Land Cover Classification Technical Report

This project aimed to develop a predictive model for land cover classification in a region experiencing rapid land-use changes. The primary objective was to classify each observation into one of three land cover categories: Buildings, Cropland, and Woody Vegetation Cover . The final deliverable included a submission file providing occurrence probabilities for each class along with critical technical documentation.

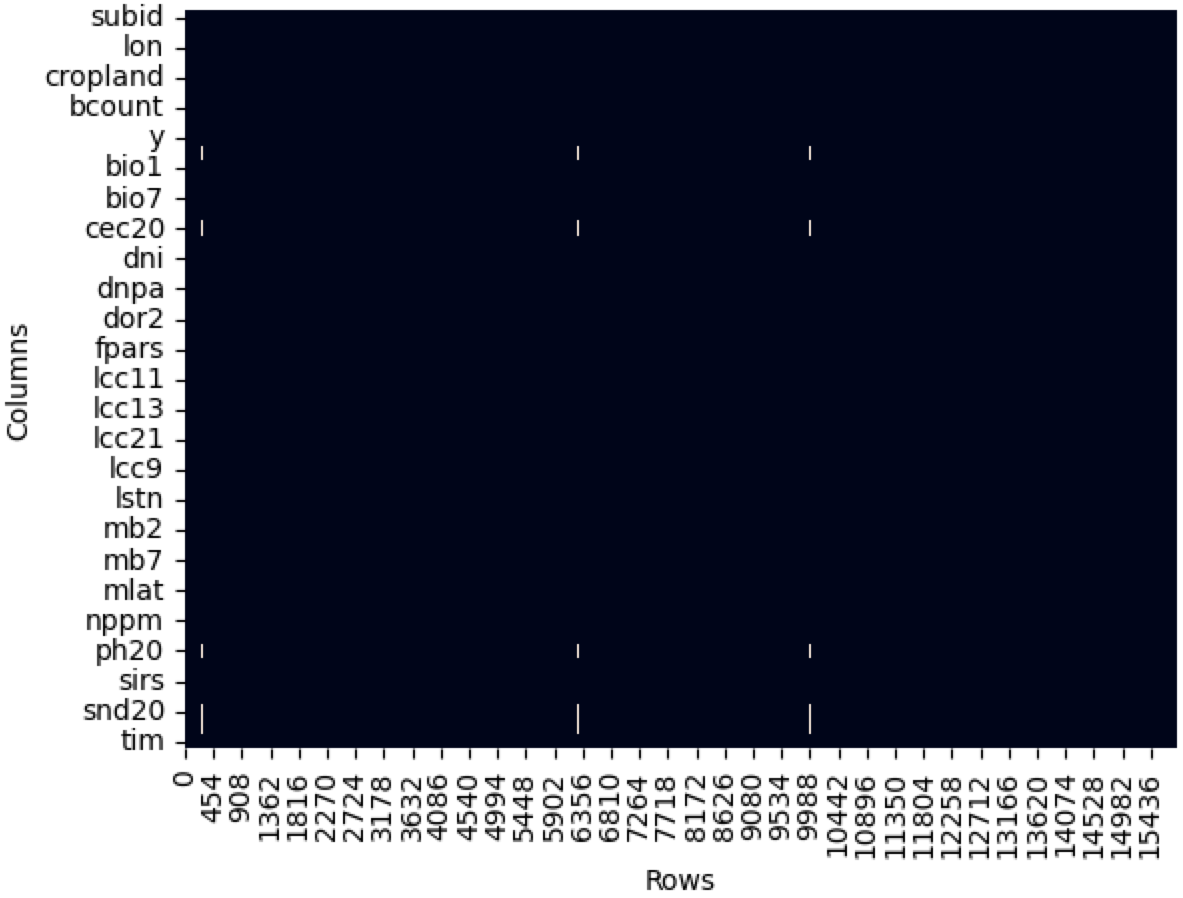
Methodology and Approach

Data Understanding

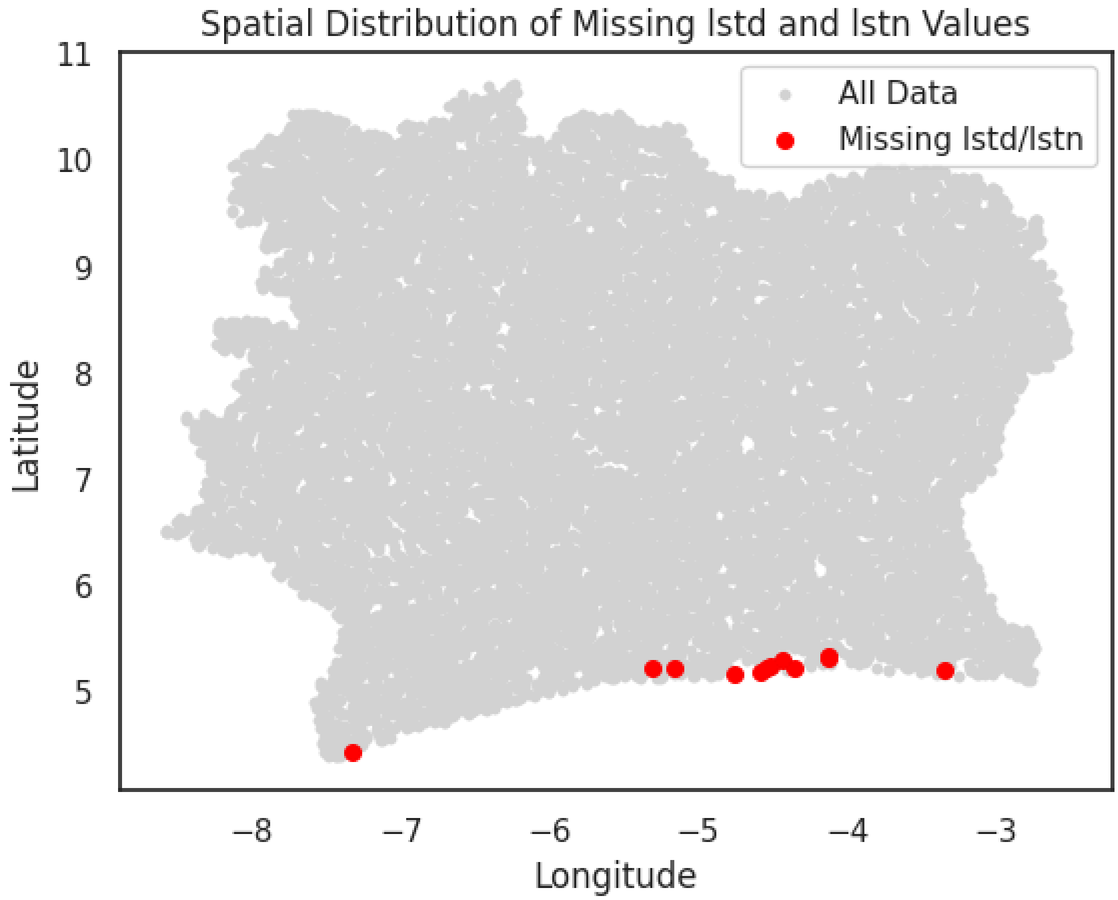
The dataset contained 15,811 training observations and a corresponding test dataset. Key columns included geographic coordinates (lat, lon), environmental features (e.g., bio1, bio12, bio7, bio15), remote-sensing variables (e.g., mb1, mb2, mb3, mb7), soil properties (bd20, cec20, ph20, soc20), and additional features without documentation (e.g., bcount, dnlt, nppm, sirs). The target variables were provided as three separate indicators: building, cropland, and wcover.

Data Analysis and Preprocessing

We began by assessing missing values using a heatmap (see Figure 1: Missing Values Heatmap), which revealed that most features had complete data with sporadic missing values in a few columns . bio1,cec20,ph20,snd20 had missing values which showed some pattern, in that the missing values was occured in the same rows. However, upon further exploration, it showed no correlation. The missing values, were only in few columns, so I went on to delete them



The lstd(Average day-time land surface temp. (deg. C , 2001-2020) and lstn (Average night-time land surface temp. (deg. C, 2001-2020) also had missing values. Upon investigation of their Spatial Distribution, it happened that the points that were with missing data were all near the southern/southwestern coast, with one row farther west. This could have been caused by persistent cloud cover, coaster influence, some other technical issues.



For this case, the missing data was imputed using KNNImputer, which included latitude (lat), longitude (lon), and elevation (mdem). The reason to use KNN Imputer was so that it could look for the closest points and average their known LST values, and therefore this would help capture coastal vs inland differences better.

]

including counts for building ("Yes"/"No"), cropland, and different ranges of woody cover (>60%, >30%, <30%).

Assessing inter-feature correlations, including those of undocumented variables, to determine their potential predictive value.

Feature Engineering and Encoding

To simplify the prediction task, we consolidated the three target columns into a single unified target variable ("landcover") using a hierarchical rule:

If building is "Yes", the observation is labeled as building.

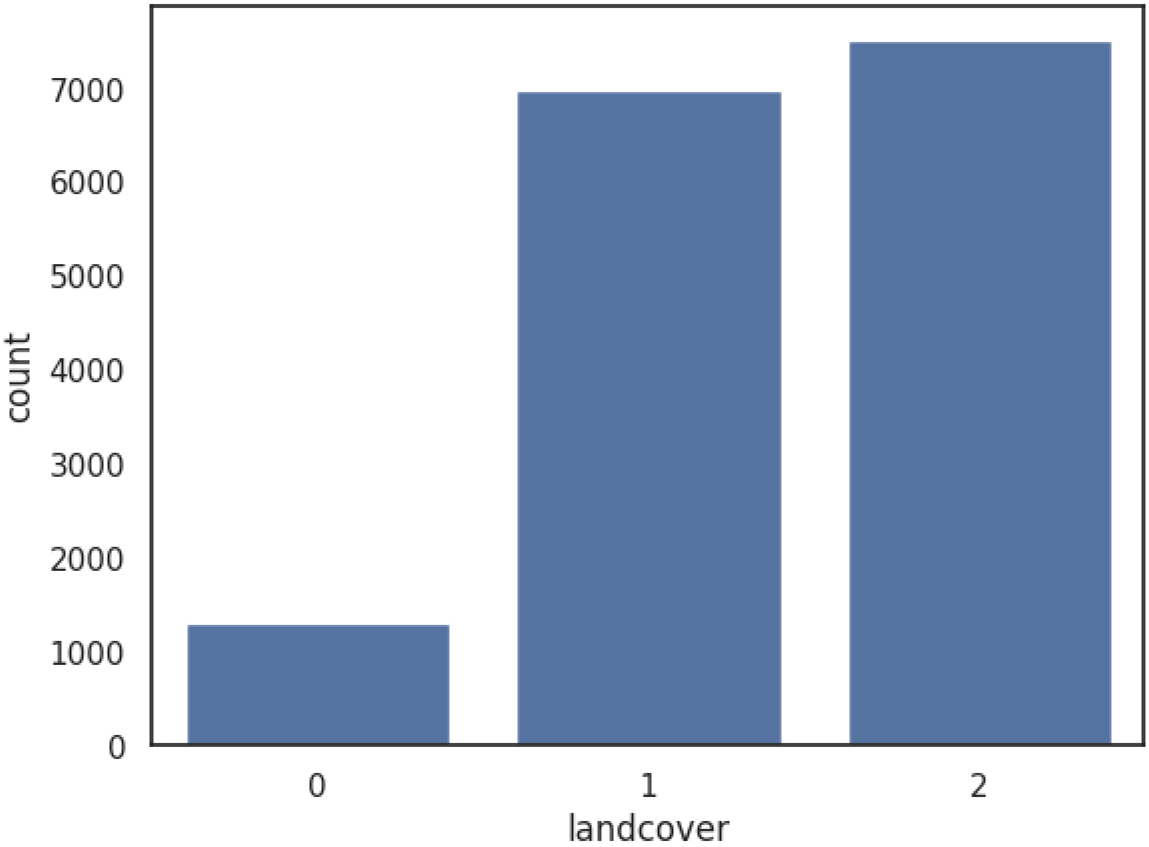
Else, if cropland is "Yes", label it as cropland.

Else, if wcover indicates “>60%”, label it as woody.

Handling Class Imbalance
The initial distribution of the target was imbalanced:

Building: 1,308 instances
Cropland: 5,232 instances (calculated)
Woody: 7,511 instances

The graph below shows the distribution:



To counteract this, we applied the SMOTE (Synthetic Minority Over-sampling Technique) to the training data. This technique generated synthetic samples for the minority class (buildings) so that all classes were balanced before training, thus mitigating bias towards majority classes.

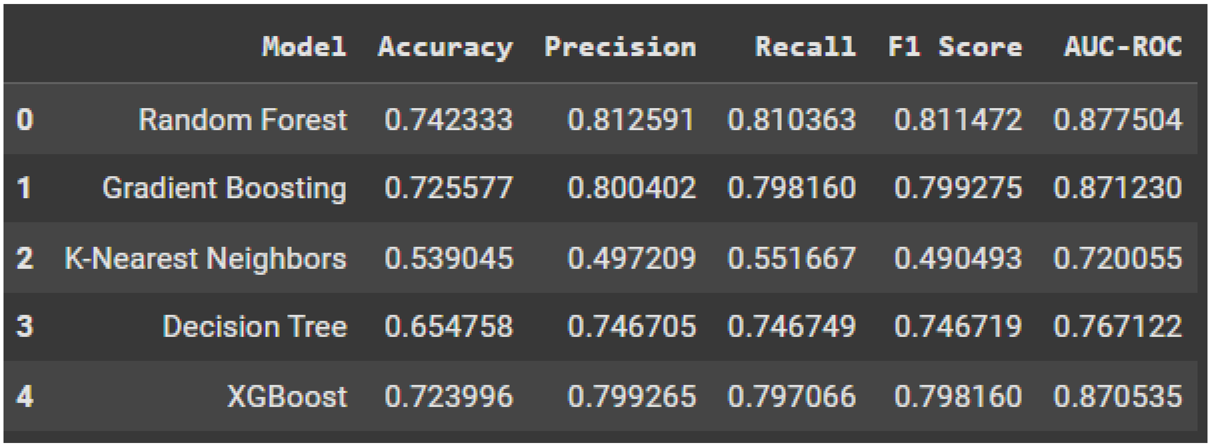
Model Training and Evaluation

We evaluated several classifiers:

- Random Forest

- Gradient Boosting
- K-Nearest Neighbors
- Decision Tree
- XGBoost

Using cross-validation and a suite of evaluation metrics (accuracy, macro-averaged precision, recall, F1 score, and multi-class ROC-AUC), the Random Forest model emerged as the top performer (Accuracy ≈ 74.2%, Macro F1 ≈ 81.1%, ROC-AUC ≈ 87.8%).



	Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC
0	Random Forest	0.742333	0.812591	0.810363	0.811472	0.877504
1	Gradient Boosting	0.725577	0.800402	0.798160	0.799275	0.871230
2	K-Nearest Neighbors	0.539045	0.497209	0.551667	0.490493	0.720055
3	Decision Tree	0.654758	0.746705	0.746749	0.746719	0.767122
4	XGBoost	0.723996	0.799265	0.797066	0.798160	0.870535

Hyperparameter tuning using GridSearchCV further refined the Random Forest settings (e.g., n_estimators=300, max_depth=None, min_samples_split=2, min_samples_leaf=1).

Feature Importance

- Feature importance analysis revealed that bcount (it documentation was not documented) was the most significant predictor, accounting for over 32% of the model's decision-making power.
- Other features, including several remote-sensing bands and derived variables, had moderate-to-low importance.
- Even undocumented features (e.g., dnlt, nppm, sirs) contributed meaningfully, validating our decision to retain them during initial exploration.

Test Set Prediction and Model Saving

After scaling features and ensuring consistency between training and test sets, the tuned Random Forest model was used to predict class probabilities for the test dataset.

Critical Findings

Dominant Role of bcount:

The bcount column (which upon googling, I assumed it represented building count) feature is the single most important variable, indicating that urban density is a strong indicator for land cover classification in this region.

Effective Handling of Imbalance:
SMOTE successfully balanced the target classes, leading to improved model performance—particularly for the underrepresented building class.

Model Performance:
Among the classifiers evaluated, Random Forest demonstrated the best balance in terms of macro F1 score and ROC-AUC, suggesting it is well-suited to the task.

Undocumented Features:
Despite lacking explicit descriptions, several undocumented variables (such as dnlt, nppm, and sirs) contributed to the model's predictive power, underscoring the benefit of retaining and testing these features.