Heart Attack Prediction

using Linear Regression

Presented by

Arnik Vephasayanant	65056099
Tosspol Kanokpipatwong	65056040
Nuttawut Tongseenoon	65056036
Puriwat Sangrawee	65056071

Prologue

- Using the collected data to determine the future potential patient which likely to have a heart attack.
- Source of the data is from Kaggle.
- https://www.kaggle.com/datasets/mokar2001/ascvd-heart-risk
- Will be using Linear Regression to forecast the trend and see how reliable is this dataset.

Data Features & Description

- The Dataset contains 1000 rows of records and 10 Columns.

- Feature will be represented as followed:
  a. isMale
  b. isBlack
  c. isSmoker
  d. isDiabetic
  e. isHypertensive
  f. Age
  g. Systolic
  h. Cholesterol
  i. HDL
  j. Risk

The dataset contain 1000 rows and 10 columns.

Data Description

Feature	Description
isMale	Female or Male
isBlack	Patient's Race
isSmoker	Have a Record of Smoking
isDiabetic	Have a Record of Diabetic
isHypertensive	Have a Record of Hypertensive
Age	Patient's Age
Systolic	Blood Pressure
Cholesterol	Have a Record of Cholesterol
HDL	High-density lipoprotein
Risk	Potential Heart Attack

Data Overview

```
df.head()
```

	isMale	isBlack	isSmoker	isDiabetic	isHypertensive	Age	Systolic	Cholesterol	HDL	Risk
0	1	1	0	1	1	49	101	181	32	11.1
1	0	0	0	1	1	69	167	155	59	30.1
2	0	1	1	1	1	50	181	147	59	37.6
3	1	1	1	1	0	42	145	166	46	13.2
4	0	0	1	0	1	66	134	199	63	15.1

- Nominal : isMale, isBlack, isSmoker, isDiabetic, isHypertensive
- Ratio : Age
- Interval : Systolic, Cholesterol, HDL, Risk

Data Info

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   isMale          1000 non-null   int64  
 1   isBlack         1000 non-null   int64  
 2   isSmoker        1000 non-null   int64  
 3   isDiabetic      1000 non-null   int64  
 4   isHypertensive  1000 non-null   int64  
 5   Age             1000 non-null   int64  
 6   Systolic        1000 non-null   int64  
 7   Cholesterol     1000 non-null   int64  
 8   HDL             1000 non-null   int64  
 9   Risk            1000 non-null   float64
dtypes: float64(1), int64(9)
memory usage: 78.2 KB
```

- Data types : int64 and float64
- In this case of the data, there is no null record in the data.

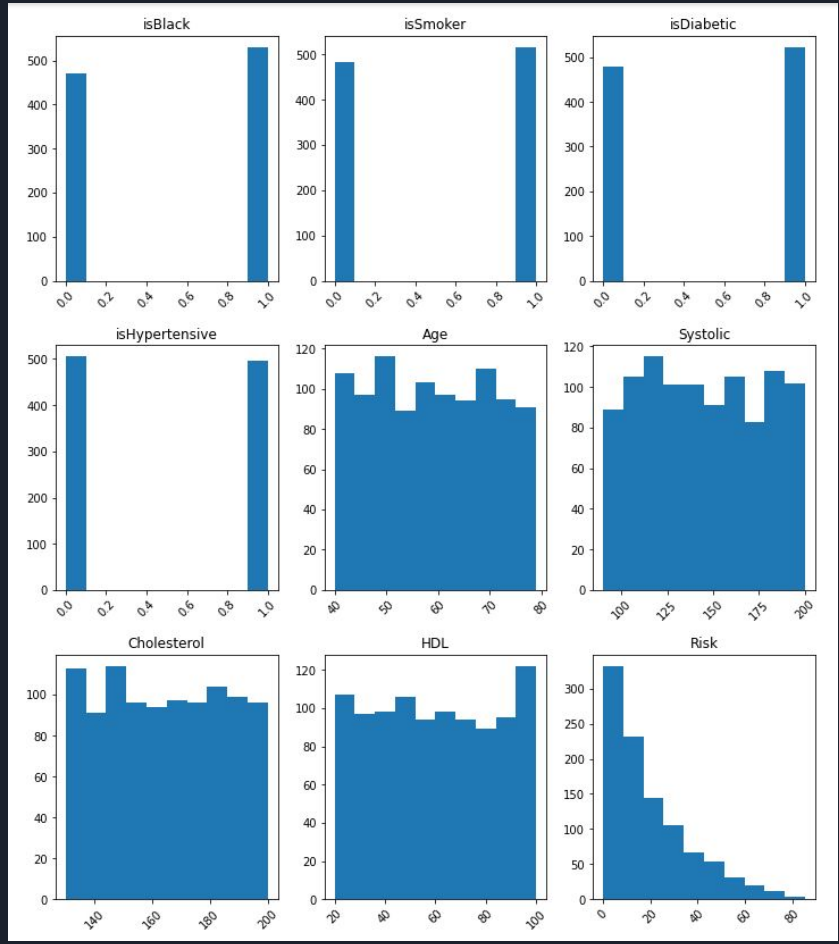
Data Overview in Statistic

```
df.describe()
```

	isMale	isBlack	isSmoker	isDiabetic	isHypertensive	Age	Systolic	Cholesterol	HDL	Risk
count	1000.00000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.49000	0.530000	0.516000	0.522000	0.495000	59.107000	144.249000	164.043000	59.603000	19.667000
std	0.50015	0.499349	0.499994	0.499766	0.500225	11.536492	31.774528	20.329891	23.863505	17.043941
min	0.00000	0.000000	0.000000	0.000000	0.000000	40.000000	90.000000	130.000000	20.000000	0.100000
25%	0.00000	0.000000	0.000000	0.000000	0.000000	49.000000	117.000000	146.000000	39.000000	6.300000
50%	0.00000	1.000000	1.000000	1.000000	0.000000	59.000000	144.000000	164.000000	59.000000	14.400000
75%	1.00000	1.000000	1.000000	1.000000	1.000000	69.000000	171.000000	182.000000	81.000000	29.000000
max	1.00000	1.000000	1.000000	1.000000	1.000000	79.000000	200.000000	200.000000	100.000000	85.400000

Inspect Data

- To check for population in each variables to see all the data in general.
- From most Nominal Variables, the dataset that was collected seem to present a balance dataset between 0 and 1.



Cleansing the Data

- In this process, the data sometimes contains null, incorrect, corrupted, incorrected format, duplicate or incomplete value.
- This needs to be fixed else it will affect the result of the prediction.
- Fortunately for this dataset, there is no null value.

```
df.isnull().sum()
```

```
isMale             0
isBlack            0
isSmoker           0
isDiabetic         0
isHypertensive     0
Age                0
Systolic           0
Cholesterol        0
HDL                0
Risk               0
dtype: int64
```

Visualize Data

- In order to understand what dataset is represented. We need to visualize the dataset and to give a better understanding of correlation between each variables.

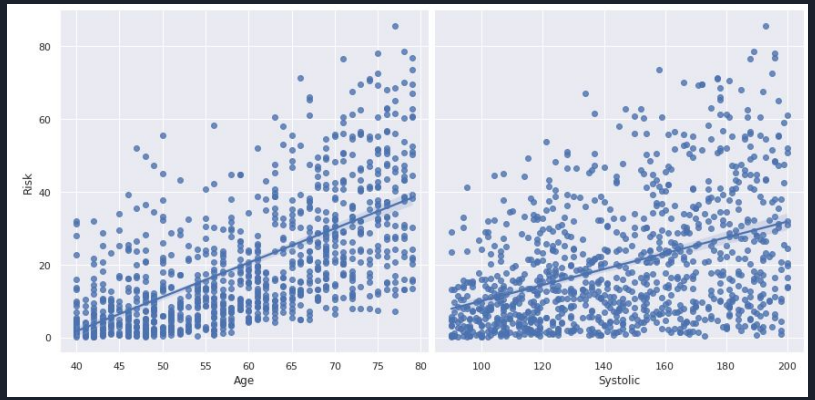
- In this part, we use heatmap to represent all the correlation between the variables.

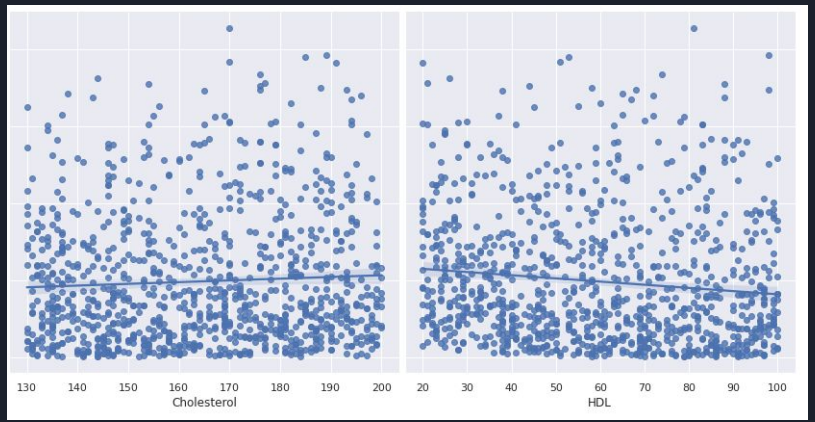
Heatmap

- As shown in the heatmap
  - isSmoker,
  - isDiabetic,
  - Age
  - Systolic
- Those are likely to have correlation with the Risk of getting Heart Attack.

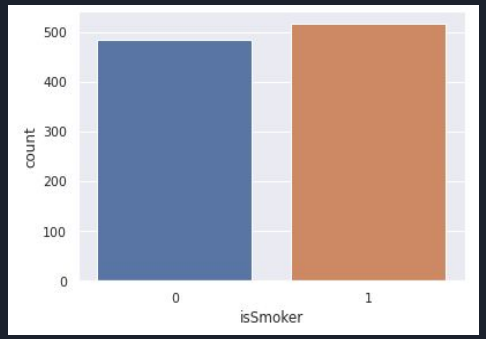
Correlation Plotting

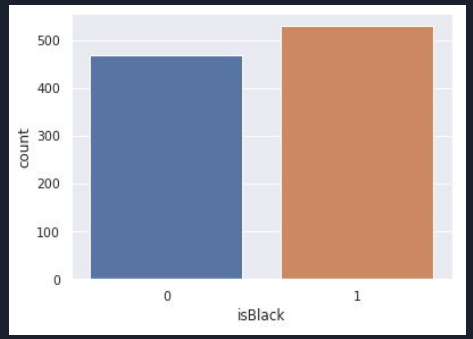
- As a result according to the heatmap.
    a. Older patient has higher risk of getting heart attack.
    b. Higher Systolic has higher risk of getting heart attack.
    c. Cholesterol is not clear enough to determine the risk of getting heart attack.
    d. HDL is not clear enough to determine the risk of getting heart attack.

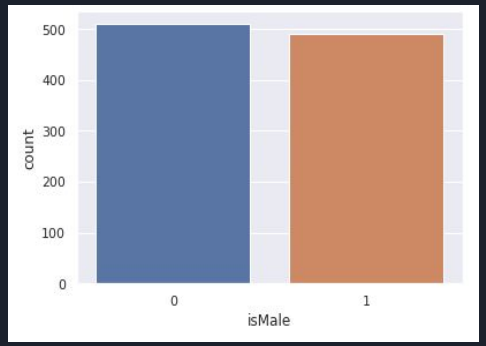


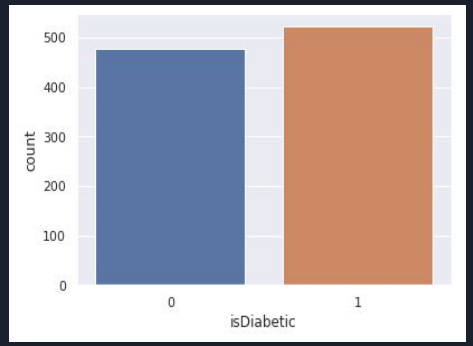


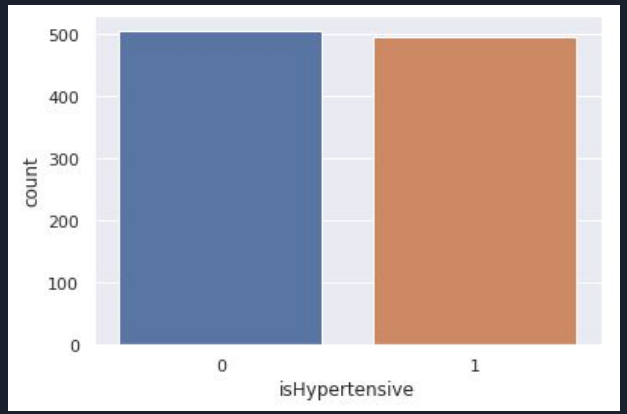
Nominal Level by Frequency



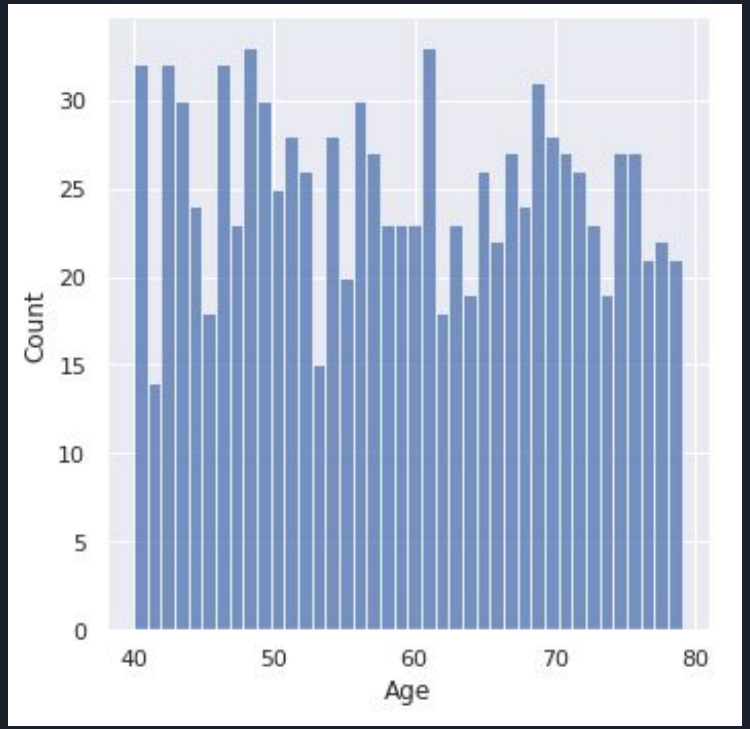




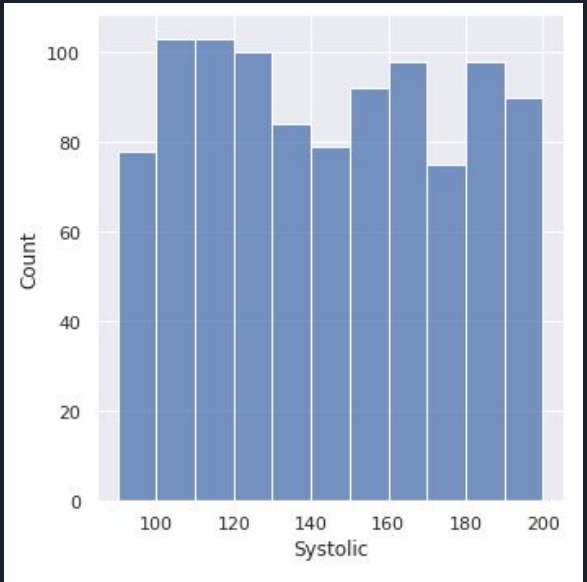


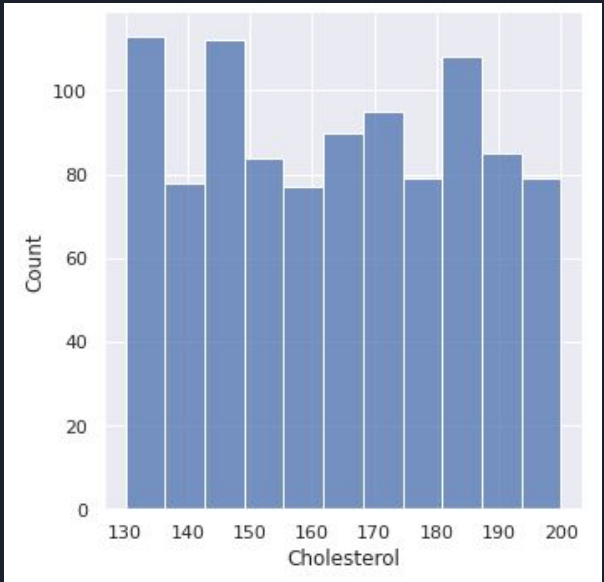


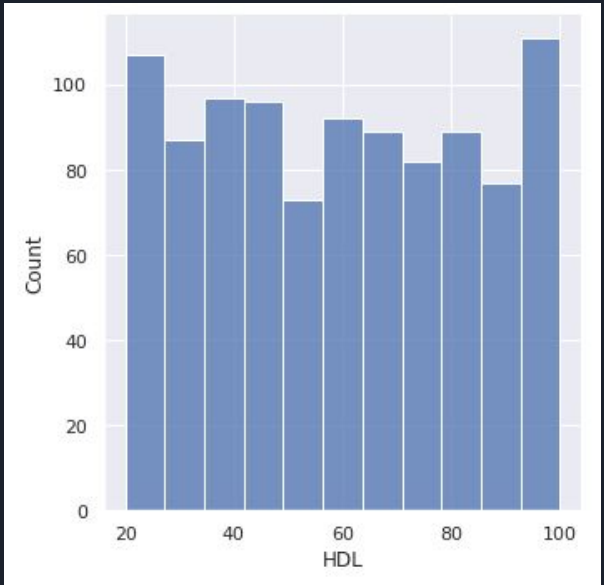
Ratio Level by Frequency



Interval Level by Frequency







Data Preprocessing

- In this step, the data will be prepared and used in the processing step.
- The data will be splitted into 70/30
  - 70% for Train Dataset
  - 30% for Test Dataset
- In this part, Linear Regression will be used to find statistical data such as Coefficient, Intercept, Mean Absolute Error, Root Mean Square Error and R-Square Score

Data Split

Split Data

```
[ ]  X = df.drop('Risk', axis=1)
     y = df['Risk']
```

Perform Split Data into Train and Test by 70/30

```
[ ]  from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3)
```

Check Size

```
[ ]  X_train.shape, y_train.shape
```

```
     ((700, 9), (700,))
```

```
[ ]  X_test.shape, y_test.shape
```

```
     ((300, 9), (300,))
```

Linear Regression Technique

```
[ ]  from sklearn.linear_model import LinearRegression

     model = LinearRegression()
     model.fit(X_train, y_train)
     y_pred = model.predict(X_test)
```

Evaluation

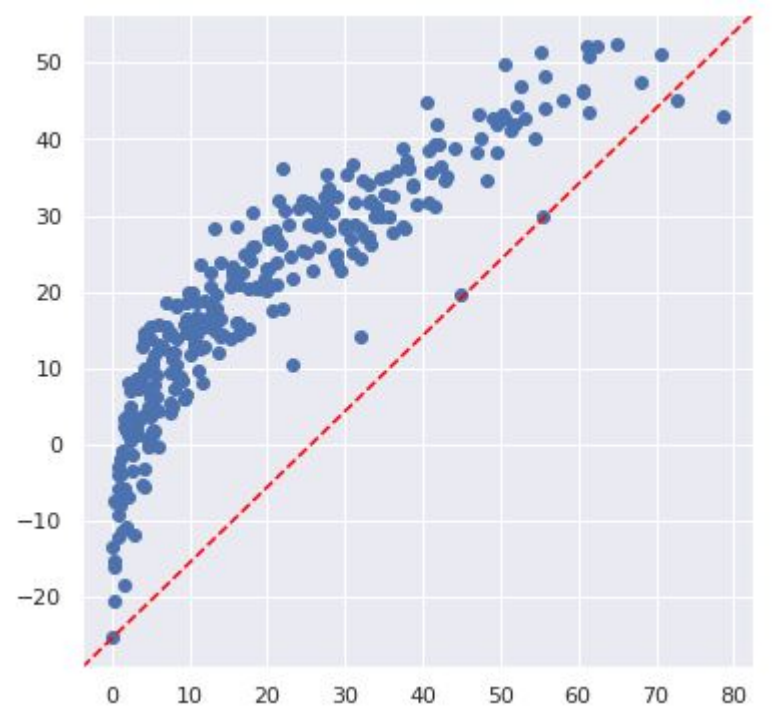
```
[ ]  from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

     lr_coef = model.coef_
     lr_intercept = model.intercept_
     lr_mae = mean_absolute_error(y_test, y_pred)
     lr_rmse = np.sqrt(mean_squared_error(y_test, y_pred))
     lr_r2 = r2_score(y_test, y_pred)

     print("Coefficients : ", lr_coef)
     print("Intercept : ", lr_intercept)
     print("mean_absolute_error : ", lr_mae)
     print("root_mean_squared_error : ", lr_rmse)
     print("r2_score : ", lr_r2)
```

```
Coefficients :  [ 4.75472723  2.73379333  9.19932881 10.45678774  5.03455277  0.92756593
  0.20384614  0.06384438 -0.0962107 ]
Intercept :  -86.06575081161874
mean_absolute_error :  5.890036613574889
root_mean_squared_error :  7.749742263697786
r2_score :  0.7988785207491148
```

```
[ ]  r2_score(y_test, y_pred)
```

```
0.7988785207491148
```

Linear Regression Model in Graph

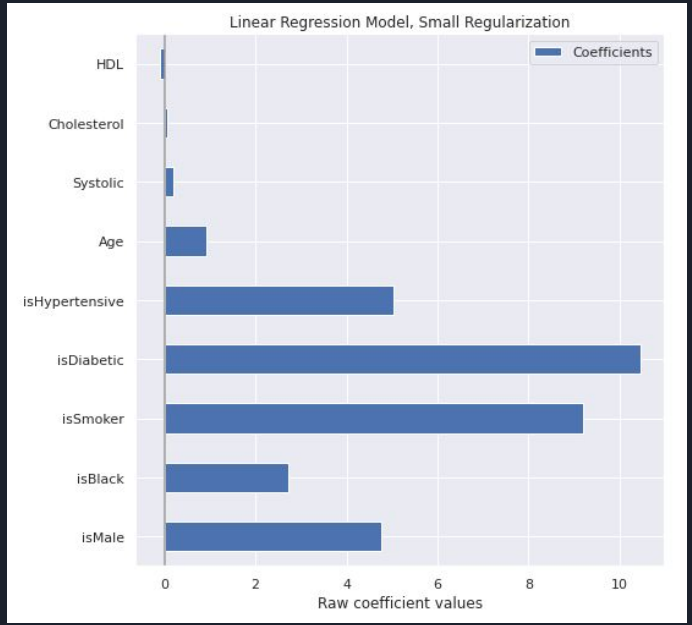


- By using Linear Regression Model. The graph shows positively correlation of getting heart attack.
- And with the calculation; 79.88% of the sums of squares of the overall variable can be explained by Risk of getting Heart Attack
- Graph is quite over predicted which in medical field, this might considered to be reliable enough to use this prediction to find the patient who will potentially has a heart attack.

Raw Coefficient

```
[ ] t1 = df.drop('Risk',axis=1)

    t1.head().columns

    Index(['isMale', 'isBlack', 'isSmoker', 'isDiabetic', 'isHypertensive', 'Age',
           'Systolic', 'Cholesterol', 'HDL'],
          dtype='object')

[ ] feature_names = t1.columns

    coefs = pd.DataFrame(lr_coef,columns=['Coefficients'],index=feature_names)

    coefs
```

	Coefficients
isMale	4.754727
isBlack	2.733793
isSmoker	9.199329
isDiabetic	10.456788
isHypertensive	5.034553
Age	0.927566
Systolic	0.203846
Cholesterol	0.063844
HDL	-0.096211



- As shown in this graph. This shows that Diabetic and Smoker are also the main source of causing heart attack.

OLS Regression Results

OLS Regression Results

Dep. Variable:	Risk	R-squared (uncentered):	0.825
Model:	OLS	Adj. R-squared (uncentered):	0.824
Method:	Least Squares	F-statistic:	520.2
Date:	Tue, 27 Sep 2022	Prob (F-statistic):	0.00
Time:	12:50:30	Log-Likelihood:	-3805.4
No. Observations:	1000	AIC:	7629.
Df Residuals:	991	BIC:	7673.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
isMale	3.1744	0.689	4.606	0.000	1.822	4.527
isBlack	1.2644	0.693	1.824	0.069	-0.096	2.625
isSmoker	7.1207	0.690	10.322	0.000	5.767	8.474
isDiabetic	9.3697	0.692	13.533	0.000	8.011	10.728
isHypertensive	3.4342	0.689	4.981	0.000	2.081	4.787
Age	0.6330	0.027	23.267	0.000	0.580	0.686
Systolic	0.1066	0.010	10.798	0.000	0.087	0.126
Cholesterol	-0.2145	0.011	-18.775	0.000	-0.237	-0.192
HDL	-0.1635	0.014	-11.474	0.000	-0.191	-0.136

Omnibus:	147.868	Durbin-Watson:	2.065
Prob(Omnibus):	0.000	Jarque-Bera (JB):	248.298
Skew:	0.948	Prob(JB):	1.21e-54
Kurtosis:	4.539	Cond. No.	491.

Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Thank you