INFO370 Problem Set: Logistic Regression, Prediction

Due: 03/06/2022

Instructions

This PS has the following goals:

- learn to use and interpret logistic regression results
- learn to handle categorical variables
- learn to predict the outcomes, and compare with the actual data.

1 Heart Attack Prediction (60pt)

In this question, we will construct a simple logistic regression model to predict the probability of a person having a heart attack. The dataset comes from Kaggle www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-d 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 variable descriptions are:

- age: age of the patient
- sex : sex of the patient (1 = male; 0 = female)
- cp : chest Pain type chest pain type
  - Value 1: typical angina
  - Value 2: atypical angina
  - Value 3: non-anginal pain
  - Value 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
  - Value 0: normal
  - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV)
  - Value 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)
- (**outcome variable**) output : 0= did not have a heart attack 1= had a heart attack

Answer the following questions:

1. (2pt) Load the data and drop the variable **slp, oldpeak** and **thall**, since they do not have descriptions available. Do some basic sanity check (descriptive statistics; missing values; data types...). The data should contain 303 rows, and 11 columns.

2. (22pt) Construct a simple logistic regression with **statsmodel.formula.api** package. The outcome variable is **output**. The rest 10 variables (**age, sex, cp, trtbps, chol, fbs, restecg, thalachh, exng, caa**) in the dataset are predictors. Print out the marginal effect summary table and answer the following questions:

   - (6pt) Interpret coefficient for sex. Is it statistically significant at significance level of 0.05?
   - (6pt) Interpret coefficient for cp. Is it statistically significant at significance level of 0.05?
   - (6pt) Interpret the coefficient for age. Is it statistically significant at significance level of 0.05?
   - (4pt) What are the variables that are associated with lower chance of having a heart attack, and also are statistically significant? Do they intuitively make sense? Why?

3. (6pt) Now let's construct the same model with sklearn package using the following code:

   ```
   LogisticRegression(penalty='none', solver='newton-cg').fit(X,y)
   ```

   Remember that sklearn package requires the predictor values (X) to be separated with the outcome variable (y). The predictor values also need to be in matrix format. Answer the following question:

   - (6pt) Print out the coefficient and intercept. Did you get the same intercept and coefficients as what you got from statsmodel.formula.api package? (you are supposed to!!)
     After using both packages, you should have a clearer sense that smf package is good for inferences becuase it provides information about pvalues, tvalues, CI etc., which can be used to determine statistical significance. Sklean package does not provide such information. However, the advantage of sklearn is that it is easier to construct models and make prediction with sklearn, especially when you are comparing multiple models.

4. (18pt) With the sklearn model, let's do some predictions on the training set (the same X and y we used to train the model from Q3):

   - (6pt) The probability of having a heart attack: $P(\text{output}=1|\text{X})$. Print out the first 10 probabilities.
   - (6pt) The outcome labels — that is, we directly predict whether or not the person will have a heart attack, instead of predicting the probability. Print out the first 10 labels.
   - (6pt) Show the steps of how to convert from probabilities to the labels using threshold 0.5.

5. (6pt) Calculate the accuracy of the predicted output labels, compared with the true output labels. You can calculate it using your own code or using sklearn.metrics package. How to interpret the accuracy? Do you think the accuracy is high enough, such that you are comfortable deploying this model in real world to predict heart attack? (**Hint:** you should get about 80% accuracy.)

6. (6pt) Create a confusion matrix on the training data. Calculate accuracy, precision, and recall based on the confusion matrix.

2 Predict AirBnB Price (40pt)

Your second task is to predict Beijing AirBnB listing price (variable price). You use the same dataset as in PS06, and the same model (model in question 2.7).

1. (5pt) Replicate the model from PS06 question 2.7. Copy paste of your old code is OK.
2. (10pt) Now use the model above to predict (log) price for each listing in your data.
3. (10pt) Compute root-mean-squared-error (RMSE) of this prediction. RMSE is explained in lecture notes, 4.1.5 "Model evaluation: MSE, RMSE, R2".
4. (10pt) Now use your model to predict the price for a 2-bedroom apartment that accommodates (i.e. a full 2BR apartment).
5. (5pt) Compute the average log price for all listings in this group (2BR apartment that accommodates 4). Compare the result with your prediction. How close did you get?