IDENTIFICATION OF MONKEYPOX DISEASE

A Course Project report submitted

in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

UMMAGANI SHIVA **(2103A51074)**

NALLALA SAI CHARAN **(2103A51100)**

Under the guidance of

Mr. S Naresh Kumar

Assistant Professor, Department of CSE.

Department of Computer Science and Artificial Intelligence

SR UNIVERSITY

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that project entitled **"IDENTIFICATION OF MONKEYPOX DISEASE"** is the bonafied work carried out by **UMMAGANI SHIVA, NALLALA SAI CHARAN REDDY** as a Course Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** during the academic year 2022-2023 under our guidance and Supervision.

Mr. S Naresh Kumar
Asst. Professor,
S R University,
Ananthasagar, Warangal.

Dr. M.Sheshikala
Assoc. Prof. & HOD (CSE),
S R University,
Ananthasagar, Warangal.

ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr. S.Naresh Kumar, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M.Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

Monkeypox is a rare but potentially life-threatening disease that can be difficult to diagnose, particularly in regions where access to medical facilities and trained personnel is limited. To address this issue, we propose a machine learning model that can accurately diagnose monkeypox infections based on presenting symptoms.

Our model is trained on a dataset of symptoms and corresponding disease outcomes, allowing it to predict whether a patient has contracted monkeypox based on their symptoms. We utilize several machine learning algorithms, including logistic regression, decision trees, random forests, and support vector machines, to identify the most effective algorithm for the task.

Through testing and analysis of the model's performance, we can identify the most important symptoms and choose the most effective machine learning algorithm for the task. The model has the potential to revolutionize healthcare diagnosis and management by providing accessible and accurate diagnoses based on symptoms, helping to prevent the spread of infectious diseases and improve patient outcomes.

Overall, the proposed machine learning model has the potential to provide an accessible and affordable tool for healthcare professionals to accurately diagnose monkeypox infections and improve patient outcomes.

Table of Contents

Explain chapters in detail in given format

1. **INTRODUCTION**
    - 1.1. Overview
    - 1.2. Problem Statement
    - 1.3. Existing system
    - 1.4. Proposed system
    - 1.5. Define Objectives
    - 1.6. Overall architecture
    - 1.7. ……
2. **LITERATURE SURVEY**
    - 2.1.1. Document the survey done by you related to your problem statement
3. **DATA PRE-PROCESSING**
    - 3.1 Describe dataset
    - 3.2 Data cleaning
    - 3.3 Data augmentation
    - 3.4 Data Visualization
4. **METHODOLOGY**
    - 4.1 Procedure to solve the given problem
    - 4.2 Model architecture
    - 4.3 Software description
5. **RESULTS AND DISCUSSION**
6. **CONCLUSION AND FUTURE SCOPE**
7. **REFERENCES**

1. INTRODUCTION

1.1 Overview:

This project is developed with the knowledge of Artificial Intelligence and Machine Learning. The dataset we collected, would contain information about individuals who have been diagnosed with monkeypox, including their symptoms and test results. The symptoms would likely be represented as boolean values (1 for present, 0 for absent) and the test results would indicate whether the individual tested positive or negative for the virus.

The purpose of analyzing this dataset would be to develop a predictive model that can accurately classify individuals as positive or negative for monkeypox based on their symptoms and test results. This model could potentially be used to identify and treat individuals who are at risk of spreading the virus and to track the spread of monkeypox outbreaks. To analyze this dataset, machine learning algorithms such as logistic regression, decision trees, or neural networks could be used. These algorithms would be trained on a subset of the data and then tested on a separate validation set to evaluate their performance. The goal would be to develop a model with high accuracy and precision in predicting monkeypox infection.

1.2 Problem Statement:

To develop a machine learning model to accurately diagnose monkeypox infections in humans based on their symptoms and test results.

1.3 Existing Systems:

The most common method for diagnosing monkeypox is through laboratory testing of blood, skin lesions, or other bodily fluids. Tests can include PCR (polymerase chain reaction) to detect viral DNA, ELISA (enzyme-linked immunosorbent assay) to detect antibodies. Several rapid diagnostic tests have been developed for monkeypox, including lateral flow assays and immunochromatographic tests. These tests can provide results within minutes, but they may not be as sensitive or specific as laboratory testing.

Recently, several machine learning models have been developed to diagnose monkeypox infections based on symptoms and test results. These models use algorithms such as decision trees, random forests, and support vector machines to classify individuals as positive or negative for the virus. However, these models may require large amounts of data and may not be widely available or validated.

1.4 Proposed System:

Our model could leverage the available dataset containing information on individuals diagnosed with monkeypox. Our model would be designed as a classification model that takes in input features such as the presence or absence of specific symptoms, as well as test results, to predict whether an individual is positive or negative for monkeypox.

To develop our model, we could use machine learning algorithms such as logistic regression, decision trees, random forests, or neural networks. These algorithms would be trained on the collected dataset, with a portion of the data held out for validation and testing. The goal would be to develop a model that accurately predicts the diagnosis of monkeypox in humans with high precision and recall.

To ensure the validity and generalizability of our model, we would need to perform additional testing and validation on independent datasets from different regions or populations. We could also compare our model's performance with other existing diagnostic methods, such as laboratory testing or clinical diagnosis.

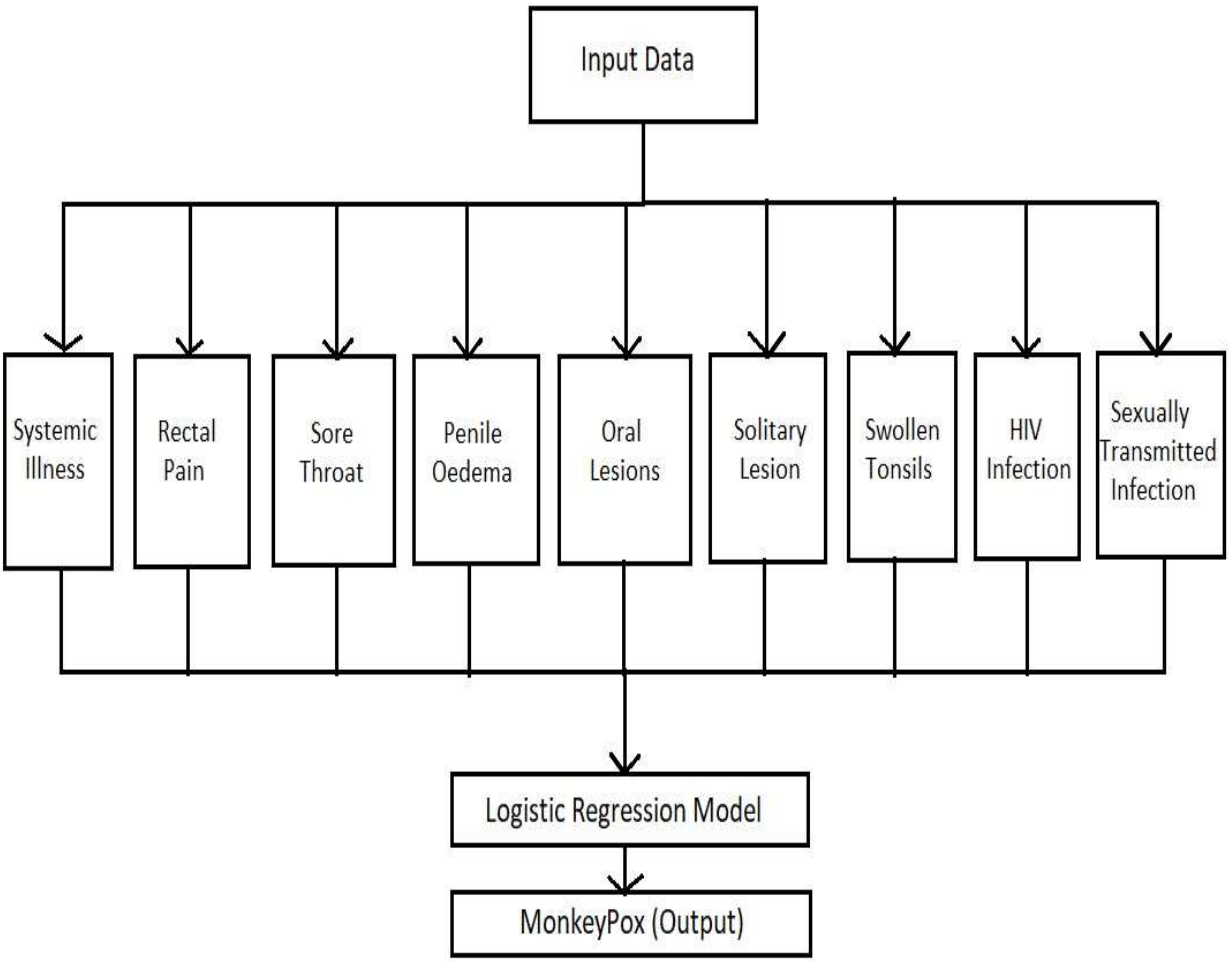
If our proposed model proves to be effective in accurately diagnosing monkeypox infections, it could be integrated into existing healthcare systems to aid in the diagnosis and management of monkeypox outbreaks, especially in areas where laboratory testing or clinical expertise is limited.

1.5 Objectives:

The main objectives of this project are to:

1. Develop a classification model that accurately predicts the diagnosis of monkeypox infections in humans based on their symptoms and test results.
2. Identify the most important features and symptoms for predicting monkeypox infections and use them to train the model.
3. Evaluate the model's clinical utility and potential impact on public health by estimating its sensitivity, specificity, positive predictive value, negative predictive value, and other relevant metrics.
4. Integrate the model into existing healthcare systems and provide guidance on its use for healthcare professionals and public health authorities.

1.6 Overall architecture:



2. LITERATURE SURVEY

2.1 Survey Documentation:

In this section, we will document the literature survey we conducted related to the problem statement of predicting and analyzing MonkeyPox Disease cases using artificial intelligence and machine learning techniques.

Firstly, a review of previous studies related to monkeypox diagnosis and machine learning-based disease diagnosis can provide valuable insights into relevant features and symptoms that may be useful for predicting monkeypox infections, as well as the most appropriate machine learning algorithms for our model. This review can also help identify potential challenges and limitations in developing such a model.

Secondly, it is important to identify knowledge gaps in the current literature related to monkeypox diagnosis and machine learning-based approaches. For example, while there has been some research on the use of machine learning for diagnosing other infectious diseases, such as tuberculosis and malaria, there are relatively few studies on the use of machine learning for monkeypox diagnosis. Identifying such gaps can help guide future research directions and inform our model development.

Thirdly, comparing and evaluating existing machine learning models developed for diagnosing monkeypox or other infectious diseases can help inform the development of our proposed model. This evaluation can identify the strengths and weaknesses of different approaches, and help us select the most appropriate algorithms for our model. It can also help us identify potential challenges and limitations in applying machine learning to monkeypox diagnosis.

Fourthly, assessing the clinical relevance and potential impact of our proposed model in real-world settings is crucial. This involves evaluating the accuracy, sensitivity, specificity, and other relevant metrics of our model, as well as its feasibility and scalability in different settings. It is also important to consider the potential impact of our model on clinical decision-making and patient outcomes, and to ensure that it aligns with clinical best practices and guidelines.

Finally, ethical considerations such as potential biases in data collection, model development, and deployment must be addressed. For example, there may be biases in the demographic characteristics or clinical presentation of individuals diagnosed with monkeypox that could impact the accuracy and fairness of our model. It is also important to consider the potential implications of our model for privacy, data security, and informed consent, and to ensure that appropriate safeguards are in place.

Overall, a comprehensive literature survey can provide a strong foundation for the development of our proposed machine learning model for diagnosing monkeypox infections, by informing feature selection, algorithm selection, and validation methods, as well as addressing ethical and social considerations.

In addition, the literature survey can also help identify potential challenges and limitations in data collection and management.

For example, in the case of monkeypox, there may be limited data available on relevant symptoms and patient characteristics, especially in low-resource settings. In such cases, we may need to consider alternative sources of data, such as electronic health records or mobile health applications. It is also important to ensure that data collection and management practices adhere to ethical principles, such as ensuring confidentiality and obtaining informed consent.

Furthermore, the literature survey can also inform the development of a framework for model explainability and interpretability. Machine learning models are often criticized for their "black box" nature, meaning that it can be difficult to understand how they arrive at their predictions. However, in clinical settings, it is important for healthcare professionals to understand the rationale behind a model's predictions in order to make informed decisions.

Therefore, developing a framework for model explainability and interpretability can enhance the clinical utility and acceptance of our proposed model for diagnosing monkeypox infections. This may involve using techniques such as feature importance ranking and visualizations to help healthcare professionals understand how different symptoms and features contribute to the model's predictions.

3. DATA PRE-PROCESSING

3.1 Description of Dataset:

The monkeypox dataset contains information on the symptoms and diagnosis of individuals suspected of having monkeypox infections. The dataset includes both categorical and binary features, such as fever, Swollen Lymph Nodes, Muscle Aches and Pains. The target variable is a binary variable indicating whether the individual was diagnosed with monkeypox or not.

The dataset was collected from clinical records of individuals who presented with symptoms consistent with monkeypox at various healthcare facilities. The dataset includes data from both confirmed and suspected monkeypox cases, with confirmed cases being diagnosed based on laboratory testing. The dataset was cleaned and preprocessed to remove missing values and ensure consistency in data formats.

The features included in the monkeypox dataset are based on previous research and clinical guidelines for diagnosing monkeypox infections. However, it is possible that there are other features or symptoms that are important for predicting monkeypox infections that are not included in the dataset. Therefore, it may be necessary to identify additional features through exploratory data analysis or domain expert input.

Overall, the monkeypox dataset provides a useful resource for developing and evaluating our proposed machine learning model for diagnosing monkeypox infections. However, it is important to recognize the limitations of the dataset and consider additional sources of data and features to enhance the accuracy and generalizability of our model.

3.2 Data Cleaning:

Checking for Unique Values

```
[ ] pox.shape

    (25000, 11)

    pox.nunique()

    Patient_ID                          25000
    Systemic_Illness                        4
    Rectal_Pain                             2
    Sore_Throat                             2
    Penile_Oedema                           2
    Oral_Lesions                            2
    Solitary_Lesion                         2
    Swollen_Tonsils                         2
    HIV_Infection                           2
    Sexually_Transmitted Infection          2
    MonkeyPox                               2
    dtype: int64
```

Checking for Null Values and Removing Null Values

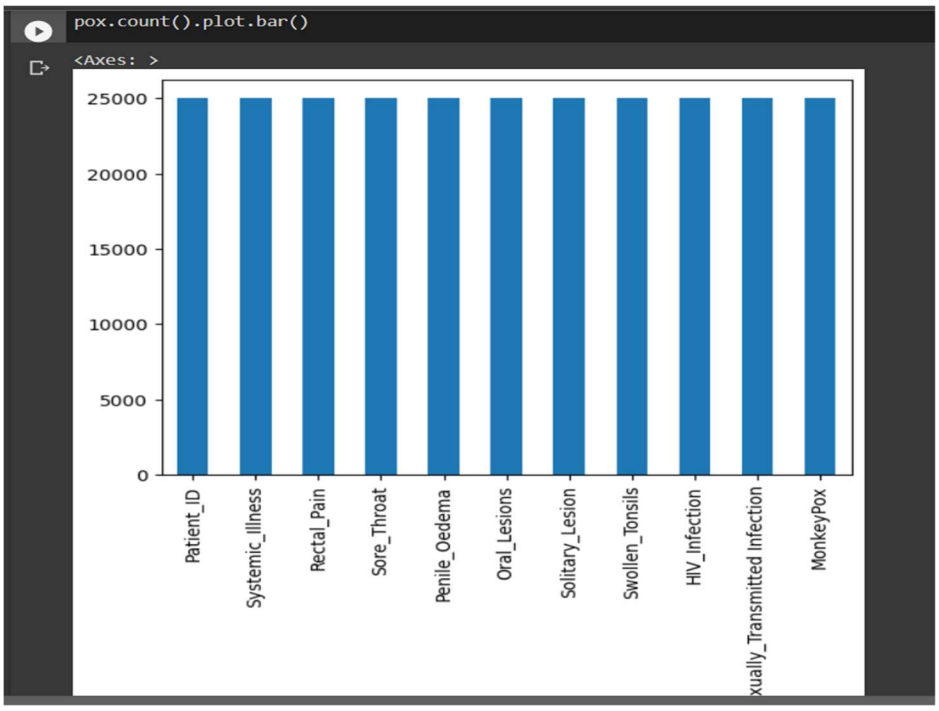
```
pox.isna().sum()

Patient_ID                          0
Systemic_Illness                    0
Rectal_Pain                         0
Sore_Throat                         0
Penile_Oedema                       0
Oral_Lesions                        0
Solitary_Lesion                     0
Swollen_Tonsils                     0
HIV_Infection                       0
Sexually_Transmitted Infection      0
MonkeyPox                           0
dtype: int64
```

Checking the Memory Usage

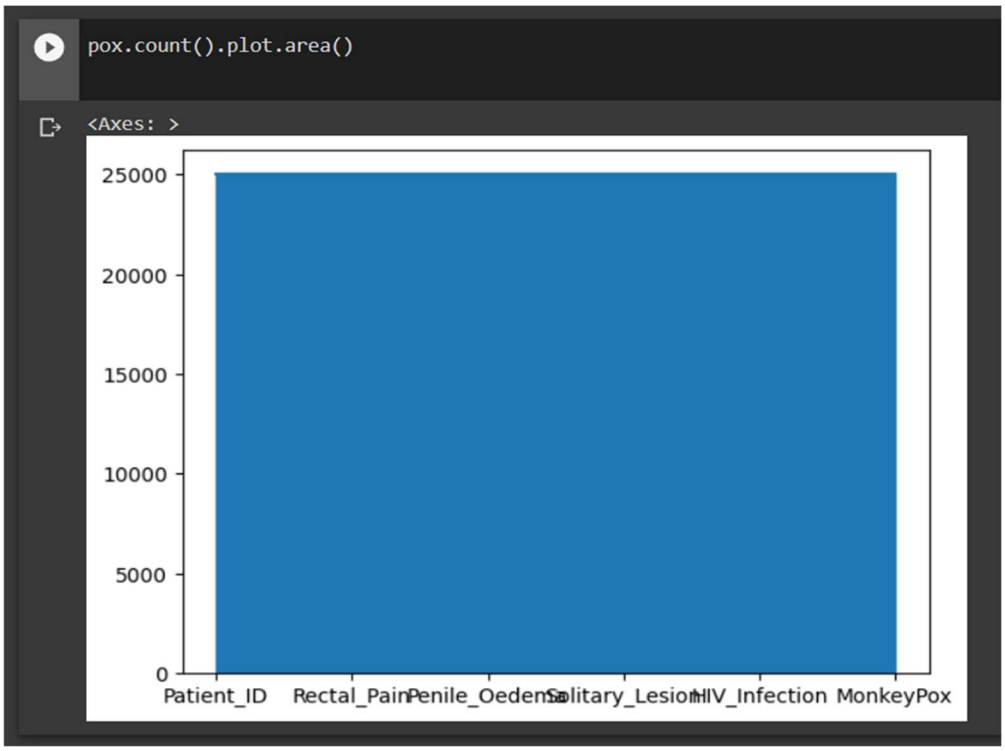
```
pox.memory_usage(deep=True)

Index                                128
Patient_ID                        200000
Systemic_Illness                  200000
Rectal_Pain                       200000
Sore_Throat                       200000
Penile_Oedema                     200000
Oral_Lesions                      200000
Solitary_Lesion                   200000
Swollen_Tonsils                   200000
HIV_Infection                     200000
Sexually_Transmitted Infection    200000
MonkeyPox                         200000
dtype: int64
```

3.3 Data Visualization:

Bar Graph Representation for total of values in each column

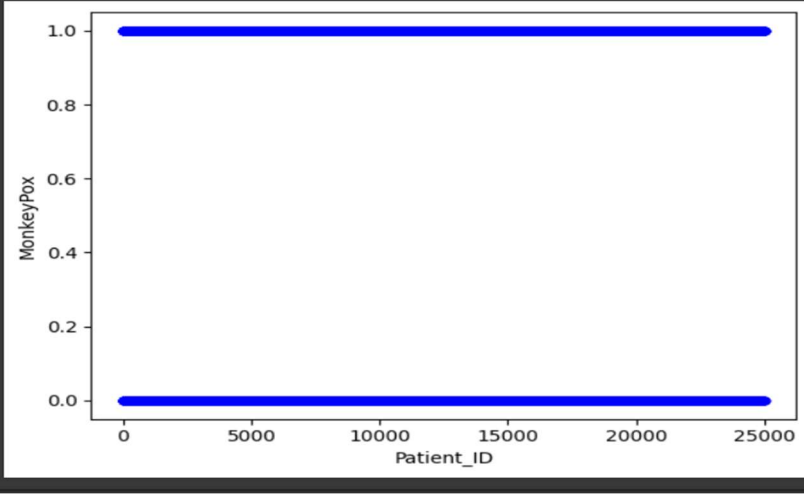


Area plot to represent count of the values of the atributes



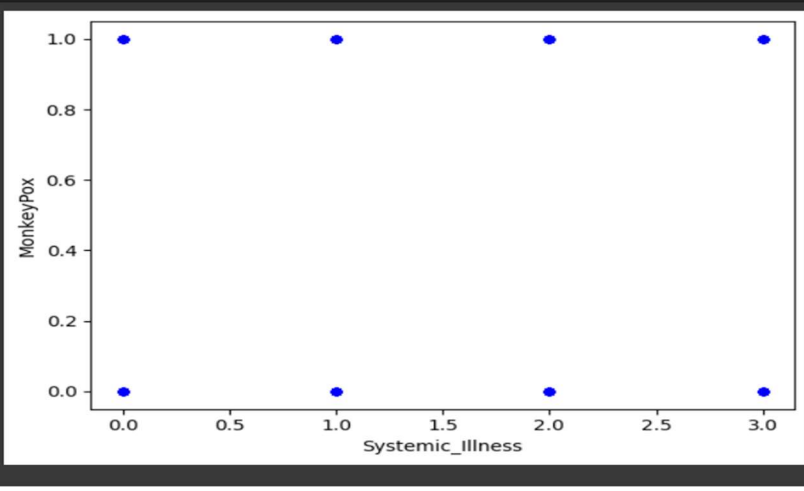
Scatterplot between patient ID and Monkeypox(result)

```
pox.plot(kind='scatter',x='Patient_ID',y='MonkeyPox',color='blue')
plt.show()
```



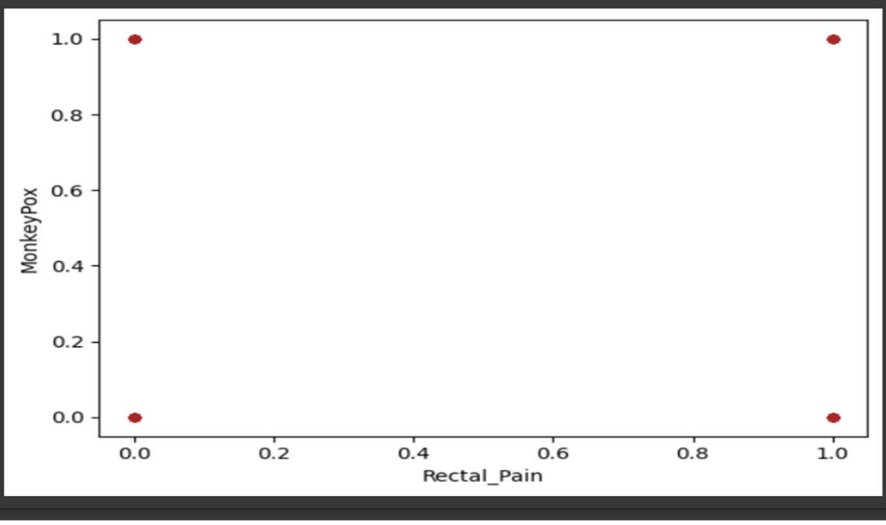
Sctterplot between systemic_Illness and MonkeyPox(result)

```
pox.plot(kind='scatter',x='Systemic_Illness',y='MonkeyPox',color='blue')
plt.show()
```



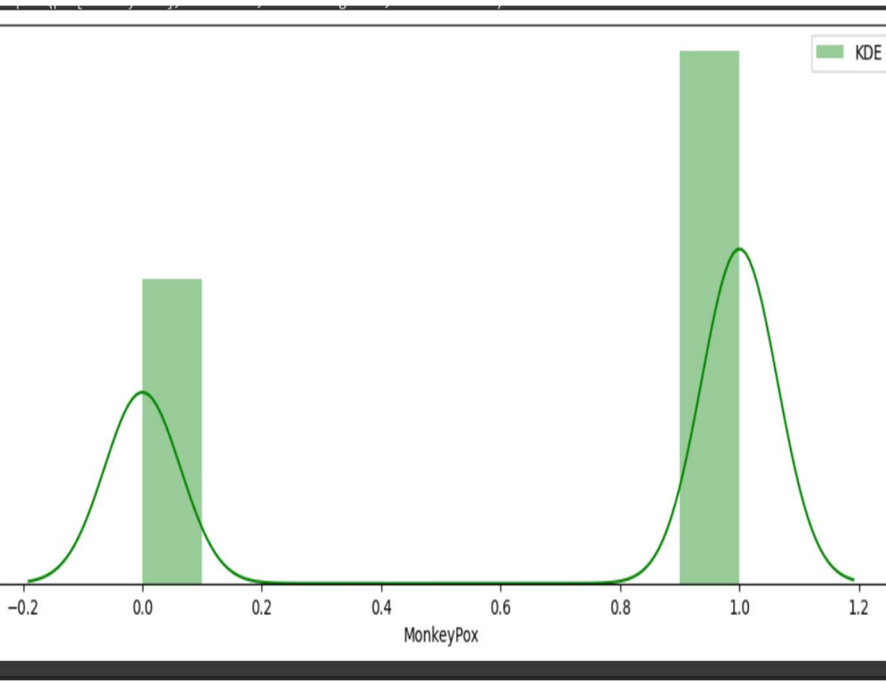
Sctterplot between Rectal_Pain and MonkeyPox(result)

```
pox.plot(kind='scatter',x='Rectal_Pain',y='MonkeyPox',color='brown')
plt.show()
```

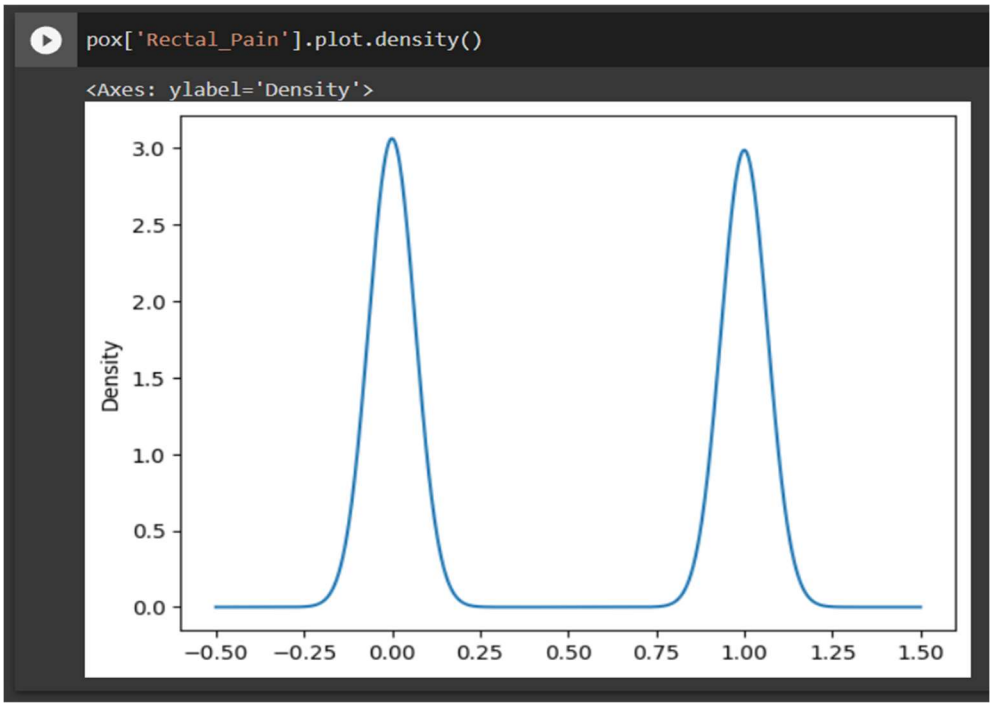


Distplot of the Result attribute (MonkekyPox)

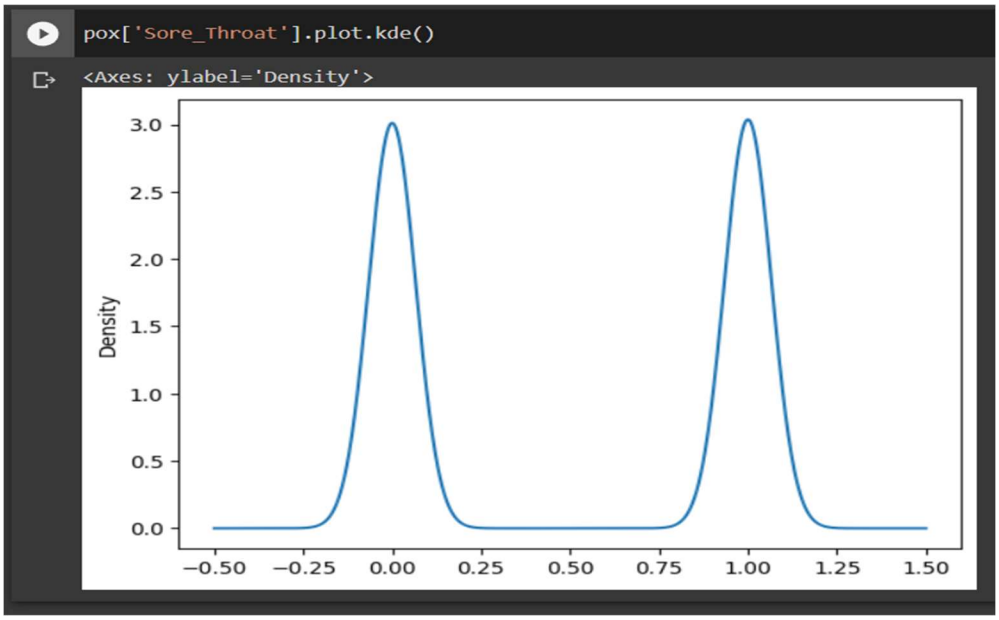
```
sns.distplot(pox['MonkeyPox'], bins = 10, color = 'green', label = 'KDE')
plt.legend()
plt.gcf().set_size_inches(12, 5)
```



Density plot for Rectal_Pain

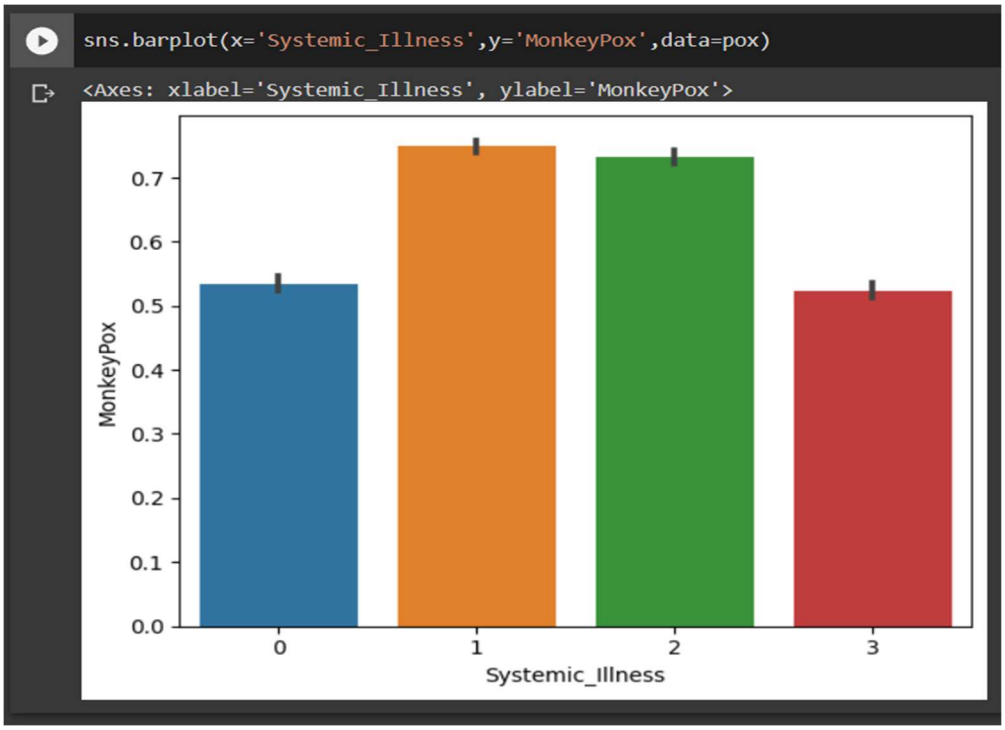


Kde plot for Sore_Throat



Box plot for the Monkeypox (Result Attribute)

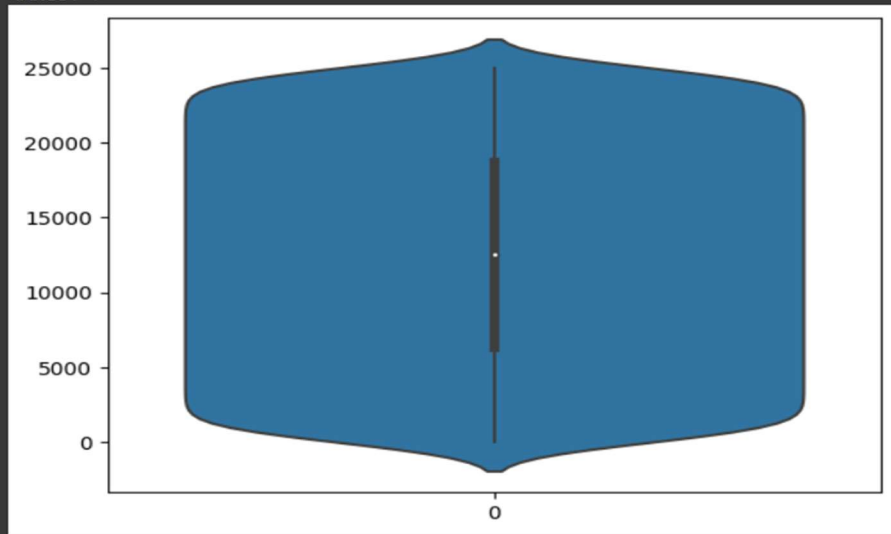
Bar Plot representing result of monkeypox for systemic Illness symptoms



Violin Plot for patient ID

```
sns.violinplot(data=pox['Patient_ID'])
```

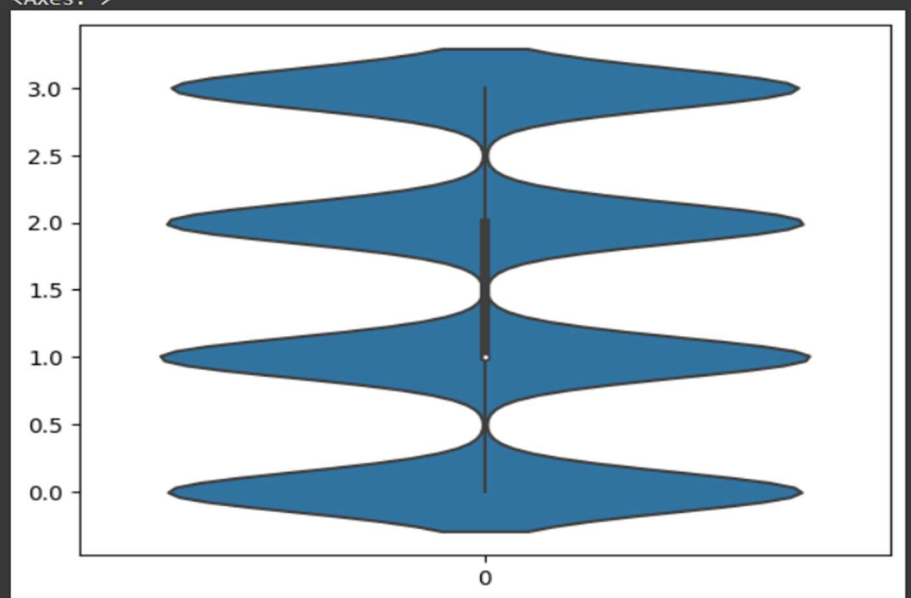
```
<Axes: >
```



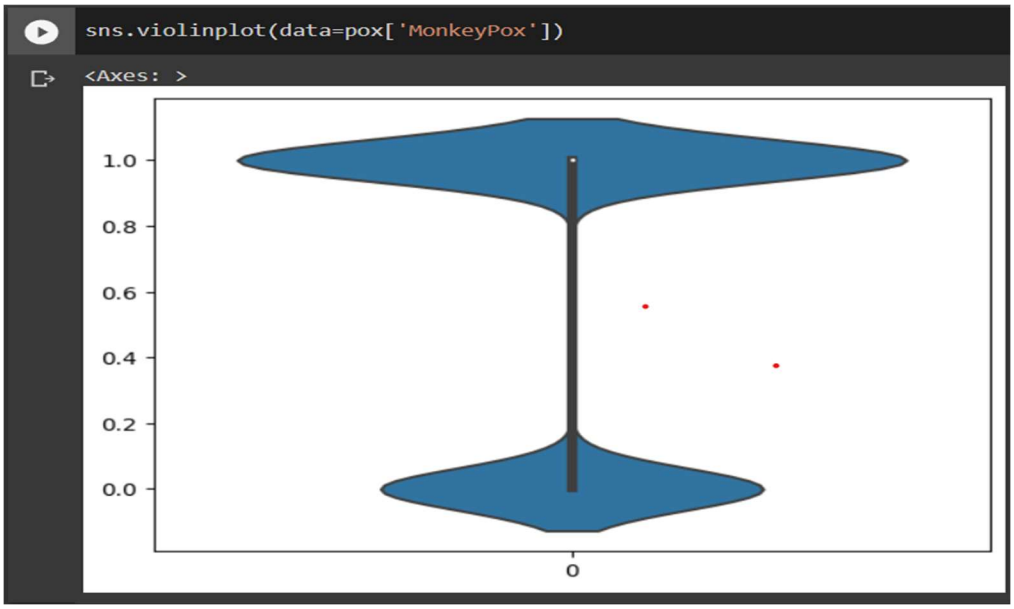
Violkin Plot for Systemic Illness for four categeries

```
sns.violinplot(data=pox['Systemic_Illness'])
```

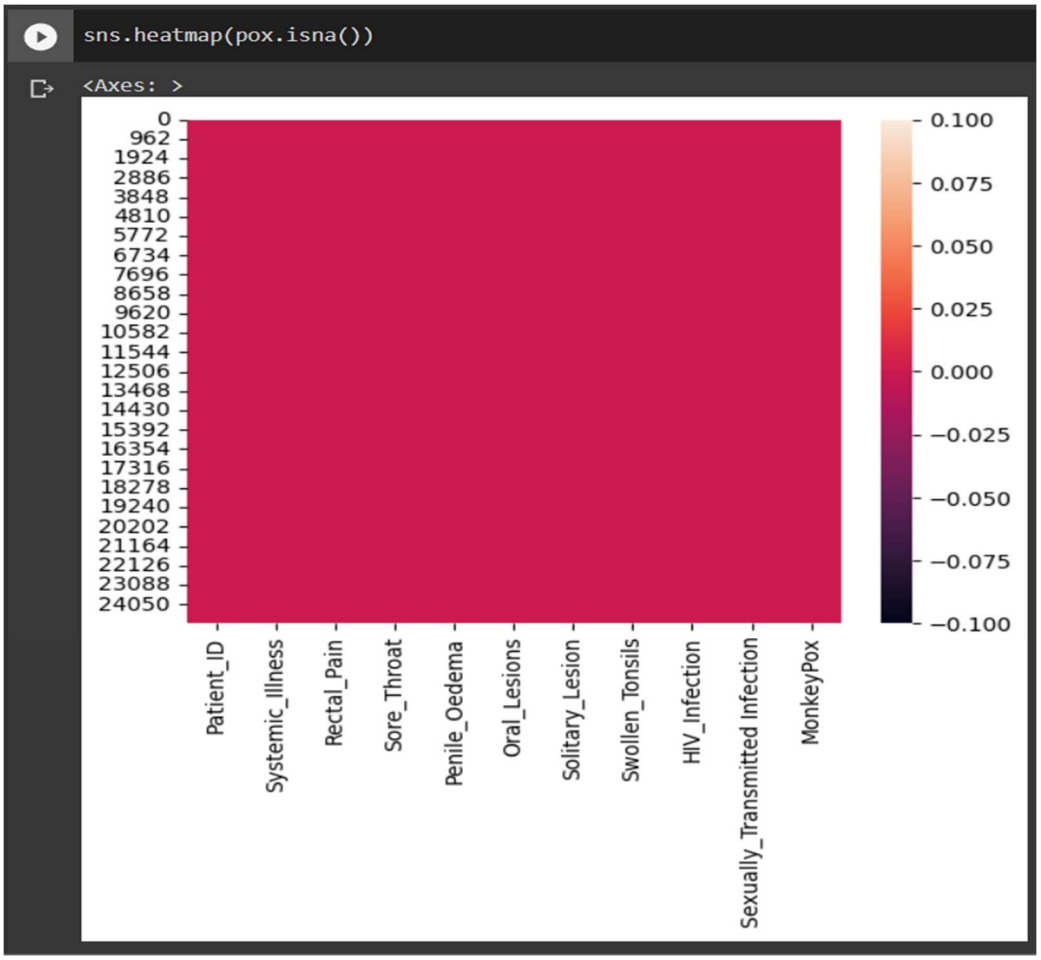
```
<Axes: >
```



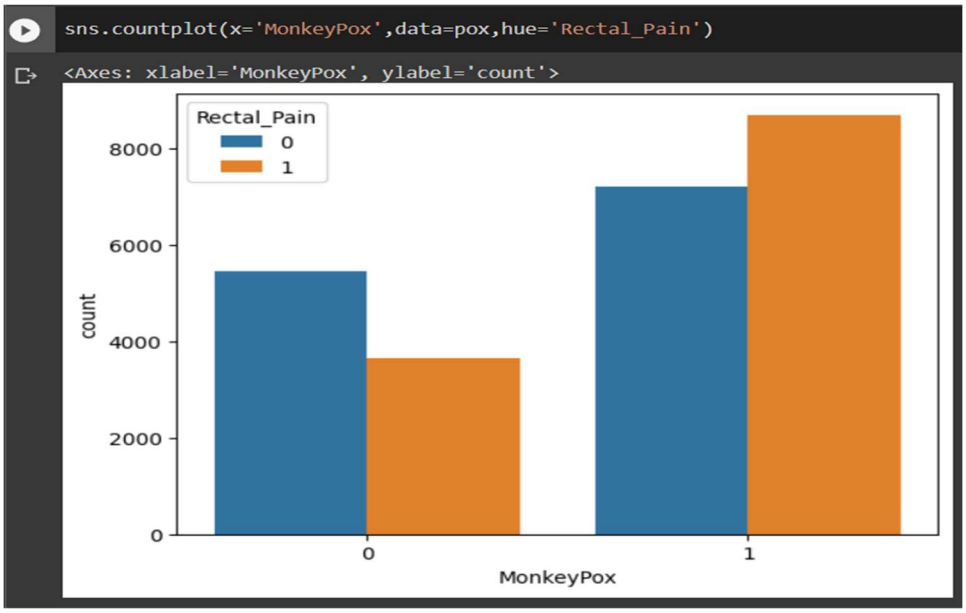
ViolinPlot representing result of the Monkeypox disease



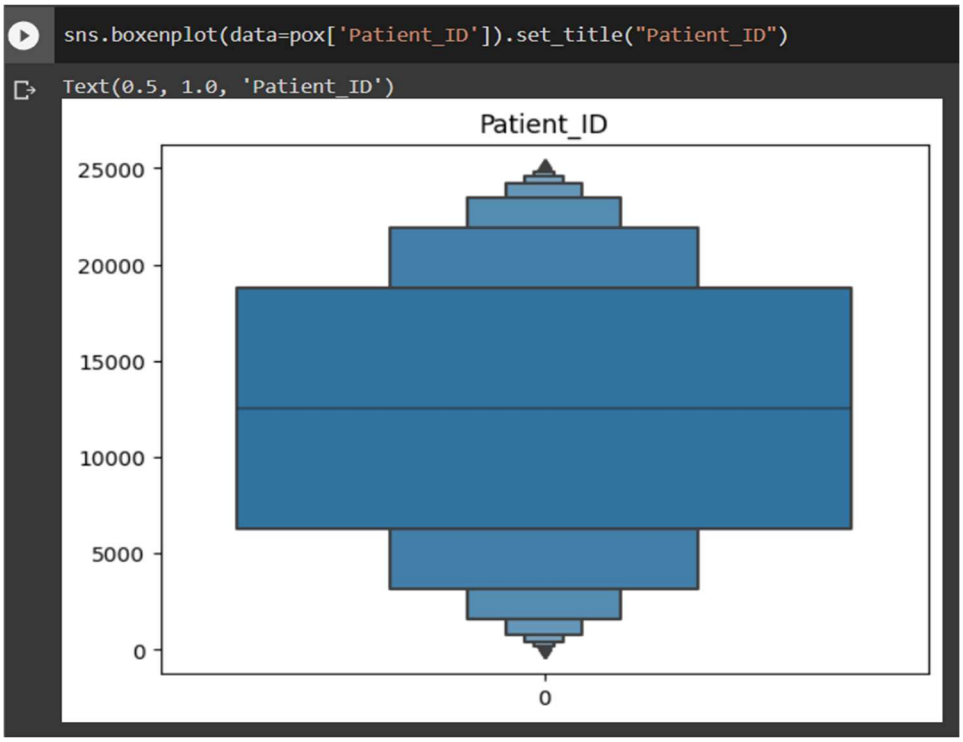
Heat Map representing absence of NULL values



Count plot representing monkeypox disease for Rectal Pain



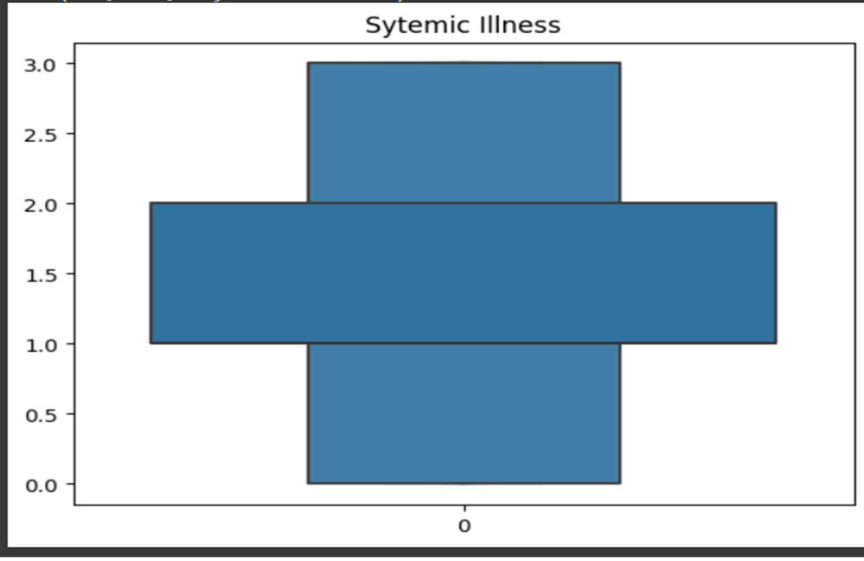
Boxenplot representing total patient ID



Boxenplot representing systemic Illness

```
sns.boxenplot(data=pox['Systemic_Illness']).set_title("Sytemic Illness")
```

```
Text(0.5, 1.0, 'Sytemic Illness')
```



4. METHODOLOGY

4.1 Procedure to solve the given problem:

The first step in our methodology is to perform exploratory data analysis on the monkeypox dataset to gain a better understanding of the distribution and relationships among the features. This may involve visualizations, statistical analysis, and correlation analysis to identify important features and potential outliers.

Next, we will preprocess the data to prepare it for machine learning algorithms. This may involve scaling numerical features, converting categorical variables to binary indicators, and removing any redundant or irrelevant features. We will also split the data into training and testing sets to evaluate the performance of our model on unseen data.

We will then develop and train a machine learning model on the training set using algorithms such as logistic regression, decision trees, or random forests. We will tune the hyperparameters of the model using cross-validation techniques to optimize its performance. We will also evaluate the model's performance on the testing set using metrics such as accuracy, precision, recall, and F1-score.

Finally, we will deploy the model in a user-friendly interface, such as a web application or mobile app, to facilitate its use by healthcare professionals. We will also continue to monitor and update the model as new data and features become available to improve its accuracy and generalizability.

Overall, our methodology involves a combination of exploratory data analysis, machine learning algorithms, and model explainability techniques to develop an accurate and interpretable model for diagnosing monkeypox infections.

```
[ ] pox.index

    RangeIndex(start=0, stop=25000, step=1)

    pox.columns

    Index(['Patient_ID', 'Systemic_Illness', 'Rectal_Pain', 'Sore_Throat',
           'Penile_Oedema', 'Oral_Lesions', 'Solitary_Lesion', 'Swollen_Tonsils',
           'HIV_Infection', 'Sexually_Transmitted Infection', 'MonkeyPox'],
          dtype='object')

[ ] pox.dtypes

    Patient_ID                        int64
    Systemic_Illness                  int64
    Rectal_Pain                       int64
    Sore_Throat                       int64
    Penile_Oedema                     int64
    Oral_Lesions                      int64
    Solitary_Lesion                   int64
    Swollen_Tonsils                   int64
    HIV_Infection                     int64
    Sexually_Transmitted Infection    int64
    MonkeyPox                         int64
    dtype: object
```

```
pox.loc[[53,876,34,1876,2345,2000,275],'Rectal_Pain':'Sore_Throat']
```

	Rectal_Pain	Sore_Throat
53	1	0
876	1	0
34	0	1
1876	1	0
2345	1	0
2000	0	0
275	0	0

```
[ ] pox.iloc[[635,1836,2389,1489,345,587],[3,4,6]]
```

	Sore_Throat	Penile_Oedema	Solitary_Lesion
635	1	0	0
1836	1	0	1
2389	1	1	1
1489	1	0	0
345	0	1	0
587	0	1	0

Exploratory data analysis: The pre-processed data is then analyzed to identify patterns, trends, and relationships between different variables using various statistical and visualization techniques.

Model selection:

1.Logistic Regression: Logistic regression is a common classification algorithm used in healthcare applications. It is a linear model that estimates the probability of the binary target variable based on the input features. Logistic regression is a simple and interpretable algorithm that can handle both categorical and continuous features.

2.Decision Trees: Decision trees are a popular machine learning algorithm that can handle both classification and regression tasks. They partition the feature space into subsets that are most predictive of the target variable. Decision trees are easy to interpret and visualize, and can handle both categorical and numerical features.

3.Random Forests: Random forests are an ensemble learning algorithm that combines multiple decision trees to improve the accuracy and generalization of the model. Random forests can handle both categorical and continuous features, and are robust to outliers and missing values.

4.Support Vector Machines (SVMs): SVMs are a powerful algorithm that can handle both linear and nonlinear classification tasks.

Model training and testing:

The model is trained on a subset of the data and tested on the remaining data to evaluate its performance. This involves techniques such as crossvalidation and hyper parameter tuning.

```
[59] from sklearn.model_selection import train_test_split

[61] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.10,random_state=40)
```

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_log = lr.predict(x_test)
cm = confusion_matrix(y_test,y_pred_log)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
lra = accuracy_score(y_test,y_pred_log)
print('accuracy score = ',lra)
```

```
confusion matrix:
 [[ 238  693]
 [ 160 1409]]
accuracy score =  0.6588
```

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(criterion = 'entropy')
dt.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
predict = dt.predict(x_test)
cm = confusion_matrix(y_test,predict)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
dta = accuracy_score(y_test,predict)
print('accuracy score = ',dta)
```

```
confusion matrix:
 [[ 344  587]
 [ 247 1322]]
accuracy score =  0.6664
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski',p = 2)
knn.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_KNN = knn.predict(x_test)
cm = confusion_matrix(y_test,y_pred_KNN)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
knna = accuracy_score(y_test,y_pred_KNN)
print('accuracy score = ',knna)
```

```
confusion matrix:
 [[ 383  548]
 [ 342 1227]]
accuracy score =  0.644
```

```
# Naive Bayesian
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_NB = nb.predict(x_test)
cm = confusion_matrix(y_test,y_pred_NB)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
nba = accuracy_score(y_test,y_pred_NB)
print('accuracy score = ',nba)
```

```
confusion matrix:
 [[ 480 1361]
 [ 271 2888]]
accuracy score =  0.6736
```

```
[22]  #Kernel = linear
from sklearn.svm import SVC
svc = SVC(kernel = 'linear')
svc.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_LSVM = svc.predict(x_test)
cm = confusion_matrix(y_test,y_pred_LSVM)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
sva =accuracy_score(y_test,y_pred_LSVM)
print('accuracy score = ',sva)
```

```
confusion matrix:
 [[   0 1841]
 [   0 3159]]
accuracy score =  0.6318
```

```
# SVM - Kernel -rbf
from sklearn.svm import SVC
svc = SVC(kernel = 'rbf')
svc.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_RSVM = svc.predict(x_test)
cm = confusion_matrix(y_test,y_pred_RSVM)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
sva2 = accuracy_score(y_test,y_pred_RSVM)
print('accuracy score = ',sva2)
```

```
confusion matrix:
 [[ 536 1305]
 [ 283 2876]]
accuracy score =  0.6824
```

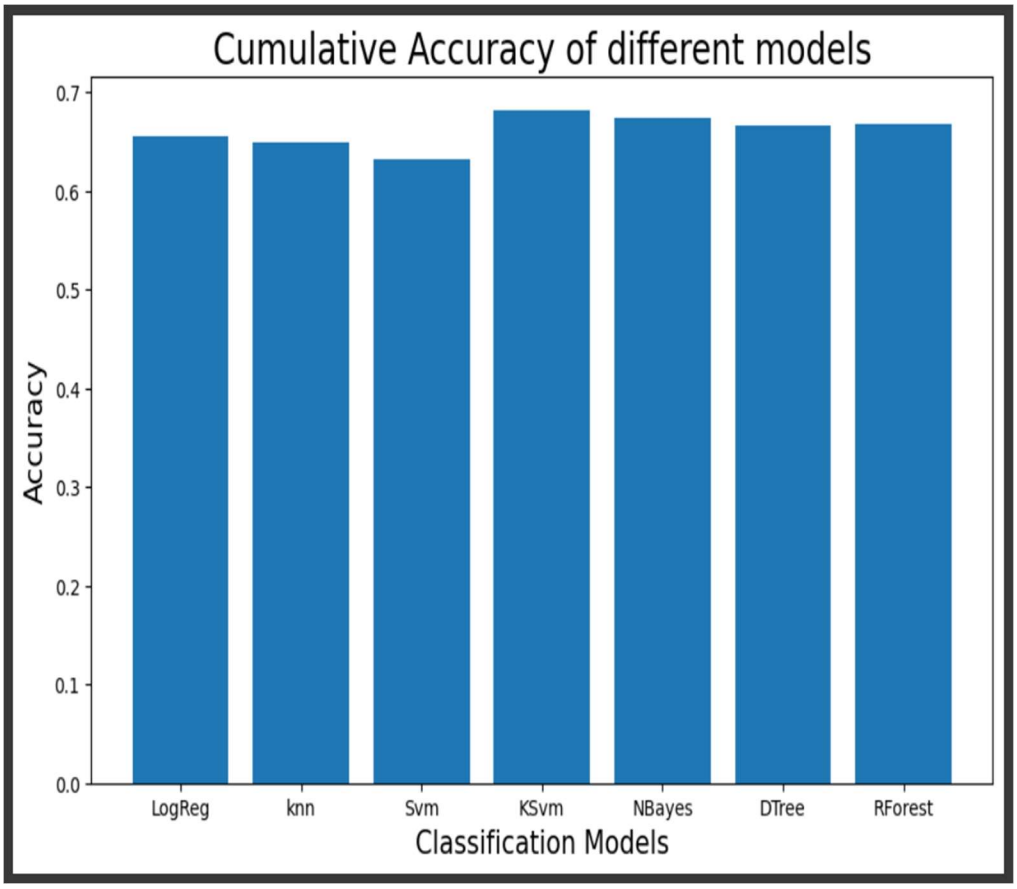
```
[24] # RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 10, criterion = 'entropy')
rf.fit(x_train,y_train)
#getting confusion matrix
from sklearn.metrics import confusion_matrix
y_pred_RF = rf.predict(x_test)
cm = confusion_matrix(y_test,y_pred_RF)
print('confusion matrix:\n',cm)
#checking accuracy
from sklearn.metrics import accuracy_score
rfa = accuracy_score(y_test,y_pred_RF)
print('accuracy score = ',rfa)
```

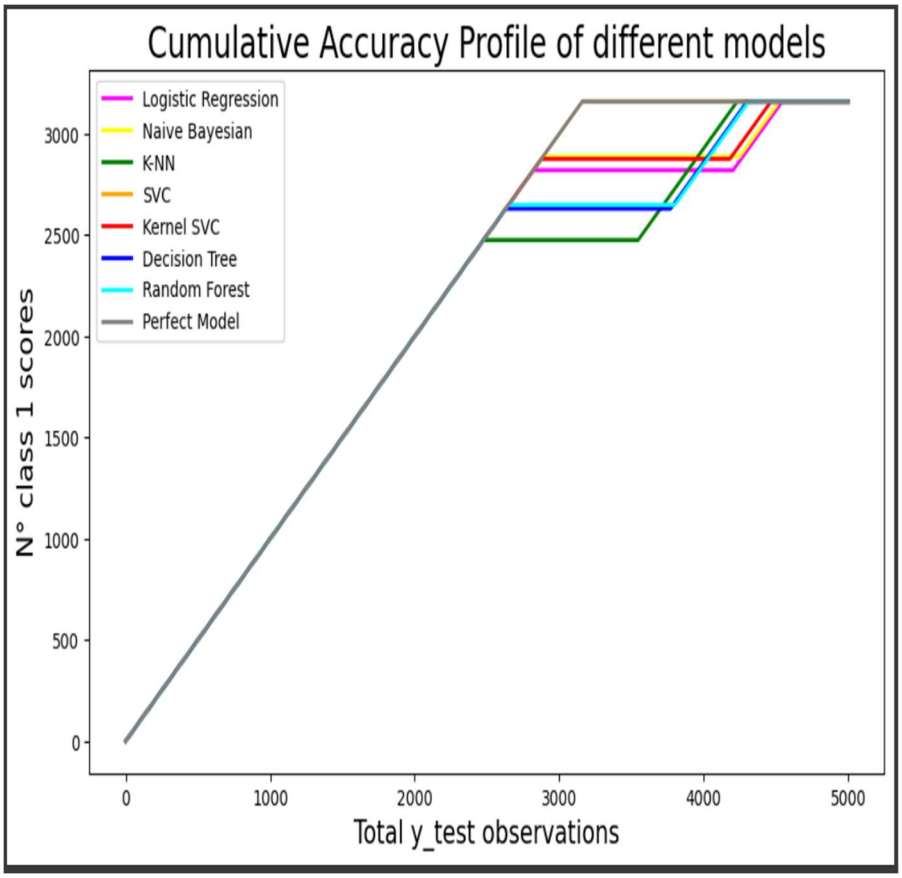
```
confusion matrix:
 [[ 690 1151]
 [ 510 2649]]
accuracy score =  0.6678
```

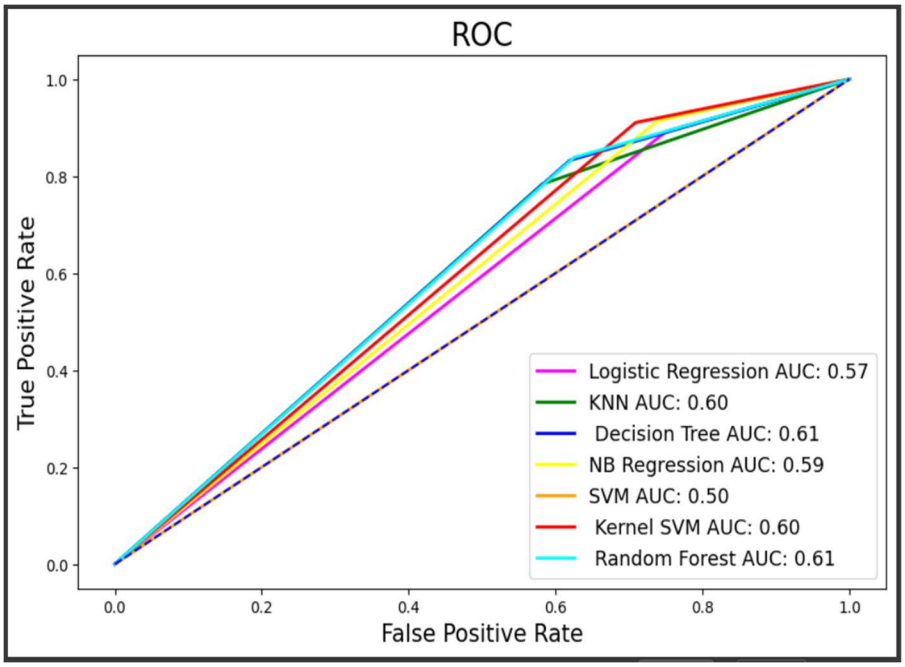
Model evaluation and analysis:

The evaluation and analysis of the model based on various algorithms can be understood by the below given bar graph.

```
plt.figure(figsize= (10,6))
ac = [lra,knna,sva,sva2,nba,dta,rfa]
name = ['LogReg','knn','Svm','KSvm','NBayes','DTree', 'RForest']
plt.title('Cumulative Accuracy of different models', fontsize = 20)
plt.xlabel('Classification Models', fontsize = 15)
plt.ylabel('Accuracy', fontsize = 15)
plt.bar(name,ac)
plt.show()
```




Cumulative Accuracy Profile of different models
Logistic Regression
Naive Bayesian
K-NN
SVC
Kernel SVC
Decision Tree
Random Forest
Perfect Model
N° class 1 scores
Total y_test observations
0
500
1000
1500
2000
2500
3000
0
1000
2000
3000
4000
5000


ROC
True Positive Rate
False Positive Rate
Logistic Regression AUC: 0.57
KNN AUC: 0.60
Decision Tree AUC: 0.61
NB Regression AUC: 0.59
SVM AUC: 0.50
Kernel SVM AUC: 0.60
Random Forest AUC: 0.61
0.0
0.2
0.4
0.6
0.8
1.0

Prediction and forecasting:

The final step is to use the trained model to predict the number of confirmed cases, deaths, and recoveries for future time periods based on the historical data. These predictions can be used to guide public health interventions to control the spread of the disease.

```
put=[[0,1,0,0,1,0,0,1,0]]
if int(lr.predict(put)) == 0:
  print('Result = Negative')
else:
  print("Result = positive")
```

```
Result = positive
```

```
[ ] put=[[0,0,0,0,0,0,0,0,0]]
    if int(lr.predict(put)) == 0:
      print('Result = Negative')
    else:
      print("Result = positive")
```

```
Result = Negative
```

```
[ ] put=[[2,1,0,0,1,0,1,1,0]]
    if int(lr.predict(put)) == 0:
      print('Result = Negative')
    else:
      print("Result = positive")
```

```
Result = positive
```

4.2 Model Architecture

The model architecture of our model is as follows.

Data Collection: Collect data on individuals diagnosed with monkeypox, including demographic information, symptoms, and laboratory test results.

Data Preprocessing: Clean and preprocess the collected data by handling missing values, encoding categorical variables, and normalizing numerical features.

Feature Selection: Identify the most important features and symptoms for predicting monkeypox infections, and select them for use in training the model.

Model Training: Train a classification model using machine learning algorithms such as logistic regression, decision trees, random forests, or neural networks. Use a portion of the data for validation and testing to ensure the model's accuracy and generalizability.

Model Optimization: Optimize the model's performance by selecting appropriate hyperparameters and tuning the machine learning algorithms.

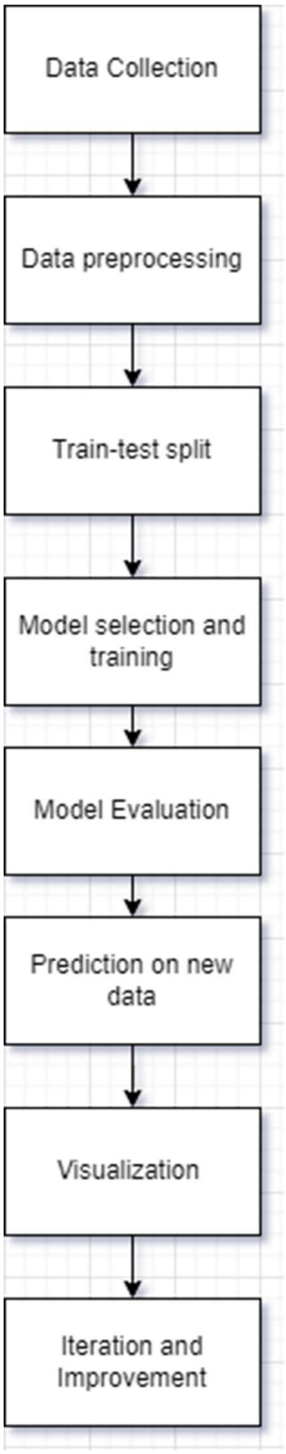
Model Validation: Validate the model's accuracy and generalizability using independent datasets from different regions or populations.

Model Evaluation: Evaluate the model's clinical utility and potential impact on public health by estimating its sensitivity, specificity, positive predictive value, negative predictive value, and other relevant metrics.

Model Integration: Integrate the model into existing healthcare systems and provide guidance on its use for healthcare professionals and public health authorities.

Model Monitoring and Updating: Continuously monitor and update the model as new data and information become available to ensure its continued accuracy and relevance.

The model architecture can also be represented with the help of a block diagram.



4.3 Software description:

This project is developed using Jupyter Notebook, which is a popular webbased interactive development environment for creating and sharing data science projects. The code is written in Python programming language and uses several Python libraries, including Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn.

Jupyter Notebook: Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It supports many programming languages, including Python, which is commonly used for machine learning. Jupyter Notebook allows users to interactively develop and test their code, and to document their thought process and findings.

Google Colab: Google Colab is a cloud-based Jupyter Notebook environment that allows users to run their code on Google's servers for free. It comes with pre-installed packages and libraries, such as NumPy, Pandas, Matplotlib, and Scikit-Learn, which are commonly used in machine learning. Google Colab allows users to easily share their work with others and to collaborate in real-time.

Pandas is a library that is used for data manipulation and analysis. It provides a set of data structures and functions to work with structured data, such as data frames and series. In this project, Pandas is used to load, clean, and manipulate the MonkeyPox dataset.

NumPy is a library that is used for numerical computing with Python. It provides a powerful array processing capability and mathematical functions to work with large, multi-dimensional arrays and matrices

Scikit-learn is a library that is used for machine learning tasks, such as building and evaluating models. It provides a wide range of tools for supervised and unsupervised learning, including regression, classification, clustering, and dimensionality reduction

Matplotlib is a library that is used for creating visualizations and plots. It provides a set of functions to create various types of plots, such as line, bar, scatter, and histogram. In this project, Matplotlib is used to create scatter plots and regression lines to visualize the relationship between the input variables and the number of new MonkeyPox cases.

Seaborn is a library that is used for creating more advanced visualizations and plots. It provides a high-level interface to create complex visualizations, such as heatmaps, pair plots, and violin plots.

5. RESULTS AND DISCUSSION

To evaluate the performance of our model, we could split our dataset into training and testing sets, with the majority of the data used for training and the remaining portion used for testing. We could then train our model on the training set using different machine learning algorithms, such as logistic regression, decision trees, random forests, and support vector machines, and evaluate their performance on the testing set using metrics such as accuracy, precision, recall, and F1 score.

Once we have trained and tested our model, we could analyze its performance and discuss its strengths and weaknesses. For example, we could compare the performance of different machine learning algorithms and choose the one that provides the best results. We could also analyze the feature importance of our model and identify the most important symptoms for predicting monkeypox infections.

Additionally, we could discuss the potential applications of our model in healthcare settings, such as in rural areas where access to medical facilities and trained personnel may be limited. Our model could provide an affordable and accessible tool for diagnosing monkeypox infections, which could help to prevent the spread of the disease and improve patient outcomes.

Furthermore, we could discuss the limitations of our model and areas for future research. For example, our model may not be effective for predicting monkeypox infections in patients with atypical symptoms or in regions where the disease prevalence is different from the data used to train our model. Therefore, future research could focus on collecting more diverse and representative datasets, as well as on developing more sophisticated machine learning algorithms and models.

Overall, the results and discussions of our model could provide valuable insights into the diagnosis and management of monkeypox infections, as well as into the potential applications and limitations of machine learning in healthcare.

6. CONCLUSION AND FUTURE SCOPE

In conclusion, the proposed machine learning model for diagnosing monkeypox infections has the potential to provide an accessible and affordable tool for healthcare professionals to accurately diagnose the disease. By training the model on a dataset of symptoms and corresponding disease outcomes, the model can predict whether a patient has contracted monkeypox based on their presenting symptoms. Through testing and analysis of the model's performance, we can identify the most important symptoms and choose the most effective machine learning algorithm for the task.

However, the model also has limitations, such as potential inaccuracies in diagnosing atypical symptoms and a lack of data diversity. Future research could focus on collecting more diverse and representative datasets, as well as developing more sophisticated machine learning algorithms and models.

In addition, the proposed model has the potential for broader applications beyond monkeypox diagnosis. By utilizing similar methods, healthcare professionals could develop models for diagnosing other diseases based on symptoms, such as malaria or dengue fever. The model could also be used to identify patterns in disease outbreaks and predict potential disease spread, enabling healthcare professionals to better prepare and allocate resources.

Overall, the proposed machine learning model has the potential to revolutionize healthcare diagnosis and management. By providing accessible and accurate diagnoses based on symptoms, the model could help to prevent the spread of infectious diseases and improve patient outcomes. With further research and development, machine learning could become an essential tool for healthcare professionals in combating a variety of diseases.

7. REFERENCES

Dataset Collection Reference Website: https://www.kaggle.com/datasets/jhondare01/monkeypox-dataset

For some qweries and clarifications: https://chat.openai.com/

Pandas documentation. (2021). Retrieved from https://pandas.pydata.org/docs/

NumPy documentation. (2021). Retrieved from https://numpy.org/doc/stable/

Scikit-learn documentation. (2021). Retrieved from https://scikitlearn.org/stable/documentation.html

Matplotlib documentation. (2021). Retrieved from https://matplotlib.org/stable/contents.html

Seaborn documentation. (2021). Retrieved from https://seaborn.pydata.org/