"Crop Yield Prediction System Using Machine Learning"

A Major Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of
Bachelor of Engineering in Computer Science Engineering

Submitted by:

Kanak Sharma (0827CS201111)
Gourav Sharma (0827CS201083)
Harsh Dad (0827CS201089)
Harsh Jaiswal(0827CS201191)

Guided by:

Prof. Krupi Saraf
Computer Science and Engineering
AITR, Indore

Acropolis Institute of Technology & Research, Indore
Jan - Jun 2024

EXAMINER APPROVAL

The Major Project entitled ***"Crop Yield Prediction System Using Machine Learning"*** submitted by **Kanak Sharma(0827CS201111), Gourav Sharma(0827CS201083), Harsh Dad(0827CS201089), Harsh Jaiswal (0827CS201191)** has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Technology degree in Computer Science Engineering*** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner)	**(External Examiner)**
Date:	**Date:**

RECOMMENDATION

This is to certify that the work embodied in this major project entitled ***"Crop Yield Prediction System Using Machine Learning"*** submitted **Kanak Sharma(0827CS201111), Gourav Sharma(0827CS201083), Harsh Dad(0827CS201089), Harsh Jaiswal (0827CS201191)** is a satisfactory account of the bonafide work done under the supervision of ***Ms. Shraddha Sharma***, is recommended towards partial fulfillment for the award of the Bachelor of Technology (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

(Dean Academics)

STUDENTS UNDERTAKING

This is to certify that the major project entitled ***"Crop Yield Prediction System Using Machine Learning"*** has developed by us under the supervision of ***Ms. Krupi Saraf***. The whole responsibility of the work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Kanak Sharma (0827CS201111)

Gourav Sharma (0827CS201083)

Harsh Dad (0827CS201089)

Harsh Jaiswal (0827CS201191)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this project would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Ms. Krupi Saraf,** Professor, AITR, Indore for her motivation, sagacious guidance, constant encouragement, vigilant supervision, and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr. Kamal Kumar Sethi**, Professor & Head CSE, AITR Indore for his support, suggestion, and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of the department for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

Kanak Sharma(0827CS201111), Gourav Sharma(0827CS201083), Harsh Dad(0827CS201089), Harsh Jaiswal (0827CS201191)

Executive Summary

Crop Yield Prediction

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Computer Science and Engineering branch under the sagacious guidance and vigilant supervision of ***Ms. Krupi Saraf***.

The project A Crop Yield Prediction and Recommendation System is a technology driven solution that aims to assist farmers and agricultural stakeholders in optimizing crop production by predicting crop yields and providing recommendations for improved farming practices. This system typically involves the use of various data sources, including historical agricultural data, weather information, soil data, and sometimes remote sensing data, to make predictions about crop yields. It can also offer insights and guidance on crop selection, planting schedules, irrigation, fertilization, pest control, and other aspects of crop management

Key words: Prediction, Agriculture, Yield

"Today is your opportunity to build the tomorrow you want"

- Ken Poirot

List of Figures

Table of Contents

Chapter 1. Introduction

Introduction

This project predicts crop yields through data-driven analysis of weather, soil quality, and crop history. Using machine learning, the system provides farmers with actionable insights for informed decision-making. By anticipating yield fluctuations, farmers can optimize resource allocation and adapt to changing conditions, ensuring more efficient and resilient crop management.

1.1 Overview

In today's dynamic agricultural landscape, achieving optimal crop yields is not only a matter of traditional expertise but also of harnessing data-driven insights. The intricate interplay between various factors such as weather patterns, soil quality, and crop history significantly influences agricultural outcomes. This project aims to revolutionize crop management by employing advanced data analysis techniques to predict yields accurately. By doing so, it empowers farmers to make well-informed decisions, strategically plan their harvests, and navigate the complexities of modern farming.

- Modern agriculture faces the challenge of feeding a growing global population amidst unpredictable environmental factors.
- This project focuses on using data analysis to predict crop yields by considering weather patterns and crop history.
- The goal is to empower farmers with accurate insights for harvest planning and informed decision-making.
- Traditional yield estimation methods struggle with the complexity of environmental influences, making data-driven approaches crucial. Historical data, weather trends, and soil attributes are analyzed to uncover hidden patterns and enhance yield predictions.

- Machine learning algorithms are employed to process large datasets, identify relationships, and improve yield forecasts.
- The project aims to provide not just predictions but a comprehensive decision support system for farmers.
- By combining historical knowledge with real-time data, farmers can proactively adjust cultivation strategies to changing conditions.
- The outcome is optimized crop productivity, resource efficiency, and a foundation for sustainable agricultural practices. This project bridges traditional farming wisdom with cutting-edge technology to reshape modern agriculture.

1.2 Background and Motivation

Crop yield prediction and recommendation projects have gained significant importance due to several compelling background factors and motivations: Food Security, With the global population steadily increasing, ensuring food security is a top priority. Accurate crop yield predictions and recommendations help optimize agricultural production, ensuring a stable food supply to meet the growing demand. Climate change has brought about increased weather variability, including extreme weather events such as droughts and floods. Predictive models can help farmers adapt to changing weather patterns and mitigate the impact of climate change on crop yields. Resource Efficiency, Agriculture consumes substantial resources, including water and fertilizers. Efficient resource management, guided by data-driven recommendations, can reduce waste and environmental impact while improving crop yields. Economic Viability, Farmers need to maximize their profits and minimize losses. Predictions and recommendations assist in making economically sound decisions regarding crop selection, planting, and harvesting timing. Technological Advancements: Advances in data collection technologies, including satellite imagery, sensors, and IoT devices, have made it easier to gather and analyze agricultural data, providing a solid foundation for predictive and recommendation systems. Access to Information, Many farmers, particularly smallholders, lack access to timely and accurate information. These projects aim to bridge the information gap by delivering personalized recommendations to farmers, even in remote areas. Government Initiatives, Governments and organizations around the world are increasingly investing in precision agriculture and sustainable farming practices. Financial Support, Crop yield prediction and recommendation projects can attract funding and support from various sources, including government grants, research institutions, and private sector partnerships, making them financially viable. Sustainability, Sustainable farming practices are essential for preserving soil health and ecosystem balance. Recommendations often include strategies for sustainable farming, promoting long-term agricultural viability. Improved Livelihoods, By helping farmers increase crop yields and income, these projects contribute to rural development and improved livelihoods for farming communities. In summary, the background and projects are rooted in addressing critical challenges such as food security, climate change, resource efficiency, and economic viability. These projects leverage technology, data, and agricultural expertise to empower farmers and enhance the sustainability and productivity of agriculture.

1.3 Problem Statement and Objectives

In The project aims to achieve the following objectives:

Objective1: Enhanced Harvest Planning: Develop predictive models that empower farmers to anticipate crop yields based on weather patterns, soil quality, and historical data, aiding in effective harvest planning and reducing uncertainties.

Objective 2: Informed Decision-Making: Provide farmers with accurate yield predictions, enabling them to make informed decisions about resource allocation, market planning, and distribution strategies. Optimized Resource Allocation, Offer insights into optimal resource usage, such as water and fertilizers, to help farmers optimize inputs, reduce waste, and improve overall efficiency.

Objective 3: Risk Mitigation: Help farmers proactively address risks associated with adverse weather events, pests, and diseases by suggesting precautionary measures and response strategies.

Objective 4: Time Efficiency: Streamline decision-making processes by providing farmers with readily available and easy-to-understand predictions and recommendations, saving them valuable time. Accessibility, Develop a user-friendly interface that is accessible even to those without extensive technical knowledge, making it convenient for farmers to interact with the system. Knowledge Enrichment: Educate farmers about the underlying factors influencing crop yields, fostering a deeper understanding of agricultural dynamics and promoting informed choices.

1.4 Scope of the Project

The scope of the project "Crop Yield Prediction and Optimization" revolves around leveraging data-driven insights and machine learning techniques to predict crop yields and empower farmers with actionable recommendations. The project encompasses the following components:

- Data Collection and Integration: Gather historical weather data, and comprehensive crop history records from reliable sources. Integrate and preprocess the collected data to create a dataset for analysis.
- Feature Engineering: Identify and engineer relevant features that impact crop yields, considering factors like temperature, precipitation, soil composition, and previous crop performance.
- Machine Learning Models: Develop predictive models using machine learning algorithms, such as regression, decision trees, or neural networks. Train the models on historical data to establish correlations between environmental factors and crop outcomes. Prediction Generation: Utilize trained models to generate accurate crop yield predictions for specific time frames and crop types. Recommendation System: Design an intelligent recommendation

system that translates yield predictions into actionable insights. Provide cultivation recommendations, irrigation strategies, and fertilization schedules to optimize yield potential.

- Real-time Interactivity: Create an intuitive user interface that allows farmers to input real-time weather data. Generate instant predictions and recommendations based on the entered data.
- Validation and Testing: Validate models using historical data that was not used during training. Rigorously test the recommendation system to ensure its accuracy and responsiveness.
- Usability and Feedback: Gather feedback from users, particularly farmers, to refine the user interface and enhance user experience. Incorporate user insights to improve the usability and effectiveness

1.5 Report Structure

The project A Crop Yield Prediction and Recommendation System is a technology driven solution that aims to assist farmers and agricultural stakeholders in optimizing crop production by predicting crop yields and providing recommendations for improved farming practices. This system typically involves the use of various data sources, including historical agricultural data, weather information, soil data, and sometimes remote sensing data, to make predictions about crop yields. It can also offer insights and guidance on crop selection, planting schedules, irrigation, fertilization, pest control, and other aspects of crop management.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2 .Review of Literature

Review of Literature

2.1 Preliminary Investigation

2.1.1 Current System

- CropIn Technology Solutions:

CropIn is an Indian agricultural technology company that offers arrange of digital solutions, including crop monitoring, predictive analytics, and farm management. They provide tools for crop yield prediction and recommendations for Indian farmers.

- Mrida Greens and Organics:

Mrida Greens and Organics is an Indian company that provides precision farming solutions. technology-driven services for soil testing, to help Indian farmers make informed decisions.

- Gramophone:

Gramophone is an agriculture technology startup in India that offers a mobile app and platform It provides crop advisory services, including yield predictions, pest and disease management recommendations, and market insights.

- Fasal:

Fasal is an Indian agtech company that specializes in providing real-time weather, crop, and pest data to farmers. Their platform helps farmers optimize irrigation and improve crop yields.

- AgroStar:

AgroStar is an Indian agtech startup that offers a mobile app for farmers. It provides crop advisory services, including yield prediction, disease management, and weather forecast.

2.2 Limitations of Current System

While the mentioned agricultural technology companies and initiatives in India offer valuable services, they also come with certain limitations and challenges.

- Limited Internet Connectivity: In many rural areas of India, access to high-speed internet is still limited. This can hinder farmers' ability to fully utilize these digital platforms, which often rely on internet connectivity data transfer and updates.

- Data Accuracy: The accuracy of predictions and recommendations depends on the quality of the input data. Inaccurate or incomplete data can lead to unreliable results and advice.
- Hardware Accessibility: Some solutions require specific hardware, such as weather stations or soil sensors, which may not be readily available or affordable for all farmers. Data
- Privacy and Security: Collecting and managing sensitive agricultural data may raise concerns about data privacy and security that their data is protected and not misused.
- Scale and Adoption: Achieving widespread adoption of these technologies among Indian farmers, especially smallholders, can be challenging. Building awareness and demonstrating the value of these solutions is an ongoing process. Integration with
- Traditional Knowledge: traditional farming practices and local knowledge into digital solutions is important. Striking a balance technology and indigenous wisdom can be a challenge.
- Cost: The cost of implementing and maintaining digital agriculture solutions, including purchasing hardware and subscribing to services, can be a barrier for small-scale farmers with limited financial resources.

2.3 Requirement Identification and Analysis for Project

Requirement identification and analysis for a crop yield prediction and recommendation system involves understanding the specific needs and goals of the project. Here's a detailed breakdown of the requirements:

- Stakeholder Identification:

  Identify all stakeholders, including farmers, agricultural experts, government agencies, and technology providers.

- Project Goals and Objectives:

  Define the project's overarching goals, such as improving crop yield and farmer income.

- Data Requirements:

  Determine the types of data needed, such as historical weather data, soil data, crop variety, and market data.

- Technology Infrastructure:

  Identify the required technology infrastructure, including hardware, software, and networking components. Consider cloud-based solutions, databases, and analytics platforms.

- Modeling and Algorithms:

  Define the machine learning or statistical modeling techniques for crop yield prediction. Specify algorithms and methods for generating recommendations, considering factors like model training, validation, and optimization. User Interface and Accessibility Decide how users will interact with the system(web application, mobile app, SMS, etc.).

- Scalability and Performance:

  Analyze scalability requirements based on user base and data volume. metrics and benchmarks to ensure timely responses.

- Security and Privacy:

  Identify security measures to safeguard sensitive data like farmer information and farm data. Comply with data privacy regulations and ensure secure data storage and transmission.

- Feedback Mechanism:

  Implement a feedback system allowing users to provide input and report issues. Use feedback for continuous system improvement.
- Integration with Existing Systems:

  Determine if the system needs to integrate with existing agricultural systems or databases. Ensure compatibility with industry standards and protocols.
- Documentation:

  Develop comprehensive project documentation, including technical documentation, user manuals, and data.
- Testing and Quality Assurance:

  Establish a testing plan to verify reliability and accuracy.
- Sustainability and Maintenance:

  Plan for long-term sustainability, including system maintenance, updates, and future enhancements.
- Ethical Considerations:

  Address ethical concerns related to data ownership, transparency, and fairness in recommendation.

2.3.1 Conclusion

The Crop Yield Prediction and Recommendation represents a significant step forward in modernizing agriculture. By providing farmers with accurate predictions, informed recommendations, and the tools to thrive in a dynamic agricultural landscape, the project paves the way for improved productivity, increased resilience, and a more sustainable approach to feeding the world's growing population. As the agricultural sector embraces data-driven insights, this project stands as a beacon of innovation and a testament to the positive impact technology can have on the lives of farmers.

Chapter 3 .Proposed System

Proposed System

3.1 The Proposal

Crop yield prediction and recommendation system consists of three basic levels:

Project Proposal: Crop Yield Management System We propose the development and implementation of a Crop Yield Prediction to assist farmers in optimizing their crop production and resource management. This system aims to enhance agricultural practices, increase crop yields, and improve the livelihoods of farmers. The project will utilize data analytics, machine learning, and data-driven recommendations to achieve these objectives.

The primary goals of the Crop Yield Management System are as follows:

- Improve crop yield prediction accuracy.
- Provide data-driven recommendations for crop selection, planting schedules, and resource allocation.
- Enhance resource efficiency, reduce input costs, and minimize environmental impact.
- Support sustainable farming practices and rural development.

3.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system:

- Saves time: By using this system lots of time can be saved.
- Easy to use: Itis very easy to use

3.3 Block Diagram

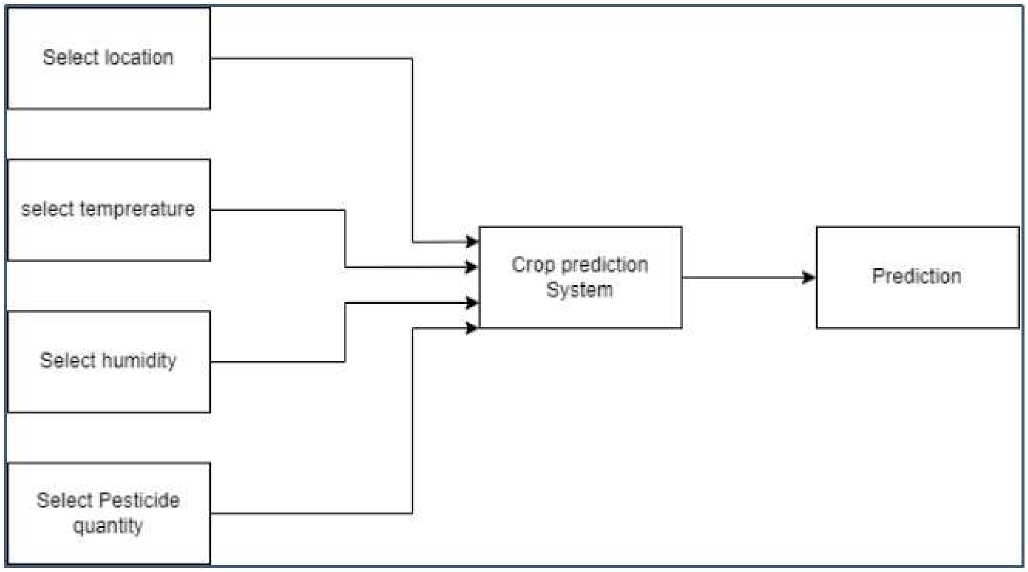


Figure 3-1 : Block Diagram

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

The project relies on advanced data analytics and machine learning to predict crop yields accurately. It features a robust recommendation engine for optimizing resource allocation. The user-friendly web interface ensures accessibility, and seamless integration with existing systems ensures dataflow. Scalability, security, and continuous improvement are key technical priorities.

Prerequisites for this project:

- **Python**: 3.x (we used Python 3.8.8 in this project)
- **Pycharm**: PyCharm is an integrated development environment (IDE) for Python programming. It is developed by JetBrains, a company

known for creating popular development tools for various programming languages. PyCharm is specifically designed to enhance the productivity of Python developers by providing a rich set of features and tools for Python development.

- **Data Availability:** Assess the availability and accessibility of necessary data sources, such as historical agricultural data, weather data, and soil data. Ensure that data quality and reliability meet project requirements.

3.4.2 Economical

Cost Estimation: Calculate the costs associated with developing, implementing, and maintaining the system. This includes software development costs, hardware costs, data acquisition costs, and ongoing operational expenses.

Revenue Generation: Identify potential revenue streams or benefits the system can generate. This might include subscription fees from farmers, partnerships with agricultural organizations, or cost savings for farmers due to increased crop yields.

Return on Investment (ROI): Calculate the ROI by comparing the projected benefits (revenue or cost savings) with the total project costs. Determine the payback period for the initial investment.

Funding Sources: Determine where the funding for the project will come from, whether it's from government grants, private investors, or other sources.

3.4.3 Operational

Operational feasibility assesses whether the system can be effectively operated and into the existing environment. Consider the following:

User Adoption: Evaluate whether farmers and agricultural stakeholders are willing and capable of adopting and using the system. Address any potential resistance to technology adoption.

Training and Support: Plan for training programs and support services to help users navigate and utilize the system effectively.

Maintenance and Updates: Determine how the system will be maintained and updated to address technical issues, incorporate new data sources, and adapt to changing agricultural practices.

Legal and Regulatory Compliance: Ensure that the system complies with relevant laws and regulations, including data privacy, agriculture, and environmental regulations.

3.5 Design Representation

3.5.1 UML Diagrams

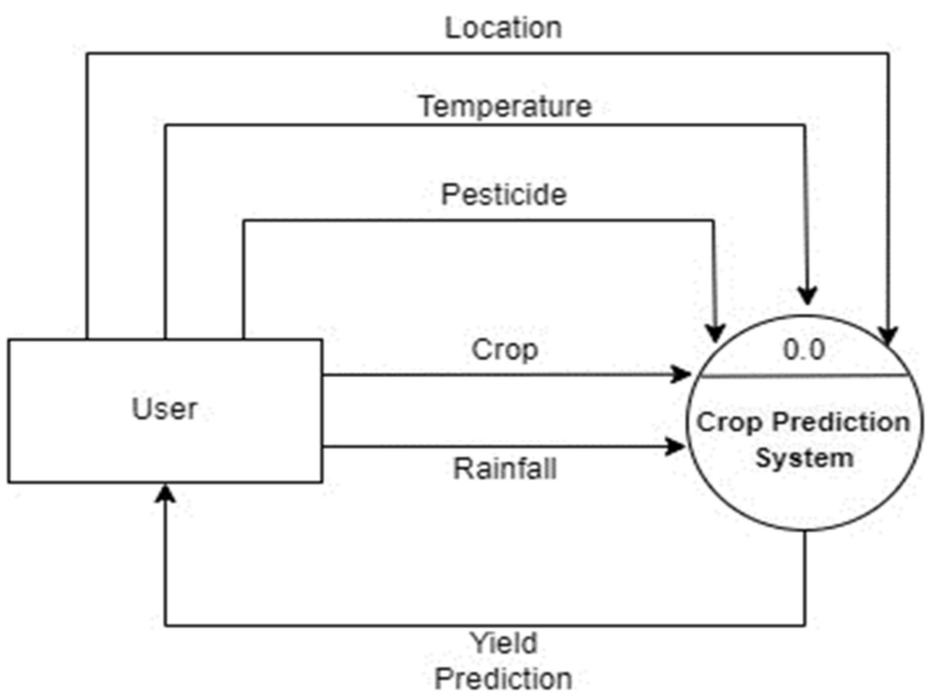


Figure 3-2 : DFD level-0

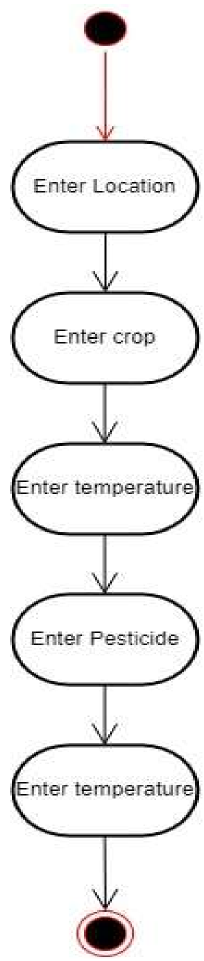


Figure 3-3 : State transition

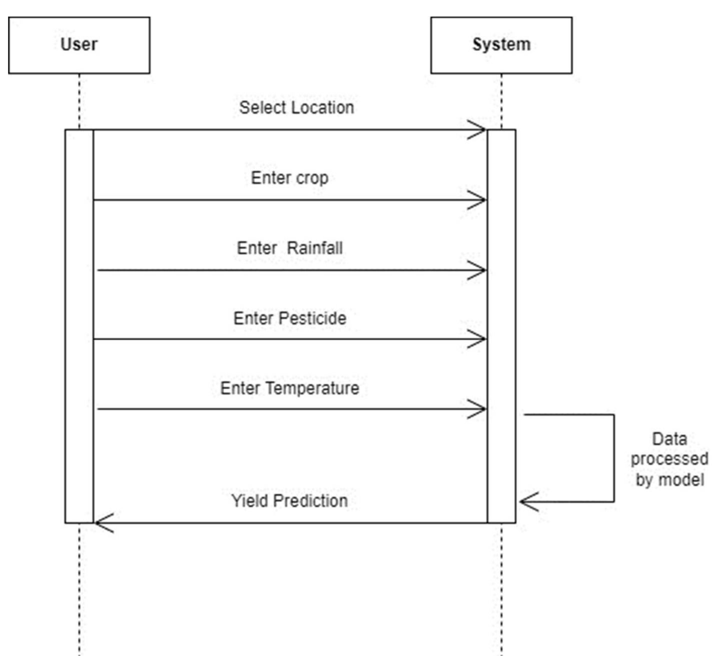


Figure 3-4 : Sequence Diagram

Figure 3-5 : Use Case Diagram

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

3.6.1 Hardware

- System
- 4GB of RAM
- 500GB of Hard disk

3.6.2 Software

- Python Ide
- Library Packages

Chapter 4 .Implementation

Implementation

For the problem of counting the number of students and vehicles entering the college campus manually, the system is designed in such a way so as to automate the process by placing a camera at the entrance gate so that students, bikes and cars getting inside the college campus can be identified and counted.

4.1 Technique Used

In the Crop Yield Management System project, several techniques will be employed to achieve the desired outcomes:

Machine Learning: Machine learning techniques, including regression, decision trees, and neural networks, will be used for crop yield prediction. These models will learn from historical and real-time data to make accurate forecasts.

Data Preprocessing: Data preprocessing techniques, such as data cleaning, feature engineering, and normalization, will be applied to ensure the quality and suitability of the data for analysis.

Recommendation Algorithms: Advanced recommendation algorithms, including collaborative filtering and content-based filtering, will be employed to generate personalized recommendations for farmers regarding crop selection, planting schedules, and resource allocation.

Web Development: Modern web development frameworks and technologies will be utilized to create an intuitive and user-friendly web interface, allowing farmers to access data and recommendations across various devices.

4.2 Tools Used

In the implementation of a Crop Yield Management System, various tools and technologies are used to facilitate data processing, analysis, and system development. Here are some commonly used tools:

- Python: Python is a versatile programming language widely used for data analysis, machine learning, and web development. Libraries like NumPy, Pandas, Scikit-Learn, and Flask are often employed.
- Pycharm: PyCharm is an integrated development environment (IDE)for Python programming. It is developed by JetBrains, a company known for creating popular development tools for various programming languages. PyCharm is specifically designed to enhance the productivity of Python developers by providing a rich set of features and tools for Python development.
- Draw.io: Designed by Seibert Media, draw.io is proprietary software for making diagrams and charts. The software lets you choose from an automatic layout function, or create a custom layout. They have a large selection of shapes and hundreds of visual elements to make your diagram or chart one-of-a-kind. The drag-and-drop feature makes it simple to create a great looking diagram or chart. Draw.io has options for storing saved charts in the cloud, on a server, or network storage at a data center, depending on your needs.

4.3 Language Used

Python language is used in the system due to the following Characterstics :

Simple :

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source :

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

Object Oriented :

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

Extensive Libraries :

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML,

XML-RPC, HTML, WAV files, cryptography, GUI(graphical user interfaces) using Tk, and also other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the "batteries included" philosophy of Python

Chapter 5. Conclusion

Conclusion

5.1 Conclusion

The Crop Yield Prediction and Recommendation represents a significant step forward in modernizing agriculture. By providing farmers with accurate predictions, informed recommendations, and the tools to thrive ina dynamic agricultural landscape, the project paves the way for improved productivity, increased resilience, and a more sustainable approach to feeding the world's growing population. As the agricultural sector embraces data-driven insights, this project stands as a beacon of innovation and a testament to the positive impact technology can have on the lives of farmers.

5.2 Limitations of the Work

- Data Availability and Quality: The accuracy of predictions heavily depends on the quality and availability of historical weather data, soil characteristics, and crop history. Inaccuracies or gaps in data can lead to unreliable predictions.
- Local Variability: Environmental conditions can vary significantly within a small geographic area. Predictive models might struggle to capture such, leading to less accurate predictions for specific regions.
- Changing Climate: Climate change can alter long-standing weather patterns and introduce unpredictability, making historical data less indicative of future conditions.
- Crop Variability: Different crop varieties might respond differently to the same environmental conditions. Developing models that cater to a wide range of crops can be challenging.

- Model Complexity: Complex machine learning models might provide high accuracy, but they can be difficult to interpret. Farmers might hesitate to adopt recommendations they don't understand.
- Dependency on Technology: The system relies on technology infrastructure and reliable internet connectivity, which might not be accessible in all farming regions.
- Human Factors: Human interventions such as pest control, irrigation practices, and soil management can greatly affect crop yields. These factors might not be fully captured in the models.
- Data Lag: Real-time weather data might have a time lag in reporting. Delays in data updates could affect the accuracy of predictions.

5.3 Suggestion and Recommendations for Future Work

In the context of a Crop Yield Management System project, providing suggestions and recommendations for future work is crucial to ensure ongoing improvements and the continued relevance of the system. Here are some suggestions and recommendations for future work:

- Enhanced Data Sources: Explore opportunities to integrate additional data sources, such as satellite imagery, drone data, and real-time sensor data, to further improve the accuracy of crop yield predictions and recommendations.
- Advanced Machine Learning Techniques: Investigate advanced machine learning techniques, including deep learning and reinforcement learning, to develop more sophisticated predictive models and recommendation engines.
- Crop Disease Detection: Extend the system to include crop disease detection capabilities. Implement image recognition models to identify recommendations for mitigating crop diseases.
- User Feedback and Behavior Analysis: Implement mechanisms to collect user feedback and behavior data to continuously refine and personalize recommendations based on preferences and needs.

Bibliography

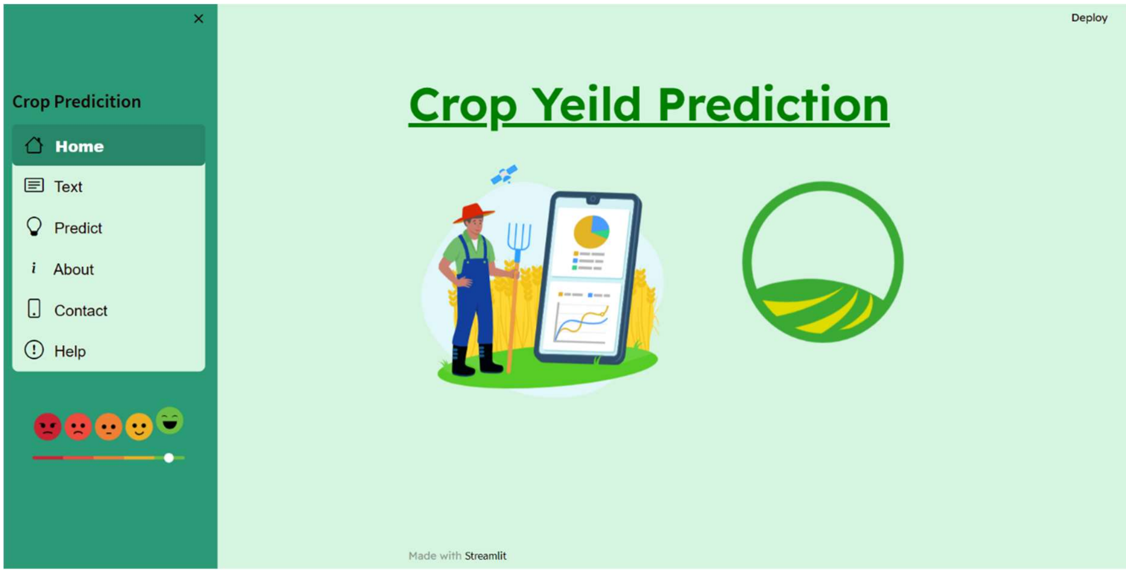
[1] https://www.geeksforgeeks.org/?newui

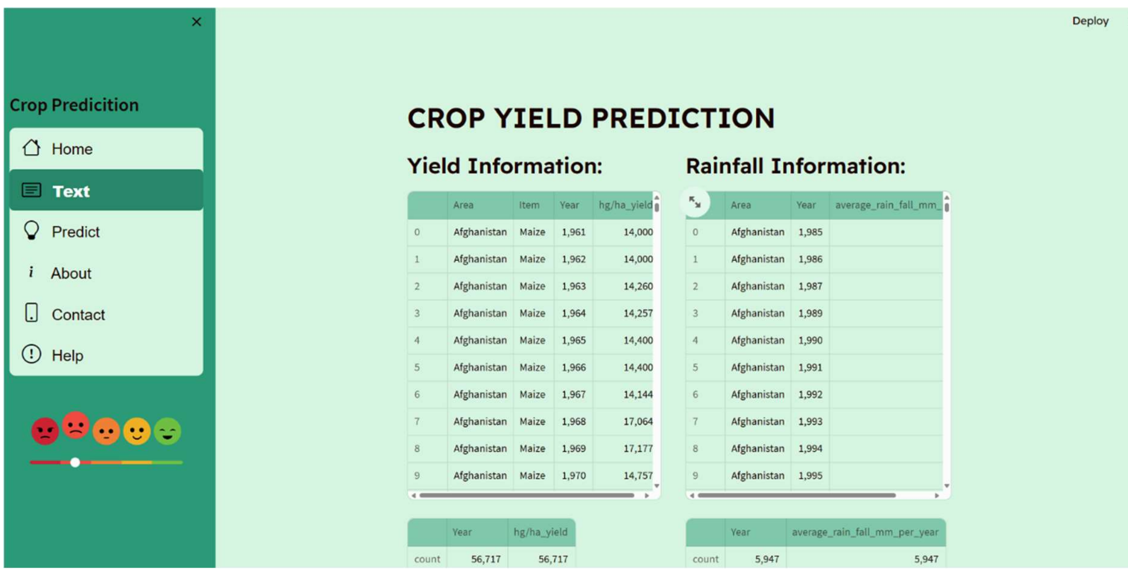
[2] https://www.javatpoint.com/

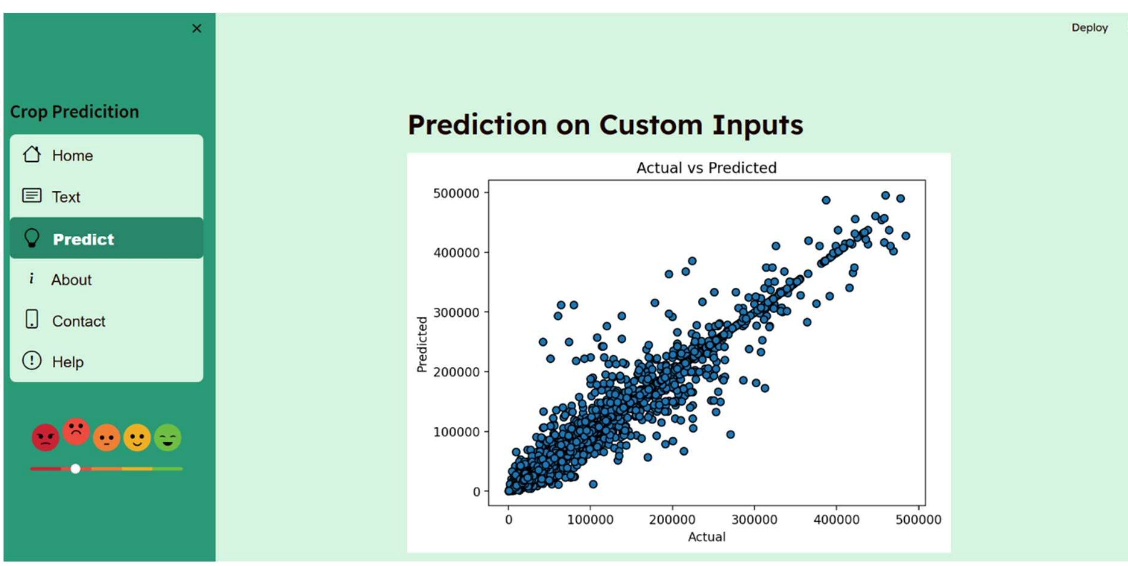
[3] https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer/python/

[4] https://pypi.org/project/opencv-python/

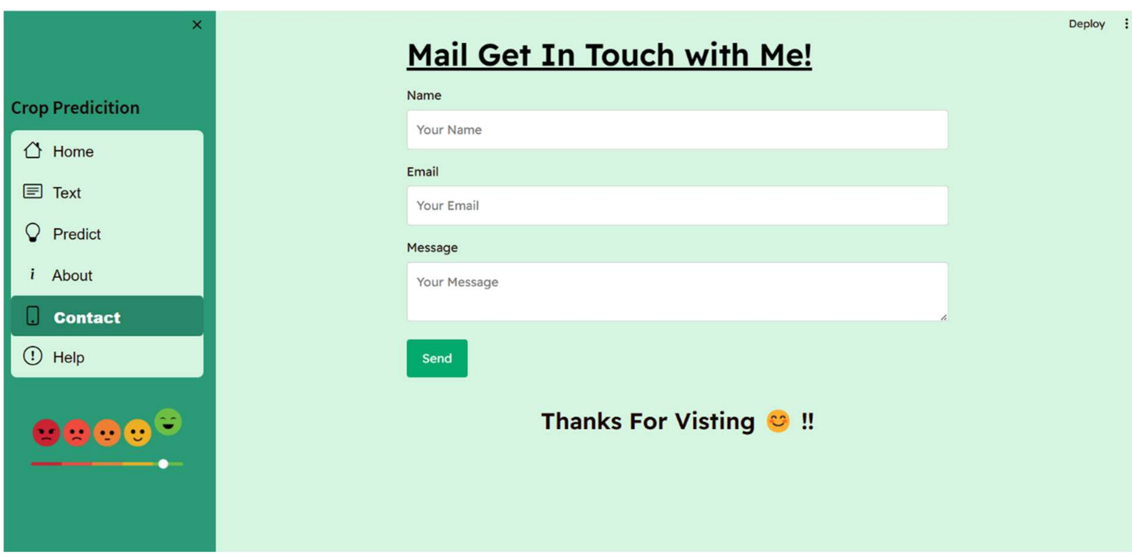
[5] https://www.ibm.com/topics/convolutional-neural-networks/


Deploy
Crop Predicition
Home
Text
Predict
About
Contact
Help
Crop Yeild Prediction
Made with Streamlit


Deploy
Crop Predicition
Home
Text
Predict
About
Contact
Help
CROP YIELD PREDICTION
Yield Information:
Rainfall Information:
Area Item Year hg/ha_yield
0 Afghanistan Maize 1,961 14,000
1 Afghanistan Maize 1,962 14,000
2 Afghanistan Maize 1,963 14,260
3 Afghanistan Maize 1,964 14,257
4 Afghanistan Maize 1,965 14,400
5 Afghanistan Maize 1,966 14,400
6 Afghanistan Maize 1,967 14,144
7 Afghanistan Maize 1,968 17,064
8 Afghanistan Maize 1,969 17,177
9 Afghanistan Maize 1,970 14,757
Area Year average_rain_fall_mm_
0 Afghanistan 1,985
1 Afghanistan 1,986
2 Afghanistan 1,987
3 Afghanistan 1,989
4 Afghanistan 1,990
5 Afghanistan 1,991
6 Afghanistan 1,992
7 Afghanistan 1,993
8 Afghanistan 1,994
9 Afghanistan 1,995
Year hg/ha_yield
count 56,717 56,717
Year average_rain_fall_mm_per_year
count 5,947 5,947


Deploy
Crop Predicition
Home
Text
Predict
About
Contact
Help
Prediction on Custom Inputs
Actual vs Predicted
Predicted
Actual
0
100000
200000
300000
400000
500000


Deploy
Crop Predicition
Home
Text
Predict
About
Contact
Help
Enter The name of the Country
Albania
Enter The name of the Crop
Maize
Rainfall
1100
55
3200
Pesticides
20000
0
1807000
Temprature
16
-14
45
Predicted hg/ha_yield is: 45237.0


Deploy
Crop Predicition
Home
Text
Predict
About
Contact
Help
Mail Get In Touch with Me!
Name
Your Name
Email
Your Email
Message
Your Message
Send
Thanks For Visting 😊 !!

Guide Interaction Sheet

Department of Computer Science and Engineering

Major Project Assessment Criteria-Jan-June 2024

No.	Project Work/metric	Marks allotted	Tentative Timelines & Remarks*
A	**Synopsis**	**10**	**9 Feb, 2024**
	Submission of synopsis	4	
	Synopsis assessment	6	
B	**Synopsis Presentation**	**10**	**9 Feb, 2024**
	Attendance of individual member	3	
	Active participation: Based on engagement in discussions and contributing constructively to team activities.	3	
	PPT assessment	4	
C	**Design Phase (ERD, DFD, Use-case & other UML diagrams)**	**10**	**16 Feb, 2024**
	Submission of designs and diagrams	4	
	Designs and diagrams assessment	6	
D	**Implementation Demo + Report**	**20**	**March 9, 2024**
	Submission of an initial prototype or implementation as per the project timeline	5	
	Attendance of individual member on scheduled presentation	5	
	Implementation assessment	10	
E	**Project based paper**	**10**	**15 March, 2024**
	On time submission of paper approved by guide	3	
	Paper published in conference/journal recommended by guide	7	
F	**Video and Technical poster**	10	**22 March, 2024**
	completion and submission by timeline	4	
	Video and poster assessment	6	
G	**Final Project Submission (Presentation + Demo + Report)**	**20**	**April 12 & 13 ,2024**
	Timely submission of the complete and finalized project.	4	
	Attendance of team members on scheduled submission date	4	
	Active participation: Based on engagement in discussions and contributing constructively to team activities.	4	
	project assessment	8	
H	**Regular attendance: Consistent attendance in project meetings, reviews, and presentations.**	**10**	**Attendance =>70% fetch max**
	Total Marks	**100**	

Source Code

```
from sklearn.preprocessing import LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import OneHotEncoder
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics  import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from PIL import Image
import streamlit as st
import numpy as np
import sklearn
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import svm
from sklearn.tree import DecisionTreeRegressor


#command to execute the code
# streamlit run webapp.py



st.header("""CROP YIELD PREDICTION""");

df_yield = pd.read_csv('/Users/Gourav/Downloads/crop-yield-predection-
main/crop-yield-predection-main/yield.csv')
df_yield = df_yield.drop(['Year Code','Element Code','Element','Year Code','Area
Code','Domain Code','Domain','Unit','Item Code'], axis=1)
df_yield = df_yield.rename(index=str, columns={"Value": "hg/ha_yield"})

df_rain = pd.read_csv('/Users/mayan/Downloads/crop-yield-predection-
main/crop-yield-predection-main/rainfall.csv')
df_rain = df_rain.rename(index=str, columns={" Area": 'Area'})
df_rain['average_rain_fall_mm_per_year'] =
pd.to_numeric(df_rain['average_rain_fall_mm_per_year'],errors = 'coerce')
df_rain = df_rain.dropna()
yield_df = pd.merge(df_yield, df_rain, on=['Year','Area'])
df_pes = pd.read_csv('/Users/Gourav/Downloads/crop-yield-predection-
main/crop-yield-predection-main/pesticides.csv')
df_pes = df_pes.rename(index=str, columns={"Value": "pesticides_tonnes"})
df_pes = df_pes.drop(['Element','Domain','Unit','Item'], axis=1)
```

```
yield_df = pd.merge(yield_df, df_pes, on=['Year','Area'])

avg_temp=  pd.read_csv('/Users/Gourav/Downloads/crop-yield-predection-
main/crop-yield-predection-main/temp.csv')

avg_temp = avg_temp.rename(index=str, columns={"year": "Year",
"country":'Area'})

yield_df = pd.merge(yield_df,avg_temp, on=['Area','Year'])


yield_df.groupby('Item').count()
yield_df['Area'].nunique()
yield_df.groupby(['Area'],sort=True)['hg/ha_yield'].sum().nlargest(10)
yield_df.groupby(['Item','Area'],sort=True)['hg/ha_yield'].sum().nlargest(10)
yield_df.to_csv()


tab1, tab2, tab3=st.tabs(["Home","Datasets","More Info"])
# with tab1:
#    st.header("Home Page")
with tab2:
   st.header("Datasets")
   # set a header
   st.subheader("Yield Information:")

   # show the data
   st.dataframe(df_yield)

   # showing the dataset of the data
   st.write(df_yield.describe())

   # set a header
   st.subheader("Rainfall Information:")

   # show the data
   st.dataframe(df_rain)

   # showing the dataset of the data
   st.write(df_rain.describe())
   st.subheader("Pesticide Information:")

   #show the data
   st.dataframe(df_pes)

   #showing the dataset of the data
   st.write(df_pes.describe())
   st.subheader("Temperature Information:")
```

```
    # show the data
    st.dataframe(avg_temp)

    # showing the dataset of the data
    st.write(avg_temp.describe())

    st.write(avg_temp.info())

    st.header("The final Dataframe")

    st.dataframe(yield_df)

    st.subheader("Information regarfing the dataframe:")

    # showing the dataset of the data
    st.write(yield_df.describe())
with tab3:
    st.header("More Info")
    correlation_data = yield_df.select_dtypes(include=[np.number]).corr()

    mask = np.zeros_like(correlation_data, dtype=np.bool_)
    mask[np.triu_indices_from(mask)] = True

    f, ax = plt.subplots(figsize=(11, 9))
    # Generate a custom diverging colormap
    cmap = sns.palette = "vlag"

    # Draw the heatmap with the mask and correct aspect ratio
    sns.heatmap(correlation_data, mask=mask, cmap=cmap, vmax=.3, center=0,
square=True, linewidths=.5,
            cbar_kws={"shrink": .5});
    st.write(f)
    st.write(
        "It can be seen from the above correlation map that there is no correlation
between any of the colmuns in the dataframe.")

st.markdown("[Jump To Results](#prediction-on-custom-
inputs)",unsafe_allow_html=True)

#get the data
# df_yield = pd.read_csv('/Users/Gourav/Downloads/crop-yield-predection-
main/crop-yield-predection-main/yield.csv')
# df_yield = df_yield.drop(['Year Code','Element Code','Element','Year Code','Area
Code','Domain Code','Domain','Unit','Item Code'], axis=1)
# df_yield = df_yield.rename(index=str, columns={"Value": "hg/ha_yield"})
# df_yield = df_yield.drop(['Year Code', 'Element Code', 'Element', 'Year Code',
'Area Code', 'Domain Code', 'Domain', 'Unit', 'Item Code'], axis=1)
# #set a header
```

```
# st.subheader("Yield Information:")

#
# #show the data
# st.dataframe(df_yield)
#
# #showing the dataset of the data
# st.write(df_yield.describe())

# st.write(df_yield.info())


###rainfall data

#get the data
# df_rain = pd.read_csv('/Users/Gourav/Downloads/crop-yield-predection-
main/crop-yield-predection-main/rainfall.csv')
# df_rain = df_rain.rename(index=str, columns={" Area": 'Area'})
# df_rain['average_rain_fall_mm_per_year'] =
pd.to_numeric(df_rain['average_rain_fall_mm_per_year'],errors = 'coerce')
# df_rain = df_rain.dropna()

#set a header
# st.subheader("Rainfall Information:")
#
# #show the data
# st.dataframe(df_rain)
#
# #showing the dataset of the data
# st.write(df_rain.describe())

# st.write(df_rain.info())


#merging the two dataframes
# yield_df = pd.merge(df_yield, df_rain, on=['Year','Area'])


###pesticides data

#get the data

# df_pes = pd.read_csv('/Users/mayan/Downloads/crop-yield-predection-
main/crop-yield-predection-main/pesticides.csv')
# df_pes = df_pes.rename(index=str, columns={"Value": "pesticides_tonnes"})
# df_pes = df_pes.drop(['Element','Domain','Unit','Item'], axis=1)
```

```
#set a header
# st.subheader("Pesticide Information:")
#
# #show the data
# st.dataframe(df_pes)
#
# #showing the dataset of the data
# st.write(df_pes.describe())

# st.write(df_pes.info())

#merging the two dataframes
# yield_df = pd.merge(yield_df, df_pes, on=['Year','Area'])


###avg temp dataset
# avg_temp=  pd.read_csv('/Users/Gourav/Downloads/crop-yield-predection-
main/crop-yield-predection-main/temp.csv')
#
# avg_temp = avg_temp.rename(index=str, columns={"year": "Year",
"country":'Area'})

#set a header
# st.subheader("Temperature Information:")
#
# #show the data
# st.dataframe(avg_temp)
#
# #showing the dataset of the data
# st.write(avg_temp.describe())
#
# st.write(avg_temp.info())

##merging the two datasets
# yield_df = pd.merge(yield_df,avg_temp, on=['Area','Year'])


####The final dataframe
# yield_df.groupby('Item').count()
# yield_df['Area'].nunique()
# yield_df.groupby(['Area'],sort=True)['hg/ha_yield'].sum().nlargest(10)
# yield_df.groupby(['Item','Area'],sort=True)['hg/ha_yield'].sum().nlargest(10)
# yield_df.to_csv()

# st.header("The final Dataframe")
#
# st.dataframe(yield_df)
# st.subheader("Information regarfing the dataframe:")
```

```
# #showing the dataset of the data
# st.write(yield_df.describe())

# st.write(yield_df.info())



###Time for data exploration, finding whether there's any correlation between
the variables

# correlation_data=yield_df.select_dtypes(include=[np.number]).corr()
#
# mask = np.zeros_like(correlation_data, dtype=np.bool_)
# mask[np.triu_indices_from(mask)] = True
#
# f, ax = plt.subplots(figsize=(11, 9))
# # Generate a custom diverging colormap
# cmap = sns.palette="vlag"
#
# # Draw the heatmap with the mask and correct aspect ratio
# sns.heatmap(correlation_data, mask=mask, cmap=cmap, vmax=.3,
center=0,square=True, linewidths=.5, cbar_kws={"shrink": .5});
# st.write(f)
# st.write("It can be seen from the above correlation map that there is no
correlation between any of the colmuns in the dataframe.")

with tab3:
  # Data Preprocessing
  st.header("Data Preprocessing")
  st.write(
    "Data Preprocessing is a technique that is used to convert the raw data into a
clean data set. In other words, whenever the data is gathered from different
sources it is collected in raw format which is not feasible for the analysis.")

  st.subheader("Encoding Categorical Variables:")
  st.write('''There are two categorical columns in the dataframe, categorical data
are variables that contain label values rather than numeric values. The number of
possible values is often limited to a fixed set, like in this case, items and countries
values. Many machine learning algorithms cannot operate on label data directly.
They require all input variables and output variables to be numeric.

  This means that categorical data must be converted to a numerical form. One
hot encoding is a process by which categorical variables are converted into a
form that could be provided to ML algorithms to do a better job in prediction. For
that purpose, One-Hot Encoding will be used to convert these two columns to
one-hot numeric array.
  The categorical value represents the numerical value of the entry in the
dataset. This encoding will create a binary column for each category and returns
```

```
a matrix with the results.''')


    features = yield_df.loc[:, yield_df.columns != 'hg/ha_yield']
    features = features.drop(['Year'], axis=1)
    label = yield_df['hg/ha_yield']
    ct1 = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1])],
remainder='passthrough')
    features = np.array(ct1.fit_transform(features))
    le = LabelEncoder()
    features[:, 10] = le.fit_transform(features[:, 10])
    yield_df_onehot = pd.DataFrame(features)
    yield_df_onehot["hg/ha_yield"] = label

    st.subheader("One-Hot Encoded data:")
    st.dataframe(yield_df_onehot.head())
    st.write(yield_df_onehot.info())

    # scaling the data
    st.subheader("Scaling Features:")
    st.write('''Taking a look at the dataset above, it contains features highly varying
in magnitudes, units and range. The features with high magnitudes will weigh in
a lot more in the distance calculations than features with low magnitudes.

    To supress this effect, we need to bring all features to the same level of
magnitudes. This can be acheived by scaling.''')

    scaler = MinMaxScaler()
    features = scaler.fit_transform(features)

    st.write(features)

    # train test split dataset

    st.header("Training Data")

    st.write("""For training of the model, we use majority of the data to training the
model and the rest of the data is used for testing the model. Here, we are using
80% of the data for training and 20% of the data for testing.

    The training dataset is the intial dataset used to train ML algorithm to learn and
produce right predictions. (80% of dataset is training dataset)

    The test dataset, however, is used to assess how well ML algorithm is trained
with the training dataset.
    """)
    train_data, test_data, train_labels, test_labels = train_test_split(features, label,
test_size=0.2, random_state=42)
```

```
# Model comparing, training and selection
st.header("Model Comparison and Selection")

st.write("""Here we test 3 different regressor algorithms namely:
1. Gradient Boosting Regressor
2. Random Forest Regressor
3. Decision Tree Regressor
""")

# not printing all of this as it takes too much time to run, rather just pasting the
output as obtained on the kaggle notebook

# models = [
#     GradientBoostingRegressor(n_estimators=200, max_depth=3,
random_state=0),
#     RandomForestRegressor(n_estimators=200, max_depth=3,
random_state=0),
#     svm.SVR(),
#     DecisionTreeRegressor()
# ]
# model_train=list(map(compare_models,models))

# print(*model_train, sep="\n") The evaluation metric is set based on R^2
(coefficient of determination) regression score function, that will represents the
proportion of the variance for items (crops) in the regression model. R^2 score
shows how well terms (data points) fit a curve or line.
st.write(
    """R ^ 2 is a statistical measure between 0 and 1 which calculates how
similar a regression line is to the data it's fitted to. If it's a 1, the model 100 %
predicts the data variance if it's a 0, the model predicts none of the variance.""")

st.write("")

st.subheader("Testing of the model on test dataset on kaggle notebook")

st.write("""The output below shows the accuracy of the model, the accuracy is
the percentage of the correct predictions.

['GradientBoostingRegressor', 0.8959545600619471]

['RandomForestRegressor', 0.6807690552605921]

['DecisionTreeRegressor', 0.9605155680634376]


Therefore, from the above output, we can see that the Decision Tree Regressor
is the best model to use for this dataset.""")
```

```
    test_df = pd.DataFrame(test_data, columns=yield_df_onehot.loc[:,
yield_df_onehot.columns != 'hg/ha_yield'].columns)

    st.write("The final dataset is ready to be used for the model. The final dataset
contains the following columns:")
    st.write(test_df.head())

#prediction
with tab1:
    st.header("Prediction")
    st.write("""As seen above, the best regressor is Decision Tree Regressor. So, we
will use this regressor to predict the yield of the crops.

    Using the same, the plot is made here using MatPlotLib showing the relation
between the predicted and the actual yield.""")
    st.success("The model predicts the results with an accuracy of  96.051%")
    clf = DecisionTreeRegressor()
    model = clf.fit(train_data, train_labels)

    test_df["yield_predicted"] = model.predict(test_data)
    test_df["yield_actual"] = pd.DataFrame(test_labels)["hg/ha_yield"].tolist()

    fig, ax = plt.subplots()

    ax.scatter(test_df["yield_actual"], test_df["yield_predicted"], edgecolors=(0, 0,
0))

    ax.set_xlabel('Actual')
    ax.set_ylabel('Predicted')
    ax.set_title("Actual vs Predicted")
    plt.show()
    st.write(fig)

    st.header("Prediction on Custom Inputs")

    data = pd.read_csv("/Users/mayan/Downloads/crop-yield-predection-
main/crop-yield-predection-main/yield_df.csv")
    data.columns =
["","Area","Item","Year","hg/ha_yield","average_rain_fall_mm_per_year","pesticid
es_tonnes","avg_temp"]
    rawlist = list(data.Area)
    rawlist2 = list(data.Item)
    #get unique elements
    area_list = []
    item_list = []
    for x in rawlist2:
        if x not in item_list:
```

```
        item_list.append(x)
    for x in rawlist:
        if x not in area_list:
            area_list.append(x)


    def getUserInput():


#Area,Item,Year,hg/ha_yield,average_rain_fall_mm_per_year,pesticides_tonnes,av
g_temp
        area_ip=st.sidebar.selectbox("Enter The name of the Country",area_list)
        item_ip = st.sidebar.selectbox("Enter The name of the Crop", item_list)
        rainfall_ip=st.sidebar.slider('Rainfall',55,3200,1100)
        pesticides_ip=st.sidebar.slider('Pesticides',0,1807000,20000)
        avg_temp_ip=st.sidebar.slider('Temprature',-14,45,16)

        userData=np.array([[area_ip,item_ip,rainfall_ip,pesticides_ip,avg_temp_ip]])
        return userData

    inputs=getUserInput()
    inputs=np.array(ct1.transform(inputs))
    inputs[:,10]=le.transform(inputs[:,10])
    inputs=scaler.transform(inputs)
    prediction = model.predict(inputs)

    predic_y = "Predicted hg/ha_yield is: " + str(prediction[0])
    st.success(predic_y)
```