Analysis of Monkey Disease Prediction Using Machine Learning Techniques

line 1: 2nd Given Name Surname
line 2: *dept. name of organization (of Affiliation)*
line 3: *name of organization (of Affiliation)*
line 4: City, Country
line 5: email address or ORCID

***Abstract*— Nowadays, we cannot understand when people will die due to any type of disease. This is not only because of proper treatment but also due to unhealthy food, tension, and depression. Monkey pox is also one of the diseases identified in May 2022 which is just similar to chickenpox. this disease is defined as a public health emergency of international concern by WHO. Risk factors, transmissions, infection outcomes, and clinical presentations are not understood properly. So, to find the proper solution, machine learning prediction will be very helpful for controlling its spread. The main objective of this research is to perform machine learning techniques to identify someone has monkey pox disease or not. Data Cleaning and preprocessing are going to be applied and performed. Machine learning techniques such as logistic regression, xg boost classification, and random forest classification will also be applied to predict monkey pox disease.**

I. INTRODUCTION

This world faced so many health emergencies such as COVID, monkey pox, and many more, This monkey pox disease emerged as a global threat and spread to more than 100 non-endemic countries. Symptoms of monkey pox are kind of similar to chickenpox. This monkey pox disease was confirmed in May 2022. Initially, a cluster of cases was found in the United Kingdom, but the first case was found in London. In monkey pox, one can contract by contact with disease animals or individuals. This transmission occurs due to mucus from the nose or mouth or through animal bites, direct contact with skin, or respiratory droplets. Some short-term symptoms include fever, fatigue, and aches and long-term symptoms contain a red lump on the skin. In the initial stage, it is very tough to detect monkey pox because most of the symptoms are similar to chicken pox. It was very tough to detect in humans because most of the medical experts lacked of necessary knowledge to diagnose the patients. So after they identify the solution that is isolation and avoiding the physical contact to make it easier to decrease the spread of this disease.. As we know, artificial intelligence and machine learning techniques are growing in the real world whether it is a department of automobile or healthcare as this field provides us a better reliability, accuracy, and better solutions output based on the symptoms. The main aim of this analysis is to apply machine learning techniques to identify whether that person has a monkey pox or not. To perform the analysis, it is very important to collect the data from proper sources so here we collect the data from Kaggle. This dataset contains some symptoms that as rectal pain, systemic illness, sore throat, penile edema, oral lesions, solitary lesions, swollen tonsils, HIV infection, sexually transmitted, and whether someone has monkey pox or not. Data cleaning is also a very important part of cleaning the data and finding out the useful symptoms. Machine Learning techniques are applied to find the accurate scorecard features to implement the proper analysis and identify the monkey pox.

II. METHODS AND TECHNIQUES USED

Methods

A. *Data Collection*

As we all know, data collection is the initial step of data analysis to implement the techniques of machine learning and artificial intelligence. To perform the analysis of monkey pox disease, it is important to collect the relevant information or data. So, we have collected the monkey pox disease dataset from the Kaggle data source. After downloading the dataset, we use the Python libraries that are pandas to read the dataset in a tabular format. We use the read_csv function to read the dataset and use the head function to show the top 5 rows.

Data Introduction

```
In [1]: #Read Dataset
        import pandas as pd
        df=pd.read_csv("DATA.csv")
        df.head()
```

Out[1]:

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection
0	P0	None	False	True	True	True	False	True	False	False
1	P1	Fever	True	False	True	True	False	False	True	False
2	P2	Fever	False	True	True	False	False	False	True	False
3	P3	None	True	False	False	False	True	True	True	False
4	P4	Swollen Lymph Nodes	True	True	True	False	False	True	True	False

Figure 1: Monkey Pox Dataset

The dataset contains various columns such as patient ID, systemic illness, rectal pain, sore throat, penile enema, oral lesions, solitary lesions, swollen tonsils, HIV infection, sexually transmitted infection, and monkey pox.

```
In [73]: #Column Name of Dataset
         df.columns

Out[73]: 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')
```

Figure 2: Column Name

We checked the dimensions of the dataset and there were 25000 patient data is there to perform the analysis and identify monkey pox on the given symptoms.

```
In [3]: #dimension of dataset
        df.shape

Out[3]: (25000, 11)
```

Figure 3: Shape of Data

We have checked the data type of columns present in the monkey pox dataset. Some columns are of object type while some are of Boolean type.

```
In [72]: #ypes of dataset
         df.dtypes

Out[72]: Patient_ID                        object
         Systemic Illness                  object
         Rectal Pain                         bool
         Sore Throat                         bool
         Penile Oedema                       bool
         Oral Lesions                        bool
         Solitary Lesion                     bool
         Swollen Tonsils                     bool
         HIV Infection                       bool
         Sexually Transmitted Infection      bool
         MonkeyPox                         object
         dtype: object
```

Figure 4: Type of Data

B. Data Cleaning and Preprocessing

Data cleaning and Preprocessing is the second step of the data analysis in which we tried to remove the unwanted data, missing data, and duplicate values and perform data transformation as well. Here, we use the is null function to check the missing rows but there are no missing values in our dataset. We analysed that data is totally useful and we can perform further steps.

Check Missing Values

```
In [74]: #check missing values
         df.isnull().sum()

Out[74]: 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
```

Figure 5: Figure 5: Missing Values

As we noticed at the time of data checking, some columns are in object and Boolean type so before applying the models, we need to convert them into numerical values. So, we applied a data preprocessing technique that is label encoding which uses the new Python library that is sklearn which can import the label encoder to convert the data into numerical values.

Data Processing

```
[79]: #Label Encoding

      from sklearn.preprocessing import LabelEncoder
      data2[['Systemic Illness', 'Rectal Pain', 'Sore Throat',
             'Penile Oedema', 'Oral Lesions', 'Solitary Lesion', 'Swollen Tonsils',
             'HIV Infection', 'Sexually Transmitted Infection', 'MonkeyPox']] = data2[
             'Penile Oedema', 'Oral Lesions', 'Solitary Lesion', 'Swollen Tonsils',
             'HIV Infection', 'Sexually Transmitted Infection', 'MonkeyPox']].apply(La

      C:\Users\SR\anaconda3\lib\site-packages\pandas\core\frame.py:3065: SettingWithC
      opyWarning:
      A value is trying to be set on a copy of a slice from a DataFrame.
      Try using .loc[row_indexer,col_indexer] = value instead

      See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
      ble/user_guide/indexing.html#returning-a-view-versus-a-copy
        self[k1] = value[k2]

[81]: data2.head()
```

[81]:

	Patient_ID	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection
1	P1	0	1	0	1	1	0	0	1	0
2	P2	0	0	1	1	0	0	0	1	0
4	P4	2	1	1	1	0	0	1	1	0
5	P5	2	0	1	0	0	0	0	0	0

Figure 6: Label Encoding

Patient ID is also a unique value that cannot play an important role in applying the model and improving its performance. So we have dropped the patient ID using the drop function.

```
2]: data=data2.drop(['Patient_ID'],axis=1)
    data.head()
```

2]:

	Systemic Illness	Rectal Pain	Sore Throat	Penile Oedema	Oral Lesions	Solitary Lesion	Swollen Tonsils	HIV Infection	Sexually Transmitted Infection	MonkeyPo
1	0	1	0	1	1	0	0	1	0	
2	0	0	1	1	0	0	0	1	0	
4	2	1	1	1	0	0	1	1	0	
5	2	0	1	0	0	0	0	0	0	
6	0	0	1	0	0	0	0	1	0	

Figure 7: Drop Column

C. Data Visualization

Data Visualization is the third step of data analysis in which we plot the graphs and charts to represent the information of data in graphical format. The main purpose of plotting the graphs is to understand the pictorial images which are easier to recognize and understand easily. Using data visualization, we can plot graphs which is helpful in decision-making as well because graphical information is more understandable than theoretical information. We use the Seaborn and matplotlib libraries of Python to visualize the information attractively.

We have plotted the count plot of the monkey pox column in which we will count the positive and negative results of monkey pox and will show in a bar graph. So, after plotting it, we saw that approximately 15000 people are infected by monkey pox while approximately 8000 people are not infected.

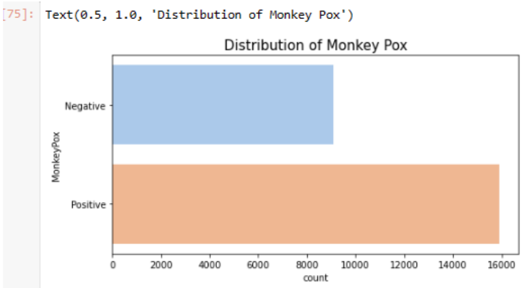

Figure 8: Distribution of Monkey Pox

The next graph we plotted is a KDE plot in which we represent the density of rectal pain on the whole dataset.

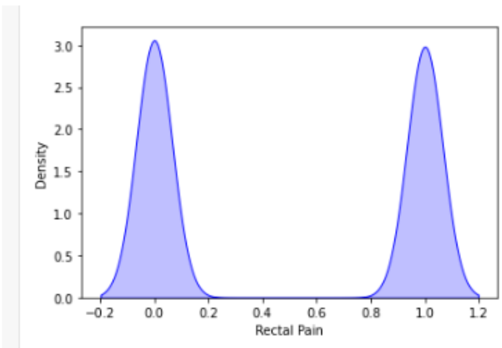

Figure 9: KDE Plot of Rectal Pain

Another graph is also a KDE plot on solitary lesions but here it is comparing data concerning the monkey pox. This graph represents the density of positive and negative values with the solitary lesion. The results that the density of positive results is greater than negative results.

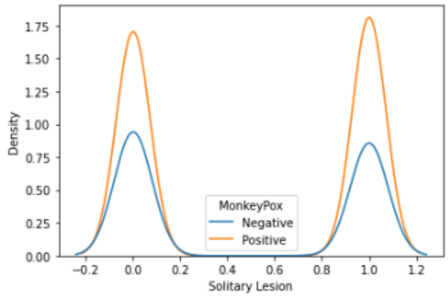

Figure 10: KDE Plot of Solitary Lesion

Another graph is also a KDE plot on the sore throat but here it is comparing data concerning the monkey pox. This graph represents the density of positive and negative values with a sore throat. The results that the density of positive results is greater than negative results.

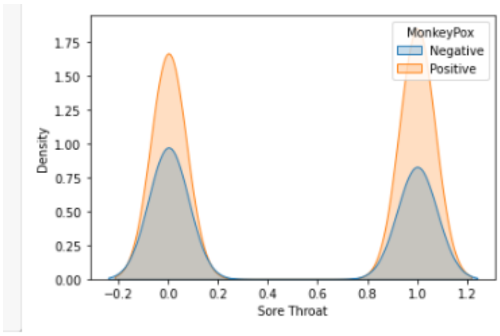

Figure 11: KDE Plot of Sore throat

D. Model Evaluation

After performing the cleaning and visualization, it's time to apply the classification models to predict the monkey pox disease. We take the monkey pox column as a dependent column while remaining as an independent variable. We split the data into train and test parts. 33% is taken for testing purposes while the remaining is taken for training purposes. Standardization is also applied to standardize the split data into a more useful format.

Train and Test Data

```
In [83]: #Data Splitting
x = data.drop(['MonkeyPox'],axis=1)
y = data['MonkeyPox']

In [84]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train , y_test = train_test_split(x,y, test_size = 0.20, rand
```

Figure 12: Train and Test Data

standardization

```
In [85]: from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
```

Figure 13: Standard Scaler

We have applied 3 machine learning techniques. But before selecting the models, need to understand dependent variable is categorical or continuous. So Data is discrete and we are going to apply the classification models.

Techniques

A. Logistic Regression

Logistic Regression is one of the classification machine learning models that works with binary data. This model predicts the probability of occurrence of an event based on other independent variables. This model used maximum likelihood estimation or gradient descent to calculate the optimal results. One of the advantages of the model is easy to implement because it demands maximum computational power. Also helpful for linearly separable data and monkey pox data is a linearly separable dataset. This model also provides valuable insights.

Logistic Regression

```
In [86]: from sklearn.linear_model import LogisticRegression
         classifier = LogisticRegression(random_state = 0)
         classifier.fit(x_train, y_train)
         y_pred = classifier.predict(x_test)

In [87]: from sklearn.metrics import confusion_matrix,accuracy_score
         cm = confusion_matrix(y_test, y_pred)
         acc=accuracy_score(y_test,y_pred)
         print("Accuracy Score:- ",acc)
         print ("Confusion Matrix : \n", cm)

         Accuracy Score:-  0.7000266169816343
         Confusion Matrix :
          [[ 202  971]
          [ 156 2428]]
```

Figure 14: Logistic Regression

B. XG Boost Classification

XG Boost is an extreme gradient boosting algorithm that is used for regression and classification techniques. This model is famous for better accuracy and scalability. We decided to use this model due to its high performance. It can easily handle the missing data and helps in preprocessing. XGBClassifier is used here to perform the prediction.

XG Boost Classifier

```
n [88]: from xgboost import XGBClassifier
        model = XGBClassifier()
        model.fit(x_train , y_train)
        y_pred2 = model.predict(x_test)

        C:\Users\SR\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: T
        he use of label encoder in XGBClassifier is deprecated and will be removed in a
        future release. To remove this warning, do the following: 1) Pass option use_la
        bel_encoder=False when constructing XGBClassifier object; and 2) Encode your la
        bels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
          warnings.warn(label_encoder_deprecation_msg, UserWarning)

        [14:40:38] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.
        1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric
        used with the objective 'binary:logistic' was changed from 'error' to 'loglos
        s'. Explicitly set eval_metric if you'd like to restore the old behavior.

n [89]: from sklearn.metrics import confusion_matrix,accuracy_score
        cm2 = confusion_matrix(y_test, y_pred2)
        acc2=accuracy_score(y_test,y_pred2)
        print("Accuracy Score:- ",(acc2*100))
        print ("Confusion Matrix : \n", cm2)

        Accuracy Score:-  70.05589566143199
        Confusion Matrix :
         [[ 349  824]
         [ 301 2283]]
```

Figure 15: XG Boost

C. Random Forest Classification

Random forest classification is also one of the classification techniques of supervised learning. The random forest model is highly efficient and provides more accurate results. This model is useful for linear datasets.

Random Forest Classification

```
[90]: from sklearn.ensemble import RandomForestClassifier
      classifier= RandomForestClassifier(n_estimators= 10, criterion="entropy")
      classifier.fit(x_train , y_train)
      y_pred3 = classifier.predict(x_test)

[91]: from sklearn.metrics import confusion_matrix,accuracy_score
      cm3 = confusion_matrix(y_test, y_pred3)
      acc3=accuracy_score(y_test,y_pred3)
      print("Accuracy Score:- ",(acc3*100))
      print ("Confusion Matrix : \n", cm3)

      Accuracy Score:-  68.59196167154644
      Confusion Matrix :
       [[ 348  825]
       [ 355 2229]]
```

Figure 16: Random Forest

III. CONCLUSION

Monkey pox is a very harmful disease which is identified in May 2022. According to earlier research, its symptoms are mostly similar to chickenpox but after that, they find out the other symptoms as well which explains that this is monkey pox. In this research analysis, we collected that data only and performed predictions to identify if this person was infected by monkey pox. Data preprocessing is also performed and the data properly using Seaborn and matplotlib libraries. Various machine learning techniques are applied for the prediction such as logistic regression, random forest regression, and xg boost classification, and calculate the accuracy score. According to the prediction scores, the accuracy score of logistic regression is 67%. The accuracy score of the xg boost classification is 69%. The accuracy score of random forest classification is 67.63%. Based on the above results, the xg boost classification performs better as compared to other models.

Figure 17: Conclusion

IV. Future Scope

According to the above research, we can suggest that transfer learning models can be implemented in the future which performs better than the machine learning techniques. We can also think of implementing the above models on large datasets and changing the techniques such as generative adversarial networks. This research work was implemented on a smaller dataset which is one of the disadvantages so in the future, we can use a huge dataset.

V. References

[1] Bhosale, Y.H., Zanwar, S.R., Jadhav, A.T., Ahmed, Z., Gaikwad, V.S. and Gandle, K.S., 2022, October. Human Monkeypox 2022 virus: Machine learning prediction model, outbreak forecasting, visualization with time-series exploratory data analysis. In 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[2] Meena, G., Mohbey, K.K., Kumar, S. and Lokesh, K., 2023. A hybrid deep learning approach for detecting sentiment polarities and knowledge graph representation on monkeypox tweets. Decision Analytics Journal, 7, p.100243.

[3] Mohbey, K.K., Meena, G., Kumar, S. and Lokesh, K., 2023. A CNN-LSTM-based hybrid deep learning approach for sentiment analysis on Monkeypox tweets. New Generation Computing, pp.1-19.

[4] Nia, Z.M., Bragazzi, N.L., Wu, J. and Kong, J.D., 2023. A Twitter Dataset for Monkeypox, May 2022. Data in Brief, 48, p.109118.

[5] Jokar, M., Rahmanian, V. and Jahanbin, K., 2023. Monkeypox outbreak reflecting rising search trend and concern in nonendemic countries: a Google trend analysis. Disaster Medicine and Public Health Preparedness, 17, p.e286.

[6] Jokar, M. and Rahmanian, V., 2023. Potential use of Google Search Trend analysis for risk communication during the mpox (formerly monkeypox) outbreak in Iran. Health Science Reports, 6(1).

[7] Jokar, M., Rahmanian, V. and Jahanbin, K., 2023. Monkeypox outbreak reflecting rising search trend and concern in nonendemic countries: a Google trend analysis. Disaster Medicine and Public Health Preparedness, 17, p.e286.

[8] Thakur, N., Duggal, Y.N. and Liu, Z., 2023. Analyzing Public Reactions during the MPox Outbreak: Findings from Topic Modeling of Tweets.

[9] Sari, P.K. and Suryono, R.R., 2024. KOMPARASI ALGORITMA SUPPORT VECTOR MACHINE DAN RANDOM FOREST UNTUK ANALISIS SENTIMEN METAVERSE. Jurnal Mnemonic, 7(1), pp.31-39.

[10] Dada, E.G., Oyewola, D.O., Joseph, S.B., Emebo, O. and Oluwagbemi, O.O., 2022. Ensemble machine learning for monkeypox transmission time series forecasting. Applied Sciences, 12(23), p.12128.

[11] Wang, A., Li, D., Shen, W. and Zhang, X., 2023. Monkeypox Cases Prediction with Machine Learning. Highlights in Science, Engineering and Technology, 39, pp.246-257.