Shiyang Ni and Stella Park

Link to github repository: https://github.com/shiyangni/STAT154SP2019/tree/master/Project%20Folder

1. Data Collection and Exploration

1. a. Study Summary

Since clouds play an important role in modulating the sensitivity of the Arctic to increasing surface air temperatures, it is important to characterize them. The goal of this research is to build cloud detection algorithms, combining clustering and classification methods.

The data is collected from MISR, which has nine cameras and views the Earth at a different angle in four spectral bands. 233 geographically distinct MISR paths are collected on a repeat cycle of 16 days. Each complete trip of MISR is an orbit, and each MISR pixel covers 275m x 275m region on the ground. For this study, only the red radiances and al channels are full resolution, whereas the others are aggregated to a 1.1km x 1.1km resolution.

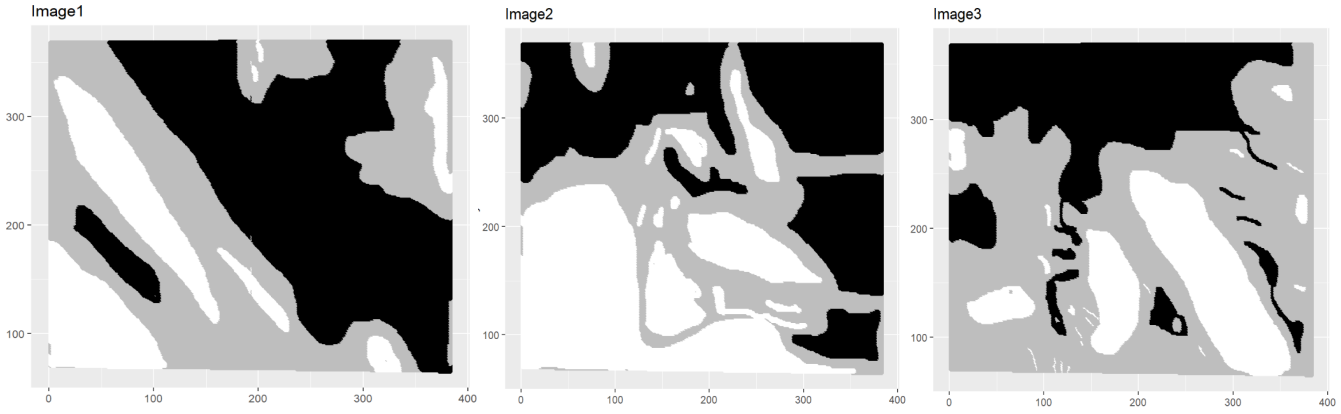
More specifically, the data were collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and Baffin Bay. The time span includes 144 days in 2002. There are six data units from each orbit, with 3 of 60 units excluded, so in total 57 data units with approximately 7.1 million 1.1-km resolution pixels with 36 radiation measurements for each pixel. Among 57 units, 5 million pixels are labeled by the expert and offline SVM is trained on 20k random expert labels from 20 data units and tested on remaining 37.

Three main features were used to assess the ELCM algorithm developed in this study: CORR, SD, and NDAI. It was concluded that ELCM algorithm is more accurate and provides better spatial coverage than existing MISR algorithms for cloud detection in the Arctic. The ELCM algorithm agreement rate was about 10% higher than the existing algorithms, and had 100% coverage rate, which is significantly higher than the others.

This study is significant because it can be applied to other scientific problems such as hurricane prediction and climate change. Also, it depicts the power of statistical thinking and the ability of statistics to contribute to modern, pressing scientific problems including environmental concerns.

1. b. Data Basics

Each image is of size 382×305382×305 pixels, and each pixel represents a 275×275m275×275m window of earth surface. For each pixel, our dataset documents its x,y coordinates, an expert label indicating whether the pixel is clouded, five angular radiance readings from MISR satellite, and three custom-created features capturing inter-pixel or inter-angular-channel dependencies.



A quick recreation of the three images is shown above, with black representing pixels not clouded, white clouded and grey not confident enough to judge. As we can see, all three pictures have a rather significant portion of pixels that experts cannot confidently label: image1 has 38% grey, image2 28% and image3 52%. These undetermined pixels contain no useful information about cloud existence, and therefore cannot be used in either training or testing.

Should we just throw them away? One might be concerned that a simple discard would break spatial/geological dependency, leading to worse parameter estimation. Our counter argument is that for basic Generalized Linear Models (logistics, LDA/QDA) that don't explicitly take into account the local interactions between variables, **sporadically** deleting some samples wouldn't degrade the model quality by much. Basic linear paradigms all assume some smooth global relationships between the class posteriors and the features; such global smoothness constraint implies the functions are unlikely to have irregular local behavior. So as long as our remaining samples can loosely cover the feature space, we can afford to delete some samples.

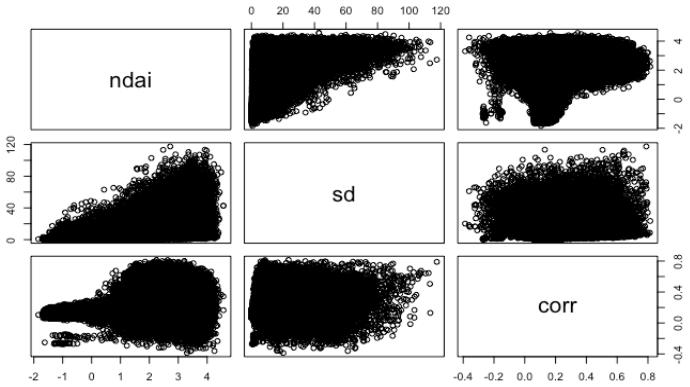
Of course, a better fix would be to go back to data collection stage, and augment these unlabeled pixels with more meaningful information, such as the confidence level with which the pixel is clouded. Under proper justification from prior knowledge, we might be able to use that confidence level as a surrogate for the class posterior, and produce a more informative model.

1. c. EDA

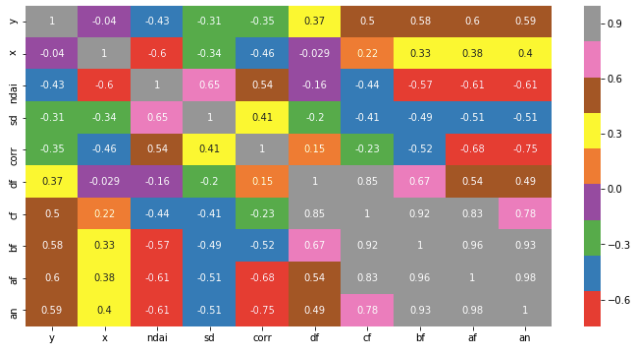
Summary of the merged data frame of the three images is shown on the left. Pair-wise scatterplot between custom-created features is displayed on the right.

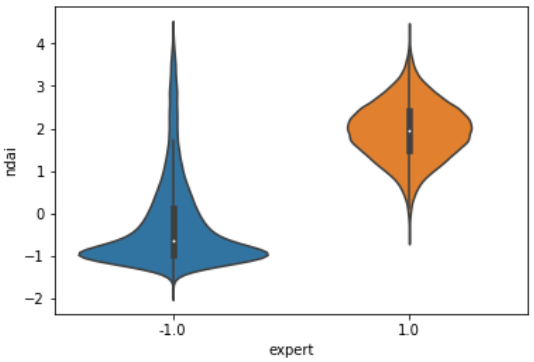
i.

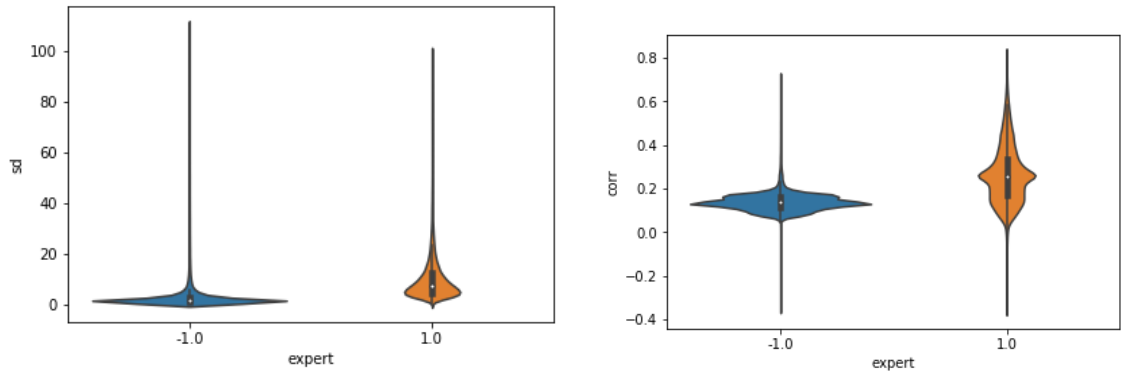
```
       y                x              expert             ndai              sd        
 Min.   :  2.0   Min.   : 65.0   Min.   :-1.0000   Min.   :-1.8420   Min.   :  0.1987  
 1st Qu.: 98.0   1st Qu.:143.0   1st Qu.:-1.0000   1st Qu.:-0.4286   1st Qu.:  1.6376  
 Median :193.0   Median :218.0   Median : 0.0000   Median : 1.3476   Median :  4.3095  
 Mean   :193.1   Mean   :218.1   Mean   :-0.1334   Mean   : 1.0847   Mean   :  8.0633  
 3rd Qu.:289.0   3rd Qu.:294.0   3rd Qu.: 0.0000   3rd Qu.: 2.3142   3rd Qu.: 10.2264  
 Max.   :383.0   Max.   :369.0   Max.   : 1.0000   Max.   : 4.5639   Max.   :117.5810  
      corr                df                cf                bf                af        
 Min.   :-0.3872   Min.   : 45.28   Min.   : 31.19   Min.   : 24.49   Min.   : 21.07  
 1st Qu.: 0.1253   1st Qu.:244.56   1st Qu.:219.27   1st Qu.:200.79   1st Qu.:185.16  
 Median : 0.1603   Median :281.91   Median :259.31   Median :236.17   Median :211.54  
 Mean   : 0.1860   Mean   :271.36   Mean   :246.37   Mean   :224.20   Mean   :201.71  
 3rd Qu.: 0.2231   3rd Qu.:300.39   3rd Qu.:279.59   3rd Qu.:258.62   3rd Qu.:235.15  
 Max.   : 0.8144   Max.   :410.53   Max.   :360.68   Max.   :335.08   Max.   :318.70  
       an        
 Min.   : 20.57  
 1st Qu.:174.88  
 Median :197.58  
 Mean   :188.29  
 3rd Qu.:216.80  
 Max.   :306.93  
```



ii.







From the plots above, it can be deduced that the mean and median of the distribution of the three important features for each expert labeled data are drastically different for ndai, and similar for sd and corr. We will explore more in part 2(c). There are more outliers in the cloud-free pixels.

2. Preparation

2. a. Data Splitting

- Test Set?

Both ways of splitting use the entire image3 as the test set. This is to best simulate real-life prediction scenarios, where the actual test picture always comes in individually. If we had created the test set using partial information from all three images, then we would have introduced some form of spatial correlation between training and test dataset that doesn't exist in a real prediction setting. For models that are able to pick up such correlation, our test set would yield a potentially higher accuracy rate than a real-life test set would.

The tradeoff is of course we are losing around 30% samples to training. Nevertheless, given that the sample size is already large, we decided to stress the structural integrity of the test set.

- Training vs Validation Set: Method One

Method: Divide each image into four quadrants, so we have 8 quadrants in total. Pick one as validation, and leave the other seven as training.

Rationale: Training set preserves space-based structure; Validation set covers unbalanced cases.

In analogy to time series data, randomly spitted training data would only partially preserve space-based structures, if such structures do exist. For models that pick up space-based dependency, this would hurt the quality of parameter estimation. By dividing each image quadrant-wise, we're attempting to best preserve any space-based structure.

Our validation set turns out to be unbalanced. But that doesn't disqualify the quadrant from being a proper validation set: an unbalanced/distorted validation set can identify the model that performs well on unbalanced data. In practice we should expect some test data to be highly imbalanced, as clouds can be very large.

- Training vs Validation Set: Method Two

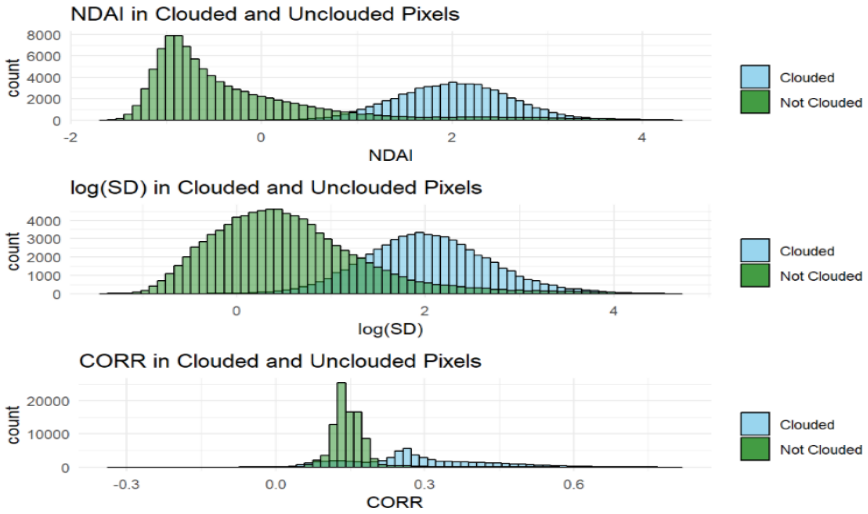
Method: We randomly sampled from each quadrant an equal amount (1/8 of total) of pixels, use them for validation, and the rest for training. In this manner, each quadrant is equally represented.

Rationale: Generalizing to potentially defected images.

2. b. Null Classifier

The validation accuracy will be 0.65 and the test accuracy will be 0.61 for a trivial classifier. Such classifier will have a high accuracy only in datasets that have the majority of the data points classified as being cloud-free (expert label=-1).

2. c. First Order Importance



If a feature performs very differently between clouded and not clouded pixels, then it likely has some predictive power. We look at summary statistics for a quick search of suspects. For each potentially influential feature, we histogram its respective distributions among clouded and unclouded pixels, and see if the two distributions are separate enough to give any predictive power.

The mean and median of `NDAI`, `SD` and `CORR` are all significantly higher among clouded pixels than non-clouded pixels. Histograms conditioned on cloud existence also look promisingly separate – for each of the three features, there exists some non-trivial feature range for which the class posterior $P(Y=1|X_j=x)$ (where X_j denotes the jth component the complete feature vector) is dominatingly high. Note that although the separateness of conditional distributions provide some intuitive justification for some thresholding scheme, it doesn't provide the threshold values. We need an in-depth analysis of the class posterior conditioned jointly on the three features to establish proper thresholds.

2. d. Cross Validation Function

Please see `report/report.Rmd` or `code/CVgeneric.R` for important limitations on input data types, fold-creation methods, and admissible models. Both copies are identical except that `code/CVgeneric.R` contains some manual testing.

3. Model Fitting

We won't touch the test set in this part. This is to ensure the model selection is completely based on using available data, which best mimics a real-life prediction situation.

3. a. Model Fitting

We're following 4 model-fitting schemes, where each of them is distinguished through different set of hyperparameter(s):

i. Penalized Logistics Regression with both L1 and L2 penalty. This is analogous to elastic net in Regression. Hyperparameters are weights of both penalty, and the overall strength of penalty represented by the sum of weights.

ii. RDA(Regularized Discriminant Analysis), a generalization of LDA/QDA where one adjusts the homogeneity of the shapes of conditional dispersion of features.

iii. LDA with feature selection. This is analogous to what Yuansi did for `AmesHousing`.

iv. Random Forest, which yields non-linear boundaries.

For each scheme, we will choose a basket of hyperparameter values, fit the model for each combination of hyperparameters, check if the assumptions are satisfied, and use cross validation (both K-fold and validation set) to pick the potentially optimal choice of hyperparameters.

i. Penalized Logistics Regression

Assumptions for Logistic Regression

- Target variable should be measured on a dichotomous scale
- There are one or more feature variables, which can be either continuous or categorical.
- The observations are independent and target variable has mutually exclusive and exhaustive categories. This means the model should have little or no multicollinearity.
- There is a linear relationship between any continuous feature variable and logit transformation of the target variable.

Assumptions in relation to our model

- The target variable (expert label) is binary, -1 indicating cloud-free and 1 indicating cloudy.
- There are 8 features which are all continuous variables.
- The data collected is not iid because of the spatial correlation between data points. Therefore, the sophisticated split method according to different x and y coordinates was used to make the observations more independent and avoid the model from having high multicollinearity.
- When the logit odds are calculated, they were approximately in linear relationship with the target, because the dataset is class-wise Gaussian in distribution, according to EDA.

Commentary

- The training size was reduced to 50,000 because training with all the training data would take 4.3 hours. Reducing the training size will increase bias because we are cutting out some of the actual data. Yet, since the test accuracy was high, sampling was not a problem.

- Hyperparameters were tuned with GridSearch and log loss was used as the loss function because it is the most commonly used one for Logistic Regression
- 5 fold CV score for split 1 = [0.8908, 0.9004, 0.8969, 0.8914, 0.8978]
- Test score for split 1 = 0.8920899610745351
- 5 fold CV score for split 2 = [0.9361, 0.9371, 0.9324, 0.9331, 0.9343]
- Test score for split 2 = 0.73128
- The accuracy scores for both splits were quite high. However, although the CV accuracies are similar for both split methods, the test accuracy is significantly lower for split method 2. This is expected because split 2 assumes that the dataset is distributed in iid format.

ii. Regularized Discriminant Analysis

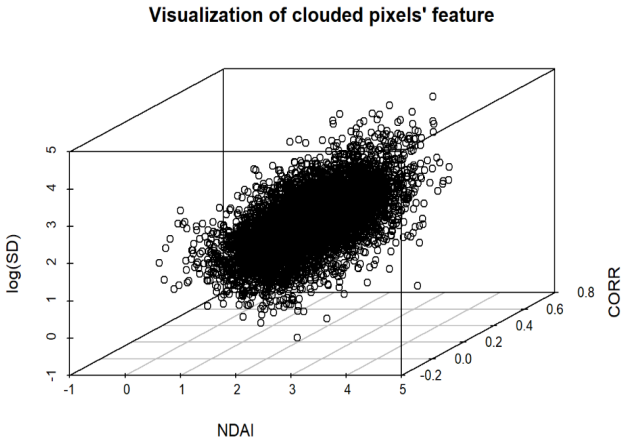
LDA model assumes each sample's features are generated through a multivariate mixed-Gaussian, and that their class-conditioned distributions share the same covariance but differ in mean. Geometrically, this means that for each class, any generated data cloud (in feature space, with suffciently large sample size) would be the shape of an ellipsoid, and that such ellipsoids for different classes only differ in mean but not in shape. QDA, on the other hand, allows each class's ellipsoid to have completely different shapes.

RDA falls in between the spectrum. It models each class's variance using a weighted average of global covariance and class-specific covariance: $\Sigma_{RDAk}=\alpha\Sigma+(1-\alpha)\Sigma_k$, and thus allows class-specific ellipsoids to possess flexible degree of homogeneity in shape. Hyperparameter ααrepresents the strength of shape-homogeneity.

We cross-validate over five α values: 0 (which yields the LDA model), 0.25, 0.5, 0.75 and 1(which yields the QDA model).

Feature selection: We only use `CORR`, `NDAI` and `log(SD)` (log to disperse the distribution of SD) for RDA fitting. Testing Gaussian-ness in high dimension is hard, but if we keep the dimension of feature to three, we might use some visualization technique to justify the Gaussian-ness.

Testing of Assumptions: We did a quick 3D sketch of feature space for clouded pixels (for the randomly spitted validation set), where ellipsoidal dispersion looks reasonable; we also sketched its projections onto pair-wise feature space (graph not included to save space), and the elliptic shapes once again show up. The same pattern is observed among unclouded pixels. We have reasons to believe the Gaussian-need of conditional distribution of features.



Professor Yu mentioned in class that even models that don't satisfy their own assumptions can prove to be useful predictive scheme. So the assumption testing isn't of paramount importance here. We focus on other model selection metrics.

Cross validation: We do two types of cross validation. First one is K-fold, the other one is single validation set. For K-fold, we report the average loss across folds, and also the time consumption. For

the validation set approach, we report the errors on two different validation sets, one sampled randomly and the other one not. As we explained in Q2 part a, we expect the non-random set to identify the method's strength over potentially unbalanced data, and the random set for general data.

```
##                  RDA Average Error RDA Time Elasped for five-fold CV(sec)
## RDA(alpha = 0)           0.06060171                               47.76108
## RDA(alpha = 0.25)        0.06471107                               43.90154
## RDA(alpha = 0.5)         0.06516186                               38.57982
## RDA(alpha = 0.75)        0.06543625                               38.81619
## RDA(alpha = 1)           0.06695848                               38.97473
```

```
##                  RDA Error Non-random RDA Error Random
## RDA(alpha = 0)              0.002833638       0.06135995
## RDA(alpha = 0.25)           0.002787934       0.06517535
## RDA(alpha = 0.5)            0.002833638       0.06548895
## RDA(alpha = 0.75)           0.002970750       0.06575027
## RDA(alpha = 1)              0.002970750       0.06757957
```

Computation-wise, all five models perform similarly. For both randomly sampled validation set and the K-fold, RDA with $\alpha=0$ (which is exactly the LDA model) wins in accuracy. RDA with $\alpha=0.25$ performs slightly better over unbalanced data. We keep these two candidates in mind.

iii. LDA with feature selection.

Does it make sense to only consider the three custom-created features? Would inclusion of geological information(x and y) help improve the model quality? What about the raw radiation readings?

In this subsection, we attempt to address these natural concerns by cross validating over four models: LDA that only includes custom features, one that has custom+geological features, another one with custom+radiation_reading, and the full model.

```
##                LDA Average Error LDA Time Elasped for five-fold CV(sec)
## custom               0.06058864                               41.72238
## custom+geo           0.05368308                               43.90154
## custom+reading       0.06478947                               38.57982
## full                 0.05118087                               38.81619
```

```
##                Error Non-Random Validation Error Random Validation
## custom                          0.002833638             0.06135995
## custom+geo                      0.003153565             0.05435635
## custom+reading                  0.006992687             0.06585481
## full                            0.005712980             0.05127267
```

Again, computationally all four models perform similarly. For validation error over generic data, custom+geo model performs best. The LDA that only includes the three custom features only performs best for validation over non-random validation. This means that geological information isn't useless!

Our winners here are custom+geo and custom.

iv. Random Forest

Assumptions for Random Forest

- It is assumed to smooth out the variance, so that the model is less biased.

- The randomness of trees "ensures" that the algorithm creates models that are not correlated with each other.
- Model assumes independent observations, dataset with no correlation
- There is no formal distributional assumptions, indicating that the model is non-parametric.

Assumptions in relation to our model

Random forest is an appropriate classification algorithm to train our dataset because we want less correlation in data as possible. Our split method endeavors to reduce spatial colinearity, which tries to ensure independence in the dataset. Since the dataset does not follow a particular distribution, it is nice that random forest is non-parametric. One disadvantage is that it is difficult to interpret the results, because it is difficult to understand what the binary splits indicate.

Commentary

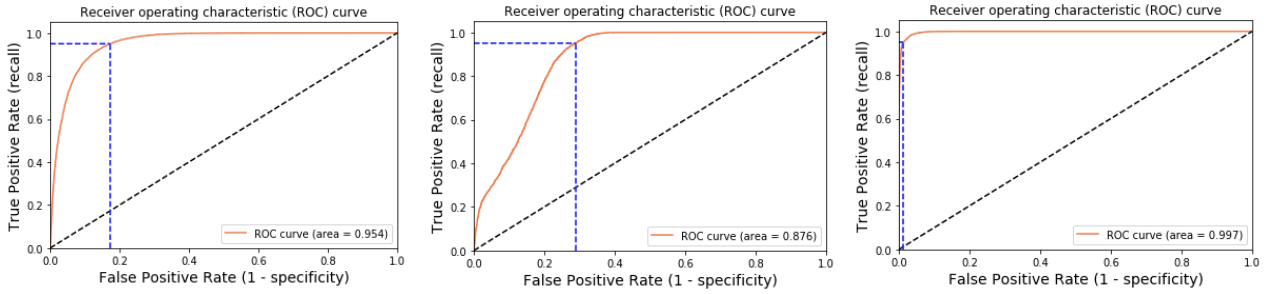
- The training size was reduced to 10,000 because training with all the training data would take 38 minutes. Reducing the training size will increase bias because we are cutting out some of the actual data. Yet, since the test accuracy was high, sampling was not a problem.
- Hyperparameters were tuned with GridSearch. The more number of decision trees in the parameter, the more accurate the CV scores became
- 5 fold CV score for split 1 = [0.9686, 0.9718, 0.9721, 0.9695, 0.9733]
- Test score for split 1 = 0.9693642174059302
- 5 fold CV score for split 2 = [0.8458, 0.8493, 0.8518, 0.8474, 0.8475]
- Test score for split 2 = 0.844946897976837
- The accuracy was the highest among all other models. This is expected since we are dealing with real world data, which is not represented in a nice distribution and random forest is non-parametric.

3. b. ROC curve

ROC curve tells us about a model's false positive and false negative rate. In part a) we picked the models with the best overall error rate (over different validation sets). In this part b) we will use the false positive and false negative rate as a thresholding mechnism – as long as a candidate model doesn't perform too poorly in terms of type I or type II error, we will regard the model as admissible.

We use the randomly-sampled validation set to calculate the false postive and negative rate for each method. We arbitrarily set both thresholds at 10% = 0.1, meaning if either false positive or false negative rate is higher than 0.1, we discard the model.

ROC curves (from left) Logistic Regression split 2, Logistic Regression split 1, Random Forest split 1



As we don't use ROC curve to pick the threshold, we only display them for the sake of completeness. ROC curves for random forest and logistics regression are shown below. The curves for LDA with tuned features and RDA are similar to that of logistics.

```
##                False Positive Rate False Negative Rate
## RDA_0                   0.06786478          0.06095596
## RDA_0.25                0.06897732          0.06538668
## LDA custom+geo          0.05400086          0.05061762
## Logistics               0.18634108          0.07971270
## RandomForest            0.01387407          0.03771734
```

The table below shows false positive and negative rate. The logistics model has a false positive rate of 0.18, which is higher than our 0.1 bar. We won't use logistics model for prediction purpose.

4. Model Diagnostics

We did the diagnostics in Part3 using validation set only. In Part4 we reintroduce the test set to see how models perform in real-life setting.

4. a. Further diagnostics on candidate models

In Part3 we find that Random Forest performs best in accuracy (~97%), but takes about 3min to train (for our validation/training split). LDA and RDA models have slightly lower accuracy (~95%), but their training takes only 40+sec. Penalized logistics is similar in overall accuracy to linear discriminant models, but its false-positive error is quite high.

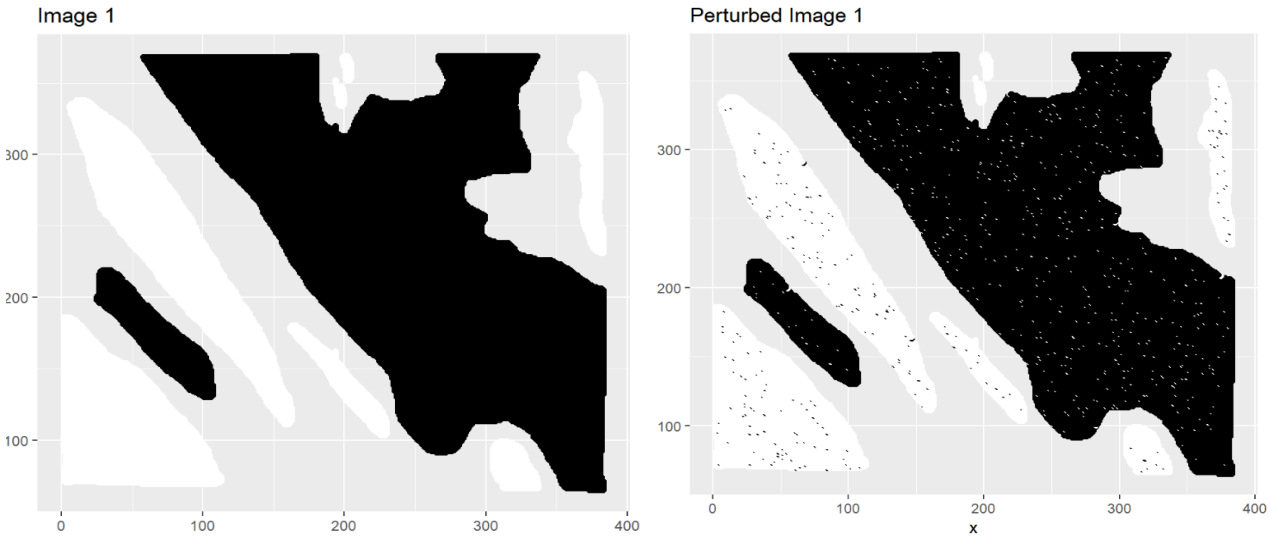
In this subsection, we use three metrics: test-time performance, interpretability of features, and stability under perturbed training data.

Test-time Accuracy: All three RDA models perform way worse on test sets than in validation set, each having error rate over 21%. Even though we arrived at the three linear discriminant models from different thought process(one from feature selection, two from regularization tuning), they are similar in nature, and in test-time performance. For simplicity's sake we will just use the LDA model with custom+geo features.

The Random Forest model performs relatively better, showing an test-time error rate around 16%. If the goal is accuracy alone, we would recommend Random Forest.

Interpretability & Feature Importance: ????

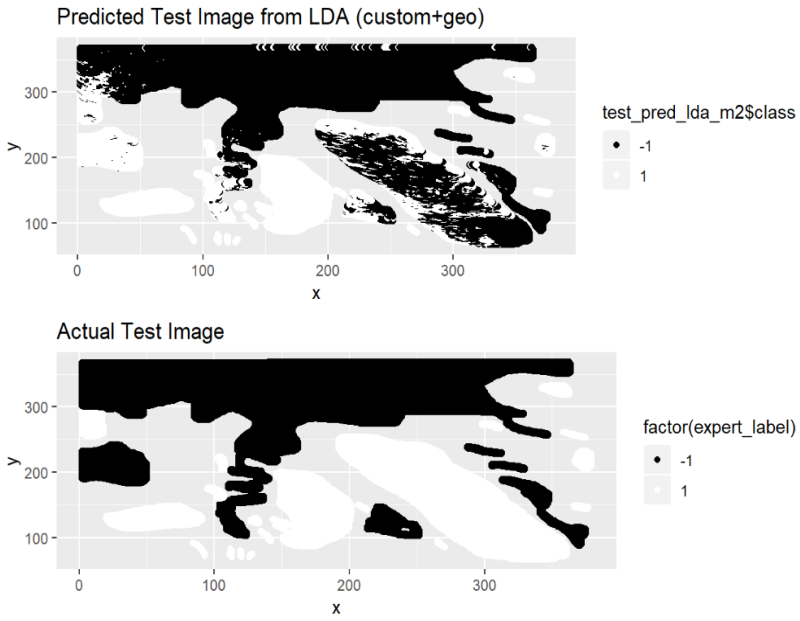
Stability under perturbed training data: We perturb a part of the training image by randomly sampling one percent of useful pixels from image1, and flip their labels. The effects are shown below. We then re-train the model using perturbed training set, and report its test accuracy.



It turns out for the LDA model with custom+geo features, the perturbed error is 0.24, a bit lower than the actual error 0.25. We repeat the experiment about ten times, and each time the perturbed error is around 0.24. We can thus conclude that the LDA model is robust under perturbance.

For Random Forest, the perturbed error falls within the range of 0.12-0.18. For stability purpose, we would recommend the LDA model using custom features + (x,y) coordinates.

4. b. Graphical distribution of misclassifications



A comparison between predicted image and the actual test image is shown below. For the LDA model, the error concentrates in two parts: one small patch near the left boundary is falsely positive, and a big patch in the fourth quadrant is false negative.

We think that the culprit for such concentrated errors is imbalance in training data. Looking back at image1 and image2, we realize in both images the left patch is mostly white, and the bottom right corners are mostly black. Such problem might be addressed through including more images in the training dataset.

4. c. Can we improve?

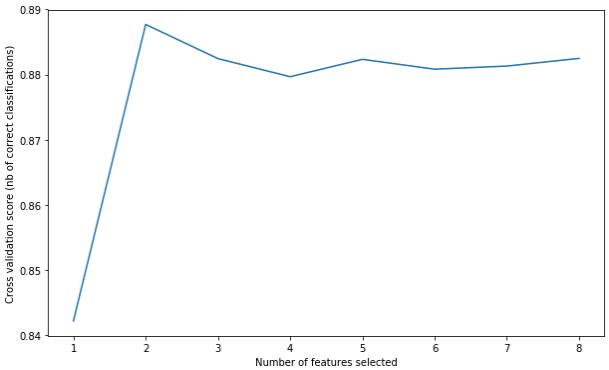
Since we tried so many classification methods and Random Forest has significantly good test accuracy rate of above 96%, choosing a different classifier is not an option for us. Instead, we decided to see the effect of feature engineering in improving the accuracy of the models, for both split methods.

We chose to improve the Logistic Regression model because for split method 2, it had a relatively low accuracy. After conducting Feature ranking with recursive feature elimination and cross-validated selection (RFECV) from Sklearn package, we found out that using 2 features, NDAI and Corr, led to the best CV accuracy results. Then, we compared the test accuracies for dataset only using the two best features and the other with all eight features. For split method 1, it did not have much impact. Yet, for split method 2, the test accuracy increased from 0.72392 to 0.7868. See the graph on the right.

4. d. Does data splitting matter?

Not by much, as long as we're using image3 as the test set. It makes sense because neither logistics nor RDA models in our implementation contains interaction terms, and thus ignore any space-based structures.

4. e. Conclusions



In conclusion, we discovered that using different models, we can predict if the pixel from the satellite images is a cloud, using different ML algorithms. It is interesting that the way we split the data impacts the accuracy of the prediction models. This indicates that the geographical correlation among data points keeps the dataset from resembling an i.i.d. random distribution. Thus, dividing the data into quadrants according to the x and y coordinates of the pixels improved the performance of the model predictions. After extensive data analysis and trying different classification models, we conclude that for accuracy, it is best to use Random Forest, for stability, use LDA, and for interpretability, use Logistics Regression. In order to further improve model performance, we would include more images in training to avoid concentrated errors.