PROJECT REPORT

Heart Disease Prediction

Eisha Fatima	**345554**
Shahab Ali	**353303**

Contents

1) Introduction and Background:

Today, healthcare faces the ongoing challenge of predicting and addressing diseases globally. Heart disease remains the leading cause of death worldwide. In a world where many have pre-existing conditions and daily choices impact health, understanding one's susceptibility is crucial. Factors like lifestyle choices (smoking, alcohol, sleep), health conditions (diabetes, stroke, asthma), and personal factors (race, age, sex) play a role in predicting heart disease.

Advancements in technology, especially in machine learning, can transform healthcare by enabling faster and more accurate diagnoses. Early detection is essential for treating heart diseases effectively. Unfortunately, in some communities and among women, these conditions often go unnoticed due to differing symptoms. A precise machine learning model can enhance awareness of susceptibility to heart disease.

The model can serve various purposes:

Self-Checking:

Individuals input their attributes into the model to predict their likelihood of heart disease. If the chances are higher, they can make lifestyle changes and schedule routine checkups for early detection.

Aiding Doctors:

Doctors can use these models when reviewing patient histories. If the model suggests a higher risk of heart disease, doctors can provide recommendations and stay vigilant.

This project aims to create a reliable solution. Analyzing the 2020 CDC survey of 400,000 adults, we explore attributes like BMI, smoking, and alcohol consumption correlated with heart disease. We use models, including logistic regression, Naive Bayes, decision trees, linear support vector machines, and neural networks, to predict heart disease susceptibility and determine the best-performing model.

2) Literature Review:

This study builds upon previous efforts in utilizing machine learning for predicting the likelihood of heart failure. A notable earlier research, conducted by Computational Intelligence and Neuroscience, aimed to forecast cardiovascular failure using two distinct datasets. The first dataset, sourced from the Hungarian Institute of Cardiology, consisted of 294 instances, while the second dataset, Statlog (heart), added to the analysis. The initial database included 76 attributes, but through data analysis, the researchers identified 14 key attributes, simplifying the dataset.

To enhance the quality of the data, the researchers performed preprocessing steps, addressing issues like fixing values and reducing noise. Subsequently, they applied various machine learning methods to both datasets, employing REP Tree, M5P, Linear Regression, and Random Tree for the Hungarian dataset, and Naive Bayes, J48, Random Tree, and JRIP for the Statlog (heart) dataset. Performance metrics such as mean absolute error, root mean squared error, accuracy, and execution time were utilized to assess the effectiveness of each model.

The results indicated that Random Tree emerged as the most accurate and efficient model for

both datasets. In the case of the Hungarian dataset, Random Tree achieved the highest accuracy at 99.81% with the second-lowest execution time of 0.02 secs. Although its root mean squared error was greater than other models, its mean absolute error was comparable. Linear Regression demonstrated notable performance on the Hungarian dataset, exhibiting the smallest root mean squared error, execution time, and a moderate mean absolute error. However, it lagged in accuracy compared to other models. For the Statlog (heart) database, Random Tree again excelled with ideal or tied values in accuracy, mean squared error, root mean squared error, and execution time. Naive Bayes also performed well on the Statlog (heart) dataset, showing similar performance to Random Tree.

3) Dataset Description and Exploratory Data Analysis:

For this project, we're using the Personal Key Indicators of Heart Disease Dataset, which is a big survey done by the CDC about the health of around 400,000 adults in the United States. We chose this dataset because it has a lot of everyday health information that's easy to measure. Our goal is to use this dataset because people can easily fill out a health form with this kind of information, making it quick to get results from any model we create.

The link for the dataset is as follows:

https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

3.1 Dataset Attribute Information:

In our dataset, Heart Disease is the main thing we want to predict – it's like a Yes or No to whether someone has coronary heart disease or a heart attack. The next few columns have different kinds of information:

BMI (**Body Mass Index**): This is a number that shows if someone is underweight, normal weight, overweight, or obese. It's calculated using weight and height.

Physical Health: This number tells us how many days out of the past 30 someone has been physically sick or injured.

Mental Health: This number shows how many days out of the past 30 someone has had bad mental health.

Sleep Time: This is a number of hours someone usually sleeps in a day.

Smoking: It's a Yes or No about whether someone has smoked a lot in their life.

Alcohol Drinking: It's a Yes or No about whether someone drinks a lot of alcohol.

Stroke: It's a Yes or No about whether someone has had a stroke.

Difficulty Walking: It's a Yes or No about whether someone has trouble walking.

Sex: This just tells us if someone is male or female.

Physical Activity: It's a Yes or No about whether someone has done exercise in the past 30 days.

Asthma: It's a Yes or No about whether someone has asthma.

Kidney Disease: It's a Yes or No about whether someone has kidney disease.

Skin Cancer: It's a Yes or No about whether someone has skin cancer.

Age Category: This puts people into groups based on their age.

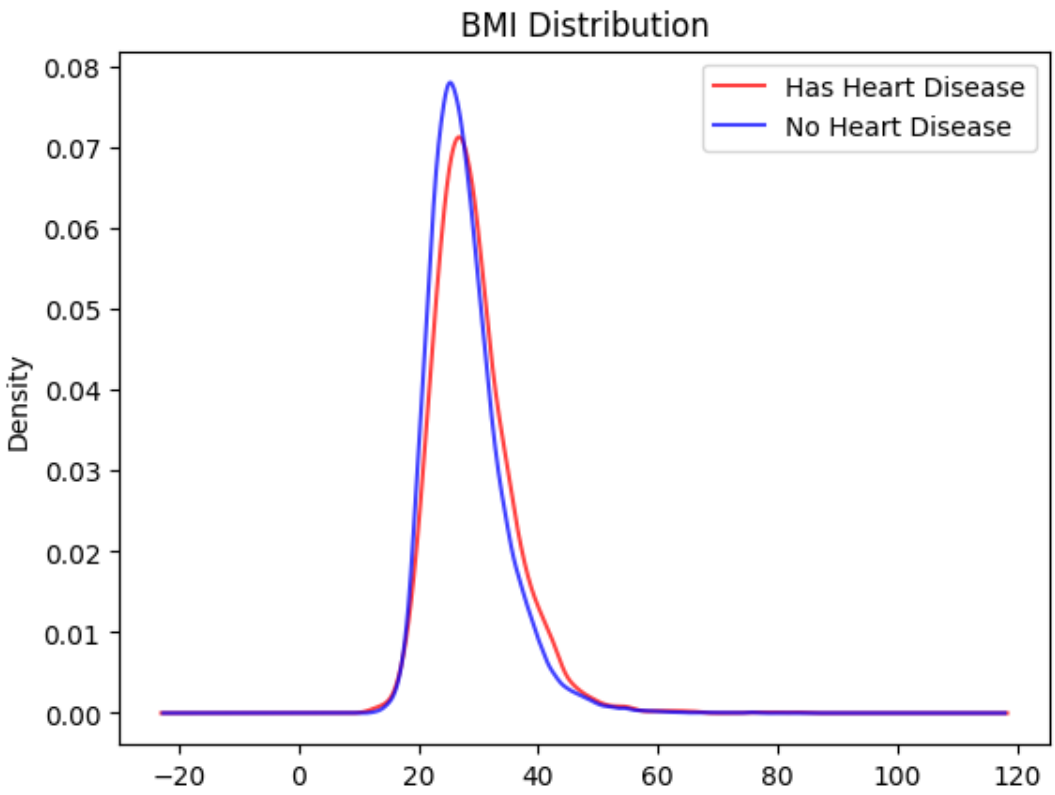
Race: This tells us the race or ethnicity of a person.

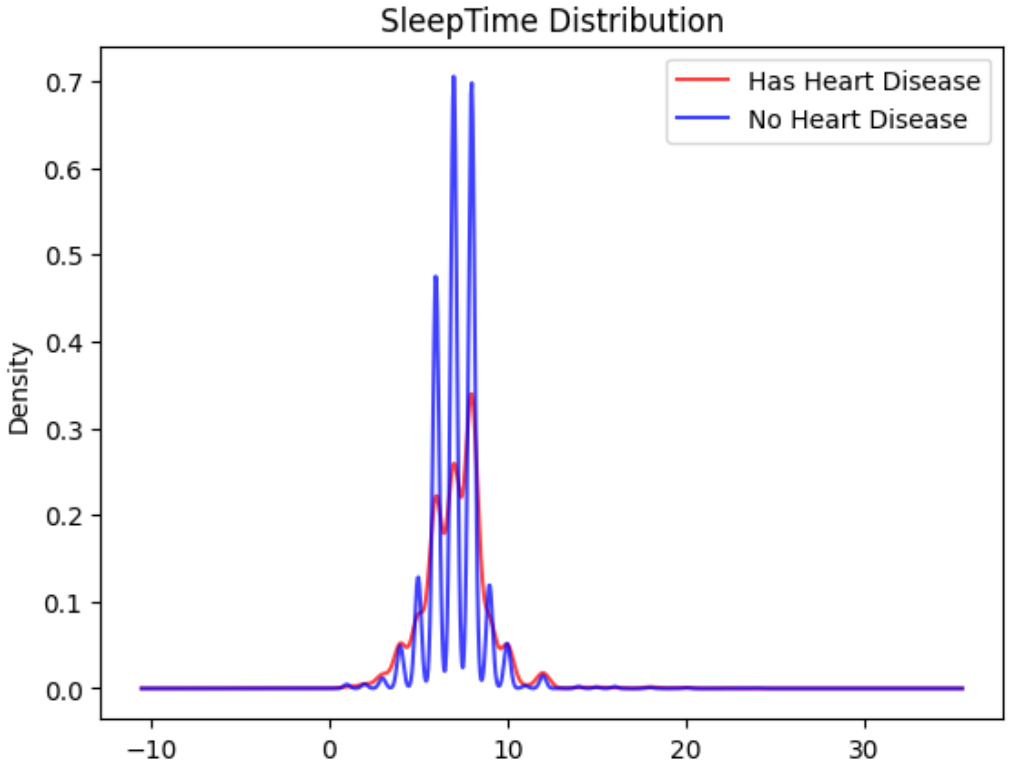
Diabetic: It's a Yes or No about whether someone has diabetes, including during pregnancy.

General Health: This is about how people rate their overall health – Excellent, Very good, Good, Fair, or Poor.

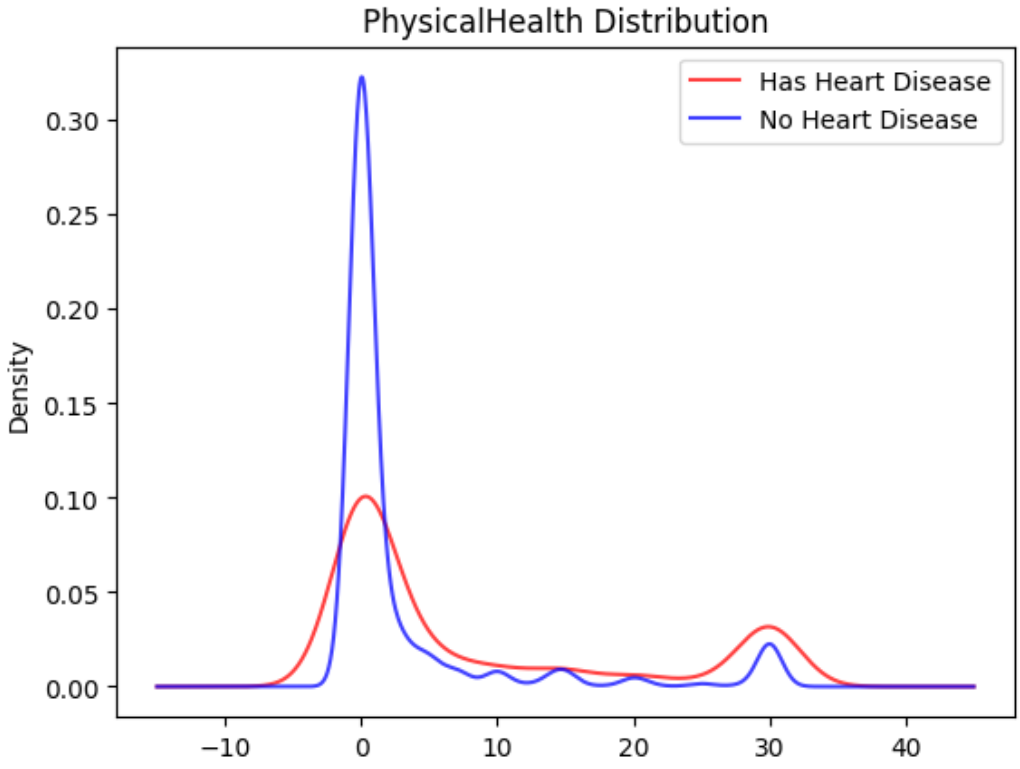
3.2 Exploratory Data Analysis:

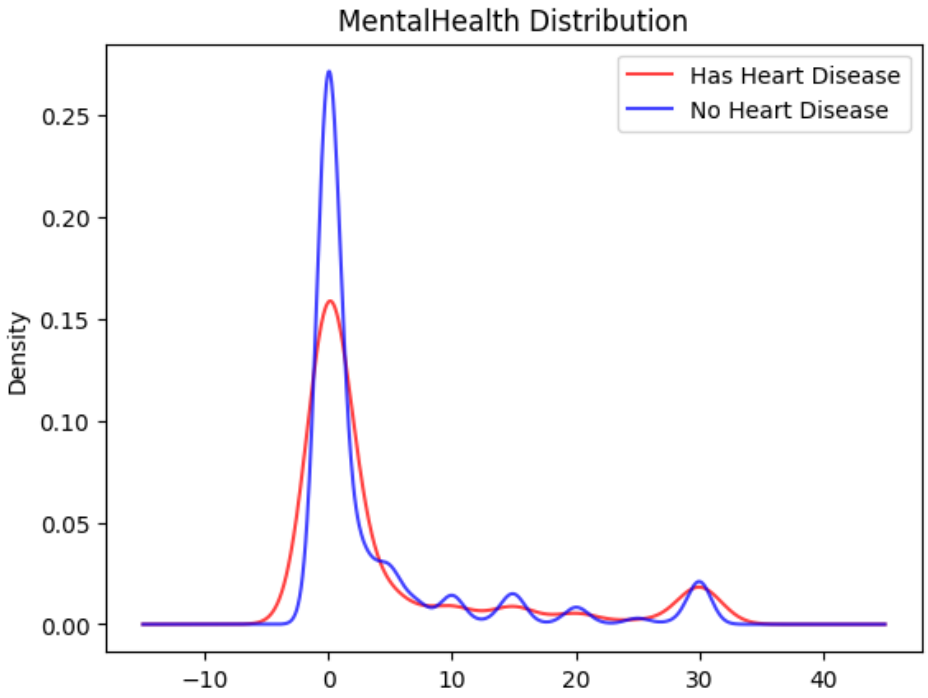
We visualized numerical information using kernel density estimation curves and categorical information using histograms. We did this separately for individuals with and without heart disease. This helped us understand how the attributes are distributed among these two groups.


BMI Distribution
Has Heart Disease
No Heart Disease
Density
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−20
0
20
40
60
80
100
120


SleepTime Distribution
Has Heart Disease
No Heart Disease
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−10
0
10
20
30

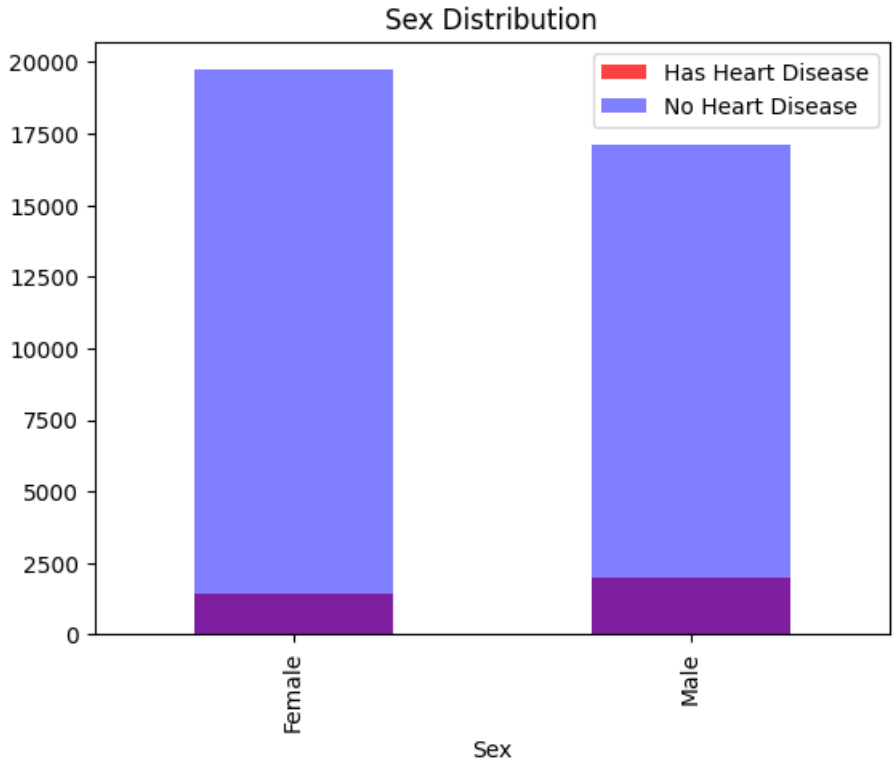
Concerning numerical variables, '**BMI**' and '**Sleep Time**' exhibit a fairly normal distribution. When we compare individuals with and without heart disease based on BMI, there is minimal disparity between the two groups. This implies that BMI might not strongly influence heart disease. A similar observation applies to sleep time, as the distribution of sleep time for individuals with and without heart disease follows a normal pattern, indicating that sleep time may not have a significant impact on heart disease.



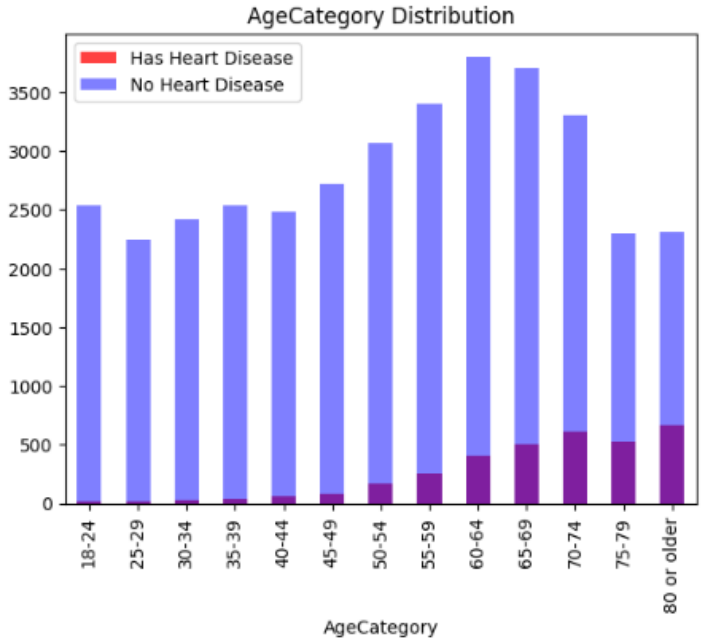


In contrast, the distributions of '**Physical Health**' and '**Mental Health**' appear more pronounced, particularly around the minimum (0 days) and maximum (30 days), with a focus on the minimum (indicating better health). This suggests that the majority of individuals tended to be on the healthier end of the spectrum.

Upon comparing individuals with and without heart disease, a clear pattern emerges. Those on the maximum side of the scale tend to experience heart disease more frequently, while those on the minimum side experience heart disease less. Consequently, we can infer that both mental health and physical health play crucial roles in determining the likelihood of someone developing heart disease.



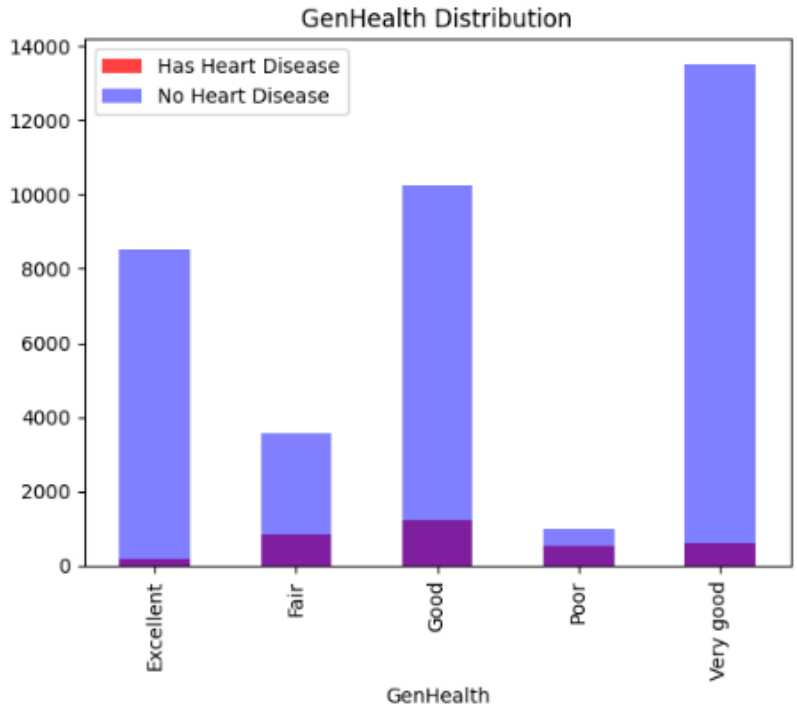

```
Probability of Heart Disease if you put Female in Sex: 6.68%
Probability of Heart Disease if you put Male in Sex: 10.32%
```




```
Probability of Heart Disease if you put 18-24 in AgeCategory: 0.70%
Probability of Heart Disease if you put 25-29 in AgeCategory: 0.53%
Probability of Heart Disease if you put 30-34 in AgeCategory: 1.14%
Probability of Heart Disease if you put 35-39 in AgeCategory: 1.44%
Probability of Heart Disease if you put 40-44 in AgeCategory: 2.47%
Probability of Heart Disease if you put 45-49 in AgeCategory: 3.06%
Probability of Heart Disease if you put 50-54 in AgeCategory: 5.08%
Probability of Heart Disease if you put 55-59 in AgeCategory: 6.91%
Probability of Heart Disease if you put 60-64 in AgeCategory: 9.74%
Probability of Heart Disease if you put 65-69 in AgeCategory: 12.04%
Probability of Heart Disease if you put 70-74 in AgeCategory: 15.53%
Probability of Heart Disease if you put 75-79 in AgeCategory: 18.76%
Probability of Heart Disease if you put 80 or older in AgeCategory: 22.37%
```

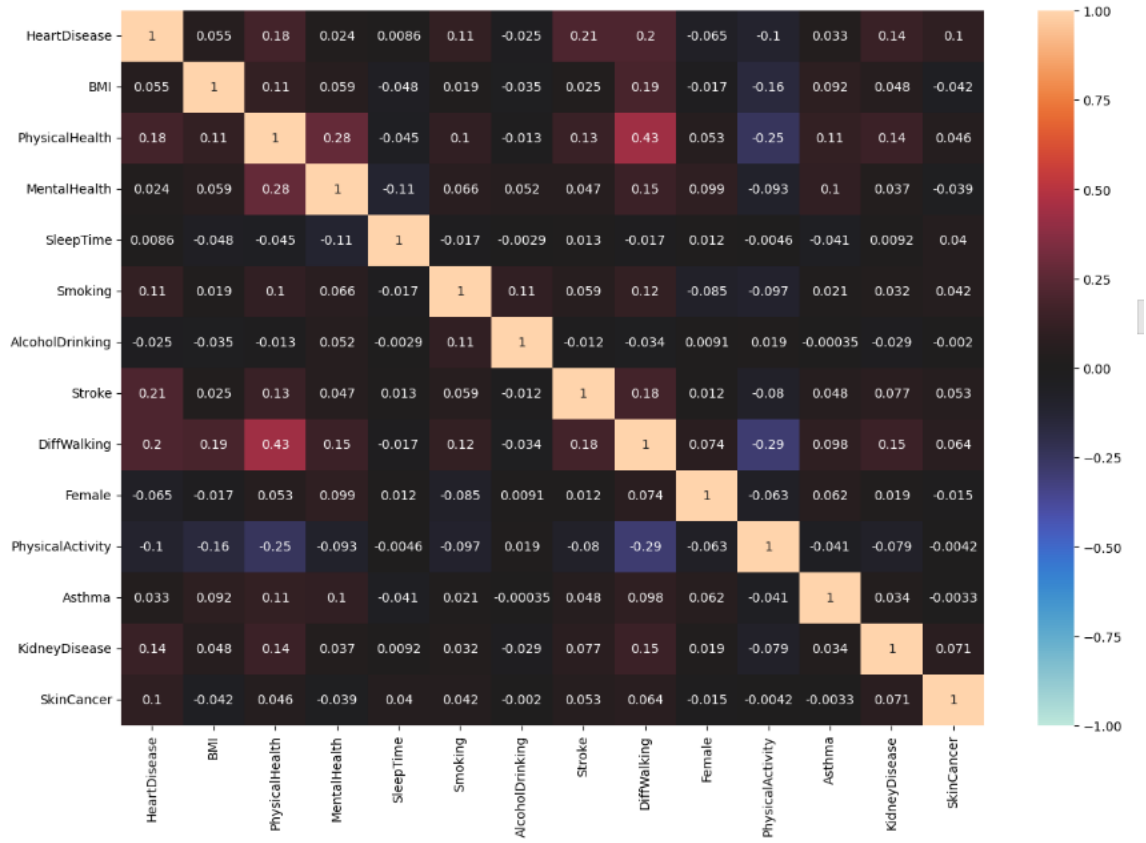
The data showed an almost equal number of males and females in the 'Sex' variable. After plotting the data and comparing 'Yes' and 'No' responses, it became clear that males are more likely to have heart disease than females. Similar patterns emerged in the 'Age Category' variable, where the age distribution of the population appeared mostly normal.

However, an interesting finding is that heart disease seems to be more prevalent among older individuals (50 years and above, constituting 5% or more) compared to younger individuals (below 49 years old).




```
Probability of Heart Disease if you put Excellent in GenHealth: 2.11%
Probability of Heart Disease if you put Fair in GenHealth: 19.06%
Probability of Heart Disease if you put Good in GenHealth: 10.71%
Probability of Heart Disease if you put Poor in GenHealth: 34.75%
Probability of Heart Disease if you put Very good in GenHealth: 4.31%
```

Looking at the distribution of 'General Health' ('GenHealth'), individuals classified as having 'Poor' health show a substantially higher likelihood of getting heart disease (34.10%) compared to those categorized as having 'Excellent' health (2.24%). This indicates a correlation between an individual's general health and their likelihood of developing heart disease.



We generated a covariance matrix encompassing all binary and numerical attributes. According to this matrix, Heart Disease shows a positive correlation with Physical Health, Stroke, Kidney Disease, Smoking, and Difficulty Walking. Regarding correlations among other independent variables, Physical Health, Physical Activity, and Difficulty Walking exhibit notable correlations with several other variables such as Stroke, BMI, and Mental Health.

Fortunately, we did not encounter data bias issues as there were no missing values (NA values). Moreover, the data appeared coherent without any instances of nonsensical information.

4) Data Preprocessing:

Exploring Unique Values and Error Checking in Dataset

```
# Analyze unique values and check for any errors
print(df.apply(lambda col: col.unique()))
```

```
HeartDisease                                                [No, Yes]
BMI                 [16.6, 20.34, 26.58, 24.21, 23.71, 28.87, 21.6...
PhysicalHealth      [3.0, 0.0, 20.0, 28.0, 6.0, 15.0, 5.0, 30.0, 7...
MentalHealth        [30.0, 0.0, 2.0, 5.0, 15.0, 8.0, 4.0, 3.0, 10....
SleepTime           [5.0, 7.0, 8.0, 6.0, 12.0, 4.0, 9.0, 10.0, 15....
Smoking                                                [Yes, No, nan]
AlcoholDrinking                                        [No, Yes, nan]
Stroke                                                 [No, Yes, nan]
DiffWalking                                            [No, Yes, nan]
Sex                                                [Female, Male, nan]
PhysicalActivity                                       [Yes, No, nan]
Asthma                                                 [Yes, No, nan]
KidneyDisease                                          [No, Yes, nan]
SkinCancer                                             [Yes, No, nan]
AgeCategory         [55-59, 80 or older, 65-69, 75-79, 40-44, 70-7...
Race                [White, Black, Asian, American Indian/Alaskan ...
Diabetic            [Yes, No, No, borderline diabetes, Yes (during...
GenHealth               [Very good, Fair, Good, Poor, Excellent, nan]
dtype: object
```

This code uses the **apply** function on the DataFrame **df** to iterate over each column and prints the unique values for each column. It helps in identifying any unexpected or erroneous values in the dataset. The output will display the unique values present in each column, allowing for a manual check to ensure data integrity and correctness.

Check for nan values

```
# Check for any None or nan values
df.isin([None, float('nan')]).any(axis=0)
```

```
HeartDisease        False
BMI                 False
PhysicalHealth       True
MentalHealth         True
SleepTime            True
Smoking              True
AlcoholDrinking      True
Stroke               True
DiffWalking          True
Sex                  True
PhysicalActivity     True
Asthma               True
KidneyDisease        True
SkinCancer           True
AgeCategory          True
Race                 True
Diabetic             True
GenHealth            True
dtype: bool
```

Replacing Binary Columns (No/Yes) with 0 and 1:

- Replaces 'No' with 0 and 'Yes' with 1 in specified columns, converting them into binary representations.

```
# Replace Binary Columns No with 0
df.iloc[:,5:14] = df.iloc[:,5:14].replace('No', 0)
df['HeartDisease'] = df['HeartDisease'].replace('No', 0)

# Replace Binary Columns Yes with 1
df.iloc[:,5:14] = df.iloc[:,5:14].replace('Yes', 1)
df['HeartDisease'] = df['HeartDisease'].replace('Yes', 1)
```

Column Renaming and Typo Corrections:

- Renames the 'Sex' column to 'Female'.
- Replaces 'Male' with 0 and 'Female' with 1 in the 'Female' column.
- Makes some typo corrections in the 'Diabetic', 'Race', and 'GenHealth' column

```python
# Replace Male with 0, Female with 1, and some typo corrections
df = df.replace({
    'Female': {
        'Female': 1,
        'Male': 0
    },
    'Diabetic': {
        'No, borderline diabetes': 'No_borderline_diabetes',
        'Yes (during pregnancy)': 'Yes_during_pregnancy'
    },
    'Race': {
        'American Indian/Alaskan Native': 'American_Indian_or_Alaskan_Native'
    },
    'GenHealth': {
        'Very good': 'Very_good'
    }

})

df.head()
```

Min-Max Normalization on Numerical Attributes:

- Applies min-max normalization on numerical attributes (columns 1 to 4) using sklearn's MinMaxScaler.

```python
# min-max normalization on numerical attributes
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df.iloc[:, 1:5] = scaler.fit_transform(df.iloc[:, 1:5])

df.head()
```

	HeartDisease	BMI	PhysicalHealth	MentalHealth	SleepTime	Smoking	AlcoholDrinking	Stroke	DiffWalking
0	0	0.059490	0.100000	1.0	0.173913	1	0	0	0
1	0	0.112465	0.000000	0.0	0.260870	0	0	1	0
2	0	0.200850	0.666667	1.0	0.304348	1	0	0	0
3	0	0.167280	0.000000	0.0	0.217391	0	0	0	0
4	0	0.160198	0.933333	0.0	0.304348	0	0	0	1

One-Hot Encoding:

- Performs one-hot encoding on categorical columns ('AgeCategory', 'Race', 'Diabetic', 'GenHealth').

```
# Additional Preprocessing for Logistic Regression, SVM, Neural Network, Decision Trees
# One Hot Encoding

df_ohe = df.copy()
for col in ['AgeCategory', 'Race', 'Diabetic', 'GenHealth']:
    df_ohe = pd.concat([df_ohe, pd.get_dummies(df_ohe[col], prefix=f'{col}_')],axis=1).drop([

df_ohe.head()
```

	HeartDisease	BMI	PhysicalHealth	MentalHealth	SleepTime	Smoking	AlcoholDrinking
0	0	0.059490	0.100000	1.0	0.173913	1	0
1	0	0.112465	0.000000	0.0	0.260870	0	0
2	0	0.200850	0.666667	1.0	0.304348	1	0
3	0	0.167280	0.000000	0.0	0.217391	0	0
4	0	0.160198	0.933333	0.0	0.304348	0	0

5 rows × 42 columns

5) Proposed Methodology:

Our approach begins with data pre-processing to ensure a clean and ready dataset. This involves checking for mislabeled attributes, eliminating data with missing values, and converting binary categorical data to numerical values (0s and 1s). We perform min-max normalization on numerical attributes, scaling them between 0 and 1 to avoid bias. Categorical attributes undergo one-hot encoding for ease of integration into our models, with the Naive Bayes model utilizing ordinal encoding on a separate dataset. The dataset is then divided into 90% for training and 10% for testing, allowing unbiased model performance assessment.

Various models are employed to determine the most effective one. Logistic regression, a common binary classification model, utilizes a linear combination of inputs, weights, and a sigmoid activation function. Decision tree classifiers use a tree-like structure for data inferences, while linear support vector machines (SVM) use hyperplanes to separate data. Naive Bayes models, including both Categorical and Gaussian versions, are applied to different datasets based on attribute types. All models leverage the scikit-learn library in Python.

The last model evaluated is Neural Network. A baseline neural network is initially created with two hidden layers, 32 nodes each, using Relu activation functions, and a final output layer of 1 node with a sigmoid activation function.

Evaluation metrics, including overall accuracy, precision, recall, and f1-score, along with a confusion matrix, aid in determining the most suitable model for demonstrations.

6) Experimental Results:

6.1) Model Performance on Testing Set:

Model	Accuracy	Precision (No Heart Disease)	Precision (Yes Heart Disease)	Recall (No Heart Disease)	Recall (Yes Heart Disease)	F1-Score (No Heart Disease)	F1-Score (Yes Heart Disease)
Logistic Regression	0.91	0.92	0.52	0.99	0.10	0.96	0.17
Decision Tree	0.86	0.93	0.23	0.92	0.25	0.92	0.24
Linear SVM	0.91	0.91	0	1	0	0.96	0
Categorical Naive Bayes	0.89	0.94	0.35	0.93	0.37	0.94	0.36
Gaussian Naive Bayes	0.87	0.93	0.22	0.93	0.21	0.93	0.22
Baseline Neural Network	0.91	0.93	0.5	0.98	0.16	0.95	0.24

6.2) Confusion Matrix of Each Model on Test Set

Model	True Positives	True Negatives	False Positives	False Negatives
Logistic Regression	2.8e+02	2.9e+04	2.5e+02	2.5e+03
Decision Tree	6.9e+02	2.7e+04	2.3e+03	2.1e+03
Linear SVM	0	2.9e+04	0	2.8e+03
Categorical Naive Bayes	1e+03	2.7e+04	2e+03	1.7e+03
Numerical Naive Bayes	5.7e+02	2.7e+04	2e+03	2.2e+03
Baseline Neural Network	4.4e+02	2.9e+04	4.5e+02	2.3e+03

6.3) Logistic Regression:

The Logistic Regression model shows commendable performance when compared to other non-Neural Network models, boasting the highest accuracy. However, its precision, recall, and f1-score specifically for individuals with heart disease are notably lower than those for individuals without heart disease. This implies that the model tends to misclassify individuals with heart disease but demonstrates accuracy in identifying those without heart disease. Notably, Logistic Regression exhibits a low recall for individuals with heart disease, registering at just 10%, indicating its recognition of only a small portion of individuals with heart disease.

6.4) Decision Tree

The Decision Tree model exhibits the lowest overall accuracy among all models. However, its f1 score and recall for individuals with heart disease surpass those of Logistic Regression. This suggests that the Decision Tree model is more inclined to classify a person as having heart disease compared to logistic regression. Nonetheless, this inclination reduces its precision, resulting in the decision tree model having the highest number of false positives among all other models. The inaccuracies in the decision tree model are likely attributed to overfitting, a common issue with Decision Trees.

6.5) Linear SVM

Despite achieving higher overall accuracy than the Decision Tree model, the Linear SVM emerges as the worst-performing model by a significant margin. This is attributed to the fact that the Linear SVM recorded zero true positives and zero false positives, indicating that it classified every person as not having heart disease. Given that 91.4% of individuals in the dataset do not have heart disease, the accuracy is inherently high. However, the model proves ineffective as a classifier. We posit that the failure of the Linear SVM stems from its reliance on a single hyperplane, which proves inadequate for modeling the data adequately.

6.6) Categorical Naive Bayes

While the overall accuracy of Categorical Naive Bayes doesn't reach remarkable levels, it excels in terms of f1-score and recall for individuals with heart disease, surpassing all other models. The precision for detecting heart disease is also relatively high, accompanied by the highest count of true positives. This implies that the model tends to classify more individuals as having heart disease while maintaining decent precision. However, it is essential to note that the number of false positives remains considerably large compared to the Neural Network and Logistic Regression models.5.7) Numerical Naive Bayes

6.7) Numerical Naive Bayes

Numerical Naive Bayes demonstrates outcomes akin to the Decision Tree model in general. While its accuracy is low, there are slight enhancements in recall and f1-score for individuals with heart disease compared to logistic regression. This subpar performance is likely attributed to the model's insufficient data, given the presence of only four numerical attributes.

6.8) Baseline Neural Network

The baseline Neural Network exhibits accuracy similar to Logistic Regression but boasts higher

recall and f1-score for individuals with heart disease. While its precision remains comparable to logistic regression, the model registers almost twice the number of false positives. This indicates that the model identifies more individuals with heart disease but comes with a higher risk of false positives.

7) Conclusion and Discussion:

Heart diseases are a major cause of global deaths, influenced by various factors like lifestyle choices (smoking, alcohol, sleep), health conditions (diabetes, strokes), and personal factors (race, age, sex). This report used machine learning models to understand these influences, including logistic regression, decision tree, linear SVM, Naive Bayes, and Neural Networks.

We measured model effectiveness using accuracy, precision, and recall. The optimized Neural Network performed best, balancing true positives and false positives. While it has a low false positive rate, it tends to predict fewer cases of heart disease (higher false negatives), taking a cautious approach.

To improve these models, we can add more factors indicating heart disease or ensure a better balance of people with heart disease in the dataset. Even though there's room for improvement, a model predicting heart disease has practical uses, potentially saving lives through preventive healthcare. Further refinements could open up more applications for this machine learning model.