other variable importance have absolute values less than 0.2. In combination with evidence in the paper, it would only make sense to always include these two variables in all of the following models and analysis.

According to the EDA done to all the variables(see figures previous pages), there is a clear difference in NDAI and SD for cloud and cloud-free groups of the data, and it can be inferred that NDAI and SD would be efficient predictors variables for the expert label, which is also consistent with the research as well.

Based on a PCA analysis, the first principal component contains roughly 67% of the variability of the data set, with (-0.31, -0.28, - 0.27, 0.26, 0.38, 0.42, 0.43, 0.42). With each variable having a coefficient within 0.28-0.43, the weight is roughly even and there isn't much to conclude from the PCA that would add to the logistic regression. Note that the first three principal components included approximately 93% of the variabilities and the second and third components also have similar weights across the variables, providing little extra information.

Based on all of the above information, to select the most efficient and reasonable predictor variables for the expert label, NDAI, SD, and CORR are the three most desirable predictors. This selection of variables is also consistent with the paper as none of the radiance angles are particularly special and unique, and thus it would make little sense to include some radiance angles while excluding the others in the models.



**Figure 6: Pairwise scatterplots of the selected best variables**

To further visualize the effectiveness of the selected NDAI, SD, and CORR predictors, pairwise scatter-plots are shown below with the expert label as a factor, and vague cut-off and patterns could be observed within each of these plots.

After selecting the “best” features, these features are applied to classification methods and cross-validation. In order to try different classification models, a general CVmaster function was written that can take in different classification methods and fold count k and perform k-fold cross validation based on the two previously specified block cv data splitting rules. This function can be accessed on github.

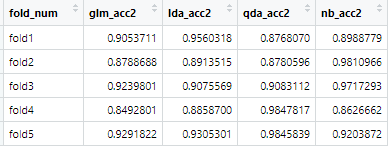
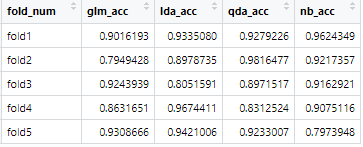
**Modeling**

To select the best classification method, logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and naive bayesian methods were implemented and put into the previously mentioned CVmaster function to perform k-fold cross validation. According to the normal standard, and for simplicity purposes, the number of folds is set to 5.

Note that all of these different classification methods have different assumptions, and not all assumptions are necessarily satisfied in this study. For logistic regression, it assumes binary response variables, independence between the observations, and no multicollinearity among the independent variables. While the response variable is certainly linear and the observations can be considered as independent, it is unclear that the multicollinearity assumption is satisfied or not. However, according to the previous figure that shows pairwise scatter-plot for the three selected variables, there should be only a little multicollinearity.

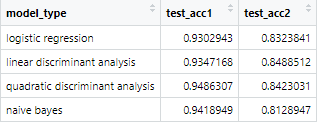
Both LDA and QDA assume that the observations are drawn from a normal distribution. While LDA assumes that feature covariance matrices are the same for each class, QDA allows each feature covariance matrix for different classes to differ. According to the exploratory data analysis in part 1, the feature variables, especially the three selected variables, NDAI, SD, and CORR, do seem to follow roughly normal distributions. Additionally, while it is unclear whether each class would have a different covariance matrices, by the shape of the NDAI, SD, and CORR density plots with respect to the expert label, it would seem that it is unsafe to take on the LDA assumption of shared covariance matrix across the two classes, but a rough guess is that the assumption would not be too far off. For QDA its assumptions are satisfied as different covariance matrices across classes are allowed.

On the other hand, naive bayes assume conditional independence between the expert label and the features, so that the expert label is only affected by the three variables independently. This is very likely a wrong assumption in this case, as the paper clearly states that the expert label is affected by a combination of NDAI, SD, and CORR. However, the pairwise scatter-plots do not show a very clear and strong pattern that would completely prove conditional independence assumption to be false, and thus while the naive bayes model cannot be trusted due to false assumptions, it would be interesting to observe its performance nonetheless.

****

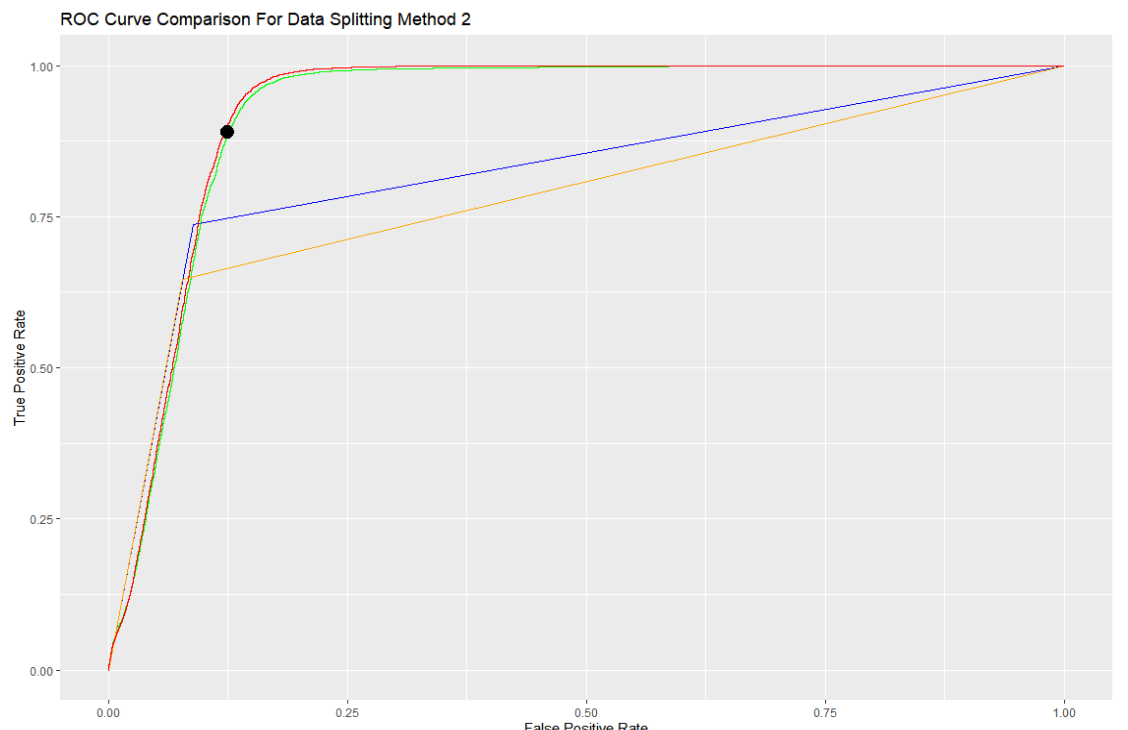
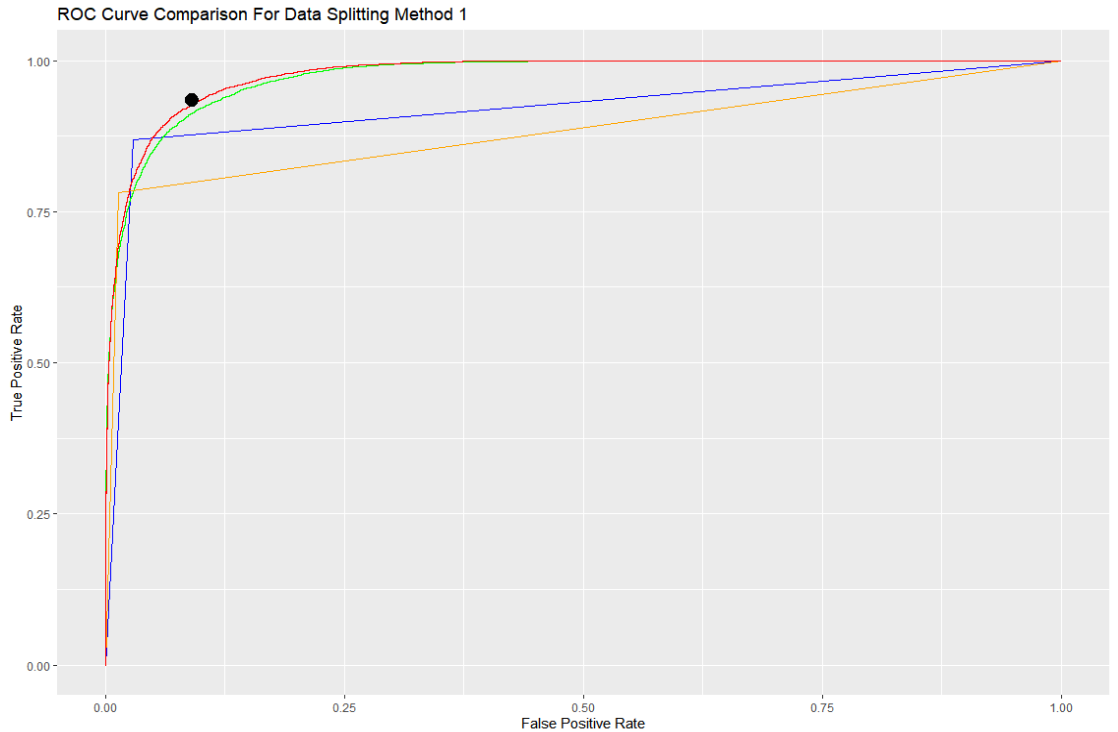
**Figure 7: fold accuracies in 5-fold cross validation with logistic regression, LDA, QDA, naive bayes, for the two data splitting methods respectively**

The figure above shows the accuracies of each fold, for each classification algorithm, and for both of the data splitting methods introduced in the very beginning of the data preparation part. The left hand table is the cv accuracies for the first data splitting method, and the right hand is for the second data splitting method. Note that all of the classification methods performed generally well, and the average fold accuracy for all the classification methods and the data splitting methods are all within 85%-95%, which is significantly higher than the accuracy of the trivial classifier, which was only 0.63.

****

**Figure 8: test accuracies for logistic, LDA, QDA, bayes and the two data splitting methods**

The figure above shows the test accuracies for all of the classification methods and the two data splitting methods. Note that while the classification methods do not differ much in test accuracies, the test accuracies for the first and default data splitting method is significantly higher, while the cv fold accuracies are very similar for the two data splitting methods. This could be caused by overfitting for the second data splitting method, or the second method splitting the data into too many pieces, breaking the important spatial relation of the data too much.

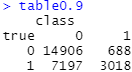
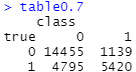
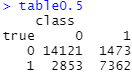
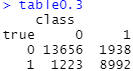
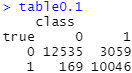
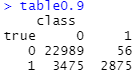
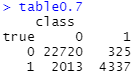
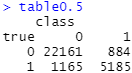
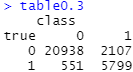
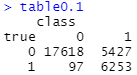


**Figure 9: ROC curves for all of the models, and for the two data splitting methods respectively. Green: logistic regression, Red: LDA, Blue: QDA, Yellow: Naive Bayes**

ROC curves were plotted for the purpose of model comparison. Both data splitting methods were considered, and the shape of the graph is slightly different between the two data splitting methods. The first data splitting method displays a generally better ROC curve in that it is more close to the ideal curve and further away from the dummy x=y curve. Note that since it makes less sense to define a clear cut-off value for QDA and naive bayes, their ROC curve is somewhat strange and would be less credible. The ROC curves for logistic regression and LDA are surprisingly similar, and thus it is hard to clearly determine which is better.

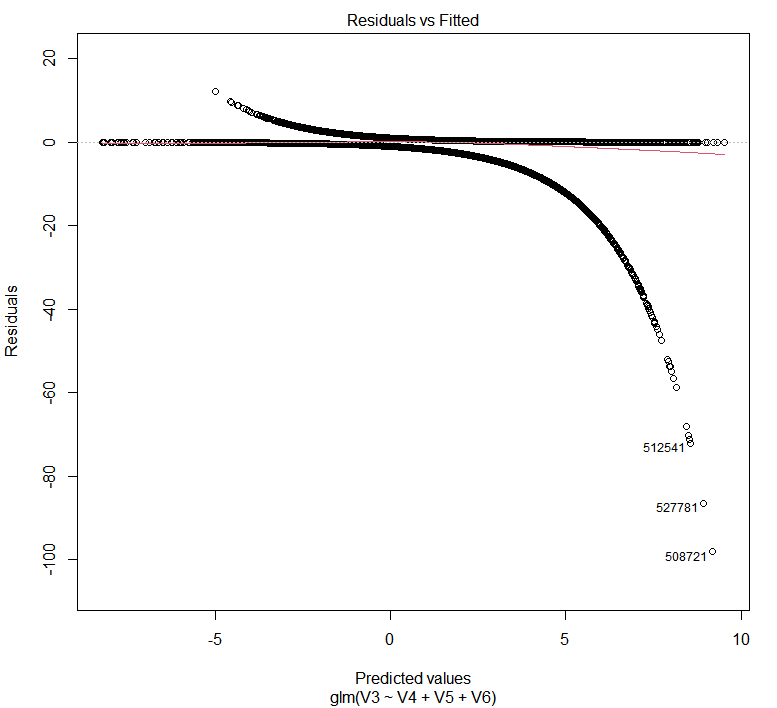
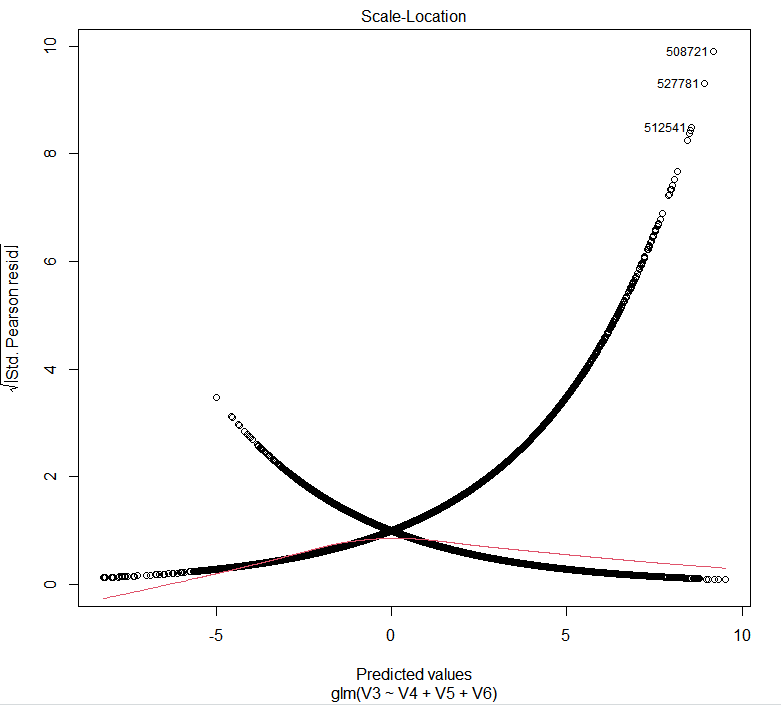
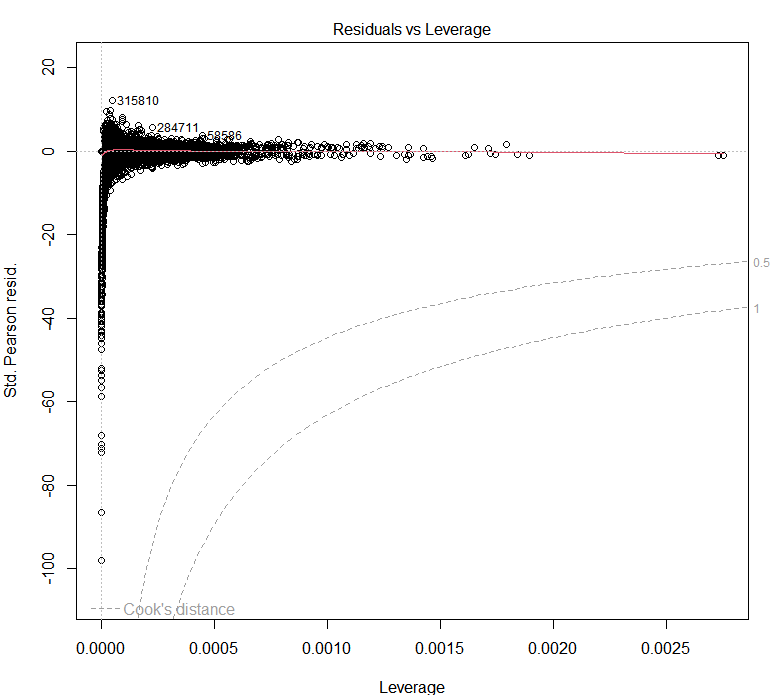
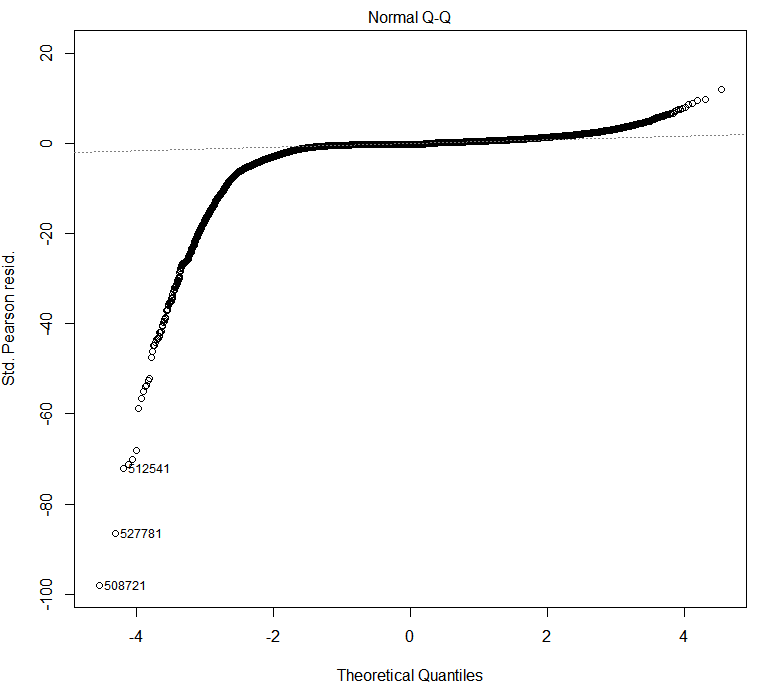
The black dots in the figures indicate the cut-off values for the logistic regression models. A cutoff value of 0.1, 0.3, 0.5, 0.7, and 0.9 was tested on both data splitting methods for the logistic model, and 0.3 is selected to be the cut-off due to a good balance between maximizing the true-positive rate while minimizing the false positive rate. As shown in the ROC curve above, the cut-off point is close to the tip of the curve and is close to the up, left corner, which would be the ideal placement.

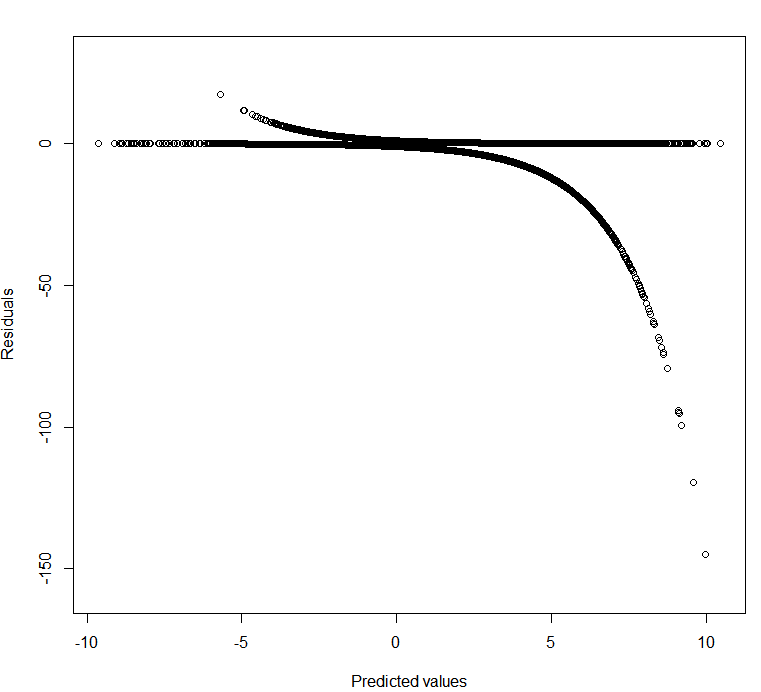
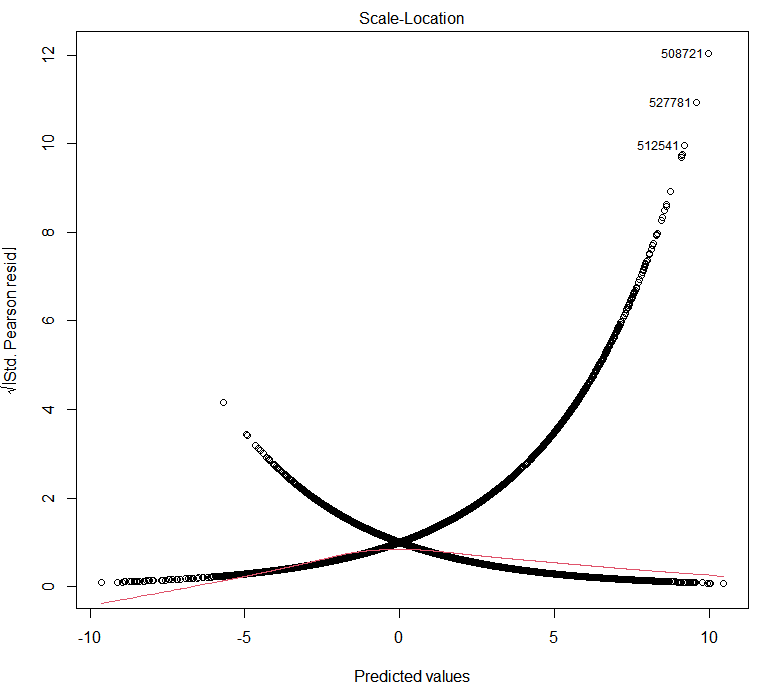
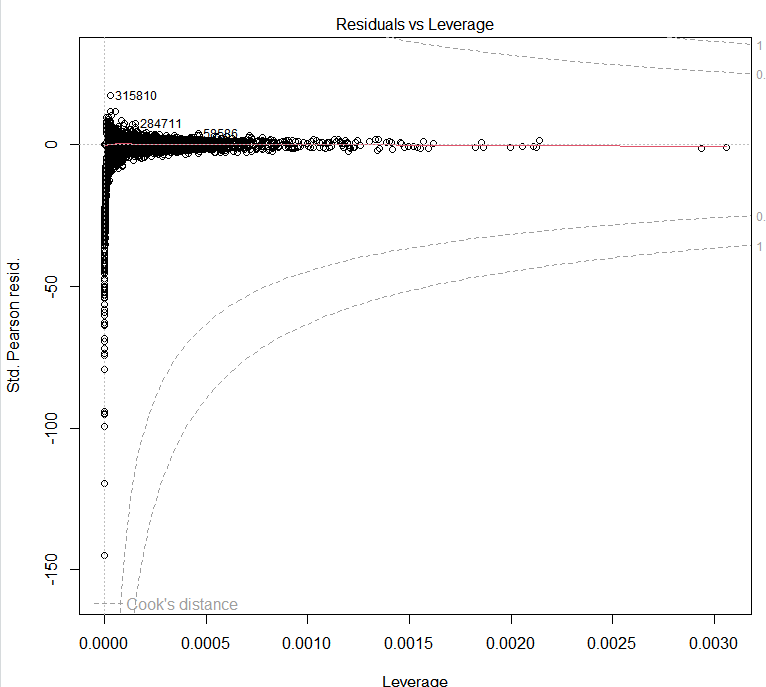
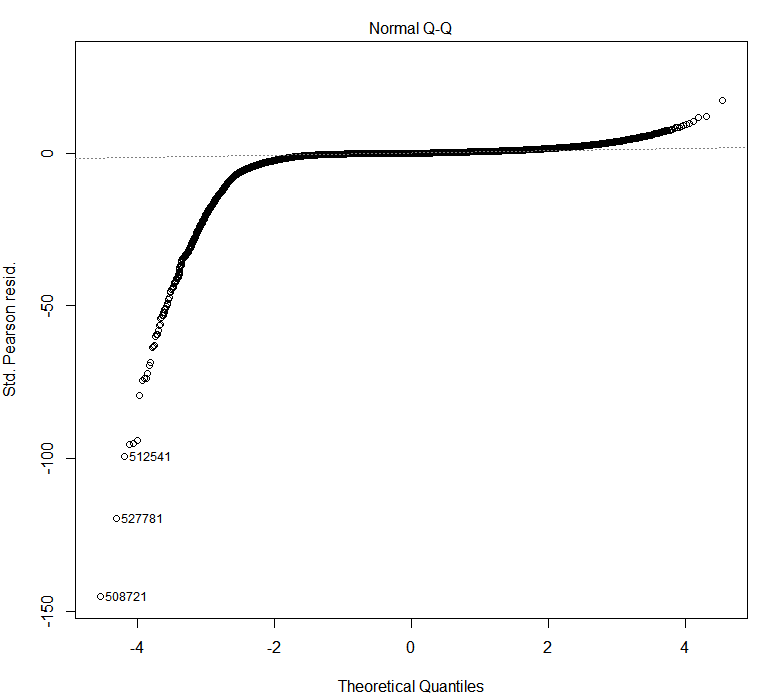
Additionally, the confusion matrices for the different cut-offs are shown below, and by simply reading off the confusion matrices it is easy to identify 0.3 as the best cut-off value within the selected set, for both sets of the confusion matrices, where the first set is for the first data splitting method and the second set is for the second data splitting method. These confusion matrices also provides some extra insights into the model.



**Figure 10: confusion matrices for the logistic regression with different cut-off values**

In order to further assess the fit, normal Q-Q, residual vs leverage. scale location, and residual vs fitted graphs are plotted for the logistic regression. The first set of graphs are for the first data splitting method and the second set of graphs are for the second data splitting method.

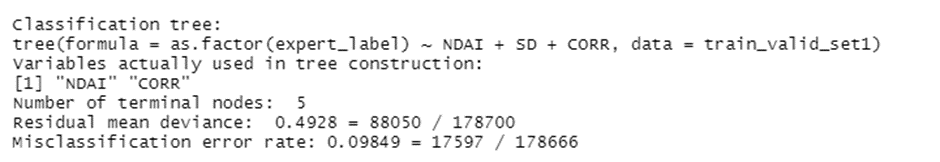




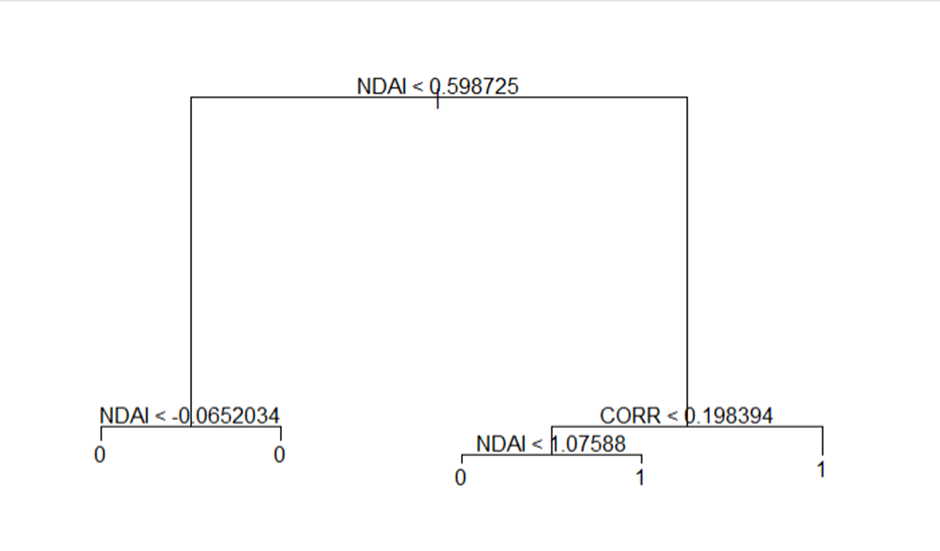
**Figure 11: normal Q-Q, residual vs leverage. scale location, and residual vs fitted for the two data splitting methods respectively**

As indicated by the graphs, there are no clear abnormalities and outlier issues to be concerned with displayed in the logistic regression fittings for both of the data splitting methods. In fact, these plots are very similar for the two data splitting methods, and indicate that the general structure of the fitting is not altered by the data splitting method.

**Diagnostics**

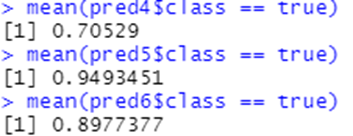
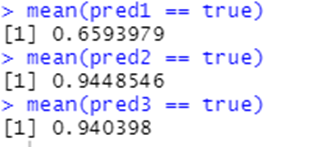
****

Based on the test accuracy, it seems that the decision tree is the best classification we have in hand. We first plot the decision rules of our tree. First notice that the decision didn’t actually use SD in its decision rule, this is surprising as we only include three features to begin with. The tree checks if NDAI is smaller than 0.6, if this is the case, then the tree model will predict the label to be 0 (clear). Otherwise, it checks if CORR is smaller than 0.2. If that is satisfied and NDAI is smaller than 1.076, the model predict label to be 0. Otherwise, the model predicts the label to be 1. The decision rule proposed by the tree model is supported by the conclusion from the paper. In the paper, it suggests that when NDAI is small and CORR is big, they predict the label to be clear. This can be viewed as the left hand side of the tree. However, the tree doesn’t agree with the paper when CORR is big and NDAI is relatively small, i.e. the rightmost branch of the tree. The paper suggests that the label should be 0 according to their algorithm.

****

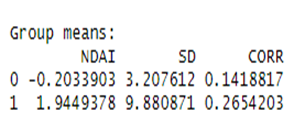
**Figure 12 : Tree Decision rule**

We also perturbed the testset to see the robustness of our tree model. The results correspond to random shuffle one feature at a time for the test set. We shuffle features in the order of NDAI, SD and CORR. Unsurprisingly, the performance of the tree dropped a lot when we shuffle the feature NDAI. This is because based on the decision rule, we see that NDAI plays the most important factor in determining the predicted class. To compare its performance, we also run a perturbation test for our second best model: QDA. The result is quite similar. We also observe that the performance of the model decreases a lot when NDAI is being perturbed.

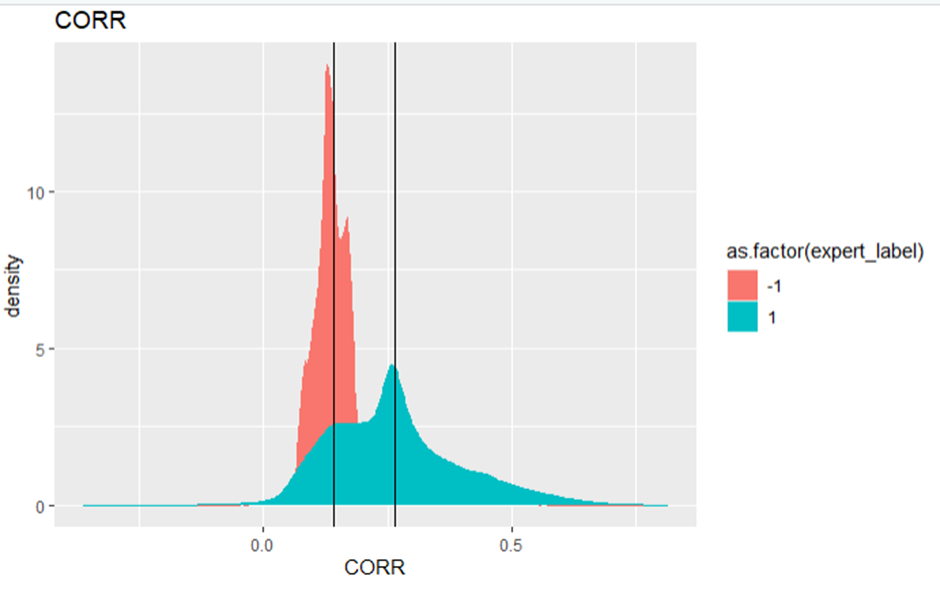
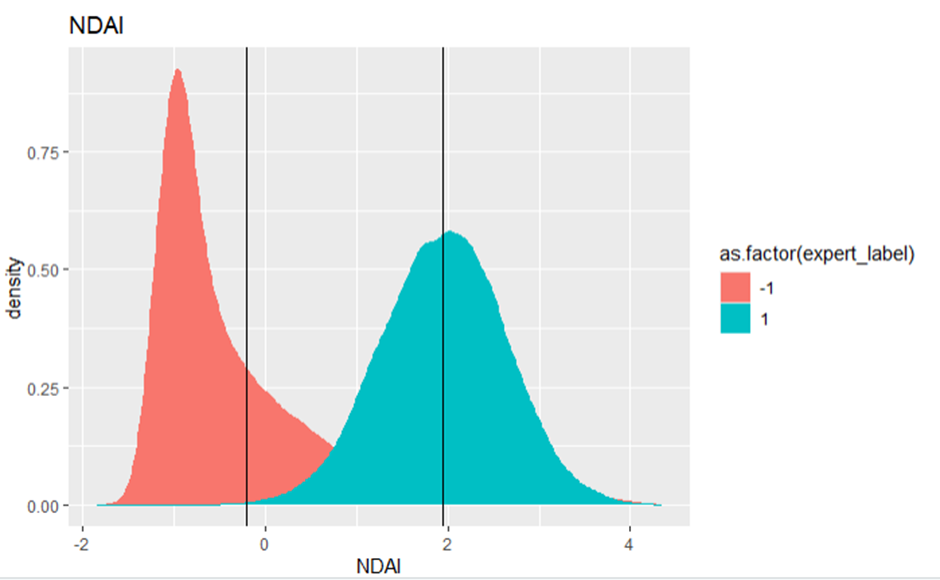
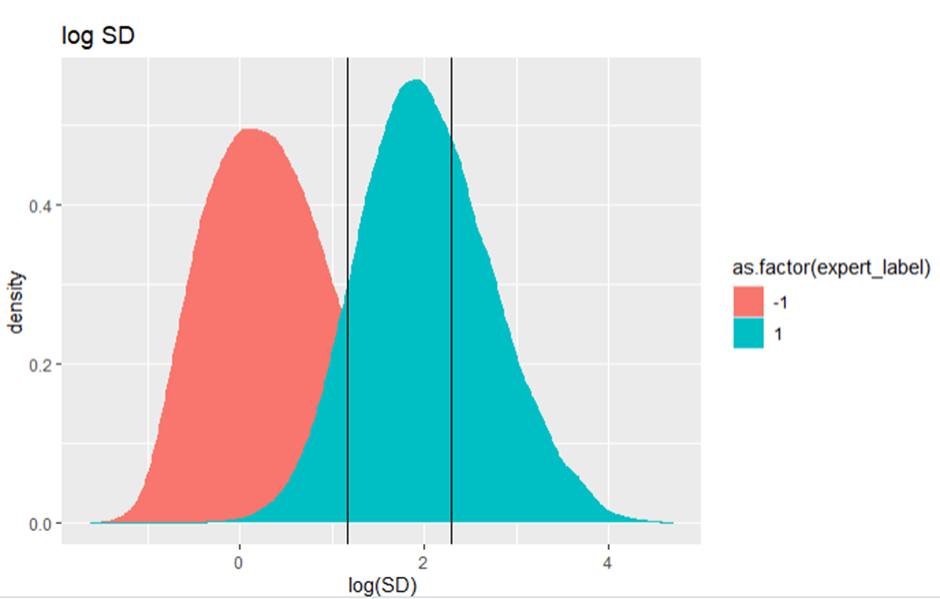


**Figure 13 : Test accuracy after perturbation for tree model and QDA**

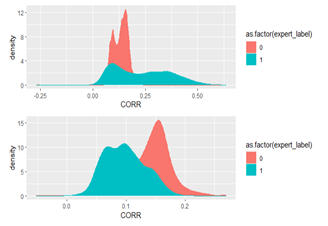
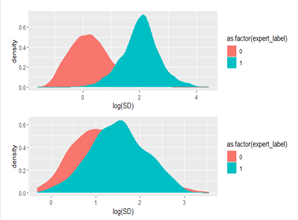
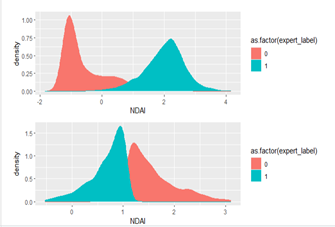
QDA models each feature to be normally distributed within each class. To better understand the QDA model, we also plot the mean value of each feature against the distribution of features based on classes. We see that the QDA model actually predict the mean value of each feature pretty well. For all three features, the mean value is close to the mean value of the distribution.



**Figure 14: Mean for QDA versus mean value from testset**

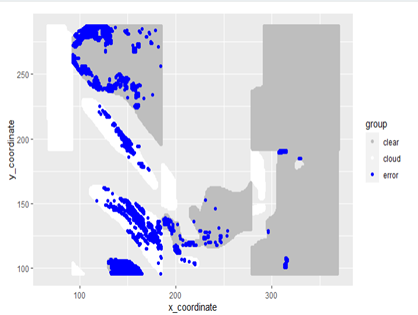


**Figure 15: Mean for QDA versus density plot of testset**



**Figure 16: Distribution of testset(Top) versus Misclassified points(Bot)**

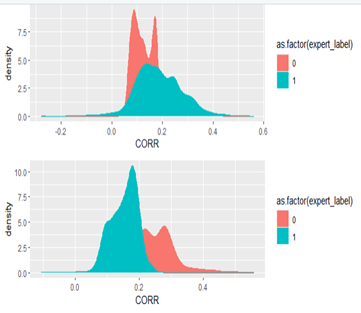
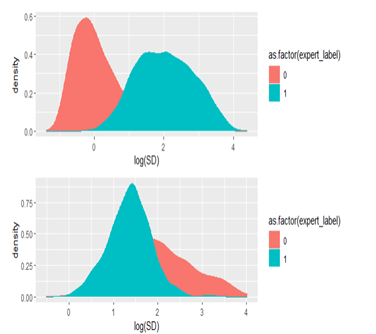
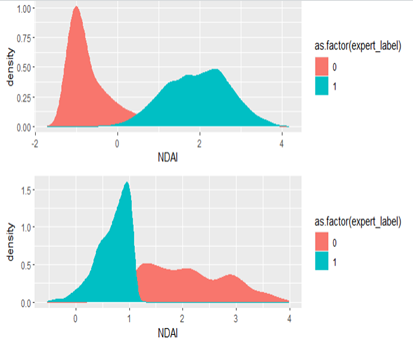
We first plot the density of features from the entire testset versus those points that are misclassified. We notice that for all three features, misclassified points have class 1 density moved towards left and class 0 density moved towards right. This makes sense since we base our prediction on those three features. If the values of those features are much larger or smaller than the median/mean of the density from that class, our model will likely to predict it to be the other class.



**Figure 17: Visualization of misclassified points**

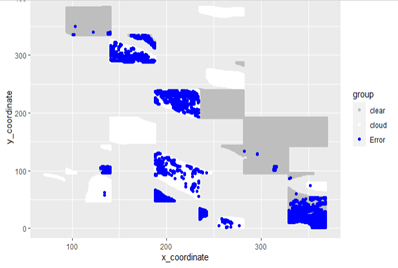
We also plot the misclassified points on the original image. We see that except for a small portion of points, most misclassified points are on the boundary between a chunk of clear space and cloud. As suggested by the paper, all three features have geospatial dependency. In other words, the values of three features near the boundary are likely to be misleading to our model, making it more prone to mistakes.

One classifer that we can try would be convolutional neural networs(CNN). The current tree model we are using can’t handle the problem of boundary cases. CNN applies a filter to image-like object. This can help perserve the spatial dependency presented in this classification problem. By applying a filter, CNN takes advantage of the stationality small image patches and has the pooling effect. All those characterastics make CNN a classfier worth trying. Even if we don’t have expert label in the future, CNN probably will perform well after training extensively with some training dataset.



**Figure 18 : Distribution of testset(Top) versus Misclassified points(Bot)**

For the distribution of the misclassified points, we observe a similar pattern even after we changed the way of splitting data. We still see that for all three features, the misclassified points have class 1 density moves towards left and class 2 density moves towards right. For the misclassified regions, areas that are close to the boundaries between cloudy and clear are still more likely to have misclassfication. Moreover, we also notice that it seems that the tree model is more likely to misclassify points that are labelled as cloud. This finding is also discussed in the paper. The paper suggests that low altitude cloud share some similar features to cloud free areas. This makes them hard to be claasified accurately.



**Figure 19: Visualization of misclassified points**

To summarize and conclude, our tree model achieves a decent accuracy. It has 94.5% test accuracy for splitting method 1 and 89% accuracy for splitting method 2. Based on the visualization of decision rule, we see that the tree model performs well in identifying clear regions with small NDAI values. However, as suggested by the paper, the model has trouble with points that have high NDAI values and high CORR values. Since NDAI plays an important role in the decision rule, the performance will decrease a lot if we perturb the NDAI features. Visually, we see that most misclassified points are located in the boundary. Since tree models can’t really handle geospatial dependencies, we might want to try a model like CNN to address this problem.

**Reproducibility**

Please refer to proj2\_reproducibility.zip on Gradescope