Predicting Lumpy Skin Disease Using Machine Learning

Final Year Project Report

Presented

By

SADAQAT ALI

CIIT/SP21-BSM-013/ISB

In Partial Fulfillment
Of the Requirement for the Degree of
Bachelor of Science in Mathematics

DEPARTMENT OF MATHEMATICS

COMSATS UNIVERSITY ISLAMABAD

DECEMBER 2024

Predicting Lumpy Skin Disease Using Machine Learning

Final Year Project Report

By

SADAQAT ALI

CIIT/SP21-BSM-013/ISB

In Partial Fulfillment

Of the Requirement for the Degree of
Bachelor of Science in Mathematics

DEPARTMENT OF MATHEMATICS

COMSATS UNIVERSITY ISLAMABAD

DECEMBER 2024

Declaration

I, Sadaqat Ali, hereby declare that the work presented in this report is my original work and has been carried out under the guidance of my supervisors, Dr. Amna Nazeer and Mudassir. This report has not been submitted for the award of any other degree or diploma at this or any other university or institute, If found I Shall Stand Responsible.

Signature: Sadaqat Ali

SADAQAT ALI

JANUARY 2025

COMSATS UNIVERSITY ISLAMABAD

Predicting Lumpy Skin Disease using Machine Learning

An Undergraduate Final Year Project Report submitted to the

Department of MATHEMATICS

As a Partial Fulfillment for the award of Degree
Bachelor of Science in Mathematics

By

Name	**Registration Number**
Sadaqat Ali	CIIT/SP21-BSM-013/ISB

Supervised by

Dr. Amna Nazeer

Tenured Associate Professor

Department Of Mathematics

COMSATS UNIVERSITY ISLAMABAD

DECEMBER 2024

Final Approval

This Project Titled

Predicting Lumpy Skin Disease using Machine Learning

Submitted for the Degree of
Bachelor of Science in Mathematics

By

Name	Registration Number
Sadaqat Ali	CIIT/SP21-BSM-013/ISB

has been approved for

COMSATS UNIVERSITY ISLAMABAD

Supervisor [illegible]

Dr. Amna Nazeer

Tenured Associate Professor

*External Examiner*________________

Head of Department [illegible]

Prof. Dr. Shams-ul-Islam

Dedication

This work is dedicated to my beloved parents, teachers, and all those who have supported and inspired me throughout my academic journey. Lastly, To Me for never giving up.

Acknowledgement

I bow my head with the deepest gratitude to Almighty Allah, Who enabled me to complete this piece of work. He is the most powerful, compassionate, kind and merciful. I offer my humblest words of thanks to the Holy Prophet Muhammad (peace be upon him) who is forever a torch of guidance for humanity and Imam-e-Asar (a.s)

I am highly thankful to my supervisor ***Dr. Amna Nazeer*** *for her co-operative and encouraging behavior. Without her kindness and humbleness, it would not have been possible for me to accomplish this project. I am grateful for her constant support and help. She was always there when I needed her. It has been a privilege and great honor for me to work under her supervision.*

I am also deeply grateful to ***Sir Mudassir*** *for his invaluable assistance and support throughout this project. His contributions were essential to its successful completion.*

I am highly indebted to all my teachers who taught me in these four years. Their encouragement and support brought me to this point that I am just to complete my BS through this thesis.

Most importantly, I acknowledge the prayers and concern of my family throughout my life.

Sadaqat Ali

Table of Contents

This Page is Intentionally Left Blank.

Chapter 1

Introduction

1.1 Background of Lumpy Skin Disease

Lumpy Skin Disease (LSD) is a contagious viral disease affecting cattle, characterized by nodules on the skin, fever, and enlarged lymph nodes. The causative agent is the Lumpy Skin Disease Virus (LSDV), a member of the ***Capripoxvirus*** genus within the ***Poxviridae*** family. LSD primarily impacts cattle and, to a lesser extent, water buffalo, leading to significant economic losses due to decreased milk production, weight loss, infertility, and damaged hides.

1.1.1 Epidemiology and Transmission

Initially confined to Africa, LSD has expanded its geographical range over the past few decades. Since 2013, the disease has spread rapidly and widely throughout Russia and Asia, demonstrating how underestimated and neglected this disease is [1].

Transmission occurs mainly through blood-feeding insects such as certain species of flies and mosquitoes, making vector control a crucial component in managing the disease [2].

1.1.2 Clinical Signs and Diagnosis

Affected cattle exhibit a high fever and extensive nodules on the mucosa or the skin, seriously influencing the cattle industry development and international import and export trade [1].

The disease can produce a chronic debility in infected cattle comparable to that caused by foot-and-mouth disease (FMD) [3]. Laboratory confirmation is essential for accurate diagnosis, typically involving polymerase chain reaction (PCR) tests to detect viral DNA.

1.1.3 Economic Impact

LSD leads to significant economic losses in the cattle industry, including decreased milk production, weight loss, infertility, and damaged hides. Mortality rates as high as 40 percent or more have been encountered but are usually lower [3]. The disease also affects international trade due to movement restrictions and loss of market access.

1.2 Significance of Machine Learning in Healthcare

Machine learning (ML), a subset of artificial intelligence, has emerged as a transformative tool in healthcare, offering the potential to revolutionize diagnostics, treatment planning, and patient care.

1.2.1 Applications in Diagnostics and Treatment

ML algorithms can analyze complex datasets to identify patterns that may not be discernible through traditional methods. In diagnostics, ML has been applied to interpret medical images, predict disease outbreaks, and personalize treatment plans. For instance, ML models have been developed to detect anomalies in medical imaging with high accuracy, aiding in early disease detection [4].

1.2.2 Operational Efficiency

Beyond diagnostics, ML enhances operational efficiency within healthcare systems. It aids in resource allocation, patient scheduling, and management of healthcare records, thereby improving overall service delivery. The integration of ML in healthcare operations has been shown to streamline processes and reduce administrative burdens [5].

1.2.3 Challenges and Considerations:

While ML offers significant benefits, challenges such as data privacy, algorithmic bias, and the need for large, high-quality datasets must be addressed. Ongoing research focuses on developing robust, ethical, and transparent ML models to ensure their safe and effective integration into healthcare [6].

1.3 Problem Statement

Lumpy Skin Disease (LSD) is a viral infection affecting cattle, leading to significant economic losses due to reduced productivity and high mortality rates. Traditional diagnostic methods are often delayed, inaccurate, and resource-intensive, making timely intervention challenging. The availability of data on LSD is typically limited and imbalanced, which hampers effective analysis and prediction. These limitations necessitate a data-driven approach that can handle the inherent challenges of imbalanced datasets and provide reliable early detection to mitigate outbreaks effectively.

1.4 Objectives of the Study

The primary objectives of this project are:

1. To preprocess and analyze a dataset on Lumpy Skin Disease.
2. To evaluate the performance of some machine learning algorithms on the dataset.
3. To address the imbalance in the dataset using oversampling (SMOTE, ADASYN) and undersampling (NearMiss) techniques.
4. To improve model accuracy by employing ensemble methods such as Voting Classifiers.
5. To provide a comparative analysis of algorithm performances with and without resampling technique

1.5 Organization of Thesis

This thesis is structured as follows:

• Chapter 2: Review of Literature—provides an overview of related work and research papers on the application of machine learning for binary classification of diseases and addressing imbalanced datasets.

• Chapter 3: Methodology—explains the data preprocessing steps, model selection, and implementation of ensemble methods.

• Chapter 4: Results—presents the performance of various models under different conditions and provides insights.

• Chapter 5: Conclusion—summarizes findings, challenges, and suggestions for future work.

• Chapter 6: References—lists all the references cited in the thesis.

Chapter 2

Review of Literature.

Lumpy Skin Disease is a viral infection that significantly impacts cattle health and the livestock industry. Traditional diagnostic methods for LSD are often time-consuming and may lack accuracy, underscoring the need for advanced diagnostic tools. Recent studies have explored the application of machine learning (ML) techniques to enhance the early detection and diagnosis of LSD.

[7] conducted a comparative analysis of pretrained deep learning models for the early detection of LSD in cattle. Their study implemented various models to classify cattle images as either healthy or affected by LSD, aiming to identify the most accurate and reliable models for this purpose. The findings suggest that deep learning approaches can significantly improve the early detection of LSD, facilitating timely intervention and control measures (MDPI, 2024).

Another study [8] emphasized the potential of deep learning in diagnosing LSD. The researchers evaluated different pretrained models to determine their efficacy in detecting LSD from cattle images. The results indicated that certain models achieved high accuracy, reinforcing the viability of deep learning techniques in veterinary diagnostics (PLOS ONE, 2024).

Addressing the challenge of imbalanced datasets in ML,[9] introduced the Random Data Partitioning with Voting Rule (RDPVR) method. This technique involves dividing an imbalanced dataset into multiple balanced sub-datasets, training a classifier on each, and then applying a majority voting rule to obtain the final classification. The RDPVR method has shown promise in improving classification performance on imbalanced datasets, which is pertinent to LSD diagnosis where data imbalance may be a concern (MDPI, 2022).

The Voting Classifier, an ensemble learning method that combines multiple base models to produce an optimal solution, has been applied in various studies to enhance classification performance. For instance, a Kaggle tutorial demonstrated the implementation of a Voting Classifier to achieve better results in classification tasks. This method can be particularly useful in medical diagnostics, including LSD detection, where combining different models may lead to improved accuracy (Kaggle, 2024) [10].

Furthermore, a study on ensemble machine learning with the Voting Classifier for imbalanced data highlighted its effectiveness in creating lightweight computational models. The research demonstrated that the Voting Classifier could serve as a robust classifier when dealing with imbalanced datasets, which is a common issue in medical data, including LSD cases (JEEEMI, 2023) [11].

2.1 Link to This Study

This project builds upon the findings from these studies by focusing on the application of the Voting Classifier as a primary technique for binary classification of LSD in cattle. Unlike studies that utilize multiple ensemble methods, this work exclusively employs the Voting Classifier, ensuring a streamlined approach. Additionally, the project addresses the challenges of imbalanced datasets by integrating oversampling techniques such as SMOTE and ADASYN, inspired by insights from existing literature. By combining these methods, this project contributes a novel perspective to the application of machine learning in veterinary diagnostics, with a focus on scalability and practical implementation.

2.2 Conclusion

The review highlights the significant potential of machine learning, particularly ensemble methods like the Voting Classifier, in enhancing the accuracy and efficiency of LSD diagnostics. By addressing gaps such as imbalanced data and leveraging existing methodologies, this project aims to provide a robust and scalable solution for early disease detection in cattle.

Chapter 3
Methodology

This chapter describes the step-by-step methodology employed to preprocess the data, select and train machine learning models, and address the challenges posed by the imbalanced dataset. The methodology is divided into three main sections: **data preprocessing**, **model selection**, and **ensemble method**.

3.1 Data Source and Description

The dataset used for this project was sourced from **Kaggle**, a popular platform for data science and machine learning challenges. The dataset focuses on environmental, climatic, and geographical factors that may influence the occurrence of **Lumpy Skin Disease (LSD)** in cattle. It is structured for binary classification, with the target variable indicating the presence or absence of the disease.

The dataset comprises a mix of numerical and categorical features, capturing the complex interplay of environmental conditions and disease occurrence. This comprehensive dataset serves as a valuable resource for building predictive models and understanding the dynamics of LSD outbreaks.

3.1.1 Feature Overview

The dataset consists of multiple features representing clinical and environmental factors that may influence the likelihood of LSD. These features are a mix of categorical and numerical variables. A brief overview of the key features is as follows:

Feature Name	Description	Type
Age	Age of the cattle in years.	Numerical
Breed	Breed of the cattle (e.g., local, hybrid).	Categorical
Location	Geographical location where the data was collected.	Categorical
Skin Lesions	Presence of visible skin lesions (Yes/No).	Categorical
Fever	Body temperature recorded during observation (in °C).	Numerical
Lactation	Whether the cattle is lactating (Yes/No).	Categorical
Vaccination Status	Indicates if the cattle was vaccinated for LSD (Yes/No).	Categorical

country	The name of the country where the data was recorded.	Categorical
tmX	maximum temperature.	Numerical
Lumpy	Does the cattle have disease? (Yes/No)	Numerical

The diverse range of features provides a robust foundation for training machine learning models, enabling them to capture patterns and relationships relevant to LSD prediction.

3.1.2 Target Variable

The target variable, **Lumpy**, is a binary classification label:

- **0**: Absence of Lumpy Skin Disease.
- **1**: Presence of Lumpy Skin Disease.

The target variable is imbalanced, with fewer instances of the disease present in the dataset compared to its absence. This imbalance reflects real-world scenarios where LSD outbreaks are relatively rare. Handling this imbalance is critical for building effective models.

3.2 Data Preprocessing

Effective preprocessing ensures the dataset is clean, consistent, and suitable for analysis. The following preprocessing steps were applied:

- Converting Categorical Values to Numerical Values
- Handling Missing Values
- Removing Outliers
- Normalization and Standardization

3.2.1 Converting Categorical Values to Numerical Values

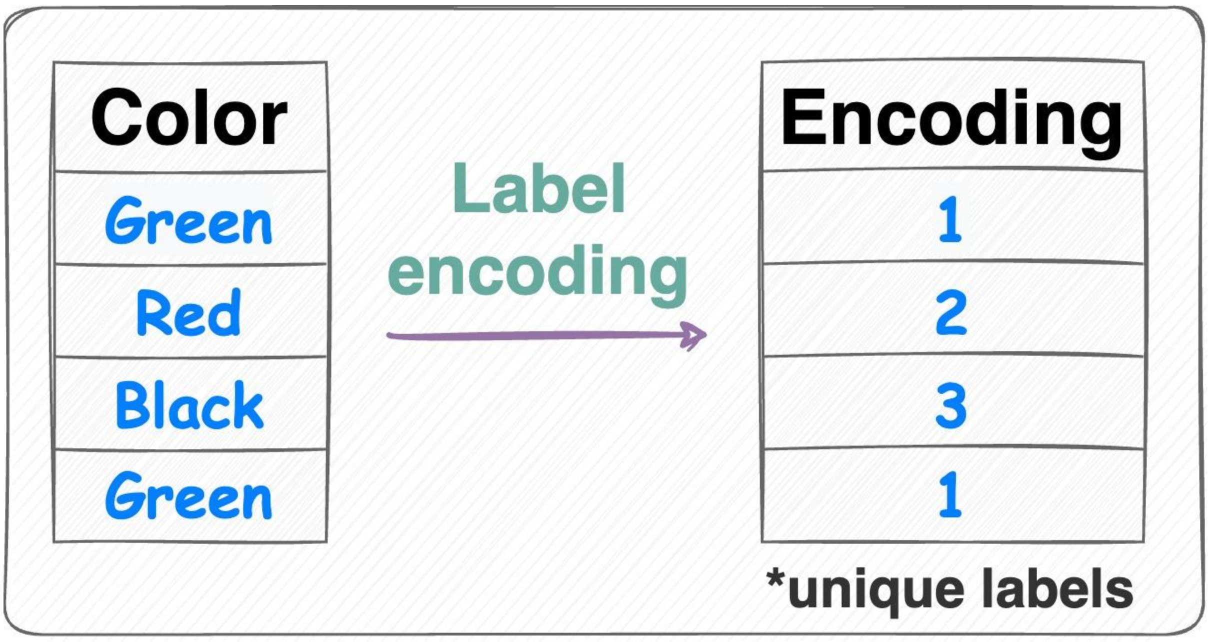
Categorical data refers to features that represent qualitative information, such as "Region" or "Vaccination Status." Machine learning models require numerical inputs, so categorical values must be transformed into numbers. In this project, two common encoding techniques were used: **Label Encoding** and **One-Hot Encoding.**

3.2.1.1 Label Encoding

Label Encoding is a simple method to convert categorical values into integers. It assigns a unique numerical value to each category in the feature. This approach is suitable when the feature has only **two unique values**, also known as **binary categories**.

3.2.1.1.1 How It Works

1. Identify the feature with binary categories.
2. Assign 0 to one category and 1 to the other.



3.2.1.1.2 Advantages

- Simple and efficient for binary categories.
- Requires minimal additional memory.

3.2.1.1.3 Limitations

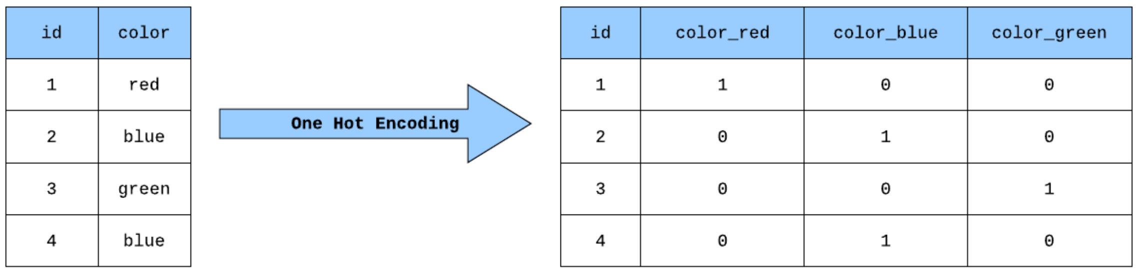
Should not be used when there are more than two categories, as it can introduce an unintended ordinal relationship between the encoded values (e.g., "Asia" = 0, "Africa" = 1, "Europe" = 2 implies an ordering that doesn't exist).

3.2.1.2 One Hot Encoding

One-Hot Encoding is used when a feature has more than two unique values. It transforms the feature into multiple binary columns, where each column represents one category, and a value of 1 indicates the presence of that category for a given data point.

3.2.1.2.1 How It Works

1. Identify the feature with multiple unique categories.
2. Create a separate binary column for each category.
3. For each row, set 1 in the column corresponding to the category and 0 in all others.


One Hot Encoding

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

3.2.1.2.2 Advantages

- Avoids introducing ordinal relationships between categories.
- Works well with algorithms that perform better with binary input (e.g., Logistic Regression).

3.2.1.2.3 Limitations

- Increases the number of columns, which can lead to higher memory usage, especially for features with many unique categories.

3.2.1.2.4 When to Use Label Encoding vs. One-Hot Encoding?

The choice between Label Encoding and One-Hot Encoding depends on the number of unique values in the feature:

1. ***Label Encoding***
    - Generally used when a feature has only **two unique categories**.
    - Example: For the feature **Lactation** with categories "Yes" and "No," label encoding is sufficient.
2. ***One-Hot Encoding***
    - Used when a feature has **more than two unique categories**.
    - Example: For the feature **Country** with categories {India, China, Bangladesh}, one-hot encoding creates three binary columns: "India," "China," and "Bangladesh."

3.2.2 Handling Missing Values

The dataset contained missing values in some features, which were addressed as follows:

- For **numerical features**, missing values were imputed using:
    - The **mean** for normally distributed features such as temperatures.
    - The **median** for skewed features like precipitation or vapor pressure.
- For **categorical features**, missing values were replaced with the **mode** (most frequent category).

This approach ensured that the dataset retained its integrity without introducing significant bias.

3.2.3 Normalization and Standardization

Feature scaling is a crucial step in preprocessing to ensure numerical features are treated uniformly across machine learning models. In this project, both **normalization** and **standardization** were applied to different features based on their characteristics and the requirements of the algorithms.

3.2.3.1 Normalization

Normalization rescales feature values to a fixed range, typically [0, 1]. This is particularly useful for algorithms that rely on distance metrics, such as Support Vector Machines (SVM) or k-Nearest Neighbors (kNN), as features with larger ranges can dominate the calculation.
The formula for normalization is:

$$x' = \frac{x - \min(x)}{\max(x) - mins(x)}$$

Where:

- x: Original value of the feature.
- $\min(x)$: Minimum value of the feature.
- $\max(x)$: Maximum value of the feature.
- x': Normalized value.

3.2.3.2 Standardization

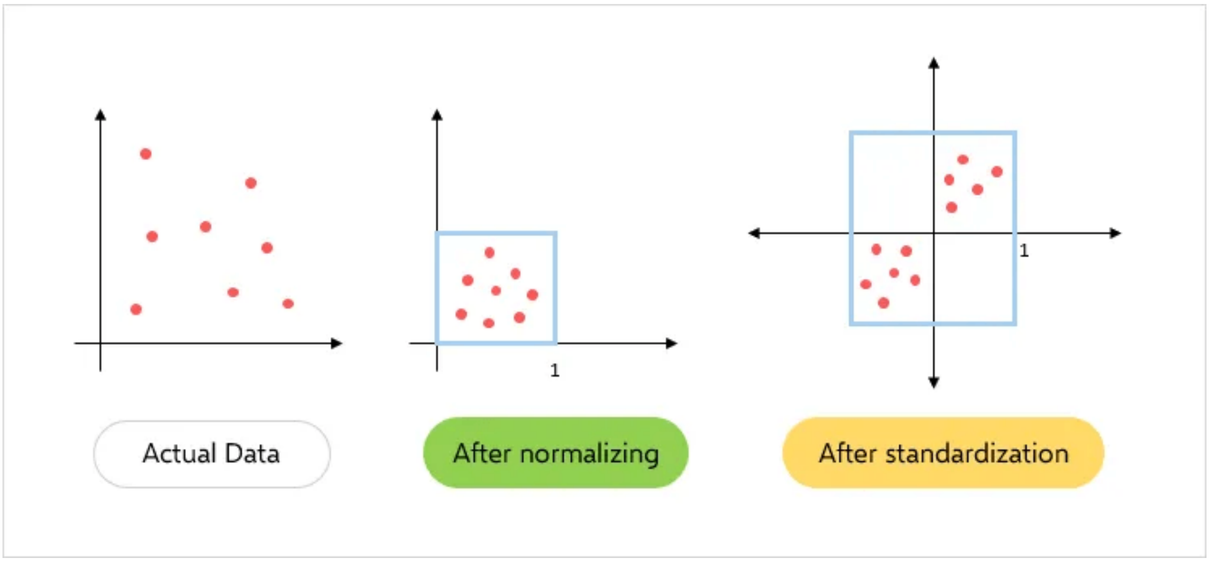
Standardization transforms features to have a **mean of 0** and a **standard deviation of 1**, making them comparable. It is particularly effective for algorithms like Logistic Regression and Neural Networks, which assume data is Gaussian-distributed.
The formula for standardization is:

$$z = \frac{x - \mu}{\sigma}$$

Where:

- x: Original value of the feature.
- μ: Mean of the feature.
- σ: Standard deviation of the feature.
- z: Standardized value (also called the Z-score).

Standardization is particularly effective when features have different units or scales, such as precipitation (measured in mm) and temperature (measured in °C).



By applying these data pre-processing the dataset was prepared to meet the requirements of a diverse set of machine learning algorithms, ensuring better convergence and performance.

3.2.4 Outlier Removal Using Z-Scores

Outliers can distort machine learning models, especially in regression-based or distance-based algorithms. In this project, Z-scores were used to identify and remove outliers from numerical features.

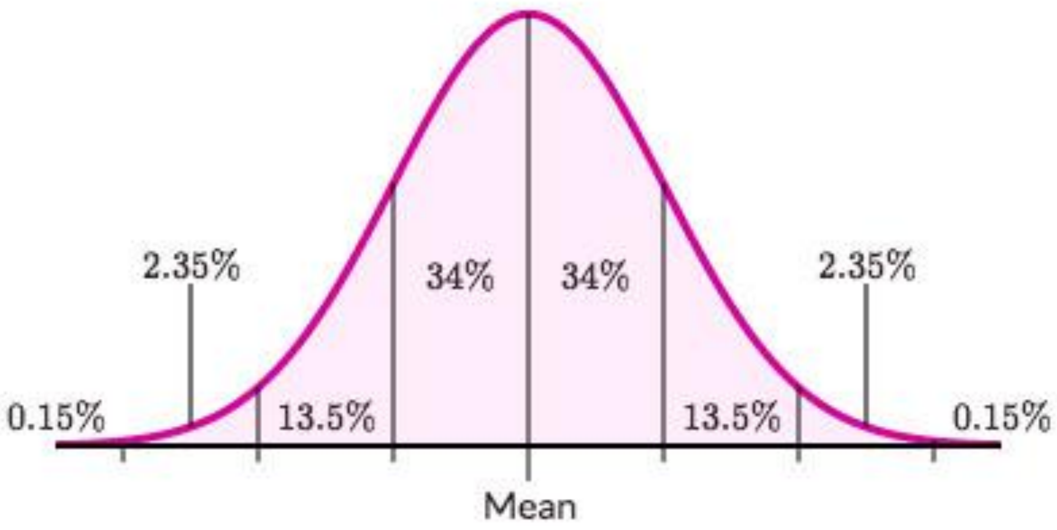
3.2.4.1 Process for Outlier Detection and Removal

1. Calculate the Z-score for each data point using the formula:

$$z = \frac{x - \mu}{\sigma}$$

2. Define a threshold, typically $|z|>3$, where:
   - $|z| \leq 3$: Data points within three standard deviations are retained.
   - $|z| > 3$: Data points beyond three standard deviations are considered outliers and removed.

This method ensures that extreme values, which may skew the model or introduce noise, are excluded from the dataset. Features were rechecked for balance after outlier removal to ensure model training was not affected negatively.

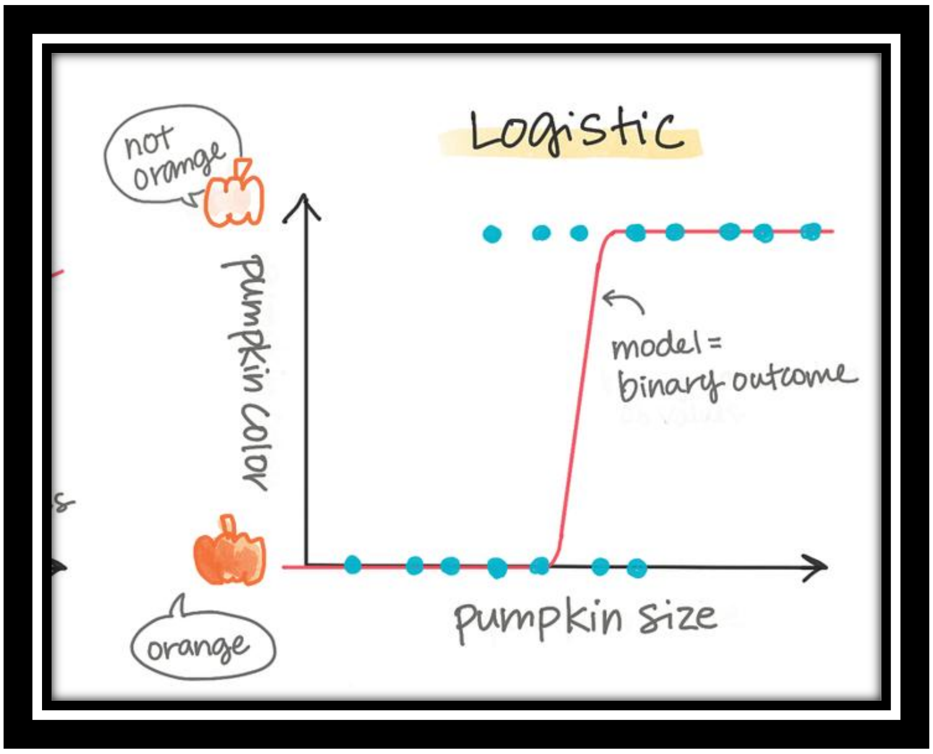


3.3 Baseline Algorithms

To establish a performance benchmark and evaluate the effectiveness of subsequent data preprocessing and modelling techniques, several baseline machine learning algorithms were initially trained and evaluated. These algorithms represent a range of model types commonly used for classification tasks. The following models were selected:

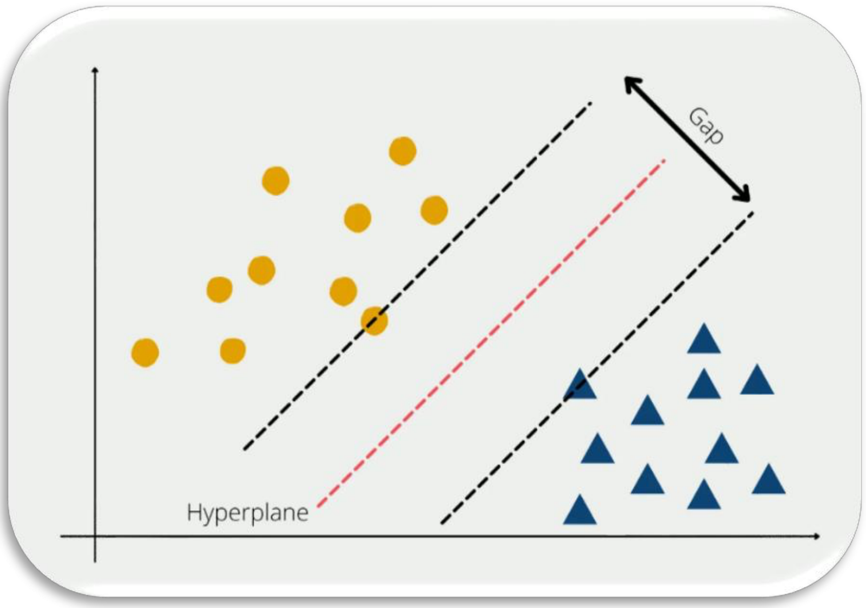
3.3.1 Logistic Regression (LR)

Logistic Regression is a linear model used for binary classification. It predicts the probability of a data point belonging to a particular class using a logistic function. Despite its simplicity, Logistic Regression often provides a robust baseline, especially when the relationship between features and the target variable is approximately linear.



3.3.2 Support Vector Machine (SVM)

Support Vector Machines aim to find the optimal hyperplane that maximally separates data points of different classes in a high-dimensional space. SVMs are effective in high-dimensional spaces and can handle non-linear data through the use of kernel functions. In this study, a linear kernel was used for the baseline SVM model.



3.3.3 Linear Discriminant Analysis (LDA)

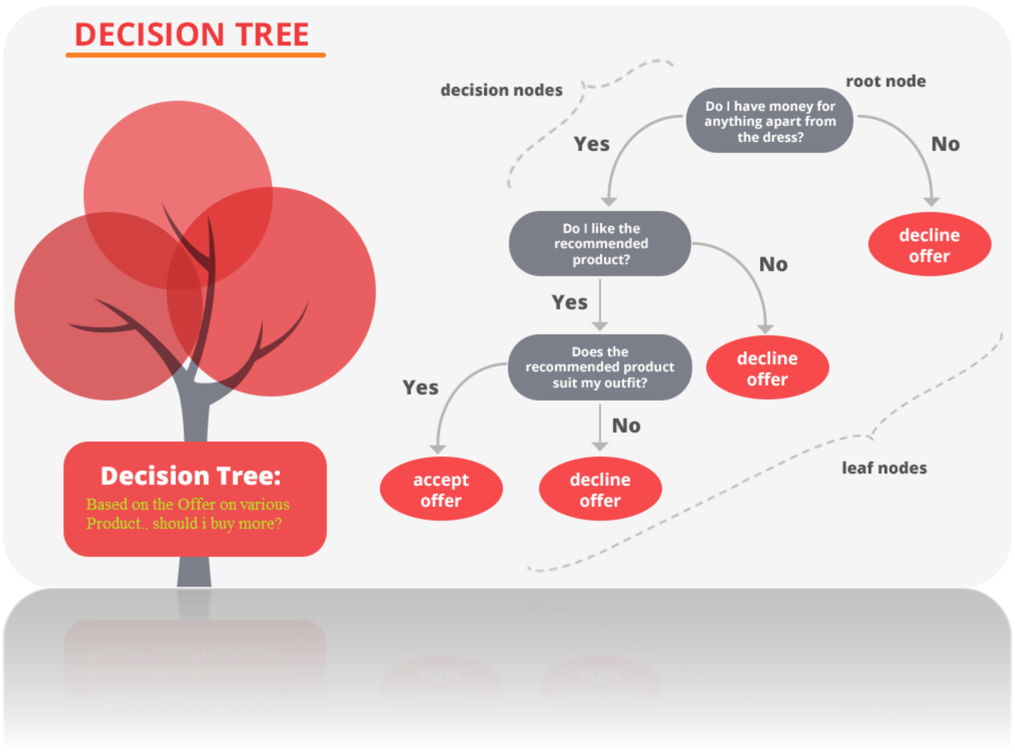
LDA is a dimensionality reduction and classification technique that seeks to find a linear combination of features that best separates classes. It assumes that the data within each class is normally distributed with equal covariance matrices. LDA is computationally efficient and performs well when these assumptions are reasonably met.

3.3.4 Gaussian Naive Bayes (Gaussian NB)

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem with the "naive" assumption of feature independence. It assumes that the features are normally distributed (Gaussian distribution). Despite its simplifying assumption, Gaussian NB can be surprisingly effective, especially in high-dimensional settings.

3.3.5 Decision Tree

Decision Trees are non-parametric supervised learning methods used for both classification and regression. They create a tree-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. Decision Trees are easy to interpret and can capture non-linear relationships in the data.


DECISION TREE
decision nodes
root node
Do I have money for anything apart from the dress?
Yes
No
Do I like the recommended product?
decline offer
No
Yes
Does the recommended product suit my outfit?
decline offer
Yes
No
accept offer
decline offer
leaf nodes
Decision Tree:
Based on the Offer on various Product.. should i buy more?

3.3.6 Stochastic Gradient Descent Classifier (SGD Classifier)

The SGD Classifier is a linear classifier (like Logistic Regression and SVM with a linear kernel) that uses Stochastic Gradient Descent for optimization. It is highly efficient for large datasets and can be used with different loss functions and regularization techniques. In this study, the hinge loss (equivalent to a linear SVM) was used.

These baseline models were trained using the raw, pre-processed data as mentioned in 3.2. Their performance was evaluated using accuracy, precision, recall, F1-score. The results of this initial evaluation provided a benchmark against which the performance of subsequent models, including those trained with oversampled data and under-sampled data, could be compared.

3.4 Class Imbalance Issue

Class imbalance is a prevalent challenge in machine learning, particularly when dealing with real-world datasets where one class is significantly underrepresented. In the context of predicting Lumpy Skin Disease (LSD), this imbalance poses a significant risk of bias in model training, as models tend to favor the majority (non-LSD) class. In this study, the original dataset exhibited a substantial class imbalance, with 86% of instances classified as non-LSD and only 14% as LSD. This imbalance can lead to several issues:

- ***Biased Predictions*:** Models may predominantly predict the non-LSD class, neglecting the crucial task of identifying LSD cases.
- ***Misleading Metrics*:** Accuracy can be a deceptive metric, as a model predicting only non-LSD could still achieve high accuracy due to the class distribution.
- ***Poor Generalization*:** Models trained on such imbalanced data may struggle to generalize to unseen data, particularly in correctly identifying LSD cases.

To address these challenges, both over-sampling and under-sampling techniques were explored, and applied *exclusively to the training set to prevent data leakage*.

3.4.1 Oversampling Techniques

3.4.1.1 SMOTE (Synthetic Minority Over-sampling Technique):

SMOTE is an oversampling technique designed to address class imbalance by generating synthetic samples for the minority class. Unlike simple oversampling methods that duplicate existing instances, SMOTE creates new instances by interpolating between existing minority class samples and their nearest neighbors in the feature space.

3.4.1.1.1 Mathematical Formulation of SMOTE:

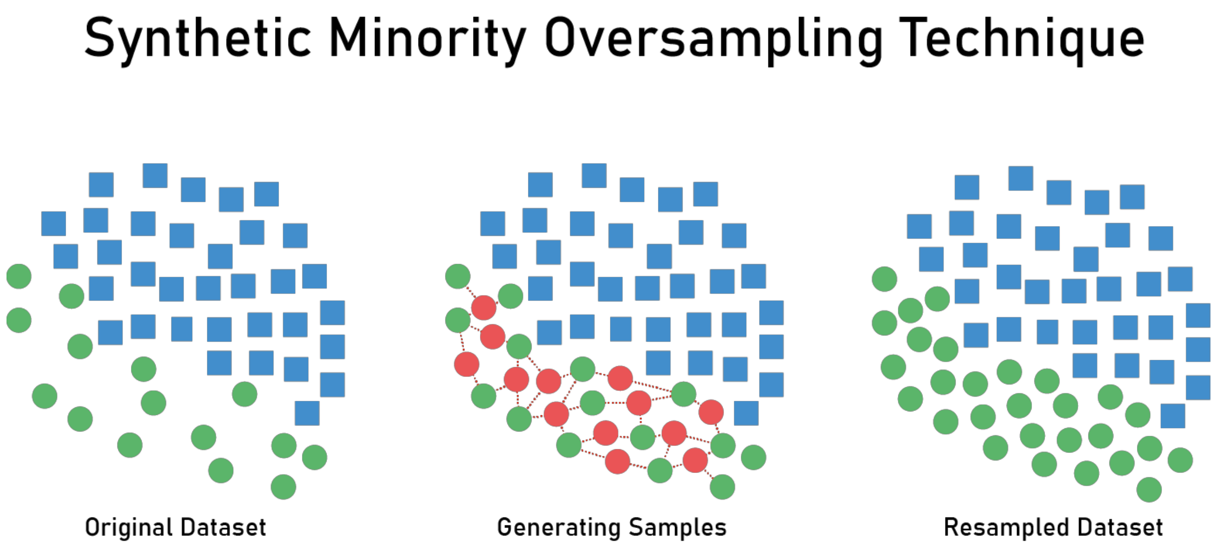
1. Let x_i represent a minority class instance within the dataset. To generate a synthetic sample, SMOTE proceeds as follows:
2. For each minority instance x_i, identify its k nearest neighbours, also belonging to the minority class.
3. Randomly select one of these k neighbors, denoted as x_j.

   Generate a new synthetic instance, x_s, using linear interpolation:

   $x_s = x_i + \lambda(x_j - x_i)$

   where λ is a random number in the interval [0, 1].

This process effectively creates synthetic samples along the line segments connecting the original minority class instances, thereby preserving the feature diversity of the minority class and mitigating the risk of overfitting associated with simple duplication.



3.4.1.1.2 Advantages

- ***Mitigates Overfitting:*** By generating synthetic samples rather than replicating existing ones, SMOTE reduces the risk of overfitting, leading to better generalization performance on unseen data.
- ***Preserves Feature Diversity:*** The interpolation process ensures that the synthetic samples are distributed within the feature space occupied by the minority class,

maintaining the diversity of the dataset and preventing the creation of identical or near-identical instances.

3.4.1.2 Adaptive Synthetic Sampling (ADASYN)

ADASYN (Adaptive Synthetic Sampling) is an oversampling technique that builds upon SMOTE by adaptively generating synthetic samples for the minority class based on the difficulty of learning from existing samples. The core idea is to generate more synthetic data for minority class instances that are surrounded by more majority class instances (i.e., those that are harder to classify).

3.4.1.2.1 Mathematical Formulation of ADASYN

a. For each minority class sample x_i, calculate its imbalance level d_i:

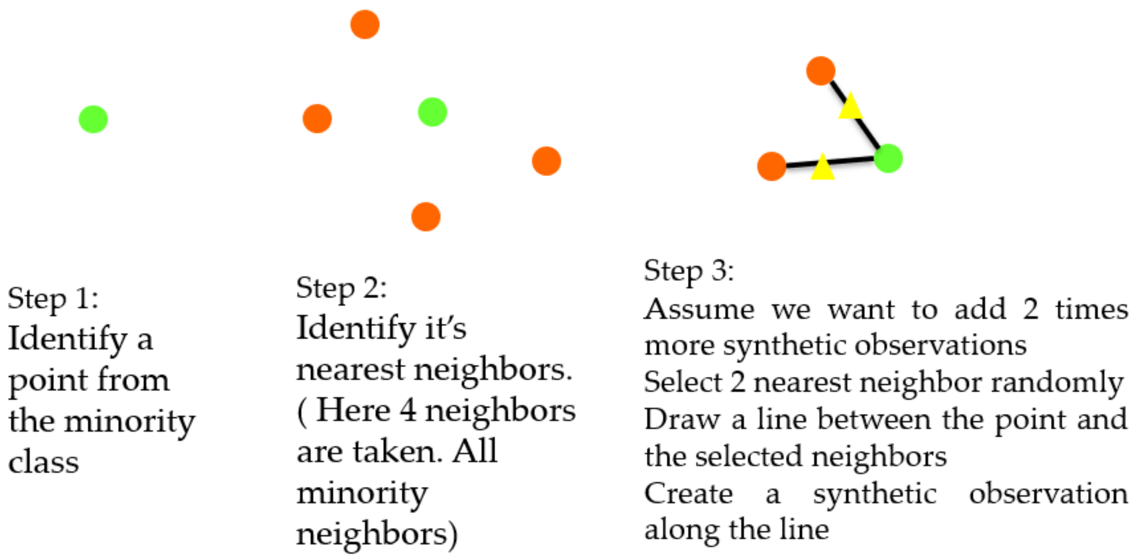
$$d_i = \frac{Number\ of\ Majority\ Class\ neighbors\ of\ x_i}{k}$$

where k is the total number of neighbors considered.

b. Normalize d_i to obtain the weight w_i:

$$w_i = \frac{d_i}{\sum_{i=1}^{m} d_i}$$

c. Generate synthetic samples proportionally to w_i, ensuring more synthetic data for harder-to-classify instances.


Step 1:
Identify a point from the minority class
Step 2:
Identify it's nearest neighbors. (Here 4 neighbors are taken. All minority neighbors)
Step 3:
Assume we want to add 2 times more synthetic observations
Select 2 nearest neighbor randomly
Draw a line between the point and the selected neighbors
Create a synthetic observation along the line

3.4.1.2.2 Advantages

- **Adaptive Sample Generation:** ADASYN adaptively generates more synthetic samples for minority class instances that are harder to learn (those surrounded by more majority class instances). This focuses the oversampling effort on the regions of the feature space where it is most needed.
- **Improved Decision Boundaries:** By focusing on difficult-to-classify instances, ADASYN can help improve the decision boundaries of the resulting models, leading to better classification performance, especially for the minority class.

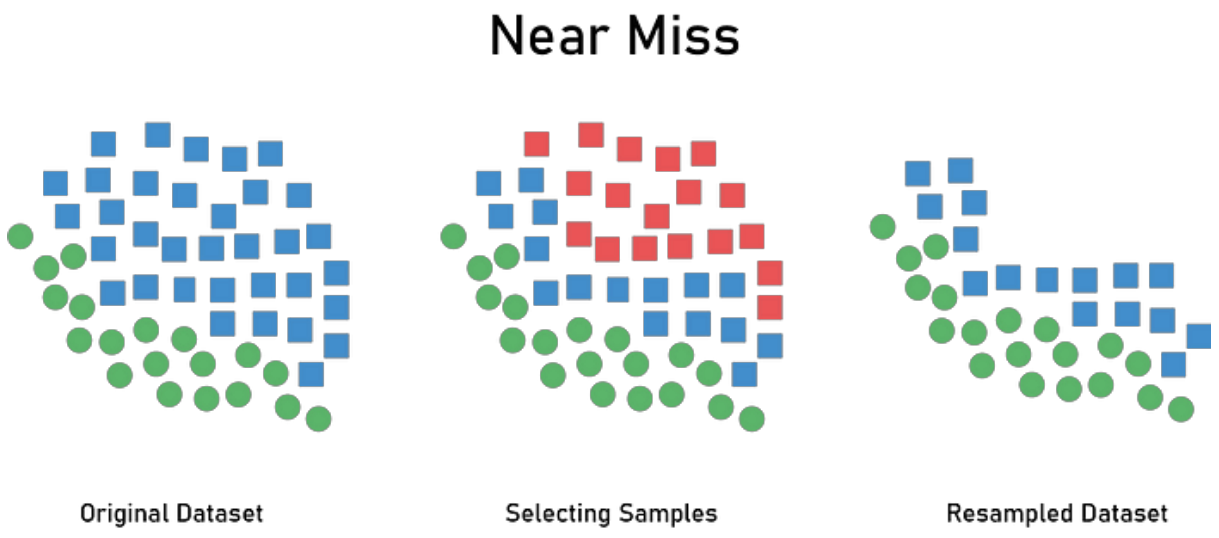
3.4.2 Under-sampling Technique

NearMiss:

NearMiss is an undersampling technique used to address class imbalance by selectively removing instances from the majority class. Unlike oversampling, NearMiss reduces the number of majority instances to create a more balanced dataset, focusing on retaining majority instances "close" to minority instances to preserve information near the decision boundary. Several versions exist, but the core principle involves selecting majority class instances based on their proximity to minority class instances.

3.4.2.1 Algorithm/Process (Generalized)

1. For each minority instance, find its *k* nearest majority class neighbors.
2. Select majority instances from these neighbors based on specific criteria (e.g., smallest average distance to minority instances, as in NearMiss-1).
3. Remove the remaining majority instances.



3.4.2.2 Advantages

- Improves the definition of the decision boundary by focusing on nearby majority instances.
- Reduces dataset size, potentially leading to faster training.

3.4.2.3 Disadvantages

- Potential loss of useful information by removing majority instances.
- Risk of overfitting if undersampling is too aggressive.

3.4.2.4 Experimental Approach

Each of these techniques (SMOTE, ADASYN, and NearMiss) was applied *independently* to the training data *after* the initial preprocessing steps (described in Section 3.2). The baseline models (Logistic Regression, SVM with a linear kernel, LDA, Gaussian NB, Decision Tree, and SGD Classifier with hinge loss) were then retrained and evaluated on each of the resulting training sets. This experimental design allowed for a direct comparison of the impact of each resampling technique on model performance.

3.5 Evaluation Metrics

To assess the performance of the classification models developed in this study, a set of evaluation metrics was employed. These metrics were chosen to provide a comprehensive evaluation, particularly in the context of the imbalanced dataset. The following metrics were used:

- Recall
- F1-score
- Precision
- Accuracy

3.5.1 Accuracy

Accuracy measures the overall correctness of the model's predictions. It is calculated as the ratio of correctly classified instances (both true positives and true negatives) to the total number of instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP: True Positives
- TN: True Negatives
- FPFP: False Positives
- FNFN: False Negatives

While accuracy is a commonly used metric, it can be misleading in imbalanced datasets, as a model that simply predicts the majority class for all instances can achieve high accuracy even if it performs poorly on the minority class.

3.5.2 Precision

Precision measures the proportion of correctly predicted positive instances (LSD cases in this study) out of all instances predicted as positive. It answers the question: "Of all the instances predicted as LSD, how many were actually LSD?”

$$Precision = \frac{TP}{TP + FP}$$

High precision indicates that the model has a low rate of false positives (i.e., it is good at avoiding predicting LSD when it is not present).

3.5.3 Recall (Sensitivity)

Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. It answers the question: "Of all the actual LSD cases, how many did the model correctly identify?"

$$Recall = \frac{TP}{TP + FN}$$

High recall indicates that the model has a low rate of false negatives (i.e., it is good at finding all the actual LSD cases).

3.5.4 F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of performance, particularly useful when dealing with imbalanced datasets. It is calculated as:

$$F1 - Score \ = \frac{Precision \, . Recall}{Precision + Recall}$$

The F1-score considers both false positives and false negatives, making it a more robust metric than accuracy alone in imbalanced scenarios.

3.5.5 Justification for Metric Selection

In this study, where the accurate detection of LSD (the minority class) is of primary importance, precision and recall are particularly relevant. We want to minimize both false positives (predicting LSD when it's not present) and false negatives (failing to detect actual LSD cases). The F1-score provides a way to balance these two competing objectives. While accuracy is reported for completeness, the F1-score, precision, and recall provide a more informative assessment of the model's ability to effectively detect LSD in the presence of class imbalance.

3.6 Model Selection and Ensembling (Voting Classifier)

3.6.1 Model Selection

Based on the evaluation of baseline models trained with various resampling techniques (as detailed in Section 3.4 on Handling Class Imbalance), three models demonstrated particularly strong performance, especially in terms of recall—a crucial metric for minimizing false negatives in Lumpy Skin Disease (LSD) detection. These models, all trained using SMOTE for class balancing, were selected as base learners for a hard-voting ensemble. The selected models are as follows:

1. **Logistic Regression with SMOTE**
2. **Linear Discriminant Analysis with SMOTE**
3. **Gaussian Naive Bayes with SMOTE**

The selection of these models was guided by their complementary strengths and the diverse decision boundaries they provide. Logistic Regression excels at linear relationships, LDA captures class separation efficiently, and Gaussian Naive Bayes is effective in probabilistic classification.

3.6.2 Hard Voting Ensemble

A hard voting classifier was implemented using scikit-learn's VotingClassifier in Python. Hard voting combines the predictions of multiple base learners by taking a majority vote, thereby leveraging the strengths of each individual model. The process is as follows:

1. Each base learner independently predicts the class label (LSD or non-LSD) for a given input instance.
2. The class label that receives the majority of votes from the base learners is assigned as the final prediction of the ensemble.

To ensure reproducibility, the default parameters for each algorithm, as implemented in scikit-learn, were used. No additional hyperparameter tuning was performed on the individual base learners prior to their inclusion in the ensemble.

3.6.2.1 Rationale for Using Hard Voting

Hard voting was chosen for its simplicity, interpretability, and ability to combine the strengths of diverse models. By leveraging the unique decision boundaries learned by Logistic Regression, Linear Discriminant Analysis, and Gaussian Naive Bayes, the ensemble achieves a more robust and accurate prediction. The perfect recall and high precision make it particularly suitable for critical applications like disease detection, where minimizing both false negatives and false positives is essential.

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents and discusses the results obtained from the experiments conducted to develop a machine learning model for accurate and early detection of Lumpy Skin Disease (LSD) using readily available data. A key challenge in this task is the inherent class imbalance present in the dataset, where instances of non-LSD significantly outnumber those of LSD. This imbalance can bias machine learning models, leading to poor performance in detecting the minority class (LSD), which is the primary focus of this study. Therefore, a core component of this research is the investigation of different resampling techniques to mitigate the effects of this imbalance.

This chapter addresses the following key research objectives:

- To develop a machine learning model for accurate and early detection of Lumpy Skin Disease using readily available data.
- To investigate whether the use of resampling techniques significantly improves the performance of classification models on imbalanced LSD data.

The chapter is structured as follows: Section 5.2 presents the performance of the baseline models before applying any resampling techniques. Section 5.3 analyzes the impact of the different resampling techniques (SMOTE, ADASYN, and NearMiss) on the baseline models. Section 5.4 presents the performance of the hard voting ensemble and compares it to the best-performing individual models. Section 5.5 provides a detailed discussion of the results, interpreting the findings and their implications. Section 5.6 discusses the challenges and limitations encountered during the study. Finally, Section 5.7 summarizes the key findings of this chapter.

4.2 Baseline Model Performance (Before Resampling)

This section presents the performance of the baseline models (Logistic Regression, SVM, LDA, GaussianNB, Decision Tree, and SGD Classifier) on the original, imbalanced dataset *before* resampling. *Preprocessing steps, including missing value imputation, encoding, and scaling, were applied as described in Section* [3.2]. Table 5.1 summarizes the performance metrics:

Table 5.1: Baseline Model Performance (Before Resampling)

Algorithm	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.9911	0.9723	0.9624	0.9641
SVM	0.9176	0.7702	0.9231	0.5928
Linear Discriminant Analysis	0.9852	1.0000	0.8952	0.9366
Gaussian Naive Bayes	0.9927	0.9449	0.8974	0.9717
Decision Tree	0.9998	0.9987	0.9123	0.9994
SGD Classifier	0.9236	0.6817	0.9465	0.7023

As shown in Table 5.1, some models achieve high accuracy, but the impact of class imbalance is evident. The recall scores for SVM, LDA, and GaussianNB are missing, hindering a full assessment. This highlights the need for resampling techniques to improve performance.

4.3 Impact of Resampling Techniques

This section evaluates the impact of the resampling techniques (SMOTE, ADASYN, and NearMiss) on the performance of the baseline models. Tables 5.2 to 5.4 summarize the results after applying each technique.

Table 5.2: Model Performance After SMOTE

Algorithm	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.9863	0.9299	0.9624	0.9459
SVM	0.7831	0.3589	0.9443	0.5202
Linear Discriminant Analysis	0.9892	0.9548	0.9585	0.9567
Gaussian Naive Bayes	0.9789	0.8549	1.0000	0.9218
Decision Tree	0.9998	0.9987	1.0000	0.9994
SGD Classifier	0.9002	0.5692	0.8148	0.6702

Table 5.3: Model Performance After ADASYN

Algorithm	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.9673	0.8096	0.9637	0.8799
SVM	0.7831	0.3589	0.9443	0.5202
Linear Discriminant Analysis	0.9655	0.8453	0.8847	0.8646
Gaussian Naive Bayes	0.9627	0.7697	1.0000	0.8699
Decision Tree	0.9998	0.9987	1.0000	0.9994
SGD Classifier	0.9002	0.5692	0.8148	0.6702

Table 5.4: Model Performance After NearMiss

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.8426	0.4408	0.9845	0.6089
SVM	0.8004	0.3662	0.8264	0.5076
Linear Discriminant Analysis	0.8839	0.5178	0.9780	0.6771
Gaussian Naive Bayes	0.9627	0.7697	1.0000	0.8699
Decision Tree	0.9942	0.9554	1.0000	0.9772
SGD Classifier	0.9002	0.5692	0.8148	0.6702

As shown in the tables, the resampling techniques had varying effects on the models. SMOTE generally improved recall for most models, particularly for Logistic Regression and SVM. ADASYN also improved recall for several models, while NearMiss had a mixed impact, significantly improving recall for some models like Logistic Regression and SVM but negatively affecting others. This suggests that SMOTE and ADASYN were more effective in addressing the class imbalance and improving the detection of the minority class (LSD) in this dataset.

4.4 Hard Voting Ensemble Performance

The hard voting ensemble, composed of Logistic Regression, Linear Discriminant Analysis, and Gaussian Naive Bayes, each trained with SMOTE, was evaluated, and its performance compared to the individual base learners. Table 5.5 presents the performance metrics for both the ensemble and the base learners.

Table 5.5: Performance Comparison of Hard Voting Ensemble and Base Learners (with SMOTE)

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression (with SMOTE)	0.9863	0.9299	0.9624	0.9459
Linear Discriminant Analysis (with SMOTE)	0.9892	0.9548	0.9585	0.9567
Gaussian Naive Bayes (with SMOTE)	0.9789	0.8549	1.0000	0.9218
Hard Voting Ensemble	0.9937	0.9519	1.0000	0.9754

As shown in Table 5.5, the hard voting ensemble demonstrated an improvement in accuracy and F1-score compared to the individual base learners. While the Gaussian Naive Bayes model with SMOTE achieved perfect recall (1.0) on its own, the ensemble maintains this perfect recall while also improving precision and F1-score. This indicates that the ensemble not only remains highly sensitive in detecting LSD cases but also becomes more precise in its predictions, reducing the number of false positives.

4.6 Challenges and Limitations

This study encountered several challenges and limitations that should be acknowledged. These limitations provide context for the interpretation of the results and suggest potential avenues for future research.

4.6.1 Dataset Limitations

While the dataset size was sufficient for the scope of this study, a key limitation lies in the lack of clear feature descriptions. The absence of detailed explanations for each feature made it challenging to fully understand their individual contributions to the model's predictive power. This lack of interpretability makes it difficult to draw definitive conclusions about the underlying biological or environmental factors associated with LSD. Furthermore, as the dataset was sourced from Kaggle, its representativeness of real-world LSD cases across different geographical regions and time periods is unknown. This limits the generalizability of the findings to broader contexts. Although the data appeared fairly consistent, without a detailed analysis of the data collection process, potential biases or inconsistencies cannot be entirely ruled out.

4.6.2 Methodological Limitations

Several methodological limitations should be acknowledged. First, no hyperparameter tuning was performed on the base learners (Logistic Regression, Linear Discriminant Analysis, and Gaussian Naive Bayes) before their inclusion in the voting ensemble. This means that the models were used with their default settings as implemented in scikit-learn. While this simplifies the experimental process, it is possible that optimizing the hyperparameters of each model could have further improved their individual performance and, consequently, the performance of the ensemble. Second, the evaluation methodology used a single train/test split. While this is a common practice, it can be sensitive to the specific random split of the data. Using cross-validation, such as k-fold cross-validation, would have provided a more robust and reliable estimate of the model's performance by averaging results across multiple train/test splits.

4.6.3 Resampling Techniques:

While SMOTE, ADASYN, and NearMiss were utilized to address the class imbalance, the inherent limitations of these techniques should be considered. SMOTE, by creating synthetic samples through interpolation, can potentially generate instances that do not accurately reflect the true distribution of the minority class, especially if the minority class is highly complex or contains outliers. ADASYN, while addressing some of SMOTE's limitations by focusing on harder-to-learn instances, might amplify noise if these instances are indeed outliers. NearMiss, by removing majority class instances, risks losing potentially valuable information that could be useful for defining the decision boundary.

Addressing these limitations in future research could further enhance the accuracy, robustness, and generalizability of LSD prediction models.

4.7 Conclusion/Summary.

This chapter presented and discussed the results of experiments conducted to develop a machine learning model for accurate and early detection of Lumpy Skin Disease (LSD). The key finding of this study is that the hard voting ensemble, combining Logistic Regression, Linear Discriminant Analysis, and Gaussian Naive Bayes, each trained with SMOTE, effectively improved the overall performance for LSD detection, particularly by enhancing precision and F1-score while maintaining perfect recall. This was demonstrated by the ensemble achieving an accuracy of 0.9937, a precision of 0.9519, a recall of 1.0, and an F1-score of 0.9754. Notably, while Gaussian Naive Bayes with SMOTE already achieved perfect recall (1.0), the ensemble further improved precision from 0.8549 to 0.9519 and the F1-score from 0.9218 to 0.9754. The ensemble also showed improvements in accuracy, precision, and F1-score compared to Logistic Regression and Linear Discriminant Analysis with SMOTE.
The use of SMOTE proved crucial in addressing the challenges posed by class imbalance and improving the performance of the classification models. The hard voting ensemble, combining the strengths of Logistic Regression, Linear Discriminant Analysis, and Gaussian Naive Bayes further enhanced the accuracy, precision, and robustness of LSD detection.
While this study provides valuable insights into LSD prediction, some limitations should be acknowledged, including the lack of detailed feature descriptions, the unknown representativeness of the Kaggle dataset, the absence of hyperparameter tuning, and the use of a single train/test split. Future research could explore hyperparameter tuning for the base learners, employ cross-validation for more robust evaluation, investigate larger and more diverse datasets, and explore other ensemble methods or cost-sensitive learning approaches.

In summary, this study demonstrates the effectiveness of combining resampling techniques, specifically SMOTE, with a hard voting ensemble for improving the detection of LSD using readily available data. The ensemble approach offers a promising avenue for developing more accurate and reliable diagnostic tools for this important disease.

References

1 Liang, Z., Yao, K., Wang, S., Yin, J., Ma, X., Yin, X., ... & Sun, Y. (2022). Understanding the research advances on lumpy skin disease: A comprehensive literature review of experimental evidence. *Frontiers in Microbiology*, *13*, 1065894.

2 Lumpy Skin Disease by European Food Safety Authority.

3 Davies, F. G. (1991). Lumpy skin disease of cattle: a growing problem in Africa and the Near East. *World Animal Review*, *68*(3), 37-42.

4 An, Q., Rahman, S., Zhou, J., & Kang, J. J. (2023). A comprehensive review on machine learning in healthcare industry: classification, restrictions, opportunities and challenges. *Sensors*, *23*(9), 4178.

5 Ardito, V., Cappellaro, G., Compagni, A., Petracca, F., & Preti, L. M. (2023). Implementation of Machine Learning Applications in Health Care Organizations: Protocol for a Systematic Review of Empirical Studies. *JMIR Research Protocols*, *12*(1), e47971.

6 Arvindhan, M., Rajeshkumar, D., & Pal, A. L. (2021). A review of challenges and opportunities in machine learning for healthcare. *Exploratory Data Analytics for Healthcare*, 67-84.

7 Senthilkumar, C., Vadivu, G., & Neethirajan, S. (2024). Early Detection of Lumpy Skin Disease in Cattle Using Deep Learning—A Comparative Analysis of Pretrained Models. *Veterinary Sciences*, *11*(10), 510.

8 Senthilkumar, C., Vadivu, G., & Neethirajan, S. (2024). Early Detection of Lumpy Skin Disease in Cattle Using Deep Learning—A Comparative Analysis of Pretrained Models. *Veterinary Sciences*, *11*(10), 510.

9 Hassanat, A. B., Tarawneh, A. S., Abed, S. S., Altarawneh, G. A., Alrashidi, M., & Alghamdi, M. (2022). Rdpvr: Random data partitioning with voting rule for machine learning from class-imbalanced datasets. *Electronics*, *11*(2), 228.

10 Voting Classifier for Better Results.

11 Jauhari, M. I., Wirakusuma, M. P., Sidqi, A., Putra, I. G. N. R., Wijayanto, I., Rizal, A., & Hadiyoso, S. (2024). Implementation of Ensemble Machine Learning with Voting Classifier for Reliable Tuberculosis Detection Using Chest X-ray Images with Imbalance Dataset. *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, *6*(4), 543-548.