Remote Sensing for Forest Recovery: Final Report

Benjamin Frizzell Zanan Pech Mavis Wong Hui Tang
Piotr Tompalski Alexi Rodríguez-Arelis

2025-06-18

Table of contents

---

1 Executive Summary

Problem. Field surveys at Year-7 are costly and delayed; need forecasts for 80% survival threshold to guide timely interventions.
The **Canada Forest Service** requires an early-warning system to flag high-risk sites for proactive intervention.

Approach.
- Merged multispectral rasters (2010–2023) with planting and Year-7 survival survey tables.
- Two modeling phases:
1. **Static classifiers** (Logistic Regression, Random Forest, XGBoost) on aggregated features.
2. **Temporal RNNs** (GRU, LSTM) on annual spectral sequences.
- Employed grouped 5-fold cross-validation by site ID; optimized for F1 score.

Key Findings.
- **Random Forest** (80% threshold) achieved the highest F1 = 0.521.
- **Gradient Boosting** F1 = 0.411; **Logistic Regression** F1 = 0.515.
- **GRU** sequence model (site + spectral features) reached F1 = 0.440 and Recall = 0.569, outperforming LSTM (F1 = 0.368).

Deliverables.
- Reproducible Makefile targets (`make data`, `make train_models`, `make train_rnn`, `make evaluate`).
- Versioned model artifacts in `models/80` and evaluation plots in `results/80`.
- Quick-start Jupyter notebook for batch scoring.

Recommendations.
- **Advisory use only:** Validate flagged sites with targeted surveys before field action.
- **Data enhancement:** Expand ground surveys in under-sampled ecozones.

- **Future exploration:** Prototype a CNN–LSTM hybrid to jointly learn spatial and temporal patterns.
- **Deployment:** Develop a Streamlit dashboard for non-technical planners.

2 Introduction

Early canopy survival surveys (Year 7) are critical for assessing reforestation success, yet they are time-consuming and expensive. The **Canada Forest Service** seeks a data-driven early-warning system to identify sites at high risk of low survival before costly field surveys occur.

2.1 Data Overview

- **Multispectral Imagery (2010–2023):** Annual raster stacks of 10 spectral indices per site, capturing vegetation health over time.
- **Planting Records:** Approximately 11,000 geolocated planting sites with attributes such as planting density, species type, and planting year.
- **Survival Surveys:** Year-7 canopy cover measurements, binarized as survival (80%) or failure (<80%).

2.2 Refined Objectives

1. **Static Classification:** Train and evaluate Logistic Regression, Random Forest, and XGBoost models on aggregated spectral and site features to predict seven-year survival.
2. **Temporal Modeling:** Develop sequence models (GRU and LSTM) that leverage annual spectral time series for improved early forecasting.
3. **Performance Comparison:** Compare static and sequence pipelines using grouped 5-fold cross-validation by site ID, optimizing for F1 score to balance precision and recall.
4. **Reproducibility:** Package the entire workflow into a Makefile and Jupyter notebooks, enabling non-technical stakeholders to run data preparation, model training, and evaluation with minimal setup.

3 Data Science Methods

3.1 Data Cleaning

Before proceeding with modeling, we performed extensive data cleaning to address some quality issues in the dataset. The preprocessing steps were outlined below:

1. **Records Removal**

   - **Replanted Sites** : To avoid introducing complex survival dynamics, we removed all records from replanted afforested sites.
   - **Out-of-Range Values** : All records that are outside of the expected range for the spectral indices and survival rates were considered invalid and removed from the dataset.
   - **Missing Spectral Data** : All rows with missing spectral data were removed.
   - **Pre-Plantation Satellite Data** : To avoid introducing noise, satellite records captured before planting were removed, as pre-plantation site conditions are not relevant when modeling afforestation survival rates.

2. **Data Engineering**

   By normalizing tree counts (`Planted`) across site sizes (`Area_ha`), we created a new feature `Density`, which provides a more informative representation of underlying site conditions.

3. **Imputing Species Type**

   The missing values from the `Type` column was imputed from the `SpcsCmp` column. Using the threshold defined in the Forestry Glossary from Natural Resources Canada, sites were labeled as `Conifer` if the proportion of softwood species exceeds 80%, `Deciduous` if hardwood species exceeds 80% and `Mixed` otherwise.

4. **Column Removal**

   To reduce redundancy and ineffectual data, the following columns were dropped :

   - `PlantDt` : This column was dropped since the majority of values in the column were missing.
   - `NmblR`, `NmblT`, `NmblO`: As we excluded all replanted site records, these columns were no longer useful.
   - `prevUse`: Due to severe class imbalance, this column has limited predictive power.
   - `SpcsCmp` : As the majority of the data does not have any detailed species composition, this column was only used for imputing the species type.
   - `Year` : Both `Year` and `DOY` can be derived from `ImgDate`. The `Year` column was dropped to avoid redundancy with `ImgDate`. `DOY` was retained for seasonality tracking in RNN modeling.

- `Area_ha`, `Planted` : These two columns were dropped after deriving the new feature `Density`.

5. **Train Test Split**

   We performed a 70:30 train test split on the processed data, splitting the data by site. This ensures that each site appears only in either the training set or the test set, avoiding data leakage and preserves the temporal structure of the satellite data.

3.2 Phase 1: Classical Modelling

After data cleaning, we performed data transformation to prepare the cleaned data for classical model training.

1. **Pivoting survival records**

   We pivot the data to combine the survival rates columns (`SrvvR_1` to `SrvvR_7`) into a single column (`target`), and the survey dates columns (`AsssD_1` to `AsssD_7`) into a survey date column (`SrvvR_Date`). We added an `Age` column to keep track of the tree's age at the time of the survey.

2. **Satellite-Survey record matching**

   The survival rates data and satellite data were recorded at irregular time intervals. A ± 16 days average spectral signal of the survey date was computed to match the satellite data with the survival rate data. This time window was chosen specifically to match the repeat cycle of the Landsat satellite.

3. **Binary Target Mapping**

   We approached the problem as a classification problem and mapped the `target` (survival rates) into binary classes `Low(0)`/ `High(1)` survival rates based on a given classification threshold.

4. **One-hot-encoding of `Type`**

   While Random Forest and Gradient Boosting models have native support for handling categorical features, logistic regression models can only handle numeric features. To maintain consistency, OneHotEncoding was applied to the `Type` column for all classical models.

5. **Standard Scaling**

   Since the logistic regression model is also sensitive to the scale of the data, we applied StandardScaler to the numeric features before fitting the logistic regression model.

3.2.1 Logistic Regression

Logistic regression is a generalized linear model widely used for binary classification tasks, valued for its simplicity and interpretability. It provides a statistically grounded baseline and serves as a proxy for the classical statistical modeling used prior to this analysis. To demonstrate the value of more sophisticated machine learning models in predicting survival rates, subsequent models were expected to achieve performance exceeding that of logistic regression.

3.2.2 Random Forest

The Random Forest model is an aggregate model composed of many decision trees, each trained on a bootstrapped subset of the training data and a randomly selected subset of the features. Although training Random Forests can be computationally intensive, each tree is trained independently, enabling efficient parallelization and scalability. Previous studies from Bergmüller and Vanderwel (2022) have demonstrated that Random Forests perform well when using vegetation indices to predict canopy tree mortality. Becuase of this, this model was selected as a candidate for the present analysis.

3.2.3 Gradient Boosting

The Gradient Boosting model is a popular model that exists in a collection of 'boosting' models, which -unlike Random Forests- consists of a sequence of underfit and biased 'weak learner' models which converge to a high-performing 'strong learner' model when combined (Zhou 2025) by training on the errors of previous iterations. This model was selected as a candidate model due to strong performance across a wide variety of machine learning tasks; in particular, the implementation offered by the XGBoost library offers optimized training and additional regularization methods (Chen and Guestrin 2016).

3.2.4 Feature Selection

To address collinearity among vegetation indices and evaluate the importance of both site-based and remote sensing features, three feature selection methods were applied prior to tuning. Each of the methods vary in interpretabilty and handling of collinearity, and were chosen to compensate for eachother's disadvantages in this regard.

3.2.4.1 Permutation Importance

We estimate each feature's importance by randomly shuffling its values across samples before training, then measuring the resulting change in the model's performance. This yields an interpretable, global importance metric. However, when predictors are highly correlated, it can misattribute importance—because different features may serve as proxies for one another—leading to misleading rankings (Pedregosa et al. 2011).

3.2.4.2 SHAP Values

SHAP (SHapley Additive exPlanations) return per-prediction feature contributions based on Shapley values from cooperative game theory (Lundberg and Lee 2017). This method provides both local (per-prediction) and global interpretability. However, SHAP may tacitly distribute credit among highly correlated features, depending on whether the model uses marginal or conditional expectations when computing the baseline.

3.2.4.3 Recursive Feature Elimination with Cross-Validation (RFECV)

Finally, RFECV is used to iteratively train the model and remove the least important features based on model-derived importance metrics (e.g., coefficients or feature gains). Each reduced feature subset was evaluated by its F_1 performance using cross-validation. This method directly handles correlated features by eliminating them if they do not contribute to the model performance, however it can be quite computationally exhaustive. Feature rankings based on how early features were removed are used as importance metrics.

3.3 Phase 2: Temporal Models

While previous models provide strong benchmarks for supervised learning, their assumption of independent input instances fails to capture the sequential and spatial dependencies inherent in the vegetation index data. To address this, the final phase of analysis employed Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, which are well-suited for modeling temporal dynamics. These models, though more computationally intensive, are efficiently implemented using modern libraries such as PyTorch (Paszke et al. 2019).

3.3.1 Processing and Sequence Generation

To prepare our data for RNN modeling, we performed a series of data preprocessing.

1. **Validation Split**

   We performed a 50:50 split on the test data to obtain a validation set for model evaluation when training the RNN models.

2. **Data Engineering**

   - `Log transformed time_delta`: This feature records the difference between the image date and survey date. It is used to capture the irregularities in the time steps of the satellite records.
   - `negative cosine transformation DOY`: We perform a negative cosine transformation on `DOY` to capture the seasonality of the spectral indices.

3. **Data Normalisation**

   Since RNN models are sensitive to the scale of the data, we normalise the data to avoid vanishing or exploding gradient. To avoid data leakage, the summary statistics (mean and standard deviation) used for normalization was computed using only the training data.

4. **One-hot-encoding of `Type`**

   Since RNN models can only handle numeric data, OneHotEncoding was applied to the species `Type` column. `Type_Mixed` was dropped to remove linear dependencies between type columns and reduce redundancy.

5. **Sequence Generation**

   We split the site survey records and satellite records into separate data frames. For each row in the site lookup table, we searched the image table for all records with match `ID`, and `PixelID` and selected all satellite records up until the survey date. This would be the sequence data we use for training our RNN model. All survey records with no sequences available were removed from the dataset.

6. **RNN Dataset and Dataloader**

   To feed the sequence data into the RNN model, the sequence within the same batch needs to have the same sequence length. Depending on the age of the site, the sequence length for each survival record varies. To optimize memory usage while still introducing randomness to the data, we created a custom Pytorch dataset with an associated method that shuffles the dataset within their Age group to minimize the padding lengths required for each batch.

7. **Target mapping**

   We had trained regression RNN models instead of classification models, as training RNN models is time-consuming, and we want to avoid training separate RNN models for each threshold value. As such, the target values mapping to binary classes was done after model training.

3.3.2 Modelling Pipeline

Following preprocessing, the deep learning pipeline proceeds as follows:

1. The sequence of vegetation indices and engineered features is passed through a bidirectional GRU or LSTM, producing a hidden state.
2. The static site features are concatenated onto the hidden state vector and passed to a multilayer FCNN.
3. The final layer of the FCNN output is a scalar, which is passed through a sigmoid activation and multiplied by 100 to produce an estimate of the survival rate of the site pixel.

In addition to these steps, **layer normalization** was experimented with, although no improvement to predictions was observed. **Dropout** was also added within the FCNN layers, decreasing overfitting. Bidirectionality was added later in he analysis, as doing so seemed to generally increase performance by decreasing overfitting. Modelling was attemped with and without site features to assess their usage in predicting survival rate. Initally, prediction output was constrained to [0,100] using a simple 'clamp' function, but it was found that a smoother, scaled sigmoid actvation produced more consistent predictions.

Hidden state and hidden layer sizes, and the number of hidden layers are all variable hyperparameters than may effect model performance (Greff et al. 2017), however model performance seemed to 'plateau' after a certain degree of model complexity. For example, increasing the hidden state size beyond 32 did not increase prediction accuracy, nor did increasing hidden state layers beyond 3.

3.4 Evaluation Metrics

Model performance was primarily evaluated using F_1 score, **precision**, and **recall**, with classification accuracy treated as a secondary metric due to class imbalance favoring high-survival sites. To enable comparison between classical classification models and deep learning regression models, continuous survival rate predictions were thresholded to produce binary labels. In the context of these metrics, **low-survival sites are treated as the positive class**, reflecting the goal of identifying potentially failing afforestation sites for targeted intervention. To evaluate classical model performance across a range of decision thresholds, **Precision-Recall (PR) Curves** and **Receiver Operating Characteristic (ROC) Curves** were produced. As is standard for machine learning analysis, the area under each curve (**AUC, AUROC** respectively) were reported as a summary of performance across all thresholds.

4 Data Product & Results

Our project delivers a comprehensive, modular machine learning pipeline for predicting low survival rates in afforestation projects. This tool is designed for land managers, researchers, and anyone interested in using data to support better decisions in tree planting. The pipeline is also a practical example of how data science can be applied to real-world environmental challenges, and is intended to be accessible to users with a range of technical backgrounds.

4.1 What Does the Data Product Do?

Our delivered data product is a comprehensive machine learning pipeline, composed of a series of interconnected Python scripts. This pipeline automates the end-to-end process of data preparation, model development, and evaluation. Key functionalities include robust data processing steps such as cleaning, transformation (pivoting), and splitting into training and testing sets. Once the data is prepared, users can leverage dedicated scripts to train various machine learning models, perform hyperparameter tuning for optimization, and rigorously evaluate model performance. The primary objective of this data product is to provide an effective AI-driven solution for identifying and predicting instances of low survival rates.

4.2 Classical Models Evaluation

We trained several classical machine learning models, including Logistic Regression, Random Forest, and Gradient Boosting. These models were selected for their proven effectiveness in similar time series research. Logistic Regression provides interpretability, while the ensemble models capture more complex, non-linear relationships in the data. Each model was evaluated across multiple thresholds (50%, 60%, 70%, 80%) for defining low survival, providing a comprehensive view of model robustness under different definitions of the target variable.

4.2.1 Permutation Feature Importance

Permutation feature importance analysis showed that at lower thresholds (50% and 60%), Logistic Regression placed greater importance on remote sensing vegetation indices (NDVI, EVI1, EVI2, NDWI). These indices are derived from satellite imagery and reflect the health and vigor of vegetation. As the threshold increased to 70% and 80%, tree-based models, especially Gradient Boosting, assigned higher importance to structural features like Density and Age, which are direct measures of stand structure and development. Random Forest maintained moderate importance across both types of features, indicating a balanced approach.

4.2.2 SHAP Feature Importance

SHAP analysis provided a more nuanced view of feature contributions. Logistic Regression consistently relied on spectral indices, while Gradient Boosting balanced between structural and spectral features. Random Forest's SHAP values were less interpretable, suggesting reliance on complex feature interactions or limitations of SHAP with tree ensembles. This analysis helps users understand which variables are driving predictions and can inform future data collection or management strategies.

4.2.3 Recursive Feature Elimination (RFE)

RFE demonstrated that as less informative features were removed, all models converged on a core set of impactful variables. At the highest threshold, features such as Type_Mixed and Coniferous became highly important, highlighting the value of both structural and categorical site characteristics. This process helps streamline the model, making it more efficient and potentially more generalizable to new data.

4.2.4 Precision-Recall Curves

Precision-recall curves showed that Random Forest maintained higher precision across a wider range of recall values, especially at the 80% threshold, indicating more robust performance in identifying low survival cases. This is particularly important in imbalanced datasets, where the minority class (low survival) is of greatest interest for intervention.

4.2.5 ROC Curves

ROC curves were relatively linear, suggesting that the models struggled to effectively distinguish between classes, particularly at lower thresholds. This highlights the challenge of predicting rare events and the need for ongoing model refinement.

4.2.6 Confusion Matrices

Confusion matrices revealed a high number of false negatives, meaning the models often failed to identify instances of low survival. This is a critical challenge for practical deployment, as missing at-risk sites could result in lost opportunities for timely intervention.

4.2.7 Evaluation Metrics

Given the pronounced class imbalance in our dataset, we prioritized the F1 score as our main evaluation metric. At the 80% threshold, Random Forest achieved the highest F1 score (0.521), outperforming both Logistic Regression (0.515) and Gradient Boosting (0.441). Performance improved as the survival threshold increased, suggesting that the models are better at identifying the most extreme cases of low survival.

4.3 Sequence Model Evaluation

We also implemented deep learning approaches, specifically Recurrent Neural Network (RNN) architectures including LSTM and GRU models, to address temporal dependencies in the data. These models are well-suited for sequential data, as they can capture patterns across time steps that classical models may overlook. The goal was to see if leveraging the time series nature of the data could improve predictive performance.

4.3.1 Residual Plots

Residual plots for the GRU and LSTM models showed that both tended to predict values close to 80% with high frequency, indicating a potential bias and limited flexibility to capture the full spectrum of true values. This suggests that the models may be overfitting to the majority class or failing to learn meaningful temporal patterns.

4.3.2 Confusion Matrices (RNNs)

At lower thresholds, the models failed to make correct predictions for low survival rates, likely due to data being heavily skewed toward higher survival rates. At the 80% threshold, the models began to show improvement but still did not match the accuracy of classical models. This highlights the challenge of applying deep learning to small or imbalanced datasets.

4.3.3 Evaluation Metrics (RNNs)

At the 50% threshold, all RNN models and feature combinations showed an F1 Score of 0, indicating no predictive capability. At the 80% threshold, performance improved: LSTM with Site + Satellite features achieved an F1 Score of 0.368, while GRU with Site + Satellite features attained 0.44. However, these results did not surpass the classical models, suggesting that more data or advanced feature engineering is needed to unlock the potential of deep learning in this context.

5 Conclusions & Recommendations

5.1 Limitations

Despite exploring both classical modeling and RNN modeling approaches, all our models failed to deliver satisfactory results for predicting tree survival rates. Here are the key limitations of our modeling approaches:

1. **Data Imbalance**

   Our data distribution was highly imbalanced, with the target values skewed heavily towards high survival rates. We believe this is the leading cause for the biased predictions across all of our models.

2. **Loss of Temporal Information in Classical Models**

   Our classical models fail to capture complex temporal structures in the satellite data. By averaging the satellite data over time, we were losing a lot of vital information, including seasonal variations and short-term vegetation responses.

3. **Lack of Spatial Information in RNN Models**

   Our RNN model lacks the ability to model spatial relationships between pixels. Each pixel is processed independently, ignoring spatial context within the same site. Neighboring pixels often share similar micro-climate and environmental conditions, without spatial data, our model may overlook key spatial dependencies that influence vegetation response.

4. **Misleading Target Labels**

   While our models were predicting at pixel level, survival records were recorded at site level and assigned uniformly to each pixel within a site. As a result, 'healthy' pixels and 'unhealthy' pixels are assigned identical targets labels, potentially misleading the model during training.

5.2 Recommendations

1. **Using Higher Resolution Satellite Data**

   Using higher-resolution satellite and field-survey data may help capture finer details and spatial variations between pixels of the a site, improving model performance.

2. **Obtaining Higher Resolution Field Survey Data**

   In the future, we recommend collecting survival rates at a finer spatial resolution to improve the precision of our training targets. When combined with high resolution satellite data, this would allow models to learn localized vegetation dynamics more efficiently.

3. **Obtaining Annual Survival Records**

   Acquiring additional training data with complete annual survival rate records would substantially enhance the dataset's temporal resolution and modeling potential.

4. **Modeling at Site Level**

   Since survival records are measured at site level, we recommend that future models should aggregate satellite information across all pixels within a site, making predictions per site rather than per pixel.

5. **Incorporating Spatial Data**

   We suggest incorporating spatial data such as GPS coordinates to the current dataset, allowing the model to capture spatial correlations across sites and pixels.

6. **CNN-LSTM model**

   Alternatively, we suggest using raw satellite imagery instead of pre-extracted spectral indices. Using satellite image directly allow us to utilize convolutional architectures to learn spatial patterns, potentially improving model performance.

   We propose exploring a CNN-LSTM architecture as a next step. In this hybrid approach, the satellite image for each site will first passed through a CNN to extract spatial features. The CNN outputs at each time step is then fed into an LSTM or GRU to capture temporal patterns. The final hidden state can be passed through fully connected layers to predict the survival rate for the entire site. This architecture naturally accommodates both spatial and temporal dependencies, addressing key shortcomings of our current models.

Tip

Rendering Instructions

```
cd reports/"final report"
conda activate mds-afforest-dev
quarto render report.qmd
open report.pdf
```

Bergmüller, Kai O, and Mark C Vanderwel. 2022. "Predicting Tree Mortality Using Spectral Indices Derived from Multispectral UAV Imagery." *Remote Sensing* 14 (9): 2195.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. ACM. https://doi.org/10.1145/2939672.2939785.

Greff, Klaus, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. "LSTM: A Search Space Odyssey." *IEEE Transactions on Neural Networks and Learning Systems* 28 (10): 2222–32. https://doi.org/10.1109/TNNLS.2016.2582924.

Lundberg, Scott M, and Su-In Lee. 2017. "A Unified Approach to Interpreting Model Predictions." In *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 4765–74. Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Paszke, Adam, Sam Gross, Francesco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Zhou, Zhi-Hua. 2025. *Ensemble Methods: Foundations and Algorithms*. CRC press.