East West University

Department of Computer Science and Engineering

CSE366: Artificial Intelligence

Section- 2

[FALL 2022]

Mini Project:

Heart Disease Prediction

<u>Submitted To:</u>

Redwan Ahmed Rizvee

Lecturer,
Department of CSE

<u>Submitted By:</u>

Name	ID No
K. M. Safin Kamal	2020-1-60-235
Mysha Maliha Priyanka	2020-1-60-230
Md. Hasibur Rahman	2020-1-60-068

Date of Submission: 28/ 12 / 2022

INDEX

Heart Disease Prediction

1. About Dataset

We have taken the 'personal-key-indicators-of-heart-disease' dataset for our project. Originally, the dataset come from the CDC and is a major part of the Behavioral Risk Factor Surveillance System (BRFSS), which conducts annual telephone surveys to gather data on the health status of U.S. residents. According to the CDC, heart disease is one of the leading causes of death for people of most races in the US. The original dataset of nearly 300 variables was reduced to just about 18 variables in KAGGLE. This dataset can be used to apply a range of machine learning methods, most notably classifier models (logistic regression, decision tree, KNN, random forest, etc.). So this dataset seems import to use for this project. In this report, we exploit the 'personal-key-indicators-of-heart-disease' to visualize the distribution of some attributes that seem to be correlated to heart disease and use some classification algorithms on the dataset.

2. Dataset analysis

The Python Pandas packages helps us work with our datasets. We start by acquiring the dataset into Pandas DataFrames. This dataset has about **319795** rows and **18** columns or features. This is a mixed dataset with numerical and categorical data. Here is a little description of the features:

Name	Type	Description
HeartDisease	Categorical	This is target column. Respondents that have ever reported having coronary heart disease
BMI	Numerical	Body Mass Index
Smoking	Categorical	A smoker or not (smoked at least 100 cigarettes in your entire life)
AlcoholDrinking	Categorical	A alcohol drinker or not
Stroke	Categorical	(Ever told) (you had) a stroke
PhysicalHealth	Numerical	How many days in the last month you felt poor physical health.
MentalHealth	Numerical	How many days in the last month you felt poor mental health.
DiffWalking	Categorical	you have any serious difficulty walking or climbing stairs
Sex	Categorical	Male or female
AgeCategory	Categorical	Range of ages. Fourteen-level age category
Race	Categorical	Imputed race/ethnicity value
Diabetic	Categorical	Any kind of diabetic
PhysicalActivity	Categorical	Adults who reported doing physical activity or exercise during the past 30 days
GenHealth	Categorical	Well-being.
SleepTime	Numerical	On average, hours of sleep in a 24-hour period
Asthma	Categorical	Ever told had Asthma or not
KidneyDisease	Categorical	Ever told had kidney disease or not
SkinCancer	Categorical	Ever told had Skin Cancer or not

So most of the feature (about 14) are categorical data and only 4 features are numerical.

2.1. Null or empty values

Let's check if our dataset have any empty or null values. For this we used df.info(). The output was:

```
RangeIndex: 319795 entries, 0 to 319794
Data columns (total 18 columns):
 #   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 
 8   Sex               319795 non-null  object 
 9   AgeCategory       319795 non-null  object 
 10  Race              319795 non-null  object 
 11  Diabetic          319795 non-null  object 
 12  PhysicalActivity  319795 non-null  object 
 13  GenHealth         319795 non-null  object 
 14  SleepTime         319795 non-null  float64
 15  Asthma            319795 non-null  object 
 16  KidneyDisease     319795 non-null  object 
 17  SkinCancer        319795 non-null  object 
```

So none of our values are null.

2.2. Distribution of numerical feature values

Let's check the mean, median, mode, standard deviation and others of our dataset's numerical feature. For this we used df. describe (). The output was:

```
                 BMI  PhysicalHealth   MentalHealth      SleepTime
count  319795.000000    319795.00000  319795.000000  319795.000000
mean       28.325399         3.37171       3.898366       7.097075
std         6.356100         7.95085       7.955235       1.436007
min        12.020000         0.00000       0.000000       1.000000
25%        24.030000         0.00000       0.000000       6.000000
50%        27.340000         0.00000       0.000000       7.000000
75%        31.420000         2.00000       3.000000       8.000000
max        94.850000        30.00000      30.000000      24.000000
```

2.3. Distribution of categorical feature values

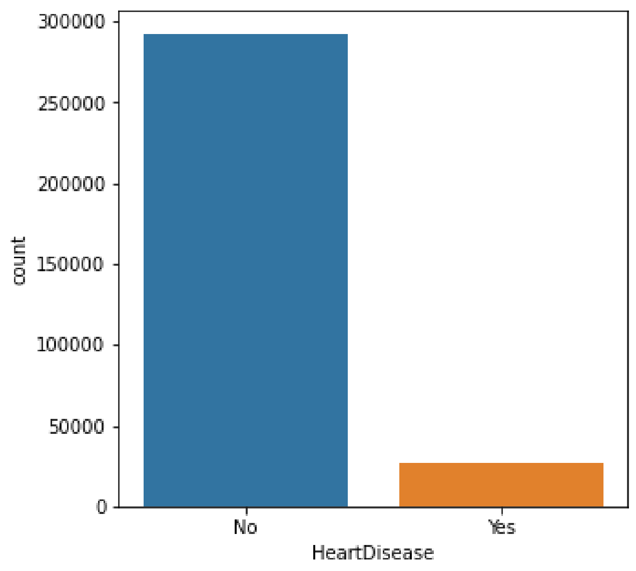
Let's check the unique, top values of our dataset's categorical feature. For this we used df.describe (include=['O']). The output was:

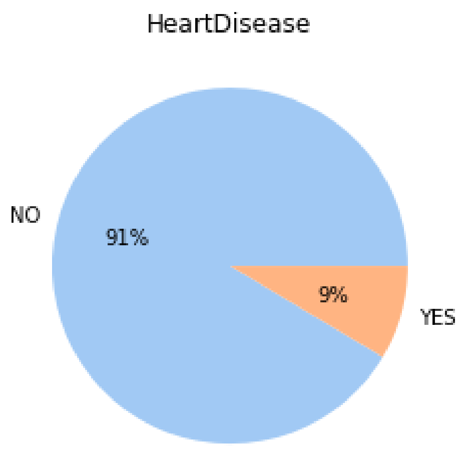
Features name	Count	Unique	Top	Frequency
HeartDisease	319795	2	No	292422
Smoking	319795	2	No	187887
AlcoholDrinking	319795	2	No	298018
Stroke	319795	2	No	307726
DiffWalking	319795	2	No	275385
Sex	319795	2	Female	167805
AgeCategory	319795	13	65-69	34151
Race	319795	6	White	245212
Diabetic	319795	4	No	269653
PhysicalActivity	319795	2	Yes	247957
GenHealth	319795	5	Very Good	113858
Asthma	319795	2	No	276923
KidneyDisease	319795	2	No	308016
SkinCancer	319795	2	No	289976

S0, 'AgeCategory' are unique across the dataset. It has 13 unique value type.

3. Analysis by visualizing data

Our target column is "HeartDisease". Almost 320 thousands personal have votes here. Let's see from bar and pie plot that how many people are suffering from various heart diseases.

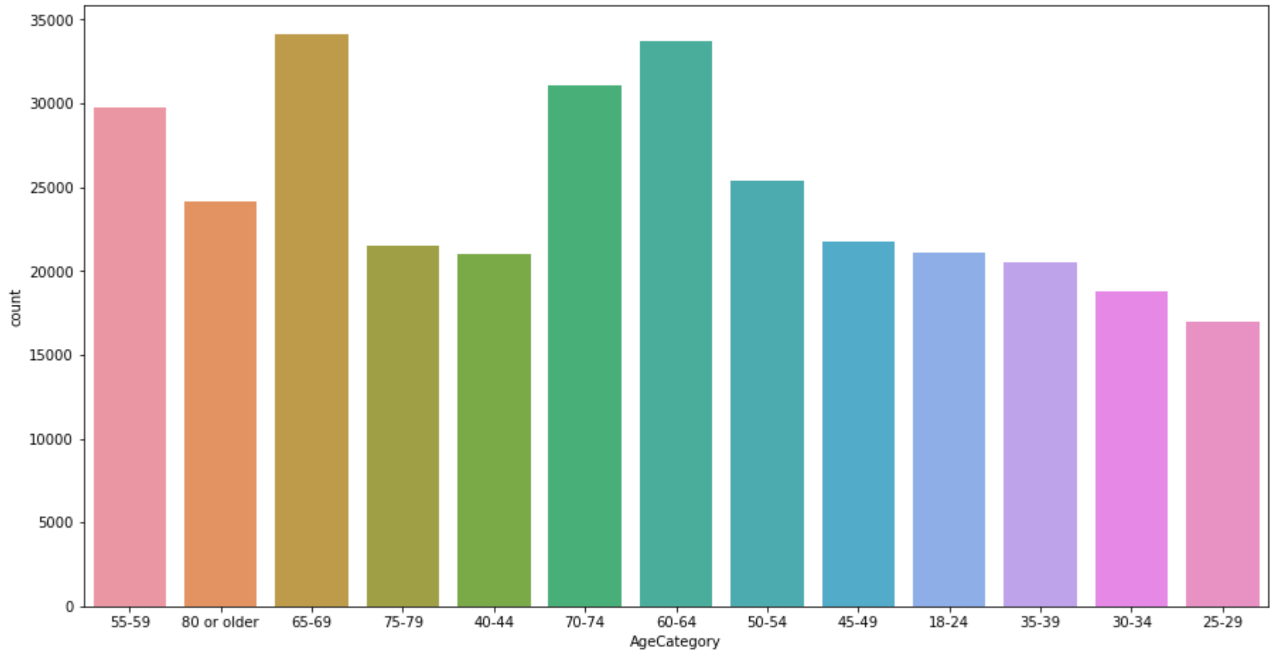




Incredibly most of the people does not suffer from heart diseases from our dataset. Only 9% people suffers here. Here we can focus on this 9% people and do more analyze on them.

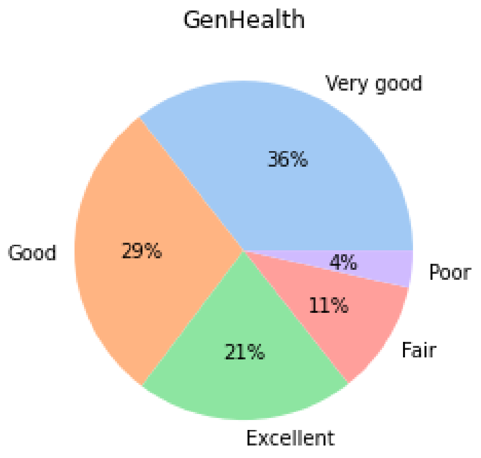
3.1. Visualization on some other features

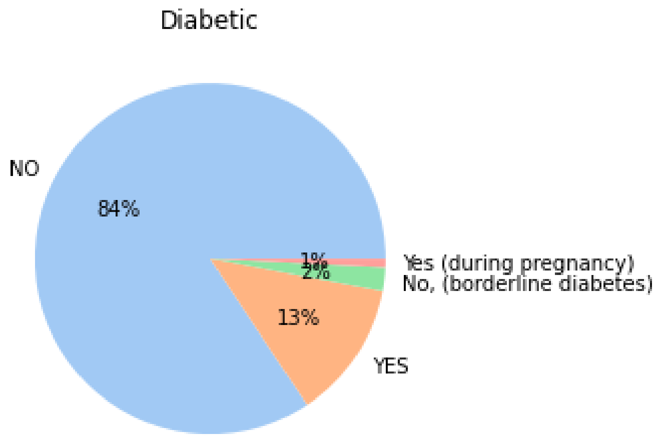
- We have seen **'AgeCategory'** are unique across the dataset from categorical type features. The bar plot of the feature is:



So people from 25 to 80+ years old performed in this survey where age 65-69 was highest in number.

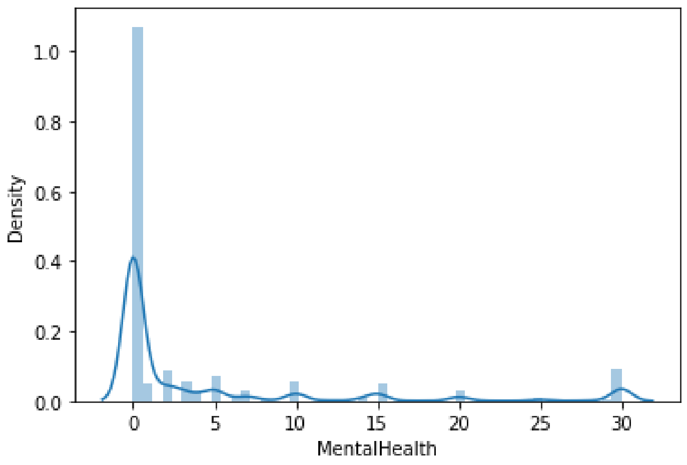
- Another two import categorical features was **"GenHealth"** and **"Diabetic".** Their pie charts are:

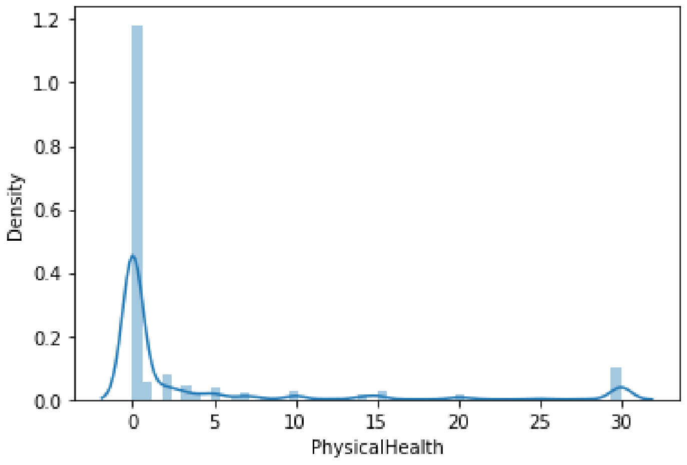


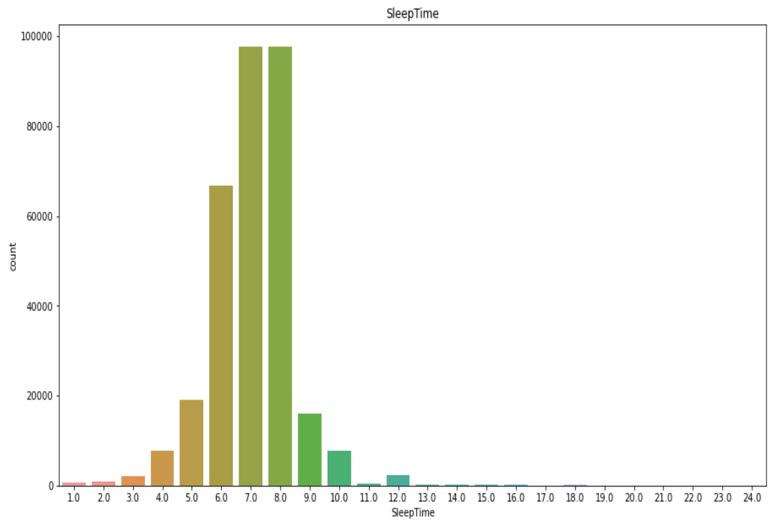


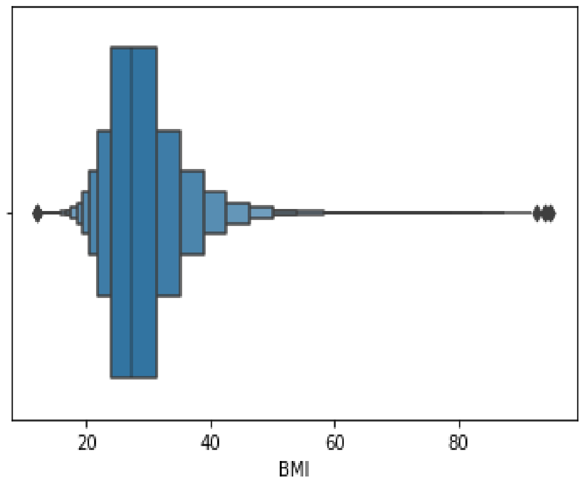
So in our dataset, most of the people are in very good health and do not have diabetes.

- From numerical features we have plotted the **'MentalHealth'**, **'PhysicalHealth'**, **'SleepTime'** and **'BMI'** features









So from the graphs, we can decide that most of people are in good mental and physical health. But some people are mentally and physically poor (high density in 30). Close inspection reveals that the graphs of mental and physical health are nearly identical. It is because people who are in good physical health also have good mental health, and vice versa. So mental and physical health are related to each other.

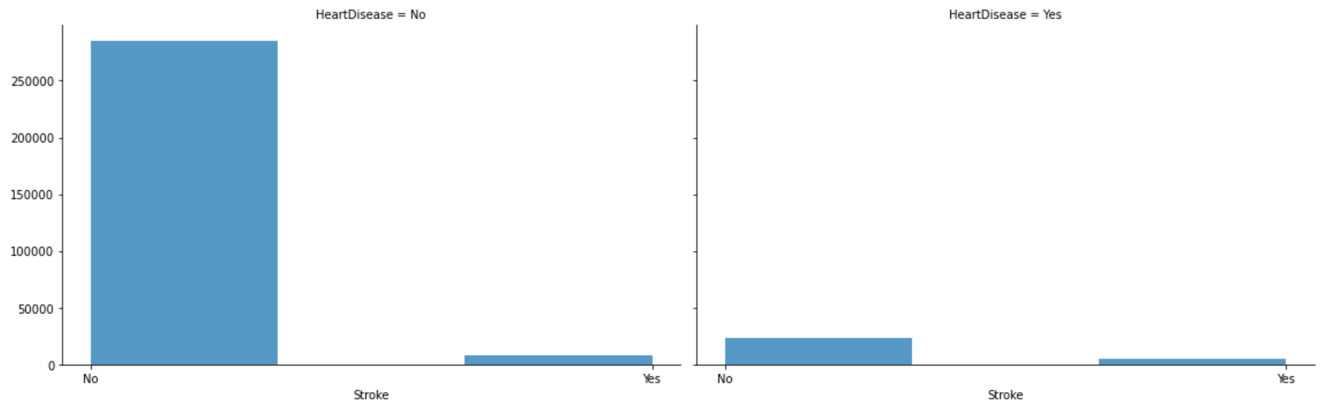
Again, most people sleep 7 to 8 hours per day and have a BMI of 25 to 30.

It is also noticeable that there are some outliers in our dataset.

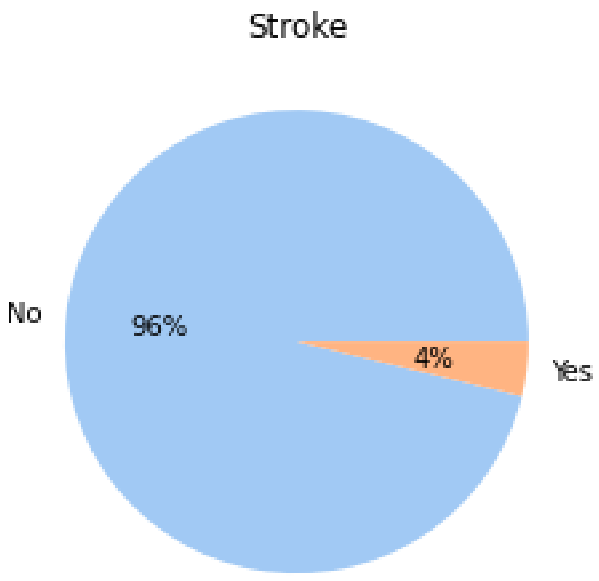
3.2. Visualizing correlation of features

As our target column is “HeartDisease”, we will see how it relates to other features.

- Heart disease and Stroke correlation:

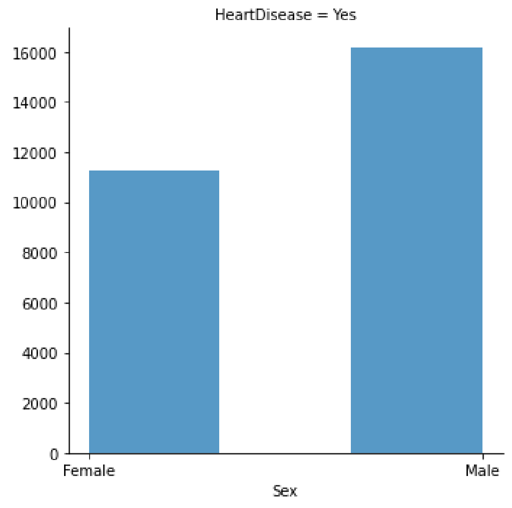


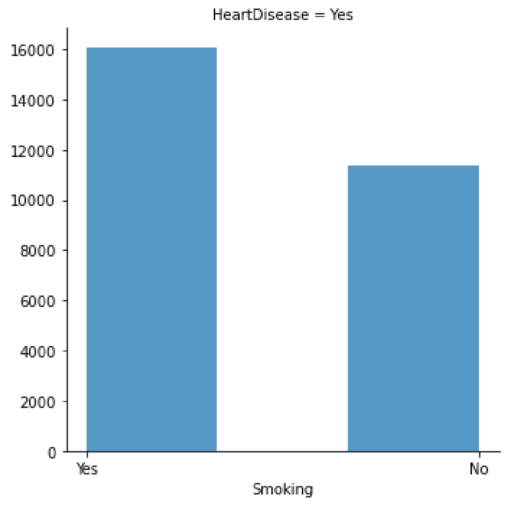
From the graph we observe that in both cases (heart disease Yes or No) non stroke person is higher. It is because in our dataset non stroke person is higher. From the pie chart we see almost 96% people did not have stroke. So we can say there is a biasness in here.

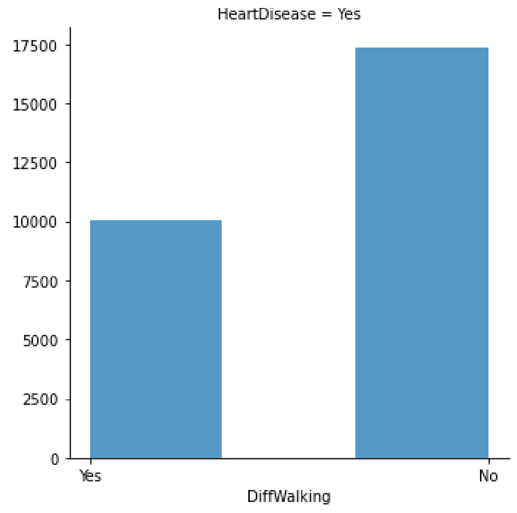


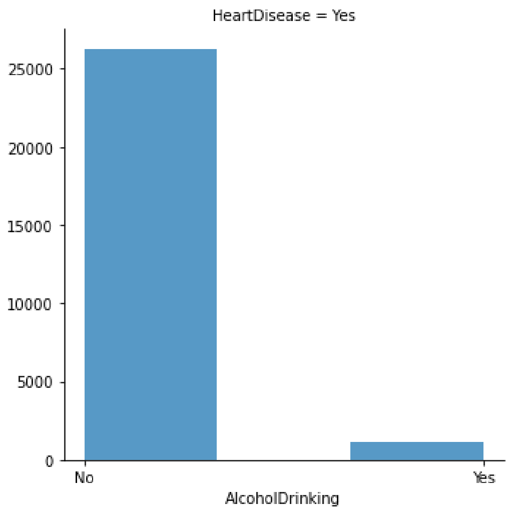
- Correlation with Heart disease (Yes) and Smoking, Sex, Difficult walking, Alcohols drinking :

This section focuses on the causes of heart disease. We will try to find out what kind of people are suffer from heart diseases.



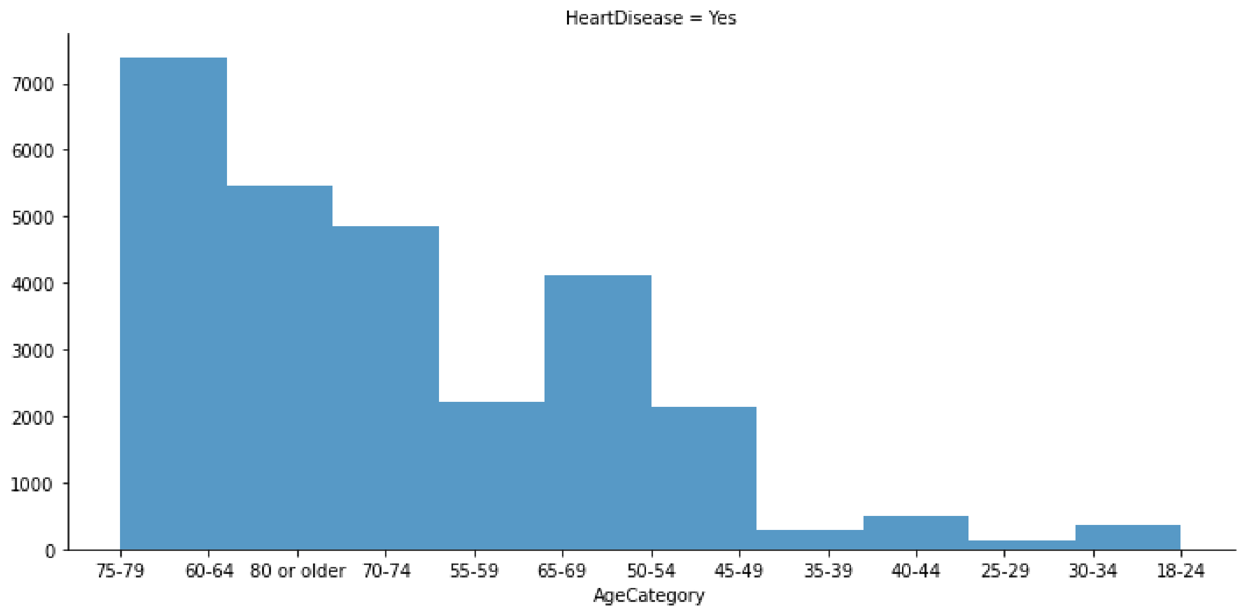






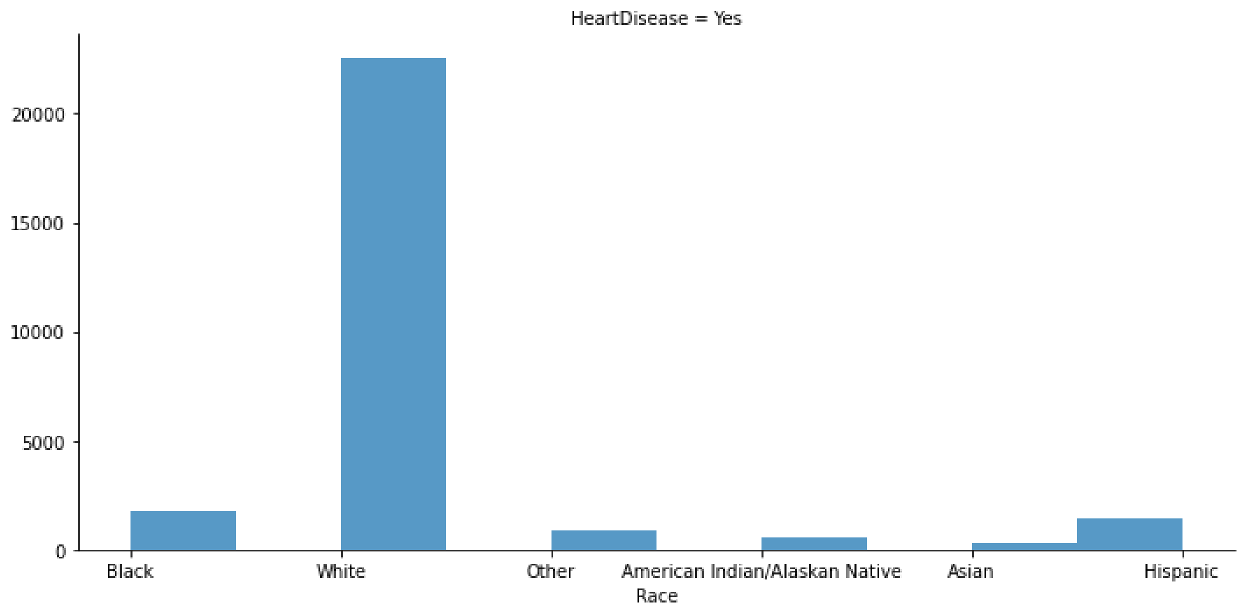
We all know, male suffers more than female from heart diseases. In our dataset have discovered the same. Smokers also suffers from heart diseases more. People who exercise or walk more have a lower risk of developing heart disease. But alcohol drinkers have low risk in our dataset. Because in our dataset almost 97% are non-alcoholic (298018 out of 319795).

- Correlation with Heart disease (Yes) and Age category:



Heart disease affects the majority of people aged 75 to 79.

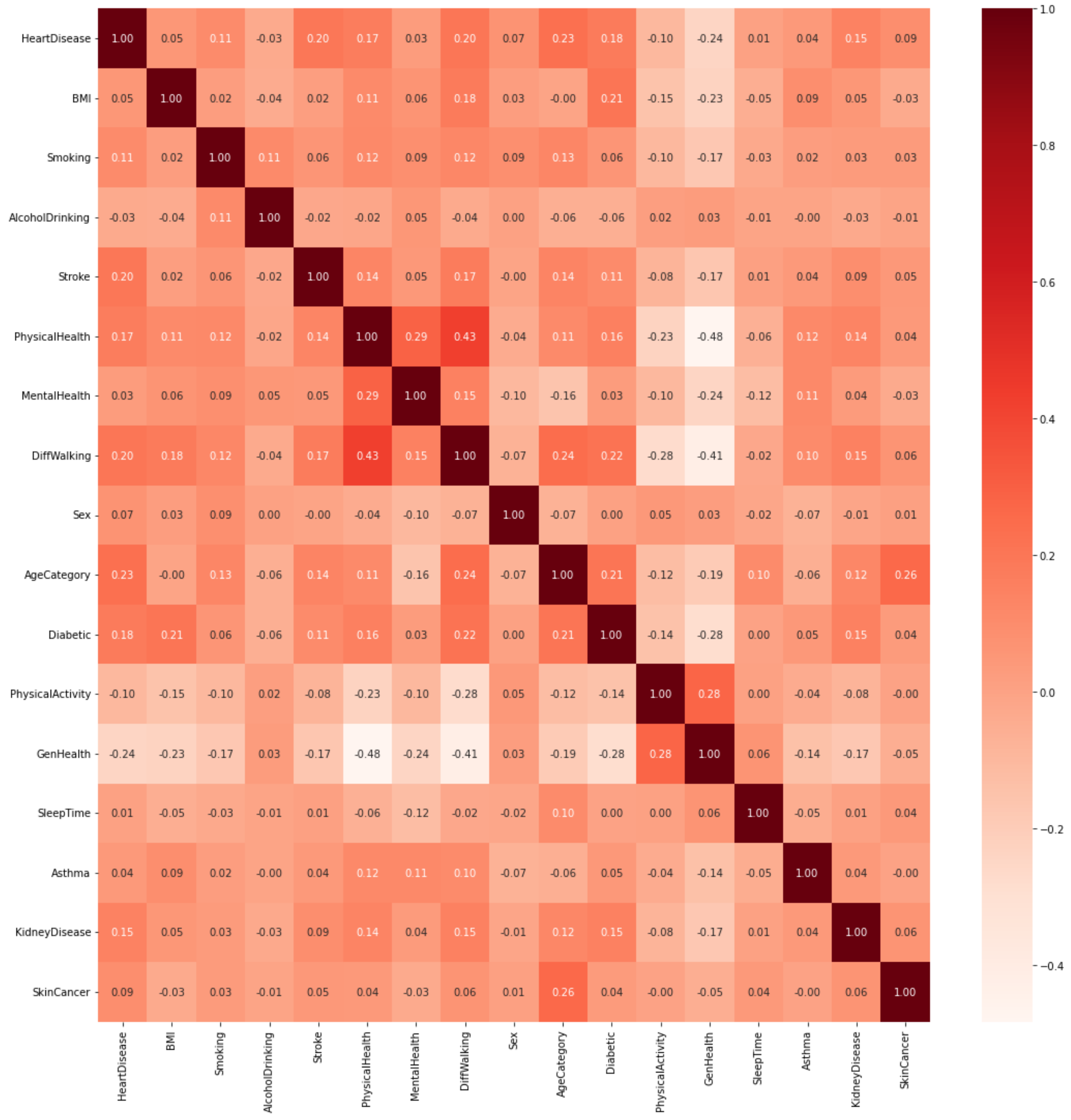
- Correlation with Heart disease (Yes) and Age category:



Here, the highest number is among white people. We know that the dataset's data is collected from people in the United States, and it is natural that the majority of the people will be white. Again, race is not related to heart disease. Any person of any race can suffer from it.

- Correlation with every features:

As our dataset has total 17 features, it is not possible to show every correlation with target column and features individually. So we use the heat map to show the Correlation. Here the deeper the color the deep the relation between two features or column.



4. Data cleaning and preprocessing

4.1. Cleaning

Data cleaning is a technique that is used to drop the unnecessary features from the dataset. In our dataset the Race feature seems unnecessary. Because heart disease does not depends on skin color, any person can be suffer from it. Even we have not found any relation in our correlation section. So we dropped the column from the dataset using df=df.drop(['Race'], axis=1).

4.2. Preprocessing (Encoding Categorical Values and Standard Scaler)

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In our dataset we have many categorical features. We cannot apply them in models until we make them numerical. The features have 2 unique values we convert them into binary (0 or 1). Those features are : 'HeartDisease', ‘Sex’, 'Smoking', 'AlcoholDrinking', 'Stroke', 'DiffWalking', 'PhysicalActivity', 'Asthma', 'KidneyDisease', 'SkinCancer' .

Again many features have more than 2 unique values, like: 'AgeCategory', 'GenHealth', 'Diabetic'. We have given values from 0 to their unique values. For example, 'AgeCategory' has 13 unique values. We marks them from 0 to 12 based on the unique values.

Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1. We could standardize our all features.

5. Applying classification algorithms

Now we are ready to train and predict the required solution. After cleaning and preprocessing, we divide our dataset into X and Y, where Y contains only “HeartDisease” and X contains other features. Then we split them into train and test where test size is 0 and random state is 42. After that we use the train and test in different classification algorithms. We analyze performance of the applied classification algorithms over the testing dataset with appropriate charts. As our dataset contains both test and train, we use cross validation (at 10 folds) for more accurate results. Our selected classification algorithms are:

- Logistic Regression
- Naive Bayes classifier
- Decision Tree
- Random Forrest
- KNN or k-Nearest Neighbors
- Linear Regression

5.1. Logistic regression

Logistic Regression is a useful model to run early in the workflow. Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. Logistic regression measures the relationship between the categorical dependent variable (feature) and one or more independent variables (features) by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

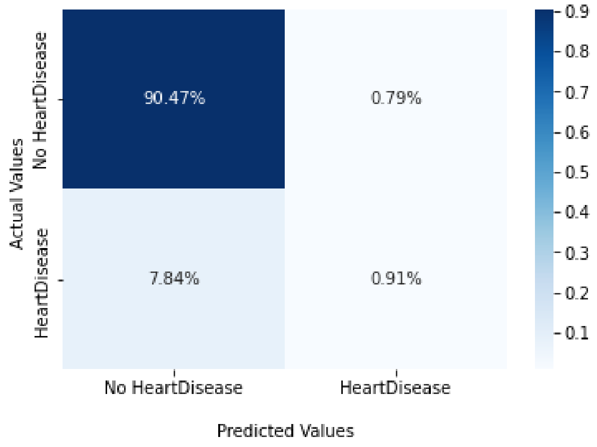
Results:

Accuracy of Logistic Regression on training dataset: 91.62%

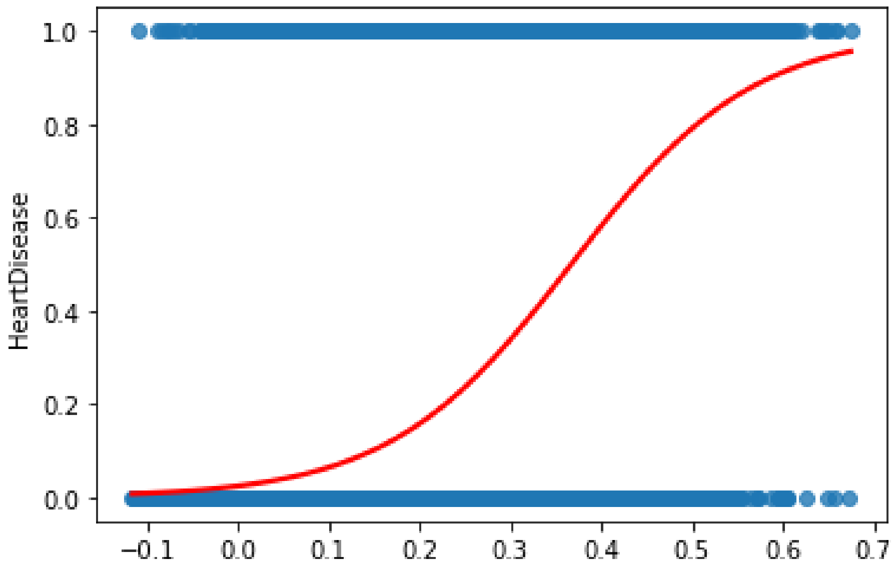
Accuracy of Logistic Regression on testing dataset: 91.38%

Then we calculate the accuracy after applying 10 folds **cross-validation** in this algorithm. Here is the result and confusion matrix:

	Train data	Test data
Mean	91.61%	91.36%
Min	91.50%	91.06%
Max	91.70%	91.65%



We have also plotted the logistic regression which looks like S. our output curve is a sigmoid function. It maps any real value into another value within a range of 0 and 1



5.2. Naive Bayes

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features) in a learning problem. The model generated confidence score is the lowest among the correct models evaluated so far.

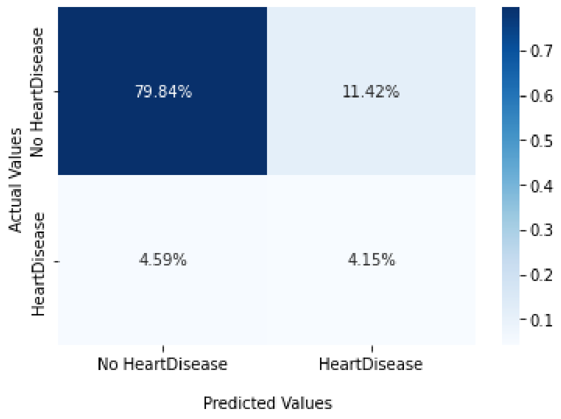
Results:

Accuracy of Naive Bayes on training dataset: 84.39%

Accuracy of Naive Bayes on testing dataset: 83.99%

Then we calculate the accuracy after applying 10 folds **cross-validation** in this algorithm. Here is the result and confusion matrix:

	Train data	Test data
Mean	84.38%	83.99%
Min	84.08%	83.36%
Max	84.68%	84.52%



5.3. Decision Tree

This model uses a decision tree as a predictive model which maps features (tree branches) to conclusions about the target value (tree leaves). Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

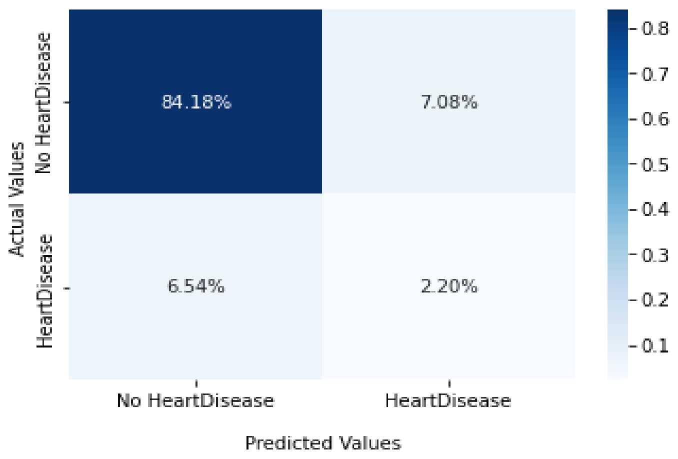
Results:

Accuracy of Decision Tree on training dataset: 99.62%

Accuracy of Decision Tree on testing dataset: 86.38%

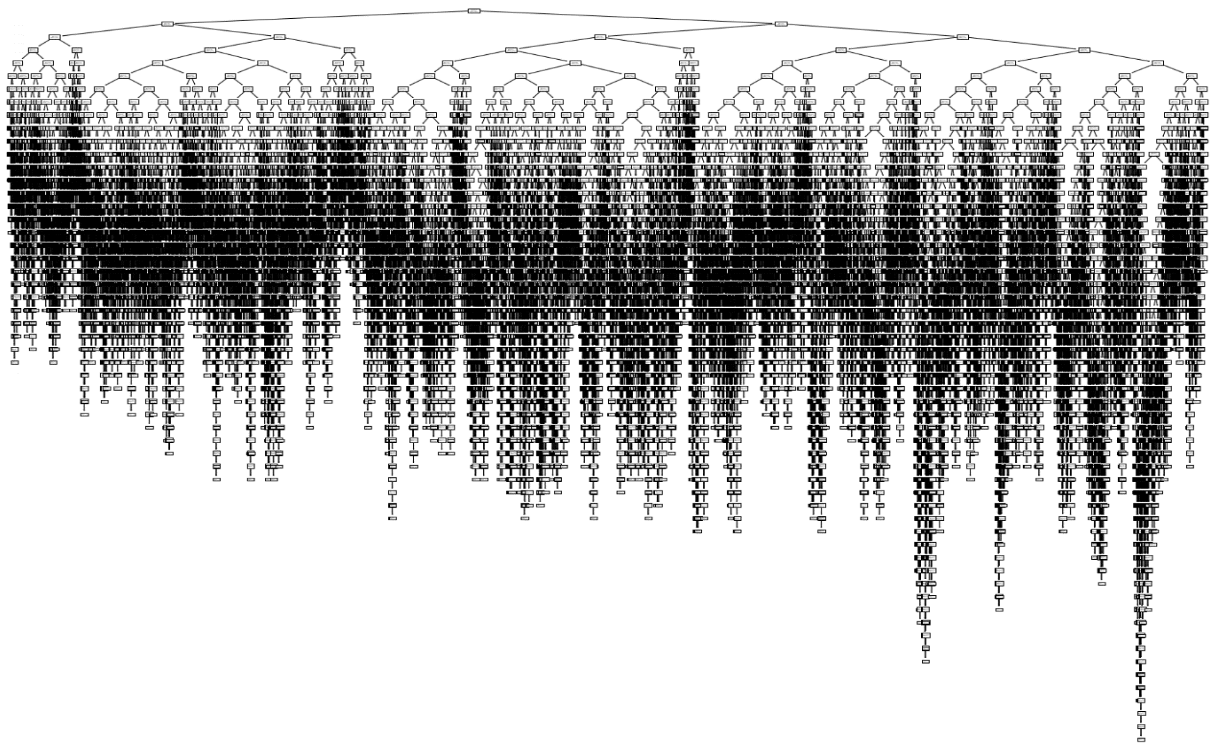
Then we calculate the accuracy after applying 10 folds **cross-validation** in this algorithm. Here is the result and confusion matrix:

	Train data	Test data
Mean	86.61%	85.70%
Min	86.24%	85.05%
Max	86.87%	86.77%



We also created the decision tree graph:

Our decision tree looks horrible because there are multiple value splits. Each column has found too much values to be splitted. We applied Standard Scaler and Binarize to check any changes but tree was same every time.



5.4. Random Forest

The next model Random Forests is one of the most popular. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. It is for classification, regression and other tasks, that operate by constructing a multitude of decision trees (n_estimators=100) at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

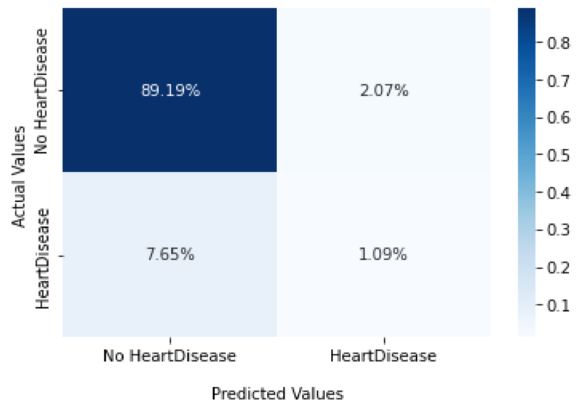
Results:

Accuracy of Random Forest on training dataset: 99.61%

Accuracy of Random Forest on test dataset: 90.28%

Then we calculate the accuracy after applying 10 folds **cross-validation** in this algorithm. Here is the result and confusion matrix:

	Train data	Test data
Mean	90.54%	90.45%
Min	90.36%	90.07%
Max	90.69%	90.78%



5.5. K Nearest Neighbors

In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. It is supervised and simple machine learning algorithm to classify instances. A sample is classified by a majority vote of its neighbors, with the sample being assigned to the class most common among its k nearest neighbors. If k = 1, then the object is simply assigned to the class of that single nearest neighbor. Mostly it is used for the Classification problems.

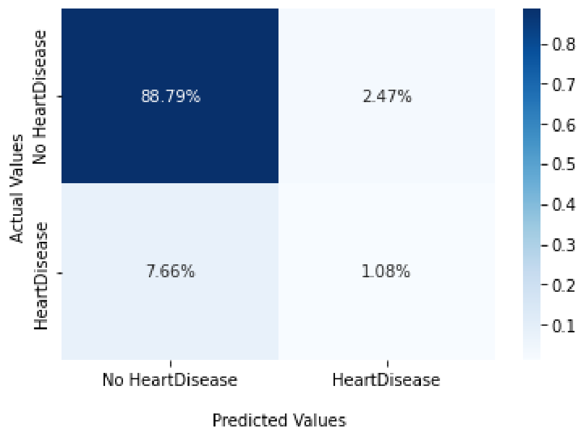
Results:

Accuracy of K Nearest Neighbors on training dataset: 93.34%

Accuracy of K Nearest Neighbors on testing dataset: 89.87

Then we calculate the accuracy after applying 10 folds **cross-validation** in this algorithm. Here is the result and confusion matrix:

	Train data	Test data
Mean	89.89%	89.68%
Min	89.72%	89.4%
Max	90.04%	90.01%



5.6. Linear regression

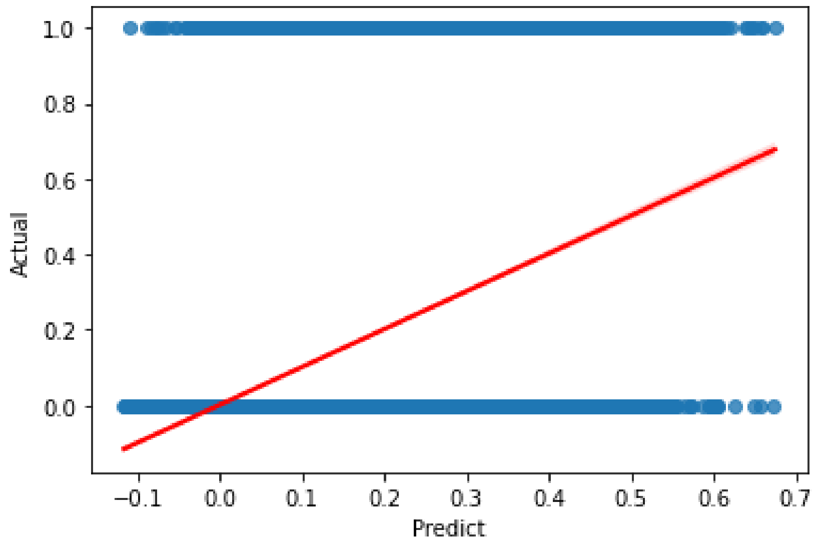
Simple linear regression is a statistical method that enables users to summarize and study relationships between two continuous (quantitative) variables. Linear regression is a linear model wherein a model that assumes a linear relationship between the input variables (x) and the single output variable (y).

Results:

Accuracy of Linear Regression on training dataset: 14.11%

Accuracy of Linear Regression on testing dataset: 13.85%

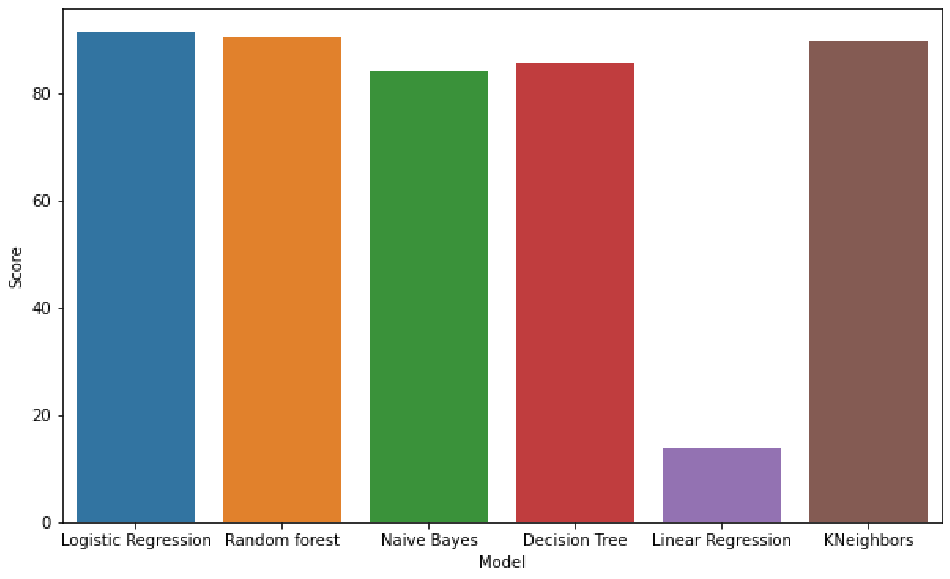
As we know linear regression is perfect for regression problem, it is showing low accuracy in our classification type dataset. We have also include the plot to see how the data are distributed.



6. Result discussion

Sorting the accuracy score of testing data of all Algorithms after cross validation

Algorithms	**Test data Output**
Logistic Regression	91.36%
Random forest	90.45%
K Nearest Neighbors	89.68%
Decision Tree	85.70%
Naive Bayes	83.99%
Linear Regression	13.85%



Here from the table and the graph we see that Logistic regression performed best in our dataset. Linear regression performed worst in here because Linear Regression is not suitable for classification problem. Random forest is in the second best and Naive Bayes is the second worst.

7. Challenges

Our first challenge was to find a perfect dataset. As we are told to use Kaggle, where thousands of datasets are available, and as we are new, it was difficult to find a suitable dataset. We have to spend a lot of time to find it.

Our second challenge was to clean and preprocess our dataset. Our dataset has 18 columns. So it was difficult to find which columns were necessary. We use different types of analysis, such as graphs, charts, and other visualizations, to find the relationship between our target column and other features. Again, 14 columns are categorical in our dataset. They must be converted to numbers. Here, use the encoding categorical values method to convert. Finally, we have a clean dataset.

Another challenge was to use appropriate algorithms in our dataset to measure its accuracy. We have to do a lot of research on our selected algorithms. We use appropriate charts to analyze the performance of the applied classification algorithms.

Again, our dataset size was so large that when we ran the code, it took a long time to process. Especially when we generated a decision tree diagram, it took almost 45 minutes to complete. Again, the KNN algorithm also took a lot of time. The whole code need one and a half hour to complete.

After facing all those challenges we tried our best. Finally, most algorithms produced the expected results.

Reference:

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