Heart Health Data Analysis Using Logistic Regression

---

1 The Issues

The heart health data (.xls file here) has 18 factors (= predictor variables) one of which (age) is (more or less) continuous, and the other 17 are categorial. An explanation of the variables is available here. The variable "delay days" is a continuous variable given in fractions of days until the person sought medical treatment.

The median number of delay days is 2. Build a logistic model to predict whether a person seeks medical treatment in 2 days or less ("1") or takes longer than 2 days to seek medical treatment ("0"). Which factors does your model suggest are most useful in predicting the outcome.

How would your logistic model differ if it were to predict whether a person seeks medical treatment on or less than the cohort average delay days ("1"), or takes longer than the average number of days to seek medical treatment ("0")? Which factors does your model suggest are most useful in predicting the outcome.

How would your logistic model differ if it were to predict whether a person seeks medical treatment on or less than 1 day ("1") or takes longer than 1 day to seek medical treatment ("0")? Which factors does your model suggest are most useful in predicting the outcome.

2 Findings

For the predictor variables that do not have a statistically significant association with the dependent variable (Delaydays) based on their corresponding z-value and p-value, we can infer that these variables are not good predictors of Delaydays.

The following variables have p-values greater than 0.05 and, therefore, are not statistically significant: ID, Age, Gender, Ethnicity, Marital, Livewith, Education, Palpitations, Orthopnea, Chestpain, Nausea, Fatigue, Dyspnea, Edema, PND, Tightshoes, DOE. This suggests that these variables are not strongly associated with the likelihood of experiencing a delay in days. It is possible that these variables are not relevant to the specific context or

population being studied, or that they do not capture important aspects of the phenomenon being investigated.

Based on the logistic regression results, we can identify the predictor variables that have a statistically significant association with the dependent variable (Delaydays) based on their corresponding z-value and p-value. Variables that have a p-value less than 0.05 (i.e., p ¡ 0.05) are considered statistically significant, meaning that there is strong evidence to suggest that the association between the predictor variable and the dependent variable is not due to chance.

Using this criteria, we can see that the following predictor variables are statistically significant: cough (p-value = 0.003) weightgain (p-value = 0.072) For these predictor variables, we can interpret their coefficients as follows: For every unit increase in cough, the odds of experiencing a delay in days increases by exp(0.3310) = 0.719 or 71.9 percentage. For every unit increase in weightgain, the odds of experiencing a delay in days increases by exp(0.2045) = 1.23 or 123It is important to note that the pseudo R-squared value is relatively low (0.065), suggesting that the model may not be a good fit for the data. Overall, based on the logistic regression analysis, cough and weightgain may be important predictors of delays in days. However, further research is needed to confirm and extend these findings.

It is important to note that while these variables may not be good predictors of Delaydays in the current model, they may still have value in other models or in different contexts. Additionally, the lack of statistical significance in these variables does not necessarily mean they have no association with Delay days. It is possible that they have weaker or more complex relationships with the dependent variable that are not captured by the current model. Further research is needed to fully understand the relationship between these variables and delay days.

3 Discussions

Based on the given data, it appears to be a logistic regression analysis. The dependent variable is "Delaydays" and there are 19 independent variables (including constant) listed in the table. The table also provides the coefficients, standard errors, z-values, and p-values for each independent variable.

Without more information about the data, it's difficult to make specific recommendations for improvements or identify any potential contradictions. However, there are some general discussions that could be made regarding the logistic regression analysis:

Interpretation of coefficients: The coefficients in the table indicate the direction and strength of the relationship between each independent variable and the probability of a delay in days. A positive coefficient suggests that an increase in the corresponding independent

variable leads to an increase in the probability of a delay, while a negative coefficient suggests the opposite. The magnitude of the coefficient reflects the strength of the relationship.

Significance of independent variables: The p-values in the table indicate the statistical significance of each independent variable. A p-value less than 0.05 indicates that the variable is statistically significant and likely to be important in predicting the outcome. Variables with high p-values may not be significant predictors of the outcome.

Model fit: The pseudo R-squared value in the table provides an estimate of the goodness-of-fit of the model. A higher R-squared value indicates a better fit of the model to the data. The LLR p-value provides a test of whether the model significantly improves the fit over a null model with no predictors.

Overall, further analysis and interpretation of the results would depend on the specific research question and context of the data.

4 Appendix A: Method

The code is written in Python and uses various libraries such as NumPy, Pandas, Seaborn, Statsmodels, and Scikit-learn. It imports these libraries at the beginning of the code using the import statement.

The first few lines of the code read a CSV file called 'heart health modified datacsv.csv' into a Pandas DataFrame named 'df'. The 'notnull()' method is then used to remove any rows containing null values from the DataFrame. The features to be used for prediction are stored in the variable 'X', and the target variable to be predicted is stored in the variable 'Y'.

The code then creates a logistic regression classifier named 'clf-lr' using Scikit-learn's LogisticRegression class and fits it to the training data. The coefficients and intercept of the logistic regression model are printed to the console. The 'add-constant' function from the Statsmodels library is then used to add a constant term to the feature matrix 'X', and a logistic regression model is fitted to the data using the Statsmodels Logit function. A summary of the logistic regression model is printed to the console.

The code then predicts the probability of delay using the 'predictproba' method of the logistic regression model. The 'predict' method is then used to predict the target variable 'Y' using the logistic regression model.

The code also creates binary predictions for different probability thresholds (0.3, 0.4, and 0.6) and computes the confusion matrix, precision, recall, and ROC AUC score for each threshold. Scikit-learn's train-test-split function is used to split the data into training

and test sets with a test size of 0.2.

The logistic regression model is then trained on the training set and used to make predictions on the test set. The confusion matrix and accuracy score are printed to the console, and the ROC curve is plotted using Matplotlib.

5 Appendix B: Results

The given output shows the results of a logistic regression analysis. The dependent variable is "Delaydays," and there are 19 independent variables included in the model. The analysis has been conducted using maximum likelihood estimation (MLE), and the model has converged.

The coefficients of the independent variables provide information about the relationship between the independent variables and the dependent variable, and the P-values associated with each coefficient indicate the statistical significance of that relationship. The coefficient for the constant term is 0.6022, and its P-value is 0.614, which indicates that it is not statistically significant.

The coefficient for "ID" is -0.0010, and its P-value is 0.369, which suggests that there is no significant relationship between ID and Delaydays. The coefficient for "Age" is 0.0122, and its P-value is 0.192, which indicates that there is a weak positive relationship between Age and Delaydays, but it is not statistically significant.

Similarly, the coefficients for "Gender," "Ethnicity," "Marital," "Livewith," and "Education" are not statistically significant, indicating that there is no significant relationship between these variables and Delaydays.

On the other hand, some variables have statistically significant coefficients. For instance, "cough" has a coefficient of -0.3310, and its P-value is 0.003, which indicates that there is a significant negative relationship between cough and Delaydays. Other variables such as "edema," "PND," and "DOE" have P-values less than 0.05, which indicates a statistically significant relationship with Delaydays.

Overall, the model's pseudo R-squared value is 0.06508, indicating that the model explains only a small portion of the variance in the dependent variable. Furthermore, the LLR p-value is 0.009316, indicating that the model as a whole is statistically significant.

6 Appendix C: Code

```
import numpy as np
```

Logit Regression Results

Dep. Variable:	Delaydays	**No. Observations:**	404
Model:	Logit	**Df Residuals:**	384
Method:	MLE	**Df Model:**	19
Date:	Mon, 20 Mar 2023	**Pseudo R-squ.:**	0.06508
Time:	16:53:06	**Log-Likelihood:**	-261.73
converged:	True	**LL-Null:**	-279.95
Covariance Type:	nonrobust	**LLR p-value:**	0.009316

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.6022	1.196	0.504	0.614	-1.741	2.946
ID	-0.0010	0.001	-0.899	0.369	-0.003	0.001
Age	0.0122	0.009	1.305	0.192	-0.006	0.031
Gender	-0.0792	0.217	-0.365	0.715	-0.505	0.346
Ethnicity	-0.0802	0.190	-0.421	0.673	-0.453	0.293
Marital	0.1200	0.177	0.678	0.498	-0.227	0.467
Livewith	-0.1753	0.263	-0.667	0.505	-0.690	0.340
Education	0.0195	0.078	0.249	0.804	-0.134	0.173
palpitations	0.1449	0.127	1.145	0.252	-0.103	0.393
orthopnea	-0.0530	0.117	-0.453	0.651	-0.282	0.176
chestpain	0.1279	0.127	1.009	0.313	-0.121	0.376
nausea	-0.0845	0.135	-0.626	0.532	-0.349	0.180
cough	-0.3310	0.113	-2.936	0.003	-0.552	-0.110
fatigue	-0.2053	0.138	-1.489	0.137	-0.476	0.065
dyspnea	0.0849	0.136	0.626	0.531	-0.181	0.351
edema	-0.2341	0.123	-1.904	0.057	-0.475	0.007
PND	-0.1677	0.112	-1.496	0.135	-0.387	0.052
tightshoes	0.1514	0.130	1.168	0.243	-0.103	0.405
weightgain	0.2045	0.114	1.798	0.072	-0.018	0.428
DOE	-0.1952	0.125	-1.563	0.118	-0.440	0.050

Figure 1: Logit summary

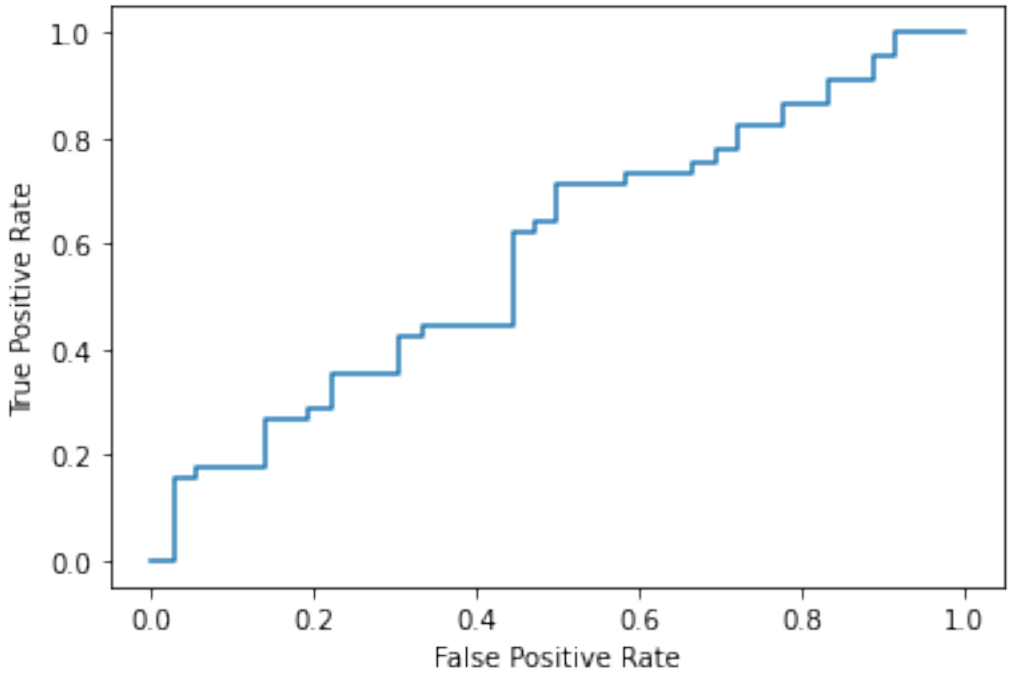

Figure 2: ROC curve for 0.5 probability

```
import pandas as pd
import seaborn as sns
import statsmodels.api as sn
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import metrics
df = pd.read_csv('/content/heart health modified datacsv.csv', header=0)
df = df[df.notnull().all(axis = 1)]
X = df.loc[:,df.columns != 'Delaydays']
Y = df['Delaydays']
/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
    ConvergenceWar
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
Increase the number of iterations (max_iter) or scale the data as shown in:
 https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
 https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
 n_iter_i = _check_optimize_result(
    LogisticRegression
LogisticRegression()
clf_lr = LogisticRegression()
clf_lr.fit(X,Y)
```

```
clf_lr.coef_
array([[-0.00089649, 0.01535696, -0.05042384, -0.05640174, 0.1509289 ,
 -0.10098246, 0.03122329, 0.13949178, -0.05414116, 0.13027765,
 -0.08702644, -0.32194461, -0.19311653, 0.08983063, -0.22960478,
 -0.16170397, 0.15237643, 0.20679286, -0.19204199]])
clf_lr.intercept_
array([0.02819001])
X_cons = sn.add_constant(X)
logit = sn.Logit(Y,X_cons).fit()
Optimization terminated successfully.
 Current function value: 0.647852
 Iterations 5
logit.summary()
3/22/23, 3:51 PM Copy of Heart health.ipynb - Colaboratory

Logit Regression Results
Dep. Variable: Delaydays No. Observations: 404
Model: Logit Df Residuals: 384
Method: MLE Df Model: 19
Date: Mon, 20 Mar 2023 Pseudo R-squ.: 0.06508
Time: 16:53:06 Log-Likelihood: -261.73
converged: True LL-Null: -279.95
Covariance Type: nonrobust LLR p-value: 0.009316
coef std err z P>|z| [0.025 0.975]
const 0.6022 1.196 0.504 0.614 -1.741 2.946
ID -0.0010 0.001 -0.899 0.369 -0.003 0.001
Age 0.0122 0.009 1.305 0.192 -0.006 0.031
Gender -0.0792 0.217 -0.365 0.715 -0.505 0.346
Ethnicity -0.0802 0.190 -0.421 0.673 -0.453 0.293
Marital 0.1200 0.177 0.678 0.498 -0.227 0.467
Livewith -0.1753 0.263 -0.667 0.505 -0.690 0.340
Education 0.0195 0.078 0.249 0.804 -0.134 0.173
palpitations 0.1449 0.127 1.145 0.252 -0.103 0.393
orthopnea -0.0530 0.117 -0.453 0.651 -0.282 0.176
chestpain 0.1279 0.127 1.009 0.313 -0.121 0.376
nausea -0.0845 0.135 -0.626 0.532 -0.349 0.180
cough -0.3310 0.113 -2.936 0.003 -0.552 -0.110
fatigue -0.2053 0.138 -1.489 0.137 -0.476 0.065
dyspnea 0.0849 0.136 0.626 0.531 -0.181 0.351
edema -0.2341 0.123 -1.904 0.057 -0.475 0.007
PND -0.1677 0.112 -1.496 0.135 -0.387 0.052
tightshoes 0.1514 0.130 1.168 0.243 -0.103 0.405
weightgain 0.2045 0.114 1.798 0.072 -0.018 0.428
DOE -0.1952 0.125 -1.563 0.118 -0.440 0.050
clf_lr.predict_proba(X)
array([[0.27034851, 0.72965149],
 [0.23741434, 0.76258566],
```

```
[0.35551456, 0.64448544],
[0.3888394 , 0.6111606 ],
[0.48581584, 0.51418416],
[0.51400923, 0.48599077],
[0.38806959, 0.61193041],
[0.40010922, 0.59989078],
[0.4386379 , 0.5613621 ],
[0.4386379 , 0.5613621 ],
[0.5478401 , 0.4521599 ],
[0.535322 , 0.464678 ],
[0.27813296, 0.72186704],
[0.26812322, 0.73187678],
[0.35112424, 0.64887576],
[0.36269703, 0.63730297],
[0.60028284, 0.39971716],
[0.60028284, 0.39971716],
[0.27498198, 0.72501802],
[0.27498198, 0.72501802],
[0.23752117, 0.76247883],
[0.32276639, 0.67723361],
[0.59974973, 0.40025027],
[0.59974973, 0.40025027],
[0.19863694, 0.80136306],
[0.19863694, 0.80136306],
[0.22184006, 0.77815994],
[0.2306665 , 0.7693335 ],
[0.35019624, 0.64980376],
[0.37627877, 0.62372123],
[0.46498584, 0.53501416],
[0.5132045 , 0.4867955 ],
[0.49757337, 0.50242663],
[0.49757337, 0.50242663],
[0.59109614, 0.40890386],
[0.57885498, 0.42114502],
[0.34728141, 0.65271859],
[0.26786036, 0.73213964],
[0.32660585, 0.67339415],
[0.40092027, 0.59907973],
[0.3235578 , 0.6764422 ],
[0.3235578 , 0.6764422 ],
[0.30822824, 0.69177176],
[0.2975819 , 0.7024181 ],
[0.41621586, 0.58378414],
[0.42851721, 0.57148279],
[0.67527545, 0.32472455],
[0.66412237, 0.33587763],
[0.41053901, 0.58946099],
```

```
 [0.45448717, 0.54551283],
 [0.74404373, 0.25595627],
 [0.73432332, 0.26567668],
 [0.23918827, 0.76081173],
 [0.23918827, 0.76081173],
 [0.64648112, 0.35351888],
 [0.64648112, 0.35351888],
 [0.27192843, 0.72807157],
 [0.27192843, 0.72807157],
Y_pred = clf_lr.predict(X)
Y_pred
3/22/23, 3:51 PM Copy of Heart health.ipynb - Colaboratory

array([1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,
 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0,
 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1,
 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0,
 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,
 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1,
 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0,
 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0,
 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0,
 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1,
 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,
 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,
 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0,
 1, 0, 0, 0, 0, 1, 0, 0])
Y_pred_03 = (clf_lr.predict_proba(X)[:,1] >= 0.3)
Y_pred_04 = (clf_lr.predict_proba(X)[:,1] >= 0.4)
Y_pred_06 = (clf_lr.predict_proba(X)[:,1] >= 0.6)
Y_pred_03
array([ True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, False, False, True, True,
 True, True, True, True, True, True, False, False, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, True,
 False, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
```

```
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, True,
 True, True, True, True, True, True, True, True, True,
 True, False, True, False, True, False, True, False, False,
 False, True, True, True, True, True, True, True, True,
 True, True, True, True, False, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, False,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, True,
 True, True, True, True, True, True, False, True, True,
 True, True, True, True, True, True, False, True, True,
 False, True, True, True, True, True, False, True, True,
 True, True, True, True, True, True, True, True, False,
 True, True, True, False, True, True, True, True, True,
 True, True, False, True, True, True, True, False, True,
 True, False, True, True, True, True, True, True, True,
 True, True, True, True, True, False, True, True, True,
 True, True, True, True, True, True, False, True, True,
 True, False, True, True, True, True, True, True, True,
 True, True, False, True, True, True, True, True, True,
 True, False, True, True, True, True, True, False, True,
 True, True, True, True, True, True, True, True])
Y_pred_04
array([ True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, False,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
3/22/23, 3:51 PM Copy of Heart health.ipynb - Colaboratory

 True, False, False, True, True, False, False, True, True,
 False, False, True, True, True, True, False, False, True,
 True, True, True, True, True, True, True, False, True,
 False, True, True, True, True, True, False, False, False,
 False, True, True, True, True, False, True, True, True,
```

```
 True, True, True, True, True, False, True, True, False,
 True, True, True, True, True, True, True, True, True,
 False, True, False, True, True, False, True, False, False,
 False, True, True, True, True, True, True, False, True,
 True, False, True, False, True, False, True, False, False,
 False, True, False, True, False, True, False, True, False,
 False, False, False, True, False, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 False, False, True, True, True, True, True, True, True,
 True, False, False, True, True, True, True, True, True,
 True, False, True, False, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, False, True, True,
 False, True, True, False, True, True, True, True, True,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, False,
 True, True, True, True, True, True, True, True, True,
 True, True, True, True, True, True, True, False, True,
 True, True, True, True, True, True, False, True, True,
 False, False, True, False, True, True, True, True, True,
 True, True, True, True, True, False, True, True, True,
 True, True, False, False, True, True, False, False, True,
 True, True, True, True, False, True, False, True, True,
 True, True, True, True, True, False, False, False, True,
 False, True, False, True, True, True, False, True, True,
 False, False, True, True, True, False, True, True, False,
 True, True, True, False, True, False, True, False, True,
 True, True, False, True, True, True, True, False, False,
 True, False, True, True, True, True, True, True, True,
 True, True, False, True, True, False, True, True, True,
 True, True, False, True, True, False, False, True, True,
 True, False, True, True, True, True, True, True, True,
 True, False, False, False, True, True, False, True, True,
 True, False, True, True, True, True, True, False, True,
 True, True, True, False, True, True, False, False])
Y_pred_06
array([ True, True, True, True, False, False, True, False, False,
 False, False, False, True, True, True, True, False, False,
 True, True, True, True, False, False, True, True, True,
 True, True, True, False, False, False, False, False, False,
 True, True, True, False, True, True, True, True, False,
 False, False, False, False, False, False, False, True, True,
 False, False, True, True, True, True, False, False, True,
 True, True, False, True, False, False, False, False, False,
 False, False, True, True, True, True, False, False, False,
 False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, False, True, False, False,
```

```
 True, False, False, False, False, False, True, False, False,
 False, True, False, True, False, False, False, False, False,
 False, False, True, False, False, False, False, False, False,
 False, False, False, False, False, False, False, False, False,
 False, False, False, False, False, True, False, True, False,
 False, False, False, True, False, True, True, True, True,
 True, False, False, True, False, False, False, False, False,
 False, False, False, False, True, True, True, False, True,
 False, False, False, False, False, False, False, False, False,
 False, False, True, False, False, False, False, True, False,
 False, True, False, False, False, False, False, False, True,
 False, False, False, False, False, False, False, False, False,
 False, True, False, False, False, False, False, False, False,
 False, False, False, True, False, False, True, False, False,
 False, False, True, False, False, False, True, False, False,
 True, True, False, False, False, False, False, False, True,
 False, False, False, True, True, False, True, False, False,
 False, False, True, True, False, False, False, False, False,
 False, False, False, False, False, False, True, False, False,
 False, False, True, False, True, False, True, False, False,
 False, False, False, False, True, True, False, False, False,
 False, False, False, False, False, True, False, True, True,
 True, True, True, False, False, False, False, False, True,
 False, True, False, False, False, False, False, False, False,
 False, False, False, True, True, False, False, True, False,
 False, True, False, False, False, False, False, False, False,
 True, False, False, False, False, True, False, False, False,
 True, False, True, False, False, False, True, False, True,
 True, False, False, False, False, False, False, True, False,
3/22/23, 3:51 PM Copy of Heart health.ipynb - Colaboratory
https://colab.research.google.com/drive/1GqMWDuh8h1R7OQtWqMI4r1WvQ_kR9EiK?usp=sharing#scrollTo=
    5/7
 False, False, False, False, False, False, False, False, False,
 True, False, False, False, False, True, True, False, False,
 False, False, False, False, True, False, False, True, False,
 False, False, False, True, False, False, True, False, False,
 True, False, False, False, False, False, False, False])
from sklearn.metrics import confusion_matrix
confusion_matrix(Y, Y_pred)
array([[122, 76],
 [ 66, 140]])
confusion_matrix(Y, Y_pred_03)
array([[ 22, 176],
 [ 10, 196]])
confusion_matrix(Y, Y_pred_04)
array([[ 64, 134],
 [ 31, 175]])
```

```
confusion_matrix(Y, Y_pred_06)
array([[167, 31],
 [127, 79]])
from sklearn.metrics import precision_score, recall_score
precision_score(Y,Y_pred)
0.6481481481481481
precision_score(Y,Y_pred_03)
0.5268817204301075
precision_score(Y,Y_pred_04)
0.5663430420711975
precision_score(Y,Y_pred_06)
0.7181818181818181
recall_score(Y,Y_pred)
0.6796116504854369
recall_score(Y,Y_pred_03)
0.9514563106796117
recall_score(Y,Y_pred_04)
0.8495145631067961
recall_score(Y,Y_pred_06)
0.38349514563106796
from sklearn.metrics import roc_auc_score
roc_auc_score(Y,Y_pred)
0.6478866333235265
3/22/23, 3:51 PM Copy of Heart health.ipynb - Colaboratory
https://colab.research.google.com/drive/1GqMWDuh8h1R7OQtWqMI4r1WvQ_kR9EiK?usp=sharing#scrollTo=
    6/7
roc_auc_score(Y,Y_pred_03)
0.5312837108953614
roc_auc_score(Y,Y_pred_04)
0.5863734431695596
roc_auc_score(Y,Y_pred_06)
0.6134647445327057
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test =train_test_split(X, Y, test_size=0.2,
    random_state=0)
print (X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
(323, 19) (81, 19) (323,) (81,)
clf_LR = LogisticRegression()
/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
    ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
Increase the number of iterations (max_iter) or scale the data as shown in:
 https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
 https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
 n_iter_i = _check_optimize_result(
    LogisticRegression
```

```
LogisticRegression()
clf_LR.fit(X_train, Y_train)
Y_test_pred = clf_LR.predict(X_test)
from sklearn.metrics import accuracy_score, confusion_matrix
confusion_matrix(Y_test,Y_test_pred)
array([[20, 16],
 [18, 27]])
accuracy_score(Y_test, Y_test_pred)
0.5802469135802469
import matplotlib.pyplot as plt
Y_pred_proba = clf_LR.predict_proba(X_test)[:,1]
fpr, tpr, _ = metrics.roc_curve(Y_test, Y_pred_proba)
#create ROC curve
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```