FPT Education

FPT UNIVERSITY

Heart Attack Prediction

Lecturer: HuongNTD13
GROUP 2 - SE1840

Full Name	Student ID	Email
Nguyễn Tuấn Kiệt	SE183979	KietNTSE183979@fpt.edu.vn
Lê Thanh Hùng	SE181763	hungltse181763@fpt.edu.vn
Thái Minh Tuấn	SE181850	tuantmse181850@fpt.edu.vn
Nguyễn Ngọc Long	SE193490	nguyenngoclong216@gmail.com
Trần Duy Đạt	SE190622	megalit2578@gmail.com
Trần Tuấn Kiệt	SE194431	akiettran2005@gmail.com
Phạm Vũ Khánh Như	SE194657	vukhanhnhu@gmail.com

TABLE OF CONTENTS

I. INTRODUCTION

Heart disease remains one of the leading causes of mortality worldwide, making early detection and risk assessment crucial for effective prevention and treatment. In this project, we utilize logistic regression models to predict the likelihood of heart attacks based on various health and lifestyle factors. By analyzing real-world patient data, we aim to identify key variables that significantly influence heart attack risk.

Our study begins with data collection and preprocessing, followed by statistical analyses to determine the most impactful predictors. We employ both **univariate** and **multivariate logistic regression** models to evaluate the relationship between different risk factors and heart disease. Through this approach, we identify variables such as **heart rate, sedentary hours per day, sleep duration, and income** as significant contributors to heart attack risk.

The results of this study provide valuable insights into heart disease prevention by emphasizing the importance of lifestyle factors in cardiovascular health. By leveraging machine learning techniques in medical research, we demonstrate the potential of predictive modeling in supporting early diagnosis and intervention strategies.

II. DATA FILE OVERVIEW

1. Data Source
   Data from surveys and assessments of patients with and without heart disease, on a variety of factors.
2. Variable Descriptions

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Patient ID	Age	Sex	Cholesterol	Blood Press	Heart Rate	Diabetes	Family Hist	Smoking	Obesity	Alcohol Cor	Exercise Ho	Diet
2	NAH2455	47	Female	320	177/105	50	1	0	1	0	0	16.136152	Average
3	YEK5141	45	Male	228	106/86	106	1	0	1	0	0	14.903771	Average
4	HWI6336	30	Male	317	145/62	69	0	0	1	1	1	7.9747979	Healthy
5	JOS4879	39	Male	359	106/107	55	1	0	1	1	1	7.8399646	Unhealthy
6	WFI8983	66	Female	123	136/68	46	1	1	1	1	1	7.1383207	Unhealthy
7	YOK1424	47	Male	177	133/94	56	1	1	1	0	1	18.067469	Unhealthy
8	TVT5972	21	Male	258	130/104	98	1	0	1	0	0	16.991402	Healthy
9	CTH1204	21	Female	327	106/106	102	1	1	0	0	0	13.199833	Average
10	LSV6385	36	Male	359	151/95	91	1	1	1	1	0	19.310412	Healthy
11	USI2077	37	Male	287	159/69	42	1	0	1	1	1	8.0536787	Average
12	FTZ2813	79	Male	205	115/90	98	1	1	1	1	0	8.0427903	Unhealthy
13	SXA0417	84	Female	159	149/107	67	0	0	1	1	1	12.515621	Healthy
14	NQI3156	23	Male	198	166/82	79	1	1	1	0	1	19.97306	Unhealthy
15	PBQ9779	89	Male	387	166/84	110	1	0	1	0	1	7.5059544	Average
16	IQN9681	90	Male	261	102/89	83	0	1	1	1	0	7.7858276	Average
17	WDY5513	42	Female	386	175/60	65	1	0	1	1	1	9.4648724	Average
18	RGE5156	87	Female	348	97/76	44	1	1	1	0	0	0.8312766	Average
19	RTP5890	59	Male	268	122/96	56	0	1	1	1	1	3.8716034	Healthy
20	BYS8921	82	Male	180	138/102	86	1	0	1	0	0	14.579057	Unhealthy
21	EQA8301	39	Male	131	118/101	71	0	0	1	1	0	3.0669503	Average
22	WFX2656	29	Male	183	148/66	95	1	0	1	0	1	3.8403211	Healthy
23	CLC9566	25	Male	243	93/69	65	1	1	1	1	1	8.6618825	Healthy
24	KYA3135	54	Male	208	98/77	92	1	1	1	1	0	16.269577	Healthy
25	NMI6940	77	Male	183	168/66	93	1	1	1	0	0	8.6253689	Average
26	ZLG0548	34	Male	338	160/74	89	1	0	1	0	0	19.086815	Healthy
27	GAP9735	37	Male	218	132/99	62	0	1	1	0	0	1.088379	Healthy
28	QPW6757	40	Male	263	149/61	55	1	0	1	1	1	2.3232472	Average
29	OSJ2917	82	Male	155	118/82	47	1	0	1	1	1	6.945524	Healthy
30	XRC4897	20	Male	343	170/69	91	1	0	1	0	1	1.2306518	Average
31	PMW6564	54	Female	165	92/110	90	1	0	1	1	1	10.103903	Average
32	BWO6796	76	Female	364	146/104	44	1	1	1	1	0	6.9733938	Healthy
33	GZB0696	40	Female	356	97/93	70	1	1	0	1	0	1.4288144	Average
34	PVW4422	89	Female	281	129/101	94	0	1	1	1	0	11.449844	Healthy
35	YTW9037	75	Male	280	94/61	70	1	1	1	0	1	5.7268142	Healthy
36	HCQ7578	81	Male	259	114/86	40	1	1	1	1	1	6.518192	Healthy
37	DRT8219	87	Female	327	163/83	73	1	0	1	0	1	4.6778583	Unhealthy

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	Previous He	Medication	Stress Level	Sedentary H	Income	BMI	Triglyceride	Physical Ac	Sleep Hours	Country	Continent	Hemisphere	Heart Attack Risk	
2	1	0	3	10.066156	43561	19.035467	799	0	10	South Korea	Asia	Northern He	0	
3	1	0	10	5.2134702	235216	35.14416	606	6	4	Australia	Australia	Southern He	1	
4	0	0	2	3.2792262	215362	21.781774	706	5	7	United State	North Amer	Northern He	0	
5	0	1	10	0.4873665	48515	32.152313	759	4	7	United State	North Amer	Northern He	0	
6	0	1	9	1.4005517	139240	39.901478	283	2	7	Germany	Europe	Northern He	0	
7	1	1	5	9.6753624	197158	22.128568	515	2	9	Nigeria	Africa	Northern He	1	
8	0	0	2	6.9999425	51166	36.35251	344	0	10	Vietnam	Asia	Northern He	1	
9	0	1	8	10.902032	30085	18.127401	647	5	5	South Korea	Asia	Northern He	0	
10	0	0	10	1.9121897	206371	20.28646	658	7	6	Japan	Asia	Northern He	1	
11	1	1	2	6.7446673	56304	28.435856	340	3	9	Germany	Europe	Northern He	0	
12	0	1	10	11.548869	30040	22.010586	551	2	5	United State	North Amer	Northern He	0	
13	0	1	8	5.0057213	55847	21.345971	488	2	6	Colombia	South Amer	Northern He	0	
14	1	1	6	2.6816853	295167	23.744046	411	5	5	France	Europe	Northern He	0	
15	0	1	10	7.0104125	134192	25.903785	369	0	6	Australia	Australia	Southern He	0	
16	1	1	9	7.0013016	31209	23.359687	211	1	7	Nigeria	Africa	Northern He	0	
17	0	0	8	2.9988048	62170	26.177842	446	5	5	Vietnam	Asia	Northern He	0	
18	1	0	2	11.824416	255032	21.141469	100	4	10	Thailand	Asia	Northern He	0	
19	0	1	5	7.2984972	126765	30.380847	374	5	5	Japan	Asia	Northern He	0	
20	0	0	4	6.7928847	61149	39.8841	732	2	5	Italy	Europe	Southern He	0	
21	1	0	5	5.8574291	157755	29.952541	652	6	8	United King	Europe	Northern He	0	
22	1	0	1	7.4760485	154165	26.553512	474	4	10	Germany	Europe	Northern He	0	
23	0	1	1	2.0702068	226524	32.859366	642	4	8	Thailand	Asia	Northern He	0	
24	0	0	7	5.9319487	172279	22.131913	414	4	6	India	Asia	Northern He	0	
25	1	1	8	1.4361066	250397	37.770234	88	7	7	Italy	Europe	Southern He	1	
26	1	0	2	6.0905545	209236	23.69521	562	5	6	New Zealan	Australia	Southern He	1	
27	1	0	10	0.9641836	78037	26.639757	652	7	8	New Zealan	Australia	Southern He	0	
28	0	0	10	7.3209806	43812	34.079333	405	7	9	Argentina	South Amer	Southern He	0	
29	0	0	8	6.1001377	251418	25.098978	797	1	7	South Korea	Asia	Northern He	1	
30	1	1	5	0.5534925	223345	21.447642	104	1	8	Argentina	South Amer	Southern He	0	
31	0	0	1	10.210611	56713	33.475207	711	0	4	Italy	Europe	Southern He	0	
32	0	0	5	9.1860306	48476	25.401338	584	3	9	South Africa	Africa	Southern He	0	
33	0	1	8	9.2859481	261120	34.677029	619	3	6	Australia	Australia	Southern He	1	
34	0	1	10	8.6727099	238403	27.274398	294	2	4	Germany	Europe	Northern He	0	
35	0	1	9	8.6179165	26953	30.107619	772	2	9	Colombia	South Amer	Northern He	1	
36	0	1	2	5.406193	292471	36.68611	366	4	6	China	Asia	Northern He	1	
37	0	1	3	6.1055049	36659	21.959107	777	1	10	Italy	Europe	Southern He	1	

Patient ID - Unique identifier for each patient

Age - Age of the patient

Sex - Gender of the patient (Male/Female)

Cholesterol - Cholesterol levels of the patient

Blood Pressure - Blood pressure of the patient (systolic/diastolic)

Heart Rate - Heart rate of the patient

Diabetes: Whether the patient has diabetes (Yes/No)

Family History - Family history of heart-related problems (1: Yes, 0: No)

Smoking: Smoking status of the patient (1: Smoker, 0: Non-smoker)

Obesity - Obesity status of the patient (1: Obese, 0: Not obese)

Alcohol Consumption - Level of alcohol consumption by the patient (None/Light/Moderate/Heavy)

Exercise Hours Per Week - Number of exercise hours per week

Diet - Dietary habits of the patient (Healthy/Average/Unhealthy)

Previous Heart Problems - Previous heart problems of the patient (1: Yes, 0: No)

Medication Use - Medication usage by the patient (1: Yes, 0: No)

Stress Level - Stress level reported by the patient (1-10)

Sedentary Hours Per Day - Hours of sedentary activity per day

Income - Income level of the patient

BMI - Body Mass Index (BMI) of the patient

Triglycerides - Triglyceride levels of the patient

Physical Activity Days Per Week - Days of physical activity per week

Sleep Hours Per Day - Hours of sleep per day

Country - Country of the patient

Continent - Continent where the patient resides

Hemisphere - Hemisphere where the patient resides

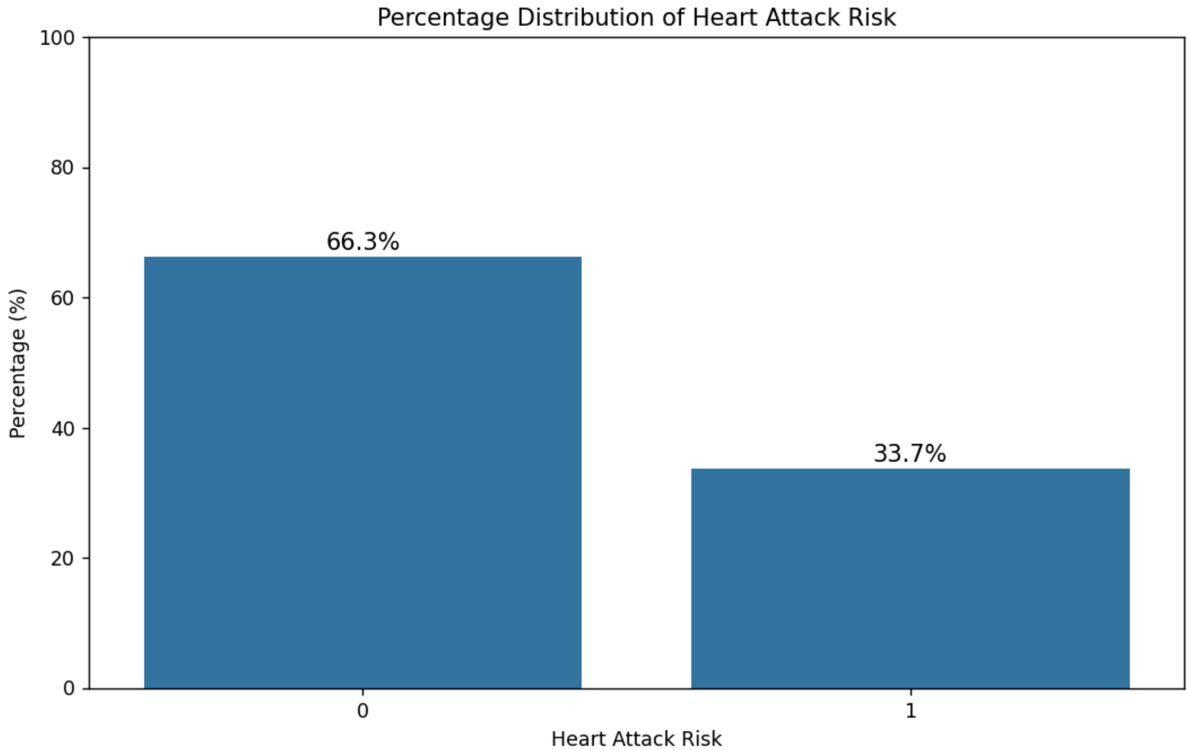
Heart Attack Risk - Presence of heart attack risk (1: Yes, 0: No)

3. Data Processing

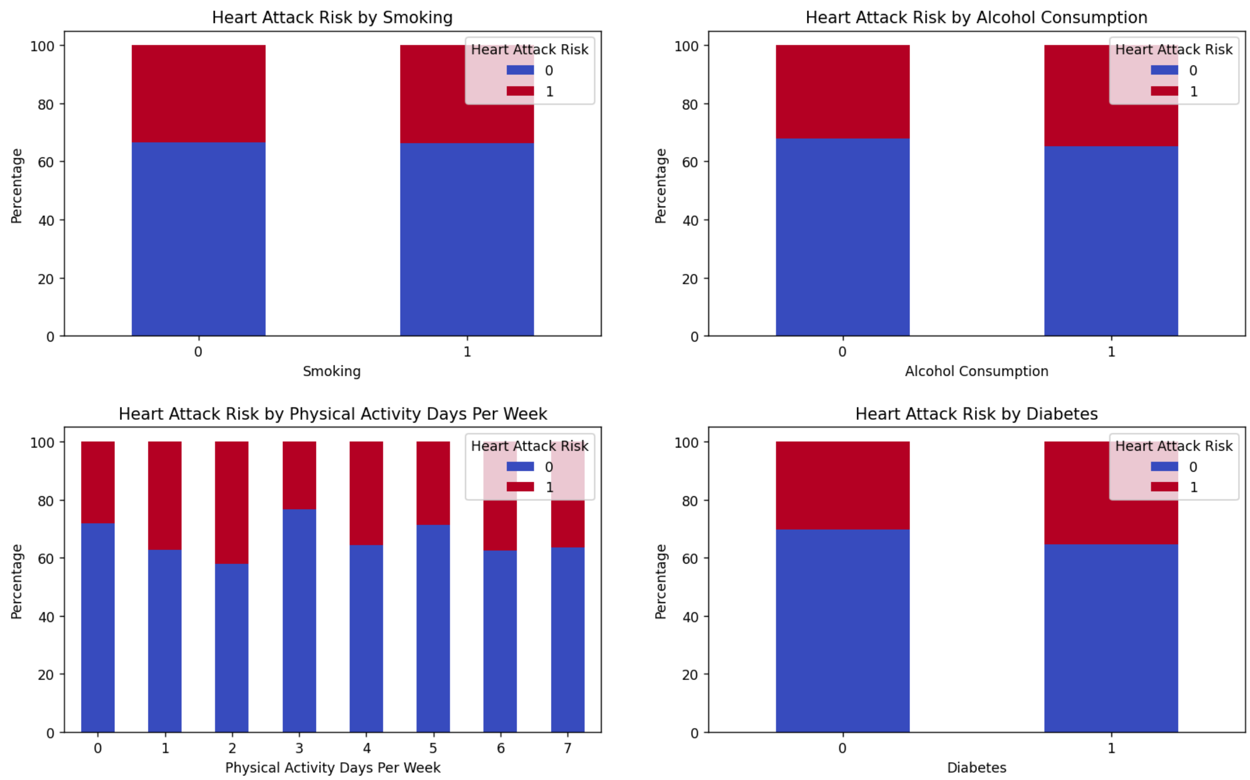
After running logistic regression, we analyzed and evaluated 4 factors that greatly affect a person's heart disease, including: Heart Rate, Sedentary Hours Per Day, Income and Sleep Hours Per Day

	A	B	C	D
1	Heart Rate	Sedentary Hours Per Day	Sleep Hours Per Day	Heart Attack Risk
2	50	10.06615579	10	0
3	106	5.213470234	4	1
4	69	3.279226179	7	0
5	55	0.487366463	7	0
6	46	1.400551734	7	0
7	56	9.675362396	9	1
8	98	6.999942543	10	1
9	102	10.9020322	5	0
10	91	1.912189688	6	1
11	42	6.744667311	9	0
12	98	11.54886853	5	0
13	67	5.005721327	6	0
14	79	2.681685339	5	0
15	110	7.010412508	6	0
16	83	7.001301563	7	0
17	65	2.998804781	5	0
18	44	11.82441616	10	0

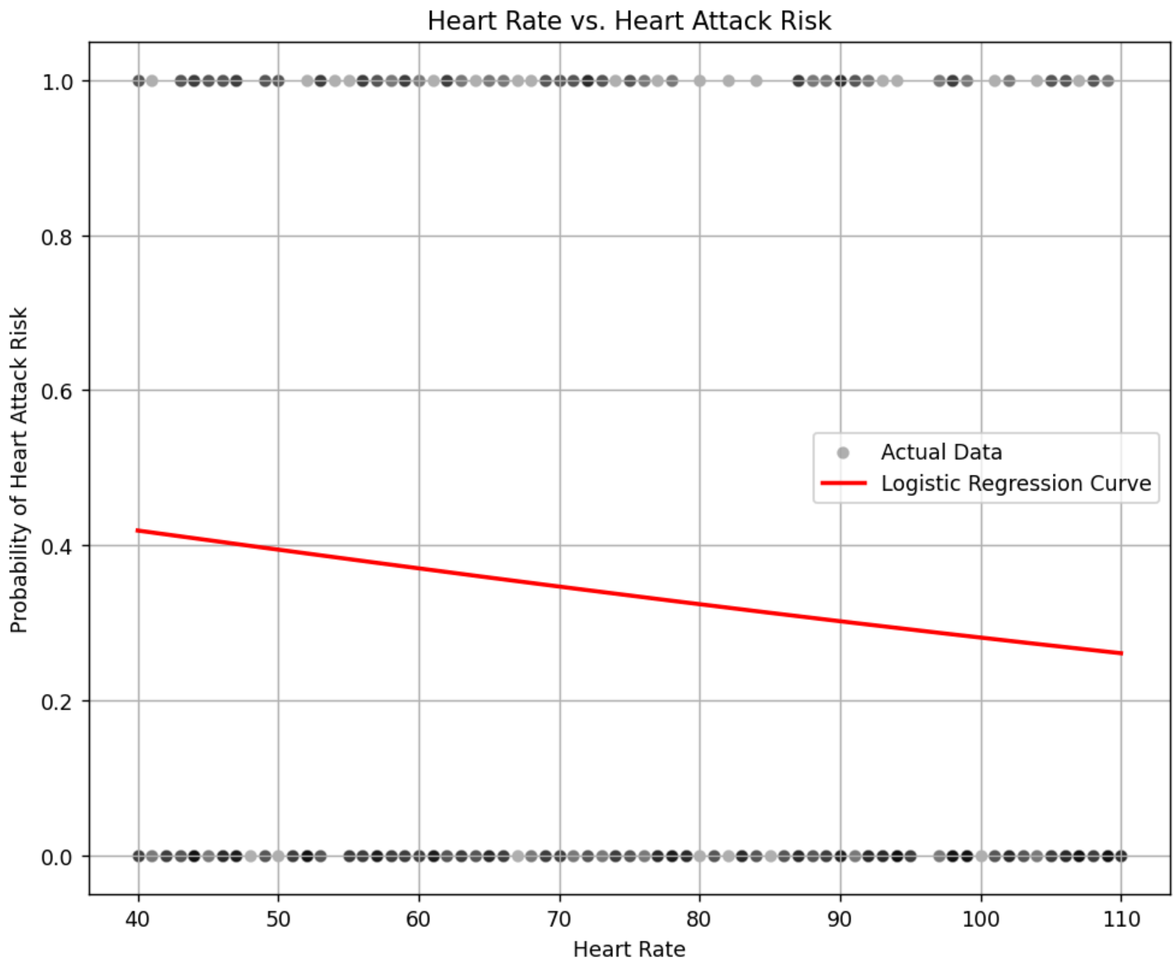
III. DATA ANALYSIS



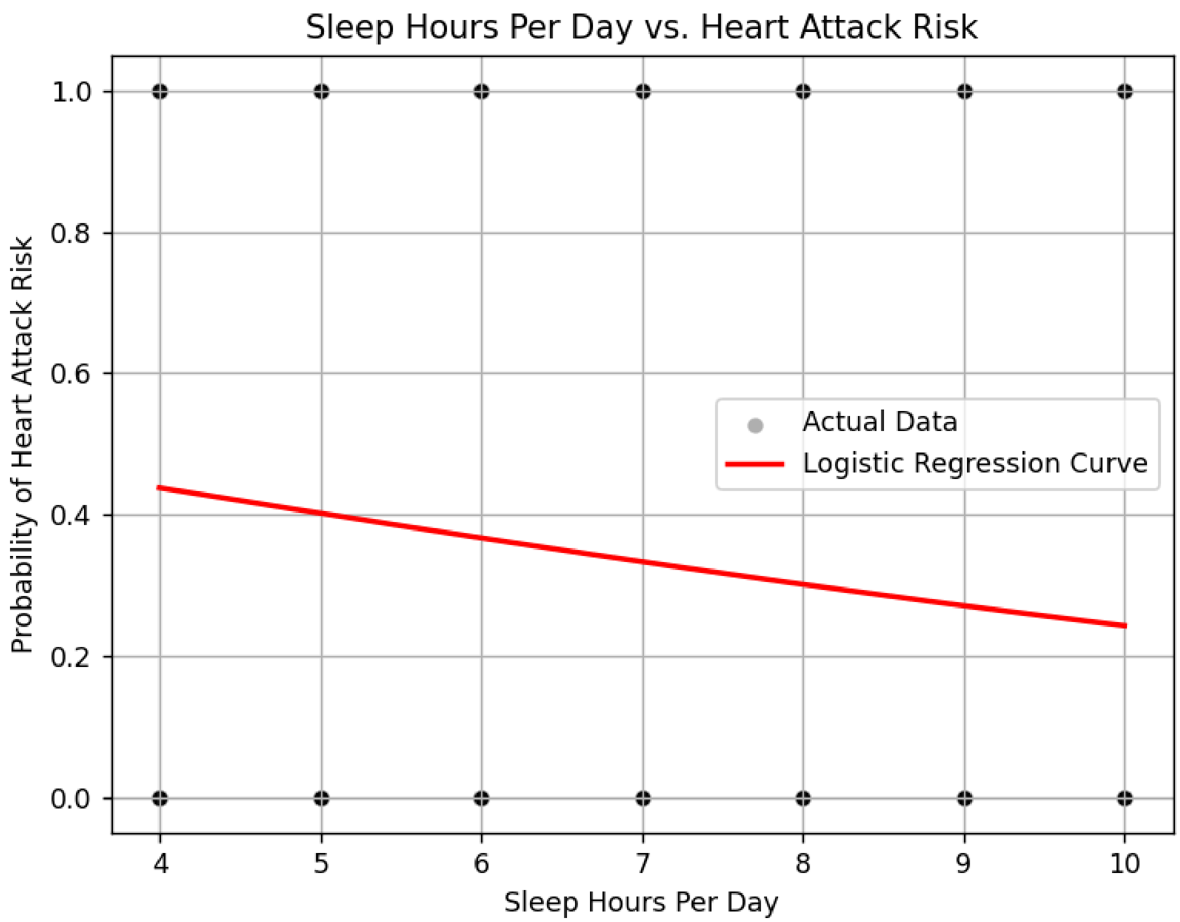
- The majority (66.3%) of individuals have no heart attack risk, nearly twice the percentage of those at risk (33.7%).
- The data suggests that most people in the dataset are not at risk of a heart attack.
- The chart is clear, but it could be improved by adding more detailed axis labels or using distinct colors for better differentiation.



- Smoking, Alcohol Consumption, and Diabetes all show similar trends, with no significant effect on heart attack risk in this dataset. The proportions of risk (red) and no risk (blue) remain nearly identical for both groups.
- Physical Activity appears to have some influence, as individuals with more active days per week tend to have a slightly lower heart attack risk.
- Despite common health concerns, smoking and alcohol consumption do not show a strong correlation with heart attack risk in this dataset.
- Diabetes, though a known risk factor, does not seem to significantly change heart attack risk distribution here.



- The logistic regression curve (red) shows a **slight downward trend**, suggesting that **higher heart rates are weakly associated with lower heart attack risk**.
- The data points (black dots) are mostly concentrated at **0 and 1**, indicating a **binary outcome with little variation** in probabilities.
- The model does **not show a strong correlation**, meaning heart rate **alone may not be a reliable predictor** of heart attack risk.
- Additional factors (e.g., cholesterol, blood pressure, lifestyle) should be considered for a more accurate risk assessment.



- The logistic regression curve (red line) suggests a negative correlation between sleep hours and heart attack risk. As sleep hours increase, the probability of a heart attack slightly decreases.
- The probability change is not drastic. From 4 to 10 sleep hours, the predicted heart attack risk only decreases from approximately 0.42 to 0.25. This indicates that sleep hours alone may not be a strong predictor of heart attack risk.
- Binary Data Distribution: The actual data points (black dots) are clustered at 0 and 1, meaning most people either had or did not have a heart attack—there is no middle ground. This suggests that other variables may play a bigger role in heart attack risk.

- Possible Confounding Factors:

    - Sleep alone may not determine heart attack risk.
    - Other lifestyle factors (exercise, diet, stress) might have stronger effects.
    - A more comprehensive model with multiple predictors could give better insights.

IV. KNOWLEDGE REQUIRED

This study uses logistic regression because:
Binary classification: Logistic regression is suitable when the dependent variable (the target to be predicted) is binary, like the risk of a heart attack (yes/no). Unlike linear regression, which works with continuous variables, logistic regression handles categorical outcomes effectively.

Analyzing influencing factors: Logistic regression can evaluate the impact of each risk factor, such as heart rate, sedentary hours, sleep duration, and income. The regression coefficients can be converted into odds ratios, helping to interpret the significance and strength of each factor.

Probability interpretation: Logistic regression outputs results as probabilities, making it easier to assess the likelihood of a heart attack based on risk factors. These probabilities range from 0 to 1, making the results clear and applicable in medical practice.

1. Univariate Logistic Regression

Univariate logistic regression is a statistical method used to analyze the relationship between a single independent variable and a binary dependent variable. Univariate logistic regression simplifies the initial analysis by focusing on one variable at a time. This approach helps filter out irrelevant variables before performing more complex multivariate analyses. It serves as a preliminary step to identify meaningful predictors, reducing noise and improving the efficiency of further statistical modeling.
Formula:

$$\boxed{P(Y=1) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}$$

Where:

X: Independent variable (e.g., age, blood pressure, cholesterol, etc.).
$\beta 0$: Intercept, representing the log-odds of the event occurring when $X=0$.
$\beta 1$: Coefficient of the independent variable X.
e: Euler's number (~2.718).
$P(Y=1|X)$: Probability of the event (heart attack risk) occurring given the value of X.

Only variables with a **p-value < 0.05** were considered statistically significant. The significant variables include:

- **Heart Rate:** Heart rate.
- **Sedentary Hours Per Day:** Hours spent sitting per day.
- **Income:** Income level.
- **Sleep Hours Per Day:** Hours of sleep per day.

2. Multivariate Logistic Regression

Multivariate logistic regression extends univariate regression by incorporating **multiple independent variables** to predict a binary outcome. It models the combined effect of multiple factors (e.g., heart rate, sedentary hours, income, sleep hours) on heart attack risk, providing a more comprehensive understanding of their interactions and overall influence.

Formula:

$$\boxed{P(Y=1) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n}}}$$

Where:
P(Y=1) is the probability of the outcome being 1.
e is Euler's number (~2.718).
β0 is the intercept.
β1,β2,...,βn are the regression coefficients for the independent variables X1,X2,...,Xn

Threshold = 0.3:

Predict class 1 if P(Y=1) ≥ 0.3.

Predict class 0 if P(Y=1) < 0.3.

V. CODE IMPLEMENTATION

Read file

```
file_path = r"heart1.csv"
try:
    df = pd.read_csv(file_path)
    print("Đọc file dữ liệu thành công!")
except FileNotFoundError:
    print("Lỗi: Không tìm thấy file heart_attack_prediction_dataset.csv. Vui lòng kiểm tra đường dẫn!")
    exit()
```

```
Đọc file dữ liệu thành công!
```

Data transforming

```
le = LabelEncoder()
categorical_cols = ['Sex', 'Diet', 'Country', 'Continent', 'Hemisphere']
for col in categorical_cols:
    if col in df.columns:
        df[col] = le.fit_transform(df[col])
    else:
        print(f"Cột {col} không tồn tại trong dữ liệu!")

if 'Blood Pressure' in df.columns:
    df[['Systolic BP', 'Diastolic BP']] = df['Blood Pressure'].str.split('/', expand=True).astype(float)
    df = df.drop('Blood Pressure', axis=1)
else:
    print("Cột 'Blood Pressure' không tồn tại trong dữ liệu!")

numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
if 'Heart Attack Risk' in numeric_cols:
    numeric_cols.remove('Heart Attack Risk')
else:
    print("Lỗi: Cột 'Heart Attack Risk' không tồn tại trong dữ liệu!")
    exit()
target = 'Heart Attack Risk'

print("Dữ liệu đã được tiền xử lý!")
```

```
Dữ liệu đã được tiền xử lý!
```

Purpose:

- The list categorical_cols contains categorical (non-numeric) features.
- LabelEncoder converts these categorical values into integers so the model can process them.

Ex: 'Sex' contains 'Male' and 'Female'. LabelEncoder will convert these values into 0 and 1 to process.

If the **"Blood Pressure"** column exists, it is split into two separate numerical columns:

- **Systolic BP** (higher blood pressure value)
- **Diastolic BP** (lower blood pressure value)

Blood pressure values are typically stored as **"120/80"**, so str.split('/') is used to separate them.

Simple Logistic Regression

```
significant_vars = []
print("\nHồi quy Logistic đơn biến:")
for col in numeric_cols:
    try:
        X = df[[col]].dropna()
        X = sm.add_constant(X)
        y = df[target].loc[X.index]
        model = sm.Logit(y, X).fit(disp=0)
        p_value = model.pvalues[col]
        print(f"{col}: P-value = {p_value:.4f}")
        if p_value < 0.05:
            significant_vars.append(col)
    except Exception as e:
        print(f"Lỗi khi chạy hồi quy cho {col}: {e}")

print("\nCác biến có ảnh hưởng (P-value < 0.05):", significant_vars)
```

Purpose:

- Stores variables (features) that have a statistically significant relationship with "Heart Attack Risk".
- X = df[[col]].dropna() selects the current feature and removes missing values.
- sm.add_constant(X) adds a **constant (intercept)** term for the regression model.
- y = df[target].loc[X.index] ensures that y (the target variable) only includes rows where X is not NaN.
- Uses statsmodels to fit a **logistic regression model** where the **predictor** is col and the **target** is "Heart Attack Risk".
- p_value = model.pvalues[col]: Extracts the **p-value** for the feature.
- **If p-value < 0.05, the feature is considered significant** (it has a meaningful impact on "Heart Attack Risk").

Multivariate Logistic Regression for Heart Attack Risk Prediction:

```python
if significant_vars:
    X_multi = df[significant_vars].dropna()
    X_multi = sm.add_constant(X_multi)
    y = df[target].loc[X_multi.index]
    try:
        multi_model = sm.Logit(y, X_multi).fit()
        print("\nKết quả hồi quy Logistic đa biến:")
        print(multi_model.summary())

    except Exception as e:
        print(f"Lỗi khi chạy hồi quy đa biến: {e}")
else:
    print("Không có biến nào có ý nghĩa thống kê để chạy hồi quy đa biến!")
```

```python
significant_vars = [var for var in significant_vars if var != 'Income']
if significant_vars:
    X_multi = df[significant_vars].dropna()
    X_multi = sm.add_constant(X_multi)
    y = df[target].loc[X_multi.index]
    try:
        multi_model = sm.Logit(y, X_multi).fit()

        coefs = multi_model.params
        equation = "logit(P(Heart Attack Risk)) = "
        equation += f"{coefs['const']:.4f}"
        for var, coef in coefs.items():
            if var != 'const' and var != 'Income':
                equation += f" + {coef:.4f} * {var}"
        print("\nCông thức dự đoán:")
        print(equation)

        example = X_multi.iloc[0:1]
        prob = multi_model.predict(example)
        print(f"\nXác suất dự đoán cho mẫu đầu tiên: {prob[0]:.4f}")
        X_train, X_test, y_train, y_test = train_test_split(X_multi, y, test_size=0.2, random_state=42)
        log_reg = LogisticRegression(max_iter=1000)
        log_reg.fit(X_train, y_train)
        accuracy = log_reg.score(X_test, y_test)
        print(f"Độ chính xác của mô hình trên tập kiểm tra: {accuracy:.4f}")

    except Exception as e:
        print(f"Lỗi khi chạy hồi quy đa biến: {e}")
else:
    print("Không có biến nào có ý nghĩa thống kê để chạy hồi quy đa biến!")
```

Purpose:

This code segment evaluates and models the relationship between statistically significant independent variables and "Heart Attack Risk".

- “significant_vars”: Stores independent variables (features) that have a statistically significant relationship with the target variable "Heart Attack Risk".
- “X_multi = df[significant_vars].dropna()”: Selects the significant features from the dataframe and removes rows with missing values.
- `sm.add_constant(X_multi)`: Adds a constant (intercept) term to the matrix of independent variables for the regression model.
- “y = df[target].loc[X_multi.index]”: Ensures that the dependent variable (target) aligns with the rows of “X_multi” where no data is missing.
- Uses “statsmodels” to fit a multivariable logistic regression model, with “significant_vars” as predictors and "Heart Attack Risk" as the target.
- Constructs and displays the logit prediction equation based on the model’s coefficients.
- Predicts the probability of "Heart Attack Risk" for the first sample in the dataset.
- Splits the data into training (80%) and testing (20%) sets, fits a logistic regression model using “scikit-learn”, and evaluates its accuracy on the test set.
- Handles exceptions (e.g., model convergence issues or invalid data) and provides feedback if no significant variables are available for modeling.

VI. RESULTS

- Univariate Logistic Regression identified 4 significant factors affecting heart attack risk: Heart Rate ($p = 0.0437$), Sedentary Hours Per Day ($p = 0.0067$), Income ($p = 0.0461$), and Sleep Hours Per Day ($p = 0.0072$).

  ($p < 0.05$ → This factor has a significant effect on heart attack risk. The chance of this effect being random is very low.)

  ★ **Heart Rate ($p = 0.0437$):** $p < 0.05$ → Heart rate affects heart attack risk.
  ★ **Sedentary Hours Per Day ($p = 0.0067$):** $p < 0.05$ → Spending more hours being inactive has a strong effect.
  ★ **Income ($p = 0.0461$):** $p < 0.05$ → Income seems to be related, but not very strong.
  ★ **Sleep Hours Per Day ($p = 0.0072$):** $p < 0.05$ → Sleep has a big effect on heart attack risk.

- ★ **Income (p = 0.064) in multivariate regression:** p > 0.05 → When considering other factors, income is no longer important.
- Multivariate Logistic Regression confirmed that Heart Rate, Sedentary Hours Per Day, and Sleep Hours Per Day remained significant, while Income (p = 0.064) was no longer statistically important.
- Predicted probability for the first sample: **20.02%** (This means that for the first sample, the estimated risk of a heart attack is 15.58%.)
- Model accuracy on test data: **67.90%** (The model correctly predicts **67,90**% of the cases in the test set. This shows that the model has quite good accuracy but still has some errors)

VII. CONCLUSION

- This project focused on predicting the risk of heart disease using logistic regression models. Univariate and multivariate analyses revealed that variables that significantly influenced the risk of heart disease included:

- **Heart Rate**
- **Sedentary Hours Per Day**,
- **Income**
- **Sleep Hours Per Day**

Sleep Hours Per Day and **Sedentary Hours Per Day** have the strongest impact.

The model accuracy of 67.90% means that the logistic regression model correctly predicts about 67.90% of the cases in the test set. This is an average level of accuracy, showing that the model performs better than random guessing (50%). However, it is not reliable enough for real medical applications.

In summary, the Logistic regression model provided important insights into factors influencing the risk of heart disease, and demonstrated the potential of applying machine learning in healthcare to support early detection and timely intervention.

 # Code

```
import pandas as pd
import statsmodels.api as sm
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder


file_path = r"heart1.csv"
try:
    df = pd.read_csv(file_path)
    print("Đọc file dữ liệu thành công!")
except FileNotFoundError:
    print("Lỗi: Không tìm thấy file heart_attack_prediction_dataset.csv. Vui lòng kiểm tra
đường dẫn!")
    exit()


le = LabelEncoder()
categorical_cols = ['Sex', 'Diet', 'Country', 'Continent', 'Hemisphere']
for col in categorical_cols:
    if col in df.columns:
        df[col] = le.fit_transform(df[col])
    else:
        print(f"Cột {col} không tồn tại trong dữ liệu!")

if 'Blood Pressure' in df.columns:
    df[['Systolic BP', 'Diastolic BP']] = df['Blood Pressure'].str.split('/',
expand=True).astype(float)
    df = df.drop('Blood Pressure', axis=1)
else:
    print("Cột 'Blood Pressure' không tồn tại trong dữ liệu!")

numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
if 'Heart Attack Risk' in numeric_cols:
    numeric_cols.remove('Heart Attack Risk')
else:
    print("Lỗi: Cột 'Heart Attack Risk' không tồn tại trong dữ liệu!")
    exit()
```

```
target = 'Heart Attack Risk'

print("Dữ liệu đã được tiền xử lý!")


significant_vars = []
print("\nHồi quy Logistic đơn biến:")
for col in numeric_cols:
    try:
        X = df[[col]].dropna()
        X = sm.add_constant(X)
        y = df[target].loc[X.index]
        model = sm.Logit(y, X).fit(disp=0)
        p_value = model.pvalues[col]
        print(f"{col}: P-value = {p_value:.4f}")
        if p_value < 0.05:
            significant_vars.append(col)
    except Exception as e:
        print(f"Lỗi khi chạy hồi quy cho {col}: {e}")

print("\nCác biến có ảnh hưởng (P-value < 0.05):", significant_vars)

if significant_vars:
    X_multi = df[significant_vars].dropna()
    X_multi = sm.add_constant(X_multi)
    y = df[target].loc[X_multi.index]
    try:
        multi_model = sm.Logit(y, X_multi).fit()
        print("\nKết quả hồi quy Logistic đa biến:")
        print(multi_model.summary())

    except Exception as e:
        print(f"Lỗi khi chạy hồi quy đa biến: {e}")
else:
    print("Không có biến nào có ý nghĩa thống kê để chạy hồi quy đa biến!")


significant_vars = [var for var in significant_vars if var != 'Income']
if significant_vars:
    X_multi = df[significant_vars].dropna()
```

```
    X_multi = sm.add_constant(X_multi)
    y = df[target].loc[X_multi.index]
    try:
        multi_model = sm.Logit(y, X_multi).fit()

        coefs = multi_model.params
        equation = "logit(P(Heart Attack Risk)) = "
        equation += f"{coefs['const']:.4f}"
        for var, coef in coefs.items():
            if var != 'const' and var != 'Income':
                equation += f" + {coef:.4f} * {var}"
        print("\nCông thức dự đoán:")
        print(equation)

        example = X_multi.iloc[0:1]
        prob = multi_model.predict(example)
        print(f"\nXác suất dự đoán cho mẫu đầu tiên: {prob[0]:.4f}")
        X_train, X_test, y_train, y_test = train_test_split(X_multi, y, test_size=0.2,
random_state=42)
        log_reg = LogisticRegression(max_iter=1000)
        log_reg.fit(X_train, y_train)
        accuracy = log_reg.score(X_test, y_test)
        print(f"Độ chính xác của mô hình trên tập kiểm tra: {accuracy:.4f}")

    except Exception as e:
        print(f"Lỗi khi chạy hồi quy đa biến: {e}")
else:
    print("Không có biến nào có ý nghĩa thống kê để chạy hồi quy đa biến!")

# ========= Data Cleaning =========
df_cleaned = df[significant_vars + [target]].dropna()

# Create Folder
output_dir = "data cleaning"
os.makedirs(output_dir, exist_ok=True)

# Save File
cleaned_file_path = os.path.join(output_dir, "data_cleaning.csv")
df_cleaned.to_csv(cleaned_file_path, index=False)
print(f"Dữ liệu đã được làm sạch và lưu tại: {cleaned_file_path}")
```