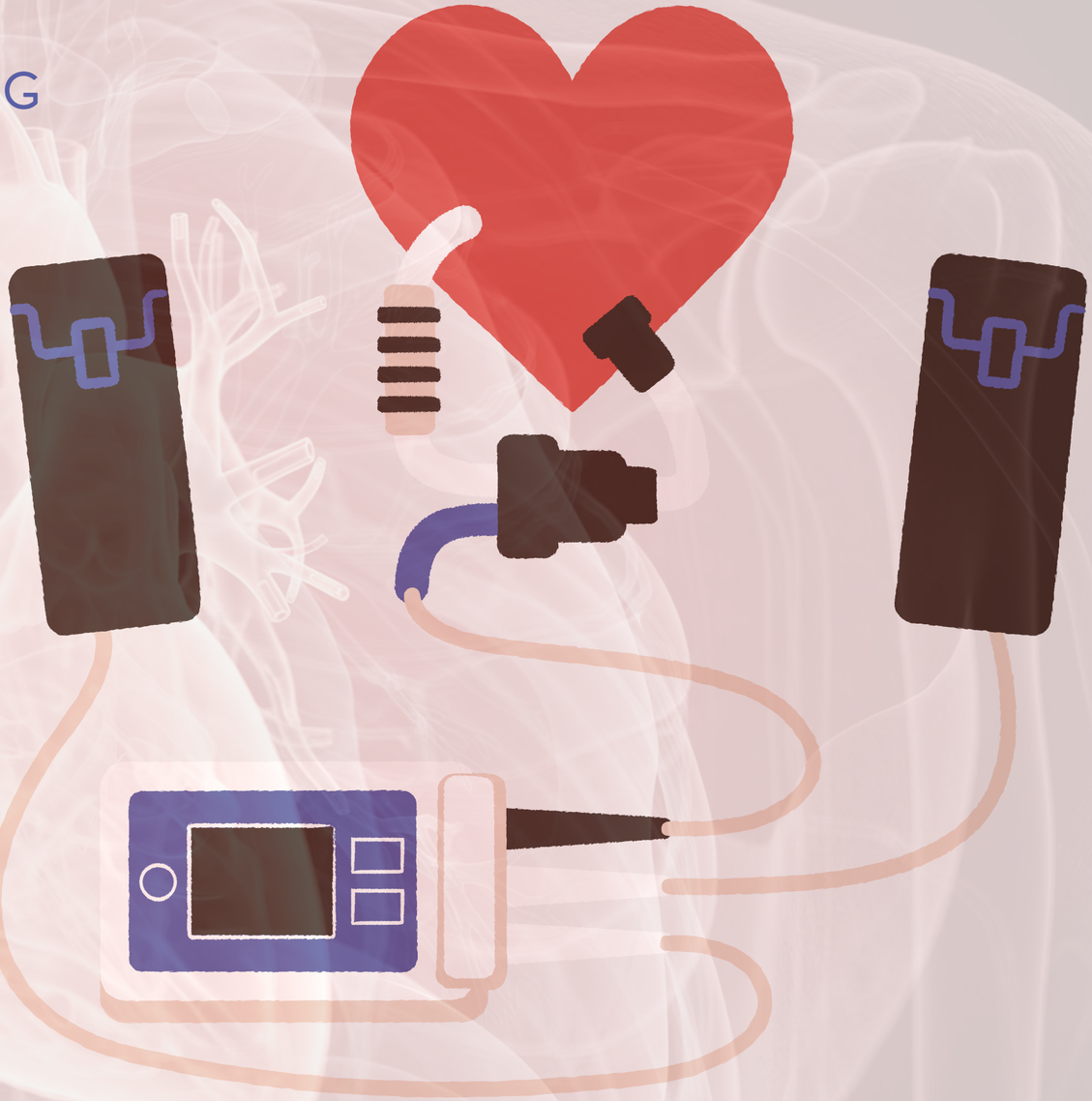
WIA1006/WID3006 MACHINE LEARNING

Heart Diseases

Group Assignment | Semester 2, 22/23

HeartGuardians

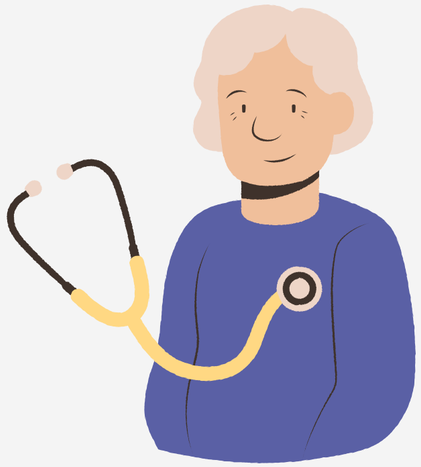
Chew Jia Hui
Teoh Zhi Yee
Carmen Lam Kah Man
Gan Zi Xiang
Ter Zhen Huang

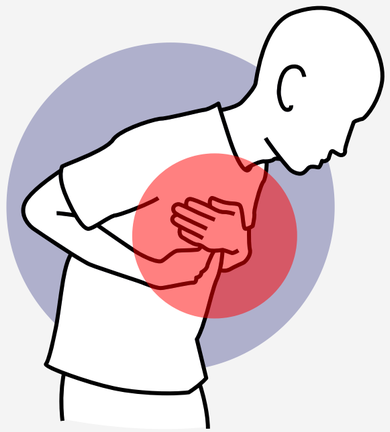


Contents

- Relevance & Significance of Problem
- Data Cleaning & Preprocessing
- Exploratory Data Analysis (EDA)
- Model Selection & Performance
- Presentation of Project Deliverable

How is heart disease prediction a relevant and significant problem?





Cause 1

Importance and Applicability

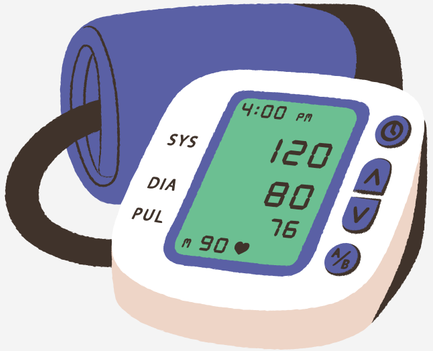
Cause 2

Practical value and Impact

Cause 3

Using machine learning models and algorithms

Relevance & Significance of the Problem



Problem Identification

Predict the likelihood of an individual experiencing a heart disease

Real-world Applicability

Heart disease is a significant health issue locally and globally (DOSM, 2021)

Practical Value & Impact

Address a crucial and critical health concern with potential life-saving implications

Stakeholder Perspective

Spark interests among patients and individuals with high health risks, healthcare providers and policymakers

DATA CLEANING

1. Import Libraries

2. Load Data

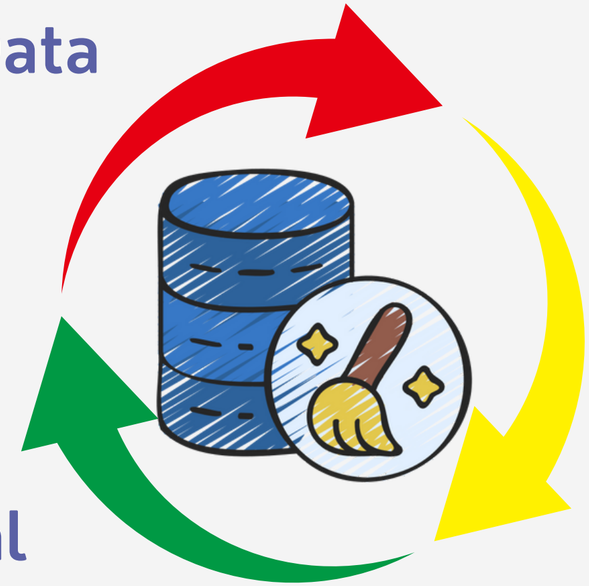
3. Explore Data

4. Handle Missing Values

5. Categorical Data Encoding

6. Handle Outliers

7. Standardize or Normalize Data



IMPORT LIBRARIES

Necessary libraries

- **pandas**
- **numpy**

Model Development

- scikit-learn
- TensorFlow

Data Vizualisation

- matplotlib
- seaborn

Performance Evaluation

- scikit-learn's metrics

Data Preprocessing and Feature Engineering

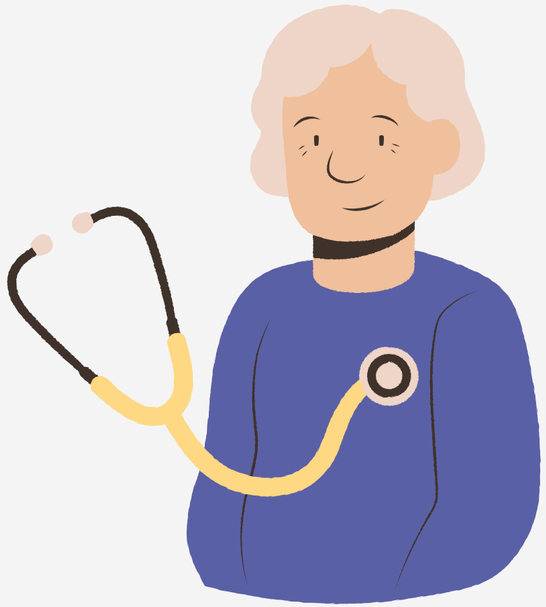
- OneHotEncoder
- MinMaxScaler
- StandardScaler

Results Visualization

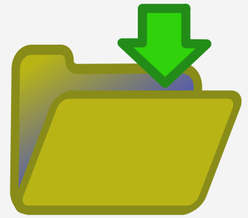
- tabulate

Normalization

- scipy



LOAD DATA



read_csv()

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	SkinCancer
0	No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Yes	Very good	5.0	Yes	No	Yes
1	No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	No	Yes	Very good	7.0	No	No	No
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Yes	Fair	8.0	Yes	No	No
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	No	Good	6.0	No	No	Yes
4	No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	No	Yes	Very good	8.0	No	No	No
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
319790	Yes	27.41	Yes	No	No	7.0	0.0	Yes	Male	60-64	Hispanic	Yes	No	Fair	6.0	Yes	No	No
319791	No	29.84	Yes	No	No	0.0	0.0	No	Male	35-39	Hispanic	No	Yes	Very good	5.0	Yes	No	No
319792	No	24.24	No	No	No	0.0	0.0	No	Female	45-49	Hispanic	No	Yes	Good	6.0	No	No	No
319793	No	32.81	No	No	No	0.0	0.0	No	Female	25-29	Hispanic	No	No	Good	12.0	No	No	No
319794	No	46.56	No	No	No	0.0	0.0	No	Female	80 or older	Hispanic	No	Yes	Good	8.0	No	No	No

319795 rows × 18 columns

EXPLORE DATA

shape()

```
(319795, 18)
```

head()

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex
0	No	16.60	Yes	No	No	3.0	30.0	No	Female
1	No	20.34	No	No	Yes	0.0	0.0	No	Female
2	No	26.58	Yes	No	No	20.0	30.0	No	Male
3	No	24.21	No	No	No	0.0	0.0	No	Female
4	No	23.71	No	No	No	28.0	0.0	Yes	Female

describe()

	BMI	PhysicalHealth	MentalHealth	SleepTime
count	319795.00	319795.00	319795.00	319795.00
mean	28.33	3.37	3.90	7.10
std	6.36	7.95	7.96	1.44
min	12.02	0.00	0.00	1.00
25%	24.03	0.00	0.00	6.00
50%	27.34	0.00	0.00	7.00
75%	31.42	2.00	3.00	8.00
max	94.85	30.00	30.00	24.00

info()

```
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   HeartDisease      319795 non-null  object
 1   BMI               319795 non-null  float64
 2   Smoking           319795 non-null  object
 3   AlcoholDrinking   319795 non-null  object
 4   Stroke            319795 non-null  object
 5   PhysicalHealth    319795 non-null  float64
 6   MentalHealth      319795 non-null  float64
 7   DiffWalking       319795 non-null  object
```

HANDLE MISSING VALUES

isnull().sum()

```
HeartDisease          0
BMI                   0
Smoking               0
AlcoholDrinking       0
Stroke                0
PhysicalHealth        0
MentalHealth          0
DiffWalking           0
Sex                   0
AgeCategory           0
Race                  0
Diabetic              0
PhysicalActivity      0
GenHealth             0
SleepTime             0
Asthma                0
KidneyDisease         0
SkinCancer            0
dtype: int64
```

CATEGORICAL DATA ENCODING

ORDINAL ENCODING

AgeCategory

18-24	0
25-29	1
30-24	2
35-39	3
40-44	4
45-49	5
50-54	6
55-59	7
60-64	8
65-69	9
70-74	10
75-79	11
80 or older	12

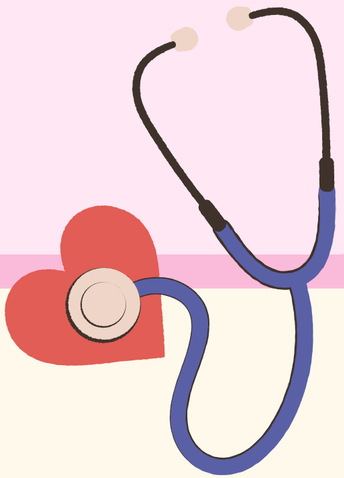
GenHealth

Poor	0
Fair	1
Good	2
Very good	3
Excellent	4

ONE-HOT ENCODING

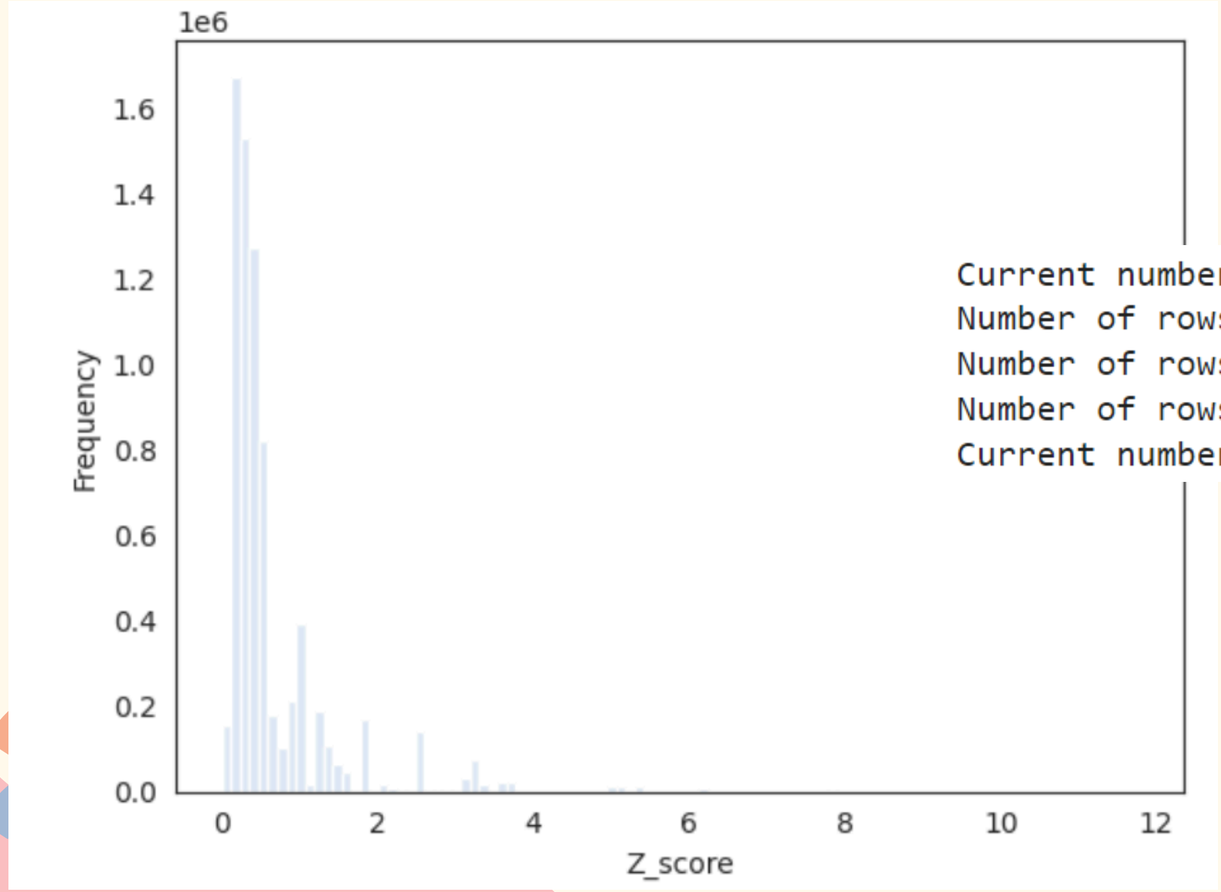
Race

- Race_American Indian/Alaskan Native
- Race_Asian
- Race_Black
- Race_Hispanic
- Race_Other
- Race_White



OUTLIERS REMOVING

Z-score method



min_threshold = 3.5
max_threshold = 4

```
Current number of rows : 319795
Number of rows after removing outliers using 2.5 std dev: 232558
Number of rows after removing outliers using 3 std dev: 272628
Number of rows that are within std dev 2.5 to 3: 40070
Current number of rows : 232558
```

Data points with Z-scores below 3.5 or above 4 are considered outliers.

NORMALIZATION OF DATA

StandardScaler()

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	...	SleepTime
0	0	-1.975467	1	0	0	0.010829	3.504975	-0.360947	0	0.123977	...	-1.720285
2	0	-0.250046	1	0	0	2.318714	3.504975	-0.360947	1	0.685718	...	0.711314
3	0	-0.659790	0	0	0	-0.396444	-0.471701	-0.360947	0	1.247458	...	-0.909752
4	0	-0.746234	0	0	0	3.404777	-0.471701	2.770486	0	-0.718633	...	0.711314
6	0	-1.105841	0	0	0	1.639924	-0.471701	-0.360947	0	0.966588	...	-2.530818
7	0	0.624767	1	0	0	0.282345	-0.471701	2.770486	0	1.528328	...	1.521847
9	0	2.189403	0	0	0	-0.396444	-0.471701	2.770486	1	0.685718	...	2.332379
11	0	0.118206	1	0	0	-0.396444	-0.471701	-0.360947	0	0.123977	...	-1.720285
12	0	0.059424	1	0	0	-0.396444	-0.471701	2.770486	1	1.247458	...	0.711314
13	0	0.021388	0	0	0	0.553861	-0.471701	2.770486	0	1.528328	...	-0.099219

EXPLORATORY DATA ANALYSIS (EDA)

13 Categorical Variables

- Smoking
- Alcohol drinking
- Stroke
- Difficult Walking
- Sex
- Age Category
- Race

- Diabetic
- Physical Activity
- General Health
- Asthma
- Kidney Disease
- Skin Cancer

4 Numerical Variables

- BMI
- Physical Health
- Mental Health
- Sleep Time

SUMMARY STATISTICS

	HeartDisease	BMI	Smoking	AlcoholDrinking	\
count	319795.000000	319795.000000	319795.000000	319795.000000	
mean	0.085595	28.325399	0.412477	0.068097	
std	0.279766	6.356100	0.492281	0.251912	
min	0.000000	12.020000	0.000000	0.000000	
25%	0.000000	24.030000	0.000000	0.000000	
50%	0.000000	27.340000	0.000000	0.000000	
75%	0.000000	31.420000	1.000000	0.000000	
max	1.000000	94.850000	1.000000	1.000000	

	Stroke	PhysicalHealth	MentalHealth	DiffWalking	\
count	319795.000000	319795.00000	319795.000000	319795.000000	
mean	0.037740	3.37171	3.898366	0.138870	
std	0.190567	7.95085	7.955235	0.345812	
min	0.000000	0.00000	0.000000	0.000000	
25%	0.000000	0.00000	0.000000	0.000000	
50%	0.000000	0.00000	0.000000	0.000000	
75%	0.000000	2.00000	3.000000	0.000000	
max	1.000000	30.00000	30.000000	1.000000	

	Sex	AgeCategory	...	SleepTime	Asthma	\
count	319795.000000	319795.000000	...	319795.000000	319795.000000	
mean	0.475273	6.514536	...	7.097075	0.134061	
std	0.499389	3.564759	...	1.436007	0.340718	
min	0.000000	0.000000	...	1.000000	0.000000	
25%	0.000000	4.000000	...	6.000000	0.000000	
50%	0.000000	7.000000	...	7.000000	0.000000	
75%	1.000000	9.000000	...	8.000000	0.000000	
max	1.000000	12.000000	...	24.000000	1.000000	

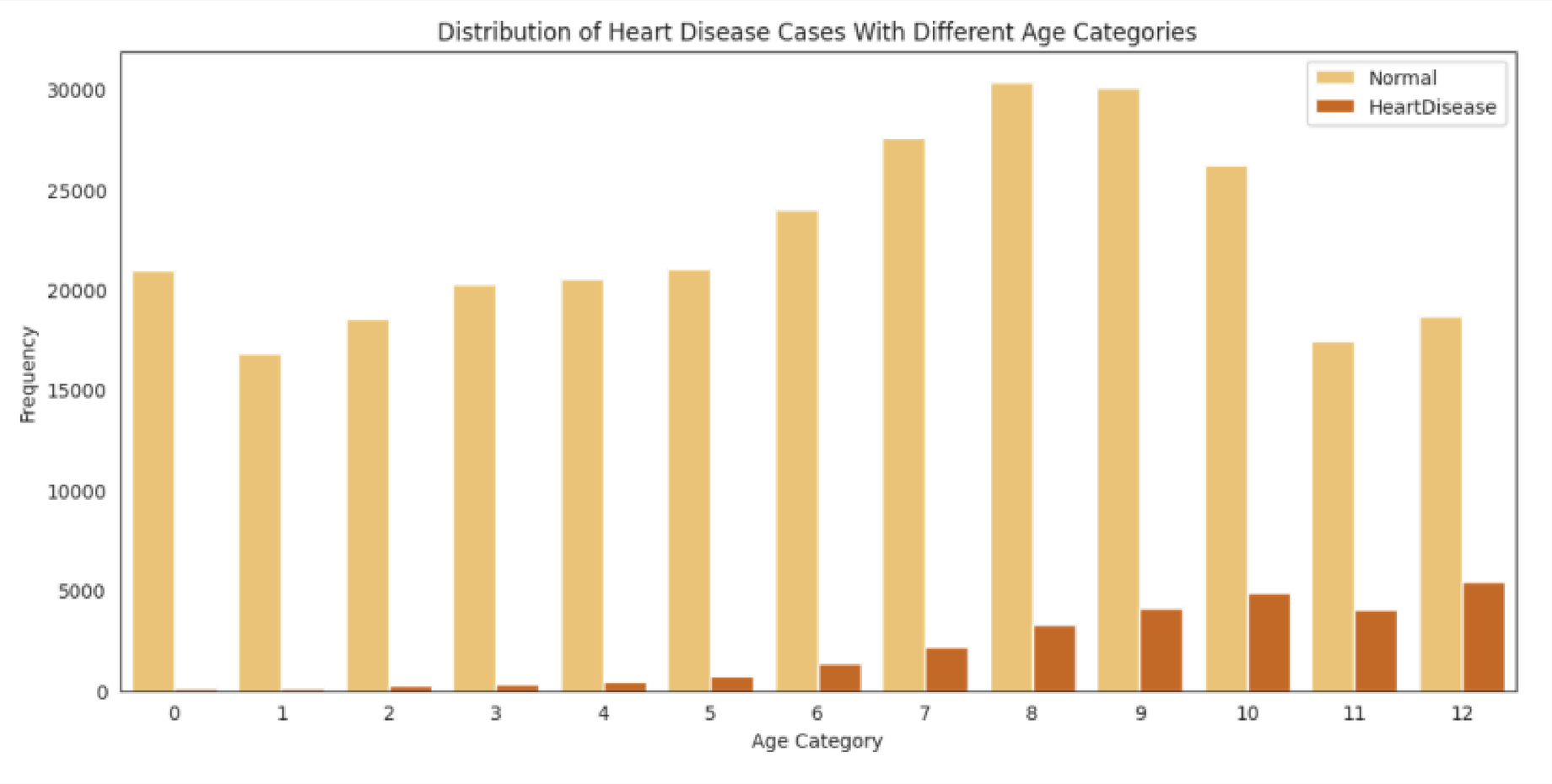
	KidneyDisease	SkinCancer	Race_American Indian/Alaskan Native	\
count	319795.000000	319795.000000	319795.000000	
mean	0.036833	0.093244	0.016267	
std	0.188352	0.290775	0.126499	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	

	Race_Asian	Race_Black	Race_Hispanic	Race_Other	\
count	319795.000000	319795.000000	319795.000000	319795.000000	
mean	0.025229	0.071730	0.085824	0.034172	
std	0.156819	0.258041	0.280104	0.181671	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

	Race_White
count	319795.000000
mean	0.766779
std	0.422883
min	0.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	1.000000

[8 rows x 23 columns]

AGE CATEGORY



0 represents 18-24

1 represents 25-29

2 represents 30-34

3 represents 35-39

4 represents 40-44

5 represents 45-49

6 represents 50-54

7 represents 55-59

8 represents 60-64

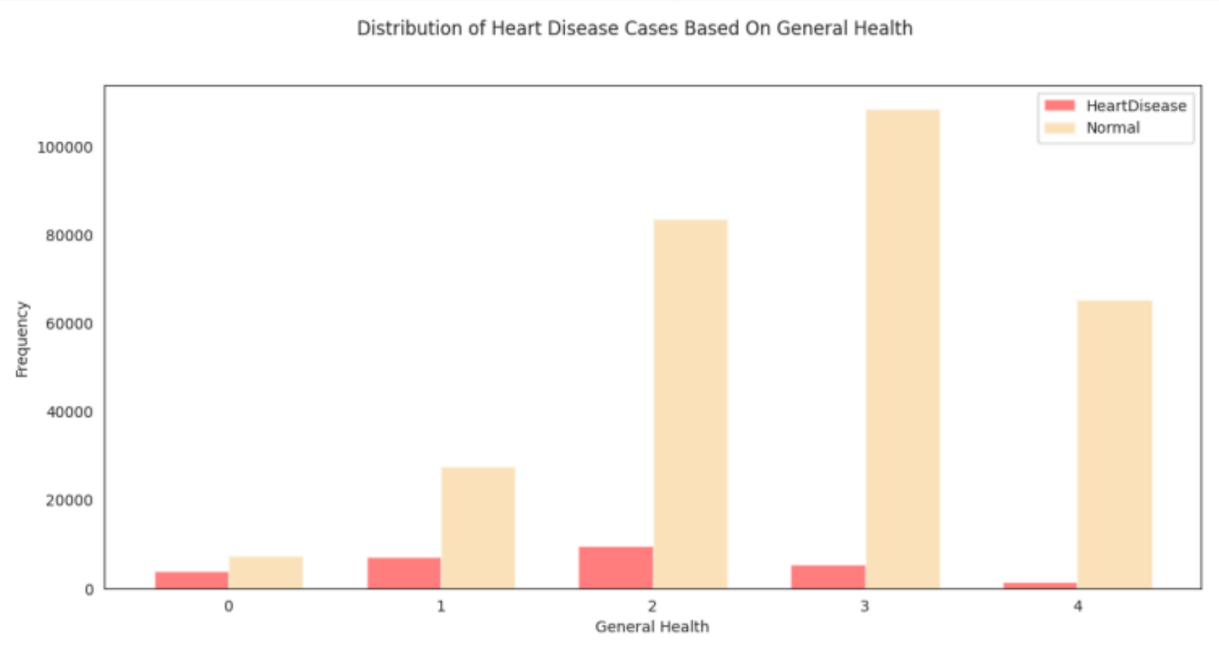
9 represents 65-69

10 represents 70-74

11 represents 75-79

12 represents 80 or older

Heart disease cases are relatively high among elderly. Old folks with age range from 60 to 80 or older are more likely to have heart diseases.



GENERAL HEALTH

0 represents poor.

1 represents fair.

2 represents good.

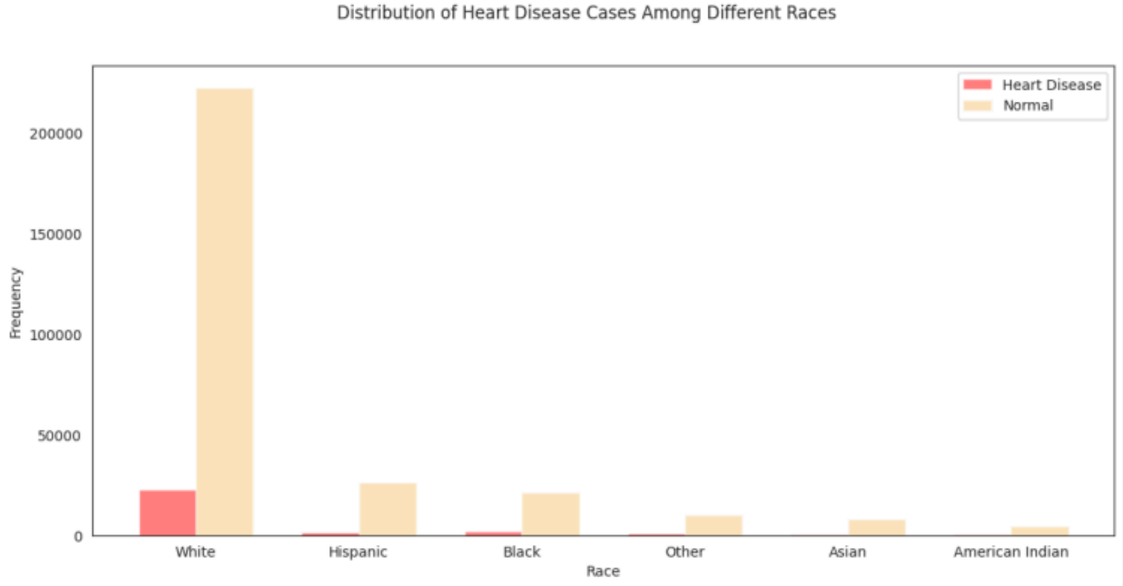
3 represents very good.

4 represents excellent.

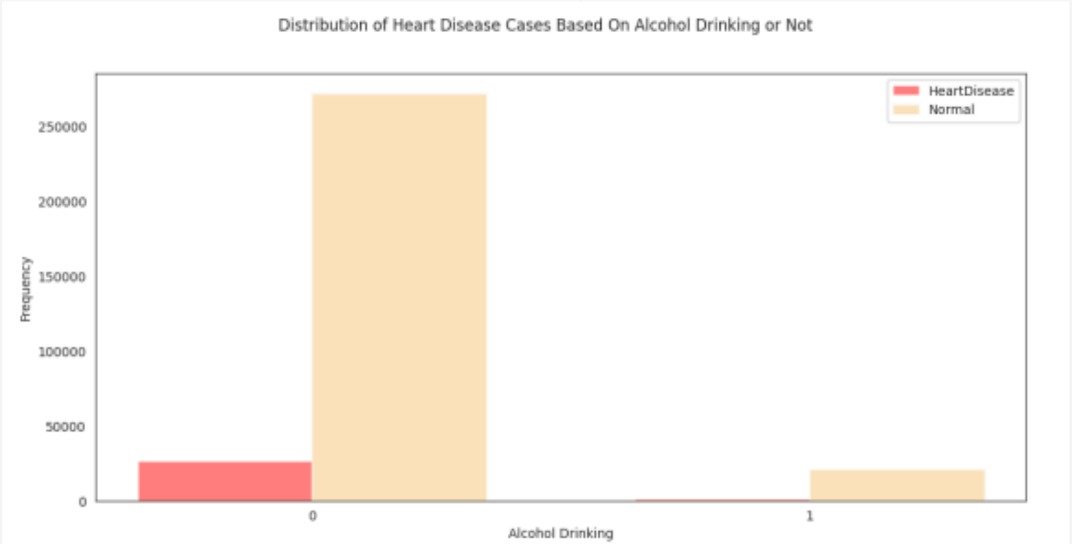
People with excellent general health are least likely to have heart diseases. On the other hand, heart diseases cases are relatively high among individuals with fair and good health.

RACES

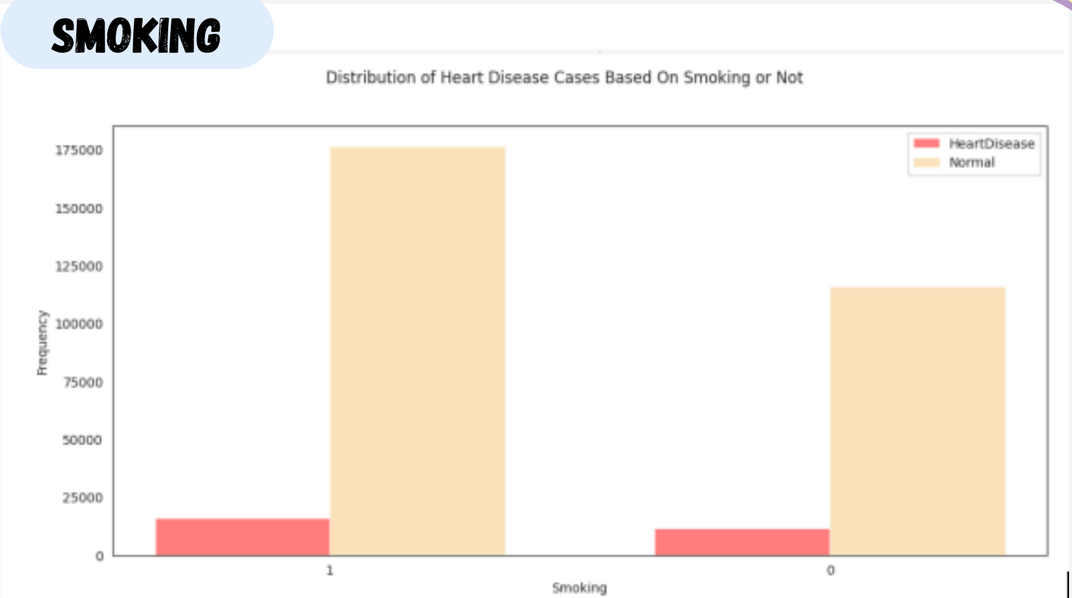
White people have the highest number of heart disease cases compared to other races. Similarly, white people with the normal condition are also highest.



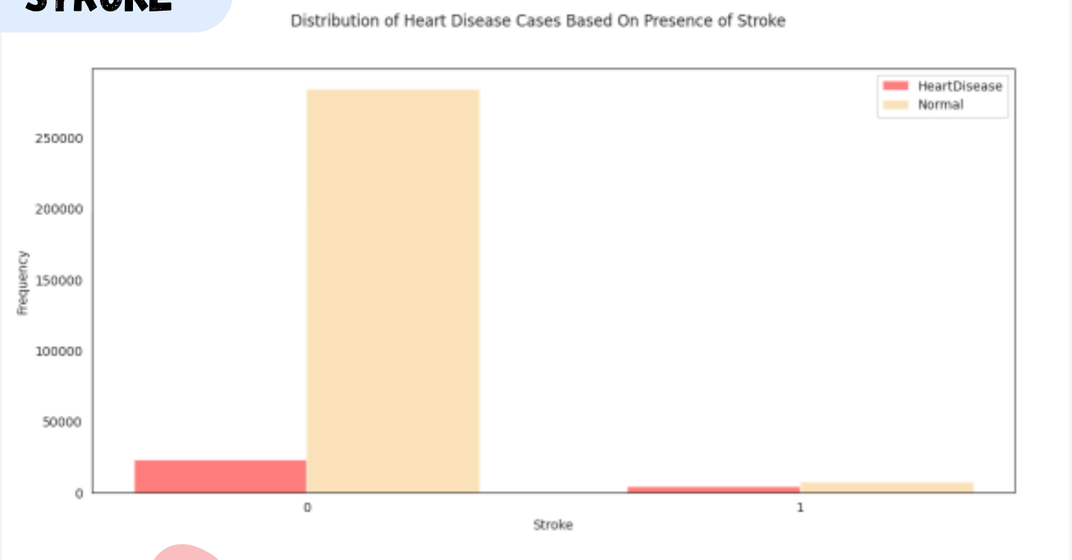
ALCOHOL DRINKING



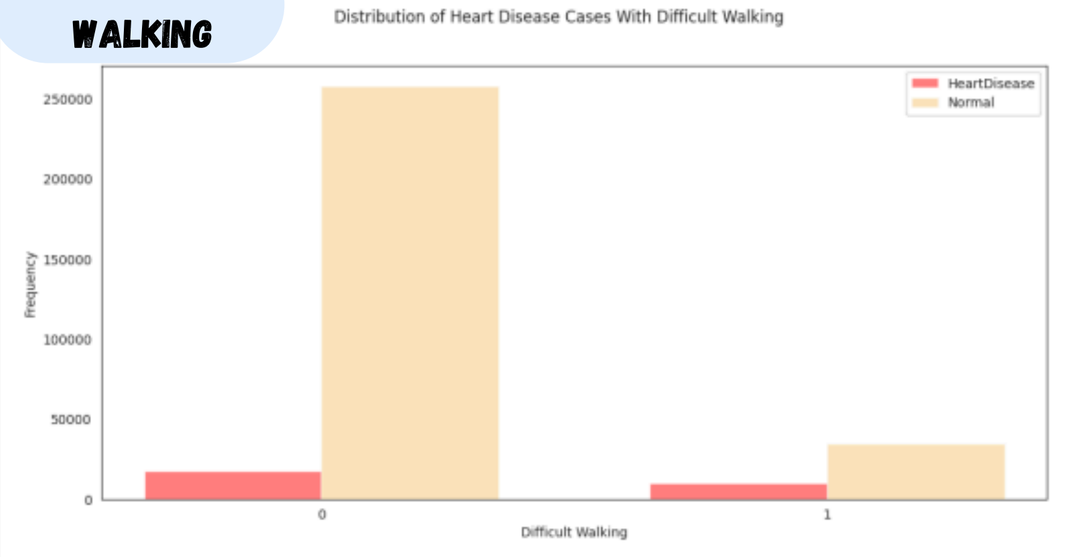
SMOKING



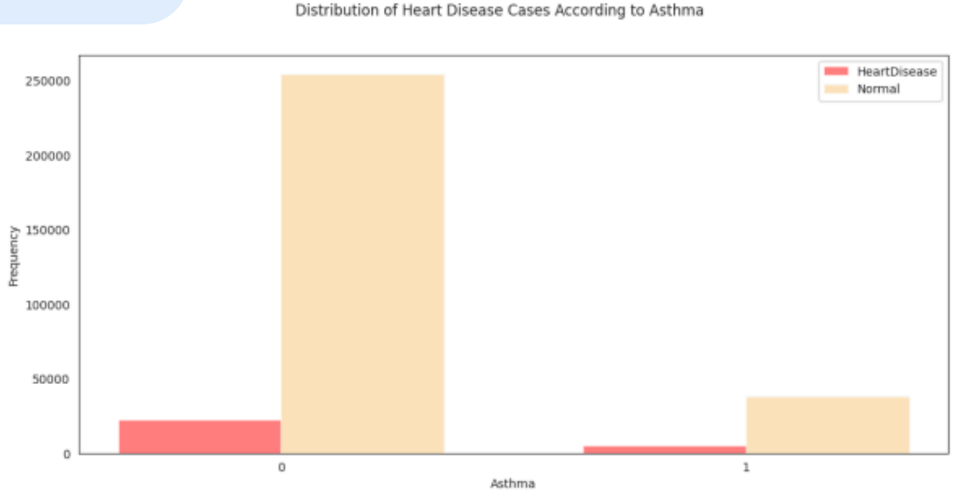
STROKE

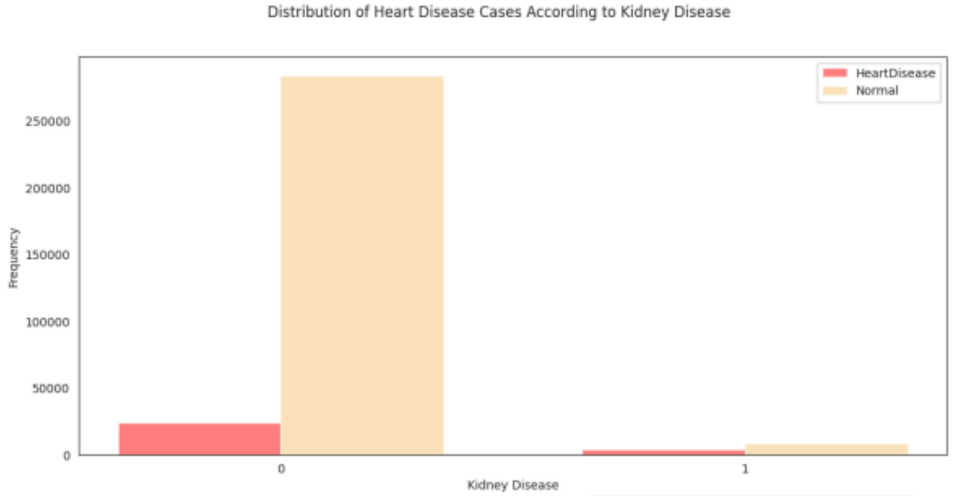


DIFFICULT WALKING



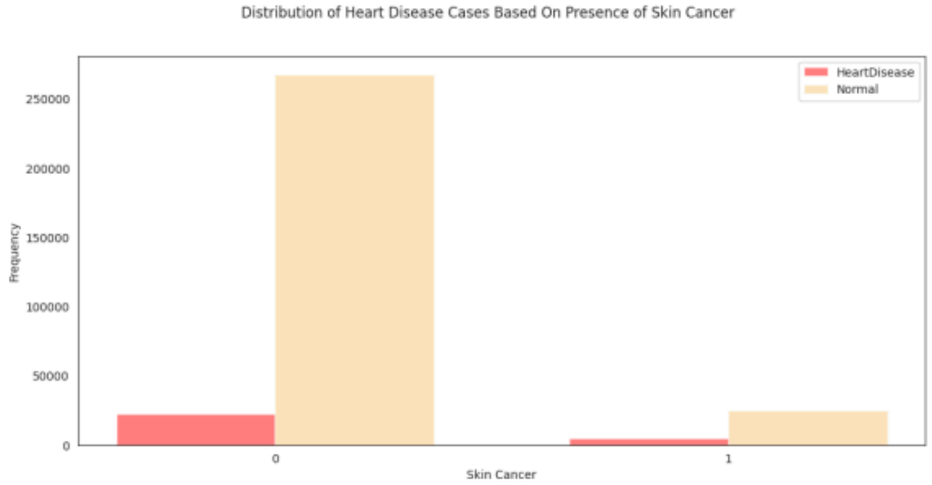
ASTHMA

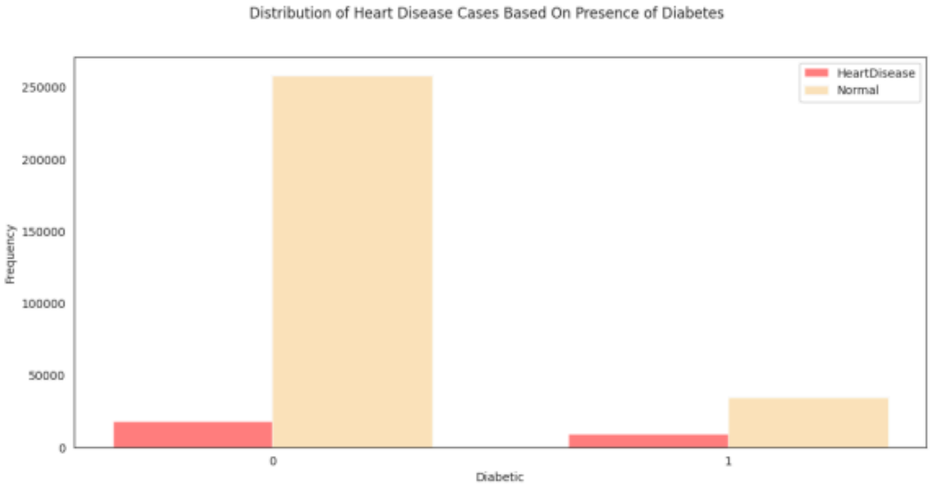




KIDNEY DISEASE

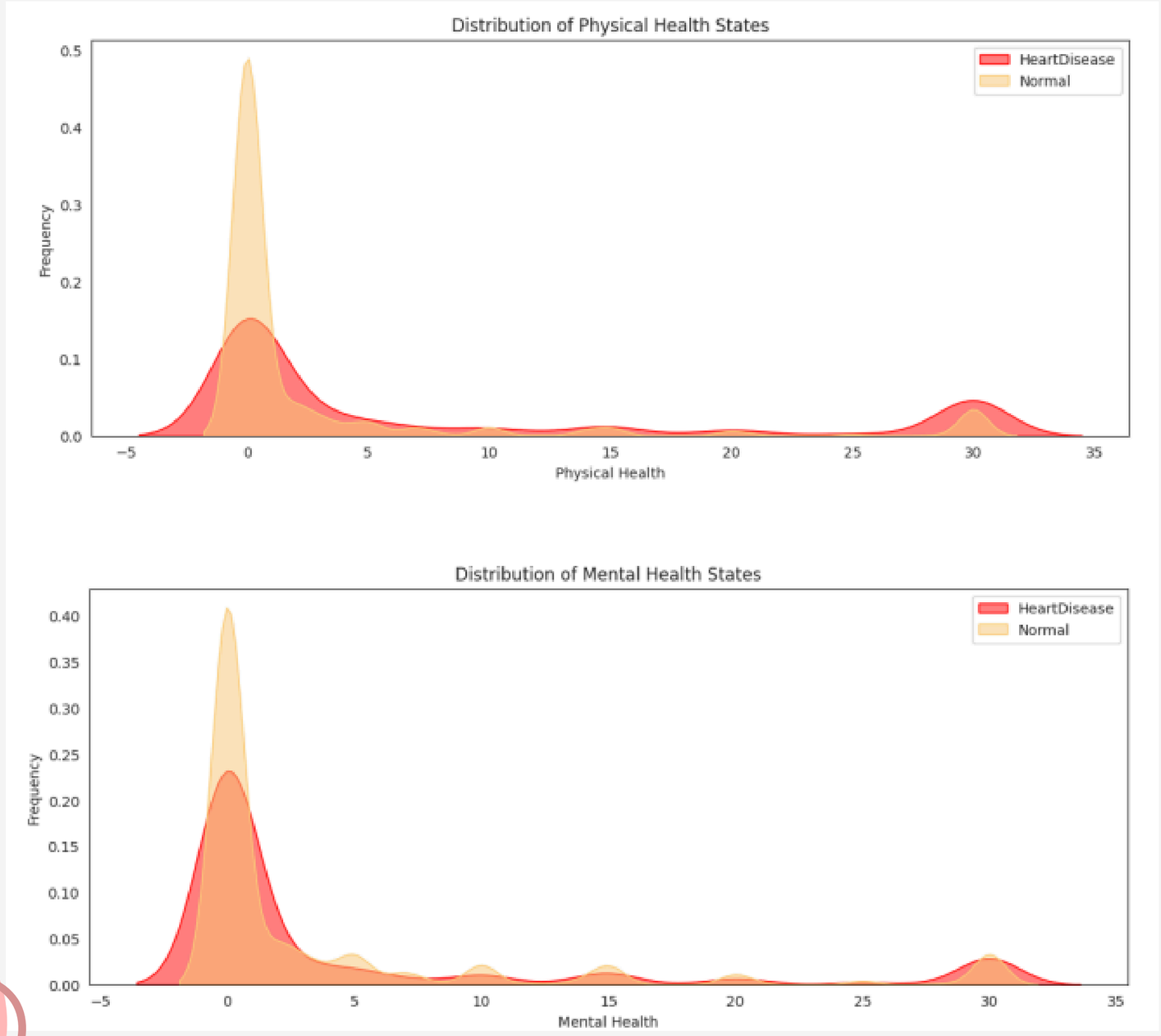
SKIN CANCER





DIABETES

DISEASE



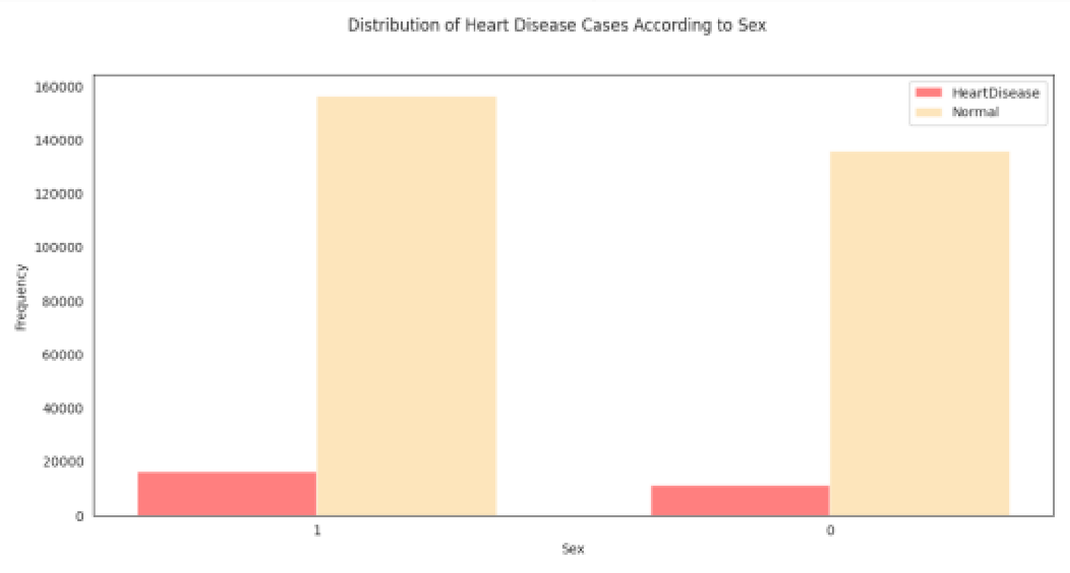
PHYSICAL HEALTH STATES

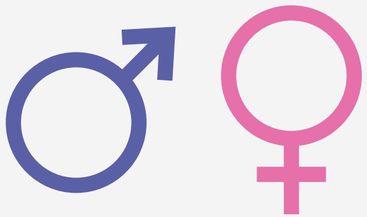
The highest number of heart disease cases occur among individuals with poor physical health.

MENTAL HEALTH STATES

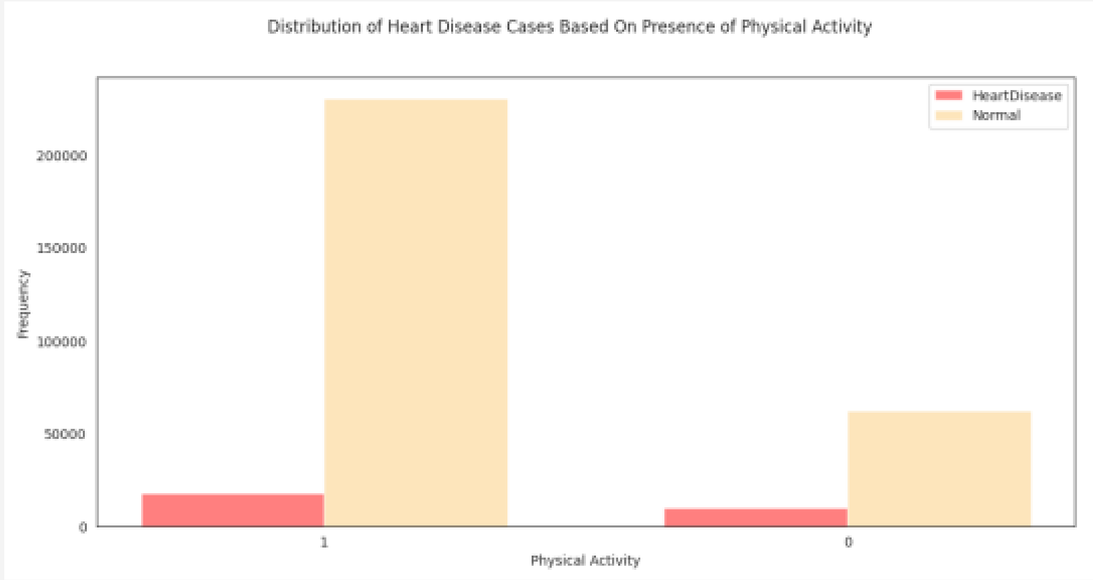
People with poor mental health states are more likely to suffer from heart diseases.

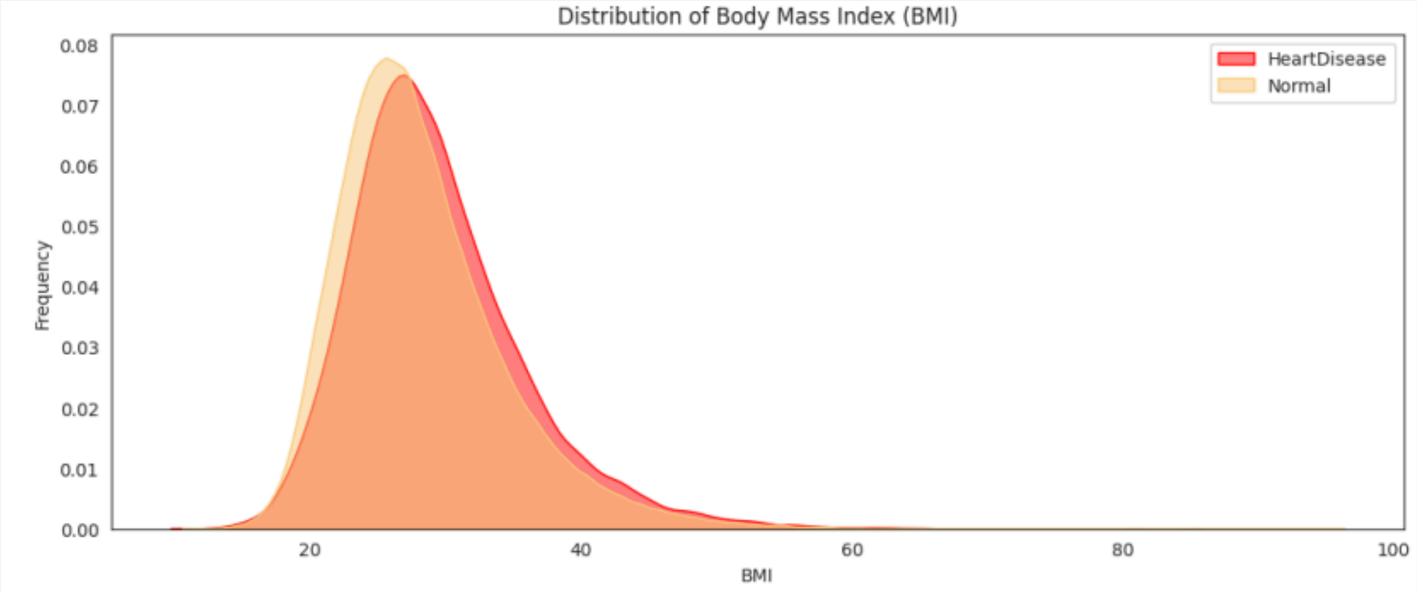
SEX





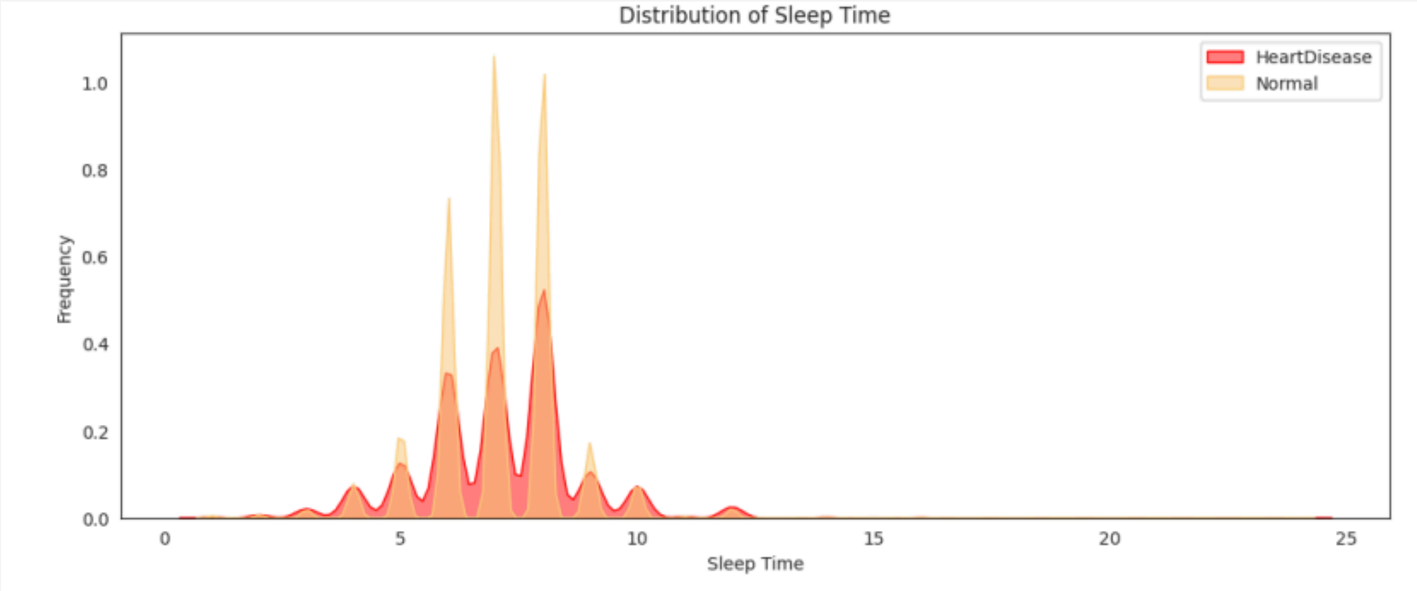
PHYSICAL ACTIVITY





BMI

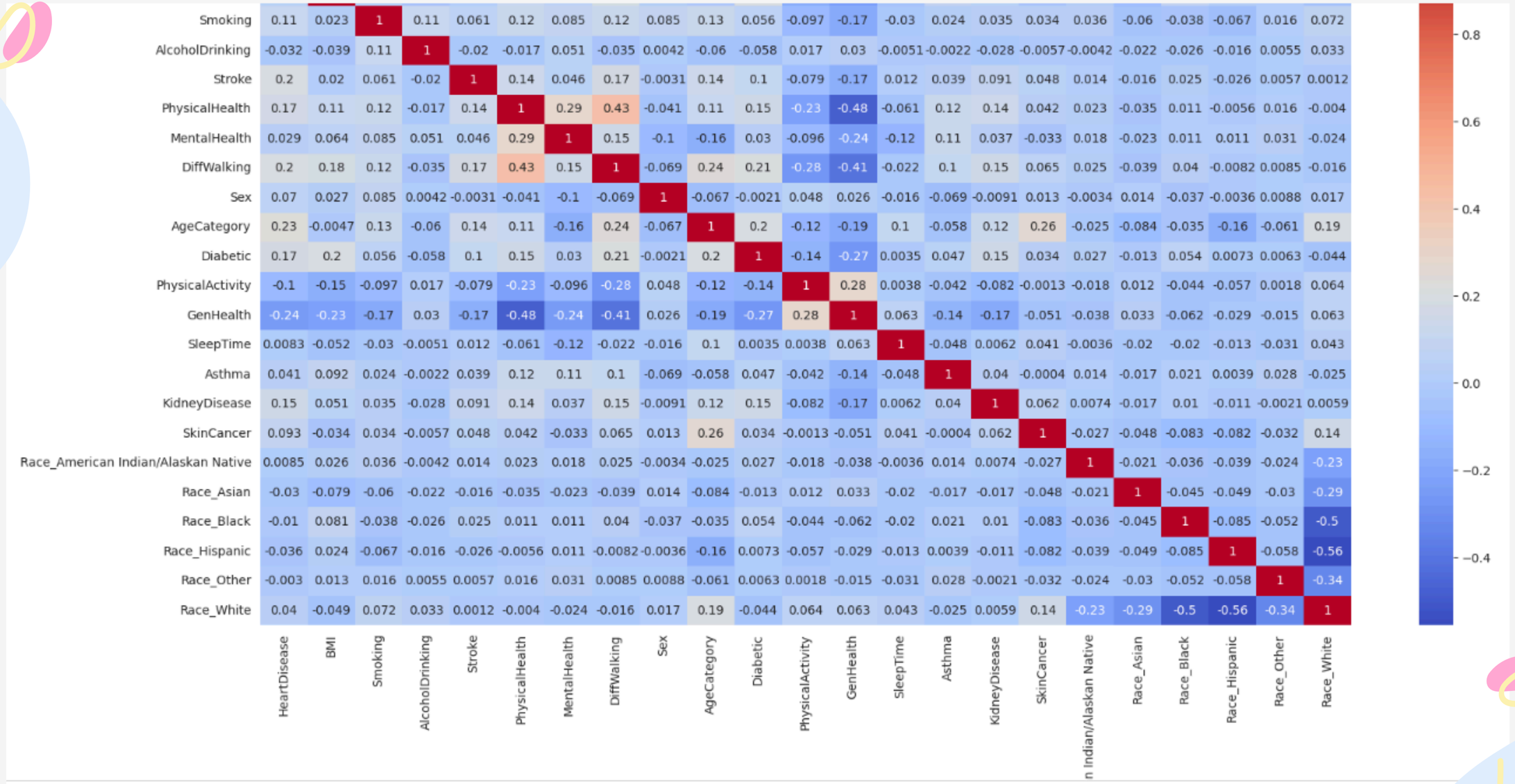
People with slightly higher BMI than normal BMI have higher likelihood to be diagnosed with heart diseases.



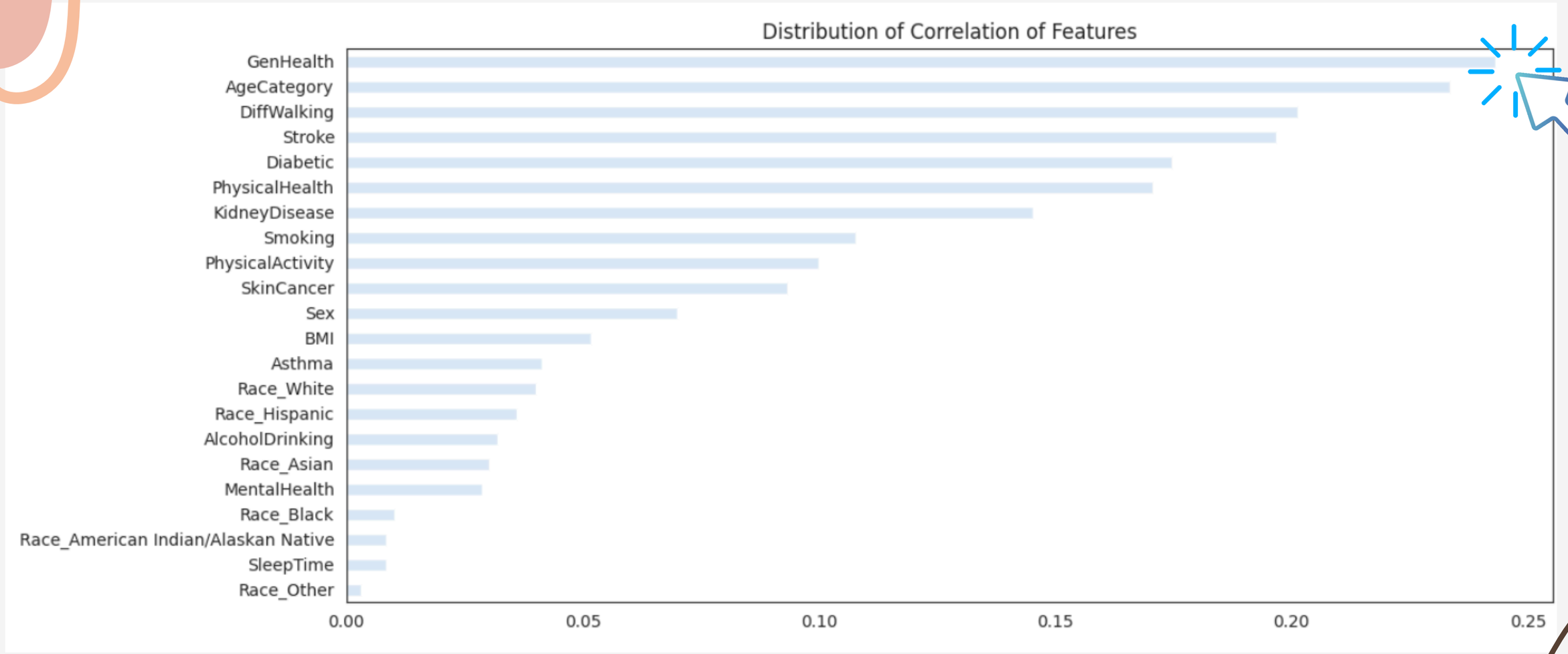
SLEEP TIME

Most people have 5 to 8 hours of sleep per day. Based on the dataset, these people have higher possibility of getting heart diseases.

CORRELATION MATRIX OF THE FEATURES



DISTRIBUTION OF CORRELATION OF FEATURES



MODEL SELECTION

Decision Tree

Decision Tree Accuracy:
0.878160474716202 3

Naive Bayes

Naive Bayes Accuracy:
0.842642758857929 2

KNN Classifier

KNN Accuracy:
0.920601135190918 5

Support Vector Machines (SVM)

Support Vector Machines Accuracy:
0.927180082559339 5

Random Forest Classifier

Random Forest Accuracy:
0.914925180598555 2

Neural Network

Neural Network Test Accuracy:
0.927223086357116 7

Decision Tree

Decision Tree Accuracy: 0.8781604747162023

Decison Tree

```
[ ]  # Initialize the Decision Tree classifier
     dt_model = DecisionTreeClassifier(random_state=42)

     # Train the Decision Tree model
     dt_model.fit(X_train, y_train)

     # Make predictions on the test set
     dt_y_pred = dt_model.predict(X_test)

     # Calculate accuracy
     dt_accuracy = accuracy_score(y_test, dt_y_pred)

     # Print the accuracy
     print("Decision Tree Accuracy:", dt_accuracy)
```

```
Decision Tree Accuracy: 0.8781604747162023
```

Support Vector Machines (SVM)

Support Vector Machines Accuracy: 0.9271800825593395

Support Vector Machines (SVM)

```
[ ]  # Initialize the SVM classifier
     svm_model = SVC(kernel='linear', random_state=42)

     # Train the SVM model
     svm_model.fit(X_train, y_train)

     # Make predictions on the test set
     svm_y_pred = svm_model.predict(X_test)

     # Calculate accuracy
     svm_accuracy = accuracy_score(y_test, svm_y_pred)

     # Print the accuracy
     print("Support Vector Machines Accuracy:", svm_accuracy)
```

```
Support Vector Machines Accuracy: 0.9271800825593395
```

Naive Bayes

Naive Bayes Accuracy: 0.8426427588579292

Naive Bayes

```
[ ]  # Initialize the Naive Bayes classifier
     nb_model = GaussianNB()

     # Train the Naive Bayes model
     nb_model.fit(X_train, y_train)

     # Make predictions on the test set
     nb_y_pred = nb_model.predict(X_test)

     # Calculate accuracy
     nb_accuracy = accuracy_score(y_test, nb_y_pred)

     # Print the accuracy
     print("Naive Bayes Accuracy:", nb_accuracy)
```

```
Naive Bayes Accuracy: 0.8426427588579292
```

Random Forest Classifier

Random Forest Accuracy: 0.9149251805985552

Random Forest Classifier

```
[ ]  # Initialize the Random Forest classifier
     rf_model = RandomForestClassifier(random_state=42)

     # Train the Random Forest model
     rf_model.fit(X_train, y_train)

     # Make predictions on the test set
     rf_y_pred = rf_model.predict(X_test)

     # Calculate accuracy
     rf_accuracy = accuracy_score(y_test, rf_y_pred)

     # Print the accuracy
     print("Random Forest Accuracy:", rf_accuracy)
```

```
Random Forest Accuracy: 0.9149251805985552
```

KNN Classifier

KNN Accuracy: 0.9206011351909185

KNN Classifier

```
[ ]  # Initialize the KNN classifier
     knn_model = KNeighborsClassifier(n_neighbors=5)

     # Train the KNN model
     knn_model.fit(X_train, y_train)

     # Make predictions on the test set
     knn_y_pred = knn_model.predict(X_test)

     # Calculate accuracy
     knn_accuracy = accuracy_score(y_test, knn_y_pred)

     # Print the accuracy
     print("KNN Accuracy:", knn_accuracy)
```

```
KNN Accuracy: 0.9206011351909185
```

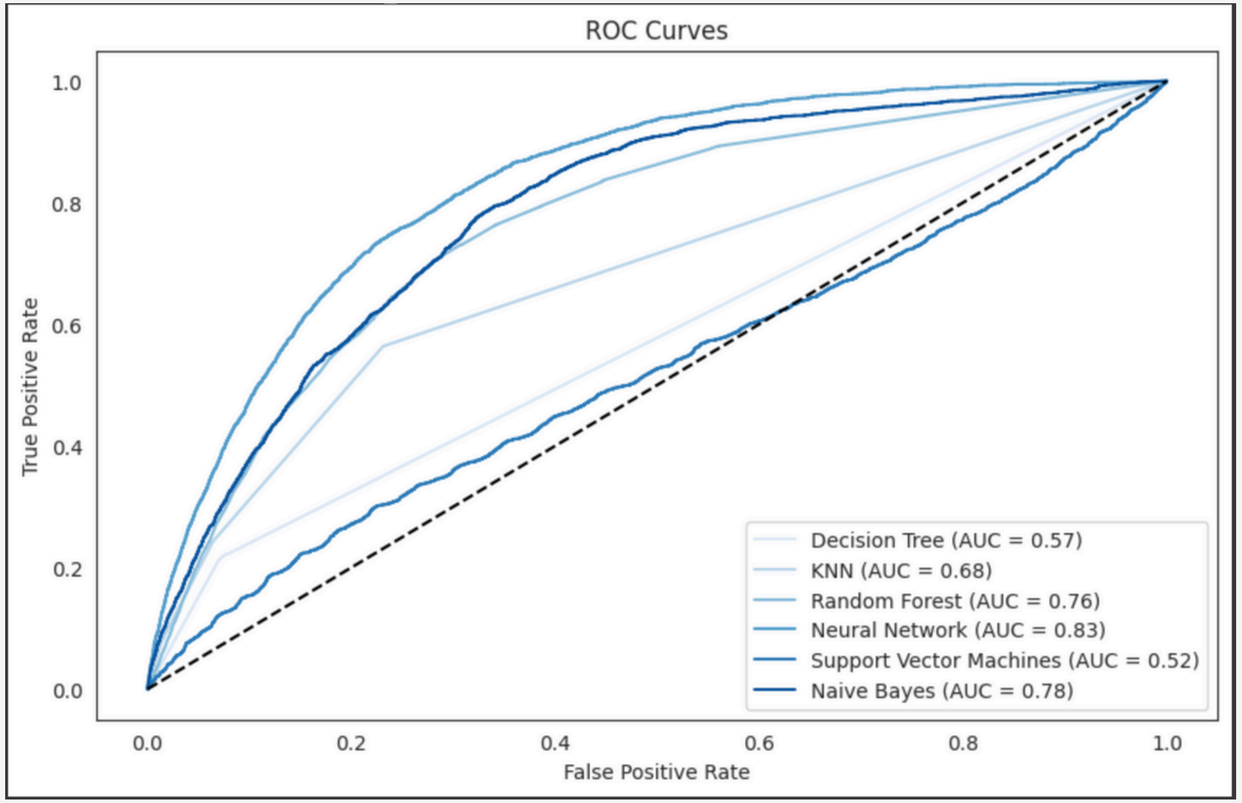
Neural Network

Neural Network Test Accuracy: 0.9272230863571167

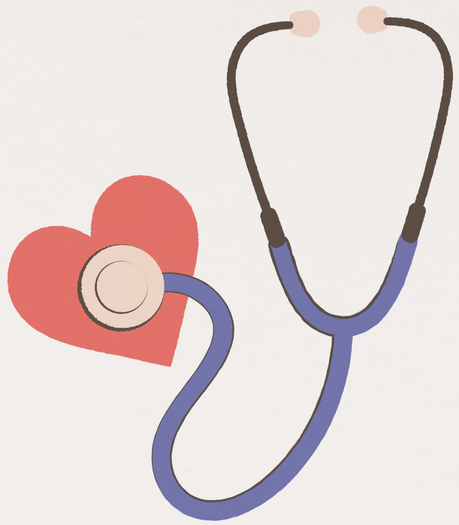
Neural Network

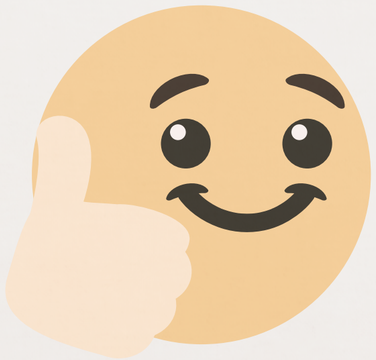
```
# Build the neural network model
nn_model = Sequential()
nn_model.add(Dense(16, activation='relu', input_shape=(22,)))
nn_model.add(Dense(8, activation='relu'))
nn_model.add(Dense(1, activation='sigmoid'))

# Compile the neural network model
nn_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the neural network model
nn_model.fit(X_train, y_train, epochs=10, batch_size=32, verbose=1)

# Evaluate the neural network model on the testing data
nn_loss, nn_accuracy = nn_model.evaluate(X_test, y_test)
print('Neural Network Test Loss:', nn_loss)
print('Neural Network Test Accuracy:', nn_accuracy)
```

MODEL SELECTION



Neural Network had been chosen





Conclusion

Through **machine learning** models, we can enhance informed decision making and early detection by

- Improving predictive accuracy
- Providing valuable insights into risk factors