Predicting Lumpy Skin Disease

Bhavya Sharma

Sanjay Haritwal

Devansh Saxena

Project Report Format

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

5.2 Solution Architecture

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

6.2 Sprint Planning & Estimation

6.3 Sprint Delivery Schedule

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

Predicting Lumpy Skin Disease

Bhavya Sharma

Sanjay Haritwal

Devansh Saxena

Introduction

Project Overview

Lumpy skin disease virus (LSDV) causes an infectious disease in cattle. Due to its direct relationship with the survival of arthropod vectors, geospatial and climatic features play a vital role in the epidemiology of the disease. The objective of this study was to assess the ability of some machine learning algorithms to forecast the occurrence of LSDV infection based on meteorological and geological attributes. Initially, ExtraTreesClassifier algorithm was used to select the important predictive features in forecasting the disease occurrence in unseen (test) data among meteorological, animal population density, dominant land cover, and elevation attributes. Some machine learning techniques revealed high accuracy in predicting the LSDV occurrence in test data (up to 97%). In terms of area under curve (AUC) and F1 performance metric scores, the artificial neural network (ANN) algorithm outperformed other machine learning methods in predicting the occurrence of LSDV infection in unseen data with the corresponding values of 0.97 and 0.94, respectively. Using this algorithm, the model consisted of all predictive features and the one which only included meteorological attributes as important features showed similar predictive performance.

Purpose

According to the findings of this research, ANN can be used to forecast the occurrence of LSDV infection with high precision using geospatial and meteorological parameters. Applying the forecasting power of these methods could be a great help in conducting screening and awareness programs, as well as taking preventive measures like vaccination in areas where the occurrence of LSDV infection is a high risk.

Literature Survey

Existing Problem

Lumpy skin disease virus (LSDV) infection is a major challenge to cattle production, causing acute or subacute disease in cattle and water buffalo population. Cattle of all breeds can become infected, and cows that are around the peak of milk production and calves are particularly susceptible to LSDV infection (Namazi and Khodakaram Tafti 2021).

The LSDV is a double-stranded DNA virus belonging to the *Capripoxvirus* genus. Fever, inappetence, a significant drop in milk production, swollen lymph nodes, and the appearance of hard, slightly elevated skin nodules quickly after the onset of fever are the main clinical signs of the infection. Despite the availability of a variety of diagnostic tests, the diagnosis is generally confirmed using a traditional or real-time PCR (polymerase chain reaction) approach.

References

In 1929, the first case of LSDV infection was recorded in Zambia (Von Backstrom 1945). LSDV has gradually expanded through Africa, the Middle East, Southeastern Europe, Central Asia, and, most recently, South Asia and China. The disease is now endemic in many African countries, as well as areas of the Middle East (Iraq, Saudi Arabia, and the Syrian Arab Republic) and Turkey (Roche et al. 2020). The disease has resulted in major economic losses in the affected countries. Due to high fever and secondary mastitis, it causes a substantial drop in milk production. Other consequences of the disease include damaged skin, a reduction in the growth rate of beef cattle, transient or lifelong infertility, abortion, treatment and vaccination costs, and the mortality in infected animals (Alemayehu et al. 2013; Namazi and Khodakaram Tafti 2021).

LSDV is transmitted by insects, in particular blood-sucking arthropods, contaminated food and drink, and at the later stages of the disease through saliva, nasal secretions, and semen (Sprygin et al. 2018; Tuppurainen et al. 2017). Due to its direct relationship with the survival of vectors, climatic conditions play an important role in the epidemiology of the disease. A warm and humid climate, environmental conditions that support an influx of vector populations, such as those seen during seasonal rains, and the introduction of new animals to a herd are all risk factors for the spread of LSDV. Furthermore, the wind's direction and intensity may play a role in the spread of the virus (Chihota et al. 2003).

The association between LSDV infection and meteorological and geospatial factors has been studied in many studies, and they have discovered that factors like temperature, precipitation, land cover, humidity, and wind speed can predict or influence the occurrence of the disease (Alkhamis and VanderWaal 2016; Allepuz et al. 2019; Machado et al. 2019; Molla et al. 2017; Sprygin et al. 2018; Tuppurainen and Oura 2012).

Due to the introduction of new technologies and analytical techniques such as big data, remote sensing, and Earth observation, many digital Earth researches are now employing big spatiotemporal data to track and define the dynamic Earth climate system, (Kovacs-Györi et al. 2020; Yang et al. 2017).

Nowadays, machine learning (ML) offers highly valuable resources for intelligent geospatial and environmental data analysis, synthesis, and visualization. ML methods, particularly deep learning approaches, have become more common as the availability of more and different types of big data has grown (Xu and Jackson 2019). These techniques use general purpose learning algorithms to look for similarities in often complex and unwieldy data (Bzdok et al. 2018). In general, they can be used effectively at all levels of environmental data mining: exploratory spatial data processing, identification and modeling of spatial–temporal patterns, and decision-driven mapping. Traditional geostatistical methods have been replaced greatly by machine learning techniques especially in big data analyses (Kanevski et al. 2008). However, ML techniques should be implemented accurately and effectively from pre-processing data to analysis and justification of the findings (Kanevski et al. 2008).

ML techniques have been evaluated in several studies for predicting the occurrence of infectious diseases in human or animals using various climatic and geospatial features.

Wang et al. (2015) developed a feed-forward back-propagation neural network model to predict the weekly number of human cases of infectious diarrhea in China (Shanghai) using meteorological factors as predictive features. Non-linear models including neural networks, support vector regression, and random forests regression showed better performance than multiple linear regression. Neural networks showed most satisfactory results when all performance evaluation criteria were considered simultaneously.

Malki et al. (2020) explored various regressor machine learning models to predict confirmed and death cases of COVID-19 in various countries. In forecasting the COVID-19 confirmed cases, the highest performance was obtained by the KNN (K-nearest neighbors) regressor. Decision tree algorithm showed best performance in predicting the rate of COVID-19 mortality. Weather variables such as temperature and humidity were more important in predicting the mortality rate when compared to the other census variables such as population, age, and urbanization.

Golden et al. (2019) collected soil and feces samples from 11 pastured poultry farms from 2014 to 2017 in the USA. They generated random forest and gradient boosting machine predictive models to predict *Listeria* spp. prevalence in samples based on meteorological factors such as temperature, wind speed, gust speed, humidity, and precipitation at the farming location. AUC performance metric for the random forest and gradient boosting machine models of fecal samples was 0.905 and 0.855, respectively. The soil gradient boosting machine model outperformed the random forest model with AUCs of 0.873 and 0.700, respectively.

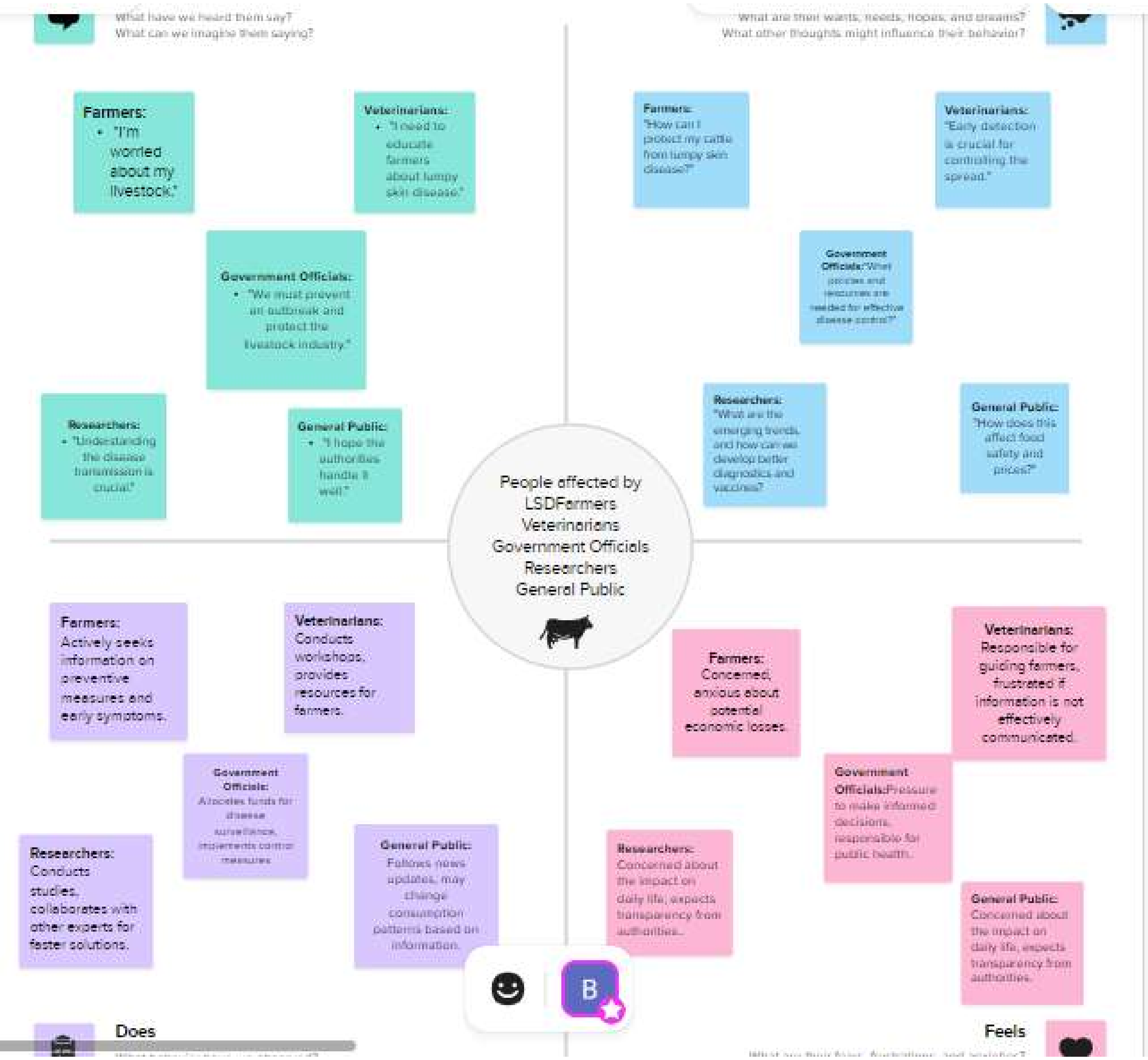
Liang et al. (2020) used machine learning methods to forecast African swine fever outbreaks around the world using bio-climatic variables. The random forest algorithm outperformed other techniques with 80.4% accuracy in the dataset containing all predictive variables, and the support vector machine algorithm showed the best accuracy in the subset dataset containing only important climatic features (76.02%).

The accuracy score of prediction varied between 47.8 and 99.6% in the study by Niu et al. (2020), which used various machine learning algorithms to forecast Peste des Petits ruminants (PPR) outbreaks based on certain bio-climatic variables and altitude data. The random forest algorithm performed best in a test dataset consisting of data from three countries that were not included in the training process.

Problem Statement Definition

Because of the importance of insects in LSDV transmission and their reliance on climatic and geographical features, the key objective of this research was to develop predictive models using some robust ML algorithms based on meteorological and geospatial features to predict the incidence of LSDV infection in countries with a prior history of disease outbreak reported between 2011 and 2021.

Empathy Map

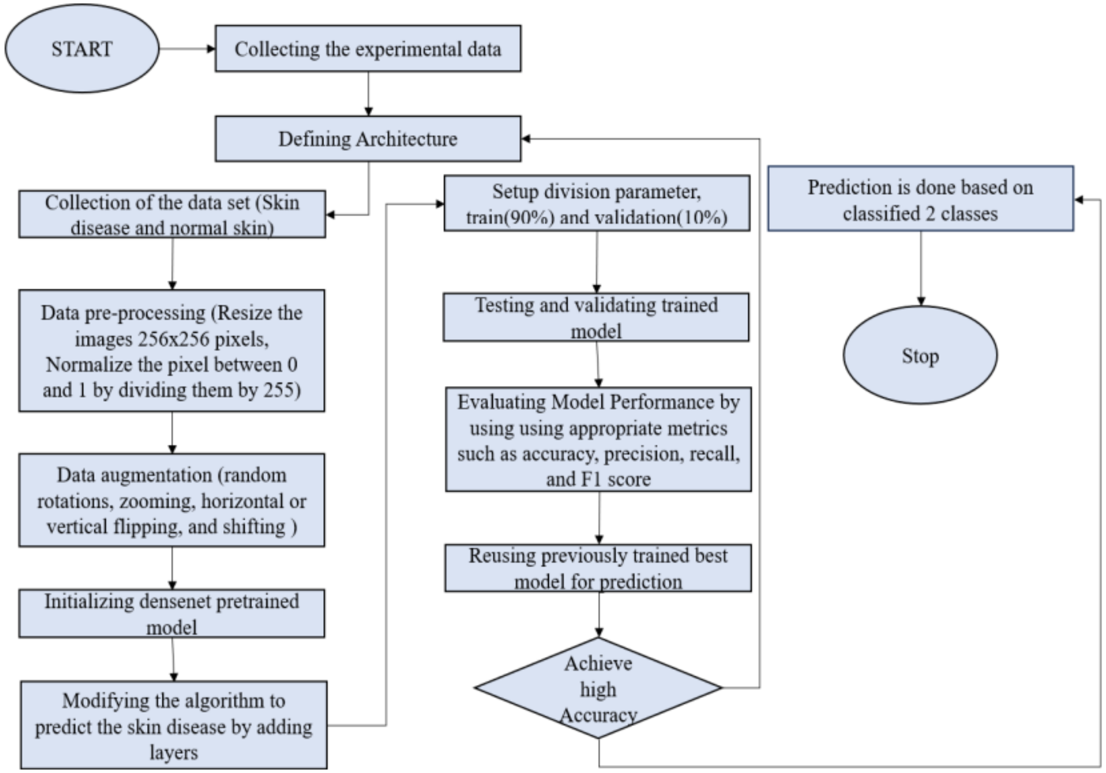

What have we heard them say?
What can we imagine them saying?
Farmers:
"I'm worried about my livestock."
Veterinarians:
"I need to educate farmers about lumpy skin disease."
Government Officials:
"We must prevent an outbreak and protect the livestock industry."
Researchers:
"Understanding the disease transmission is crucial."
General Public:
"I hope the authorities handle it well."
What are their wants, needs, hopes, and dreams?
What other thoughts might influence their behavior?
Farmers:
"How can I protect my cattle from lumpy skin disease?"
Veterinarians:
"Early detection is crucial for controlling the spread."
Government Officials:"What policies and resources are needed for effective disease control?"
Researchers:
"What are the emerging trends, and how can we develop better diagnostics and vaccines?"
General Public:
"How does this affect food safety and prices?"
People affected by LSDFarmers
Veterinarians
Government Officials
Researchers
General Public
Farmers:
Actively seeks information on preventive measures and early symptoms.
Veterinarians:
Conducts workshops, provides resources for farmers.
Government Officials:
Allocates funds for disease surveillance, implements control measures
Researchers:
Conducts studies, collaborates with other experts for faster solutions.
General Public:
Follows news updates, may change consumption patterns based on information.
Does
Farmers:
Concerned, anxious about potential economic losses.
Veterinarians:
Responsible for guiding farmers, frustrated if information is not effectively communicated.
Government Officials:Pressure to make informed decisions, responsible for public health.
Researchers:
Concerned about the impact on daily life, expects transparency from authorities.
General Public:
Concerned about the impact on daily life, expects transparency from authorities.
Feels

Project Design Phase-II
Data Flow Diagram & User Stories

Date	03 October 2022
Team ID	PNT2022TMIDxxxxxx
Project Name	Project - Predicting Lumpy Skin Disease
Maximum Marks	4 Marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


START
Collecting the experimental data
Defining Architecture
Collection of the data set (Skin disease and normal skin)
Data pre-processing (Resize the images 256x256 pixels, Normalize the pixel between 0 and 1 by dividing them by 255)
Data augmentation (random rotations, zooming, horizontal or vertical flipping, and shifting)
Initializing densenet pretrained model
Modifying the algorithm to predict the skin disease by adding layers
Setup division parameter, train(90%) and validation(10%)
Testing and validating trained model
Evaluating Model Performance by using using appropriate metrics such as accuracy, precision, recall, and F1 score
Reusing previously trained best model for prediction
Achieve high Accuracy
Prediction is done based on classified 2 classes
Stop

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task Description	Acceptance Criteria	Priority	Release
Farmer (User)	Registration	PLSD-1	As a farmer, I can register for the system by providing my contact information, farm details, and creating a password.	- I can access my dashboard after successful registration.	High	Sprint-1
		PLSD-2	As a farmer, I will receive a confirmation email upon successful registration.	- I can receive the confirmation email and click to confirm my account.	High	Sprint-1
	Data Input and Management	PLSD-3	As a farmer, I can enter cattle health records, environmental data, and disease-related information into the system.	- The system validates and saves the entered data.	High	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task Description	Acceptance Criteria	Priority	Release
		PLSD-4	As a farmer, I can upload images of cattle for disease prediction.	- The system processes the uploaded images for prediction.	High	Sprint-2
		PLSD-5	As a farmer, I can report diagnosed cases to provide feedback for model improvement.	- I can submit diagnosed cases and add relevant details.	Medium	Sprint-3
	Disease Prediction	PLSD-6	As a farmer, I can receive real-time disease predictions for my cattle based on the data provided.	- The system displays disease predictions for my cattle.	High	Sprint-2
Veterinarian (User)	Dashboard	PLSD-7	As a veterinarian, I can log in and access a dashboard with a list of farmers under my care.	- The dashboard displays a list of farmers and their cattle.	High	Sprint-3
		PLSD-8	As a veterinarian, I can view detailed health records and disease predictions for the cattle of each farmer.	- I can access detailed cattle health records and predictions.	High	Sprint-3
	Support and Feedback	PLSD-9	As a veterinarian, I can provide recommendations and interventions based on disease predictions.	- I can add recommendations for disease management.	High	Sprint-3
		PLSD-10	As a veterinarian, I can submit feedback and updates to improve the system's prediction accuracy.	- I can report issues and provide suggestions for improvement.	Medium	Sprint-4

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task Description	Acceptance Criteria	Priority	Release
System Administrator	User Management	PLSD-11	As an administrator, I can manage user accounts, including creating, updating, and deactivating accounts.	- I can create, edit, and deactivate user accounts.	High	Sprint-4
	Data Management	PLSD-12	As an administrator, I can oversee data integration and management, ensuring data quality and security.	- I can monitor data flow, validation, and security measures.	High	Sprint-4
	System Monitoring and Maintenance	PLSD-13	As an administrator, I can access a dashboard for system performance and maintain the system.	- The dashboard shows system usage, error logs, and key performance metrics.	Medium	Sprint-4

Project Design Phase-I
Proposed Solution Template

Date	19 September 2022
Team ID	PNT2022TMIDxxxxxx
Project Name	Project - Predicting Lumpy Skin Disease
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	Problem Statement: The problem statement is centered around addressing the significant issue of Lumpy Skin Disease (LSD) in cattle populations. Lumpy Skin Disease is a highly contagious and economically damaging viral disease that affects cattle. It results in decreased milk and meat production, increased mortality rates, and economic losses for farmers and the agricultural industry. The need for early detection and intervention is crucial to mitigate the spread and impact of this disease. The problem statement should encompass the prevalence of LSD, its consequences on the livestock industry, and the urgency of finding a solution to detect it at an early stage.
2.	Idea / Solution description	Idea / Solution Description: Our proposed solution involves the development of a machine learning model for predicting Lumpy Skin Disease in cattle. The model will use a combination of data sources, including cattle health records, environmental variables (such as climate, vegetation, and soil conditions), and remote sensing data from satellites or drones. The model will be trained to recognize patterns and early symptoms of LSD, enabling early detection and timely intervention. We plan to implement state-of-the-art machine learning algorithms, such as deep learning and ensemble methods, to achieve high prediction accuracy. Additionally, we will develop a user-friendly interface for farmers and veterinarians to input data and receive real-time predictions and disease risk assessments. This will empower

		them to make informed decisions regarding disease management. Our solution will leverage cloud computing and data storage for scalability and data management. The project will also explore the use of Internet of Things (IoT) devices and sensors for real-time data collection.
3.	Novelty / Uniqueness	Novelty / Uniqueness: What sets our solution apart from existing approaches is its integration of multiple data sources and advanced machine learning techniques. We plan to incorporate data from satellite imagery and IoT devices, providing a comprehensive view of the cattle's environment and health. This holistic approach allows us to capture early warning signs and environmental factors that may contribute to LSD outbreaks. Furthermore, our solution aims to adapt to regional variations and evolving disease strains by continuously learning and updating its predictive capabilities. We also plan to make our solution open-source, facilitating collaboration and innovation within the veterinary and data science communities, and thereby contributing to a broader understanding of disease prediction. In addition, the project will include a feedback loop for users to report diagnosed cases, further enhancing the model's performance and accuracy over time
4.	Social Impact / Customer Satisfaction	Social Impact / Customer Satisfaction: The social impact of our solution is multifaceted. Firstly, it will lead to a significant reduction in the suffering of cattle by enabling early diagnosis and treatment of Lumpy Skin Disease. Timely intervention can prevent the development of severe clinical signs and reduce animal distress. This, in turn, contributes to better animal welfare practices.
5.	Business Model (Revenue Model)	The revenue model for addressing the Lumpy Skin Disease problem encompasses subscription-based services for cattle stakeholders, partnerships with veterinary services leveraging the project's data, government and NGO funding to support accessibility, data sales to research institutions, consulting and training services for knowledge dissemination, technology licensing for

		expansion, and the possibility of donations to sustain the project's growth and mission of early disease prediction and livestock welfare. This multi-pronged approach ensures financial sustainability and widespread impact.
6.	Scalability of the Solution	The scalability of our solution to predict Lumpy Skin Disease lies in its adaptability to varying environmental conditions and regional disease strains through IoT data collection. Open-source collaboration allows customization for diverse communities, while an update and maintenance plan ensures relevance amid evolving disease patterns, fostering the project's long-term success and global reach.

Project Design Phase-I
Solution architecture

Date	19 September 2022
Team ID	PNT2022TMIDxxxxxx
Project Name	Project - **Predicting Lumpy Skin Disease**
Maximum Marks	2 Marks

Solution architecture:
Best tech solution to solve existing business problem:

Artificial Intelligence Algorithm in Image Processing for Cattle Disease Diagnosis

1. **Software Structure**:
    - **Modular Architecture**: Explain that the software is designed with a modular architecture, allowing for flexibility and scalability. Modules include data input, preprocessing, feature extraction, machine learning models, and user interface components.
    - **Data Flow**: Describe how data flows through the system, starting with input images and metadata, followed by preprocessing, feature extraction, machine learning models, and ending with disease diagnosis or classification.
2. **Characteristics**:
    - **Image Preprocessing**: Detail the image preprocessing techniques used, such as resizing, normalization, and noise reduction, to enhance the quality and suitability of input images for AI analysis.
    - **Feature Extraction**: Highlight the feature extraction methods, which capture relevant information from images, such as color, texture, and shape features.
    - **Machine Learning Models**: Discuss the AI algorithms used, such as convolutional neural networks (CNNs), and explain how they learn from the data to make predictions.
    - **User Interface**: Describe the user interface, emphasizing its user-friendliness, accessibility, and ability to display diagnosis results and recommendations.
    - **Scalability**: Mention that the software is designed for scalability, accommodating a growing database of cattle images and diseases for improved accuracy.
3. **Behavior**:
    - **Prediction and Classification**: Explain that the software analyzes input images to predict and classify cattle diseases. It identifies the disease based on the features extracted from the images.

- **Real-Time Processing**: Highlight that the software offers real-time image processing, allowing for immediate feedback to users.
- **Feedback Loop**: Emphasize that the software has a feedback loop where users can report diagnosed cases and improve the model's performance over time.
- **Alerts and Notifications**: Mention that the system provides alerts and notifications to farmers and veterinarians when potential diseases are detected.

4. **Other Aspects**:
    - **Data Security and Privacy**: Describe the software's robust data security measures to protect sensitive information. Mention compliance with data privacy regulations and the confidentiality of user data.
    - **Documentation**: Explain that comprehensive documentation is provided to guide users in understanding the software, from data input to interpreting results.
    - **Model Updates**: Communicate the plan for regular model updates and improvements based on new research findings and user feedback.
    - **Collaborations**: Mention any partnerships with veterinary services, agricultural organizations, or research institutions to enhance disease diagnosis and management.
    - **Training and Support**: Detail the provision of training for users and ongoing technical support to address any issues or questions.
    - **User Feedback Mechanism**: Highlight the presence of a user feedback mechanism to gather input for software enhancement and fine-tuning.

Features:

1. **Image Processing**:
    - Image preprocessing techniques for enhancing image quality.
    - Feature extraction to capture relevant image characteristics.
    - Advanced algorithms for image segmentation.
2. **Disease Classification**:
    - Machine learning models for disease prediction.
    - Multi-class classification for different cattle diseases.
    - Real-time analysis of input images.
3. **User Interface**:

- User-friendly web or mobile interface.
- Ability to upload cattle images.
- Real-time disease prediction display.
- Disease management recommendations.
- Data visualization tools.

4. **Feedback Loop**:
    - User reporting of diagnosed cases.
    - Continuous model improvement based on user feedback.
    - Regular model updates.
5. **Alerts and Notifications**:
    - Mobile alerts to farmers and veterinarians.
    - Immediate notification of potential disease outbreaks.
    - Disease hotspots identification on maps.
6. **Data Security and Privacy**:
    - Strong data encryption and security protocols.
    - Compliance with data privacy regulations.
    - Secure storage of sensitive user data.
7. **Scalability**:
    - Ability to handle a large database of cattle images and diseases.
    - Cloud-based architecture for scalability.
8. **Documentation**:
    - Comprehensive user guides and documentation.
    - Help resources for users.
    - Developer documentation for system maintenance.

Development Phases:

1. **Planning Phase**:
    - Define project scope and objectives.
    - Identify stakeholders and their needs.
    - Develop a project timeline and budget.
2. **Data Collection and Integration**:
    - Gather diverse data sources, including cattle health records and environmental data.

- Set up data integration and storage.

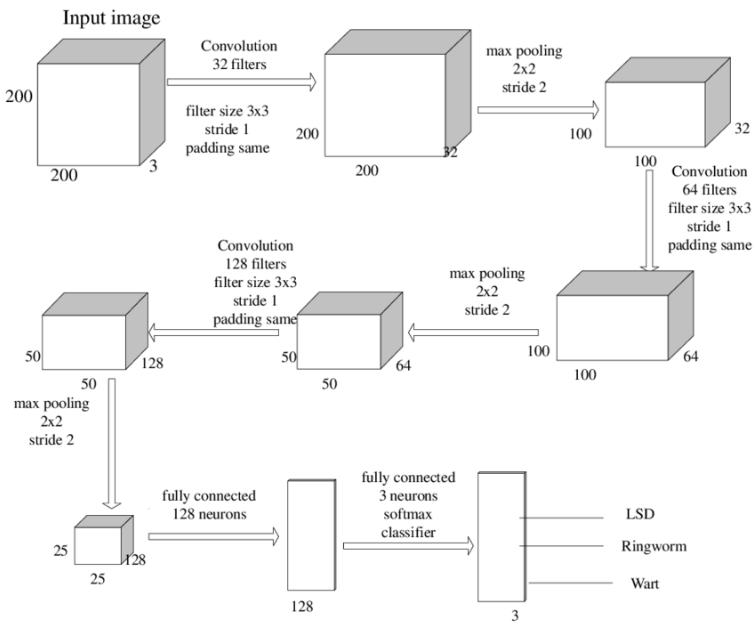
3. **Preprocessing and Feature Extraction**:
    - Implement image preprocessing techniques.
    - Develop feature extraction algorithms.
4. **Machine Learning Model Development**:
    - Choose appropriate machine learning algorithms (e.g., CNNs).
    - Train models using historical data.
    - Fine-tune models for optimal performance.
5. **User Interface Development**:
    - Design and develop a user-friendly interface.
    - Implement real-time image upload and prediction display.
6. **Feedback Mechanism**:
    - Create a user feedback mechanism.
    - Establish a system for user-reported diagnosed cases.
7. **Alerts and Notifications**:
    - Develop mobile notification systems.
    - Design alerts for disease outbreaks and hotspots.
8. **Data Security and Compliance**:
    - Implement data security measures.
    - Ensure compliance with relevant regulations.
9. **Scalability and Cloud Integration**:
    - Set up cloud-based infrastructure for scalability.
    - Ensure the system can handle increased data and user loads.
10. **Documentation and Support**:
    - Develop user guides and documentation.
    - Provide training for users and technical support.
11. **Testing and Quality Assurance**:
    - Perform extensive testing of the system.
    - Address and resolve bugs and issues.
12. **Deployment and Monitoring**:
    - Deploy the system for use.

- Monitor its performance and user feedback.

13. **Continuous Improvement**:
    - Regularly update machine learning models.
    - Analyze user feedback for improvements.
    - Adapt to evolving disease patterns.

Solution Requirements:

1. **Accuracy**: The system must provide accurate disease predictions and classifications.
2. **Speed**: Real-time or near-real-time image processing and predictions.
3. **User-Friendly Interface**: An easy-to-use interface for farmers and veterinarians.
4. **Data Security**: Robust data security measures to protect sensitive information.
5. **Scalability**: Ability to scale the system as the user base and data volume grow.
6. **Data Privacy Compliance**: Compliance with data privacy regulations and standards.
7. **Documentation**: Comprehensive user guides and documentation.
8. **User Training**: Training resources and support for users.
9. **Feedback Mechanism**: A system for user-reported diagnosed cases and feedback.
10. **Mobile Alerts**: Mobile notifications for disease outbreaks and hotspots.
11. **Continuous Improvement**: Regular model updates and system enhancements based on feedback and new research findings.



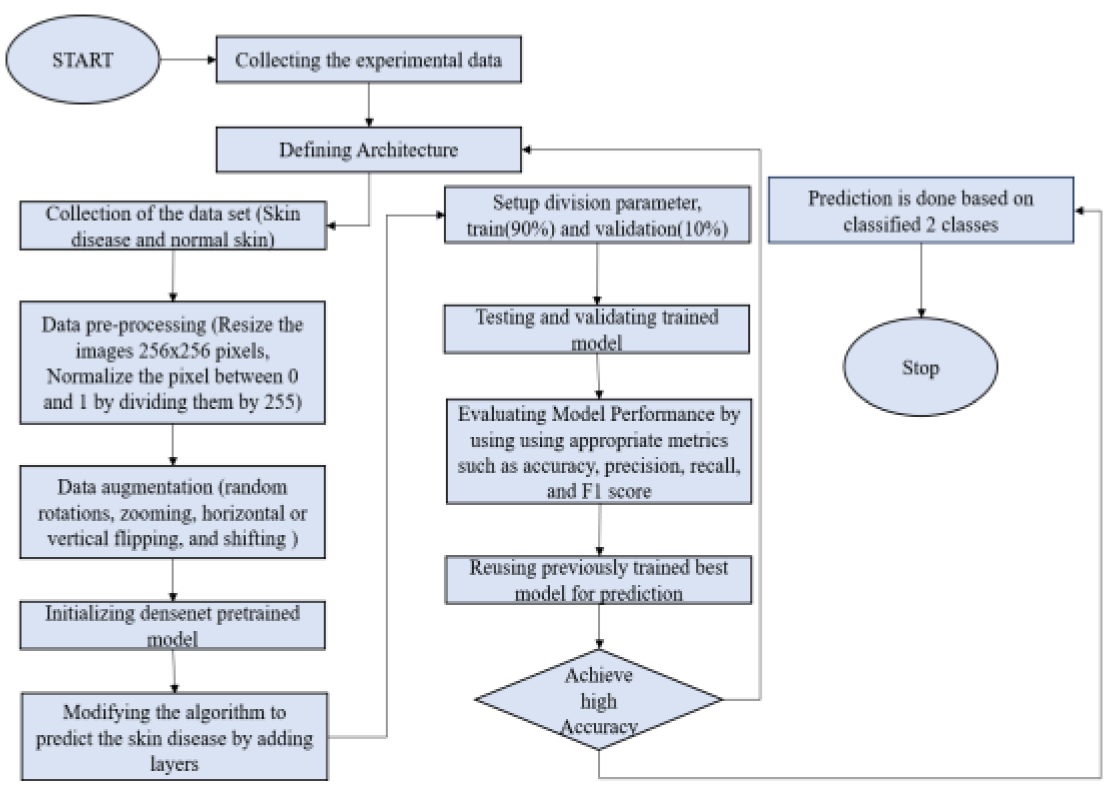
Project Planning Phase – I

Technology Stack (Architecture and Stack)

Date	25th October 2023
Team ID	
Project Name	Predicting Lumpy Skin Disease
Maximum Marks	4 Marks

Technological Architecture

The technical architecture of a web app for predicting lumpy skin disease using a **Convolutional Neural Network** involves multiple components that work together to deliver a functional and efficient application.



Component and Technologies :

S.no	Component	Description	Technology
1	Front-end	The user interface that allows users to interact with the application. It includes elements like input forms, image uploads, and result displays.	HTML, CSS, JavaScript, React, Angular, Vue.js
2	Back-end	The server-side of the application responsible for handling user requests, processing data, and interfacing with the machine learning model.	Python, Flask, Django, Node.js, Express.js
3	Database	Storage for user profiles, image data, and potentially the model parameters. Can be SQL or NoSQL, depending on the data requirements..	PostgreSQL,MySQL,MongoDB,Firebase Realtime Database
4	Machine Learning Model	The CNN model trained to predict lumpy skin disease from images. It processes input images and generates predictions.	TensorFlow, PyTorch, Keras
5	Image Processing	Preprocessing and manipulation of input images, including resizing, normalization, and data augmentation.	OpenCV, Pillow, Python Imaging Library (PIL)
6	User Authentication	User authentication and authorization to secure user data and model access.	OAuth, JWT tokens, Firebase Authentication
7	Cloud Deployment	Hosting the web app on cloud platforms for scalability, reliability, and global accessibility.	Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP)
8	Image Storage	Storing and managing user-uploaded images and model-related data.	AWS S3, Google Cloud Storage, Dropbox
9	Payment Processing	Handling payments, if the app charges users for access or premium features.	Stripe, PayPal, Square, Braintree
10	Third-Party APIs	Integration of external services,like geoloaction services or disease database APIs	Google Maps API, disease data APIs, etc.

Application and Characteristics:

S.no	Characteristic	Description	Technologies
1	User-Friendly Interface	Create an intuitive and user-friendly interface for easy interaction.	HTML, CSS, JavaScript, React, Angular, Vue.js
2	Real-time Image Upload	Allow users to upload images of skin lesions for disease prediction.	HTML file input, Python (Flask, Django) for server-side processing
3	Machine Learning Integration	Seamlessly integrate the trained CNN model for disease prediction	TensorFlow, PyTorch, RESTful API for communication
4	Data Storage and Retrieval	Store user data, images, and model-related data securely. Retrieve historical data for analysis and model improvement.	PostgreSQL, MySQL, MongoDB (for data storage), SQLalchemy, Mongoose (for database interaction)
5	User Authentication and Authorization	Implement secure user authentication to protect sensitive data and control access.	OAuth, JWT tokens, Firebase Authentication
6	Cloud Deployment	Deploy the application on cloud platforms for scalability and accessibility.	AWS, Azure, Google Cloud Platform (GCP)
7	Image Processing	Preprocess and normalize images before feeding them to the CNN model.	OpenCV, Pillow, Python Imaging Library (PIL)
8	Geolocation Services	If necessary, integrate geolocation services for location-specific data.	Google Maps API, Mapbox, Geocoding APIs
9	Payment Processing	Handle payments if users are charged for premium features	Stripe, PayPal, Square, Braintree, Payment gateways
10	Third-party APIs	Integrate external APIs to enhance the application's functionality, such as disease databases or additional resources.	External APIs (e.g., disease data APIs)

References :

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Devin, M. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint* arXiv:1603.04467.

2. Abbass HA. An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artificial Intelligence in Medicine.* 2002;25(3):265–281. doi: 10.1016/S0933-3657(02)00028-3. [PubMed] [CrossRef] [Google Scholar]

3. Alemayehu, G., Zewde, G., & Admassu, B. (2013). Risk assessments of lumpy skin diseases in Borena bull market chain and its implication for livelihoods and international trade. *Tropical Animal Health and Production, 45*(5), 1153–1159. 10.1007/s11250-012-0340-9 [PMC free article] [PubMed]

4. Alkhamis MA, VanderWaal K. Spatial and temporal epidemiology of lumpy skin disease in the Middle East, 2012–2015. *Frontiers in Veterinary Science.* 2016;3:19. doi: 10.3389/fvets.2016.00019. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

5. Allepuz A, Casal J, Beltrán-Alcrudo D. Spatial analysis of lumpy skin disease in Eurasia—Predicting areas at risk for further spread within the region. *Transboundary and Emerging Diseases.* 2019;66(2):813–822. doi: 10.1111/tbed.13090. [PubMed] [CrossRef] [Google Scholar]

Project Planning Phase-II

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	27th October,2023
Team ID	
Project Name	Predicting Lumpy Skin Disease
Maximum Marks	8 Marks

Sprint Planning Table

User Stories	Acceptance Criteria	Story Points	Priority	Team Members
Ideation Phase Empathy Phase Brainstorming Phase	5	20	High	Devansh
Image Uploading and Storage Enable users to upload images of skin lesions. Store user-uploaded images securely in cloud storage	5	20	Medium	Devansh
Image Preprocessing Implement image preprocessing to prepare images for the CNN model. CNN Model Integration Develop and	3	20	Low	Bhavya

integrate the CNN model for lumpy skin disease prediction. Train the model using a labeled dataset.				
Real-time Prediction Allow users to submit images for real-time prediction. Display the prediction results to the user.	4	20	Medium	Sanjay
User Profile Management Allow users to edit their profiles. Implement profile picture uploads.	2	20	Medium	Sanjay
Database Integration Set up a database (e.g., PostgreSQL) for user profiles and data storage.	4	20	High	Bhavya

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint1	20	8	20th October,2023	27th October,2023	20	29th October ,2023
Sprint1	20	4	21st October,2023	27th October,2023		
Sprint1	20	4	23rd October,2023	27th October,2023		
Sprint1	20	13	14th October,2023	27th October,2023		
Sprint1	20	5	22nd October,2023	27th October,2023		

Sprint1	20	5	23rd October,2023	28th October,2023		
Sprint1	20	4	27th October,2023	31st October,2023		

Velocity:

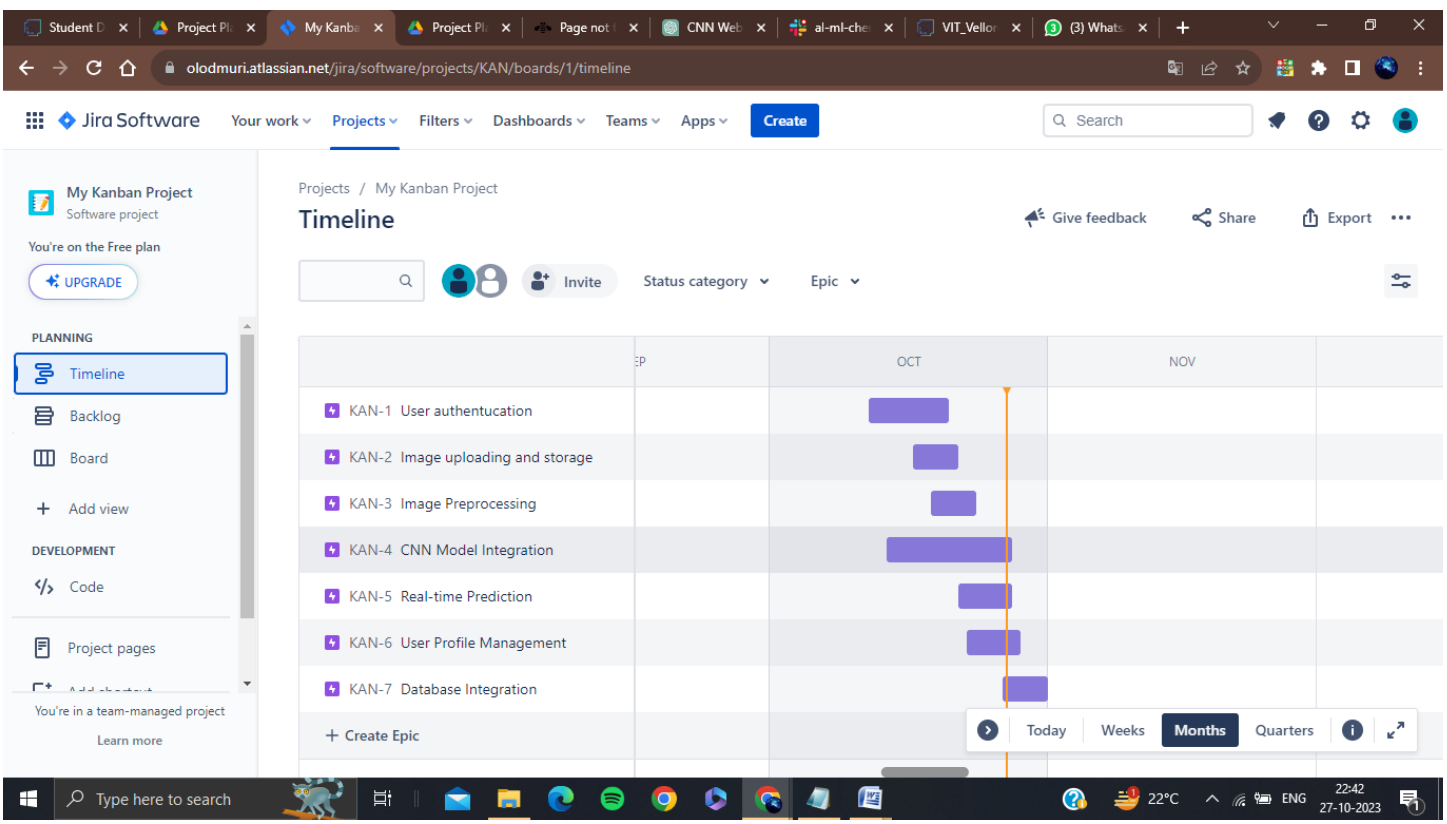
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per

iteration unit (story points per day)

AV= Sprint Duration/Velocity

19/10=1.9

Timeline:



References:

https://www.atlassian.com/agile/project-management

https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software

https://www.atlassian.com/agile/tutorials/epics

https://www.atlassian.com/agile/tutorials/sprints

https://www.atlassian.com/agile/project-management/estimation

https://www.atlassian.com/agile/tutorials/burndown-charts

Predicting Lumpy Skin Disease

Introduction

Lumpy Skin Disease (LSD) is a highly contagious viral disease that affects cattle and poses

a significant threat to livestock industries worldwide. Early detection and accurate

prediction of LSD outbreaks are crucial for effective disease control and prevention. In this

project, we aim to develop a machine learning model to predict the occurrence of Lumpy

Skin Disease using a dataset containing various geographical and environmental factors.

Dataset Description

The dataset used for this project includes the following columns:

- Longitude (X-axis spatial coordinates)
- Continent of the outbreak
- Latitude (Y-axis spatial coordinates)
- Monthly Cloud Cover in percent
- Diurnal Temperature Range in degrees Celsius
- Country of outbreak
- Frost Day Frequency in a month
- Potential Evapotranspiration in millimetres per day
- Precipitation in millimetres per month
- Daily Mean Temperature in degrees Celsius
- Temperature in degrees Celsius
- Monthly Average Maximum and Minimum Temperature in degrees Celsius
- Vapor Pressure in hectopascals
- Wet Day Frequency in days
- Altitude of geographic location in meters
- Dominant Land Cover
- Lumpy (target variable)

Project Flow:

1. Data Collection and Preparation

- Collect the dataset from reliable sources.
- Perform data cleaning and preprocessing.

2. Exploratory Data Analysis (EDA)

● Analyse the dataset using descriptive statistics and visualizations.

● Explore the distribution of variables and identify any patterns or trends.

3. Feature Engineering

● Extract relevant features from the dataset.

● Handle missing values and outliers, if any.

● Transform categorical variables into numerical representations, if required.

4. Model Building

● Split the dataset into training and testing sets.

● Train various machine learning models on the training set.

● Evaluate the performance of each model using appropriate evaluation metrics.

● Select the best-performing model for further analysis.

5. Model Evaluation

● Evaluate the optimized model on the testing set.

● Assess its predictive accuracy and reliability.

6. Model Deployment

● Deploy the final model to make predictions on new, unseen data.

● Develop a user-friendly interface or API for easy access to the model's predictions.

7. Documentation and Reporting

● Prepare a comprehensive project report documenting the entire process.

● Present the findings, insights, and conclusions derived from the project.

● Provide recommendations for further improvements or future research.

By accurately predicting the occurrence of Lumpy Skin Disease, this machine learning

project can significantly contribute to early detection and effective management of the

disease, ultimately leading to improved livestock health and the prevention of economic

losses in the livestock industry.

Milestone 1: Define Problem / Problem Understanding

Activity 1: Specify the Business Problem

The business problem for the accurate prediction of Lumpy Skin Disease is to develop a machine learning model that can effectively predict the occurrence of Lumpy Skin Disease in cattle. Lumpy Skin Disease is a highly contagious viral disease that affects cattle, causing significant economic losses in the livestock industry. By accurately predicting the disease occurrence, proactive measures can be taken for disease control and prevention, reducing the spread and impact of Lumpy Skin Disease.

Activity 2: Business Requirements

To ensure that the Lumpy Skin Disease prediction model meets business requirements

and can be deployed effectively, the following rules and requirements need to be

considered:

1. Accuracy: The model should demonstrate a high level of accuracy in predicting the occurrence of Lumpy Skin Disease. It should provide reliable and precise predictions to support decision-making processes related to disease control and prevention.
2. Early Detection: The model should be able to detect the presence of Lumpy Skin Disease at an early stage to facilitate timely intervention and minimize the risk of disease spread within cattle populations.
3. Scalability: The model should be scalable to handle large volumes of data and accommodate future growth in the livestock industry. It should be capable of processing data from multiple sources and adapting to evolving disease patterns.
4. Interpretability: The model should be interpretable, meaning that its predictions can be explained and understood by stakeholders. Interpretability is essential for building trust in the model and enabling informed decision-making based on its outputs.

5. Privacy and Security: The model should adhere to privacy and security regulations to protect sensitive data. Measures should be implemented to ensure secure storage, handling, and access to data used for training and prediction purposes.

Activity 3: Literature Survey

A literature survey for the accurate prediction of Lumpy Skin Disease would involve researching and reviewing existing studies, articles, and publications related to Lumpy Skin Disease in cattle. The survey aims to gather insights on the following aspects:

1. Disease Characteristics: Understanding the aetiology, epidemiology, and clinical manifestations of Lumpy Skin Disease in cattle. Exploring factors that contribute to disease transmission and spread.
2. Risk Factors: Identifying risk factors associated with Lumpy Skin Disease, such as breed susceptibility, age, geographical location, and environmental conditions.
3. Diagnostic Methods: Reviewing existing diagnostic methods for Lumpy Skin Disease, including clinical observations, laboratory tests, and imaging techniques. Exploring their limitations and potential for improvement.
4. Machine Learning Approaches: Investigating previous studies that have utilized machine learning techniques for disease prediction in cattle. Assessing the performance of different algorithms and feature selection methods.
5. Data Availability: Identifying potential sources of data for training and validating the prediction model. Assessing the quality, completeness, and reliability of available datasets.

The literature survey will help in gaining a comprehensive understanding of Lumpy Skin Disease, its predictive modelling approaches, and the gaps in knowledge that can be addressed through this project.

Milestone 2: Data Collection & Preparation

Activity 1: Collect the Dataset

To develop an accurate prediction model for Lumpy Skin Disease, a comprehensive dataset related to the disease and cattle characteristics needs to be collected. The dataset should include relevant features that can contribute to the prediction of Lumpy Skin Disease occurrence. The following steps should be followed to collect the dataset:

Activity 1.1: Importing the libraries

Utilize the necessary software frameworks and dependencies as illustrated in the accompanying visual representation, in order to facilitate the successful implementation of this machine learning endeavour.

Activity 1.2: Dataset Reading

The dataset provided may be in various formats such as .csv, Excel files, .txt, .json, among others. To effectively process the dataset, we will employ the pandas library.

Considering that the dataset is in a CSV file format, we will utilize the pandas function read_csv() to ingest the dataset. This function requires the directory path to the CSV file as a parameter.

To preview the initial 5 rows of the dataset, we will employ the df.head() function, which displays the desired subset of the data.

Activity 2: Data Preparation

Data preparation, or data preprocessing, refers to the essential steps of refining, transforming, and organizing raw data prior to its utilization in data analysis or machine learning models.

The outlined activity encompasses the following steps:

- Identification and removal of missing values
- Restoring the missing values.
- Encoding categorical variables.
- Normalizing the data.

Please note that these steps serve as a general guideline for pre-processing data before its application in machine learning training. The specific pre-processing requirements may vary based on the characteristics of the dataset.

2.1 Identification and removal of missing values.

Upon thorough examination, it has come to our attention that there exists a discernible pattern among the missing values observed in three specific variables. However, we have been unable to identify the precise reporting date within our dataset. Consequently, we have made the decision to remove the column pertaining to the reporting date. Nonetheless, after careful consideration, we have determined that the continent and countries columns bear significant importance as they play a pivotal role in exploratory data analysis, visualization, and overall model construction. Therefore, we have opted to retain these columns within our dataset, recognizing their value and relevance to our objectives.

2.2 Restoring the missing values.

Remarkably, approximately 80% of the data contained within the continent and country columns has been identified as missing. Fortunately, we possess comprehensive information in the form of longitude and latitude coordinates. Leveraging the capabilities of the Python modules "pycountry" and "geocoder," we can utilize geospatial coordinates to derive and compute the corresponding country and continent for each data point. This approach enables us to bridge the gap in the dataset and successfully determine the missing values for the continent and country variables.

Executing the aforementioned code snippet to implement the proposed solution.

In order to enhance comprehension and facilitate better understanding, we will assign country names based on the existing country codes available in the dataset. By utilizing the

country codes as references, we can replace the country codes with corresponding country names, enabling clearer interpretation of the data.

Upon restoring a significant portion of the missing values in the two columns, a subsequent examination reveals that approximately 14% of the country names remain unresolved.

Further scrutiny has confirmed that all of these countries, except for one European country, belong to the African continent. To address this, we shall replace the remaining null values with suitable values, taking into consideration the geographic context and assigning the appropriate country names accordingly.

2.3 Encoding categorical variables.

We have identified two categorical columns within our dataset. Considering the extensive number of countries, which exceeds a hundred, we have made the decision not to encode the country column. Instead, we will focus on encoding the continent column. To achieve this, we will leverage the column transformer functionality offered by the sklearn module. It is important to note that the column transformer converts the provided data into an array format following the transformation process. However, for our subsequent analysis, we require the dataset to be in a dataframe format. As a result, we will apply the column transformer at the initial stages of our model building process, subsequent to the completion of exploratory data analysis (EDA) and visualization tasks.

Milestone 3: Exploratory Data Analysis

Activity 1: Descriptive Statistical Analysis

In this activity, the collected dataset for Lumpy Skin Disease is subjected to descriptive statistical analysis to gain insights into the data. Various statistical measures such as mean, median, mode, standard deviation, and quartiles are calculated for numerical variables

related to the disease, such as lesion size, duration of symptoms, and severity of infection. Frequency distributions and histograms are generated to visualize the distribution of categorical variables, including geographic regions, affected livestock breeds, and vaccination status. These descriptive statistics help in understanding the central tendencies, variabilities, and distributions of the dataset, providing initial insights into the prevalence and characteristics of Lumpy Skin Disease.

Activity 2: Visual analysis

Activity 2.1: Univariate analysis

The code snippet presented below facilitates the generation of histograms to visualize the distribution of numerical columns, namely "wet day" and "temperatures." By employing Python's Matplotlib library, these histograms provide a graphical representation of the frequency distribution for each respective column. This aids in gaining a deeper understanding of the data's characteristics and patterns related to "wet day" and "temperatures" variables.

Activity 2.2: Bivariate analysis

Utilizing the code provided in the accompanying visual representation, we can ascertain the top ten countries that experienced the highest impact from the disease. The code employs a specific methodology to analyze the dataset and extract the relevant information, enabling the identification of the countries that suffered the most significant effects of the disease outbreak.

Similarly, employing the code depicted in the aforementioned visual representation, we can also determine the ten least affected countries. This code utilizes a specific approach to analyze the dataset and extract the pertinent information, enabling the identification of countries that experienced relatively lower impact from the disease outbreak. By examining the data, we can ascertain the countries that were least affected by the disease.

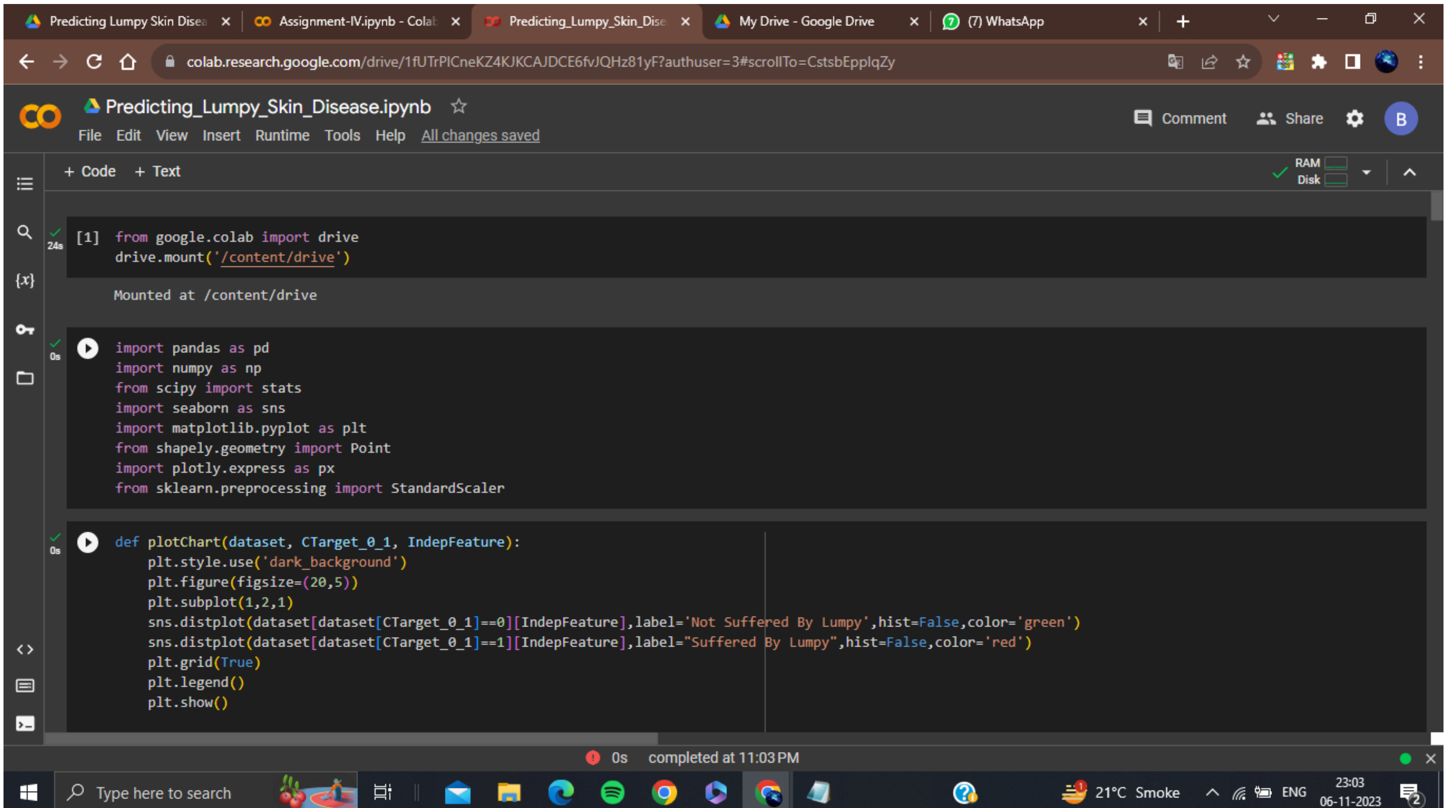
To determine the quantity of datapoints available in this dataset, we can leverage the Plotly module and its Scatter Geo function. By plotting the "longitude" and "latitude" columns on a map using this function, we can visualize the geographical distribution of the data points. This enables us to gain insights into the density and spread of the datapoints across different locations on the map, providing an estimate of the dataset's extent and coverage.

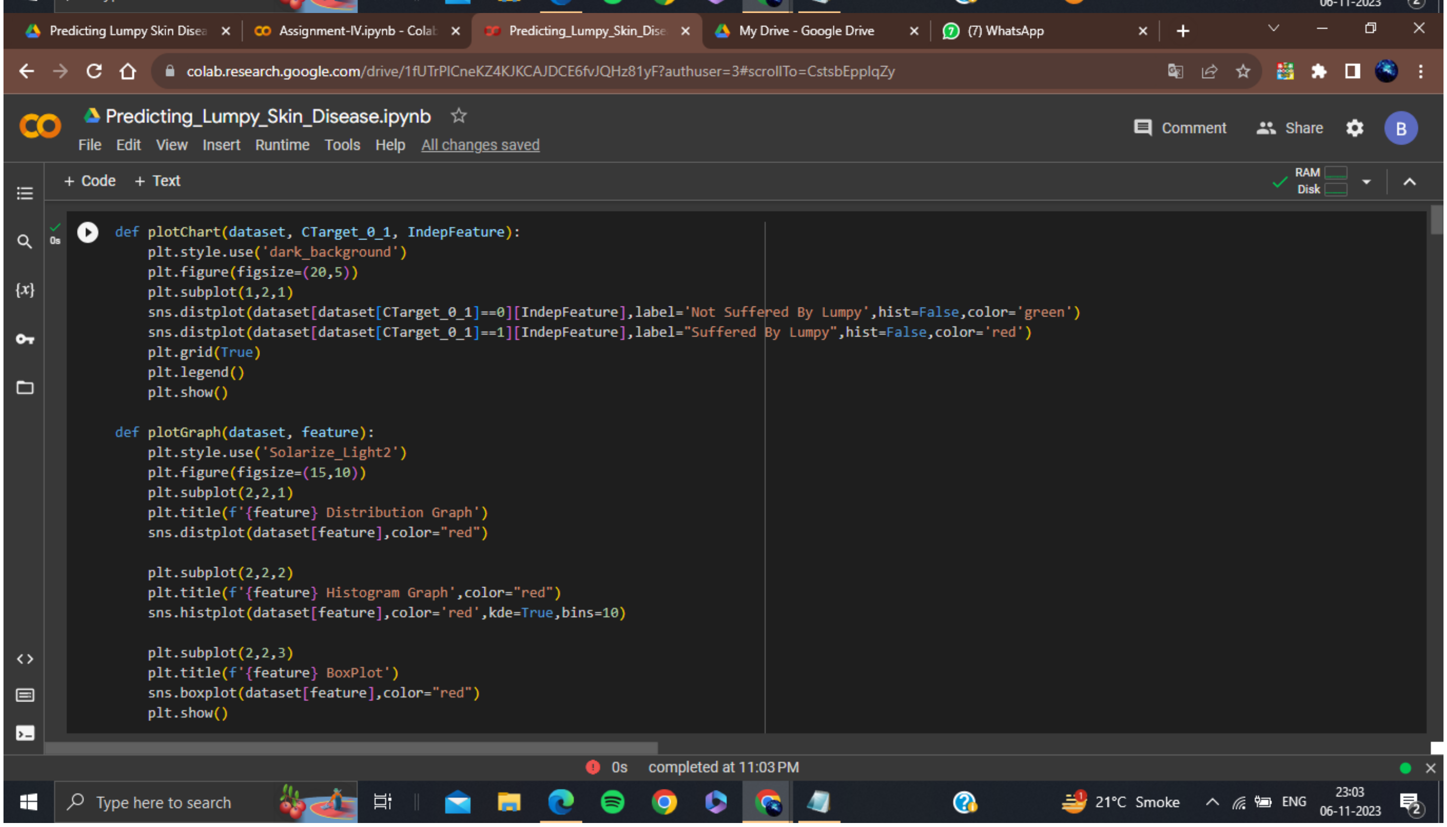
Activity 2.3: Multivariate analysis

Continuing our utilization of the Plotly module, we employ the Scatter mapbox functionality for multivariate analysis. By leveraging the Scatter mapbox function, we can visualize the distinction between locations that were affected by the disease and those that were not. This analysis allows us to observe and discern any discernible patterns, spatial relationships, or differences between diseased and non-diseased locations on a geographical map. The interactive nature of Plotly enables us to explore and gain deeper insights into the spatial dynamics of the disease's impact.

Once again, we harness the power of the Plotly module to explore the relationship between two numerical variables and a categorical column. By employing Plotly's visualization capabilities, we can create interactive charts or graphs that provide insights into the connections, dependencies, or patterns that may exist between these variables. This analysis enables us to better comprehend how the categorical column interacts with and influences the numerical variables, allowing for a more comprehensive understanding of the dataset's underlying dynamics.

To ascertain the interplay between elevation and geographical coordinates, we employ the following code snippet as part of our analytical methodology.





Predicting_Lumpy_Skin_Disease.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[3]  sns.boxplot(dataset[feature],color="red")
     plt.show()
```

```
path="/content/drive/MyDrive/Lumpy skin disease data.csv"
df_lumpy=pd.read_csv(path)
df_lumpy.head()
```

	x	y	region	country	reportingDate	cld	dtr	frs	pet	pre	tmn	tmp	tmx	vap	wet	elevation	dominant_land_cover	X5_Ct_2010_Da	X5_Bf_2(
0	90.380931	22.437184	Asia	Bangladesh	10/9/2020	41.6	12.8	0.00	2.3	1.7	12.7	19.1	25.5	15.7	0.00	147	2	27970.983100	3691.
1	87.854975	22.986757	Asia	India	20/12/2019	40.5	13.3	0.00	2.4	0.0	13.2	19.8	26.5	16.3	0.00	145	2	25063.646690	671
2	85.279935	23.610181	Asia	India	20/12/2019	27.3	13.6	0.08	2.3	0.6	9.4	16.2	23.0	13.0	0.98	158	2	6038.477155	1426
3	81.564510	43.882221	Asia	China	25/10/2019	45.3	12.8	31.00	0.4	8.8	-22.5	-16.1	-9.7	0.9	4.64	178	2	760.703340	0.
4	81.161057	43.834976	Asia	China	25/10/2019	38.8	13.2	31.00	0.4	10.5	-20.4	-13.8	-7.2	1.2	1.69	185	3	270.367426	0.

```
[6]  df_lumpy.describe()
```

	x	y	cld	dtr	frs	pet	pre	tmn	tmp	tmx	vap	wet
count	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000	24803.000000
mean	79.221374	46.370056	59.452159	9.107777	23.978048	0.803487	26.271137	-15.794755	-11.227807	-6.681212	3.728230	8.542482
std	43.338530	19.220555	19.423029	2.988448	11.518315	1.172915	33.630747	17.587685	17.989715	18.540915	4.952353	6.205199
min	-179.750000	-28.750000	0.000000	2.000000	0.000000	0.000000	0.000000	-52.100000	-48.100000	-44.200000	0.000000	0.000000
25%	45.083150	34.750000	43.800000	6.800000	23.210000	0.000000	5.900000	-30.100000	-25.500000	-20.900000	0.400000	3.000000
50%	80.750000	48.250000	62.300000	8.300000	31.000000	0.200000	14.700000	-19.100000	-14.200000	-9.700000	1.500000	8.020000
75%	109.750000	61.750000	75.300000	11.100000	31.000000	1.100000	33.400000	-2.200000	1.400000	4.900000	4.800000	12.710000
max	179.750000	81.750000	98.700000	20.600000	31.000000	7.500000	341.900000	23.900000	28.500000	36.400000	28.600000	30.920000

```
[7]  df_lumpy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24803 entries, 0 to 24802
Data columns (total 20 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   x              24803 non-null  float64
 1   y              24803 non-null  float64
 2   region         3039 non-null   object
 3   country        3039 non-null   object
```

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

```
[8] df_lumpy.duplicated().sum()
```

```
608
```

```
[9] df_lumpy = df_lumpy.drop_duplicates()
```

```
df_lumpy.shape
```

```
(24195, 20)
```

```
[11] df_lumpy.isnull().sum()
```

```
x                    0
y                    0
region           21764
country          21764
reportingDate    21764
cld                  0
dtr                  0
frs                  0
pet                  0
pre                  0
tmn                  0
tmp                  0
tmx                  0
vap                  0
```

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

```
[12] df_lumpy.rename(columns={"x":"Longitude" , "y":"Latitude", "cld":"Monthly Cloud Cover", "dtr":"Diurnal Temperature Range", "frs":"Frost Day Frequency", "pet":"Pc
```

```
df_lumpy.head()
```

	Longitude	Latitude	region	country	reportingDate	Monthly Cloud Cover	Diurnal Temperature Range	Frost Day Frequency	Potential EvapoTranspiration	Precipitation	Minimum Temperature	Mean Temperature	Maximum Temperature
0	90.380931	22.437184	Asia	Bangladesh	10/9/2020	41.6	12.8	0.00	2.3	1.7	12.7	19.1	25.5
1	87.854975	22.986757	Asia	India	20/12/2019	40.5	13.3	0.00	2.4	0.0	13.2	19.8	26.5
2	85.279935	23.610181	Asia	India	20/12/2019	27.3	13.6	0.08	2.3	0.6	9.4	16.2	23.0
3	81.564510	43.882221	Asia	China	25/10/2019	45.3	12.8	31.00	0.4	8.8	-22.5	-16.1	-9.7
4	81.161057	43.834976	Asia	China	25/10/2019	38.8	13.2	31.00	0.4	10.5	-20.4	-13.8	-7.2

```
[14] z = np.abs(stats.zscore(df_lumpy['Monthly Cloud Cover']))
     out=np.where(z>3)[0]
     print('Monthly Cloud Cover'+ " " + str(out))
     df_lumpy['Monthly Cloud Cover'] = df_lumpy['Monthly Cloud Cover'].drop(out)
```

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

```
Diurnal Temperature Range [18309 18310 18311 18312 18325 18326 18327 18328 18341 18342 18343 18344
 18358 19872 19906 19907 19908 19910 19911 19912 19913 19914 19941 19942
 19943 19944 19945 19946 19947 19948 19949 19950 19979 19981 20039 20078
 20079 20133 20134 20188 20241 20296 21139 21151 21152 21159 21160 21166
 22524 22525 22526 22569 22570 22571 22572 22613 22614 22615 22616 22617
 22656 22657 22658 22659 22660 22661 22702 22703 22704 22705 22706 22707
 22748 22749 22750 22751 22752 22753 22793 22794 22795 22796 22797 22798
 22799 22840 22841 22842 22843 22844 22889 22890 22891 22892 22939 22940]
```

```python
z = np.abs(stats.zscore(df_lumpy['Frost Day Frequency']))
out = np.where(z>3)[0]
print('Frost Day Frequency'+ " " + str(out))
df_lumpy['Frost Day Frequency'] = df_lumpy['Frost Day Frequency'].drop(out)
```

```
Frost Day Frequency []
```

```python
[17] z = np.abs(stats.zscore(df_lumpy['Mean Temperature']))
out = np.where(z>3)[0]
print('Mean Temperature'+ " " + str(out))
df_lumpy['Mean Temperature'] = df_lumpy['Mean Temperature'].drop(out)
```

```
Mean Temperature []
```

```python
[18] z = np.abs(stats.zscore(df_lumpy['Minimum Temperature']))
out = np.where(z>3)[0]
```

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

```python
[17] print('Mean Temperature'+ " " + str(out))
df_lumpy['Mean Temperature'] = df_lumpy['Mean Temperature'].drop(out)
```

```
Mean Temperature []
```


```python
[18] z = np.abs(stats.zscore(df_lumpy['Minimum Temperature']))
out = np.where(z>3)[0]
print('Minimum Temperature'+ " " + str(out))
df_lumpy['Minimum Temperature'] = df_lumpy['Minimum Temperature'].drop(out)
```

```
Minimum Temperature []
```

```python
[20] z = np.abs(stats.zscore(df_lumpy['Maximum Temperature']))
out = np.where(z>3)[0]
print('Maximum Temperature'+ " " + str(out))
df_lumpy['Maximum Temperature'] = df_lumpy['Maximum Temperature'].drop(out)
```

```
Maximum Temperature []
```

```python
[21] z = np.abs(stats.zscore(df_lumpy['Wet Day Frequency']))
out = np.where(z>3)[0]
print('Wet Day Frequency'+ " " + str(out))
df_lumpy['Wet Day Frequency'] = df_lumpy['Wet Day Frequency'].drop(out)
```

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
[23] df_lumpy = df_lumpy.fillna(method='bfill')
     df_lumpy.isnull().sum()
```

```
Longitude                        0
Latitude                         0
region                       21764
country                      21764
reportingDate                21764
Monthly Cloud Cover              0
Diurnal Temperature Range        0
Frost Day Frequency              0
Potential EvapoTranspiration     0
Precipitation                    0
Minimum Temperature              0
Mean Temperature                 0
Maximum Temperature              0
Vapour Pressure                  0
Wet Day Frequency                0
elevation                        0
dominant_land_cover              0
X5_Ct_2010_Da                    0
X5_Bf_2010_Da                    0
lumpy                            0
dtype: int64
```

```
[24] geolocation = df_lumpy.loc[df_lumpy['lumpy'] != 0][['Longitude', 'Latitude', 'country']]
     geolocation.head()
```

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

File Edit View Insert Runtime Tools Help All changes saved

```
dominant_land_cover              0
X5_Ct_2010_Da                    0
X5_Bf_2010_Da                    0
lumpy                            0
dtype: int64
```

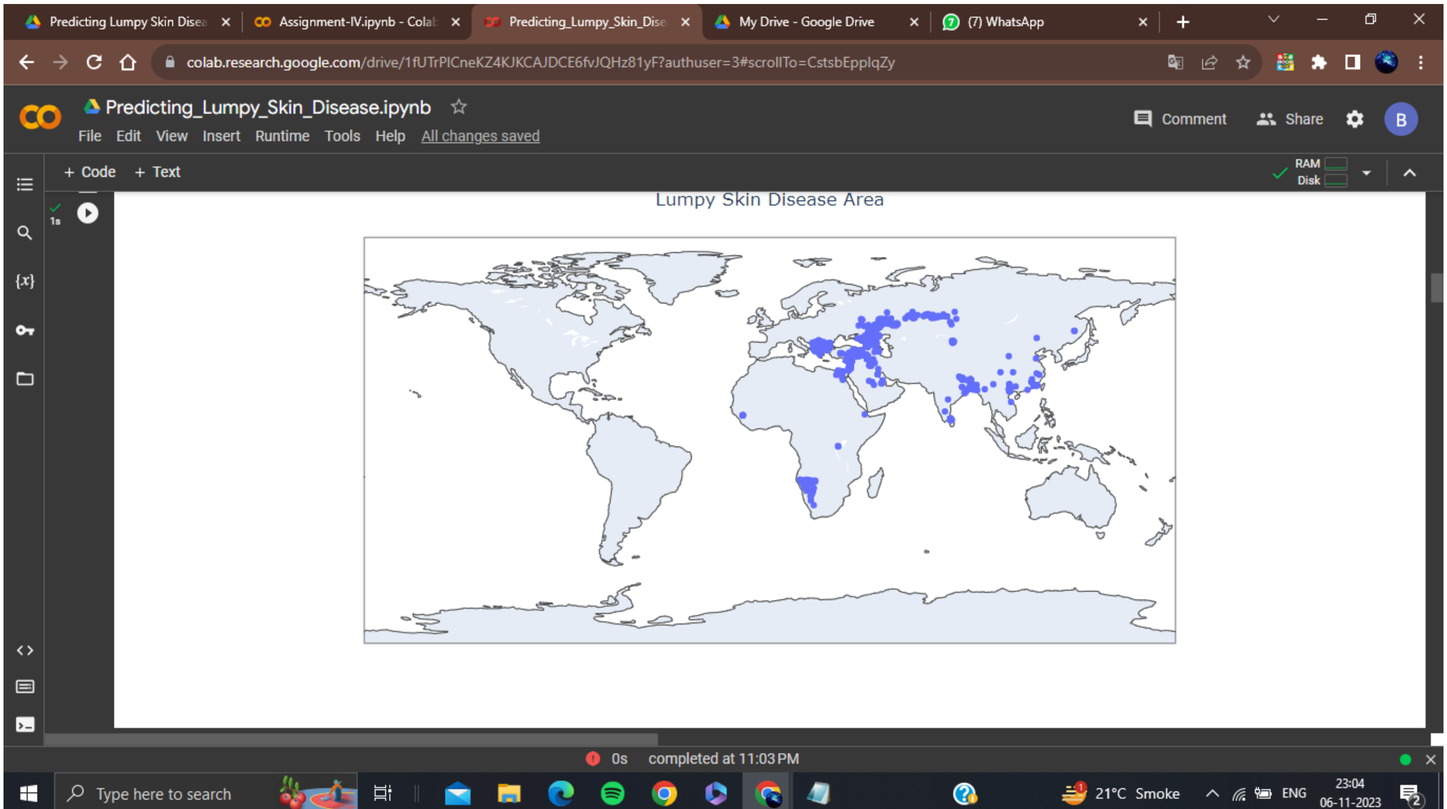
```
geolocation = df_lumpy.loc[df_lumpy['lumpy'] != 0][['Longitude', 'Latitude', 'country']]
geolocation.head()
```

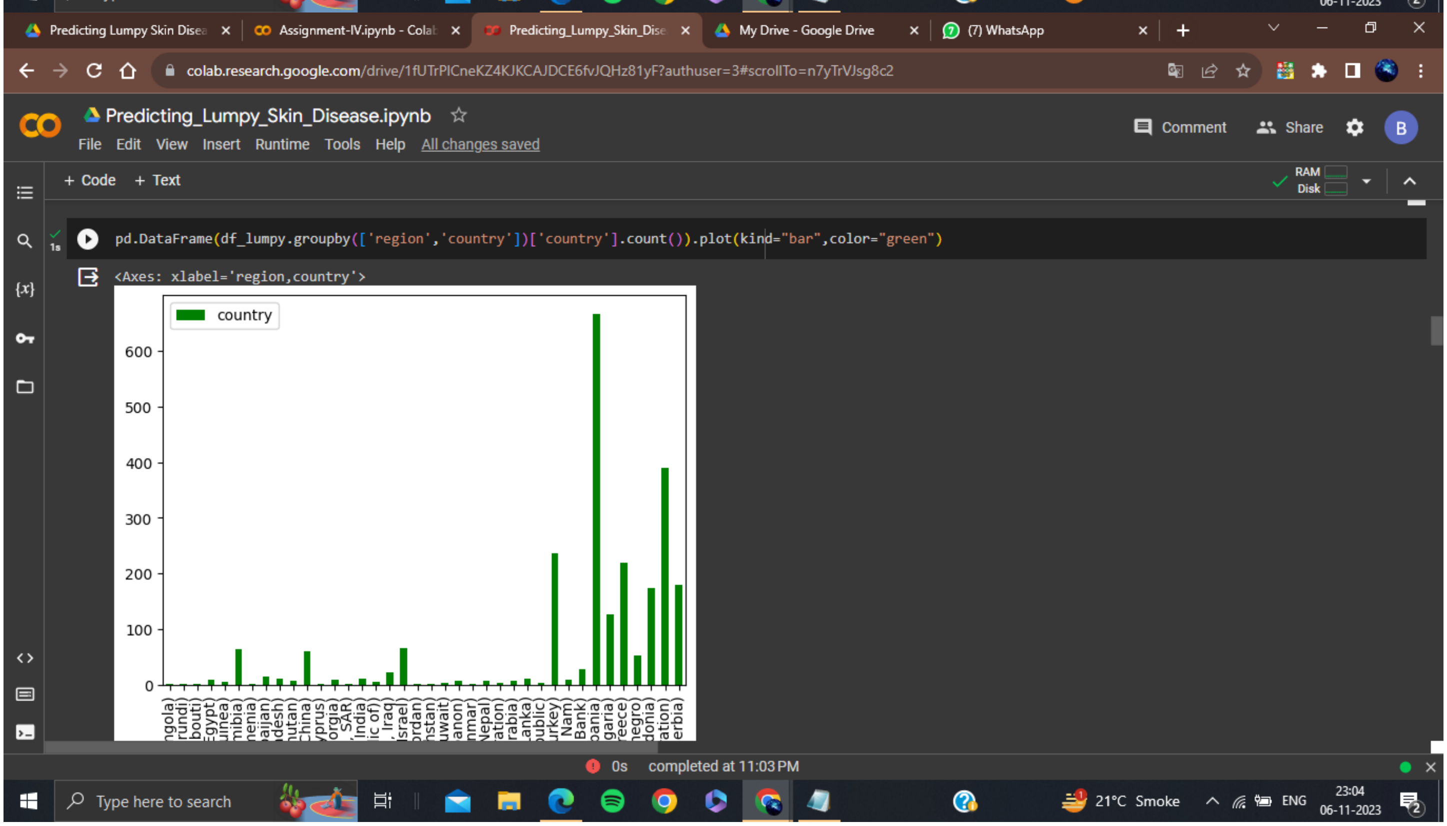

	Longitude	Latitude	country
0	90.380931	22.437184	Bangladesh
1	87.854975	22.986757	India
2	85.279935	23.610181	India
3	81.564510	43.882221	China
4	81.161057	43.834976	China

```
[25] fig = px.scatter_geo(geolocation,lat='Latitude',lon='Longitude')
     fig.update_layout(title = 'Lumpy Skin Disease Area', title_x=0.5)
     fig.show()
```

Lumpy Skin Disease Area

0s completed at 11:03 PM


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Lumpy Skin Disease Area
0s completed at 11:03 PM


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
pd.DataFrame(df_lumpy.groupby(['region','country'])['country'].count()).plot(kind="bar",color="green")
<Axes: xlabel='region,country'>
country
600
500
400
300
200
100
0
0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

File Edit View Insert Runtime Tools Help All changes saved


```
[31] df_lumpy['reportingDate'] =pd.to_datetime(df_lumpy['reportingDate'])
     df_lumpy['Year'] = df_lumpy['reportingDate'].dt.year
     pd.DataFrame(df_lumpy['Year'].value_counts().sort_values(ascending=True)).rename(columns={"Year" : "Case Report Count"})
```

<ipython-input-31-479aa8050f70>:1: UserWarning:

Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead to inconsistently parsed dates! Specify a format to ensure consi

	Case Report Count
2011.0	6
2021.0	11
2012.0	21

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

File Edit View Insert Runtime Tools Help All changes saved

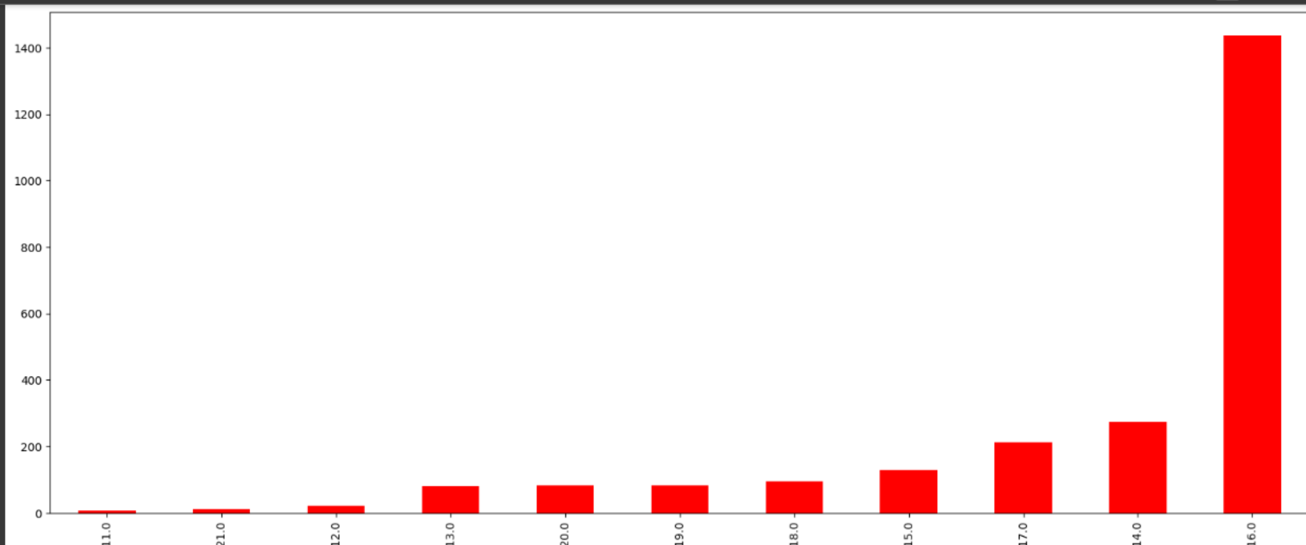
	Case Report Count
2011.0	6
2021.0	11
2012.0	21
2013.0	81
2020.0	83
2019.0	83
2018.0	94
2015.0	129
2017.0	212
2014.0	275
2016.0	1436

```
[32] plt.figure(figsize=(20,8))
     df_lumpy[df_lumpy['Year'] == df_lumpy['Year']]['Year'].value_counts().sort_values(ascending=True).plot(kind="bar",color="red")
```

<Axes: >

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb



0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

```
[33] df_lumpy['Month'] = df_lumpy['reportingDate'].dt.month
     df_lumpy['MonthTitle']= df_lumpy['reportingDate'].dt.month_name()
```

```
pd.DataFrame(df_lumpy.groupby(['Year','Month'])['Month'].count()).rename(columns={'Month':'CaseCounts'})
```

Year	Month	CaseCounts
2011.0	7.0	1
	10.0	3
	11.0	2
2012.0	5.0	2
	7.0	2
...	...	...
2020.0	9.0	8
	10.0	13
	11.0	7
	12.0	10

0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

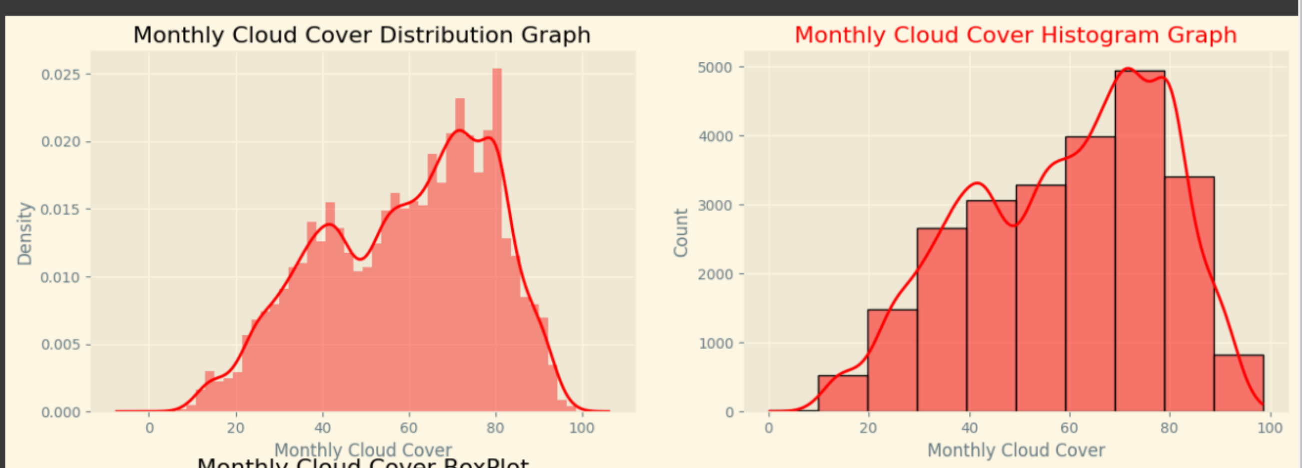
```
df_lumpy.pivot_table(index=['Month','MonthTitle'],columns=['Year','region'],values=['lumpy'],aggfunc='count')
```

	lumpy																				
Year	2011.0	2012.0	2013.0		2014.0		2015.0		2016.0		...	2017.0		2018.0			2019.0		2020.0		2021.0
region	Africa	Asia	Africa	Asia	Africa	Asia	Asia	Europe	Africa	Asia	...	Asia	Europe	Africa	Asia	Europe	Asia	Europe	Africa	Asia	Asia
MonthTitle																					
January	NaN	NaN	NaN	1.0	NaN	11.0	1.0	NaN	2.0	1.0	...	NaN	31.0	NaN	NaN	24.0	NaN	NaN	NaN	NaN	11.0
February	NaN	NaN	NaN	3.0	NaN	4.0	NaN	6.0	NaN	2.0	...	NaN	12.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
March	NaN	NaN	NaN	10.0	NaN	10.0	NaN	2.0	NaN	NaN	...	NaN	6.0	NaN	3.0	12.0	1.0	1.0	NaN	1.0	NaN
April	NaN	NaN	NaN	32.0	NaN	50.0	2.0	8.0	NaN	NaN	...	NaN	5.0	NaN	NaN	NaN	5.0	3.0	NaN	1.0	NaN
May	NaN	2.0	NaN	5.0	NaN	33.0	1.0	NaN	NaN	NaN	...	NaN	17.0	NaN	NaN	2.0	NaN	NaN	NaN	1.0	NaN
June	NaN	NaN	NaN	NaN	NaN	31.0	NaN	NaN	NaN	NaN	...	NaN	31.0	NaN	NaN	NaN	2.0	1.0	NaN	2.0	NaN
July	1.0	2.0	NaN	4.0	NaN	93.0	NaN	NaN	2.0	6.0	...	1.0	20.0	10.0	NaN	13.0	8.0	3.0	NaN	13.0	NaN
August	NaN	4.0	1.0	NaN	NaN	13.0	1.0	15.0	NaN	2.0	...	NaN	14.0	NaN	1.0	8.0	4.0	8.0	NaN	27.0	NaN
September	NaN	1.0	NaN	4.0	6.0	NaN	2.0	63.0	2.0	NaN	...	NaN	4.0	NaN	NaN	11.0	8.0	6.0	NaN	8.0	NaN
October	3.0	3.0	NaN	NaN	NaN	19.0	NaN	28.0	NaN	NaN	...	NaN	10.0	NaN	1.0	4.0	21.0	1.0	NaN	13.0	NaN

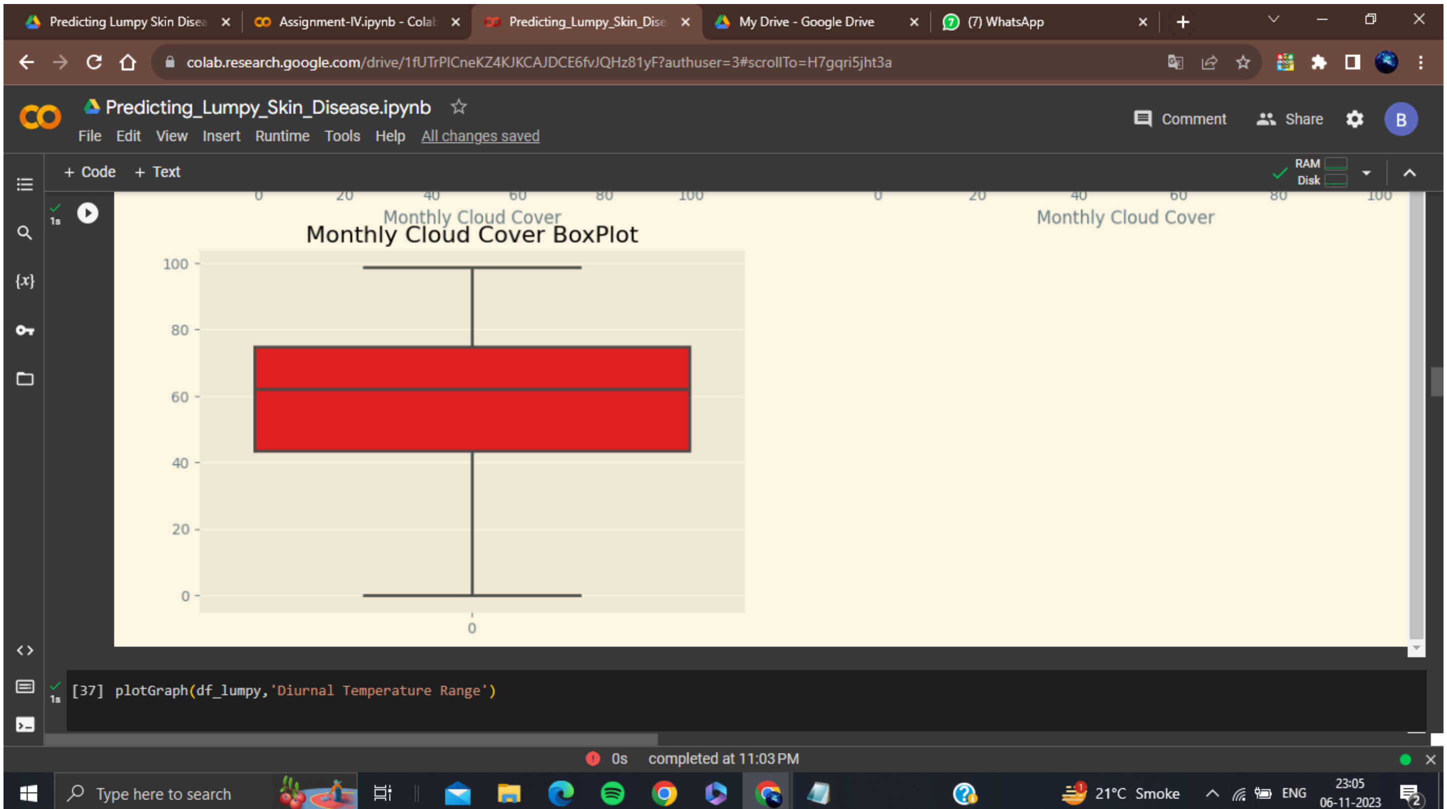
0s completed at 11:03 PM

Predicting_Lumpy_Skin_Disease.ipynb

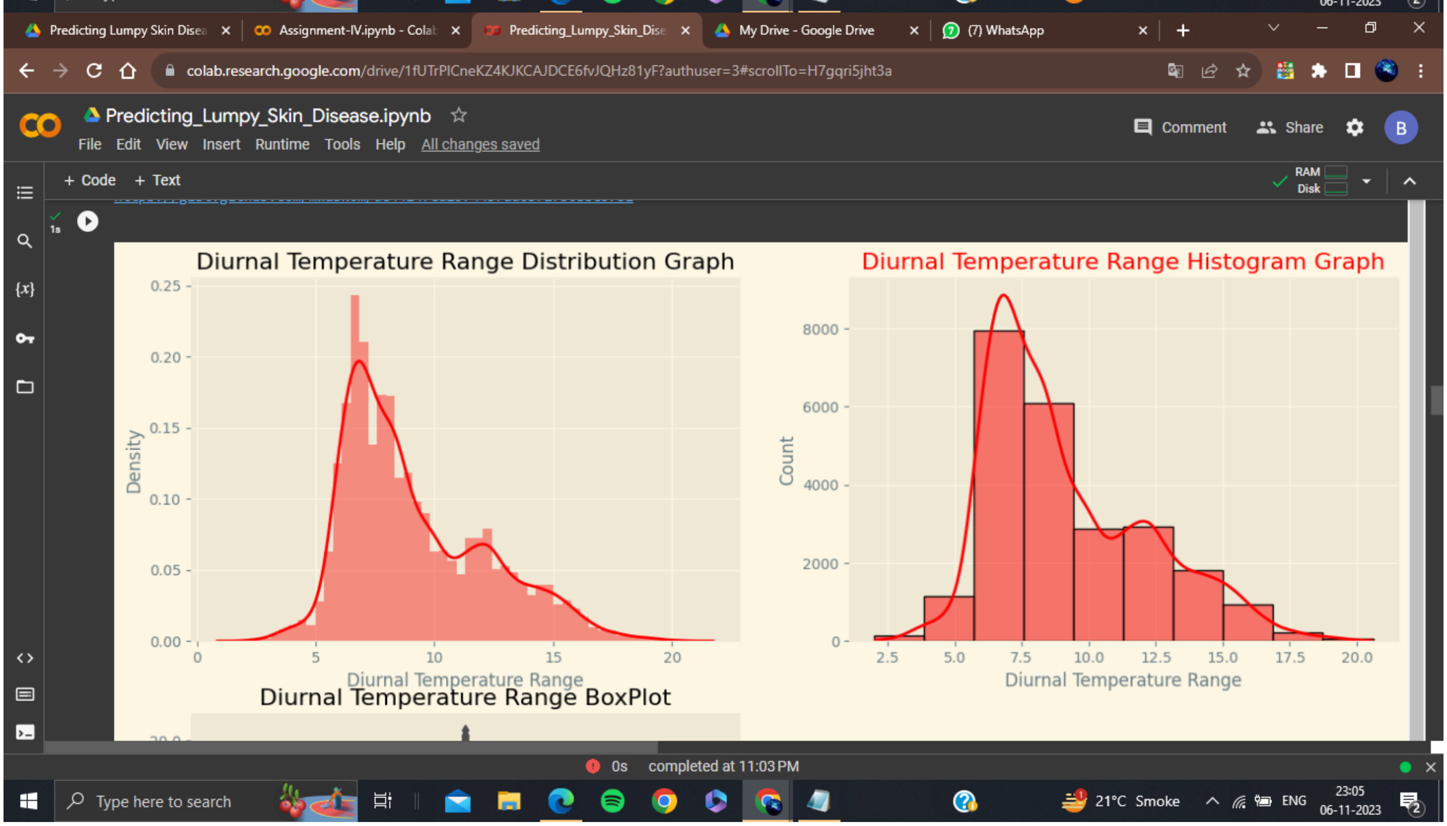
For a guide to updating your code to use the new functions, please see https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751



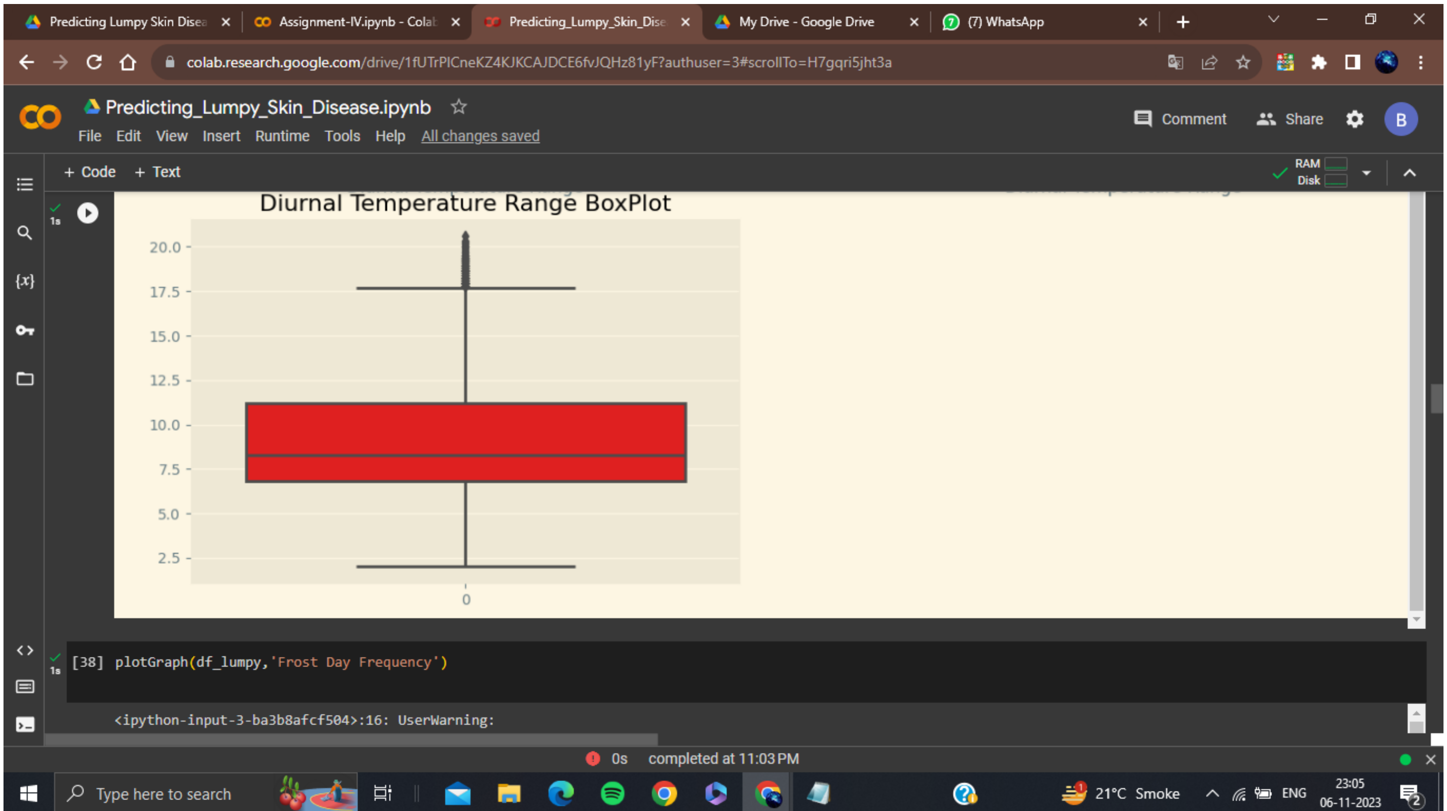
0s completed at 11:03 PM

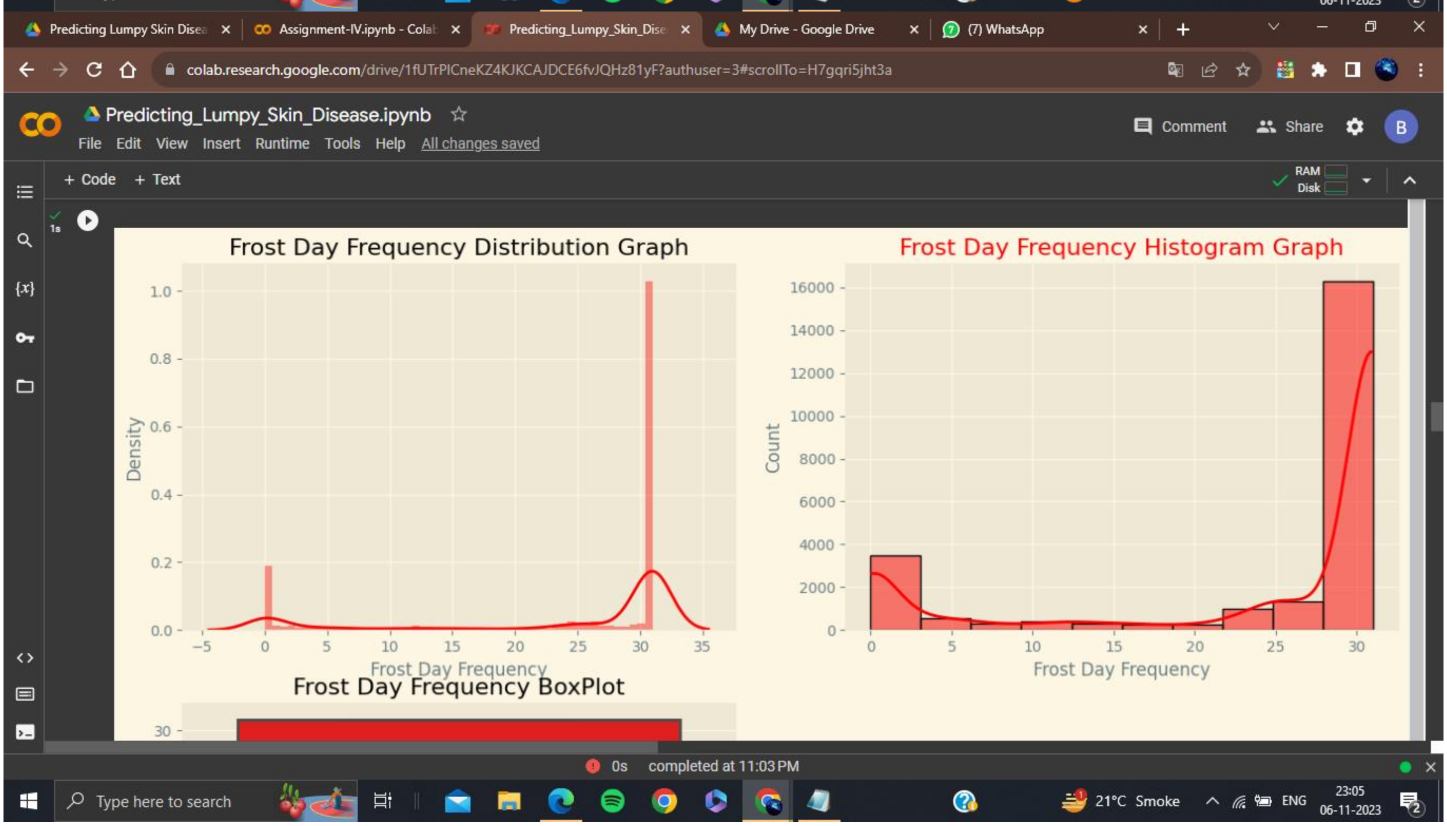


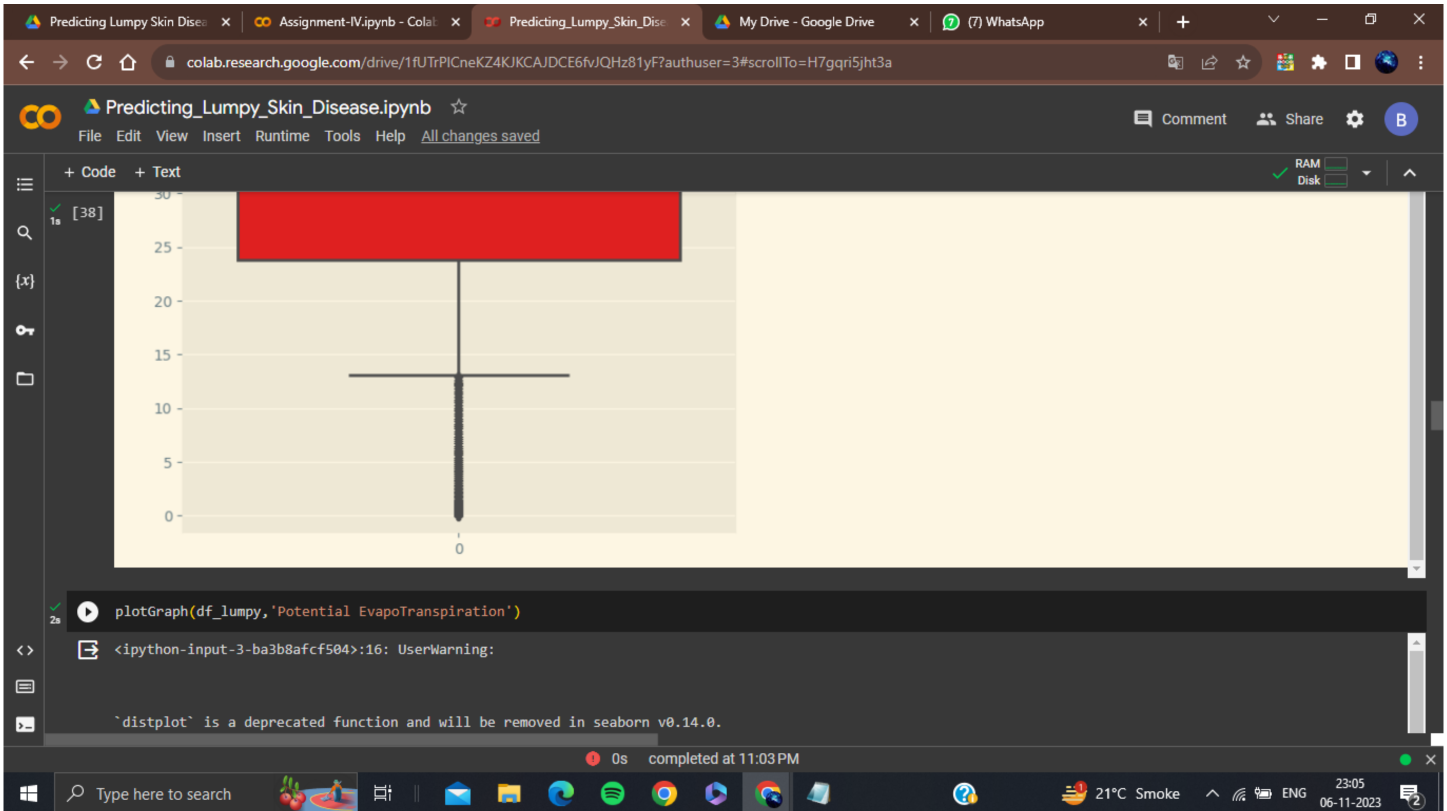
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Monthly Cloud Cover
Monthly Cloud Cover BoxPlot
[37] plotGraph(df_lumpy,'Diurnal Temperature Range')
0s completed at 11:03 PM
Type here to search
21°C Smoke ENG 23:05 06-11-2023

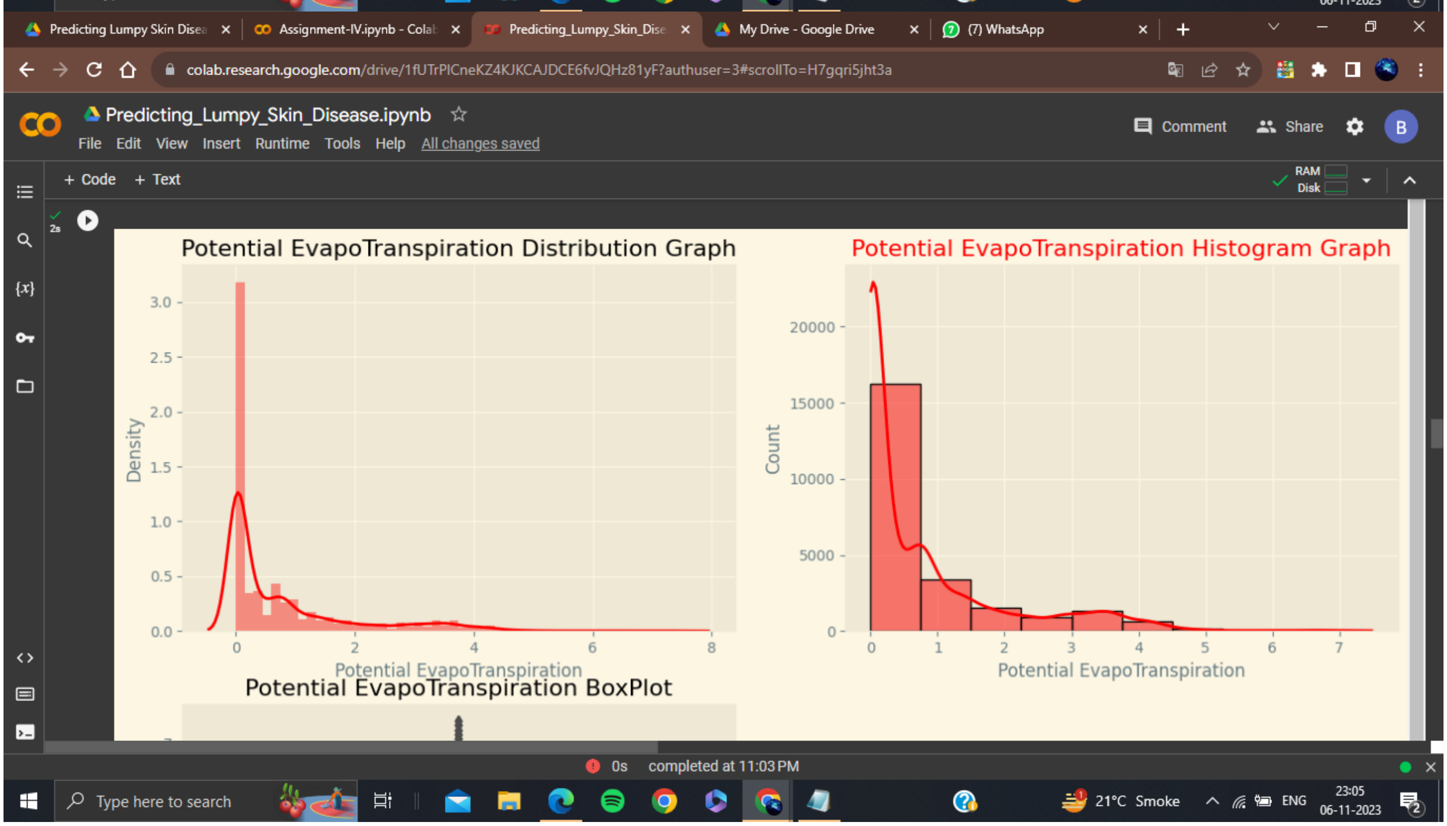


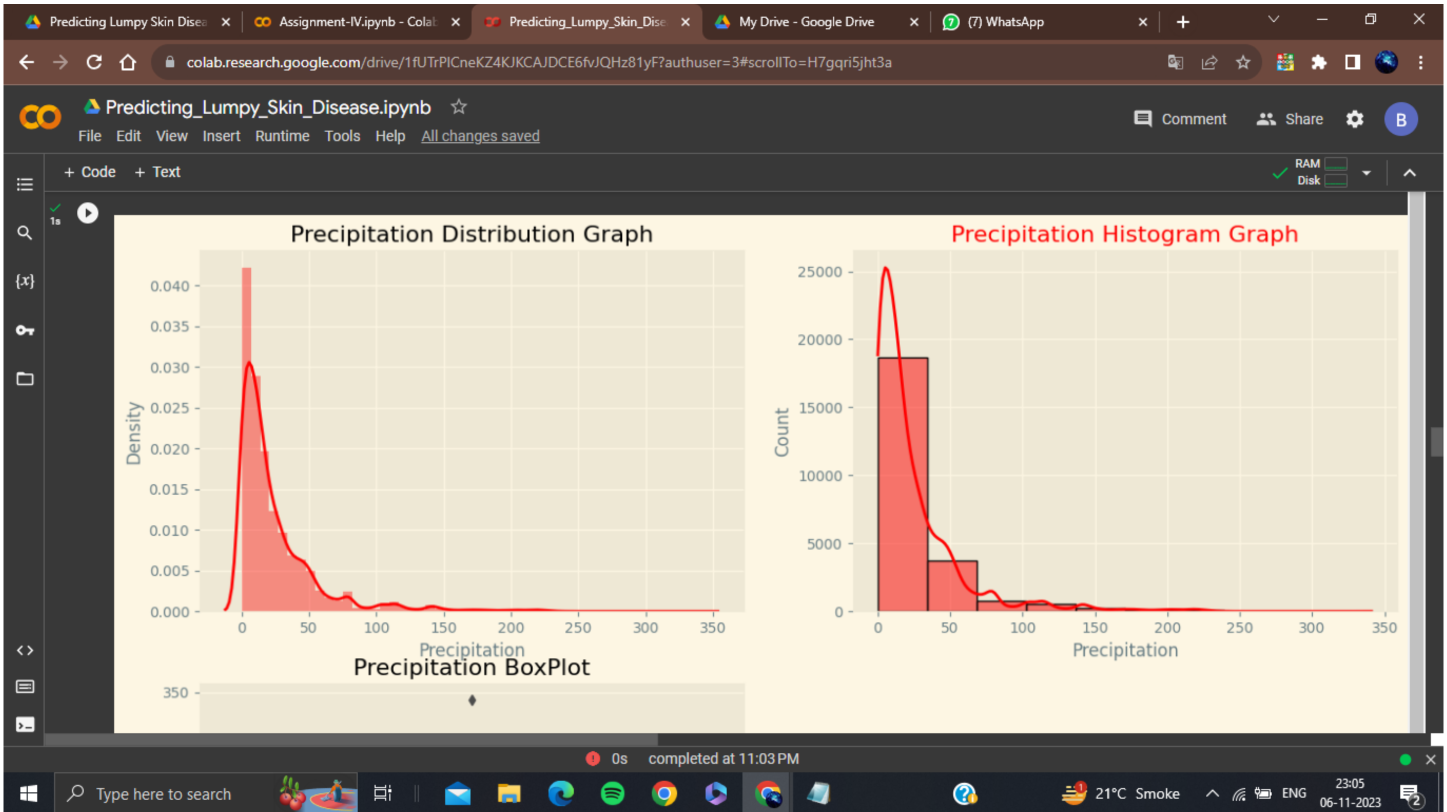
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Diurnal Temperature Range Distribution Graph
Density
Diurnal Temperature Range
Diurnal Temperature Range Histogram Graph
Count
Diurnal Temperature Range
Diurnal Temperature Range BoxPlot
0s completed at 11:03 PM
Type here to search
21°C Smoke ENG 23:05 06-11-2023


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Diurnal Temperature Range BoxPlot
[38] plotGraph(df_lumpy,'Frost Day Frequency')
<ipython-input-3-ba3b8afcf504>:16: UserWarning:
completed at 11:03 PM

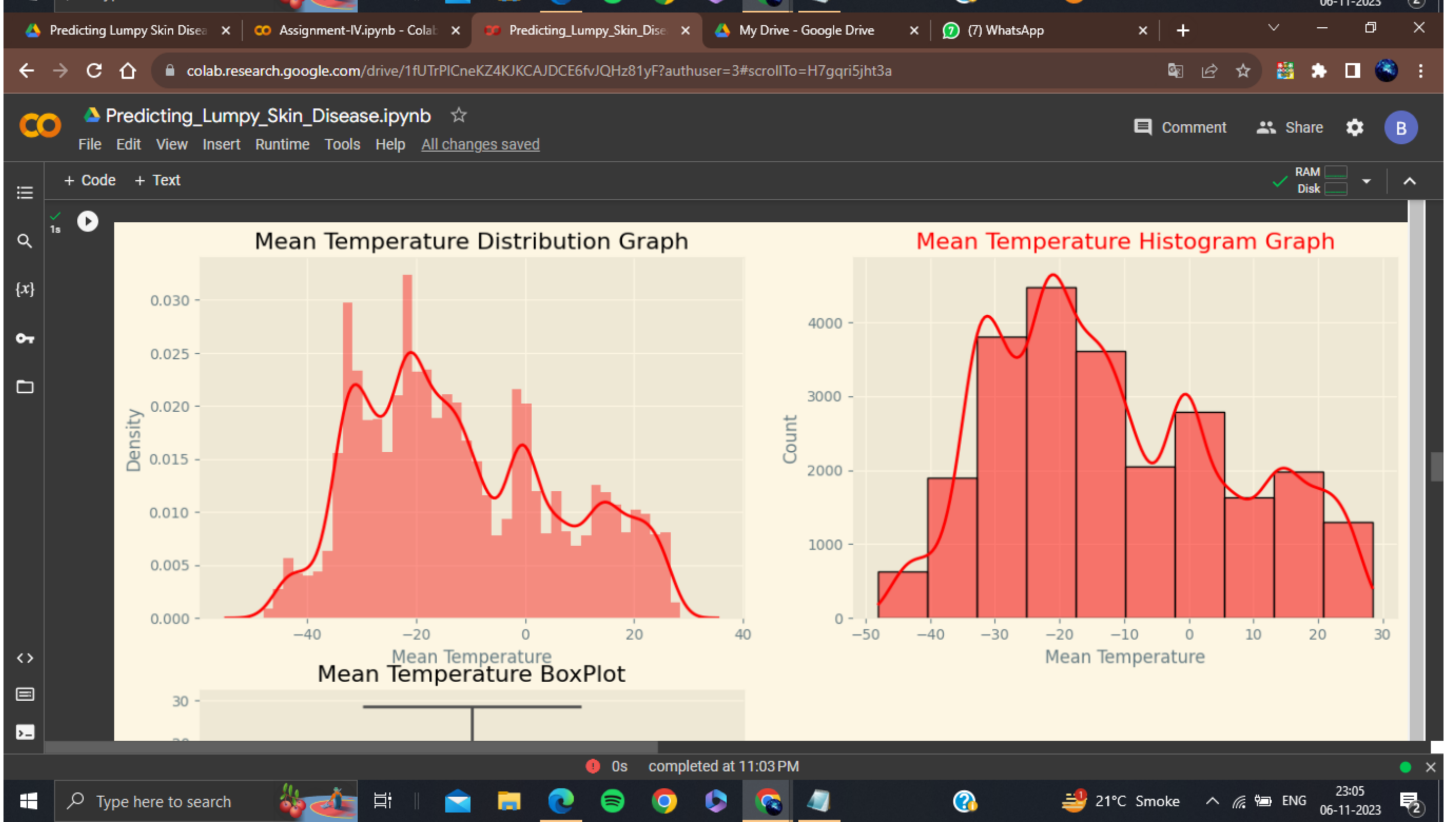

Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Frost Day Frequency Distribution Graph
Density
Frost Day Frequency
Frost Day Frequency Histogram Graph
Count
Frost Day Frequency
Frost Day Frequency BoxPlot
completed at 11:03 PM


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[38]
plotGraph(df_lumpy,'Potential EvapoTranspiration')
<ipython-input-3-ba3b8afcf504>:16: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
0s completed at 11:03 PM

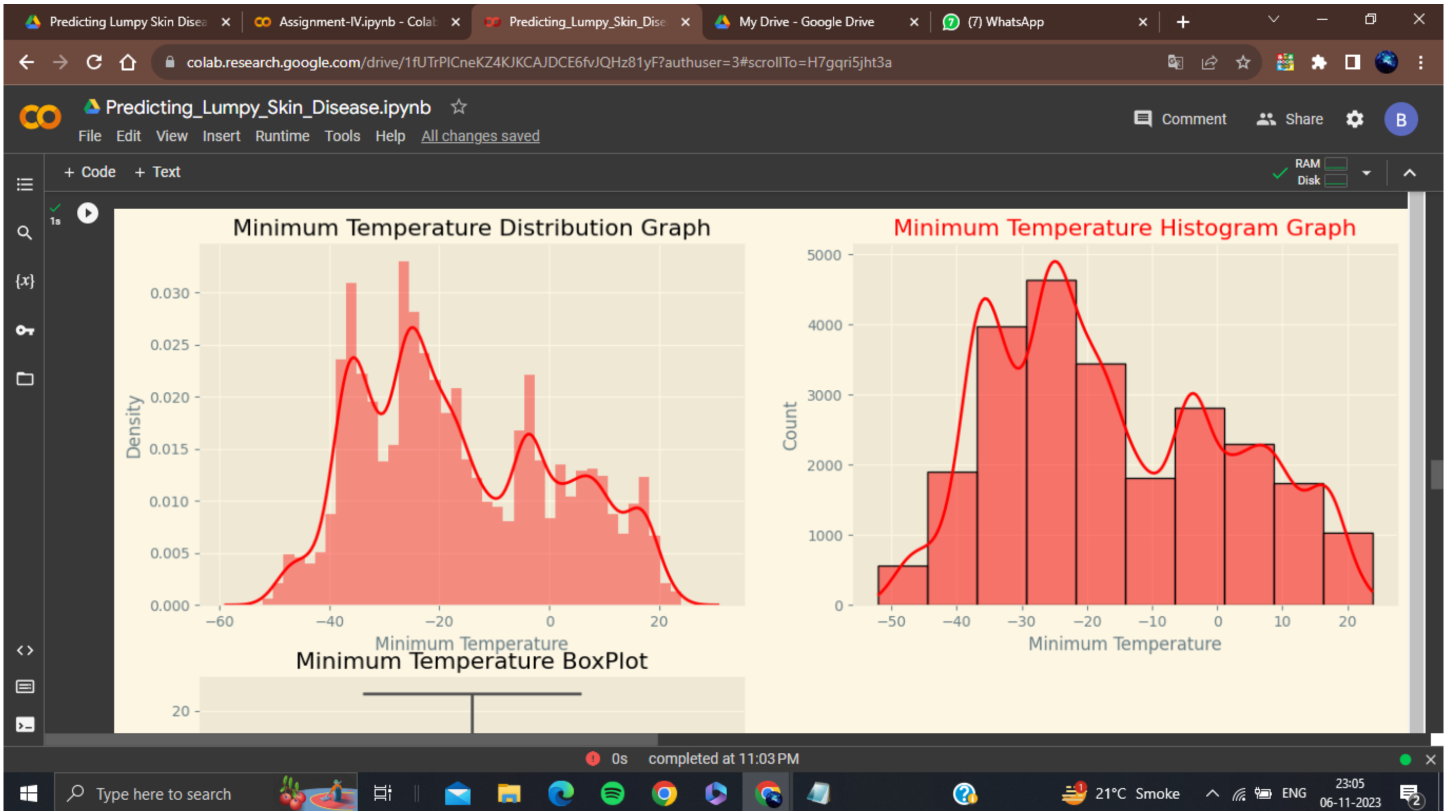

Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Potential EvapoTranspiration Distribution Graph
Density
Potential EvapoTranspiration
Potential EvapoTranspiration Histogram Graph
Count
Potential EvapoTranspiration
Potential EvapoTranspiration BoxPlot
0s completed at 11:03 PM

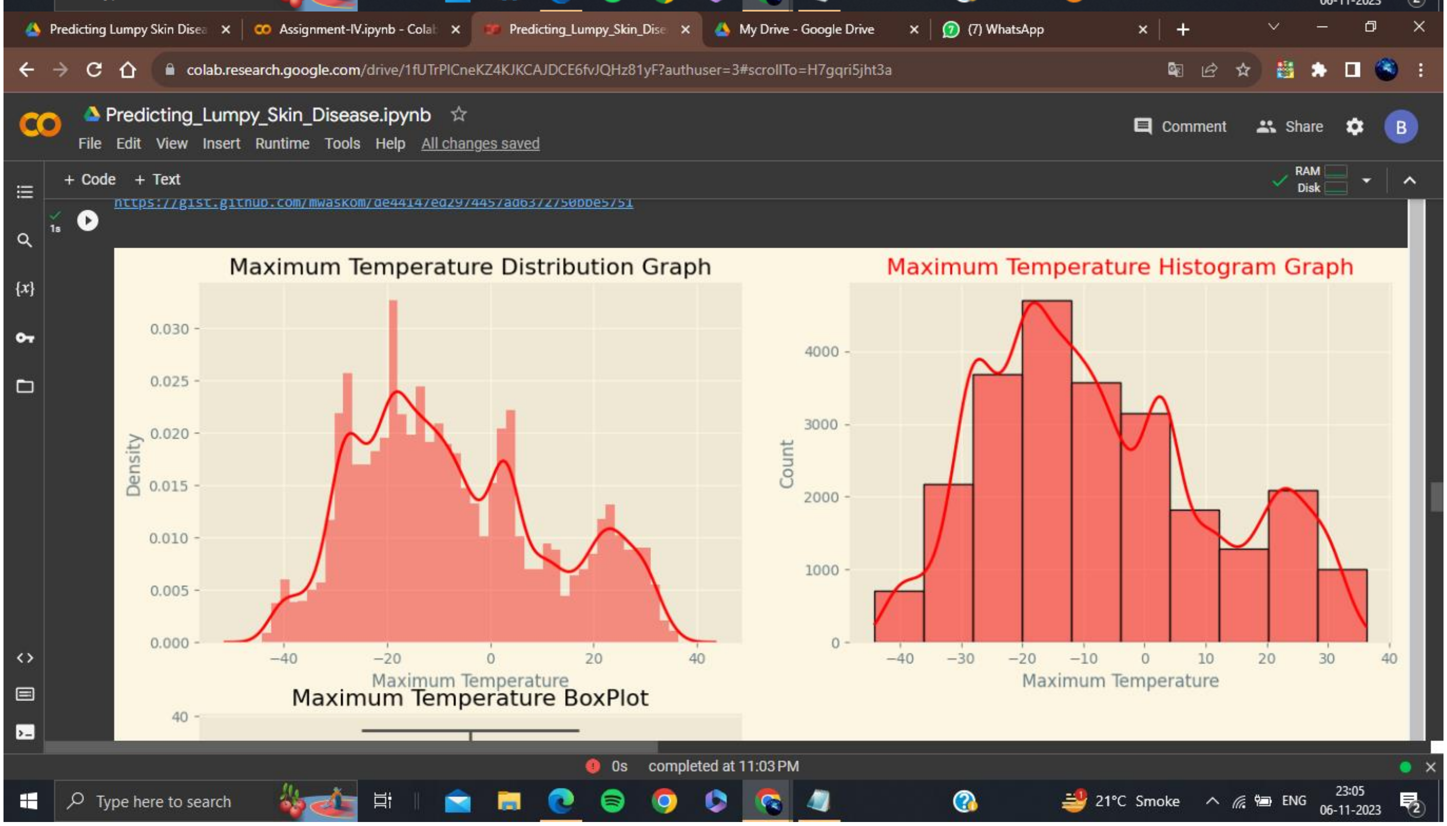


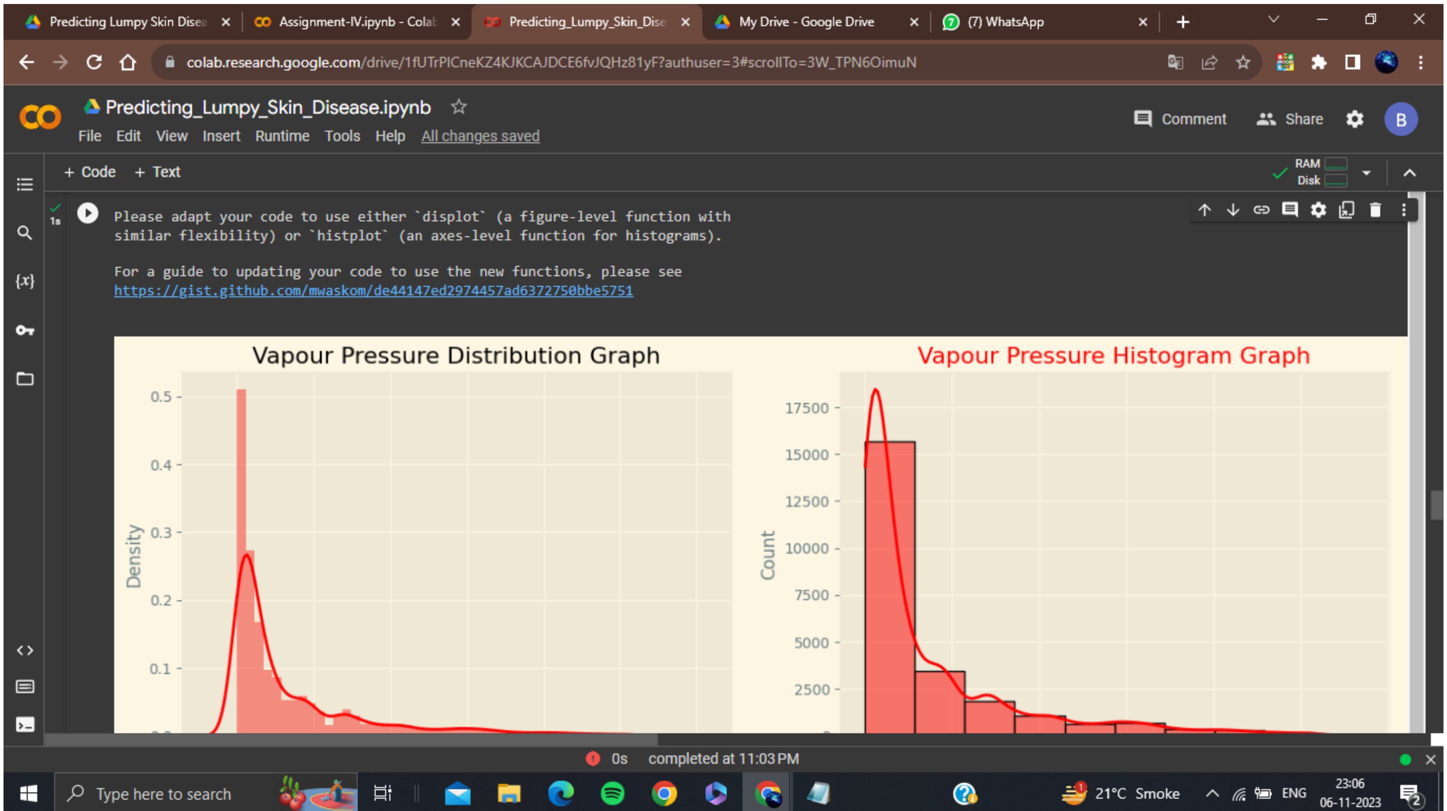
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Precipitation Distribution Graph
Density
Precipitation
Precipitation Histogram Graph
Count
Precipitation
Precipitation BoxPlot
completed at 11:03 PM



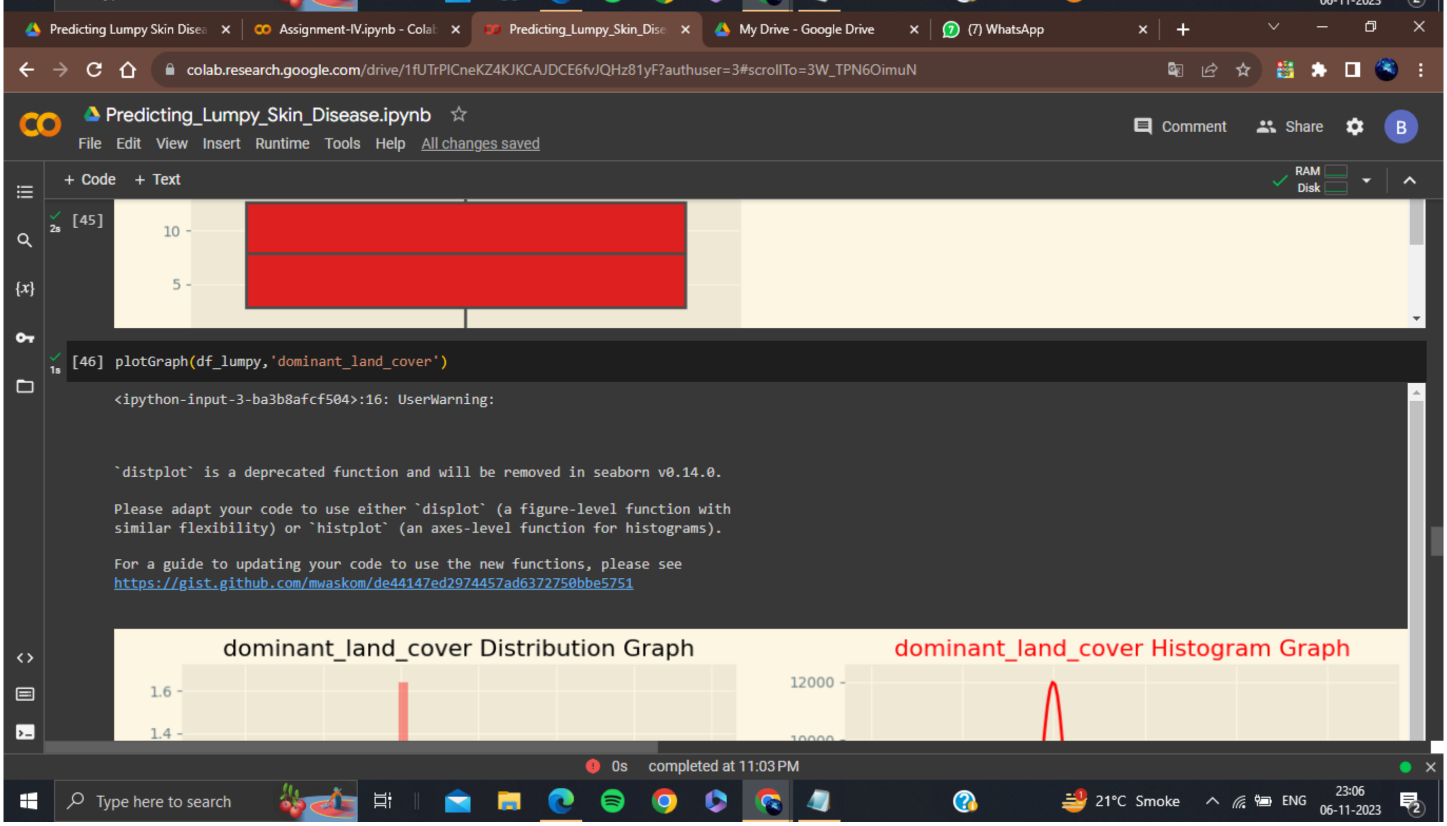
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Mean Temperature Distribution Graph
Density
Mean Temperature
Mean Temperature Histogram Graph
Count
Mean Temperature
Mean Temperature BoxPlot
completed at 11:03 PM


Predicting Lumpy Skin Disea
Assignment-IV.ipynb - Colab
Predicting_Lumpy_Skin_Dise
My Drive - Google Drive
(7) WhatsApp
colab.research.google.com/drive/1fUTrPICneKZ4KJKCAJDCE6fvJQHz81yF?authuser=3#scrollTo=H7gqri5jht3a
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Minimum Temperature Distribution Graph
Density
Minimum Temperature
Minimum Temperature Histogram Graph
Count
Minimum Temperature
Minimum Temperature BoxPlot
0s completed at 11:03 PM
Type here to search
21°C Smoke
ENG
23:05
06-11-2023

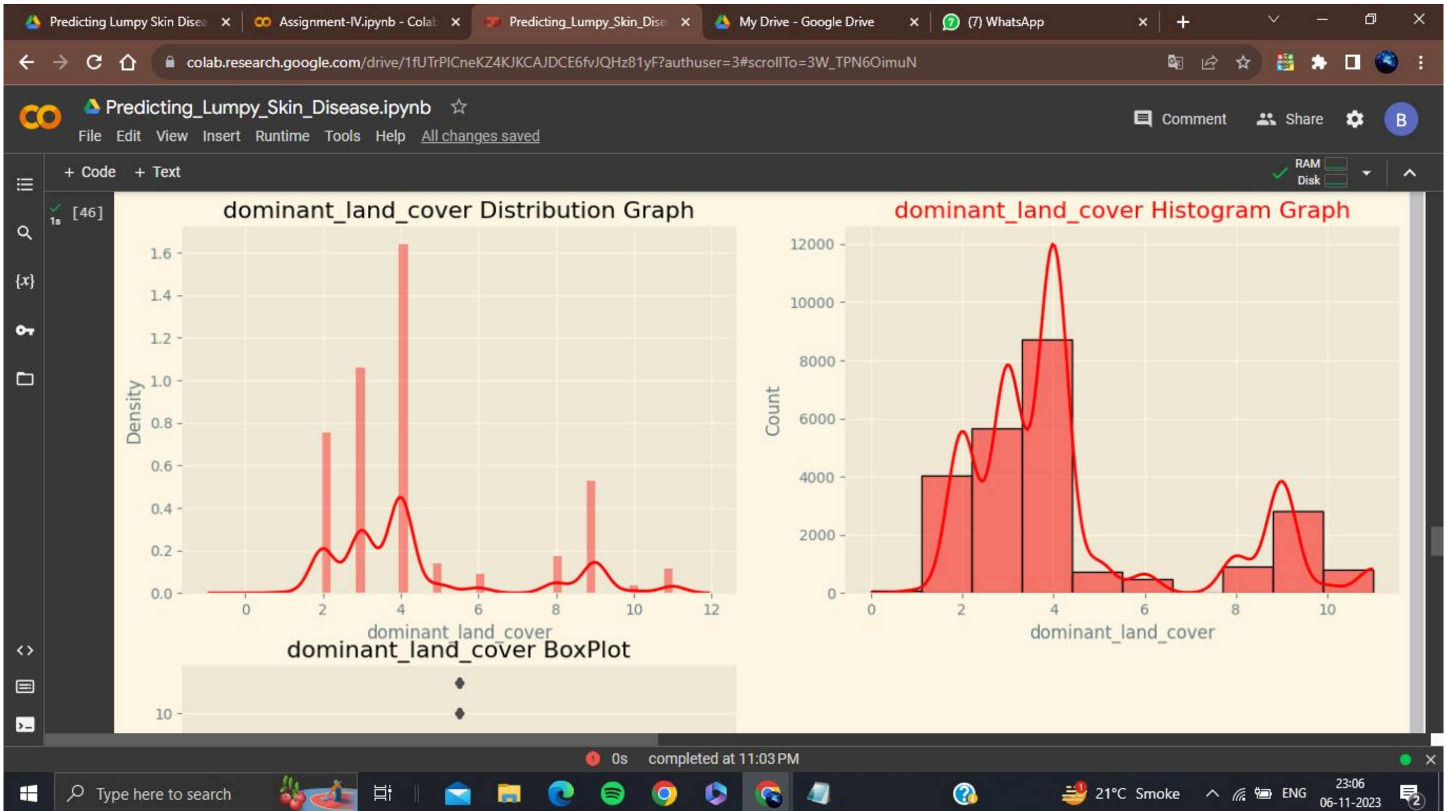

Predicting Lumpy Skin Disea
Assignment-IV.ipynb - Colab
Predicting_Lumpy_Skin_Dise
My Drive - Google Drive
(7) WhatsApp
colab.research.google.com/drive/1fUTrPICneKZ4KJKCAJDCE6fvJQHz81yF?authuser=3#scrollTo=H7gqri5jht3a
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Maximum Temperature Distribution Graph
Density
Maximum Temperature
Maximum Temperature Histogram Graph
Count
Maximum Temperature
Maximum Temperature BoxPlot
0s completed at 11:03 PM
Type here to search
21°C Smoke
ENG
23:05
06-11-2023

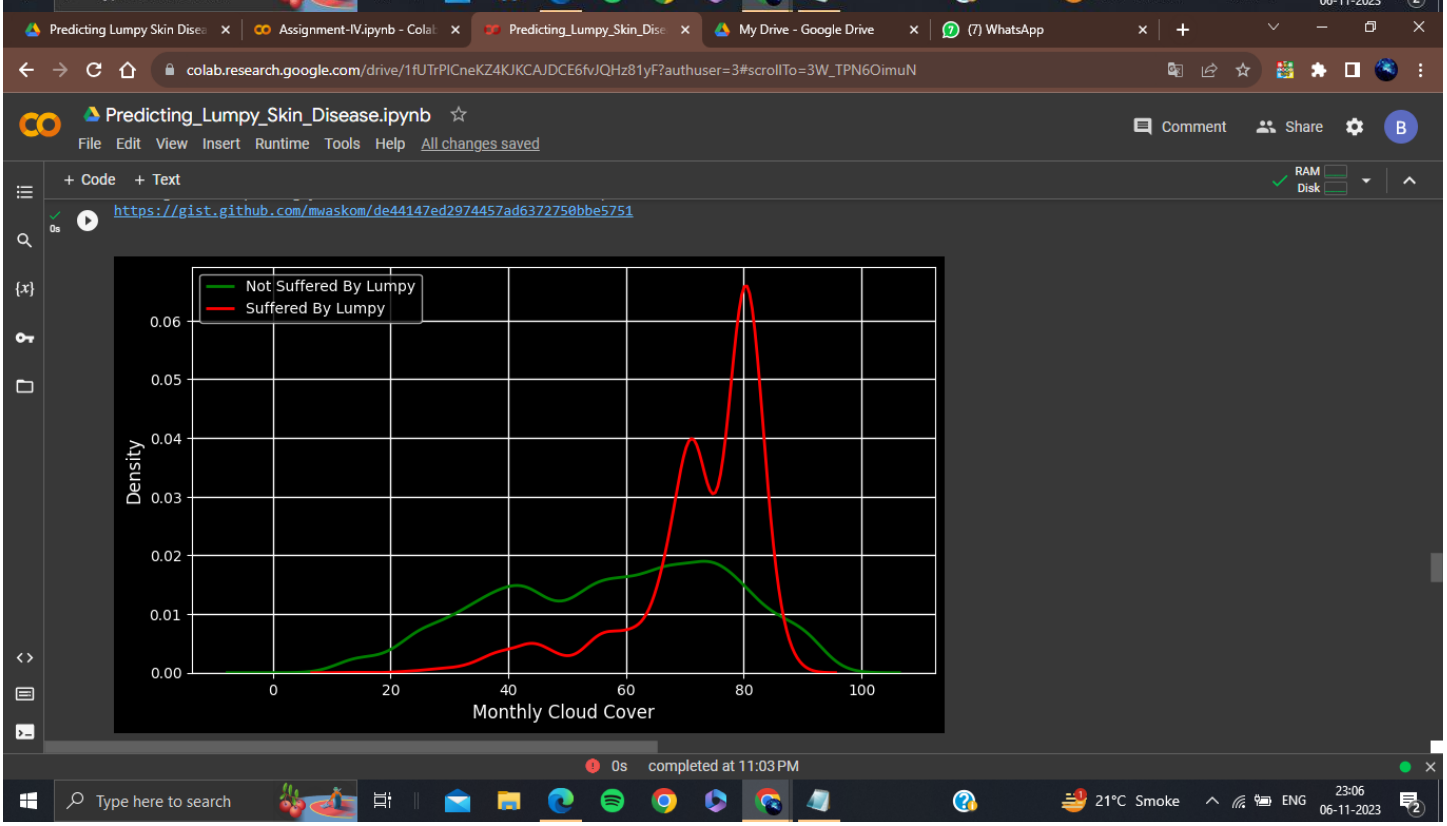


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
Vapour Pressure Distribution Graph
Density
Vapour Pressure Histogram Graph
Count
0s completed at 11:03 PM



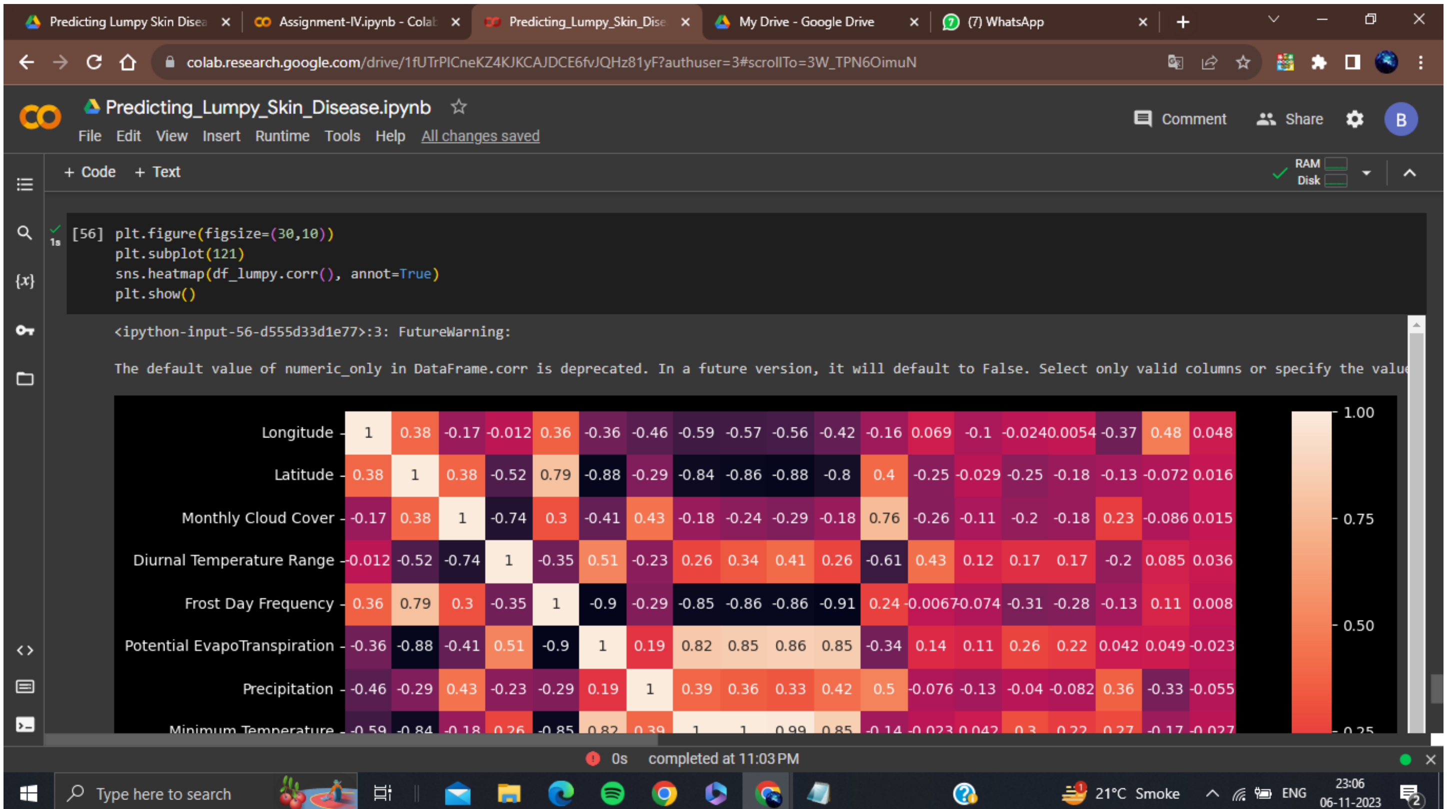
Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[45]
[46] plotGraph(df_lumpy,'dominant_land_cover')
<ipython-input-3-ba3b8afcf504>:16: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
dominant_land_cover Distribution Graph
dominant_land_cover Histogram Graph
0s completed at 11:03 PM

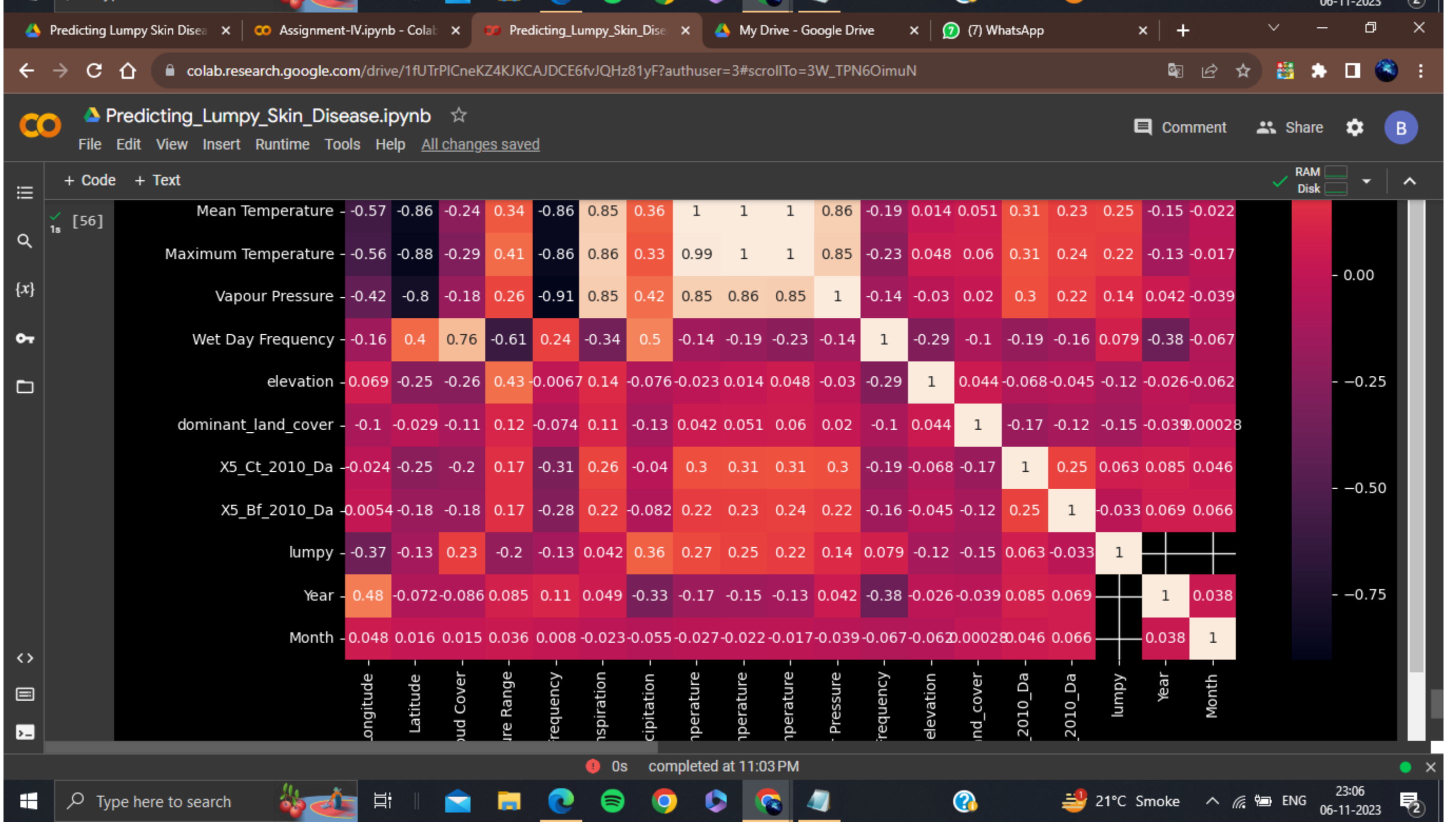

Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
[46]
dominant_land_cover Distribution Graph
Density
dominant_land_cover
dominant_land_cover Histogram Graph
Count
dominant_land_cover
dominant_land_cover BoxPlot
0s completed at 11:03 PM
21°C Smoke
23:06
06-11-2023


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
Not Suffered By Lumpy
Suffered By Lumpy
Density
Monthly Cloud Cover
0s completed at 11:03 PM
21°C Smoke
23:06
06-11-2023


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
[48]
Not Suffered By Lumpy
Suffered By Lumpy
Density
Diurnal Temperature Range
[49] plotChart(df_lumpy,'lumpy','Potential EvapoTranspiration')
completed at 11:03 PM


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
[49]
Suffered By Lumpy
Density
Potential EvapoTranspiration
[50] plotChart(df_lumpy,'lumpy','Mean Temperature')
<ipython-input-3-ba3b8afcf504>:5: UserWarning:
completed at 11:03 PM


Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
[56] plt.figure(figsize=(30,10))
plt.subplot(121)
sns.heatmap(df_lumpy.corr(), annot=True)
plt.show()
<ipython-input-56-d555d33d1e77>:3: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the valu
Longitude
Latitude
Monthly Cloud Cover
Diurnal Temperature Range
Frost Day Frequency
Potential EvapoTranspiration
Precipitation
1.00
0.75
0.50
completed at 11:03 PM

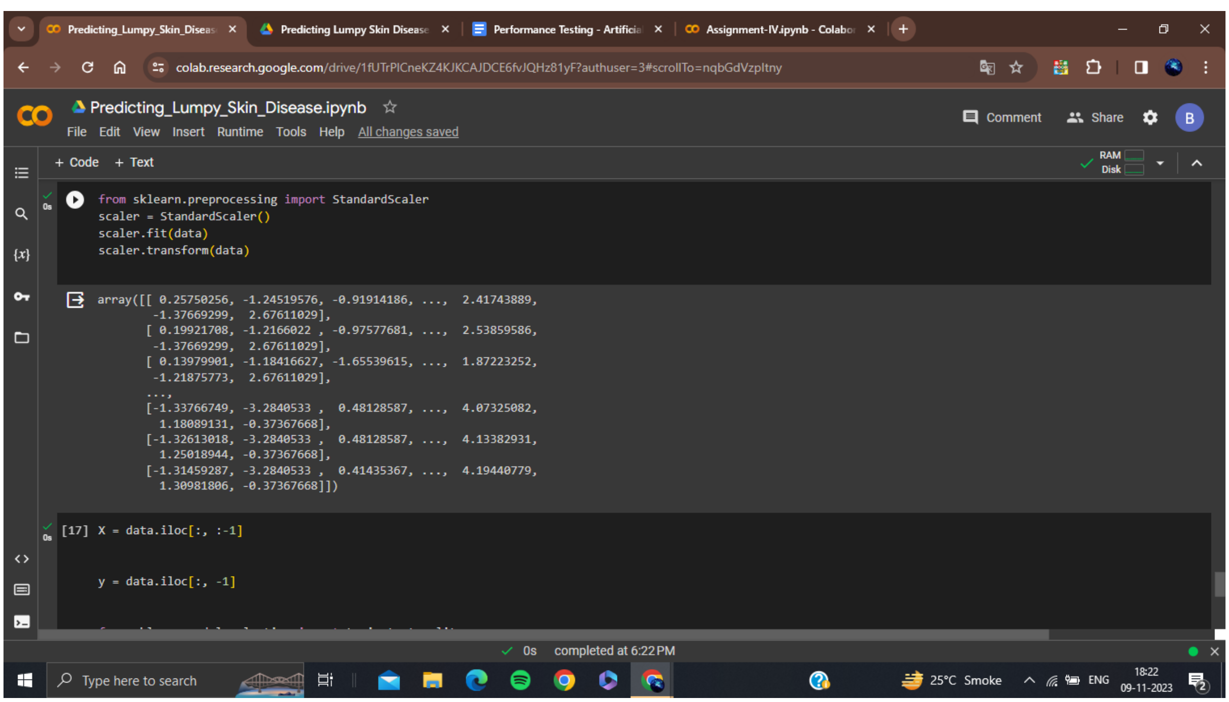

Predicting_Lumpy_Skin_Disease.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Mean Temperature
Maximum Temperature
Vapour Pressure
Wet Day Frequency
elevation
dominant_land_cover
X5_Ct_2010_Da
X5_Bf_2010_Da
lumpy
Year
Month
0.00
−0.25
−0.50
−0.75
completed at 11:03 PM

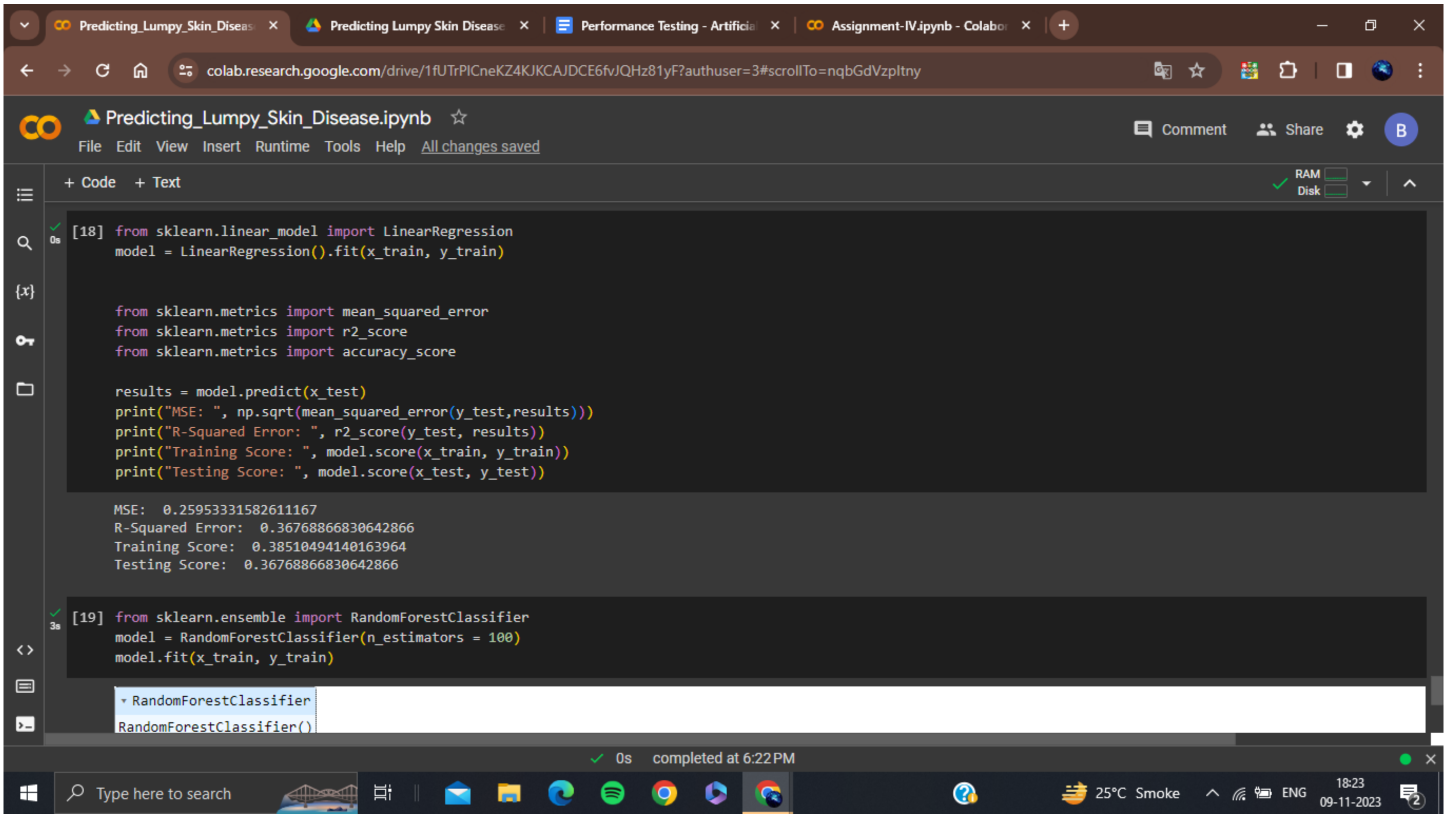
Activity 2: Training the Model with Multiple Algorithms

With the dataset now cleaned and prepared, we proceed to construct our model. To ensure a comprehensive evaluation, we train our data using multiple algorithms. In this particular project, we have selected four classification algorithms to apply. By employing this ensemble of algorithms, we can leverage their unique strengths and characteristics, enabling us to obtain a more robust and accurate model.

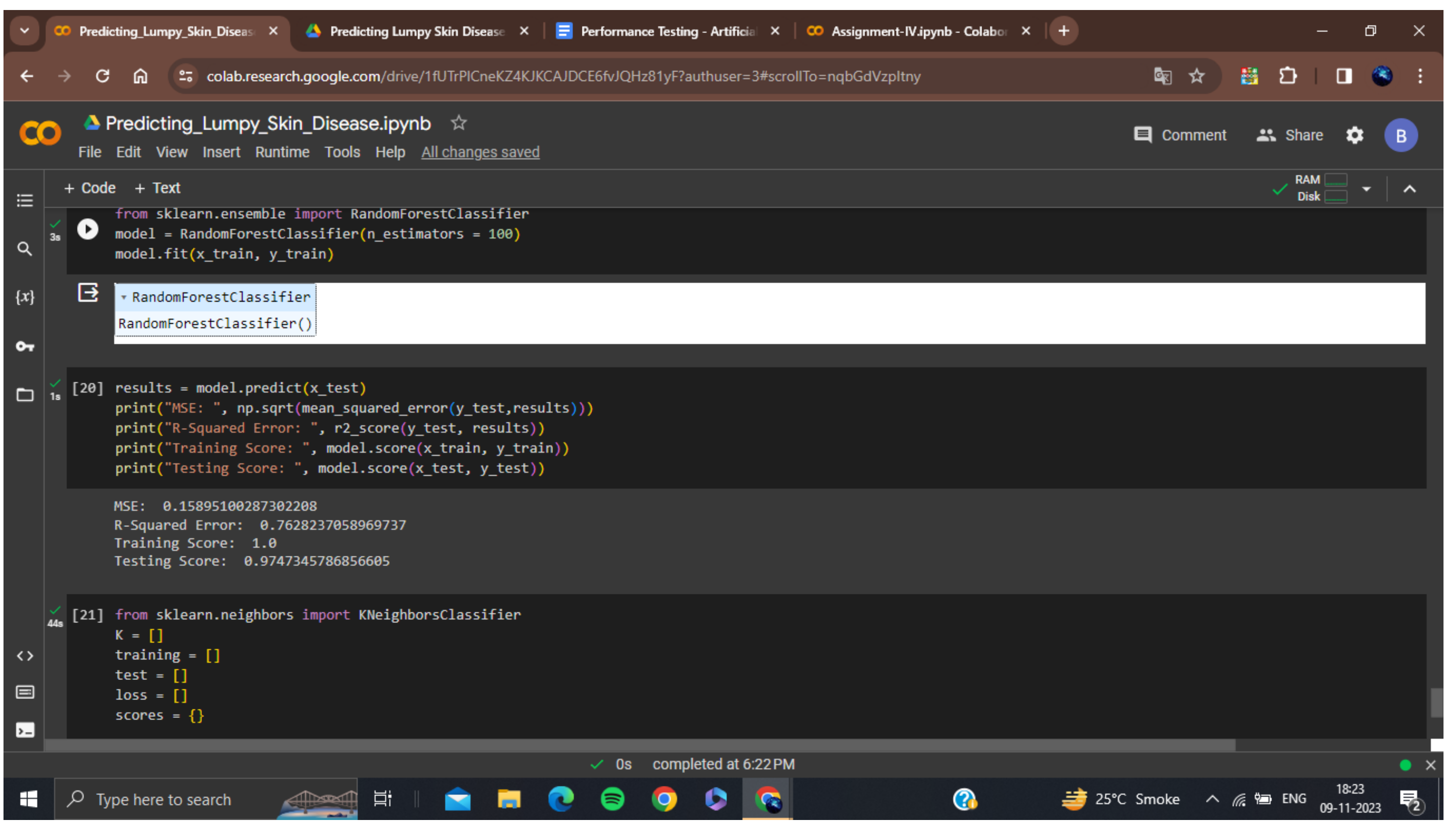
During the training process, we carefully monitor the performance of each algorithm. Based on their respective performance metrics, we identify the best-performing model. This superior model is then saved, ensuring that we retain the optimal solution for subsequent use and further analysis.

Acticity 2.1:Linear Regression Model

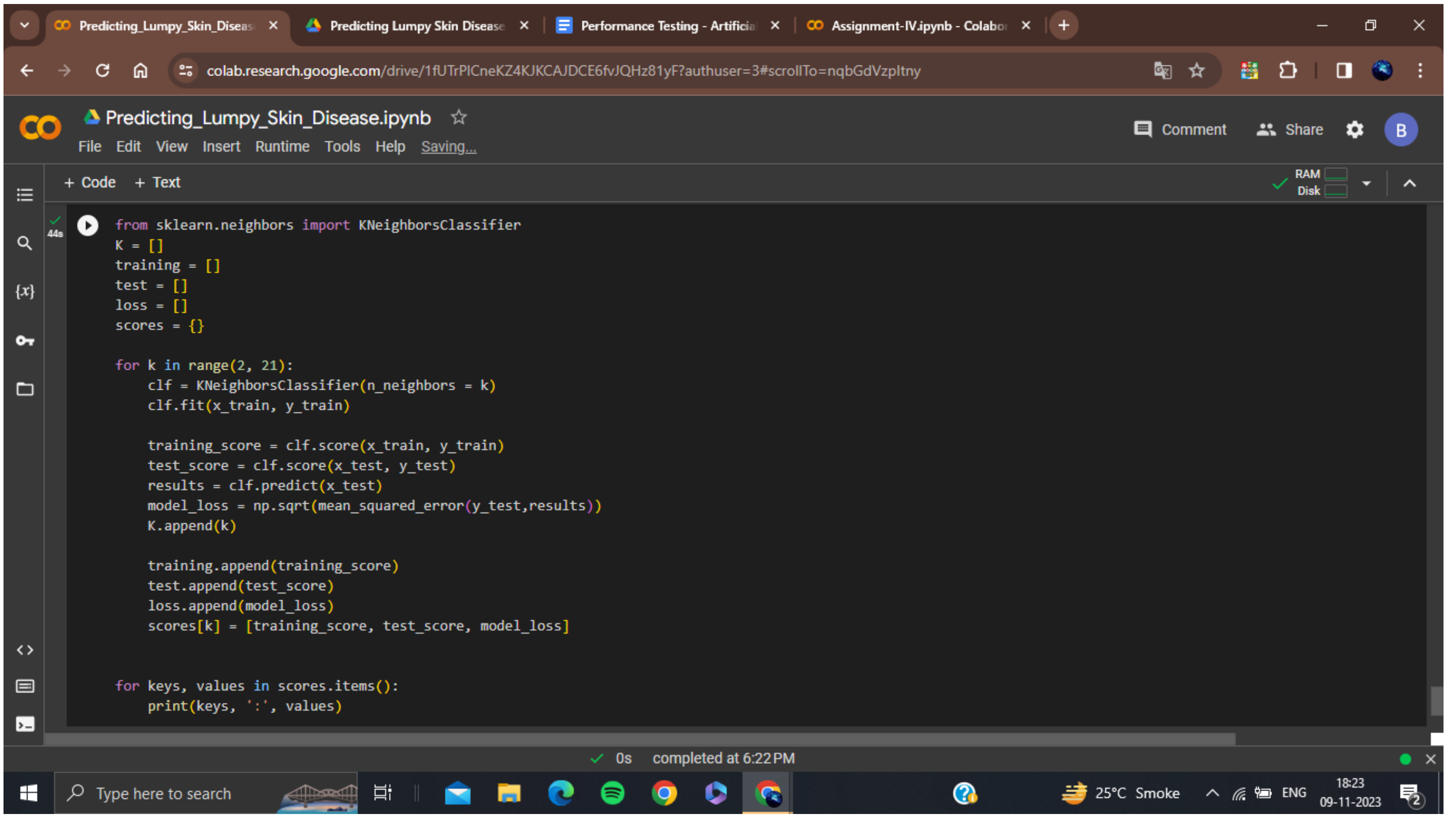


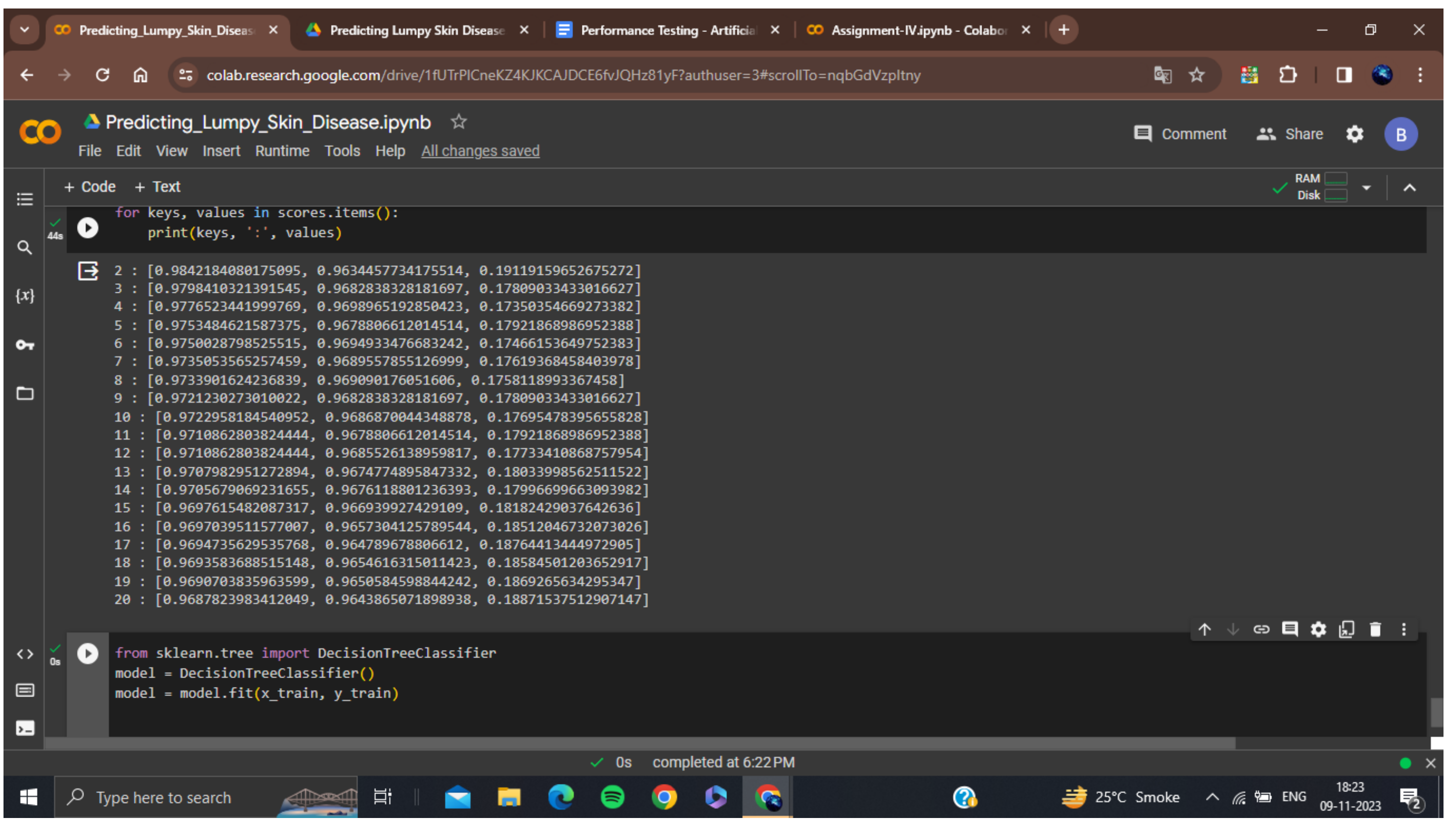


Activity2.2 Random Forest Classifier

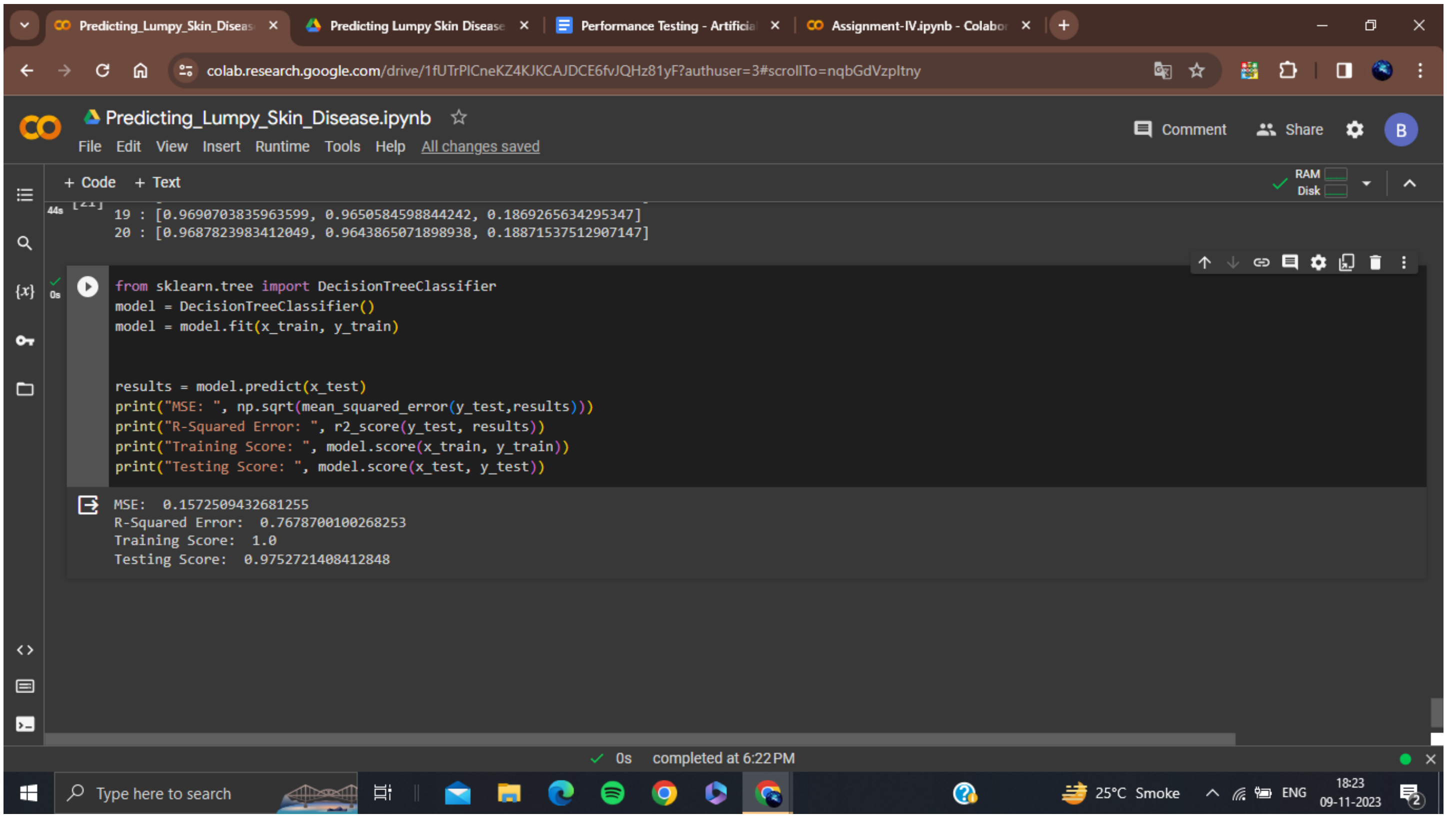


Activity 2.3 K Neighbor Classifier



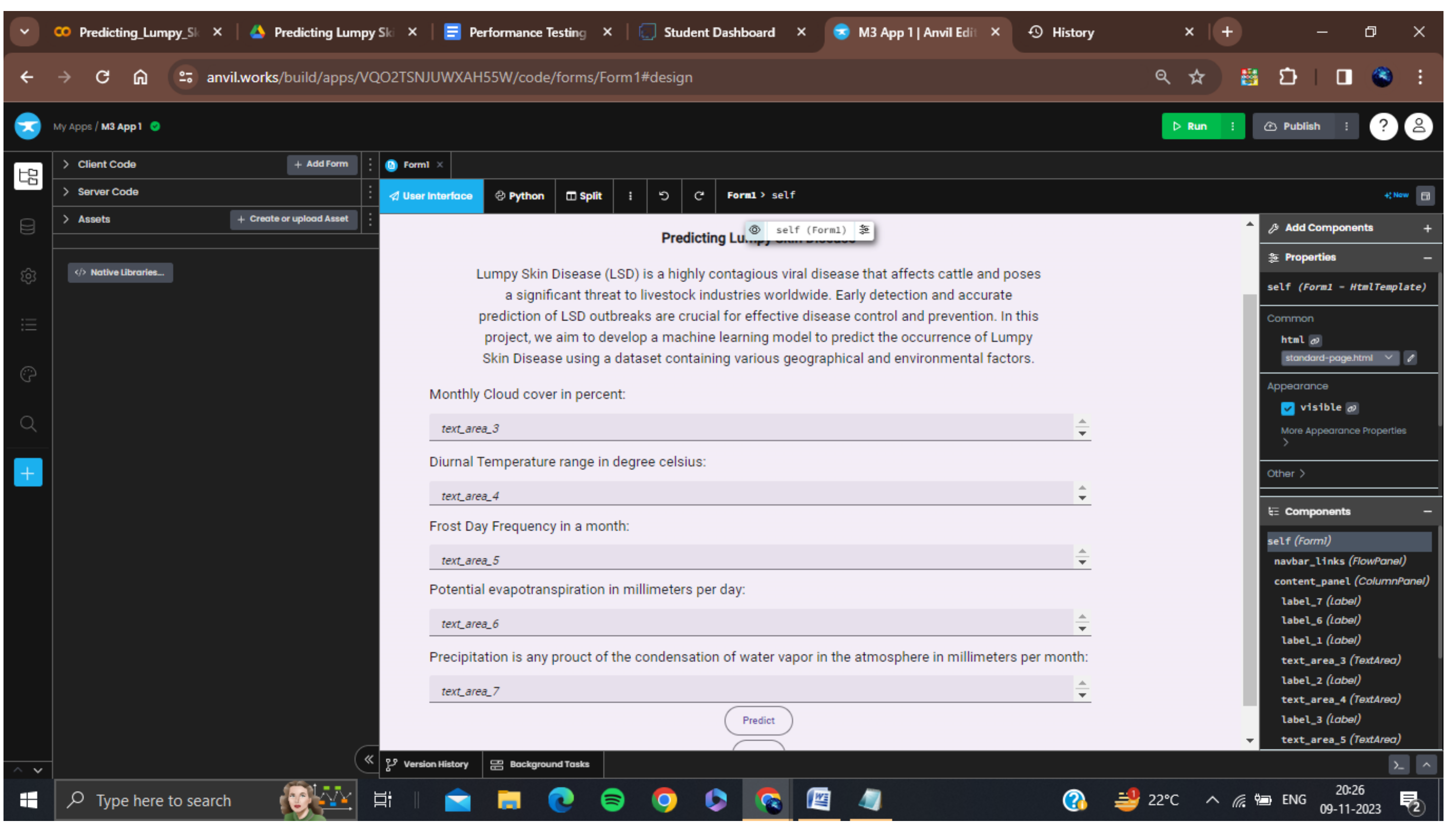


Activity 2.4 Decision Tree Classifier



Milestone 6: Model Deployment

Using Anvil and Google colab



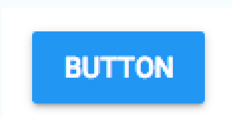
Step 1:Design the page

To classify the species of iris a flower comes from, we need to collect several measurements, so let's design the user interface for entering that data.

We construct the UI by dragging-and-dropping components from the Toolbox. Let's start by dropping a Card into our form – this will be a neat container for the other components. Then let's add a Label and a TextBox into the card component:

Next we will set up the `Label` and `TextBox` components to collect enter the sepal length.

Select the `Label` we just added and, in the properties panel on the right, change the text to 'Sepal length: ' and align the text to the right.

Next, let's add a Button 

to run the classifier. Name it `predict_button` and change the text to 'Categorise'. Clicking this button will trigger a Python function to send the iris measurements to our Colab notebook. (We'll set that up in a moment.)

Step 2:Set up the predict button

We want our `predict_button` to do something when it's clicked, so let's add a click event.

With the button selected, go to the bottom of the properties panel. Then click the blue button with two arrows in it next to the click event box. This will open our code view and create a function called `predict_button_click()`. From now on, every time the button is clicked by a user, this function will be called.

Step 3:Enable the uplink

From the Anvil editor, let's enable the Uplink. This gives us everything we need to connect our web app to our Colab notebook. Select the blue '+' button in the Sidebar Menu to open the list of available services. Then add the `Uplink` and click 'Enable Server Uplink':

This will then give us an Uplink key we can use in our Google Colab notebook, to connect to this app.

Now let's install the Uplink in our Colab environment, and connect our script using the key we just created.

Step 4: Install the Uplink Library in our Colab Environment

In the next few steps, we will be connecting a Colab notebook to the web app we have built. For simplicity, I've created a notebook that already handles the iris classification for us. Make a copy of the following notebook to follow along:

The first thing we need to do is install the anvil-uplink **library in our Colab environment. Let's add** !pip install anvil-uplink **to the top of our notebook.**

Step 5:Connecting our Script

Now that the Uplink library will be installed when we start our notebook, we can connect our notebook in the same way as any other Uplink script.

Start by importing the `anvil.server` module:

Then connect to the Uplink:

Replace "your-uplink-key" with the Uplink key from your app.

That's it! When we run our notebook, it will now connect to our web app via the Uplink. Next, let's create a function we can call from our Anvil app.

Step 6 - Creating a callable function

With a classification model built and trained, we can create a function that takes our iris data and returns the name of the iris species. Let's create a **`predict_`** function and add **`@anvil.server.callable`** so it is available to call from our app. Finally at the end of our notebook we will call the **`wait_forever()`** function. This keeps our notebook running and allows our app to call functions indefinitely.

Step 9 - Deploying your model

Downloading your model

We'll start by going back into our Colab notebook. At the end of the cell that builds and trains the iris classification model, we'll import the **`joblib`** library and the files module from google.colab.

Uploading the model to our app

Now, back in the Anvil editor, let's add the Data Files service. Select the blue '+' button in the [Sidebar Menu]() and add [Data Files]().

Next, we can upload our model as a file by clicking the 'Upload' button and selecting the model we downloaded earlier.

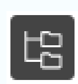
Configuring your server environment

With our model uploaded, we need to configure our app's server environment to include all the packages we need to use the model.

We'll start by selecting settings icon  from the Sidebar Menu and opening 'Python versions'.

Then, in the Python version dropdown, select 'Python 3.10'. Under 'Base packages', choose the 'Machine Learning' base image. This includes all of the packages we'll need to run the model.

Using your hosted model

Create a Server Module by selecting the App Browser  in the [Sidebar Menu]() and clicking '+ Add Server Module'.

Inside the predict_ function, we'll reconstruct our model using **`joblib.load()`**. We will get the path to the model file on disk using **`data_files['knn.skmodel']`**. Lastly, we'll get the classification using the same code we used in our Colab notebook.

Advantages

The findings of current study demonstrated that by applying machine learning methods and using climatic and geospatial features as predictive variables, the occurrence of LSDV infection could be predicted in test set (unseen data) with high accuracy. For instance, ANN algorithm indicated 97% accuracy score. However, the accuracy score is not the preferred performance measure for classifiers, particularly where certain classes are more frequent than others (Géron 2019). As a result, when assessing the predictive power of algorithms, it makes more sense to consider performance metrics such as precision, recall, F1 score, and AUC. Regarding AUC metric and by incorporating all predictive variables in the model or using only meteorological variables as predictors, the highest performance was associated with ANN algorithm (97% in both models) (Table 3). Artificial neural networks have been widely used in different fields including medical and health field, such as medical diagnosis and disease prediction and obtained the very good prediction results (Abbass 2002; Al-Shayea 2011; Baxt 1995; Fang et al. 2014; Flores-Fernández et al. 2012; Kara and Dirgenali 2007; Kia et al. 2013; Ma and Wang 2010; Wang and Gupta 2013; Wang et al. 2001; Zhu and Wang 2010). The reason for better performance of ANN could be attributed to the fact that this algorithm is a universal approximator which can approximate a large class of functions with a high degree of accuracy (Y. Wang et al. 2015)

The predictive performance of ANN was almost the same in both models (using all predictor variables vs only climatic predictive variables) with AUC of 0.97. The literature shows that feature selection can boost the classifier's prediction accuracy, scalability, and generalization capability. This technique is critical in information discovery because it reduces computational complexity, storage, and cost (Gutkin et al. 2009). It should be noted, however, that any predictive feature may be irrelevant individually, but when combined with others, it becomes relevant (Gheyas and Smith 2010). As a result, feature selection does not always imply improved results, and in some cases, eliminating features could be detrimental (Guyon et al. 2008). To the best of the author's knowledge, no other study has used machine learning algorithms to forecast the incidence of LSDV infection using geospatial and meteorological predictive parameters. However, some similar studies utilized machine learning methods to predict the occurrence of some viral livestock diseases based on climatic data. Liang et al. (2020) used machine learning methods to forecast African swine fever outbreaks around the world using bio-climatic variables, and Niu et al. (2020) applied Fig. 4 Receiver operating characteristic (ROC) curves of various machine learning algorithms for model 1 (including all predictors) Fig. 5 Receiver operating characteristic (ROC) curves of various machine learning algorithms for model 2 (including only predictive meteorological variables) 55 Page 8 of 11 Tropical Animal Health and Production (2022) 54: 55 1 3 various machine learning algorithms to forecast Peste des Petits ruminants (PPR) outbreaks based on certain bioclimatic variables and altitude data. Nevertheless, the time frame during which climate data (WordClim database which contains data for 1970–2000) used in these studies was before the time period during which disease outbreaks data utilized and this could be a potential source of bias. In contrast, in the present study, meteorological data were downloaded for the period 2011–2019 from CRU TS4.04 database (Harris et al. 2020) to provide better time concordance with event data of LSDV infection.

According to the feature selection algorithm, out of meteorological, animal density, land cover, and elevation data, only meteorological variables were chosen as significant predictive factors in the present study. Similarly, wet and warm climates which are prime habitat for blood-feeding arthropods have been linked to the occurrence of LSDV infection previously (Alkhamis and VanderWaal 2016; Chihota et al. 2003; Weiss 1968). Some studies which used statistical methods have found a connection between land cover characteristics and/or animal density and disease incidence. For instance, Alkhamis and VanderWaal (2016) examined LSDV outbreak records in the Middle East between 2012 and 2015. The most important environmental predictors that contributed to the ecological niche of LSDV were annual precipitation, land cover, mean diurnal range, type of livestock production system, and global livestock densities, according to ecological niche modeling. Allepuz et al. (2019) investigated the relationship between confirmed LSDV infection outbreaks and climatic factors, land cover, and cattle density in the Balkans, Caucasus, and Middle East between 2012 and 2018. The findings revealed that the likelihood of disease incidence was considerably higher in areas dominated by croplands, grassland, or shrub land. Higher cattle populations, as well as regions with a higher annual mean temperature and a larger diurnal temperature range, increased the odds. In contrast to areas covered mostly by forest, areas with sparse vegetation have a lower risk of infection. Gari et al. (2010) conducted a questionnaire survey to perform a cross-sectional analysis to assess the distribution of LSDV infection and related risk factors in Ethiopia’s three major agro-climatic areas. Across agro-climate zones, herdlevel prevalence of LSDV infection was slightly higher in the midland agro-climate than in the highland and lowland agroclimate zones. The odds ratio of LSDV infection incidence was 3.86 (95% confidence interval: 2.61–5.11) in the midland vs. highland region and 4.85 (95% confidence interval: 2.59–7.1) in the lowland vs. highland zone. The introduction of new animal, as well as communal grazing and watering management, was correlated with a significantly increased risk of LSDV infection incidence. Molla et al. (2017) conducted a research between 2000 and 2015 with the goals of determining the geographical and temporal spread of LSDV infection outbreaks and forecasting the possible outbreaks in Ethiopia. The incidence varied by region, with the lowest in hot dry lowlands and the highest in wet moist highlands. They discovered that outbreaks were seasonal, occurring most often in the months after a long rainy season. All the mentioned researches used statistical methods which are designed for inference about the relationships between variables and not making predictions. On the contrary, prediction made by machine learning algorithms aims at forecasting unobserved outcomes (Bzdok et al. 2018) which is what has been used in the present study. In addition to the different methods used, discrepancies in the results of similar researches could also be caused by the use of different independent variables (risk factors) and different study locations. However, it is worth mentioning that the LSDV outbreak data used in the present study were mainly passive accounts from veterinary facilities in various countries.

Diasadvantages

There are some drawbacks of using passive monitoring data that should be addressed when analyzing the findings. The presence or quality of compensation schemes, the capability and transparency of veterinary facilities, the remoteness of some regions, and farmer visibility all impede reporting in some countries. Nevertheless, the lack of LSDV reports in some areas of the surveyed countries could be attributed to a lack of suitable environmental conditions for the dissemination of the disease in the area. Other limitations of the current study include the small amount of data used, the small number of predictor variables used, and the possibility that the disease has spread to other regions of the studied countries with different climatic and geographical conditions since conducting this research. In conclusion, some machine learning algorithms like ANN could be potentially used to accurately forecast the occurrence of LSDV infection based on some geospatial and meteorological parameters. Using this approach could be extremely beneficial to implement monitoring and awareness schemes, as well as preventive measures such as vaccine campaigns in areas where LSDV infection is a high risk. Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11250-022-03073-2. Author contribution Study conception and design, material preparation, data collection and analysis, and writing the manuscript were all performed by E.A.S.. E.A.S. read and approved the final manuscript. Data availability The datasets generated during and/or analyzed during the current study are available in the Mendeley repository, https://data. mendeley.com/datasets/7pyhbzb2n9/1.

Conclusion

Results Distribution of outbreaks points Between January 2011 and March 2021, 3039 LSDV infection outbreaks were recorded across Africa, Asia, and Europe which indicates the distribution of outbreaks points along with 21,757 free points

In conclusion, some machine learning algorithms like ANN could be potentially used to accurately forecast the occurrence of LSDV infection based on some geospatial and meteorological parameters. Using this approach could be extremely beneficial to implement monitoring and awareness schemes, as well as preventive measures such as vaccine campaigns in areas where LSDV infection is a high risk.

Appendix

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Devin, M. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint* arXiv:1603.04467.

Abbass, H. A. (2002). An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artificial intelligence in Medicine, 25*(3), 265-281.

Alemayehu, G., Zewde, G., & Admassu, B. (2013). Risk assessments of lumpy skin diseases in Borena bull market chain and its implication for livelihoods and international trade. *Tropical Animal Health and Production, 45*(5), 1153–1159. https://doi.org/10.1007/s11250-012-0340-9

Alkhamis, M. A., & VanderWaal, K. (2016). Spatial and temporal epidemiology of lumpy skin disease in the Middle East, 2012–2015. *Frontiers in veterinary science, 3*, 19. https://doi.org/10.3389/fvets.2016.00019

Allepuz, A., Casal, J., & Beltrán-Alcrudo, D. (2019). Spatial analysis of lumpy skin disease in Eurasia—Predicting areas at risk for further spread within the region. *Transboundary and emerging diseases, 66*(2), 813-822. https://doi.org/10.1111/tbed.13090

Al-Shayea, Q. K. (2011). Artificial neural networks in medical diagnosis. *International Journal of Computer Science Issues, 8*(2), 150-154.

Von Backstrom, U. (1945). Ngamiland cattle disease: preliminary report on a new disease, the etiological agent being probably of an infectious nature. *Journal of the South African Veterinary Association, 16*(1), 29–35. https://hdl.handle.net/10520/AJA00382809_377

Baxt, W. G. (1995). Application of artificial neural networks to clinical medicine. *The lancet, 346*(8983), 1135-1138.

Breiman, L. (1996). Bagging predictors. *Machine learning, 24*(2), 123-140.https://doi.org/10.1023/A:1018054314350

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.https://doi.org/10.1023/A:1010933404324

Bzdok, D., Altman, N., & Krzywinski, M. (2018). Points of significance: statistics versus machine learning. In: Nature Publishing Group.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system*. Paper presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.

Chihota, C., Rennie, L., Kitching, R., & Mellor, P. (2003). Attempted mechanical transmission of lumpy skin disease virus by biting insects. *Medical and Veterinary Entomology, 17*(3), 294-300. https://doi.org/10.1046/j.1365-2915.2003.00445.x

Chollet, F. (2018). *Deep learning with Python* (Vol. 361): Manning

skin disease in Ethiopia. *Epidemiology & Infection, 138*(11), 1657-1666. https://doi.org/10.1017/s0950268810000506

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*: O'Reilly Media.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning, 63*(1), 3-42.

Gheyas, I. A., & Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern recognition, 43*(1), 5-13. https://doi.org/10.1016/j.patcog.2009.06.009

Gilbert, M., Nicolas, G., Cinardi, G., Van Boeckel, T. P., Vanwambeke, S. O., Wint, G. W., & Robinson, T. P. (2018). Global distribution data for cattle, buffaloes, horses, sheep, goats, pigs, chickens and ducks in 2010. *Scientific data, 5*(1), 1-11. https://doi.org/10.1038/sdata.2018.227

Golden, C. E., Rothrock Jr, M. J., & Mishra, A. (2019). Comparison between random forest and gradient boosting machine methods for predicting Listeria spp. prevalence in the environment of pastured poultry farms. *Food research international, 122*, 47-55.

Gutkin, M., Shamir, R., & Dror, G. (2009). SlimPLS: a method for feature selection in gene expression-based disease classification. *PloS one, 4*(7), e6416. https://doi.org/10.1371/journal.pone.0006416

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2008). *Feature extraction: foundations and applications* (Vol. 207): Springer.

Harris, I., Osborn, T. J., Jones, P., & Lister, D. (2020). Version 4 of the CRU TS monthly high-resolution gridded multivariate climate dataset. *Scientific data, 7*(1), 1-18. https://doi.org/10.6084/m9.figshare.11980500

Kanevski, M., Pozdnukhov, A., & Timonin, V. (2008). Machine learning algorithms for geospatial data. Applications and software tools.

Kara, S., & Dirgenali, F. (2007). A system to diagnose atherosclerosis via wavelet transforms, principal component analysis and artificial neural networks. *Expert Systems with Applications, 32*(2), 632-640.

Kia, S., Setayeshi, S., Shamsaei, M., & Kia, M. (2013). Computer-aided diagnosis (CAD) of the skin disease based on an intelligent classification of sonogram using neural network. *Neural Computing and Applications, 22*(6), 1049-1062.

Kovacs-Györi, A., Ristea, A., Havas, C., Mehaffy, M., Hochmair, H. H., Resch, B., . . . Blaschke, T. (2020). Opportunities and Challenges of Geospatial Analysis for Promoting Urban Livability in the Era of Big Data and Machine Learning. *ISPRS International Journal of Geo-Information, 9*(12), 752.https://doi.org/10.3390/

Model performance test

Date	9 november 2023
Team ID	PNT2022TMID611670
Project name	
Maximum marks	10 marks

Model performance Testing :

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	Lumpy skin disease (LSD) is a highly contagious viral disease that affects cattle, buffaloes, and other bovines. It is caused by the lumpy skin disease virus (LSDV), which is a member of the poxvirus family. LSD is characterized by the formation of nodules on the skin of infected animals, as well as fever, lymphadenopathy, and reduced milk production. Machine learning (ML) and artificial intelligence (AI) can be used to predict the occurrence of LSD outbreaks. This can be done by developing ML models that are trained on historical data of LSD outbreaks, such as meteorological data, geospatial data, and livestock data. Once trained, the ML model

			can be used to predict the risk of LSD outbreaks in a given area. Model inputs: Meteorological data: temperature, humidity, rainfall, wind speed, etc. Geospatial data: elevation, land cover, proximity to water bodies, etc. Livestock data: population of cattle, buffaloes, and other bovines in the area. Model output: Risk of LSD outbreak: low, medium, or high. The model can be trained on a dataset of historical LSD outbreaks, which would include the model inputs and outputs. Once trained, the model can be used to predict the risk of LSD outbreaks in new areas.
2.	Accuracy	Training Accuracy - Validation Accuracy -	100 97
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	 80