Project 2: Cloud Detection

Student 1: Yi-Nung Huang ID: 26198562

Student 2: Janice Ji ID: 3031816230

1 Data Collection and Exploration

Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data with Case Studies is a paper whose main purpose is to demonstrate the behavior of clouds in the Arctic and detecting them using two algorithms. The first algorithm is a cloud detection algorithm called Multiangle Imaging SpectroRadiometer, or MISR. In this method, data is collected by examining images of the Earth's surface composed of 275 by 275 meter pixels. These pixels are taken by MISR cameras and the primary objective is to differentiate surface pixels of ice and snow with actual cloudy ones. The MISR imagery is taken at nine different view angles (Df, Cf, Bf, Af, An, Aa, Ba, Ca, Da) with four forward and aft angles, ranging from 26.1° to 70.5° in both directions, and one nadir angle, with the Da camera collecting data around 7 minutes after the Df. MISR covers a total of 233 distinct paths with 180 blocks that may be overlapping, with the same path covered every 16 days. However, this detection algorithm is flawed because it does not work satisfactorily when applied on polar regions. To resolve this, another algorithm, the Enhanced Linear Correlation Matching, or ELCM, an efficient processor of MISR data which considers more features necessary for cloud detection, is introduced. The three features deemed useful are the correlation of MISR images from different viewing angles (CORR), the standard deviation of the An camera pixel values (SD), and the normalized distance angular index that characterizes a change in scene with change in MISR viewing direction (NDAI). They contain information that will further allow one to separate cloudy regions from ice- and snow-covered regions. CORR and SD are fixed values, whereas NDAI adapts to the data. Labels from ELCM are then used to train the QDA to produce "probability of cloudiness" labels. This step is called the ELCM-QDA. Contained in the cloud data set are the following 11 features in order: y coordinate, x coordinate, the expert label, NDAI, SD, CORR, Df, Cf, Bf, Af, and An. The presence of clouds, their movement and their distribution in the Arctic all contribute the surface's sensitivity to increasing temperature, so it is an important subject to study for Earth scientists. The warming also contributes to a rise in the carbon dioxide content in the atmosphere, posing a massive threat to the global climate condition. By using Statistical methods, one can delve deeper into this topic as Statistics to ultimately find ways to solve this modern day climate issue.

b. Data summary and maps

The first step is to take a closer look at the data set provided. The expert labels of the pixels should give a good summary and tell us the percentage of pixels in every class. After calculating the relative proportions, the percentage of cloudiness for each image can be summarized in a table as depicted below, where 1 denotes a cloudy pixel, -1 denotes a cloud-free pixel, and 0 denotes an unlabeled pixel.

```
-1         0         1
image1 43.77891 38.45560 17.76549
image2 37.25306 28.63522 34.11172
image3 29.29429 52.26746 18.43825
```

According to the table, image 2 has the highest proportion of cloudy pixels at 34.11%, while image 1 has the lowest at 17.77%. Nearly half of the pixels in image3 are unlabeled, showing that image 3 is not as informative as the other two images.

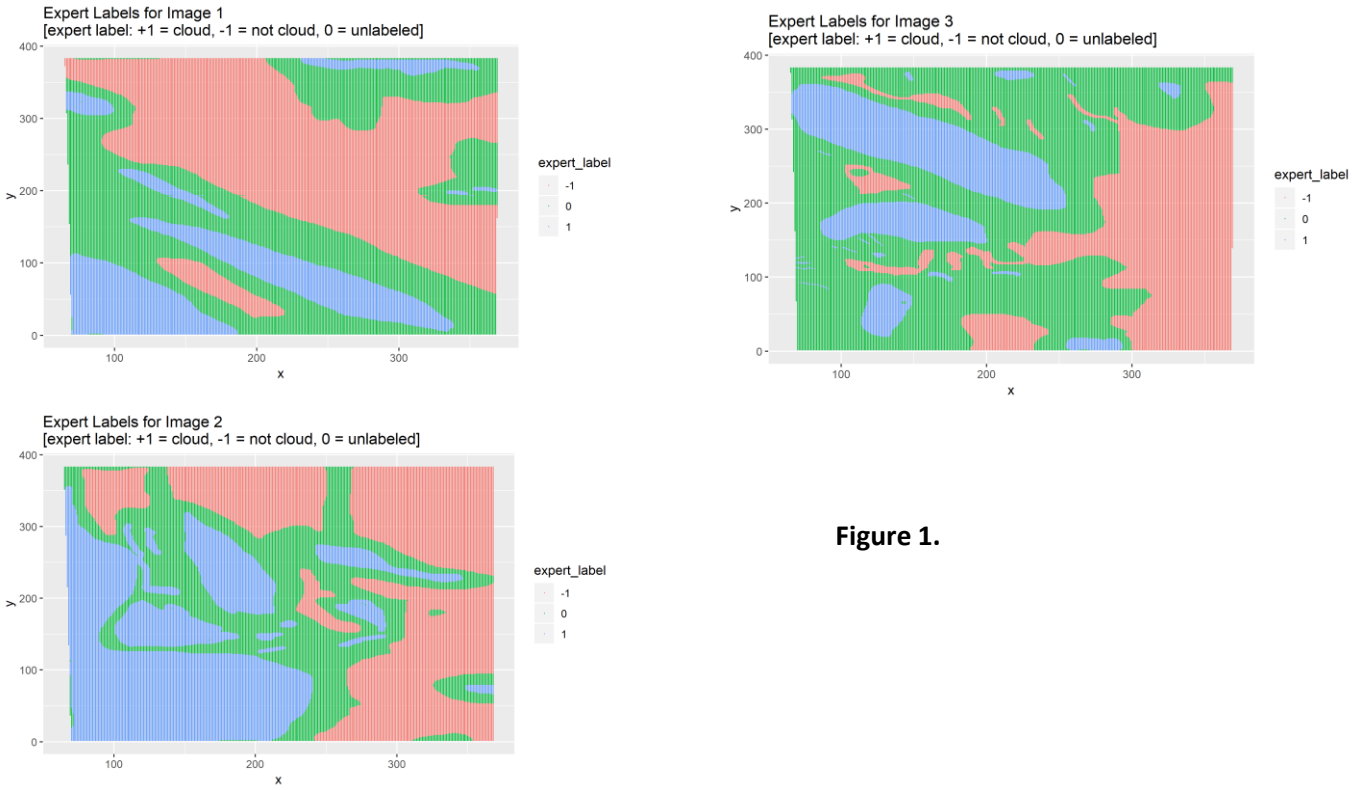

Figure 1.

The three different images were taken over the same region of the Arctic. The blue area in **Figure 1** corresponds to the cloudy regions of the map displayed using each pixel's (x, y) coordinates. The clearly different patterns in the three graphs in **Figure 1** suggests that i.i.d. assumptions for the samples are, in fact, not justified due to the spatial dependencies of the data. Also, the images are taken at different times, so there are temporal dependencies as well.

c. Visual and Quantitative EDA

Afterwards, it is necessary to do visual and exploratory data analysis on the separate physically important features themselves. First, we plotted the correlation (between different camera angles) against the standard deviation.

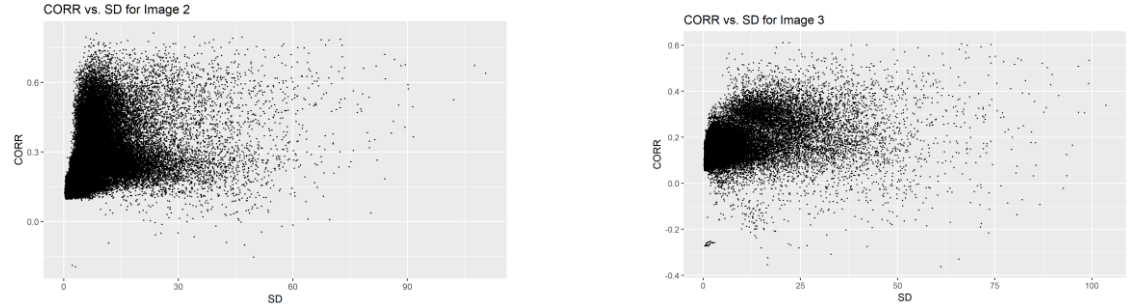

Figure 2: CORR vs SD

Upon observing the scatterplot of CORR against SD, there seems to be a lack of correlation as the data points are becoming more dispersed as the standard deviation gets higher. There are also very few data points with standard deviation above 50.

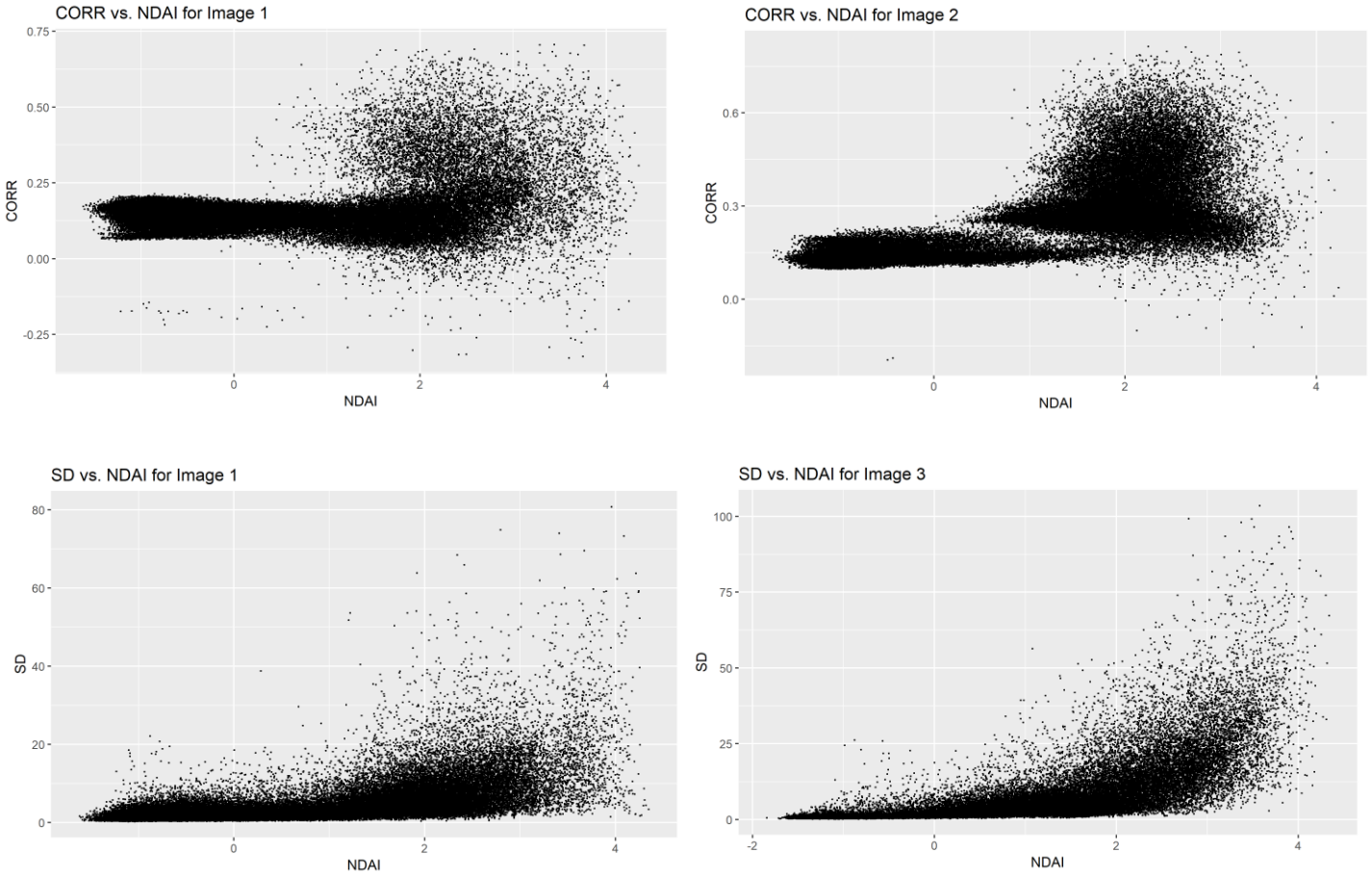

Figure 3: Scatterplot of CORR vs NDAI and SD vs NDAI

When we analyze CORR against NDAI, we see that as NDAI increases, the correlation because more spread out. We also plotted SD against NDAI, and see that NDAI has a positive relationship with SD in all three images. For the three kinds of scatterplots, the trends are similar for all images, so we only show the plots for two of the images in each case.

Next, we will examine the relationship between the expert labels and the features by plotting them with boxplots.

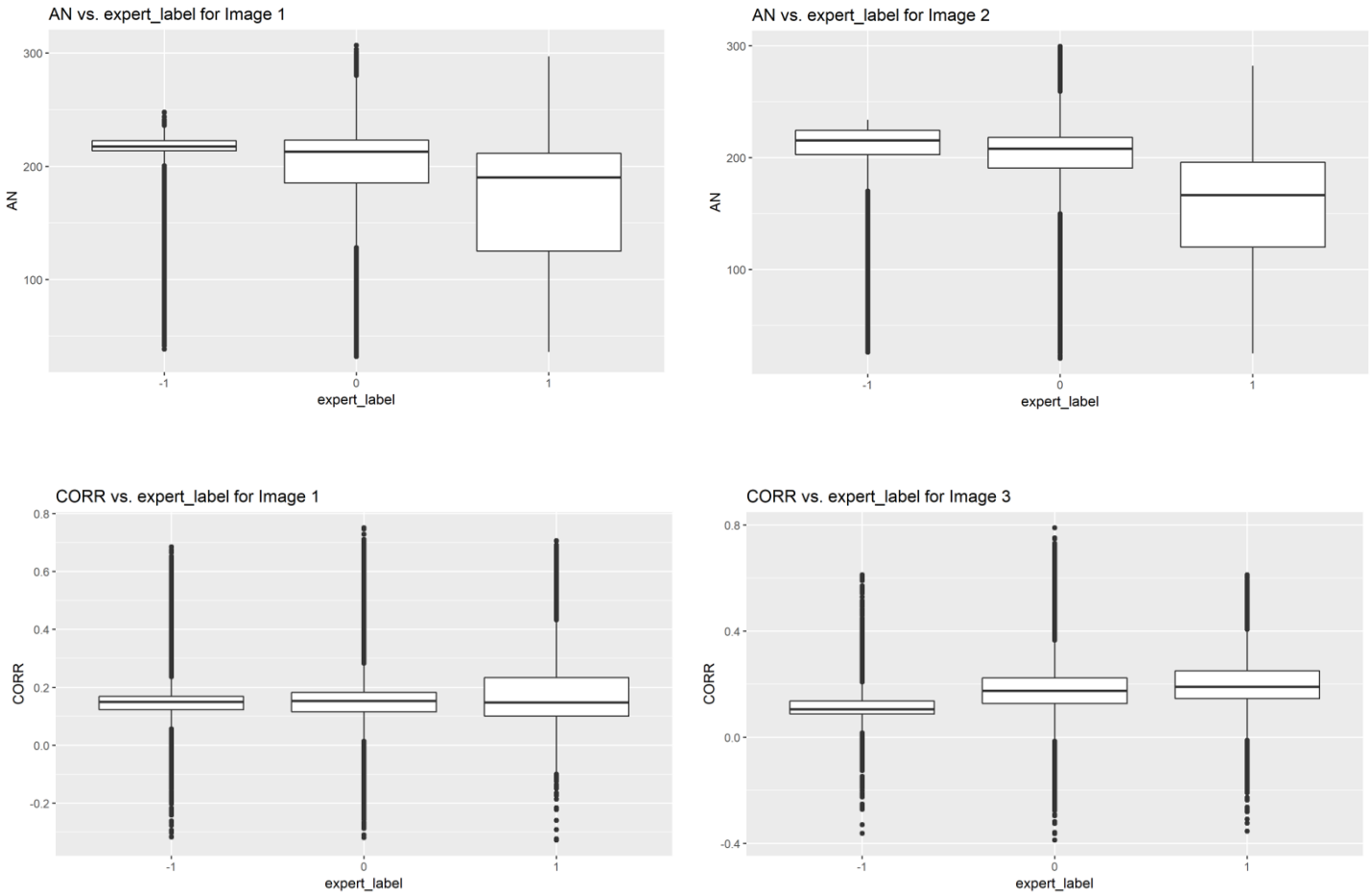

Figure 4: Boxplot of features vs expert labels

Based on the boxplots, lower AN and higher CORR seem to be associated with cloudiness of the pixel.

Now that we have done the necessary exploration of data, we can move on to training our model by splitting the data set.

2 Preparation

a. Data Split

Since the data is not i.i.d., we split the entire data set into training, validation, and testing set in two different ways that takes this information into consideration. Before splitting, all unlabeled data are removed from our model.

For the first method, we take care of the spatial correlation by avoiding the separation between close pixels. We split the data by dividing each image into twenty separate blocks based on their coordinates on the map, so that the images are laid out on a five by four grid. Let's called this **Method A**. We proceed to randomly select 70% of the blocks to use as the training set, 15% for validation, and 15% for testing.

Finally, we combined the training sets, validation sets, and test sets for all three images to form the final training, validation, and test sets.

For the first method, we take care of the spatial correlation by avoiding the separation between data taken at the same time. That is, we simply take image1 as the training set, image2 as the validation set, and image3 as the test set. Let's call this **Method B**.

b. Baseline

We report the accuracy of a trivial classifier which sets all labels to -1 on the validation and test set. The validation accuracy for Method A is about 77.59% and the test accuracy is about 61.02%. The validation accuracy for Method B is about 52.2% and the test accuracy is about 61.37%. Such classifier will have high average accuracy only if most of the data have expert labels -1.

c. First Order Importance

Under the assumption that the expert labels are correct, we try to find three features that contribute significantly to the prediction of the labels. We plotted the boxplots of features against the expert labels on the training set.

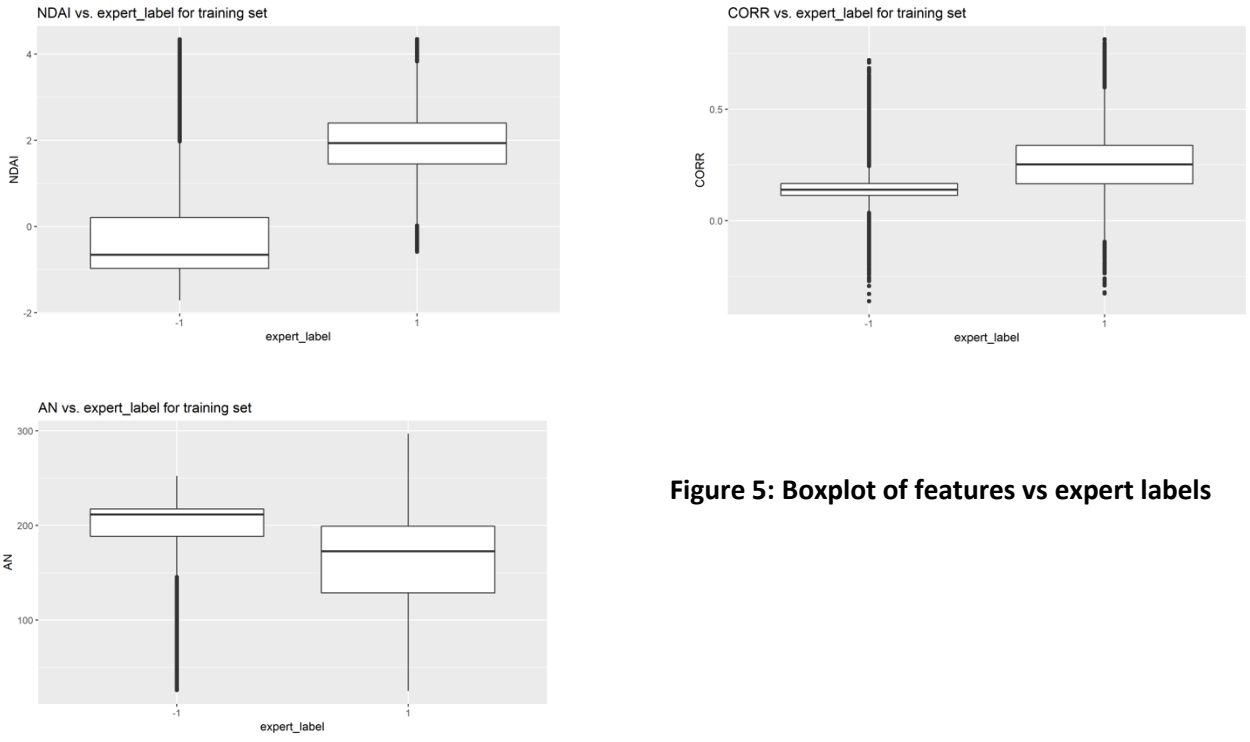

Figure 5: Boxplot of features vs expert labels

There seems to be a large discrepancy in the values of NDAI between cloudy pixels and non-cloudy pixels. The interquartile ranges between the two boxplots don't overlap, so this appears to be a good feature. For similar reason, CORR and AN also seem to be good features.

We also plotted the correlation plot, from which we can see that expert label is highly correlated with NDAI, CORR, and AN.

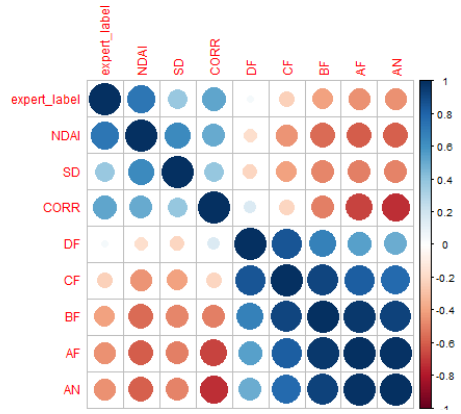

Figure 6: Correlation plot of features

Therefore, we would suggest that the three best features that could help with the prediction of expert labels to be NDAI, CORR, and AN.

d. CVgeneric Function

We wrote a generic cross validation function CVgeneric in R, which can be referenced from our GitHub repository (https://github.com/janiceji/stat154proj2).

3 Modeling

In the following section, we attempt to fit different classification models using various classifiers, such as Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, and K-Nearest Neighbors.

a. Classification and Cross-Validation

First, we merged our training and validation sets to fit our CV model and performed the various classification models listed above onto it.

Let's explore this for the case of **Logistic Regression**. Logistic regression is a generalization of a linear model. It requires the assumption that the observations are independent of one another, which is not held in this case. After performing logistic regression on the training data of **Method A**, the overall accuracy is 89.16% and the K-fold CV loss is 10.84%. For **Method B**, the overall accuracy is 92.67% and the K-fold CV loss is 7.33%. In addition, we calculate the accuracy among the five folds (**Figure 7**)

Folds <int>	Accuracy <fctr>
1	89.03%
2	89.2%
3	89.07%
4	89.29%
5	89.2%

Figure 7: Accuracy across Five Folds for Method A (left) and Method B (right)

Folds <int>	Accuracy <fctr>
1	92.87%
2	92.74%
3	92.58%
4	92.66%
5	92.52%

Then we make predictions on the test set to get the prediction probability for each subject. If the prediction probability is greater than 0.5, the predicted class is 1 and if it is less than or equal to 0.5, then the predicted class is 0. We compare the predicted classes and the actual classes on the expert labels to get the test accuracy. The test accuracy for **Method A** is 88.08% and the test accuracy for **Method B** is 76.98%.

The same idea can be applied to the other classification methods.

Method A Classification	**Overall Accuracy**	**Fold 1 Accuracy**	**Fold 2 Accuracy**	**Fold 3 Accuracy**	**Fold 4 Accuracy**	**Fold 5 Accuracy**	**5-Fold CV Loss**	**Test Accuracy**
Logistic Regression	89.16%	89.03%	89.2%	89.07%	89.29%	89.2%	10.84%	88.08%
Linear Discriminant Analysis	89.33%	89.2%	89.38%	89.28%	89.48%	89.31%	10.67%	88.88%
Quadratic Discriminant Analysis	88.82%	88.8%	88.71%	88.88%	89.08%	88.62%	11.18%	90.35%
K-Nearest Neighbors	93.31%	93.16%	93.39%	93.33%	93.46%	93.2%	6.69%	88.99%

Method B Classification	**Overall Accuracy**	**Fold 1 Accuracy**	**Fold 2 Accuracy**	**Fold 3 Accuracy**	**Fold 4 Accuracy**	**Fold 5 Accuracy**	**5-Fold CV Loss**	**Test Accuracy**
Logistic Regression	92.67%	92.87%	92.74%	92.58%	92.66%	92.52%	7.33%	76.98%
Linear Discriminant Analysis	92.74%	92.9%	92.79%	92.66%	92.79%	92.58%	7.26%	77.51%
Quadratic Discriminant Analysis	92.41%	92.41%	92.33%	92.42%	92.45%	92.44%	7.59%	81.76%
K-Nearest Neighbors	94.69%	94.72%	94.68%	94.72%	94.68%	94.63%	5.31%	43%

Table 1: Summary of Accuracies for various classification methods

Linear discriminant analysis would only work well under the assumption that the classes come from a Gaussian distribution with a unique mean and have a common covariance matrix across all classes. Quadratic discriminant analysis holds the same assumptions as LDA except that the covariance matrices between the separate classes do not have to be the same. Moreover, QDA works best when the number of subjects is large.

Unlike logistic regression, LDA, and QDA, KNN is a non-parametric classification algorithm, meaning there are no assumptions made on the underlying distribution of the data. This is better when working with data where the distribution data is unknown.

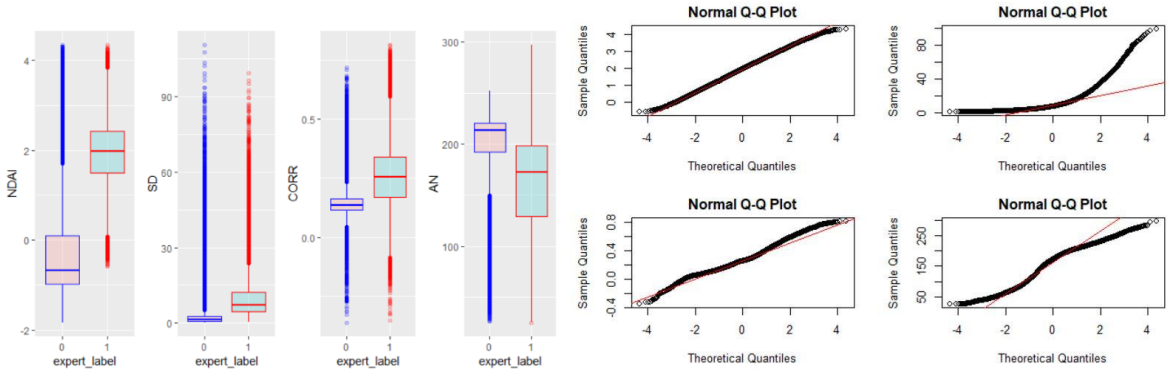

Figure 8: Assumption Checks

In **Figure 8**, we pick the features NDAI, SD, CORR, and AN to test and see if the distributional assumptions for normality and equal variance are met. The difference in the size of the boxplots is an indication of the unequal variances between the features. Q-Q plots are also plotted to check normality on. We found that the assumptions are not satisfied.

The results are summarized in **Table 1**. We can see that the CV accuracies across folds are all high, meaning that the classification models are reasonable. QDA has the highest test accuracy. Hence, it is the best classifier among the ones we tried. KNN works decently well when we split the data into block, but it works badly for Method B, where we took image 3 to be the test set.

b. ROC Curves

We plotted the ROC curves to compare the four classification methods that we tried. According to the paper, the initial reason for cloud detection problem is that clouds could lead to further global warming. Therefore, we believe that failing to detect the clouds in cloudy regions is a more severe error than incorrectly predict the clouds in non-cloudy regions. That is, we are aiming for small number of false negatives, which implies a large true positive rate. So to choose the cutoff point for the ROC curve, we set 0.95 to be the threshold for TPR. For all points with a TPR larger than 0.95, we pick the point with the lowest FPR to be the cutoff point, which is highlighted on the curve.

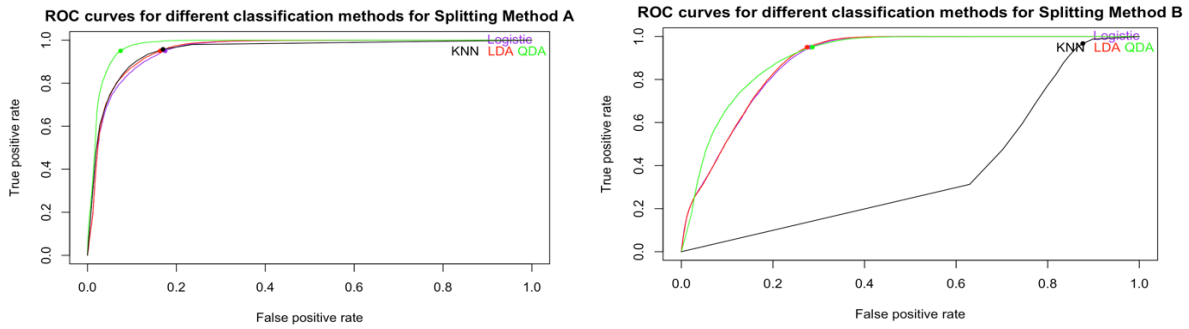

Figure 9: ROC curves

We then compare the performance of the four different classification methods based on the ROC curves. We calculate the AUC for each curve because higher AUC implies better prediction. For both splitting methods, we found that QDA achieves the highest AUC (0.977 for splitting method A, and 0.907 for splitting method B). That is, in both cases, QDA is the best classification methods among all four.

c. Other Relevant Metrics

As stated in the previous part, we are aiming for small number of false negatives. Therefore, one good metrics to use for assessing the fit for different classification methods would be FNR (False Negative Rate).

CLASSIFICATION METHODS	FNR (FALSE NEGATIVE RATE)
LOGISTIC REGRESSION (SPLITTING METHOD A)	0.17
LOGISTIC REGRESSION (SPLITTING METHOD B)	0.39
LDA (SPLITTING METHOD A)	0.15
LDA (SPLITTING METHOD B)	0.37
QDA (SPLITTING METHOD A)	0.19
QDA (SPLITTING METHOD B)	0.28
KNN (SPLITTING METHOD A)	0.16
KNN (SPLITTING METHOD B)	0.18

Table 2: FNR for various classification methods

If we use this metric to compare the four different classification methods, the lower the FNR, the better the model. Therefore, we would conclude that LDA is the best model for splitting method A, and KNN is the best model for splitting method B. Such conclusion is quite different from our conclusion in previous parts, where we used the accuracy as our metric.

4 Diagnostics

a. Analysis of QDA Classification Model

Based on part 3b., we propose that QDA is a good classification model for this cloud detection problem. To analyze this model, for both splitting methods, we train QDA on all eight features (NDAI, SD, CORR, DF, CF, BF, AF, AN) using different proportion of training data each time, and examine the convergence of the

group means. We find that the group means for most features converge as the proportion of training data increases. The following plot of the convergence of NDAI for splitting method A is given as an example.

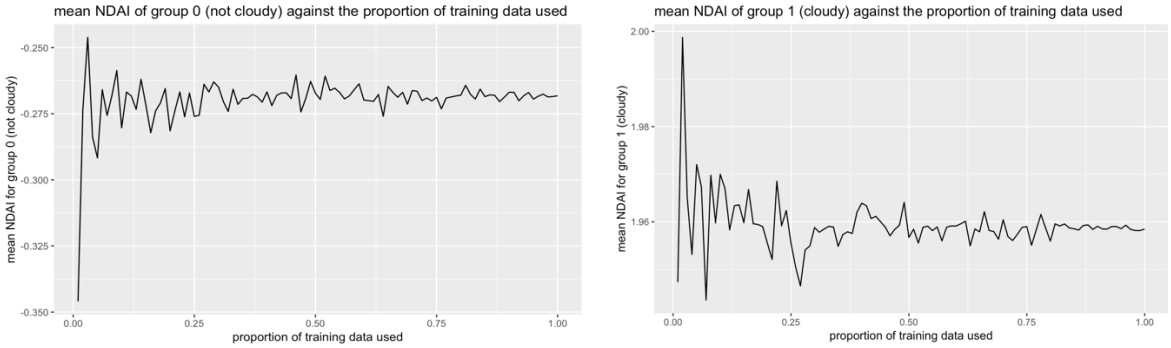

Figure 10: Group NDAI means vs Proportion of training data used (Splitting method A)

For splitting method B, the group means for most features also converge. Example of the convergence of NDAI is shown here.

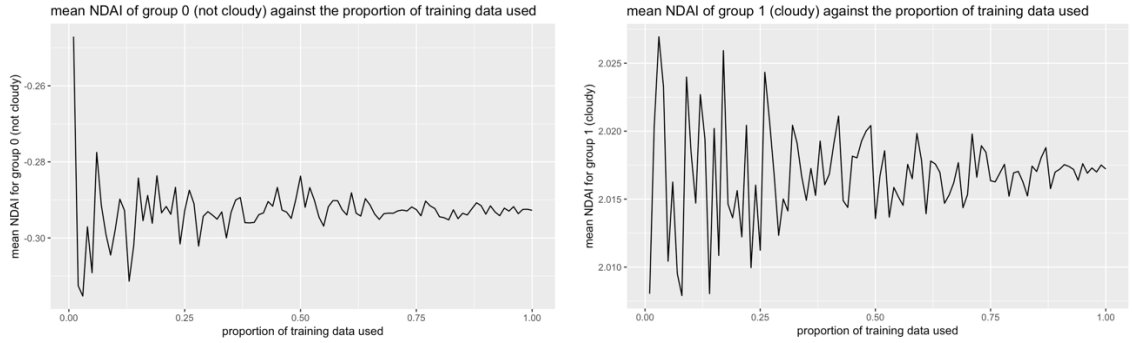

Figure 11: Group NDAI means vs Proportion of training data used (Splitting method B)

b. Misclassification Errors

For splitting method A, we examine the misclassification errors by plotting both the misclassified test data and the entire test data using the x, y coordinates, and compare the two plots.

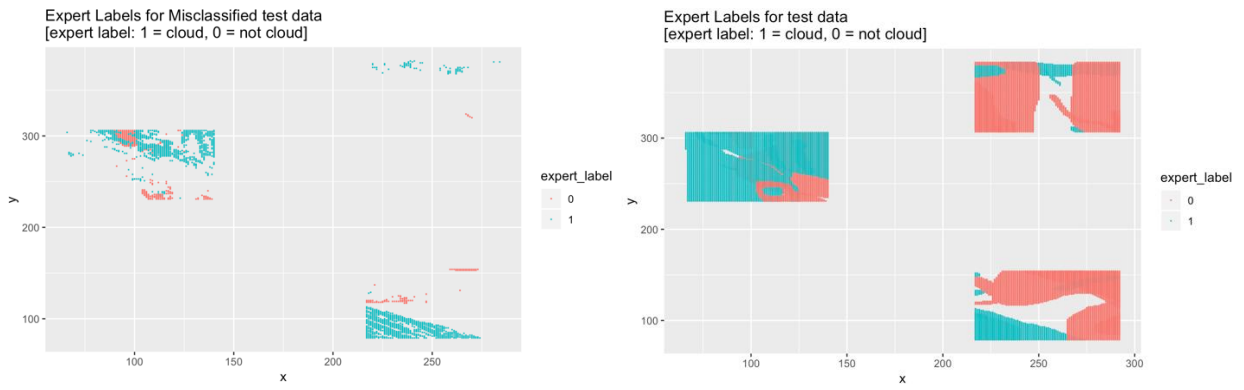

Figure 12: Misclassified test data vs Entire test data (Splitting method A)

We find that most of the errors occur in the cloudy regions. That is, we tend to fail to detect the clouds. We then examine whether the misclassification errors occur for specific ranges of feature values by summarizing the data. We found that the misclassified test data and the entire test data have quite different ranges of SD, NDAI, CORR, AN, and AF. For example, the misclassified test data has SD ranging from 0.78 to 35.83, while the entire test data has SD ranging from 0.26 to 56.55.

For splitting method B, we follow the same process and generate the following figure.

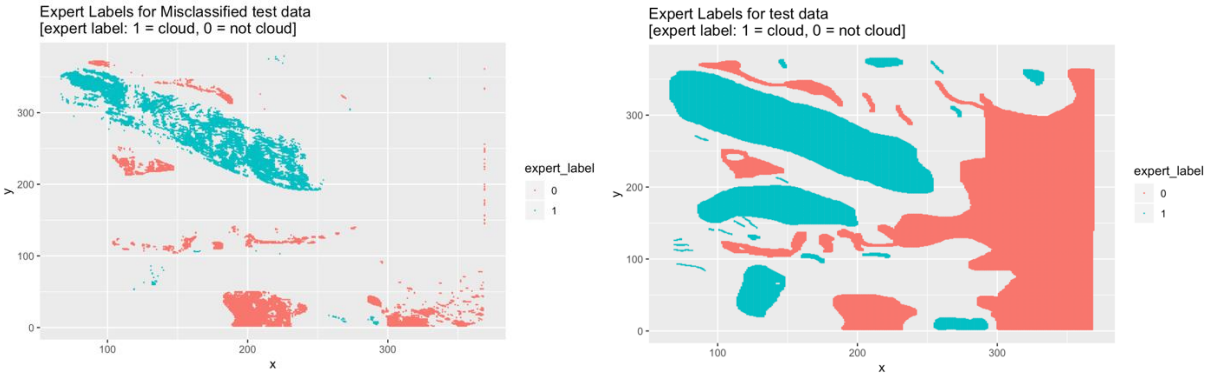

Figure 13: Misclassified test data vs Entire test data (Splitting method B)

Again, we see that most of the errors occur in the cloudy regions. Then, we summarize the data and found that the misclassified test data and the entire test data have quite different ranges of SD, NDAI, and AN. For example, the misclassified test data has AN ranging from 46.89 to 252.4, while the entire test data has AN ranging from 34.54 to 274.49.

c. Better Model

Part 4a. showed that QDA does a good job as the group means converge for most features, and part 4b. showed that the features that seem relevant to improve our classifier would be NDAI, SD, CORR, AF, and AN. Therefore, based on parts 4a. and 4b., we propose that a better model would be a QDA model trained only on these five features instead of the original eight features.

We train this proposed model and got accuracy 91.20% for splitting method A, and 80.62% for splitting method B. The original model gives accuracy 90.35% and 81.76%, respectively. Although the test accuracy did not necessarily improve, we found that for splitting method A, the proportion of cloudy pixels among all misclassified data went from 0.60 to 0.53. For splitting method B, such proportion went from 0.78 to 0.71. In other words, the proportion of misclassified errors are more evenly distributed across cloudy and non-cloudy regions with this new model, meaning that it can detect the cloudy regions better. Therefore, I think this new model will work well on future data without expert labels.

d. Change of Splitting Method

The results for both ways of splitting the data are shown in parts 4a. and 4b. They are not too different in part 4a. because almost all features converge in both cases. However, in part 4b., the features that have specific ranges in the misclassified data are different for both splitting methods.

e. Conclusion

Based on the exploration above, we conclude that QDA is a good model to use in this cloud detection project. The CV error is acceptably low, the test accuracy is high, and the group means converge well. In terms of the splitting method, we found that method A performs better than method B under all four classification models, meaning that splitting the data using blocks is preferable. Combining the analysis of both part 2c. and part 4b., we also conclude that NDAI, SD, CORR, AF, and AN are more significant among all eight features. Therefore, for this project, one should consider training classification models on subsets of these five features, or any kinds of new features generated from these five features.