https://doi.org/10.33472/AFJBS.6.6.2024.1277-1289

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

From Data to Harvest - Optimizing Crop Selection with Crop Recommendation System

Isaac Malsawma[1], R. Chawngsangpuii[2*] and Lalhmingliana[3]

[1]Department of Information Technology, Mizoram University,Aizawl, India
[2]Department of Information Technology, Mizoram University,Aizawl, India
[3]Department of Information Technology, Mizoram University,Aizawl, India

***Corresponding author Dr. R. Chawngsangpuii,** Email: mzut126@mzu.edu.in

Article Info

Volume 6, Issue 13, July 2024

Received: 02 June 2024

Accepted: 30 June 2024

Published: 24 July 2024

doi: 10.33472/AFJBS.6.13.2024.1277-1289

ABSTRACT:

As the foundation of the source of livelihood for a significant amount of the population and the Indian economy, agriculture is of paramount significance in the country. Being a major employer and contributor to India's GDP, farmers are crucial to the country's economic health. They produce a wide range of crops and engage in a wide range of agricultural activities, all of which contribute significantly to food security. Small landholdings, outmoded agricultural methods, water shortages, and market swings are just a few of the obstacles that Indian farmers encounter, despite the significance of their work. One such approach that shows promise in solving this problem is the combination of Machine Learning and Data Analytics. Precision agriculture aims in improving the efficiency, accuracy, and sustainability in farming by using data-driven approaches and cutting-edge technology. We provide a crop suggestion system in this article with the goal of assisting Indian farmers in making better judgments regarding the crops to plant by examining the geographical features of their fields, including other parameters such as soil type and environmental conditions, using data analytics.

Keywords: Machine learning, Data visualization & analysis, Classification, Crop recommendation, Gradient boosting

1. Introduction

Agriculture holds immense importance in India as it is the foundation of the country's economy and sustains the livelihoods of a significant portion of its population, especially in rural areas. Contributing to 19% of India's GDP and employing over 50% of its workforce, agriculture and its allied sectors are a cornerstone of India's economy (Mahesh *et al.*, 2023; Bais and Bahadur 2023).

Agriculture is not only a vital component of India's economy but also deeply intertwines with its cultural fabric, societal impact, and political landscape. Its significance goes beyond economic contributions, impacting the livelihoods and well-being of crores of Indians. Thus, for Indian farmers, their well-being and prosperity are paramount for the nation's overall economic health and development. Ensuring the sustainability and growth of the agricultural sector is crucial for India's overall development and food security (Kumar, 2019).

Despite its importance, Indian agriculture faces numerous challenges, including small landholdings, outdated farming techniques, water scarcity, and market fluctuations. Modernization techniques, technological advancements, and political reforms are necessary to address these issues and enhance agricultural productivity (Priyadarshini and Abhilash, 2020). Precision farming entails employing technology and data-driven approaches to enhance various facets of agricultural practices, rendering them more efficient, precise, and sustainable (Singh, 2022). In this paper, we intend to implement this using a crop recommendation system that specifically focuses on providing farmers with tailored guidance on crop selection based on various factors, encompassing soil conditions, climate patterns, historical data, and market trends.

These systems use data analytics, and machine learning algorithms, and often employ sensors and remote sensing technologies to assess and analyse information related to the farm environment. By considering various parameters, A crop recommendation system seeks to assist farmers in choosing the right crops for their particular circumstances, leading to improved yields, resource efficiency, and overall productivity (Elbasi *et al.*, 2023).

2. Literature Review

A great deal of effort has been made in this area, as seen by the many research papers that have been published. Data science and machine learning algorithms have been used to produce precision agriculture tools (Araújo *et al.*, 2023; Sagana *et al.*, 2023 and Sidhu *et al.*, 2021), which assist farmers in making data-driven decisions about what crops to cultivate on their property (Roy *et al.*, 2023 and Meshram *et al.*, 2021). Many different kinds of crops have been investigated in many different places around the Indian subcontinent.

A crop yield prediction system was presented in (Kumar *et al.*, 2020) which makes use of historical information from the past, like rainfall, pH, temperature, and humidity to predict the output of various crops. The dataset used contained various types of crops across different locations in India. This data was then represented in a graphical format which was then used to extract correlation between various attributes. The system used random forest algorithms and decision trees as machine-learning techniques. Using random forest algorithm leads to a higher accuracy score and therefore, results in a higher chance of accurately predicting the forecasted yield of a particular crop.

A system was developed (Rajak *et al.*, 2017) that predicts the crop best suited to a particular type of soil at a location. The dataset used particularly contained soil-specific attributes and pertained to specific districts of Maharashtra as it was derived from a soil testing lab in Maharashtra. The dataset contains various attributes about the soil like permeability, depth, color, texture, pH, erosion, drainage, and water holding. It takes several crops like sorghum, paddy, cotton, bananas, pulses, groundnut, sugarcane, vegetables, and coriander into consideration. Learning models used to develop the system were: Support Vector Machine (SVM), Naive Bayes, Random forest and Artificial Neural Network.

Doshi *et al.*, (2018) suggested AgroConsultant, a sophisticated system which works depending on the farm's location, the type of soil, the sowing season, and weather conditions such as temperature and rainfall, choose a crop to sow with knowledge by Indian farmers. The system consisted of two subsystems: Crop recommender and Rainfall predictor. The technique was applied to fifteen more minor crops in addition to the five major crops (bajra, wheat, rice, jowar, and maize). The dataset comprised the following attributes: temperature, precipitation, area parameters, soil type, top-soil thickness, soil PH, and aquifer thickness. Various machine learning techniques were employed, including Neural Network, Decision Tree, Random Forest and K-NN. Neural Network gave the best result.

Integrating machine learning and IoT was suggested that could help in the prevention of soil degradation and the maintenance of healthy crops (Gosai *et al.*, 2021). This is achieved through the utilization of an assortment of sensors capable of discerning factors including moisture content, pH, NPK, and soil temperature. Proposing a crop that would flourish in a particular location is the objective. Crop health is maintained and evaluated through the system's surveillance of soil moisture, pH, temperature, humidity, and NPK concentrations. Numerous machine learning approaches, notably Logistic Regression, Random Forest, Naïve Bayes, XGB, Decision Tree and Support Vector Machine were implemented by this system.

Another work combines two previously discussed systems which can predict crop yield and recommended an appropriate crop (Israni *et al.*, 2022). By predicting the yield of multiple crops, a farmer can gain valuable insights and he can choose the suitable crop to grow with the help of a crop recommendation system. The dataset they used constituted region-specific attributes which are collected from districts of Karnataka and the following crops were considered - Arecanut, cotton, Jowar, Maize (Corn), Bajra, and Rice. Machine Learning algorithms like Ridge Regression and XGB Regressor were used for the crop yield predictor and LightGBM was used for the crop recommendation system. Hyperparameter Tuning technique was used on these models to achieve better accuracy.

3. Methodology

The figure below illustrates the architecture of the system. The methods applied in the system are discussed below and summarized in "figure 1".

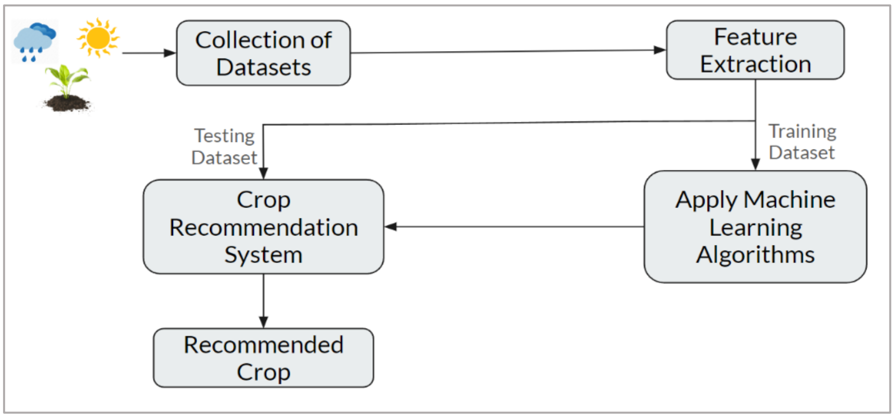

Figure 1. Block schematic of the suggested system

1. Dataset Collection

We plan to utilize a complete dataset that we acquired from (Ingle, 2020) for this system, which includes variables such as humidity, soil pH, rainfall level, temperature, and the concentrations of phosphorus (P), nitrogen (N), and potassium(K), i.e., a total of 7 attributes. 2200 data examples from historical records collected till the year 2020 from various parts of India are included in the collection. The aforementioned parameters are provided for 22 different crops namely orange, musk melon, papaya, pigeon peas, rice, banana, maize, kidney beans, pomegranate, chickpea, watermelon, black gram, coconut, cotton, coffee, grape, jute, lentil, mango, moth beans, mung beans, and papaya.

2. Data Pre-Processing

To maximize the application of algorithms for machine learning, processing data is an essential initial step. The data is collected in its original, unprocessed state from several sources. Some data fields may be duplicated, inconsistent, or incomplete. Therefore, it is essential to use various data pre-processing procedures at this phase to filter and analyse the data (Garcia and Herrera, 2015; Garcia *et al.*, 2016).

3. Data Visualization and Analysis

A key component of data science is data visualization, which is the graphic representation of data using graphs, charts, and maps to reveal trends, offer insights, and effectively convey findings. Data visualization tools facilitate an avenue to see and understand trends & patterns and to single out anomalies in the data (Ajibade and Adediran, 2016; Keim and Kriegel, 1996). Data visualization tools and technologies are quintessential to analysing a plethora of information which in turn helps us make data-backed decisions.

The following plots were generated through Jupyter Notebook (Project Jupyter Documentation, 2015) with the help of libraries such as matplotlib (Tosi, 2009) and seaborn (Waskom, 2021).

Nitrogen Analysis

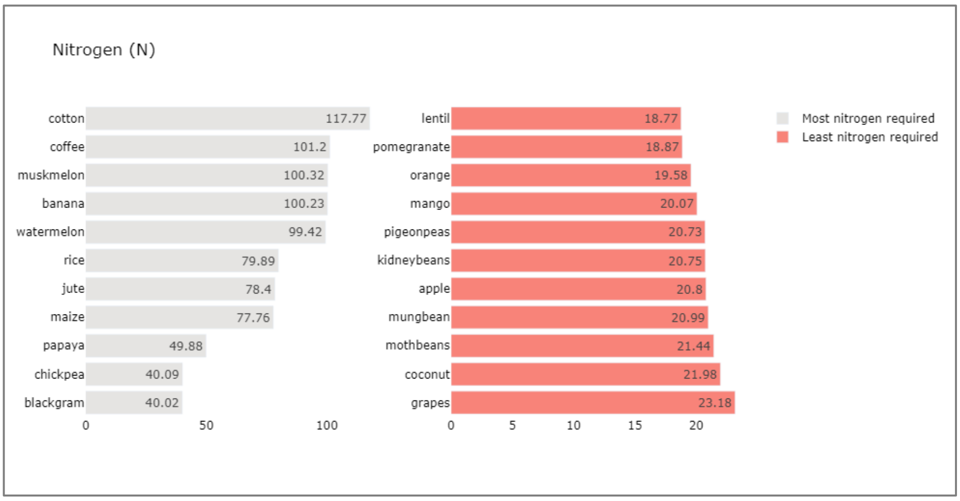

Figure 2. Bar chart featuring crops requiring most and least nitrogen

We can see in “figure 2” that cotton, coffee, muskmelon, banana, and watermelon require the highest amount of nitrogen while lentils, pomegranates, and oranges require the least.

Phosphorus Analysis

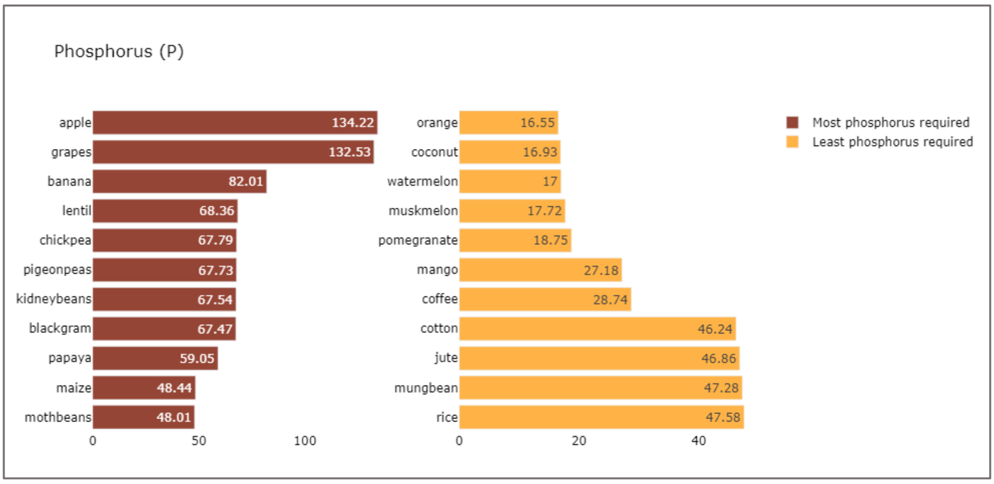

Figure 3. Bar chart featuring crops requiring most and least phosphorus

It is observed from "figure 3" that apples and grapes specifically require a high amount of phosphorus. On the other hand, we have orange, coconut, watermelon, musk melon, and pomegranate requiring very little phosphorus.

Potassium Analysis

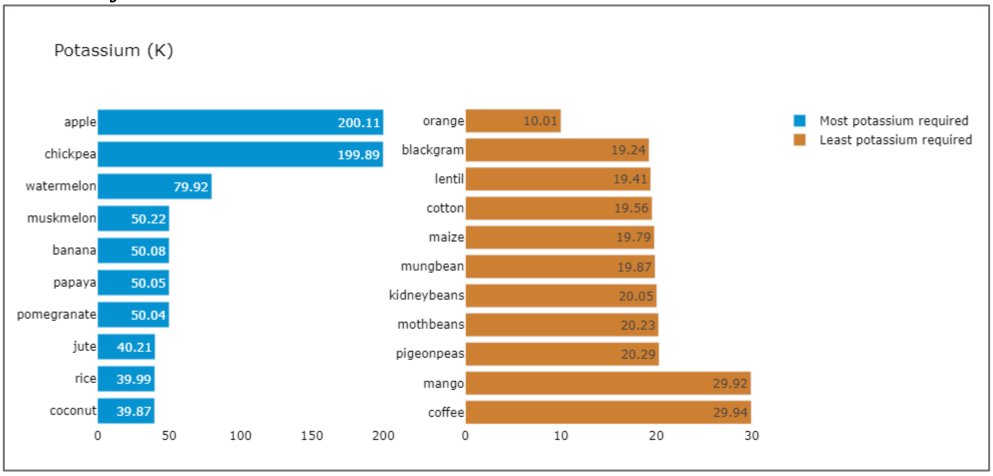

Figure 4. Bar chart featuring crops requiring most and least potassium

It can be seen from "figure 4" that apples and grapes likewise need a high concentration of potassium in the soil, following a similar trend to the previous graph. In line with the trend, oranges seem to require meagre amount of potassium.

NPK Comparison

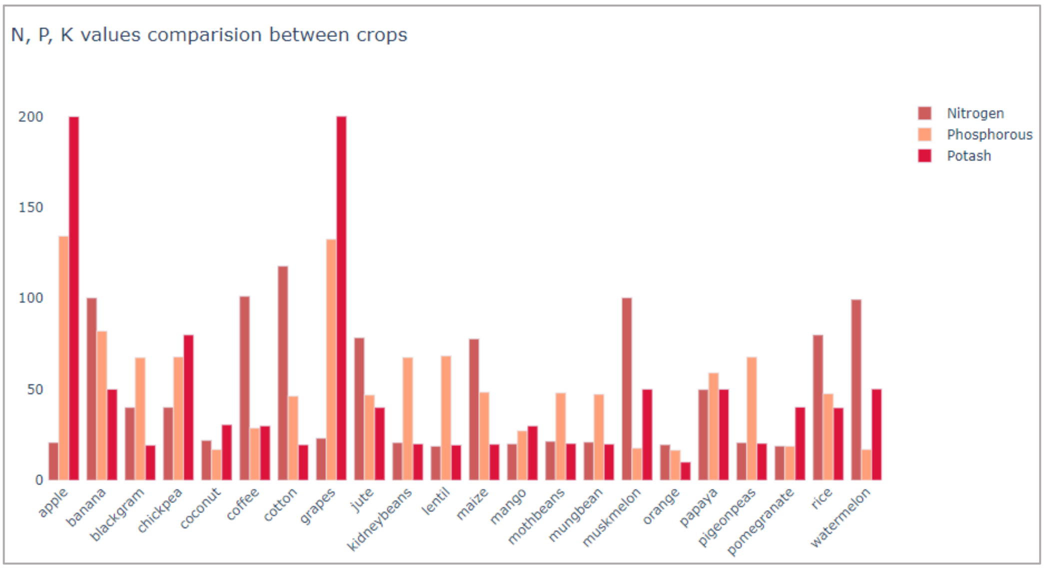

Figure 5. Bar chart comparing NPK values of various crops

A comparison of the potassium, phosphorus, and nitrogen levels required across the 22 different crops is given in “figure 5”. We can observe that kidneybeans, lentils, and pigeonpeas have very similar values for their N, P, and K requirements. The same observation can also be said for mothbeans and mungbeans. This information could imply that the selection of these particular crops would depend on their other features.

Rainfall, Temperature, and Humidity Analysis

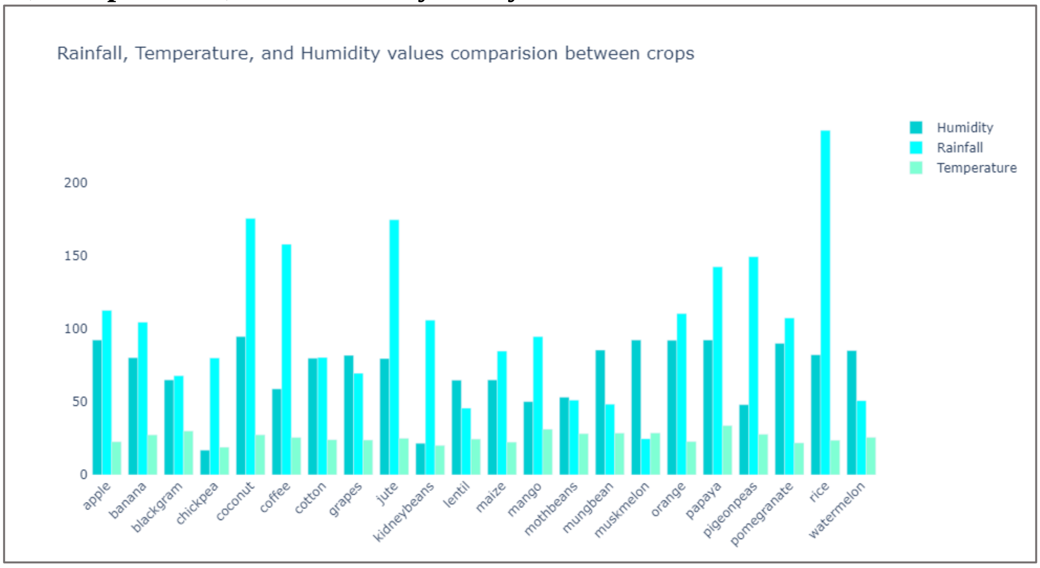

Figure 6. Bar chart comparing rainfall, temperature, and humidity requirements

In the comparison between rainfall, temperature, and humidity in “figure 6”, we can observe that rice has the highest bar. This can be attributed to rice being a Kharif crop that requires high temperatures (over 25 degrees), high humidity, and high rainfall with over 100cm per annum. Chickpeas on the other hand have the shortest bar due to them being relatively drought-tolerant since they possess long taproots which can draw out water from deep within the soil.

Correlation Matrix

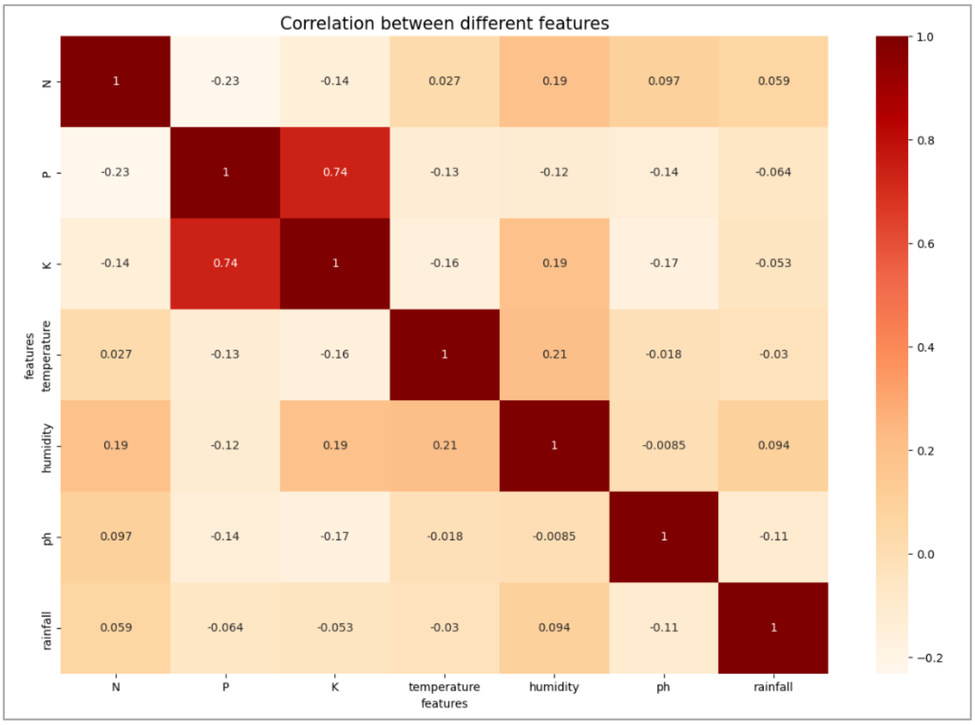

Figure 7. Heatmap showing correlation between different features

The heatmap of the various features of the dataset can be seen in “figure 7”. We can get insights into the correlation between different features with the help of this map with darker colours representing higher levels of correlation.

We observe that potassium and phosphorus have a high level of correlation, which we also observed in the earlier graph where apples and grapes required high levels of both nutrients.

Machine Learning Algorithms

When working on a machine learning process, it is common to test many algorithms to determine the most effective one for a particular use case. The decision is influenced by elements such as the characteristics of the data, the kind of issue, interpretability needs, and computing constraints (Mahesh, 2020). Hence, we want to use a diverse array of machine learning methods to achieve a well-rounded approach.

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=1,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)
```

```
DecisionTrees's Accuracy is:  92.27272727272727
```

Figure 8. Accuracy Value given by Decision Tree

Decision trees are a type of machine learning method used for classification that divides a dataset into subgroups according to the values of input attributes. The idea is to build a structure that looks like a tree, where each node in the leaf indicates a class label and each interior node indicates a choice based on a characteristic The decision-making process involves recursively splitting the data until a stopping criterion is met as given in “figure 8”. The splits are determined by choosing the features that best separate the classes, often using measures like Gini impurity or information gain (Song and Lu, 2015).

Gaussian Naive Bayes

```
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x*100)
```

```
Naive Bayes's Accuracy is:  99.54545454545455
```

Figure 9. Accuracy Value given by Gaussian Naive Bayes

One method of categorizing data using probabilistic machine learning is Gaussian Naive Bayes. It was created as a countermeasure to the Naive Bayes technique and is specially tailored for datasets with continuous characteristics that, presumably, follow a normal distribution. Gaussian Naive Bayes assumes the features are conditionally independent, which is considered "naive" given the class label, which makes the probability calculation easier (Kumar *et al.*, 2014). This method determines the probability of witnessing the feature values in order to provide predictions by using prior probabilities and a dataset that contains attribute values for every class as given in “figure 9”.

Support Vector Machine

```
from sklearn.svm import SVC

SVM = SVC(gamma='auto')

SVM.fit(Xtrain,Ytrain)

predicted_values = SVM.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x*100)
```

```
SVM's Accuracy is:  9.772727272727273
```

Figure 10. Accuracy Value given by Support Vector Machine

Support Vector Machine (SVM) is a helpful supervised machine-learning technique for regression and classification tasks. In order to do this, it locates the hyperplane in three dimensions that efficiently separates the input data into many classes. Enhancing the boundary which is the distance between each class's nearest data point and the hyperplane is the ultimate objective of the support vector machine (SVM) technique. This method can handle both linear and non-linear interactions and complex decision boundaries (Cortes and Vapnik, 1995) since it makes use of kernel functions as shown in “figure 10”.

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=2)

LogReg.fit(Xtrain,Ytrain)

predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x*100)

Logistic Regression's Accuracy is:  97.04545454545455
```

Figure 11. Accuracy Value given by Logistic Regression

Another kind of supervised machine learning technique is logistic regression. Its primary use is in classification problems with binary target classes. Logistic Regression is primarily employed for classification tasks, contrary to its name, instead of regression tasks as given in “figure 11”. Using the logistic function, the link between the input characteristics and the likelihood of falling into a specific class is predicted using logistic regression. The algorithm predicts the probability that an instance falls into the positive class, and a decision boundary is set to classify instances into either the positive or negative class based on a chosen threshold (Lavalley, 2008).

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x*100)

RF's Accuracy is:  99.31818181818181
```

Figure 12. Accuracy Value given by Random Forest

In the domains of classification and regression, Random Forest emerges as an exceptional ensemble machine learning methodology. By employing bootstrapped sampling, a forest of decision trees is generated, with each tree being trained on a unique and randomly selected subset of the data. Randomization is implemented as the method constructs the trees; at each node, a randomly selected subset of attributes is considered as shown in “figure 12”. The outcomes derived from each tree are aggregated in order to generate a forecast, while for classification purposes the class that receives the most votes are chosen (Lantz, 2015).

eXtreme Gradient Boosting (XGBoost)

Belonging to the gradient boosting framework family, Extreme Gradient Boosting, or XGBoost, is a robust and highly effective machine learning technique. It really helps with regression and classification problems. The objective of XGBoost's cascade of decision trees is to correct the errors made by preceding trees. To achieve the minimization of a loss function, gradient descent optimization is utilized. XGBoost is distinguished by its ability to process missing data, its regularization strategies to prevent overfitting, and the parallelization of training to accelerate calculations as shown in "figure 13". In numerous practical applications and machine learning competitions, XGBoost has garnered acclaim for its lightning-fast execution, scalability, and exceptional predicted precision (Chen and Guestrin, 2016).

```
import xgboost as xgb
XB = xgb.XGBClassifier()
XB.fit(Xtrain,Ytrain)

predicted_values = XB.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('XGBoost')
print("XGBoost's Accuracy is: ", x*100)
```

```
XGBoost's Accuracy is:  99.54545454545455
```

Figure 13. Accuracy Value given by XGBoost

Light Gradient Boosting Machine (LightGBM)

Microsoft developed the Light Gradient Boosting Machine, or LightGBM, which is a very effective and scalable gradient boosting system. When applied to large-scale machine learning tasks, it truly excels. LightGBM implements a methodology for leaf-by-leaf tree development, histogram-based learning to facilitate accelerated computations, and distinctive feature aggregation to optimize the processing of categorical data. In order to efficiently handle extensive datasets, gradient-based one-side sampling (GOSS) is implemented. LightGBM has garnered significant acclaim in practical scenarios and machine learning competitions due to its rapidity, flexibility, and capability to process massive amounts of data in parallel as given in "figure 14". These characteristics are particularly critical in situations where quickness and precision are of the utmost importance (Ke *et al.*, 2017).

```
import lightgbm as lgb

LGBM = lgb.LGBMClassifier(verbose=-1)
LGBM.fit(Xtrain, Ytrain)

predicted_values = LGBM.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('LightGBM')
print("LightGBM's Accuracy is: ", x*100)
```

```
LightGBM's Accuracy is:  99.54545454545455
```

Figure 14. Accuracy Value given by LightGBM

4. Results and Observations

```
Decision Tree --> 0.9227272727272727
Naive Bayes --> 0.9954545454545455
SVM --> 0.09772727272727273
Logistic Regression --> 0.9704545454545455
RF --> 0.9931818181818182
XGBoost --> 0.9954545454545455
LightGBM --> 0.9954545454545455
```

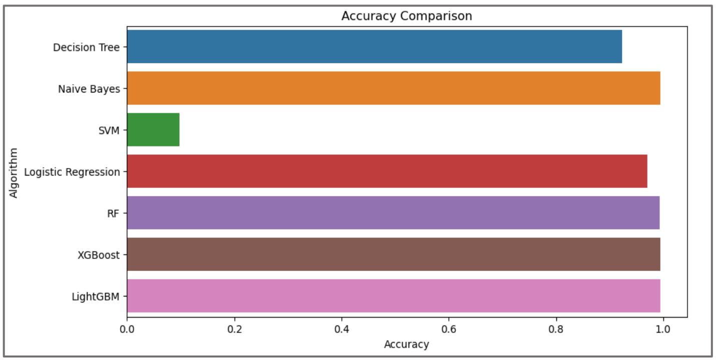

Figure 15. Accuracy Values given by Machine Learning Models

Figure 16. Comparison of the machine learning models' accuracy

The accuracy comparison between several models for machine learning that were built for the study is shown in "figure 16". Among the machine learning models we trained, XGBoost and LightGBM models resulted in the highest accuracy scores as highlighted in "figure 15". Due to the nature of these two machine learning algorithms being fast, efficient, and lightweight, their models are perfect for being implemented in real-life applications such as a web/mobile app for a crop recommendation system.

Conclusion and Future work

The main goal of the work was to develop an all-encompassing approach for crop recommendation that considers an extensive array of environmental parameters with the intention of identifying the most optimal crop to cultivate from a multitude of feasible alternatives. This was achieved by developing models utilizing the aforementioned machine learning techniques and subsequently identifying the model that generated the highest level of accuracy for our dataset.

For the initial release of the web application, the most precise model will be utilized. Another future work is to create a feature selection technique that reduces the total number of input variables that are used in the model. In the real world, it may be challenging to quantify every characteristic in our dataset which could be explored further.

Acknowledgement

This work is supported by the Department of Information Technology, Mizoram University. Authors are grateful to the University for laboratory facilities for carrying out the work.

Conflicts of interest

The authors have no conflicts of interest to declare.

5. References

1. Ajibade, S. S., & Adediran, A. (2016). An overview of big data visualization techniques in data mining. International Journal of Computer Science and Information Technology Research. 4(3), 105-113.
2. Araújo, S.O., Peres, R.S., Ramalho, J.C., Lidon, F., & Barata, J. (2023). Machine Learning Applications in Agriculture: Current Trends, Challenges, and Future Perspectives. Agronomy. 13(12), 2976.
3. Sagana, C., Sangeetha, M., Savitha, S., Devendran, K., Kavin, T., Kavinsri, K., & Mithun, P. (2023). Machine Learning-Based Crop Recommendations for Precision Farming to

Maximize Crop Yields. In Proceedings of International Conference on Computer Communication and Informatics (ICCCI), 1-5.

4. Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, 785-794. https://dl.acm.org/doi/pdf/10.1145/2939672.2939785
5. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning. 20(3), 273-297. https://doi.org/10.1007/BF00994018
6. Gosai, D., Raval, C.U., Nayak, R.J., Jayswal, H., & Patel, A. (2021). Crop Recommendation System using Machine Learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 7(3), 554-557.
7. Doshi, Z., Nadkarni, S., Agrawal, R., & Shah, N.K. (2018). AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms. In Proceedings of the 4th International Conference on Computing Communication Control and Automation (ICCUBEA), 1-6. doi: 10.1109/ICCUBEA.2018.8697349
8. Elbasi, E., Zaki, C., Topcu, A.E., Abdelbaki, W., Zreikat, A.I., Cina, E., Shdefat, A.Y., & Saker, L. (2023). Crop Prediction Model Using Machine Learning Algorithms. Applied Sciences. 13(9288), 1-20. https://doi.org/10.3390/app13169288
9. García, S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. Intelligent Systems Reference Library (vol 72, pp. 59-139). Cham, Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4
10. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big Data Preprocessing: Methods and Prospects. Big Data Analytics. 1, 1-22. https://doi.org/10.1186/s41044-016-0014-0
11. Ingle, Atharva (2020). Crop Recommendation Dataset. [Data set]. https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset
12. Israni, D., Masalia, K., Khasgiwal, T., Tolani, M., & Edinburgh, M.R. (2022). Crop-Yield Prediction and Crop Recommendation System. In Proceedings of the 7th International Conference on Innovations and Research in Technology and Engineering (ICIRTE-2022). http://dx.doi.org/10.2139/ssrn.4111856
13. Jupyter Team (2015 - 2024). Project Jupyter Documentation. Retrieved 5 February 2024. https://docs.jupyter.org/en/latest/
14. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 3149–3157.
15. Keim, D. A., & Kriegel, H. P. (1996). Visualization Techniques for Mining Large Databases: A comparison. IEEE Transactions on knowledge and data engineering. 8(6), 923-938. DOI: 10.1109/69.553159
16. Kumar, V., Vijaykumar, B., & Trilochan. (2014). Bayes and Naive Bayes Classifier. arXiv. https://doi.org/10.48550/arXiv.1404.0933
17. Jeevan, D.Y., Kumar, N., Spandana, V., Vaishnavi, V.S., Neha, K., & Devi, V.N. (2020). Supervised Machine learning Approach for Crop Yield Prediction in Agriculture Sector. In Proceedings of the 5th International Conference on Communication and Electronics Systems (ICCES), 736-741. doi: 10.1109/ICCES48766.2020.9137868
18. Lantz, B. H. (2015). Random Forest. Journal of Insurance Medicine. 47(1), 31-39. https://doi.org/10.17848/1933-2861.47.1.31
19. Lavalley, M. P. (2008). Logistic regression. Circulation. 117(18), 2395-2399. https://doi.org/10.1161/CIRCULATIONAHA.106.682658
20. Mahesh, B. (2020). Machine Learning Algorithms - A Review. International Journal of Science and Research (IJSR). 9(1), 381-386. DOI:10.21275/ART20203995

21. Mahesh, K. M., Aithal, P. S. & Sharma, K. R. S. (2023). Impact of Aatmanirbharta (Self-reliance) Agriculture and Sustainable Farming for the 21st Century to Achieve Sustainable Growth. International Journal of Applied Engineering and Management Letters. 7(2), 175-190. DOI:10.47992/IJAEML.2581.7000.0181.
22. Kumar, M. (2019). Agriculture: Status, Challenges, Policies and Strategies for India. International Journal of Engineering Research & Technology (IJERT), 8(12), 1-5.
23. Bais, P. & Bahadur, P. S. (2023). Agriculture in Indian Economy and Contribution of Science and Technology. Asian Journal of Applied Science and Technology (AJAST). 7(1), 167-176. DOI: https://doi.org/10.38177/ajast.2023.7116
24. Priyadarshini, P., & Abhilash, P.C. (2020). Policy recommendations for enabling transition towards sustainable agriculture in India. Land Use Policy. 96, 104718.

a. https://doi.org/10.1016/j.landusepol.2020.104718.

25. Rajak, R. K., Pawar, A., Pendke, M., Shinde, P., Rathod, S., & Devare, A. (2017). Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique. International Research Journal of Engineering and Technology. 4(12), 950-953.
26. Sidhu, K.S., Singh, R., Singh, S., & Singh, G. (2021). Data Science and Analytics in Agricultural Development. Environment Conservation Journal. 22, 9-19.
27. Singh, A. K. (2022). Precision Agriculture in India - Opportunities and Challenges. Indian Journal of Fertilisers. 18(4), 308-331.
28. Song, Y. Y., & Lu, Y. (2015). Decision Tree Methods: Applications for Classification and Prediction. Shanghai Archives of Psychiatry. 27(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044
29. Roy, S., Vatsa, Y., Sahoo, M., & Karan, S. (2023). Machine Learning-Based Algorithms Towards Crop Recommendation Systems. Novel Research and Development Approaches in Heterogeneous Systems and Algorithms (pp. 236-257). IGI Global. DOI: 10.4018/978-1-6684-7524-9.ch013
30. Tosi, S. (2009). Matplotlib for Python developers. Packt Publishing Ltd.
31. Meshram, V.A., Patil, K., Meshram, V.V., Hanchate, D.B., & Ramkteke, S.D. (2021). Machine Learning in Agriculture Domain: A state-of-art Survey. Artificial Intelligence in the Life Sciences. 1(5), 100010. https://doi.org/10.1016/j.ailsci.2021.100010
32. Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. Journal of Open Source Software. 6(60), 3021.