Data Report: GEO-AI Cropland Mapping Project

1. Project Overview

This project focused on mapping cropland (crop vs non-crop) using multi-temporal satellite data with the aim of producing per-location cropland probabilities and a competition-ready submission. The notebook implements a full pipeline from data loading and preprocessing, through exploratory data analysis, feature engineering, model training and validation, to final inference and deployment (Streamlit).

2. Objectives & Audience

Primary objective: produce an accurate, reproducible cropland classification pipeline suitable for Zindi-style submission. Secondary objective: create an explainable and deployable inference pipeline for stakeholders.

Intended audience for this report: technical readers (data scientists/engineers) who want a concise but technical summary of the work.

3. Notebook Structure & Contents

The top notebook sections include business understanding, data understanding, data preparation checklist, EDA, feature engineering, modeling strategy, model evaluation, and final inference/submission checklist.

4. Tools & Libraries

Major libraries used throughout the notebook:
- Data & geospatial: pandas, geopandas, rasterio, fiona, pyproj
- Numerical: numpy
- Visualization: matplotlib, seaborn, folium
- Modeling: scikit-learn, imbalanced-learn (SMOTE), xgboost/lightgbm (if used)
- Deployment: streamlit

5. Data Sources & Loading

Data was loaded from multiple geospatial sources: CSVs, shapefiles, and satellites(Sentinel-1, Sentinel-2). The notebook uses pandas for table data and GeoPandas / Rasterio for spatial data handling. Key libraries used: pandas, geopandas, rasterio, fiona, pyproj.

6. Data Preparation & Preprocessing

This stage covers coordinate transforms, merging spatial layers, handling missing values, and scaling. Preprocessing steps implemented in the notebook include:

- Reprojecting spatial layers to a common CRS using pyproj/GeoPandas.
- Merging tabular labels to spatial centroids / grid using geopandas joins.
- Imputing missing values using scikit-learn's SimpleImputer where needed.

- Scaling numeric predictors with StandardScaler and/or MinMaxScaler in model pipelines.
- Encoding categorical variables with LabelEncoder and OneHotEncoder where applicable.

Key libraries used: scikit-learn (impute, preprocessing), numpy, pandas, geopandas.

7. Exploratory Data Analysis (EDA)

EDA in the notebook balances spatial and tabular exploration. Notable analyses and visualizations include:

- Class balance checks and distribution plots (countplots, histograms) using seaborn/matplotlib.
- Spatial plots of sample locations and predicted labels using GeoPandas plotting and folium for interactive maps.
- Temporal profile checks across spectral bands (Sentinel time series) to see seasonal differences between crop vs non-crop.
- Correlation matrices and feature importance previews to identify multicollinearity.

Key libraries used: matplotlib, seaborn, geopandas, folium.

8. Feature Engineering

Feature engineering is a major focus of the notebook . The notebook constructs spectral and temporal features and derives indices that are informative for crop detection.

- Computed spectral indices from Sentinel-2 bands (e.g., NDVI-like indices, band ratios) across multiple dates.
- Aggregated temporal statistics per location (mean, median, std, min, max) across the time series to capture seasonality.
- Engineered features capturing change/seasonality (differences between early and late season, percentile-based features).
- Spatial/contextual features such as neighborhood aggregates (mean within window), distance-to-features where applicable.
- Feature selection via SelectKBest(mutual_info_classif) to shortlist predictive variables.

Key libraries used: numpy, pandas, scikit-learn (feature_selection), xarray , rasterio for band access.

9. Modeling Approach & Pipelines

Modeling is built with scikit-learn-style Pipelines that chain preprocessing, feature selection, and an estimator. Cross-validation strategies are spatially-aware where possible.

- Pipelines combine SimpleImputer, StandardScaler, SelectKBest, and an estimator.
- Estimators explored: RandomForestClassifier, GradientBoostingClassifier, LogisticRegression (baseline).
- Class imbalance handled via SMOTE / imblearn methods in some parts of the notebook.
- Cross-validation: StratifiedKFold and KFold are used; GroupKFold / custom spatial CV is referenced to avoid spatial leakage.

- Model evaluation via cross_val_score, and metrics calculated include accuracy, F1-score, and ROC-AUC.

Representative keyword counts: RandomForestClassifier found 9 times; StratifiedKFold and KFold found 8 times each.

Key libraries used: scikit-learn, imbalanced-learn (SMOTE), joblib/pickle for model persistence.

10. Model Evaluation & Selection

Models were evaluated using a mix of cross-validated metrics and out-of-fold assessments. The notebook computes accuracy, precision/recall, F1, and ROC-AUC for comparisons. Confusion matrices and classification reports support qualitative assessment.

Key libraries used: scikit-learn.metrics, matplotlib for plotting evaluation curves.

11. Final Inference & Submission

The notebook includes a final inference pipeline: loading the trained model, generating predict_proba/predict outputs, and writing a submission CSV. Predict and predict_proba calls and to_csv writeouts were present.

Key libraries used: pandas for CSV I/O, scikit-learn for pipeline inference.

12. Deployment & Reproducibility

The final solution was deployed as an interactive web application (AgriVista ML Suite) using Streamlit. The platform enables both technical and non-technical users to interact with the trained cropland classification models in real time.

Key Features of the Deployed Application are :

I) Model Selection and Single-Point Prediction

Users can choose among several trained models, including Random Forest (best performer, 75.8% accuracy), XGBoost, and LightGBM.

The “Make Predictions” interface supports single-point inference, where users input values for up to 40 engineered features (e.g., NDVI, GNDVI, NBR, NDWI, MSAVI transformations, Sentinel-2 band indices).

A confidence threshold slider in the sidebar allows fine-tuning of sensitivity vs. specificity for classification output.

II) Interactive Geospatial Exploration

An integrated interactive map allows users to explore cropland training data, with clustering and heatmap visualizations.

Clicking on a map location reveals precise coordinates, which can be exported as a CSV file for later prediction.

III) Batch Predictions

Users can drag and drop a CSV file (e.g., exported from the interactive map) into the batch prediction tool.

The system processes multiple coordinate entries at once, returning cropland/non-cropland classifications with associated confidence scores.

IV) Model Explainability

A dedicated section provides model interpretability insights, enabling users to understand feature importance and prediction rationale, increasing transparency and trust in the results.

Deployment Outcome:
By integrating model inference, geospatial exploration, and explainability into one interface, the AgriVista ML Suite bridges the gap between advanced remote sensing analytics and practical decision-making in agricultural land monitoring.

13. Conclusions & Recommendations

The developed model demonstrates the potential of leveraging Sentinel-1 and Sentinel-2 satellite data for crop land classification. The results indicate that, with appropriate preprocessing and feature engineering, satellite imagery can deliver actionable insights at scale.

Recommendations for Model Utilization and Future Development:

I) Expand Data Sources for Greater Feature Diversity

Integrate data from additional satellites such as Sentinel-3 (for sea and land surface temperature) or Landsat 8/9 (for additional spectral bands). This could improve the model's sensitivity to environmental variations and increase classification accuracy.

II) Incorporate Temporal Analysis

Move beyond single-date predictions by incorporating multi-temporal imagery to track changes over time. This would help detect seasonal trends, anomalies, and progression patterns that are invisible in static snapshots.

III) Refine Spatial Resolution

Evaluate the impact of higher-resolution datasets (e.g., PlanetScope, WorldView) to improve small-area detection and boundary accuracy. While this may involve licensing costs, the gains in precision could be substantial for targeted monitoring.

IV) Integrate Ancillary Datasets

Combine satellite-derived features with on-the-ground data sources such as weather station data, soil quality measurements, or historical agricultural records to improve contextual decision-making.

V) Model Deployment for Real-Time Insights

Package the model into a deployable pipeline (e.g., using AWS SageMaker or Google Earth Engine) to enable automated, real-time monitoring and reduce turnaround time from data acquisition to decision-making.

VI) Field Validation and Stakeholder Feedback

Conduct pilot deployments in selected regions to compare model outputs with field-verified observations. This will help calibrate the model for regional variations and enhance stakeholder confidence.