STAT 154 Project 2: Cloud Data

John Tae 26154871
Dorothy Leung 3033384317

May 1, 2019

Data Collection and Exploration

1a)

This paper is concerned with analyzing and detecting clouds in order to better ascertain the effects that global warming may be having. However, doing so is very difficult to an accurate enough degree for analysis, so the researchers make use of a nine camera MISR imaging, which can produce a vast amount of data. The MISR data however only fully transmits the red radiances and nadir camera channels at 275m resolution. The data used in this study were collected from 10 such MISR objects over path 26 over the Arctic, Northern Greenland, and Baffin Bay with a repeat time of 16 days between consecutive orbits. Overall the paper uses collected data from April 28 through September 16, which roughly translate to the daylight season in the Arctic. Six data units from each orbit of the MISR blocks are used, which some being excluded due to natural occurrences. Overall there are 57 data units used, comprising a very large data set when pixel size is taken into account. In performing the modeling, itself, the formerly used MISR algorithms were not handling certain edge cases well, so the researchers decide to combine classification and clustering schemes while also focusing on efficiency to circumvent the huge data size. The algorithm is designed as an enhanced linear correlation matcher which utilizes calculated thresholds and incorporates three key physical characteristic features, the correlation of MISR images of the same scene from different viewing directions (CORR), the standard deviation of MISR nadir pixel values (SD) and the normalized difference angular index (NDAI). Next the labels are used to train a QDA model which provide probability for the labels outputted, providing another layer of analysis rather than spitting out labels. Generally, in comparison to primarily used algorithms such as SVM, ASCM, and SDCM, ELCM and ELCM-QDA had strong results, save for some caveats that are mostly due to corrupted data. It is safe to say however that even given these caveats, using the ELCM algorithm with these features provided is more accurate and provides more coverage. This work is significant because it highlights the importance of statisticians being involved in the data processing itself, rather than a post collection analysis. Outside of the impact on statistics, the study also makes a strong point in better understanding the Earth and its change.

1b)

Percentage Counts Image3

expert_label	
cloud	0.184363
not_cloud	0.292912
unlabeled	0.522620

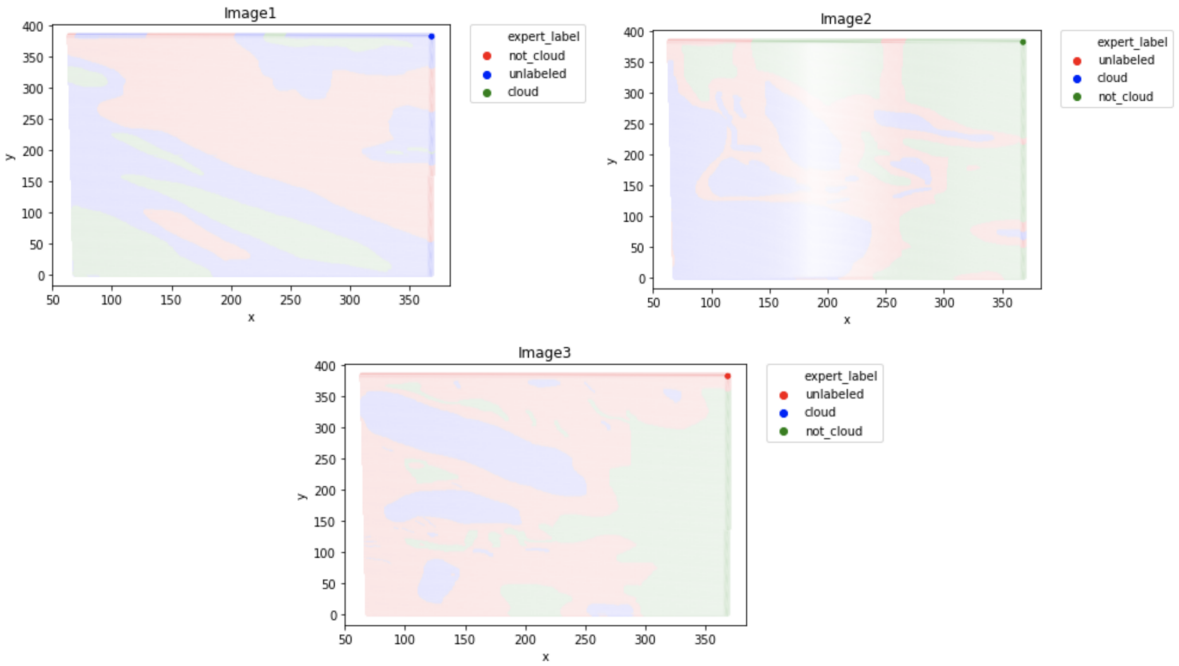
Percentage Counts Image2

expert_label	
cloud	0.340765
not_cloud	0.372146
unlabeled	0.286056

Percentage Counts Image1

expert_label	
cloud	0.177655
not_cloud	0.437789
unlabeled	0.384556

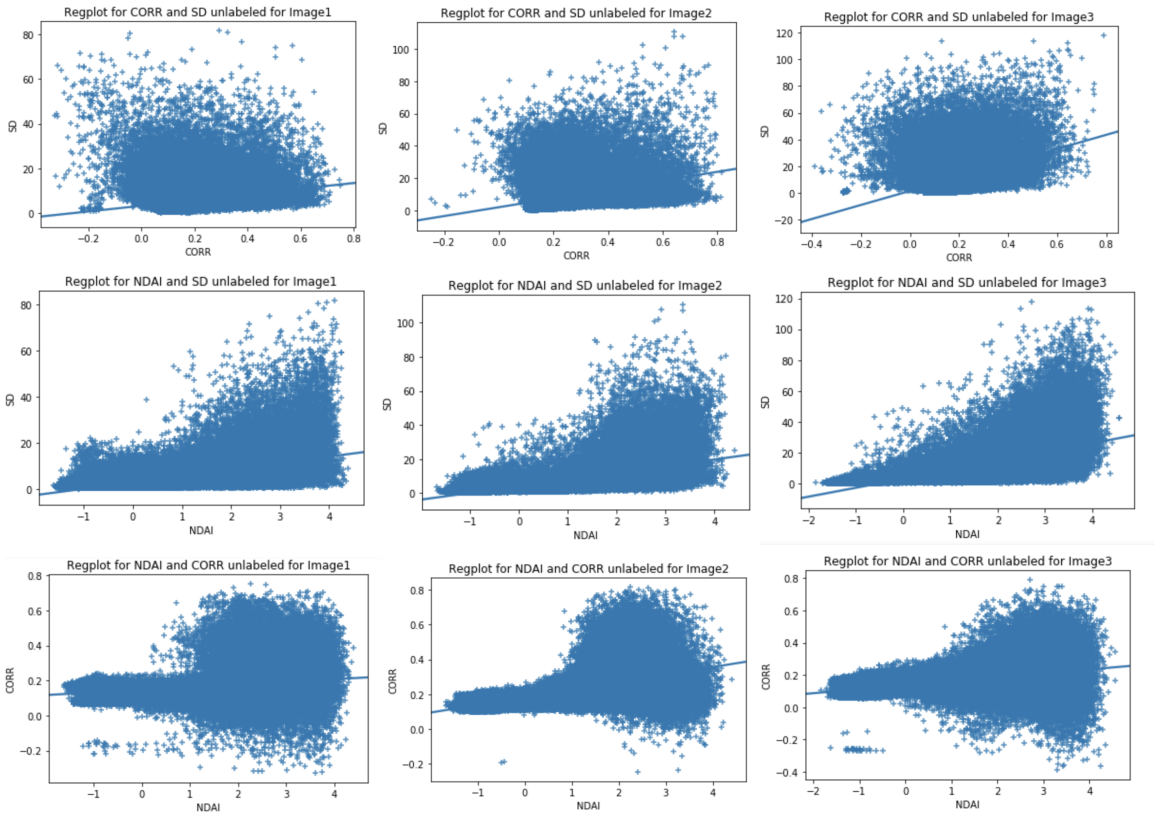
In order to better understand the data, we group the data into their labels and compute the percentage of the pixels within each. After summarizing the data into the separable expert labels for the respective data sets, we see that image 1, image 2, and image 3 have fairly different percentage compositions. Image 2 in particular has a high percentage of cloud pixels in comparison. Thus, we may want to be careful in our future analysis when deciding whether to merge the data or treat each as a separate entity.



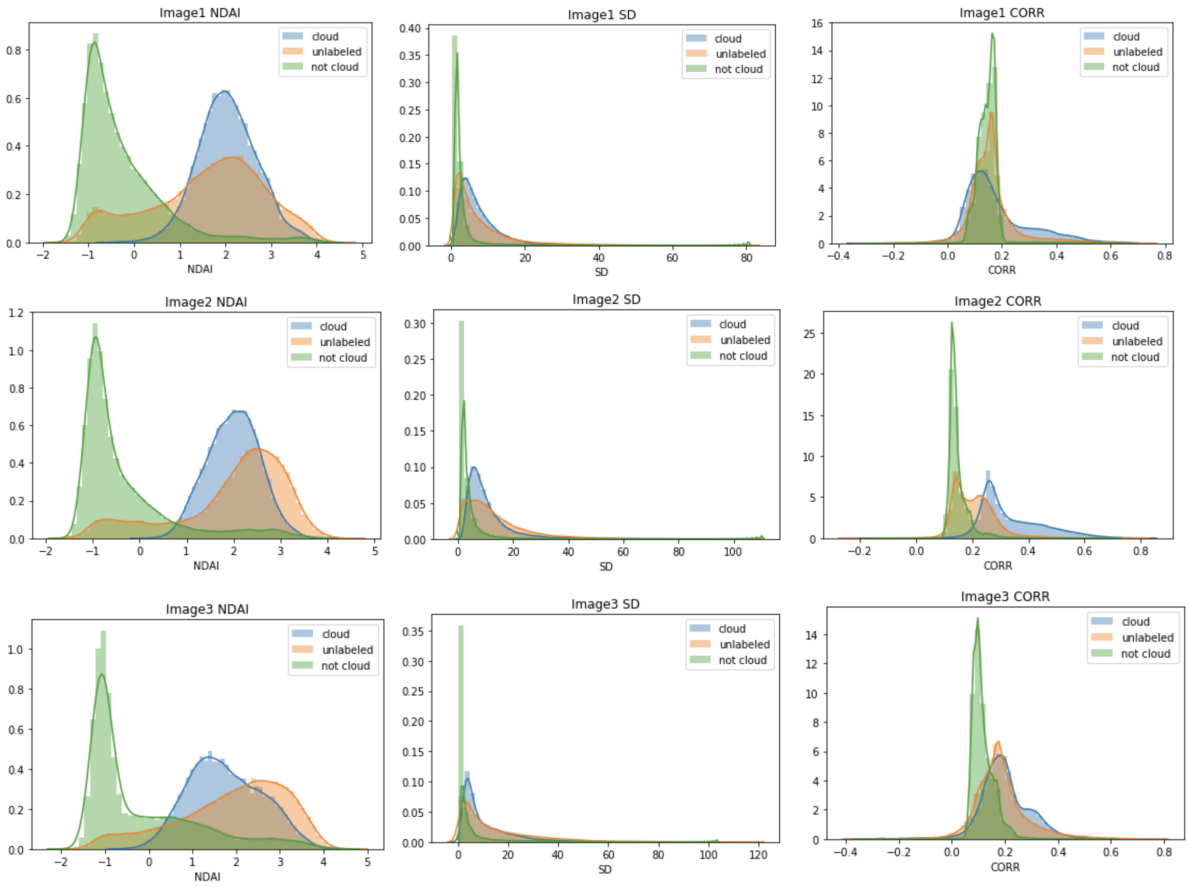
Next, we would like to explore the geographical relationship to the labels of the data. We graph the data into color gradients, depicting the shape and trend of the data according to its coordinate location, as well as expert label. Interesting to note is that in image 1 and image 3 exists the semblance of a linear relationship for the cloud labeled pixels, almost with a negative like slope. In these two data sets we also note that the pixels are grouped quite nicely alongside their labels. In image 1 we see a similar trend with the not cloud and unlabeled data as well. Image 2 is interesting as we see that the data is much more scattered and interweaved with each other, particularly the unlabeled data. It is also important to remark that image 2 has the highest percentage of cloud

labeled pixels. For this data set, an i.i.d assumption would not be justified as we can see that there are clear signs of relationships through our analysis.

1c)



Next, we dive into the features themselves and possible relationships we can explore. There are numerous things to point out, however most glaringly obvious and perhaps applicable is the relationship between NDAI and SD. There is a strong positive slope relationship between the two, which actually may come as a bit of a surprise. There is an interesting interaction between NDAI and CORR as well, as the variance of CORR seems to drastically increase as
NDAI increases. This means that low values of NDAI were recording monotonous values of correlation between the MISR images.



With these plots we analyze the distribution of the features grouped into their expert labels per image set. We see that all of the image sets have pretty different distributions of feature values in the cloud, not cloud, and unlabeled data sets. Most notably perhaps is the distribution of NDAI, which is most extreme for cloud relative to not cloud. Although we would expect this to be different given then these are our hand labeled labels, it is important to note for the future. Due to these differences, we might hope to use it later in our modeling to help us distinguish between future cases.

Preparation

2a)

As we observe the incoming data does not follow iid assumption, we cannot split the data directly through random split as random splitting is a bad option when some features have dependencies over others. Also, it is easy to classify label if we are given information between the previous and the next data to get our next prediction. However, as we do not know how future data is coming in, we can go about by extracting a set of data from the training set to be our validation test to resemble the train and test split. So, we decided one non-trivial approach of splitting the data is by splitting

proportionally. As this could avoid any imbalanced data contained in any of the set which later result in building a complicated model. This method will ensure each set is separated with the same ratio from image1, image2, and image3 into training, validation and test set. Will merge the proportioned data set to be the final train, validation and test set for future training classifier to train on multiple models. Another approach of non-trivial splitting is based on coordinate splitting. By coordinate splitting, we set the boundary between smallest and greatest value of x-coordinate as well as for y-coordinate, then we found out there will be a total of 127 blocks distributed evenly along y-axis and 76 blocks along x-axis, which tells us each block will have a 4x3 dimension across the whole image. Thus, we are making sure that we do not ignore or left any data behind unused. Following the process, we resample each block/region which satisfy the range condition and extract 20% to be the test set, and 80% temporary train set that later split again with another 20% to be the validation set by resampling method. Though this process is lengthy and time consuming, it is one of the optimal ways of splitting data that is not balanced. In addition to that, no matter how we split the data, the resulting train, validation, and test set will always remain consistent proportionally. These ways of splitting will give us a more general and unbiased approach.

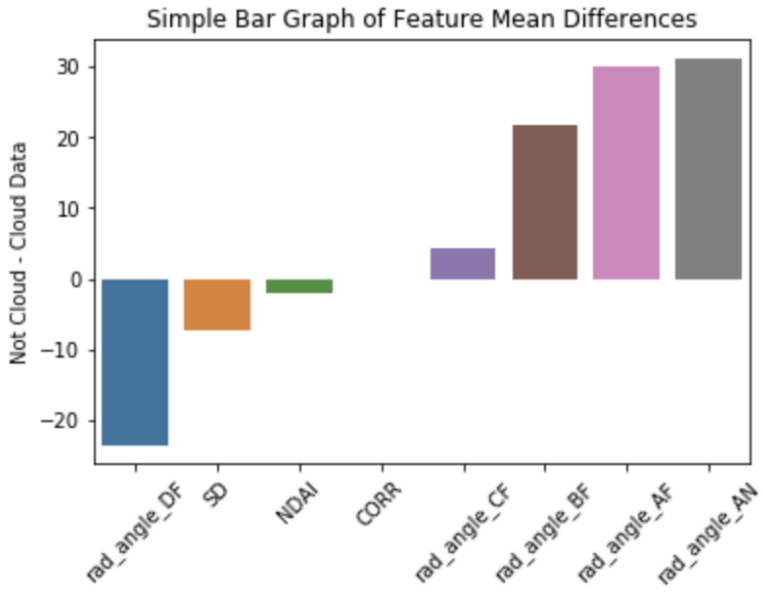
2b)

The accuracy of a trivial classifier is expected to be roughly low when we set all labels to -1. Under the assumption of the data is iid and unique, a trivial approach to split the data is to perform a native split. A native split splits data into three sets randomly, without further process. Thus, using logistic regression classifier, our average accuracy prediction from validation set and test set come out to be around 40% approximately. When our model overfits based on a complicated and imbalanced model, our classifier will have high average accuracy on training data but resulting in low test accuracy.

2c)

Assuming the expert labels are true, we have a variety of methods that we can use to discern the most important features. Note that at this point we have merged the three separate into one image under the assumption that each comes from an identical distribution, even if it not i.i.d. We have also removed the unlabeled expert labels because we believe that it would only lower the accuracy of our classifiers and muddle our analysis.

We might first want to simply look at which features are the most different between cloud and not cloud labeled data.



Interestingly the radiance angle for DF is much more different than the other radiance angles. However, we also may want to compute some anova analysis in order to further determine which variables are actually statistically significantly different than their labeled counterparts. We used Anova analysis and recursive elimination, from the SciPy and sklearn packages respectively for this next step. We received mixed results from using these two methods. Using Anova we actually did not receive any strong p-value indications.

	F	P_value
NDAI	207704.644016	0.000000e+00
SD	38265.372810	0.000000e+00
CORR	116670.701958	0.000000e+00
rad_angle_DF	13967.728120	0.000000e+00
rad_angle_CF	599.931509	2.769065e-132
rad_angle_BF	17456.921592	0.000000e+00
rad_angle_AF	38463.670566	0.000000e+00
rad_angle_AN	45387.509114	0.000000e+00

However, using recursive feature elimination we found very strong results indicating that indeed the best features are the best features determined from the analysis from the paper, and are tied as the best features as determined by the algorithm. Details on the algorithm are included in the accompanying jupyter notebook within the GitHub repository.

	column	support	ranking
0	NDAI	True	1
1	SD	True	1
2	CORR	True	1
3	rad_angle_DF	False	4
4	rad_angle_CF	False	3
5	rad_angle_BF	False	5
6	rad_angle_AF	False	6
7	rad_angle_AN	False	2

If we were select four features however, radiance angle ‘AN’ would have been chosen as the fourth. This was a surprising result given our analysis but makes sense in accordance with the research findings. Using a more complex classification method to select features may also have netted stronger results.

Modeling

3a)

Going forward we focus on implementing our classification models and build upon our former analysis. For our classification methods we begin with using logistic regression, k-nearest neighbors, support vector machines, QDA, and random forest. We also provide three separate methods of cross validation and will combine the results for an in-depth coverage. We are also interested in comparing our different splitting methods to see which nets the best accuracy. The three separate methods pertain to normal randomized splitting of our data, method one from 2(a) for splitting the data, and method two. In this next page we summarize our results from the different machine learning methods and then will go into some details for some.

KNN

Trivial Splitting

	Accuracy	Fold
0	0.867987	1
1	0.933538	2
2	0.915686	3
3	0.859165	4
4	0.908947	5
5	0.897065	Average

Method One

	Accuracy	Fold
0	0.999115	1
1	0.833429	2
2	0.825804	3
3	0.899917	4
4	0.999714	5
5	0.911596	Average

Method Two

	Accuracy	Fold
0	0.998985	1
1	0.895129	2
2	0.646586	3
3	0.926330	4
4	0.999193	5
5	0.893245	Average

Logistic Regression

Trivial Splitting

	Accuracy	Fold
0	0.721922	1
1	0.956620	2
2	0.951416	3
3	0.820678	4
4	0.855470	5
5	0.861221	Average

Method One

	Accuracy	Fold
0	0.873578	1
1	0.870160	2
2	0.868684	3
3	0.867242	4
4	0.867436	5
5	0.869420	Average

Method Two

	Accuracy	Fold
0	0.873455	1
1	0.849329	2
2	0.953029	3
3	0.875898	4
4	0.891483	5
5	0.888639	Average

QDA

Trivial Splitting

	Accuracy	Fold
0	0.717966	1
1	0.967524	2
2	0.959014	3
3	0.831685	4
4	0.836447	5
5	0.862527	Average

Method One

	Accuracy	Fold
0	0.871269	1
1	0.919694	2
2	0.957609	3
3	0.912486	4
4	0.879330	5
5	0.908077	Average

Method Two

	Accuracy	Fold
0	0.827183	1
1	0.985401	2
2	0.971739	3
3	0.929400	4
4	0.882716	5
5	0.919288	Average

Random Forest

Trivial Splitting *Method One* *Method Two*

	Accuracy	Fold
0	0.999974	1
1	0.967472	2
2	0.947044	3
3	0.950583	4
4	1.000000	5
5	0.973014	Average

	Accuracy	Fold
0	1.000000	1
1	0.887166	2
2	0.991725	3
3	0.864214	4
4	1.000000	5
5	0.948621	Average

	Accuracy	Fold
0	1.000000	1
1	0.885292	2
2	0.992896	3
3	0.837827	4
4	1.000000	5
5	0.943203	Average

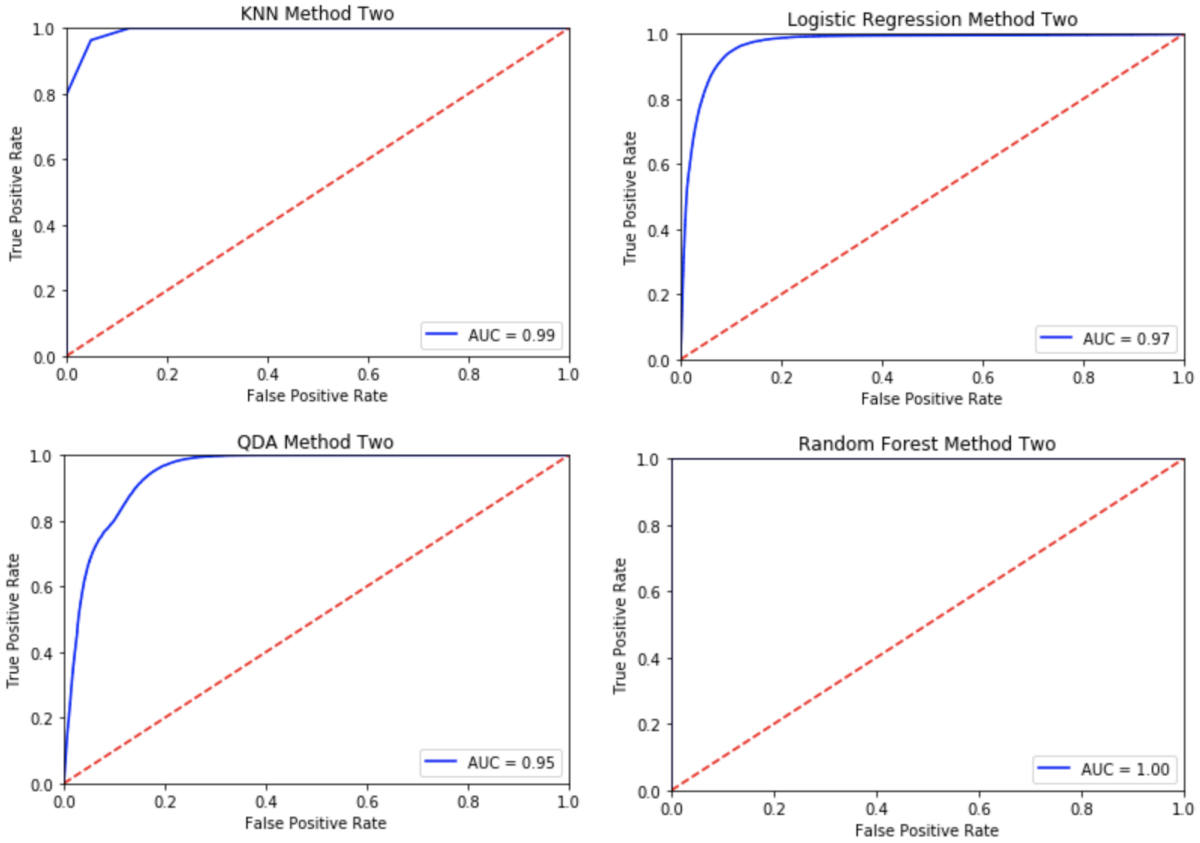
Unfortunately, we were unable to run SVM due to the computational difficulties. At first, we believed that this was due to using too high of a degree polynomial to fit the data, and thus tried to assume the linear form. However, SVM proved too complex for our kernel to run and therefore we left it out in our analysis. If we were determined to run this particular model, we could use a cluster to compute. However, we did receive good results from our other methods and thus felt confident we could leave it out.

For each of the respective models, we dealt with a myriad of assumptions. For KNN, the model is a nonparametric lazy algorithm that makes no assumptions about the underlying data and its form, so we believe that the use is justified. We were also surprised at the accuracy of the model, partly due to the fact that KNN is often considered a simpler model versus its model peers. Logistic regression was a strong choice due to the underlying assumptions being different from its linear counterparts. It does not assume a linear relationship, it does not assume normality of the error terms (which was a large concern in this analysis), and it does not need to assume homoscedasticity. QDA has some assumptions that we were shaky on. One is the assumption that each class is drawn from a normal distribution. When we combine the data, the data appears to be arguably normal, but it is worth noting that when the data was separated into the respective images, some of the data did not appear to be strongly normal. Therefore, QDA may be shaky in terms of the strength of its application. Secondly it also assumes that each class has its own covariance matrix. We believe that this is true and holds, although it not shown here in the paper. Our final model was random forest, which also gave us the highest accuracy. For this model we picked from many different number of estimators, as well as tree depth. We eventually decided to not limit the tree depth and use 50 estimators. This gave us strong accuracy and we also believe that the number of estimators will not have an adverse effect on bias. In terms of assumptions random forest is also a strong model because it has no assumptions about the underlying data, however, assumes that sampling is representative of the data. This may have some wiggle room for argument when one considers the stark differences in terrain and images, but we believe that for sake of modelling this model holds strong, especially when considering that we are using three different images.

As for comparing the different methods of splitting in our CV function, it seemed to us that method two, which involved non trivially splitting the data based on their

coordinate position and resampling, was the strongest method. Therefore, going forward, we will use the data generated from method two in the ROC curve comparisons.

3b)



All of our models obtained very good accuracy and accordingly also strong AUC metrics. The random forest model obtained oddly extremely accurate results. As for a cutoff point we determined that an AUC ~ 0.95 would be appropriate according to our graphs. Also note that there is only one ROC curve per model although we produced three different versions of the model using variants of data splitting.

Diagnostics

4a)

For our analysis we chose to look into the random forest model applied to the fourth fold for splitting method two, which netted us the best accuracy out of our tries. Upon fitting and then predicting using the test data set, we found that the score is an astounding 94%. Random forest is a very popular model used due it's generally high

accuracy as well easily satisfied assumptions. We would like to look into why this model was so successful.

	Variable Importance
NDAI	0.39
SD	0.31
CORR	0.30

We see that the most important variable was actually NDAI, however the variable was fairly similar in terms of importance. We were able to visualize the decision tree itself, however with so many nodes it was no feasible to actually have the image in this paper.

4b)

In order to analyze the misclassification errors, we look at the differences between the correctly guessed and incorrectly guessed units in our testing data set. We look at the differences between the correct and incorrectly guessed rows for our random forest model.

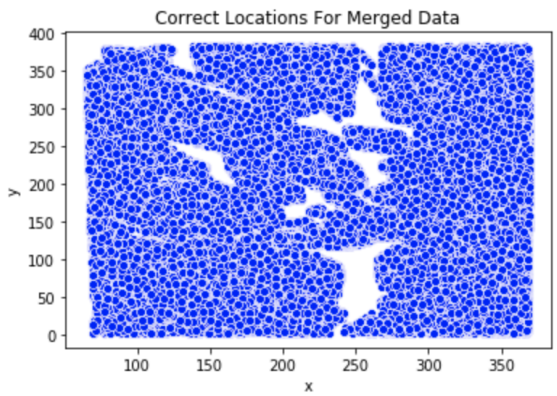
Incorrect

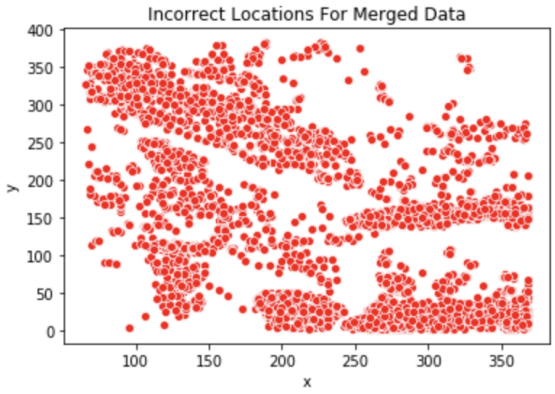
	NDAI	SD	CORR
count	4865.000000	4865.000000	4865.000000
mean	1.374822	8.963276	0.173354
std	1.248512	10.832750	0.081981
min	-1.634195	0.312184	-0.211880
25%	0.651565	2.779175	0.120778
50%	1.449404	5.111278	0.159152
75%	2.344949	10.491170	0.211835
max	4.196991	99.267700	0.721547

Correct

	NDAI	SD	CORR
count	60608.000000	60608.000000	60608.000000
mean	0.611875	6.290293	0.187943
std	1.459273	8.718341	0.111952
min	-1.841971	0.198708	-0.353672
25%	-0.885413	1.004849	0.113629
50%	0.644263	3.052425	0.148052
75%	1.898180	7.789158	0.243183
max	4.338682	110.467640	0.796298

We see that there are some significant differences in the summarized values. Most notably perhaps is the difference in average NDAI, with the incorrectly guessed values having almost twice that of the correct ones. NDAI was also the most important feature in our models from our prior analysis. From an intuitive point this makes sense because we would expect the angular differences to be one of the key points of difference. Next, we are interested in seeing if there are any trends in the misclassifications.





We do see that there are some obvious geographic trends in the misclassified data, most notably the long streak located from 100 ~ x ~ 200, and 200 ~ y ~350. Interestingly this correlates with a similar streak belonging to the image 3 data.

4c)

Based on parts a and b, although our model achieved a strong accuracy, there are still problematic spots it seems. Areas with a large NDAI seem to be more likely to be incorrectly classified, which creates problem areas geographically, as shown above. Overall however we believe that other models would have this same problem, as this goes beyond the model algorithm. We reasonably believe that our model would still perform very well without expert labels, as the overall accuracy is still top notch even in the face of the cyclical problems. However, an extra thing to consider is the efficiency of random forest, which is known as a very complex model and takes very long to train.

4d)

Our models generally did not have large differences resulting from how we split our data luckily. This may be tribute to how the data is not so skewed as we would believe.

4e)

Overall there are still some problems that are difficult to circumvent as mentioned prior above. One possible way to improve the results would be to introduce new features, perhaps implementing one of the radiance angles. However, we were pleased with our overall scores from our models and were surprised especially at the effectiveness of the KNN. When thinking about actually implementing in the field one must consider that terrain images are very different from one and the other, and so we must assume that the data will still be similar enough for our model to stay strong.

Github Link : https://github.com/Johnjuantae/Project2_154