FOML Hackathon Report

CS24MTECH14005 — Saumay Lunawat
CS24MTECH14014 — Mohit Patil

10 November 2024

1 Data Cleaning & Preprocessing:

1. **Data Splitting**:
   - Splits data into training and validation sets using an 80-20 split.
2. **Dropping Null Features**:
   - Drops columns with more than 90% missing values to reduce feature redundancy and improve data quality.
3. **Mean, Median, Mode Imputation**:
   - Fills missing values in selected features using the mean, median, or mode based on the features characteristics.
4. **Preprocessing Pipeline**:
   - Applies the above steps sequentially on training data, including specific imputation for selected features and final null value handling.

2 Feature Engineering:

1. **Tax Burden Ratio**:
   - Creates a feature representing the ratio of total assessed tax to total value, calculated as:

$$\text{TaxBurdenRatio} = \frac{\text{TotalTaxAssessed}}{\text{TotalValue} + 1}$$

2. **Agrarian Value Ratio**:
   - Calculates the proportion of agricultural tax value relative to the total value:

$$\text{AgrarianValueRatio} = \frac{\text{TaxAgrarianValue}}{\text{TotalValue} + 1}$$

3. **Farm Age**:
   - Derives the number of years since the field was established, calculated as the difference between the maximum `ValuationYear` and `FieldEstablishedYear`.

4. **Field Efficiency**:
   - Determines field efficiency by summing soil fertility, water resources, vehicle count, irrigation, and storage, divided by field size.

5. **Gini-Based Feature Importance Calculation**:
   - Uses a random forest classifier with the Gini impurity criterion to evaluate feature importance.
   - Drops the `Target` and `UID` columns, then fits the model to determine feature contributions.
   - Outputs a sorted list of features based on their importance scores for interpretability.

3 Training & Prediction:

This pipeline walks through preparing data, training a machine learning model, and making predictions.

1. **Load and Label Data:**
    - Load the data, then encode the target column so the model can understand it.
2. **Clean and Process Data:**
    - Use preprocessing to fill in missing values, handle nulls, and turn text into numbers.
3. **Create New Features:**
    - Add useful features like the age of the farm, tax ratios, and irrigation density to give the model more insights.
4. **Remove Redundant Features:**
    - Drop columns like unique IDs or coordinates that dont add much value to predictions.
5. **Split Data and Choose Features:**
    - Separate input features from the target. Apply feature selection to keep only whats important.
6. **Train Model and Make Predictions:**
    - Use a balanced random forest to handle any class imbalance. Train it on the data, then predict on test data.
7. **Decode and Save Predictions:**
    - Convert the predictions back to original labels, then save them in a CSV format for easy use.

4 Observations:

4.a Observations Of Data:

- The dataset contains a high degree of missing values in many features, notably: `CropFieldConfiguration` (99.74%), `FarmShedAreaSqft` (97.03%), and `TaxOverdueStatus` (97.29%) so we dropped columns having 90% or more null values.
- Certain features, such as `FieldShadeCover`, `FieldZoneLevel`, `ReservoirType`, and `WaterReservoirCount`, have only one unique value, indicating a lack of variability, this is handled during feature engg.
- `Latitude` and `Longitude` have a high number of unique values, indicating precise location information for most records, these can be used to impute other features.
- Features related to area, equipment, and tax values (e.g., `FieldSizeSqft`, `TotalAreaSqft`, `TaxLandValue`) show a significant spread with thousands of unique values, this needs to be handled.
- Some features, such as `DistrictId` and `NationalRegionCode`, have very few unique values, suggesting a limited geographical distribution, so these can be used to establish relationship with other features.
- Target variable (`Target`) has 3 unique values, suitable for a classification task, so we used RandomForest Classification for this task,.

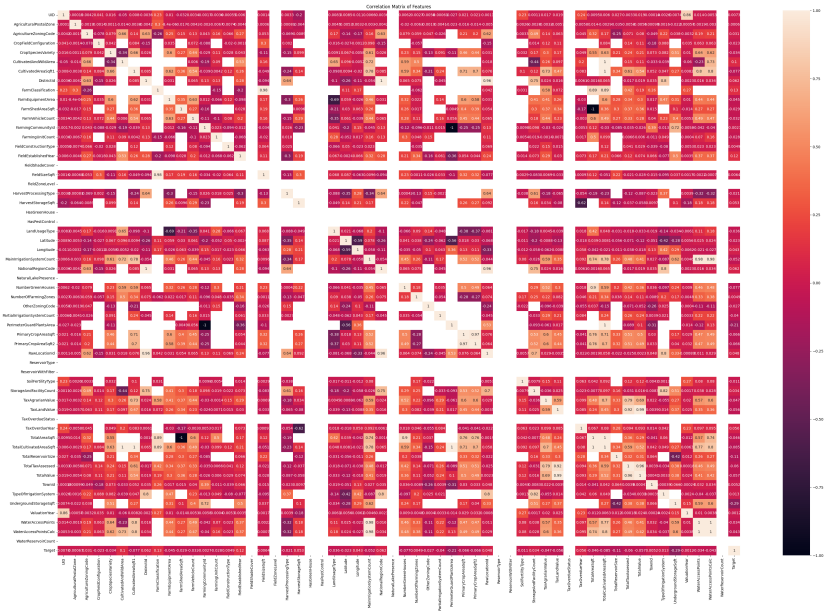

Figure 1: Correlation Matrix

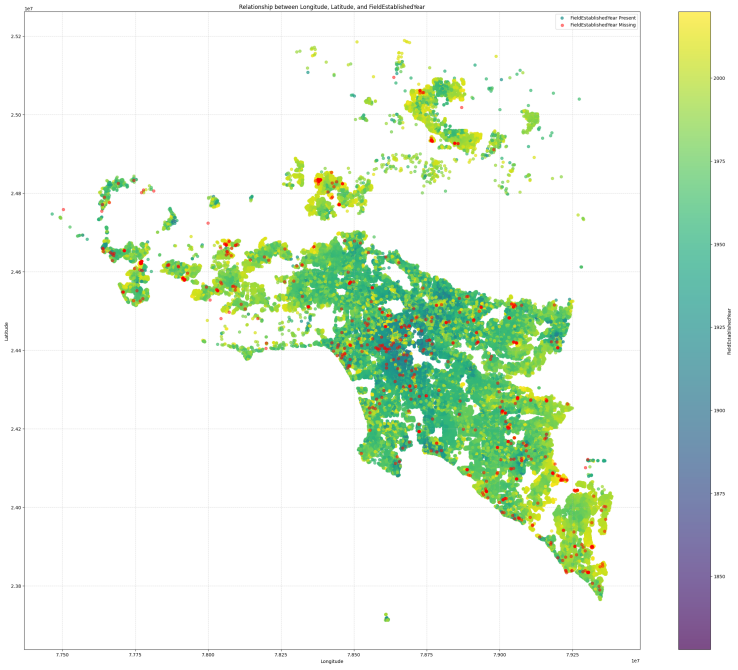

Figure 2: Latitude, Longitude vs FieldEstablishedYear with NULL VALUES

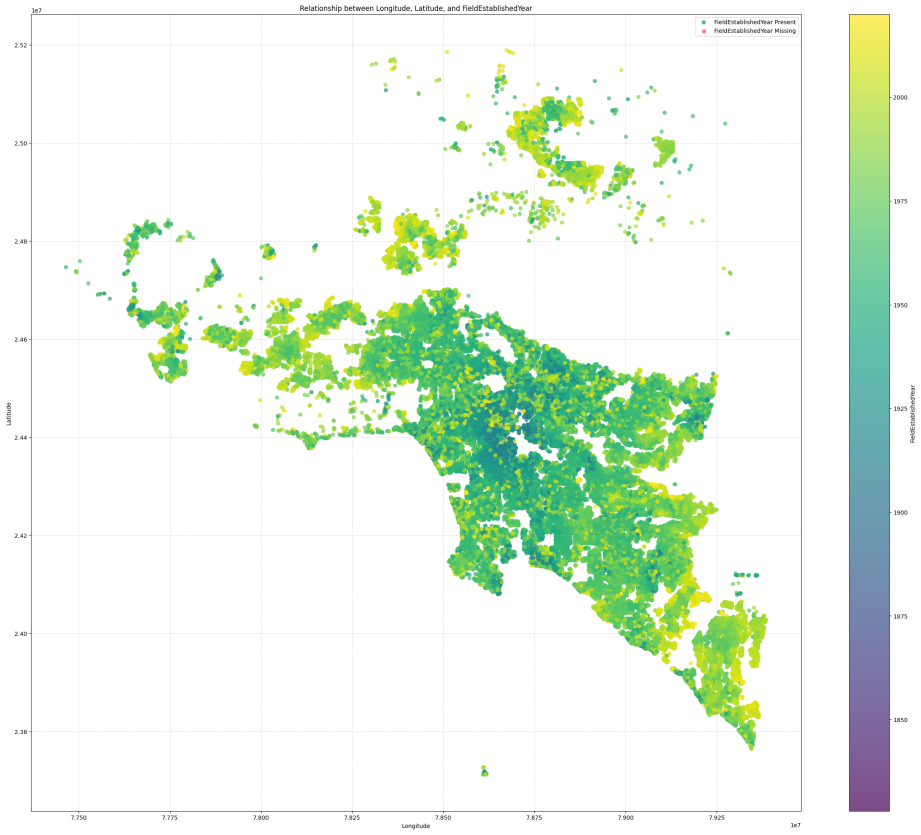

Figure 3: Latitude, Longitude vs FieldEstablishedYear after Imputing NULL VALUES

4.b Model Selection & Training:

- We chose BalancedRandomForestClassifier for this model as it showed as best results, we tried XGBoost too but the best results we got were from Random Forest.
- We did hyper-parameter tuning for getting perfect hyper-parameters for our RandomForestClassifier.
- After training the model on training set, and predicting it on test we got best f1-score of 0.426.