Disease Prediction System

Prepared by: Shreya Paul(Team Leader)
Sneha Jitendra Gaupale
Sayan Banerjee
Sidarathul Munthaha PV

Abstract:

Machine learning and artificial intelligence are revolutionizing the healthcare domain, addressing crucial needs and ushering in a new era of medical innovation. One of the primary reasons for their necessity is the vast amount of data generated within healthcare. With electronic health records, medical images, genomic information, and patient data expanding exponentially, machine learning and AI algorithms are essential for processing and extracting meaningful insights from this wealth of information. These technologies enable healthcare professionals to make more accurate diagnoses and treatment decisions. They can analyze patient data to identify patterns and trends that may be imperceptible to the human eye, leading to early detection of diseases and tailored treatment plans. Additionally, AI-driven predictive analytics can help in forecasting disease outbreaks and optimizing resource allocation, making healthcare systems more efficient and responsive.

Machine learning and AI also play a pivotal role in drug discovery and development. They can sift through vast datasets to identify potential drug candidates, significantly accelerating the research process and reducing costs. In personalized medicine, AI can match patients with the most suitable treatments based on their genetic profiles, improving the efficacy and safety of medical interventions.

Furthermore, AI-driven tools are enhancing patient care by offering remote monitoring, telemedicine, and virtual health assistants, making healthcare more accessible and convenient. These technologies can also aid in managing and optimizing healthcare resources, such as hospital operations and supply chain management, contributing to cost savings and improved patient outcomes.

Problem Statement

The COVID-19 pandemic has underscored the deficiencies within our healthcare system and underscored the imperative of technology integration in the healthcare sector. The scarcity of accessible information resulted in susceptibility to fraudulent activities and facilitated the rapid proliferation of the disease. However in the post-COVID era, there has been a notable elevation in public health consciousness. Individuals have exhibited a heightened propensity to seek health-related information on online platforms. Nevertheless, the online resources frequently offer limited and technically intricate content, thereby posing challenges for comprehension among consumers. As a part of our project, we have decided to build a ML based recommendation system platform for the doctors as well as customers which will fulfil their information needs.

Market/Business Need Assessment:

In recent years, there has been a substantial surge in the demand for healthcare services. The COVID-19 pandemic underscored the imperative for the integration of telehealth solutions, emphasizing the need for scalable healthcare delivery. Addressing this increased demand necessitates a comprehensive solution involving large-scale treatment system implementation. One viable approach to address this challenge is the automation of disease diagnosis to a significant extent. Leveraging the advancements in machine learning and artificial intelligence, the development of an automated diagnostic model represents a groundbreaking innovation within the healthcare sector. This not only holds the potential to benefit healthcare providers but also offers a user-friendly interface for consumers to conveniently diagnose their ailments by inputting their symptoms. This system is designed to recognize both common diseases and chronic conditions, thanks to the training of its sophisticated model. User can input their symptoms and upload their reports(if any) to the user interphase. ML model will predict the possible disease based on the symptoms. The process will faster the pace of healthcare delivery.

Machine learning models can also be integrated into clinical workflows to assist healthcare professionals in diagnosing diseases, recommending treatment options, and predicting patient outcomes. The data driven recommendations of the ML model will be much more accurate and freer from human errors. Model will also help in predicting disease outbreaks, identifying high-risk patients, and managing population health more effectively. With the increasing use of machine learning, there is a growing emphasis on data interoperability and security to ensure patient privacy and compliance with regulations such as HIPAA. Machine learning is driving the development of personalized treatment plans based on a patient's unique genetic and clinical profile. So far, the healthcare sector in the country is informal in nature, hence this application aims to give doctors as well as customers an integrated holistic platform.

Target Specification and characterization :

This comprehensive target specification and characterization provide a roadmap for the development, deployment, and ongoing refinement of machine learning algorithms for disease determination in the healthcare domain, ensuring that they meet the highest standards of accuracy, reliability, and ethical considerations. In this case the targeted customers are common people who are in need of diagnosing a disease. The system will take the input of the user, the patients will also have the option to upload their reports. The machine learning model will use these inputs and based on the training of the model, it will predict the disease for the patient. The platform will also protect the sensitive information of the patient as per the government regulations. Highly accurate data driven recommendation will also boost up the confidence of the customers and helps in providing personalized treatments.

External Search (Information and Data Analysis):

I have gathered additional information from various sources to analyze the data in this content.

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8896926/

2. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3661426

So, for this project I am going to use this Dataset:

1. Dataset1

Dataset Description:

The first dataset provides information about diseases and their associated symptoms, which is a crucial component of this system. It helps us identify solutions and enables us to inform users about potential problems and their corresponding solutions based on their symptoms.

Now, let's start with the process of this system:

1. Import the libraries for Data Preprocessing:

Data Preprocessing

Import Libraries

```
import numpy as np
import pandas as pd
```

Now let's see the Dataset:

```
df = pd.read_csv("Dataset.csv")
```

```
df.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scurrin
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	1	1	0	0	0	0	0	0	0	...	0	
2	1	0	1	0	0	0	0	0	0	0	...	0	
3	1	1	0	0	0	0	0	0	0	0	...	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	

5 rows × 133 columns

Now, it's time to see the information about dataset:

```
1 df.shape
```

```
(4920, 133)
```

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 5.0+ MB
```

Description of Dataset:

```
1 df.describe()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_ton
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000
mean	0.137805	0.159756	0.021951	0.045122	0.021951	0.162195	0.139024	0.045122	0.045122	0.021
std	0.344730	0.366417	0.146539	0.207593	0.146539	0.368667	0.346007	0.207593	0.207593	0.146
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000

8 rows × 132 columns

As we can see from the cells above, we have a total of 132 features and one label/class. Depending on the different features, you can determine which disease you are suffering from.

2. It's time to clean the dataset:

Data Cleaning

Checking Missing Values of Vehicle data.

```
1 df.isnull().sum()
```

```
itching               0
skin_rash             0
nodal_skin_eruptions  0
continuous_sneezing   0
shivering             0
                     ..
inflammatory_nails    0
blister               0
red_sore_around_nose  0
yellow_crust_ooze     0
prognosis             0
Length: 133, dtype: int64
```

Handling Categorical Data:

```
for label, content in df.items():
    if not pd.api.types.is_numeric_dtype(df[label]):
        lst = df[label].unique()
        label_map = {}
        for i in range(len(lst)):
            label_map[lst[i]] = i
        df[label] = df[label].map(label_map)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 173 entries, itching to prognosis_hepatitis A
dtypes: int64(132), uint8(41)
memory usage: 5.1 MB
```

```
df
```

prognosis_Osteoarthristis	prognosis_Paralysis (brain hemorrhage)	prognosis_Peptic ulcer diseae	prognosis_Pneumonia	prognosis_Psoriasis	prognosis_Tuberculosis	prognosis_Typhoid	prognos tra
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
...	...	...	...	...	...	...	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	1	0	0	
0	0	0	0	0	0	0	

To check whether the dataset is balanced or not by using Bar Plot:

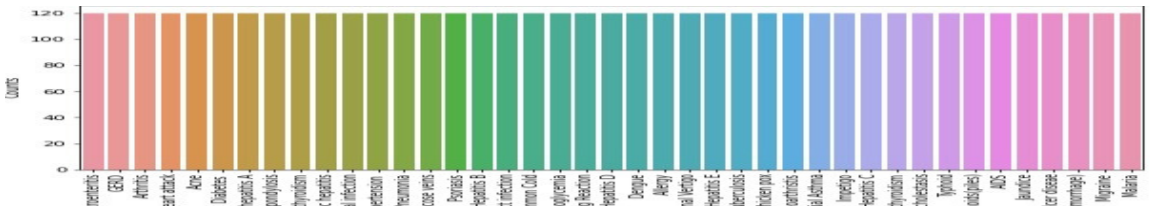
```

data = "Dataset.csv"
df = pd.read_csv(data).dropna(axis = 1)

dise_count = df["prognosis"].value_counts()
temp_df = pd.DataFrame({
"Disease": dise_count.index,
"Counts": dise_count.values})

plt.figure(figsize = (10,5))
sns.barplot(x = "Disease", y = "Counts", data = temp_df)
plt.xticks(rotation=90)
plt.show()

```



As we can see, the dataset is completely balanced, and we have already converted our categorical class into numerical ones for a better view.

Benchmarking for Disease Prediction System:

Introduction:

Working on the Performance and Capabilities of this system, benchmarking played a very crucial role. An overview of the benchmarking process, goals, methodology, and its effects on our project are given in this section.

Objectives of Benchmarking:

Our benchmarking guided by the following objectives:

- To evaluate the “Disease prediction system” based on the speed, accuracy and overall performance of this system.
- Compare our system with industrial standards and their solutions.
- Identify the area for improvement of our system’s algorithm and user interface.

Methodology:

The benchmarking process involved the following key steps:

Data Collection:

We analyze various datasets, data sources to understand the specific requirements of our system, ensuring it covered various disease and symptoms.

Selection of Benchmarks:

We found that, industry standards and benchmark systems that are well-known for their superior disease prediction capabilities.

Analysis and Comparison:

Based on the benchmarking data, we observed the following:

We recognize the need of continuous improvement in our search of excellence. We aim to generate better results and outputs because we are always looking for better ways to do things. This is a commitment to the health and welfare of our users as well as to the success of our project, as they depend on our system for accurate illness forecasts, medical advice, and easy access to a large database of medical data.

Conclusion of Benchmarking:

The practice of benchmarking played a crucial role in evaluating and improving this system .We were able to improve system performance and bring it into line with the standards of the industry as a result.

Applicable Patents:

- https://patents.google.com/patent/CN109036553B/en?q=(Disease+Prediction+system)&oq=Dis ease+Prediction+system
- https://patents.google.com/patent/KR101884609B1/en?q=(Disease+Prediction+system)&oq=Disease+Prediction+system
- https://patents.google.com/patent/EP2652684B1/en?q=(pateint+Diagnosis+Prediction+system)&oq=pateint+Diagnosis+Prediction+sytem

Applicable Regulations (Government and Environmental):

- Healthcare Data Regulations.
- Data Privacy and Protection.
- Ethical Considerations.

Applicable Constraints :

- The use of GUI or web app for the user interface.
- Building database of diseases and symptoms.
- For Evaluation of the model is done using data visualisation. For modelling, Logistic Regression is applied.

Business Opportunity :

Health care system can utilise the system and make huge profit by automating the tasks which allows the services to implement on a large scale basis in a reduced cost margin. This systematic review aims to determine the performance, limitations, and future use of Software in health care. Findings may help inform future developers of Disease Predictability Software and promote personalized patient care.

Concept Generation:

This product needs EDA and machine learning modelling to build a predictive system. For this purpose it uses logistic regression algorithm. It also needs front end development besides machine learning part. Here we have to decide which disease a patient will have according to the symptom they poses. So here we are dividing the patients into different classes of diseases according to their certain symptoms. For that purpose we can use logistic regression since it is a classification algorithm. Logistic regression uses the input data to decide the output class.

Logistic regression uses probability to decide the classes. The formula for the logistic regression is as follows:

$$h_{\Theta}(X) = \frac{1}{1 + e^{-\Theta X}}$$

As the formula above Θ is the parameter we want to learn or train or optimize and $\mathbf{X}$ is the input data.

Training the model :

Splitting the trainig and testing data set ¶

```
In [28]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,stratify=y, random_state=2)

In [29]: print(X.shape, X_train.shape, X_test.shape)

(4920, 132) (3936, 132) (984, 132)

In [30]: print(y.shape, y_train.shape, y_test.shape)

(4920,) (3936,) (984,)
```

Training the model

```
In [31]: model=LogisticRegression()

In [32]: model.fit(X_train,y_train)

Out[32]: LogisticRegression()
```

Obtaining the result:

checking which disease will the patient have according to the given symptoms

```
In [34]: #giving 1 if the patient have that particular symptom else the value for the column is 0
input_data=(0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

input_data_np=np.asarray(input_data)

input_data_reshaped= input_data_np.reshape(1,-1)

prediction=model.predict(input_data_reshaped)
print(prediction)

['Fungal infection']
```

Concept Development :

Our system of disease prediction can be launched by using the appropriate API (like Flask and Django). Deployment can be done using the cloud services according to our requirement.

Final Result :

```
In [42]: test_data=test_data.drop(columns='prognosis',axis=1)

In [43]: input_data=test_data

         # input_data_np=np.asarray(input_data)

         # input_data_reshaped= input_data_np.reshape(1,-1)

         prediction=model.predict(input_data)
         print(prediction)

         ['Fungal infection' 'Allergy' 'GERD' 'Chronic cholestasis' 'Drug Reaction'
          'Peptic ulcer diseae' 'AIDS' 'Diabetes ' 'Gastroenteritis'
          'Bronchial Asthma' 'Hypertension ' 'Migraine' 'Cervical spondylosis'
          'Paralysis (brain hemorrhage)' 'Jaundice' 'Malaria' 'Chicken pox'
          'Dengue' 'Typhoid' 'hepatitis A' 'Hepatitis B' 'Hepatitis C'
          'Hepatitis D' 'Hepatitis E' 'Alcoholic hepatitis' 'Tuberculosis'
          'Common Cold' 'Pneumonia' 'Dimorphic hemmorhoids(piles)' 'Heart attack'
          'Varicose veins' 'Hypothyroidism' 'Hyperthyroidism' 'Hypoglycemia'
          'Osteoarthristis' 'Arthritis' '(vertigo) Paroymsal  Positional Vertigo'
          'Acne' 'Urinary tract infection' 'Psoriasis' 'Impetigo'
          'Fungal infection']
```

Final Product Prototype/ Product Details :

Final product helps to determine in diagnosing diseases without the necessity of consulting a doctor. The system takes the symptoms as the input and give the result as a disease.
EDA or exploratory data analysis is done before using the data set to analyse and prepare the data set for the predictive purpose.
It simplifies the process of diagnosis and reduce the cost of consulting.it also brings automation in the field of health care which can bring a revolution in the health care business.
The system can predict different diseases according to the symptoms. And it uses the probability of having that disease according to the symptoms. Logistic regression used for classification helps in building a predictive model for the same.

Feasibility

The project has its own place in the market since the automation of health care system is in high demand anywhere in the world anytime for the betterment of the society.It provides a simple yet effective approach for predicting the disease, if the provided values of vitals are accurate.

Viability

It has the potential to improve healthcare outcomes and reduce healthcare costs by predicting and preventing disease early.

Monetization

The main challenge is to get the data set and the project is monetizable for any scale according to its specified requirements.

Final Product Prototype :

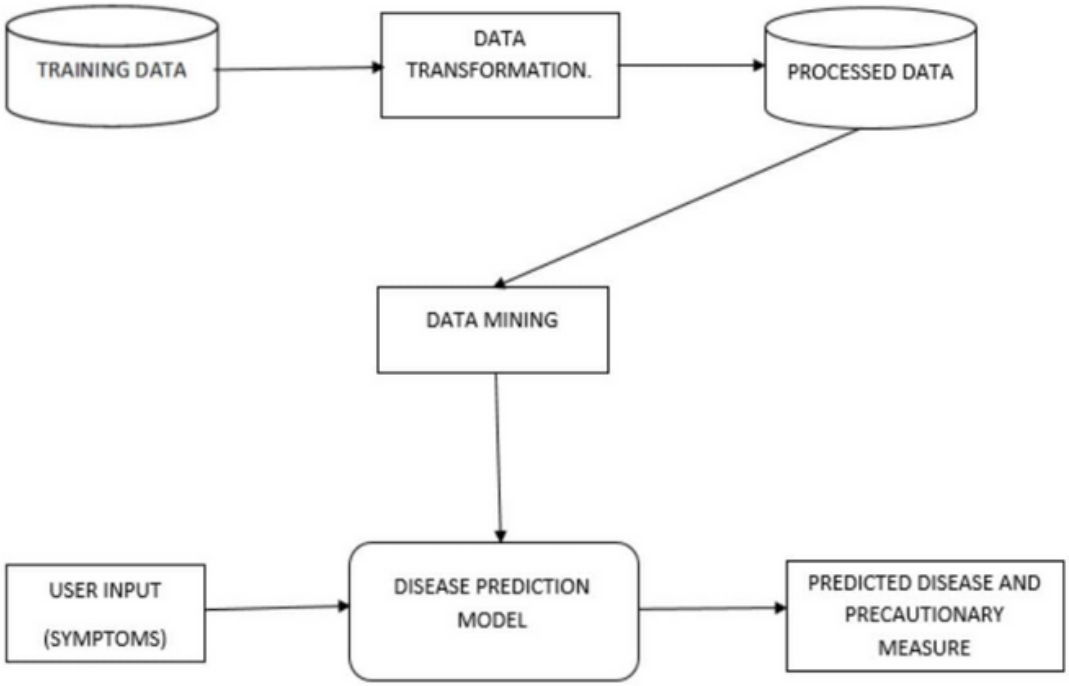


Fig: System architecture of disease prediction system

Code Implementation/Validation :

Algorithm:

- Decision Tree
- Random Forest
- KNearestNeighbour
- Naive Bayes

Python Libraries used:

- **Pandas:** Pandas is defined as an open-source library that provides high-performance data manipulation in Python. Data analysis becomes easier with this Python library.
- **NumPy:** NumPy is used for mathematical function purpose. NumPy arrays are more memory efficient and compact than Python lists.
- **Matplotlib:** Matplotlib is a visualization library of Python. We can graphically represent data and their relations through Matplotlib.
- **Sklearn:** Sklearn is predictive analysis based Python library. Data splitting, algorithm implementation etc. are done through Sklearn.

- **Decision Tree:** A decision tree is a structure that can be used to divide up a large collection of records into successfully smaller sets of records by applying a sequence of simple decision tree. With each successive division, the members of the resulting sets become more and more similar to each other.
- **K Nearest Neighbour (KNN):** KNN could be terribly easy, simple to grasp, versatile and one amongst the uppermost machine learning algorithms. In the Healthcare System, the user will predict the disease. In this system, the user can predict whether the disease will detect or not. In the proposed system, classifying disease in various classes that shows which disease will happen on the basis of symptoms. KNN rule used for each classification and regression issue.
- **Naive Bayes :**Naive Bayes is an easy however amazingly powerful rule for prognosticative modelling. This simplified Bayesian classifier is called as naive Bayes. The Naive Bayes classifier assumes the presence of a particular feature in a class is unrelated to the presence of any other feature. It is very easy to build and useful for large datasets. Naive Bayes is a supervised learning model. Bayes theorem provides some way of calculative posterior chance P(b|a) from P(b), P(a) and P(a|b). Look at the equation below:

  P(b v a)= P(a v b)P(b)/P(a)

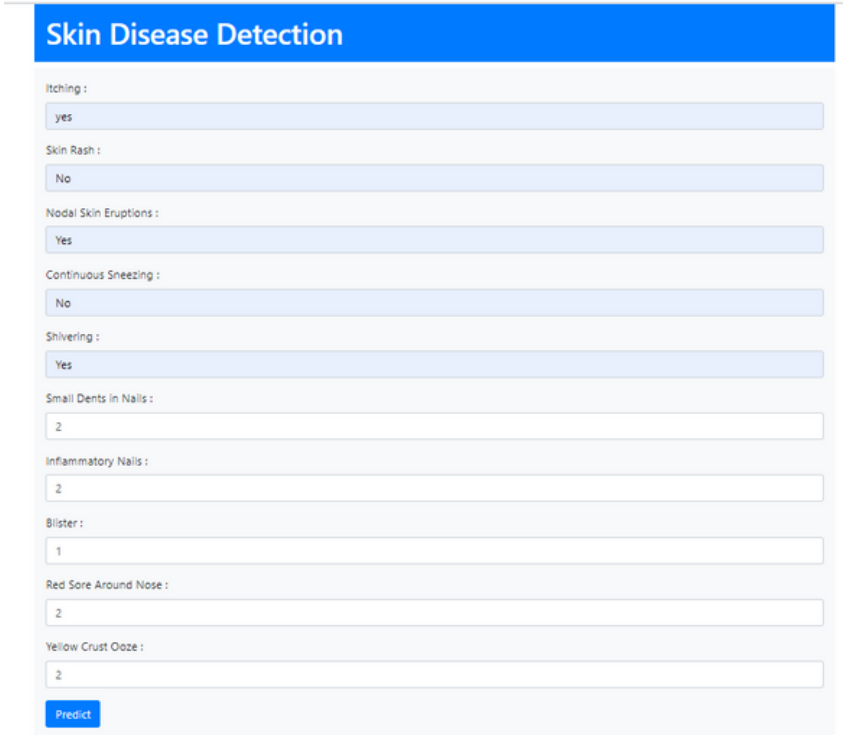
Back-end:

- This calls for the gathering of data, pre-processing, and integration of the model with the web application.
- The customer's consent should also be obtained before collecting and storing any of their entered data.

Front-end:

- The front end is important since it is the user interface that the customer will interact with.
- The web application could have mainly two pages. Input taking on the first page, then result comes on the second page.
- It needs to be very user-friendly; otherwise, users can enter incorrect information and the prediction will be completely wrong.

Web implementatio




```python
import pickle
import pandas as pd
import numpy as np

app = Flask(__name__, static_url_path='/static')

# Load the model from the pickle file
with open('model.pkl', 'rb') as f:
    model = pickle.load(f)

@app.route('/')
def index():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
    try:
        # Get the form data from the request
        itching = int(request.form['itching'])
        skin_rash = int(request.form['skin_rash'])
        nodal_skin_eruptions = int(request.form['nodal_skin_eruptions'])
        continuous_sneezing = int(request.form['continuous_sneezing'])
        shivering = int(request.form['shivering'])
        small_dents_in_nails = int(request.form['small_dents_in_nails'])
        inflammatory_nails = int(request.form['inflammatory_nails'])
        blister = int(request.form['blister'])
        red_sore_around_nose = int(request.form['red_sore_around_nose'])
        yellow_crust_ooze = int(request.form['yellow_crust_ooze'])

        # Process the data and make the prediction using the loaded model
        data = {
            "itching": itching,
            "skin_rash": skin_rash,
            "nodal_skin_eruptions": nodal_skin_eruptions,
            "continuous_sneezing": continuous_sneezing,
            "shivering": shivering,
            "small_dents_in_nails": small_dents_in_nails,
            "inflammatory_nails": inflammatory_nails,
            "blister": blister,
            "red_sore_around_nose": red_sore_around_nose,
            "yellow_crust_ooze": yellow_crust_ooze
        }

        # Convert the dictionary to a DataFrame and reshape it for prediction
        df = pd.DataFrame([data])
        first_element = df.iloc[0]
        first_element_array = np.array(first_element)
        first_element_reshaped = first_element_array.reshape(1, -1)

        # Make the prediction
        prediction = model.predict(first_element_reshaped)
        prediction_value = "Positive" if prediction == 1 else "Negative"  # Change labels based on model predictions

        # Return the prediction as a JSON response
        return jsonify({'prediction': prediction_value})
    except Exception as e:
        return jsonify({'error': str(e)})

if __name__ == '__main__':
    app.run(debug=True)
```

Business Model:

In this part of the report, we will look at the business model suggested for the idea presented earlier. There are many business models available but we have chosen the best suitable model for our dataset.

Step1: We meet the user through online interface. As our model will be embedded in the website, so its benefit will gain maximum reach. The hassle of physical presence for diabetics check will be nullified.

Step2: To give the personalized experience to all the user we have arranged the login system. This will also help the users to keep their trail of their diabetic checks.

Step3: For Credit Card Fraud Detection check we need certain required details from the user. We will take all these details using our interface.

Step4: From the user interface these attributes will be passed to the backend through API. And in the backend, we have kept our trained model which will show predictions based on those collected information from the respective user.

Step 5: The prediction will be reflected in the webpage. If the prediction says that the user in not diabetic then he will be congratulated.

Step 6: If prediction says that he/she is diabetic the proper recommendation of various hospitals and doctors will be shown, so that user can take proper steps to improve his/her health.

User will also be guided for their medication which are highly recommended by the prestigious doctors.

Market Analysis:

The medical industry is in urgent need of such kind of automation. As we all know for any kind of surgery or serious operations the level of sugar should be at desired level. It becomes very difficult for any patient to keep a regular check of their diabetic level. As level of diabetics is unpredictable so it become crucial to keep it in regular check. But for regular check the physical presence at various clinics and leads to lots of chaos. To make this hassle free we came up this idea of developing diabetes check. This will reduce the hassle of regular check-up. Model will be train over huge number of records, so the accuracy of the model will be high and precise.

Since this is an exquisite combination of ML with medical problem so this implantation is new and attractive. Therefore, there is a huge opportunity for those who enter the market first.

Operating Plan:

The important part of our operation is to have ML/DS engineers with a good amount of knowledge about the industry. The product developing team's size should be 3 to 4 where one of the members must be a full stack web developer and the remaining members must be ML engineers. It would be beneficial if almost all the ML engineers had knowledge about the industry. The time for developing the product must be decided after a meeting with the client and the team developing the product. Having a clear idea about the deadline is a must and based on that the team can accelerate certain parts of the developing process. When we start providing this service, we must set a low price because initially we need the maximum reach. Once we have successfully implemented our model for the first client and depending on our model's performance, we can take stock about our position and decide on the pricing. Pricing should also be based on the type of hotel we offer our service to.

Financial Equation:

As we all know the demand for medical services will always remain high and it will increase with time. Therefore, our product will have higher chance to boom. To decide the approx. salary of our team, let's assume that to recommend the bigger hospitals we can charge around Rs. 15000 for three months and for pharmaceutical shops around Rs 10000 for 3 months. Once the customer base increases, we can either increase the price or reduce the duration for which our product will be available.

Let's assume that the duration of developing the ML model takes about 1 to 3 weeks and the cost for producing the model is the salary of the members the team.

Let there be two ML engineers and one full stack web developer. Let the salary of the ML engineers be 'ml' 18 and the full stack web developer be 'fs'.

So, the total
cost c = 2*ml + fs. So, the profit or financial equation will look like this
y = 25000*x(t) – (2*ml + fs)
Here x(t) is a function that represents the growth of the customer base and y is the profit.

GitHub repository link:

https://github.com/Shreyaa5/Disease_Prediction_System

Conclusion :

The Disease Prediction System was developed to predict the likelihood of individuals developing specific diseases using medical data. The challenge of dealing with imbalanced data.
We utilized three classification models: Logistic Regression, Random Forest Classifier, and possibly another model, evaluating them with metrics like accuracy, confusion matrices, and classification reports.
Key findings and recommendations may include which model performed best in terms of accuracy or F1-score for disease prediction, with suggestions for practical use in medical settings. Future work may involve improving data collection, parameter tuning, and exploring alternative machine learning algorithms.