Heart-Disease Capstone Project

Initial Report and Exploratory Data Analysis (EDA)

Professional Certificate in Machine Learning and Artificial Intelligence

UC Berkeley College of Engineering / Haas School of Business

by

Sara Obergassel

26 April 2025

1. Heart-Disease Capstone Project

This project aims to explore and model health-related risk factors to predict whether an individual is likely to experience heart disease or stroke. The model is developed as part of a machine learning capstone project at UC Berkeley Haas School of Business, using real-world medical data.

2. Expected Data Source

The dataset used is from Kaggle titled heart_disease.csv, which includes basic personal and health information in CSV format

"https://www.kaggle.com/datasets/mirzahasnine/heart-disease-dataset?select=heart_disease.csv"

3. Techniques

To address the research question and prepare the data for modeling, the following techniques and methodologies were applied:

- Data cleaning (handling missing values, outlier analysis, feature engineering)
- Exploratory Data Analysis (EDA) to understand distributions and relationships
- Encoding of categorical variables and feature scaling
- Baseline classification model using Logistic Regression
- Model evaluation using accuracy, precision, recall, and confusion matrix

4. Expected Results

I aim to create a reliable tool that can predict who might develop heart disease based on their basic personal and health details. This tool will show us which factors are most important for predicting heart disease, helping doctors and patients take action early.

5. Why This Question is Important

Heart disease is a major cause of death around the world. If we can predict heart disease early, we can help people make lifestyle changes or get medical treatment sooner. Without answering this question, people might not know they are at risk, missing the chance to prevent serious health issues. A simple predictive tool can help doctors and patients make better decisions, saving lives and reducing healthcare costs by allowing earlier treatment and better personal care plans.

6. Import Libraries & Load Dataset

In the first step, the following libraries were loaded

- pandas
- numpy
- matplotlib.pyplot
- seaborn
- sklearn.model_selection (train_test_split)
- sklearn.linear_model (LogisticRegression)
- sklearn.metrics (accuracy_score, classification_report, confusion_matrix)

After that the used dataset from https://www.kaggle.com/datasets/mirzahasnine/heart-disease-dataset?select=heart_disease.csv was downloaded and saved as "heart_disease.csv".

Data Overview & Initial Checks

In this section, the insights of the dataset have been investigated.

- The dataset df was previewed by the code df.head(), to have a view of the first 5 rows, which contains the following columns:

Gender, age, education, currentSmoker, cigsPerDay, BPMeds, prevalentStroke, prevalentHyp, diabetes, totChol, sysBP, diaBP, BMI, heartrate, glucose, and Heart_ stroke (target variable)

	Gender	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	Heart_ stroke
0	Male	39	postgraduate	0	0.0	0.0	no	0	0	195.0	106.0	70.0	26.97	80.0	77.0	No
1	Female	46	primaryschool	0	0.0	0.0	no	0	0	250.0	121.0	81.0	28.73	95.0	76.0	No
2	Male	48	uneducated	1	20.0	0.0	no	0	0	245.0	127.5	80.0	25.34	75.0	70.0	No
3	Female	61	graduate	1	30.0	0.0	no	1	0	225.0	150.0	95.0	28.58	65.0	103.0	yes
4	Female	46	graduate	1	23.0	0.0	no	0	0	285.0	130.0	84.0	23.10	85.0	85.0	No

- Using df.info(), the types of each column of the dataset were found as follows:

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Gender           4238 non-null   object
 1   age              4238 non-null   int64
 2   education        4133 non-null   object
 3   currentSmoker    4238 non-null   int64
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   object
 7   prevalentHyp     4238 non-null   int64
 8   diabetes         4238 non-null   int64
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  Heart_ stroke    4238 non-null   object
dtypes: float64(8), int64(4), object(4)
memory usage: 529.9+ KB
```

- The shape of the dataset was checked and found:

  Rows: 4238
  Columns: 16

- These columns contain missing values:

Column	Missing Values
education	105
cigsPerDay	29
BPMeds	53
totChol	50
BMI	19
heartRate	1
glucose	388

- There are no duplicates

- Unique values in categorical features are:
  - Gender: 2 unique values
  - education: 4 unique values
  - prevalentStroke: 2 unique values
  - Heart_ stroke: 2 unique values

Column	Unique Categories
Gender	2 (Male, Female)
education	4 (graduate, postgraduate, etc.)
prevalentStroke	2 (yes, no)
Heart_ stroke	2 (yes, No)

7. Data Cleaning

In this step, I made a copy of the dataset “df_clean = df.copy()” and did the following steps for df_clean:

- Cleaned column names (remove spaces, fix inconsistent naming)
- Renaming the target column for clarity 'Heart_stroke' to 'HeartStroke'.
- Handling missing values by imputing numerical columns with median
- Convert categorical columns to lowercase to avoid for example 'No' vs 'no' issues

The cleaned dataset df_clean is previewed for checking using df_clean.info():

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Gender           4238 non-null   object 
 1   age              4238 non-null   int64  
 2   education        4238 non-null   object 
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4238 non-null   float64
 5   BPMeds           4238 non-null   float64
 6   prevalentStroke  4238 non-null   object 
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4238 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4238 non-null   float64
 13  heartRate        4238 non-null   float64
 14  glucose          4238 non-null   float64
 15  HeartStroke      4238 non-null   object 
dtypes: float64(8), int64(4), object(4)
memory usage: 529.9+ KB
```

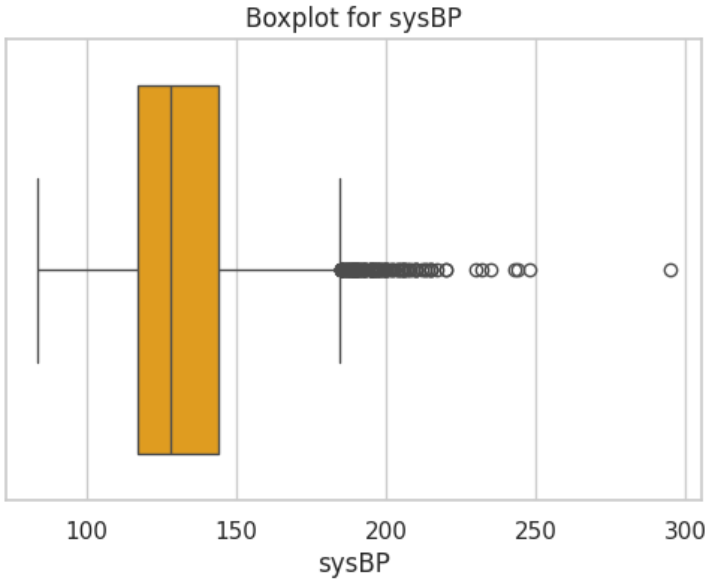
- The missing values in each column were checked again, the result was 0 missing:

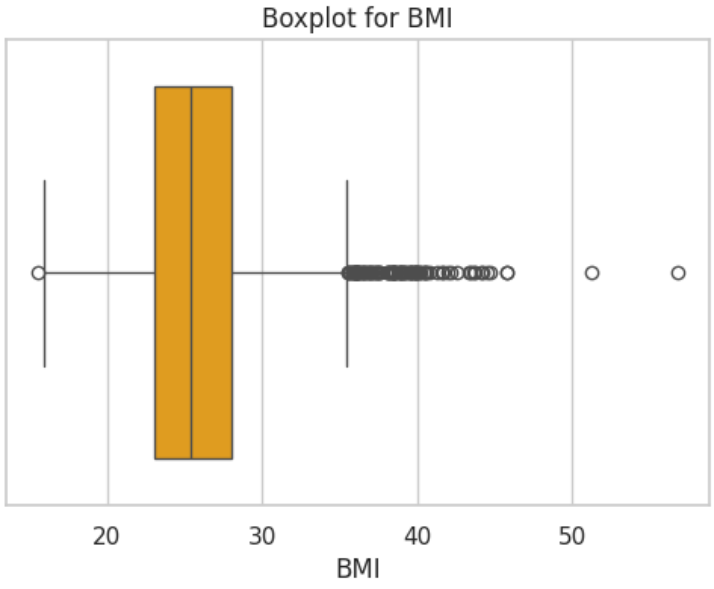
Gender	0
age	0
education	0
currentSmoker	0
cigsPerDay	0
BPMeds	0
prevalentStroke	0
prevalentHyp	0
diabetes	0
totChol	0
sysBP	0
diaBP	0
BMI	0
heartRate	0
glucose	0
HeartStroke	0

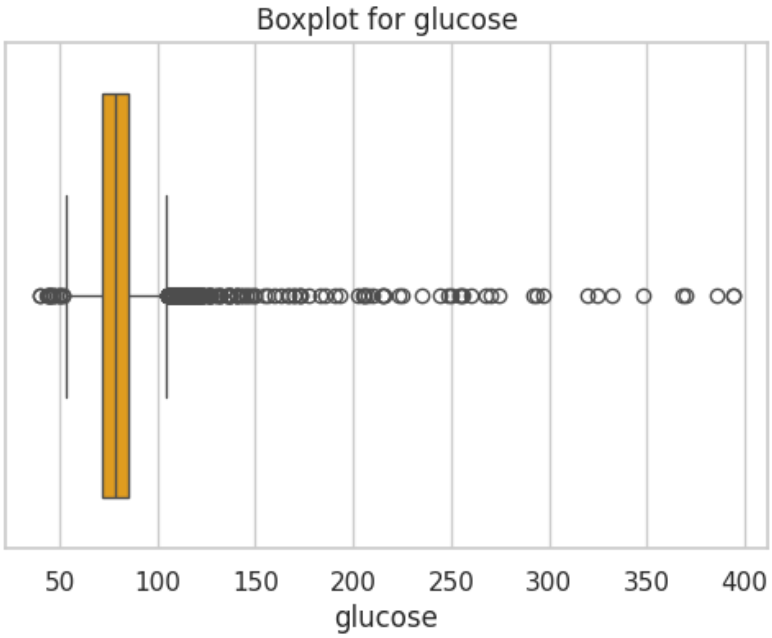
8. Outlier Analysis

Outlier analysis was conducted using boxplots and the Interquartile Range (IQR) method. Features like BMI, glucose, and systolic blood pressure (sysBP) showed some extreme values, which were flagged as potential outliers.

```
Number of outliers in age: 0
Number of outliers in currentSmoker: 0
Number of outliers in cigsPerDay: 12
Number of outliers in BPMeds: 124
Number of outliers in prevalentHyp: 0
Number of outliers in diabetes: 109
Number of outliers in totChol: 57
Number of outliers in sysBP: 126
Number of outliers in diaBP: 81
Number of outliers in BMI: 97
Number of outliers in heartRate: 76
Number of outliers in glucose: 262
```





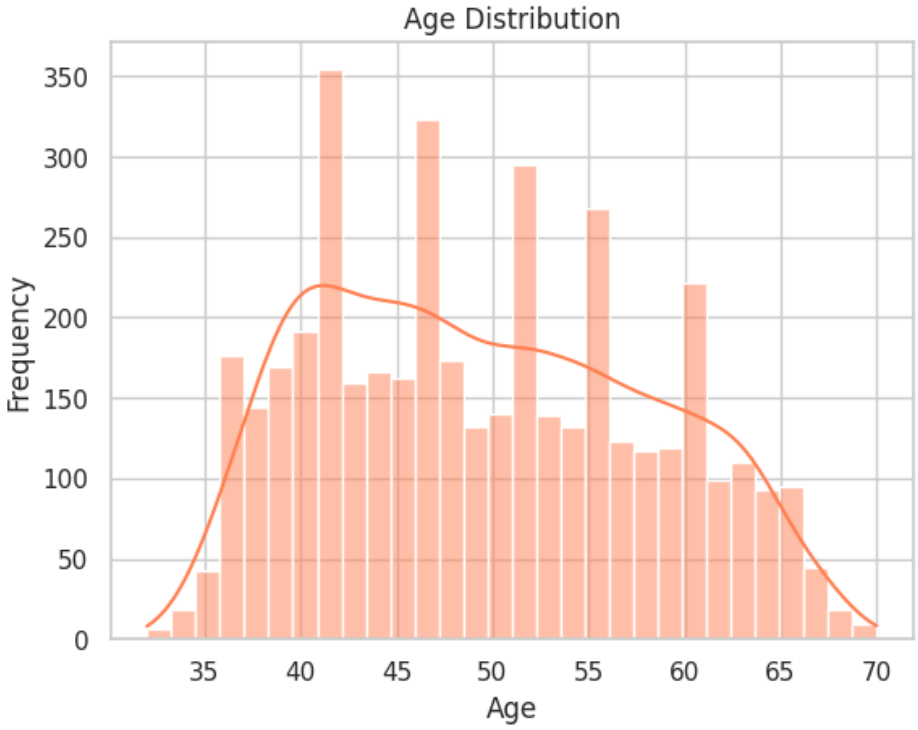


The outeliers have been removed from df_clean and saved as df_clean_no_outliers.

9. Visual Exploratory Data Analysis (EDA)

In this section the data has been explored as follows:

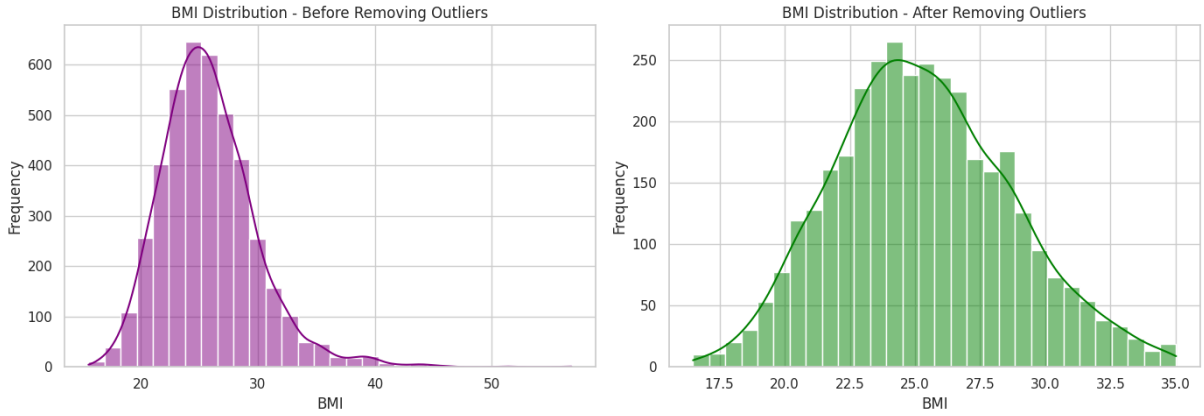
9.1- The following plot shows the age distribution:



The side-by-side comparison of Age distribution shows that removing outliers had minimal impact on the overall age distribution, confirming that the dataset is stable in this feature

The age distribution is approximately normal, with most individuals falling between 40 and 60 years old. This suggests the dataset mainly includes middle-aged adults, a group at higher risk for heart disease and stroke.

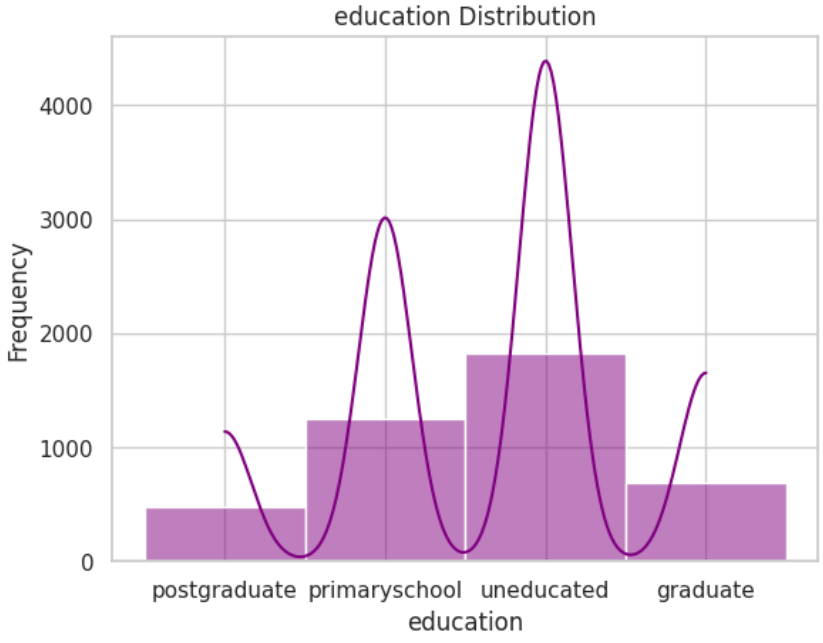
9.2 The following plot shows the BMI distribution



The BMI distribution is right-skewed, meaning more people are slightly overweight or obese. A few individuals have high BMI values above 40, which are considered extreme and may be outliers.

The side-by-side comparison shows that removing outliers made the BMI distribution tighter and reduced extreme values above 40.

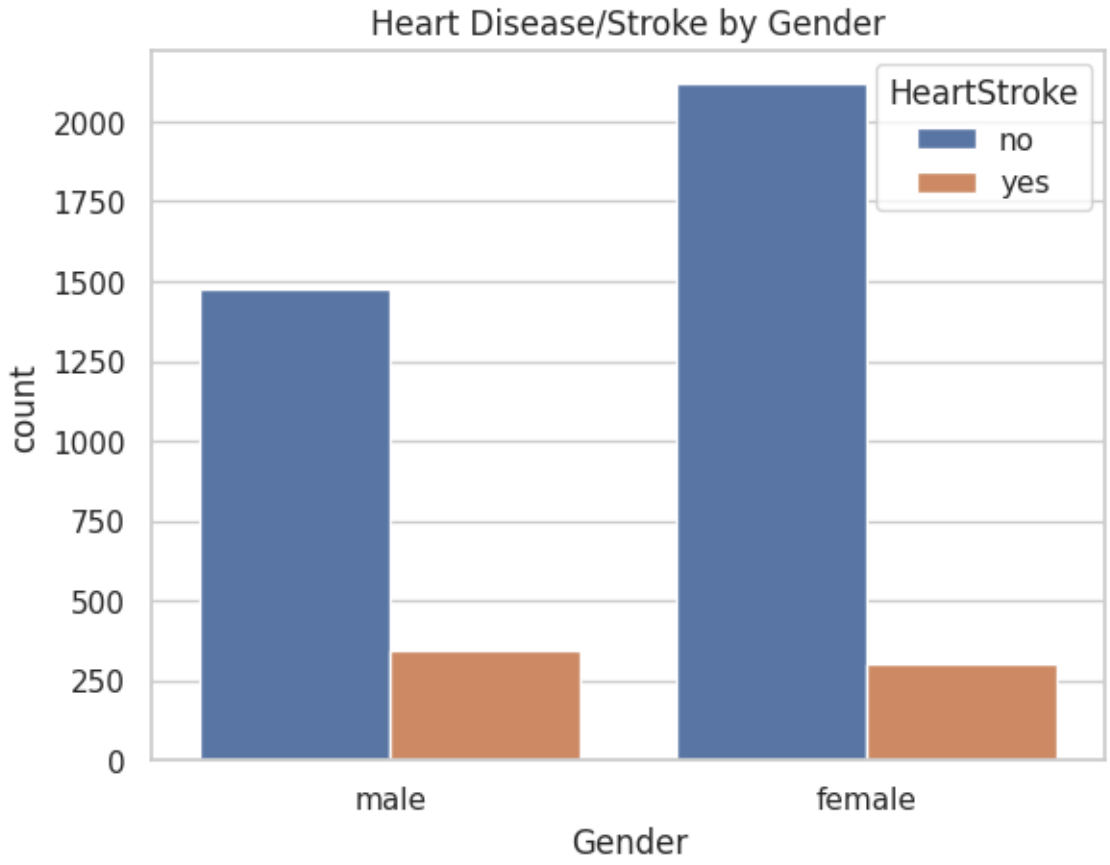
9.3 The distribution about education can be seen in the following graph:

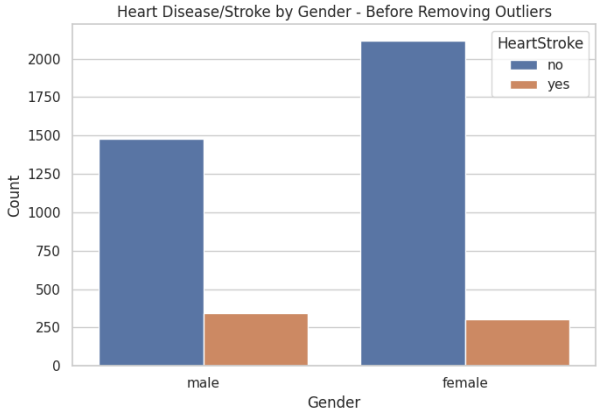


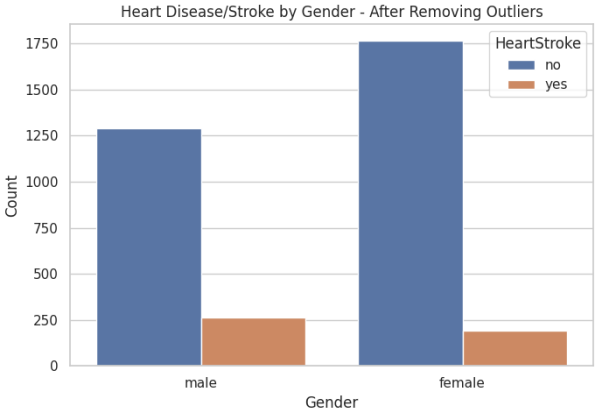
There are more people with low education (primary or no schooling) than with higher education (graduate or postgraduate).

9.4 Bar Plots for Categorical vs Target (HeartStroke): Gender vs HeartStroke

Both males and females are represented in the dataset. However, the number of people with heart disease/stroke appears slightly higher among males. This could indicate a higher risk among male participants, though more analysis is needed.

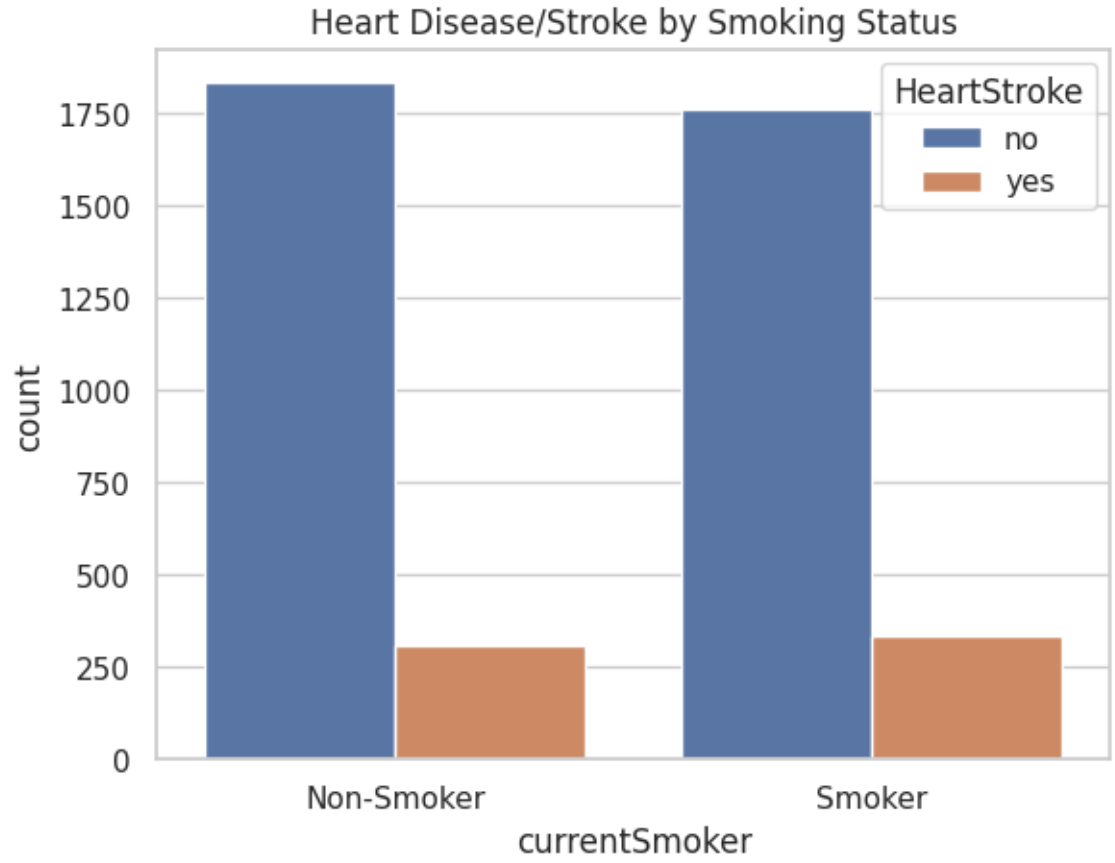


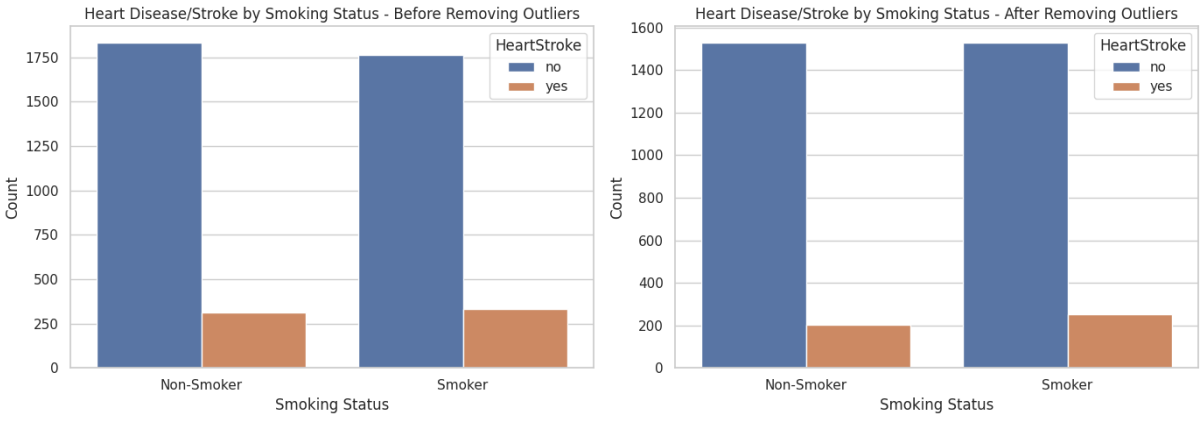




4.5 Bar Plots for Categorical vs Target (HeartStroke): Smoking status vs HeartStroke

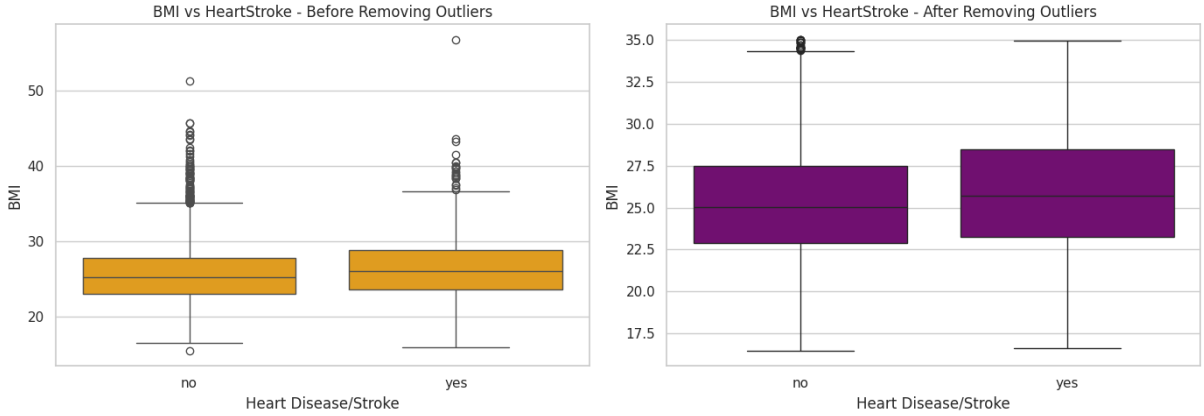
Smokers are more likely to be in the heart disease/stroke group than non-smokers. This supports the well-known link between smoking and cardiovascular risk.





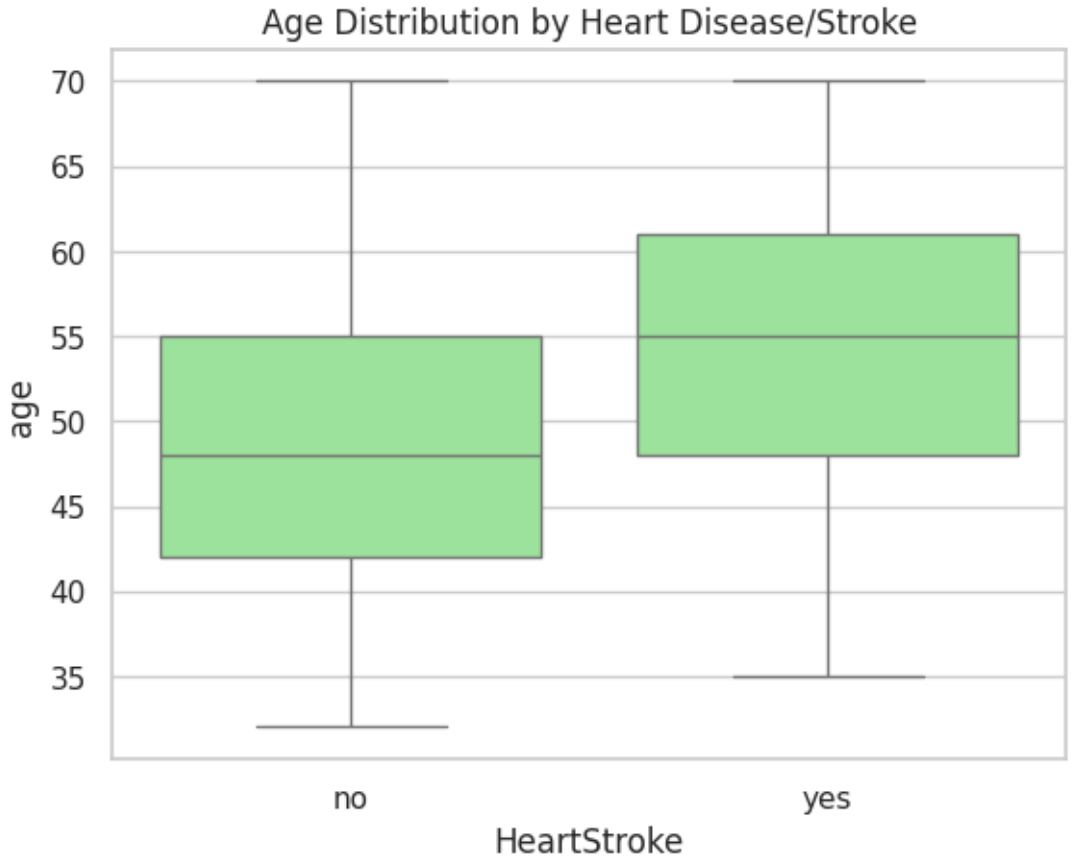
4.6 Boxplots (To Check Impact of Numeric on Target): BMI vs HeartStroke

The boxplot shows that individuals with heart disease or stroke tend to have slightly higher BMI on average. However, the difference is not dramatic, suggesting BMI may be a contributing factor, but not the strongest one.



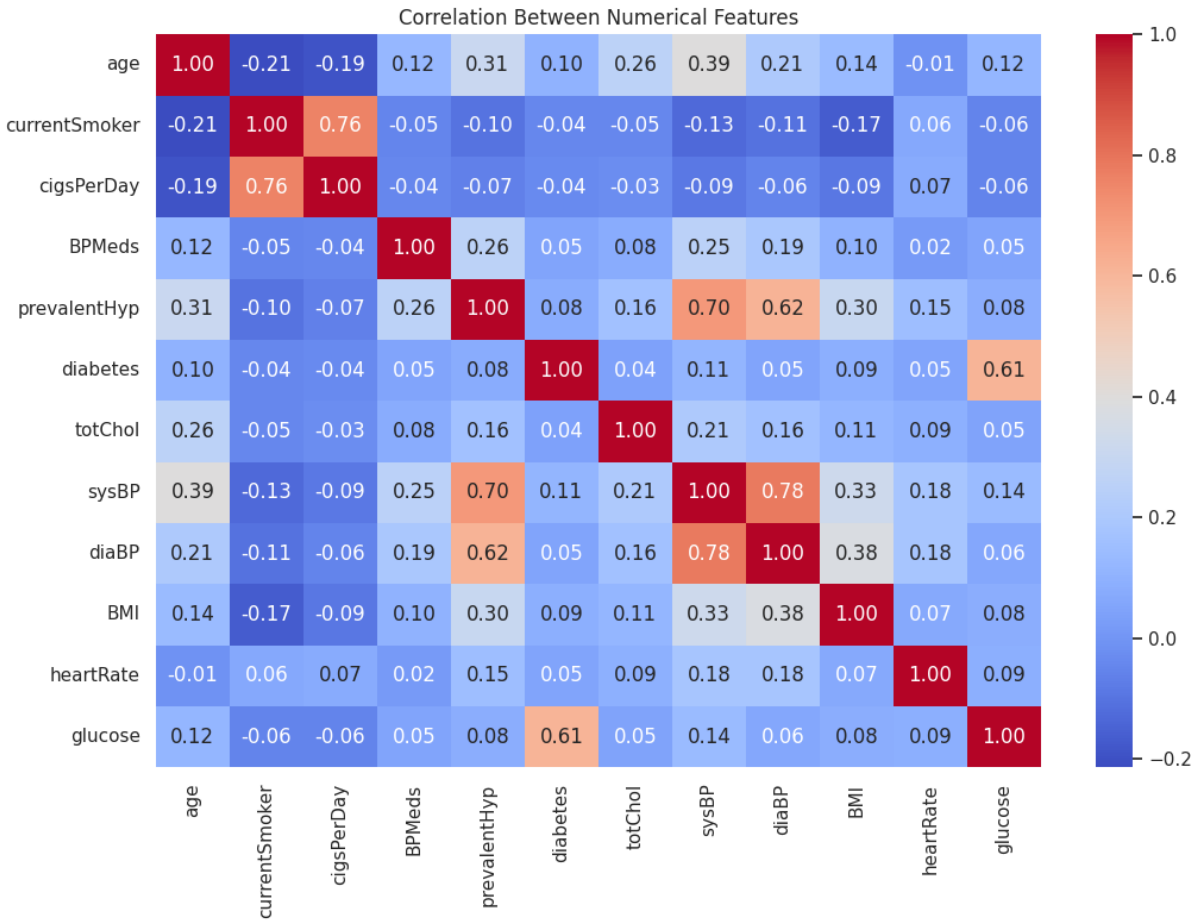
4.7 Age vs HeartStroke

There is a clear trend: individuals who experienced heart disease or stroke tend to be older. This reinforces the role of age as a key risk factor in cardiovascular events.



4.8 Correlation Heatmap (Numerical Features)

The heatmap highlights positive correlations between systolic blood pressure, BMI, and age with heart-related outcomes. It also shows that some variables like diastolic BP and heart rate are only weakly related to the target. This helps narrow down which variables may be most useful for prediction



4.9 BMI Category vs Heart Disease/Stroke

Firstly, I categorized BMI as follows:

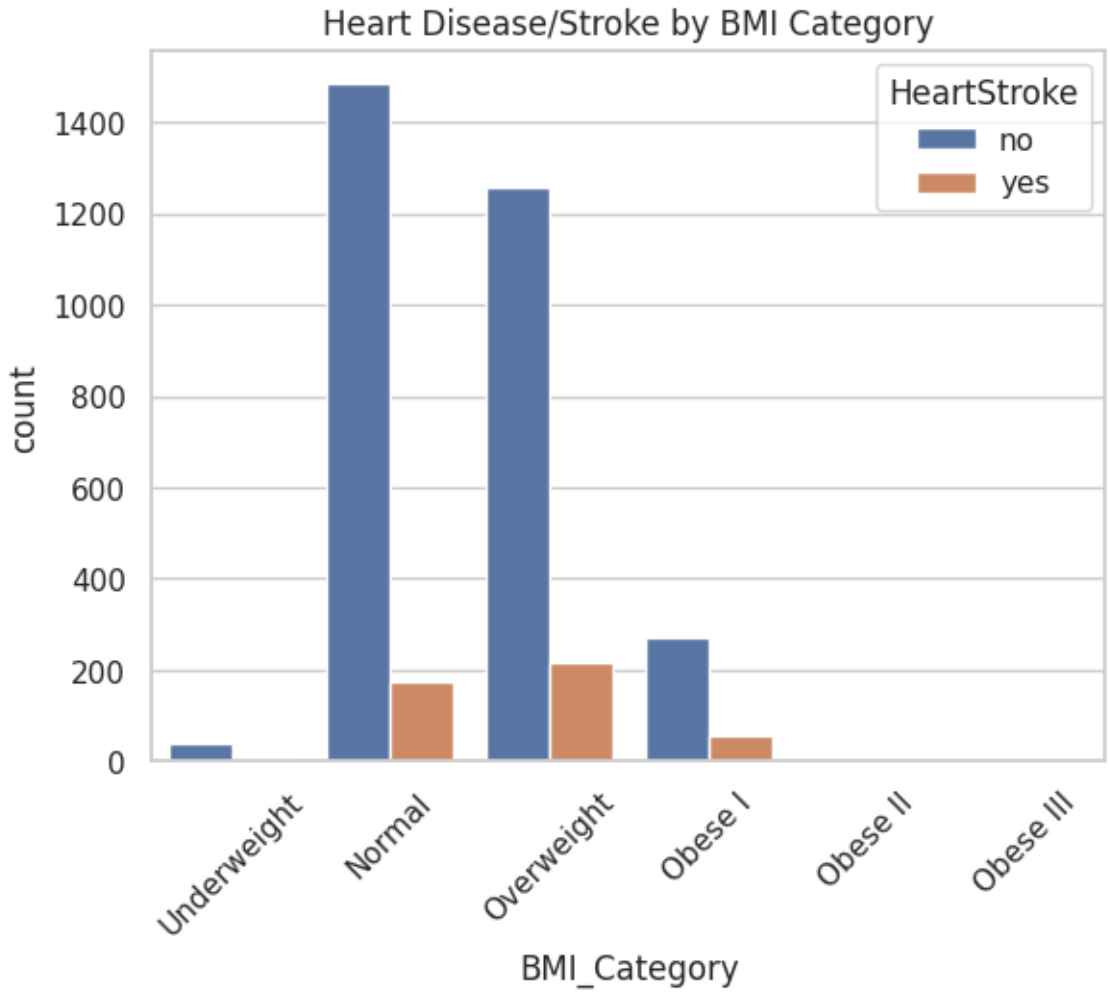
BMI_Category	count
Normal	1660
Overweight	1478
Obese I	324
Underweight	45
Obese II	3

dtype: int64

As can be seen in the plot of BMI Category vs Heart Disease/Stroke, people in the Overweight category appear to have higher risk than those at Normal weight.

Very high BMI (Obese II and III) might also increase risk, but the sample size is too small in the dataset to be confident.

Being "Normal" BMI doesn't completely protect against heart disease/stroke, other factors (like age, smoking, blood pressure) are likely at play.



10. Baseline Model — Logistic Regression

In this section, I created a simple classification model to predict whether a person has heart disease/stroke using the available health features.

I selected Logistic Regression as the baseline model for this project.

Logistic Regression is a widely used classification algorithm suitable for binary outcomes, such as predicting the presence (yes) or absence (no) of heart disease/stroke.

It provides interpretable coefficients and is a strong starting point before moving to more complex models.

10.1 Data Preparation

- Target Variable Encoding

  The target column HeartStroke was originally categorical (yes/no).

  It was converted to binary numeric values:

  1 = yes (positive case)

  0 = no (negative case)

10.2 Feature Engineering

Dropped BMI_Category, which was a derived feature from BMI, to avoid redundancy.

One-hot encoding was applied to all categorical variables, with drop_first=True to avoid multicollinearity.

All rows with missing values in either features or the target were removed to ensure clean input for the model.

10.3 Data Splitting

The dataset was split into 80% training and 20% testing subsets using train_test_split().

- Resulting shapes:

  Train set: 2808 rows × 17 features

  Test set: 702 rows × 17 features

  Target distribution (Train set):

  87.4% negative class (no heart disease/stroke)

  12.6% positive class (yes heart disease/stroke)

10.4 Model Training

Logistic Regression

A Logistic Regression model was used as the baseline classifier.

Data was standardized using StandardScaler to improve model convergence and scale features equally.

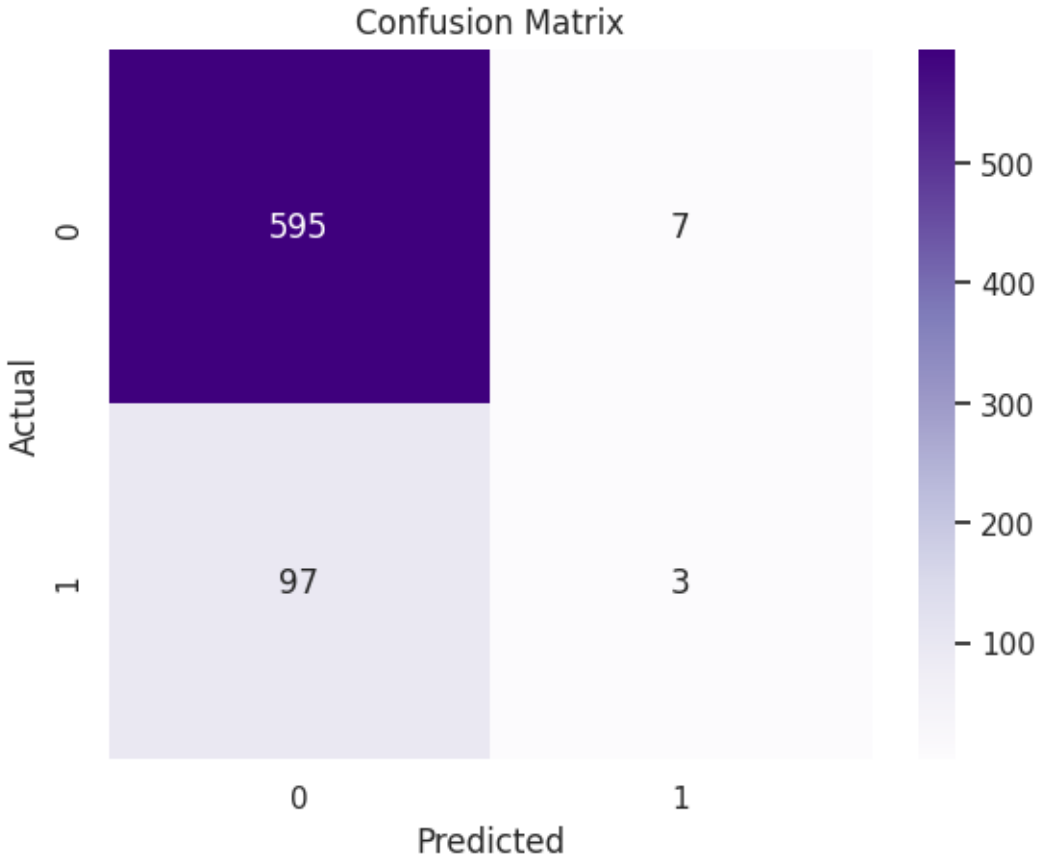
Model hyperparameter:

max_iter=1000 to ensure convergence without warnings.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.99      0.92       602
           1       0.30      0.03      0.05       100

    accuracy                           0.85       702
   macro avg       0.58      0.51      0.49       702
weighted avg       0.78      0.85      0.80       702
```



Feature scaling had a small positive effect on the performance of the Logistic Regression model. Accuracy improved from 85.04% without scaling to 85.19% with scaling. Although the overall improvement is minor, scaling ensures better model convergence and more reliable coefficient estimation, particularly when applying linear models.

11.Conclusion

The baseline Logistic Regression model achieved an accuracy of approximately 85.19% after feature scaling. This means the model correctly predicted heart disease/stroke outcomes for about 85% of the individuals in the test set.

However, I also observed through the classification report that the model struggled to detect positive cases (heart disease/stroke), highlighting the effect of class imbalance on model performance.

Rationale for Metric Choice

Accuracy was chosen as an initial simple and intuitive metric to evaluate overall model performance. While accuracy gives a quick view of model effectiveness, I also reviewed precision, recall, and the confusion matrix to understand performance on each class, especially since detecting heart disease/stroke (positive class) is crucial.

In future steps, improving recall for the positive class will be a focus to better identify high-risk individuals.

Next Steps

- Explore advanced models like Random Forest, XGBoost, and Gradient Boosting to improve recall.
- Apply techniques like SMOTE (Synthetic Minority Over-sampling) to balance the dataset.
- Conduct hyperparameter tuning to optimize model performance.
- Deploy the model in a simple app interface for practical health screening applications.