Cloud Detection at Poles with Bancroft Transformation

Flying Ramen Pokemon
Ling Xie 26826715 and Chenyang Zhu 3034425086
{xieling, chenyang.zhu}@berkeley.edu

May 4, 2019

Abstract

In this project, we are aiming to detect cloudiness at pixel-level, given with coordinates, MISR data and three designed features of each pixel. We created a new method called Bancroft Transformation. It collects information of all surrounding pixels, creates new features and adds them to pixel of interest. With this method, we are able to solve the problems of balanced splitting and spatial splitting methods and improve our classification accuracy. Our code is available at https://github.com/chenyangzhu/stat154-project2.

1 Data Collection and Exploration

1.1 Paper Summary

This paper proposes two algorithms to detect clouds using the image data collected from 10 MISR orbits of path 26 over Arctic, northern Greenland, and Baffin Bay. Their algorithms are based on three physical features: CORR, SD and NDAI, where CORR is the correlation of MISR images of the same scene taken from different viewing directions, SD is the standard deviation of MISR nadir camera pixel values across a scene and NDAI is the normalized difference angular index that characterizes the changes in a scene with changes in the MISR viewing direction. The researcher built the first algorithm (ELCM) by thresholding the three features with certain cutoff values and apply the algorithm to each data unit to produce the labels. The second algorithm (Fishers QDA) takes the labels output from the first algorithm and provides an estimate of probability of cloudiness for each data unit. This work provides support to current weather and climate research by helping researcher study the cloud coverage, particularly at the poles. Furthermore, this work also demonstrates the importance of statistical thinking in approaching a specific scientific problem.

1.2 Data Exploration

The data set contains three images with information of each of its pixels where eleven features are given to each pixel. The `NDAI`, `SD`, and `CORR` are computed matrices of radiations and the `DF`, `CF`, `BF`, `AF`, and `AN` are different angles. For the three different images, the number of each classes have different distributions.

	% of Cloud	% of Ice	% of Unlabeled
image1	17.77	43.78	38.45
image2	34.11	37.25	28.63
image3	18.43	29.29	52.26

Table 1: Percentage of Classifications

We can see from the Figure 1 that the data points are clustered together, i.e. there are large crowds of clouds or ice. Therefore the data is not i.i.d distributed.

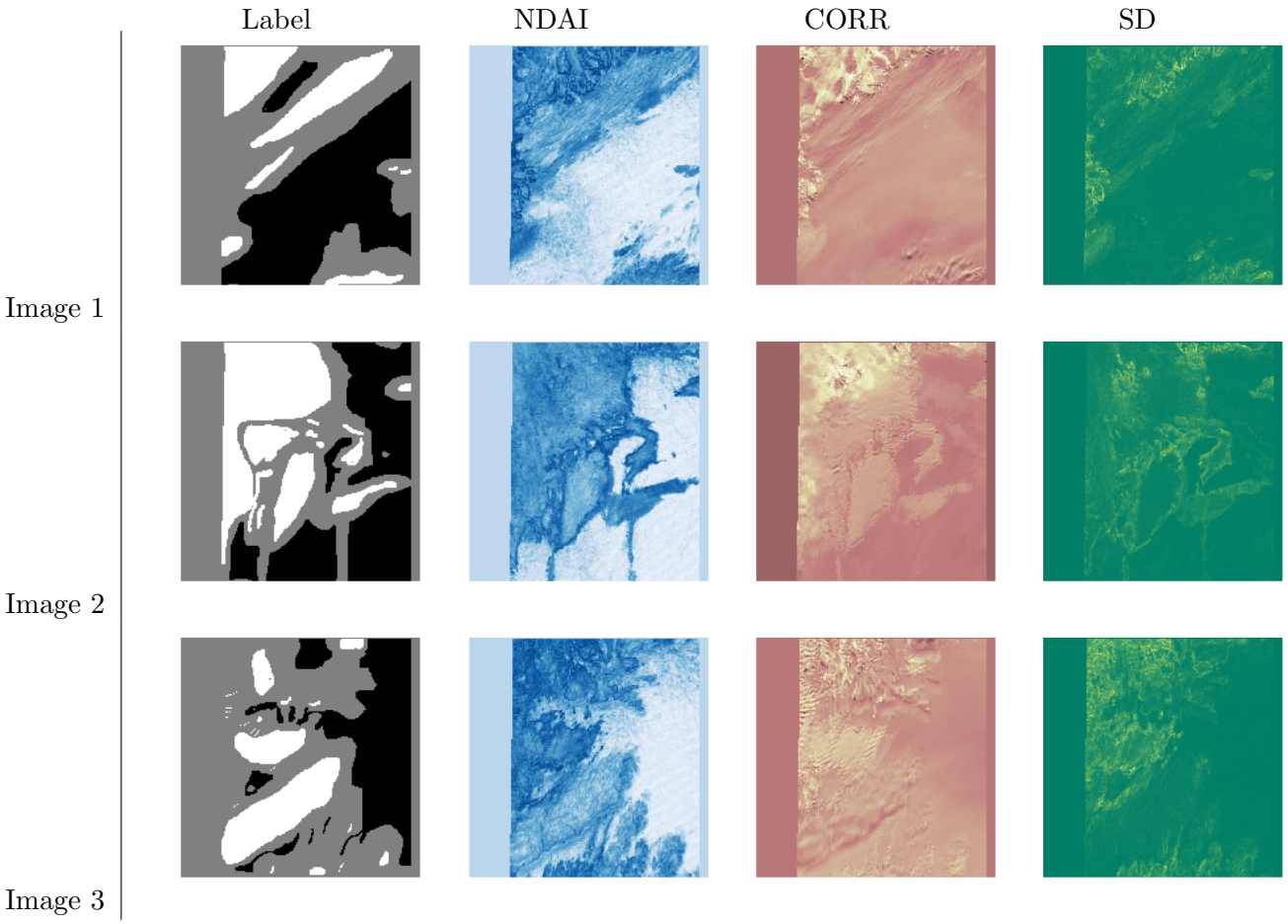

Figure 1: Visualization. From top to down are three figures in the data set. From left to right are the visualization of Label, NDAI, CORR and SD respectively. In the Label picture, the black areas are ice, the grey areas are unlabeled adn the white areas are cloud. We can see very clearly the visual correlation of each variables.

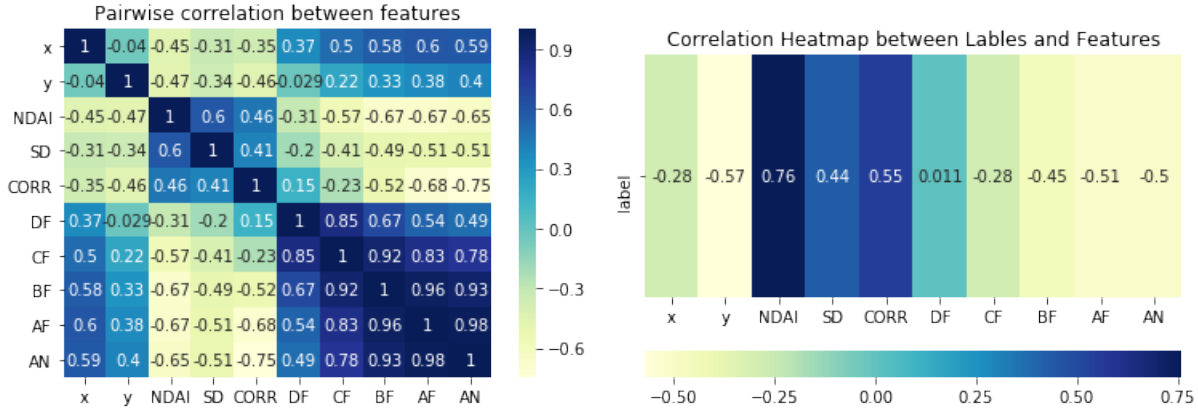

Figure 2: Pairplot. The plot on the left-hand side is the pairwise correlation between features. The plot on the right-hand side is the correlation heatmap between lables and features. The third plot is the boxplot of NDAI with respect to cloudiness

1.3 Visual and Quantitative EDA

The first correlation heatmap between features shows that features are moderately/highly correlated to each other. Especially for the features taken from MISR cameras (DF, CF, BF, AF, AN), their correlations are as high as 0.9. The second correlation heatmap illustrates the correlation between features and expert labels. NDAI has the highest correlation with the expert label among all features, which is 0.76.

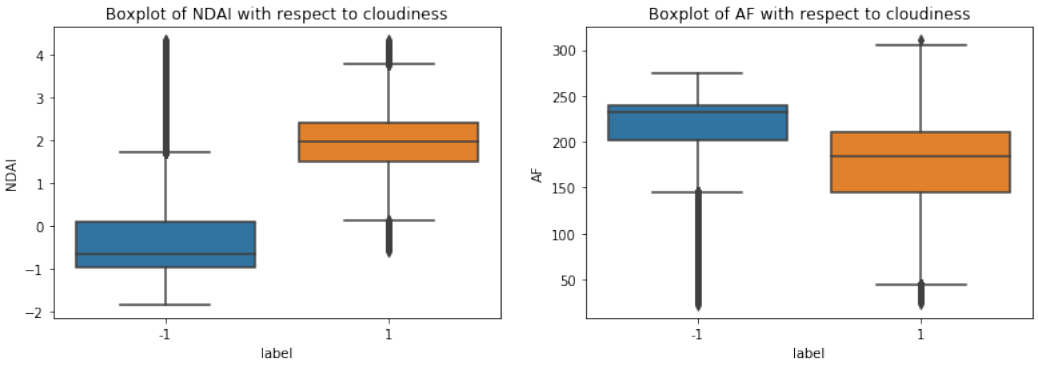

Figure 3: Boxplot. The left figure is the boxplot for `NDAI` with different labels and the right figure is the boxplot for `AF`

First of all, we removed all unlabelled pixels (where label = 0) in the images. Among all features, we chose NDAI and AF, which has the highest correlation among either the three physical features used in the paper or MISR features, to draw box plots and see their distributions grouped by cloudiness. From the graph above, we notice that cloudy pixels tend to have higher NDAI and lower AF compared to cloud-free pixel.

2 Preparation

2.1 Data Split

1. Method 1: The first method is to do proportional sampling on each image has the same representation in each set. To do this, we randomly shuffle the pixels in each image and split each image into three sets, each for train, valid and test. Finally we combine the three images together.

2. Method 2: Considering the spatial dependence of the image pixels, we decided not to randomly shuffle the pixels. Instead, in order the preserve the spatial pattern, we split the data based on the sequential order for each image: first 60% of the data (rows in the data set) in each image constitutes the training set, the following 20% goes to validation set and the last 20% will be used for testing.

2.2 Baseline Trivial Classifier

To calculate the trivial accuracy, we only need to calculate the proportion of cloud-free pixels in validation and test sets .

Split Method	Val.	Test
Spatial Split	0.8521	0.9430
Balanced Split	0.6107	0.6107

Table 2: Trivial Estimator Accuracy

As the spatial split only preserves the regional structure but ignores the global pattern, the classification accuracy based on the trivial classifier vary significantly from validation set to test set. As the balance split

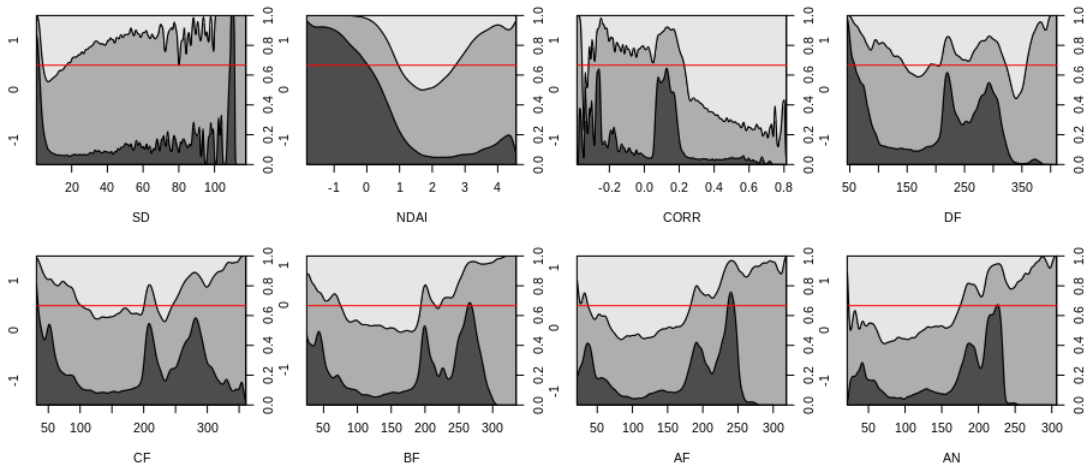

Figure 4: Conditional Density Plot of all variables. The white area indicates the probability of being classified into one. The red line indicates random guess 0.33.

is designed to handle possible class imbalance across data sets, its validation accuracy and test accuracy are identical.

2.3 First Order Importance

Our main goal is to come up with a list of features that help to create the most stable and powerful classifier of label. We introduce the notation $x^{(i)}$ as the ith feature from design matrix. To be more specific on the constraint, all features must satisfy the following criteria, $\forall i \in [1, p]$,

1. We want to maximize the overall possibility of determining the cloud's existence given this feature.

$$x^{(i)} := \arg\max_i \int \mathbb{P}(y_i|x^{(i)})dx^{(i)}$$

2. We also want to have stable or smooth possibility so that our prediction would be more robust. That is $\forall i \in [1, p]$ and some constant L.

$$|\mathbb{P}(y_i|x_1^{(i)}) - \mathbb{P}(y_i|x_2^{(i)})| < L|x_1^{(i)} - x_2^{(i)}|$$

We observe the following patterns on all our data points. In Figure (4), we see all 8 variables' conditional probability plots. We can simply observe that to satisfy Criteria (a) and (b), this lead to the three best predictors `CORR`, `NDAI` and `SD`.

3 Modeling

3.1 Model and Accuracy

In this section we use several methods by calling `CVGeneric` to fit the data. We will show the results and computing time of four models we have used. We also tried SVM as the beginning. As in this case, we have much more data points than the number of features, it is not computationally efficient to use a kernel method and in practice it takes us hours to train the model. Therefore, we decided not to pursue SVM anymore. In Table 3, we listed all of the models we have used so far and their results based on two different splitting methods. The average CV accuracy vary among models and split methods. Comparing the two splitting methods, spatial splits have higher accuracy in average but also higher variance across folds. The high variance is expected because each training fold has very different composition. Random forest based

on balanced splits gives out the highest accuracy of all types of models and splitting methods. For balanced split, its performance is more stable across folds. Comparing five algorithms, Random Forest is the winner in terms of accuracy for both splitting methods.

	Log-reg		LDA		QDA		Rand-forest*	
fold	bal.	spa.	bal.	spa.	bal.	spa.	bal.	spa.
1	0.8862	0.9098	0.8899	0.9245	0.8820	0.8147	0.9570	0.9413
2	0.8842	0.9753	0.8906	0.9220	0.8836	0.8961	0.9588	0.9503
3	0.8820	0.8602	0.8904	0.8603	0.8851	0.8716	0.9602	0.9169
4	0.8877	0.8372	0.8898	0.8311	0.8824	0.8530	0.9595	0.8883
5	0.8852	0.8543	0.8900	0.8550	0.8842	0.8552	0.9605	0.8931
6	0.8824	0.8899	0.8891	0.8927	0.8863	0.8782	0.9585	0.9318
7	0.8854	0.9193	0.8874	0.9225	0.8845	0.8903	0.9597	0.9498
8	0.8885	0.9343	0.8879	0.9356	0.8827	0.9214	0.9573	0.9522
9	0.8876	0.9568	0.8881	0.9582	0.8841	0.9414	0.9588	0.9729
10	0.8825	0.9482	0.8843	0.9534	0.8859	0.9503	0.9603	0.9614
avg.	0.8703	0.9025	0.8722	0.8732	0.8732	0.8359	0.9534	0.9358
time	8.1s		2.6s		1.0s		283s	
test**	0.8911	0.9220	0.8990	0.9271	0.9063	0.9120	0.9450	0.9300

Table 3: Results from different methods. The two cross validation split methods are used in all methods, where we denote the balanced split as **bal.** and the spatial split as **spa.**. *With 100 trees and max depth 20. Hyperparameter tuning will be explained in detail in Section 4. **To make sure that each split methods are comparable, test data is the combined test set of split 1 and split 2.

3.2 ROC Curves

Our choice of the ROC cutoff point is based on the maximum value of the difference of the true positive ratio and false positive ratio. In other words, we choose the cutoff threshold α such that $\alpha = \arg\max_\alpha [P(\hat{y} = 1|y = 1) - P(\hat{y} = 1|y = 0)]$. The intuition is that we want to maximize the true positive ratio and minimize the false positive ratio. From the ROC curves in Figure 5, Random Forest based on balanced splits and QDA based on spatial splits are the closet to the upper left corner and therefore have the highest AUC score.

3.3 Other Metrics

1. F1-score. The F1-score is defined as the harmonic average of precision and recall, which leads to a range of [0,1]. It characterized the accuracy of a test with the best F1-score as 1 and the worst as 0. The motivation of using F1 score is that we have a lot more actual negative classes in our entire data set and we would like to take into account of false negatives in our analysis which is ignored in ROC curve. However, the drawback of F1-score is that it does not take true negatives into account, so even if the model did poorly on predicting the negative, the F1-score could still have a chance to be good. From figure 6, the algorithm with the best F1 score is random forest. Random forest based on balanced split has a slightly higher F1 score than the one based on spatial split.

$$\text{F1} = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}; \quad \text{log-loss} = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

2. Logistic Loss. Using logistic loss, we find that the curve of logistic loss across folds is flat and smooth for balanced splits. However,for spatial splits, the losses of all four methods are generally higher and have more bumps in their curves. Random Forest is found out to have the smallest logistic loss in every scenario.

The results for both metrics are plotted in Figure 6.

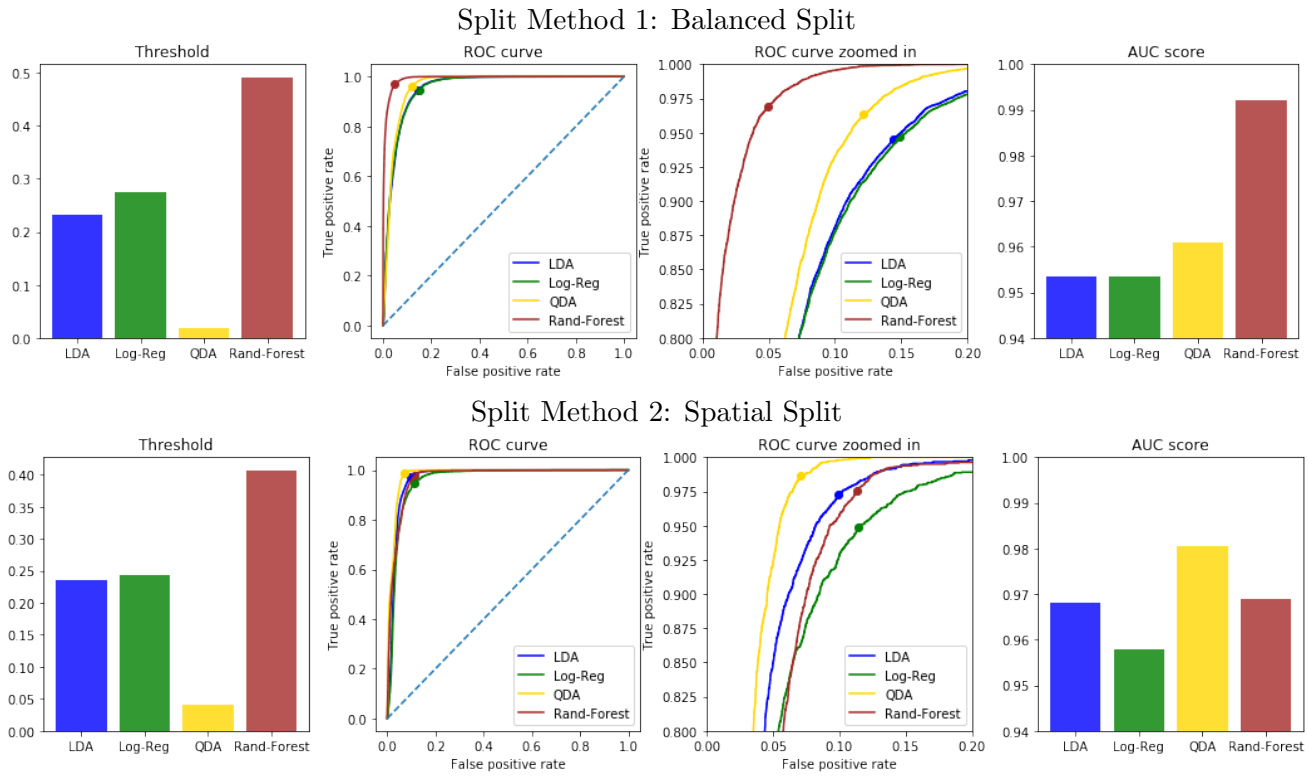

Figure 5: ROC Curve and AUC score. The first left hand figure is the threshold applying our criteria. The second figure is the ROC curve. The third figure is the ROC curve zoomed into the left corner. The fourth figure is the corresponding AUC score.

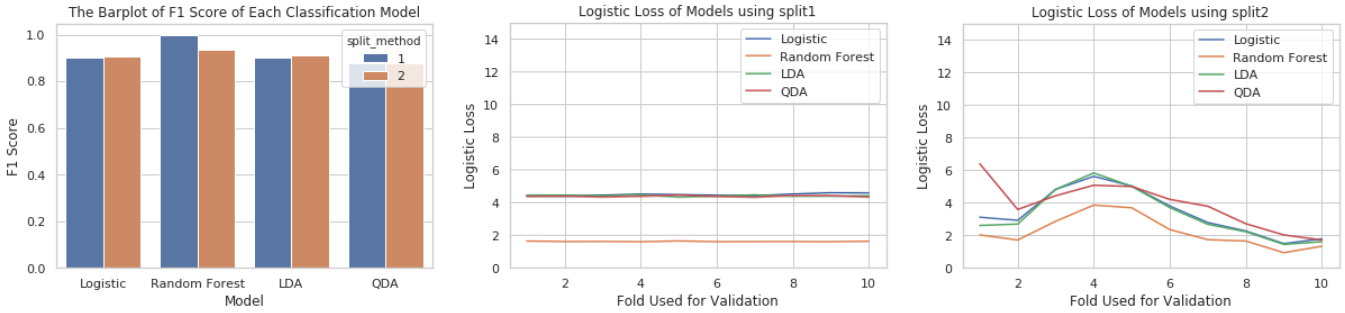

Figure 6: Plots of other metrics. The left-hand side bar chart is the F1-score of four methods. The two right-hand side plots are logistic loss across folds. The middle one uses balanced split and the right hand one uses spatial split

3.4 Summary of Model Selection

Among all models selected above, we recommend using random forest. The pro side of random forest is that it has persistently high classification accuracy across the folds if we split the data randomly based on their proportions (using split method 1). Even if using split method 2, which we split each images into small chunks and may cause a covariate shift from training data to validation data, the random forest performs the best in terms of accuracy. It also has the highest AUC score, highest F1-score and lowest logsitic loss among all four models. In addition, it can deal with multiple correlated feature and it is more interpretable as the thresholds used to split the nodes are the actual values. The down side of random forest is it takes much longer time to train and evaluate and it also has more hyper-parameters compared to other models. As we don't need to get a real-time result every time and train the model frequently, it takes reasonable

time to fit the model and get the result. Hyper-parameter tuning will be done in section 4.

4 Diagnostics

In the previous section, we chose Random Forest to be our recommended model in terms of its accuracy, interpretability and robustness. For this section, we do an in-depth research into random forest algorithm.

4.1 Hyperparameter Tuning and Convergence

In this subsection, we will discuss how we select our hyperparameters for our random forest classifier, including the number of estimators, maximum depth of each tree and minimum samples split in our random forest classifier.

4.1.1 Hyperparameter Tuning

We first use cross validation and grid search to choose the best hyper-parameter pairs of maximum depth and number of trees in random forest. The hyper parameter sets that we are interested in are number of trees: [50, 100, 200], maximum depth: [10, 20, 30] and minimum samples split: [300, 600, 900]. The minimum samples split decides the criteria for which we need to split the leaf node again. We choose the best parameter pairs by fitting both CV split methods to the training set and compare the best average CV validation accuracy. The results are shown in Table 4.

Method	Min Samples Split	Tree Number	Max Depth	Val. Acc.*	Val. All.**
Balanced Method	300	200	30	0.9424	0.9526
Spatial Method	300	50	10	0.8980	0.9103

Table 4: Hyperparameter Tuning Result. *Validation set of specific method. **The validation set is the combination of split method 1 and split method 2 with duplication dropped, so the expanded validation set might cause the accuracy to be smaller than that in Table 3.

4.1.2 Convergence

In this section, we will discuss the convergence of random forest estimator. Figure 7 shows the learning curve for two split methods. With split method 2, the CV training score is very low at the beginning but gradually improves as training sample sizes increases. However, there is still a large gap between the training and cross-validation curve at the end. The balanced split works better for this scenario, as its training score is very close to the maximum at the end.

4.2 Misclassification Error(4b)

To see the pattern of those misclassified points, we fit our model on the entire training set, predict the labels in validation set, and take out the misclassified data points to see if there exits any pattern. We applied the model with the best hyperparameters in Table 4. The accuracy could also be found in Table 4

We believe that the abnormal high accuracy from spatial split is due to the splitting method itself. Recall that in table 2, if we set all pixels to clouds, we could have already 80% accuracy if tested on validation set. With the random forest estimator, it's easy to achieve more than 80% or even higher accuracy. Therefore, in this section, we turn our attention to a less perfect but more robust balanced split method.

The results are plotted in Figure 10. We see from the figure that there's a major difference in the histogram with feature `NDAI`. The wrongly classified points have mean 2 but those correctly classified points have mean around 0, as shown with the orange line. This result is intuitive if we look at the histogram of NDAI, where a bimodal distribution could be clearly identified. The bimodal pattern is expected because

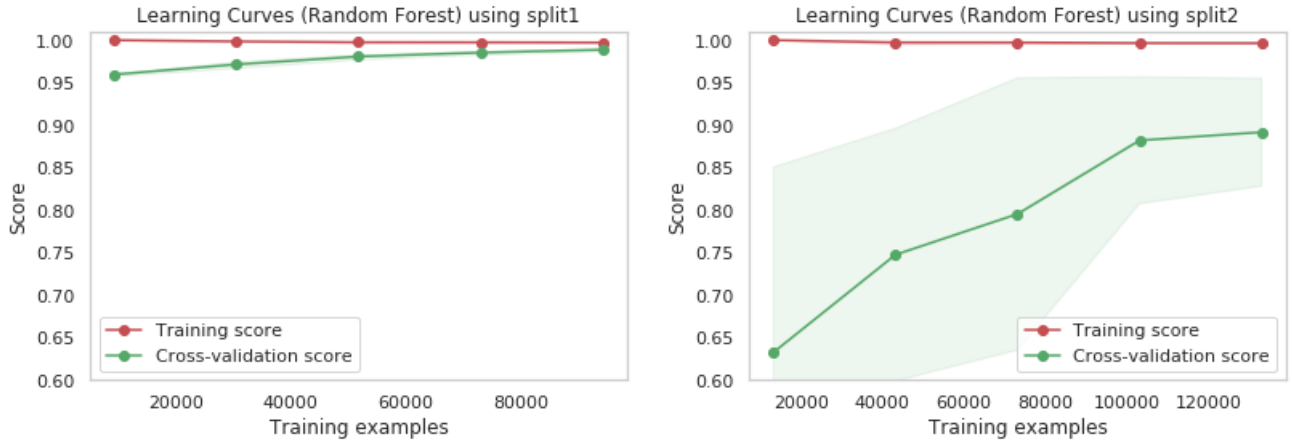

Figure 7: Learning Curves for two split methods. The left figure is for balanced split and the right figure is for spatial split. The two figures are plotted to match the same scale.

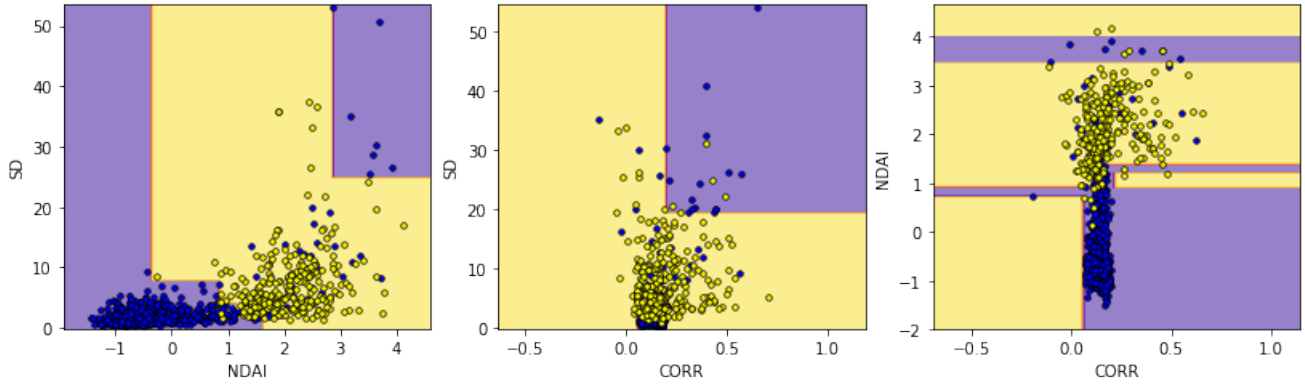

Figure 8: Random Forest split with max depth pruned to 4. The three figures are plotted against different pairs of `NDAI`, `SD`, `CORR`. The blue dots are ice pixels and yellow dots are cloud pixels in the dataset.

NDAI has the highest correlation with the expert label and we expect it to be powerful in distinguish cloudy pixels from cloud-free pixels. Noticeably, the distribution of `x` and `y` are different. This arouses our interest to reconstruct the images with all incorrectly classified data points, which would allow us to see if there exits any spatial pattern.

In Figure 9, We plotted the incorrectly classified pixels on the background image with combined validation sets for both methods, which is generated by merging the validation set of each splitting method. Both methods did equally well in classifying pixels in the first image, while both of them did worse image 2 and 3, particularly at the right lower corner, where we can see a cluster of red dots. Observed from the histograms in the previous section, NDAIs of correctly and incorrectly classified data points are drawn from two different distributions. For other features, the shapes of their distributions for the correct and incorrect classes are very similar. Therefore, one major reasons of the incorrect classification is the uneven distribution of NDAIs across data sets and limited exposure to some certain values which woud lead to poor understanding.

4.3 Review of split method (For Problem 4(d))

From the results we have seen so far, the spatial split method does not have a better performance than balanced split method. The reason behind this observation may lie in different aspects,

1. The feature `CORR` already contains spatial information. In Shi et al.[2018], the authors explicitly stated that `CORR` are derived from an average of neighboring features. If the pixels already have spatial

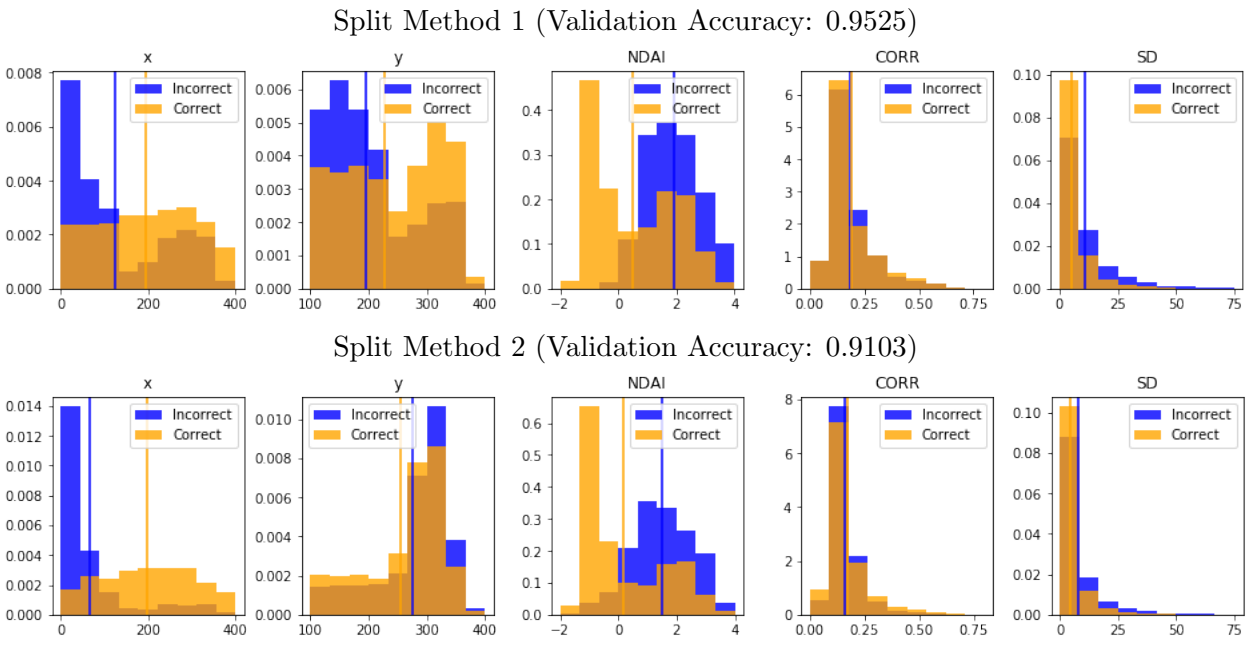

Figure 9: Class Error. Misclassified points are selected from split method 1, and we plot histogram of the points that are wrongly predicted in blue and the correctly predicted points in orange. The corresponding vertical lines are the mean of the two types.

information, we could just view it as i.i.d. points. Therefore, a spatial split might not be as necessary as it seems in terms of solving the non-i.i.d problem in this setting. The only tricky part is `NDAI` and `SD`, which do not have any information about its surroundings. We would add new features to our data set such that they will contain spatial information and satisfy i.i.d. assumptions.

2. The highly imbalanced data set is another threat to our model performance. In Table 1, we have shown that the number of cloud-free pixels are almost as twice as the number of cloudy pixels. Our classifiers would therefore be more likely to classify a pixel as a cloud-free pixel without penalty. Also, if we randomly split the pixels, it would be possible that some chunks have a larger portion of a specific label, which might lower classification accuracy at the end.

3. Spatial splits rely on large sample sizes to make each split samples i.i.d, while we only have three images. For example, the bottom left corners of each three image are all white or non-identified. If we train on such a subset, our model would wrongly infer that there might be no ice at all. If we test on such a subset, our model would not perform well if it has not seen anything like this area before. Therefore, the idea of spatial split requires a large sample size and the underlying spatial pattern should be repeated almost everywhere.

5 Bancroft Model: a way to capture neighbor information (4(c))

Considering the deficiency of current split method, we propose an new approach to capture the spatial information and to solve the imbalanced class problem at the same time. We namet this new appraoch "Bancroft", the street (born in MLK) where its creator wrote the first line of codes for this proejct. The basic idea of this model is that, for every pixel, we smooth out the surrounding pixels' features by averaging their values and generate new features based on that. The features we consider to do average pooling are `NDAI`, `SD` and `CORR`, the new features are therefore named as `Neighbor NDAI`, `Neighbor SD`, `Neighbor CORR`. From Shi et al. [2018], `CORR` is a metric that already contains spatial information, while `NDAI` and `SD` only contains information of a single pixel. Therefore, smoothing out the three features of one pixel is to obtain

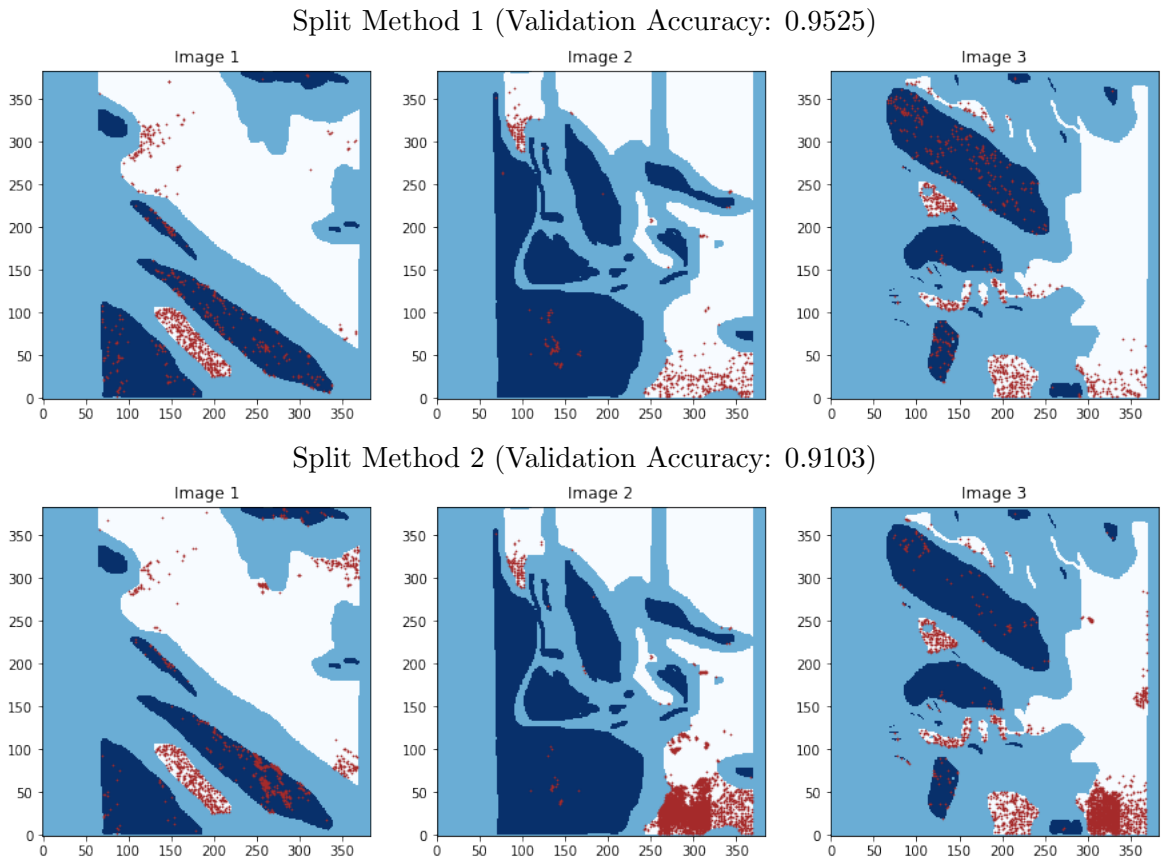

Figure 10: Class Error. Misclassified points are selected from split method 1, and we plot histograms of the points that are wrongly classified on the top row and the correctly predicted points on the second row. The orange vertical line is the mean of this feature.

extra spatial information for `NDAI` and `SD`, and a wider range of `CORR` than in the paper.

After applying the "Bancroft transformation" to our data, we fit a random forest classifier to all 11 features and see if the newly added features play important roles in predicting the label. In Figure 11, we show the feature importance, where three features having the highest importance scores are `Neighbor NDAI`, `AF`, `Neighbor SD`. The first and third ones are the newly introduced features and they even perform better than the features used in the paper.

We also predict the labels in validation and test set after applying the "Bancroft transformation". Shown in table 5, the validation results are better than what we have obtained in Table 3. For both splitting methods, they perform better on total valid sets, which makes sense, since some of the data points are used in the training phase. The reason to compare total validation set is that it is fairer to compare two methods if they are competing the accuracy on the same validation set.

The output pixels are shown in Figure 11. The clusters of red crosses are more sparse than that in Figure 10. This shows that our model have improved the prediction accuracy visually.

6 Summary(4e)

Given with three images taken by MISR camera, our goal is to build a classifier which can distinguish cloud-free pixels from cloudy pixels. We used two methods to split the data set. One method is based on proportional sampling while the other one is to split each image into a number of sub-regions and randomly select a sub-region. We used four classification models, including Logistic Regression, Random Forest, LDA and QDA to fit our data. Regarding to evaluation metrics, we used ROC curve, F1 score and logistic loss

Method	Valid*	Test*	Valid All**	Test all**
Split Method 1	0.9522	0.9516	0.9622	0.9633
Split Method 2	0.9122	0.9091	0.9242	0.9217

Table 5: Results using new Feature. *denotes the corresponding validation and test set. **denotes that these sets are combination of the two split methods. Since the two split methods choose the same number of validation and test samples, the result after dropping duplicate points is comparable.

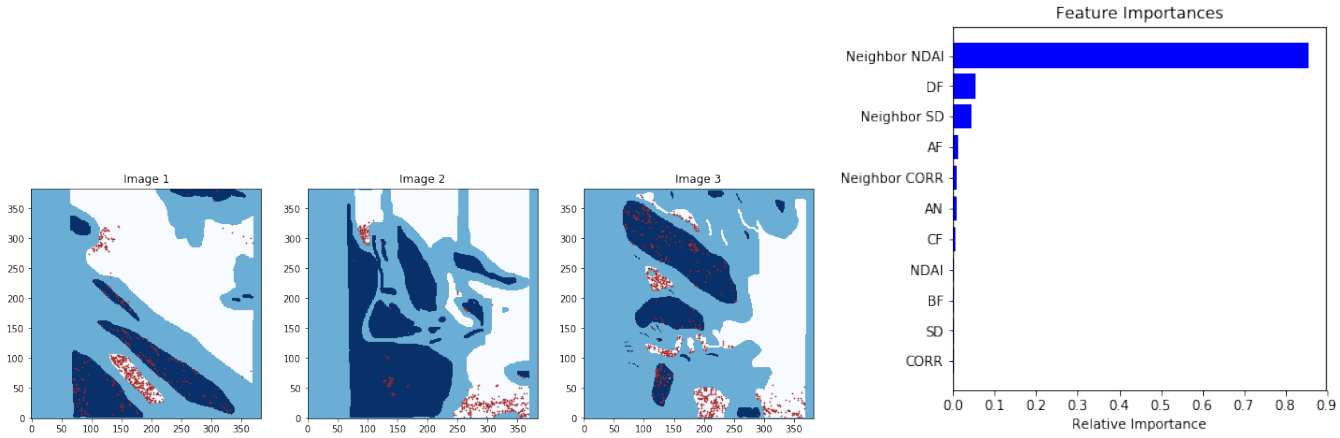

Figure 11: Left-hand figure: Pixel visualization with Bancroft Model. Right-hand: Feature Importance with Bancroft Model.

to measure our model performance. We recommended using random forest because it has the highest score in every metric we used and it is able to learn the features well even when features are highly correlated. Then, we tuned the hyper-parameters of random forest classifier using cross-validation. After thoroughly reviewing the performances based on two splitting methods, we built a new classifier that aims to combine these two methods. This method allows us to add spatial informatino to each pixel so that we can treat each pixel i.i.d. For each pixel, we did average pooling for NDAI, SD and CORR of all neighboring pixels (at most eight neighboring pixels) and added three extra features containing these values. The test accuracy of splitting method one is 96.33% and that of splitting method two is 92.17%. Compared to the models without "Bancroft Transformation", we did see an improvement in accuracy after applying in terms of test accuracy. Our code is available at https://github.com/chenyangzhu/stat154-project2.

7 Acknowledgement

Chenyang was responsible for writing data exploration, first order importance, ROC curves, misclassification, most data visualization work, hyperparameter tuning, splitting method one (balanced splits) and its following training and testing. He also contributed to the GitHub Readme and prepared the LaTex reports. Ling was responsible for writing paper summary, data exploration, F1-score, logistic loss, splitting method two (spatial split) and its following analysis and learning curves and rest of the report-writing.

We started the project very early but it turned out that it took much longer than we expected. As we didn't do the data cleaning very cautiously at the beginning, we had to reran most of our codes once we found our that the data we read was deprecated. During the process of reloading our data, we also found some other bugs that need to be fixed. It took us long to find out these errors as they were not obvious in the results. From this project, we learned the importance of data cleaning and preprocessing and learned to be more cautious when making any judgement