Cloud Data Prediction

STA521, 2021 Fall

Elena Wang elena.wang@duke.edu

11/13/2021

1. Data Collection and Exploration

1.1 Summary

With Multi Angle Imaging Spectro Radiometer (MISR) imagery, this study focuses on cloud detection in the Arctic since there is the strongest dependency of surface air temperature on increasing atmospheric carbon dioxide levels and cloud is important for modulating sensitivity of that. Based on the true classification from expert labels, our dataset is used to train the classification models from the three images provided to predict the presence of clouds from the absence of clouds. The used data was collated from 10 MISR orbits of path 26 between two consecutive orbits over the same path every 16 days from April 18 to September 19, 2002. Each path is divided into 180 blocks, and each MISR pixel covers a 275m * 275m region to produce huge amounts of data from global coverage. The variables are from MISR sensor comprising nine cameras, with each of them viewing Earth scenes at different angles in four spectral bands (blue, green, red, and near-infrared), and six zenith angles of the cameras are 70.5° (Df), 60.0° (Cf), 45.6° (Bf), 26.1° (Af) in the forward direction and 0.0° (An) in the nadir direction. However, only MISR red radiation data are used due to the similar reflectance signatures of all four bands and the highest spatial resolution reading. The significant three features of MISR algorithms collected are an average linear correlation of radiation measurements at different view angles (CORR), the standard deviation of MISR nadir camera pixel values across a scene (SD), and normalized difference angular index which is average radiation measurements (NDAI) to classify the cloud, for example, low correlations mean high clouds, larger NDAI presents the presence of clouds.

Overall, compared by the ELCM, ELCM-QDA, ASCM, SDCM, and SCM algorithms of cloud detection, ELSM-QDA has less error and performs well and ELCM based on 3 features is more accurate and provides better spatial coverage. This study is beyond traditional statistical methods and develops more powerful thinking of statistics in other fields. This research not only improved understanding of the flow of visible and infrared radiation through the atmosphere, but also enabled the community to study the relationship between cloud property and global warming.

1.2 Plots the Expertly Labelled Maps and Observation

Combined with all three image datasets, the proportions of clear labeled, unknown, and cloudy labels are 36.8%, 39.8%, and 23.4% respectively. It is worth mentioning that the cloudy data is labeled as 1, -1 for the clear data, and 0 for unknown. Figure 1 is the maps of well-labeled clouds with x and y coordinate and corresponding colors. It is not difficult to distinguish the labels from the plots, in which the clear area tends to be located on the right side of the image, and the cloudy area tends to be located on the left, while the unknown region is surrounded by the clear and cloudy region. Since the distribution of the labels have high correlation with those in the neighboring region, which means that the classifiers have the spatial dependency, they are not i.i.d assumptions for the samples justified for the dataset and we would use blocked cross-validation for further prediction.

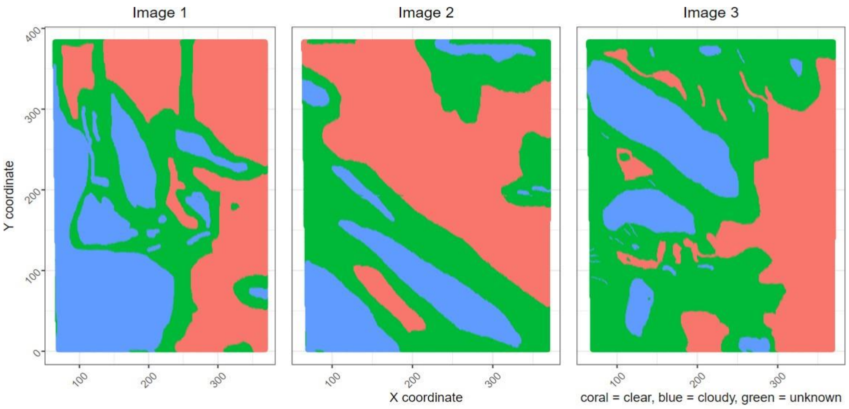


Figure 1: Images Based on Labels

1.3 Visual and Quantitative EDA

In Figure 2, the pairwise relationship between clear (-1) and cloudy (1) levels, we could see that the selected features (CORR, NDAI, and SD) separate the two classes well. Moreover, the rounded boundaries suggest that we could use non-linear models such as kernel SVM or QDA for future prediction.

The following density plots, Figure 3, of all features compare the distribution of clear and cloudy classes. It is obvious that the NDAI best separates the two classifiers in all three images, which means that the NDAI would be the significant variable to do the class prediction in our later models. SD has the worst separation between two variables within the lower SD. However, it still could separate well within the higher SD. Other features have a similar separation performance, which could separate in some certain ranges but worse than NDAI.

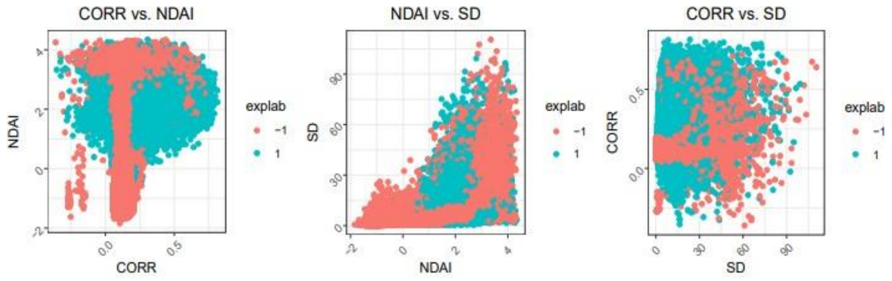


Figure 2: Pairwise Comparison

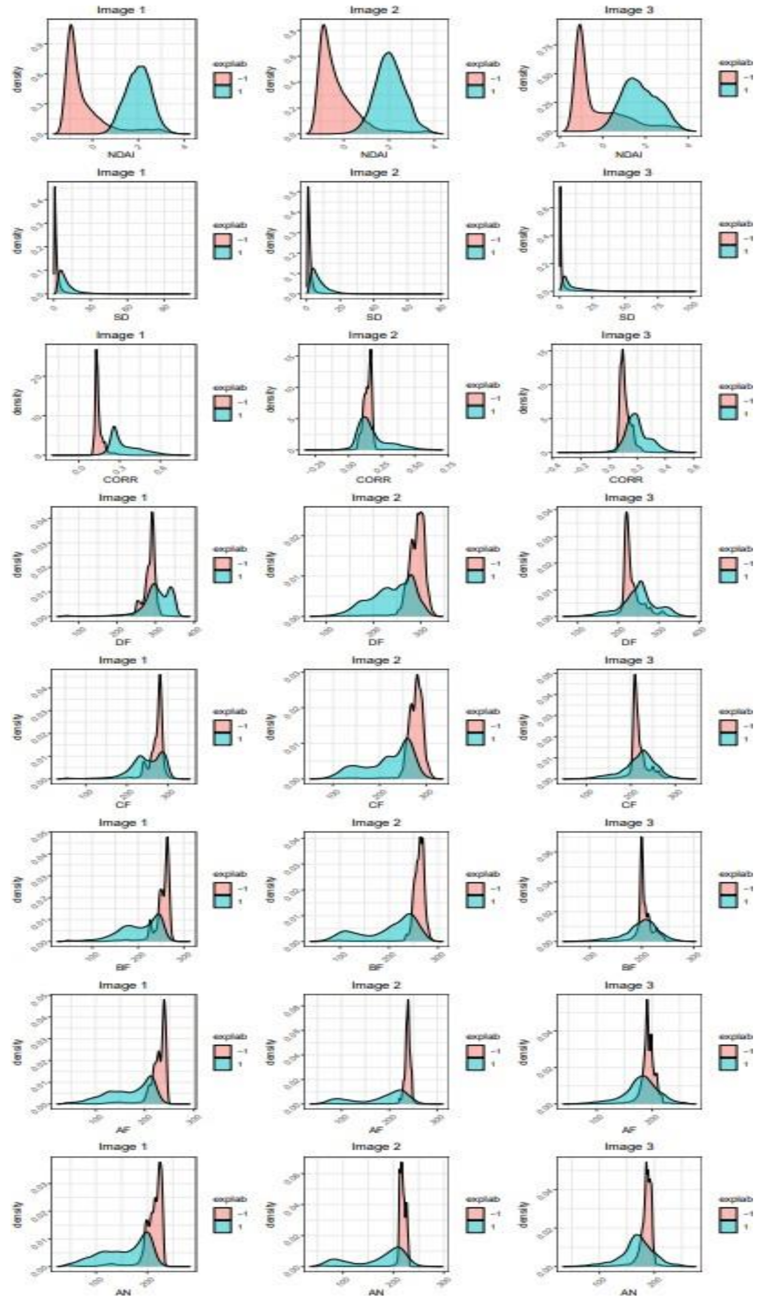


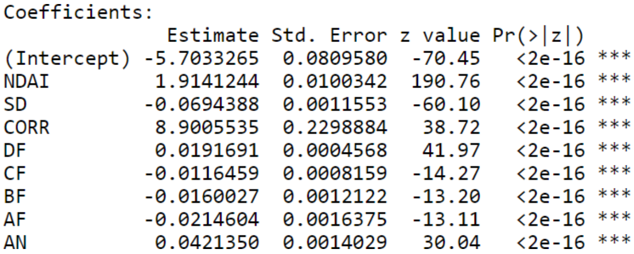
Figure 3: Density Plots for Features

2. Preparation

We should firstly split the dataset into training, validation, and testing sets for future analysis. Since the data points depend on the x, and y coordinate and have high spatial correlation, we choose to split the coordinate into quadrants (4 blocks in each image) based on the median of x and y coordinates. Alternatively, we split each picture into 16 blocks based on the quantiles of the x and y coordinates and sample the total 48 blocks. We decided to use both approaches for further exploration, since more blocks will have more accurate predictions due to spatial correlation in 16 blocks, while less blocks would be more variant inside the 4 blocks. The training and validation sets are randomly selected from 90% of the total 48 blocks and 12 blocks, which are 43 blocks and 9 blocks, respectively. In addition, the test set contains 5 blocks in method 1 and 3 blocks in method 2 accordingly, which are randomly selected from the rest of the datasets.

In order to get the trivial classifier and to compare the accuracy obtained from the later models, we predict all labels to -1 on the validation and test set and set all training sets as 1 from the worst model. In other words, the testing and validation accuracy is the proportion of -1s inside the two sets; similarly, the training accuracy is the proportion of 1s in the training set. In this sense, we get 76.9% and 95.0% of accuracy for the validation set and test set in -1, meanwhile the training accuracy achieves 42.2% as the baselines. In terms of the trivial classifier, its testing, validation, and training accuracy would achieve its maximum when we have all the data points labeled -1 in the testing and validation sets, while the rest data are as the training set. An example would be splitting the data in the middle of x coordinate and assign those on the right to the testing and validation set, while those on the left are assigned to the training set.

In order to find the most significant variables used in the future models, we should do the first order importance methods. First, we try the backward step function to find the leftover variables from the logistic regression. As shown in the table below, we could see that all the variables are significant. Additionally, we fit the single value classifier method to compare the accuracies of each variable in the logistic regression model and get the top three features which are NDAI (87.68%), BF (78.6%), and SD (78%). From Figure 3, NDAI, BF, and SD also separate the two classes well. Nevertheless, the accuracies among all the variables are really similar and the density plots perform well in all variables, thus we would like to use all the features in future models.



```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.7033265  0.0809580  -70.45   <2e-16 ***
NDAI         1.9141244  0.0100342  190.76   <2e-16 ***
SD          -0.0694388  0.0011553  -60.10   <2e-16 ***
CORR         8.9005535  0.2298884   38.72   <2e-16 ***
DF           0.0191691  0.0004568   41.97   <2e-16 ***
CF          -0.0116459  0.0008159  -14.27   <2e-16 ***
BF          -0.0160027  0.0012122  -13.20   <2e-16 ***
AF          -0.0214604  0.0016375  -13.11   <2e-16 ***
AN           0.0421350  0.0014029   30.04   <2e-16 ***
```

3. Modeling

3.1 Model Assumption

We first tried the models that involve tuning parameters, such as, KNN, kernel SVM, and random forest. KNN's “model - free” classification method assumes that the close data points share similar labels and does not need any additional assumptions concerning the distribution that generates the data, which satisfies the spatial characteristic in this cloud dataset. Kernel SVM is motivated by the principle of optimal separation if the dataset is not linearly separable. In order to create more independent classifiers to average over, random forest is assumed to improve the bagged trees by small tweaks to decorrelate the trees. However, since these models are extremely time-consuming, we were not able to run these models.

For logistic regression, we assume that there is a linearity between independent variables and log odds, although the labels and features are not related linearly. Moreover, from the EDA above, some variables have normal distribution tendency although there are some distributions skewed among different features. We can, thus, analyze using linear discriminant analysis (LDA) model and quadratic discriminant analysis (QDA). The features are assumed to have equal covariance across the level of labels under normal distribution, in LDA, and the features are assumed to have its own covariance matrix and normally distributed, in QDA. Additionally, the features in this cloud dataset could be considered as independent to each other, such as, DF is one of the six zenith angles of the cameras, which is not related to the other angles and the measurements of NDAI, SD, and CORR. Therefore, the Naive Bayes model could be implemented in this prediction. Furthermore, the whole cloud dataset could be considered as the root, and continuous variables in the features are discretized prior to building the continuous variables decision tree in order to record the distributed recursiveness based on the attribute values. The models mentioned in this section are effective to achieve the results and will be used in the future prediction and comparison.

3.2 Accuracy Comparison across Models and Folds

By running a 5-fold cross validation on models mentioned above with blocks split in two ways, we are able to estimate the CV loss within each fold. As shown in Figure 4, the CV loss of models for each fold in two splitting methods are quite similar to each other. There is no obviously dominating model that returns extremely low CV loss. Though the difference is tiny, as shown in the table below, QDA has the lowest average CV loss under the 48 blocks method, while Decision Tree has the lowest average CV loss under the 12 blocks method. After fitting the models on the training sets, we are able to estimate the prediction accuracy under the testing set.

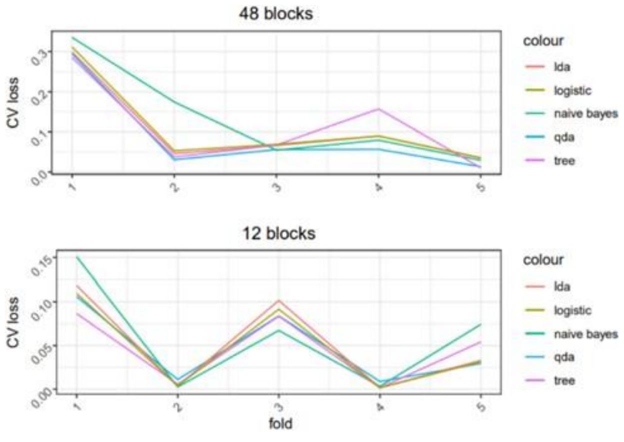


Figure 4: CV loss for different methods of splitting

Average CV loss across folds in two splitting methods:

Average CV loss / Methods	48 Blocks	12 Blocks
QDA	0.09016327	0.04788387
LDA	0.1070809	0.05178267
Naïve Bayes	0.13456384	0.05965858
Decision Trees	0.1115259	0.04639067
Logistic Regression	0.11176822	0.04763793

Accuracies for test dataset across different models in two splitting methods:

Accuracy / Methods	48 Blocks	12 Blocks
QDA	0.7948	0.8876
LDA	0.7955	0.8261
Naïve Bayes	0.7371	0.8076
Decision Trees	0.8267	0.8755
Logistic Regression	0.7908	0.8426

From the table above, the12 blocks splitting method has overall higher accuracy than the 48 blocks method. Besides, the decision tree model has the highest accuracy among all models both in 48 folds and 12 folds, which means that the decision tree has the best performance. Overall, the 12 blocks splitting method returns lower CV loss than the 48 blocks method, which means that the 12 blocks method returns higher accuracy.

3.3 ROC curve compared to different models

3.3.1 48 blocks

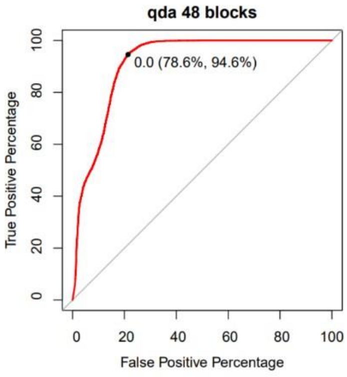


Figure 5: ROC for qda

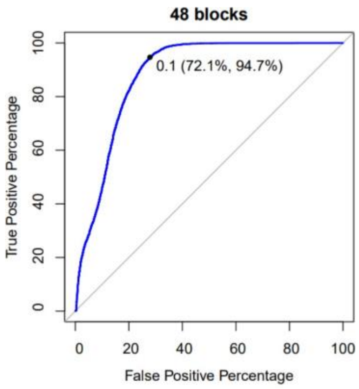


Figure 6: ROC for lda

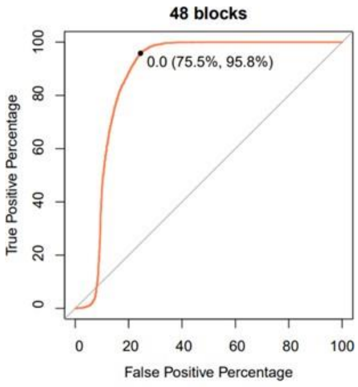


Figure 9: ROC for Naive Bayes

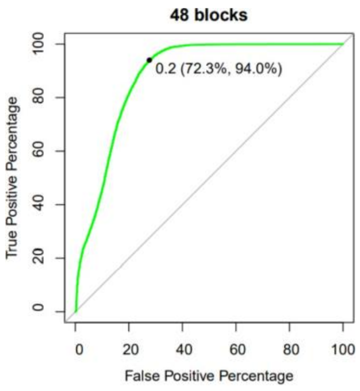


Figure 7: ROC for logistic regression

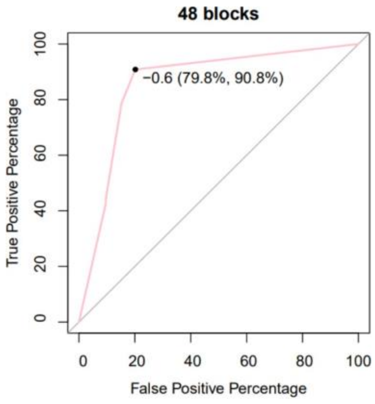


Figure 8: ROC for Decision Tree

The Figures 5 to 9 are the ROC curves for QDA, LQA, Logistic Regression, Decision Tree and Naïve Bayes in 48 blocks. Based on the Youden algorithm (J = sensitivity + specificity – 1), we choose the point that optimized the Youden function as our cutoff point. Compared with the cut off points in terms of specificity and sensitivity, Decision Tree has the highest specificity and Naïve Bayes has the highest sensitivity. However, the curve in Naïve Bayes is only above the baseline a bit, which is not good. On the other hand, the AUC of these models are 91.37%, 88.24%, 88.12%, 85.86%, and 87.36% accordingly, where QDA has the highest AUC and Decision Tree returns the lowest since ROC does not really work well when we have discrete classifiers for decision tree.

3.3.1 12 blocks

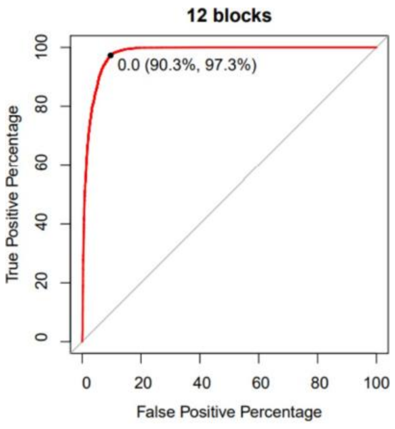


Figure 10: ROC for qda

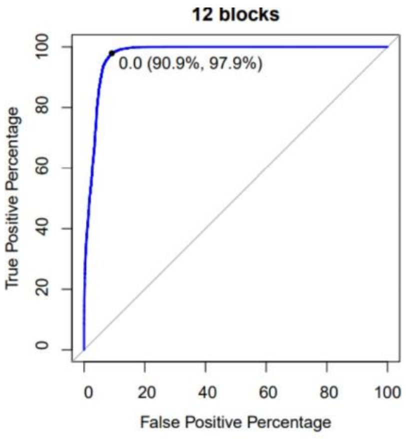


Figure 11: ROC for lda

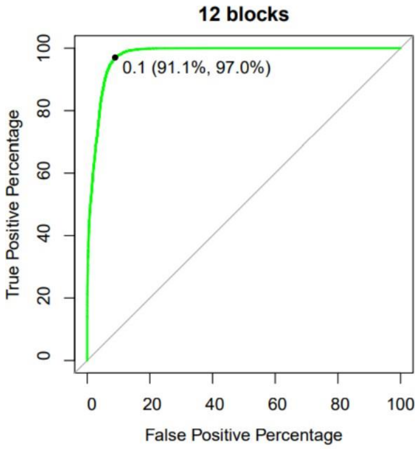


Figure 12: ROC for logistic regression

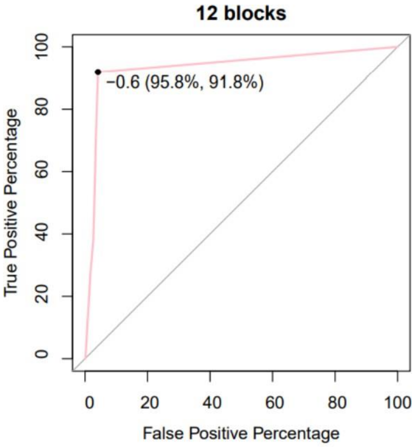


Figure 13: ROC for Decision Tree

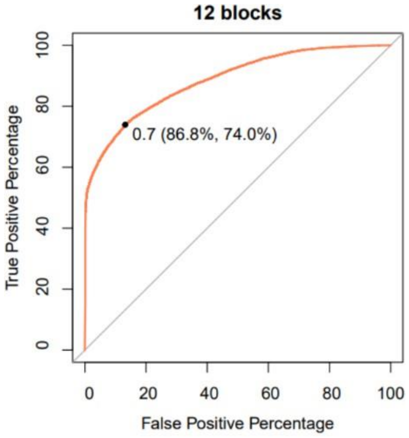


Figure 14: ROC for Naive Bayes

Figures 10 to 14 are the ROC curves for QDA, LQA, Logistic Regression, Decision Tree and Naïve Bayes under 12 blocks. This method has better overall performance than 48 blocks. Compared by the left corner points, we could see that Decision Tree has the highest specificity, and LDA has the highest sensitivity, while Naïve Bayes returns the lowest. The AUC of these models are separately 97.89%, 97.52%, 97.82%, 93.44%, and 88.51%. Additionally, we computed the F1 score based on the prediction we made from the testing set and its true labels. We are always looking for a F1 score that is close to 1. In this sense, Decision Tree returns the highest F1 score under the 48 blocks method, and QDA returns the highest F1 Score under the 12 Blocks splitting method, while Decision Tree returns the second highest F1 scores under the 12 blocks method.

F1 Scores for models under different block splitting method:

	QDA	LDA	Naïve Bayes	Decision Tree	Logistic
48 Blocks	0.8486	0.8428	0.8165	0.8631	0.8401
12 Blocks	0.9190	0.8815	0.8503	0.9095	0.8920

4. Diagnostics:

Based on the results above, we would like to pick Decision Tree as the best model out of the five models we have run on the ground that, though the Decision Tree returns a low AUC, it has a consistent and leading performance in the testing accuracy across the two different splitting methods. Similarly, it has one of the highest F1 scores across the two splitting methods. Compared to the other classifiers that depend on assumptions on the features, Decision Tree releases the pressure from prior beliefs that may not perfectly fit to the data observed. Because of the hierarchical property in feature splitting, it is worth pointing out that a small change in variables may have a huge impact on the Decision Tree and its performance. In this sense, we would like to see how much a change in features would influence the performance of Decision Tree based on our data. Thus, we would like to use the Model Reliance Algorithm to estimate the significance of each feature to the model. Model Reliance Algorithm is to scramble one feature at a time and to determine how much the loss by scrambling that feature. In our case, we scrambled the eight features one at a time on the testing set, fitting the Decision Tree on the training data and estimate the difference between the testing accuracy after scrambling and the testing accuracy before scrambling. As shown in Figure 15, for both of the splitting methods, the feature AN, CORR, and NDAI shows a non-zero reduction in testing accuracy for the Decision Tree model. It is worth mentioning that the 12 blocks splitting method generally has a larger loss in accuracy than that of the 48 blocks method. Additionally, the NDAI feature seems to influence the Decision Tree performance most effectively. This is not surprising on the ground that we also observed a very good separation of labels by NDAI from EDA. In addition to the Model Reliance Algorithm, we also ran an Algorithm Reliance on the Decision Tree. For Algorithm Reliance, in our case, we fit the decision tree with one feature dropped and estimate the testing accuracy loss. As shown in Figure 16, in both of the splitting methods, it seems that AN, CORR, and NDAI has a huge influence on the testing accuracy. It is worth mentioning that a similar result is also observed in Figure 15, when we are estimating the Model Reliance Loss. By

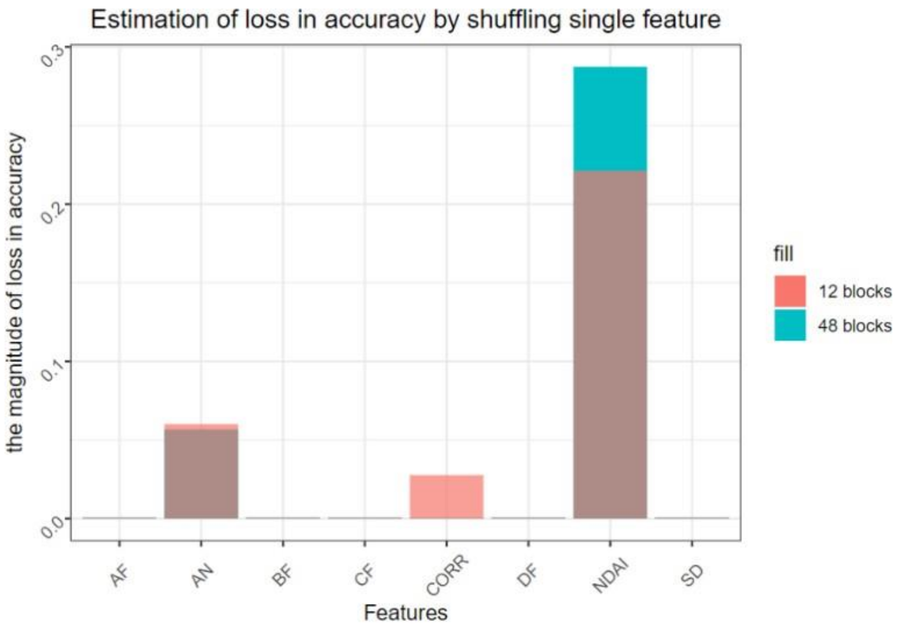


Figure 15: Model Reliance

contrast, the CORR feature has the largest impact on the Algorithm Reliance, while NDAI has the largest effect on the Model Reliance. Based on the tests, we may conclude that NDAI, CORR, and AN are the three most important features to the model. Although this may quite be different from what we derived from the single value classification, we notice that NDAI has appeared as one of the most significant features frequently in different kinds of measurement, and its influence seems consistent across different ways of splitting the data.

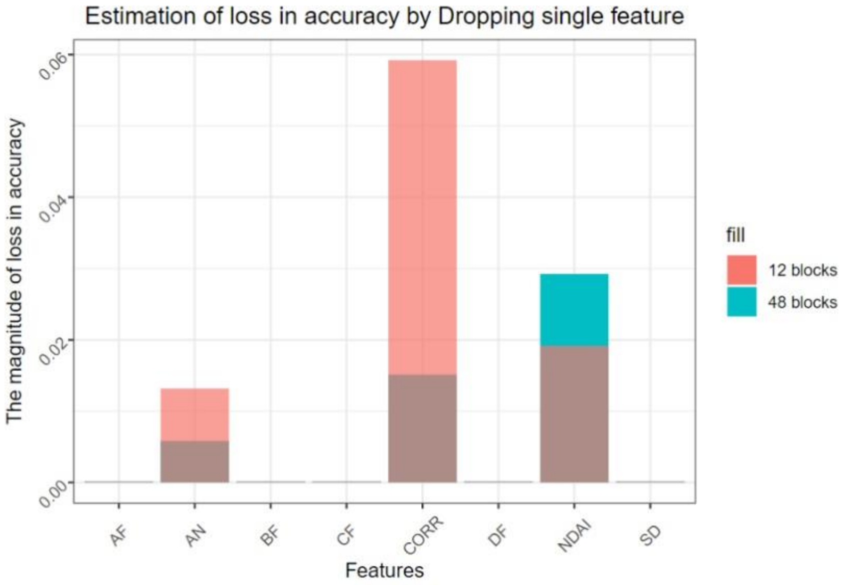


Figure 16: Algorithm Reliance

Built upon the understanding that AN, CORR, and NDAI has become the three most important features, we would like to explore whether there is a pattern of misclassification inside the distribution of these features.

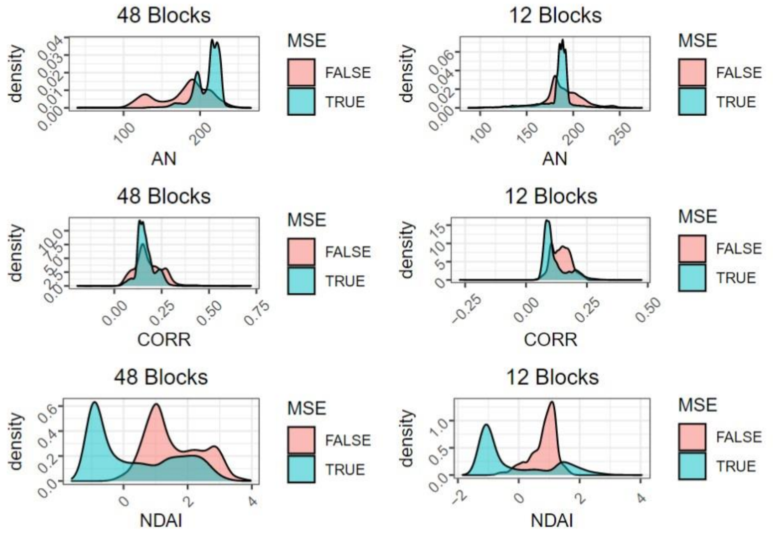


Figure 17: Misclassification distribution by features

In Figure 17, the TRUE label shown in green represents that the model correctly predicted the label in the testing set, and FALSE, in red, otherwise. It is worth mentioning that when NDAI reaches to an interval between 0 and 2, there are dominantly more points misclassified. The number of the misclassification reaches its peak, when the value of NDAI reaches to the middle of that interval, and this trend is quite obvious in both of the splitting methods. However, I do not see a strong trend of misclassification appearing in the other two features, except that, under 48 blocks, AN lower than 200 tend to be dominantly misclassified. It is worth pointing out that this observation does not suggest that there is no trend of misclassification exhibited for the other features. A more thorough exploration on the other features and a repeated experiment may be needed for the future study.

Considering the fact that we find that NDAI, CORR, and AN features have a strong influence on our model. That is a perturbation in those variables such as shuffling would influence the testing accuracy our model returns. In order to stabilize the model, I believe a random forest could be a better classifier than Decision Tree in this case, although it has a huge running time cost if tuning parameters is involved. However, we have to acknowledge that in terms of our model, given these data, the Decision Tree performs decently well in both of the splitting methods. Since it has good testing accuracy which is around 80%, I believe that our model can perform well without expert labels provided in the dataset. Considering its consistent performance across the two methods of splitting blocks, I believe that it should have a relatively stable performance across different ways of splitting the data.

Overall, Decision Tree works consistently well on this data across different data splitting methods. It achieves a decent level of testing accuracy. Although we recognize that the Decision Tree may not be stable in terms of feature perturbation and there exists a better classifier, random forest, we realize that the much less time cost that Decision Tree would compensate for the lack of stability in feature perturbation. Since the percentage of training set is quite large compared to the testing set, we recognize that the testing accuracy provided in this study may not be generalized enough, and thus a repeated experiment with exploration in the other features associated with misclassification pattern is suggested.

5. Reproducibility

For reproducibility, please refer to the ReadMe file.