Multilingual Agricultural Query

Classification System

Yash Joshi - U23AI119

Harshit Shah - U23AI075

Diwyanshu - U23AI089

Pawan - U23AI130

Ashish - U23AI129

Department of Artificial Intelligence

2025

Project Repository

Project Explanation Video

Abstract

The advancement of technology in the agricultural sector has created immense opportunities to support farmers with reliable and fast information services. However, language barriers and diverse farming queries present challenges in delivering accurate assistance. This project presents a web-based multilingual agricultural query classification system that categorizes farmers' free-form queries submitted in Hindi, Gujarati, or English into specific agricultural categories such as Cultural Practices, Fertilizer Use, Market Information, Weather, and others.

The system employs a comprehensive multi-algorithm and ensemble-based Machine Learning pipeline. Six base classifiers—Naive Bayes, Logistic Regression, Support Vector Machine, Random Forest, Stochastic Gradient Descent Classifier, and XGBoost—were implemented and evaluated independently for each language. To improve classification robustness, we constructed voting classifier ensembles combining three and five base models from the pool of six, using both hard and soft voting strategies. Extensive hyperparameter tuning and performance evaluation were performed to identify the optimal model configuration for each language and a combined multilingual setting.

The final models were deployed through a web-based interface built with Flask, providing an intuitive platform for farmers to submit queries and receive categorized outputs in real-time. Experimental results demonstrated high classification accuracy across all languages, with the ensemble models outperforming individual classifiers. The architecture supports future scalability for additional languages or voice-to-text integration. This system contributes to agricultural technology by offering a scalable, language-sensitive query classification framework to bridge the information gap for farmers in multilingual regions.

1 Introduction

Agricultural information systems have made significant strides over the past decade, yet language barriers continue to pose a major challenge, particularly in multilingual regions. In India, approximately 43% of the rural population has limited proficiency in English or standard Hindi, creating a substantial gap in the accessibility of agricultural knowledge. This project addresses the need for language-sensitive agricultural query classification by developing an intelligent framework capable of categorizing farmers' free-form queries in Hindi, Gujarati, and English.

Problem Statement

Farmers in multilingual regions like India frequently submit agricultural queries in local languages such as Hindi and Gujarati. However, existing systems struggle to classify these queries due to language barriers, overlapping categories, and ambiguous query intent. This project aims to develop a multilingual, intelligent query classification system to accurately categorize free-form agricultural queries and bridge the information gap for non-English-speaking farmers.

Dataset Description

The dataset used in this project contains 50,000 free-form agricultural queries collected from farmer helplines, forums, and community platforms. The queries span 10 agricultural categories and are provided in Hindi, Gujarati, and English. Each entry includes:

- **Query ID:** A unique identifier.
- **Query:** The farmer's query text.
- **Class Label:** The category label (0–9).

The categories are summarized in Table 1.

Table 1: Agricultural Query Categories

Class	**Category Name**
0	Cultural Practices
1	Fertilizer Use and Availability
2	Field Preparation
3	Government Schemes
4	Market Information
5	Nutrient Management
6	Plant Protection
7	Varieties
8	Weather
9	Weed Management

1.1 Background and Motivation of the Problem

Agriculture remains the backbone of many economies, particularly in countries like India. Farmers often seek information regarding best farming practices, government schemes, weather forecasts, and market prices. However, a significant portion of the rural population communicates primarily in regional languages such as Gujarati and Hindi. Traditional agricultural information portals predominantly use English or Hindi, posing challenges for non-English-speaking farmers.

There is a strong need for an intelligent system capable of classifying agricultural queries in regional languages and directing farmers to appropriate resources. Machine Learning (ML) and Natural Language Processing (NLP) present powerful tools to address this gap. The motivation behind this project is to create a language-sensitive, category-specific query classification system to empower farmers and promote smarter agricultural practices.

1.2 Literature Survey or Related Works

Several research initiatives have explored the integration of agriculture and machine learning:

- **KisanSuvidhaApp:** Provides a broad range of agricultural information (weather, market prices, advisory content), but relies on keyword-based retrieval and does not perform detailed free-text query classification.
- **Agro Advisory Systems:** Many existing systems focus on weather forecasting and rule-based advisory generation for farmers. These solutions are typically limited to English, require structured or templated queries, and do not support nuanced classification of unstructured user input.
- **Regional Language Models:** Recent advances in multilingual BERT and Indian-language vectorizers (e.g., Indic-BERT, MuRIL) demonstrate strong performance on language-sensitive NLP tasks. However, these models have rarely been applied end-to-end for fine-grained classification of agricultural queries in Hindi or Gujarati.
- **Multi-algorithm Ensembles in Text Classification:** Prior work in general text classification has shown that voting and stacking ensembles of diverse base classifiers (SVM, Random Forest, XGBoost, etc.) can significantly boost accuracy and robustness, especially on imbalanced datasets.

Despite these advances, very few systems provide:

- Free-text classification of farmer queries in Hindi and Gujarati into detailed agricultural categories.
- A unified multilingual solution that leverages multiple algorithms and ensemble voting to improve classification accuracy across languages.

Our work builds upon these foundations by integrating advanced vectorization techniques (word- and character-level TF–IDF), a six-model voting ensemble framework, and a dedicated multilingual classifier, enabling accurate, language-agnostic categorization of unstructured agricultural queries.

1.3 Contributions

This project introduces several key contributions:

- Development of separate machine learning pipelines for English, Hindi, and Gujarati language queries, leveraging a unified voting ensemble classification approach for all languages.
- Implementation of a robust voting ensemble classifier, combining multiple base classifiers to improve prediction accuracy and handle multi-class agricultural queries.
- Design of a multilingual model capable of processing agricultural queries across multiple languages, ensuring seamless classification for diverse linguistic inputs.
- Integration of the models into a Flask web application, providing a user-friendly interface for farmers to input queries and receive actionable, categorized results.
- Categorization of agricultural queries into meaningful topics, enabling farmers to receive relevant and insightful responses based on their queries.
- Designed with flexibility in mind, allowing easy extension to add more languages and categories without requiring major architectural changes.

2 System Architecture

Figure 1 illustrates the workflow of our multilingual agricultural query classification system. The process begins with a user query in either Hindi, Gujarati, or English, followed by automatic language detection. After language identification, the text undergoes preprocessing to remove noise, normalize text, and prepare it for vectorization.

The vectorized text is then fed into a voting ensemble classifier composed of multiple base classifiers. Each classifier independently predicts a category for the query. The final predicted category is determined based on the **majority class output** from the ensemble of classifiers. This ensemble approach leverages the strengths of individual models to improve overall classification accuracy and robustness across languages.

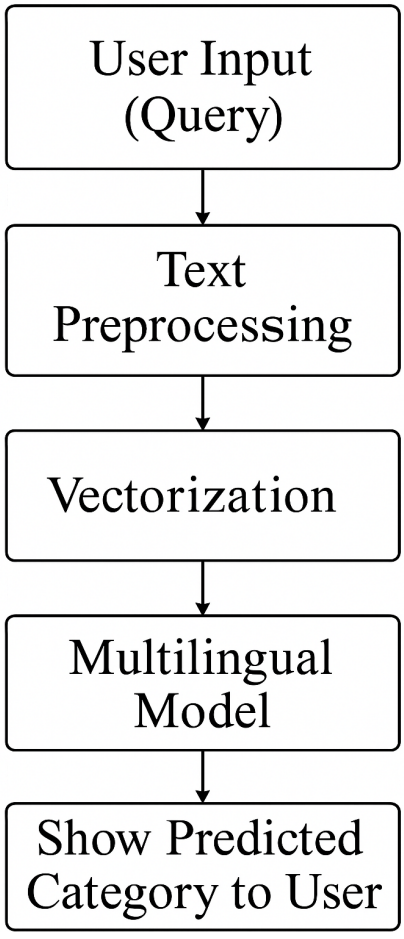

Figure 1: System Architecture for Multilingual Query Classification

3 Detailed Methodology

This section provides a comprehensive overview of the techniques and mechanisms employed in our multilingual agricultural query classification system. Our approach involves processing queries in three different languages (Hindi, English, and Gujarati) using specialized models for each language, with a common preprocessing foundation.

3.1 Data Preprocessing

Data preprocessing is a critical first step in our methodology, ensuring that the raw text queries are converted into a clean, standardized format suitable for machine learning algorithms. The preprocessing pipeline includes:

3.1.1 English Model

The preprocessing for the English model follows standard NLP practices to clean and normalize the text for analysis.

- **Punctuation Removal:** All punctuation marks are removed using regular expressions.
- **Tokenization:** The text is split into tokens using the `nltk.word_tokenize` function.
- **Stopword Removal:** Common English stopwords are removed using NLTK's predefined list.
- **Stemming:** The Porter Stemmer is applied to reduce words to their root forms (e.g., "farming" to "farm").
- **Alphabet Filtering:** Non-alphabetic tokens such as numbers or special characters are removed.

3.1.2 Hindi Model

The Hindi model's preprocessing is tailored to handle the Devanagari script and language-specific elements.

- **Non-Hindi Character Removal:** Only characters in the Unicode range `\u0900-\u097F` are retained.
- **Whitespace Normalization:** Extra spaces, tabs, and newlines are collapsed into a single space.

3.1.3 Gujarati Model

Gujarati preprocessing is focused on handling Unicode text specific to the Gujarati script.

- **Non-Gujarati Character Removal:** All non-Gujarati characters are removed, keeping only those in the Unicode range `\u0A80-\u0AFF`.
- **Whitespace Normalization:** Irregular spacing is normalized to a single space.

3.1.4 Multilingual Model

The multilingual model combines data from all three languages using a unified preprocessing strategy.

- **Punctuation Removal:** Irrelevant punctuation is removed via regular expressions.
- **Tokenization:** `nltk.word_tokenize` is used for splitting text into words across all languages.
- **Stopword Removal:** Language-specific stopwords (English, Hindi, Gujarati) are removed using dedicated lists.
- **Stemming:** Porter Stemmer is used for English, and Unicode-based stemmers are applied for Hindi and Gujarati.
- **Alphabet Filtering:** Only alphabetic tokens are retained, excluding numbers and symbols.

3.2 Feature Extraction using TF-IDF Vectorization

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (corpus).

- **Term Frequency (TF):** Measures how frequently a term appears in a document.

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

- **Inverse Document Frequency (IDF):** Measures how important a term is by weighting down frequently occurring terms and scaling up rare ones.

$$IDF(t) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing term } t}\right)$$

- **TF-IDF Score:** The product of TF and IDF.

$$TF\text{-}IDF(t, d) = TF(t, d) \times IDF(t)$$

The TF-IDF vectorizer transforms our text data into a sparse matrix where each row represents a query and each column represents a unique term in the corpus. The value in each cell is the TF-IDF score, which reflects the importance of that term in the query. This numerical representation enables machine learning algorithms to process the text data efficiently.

We employ both word-level (unigram) and character-level (n-gram range: 2 to 5) TF-IDF features to capture semantic and subword patterns. These are combined using a feature union strategy, resulting in a robust representation suitable for multilingual and morphologically rich languages such as English, Hindi, and Gujarati. .

3.3 Language-Specific Models

Our system employs four distinct models tailored to process agricultural queries in English, Hindi, Gujarati, and a combined Multilingual setting. While the preprocessing steps are common across all languages, each model has been trained on language-specific or combined multilingual datasets to capture the unique linguistic patterns and agricultural terminology of each language.

For each language, we implemented a comprehensive multi-algorithm approach utilizing six distinct classification algorithms and ensemble techniques to achieve optimal performance. The implementation follows a rigorous comparative and combinatorial methodology to identify the most effective classification solution.

Base Classification Models Implemented:

- **Naive Bayes (NB):** A probabilistic classifier based on Bayes' theorem that assumes feature independence, providing a fast baseline with surprisingly good performance for text classification tasks.

- **Logistic Regression (LR):** A linear model that applies the logistic function to model the probability of a certain class, effective for text classification due to its ability to handle high-dimensional data.

- **Support Vector Machine (SVM):** Implemented with both linear and RBF kernels, this algorithm identifies the optimal hyperplane that maximizes the margin between different classes, making it particularly effective for text classification tasks.

- **Random Forest (RF):** An ensemble of decision trees where each tree is trained on a bootstrap sample of the training data, providing robustness through aggregation of multiple decision boundaries.

- **Stochastic Gradient Descent Classifier (SGD):** An optimization algorithm that efficiently updates model parameters with each training sample, particularly useful for large datasets with elastic net regularization.

- **XGBoost (XGB):** An implementation of gradient boosted decision trees designed for speed and performance, incorporating regularization to prevent overfitting.

Ensemble Approach: Voting Classifiers

To leverage the strengths of individual models while mitigating their respective weaknesses, we implemented voting classifier ensembles across all languages:

- **Voting Classifier (${}^6C_3$):** Multiple combinations of three base classifiers from the pool of six models, utilizing both hard voting (majority rule) and soft voting (weighted probability averaging) strategies.

- **Voting Classifier (${}^6C_5$):** Combinations of five base classifiers from the pool of six models, providing more robust predictions while still allowing for diversity in the decision-making process.

The voting classifiers were implemented as follows:

$$VC_{hard}(x) = \text{mode}\{C_1(x), C_2(x), \ldots, C_n(x)\}$$

$$VC_{soft}(x) = \arg\max_i \sum_{j=1}^{n} w_j \cdot P(y = i|x, C_j)$$

where C_j represents the j-th classifier, w_j is the weight assigned to the j-th classifier, and $P(y = i|x, C_j)$ is the probability of class i given input x according to classifier C_j.

Model Selection Process

We employed a systematic approach to evaluate and select the optimal model configuration for each language:

- **Individual Model Evaluation:** Each of the six base classifiers was independently trained and evaluated on the dataset.

- **Hyperparameter Tuning:** Grid search was performed for each algorithm to identify optimal hyperparameters (e.g., regularization strength for SVM, number of estimators for Random Forest).

- **Ensemble Construction:** Various voting classifier combinations were constructed and evaluated: $\binom{6}{3} = 20$ distinct combinations of three classifiers and $\binom{6}{5} = 6$ distinct combinations of five classifiers.

- **Performance Comparison:** All models were compared based on accuracy, F1-score, precision, recall, and confusion matrix analysis.

Hindi Query Model

The Hindi query classification model was trained and evaluated using the above multi-algorithm and ensemble framework on a dataset of agricultural queries collected from farmer helplines and agricultural forums.

English Query Model

The English query classification model was developed using the same classification and ensemble methodology on a dataset of English agricultural queries.

Gujarati Query Model

The Gujarati query classification model followed the same multi-algorithm and ensemble approach as the English and Hindi models, trained on a dataset of Gujarati agricultural queries to capture relevant linguistic and domain-specific features.

Multilingual Query Model

The multilingual query classification model was trained on a combined dataset of English, Hindi, and Gujarati queries using the same classification and ensemble framework. This model was designed to generalize across languages by leveraging both language-specific and shared linguistic features, offering a unified solution for multilingual query classification.

3.3 Performance Evaluation Metrics

To evaluate the performance of our classification system, we used the **Micro F1-score** as the primary evaluation metric.

The Micro F1-score calculates the harmonic mean of precision and recall by globally counting the total true positives, false negatives, and false positives across all classes. This approach is particularly effective for multi-class classification with imbalanced datasets, as it gives equal weight to each instance rather than each class.

$$Micro\text{-}F1 = \frac{2 \times Micro\text{-}Precision \times Micro\text{-}Recall}{Micro\text{-}Precision + Micro\text{-}Recall}$$

The Micro F1-score was used to compare the performance of different models and ensemble methods across all languages.

4 Implementation Environment

This section outlines the software tools, development setup, and workflow used for the Multilingual Agricultural Query Classification System.

4.1 Software Requirements

The system was implemented using the following core modules and frameworks:

- **Python 3.x:** The primary programming language for building the system.
- **NumPy:** Used for numerical operations and array handling during data preprocessing.
- **Pandas:** Utilized for data manipulation, cleaning, and loading datasets.
- **Scikit-learn:** The main machine learning library used for implementing classification algorithms, model evaluation, and vectorization techniques.
- **NLTK:** Employed for natural language preprocessing tasks such as tokenization, stopword removal, and stemming.
- **Flask:** Lightweight web framework used to develop the web application interface.
- **HTML and CSS:** Used to design and style the front-end interface of the web application.

4.2 Development Environment

The system was developed and tested on the following environment:

- **Operating System:** Windows 10
- **Processor:** Intel Core i5
- **RAM:** 8 GB
- **Storage:** 256 GB SSD

4.3 Development Workflow

A modular and iterative development process was followed:

- Code written in Python using Jupyter Notebook and Visual Studio Code.
- Models trained, evaluated, and stored using joblib for persistence.
- Flask used to serve the trained models as an interactive web application.
- Basic HTML and CSS used to build a simple, user-friendly interface.

5 Results and Analysis

This section presents the outcomes of our experiments, including the performance of various classification models and the data characteristics. The analysis focuses on class distribution, model accuracy, and insights from confusion matrices and performance metrics.

5.1 Class Distribution

Understanding the distribution of query categories is essential for assessing model fairness and class imbalance. The following figures illustrate the class-wise distribution of queries across different datasets.

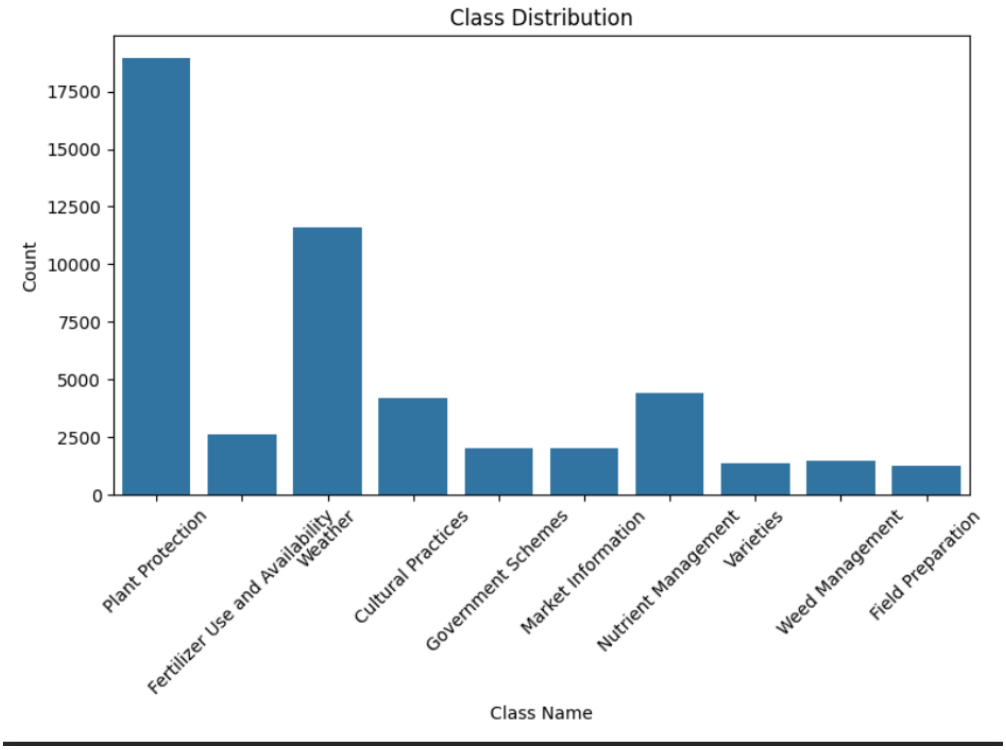

Figure 2: Class Distribution in English, Hindi, and Gujarati Datasets

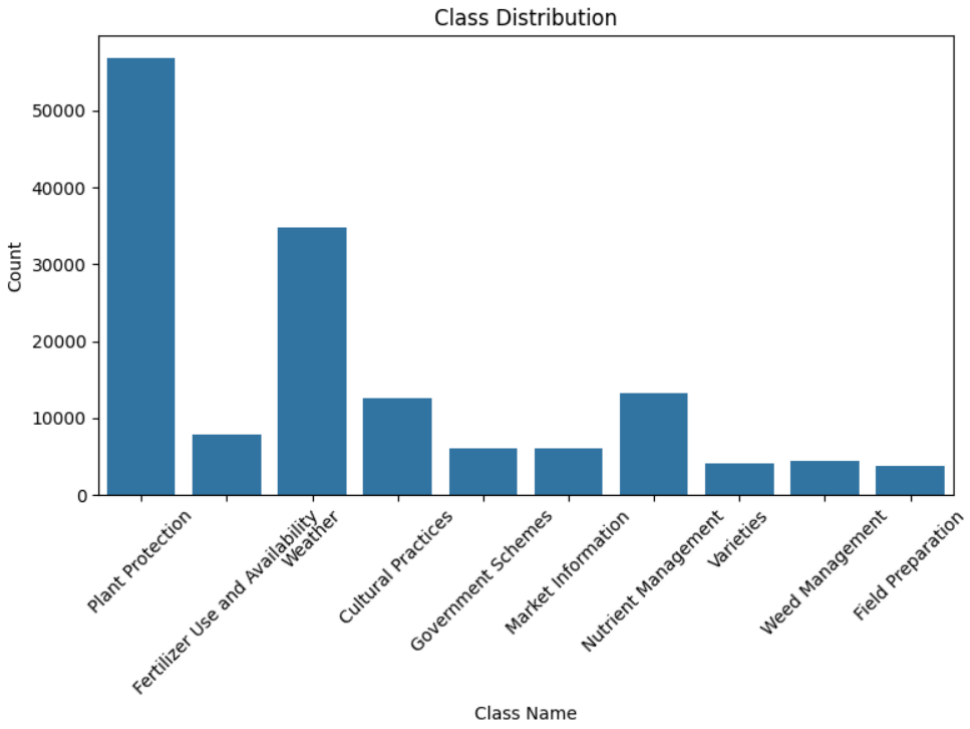

Figure 3: Class Distribution in the Multilingual Dataset

These visualizations reveal that categories such as *Plant Protection*, *Weather*, and *Nutrient Management* are more prevalent, indicating a significant imbalance across classes. To mitigate this issue and improve model generalization, we applied **undersampling** to the dominant classes. This strategy ensures that each category contributes equally during model training, reducing the bias toward frequently occurring classes and enabling fairer evaluation across all categories.

5.2 Model Performance

English Model (LR_RF_MNB Ensemble)

The following results show the performance of our classification models, with LR_RF_MNB emerging as the top performer.

Model	Feature	Micro F1
LR_RF_MNB	word(1gram)+char(2-	0.7071
LR_RF_MNB_SGD_	word(1gram)+char(2-	0.7035
MNB_SGD_XGB	word(1gram)+char(2-	0.7016
LR_MNB_XGB	word(1gram)+char(2-	0.7014
RF_MNB_SGD	word(1gram)+char(2-	0.6972
LR_MNB_SGD	word(1gram)+char(2-	0.6969
RF_MNB_SVM_SGD	word(1gram)+char(2-	0.6964
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.6962
RF_MNB_SVM	word(1gram)+char(2-	0.6935
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.6928
LR_MNB_SVM_SGD	word(1gram)+char(2-	0.6909
LR_RF_SGD	word(1gram)+char(2-	0.6899
MNB_SVM_XGB	word(1gram)+char(2-	0.6889
LR_MNB_SVM	word(1gram)+char(2-	0.6889
LR_RF_SVM_SGD_	word(1gram)+char(2-	0.685
LR_SGD_XGB	word(1gram)+char(2-	0.6845
MNB_SVM_SGD	word(1gram)+char(2-	0.6835
LR_RF_SVM	word(1gram)+char(2-	0.6819
RF_MNB_XGB	word(1gram)+char(2-	0.6814
LR_SVM_SGD	word(1gram)+char(2-	0.6807
LR_SVM_XGB	word(1gram)+char(2-	0.6803
LR_RF_XGB	word(1gram)+char(2-	0.6792
RF_SVM_SGD	word(1gram)+char(2-	0.6789
RF_SGD_XGB	word(1gram)+char(2-	0.6781
SVM_SGD_XGB	word(1gram)+char(2-	0.6775
RF_SVM_XGB	word(1gram)+char(2-	0.6727

Figure 4: Result all models

The LR_RF_MNB ensemble achieved a micro F1-score of 0.71, demonstrating robust performance across all agricultural query categories. This balanced metric confirms the

model's effectiveness in handling both frequent and rare classes within the English dataset.

	precision	recall	f1-score	support
Cultural Practices	0.47	0.36	0.41	842
Fertilizer Use and Availability	0.48	0.65	0.55	520
Field Preparation	0.14	0.24	0.18	256
Government Schemes	0.48	0.77	0.59	404
Market Information	0.81	0.78	0.80	408
Nutrient Management	0.51	0.51	0.51	885
Plant Protection	0.88	0.77	0.82	3792
Varieties	0.50	0.79	0.62	270
Weather	0.90	0.81	0.85	2323
Weed Management	0.63	0.84	0.72	300
accuracy			0.71	10000
macro avg	0.58	0.65	0.60	10000
weighted avg	0.74	0.71	0.72	10000

Figure 5: Performance metrics of the LR_RF_MNB ensemble

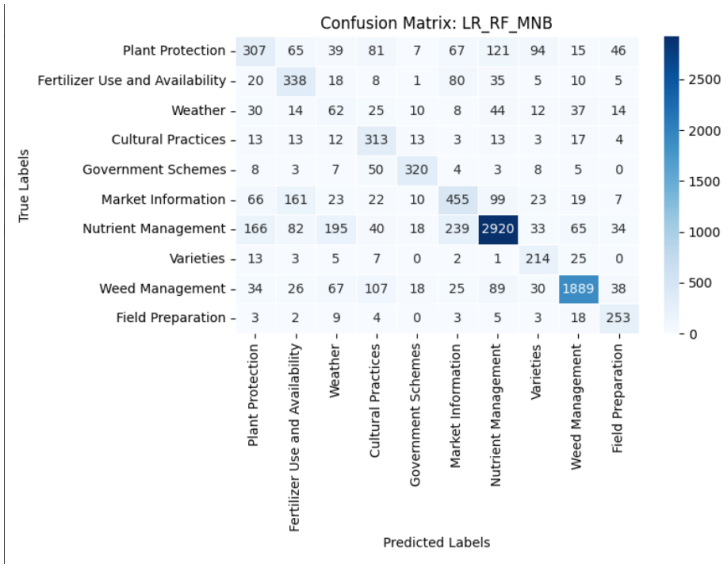

Figure 6: Classification results of the LR_RF_MNB ensemble

Hindi Model (LR_RF_MNB_SGD_XGB Ensemble)

The following results show the performance of our classification models, with LR_RF_MNB_SGD_XGB emerging as the top performer.

Model	Feature	Micro F1
LR_RF_MNB_SGD_	word(1gram)+char(2-	0.6873
LR_MNB_SVM_SGD	word(1gram)+char(2-	0.6846
LR_RF_MNB	word(1gram)+char(2-	0.6822
LR_MNB_XGB	word(1gram)+char(2-	0.6817
LR_MNB_SGD	word(1gram)+char(2-	0.6812
RF_MNB_SGD	word(1gram)+char(2-	0.6808
RF_MNB_SVM_SGD	word(1gram)+char(2-	0.6808
LR_RF_SGD	word(1gram)+char(2-	0.6784
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.6784
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.6776
LR_MNB_SVM	word(1gram)+char(2-	0.6769
LR_RF_SVM_SGD_	word(1gram)+char(2-	0.6761
MNB_SGD_XGB	word(1gram)+char(2-	0.676
MNB_SVM_SGD	word(1gram)+char(2-	0.6748
RF_MNB_SVM	word(1gram)+char(2-	0.6741
MNB_SVM_XGB	word(1gram)+char(2-	0.674
LR_SGD_XGB	word(1gram)+char(2-	0.6707
LR_SVM_SGD	word(1gram)+char(2-	0.6686
RF_SVM_SGD	word(1gram)+char(2-	0.6668
LR_SVM_XGB	word(1gram)+char(2-	0.666
LR_RF_SVM	word(1gram)+char(2-	0.6645
SVM_SGD_XGB	word(1gram)+char(2-	0.659
RF_MNB_XGB	word(1gram)+char(2-	0.6544
RF_SGD_XGB	word(1gram)+char(2-	0.6542
LR_RF_XGB	word(1gram)+char(2-	0.6538
RF_SVM_XGB	word(1gram)+char(2-	0.6438

Figure 7: Result of all models (Hindi)

The LR_RF_MNB_SGD_XGB ensemble achieved a micro F1-score of 0.68, indicating strong performance across various agricultural queries in the Hindi dataset. This reflects the ensemble's ability to generalize well even with class imbalance.

	precision	recall	f1-score	support
Cultural Practices	0.48	0.34	0.40	842
Fertilizer Use and Availability	0.47	0.62	0.53	520
Field Preparation	0.12	0.25	0.16	256
Government Schemes	0.43	0.72	0.54	404
Market Information	0.75	0.77	0.76	408
Nutrient Management	0.49	0.53	0.51	885
Plant Protection	0.87	0.74	0.80	3792
Varieties	0.49	0.80	0.61	270
Weather	0.91	0.80	0.85	2323
Weed Management	0.60	0.83	0.70	300
accuracy			0.69	10000
macro avg	0.56	0.64	0.59	10000
weighted avg	0.73	0.69	0.70	10000

Figure 8: Performance metrics of the LR_RF_MNB_SGD_XGB ensemble

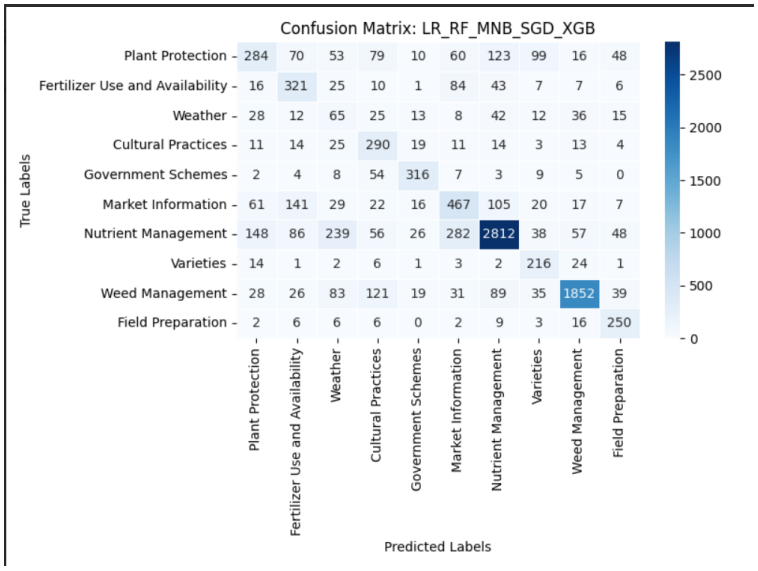

Figure 9: Classification results of the LR_RF_MNB_SGD_XGB ensemble

Gujarati Model (LR_RF_MNB_SGD_XGB Ensemble)

The following results show the performance of our classification models, with LR_RF_MNB_SGD_XGB emerging as the top performer.

Model	Feature	Micro F1
LR_RF_MNB_SGD_	word(1gram)+char(2-	0.6893
MNB_SGD_XGB	word(1gram)+char(2-	0.6828
LR_RF_MNB	word(1gram)+char(2-	0.6791
LR_MNB_SGD	word(1gram)+char(2-	0.6785
LR_MNB_XGB	word(1gram)+char(2-	0.6781
LR_MNB_SVM_SGD	word(1gram)+char(2-	0.6769
MNB_SVM_SGD	word(1gram)+char(2-	0.6745
LR_MNB_SVM	word(1gram)+char(2-	0.6745
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.6735
RF_MNB_SGD	word(1gram)+char(2-	0.6729
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.6698
LR_SVM_SGD	word(1gram)+char(2-	0.6698
RF_MNB_SVM_SGD	word(1gram)+char(2-	0.6693
LR_SGD_XGB	word(1gram)+char(2-	0.6687
LR_RF_SGD	word(1gram)+char(2-	0.6673
LR_RF_SVM_SGD_	word(1gram)+char(2-	0.6662
RF_MNB_SVM	word(1gram)+char(2-	0.6659
MNB_SVM_XGB	word(1gram)+char(2-	0.6647
LR_RF_SVM	word(1gram)+char(2-	0.6603
RF_SGD_XGB	word(1gram)+char(2-	0.6591
RF_SVM_SGD	word(1gram)+char(2-	0.6586
LR_SVM_XGB	word(1gram)+char(2-	0.6586
SVM_SGD_XGB	word(1gram)+char(2-	0.6575
RF_MNB_XGB	word(1gram)+char(2-	0.6574
LR_RF_XGB	word(1gram)+char(2-	0.6568
RF_SVM_XGB	word(1gram)+char(2-	0.6507

Figure 10: Result of all models (Gujarati)

The LR_RF_MNB_SGD_XGB ensemble achieved a micro F1-score of 0.68, showcasing strong and consistent performance across the Gujarati agricultural query dataset. The ensemble approach helps balance precision and recall even for lower-resource categories.

	precision	recall	f1-score	support
Cultural Practices	0.45	0.34	0.39	842
Fertilizer Use and Availability	0.46	0.63	0.53	520
Field Preparation	0.12	0.24	0.16	256
Government Schemes	0.47	0.74	0.57	404
Market Information	0.79	0.78	0.79	408
Nutrient Management	0.47	0.53	0.50	885
Plant Protection	0.88	0.74	0.80	3792
Varieties	0.48	0.79	0.60	270
Weather	0.90	0.80	0.85	2323
Weed Management	0.62	0.83	0.71	300
accuracy			0.69	10000
macro avg	0.56	0.64	0.59	10000
weighted avg	0.73	0.69	0.70	10000

Figure 11: Performance metrics of the LR_RF_MNB_SGD_XGB ensemble

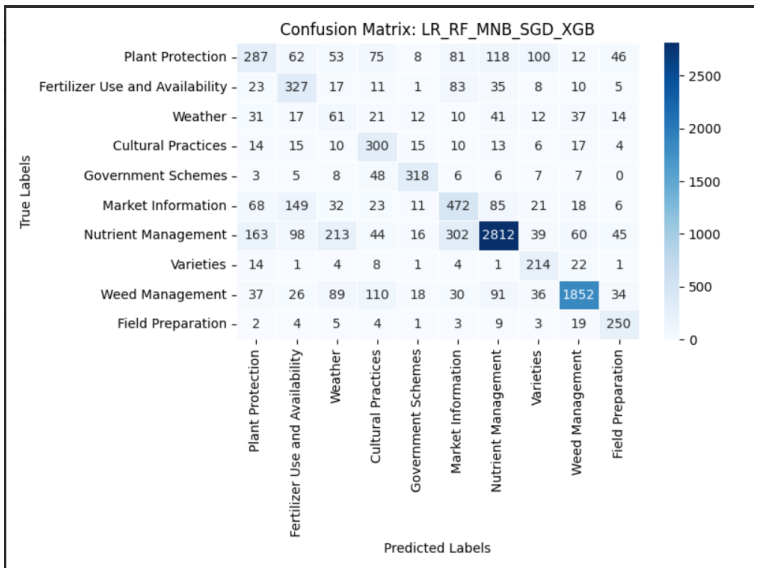

Figure 12: Classification results of the LR_RF_MNB_SGD_XGB ensemble

Multilingual Model (LR_RF_MNB_SGD_XGB Ensemble)

The following results illustrate the model performance on the combined multilingual dataset. LR_RF_MNB_SGD_XGB achieved the highest score among the tested ensembles.

Model	Feature	Micro F1
LR_RF_MNB_SGD_	word(1gram)+char(2-	0.4222333333
LR_RF_MNB	word(1gram)+char(2-	0.4212
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.4205
RF_MNB_SVM_SGD	word(1gram)+char(2-	0.4195333333
LR_RF_SGD	word(1gram)+char(2-	0.4187333333
LR_MNB_XGB	word(1gram)+char(2-	0.4185666667
RF_MNB_SVM	word(1gram)+char(2-	0.4178666667
LR_MNB_SVM_SGD	word(1gram)+char(2-	0.4177
LR_RF_MNB_SVM_	word(1gram)+char(2-	0.4174666667
LR_MNB_SVM	word(1gram)+char(2-	0.4166333333
LR_SGD_XGB	word(1gram)+char(2-	0.4156666667
LR_SVM_SGD	word(1gram)+char(2-	0.4152333333
LR_RF_SVM_SGD_	word(1gram)+char(2-	0.4151666667
RF_SVM_SGD	word(1gram)+char(2-	0.4142
MNB_SVM_XGB	word(1gram)+char(2-	0.4141333333
LR_RF_SVM	word(1gram)+char(2-	0.4123
SVM_SGD_XGB	word(1gram)+char(2-	0.4119
RF_MNB_XGB	word(1gram)+char(2-	0.4117666667
LR_SVM_XGB	word(1gram)+char(2-	0.4112666667
LR_RF_XGB	word(1gram)+char(2-	0.4111333333
RF_SGD_XGB	word(1gram)+char(2-	0.4108666667
RF_SVM_XGB	word(1gram)+char(2-	0.4084
RF_MNB_SGD	word(1gram)+char(2-	0.3227666667
MNB_SGD_XGB	word(1gram)+char(2-	0.3205333333
LR_MNB_SGD	word(1gram)+char(2-	0.32
MNB_SVM_SGD	word(1gram)+char(2-	0.3189

Figure 13: Result of all models (Multilingual)

The LR_RF_MNB_SGD_XGB ensemble attained a micro F1-score of 0.42. While lower than monolingual models, this indicates potential for improvement when handling mixed-language queries in a single model, possibly due to language diversity and vocabulary mismatch.

	precision	recall	f1-score	support
Cultural Practices	0.44	0.16	0.23	2526
Fertilizer Use and Availability	0.44	0.23	0.30	1561
Field Preparation	0.15	0.12	0.13	768
Government Schemes	0.31	0.36	0.34	1211
Market Information	0.74	0.57	0.64	1223
Nutrient Management	0.45	0.19	0.27	2654
Plant Protection	0.86	0.29	0.44	11376
Varieties	0.34	0.32	0.33	810
Weather	0.34	0.90	0.50	6969
Weed Management	0.25	0.35	0.29	902
accuracy			0.42	30000
macro avg	0.43	0.35	0.35	30000
weighted avg	0.57	0.42	0.40	30000

Figure 14: Performance metrics of the LR_RF_MNB_SGD_XGB ensemble

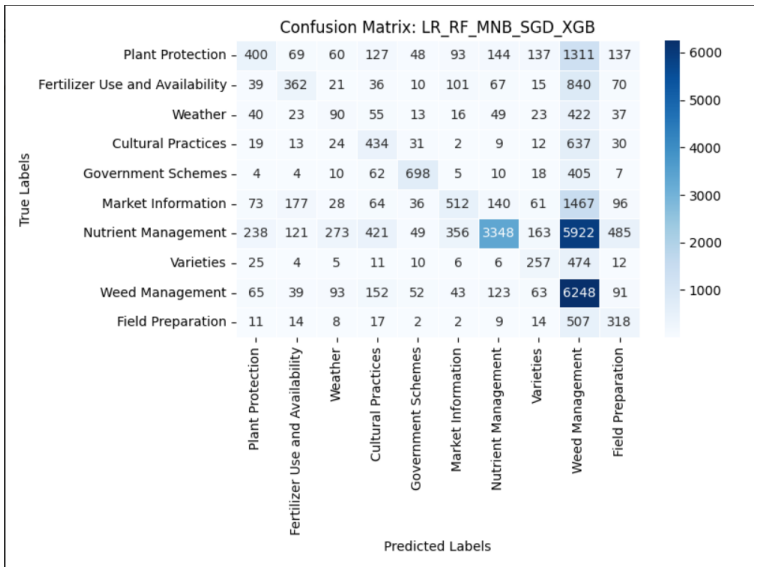

Figure 15: Classification results of the LR_RF_MNB_SGD_XGB ensemble

5.3 Error Analysis

To better understand the limitations of our models, we conducted an error analysis on the misclassified queries across all languages. The goal was to identify recurring patterns and potential areas for improvement.

- **Ambiguity in Queries:** Several misclassified queries contained ambiguous language or lacked sufficient context, making it difficult for the model to associate them with the correct category.
- **Overlapping Categories:** Errors often occurred between semantically similar categories, indicating the need for finer category definitions or hierarchical classification.
- **Crop Names as Dominant Features:** Queries that contain crop names are frequently classified into crop-specific categories. However, this often leads to misclassification when the query actually belongs to a different category. This suggests the model is over-relying on the presence of crop names as primary indicators for classification.
- **Code-mixed Language:** In the multilingual dataset, the presence of code-mixed queries—where multiple languages are used in a single sentence—led to confusion in token representation and reduced classification accuracy.
- **Imbalanced Data:** Categories with fewer training samples were more prone to misclassification, highlighting the model's difficulty in learning underrepresented patterns.
- **Translation and Transliteration Errors:** In the Gujarati and Hindi datasets, inconsistencies introduced during translation or transliteration resulted in noisy inputs, negatively affecting the model's predictions.

These observations suggest that future improvements can be made by enriching the training dataset, applying language normalization, and exploring advanced modeling techniques such as attention-based architectures and multilingual transformers.

6 Conclusion

In this work, we successfully developed a multilingual agricultural query classification system tailored for Hindi, Gujarati, and English queries. By employing a multi-algorithm approach with six different classifiers and their ensemble combinations, we systematically evaluated and identified optimal models for each language. Our system demonstrated promising classification performance across diverse agricultural categories, showcasing the effectiveness of ensemble learning in multilingual settings.

The project highlights the potential of machine learning and natural language processing techniques in bridging language barriers and improving information accessibility for farmers. Through a modular architecture, the system allows for easy integration of additional languages and models, supporting scalability and future enhancements.

For future work, we plan to explore deep learning techniques, such as transformer-based architectures (e.g., BERT, XLM-R), to further improve classification accuracy and handle complex linguistic variations across languages. Additionally, we aim to expand the system to support more regional languages, enabling broader coverage and inclusivity for farmers across different linguistic backgrounds.

7 Bibliography

[1] https://www.geeksforgeeks.org/text-classification-using-scikit-learn-in-nlp/

[2] https://dylancastillo.co/posts/text-classification-using-python-and-scikit-learn.html

[3] https://github.com/Pruthwik/SentimentAnalysis

[4] https://ssmt.iiit.ac.in/translatev4