Cloud Detection in the Arctic

Salim Damerdji and Hubert Luo

1. Data Collection and Exploration

Summary

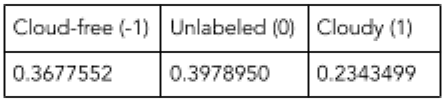
Because studying cloud coverage is key to climate science, Shi et al. wanted to create cloud detection algorithms that process images of the Arctic. The images comes from the Multiangle Imaging SpectroRadiometer (MISR) imager, which uses 9 cameras, each at a different angle and spectral band. Shi et al.'s data set came from ten 16-day-long MISR orbits over the Arctic. The data was collected from April 28 to September 19, 2002. The orbits covered a path (viz. 26) that covered a variety of terrain, including permanent ice, mountains, and glaciers. For each orbit, there were 57 data units with about 7.1 million pixels with 36 radiation measurements for each pixel. Each pixel represents a square region of side length 275 meters. Experts labelled pixels that had clouds, but only for the 71.5% of pixels for which the experts were highly confident in.

The study broke with previous literature by searching for cloud-free surfaces, not cloud-covered surfaces, to create an enhanced linear correlation matching (ELCM) algorithm. Using the three features of 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 the normalized difference angular index (NDAI), the study predicted probabilities of cloudiness to present a more accurate picture of cloud coverage in the Arctic and its effect on changes in the climate induced by increasing amounts of carbon dioxide.

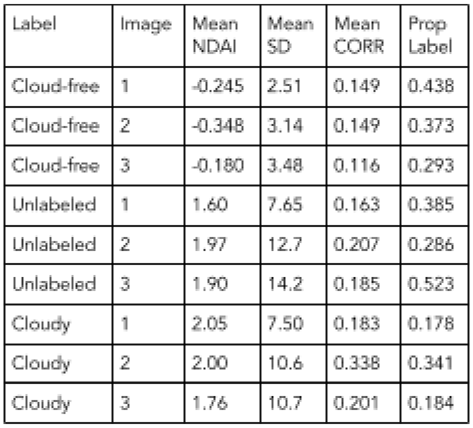
The study's upshot is in creating a cutting-edge cloud classifier with 92% accuracy, 100% coverage, and the speed to work in real time. The study can also impact future Earth science work by demonstrating that even simple models with QDA and just three features can distinguish cloud-free areas.

Summary of Data

The table below contains the proportion of pixels from all three images that are labeled as either cloud-free (-1), cloudy (1), or unlabeled (0). The table to the right has the average feature value for each label and image. It is noteworthy that the average feature values are all much higher for cloudy pixels than it they are for cloud-free pixels. That table also lists the proportion of



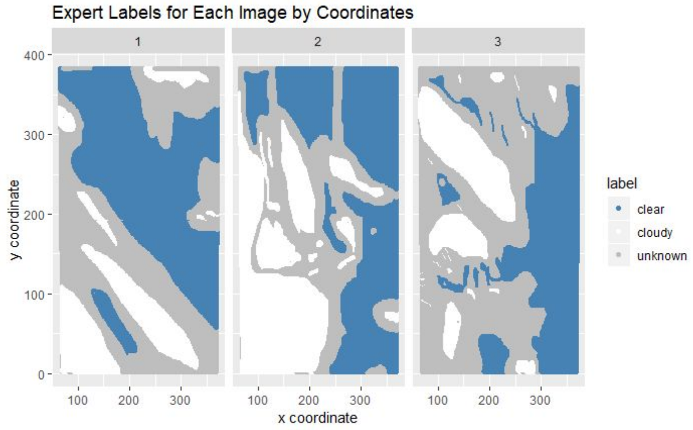
Cloud-free (-1)	Unlabeled (0)	Cloudy (1)
0.3677552	0.3978950	0.2343499



Label	Image	Mean NDAI	Mean SD	Mean CORR	Prop Label
Cloud-free	1	-0.245	2.51	0.149	0.438
Cloud-free	2	-0.348	3.14	0.149	0.373
Cloud-free	3	-0.180	3.48	0.116	0.293
Unlabeled	1	1.60	7.65	0.163	0.385
Unlabeled	2	1.97	12.7	0.207	0.286
Unlabeled	3	1.90	14.2	0.185	0.523
Cloudy	1	2.05	7.50	0.183	0.178
Cloudy	2	2.00	10.6	0.338	0.341
Cloudy	3	1.76	10.7	0.201	0.184

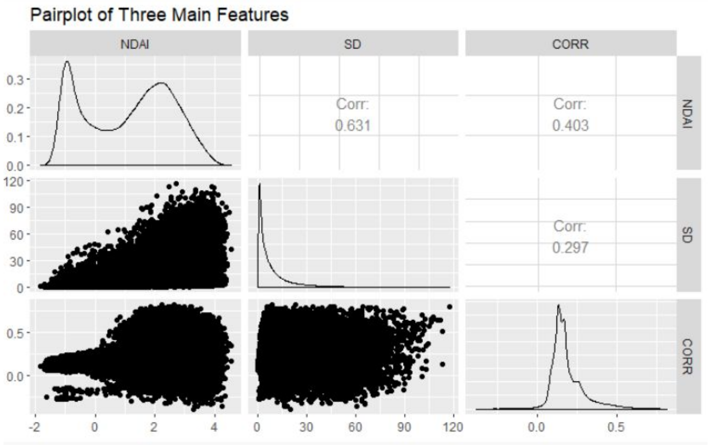
pixels per image for each label. Image 1 was mostly cloud-free, while image 2 was split relatively evenly between cloudy and cloud-free. Image 3 had a large portion which was unlabeled, meaning there was more ambiguity in the image with regards to cloud coverage.

This plot shows the images colored by the expert labels. There are clear areas where the pixels were predominantly of one label - for example, in the bottom left of image 1, it is almost entirely cloudy pixels (label 1). Therefore, an iid assumption for the samples is not justified for this dataset.

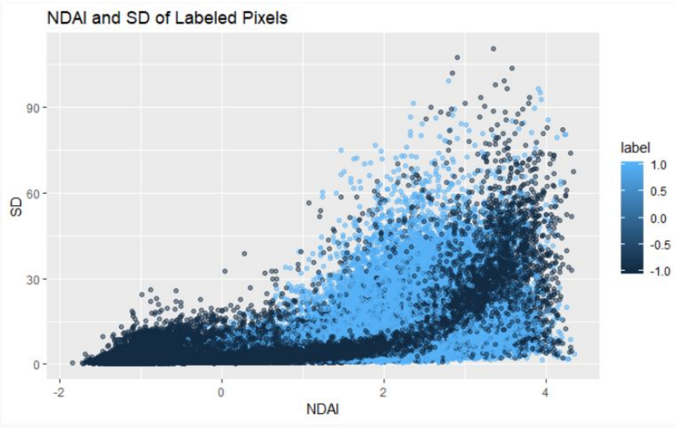


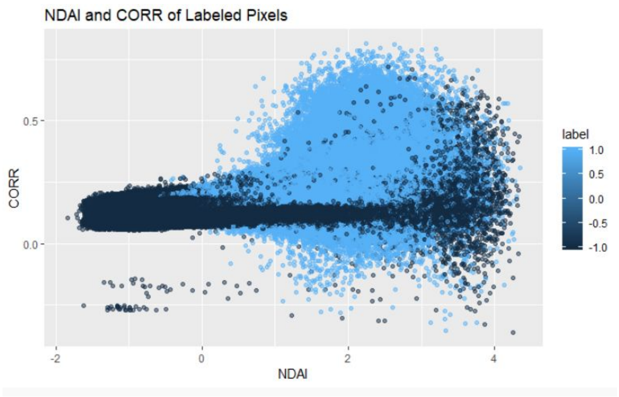
Exploratory Data Analysis

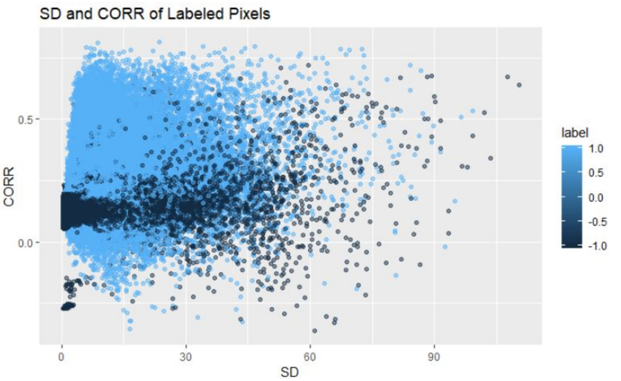
Quantitatively, there is a positive correlation between SD and the NDAI, with a correlation of 0.631. There is a weaker positive correlation between that of CORR and NDAI at 0.403 and an even weaker positive correlation between CORR and SD of 0.297. Visually, this is clear from the pairplot between NDAI and SD, where in general as the NDAI increases, so does the SD - the dots in the plot form a roughly triangular shape below the diagonal.



The following three plots only consist of pixels which had a label, i.e., excluding pixels with a label of 0. In general, there were a few noticeable differences between the expert labels based on the features. Some of the situations indicative of cloudy areas included: high NDAI in conjunction with low SD; medium NDAI along with high CORR; and either high or low CORR (i.e., greater than 0.2 or less than 0). Situations indicative of cloud-free areas included low values of both SD and NDAI and medium values of CORR (around 0.1-0.15).







2. Data Preparation

Data Splitting

Here are two approaches:

- Split A: you could divide each image into a 5x5 grid, yielding 75 such cells across all 3 images. Then randomly assign 50% of these cells to the training set, 25% to the validation set, and 25% to the test set. (That is, 29 cells for training, 14 for validation, and 14 for testing.)
- Split B: you could randomly assign one image to the training set, a second to the validation set, and a third image to the test set.

Both approaches are better than a naive approach where we randomly assign pixels to sets. However, this permits two neighboring pixels to be split up, such that one is in the training set and the other in the test set - that would be a trivial test. By merely memorizing the label assigned to the neighboring pixel from the training set, our model could accurately predict the label for the pixel in the test set. However, it would not have learned anything, and we would be underestimating our true error. Thus, we need a data split that takes into account the fact that the data is not iid.

Split B solves this problem because it will never split up two neighboring pixels: both will be in the training set, or both will be in the test set. However, this comes at a cost as we only get a third of your pixels to train with. In addition, we may get unlucky with which image is the test set. Perhaps the hardest image is in the test set, in which case we would overestimate our error. Or the opposite could happen and we would underestimate our error.

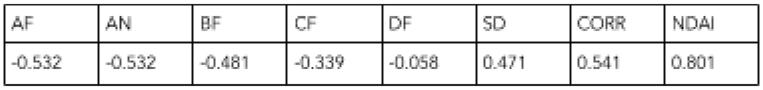
Split A solves this problem but at a cost: namely, it may sometimes split up two neighboring pixels. However, this should be fairly rare since it would only occur on the border of two cells that are split between two sets. This small downside is outweighed by the larger disadvantages to split B. We try both splits in this paper, although we think split A is more reliable in evaluating models.

Baseline Model

Suppose our model predicted all pixels were cloud free. On split A, this would yield a 36.4% accuracy on the validation set, and a 39.4% accuracy on the test set. On split B, this would yield a 29.3% accuracy on the validation set, and a 43.8% accuracy on the test set. This baseline model would have a higher average accuracy if there were more cloud-free pixels in the validation and test set.

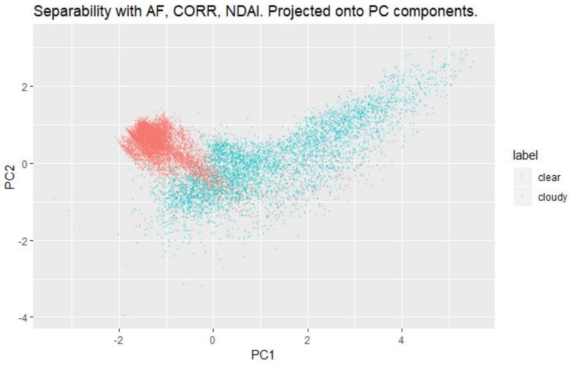
Best Features

The best three features should explain the most variance in the response, as compared with any other set of three features. The higher the absolute value of the correlation between the label and the explanatory variable, the likelier it is that knowing something about the explanatory variable gives us some insight into the label. Thus, for a first pass on feature selection, we look at the correlations between the predictor and label. After removing pixels that aren't labelled as clear or cloudy, here is how each predictor correlates with label:

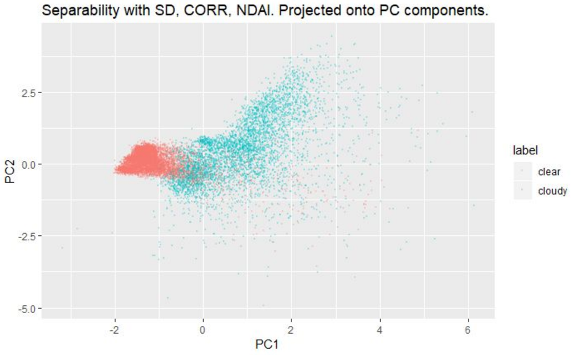


AF	AN	BF	CF	DF	SD	CORR	NDAI
-0.532	-0.532	-0.481	-0.339	-0.058	0.471	0.541	0.801

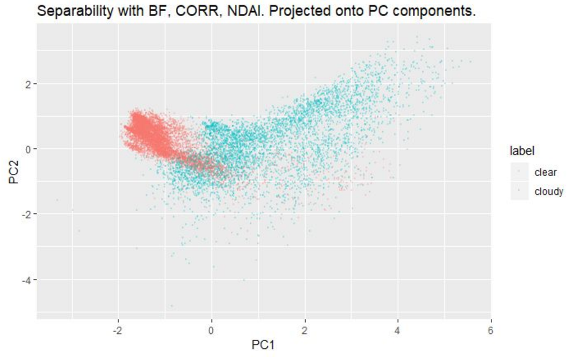
NDAI, CORR and AF correlate most strongly with the label. Now let's check for collinearity since if their collinearity is high, a third *independent* predictor may improve how much our feature vector explains the response variance. NDAI correlates with CORR by .514; that NDAI correlates with AF by -.65; and that CORR correlates with AF by -.71. This is a high amount of collinearity, and we'll check visually whether it's worth the sacrifice.



Separability with AF, CORR, NDAI. Projected onto PC components.



Separability with SD, CORR, NDAI. Projected onto PC components.



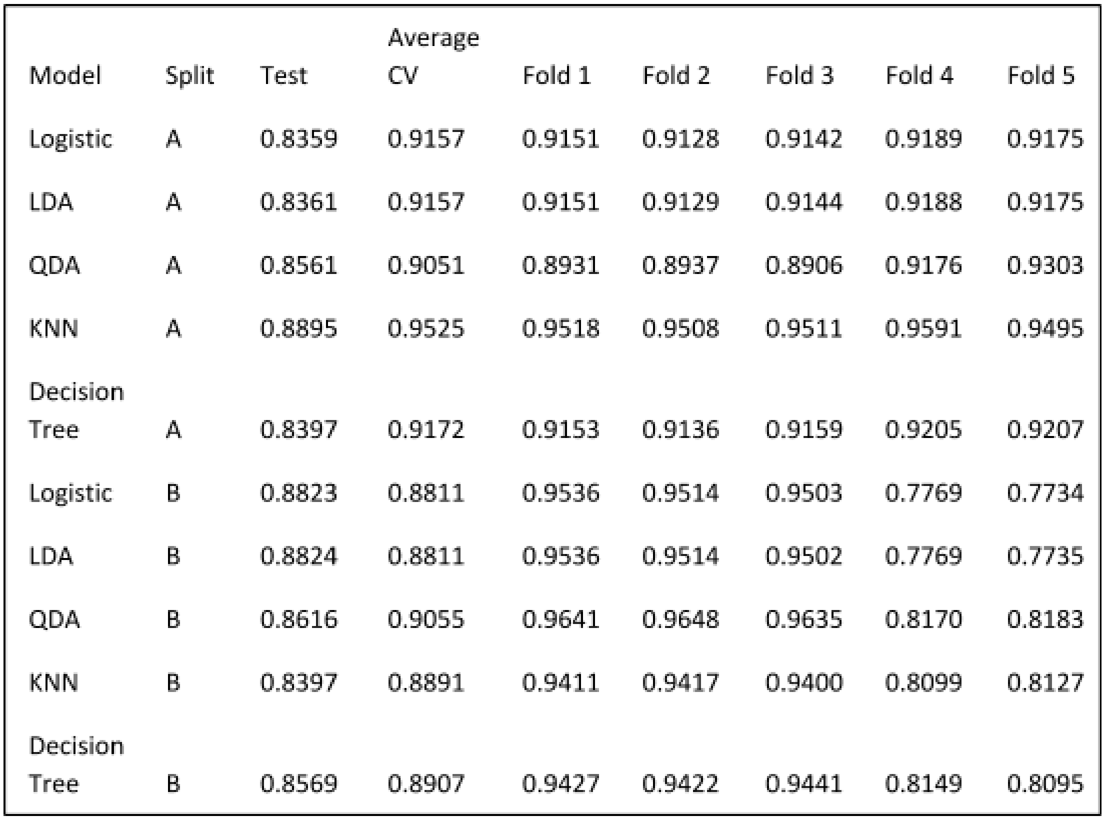
Separability with BF, CORR, NDAI. Projected onto PC components.

Now, let's see which set of three features lead to the most separation visually. Based on the last analysis, we will consider {AF, CORR, NDAI}, {BF, CORR, NDAI}, and {SD, CORR, NDAI} as our three candidate sets. We don't consider a feature vector with AN since it correlates with AF by 0.98, so adding AN in addition to or in lieu of AF won't change our predictive power.

To visually inspect separation, we project our data from 3-dimensional space to 2-dimensional space using PCA. This 2d space captures 88%, 88%, and 84% of the respective variances of the data using our three feature vectors. We use a random sample of twenty thousand points of our training data for plotting, so we don't spoil the test set. While this is a judgment call, our view is that the feature vector with AF has the best separability, confirming our analysis of correlation. This visual analysis takes into account concerns about collinearity because, based on visual inspection, it seems like more response is explained by the feature vector with AF, collinearity notwithstanding.

3. Modelling

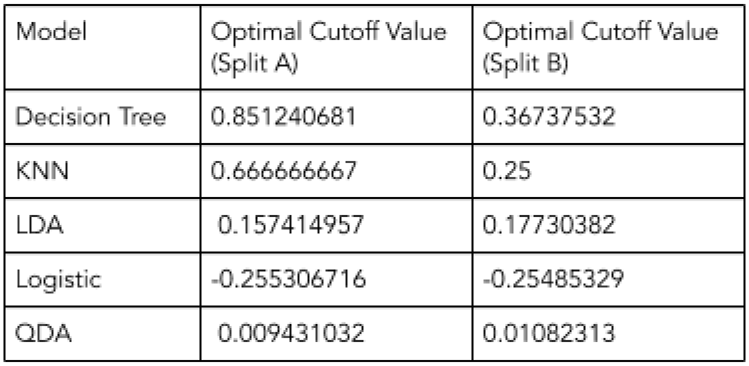
Logistic regression, LDA, QDA, k-nearest neighbour, and decision tree models were evaluated using 5-fold cross validation. The number of folds (k=5) was chosen to have samples of the data large enough to be representative of the overall dataset (lower bias than the bias from having smaller values of k) while also not having as much variance as larger values of k.



Model	Split	Test	Average CV	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Logistic	A	0.8359	0.9157	0.9151	0.9128	0.9142	0.9189	0.9175
LDA	A	0.8361	0.9157	0.9151	0.9129	0.9144	0.9188	0.9175
QDA	A	0.8561	0.9051	0.8931	0.8937	0.8906	0.9176	0.9303
KNN	A	0.8895	0.9525	0.9518	0.9508	0.9511	0.9591	0.9495
Decision Tree	A	0.8397	0.9172	0.9153	0.9136	0.9159	0.9205	0.9207
Logistic	B	0.8823	0.8811	0.9536	0.9514	0.9503	0.7769	0.7734
LDA	B	0.8824	0.8811	0.9536	0.9514	0.9502	0.7769	0.7735
QDA	B	0.8616	0.9055	0.9641	0.9648	0.9635	0.8170	0.8183
KNN	B	0.8397	0.8891	0.9411	0.9417	0.9400	0.8099	0.8127
Decision Tree	B	0.8569	0.8907	0.9427	0.9422	0.9441	0.8149	0.8095

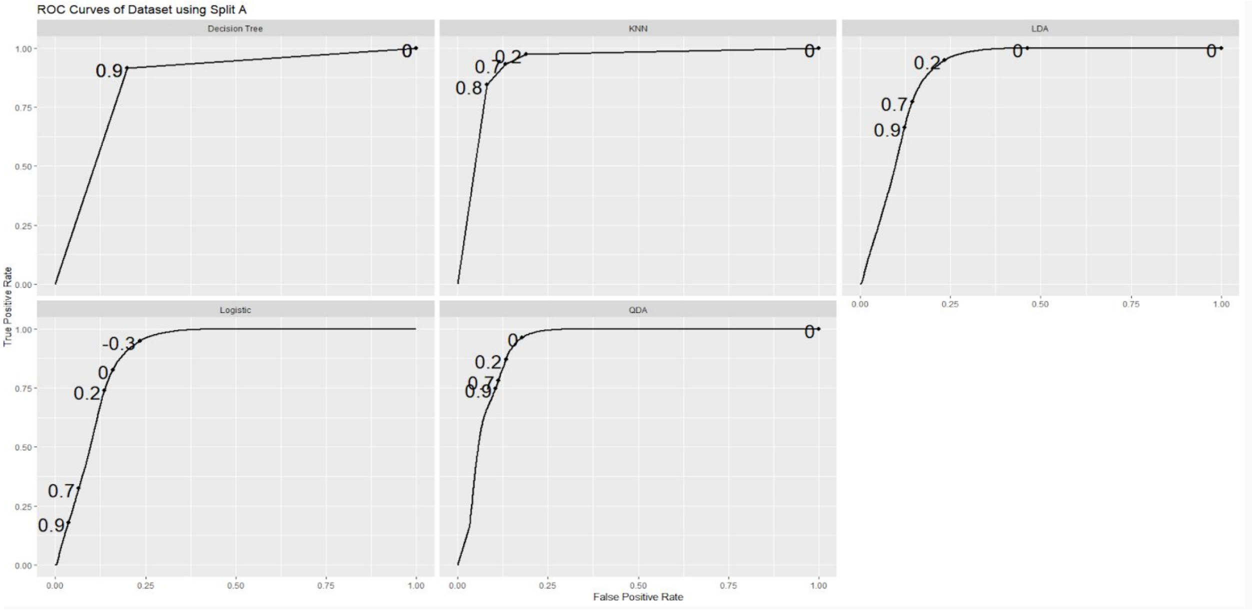
For knn, we let the number of neighbours k=3. We show later in Part 4a that this parameter is close to optimal. The table of accuracies on the left demonstrates that the k-nearest neighbours model performed best on the data from split A, while LDA, Logistic, and QDA all performed similarly well on the data from split B. However, most of the accuracies were within a similar range, i.e., between 83% and 95%, and differences between model accuracies were relatively minor for data from the same split. In addition, the average cross-validation accuracies were generally higher than those of the testing accuracies.

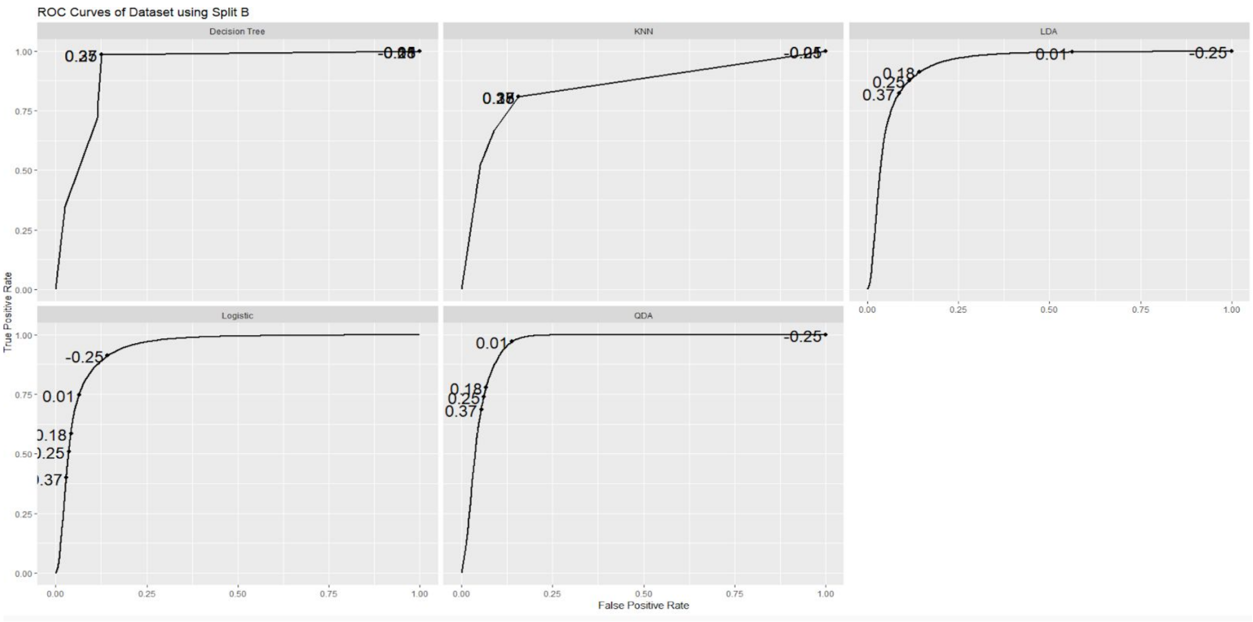
Folds 1, 2, and 3 for split B had significantly higher accuracies than folds 4 and 5, perhaps indicative of issues in the original split of data using the second method. Other potential issues with the results from split B are that if the data has a high number of clouds, we may be disproportionately rewarding models with high recall and not punishing low precision enough. The opposite problem occurs if the test image has few clouds, and thus generally split A would be advisable to split B due to the potential issues inherent in split B.



Model	Optimal Cutoff Value (Split A)	Optimal Cutoff Value (Split B)
Decision Tree	0.851240681	0.36737532
KNN	0.666666667	0.25
LDA	0.157414957	0.17730382
Logistic	-0.255306716	-0.25485329
QDA	0.009431032	0.01082313

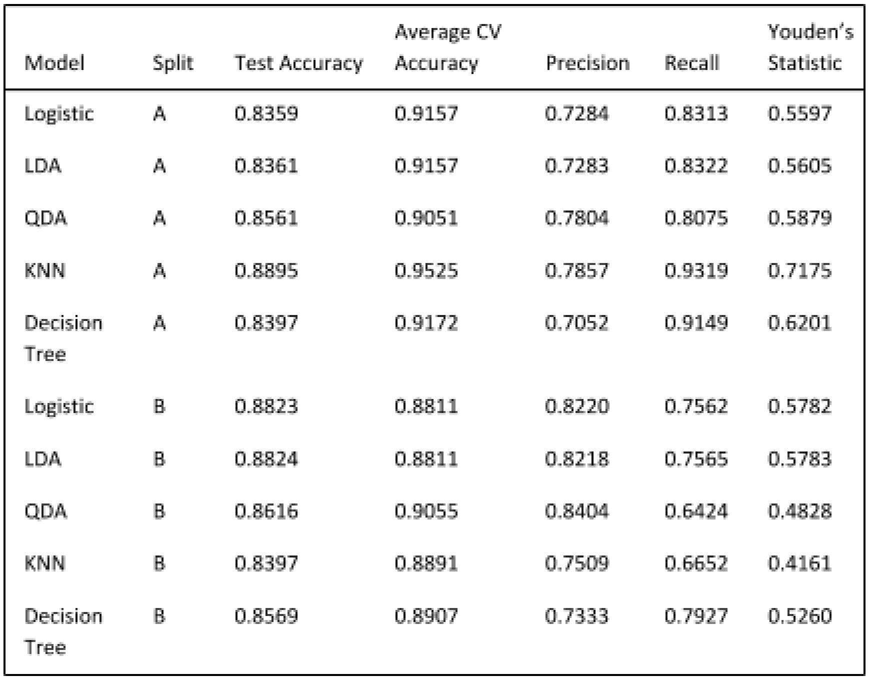
The cutoff value varies depending on the specific model used. To calculate the optimal cutoff value, Youden's J statistic (True Positive Rate - False Positive Rate, equivalent to Precision + Recall - 1) was used. Youden's statistic was chosen as it balances the desire to maximize the true positive rate while also minimizing the false positive rate to estimate the informedness of a prediction. The optimal cutoff values as determined by Youden's statistic are displayed in the table above for both splits A and B, and the cutoff values are marked on the ROC curves as well for both splits A and B below. The ROC curves for data from split A on this page, and the ROC curves for data from split B on the following page.





The table below contains additional performance metrics such as precision, recall, and Youden’s J statistic. Test and Average CV accuracies were discussed earlier in this report. For split A, k-nearest neighbours had not only the highest test/average CV accuracies, but also the highest precision, recall, and Youden’s statistic values, indicating it was the preferable model for split A out of all the ones examined. In general, split A resulted in higher recall than precision, indicating that these models are more sensitive, i.e., they do a better job of identifying the presence of a cloud. However, there were a relatively high number of false positives where the models using split A thought there were clouds when the area was actually cloud-free.

For split B, QDA had the highest precision out of all the models examined but also the lowest recall, thus demonstrating that the model was often classifying areas as cloud-free even when they were cloudy. QDA had the lowest Youden’s statistic out of the models for split B and is thus not a good choice for split B. On the other hand, the Logistic and LDA models both had relatively



Model	Split	Test Accuracy	Average CV Accuracy	Precision	Recall	Youden’s Statistic
Logistic	A	0.8359	0.9157	0.7284	0.8313	0.5597
LDA	A	0.8361	0.9157	0.7283	0.8322	0.5605
QDA	A	0.8561	0.9051	0.7804	0.8075	0.5879
KNN	A	0.8895	0.9525	0.7857	0.9319	0.7175
Decision Tree	A	0.8397	0.9172	0.7052	0.9149	0.6201
Logistic	B	0.8823	0.8811	0.8220	0.7562	0.5782
LDA	B	0.8824	0.8811	0.8218	0.7565	0.5783
QDA	B	0.8616	0.9055	0.8404	0.6424	0.4828
KNN	B	0.8397	0.8891	0.7509	0.6652	0.4161
Decision Tree	B	0.8569	0.8907	0.7333	0.7927	0.5260

high Youden’s statistics, although both were much lower than the Youden’s statistic for the k-nearest neighbours model in split A. This indicated that for split B, logistic or LDA were relatively better - the differences in the metrics between these two models were marginal and it is unrealistic to definitively conclude that one model is better suited. For split B overall, the models generally had higher precision than recall, with the exception of the decision tree model, which meant they were more adept at minimizing the number of false positives when a cloud-free area was identified as cloudy at the cost of fewer instances of correctly identifying the presence of a cloud.

Comparing the results for splits A and B overall, it is clear that split A overall is preferable over split B and the k-nearest neighbours model for split A especially is ideal as it has both high precision and recall, resulting in a high Youden’s statistic. Although some models for split B such as Logistic and LDA had high precision, the recall was much lower relative to the models in Split A. Comparing their Youden’s statistics, the models in split A and k-nearest neighbours especially for split A were preferable. Further discussion and support is provided in the next section below.

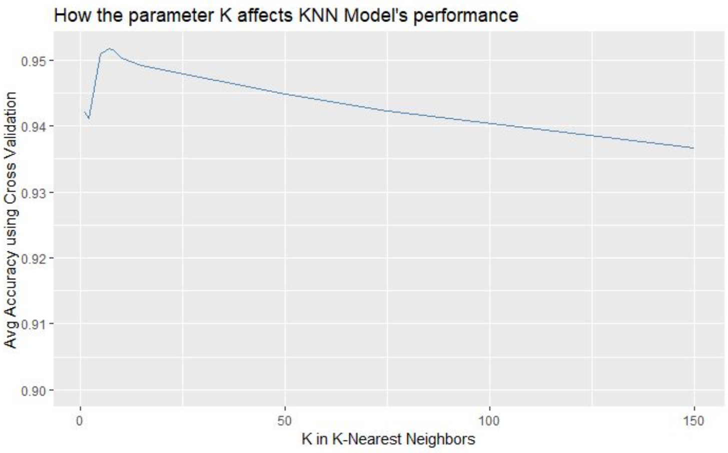
4. Diagnostics

We choose to evaluate our models based on split A, based on the results from Section 3 above. In addition, these models are trained on a training set that is 2.3x larger, so these models would be expected to be relatively better. Even though the models trained on split B score the same on their test set, these results are less reliable since Split B’s test set is a single image, so there’s no guarantee of generalizability. This view is supported by the fact that split A yields models with a higher average cv accuracy. Within split A, it looks as though knn has the highest performance. Its test accuracy is 89% and its average cv accuracy is 95%.

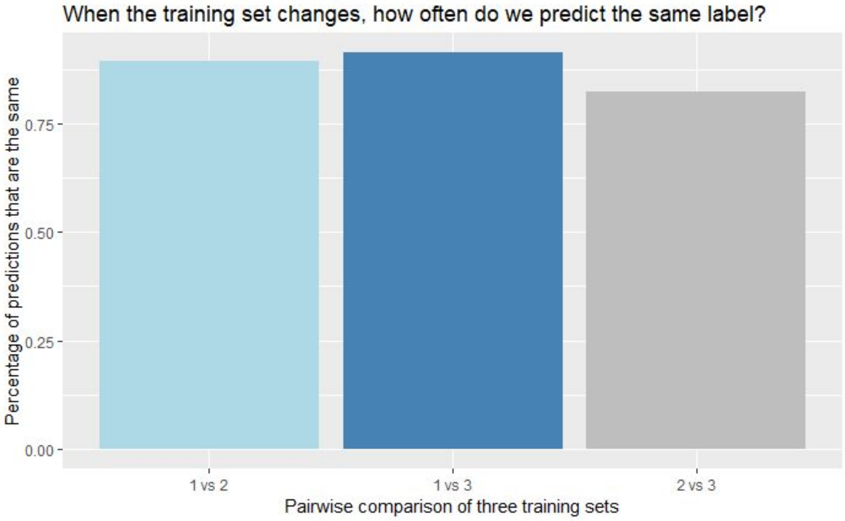
A - Analysis of Model

To combine our precision and recall, we can look at our f1 score: 0.8525303. This diagnostic tells us the overall performance of our binary classifier in a way that equally weights precision and recall.

On the right, you can see the results of using cross-validation to find an optimal k value for our model. Our accuracy peaks between k=5 and k=10. While a small k decreases bias and increases variance, it looks as though a relatively small k is near the optima. Let’s use k=5 since it’s less memory intensive. (Using k=5 yielded a



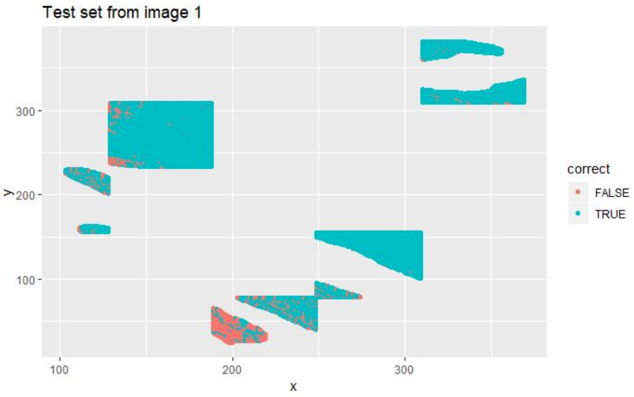
similar accuracy on the test set.) Our model appears to be somewhat robust to the parameter k since our results do not vary dramatically; any k ranging from 1 to 100 will yield at least 94% accuracy. That said, there is a big bump to picking at least k=3, and this gain vanishes around k=100.

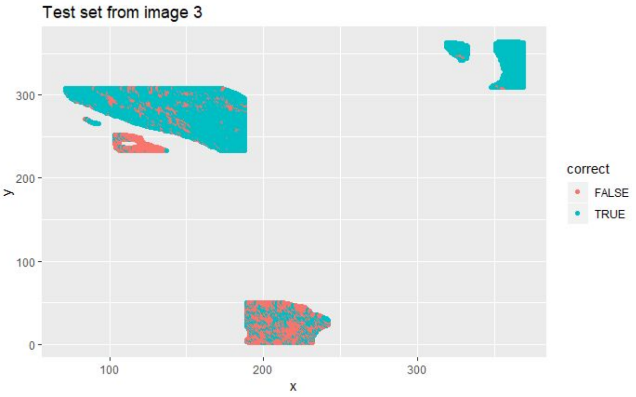


Our model also yields stable predictions across different training sets. We trained our algorithm on three small training sets of 1,000 samples, and compared their predictions on one test set. Looking at pairwise comparisons of these three resulting models, they make the same predictions 89%, 91%, and 82% of the time. Thus, our knn approach is fairly robust and stable, despite changes in the training set. Furthermore, these are conservative estimates: using a larger sample should decrease instability further. We only used training sets of 3,000 samples, but we would have access to 160,000 samples if we train on the train and validation sets.

B - Misclassification

Our errors seem largely concentrated near each other. For image 1, our errors are mostly in one contiguous region on the bottom. The same can be said for image 3. In addition there's an island on the left in image 3 that we fail to classify.





After doing some snooping on the poorly labelled area in image 1, the problem is as follows. The region should be labelled clear, but it's labelled cloudy because its SD value is so unlike the other clear regions. Here are the SD quantiles for the poorly labelled cluster on image 1:

25%	50%	75%
6.440247	13.460489	25.516576

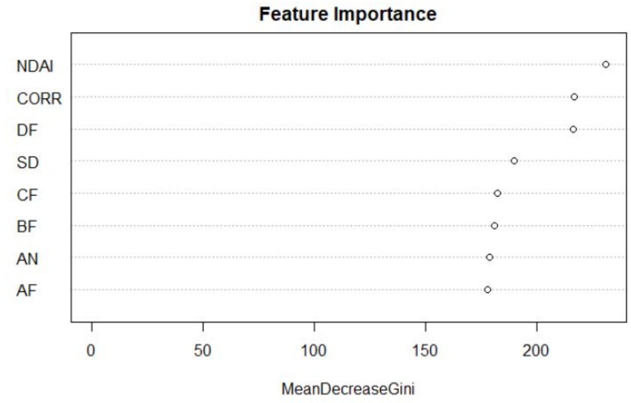
And here are the SD quantiles for the rest of the cloud-free pixels on image 1:

25%	50%	75%
1.212182	1.782695	2.737428

In fact, all three poorly labelled regions have the following in common: they all should be labelled cloud-free, and they all have anomalous SD values.

C - Improvements and Future Data without Expert Labels

There are a few potential solutions. First, it's worth noting that SD was one of the predictors that our KNN model was trained on. So, we could simply remove SD from our KNN model, and consequently, our model would not be fooled by cloudfree regions with SD values that are atypical for cloudfree regions. This is not a perfect solution since SD could still be informative some of the time, and we don't want to lose this information if we don't have to. However, it would be a simple fix.



Second, we could train a random forest on the subset of the data that we have incorrectly labelled. (Of course, we should not train our model using the incorrectly labelled data from the test set, but rather from the training set.) Then from this model, we could examine the feature importance vector, and identify the 3 most important features for these regions. An example of such a plot is seen on the right, where a feature importance plot was constructed from a random forest model on the misclassified training data - the three most important features identified by the model are NDAI, CORR, and DF. Then we could use these features to construct another knn model.

At that point, we would have a choice. If our second model performed better than the first on average cross-validation accuracy, we could simply ditch the first model. Otherwise, we could combine both models into one ensemble models. To predict a label for a pixel, we would make a probability prediction with each of the two knn models, and take the average. If the average was greater than zero, we would predict the pixel is cloudy; if less than zero, then cloud-free.

While both models will have some deficiencies on certain locations, our prior is that these deficiencies should be different, and thus, that they should cancel out when we take the average of our two models. Of course, whether this pans out in practice is something we can only ascertain by trying.

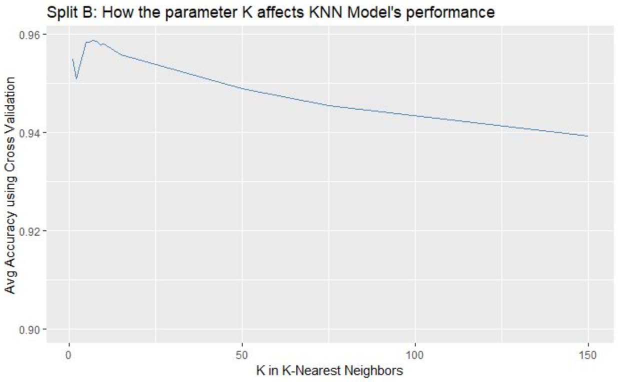
A downside to this approach is that fitting values will take twice as much time since we will have to run each model on each pixel. But this is only a scalar increase in complexity, which drops out asymptotically. Moreover, for climate change research, these values don't typically need to be recomputed in real time, so this extra computation is not prohibitively expensive.

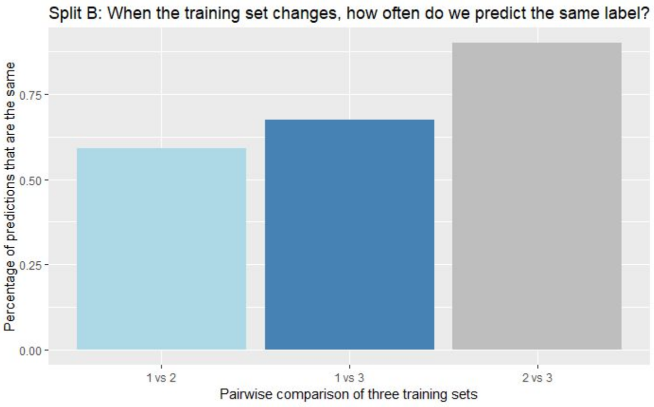
We suggest trying both options outlined above, and moving forward with the model that performs the best in practice.

On future data, we expect at least an accuracy of 89% on future data, and this is a conservative estimate because we have yet to train our model on the entirety of the training set and validation set. However, our model underperforms at precision compared to recall, so we will do worse with data sets where there are few clouds.

D - Changes in Splitting the Data

Using split B changes our diagnostics slightly. Our f-1 score dips to 0.7054679. Our new model seems to perform roughly the same as a function of k. But its stability is far less impressive when trained on different training sets.





Following our recipe with split A, we constructed training sets and compared them as with split A. Looking at pairwise comparisons of these three resulting models, they make the same predictions 59%, 92%, 67% of the time when we use different training sets. This is far lower on average.

We do get the same clusters of mislabelled points when we use the second split. Thus, we still think the same fixes could be applied no matter the model.

E - Conclusion

In this section, we've shown that our knn model has .85 f1 score, that we have 95% accuracy when k=5, that our knn model gives stable predictions (approximately 87% of predictions are the same) even when trained on completely different training sets; and that our knn model is not hyper-sensitive to the parameter k. We've also discussed limitations of our model: it will underperform on less cloudy images, and on cloud-free regions with anomalous SD values. We believe using a model that is less sensitive to anomalous SD values is will address that shortcoming; in particular, we could simply drop SD from our model, use an ensemble approach, or use a decision tree to point us towards a better feature vector for those regions.