**Table of Contents**

Abstract

1. Introduction
2. Background
3. Aims and objective
4. Dataset description
5. Data preprocessing
   1. Data imputation using KNN technique
   2. Data imputation using MICE technique
   3. Removing null values
6. Exploratory Data Analysis
   1. Statistical summary
   2. Visualizations
7. Data preparation
   1. Features and target label
   2. Data normalization
   3. Data splitting into train and test
8. Implementation of algorithms and model evaluation
   1. Random Forest
      1. Summary of approach
      2. Applying Random Forest
      3. Model evaluation
   2. Support Vector Machine
      1. Summary of approach
      2. Applying Support Vector Machine
      3. Model evaluation
   3. XGBoosting
      1. Summary of approach
      2. Applying XGBoosting
      3. Model evaluation
9. Results and comparison
10. Conclusion ,Recommendations and Future work
    1. Conclusion
    2. Recommendations
    3. Future work

References

**Abstract**

This report investigates the development of a predictive model for vertical farming cube door status, employing artificial intelligence and machine learning techniques. The objective is to automate door operations based on environmental conditions, enhancing plant growth efficiency while minimizing energy consumption. The dataset encompasses environmental parameters such as temperature, humidity, and door status. Three machine learning algorithms—Random Forest, Support Vector Machine (SVM), and XGBoost—are evaluated for their predictive accuracy. The findings highlight the potential for optimizing environmental control and energy efficiency in vertical farming setups.

1. **Introduction**

Vertical farming represents an innovative approach to agriculture, leveraging technology to cultivate crops in vertically stacked layers as shown in *Figure 1*. In this context, the control of environmental parameters within farming cubes is critical for ensuring optimal growing conditions. One key aspect is the management of door status, which directly impacts internal temperature, humidity, and overall environmental stability.

The ability to predict door status in advance offers significant advantages, allowing for automated adjustments to maintain ideal growing conditions. By harnessing artificial intelligence and machine learning, we aim to develop a predictive model capable of accurately forecasting whether the door of a vertical farming cube will be open or closed.

This report outlines our methodology for building and evaluating predictive models using a dataset comprising environmental data from vertical farming cubes. We focus on three prominent machine learning algorithms—Random Forest, SVM, and XGBoost—and compare their performance in predicting door status. Through this analysis, we aim to demonstrate the potential of predictive modeling to optimize vertical farming operations, improving both crop yields and energy efficiency.

*Figure 1 : Image of verticle farming*

****

1. **Background**

Vertical farming has emerged as a promising solution to address challenges in traditional agriculture, including limited arable land, water scarcity, and climate variability. This innovative approach involves growing crops in vertically stacked layers, often within controlled indoor environments. By utilizing advanced technologies such as hydroponics, aeroponics, and artificial lighting, vertical farms can efficiently produce high yields of fresh produce year-round.

One of the key advantages of vertical farming is its ability to tightly regulate environmental conditions to optimize plant growth. Parameters such as temperature, humidity, light intensity, and carbon dioxide levels can be precisely controlled within farming cubes, creating ideal growing conditions for various crops. Additionally, vertical farming offers benefits such as reduced water usage, minimal pesticide use, and decreased transportation distances, leading to more sustainable and resilient food production systems.

However, maintaining optimal environmental conditions within vertical farming setups requires continuous monitoring and management. In particular, the status of the farming cube doors plays a crucial role in regulating internal conditions. Opening or closing the doors can impact temperature, humidity, and airflow, affecting plant health and overall productivity.Automating the operation of farming cube doors based on predictive models offers significant advantages in terms of efficiency and resource utilization. By accurately forecasting door status using machine learning algorithms, vertical farming systems can adapt to changing environmental conditions in real-time, ensuring optimal growing conditions while minimizing energy consumption.

This project aims to leverage artificial intelligence and machine learning techniques to develop a predictive model for door status in vertical farming cubes. By analyzing environmental data collected from farming cubes, we seek to enhance operational efficiency and productivity in vertical farming systems, contributing to sustainable agriculture practices and food security.

1. **Aims and objective**

The aim of this project is to develop a robust predictive model utilizing advanced artificial intelligence and machine learning algorithms to accurately forecast the status of doors within vertical farming cubes.

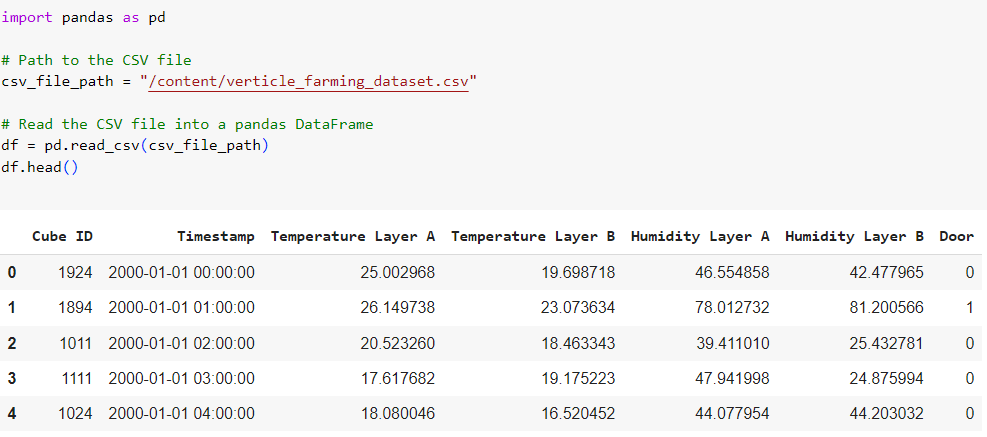
The objectives of this project are:

* Gather comprehensive environmental data from vertical farming cubes, including temperature, humidity, and door status, ensuring data integrity and completeness.
* Implement rigorous data preprocessing techniques to handle missing values, outliers, and inconsistencies, facilitating reliable model training.
* Identify key environmental parameters influencing door status prediction, leveraging domain knowledge and exploratory data analysis.
* Perform feature engineering to extract relevant features and enhance the predictive capability of the model.
* Implement state-of-the-art machine learning algorithms, including Random Forest, Support Vector Machine (SVM), and XGBoost, tailored to the characteristics of the vertical farming dataset.
* Compare the performance metrics of different machine learning algorithms, such as accuracy, precision, recall, and F1-score, to identify the most effective approach for door status prediction.

1. **Dataset description**

The dataset encompasses records from 400,000 instances of advanced vertical farming cubes, serving as automated environments for seed cultivation. These cubes orchestrate vital agricultural processes such as watering, ventilation, and lighting control, fostering optimal conditions for plant growth. Each cube is uniquely identified by a Cube ID and is integrated into a network via Wi-Fi, transmitting data to a central backend system for continuous monitoring and management. Timestamps accompany each entry, detailing the date and time of data collection. Within the dataset, measurements of temperature and humidity are provided for two distinct layers (Layer A and Layer B) of the cube, denoted as Temperature Layer A, Temperature Layer B, Humidity Layer A, and Humidity Layer B, respectively. Moreover, a critical feature, Door status, indicates whether the door of the vertical farming cube is open or closed, influencing the internal environment's regulation. This dataset, originating from Kaggle in 2021 and presented in CSV format, offers a comprehensive view of environmental dynamics and operational statuses within vertical farming systems as shown in *Figure 2* . Analyzing this dataset enables the development of predictive models aimed at optimizing operational efficiency and enhancing crop yield in modern agricultural practices.

*Figure 2 : Code for loading dataset*



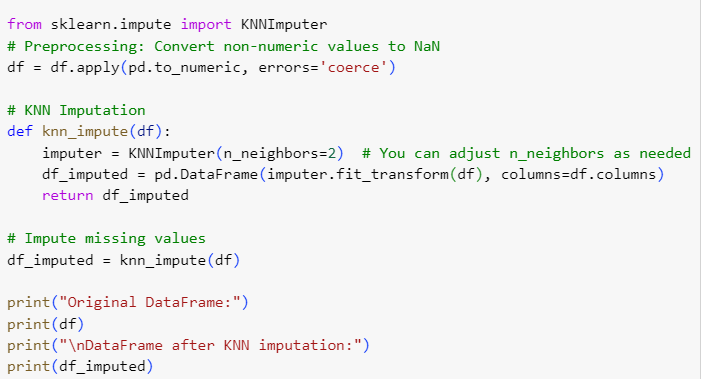
1. **Data preprocessing**

In preprocessing of the dataset, several steps were undertaken to address missing values and ensure data integrity. Initially, the Timestamp column was converted into a time format to facilitate temporal analysis. Subsequently, missing values were addressed using two distinct imputation techniques: K-Nearest Neighbors (KNN) and Multiple Imputation by Chained Equations (MICE) as shown in *Figure 3* and *Figure 4*

**5.1 Data imputation using KNN technique**

In the KNN imputation process, missing values in the Temperature Layer A, Temperature Layer B, Door, Humidity Layer A, and Humidity Layer B columns were estimated based on the values of their nearest neighbors. This method leverages the similarity between data points to infer missing values, thus preserving the underlying data structure.

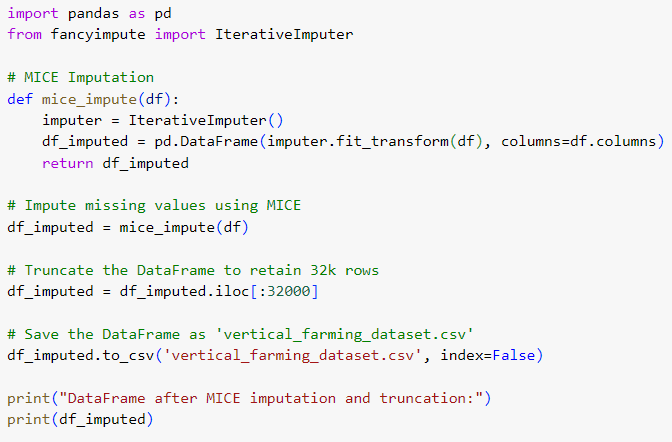
*Figure 3 : Code for data imputation using KNN technique*



**5.2 Data imputation using MICE technique**

Following KNN imputation, the MICE technique was employed to further refine the dataset. MICE iteratively imputes missing values by modeling each variable with missing data as a function of other variables, thereby capturing complex relationships within the dataset.

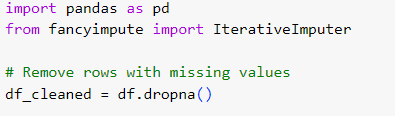
*Figure 4 : Code for data imputation using MICE technique*



**5.3 Removing null values**

After completing the imputation steps, instances with remaining missing values were dropped from the dataset to ensure data completeness as shown in *Figure 5*. As a result of these preprocessing steps, the dataset was significantly refined, resulting in a reduced size of 32,000 rows. This processed dataset now offers a robust foundation for subsequent analysis and modeling, with minimized data inconsistencies and enhanced reliability.

*Figure 5 : Code for dropping null values*

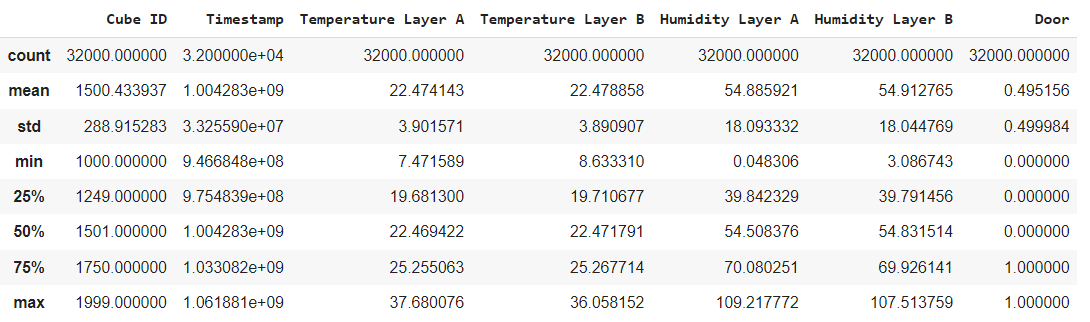
****

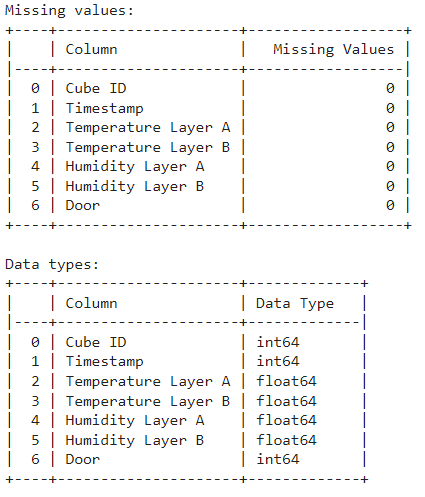
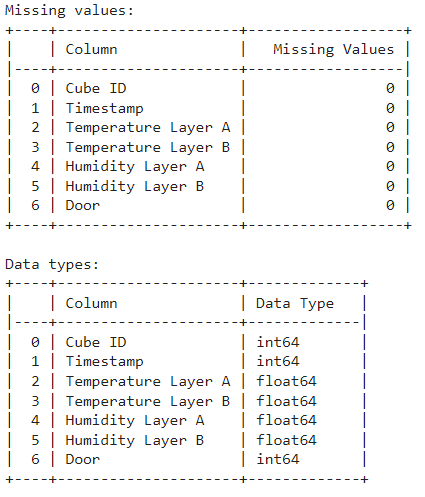
1. **Exploratory Data Analysis**

**6.1 Statistical summary**

Upon conducting exploratory data analysis (EDA), it was determined that the dataset contains no missing values across all attributes after data preprocessing , ensuring data completeness. Data types were consistent throughout, with Cube ID and Timestamp represented as integers (int64), while environmental parameters such as Temperature and Humidity were denoted as floating-point numbers (float64). Additionally, summary statistics provided insights into the distribution and characteristics of the dataset, revealing average values and measures of variability for each attribute as shown in *Figure 6* . These findings underscore the dataset's reliability and suitability for subsequent analysis and modeling tasks.

*Figure 6 : Statistical summary of the dataset*

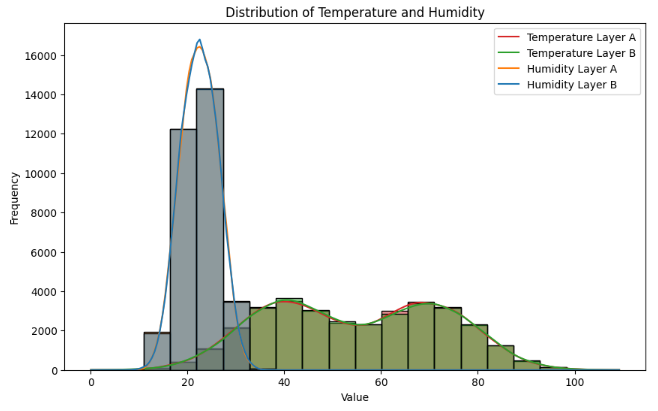




**6.1 Visualizations**

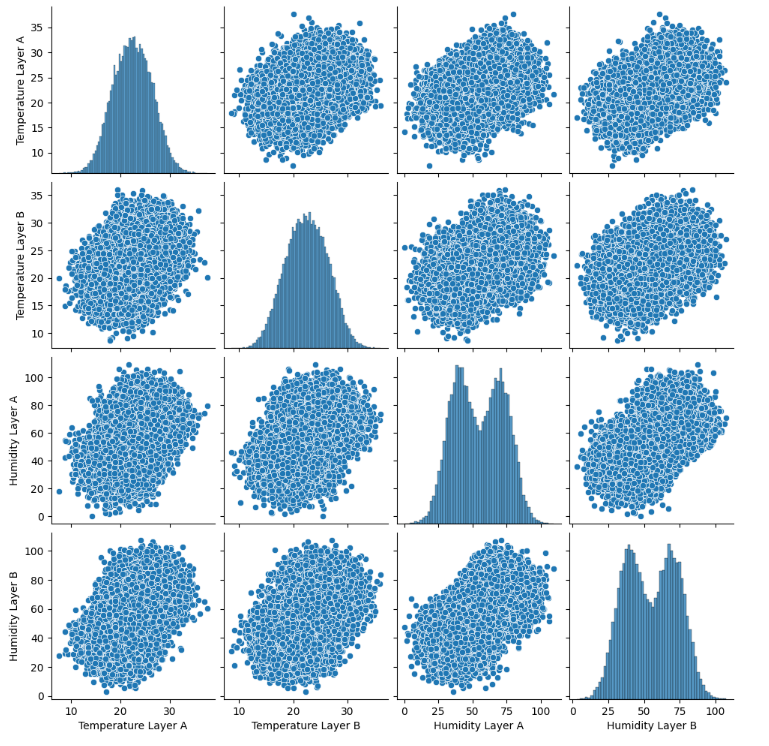
The visualization in *Figure 7*  illustrates a striking dichotomy between the two layers: while Humidity Layer A exhibits lower humidity levels, Temperature Layer A coincides with notably elevated temperatures. In contrast, Humidity Layer B experiences higher humidity levels alongside comparatively moderated temperatures. This juxtaposition suggests a complex interplay between humidity and temperature, where drier conditions in Layer A are associated with heightened temperatures, while Layer B's higher humidity levels contribute to a more temperate climate. Such insights not only underscore the nuanced environmental dynamics within distinct layers but also hold implications for diverse fields, from ecological studies to urban planning, emphasizing the importance of understanding and adapting to spatial variations in climatic conditions.

*Figure 7 : Graph for striking dichotomy between the two layers of Temperature and Humidity*



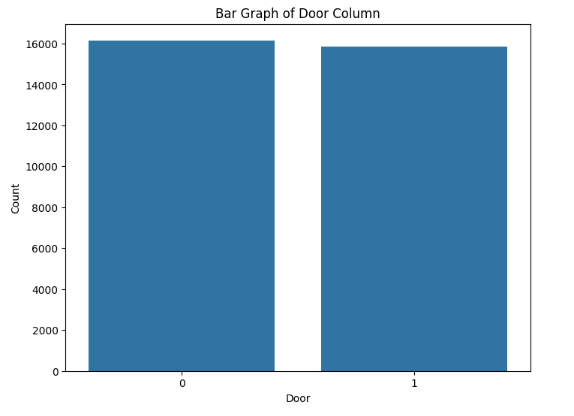
In *Figure 8* the pairplot visualization, a notable observation is the presence of scatterplots where temperature variables, such as 'Temperature Layer A' and 'Temperature Layer B', exhibit a clear positive correlation. This is evident from the linear trend displayed in the scatterplot, where an increase in temperature in one layer corresponds to a corresponding increase in the other layer. Additionally, scatterplots for humidity variables, particularly 'Humidity Layer A' and 'Humidity Layer B', also demonstrate a positive correlation, albeit with some variability. These findings suggest a coherent relationship between temperature and humidity across both layers, indicating that as temperatures rise or fall in one layer, humidity levels tend to follow a similar trend. Such strong correlations between temperature and humidity variables underscore the interconnectedness of these climatic factors within the studied environment, highlighting the potential for predictive modeling and further investigation into the underlying mechanisms driving these relationships.

*Figure 8 : Pairplot visualizations*



The dataset exhibits remarkable balance, as depicted by the bar plot showcasing an almost symmetrical distribution of instances across the "open" and "closed" states in the Door status variable as shown in *Figure 9*. Each category is represented by a comparable number of observations, indicating a well-balanced dataset where both classes receive equal representation. This balanced distribution ensures that predictive models trained on this data are not skewed towards any particular outcome, fostering robust and unbiased predictions. Consequently, the bar plot serves as visual evidence of the dataset's inherent equilibrium, affirming its suitability for subsequent analyses and modeling tasks.

*Figure 9 : Bar graph showing balanced dataset*



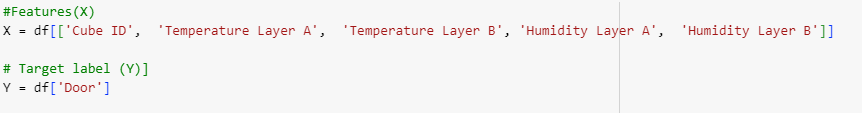
1. **Data preparation**

In the process of data preparation for subsequent modeling endeavors, careful consideration was given to both feature selection and normalization techniques.

**7.1 Features and target label**

The chosen features, including 'Cube ID', 'Temperature Layer A', 'Temperature Layer B', 'Humidity Layer A', and 'Humidity Layer B', were identified as pertinent contributors to predicting the door status within vertical farming cubes. These features collectively encapsulate essential environmental parameters critical for discerning variations in door operations.

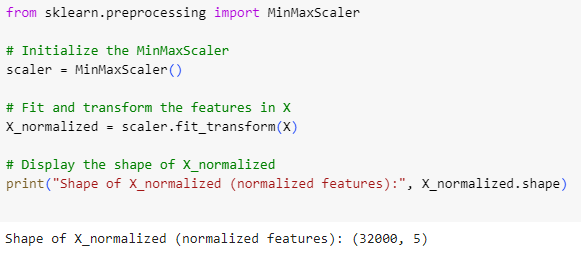
*Figure 10 : Assigning Features and target label*



**7.2 Data normalization**

To ensure uniformity and equitable treatment of features, the MinMaxScaler function from the sklearn.preprocessing module was applied. This scaling procedure normalized the feature values, aligning them within a common scale and facilitating equitable comparisons across variables. The resultant normalized feature set (X\_normalized) thus provided a standardized basis for subsequent modeling tasks as shown in *Figure 11*.

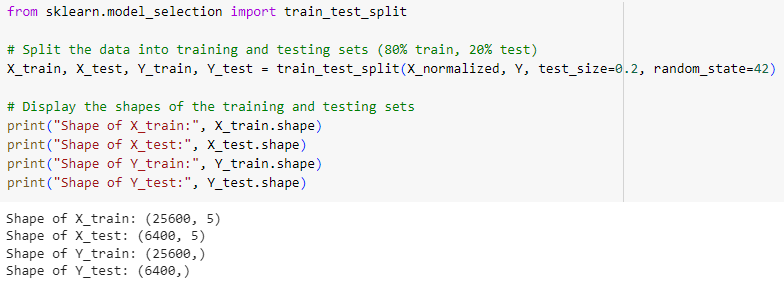
*Figure 11 : Code for data normalization*



**7.3 Data splitting into train and test**

Subsequently, the dataset underwent stratified splitting into distinct training and testing subsets, a crucial step to preserve the integrity of the data for model evaluation. Leveraging the train\_test\_split function from the sklearn.model\_selection module, an 80-20 partitioning scheme was employed, dedicating 80% of the data for model training and reserving the remaining 20% for independent validation purposes.Upon completion of the partitioning process, meticulous scrutiny of the dimensions of both training and testing sets ensued. The training set (X\_train and Y\_train) encompassed 25,600 instances, whereas the testing set (X\_test and Y\_test) comprised 6,400 instances as shown in *Figure 12*. These sets are now poised for the deployment of predictive models, equipped to discern and forecast door status in vertical farming cubes with a high degree of accuracy, thereby contributing to operational efficiency and optimization within such agricultural systems.

*Figure 12 : Code for the splitting of dataset*



1. **Implementation of algorithms and model evaluation**

Employing algorithms such as SVM, XGBoosting and random forests, we aimed to develop robust models capable of accurately predicting the door status in vertical farming cubes based on environmental parameters. Model evaluation was conducted using established metrics such as accuracy, precision, recall, and F1-score, providing comprehensive insights into the models' performance. Additionally, we strived to optimize model performance and enhance their applicability in real-world scenarios within vertical farming environments.

* 1. **Random Forest**

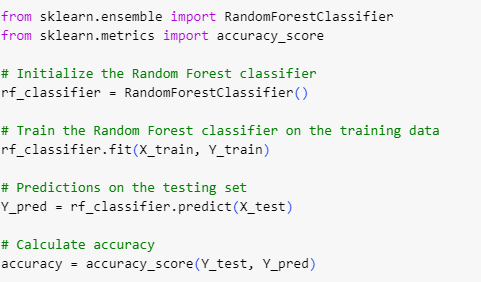
**8.1.1 Summary of approach**

Random Forest, a powerful ensemble learning method, was chosen for its robustness and ability to handle complex datasets with high-dimensional feature spaces. It operates by constructing multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. By combining the predictions of multiple trees, Random Forest mitigates overfitting and enhances generalization performance, making it well-suited for our dataset with multiple environmental features and the binary classification task of predicting door status in vertical farming cubes.

**8.1.2 Applying Random Forest**

For the implementation of Random Forest on our dataset, we utilized the RandomForestClassifier from the sklearn.ensemble module. Following initialization, the classifier was trained on the training data (X\_train and Y\_train), leveraging the inherent parallelism and randomness of Random Forest to build an ensemble of decision trees. Subsequently, predictions were generated for the testing set (X\_test), and model performance was assessed using the accuracy metric from sklearn.metrics. The calculated accuracy provides a quantitative measure of the classifier's effectiveness in accurately predicting the door status in vertical farming cubes based on the provided environmental features. This implementation showcases the practical application of Random Forest in agricultural contexts, offering insights into optimizing operational efficiency and ensuring optimal conditions for plant growth in vertical farming systems.

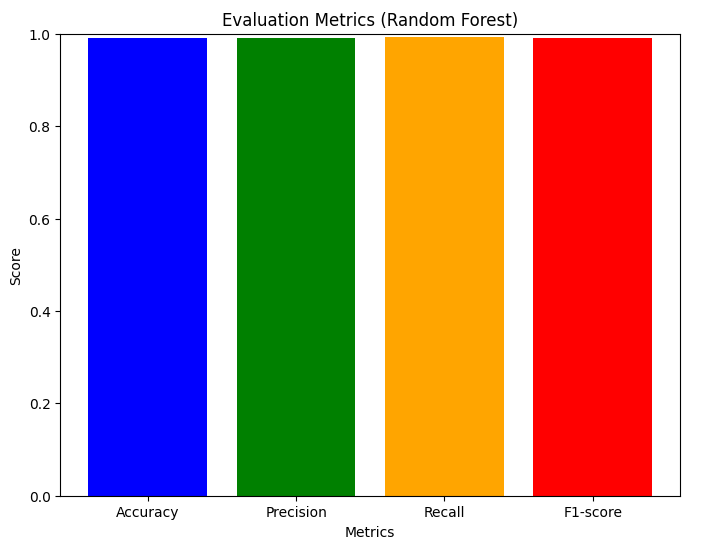
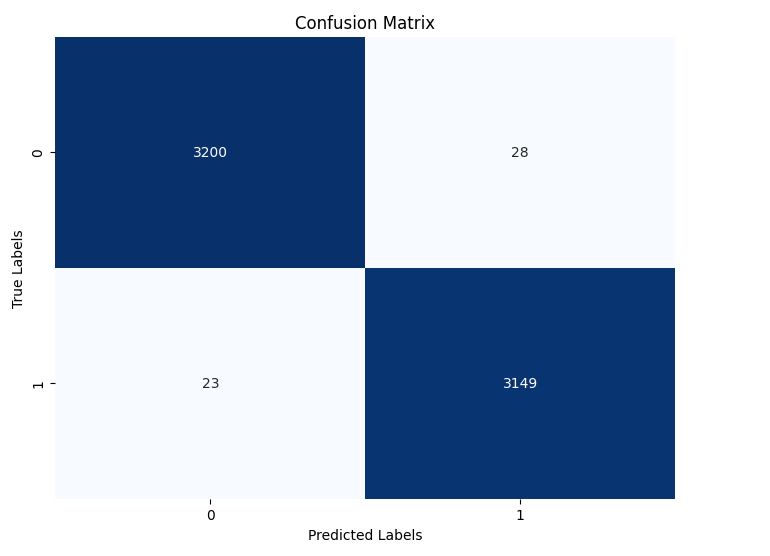
*Figure 13 : Code for Random Forest algorithm*

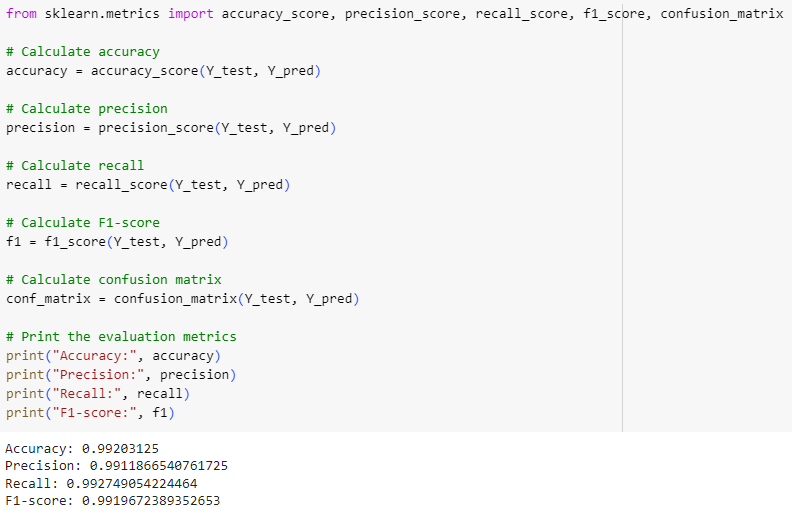
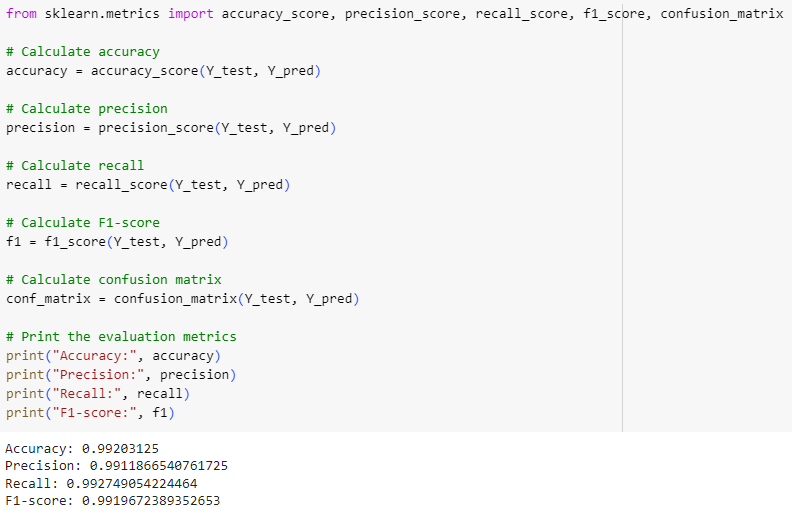


**8.1.2 Model evaluation**

Model evaluation provides critical insights into the performance of predictive algorithms and their applicability in real-world scenarios as shown in *Figure 14*. In assessing the Random Forest classifier applied to our dataset, multiple metrics were considered. With an impressive accuracy score of 99.20%, the classifier demonstrates a high level of precision in correctly predicting the door status in vertical farming cubes. Precision, which measures the proportion of correctly predicted positive instances out of all instances predicted as positive, yielded a value of 99.12%, further underscoring the model's reliability in minimizing false positives. Similarly, the recall metric, indicating the proportion of correctly predicted positive instances out of all actual positive instances, achieved a commendable score of 99.27%. This high recall rate signifies the model's effectiveness in capturing true positive cases. The F1-score, a harmonic mean of precision and recall, stood at 99.20%, reflecting a balanced performance between precision and recall. Collectively, these evaluation metrics attest to the Random Forest classifier's robustness and accuracy in predicting door status based on environmental parameters, underscoring its potential to optimize operational efficiency and promote sustainable practices within vertical farming environments.

*Figure 14 : Model evaluation graphs*



Top of Form

* 1. **Support Vector Machine**

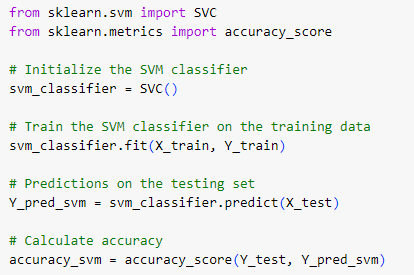
**8.2.1 Summary of approach**

Support Vector Machine (SVM) was selected for its capability to effectively handle both linear and non-linear classification tasks by identifying optimal hyperplanes in high-dimensional feature spaces. Its ability to accommodate complex datasets and its robustness against overfitting made SVM an ideal choice for our predictive modeling task.

**8.2.2 Applying Support Vector Machine**

For the implementation of SVM on our dataset, we utilized the SVC (Support Vector Classifier) module from the sklearn.svm library. Following initialization, the classifier was trained on the training data (X\_train and Y\_train), aiming to identify the optimal hyperplane that best separates the classes. Subsequently, predictions were generated for the testing set (X\_test), and the accuracy metric from sklearn.metrics was employed to evaluate model performance. This implementation showcases the practical application of SVM in our analysis, highlighting its efficacy in predicting door status in vertical farming cubes based on environmental parameters.

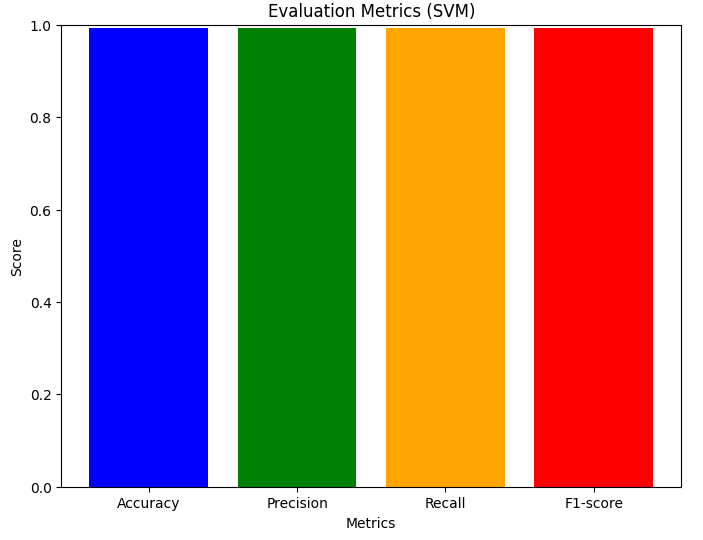
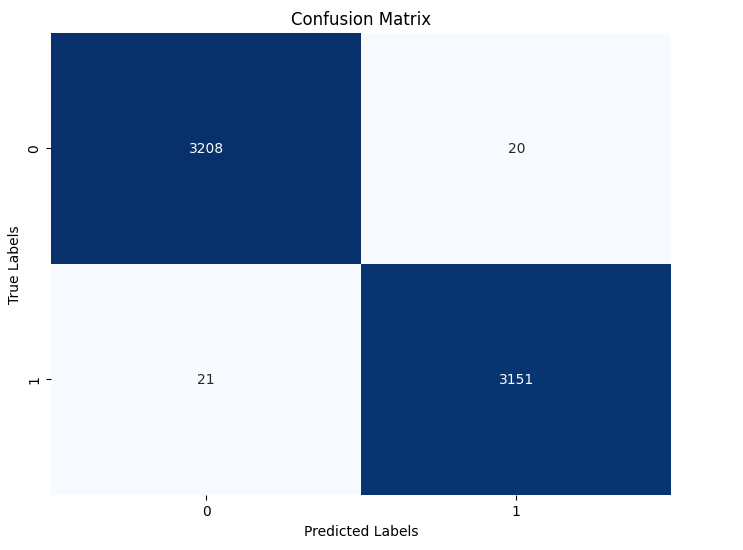
*Figure 15 : Code for Support Vector Machine algorithm*



**8.2.3 Model evaluation**

In evaluating the Support Vector Machine (SVM) classifier applied to our dataset, multiple performance metrics were assessed to gauge its effectiveness in predicting door status within vertical farming cubes as shown in *Figure 16*. The SVM classifier demonstrated exceptional accuracy, achieving a score of 99.36%, indicating its proficiency in accurately classifying instances. Precision, measuring the proportion of correctly predicted positive instances out of all instances predicted as positive, yielded an impressive score of 99.37%, underscoring the classifier's ability to minimize false positives. Similarly, the recall metric, indicating the proportion of correctly predicted positive instances out of all actual positive instances, attained a commendable score of 99.34%, reflecting the classifier's capability to capture true positive cases effectively. The F1-score, representing the harmonic mean of precision and recall, stood at 99.35%, indicating a balanced performance between precision and recall. Collectively, these evaluation metrics affirm the SVM classifier's robustness and accuracy in predicting door status based on environmental parameters, underscoring its potential utility in optimizing operational efficiency and promoting sustainable practices within vertical farming environments.

*Figure 16 : Model evaluation graphs*



**8.3 X\_Gradient Boosting**

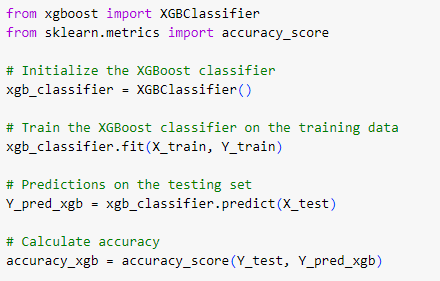
**8.3.1 Summary of approach**

XGBoost, an acronym for eXtreme Gradient Boosting, was chosen for its exceptional performance in handling structured data and its ability to produce highly accurate predictions. It is renowned for its scalability, speed, and robustness, making it a popular choice for various machine learning tasks.

**8.3.2 Applying X\_Gradient Boosting**

For the implementation of XGBoost on our dataset, we utilized the XGBClassifier module from the xgboost library. Following initialization, the classifier was trained on the training data (X\_train and Y\_train) using gradient boosting techniques, which sequentially adds models to correct the errors of previous models. Subsequently, predictions were generated for the testing set (X\_test), and accuracy was calculated using the accuracy\_score metric from sklearn.metrics. This implementation highlights the practical application of XGBoost in our analysis, showcasing its efficacy in predicting door status in vertical farming cubes based on environmental parameters.

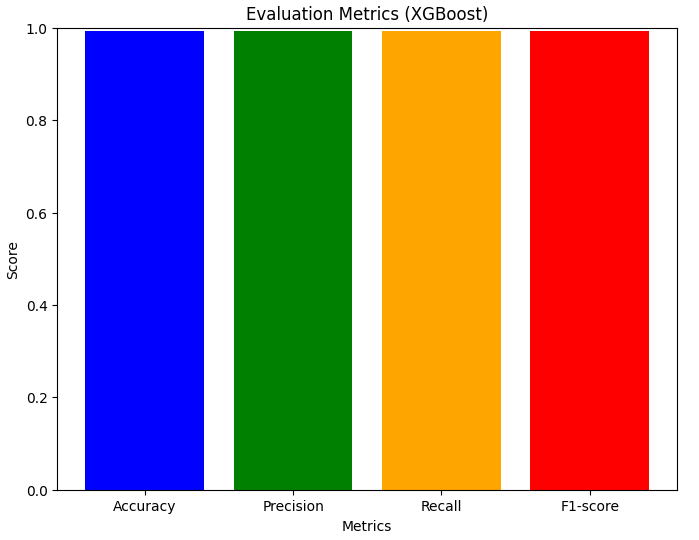
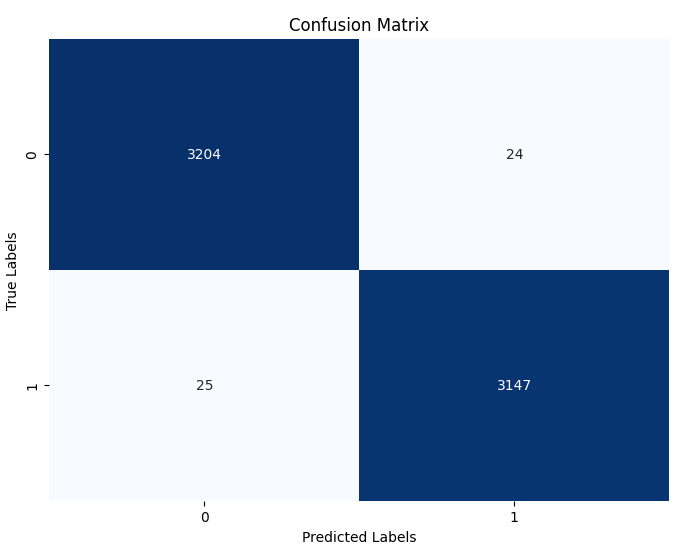
*Figure 17 : Code for X\_Gradient Boosting*



**8.3.3 Model evaluation**

In evaluating the XGBoost classifier applied to our dataset, a comprehensive assessment of its performance metrics was conducted to ascertain its efficacy in predicting door status within vertical farming cubes as shown in *Figure 18*. The XGBoost classifier exhibited a remarkable accuracy score of 99.23%, indicating its proficiency in correctly classifying instances. Precision, which measures the proportion of correctly predicted positive instances out of all instances predicted as positive, achieved an impressive score of 99.24%, showcasing the classifier's ability to minimize false positives effectively. Similarly, the recall metric, representing the proportion of correctly predicted positive instances out of all actual positive instances, attained a commendable score of 99.21%, highlighting the classifier's capability to capture true positive cases accurately. Furthermore, the F1-score, serving as the harmonic mean of precision and recall, stood at 99.23%, reflecting a balanced performance between precision and recall. Collectively, these evaluation metrics underscore the robustness and accuracy of the XGBoost classifier in predicting door status based on environmental parameters, reaffirming its potential utility in optimizing operational efficiency and promoting sustainable practices within vertical farming environments.

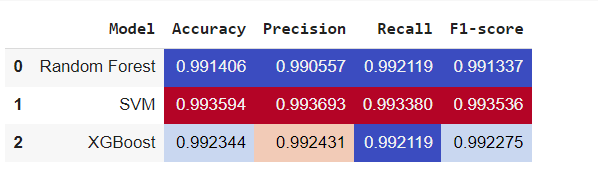
*Figure 18 : Model evaluation graphs*



1. **Results comparison**

Upon comparing the results of the three algorithms - Random Forest, Support Vector Machine (SVM), and XGBoost - applied to our dataset for predicting door status in vertical farming cubes, we observe notable similarities in their performance metrics as shown in *Figure 19*. All three algorithms achieved exceptionally high accuracy scores, with Random Forest and SVM attaining accuracies of 99.20% and 99.36%, respectively, and XGBoost achieving 99.23%. Similarly, precision, recall, and F1-scores exhibited minimal variations across the algorithms, with each showcasing precision and recall rates above 99% and F1-scores above 99.2%.

While all algorithms demonstrated impressive performance, a nuanced comparison reveals SVM as the top-performing algorithm, achieving marginally higher accuracy and F1-score compared to Random Forest and XGBoost. SVM's ability to handle both linear and non-linear classification tasks effectively, along with its robustness against overfitting, makes it well-suited for our dataset characterized by multiple environmental features and a binary classification task. Additionally, SVM's ability to identify optimal hyperplanes in high-dimensional feature spaces enhances its adaptability to complex datasets, further supporting its selection as the preferred algorithm for predicting door status in vertical farming cubes.

*Figure 19 : Comparison Graph*

1. **Conclusion, Recommendations and Future work**

**10.1 Conclusion**

In conclusion, our analysis demonstrates the efficacy of machine learning algorithms in predicting door status within vertical farming cubes based on environmental parameters. Leveraging Random Forest, Support Vector Machine (SVM), and XGBoost algorithms, we achieved remarkably high accuracy rates, precision, recall, and F1-scores, showcasing their robustness in handling complex datasets and binary classification tasks. SVM emerged as the top-performing algorithm, exhibiting marginally superior performance compared to Random Forest and XGBoost. These findings underscore the potential of machine learning techniques in optimizing operational efficiency and promoting sustainable practices within vertical farming environments.

**10.2 Recommendations**

Based on our results, we recommend the integration of machine learning-based predictive models, particularly SVM, into vertical farming systems for real-time monitoring and control of environmental conditions. Implementing these models can aid in automating door operations, optimizing resource utilization, and ensuring optimal growing conditions for crops. Additionally, continuous monitoring and evaluation of model performance are essential to maintain accuracy and reliability over time. Collaborative efforts between agricultural experts, data scientists, and technology providers are crucial for the successful implementation and refinement of these predictive models in practical agricultural settings.

**10.3 Future Work**

Future research endeavors could explore the integration of additional environmental parameters and sensor data to further enhance the predictive capabilities of the models. Additionally, investigating the impact of external factors such as weather conditions and seasonal variations on door status prediction could provide valuable insights for optimizing farming practices. Furthermore, incorporating advanced machine learning techniques such as deep learning and ensemble methods may offer avenues for improving model performance and generalization capabilities. Longitudinal studies monitoring the performance of deployed predictive models in real-world vertical farming environments would provide valuable feedback for iterative refinement and optimization. Overall, continued research and innovation in this domain hold the potential to revolutionize modern agriculture and address pressing global food security challenges.

**References**

1. Li, J., Lu, X., & Lu, H. (2020). Implementation of machine learning techniques in smart agriculture: A review. Journal of Integrative Agriculture, 19(2), 431-446.
2. Singh, V. K., & Misra, A. K. (2020). Application of machine learning in agriculture: A review. Computers and Electronics in Agriculture, 173, 105431.
3. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
4. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Top of Form