Names: Sage McGinley-Smith (sagems), Hannah Norman (hnorman)

Final Project Milestone

Achievements

Raw Data to Training Data Pipeline

Over the past two weeks, we spent a large amount of time working directly with the satellite imagery in Google Earth Engine and the Google Earth Engine API in Python. The primary challenge of this part of the project involved combining multiple satellite datasets together to build a training dataset for the model. As discussed in class, the training data is just as, if not more, important than the model itself in terms of the accuracy of results. Thus, we were very particular about choosing a dataset that was preprocessed so as to avoid having to clean raw sensor data from the satellites. So, we selected the Harmonized Sentinel 2 dataset [1]. We found that high quality data only aligned with two years of our ground-truthed plantation mapping datasets from the Costa Rican Government, as the rest of the government data was from years where high quality 10m resolution Sentinel data was not available.

Ultimately, we were able to use Google Earth Engine to extract raster data for the years 2018 and 2019 with the bands of interest. We randomly selected 3500 points from the pineapple plantations and 3500 points from the background (non-pineapple) regions of Costa Rica for each year. Those shapefiles were then used to build out a master csv file with all of the band values at each of the 14,000 training pixels using the preprocessed Sentinel data.

Finally, we plan to investigate additional feature descriptors beyond the basic bands of interest. Specifically, we will examine how hydrologic and vegetative indexes, like the Normalized Difference Vegetation Index (NDVI) and the Normalized Difference Water Index (NDWI), might improve model pixel classification. So, the final step of training data collection required computing these indexes and adding them to the training data set so that we can later manipulate the data fed to the model during training as a way to study model optimization. The same was also done for latitude and longitude coordinate values.

All scripts for the training dataset creation can be found here:
1. Identifying and Exporting Background Points
2. Identifying and Exporting Pineapple Labeled Points
3. Extracting Pixel Values and Index Computation
4. Cleaning and Combining Datasets

In the appendix at the end of this document in **Figure 1** and **Figure 2**, you can see visualizations of the cleaned training data displayed across a map of Costa Rica.

Build Initial Decision Tree Model

Once the training data for the years 2018 and 2019 was cleaned and combined, we built an initial decision tree model using the scikit-learn Python library [2]. Data was cleanly divided into sets containing (a) bands, (b) additional index features, and (c) a combination of both data types. For each data category, a decision tree classifier was built and fit on an 80-20 train-test data split for a range of maximum depth levels, spanning from 1 to 20. See **Figure 3** for a plot illustrating model performance at each tested depth. With the optimal depth identified, the classifier was evaluated and its results analyzed with a confusion matrix, bar graph, and ROC curve.

In the interest of space, we will limit our discussion in this report to the optimal performance of the model, which occurred with set c, the complete data. Looking ahead, we want to identify the model which exhibits the best performance so that we can apply it to new, unseen data from the past four years, 2020-23. With our ground-truth data, the decision tree classifier offered promising results with an average accuracy of 85.3%. Furthermore, when identifying pineapple plantations, the model achieved a precision rating of 0.85, sensitivity of 0.83, and F1-score of 0.84. See **Figure 4** in the appendix for a confusion matrix representation of model performance, **Figure 5** for a bar graph depicting counts by class and type, and **Figure 6** for an ROC curve.

Our Python Notebook for the decision tree classifier can be found here: [3].

Next Steps

Moving forward, we will continue to optimize our decision tree model, and we plan to investigate alternative tree models that might perform better, including a random forest model and an XGBoost model for comparison. If time allows, we would also like to explore a simple CNN model and perhaps a U-Net model, too. And of course, the end goal is to apply the optimized model with the best performance to the unseen data from 2020-23.

Additionally, while testing each of these models, we will be exploring the effect of different features on model performance. We currently have basic bands, a few more complex index feature descriptors, and geographic coordinates, but depending on time constraints, we may add others. Finally, additional data may be incorporated into the training set if a need is identified.

Updated Timeline and Objectives

Week 3: optimize decision tree model, investigate alternative models (random forest, XGBoost, CNN, U-Net), evaluate final optimized model on 2020-23 test set, visualize predicted mapping
Week 4: code clean-up, write report, create figures, make presentation, practice, and DEMO!

Appendix

Figure 1. *Pineapple plantation points within pineapple plantation regions, 2018 dataset.*

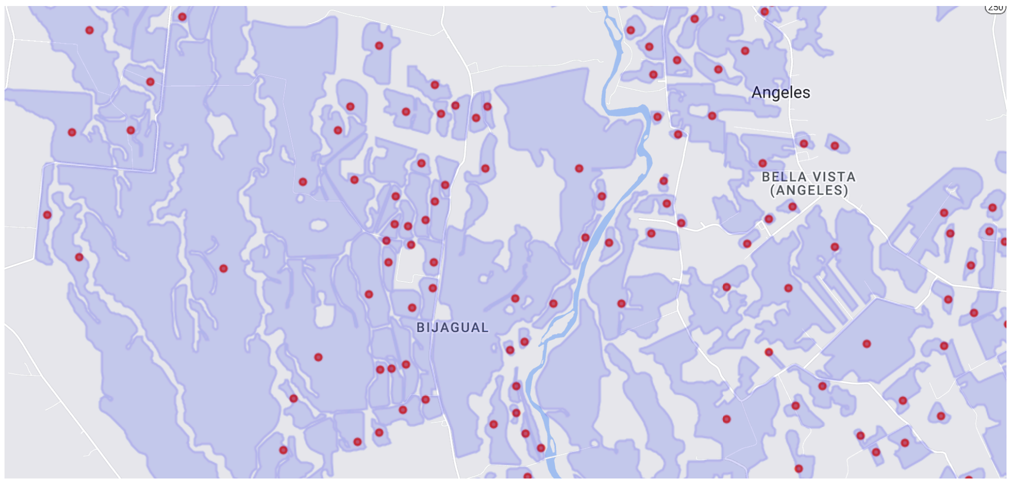


Figure 2. *All pineapple plantation and background (non-pineapple plantation) points, 2018 dataset. Visualized together across the entirety of Costa Rica.*

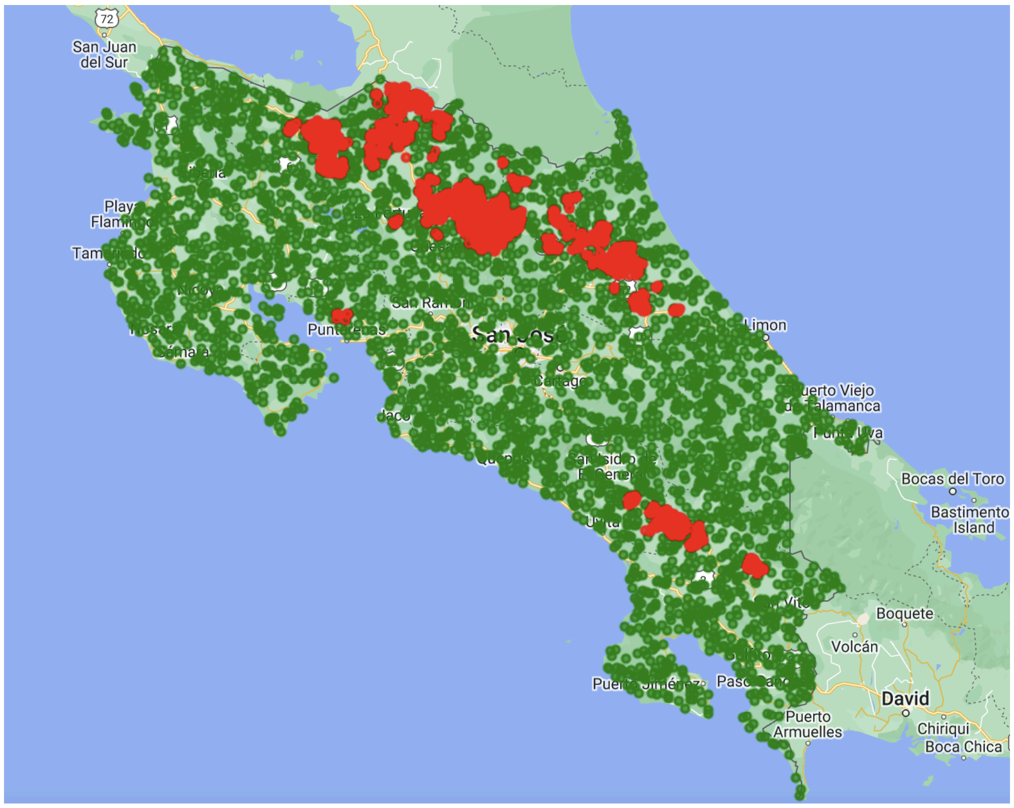


Figure 3. *Decision tree classifier performance at different max depths.*

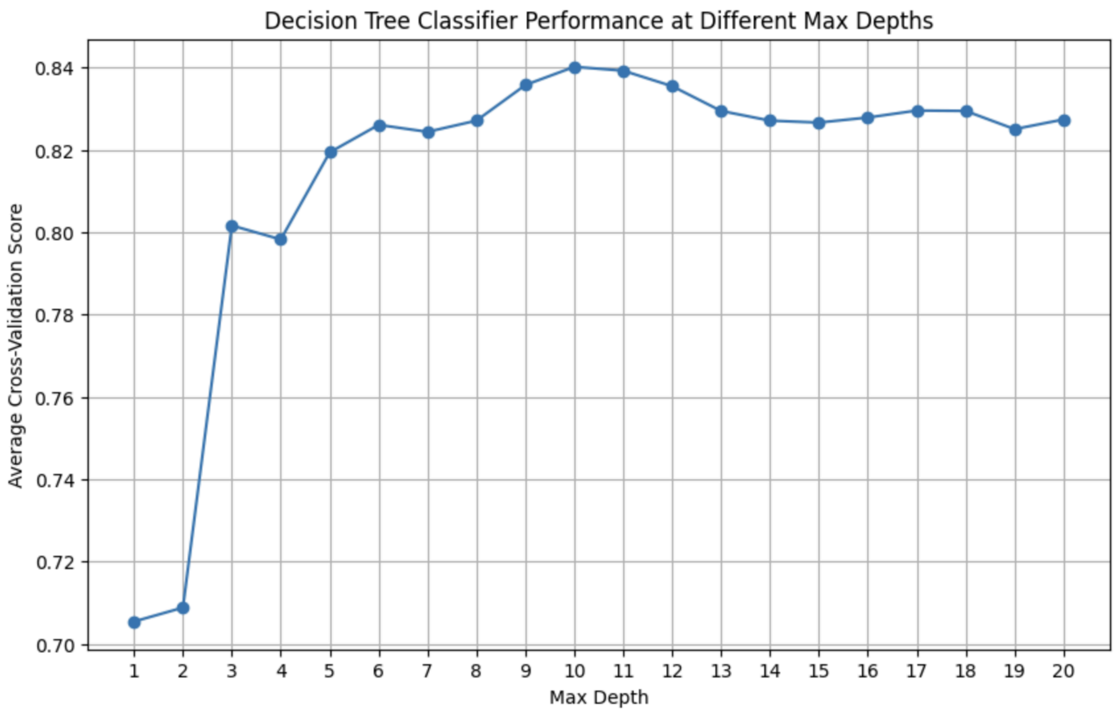


Figure 4. *Confusion matrix.*

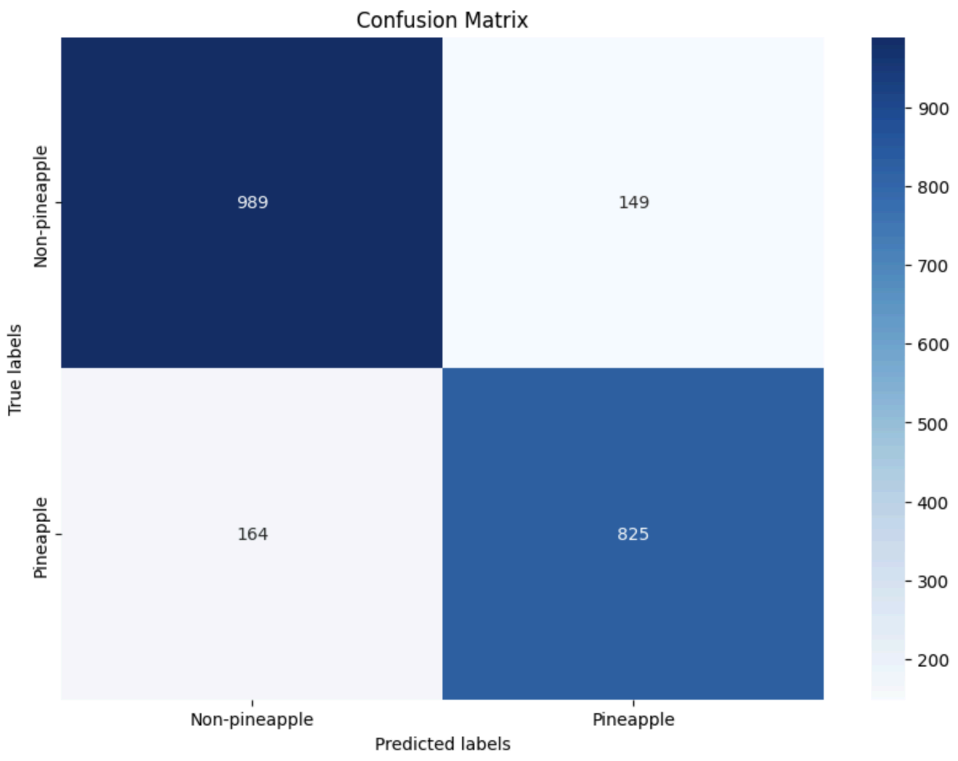


Figure 5. *Bar graph for counts by class and type.*

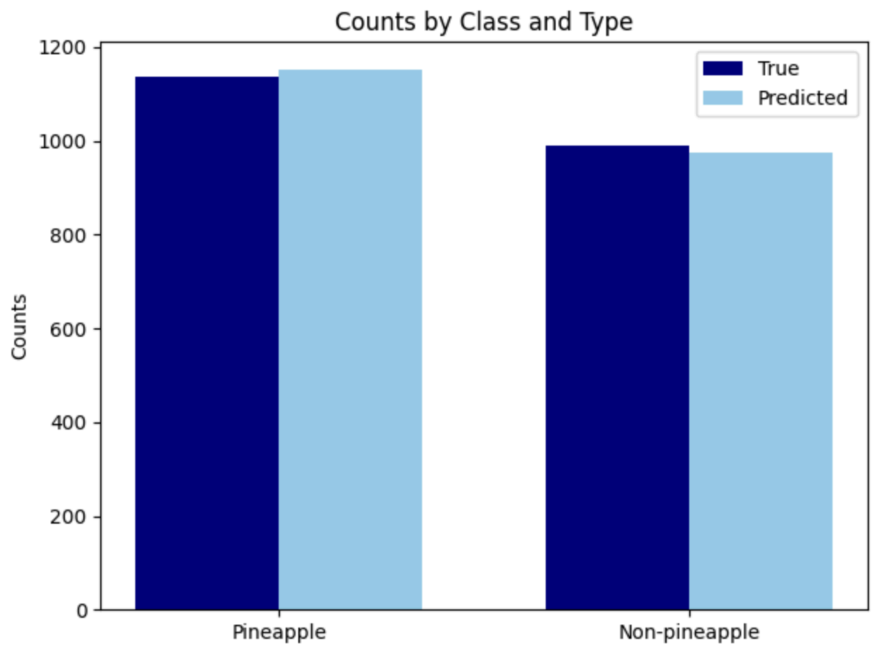
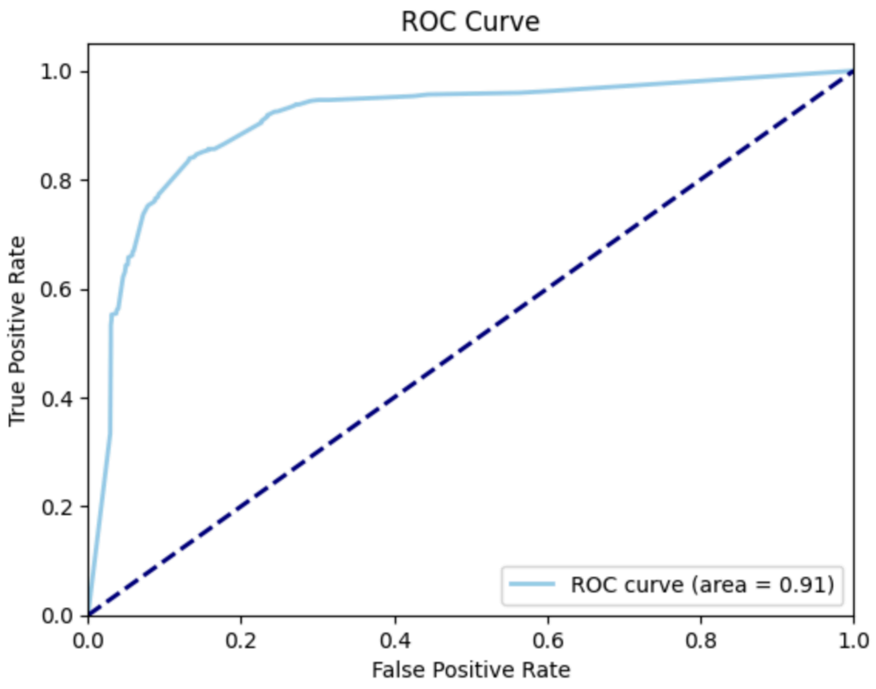


Figure 6. *ROC curve.*


ROC Curve
True Positive Rate
False Positive Rate
ROC curve (area = 0.91)