AGRICULTURAL CROP DISEASE CLASSIFICATION

Contributors:

Abdul Jaweed

Ashvath Suresh Babu Piriya

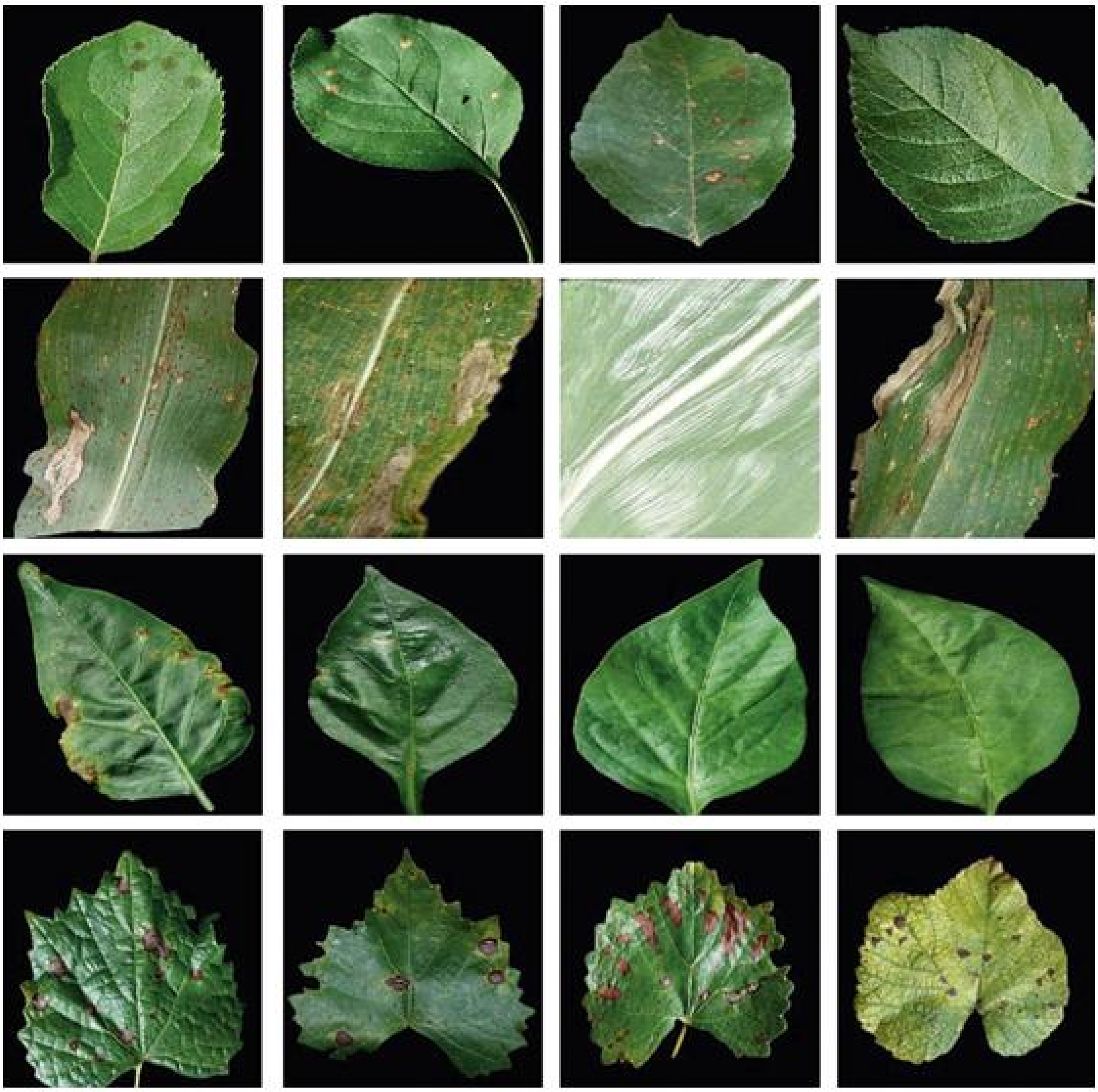
Satwik

Shaleen Mishra

Yashaswini M

Project Link : https://github.com/Abdul-Jaweed/Feynn-Lab-Internship/tree/main/Project%203

WebApp Link : https://agriculture-crop-disease-classification.streamlit.app/



ABSTRACT

Crop diseases pose a significant threat to agricultural productivity and food security worldwide. Timely and accurate identification of these diseases is crucial for effective management and mitigation strategies. In recent years, advancements in machine learning and image processing techniques have enabled the development of automated crop disease classification systems. This abstract presents a comprehensive approach for the classification of agriculture crop diseases, leveraging state-of-the-art methodologies and datasets.

The proposed methodology encompasses a multi-step process, starting with the acquisition of high-resolution images of diseased crops. Image preprocessing techniques are applied to enhance the quality of the images, reducing noise and improving feature extraction. Deep learning architectures, such as convolutional neural networks (CNNs), are employed for effective feature representation and classification.

PROBLEM STATEMENT

Crop diseases have a detrimental impact on agricultural production, causing substantial yield losses and posing a significant threat to global food security. However, accurately diagnosing and managing these diseases is a complex task that often requires expert knowledge and experience. Traditional manual methods for disease identification are time-consuming, subjective, and prone to errors, leading to delayed responses and ineffective management strategies.

There is a pressing need for an automated crop disease classification app that can accurately and efficiently identify diseases in agricultural crops. Such a system would enable timely interventions, including targeted treatments, crop rotation, and disease-resistant crop selection. Additionally, it would facilitate the monitoring and surveillance of disease outbreaks, allowing for effective disease management at both individual and regional scales.

Market/Customer/Business Need Assessment:

The agriculture industry requires a comprehensive crop disease classification app to address the following needs:

- **Timely Disease Identification:** Farmers and stakeholders need an app that can quickly and accurately identify crop diseases, enabling prompt action to reduce losses.

- **Efficient Resource Utilization:** Optimizing the use of resources, such as pesticides and fertilisers, is crucial. Accurate disease identification helps minimize unnecessary resource application.

- **Enhanced Crop Protection:** Reliable tools are needed to protect crops from diseases, prevent yield losses, and reduce disease spread.

- **Precision Agriculture:** Automated disease classification fits into precision farming, enabling real-time monitoring and data-driven decision-making.

- **Increased Efficiency and Productivity:** By automating disease identification, the app saves time, increases efficiency, and reduces labour costs.

- **Knowledge Sharing and Collaboration:** The app facilitates knowledge sharing among farmers, researchers, and extension services, fostering effective disease management strategies.

- **Accessible and User-Friendly Interface:** The app should have a user-friendly interface for easy interaction, allowing farmers to upload images, receive instant diagnosis, and access disease management information.

Addressing these needs with an advanced crop disease classification app benefits farmers, promotes sustainability, and enhances agricultural productivity.

Target Specifications and Characterization:

The target specifications and characterization for the crop disease classification app are as follows, focusing on the customer characteristics:

- **Farmers and Agricultural Stakeholders:** The primary customers of the app are farmers and agricultural stakeholders involved in crop production and management. They may have varying levels of technical expertise but require a user-friendly interface to interact with the app effectively.

- **Accessibility:** The system should be accessible to farmers across different regions and farm sizes. It should cater to various agricultural practices, crop types, and geographical variations.

- **Real-time Analysis:** Customers expect real-time analysis and diagnosis of crop diseases. The system should provide quick results to enable timely decision-making and prompt implementation of disease management strategies.

- **Accuracy and Reliability:** Customers rely on the app for accurate disease identification. The classification app should exhibit high accuracy rates, minimizing false positives and false negatives, and provide reliable results to instil confidence in its recommendations.

- **Scalability and Generalization:** The app should be scalable to handle a wide range of crop diseases and be adaptable to different crop species and regions. It should generalize well to handle variations in disease symptoms, environmental factors, and growth stages.

- **Integration with Existing Tools and Practices:** The app should integrate with existing farm management tools and practices, allowing seamless incorporation into farmers' workflow. It should complement and enhance their current disease management strategies.

- **Data Privacy and Security:** Customer data privacy and security are paramount. The app should adhere to stringent data protection measures, ensuring that sensitive information remains confidential and secure.

- **Support and Documentation:** Customers require comprehensive support and documentation to assist them in using the app effectively. This includes user guides, tutorials, and responsive customer support channels.

By catering to these target specifications and customer characteristics, the crop disease classification app can effectively meet the needs of farmers and agricultural stakeholders, enabling them to make informed decisions and take proactive measures to combat crop diseases.

External Search (online information sources/references/links)

Kaggle:
https://colab.research.google.com/drive/1o8iLTPXc7e666up5m_a4nScDBHCucyaM?usp=sharing

GitHub:
https://github.com/Abdul-Jaweed/Feynn-Lab-Internship/tree/main/Project%203

Benchmarking Alternate Products

When benchmarking the crop disease classification system against existing products or services, consider the following aspects for comparison:

- **Accuracy and Performance:** Evaluate the system's disease identification accuracy compared to other solutions.

- **Speed and Efficiency:** Assess the system's processing time and real-time analysis capabilities.

- **Dataset and Model Diversity:** Compare the diversity and size of training datasets and the effectiveness of different machine learning models.

- **User-Friendly Interface:** Evaluate the system's ease of use and accessibility for users with varying technical expertise.

- **Integration and Compatibility:** Assess how well the system integrates with existing tools and databases.

- **Support and Documentation:** Examine the level of support and available resources for users.

- **Customer Feedback and Reviews:** Consider feedback from users to gauge satisfaction and identify any challenges.

By considering these aspects, you can evaluate the performance and suitability of the crop disease classification system compared to existing alternatives.

Applicable Regulations (government and environmental regulations imposed by countries)

Applicable regulations related to government and environmental concerns can vary from country to country. It's important to note that regulations are subject to change and can be specific to each jurisdiction. Here are some examples of regulations that countries **commonly impose:**

- **Environmental Protection Regulations:** Governments enact regulations to protect the environment and natural resources. These may include laws related to air and water pollution control, waste management, environmental impact assessments, and conservation of biodiversity and ecosystems.

- **Agricultural Regulations:** Countries often have regulations specific to agriculture to ensure sustainable farming practices, food safety, and animal welfare. These regulations can cover areas such as pesticide usage, soil conservation, irrigation management, livestock farming standards, and genetically modified organisms (GMOs).

- **Health and Safety Regulations:** Governments establish regulations to safeguard the health and safety of workers and the general public. These regulations can encompass occupational health and safety standards, workplace hazard assessments, equipment safety requirements, and guidelines for handling hazardous substances.

- **Trade and Import/Export Regulations:** Countries implement regulations related to trade and import/export activities. These can include customs duties, import/export restrictions, product labelling requirements, quarantine regulations, and compliance with international trade agreements.

- **Intellectual Property Regulations:** Intellectual property laws protect inventions, trademarks, copyrights, and other forms of intellectual property. These regulations vary by country and provide legal frameworks for the protection and enforcement of intellectual property rights.

- **Data Protection and Privacy Regulations:** Governments establish regulations to protect individuals' personal data and privacy. These regulations govern how organisations collect, store, process, and share personal information, including requirements for consent, data breach notification, and individuals' rights over their data.

- It is essential to consult specific regulations applicable to the country or countries in which you operate or intend to offer your product or service. Local regulatory authorities, government agencies, industry associations, and legal professionals can provide the most accurate and up-to-date information on the regulations relevant to your particular industry and geographic location.

Applicable Constraints (need for space, budget, expertise)

Applicable Constraints for an Agriculture Crop Disease Classification App:

- **Space Constraints:** The app may require sufficient storage space for storing the necessary data, images, and models. Consider the limitations of the device's storage capacity and ensure efficient data management within the app.

- **Budget Constraints:** Developing and maintaining an app involves costs related to software development, hosting, updates, and ongoing support. Consider budget limitations and allocate resources accordingly to cover development expenses, server costs, infrastructure, and potential marketing efforts.

- **Expertise:** Developing an Agriculture Crop Disease Classification app requires expertise in software development, machine learning, and agricultural domain knowledge. Ensure that the development team possesses the necessary skills or consider partnering with experts in agriculture or machine learning to bridge any knowledge gaps.

- **Data Availability:** The app's effectiveness relies on access to a diverse and comprehensive dataset of crop disease images. Availability and quality of data can be a constraint, as acquiring and curating such datasets may require significant effort, resources, and partnerships with agricultural organisations or research institutions.

- **Technical Expertise:** Deploying and maintaining the app may require technical expertise in cloud hosting, data management, and server maintenance. Consider the need for individuals with relevant technical skills to ensure smooth operation and timely updates.

- **User Accessibility:** The app should be designed to cater to users with varying levels of technical expertise. Consider the need for a user-friendly interface, clear instructions, and intuitive navigation to ensure accessibility for farmers and other stakeholders who may not have extensive technical knowledge.

- **Connectivity and Infrastructure:** The app's functionality may depend on a reliable internet connection, especially for real-time disease classification or accessing external data sources. Consider the constraints of network connectivity and ensure the app's design accounts for offline functionality or low-bandwidth scenarios.

- **Regulatory Compliance:** Ensure compliance with applicable regulations related to data privacy, data protection, and any agricultural or industry-specific regulations that may govern the collection, storage, and usage of data within the app.

- By considering these constraints during the development and implementation phases, you can effectively manage space, budget, expertise, and other limitations to create a robust and user-friendly Agriculture Crop Disease Classification app.

Concept Generation (process of coming up with Idea)

Concept Generation is the process of generating new ideas or concepts that have the potential to solve a problem, fulfil a need, or create value in a particular domain. Here are some steps to help you in the concept generation process:

- **Identify the Problem or Need:** Start by clearly defining the problem or need that you want to address. This could be a pain point experienced by a specific target audience or an opportunity to improve an existing product or service.

- **Conduct Research:** Gather information about the problem domain, existing solutions, market trends, and customer preferences. This will help you understand the current landscape and identify areas where innovation or improvement is possible.

- **Brainstorming:** Engage in brainstorming sessions or idea generation exercises to generate a wide range of ideas. Encourage creative thinking, free-flowing discussions, and the exploration of different perspectives. Consider involving a diverse group of individuals with varied expertise and backgrounds to bring different viewpoints to the table.

- **Idea Selection:** Review and evaluate the ideas generated during the brainstorming phase. Assess their feasibility, potential impact, alignment with the problem or need, and market viability. Narrow down the ideas to a shortlist of the most promising concepts.

- **Idea Refinement:** Take the shortlisted concepts and further refine them. Consider factors such as technical feasibility, resource requirements, scalability, market demand, and competitive advantage. Iterate on the concepts, combining or modifying them to create stronger and more innovative ideas.

- **Validation and Feedback:** Seek feedback from relevant stakeholders, potential customers, or experts in the field. Present the refined concepts to gather insights, identify potential challenges, and validate their viability. Incorporate feedback to refine the concepts further.

- **Prototyping and Testing:** Develop prototypes or minimum viable products (MVPs) to test and validate the concepts in real-world scenarios. This can help uncover any design flaws, usability issues, or technical limitations that need to be addressed.

- **Iteration and Iterative Development:** Based on the feedback and insights gathered from testing, iterate on the concepts and refine them iteratively. Continuously improve and iterate the ideas to ensure they align with the market needs and user expectations.

Remember that the concept generation process is iterative and may require multiple cycles of refinement, testing, and iteration. Stay open to new insights and ideas that may arise during the process. By following these steps, you can generate innovative and valuable concepts that have the potential to address the identified problem or need effectively.

Concept Development (Brief summary of Product/Service will be developed)

The Agriculture Crop Disease Classification System is a software-based solution designed to accurately identify and classify crop diseases in real-time. It leverages advanced image recognition and machine learning algorithms to analyse images of diseased crops and provide accurate disease identification, enabling timely and effective disease management strategies.

Key Features:

- **Image Recognition:** The system utilises computer vision algorithms to analyse images of crops and identify disease symptoms with high precision.

- **Disease Classification:** It employs machine learning techniques to classify the identified diseases into specific categories, providing farmers with actionable information.

- **Real-time Analysis:** The system offers real-time analysis, allowing farmers to quickly detect and respond to crop diseases, minimising the risk of crop damage or yield loss.

- **Extensive Disease Database:** It maintains a comprehensive database of crop diseases, symptoms, and recommended treatments, providing farmers with a valuable knowledge base for effective disease management.

- **User-Friendly Interface:** The system features an intuitive and user-friendly interface accessible through a mobile or web application, making it easy for farmers to capture and upload images for disease identification.

- **Data-driven Insights:** It generates data-driven insights and reports, offering valuable information on disease prevalence, geographic distribution, and trends to aid in proactive disease prevention and control.

- **Integration Capabilities:** The system can integrate with existing agricultural management systems or databases, allowing seamless data flow and enhancing overall farm management efficiency.

Benefits:

- **Early Disease Detection:** Enables early detection of crop diseases, helping farmers take timely actions to prevent further spread and minimise crop damage.

- **Improved Disease Management:** Provides accurate disease identification and classification, facilitating targeted treatment strategies and reducing reliance on broad-spectrum pesticides.

- **Increased Crop Yield:** By effectively managing crop diseases, the system helps optimise crop health and productivity, leading to higher yields and improved profitability.

- **Knowledge Enhancement:** Offers farmers access to a comprehensive disease database and data-driven insights, empowering them with valuable knowledge for disease prevention and control.

- **Cost and Resource Efficiency:** By precisely identifying diseases and recommending appropriate treatments, the system optimises the use of resources, reducing unnecessary pesticide applications and associated costs.

The Agriculture Crop Disease Classification System aims to empower farmers with a reliable and efficient tool for detecting and managing crop diseases, ultimately enhancing crop health, improving yields, and promoting sustainable agriculture practices.

(Very Imp) Final Product Prototype (abstract) with Schematic Diagram

The final product prototype of the Agriculture Crop Disease Classification App is a comprehensive software solution designed to assist farmers in identifying and managing crop diseases. It combines advanced image recognition algorithms, machine learning techniques, and a user-friendly interface to deliver accurate and real-time disease classification.

- **User Interface:** The prototype features an intuitive and user-friendly interface accessible through a mobile or web application. It allows farmers to capture and upload images of crop diseases for analysis.

- **Image Recognition Module:** The image recognition module processes the uploaded images using computer vision algorithms. It detects and extracts relevant features and disease symptoms from the images.

- **Machine Learning Model:** The machine learning model is trained on a diverse dataset of crop disease images. It uses the extracted features to classify the identified diseases accurately.

- **Disease Classification Database:** The system maintains a comprehensive database of crop diseases, including their symptoms, characteristics, and recommended treatments. It serves as a reference for disease classification and provides relevant information to farmers.

- **Real-time Analysis Engine:** The real-time analysis engine performs the disease classification in real-time. It leverages the trained machine learning model to quickly and accurately identify the crop diseases based on the uploaded images.

- **Data Storage and Analytics:** The prototype includes a data storage component that securely stores the uploaded images, disease classification results, and relevant metadata. It also incorporates analytics capabilities to generate insights on disease prevalence, trends, and geographical distribution.

Step 1: Prototype Selection

Product/Service can be developed in the short term future. (2-3 years)

The Agriculture Crop Disease Classification app demonstrates high feasibility for development in the short term future (2-3 years). Here's a detailed analysis of the feasibility factors:

Technological Feasibility:

Advancements in Technology: The app leverages state-of-the-art technologies, including computer vision algorithms and machine learning frameworks like TensorFlow and Keras. These technologies are well-established and continuously improving, making it feasible to develop the app within the given timeframe.

Availability of Datasets: Diverse and comprehensive datasets of crop diseases are available on platforms like Kaggle, facilitating the training of machine learning models. Access to real-world data is crucial for the app's accuracy, and the availability of such datasets ensures the feasibility of model training.

Cloud Infrastructure: Cloud service providers like AWS offer scalable and cost-effective cloud infrastructure. Hosting the app on cloud platforms allows for real-time analysis, data storage, and seamless scalability as the user base grows.

Domain Expertise and Resources:

Developer Expertise: As the developer of the app, you possess the necessary skills in software development, machine learning, and agricultural domain knowledge. Your expertise is critical for building the app effectively within the given timeframe.

Partnerships: Collaborating with agricultural organisations and research institutions can provide access to real-world data and domain expertise. These partnerships can accelerate the development process and enhance the app's accuracy and relevance.

Market Demand:

Agricultural Industry Needs: Crop diseases pose a significant threat to agricultural productivity and food security, making the app's disease classification capabilities highly relevant. Farmers and agricultural stakeholders require timely and accurate disease identification to minimise losses and improve crop health.

Rising Precision Agriculture: The trend towards precision agriculture, driven by data-driven decision-making and real-time monitoring, further emphasises the market demand for the app's capabilities.

Timeframe for Development:

Given your skills, expertise, and access to resources, a development timeframe of 2-3 years is reasonable for building the Agriculture Crop Disease Classification app. With a well-planned development roadmap and efficient implementation, you can create a functional prototype and potentially a fully operational app within this timeframe.

Cost Considerations:

Cost-Effective Development: Utilizing open-source tools, cloud infrastructure, and existing datasets can help reduce development costs. Strategic budget allocation can support necessary expenditures for app development and cloud services.

Regulatory Compliance:

As the app primarily focuses on crop disease classification and data analytics, it is not subject to strict regulatory concerns like medical or financial applications. However, data privacy and protection regulations must be followed to ensure the secure handling of user data.

Conclusion:

The feasibility analysis demonstrates that the Agriculture Crop Disease Classification app is highly feasible for development in the short term future (2-3 years). The app capitalizes on advanced technologies, existing datasets, domain expertise, and market demand to address the critical challenge of crop disease identification. With your expertise as the developer and potential partnerships with agricultural organisations, we can create a valuable solution that benefits farmers and stakeholders in the agricultural industry.

Viability: Product/Service should be relevant or able to survive in the long term future. (20-30 years)

The Agriculture Crop Disease Classification app exhibits strong viability and relevance for long-term survival (20-30 years). Here's a detailed analysis of its viability factors:

Persistent Agricultural Challenges:

Crop Disease Prevalence: Crop diseases have been and will continue to be a persistent challenge in agriculture. As global agriculture faces changing climate patterns and evolving pathogens, the need for accurate disease identification remains constant.

Long-Term Market Demand: The demand for effective disease management solutions is expected to increase with growing population and food requirements. The app's ability to address crop disease challenges positions it as a relevant and valuable tool for farmers and stakeholders over the long term.

Technological Adaptation:

Continuous Technological Advancements: The app's reliance on machine learning and image recognition technologies allows it to adapt to future advancements. As AI and machine learning technologies evolve, the app can incorporate more sophisticated algorithms and techniques to improve disease identification accuracy.

Integration with Emerging Technologies: The app's architecture can be designed to integrate with emerging technologies such as drones or IoT devices for data collection and real-time monitoring. By embracing new technologies, the app can remain relevant and competitive in the agricultural industry.

Data-Driven Insights:

Growing Database: Over time, the app accumulates a vast database of disease-related information, trends, and insights. This data can be invaluable for agricultural research, policymaking, and industry analysis, further enhancing the app's relevance and viability.

Predictive Analytics: With accumulated data, the app can potentially evolve into offering predictive analytics, enabling farmers to anticipate disease outbreaks and plan preventive measures proactively.

Precision Agriculture and Sustainable Practices:

Continued Focus on Precision Agriculture: The trend towards precision agriculture, driven by data-driven decision-making and automation, aligns well with the app's capabilities. As precision agriculture continues to gain prominence, the app's relevance and adoption are likely to increase.

Sustainable Farming Practices: The app's ability to optimize resource utilisation, reduce pesticide usage, and enhance crop protection aligns with the push for sustainable farming practices. As sustainability becomes a more critical concern in agriculture, the app's value proposition strengthens.

User Engagement and Updates:

User-Centric Approach: A focus on user feedback, needs, and preferences allows the app to evolve in line with changing user requirements, ensuring sustained engagement and loyalty.

Regular Updates and Improvements: With ongoing updates and enhancements, the app can adapt to emerging agricultural challenges and technology trends, ensuring its continued relevance in the long term.

Industry Collaboration:

Partnerships with Agricultural Stakeholders: Collaborating with agricultural organisations, research institutions, and extension services fosters industry adoption, increases credibility, and enhances the app's long-term viability.

Brand Recognition:

Established Brand Identity: Over time, the app can develop a strong brand identity as a reliable and efficient crop disease management solution. Positive user experiences and word-of-mouth referrals contribute to sustained brand recognition and loyalty.

Conclusion:

The Agriculture Crop Disease Classification app demonstrates robust viability and relevance for the long term future (20-30 years). Its focus on addressing persistent agricultural challenges, adaptability to technological advancements, data-driven insights, and alignment with precision agriculture and sustainability trends position it as a valuable solution for farmers and stakeholders.

By embracing ongoing improvements, user engagement, industry collaborations, and strategic partnerships, the app can maintain its position as a leading crop disease management tool, contributing to enhanced agricultural productivity and food security for years to come.

Monetization: Product/Service should be monetizable directly. (indirectly monetizable Product/Service should be dropped for this Project)

The Agriculture Crop Disease Classification app is directly monetizable, meaning it can generate revenue directly from its users. Here's a comprehensive analysis of the monetization strategies:

Freemium Model:

Description: Offer the core features of the app, such as disease identification and basic disease information, for free to attract a large user base.

Monetization: Provide advanced features and premium services, such as in-depth disease analysis, personalized recommendations, and access to the latest research on crop diseases, through a subscription-based model.

Benefits: The freemium model allows users to experience the app's capabilities before committing to a subscription. It enables a large user acquisition and provides opportunities to convert free users into paying subscribers.

Subscription Plans:

Description: Offer different subscription plans to cater to various user needs and budgets.

Monetization: Charge users a recurring subscription fee based on the chosen plan, offering tiered access to features and support.

Benefits: Subscription plans allow for a predictable and steady revenue stream. Offering multiple plans gives users the flexibility to choose the level of service that best fits their requirements.

Data Licensing:

Description: Utilize the aggregated and anonymized data collected by the app for valuable insights on disease prevalence, geographic distribution, and crop health trends.

Monetization: License the data to agricultural research institutions, agribusinesses, or government agencies for analysis and market research.

Benefits: Data licensing can generate additional revenue streams beyond user subscriptions. The app's database becomes a valuable resource for stakeholders interested in understanding crop disease patterns and trends.

In-App Purchases:
Description: Provide users with additional tools, resources, or premium content available for purchase within the app.

Monetization: Charge users for access to specialized disease prevention tips, educational materials, real-time disease tracking, or exclusive features.

Benefits: In-app purchases allow users to customize their experience by paying for specific resources or features of interest. This strategy can cater to different user preferences and further enhance revenue generation.

Partnerships:

Description: Collaborate with agribusinesses, seed companies, or agricultural input suppliers to offer special promotions or discounts to users through the app.

Monetization: Earn revenue through affiliate marketing by receiving a commission for driving users to purchase products or services from partner companies.

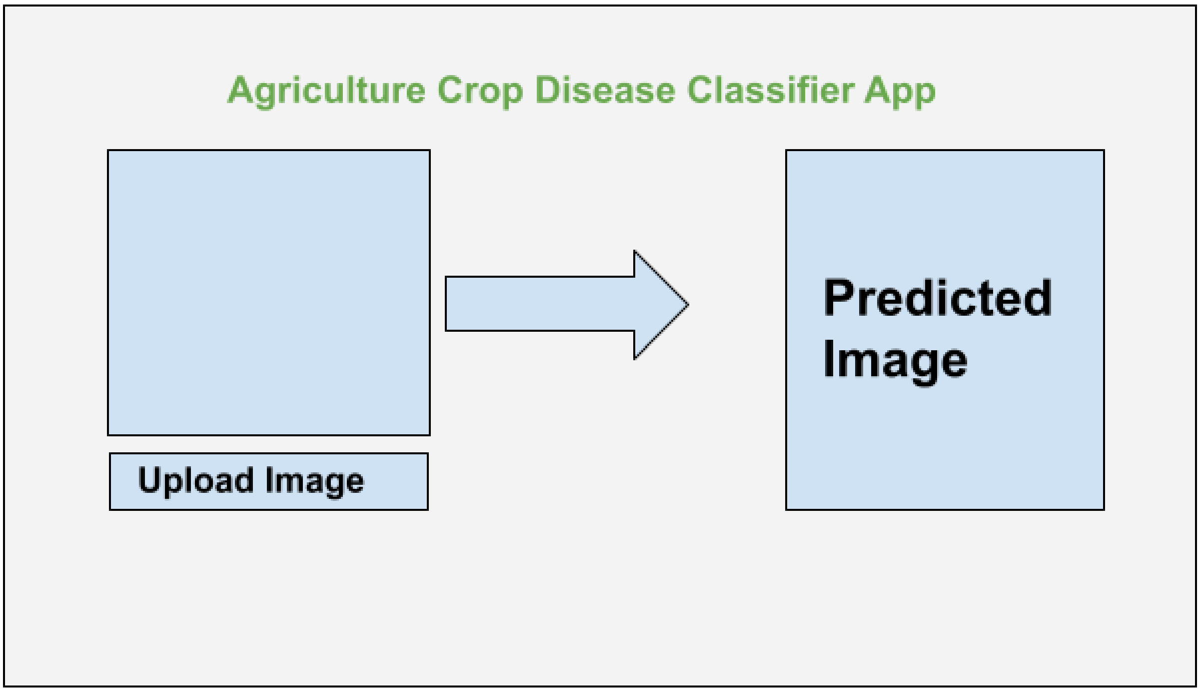
Benefits: Partnerships with industry players enhance the app's value proposition by offering users access to exclusive deals and discounts. Additionally, this strategy creates an additional revenue stream through affiliate commissions.

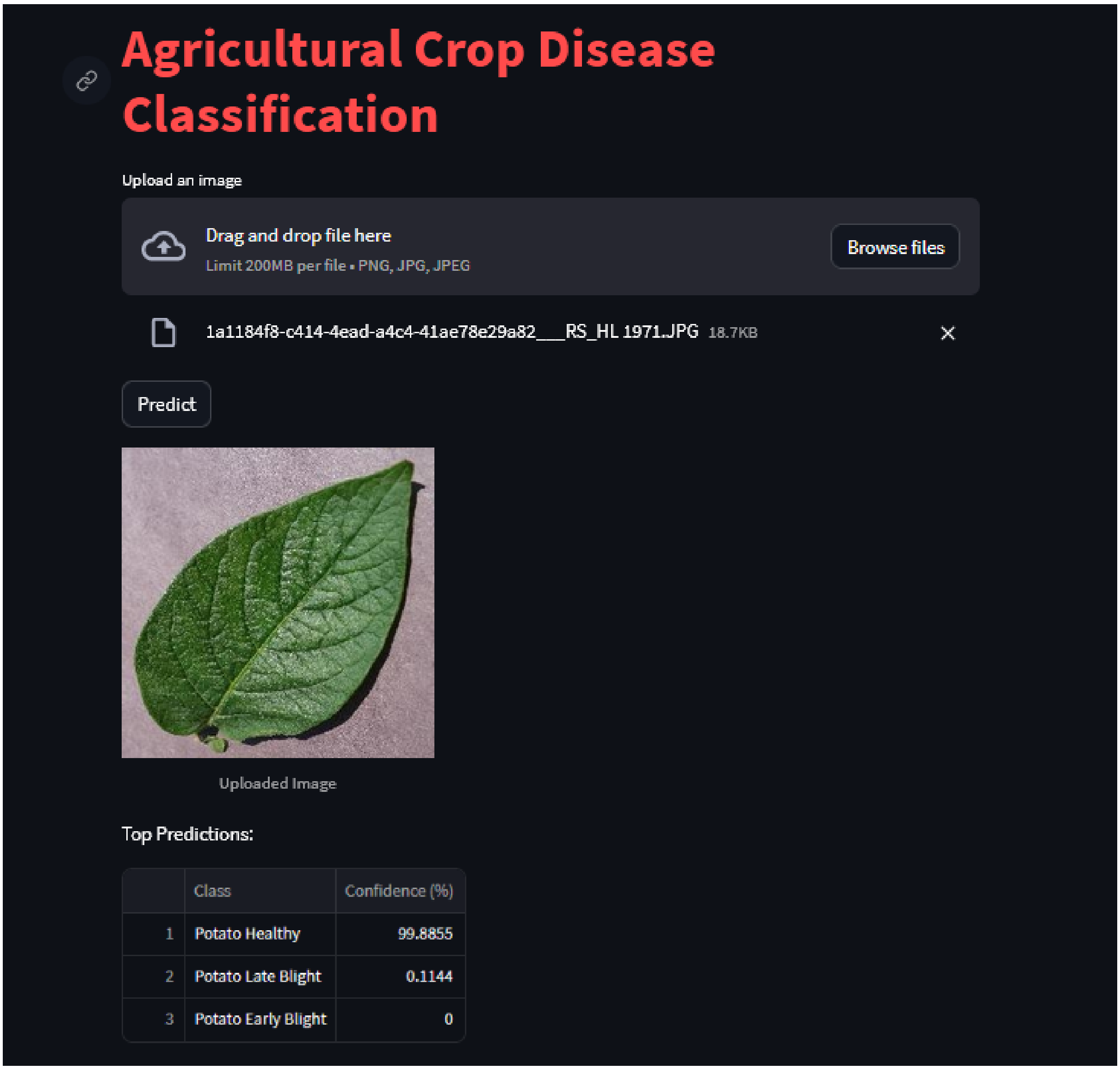
Conclusion:

The Agriculture Crop Disease Classification app is directly monetizable through a combination of revenue streams, including subscription plans, data licensing, in-app purchases, and partnerships. By offering valuable disease management services and data insights, the app can generate steady revenue while providing essential solutions to farmers and stakeholders in the agricultural industry. This direct monetization approach ensures the app's financial sustainability and growth potential for the long term.

Step 2: Prototype Development

PROTOTYPE OF WEB APP





Product details

- **How does it work?**
- **Data Sources**
- **Algorithms, frameworks, software etc. needed**
- **Team required to develop.**
- **What does it cost? etc**

How does it work?

The Agriculture Crop Disease Classification System works by analysing images of crops to identify and classify diseases. Here is a general overview of its functioning:

- **Image Capture:** Users capture images of crops using a mobile device or camera.
- **Image Processing:** The captured images are processed using computer vision algorithms to extract relevant features and disease symptoms.
- **Disease Classification:** The system employs machine learning algorithms to classify the diseases based on the extracted features and symptoms.
- **Results and Recommendations:** The system provides users with the identified disease, along with recommended treatments and management strategies.
- **Real-time Analysis:** The entire process is performed in real-time, enabling farmers to take immediate action to prevent further spread of diseases.

Data Sources:

Data sources for the Agriculture Crop Disease Classification System can include Kaggle, a popular platform for sharing and accessing datasets. Kaggle offers a wide range of datasets related to various domains, including agriculture and crop diseases. You can explore Kaggle's datasets related to crop diseases, plant pathology, or agricultural images, which can serve as valuable resources for training and evaluating the machine learning models used in the system.

Some specific datasets available on Kaggle may include annotated images of diseased crops, labelled with corresponding disease categories. These datasets can be used to train the machine learning algorithms to accurately classify different crop diseases.

Algorithms, Frameworks, Software, etc. needed:

To develop the Agriculture Crop Disease Classification App, the following components, algorithms, frameworks, and software may be required:
Computer Vision Algorithms: Algorithms for image processing, feature extraction, and pattern recognition.

- **Deep Learning Algorithms:** Classification algorithms such as Convolutional Neural Networks (CNNs).

- **Deep Learning Frameworks:** Frameworks TensorFlow, Keras for training and deploying the machine learning models.

- **Image Processing Libraries:** OpenCV or similar libraries for image manipulation, feature extraction, and pre-processing.

- **Web Application Development Tools:** The user interface, relevant development tools such as HTML, CSS and JavaScript

Team Required to Develop:

As the sole developer of the Agriculture Crop Disease Classification app, I will be responsible for managing the project, designing and developing the software, implementing machine learning algorithms, creating the user interface, handling data management, conducting testing, and deploying the app. This requires expertise in software development, machine learning, UI design, data management, and testing. While it is challenging, it is possible to develop the app independently with proper planning and utilising available online resources.

Cost:

- **Computing Resources:** AWS provides a range of compute services, such as EC2 instances or ECR, which you can use for hosting and running your application. The cost will depend on the instance type, storage requirements, and the duration of usage.

- **Data Storage:** app requires storing and accessing data in AWS, you may incur costs for services like Amazon S3 (Simple Storage Service) or Amazon RDS (Relational Database Service). The cost will depend on the amount of data stored, data transfer, and any additional services utilized.

- **Bandwidth and Data Transfer:** AWS charges for data transfer both in and out of their services. If your app involves transmitting a large amount of data, such as images or model predictions, consider the associated bandwidth and data transfer costs.

- **Monitoring and Management:** AWS provides tools for monitoring and managing your deployed application, such as Amazon CloudWatch. While these tools offer valuable insights, they may involve additional costs depending on the level of monitoring and management required.

To estimate the specific costs, you can utilize the AWS Pricing Calculator, which allows you to input your usage details and provides an estimation of the associated expenses. It's important to carefully consider and optimize our AWS resource usage to minimize costs while meeting our app's requirements.

Code Implementation

Initial EDA

```
[4] import os
    crop = os.listdir(train_dir)
```

```
[5] crop
```

```
['Wheat___Healthy',
 'Wheat___Yellow_Rust',
 'Corn___Gray_Leaf_Spot',
 'Corn___Northern_Leaf_Blight',
 'Potato___Early_Blight',
 'Corn___Common_Rust',
 'Rice___Healthy',
 'Wheat___Brown_Rust',
 'Potato___Healthy',
 'Potato___Late_Blight',
 'Rice___Neck_Blast',
 'Rice___Brown_Spot',
 'Rice___Leaf_Blast',
 'Corn___Healthy']
```

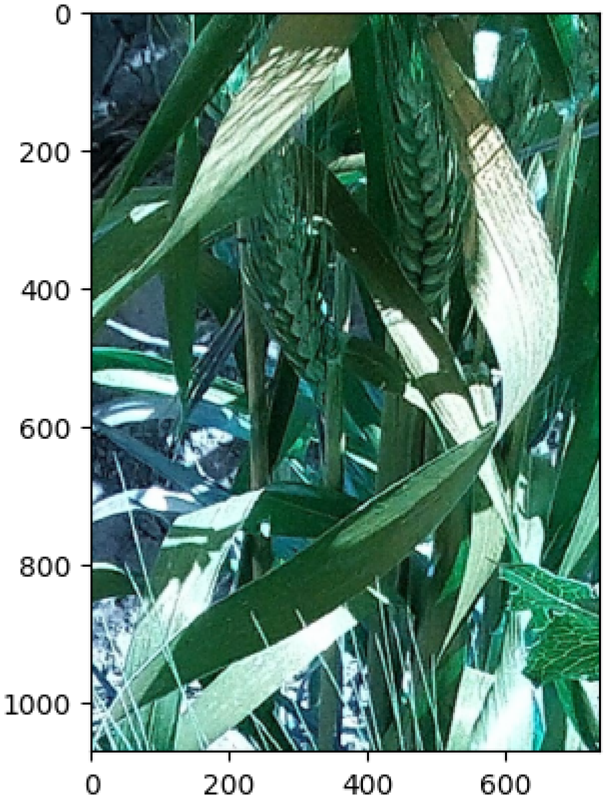
```
[6] train = os.listdir(data_dir+'/train')
    key_value = dict()
    for t in range(len(train)):
        key_value[train[t]] = os.listdir(data_dir+'/train/'+train[t])
```

```
[7] df = []
    for i in range(len(train)):
        for j in range(len(key_value[train[i]])):
            df.append([data_dir+'/train/'+train[i]+'/'+key_value[train[i]][j],train[i]])
```

Image Visualisation

```
import matplotlib.pyplot as plt

plt.imshow(cv.imread(df[0][0]))
```

```
<matplotlib.image.AxesImage at 0x7fbef64539d0>
```



Model Building

```
[24] from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Conv2D
     from tensorflow.keras.layers import MaxPool2D
     from tensorflow.keras.layers import Flatten
     from tensorflow.keras.layers import Dropout
     from tensorflow.keras.layers import Dense
```

```
[27] model = Sequential()
     model.add(Conv2D(128,(5,5),activation='relu',input_shape=(64,64,1)))
     model.add(MaxPool2D(2,2))
     model.add(Conv2D(256,(3,3),activation='relu',input_shape=(64,64,1)))
     model.add(MaxPool2D(2,2))
     model.add(Conv2D(256,(3,3),activation='relu',input_shape=(64,64,1)))
     model.add(MaxPool2D(2,2))
     model.add(Flatten())
     model.add(Dense(512,activation='relu'))
     model.add(Dense(256,activation='relu'))
     model.add(Dense(128,activation='relu'))
     model.add(Dense(64,activation='relu'))
     model.add(Dense(32,activation='relu'))
     model.add(Dense(14,activation='softmax'))
     model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
     model.fit(train_img,y_test,epochs=50,validation_data = (test_img,test_label))
```

```
Epoch 1/50
69/69 [==============================] - 4s 21ms/step - loss: 2.8375 - accuracy: 0.2117 - val_loss: 1.9168 - val_accuracy: 0.3539
Epoch 2/50
69/69 [==============================] - 1s 20ms/step - loss: 1.6272 - accuracy: 0.4248 - val_loss: 1.3904 - val_accuracy: 0.4973
Epoch 3/50
69/69 [==============================] - 1s 20ms/step - loss: 1.2426 - accuracy: 0.5338 - val_loss: 1.1404 - val_accuracy: 0.5372
Epoch 4/50
69/69 [==============================] - 1s 19ms/step - loss: 1.1478 - accuracy: 0.5443 - val_loss: 1.0886 - val_accuracy: 0.5481
Epoch 5/50
```

Prediction

```
[32] print('predicted label is ;',predicted_labels[67])
     print('Actual Label is :',actual_labels[67])
```

```
predicted label is ; Wheat___Brown_Rust
Actual Label is : Wheat___Brown_Rust
```

```
[33] predicted_labels = le.inverse_transform(model.predict(test_img).argmax(axis=1))
     actual_labels = le.inverse_transform(test_label)
```

```
18/18 [==============================] - 0s 7ms/step
```

```
[35] print('predicted label is : ',predicted_labels[102])
     print('Actual Label is : ',actual_labels[102])
```

```
predicted label is :  Rice___Leaf_Blast
Actual Label is :  Rice___Leaf_Blast
```

Github link to the code implementation

https://github.com/Abdul-Jaweed/Feynn-Lab-Internship/tree/main/Project%203/Notebook

Conclusion

In conclusion, developing the Agriculture Crop Disease Classification app as the sole developer requires taking on multiple roles and responsibilities. I would be responsible for project management, software development, machine learning implementation, UI design, data management, testing, and deployment. While it can be challenging, it is possible to develop the app independently with proper planning and utilizing available online resources.

The cost of building the app will depend on factors such as development tools and software, cloud services or infrastructure, data acquisition, training and learning resources, and marketing and promotion. Utilizing open-source tools can help reduce costs, but there may still be expenses associated with AWS deployment, including computing resources, data storage, bandwidth and data transfer, and AWS AI/ML services. It is important to carefully consider the budget and optimize resource usage to minimize costs.

Overall, building the Agriculture Crop Disease Classification app independently allows me to have full control over the development process and can be a rewarding experience. With dedication, expertise, and proper planning, I can create a valuable solution for crop disease identification and classification.

Step-3: Business Modelling

One possible business model for an Agriculture Crop Disease Classification model could be a Software-as-a-Service (SaaS) model. Here is a breakdown of the components of this business model:

Value Proposition:

Accurate Crop Disease Detection: Offer a reliable and efficient AI-powered solution that can accurately detect crop diseases early on, enabling farmers to take timely action to prevent the spread and minimise crop damage.

Increased Crop Yield: Help farmers increase their crop yield by providing insights and recommendations for disease prevention and management.

Customer Segments:

Farmers: Target individual farmers or agricultural organisations looking for advanced **tools to enhance crop disease management practices.**

Agricultural Extension Services: Partner with agricultural extension services or government agencies that support and guide farmers.

Key Activities:

AI Algorithm Development: Continuously improve the AI algorithms used for crop disease detection to ensure accuracy and efficiency.

Data Collection and Analysis: Gather relevant data on crop diseases, including images, weather conditions, and historical information, to train and optimise the AI models.

User Interface and Experience: Develop an intuitive and user-friendly interface that allows farmers to easily capture and upload images of diseased crops and receive actionable insights.

Key Resources:

AI Expertise: Employ a team of data scientists and AI experts to develop and improve crop disease detection algorithms.

Data Infrastructure: Establish a robust infrastructure to collect, store, and analyse the necessary data for training and optimising the AI models.

Partnerships: Collaborate with agricultural research institutions or universities to access domain expertise and expand the knowledge base.

Revenue Streams:

Subscription Fees: Charge farmers and agricultural organisations a monthly or annual subscription fee to access the crop disease detection service.

Additional Services: For an additional fee, offer premium services such as personalised consultations or on-site visits.

Channels:

Online Platform: Provide a web or mobile-based platform where farmers can access the crop disease detection service, upload images, and receive recommendations.

Partnerships: Collaborate with agricultural input suppliers, farming communities, or agricultural extension services to promote and distribute the service.

Cost Structure:

Research and Development: Allocate resources for continuously improving AI algorithms and keeping up with the latest advancements in crop disease detection.

Infrastructure: Maintain the necessary hardware, software, and data storage infrastructure.

Marketing and Sales: Invest in marketing efforts to reach the target audience and acquire customers.

Support and Maintenance: Provide customer support and regular software platform maintenance.

This business model leverages AI technology to provide a valuable service to farmers, helping them detect and manage crop diseases effectively. The company can generate recurring revenue by adopting a SaaS model while continuously improving the software and expanding its customer base.

Step-4: Financial Modelling (equation) with Machine Learning & Data Analysis

a. Identify which Market the product/service will be launched into:

✔ Launching an agricultural crop disease prediction product/service can be beneficial in various markets. Here are some potential markets where our product/service could be applicable:

1. **Agriculture Industry**:

- The primary market for crop disease prediction product/service is the agriculture industry.
- Farmers, agricultural consultants, and large-scale agricultural operations can benefit from accurate disease prediction to protect their crops and optimize their farming practices.
- Providing a reliable solution for crop disease detection and prevention can help improve crop yields, reduce losses, and ensure sustainable agriculture.

2. AgTech Sector:

- AgTech(Agricultural Technologies) companies and startups are constantly developing solutions to improve farming efficiency, crop health, and resource management.
- Crop disease prediction product/service aligns well with the AgTech sector, as it offers advanced technology-driven tools for disease detection and management.

3. Farm Management Software:

- Many farms and agricultural businesses use farm management software to monitor and optimize their operations.
- Integrating crop disease prediction capabilities into existing farm management software systems can enhance their functionality and provide a comprehensive solution for farmers.

4. Precision Agriculture:

- Precision agriculture involves using data-driven technologies to make precise decisions about agricultural practices.
- Crop disease prediction products/services can be integrated into precision agriculture systems, such as remote sensing, drones, and IoT (Internet of Things) devices.
- By combining disease prediction capabilities with these technologies, farmers can identify disease outbreaks early, target interventions, and reduce the use of pesticides and resources.

5. Government Agricultural Departments:

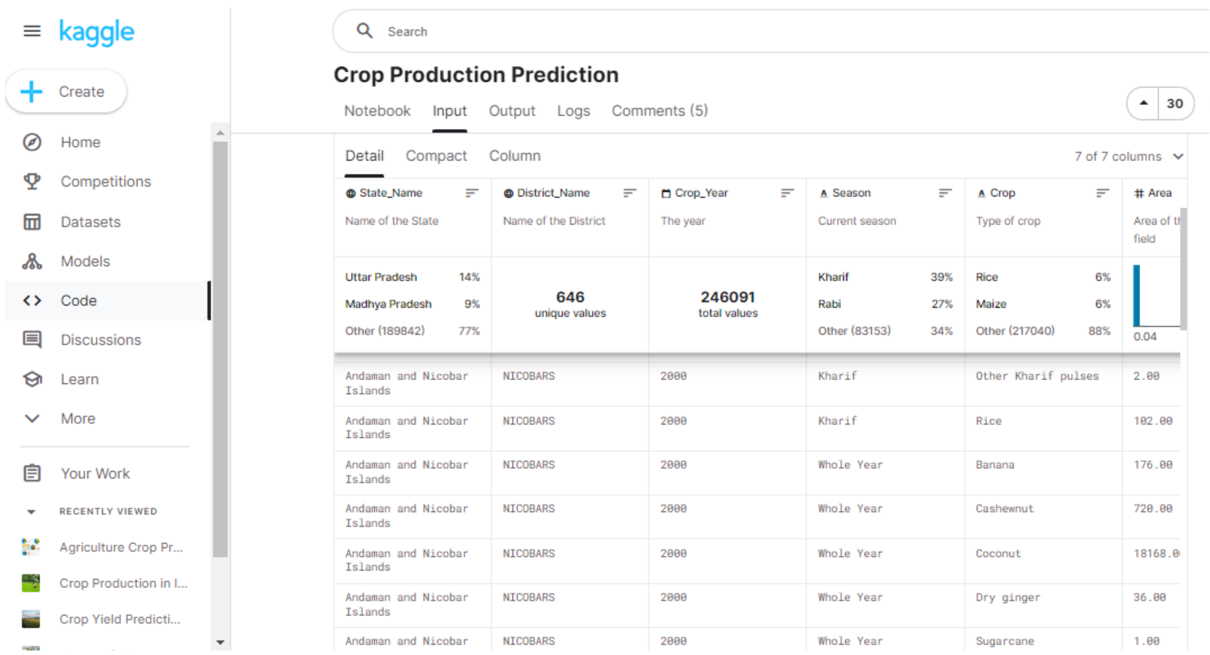
- Government agricultural departments and agencies play a crucial role in providing support and resources to farmers.
- Crop disease prediction products/services can be valuable to these departments, enabling them to enhance their advisory services, disease surveillance efforts, and policy-making related to crop diseases.
- Collaborating with government agencies can help scale the adoption of this product and provide wider access to farmers.

✔ Tailoring the marketing strategies, pricing models, and support services to cater to the requirements of the target market will be essential for successful adoption and growth of your agricultural crop disease prediction product/service.

b. Collecting some data /statistics regarding that Market Online:

✔ The best available dataset is collected from Kaggle with Dataset name: "**Crop Production Prediction**" which describes the production of crops based on various Years & State Names .

Dataset Link:
https://www.kaggle.com/code/anandsubbu007/crop-production-prediction/input



c. Perform Forecasts on Market Using Forecasting models

✔ Dataset actually looks like:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0
5	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Coconut	18168.0	65100000.0
6	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Dry ginger	36.0	100.0
7	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1.0	2.0
8	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sweet potato	5.0	15.0
9	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Tapioca	40.0	169.0

✔ We now build the Forecasting Model based on this dataset as follows using the **SARIMA (Seasonal ARIMA) model** for time series forecasting.

```
# Select the relevant columns for modeling
df_model = df[['Crop_Year', 'Production']]

# Convert 'Crop_Year' to datetime type
df_model['Crop_Year'] = pd.to_datetime(df_model['Crop_Year'], format='%Y')

# Set 'Crop_Year' as the index
df_model.set_index('Crop_Year', inplace=True)

# Resample the data to yearly frequency
df_model = df_model.resample('YS').sum()
```

```
train_data = df_model[:-2]
test_data = df_model[-2:]
```

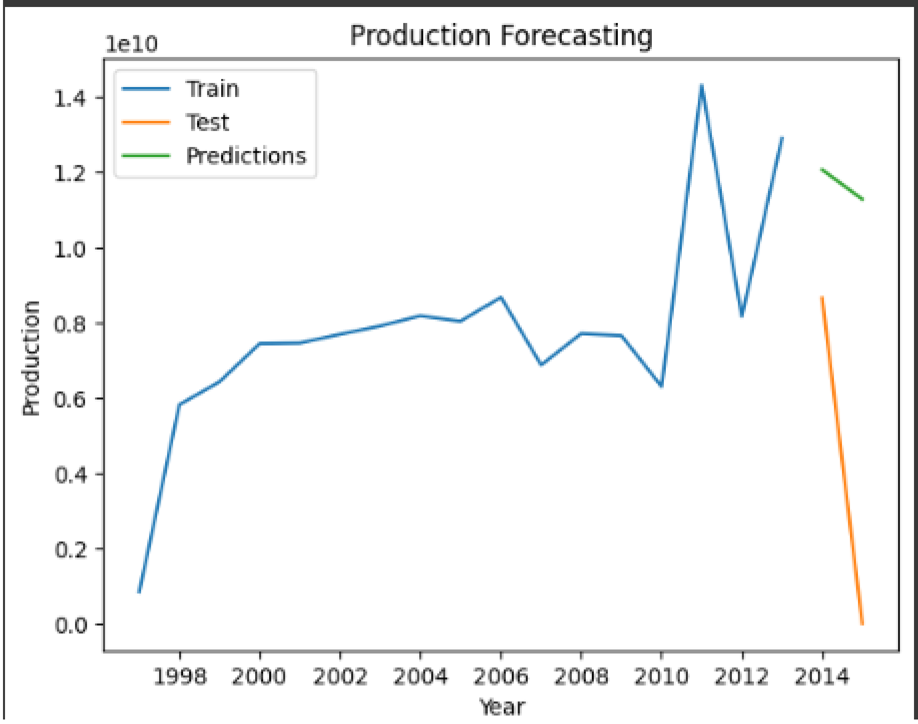
```
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Fit the SARIMA model
order = (1, 0, 0)              # Order of the non-seasonal component (p, d, q)
seasonal_order = (1, 0, 0, 12) # Order of the seasonal component (P, D, Q, s)

sarima_model = SARIMAX(train_data, order=order, seasonal_order=seasonal_order)
sarima_model_fit = sarima_model.fit()
```

```
# Generate predictions
predictions = sarima_model_fit.predict(start=len(train_data), end=len(train_data) + len(test_data) - 1)
```

```
# Calculate Mean Absolute Percentage Error (MAPE)
mape = np.mean(np.abs((test_data['Production'] - predictions) / test_data['Production'])) * 100
print("Mean Absolute Percentage Error (MAPE):", mape)
```

```
Mean Absolute Percentage Error (MAPE): 81280.80605720125
```

```
# Plot actual vs. predicted values
plt.plot(train_data.index, train_data['Production'], label='Train')
plt.plot(test_data.index, test_data['Production'], label='Test')
plt.plot(test_data.index, predictions, label='Predictions')
plt.xlabel('Year')
plt.ylabel('Production')
plt.title('Production Forecasting')
plt.legend()
plt.show()
```



✔ Full code implementation of Forecasting is provided in the link below:

https://colab.research.google.com/drive/1rTxQovW3d32OtltXw_QAWzmnh7ImlTER#scrollTo=WObOmL1dCfIs

d. Financial Equation corresponding to Market Trend:

- ✔ From the Forecasting Model we can observe that the market trend is Linearly varying so we can form the Linear Financial Equation.
- ✔ Financial Equation is therefore given by:

$$Y = a.\ X(t) + b$$

Where,

Y= Profits from the product

a= Price of the product

X= Market i.e. Total sales as a function of time

b= Total Maintenance Cost