Methodology and Modeling Steps

1. Introduction

This presentation outlines the methodology and modeling steps used to predict farmer income in India. The primary objective is to build a robust machine learning model that accurately predicts income based on a variety of features.

2. Data Cleaning and Preprocessing

- Loaded the training and test datasets.
- Stripped leading/trailing whitespaces from column names.
- Handled missing values in 'Avg_Disbursement_Amount_Bureau' by filling them with the mean for the training set and 0 for the test set.
- Applied log transformation to skewed numerical columns: 'No_of_Active_Loan_In_Bureau', 'Non_Agriculture_Income', and 'Target_Variable/Total Income' to normalize their distributions.
- One-hot encoded categorical features related to soil type and water bodies for the year 2020 to convert them into a numerical format.

3. Feature Engineering

- Created a 'Village_Population' feature by counting the occurrences of each village.
- Engineered temperature-related features ('min', 'max', 'range') from the 'Ambient temperature (min & max)' columns for different seasons and years.
- Generated interaction features: 'Land_x_SocioEconomicScore' (Total_Land_For_Agriculture * KO22-Village score) and 'Land_per_Person' (Total_Land_For_Agriculture / Village_Population).
- Applied target encoding to 'State' and 'VILLAGE' columns based on the mean of the target variable to capture geographical influence on income.
- Created seasonal interaction features for soil and water body types between Kharif and Rabi

seasons.

4. Modeling

- Chose LightGBM, a gradient boosting framework, for its high performance and efficiency.
- Split the data into training (80%) and validation (20%) sets.
- Trained an initial LightGBM model with a basic set of parameters to establish a baseline.
- Performed hyperparameter tuning using RandomizedSearchCV with 3-fold cross-validation to find the optimal set of parameters for the LightGBM model.
- The best performing model was saved for generating predictions.

5. Prediction

- Loaded the preprocessed test data.
- Aligned the columns of the test set with the training set to ensure consistency.
- Used the trained LightGBM model to predict the log-transformed income values.
- Applied an inverse transformation (np.expm1) to the predictions to get the actual income values.
- Saved the final predictions in a CSV file named 'Predicted_Farmer_Income.csv'.