Crop Recommendation System

Mrinal Singh[1], Lavi Jain[1], and Shubham Kapoor[1] Sanjay Kashyap[1]

Indian Institute of Technology Ropar, Rupnagar, Punjab - 140001, India
{2023csm1009,2023csm1005,2023csm1014,2023aim1013}@iitrpr.ac.in

Abstract. Innovations in the field of agriculture has been increasing rapidly. Conventional crop recommendation methods often rely on simplistic data aggregation techniques or averages, overlooking the critical influence of weather patterns throughout the entire crop cycle, from sowing to harvesting. In this study, we propose a novel approach to crop recommendation that integrates comprehensive weather pattern analysis. Leveraging data obtained from the International Crops Research Institute for the Semi-Arid Tropics (ICRISAT), we meticulously assessed the ideal weather conditions for optimal crop yield, spanning the entirety of the crop cycle based on historical data. Our methodology employs advanced machine learning techniques, specifically autoencoder and classification models, to process the collected data and generate precise crop recommendations. By prioritizing the consideration of weather patterns throughout the crop cycle, our approach offers a significant difference from traditional recommendation systems. Additionally, we have developed a user-friendly application that empowers farmers to access crop recommendations effortlessly with just the click of a button. This application seamlessly integrates with weather and location APIs, to provide tailored recommendations based on real-time environmental conditions specific to their region. By harnessing the power of technology, our application enables farmers to make informed decisions, thereby optimizing their agricultural practices and maximizing crop yields.

Keywords: Crop Recommendation · Classification · Autoencoder

1 Introduction

Agriculture is the backbone of many economies around the world, and ensuring food security for a growing global population remains a significant challenge. With limited cultivable land and the increasing effects of climate change, selecting the right crops to plant and implementing proper farming practices has become increasingly important. Traditionally, farmers have relied heavily on experience and local knowledge to determine which specific crops to grow. However, this approach may prove inadequate, particularly in regions facing rapidly changing environmental conditions and the pressing need for higher yields.

Over time, various sectors have witnessed significant improvements through the application of machine learning techniques, providing hope for better outcomes in the agricultural domain as well. These advanced algorithms can offer critical insights to support crop selection decisions and optimize management practices by leveraging historical data on crop yield patterns, soil quality, and other relevant factors.

This research paper presents a machine learning-based crop recommendation system designed to assist farmers in making informed choices about crop cultivation within their specific environmental conditions, available resources, and desired objectives. By harnessing the power of machine learning algorithms and analyzing diverse datasets, the proposed system aims to provide tailored recommendations to farmers, guiding them in selecting the most suitable crops, determining optimal planting times, and implementing effective cultivation methods.

The development of such a data-driven crop recommendation system holds the potential to revolutionize agricultural practices, particularly in regions where traditional knowledge may be insufficient or where environmental conditions are rapidly evolving due to climate change. By empowering farmers with insights derived from machine learning models, this system can contribute to increased crop yields, improved resource efficiency, and enhanced food security.

The remainder of this paper is structured as follows Section 2 provides an overview of related work in the field of crop recommendation systems and machine learning applications in agriculture. Section 3 outlines the methodology employed in this research, including data collection, preprocessing, and the specific machine learning algorithms used. Section 4 presents the results and evaluation of the proposed system. Finally, Section 5 concludes the paper and summarizes the key findings.

2 Related Works

Several studies have explored the application of machine learning techniques to tackle challenges in the agricultural domain. These research efforts have focused on leveraging historical data and environmental factors to provide insights and recommendations for crop cultivation.

One line of work has proposed crop recommendation systems using random forest classifiers. These approaches utilize soil properties, weather data, and crop yield information to recommend suitable crops for a given region. However, some of these studies were limited to specific geographical areas and did not consider long-term climate forecasts[1].

Other researchers have developed crop selection models based on artificial neural networks. These models incorporate soil characteristics, rainfall patterns, and temperature data to recommend crops for different regions. While accounting for various environmental factors, some did not consider the temporal aspect of weather data throughout the crop growth cycle[2][3][4].

Hybrid approaches combining techniques like k-nearest neighbors and support vector machines have also been explored for crop yield prediction. These models utilize satellite imagery, soil data, and historical yield records to estimate crop yields for specific crops[5]. However, the focus was primarily on yield prediction rather than providing crop recommendations based on forecasted weather conditions.

Deep learning techniques, such as convolutional neural networks, have been applied to predict crop yields based on satellite imagery and weather data[6]. These models have shown promising results in yield estimation but did not extend to crop recommendation based on future climate forecasts.

While these existing studies have contributed to the field of precision agriculture, there is a need for a more comprehensive approach that considers both historical data and long-term climate forecasts to provide tailored crop recommendations. The proposed research aims to address this gap by developing a machine learning-based crop recommendation system that incorporates weather forecast data spanning the entire crop growth cycle, enabling farmers to make informed decisions about crop selection based on anticipated environmental conditions.

3 Methodology

3.1 Dataset Collection

In this research, we assembled a dataset to train the model and analyze the potential influence of weather conditions on crop yield across diverse Indian districts from sowing to harvesting. This section details the acquisition process for both crop yield and weather data.

Crop Yield Data The foundation of our dataset is historical crop yield data encompassing district-wise and season-wise information on crop production. We obtained this data for the period spanning 1997 to 2015 from a reliable source (ICRISAT:- http://data.icrisat.org/dld/src/additional.html). This timeframe provides a valuable 19-year perspective on historical crop performance at the district level. The data itself consists of numerical values representing the yield of crops for various districts within India. Notably, the data is further categorized by season within each year, allowing for a more nuanced analysis that considers seasonal variations in crop production patterns.

Weather Data Acquisition To complement the rich crop yield data and gain insights into potential environmental influences, we collected historical weather information for the relevant districts. We leveraged the Visual Crossing weather API (https://www.visualcrossing.com/) to achieve this. This API serves as a powerful tool, offering programmatic access to historical weather data for a vast number of locations across the globe. The weather data obtained through the Visual Crossing API encompasses a range of parameters critical for understanding the prevailing weather conditions during each crop season within a specific district and year. These parameters include temperature, humidity, dew point, precipitation, cloud cover, solar radiation, solar energy, and UV index. By incorporating this data, we gain valuable insights into the environmental context surrounding crop growth within each district-year combination.

Data Collection Script To streamline the process of fetching weather data using the Visual Crossing API, we developed a Python script. This script acts as an automated tool, taking district names, crop types and years as input. It then interacts with the API to retrieve daily weather data for the desired parameters within the relevant timeframe (sowing day to harvest day) for each district-year combination.

3.2 Data Preprocessing

There are multiple datasets for every crop which has parameters temperature, humidity, dew point, cloud cover, solar radiation, solar energy, and UV index. First, we normalized the dataset using min max normalization where minimum and maximum is calculated after iterating through every dataset so the relation between different dataset remains. After normalization in each dataset, we selected 180 rows in sequential manner and appended them in single row by renaming their parameters to parameter type and day number.

Finally we appended each dataset into a single one where each rows represent data for different crop type and location. The number of features increased from 7 to 1260 after implementing this method. The relation between days as well as 7 features for different crops are captured through this implementation.

3.3 Autoencoder and Classification

Neural network-based models known as autoencoders are used in unsupervised learning to find underlying correlations in data and represent data in a reduced dimension. In order to train a neural network model, the autoencoders formulate unsupervised learning issues as supervised learning problems.

PCA or principal component analysis is not preferred over autoencoders because as we can see in the above diagram autoencoders cover non-linear data dependencies, thus are a better way than PCA for dimensionality reduction.[**?**]
We used the trained autoencoder part to further train our classification model. We designed a neural network architecture such that we impose a bottleneck in the network which forces a compressed knowledge representation of the original input. It is referred as the latent space or bottleneck of the autoencoder architecture.

In the fig 2 representing autoencoder architecture , input is a single row from the dataset of size 1026*1 in yellow color . Convolution 1D layers in pink color and then we flatten the data after that dense layers(blue color) were used. The output layer from the final dense layer is also of the size 1026*1 for autoencoder. After that we used the trained autoencoder to further train our classifier model which has been made up of dense layers. The latent space or the bottleneck from the autoencoder is used as the input to the classifier model.

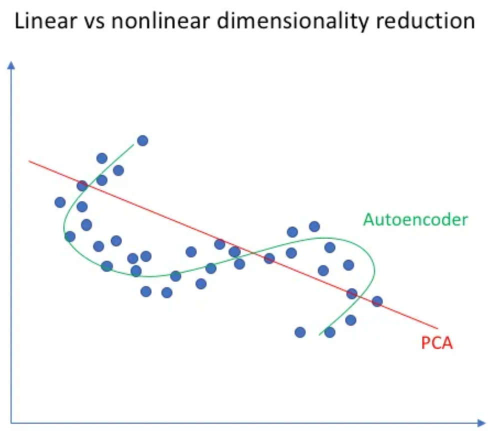


Fig. 1: PCA and Autoencoders

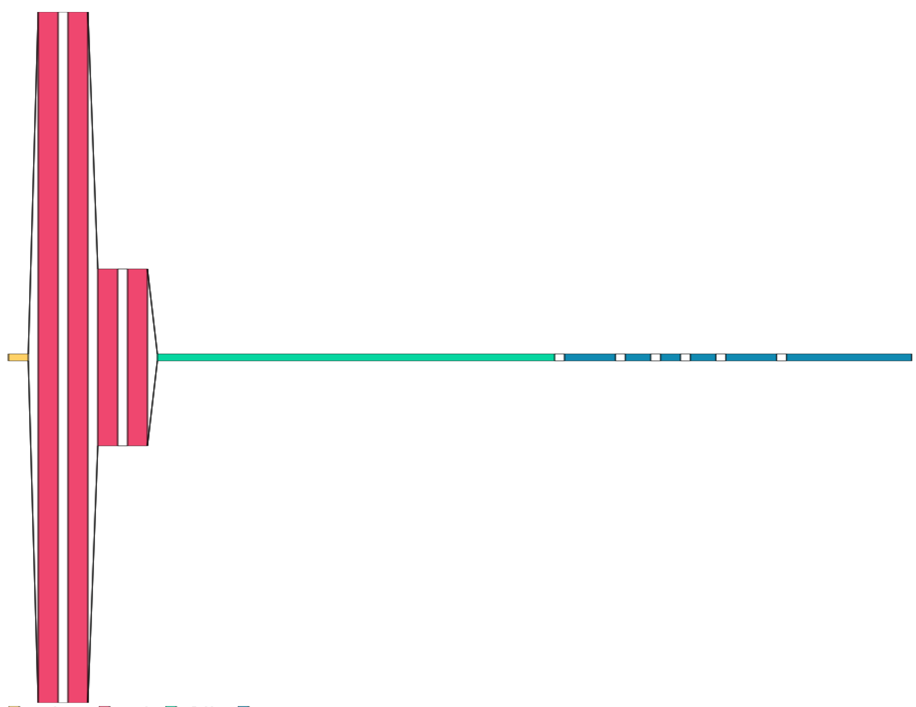


Fig. 2: Autoencoder Architecture

3.4 Working of Application

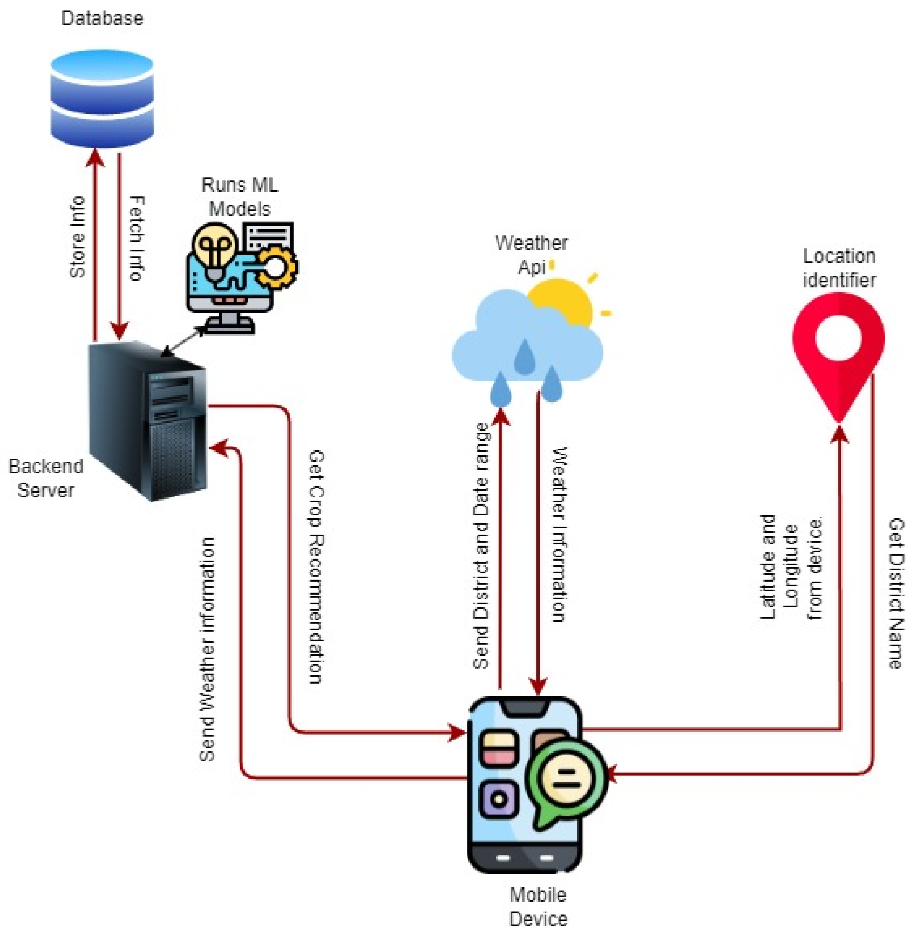


Fig. 3: High Level Application overview

The application operates through a collaborative effort between the user interface, a backend server, and deployed machine learning models. Built with React Native, the user interacts with a mobile-friendly interface designed for ease of use.

On the backend, Node.js and Express.js handle data processing and communication with the Python models trained using TensorFlow. These models reside on a separate server and are responsible for generating crop recommendations.

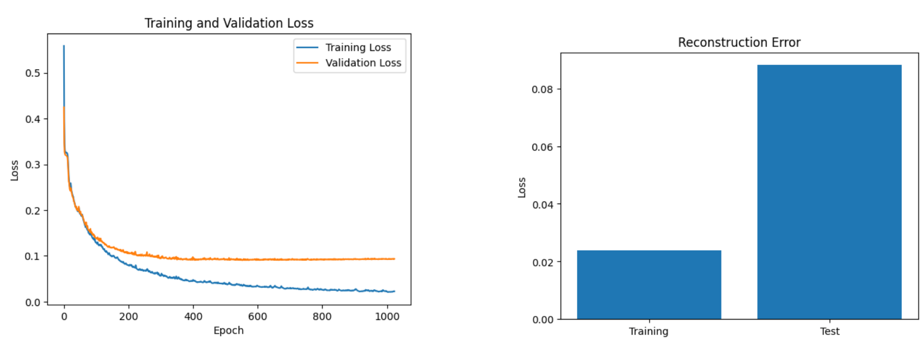
Data acquisition begins with the application capturing the user's location through the device's built-in location services. This information, in the form of latitude and longitude, is then translated into a corresponding district using a geocoding library. Finally, an API endpoint, such as Visual Crossing, retrieves real-time weather data of approximately future 6 months specific to the user's district.

Once this data is collected, the user's location and the retrieved weather information are sent to the backend server via an API request. The server then interacts with the deployed Python models, feeding them the received data. These models, trained on vast agricultural datasets, analyze the information and generate tailored crop recommendations. Finally, the server transmits these recommendations back to the mobile application.

The user interface plays a crucial role in presenting the recommendations. After prompting the user for location access and potentially capturing additional data, the application displays the recommended crops in a clear and concise format. This architecture provides a seamless user experience by combining the strengths of React Native for mobile development with the robust functionalities of Node.js and Python for backend processing and machine learning. It empowers users with informed decision-making by providing data-driven crop recommendations tailored to their specific location and weather conditions.

4 Results and Analysis

Accuracy of Autoencoders are measured in terms of reconstruction error which means the error in between the output and the expected output which is the input here. Reconstruction error in autoencoders can range from 0 to infinity. Here, we are getting reconstruction error of 0.08 on test set and 0.03 on training set.



(a) Training and Validation Loss (b) Reconstruction Error

Fig. 4: AutoEncoder Reconstruction Error

We selected 10 major crops for this project based on their production in Punjab, India. The crops are arhar, bajra, barley, gram, jowar, maize, ragi, rice, urad and wheat. We trained our classification model on these crops using input from the latent space of Autoencoder. The accuracy of the model comes out to be 84% on the test set for the classification model. The accuracy during training has been presented in the figure 5.

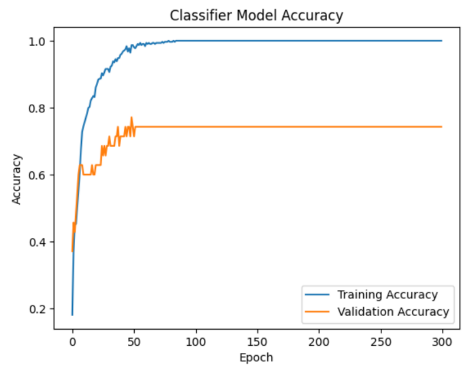
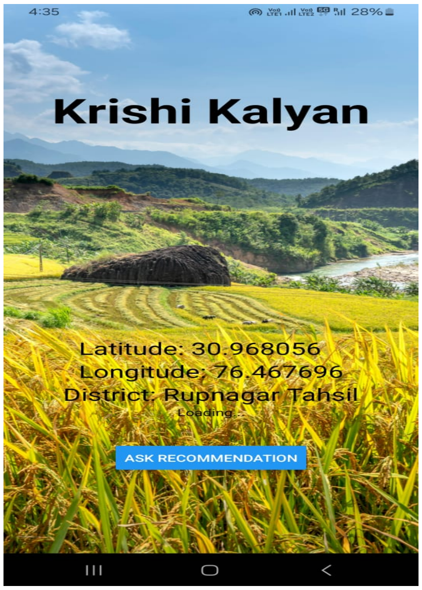


Fig. 5: Classification Model Training and Validation Accuracy

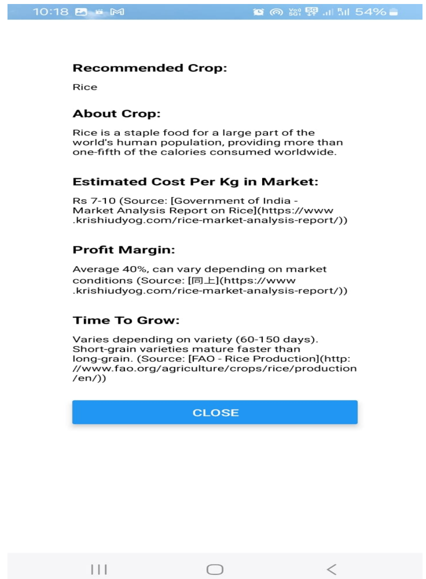
Finally, we integrated this trained model in our application called Krishi Kalyan for users. The application accesses users location and based on their location it fetches the future 180 days climate data. That data is then preprocessed accordingly and used as an input to the machine learning model. The output of the model is then shown to the user as seen in the figure 6.

5 Conclusion

In conclusion, our project endeavors to harness the power of machine learning to provide valuable insights into the complex interplay between weather conditions and crop yield across various districts in India. This project introduces a novel concept where users can explore the cultivation of non-native crops under similar weather conditions from the sowing period to harvesting. We curated a comprehensive dataset encompassing historical crop yield data and corresponding weather information from reliable sources. This dataset spans multiple years and districts, enabling a thorough analysis of crop performance under diverse environmental conditions. We leveraged neural network-based models, particularly autoencoders. The trained autoencoder served as a crucial component enabling us to extract meaningful features and achieve accurate crop classification. We developed a user-friendly mobile application, named Krishi Kalyan, which provides personalized crop recommendations based on real-time weather data.



(a) HomeScreen



(b) Crop Recommended based on weather data.

Fig. 6: Crop recommendation app based on weather data.

Our models exhibited promising performance, with the autoencoder achieving low reconstruction errors and the classification model achieving an accuracy of 84%. Real-world Application of our work is that the Krishi Kalyan application empowers farmers and stakeholders in the agricultural sector with data-driven insights, facilitating informed decision-making regarding crop selection. Future work, Newer models can be trained to work on dynamic size datasets of crops which takes 5 months to 10 months to grow. The dataset can be increased to include more crops for recommendation to users. We are able to provide single recommendation to users after increasing the dataset, machine learning models can be trained to recommend multiple crops.

References

1. M. Pathan, N. Patel, H. Yagnik, and M. Shah, "Artificial cognition for applications in smart agriculture: A comprehensive review," Artificial Intelligence in Agriculture, vol. 4, pp. 81-95, 2020.
2. T. Setiali, F. Noviyanto, H. Hardianto, A. Tarmuji, A. Fadili, and M. Wibowo, "Implementation of naive bayes method in food crops planting recommendations," Int. J. Sci. Technol. Res., vol. 9, no. 02, pp. 4750-4755, 2020.
3. T. K. Mishra, S. K. Mishra, K. J. Sai, S. Peddi, and M. Surusomayajula, "Crop recommendation system using support vector machine considering indian dataset," in

Advances in Distributed Computing and Machine Learning: Proceedings of ICAD-CML 2022, Springer, 2022, pp. 501-510.
4. K. G. Sandhya, S. Vemuri, K. S. Deeksha, and T. Anvitha, "Crop recommendation system using ensembling technique," in 2022 International Conference on Breakthrough in Heuristics And Reciprocation of Advanced Technologies (BHARAT), IEEE, 2022, pp. 55-58.
5. A. Chougule, V. K. Jha, and D. Mukhopadhyay, "Crop suitability and fertilizers recommendation using data mining techniques," in Progress in Advanced Computing and Intelligent Engineering, Springer, 2019, pp. 205-213.
6. A. Motwani, P. Patil, V. Nagaria, S. Verma, and S. Ghane, "Soil analysis and crop recommendation using machine learning," in 2022 International Conference for Advancement in Technology (ICONAT), IEEE, 2022, pp. 1-7.
7. Indian Krishi repositories, https://krishi.icar.gov.in/
8. ICRISAT, https://www.icrisat.org/
9. AutoEncoders introduction, https://towardsdatascience.com/introduction-to-autoencoders-7a47cf4ef14b
10. Tensorflow AutoEncoders, https://www.tensorflow.org/tutorials/generative/autoencoder
11. Intro to Autoencoders, https://www.jeremyjordan.me/autoencoders/