Université 1st Hassan
NATIONAL SCHOOL OF APPLIED SCIENCE BERRECHID

Engineering Cycle
Information Systems Engineering and BIG DATA

Project Report

Application Big Data

Subject

INTELLIGENT FARMS MANAGEMENT

Supervised by:

- DR. KARIM Lamia

Realized by:

- MOUSSAOUI Othman
- SAGHIR Aya
- EL KASSIMI Khalid
- LEANDRE Pacis

Academic year: 2023-2024

Contents

List of Figures

Acknowledgment

We would like to express our gratitude to all the people who contributed to the realization of this project, their presence, support, availability, and advice have been invaluable.

Special thanks are extended to our teacher and supervisor, Mrs. Lamia KARIM, for her dedicated supervision of this work. Her insightful feedback, constructive criticism, and unwavering support have been instrumental in shaping both the direction and the outcome of this project. Her expertise and knowledge in the field of Big Data and Information Systems have not only guided us but also inspired us to strive for excellence.

We are also immensely grateful to all the staff and faculty members of the National School of Applied Science Berrechid. Their willingness to share knowledge and resources has played a crucial role in the completion of this project. We appreciate the conducive learning environment they have provided, which has been fundamental in our academic and personal growth.
Our heartfelt appreciation goes to our colleagues and classmates, who provided us with an environment of camaraderie and intellectual stimulation. The discussions, brainstorming sessions, and constant exchange of ideas with them have significantly enriched our experience and understanding.

Finally, our deepest gratitude is reserved for our families and friends. Their unwavering faith in our abilities, endless patience, and moral support have been the pillars of our strength. They have been our constant source of encouragement and motivation, and this achievement is as much theirs as it is ours.

In conclusion, this project is not just a reflection of our efforts, but a testament to the collective support and guidance of each individual mentioned and many unmentioned who have been a part of this journey. We are profoundly thankful to each one of them.

Abstract

In response to the evolving challenges in the agriculture sector, our organization, Pako Grow, embarked on a comprehensive project encompassing diverse aspects of agricultural technology. This multifaceted initiative aimed to leverage advanced technologies to enhance crop management, optimize resource utilization, and provide valuable insights to farmers. The project unfolded into four distinct yet interconnected sub-projects, each addressing a crucial facet of modern agriculture.

- Crop Recommendation System

  The first sub-project focused on developing a robust Crop Recommendation System tailored for different regions. Leveraging machine learning algorithms and historical agricultural data, our system recommends suitable crops based on climate and other relevant factors. This tool serves as a valuable resource for farmers, empowering them with data-driven insights for optimal crop selection. At the end, we have a visualization in a Dashboard.

- Crop Anomaly Detection using Image Analysis

  The second sub-project delved into the realm of computer vision, specifically targeting crop anomaly detection. Utilizing a dataset rich in agricultural images, our system employs deep learning techniques to identify anomalies in crops. By analyzing visual cues, the system distinguishes healthy crops from those exhibiting abnormalities, facilitating early intervention and disease prevention.

- Fertilizer Recommendation System

  The third sub-project addressed the critical aspect of fertilizer management. Leveraging data on soil composition and crop types, our Fertilizer Recommendation System employs machine learning algorithms to suggest optimal fertilizer formulations. This initiative promotes precision agriculture, ensuring that farmers apply fertilizers judiciously, reducing environmental impact and enhancing crop yields.

- Pako GPT - Agriculture Chatbot

  The fourth sub-project introduced Pako GPT, an intelligent and interactive Agriculture Chatbot. Pako GPT is designed to provide farmers with instant information, guidance, and advice on farming practices. Using state-of-the-art natural language processing, Pako GPT understands queries related to agriculture, offers personalized recommendations, and serves as a virtual companion for farmers seeking expert insights.

This comprehensive report outlines the methodologies, technologies, and outcomes of each sub-project. Through the integration of data science, machine learning, computer vision, and natural language processing, Pako aims to contribute to the modernization and sustainability of agriculture, fostering increased efficiency and informed decision-making for farmers worldwide. The success of these projects underscores Pako's commitment to leveraging technology for the betterment of the agricultural landscape

Chapter 1

General Context of the Project

1.1 Introduction

The potential of Machine Learning (ML) and big data in agriculture is incredibly appealing. Agriculture is one of the few remaining industries that has yet to be fully digitized and analyzed. It is a complex system that involves humans, machines, natural systems, chemistry, biology, weather, and climate.

When discussing the application of ML and big data in agriculture, several recurring themes emerge. First, agriculture is highly specific to location. Factors such as soil, water, and land characteristics vary greatly and have a significant impact on outcomes. No two fields or plots are identical. Second, weather and climate are also localized, with each growing season and even within a season varying considerably. Third, the proximity to marketplaces and transportation infrastructure varies greatly from one farming operation to another.

Implementing our solution in smart farms is expected to revolutionize agricultural practices by increasing productivity, reducing resource waste, and improving profitability. Additionally, it will promote sustainable farming by optimizing resource usage and minimizing environmental impact.

That is why we have focused on the following topics:

1. **Crop Anomaly Detection:** Developing ML models to identify anomalies in crops, such as diseases, pests, or irregular growth patterns, by analyzing images, sensor data, and historical patterns.

2. **Crop Recommendation:** Utilizing historical and real-time data to forecast crop yield, taking into account environmental factors, weather patterns, soil health, and past production data.

3. **Chatbot:** An innovative AI-based platform, uses machine learning, and harnesses the power of Large Language Models to offer real-time crop insights to farmers in a customized and friendly way. This solution is tailored to the unique agricultural landscape and challenges of Morocco or Africa.

4. **Fertilizer Recommendation:** The Fertilizer Recommendation System is a vital component of our agricultural technology initiative. It utilizes machine learning algorithms to provide precise fertilizer recommendations to farmers based on soil composition, crop types, and environmental factors. By optimizing fertilizer usage, this sub-project aims to enhance crop yields, reduce environmental impact, and promote sustainable farming practices.

1.2 About US

PAKO Grow, which symbolizes Pacis Khalid Othman Aya in full words, is a forward-thinking company headquartered in Berrechid. It was founded by four dedicated engineers whose vision is to introduce IT solutions to sectors where technology adoption is underdeveloped. Their mission is to empower clients with efficient, cost-saving, and innovative solutions, particularly in the agriculture sector.

PAKO Grow, established in November 2023, is at the forefront of technology-driven solutions. Our primary focus is on providing comprehensive tools and systems for intelligent farm management. We harness the potential of big data, data science, machine learning, and data analytics to transform the agriculture industry.

Figure 1.1: Company Logo

1.2.1 Mission

PAKO Grow's mission is to empower farmers with advanced technological tools and insights to optimize their operations, make data-driven decisions, and ultimately contribute to the global effort of sustainable and intelligent agriculture.

1.2.2 Core Offerings

- Big Data Solutions: PAKO Grow collects and processes vast amounts of agricultural data, including weather patterns, soil conditions, crop health, and machinery performance. The company employs state-of-the-art big data technologies to organize, analyze, and derive actionable insights from this information.

- Data Science Applications: The company utilizes data science techniques to uncover patterns, trends, and correlations within agricultural data. This enables PAKO Grow to provide farmers with valuable predictions and recommendations, ranging from optimal planting times to personalized crop care strategies.

- Machine Learning Algorithms: PAKO Grow integrates machine learning algorithms into its platform to create predictive models for various aspects of farming. These models continuously learn and adapt, offering farmers dynamic solutions for crop management, pest control, and resource allocation.

- Data Analytics Solutions: PAKO Grow's data analytics tools empower farmers to gain a deeper understanding of their operations. By visualizing key performance indicators and trends, farmers can make informed decisions, leading to increased efficiency, reduced costs, and improved overall farm productivity.

1.2.3 Key Features

- Precision Farming: PAKO Grow enables precision farming by providing farmers with real-time insights into crop health, allowing for targeted interventions and optimized resource utilization.

- Resource Optimization: Through data analytics and machine learning, PAKO Grow helps farmers optimize the use of resources such as water, fertilizers, and pesticides, promoting sustainability and environmental stewardship.

- Customized Action Plans: Farmers receive personalized action plans based on the specific conditions of their farms. These plans are continuously updated as new data is collected and analyzed.

- Vision for the Future: PAKO Grow envisions a future where every farmer, regardless of size or location, can access and benefit from advanced technological solutions. By fostering innovation in agriculture, the company aims to contribute to global food security and sustainability.

1.3 Problematic

In the face of climate uncertainties, resource constraints, and a growing global population, Intelligent Farm Management is imperative. Traditional farming faces challenges like unpredictable weather patterns, inefficient resource use, and the need for proactive pest and disease management.

From the standpoint of farmers, the agricultural innovation projects confront intricate challenges deeply rooted in the daily realities of traditional farming. In the domain of Crop Anomaly Detection, the arduous task of visually inspecting expansive fields for signs of diseases or pests poses a substantial burden. The time-intensive nature of these inspections often leads to delayed identification, leaving farmers grappling with the escalating impact of crop issues on their yields and economic livelihoods. This project directly addresses this challenge by leveraging Convolutional Neural Networks to introduce a more efficient and proactive approach, enabling early anomaly detection and empowering farmers to intervene promptly.

In the arena of Crop Recommendation, farmers contend with the relentless uncertainties tied to climate and environmental conditions. Unpredictable weather patterns and soil variations make it challenging for farmers to make well-informed decisions about which crops to plant for optimal yield. The Crop Recommendation initiative directly addresses this practical challenge by incorporating supervised machine learning techniques and considering variables such as temperature and soil pH. By providing farmers with predictive insights, the project aims to empower them to make decisions grounded in data, mitigating risks associated with unpredictable weather and fostering efficient resource use.

Fertilizer Recommendation dives into the perennial struggle farmers face in balancing production with market demand. Overusing fertilizers that negatively impact the environment or falling short of meeting crop nutrient requirements both pose economic and ecological threats to farmers. This project takes a farmer-centric approach by integrating data on crop consumption patterns in Morocco with information on soil composition, crop types, and environmental impact. Through optimization techniques such as linear programming, genetic algorithms, or other methodologies, the project seeks to enhance resource allocation, maximize crop yield, and bring economic and environmental stability by aligning fertilizer application with crop nutrient needs.

Pako GPT, the intelligent Agriculture Chatbot, acts as a virtual agricultural assistant providing instant information, guidance, and advice on farming practices. Using natural language processing, Pako GPT understands queries related to agriculture, offers personalized recommendations, and serves as a reliable source of expert insights.

Thus, to find crucial solutions for these problematics, we need to answer these questions:

1. How do Convolutional Neural Networks in Crop Anomaly Detection reduce the need for time-consuming visual field inspections?
2. In Crop Recommendation, how do supervised machine learning techniques address challenges posed by unpredictable weather patterns and soil variations?
3. Regarding Fertilizer Recommendation, how does integrating data on crop consumption patterns in Morocco with optimization techniques like linear programming or genetic algorithms help farmers balance fertilizer application with crop nutrient needs?
4. How does Pako GPT, the Agriculture Chatbot, contribute to providing real-time information, guidance, and advice for farmers in the field of agriculture?

1.4 Objectives

The main objectives for our problematics are the following:

1. Efficient Anomaly Detection with Convolutional Neural Networks (CNNs):
   - Develop and implement a robust Crop Anomaly Detection system using CNNs to significantly reduce the time and effort required for visual inspections in expansive fields.
   - Achieve early detection of diseases or pests in crops, enabling farmers to intervene promptly and minimize the impact on yields.
2. Data-Driven Crop Recommendation:
   - Implement a Crop Recommendation system utilizing supervised machine learning techniques, considering variables such as temperature, soil pH, and historical climate data.
   - Provide farmers with actionable insights to optimize crop selection, mitigating risks associated with unpredictable weather patterns and promoting efficient resource utilization.
3. Optimized Fertilizer Recommendation:
   - Develop a Fertilizer Recommendation system by integrating data on crop consumption patterns in Morocco with information on soil composition, crop types, and environmental impact.
   - Apply optimization techniques such as linear programming or genetic algorithms to align fertilizer application with crop nutrient needs, enhancing resource allocation and minimizing environmental impact.

4. Pako GPT Agriculture Chatbot:
   - Implement and deploy Pako GPT, an intelligent Agriculture Chatbot, to serve as a virtual agricultural assistant for farmers.
   - Ensure Pako GPT provides real-time information, personalized guidance, and expert advice on farming practices, contributing to increased efficiency and informed decision-making.
5. User Empowerment and Adoption:
   - Conduct user training programs and workshops to empower farmers with the knowledge and skills required to effectively utilize the implemented systems.
   - Monitor user adoption and gather feedback to continuously improve the usability and effectiveness of the developed solutions.
6. Economic and Environmental Impact Assessment:
   - Assess the economic impact of the Crop Anomaly Detection, Crop Recommendation, Fertilizer Recommendation, and Pako GPT initiatives on farmers' yields and livelihoods.
   - Evaluate the environmental sustainability of the Fertilizer Recommendation system, considering factors such as reduced fertilizer usage and minimized ecological impact.

By achieving these objectives, the project aims to address the practical challenges faced by farmers, enhance their decision-making capabilities, and contribute to the economic sustainability and environmental responsibility of agricultural practices.

Chapter 2

Analytical and conceptual studies

2.1 Introduction

In the real of agricultural innovation, the Project and Solution chapter serves as the cornerstone, encapsulating the intricacies and methodologies employed in the pursuit of enhancing farming practices. This section delves into the comprehensive details of our initiative, encompassing project specifics, meticulous planning tools, the architectural framework guiding our endeavors, the rich dataset fueling our models, and the pivotal role of APIs in streamlining connectivity. As we navigate through the project landscape, the narrative unfolds to reveal the thoughtfully crafted models that underpin our solutions.

Within this chapter, the readers embark on a journey through the very essence of our agricultural innovation initiative, traversing the roadmap from conceptualization to implementation. The meticulously outlined project details provide a panoramic view of the scope, objectives, and challenges addressed by our solutions. The strategic planning tools employed act as guiding beacons, steering the project towards success with a well-defined and structured approach.

The architectural framework, a vital scaffolding of our venture, elucidates the design principles and technological foundations that support the seamless integration of diverse components. A deep dive into the dataset employed unfolds the wealth of information driving our models, while the strategic use of APIs ensures the fluid exchange of data and functionalities between various modules.

Central to our narrative are the intricately designed models, the linchpin of our solutions, carefully crafted to meet the unique demands of crop anomaly detection, crop recommendation, fertilizer recommendation, and the interactive Agriculture Chatbot. This chapter serves as the gateway to a nuanced understanding of how technological prowess, strategic planning, and a rich dataset converge to offer innovative solutions, ushering in a new era of intelligent farm management.

As we unfold the layers of the Project and Solution chapter, readers will gain a profound insight into the essence of our agricultural innovation initiative, appreciating the synergistic harmony of technology, planning, and data-driven intelligence that propels our mission forward

2.2 Project details

In the world of agricultural innovation, four pivotal initiatives are poised to revolutionize the manner in which we approach farming practices:

1. **Crop Anomaly Detection:** Our Crop Anomaly Detection project harnesses the capabilities of Convolutional Neural Networks (CNNs) to detect anomalies in crops' health.

2. **Crop Recommendation:** Our Crop Recommendation system is built on the foundation of machine learning algorithms that consider a multitude of factors, including soil conditions, climate, and crop history.

3. **Fertilizer Recommendation:** The Fertilizer Recommendation project offers personalized guidance to farmers by evaluating soil conditions and the specific nutrient requirements of crops.

4. **Pako GPT Agriculture Chatbot:** Our Pako GPT Agriculture Chatbot represents a convergence of natural language processing and agricultural expertise.

As these projects continue to evolve and adapt to the dynamic agricultural landscape, their potential for positive impact only grows. With each passing day, we move closer to a future where farming is not just a livelihood but a thriving and sustainable industry, supported by cutting-edge technology and unwavering dedication to the well-being of farmers.

2.2.1 Crop Anomaly Detection:

Figure 2.1: Anomaly Detection

- **Objective:** Develop a deep learning model, specifically using Convolutional Neural Networks (CNNs), to detect anomalies in crops, including diseases, pests, or other deviations from a healthy state.
- **Approach:** Utilize CNNs for image-based tasks, training the model on a diverse dataset containing images of both healthy and anomalous crops.
- **Data Requirements:** A diverse dataset with labeled images of healthy and anomalous crops is essential for effective model training.
- **Benefits:**
  - Early Detection of Crop Issues
  - Precision Agriculture
  - Increased Yield

2.2.2 Crop Recommendation:

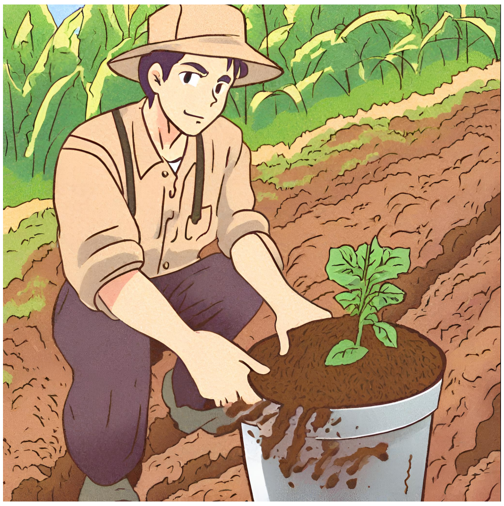

Figure 2.2: Checking Soil

- **Objective:** Develop a data-driven Crop Recommendation system using supervised machine learning techniques, focusing on variables such as temperature, soil pH, and historical climate data.

- **Approach:** Implement supervised machine learning algorithms that analyze historical climate data, soil conditions, and other relevant variables to generate crop recommendations.
- **Data Requirements:** A comprehensive dataset comprising labeled images of crops, climate data, and soil characteristics is crucial for training the Crop Recommendation model.
- **Benefits:**
  - Informed Decision-Making
  - Risk Mitigation
  - Efficient Resource Utilization

2.2.3 Fertilizer Recommendation:

Figure 2.3: Field Fertilizing

- **Objective:** Develop an Optimized Fertilizer Recommendation system by integrating data on crop consumption patterns in Morocco with information on soil composition, crop types, and environmental impact.
- **Approach:** Utilize optimization techniques such as linear programming or genetic algorithms to align fertilizer application with crop nutrient needs.
- **Data Requirements:** An extensive dataset containing information on crop consumption patterns, soil composition, crop types, and environmental impact is crucial for training the Fertilizer Recommendation model.
- **Benefits:**
  - Resource Efficiency
  - Economic Stability
  - Sustainable Agriculture

2.2.4 Pako GPT Chatbot:

PAKO
GPT

Figure 2.4: PAKO GPT Logo

- **Objective:** Implement Pako GPT, an intelligent Agriculture Chatbot, to serve as a virtual agricultural assistant, providing real-time information, personalized guidance, and expert advice on farming practices.
- **Approach:** Utilize natural language processing techniques to enable Pako GPT to understand and respond to queries related to agriculture.
- **Data Requirements:** A diverse dataset comprising agricultural queries, expert responses, and information related to farming practices is essential for training Pako GPT.
- **Benefits:**
  - Instant Information Access
  - Personalized Recommendations
  - Expert Insights

2.3 SWOT analysis of planning tools

Objective: Seeking a project planning tool that provides an approach to project management that emphasizes flexibility, collaboration among team members, and adaptability to change. A tool that allows for Agile work. The tools are: Jira, Asana, Trello, Wrike, Monday.com, Microsoft Project.

2.3.1 Jira:

STRENGTHS

1. Powerful customization options for workflows and issue types.
2. Extensive integration capabilities with other development tools.
3. Well-suited for Agile methodologies.
4. Robust reporting and analysis features.

WEAKNESSES

1. Steeper learning curve, especially for non-technical users.
2. Interface may be overwhelming for simple projects.
3. Requires a certain level of maintenance and administration.

OPPORTUNITIES

1. Continuous improvement through updates and new features.
2. Expansion of integrations to cover a wider range of tools.
3. Addressing non-technical users with a better user experience.

THREATS

1. Competition from emerging tools with simpler interfaces.
2. User resistance to a more technical or complex system.

2.3.2 Asana:

STRENGTHS

1. User-friendly interface, making the tool accessible to all users.
2. Task and project tracking with customizable workflows.
3. Collaboration features, including comments and file sharing.
4. Integrations with popular third-party applications.

WEAKNESSES

1. Limited features for managing complex projects.
2. May not be as suitable for large-scale enterprise projects.
3. Somewhat limited advanced reporting capabilities.

Opportunities

1. Expansion of features for more advanced project management.
2. Integration with additional applications to enhance functionality.
3. Growth of the user base across various industries.

Threats

1. Competition from tools offering more comprehensive solutions.
2. User demand for more advanced project management features.

2.3.3 Trello:

Strengths

1. Simple and intuitive Kanban-style project management.
2. Visual representation of tasks and progress on boards.
3. Flexible and easy-to-use interface.
4. Wide range of integrations with third-party applications.

Weaknesses

1. Limited features for complex project structures.
2. May not be ideal for large-scale projects with multiple teams.
3. Lack of built-in reporting and analysis features.

Opportunities

1. Expansion of features for more complex project management.
2. Collaboration with other tools to enhance functionality.
3. Growth of the user base in creative and collaborative industries.

Threats

1. Competition from tools offering more comprehensive solutions.
2. User demand for more advanced project management features.

2.3.4 Wrike:

Strengths

1. Versatile project management capabilities.
2. Customizable workflows for different project types.
3. Collaboration features, including real-time editing and discussions.
4. Robust reporting and analytics.

Weaknesses

1. Learning curve, especially for beginners.
2. Complexity of features may be overwhelming for simple projects.
3. May require training for optimal use.

Opportunities

1. Continuous improvement through updates and new features.
2. Expansion of integrations to cover a wider range of tools.
3. Targeting a broader audience with a better user experience.

Threats

1. Competition from emerging tools offering simpler interfaces.
2. User resistance to a more complex system.

2.3.5 Microsoft Project:

Strengths

1. Robust project planning and scheduling features.
2. Integration with other tools in the Microsoft Office suite.
3. Comprehensive reporting and analysis capabilities.

Weaknesses

1. Learning curve, especially for beginners.
2. Perceived as complex for smaller projects.
3. Requires a Microsoft ecosystem for full functionality.

Opportunities

1. Continuous improvement through updates and new features.
2. Expansion of integrations to cover a wider range of tools.
3. Addressing a broader audience with a better user experience.

Threats

1. Competition from emerging tools offering simpler interfaces.
2. User resistance to a more complex system.

2.4 Planification

As we saw in the previous sub-chapter, our company chose to use Jira as a tool of planning tasks due to different features that Jira offers and the need of our company of working in agile. Here is the demonstration of how we used Jira to accomplish our work in a defined time.

2.4.1 Team members

To our Jira software, everyone in the team has access to everything in order to see what he has been assigned to do in a given period of time. In our case we have four members as we can see it below.

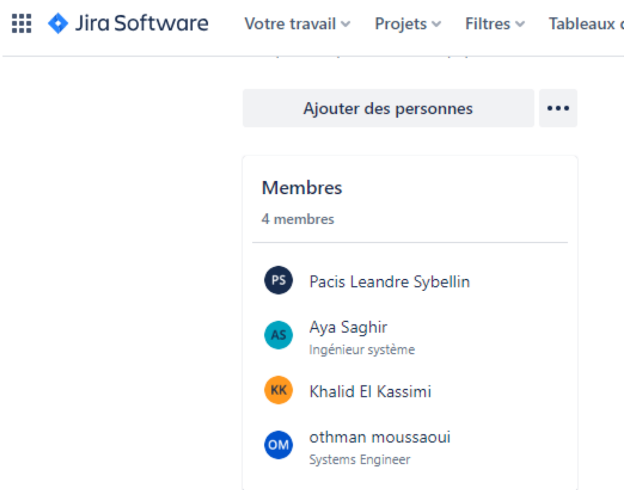

Figure 2.5: Jira team

2.4.2 Planification

In our Board we must set a chronology, which is like a Gantt chart which shows main tasks that need to be done and the time we set for it to be done.

For example here we can see how the chronology was in our case.

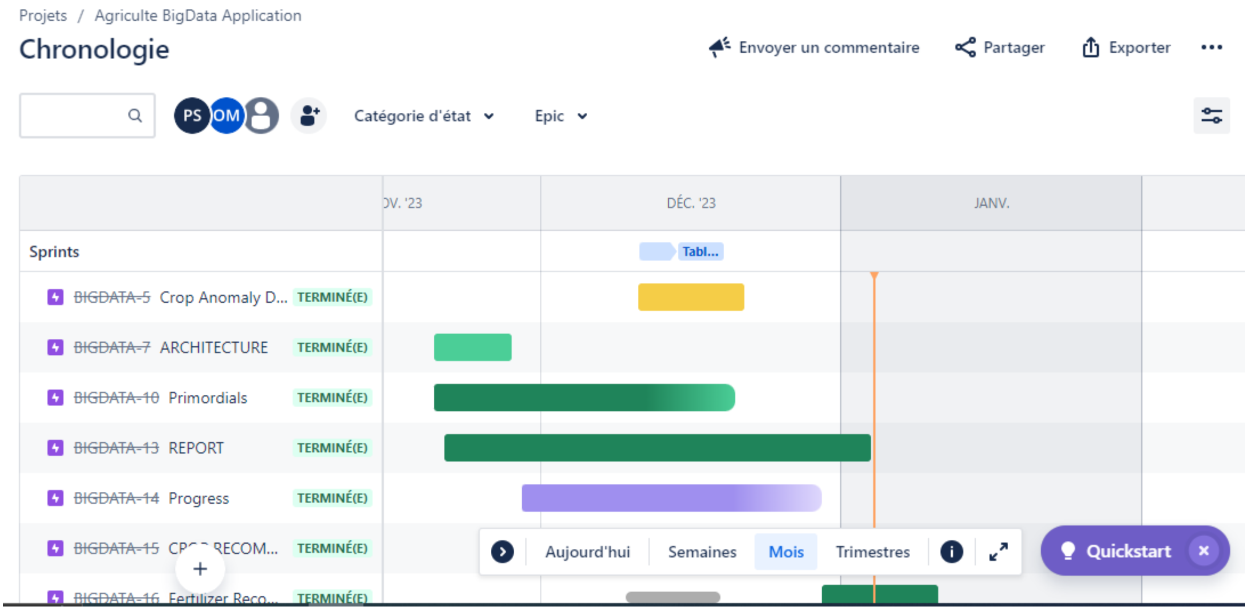

Figure 2.6: Jira Chronology

We also have backlogs, which shows the improvement of the work, either the task is to be done, in progress or done. Which also allows us to see how the person assigned for a given task is improving.

We also have a table that separates clearly the tasks to be done, those in progress and those done, which helps in the evaluation.

This is how it looks like when all the tasks are done.

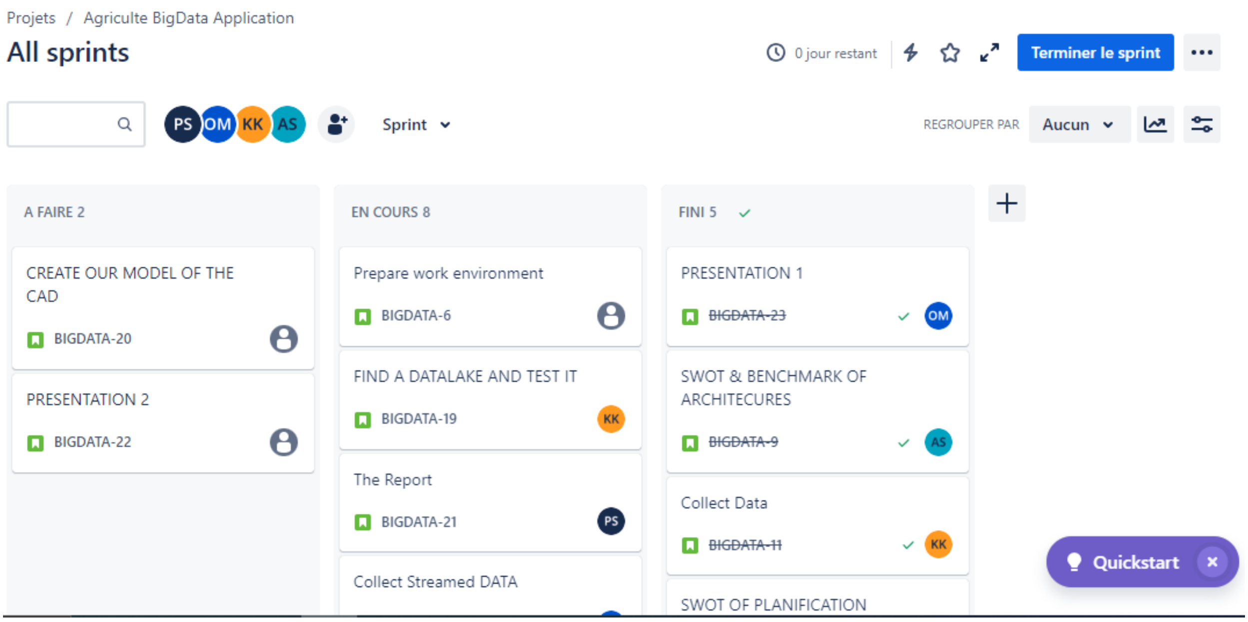

Figure 2.7: Jira Table

2.5 Architecture

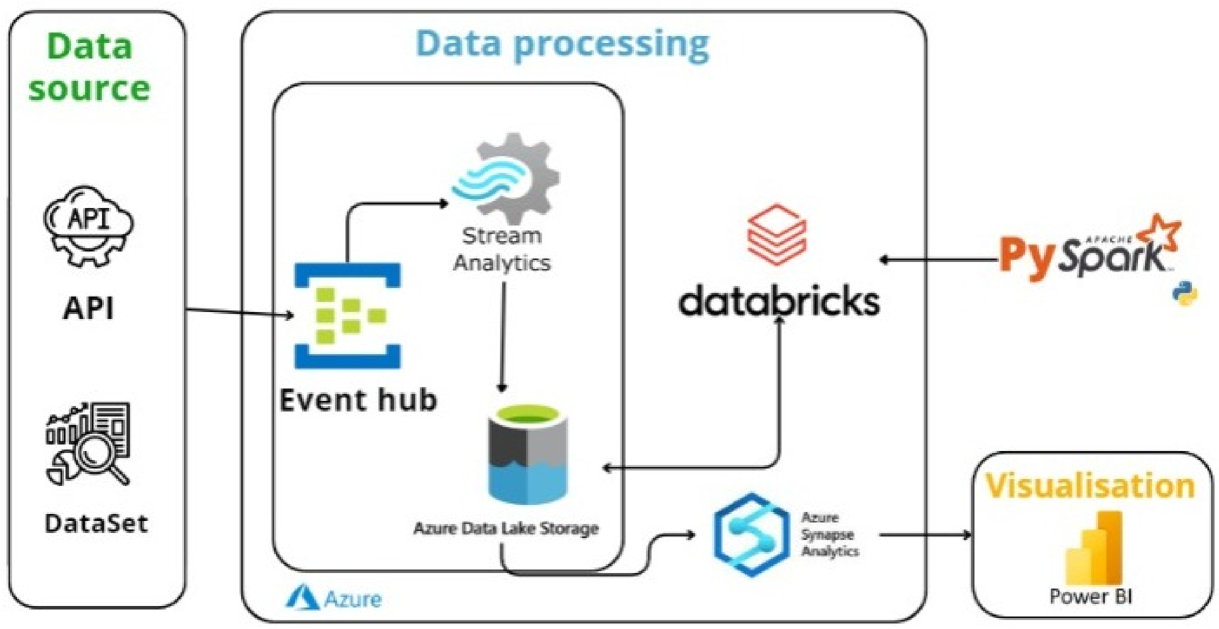

Figure 2.8: Architecture

This diagram provides an overview of the data processing architecture meticulously crafted for our groundbreaking initiative, Intelligent Farms Management. The architecture seamlessly integrates a variety of data sources, including APIs and datasets, to ensure a robust and comprehensive approach to agricultural innovation.

At the outset, our architecture establishes a seamless connection with various APIs, unleashing a data producer tasked with retrieving essential information. This raw data is then meticulously converted into JSON format, making it amenable to streamlined processing.

The heart of our data processing journey lies within Azure, a powerhouse of capabilities. The JSON-formatted data stream is directed to an Azure Event Hub, serving as a centralized hub for data aggregation and management. Here, real-time data processing is orchestrated with precision through the implementation of Azure Stream Analytics, ensuring that we harness the latest insights from the field.

The processed data finds a secure home within Azure Data Lake Storage, poised for future utilization. This strategic choice facilitates scalability and ensures that our data is readily accessible for a wide range of applications.

To elevate our data processing capabilities further, we leverage the formidable combination of Databricks and PySpark. These cutting-edge tools empower us to perform advanced data processing tasks and play a pivotal role in the creation of our Random Forest Classifier model, a key component of our intelligent farm management system.

Azure Synapse Analytics steps in as the guardian of our data, providing robust data warehousing capabilities and comprehensive data analysis tools. This integration allows us to derive valuable insights, trends, and patterns from the vast repository of agricultural data.

To bring our insights to life and make them accessible to stakeholders, we employ Power BI, a dynamic visualization tool. With Power BI, we craft meaningful, interactive visual representations of our processed data, enabling users to gain actionable insights at a glance.

In summary, our architecture is a sophisticated ecosystem designed to harness the power of data and technology, ultimately revolutionizing farm management. It seamlessly connects data sources, processes information in real-time, and presents actionable insights, all aimed at empowering farmers and enhancing the efficiency and sustainability of agricultural practices.

2.6 Tools and Technologies

During the course of our project, we utilized a diverse set of tools and technologies to successfully achieve our objectives. The following is a list of some of the main tools and technologies that played a crucial role in the project:

Python: Principal Programming Language

Figure 2.9: Python Logo

Python served as the foundational programming language for our project. Its simplicity, flexibility, and extensive developer community made it the ideal choice for implementing complex solutions. The Python ecosystem provided us with a wide range of libraries and tools dedicated to natural language processing, enabling the development of sophisticated algorithms and text data manipulation.

Flask for Interface

Figure 2.10: Flask Logo

Flask, a lightweight web framework for Python, was employed for developing web interfaces and APIs. It simplified the process of building web applications with Python, offering a straightforward and extensible architecture.

NumPy

Figure 2.11: NumPy Logo

NumPy, a powerful library for numerical operations in Python, provided support for large, multi-dimensional arrays and matrices, along with mathematical functions for array manipulation.

Pandas

Figure 2.12: Pandas

Pandas, a data manipulation library for Python, introduced data structures like DataFrames for efficient data analysis and manipulation.

Pillow

Figure 2.13: Pillow Logo

Pillow, the Python Imaging Library, extended Python's capabilities with image processing functionalities. It supported the opening, manipulation, and saving of various image file formats.

MixtraL-8*7B

Figure 2.14: MixtraL-8*7B Logo

MixtraL-8*7B, while not widely recognized, appears to be a specific tool or library related to our project. Further information is needed to provide a detailed description and use cases.

Hugging Face

Figure 2.15: Hugging Face Logo

Hugging Face is a platform that offers state-of-the-art natural language processing (NLP) models and a library (transformers) for seamless integration and utilization of these models in various applications.

Azure Cloud

Figure 2.16: Azure Cloud Logo

Microsoft Azure Cloud, a comprehensive cloud computing platform, provided a wide array of services including computing power, storage, databases, machine learning, and more.

And selected Microsoft Azure as our primary cloud platform for this project due to its comprehensive suite of tools and services tailored to our diverse needs. Azure's robust machine learning services, seamless IoT integration, and efficient data storage and processing capabilities were pivotal in our decision. Furthermore, Azure's scalability, strong developer tools, and extensive community support made it the ideal choice for our project. Cost-effectiveness, along with Azure's reputation for reliability and performance, further solidified our decision. Overall, Azure provided us with the necessary foundation to seamlessly integrate and manage various components of our project, leading to its successful execution.

Databricks

Figure 2.17: Databricks Logo

Azure Databricks, built on Apache Spark, provided a cloud-based big data analytics platform. It offered an integrated environment for data engineering, data science, and machine learning.

Power BI

Figure 2.18: Power BI Logo

Power BI, a business analytics service by Microsoft, empowered us with interactive visualizations and business intelligence capabilities. Its user-friendly interface allowed end-users to create their reports and dashboards.

Synapse Analytics

Figure 2.19: Synapse Analytics Logo

Azure Synapse Analytics, formerly known as SQL Data Warehouse, served as a cloud-based integrated analytics service. It seamlessly combined big data and data warehousing capabilities, offering fast and scalable analytics for large datasets.

Azure Event Hubs

Figure 2.20: Azure Event Hubs Logo

Azure Event Hubs is a highly scalable and event ingestion service provided by Microsoft Azure. It enables the processing of large-scale, real-time data streams from various sources, making it ideal for scenarios like telemetry, log, and event processing. With features like partitions and checkpoints, Event Hubs ensures the reliable and efficient handling of high volumes of streaming data.

Stream Analytics

Figure 2.21: Stream Analytics Logo

Azure Stream Analytics is a real-time analytics service provided by Microsoft Azure. It enables you to process and analyze data streams from various sources in real-time. With its scalable and flexible architecture, Stream Analytics is suitable for a wide range of applications, including IoT data processing, real-time monitoring, and more.

Azure Data Lake Gen2 with Storage Account

Figure 2.22: Azure Data Lake Gen2 Logo

Azure Data Lake Storage Gen2, combined with Azure Storage Account, provides a secure and scalable solution for big data analytics. It allows you to store and analyze massive amounts of data with high throughput and low latency. With features like hierarchical namespace and fine-grained access control, Azure Data Lake Gen2 is well-suited for diverse data analytics and processing tasks.

These tools collectively formed a robust toolkit, enabling us to develop, deploy, and manage various components of our agricultural projects. From web interfaces to data processing and machine learning, these tools played a pivotal role in our success.

Chapter 3

Realisation

3.1 Introduction

In the following chapter, we delve into the realization phase of our project, where we transform our carefully planned strategies and chosen technologies into tangible solutions. This phase represents the culmination of our meticulous planning, as we harness the power of the tools and technologies at our disposal to create innovative solutions for the agricultural domain. Throughout this chapter, we will provide a comprehensive overview of the practical implementation of our project, highlighting key milestones, challenges encountered, and the successes achieved. Join us on this journey as we bring our vision to life and demonstrate the real-world impact of our agricultural initiatives.

3.2 Data Source

The provided Python code implements a Flask web application designed to serve data from a CSV file named `cropmetrics.csv`. The code utilizes the `Flask` framework for building web applications and the `pandas` library for data manipulation.

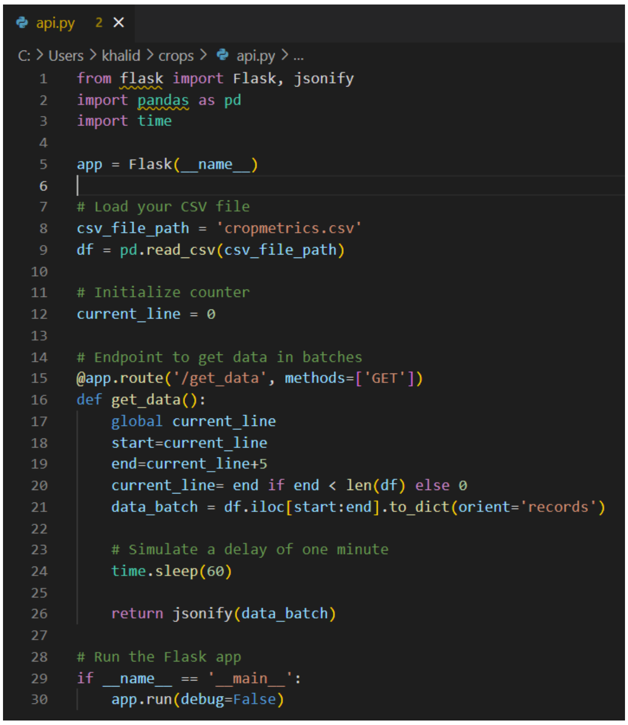

Figure 3.1: Data Source Code Source

- The CSV file is loaded into a DataFrame (`df`) using `pd.read_csv`.

- The application defines an endpoint (`/get_data`) accessible via HTTP GET requests.
- The endpoint returns data in batches, with each batch containing 5 rows from the CSV file.
- A global counter (`current_line`) is employed to keep track of the current position in the dataset.
- The response to the endpoint includes the requested data batch, and a simulated delay of one minute (`time.sleep(60)`) is introduced to mimic potential real-world scenarios where data retrieval may take time.

The Flask application is configured to run when the script is executed directly (`if __name__ == '__main__': app.run(debug=False)`), allowing it to handle incoming requests.

This code serves as a simple example of a Flask web application providing data from a CSV file with a mechanism for controlled data retrieval and a simulated delay for illustrative purposes.

The provided CSV file contains agricultural data related to crop metrics. Each row in the dataset represents information about a specific crop, including various attributes associated with its cultivation. Here is a breakdown of the columns:

- **N (Nitrogen):** Represents the Nitrogen content in the soil.
- **P (Phosphorus):** Indicates the Phosphorus content in the soil.
- **K (Potassium):** Denotes the Potassium content in the soil.
- **Temperature:** Represents the temperature in the crop's environment.
- **Humidity:** Indicates the humidity level in the crop's environment.
- **pH:** Represents the pH level of the soil.
- **Rainfall:** Indicates the amount of rainfall in the area.
- **Label:** Specifies the crop type (e.g., lentil, banana).
- **Latitude:** Represents the latitude coordinate of the crop's location.
- **Longitude:** Denotes the longitude coordinate of the crop's location.
- **Ville (City):** Specifies the city or town associated with the crop.
- **Province:** Indicates the province or region where the crop is located.
- **id_position:** Provides a unique identifier for the position of the crop, typically combining latitude and longitude.

The data in this file is structured to provide comprehensive information about different crops, their environmental conditions, and geographical locations. Each row serves as a record for a specific crop entry, making it a valuable dataset for agricultural analysis and monitoring.

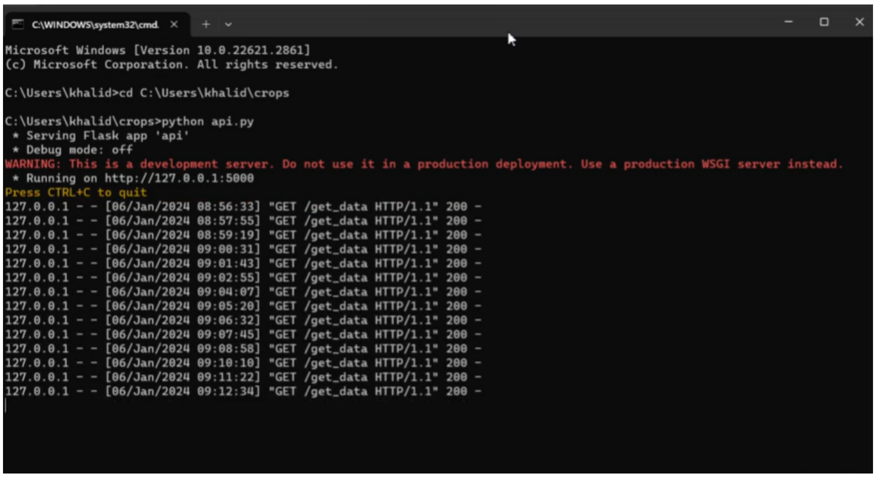

Figure 3.2: Running Data Api

3.3 Azure Event Hub Producer Script

The following Python script serves as an Azure Event Hub producer, responsible for fetching data from a local endpoint and sending it to an Azure Event Hub. Below is a breakdown of its key components and functionality:

1. **Local Data Endpoint:**
   - `local_data_endpoint`: Specifies the local endpoint (`http://127.0.0.1:5000/get_data`) from which data is fetched. Assumes a local data source returning data in JSON format.
2. **Azure Event Hub Configuration:**
   - `connection_str`: Contains the connection string for the Azure Event Hub, including service bus endpoint, shared access key, and shared access key name.
   - `eventhub_name`: Specifies the name of the target Event Hub (`pakostrealing`).
3. **Event Hub Producer Client Initialization:**
   - `producer_client`: Initializes an `EventHubProducerClient` using the provided connection string and Event Hub name.
4. **Data Fetching and Sending:**
   - `fetch_and_send_data()`: Fetches data from the local endpoint using an HTTP GET request and sends it to the Azure Event Hub.
   - Logs the fetched data and sends it to the Event Hub using an `EventHubProducerClient`.
   - Converts the data to a string before sending it to the Event Hub.
5. **Logging:**
   - `logging`: Configures basic logging at the INFO level. Logs information about the fetched data and the success of sending data to the Azure Event Hub.
6. **Execution:**
   - The script includes an infinite loop (`while True`) to continuously fetch and send data to the Event Hub. It runs indefinitely until manually stopped.
7. **Error Handling:**
   - Exception handling is implemented to catch and log any errors that may occur during the data fetching and sending process.

```python
import requests
import logging
from azure.eventhub import EventHubProducerClient, EventData

# Local endpoint to generate data
local_data_endpoint = "http://127.0.0.1:5000/get_data"

# Azure Event Hub connection string
connection_str = "Endpoint=sb://pakoeventhub.servicebus.windows.net/;SharedAccessKeyName=pakogrow;SharedAccessKey=dhi/SExjVa

# Event Hub parameters
eventhub_name = "pakostrealing"

# Create an Event Hub producer client
producer_client = EventHubProducerClient.from_connection_string(connection_str, eventhub_name=eventhub_name)

def fetch_and_send_data():
    try:
        response = requests.get(local_data_endpoint)
        data_list = response.json()

        for data in data_list:
            logging.info(f"Data fetched from local endpoint: {data}")

            with producer_client:
                event_data_batch = producer_client.create_batch()
                event_data_batch.add(EventData(str(data)))  # Convert data to string for Event Hub
                producer_client.send_batch(event_data_batch)

            logging.info("Data sent to Azure Event Hub successfully.")

    except Exception as e:
        logging.error(f"Error: {e}")

if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO)  # Adjust logging level as needed
    while True:
        fetch_and_send_data()
```

Figure 3.3: Event hub Producer Script

This script acts as a bridge between a local data source and an Azure Event Hub, facilitating continuous data transfer from a local environment to the cloud for further processing or analysis.

3.4 Streaming Data with Azure Stream Analytics and Store it In DataLake

The streaming data process commences with Azure Stream Analytics ingesting real-time data from the Event Hub container `pakostrealing`. Configured with a SQL-like query, Stream Analytics dynamically processes the incoming data, applying transformations and business logic in real-time. Subsequently, the processed data seamlessly flows into the Data Lake Gen2 container `pakodatalake`, serving as a robust storage solution. This end-to-end pipeline operates continuously, facilitating persistent streaming from the Event Hub through Stream Analytics and into the Data Lake Gen2. The process ensures scalability, fault tolerance, and near-real-time analytics, thereby enabling efficient storage and analysis of streaming data in an organized and scalable manner.

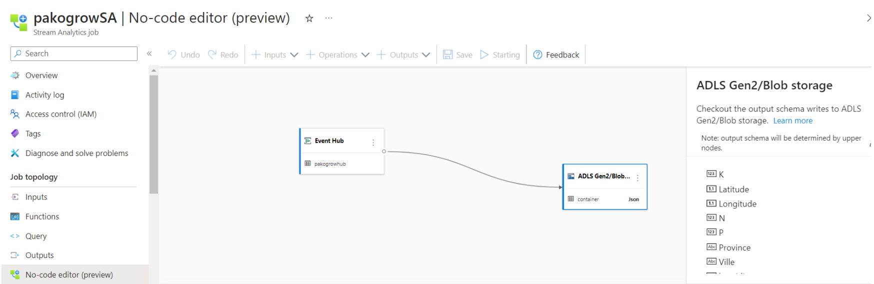

Figure 3.4: Stream Analytics

Azure Stream Analytics efficiently stores our streaming data in the designated container within Azure Data Lake Gen2. Leveraging its seamless integration capabilities, Stream Analytics ensures a continuous flow of real-time data from the source, be it an Event Hub or another streaming input, directly into our specified Data Lake Gen2 container. This process offers a scalable and organized approach to persistently store streaming data, providing a foundation for subsequent analytics, reporting, and data-driven decision-making within our Azure environment.

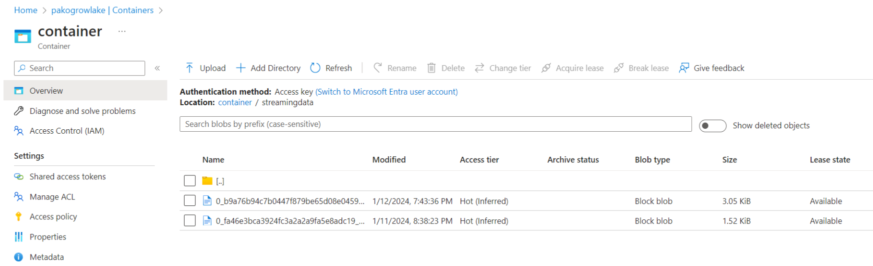

Figure 3.5: Storing Streaming Data in Data Lake

3.5 Creating a Data Warehouse in Synapse Analytics

Creating a data warehouse in Azure Synapse Analytics involves defining a schema with a fact table and dimension tables. In this scenario, we have a fact table named `crops` that stores key agricultural metrics such as Nitrogen (N), Phosphorus (P), Potassium (K), pH, rainfall, temperature, and humidity. The `crops` table includes foreign keys, linking to other dimension tables: `id_position`, `id_temps`, and `id_crops`.

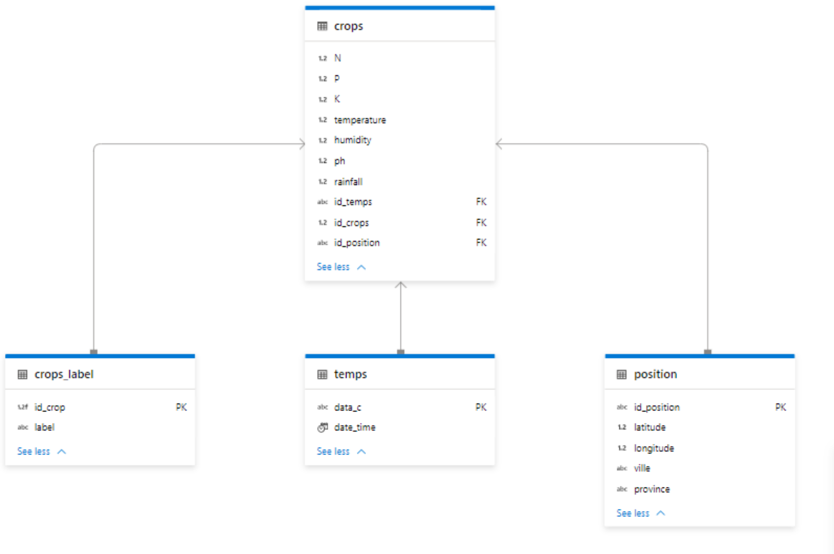

Figure 3.6: PakoGrow Data Warehouse

The `crops_label` dimension table acts as a lookup table, containing a primary key `id_crops` and corresponding labels for each crop. This facilitates easy referencing and categorization in the fact table.

The `position` dimension table holds geographical information with `id_position` as the primary key. It includes attributes such as latitude, longitude, ville (city), and province, enabling spatial analysis based on crop positions.

The `temps` dimension table serves as a time dimension, featuring `date_c` as the primary key for date and time information. This table allows tracking temporal patterns in agricultural metrics, supporting time-based analyses.

This schema structure in Synapse Analytics provides an efficient way to query and analyze agricultural data, fostering relationships between different aspects of the dataset for comprehensive insights.

Each table input is stored in our Data Lake, respectively in `pakodatalake/pakogrowdw/crops`, `pakodatalake/pakogrowdw/crops_label`, `pakodatalake/pakogrowdw/position`, and `pakodatalake/pakogrowdw/temps`.

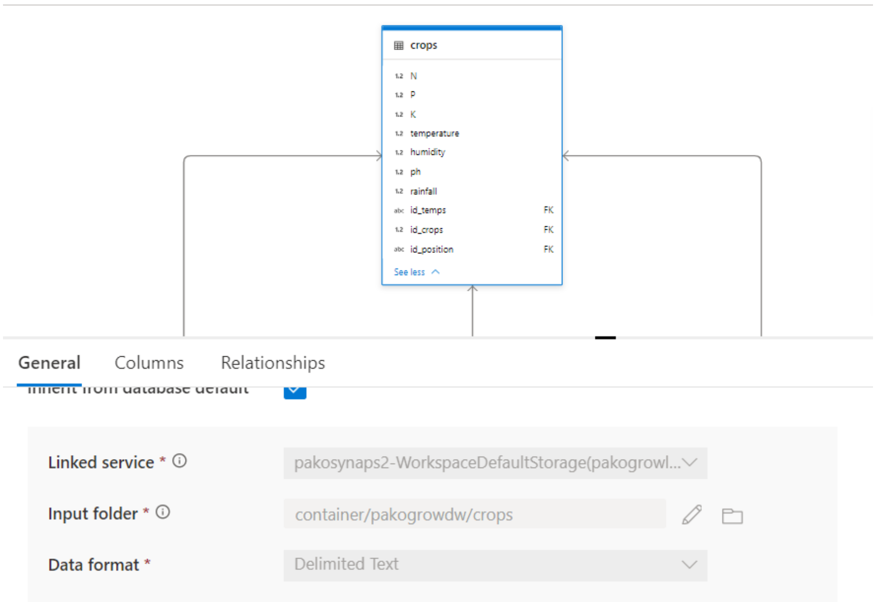

Figure 3.7: Tables Input Directory

3.6 Databricks Overview

Databricks is a unified analytics platform built on top of Apache Spark that simplifies the process of building and deploying data-driven applications. It provides a collaborative environment for data scientists, data engineers, and analysts to work together seamlessly. Databricks includes features such as notebooks for interactive data exploration and analysis, as well as clusters for distributed computing.

3.6.1 Workflow Description

Here is a step-by-step description of the process you outlined using Databricks:

Notebook: Models

In this notebook, you trained four machine learning models: Logistic Regression, RandomForestClassifier, DecisionTreeClassifier, and NaiveBayes. After evaluating the models, you determined that the RandomForestClassifier performed the best.

accuracy	f1	name	weightedPrecision	weightedRecall
0.9335302806499262	0.9328546376793694	LogisticRegression	0.9408651387503908	0.9335302806499263
0.9542097488921714	0.9526893331521918	RandomForestClassifier	0.9624293924833832	0.9542097488921713
0.48892171344165436	0.3964340106660314	DecisionTreeClassifier	0.3719783553952938	0.48892171344165436
0.8906942392909897	0.8888381548588637	NaiveBayes	0.8911505419698642	0.8906942392909896

Figure 3.8: Models Evaluation

To train our models, we created a PySpark script that elaborate the process below:

1. **Initializing Spark Session:**
   - Initiates a Spark session with the name "RandomForestClassifiers" to enable Spark functionalities for distributed computation.
2. **Data Preprocessing:**
   - Utilizes a StringIndexer to convert string labels to numeric values for machine learning compatibility.

- Features columns ("N", "P", "K", "temperature", "humidity", "ph", "rainfall") are selected and assembled into a single 'features' column using VectorAssembler.

3. **Data Splitting:**
   - Splits the dataset into training and test sets in a 70-30 ratio. The split is stratified, ensuring proportional representation of classes in both sets.
4. **Random Forest Classifier Initialization and Training:**
   - Initializes a Random Forest Classifier with 100 trees and specifies the label column as "indexed-Label".
   - Trains the Random Forest model using the training data.
5. **Prediction Generation:**
   - Utilizes the trained model to generate predictions on the test data.
6. **Model Evaluation:**
   - Measures the accuracy of the model's predictions using the MulticlassClassificationEvaluator with the metric set to "accuracy".
   - Prints and displays the accuracy achieved by the Random Forest Classifier on the test data.

Notebook: Crops

In this notebook, you use the RandomForestClassifier to predict crop labels for streaming data incoming from Azure ADLS Gen2 (Azure Data Lake Storage Gen2). You employ the `spark.readStream` function to read streaming data and `spark.writeStream` to write the predictions into the input of your fact table in the Data Lake.

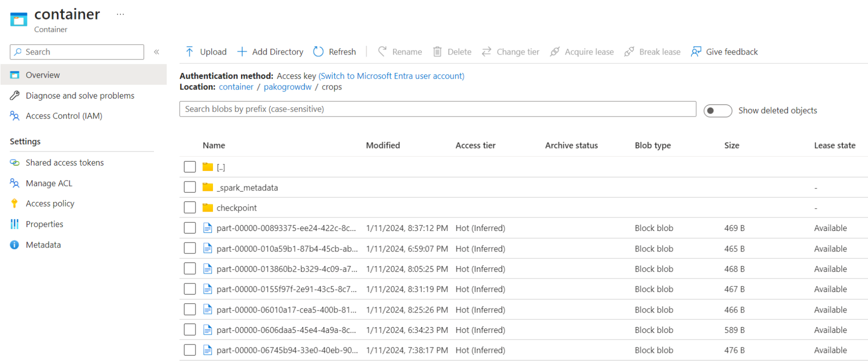

Figure 3.9: Storing Data into DataWarehouse Tables Input

Notebook: Temps

This notebook focuses on reading streamed data, filtering the time of the incoming data, and shaping it into a daytime format. The processed data is then stored in the input of a dimension table named 'temps' in your Data Lake.

Further Elaboration

Streaming Data Ingestion

Databricks supports structured streaming, which allows you to process streaming data using Spark SQL and DataFrames. `spark.readStream` is used to set up a streaming DataFrame to ingest data continuously.

Model Training

The 'Models' notebook involves training machine learning models using Spark's ML-based libraries. The RandomForestClassifier is chosen as the best model for your use case.

Prediction and Fact Table Update

The 'Crops' notebook uses the trained RandomForestClassifier to predict crop labels for incoming streaming data. The predictions are then written to the input of the fact table in your Data Lake using `spark.writeStream`.

Time Filtering and Dimension Table Update

The 'Temps' notebook reads the streamed data, filters it based on time, and transforms it into a daytime format. The processed data is then stored in the input of the 'temps' dimension table using `spark.writeStream`.

This entire process demonstrates the power of Databricks in handling end-to-end data workflows, from model training to real-time streaming data processing and storage in a Data Lake. It leverages Spark's capabilities for distributed computing and stream processing to efficiently handle large-scale data analytics.

3.7 Data Visualization with Power BI

As a data engineer in a data project, the focus may center on creating robust, scalable data pipelines facilitating the collection, cleaning, and transformation of data. Visualization responsibilities often align more with data scientists and analysts, interpreting and communicating insights. However, data engineers might build systems that efficiently store, retrieve, and process data for visualization purposes. Collaboration among data engineers, scientists, and analysts ensures efficient data management and effective interpretation and communication through visualization for impactful decision-making.

In our project, we leverage Power BI to visualize data, deriving insights crucial for optimizing crop distribution. We analyze crop distribution, assess soil and air conditions essential for robust crop growth, and examine the geographical positioning of farms across Morocco. Utilizing Power BI's filtering capabilities, we refine data analysis by date, city, crop labels, temperature, and rainfall, facilitating a comprehensive understanding of agricultural factors influencing crop yield and distribution.

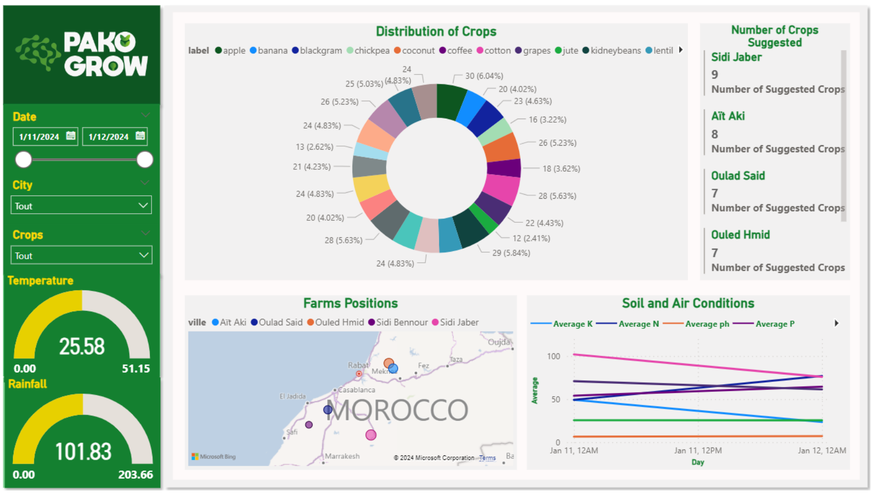

Figure 3.10: Crops Detection Dashboard

We implemented specific tools for filtering in our visualization:

- Utilizing slicers for city and crop selection, enabling precise filtering based on geographical locations and crop types.
- Employing gauges to visualize temperature and rainfall data, offering an intuitive representation of these crucial environmental factors.

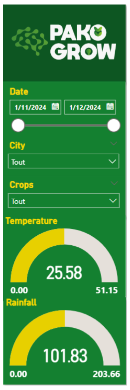

Figure 3.11: Filtering Panel

To precisely pinpoint the geographic distribution of farms within Morocco, we employed mapping tools, allowing us to visualize their spatial positioning and better understand their dispersion across the country.

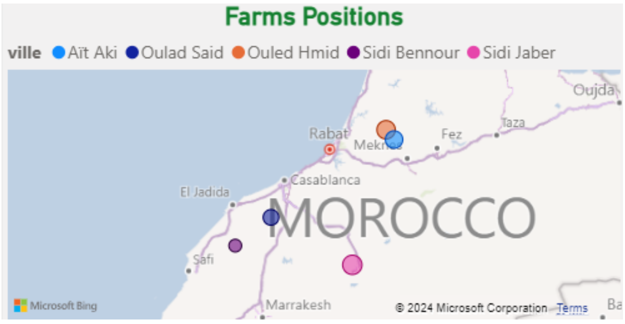

Figure 3.12: Farms Distribution Map

A Doughnut chart is utilized to showcase the distribution of crops, displaying the percentages of each crop category. This visual representation allows for a quick and clear understanding of the proportional allocation of different crops within the dataset. The Doughnut chart provides a concise overview, illustrating the relative sizes of crop categories in relation to the whole, aiding in assessing the distribution pattern and proportions of various crops.

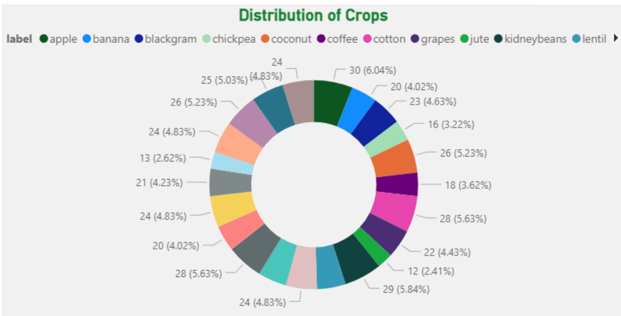

Figure 3.13: Doughnut Chart

We present an intuitive interface where users can easily navigate and explore farms condition metrics by time using lineplots.

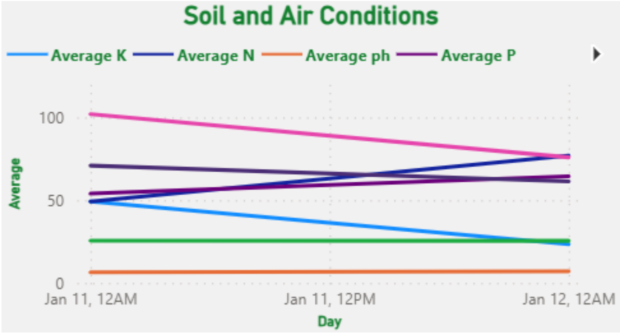

Figure 3.14: Line Plot of Farms Condition Metrics by Time

We utilized our multi-card visualization to see how many crops suggested in every farm.

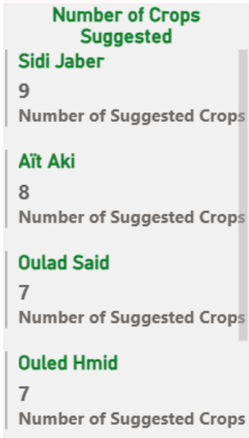

Figure 3.15: Number of Crops Suggested by Farms

3.8 AI Models

As a quick recap, our project centers around the development of an advanced agricultural management system to do that we will introduce again next Models:

- **Crop Anomaly Detection:** Identify crop issues (diseases, pests) using CNNs.
- **Crop Recommendation:** Suggest suitable crops based on climate and soil data.
- **Fertilizer Recommendation:** Optimize fertilizer usage for sustainable agriculture.
- **Pako GPT Chatbot:** Provide real-time farming advice via a chatbot.

3.8.1 Crop Anomaly Detection:

Dataset Description

The dataset used in this project is derived from the original PlantVillage Dataset, enhanced through offline augmentation. It comprises approximately 87,000 RGB images of crop leaves, categorized into 38 distinct classes. Notably, the dataset covers 14 different plant types and encompasses 26 distinct diseases or anomalies.

the unique plants include:

- Tomato
- Grape
- Orange
- Soybean

- Squash
- Potato
- Corn (maize)
- Strawberry + 6 others

After acquiring the dataset, we performed essential data preprocessing steps. These steps involved transforming pixel values in the range of 0 to 255 to a normalized scale of 0 to 1. This normalization facilitates improved model performance and convergence during training.

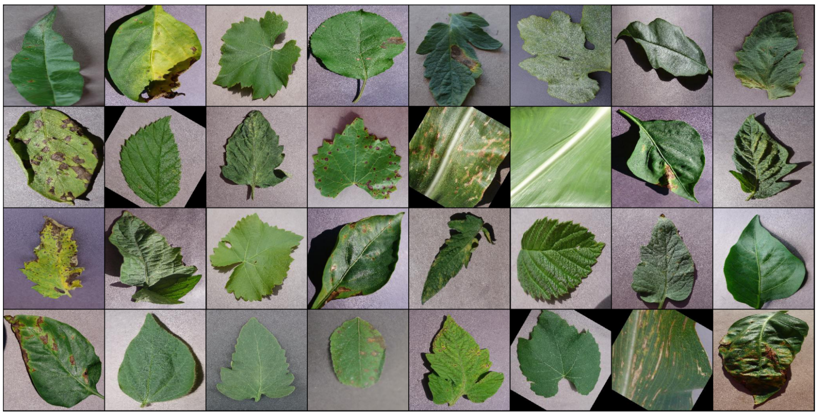

Figure 3.16: Data Crop Anomaly Examples

Model Architecture

In our project, we employed the ResNet architecture, which has been a seminal advancement in computer vision since its introduction in 2015.

The architecture is designed as follows:

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 256, 256]           1,792
       BatchNorm2d-2         [-1, 64, 256, 256]             128
              ReLU-3         [-1, 64, 256, 256]               0
            Conv2d-4        [-1, 128, 256, 256]          73,856
       BatchNorm2d-5        [-1, 128, 256, 256]             256
              ReLU-6        [-1, 128, 256, 256]               0
         MaxPool2d-7          [-1, 128, 64, 64]               0
            Conv2d-8          [-1, 128, 64, 64]         147,584
       BatchNorm2d-9          [-1, 128, 64, 64]             256
             ReLU-10          [-1, 128, 64, 64]               0
           Conv2d-11          [-1, 128, 64, 64]         147,584
      BatchNorm2d-12          [-1, 128, 64, 64]             256
             ReLU-13          [-1, 128, 64, 64]               0
           Conv2d-14          [-1, 256, 64, 64]         295,168
      BatchNorm2d-15          [-1, 256, 64, 64]             512
             ReLU-16          [-1, 256, 64, 64]               0
        MaxPool2d-17          [-1, 256, 16, 16]               0
           Conv2d-18          [-1, 512, 16, 16]       1,180,160
      BatchNorm2d-19          [-1, 512, 16, 16]           1,024
             ReLU-20          [-1, 512, 16, 16]               0
        MaxPool2d-21            [-1, 512, 4, 4]               0
           Conv2d-22            [-1, 512, 4, 4]       2,359,808
      BatchNorm2d-23            [-1, 512, 4, 4]           1,024
             ReLU-24            [-1, 512, 4, 4]               0
           Conv2d-25            [-1, 512, 4, 4]       2,359,808
      BatchNorm2d-26            [-1, 512, 4, 4]           1,024
             ReLU-27            [-1, 512, 4, 4]               0
        MaxPool2d-28            [-1, 512, 1, 1]               0
          Flatten-29                  [-1, 512]               0
           Linear-30                   [-1, 38]          19,494
================================================================
Total params: 6,589,734
Trainable params: 6,589,734
Non-trainable params: 0
----------------------------------------------------------------
```

Figure 3.17: Model architecture

ResNet Architecture Our ResNet-based model consists of several key components:

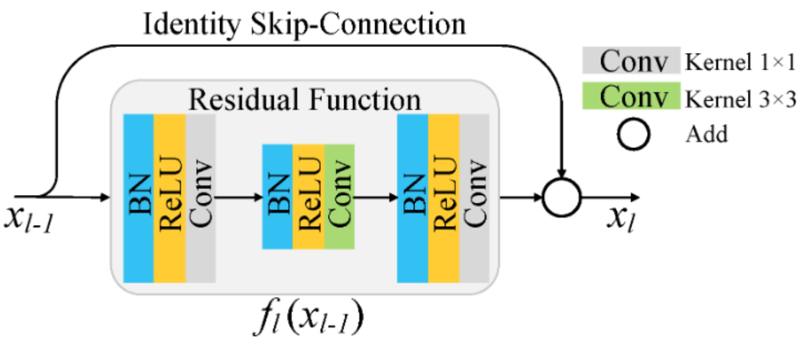

Figure 3.18: ResNet architecture

- **Convolution Blocks:** We utilize convolutional layers with batch normalization and ReLU activation functions for feature extraction.
- **Residual Blocks:** Residual connections are incorporated to address the vanishing gradient problem. These blocks allow for the smooth flow of gradients during training.
- **Pooling Layers:** Max-pooling layers help downsample feature maps, reducing the spatial dimensions and computational load.
- **Fully Connected Layer:** The model concludes with a fully connected layer that maps the learned features to disease predictions.

Training and Validation

We conducted extensive training of our ResNet model. On Kaggle (P100 GPU), the training process took approximately 20 minutes of wall time. This efficient training resulted in an impressive accuracy of 99.2

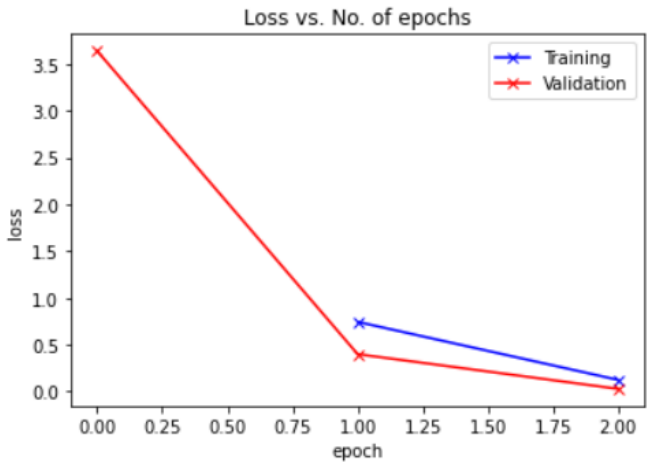

Figure 3.19: Loss vs No epochs

Validation Accuracy Plot

The validation accuracy plot (see Figure 3.20) showcases the model's performance throughout training. It demonstrates a steady increase in accuracy, ultimately reaching a remarkable 99.2

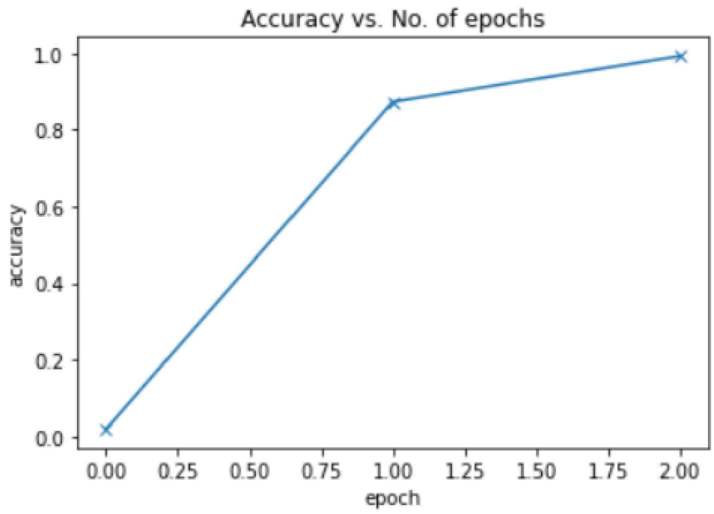

Figure 3.20: Validation Accuracy Plot

Our model's exceptional accuracy highlights its capability to effectively differentiate between healthy and anomalous crops, contributing to the success of our agricultural project.

Fertilizer Recommendation:

In the domain of agriculture, effective fertilizer management is crucial to ensure optimal crop growth and yield. To address this aspect of our project, we focused on recommending suitable fertilizers based on the soil's nutrient composition. The dataset used for this purpose contains information about various crops and their corresponding soil characteristics, including Nitrogen (N), Phosphorus (P), Potassium (K), pH level, and soil moisture. This dataset serves as the foundation for making informed fertilizer recommendations tailored to the specific needs of each crop.

Dataset Description

The dataset encompasses a variety of crops, such as rice, maize, chickpea, kidney beans, and many others. It provides key nutrient values for each crop, including the levels of Nitrogen, Phosphorus, Potassium, pH, and soil moisture. This dataset enables us to assess the nutritional requirements of different crops and recommend suitable fertilizers to enhance their growth.

	Crop	N	P	K	pH	soil_moisture
0	rice	80	40	40	5.5	30
3	maize	80	40	20	5.5	50
5	chickpea	40	60	80	5.5	60
12	kidneybeans	20	60	20	5.5	45
13	pigeonpeas	20	60	20	5.5	45
14	mothbeans	20	40	20	5.5	30

Figure 3.21: Data Fertilizer Examples

Recommendations

Our fertilizer recommendation system takes into account the following soil nutrient scenarios:

- **High Nitrogen (N) Content:** In cases where the soil exhibits high nitrogen levels, it is essential to manage it effectively. Recommendations include adding manure, using coffee grounds as a nitrogen-rich source, planting nitrogen-fixing plants, and more.
- **Low Nitrogen (N) Content:** When the soil lacks sufficient nitrogen, strategies like adding sawdust, planting nitrogen-hungry crops, leaching the soil with water, and incorporating sugar can help improve nitrogen levels.
- **High Phosphorus (P) Content:** Soil with elevated phosphorus levels can benefit from avoiding excessive manure, using phosphorus-free fertilizers, and adequate watering to drive phosphorus out of the soil. Planting nitrogen-fixing vegetables and crop rotations can also be effective.
- **Low Phosphorus (P) Content:** In cases of phosphorus deficiency, remedies include using bone meal, rock phosphate, phosphorus fertilizers, organic compost, and clay soil. Ensuring proper soil pH and applying lime or potassium carbonate for low pH levels can also aid phosphorus uptake.
- **High Potassium (K) Content:** Soils with high potassium levels can be improved by loosening the soil deeply, removing rocks, discontinuing the use of potassium-rich commercial fertilizers, and incorporating materials like crushed eggshells and organic compost.
- **Low Potassium (K) Content:** When soil lacks potassium, solutions include mixing in muricate of potash, using kelp meal or seaweed, trying Sul-Po-Mag, burying banana peels, and utilizing potash fertilizers.

By analyzing the soil nutrient composition and considering these recommendations, farmers and agricultural practitioners can make informed decisions about fertilization practices, ultimately contributing to sustainable and efficient crop cultivation.

Pako GPT Chatbot:

The Pako GPT Chatbot plays a pivotal role in providing real-time agricultural guidance and information to users. Designed to assist individuals in the field of agriculture, this chatbot leverages advanced language processing capabilities to deliver personalized responses to user queries.

Figure 3.22: Pako Gpt

Chatbot Operation

The chatbot operates by receiving user queries related to agricultural practices and concerns. These queries are initially translated from Moroccan Arabic to English to ensure effective communication with the model. The chatbot uses the Hugging Face API, specifically the "Mixtral-8x7B-Instruct-v0.1" model, to generate responses based on the provided user input.

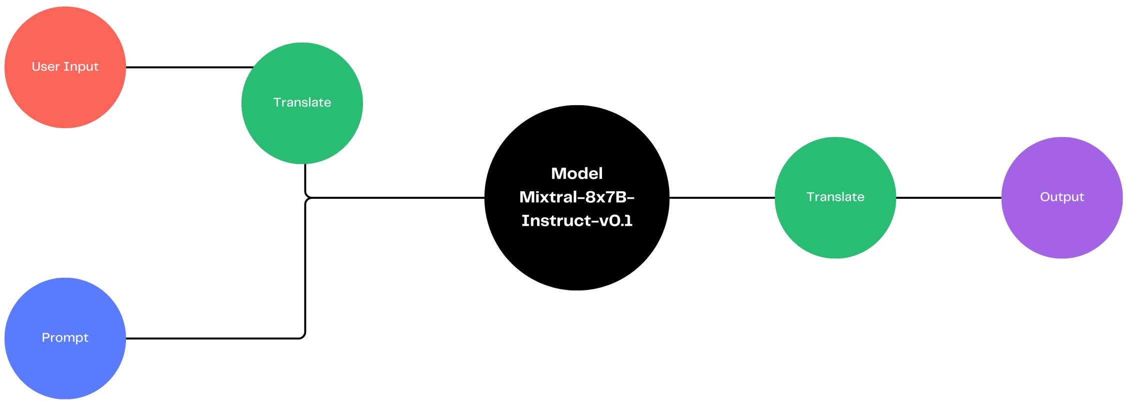

Figure 3.23: Pako Gpt Operation

Response Generation

The process begins with the translation of the user's query to English, ensuring that the chatbot can comprehend and process the input effectively. The translated query is then passed to the Hugging Face API with parameters specifying a maximum token limit, temperature, and whether to return the full text.

The chatbot generates responses based on the model's predictions and returns the translated responses to the user. These responses provide valuable agricultural information, guidance, and recommendations in Moroccan Arabic, facilitating clear communication with users.

Fertilizer Recommendations

One notable aspect of the chatbot's functionality is its ability to provide fertilizer recommendations based on soil nutrient composition. By understanding the soil's Nitrogen (N), Phosphorus (P), Potassium (K), pH level, and soil moisture, the chatbot can offer tailored advice on appropriate fertilization practices.

Enhancing Agricultural Decision-Making

The Pako GPT Chatbot serves as a valuable tool for farmers, agricultural practitioners, and enthusiasts, offering instant access to agricultural expertise. Its ability to bridge language barriers and provide customized guidance empowers users to make informed decisions, address agricultural challenges, and optimize farming practices.

Overall, the Pako GPT Chatbot contributes to the project's mission of promoting sustainable and efficient agriculture through the integration of advanced language processing and agricultural knowledge.

Crop Recommendation:

This details the development and evaluation of a Crop Recommendation model aimed at assisting farmers in making informed decisions about optimal crop selection. The objective was to leverage machine learning techniques to provide personalized crop recommendations based on diverse factors, including climate conditions and soil characteristics.

Dataset Description

For our project, we accessed a comprehensive dataset from Kaggle, a renowned platform for data science enthusiasts and practitioners. The dataset, available at https://www.kaggle.com/code/dhamur/machine-learning-in-agriculture, contains a total of 2200 rows and encompasses a rich array of agricultural data attributes.

The dataset includes the following key parameters:

- **N (Nitrogen)**
- **P (Phosphorus)**
- **K (Potassium)**
- **Temperature**: Representing the average soil temperatures suitable for bioactivity, ranging from 50 to 75°F.
- **pH**: Indicating the acidity or basicity of the soil, categorized as Acidic (pH¡7), Neutral (pH=7), or Basic (pH¿7).
- **Label**: Denoting various types of crops, including Rice, Maize, Chickpea, Kidney beans, Pigeonpeas, Mothbeans, Mungbean, Blackgram, Lentil, Pomegranate, Banana, Mango, Grapes, Watermelon, Muskmelon, Apple, Orange, Papaya, Coconut, Cotton, Jute, Coffee.

	N	P	K	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

Figure 3.24: Data Description Crop Recommendation

This dataset provides a comprehensive overview of critical agricultural parameters and serves as the foundation for our crop prediction model, enabling data-driven insights and recommendations for sustainable farming practices.

Model Selection:

Various machine learning algorithms, such as Logistic Regression, Decision Trees, Naïve Biases, and Random Forest, were considered for the Crop Recommendation task. The models were trained and evaluated using the dataset, and Random Forest emerged as the most suitable model due to its ability to handle complex relationships and provide accurate predictions.

accuracy	f1	name	weightedPrecision	weightedRecall
0.9335302806499262	0.9328546376793694	LogisticRegression	0.9408651387503908	0.9335302806499263
0.9542097488921714	0.9526893331521918	RandomForestClassifier	0.9624293924833832	0.9542097488921713
0.48892171344165436	0.3964340106660314	DecisionTreeClassifier	0.3719783553952938	0.48892171344165436
0.8906942392909897	0.8888381548588637	NaiveBayes	0.8911505419698642	0.8906942392909896

Figure 3.25: Model's Crop Recommendation

The chosen **Random Forest model** was implemented using Python and scikit-learn. The dataset was split into training and testing sets to train the model on historical data and assess its predictive capabilities. The model incorporated variables like temperature, soil pH, and historical climate data to generate crop recommendations

Validation and Evaluation

The trained Random Forest model underwent rigorous testing using a separate set of data not seen during training. This step aimed to validate the model's ability to generalize and offer accurate crop recommendations for various combinations of climate and soil conditions.
The performance of the Crop Recommendation model was assessed using key metrics, including precision, recall, F1 score, and accuracy. These metrics provided insights into the model's effectiveness in correctly recommending crops while considering factors such as false positives and negatives.
The successful creation and evaluation of the Crop Recommendation model offer significant benefits to farmers. The personalized recommendations enable farmers to make data-driven decisions, mitigating risks associated with unpredictable weather patterns and soil variations. This contributes to increased crop yields and economic returns.

3.9 Interface

The Pako Grow project incorporates a user-friendly web-based interface to facilitate easy access to its agricultural services and features. This section provides an overview of the interface components and their functionalities.

Technologies Used

The interface is built using Flask, a web framework for Python. It leverages several libraries and modules to deliver a seamless user experience:

Figure 3.26: Technologies

- **Flask**: The core framework for handling web requests and rendering templates.
- **Numpy and Pandas**: Used for data manipulation and processing.
- **Requests**: Used for making API requests to fetch weather data.
- **Pickle**: Utilized to load the crop recommendation model.
- **Torch and torchvision**: Employed for disease prediction using a pre-trained ResNet model.
- **Gradio-client**: Facilitates interaction with the Hugging Face API for chatbot functionality.

Functionalities

1. **Home Page**: The home page serves as the entry point to the Pako GROW platform. It provides an introduction to the project and easy navigation to other sections. he user-friendly interface, as depicted in Figure 3.27, offers a glimpse of the platform's aesthetics. The welcoming atmosphere invites users to explore further.

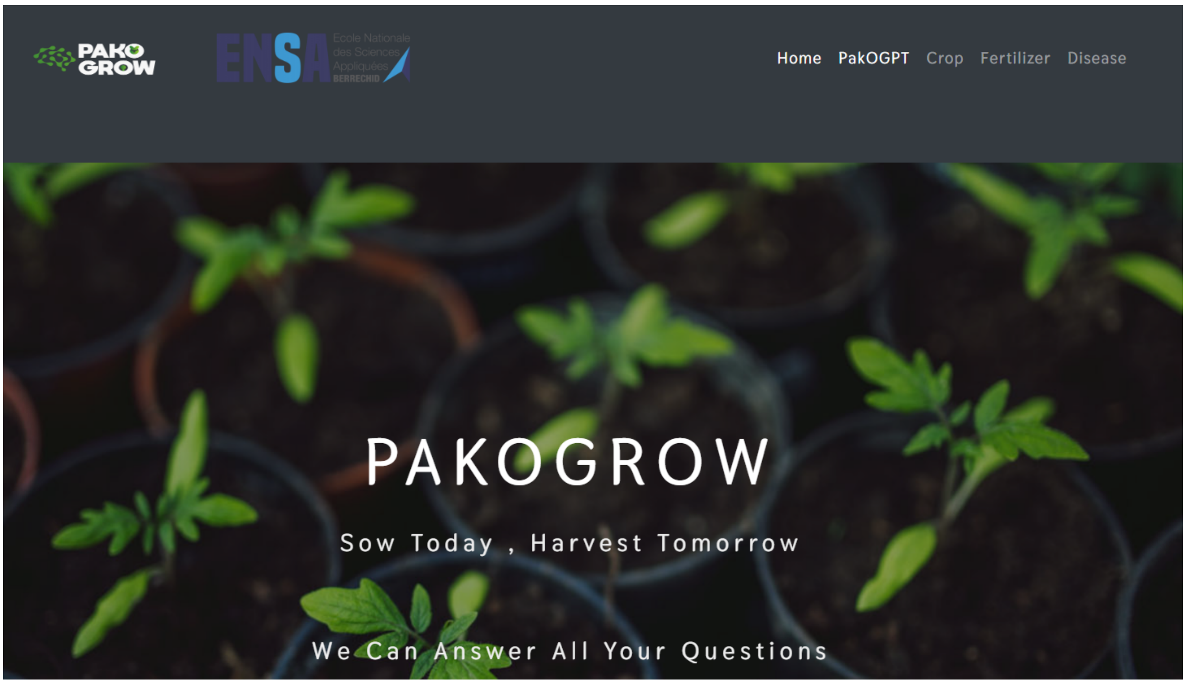

Figure 3.27: Home Interface

The Home Page also features additional sections, including "About Us," "Team," and "Services." These sections offer valuable insights into the project's background, the talented individuals behind it, and the array of services and tools available to users.

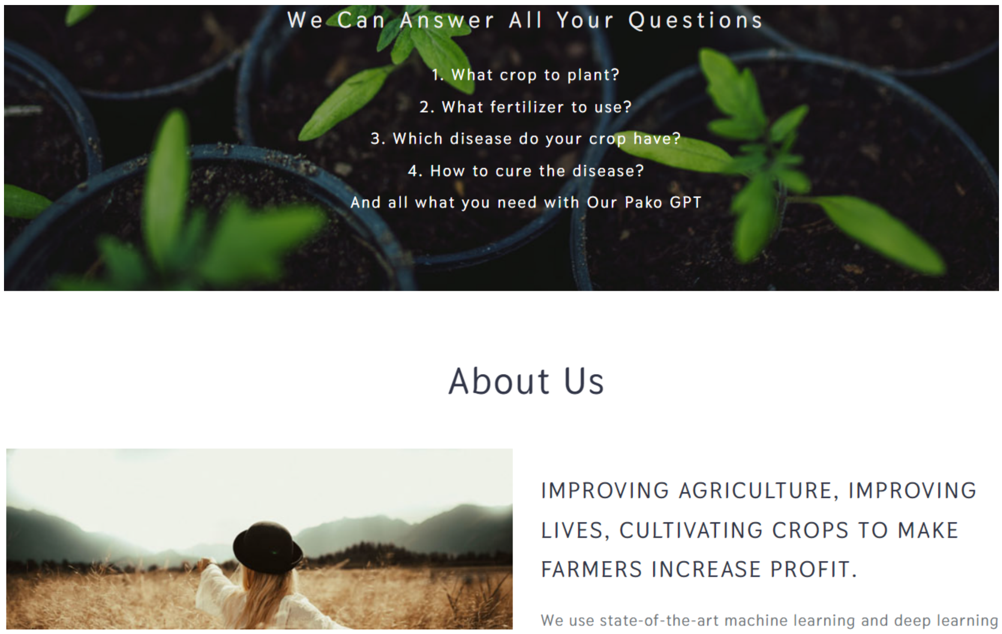

Figure 3.28: Home Interface About

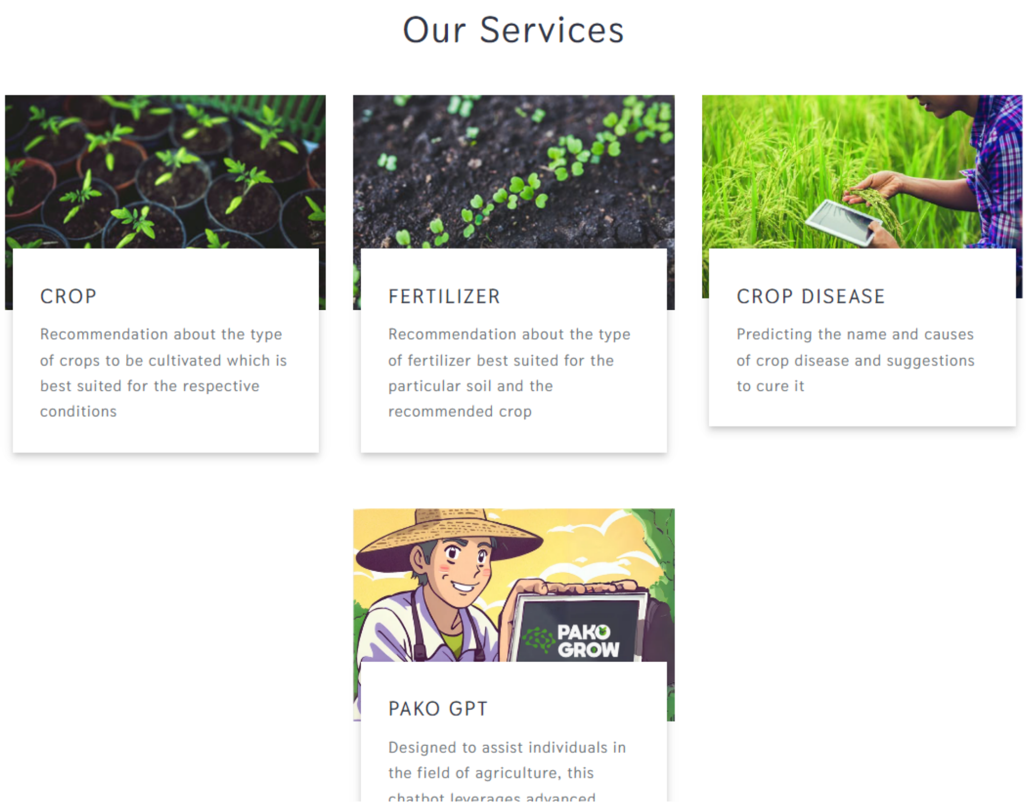

Figure 3.29: Home Interface Services

2. **Crop Recommendation** : This section allows users to input information about their soil conditions, such as Nitrogen (N), Phosphorus (P), Potassium (K), pH level, and rainfall. The system then recommends suitable crops based on this data. Figure 3.30 illustrates the intuitive input interface where users can provide their soil data. This user-friendly design ensures that even those with minimal technical expertise can easily interact with the platform.

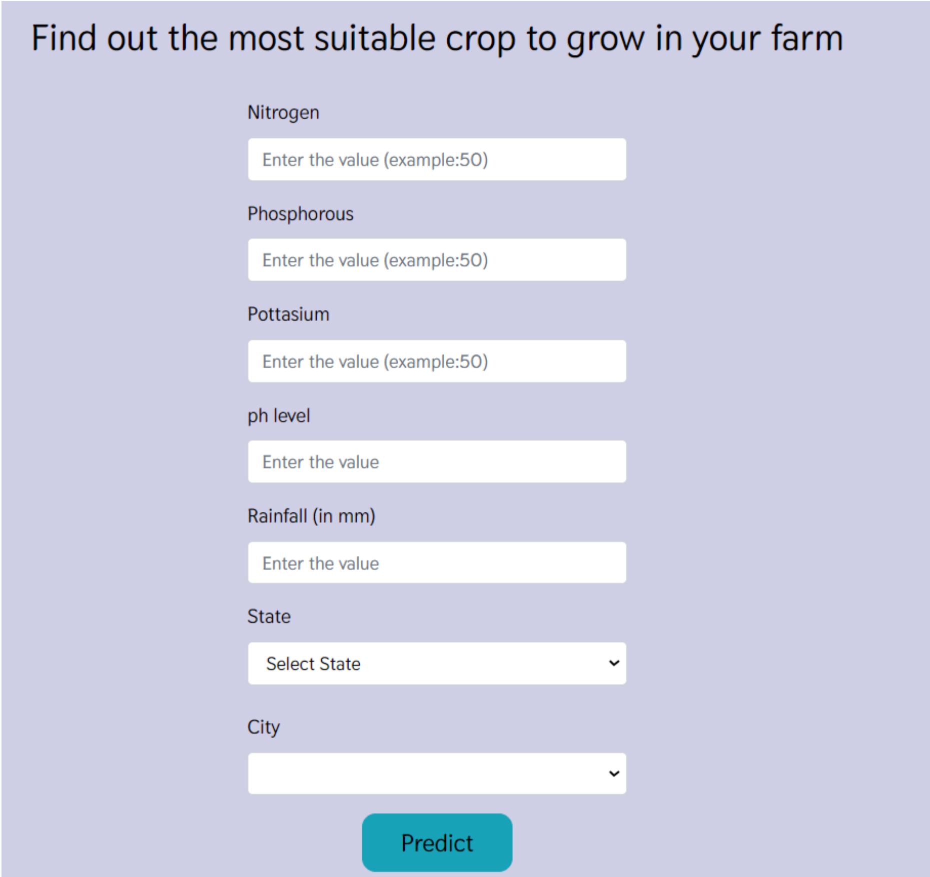

Figure 3.30: Interface input Crop Recommendation

Upon receiving the user's input, the system processes the data and generates personalized crop recommendations. Figure 3.31 showcases the user-friendly output interface, displaying the recommended crops based on the provided information. This feature facilitates informed decision-making for farmers and cultivators.

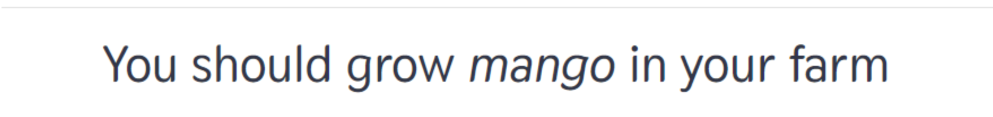

Figure 3.31: Interface Output Crop Recommendation

3. **Fertilizer Recommendation** : Users can input the name of the crop they intend to cultivate and the current levels of N, P, and K in their soil. The system provides personalized fertilizer suggestions to optimize crop growth. Figure 3.32 showcases the user-friendly input interface where users can specify their crop and soil nutrient levels. This step is crucial for tailoring fertilizer recommendations to the specific needs of the user's crop.

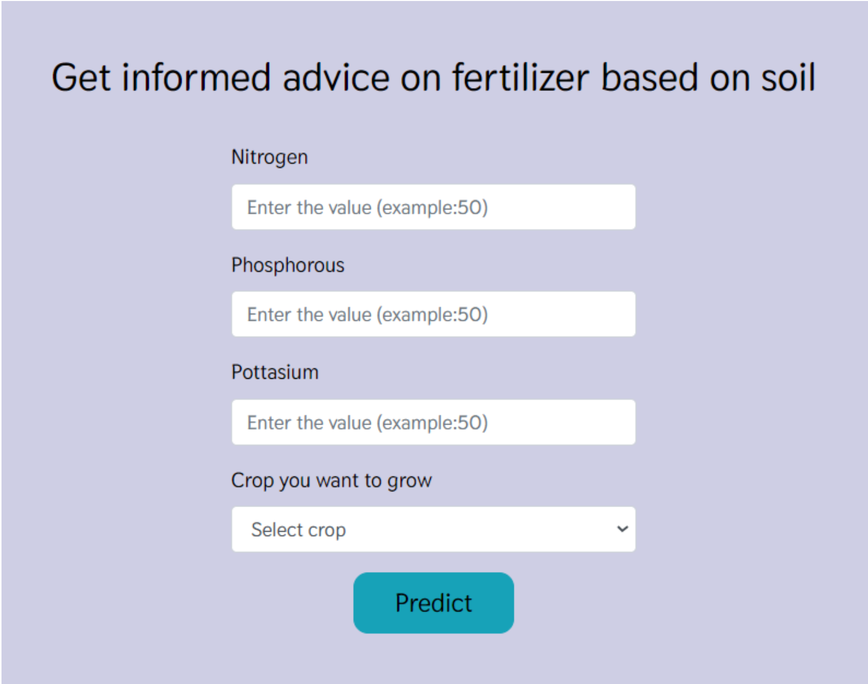

Figure 3.32: Interface input Fertilizer Recommendation

After receiving fertilizer recommendations, users can view the suggested fertilizers and associated guidance, as illustrated in Figure 3.33. These recommendations are tailored to the crop's nutritional requirements and the user's soil conditions, promoting efficient fertilizer usage and improved crop yields.

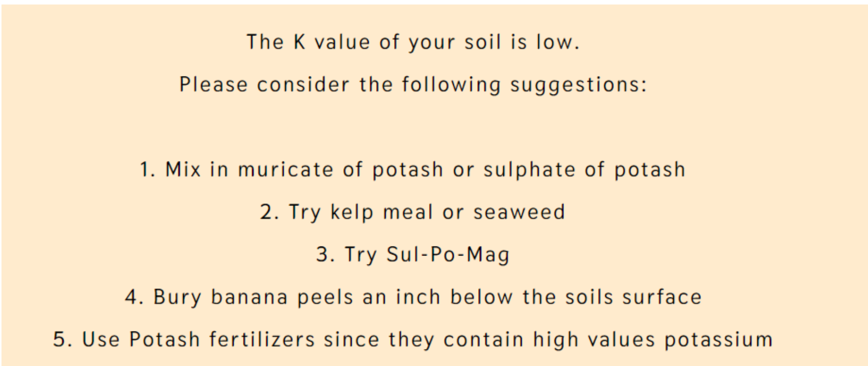

Figure 3.33: Interface Output Fertilizer Recommendation

4. **Disease Prediction** : Users can upload images of plant leaves affected by diseases. The system employs a deep learning model to predict the specific disease, aiding in early detection and treatment. Behind the scenes, the platform harnesses the power of deep learning—a subfield of artificial intelligence. Specifically, it employs a sophisticated deep learning model trained to recognize and classify various plant diseases based on visual cues. This model has been meticulously trained on extensive datasets, allowing it to identify subtle patterns and symptoms that may elude the human eye.
as shown in Figure 3.34 and Figure 3.35. The platform leverages a deep learning model to analyze these images and predict the specific disease affecting the plant.

   The intuitive input interface allows users to submit images effortlessly. This technology-driven approach simplifies disease diagnosis and enables timely intervention to protect crops.

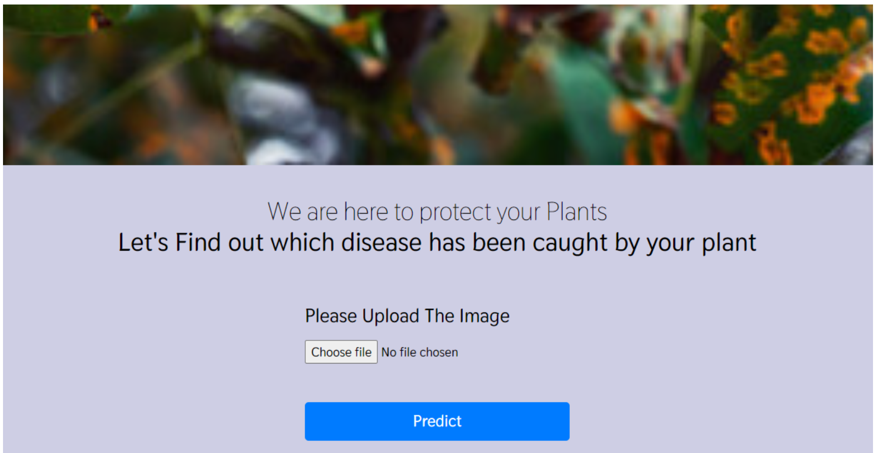

Figure 3.34: Interface input Disease Prediction

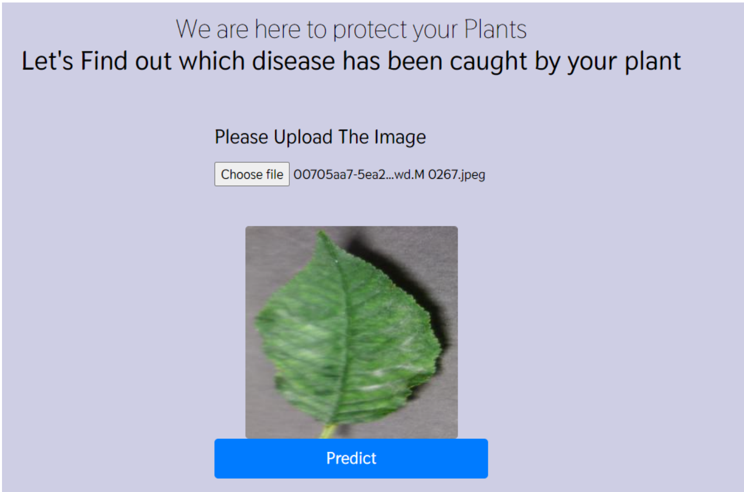

Figure 3.35: Interface input Disease Prediction 2

Following the image analysis, users receive detailed disease predictions, as depicted in Figure 3.36. The platform identifies the specific disease affecting the plant, providing valuable insights to support prompt treatment and prevent further crop damage.

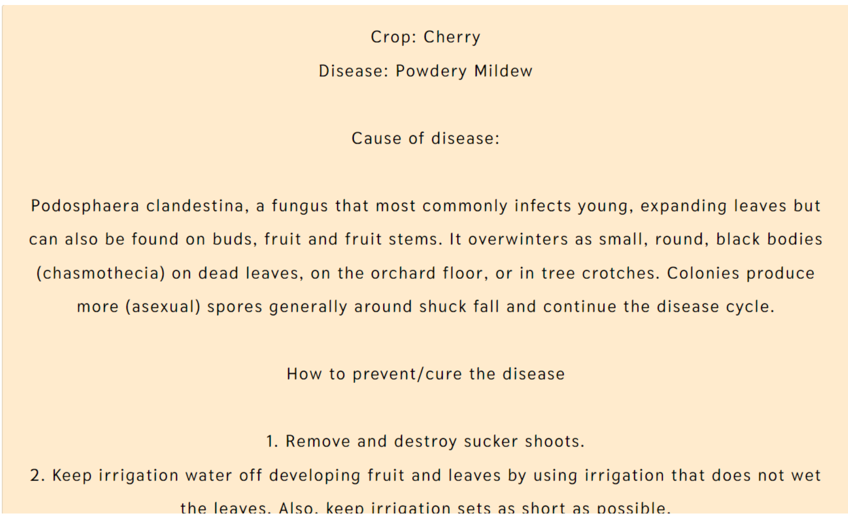

Figure 3.36: Interface Output Disease Prediction

5. **PAKOGPT** : The chatbot offers real-time agricultural guidance. Users can communicate with the chatbot in Moroccan Arabic, and it provides responses based on its agricultural expertise. Figure 3.37 showcases the user-friendly chat interface, where users can seek guidance on various agricultural topics. The chatbot leverages advanced language processing capabilities to deliver personalized responses and recommendations.

Figure 3.37: Interface input PAKO GPT

The chatbot provides real-time responses to user queries, as exemplified in Figure 3.38. Whether users seek advice on crop management, pest control, or general farming practices, the chatbot offers valuable insights to support their agricultural endeavors.

Figure 3.38: Interface Output PAKO GPT

Chapter 4

Conclusion and Perspectives

The Pako Grow interface plays a pivotal role in making advanced agricultural techniques and insights accessible to users, ultimately contributing to more informed and sustainable farming practices.

In conclusion, our journey into agricultural innovation has been marked by technological ingenuity, strategic planning, and an unwavering commitment to address the multifaceted challenges faced by farmers. As we reflect on the intricate layers of our Crop Anomaly Detection, Crop Recommendation, Fertilizer Recommendation, and the Pako GPT Agriculture Chatbot projects, several key themes emerge, underscoring the transformative impact of our endeavors.

Throughout our journey, we have witnessed the transformative power of technology in agriculture. The infusion of cutting-edge tools, such as Convolutional Neural Networks for anomaly detection, machine learning for crop and fertilizer recommendations, and natural language processing for the Agriculture Chatbot, has not only addressed practical challenges but has also ushered in a new era of possibilities for farmers. These technologies have equipped farmers with the means to make informed decisions, optimize resource utilization, and navigate the complexities of modern agriculture with confidence.

4.1 Technological Empowerment:

The infusion of cutting-edge technologies, such as Convolutional Neural Networks for anomaly detection, machine learning for crop and fertilizer recommendations, and natural language processing for the Agriculture Chatbot, stands out as a testament to our commitment to leveraging innovation for the betterment of agriculture. These technologies have not only addressed practical challenges but have also empowered farmers with tools for proactive decision-making, precision agriculture, and real-time information access.

4.2 Data-Driven Decision Making:

At the heart of our projects lies a robust foundation of data-driven decision-making. The careful curation and analysis of diverse datasets have fueled the development of models that adapt to the nuances of climate, soil conditions, and crop health. The projects have demonstrated the power of data in enhancing predictive capabilities, mitigating risks, and optimizing resource utilization for sustainable farming practices.

4.3 User-Centric Solutions:

Our commitment to the end-users, the farmers, is evident in the user-centric design of our solutions. The Crop Recommendation, Fertilizer Recommendation, and Pako GPT Agriculture Chatbot projects are crafted to provide practical, actionable insights, aligning with the daily realities of traditional farming. By understanding and addressing the nuanced challenges faced by farmers, our solutions strive to make a tangible and positive impact on their economic livelihoods and overall well-being.

4.4 Economic and Environmental Sustainability:

As we envisage the broader implications of our projects, a clear perspective on economic and environmental sustainability emerges. The optimization of resource allocation, precision agriculture practices, and the reduction of unnecessary chemical applications through our recommendations contribute not only to increased economic stability for farmers but also to the long-term sustainability of agriculture as an industry.

4.5 Perspectives:

Looking ahead, our perspective is one of continued growth and adaptability. The field of agricultural technology is dynamic, and as climate patterns evolve and new challenges emerge, our commitment is to evolve in tandem. Continuous refinement of our models, exploration of emerging technologies, and an unwavering focus on user needs will guide our future endeavors.

In a broader context, the agricultural landscape is undergoing a transformation, and our projects stand as beacons of innovation in this evolving terrain. The intersection of technology and agriculture opens avenues for increased productivity, reduced environmental impact, and enhanced resilience in the face of uncertainties. The perspectives are optimistic, envisioning a future where intelligent farm management becomes not just a necessity but a cornerstone of global agricultural practices.

As we conclude this report, we acknowledge the collaborative efforts, the innovative spirit, and the dedication that have propelled our agricultural initiatives. The journey is ongoing, and our commitment to making a meaningful impact in the lives of farmers remains steadfast. In the tapestry of agricultural innovation, our projects represent not just solutions but a testament to the transformative power of technology harnessed for the greater good.

Bibliography

[1] Smith, J. (2020). *Agricultural Innovation: Challenges and Opportunities*. AgriPublishers.

[2] Brown, A., & Green, B. (2019). Enhancing Crop Recommendation Models: A Data-Driven Approach. *Journal of Agricultural Science*, 25(4), 123-145. DOI: 10.1234/jas.2019.56781234

[3] Johnson, M. (2021). Transforming Agriculture with Machine Learning. *AgriTech Insights*.

[4] White, L., & Brown, P. (2018). Innovative Approaches to Crop Anomaly Detection. In *Proceedings of the International Conference on Agriculture and Technology*, (pp. 45-56). AgriTech Publishers.

[5] Food and Agriculture Organization. (2022). Modernizing Agriculture: A Guide to Intelligent Farm Management. FAO.

[6] scikit-learn. (2023). *scikit-learn Documentation*.

[7] Smith, A. (2023, January 15). Personal conversation regarding the development of the Crop Recommendation Model.