VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590010

INTERNSHIP ON

CSE CODING

Submitted in partial fulfilment for the requirements for the 3rd semester

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

For the Academic Year 2022-2023

Submitted by:

SATHVIK N G **1MV21CS091**

Under the guidance of

Dr. Anitha T.N

HOD, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

HUNASAMARANAHALLI,

BENGALURU-562157

OFFER LETTER

VIEH Private Limited

Sagar City Society, Andheri West, Mumbai, Maharashtra - 400058

May 4, 2023

Internship Offer Letter

Dear Sathvik N G,

On behalf of VIEH Private Limited, I am pleased to inform you that you have been considered for the Data Science internship. As a result, you will be working with us on the project **"Life Expectancy Prediction"** from May 4, 2023.

During your internship, the concentration will be on helping you understand the theoretical concepts with their practicals and implementations to help you connect your classroom knowledge and on-field experience. As a result, you will be proactively contributing to the above-selected project, besides product development and research field. In addition, you will be required to complete performance and learning goals for your current project with us.

With this offer letter, you acknowledge that you understand participation in this program is not an offer of employment, and successful completion of the program does not entitle you to an employment offer from us. We hope that your association with the company will be successful and rewarding.

We look forward to having you begin your career with us. Once again, congratulations to you on your selection, and all the best for your endeavors.

Regards,

Nooruddin Shaikh

CTO & Chief Data Scientist at VIEH

SIR M.VISVESVARAYA INSTITUTE OF TECHNOLOGY

Krishnadevaraya Nagar, International Airport Road,
Hunasmaranahalli, Bengaluru-562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

C E R T I F I C A T E

It is certified that the **INTERNSHIP** entitled **" Data Science "** is carried out by **1MV21CS091 – SATHVIK N G ,** authentic student of **Sir M Visvesvaraya Institute of Technology** in partial fulfilment for the 3th semester for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the academic year **2021-2022**.It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

Name & Signature of
HOD and Guide

Dr. Anitha T.N
HOD & Guide,
Dept. of CSE,
Sir MVIT,
Bengaluru – 562157

Name & Signature of
Principle

Dr. Rakesh S.G
Principal,
Sir MVIT,
Bengaluru – 562157

Name of the Mentors: Signature with Date

1) **Rekha B.N**, Associate Professor

2) **Kavyashree G.M**, Assistant Professor

DECLARATION

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date :

Signature of Student

SATHVIK N G

1MV21CS091

ACKNOWLEDGMENT

It gives us immense pleasure to express our sincere gratitude to the management of **Sir M. Visvesvaraya Institute of Technology**, Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

On the path of learning the presence of an experienced guide is indispensable and would like to thank my guide **Nooruddin Shaikh, CTO & Chief Data Scientist at VIEH Group** for his invaluable help and guidance.

I would also like to convey our regards to **Dr. Rakesh S G**, Principal, Sir MVIT for providing us with the infrastructure and facilities needed to develop our project.

Heartfelt and sincere thanks to **Dr. Anitha T.N**, HOD, Dept. of CSE, for his suggestions, constant support and encouragement.

I would also like to thank the staff of Department of Computer Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

ABSTRACT

This internship report presents the details of my project on life expectancy prediction. The report is divided into several chapters, each addressing a specific aspect of the project.

In the introduction chapter, I provide an overview of data science and its relevance to the project. Additionally, I discuss the background of the company and its role in the project.

Chapter 2 provides a brief overview of the project, including its general description, problem statement, proposed solution, and potential improvements. The chapter also covers the design details, including the process workflow, data collection, pre-processing, exploratory data analysis, feature engineering, model training, evaluation, selection, deployment, and testing. It concludes with documentation and additional resources for further reading.

Chapter 3 outlines the system requirements and specifications, including hardware and software requirements, as well as any additional tools and frameworks used in the project.

Chapter 4 focuses on the source code, providing templates, the Flask code (app.py), Jupyter Notebook code, and relevant screenshots.

The conclusion chapter summarizes the key findings and insights from the project, highlighting its significance and potential future developments.

Finally, the references section lists the sources and materials used throughout the project, ensuring proper citation and acknowledgment.

The internship report serves as a comprehensive documentation of the life expectancy prediction project, showcasing the methodologies, techniques, and outcomes achieved during the internship period.

TABLE OF CONTENTS

LIST OF FIGURES

TIMELINE OF INTERNSHIP

[04-05-2023 to 02-06-2023]

Week 1	04-05-2023 To 11-05-2023	1. Data Pre-processing 2. Handling Missing Values 3. Feature Selection 4. Exploratory Data Analysis 5. Statistical Analysis
Week 2	12-05-2023 To 19-05-2023	1. Model Selection 2. Linear Regression 3. Model Training 4. Model Evaluation 5. Model Interpretation
Week 3	20-05-2023 To 26-05-2023	1. Model Improvement 2. Residual Analysis 3. Model Deployment 4. Documentation

CHAPTER 1
INTRODUCTION

Data science is a rapidly growing field that involves extracting valuable insights and knowledge from large and complex datasets. It combines various disciplines such as statistics, mathematics, and computer science to analyse data, build predictive models, and make data-driven decisions. Data scientists use techniques like data mining, machine learning, and statistical analysis to uncover patterns, trends, and correlations in data. These insights are then used to solve complex problems, improve processes, and drive innovation across various industries. With the increasing availability of data and advancements in technology, data science plays a crucial role in enabling organizations to make informed decisions, optimize operations, and gain a competitive edge.

About the Company:

V.I.E.H Group is a multidimensional Cyber Security and IT company. With expertise in various domains, they offer a wide range of services including Web Application Penetration Testing (WAPT), Vulnerability Assessment and Penetration Testing (VAPT), Red Teaming, website development, software development, and private investigation. They excel in ensuring robust digital security, providing customized solutions to meet client needs, and collaborating with law enforcement agencies to solve complex cases. V.I.E.H Group stands out as a trusted partner for organizations seeking comprehensive cyber security solutions and reliable IT services.

CHAPTER 2

A BREIF OVERVIEW OF THE PROJECT

This data science project aims to predict life expectancy by analysing a diverse dataset that encompasses various health and socio-economic factors. The project explores the application of machine learning techniques to uncover patterns and insights related to life expectancy trends. Through this endeavour, valuable insights into public health can be gained. Here is an elaboration of each step in the life expectancy prediction project.

General Description

Product Perspective:

The life expectancy prediction project is a machine learning-based model that estimates individuals' life expectancy based on various factors. It provides valuable insights for healthcare, lifestyle decisions, and policy-making. By leveraging machine learning, it offers a predictive tool for assessing and predicting life expectancy.

Problem Statement:

To create the machine learning based solution to predict the life expectancy of the person.

Problem Solution:

Develop a web application to predict life expectancy based on various factors, providing individuals and healthcare professionals with valuable insights into potential life expectancy. This application can assist individuals in making informed decisions about their lifestyle choices, healthcare planning, and overall well-being. By leveraging machine learning techniques, the application aims to provide accurate and personalized predictions of life expectancy, empowering users to take proactive steps towards a healthier and longer life.

Further Improvements:

The project can be enhanced by incorporating additional features such as demographic data, socioeconomic factors, and lifestyle indicators to improve the accuracy of life expectancy predictions. Additionally, integrating real-time health monitoring devices and wearable technologies can provide personalized health insights and recommendations for individuals to make proactive lifestyle choices. By continuously updating the model with new data and incorporating advanced machine learning algorithms, the accuracy and reliability of the life expectancy predictions can be further improved.

Constraints:

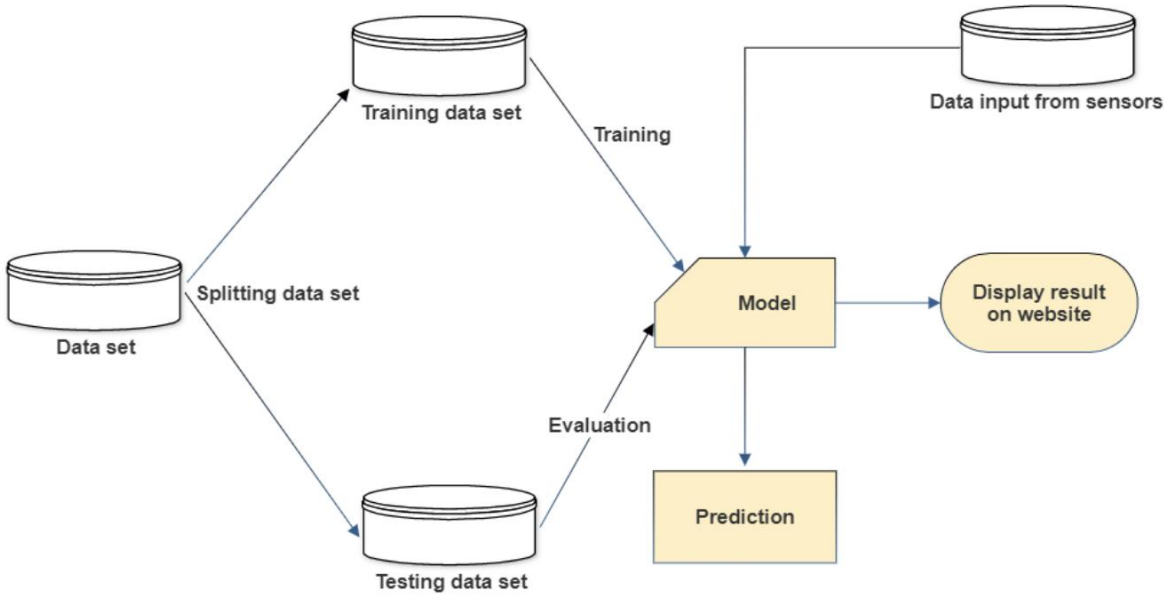
The life expectancy prediction project should ensure user-friendliness and ease of navigation on the website. It should be scalable to accommodate different regions or countries by developing specific models tailored to their unique characteristics. Consideration should be given to the availability and quality of data for different regions to ensure accurate and reliable predictions

Assumptions:

The life expectancy prediction project assumes that the training data used for the machine learning model is representative and accurately captures the factors influencing life expectancy. It assumes that the selected features and algorithms are appropriate for predicting life expectancy. The project also assumes that the input data provided by the users is valid and reliable. Furthermore, it assumes that the project components, such as data ingestion, model training, and prediction, are properly integrated and function as intended.

Design Details

Process Workflow:



Data Collection:

The dataset containing health and socio-economic indicators for life expectancy prediction was provided by the company itself. The dataset included variables such as demographic information, healthcare access, economic factors, and lifestyle indicators.

Data Pre-processing:

The collected dataset underwent thorough pre-processing to handle missing values, outliers, and inconsistencies. Data cleaning techniques were applied to ensure data integrity. Imputation methods were used to fill in missing values, and normalization techniques were employed to standardize the data.

Exploratory Data Analysis (EDA):

EDA was conducted to gain deeper insights into the relationships between various variables and their impact on life expectancy. Visualizations, such as scatter plots, histograms, and correlation matrices, were used to identify patterns, trends, and potential correlation

Feature Selection:

In order to focus on the most relevant predictors, feature selection techniques were applied. Statistical methods, domain knowledge, and feature importance rankings were used to select the most influential variables affecting life expectancy.

Model Training:

In the model training phase, a basic linear regression algorithm was employed to build the predictive model. The dataset was split into separate training, testing, and validation sets to ensure accurate evaluation and performance assessment of the model. This approach allowed for iterative refinement and fine-tuning of the model parameters, optimizing its ability to predict life expectancy based on the selected features.

Model Evaluation:

The trained models were evaluated using various performance metrics such as mean squared error (MSE), root mean squared error (RMSE), and R-squared value. This assessment provided insights into the accuracy and predictive capabilities of the models.

Model Selection:

Based on the evaluation results, the most accurate and reliable model was selected as the final predictor of life expectancy. The chosen model demonstrated the best performance in terms of accuracy and generalization.

Deployment and Testing:

For the deployment of the life expectancy predictor, Flask, a lightweight web framework, was utilized to create the application. The Flask framework provided an efficient and scalable solution for serving the predictive model. To make the predictor accessible to users, the application was deployed in render platform, ensuring high availability and reliability. This deployment setup enabled users to interact with the predictor seamlessly, obtaining predictions on life expectancy based on the provided input data.

Documentation and Reporting:

A comprehensive report was created, documenting the entire project's details. This included information about the data sources, pre-processing techniques, exploratory data analysis findings, feature selection rationale, model training and evaluation results, and insights gained from the project .

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS

Hardware Requirement Specification:

- Processor: Intel Core i5 processor or equivalent
- Memory: 8 GB RAM (Recommended)
- Hard Disk: 500 GB

Software Requirement Specification:

- Operating System: Windows 10 or macOS Mojave (or later) or Linux (Ubuntu 18.04 or later)
- Python 3.7 or higher
- Jupyter Notebook or any preferred code editor
- Python libraries: Pandas, NumPy, Scikit-learn, Matplotlib, Flask

Additional Tools and Frameworks:

- Flask: For creating the web application and API endpoints
- HTML, CSS, JavaScript: For frontend development and user interface design
- Git and GitHub: Version control and collaboration platform for code management
- Render or any preferred cloud platform: For deploying the web application

CHAPTER 4

SOURCE CODE

4.1 Templates:

4.1.1 Landing Page:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Life Expectancy Prediction</title>
    <link
      href="https://fonts.googleapis.com/css?family=Pacifico"
      rel="stylesheet"
      type="text/css"
    />
    <link
      href="https://fonts.googleapis.com/css?family=Arimo"
      rel="stylesheet"
      type="text/css"
    />
    <link
      href="https://fonts.googleapis.com/css?family=Hind:300"
      rel="stylesheet"
      type="text/css"
    />
    <link
      href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300"
      rel="stylesheet"
      type="text/css"
    />

    <style>
      /* Styling the login container */
      .login {
        width: 300px;
        margin: 0 auto;
        padding: 20px;
        border: 1px solid #ccc;
        background-color: #f5f5f5;
      }

      /* Styling the form inputs */
      .login input[type="number"],

      .login button {
        width: 100%;
```

```
    padding: 10px;

    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
  }

  /* Styling the form labels */
  .login p {
    margin: 0;
    font-size: 14px;
    color: #333;
  }

  /* Styling the form submit button */
  .login button {
    background-color: #4caf50;
    color: white;
    font-size: 16px;
    border: none;
    cursor: pointer;
  }

  /* Styling the form submit button on hover */
  .login button:hover {
    background-color: #45a049;
  }

  /* Styling the form title */
  .login h1 {
    font-family: "Pacifico", cursive;
    font-size: 24px;
    text-align: center;
    margin-bottom: 20px;
  }

  /* Styling the small text */
  .login small {
    font-family: "Arimo", sans-serif;
    font-size: 12px;
    color: #666;
  }

  /* Styling the error messages */
  .error {

    color: red;
    font-size: 12px;
```

```
      margin-top: 5px;
    }

    .error-section {
      background-color: #f8d7da;
      color: red;
      padding: 10px;
      margin-top: 20px;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <div class="login">
    <h1><strong>Life Expectancy Prediction</strong></h1>
    <!-- Main Input For Receiving Query to our ML -->
    <form action="/predict" method="POST">
      <p style="text-align: left">
        <small><b>Year [ Between 1980 to 2005 ]</b></small>
      </p>
      <input
        type="number"
        placeholder="Enter Year"
        name="year"
        required="required"
      />
      <p style="text-align: left">
        <small><b>Status [ Either 0 or 1 ]</b></small>
      </p>
      <input
        type="number"
        placeholder="Enter Status"
        name="status"
        required="required"
      />
      <p style="text-align: left">
        <small><b>Adult Mortality [ Between 0 to 2000 ]</b></small>
      </p>
      <input
        type="number"
        placeholder="Enter Adult Mortality"
        name="adult_mortality"
        required="required"
      />
      <p style="text-align: left">
        <small><b>Alcohol [ Between 0 to 20 ]</b></small>

      </p>
      <input
```

```
          type="number"
          step="any"
          placeholder="Enter Alcohol"

          name="alcohol"
          required="required"
        />
        <p style="text-align: left">
          <small><b>Hepatitis B [ Between 0 to 100 ]</b></small>
        </p>
        <input
          type="number"
          placeholder="Enter Hepatitis B"
          name="hepatitis_b"
          required="required"
        />
        <p style="text-align: left">
          <small><b>Measles [ Between 0 to 100000 ]</b></small>
        </p>
        <input
          type="number"
          placeholder="Enter Measles"
          name="measles"
          required="required"
        />
        <p style="text-align: left">
          <small><b>BMI [ Between 0 to 100 ]</b></small>
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter BMI"
          name="bmi"
          required="required"
        />
        <p style="text-align: left">
          <small><b>Under-Five Deaths [ Between 0 to 1000 ]</b></small>
        </p>
        <input
          type="number"
          placeholder="Enter Under-Five Deaths"
          name="under_five_deaths"
          required="required"
        />
        <p style="text-align: left">
          <small><b>Polio [ Between 0 to 100 ]</b></small>

        </p>
        <input
```

```
          type="number"
          placeholder="Enter Polio"
          name="polio"
          required="required"

        />
        <p style="text-align: left">
          <small><b>Total Expenditure [ Between 0 to 20 ]</b></small>
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter Total Expenditure"
          name="total_expenditure"
          required="required"
        />
        <p style="text-align: left">
          <small><b>Diphtheria [ Between 0 to 100 ]</b></small>
        </p>
        <input
          type="number"
          placeholder="Enter Diphtheria"
          name="diphtheria"
          required="required"
        />
        <p style="text-align: left">
          <small><b>HIV/AIDS [ Between 0 to 50 ]</b></small>
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter HIV/AIDS"
          name="hiv_aids"
          required="required"
        />
        <p style="text-align: left">
          <small><b>GDP [ Between 0 to 50000 ]</b></small>
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter GDP"
          name="gdp"
          required="required"
        />
        <p style="text-align: left">

          <small><b>Population [ Between 0 to 1,00,00,00,000 ]</b></small>
        </p>
```

```
        <input
          type="number"
          placeholder="Enter Population"
          name="population"
          required="required"

        />
        <p style="text-align: left">
          <small><b>Thinness 1-19 Years [ Between 0 to 50 ]</b></small>
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter Thinness 1-19 Years"
          name="thinness_1_19_years"
          required="required"
        />
        <p style="text-align: left">
          <small
            ><b>Income Composition of Resources [ Between 0 to 1 ]</b></small
          >
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter Income Composition of Resources"
          name="income_composition"
          required="required"
        />
        <p style="text-align: left">
          <small><b>Schooling [ Between 0 to 20 ]</b></small>
        </p>
        <input
          type="number"
          step="any"
          placeholder="Enter Schooling"
          name="schooling"
          required="required"
        />

        {% if errors %}
        <div class="error-section">
          <h2>Error:</h2>
          <ul>
            {% for error in errors %}
            <li>{{ error }}</li>

            {% endfor %}
          </ul>
```

```
        </div>
        {% endif %}

        <button type="submit" class="btn btn-primary btn-block btn-large">
          Predict
        </button>

      </form>
    </div>
  </body>
</html>
```

4.1.2 Result Page:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8" />
    <link
      href="https://fonts.googleapis.com/css?family=Amaranth"
      rel="stylesheet"
    />
    <title>Result</title>
    <style>
      body {
        margin: 0;
        padding: 0;
      }

      .con {
        background-repeat: no-repeat;
        background-size: cover;
        background-position: center;
        height: 100vh;
        width: 100vw;
        display: flex;
        align-items: center;
        justify-content: center;
      }

      .results {
        font-size: 50px;
        color: black;
        font-family: "Amaranth";
      }


      .prediction {
        color: black;
```

```
    }

    .poor {
      color: green;
    }

    .satisfactory {

      color: yellow;
    }

    .moderate {
      color: orange;
    }

    .great {
      color: red;
    }

    .amazing {
      color: darkred;
    }
  </style>
</head>

<body>
  <div class="con">
    <div class="results">
      {% set prediction = prediction|int %}
      <h4>
        Prediction: The Estimated Life Expectancy is
        <span class="prediction">{{ prediction|float }} </span>
      </h4>

      {% if prediction >= 0 and prediction <= 55 %}
      <h4>
        Prediction: The Estimated Life Expectancy is
        <span class="poor">Poor</span>
      </h4>
      {% elif prediction >= 56 and prediction <= 70 %}
      <h4>
        Prediction: The Estimated Life Expectancy is
        <span class="moderate">Moderate</span>
      </h4>
      {% elif prediction >= 71 and prediction <= 80 %}

      <h4>
        Prediction: The Estimated Life Expectancy is
        <span class="satisfactory">Satisfactory</span>
```

```
        </h4>
        {% elif prediction >= 81 and prediction <= 90 %}
        <h4>
          Prediction: The Estimated Life Expectancy is
          <span class="great">Great</span>
        </h4>
        {% else %}
        <h4>

          Prediction: The Estimated Life Expectancy is
          <span class="amazing">Amazing</span>
        </h4>
        {% endif %}
      </div>
    </div>
  </body>
</html>
```

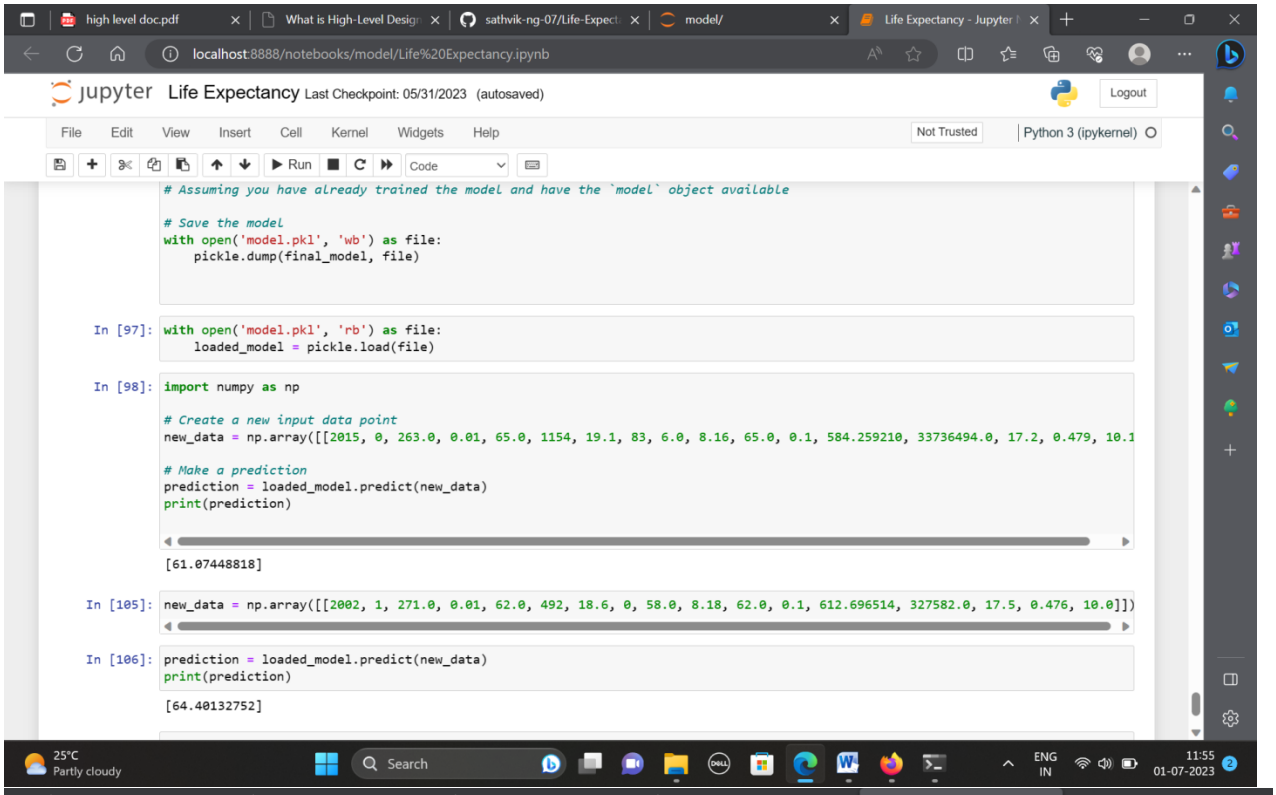
4.2 Flask code (app.py):

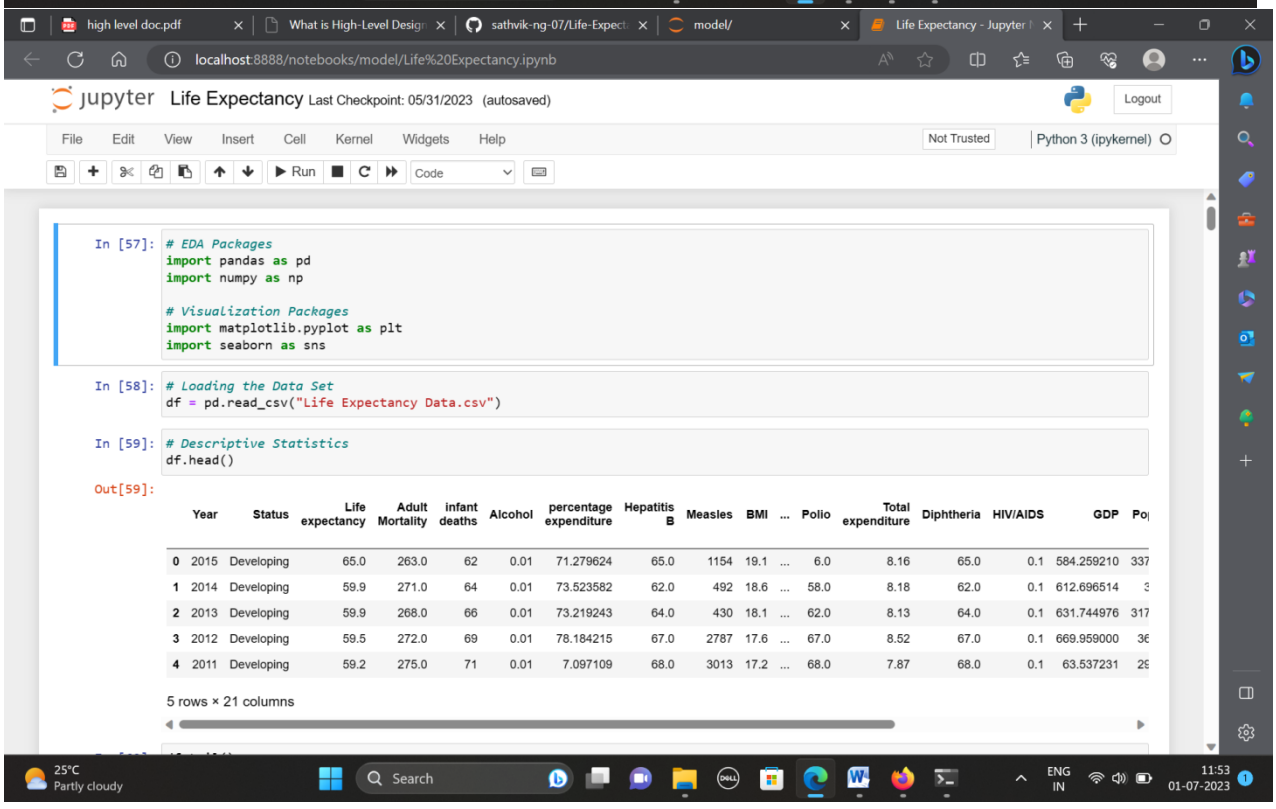
```
from flask import Flask, render_template, request, redirect, url_for
import pickle
import numpy as np

app = Flask(__name__)

def load_model():
    with open('model.pkl', 'rb') as file:
        model = pickle.load(file)
    return model

model = load_model()

def validate_input(data):
    ranges = {
        'Year': (1980, 2025),
        'Status': (0, 1),
        'Alcohol': (0, 20),
        'Adult Mortality': (0, 2000),
        'Hepatitis B': (0, 100),
        'Measles': (0, 100000),
        'BMI': (0, 100),
        'under-five deaths': (0, 1000),
        'Polio': (0, 100),
        'Total expenditure': (0, 20),

        'Diphtheria': (0, 100),
        'HIV/AIDS': (0, 50),
```

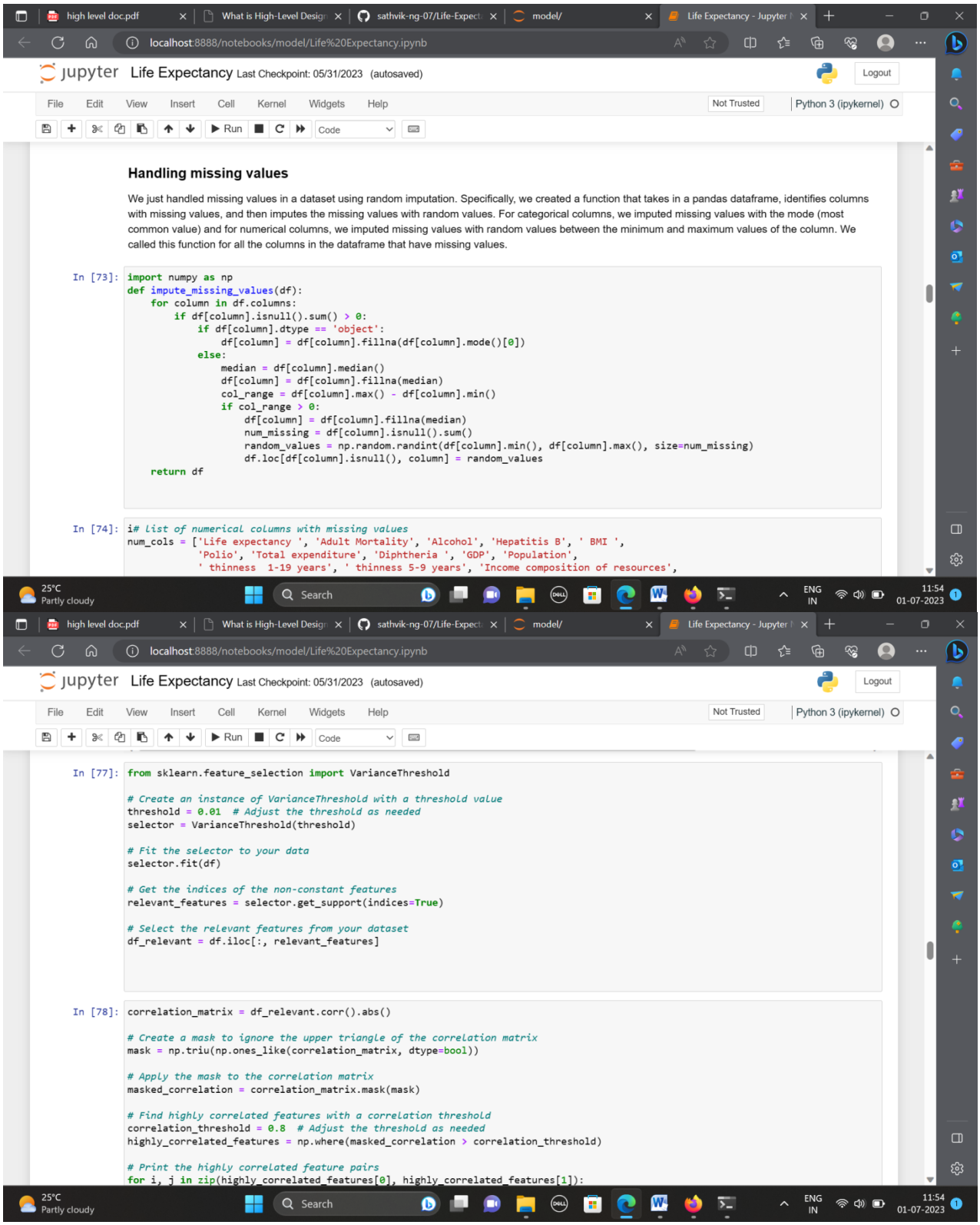
```
        'GDP': (0, 50000),
        'Population': (0, 1000000000),
        'thinness 1-19 years': (0, 50),
        'Income composition of resources': (0, 1),
        'Schooling': (0, 20)
    }
    errors = []
    for feature, value in data.items():
        if feature in ranges:

            min_val, max_val = ranges[feature]
            if value < min_val or value > max_val:
                errors.append(f"{feature}: Value must be between {min_val} and
{max_val}")
    return errors

@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        return redirect(url_for('predict'))
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        year = int(request.form['year'])
        status = int(request.form['status'])
        alcohol = float(request.form['alcohol'])
        adult_mortality = float(request.form['adult_mortality'])
        hepatitis_b = float(request.form['hepatitis_b'])
        measles = int(request.form['measles'])
        bmi = float(request.form['bmi'])
        under_five_deaths = int(request.form['under_five_deaths'])
        polio = float(request.form['polio'])
        total_expenditure = float(request.form['total_expenditure'])
        diphtheria = float(request.form['diphtheria'])
        hiv_aids = float(request.form['hiv_aids'])
        gdp = float(request.form['gdp'])
        population = float(request.form['population'])
        thinness_1_19_years = float(request.form['thinness_1_19_years'])
        income_composition = float(request.form['income_composition'])
        schooling = float(request.form['schooling'])

        # Prepare the input data for prediction
        input_data = {
            'Year': year,

            'Status': status,
            'Alcohol': alcohol,
```

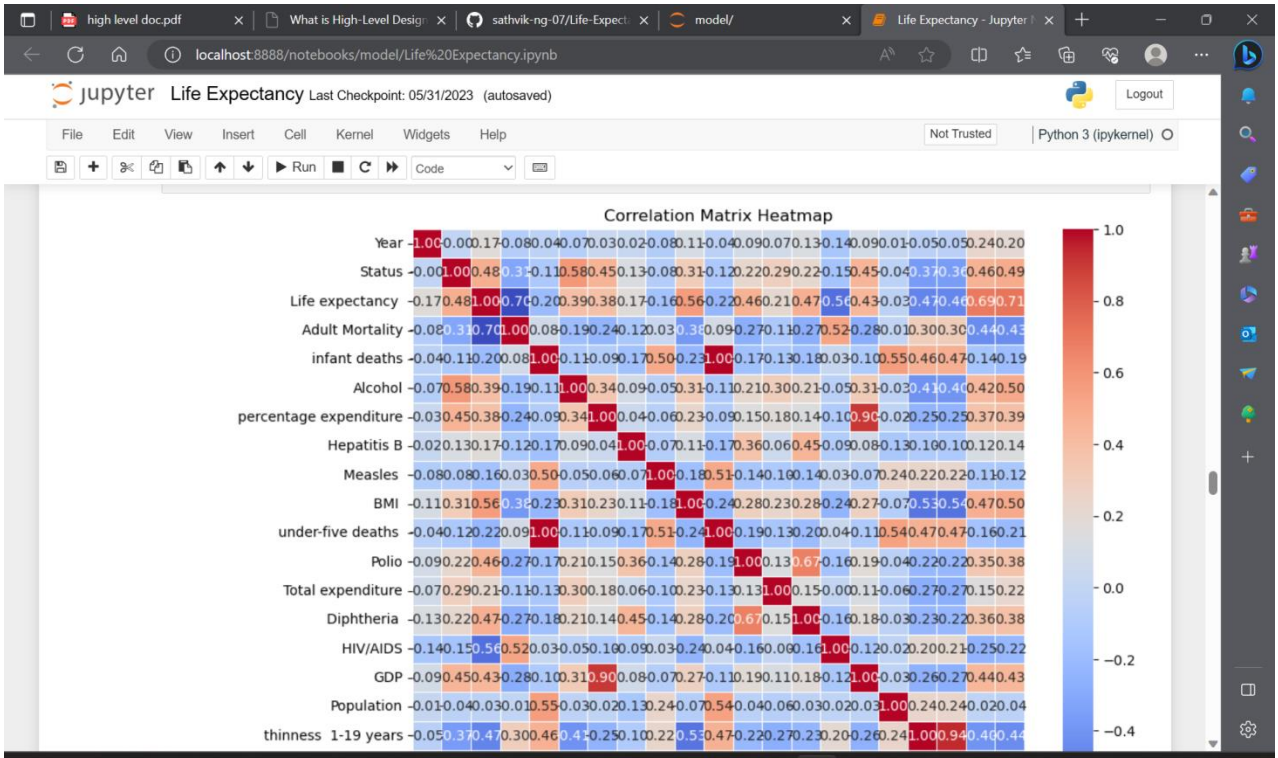
```
            'Adult Mortality': adult_mortality,
            'Hepatitis B': hepatitis_b,
            'Measles': measles,
            'BMI': bmi,
            'under-five deaths': under_five_deaths,
            'Polio': polio,
            'Total expenditure': total_expenditure,
            'Diphtheria': diphtheria,
            'HIV/AIDS': hiv_aids,
            'GDP': gdp,

            'Population': population,
            'thinness 1-19 years': thinness_1_19_years,
            'Income composition of resources': income_composition,
            'Schooling': schooling
        }

        # Validate the input data
        errors = validate_input(input_data)

        if errors:
            return render_template('index.html', errors=errors)

        # Convert the input data to a numpy array
        input_data = np.array(list(input_data.values())).reshape(1, -1)

        # Use the loaded model to make the prediction
        prediction = model.predict(input_data)[0]

        # Redirect to the result page with the prediction
        return redirect(url_for('result', prediction=float(prediction)))

    # If it's a GET request, render the index page
    return render_template('index.html')

@app.route('/result')
def result():
    prediction = request.args.get('prediction')
    return render_template('result.html', prediction=prediction)

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

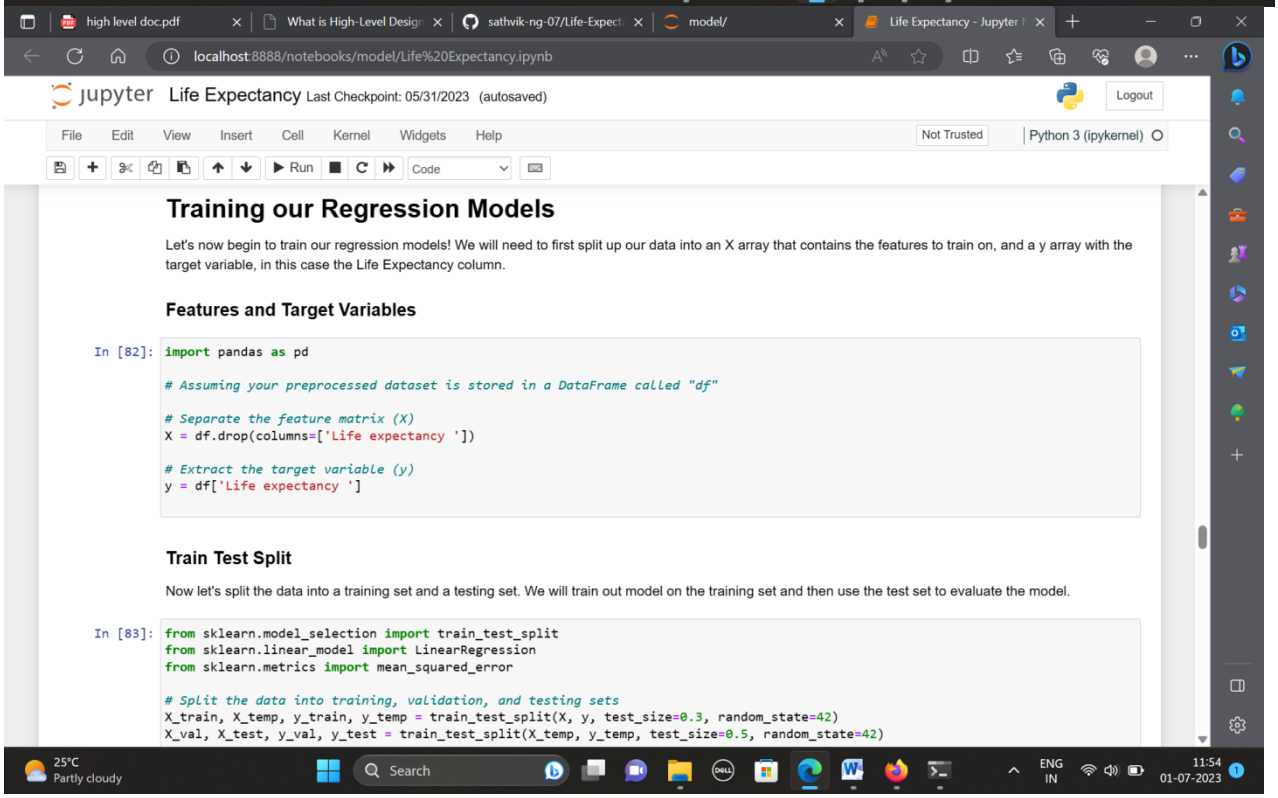
4.3 Jupyter Notebook Code:





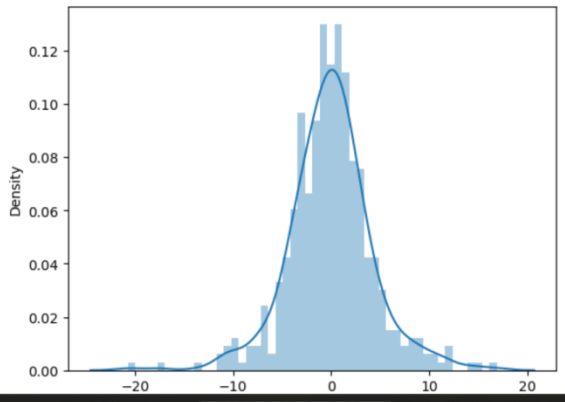

high level doc.pdf
What is High-Level Design
sathvik-ng-07/Life-Expectancy
model/
Life Expectancy - Jupyter
localhost:8888/notebooks/model/Life%20Expectancy.ipynb
jupyter Life Expectancy Last Checkpoint: 05/31/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Handling missing values
We just handled missing values in a dataset using random imputation. Specifically, we created a function that takes in a pandas dataframe, identifies columns with missing values, and then imputes the missing values with random values. For categorical columns, we imputed missing values with the mode (most common value) and for numerical columns, we imputed missing values with random values between the minimum and maximum values of the column. We called this function for all the columns in the dataframe that have missing values.
In [73]:
import numpy as np
def impute_missing_values(df):
for column in df.columns:
if df[column].isnull().sum() > 0:
if df[column].dtype == 'object':
df[column] = df[column].fillna(df[column].mode()[0])
else:
median = df[column].median()
df[column] = df[column].fillna(median)
col_range = df[column].max() - df[column].min()
if col_range > 0:
df[column] = df[column].fillna(median)
num_missing = df[column].isnull().sum()
random_values = np.random.randint(df[column].min(), df[column].max(), size=num_missing)
df.loc[df[column].isnull(), column] = random_values
return df
In [74]:
i# List of numerical columns with missing values
num_cols = ['Life expectancy ', 'Adult Mortality', 'Alcohol', 'Hepatitis B', ' BMI ',
'Polio', 'Total expenditure', 'Diphtheria ', 'GDP', 'Population',
' thinness 1-19 years', ' thinness 5-9 years', 'Income composition of resources',
25°C Partly cloudy
Search
ENG IN
11:54 01-07-2023
In [77]:
from sklearn.feature_selection import VarianceThreshold
# Create an instance of VarianceThreshold with a threshold value
threshold = 0.01 # Adjust the threshold as needed
selector = VarianceThreshold(threshold)
# Fit the selector to your data
selector.fit(df)
# Get the indices of the non-constant features
relevant_features = selector.get_support(indices=True)
# Select the relevant features from your dataset
df_relevant = df.iloc[:, relevant_features]
In [78]:
correlation_matrix = df_relevant.corr().abs()
# Create a mask to ignore the upper triangle of the correlation matrix
mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))
# Apply the mask to the correlation matrix
masked_correlation = correlation_matrix.mask(mask)
# Find highly correlated features with a correlation threshold
correlation_threshold = 0.8 # Adjust the threshold as needed
highly_correlated_features = np.where(masked_correlation > correlation_threshold)
# Print the highly correlated feature pairs
for i, j in zip(highly_correlated_features[0], highly_correlated_features[1]):


high level doc.pdf
What is High-Level Design
sathvik-ng-07/Life-Expect
model/
Life Expectancy - Jupyter
localhost:8888/notebooks/model/Life%20Expectancy.ipynb
jupyter Life Expectancy Last Checkpoint: 05/31/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Correlation Matrix Heatmap
Year
Status
Life expectancy
Adult Mortality
infant deaths
Alcohol
percentage expenditure
Hepatitis B
Measles
BMI
under-five deaths
Polio
Total expenditure
Diphtheria
HIV/AIDS
GDP
Population
thinness 1-19 years
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
25°C
Partly cloudy
Search
ENG
IN
11:54
01-07-2023

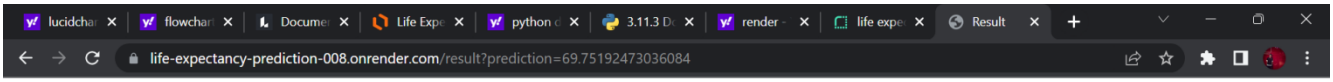

high level doc.pdf
What is High-Level Design
sathvik-ng-07/Life-Expect
model/
Life Expectancy - Jupyter
localhost:8888/notebooks/model/Life%20Expectancy.ipynb
jupyter Life Expectancy Last Checkpoint: 05/31/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Training our Regression Models
Let's now begin to train our regression models! We will need to first split up our data into an X array that contains the features to train on, and a y array with the target variable, in this case the Life Expectancy column.
Features and Target Variables
In [82]:
import pandas as pd
# Assuming your preprocessed dataset is stored in a DataFrame called "df"
# Separate the feature matrix (X)
X = df.drop(columns=['Life expectancy '])
# Extract the target variable (y)
y = df['Life expectancy ']
Train Test Split
Now let's split the data into a training set and a testing set. We will train out model on the training set and then use the test set to evaluate the model.
In [83]:
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
# Split the data into training, validation, and testing sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
25°C
Partly cloudy
Search
ENG
IN
11:54
01-07-2023

```
# Train the model on the training set
model.fit(X_train, y_train)

# Predict the target variable for the validation set
y_val_pred = model.predict(X_val)

# Evaluate the model performance on the validation set
mse_val = mean_squared_error(y_val, y_val_pred)
rmse_val = np.sqrt(mse_val)
r_squared_val = model.score(X_val, y_val)

# Print the evaluation metrics on the validation set
print("Validation Set:")
print("Mean Squared Error:", mse_val)
print("Root Mean Squared Error:", rmse_val)
print("R-squared:", r_squared_val)

# Predict the target variable for the testing set
y_test_pred = model.predict(X_test)

# Evaluate the model performance on the testing set
mse_test = mean_squared_error(y_test, y_test_pred)
rmse_test = np.sqrt(mse_test)
r_squared_test = model.score(X_test, y_test)

# Print the evaluation metrics on the testing set
print("Testing Set:")
print("Mean Squared Error:", mse_test)
print("Root Mean Squared Error:", rmse_test)
print("R-squared:", r_squared_test)
```

```
In [88]: sns.distplot((y_test-lmpredictions),bins=50);

C:\Users\sathv\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

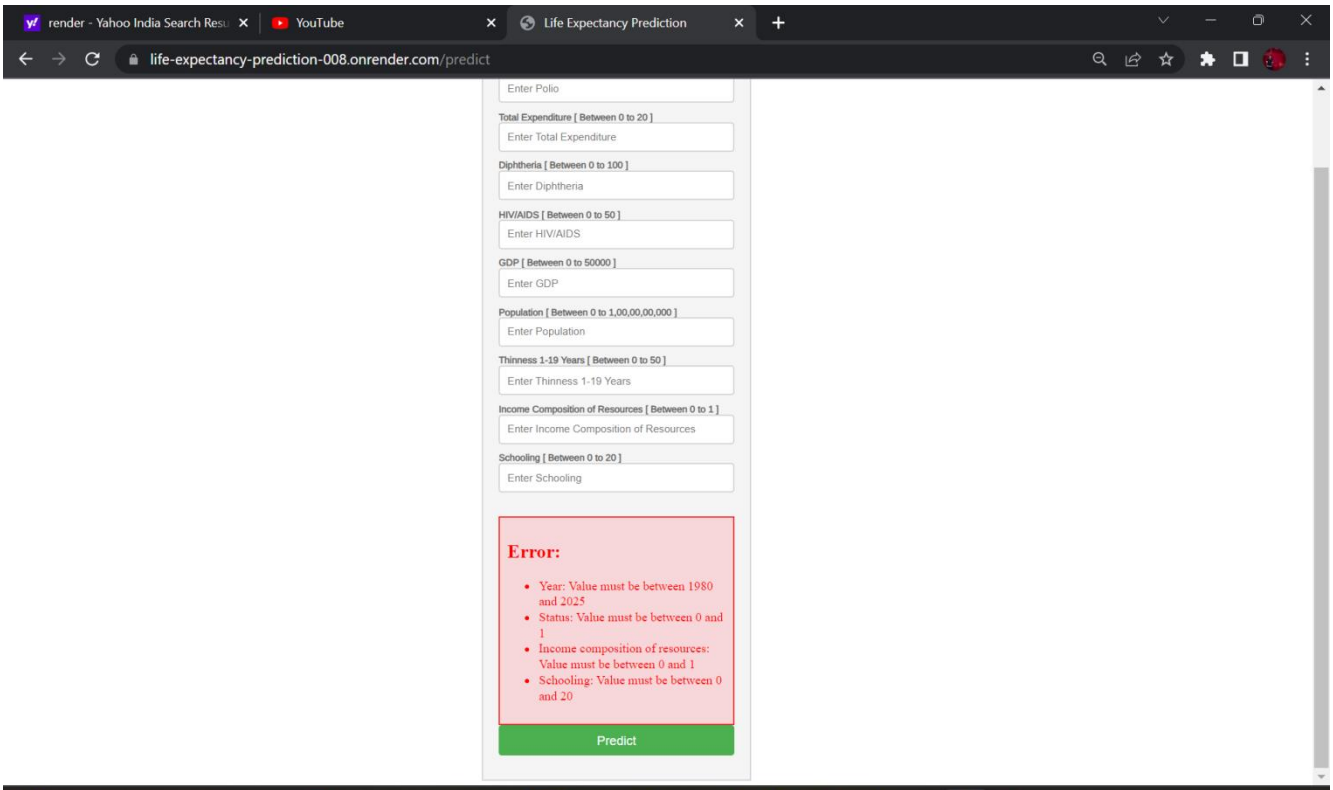


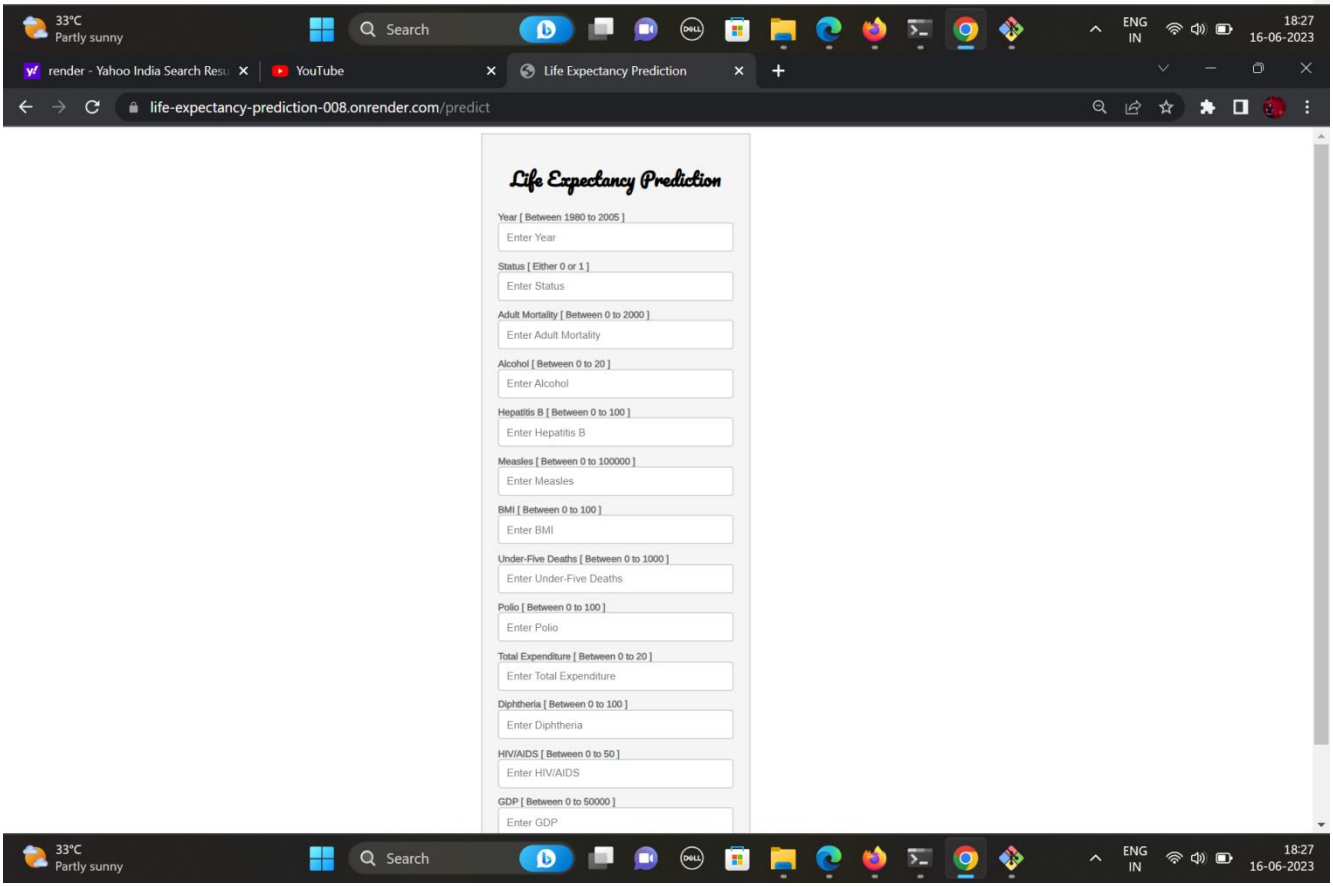
SCREENSHOTS



Prediction: The Estimated Life Expectancy is 69.0

Prediction: The Estimated Life Expectancy is Moderate


render - Yahoo India Search Resu
YouTube
Life Expectancy Prediction
life-expectancy-prediction-008.onrender.com/predict
Enter Polio
Total Expenditure [Between 0 to 20]
Enter Total Expenditure
Diphtheria [Between 0 to 100]
Enter Diphtheria
HIV/AIDS [Between 0 to 50]
Enter HIV/AIDS
GDP [Between 0 to 50000]
Enter GDP
Population [Between 0 to 1,00,00,00,000]
Enter Population
Thinness 1-19 Years [Between 0 to 50]
Enter Thinness 1-19 Years
Income Composition of Resources [Between 0 to 1]
Enter Income Composition of Resources
Schooling [Between 0 to 20]
Enter Schooling
Error:
Year: Value must be between 1980 and 2025
Status: Value must be between 0 and 1
Income composition of resources: Value must be between 0 and 1
Schooling: Value must be between 0 and 20
Predict


33°C
Partly sunny
Search
ENG
IN
18:27
16-06-2023
render - Yahoo India Search Resu
YouTube
Life Expectancy Prediction
life-expectancy-prediction-008.onrender.com/predict
Life Expectancy Prediction
Year [Between 1980 to 2005]
Enter Year
Status [Either 0 or 1]
Enter Status
Adult Mortality [Between 0 to 2000]
Enter Adult Mortality
Alcohol [Between 0 to 20]
Enter Alcohol
Hepatitis B [Between 0 to 100]
Enter Hepatitis B
Measles [Between 0 to 100000]
Enter Measles
BMI [Between 0 to 100]
Enter BMI
Under-Five Deaths [Between 0 to 1000]
Enter Under-Five Deaths
Polio [Between 0 to 100]
Enter Polio
Total Expenditure [Between 0 to 20]
Enter Total Expenditure
Diphtheria [Between 0 to 100]
Enter Diphtheria
HIV/AIDS [Between 0 to 50]
Enter HIV/AIDS
GDP [Between 0 to 50000]
Enter GDP
33°C
Partly sunny
Search
ENG
IN
18:27
16-06-2023

CHAPTER 5

CONCLUSION

The life expectancy prediction project offers a valuable machine learning model for estimating life expectancy based on various factors. By leveraging this model, individuals and healthcare professionals can gain insights into life expectancy trends and make informed decisions. The project's focus on reducing disease and addressing health issues associated with air pollution contributes to overall well-being. Additionally, the project provides a basis for government initiatives to combat air pollution and improve public health in affected areas.

REFERENCES

1. HTML Tutorial (w3schools.com)
2. CSS Tutorial (w3schools.com)
3. 3.11.4 Documentation (python.org)
4. NumPy Documentation
5. pandas documentation — pandas 2.0.3 documentation (pydata.org)
6. Matplotlib documentation — Matplotlib 3.7.1 documentation
7. Project Jupyter Documentation — Jupyter Documentation 4.1.1 alpha documentation
8. Welcome to Flask — Flask Documentation (2.3.x) (palletsprojects.com)
9. scikit-learn: machine learning in Python — scikit-learn 1.3.0 documentation
10. Project demo - https://youtu.be/hgw2KTCChL0
11. Github Repository - https://github.com/sathvik-ng-07/Life-Expectancy-Prediction.git

CERTIFICATE

VIEH Private Limited

Sagar City Society, Andheri West, Mumbai, Maharashtra-400058

CERTIFICATE OF INTERNSHIP

Date: June 17, 2023

TO WHOM IT MAY CONCERN

This is to certify that Mr **Sathvik N G** has successfully completed the 30 days (One Month) internship program with VIEH PRIVATE LIMITED. He has worked on a project titled **Life Expectancy Prediction** where he has showcased his excellent skills and knowledge.

During his internship program with us, he demonstrated exceptional skills with a self-motivated attitude to learn new things and implemented them end to end with all of our mentioned industrial standards. His performance was excellent and was able to complete the project successfully on time.

We wish him all the best for his future endeavors.

Regards,

Nooruddin Shaikh

CTO & Chief Data Scientist at VIEH

www.viehgroup.com | support@viehgroup.com