Technical Report: Forest Cover Type Prediction

Hector A. Guzman

April 24, 2025

Abstract

This project uses two machine learning models to classify forest data sets into one of seven types of forests. The models used are Neural Network, Random Forest, and XGBoost. Results show that Random Forest outperforms the Nueral Network and XGBoost just slightly outperforms Random Forest

1 Introduction

The purpose of the project is to use gathered information on a forest and classify what type of forest it is based on the information gathered. The data set provided has over 15,000 entries and 55 values to use in the models; and we will classify into seven different types of forests. For our machine learning models we will use a Nueral Network, Random Forest, and XGBoost for a performance increase.

2 Methodology

2.1 Data Preprocessing

The dataset has over 15,000 entries and 55 values that we can use to train the models. For the values ranging in the thousands, we will normalize those values to 0 and 1. There are three values that range from 0 to 255, we will normalize these as well. The soil types are already one-hot encoded so those will be left alone. The Id will be removed for training purposes to not skew the model.

For documentation purposes the model will be split into two separate splits. One will be the Nueral Network(nn) and the other will be the Random Forest(rf).

2.2 Neural Network

For the Neural Network we used 6 layers and Learning Rate Reduction optimization. Finally, a softmax is applied to help classify into the 7 types.

- **Regularization** – reduce learning rate down to 0.000001 (1e-6) when needed
- **Sigmoid Layers** – 5 sigmoid layers [256, 128, 64, 32 ,16]
- **Linear Layer** – Final layer with 7 nodes for the 7 types of classification
- **Softmax** – a softmax is applied afterwards

2.3 Random Forest

On the Random Forest we first ran 3 tests to check the sample splits, depth, and estimators. Then the best values are chosen to create the Random Forest model to train on.

- **Tests** – ran a test on samples splits, depth, and estimators.
  - *Samples*: 30
  - *Depth*: 32
  - *Estimators*: 300
- **RF Model** – create the model based on the parameters gathered

2.4 XGBoost

Create an XGBoost model with a softmax.

- Softmax
- 7 classes
- Random State of 42
- Early Stopping Round of 16

3 Results

We measured the accuracy of the validation and test sets for all models and the F-1 score. The following table shows the results

Model	Validation Accuracy	Test Accuracy	F-1 Score
Neural Network	70.34%	72.42%	70%
Random Forest	83.30%	89.97%	83%
XGBoost	86.44%	99.02%	86%

3.1 Lost & Accuracy Curves

For the Neural Network we plotted the lost curves over each iteration of the training. See Figure 1 and 2.

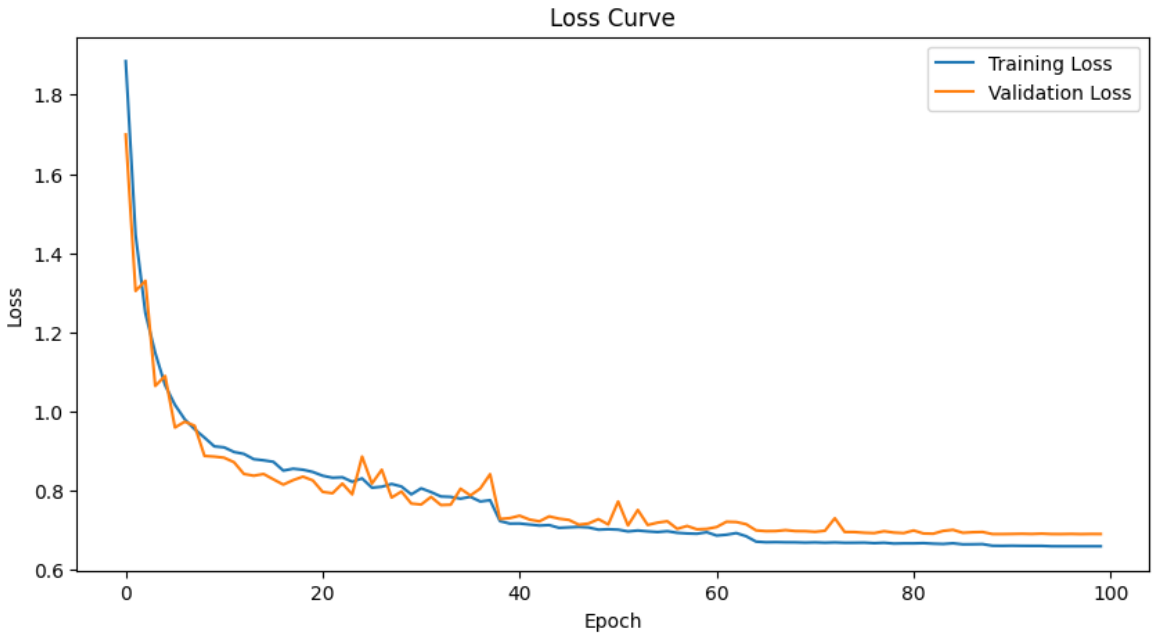

Figure 1: Loss curve for Neural Network

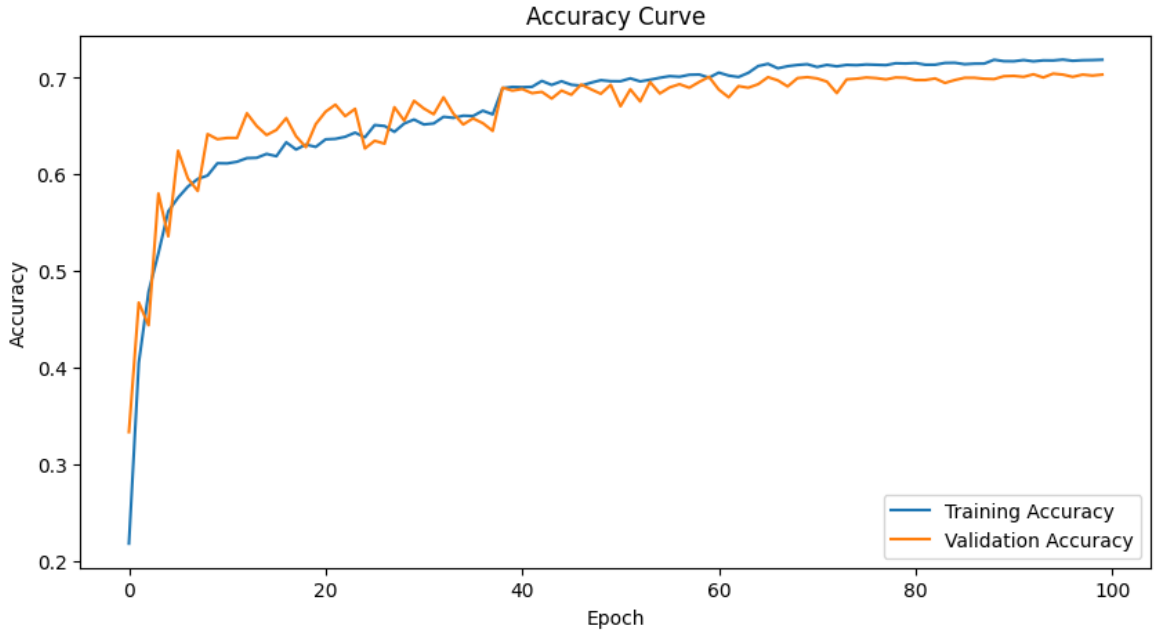

Figure 2: Accuracy curve for Neural Network

3.2 Confusion Matrix

For each model a confusion matrix was created to visualize what was accurately predicted. See Figure 3.1, 3.2, and 3.3.

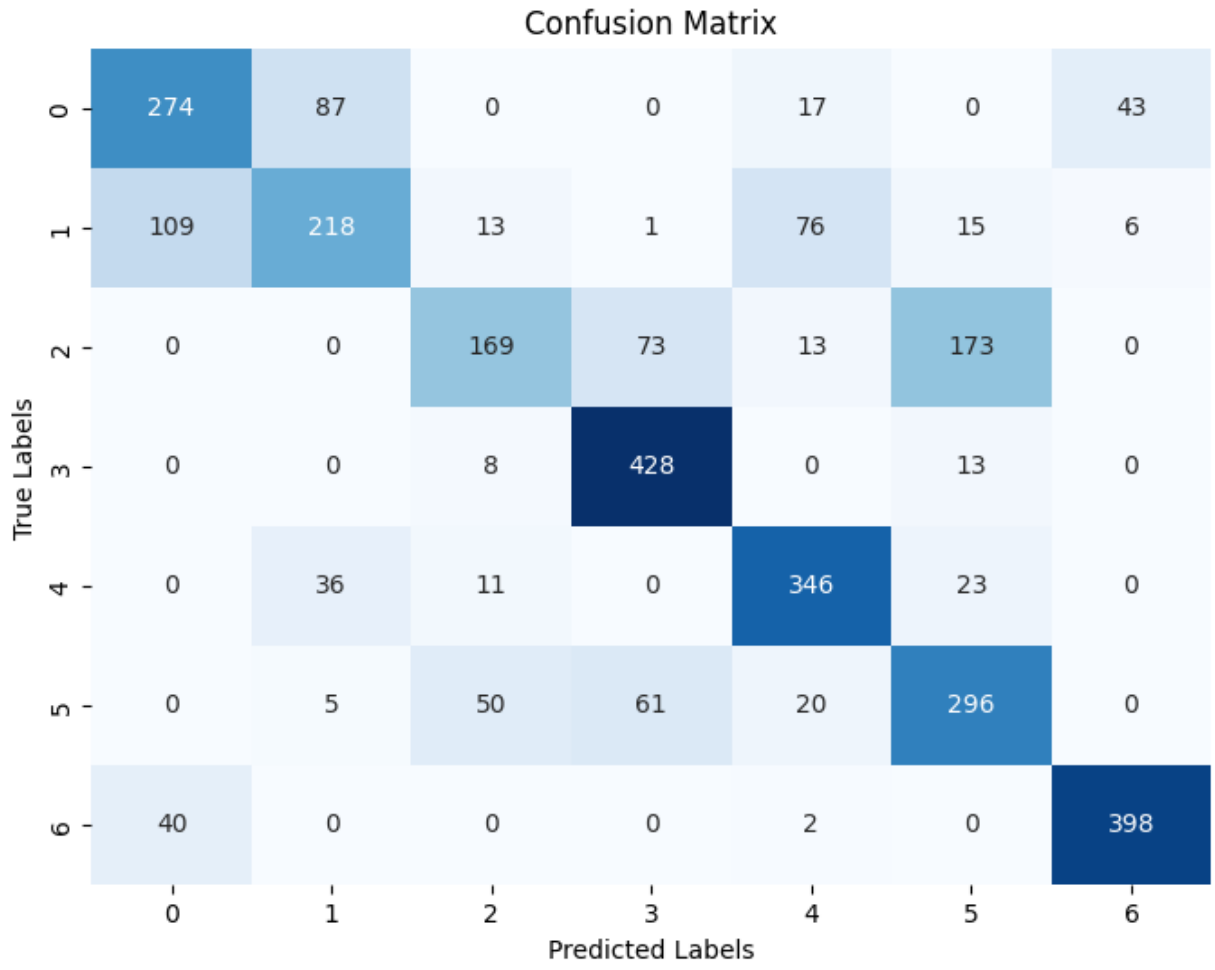

Figure 3.1: Confusion Matrix for Neural Network

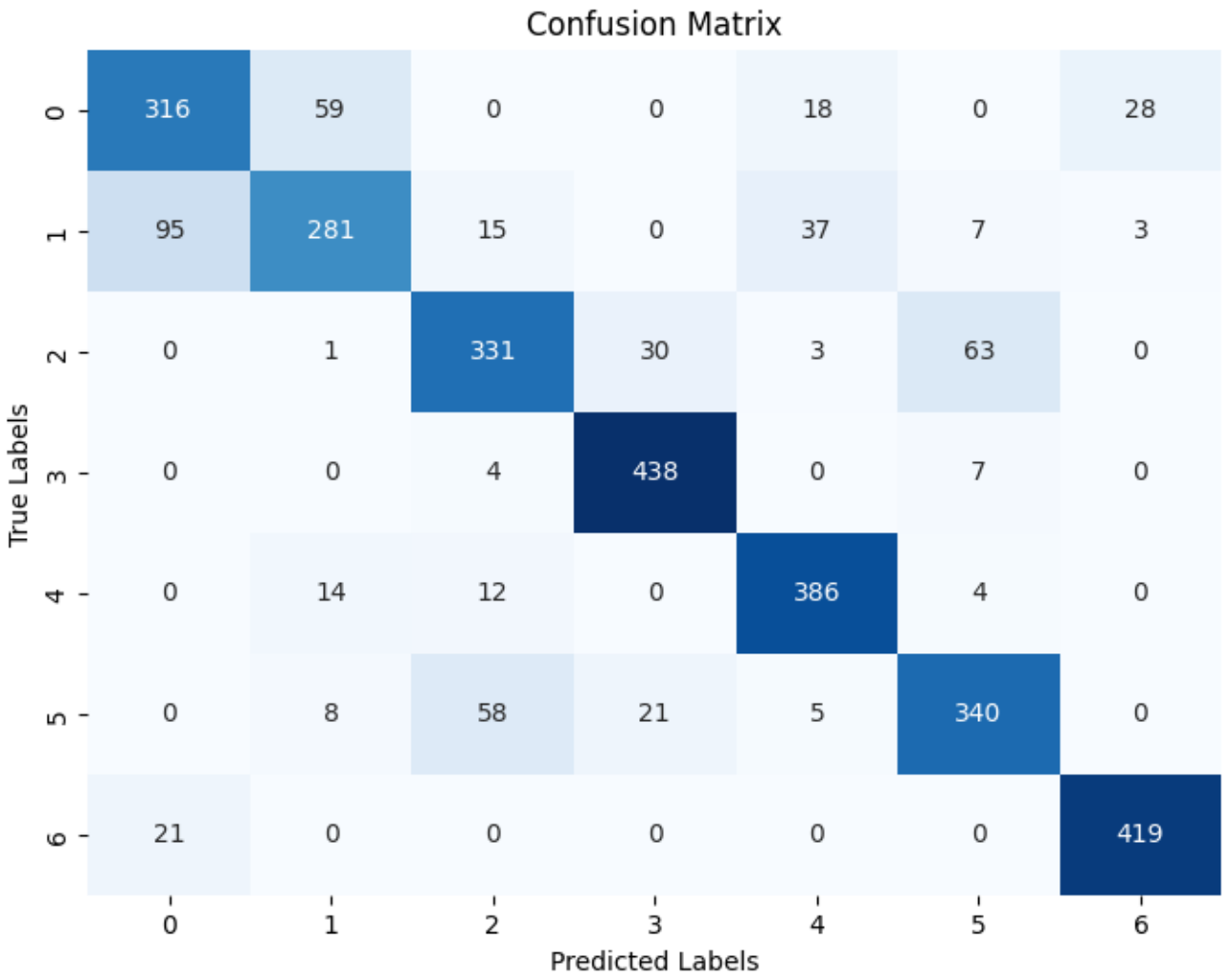

Figure 3.2 Confusion Matrix for Random Forest

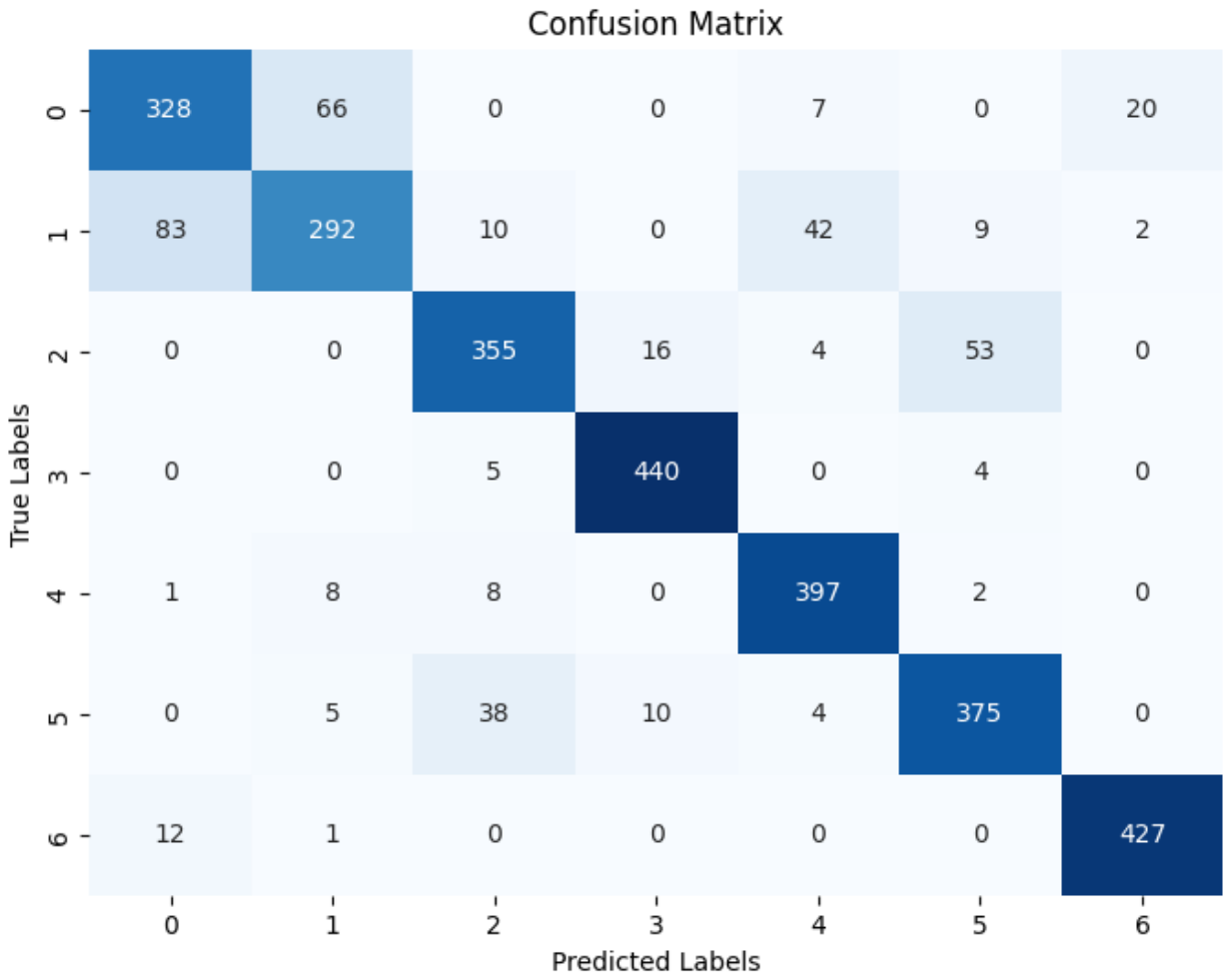

Figure 3.3 Confusion Matrix for XGBoost

4 Discussion

The results show that XGBoost outperforms both models, Neural Network and Random Forest.

For the Nueral Network we can infer that the reason for the poor performance is the amount of one-hot encoded values that dominate the dataset. The neurons do not work well with these values and there is a lot of information lost. In some iterations of the experiment, we changed the activation layers to “relu”, and it performed worse than the ‘sigmoid’ activations.

As for the Random Forest, the one-hot encoded values work well with this model due to the nature of a simple split in these values.

The XGBoost model outperformed the Random Forest model only by 3% in the F-1 score. The improvement is attested to how XGBoost reuses failed classifications to redo, and it may have been better to increase the early stopping parameter.

For the Nueral Network some more feature engineering will be required to improve the model and will all the models some regularization will be needed to reduce the noise in the validation sets.

5 Conclusion

In this project we concluded that Random Forest and XGBoost models are best for datasets that have values of either 0 or 1 that dominate the dataset. At best, the Nueral Network in these datasets is best viewed as a base line due to the nature of the data set. However, a hypothesis for when a neural network will perform best on a dataset of this category would be how much of each soil was found in the ground in each area. This type of dataset would work best on a neural network.