# Disease Mortality Prediction: “Hepatitis B”

A PROJECT REPORT

*Submitted by*

**Aarti Deokar**

*in partial fulfilment for the award of the degree*

*Of*

**M.Sc. Data Science and Artificial Intelligence – Part 2**

****

**RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS,**

**SCIENCE & COMMERCE (AUTONOMOUS),**

**GHATKOPAR W**

**April-2024**

**RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ART’S, SCIENCE & COMMERCE (AUTONOMOUS),**

**GHATKOPAR (W)**

****

***(Affiliated to University of Mumbai)***

**Certificate**

*This is to certify that the Project entitled* ***“Disease Mortalilty Prediction:Hepatitis B”*** *is bonafide work of* ***Aarti Deokar*** *bearing Seat No* ***730*** *submitted in partial fulfilment of the requirements for the award of Degree* ***Master of Science*** *in* ***Data Science & Artificial Intelligence.***

**Signature of Internal Guide Signature of Co-Ordinator**

**College Seal and Date Signature Examiner**

**ABSTRACT**

Hepatitis B is a serious liver infection caused by the hepatitis B virus (HBV). For most people, hepatitis B is short term, also called acute, and lasts less than six months. But for others, the infection becomes chronic, meaning it lasts more than six months. Having chronic hepatitis B increases your risk of developing liver failure, liver cancer or cirrhosis, a condition that permanently scars the liver.

Most adults with hepatitis B recover fully, even if their symptoms are severe. Infants and children are more likely to develop a long-lasting hepatitis B infection. This is known as a chronic infection.

**ACKNOWLEDGEMENT**

Before we get into the thick of things, we would like to add a few heartfelt words for the people who were part of the ***Disease Mortality Prediction: Hepatitis B*** project in numerous ways, people who gave unending support right from the stage the project idea was conceived.

A project report is such a comprehensive coverage; it would not have been materialized without the help of many. The four things that go on to make a successful endeavour are dedication, hard work, patience and correct guidance. Able and timely guidance not only helps in making an effort fruitful but also transforms the whole process of learning and implementing into an enjoyable experience.

In particular, I would like to thank our **Mentor/Director Dr. (Mrs.) Usha Mukundan**, R.J. College. I would like to give a very special honour and respect to our professor,(Mrs) **Sujata Kotian** who took keen interest in checking the minute details of the project work and guided us throughout the same. A sincere quote of thanks to the non-teaching staff for providing us software their time.

**DECLARATION**

I hereby declare that the Project entitled, ***“Disease Mortalilty Prediction:Hepatitis B”*** done at R. J. COLLEGE, Ghatkopar(W), Mumbai, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my Knowledge other than me, No one has submitted to any other University. The Project is done in partial fulfilment of the requirements for the award of degree of ***Master of Science Data Science and Artificial Intelligence*** to be submitted as a major project as part of our curriculum.

**Aarti Deokar**

**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| SR. NO. |  | CHAPTER NAME | PAGE NO. |
|  |  | ABSTRACT | 3 |
|  |  | ACKNOWLEDGEMENT | 4 |
|  |  | DECLARATION | 5 |
| 1. |  | INTRODUCTION | 6 |
| 2. |  | OBJECTIVE | 9 |
| 3. |  | METHODOLOGY | 10 |
|  | 3.1 | MODEL BUILDING | 11 |
|  | 3.2 | Dataset | 12 |
| 4. |  | REQUIREMNETS | 15 |
| 5. |  | CODE | 16 |
| 6. |  | DEPLOYMENT | 25 |
| 7. |  | IMPLEMENTATION | 26 |
| 8. |  | CONCLUSION | 30 |
| 9. |  | REFERENCES | 30 |

**INTRODUCTION**

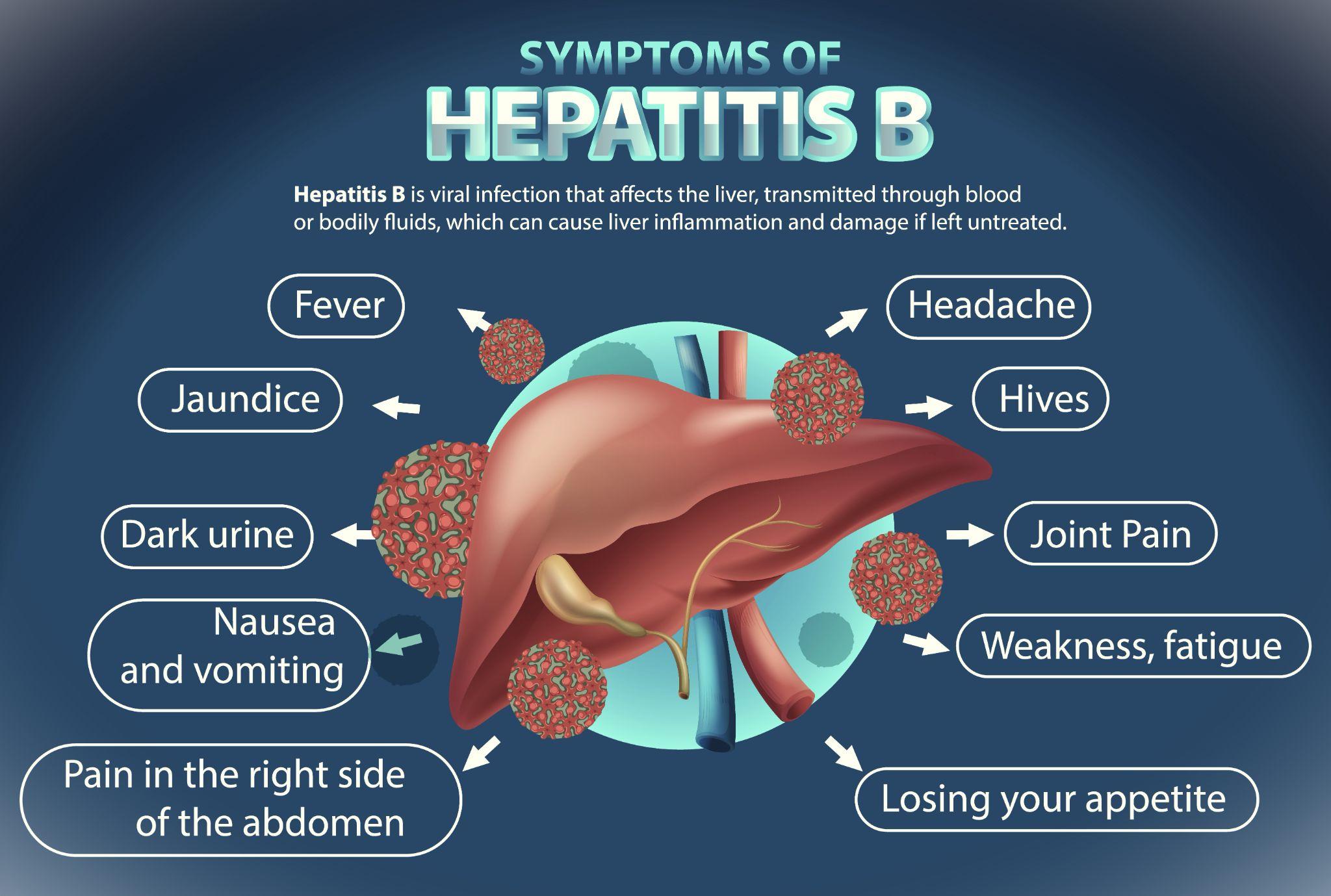
Hepatitis B is a serious liver infection caused by the hepatitis B virus (HBV). For most people, hepatitis B is short term, also called acute, and lasts less than six months. But for others, the infection becomes chronic, meaning it lasts more than six months. Having chronic hepatitis B increases your risk of developing liver failure, liver cancer or cirrhosis — a condition that permanently scars the liver.

Most adults with hepatitis B recover fully, even if their symptoms are severe. Infants and children are more likely to develop a long-lasting hepatitis B infection. This is known as a chronic infection.

A vaccine can prevent hepatitis B, but there's no cure if you have the condition. If you're infected, taking certain precautions can help prevent spreading the virus to others

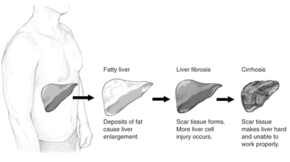
Hepatitis B signs and symptoms may include:

* Abdominal pain
* Dark urine
* Fever
* Joint pain
* Loss of appetite



**Laboratory Testing:**

Diagnosis is based on clinical, laboratory, and epidemiologic findings. HBV infection cannot be differentiated based on clinical symptoms alone, and definitive diagnosis depends on the results of serologic testing. Serologic markers of HBV infection vary depending on whether the infection is acute or chronic.

**Stages of Liver Disease:**

**Medical Management:**

There is no specific therapy for acute HBV infection. Treatment is supportive. Guidelines for management of chronic HBV infection in children and adults, including disease monitoring and antiviral therapy, are available from the American Association for the Study of Liver Diseases (AASLD, https://www. AASLD.org).

**Vaccination:**

***Routine vaccination***

• All newborns within 24 hours of birth

• All children and teens ages 0 through 18 years

• All adults through age 59 years

**OBJECTIVE**

**The main objectives of our project is :**

“To predict if a patient will *live* or *die* based on the parameters using Machine Learning.”

* ***Parameters:***
* Class
* AGE
* SEX
* STEROID
* ANTIVIRALS
* FATIGUE
* MALAISE
* ANOREXIA
* LIVER BIG
* LIVER FIRM
* SPLEEN PALPABLE
* SPIDERS
* ASCITES
* VARICES
* BILIRUBIN
* ALK PHOSPHATASE
* SGOT
* ALBUMIN
* PROTIME
* HISTOLOGY

**METHODOLOGY**

**Data Preparation:** is the process of transforming raw data so that data scientists and analysts can run it through machine learning algorithms to uncover insights or make predictions**.**

**EDA:** Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

**Feature Selection:** the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data.It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve

**Model Building:** Building a model in machine learning is creating a mathematical representation by generalizing and learning from training data. Then, the built machine learning model is applied to new data to make predictions and obtain results.

**Interpret Model:** is used to find out ways to understand model decision making policies better. This is to enable fairness, accountability and transparency which will give humans enough confidence to use these models in real-world problems which have a lot of impact on business and society.

**Serialization:** Serialization refers to the process of converting a data object into a format that allows us to store or transmit the data and then recreate the object when needed using the reverse process of deserialization.

**Deployment:** Model deployment is the process of implementing a fully functioning machine learning model into production where it can make predictions based on data.

**MODEL BUILDING**

***Supervised Learning Model:***

Supervised learning is the type of machine learning in which machines are trained using well "labeled" training data, and on the basis of that data, machines predict the output. The labeled data means some input data is already tagged with the correct output.

***Classification model:***

A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.

As the dataset we have is of classification type and the models used to build the whole project are as follows:

1. ***K-Nearest Neighbour:***

Neighbour based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbors of each point.

1. ***Decision Tree:***

Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

1. ***Logistic Regression:***

Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

P(Y=1|X) or P(Y=0|X)

It calculates the probability of dependent variable Y, given independent variable X

**DATASET**

We have retrieved the “clean\_hepatitis\_dataset.csv” file dataset from [Kaggle.com](https://www.kaggle.com/) website.

*Following is the link to download the dataset:*

[Hepatitis\_Dataset](https://www.kaggle.com/datasets/harinir/hepatitis)

**Attribute information:**

1. *Class:* DIE, LIVE

2. *AGE:* 10, 20, 30, 40, 50, 60, 70, 80

3. *SEX:* male, female

4. ***STEROID:*** no, yes

5. *ANTIVIRALS:* no, yes

6. *FATIGUE*: no, yes

7. *MALAISE:* no, yes

8. *ANOREXIA:* no, yes

9. *LIVER BIG:* no, yes

10. *LIVER FIRM:* no, yes

11. *SPLEEN PALPABLE:* no, yes

12. *SPIDERS:* no, yes

13. ASCITES: no, yes

14. VARICES: no, yes

15. BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00

16. ALK PHOSPHATE: 33, 80, 120, 160, 200, 250

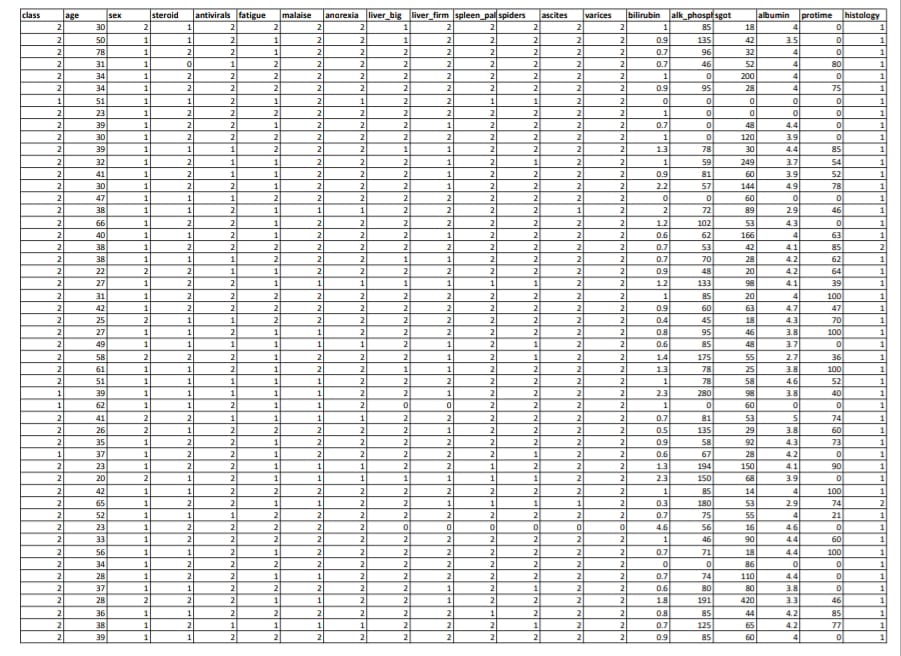
17. SGOT: 13, 100, 200, 300, 400, 500,

18. ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0

19. PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90

20. HISTOLOGY: no, yes

**clean\_hepatitis\_dataset.csv**

****

**REQUIREMENTS**

* **Software Requirements:**
  + Visual studio Code
  + Jupyter Notebook/Stream lit
* **Hardware Requirements:**

Minimum Requirements / Recommended:

* + Operating System - Microsoft Windows 10 /11
  + Processor - Intel(R) Core(TM) i5,i7,i9,Ryzen 5,7
  + Installed Physical Memory (RAM)- Up to 8.00 GB
  + Storage- 512 SSD /HDD

**CODE**

**App.py**

# ---pip install pipreqs---

# to download requirement file

# pipreqs ./

# Core Pkgs

import pickle

from pathlib import Path

import streamlit as st

# EDA Pkgs

import pandas as pd

import numpy as np

# Utils

import os

import joblib

# passlib,bcrypt

# Data Viz Pkgs

import matplotlib.pyplot as plt

import matplotlib

matplotlib.use('Agg')

import streamlit\_authenticator as stauth

from PIL import Image

# ML Interpretation

import lime

import lime.lime\_tabular

st.set\_option('deprecation.showPyplotGlobalUse', False)

st.set\_page\_config(page\_title="Hepatitis Mortality App",page\_icon="⚕️",layout="wide")

html\_temp = """

        <div style="background-color:darkblue;padding:10px;border-radius:5px">

        <h1 style="color:white;text-align:center;"> Disease Mortality Prediction </h1>

        <h3 style="color:white;text-align:center;"> "Hepatitis B" </h3>

        </div>

        """

result\_temp2 = """

    <div style="background-color:#464e5f;padding:10px;border-radius:10px;margin:10px;">

    <h4 style="color:white;text-align:center;">Algorithm:: {}</h4>

    <img src="https://www.w3schools.com/howto/{}" alt="Avatar" style="vertical-align: middle;float:left;width: 50px;height: 50px;border-radius: 50%;" >

    <br/>

    <br/>

    <p style="text-align:justify;color:white">{} % probabilty that Patient {}s</p>

    </div>

    """

prescriptive\_message\_temp = """

    <div style="background-color:silver;overflow-x: auto; padding:10px;border-radius:5px;margin:10px;">

        <h3 style="text-align:justify;color:black;padding:10px">Recommended Life style modification</h3>

        <ul>

        <li style="text-align:justify;color:black;padding:10px">Exercise Daily</li>

        <li style="text-align:justify;color:black;padding:10px">Get Plenty of Rest</li>

        <li style="text-align:justify;color:black;padding:10px">Exercise Daily</li>

        <li style="text-align:justify;color:black;padding:10px">Avoid Alchol</li>

        <li style="text-align:justify;color:black;padding:10px">Proper diet</li>

        <ul>

        <h3 style="text-align:justify;color:black;padding:10px">Medical Mgmt</h3>

        <ul>

        <li style="text-align:justify;color:black;padding:10px">Consult your doctor</li>

        <li style="text-align:justify;color:black;padding:10px">Take your interferons</li>

        <li style="text-align:justify;color:black;padding:10px">Go for checkups</li>

        <ul>

    </div>

    """

descriptive\_message\_temp = """

    <div style="background-color:lightgray;overflow-x: auto; padding:10px;border-radius:5px;margin:10px;">

        <p style="text-align:justify">Hepatitis B is a viral infection that attacks the liver and can cause both acute and chronic disease i.e. Hepatitis B can range from a mild illness, lasting a few weeks, to a serious,

        life-long (chronic) condition.

        Hepatitis B is primarily spread when blood, semen, or certain other body fluids – even in microscopic amounts – from a person infected with the hepatitis B virus enters the body of someone who is not infected.

        Acute hepatitis B is a short-term illness that occurs within the first 6 months after someone is exposed to the hepatitis B virus. Some people with acute hepatitis B have no symptoms at all or only mild illness.

        For others, acute hepatitis B causes a more severe illness that requires hospitalization.<br>

        <br>

        The hepatitis B virus can also be transmitted by :-</p>

        <ul>

        <li style="text-align:justify;color:black">Birth to an infected pregnant person.</li>

        <li style="text-align:justify;color:black">Sex with an infected person.</li>

        <li style="text-align:justify;color:black">Sharing personal items such as toothbrushes or razors, but is less common.</li>

        <li style="text-align:justify;color:black">Poor infection control in health care facilities.</li>

        <li style="text-align:justify;color:black">Direct contact with the blood or open sores of a person who has hepatitis B, etc.</li>

        <ul>

        </div>

    """

names = ["Aarti Deokar", "Anushka Tawde"]

usernames = ["sYadav", "aTawde"]

feature\_names\_best = ['age', 'sex', 'steroid', 'antivirals', 'fatigue', 'spiders', 'ascites', 'varices', 'bilirubin',

                      'alk\_phosphate', 'sgot', 'albumin', 'protime', 'histology']

gender\_dict = {"male": 1, "female": 2}

feature\_dict = {"No": 1, "Yes": 2}

def get\_value(val, my\_dict):

    for key, value in my\_dict.items():

        if val == key:

            return value

def get\_key(val, my\_dict):

    for key, value in my\_dict.items():

        if val == key:

            return key

def get\_fvalue(val):

    feature\_dict = {"No": 1, "Yes": 2}

    for key, value in feature\_dict.items():

        if val == key:

            return value

def load\_model(model\_file):

    loaded\_model = joblib.load(open(os.path.join(model\_file), "rb"))

    return loaded\_model

def main():

    """Hep Mortality Prediction App"""

    st.sidebar.title("Hepatitis B Disease App")

    st.markdown(html\_temp.format('royalblue'), unsafe\_allow\_html=True)

    menu = ["Home", "Login"]

    sub\_menu = ["Plot", "Prediction"]

    st.sidebar.header(f"Please Filter Here:")

    choice = st.sidebar.selectbox("Select pages:", menu)

    if choice == "Home":

        st.write("")

        st.write("")

        #Loading image

        image = Image.open('hepDis.jpg')

        st.image(image, caption='Human Liver')

        st.write("")

        st.markdown("<h2 >What is Hepatitis?</h2>", unsafe\_allow\_html=True)

        st.markdown(descriptive\_message\_temp, unsafe\_allow\_html=True)

        st.write("")

        st.write("")

        st.write("")

        image2 = Image.open('hepimage.jpg')

        st.image(image2, caption='Symptoms of hepatitis')

        #st.image(load\_image('hepimage.jpg'))

    elif choice == "Login":

        #username = st.sidebar.text\_input("Username")

        #password = st.sidebar.text\_input("Password", type='password')

        #if st.checkbox("Login"):

            file\_path = Path(\_\_file\_\_).parent / "hashed\_pws.pkl"

            with file\_path.open("rb") as file:

                hashed\_passwords = pickle.load(file)

            authenticator = stauth.Authenticate(names, usernames, hashed\_passwords,

                                                "sales\_dashboard", "abcdef",cookie\_expiry\_days=30)

            name, authentication\_status, username = authenticator.login("Login", "main")

            if authentication\_status == False:

                st.error("Username/password is incorrect")

            if authentication\_status == None:

                st.warning("Please enter your username and password")

            if authentication\_status:

                authenticator.logout("Logout", "sidebar")

                st.sidebar.title(f"Welcome {name}")

                activity = st.sidebar.selectbox("Activity", sub\_menu)

                if activity == "Plot":

                    st.markdown("<h2 style='text-align: center; color: darkblue;'>Data Visualization </h2>",

                                unsafe\_allow\_html=True)

                    df = pd.read\_csv("data/clean\_hepatitis\_dataset.csv")

                    if st.checkbox("View Dataset"):

                        st.dataframe(df)

                    st.write("#### Below is the description of our Dataset:")

                    st.write(df.describe())

                    st.markdown("""---""")

                    st.write("#### Bar Plot: ")

                    st.write("feature 'class' = {'Die':1, 'Live':2} ")

                    df['class'].value\_counts().plot(kind='bar')

                    st.pyplot()

                    st.markdown("""---""")

                    # Freq Dist Plot

                    freq\_df = pd.read\_csv("data/freq\_df\_hepatitis\_dataset.csv")

                    st.write("#### Frequency Distribution Plot")

                    st.write("based on 'age' variable")

                    if st.checkbox("View groupwise Count of the above:"):

                        st.dataframe(freq\_df)

                    st.bar\_chart(freq\_df['count'])

                    st.markdown("""---""")

                    if st.checkbox("Area Chart"):

                        all\_columns = df.columns.to\_list()

                        feat\_choices = st.multiselect("Choose a Feature", all\_columns)

                        new\_df = df[feat\_choices]

                        st.write(f"##### Area chart ")

                        st.area\_chart(new\_df)

                elif activity == "Prediction":

                    st.subheader("Predictive Analytics")

                    age = st.number\_input("Age", 7, 80)

                    sex = st.radio("Sex", tuple(gender\_dict.keys()))

                    steroid = st.radio("Do You Take Steroids?", tuple(feature\_dict.keys()))

                    antivirals = st.radio("Do You Take Antivirals?", tuple(feature\_dict.keys()))

                    fatigue = st.radio("Do You Have Fatigue", tuple(feature\_dict.keys()))

                    spiders = st.radio("Presence of Spider Naeve", tuple(feature\_dict.keys()))

                    ascites = st.selectbox("Ascities", tuple(feature\_dict.keys()))

                    varices = st.selectbox("Presence of Varices", tuple(feature\_dict.keys()))

                    bilirubin = st.number\_input("bilirubin Content", 0.0, 8.0)

                    alk\_phosphate = st.number\_input("Alkaline Phosphate Content", 0.0, 296.0)

                    sgot = st.number\_input("Sgot", 0.0, 648.0)

                    albumin = st.number\_input("Albumin", 0.0, 6.4)

                    protime = st.number\_input("Prothrombin Time", 0.0, 100.0)

                    histology = st.selectbox("Histology", tuple(feature\_dict.keys()))

                    feature\_list = [age, get\_value(sex, gender\_dict), get\_fvalue(steroid), get\_fvalue(antivirals),

                                    get\_fvalue(fatigue), get\_fvalue(spiders), get\_fvalue(ascites), get\_fvalue(varices),

                                    bilirubin, alk\_phosphate, sgot, albumin, int(protime), get\_fvalue(histology)]

                    st.write(len(feature\_list))

                    st.write(feature\_list)

                    pretty\_result = {"age": age, "sex": sex, "steroid": steroid, "antivirals": antivirals,

                                     "fatigue": fatigue, "spiders": spiders, "ascites": ascites, "varices": varices,

                                     "bilirubin": bilirubin, "alk\_phosphate": alk\_phosphate, "sgot": sgot,

                                     "albumin": albumin, "protime": protime, "histolog": histology}

                    st.json(pretty\_result)

                    single\_sample = np.array(feature\_list).reshape(1, -1)

                    # ML

                    model\_choice = st.selectbox("Select Model", ["LR", "KNN", "DecisionTree"])

                    if st.button("Predict"):

                        if model\_choice == "KNN":

                            loaded\_model = load\_model("models/knn\_hepB\_model.pkl")

                            prediction = loaded\_model.predict(single\_sample)

                            pred\_prob = loaded\_model.predict\_proba(single\_sample)

                        elif model\_choice == "DecisionTree":

                            loaded\_model = load\_model("models/decision\_tree\_clf\_hepB\_model.pkl")

                            prediction = loaded\_model.predict(single\_sample)

                            pred\_prob = loaded\_model.predict\_proba(single\_sample)

                        else:

                            loaded\_model = load\_model("models/logistic\_regression\_hepB\_model.pkl")

                            prediction = loaded\_model.predict(single\_sample)

                            pred\_prob = loaded\_model.predict\_proba(single\_sample)

                        # st.write(prediction)

                        # prediction\_label = {"Die":1,"Live":2}

                        # final\_result = get\_key(prediction,prediction\_label)

                        if prediction == 1:

                            st.warning("Patient Dies")

                            pred\_probability\_score = {"Die": pred\_prob[0][0] \* 100, "Live": pred\_prob[0][1] \* 100}

                            st.subheader("Prediction Probability Score using {}".format(model\_choice))

                            st.json(pred\_probability\_score)

                            st.subheader("Prescriptive Analytics")

                            st.markdown(prescriptive\_message\_temp, unsafe\_allow\_html=True)

                        else:

                            st.success("Patient Lives")

                            pred\_probability\_score = {"Die": pred\_prob[0][0] \* 100, "Live": pred\_prob[0][1] \* 100}

                            st.subheader("Prediction Probability Score using {}".format(model\_choice))

                            st.json(pred\_probability\_score)

                    if st.checkbox("Interpret"):

                        if model\_choice == "KNN":

                            loaded\_model = load\_model("models/knn\_hepB\_model.pkl")

                        elif model\_choice == "DecisionTree":

                            loaded\_model = load\_model("models/decision\_tree\_clf\_hepB\_model.pkl")

                        else:

                            loaded\_model = load\_model("models/logistic\_regression\_hepB\_model.pkl")

                            # loaded\_model = load\_model("models/logistic\_regression\_model.pkl")

                            # 1 Die and 2 Live

                            df = pd.read\_csv("data/clean\_hepatitis\_dataset.csv")

                            x = df[['age', 'sex', 'steroid', 'antivirals', 'fatigue', 'spiders', 'ascites', 'varices',

                                    'bilirubin', 'alk\_phosphate', 'sgot', 'albumin', 'protime', 'histology']]

                            feature\_names = ['age', 'sex', 'steroid', 'antivirals', 'fatigue', 'spiders', 'ascites',

                                             'varices', 'bilirubin', 'alk\_phosphate', 'sgot', 'albumin', 'protime',

                                             'histology']

                            class\_names = ['Die(1)', 'Live(2)']

                            explainer = lime.lime\_tabular.LimeTabularExplainer(x.values, feature\_names=feature\_names,

                                                                               class\_names=class\_names,

                                                                               discretize\_continuous=True)

                            # The Explainer Instance

                            exp = explainer.explain\_instance(np.array(feature\_list), loaded\_model.predict\_proba,

                                                             num\_features=13, top\_labels=1)

                            exp.show\_in\_notebook(show\_table=True, show\_all=False)

                            # exp.save\_to\_file('lime\_oi.html')

                            st.write(exp.as\_list())

                            new\_exp = exp.as\_list()

                            label\_limits = [i[0] for i in new\_exp]

                            # st.write(label\_limits)

                            label\_scores = [i[1] for i in new\_exp]

                            plt.barh(label\_limits, label\_scores)

                            st.pyplot()

                            plt.figure(figsize=(20, 10))

                            fig = exp.as\_pyplot\_figure()

                            st.pyplot()

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**DEPLOYMENT**

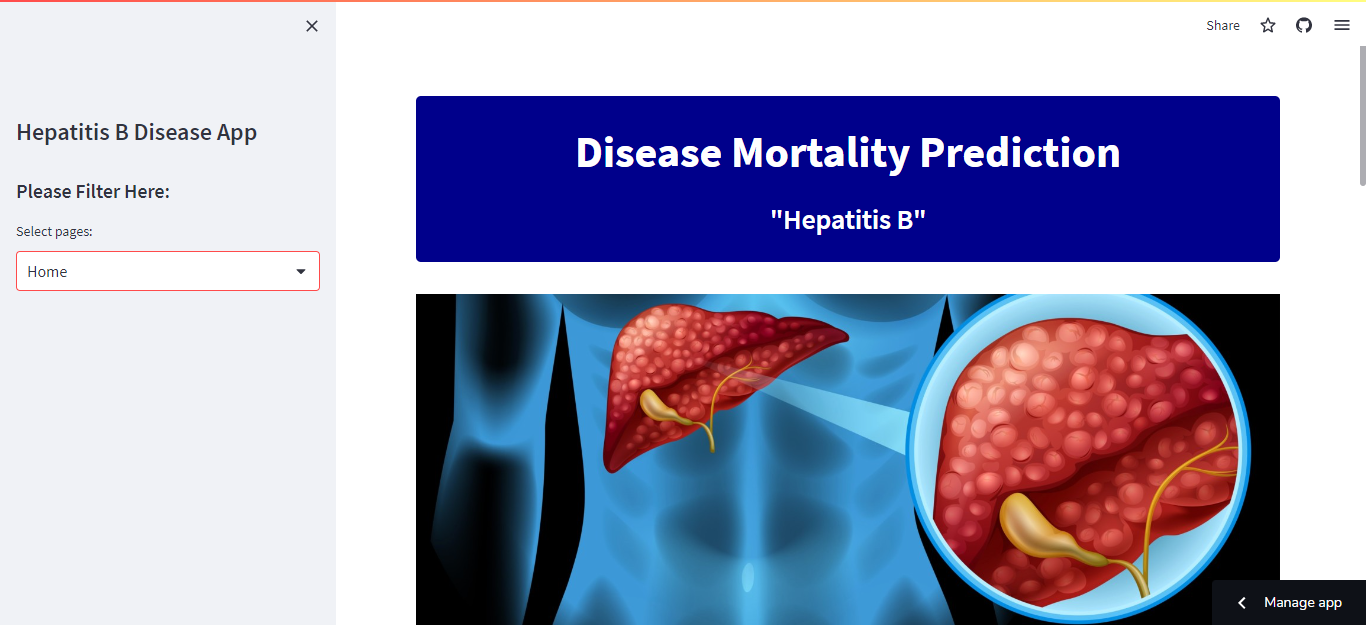
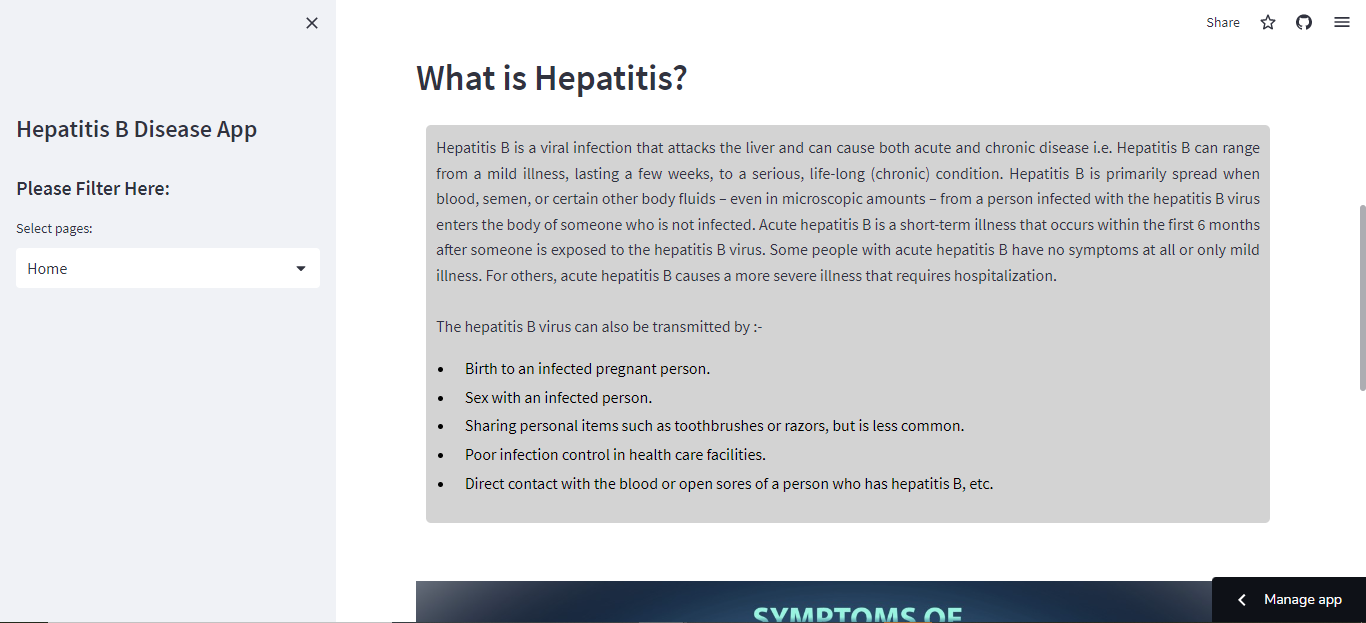
***Streamlit***

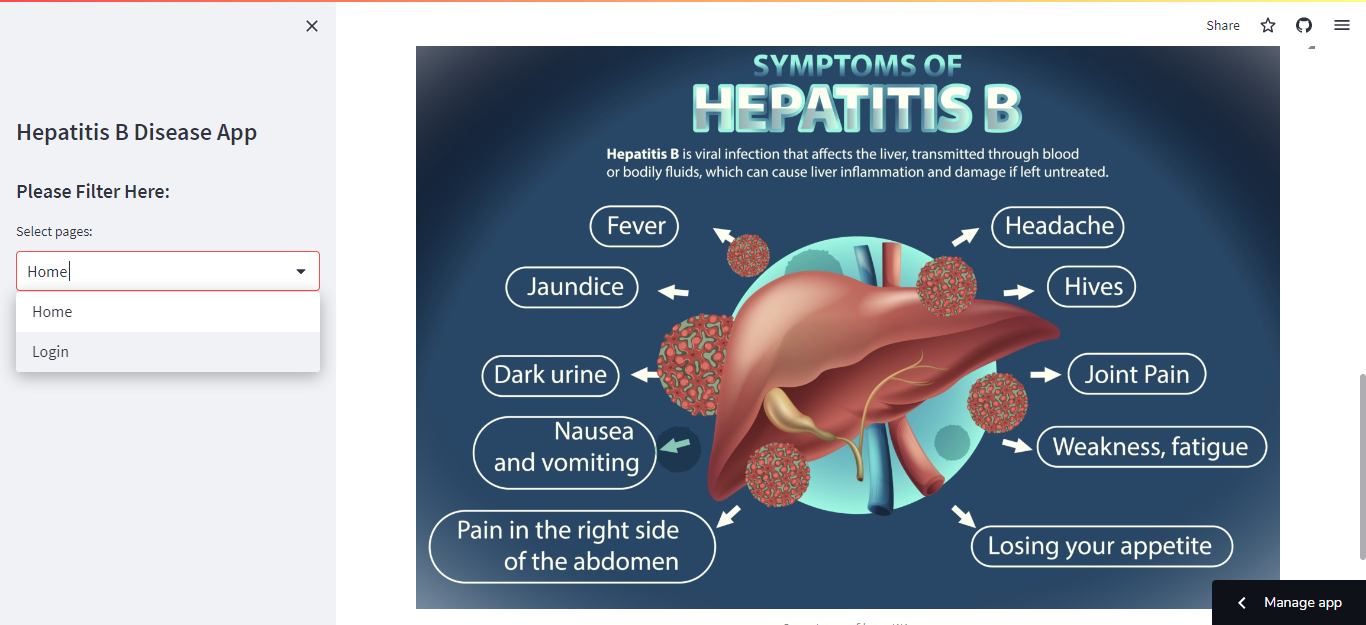
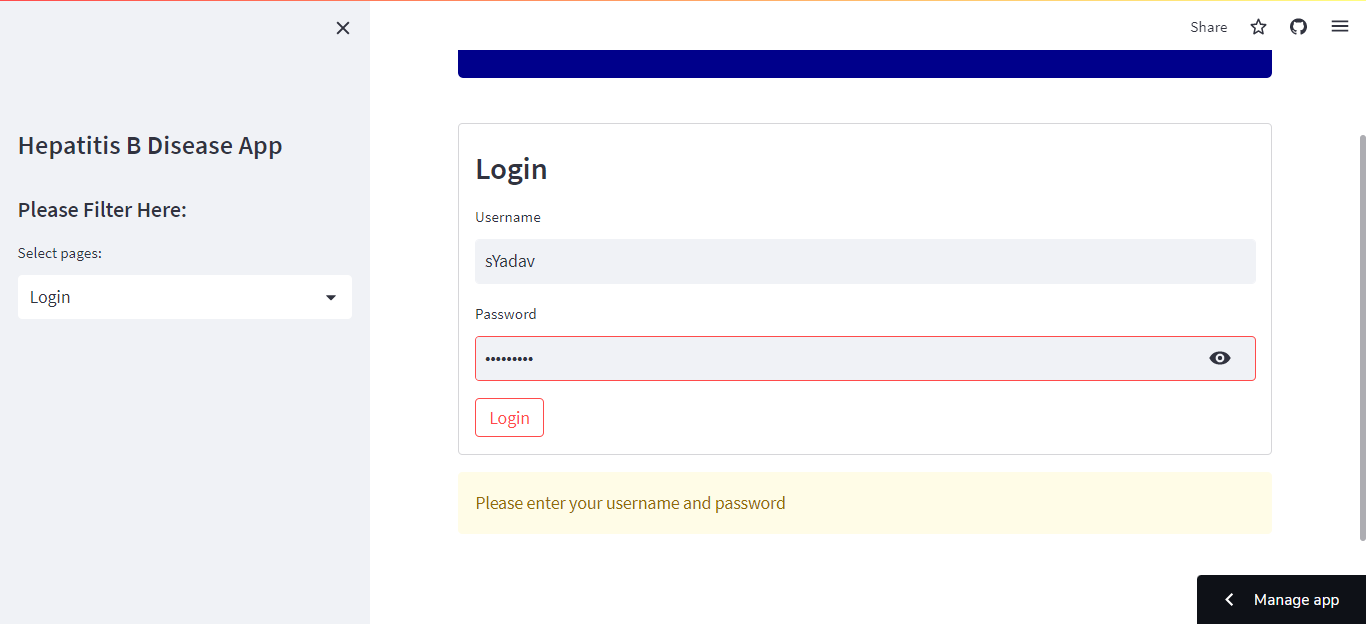
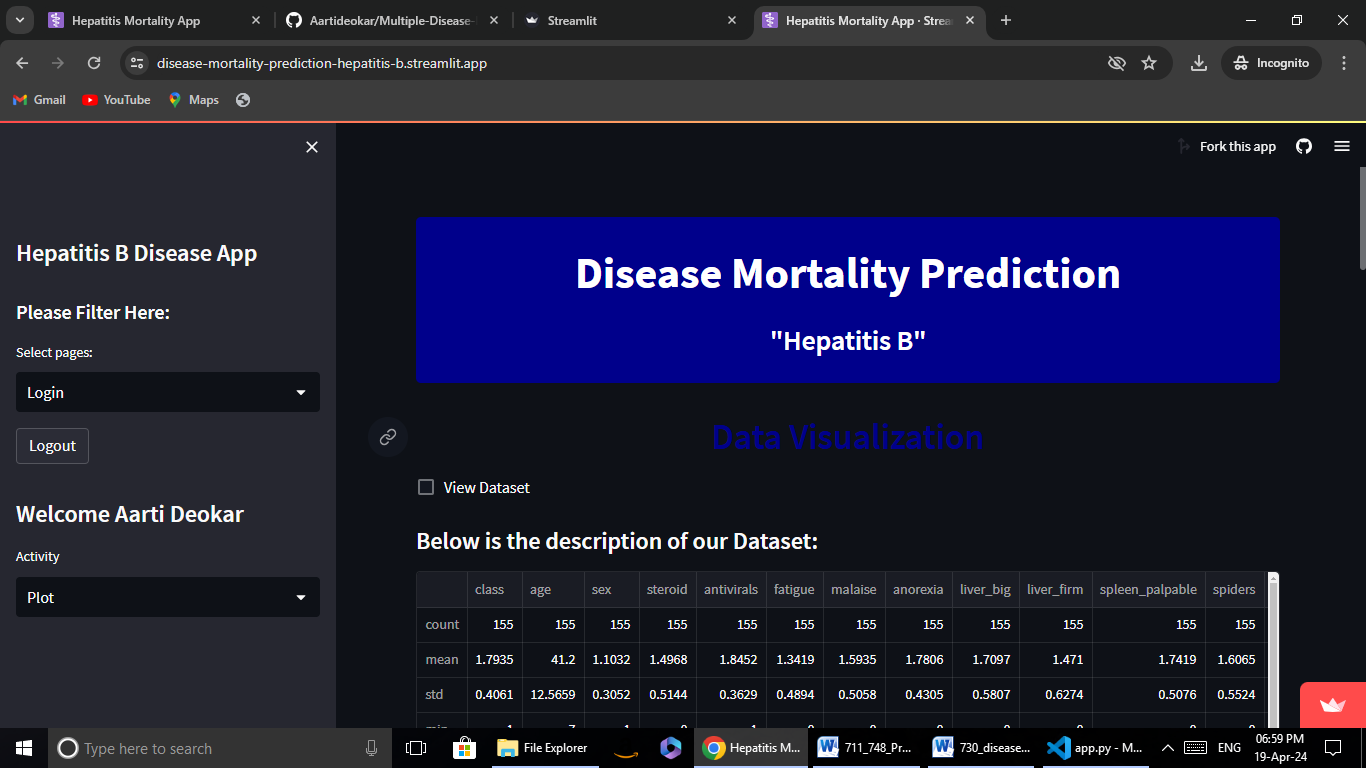
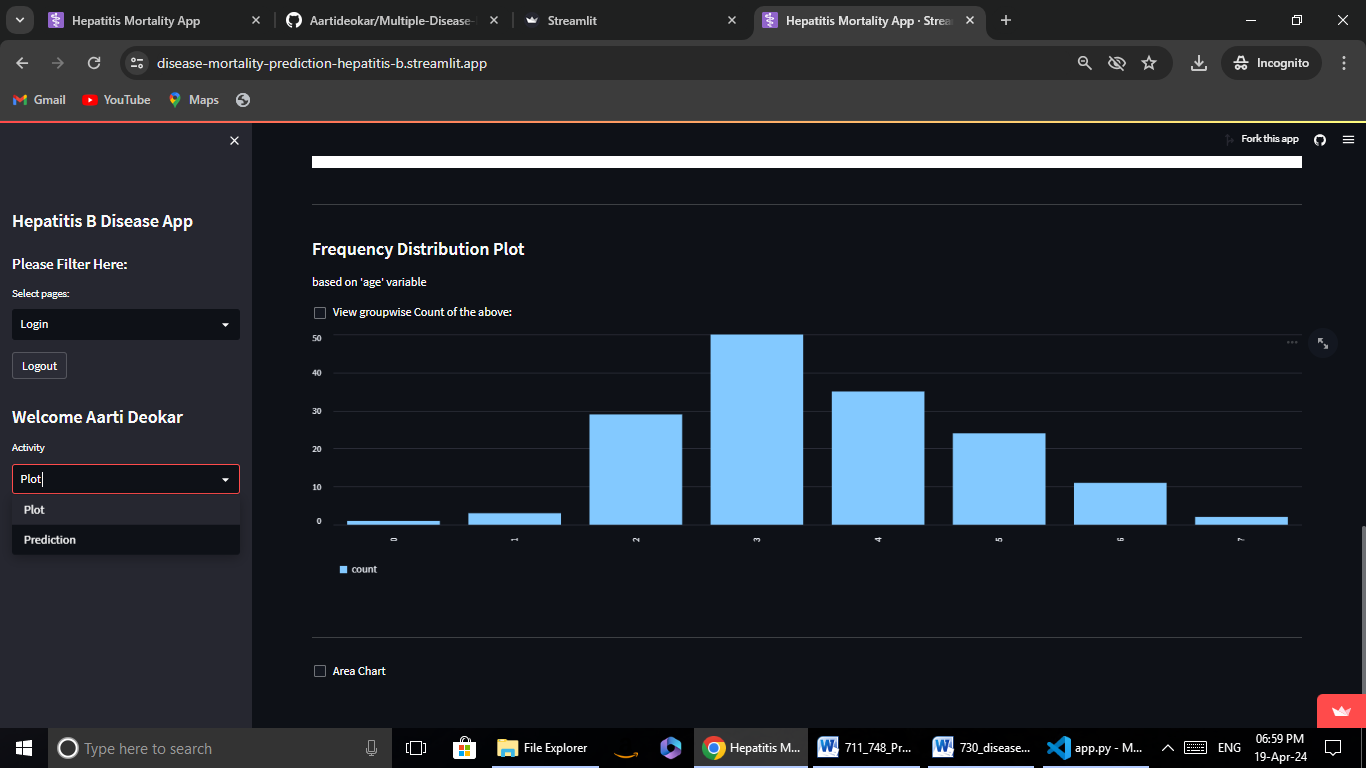
Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks to build web apps. Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modeling. Streamlit allows you to create a stunning-looking application with only a few lines of code.

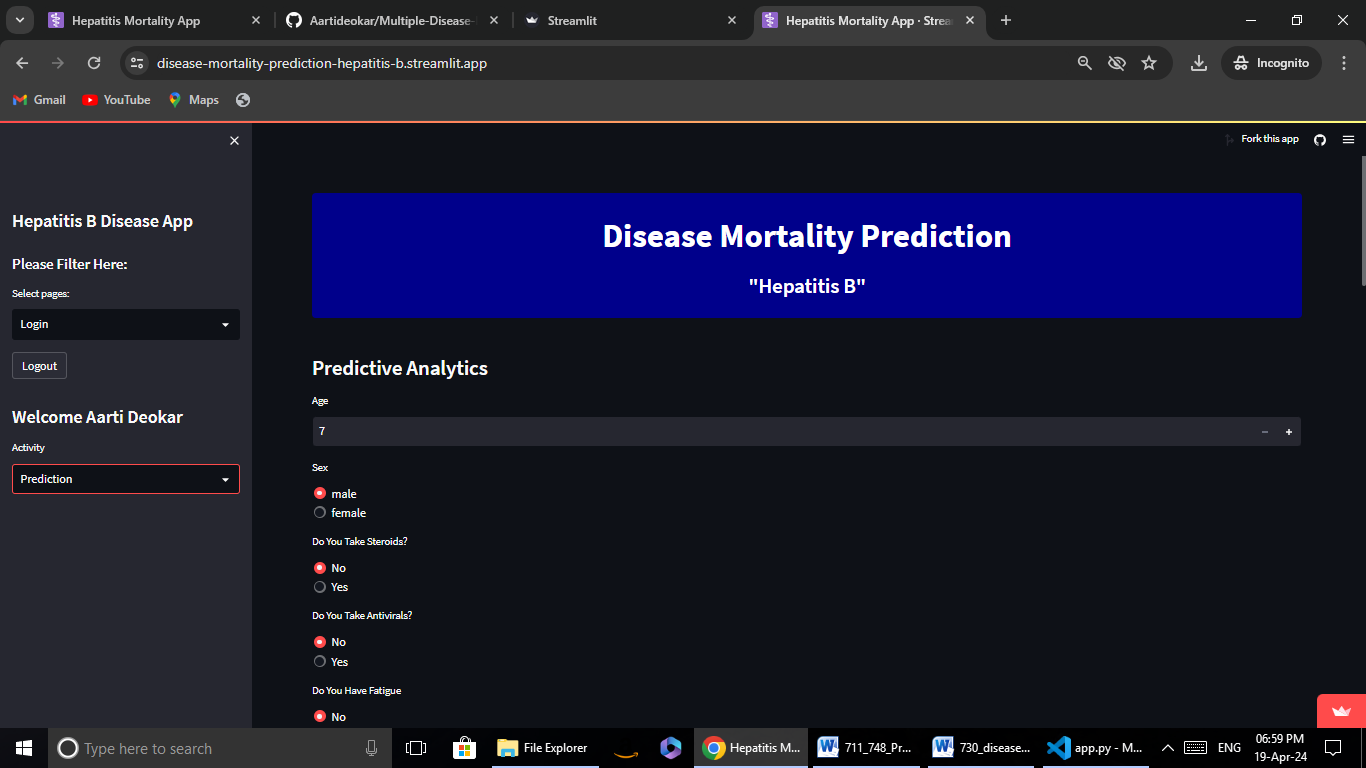
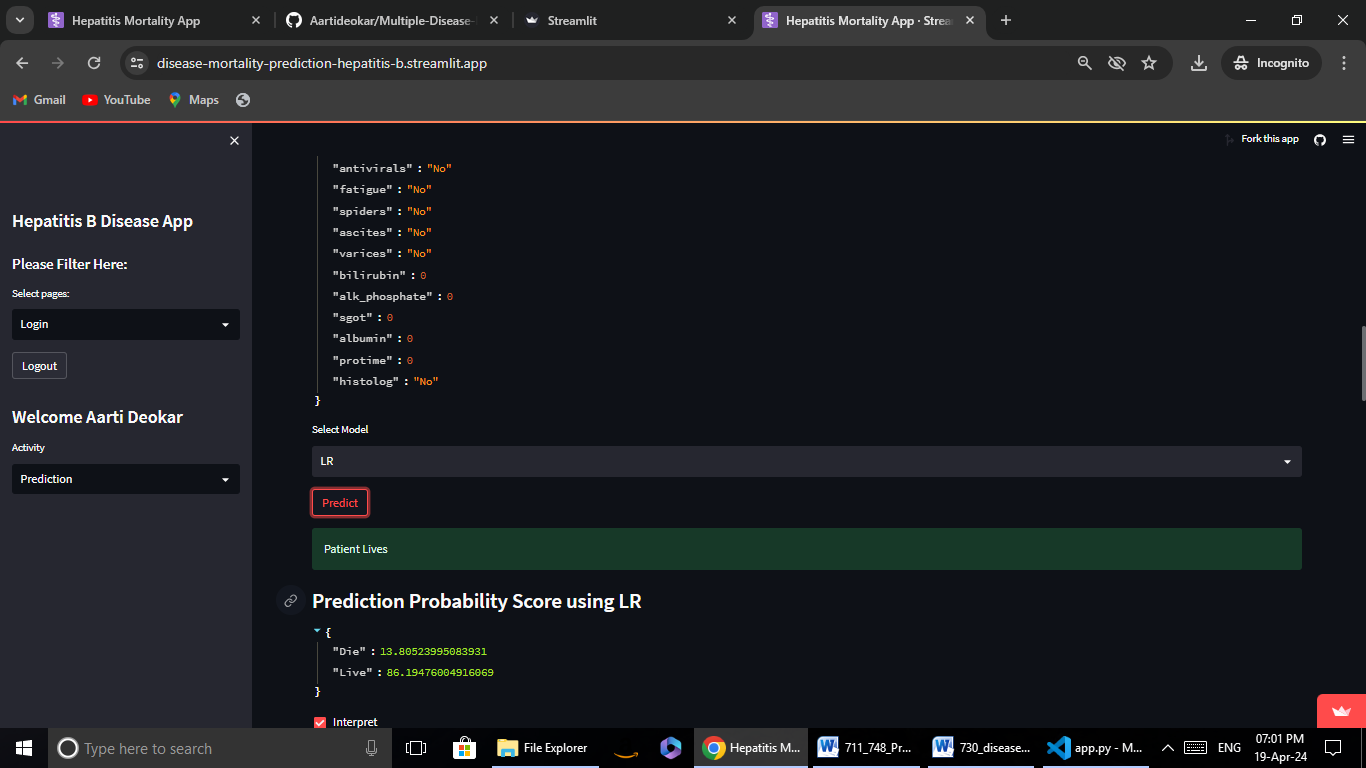
***Streamlit app* Link:**

<https://disease-mortality-prediction-hepatitis-b.streamlit.app/>

**IMPLEMENTATION**

* **Web applicatio****n**

**CONCLUSION**

* Most of the data points were in objects.
* Two of them were of float type.
* There were more males than females in our dataset.
* Highest prevalence of Hepatitis is from 30-40 followed by 40-50.
* The least are the individuals under 10, and elderly above 70.
* All the methods gave us almost the same features to use i.e. all the features are important.

**REFERENCE**

* <https://www.cdc.gov/vaccines/pubs/pinkbook/downloads/hepb.pdf>
* <https://www.kaggle.com/datasets/harinir/hepatitis>
* <https://monkeylearn.com/blog/classification-algorithms/>
* <https://www.youtube.com/watch?v=HU_kd-1uIkQ&t=892s>
* <https://en.wikipedia.org/wiki/Hepatitis>