Lab 4

Huong Vu, Aliyah Hsu, Omer Ronen

November 20, 2020

1 Introduction

Global warming is an undeniable phenomenon and it is happening at an increasing rate. To build highly accurate global climate models, we need a better understanding of the dependency of the Earth's surface air temperatures and the atmospheric carbon dioxide levels. One way to understand this dependency is through analyzing cloud coverage, especially at the Arctic where the dependency is predicted to be the strongest. However, cloud detection on Arctic satellite images is particularly challenging due to similarities in characteristics of cloud-, snow- and ice-covered surfaces in the Arctic. Therefore, the goal of this project is to develop accurate cloud detection algorithms using data from Multi-angle Imaging SpectroRadio (MISR). In this project, we will first analyze the relationship between radiance features and engineered features from Shi et al. [2008], then implement three classifiers: Logistic Regression, Random Forest and Neural Network and analyze the result of the best performed classifiers.

2 Data Description

The data used in this report is three images collected from MISR. Each pixel in the images is labeled with its `x` and `y` coordinates, categorical cloud indicator (no cloud = -1, unlabeled = 0, and cloud = 1), three engineered features (`NDAI`, `SD` and `CORR`), and radiance features (`DF`, `CF`, `BF`, `AF` and `AN`). The radiance features are essentially the radiance received by satellite cameras at different zenith angles: 70.5°(`DF`), 60.0°(`CF`), 45.6°(`BF`), 26.1°(`AF`) in the forward direction; 0.0°(`AN`) in the nadir direction. `NDAI` stands for Normalized Difference Angular Index, which is the normalized difference of the means of `DF` and `AN`. `SD` is the standard deviation of MISR nadir camera pixel values across a scene. `CORR` is the correlation of MISR images of the same scene from different MISR viewing directions. The categorical cloud indicator is hand-labeled by domain experts and will serve as the ground truth in our analysis. Since unlabeled pixels provide no information to help us evaluate our models, we exclude them and make our data with only binary classes (no cloud = -1, cloud = 1).

3 EDA

We first visualize the raw data to get a general picture of the images (Figure 1). Again, even though here we present all pixels but since the unlabeled pixels would not contribute to the cloud prediction, we exclude the unlabeled pixels in the following analysis. Note that the three now binary-labeled (no cloud / cloud) images will serve as the ground truth throughout the analysis.

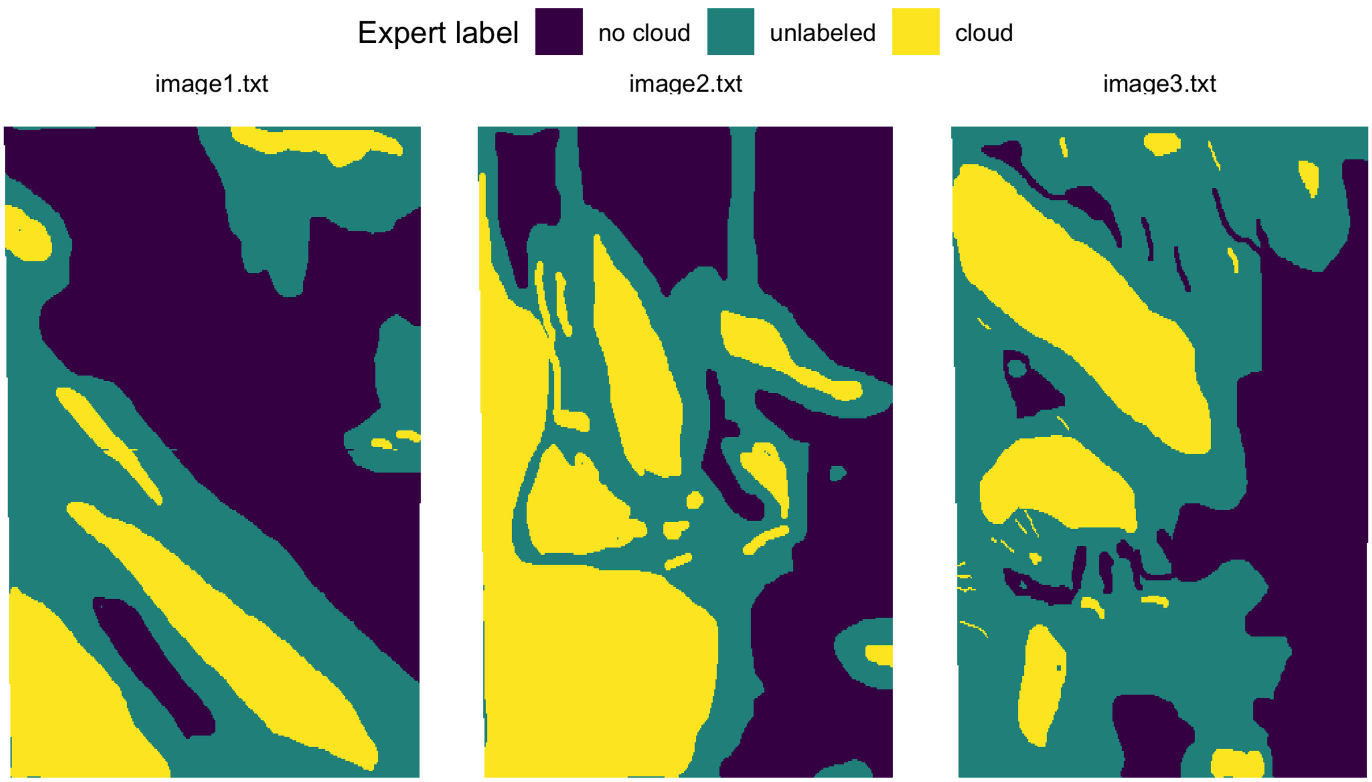

Figure 1: Plots of expert labeled pixels

3.1 Features Correlations

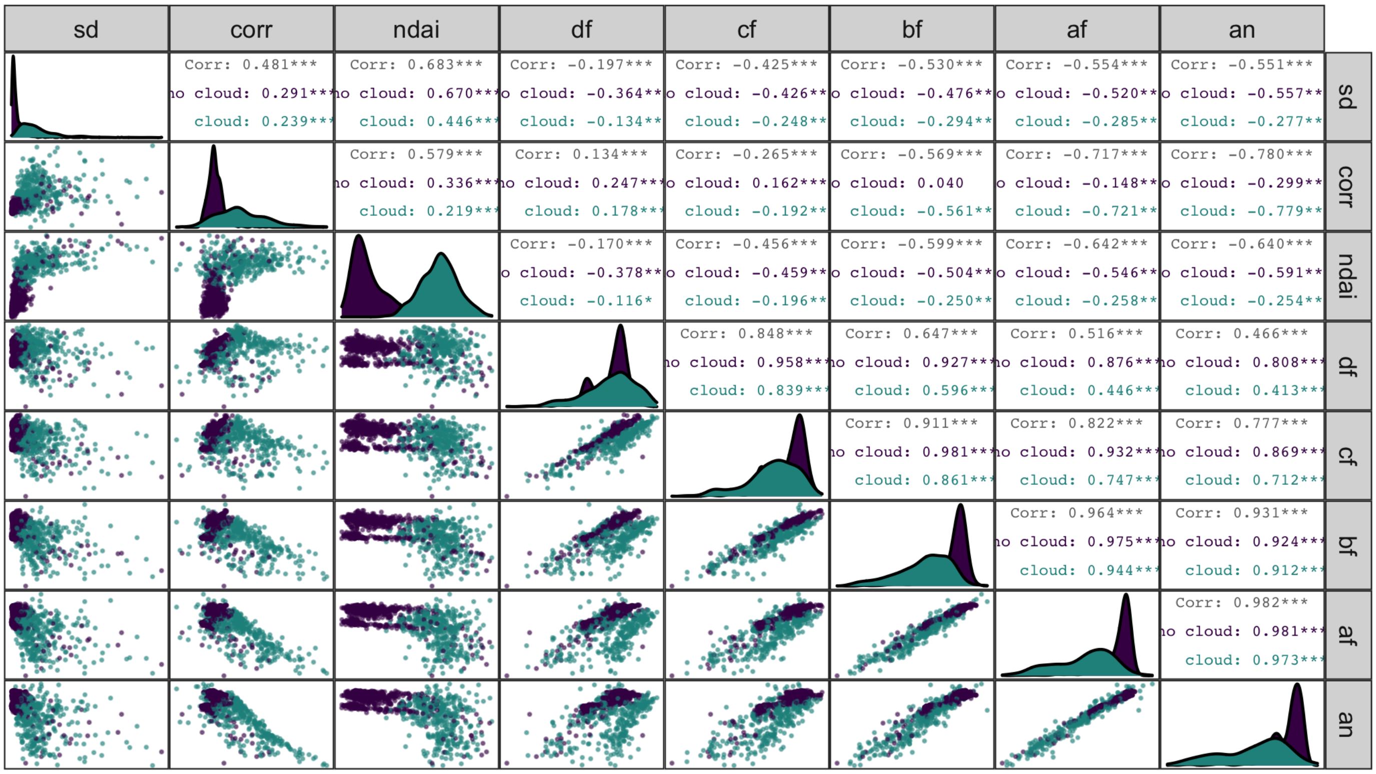

Figure 2: Correlations of the eight features: SD, CORR, NDAI, DF, CF, BF, AF, AN of the data

We next look at the relationships of the features in the data, and try to figure out which features are the most representative for cloud / no cloud separation. After plotting the correlation of the features in Figure 2, we can observe the following trends:

1. The three engineered features `SD`, `CORR` and `NDAI` are approximately negatively correlated with the radiance.
2. The radiance are not only positively but also highly correlated with each other. In addition, the correlation between the radiance is higher when a pair of radiance has similar angle, like `DF` and `CF` or `AF` and `AN`.
3. The separation of the no cloud and cloud classes is more distinct in the three engineered features and `AN`. If we would like to further rank them by the degree of separation of the two classes, the ranking would be `NDAI` > `SD` > `CORR` > `AN`.

3.2 Principle Component Analysis

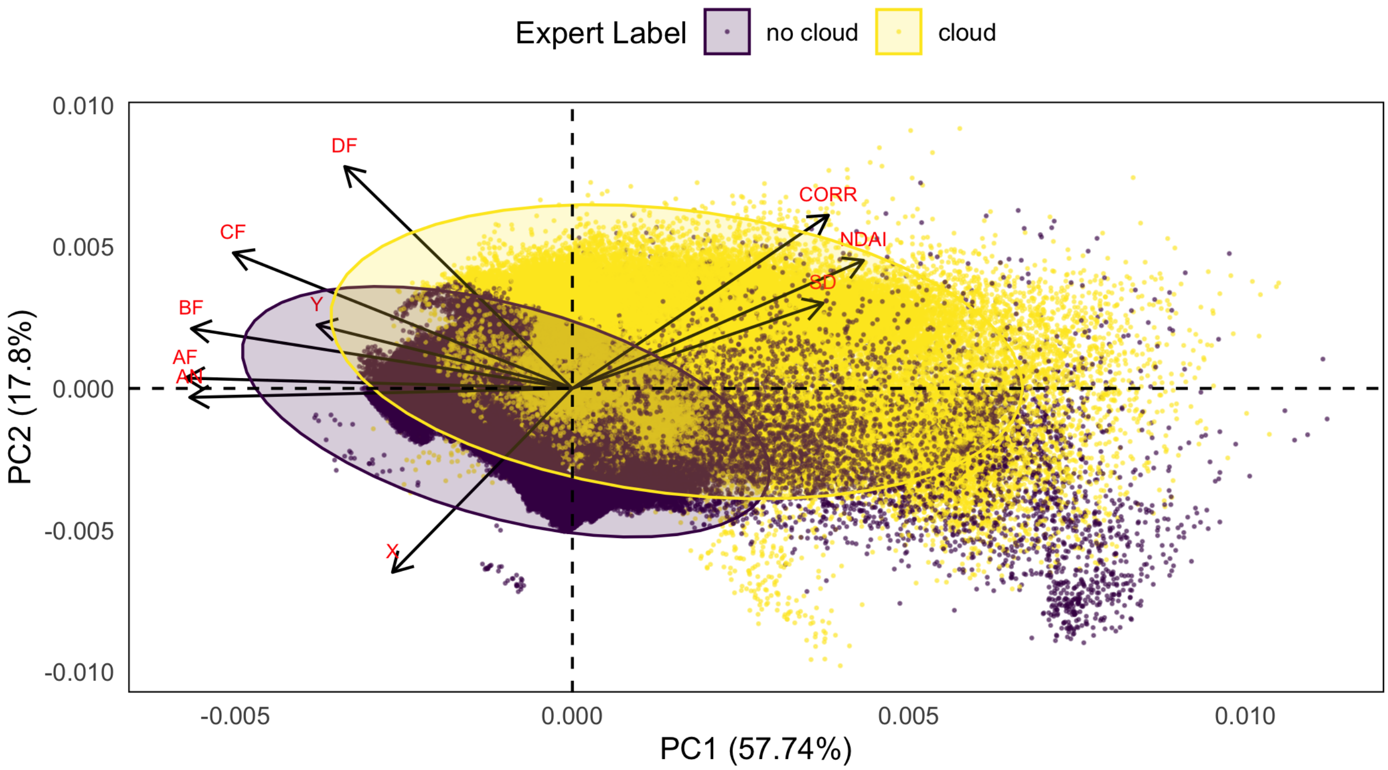

Figure 3: First two principle components and features correlation and contributions to top two principle components

We look deeper into the relationships between features using the Principle Components Analysis (PCA). Since the features have different values scales, we center and scale before performing PCA. In addition, since using different subsets of data will result to different plots of PCA, we use all labeled data to perform PCA to have a broad understanding of the data. In Figure 3, the ellipses represent 95% concentrated level between cloud and no cloud groups. From the figure, we can also see the relationships between all features: `CORR`, `SD`, and `NDAI` are positively correlated with each other and negatively correlated with radiance features. This observation matches with the observations we find using the correlation plots. The arrows also indicate the contributions of each feature to the top two principle components. For example, features `AF` and `AN` contribute largely to the first component while features `DF` and `CORR` have large contributions to the second principle component.

3.3 Features Heatmap

We know that `SD`, `CORR` and `NDAI` are calculated from other radiance features hence will contain information of other features. From the features correlation analysis and PCA results, we know `SD`, `CORR`, `NDAI` and `AN` are the most representative features to separate the no cloud and cloud classes. We next look deeper into the spatial relationships of the features through exploring their heatmaps. In Figure 4, we show the heatmaps generated from image 3 as an example. Comparing with the ground truth of image 3, we can see the features indeed captured certain information of the ground truth. In addition, the four features' ability to capture image information can be ranked roughly as `NDAI` > `AN` > `SD` > `CORR`.

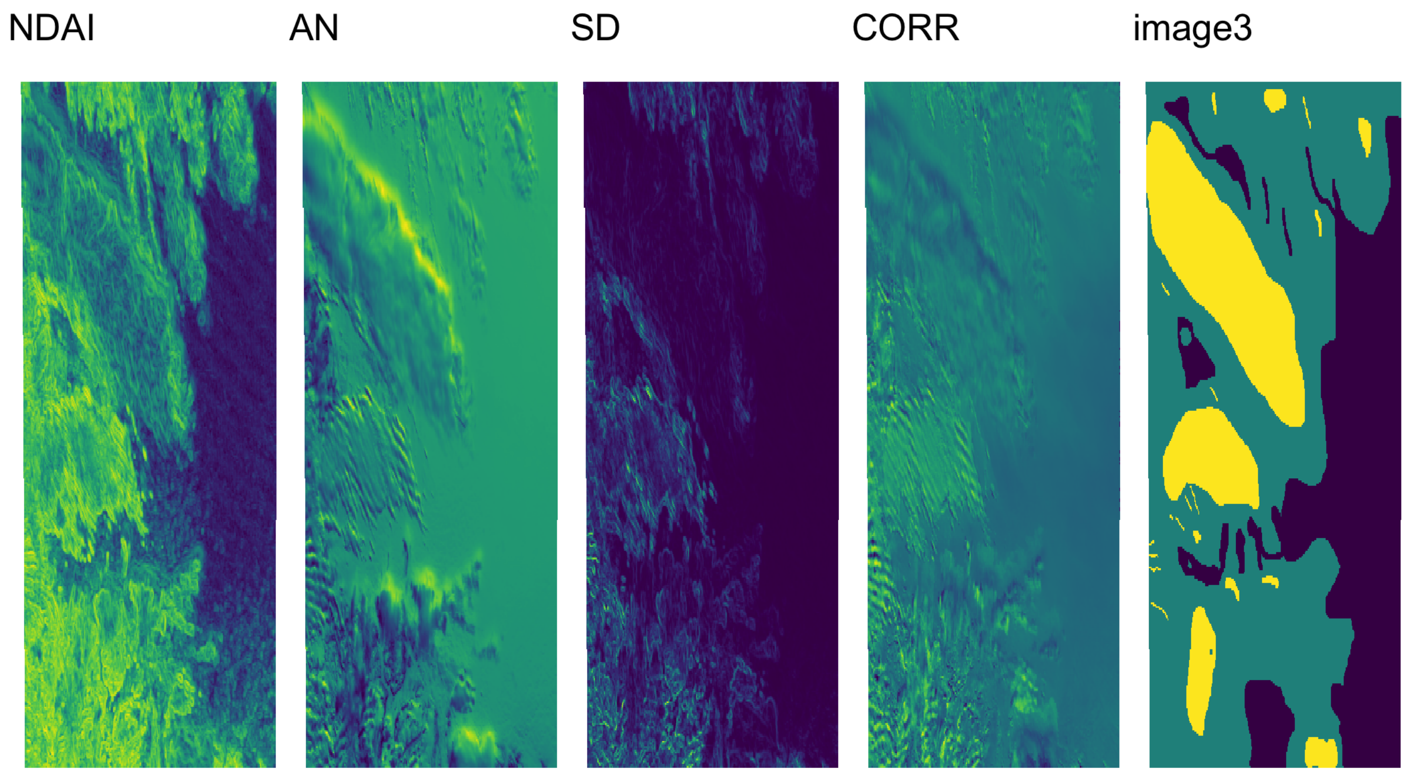

Figure 4: Heatmaps of features: NDAI, AN, SD and CORR, with ground truth aside for easier comparison

4 Data splitting

Since the pixels are spatially correlated with each other, we use the entire image 2 as the testing set to ensure no information of the testing data leaked into the training data. For cross validation, preferably, we would like to have each cross validation fold is a set of images. However, we only have 2 images for training data. Therefore, we decide to cut image 1 and 3 into 4 smaller images on the vertical axis. Each smaller image will be a cross validation fold. This method does not completely prevent the leaked spatial information between folds but minimizes the impact of the spatial relationship between pixels in the same image on training performance.

5 Most Importance Features

Even though PCA analysis gives us how features contribute to the top principle components of the data, the result is inconsistent with different subsamples. Hence, we decide to use random forest model to find the top important features and have stronger support evidence for the top important features. Random forest model does not involve any assumption besides the sample data is representative. To ensure the assumption, we subsample 80% of training data and run Random Forest model for 5 times to obtain feature importance and top 3 most important features by ranking are `NDAI`, `SD` and `AN`. This result is consistent with our analysis on feature correlation earlier. The feature importance from Random Forest model is calculated by the average decrease in Gini index over all decision trees and Gini index is the measurement of node impurity from splitting on the variable.

6 Classifiers

Here we introduce the three classifiers used in our analysis. We also discuss the training process of the models and how we tackle with their model assumptions.

6.1 Logistic Regression Model

Logistic Regression model is often used when the outcome is a two-level categorical variable, like the no cloud or cloud classes we are predicting. Logistic Regression is a type of generalized linear model, and can be essentially think of as a two-stage modeling approach. In other words, we first model the response variable with binomial probability distribution, and then we model the parameter of the distribution using a set of predictors and a logit transformation. We train the Logistic Regression model with cross validation with assigned folds.

There are two main assumptions for Logistic Regression to work:

1. Linear relationship between predictors and log-odds.
2. Correlated predictors can inflate variance and bias of coefficients.

We explore the relationships between the three predictors (`NDAI`, `SD` and `AN`) and the log probability of the cloud prediction to test the first assumption. We found empirically from several training trials that the relationships of the three features and the log-odds are only linear under this condition: `NDAI` less than or equal to 1.5, `SD` less than or equal to 5, and `AN` in the range of 175 to 225. So we restrict our data for the three features to be within the ranges to meet the assumption when training the logistic regression model. We demonstrate the linear relationships of the three features and log-odds in Figure 5. The negative relationship of `AN` and log-odds makes sense since it's natural to think if there's cloud then the radiance received by the satellite camera in the `AN` direction should be blocked.

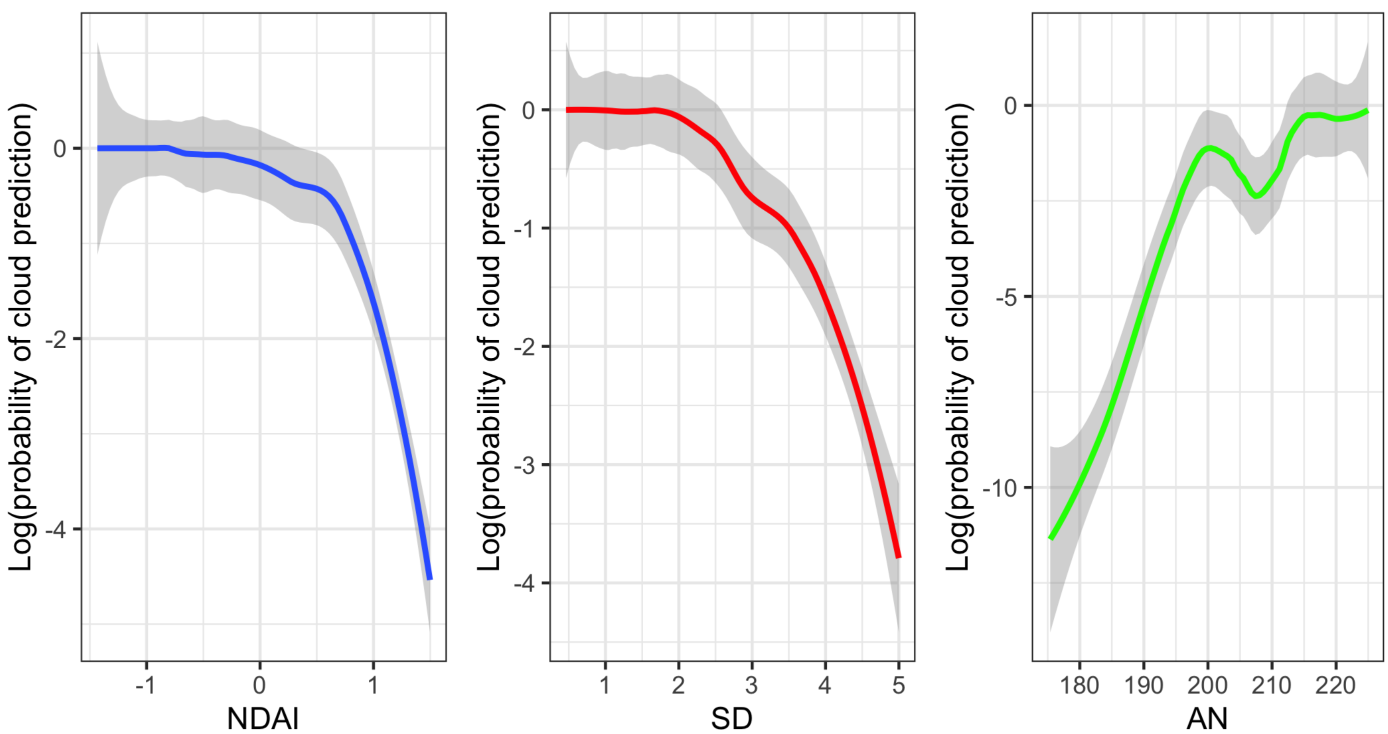

Figure 5: Assumption test of Logistic regression: relationships between predictors and log-odds

Next, for assumption two, we would like to investigate whether correlated predictors can inflate variance and bias of the coefficients. To test this assumption, we add in the radiance features, in addition to the three features, to train a logistic model. Since we know the radiance is highly correlated with each other, as demonstrated in the EDA section, the predictors in the model are correlated. We compare the variance of this model with our original logistic model trained with only three features in Table 1, and find the variances of the coefficients of `NDAI`, `SD` and `AN` indeed inflate after adding the correlated predictors to our model. Hence, assumption two is also fulfilled in our logistic model.

Table 1: Comparison of the variance of the coefficients

	NDAI (e-03)	SD (e-04)	AN (e-04)
original logistic model	1.40	4.58	4.31
correlated logistic model	1.83	5.80	209.00

6.2 Random Forest

A decision tree is a building block of Random forest model. In a decision tree, at each node, a condition on a feature to split the data to left or right branch is implemented. In the end, the decision tree will provide the class in which the data is most likely belong to. A random forest model consists a large number of decision trees and the final class for a data point will be the one with the most number of votes. We have two assumptions for random forest model to run well:

1. There must be some signals in the features so the model can pick up and perform better than just random guessing.
2. The predictions made by the individual trees have low correlations with each other.

The first assumption is safe to make since even when snow- or ice-covered surfaces look similar to cloud on image, snow, ice and cloud are different materials and hence would reflect different levels of light radiance. To ensure the second assumption, Random Forest uses both bootstrap aggregation method which allows each decision trees to randomly sample from the dataset with replacement and feature randomness which allows individual trees choosing features to split from a random subset of features.

In the project, we use cross validation when training the model. After training Random Forest model with all features, we observe that the feature importance drops significantly after the first 3 features and the performance of the model is actually better based on ROC metric when trained with only the most important features: `NDAI`, `SD`, `AN`. Hence, we use only the top 3 most important features for Random Forest model.

6.3 Neural Network

We used a feed forward Neural Network as a classifier to predict the cloud/no cloud label. The architecture includes three hidden layers with 30 neurons and softmax output. The model was trained using the features: `NDAI`, `SD`, `CORR`, `DF`, `CF`, `BF`, `AF`, `AN`. We trained for 20 epoches with batch size 240 using Adam optimizer (default learning rate) and binary cross entropy loss. The chosen architecture defines a family of parametric distributions of the label given the features. We assume that after the training process is completed, the distribution we get is a good approximation of the distribution of the data.

It is rather hard to validate this assumption, looking at how the predictions compared with other methods we can probably say that the assumption does not hold but provides a reasonable approximation.

7 Model Evaluation

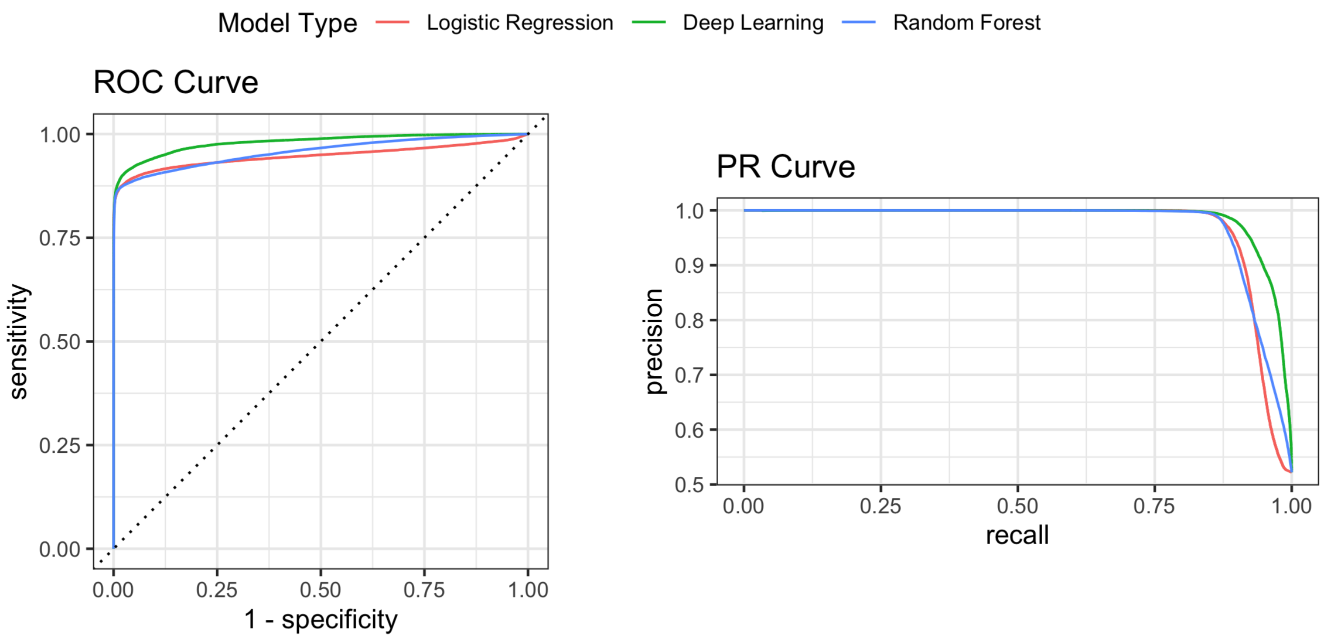

Figure 6: ROC curve and PR curve of Logistic Regression, Random Forest, Deep Learning models

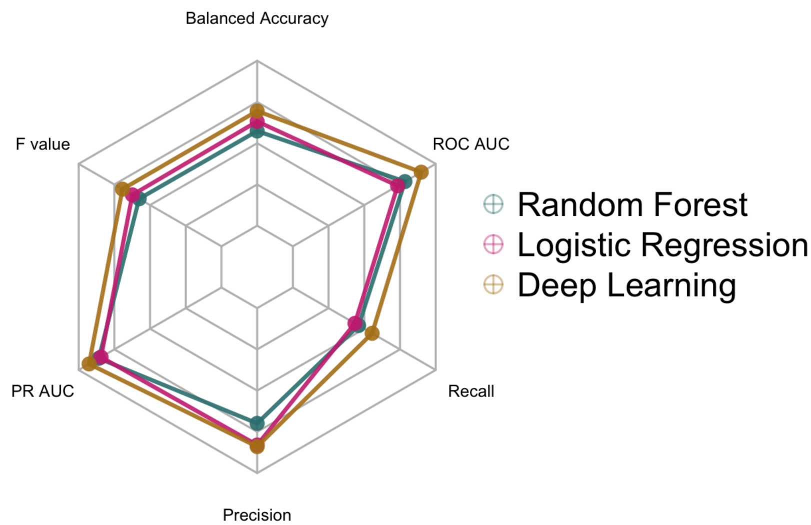

Figure 7: Radar plot to compare all three models performance

After evaluating and comparing the performance of Logistic Regression, Random Forest, and Deep Learning Neural Network model, we decide to choose Deep Learning Neural Network model as the final model. When comparing the models in ROC and PR curve plots (Figure 6), we can see that overall Deep Learning model performs better than Logistic Regression and Random Forest model. Similarly, when comparing between models in other metrics (Figure 7), we see that Deep Learning model has highest values in ROC AUC and PR AUC. Random Forest and Logistic Regression alternatively perform better than the other in different metrics. We choose Deep Learning model as our final model because we favor the model that has highest ROC AUC and PR AUC. Since those two metrics can ensure the generality of the model since they take into account a range of probability thresholds unlike other metrics where we set the probability threshold at 50% and calculate the value. Since the given data for this project is very limited, ensuring the generality of the model so that it will work well on new data is our top choice.

8 Post-hoc EDA of Deep Learning Neural Network Model

8.1 Training analysis

The first thing to inspect is the convergence of gradient decent in training to assume we have reached a local minima.

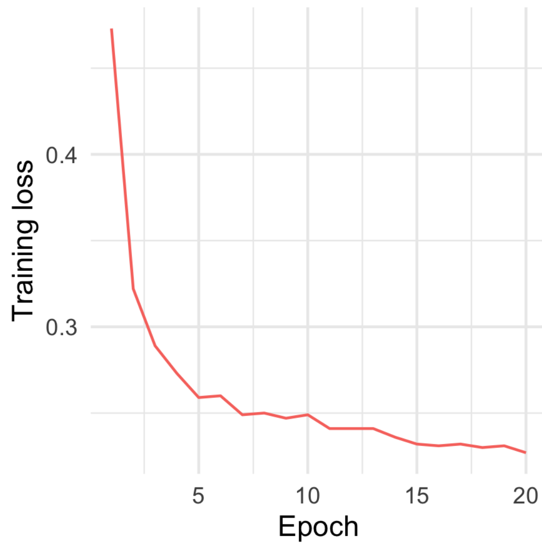

Figure 8: Training loss value along training epochs

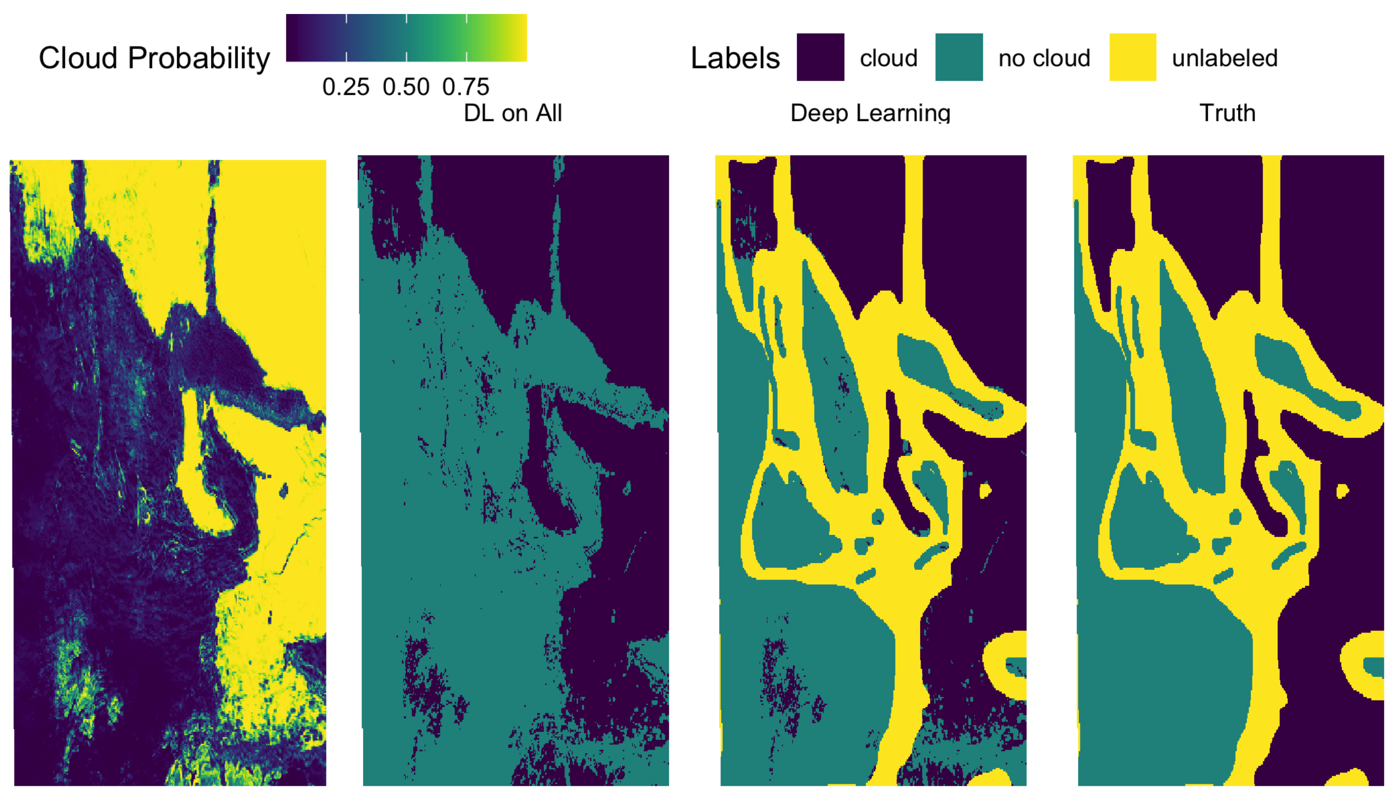

Figure 9: 1. Heatmap of Deep Learning model probability prediction for cloud; 2. Deep Learning prediction on all pixels; 3. Deep Learning prediction on labeled pixels; 4. Original image

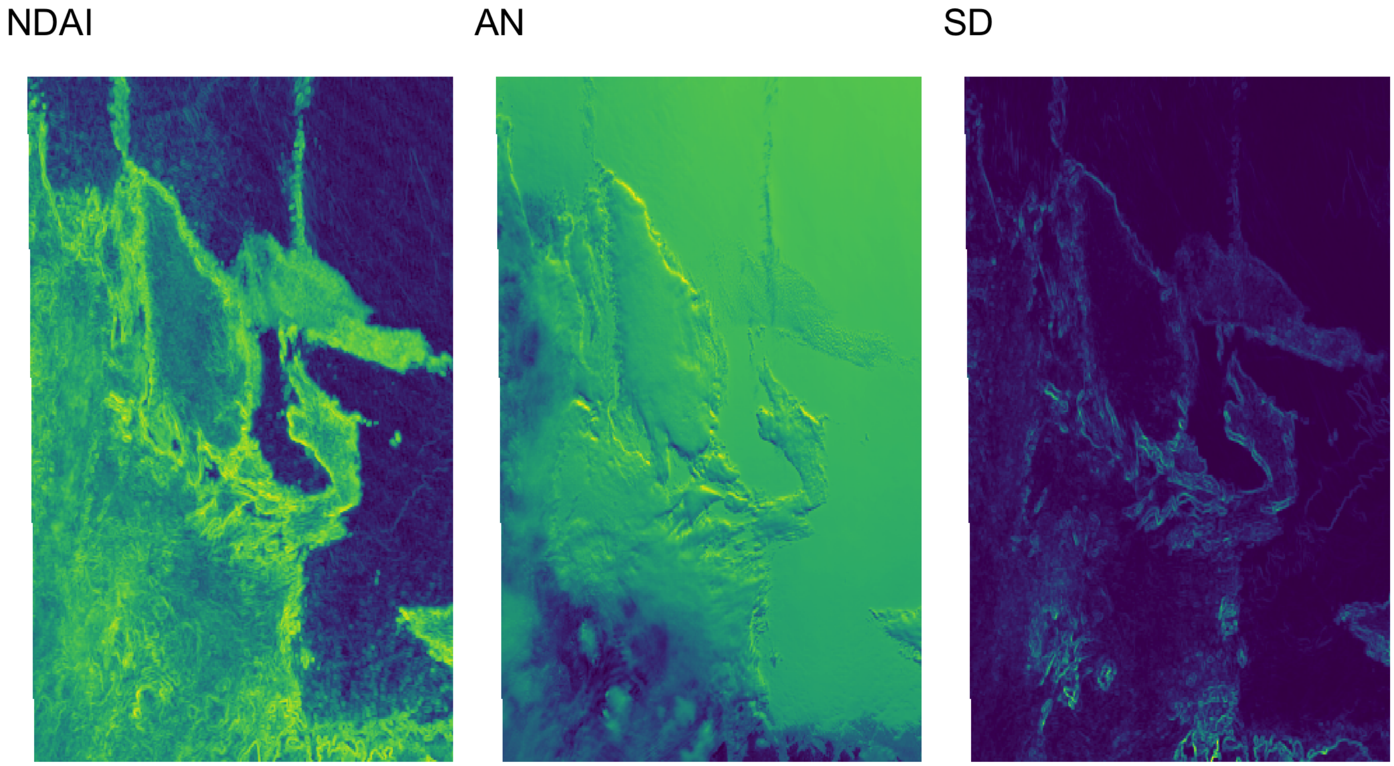

Figure 10: The heatmaps of the three important features on the testing image

Examining the prediction of Deep Learning model on labeled pixels with the ground truth image, we see that the majority of the pixels are predicted correctly except for those at edges of regions, a cluster of pixels in the middle of image 2 and a cluster of pixels at the left and right bottom corners being missclassified as no cloud instead of cloud (Figure 9). We look deeper into the feature distributions of the misclassified pixels to identify ranges of features that help explain the missclassification pattern. From Figure 10, we can see that the predicted cloud probability depends heavily on `NDAI` features. The heatmap of `NDAI` helps us explain a lot about the missclassification pattern in our prediction, especially those at the edges of the regions and in the middle of the image. For the missclassified pixels at the left and right bottom corners, those can be explained by looking at left and right bottom corner of `AN` and `SD` heatmaps, we can see the two opposite values where the color is darker in `AN` heatmap and lighter in `SD` heatmap. We further analyze the missclassification pattern quantitatively by dividing the misclassified pixels into type I and type II error groups, and comparing them with the feature distribution of the whole testing set to see if we can observe similar patterns with the heatmap. From Figure 11, we find that the mean of `NDAI` feature between type I and type II error differ from each other noticeably. For `AN` and `SD` features, even though the difference in means between groups are not significant, within the same type of error, the means of `AN` and `SD` features are often in the opposite direction compared to the whole testing set's mean.

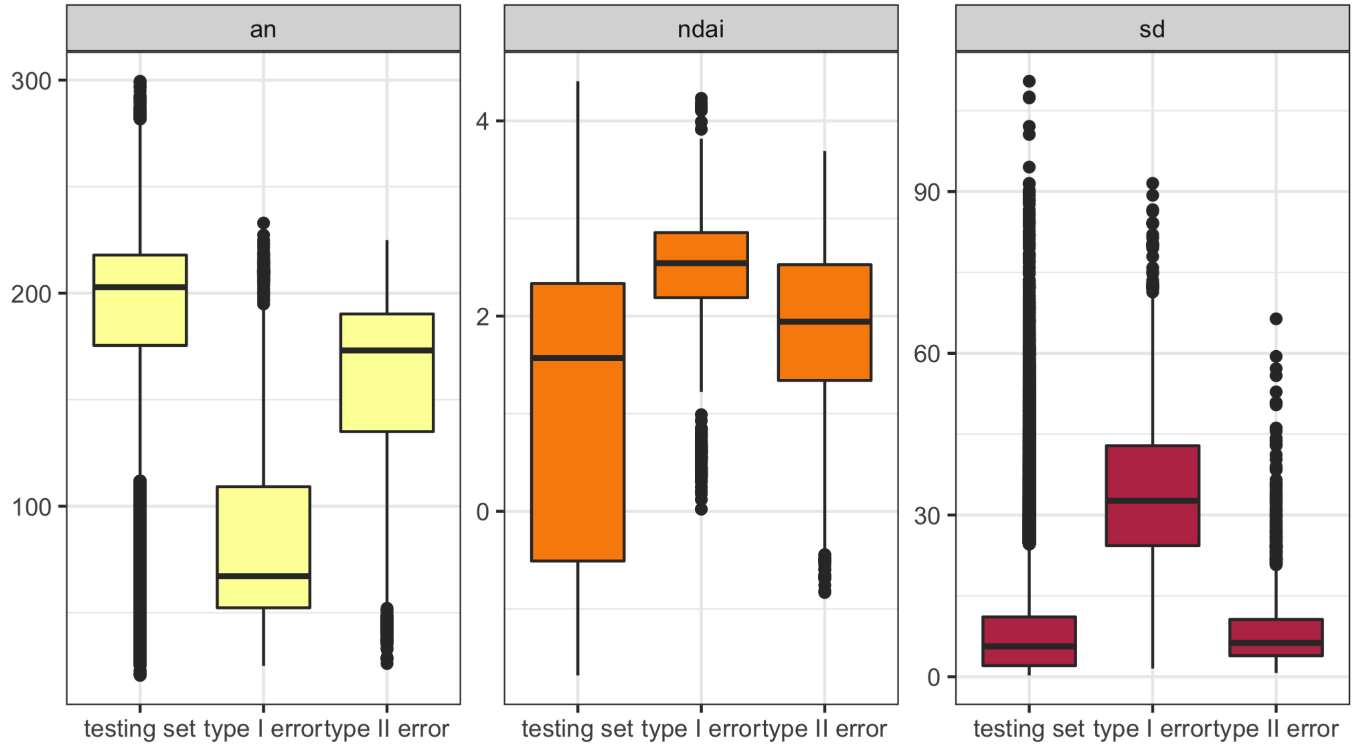

Figure 11: Feature distributions of the testing set, type I error group and type II error group

9 Conclusion

For the future use of the Deep Learning model in classifying cloud and no cloud regions on MISR images, we would need a better understanding of the ultimate goal of the classification in building climate models. Cloud is not a uniformly dense object and there will be regions in a cloud that are less dense than other regions. The density of clouds is reflected by different gradients in the features' heatmaps. Therefore, if the end goal of the classification step is to reflect the true cloud coverage on the image, then our predictions (both probability prediction and hard prediction) reflect the reality of cloud coverage truly. However, if the end goal of the classification step is to identify block of cloud and no cloud pixels like the expert labels, we can using a smoothing algorithm to provide a smoother areas of cloud and no cloud on the image. In conclusion, given the amount of data, the Deep Learning model has performed very well and with the priority as model's generality when choosing the final model, we believe that our model will perform well on future data even without expert labels.

Bibliography

Tao Shi, Bin Yu, Eugene E Clothiaux, and Amy J Braverman. Daytime arctic cloud detection based on multi-angle satellite data with case studies. *Journal of the American Statistical Association*, 103(482):584–593, 2008.