International Journal of Multidisciplinary Research and Growth Evaluation.

Disease identification

Umar Muzaffar [1*], Omar Momin [2], Shaik Neeha [3]
[1-2] BE CSE (AI& DS) MJCET OU Hyderabad, Telangana, India
[3] Assistant Professor CS & AI Department MJCET OU Hyderabad Telangana, India

* Corresponding Author: **Umar Muzaffar**

Article Info

ISSN (online): 2582-7138
Impact Factor: 5.307 (SJIF)
Volume: 05
Issue: 01
January-February 2024
Received: 21-10-2023;
Accepted: 23-11-2023
Page No: 71-78

Abstract
The purpose of the project '**Disease Identification Model'** is to develop an automated system for identifying diseases based on symptoms. The system will use machine learning algorithms to analyze and interpret the data provided by patients and make accurate predictions about the underlying condition. The system will be designed to provide an efficient and reliable method for disease **Prognosis**, reducing the time and cost associated with traditional methods. By incorporating medical expertise and utilizing the latest technology, this project aims to improve the accuracy and speed of disease identification, and ultimately improve patient outcomes.
The project is deployed as a web app for identifying diseases and recommending quick relief medicine by taking input list of symptoms from the user via a questionnaire.
The identifications are specific to a small group of diseases where the **common symptom** is **cold and fever**, diseases like Typhoid, Jaundice and Cholera. We offer some other disease identifications like identifying if the person is Cardiovascular Disease symptomatic.

DOI: https://doi.org/10.54660/.IJMRGE.2024.5.1.71-78

Keywords: Disease identification, Automated Prognosis

1. Introduction

Disease Identification is the process of determining the cause and nature of a person's illness based on symptoms, medical history, and diagnostic tests. It involves a systematic evaluation of symptoms, physical examination, and laboratory tests to identify the underlying disease or condition and make an accurate diagnosis. The ultimate goal of disease identification is to provide effective treatment and improve the health outcomes for the patient.
The project is a web app for identifying diseases based on user input list of symptoms and recommending quick relief medication. It is capable of making predictions with high accuracy. The user input is taken through a questionnaire under tabs for each disease.
We use multiple datasets for each disease and train a model. All disease prediction models are then combined to make the final application. This was done instead of combining into one questionnaire since it was found that the prediction models did not give high enough accuracy when combined into one.

2. Existing System

The existing system of disease identification primarily relies on manual diagnosis by healthcare providers, who use a combination of patient interviews, physical examinations, laboratory tests, and medical imaging to identify the underlying condition. In some cases, specialist consultations may also be sought to reach a definitive diagnosis. Electronic records and computerized systems are used to store and manage patient data, but these systems often rely on manual input and do not provide automatic disease identification.
There are also some disease diagnosis systems that use decision tree algorithms or other machine learning techniques, but these

are not widely used in clinical settings and have limited accuracy and precision.
Overall, the current system of disease identification is time-consuming and often lacks the necessary accuracy and efficiency to meet the demands of modern healthcare.

3. Proposed System

Problem Statement: Disease identification models based on symptom inputs used to automate the preliminary questioning phase between a doctor and patient.
It is similar in the way where some hospitals have a questionnaire document a patient has to fill out while waiting for an appointment with the doctor which asks basic questions about their problems. The web app helps automate the same process and speed it up for the doctor with an online record of the patient's report. The model recommends quick relief medication but it cannot be considered as treatment to the ailment. The model is not an Expert System and it still needs a look by an actual expert in this case the doctor for proper treatment. The medication recommendation is for the case where the user is in immediate need for some relief.
Most of the diseases in this project namely Typhoid, Jaundice and Cholera are diseases with common symptoms of cold and fever. It can help users and doctors to get clarity on their ailment and direct them to more concrete tests to finalize their diagnosis. **The model is akin to a Prognosis** (i.e. an opinion, based on medical experience, of the likely course of a medical condition.).

4. Methodology

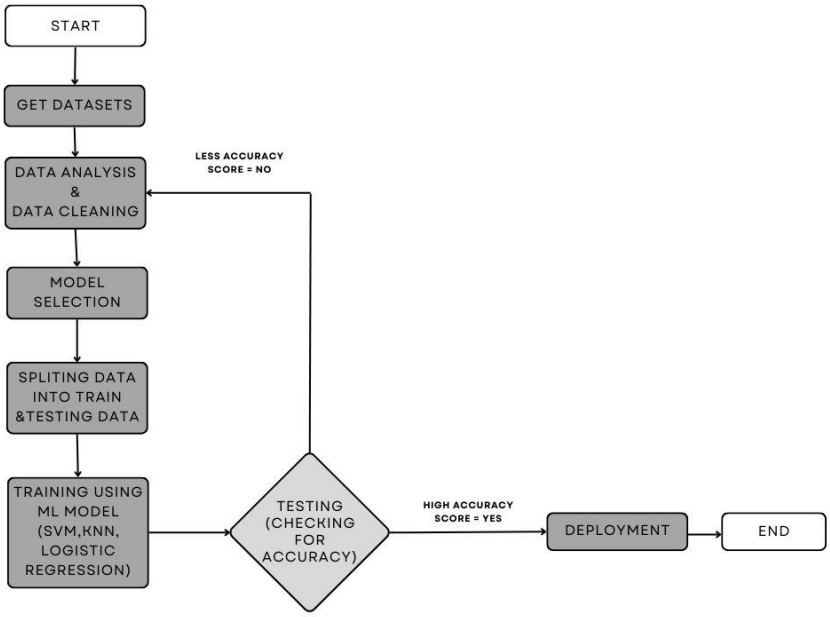


Fig 1: Model Pipeline

The above figure shows the project pipeline. This is the standard process of a data science project.

4.1. Datasets

There are a total of 5 datasets used

1. Cholera
2. Jaundice
3. Typhoid
4. Diabetes
5. Cardiovascular Diseases

id	name	gender	fever	headaches_and_nausea	vomiting	fatigue	severe_dehydration	diarrhea	pain_in_abdomen	target
1	Ulick	Male	1	1	1	0	0	0	1	0
2	Brynna	Female	1	1	1	0	1	1	1	1
3	Ahmad	Male	1	0	1	1	0	1	0	0
4	Randell	Male	1	0	0	1	0	1	0	0
5	Charlton	Male	1	1	0	0	1	0	0	0
6	Augustin	Male	1	1	1	0	1	0	0	1
7	Obadias	Male	1	1	0	0	0	0	1	0
8	Arvie	Male	1	1	0	0	1	0	0	0
9	Lauralee	Female	1	1	1	1	1	0	0	1
10	Jaine	Female	1	1	1	1	0	1	0	1
11	Cart	Male	1	0	1	1	0	1	1	1

Fig 2: Cholera Dataset

id	name	gender	age	fever	chills	abdominal_pain	yellowing_of_skin	yellowing_of_eyes	dark_colored_urine	target
1	Vincent Tamblyn	Male	68	1	1	0	0	0	0	0
2	Mariquilla O'Duilleain	Female	3	1	1	0	0	1	1	1
3	Lia Stollwerck	Female	12	1	0	0	1	1	0	1
4	Larisa Kovnot	Female	6	1	1	1	1	1	0	1
5	Brett Livsey	Male	20	1	1	1	0	1	0	0
6	Raffarty Ashman	Male	9	1	1	0	0	1	1	1
7	Daffy Bebb	Female	36	1	1	0	0	1	0	0
8	Aldon Barrie	Male	4	1	1	0	1	1	1	1
9	Adamo Seabert	Male	6	1	0	1	0	1	1	1
10	Diena Crankhorn	Female	57	1	1	0	0	1	0	0
11	Nevin Roberto	Male	5	1	1	1	0	1	1	1

Fig 3: Jaundice Dataset

id	name	gender	age	fever	chills	bloating	red_dots_on_skin	loss_of_appetite	target
1	Bernardina Ferriere	Female	61	1	0	0	0	1	0
2	Deloris Rubinivitz	Female	49	1	0	0	0	0	0
3	Maxine Tittershill	Female	27	1	0	1	0	0	0
4	Jodie Pead	Female	34	1	1	1	0	0	0
5	Hughie McDougall	Male	32	1	0	0	1	1	1
6	Marten Bainbridge	Male	26	1	1	1	0	0	0
7	Even Round	Male	57	1	0	1	1	0	1
8	Roland Durden	Male	48	1	0	0	1	1	1
9	Benedikta Hackly	Female	50	1	0	0	1	0	0
10	Darlene Lindblom	Female	58	1	0	0	0	1	0
11	Giulietta Fremantle	Female	65	1	0	1	0	0	0

Fig 4: Typhoid Dataset

The datasets were separated into training and testing datasets using the 80-20 split for training and testing respectively.

- **Training data:** Most of the data (about 80%) should be training / validating data since that is best used for gaining accuracy. Testing process needs only little data to verify its model accuracy and would be under-trained if spent on testing only.
- **Testing data:** Testing data is used to verify working and accuracy of the model and is best split as 20% of all data.

The dataset splitting library used was: **sklearn model_ selection. train_ test_ split**

```
from sklearn.model_selection import train_test_split
```

Fig 5: Data splitting library import

4.2. Data Pre-Processing

Data Pre - Processing is when you clean the data and prepare it for training. We cleaned the data by

- Filling null values or dropping null values entirely,
- removing duplicates,
- standardization,
- removing outliers

The preprocessing library used was: **sklearn.preprocessing.StandardScalar**

```
from sklearn.preprocessing import StandardScaler
```

Fig 6: Data Pre-Processing library import

4.3. Feature Selection

Necessary features are selected as needed for the model.
E.g. in the jaundice dataset there were attributes for age, gender, etc. These features are not needed to train our model as we do not need all of those attributes to get a working model. They aren't unnecessary however but they require higher processing power and ultimately do not serve a vital purpose in the results. They can be used for more sophisticated models but that comes under Expert Systems.

```
X=cholera_data.drop(['id','name','gender','target'],axis=1)
Y=cholera_data['target']
```

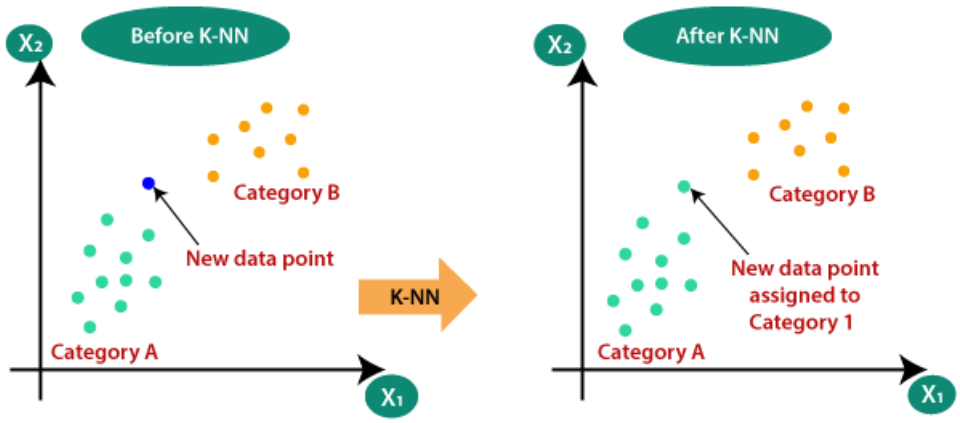
Fig 7: Feature selection example

4.4. Machine Learning Models

The machine learning models used in this project are

- **KNN (K-Nearest Neighbors) Algorithm**

It is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

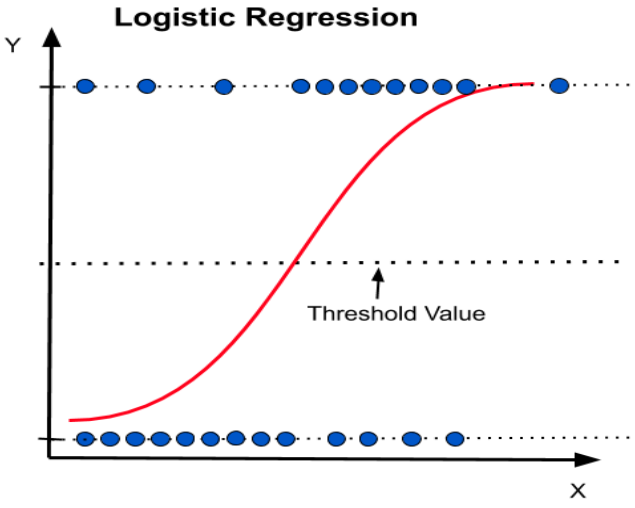



```
from sklearn.neighbors import KNeighborsClassifier
```

Figure 8: KNN Model

- **Logistic Regression**

It is a technique that uses mathematics to find the relationships between two data factors and uses this relationship to predict the value of one of those factors based on the other. The prediction usually has binary outcomes.

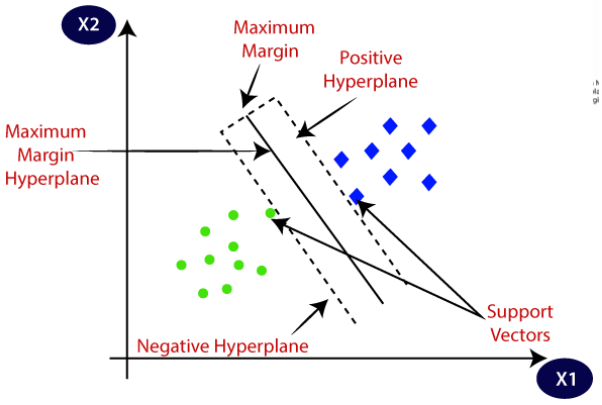



```
from sklearn.linear_model import LogisticRegression
```

Fig 9: Logistic Regression

- **SVM (Supported Vector Machine)**

It is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.




```
from sklearn import svm
```

Fig 10: SVM Model

5. Implementation

5.1. Software and hardware requirements

Project was done on **Google Colab** hence, no software or hardware is required.
It works on any Operating System and Browser.

5.2. Languages

- **Python:** It offers readable and concise codes. Since machine learning and artificial intelligence involve complex algorithms, the simplicity of Python adds value and enables the creation of reliable systems. It is done through its extensive list of libraries and tools available for machine learning and artificial intelligence.

5.3. Applications

- **Anaconda**
- **Spyder IDE**
- **Excel:-** For storing data

5.4. Libraries

- **Pandas:** It is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.
- **Numpy:** It is a library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.
- **Sklearn:** It is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.
- **Streamlit:** It is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time.
- **Pickle:** It is primarily used in serializing and deserializing a Python object structure.

5.5 Cholera Model (USING SVM)

```
import numpy as np
import pandas as pd
from sklearn import svm
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

```
cholera_data=pd.read_csv('/content/CholeraSymptoms.csv')
```

Fig 11: Cholera libraries and dataset import

```
cholera_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    150 non-null    int64
 1   name                  150 non-null    object
 2   gender                150 non-null    object
 3   fever                 150 non-null    int64
 4   headaches_and_nausea  150 non-null    int64
 5   vomiting              150 non-null    int64
 6   fatigue               150 non-null    int64
 7   severe_dehydration    150 non-null    int64
 8   diarrhea              150 non-null    int64
 9   pain_in_abdomen       150 non-null    int64
 10  target                150 non-null    int64
dtypes: int64(9), object(2)
memory usage: 13.0+ KB
```

Fig 12: Cholera dataset information

```
X_train_prediction=classifier.predict(X_train)
training_data_accuracy=accuracy_score(X_train_prediction,Y_train)
```

Fig 13: Cholera model training and its accuracy

```
X_test_prediction=classifier.predict(X_test)
X_test_accuracy=accuracy_score(X_test_prediction,Y_test)
```

Fig 14: Cholera model testing and its accuracy

5.6 Jaundice Model (Using Logistic Regression)

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
jaundice_data=pd.read_csv("/content/JaundiceDataSet.csv")
```

Figure 15: Jaundice libraries and dataset import

```
jaundice_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   id                  250 non-null    int64
 1   name                250 non-null    object
 2   gender              250 non-null    object
 3   age                 250 non-null    int64
 4   fever               250 non-null    int64
 5   chills              250 non-null    int64
 6   abdominal_pain      250 non-null    int64
 7   yellowing_of_skin   250 non-null    int64
 8   yellowing_of_eyes   250 non-null    int64
 9   dark_colored_urine  250 non-null    int64
 10  target              250 non-null    int64
dtypes: int64(9), object(2)
memory usage: 21.6+ KB
```

Fig 16: Jaundice dataset information

```
X_train_prediction=model.predict(X_train)
X_train_accuracy=accuracy_score(X_train_prediction,Y_train)
```

Fig 17: Jaundice model training and its accuracy

```
X_test_prediction=model.predict(X_test)
X_test_accuracy=accuracy_score(X_test_prediction,Y_test)
```

Fig 18: Jaundice model testing and its accuracy

5.7 Typhoid Model (Using KNN)

```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
```

```
thypoid_data=pd.read_csv("/content/ThypoidDataSet.csv")
```

Fig 19: Typhoid libraries and dataset import

```
X_train_prediction=classifier.predict(X_train)
training_data_accuracy=accuracy_score(X_train_prediction,Y_train)
print(training_data_accuracy)
```

Fig 20: Typhoid model training and its accuracy

```
X_test_prediction=classifier.predict(X_test)
X_test_accuracy=accuracy_score(X_test_prediction,Y_test)
print(X_test_accuracy)
```

Fig 21: Typhoid model testing and its accuracy

6. Result Analysis

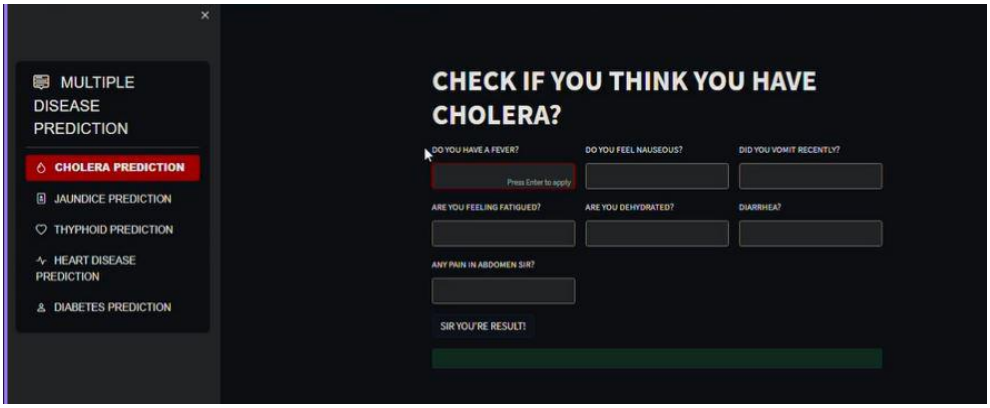


Fig 22: Final Streamlit web app

6.1. Accuracy Scores

The project was forced to be separated into different modules since a combined model did not provide satisfactory accuracy. These accuracy scores were measured by following scores

- **Accuracy score:** Accuracy score is used to measure the model performance in terms of measuring the ratio of sum of true positives and true negatives out of all the predictions made.

```
from sklearn.metrics import accuracy_score
```

Fig 23: Accuracy score import used in all models

- **F1 score:** F1 score is commonly used to measure performance of binary classification, but extensions to multi-class classifications exist.
- **Confusion Matrix:** Confusion matrix is a table that is used to define the performance of a classification algorithm.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
```

Figure 24: Accuracy scores import used in KNN model

7. Conclusion and future work

To conclude the model works satisfactorily and predicts diseases with needed accuracy. To reiterate this is not a professional diagnosis despite consulting a medical student for our project. It is simply a Prognosis and a quick tool for the user to have a good guess on what they are suffering from and buy relief medication before getting fully checked since it might take time before they get a doctor's opinion.

Future enhancements include

Merging the different models into one where it can directly chat with patients and automatically give a prognosis or if it's advanced enough, even give a diagnosis. We would not have to check for every disease individually instead train it so that it can identify any disease using one model.

Making a doctor - patient interface where it automatically records patient data and identifies diseases eliminating any need for manual entries. A **doctor recommendation system** would be the next improvement where it can give the patient a list of nearest available doctors and a rating system for the patient to pick a doctor of their choosing.

References

1. https://www.kaggle.com/datasets/yasserh/heart-disease-dataset
2. https://www.kaggle.com/datasets/mathchi/diabetes-data-set
3. https://github.com/UmarMuzAhmed/MultipleDiseasePrediction/tree/main/MLprojectsFolder