# Prediction of Coronary Heart Disease in Ten Years with Framingham Heart Study Data

Lundyn Harrelson, Rodrick Theyard Jr, and Seonwoo Kim

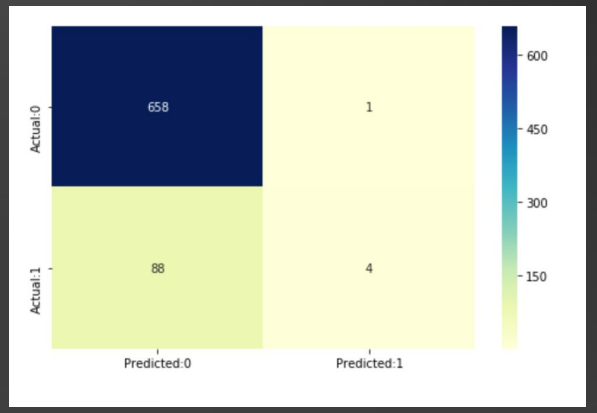
https://github.com/formulated/ISDS7070_group5.git

# Why We Choose to Analyze Data?

- Being able to predict whether someone will develop Coronary Heart Disease in ten years can aid in early detection and prevention. Knowing what factors contribute to heart disease and whether you are at risk can help the patient and their doctor develop a more accurate, long term health care plan. Making lifestyle changes and taking preventative medical treatments can help prevent or decrease the severity of Coronary Heart Disease.

# Analysis Overview

- Data: Framingham heart study dataset ($n$ = 4,238)
- Variables: 15 predictors / 1 outcome (binary)
- Success operationalization: Beat the current Kaggle best result (Accuracy .88) among 12 people
- We used three Models: Logistic regression, Randomforest, and Gradient boosting




The current Kaggle Best model (https://www.kaggle.com/neisha/heart-disease-prediction-using-logistic-regression)

# Our Predictors

Our dataset includes 15 predictors.

- Male: 0 = Female; 1 = Male
- Age: Age at time of exam
- Education: 1 = Some High School; 2 = High School or GED; 3 Some College or Vocational School; 4 = college
- CurrentSmoker: 0 = nonsmoker; 1 = smoker
- CigsPerDay: number of cigarettes smoked per day (estimated average)
- BPMeds: 0 = Not on Blood Pressure medications; 1 = Is on Blood Pressure medications
- PrevalentStroke: Patient had a stroke
- PrevalentHyp: Patient has Hypertension (≥140 mm Hg systolic and/or ≥90 diastolic)
- Diabetes: 0 = No; 1 = Yes
- totChol: cholesterol levels (mg/dL- milligrams/deciliter)
- SystolicBP: systolic blood pressure
- diaBP: Diastolic blood pressure
- BMI: Body Mass Index calculated as: Weight (kg) / Height(meter-squared)
- heartRate: BPM
- glucose: glucose levels (mg/dL- milligrams/deciliter)

We will be using those variables to predict if our observations will be at risk of developing Coronary Heart Disease within the next 10 years.

- TenYearCHD: 10 Year Risk of Coronary Heart Disease (0 = No, 1 = Yes)

## What additional data have been helpful?

Even though our dataset includes variables that are risk factors associated with coronary heart disease, there are other factors that may have been helpful if they were included in our dataset.

Using information about Heart Disease from the World Health Organization, we concluded that the inclusion of variables such as exercise, diet, average sleep time and alcohol abuse might be helpful in predicting our target variable.

# Data Exploration and Cleaning

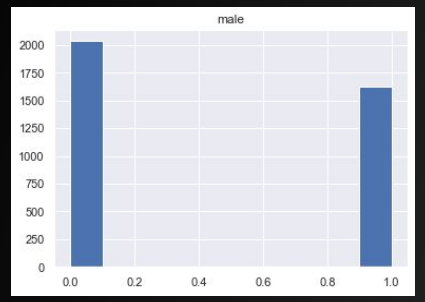
# Descriptive Statistics

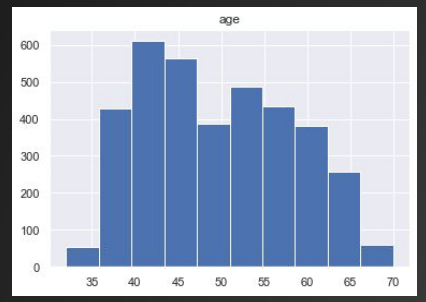
- Using the describe method, we explore variable distributions
- Some of the max values are quite larger than we would expect, meaning we may have some outliers.
  - Need to remove them to get improvement in accuracy score.
  - cigsPerDay, totChol, sysBP, diaBP, glucose

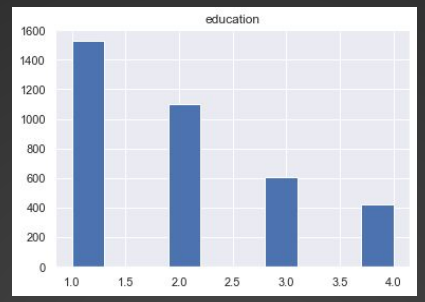
| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp |
|---|---|---|---|---|---|---|---|---|
| count | 4240.000000 | 4240.000000 | 4135.000000 | 4240.000000 | 4211.000000 | 4187.000000 | 4240.000000 | 4240.000000 |
| mean | 0.429245 | 49.580189 | 1.979444 | 0.494104 | 9.005937 | 0.029615 | 0.005896 | 0.310613 |
| std | 0.495027 | 8.572942 | 1.019791 | 0.500024 | 11.922462 | 0.169544 | 0.076569 | 0.462799 |
| min | 0.000000 | 32.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 42.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 50% | 0.000000 | 49.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 75% | 1.000000 | 56.000000 | 3.000000 | 1.000000 | 20.000000 | 0.000000 | 0.000000 | 1.000000 |
| max | 1.000000 | 70.000000 | 4.000000 | 1.000000 | 70.000000 | 1.000000 | 1.000000 | 1.000000 |

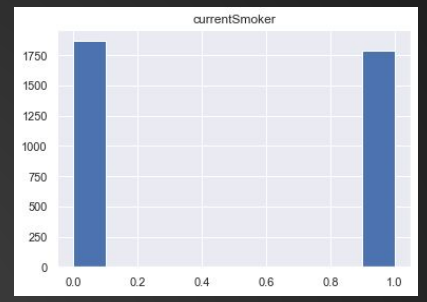
| diabetes | totChol | sysBP | diaBP | BMI | heartRate | glucose | TenYearCHD |
|---|---|---|---|---|---|---|---|
| 4240.000000 | 4190.000000 | 4240.000000 | 4240.000000 | 4221.000000 | 4239.000000 | 3852.000000 | 4240.000000 |
| 0.025708 | 236.699523 | 132.354599 | 82.897759 | 25.800801 | 75.878981 | 81.963655 | 0.151887 |
| 0.158280 | 44.591284 | 22.033300 | 11.910394 | 4.079840 | 12.025348 | 23.954335 | 0.358953 |
| 0.000000 | 107.000000 | 83.500000 | 48.000000 | 15.540000 | 44.000000 | 40.000000 | 0.000000 |
| 0.000000 | 206.000000 | 117.000000 | 75.000000 | 23.070000 | 68.000000 | 71.000000 | 0.000000 |
| 0.000000 | 234.000000 | 128.000000 | 82.000000 | 25.400000 | 75.000000 | 78.000000 | 0.000000 |
| 0.000000 | 263.000000 | 144.000000 | 90.000000 | 28.040000 | 83.000000 | 87.000000 | 0.000000 |
| 1.000000 | 696.000000 | 295.000000 | 142.500000 | 56.800000 | 143.000000 | 394.000000 | 1.000000 |

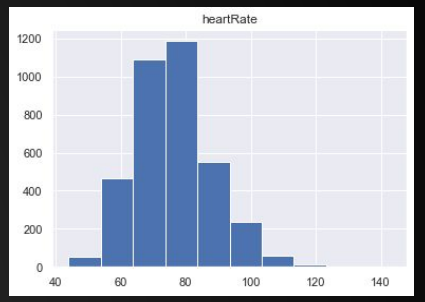
# Distributions of the Variables

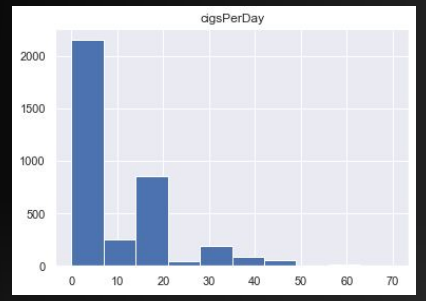


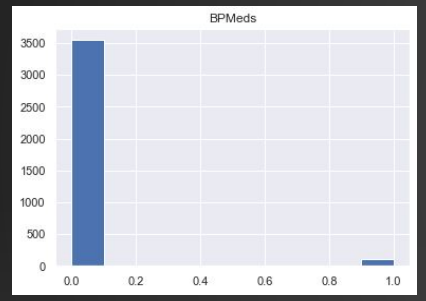



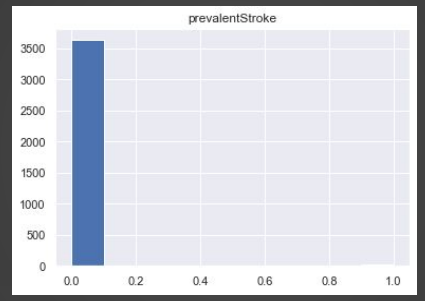



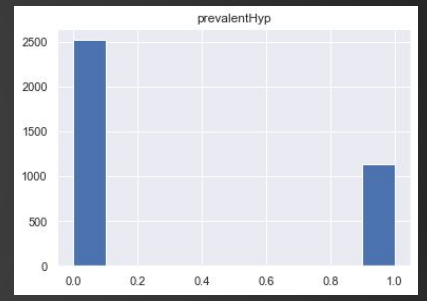



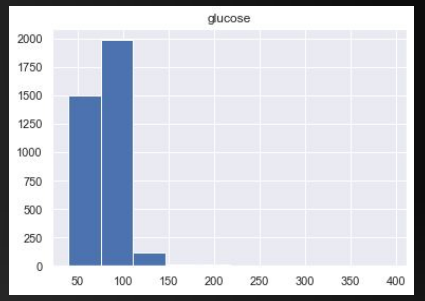



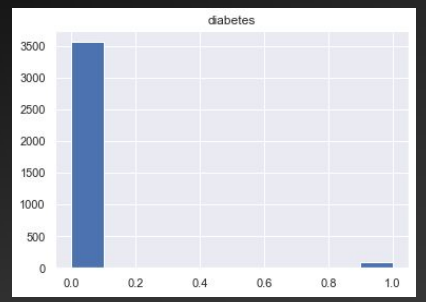



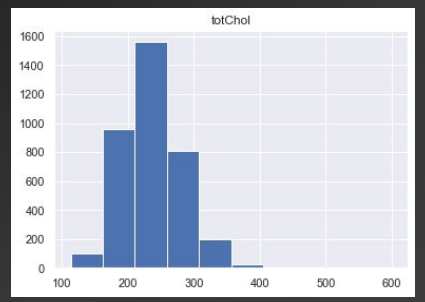



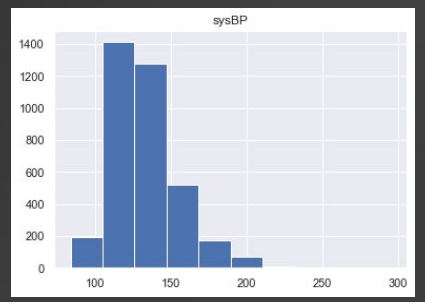



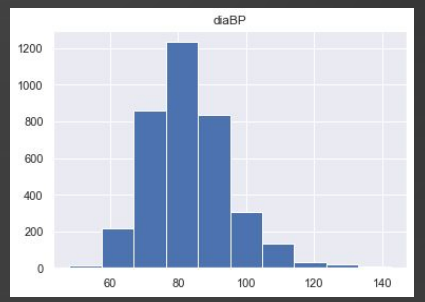



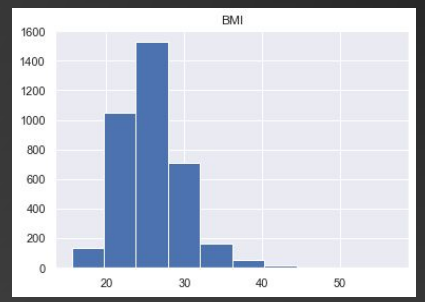



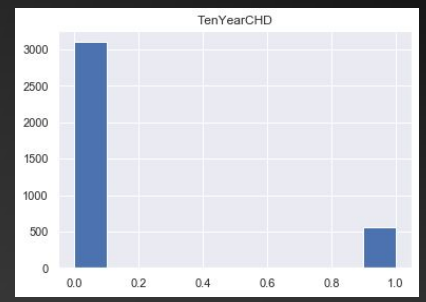



# Initial Observations of the Data

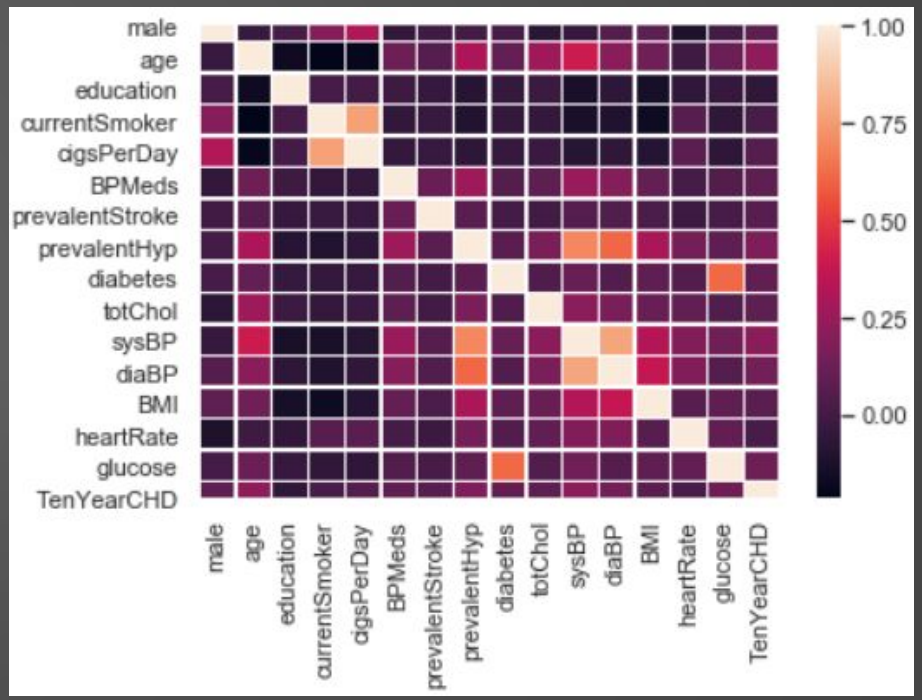
At first glance of our data set, we concluded that this is good data.

1) We have 4,238 observations which is not big data as usual in medical data, but is enough data to wrangle, build a model and develop conclusions based on our model.
2) A couple hundred missing data points which can be addressed by wrangling.
3) Possible multicollinearity between variables showing high correlation.
    a) currentSmoker and cigsPerDay
    b) sysBP, diaBP, and prevalentHyp

# Correlations

- High correlations were detected
  - Correlation of prevalentHyp and sysBP is .70.
  - Correlation of prevalentHyp and diaBP is .62.
  - Correlation of diaBP and sysBP is .78.

| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totChol | sysBP | diaBP | BMI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| male | 1.00 | -0.03 | 0.02 | 0.20 | 0.32 | -0.05 | -0.00 | 0.01 | 0.02 | -0.07 | -0.04 | 0.06 | 0.08 |
| age | -0.03 | 1.00 | -0.17 | -0.21 | -0.19 | 0.12 | 0.06 | 0.31 | 0.10 | 0.26 | 0.39 | 0.21 | 0.14 |
| education | 0.02 | -0.17 | 1.00 | 0.02 | 0.01 | -0.01 | -0.04 | -0.08 | -0.04 | -0.02 | -0.13 | -0.06 | -0.14 |
| currentSmoker | 0.20 | -0.21 | 0.02 | 1.00 | 0.77 | -0.05 | -0.03 | -0.10 | -0.04 | -0.05 | -0.13 | -0.11 | -0.17 |
| cigsPerDay | 0.32 | -0.19 | 0.01 | 0.77 | 1.00 | -0.05 | -0.03 | -0.07 | -0.04 | -0.03 | -0.09 | -0.06 | -0.09 |
| BPMeds | -0.05 | 0.12 | -0.01 | -0.05 | -0.05 | 1.00 | 0.12 | 0.26 | 0.05 | 0.08 | 0.25 | 0.19 | 0.10 |
| prevalentStroke | -0.00 | 0.06 | -0.04 | -0.03 | -0.03 | 0.12 | 1.00 | 0.07 | 0.01 | 0.00 | 0.06 | 0.05 | 0.03 |
| prevalentHyp | 0.01 | 0.31 | -0.08 | -0.10 | -0.07 | 0.26 | 0.07 | 1.00 | 0.08 | 0.16 | 0.70 | 0.62 | 0.30 |
| diabetes | 0.02 | 0.10 | -0.04 | -0.04 | -0.04 | 0.05 | 0.01 | 0.08 | 1.00 | 0.04 | 0.11 | 0.05 | 0.09 |
| totChol | -0.07 | 0.26 | -0.02 | -0.05 | -0.03 | 0.08 | 0.00 | 0.16 | 0.04 | 1.00 | 0.21 | 0.16 | 0.12 |
| sysBP | -0.04 | 0.39 | -0.13 | -0.13 | -0.09 | 0.25 | 0.06 | 0.70 | 0.11 | 0.21 | 1.00 | 0.78 | 0.33 |
| diaBP | 0.06 | 0.21 | -0.06 | -0.11 | -0.06 | 0.19 | 0.05 | 0.62 | 0.05 | 0.16 | 0.78 | 1.00 | 0.38 |
| BMI | 0.08 | 0.14 | -0.14 | -0.17 | -0.09 | 0.10 | 0.03 | 0.30 | 0.09 | 0.12 | 0.33 | 0.38 | 1.00 |
| heartRate | -0.12 | -0.01 | -0.05 | 0.06 | 0.08 | 0.02 | -0.02 | 0.15 | 0.05 | 0.09 | 0.18 | 0.18 | 0.07 |
| glucose | 0.01 | 0.12 | -0.04 | -0.06 | -0.06 | 0.05 | 0.02 | 0.09 | 0.62 | 0.05 | 0.14 | 0.06 | 0.09 |
| TenYearCHD | 0.09 | 0.23 | -0.05 | 0.02 | 0.06 | 0.09 | 0.06 | 0.18 | 0.10 | 0.08 | 0.22 | 0.15 | 0.08 |



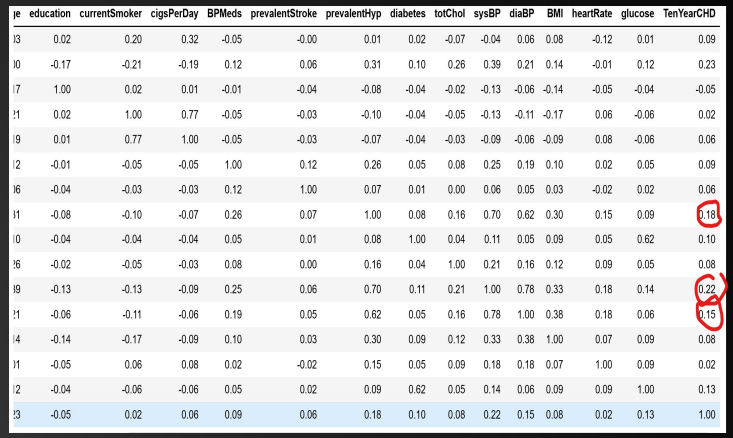
# Dealing with multicollinearity

- To prevent multicollinearity involving three of our variables, prevalentHyp, sysBP, and diaBP, we decided to keep sysBP due to it having a better correlation with the outcome than others

- From medical perspective, systolic blood pressure is given more attention as a major risk factor for cardiovascular disease for people over 50.

- current smoker and cigs per day have an extremely high correlation (r= .77). We keep current smoker because cig per day is highly skewed even with several transformation strategies such as log and sqrt

- High glucose is a representative symptom for diabetes. Thus, they share high correlation (r = .62). Because diabetes has higher correlation with outcome (r = .13) than glucose (r = .10), and it has more external validity to explain result, we only keep diabetes.



| je | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totChol | sysBP | diaBP | BMI | heartRate | glucose | TenYearCHD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 0.02 | 0.20 | 0.32 | -0.05 | -0.00 | 0.01 | 0.02 | -0.07 | -0.04 | 0.06 | 0.08 | -0.12 | 0.01 | 0.09 |
| 10 | -0.17 | -0.21 | -0.19 | 0.12 | 0.06 | 0.31 | 0.10 | 0.26 | 0.39 | 0.21 | 0.14 | -0.01 | 0.12 | 0.23 |
| 17 | 1.00 | 0.02 | 0.01 | -0.01 | -0.04 | -0.08 | -0.04 | -0.02 | -0.13 | -0.06 | -0.14 | -0.05 | -0.04 | -0.05 |
| 21 | 0.02 | 1.00 | 0.77 | -0.05 | -0.03 | -0.10 | -0.04 | -0.05 | -0.13 | -0.11 | -0.17 | 0.06 | -0.06 | 0.02 |
| 19 | 0.01 | 0.77 | 1.00 | -0.05 | -0.03 | -0.07 | -0.04 | -0.03 | -0.09 | -0.06 | -0.09 | 0.08 | -0.06 | 0.06 |
| 12 | -0.01 | -0.05 | -0.05 | 1.00 | 0.12 | 0.26 | 0.05 | 0.08 | 0.25 | 0.19 | 0.10 | 0.02 | 0.05 | 0.09 |
| 16 | -0.04 | -0.03 | -0.03 | 0.12 | 1.00 | 0.07 | 0.01 | 0.00 | 0.06 | 0.05 | 0.03 | -0.02 | 0.02 | 0.06 |
| 11 | -0.08 | -0.10 | -0.07 | 0.26 | 0.07 | 1.00 | 0.08 | 0.16 | 0.70 | 0.62 | 0.30 | 0.15 | 0.09 | 0.18 |
| 10 | -0.04 | -0.04 | -0.04 | 0.05 | 0.01 | 0.08 | 1.00 | 0.04 | 0.11 | 0.05 | 0.09 | 0.05 | 0.62 | 0.10 |
| 16 | -0.02 | -0.05 | -0.03 | 0.08 | 0.00 | 0.16 | 0.04 | 1.00 | 0.21 | 0.16 | 0.12 | 0.09 | 0.05 | 0.08 |
| 19 | -0.13 | -0.13 | -0.09 | 0.25 | 0.06 | 0.70 | 0.11 | 0.21 | 1.00 | 0.78 | 0.33 | 0.18 | 0.14 | 0.22 |
| 21 | -0.06 | -0.11 | -0.06 | 0.19 | 0.05 | 0.62 | 0.05 | 0.16 | 0.78 | 1.00 | 0.38 | 0.18 | 0.06 | 0.15 |
| 14 | -0.14 | -0.17 | -0.09 | 0.10 | 0.03 | 0.30 | 0.09 | 0.12 | 0.33 | 0.38 | 1.00 | 0.07 | 0.09 | 0.08 |
| 11 | -0.05 | 0.06 | 0.08 | 0.02 | -0.02 | 0.15 | 0.05 | 0.09 | 0.18 | 0.18 | 0.07 | 1.00 | 0.09 | 0.02 |
| 12 | -0.04 | -0.06 | -0.06 | 0.05 | 0.02 | 0.09 | 0.62 | 0.05 | 0.14 | 0.06 | 0.09 | 0.09 | 1.00 | 0.13 |
| 13 | -0.05 | 0.02 | 0.06 | 0.09 | 0.06 | 0.18 | 0.10 | 0.08 | 0.22 | 0.15 | 0.08 | 0.02 | 0.13 | 1.00 |

# EXAMINING THE DATA TYPES

```
# Check data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4240 entries, 0 to 4239
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   male             4240 non-null   int64
 1   age              4240 non-null   int64
 2   education        4135 non-null   float64
 3   currentSmoker    4240 non-null   int64
 4   cigsPerDay       4211 non-null   float64
 5   BPMeds           4187 non-null   float64
 6   prevalentStroke  4240 non-null   int64
 7   prevalentHyp     4240 non-null   int64
 8   diabetes         4240 non-null   int64
 9   totChol          4190 non-null   float64
 10  sysBP            4240 non-null   float64
 11  diaBP            4240 non-null   float64
 12  BMI              4221 non-null   float64
 13  heartRate        4239 non-null   float64
 14  glucose          3852 non-null   float64
 15  TenYearCHD       4240 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 530.1 KB
```

In order to correctly use our data, we must verify that it recognized as the correct data type. Using code and examining the output we concluded that education, cigsPerDay, BPMeds, heartRate, totChol and glucose are all listed as floats but need to be correctly identified as integers. On the left is the original data types, and on the right is how we corrected the data types and verified that it was corrected.

```
df[['education']]=df[['education']].astype('int')
df[['cigsPerDay']]=df[['cigsPerDay']].astype('int')
df[['BPMeds']]=df[['BPMeds']].astype('int')
df[['glucose']]=df[['glucose']].astype('int')
df[['totChol']]=df[['totChol']].astype('int')
df[['heartRate']]=df[['heartRate']].astype('int')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3658 entries, 0 to 4239
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   male             3658 non-null   int64
 1   age              3658 non-null   int64
 2   education        3658 non-null   int32
 3   currentSmoker    3658 non-null   int64
 4   cigsPerDay       3658 non-null   int32
 5   BPMeds           3658 non-null   int32
 6   prevalentStroke  3658 non-null   int64
 7   prevalentHyp     3658 non-null   int64
 8   diabetes         3658 non-null   int64
 9   totChol          3658 non-null   int32
 10  sysBP            3658 non-null   float64
 11  diaBP            3658 non-null   float64
 12  BMI              3658 non-null   float64
 13  heartRate        3658 non-null   int32
 14  glucose          3658 non-null   int32
 15  TenYearCHD       3658 non-null   int64
```

# WHAT ABOUT MISSING DATA

```
# missing data check
df.isnull().sum()
```

```
male                0
age                 0
education         105
currentSmoker       0
cigsPerDay         29
BPMeds             53
prevalentStroke     0
prevalentHyp        0
diabetes            0
totChol            50
sysBP               0
diaBP               0
BMI                19
heartRate           1
glucose           388
TenYearCHD          0
```

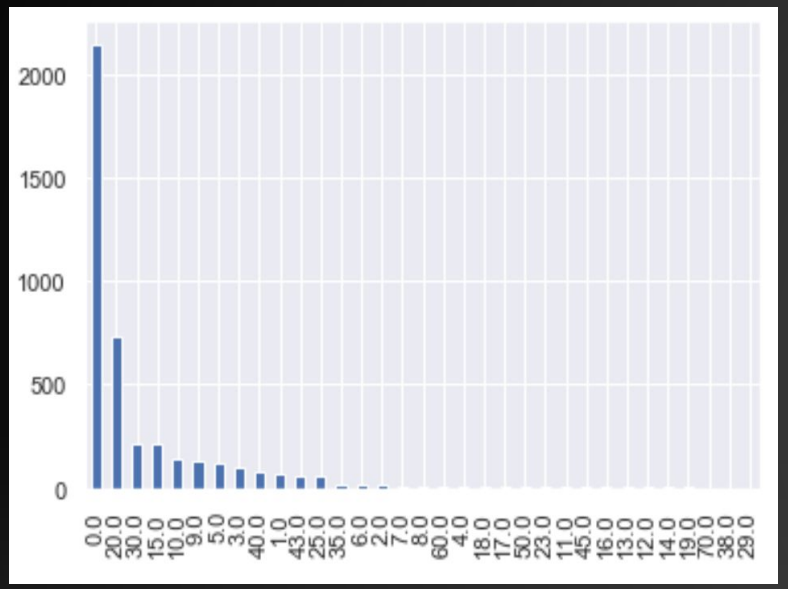
Using the isnull method, we can see that there is a high number of missing values for the variables education and glucose in relation to the other predictors with missing values. However, we are not using glucose due to its positively skewed distribution so its missing values are not an issue. Additionally, the missing values in the variable cigsPerDay are irrelevant because we will be removing that predictor as well. Since this dataset contains 4,238 observations, the missing data for the other predictors are not large enough to create a misconception of the analysis. Because of this, we decided to drop the rows that have missing values and only include the rows the rows that contain all of the appropriate data for our analysis.

```
## Removing missing value
# Drop rows which contain missing values
df.dropna(axis = 0, inplace = True)
df.isnull().sum()
```

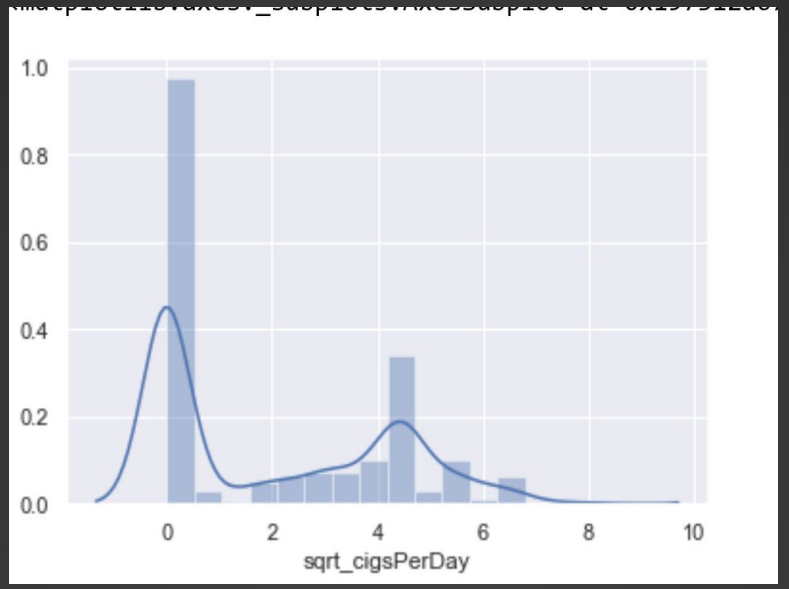
```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

# LOOKING INTO DATA TRANSFORMATIONS

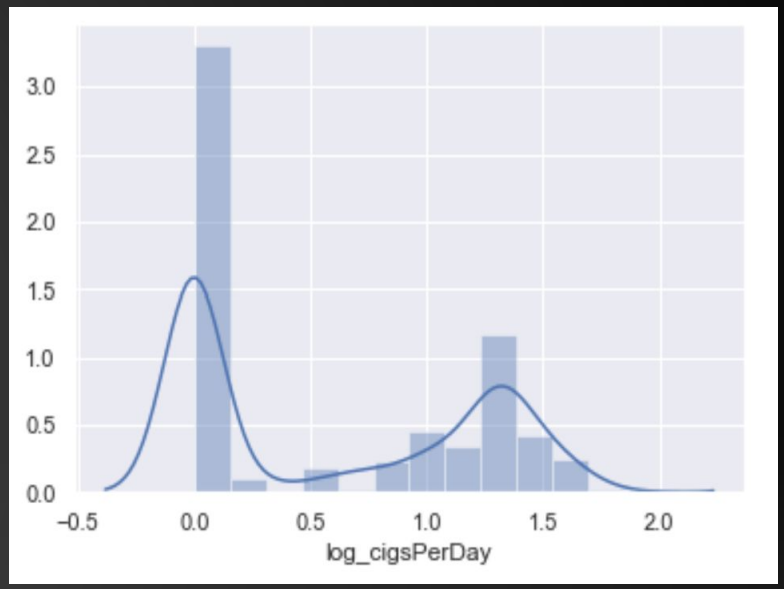
Original distribution of cigsPerDay.




Distribution after square root transformation



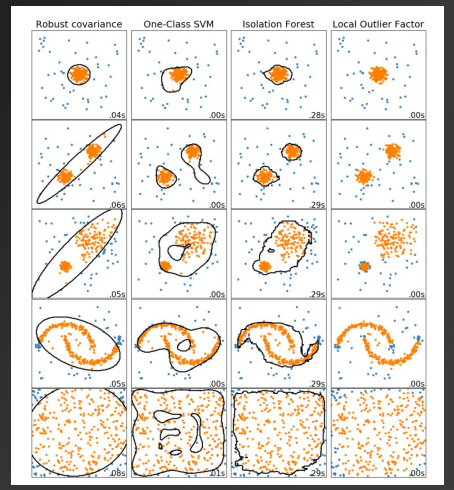

Distribution after log transformation.

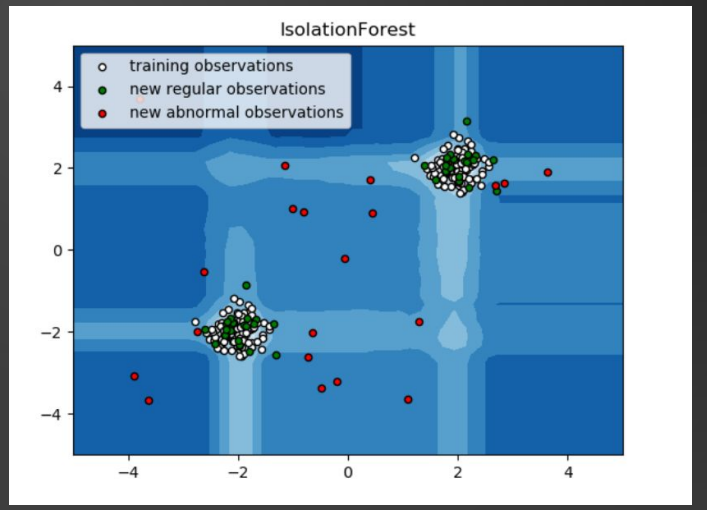



As stated in the previous slide, the variable cigsPerDay was severely right skewed. Even after attempting data transformation such as log and square root, the issue of skewness persists.

# Outlier Remove

- IsolationForest in sklearn is used to remove outliers
- Among 3,656 cases, 366 cases were removed. As a result, 3,290 cases were analyzed
- IsolationForest ‘isolates’ observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.



# Analysis

# SEPARATING TRAINING DATA AND VALIDATION DATA

Using a 70/30 split, we partitioned our data into a training data set consisting of 70% of our data and the remaining 30% of our data set was used as the validation data. The validation set will be used to validate the accuracy of the model developed by the training data.

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=2020)
```

# Logistic Regression Assumption

- Logistic regression does require that observations are independent of each other and that there is little collinearity among predictor variables and attention should be paid to outliers.
- Unlike linear regression, there is no normality assumption, nor constant variance assumption when conducting logistic regression analysis.
- Logistic model does not require that the dependent variable and the predictor variable be linearly related, it does require that the logit or log-odds is linearly related to the predictor variables.
- These assumptions listed above are from SAS Certification Prep Guide: Statistical Business Analysis Using SAS9

## Logistic regression assumptions

The logistic regression method assumes that:

- The outcome is a binary or dichotomous variable like yes vs no, positive vs negative, 1 vs 0.
- There is a linear relationship between the logit of the outcome and each predictor variables. Recall that the logit function is `logit(p) = log(p/(1-p))`, where p is the probabilities of the outcome (see Chapter @ref(logistic-regression)).
- There is no influential values (extreme values or outliers) in the continuous predictors
- There is no high intercorrelations (i.e. multicollinearity) among the predictors.

To improve the accuracy of your model, you should make sure that these assumptions hold true for your data. In the following sections, we'll describe how to diagnostic potential problems in the data.

These assumptions on the left are from STHDA (http://www.sthda.com/english/articles/36-classification-methods-essentials/148-logistic-regression-assumptions-and-diagnostics-in-r/#:~:text=The%20logistic%20regression%20method%20assumes%20that%3A%20The%20outcome,logit%20of%20the%20outcome%20and%20each%20predictor%20variables.

# Logistic Regression

Significant variables ($p < .05$): male, age, education, currentSmoker, and heartRate..
Non significant variables ($p < .05$): BPMeds, prevalentStroke, diabetes, totChol, sysBP, and BMI..

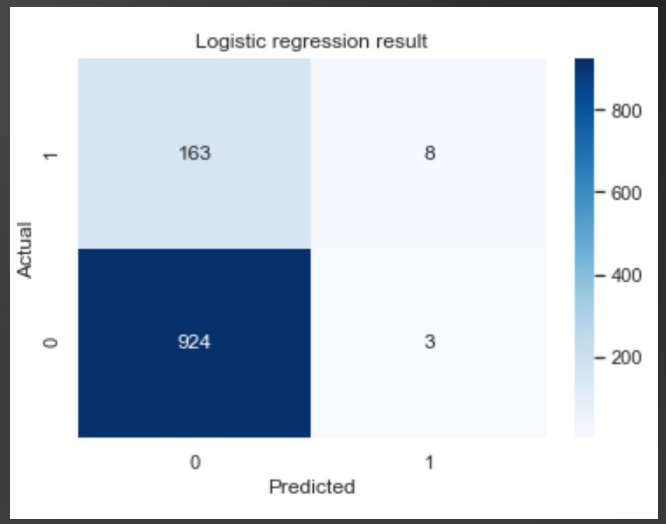
Logistic model's accuracy score is 0.90, which is slightly better than the accuracy score of the best model produced on Kaggle.

We found many false negatives. This likely means that the the logistic prediction model is not good at properly classifying if a patient is at risk of heart disease within the next 10 years (Type 2 error).

Logit Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | TenYearCHD | No. Observations: | 2303 |
| Model: | Logit | Df Residuals: | 2292 |
| Method: | MLE | Df Model: | 10 |
| Date: | Fri, 24 Jul 2020 | Pseudo R-squ.: | 0.07919 |
| Time: | 20:08:19 | Log-Likelihood: | -657.90 |
| converged: | False | LL-Null: | -714.47 |
| Covariance Type: | nonrobust | LLR p-value: | 1.232e-19 |

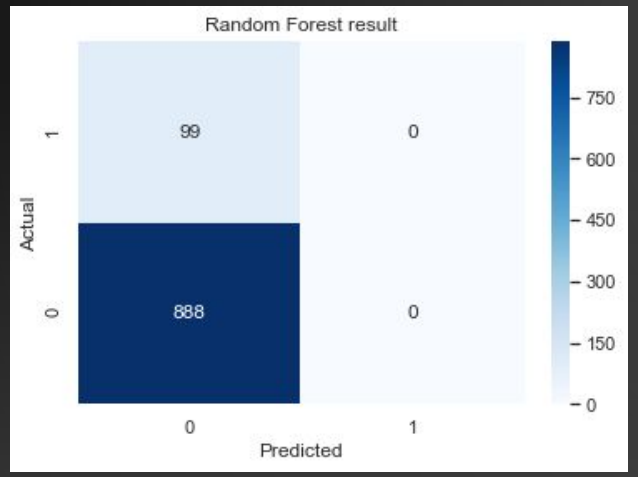
| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| male | 0.3141 | 0.153 | 2.053 | 0.040 | 0.014 | 0.614 |
| age | 0.0352 | 0.009 | 3.861 | 0.000 | 0.017 | 0.053 |
| education | -0.6401 | 0.091 | -7.000 | 0.000 | -0.819 | -0.461 |
| currentSmoker | 0.4999 | 0.157 | 3.186 | 0.001 | 0.192 | 0.807 |
| BPMeds | -24.2125 | 1.35e+05 | -0.000 | 1.000 | -2.65e+05 | 2.65e+05 |
| prevalentStroke | -12.7194 | 732.566 | -0.017 | 0.986 | -1448.523 | 1423.084 |
| diabetes | -18.5060 | 7836.803 | -0.002 | 0.998 | -1.54e+04 | 1.53e+04 |
| totChol | -0.0025 | 0.002 | -1.383 | 0.167 | -0.006 | 0.001 |
| sysBP | 0.0075 | 0.004 | 1.825 | 0.068 | -0.001 | 0.016 |
| BMI | -0.0301 | 0.019 | -1.559 | 0.119 | -0.068 | 0.008 |
| heartRate | -0.0397 | 0.007 | -6.113 | 0.000 | -0.052 | -0.027 |



# RandomForest Classification

Like the logistic regression, Random forest classification model also have good model performance (accuracy = .90), which is 2 point higher than Kaggle's best model (accuracy = .88)

However, this model also fails to find real patients with heart diseases (zero precision).




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.90 | 1.00 | 0.95 | 888 |
| 1 | 0.00 | 0.00 | 0.00 | 99 |
| accuracy | | | 0.90 | 987 |
| macro avg | 0.45 | 0.50 | 0.47 | 987 |
| weighted avg | 0.81 | 0.90 | 0.85 | 987 |

# Model Tuning

To get the best model, we tuned hyperparameters of RandomForest using RandomizedSearchCV method from sklearn.model_selection import

However, model tuning strategy does not help to get a better accuracy in Random Forest.

**Randomforest hyperparameter tune**

In [80]:
```
from sklearn.model_selection import RandomizedSearchCV
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
print(random_grid)
```

```
{'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000], 'max_features': ['auto', 'sqrt'], 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'bootstrap': [True, False]}
```

In [81]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid, n_iter =
# Fit the random search model
rf_random.fit(x_train, y_train)
```

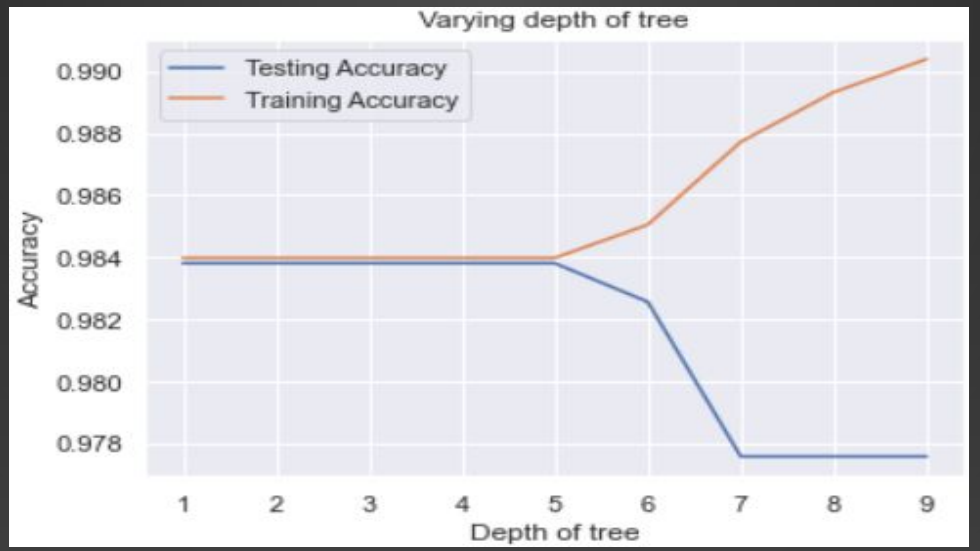
```
Fitting 3 folds for each of 100 candidates, totalling 300 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done  17 tasks      | elapsed:   12.2s
[Parallel(n_jobs=-1)]: Done 138 tasks      | elapsed:  1.2min
[Parallel(n_jobs=-1)]: Done 300 out of 300 | elapsed:  2.5min finished
```

```
Model Performance
Average Error: 0.0165 degrees.
Accuracy = 82.33%.
Model Performance
Average Error: 0.0851 degrees.
Accuracy = 8.84%.
Improvement of -89.27%.
```

# Depth of Tree

We decided to experiment with finding the best max depth of in our random forest. We chose a max depth of 5 because based on the graph below, we can see that we begin to lose accuracy as the tree depth gets larger than 5.

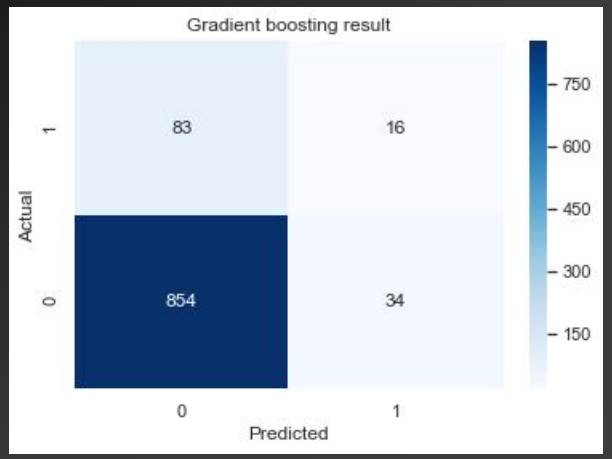


# Gradient Boosting Classification

Decided to use Gradient Boosting because this is one of the most advanced decision tree based methods. On Kaggle, this method have been one of the most popular except deep learning (Sebastian, 2017).

The result shows that Gradient Boosting show highest precision (.32) and good accuracy (.88). Unlike logistic regression and Randomforest, it is practically a good method to detect heart disease patients.

We changed of hyperparameters (e.g., max_depth and min_child_weight) hundreds times to find the best gradient boosting models. We found that eta = 1, max_depth = 5, and min-child_weight = 6 are the best in terms of accuracy and precision.




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.91 | 0.96 | 0.94 | 888 |
| 1 | 0.32 | 0.16 | 0.21 | 99 |
| accuracy | | | 0.88 | 987 |
| macro avg | 0.62 | 0.56 | 0.58 | 987 |
| weighted avg | 0.85 | 0.88 | 0.86 | 987 |

# Conclusion

- Beat Kaggle Best Accuracy Scores
  - The current best Kaggle accuracy is 0.88 and precision is .001. But all our models beat the accuracy, and gradient boosting's precision is .32, which is much higher than the current Kaggle result.
- Three models comparison
  - Although logistic regression and RandomForest were better than regarding accuracy (.90), gradient boosting was better to detect patients (high precision). Practically, we think gradient boosting is the best model
- Type 2 error (false negative) are problem
  - But all other models in Kaggle suffer from the same issue. To solve problems, we need other predictors to detect patients.

# Reference

Sebastian, R (2017) Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, Packt Publishing.

https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings

https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings
http://www.fairlynerdy.com/what-is-r-squared/