Stat 154: Project 2

Team name: Team Super-duper
Author: Jiawei Wu, Ming Chen

Data Collection and Exploration:

The research paper named "Daytime Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies" illustrates a state-of-the-art solution about cloud detection in the Arctic to assist atmosphere scientists to form systematic studies of the dependences of surface air temperatures on increasing atmospheric carbon dioxide levels that will occur in the Arctic. One of the challenges to do cloud detection in the Arctic is that we have to deal with the similar remote sensing characteristics of clouds and ice and snow-covered surfaces. Therefore, the key part of this research is to find the useful features and come up with an algorithm that can accurately separates clouds from ice and snow-covered surfaces.
This study collected the data from 10 Multiangle Imaging SpectroRadiometer (MISR) orbits of path 26 over the Arctic Northern Greenland and Baffin Bay. the repeat time between two consecutive orbits over the same path was 16 days, so the 10 orbits span approximately 144 days from April 28 through September 19, 2002 (a daylight season in the Arctic). Moreover, each path is divided into 180 blocks, and each MISR pixel covers a 275m * 275m region on the ground. Within the study, 6 MISR blocks form each orbit are included, making a total of 60 data units. However, 3 of the 60 data units excluded because the surfaces were open water and the current algorithms are already able to separate clouds from water very well.
With EDA and domain knowledge, three useful features that can differentiate surface pixels form cloudy ones are demonstrated in the paper, which are the linear correlation of radiation measurements form different MISR view directions (CORR), the standard deviation of MISR nadir red radiation measurement within a small region (SDAn), and a normalized difference angular index (NDAI). More importantly, based on these three features, the study intend to make an enhanced linear correlation matching (ELCM) algorithm that is more accurate and provides better spatial coverage than the existing MISR operational algorithms for cloud detection in the Arctic. The paper also introduced an ELCM-QDA algorithm, even though it does not improve overall agreement rates with expert labels relative to the ELCM algorithm, it goes beyond ELCM's binary labels of cloudy versus clear by providing probability labels.
There are many positive impacts of this study. It will not only enable the scientific community to study how changing cloud properties may enhance or ameliorate any changes in the Arctic when concentrations of atmospheric carbon dioxide increase, but also this study serves as a great example of the power of statistical thinking and show us how to apply our statistics knowledge to contribute solving modern scientific problems.

To summarize the data, each of the dataset contain one picture from the satellite, and each of these files contains several rows(pixels) each with 11 columns. For images 1, it has totally 115229 pixels with 17.76% of them are labeled as cloud, 43.77% are not cloud, 17.76% are shown unlabeled. For image 2, it has totally 115110 pixels with 34.11% of them are labeled as cloud, 37.25% are not cloud, 28.63% are shown unlabeled. For image 3, it has totally 115217 pixels with 18.43% of it are labeled as cloud, 29.29% are not cloud, 52.26% are shown unlabeled. Following we will show the labeled maps using x, y coordinates the expert labels with color of the region based on the expert labels.

Figure 1. Expert labels for image 1,2 and 3. White region represents high confidence cloudy; gray, high confidence clear; and black, unlabeled pixels.

From these three labeled maps, we can be noticed that there exist pattern for each of the image. Generally speaking, it's clear to see that there exists apparent regionalization for the data, and this proved that the data is dependent because if it's not the data will distribute randomly instead of showing specific pattern. If we take a closer look at the map for image 1, we can be noticed that the black and gray region are distributed at the left lower conner and right upper corner, while white region are gathered at the center of the map. For image 2 and 3, the black and grey region data are distributed at the left of the map while the white region data are distributed at the right. With all that being said, iid assumption for the samples cannot be justified for this dataset.

For the following analysis we tend to focus on performing a visual and quantitative EDA of the dataset. First we merged the 3 image datasets into one joint dataset and then plot the pairwise relationship among the features themselves, the following bullet points are the correlation that we can see from the Figure 2:

- y_cor and x_cor are positively correlated to features DF,CF,BF,AF,AN, while are negative correlated to the features label, NDAL, SD, CORR.
- Label and CORR are positively correlated to features label, NDAI, SD, CORR, DF, while are negative correlated to the features y_cor, x_cor, CF,BF,AF,AN.
- NDAI and SD are positively correlated to features label,NDAI,SD,CORR, while are negative correlated to the features DF,CF,BF,AF,AN
- DF are positively correlated to features y_cor, x_cor, label,CORR,DF,CF,BF,AF,AN, while are negative correlated to the features SD, CORR
- CF, BF, AF, AN are positively correlated to features x_cor, y_cor, DF, CF, BF, AF, AN, while are negative correlated to the features label, NDAR, SD, CORR.

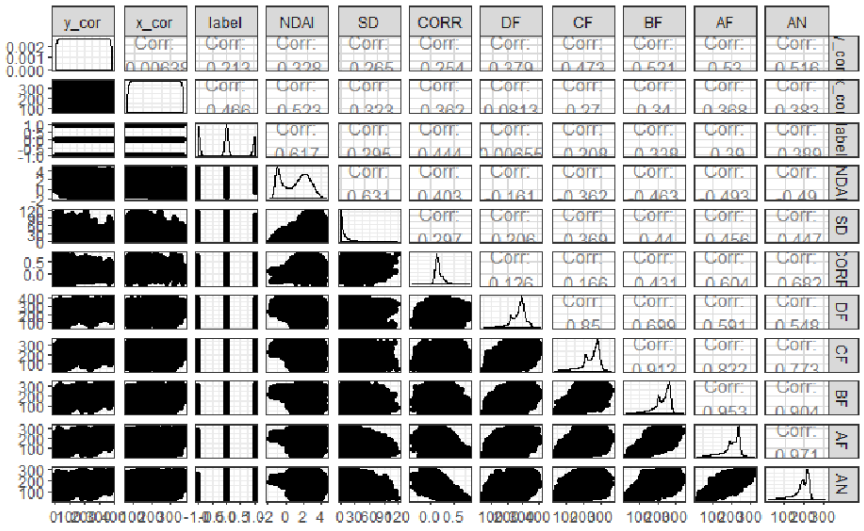
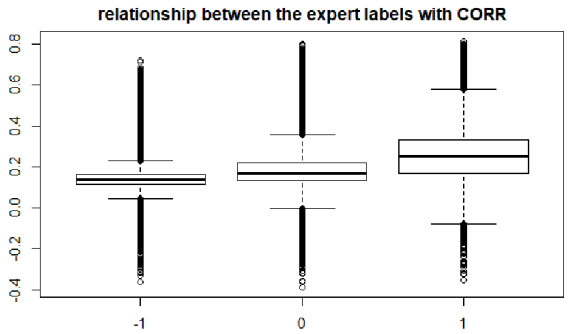
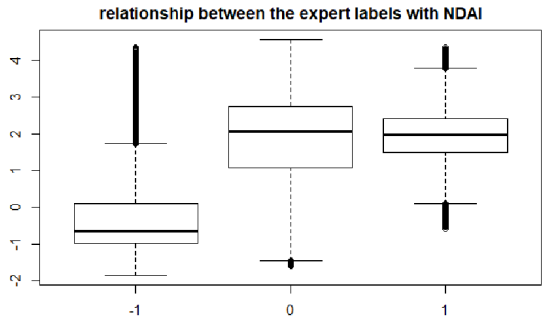
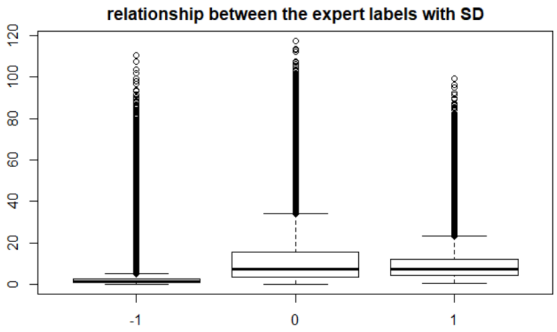


Figure 2: Pairwise correlation among all feature for the dataset

After we figured out the pairwise relationship among all the features, next we will continually focus on looking for the relationship between the expert labels with the individual features. If we look at the relationship between the expert labels with CORR, we can be noticed that there exists a slightly positive trend as expert labels goes from no cloud (-1) to cloud (1). And no cloud label (-1) will reflect a slightly lower CORR value compare to cloud label(1). For the relationship between expert labels with NDAI, label no cloud (-1) will reflect negative NDAI value while cloud label (1) will reflect a positive NDAI value. For the relationship between the expert labels with SD, it shown a slightly positive trend as expert labels goes from -1 to 1, and as for no cloud label, the SD value has strong similarity compare to cloud label since the range of its box plot is really narrow.







Preparation:

(a) As we already discussed above, we know that the data is dependent and by that means we cannot just sampling out the training data from the merged dataset because if we do so, we cannot really get a valid and representative data. Instead, we need to split the data based on the structure of the dataset.

So after we run few R code to check and we figured out the dataset has X_cor range from 65 to 369 and y_cor range from 2 to 383, which mean x_cor has a length of 304 and y_cor has a length of 381. Since there exists regionalization for our data, to sample the data in a more accurate way, we decided first to split the data into little blocks and then sample our training, validate and test data from each of them. Eventually we set the size of the blocks to become 4*3 because 4 can be divided by 304 and 3 can be divided by 381, so that we can make the best use of the whole dataset and will not wasting any data.

Next, for each block, we sample 20% of data to be the test data. For the rest of the 80% data, we set the 20% as the validation data and 80% to become the training data. And all of these steps speaks our first method to split the data.

The second method that we used to split the data is we will split the data based on the expert labels. With that being said, we totally have three groups of data and each of them contains either no cloud (-1), unlabeled (0) or cloud (+1) data. After we set each set of data as dataframe, for each dataframe, we will use the same sampling proportion as our method 1 did (sample 20% of data to be the test data. For the rest of the 80% data, we set the 20% as the validation data and 80% to become the training data). The reason why we are doing this is we want to make sure to get the same proportion for each labels as the original dataset

. (b) For this part we will use a naive way to look at our data, by that meaning that we will not take the dependency of the data into account for the following prediction. First we decided to use the merged data and converted it into a dataframe, then we sampled 20% of data to be the test data. For the rest of the 80% data, we set the 20% as the validation data and 80% to become the training data. After that, we set all the labels to no cloud labels (-1) and used the QDA classifier to make prediction for our validation and test set. And we eventually get the accuracy of 50.51% for the validation set and the accuracy of 50.72% for the test set. Both of the prediction reflect a relatively low accuracy, and the main reason causing this is because we didn't consider the dependency of our merged data.

(c) For this part we will apply lasso regression to select our best features. And the three features that we get are NDAI, SD, and CORR. And Figure 3 below is the lasso variables trace

plots(Figure 3), and we can be noticed that all traces converge to 0. And the quantitative table(Figure 4) shown that only NDAI， SD， CORR coefficients estimates are non-zero.

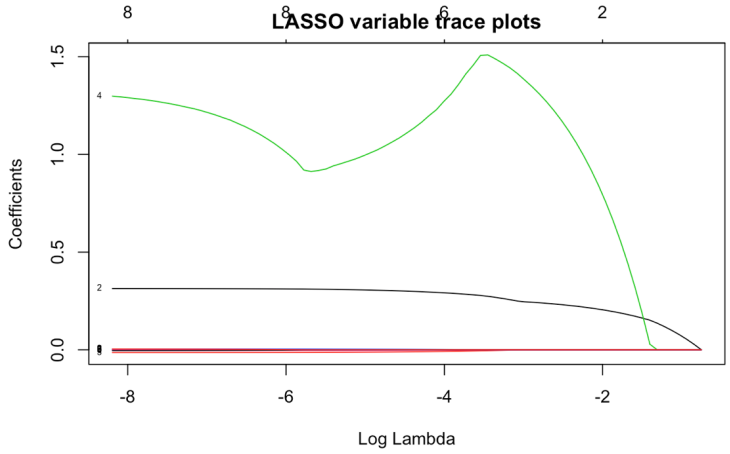


Figure 3

(Intercept)	(Intercept)	NDAI	SD	CORR	DF	CF
-0.6636494910	0.0000000000	0.2496618312	-0.0004494802	1.4140851405	0.0000000000	0.0000000000
BF	AF	AN				
0.0000000000	0.0000000000	0.0000000000				

Figure 4

Modeling:

(a): For this part we will need to try several classification methods and assess their fit using cross-validation. For CV we don't have the validation set, so we need to merge our training and validation set to fit our CV model. For calculating the accuracy across folds and test set, we will be applying four classification methods on two different data splitting method as we mentioned in Preparation part (a) above, which are QDA, LDA, Naive Bayes, neural network,and logistic regression classifier. And the below Figure 5 detailedly demonstrated the accuracy across folds, CV accuracy, test accuracy and loss. Neural network classifier works best among all classifiers we use, which gives 0.90 test accuracies for both data sets. It makes sense because we don't assume any underlying pattern for the data when we use neural network. However, there are assumptions for other classifiers. For example, the covariance of each of the classes is identical for QDA; the covariance of each of the classes is same for LDA; little or no multicollinearity among the independent variables for logistic regression; independence assumption for naive bayes.

Split Method	Classifier	fold 1 accuracy	fold 2 Accuracy	fold 3 accuracy	fold 4 accuracy	CV Accuracy	Test Accuracy	loss
Block split	QDA	0.8991773	0.8960209	0.8983752	0.9005485	0.8985305	0.8964091	0.1014695
Block split	LDA	0.8993842	0.8994619	0.901247	0.9012729	0.9003415	0.8982363	0.09965849
Block split	Naïve Bayes	0.891157	0.8903808	0.8903808	0.8872245	0.8897858	0.8878311	0.1102142
Block split	Logistic	0.8755045	0.8777295	0.8763841	0.8787644	0.8770956	0.8750986	0.1229044
Label Split	QDA	0.8961863	0.8991637	0.8960396	0.8976257	0.8972538	0.8952946	0.1027462
Label Split	LDA	0.8985413	0.8985629	0.8992598	0.8952946	0.8979147	0.8962799	0.1020853
Label Split	Naïve Bayes	0.8855886	0.8903441	0.8886139	0.8863309	0.8877194	0.8865952	0.1122806
Label Split	Logistic	0.8724437	0.8730895	0.8767423	0.8740988	0.8740936	0.875012	0.1259064
Block split	neural network	0.9100179	0.910742	0.9100435	0.912165	0.910742	0.9093339	0.08925799
Label Split	neural network	0.9078894	0.9101221	0.9075747	0.910843	0.9091073	0.9076709	0.0908927

Figure 5

(b) In this part, we plot the ROC curve for two data splitting methods. Classifiers that give curves closer to the top-left corner indicate a better performance. To compare different classifiers, it can be useful to summarize the performance of each classifier. One common approach is to calculate the area under the ROC curve, which is abbreviated to AUC. A classifier with high AUC (Area Under Curve) can usually score better in a specific region than another classifier with lower AUC. In Figure 6, the data split method we use is the proportion of each label. In Figure 7, the data split method we use is dividing blocks to sample. The AUC for logistic regression is the smallest among other AUCs. The naive bayes classifier works better in Figure 6. Overall, neural network classifier gives good performance in both figures.

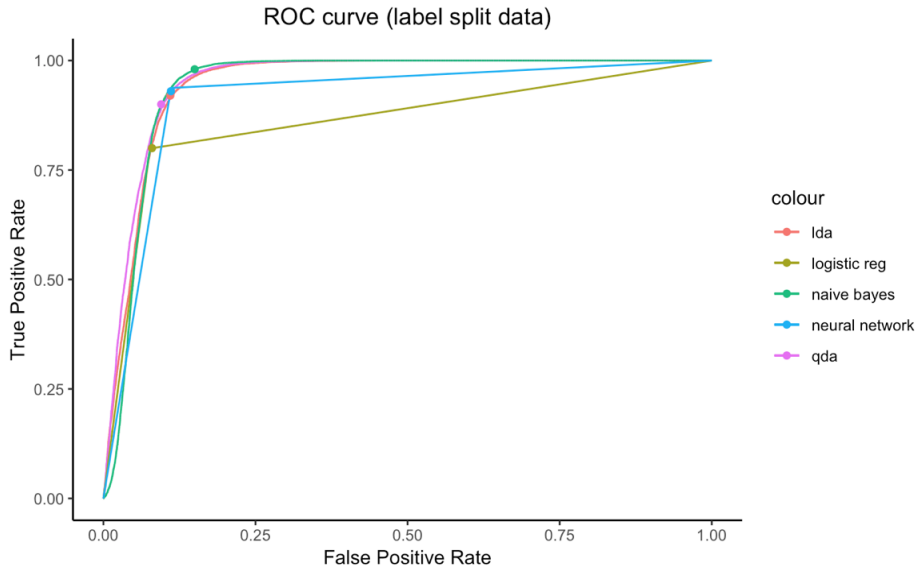


Figure 6

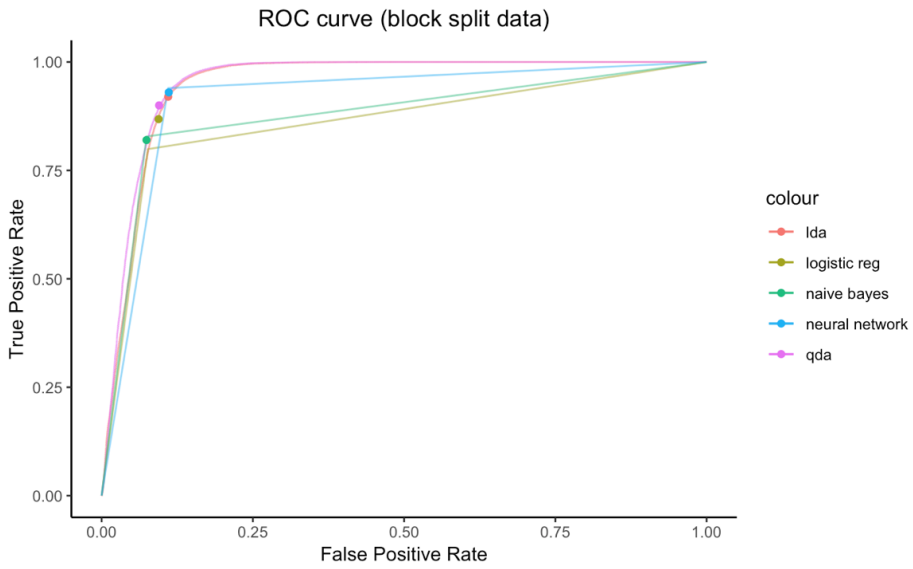


Figure 7

(c)

(bonus) We add a naive bayes classifier other than 4 classifiers mentioned above.

Diagnostics:

(a) For this part we want to figure out our model is stable or not for us to accurately do the classification. In order to approach this, we sampled our training data from our dataset with various types of percentage (0.1, 0.2, 0.4, 0.6, 0.8,1), and we plotted the loss relates to each percentage. As we can clearly see from Figure 8, the loss doesn't change much (loss range between 0.09090 and 0.09060) as the training data percentage goes from 0.1 to 1, and that provided us with a strong confidence to believe our classification model is relatively stable enough for us to do further prediction.

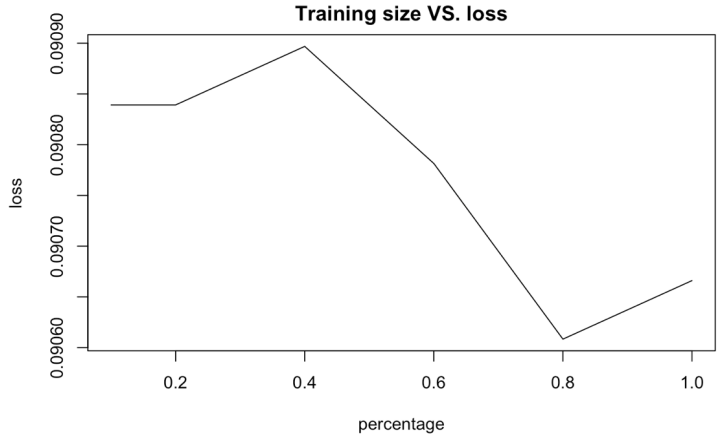


Figure 8

Now we sample the training data with various percentage as I mentioned above (0.1, 0.2, 0.4, 0.6, 0.8,1), and we again plot the relationship between each percentage that we sample our training data and the weight. From the figure 9 that I shown below, we can be noticed that the weight almost remain constant as we choose different training size (percentage goes from 0.1 to 1), and that implicitly proved that this model is stable because the weights is the model parameter estimates.

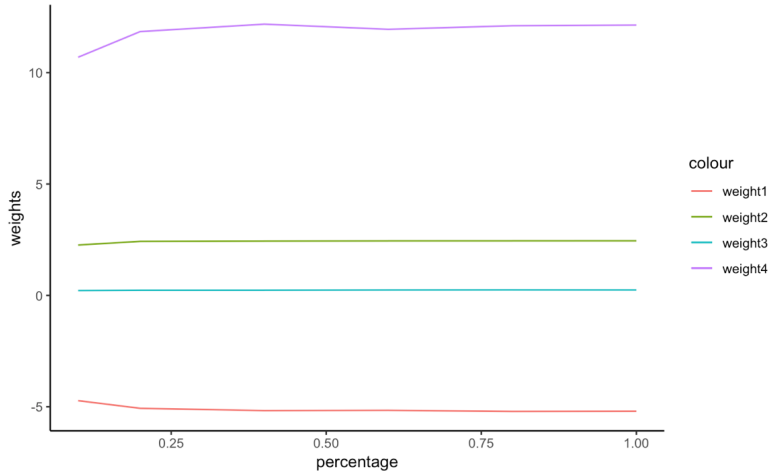


Figure 9

(b) Now we already get our best classification model, to see whether there exists any patterns in the misclassification error, we plot a scatter plot and we want to see the pattern from it. So what should we consider when plotting this scatterplot? We are now have the label and the prediction result, and if create a new variable "hit". If the prediction we got match the label itself, we will label it in red color to show in the scatterplot. On the other hand, if the label and the prediction result not match, that we label it in blue. And with this being said, the figure 10 below is the scatter plot to show the correctness of our prediction. And we can clearly see that there exist an obvious cyclical area that only contains wrong predictions,it may cause by the high NDAI values. The percentage of the correctness is 0.93, which is high enough to give strong predictions.

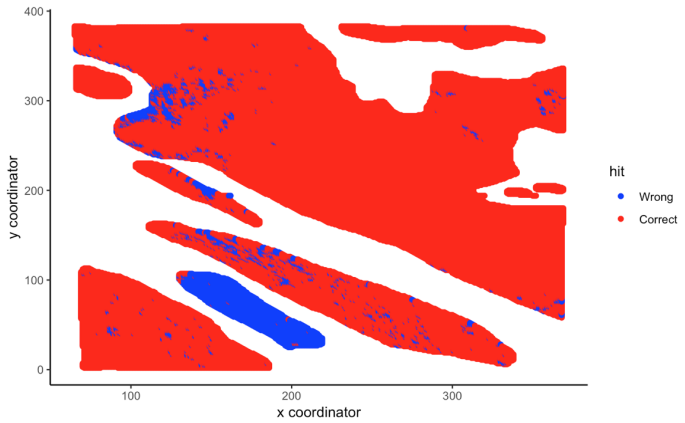


Figure 10

```
  correct     wrong
1 0.9350931 0.06490686
```

Figure 11

(c) Since there is a cyclical area with wrong prediction, we can try to increase the hidden layer to see if the correct prediction point will cover the cyclical area. Usually, the number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer. Since it take too long to run the model with 3 hidden layer, we can try it in Python in the future. For the future inputs, we expect it takes shorter run time in python and will give strong predictions.

(d) As we mentioned before the way we split the data will not have much differences on folds accuracy, test accuracy and CV accuracy, and that mean our models generally did not have large differences resulting from how we split the data, so the results in part 4(a) and 4(b) will not change as we modify the way splitting the data, and that might be tribute to how the data is not so skewed as we would believe.

(e) Conclusion: Through the whole process analyzing and studying the image data, we eventually come up with our finalize found that the neural network method can achieve a higher accuracy rate when we test out the model performance across various model classifiers. This mostly because we broke down all the datasets into 3x4 blocks and split them without causing too much bias for our further analysis. We found that the neural network classifier gives the best performance in terms of accuracy in labeling comparing to LDA, QDA, logistic regression and naive bayes classifiers. Also, satisfying the assumption is really important when choosing data. The neural network classifier is great because we don't need to assume any underlying pattern for the data.We also believe that the random forest model classifier may perform higher accuracy compared to the other model classifiers but the runtime is too long in R. Due to the issue of computational complexity, we think that neural network is more practical and efficient than others.

Github Link: https://github.com/xiaowanzio8/Stat-154