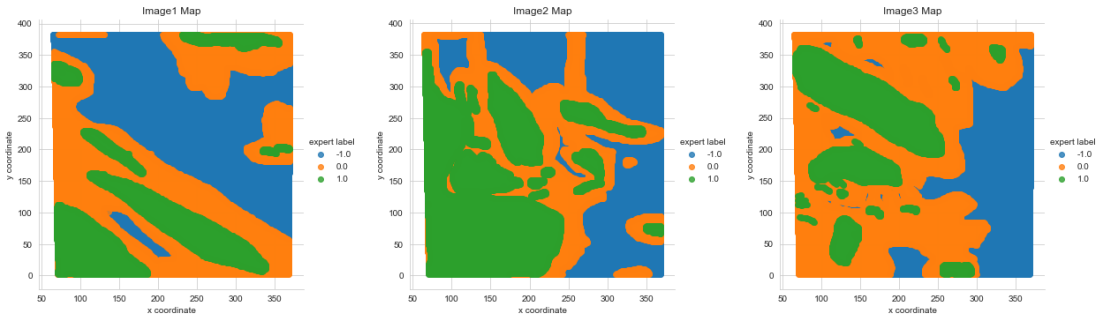
Stat 154: Project 2 Cloud Data

Yilin Ye, SID 3031977092
Shengyi Wu, SID 3032004808

1 Data Collection and Exploration

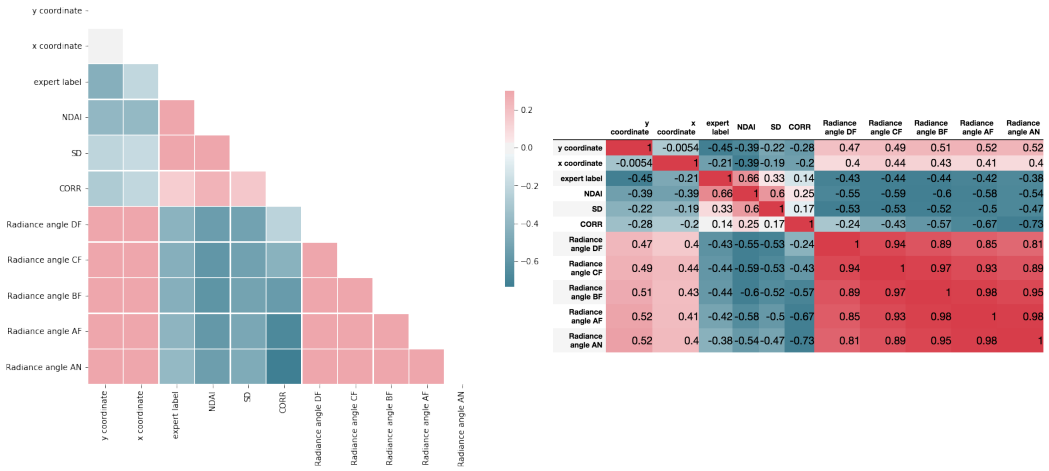
(a) **Summary:** Most global climate models predict that the strongest dependencies of surface air temperatures on increasing atmospheric carbon dioxide levels will occur in the Arctic. To systematically study these dependencies, researchers need accurate Arctic-wide measurements, especially of cloud coverage. However, because liquid- and ice-water cloud particles often have similar remote sensing characteristics to those of the particles that compose ice- and snow-covered surfaces, its hard for standard classification frameworks to characterize clouds accurately. Therefore, the paper proposes two new operational Arctic cloud detection algorithms which use and process data from Multiangle Imaging SpectroRadiometer (MISR). Unlike traditional multispectual sensors that take measurements in a single view, the MISR sensor comprises nice cameras with each camera viewing Earth scenes at a different angle. These algorithms also include novel method which search for cloud-free instead of cloudy ice- and snow-covered surface image pixels because the surface does not change much between different views. Additionally, three physically useful features including the correlation (CORR) of MISR images of the same scene from different MISR viewing directions, the standard deviation (SD_{An}) of MISE nadir camera pixel values across a scene, and a normalized difference angular index (NDAI) that characterizes the changes in a scene with changes in the MISE view direction. The first algorithm, enhanced linear correlation matching (ELCM), sets the features with either fixed or data-adaptive cutoff values and obtains probability labels using ELCM labels as training data for Fishers quadratic discriminant analysis (QDA), which leads to the second algorithm called ELCM-QDA. The results show that ELCM results are significant for both accuracy (92%) and coverage (100%). The ELCM-QDA probability prediction is also consistent with the expert labels. As a result, both ELCM and ELCM-QDA perform the best to date among all available operational algorithms using MISR data.

(b) **1. Summarize the data**: For image1, 43.78% of pixels are labeled as not cloud(-1). 38.46% of pixels are unlabeled(0). 17.77% of pixels are labeled as cloud(1). For image2, 37.25% of pixels are labeled as not cloud(-1). 28.64% of pixels are unlabeled(0). 34.11% of pixels are labeled as cloud(1). For image3, 29.29% of pixels are labeled as not cloud(-1). 58.27% of pixels are unlabeled(0). 18.44% of pixels are labeled as cloud(1).
2. Plot the observation: Image1 Map shows that pixels labeled as cloud and unlabeled are mostly on left lower part of the plot and pixels labeled as not cloud are mostly on the right upper part of the plot. Image2 Map shows that pixels labeled as cloud are mostly on left, pixels unlabeled are mostly in the middle, and pixels labeled as not cloud are mostly on the right. Image 3 shows that pixels labeled as cloud and unlabeled are mostly on left upper part of the plot and pixels labeled as not cloud are mostly on the right lower part of the plot. There are always pixels unlabeled between pixels labeled as cloud and pixels labeled as not cloud, which show that researchers are not sure what is at the boundary of cloud and not cloud. Pixels labeled as not cloud always cluster together and pixels labeled as cloud are more sparse. The three maps show that an i.i.d. assumption for the samples is not justified for the data set because there exists strong spatial correlation between data point.



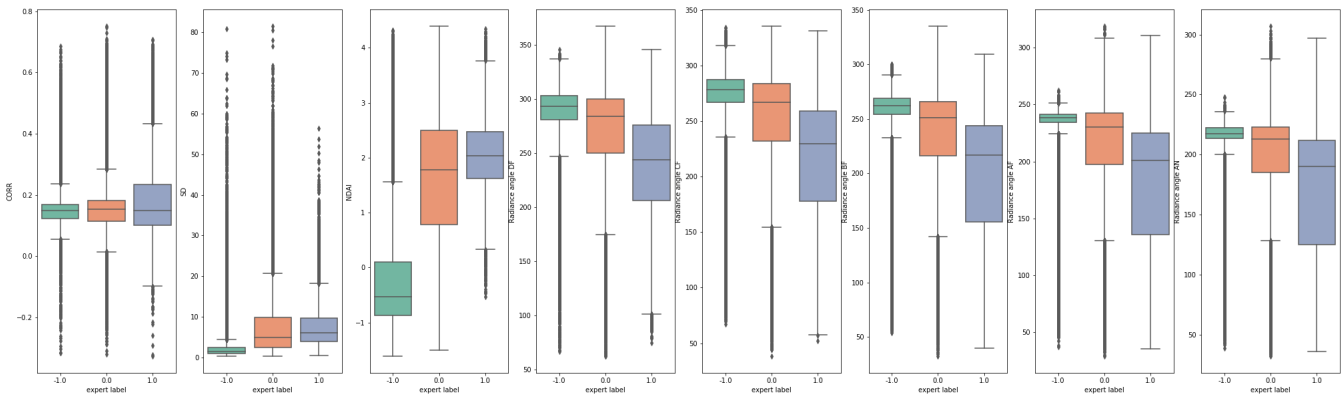
(c) **Visual and Quantitative EDA:**

(i) To understand the pairwise relationship between the features, we plotted a correlation heat-map and a correlation map for all variables. These two plots (for image1) show that all *Radiance angle* variables are highly positively (corr≈0.81) correlated with each other. $NDAI$, SD, and $CORR$ are all negatively correlated with all *Radiance angle* variables. $NDAI$ is positively correlated with SD (corr=0.6), positively correlated with $CORR$ (corr=0.25). SD is slightly positively correlated with $CORR$ (corr=0.17).
(ii) *expert label* is positively correlated with $NDAI$ (corr=0.66), SD (corr=0.33), $CORR$ (corr=0.14). *expertlabel* is negatively correlated to all *Radiance angle* variables (corr≈-0.4).



	y coordinate	x coordinate	expert label	NDAI	SD	CORR	Radiance angle DF	Radiance angle CF	Radiance angle BF	Radiance angle AF	Radiance angle AN
y coordinate	1	-0.0054	-0.45	-0.39	-0.22	-0.28	0.47	0.49	0.51	0.52	0.52
x coordinate	-0.0054	1	-0.21	-0.39	-0.19	-0.2	0.4	0.44	0.43	0.41	0.4
expert label	-0.45	-0.21	1	0.66	0.33	0.14	-0.43	-0.44	-0.44	-0.42	-0.38
NDAI	-0.39	-0.39	0.66	1	0.6	0.25	-0.55	-0.59	-0.6	-0.58	-0.54
SD	-0.22	-0.19	0.33	0.6	1	0.17	-0.53	-0.53	-0.52	-0.5	-0.47
CORR	-0.28	-0.2	0.14	0.25	0.17	1	-0.24	-0.43	-0.57	-0.67	-0.73
Radiance angle DF	0.47	0.4	-0.43	-0.55	-0.53	-0.24	1	0.94	0.89	0.85	0.81
Radiance angle CF	0.49	0.44	-0.44	-0.59	-0.53	-0.43	0.94	1	0.97	0.93	0.89
Radiance angle BF	0.51	0.43	-0.44	-0.6	-0.52	-0.57	0.89	0.97	1	0.98	0.95
Radiance angle AF	0.52	0.41	-0.42	-0.58	-0.5	-0.67	0.85	0.93	0.98	1	0.98
Radiance angle AN	0.52	0.4	-0.38	-0.54	-0.47	-0.73	0.81	0.89	0.95	0.98	1

To understand the differences between two classes(cloud, no cloud), we plotted a box-plot for each variable. For $CORR$, cloud and not cloud classes have similar median value, but cloud class has wider interquartile range than not cloud class. For SD, not cloud class has lower median value than cloud class, and cloud class has much wider interquartile range than not cloud class. For $NDAI$, not cloud class has much lower median value than cloud class. For all *Radiance angle* variables, not cloud class has higher median value than the cloud class, but the cloud class always has much wider interquartile range than not cloud class.



2. Preparation

(a) **Data Split:** The data we used in this project are geospatial data in three images. As each pixel is a part of one picture and spatially correlated with each other, we can't use naive approach (random split) which will disrupt the spatial information that these data contain. Therefore, we decided to split the data into smaller groups and then randomly shuffle based on two methods. In Method 1, we computed the ranges of x-coordinate and y-coordinate and found that all three images are within same ranges (only the range of x-coordinate for image 2 is off by 1). Hence, we assumed that all data can be grouped together. For the whole merged data, we divided the ranges of x-coordinate and y-coordinate into 20 groups and split the data into 400 small data chunks according to their x-coordinate and y-coordinate. 70%, 10 %, and 10% of the 400 data chunks were randomly selected as training set, validation set, and test set respectively. (*Train, Val, Test data obtained by this method will be referred to as Method 1 data in the report below.) In Method 2, through EDA, we found that cloud classification in image1 and image3 can be split by diagonal line with negative slope. Thus, we assumed that we can use the same method to split data in these two images. We split the data for each of image1 and image3 into 130 groups by 129 lines that have the same slope as the downward diagonal line with different intercepts. For image 2, we observed that the classification are mostly vertical. Therefore, to be consistent with the method of the other two images, we split data for image2 vertically into 130 groups according to their x-coordinate. Eventually, we combined all the groups for three images together and randomly select 70%, 10 %, and 10% of the 390 pieces as training set, validation set, and test set. (*Train, Val, Test data obtained by this method will be referred to as Method 2 data in the report below.)

(b) **Baseline:** The accuracy of the trivial classifier for Method 1 test set is around 58.169%. The accuracy of the trivial classifier for Method 1 validation set is around 60.705%. The classification accuracy for both validation and test set is similar, so the model trained by training data can apply to test data better. The accuracy of the trivial classifier for Method 2 test set is around 58.017%. The accuracy of the trivial classifier for Method 2 validation set is around 60.607%. The accuracy for the trivial classifier for Method 1 and Method 2 is similar so in future analysis, we know that different result is not due to the difference in data (baseline). Additionally, because all data in the trivial classifier are labeled as cloud-less, such a classifier have high average accuracy for data of cloud-less area.

(c) **First order importance:** The "best" feature criteria that we used are strong correlation with *expert label* as well as big difference between cloud and not cloud within the variables. According to the two correlation plots in 1(c), we can see that *expert label* is most strongly positively correlated with $NDAI$ (corr = 0.5). Therefore, we chose $NDAI$ as our first feature. Also, the correlation plots in 1(c) show that $CORR$ is also positively correlated with *expert label* (corr = 0.34). Even though *expert label* vs $CORR$ correlation is not as strong as *expert label* vs $NDAI$ correlation, the box-plot for $NDAI$ shows huge difference in median value between cloud class and not cloud class. Hence, the second feature we chose is $CORR$. According to the box-plots for all *Radiance angle* variables, they all shows huge difference in the median values between cloud class and not cloud class. Taking the correlation into consideration, *Radiance angle AN* shows the strongest negative correlation with *expert label* (corr=-0.17). Therefore, we chose *Radiance angle AN* as the third feature.

	y coordinate	x coordinate	expert label	NDAI	SD	CORR	Radiance angle DF	Radiance angle CF	Radiance angle BF	Radiance angle AF	Radiance angle AN
y coordinate	1	-0.0055	0.21	-0.19	-0.28	0.09	0.5	0.54	0.53	0.47	0.41
x coordinate	-0.0055	1	-0.57	-0.68	-0.5	-0.39	-0.0014	0.17	0.26	0.33	0.37
expert label	0.21	-0.57	1	0.5	0.24	0.34	0.14	0.022	-0.057	-0.13	-0.17

(d) **CV generic** see github code

3. Modeling

(a) In the modeling part, we used 4 different models including logistic regression, QDA, KNN, and random forest, and trained each model based on all features. Also, as the problem suggested, we merged both training set, and validation set to train the model, and test on test set.

Logistic Regression:

Assumptions :

Firstly, binary logistic regression requires the dependent variable to be binary and ordinal logistic regression requires the dependent variable to be ordinal. The dependent variable in our case is *expert label*, which has -1 (not cloud) and 1 (cloud) values. These values are binary and ordinal. Secondly, logistic regression requires the observations to be independent of each other. In other words, the observations should not come from repeated measurements or matched data. Even though the original data contains spatial correlation information, after we splitting them into smaller pieces and randomly shuffling them, the correlation was reduced. Therefore, we can assume the the current data are independent of each other. Thirdly, logistic regression requires there to be little or no multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other. Except *Radiance angle* variables, all other variable don't have very strong correlation with each other. Finally, logistic regression typically requires a large sample size. Totally, there are 345449 data in three images, which is a pretty big data set.

Logistic Loss & Accuracy Rates(Test & Val):

Table 1: Logistic Regression CV Results (Method 1 Data)

CV#	**Logistic Loss**	**Val Accuracy(%)**	**Test Accuracy(%)**
1	0.229	91.47	91.69
2	0.257	89.69	91.45
3	0.354	86.0	91.98
4	0.279	87.81	90.83
5	0.272	88.08	91.0
Average	0.278	88.61	91.39

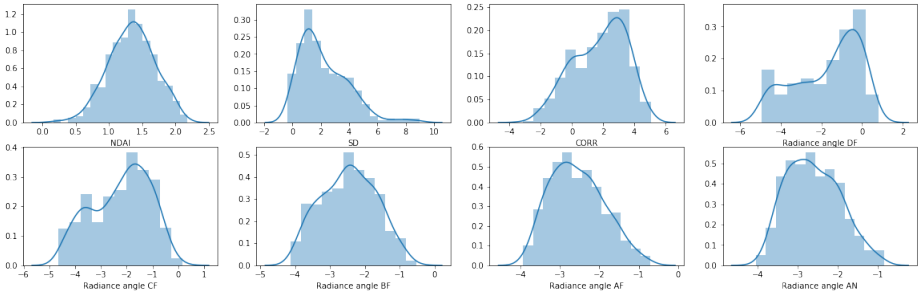
Table 2: Logistic Regression CV Results (Method 2 Data)

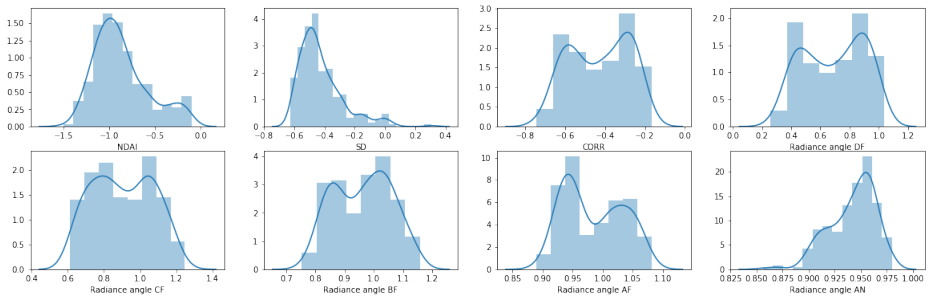
CV#	**Logistic Loss**	**Val Accuracy(%)**	**Test Accuracy(%)**
1	0.275	89.93	91.23
2	0.273	89.07	91.33
3	0.287	87.11	91.36
4	0.317	86.41	91.36
5	0.217	91.19	91.25
Average	0.278	88.74	91.31

QDA:

Assumptions :

QDA assumes the predictor variables X are drawn from a normal distribution. After we normalized the data, we can see the the distribution of most of variables are approximately normal. Also, QDA requires the number of predictor variables (p) to be less then the sample size (n). AS we only have 9 features but more than 30 thousand data, this assumption is met.





Logistic Loss & Accuracy Rates(Test & Val):

Table 3: QDA CV Results (Method 1 Data)

CV#	**Logistic Loss**	**Val Accuracy(%)**	**Test Accuracy(%)**
1	0.495	91.60	92.13
2	0.707	89.07	92.00
3	1.263	87.04	92.04
4	0.562	88.21	92.04
5	0.789	87.16	91.83
Average	0.763	88.62	92.01

Table 4: QDA CV Results (Method 2 Data)

CV#	**Logistic Loss**	**Val Accuracy(%)**	**Test Accuracy(%)**
1	0.731	89.37	89.47
2	0.782	89.79	89.72
3	0.690	86.93	89.52
4	0.973	8667	89.71
5	0.601	91.67	89.38
Average	0.755	88.89	89.56

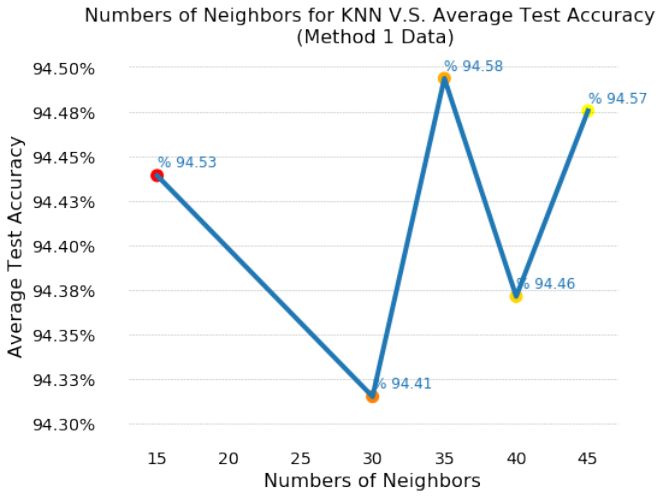
k-nearest neighbors algorithm (KNN):

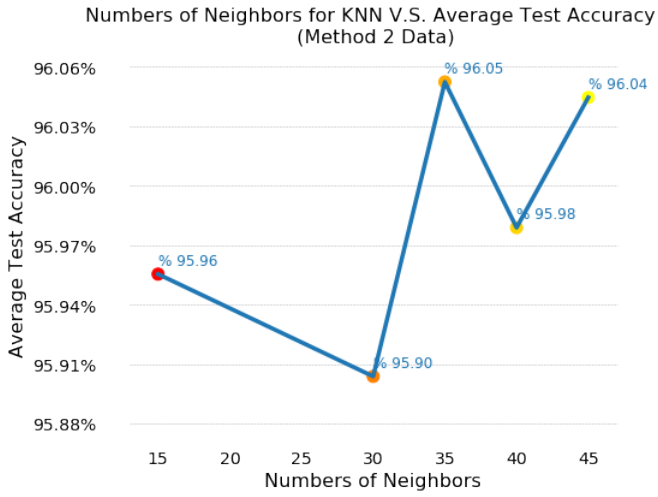
Assumptions :

Because KNN classifier is a non parametric and instance-based learning algorithm, there is no explicit assumptions for underlying data distribution.

Comment :

We did a hyper-parameter tuning on our KNN model. We tried numbers of neighbours equal to 15, 30, 35, 40, and 45, and found that when the number of k-nearest neighbours is 35, the test accuracy achieves its highest. Therefor, we chose to use the CV-3 data with n =35.





Logistic Loss & Accuracy Rates(Test & Val):

Table 5: KNN CV Results (Method 1 Data)

CV#	Logistic Loss	Val Accuracy(%)	Test Accuracy(%)
1	0.184	95.17	95.13
2	0.326	91.95	94.20
3	0.213	94.14	94.43
Average	0.241	93.75	94.43

Table 6: KNN CV Results (Method 2 Data)

CV#	Logistic Loss	Val Accuracy(%)	Test Accuracy(%)
1	0.172	94.22	95.94
2	0.239	93.50	96.18
3	0.153	95.40	96.03
Average	0.188	94.37	96.05

Random Forest:

Assumptions :

There is no formal distributional assumptions for random forests. Random forest is a non-parametric model and can thus handle skewed and multi-modal data as well as categorical data that are ordinal or non-ordinal.

Comments :

After similar hyper-parameter tuning method as in K-nearest neighbor part, we found that the larger the numbers of estimator, the better test accuracy we obtain. However, it takes longer time when the number of estimators is too large. Therefore, we set our numbers of estimators to be 1000. Also, after testing the model with different hyper-parameters, we found that our model perform the best for both ways of splitting the data when max depth for the forest to be 5, and minimum nodes to split to be 3 with entropy loss. So, we set the hyper-parameters as above when training the data.

Logistic Loss & Accuracy Rates(Test & Val):

Commentary

Through reporting the validation accuracy and test accuracy across each cross validation fold of each model, we found that Random Forest performs the best, with average validation accuracy 91.79% and average test

Table 7: Random Forest CV Results (Method 1 Data)

CV#	**Logistic Loss**	**Val Accuracy(%)**	**Test Accuracy(%)**
1	0.156	93.08	95.55
2	0.189	91.47	95.62
3	0.248	88.43	95.62
4	0.183	93.09	95.54
5	0.175	92.90	95.64
Average	0.190	91.79	95.59

Table 8: Random Forest CV Results (Method 2 Data)

CV#	**Logistic Loss**	**Val Accuracy(%)**	**Test Accuracy(%)**
1	0.173	93.38	95.21
2	0.197	91.40	95.40
3	0.203	90.39	95.39
4	0.214	89.94	95.06
5	0.145	93.81	95.16
Average	0.186	91.79	95.24

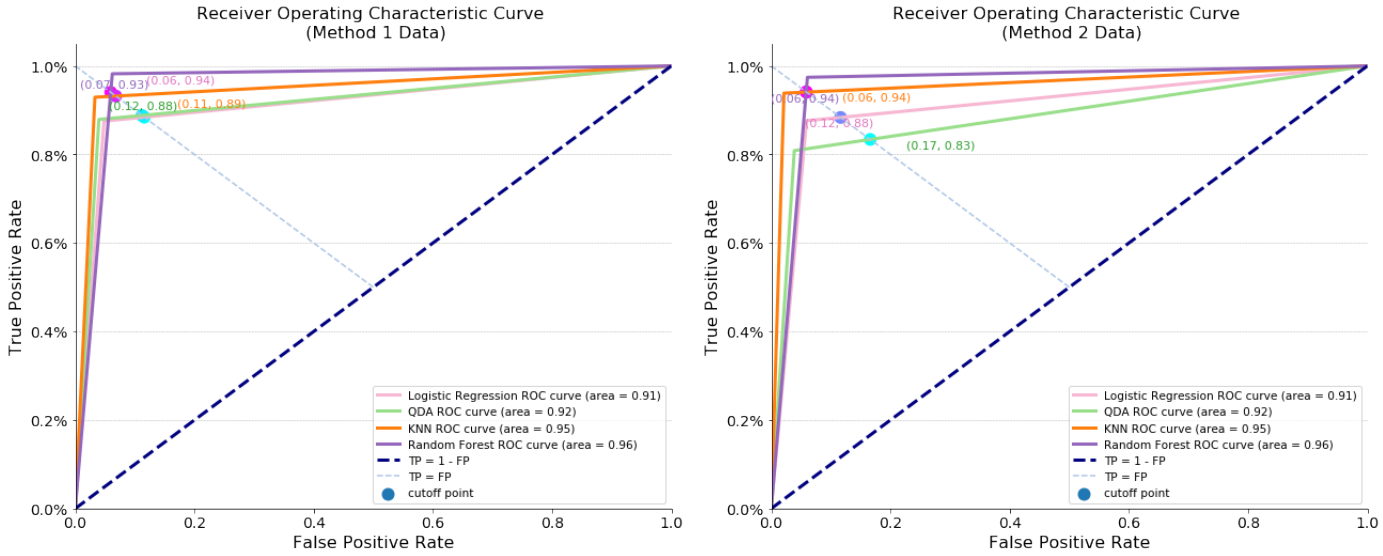
accuracy 95.59%. KNN model performs pretty well too, with average validation accuracy 93.75% and average test accuracy 94.43%. Logistic regression perform well as well, with average validation accuracy 88.74% and average test accuracy 91.39%. QDA doesn't perform as well as the other three models, probably because some variables don't fulfill the assumption of normality that QDA requires.

(b) The ROC cruve summarizes the model's performance by evaluating the trade offs between true positive rate(TP) and false positive rate(FP). The higher the area under curve(AUC) is, the better the prediction power of the model. Therefore, from the ROC graphs below, we observe that for both Method 1 Data and Method 2 Data, the Random Forest Classifier has the best performance(AUC = 0.96), and kNN(AUC = 0.95) classifier also has pretty good performance. For both data, QDA Classifier has the worst performance. But, overall, all of our classifiers for both data set have all performed much better than the ROC baseline. (the navy dotted straight diagonal lines on the graphs).

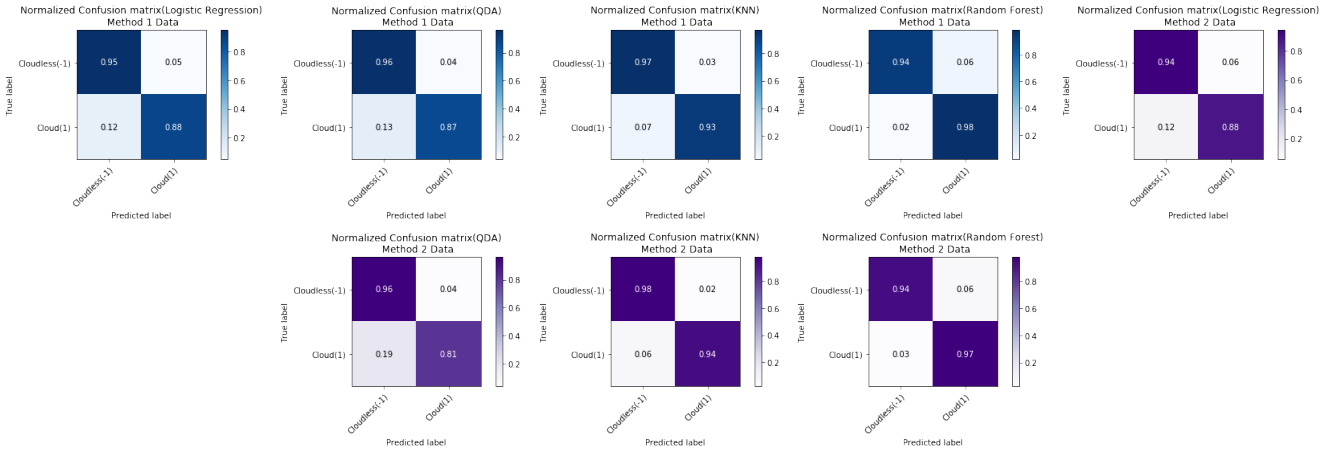
Optimal Cutoff Point:

For our cloud data set, there isn't any obvious need to either favor True Positive rate to label more images as cloudless or favor False Positive rate to label more images as cloud. Therefore, we decide to choose the cut-off value where the test True Positive rate is equivalent to the test False Positive Rate.

Hence, we obtain the optimal cutoff points for each classifier by finding the intersection of the line connecting the left-upper corner and the right-lower corner of the unit square (the line TP = FP), and the ROC curve. (each point is highlighted on the graphs)



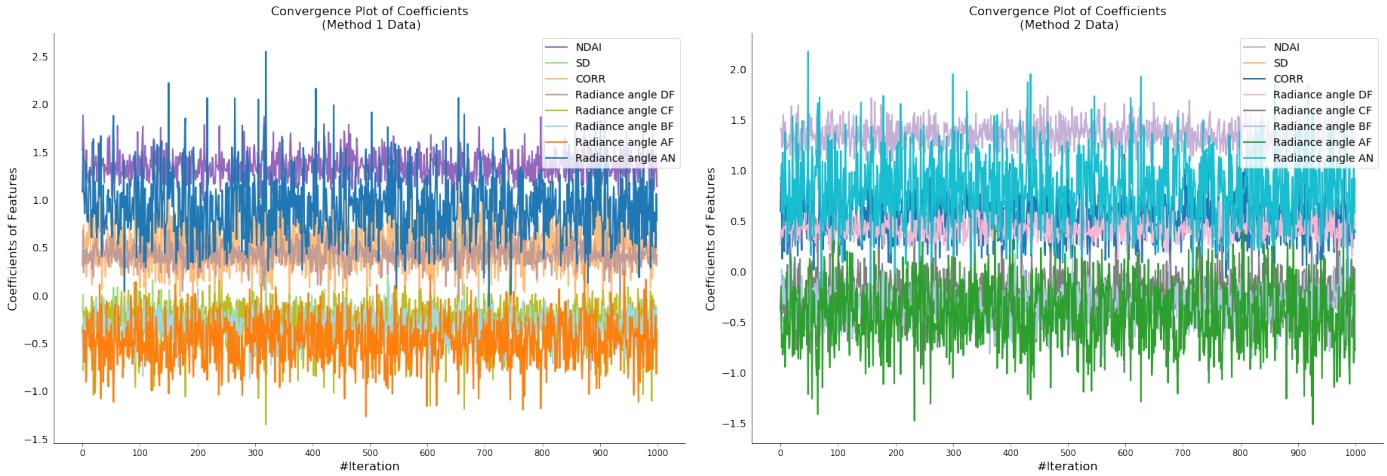
(c) We suggest to use the normalized confusion matrix for better evaluation of the classifiers, since normalized confusion matrix can help us conveniently visualize the respective performance of each classifier on cloudless pixel and cloud pixel. For each classifier, we graphed their normalized confusion matrix using models that are trained on all features and the combinations of training and validation set as the problem suggested. (Blue represents the Method 1 Data, and purple represents the Method 2 Data.)
For the Method 1 data, we observe that kNN has the highest accuracy rate on cloudless pixel(0.97), Random Forest has the worst performance(0.94). For the cloud pixel, Random Forest has the best performance(0.98), and QDA has the worst performance(0.88).
For the Method 2 data, it's shown on the graph that kNN also has the highest accuracy rate on cloudless pixels, and both Logistic Regression and Random Forest both have the worst performance(0.94). For the cloud pixels, similar to method 1 data, Random Forest has the best performance(0.97), and QDA has the worst accuracy rate(0.81)
Overall, for both data, Random Forest has the best performance on identifying the true cloud pixel but has the tendency to misclassify the cloudless pixel as cloud pixel. Contrary to Random Forest, kNN has the best performance on identifying the cloudless pixel, but has the tendency to misclassify cloud pixel as cloudless pixel. QDA perform the worst for the cloud pixel on both data sets.



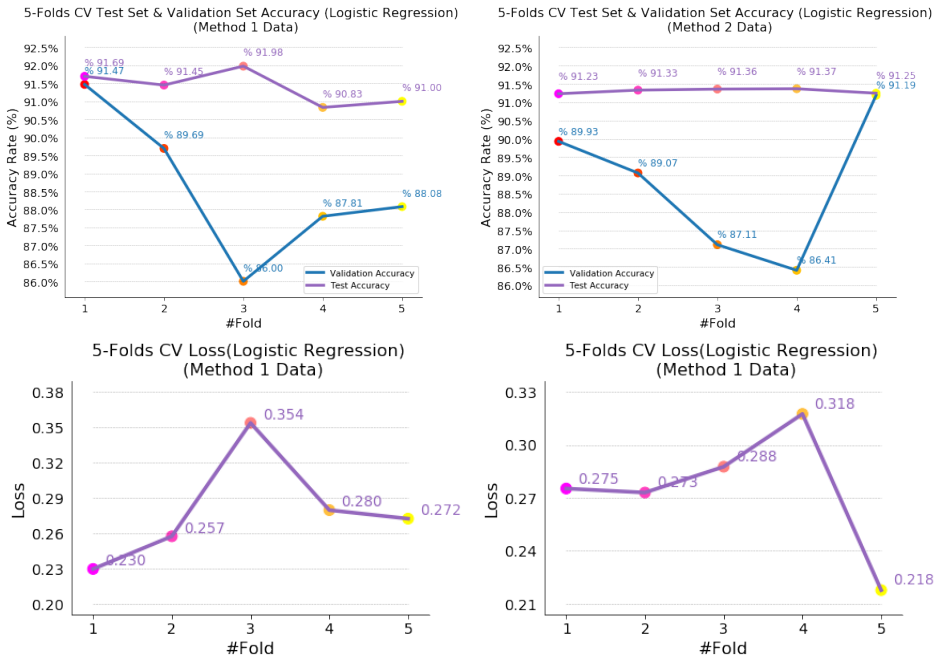
4. Diagnostics

(a) From the above analysis based on ROC curve and Confusion Matrix, we observe that logistic regression performed decently well. Also since the assumption for logistic regression is that the observations are independent from each other, and our raw data is geospatially correlated, we are curious to further examine the performance

of logistic regression. Therefore, we decide to do an in-depth analysis of it.



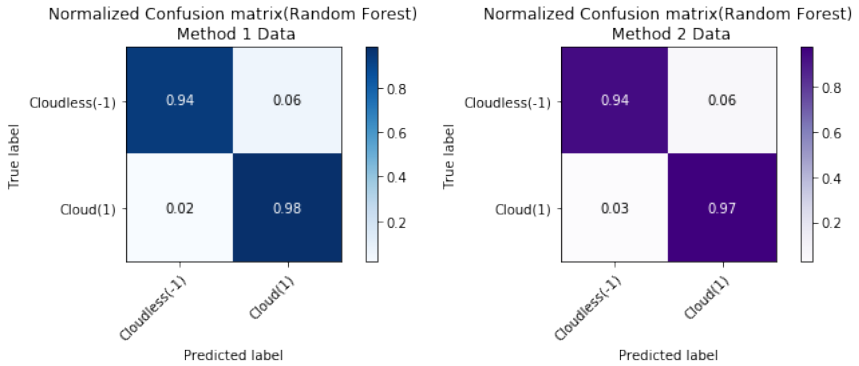
First, we want to analyze the convergence of the coefficients for 8 features of logistic regression. So, we first fit the logistic regression classifier on the *train_val* data set of two data sets respectively for 1000 iterations, record and graph the coefficients for 8 features for each iteration.
From the above two graphs, we observe that for each of the 8 features, although their coefficients oscillate, all of them oscillate within relatively small bands, and are relatively stable throughout the 1000 iterations. The graph shows that the coefficients of $NDAI$, $CORR$, $RaidanceAngleDF$, $RadianceAngleBF$, and $RadianceAngleCF$ oscillate between their mean$\pm$0.8, and are relatively constant overtime. For both data set, the coefficients of Radiance Angle AF, and Radiance Angle AN oscillate the most. The range of their oscillation is around 2.5.



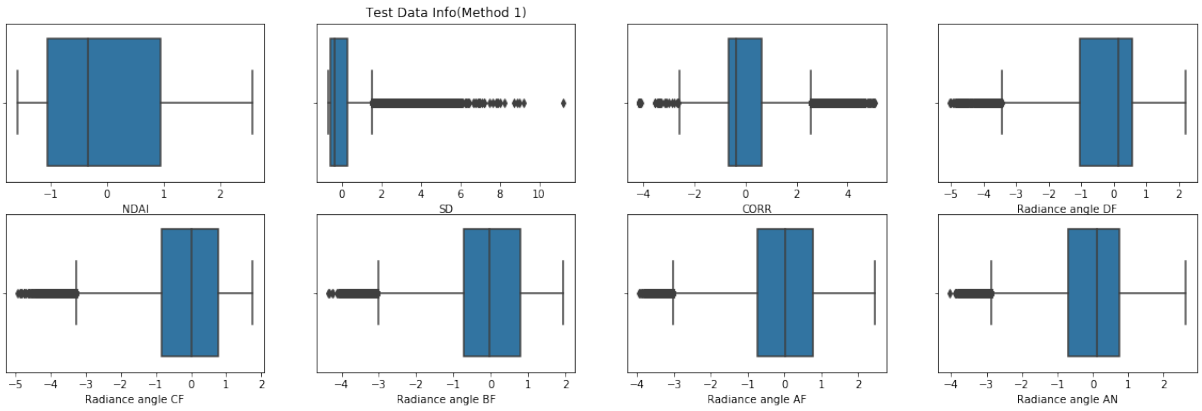
From the above graph of logistic loss, validation and test accuracy, we can see that the test set accuracy always are better than the validation set accuracy, and the test accuracy are relatively more stable than the validation accuracy across 5 folders, which is a good sign since it implies that the logistic regression classifier may perform well on the future data set.
According to the normalized confusion matrix of logistic regression in 3c.), we observe that logistic regression performs pretty well on the cloudless pixels. It successfully identify 95% of the cloudless pixels as cloudless on method 1 data, and 94% of the cloudless pixels as cloudless on method 2 data. However, logistic regression

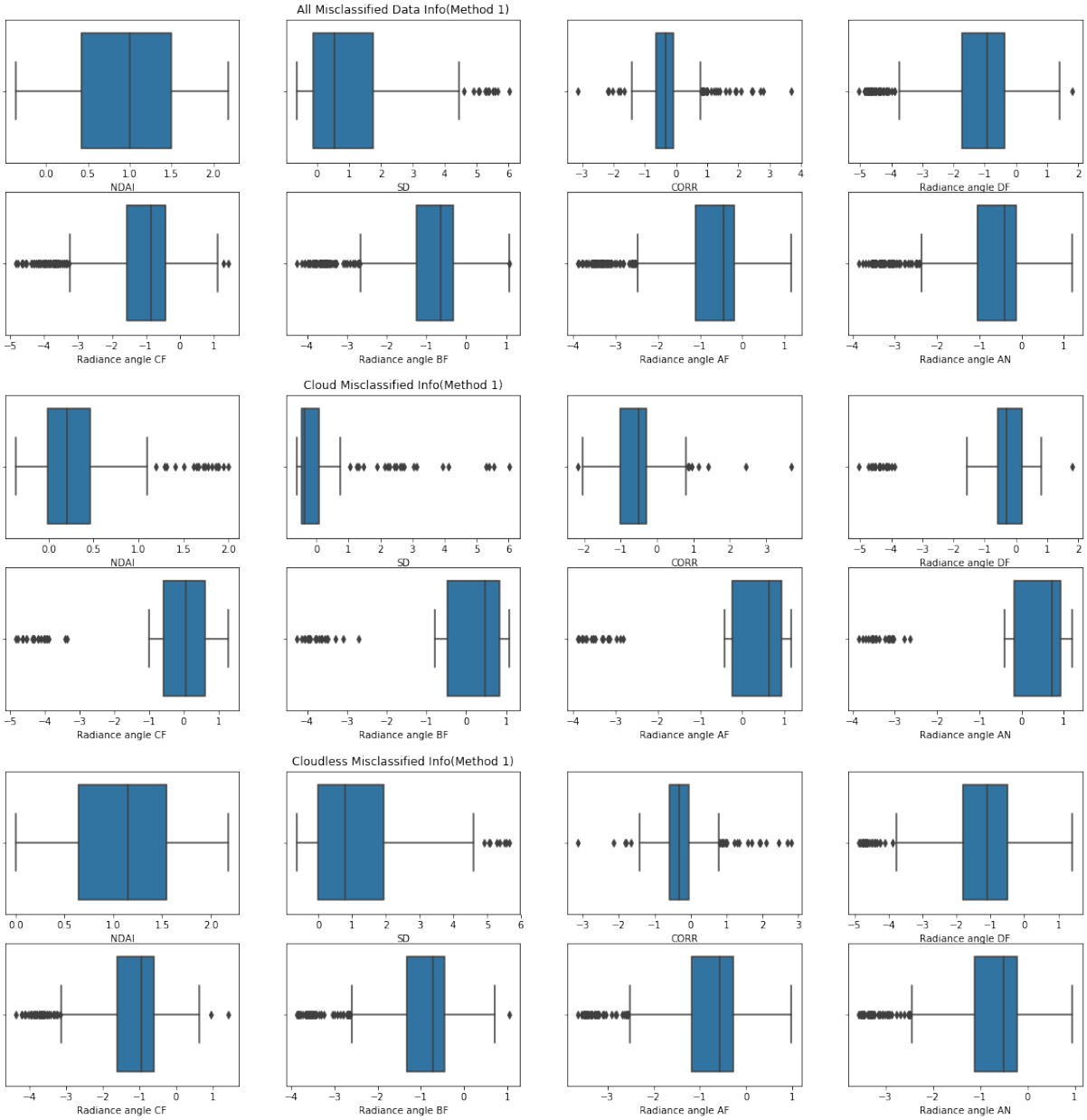
classifier has a hard time to label the cloud pixel as cloud. It only has 88% accuracy rate for cloudless data on both method 1 and method 2 data. We can see that logistic regression has a tendency to label a pixel as cloudless.

(b) *Misclassification*: The best classification model we have is random forest based on the average test accuracy. According to these two confusion matrices, for data split by both methods, random forest model does very well in classifying cloud pixels as cloud class (high true positive rate). On the other hand, it's slightly more likely to classify cloudless pixels as cloud. Therefore, there are more misclassification of cloudless pixel as cloud pixels than that of cloud pixels as cloudless pixel.



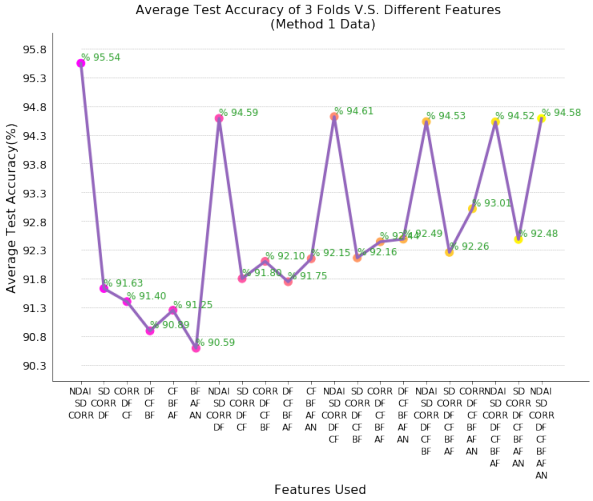
The first two rows of the plots below are boxplots for all 8 features in test data set. The second two rows of the plots are boxplots for all 8 featueres in combined misclassification data set (including classifying cloudless as cloud and cloud as cloudless). The third and forth two rows of plots are boxplots for all features in cloud misclassification (as cloudless) data set and in cloudless misclassification (as cloud) data set.
According to the boxplots for test set data and all misclassifying data, variables such as $NDAI$, SD, $CORR$ in all misclassifying data have higher median than those in test set data. Comparing boxplots for cloud misclassifying data and for cloudless misclassifying data, we found that variables such as $NDAI$, SD, $CORR$ in cloudless misclassifying data have higher median than those in cloud misclassifying data. Also, for all *Radiance angle* variables in cloudless misclassifying data, their interquartile ranges are all much wider than those in cloud misclassifying data. Therefore, we concluded that data misclassification happens more frequently when $NDAI$, SD, $CORR$ values are higher. Cloudless misclassifying data has wider interquartile range for all *Radiance angle* variables than cloud misclassifying data. We observed similar pattern for data split by both methods, so we only showed the boxplots for data split by Method 1.

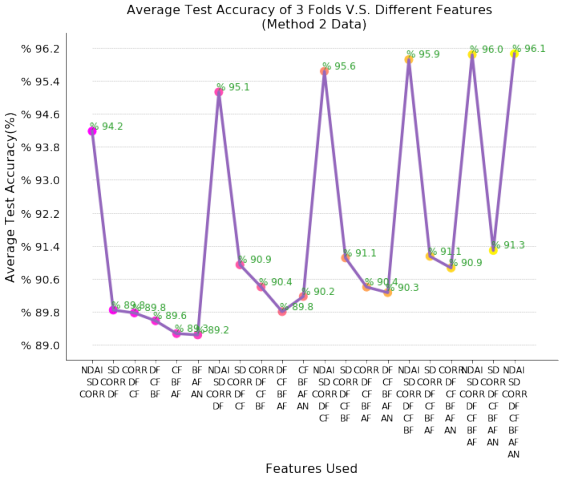




(c) *Better classifier* : We used all features to fit our logistic regression model in 4(a) and we noticed some misclassification problems in our random forest model. According to the confusion matrices in 3(c), kNN performs equally well in data split by both methods, so we used kNN model for further improvement on feature selection. For method 1, the model works best (95.54% average test accuracy) when we use features $NDAI$, SD, and $CORR$, which is consistent with the feature selection on the Yu's paper. On the other hand, if we applied kNN model on data split by Method 2, it works best (96.1% average test accuracy) if we include all 8 features. As a result, when future data come in, if we split them according to Method 1, it's the best to include $NDAI$, SD, and $CORR$ features; if we split them according to Method 2, it's best to include all features. As long as the future data are similar to our data, we are pretty confident about our model because of high test accuracy rate.

Also, we think the convolutional neural network can perform better on this data set than all of the four classifiers we have tried since we dont need to assume any underlying pattern for the data for neural network, and through exhaustive search, the neural net can almost capture all patterns. However, because of the runtime and GPU limit, we couldn't run the convolutional neural network on our data.





(d) For our two different ways of splitting the data, our results in 4a.) and 4b.) are pretty similar.
In 4a.), from the two plots of logistic regression's coefficients estimations, we observe that the results are close. All of the eight features have similar means of oscillation, and similar ranges of oscillation. Both *Radiance Angle AN*, and *Radiance Angle AF* oscillates the most, and *NDAI* and *Radiance Angle DF* oscillate the least on both data set.
Also from the 5 folds Test Set & Validation Set Accuracy and Logistic Loss graphs in 4a.), we find out that the ways of splitting the data didn't affect the results too much. The test accuracy rate for both ways of splitting the data is around 90.5% and 92%; the validation set accuracy is around 86% and 91.2%; the logistic loss for both is around 0.21 and 0.36. Besides the results for logistic regression, in 4b.), the confusion matrices for random forest show similar pattern. As a result, no matter we split the data either by Method 1 or Method 2, we have similar clasification results.

(e) *Conclusion* : There are mainly two parts that we worked on for this project. The first main part is EDA. Through extensively exploratory data analysis, we found that *NDAI*, *SD*, and *CORR* are negatively correlated with *expert label* and all *Radiance angle* variables are negatively correlated with *expert label*. Also, we observed that except for *SD*, the distribution of data for cloud class and not cloud class for other variables are quite different. The second main part is modeling. We applied logistic regression, KNN, QDA, and Random Forest on training and validation set, and trained them on all features for two different ways of splitting the data. Overall, for data split by Method 1 and Method 2, Random Forest has the best performance on identifying the true cloud pixel but has the tendency to misclassify the cloudless pixel as cloud pixel. On the contrary, kNN has the best performance on identifying the cloudless pixel, but has the tendency to misclassify cloud pixel as cloudless pixel. QDA perform the worst for the cloud pixel on both data sets. This might because some of the variables we fitted in the model don't fulfill the assumption of normality that QDA requires.

5. Reproducibility

Github Links:
https://github.com/angelynaye/Stat-154-Project-2

6. Contribution of Partners

Both Shengyi Wu and Yilin Ye's contribution are equally important to the completion of this project. There's no conflict, and we cooperate with each other pretty well.