INFO370 Problem Set: Predictive modeling, validation

May 20, 2023

Introduction

This PS has the following goals:

- use *sklearn* library for predictive modeling
- learn to predict the outcomes and probabilities
- understand confusion matrix and related concepts
- use different methods for predictive modeling
- do 3-fold testing-validation-training workflow

In this question, we will construct a logistic regression model to predict the probability of a person having a heart attack. The dataset *heart.csv* comes from Kaggle www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset, which contains health information of each person (predictors) and whether or not the person had a heart attack before (outcome variable). You can download the data *heart.csv* from Canvas. The variables are (as described on the webpage):

age age of the patient

sex sex of the patient (1 = male; 0 = female)

cp chest Pain type chest pain type (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic

trtbps resting blood pressure (in mm Hg)

chol cholestoral in mg/dl fetched via BMI sensor

fbs (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

restecg resting electrocardiographic results. 0: normal; 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV); 2: showing probable or definite left ventricular hypertrophy by Estes' criteria.

thalachh maximum heart rate achieved

exng exercise induced angina (1 = yes; 0 = no)

caa number of major vessels (0-3)

output 0: did not have a heart attack, 1: had a heart attack

slp

oldpeak

thall

Note that the last three variables are not documented. Neither do we know how the data was collected.

We use the dataset to predict heart attack *output*.

The problem set is constructed in a way that it is much easier to use *sklearn* package instead of *statsmodels* below. See Python Notes 11.2.2 "Scikit-learn and LogisticRegression" for how to do logistic regression with sklearn.

1 Prepare data (19pt)

First, let's do some descriptive statistics.

1. (1pt) Load data. The data should contain 303 rows, and 11 columns.
2. (3pt) Do some basic checks. Do we have any missing values? What are the data types? What are ranges of numeric variables, and possible values of categorical variables? What is the percentage of heart attack among these patients?

   Compare the values with the documentation and comment what do you see.
3. (2pt) You probably noticed that all the above variables are coded as numbers. However, not all of these are in fact of numeric (interval, ratio) measure type. Which variables above are inherently non numeric (nominal or ordinal)?

   Hint: Lecture Notes *1.1.1 Measures: Possible Mathematical Operations* explains the measure types.
4. (4pt) We are going to use sklearn library. Construct the outcome vector $\boldsymbol{y}$ and the design matrix X. It should include all explanatory variable (but not *output*!). The variables that are inherently categorical should be converted to categorical. How many columns do you get?

   Hint: https://faculty.washington.edu/otoomet/machinelearning-py/ discusses how to convert categoricals to dummies. I end up with 16 columns.
5. (3pt) Split data (both $\boldsymbol{y}$ and X) into work and testing chunks (80/20). Do not look at the testing chunk.

   Your work data should be 242 and test data 61 rows or so.
6. (5pt) Save your test data (both $\boldsymbol{y}$ and X to a file to be used later. Delete all datasets that contain the testing data from memory. This includes the original data frame, X, and $\boldsymbol{y}$, and the test data itself.

   (Check out the command `del`).
7. (2pt) Split your work data into training and validation chunks (80/20). Below, you will only need these two chunks, until the very last question.

   Hint: you should have 193 and 49 rows.

Further you use *only* work data, until the final test in Question 5.

---

2 Logistic regression (16pt)

Here your task is to work with a logistic regression model.

1. (4pt) What do you think, which measure–precision, recall, or F-score–will be most relevant in order to make this model more applicable in medicine? Explain!

   Below, you will compare models based on both accuracy and the measure you suggest here.

2. (4pt) First, imagine we create a very simple model ("naive model") that predicts everyone the same result (attack or no attack), whichever category is more common in data (the majority category).

   How would the confusion matrix of this model look like? What are the corresponding accuracy, precision and recall? Show this confusion matrix and explain!

   Note: you should not fit any model here, you are able to compute these all these values manually with just a calculator!

3. (2pt) Construct a logistic regression model in sklearn and fit it with your training data

4. (2pt) Use your validation data to predict the outcome—that is, whether someone has heart attack or not. Print out the first 10 labels. Which values denote attack, which ones non-attack?

5. (2pt) Display the confusion matrix.

6. (2pt) Compute and display accuracy and the measure you suggested in 1. How do these results compare to the naive model?

---

3 Other ML methods (30pt)

Now it is time to try other machine learning methods. Which one gives you the best predictions? You can use the same design matrix and the outcome vector as what you used for logistic regression above. But remember: use only training data for training!

3.1 Nearest Neighbors (20pt)

Here you will use nearest neighbors to model heart attack, and try to find the best number of neighbors k.

1. (8pt) Loop over different number of neighbors k (from a single neighbor to all data points). For each k value:

   - train the model on training data and
   - compute the performance (accuracy and your suggested measure) on training data
   - compute the performance (accuracy and your suggested measure) on validation data
   - store the performance value(s)

2. (4pt) Make a plot where you show both the training and validation performance as a function of k.

3. (4pt) Find the k value that gives you the best performance (both accuracy and your suggested measure). How good measures do you get? How do they compare with the naive model? How do they compare with logistic regression?

4. (4pt) Which k values (if any) are overfitting, which ones underfitting? Why do the model performance measure flatten out when k is very large?

3.2 Decision trees (10pt)

Now let's repeat the above with decision trees, and find which max depth value gives the best results.

1. (2pt) Loop over different values of max depth, from 1 to 10. For each max depth value:
   - train the model on training data and ...
   - ... compute the performance (accuracy and your suggested measure) on training data
   - compute the performance (accuracy and your suggested measure) on validation data
   - store the performance value(s) in a list or arrya
2. (2pt) Make a plot where you show both the training and validation performance as a function of k.
3. (2pt) Find the k value that gives you the best performance (both accuracy and your suggested measure). How good measures do you get? How do they compare with the naive model? How do they compare with logistic regression?
4. (2pt) Which k values (if any) are overfitting, which ones underfitting?
5. (2pt) Out of the models you tried–logistic, k-NN, and trees–which is the best model in terms of validation accuracy? How much was the accuracy over what the naive model gave you?

   In the next two questions you are using just that model.

---

4 How confident are we in the results? (15pt)

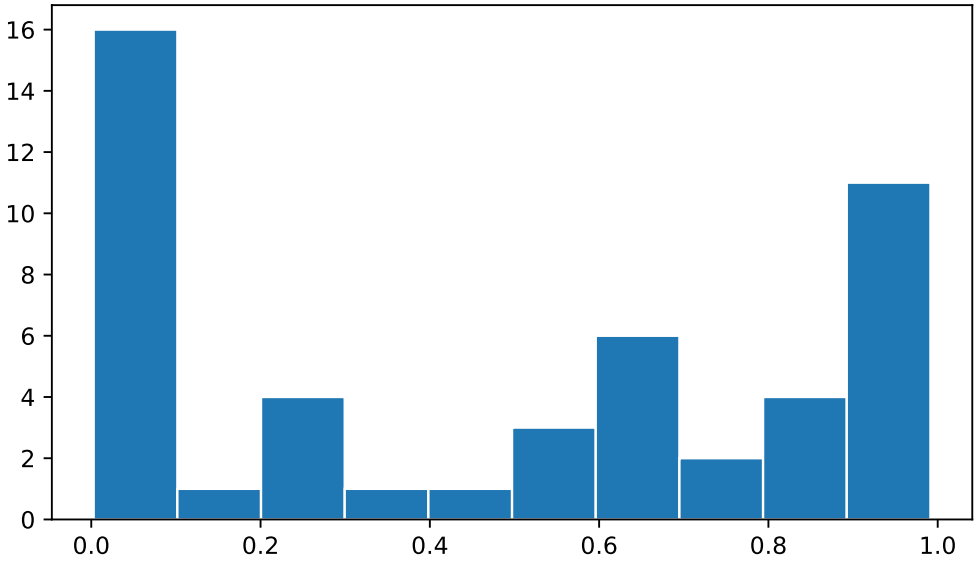
The (almost) last task is to evaluate the confidence in the results. You will predict probability of hearth attack for everyone in the validation data and analyze the results–if the probabilities tend to be close to 0 or 1, then we are rather certain in the predictions. If they pile up around 0.5, then the predictions are rather unclear. Let's focus on accuracy only here.

1. (5pt) Predict the *probability* of having a heart attack $\Pr(output = 1|\boldsymbol{x})$ for everyone in data. Print out the first 10 probabilities.

   Note: print *only* probability for heart attack, not probability of non–heart attack!

   Hint: Python Notes 12.2.4 discusses predicting logistic regression results with *sklearn*.
2. (5pt) make a histogram of your predictions. What do you see–are the predicted probabilities more extreme (either 0 or 1) or in the middle (around 0.5)?

   Hint: this is my histogram looks like:



3. (5pt) Why does predicted probability around 0.5 indicate that the results are uncertain?

---

5 Final model goodness (10pt)

This is your final task. Do not start it before you have done all the tasks above–this is because after you have looked at your test data, you are not allowed to work on the models any more.

1. Take your best model, the one you used in Question 4. But now fit it with *complete work data*, not just with your training data.
2. Load your test data, the one you saved earlier. Use the test data to compute test accuracy. This is your final model performance measure. Show it!

Now where you have used your test data you cannot go back and change anything. Congrats, you are done!

Finally...

... how much time did you spend on this PS?