Analysis of heart health 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

The findings of the analysis suggest that the model is statistically significant at a 0.05 level, indicating that at least one of the predictor variables has a significant relationship with the outcome variable. However, the overall Pseudo R-squared value of 0.06508 indicates that the model only explains a small proportion of the variation in the "Delaydays" variable.

The analysis found that some of the predictor variables are significantly associated with "Delaydays", including cough, edema, and weight gain. However, many of the predictor variables do not have a significant relationship with "Delaydays".

3 Discussions

The Following discussions come up for the analysis :

Sample size: The dataset contains 404 observations, which may be considered small for a logistic regression model with 19 predictor variables. One discussion could be whether a larger sample size would lead to more accurate predictions and a better understanding of the relationships between predictor variables and the outcome variable.

Model fit: The model's Pseudo R-squared value of 0.06508 suggests that the model has low explanatory power. One discussion could be whether the model is appropriate for the data and whether other models, such as a random forest or a neural network, would yield better results.

Predictor variables: Several predictor variables, such as age, gender, ethnicity, and education, were found to be not significant in predicting "Delaydays." A discussion could be about whether these variables should be removed from the model to simplify it or whether other variables should be added to improve its accuracy.

Clinical significance: The logistic regression analysis identified several symptoms that are significantly associated with "Delaydays," including cough, edema, and weight gain. A discussion could be about the clinical significance of these findings and whether they have implications for medical practice or patient care.

Model assumptions: The logistic regression analysis assumes that the relationships between predictor variables and the outcome variable are linear, and that there is no multicollinearity or other violations of the assumptions. A discussion could be about whether these assumptions are valid for the data and whether alternative statistical methods, such as a generalized linear model or a non-parametric regression, would be more appropriate.

4 Appendix A: Method

Firstly, the necessary libraries such as numpy, pandas, seaborn, statsmodels, and scikit-learn are imported. Then the heart health data is loaded into a pandas DataFrame named df.

The data is preprocessed by removing any rows that contain null values using the .notnull() and .all() methods.

The X variable is created to hold all columns except the target variable 'Delaydays' and Y variable is created to hold the target variable 'Delaydays'.

A Logistic Regression model is created using scikit-learn's LogisticRegression() function and trained using the .fit() method on the training data X-train and Y-train.

The logistic regression model's coefficient and intercept values are calculated and displayed using the .coef and .intercept methods, respectively. Additionally, a logistic regression model is created using statsmodels' Logit() function, and the model summary is displayed.

The logistic regression model's predicted probabilities and predicted labels are calculated using the .predict-proba() and .predict() methods, respectively.

Different classification thresholds (0.3, 0.4, 0.6) are used to create new predicted labels and confusion matrices are generated to evaluate the model's performance at each threshold.

Precision and Recall scores are calculated using the precision-score() and recall-score() functions, respectively.

ROC AUC (Receiver Operating Characteristic Area Under the Curve) scores are calculated using the roc-auc-score() function for all predicted labels.

The data is then split into training and test sets using the train-test-split() function from scikit-learn, with 20 percentage of the data used for testing. The logistic regression model is trained on the training data and evaluated on the test data.

Confusion matrix and accuracy score are generated using the confusion-matrix() and accuracy-score() functions, respectively.

Finally, an ROC curve is plotted using the predict-proba() method to obtain predicted probabilities and the roc-curve() function to calculate the FPR (False Positive Rate) and TPR (True Positive Rate). The plot is generated using matplotlib's pyplot library.

5 Appendix B: Results

The dependent variable is "Delaydays" and the independent variables are the various demographic and medical variables included in the analysis.

The model as a whole has a pseudo R-squared value of 0.06508, indicating that it explains only a small proportion of the variance in the dependent variable.

The coefficients for each independent variable indicate the direction and magnitude of its effect on the probability of experiencing delayed days. A positive coefficient indicates

that an increase in the independent variable is associated with an increase in the probability of experiencing delayed days, while a negative coefficient indicates the opposite.

The p-values for each coefficient indicate the statistical significance of the corresponding independent variable. A p-value less than 0.05 is considered statistically significant, suggesting that the variable has a significant effect on the dependent variable.

In this analysis, only the variable "cough" has a statistically significant effect on the probability of experiencing delayed days ($p = 0.003$), while several other variables have p-values that are marginally significant (p ¡ 0.1).

It is important to note that logistic regression assumes that the relationship between the independent variables and the dependent variable is linear, which may not be the case in reality. Additionally, the model assumes that the independent variables are independent of each other, which may also not hold true in practice. Therefore, the results of this analysis should be interpreted with caution and validated with additional analyses.

6 Appendix C: Code

```
import numpy as np
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  ]
clf_lr = LogisticRegression()
clf_lr.fit(X,Y)
clf_lr.coef_
clf_lr.intercept_
X_cons = sn.add_constant(X)
logit = sn.Logit(Y,X_cons).fit()
logit.summary()
clf_lr.predict_proba(X)
Y_pred = clf_lr.predict(X)
Y_pred
Y_pred_03 = (clf_lr.predict_proba(X)[:,1] >= 0.3)
```

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: Summary of analysis

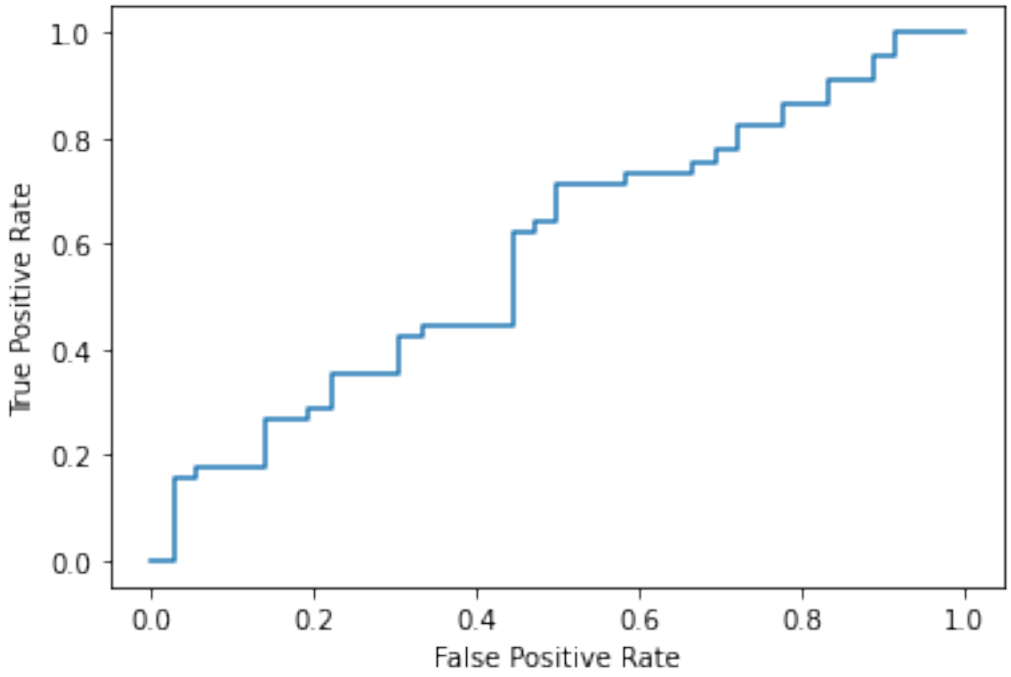

Figure 2: ROC curve for 0.5 probability

```
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
Y_pred_04
Y_pred_06
from sklearn.metrics import confusion_matrix
confusion_matrix(Y, Y_pred)
confusion_matrix(Y, Y_pred_03)
confusion_matrix(Y, Y_pred_04)
confusion_matrix(Y, Y_pred_06)
from sklearn.metrics import precision_score, recall_score
precision_score(Y,Y_pred)
precision_score(Y,Y_pred_03)
precision_score(Y,Y_pred_04)
precision_score(Y,Y_pred_06)
recall_score(Y,Y_pred)
recall_score(Y,Y_pred_03)
recall_score(Y,Y_pred_04)
recall_score(Y,Y_pred_06)
from sklearn.metrics import roc_auc_score
roc_auc_score(Y,Y_pred)
roc_auc_score(Y,Y_pred_03)
roc_auc_score(Y,Y_pred_04)
roc_auc_score(Y,Y_pred_06)
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)
clf_LR = LogisticRegression()
n_iter_i = _check_optimize_result(
LogisticRegression
13
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)
accuracy_score(Y_test, Y_test_pred)
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()
```