Problem:
Since 1990 more than 7000 civilians and 5000 members of the armed forces have been killed or injured by landmines and explosive remnants of war (ERW) in Columbia. The United Nations Mine Action Service (UNMAS) has been dedicated to clearing these hazardous items since 2006. To assess the impact of these mine action interventions, their current approach is to manually extract features from raw data, such as digitizing footprints or generating land cover maps. We propose to use machine learning techniques to improve efficiency by building machine learning models to identify and classify new buildings, roads, and land cover in pre-selected areas of interest.

Methods:
We feature-engineered satellite images of areas of interest, and built ML models, including CART, Logistic Regression, Random Forest, and Xgboost, to identify buildings, roads, and land cover.

Findings:
Our models return predictions with high accuracy of above 90%. Due to class imbalance, the precision of our models is lower. Below is a demonstration of our classification models.

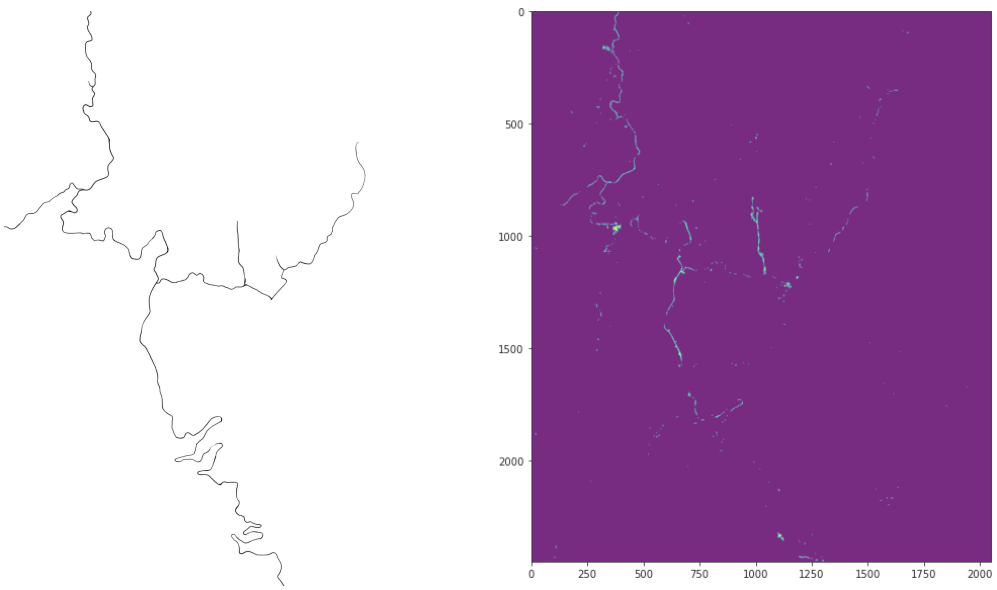


Figure 1: Comparison of Road in Reality and XGBoost Model Prediction (AOI2)

Impacts:
1. Our models can be used as tools to narrow down the possible area of buildings, roads, and land cover. They automate manual processes with high accuracy. And they can be generalized to the country level with new imagery inputs.
2. Our models pave the way for later stages of the project, such as comparing pre-clearance and post-clearance images. It is a necessary step towards the greater objective of assessing the impacts of mine intervention actions with causal inference analysis.





Land Release Impact Assessment in Columbia

Shurui Cao
Daniel Chung
Xinyao Han
Zehao (Andy) Zhao

Problem:

Since 1990, more than 7000 civilians and 5000 members of the armed forces have been killed or injured by landmines and explosive remnants of war (ERW) in Columbia. The United Nations Mine Action Service (UNMAS) has been dedicated to clearing these hazardous items since 2006. To assess the impact of their mine action interventions, UNMAS currently manually extract features from raw data, such as digitizing footprints or generating land cover maps, which can be time-consuming. We propose to use machine learning techniques to improve the efficiency in comparing pre-clearance and post-clearance images to measure the impacts. Our goal is to build and train ML models to identify and classify new buildings, roads, and land cover in pre-selected areas of interest.

1 Data preprocessing

1.1 Data Description

Our dataset contains 5 satellite images of areas of interest in San Carlos and shapefiles with buildings labels. From each satellite image, we are able to extract band information, which are image layers created with a range of specific wavelengths of light. In our case, satellite images consist of 8 bands. The building shapefiles contain polygons with types of buildings (house, facility, etc.), from which we can extract coordinate information. Because roads and land cover did not have label shapefiles, we created our own label jpg files with pixel dimensions identical to the original satellite image. This allowed us to convert the jpg files to labels in the form of binary np arrays, which is how we extracted coordinate information.

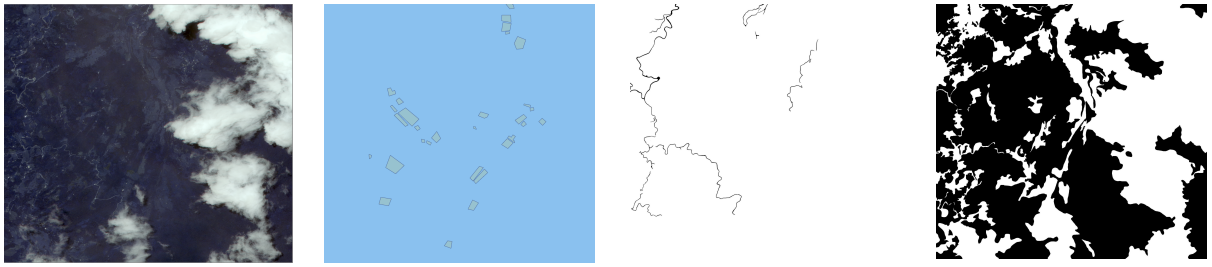


Figure 1:
AOI1 Satellite Imagery, AOI1 Building Labels, AOI1 Road Labels, AOI1 Land Cover Labels

1.2 Slicing

Building Classification

We sliced each satellite imagery into a grid of smaller patches, each containing some pixels. For consistency, we used the same size for all patches in each imagery. However, it was difficult to find a common divider for the dimensions of all 5 images. Hence, we selected 20x20 as this size provided enough information without taking up too much computing power. In this case, when each imagery is converted into patches, it is cropped slightly and hence lost some information on the rightmost side. After we had a grid of patches, we extracted the following information from each patch:

1. Coordinates of the patch;
2. The mean value of band information of pixels in each patch for each band;
3. The standard deviation of band information of pixels in each patch for each band.

We then found the intersection of building polygons with each patch and computed the area of the intersection.

Road and Land Cover Classification

Because roads are more narrow, we decided to use a patch size of 1x1, which is equivalent to classification at the pixel level. Pixel-level classification allowed for the side-stepping of band averaging, as band information for pixels already summarized the contents of the patch.
Although we originally intended to use larger patch sizes to classify land cover, which is structurally larger than both roads and buildings, we found that we could classify land cover at the pixel level as well without prohibitively high runtime costs, so we also decided on a 1x1 patch size for land cover.

2 Modeling

2.1 Train & Test Dataset Preparation

Building Classification:

We prepare the data for the modeling by splitting all the five AOIs for train and test purposes, and choosing the threshold for the fraction of building within each grid. We analyze that the first AOI has the most labeled buildings, and the second AOI contains no labeled buildings. We drop the second AOI and use the first, third, forth AOI as train data, and use the fifth AOI as the test data. This train and test split of AOIs ensure that the distribution of labeled buildings are consistent.
We decide the threshold of building fractions to generate the building labels based only on the train set. The fraction of buildings has a maximum value of 0.8 and a minimum value of close to 0. We define those grids with a fraction of buildings close to 0 to be no buildings, and we use a threshold that maintains 80% of the buildings, which is 0.02.
The processed data is an aggregated array of grids with *x_min, x_max, y_min & y_max* coordinates, average and standard deviation of the 8 bands, the fraction of building within the grids, and a building label for each grid. We apply the same threshold on the test set due to consistency. The train set contains 13259 grid entries, and the test set contains 14495 grid entries.

Road and Land Cover Classification:

To diversify the training set for different road and land cover contexts, we trained our models on AOIs 3, 4 and 5, which contained bodies of water, clouds, rivers, and ample vegetation. We had found that models trained on images without bodies of water classified them as roads because of their reflectivity, which is why we included AOI5, which features a lake, in the set of training images.
Because our road and land cover models used 1x1 patches, each row of processed data is one pixel of the original satellite image, indexed by x and y value. The spectral data from all 8 bands is recorded for each row as well as a label for whether the pixel depicts roads or land cover respectively. Originally, we included band combinations such as TM3/TM4 and TM3/TM5 to distinguish between urban surfaces and surrounding vegetation, but these did not contribute to model performance, so we dropped them. The total train set for both road and land models contains over 12 million entries, each its own pixel in a training set consisting of three satellite images.

2.2.1 Rebalancing (Building Classification)

The dataset is a highly imbalanced dataset, with only 0.02% of the training set containing labeled buildings. We approach the imbalance problem using: 1) Rebalancing techniques, 2) Weighted decision trees for class imbalance.

1. Rebalancing Techniques

   With those characteristics of the dataset in mind, we choose oversampling methods, which duplicate the minority entries or synthesize new entries from the examples in the minority class. We apply Random Oversampling and SMOTE in our training dataset.
   Random Oversampling is the simplest oversampling method, which randomly duplicates entries of the minority class. SMOTE is a synthetic minority over-sampling technique, it takes the difference between a chosen sample and its nearest neighbor, multiply the difference by a random number between 0 and 1, add this difference to the sample to generate a new synthetic example in feature space and continue on with next nearest neighbor up to user-defined number.

2. Weighted Decision Trees
   We build a cost-sensitive tree by adjusting the weight of each class, which is proportional to the cost of misclassifying the class to which the entry belonged. We assign a small weight to the majority class and a large weight to the minority class, which will lower the purity score of a majority class entry.

2.2.2 Adding Band Combinations (Road and Land Cover Classification)

In a bid to boost the performance of our models, we briefly considered the inclusion of spectral band combinations, which are known to capture more spectral information than data from individual bands. In particular, we considered well-known spectral ratios, which we calculated by dividing the value of one band by the value of another. These ratios are as follows:

- TM3/TM4: distinguishes barren land and urban area
- TM4/TM3: distinguishes vegetation, water, and cropland
- TM5/TM7: distinguishes land and water
- TM2/TM3: distinguishes cropland and barren land
- TM3/TM2: distinguishes forest and cropland
- TM4/TM5: distinguishes land and water
- TM5/TM4: distinguishes water, forest, and barren land
- TM3/TM5: distinguishes barren land and urban areas
- TM7/TM2: distinguishes forests and cropland

Adding these ratios to the 8 spectral bands already in our processed data, this made the total number of features 8 spectral bands plus 9 band combinations. Unfortunately, band combinations did not improve the performance of our models (AUC and accuracy remained unchanged), so our final models did not include band combinations to preserve training runtime.

2.3 Evaluation Methods

We use Classification Accuracy, Receiver-Operator Curve (ROC Curve) and Area Under the Curve (AUC) and Confusion Matrix as evaluation criteria. Classification Accuracy is the easiest classification metric to understand, while it does not indicate the type of error the model is making,

and is problematic when the class is imbalanced. Confusion Matrix shows the error the model made as well as the type of errors. AUC is the measure of the ability of the model to distinguish between classes and is used as a summary of the ROC curve. In our case, we compare models based on AUC. A higher AUC indicates that the model is better at distinguishing between the positive and negative classes.

2.4 Models

We start with basic logistic regression and CART model. For building classification, resampled data is used for these models. We improve on these models by utilizing random forest, which is a kind of ensemble model that combines different decision trees in order to optimize the result. In this paper, we are using scikit-learn *RandomForestClassifier*. We used scikit-learn's random search CV to run 5 epochs between 1 to 150 estimators in order to get the best results. Gradient Boosting is also used for the same purpose. Gradient Boosting is a technique for regression problems, which produces a prediction model in the form of an ensemble of weak prediction models, in our case, decision trees. In this paper, we are using scikit-learn's *GradientBoostingRegressor* by scikit-learn. We also used scikit-learn's random search to run 20 epochs between 1000 to 5000 estimators, a random subsample rate from 0 to 1, and the learning rate of 0.01.
We use the second resampling methods mentioned above by adjusting the class weights of decision trees and compare the results with the previous resampling methods.

3 Results Analysis

3.1 Modeling AUC Results

Building Classification

We compare the AUC results of CART, Logistic Regression, Random Forest and XGBoost, which can be seen from the charts below. The decision tree model has the lowest AUC, and Logistic Regression has the highest AUC, followed by Random Forest with the second highest AUC. One of the potential reasons is that CART, Random Forest and XGBoost overfits and the simple logistic regression gives a good performance.

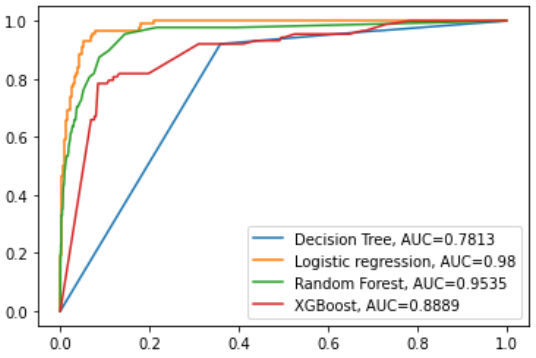


Figure 3: Comparative Performance of Building Classifiers (ROC and AUC)

Road Classification

These 4 methods had different results when applied to road classification. Here, as in the building models, CART has the lowest AUC, but XGBoost has the highest AUC at 0.93. Unlike with building classification, logistic regression performed worse than XGBoost, not better.

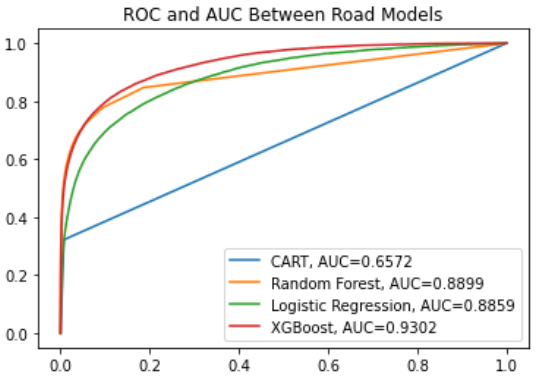


Figure 4: Comparative Performance of Road Classifiers (ROC and AUC)

Land Cover Classification

Land cover models performed similarly to road models, with CART yielding the lowest AUC and XGBoost yielding the highest (0.9102). It should be noted that with land cover, a random forest had a nearly identical AUC (0.9101) to that of XGBoost, which was not the case for roads or buildings.

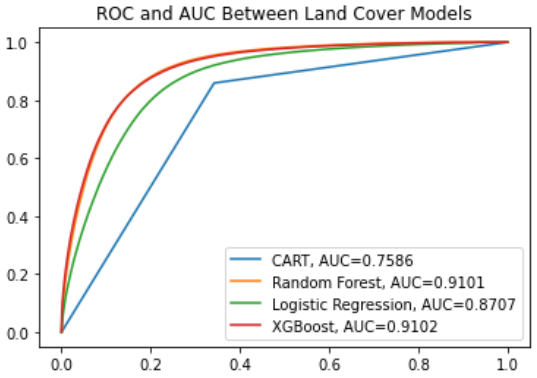


Figure 5: Comparative Performance of Land Cover Classifiers (ROC and AUC)

3.2 Important Features

Building Classification

Based on the logistic regression results in the appendix, we conclude that all the band information we use (mean and standard deviation of different bands) contributes to the prediction of building labels, where they don't differentiate with each other in feature importance.

Road Classification

An analysis of the logistic regression model revealed that bands 2 and 6 had the largest positive coefficients, suggesting that they are most significant for identifying roads. Indeed, it is currently held that band 2 makes urban surfaces appear brighter, corroborating our finding that higher values of band 2 raise the probability of that pixel being part of a road. Why band 6 was so significant to the model's predictions is something that is not as apparent in the literature, however.

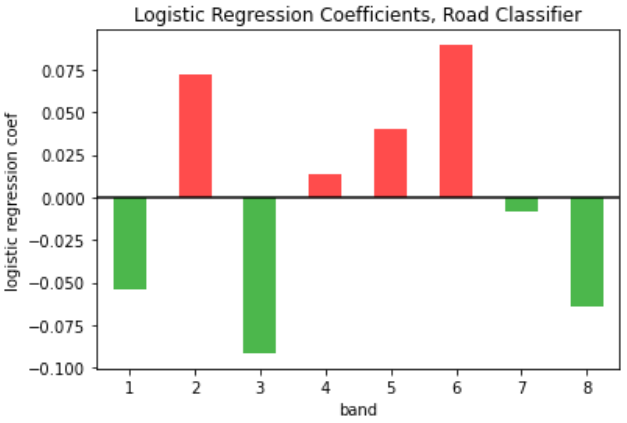


Fig 6. Logistic Regression Coefficients for Road Classifier

Land Cover Classification

The logistic regression coefficients for land cover are much different, as expected. Here band 1 has the greatest positive coefficient, suggesting it is most significant for identifying denser vegetation. Indeed, band 1 is known to differentiate between soil and heavy vegetation, which would explain why it has the greatest positive impact on the predicted probability of a pixel being part of land cover.

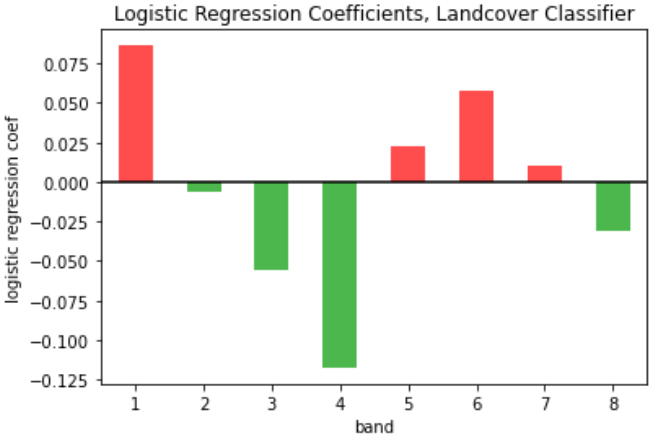


Fig 7. Logistic Regression Coefficients for Land Cover Classifier

3.3 Visualization

Building Classification

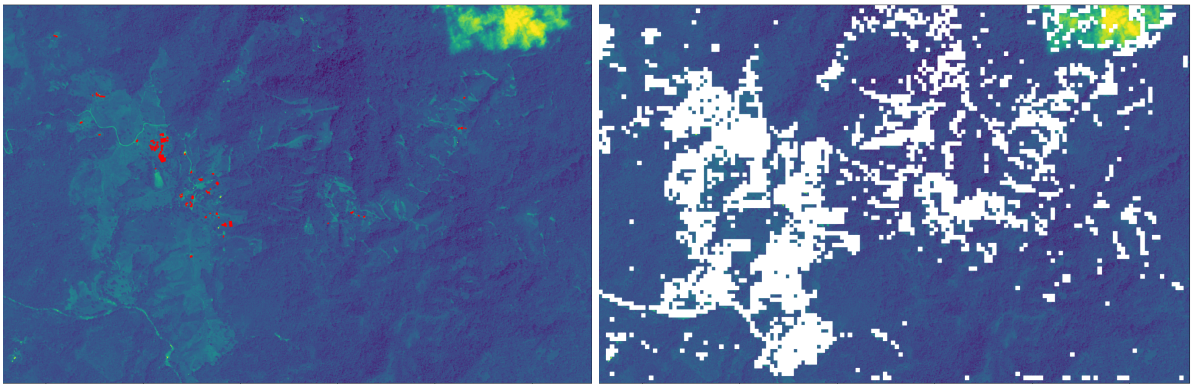


Figure 8: Comparison of Positive Labels in Reality and Model Classification

From the comparison, we can see that the classification models provide a large range of possible buildings. One interesting discovery is that the model tends to give positive predictions on areas that are visually brighter in color.

3.4 Human Cluster

We define human clusters based on our predicted building labels. Because we aggregate pixels into grids, where each pixel contains buildings, we define human clusters as grids where a larger proportion of buildings are found. We visualize the results in the following figure:

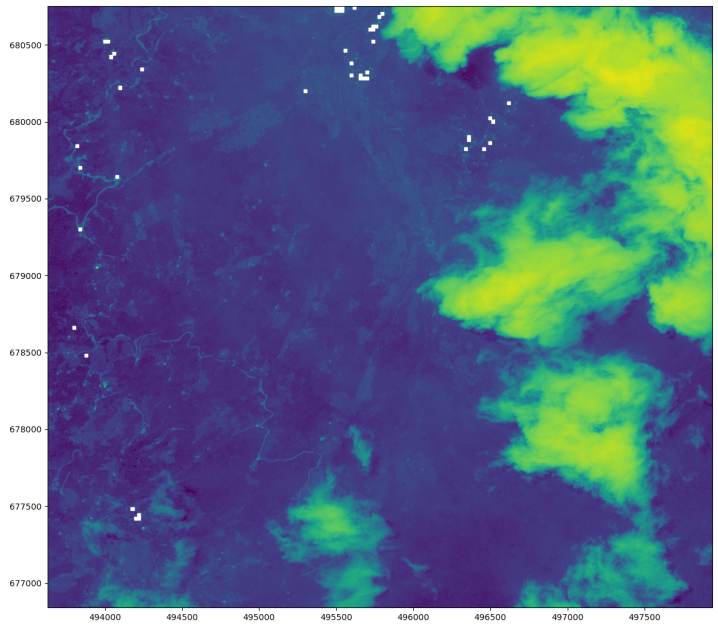


Figure 9: Human Clusters (marked in white grids)

Road Classification

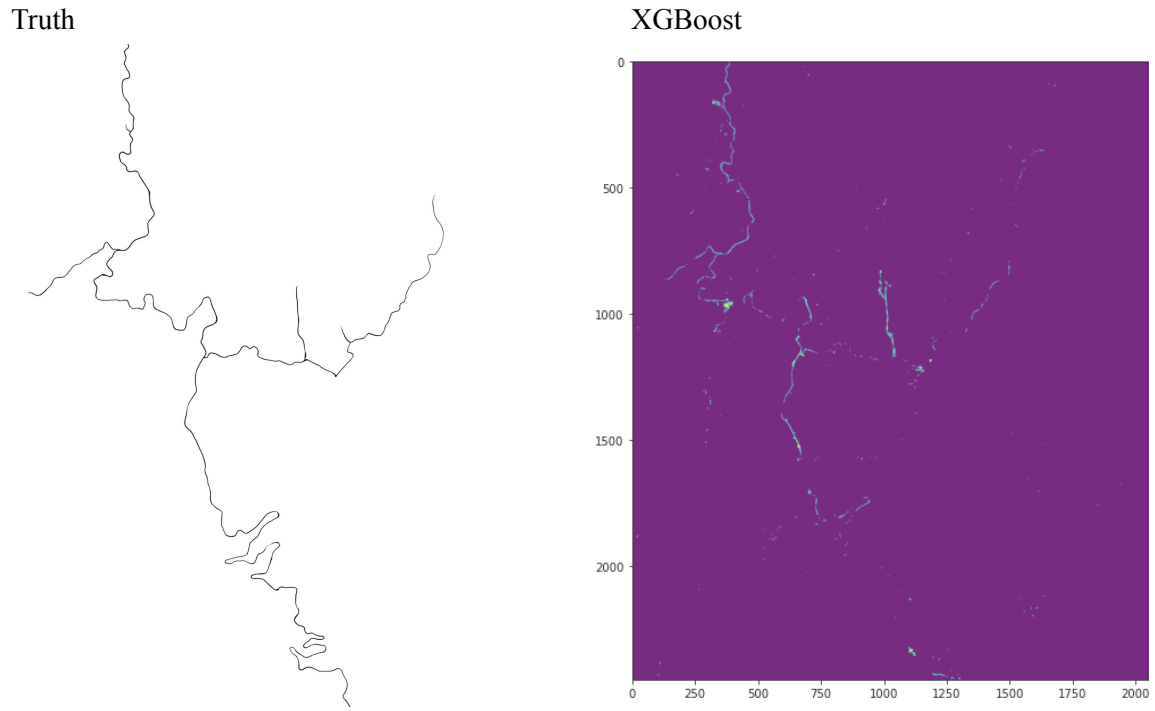


Figure 10: Truth mask for AOI2 vs road predictions from XGBoost (yellow=roads), the best of the four models

To visualize a prediction, we classify every test pixel and rearrange it into a two dimensional array we can plot. As can be seen, our models were able to pick up on most of the roads in this sample AOI, with some minor improvements in coverage to be made. We discovered in AOI1, however, that our road models often misclassify the edges of clouds as roads as well due to their reflectivity.

Land Cover Classification

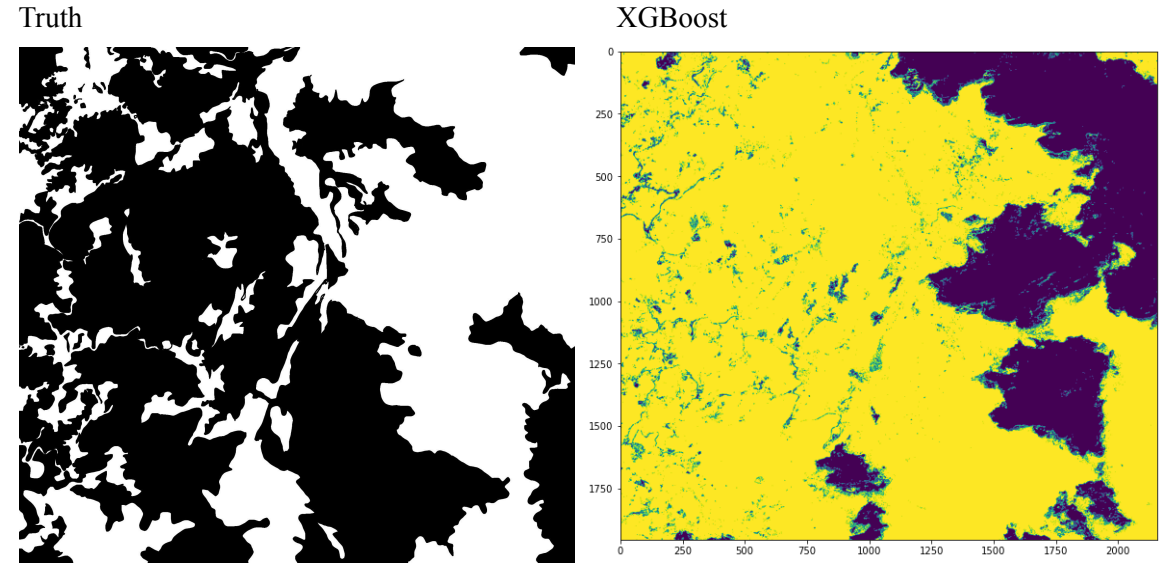


Figure 11: Truth mask for AOI1 (black=landcover) vs XGBoost (yellow=landcover), the best of the four models

With land cover predictions, we observe that our approach cannot circumvent clouds, as these are highly reflective bodies that hide true land cover beneath. This is apparent here in our land cover predictions for AOI1, where cloud patches interfere with predictions in the right half.

4 Impacts and limitations

Each of our models can be used as a tool to narrow down the possible area of buildings, roads, and landcover in the initial stages of identification for future investigations into the effect of mine clearance on Colombian areas of interest. Instead of manually examining the satellite imagery and labeling the buildings, we can apply the models and only examine the area with predicted positive labels, saving time. Each model has unique limitations, however, which are discussed as follows.

Building Classification

Due to our preprocessing approach of slicing and labeling, we treat every building as the same. While this approach is helpful for understanding the intensity of buildings, we can not infer information about the possible locations of a specific type of buildings. Secondly, our chosen patch size might be too large, resulting in too few positive labels and class imbalance. We also cannot predict the specific shape of buildings with a large patch size. Lastly, our imagery is restricted to 5 areas of interest. Hence, our modeling might not have enough generality if we need to apply it to new images.

Road Classification

Roads tend to be more reflective than background vegetation, but this makes them easy to mistake for other reflective surfaces like open water or clouds, as we found when training our models. Even after model tweaking, our road classifiers output road labels around the edges of clouds, which produces misleading predictions. Furthermore, our models had excellent accuracy but tended to have poor precision and recall, meaning they did not pick all pixels where a road was present and that the pixels they did predict a road for were not overwhelmingly roads either. The excellent accuracy is attributable to the fact that most of the pixels are not roads, meaning the model scores an overwhelming number of true negative predictions. The best approach to these limitations is a convolutional neural network (CNN) because it considers pixel context, which can help distinguish pixels in a spindly road from similarly reflective pixels in a globular cloud. Given time constraints, however, this will have to remain a task for future teams.

Land Cover Classification

Our models are unable to detect land cover hidden underneath clouds, which becomes increasingly problematic when clouds cover a noticeable portion of a satellite image, as is the case for AOI1. One solution to this problem would be to take multiple images of the same AOI and create a composite image with clouds removed, as nothing in the data as it is can be amended to detect land cover beneath cloud cover.

Appendix

Band Number	Description	Wavelength	Resolution
Band 1	Visible blue	0.45 to 0.52 µm	30 meter
Band 2	Visible green	0.52 to 0.60 µm	30 meter
Band 3	Visible red	0.63 to 0.69 µm	30 meter
Band 4	Near-infrared	0.76 to 0.90 µm	30 meter
Band 5	Near-infrared	1.55 to 1.75 µm	30 meter
Band 6	Thermal	10.4 to 12.3 µm	60 meter
Band 7	Mid-infrared	2.08 to 2.35 µm	30 meter
Band 8	Panchromatic	0.52 to 0.90 µm	15 meter

Figure 12: Different Bands for Satellite Imagery

Logit Regression Results

Dep. Variable:	Label	No. Observations:	290600
Model:	Logit	Df Residuals:	290584
Method:	MLE	Df Model:	15
Date:	Wed, 30 Nov 2022	Pseudo R-squ.:	0.7242
Time:	16:30:27	Log-Likelihood:	-55552.
converged:	True	LL-Null:	-2.0143e+05
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
B1_m	-0.0422	0.002	-17.439	0.000	-0.047	-0.037
B2_m	0.0071	0.003	2.070	0.038	0.000	0.014
B3_m	-0.0895	0.003	-28.504	0.000	-0.096	-0.083
B4_m	0.0580	0.003	17.478	0.000	0.051	0.064
B5_m	0.0316	0.003	12.508	0.000	0.027	0.037
B6_m	0.1022	0.002	62.069	0.000	0.099	0.105
B7_m	-0.0491	0.001	-41.467	0.000	-0.051	-0.047
B8_m	-0.0386	0.001	-47.736	0.000	-0.040	-0.037
B1_std	0.0627	0.008	8.228	0.000	0.048	0.078
B2_std	0.4715	0.010	48.350	0.000	0.452	0.491
B3_std	-0.5094	0.007	-73.369	0.000	-0.523	-0.496
B4_std	0.1021	0.009	11.663	0.000	0.085	0.119
B5_std	0.0378	0.006	5.863	0.000	0.025	0.050
B6_std	0.0297	0.004	7.335	0.000	0.022	0.038
B7_std	0.1483	0.006	25.037	0.000	0.137	0.160
B8_std	-0.0677	0.005	-12.521	0.000	-0.078	-0.057

Figure 13: Summary of Logistic Regression (Building Classifier)