Lab 4- Cloud Identification

Hoang Trong, Rahul Verma, Andre Waschka

November 7, 2014

1. Introduction

When it comes to understanding and predicting global climate change, identification of cloud cover is a vital portion of the equation. To do this, scientists create algorithms using satellite images to distingush between clouds and non-clouds. Since most of these models use the reflecting light from the sun as a primary distingusher, clouds are easy to identify because they reflect light at a much higher rate than land or ocean. However, an issue arises when trying to identify cloud cover in polar regions. This is due to the snow and ice reflecting sun in a similar way to clouds and thus making differentiation much more difficult. Our goal is to use the few images that we have that were labeled by an expert to train a model to be able to identify clouds vs non-clouds in polar regions.

2. Exploratory Data Analysis

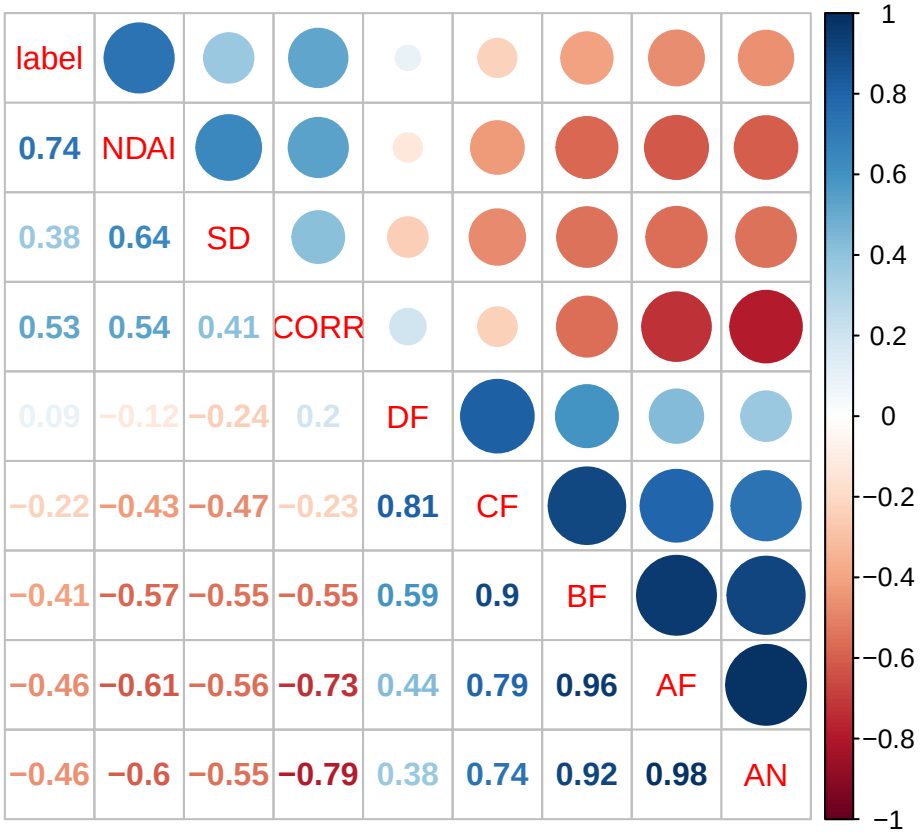


Figure 1: Heat and correlation plot of the training data combined to visually and quantitatively see the relationships between the variables.

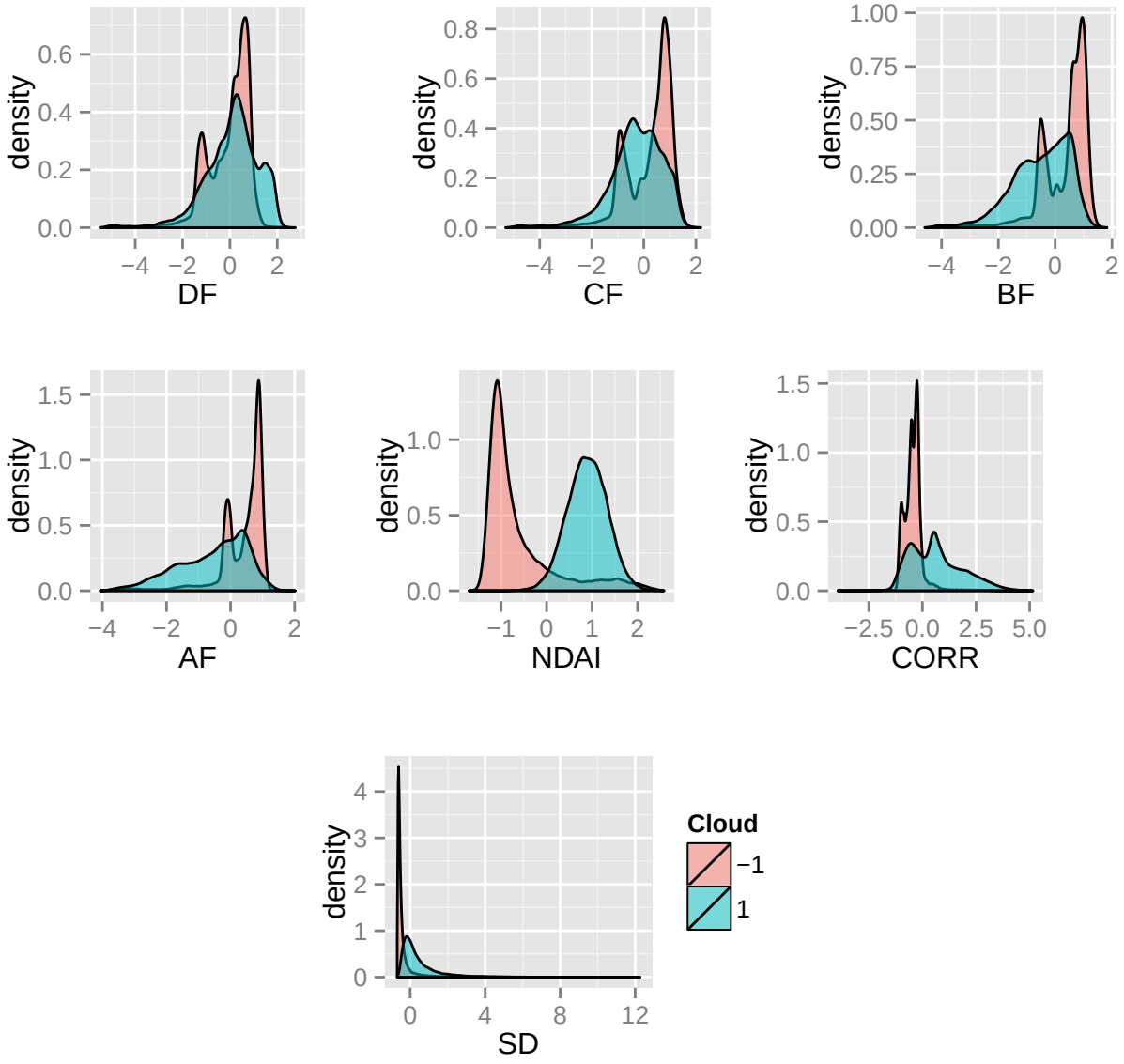


Figure 2: Class conditional density plots of the training data on the five camera angles and SD, CORR, and NDAI

After some preliminary exploratory data analysis, it quickly became clear that our first major decision would be to decide what to do with the portions of the image where the expert was uncertain whether it was a cloud or not. As a result, our figures and plots show both options. However, after deliberation, it was decided that the zero's(Unlabeled) data should be removed. This makes sense because a model should not work to predict unlabeled areas of the image. Instead it should give a prediction of cloud or not cloud and then give some measure of confidence.

Once this decision was made, we were able to look at the relationship between the radiances of different camera angles. After looking at the correlations between each of the five camera angles we can see that the radiances of cameras closest to each other are much more similar to each other than to ones with drastically different angles. When the focus is shifted to the label and the radiances we see that the range of radiances is much larger for clouds than for not cloud. Furthermore for all camera angles we notice a bi-modal distribution of radiances for not cloud. Finally we looked to see if there was a difference based on the features NDAI, SD, and CORR. From the correlation plot above we can see that all three features' correlations consistently

become more negative as the camera angles face more directly downward. However, CORR has the largest changes in correlation between camera angles, going from positive in DF, to the strongest negative correlation of all the features in AN.

3. Modeling

3.1 Feature Selection

To predict what are the three best features to predict the presence of clouds, we used several methods. The first and simplest was to look at the heat map and the conditional density plots. From these diagrams, we came to the conclusion that NDAI, CORR, and AF were the best at predicting the truth. NDAI was the clear favorite visually. It had the strongest blue color of all of the features in the heat map (Figure 1) and it had clear separation in the density plot in Figure 3. CORR seemed to have a similar correlation with SD based on the heat map so it was necessary to observe the density plots to make a final decision. From these plots it seemed that CORR had more seperated peaks than SD. Finally, AF was selected as the radiance angle. Looking at the heat map, AF and AN seemed to be the same color with a negative correlation so it was necessary to compare their density plots. Again these seemed very close. However it looked like AF had a slightly larger green(cloud) tail so we decided upon AF over AN.

The second method that was used to identify the three best features was to simply look at the correlation between label and the remaining eight features. These numbers can be seen in Figure 1. Selecting the three variables with the three largest correlations would result in NDAI, CORR, and AF being selected. Choosing NDAI and CORR was simple because they had the two largest correlations. However deciding between AF and AN was difficult. Since the correlations with label are the same, ideally we would choose the one that was least correlated with NDAI and CORR. In this situation NDAI is more correlated with AF than AN and CORR is more correlated with AN than AF. As a result we decided to choose AF due to its larger difference in correlation with CORR.

Finally, we try to use a measure that is more relevant to binary variable. In term of performance measure, accuracy is a simple yet effective measurement. It is symmetric with respect to the zero and one class, unlike some other measurements with emphasizes one class more than te other. In our case, the problem of detecting cloud and no cloud is symmetric, so accuracy is an appropriate measure. Accuracy simpy means the percentage of time a model classifies correctly. The only downside is that most models return a probability based prediction, and accuracy depends on the threshold at which one cut the probability to classify as positive and negative. One way to fix this dependence on threshold is to pick the threshold with the best possible accuracy.

The other way is to use area under curve (AUC) of the ROC curve. This measure is independent of threshold. The continuous variable (predictor) does not need to be between 0 and 1. On the down side, AUC is hard to generalize for the case of more than two classes. Also, naive AUC calculation that is based on rectangular approximation can be slow, at $O(n^2)$. If we use the probability based method, and sort the data with respect to the continuous variable, we can get $O(n \log n)$ time. With the AUC approach, we can pick the three inputs with the highest AUC with respect to the training label. The AUC for each input with respect to the label for the all three images (after getting rid of zero labels) is as followed:

AUC	NDAI	SD	CORR	DF	CF	BF	AF	AN
Running Time	0.9344	0.9043	0.8160	0.5215	0.3334	0.2148	0.1858	0.1928

Table 1: AUC of inputs with respect to output

Note that AUC is a measurement of between 0 and 1, which 1 can be thought of perfect positive correlation, 0 is similar to perfect negative correlation, and 0.5 means no correlation. Based on the table, we would pick

the three most "correlated" inputs, which are NDAI, SD, and CORR. AF's performance is quite closed to that of CORR, as AUC 0.1858 is equivalent (opposite sign) of 1 - 0.1858 = 0.8142.

Throughout this paper, we will use AUC as the main measurement for to choose the best model. When we need to get the actual predicted value of 0 and 1 instead of continuous predicted value (e.g. predicted probability), we will use a threshold, and measure performance by accuracy. We also note that just using raw NDAI as a predictor for classifying cloud and no cloud, the AUC is already 0.9344. Any model that performs not as good as this very naive approach should be discarded. And if fact as we see later, many of the simple models only have performance slightly higher than that benchmark. Put it in another perspective, the scientists who designed this NDAI signal have done a great job transforming somewhat weak radiances signal into a powerful signal.

3.2. Overview of Classifiers

When solving a classification problem, we are presented with an abundance of choices to make. Following is a broad breakdown by:

1. Model 1.1. Linear Regression 1.2. Logistic Regression 1.3. LDA, QDA 1.4. SVM 1.5. naiveBayes 1.6. randomForest 1.7. neural network
2. Feature Engineering 2.1. Include polynomial term, interactive term, e.g. $x_i^2, x_i x_j$ 2.2. Log-Rescale, squareroot rescale: $sign(x)\log(|x|+1)$, $sign(x)\sqrt(|x|)$
3. Regularization 3.1. L1 loss, L1 then OLS on selected variables, OLS then L1 on selected variables 3.2. L2 loss 3.3. L1 + L2 (Elastic Net) 3.4. Adaptive L1 (weighted L1) 3.5. Forward stepwise, backward stepwise selection
4. Model Selection, Choosing Model Parameter 4.1. Cross Validation 4.2. AIC, AICc, BIC
5. Performance Measure 5.1. AUC 5.2. Accuracy 5.3. Logloss, deviance, mutual information 5.4. F1 Score, Mean Average Precision, Cohen's Kappa
6. Optimization Algorithm 6.1. Gradient Descent family: Stochastic Gradient Descent, Coordinate Descent 6.2. Newton Method family: Quasi-Newton, BFGS 6.3. LARS (for L1 and Elastic Net)

Of course not all combination is possible, for example LAR algorithm is only applied for L1 and Elastic Net regularization. Still, we are left with a very wide range of options to choose from. For the scope of this lab, we won't have time to study and implement all the possible combination, and so we heuristically restrict ourselves to some specific set of options.

For most of the model, we try out of the box implementation with out much calibration. We pay more attention to Logistic Regression, and SVM in particular, which represenst the statistical approach, and optimization approach to classification respectively. For Logistic Regression, we try polynomial and interactive terms, L1 regularization with cross validation as the tool for picking the best regularization. Cross validation uses Area Under Curve of ROC curve as the measurement. We use the "glmnet" package, which impliments Coordinate Descent algorithm. Following is the list of models that we run on our dataset:

Model Specification:

1. Linear Regression: The response variable (binary 0 and 1 in our case) is a linear function of X with white noise.
2. Logistic Regression: Condition on X, the log odd is linear function of X.
3. naiveBayes: The input X's are conditional independent given Y.
4. Quadratic Discriminant Analysis: X for each group is Multivariate Gaussian
5. Neural Network: 1 hidden layers with 10 hidden nodes
6. Random Forest: 640 trees
7. Logistic with L1 Loss, CV on AUC
8. Logistic with interactive terms

9. Support Vector Machine: with optimized cost and gamma parameter

Some of the models are more of an optimization procedures than a statistical models, namely Neural Network, Random Forest (decision tree), and Support Vector Machine. As such there are really no assumption. We instead check the model assumption for the probabilistic models. For Linear Regression, it is clear that the assumption will not be met for binary responses. However, as we will see Linear Regression thought of as a Least Square method can still perform very well.

To illustrate the differences among classification algorithms, we borrow this image from Mark Landry. The two missing algorithms in his list that we did not get to try is Nearest Neighbor and Gradient Boosted Machines.

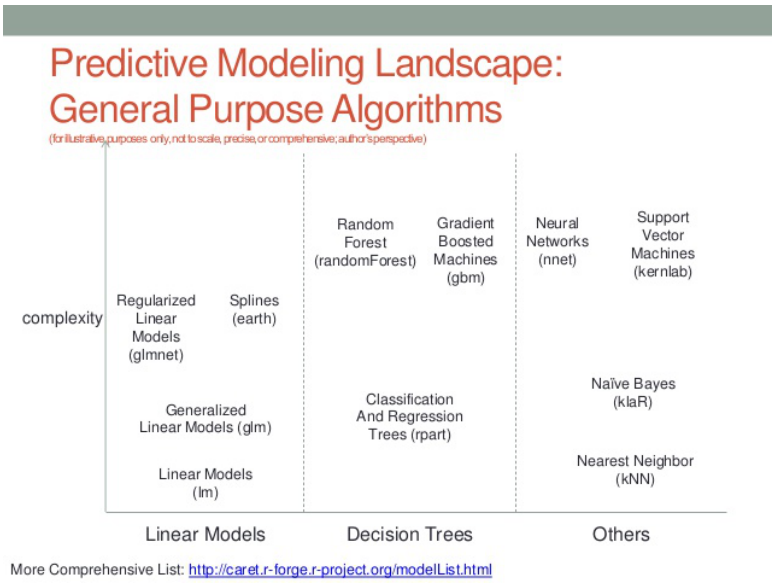
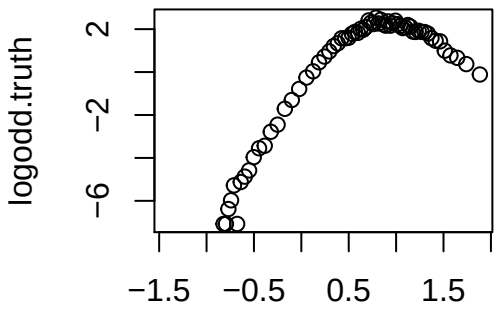
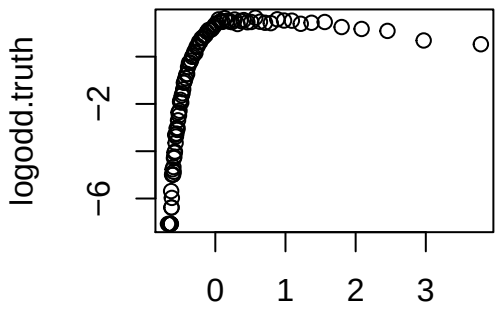
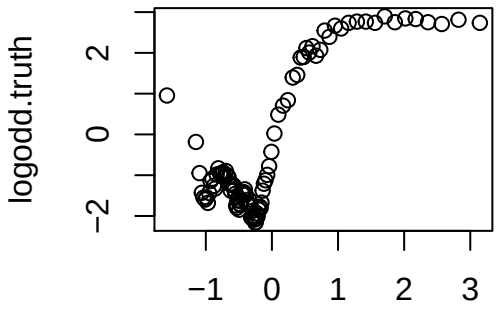


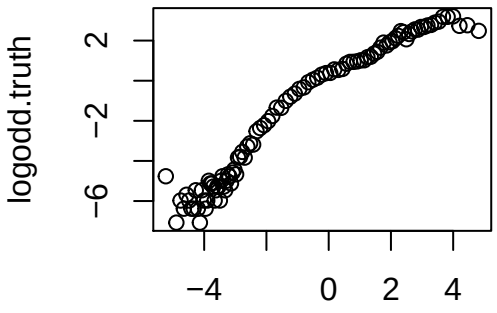
Figure 3: Classification Algorithms. Owner: Mark Landry. Url: http://www.slideshare.net/mark_landry/gbm-package-in-r

We check the model assumption for Logistic Regression. The log-odd should be linear in each of the inputs.









Looking at the plots of log-odd (in buckets) versus each of the inputs, we see that the plot of log odd with respect to NDAI, SD, and CORR are not linear. Including quadratic terms would help. This explains why we see a higher performance in QDA, or non-linear methods such as random forest and neural network.

For naive Bayes, condition on the label equal 1, we have the correlation of inputs are:

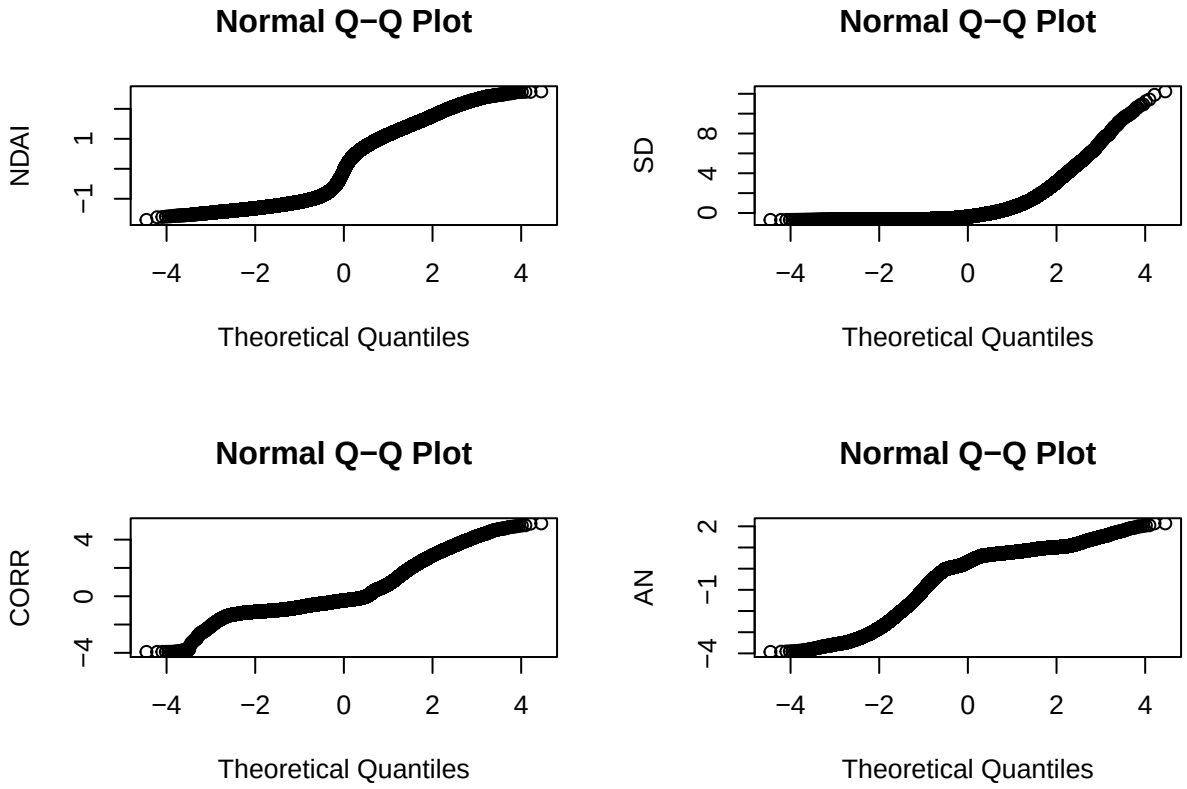
	NDAI	SD	CORR	DF	CF	BF	AF	AN
NDAI	1.00	0.69	0.39	-0.43	-0.51	-0.55	-0.61	-0.64
SD	0.69	1.00	0.34	-0.42	-0.48	-0.54	-0.59	-0.61
CORR	0.39	0.34	1.00	0.12	0.00	-0.13	-0.30	-0.44
DF	-0.43	-0.42	0.12	1.00	0.95	0.92	0.86	0.78
CF	-0.51	-0.48	0.00	0.95	1.00	0.98	0.92	0.86
BF	-0.55	-0.54	-0.13	0.92	0.98	1.00	0.97	0.92
AF	-0.61	-0.59	-0.30	0.86	0.92	0.97	1.00	0.98

	NDAI	SD	CORR	DF	CF	BF	AF	AN
AN	-0.64	-0.61	-0.44	0.78	0.86	0.92	0.98	1.00

Table 2: Correlation matrix condition on there is cloud

It is clearly that the correlation are quite high, thus the assumptions are not met. But still naive Bayes method often performs quite well even when the assumptions are not met.

For QDA, we need the inputs to be Gaussian condition on the class. Looking at the marginal Q-Q plot with respect to the normal quantiles, we see that none of the inputs have a linear Q-Q plot. So again the assumptions are not met.



3.3. Cross Validation Result for Different Classifications

For our data, the rows are not i.i.d. As such we have to be more careful in choosing the train set, test set, and cross validation sets. Our end goal in this data problem is to be able to classify cloud and no cloud in new images. We only have three images, one way to create more observations is to divide each image into k by k smaller images. Doing this, each block can be thought of as a separate image, and we have $3k^2$ images. These newly created images are not totally independent; still, dividing three images into small images should help us in building a more stable model on new images.

In our data, we choose k = 3, as such there are 27 small images. We choose 15 blocks at random to use as train, and leave the remaining as test. We do this 200 times, each time choosing 15 blocks at random,

calculating the AUC of the predictor with respect to the label in test set. The result is reported in the box plot below.

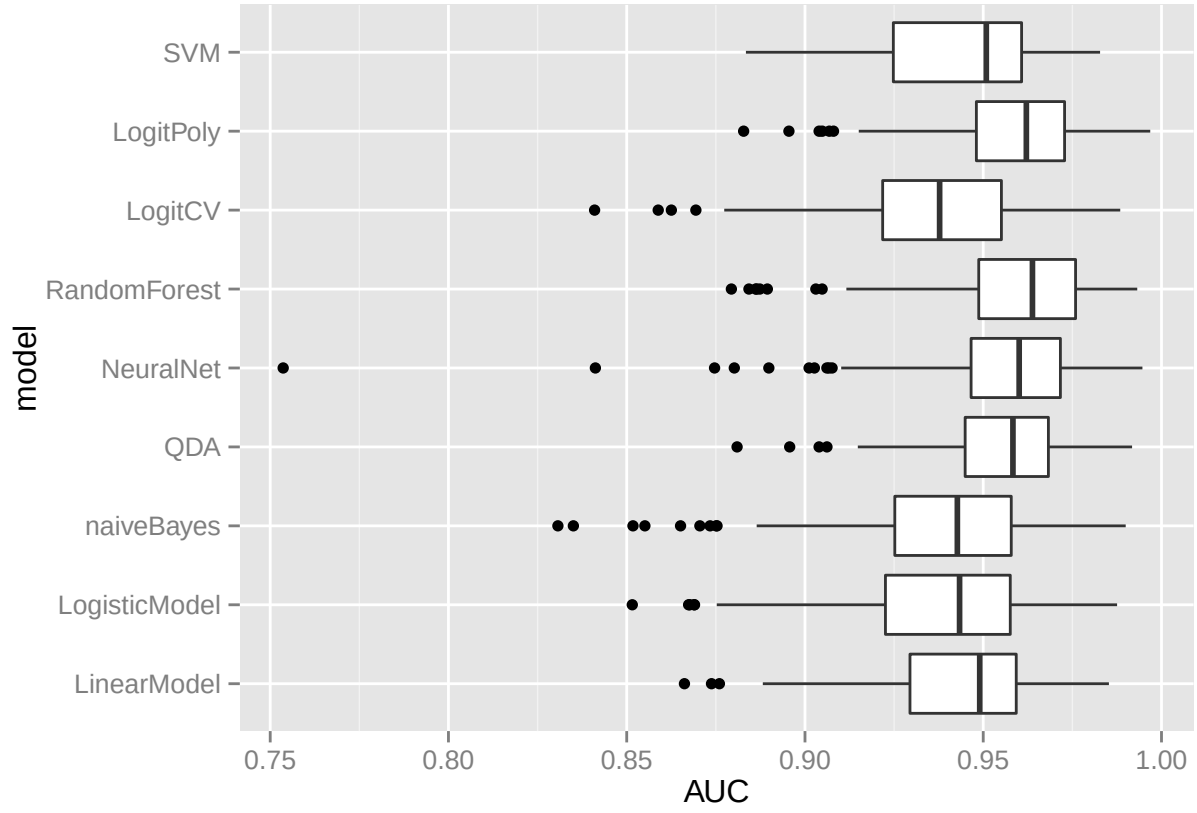


Figure 4: Classification Algorithm AUC Performance

We see that random forest have the highest peformance, followed by logistic with interactive terms, neural network, and QDA. The class of simple models namely linear model, logistic model, and naive Bayes are not as good but not too far behind. Regularization does not help logistic regression much, as we only have 7 variable and 100,000 observations, so there is not much overfitting.

In term of run time, we have the following table:

Model	Linear	Logistic	PolyLogit	QDA	NaiveBayes	NeuralNet	RandomForest	SVM
Run Time (s)	0.14	1.32	5.30	0.58	1.70	20.00	42.92	2hr

Table 3: Algorithm Running Time for one run on train data (100k observations, 8 inputs)

We see that in general simple methods run much faster. QDA seems to have a good combination of performance and computational cost. The SVM was run on the cluster while the remainder of the models were run on personal computers. The 2 hours shown is approximate.

3.4. Convergence of Parameter Estimation

Since random forest model does not really return any meaningful parameters, we will work on the logistic regression model for this subsection. We will look at the estimated β, when using one block to train, two blocks, and so on until we use all 27 blocks to train the logistic model. The inputs are standardized to zero mean and unit variance.

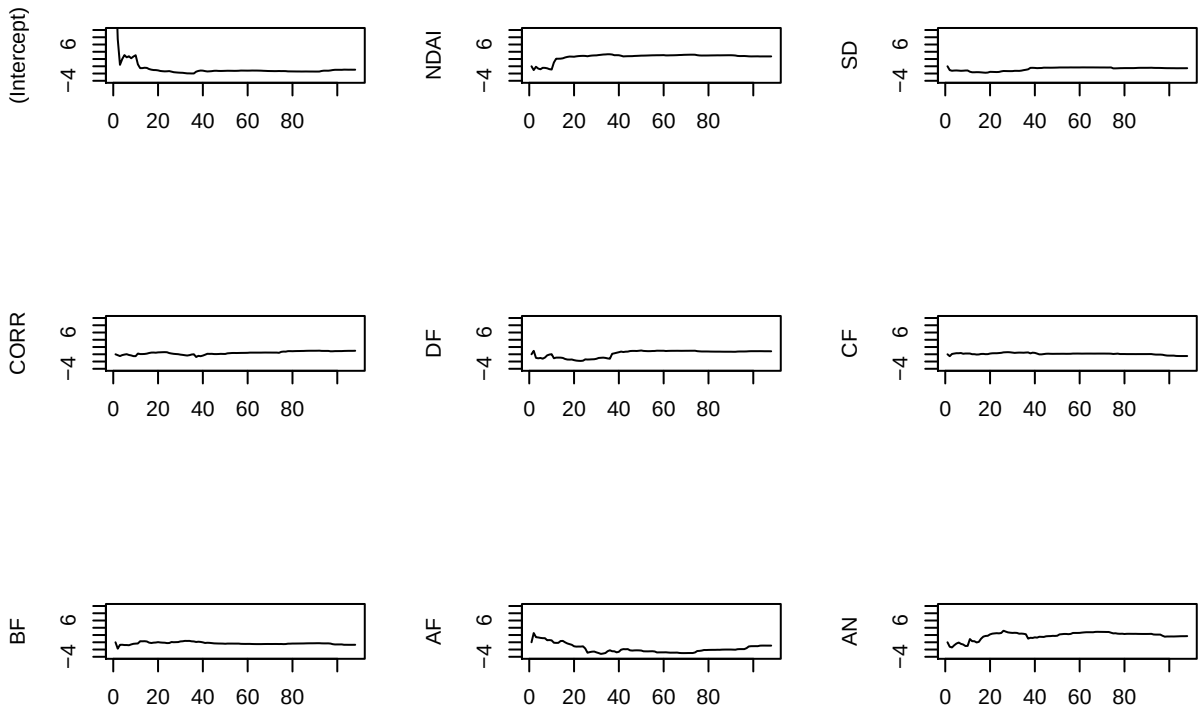


Figure: Convergence of betahat for Logistic Regression when training size increases

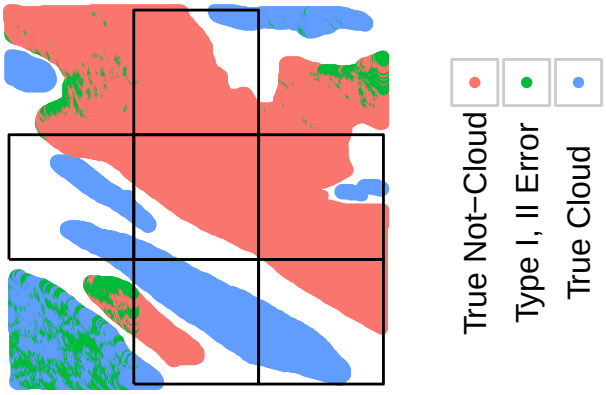
We see that the model is quite stable with respect to adding more blocks into the training data.

3.5. Missclasification Error

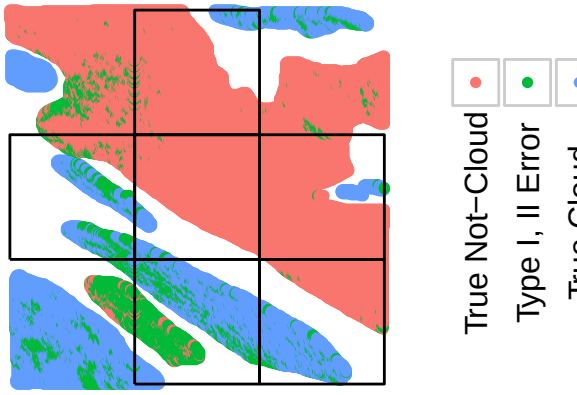
3.5.1. By image region

We first see missclassification error with respect to region in the image. We run Random Forest model on 15 blocks and then use the model to predict both the training and testing data. The training region is in the black boxes. We include training regions firstly for a comparision of how models work in sample and out sample, and secondly to make the image complete. We do the same for Logistic Model, as a comparision to the more complex Random Forest. The blue signifies cloud and red signifies no cloud. The three images on the left are for Random Forest, and the three images on the right are for logistic regression.

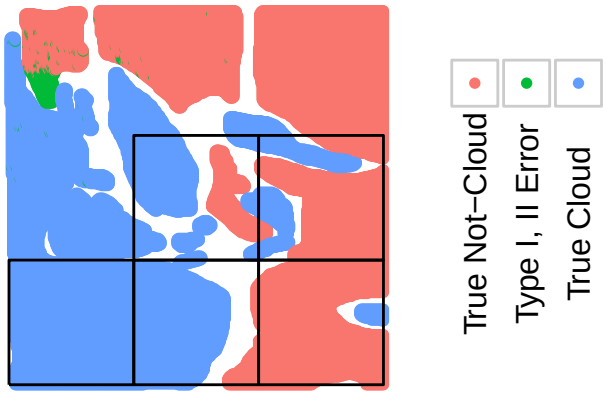
Classification Error for Image 1



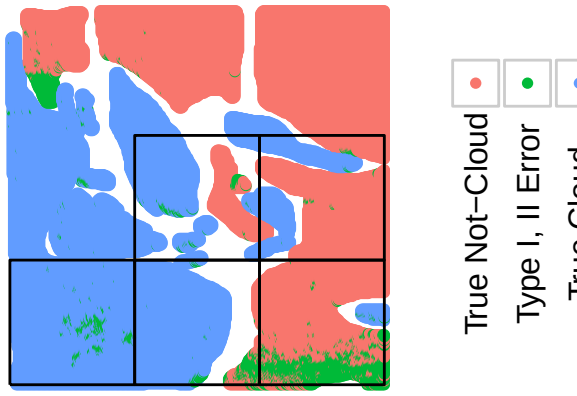
Classification Error for Image 1



Classification Error for Image 2



Classification Error for Image 2



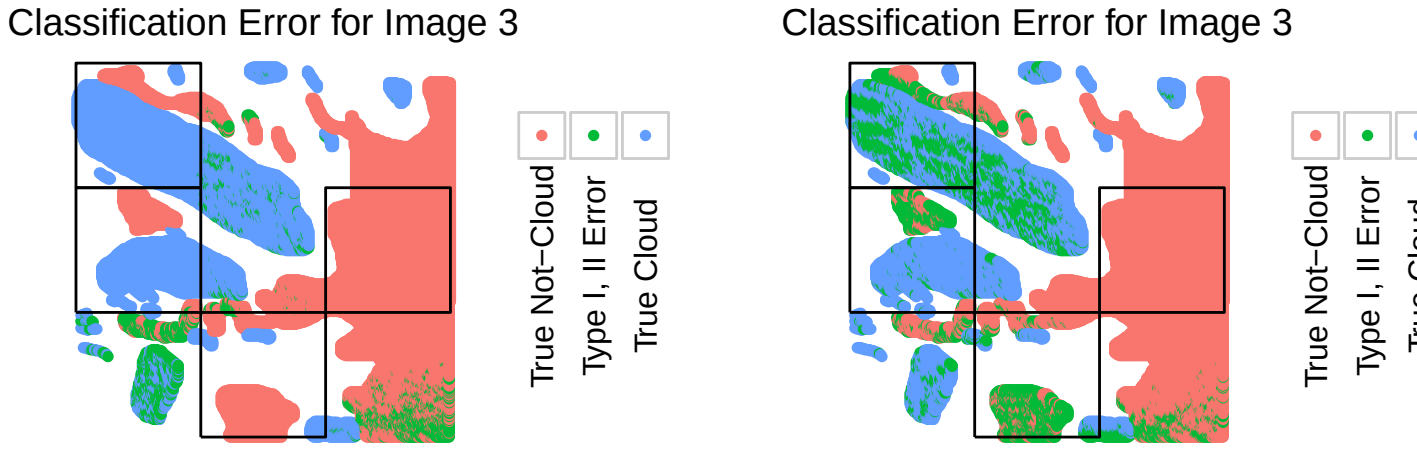


Figure: Left - Classification Error for Random Forest. Right - Classification Error for Logistic Regression. Regions inside blackbox are training data. Region outside blackbox are test data.

For both model, we can see that our models seem to do a pretty good job of predicting. However there are a few areas where we run into problems. One consistant problem is our model predicting clouds on land that is on the edge of the unknown(the white areas). However, once we get to the middle of a big cloud or a big no cloud area, our models do a very good job of consistantly predicting correctly. Another problem is that it was trying to predict land in the middle of clouds. There seems to be a scattershot of error throughout most large clouds where the model sprinkles land throughout a cloud.

The one very interesting feature of Random Forest is that it almost did not make any errors in the training blocks. This may be because Random Forests can overfit to the training data. As a decision tree model, Random Forest can easily fit the training data perfectly. Logistic regression as expected have equally good performances in and out of sample.

3.5.2. By range of input

NDAI	SD	CORR	DF	CF	BF	AF	AN
0.00	0.10	0.52	0.61	0.30	0.23	0.22	0.20
0.19	0.27	0.09	0.15	0.26	0.20	0.21	0.23
0.16	0.38	0.14	0.09	0.12	0.14	0.09	0.08
0.15	0.69	0.06	0.03	0.02	0.04	0.03	0.08

Table 4: Table of Classification Error for Logistic Model. The row are quartile of data. For example, the top row entry in the column of NDAI means for the lowest quartile of NDAI value, Logistic Model misclassification rate is 0.00.

From the table, we see that when SD is high, Logistic Model performs worse. These mean when the radiances of different angles around one pixel are very different, and correlation are very low, our model perform worse. We are not sure why this is the case. For the radiances, when the radiences are low, which means weak light, our model performs worse. This meets our expectation as when cloud is high, it is easier to predict cloud,

and high cloud means high radiences.

We also provide the table for Random Forest.

NDAI	SD	CORR	DF	CF	BF	AF	AN
0.01	0.10	0.23	0.43	0.58	0.62	0.49	0.45
0.12	0.20	0.09	0.27	0.38	0.30	0.22	0.20
0.18	0.16	0.13	0.06	0.09	0.12	0.07	0.07
0.39	0.28	0.36	0.03	0.02	0.04	0.06	0.04

Table 5: Table of Classification Error for Random Forest. The row are quartile of data. For example, the top row entry in the column of NDAI means for the lowest quartile of NDAI value, Logistic Model misclassification rate is 0.00.

We see quite similar pattern of misclassification for Random Forest. It seems to not perform well for high NDAI, high SD, and low radiances. Exception is CORR where Random Forest does not perform well for high CORR.

3.6. Future Data

We believe that our model should perform well in out of sample data, as long as the new images are taken from similar regions. This is because we randomly select a different blocks of images as train and test out of sample performance. We also did this multiple times (200) to increase our confidence in out of sample performance. For Random Forest, it should perform better with regularization, as we saw that in sample it overfits.