Stat-154 Project 2: Cloud Data

Cheng Lin, Junyan Tan
3033412059, 3033370095

Part I: Data Collection and Exploration (30 pts)

The purpose of the study

The research paper, “Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies”, studied and researched on the climate changes in the Arctic by a systematical study of accurate Arctic-wide measurements such as cloud coverage. Thus, the researchers focused on performing accurate cloud detection because this technique can assist in predicting the dependence of the surface air temperatures on increasing atmospheric carbon dioxide levels. This research paper proposed the enhanced linear correlation matching (ELCM) algorithm and the Fisher’s quadratic discriminant analysis (ELCM–QDA) algorithm using the Multiangle Imaging SpectroRadiometer (MISR) imagery to optimize the operational process to perform accurate cloud detection and identify cloud-free surface pixels in the imagery.

The dataset & The methodologies of data collection

The MISR data set was the primary source for data analysis and obtained at the courtesy of the NASA Langley Research Center Atmospheric Sciences Data Center. The data used in this research paper were collected from 10 MISR orbits of path 26 over the Arctic, northern Greenland, and the Baffin Bay. The 10 orbits span approximately 144 days from April 28 through September 19, 2002, including rich surface features, such as permanent sea ice in the Arctic Ocean, snow-covered and snow-free coastal mountains in Greenland, permanent glacial snow and ice, and sea ice that melted across the Baffin Bay. The MISR sensors have nice cameras: 70.5◦ (Df), 60.0◦ (Cf), 45.6◦ (Bf), and 26.1◦ (Af) in the forward direction; 0.0◦ (An) in the nadir direction and 26.1◦ (Aa), 45.6◦ (Ba), 60.0◦ (Ca), and 70.5◦ (Da) in the aft direction. The seven MISR cameras can cover an approximate 360-km-wide swath on the Earth surface that extends across the daylight side of the Earth from the Arctic down to Antarctica in about 45 minutes. MISR collects data from 233 geographically distinct paths on a repeat cycle of 16 days.

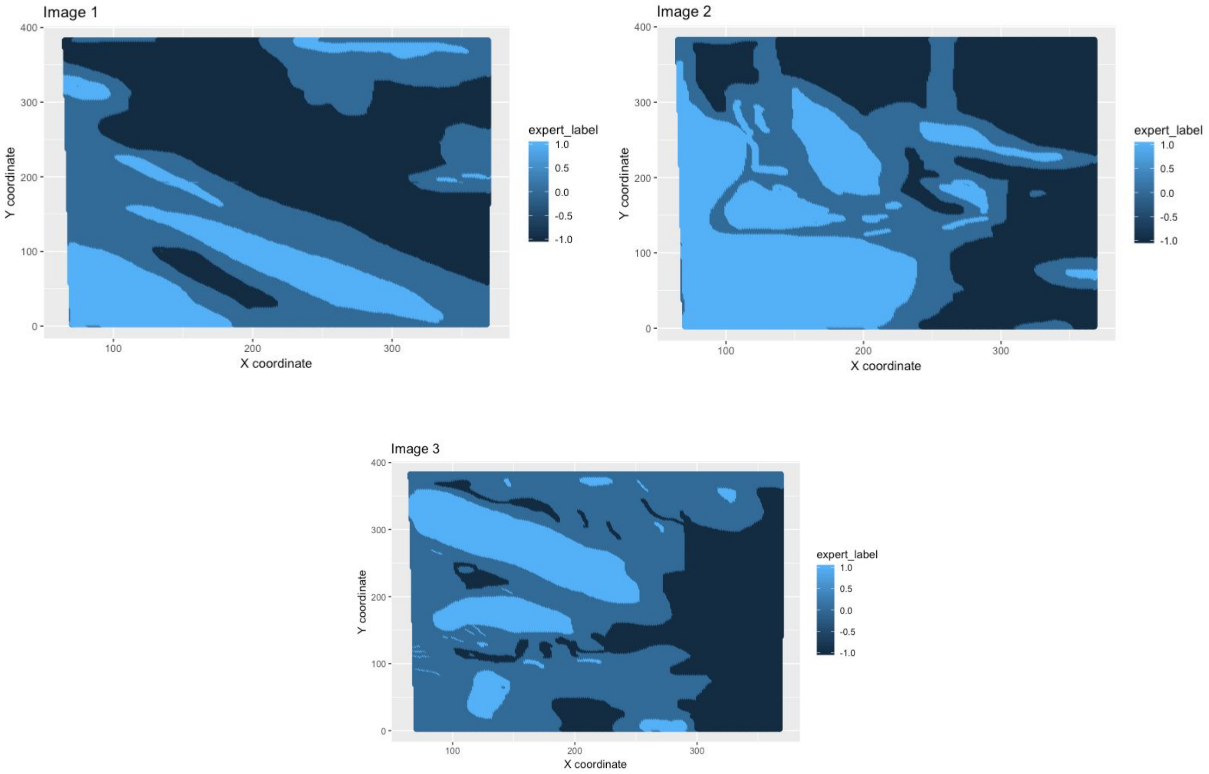
The main conclusion and Impact

This research paper showed that three physical features, the linear correlation of radiation measurements from different MISR view directions (CORR), the standard deviation of MISR nadir red radiation measurements within a small region (SDAn), and a normalized difference angular index (NDAI) can significantly increase the accuracy the ELCM algorithm hence it would provide better spatial coverage than the MISR operation algorithm for cloud detection in the Arctic. There are two major impacts of this research. First, there are many applications that require appropriate statistical methods to solve weather and climate problems. In order to achieve great outcomes from different studies, statisticians should work closely with atmospheric scientists and the MISR science and instrument teams at the Jet Propulsion Laboratory in order to determine the most appropriate and optimized statistical methods for analyzing Earth science data. Second, the power of statistical thinking is the combination of statistical principles and application-specific knowledge. A strong power of statistical thinking is fundamental to solve any scientific issues of a specific field.

Summary of the Dataset

All of the three images contain a total of 345556 observations and 11 features named as y coordinate, x coordinate, expert label, NDAI, SD, CORR, Radiance angle DF, Radiance angle CF, Radiance angle BF, Radiance angle AF, and Radiance angle AN.

For the expert labels, the image 1, 2, and 3 have correspondingly 43.78%, 37.25%, 29.29% observations that are labeled as **no cloud**, 17.77%, 34.11%, 18.44% observations that are labeled as **cloud**, and 38.46%, 28.64%, 52.27% observations that are **unlabeled**. As we can tell from the figure 1.b, cloud or not cloud, or unlabeled, the cloud and not cloud areas appear to have region concentrated together which implies there is a correlation among neighbor pixels for each expert label. In fact, it actually makes sense because the cloud area is usually gathered and spread across many pixels considering the size of the cloud. In conclusion, the assumption of i.i.d is invalid.



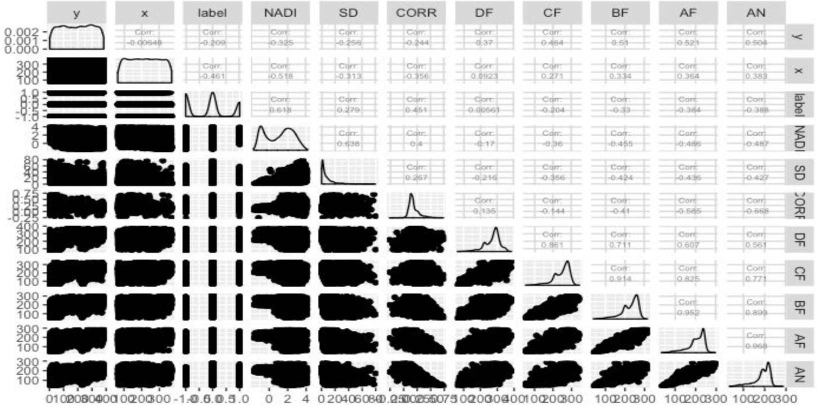
(Figure 1.b)

Visual and Quantitative EDA

Pair-wise relationship

In order to better visualize the relationships across variables, we used the pair-wise plotting method to show the correlations in order to identify the relationships. Based on Figure 1.c.1, we can see that features of radiance angle have significantly stronger correlations with other types of radiance angle than the rest of variables. The details as follows:

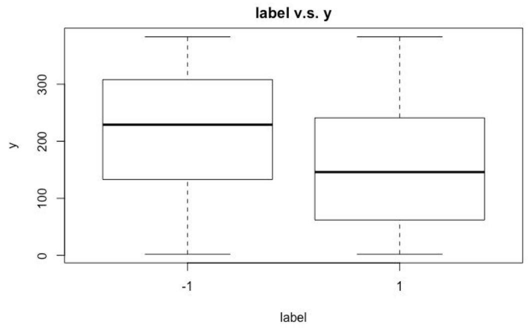
- “y” is more correlated to “NADI”, “CF”, “BF”, “AF” and “AN” than to “x”, “label”, ”SD”, and “CORR”.
- “x” is significantly more correlated to “label”, “NADI”, “SD”, “CORR”, “BF”, “AF”, “AN” and “CF” than to “DF”.
- “label” is significantly more correlated to “NADI”, “SD”, “CORR”, “CF”, “BF, “AF” and “AF” than to “DF”.
- “NADI” is significantly more correlated to “SD”, “CORR”, “CF”, “BF”, “AF” and “AN” than to “DF”.
- “CORR” is significantly more correlated to “BF”, “AF” and “AN” than to “DF” and “CF”.
- “DF” is more correlated to “CF”, “BF” and “AF” than to “AN”.

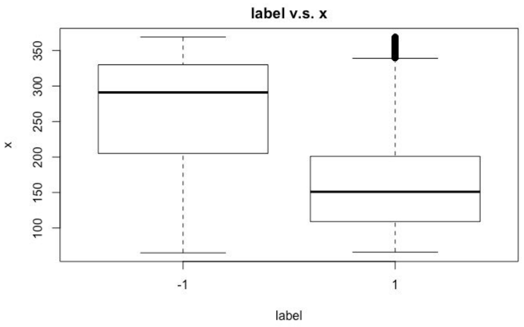


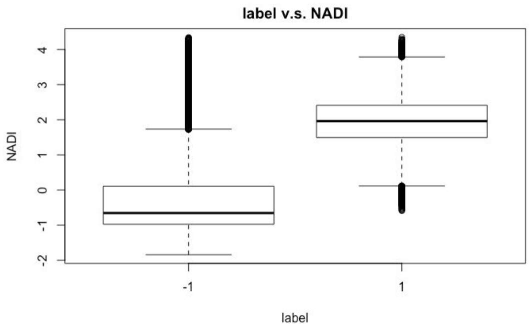
(Figure 1.c.1)

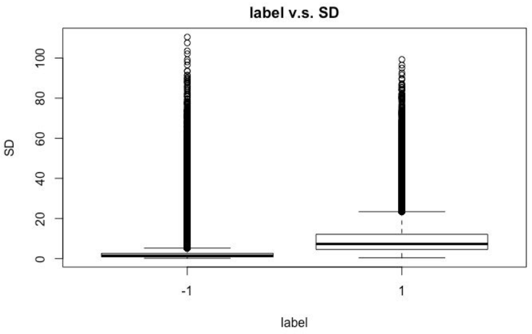
Relationships between the expert labels with the individual features

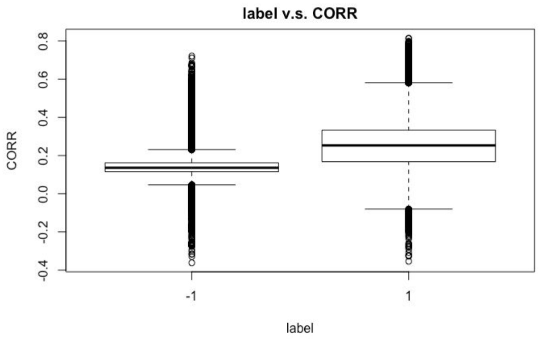
Since the labels are discrete and all other variables are continuous, we used boxplot to find out patterns and trends between them. We selected the labels that are either -1 or 1 because we want to examine the cloud and not-cloud data. Based on Figure 1.c.2, “y”, “x”, “CF”, “BF”, “AF”, and “AN” have larger values of the median in non-cloud label than cloud label. “NADI”, “SD”, and “CORR” have smaller values of the median in non-cloud label than cloud label. “DF” has similar values of the median in non-cloud label and cloud label. We also noticed that “NADI”, “SD”, “CORR”, “SD”, “CORR”, “DF”, “CF”, “BF”, “AF”, and “AN” produced significant amounts of outliers in the non-cloud label comparing to the outliers in the cloud label for these features.

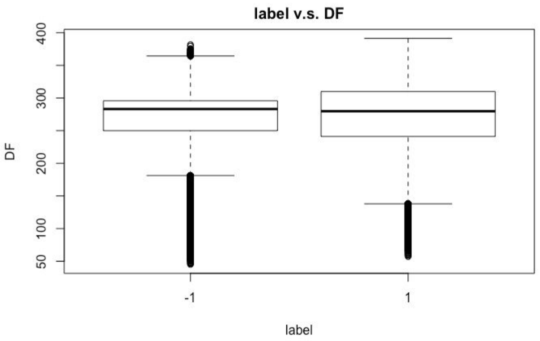

label v.s. y
y
0
100
200
300
-1
1
label

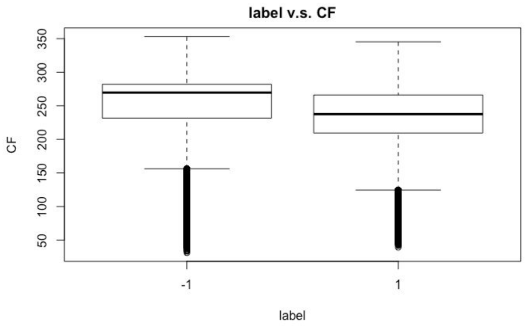

label v.s. x
x
100
150
200
250
300
350
-1
1
label

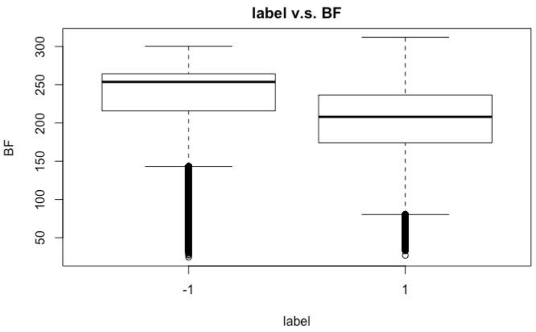

label v.s. NADI
NADI
-2
-1
0
1
2
3
4
-1
1
label

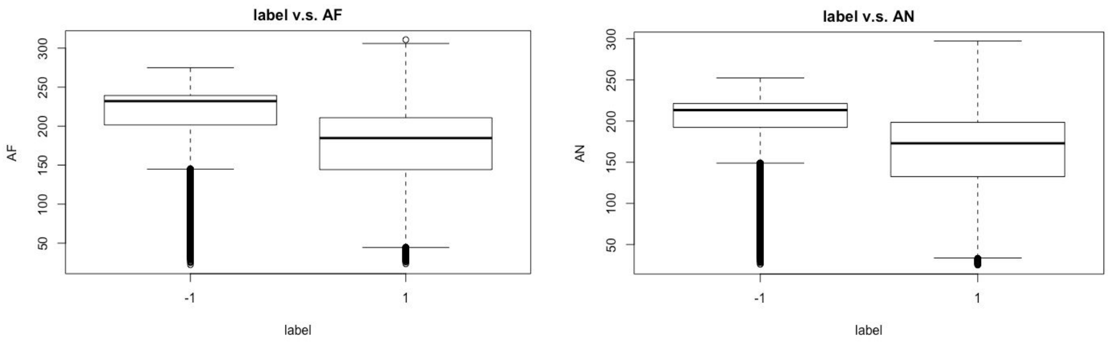

label v.s. SD
SD
0
20
40
60
80
100
-1
1
label


label v.s. CORR
CORR
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
-1
1
label


label v.s. DF
DF
50
100
200
300
400
-1
1
label


label v.s. CF
CF
50
100
150
200
250
300
350
-1
1
label


label v.s. BF
BF
50
100
150
200
250
300
-1
1
label



(Figure 1.c.2)

Part II: Preparation (40 pts)

Data Split

The first method (2-stage cluster), we want to split our data into blocks (each block has a size of 4 x 3) and do a simple random sampling in the block level. We divided the interval of the x-axis (65, 369) and y-axis (2, 383) into 3 by 4 blocks because we want to get the integer-number of blocks. Then, we carefully inspected how the labels distributed within each block and perform a proportional sampling in each block. This method is more accurate because we were evenly splitting the data at the block level in order to lower the bias in the training set.

The second method, we want to split our data set into a training set, a validation set as well as a test set with proportional to the size of expert labels in the merged image(ie. Image 1, 2, 3). Hence, our data splitting method can solve the problems of over-represent or under-represent the cloud, no cloud, the unlabeled region data. In other words, the use of proportional sizes can calibrate this bias. Even though this is not a perfect way to split the data, at least, such as, it avoids the situation that we bias select data as training data.

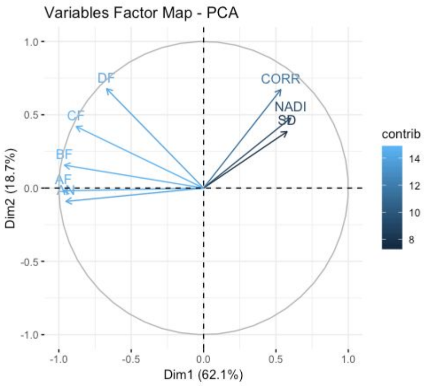
Overall, we split the data into 64% (training), 16% (validation), and 20% (test) for both methods for analysis.

The accuracy of a trivial classifier

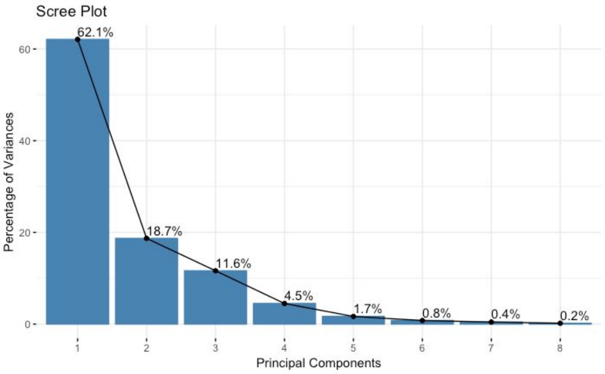
We have split the data randomly and set all the labels as -1 on the validation sets and test sets for the trivial classifier. This trivial classifier gave the accuracy of around 50%. Hence, it’s not a ‘good’ classifier in terms of accuracy. In the extreme setting, the accuracy will be quite high if most of the test or validation data consist of label -1.

PCA analysis

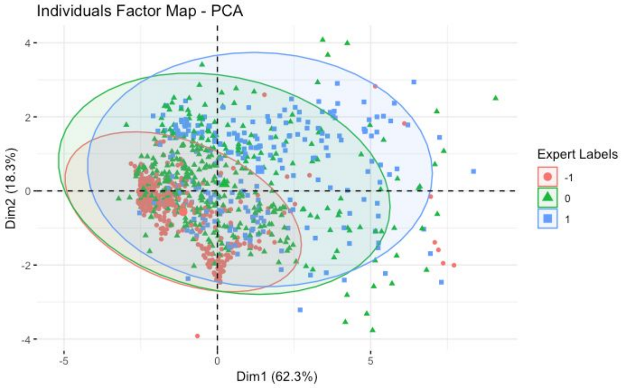
The PCA will be a good tool to select the best features. After performing the PCA analysis (Figure 2.c.2), we have realized there is a feature with a 62% explanation rate, a feature with an 18.7% explanation rate, and a feature with an 11.6% explanation rate which considered the most relevant features for the model. According to Figure 2.c.1, we also found that "SD", "NDAI", and "CORR" are the most relevant features which contribute the most to the model. Moreover, the best 3 features add up to a total of 92.3% rate of explanation rate which is definitely enough for being "best 3 features".



(Figure 2.c.1)



(Figure 2.c.2)



(Figure 2.c.3)

Part III: Modeling (40 pts)

Model Classifier analysis

The assumptions for each classification method:

1. **Logistic Regression**
   a. Intending to solve two-class or binary classification problems.
   b. Two classes should not be well-separated since LR will become unstable.
   c. Requiring a large amount of data to be trained.
   d. Requiring a linear relationship between the response variable and predictor variables.
   e. No multicollinearity among predictor variables.
2. **LDA**

    a. Intending to solve two-class and multi-class classification problems.
    b. Requiring the data is Gaussian distributed.
    c. Each feature has the same covariance.
    d. Requiring a linear combination of predictor variables.

3. **QDA**
    a. Intending to solve two-class and multi-class classification problems.
    b. Requiring the data is Gaussian distributed.
    c. Each feature doesn’t require to have the same covariance.
    d. Requiring a non-linear combination of predictor variables.

4. **Random Forest:** Constant covariance.

5. **SVM & 6. KNN**
    a. Both methods are non-parametric methods which imply that there are no assumptions when implementing either method.

The below tables for accuracy rate:

(Note: Random Forest has a very bad accuracy due to computational complexity. Only trying 1000 rows of data for random forest classification.)

Accuracy Rate for Method 1

Classifier	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Test
QDA	0.9000750	0.8973627	0.8964970	0.8933229	0.8968403	0.8955782
LDA	0.8975935	0.8973049	0.8976771	0.8968144	0.8982860	0.8967569
Logistic	0.8939000	0.8943040	0.8898572	0.8911877	0.8935538	0.8919556
Naive Bayes	0.7881002	0.7867119	0.7909568	0.7845799	0.7851570	0.787643
KNN	0.9012412	0.9009485	0.9008593	0.910005	0.9098782	0.9042608

(Table 3.a.1)

Accuracy Rate for Method 2

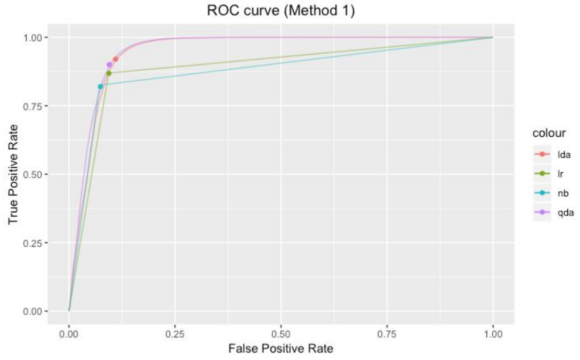
Classifier	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Test
QDA	0.8956358	0.8975347	0.8958921	0.8957860	0.8976097	0.8973133

LDA	0.8973726	0.8969339	0.8975239	0.8966335	0.8967622	0.8956311
Logistic	0.8935974	0.8907759	0.8914196	0.8923851	0.8942829	0.890248
Naive Bayes	0.7880445	0.7869296	0.7875338	0.7883749	0.7872634	0.7871479
KNN	0.9014325	0.9012355	0.9120345	0.9000234	0.9012934	0.9014948

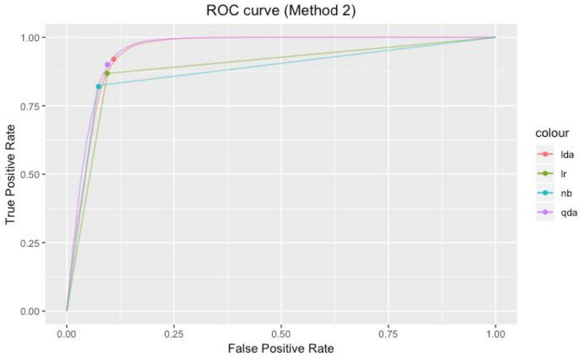
(Table 3.a.2)

Before we start the model analysis, we've filtered out the 0 expert labels since we want to perform logistic regression (binary classification) on these data. Moreover, this method can help us to keep consistency across all the datasets which allow us to compare the test accuracies among different classifiers given the same split data. Second, it allows comparing the accuracy among different data splitting methods given the same classifier. We've decided to test with the expert label of 1 and -1 with five-fold cross-validation. According to Table 3.a.1, and Table 3.a.2, QDA, LDA, and Logistic regression appear to have similar accuracies. In fact, the outcome makes sense since assumptions hold for these classifiers given with a large amount of data. Also, the test data confirmed the result of cross-validation as well. But, the random forest gave us a bad result on the contrary because the datasets violated the assumption of being i.i.d, the quantity of data input is too few, and computational complexity is too large.

ROC analysis



(Figure 3.b.1 (Method 1))



(Figure 3.b.2 (Method 2))

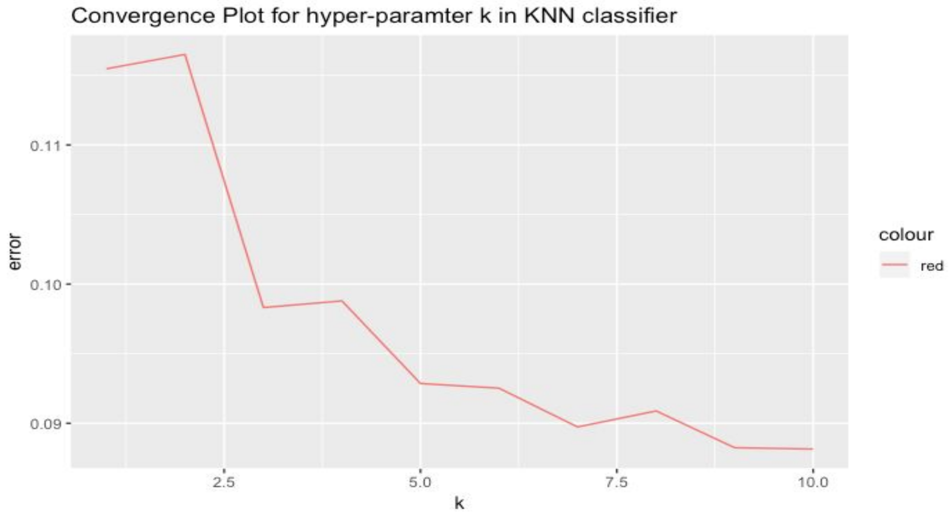
To compare the performance of different classification methods by using ROC curves in figures (3.b.1 and 3.b.2) below as a visual aid. It is a plot of the true positive rate (TPR) against the false positive rate (FPR) for the different possible cut-points of a diagnostic test. The basic rule of determining the best performance is the closer the curve to the left-hand border and the top border of the ROC space, the more accurate the test. Hence, the Figure 3.b.1 and the Figure

3.b.2 clearly showed that using both methods of splitting data give the very similar ROC curves, and the best performance are from qda for method 1 as shown in figure 3.b.1 and as well as for method 2 as shown in figure 3.b.2.

Part IV: Diagnostics (50 pts)

In-depth analysis of Model Classifier

We've decided to perform a k-fold cross-validation with the use of KNN classifier, and it gave the highest accuracy rate compared to all the other classifiers but not the random forest. Since computational complexity issues. After considering the computational cost of the random forest classifier, we think KNN is the best classifier comparing to others. For KNN, we figured out the specific value of k which can minimize the error rate of the model. According to Figure 4.a.1, convergence plot k against the error rate(roughly 8%) can give us the best choice when k = 10.



(Figure 4.a.1)

Table of C.V. accuracy for KNN

KNN	0.9012412	0.9009485	0.9008593	0.910005	0.9098782	0.9042608

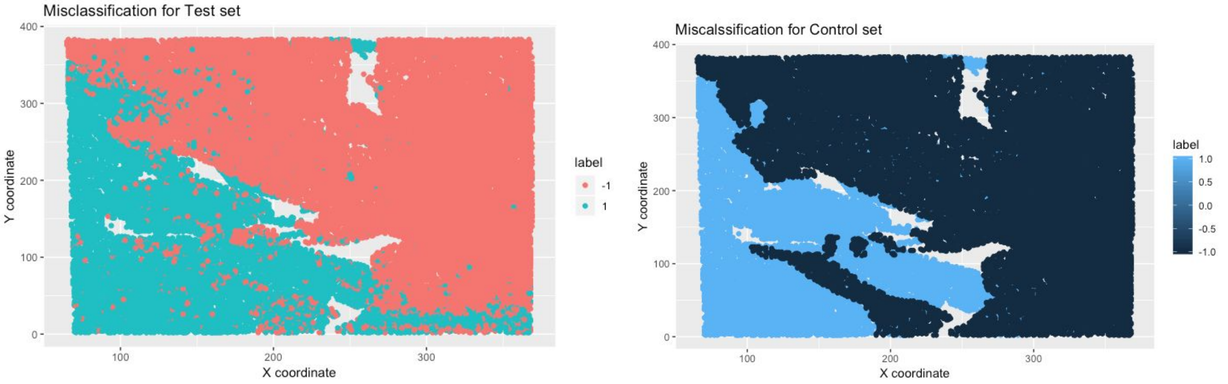
(Table 4.a.2)

In-depth analysis of KNN Model Classifier

In order to find out the misclassification of this particular classifier, we plotted the x-y coordinates with color differentiating expert label v.s. the predicted label.

There are misclassifications majorly occur under the region y = - x and around the geographical region bounded by (x-axis) [100, 400] x [0, 100](y-axis). Also, there are many 'small' region of misclassification spread evenly under the line y = -x. (Mostly are a high value of NADI data)

Left side: Prediction label Right side: Expert label



(Figure 4.b.1)

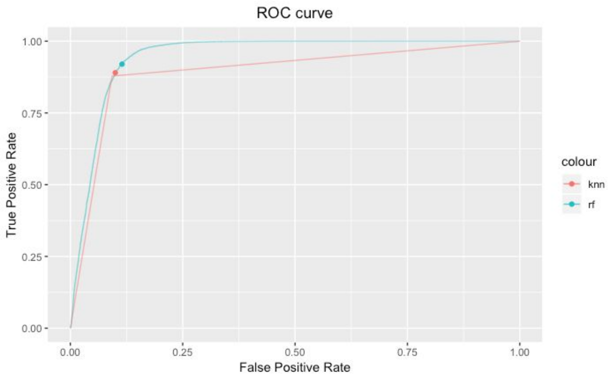
Better Classifier

The better classifier is Random Forest since this is unsupervised learning(i.e there is no such assumptions for the data set). We have performed the cross-validation on training set as well as test set, it turned out that random forest has the highest accuracy among all other classifiers.

Random Forest	0.9382465	0.9386955	0.9390455	0.9391235	0.9396452	0.9401465

(Accuracy Table 4.c.1 for Random Forest)

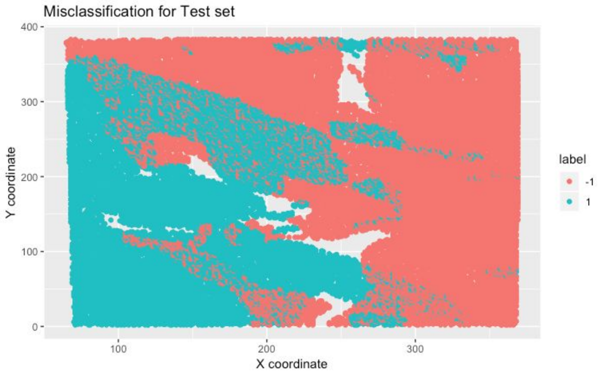
As well as the **Random Forest's ROC curve** below which has the biggest area under it's curve and has the closest curve to the (1,1). This gives us the confidence of believing better performance in terms of accuracy rate of classification for future data without the expert label.

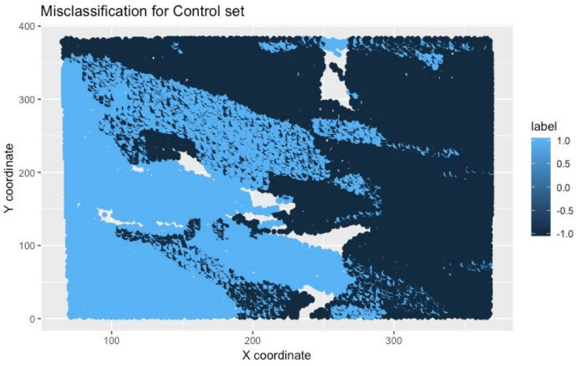


(Figure 4.c.2)

Changes in the data splitting

It does change in terms of accuracy when using the different splitting method. It appears to have a reasonably high accuracy which implies the method of splitting data is as important as which classifier we choose to perform classification over a dataset. However, it does not change the result from the previous question when using the random forest classification method.





Conclusion

After careful studying and analyzing the image data, we found that the first method can achieve a higher accuracy rate when testing out the model performance across different model classifiers. This happens because we examine through all the datasets into 3x4 blocks and split them without causing too much bias on the datasets. Under the KNN classifier, we found the best performance in terms of accuracy in labeling comparing to LDA, QDA, logistic regression model classifiers. In addition, we should be very careful about the assumptions when using the model classifiers. We also believe that the random forest model classifier should have higher accuracy than the other model classifiers that were being used. In practical, we tried to use Python to run random forest in the dataset and the result turned out to be the most optimized method. Due to the issue of computational complexity, we think that KNN is more practical and efficient than random forest.

Part V: Reproducibility (10pts)

Github: https://github.com/tanlibra18/Stat-154/blob/master/README.md

Acknowledgment

Cheng Lin and Junyan Tan completed all the parts of this machine learning project together.