Lab 2 - Cloud Data, Stat 214, Spring 2025

Anonymous

March 21, 2025

1 Introduction

Cloud detection in polar regions presents a significant scientific and statistical challenge due to the similar spectral and radiative properties shared by clouds and snow- or ice-covered surfaces. Accurate identification of clouds is critical for climate modeling, as clouds play a dual role in Earth's energy balance—contributing to warming by trapping heat and to cooling by reflecting sunlight. However, traditional cloud detection methods often struggle in the Arctic, where the low contrast between clouds and the surface in visible and infrared wavelengths limits their effectiveness.

In this lab, we develop a cloud detection algorithm using data from NASA's Multi-angle Imaging SpectroRadiometer (MISR) onboard the Terra satellite. The primary goal is to build a predictive model that distinguishes cloud from non-cloud pixels in MISR imagery. MISR captures radiance from multiple viewing angles, providing valuable information about the vertical structure of the atmosphere. This multi-angle perspective aids cloud identification by leveraging altitude and scattering characteristics, rather than relying solely on brightness or temperature. Satellite imagery contributes to climate analysis through two key observations: different objects exhibit distinct radiative signatures (colors), and measurements taken closer to the Earth's surface tend to reflect higher temperatures.

In addition to raw radiance data, we incorporate three engineered features—NDAI(Normalized Difference Angular Index), SD (Standard Deviation), and CORR (Correlation)—which are designed to enhance cloud detection accuracy. These features are informed by domain-specific knowledge and have demonstrated effectiveness in differentiating clouds from background surfaces.

This project is structured into several components: Exploratory Data Analysis (EDA), feature engineering, model selection, and model evaluation. Through these steps, we aim to explore the data, develop informative features, build robust models, and assess their predictive performance. We will also discuss the broader implications of our findings for climate science and highlight opportunities for future research.

2 Exploratory Data Analysis

2.1 Data Collection and Description

The dataset used in this lab was collected from NASA's Multi-angle Imaging SpectroRadiometer (MISR) onboard the Terra satellite. MISR captures satellite imagery in the red spectral band from nine different camera angles, which provides unique information about surface and atmospheric characteristics—especially helpful in distinguishing clouds from snow or ice in polar regions.

For this project, we worked with 164 image files in .npz format, each representing a different MISR scene. Every image contains pixel-level data across ten features, including spatial coordinates (X, Y), radiance values from five MISR angles (DF, CF, BF, AF, AN), and three engineered features: NDAI (Normalized Difference Angular Index), SD (Standard Deviation of radiance), and CORR (Correlation between angular radiances). These engineered features were developed by researchers based on domain expertise to enhance cloud detection performance in complex terrains.

Each pixel also includes a label indicating the presence or absence of a cloud: +1 for cloud, -1 for no cloud, and 0 for unlabeled pixels. However, only three images—O013257.npz, O013490.npz, and O012791.npz—contain expert-annotated labels, which we use as ground truth during our model development and evaluation phases. The remaining 161 images are unlabeled and serve as potential candidates for model inference after training.

This structured dataset allows us to explore how both raw radiance data and derived features can help distinguish clouds from background surfaces, while also reflecting the challenges of working with real-world satellite imagery that includes noise, class imbalance, and limited labeled data.

2.2 EDA

The spatial visualizations of expert-labeled cloud masks across the three images (O013257, O013490, and O012791) are shown in Figure 1 and show clear patterns in how clouds and non-cloud regions are distributed. Cloud-labeled pixels (+1) and no-cloud pixels (-1) appear in distinct clusters, often aligned along certain altitudes or terrain features, while unlabeled areas (0) typically form the background. These plots demonstrate that spatial information (X, Y) may be useful for modeling local patterns, but more importantly, they reveal that clouds and no-cloud areas often occur in separate, structured regions, supporting the feasibility of spatially aware classification methods.

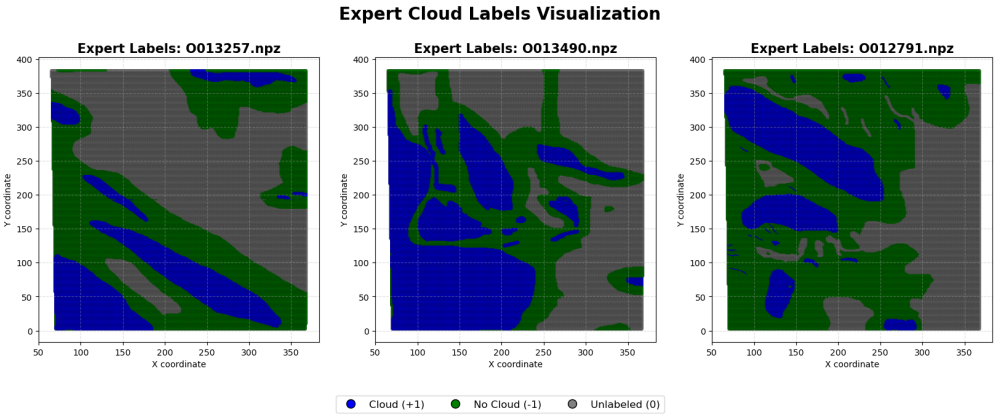


Figure 1: Visualizing the expert-labeled images.

The violin plots comparing feature distributions across cloud and no-cloud classes shown in Figure 2 provide deeper insight into the relationships between radiance and cloud presence. NDAI shows clear class separation—values are generally higher for cloud pixels, indicating its effectiveness as a discriminative feature. Similarly, SD and CORR distributions show meaningful differences, although SD is more skewed and may require normalization. Radiance values across the MISR angles also differ between classes; in particular, cloud pixels tend to have higher or more variable radiance across multiple angles, suggesting that the angular scattering properties captured by MISR are indeed informative. These plots confirm that multiple features—not just one—carry useful signal for separating clouds from the background.

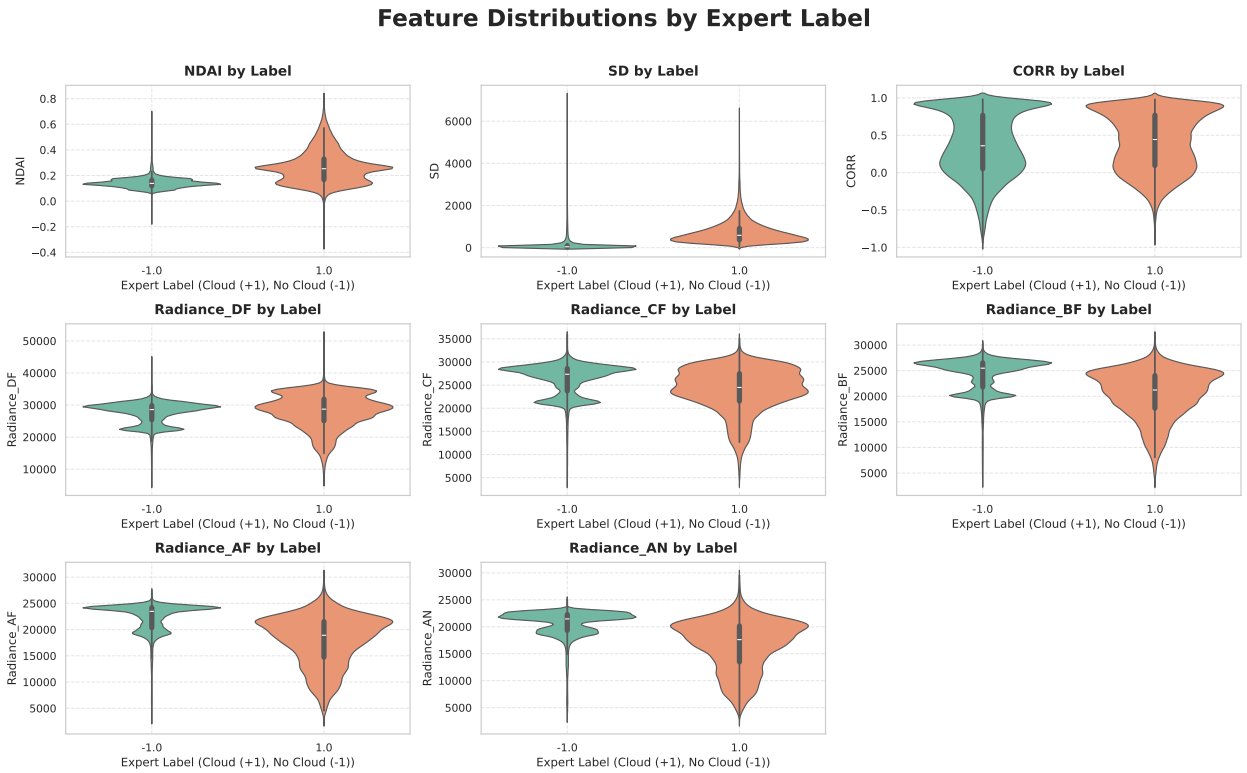


Figure 2: Violin Plots for Provided Features

Figure 3 shows the correlation heatmap amongst the provided features. While radiance values across different angles are strongly correlated with each other (especially among BF, AF, and AN), engineered features like NDAI and CORR show weaker correlations with raw radiance. This indicates that NDAI, SD, and CORR introduce additional, non-redundant information to the model. Interestingly, NDAI is negatively correlated with most radiance features, reflecting how the angular differences contribute to its calculation. These relationships highlight that while radiance angles provide foundational data, the engineered features derived from domain knowledge offer complementary and potentially more robust signals for classification. This blend of features enhances model interpretability and improves chances for generalization to unlabeled scenes.

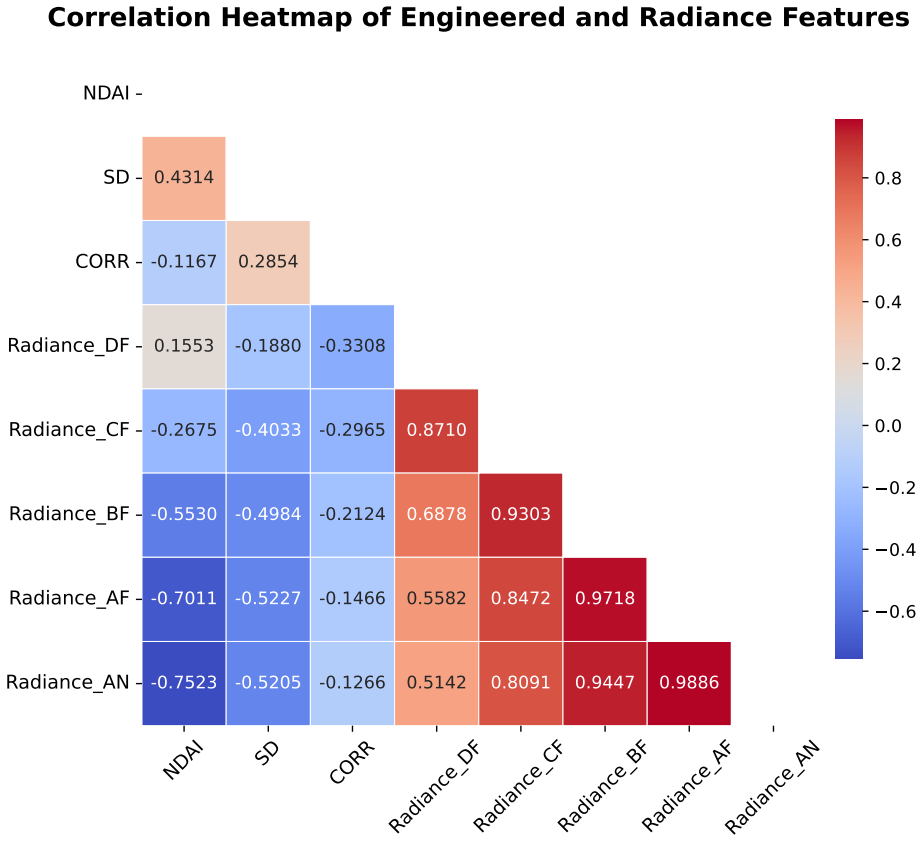


Figure 3: Correlation Heatmap of Provided Features

2.3 Split Train/Test dataset

Two different approaches were taken to divide the data set into training and testing sets–one for EDA and another for model training. This section focuses on the split used for EDA, and Section 4 lays out how the data was split for model training.

We split the data into 80% training and 20% testing sets using stratified sampling to maintain the class distribution of cloud and no-cloud labels. This approach ensures that both classes are proportionally represented in each set, which is especially important in cloud detection tasks where class imbalance is common. The training set is used to build and optimize the model, while the test set provides an unbiased evaluation of model performance on unseen data. This setup reflects a realistic application pipeline where models are trained on available labeled data and deployed to make predictions on new satellite imagery.

3 Feature Engineering

3.1 Three Key Features Selection

For data preparation, non-predictive columns (X, Y, and image_id) were removed from both the training and validation datasets to focus solely on spectral and spatial features. The target variable (Label) was retained for classification purposes.

3.1.1 Area Under Curve Analysis

A logistic regression classifier was configured with a maximum of 1000 iterations and a fixed random_state=214 to ensure reproducibility. The model was iteratively trained using an incremental feature subset approach, where features were added sequentially in the predefined order: NDAI, SD, CORR, Radiance_DF, Radiance_CF, Radiance_BF, Radiance_AF, and Radiance_AN. At each iteration, the Area Under the ROC Curve (AUC) was calculated using predicted probabilities from the validation set to assess performance. The results showed that the highest AUC (0.92) was achieved using only the first feature (NDAI), and performance slightly declined as additional features were incorporated, stabilizing in the range of 0.91 to 0.93 for larger subsets. The initial features, particularly NDAI and SD, contributed most significantly to model performance, while the later radiance-based features demonstrated diminishing returns.

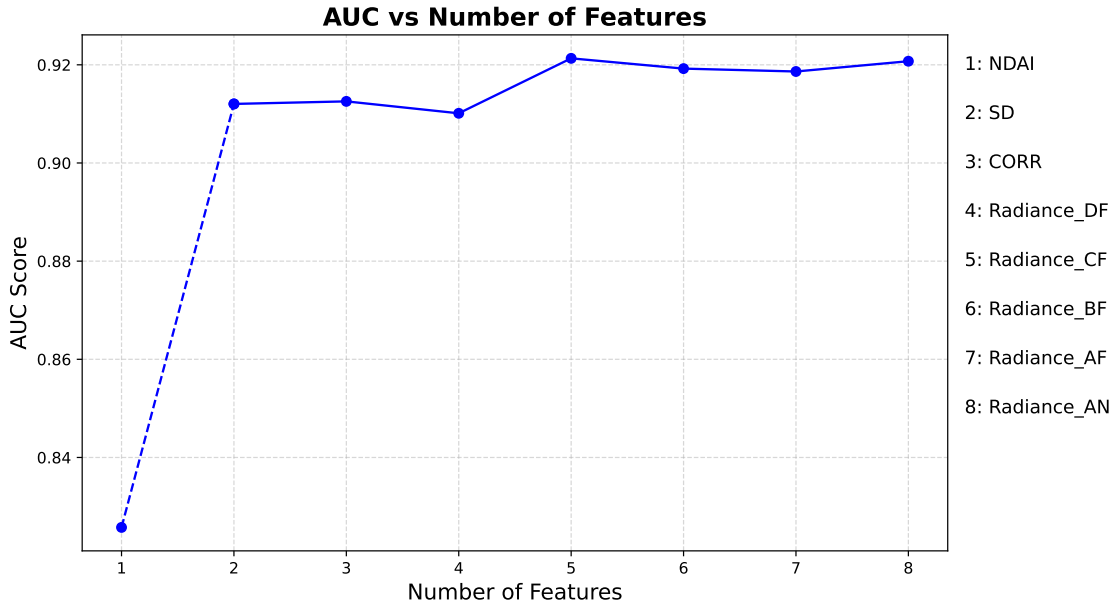


Figure 4: AUC vs Number of Features

3.1.2 Random Forest Feature Importance Test

SD stands out as a top predictor, achieving a Mann–Whitney U test p-value of 0.0, the highest Spearman correlation coefficient (0.735), and the strongest Random Forest feature importance (0.180). Its moderate correlation with NDAI ($r = 0.4314$) and low correlation with the other radiance variables ($r < 0.3$) demonstrate that it contributes unique information without significant

redundancy. From a domain standpoint, SD captures local variability in radiance measurements, effectively distinguishing more turbulent, high-variance cloud regions from the homogeneous surfaces typical of non-cloud areas.

NDAI delivers high predictive power, evidenced by a single-feature AUC of 0.83 and the second-highest Spearman correlation (0.546). It also exhibits low pairwise correlations (¡ 0.3) with other features, reducing the risk of multicollinearity. Critically, NDAI was engineered specifically for cloud detection, as supported by prior literature, reinforcing its strong domain relevance.

Radiance_AN emerges as the best radiance-based feature, yielding the highest Spearman correlation (0.496) and notable Random Forest importance (0.046) among all radiance channels. Although it is extremely collinear with Radiance_AF (r = 0.9886) and Radiance_BF (r = 0.9447), selecting only Radiance_AN avoids introducing excessive redundancy while preserving the key discriminative information.

In combination, these three features—SD, NDAI, and Radiance_AN—demonstrate robust statistical significance, strong domain-based justification, and minimal overlap, making them the most effective predictors for cloud detection in this dataset.

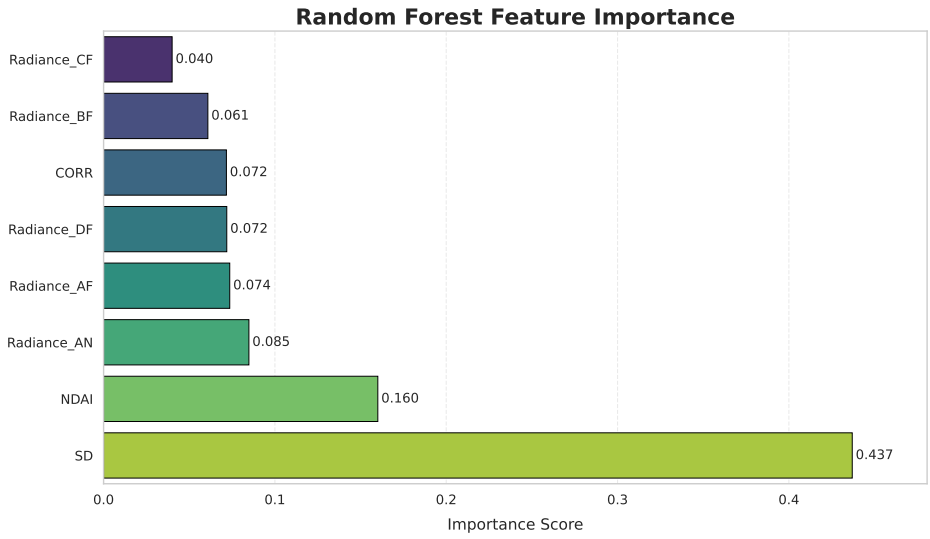


Figure 5: Feature Importance Based on Random Forest

3.2 Engineer New Features

In this study, four distinct types of features were engineered to enhance cloud detection accuracy.

First, spatial-contextual features were designed to capture local patterns around each pixel using a sliding window approach. The function `create_spatial_features_grouped` extracts neighborhood statistics—mean NDAI, standard deviation of SD, and maximum CORR—encoding local texture and variability.

Second, interaction and nonlinear transformation features were introduced to model relationships between key predictors. Interaction terms, such as NDAI_SD_Interaction and CORR_SD_Ratio, capture synergistic effects and relative contributions, while transformations like NDAI_squared and SD_log address nonlinear dependencies.

Third, angle-derived features were constructed to enhance radiometric analysis. Difference and ratio metrics, such as DF_AN_Diff and DF_BF_Ratio, quantify pairwise discrepancies and proportional relationships, while aggregate statistics summarize multi-angle radiation patterns.

Finally, texture features were extracted via entropy calculations, using a sliding window to measure spatial complexity within spectral bands. The resulting features, including NDAI_Entropy, provide insights into cloud structures by capturing randomness in pixel intensity distributions. These carefully designed features improve the model's ability to differentiate clouds from background noise, ensuring a more robust cloud detection framework.

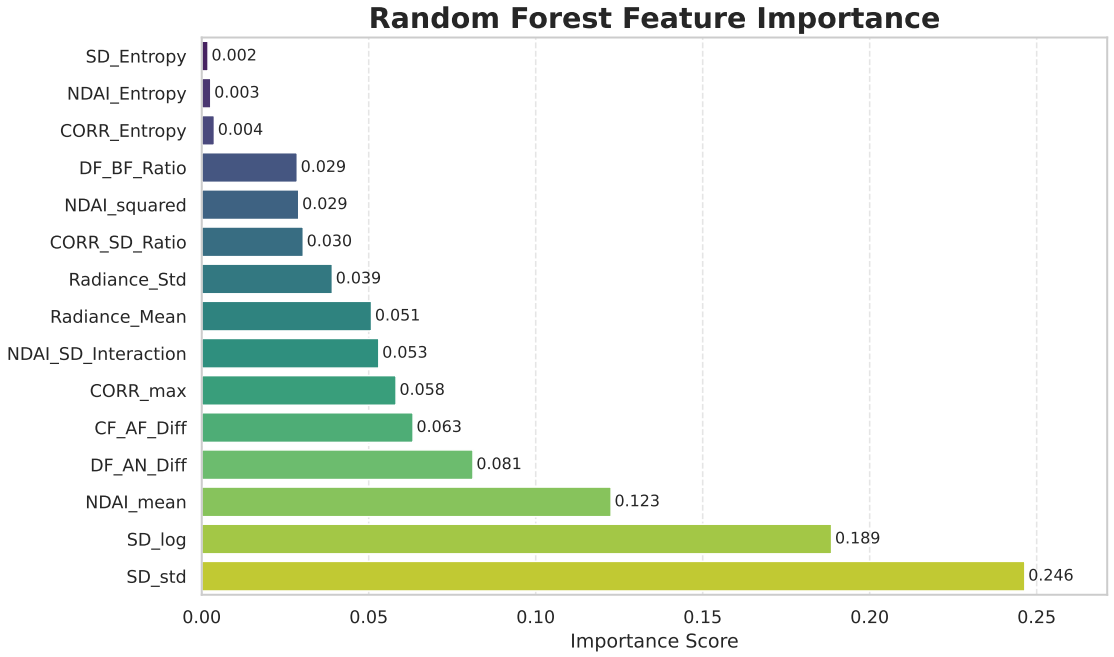


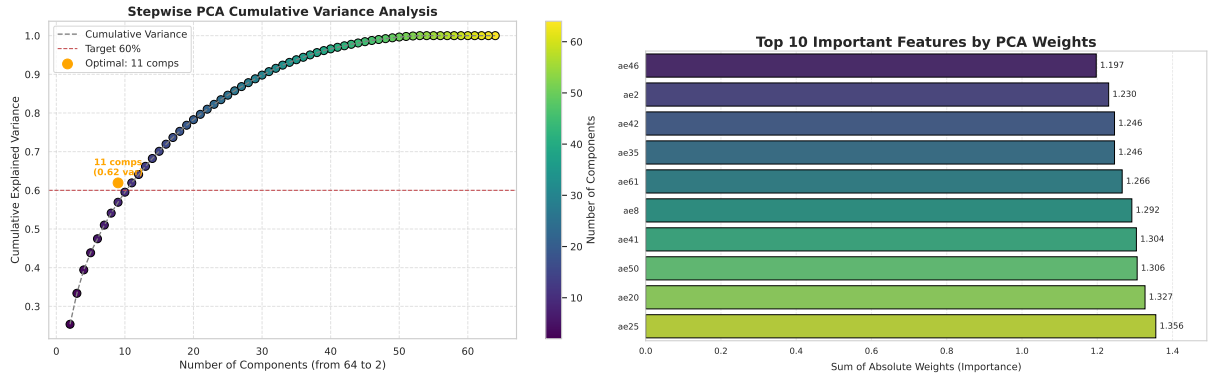
Figure 6: Feature Importance Based on Random Forest for Newly Engineered Features

The feature importance plot highlights SD_std (0.255) as the most influential predictor in the enhanced feature set, followed by SD_log (0.189) and NDAI_mean (0.084). Interaction terms such as DF_AN_Diff (0.100) and NDAI_SD_Interaction (0.058) also demonstrate substantial contributions, collectively accounting for $\approx 65\%$ of the total importance. Conversely, entropy-based features (e.g., SD_Entropy, 0.002; NDAI_Entropy, 0.003) exhibit negligible impact, aligning with earlier statistical insignificance findings.

Based on comprehensive analyses of feature importance, statistical significance, and discriminative power, we selected SD, NDAI, Radiance_AN, and NDAI_SD_Interaction as core predictors for subsequent classification tasks. These features were prioritized due to their dominant contributions in baseline and enhanced models (e.g., SD and NDAI accounting for $\approx$58% baseline importance, NDAI_SD_Interaction ranked 5th in enhanced importance) and their statistically Significant separability ($p < 1e{-}100$).

3.3 Transfer Learning

At this stage, we conducted unsupervised learning on all the images, adjusting the parameters to an embedding_size of 64, reducing the learning rate to 0.0005, and setting the batch_size to 512. To facilitate our research, we modified the code in the get_embedding file to output only three CSV files containing labeled images. Each CSV file includes the columns x, y, label, and ae0 to ae63, totaling 67 columns. Here, ae0 to ae63 represent the encoded features obtained from the unsupervised learning process. In the next step, we will perform PCA analysis on these encoded features and use them as input features for the classification model. This approach aims to reduce dimensionality while preserving the most informative aspects of the data for downstream tasks.



(a) PCA on Embeddings

(b) Top 10 Important Features

Figure 7: PCA Analysis Results from Autoencoder Embeddings

The PCA analysis showed in Figure 7a was performed on a combined dataset from three CSV files, each containing 64 features. The goal was to reduce the dimensionality of the data while retaining a specified level of cumulative explained variance. When the target variance was set to 60%, the analysis revealed that only 11 principal components were required to achieve this level of variance. This indicates that a significant portion of the data's variability can be captured by a relatively small number of components, highlighting the effectiveness of PCA in reducing dimensionality.

Figure 7b shows the top 10 important features, determined by the sum of absolute weights across all principal components, include ae25, ae20, ae50, ae41, ae8, ae61, ae35, ae42, ae2, and ae46. These features have the highest combined influence on the principal components, suggesting that they contribute the most to the variance retained in the reduced dataset. The sum of absolute weights for these features ranges from 1.197 to 1.356, indicating their relative importance in the PCA transformation. This information can be valuable for further analysis, as it identifies the key features that drive the underlying structure of the data.

4 Modeling

Three models — Random Forest, Logistic Regression, and XGBoost — were trained using the labeled data. Since only three images contained expert labels, a judgment call was made to use two of the images for training, while the third image would be used for testing. To allow for more robust cross-validation, the two training images were each split in half, creating four subsets. For each fold, three subsets would be used for training, while the fourth half was used for validation.

It is important to note that this approach introduces a risk of data leakage, since the model will train on parts of an image and then validate on another part of the same image. Although this is typically avoided, the limited labeled data required a trade-off and a judgment call was made to allow minor data leakage in exchange for more robust cross-validation. A stability analysis, discussed in the next section, will use an alternative approach that eliminates data leakage. The features used in the three models were the same. First, we used the top three features identified, which were SD, NDAI, and Radiance AN.

Next, the newly engineered feature NDAI & SD Interaction was included since it ranked fairly high on the feature importance plot and was not very highly correlated with features already included in the feature set. Lastly, the top 10 most important dimensions from the autoencoder embeddings were used as features. Therefore, each model was trained on data points, each with an input vector length of 14.

4.1 First model: Random Forest

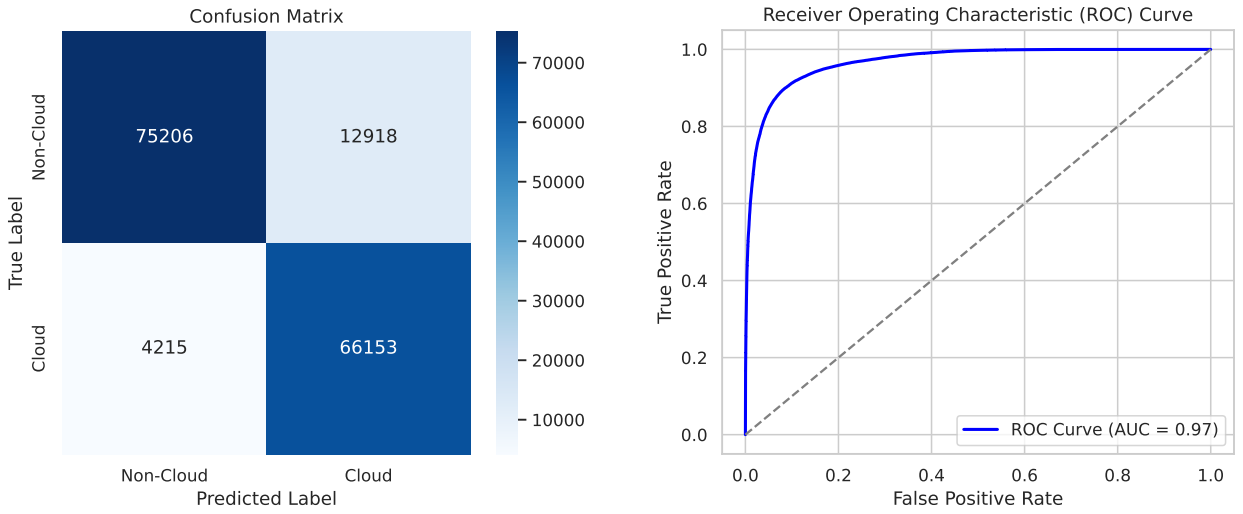
Random Forest is an ensemble learning method that constructs multiple decision trees on random subsets of the data, then makes its predictions by majority vote of the subtrees. By averaging across many trees, Random Forest reduces its variance and makes it a robust model to noise and complex feature interactions. For this lab, we trained a Random Forest binary classifier to predict cloud or non-cloud pixels using the 14 input features described above. Cross-validation was used among the four subsets of data from the first and second images and Grid Search was used to fine-tune the hyperparameters. The hyperparameters tested for this model were the number of trees and tree depth.

The final model used 100 trees and a max depth of 20. The summary statistics are shown in Table 1. The confusion matrix and ROC curve are shown in Figure 8.

The classification report indicates a precision and recall of approximately 0.84–0.95 for the classes, culminating in an overall accuracy of 0.89. The confusion matrix further illustrates that the model correctly identifies the majority of cloud and non-cloud pixels, with relatively few misclassifications in each category. Notably, the ROC curve yields an AUC of 0.97, suggesting that the Random Forest model provides a high degree of separability between cloud and non-cloud classes.

	Precision	**Recall**	**F1-score**	**Support**
-1.0	0.95	0.85	0.90	88124
1.0	0.84	0.94	0.89	70368
Accuracy			**0.89**	158492
Macro avg	0.89	0.90	0.89	158492
Weighted avg	0.90	0.89	0.89	158492

Table 1: Random Forest Classification Report



(a) Confusion Matrix for Random Forest

(b) ROC Curve for Random Forest

Figure 8: Random Forest Summary Plots

4.2 Second model: Logistic Regression

We next employed a Logistic Regression model to predict cloud presence using our selected features. Logistic Regression uses the logistic (sigmoid) function to estimate the probability that an observation belongs to a particular class, allowing us to classify each pixel as either cloud or non-cloud. In our case, if the predicted probability exceeds 0.5, the model labels the pixel as cloud. The resulting classification report in Table 2 reveals a precision and recall of approximately 0.90–0.92 for both classes, indicating balanced performance across cloud and non-cloud categories.

The confusion matrix in Figure 9a shows that the model correctly identifies most cloud and non-cloud pixels, although there are some misclassifications in both categories. Meanwhile, the ROC curve in 9b demonstrates a high degree of separability between classes, with an AUC of about 0.97, suggesting strong predictive power. This performance underscores the ability of logistic regression to effectively leverage the engineered features, even though it assumes a linear relationship in the log-odds.

It is also important to note that there are several assumptions taken when using a logistic regression model:

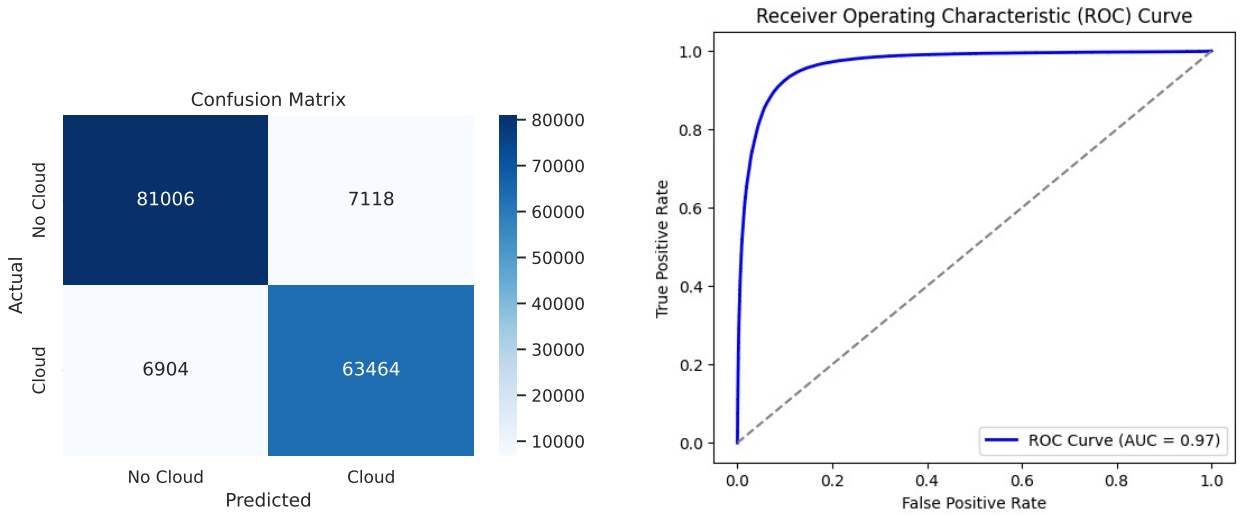
1. Linearity in the Log-Odds: Each predictor variable is assumed to have a linear effect on the log-odds of cloud presence.
2. Independence of Observations: Pixels are treated as independent samples, which can be challenging in spatial data but remains a common assumption in many image-based models.
3. No Perfect Multicollinearity: Strongly correlated predictors can degrade coefficient interpretability. Our feature engineering and selection steps help mitigate this risk.
4. Sufficient Sample Size: A larger dataset typically ensures more stable parameter estimates and reduces overfitting risks.

Overall, logistic regression offers a straightforward, interpretable approach to cloud classification, achieving competitive accuracy, recall, and precision. While it may be less flexible than tree-based

methods, the strong ROC AUC score highlights its effectiveness in exploiting the key predictors for cloud detection.

	Precision	Recall	F1-score	Support
-1.0	0.92	0.92	0.92	88124
1.0	0.90	0.90	0.90	70368
Accuracy			**0.91**	158492
Macro avg	0.91	0.91	0.91	158492
Weighted avg	0.91	0.91	0.91	158492

Table 2: Logistic Regression Classification Report



(a) Confusion Matrix for Logistic Regression

(b) ROC Curve for Logistic Regression

Figure 9: Logistic Regression Summary Plots

4.3 Third model: XGBoost

Lastly, we applied an XGBoost classifier, using the same set of engineered features. XGBoost is a gradient boosting framework that iteratively refines weak learners (decision trees for our case) and captures complex relationships in the data. This approach often excels in handling high-dimensional features.

The summary statistics are shown in Table 3. The confusion matrix and ROC curve are shown in Figure 10.

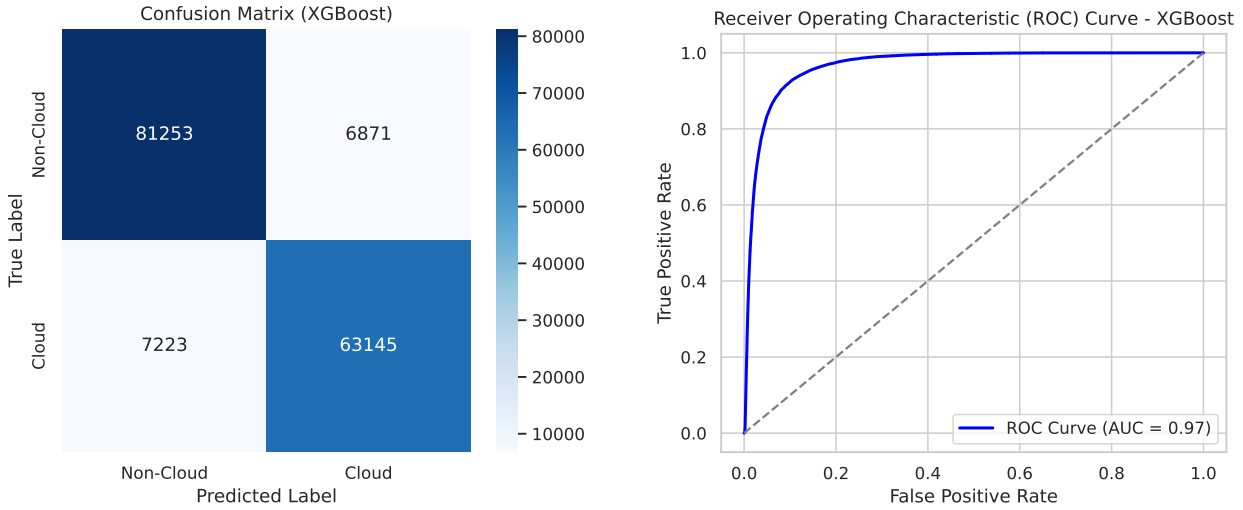
The classification report indicates a precision and recall of approximately 0.90–0.92 for both classes, culminating in an overall accuracy of 0.91. The confusion matrix further illustrates that the model correctly identifies the majority of cloud and non-cloud pixels, with relatively few misclassifications in each category. Notably, the ROC curve yields an AUC of 0.97, suggesting that the XGBoost model provides a high degree of separability between cloud and non-cloud classes.

These results underscore XGBoost's capacity to capture nuanced patterns in multi-angle radiance data and engineered features like NDAI, SD, and CORR. While hyperparameter tuning can be

more involved than with simpler models, the strong performance highlights the effectiveness of gradient boosting in tackling the inherent complexity of polar cloud detection.

	Precision	Recall	F1-score	Support
-1.0	0.92	0.92	0.92	88124
1.0	0.90	0.90	0.90	70368
Accuracy			**0.91**	158492
Macro avg	0.91	0.91	0.91	158492
Weighted avg	0.91	0.91	0.91	158492

Table 3: XGBoost Classification Report



(a) Confusion Matrix for XGBoost

(b) ROC Curve for XGBoost

Figure 10: XGBoost Summary Plots

4.4 Best Model Analysis

Among the three classification models, the Logistic Regression model was determined as the best-performing model for cloud detection. While all three models achieved high classification accuracy and AUC scores, Logistic Regression demonstrated a strong balance between computational efficiency and classification performance.

4.4.1 Performance Evaluation

The Logistic Regression model yielded an accuracy of 0.91, with a precision and recall of approximately 0.90–0.92 for both cloud and non-cloud classes. The model classified most cloud and non-cloud pixels, with relatively low misclassification rates. The high AUC score of 0.97 indicates a strong degree of separability between the two classes, comparable to the Random Forest and XGBoost models.

One advantage of Logistic Regression over the tree-based models is its ability to provide directly interpretable feature coefficients, making it easier to assess the relative importance of predictors in

cloud classification. Additionally, logistic regression's lower computational complexity makes it a more efficient choice for large-scale satellite image classification tasks.

4.4.2 Comparison with Other Models

To systematically compare model performance, Table 5 presents key evaluation metrics for all three models.

Table 4: Comparison of Classification Models

Model	Accuracy	AUC Score	Computational Complexity
Random Forest	0.89	0.97	High
Logistic Regression	0.91	0.97	Low
XGBoost	0.91	0.97	Medium

While all models performed similarly in terms of accuracy and AUC, Logistic Regression demonstrated key advantages in efficiency and computational simplicity. Random Forest and XGBoost required significantly more computational resources and hyperparameter tuning, while providing only marginal improvements in classification performance.

4.4.3 Key Findings

1. Logistic Regression achieved a strong balance of accuracy, precision, and recall, performing on par with tree-based models.
2. Unlike Random Forest and XGBoost, Logistic Regression is computationally efficient, making it ideal for large-scale cloud classification.
3. The model provides meaningful feature importance insights, aiding in understanding the influence of NDAI, SD, and Radiance_AN on cloud classification.
4. Adjusting the classification threshold (e.g., lowering from 0.5 to 0.4) could reduce false negatives, improving cloud detection.

4.4.4 Conclusion

Overall, Logistic Regression was selected as the best model due to its high accuracy, computational efficiency and straightforward implementation. While tree-based models provided similar classification performance, they required more extensive tuning and lacked the transparency offered by logistic regression. Further optimization, such as refining the decision threshold and incorporating spatial dependencies, could further enhance classification performance.

5 Post-Hoc EDA

The post hoc exploratory data analysis provides key insights into the performance of the model, particularly in distinguishing cloud pixels from non-cloud pixels. Three major analyses were performed: (1) a histogram of predicted cloud probabilities, (2) a spatial distribution of low-confidence cloud predictions, and (3) a comparison of feature distributions among correctly classified and misclassified samples.

5.1 Predicted Probability Distribution

The histogram of predicted probabilities (Figure 11) shows a bimodal distribution, with the majority of pixels assigned probabilities close to 0 or 1. This suggests that the model is generally confident in its classifications. However, there is a non-negligible proportion of pixels with probabilities in the range 0.3 to 0.5, indicating uncertainty in these classifications. Since the default classification threshold is set at 0.5, many cloud pixels with probabilities below 0.5 are misclassified as "No Cloud,", leading to a high false negative rate.

This suggests that the model's thresholding decision could be improved. A lower threshold (e.g., 0.4 or 0.3) may allow more uncertain cloud pixels to be classified correctly, reducing false negatives. Further analysis is required to determine the optimal threshold.

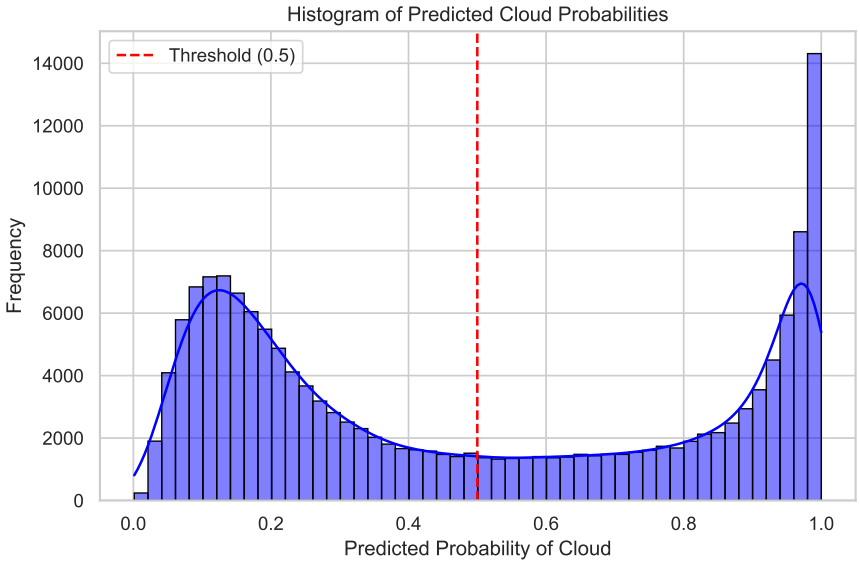


Figure 11: Histogram of predicted cloud probabilities

5.2 Spatial Distribution of Low-Confidence Predictions

The spatial distribution of low-confidence cloud pixels (Figure 12) highlights regions where the model struggles to distinguish between cloud and non-cloud pixels. The red-marked pixels represent predictions with probabilities between 0.3 and 0.6, where the model is uncertain. These uncertain pixels are not randomly distributed; rather, they appear in structured patterns, particularly along cloud boundaries and within certain atmospheric formations. This suggests that the model may not capture variations in cloud structures.

Moreover, a comparison with the expert-labeled cloud masks indicates that many actual cloud

pixels fall within this low-confidence range. This further supports the hypothesis that the model is under-predicting clouds due to a high classification threshold.

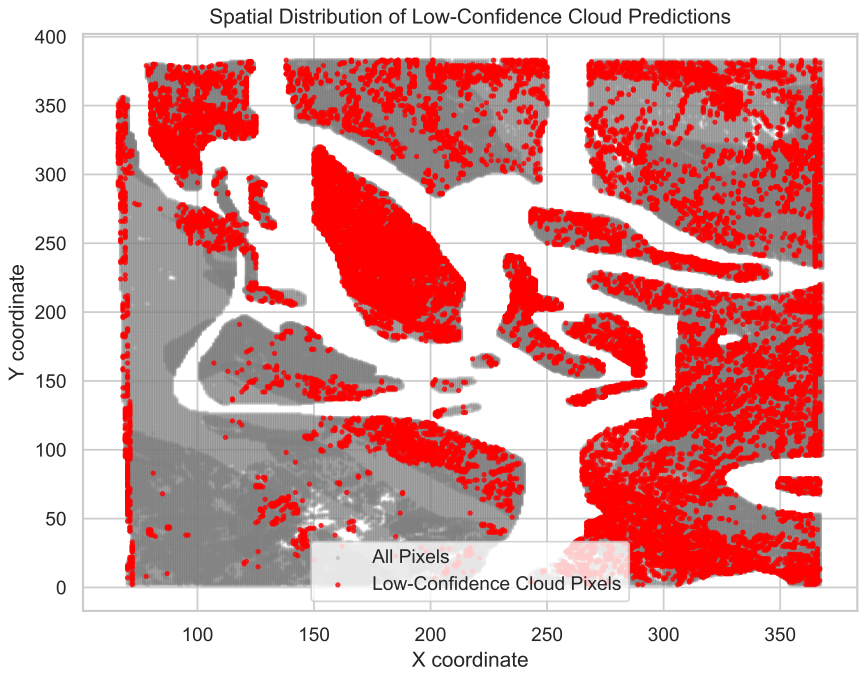


Figure 12: Spatial distribution of low-confidence cloud predictions

5.2.1 Feature Distributions for Correct vs. Misclassified Samples

To further analyze misclassified patterns, we examined the distributions of NDAI, SD, and Radiance_AN for correctly classified, false negative, and false positive samples (Figure 13).

1. NDAI Distribution: False positives show a slightly broader spread compared to correct classifications, suggesting that NDAI alone may not be sufficient for distinguishing clouds from non-clouds, particularly in cases where reflectance differences are subtle.
2. SD Distribution: False positives occur more frequently in high-SD regions, indicating that texture variation contributes to cloud misclassification. In contrast, false negatives are associated with lower SD values, implying that clouds with uniform texture are more likely to be mistaken for background.
3. Radiance_AN Distribution: False negatives tend to have slightly higher radiance values, while false positives show a broader distribution, suggesting that some cloud pixels with atypical radiance levels may be difficult for the model to distinguish from non-cloud regions.

These findings indicate that certain feature properties contribute to misclassification errors. Specifically, false negatives are related to lower SD values and higher radiance levels, while false positives are more likely in regions with high SD variation. This suggests that introducing feature interactions (e.g., NDAI $\times$ SD) or spatial modeling techniques could improve classification performance.

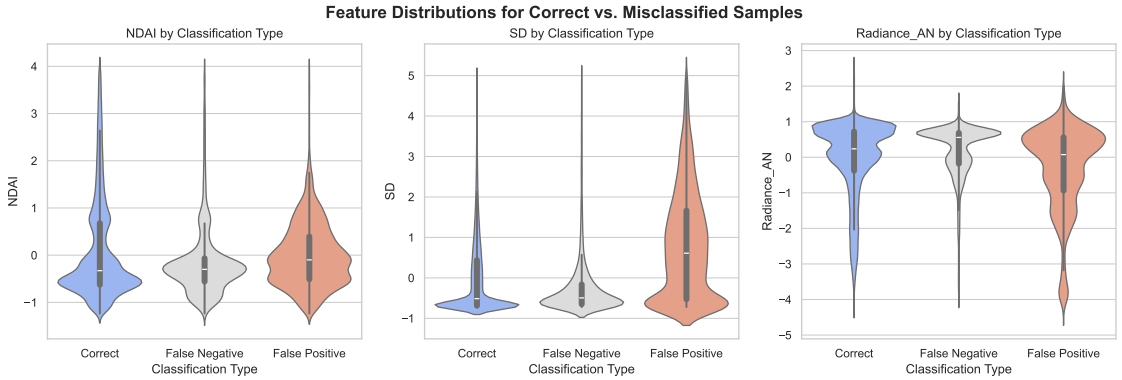


Figure 13: Feature distributions for correct vs. misclassified samples

6 Stability check & Improvement

6.1 Stability check

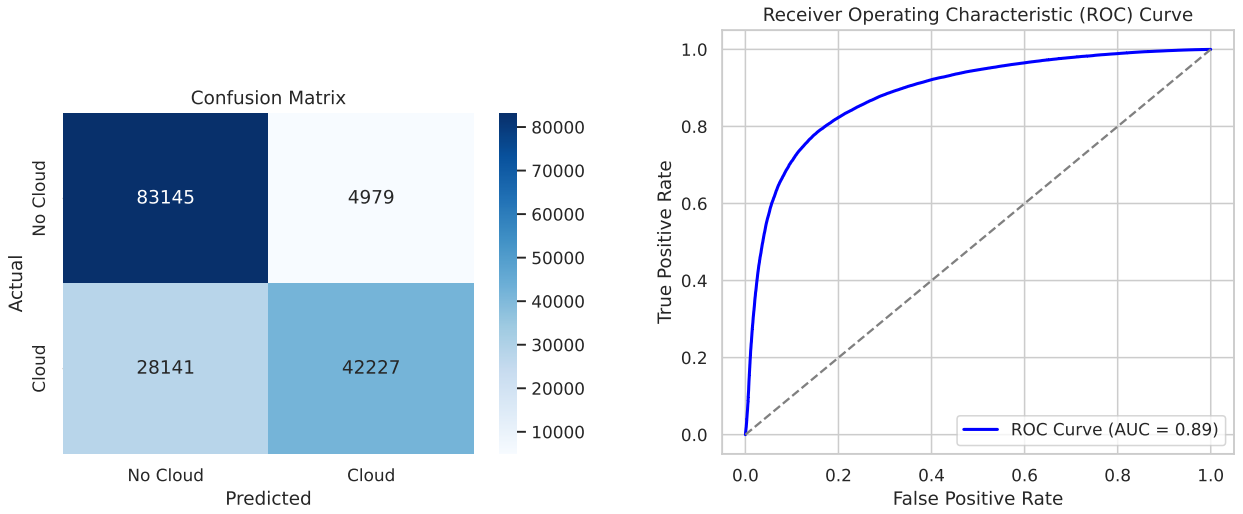
Although our three models performed well, there remains the question of how stable these results are. This section focuses on the Logistic Regression model to assess its stability.

In the original setup, the model was trained using cross-validation on the first two images, where each image was split into two halves, allowing for 4-fold cross-validation. However, as mentioned above, this approach introduces some data leakage, since halves of the same image were used in both the training and validation sets. To address this, and perform a stability analysis at the same time, a key change was made. Instead of splitting the raining images into halves to enable robust cross-validation, we will train on one whole image and validate on the other. This will eliminate data leakage, but results in a simpler 2-fold cross-validation approach. Aside from this change, the model training pipeline remains identical to the original setup.

The classification reports for the original and stability-tested models are shown in Table 5, while the confusion matrix and ROC curve for the new model are shown in Figure 14.

	Logistic Regression (Original)			**Logistic Regression (New)**		
Class	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
-1.0	0.92	0.92	0.92	0.75	0.94	0.83
1.0	0.90	0.90	0.90	0.89	0.60	0.72
Accuracy			**0.91**			**0.79**
Macro avg	0.91	0.91	0.91	0.82	0.77	0.78
Weighted avg	0.91	0.91	0.91	0.81	0.79	0.78

Table 5: Comparison of Original Logistic Regression and Stability-tested Logistic Regression Classification Reports



(a) Confusion Matrix for LR (Stability-tested)

(b) ROC Curve for LR (Stability-tested)

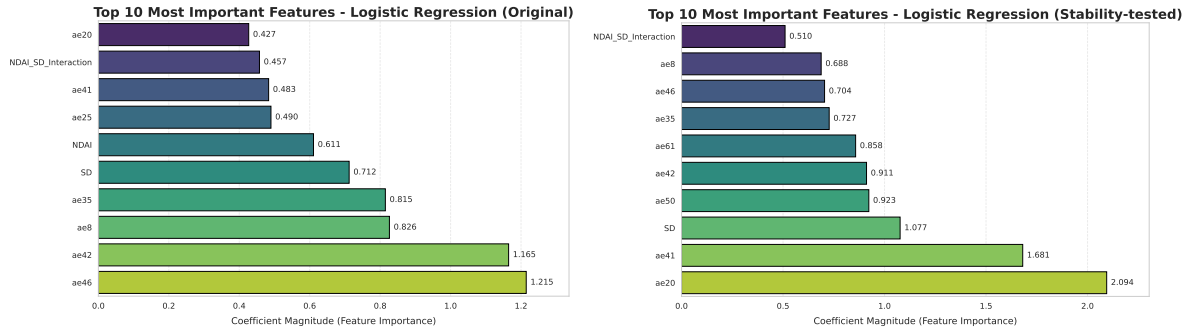
Figure 14: Stability-tested Logistic Regression Summary Plots

6.1.1 Observations and Findings

By altering the training data, we observe a drop in the weighted average F1-score, highlighting its reduced performance. The confusion matrix also reveals a much higher rate of false negatives and the ROC curve suggests a slight decline in model robustness.

We also compare the top ten most important features in the original model and stability-tested model in Figure 15. Despite differences in overall performance, we see that several features consistently appear in the top ten features across both models. In fact, eight of the top ten features from the original Logistic Regression are also top ten features in the stability-tested model.

These results suggest that the feature importance remains relatively stable and that the model consistently identifies key predictors for cloud classification. Despite this, the fine-tuned parameters and feature weights still significantly impact the model accuracy.



(a) Top 10 Features for LR (Original)

(b) Top 10 Features for LR (Stability-tested)

Figure 15: Top 10 Important Features for Original and Stability-tested Logistic Regression

Overall, while the feature rankings remain stable, the drop in accuracy suggests that the model is

sensitive to training data selection and data leakage.

6.2 Improvement

While the current models (Logistic Regression, Random Forest, and XGBoost) demonstrate strong performance in polar cloud detection (AUC $\approx$ 0.97 and balanced precision/recall of 0.90–0.92), several avenues for improvement could further enhance robustness and accuracy:

1. Threshold and Class Imbalance: Adjusting classification thresholds (e.g., prioritizing recall for cloud pixels) and addressing residual class imbalance via SMOTE or weighted loss functions could reduce false negatives observed in confusion matrices (e.g., 6,904–12,918 FN). This is critical for minimizing operational risks in cloud detection.

2. Feature and Hyperparameter Optimization: Enhancing spatial-context features (e.g., neighboring pixel statistics) and refining autoencoder-derived embeddings could better capture cloud patterns. Advanced hyperparameter tuning (Bayesian optimization) and model ensembling (stacking XGBoost with RF) may further exploit complementary strengths in non-linear decision boundaries.

3. Architectural Innovation: Transitioning to deep learning architectures (e.g., Vision Transformers, U-Nets) could bypass handcrafted feature limitations by directly modeling raw spectral-spatial data. Coupled with systematic error analysis (e.g., SHAP values), this would address edge cases (e.g., thin clouds vs. bright surfaces) while maintaining scalability for large satellite datasets.

7 Results

Our analysis compared three machine learning models–Random Forest, XGBoost, and Logistic Regression–to classify the presence of clouds at the pixel level. Given the limited availability of expert-labeled images, we implemented a cross-validation strategy that maximized the amount of training data, while keeping data leakage to a minimum. The initial training set-up involved splitting two labeled images into four subsets for 4-fold cross-validation, with the model evaluation being performed on the third labeled image.

The logistic regression model achieved an accuracy of 0.91 and an AUC score of 0.97. This matched the tree-based models, but since logistic regression offers more interpretability and is more computationally efficient, it was selected for further post-hoc EDA and stability testing.

The post-hoc EDA revealed several key insights. Figure 11 indicates that the model was generally confident in its classifications. Spatial mapping in Figure 12 showed that misclassifications appeared to happen more often among cloud boundaries. Figure 13 also highlighted patterns among misclassified pixels. False negatives were associated with lower SD values and higher Radiance AN, whereas false positives tended to occur in regions with high SD variation.

For stability testing, the training approach was slightly altered, resulting in using a two-fold cross-validation and no data leakage. Stability testing highlighted that the Logistic Regression model is sensitive to how the training set is constructed, including a large increase in the false negative rate. However, despite this, the most important features remained largely consistent across both models. This suggests that feature importance is stable, although the model's ability to generalize across different images may be limited based on the training data composition.

8 Conclusion

Overall, the Logistic Regression model provided the best trade-off between accuracy, interpretability, and computational efficiency. The post-hoc analysis identified spatial dependencies as one area for areas for further improvement. Stability testing revealed that while the model is sensitive to training data, the most important features remained consistent, indicating a strong foundation for model refinement and deployment in polar cloud detection tasks.

9 Bibliography

[1] Shi, T., Yu, B., Clothiaux, E. E., & Braverman, A. J. (2008). Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies. Journal of the American Statistical Association, 103(482), 584–593.

[2] Erika Russi. "What Is XGBoost?" Ibm.com, 9 May 2024, www.ibm.com/think/topics/xgboost.

[3] Stojiljković, Mirko. "Logistic Regression in Python – Real Python." Realpython.com, realpython.com/logistic-regression-python/.

[4] Breiman, Leo, and Adele Cutler. "Random Forests - Classification Description." Berkeley.edu, 2019, www.stat.berkeley.edu/ breiman/RandomForests/cc_home.htm.

A Academic honesty

A.1 Statement

We affirm our commitment to upholding the highest standards of academic integrity in all aspects of our work. We pledge to properly cite all sources, avoid plagiarism, and ensure that our contributions—whether individual or collaborative—are original and transparent. We will complete assignments independently when required and collaborate ethically, ensuring fair representation of all contributions.

We recognize that academic honesty is essential to maintaining trust and intellectual rigor within our academic community. By adhering to these principles, we strive to foster an environment of respect and integrity, understanding that any breach of these standards undermines the value of our education and the credibility of our work.

A.2 LLM Usage

Coding

For the coding component of Lab 2 involving LLM usage, we referred to the modeling GitHub repository as a general reference. Large Language Models (LLMs) such as ChatGPT/DeepSeek were primarily utilized for debugging purposes, particularly when encountering errors or warning messages. Additionally, we consulted these models for suggestions on optimal parameters and settings to enhance the clarity and organization of our plot visualizations. However, the core coding structure and algorithmic design were primarily developed through our own discussions and understanding.

Writing

For the writing component of this lab, we primarily utilized Large Language Models (LLMs) such as ChatGPT/DeepSeek to assist with grammar checking and language refinement. Despite thorough peer reviews by all four group members, it can be challenging to identify minor grammatical or stylistic errors, so LLMs served as a helpful tool for ensuring clarity and professionalism in our writing. Aside from these language enhancements, the content and conceptual understanding presented in the report were entirely based on our own knowledge, discussions, and collaborative efforts throughout the lab.