**PROACTIVE HEALTH ANALYSIS OF CALF AND CATTLE**

**SHAMITA M**

**Register No: 2034045**

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

M.Sc. Artificial Intelligence and Machine Learning

OF ANNA UNIVERSITY



November 2023

**DEPARTMENT OF COMPUTING**

**(Artificial Intelligence and Machine Learning)**

**COIMBATORE INSTITUTE OF TECHNOLOGY**

**(Autonomous Institution affiliated to Anna University)**

**COIMBATORE – 641014**

**COIMBATORE INSTITUTE OF TECHNOLOGY**

**(Autonomous Institution affiliated to Anna University)**

**COIMBATORE 641014**

(Bonafide Certificate)

Project Work - I

Seventh Semester

**PROACTIVE HEALTH ANALYSIS OF CALF AND CATTLE**

Bonafide record of work done by

**SHAMITA M**

**(Register No:2034045)**

Submitted in partial fulfilment of the requirements for the degree of

M.Sc. (Artificial Intelligence and Machine Learning) of Anna University

November 2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Faculty Guide Head of the Department

Submitted for the viva-voce held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner External Examiner

## CONTENTS

**CHAPTER PAGE NO**

ACKNOWLEDGEMENT i

[SYNOPSIS ii](#_gjdgxs)

[PREFACE iii](#_30j0zll)

1. **INTRODUCTION 1**
   1. [ORGANIZATION PROFILE 1](#_1fob9te)
   2. [PROBLEM DEFINITION 2](#_3znysh7)
   3. [DESCRIPTIVE STATISTICAL SUMMARY 3](#_2et92p0)
   4. [OVERVIEW OF ANALYSIS 5](#_tyjcwt)
   5. [INFERENCES SUMMARY 8](#_3dy6vkm)
2. **DATA MODELING AND EXPLORATION 9**
   1. [PROBLEM ANALYSIS 9](#_1t3h5sf)
   2. [DATA MODEL 10](#_4d34og8)
   3. [EXPLORATORY DATA ANALYSIS 16](#_2s8eyo1)
3. **PREDICTIVE ANALYTICS PROCESS 17**
   1. [PREDICTIVE ANALYTICS MODEL 17](#_17dp8vu)
   2. [TOOL DESCRIPTION 22](#_3rdcrjn)
   3. [IMPLEMENTATION USING TOOL 24](#_26in1rg)
4. **ANALYTICAL MODEL EVALUATION 26**
   1. [PERFORMANCE MEASURES 26](#_lnxbz9)
   2. HYPOTHESIS TESTING / CONFUSION MATRIX 29
5. **ANALYSIS REPORTS AND INFERENCES 30**

[5.1 REPORTS / VISUAL FORMATS 30](#_35nkun2)

1. **CONCLUSION 41**

[**BIBLIOGRAPHY 4**](#_1ksv4uv)**2**

# ACKNOWLEDGEMENT

I sincerely thank **DR. A. RAJESHWARI**, Principal, Coimbatore Institute of Technology, for permitting me to undertake this project work and for her constant encouragement.

I am indebted to **DR. K. SAKTHI MALA**, Dean of Computing, Coimbatore Institute of Technology for her constant support and guidance throughout the duration of this project.

I am grateful to **DR. J. SHANA**, Head In-Charge, Department of Computing (Artificial Intelligence and Machine Learning), Coimbatore Institute of Technology for her encouragement and continual support throughout this project.

I would like to express my sincere gratitude to **MS. HEMASHREE**, Assistant Professor, Department of Computing (Artificial Intelligence and Machine Learning), Coimbatore Institute of Technology for her invaluable guidance and support through all phases of this project work.

I would like to thank **Mr. JOSE CHITY**, Chief Operating Officer of SmartBell, for being my mentor and guiding me in the successful completion of my project.

I extend my heartfelt gratitude to my Chief Executive Officer, **Miss VENNA ADITYAN**, of SmartBell, for her valuable assistance during the project period.

I perceive this opportunity as a learning curve in my career development in the field of analytics. The learning has been great and wish to learn more and explore new avenues of practical learning through work experience.

# SYNOPSIS

The main objective of the project, titled 'Proactive Health Analysis of Calves and Cattle,' is to achieve early sickness detection, assess respiratory capacity, and identify behaviors such as standing, lying, and ruminating for farms integrated with SmartBell. Additionally, it aims to automate feeding machine data using predictions for feed intake, feeding times, and other related factors

The existing system has the ability to detect sickness in cattle and calves by visual observation but lacks the capability to identify respiratory organ issues and the underlying causes of sickness. These are the issues with the existing system.

The proposed system, 'Proactive Health Analysis,' is an analytics project designed for cattle and calf farms in the UK. Its primary objective is to create a unified model for early sickness detection in animals and to take timely preventive measures based on alerts. Additionally, the project establishes a system to closely monitor lung health, all with the aim of enhancing overall animal health. The proposed system achieves sickness prediction through a comprehensive analysis of the 10 Hertz data

The system processes the 10 Hertz data, which is considered as raw data because it is collected from the ear tags of each animal on the farms. The system then generates features from this raw data and performs ensemble analysis techniques.

# PREFACE

**CHAPTER I – INTRODUCTION** provides an overview of the problem definition, the analytical model used for descriptive statistical data summary, and the overview of analysis.

**CHAPTER II - DATA MODELING AND EXPLORATION** provides an overview of the dataset, techniques, and models used for automation. It includes a detailed discussion of a comparative study of different techniques, the selection of specific techniques, and the associated parameters.

**CHAPTER III - PREDICTIVE ANALYTICS PROCESS** provides an overview of the models used for prediction, including the tools, packages, software used for implementation, and the implementation code

**CHAPTER IV - ANALYTICAL MODEL EVALUATION** gives the performance measure, hypothesis testing, and test results.

**CHAPTER V - ANALYSIS REPORTS AND INFERENCES** discusses the detailed inference and visual representation of the analysis reports.

**CHAPTER VI - CONCLUSION** puts forth the features and suggestions for future enhancements of this project.

# 

# CHAPTER - I

# INTRODUCTION

# 

## This chapter provides a detailed description of the organizational environment for which the project is developed. It also offers a brief overview of the problem statement and the analysis conducted to achieve the project's objectives.

## 1.1 ORGANIZATION PROFILE

SmartBell is a pioneering company in the field of animal health monitoring technologies, offering a range of innovative solutions for animal welfare. Their flagship product, WellCalf, is a game-changer in calf rearing systems, designed to improve livestock management.

WellCalf stands out as an advanced system for monitoring calf health, providing early detection of diseases up to three days before visible symptoms appear. This not only facilitates quicker intervention but also reduces long-term health risks. SmartBell's approach includes lightweight tags, expert data analysis, and the ability to monitor behavior, activity, and temperature.

SmartBell's WellCalf system delivers several advantages, including improved weight gains, lower mortality rates, enhanced age at first calving, increased future milk yields, and a reduced need for antibiotics. These benefits contribute to better overall herd health and profitability for farmers.

The company is dedicated to enhancing animal welfare and offers additional features like environmental monitoring to detect stress conditions. SmartBell collaborates closely with farmers, veterinarians, and farm teams, considering themselves an integral part of the customer's operation. Their advanced data analytics and dedication to animal health make them a leader in the field, with solutions suitable for dairy grazing, dairy indoor environments, and calf and heifer rearing.

**1.2 PROBLEM STATEMENT**

**OBJECTIVE**

The “Proactive Health Analysis” project aims to automate the early detection of illnesses in cattle and calves, as well as behavioral identification (standing, lying, feeding, ruminating). This will be achieved by analyzing features extracted from ear tag sensor data and feeding machine data.

**SCOPE**

The proposed system aims to develop an advanced data analysis system that streamlines cattle and calf monitoring, significantly reducing the need for manual oversight. The system will proactively generate alerts when necessary, ensuring the animals receive timely care. This approach is designed to enhance animal health, minimize mortality rates, increase milk yields, and reduce antibiotic usage.

The project encompasses a comprehensive system with integrated components. These include providing daily reports and specific alerts with detailed descriptions. These alerts are further complemented by comprehensive investigation reports, empowering more effective farm management. By automating monitoring and delivering valuable insights, it is dedicated to enhancing animal welfare, boosting farm productivity, and advocating sustainable agricultural practices.

**USERS**

The envisioned Proactive system is meticulously tailored to meet the unique needs of farms operating within the United Kingdom. It has been fine-tuned to address the specific challenges and requirements of the region's agricultural landscape, ensuring its effectiveness in diverse farming scenarios.

**1.3 DESCRIPTIVE STATISTICAL SUMMARY**

***DATASET DESCRIPTION:***

The dataset encompasses a diverse range of data attributes, making it a multimodal dataset. It includes time-series data, feeding behavior records, and essential metadata. The data sources comprise ear tag acceleration data, biocontrol feeding information, calf metadata, sickness status records, and labeled behavior descriptions. The ear tag acceleration data offers insights into animal movement, while biocontrol feeding details encompass feeding times, animal weights, and feed consumption specifics. Calf metadata provides information about birth dates and breed types, and sickness information indicates the daily health status of animals. Lastly, labeled behavior data records key behaviors and their durations. This rich dataset presents a unique opportunity for a comprehensive analysis, offering valuable insights into both animal health and behavioral patterns.

***SUMMARY:***

Data analysis can be done more effectively by first understanding the nature of the data before applying any machine learning features. In this project, exploring the data is crucial for two reasons: it leads to better model building and helps in addressing the unique challenges faced by each farm. These challenges include the impact of the surroundings, the presence of standard animals, and the identification of animals with poor health states, whose causes need to be analyzed. This, in turn, helps the farm identify and mitigate crucial issues, reducing their overall impact.

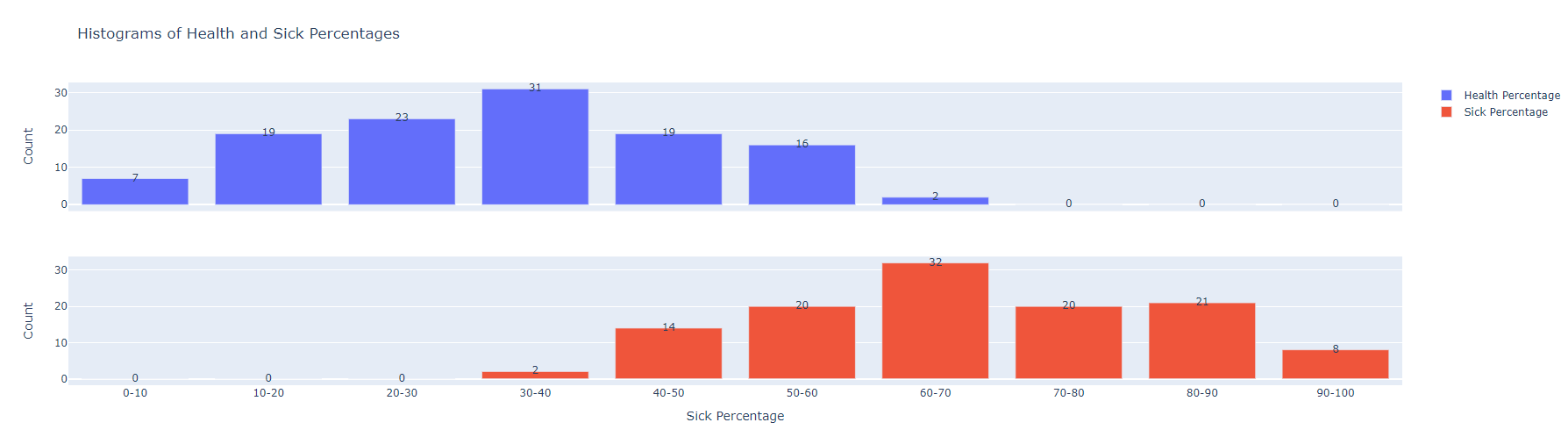


Figure 1.1 Histogram of sick and healthy percentage of calves

Figure 1.1 provides a descriptive analysis of the percentage of sick and healthy days among calves. Given that calves are more prone to sickness in their just-born state, it is essential to prioritize their health. This analysis allows us to exclude calves with a lower sick percentage and concentrate on those with a higher sick percentage for further examination.

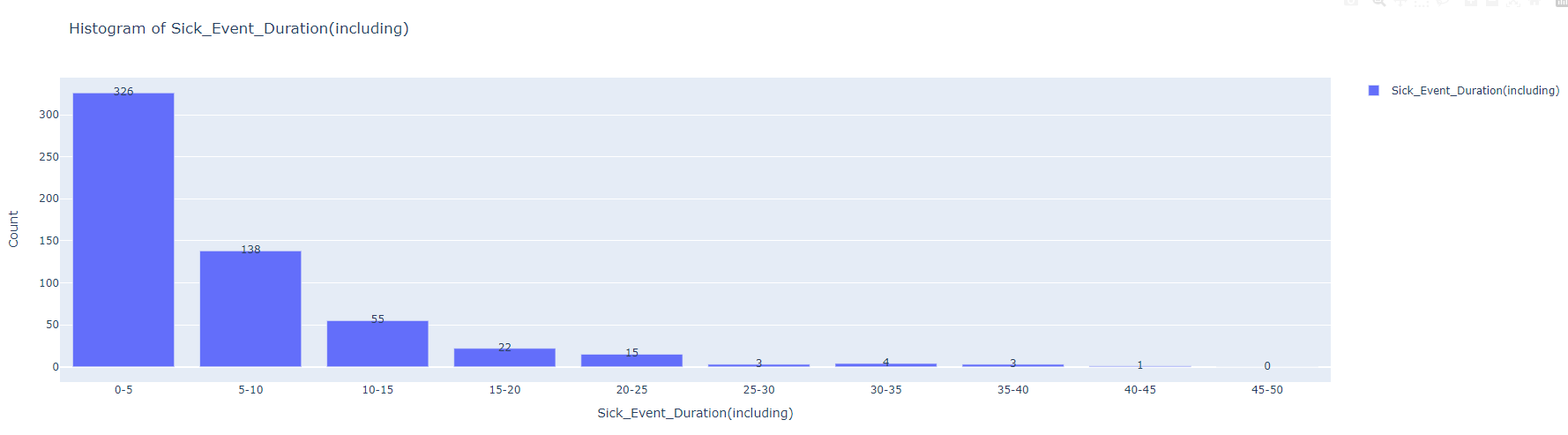
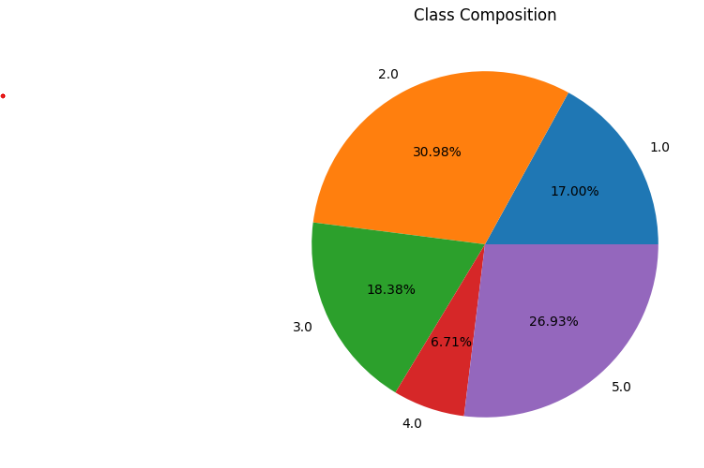
Figure 1.2 Sick Event Duration bins and their count

Figure 1.2 shows the distribution of sick events and the count of animals belonging to each sick event. A sick event is defined as the number of days animals were sick before returning to a healthy state. This shows the duration of sickness and helps to monitor the calves that are sick for long days.



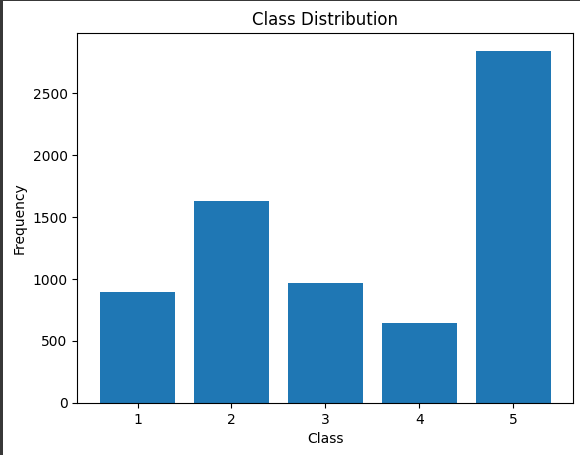


Figure 1.3 Distribution of Health States

Figure 1.3 displays the distribution of health states based on classes. In this classification, classes 1, 2, and 3 represent sickness, while classes 4 and 5 represent a healthy state. And it is inferred from the composition that the percentage of sick calf’s are higher than healthy, indicating the calf's will be mostly sick during its initial stage of life.

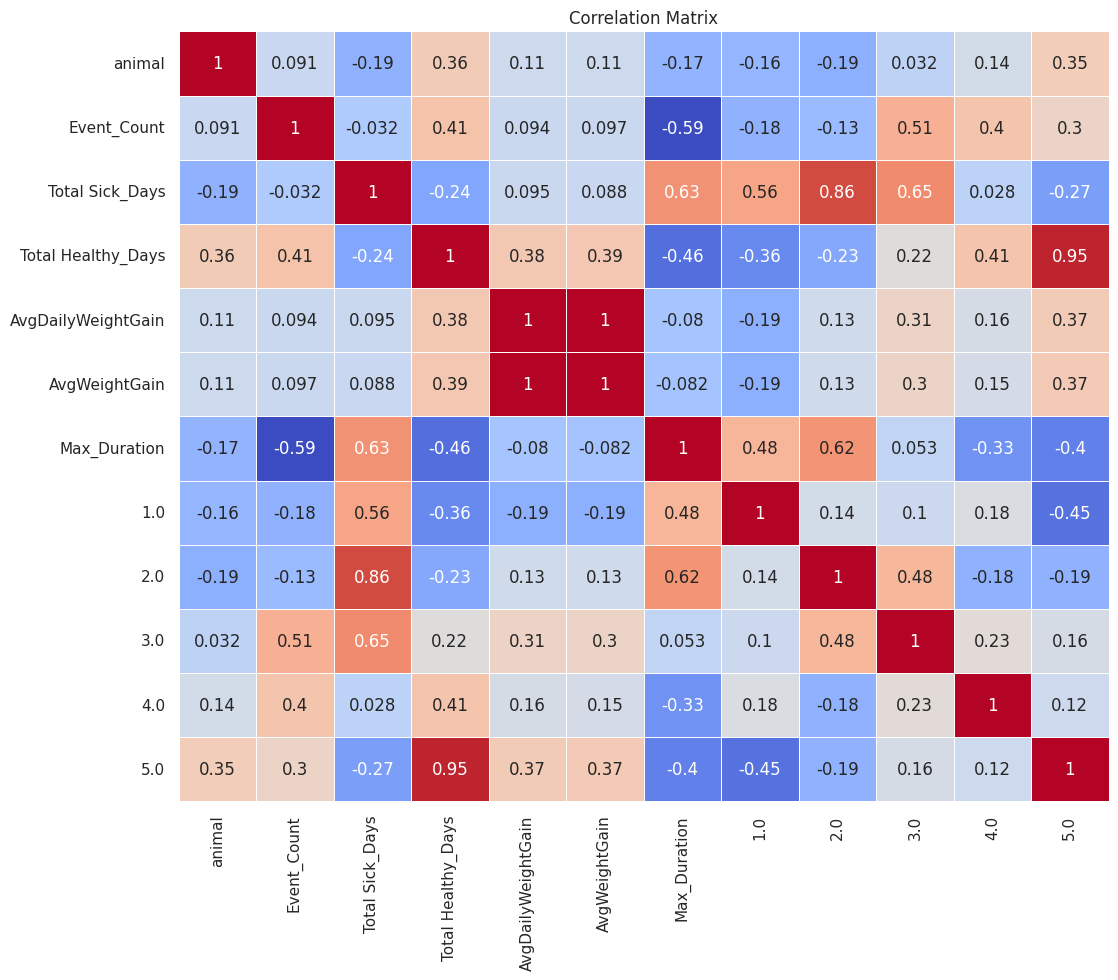


Figure 1.4 Correlation of all important features for prediction of sickness

# Figure 1.4 illustrates Maximum duration of sickness for the animals is predominantly spent in class 2, followed by class 1. If the animal remains in class 2 for an extended period, its likelihood of developing lung-related issues increases.

# Only the features that exhibit significant correlations are considered for prediction, and the effects of these features on each other are also identified.

# 

Figure 1.5 Difference in lung score with count of cattle

# Figure 1.5 illustrates the difference in lung scores, the higher the score the greater the chance of being affected by chronic sickness. This count can be useful in identifying the major scores and chance of calf in healthy category to move to sick.

# 1.4 OVERVIEW OF PREDICTIVE ANALYSIS

# *1.4.1 METHODOLOGY:*

# In the proposed structure, the implementation is bifurcated into two phases. The first phase entails the prediction of early detection of sickness, the assessment of respiratory capacity, and the generation of temperature-related alerts. These activities are based on the actual eartag data captured at 10 hertz and the feeding machine data. In the second phase, the focus shifts to predicting feeding machine data using the eartag data. This predicted data is subsequently utilized as a feature in the health or sickness classification process, contributing to more efficient and proactive livestock management.

# Behavior classification involves the assessment of estrus windows and the identification of various behavioral patterns. Functions for feature generation have been meticulously designed, rigorously unit-tested, and prepared for deployment. The exploratory journey extended to an array of machine learning models, through extensive evaluation the best-performing models have been selected, fine-tuned for optimal results. Cross-validation techniques have been employed to rigorously assess and validate the performance of these models. The power of ensemble methods has been harnessed to further enhance the accuracy and reliability of the predictions.

# *1.4.2 TOOLS AND TECHNOLOGIES USED:*

# Programming Language: Python

# Project Repository: BitBucket

# Framework:

# Scikit-learn

# TensorFlow

# Libraries:

# Pandas

# Keras

# Plotly

# Matplot

# APIs Used:

# Matplotlib.pyplot

# Pandas.dataframe

# Sklearn.cluster

# Sklearn.decomposition

# Sklearn.feature.selection

# Sklearn.transformer

# Keras.layers

# Keras.optimizers

# Keras.metrics

**1.5 INFERENCES SUMMARY**

The project initiates with the creation and deployment of a generalized code for feature generation, specifically from accelerometer data. The features required for the predictive models are identified using the SelectKBest method from mutual information. These predictive models are not only implemented but also rigorously evaluated through metrics tailored for handling unbalanced data.

# Artificial Neural Network (ANN) was developed for predicting key indicators of sickness. For the sickness prediction model, a combination of Random Forest (RF) and LightGBM was employed to create an ensemble model. For behavior classification, the model utilized Random Forest (RF) in conjunction with Gradient Boosting (GB). In the context of fan state prediction, a Decision Tree Classifier (DTC) was employed. The accuracies provided by the models are shown in the Table 1.1.

# Table 1.1. Accuracy of the learning models

|  |  |
| --- | --- |
| Model | Accuracy |
| Sickness prediction (Random Forest) | 90% |
| Behavior Classification (Ensembled Random Forest and Gradient Boosting) | Lying – 93%Ruminating – 92% |
| Fan-State Prediction (Decision Tree Classifier) | 93% |

# CHAPTER II

**DATA MODELING AND EXPLORATION**

Data modeling is the process of creating a visual representation of either a whole information system or parts of it to communicate connections between data points and structures. Data exploration is the initial step in data analysis, where users explore a large data set in an unstructured way to uncover initial patterns, characteristics, and points of interest.

**2.1 PROBLEM ANALYSIS**

***2.2.1 PROBLEM UNDERSTANDING:***

This project aims to enhance milk yield in livestock by analyzing data from eartags (sensor data) and feeding machines to identify sickness and behavior patterns. The data sources provide vital information for early sickness detection and behavioral insights. Behavioral analysis helps in recognizing estrus behavior in animals, which, in turn, aids in the identification of any underlying health issues or factors that might affect estrus detection rates. Additionally, in some farms, temperature-related issues are prevalent, and the eartag sensor can capture both the animal's temperature and ambient temperature, which enables the mitigation of temperature-related problems, further improving livestock health and milk production.

***2.1.2 BUSINESS UNDERSTANDING:***

Manual monitoring often falls short in providing early insights into potential issues. As a result, there is a compelling need for a data-driven approach to address various challenges, including feeding optimization, early illness detection, calfhood health, and resource efficiency. Changes in acceleration patterns can serve as early indicators of illness, facilitating timely intervention and treatment. Early intervention during calfhood is crucial to prevent lung damage, ensuring future calving and lactation success, thereby improving overall animal health outcomes. Furthermore, examining an animal's feed intake patterns not only helps in understanding their dietary needs but also provides valuable insights for optimizing feed purchases and determining the most effective feeding intervals.

## *2.1.3 FEATURE IDENTIFICATION:*

The process of feature identification plays a crucial role in the development of advanced animal health monitoring models. The primary focus of the endeavor is on leveraging data collected from the WellCalf sensor, which is integrated into each animal's eartag. Embedded in each animal's eartag, the sensor serves as a critical data collection point, aggregating information from individual animals and transmitting it to a central gateway for analysis.

The sensor captures a comprehensive range of information, movement patterns, weight fluctuations, milk intake, and various other critical attributes. From this raw data, essential features are derived to capture underlying patterns that are vital for the predictive models. This feature extraction step is of utmost importance because, without the identification of these crucial features, predictions cannot be made solely based on the raw data. It serves as the foundation for the models to understand and predict the target variables accurately.

**2.2 DATA MODEL**

***2.2.1 DATA COLLECTION:***

**Eartag Data (Sensor data):**

The raw data is sourced directly from the eartags of individual animals, serving as a comprehensive repository of information concerning their well-being and activity. Eartag data is sensor data, equipped with the ability to capture acceleration and gyro data along different directions, as well as the temperature of the animal and the surrounding ambient temperature. This rich dataset is aggregated into a central gateway, which is deployed at each farm.

Importantly, this data is collected at a high frequency, specifically at a rate of 10 Hertz, providing a data point for every millisecond. This high data capture rate ensures that even the most nuanced changes in animal behavior and environmental conditions are accurately recorded and can be utilized for comprehensive analysis and decision-making in livestock management.

Table 2.1. Eartag Data Feature and its Description

|  |  |
| --- | --- |
| **Feature Name** | **Description** |
| accX\_mean | Mean of accelerometer data (X-axis) within a 1-minute interval. |
| accY\_mean | Mean of accelerometer data (Y-axis) within a 1-minute interval. |
| accZ\_mean | Mean of accelerometer data (Z-axis) within a 1-minute interval. |
| gyroX\_mean | Mean of gyroscope data (X-axis) within a 1-minute interval. |
| gyroY\_mean | Mean of gyroscope data (Y-axis) within a 1-minute interval. |
| gyroZ\_mean | Mean of gyroscope data (Z-axis) within a 1-minute interval. |
| temp\_mean | Mean of temperature data within a 1-minute interval. |
| amb\_temp\_mean | Mean of ambient temperature data within a 1-minute interval. |
| activity\_zone | Zone classification based on activity data. |

**Feeding Machine Data:**

Simultaneously, data from the feeding machine is harnessed, providing a granular account of each feeding bout. This dataset captures not only the frequency and timing of feed intake but also the unique preferences of each animal. Such detailed records offer valuable insights into dietary patterns, consumption trends, and the overall nutritional health of the herd.

Table 2.2. Feeding Machine Data Feature and its Description

|  |  |
| --- | --- |
| **Feature Name** | **Description** |
| MeanWeight | Average weight of animals over a given duration, calculated as the mean of all recorded weights. |
| StdDevWeight | Measure of the amount of variation or dispersion in weights for each animal. |
| MeanEatingSpeed | Average eating speed of each animal. |
| StdDevEatingSpeed | Measure of the amount of variation or dispersion in eating speeds of each animal. |
| TotalFeedEaten | Total amount of feed consumed by each animal in each day. |
| StdDevFeedEaten | Measure of the amount of variation or dispersion in the quantity of feed eaten for each animal. |
| TotalVisitDuration | Total duration of visits made by each animal within each day. |
| StdDevVisitDuration | Measure of the amount of variation or dispersion in the duration of visits for each animal. |
| TotalDuration | Total duration of visits made by each animal. |
| TotalVisitCount | Total number of visits made by each animal. |
| Biocontrol.FeedType | Type of feed intake. |
| Biocontrol.visitDuration | Duration of visits made by each animal. |
| Amount | Amount of milk intake of the calf. |
| Temp\_Difference\_std | Standard deviation of the difference between the ear temperature of the animal and the ambient temperature. |
| TempGroup\_3\_Days | Grouping of temperature data over a 3-day window. |
| TempGroup\_7\_Days | Grouping of temperature data over a 7-day window. |

***2.2.2 DATA STANDARIZATION:***

This process is of paramount importance, particularly when dealing with high-frequency 10 Hertz data. Managing such data requires substantial computational resources. To streamline the process, the initial step involves verifying whether, within a 10-second window, at least 70% of the data points are available. If this condition is not met, the entire 10-second data segment is discarded. This approach serves a dual purpose: it conserves computational resources and ensures that the model isn't compromised by the inclusion of incomplete or sparse data points.

**2.2.3 DATA PREPROCESSING:**

The eartag features are significantly influenced by acceleration due to gravity, as each value extracted from the eartag of the animal contains an acceleration due to gravity component. This component needs to be removed from the data in order to obtain accurate and stable values. The ***Kalman filter*** can be customized as a low-pass filter to efficiently eliminate high-frequency components, such as acceleration due to gravity, from sensor data.

Rather than simply subtracting the actual value of acceleration due to gravity from the data, the Kalman filter identifies and accounts for the gravity's effect at various states, allowing it to effectively eliminate this influence and yield the true values.

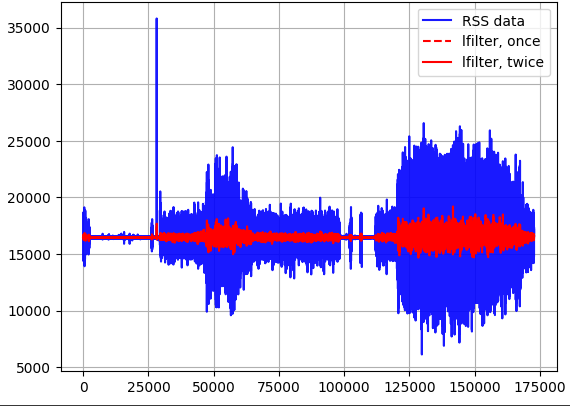
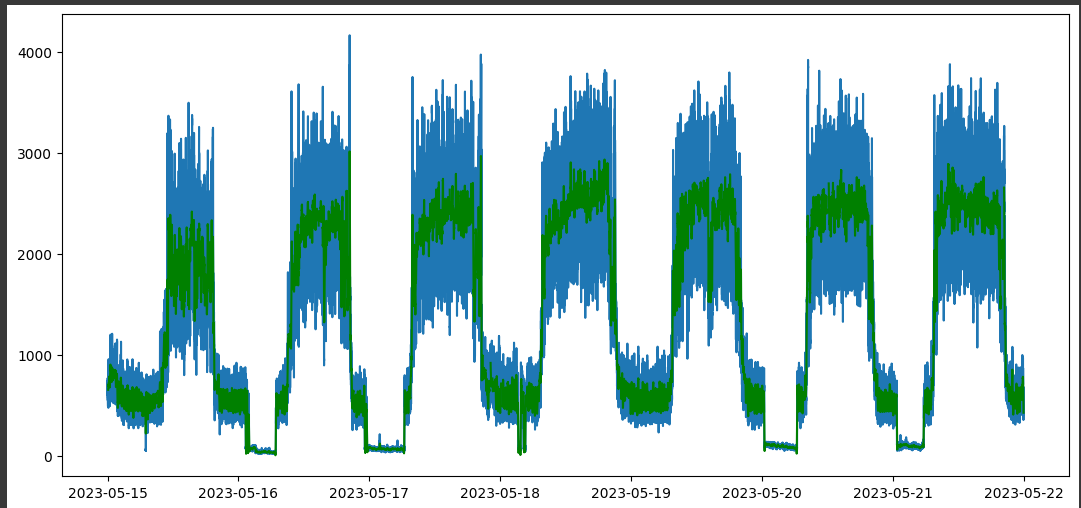


Figure 2.1- The red line represents the data filtered to remove the effect of gravity. The impact of gravity on the acceleration values is isolated and subsequently subtracted from the original data, effectively removing the gravitational component.

After processing the data with the Kalman filter, a rolling mean and ***Savitzky-Golay filter*** are applied to smooth out the data, making it easier for models to understand.



In Figure 2.2, the green color signifies smoothed values achieved using the Savitzky-Golay (Savgol) method. Savgol is a smoothing technique that operates by applying a sliding window and polynomial smoothing without causing a loss of information. It effectively removes noise and variations in the data, enhancing the clarity of underlying patterns while preserving essential data characteristics. This process is essential for improving data quality and facilitating more accurate analysis and predictions.

### 2.2.4 DATA TRANSFORMATION

One critical objective of data transformation is to ensure that all relevant features are standardized to a consistent scale. In the context of eartag data, attributes such as acceleration and gyro readings often exhibit substantial fluctuations. Using these unprocessed values directly in a model can compromise its robustness and predictive accuracy. To address this challenge, data scaling techniques are employed. The 'Standard Scaler' is utilized to rescale the data within the range of -1 to 1, which is particularly suitable for eartag data features following a Gaussian distribution

For features with distributions that differ from Gaussian, the "Min-Max Scaler" is utilized to rescale the data. This comprehensive data scaling ensures that the model performs at its best, capturing underlying patterns and relationships accurately.

***2.2.5 FEATURE SELECTION***

Various features are generated from the raw data to capture the patterns and to predict the target variable. To further refine the data input for optimal model performance, the selection of important features plays a pivotal role. These features are identified through various methods, including correlation techniques like Pearson and Spearman's correlation coefficients

Furthermore, the top features are determined by employing the SelectKBest method, which utilizes mutual information for feature selection, ensuring that only the most relevant attributes are used for model development. This process enhances the model's efficiency and accuracy by focusing on the key input variables.

**2.3 EXPLORATORY DATA ANALYSIS**

Exploratory Data Analysis summarizes main characteristics of dataset using graphical representations and statistical methods. EDA includes crucial operations such as data scaling, categorical variable encoding, and the creation of new features to improve data quality for subsequent modeling efforts. Additionally, correlation analysis is conducted to assess the relationships among variables. This phase equips the project with essential insights, enabling data-driven and informed decision-making in subsequent project stages.

**CHAPTER III**

**PREDICTIVE ANALYTICS PROCESS**

This section includes the predictive modeling process, description about the libraries, packages and tools used in the whole process along with implementation using the tools used in the process.

**3.1 PREDICTIVE ANALYTICS MODEL**

***3.1.1 TYPES OF ANALYSIS***

In this project, a classification model has been used to categorize structured data. Classification is a technique for categorizing data into a predetermined number of classes. Since the key indicators of the target and the target variables focused on in this project are categorical, models have been chosen accordingly. And for continuous target variables regression methods have been applied.

The following are the steps involved in building a classification model:

* **Initialize** the classifier to be used.
* **Feature Selection:** Generate and select the features that have high mutual information with the target.
* **Train the classifier:** All classifiers in [scikit-learn](https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/) uses a fit (X, y) method to fit the model (training) for the given train data X and train label y.
* **Predict the target:** Given an unlabeled observation X, the predict(X) returns the predicted label y.
* **Cross Validate** the results, keep individual subject at test which is totally unseen during training.
* **Evaluate** the classifier model.

There are various classification models like:

* Binary Classification
* Multi-Class Classification

**Binary Classification**

[Binary classification](https://en.wikipedia.org/wiki/Binary_classification) refers to those classification tasks that have two class labels.

* Health Identification (Healthy/Sick)
* Behaviour Identification (Lying/Standing)
* Ruminating Identification (Yes/No)

**Multi-Class Classification**

* Fan-State Identification (Off, On (Low Speed), On (High Speed)
* Feed Intake Identification (Low, Average, High)

**Regression Analysis**

* Weight of animal – predicted from age and various features extracted from eartag information.

### 3.1.2 CHOSEN MODEL

For every predictive target variable, a pipeline of the top 10 classifier models will be tested. From these pipelines, the top-performing models are selected.

**SICKNESS PREDICTION:**

To address the dataset's skewed label distribution, with more sick days than healthy ones, data was **upsampled** for improved model balance and accurate predictions. After selecting the optimal model, parameter fine-tuning was conducted using search techniques like grid search and **Optuna**. Notably, Optuna outperformed others, delivering the most effective and precise results in parameter optimization.

**RANDOM FOREST:**

Random Forest is the selected model for this project primarily because of its capacity to address overfitting and manage imbalanced data. In this context, there are more instances of healthy days compared to sick days.

The model functions by utilizing an ensemble of decision trees, with each tree being trained on a random subset of the data and its associated features. By amalgamating the predictions from these varied trees, Random Forest mitigates the risk of overfitting, guaranteeing that the model doesn't become overly specialized to the training data and can effectively extend its predictive capabilities to new, unseen data.

Table 3.1 Cross-validation results for sickness prediction using a RF model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | **Average** | **Median** | **Mode** | **Minimum** | **Maximum** |
| Sensitivity | 0.69 | 0.71 | 0.73 | 0.43 | 0.93 |
| Specificity | 0.57 | 0.59 | 0.56 | 0.26 | 0.80 |
| Positive Predictive Value | 0.76 | 0.77 | 0.69 | 0.61 | 0.91 |
| Negative Predictive Value | 0.48 | 0.49 | 0.52 | 0.20 | 0.70 |

**ENSEMBLING OF RANDOM FOREST WITH XGBoost AND LightGBM (HEALTH/SICK PREDICTION):**

XGBoost and LightGBM are advanced gradient boosting algorithms valued for their efficiency and accuracy. These algorithms employ boosting techniques to consolidate predictions from various models, resulting in improved metrics like accuracy, precision, and recall. By integrating XGBoost and LightGBM into the ensemble alongside Random Forest, we can tap into their strengths to boost overall model performance while minimizing misclassifications.

The final category is determined through a soft voting mechanism among these models. For every data point, all three models contribute to the prediction process, ensuring a robust and precise classification approach. This collaborative effort among XGBoost, LightGBM, and Random Forest collectively enhances the ensemble's predictive power and effectiveness.

Table 3.2 Cross-validation results of Sickness prediction using ensemble model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | **Average** | **Median** | **Mode** | **Minimum** | **Maximum** |
| Sensitivity | 0.73 | 0.724 | 0.693 | 0.537 | 0.92 |
| Specificity | 0.521 | 0.53 | 0.607 | 0.196 | 0.82 |
| Positive Predictive Value | 0.759 | 0.76 | 0.759 | 0.607 | 0.901 |
| Negative Predictive Value | 0.491 | 0.492 | 0.417 | 0.193 | 0.673 |

Table 3.2 illustrates the cross-validation results for health/sickness prediction using a custom ensemble model that combines Random Forest with XGBoost and LightGBM. The ensemble model outperformed the standalone Random Forest model, significantly enhancing overall performance and accuracy.

**FEED INTAKE:**

In the context of predicting milk intake, where exact values are challenging due to their continuous nature, a classification approach is adopted. The milk intake is categorized into three distinct classes to facilitate predictions. Subsequently, these three categories are accurately predicted using classifier models, with the Gradient Boosting Classifier emerging as the top performer, achieving an impressive accuracy rate of **82%**. GB excels in this three-class classification task by effectively identifying the boundaries between the categories, making it a robust choice for this specific prediction problem.

Table 3.3 illustrates the cross-validation results for Feed intake amount prediction using a custom ensemble model that combines Gradient Boosting with Linear Regression. This ensembled results outperformed the GB model.

Table 3.3 Results of Feed Intake prediction using ensemble model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric | **Mean** | **Median** | **Max** | **Min** |
| Sensitivity | 0.8202 | 0.8345 | 1 | 0.7457 |
| Specificity | 0.7014 | 0.7087 | 0.8465 | 0.4514 |
| Positive Predictive Value | 0.6071 | 0.6455 | 0.7838 | 0.4055 |
| Negative Predictive Value | 0.9382 | 0.9448 | 1 | 0.8178 |

**WEIGHT PREDICTION:**

In the project, animal weight prediction was accomplished using regression models within a pipeline. The initial ***K-Nearest Neighbour*** model achieved an R2 score of **0.83**. To address data imbalance, the dataset was up sampled, yielding a more balanced representation of healthy and sick animals. Upon reevaluation, the ***Random Forest model*** improved the R2 score to **0.85,** resulting in more accurate predictions and enhanced model performance.

**BEHAVIOUR CLASSIFICATION:**

When mapping feature sets to activities, a comparison of machine learning models revealed that Gradient Boosting (GB) outperformed Decision Trees, Logistic Regression, and Multilayer Perceptron (MLP). MLP faced challenges such as local optima and the vanishing gradient problem. GB excelled in effectively handling complex mapping tasks.

**GRADIENT BOOSTING:**

The Gradient Boosting (GB) model was chosen for predicting behavior from acceleration data due to its ability to effectively capture patterns in time-series data. In a scenario where nuances in behavior play a crucial role, GB stands out. It operates by sequentially improving the performance of decision trees, addressing the intricate dynamics of time-series data. Each decision tree corrects the errors of the preceding one, gradually enhancing predictive accuracy.

GB's proficiency in capturing these temporal patterns sets it apart. By continuously learning and adapting to the data's subtleties, it excels in recognizing changes and trends in animal behavior over time. This makes it a suitable choice for predicting behavior from acceleration data, where nuances and temporal dependencies are paramount for accurate insights.

Table 3.4 Cross-validation results of Lying prediction using Gradient Boosting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric | **Mean** | **Median** | **Max** | **Min** |
| Sensitivity | 0.7092 | 0.7402 | 0.9573 | 0.3196 |
| Specificity | 0.69 | 0.6726 | 0.9776 | 0.4979 |
| Positive Predictive Value | 0.6369 | 0.6491 | 0.9242 | 0.3296 |
| Negative Predictive Value | 0.7918 | 0.8176 | 0.9487 | 0.4359 |

**ENSEMBLING OF GRADIENT BOOSTING WITH RANDOM FOREST AND LOGISTIC REGRESSION:**

Gradient Boosting and Logistic Regression are powerful techniques known for their efficiency and accuracy in predictive modeling. These algorithms utilize boosting techniques and probabilistic modeling to enhance model performance and improve key metrics such as accuracy, precision, and recall.

By integrating Gradient Boosting and Logistic Regression into the ensemble alongside Random Forest, we can harness their strengths to elevate the overall performance of our model while reducing misclassifications. This collaborative effort creates a more reliable and robust classification approach.

The final category is determined through a soft voting mechanism among these models. For each data point, all three models contribute to the prediction process, ensuring a well-rounded and accurate classification approach.

Table 3.5 illustrates the cross-validation results for Lying/Standing prediction using a custom ensemble model that combines Random Forest and Gradient Boosting

Table 3.5 Cross-validation results of Lying prediction using ensemble model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric | **Mean** | **Median** | **Max** | **Min** |
| Sensitivity | 0.9211 | 0.9298 | 0.9917 | 0.8289 |
| Specificity | 0.9051 | 0.9074 | 0.9905 | 0.7934 |
| Positive Predictive Value | 0.9438 | 0.9431 | 0.9868 | 0.875 |
| Negative Predictive Value | 0.9101 | 0.9115 | 0.9795 | 0.788 |

**3.1.3 UNIT TEST**

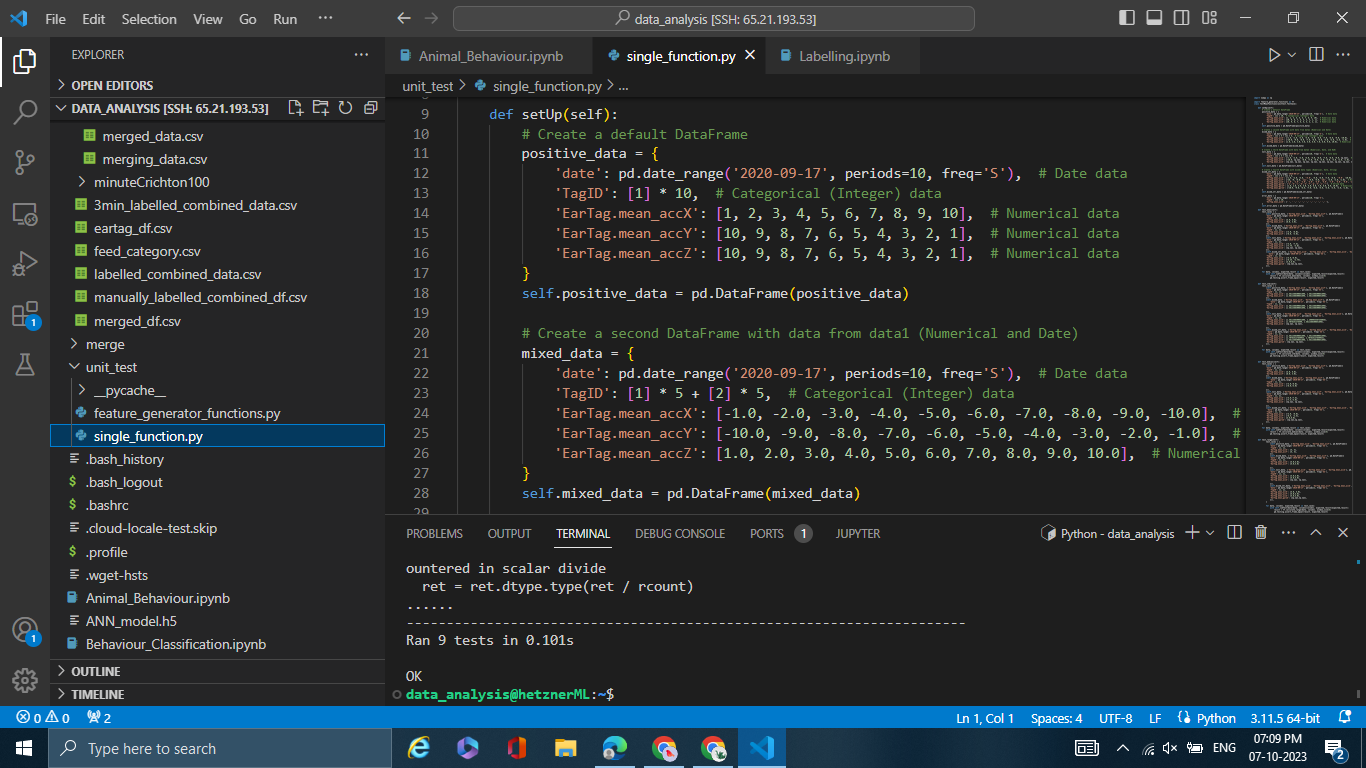
To streamline the feature generation process to make predictions after deployment of model, functions were created to generate the necessary features from the raw acceleration data. Additionally, a function was developed to standardize the raw data obtained from the gateway.

Several data preprocessing steps were implemented, including:

**1. Removing Outliers:** Outliers, accounting for 5% of the data, were filtered to ensure data integrity.

**2. Imputation of Missing Values:** A condition-based imputation or data point removal strategy was applied to address missing values.

**3. Data Filtering:** The dataset was filtered to include 70% of the data within a specified window. This step was crucial to ensure data stability and the extraction of reliable information from a substantial dataset.

The functions responsible for feature generation underwent rigorous testing across diverse scenarios to ensure stability and adaptability to varying conditions and datasets. This testing included unit testing, analysis of different scenarios, and subsequent modifications to ensure successful execution in all cases. Testing encompassed null cases, mixed data types, and exception errors. Additionally, challenging scenarios involving data outliers, null values, special characters, or strings were intentionally created, and the functions accurately identified the relevant data and successfully generated the required features. 

Screen 3.1 showcases the unit tests conducted and their status.

Screen 3.1 displays a sample of unit testing functions and their corresponding test results, with all tests successfully passing. This demonstrates the robustness and reliability of the code and its ability to consistently produce the expected results during unit testing.

**3.2 TOOLS DESCRIPTION**

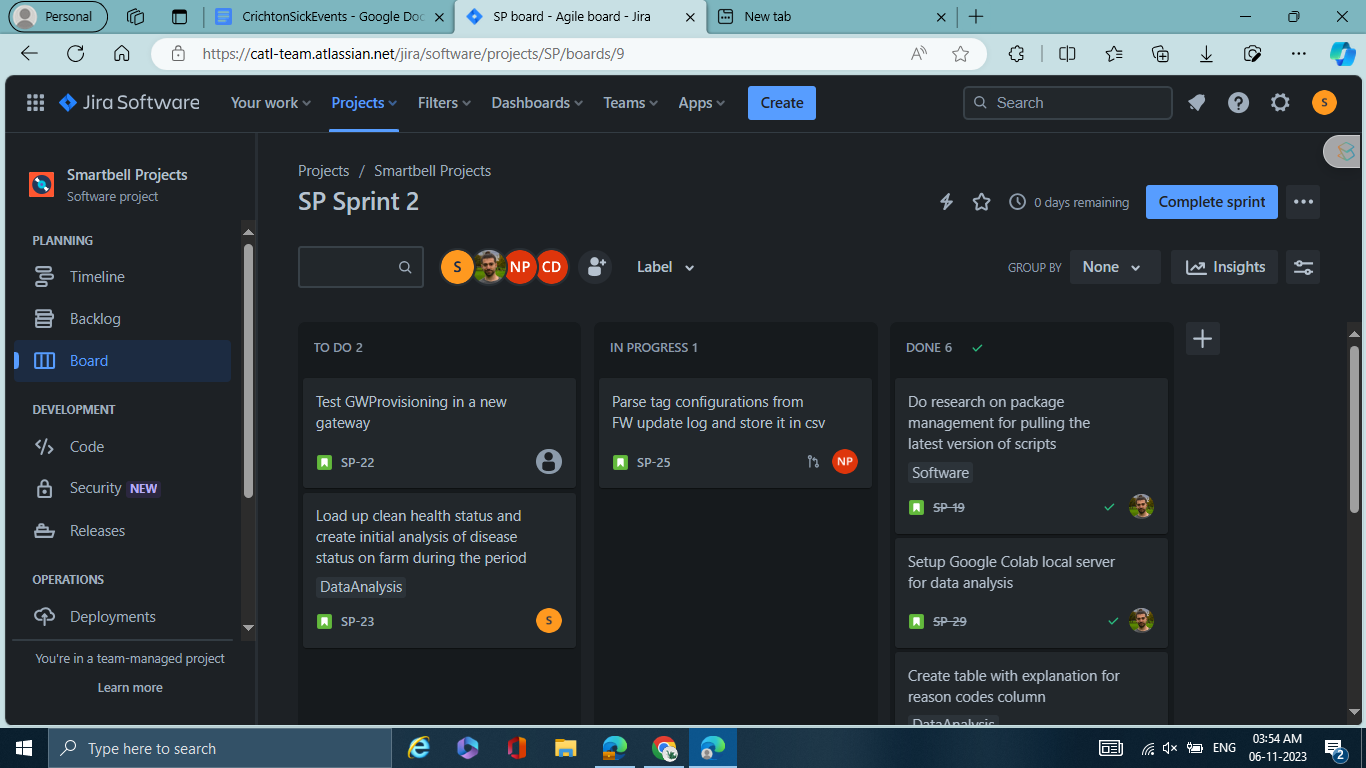
## *Python*

Python is an interpreter, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

***JIRA - PROJECT MANAGEMENT***

Jira is one of the best, most feature-complete project management tools out there, with native capabilities that can be enhanced by a variety of plugins. Jira was originally developed for development teams to use, it's now a full-on project management tool that can handle any kind of project; which means that project managers use Jira for a variety of reasons. Some of the reasons are

* Create and link issues to plan and communicate
* Monitor and report to manage teams
* Practice agile project management for flexibility
* Work with components to structure better
* Create and customize workflows for efficiency.



**Screen 3.1 Project management in Jira**

Screen 3.1 shows the Jira board and how different tickets are assigned for each project.

## *Visual Studio Code*

Visual Studio Code is a freeware source code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

***Bitbucket:***

Bitbucket is a platform designed to facilitate efficient collaboration and version control, particularly for software development teams working on source code projects. Its core objectives encompass enhancing productivity, ensuring data integrity, and providing seamless support for distributed and non-linear workflows, enabling the smooth management of thousands of concurrent branches across various systems.

All the above-mentioned tools have been extensively used for this project right from data collection to pre-processing and building a predictive model.

## *3.2.1 Libraries and Packages used*

## Pandas

Pandas is a high-level data manipulation tool. It is built on the NumPy package and its key data structure is called the Data Frame. Data Frames allow us to store and manipulate tabular data in rows of observations and columns of variables.

## NumPy

NumPy is a base n-dimensional array package in Python for scientific computing.

## Scikit-learn:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.

## Statsmodels:

As its name implies, statsmodels is a Python library built specifically for statistics. Statsmodels is built on top of NumPy, SciPy, and matplotlib, but it contains more advanced functions for statistical testing and modeling that you won't find in numerical libraries like NumPy or SciPy.

**Plotly:**

Plotly is a Python library specializing in creating interactive and visually appealing data visualizations, with a focus on web-based applications and data dashboards. It enables the generation of interactive charts and graphs, making it ideal for data exploration and sharing insights with features like zooming, panning, tooltips, and data point highlighting.

**Matplotlib:**

It offers a broad range of plotting functions and customization options, making it the preferred choice for generating non-interactive, 2D plots for research papers, presentations, and reports in the scientific and academic domains.

**3.3 PSEUDOCODE OF BEHAVIOUR CLASSIFICATION**

1. Import necessary libraries (NumPy, pandas, scikit-learn, imbalanced-learn).
2. Load your labeled data.
3. Define your features (X) and target (y). Remove unwanted columns and encode target labels.
4. Create a group identifier for each animal based on TagID.
5. Initialize cross-validation method (LeavePGroupsOut) with n\_groups=1.
6. Initialize lists to store evaluation metrics.
7. Initialize the SMOTE resampler and specify a random state.
8. Perform cross-validation using the LeavePGroupsOut method.
9. For each fold:

Split the data into training and testing sets.

Apply SMOTE resampling to the training data.

Standardize the data using StandardScaler.

Create a GradientBoostingClassifier model.

Fit the model on the resampled training data.

Make predictions on the test data and calculate evaluation metrics.

Store metrics and counts in respective lists.

1. Create a DataFrame to store metrics for all iterations.
2. Calculate mean, median, maximum, and minimum values for the metrics.
3. Print the aggregated metrics.

# CHAPTER IV

# ANALYSIS PROCESS EVALUATION

### 4.1 PERFORMANCE MEASURES

The model performances are measured using various error metrics for both classification and regression. For classification models, the performances are measured using ***Confusion Matrix, Accuracy, Sensitivity, specificity, Positive predictive value, negative predictive value.***

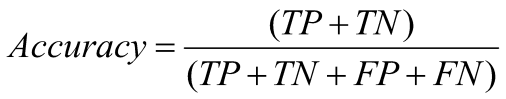
### Confusion Matrix

A confusion matrix is an N X N matrix, where N is the number of classes being predicted. For example, If N=2, and hence results in a 2 X 2 matrix. The elements of the matrices are the True Positive Rate, False Positive Rate, True Negative Rate and False Negative Rate. Those are used to calculate the accuracy, specificity, recall, f1 score.

### Accuracy

Accuracy is the proportion of true results among the total number of cases examined.

It is calculated as follows,



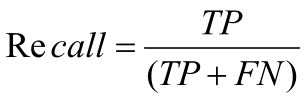
Tabel 4.1 Results of the model pipeline used for behavior classification.

|  |  |
| --- | --- |
| RandomForest Accuracy | 0.84 |
| GradientBoosting Accuracy | 0.7 |
| LogisticRegression Accuracy | 0.8 |
| SVM Accuracy | 0.84 |
| KNeighbours Accuracy | 0.76 |
| NaiveBayes Accuracy | 0.71 |
| DecissionTree Accuracy | 0.51 |
| AdaBoost Accuracy | 0.61 |
| Bagging Accuracy | 0.78 |
| ExtraTrees Accuracy | 0.8 |

From Tabel 4.1, it can be inferred that, for behavior classification, the random forest outperforms all 10 different models evaluated.

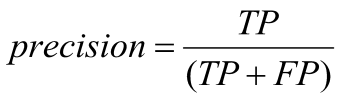
### Recall (SENSITIVITY)

Recall is also known as True Positive Rate or sensitivity ratio. It is calculated as the ratio of correctly predicted positive elements to total positive elements in the data.



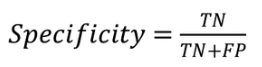
### Precision (POSITIVE PREDICITVE VALUE)

Precision is also known as specificity ratio. It is calculated as the ratio of correctly predicted positive elements to total predictive positive elements in the data.



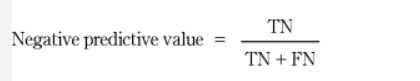
**Specificity:**

Specificity, also known as the true negative rate, is calculated as the ratio of correctly predicted negative elements to the total actual negative elements in the data. It measures how well a model can correctly identify true negative cases.



**Negative Predictive Value (NPV):**

Negative Predictive Value is calculated as the ratio of correctly predicted negative elements to the total predicted negative elements in the data. NPV assesses the probability that a predicted negative result is accurate and represents the model's ability to correctly identify true negatives.



Tabel 4.2 Results of the metrics for health/sick prediction using a random forest classifier for two datasets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **Sensitivity** | **Specificity** | **Positive Predictive Value** | **Negative Predictive Value** |
| **Accelerometer** | 0.7409 | 0.6343 | 0.7989 | 0.5552 |
| **Feeding** | 0.8086 | 0.7167 | 0.8430 | 0.5885 |

From Tabel 4.2, it is easy to infer that features from the feeding machine are crucial for improving the metrics of health/sickness classification.

Tabel 4.3 Results of the metrics for feed intake per day based on an ANN classifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **METRICS** | **Sensitivity** | **Specificity** | **Positive Predictive Value** | **Negative Predictive Value** |
| Feed Intake | 0.89 | 0.91 | 0.81 | 0.87 |

From Tabel 4.3, we can infer that predicting the exact total feed intake per day for each animal is a challenging task. Moreover, including either the exact feed intake amount or categorized values in sickness prediction produced similar results.

From Tabel 4.4, It is easy to infer that the feeding data is currently utilized solely for health/sickness classification. It is crucial to note that using predicted features like feeding amount and feeding time as additional features for health/sickness classification can significantly enhance the results and enrich the content.

Table 4.4 Metrics for health/sick prediction using a random forest classifier, incorporating predicted feeding amount as a feature alongside ear-tag features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **METRICS** | **Sensitivity** | **Specificity** | **Positive Predictive Value** | **Negative Predictive Value** |
| **Health/Sick** | 0.95 | 0.7 | 0.88 | 0.85 |

**4.2 CROSS VALIDATION**

As part of the model evaluation process, a grouped stratified k-fold validation technique is employed to assess the model's robustness. The grouped stratified k-fold method is particularly useful in scenarios where the data exhibits grouping, as it ensures that animals with similar characteristics are kept together during the training and testing phases.

In each fold of this process, the dataset is divided in such a way that it maintains the distribution of the animals' characteristics. This ensures that the model is tested on unseen data that closely resembles the overall dataset. Specifically, for each fold, five animals are set aside for testing, while the model is trained on the remaining 95 animals from the total dataset, which comprises approximately 100 animals in total. This approach allows for a comprehensive evaluation of the model's performance while accounting for the inherent groupings within the data, which can be especially valuable in situations where animal behavior is influenced by various factors such as age, breed, or environment.

Table 4.5 depicts the cross-validation results for predicting 'Ruminating' (yes/no) in cattle using a custom ensemble of random forest and extra tree classifier.

Table 4.5 Cross-validation results for each fold in prediction ruminating in cattle

### 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric | **Mean** | **Median** | **Max** | **Min** |
| Sensitivity | 0.7957 | 0.8711 | 0.8913 | 0.4078 |
| Specificity | 0.621 | 0.5306 | 0.9097 | 0.4205 |
| Positive Predictive Value | 0.6422 | 0.6231 | 0.8819 | 0.4526 |
| Negative Predictive Value | 0.8041 | 0.8046 | 0.9103 | 0.6491 |

Table 4.6 Cross-validation results for each fold in prediction lying or standing in cattle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Animal ID** | **Sensitivity** | **Specificity** | **PPV** | **NPV** | **Accuracy** | **Test Animal Count** | **LyingCount** | **Confusion Matrix** |
| 0 | 0.99 | 0.77 | 0.91 | 0.97 | 0.92 | 296 | 210 | [[67, 19], [2, 208]] |
| 1 | 0.89 | 0.96 | 0.97 | 0.83 | 0.91 | 247 | 157 | [[87, 3], [17, 140]] |
| 2 | 0.85 | 0.90 | 0.77 | 0.93 | 0.88 | 266 | 75 | [[172, 19], [11, 64]] |
| 3 | 0.91 | 0.71 | 0.84 | 0.83 | 0.84 | 276 | 174 | [[73, 29], [14, 160]] |
| 4 | 0.92 | 0.68 | 0.72 | 0.90 | 0.79 | 263 | 125 | [[94, 44], [10, 115]] |
| 5 | 0.85 | 0.81 | 0.91 | 0.68 | 0.85 | 281 | 201 | [[65, 15], [30, 171]] |
| 6 | 0.92 | 0.87 | 0.95 | 0.81 | 0.91 | 266 | 196 | [[61, 9], [14, 182]] |
| 7 | 0.82 | 0.60 | 0.63 | 0.82 | 0.72 | 271 | 122 | [[90, 59], [21, 101]] |
| 8 | 0.97 | 0.67 | 0.84 | 0.94 | 0.88 | 231 | 148 | [[56, 27], [3, 145]] |
| 9 | 0.85 | 0.90 | 0.77 | 0.93 | 0.88 | 266 | 75 | [[172, 19], [11, 64]] |

Table 4.6 displays a comprehensive set of cross-validation metrics (unseen animals), including sensitivity, specificity, positive predictive value, negative predictive value, and confusion matrix results.

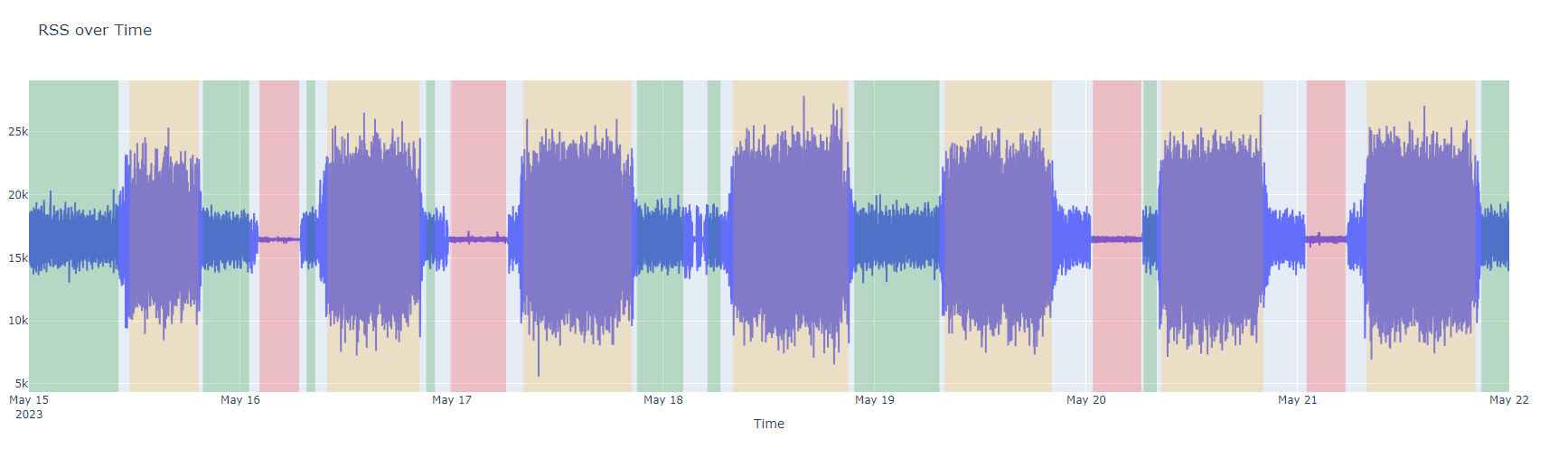
**CHAPTER V**

**ANALYSIS REPORTS AND INFERENCES**

### 5.1 REPORTS/VISUAL FORMATS

### 5.1.1 FAN-SATATE PREDICTION

### Based on the root sum of squares (RSS) values calculated for acceleration variables x, y, and z, a threshold-based classifier was employed to label the data according to plot intensity. This classification segregated the dataset into three distinct categories.



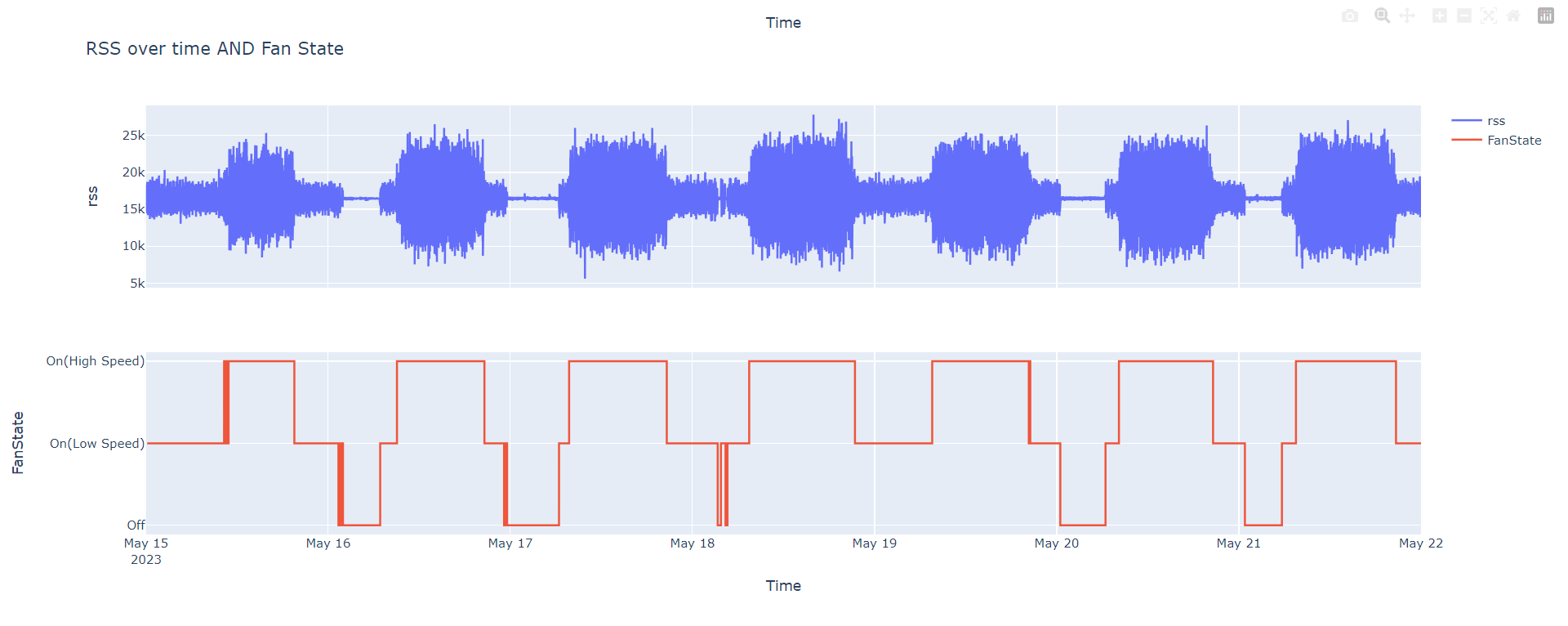
### Figure 5.1 A graph represents the fan state based on a threshold-based approach

### The data labeling is primarily dependent on the RSS (Root Sum of Square) values, which help categorize and understand the system's different states.

### Off\_time\_intervals: In this category, represented by the color red, data points exhibit RSS values less than 220. These intervals correspond to periods when the system is in an 'off' state.

* **Low\_speed\_time\_intervals:** Marked in green, these intervals have RSS values ranging from 220 to 1000. During these times, the system operates at low speeds.
* **High\_speed\_time\_intervals:** Represented in orange, this category includes data points with RSS values exceeding 1000. It signifies intervals when the system operates at high speeds.

The labeled data is used for training and tested with **10** different classifiers, among which the **Decision tree classifier** yielded the best results with accuracy of **0.93**.



### Figure 5.2 comparison of raw data and the predicted states using decision tree classifiers.

### Figure 5.2illustrates that the decision tree classifier model classified the fan states Accurately.

### 5.1.2 SICKNESS AND FEED INTAKE

### Figure 5.3 illustrates younger calves tend to consume more feed compared to older ones. As calf age increases, their feed intake stabilizes within a specific range, whereas younger calves exhibit a broader and more variable range of intake, spanning from the highest to the lowest levels. This observation highlights an age-dependent pattern in calf feed consumption.

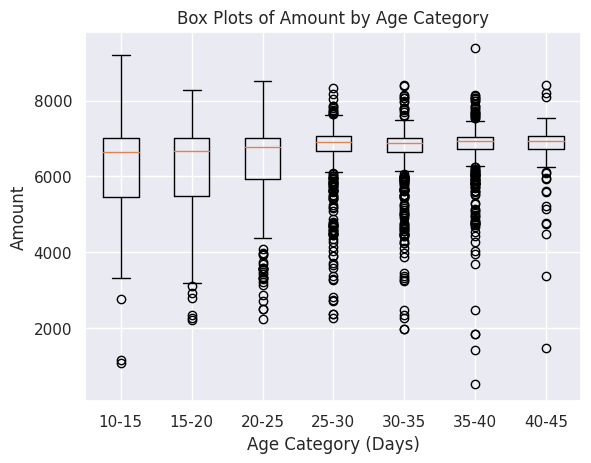


Fig 5.3 illustrates the variation in feed intake amount based on their age categories.

### 

### Figure 5.4 Health status of cattle over a period of 3 months

Figure 5.4 provides an overview of the overall health status of cattle, representing the count of both healthy and sick cattle. The green bars indicate the total count, the blue bars represent sick cattle, and the red bars depict healthy cattle. A crucial inference drawn from this visualization is that, over the specified period, the number of sick cattle surpasses that of healthy ones. This observation suggests that during this timeframe, cattle are more susceptible to illness, underlining a potential health concern.

### 5.1.3 DAY-WISE PROFILE

Utilizing the available data, a day-wise profile for the cows’ behaviors has been generated and integrated as a feature, while using the same data for training and labeling the entire month.

Figure 5.5 presents durations in minutes, revealing that some animals exhibit behavior for less time than a cow typically spends lying and ruminating in a day. This suggests the presence of estrus behavior in certain animals.

### 

### Fig 5.5 illustrates the total minutes of lying and ruminating for each animal on a day

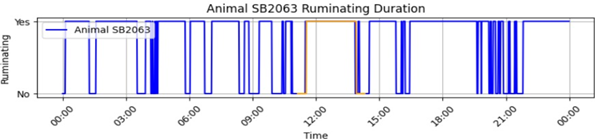
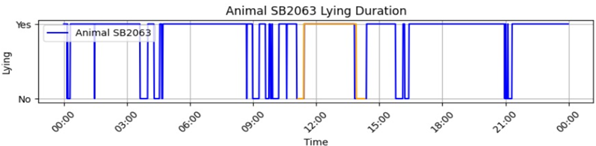


Fig 5.6 depicts lying and ruminating behaviors throughout the entire 24-hour period, with actual labels represented in yellow.

In Figure 5.6, the behavior prediction model performs exceptionally well, accurately predicting actual labels and consistently indicating extended lying times, which is vital for cow health. This alignment between predictions and expected behavior affirms the model's effectiveness, passing the test against logic.

**CHAPTER IV**

**CONCLUSION**

In this project to enhance milk yield and livestock management, a notable milestone has been the successful integration of predicted feeding time and amount categories. This innovation has shown immense potential in optimizing the feeding machine data. By streamlining the feeding process and ensuring the efficient utilization of resources, the project sets the stage for a more sustainable and efficient livestock management approach.

Another significant accomplishment is predictions related to animal behaviors. The system can now accurately predict behaviors like standing, lying, and feeding. These predictions are then thoughtfully aggregated into a day-wise profile. This holistic view of livestock behavior over time is a valuable asset for livestock management, providing insights into the well-being and daily routines of the animals.

Reducing misclassifications lead to various benefits, including better weight gain, lower mortality rates, and a positive impact on future milk yield, all while minimizing the need for antibiotics. By adopting a data-driven approach, the project aims to maximize resource efficiency in feeding, health management, and milk production. It provides a sustainable approach to livestock management that supports economic and environmental goals.

**BIBLIOGRAPHY**

**JOURNAL REFERENCE**:

1. Reza Arablouei, Lachlan Currie, Brano Kusy, Aaron Ingham, Paul L. Greenwood, and Greg Bishop-Hurley. “In-situ classification of cattle behavior using accelerometry data.” Computers and Electronics in Agriculture, 183, 2021.
2. Phung Cong Phi Khanh, Duc-Tan Tran, Van Tu Duong, Nguyen Hong Thinh, and Duc-Nghia Tran. “The new design of cows’ behavior classifier based on acceleration data and proposed feature set.” Mathematical Biosciences and Engineering, 17(4), 2020.
3. V. Schweinzer, E. Gusterer, P. Kanz, S. Krieger, D. Süss, L. Lidauer, A. Berger, F. Kickinger, M. Öhlschuster, W. Auer, M. Drillich, and M. Iwersen. “Evaluation of an ear-attached accelerometer for detecting estrus events in indoor housed dairy cows.” Theriogenology, 130, 2019.
4. J. A. Vázquez Diosdado, Z. E. Barker, H. R. Hodges, J. R. Amory, D. P. Croft, N. J. Bell, and E. A. Codling. “Classification of Behavior in Housed Dairy Cows Using an Accelerometer-based Activity Monitoring System.” Animal Biotelemetry, 3(1), 2015.

**WEBSITES**:

1. [scipy.signal.savgol\_filter — SciPy v1.11.3 Manual](https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html)
2. <https://www.analyticsvidhya.com/blog/2020/11/hyperparameter-tuning-using-optuna/>
3. <https://medium.com/analytics-vidhya/what-is-a-pipeline-in-machine-learning-how-to-create-one-bda91>
4. [unittest — Unit testing framework — Python 3.12.0 documentation](https://docs.python.org/3/library/unittest.html)
5. <https://plotly.com/python/>