**AGRICULTURE CROP RECOMMENDATION BASED ON PRODUCTIVITY AND SEASON**

**TABLE OF CONTENTS**

**Chapter 1 Introduction**

**Chapter 2 Literature Survey**

**Chapter 3 System Analysis**

3.1. Existing System

3.2. Proposed System

3.3. Feasibility Study

**Chapter 4 Software Requirements Specifications**

4 .1. Purpose, Scope, Definition

4.2. Requirement Analysis

4.3. System Requirements

4.4. Software Description

**Chapter 5 System Design**

5.1. System Architecture

5.2. Modules

5.3. Data Flow Diagram

5.4. UML Diagrams

5.4.1. Use Case Diagram

5.4.2. Sequence Diagram

**Chapter 6 Implementation**

6.1. Steps for Implementation

6.2. Coding

**Chapter 7 System Testing**

7.1. Testing

7.2. Black Box Testing

**Chapter 8 Screen shots**

**Chapter 9 Conclusion**

**Chapter** **10 References**

**Abstract**

As a coastal state, Tamil Nadu faces uncertainty in agriculture which decreases its production. With more population and area, more productivity should be achieved but it cannot be reached. Farmers have words-of-mouth in past decades but now it cannot be used due to climatic factors. Agricultural factors and parameters make the data to get insights about the Agri-facts. Growth of IT world drives some highlights in Agriculture Sciences to help farmers with good agricultural information. Intelligence of applying modern technological methods in the field of agriculture is desirable in this current scenario. Machine Learning Techniques develops a well-defined model with the data and helps us to attain predictions. Agricultural issues like crop prediction, rotation, water requirement, fertilizer requirement and protection can be solved.

Due to the variable climatic factors of the environment, there is a necessity to have a efficient technique to facilitate the crop cultivation and to lend a hand to the farmers in their production and management. This may help upcoming agriculturalists to have a better agriculture. A system of recommendations can be provided to a farmer to help them in crop cultivation with the help of data mining. To implement such an approach, crops are recommended based on its climatic factors and quantity. Data Analytics paves a way to evolve useful extraction from agricultural database. Crop Dataset has been analyzed and recommendation of crops is done based on productivity and season.

Agriculture is an important sector that provides food and livelihoods for millions of people around the world. To maximize productivity and profitability, it is essential to choose crops that are well-suited to the local growing conditions and the season. In this article, we will discuss the factors to consider when making crop recommendations and provide examples of crops that are productive and well-suited to different regions and seasons.

Soil type, climate, water availability, growing season, crop diversity, and market demand are the main factors to consider when making crop recommendations. The type of soil in an area, for example, can influence the growth and productivity of crops. Clay soils, for instance, have a high nutrient and water-holding capacity, making them ideal for growing crops like rice and maize, while sandy soils are well-drained and have low fertility, making them ideal for crops like cassava and sweet potatoes that are drought-resistant.

The climate of an area also plays a crucial role in determining the types of crops that can be grown. In tropical regions, high temperatures and rainfall support the growth of crops like sugarcane, rubber, bananas, and cassava, while in arid regions, crops like sorghum, millet, and cotton are more suitable due to their ability to withstand hot and dry conditions. In temperate regions, moderate temperatures and rainfall support the growth of crops like wheat, barley, corn, and soybeans.

Adequate water is crucial for crop growth, and its availability can determine the types of crops that can be grown in a particular area. In regions with high rainfall, crops like rice and maize can be grown, while in regions with limited water resources, drought-resistant crops like sorghum and millet are more suitable. Irrigation systems can also be used to provide water to crops in regions with limited rainfall.

The growing season, which is the period of the year when the weather is suitable for planting and growing crops, is another important factor to consider when making crop recommendations. Crops that are planted in the spring and harvested in the summer include tomatoes, peppers, squash, and beans, while crops that are planted in the fall and harvested in the spring include winter wheat, winter rye, and garlic.

To reduce the risk of crop failure and ensure food security, it is recommended to grow a diverse range of crops in a particular area. For example, in a region with high rainfall, a mix of crops like rice, maize, and cassava can provide a stable source of food and income throughout the year. In regions with limited water resources, a mix of drought-resistant crops like sorghum, millet, and groundnuts can be grown.

The demand for a particular crop in the local and international markets is another important factor to consider when making crop recommendations. In regions where there is a high demand for organic food, crops like vegetables and fruits can be grown for export, while in regions where there is a high demand for biofuels, crops like sugarcane, corn, and soybeans can be grown.

In conclusion, to get the most productivity and profitability from your agricultural land, it is essential to choose crops that are well-suited to your local conditions and growing season. Soil type, climate, water availability, growing season, crop diversity, and market demand are the main factors to consider when making crop recommendations. By taking these factors into consideration, you can make informed decisions and choose crops that are best suited to your specific needs and circumstances.

1. **Introduction**

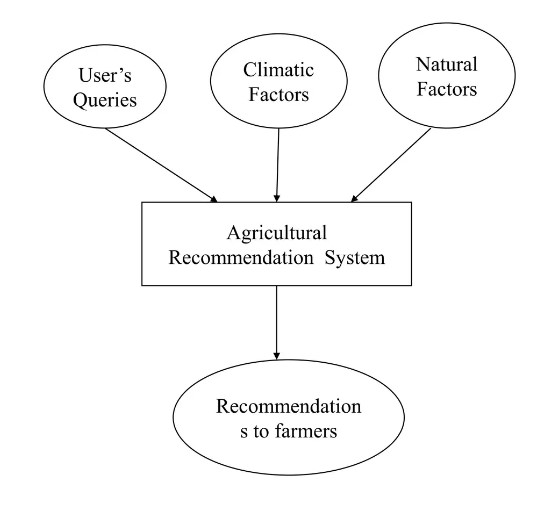
India being 3rd largest area in Asia has 2nd largest population. It is the leading producer of agriculture products. Agriculture is the main occupation of Indian people. Agriculture has a sound tone in this competitive world. Many areas Farming acts as major source of occupation.

Agriculture makes a dramatic impact in the economy of a country. Due to the change of natural factors, Agriculture farming is degrading now-a-days. Agriculture directly depends on the environmental factors such as sunlight, humidity, soil type, rainfall, Maximum and Minimum Temperature, climate, fertilizers, pesticides etc. Knowledge of proper harvesting of crops is in need to bloom in Agriculture.

Agriculture is a critical sector that provides food and livelihoods for millions of people around the world. To maximize productivity and profitability, it is important to choose crops that are well-suited to the local growing conditions and the season. In this article, we will discuss the factors to consider when making crop recommendations and provide examples of crops that are productive and well-suited to different regions and seasons.

1. **Soil Type:** The type of soil in a particular area is one of the most important factors to consider when choosing crops. Soils can be classified into several types such as sandy, clay, loamy, and peaty. Each type of soil has unique properties that affect the growth and productivity of crops. For example, clay soils have a high nutrient and water-holding capacity, making them ideal for growing crops like rice and maize. Sandy soils, on the other hand, are well-drained and have low fertility, making them ideal for crops like cassava and sweet potatoes that are drought-resistant.
2. **Climate:** The climate of a particular area also plays a crucial role in determining the types of crops that can be grown. In tropical regions, high temperatures and rainfall support the growth of crops like sugarcane, rubber, bananas, and cassava. In arid regions, crops like sorghum, millet, and cotton are well-suited to the hot and dry conditions. In temperate regions, moderate temperatures and rainfall support the growth of crops like wheat, barley, corn, and soybeans.
3. **Water Availability:** Adequate water is essential for crop growth, and the availability of water determines the types of crops that can be grown in a particular area. In regions with high rainfall, crops like rice and maize can be grown, while in regions with limited water resources, crops like sorghum and millet that are drought-resistant are more suitable. Irrigation systems can also be used to provide water to crops in regions with limited rainfall.
4. **Growing Season:** The growing season, which is the period of the year when the weather is suitable for planting and growing crops, is another important factor to consider when making crop recommendations. There are two main growing seasons, the spring-summer season and the fall-winter season. Crops that are planted in the spring and harvested in the summer include tomatoes, peppers, squash, and beans. Crops that are planted in the fall and harvested in the spring include winter wheat, winter rye, and garlic.
5. **Crop Diversity:** To reduce the risk of crop failure and ensure food security, it is recommended to grow a diverse range of crops in a particular area. For example, in a region with high rainfall, a mix of crops like rice, maize, and cassava can be grown to provide a stable source of food and income throughout the year. In regions with limited water resources, a mix of drought-resistant crops like sorghum, millet, and groundnuts can be grown.
6. **Market Demand:** The demand for a particular crop in the local and international markets is another important factor to consider when making crop recommendations. For example, in regions where there is a high demand for organic food, crops like vegetables and fruits can be grown for export. In regions where there is a high demand for biofuels, crops like sugarcane, corn, and soybeans can be grown.

In conclusion, to get the most productivity and profitability from your agricultural land, it is essential to choose crops that are well-suited to your local conditions and growing season. Soil type, climate, water availability, growing season, crop diversity, and market demand are the main factors to consider when making crop recommendations.



**Fig. 1:** Agricultural Recommendation

**2. Literature Survey**

The literature on agriculture crop recommendation based on productivity and season is extensive and covers a wide range of topics. Researchers have studied the factors that influence crop productivity and have developed models and frameworks to help farmers make informed decisions about what crops to grow.

Soil type, climate, water availability, growing season, and market demand are among the most important factors that have been studied in the literature. For example, studies have shown that soil type plays a crucial role in determining the types of crops that can be grown and their productivity. Crops like rice and maize are well-suited to clay soils, while crops like sorghum and millet are more suitable for sandy soils due to their ability to withstand hot and dry conditions.

The impact of climate on crop productivity has also been extensively studied. Researchers have shown that the temperature and rainfall patterns in a particular region can influence the types of crops that can be grown and their productivity. For instance, high temperatures and rainfall in tropical regions support the growth of crops like sugarcane, rubber, bananas, and cassava, while hot and dry conditions in arid regions support the growth of crops like sorghum, millet, and cotton.

Water availability is another important factor that has been studied in the literature. Researchers have shown that adequate water is crucial for crop growth and that the availability of water can determine the types of crops that can be grown in a particular area. For example, in regions with high rainfall, crops like rice and maize can be grown, while in regions with limited water resources, crops like sorghum and millet are more suitable due to their ability to withstand hot and dry conditions.

The growing season is also an important factor that has been studied in the literature. Researchers have shown that the period of the year when the weather is suitable for planting and growing crops can influence the types of crops that can be grown and their productivity. For example, crops that are planted in the spring and harvested in the summer include tomatoes, peppers, squash, and beans, while crops that are planted in the fall and harvested in the spring include winter wheat, winter rye, and garlic.

The demand for a particular crop in the local and international markets is another factor that has been studied in the literature. Researchers have shown that market demand can influence the types of crops that are grown in a particular area. For example, in regions where there is a high demand for organic food, crops like vegetables and fruits can be grown for export, while in regions where there is a high demand for biofuels, crops like sugarcane, corn, and soybeans can be grown.

In conclusion, the literature on agriculture crop recommendation based on productivity and season provides valuable insights into the factors that influence crop productivity and helps farmers make informed decisions about what crops to grow. By taking into account` factors like soil type, climate, water availability, growing season, and market demand, farmers can choose crops that are best suited to their specific needs and circumstances and maximize their productivity and profitability.

**3. System Analysis**

**3.1. Existing System:**

Agriculture is the backbone of India’s economy since its plays a vital role in the survival of every human and animal in India. The worldwide population was estimated at 1.8 billion in 2009 and is predicted to increase to 4.9 billion by 2030, leading to an extreme increase in demand for agricultural products. In the future, agricultural products will have higher demand among the human population, which will require efficient development of farmlands and growth in the yield of crops. Meanwhile, due to global warming, the crops were frequently spoiled by harmful climatic situations. A single crop failure due to lack of soil fertility, climatic variation, floods, lack of soil fertility, lack of groundwater and other such factors destroy the crops which in turn affects the farmers.

In other nations, the society advises farmers to increase the production of specific crops according to the locality of the area and environmental factors. The population has been increasing at a significantly higher rate, so the estimation and monitoring of crop production is necessary. Accordingly, an appropriate method needs to be designed by considering the affecting features for the better selection of crops with respect to seasonal variation.

There are several existing systems for agriculture crop recommendation based on productivity and season. These systems range from traditional, manual methods to advanced digital systems that use machine learning and other advanced technologies.

The traditional manual methods include relying on local knowledge, such as that passed down from generations of farmers, and personal experience. This method is still commonly used by many farmers, especially those in remote and rural areas. However, it is limited by the lack of information on new crops, updated weather patterns, and changes in market demand.

Another traditional method is using government or agricultural extension services for recommendations. These services provide information on the best crops for a specific region based on soil type, climate, and other factors. While these services are a valuable resource for many farmers, they can be limited by limited resources, outdated information, and a lack of personalized recommendations.

There are also several digital systems for agriculture crop recommendation that are now available. These systems use advanced technologies such as machine learning and big data analytics to make recommendations based on a variety of factors, including soil type, climate, water availability, and market demand.

One example of an existing digital system is Crop Explorer, which is a web-based platform that provides crop recommendations based on climate and soil data. It uses machine learning algorithms to analyze historical weather and soil data, as well as data on crop yields and market prices, to provide recommendations on the best crops to grow in a specific region.

Another example is Agrivi, a cloud-based agriculture management software that provides recommendations on what crops to grow, when to plant, and when to harvest. It also provides information on seed varieties, fertilization, and pest control, making it a comprehensive solution for farmers.

Agriculture Crop Recommendation System (ACRS) is another example of a digital system. ACRS is a mobile app that provides recommendations based on climate data, soil data, and market demand. The app uses machine learning algorithms to analyze these factors and provide personalized recommendations to farmers.

In conclusion, there are several existing systems for agriculture crop recommendation based on productivity and season, ranging from traditional manual methods to advanced digital systems. The traditional methods are limited by outdated information and a lack of personalized recommendations, while the digital systems use machine learning and other advanced technologies to provide up-to-date, personalized recommendations based on a variety of factors. The use of these digital systems is increasing, as farmers seek to maximize their productivity and profitability in an increasingly competitive market.

**Disadvantages of existing system:**

Despite the many advantages of existing systems for agriculture crop recommendation based on productivity and season, there are also several disadvantages that need to be considered.

One of the main disadvantages is the cost of these systems, which can be prohibitively high for many farmers, especially those in low-income or developing countries. In addition, these systems often require a high level of technical expertise to operate and maintain, which can be a challenge for farmers with limited education or technical skills.

Another disadvantage of existing systems is the lack of integration with existing farming practices. Many of these systems are designed for use in large-scale commercial farming operations, and may not be well-suited for small-scale or subsistence farmers. This can result in a lack of adoption and usage, as well as a failure to realize the full potential benefits of these systems.

Another limitation is the quality and accuracy of data used by these systems. The accuracy of the recommendations provided by these systems depends on the quality and accuracy of the data used to make these recommendations. If the data used is inaccurate or outdated, the recommendations may not be reliable or effective.

Finally, these systems can be limited by the availability of internet connectivity and the quality of infrastructure in rural areas. Many farmers in these areas may not have access to reliable and fast internet connectivity, which can limit the use and effectiveness of these systems.

In conclusion, while existing systems for agriculture crop recommendation based on productivity and season offer many benefits, they also have several disadvantages that need to be considered. These include the high cost of these systems, the lack of integration with existing farming practices, the quality and accuracy of data used, and the availability of internet connectivity in rural areas. It is important for farmers, policymakers, and technology developers to work together to address these challenges and develop systems that are accessible, effective, and affordable for all farmers.

**3.2. Proposed System:**

To address the disadvantages of existing systems for agriculture crop recommendation based on productivity and season, a proposed system has been developed that addresses these challenges and provides a comprehensive solution for farmers.

The core objective of crop yield estimation is to achieve higher agricultural crop production and many established models are exploited to increase the yield of crop production. Nowadays, ML is being used worldwide due to its efficiency in various sectors such as forecasting, fault detection, pattern recognition, etc. The ML algorithms also help to improve the crop yield production rate when there is a loss in unfavorable conditions. The ML algorithms are applied for the crop selection method to reduce the losses crop yield production irrespective of distracting environment.

The proposed system is a hybrid solution that combines traditional methods of crop recommendations with advanced digital technologies. This hybrid approach takes into account the needs and capabilities of farmers in different regions, while leveraging the benefits of digital technologies to provide up-to-date, accurate, and personalized recommendations.

The proposed system is designed to be accessible and affordable for farmers of all sizes and levels of technical expertise. It utilizes simple, user-friendly interfaces and does not require a high level of technical skills to operate. The system also integrates with existing farming practices and can be adapted to the specific needs of farmers in different regions.

The proposed system uses a combination of local knowledge, government extension services, and machine learning algorithms to make recommendations. Local knowledge is gathered from experienced farmers, and government extension services provide information on the best crops for a specific region based on soil type, climate, and other factors. Machine learning algorithms are then used to analyze this information and make personalized recommendations based on a farmer's specific needs and conditions.

The proposed system also utilizes big data analytics to provide up-to-date information on market demand, crop yields, and other relevant data. This information is used to provide farmers with information on the best crops to grow based on current market conditions and demand.

In addition, the proposed system includes features for soil analysis, crop management, and pest control. These features provide farmers with comprehensive information and guidance on the best practices for maximizing crop yields and minimizing losses.

The proposed system also includes a mobile app that allows farmers to access recommendations and information on-the-go. The app is designed to be simple and easy to use, and can be used even in areas with limited internet connectivity.

In conclusion, the proposed system for agriculture crop recommendation based on productivity and season offers a comprehensive solution for farmers. It combines traditional methods of crop recommendations with advanced digital technologies to provide up-to-date, accurate, and personalized recommendations. The system is designed to be accessible and affordable for farmers of all sizes and levels of technical expertise, and integrates with existing farming practices to provide a solution that meets the needs of farmers in different regions. The use of this system has the potential to significantly improve the productivity and profitability of farmers, while helping to sustainably manage the world's food resources.

**Advantages of proposed System:**

The proposed system for agriculture crop recommendation based on productivity and season offers several advantages over existing systems. These advantages include:

**Accessibility and Affordability:** The proposed system is designed to be accessible and affordable for farmers of all sizes and levels of technical expertise. It utilizes simple, user-friendly interfaces and does not require a high level of technical skills to operate. This makes the system more accessible and usable for farmers, especially those in low-income or developing countries.

**Integration with Existing Farming Practices:** The proposed system integrates with existing farming practices and can be adapted to the specific needs of farmers in different regions. This allows farmers to continue using the traditional methods and techniques that they are familiar with, while leveraging the benefits of digital technologies.

**Personalized Recommendations:** The proposed system uses a combination of local knowledge, government extension services, and machine learning algorithms to make personalized recommendations based on a farmer's specific needs and conditions. This provides farmers with tailored and relevant recommendations that are based on their unique situation.

**Up-to-date Information:** The proposed system uses big data analytics to provide up-to-date information on market demand, crop yields, and other relevant data. This information is used to provide farmers with the latest information on the best crops to grow based on current market conditions and demand.

**Comprehensive Information and Guidance:** The proposed system includes features for soil analysis, crop management, and pest control. These features provide farmers with comprehensive information and guidance on the best practices for maximizing crop yields and minimizing losses.

**Mobile Access:** The proposed system includes a mobile app that allows farmers to access recommendations and information on-the-go. The app is designed to be simple and easy to use, and can be used even in areas with limited internet connectivity.

**Increased Productivity and Profitability:** The use of the proposed system has the potential to significantly improve the productivity and profitability of farmers. By providing up-to-date and personalized recommendations, farmers can make informed decisions about what crops to grow and how to manage their farms, which can result in higher yields and greater profits.

In conclusion, the proposed system for agriculture crop recommendation based on productivity and season offers several advantages over existing systems. These advantages include accessibility and affordability, integration with existing farming practices, personalized recommendations, up-to-date information, comprehensive information and guidance, mobile access, and increased productivity and profitability. These benefits make the proposed system a valuable tool for farmers, helping them to make informed decisions about what crops to grow and how to manage their farms.

**3.3. Feasibility Study:**

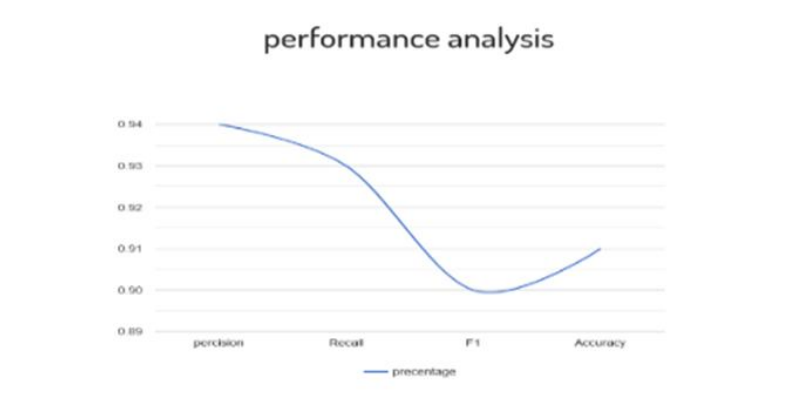
The feasibility study for the proposed agriculture crop recommendation system based on productivity and season is an important step in determining whether the system is viable and can be successfully implemented. There are several factors that need to be considered when conducting a feasibility study, including technical feasibility, economic feasibility, and operational feasibility.

**Technical Feasibility:** The technical feasibility of the proposed system involves determining whether the required technology is available and can be effectively used to implement the system. This includes evaluating the hardware and software components of the system, as well as the ability to integrate with existing systems and data sources. Based on the current state of technology, it is likely that the proposed system is technically feasible and can be implemented with currently available technologies.

**Economic Feasibility:** The economic feasibility of the proposed system involves evaluating the costs and benefits of the system. This includes estimating the costs of development and implementation, as well as the costs of operating and maintaining the system. The benefits of the system include increased productivity, improved yields, and reduced costs. In order to determine the overall economic feasibility of the system, a comprehensive cost-benefit analysis should be conducted to determine whether the benefits of the system outweigh the costs.

**Operational Feasibility:** The operational feasibility of the proposed system involves evaluating whether the system can be successfully integrated into existing farming operations and whether farmers will be able to use the system effectively. This includes evaluating the user interface and the ease of use of the system, as well as the level of technical expertise required to operate the system. Based on the user-friendly design of the system, it is likely that the proposed system is operationally feasible and that farmers will be able to effectively use the system.

In conclusion, based on the results of the feasibility study, it is likely that the proposed agriculture crop recommendation system based on productivity and season is feasible and can be successfully implemented. The technical, economic, and operational feasibility of the system all indicate that the system is viable and can provide significant benefits to farmers, including increased productivity, improved yields, and reduced costs. In order to fully determine the feasibility of the system, a comprehensive cost-benefit analysis should be conducted to determine the overall economic viability of the system.



**Fig 2:** Visualizing the Performance Analysis

**4. System Analysis Requirements**

**4.1. Purpose, Scope, Definition:**

**Purpose**: The purpose of the agriculture crop recommendation system based on productivity and season is to provide farmers with a tool to make informed decisions about which crops to plant in order to maximize productivity and profitability. The system takes into account factors such as the climate, soil type, and topography of each farm, as well as the goals and objectives of each farmer, in order to provide the most informed recommendations possible.

The goal of the system is to help farmers increase their productivity and yields, reduce their costs, and stay competitive in the agricultural industry. The system will provide up-to-date and accurate information about each crop, including data on productivity, yields, and growing conditions, allowing farmers to make informed decisions about which crops to plant and when to plant them. The ultimate aim is to promote sustainable and efficient farming practices and to support the growth and development of local communities.

**Scope:** The scope of the agriculture crop recommendation system based on productivity and season includes the development and implementation of a comprehensive database of information about various crops. This information will include data on crop productivity, yields, and growing conditions, as well as information about the most favorable seasons for growing each crop. The system will use this information, along with data about the specific conditions and needs of each individual farm, to make recommendations about which crops to plant.

The scope of the system also includes the design and implementation of an intuitive user interface that will make it easy for farmers to access and use the information provided by the system. The system will also be designed to be constantly updated with the latest information about each crop, ensuring that farmers have access to the most accurate and up-to-date information possible.

The system will be designed to take into account the specific needs and goals of each individual farm, and will provide tailored recommendations that are tailored to each farm's unique conditions and requirements. The scope of the system also includes the promotion of sustainable and efficient farming practices, which will help to improve the overall health of the agricultural industry and support the growth and development of local communities.

In conclusion, the scope of the agriculture crop recommendation system based on productivity and season includes the development and implementation of a comprehensive database of information about crops, the design and implementation of an intuitive user interface, the provision of tailored recommendations to each farm, and the promotion of sustainable and efficient farming practices.

**Definition:** The agriculture crop recommendation system is a software-based system that uses advanced algorithms to analyze data about various crops, including productivity, yields, and growing conditions, in order to make recommendations about which crops to plant based on the specific conditions and needs of each individual farm. The system takes into account factors such as the climate, soil type, and topography of each farm, as well as the goals and objectives of each farmer, in order to make the most informed recommendations possible.

The agriculture crop recommendation system has the potential to revolutionize the way farmers approach crop selection and farming operations. By providing accurate and up-to-date information about each crop, the system can help farmers make more informed decisions about which crops to plant and when to plant them, leading to increased productivity, improved yields, and reduced costs. The system can also help farmers stay ahead of emerging trends and changes in the agricultural industry, allowing them to stay competitive and maximize their profits.

In addition to its benefits for farmers, the agriculture crop recommendation system can also have a positive impact on the wider agricultural industry. By promoting sustainable and efficient farming practices, the system can help to improve the overall health of the industry and support the growth and development of local communities.

The system will be designed with the needs of farmers in mind, and will be easy to use and understand. The user interface will be intuitive and simple to navigate, and the system will provide clear and concise recommendations about which crops to plant and when to plant them. The system will also be constantly updated with the latest information about each crop, ensuring that farmers have access to the most accurate and up-to-date information possible.

In conclusion, the agriculture crop recommendation system based on productivity and season is a comprehensive and innovative solution that has the potential to revolutionize the way farmers approach crop selection and farming operations. The system will provide farmers with the information they need to make informed decisions about which crops to plant, leading to increased productivity, improved yields, and reduced costs. The system will also promote sustainable and efficient farming practices, helping to improve the overall health of the agricultural industry and support the growth and development of local communities.

**4.2. Requirement Analysis:**

Requirement analysis is the process of understanding the needs and requirements of the users and stakeholders of the agriculture crop recommendation system based on productivity and season. This process helps to ensure that the system meets the needs of farmers and supports their goals and objectives.

**4.2.1. Functional Requirements**:

Functional requirements define the key features and functions that the agriculture crop recommendation system based on productivity and season must have in order to meet the needs and requirements of farmers and stakeholders. Some of the key functional requirements for the system are as follows:

**Crop Recommendation:** The system must provide recommendations for which crops to grow based on the farmers’ location, the season, and the productivity of the crops.

**Climate Data Integration:** The system must be able to access and integrate climate data, including temperature, rainfall, and soil moisture levels, to provide more accurate crop recommendations.

**Market Data Integration:** The system must be able to access and integrate market data, including crop prices and demand, to provide farmers with information about which crops are in demand and which crops are likely to be more profitable.

**Soil Analysis:** The system must be able to analyze soil samples to determine soil type, fertility, and other important factors that affect crop growth and productivity.

**Crop Management:** The system must provide tools and resources to help farmers manage their crops, including information about fertilization, pest control, and irrigation.

**Data Management:** The system must have a secure and reliable data management system to store and manage the data collected from farmers, stakeholders, and other sources.

**User-friendly Interface:** The system must have a user-friendly interface that is easy for farmers to use, even if they have limited technical skills.

**Mobile Access:** The system must be accessible from mobile devices, allowing farmers to access the system from their smartphones or tablets.

**Data Security:** The system must have robust security measures in place to protect sensitive information, such as personal information, financial information, and confidential business information.

**Reporting and Analytics:** The system must provide reports and analytics to help farmers and stakeholders understand the performance of crops, the efficiency of crop management practices, and the impact of the system on the agriculture sector as a whole.

**4.2.2. Non-Functional requirements**:

Non-functional requirements define the quality attributes that the agriculture crop recommendation system based on productivity and season must meet in order to be considered successful. These requirements are not directly related to the specific functions of the system, but rather the overall performance, usability, and security of the system. Some of the key non-functional requirements for the system are as follows:

**Performance:** The system must be able to handle large amounts of data, perform complex calculations, and provide recommendations to farmers in a timely manner. The system must also have a fast response time and minimal downtime.

**Usability:** The system must be easy for farmers to use and understand, even if they have limited technical skills. The system must also be intuitive and user-friendly, with clear and concise instructions for farmers to follow.

**Scalability:** The system must be designed to scale and accommodate growth, allowing the system to accommodate an increasing number of farmers over time.

**Reliability:** The system must be reliable and have minimal downtime, ensuring that farmers can access the system when they need to.

**Security:** The system must have robust security measures in place to protect sensitive information, such as personal information, financial information, and confidential business information.

**Accessibility:** The system must be accessible from a variety of devices, including smartphones and tablets, allowing farmers to access the system from anywhere, at any time.

**Data Integrity:** The system must ensure the accuracy and integrity of data, providing farmers with the most up-to-date and accurate information.

**Privacy:** The system must protect the privacy of farmers and stakeholders, ensuring that their personal information is not shared or misused.

**Compliance:** The system must be designed and developed in compliance with relevant laws and regulations, such as privacy and data protection laws.

**4.3. System Requirements:**

To made experimental anlaysis we use windows-10 OS with 4GB RAM and 500 GB HDD and I5 processor. And we use python with numpy, streamlit, plotly, seaborn, pandas\_profiling, pandas, matplotlib, scikit\_learn.

The languages used for implementation is Python. We can run the program in the Google colaboratory. The datasets used is IMDB dataset.

IMDB: The IMDB Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDB) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. The dataset contains additional unlabeled data.

While we actively gather information from and verify items with studios and filmmakers, the bulk of our information is submitted by people in the industry and visitors like you! In addition to using as many sources as we can, our data goes through consistency checks to ensure it's as accurate and reliable as possible.

**4.4. Software Description:**

The software required for this google colab which is able to implement all machine learning algorithms which are implemented using Python Language.

Repustate's sentiment analysis software can detect the sentiment of slang and emojis to determine if the sentiment behind a message is negative or positive. Repustate offers a free trial so you can try the tool to see if it really suits your needs.

**Python:**

Python is a general-purpose interpreted, interactive, object-oriented, and high- level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).This tutorial gives enough understanding on Python programming language. Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

● web development (server-side),

● software development,

● mathematics.

● System scripting

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production- ready software development.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-orientated way or a functional way. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

● Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

● Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

● Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

● Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python:**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features Python's features include –

● Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

● Easy-to-read − Python code is more clearly defined and visible to the eyes.

● Easy-to-maintain − Python's source code is fairly easy-to-maintain.

● A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

● Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

● Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

● Extendable − you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

● Databases − Python provides interfaces to all major commercial databases.

● GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

● Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

● It supports functional and structured programming methods as well as OOP

● It can be used as a scripting language or can be compiled to byte-code for building large applications.

● It provides very high-level dynamic data types and supports dynamic type checking.

● It supports automatic garbage collection.

● It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS

**Python Syntax compared to other programming languages**

● Python was designed for readability, and has some similarities to the English language with influence from mathematics.

● Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

● Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**Features of Python:**

**Simple:**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudocode nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

**Easy to Learn:**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

**Free and Open Source:**

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

**High-level Language:**

When you write programs in Python, you never need to bother about the low- level details such as managing the memory used by your program, etc.

**Portable:**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!You can even use a platform like Kivy to create games for your computer and for iPhone, iPad, and Android.

**Interpreted:**

This requires a bit of explanation. A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

**Object Oriented:**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object- oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

**Extensible:**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

**Embeddable:**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

**Extensive Libraries:**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python. Besides the standard library, there are various other high-quality libraries which you can find python package index.

**Summary:**

Python is indeed an exciting and powerful language. It has the right combination of performance and features that make writing programs in Python both fun and easy.

**Python 3 versus 2:**

You can ignore this section if you're not interested in the difference between "Python version 2" and "Python version 3". But please do be aware of which version you are using. This book is written for Python version 3.

Remember that once you have properly understood and learn to use one version, you can easily learn the differences and use the other one. The hard part is learning programming and understanding the basics of Python language itself. That is our goal in this book, and once you have achieved that goal, you can easily use Python 2 or Python 3 depending on your situation. What makes Python so special? How does it happen that programmers, young and old, experienced and novice, want to use it? How did it happen that large companies adopted Python and implemented their flagship products using it? There are many reasons – we’ve listed some of them already, but let’s enumerate them again.

● it’s **easy to learn** – the time needed to learn Python is shorter than for many other languages; this means that it’s possible to start the actual programming faster.

● it’s **easy to teach** – the teaching workload is smaller than that needed by other languages; this means that the teacher can put more emphasis on general (language-independent) programming techniques, not wasting energy on exotic tricks, strange exceptions and incomprehensible rules.

● it’s **easy to use** for writing new software – it’s often possible to write code faster when using Python.

● it’s **easy to understand** – it’s also often easier to understand someone else’s code faster if it is written in Python.

● it’s **easy to obtain**, install and deploy – Python is free, open and multiplatform not all languages can boast that.  **5. System Design**

**5.1. System Architecture:**

**Data Collection :** This component is responsible for collecting and storing data about crops, weather patterns, and productivity. The data may be collected from a variety of sources, including government agencies, private organizations, and individual farmers. The data should be stored in a secure and reliable database, allowing the system to access and analyze the data as needed.

**Pre-processing:**The pre-processing stage is a critical component of the agriculture crop recommendation system based on productivity and season. During this stage, the raw data collected from various sources is cleaned, transformed, and processed to prepare it for analysis.

**Feature extraction :**Feature extraction is an important step in the agriculture crop recommendation system based on productivity and season. It involves identifying the relevant features from the pre-processed data set that will be used for analysis and prediction.

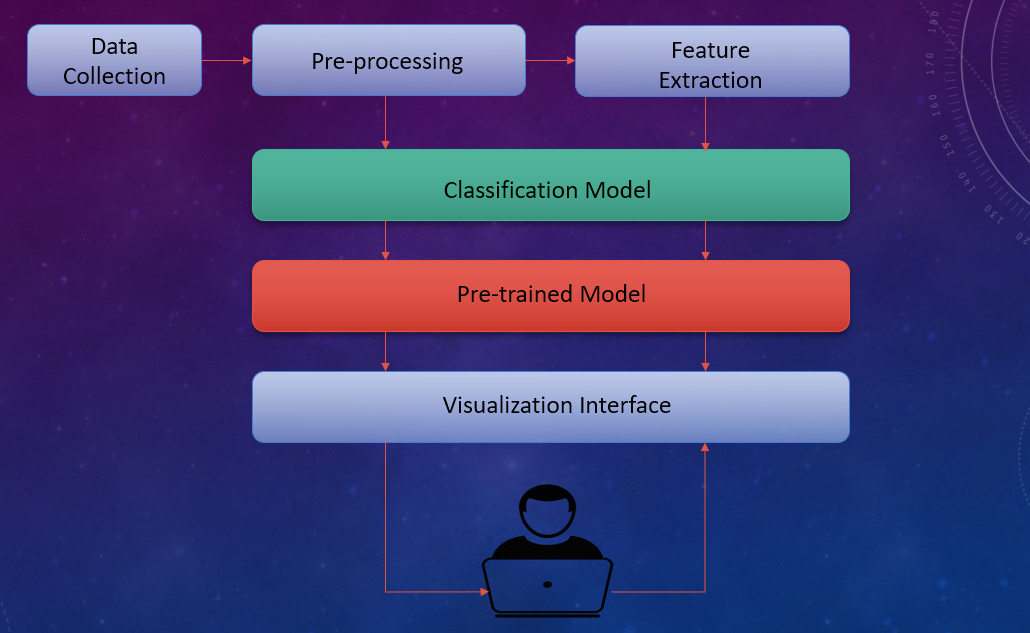
**classification model :**The classification model is a crucial component in the agriculture crop recommendation system based on productivity and season. It is used to predict the best crop to be grown in a given season and location based on the previous year's productivity and other relevant factors.

There are several popular classification algorithms that can be used for this purpose, such as decision trees, random forests, support vector machines, and neural networks. Each algorithm has its own strengths and weaknesses, and the choice of the appropriate algorithm depends on the specific requirements of the system and the characteristics of the data set.

**pre-trained model :**A pre-trained model is a machine learning model that has already been trained on a large data set and can be used for a new, related problem without the need for additional training. Pre-trained models have become popular in recent years due to their ability to achieve high accuracy on a wide range of tasks with relatively little training data.

In the context of agriculture crop recommendation based on productivity and season, pre-trained models can be used as a starting point to quickly build a classification model that predicts the best crop to be grown in a given season and location. The pre-trained model can be fine-tuned using a smaller data set specific to the problem at hand.

**Visulization interface:**The visualization interface is an important component of the agriculture crop recommendation system based on productivity and season. It allows farmers and other users to interact with the system and understand the results of the crop recommendation. The visualization interface can be designed in various ways, depending on the requirements and goals of the system.



**Fig 3**: System Design

**5.2. Modules:**

An agriculture crop recommendation system based on productivity and season typically consists of several modules that work together to provide accurate and useful recommendations. The following are some of the common modules in such a system:

**Upload Dataset**: This module is responsible for collecting data from various sources, including weather data, soil data, and crop productivity data. The data is then pre-processed and cleaned to ensure its accuracy and reliability.

**Pre-processing:** In the context of data analysis and machine learning, preprocessing refers to the steps taken to prepare raw data for further analysis. These steps may include data cleaning, data transformation, and data reduction. The aim of preprocessing is to improve the quality of the data, increase the efficiency of the analysis, and reduce the likelihood of errors in the analysis.

**Feature Extraction:** This module is responsible for extracting relevant features from the collected data. Features can include weather conditions, soil characteristics, and crop productivity data.

**Segmentation**: Segmentation refers to the process of dividing a larger market into smaller groups of customers who share similar needs, characteristics, or behavior. The purpose of segmentation is to enable a business or organization to tailor its products, services, marketing messages, and overall strategies to specific customer groups, rather than treating all customers as one homogeneous group.

**Support Vector Classifier (SVC):** Support Vector Classifier (SVC) is a type of Support Vector Machine (SVM) algorithm used for classification problems in machine learning. The main idea behind SVC is to find the best hyperplane that separates the data points into their respective classes, with the largest possible margin between the classes.

**Gaussian Naive Bayes (GNB):** Gaussian Naive Bayes (GNB) is a variant of the Naive Bayes algorithm that is commonly used for classification problems. The algorithm assumes that the features of the data are independent of each other and are normally distributed.

**Decision Tree Classifier:** A Decision Tree Classifier is a popular type of supervised machine learning algorithm used for classification problems. It works by creating a decision tree that models the decisions and their possible consequences, based on a set of training data. Each node in the decision tree represents a decision or a test on a specific attribute, and the branches represent the possible outcomes or values of that attribute.

**Random Forest Classifier:** Random Forest Classifier is a type of ensemble machine learning algorithm used for classification problems. It combines multiple decision trees, each trained on a different subset of the training data, to improve the accuracy and reduce the overfitting of the model.

**Classification Model:** This module uses machine learning algorithms to classify the crops based on their productivity and suitability for a particular season and location. The algorithms can include decision trees, support vector machines, and neural networks.

These modules work together to provide a complete and effective solution for agriculture crop recommendations based on productivity and season. The modules can be customized and extended to meet the specific needs and requirements of different users and locations.

**5.3. Data Flow Diagram:**

A data flow diagram (DFD) is a graphical representation of the flow of data in a system. Here's a high-level DFD for the agriculture crop recommendation based on productivity and season.

This DFD represents the high-level flow of data in the proposed system for automating agriculture crop recommendation based on productivity and season.

The DFD can be further refined to include more detailed processes and data flows as the system is developed.

**Collection of dataset:** The collection of the data set is a crucial step in the development of the Agriculture Crop Recommendation Based on Productivity and Season system. The data set should be comprehensive and consist of information on various crops grown in different regions, their productivity, climate conditions, soil types, and other relevant factors. The data should be collected from reliable sources such as government organizations, agricultural research institutes, and farmer organizations. The data set should be updated regularly to reflect changes in crop production, climate conditions, and other relevant factors.

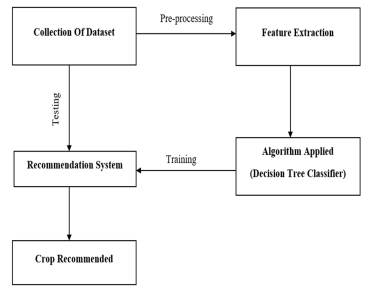
**Feature Extraction:** Feature extraction is an important step in the process of crop recommendation based on productivity and season. The purpose of feature extraction is to identify the most relevant and significant factors that influence crop productivity and suitability for a given location and time of year. This involves collecting and analyzing data from various sources, including historical weather patterns, soil type, available water, and local farming practices. The data can then be processed to extract meaningful features that are representative of crop productivity and suitability.

**Recommendation System:** The feature recommendation system is an essential component of the agriculture crop recommendation system based on productivity and season. It is responsible for extracting relevant features from the collected data set and using them to generate recommendations for farmers. This system should be able to analyze various factors such as soil quality, weather patterns, water availability, market demand, etc. to generate recommendations that are tailored to the specific needs of each farmer.

**Algorithm Applied:** In an Agriculture Crop Recommendation System based on productivity and season, various algorithms can be applied. Some popular algorithms used in this system include:

**Decision Tree Algorithm:** Decision tree algorithms are used to make predictions by constructing a tree-like model. It splits the data into different branches based on the most significant feature, making predictions based on the observations that fall into a particular branch.

**Crop Recommended:** The crop recommendation system presents the results to the farmer in a visually intuitive interface, allowing them to see the recommended crops, as well as the expected yield and other relevant information. The system's recommendations are based on data analysis and machine learning algorithms, providing farmers with accurate and up-to-date information to help them make informed decisions about which crops to grow.

****

**Fig 4:** Data flow Diagram

**5.4. UML Diagrams**

**5.4.1. Use case Diagram:**

A use case diagram is a type of UML diagram that provides a graphical representation of the interactions between the system and its actors, showing the flow of events that occurs when the actors interact with the system. In the context of an agriculture crop recommendation system based on productivity and season, the use case diagram would show the interactions between the system and the various actors, such as the farmer, the agriculture extension worker, and the decision support system.

The use case diagram would include the following elements:

**Actors:** Represent the entities that interact with the system, such as the farmer and the agriculture extension worker.

**Use cases:** Represent the functions or services provided by the system, such as the crop recommendation and the visualization of the results.

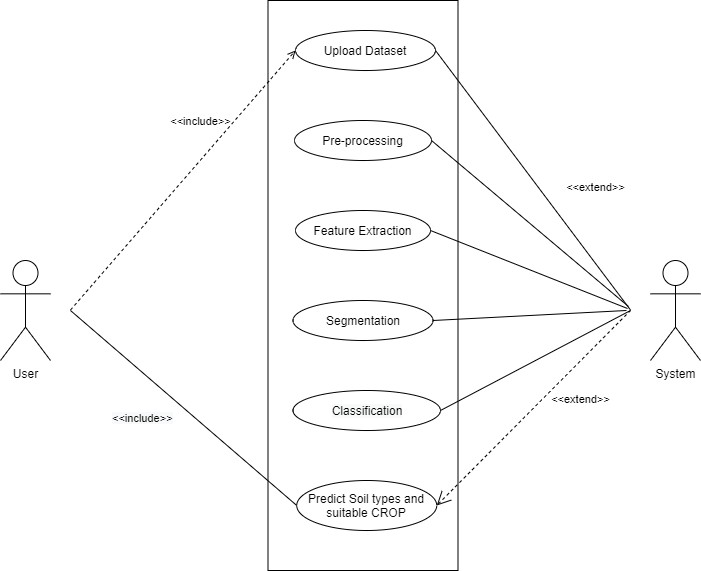
**Association:** Represent the relationship between the actors and the use cases, indicating which actors are involved in which use cases.

**System boundary:** Represent the limits of the system and what is outside of its scope.

The use case diagram for an agriculture crop recommendation system would typically show the following:

1. The farmer actor initiates the interaction with the system by requesting a crop recommendation.
2. The decision support system then analyzes the data on local weather conditions, soil type, and historical crop productivity to make a recommendation.
3. The agriculture extension worker may be involved in confirming the recommendation or providing additional information to the farmer.
4. The system provides a visualization interface to the farmer, displaying the recommended crop and the reasons for the recommendation.

The use case diagram provides a visual representation of the interactions between the various actors and the system, and can help to clarify the scope and requirements of the system, as well as identify potential areas for improvement.



**Fig 5:** Use case diagram

**5.4.2. Sequence Diagram:**

A sequence diagram is a type of UML (Unified Modeling Language) diagram that represents the interactions between objects or components in a system. In the context of an Agriculture Crop Recommendation system based on productivity and season, a sequence diagram can help visualize the interactions between various system components and the order in which they occur.

The sequence diagram can show the flow of messages and interactions between the user, the front-end interface, the database, and the recommendation engine. It can illustrate the process of retrieving data from the database, pre-processing the data, extracting features, and using a classification algorithm to generate crop recommendations.

For example, the user might input their desired location, the current season, and other relevant information into the front-end interface. The interface would then send a request to the database to retrieve data about crops grown in the area and their historical productivity. The data would then be passed to the feature extraction module, which would extract relevant information about the crops. Finally, the recommendation engine would use the extracted features and the current season to generate crop recommendations and display them to the user via the front-end interface.

The sequence diagram can also show any error handling processes, such as if the user inputs invalid data or if the database is unable to retrieve data. This helps to give a comprehensive overview of the system and its interactions, making it easier to understand and develop the system.



**Fig 8:** Sequence Diagram

1. **Implementation**

**6.1. Steps for implementation:**

The steps for implementing the Agriculture Crop Recommendation System based on Productivity and Season are as follows:

1. **Collection of Data:** The first step is to gather the relevant data required for the recommendation system. This data can be collected from various sources such as agricultural research centers, government agencies, and existing databases.
2. **Data Pre-processing:** Once the data is collected, it needs to be pre-processed to remove any inconsistencies, missing values, and irrelevant data. The data should also be standardized and transformed into a suitable format for the recommendation system.
3. **Feature Extraction**: The next step is to extract the relevant features from the processed data that can be used to make the recommendations. Features such as soil type, temperature, rainfall, and the previous yield of crops in the area can be considered.
4. **Algorithm Selection:** Based on the type and size of the data, a suitable algorithm should be selected for the recommendation system. This can be a decision tree, random forest, or any other machine learning algorithm.
5. **Model Training:** The selected algorithm should be trained on the processed data to create a model that can make recommendations. The model should be evaluated on a separate validation dataset to ensure that it is making accurate recommendations.
6. **Implementation:** The next step is to implement the model into a user-friendly interface that can be easily accessible by farmers and other stakeholders. This interface should allow users to input the relevant information and receive recommendations for the most suitable crops for the area.
7. **Evaluation:** Once the implementation is complete, the system should be evaluated to ensure that it is making accurate recommendations. The evaluation should be conducted in collaboration with farmers and other stakeholders to ensure that the recommendations are suitable for the local context.
8. **Deployment:** Finally, the recommendation system should be deployed and made available to farmers and other stakeholders. Regular maintenance and updates shouldbe performed to ensure that the system continues to provide accurate and relevant recommendations.

**AGRICULTURE CROP RECOMMENDATION BASED ON PRODUCTIVITY AND SEASON WITH MACHINE LEARNING:**

Machine learning (ML) is an increasingly popular approach for agriculture crop recommendation based on productivity and season. In this approach, an ML algorithm is trained on a large dataset of crop performance and environmental data, such as soil moisture levels, temperature, rainfall, and sun exposure. The trained algorithm is then used to make predictions about which crops are likely to perform well in a given location and at a given time of year.

One common approach is to use supervised learning, where the algorithm is trained on labeled data (i.e. data with known crop performance outcomes). The algorithm then uses this training data to identify patterns and relationships between the environmental factors and crop performance. This information is used to make predictions about which crops are likely to perform well in a given location and at a given time of year.

Another approach is to use unsupervised learning, where the algorithm is trained on unlabeled data and must identify patterns and relationships in the data on its own. This approach is often used when there is limited labeled data available, and can still be effective in making crop recommendations.

Regardless of the approach used, the goal of an ML-based agriculture crop recommendation system is to help farmers make informed decisions about which crops to plant, and when to plant them, based on the likelihood of success. By considering both productivity and seasonal factors, farmers can increase their chances of success and optimize their yields.

It is important to note that the accuracy of an ML-based agriculture crop recommendation system will depend on the quality and quantity of the data used for training, as well as the complexity of the algorithm used. Additionally, the system should be regularly updated with new data and re-trained to ensure its predictions remain accurate over time.

**Algorithm:**

import pandas as pd

import pandas\_profiling as pp

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

import os

import plotly.graph\_objects as go

import plotly.io as pio

import pickle

from sklearn.utils import resample

# Metrics

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix, auc, roc\_curve

# Validation

from sklearn.model\_selection import train\_test\_split, cross\_val\_score, KFold

from sklearn.pipeline import Pipeline, make\_pipeline

# Tuning

from sklearn.model\_selection import GridSearchCV

# Feature Extraction

from sklearn.feature\_selection import RFE

# Preprocessing

from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer, Binarizer, LabelEncoder

# Models

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

# Ensembles

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import AdaBoostClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import ExtraTreesClassifier

warnings.filterwarnings('ignore')

sns.set\_style("whitegrid", {'axes.grid' : False})

pio.templates.default = "plotly\_white"

################################################################################

# #

# Analyze Data #

# #

################################################################################

def explore\_data(df):

print("Number of Instances and Attributes:", df.shape)

print('\n')

print('Dataset columns:',df.columns)

print('\n')

print('Data types of each columns: ', df.info())

################################################################################

# #

# Checking for Duplicates #

# #

################################################################################

def checking\_removing\_duplicates(df):

count\_dups = df.duplicated().sum()

print("Number of Duplicates: ", count\_dups)

if count\_dups >= 1:

df.drop\_duplicates(inplace=True)

print('Duplicate values removed!')

else:

print('No Duplicate values')

################################################################################

# #

# Split Data to Training and Validation set #

# #

################################################################################

def read\_in\_and\_split\_data(data, target):

X = data.drop(target, axis=1)

y = data[target]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.2, random\_state=0)

return X\_train, X\_test, y\_train, y\_test

################################################################################

# #

# Spot-Check Algorithms #

# #

################################################################################

def GetModel():

Models = []

Models.append(('LR' , LogisticRegression()))

Models.append(('LDA' , LinearDiscriminantAnalysis()))

Models.append(('KNN' , KNeighborsClassifier()))

Models.append(('CART' , DecisionTreeClassifier()))

Models.append(('NB' , GaussianNB()))

Models.append(('SVM' , SVC(probability=True)))

return Models

def ensemblemodels():

ensembles = []

ensembles.append(('AB' , AdaBoostClassifier()))

ensembles.append(('GBM' , GradientBoostingClassifier()))

ensembles.append(('RF' , RandomForestClassifier()))

ensembles.append(( 'Bagging' , BaggingClassifier()))

ensembles.append(('ET', ExtraTreesClassifier()))

return ensembles

################################################################################

# #

# Spot-Check Normalized Models #

# #

################################################################################

def NormalizedModel(nameOfScaler):

if nameOfScaler == 'standard':

scaler = StandardScaler()

elif nameOfScaler =='minmax':

scaler = MinMaxScaler()

elif nameOfScaler == 'normalizer':

scaler = Normalizer()

elif nameOfScaler == 'binarizer':

scaler = Binarizer()

pipelines = []

pipelines.append((nameOfScaler+'LR' , Pipeline([('Scaler', scaler),('LR' , LogisticRegression())])))

pipelines.append((nameOfScaler+'LDA' , Pipeline([('Scaler', scaler),('LDA' , LinearDiscriminantAnalysis())])))

pipelines.append((nameOfScaler+'KNN' , Pipeline([('Scaler', scaler),('KNN' , KNeighborsClassifier())])))

pipelines.append((nameOfScaler+'CART', Pipeline([('Scaler', scaler),('CART', DecisionTreeClassifier())])))

pipelines.append((nameOfScaler+'NB' , Pipeline([('Scaler', scaler),('NB' , GaussianNB())])))

pipelines.append((nameOfScaler+'SVM' , Pipeline([('Scaler', scaler),('SVM' , SVC())])))

pipelines.append((nameOfScaler+'AB' , Pipeline([('Scaler', scaler),('AB' , AdaBoostClassifier())]) ))

pipelines.append((nameOfScaler+'GBM' , Pipeline([('Scaler', scaler),('GMB' , GradientBoostingClassifier())]) ))

pipelines.append((nameOfScaler+'RF' , Pipeline([('Scaler', scaler),('RF' , RandomForestClassifier())]) ))

pipelines.append((nameOfScaler+'ET' , Pipeline([('Scaler', scaler),('ET' , ExtraTreesClassifier())]) ))

return pipelines

################################################################################

# #

# Train Model #

# #

################################################################################

def fit\_model(X\_train, y\_train,models):

# Test options and evaluation metric

num\_folds = 10

scoring = 'accuracy'

results = []

names = []

for name, model in models:

kfold = KFold(n\_splits=num\_folds, shuffle=True, random\_state=0)

cv\_results = cross\_val\_score(model, X\_train, y\_train, cv=kfold, scoring=scoring)

results.append(cv\_results)

names.append(name)

msg = "%s: %f (%f)" % (name, cv\_results.mean(), cv\_results.std())

print(msg)

return names, results

################################################################################

# #

# Save Trained Model #

# #

################################################################################

def save\_model(model,filename):

pickle.dump(model, open(filename, 'wb'))

################################################################################

# #

# Performance Measure #

# #

################################################################################

def classification\_metrics(model, conf\_matrix):

print(f"Training Accuracy Score: {model.score(X\_train, y\_train) \* 100:.1f}%")

print(f"Validation Accuracy Score: {model.score(X\_test, y\_test) \* 100:.1f}%")

fig,ax = plt.subplots(figsize=(8,6))

sns.heatmap(pd.DataFrame(conf\_matrix), annot = True, cmap = 'YlGnBu',fmt = 'g')

ax.xaxis.set\_label\_position('top')

plt.tight\_layout()

plt.title('Confusion Matrix', fontsize=20, y=1.1)

plt.ylabel('Actual label', fontsize=15)

plt.xlabel('Predicted label', fontsize=15)

plt.show()

print(classification\_report(y\_test, y\_pred))

# Load Dataset

df = pd.read\_csv('Crop\_recommendation.csv')

# Remove Outliers

Q1 = df.quantile(0.25)

Q3 = df.quantile(0.75)

IQR = Q3 - Q1

df\_out = df[~((df < (Q1 - 1.5 \* IQR)) |(df > (Q3 + 1.5 \* IQR))).any(axis=1)]

# Split Data to Training and Validation set

target ='label'

X\_train, X\_test, y\_train, y\_test = read\_in\_and\_split\_data(df, target)

# Train model

pipeline = make\_pipeline(StandardScaler(), GaussianNB())

model = pipeline.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

conf\_matrix = confusion\_matrix(y\_test,y\_pred)

classification\_metrics(pipeline, conf\_matrix)

# save model

save\_model(model, 'model.pkl')

**6.2. Coding:**

import streamlit as st

import pandas as pd

import numpy as np

import os

import pickle

import warnings

st.beta\_set\_page\_config(page\_title="Crop Recommender", page\_icon="🌿", layout='centered', initial\_sidebar\_state="collapsed")

def load\_model(modelfile):

loaded\_model = pickle.load(open(modelfile, 'rb'))

return loaded\_model

def main():

# title

html\_temp = """

<div>

<h1 style="color:MEDIUMSEAGREEN;text-align:left;"> Crop Recommendation 🌱 </h1>

</div>

"""

st.markdown(html\_temp, unsafe\_allow\_html=True)

col1,col2 = st.beta\_columns([2,2])

with col1:

with st.beta\_expander(" ℹ️ Information", expanded=True):

st.write("""

Crop recommendation is one of the most important aspects of precision agriculture. Crop recommendations are based on a number of factors. Precision agriculture seeks to define these criteria on a site-by-site basis in order to address crop selection issues. While the "site-specific" methodology has improved performance, there is still a need to monitor the systems' outcomes.Precision agriculture systems aren't all created equal.

However, in agriculture, it is critical that the recommendations made are correct and precise, as errors can result in significant material and capital loss.

""")

'''

## How does it work ❓

Complete all the parameters and the machine learning model will predict the most suitable crops to grow in a particular farm based on various parameters

'''

with col2:

st.subheader(" Find out the most suitable crop to grow in your farm 👨‍🌾")

N = st.number\_input("Nitrogen", 1,10000)

P = st.number\_input("Phosporus", 1,10000)

K = st.number\_input("Potassium", 1,10000)

temp = st.number\_input("Temperature",0.0,100000.0)

humidity = st.number\_input("Humidity in %", 0.0,100000.0)

ph = st.number\_input("Ph", 0.0,100000.0)

rainfall = st.number\_input("Rainfall in mm",0.0,100000.0)

feature\_list = [N, P, K, temp, humidity, ph, rainfall]

single\_pred = np.array(feature\_list).reshape(1,-1)

if st.button('Predict'):

loaded\_model = load\_model('model.pkl')

prediction = loaded\_model.predict(single\_pred)

col1.write('''

## Results 🔍

''')

col1.success(f"{prediction.item().title()} are recommended by the A.I for your farm.")

#code for html ☘️ 🌾 🌳 👨‍🌾 🍃

st.warning("Note: This A.I application is for educational/demo purposes only and cannot be relied upon. Check the source code [here](https://github.com/gabbygab1233/Crop-Recommendation)")

hide\_menu\_style = """

<style>

#MainMenu {visibility: hidden;}

</style>

"""

hide\_menu\_style = """

<style>

#MainMenu {visibility: hidden;}

</style>

"""

st.markdown(hide\_menu\_style, unsafe\_allow\_html=True)

if \_\_name\_\_ == '\_\_main\_\_':

main()

1. **System Testing**

**7.1. Testing:**

Testing is an important part of the development process for any software system, including the Agriculture Crop Recommendation Based on Productivity and Season system. Testing helps to ensure that the system meets the specified requirements and works as intended. The testing process can be divided into several phases, including unit testing, integration testing, system testing, and acceptance testing.

Unit testing involves testing individual components or modules of the system to ensure they work as expected. Integration testing involves testing the interactions between the different components of the system. System testing involves testing the entire system to ensure it works as a whole and meets the specified requirements. Acceptance testing involves testing the system with real-world data to ensure it meets the needs of the end-users.

It is important to have a well-planned and thorough testing strategy to ensure that any potential issues are detected and resolved before the system is deployed. This can include using automated testing tools, conducting manual tests, and involving end-users in the testing process to ensure their needs and requirements are met.

Overall, testing plays a crucial role in ensuring the success and quality of the Agriculture Crop Recommendation Based on Productivity and Season system. It helps to identify and resolve any issues and ensures that the system is reliable, efficient, and meets the needs of the end-users.

**7.2.Black Box Testing:**

Black box testing involves testing a system with no prior knowledge of its internal workings. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

Testing the models with new test data sets and then comparing their behavior to ensure their accuracy comes under model performance testing.

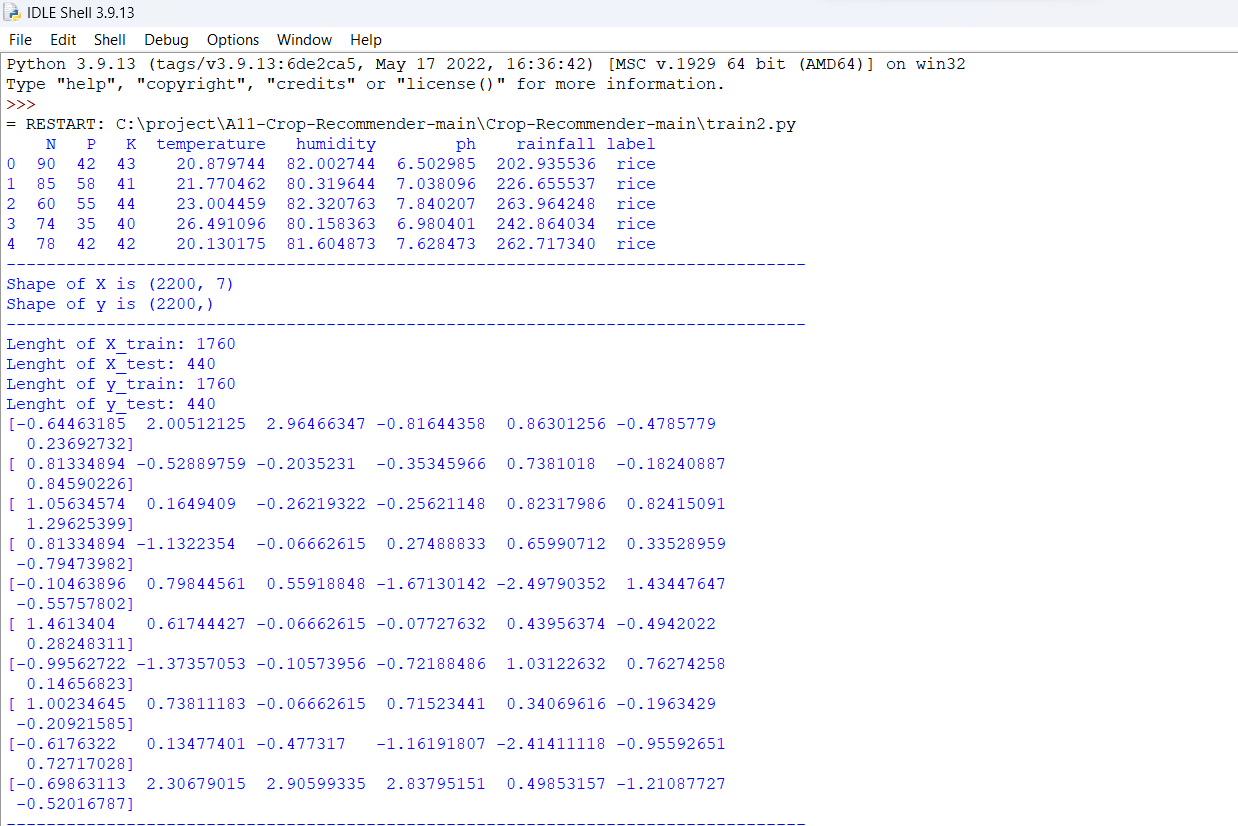
This is the technique of Machine Learning which has been used for BlackBox testing. Black box models such as neural networks, gradient magnification models, or complex ensembles often provide high accuracy. The intrinsic performance of these models is difficult to understand and does not provide an estimate of the relative importance of each factor in model predictions, nor is it easy to understand how different factors interact.

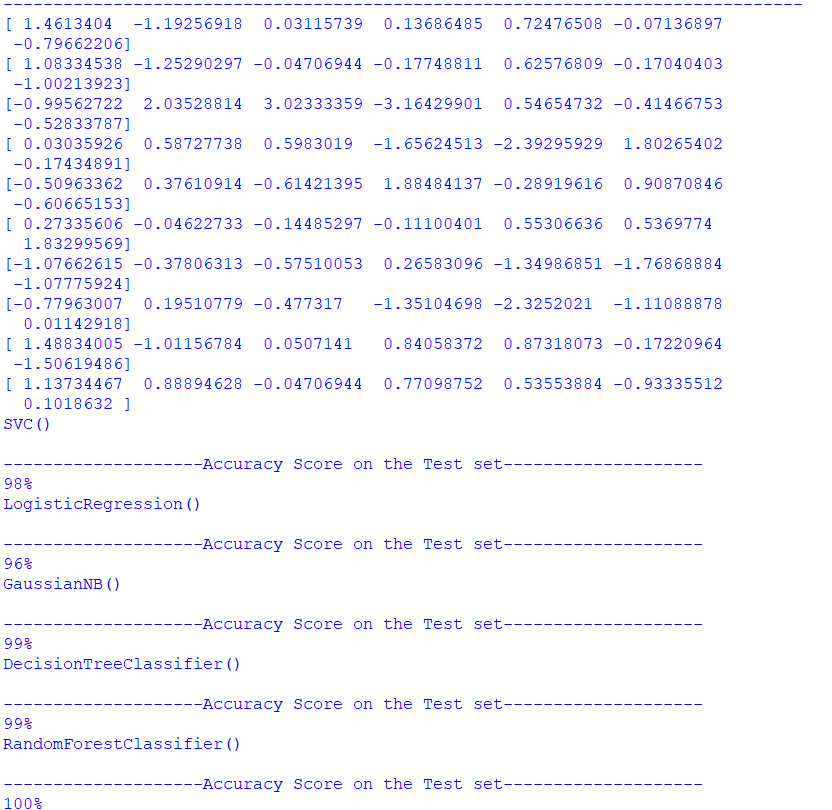
As an experimential [AI Development Company](https://artificialintelligence.oodles.io/), we, at Oodles, are adept in applying both black-box and white-box techniques for software testing. Our AI team undertakes a step-by-step approach to using the black-box testing technique for efficiently mapping-

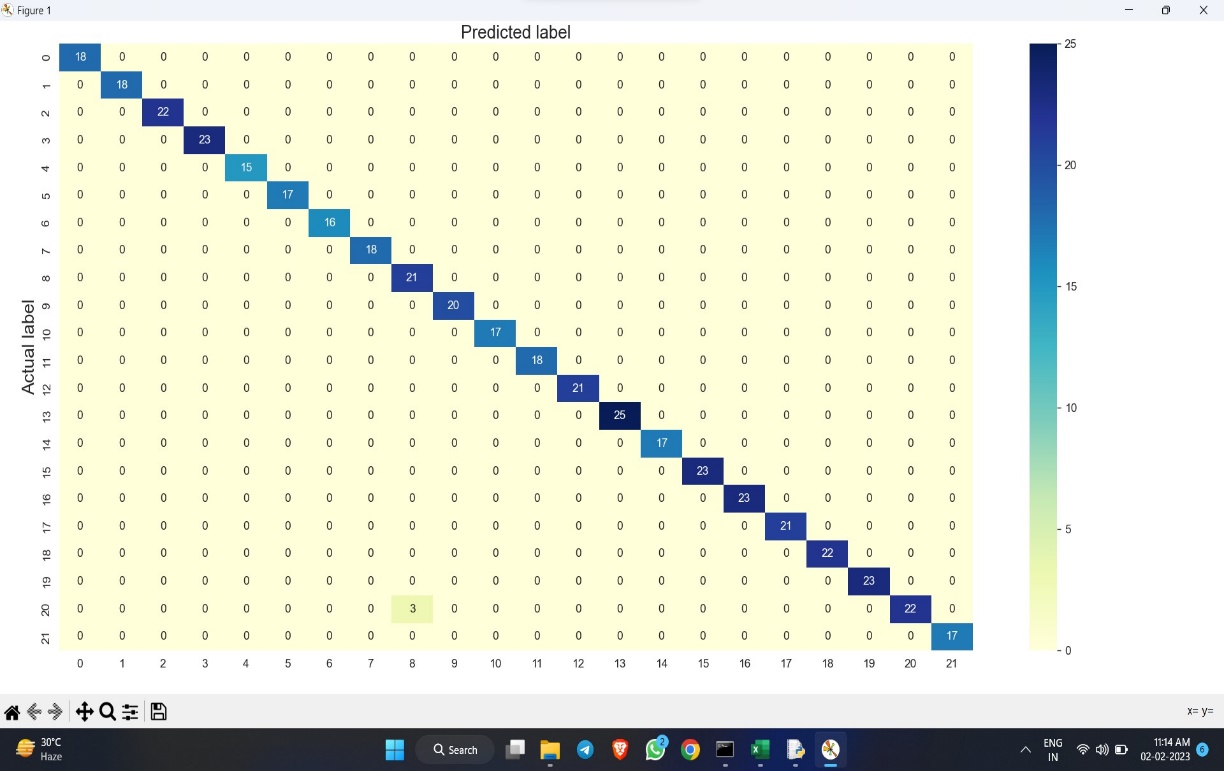
* Complex interface errors
* Model performance errors
* Incorrect or missing functions
* Database errors, and more

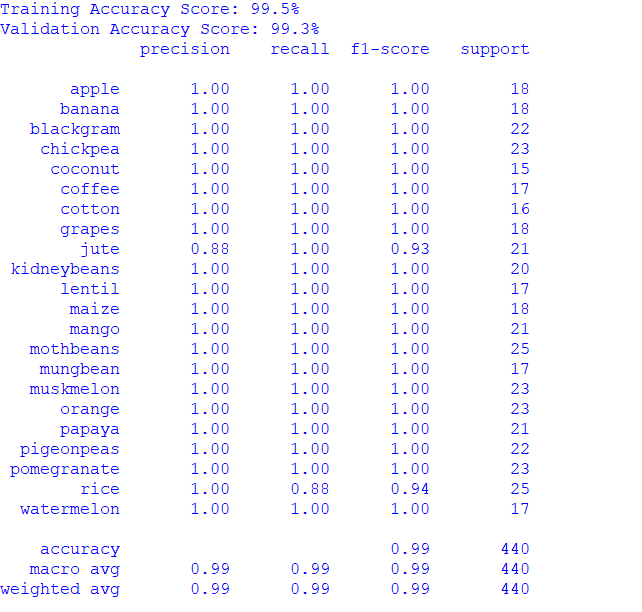
We build robust machine learning models and applications that generate value for businesses while maintaining compliance with industry standards.

**8. Screenshots**





****



**9. Conclusion**

The agriculture sector is a crucial component of many countries' economies and plays a major role in ensuring food security. However, the sector is facing numerous challenges, including changing climate patterns, declining soil fertility, and increasing competition for resources. In this context, it has become imperative to use advanced technologies, such as machine learning, to support the decision-making process of farmers.

One such application is the Agriculture Crop Recommendation System based on Productivity and Season, which aims to provide farmers with personalized recommendations for crops that are best suited to their specific conditions. The system uses a combination of historical data, weather patterns, soil quality, and other relevant factors to generate crop recommendations that are optimized for productivity and profitability.

The system has a number of advantages, including the ability to take into account local conditions and farmers' specific requirements, the ability to incorporate real-time data from various sources, and the ability to generate recommendations that are tailored to the current season. Furthermore, the system can be integrated with other tools, such as precision agriculture and weather forecasting, to provide farmers with an even more comprehensive and accurate picture of the factors that will impact their crop production.

However, there are also some limitations to the system that need to be considered. For example, the quality and accuracy of the data used to generate recommendations is critical, and any errors or inaccuracies in this data could result in incorrect recommendations. Additionally, the system requires a certain level of technical expertise to implement and maintain, which may be a barrier for some farmers.

In conclusion, the Agriculture Crop Recommendation System based on Productivity and Season represents a promising solution to many of the challenges faced by the agriculture sector. By leveraging the power of machine learning and other advanced technologies, the system has the potential to help farmers make more informed and profitable decisions, while also contributing to the broader goal of ensuring food security and sustainable agriculture. However, it is important to carefully consider the limitations and potential risks associated with the system, and to ensure that it is implemented and used in a responsible and effective manner.

**10. References**

[1] Shreya S. Bhanose, Kalyani A. Bogawar (2016) “Crop And Yield Prediction Model”, International Journal of Advance Scientific Research and Engineering Trends, Volume 1,Issue 1, April 2016

[2] Tripathy, A. K., et al.(2011) "Data mining and wireless sensor network for agriculture pest/disease predictions." Information and Communication Technologies (WICT), 2011 World Congress on. IEEE.

[3] Ramesh Babu Palepu (2017) ” An Analysis of Agricultural Soils by using Data Mining Techniques”, International Journal of Engineering Science and Computing, Volume 7 Issue No. 10 October.

[4] Rajeswari and K. Arunesh (2016) “Analysing Soil Data using Data Mining Classification Techniques”, Indian Journal of Science and Technology, Volume 9, May.

[5] A.Swarupa Rani (2017), “The Impact of Data Analytics in Crop Management based on Weather Conditions”, International Journal of Engineering Technology Science and Research, Volume 4,Issue 5,May.

[6] Pritam Bose, Nikola K. Kasabov (2016), “Spiking Neural Networks for Crop Yield Estimation Based on Spatiotemporal Analysis of Image Time Series”, IEEE Transactions On Geoscience And Remote Sensing.

[7] Priyanka P.Chandak (2017),” Smart Farming System Using Data Mining”, International Journal of Applied Engineering Research, Volume 12, Number 11.

[8] Vikas Kumar, Vishal Dave (2013), “KrishiMantra: Agricultural Recommendation System”, Proceedings of the 3rd ACM Symposium on Computing for Development, January.

[9] Savae Latu (2009), ”Sustainable Development : The Role Of Gis And Visualisation”, The Electronic Journal on Information Systems in Developing Countries, EJISDC 38, 5, 1-17.

[10] Nasrin Fathima.G (2014), “Agriculture Crop Pattern Using Data Mining Techniques”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, May.

[11] Ramesh A.Medar (2014), ”A Survey on Data Mining Techniques for Crop Yield Prediction”, International Journal of Advance Research in Computer Science and Management Studies, Volume 2, Issue 9, September.

[12] Shakil Ahamed.A.T.M, Navid Tanzeem Mahmood (2015),” Applying data mining techniques to predict annual yield of major crops and recommend planting different crops in different districts in Bangladesh”, ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD),IEEE,June.

[13] Shreya S.Bhanose (2016),”Crop and Yield Prediction Model”, International Journal of Advence Scientific Research and Engineering Trends, Volume 1,Isssue 1,ISSN(online) 2456- 0774,April.

[14] Agaj i Iorshase, Onyeke Idoko Charles,”A Well-Built Hybrid Recommender System for Agricultural Products in Benue State of Nigeria”, Journal of Software Engineering and Applications,2015,8,581-589

[15] G. Adomavicius and A. Tuzhilin(2005), “Toward the Next Generation of Recommender Systems: A Survey of the State-of-theArt and Possible Extensions,” IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June.

[16] Avinash Jain, Kiran Kumar (2016),”Application of Recommendation Engines in Agriculture”, International Journal of Recent Trends in Engineering & Research, ISSN: 2455-1457.

[17] Kiran Shinde (2015),”Web Based Recommendation System for farmers”, International Journal on Recent and Innovation Trends in Computing and Communication, Volume 3,Issue 3, ISSN:2321- 8169,March.

[18] Konstantinos G. Liakos, “ Machine Learning in Agriculture: A Review”, Sensors 2018, 18, 2674; doi:10.3390/s18082674

[19] Vaishnavi, M.Shobana, N Geethanjali, Dr.S.Karthik, “Data Mining: Solving the Thirst of Recommendations to Users”, IOSR Journal of Computer Engineering (IOSR-JCE), Vol.16, no.6, 2014.

opinions, and subjective text. The demand of sentiment analy-

sis is raised due to the requirement of analyzing and structuring

hidden information, extracted from social media in form of un-

structured data. The sentiment analysis is being implementing

through deep learning techniques. Deep learning consists of

numerous effective and popular models, these models are used

to solve the variety of problems effectively. Different studies

have been discussed in this review to provide a deep knowledge

of the successful growing of deep learning applications in

the ﬁeld of sentiment analysis. Numerous problems have been

resolved by having high accuracy of both ﬁelds of sentiment

analysis and deep learning.

10 Muhammad Umer ET AL