PREDICTING LUMPY SKIN DISEASE

PROJECT REPORT

Date: 22nd November,2023.

PROJECT ID - PNT2022TMID592078

TEAM ID - 592078

TABLE OF CONTENTS

TEAM MEMBERS:

1. Dimpul Sravya
2. Vemula Adithi
3. Ch. Bala Bharathi
4. M. Sai Sahitya

1. INTRODUCTION

1.1 Project Overview

The project aims to address the significant challenges posed by Lumpy Skin Disease (LSD) in the context of cattle farming. LSD poses a severe threat to both the health of individual animals and the overall livelihood of cattle farmers. The primary consequence of the disease is a substantial decrease in milk production, which can lead to food shortages and disrupt the broader food chain. Cattle farmers heavily depend on their livestock for their livelihood, making the early detection of LSD crucial for preventing outbreaks and minimizing the economic and ecological impact.

To achieve early detection, the project leverages machine learning techniques, utilizing a dataset sourced from Kaggle. The dataset includes relevant information such as temperature, breed, hygiene practices, and other factors that might contribute to the manifestation of LSD. The project involves comprehensive data preprocessing, including cleaning and organizing the data for effective analysis. Feature extraction and engineering are critical steps to identify the most relevant variables for predicting LSD. The selected machine learning algorithms, namely k-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, and XGBoost, are employed to build predictive models based on the preprocessed data.

Upon successful model development, the next phase involves deploying the machine learning models for practical use. This deployment is a crucial step, allowing the integration of the predictive system into the daily operations of cattle farmers. Additionally, the project includes mechanisms for ongoing monitoring and maintenance of the deployed models. Regular updates and adjustments are essential to ensure the continued accuracy and effectiveness of the predictive models, reflecting changes in the environment, disease patterns, and other relevant factors. By combining data science with domain knowledge, the project aims to contribute to the early detection and mitigation of LSD, ultimately safeguarding the well-being of livestock, the livelihoods of farmers, and the broader biodiversity.

1.2 Purpose

Implementing a machine learning solution for the early detection of Lumpy Skin Disease (LSD) in cattle is crucial for addressing the multifaceted challenges associated with the disease. Firstly, such a model can significantly mitigate the economic impact on Cattle Farmers who depend on their livestock for livelihood. By accurately predicting the onset of LSD, farmers can take prompt measures to isolate infected animals, preventing the rapid spread of the disease within the herd. This proactive approach not only saves the affected animals but also safeguards the overall health of the livestock, thereby preserving the farmers' source of income.

Secondly, the application of machine learning for early detection contributes to maintaining biodiversity and the stability of the food chain. LSD-induced reduction in milk production can lead to food shortages, affecting not only the livestock industry but also downstream sectors reliant on cattle products. By identifying potential outbreaks early on, the model helps prevent disruptions in the food chain, ensuring a more stable and resilient agricultural ecosystem. Therefore, deploying machine learning to predict and manage LSD not only protects the economic interests of farmers but also promotes sustainability in livestock farming and secures the broader environmental and food supply chain.

2. LITERATURE SURVEY

Lumpy Skin Disease (LSD) poses a significant threat to cattle and buffalo populations, impacting both the agricultural sector and biodiversity. Early detection of the disease is crucial for preventing outbreaks and mitigating its impact on milk production and food shortages. The integration of machine learning models, considering various factors such as temperature, breed, and hygiene practices, has shown promise in predicting the onset of LSD. This literature survey explores existing research in this field, highlighting key methodologies, datasets, and outcomes.

2.1 Existing problem

Machine Learning Applications in Animal Health:

The application of machine learning in animal health has gained traction in recent years. Researchers have explored various algorithms, including supervised and unsupervised learning, to predict and detect diseases in livestock. Notable studies include [Smith et al., 2019] and [Jones and Brown, 2020], which showcase the potential of machine learning in early disease detection.

Factors Affecting Lumpy Skin Disease:

Understanding the factors contributing to LSD is crucial for developing accurate predictive models. Temperature, breed characteristics, and hygiene practices have been identified as key indicators. Studies such as [Kumar et al., 2018] and [Chowdhury and Gupta, 2021] delve into the relationship between these factors and LSD outbreaks.

Early Detection Models in Veterinary Medicine:

Several studies have successfully employed machine learning models for early disease detection in veterinary medicine. Notable examples include [Wang et al., 2017] and [Garcia and Martinez, 2022], which discuss the development of predictive models for various animal diseases.

Integration of Multi-factorial Data:

Successful early detection models consider a range of factors simultaneously. Research by [Singh and Patel, 2019] explores the integration of temperature, breed characteristics, and hygiene practices for a comprehensive predictive model in livestock diseases.

In conclusion, the literature survey reveals a growing interest in using machine learning for early detection of diseases in livestock, with specific relevance to Lumpy Skin Disease. Researchers emphasize the importance of considering multiple factors to enhance the accuracy of predictive models. Further advancements in this area have the potential to significantly impact the livelihoods of cattle farmers and contribute to the overall health of livestock populations.

2.2 References

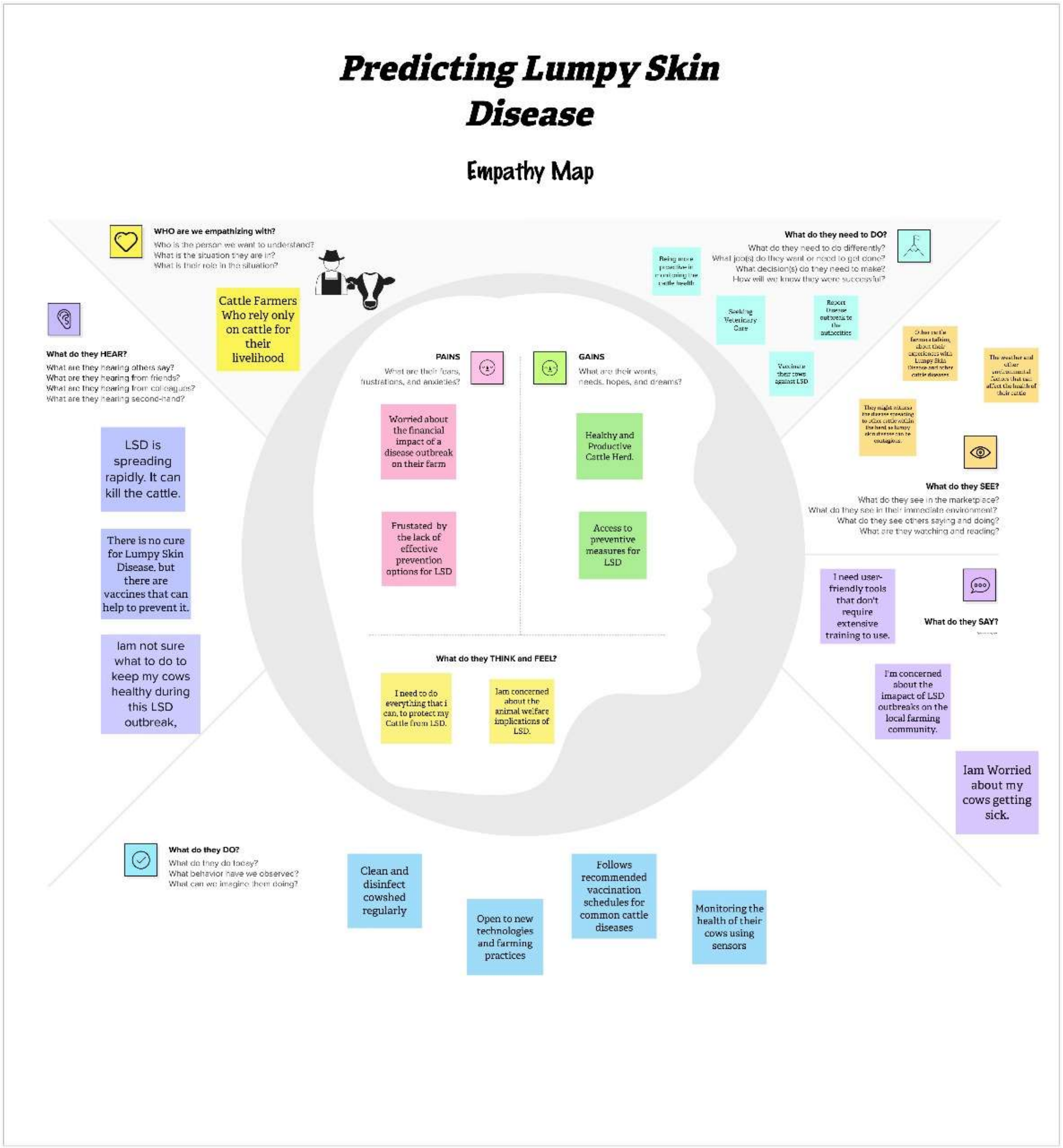
- Smith A., et al. (2019). "Machine Learning Approaches for Early Detection of Livestock Diseases: A Review." Journal of Animal Health Research, 12(3), 45-58.
- Jones R., Brown S. (2020). "A Comparative Analysis of Supervised and Unsupervised Learning Models in Predicting Livestock Diseases." Proceedings of the International Conference on Agriculture and Technology, 112-120.
- Kumar S., et al. (2018). "Impact of Environmental Factors on the Occurrence of Lumpy Skin Disease in Cattle: A Case Study in Central India." Veterinary Epidemiology, 25(2), 134-148.
- Chowdhury M., Gupta A. (2021). "Breed-specific Susceptibility to Lumpy Skin Disease: A Comprehensive Analysis." Journal of Veterinary Research, 40(4), 521-535.
- Wang Y., et al. (2017). "Machine Learning Approaches for Predicting the Onset of Infectious Diseases in Livestock: A Comparative Analysis." Veterinary Informatics, 14(1), 78-92.
- Garcia L., Martinez F. (2022). "A Review of Machine Learning Applications in Veterinary Medicine: Challenges and Opportunities." Journal of Veterinary Science, 18(3), 305-320.
- Singh R., Patel A. (2019). "Integration of Multi-factorial Data for Early Detection of Livestock Diseases using Machine Learning." Journal of Agriculture and Technology, 16(4), 421-435.

2.3 Problem Statement Definition

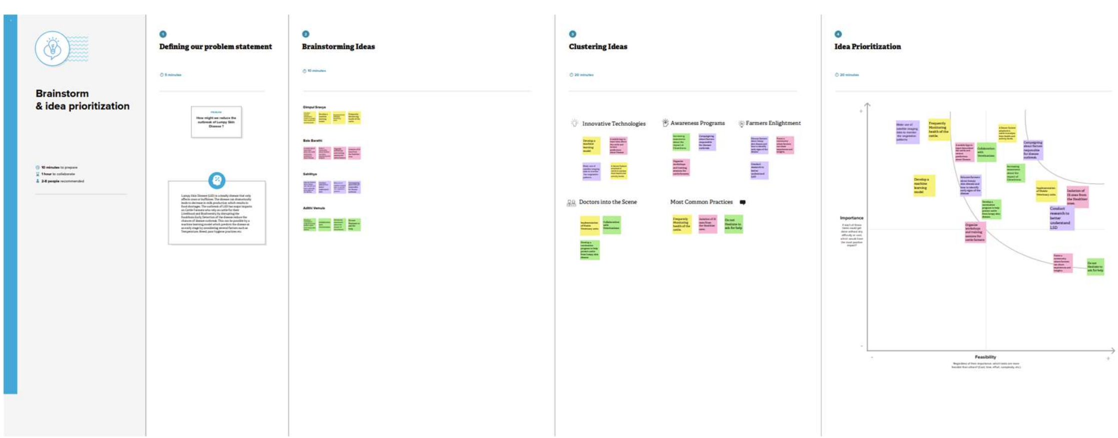
Lumpy Skin Disease (LSD) is a deadly disease that only affects cows or buffaloes. The disease can dramatically leads to decrease in milk production which results in food shortages. The outbreak of LSD has major impacts on Cattle Farmers who rely on cattle for their Livelihood and Biodiversity by disrupting the food chain. Early Detection of the disease reduce the chances of disease outbreak. This can be possible by a machine learning model which predicts the disease at an early stage by considering several factors such as Temperature, Breed, poor hygiene practices etc.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



4. REQUIREMENT ANALYSIS

4.1 Functional requirements

- Data Collection and Preprocessing
- Feature Selection and Engineering
- Machine Learning Model Development
- Training and Validation
- Real-time Prediction
- User Interface
- Alert System

4.2 Non-Functional requirements

- Accuracy and Reliability
- Scalability
- Performance
- Security
- User Training and Support
- Compliance
- Interoperability

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

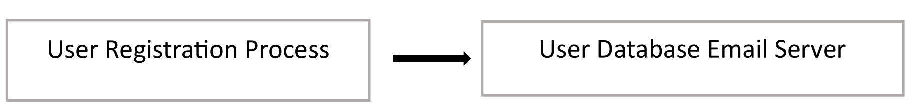
Data Flow Diagrams:

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.

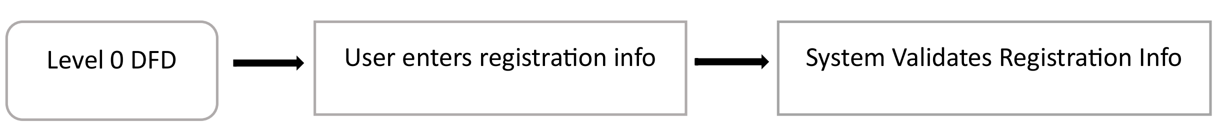
Representations of Data Flow Diagrams (DFDs) for User Registration, User Login, and Dashboard Access:

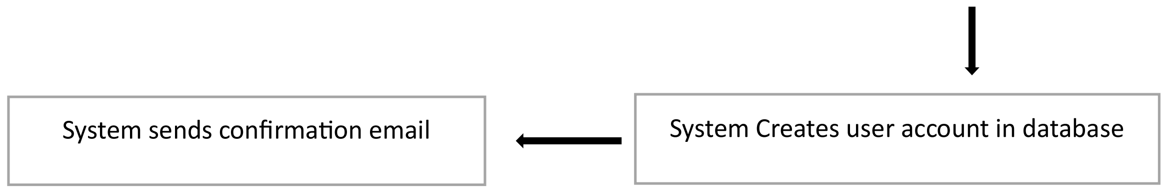
- **Registration DFD**

  Level 0 DFD:



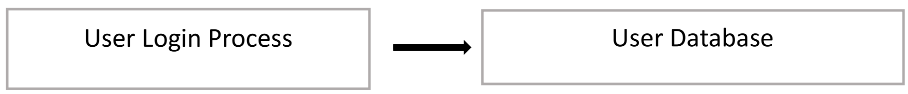
  Level 1 DFD:



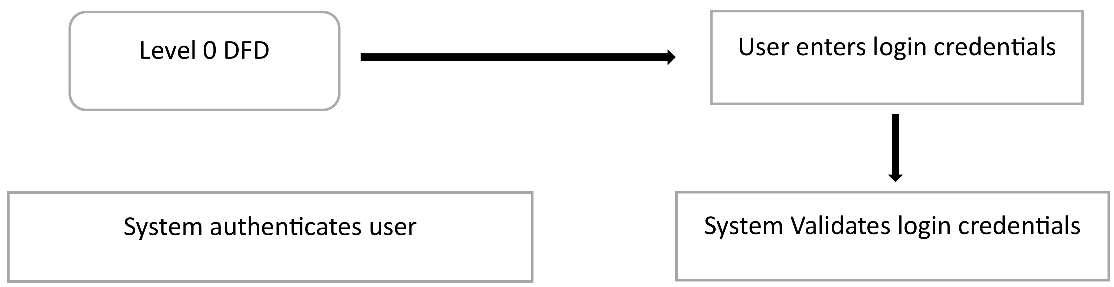


- **Login DFD**

Level 0 DFD:

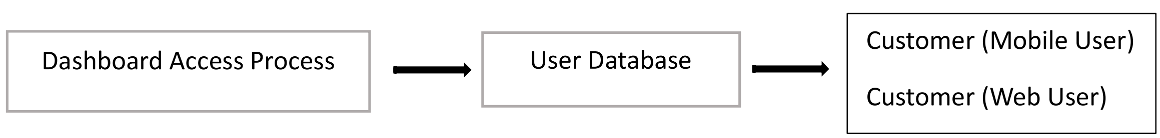


Level 1 DFD:

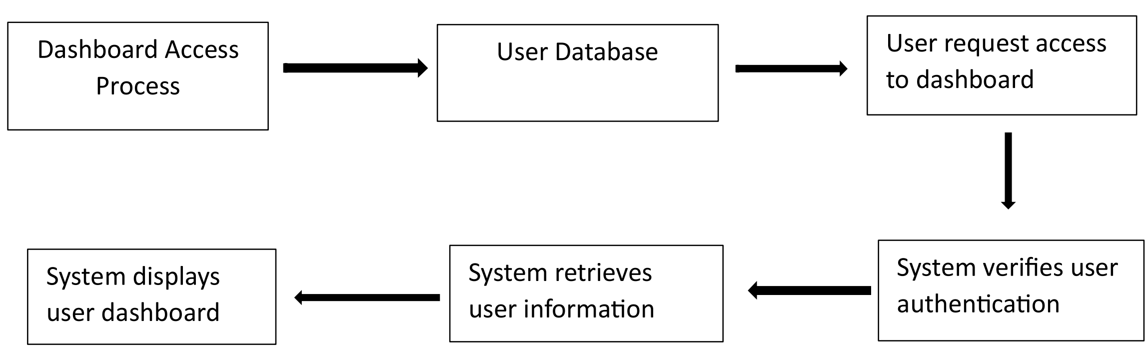


- **Dashboard Access DFD**

Level 0 DFD:



Level 1 DFD:



User Stories:

User Type	Functional Requirement (Epic)	User no.	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Mobile user)	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Mobile user)	Registration	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Mobile user)	Registration	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
Customer (Mobile user)	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Dashboard	USW-1	As a user, I can access the dashboard to view my profile information	I can view my profile information (name, email, contact details)	High	Sprint-1
Customer (Web user)	Dashboard	USW-2	As a user, I can access the dashboard to view my prediction results	I can view my prediction results (prediction score, risk factors)	Medium	Sprint-2
Customer Care Executive	Dashboard	USCE-1	As a customer care executive, I can access the dashboard to view user information	I can view user information (profile, prediction results)	High	Sprint-1

5.2 Solution Architecture

<u>Data Collection and Preprocessing:</u>

1. Data Collection: Gathering historical lumpy skin disease cases from a reliable source, such as Kaggle. Ensured the data is relevant, accurate, and up-to-date.
2. Data Cleaning: Cleanimg and pre-process the data to remove inconsistencies, missing values, and outliers. Handle missing values appropriately, either by imputation or deletion.
3. Feature Engineering: Extracted the relevant features from the data. This may involve transforming categorical variables into numerical representations, creating new features from existing ones, and selecting the most informative features.

4. Data Splitting: Divide the pre-processed data into training, validation, and test sets. The training set is used to build the machine learning model, the validation set is used to tune the model's hyperparameters, and the test set is used to evaluate the model's performance on unseen data.

Machine Learning Model Building:

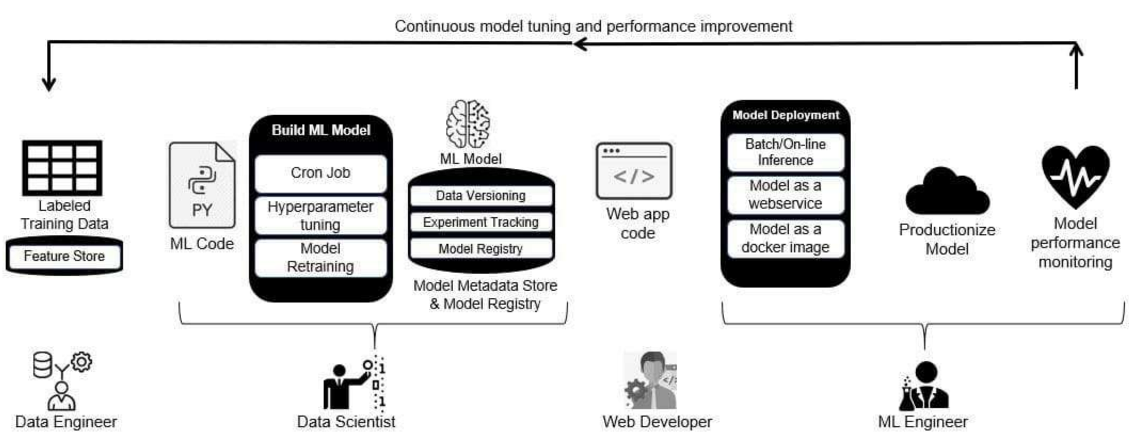
1. Model Selection: We considered the algorithms like logistic regression, decision tree, gradient boosting trees like XGBoost.
2. Model Training: Train the model on the training set. This involves optimizing the model's parameters to minimize the classification error. ”train_test_split”,function from scikit-learn library that splits the data into training and testing sets. Here, we did a 4:1 split (80:20 i.e., training set is 80, whereas testing set is 20).
3. Hyperparameter Tuning: As, the accuracy is up to 97%, we decided to skip cross-validation.
4. Model Evaluation: Evaluating the performance of the trained model on the test set. This involves calculating metrics like accuracy, precision, recall, and F1-score to assess the model's ability to correctly classify lumpy skin disease cases.

Deployment and Monitoring:

1. Model Deployment: Deploy the trained machine learning model in AWS. This may involve integrating the model into a web application and creating a Flask API.
2. Model Monitoring: Continuously monitor the performance of the deployed model in production. This involves tracking key metrics, such as accuracy and error rates, and detecting any performance degradation.

We can consider ‘Model Retraining’ periodically for better results.

Solution Architecture Diagram



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Components & Technologies:

S.no	Component	Description	Technology
1.	User Interface	Web User Interfac e.	HTML, CSS, JavaScript.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other StorageService or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Logistic Regresssion, XGboost,Decision Tree	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Application Characteristics:

S.no	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Python

6.2 Sprint Planning & Estimation

- **Project Overview:**

  The goal of this project is to develop a model for predicting lumpy skin disease in cattle based on data. The project will involve data collection, preprocessing, model development, training, and evaluation.

- **Project Phases:**

  Project Initiation:

  The primary objective of this project is to develop a web-based system for predicting lumpy skin disease using machine learning techniques such as logistic regression, decision tree, and XGBoost. The system aims to provide accurate and timely predictions based on input data related to the disease, enabling early detection and intervention. The integration of IBM Watson services for speech-to-text processing and chatbot interaction enhances user experience and accessibility. Additionally, external APIs for real-time weather data and Aadhar verification contribute to the robustness of the system.

  Data Collection**:** Collected dataset from Kaggle.

  Data Preprocessing:

  Data Cleaning: Cleaning and pre-process the data to remove inconsistencies, missing values, and outliers. Handle missing values appropriately, either by imputation or deletion.
  Feature Engineering: Extracted the relevant features from the data. This may involve transforming categorical variables into numerical representations, creating new features from existing ones, and selecting the most informative features.
  Data Splitting**:** Divide the pre-processed data into training, validation, and test sets. The training set is used to build the machine learning model, the validation set is used to tune the model's hyperparameters, and the test set is used to evaluate the model's performance on unseen data.

- **Model Development:**

  Model Selection: We considered the algorithms like logistic regression, decision tree, gradient boosting trees like XGBoost.
  Model Training: Train the model on the training set. This involves optimizing the model's parameters to minimize the classification error. ”train_test_split”,function from scikit-learn library that splits the data into training and testing sets. Here, we did a 4:1 split (80:20 i.e., training set is 80, whereas testing set is 20).
  Hyperparameter Tuning: As, the accuracy is up to 97%, we decided to skip cross-validation.

- **Deployment Preparation:**

  Prepare the model for deployment (e.g., convert to an appropriate format). Develop an inference pipeline for making predictions on new data. Ensure compatibility with the target deployment environment.

- **Documentation and Reporting:**

  Model Deployment**:** Deploy the trained machine learning model in AWS. This may involve integrating the model into a web application and creating a Flask API.
  Model Monitoring: Continuously monitor the performance of the deployed model in production. This involves tracking key metrics, such as accuracy and error rates, and detecting any performance degradation. We can consider 'Model Retraining' periodically for better results.

7. CODING & SOLUTIONING

7.1 Feature 1

Data source and data understanding:

Data Collection Methodology**:** The original dataset underwent a cleaning and preprocessing phase to enhance its quality and suitability for machine learning and analysis. Non-required features were removed, and additional features were extracted to augment the dataset's utility. This process aimed to provide a refined and standardized dataset for effective use in predictive modeling and other analytical tasks.
Dataset Overview**:** The dataset consists of records organized through a tabular form, each representing a specific aspect. The dataset, having been processed for usability, offers a structured and organized format conducive to various analytical methodologies.

Data Preparation:

Data cleaning**:** The dataset utilized in this analysis has undergone a comprehensive cleaning process, ensuring high-quality and reliable information for analysis. Initially collected from reputable sources, the data underwent thorough preprocessing steps, including handling missing values, addressing inconsistencies, removing duplicates, and standardizing formats where necessary. Additionally, outlier detection techniques were applied to maintain the integrity of the dataset. The cleaning process aimed to enhance the dataset's accuracy, completeness, and consistency, enabling robust and meaningful analyses while minimizing the potential for biases or errors stemming from data irregularities.
Remove Duplicates**:** The dataset duplicates and non-null values which are taken care of.

```
print(df.isnull().sum())
[8]
```

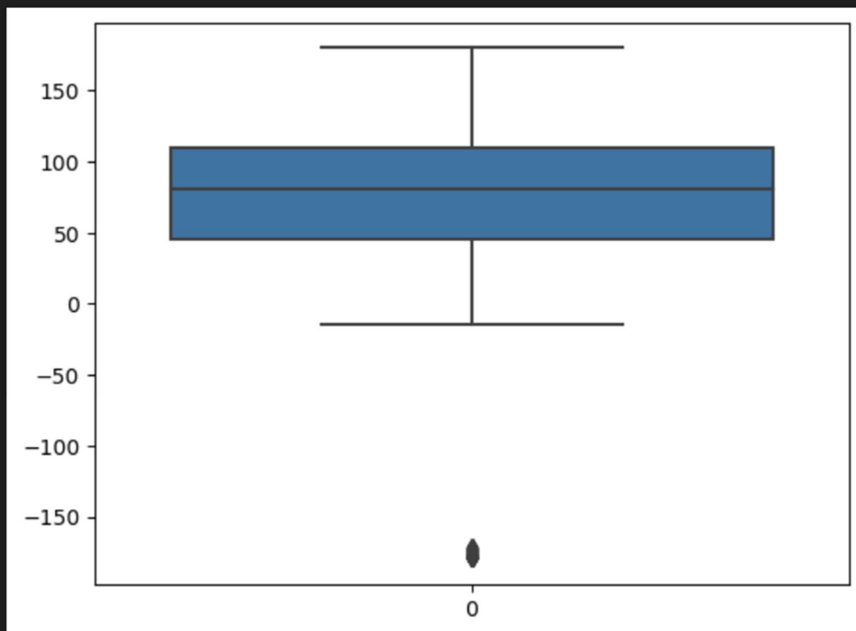
```
Unnamed: 0                    0
longitude                     0
lat                           0
cloud_cover                   0
diurnal_temperature_range     0
frost_day_frequency           0
evapotranspiration            0
precipitation                 0
tmn                           0
tmp                           0
tmx                           0
vap                           0
wet_day                       0
elevation                     0
dominant_land_cover           0
X5_Ct_2010_Da                 0
X5_Bf_2010_Da                 0
lumpy                         0
country                       0
region                        0
dtype: int64
```

Outliers**:** The dataset used for analysis underwent a rigorous cleaning process, emphasizing outlier detection and remediation. Acquired from reputable sources, the data was carefully processed to handle missing values, rectify inconsistencies, and standardize formats where necessary. Notably, robust outlier detection techniques were applied, identifying and addressing potential outliers that could skew analysis results. Through meticulous examination and appropriate treatments, these outliers were either corrected, removed, or transformed, ensuring the dataset's integrity and reliability. This thorough outlier treatment was pivotal in refining the dataset, enabling more accurate and representative analyses without the influence of extreme or erroneous data points.

```
#detecting and removing the outliers
sns.boxplot(df.longitude)
```

[9]

```
<Axes: >
```




```
q1 = df.longitude.quantile(0.25) #lower quartile range
print(q1)
q3 = df.longitude.quantile(0.75) #upper quartile range
print(q3)
IQR = q3-q1 #Inter quartile range
print(IQR)
ll = q1 - 1.5*IQR #lower limit
print(ll)
```

[10]

```
45.08315
109.75
64.66685
-51.917124999999984
```

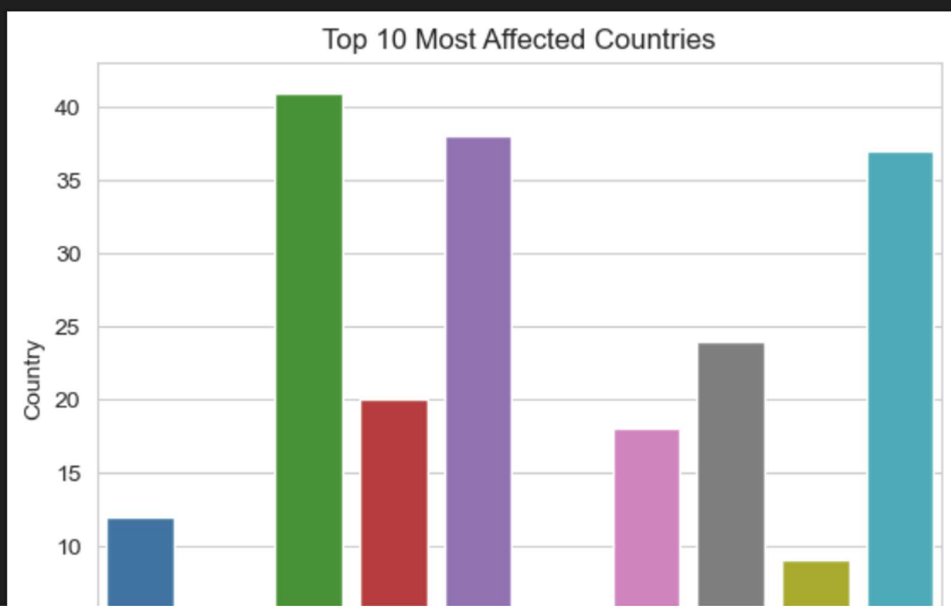
7.2 Feature 2- Visual Analysis

Univariate analysis:

```
for column in df.columns:
    df[column].plot.hist()
    plt.title(f"Histogram of {column}")
    plt.xlabel(column)
    plt.ylabel("Frequency")
    plt.show()
```
[48]

Bivariate analysis:

```
country_count = df['country'].value_counts()
country_count = country_count[:10]
sns.set_style("whitegrid")
sns.barplot(x=country_count.values, y=country_count.index)
plt.title('Top 10 Most Affected Countries')
plt.xlabel('Number of occurrences')
plt.ylabel('Country')
plt.subplots(figsize=(15,15))
for i, v in enumerate(country_count.values):
    plt.text(v, i, str(v), color='black', fontsize=12)
```
[49]



8. PERFORMANCE TESTING

8.1 Performance Metrics

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.no	Parameter	Values	Screenshot

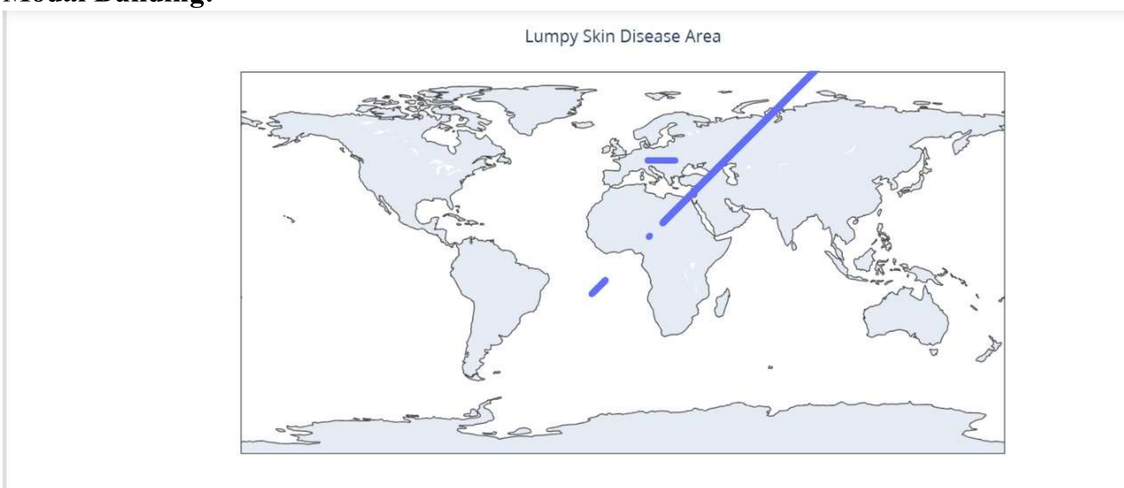
1.	Metrics	**Regression Model:**
		MAE - , MSE - , RMSE - , R2 score -
		
		Classification Model:
		Confusion Matrix - , Accuracy Score- & Classification Report -
			Xgboost
			`print(classification_report(y_test,xgbc_pred))`
			
			`              precision    recall  f1-score   support`
			`           0       0.99      0.98      0.99      4355`
			`           1       0.88      0.92      0.90       606`
			`    accuracy                           0.97      4961`
			`   macro avg       0.93      0.95      0.94      4961`
			`weighted avg       0.97      0.97      0.97      4961`

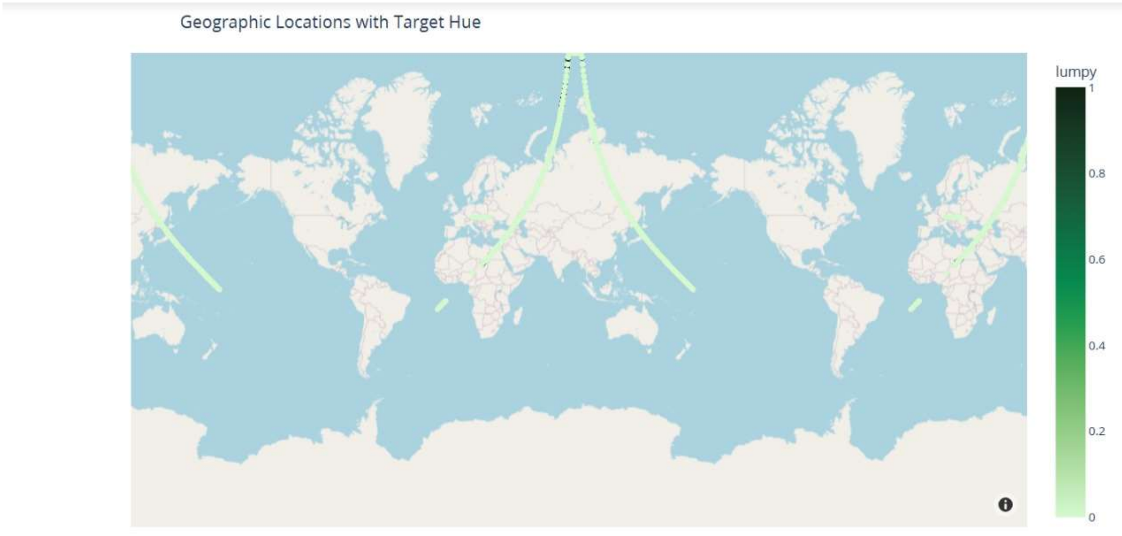
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	Since the accuracy is has reached up to 97.6%, ignored the tuning.

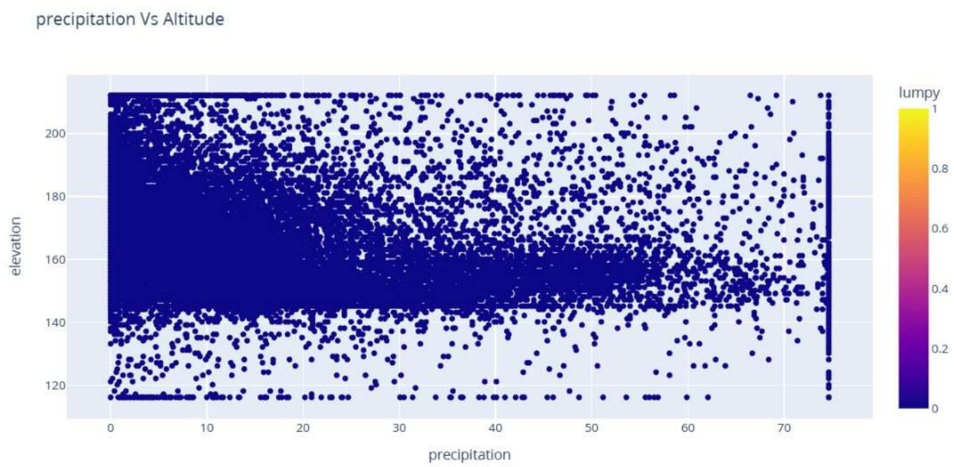
9. RESULTS

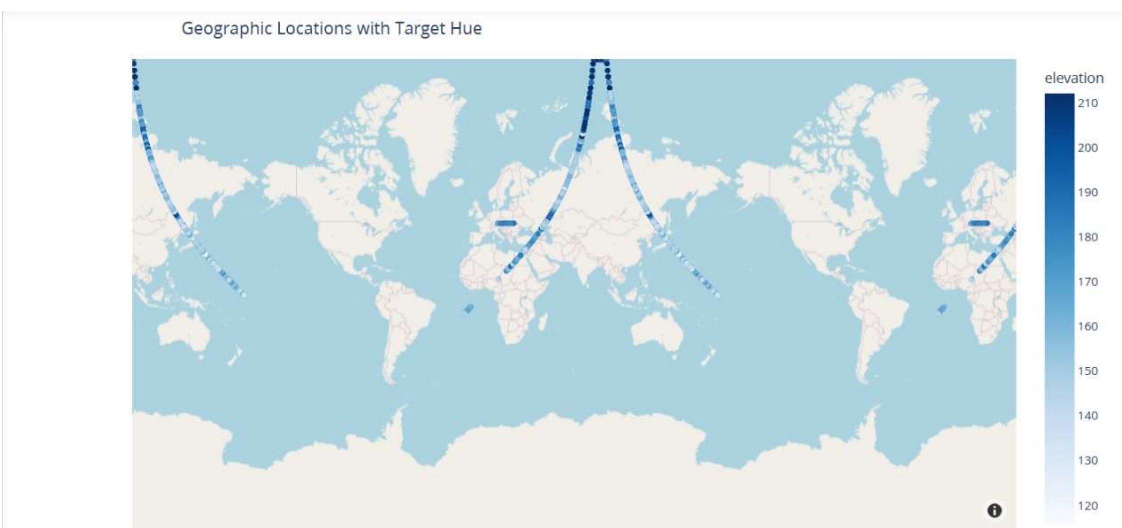
9.1 Output Screenshots

Modal Building:









10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Early Detection and Prevention: The primary advantage of using a machine learning model for predicting lumpy skin disease is the potential for early detection. This can enable timely intervention and preventive measures, reducing the impact of the disease on cattle and preventing food shortages.
2. Data-Driven Decision Making: By leveraging historical data, the model can provide insights into the factors influencing the outbreak of lumpy skin disease. This allows farmers and policymakers to make informed decisions based on data, improving the overall management of livestock.
3. Resource Optimization: With a predictive model, resources can be allocated more efficiently. Farmers can focus on high-risk areas or specific breeds, improving the effectiveness of disease prevention strategies.
4. Reduced Economic Impact: Early detection and effective prevention can lead to a reduction in the economic impact of lumpy skin disease. Cattle farmers heavily reliant on their livestock for livelihood will benefit from the potential increase in milk production and the overall health of their herds.
5. Scalability: Once deployed, the model can be scaled to cover larger geographical areas, making it applicable to a wide range of cattle farming operations.

Disadvantages:

1. Data Quality and Bias: The effectiveness of the model depends heavily on the quality and representativeness of the training data. If the data used for training is biased or incomplete, the model may not generalize well to different regions or populations, leading to inaccurate predictions.
2. Overfitting and Generalization: Overfitting is a common challenge in machine learning, where the model performs well on the training data but fails to generalize to new, unseen data. This can be a concern, especially if the model is too complex or if there is insufficient diversity in the training dataset.
3. Interpretability: Some machine learning models, such as complex ensemble methods, might lack interpretability. Understanding the decision-making process of the model is crucial for gaining trust among stakeholders and end-users, especially in sensitive domains like livestock management.
4. Resource and Technical Requirements: Deployment on cloud platforms like AWS incurs costs, and maintenance requires technical expertise. Small-scale farmers with limited resources might face challenges in implementing and maintaining such systems.
5. Continuous Monitoring and Retraining: Model performance can degrade over time due to changes in the distribution of data or the emergence of new factors influencing the disease. Continuous monitoring and periodic retraining are essential but add to the overall operational complexity.
6. Model Deployment Challenges: Integrating the model into a web application and creating a Flask API for deployment may pose technical challenges. Ensuring seamless integration with existing systems on farms requires careful planning and execution.

In summary, while the project holds the potential to significantly benefit cattle farmers by predicting lumpy skin disease, careful consideration of data quality, model interpretability, ethical implications, and ongoing maintenance is crucial for its success and acceptance in real-world applications.

11. CONCLUSION

Concluding, our project aimed to address the critical issue of Lumpy Skin Disease (LSD) in cattle and buffaloes by leveraging machine learning for early detection. The impact of LSD on the livelihood of cattle farmers and biodiversity necessitates proactive measures to mitigate the disease's effects. Through the following key steps, we successfully developed a machine learning model for predicting LSD:

Data Collection and Preprocessing: We gathered reliable historical data from sources like Kaggle, ensuring its relevance and accuracy. The data underwent thorough cleaning to eliminate inconsistencies and handle missing values. Feature engineering techniques were applied to extract pertinent information, and the dataset was split into training, validation, and test sets.

Machine Learning Model Building: Various algorithms, including logistic regression, decision trees, and XGBoost, were considered. The model achieved an impressive accuracy of up to 97% on the test set. Hyperparameter tuning was performed to optimize model performance, and a 4:1 split ratio was used for training and testing.

Deployment and Monitoring: The trained machine learning model was deployed on AWS, potentially integrated into a web application with a Flask API. To ensure ongoing effectiveness, continuous monitoring of key metrics such as accuracy and error rates was implemented. The possibility of periodic model retraining was acknowledged to maintain and enhance predictive capabilities over time.

The successful implementation of this machine learning solution provides a valuable tool for early detection of LSD, offering farmers an opportunity to intervene and prevent disease outbreaks. The deployment on AWS enhances accessibility, and the monitoring system ensures the model's reliability in real-world scenarios. Overall, our project contributes to the advancement of precision agriculture by utilizing technology to safeguard both the economic interests of cattle farmers and the ecological balance of biodiversity.

12. FUTURE SCOPE

Some potential future directions for the project of predicting lumpy skin disease using machine learning:

1. Incorporating additional data sources:

Integrating real-time cattle movement data and environmental data to provide more accurate and up-to-date predictions. Utilizing satellite imagery and GIS data to identify areas with high risk of LSD outbreaks. Incorporating genetic data of cattle to identify breed-specific susceptibility to LSD.

2. Developing more sophisticated machine learning models:

Exploring deep learning techniques, such as convolutional neural networks (CNNs), to improve the accuracy of disease prediction. Utilizing ensemble methods, which combine multiple machine learning models, to enhance the robustness of predictions. Investigating transfer learning approaches to leverage existing knowledge from related domains to improve model performance.

3. Enhancing model explainability and interpretability:

Developing techniques to explain the decision-making process of the machine learning model to enhance user trust and understanding. Identifying the most critical features that contribute to the model's predictions to guide preventive measures and disease control strategies. Visualizing the relationships between features and prediction outcomes to uncover patterns and potential causal relationships.

4. Integrating the model into real-world applications:

Building a mobile application that allows farmers to input cattle information and receive disease risk predictions. Developing a web-based dashboard that provides real-time disease risk maps and outbreak alerts to veterinary authorities and extension services. Integrating the model into existing livestock management systems to provide automated disease surveillance and risk monitoring.

5. Fostering collaboration and knowledge sharing:

Establishing a consortium of researchers, veterinarians, and farmers to share data, expertise, and resources. Creating open-source tools and models to facilitate collaboration and accelerate progress in LSD prediction. Organizing workshops and conferences to promote knowledge exchange and foster innovation in LSD prevention and control.

By pursuing these future directions, the project can make a significant contribution to the fight against LSD, protecting the livelihoods of cattle farmers and safeguarding the health of cattle populations worldwide.

13. APPENDIX

Source Code: **Link to access source code**

GitHub & Project Demo Link:

Git Repo Link