Automated Classification of Animal Vocalization into Estrus and Non-estrus Phase

MAJOR PROJECT REPORT

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

BACHELOR OF TECHNOLOGY

(Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning)

Submitted By:

Anamika (R177219026)

Ananya Goel (R177219029)

Submitted To:

Ms Sugandha Sharma

(Assistant Professor)

University of Petroleum and Energy Studies

Energy Acres, Bidholi

May 2023

ABSTRACT

Communication is a fundamental aspect of animal behavior, and vocalizations play a critical role in conveying information, especially during the estrus phase when animals use vocalizations to attract mates. The automatic classification of animal vocalizations into estrus and non-estrus phases has significant implications for animal husbandry and wildlife conservation. In this project, we explore the use of machine learning and deep learning models to automatically classify animal vocalizations into estrus and non-estrus phases. We employ a dataset of animal vocalizations collected from different species and breeds to train and test the models.

The machine learning and deep learning models utilized in this project comprise Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), Random Forest, Naive Bayes, Recurrent Neural Networks (RNN), Convolutional Recurrent Neural Networks (CRNN), and ResNet34. These models were selected based on their superior performance in previous studies on animal vocalization classification and their ability to handle complex patterns in the data.

We initially preprocessed the raw audio data utilizing feature extraction techniques, including Mel Frequency Cepstral Coefficients (MFCC). The extracted features were then used as inputs to the classification models. We conducted a comparative analysis of the models, evaluating their performance based on several metrics, including accuracy, precision, recall, and F1-score. Our results demonstrate that though the deep learning models RNN and CRNN performed relatively well, the best accuracy was given by CNN, SVM and Random Forest. On the other hand, ResNet34 performed very poorly because of not being able to adapt to high complexity data.

To further evaluate the models' performance, we utilized various evaluation techniques, including confusion matrix, Receiver Operating Characteristic (ROC) curve, and Area Under the Curve (AUC) metrics.

The proposed automated classification system can be applied in animal husbandry and wildlife conservation to detect estrus phases automatically in various animal species. Our project provides a robust framework for the development of automated animal vocalization classification systems and showcases the potential of deep learning models in this field. Future research can focus on optimizing the deep learning models to improve their performance, as well as developing real-time animal vocalization classification systems.

ACKNOWLEDGEMENT

We, Anamika and Ananya Goel, would like to express our deepest gratitude to our esteemed mentor, Ms Sugandha Sharma (Assistant Professor) for her patient guidance, enthusiastic encouragement and useful critiques of this work. Her advice and constructive criticism have helped this research work take its present shape. Her insights, comments and suggestions throughout the course of the project opened up new avenues and constantly motivated us to work harder.

This work has been an endeavour which would not have been possible without the guidance from the Informatics Cluster, University of Petroleum and Energy Studies (UPES) and we offer our deepest gratitude that as a result of this major, this compelling work came into being. Last but not the least, we appreciate the panel of all faculty members for their time in our project presentation and for all the comments and insights they offer.

Anamika

Ananya Goel

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Introduction to Project

Analysing animal sounds paves the way to identifying the various phases of an animal's life cycle. This can be a challenging task due to the diverse nature of animal sounds, which can vary greatly between species, individuals, and even within individuals over time. While working with audio files, the data is first pre-processed to extract features that are indicative of emotions, such as pitch, intensity, and spectral features using MFCC (Mel-frequency cepstral coefficients). These features are then used as inputs to machine learning or deep learning models, which are trained on labelled audio data to recognize the context behind them.

In this project, we aim to classify animal sounds into estrus or non-estrus phases using machine learning and deep learning techniques, and providing a comparative analysis between different models to recognize the models that work best for classifying animal sounds.

Animal sound classification as estrus or non-estrus is an important area of research in the field of animal behaviour and welfare. Estrus refers to the period of sexual receptivity in animals, during which they are more likely to mate. Accurate classification of these sounds asestrus or non-estrus can provide valuable information for breeding and reproductive management.

Classification in animal sounds is a complex and ongoing area of research, which requires the use of sophisticated machine learning and deep learning techniques. The goal is to select and improve models that can accurately identify emotions in animal sounds, which can provide valuable insights into animal behaviour and welfare.

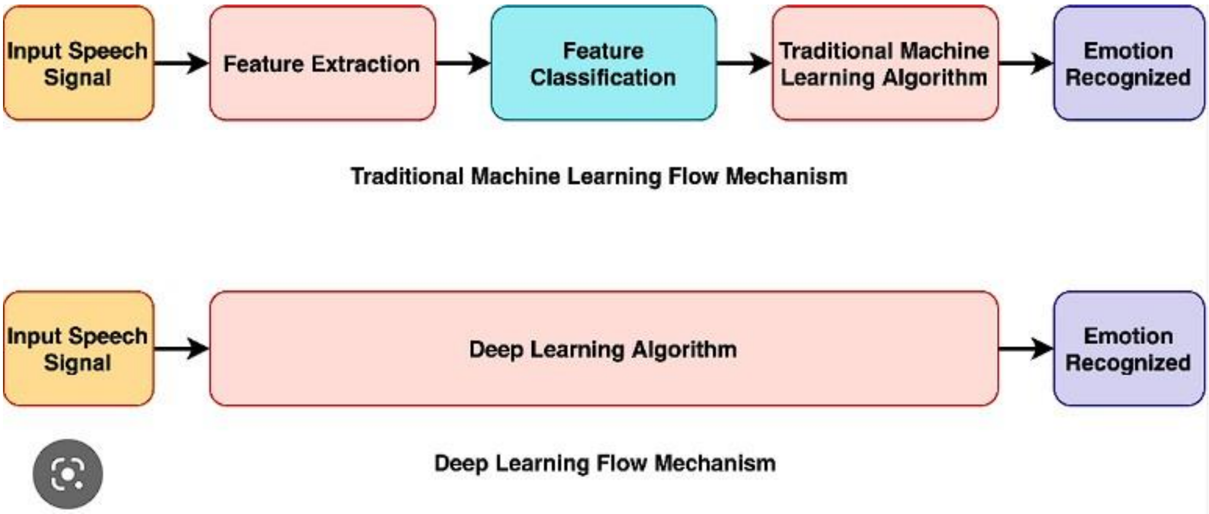

Figure 1.1 Machine Learning and Deep Learning Flow Mechanisms for Audio Classification

1.2 Project Category

The project is an industry automation-based project with the goal of this research is to create an automatic classification system that can tell the difference between animal vocalisations produced during estrus and those produced during other phases of the animal's life.

This approach is intended to be used for managing animals, especially dairy cows. An automated approach for categorising animal vocalisations into estrus and non-estrus phases would enable the management of livestock reproduction in a way that is both economical and effective. Farmers will be able to swiftly and correctly identify estrus thanks to this method, which will raise breeding success rates and enhance herd health.

1.3 Objective

Reproductive Efficiency is the main goal of this method is to increase animal reproduction efficiency. Farmers may ensure that they are breeding their cows at the best moment, resulting in greater pregnancy rates and increased milk output, by properly detecting estrus incows.

Using an automated method rather than manual observation and detection to identify cow estrus can help cut labor costs. Additionally, it eliminates the requirement for the farmer to be present at all times, saving money. It is useful in early estrus detection, the technology seeks to identify cow estrus early, enabling timely breeding and improved reproductive results.

Additionally, this can lessen the possibility of missed or delayed estrus detection, which can result in longer calving intervals and lower milk production. Apart from this, it creates a technique that can be used with a variety of animal species, including those with various vocalization styles and physical traits.

Last but not least, advance the study of signal processing and machine learning as they are applied to animal vocalizations.

1.4 Problem Formulation

It takes a lot of time and effort to manually separate animal vocalisations into estrus and non-estrus periods, and the process is susceptible to human mistake and variability. Animal reproductive management and health outcomes may be improved by automated estrus detection techniques, although these techniques are frequently inaccurate and not applicable to all animal species.

In order to improve the effectiveness and precision of estrus detection inanimals, our project's goal is to develop an automated system for classifying animal vocalisations into estrus and non-estrus phases that can be used to categorise a variety of animal species.

1.5 Identification of Need

The need behind a project classifying animal sounds into estrus and non-estrus phases is to improve reproductive management in livestock and potentially improve animal welfare.
During the estrus phase, also known as the heat phase, female animals are fertile and can become pregnant.

However, the timing and duration of this phase can be difficult to detect, particularly in large herds or flocks. By developing a system that can accurately classify animal sounds as either indicating estrus or non-estrus phases, farmers and animal managerscan more effectively time breeding and increase the likelihood of successful pregnancies.

Additionally, accurately detecting estrus phases can reduce the need for artificial insemination, which can be costly and time-consuming. By improving reproductive management, farmers can potentially increase the productivity and profitability of their herdsor flocks. Furthermore, detecting estrus phases can also have animal welfare benefits by reducing the need for prolonged confinement or separation of animals, which can be stressfuland detrimental to their health and well-being.

1.6 Existing System

There has been a slight dearth in the amount of research done on using audio data from animals to classify them as estrus or non-estrus, but resources are still available. Many of these studies have used machine learning and deep learning techniques to classify audio signals from various animal species, including buffalos, cows, pigs, and goats.

For example, some studies have used decision tree algorithms, such as random forests, to classify the audiodata into estrus or non-estrus categories. Other studies have used support vector machine (SVM) algorithms, which are particularly effective in handling high-dimensional and non- linear data.

In recent years, deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have become popular in the analysis of audiodata from animals. These techniques have shown promising results in terms of accuracy and have outperformed traditional machine learning algorithms in several cases.

Moreover, some studies have focused on extracting features from the audio data that are indicative of the estrus state, such as the frequency and amplitude of vocalizations. These features are then used as input to machine learning algorithms to classify the audio data.

In summary, classifying audio data from animals as estrus or non-estrus has a wide range of scope, and thefield continues to evolve as new techniques and algorithms are developed. Our aim is to apply a number of machine learning and deep learning models on the same dataset to closely study the difference in their results and judge their efficiency.

1.7 Proposed System

For this project, first a dataset of animal sounds was obtained, along with annotations of the phase of the animal during each sound clip. The acoustic properties of the animal sounds should be analysed to extract relevant features.

This can involve the measurement of parameters such as pitch, intensity, spectral content, and spectral shape. Inputting the extracted features from the audio files rather than using the raw audio signal directly results in greater efficiency and performance of the model, as there is a lot of noise present in the raw data.

Our goal is to extract features from the raw data and create a dense representation of the content. For this, we use MFCC wherein we create sliding windows of a pre- determined length to capture information such that the features inside this frame are relativelystationary, allowing us to work on them. The MFCC technique consists of 39 features. This includes 12 Cepstrum coefficients along with the energy term, and two more sets corresponding to the delta and double delta values. The steps in the MFCC technique are as follows:

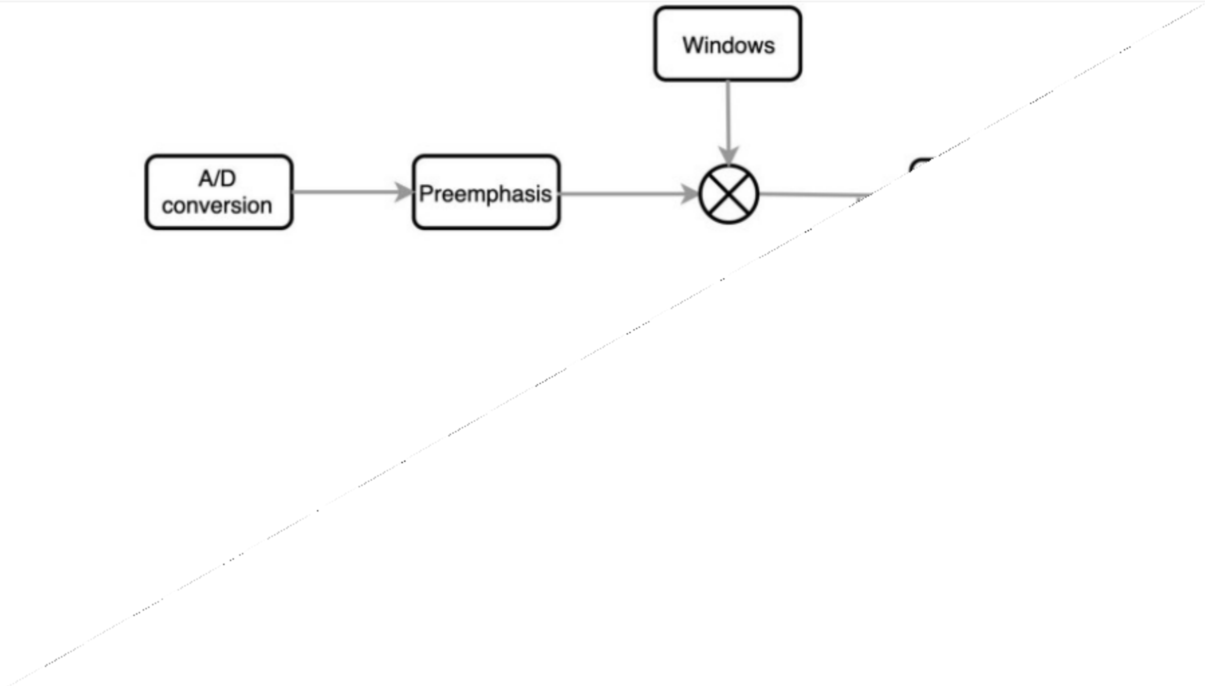

Figure 1.2 Step-by-step working of the MFCC technique for feature extraction

This technique includes windowing the signal, applying the DFT (Discrete Fourier Transform), followed by taking the log of the magnitude. After this, the frequencies are warped on a Mel scale and inverse DCT (Discrete Cosine Transform) is applied.

This dataset was then converted into a spectrogram. Machine learning models are to be selected for the classification of animal sounds based on their content into respective phases.We will be using approximately four machine learning classification models like a decision tree, support vector machine, random forest, KNN, naïve bayes, etc. and four deep neural network like an RNN, CNN, CRNN, and ResNet34.

The collection of animal noises should be used to trainthe models, which will then use the annotated labels as outputs and extracted features as inputs. The performance of all the models should be evaluated using a test dataset. This can involve the use of metrics such as accuracy, precision, recall, and F1 score. The suggested method includes data collection, feature extraction, model selection, training, assessment, refinement, and deployment for sound classification in animal species.

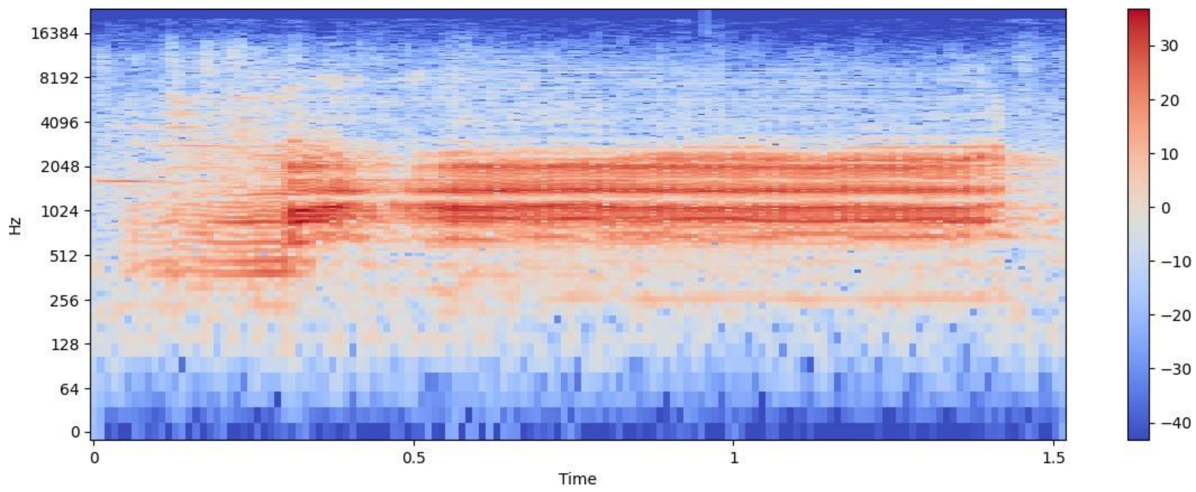

Figure 1.3 Visual representation of audio file including buffalo sounds as a spectrogram

1.8 Unique Features of the System

The major features and functions of this project include:

- Audio file input: Users can input audio files containing animal vocalization for classification.
- MFCC feature extraction: The tool extracts Mel-frequency cepstral coefficients (MFCC) features from the audio files, which are used as input to the machine learning and deep learning models.
- Model training and validation: The tool trains machine learning and deep learning models on the extracted MFCC features, and validates their performance using a validation dataset.
- Classification of animal vocalization: The trained models are used to classify animal vocalization into estrus and non-estrus phases.
- Output: The tool generates output files indicating the classification results for each input audio file.

Chapter 2 : Requirement Analysis and System Specification

2.1 Software Requirements

User Interface : The user input for this project comes in the form of the audio files collected for various animals in different phases of their reproductive cycles. The output presented is the target label (estrus/non-estrus) for each audio file.

Software Interface : The application will require python to be installed (any version > 3.6), necessary python modules installed and an active internet connection and could work seamlesslyon any ide like jupyter notebook, google collab etc.

Database Interface : The project uses .wav(audio) files as input for pre-processing the audio data and extracting valuable features. Afterwards, the extracted features are stored in a .xlsx file whichis used as input for the machine learning models.

Performance Requirements

We will implement algorithms that will read and change audio data and also create its image representation. So, having an 8 GB RAM is preferable. However, we have tested the same algorithms in 4 GB RAM system and found no issues in execution.

Security Requirements

There are no security requirements for this project. It doesn't require any security permissions from any system.

Software Quality Attributes

Adaptability: Adaptability influences how easy it is to change the system if therequirements have changed.

Availability: Availability is the ability of a service to be available and answer requests orrespond at all times. Our system should be available for use at any time.

Correctness: Correctness refers to the accuracy of the project and what are the chances of the prediction being correct, and the correct working of the model.

Flexibility: Flexibility enables system acceptance by allowing users to better understand the system and contributes to clear and consistent system documentation.

Interoperability: Interoperability is the ability of the service to communicate with other services properly.

Maintainability: This refers to the time and cost efficiency in maintaining a service to run. Our project is easily maintainable.

Portability: This can be measured in terms of costing issues related to porting, technicalissues related to porting, and behavioural issues related to porting.

Reliability: Reliability refers to the ability of the project to run correctly, defining whether we can rely on or trust the final result presented to us or not.

Reusability: Reusability is a chance of using a component or system in other components/systems with small or no change.

Robustness: Robustness is the adjudged ability of a software entity to behave according to the expectations of its stakeholders.

Testability: Testability matters when it comes to building and automating tests of individual components, interactions between components, as well as the system as a whole. In addition to that it is also crucial to know how well these tests can detect errors.

Usability: Every software-driven system is designed for ease of use to accomplish certain tasks. The attribute of usability denotes the ease with which users are able to execute tasks onthe system; it also indicates the kind of user support provided by the system.

Chapter 3 : System Design

3.1 Design Approach

Our project is based on the object oriented approach, that consists of classes and objects. Python is a computer language that emphasises objects.In Python, almost everything is an object with properties and functions. A class functions as a kind of "blueprint" or object constructor.

Based on the project scope and objectives, the following are the anticipated user classesfor this project:

- Researchers in animal behaviour and breeding: These users are expected to use the tool toanalyse and classify animal vocalization into estrus and non-estrus phases to aid in their research in animal behaviour and breeding.

- Veterinarians: Veterinarians may use this tool to help them detect when an animal is in the estrus phase, which can be helpful in monitoring animal health and reproduction.

- Livestock farmers: Livestock farmers can use this tool to improve their animal breedingprograms, by detecting the estrus phase and facilitating breeding at the optimal time.

- Animal behaviourists: Animal behaviourists may use this tool to study animal vocalizations and understand the communication patterns of different species duringdifferent phases of their reproductive cycle.

- Students: Students studying animal behaviour and related fields can use this tool for educational purposes, to learn about animal vocalizations and how they can be analysedand classified using machine learning and deep learning models.

3.2 Detail Design

The specific technologies and tools that will be used in this project include Python as the main programming language, along with libraries such as NumPy, SciPy, scikit-learn, and Keras. These libraries provide the necessary functions and tools for data processing, feature extraction, model development, and evaluation.

Parallel operations may be used to speed up the training process of the models. This can be achieved through the use of specialized hardware, such as GPUs or TPUs, or through the use of distributed computing frameworks, such as Apache Spark or TensorFlow.
Security considerations will need to be taken into account, particularly if the data being used in this project is sensitive. Measures such as encryption and access control may be necessary to ensure the security of the data.

Design conventions and programming standards will be important for ensuring the maintainability and scalability of the code. These conventions may include guidelines fornaming conventions, commenting, and documentation.

3.3 System Design using various Structured analysis and design tools such as: DFD's, Data Dictionary, Structured charts, Flowcharts.

Figure 3.1 Flowchart for the Project

3.4 User Interface Design

Dashboard: A summary of the system's performance will be shown on the dashboard of the user interface, including the quantity of vocalisations analysed, the quantity of vocalisations divided into estrus and non-estrus, and the system's overall accuracy rate.

Live Audio Feed: A live audio feed from the microphones in the barn or other locations where the cows are present is provided by the user interface. The user will be able to keep track of the vocalisations in real-time and look for any alterations or trends that might point to estrus.

Alerts for Estrus Detection: When a vocalisation is identified as being in estrus, the system will send out alerts. The user will always be informed of any changes in the cows' vocalisations because to the fact that these warnings can be delivered by email or mobile device.

3.5 Database Design

1. Activity Diagram

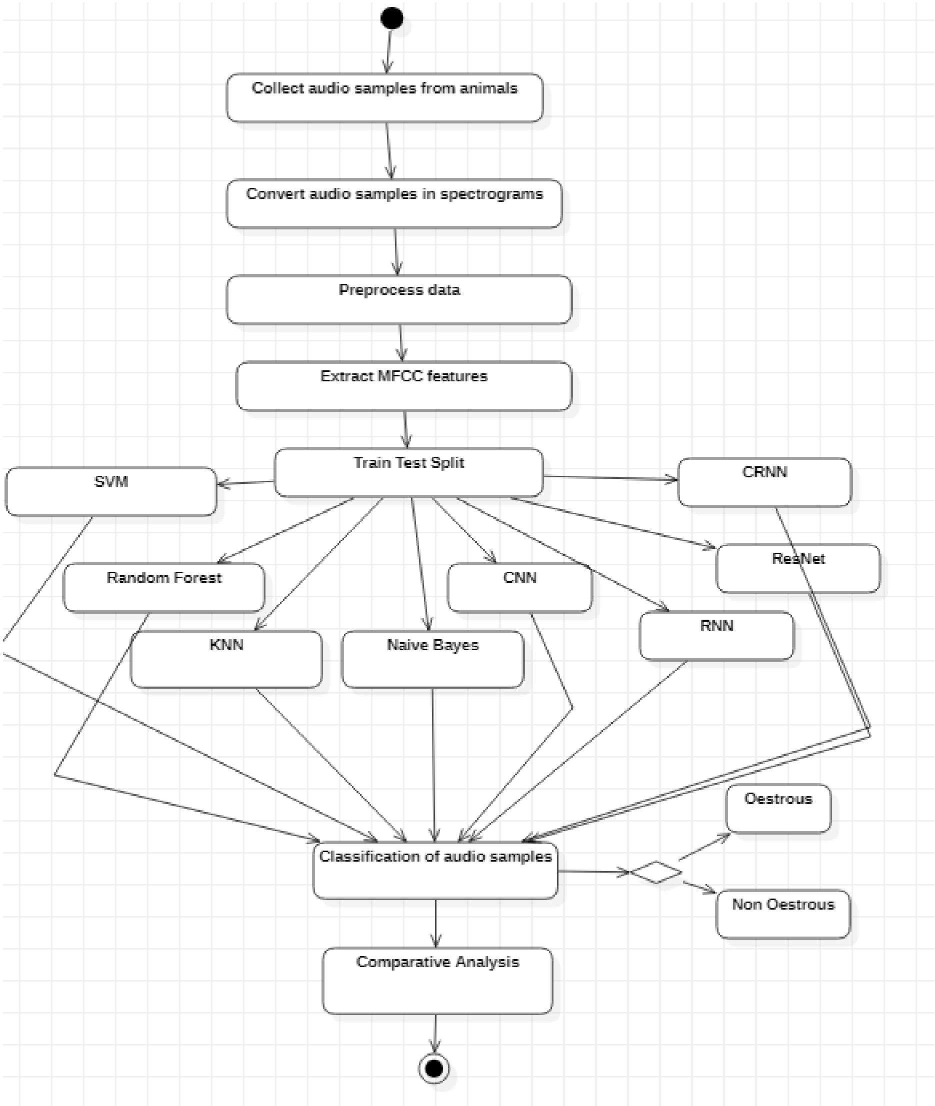

Figure 3.2 Activity Diagram

2. Sequence Diagram

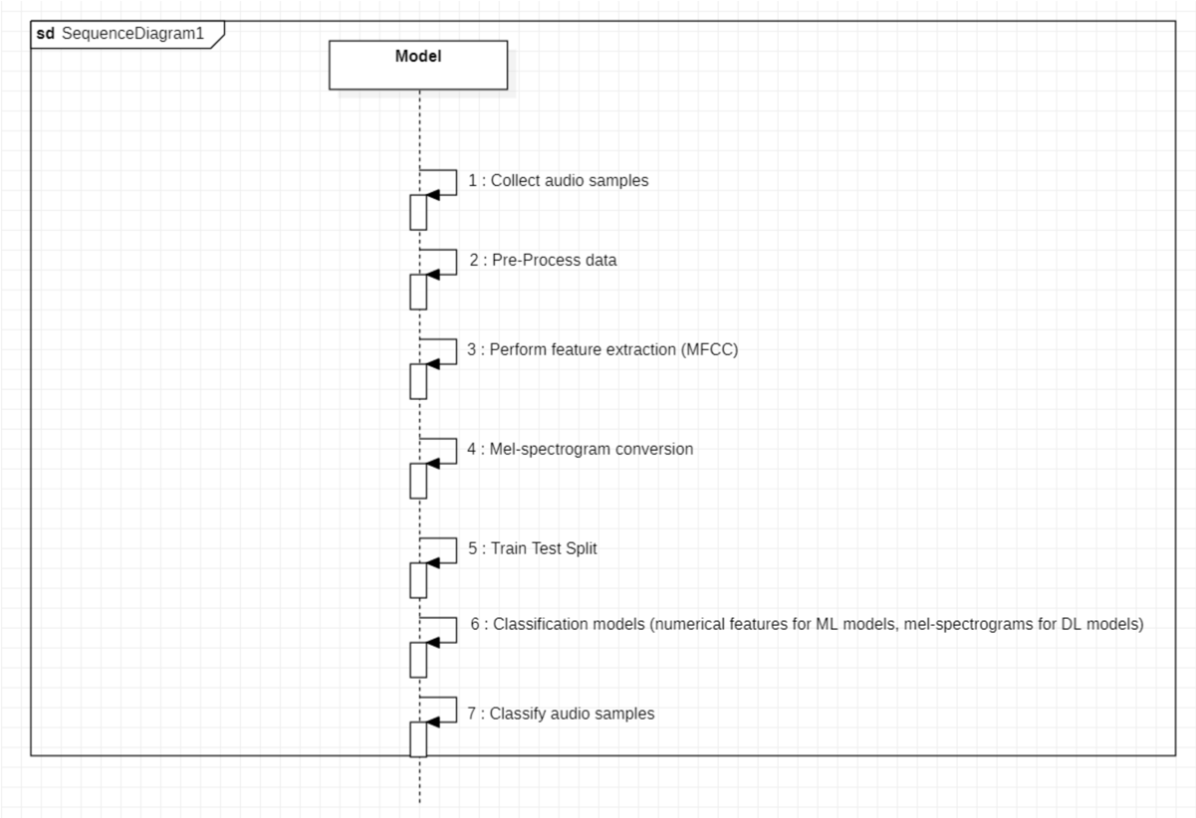

Figure 3.3 Sequence Diagram

3.6 Methodology

a. Data collection: Collect audio data of buffalos in heat and those that are not in heat. This data should be collected in a controlled environment to minimize any external factors that could affectthe audio recordings.

b. Feature extraction: Essential features from the raw audio data are extracted using MFCC. Thisstep consists of A/D conversion i.e., converting the audio signals from analog to digital format with a sampling frequency of 8kHz or 16kHz. Then pre-emphasis takes place which supplementsthe magnitude of energy in the higher frequencies.

$$x'[t_d] = x[t_d] - \alpha x[t_d - 1] \qquad 0.95 < \alpha < 0.99$$

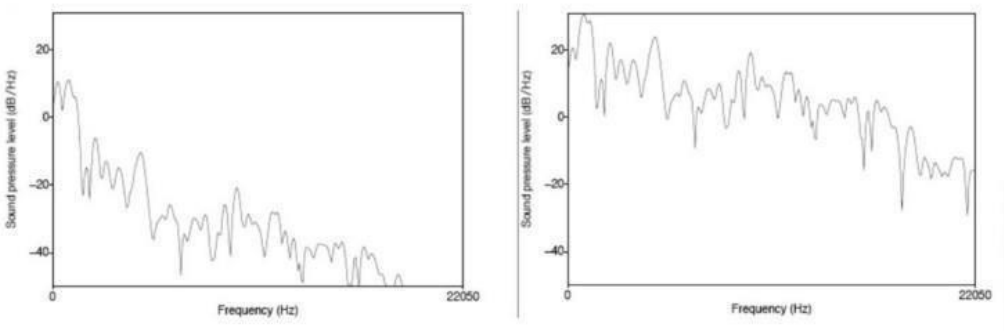

Figure 3.4 High-frequency signal before and after being boosted during pre-emphasis stage

Following this, slicing the audio waveform into sliding frames (windowing) takes place, succeeded by applying DFT. The next steps for MFCC include modelling human hearing property at the feature extraction stage by mapping Mel-Filter Bank. The formula for the same is $mel(f) = 1127 ln(1 + f/700)$.

c. Data pre-processing: Pre-process the audio data to make it suitable for use in machine learning models. This could include converting the audio data into a spectrogram representation, normalizing the data, and splitting the data into training and testing sets.

d. Model selection: Choose the machine learning models and neural network architectures to be used for the analysis. This could include decision trees, random forests, support vector machines,and feedforward neural networks, among others.

e. Model training: Train the selected models on the pre-processed audio data using the training set. This involves using the training data to adjust the model parameters such that the model canaccurately classify audio recordings as either being from a buffalo in heat or not in heat.

f. Model evaluation: Evaluate the performance of each model using the testing set. This couldinvolve measuring the accuracy, precision, recall, and F1-score of each model.

Chapter 4: Implementation, Testing and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

The project was implemented using Python as the primary programming language. Python is a popular language in the field of machine learning and deep learning due to its simplicity, ease of use, and extensive range of libraries and frameworks.

In this project, the Jupyter Notebook IDE was used for development and experimentation. Jupyter Notebook provides an interactive environment for writing and executing code, making it a suitable choice for machine learning projects that require rapid prototyping and experimentation.

Several in-built libraries and models were used in the project, including scikit-learn, Keras, and Tensorflow. scikit-learn is a powerful library for machine learning that provides a range of algorithms such as SVM, KNN, Random Forest, and Naive Bayes for classification tasks. Keras is a high-level neural networks API that is built on top of TensorFlow and provides an easy-to-use interface for building and training deep learning models such as CNN, RNN, and CRNN. Tensorflow is a popular deep learning framework that provides support for dynamic computation graphs, making it suitable for implementing complex architectures such as ResNet34.

Python is also used to visualize the performance of the different models and compare their accuracy, precision, recall, and F1 score.

The use of Python and the various libraries and frameworks made it possible to implement a robust and accurate automated classification system for animal vocalization into estrus and non-estrus phase.

4.2 Coding Standards of Language used

Python has a well-established set of coding standards known as PEP 8, which provides guidelines for writing readable and maintainable code. Following PEP 8 coding standards was essential to ensure consistency and clarity in the code.

Some of the coding standards that were followed include using descriptive variable and function names, maintaining proper indentation, avoiding excessively long lines of code, and commenting code sections that require explanation. These standards were followed to make the code more readable, understandable and maintainable.

4.3 Project Scheduling

Table 4.1: Project Timeline

S. No.	Tasks to be Performed	Targeted Completion Time	% Of work Completed
1.	• Collect and preprocess animal vocalization data • Label the data into estrus and non-estrus phase	January	25%

2.	• Analyze the data and extract relevant features • Select machine learning and deep learning models to classify the vocalization	February	50%
3.	• Train the models on the labeled data • Test the models on the unseen data and evaluate their performance	March	75%
4.	• Fine-tune the models for better performance	April	100%

4.4 Testing Techniques and Test Plans

All models were tested using validation data comprising 20% of the total dataset to evaluate their performance and accuracy. Going forward, the plan is to test these models using audio data from other animal species to assess their generalizability and to determine their applicability in a broader context. Such testing will be crucial in determining the robustness and effectiveness of the models and their ability to classify animal vocalizations with a high degree of accuracy.

Chapter 5: Results and Discussions

5.1 Brief Description of Various Modules of the Project

The following Python modules were used in the implementation of this project:

Pandas: Pandas is a Python library that provides high-performance, easy-to-use data structures and data analysis tools for easy manipulation and analysis of tabular data, including data cleaning, data exploration, and data visualization.

Numpy: Numpy is a Python library for working with arrays and numerical operations. It provides a powerful and efficient multi-dimensional array object, as well as tools for performing complex mathematical operations on these arrays.

Scikit-learn: Scikit-learn is a Python library for machine learning built on top of Numpy and SciPy. It provides tools for data preprocessing, feature selection, model selection, and model evaluation, as well as a range of machine learning algorithms for classification, regression, clustering, and dimensionality reduction.

Keras: Keras is a high-level neural network API written in Python. It provides an easy-to-use interface for building and training deep learning models, with support for a wide range of neural network architectures and optimization algorithms.

Tensorflow: Tensorflow is a powerful open-source library for numerical computation and large-scale machine learning. It provides a flexible platform for building and deploying machine learning models across a range of devices and platforms.

5.2 Snapshots of Database Tables with Brief Description

The dataset used in this project consists of a total of 235 audio files of cattle vocalizations, with 115 files labelled as non-estrus and 120 labelled as estrus. The audio files were recorded in a controlled environment using high-quality microphones to ensure good audio quality.

To extract useful features from the audio data, the Mel-Frequency Cepstral Coefficients (MFCCs) were used. MFCCs are a commonly used feature extraction method for audio data and provide a representation of the spectral envelope of the audio signal. In this project, the MFCCs were extracted using the librosa Python library, which is a popular package for analysing and processing audio signals.

The audio signals were first pre-processed to ensure that they all had the same length and sample rate (which was set to be 22,050 Hz). This was followed by resampling of the files using the 'kaiser_fast' method. Afterwards, we extracted 13 MFCC features for each audio file and used the mean of the transpose of the MFCCs matrix, which results in a Numpy array of 13 values and 1 target variable, as the input to the machine learning models.

	Coefficient 1	Coefficient2	Coefficient3	Coefficient4	Coefficient5	Coefficient6	Coefficient7	Coefficient8	Coefficient9	Coefficient10	Coefficient11	Coefficier
0	-273.050873	84.870125	-83.064804	11.823836	-58.650967	8.925778	-25.174921	13.277807	-14.353135	2.005484	-8.787545	7.413
1	-259.901154	98.847145	-87.239540	12.740252	-61.284405	2.679637	-30.025986	2.815952	-10.823910	-2.111652	-7.157680	12.266
2	-268.878662	81.200714	-90.944641	7.519948	-66.961296	5.050999	-26.816608	10.817904	-3.796612	1.286283	-6.812743	11.014
3	-307.030548	58.157440	-64.420914	27.427996	-34.898441	26.393442	-28.055254	5.019772	-23.750452	7.257837	-8.971681	6.523
4	-293.598419	70.562592	-56.458931	27.222097	-42.755630	28.099995	-22.653275	0.238418	-29.487917	6.432107	-11.107240	4.561
...	...	...	...	...	...	...	...	...	...	...	...	
230	-232.084930	41.194233	-77.909401	-2.140805	-50.534374	-0.607692	-44.580551	1.144995	-15.478563	10.937945	-12.198841	15.199
231	-232.428665	52.139889	-107.979271	5.868610	-63.695179	-12.369599	-46.991211	9.180314	-4.481325	17.243410	-9.082663	0.583
232	-197.029663	51.540428	-78.534447	2.666077	-58.547775	-1.444966	-39.257317	5.448378	-24.502949	12.243127	-3.864461	9.917
233	-225.854141	47.785435	-88.805153	3.629064	-50.521111	-0.199535	-38.784241	3.886981	-26.346024	3.102283	-10.442842	7.122
234	-284.266418	79.883354	-76.688538	0.031066	-55.652893	-19.982557	-49.005852	4.874951	-16.822191	11.389825	-8.453833	13.592

235 rows × 14 columns

Figure 5.1 Database used for training and testing the models

Chapter 6: Conclusion and Future Scope

Conclusion:

In conclusion, the goal of the project was to automate the classification of animal vocalizations into estrus and non-estrus phases. In order to achieve this goal, we used various machine learning and deep learning models such as SVM, KNN, Random Forest, Naive Bayes, CNN, ResNet34, RNN, and CRNN. We then performed a comparative analysis of the models to determine their accuracy.

Based on the results of the analysis, CNN, SVM, and Random Forest showed good performance with an accuracy of 0.8936, 0.8511, and 0.8298, respectively. On the other hand, ResNet34 showed poor performance with an accuracy of 0.4681. The reason for the good performance of CNN, SVM, and Random Forest could be attributed to their ability to capture complex patterns and relationships in the data. On the other hand, ResNet34, which is a deep residual network, showed poor performance, which could be attributed to its inability to capture the complex patterns and relationships in the data.

Also, CNN, SVM, and Random Forest are generally able to learn well from smaller amounts of training data. ResNet34, on the other hand, may require a much larger amount of training data than we had at hand to learn effectively.

Future Scope:

The current project can be further improved in several ways. First, more data can be collected from different sources to improve the accuracy of the models. The current dataset used for the project was limited to the vocalizations of buffaloes during estrus and non-estrus phases. Collecting vocalizations of other animals can help in developing more robust and accurate models for automated animal vocalization classification.

In extension to MFCC, there are several other techniques for feature extraction that can be explored in the future scope of this project. One such technique is Gammatone filterbank, which has been found to be effective for animal vocalization classification.

Additionally, the use of transfer learning can be explored to further improve the performance of the models. Transfer learning allows us to utilize pre-trained models that have been trained on larger and more diverse datasets, thereby avoiding the need for large amounts of data and computation resources.

Furthermore, the development of a real-time classification system can be explored to enable the automatic classification of animal vocalizations in the wild. This can be achieved by utilizing small, low-power embedded systems such as Raspberry Pi, which can be placed in the field to capture and classify animal vocalizations in real-time.

Another area that can be explored in the future is the use of ensemble methods. Ensemble methods combine the predictions of multiple models to improve overall performance. Techniques such as bagging, boosting, and stacking can be explored in the context of animal vocalization classification.

REFERENCES

1. Context-Dependent Domain Adversarial Neural Network for Multimodal Emotion Recognition by Zheng Lian, JianHua Tao, Bin Liu, Jian Huang, Zhanlei Yang, Rongjun Li (2020)

2. AST: Audio Spectrogram Transformer by Yuan Gong, Yu-An Chung, James Glass (2021)

3. Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings by Leonardo Pepino, Pablo Riera, Luciana Ferrer (2021)

4. Attention Based Fully Convolutional Network for Speech Emotion Recognition by Yuanyuan Zhang, Jun Du, Zi-Rui Wang, Jianshu Zhang (2018)

5. Evaluating Gammatone Frequency Cepstral Coefficients with Neural Networks for Emotion Recognition from Speech by Gabrielle K. Liu (2018)

6. Ensemble of convolutional neural networks to improve animal audio classification by Loris Nanni, Yandre M.G. Costa, Rafael L. Aguiar, Rafael B. Mangolin, Sheryl Brahnam, Carlos N. Silla Jr. (2020)

7. Vocalization patterns of dairy animals to detect animal state by Om Deshmukh, Nitendra Rajput, Yajuvendra Singh, Surender Lathwal (2013)

8. Automatic recognition of animal vocalizations using averaged MFCC and linear discriminant analysis by Chang-Hsing Lee, Chih-Hsun Chou, Chin-Chuan Han, Ren-Zhuang Huang (2006)

9. Animal Sound Classification Using a Convolutional Neural Network by Emre Sasmaz, F. Boray Tek (2018)

10. Animal Sounds Classification Scheme Based on Multi-Feature Network with Mixed Datasets by Kim Chung-Il, Cho Yongjang, Jung Seungwon, Rew Jehyeok, Hwang Eenjun (2020)

11. Convolutional Feature Vectors and Support Vector Machine for Animal Sound Classification by Kyungdeuk Ko, Sangwook Park, Hanseok Ko (2018)

12. Deep Learning-Based Cattle Vocal Classification Model and Real-Time Livestock Monitoring System with Noise Filtering by Dae-Hyun Jung, Na Yeon Kim, Sang Ho Moon, Changho Jhin, Hak-Jin Kim, Jung-Seok Yang, Hyoung Seok Kim, Taek Sung Lee, Ju Young Lee, Soo Hyun Park (2021)

13. Sow Estrus Diagnosis from Sound Samples Based on Improved Deep Learning by Peng Chen, Wengsheng Tang, Dan Yin, Bo Yang (2021)

14. Formant-based Acoustic Features for Cow's Estrus Detection in Audio Surveillance System by Jonguk Lee, Shangsu Zuo, Youngwha Chung, Daihee Park, Hong-Hee Chang, Suk Kim (2014)

15. Implication of Season and Parity on Bioacoustics Features for Detection of Estrus Phase in Murrah Buffalo by Devi Indu, Singh Pawan, Dudi Kuldeep, Singh Yajuvendra (2021)

16. Vocal Cues Based Decision Support System for Estrus Detection in Water Buffaloes by Indu Devi, Pawan Singh, Kuldeep Dudi, Surender S. Lathwal, Anand P. Ruhil, Yajuvendra Singh, R. Malhotra, Rubina K. Baithalu, Ranjana Sinha (2019)

17. Discriminate Animal Sounds Using TESPAR Analysis by G.P. Pop (2017)

18. Better Than MFCC Audio Classification Features by Ruben Gonzalez (2012)

19. Data Augmentation Approaches for Improving Animal Audio Classification by Loris Nanni, Gianluca Maguolo, Michelangelo Paci (2020)

20. Development of a method for automated detection of estrus in dairy cattle using data from an electronic tag by G. C. Schoenbaum, M. M. Liang, and C. L. Honaker (2007)

21. Sound Analysis for Cattle Disease Diagnosis by F. J. Santos, A. D. Chaves, E. S. S. Vieira, D. R. A. Moraes, and C. M. Carvalho (2015)

22. Prediction of Estrus in Dairy Cattle Using Machine Learning Algorithms by J. H. Park, J. W. Kim, J. S. Yoon, S. Y. Park, and J. H. Kim (2019)

23. Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach by Bergstra et al. (2019)

24. Automatic bird species recognition using ensemble learning by Basu et al. (2020)

25. Deep neural networks for bird song classification in audio recordings by Stowell et al. (2016)

26. Automatic classification of mammal vocalizations using acoustic features and machine learning by Satish et al. (2019)

27. Classification of animal vocalizations using machine learning algorithms by Alghamdi et al. (2020)

28. Deep learning for bird song classification and inference by Stowell et al. (2018)

29. Acoustic classification of bat echolocation calls using machine learning techniques by Jain et al. (2016)

30. Automatic recognition of animal vocalizations: a review by Zhang et al. (2020)

31. Machine learning approaches for animal sound classification: a comparative analysis by Debnath et al. (2020)

Plagiarism Report

100%

Plagiarism: **8%** | Unique: **92%**

Rewrite Content to Make it Unique

Extra Word Count Statistics

Syllables	1726	Average Sentence Length (word)	3.8
Sentences	234	Syllables Per Word(s)	2
Unique Word(s)	387 (44%)	Paragraph(s)	176
Average Word Length (characters)	5.4	Difficult Word(s)	410 (46%)
Reading Time	5 mins	Speak Time	8 mins

100%

Plagiarism: **1%** | Unique: **99%**

Rewrite Content to Make it Unique

Extra Word Count Statistics

Syllables	2012	Average Sentence Length (word)	7.4
Sentences	137	Syllables Per Word(s)	2
Unique Word(s)	389 (38%)	Paragraph(s)	91
Average Word Length (characters)	5.5	Difficult Word(s)	456 (45%)
Reading Time	6 mins	Speak Time	9 mins

100%

Plagiarism: **5%** | Unique: **95%**

Rewrite Content to Make it Unique

Extra Word Count Statistics

Syllables	1940	Average Sentence Length (word)	5.8
Sentences	174	Syllables Per Word(s)	1.9
Unique Word(s)	414 (41%)	Paragraph(s)	98
Average Word Length (characters)	5.3	Difficult Word(s)	459 (45%)
Reading Time	6 mins	Speak Time	9 mins

100%

Plagiarism: **11%** | Unique: **89%**

Rewrite Content to Make it Unique

Extra Word Count Statistics

Syllables	1889	Average Sentence Length (word)	5.5
Sentences	185	Syllables Per Word(s)	1.9
Unique Word(s)	438 (43%)	Paragraph(s)	101
Average Word Length (characters)	5.3	Difficult Word(s)	452 (45%)
Reading Time	6 mins	Speak Time	9 mins

100%

Plagiarism: **7%** | Unique: **93%**

Rewrite Content to Make it Unique

Extra Word Count Statistics

Syllables	1962	Average Sentence Length (word)	5.9
Sentences	174	Syllables Per Word(s)	1.9
Unique Word(s)	383 (37%)	Paragraph(s)	110
Average Word Length (characters)	5.3	Difficult Word(s)	472 (46%)
Reading Time	6 mins	Speak Time	9 mins

100%

Plagiarism: **12%** | Unique: **88%**

Rewrite Content to Make it Unique

Extra Word Count Statistics

Syllables	1895	Average Sentence Length (word)	5
Sentences	202	Syllables Per Word(s)	1.9
Unique Word(s)	478 (47%)	Paragraph(s)	97
Average Word Length (characters)	5.4	Difficult Word(s)	569 (56%)
Reading Time	6 mins	Speak Time	9 mins