INFO370 Problem Set: Prediction, confusion matrix

February 26, 2023

Introduction

This PS has the following goals:

- learn to predict the outcomes, and compare those with the actual data.
- Compute and understand RMSE
- understand confusion matrix and related concepts
- learn to use *sklearn* library
- use different methods for predictive modeling

1 Predict AirBnB Price (40pt)

Your first task is to predict Beijing AirBnB listing price (variable *price*). You use the same dataset as in PS05, 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. You also need to convert price to numeric and whatever cleaning you did in PS5.

   If you did it correctly, then you should be predicting log price using BR, room type, accomodates, and bathrooms.

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, R^2".

4. (10pt) Now use your model to predict the price for a 2-bedroom apartment that accommodates 4 (i.e. a full 2BR apartment).

   If you need more information, then you can decide yourself what to do–e.g. you can pick a reasonable value, or maybe you want to pick the sample average value.

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?

2 Heart attack: predictive modeling (50pt)

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*.

2.1 Load and check (8pt)

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. (4pt) 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: 1.1.1 Measures: Possible Mathematical Operations explains the measure types.

2.2 Logistic regression (42pt)

Now let's do a logistic regression model using sklearn package. Remember that sklearn package requires the matrix of predictor values (X) to be separated from the vector of outcome variable ($\boldsymbol{y}$). The predictor values must be a matrix, not a vector. See Python Notes 11.2.2 for how to use *sklearn* for logistic regression.

1. (4pt) Construct the design matrix X. It should include all explanatory variable (not *output*!).
2. (4pt) Convert those variables to categorical that should be treated as categorical. How many columns do you get?

   Hint: I have 16 columns.
3. (4pt) Construct a logistic regression model in sklearn and fit it with your X and the outcome variable *output*.

Now it is time to use sklearn for predictive modeling.

4. (4pt) Predict the *probability* of having a heart attack $P(output = 1|\boldsymbol{x})$ 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*.
5. (4pt) Predict the *label* (*outcome*)—that is, whether someone has heart attack or not, instead of predicting the probability. Print out the first 10 labels.
6. (4pt) You can predict labels in two ways: you can use `.predict` method, or alternatively, you can compute probabilities and find where they are larger than 0.5.

   Show that both methods give you the same results.

   Hint: you can use your probability/label predictions from above.
7. (4pt) Display the confusion matrix.
8. (4pt) Compute and display precision and recall.
9. (5pt) What do you think, which measure–accuracy, precision, recall, or F-score should we try to improve in order to make this model more applicable in medicine? Explain!
10. (5pt) Now imagine we create another 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 Other ML methods (10pt)

Finally, it is time to try other machine learning methods. Which one gives you the best predictions? You can use the same design matrix X and the outcome vector $\boldsymbol{y}$ as above.

1. (4×2 pt) Now use the following models:

   - 1-NN
   - 5-NN
   - 15-NN
   - Decision trees

   Each time display the confusion matrix, and compute the measure (accuracy, precision, recall, or f-score) you think is the best for this task (as you answered in 2.9).

2. (2pt) If you did it correctly, you noticed that 1-NN gave you the best result. Why is it so? Do you think this indicates that 1-NN is the best approach here?

Finally...

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