**Health Data Analysis Report**

Subject: Python Programming(24CAH-606)

Submitted by: Anuj (24MCI10020)

Section\group:24MAM1(A) Branch:MCA(AIML)

Github:- https://github.com/anujdhiman28/Health-Data-Analysis-.git

**1. Aim**

The aim of this project is to create an interactive application that helps users analyze health-related data stored in CSV files. The application allows users to upload their data, view statistical summaries, and visualize key health metrics.

**2. Objective/Problem Definition**

The main objectives of this project are:

* To provide a simple interface for users to upload health data in CSV format.
* To generate statistical summaries (like mean, median, etc.) of the uploaded data.
* To create visual representations of health data to identify trends and patterns.
* To streamline the process of data analysis for healthcare professionals and researchers.

**3. Programming Languages Used**

The application is developed using:

* **Python**: The main programming language for developing the application.
* **Tkinter**: A GUI toolkit used to create the user interface.
* **Matplotlib**: A library for generating visualizations and plots.
* **Pandas**: A library used for data manipulation and analysis.

**4. Algorithm or Pseudo Code**

Here is a simple algorithm for the application:

1. **Start Application**
2. **Create GUI Interface**
3. **Define Function to Upload CSV**
   * Prompt the user to select a CSV file.
   * Read the CSV data into a data structure.
4. **Define Function to Show Statistics**
   * Calculate and display statistics (mean, median, etc.) of the data.
5. **Define Function to Visualize Data**
   * Ask the user for the column to visualize.
   * Generate and display a histogram or other visualizations.
6. **End Application**

**5. Implementation**

The implementation involves creating a user-friendly interface where users can upload their health data and perform analyses.

import tkinter as tk

from tkinter import filedialog, messagebox, Frame, Label

import tkinter.simpledialog as simpledialog

import pandas as pd

import matplotlib.pyplot as plt

class HealthDataAnalyzer:

def \_\_init\_\_(self, master):

self.master = master

self.master.title("Health Data Analyzer")

self.master.geometry("400x300")

self.master.config(bg="#f0f0f0") # Light gray background

# Frame for buttons

self.frame = Frame(master, bg="#f0f0f0")

self.frame.pack(pady=20)

# Title Label

self.title\_label = Label(master, text="Health Data Analyzer", font=("Helvetica", 16), bg="#f0f0f0")

self.title\_label.pack(pady=10)

# Upload button

self.upload\_button = tk.Button(self.frame, text="Upload CSV", command=self.upload\_data, width=20, bg="#4CAF50", fg="white", font=("Helvetica", 12))

self.upload\_button.pack(pady=10)

# Statistics button

self.stats\_button = tk.Button(self.frame, text="Show Statistics", command=self.show\_statistics, width=20, bg="#2196F3", fg="white", font=("Helvetica", 12))

self.stats\_button.pack(pady=10)

# Visualize button

self.visualize\_button = tk.Button(self.frame, text="Visualize Data", command=self.visualize\_data, width=20, bg="#FF5722", fg="white", font=("Helvetica", 12))

self.visualize\_button.pack(pady=10)

self.data = None

def upload\_data(self):

file\_path = filedialog.askopenfilename(filetypes=[("CSV Files", "\*.csv")])

if file\_path:

try:

self.data = pd.read\_csv(file\_path)

messagebox.showinfo("Success", "Data uploaded successfully!")

except Exception as e:

messagebox.showerror("Error", f"Failed to upload data: {e}")

def show\_statistics(self):

if self.data is not None:

stats = self.data.describe()

messagebox.showinfo("Statistics", stats.to\_string())

else:

messagebox.showwarning("Warning", "No data uploaded!")

def visualize\_data(self):

if self.data is not None:

# List available columns

available\_columns = self.data.columns.tolist()

column\_name = simpledialog.askstring("Input", f"Available columns: {available\_columns}\nEnter the column name to visualize:")

if column\_name in available\_columns:

plt.figure(figsize=(8, 6))

plt.hist(self.data[column\_name], bins=10, edgecolor='black')

plt.title(f'Distribution of {column\_name}', fontsize=14)

plt.xlabel(column\_name, fontsize=12)

plt.ylabel('Frequency', fontsize=12)

plt.grid(axis='y', alpha=0.75)

plt.show()

else:

messagebox.showwarning("Warning", f"'{column\_name}' column not found in data.")

else:

messagebox.showwarning("Warning", "No data uploaded!")

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = HealthDataAnalyzer(root)

root.mainloop()

**6. Output**

When the application runs, users can:

* Upload a CSV file containing health data and receive a success message.
* View a statistical summary of the data, such as count, mean, and standard deviation.
* Generate visualizations, such as histograms, to analyze data distributions.

**Example Output:**

* **Statistical Summary**:

Count Mean Std Dev

Age 100 45.6 10.5A screenshot of a computer

Description automatically generated

* **Histogram**: A visual representation of the age distribution in the dataset.
* A graph with blue and white bars

  Description automatically generated

**7. Conclusion**

The Health Data Analyzer application successfully fulfills its objectives by providing an intuitive interface for users to upload health data and analyze it. The application facilitates informed decision-making in healthcare by enabling users to visualize and understand their data better.

**8. Future Framework**

Future enhancements could include:

* Support for additional file formats (like Excel).
* Advanced visualizations, such as scatter plots and line graphs.
* Machine learning capabilities for predictive analytics.
* Enhanced user authentication features for data security.

**9. Learning Outcomes**

The development of this application led to several valuable learning experiences:

* Improved understanding of Python programming and data analysis techniques.
* Gained experience in developing user interfaces using Tkinter.
* Learned how to create effective visualizations using Matplotlib.
* Developed insights into the practical applications of data analytics in the healthcare sector.