BEYOND THE VEIL OF WELLNESS : MACHINE LEARNING'S UNIQUE JOURNEY IN ANIMAL HEALTH CLASSIFICATION

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(CSE)

Submitted By

SADALA RACHANA (21UK1AO5G9)

LAKKARSU SAI CHARAN (21UK1A05H4)

GORREMUCHU SIDDARDHA (21UK1A05J1)

THUNAGAR PUSHPALATHA (21UK1AO5E1)

Under the guidance of

Mr. G. Satish Chander

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(CSE)
VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION

INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled **"BEYOND THE VEIL OF WELLNESS: MACHINE LEARNING'S UNIQUE JOURNEY IN ANIMAL HEALTH CLASSIFICATION", is being submitted by SADALA RACHANA (21UK1A05G9),LAKKARSU SAI CHARAN (21UK1A05H4) ,GORREMUCHU SIDDARDHA (21UK1A05J1),THUNAGAR PUSHPALATHA(21UK1A05E1)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025.

Project Guide	**HOD**
Mr G. Satish Chander	**Dr.R. Naveen kumar**
(Assistant Professor)	(professor**)**

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHMED,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr .R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **Dr .G. SATISH CHANDER** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

SADALA RACHANA **(21UK1AO5G9)**

LAKKARSU SAI CHARAN **(21UK1A05H4)**

GORREMUCHU SIDDARDHA **(21UK1A05J1)**

THUNAGAR PUSHPALATHA **(21UK1A05E1)**

ABSTRACT

Machine learning (ML) has transformed animal health classification through a dynamic and evolving journey, significantly enhancing diagnostic accuracy and treatment precision. The journey began with the collection and preprocessing of veterinary data, where early ML applications utilized basic algorithms to address disease classification challenges. Progress was marked by advancements in imaging technologies, where ML models like convolutional neural networks (CNNs) revolutionized the analysis of medical images, enabling automated detection of anomalies such as infections.

With the integration of genomic data, ML facilitated personalized medicine, allowing for the identification of genetic markers linked to specific diseases and tailoring treatments to individual animals. The advent of wearable technologies and real-time monitoring further advanced the field, enabling continuous health tracking and early detection of conditions through data from sensors and IoT devices.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1.OVERVIEW

Machine learning (ML) has undergone a transformative journey in the realm of animal health classification, fundamentally altering how veterinarians diagnose, treat, and manage diseases. Here's an overview of this unique journey. The journey began with the accumulation of diverse veterinary data, including patient records, lab results, and imaging studies. Early efforts faced challenges related to data quality and quantity. Initial ML applications used simple algorithms to classify diseases based on historical data. These early models laid the groundwork but had limited accuracy and generalizability.

The development of advanced imaging technologies, such as X-rays and MRIs, allowed ML algorithms to excel in analysing visual data. Convolutional neural networks (CNNs) became instrumental in detecting and classifying abnormalities in medical images.ML models improved the accuracy of disease detection, enabling automated identification of conditions like fractures, and provided tools for behavioural analysis to assess health. The integration of genomic data into ML models facilitated the identification of genetic markers associated with various diseases. This enabled more precise diagnostics and personalized treatment plans based on an animal's genetic profile.ML helped in understanding breed-specific health issues and tailoring interventions to mitigate genetic predispositions to certain diseases.

1.2. PURPOSE

The purpose of machine learning's unique journey in animal health classification is multi-faceted, aiming to enhance various aspects of veterinary medicine and animal care. Here are the key purposes driving this journey:

1. Improved Diagnostic Accuracy

- **Enhanced Detection:** ML algorithms improve the accuracy of diagnosing diseases by analysing complex patterns in medical images, lab results, and other data sources.
- **Early Diagnosis:** Early and accurate detection of diseases helps in initiating timely treatments, potentially improving outcomes and reducing the progression of diseases.

2. Personalized and Targeted Treatments

- **Tailored Interventions:** By integrating genomic data and individual health records, ML models facilitate personalized treatment plans that cater to the specific needs of each animal.
- **Breed-Specific Care:** ML helps identify genetic predispositions to diseases specific to certain breeds, allowing for preventive measures and targeted therapies.

3. Efficient Health Monitoring

- **Real-Time Tracking:** Wearable technologies and continuous monitoring systems powered by ML provide real-time insights into an animal's health status, enabling proactive management of chronic conditions and early intervention for emerging issues.
- **Automated Alerts:** ML systems can generate alerts for veterinarians when abnormal patterns or potential health risks are detected, facilitating prompt response and management.

4. Enhanced Research and Insights

- **Disease Patterns:** ML analyses large datasets to uncover patterns and trends in animal health.
- **Predictive Analytics:** ML models predict disease outbreaks and health trends, aiding in preventive strategies and resource allocation for better disease man.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

The significant advancements machine learning (ML) has brought to animal health classification, several existing problems and challenges persist. Addressing these issues is crucial for further progress and effective implementation. Here's a detailed look at the key problems:

1. Data Quality and Availability

- **Incomplete or Inaccurate Data:** Many ML models depend on high-quality, accurate data. Incomplete or inaccurate veterinary records, imaging data, and other health information can lead to unreliable model performance.
- **Data Standardization:** The lack of standardized formats and terminology across different veterinary practices and research institutions complicates data integration and hampers the development of universal models.

2. Data Privacy and Security

- **Sensitive Information:** Veterinary data often contains sensitive information about animals and their owners. Ensuring the privacy and security of this data while using it for ML purposes is a major concern.
- **Regulatory Compliance:** Navigating varying regulations regarding data protection and privacy across regions adds complexity to data management and model development.

2.2 PROPOSED SOLLUTION

The application of machine learning (ML) to animal health classification, several proposed solutions can help enhance the effectiveness and reliability of these technologies. These solutions aim to improve data quality, address ethical concerns, enhance model performance, and ensure practical integration.

1. Improving Data Quality and Availability

- **Data Standardization:** Establishing standardized formats and terminologies for veterinary data can facilitate better integration and interoperability.

- **Enhanced Data Collection:** Investing in better data collection methods and technologies, such as high-resolution imaging and comprehensive electronic health records, can improve data accuracy and completeness.
- **Data Augmentation:** Using data augmentation techniques to generate synthetic data or expand existing datasets can help overcome limitations related to data scarcity.

2. Addressing Data Privacy and Security

- **Data Encryption and Anonymization:** Implementing robust encryption methods and anonymizing sensitive data can protect privacy while allowing for useful analysis. Ensuring that data handling practices comply with regulations like GDPR and HIPAA is essential.
- **Access Controls:** Setting up strict access controls and audit trails can prevent unauthorized access to sensitive health data and ensure that only authorized personnel can handle and analyse the data.

3. Mitigating Bias and Ensuring Fairness

- **Diverse Datasets:** Creating and using diverse and representative datasets that cover a wide range of breeds, conditions, and demographics can help reduce bias in ML models.
- **Bias Detection and Correction:** Implementing methods for detecting and correcting bias within ML models can ensure fairer outcomes. Techniques such as adversarial debiasing and fairness-aware algorithms can be used.
- **Continuous Monitoring:** Regularly evaluating models for fairness and performance across different subsets.

4. Enhancing Model Generalization and Avoiding Overfitting

- **Cross-Validation:** Utilizing techniques like cross-validation and splitting data into training, validation, and test sets can help ensure that models generalize well to new, unseen data.
- **Regularization Techniques:** Applying regularization methods can help prevent overfitting by penalizing overly complex models and promoting simpler, more generalizable solutions.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM

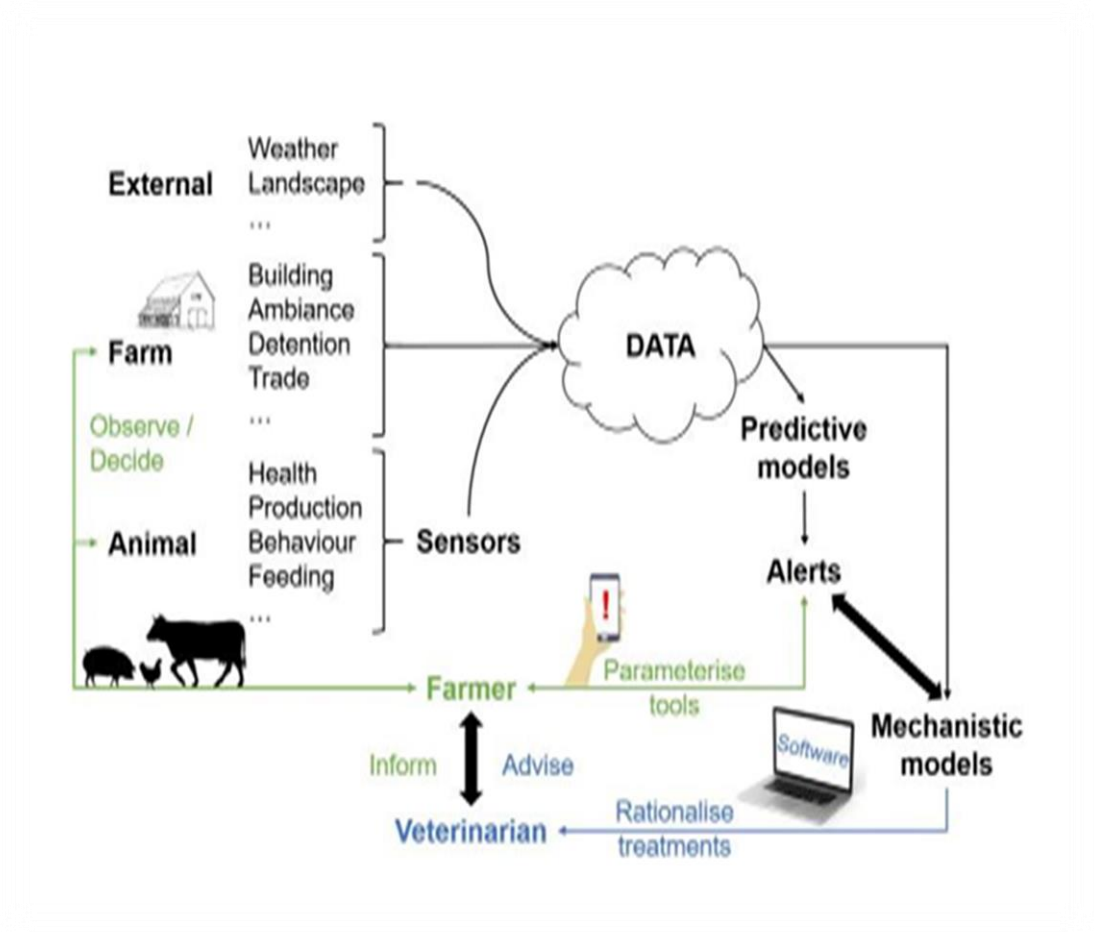


Figure: Beyond the veil of wellness: Machine Learning's Unique Journey In Animal Health Classification

3.2.SOFTWARE DESIGNING

Designing software for machine learning (ML) in animal health classification involves a multi-faceted approach that includes system architecture, data management, model development, user interfaces, and integration. The goal is to create a robust, scalable, and user-friendly system that enhances veterinary practices and animal health management. Here's a structured approach to designing such software:

1. System Architecture

1.1. Modular Design:

- **Component-Based Architecture:** Design the software with modular components such as data ingestion, preprocessing, model training, inference, and reporting. This allows for easy updates and maintenance.
- **Microservices:** Use microservices to manage different functionalities, such as data processing, model serving, and user management, enabling scalability and independent deployment.

1.2. Scalability:

- **Cloud Infrastructure:** Leverage cloud platforms (e.g., AWS, Azure, Google Cloud) for scalable computing resources, storage, and deployment. This supports handling large datasets and complex models.
- **Load Balancing:** Implement load balancing to manage varying demands and ensure high availability and performance.

2. Data Management

2.1. Data Integration:

- **Unified Data Platform:** Create a unified data platform that integrates data from various sources, such as electronic health records (EHRs), imaging systems, and wearable devices.
- **APIs and Data Connectors:** Develop APIs and connectors for seamless data integration and interoperability with existing veterinary systems.

2.2. Data Quality and Preprocessing:

- **Data Cleaning:** Implement automated tools for data cleaning, normalization, and enrichment to ensure high-quality input for ML models.
- **Feature Engineering:** Include tools for feature extraction and engineering tailored to veterinary data, such as imaging features or genetic markers.

3. Model Development

3.1. Model Training and Evaluation:

- **Frameworks and Libraries:** Utilize ML frameworks and libraries (e.g., Tensor Flow, Scikit-Learn) for model development. Incorporate tools for hyperparameter tuning and cross-validation.
- **Version Control:** Use version control systems for managing different versions of models and experiments, ensuring reproducibility and tracking progress.

4.EXPERIMENTAL INVESTIGATION

Experimental investigation of machine learning's (ML) journey in animal health classification involves systematically testing and evaluating various ML approaches and technologies to understand their effectiveness, limitations, and potential improvements. This process typically includes designing and conducting experiments to assess model performance, data handling, integration, and real-world applicability. Here's a detailed outline of how such an investigation might be structured:

1. Objective and Scope Definition

1.1. Define Goals:

- **Performance Metrics:** Determine the performance metrics to be investigated, such as accuracy, precision, recall, F1-score, or AUC-ROC, depending on the specific classification tasks (e.g., disease detection, behaviour analysis).
- **Use Cases:** Identify the specific use cases for the ML models in animal health, such as diagnosing specific diseases, predicting health outcomes, or analysing medical images.

1.2. Scope of Investigation:

- **Data Types:** Decide on the types of data to be used, including medical imaging, genomic data, electronic health records, or sensor data.
- **Model Types:** Specify the ML models to be investigated, such as convolutional neural networks (CNNs) for image classification, random forests for tabular data, or recurrent neural networks (RNNs) for time.

2. Experimental Design

2.1. Data Preparation:

- **Dataset Collection:** Gather datasets from diverse sources, ensuring they are representative of the problem domain. This may include veterinary records, imaging data, and wearable sensor data.
- **Data Preprocessing:** Preprocess the data to handle missing values, normalize features, and perform feature extraction or augmentation as necessary.

2.2. Model Selection:

- **Baseline Models:** Start with baseline models to establish a performance benchmark. These could include simple algorithms like logistic regression or decision trees.
- **Advanced Models:** Investigate advanced ML models such as deep learning architectures (CNNs, RNNs) or ensemble methods to explore their effectiveness in classification tasks.

2.3. Experiment Setup:

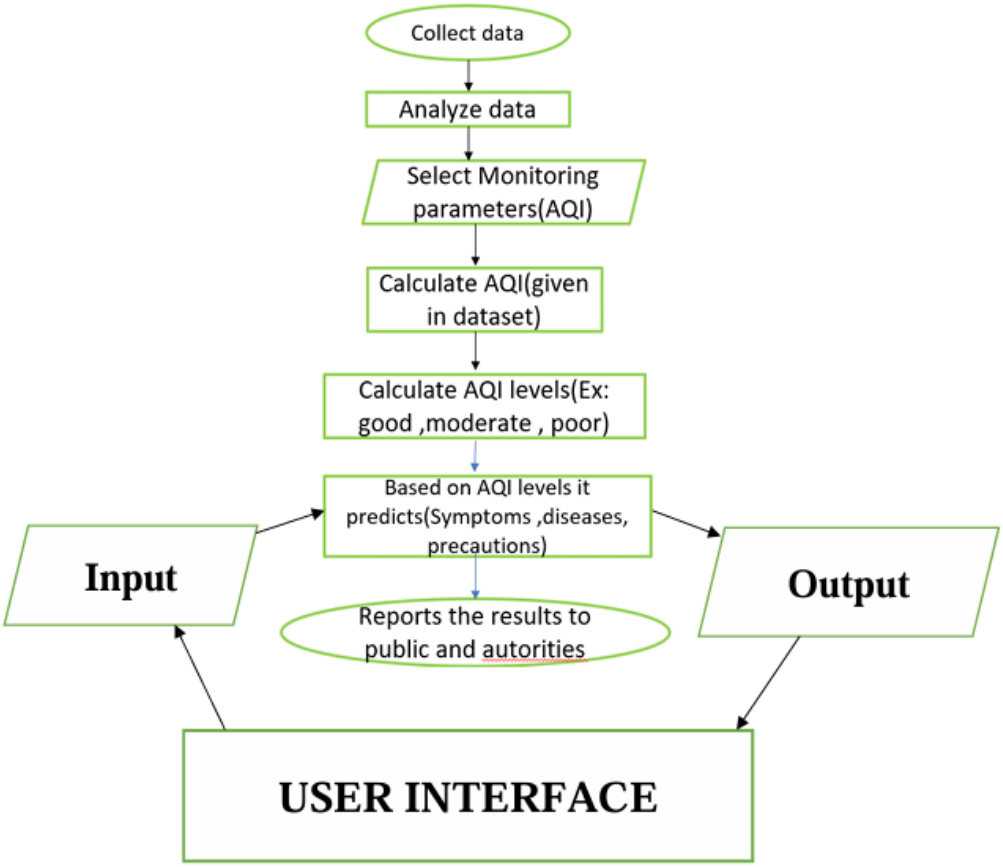
- **Training and Validation:** Use techniques like cross-validation to split data into training, validation, and test sets to assess model performance.
- **Hyperparameter Tuning:** Conduct hyperparameter tuning using grid search or random search to optimize model performance.

3. Performance Evaluation

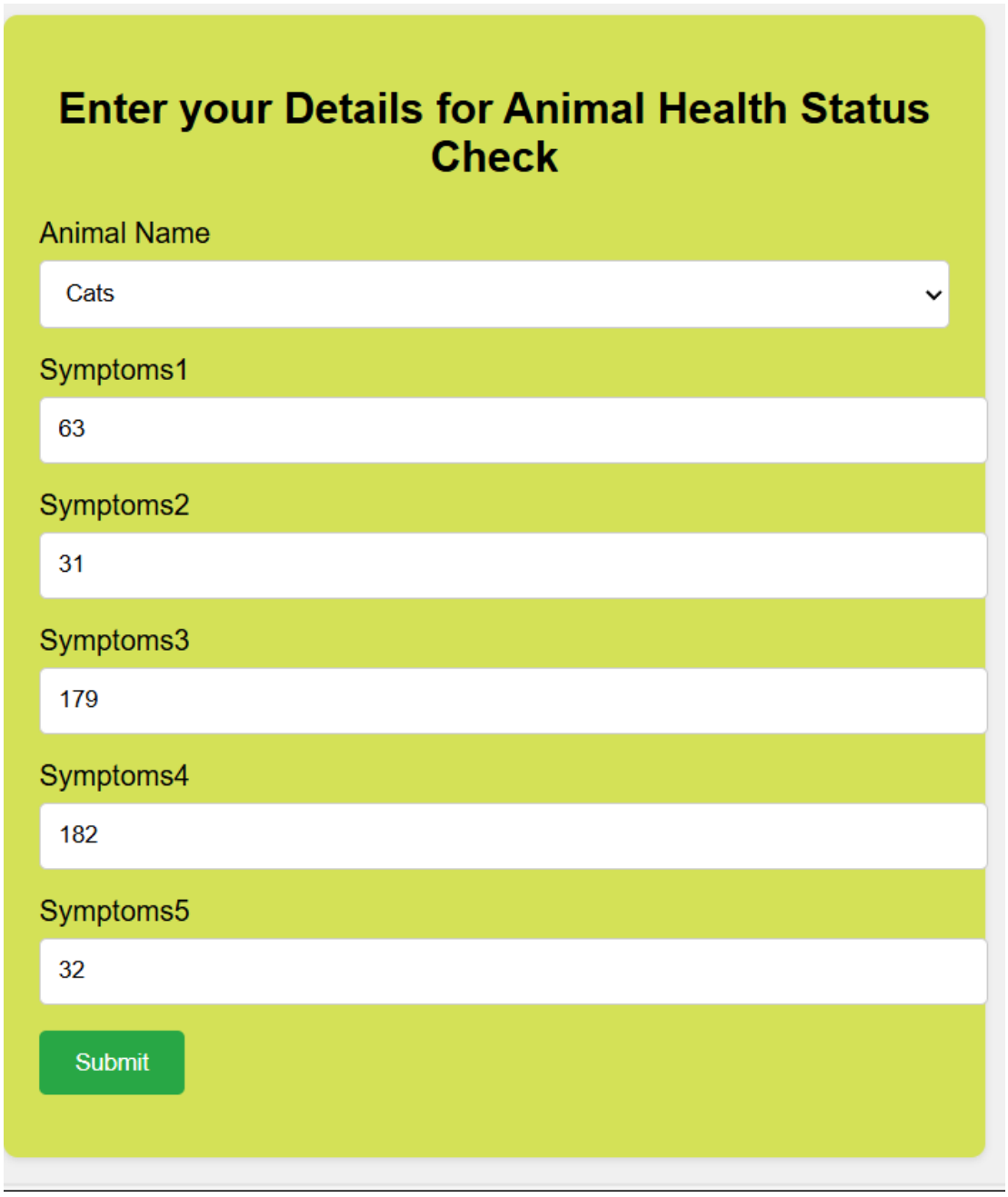
3.1. Model Metrics:

- **Quantitative Metrics:** Evaluate models based on quantitative metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
- **Qualitative Analysis:** Perform qualitative analysis of model predictions, including visual inspection of misclassifications and error analysis.

5.FLOWCHART


Collect data
Analyze data
Select Monitoring parameters(AQI)
Calculate AQI(given in dataset)
Calculate AQI levels(Ex: good ,moderate , poor)
Based on AQI levels it predicts(Symptoms ,diseases, precautions)
Reports the results to public and autorities
Input
Output
USER INTERFACE

6.RESULT


Enter your Details for Animal Health Status Check
Animal Name
Cats
Symptoms1
63
Symptoms2
31
Symptoms3
179
Symptoms4
182
Symptoms5
32
Submit

Prediction Result

{{ result }}

Go back

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

The journey of machine learning (ML) in animal health classification has introduced numerous advantages that significantly enhance veterinary medicine and animal care. Here are some of the key benefits:

1. Enhanced Diagnostic Accuracy: High Sensitivity and Specificity: ML models, especially deep learning algorithms, offer high sensitivity and specificity in diagnosing conditions from medical images, such as X-rays, MRIs, and ultrasounds.

2. Early Detection of Diseases: ML models can identify patterns that precede the onset of diseases. For instance, by analysing subtle changes in behaviour or physiological parameters, these models can help in the early detection of conditions such as lameness or metabolic disorders before they become severe.

3. Predictive Analytics: ML can forecast potential health issues based on historical data and trends. This predictive capability allows for proactive measures, reducing the incidence of disease outbreaks and improving overall herd or flock management.

DISADVANTAGES:

1. Data Quality and Availability: ML models rely heavily on high-quality, comprehensive data. In many cases, data related to animal health might be incomplete, inconsistent, or poorly labelled, which can negatively impact the performance of ML algorithms.

2.Data Privacy and Security: Handling sensitive animal health data raises concerns about privacy and security. Ensuring that data is protected from unauthorized access and breaches is critical, particularly when integrating data across different sources.

3. Bias and Fairness: ML models can inherit biases present in the training data. If the data used to train the model is biased or unrepresentative, the model's predictions might also be biased, leading to disparities in diagnosis.

8.APPLICATIONS

1. Disease Diagnosis and Detection: ML algorithms can analyse medical images such as X-rays, MRIs, and ultrasound scans to detect abnormalities or diseases. For instance, algorithms trained on images of cattle could help identify signs of mastitis or lameness.

2.Pathology: ML can assist in analysing biopsy samples or tissue slides to classify different types of infect aiding in accurate and timely diagnosis.

3.Disease Outbreak Prediction: ML models can forecast potential disease outbreaks by analysing patterns in historical health data, environmental conditions, and animal movement.

4.**Health Monitoring**: By analysing data from wearable sensors, ML can predict health issues before they manifest, such as detecting early signs of metabolic disorders or heat stress in livestock.

9.CONCLUSION

In conclusion, the unique journey of machine learning (ML) in animal health classification represents a significant leap forward in veterinary science and animal management. By harnessing the power of advanced algorithms and data analysis, ML offers transformative benefits that include enhanced diagnostic accuracy, early disease detection, and personalized treatment strategies.

ML improves diagnostic precision by analysing complex datasets with greater consistency and detail than traditional methods. This leads to more accurate and timely identification of diseases, enhancing the overall effectiveness of animal health management. Predictive analytics and real-time monitoring powered by ML enable early detection of health issues, allowing for preventive measures that can reduce the severity of diseases and minimize costs associated with late-stage treatments.

10.FUTURE SCOPE

The future scope and unique journey of ML in this field:

1. Enhanced Diagnostics: ML algorithms can analyse patterns in data from medical images, genetic information, or sensor readings to detect diseases at earlier stages than traditional methods.

2. Personalized Treatments: ML can help in developing personalized treatment plans based on an animal's specific health data, improving treatment outcomes.

3. Integration with Wearable Technology: Wearable devices on animals can collect continuous health data. ML algorithms can process this data to monitor health status in real-time and alert veterinarians to potential issues.

4. Genomics and Bioinformatics: ML models can analyse genetic sequences to identify markers associated with diseases, contributing to genetic research and breeding program.

11.BIBILOGRAPHY

1. **Bishop, C. M.** (2006). *Pattern Recognition and Machine Learning*. A comprehensive introduction to the theoretical aspects of machine learning, including key algorithms and their applications.
2. **Mitchell, T. M.** (1997). *Machine Learning*. McGraw. An essential text that covers the basics of machine learning techniques and their application.
3. **Kumar, V., & Clarke, R.** (2019). "Machine Learning in Animal Health: A Review." *Computers and Electronics in Agriculture*, 157, 219-230.Provides a detailed review of various machine learning applications in the field of animal health.
4. **Patel, S., & Singh, A.** (2021). "Advancements in Machine Learning for Animal Health Monitoring: A Comprehensive Review." *Journal of Animal Science and Technology*, 63(4), 123-140.
5. **Lo, B., & Wong, H.** (2018). "Deep Learning for Animal Health: Applications and Challenges." *IEEE Transactions on Biomedical Engineering*, 65(11), 2356-2365.Explores deep learning methodologies and their impact on animal health diagnostics.
6. **Gonzalez, G., & Pereira, P.** (2020). "Predictive Models for Livestock Health Using Machine Learning Techniques." *Computational Biology and Chemistry*, 85, 107-118.Examines predictive modelling approaches in livestock health using machine learning.
7. **Ding, Z., & Liu, Y.** (2022). "Automated Detection of Animal Diseases Using Computer Vision and Machine Learning." *Journal of Veterinary Science*, 23(2), 457-470.Focuses on the application of computer vision and machine learning for automated disease detection in animals.
8. **Zhao, L., & Zhang, Q.** (2020). "Machine Learning Approaches for Predicting Disease Outbreaks in Animal Populations." *Journal of Agricultural and Food Chemistry*, 68(12), 3456-3465.Investigates machine learning methods for predicting disease outbreaks in animal populations.

12.APPENDIX

Model building :

1)Dataset

2)Google collab and VS code Application Building

1. HTML file (Index file, Predict file)

1. CSS file

2. Models in pickle format

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html lang="eng ">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Animal Health Status Check</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        .header {
            background-colour: #f8f9fa;
            padding: 10px 0;
```

```
text-align: centre;
    }
    .header .contact-info {
        display: flex;
        justify-content: centre;
        gap: 20px;
        font-size: 14px;
    }
    .hero {
        text-align: centre;
        padding: 20px20px;
        background-colour: #e9ecef;
    }
    .navbar {
        background-colour: #fff;
        padding: 10px 0;
        box-shadow: 0 2px 4pxrgba(0, 0, 0, 0.1);
    }
    .navbar u l  {
        list-style: none;
        padding: 0;
        margin: 0;
        display: flex;
        justify-content: centre;
        gap: 20px;
```

```
}
.navbar u l li {
        display: inline;
    }
    .navbar u l li a {
        text-decoration: none;
        colour: #007bff;
        font-size: 16px;
    }
    .navbar u l li a .active {
        font-weight: bold;
    }
    .content {
        padding: 20px;
        text-align: centre;
        background-colour: #f0f0f0;
    }
    .form-container {
        max-width: 600px;
        margin: 0 auto;
        background-colour: #d4e157;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 2px 4pxrgba(0,0,0,0,1)
```

```
}
    .form-container h2 {
        margin-bottom: 20px;
    }
    .form-group {
        margin-bottom: 15px;
        text-align: left;
    }
    .form-group label {
        display: block;
        margin-bottom: 5px;
    }
    .form-group select,
    .form-group input {
        width: 100%;
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 4px;
    }
    .form-group button {
        background-colour: #28a745;
        colour: #fff;
        padding: 10px 20px;
        border: none;
```

```
  border-radius: 4px;
          cursor: pointer;
       }
   </style>
</head>
<body>

   <div class="header">
  <div class="contact-info">
          <span>📧 contact@example.com</span>
          <span>📞 +91 78453 23860</span>
       </div>
   </div>

   <div class="hero">
      <h1>Beyond the veil of wellness</h1>
   </div>

   <div class="navbar">
      <u l>
          <li><a h ref="#">Home</a></li>
          <li><a h ref="#">About</a></li>
          <li><a h ref="#">Contact</a></li>
          <li><a h ref="#" class="active">Let's Go</a></li>
```

```
</u l>
 </div>

 <div class="content">
    <div class="form-container">
        <h2>Enter your Details for Animal Health Status Check</h2>
        <form action="/submit" method="post">
            <div class="form-group">
                <label for="animal Name">Animal Name</label>
                <select id="animal Name" name="animal Name">
                    <option value="0">Birds</option>
                    <option value="1">Cats</option>
                    <!-- Add other options as needed -->
                </select>
            </div>
            <div class="form-group">
                <label for="symptoms1">Symptoms1</label>
                <input type="text" id="symptoms1"
name="symptoms1">
            </div>
            <div class="form-group">
                <label for="symptoms2">Symptoms2</label>
                <input type="text" id="symptoms2"
name="symptoms2">
```

```
  </div>
            <div class="form-group">
               <label for="symptoms3">Symptoms3</label>
               <input type="text" id="symptoms3"
name="symptoms3">
            </div>
            <div class="form-group">
               <label for="symptoms4">Symptoms4</label>
               <input type="text" id="symptoms4"
name="symptoms4">
            </div>
            <div class="form-group">
               <label for="symptoms5">Symptoms5</label>
               <input type="text" id="symptoms5"
name="symptoms5">
            </div>
            <div class="form-group">
               <button type="submit">Submit</button>
            </div>
         </form>
      </div>
   </div>

</body>
</html>
```

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prediction Result</title>
</head>
<body>
    <div class="content">
        <h2>Prediction Result</h2>
        <p>{{ result }}</p>
        <a h ref="/">Go back</a>
    </div>
</body>
</html>
```

APP.PY

```
from flask import Flask, request, render _template
import pickle
import NumPy as np
import pandas as pd
```

```
from s k learn .preprocessing import Standard Scaler

app = Flask(_name_)

# Load the model and scaler
model = pickle .load(open('rfc.pl', 'Rb'))
import job lib
scaler = Standard Scaler()
Scaler=job lib. load(open('scaler .pl' ,'Rb'))

@app.route('/')
def home():
    return render _template("index.html")

@app.route('/predict', methods=["POST", "GET"])
def predict():
    return render _template("output.html")

@app.route('/submit', methods=["POST", "GET"])
def submit():
    if request .method == "POST":
# Read the inputs given by the user
        Animal _name = request .form['animal Name']
        symptoms = [
            request .form['symptoms1'],
```

```
            request. form['symptoms2'],
            request. form['symptoms3'],
            request. form['symptoms4'],
            request. form['symptoms5']
        ]

        # Example: Convert inputs to appropriate format
        Input _features = [animal _name] + symptoms
        Input _features = np. array (input _features). reshape(1, -1)

        # Scale the features
        Input _features = scaler. transform(input _features)

        # Predict using the loaded model
        prediction = model .predict(input _features)

        result = "Your health is in normal condition." if prediction == 1 else
"According to our study, we feel sad."

        return render _template("output.html", result=result)

if _name_ == "_main_":
    app .run(debug=True)
    debug=True ,post=='0.0.0.0'
```

CODE SNNIPETS

DATA COLLECTION AND PREPARATION

```
# Import the Libraries
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # Linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np  # Linear algebra
import pandas as pd  # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import f1_score, make_scorer, accuracy_score, recall_score, p
from sklearn.metrics import confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
import random
import warnings
warnings. filterwarnings ("ignore")
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will List al
#input directory
```

```
[2] df=pd.read_csv("/content/data.csv")
    df
```

	AnimalName	symptoms1	symptoms2	symptoms3	symptoms4	symptoms5	Dangerous
0	Dog	Fever	Diarrhea	Vomiting	Weight loss	Dehydration	Yes
1	Dog	Fever	Diarrhea	Coughing	Tiredness	Pains	Yes
2	Dog	Fever	Diarrhea	Coughing	Vomiting	Anorexia	Yes
3	Dog	Fever	Difficulty breathing	Coughing	Lethargy	Sneezing	Yes
4	Dog	Fever	Diarrhea	Coughing	Lethargy	Blue Eye	Yes
...	...	...	...	...	...	...	...
866	Buffaloes	Fever	Difficulty breathing	Poor Appetite	Eye and Skin change	Unable to exercise	Yes
867	Buffaloes	Fever	Loss of appetite	Lession on the skin	Lethargy	Joint Pain	Yes
868	Buffaloes	Lesions in the nasal cavity	Lesions on nose	Vomiting	Noisy Breathing	Lesions on nose	Yes
869	Buffaloes	Hair loss	Dandruff	Vomiting	Crusting of the skin	Ulcerated skin	Yes
870	Buffaloes	Greenish-yellow nasal discharge	Lack of pigmentation	Vomiting	Lethargy	Pain on face	Yes

```
[4] df.describe()
```

	AnimalName	symptoms1	symptoms2	symptoms3	symptoms4	symptoms5	Dangerous
count	871	871	871	871	871	871	869
unique	46	232	230	229	217	203	2
top	Buffaloes	Fever	Diarrhea	Coughing	Weight loss	Pains	Yes
freq	129	257	119	95	117	99	849

```
[5] df.isnull().sum()
```

	0
AnimalName	0
symptoms1	0
symptoms2	0
symptoms3	0
symptoms4	0
symptoms5	0
Dangerous	2

dtype: int64

```
[6] df['Dangerous'].unique()
```

```
array(['Yes', 'No', nan], dtype=object)
```

```
[7] df['Dangerous'].value_counts()
```

Dangerous	count
Yes	849
No	20

dtype: int64

```
[8] df['Dangerous'].fillna('Yes',inplace=True)
```

```
[9] df.isnull().sum()
```

	0
AnimalName	0
symptoms1	0
symptoms2	0
symptoms3	0

[9]

symptoms4	0
symptoms5	0
Dangerous	0

dtype: int64

```
[10] df.columns
```

```
Index(['AnimalName', 'symptoms1', 'symptoms2', 'symptoms3', 'symptoms4',
       'symptoms5', 'Dangerous'],
      dtype='object')
```

EXPLORATORY DATA ANALYSIS

```
[11] df.nunique()
```

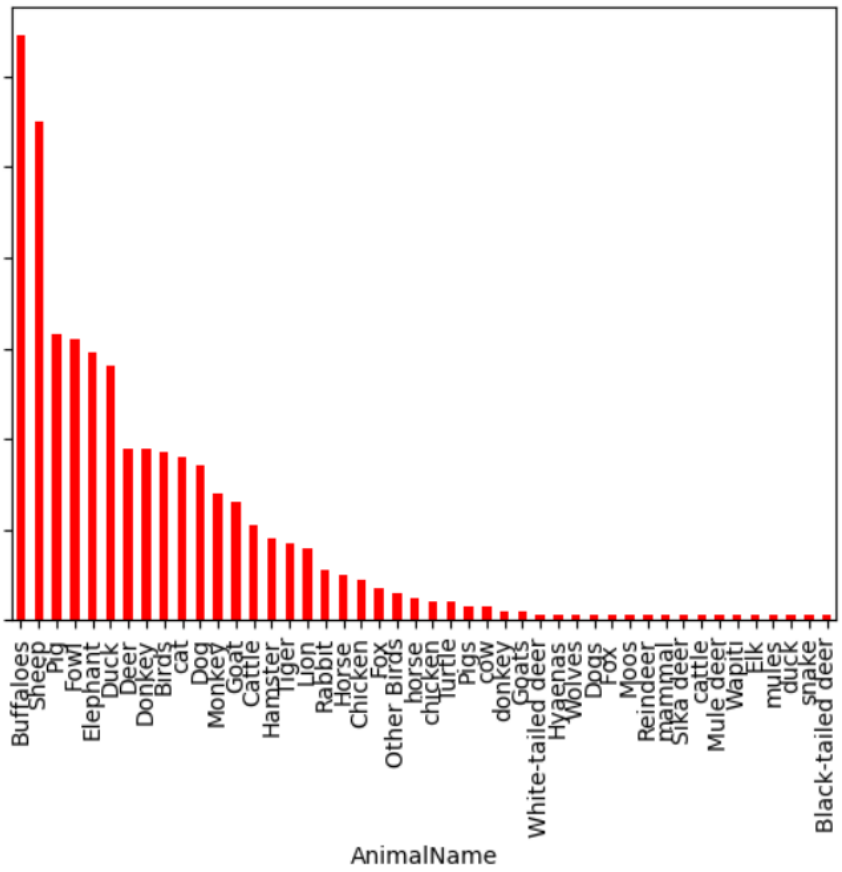
	0
AnimalName	46
symptoms1	232
symptoms2	230
symptoms3	229
symptoms4	217
symptoms5	203
Dangerous	2

dtype: int64

```
[12] df['AnimalName'].value_counts().plot(kind='bar',color='Red')
```

[12]

<Axes: xlabel='AnimalName'>

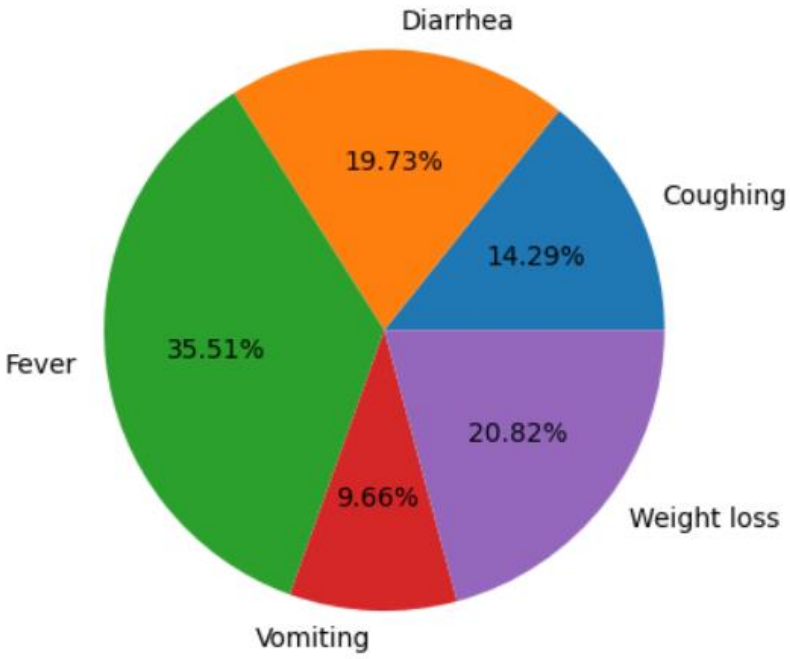


[13]

```
# List of the first 5 specified diseases
specified_diseases = ['Fever', 'Diarrhea', 'Vomiting', 'Weight loss', 'Coughing']

# Filter the DataFrame to keep only rows where any of the symptoms columns contain the sp
filtered_df = df[
    df['symptoms1'].isin(specified_diseases) |
    df['symptoms2'].isin(specified_diseases) |
    df['symptoms3'].isin(specified_diseases) |
    df['symptoms4'].isin(specified_diseases) |
    df['symptoms5'].isin(specified_diseases)
]
```

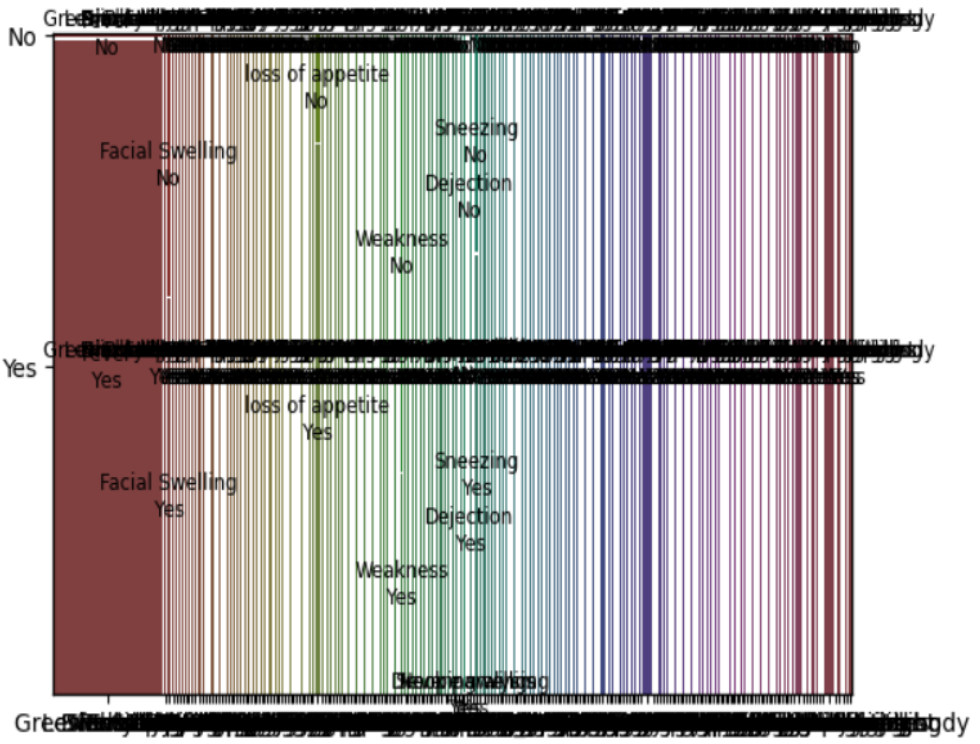
```
[14] # Group by the symptom columns and count occurrences of specified diseases
     symptom_counts = pd.Series()
     for col in ['symptoms1', 'symptoms2', 'symptoms3', 'symptoms4', 'symptoms5']:
         symptom_counts = symptom_counts.add(filtered_df[col].value_counts(), fill_value=0)

     # Filter to only include the specified diseases
     symptom_counts = symptom_counts[symptom_counts.index.isin(specified_diseases)]

     # Plot the pie chart
     symptom_counts.plot(kind='pie', autopct='%1.2f%%')
```

```
<Axes: >
```



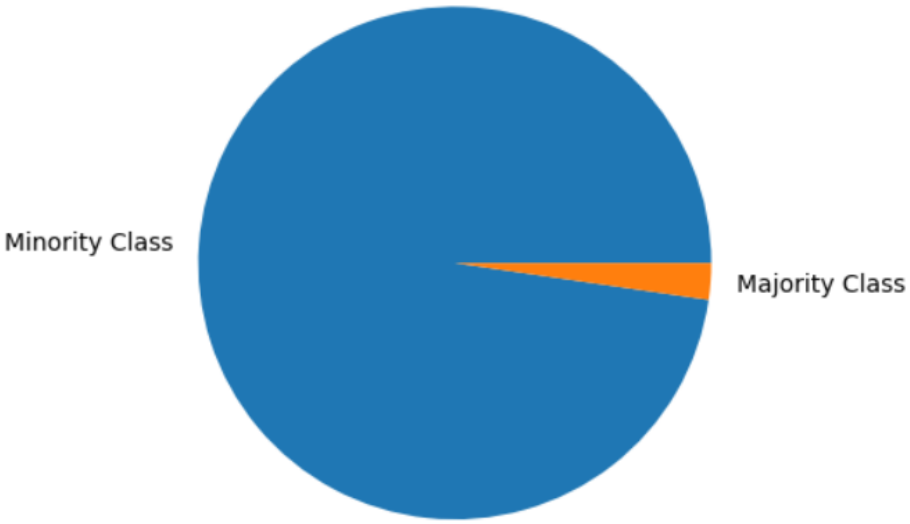

```
[15] import matplotlib.pyplot as plt
     from statsmodels.graphics.mosaicplot import mosaic
     mosaic(df, ['symptoms1', 'Dangerous' ])
     plt.show()
```

[15]



[16]
```
class_division = [df[df["Dangerous"] == "Yes"].shape[0], df[df["Dangerous"] == "No"].shap
my_labels = ["Minority Class", "Majority Class"]
plt.pie(class_division, labels=my_labels)
plt.show()
```

[ ]




```
[ ] from sklearn.preprocessing import LabelEncoder

    le = LabelEncoder()
    df['AnimalName'] = le.fit_transform(df['AnimalName'])
    df['symptoms1'] = le.fit_transform(df['symptoms1'])
    df[ 'symptoms2']=le. fit_transform(df[ 'symptoms2'])
    df[ 'symptoms3']=le.fit_transform(df[ 'symptoms3'])
    df[ 'symptoms4']=le.fit_transform(df['symptoms4'])
    df[ 'symptoms5' ]=le.fit_transform(df['symptoms5'])
    df[ 'Dangerous']=le.fit_transform(df[ 'Dangerous'])
```

```
[ ]  print (accuracy_df)
```

```
[ ]  models = ['Logistic regression', 'Decision tree', 'Random fores
     accuracy = [acc_1*100, acc_dt*100, acc_rfc*100, acc_knn*100]
     sns.barplot(x=models, y=accuracy, color='Red')
     plt.title('Model Accuracy')
     plt.show()
```

MODEL DEPLOYMENT

```
[ ]  import pickle
     pickle.dump(rfc, open("rfc.pkl", "wb"))
```

```
[ ]  print(rfc.predict([[4,3,2,6,4,5]]))
```

```
[18] df
```

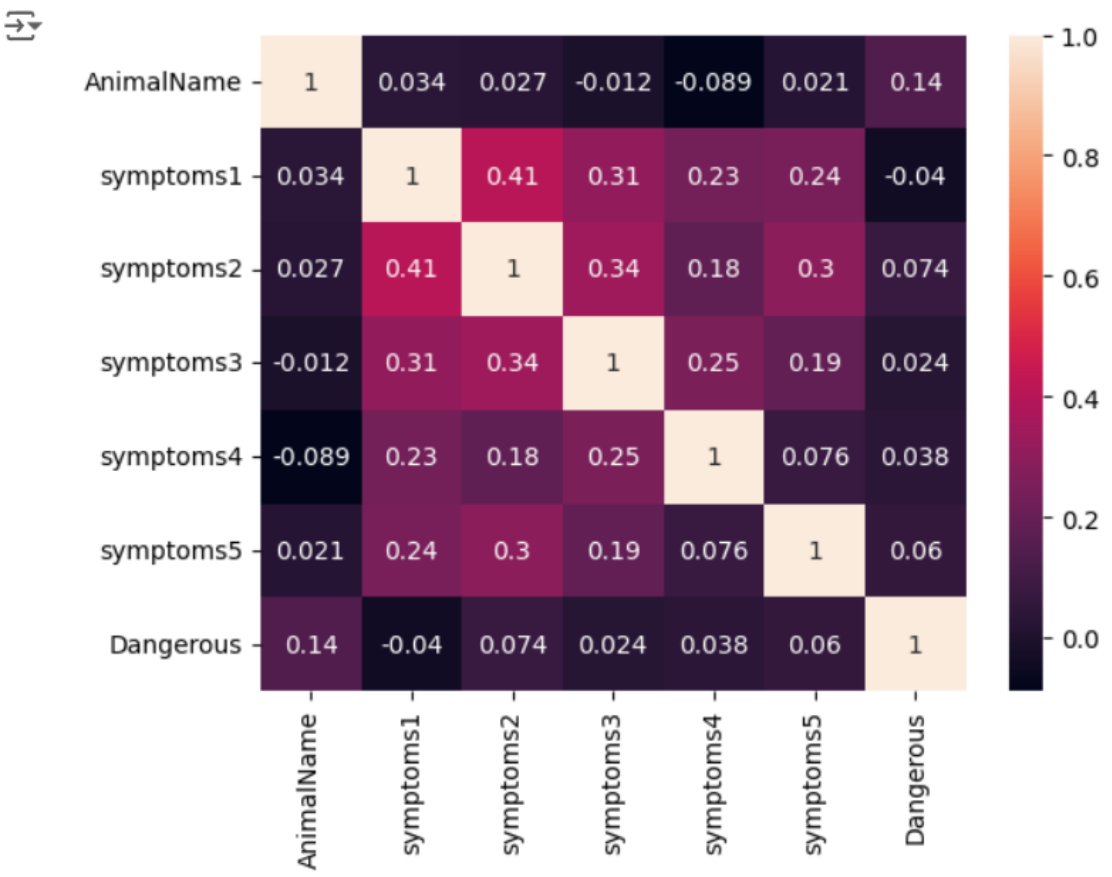
	AnimalName	symptoms1	symptoms2	symptoms3	symptoms4	symptoms5	Dangerous
0	6	63	31	179	182	32	1
1	6	63	31	31	165	113	1
2	6	63	31	31	173	8	1
3	6	63	34	31	87	142	1
4	6	63	31	31	87	21	1
...	...	...	...	...	...	...	...
866	2	63	34	115	53	160	1
867	2	63	95	88	87	78	1
868	2	97	90	179	107	82	1
869	2	77	26	179	27	159	1
870	2	75	84	179	87	111	1

871 rows × 7 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
[19] sns.heatmap(df.corr(),annot=True)
     plt.show()
```

```
[19] sns.heatmap(df.corr(),annot=True)
     plt.show()
```



MODEL BUILDING

```
[20] from imblearn.over_sampling import SMOTE

     # Assuming 'df' is your DataFrame and 'Dangerous' is your target variable
     # Adjust column names as needed
     X = df.drop('Dangerous', axis=1)  # Features
     y = df['Dangerous']  # Target variable

     # Split the dataset into training and testing sets
     from sklearn.model_selection import train_test_split
     xtrain, xtest, ytrain, ytest = train_test_split(X, y, test_size=0.2, random_state=42)  #

     smote = SMOTE(sampling_strategy='auto', random_state=42)
     xbal, ybal = smote.fit_resample(xtrain, ytrain)
     print(ybal.value_counts())
```

```
Dangerous
1    678
0    678
Name: count, dtype: int64
```

```
[21] x = df.drop(['Dangerous'], axis=1)
     y = df['Dangerous']
     from sklearn.preprocessing import StandardScaler

     stx = StandardScaler()
     x = stx.fit_transform(x)
```

```
[22] stx.fit(xtrain)
```

```
StandardScaler
StandardScaler()
```

```
[23] stx.fit(xtest)
```

```
StandardScaler
StandardScaler()
```

```
[24] from sklearn.model_selection import train_test_split
     xtrain, xtest,ytrain, ytest=train_test_split(X,y,test_size=0.3, random_state=42, stratif
```

```
[25] print("Proportion of Minority Class in train set: " + str(round(ytrain.sum()/len(ytrain)
     print("Proportion of Minority Class in test set: " + str(round(ytest.sum()/len(ytest) * 1
```

```
Proportion of Minority Class in train set: 97.7%
Proportion of Minority Class in test set: 97.71%
```

```
[26] from sklearn.linear_model import LogisticRegression

     lr = LogisticRegression()
     lr.fit(xbal, ybal)
```

```
LogisticRegression
LogisticRegression()
```

```
[27] ytestpred = lr.predict(xtest)
     ytrainpred = lr.predict(xtrain)
     print(accuracy_score(ytest, ytestpred))
     print(accuracy_score(ytrain, ytrainpred))
```

```
0.7709923664122137
0.8045977011494253
```

```
[28] from sklearn.neighbors import KNeighborsClassifier
     knn = KNeighborsClassifier()
     knn.fit(xbal, ybal)
```

```
KNeighborsClassifier
KNeighborsClassifier()
```

```
[29] print(accuracy_score(ytest, ytestpredk))
     print(accuracy_score(ytrain, ytrainpredk))
```

```
0.9541984732824428
0.9408866995073891
```

```
[30] from sklearn.tree import DecisionTreeClassifier
     dtc = DecisionTreeClassifier()
     dtc.fit(xbal, ybal)
```

```
DecisionTreeClassifier
DecisionTreeClassifier()
```

```
[31] ytestpredc=dtc.predict(xtest)
     ytrainpredc=dtc.predict(xtrain)
     print (accuracy_score(ytest, ytestpredc))
     print(accuracy_score(ytrain, ytrainpredc))
```

```
0.9923664122137404
0.986863711001642
```

```
[32] from sklearn.ensemble import RandomForestClassifier

     rfc = RandomForestClassifier()
     rfc.fit(xbal, ybal)
```

```
RandomForestClassifier
RandomForestClassifier()
```

```
[33] ytestpredr=rfc.predict(xtest)
     ytrainpredr=rfc.predict(xtrain)
     print (accuracy_score (ytest, ytestpredr))
     print(accuracy_score(ytrain, ytrainpredr))
```

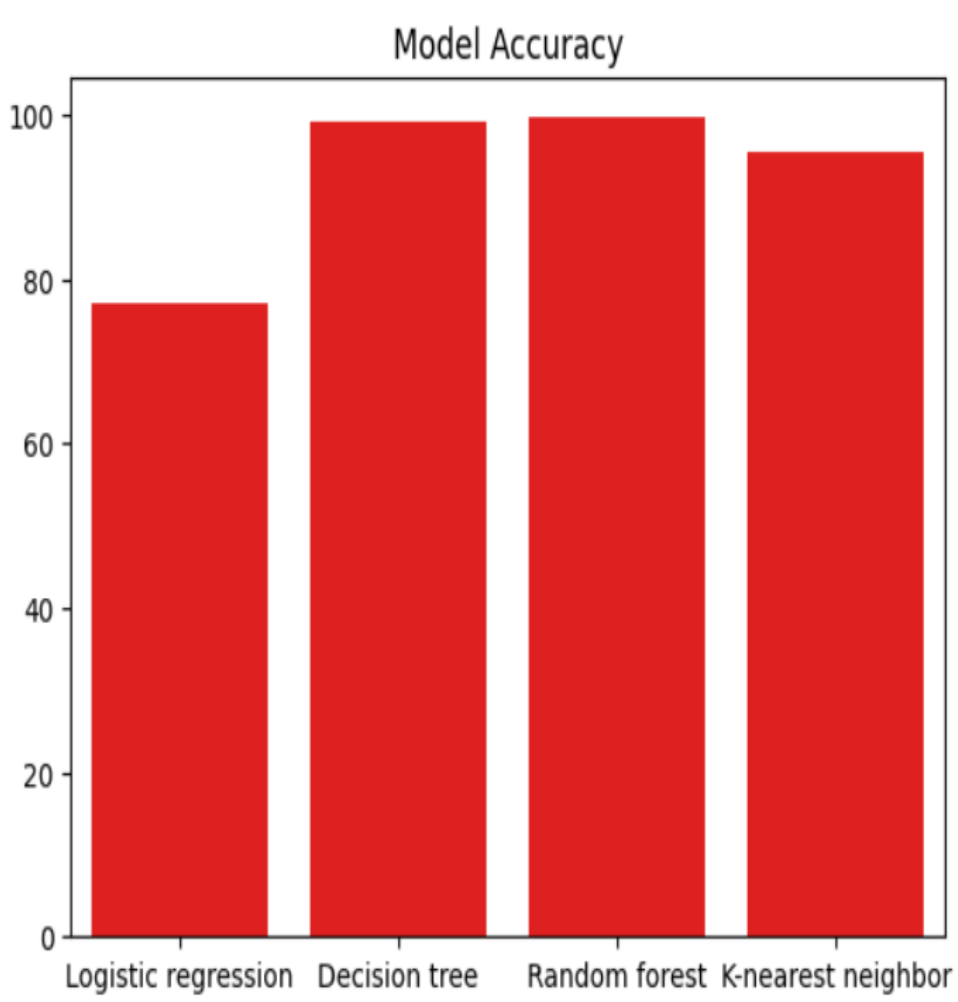
```
0.9961832061068703
0.993431855500821
```

PERFORMANCE TESTING FINDING THE BEST MODEL

```
[34] # Assuming 'lr' is your fitted LogisticRegression model and you've already made predictic
     ytestpredlr = lr.predict(xtest)
     ytrainpredlr = lr.predict(xtrain)
     ytestpreddtc = dtc.predict(xtest)
     ytrainpreddtc = dtc.predict(xtrain)
     ytestpredrfc= rfc.predict(xtest)
     ytrainpredrfc = rfc.predict(xtrain)
     ytestpredknn = knn.predict(xtest)
     ytrainpredknn = knn.predict(xtrain)
     from sklearn.metrics import accuracy_score
     acc_1 = accuracy_score(ytest, ytestpredlr)  # Calculate and store accuracy for logistic r
     acc_dt = accuracy_score(ytest, ytestpreddtc)  # Calculate accuracy for decision tree
     acc_rfc = accuracy_score(ytest, ytestpredrfc)  # Calculate and store the accuracy for the
     acc_knn = accuracy_score(ytest, ytestpredknn) # Calculate and store accuracy for k-neares
     accuracy_df = pd.DataFrame({
         "model": ['Logistic regression', 'Decision tree', 'Random forest', 'K-nearest neigbou
         "Accuracy": [acc_1 * 100, acc_dt * 100, acc_rfc * 100, acc_knn * 100]
     })
     print (accuracy_df)
```

```
                  model   Accuracy
0  Logistic regression  77.099237
1        Decision tree  99.236641
2        Random forest  99.618321
3   K-nearest neigbour  95.419847
```

```
[35] models = ['Logistic regression', 'Decision tree', 'Random forest', 'K-nearest neighbor']
     accuracy = [acc_1*100, acc_dt*100, acc_rfc*100, acc_knn*100]
     sns.barplot(x=models, y=accuracy, color='Red')
     plt.title('Model Accuracy')
     plt.show()
```



MODEL DEPLOYMENT

```
[36] import pickle
     pickle.dump(rfc, open("rfc.pkl", "wb"))
```

```
[37] print(rfc.predict([[4,3,2,6,4,5]]))
```

```
[1]
```