

Project Title	Problem Statement Two(2) Healthcare
Technologies	Machine Learning Technology
Domain	AI and ML

Problem Statement:

For healthcare companies like Apollo Hospitals Enterprise Limited to plan effectively and operate efficiently, they must understand their patients' disease conditions and see how those conditions progress over time. To segment patients by their disease conditions and severity of illnesses, the healthcare company's management team approached their AI team for machine learning applications (MLOps). In a meeting with the management team of the healthcare company, it was suggested that the machine learning model be trained and tested with a variety of data sets as needed. As a result, business users should be able to upload training data to MLOps and select features through the user interface (UI). Users should also be able to upload and preview test data to test the model. Explanations AI functionality should be implemented by MLOps to help business users understand what the model outcomes mean. To simplify and better understand model outcomes, business users requested visual data analysis functionality.

Document Link: <https://github.com/Subi13/RIT-HACK>

User Interface and Deployment:

Implemented Using Streamlit Platform.

Code:

```
!ngrok authtoken 2XW7ZY5v0dsc17IVyhm6KnWd0wW_89tFPKaR6yqzw4AxH9nbX
```

```
from pyngrok import ngrok
!streamlit run app.py&>/dev/null&
!pgrep streamlit
```

```
public_url = ngrok.connect(port='8501')
public_url
```

```
%%writefile app.py
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
st.title("Welcome to HEALTH CARE SYSTEM")
#option=st.selectbox('Select Feature',('Cough','Fatigue','Fever','Blood Pressure','Cholesterol
Level','Difficulty Breathing'))
```

```

uploaded_file = st.file_uploader("Upload a CSV file", type=["csv"])

# Check if a file has been uploaded
if uploaded_file is not None:

    selected_values = []

    if st.checkbox('Cough'):
        selected_values.append('Cough')
    if st.checkbox('Fever'):
        selected_values.append('Fever')

    if st.checkbox('Fatigue'):
        selected_values.append('Fatigue')

    if st.checkbox('Blood Pressure'):
        selected_values.append('Blood Pressure')

    if st.button("Predict"):
        st.write("Predicting....")
        # Read the uploaded CSV file
        df = pd.read_csv(uploaded_file)
        # Display the data
        st.write("Uploaded CSV data:")
        st.write("Number of features selected: ",len(selected_values))
        x = df[selected_values].values
        y= df.iloc[:, 9].values
        from sklearn.model_selection import train_test_split
        x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25, random_state=0)
        #feature Scaling
        from sklearn.preprocessing import StandardScaler
        st_x= StandardScaler()
        x_train= st_x.fit_transform(x_train)
        x_test= st_x.transform(x_test)
        from sklearn.linear_model import LogisticRegression
        classifier= LogisticRegression(random_state=0)
        classifier.fit(x_train, y_train)
        y_pred= classifier.predict(x_test)
        from sklearn.metrics import accuracy_score
        accuracy = accuracy_score(y_test, y_pred)
        st.write("Accuracy is",accuracy)
        count =df['Disease'].value_counts()
        count

```