20210927_HW2_prompts

Will Wang

9/27/2021

BIOSTAT 707 Homework 2

In this homework, the objectives are to

1. Use R to examine and preprocess a dataset
2. Implement Unsupervised learning in a real-world problem, including: Principal Component Analysis (PCA), Hierarchical Clustering, and K-means Clustering in R
3. Visualize and understand PCA, Hierarchical Clustering Dendrograms, and K-means Clustering in R
4. Implement a k-Nearest Neighbors (kNN) Classifier on a real world dataset
5. Implement cross validation with kNN Classifier

Assignments will only be accepted in electronic format knitted HTML files from RMD. **5 points will be deducted for every assignment submission is not knitted PDF file.** I recommend you knit the file to HTML and then open the HTML in browser, which will allow you to save the HTML as PDF file. Your code should be adequately commented to clearly explain the steps you used to produce the analyses and your codes and texts must wrap appropriately so that the graders can see all of your work. Homework submissions should be uploaded to Sakai with the naming convention date_lastname_firstname_HW[X].Rmd. For example, my first homework assignment would be named 20210831_Dunn_Jessilyn_HW1.Rmd. **It is important to note that 5 points will be deducted for every assignment that is named improperly.** Please add your answer to each question directly after the question prompt in the homework .Rmd file template provided below.

```
library(tidyverse)
library(ggplot2)
library(lubridate)
library(patchwork)
library(gridExtra)
library(psych)
library(corrplot)
library(ggfortify)
library(factoextra)
library(class) #knn
library(gmodels) # CrossTable()
library(caret) # creatFolds()
library(caTools) #sample.split()
library(ROCR) # prediction(), performance()
set.seed(2021)
```

Dataset: Hepatitis C Virus Diagnosis (HCV dataset)

https://archive.ics.uci.edu/ml/datasets/HCV+data (https://archive.ics.uci.edu/ml/datasets/HCV+data)

Please refer to this website for additional information about the dataset we are using in this HW.

The target attribute for classification is Category (blood donors vs. Hepatitis C).

Attribute Information:

All attributes except Category and Sex are numerical. The laboratory data are the attributed numbered 5-14 below. 1) X (Patient ID/No.) 2) Category (diagnosis) (values: '0=Blood Donor', '0s=suspect Blood Donor', '1=Hepatitis', '2=Fibrosis', '3=Cirrhosis') 3) Age (in years) 4) Sex (f,m) 5) ALB 6) ALP 7) ALT 8) AST 9) BIL 10) CHE 11) CHOL 12) CREA 13) GGT 14) PROT

Data Preparation

1. Download the HCV data titled "hcv.csv" from Sakai and import it into R. Look at the first 5 lines of the data to learn about the dataset.

2. Answer the following questions by using the summary function or other methods of your choice:

a. How many observations are there?
b. How many independent variables are there?
c. Which columns have missing values and how many values were missing in each?
d. How many observations are there with positive diagnosis and how many are there with negative diagnosis?

For this question, please type your answers clearly outside of R chunks, and do not just show with running your codes.

3. Perform the following tasks to prepare this dataset for analysis:

- Drop observations with NAs
- Convert *Sex* variable from character string into interger form of 1's and 0's; we will arbitrarily set "m" to 1 and "f" to 0.
- Convert *Category* variable from characters into Boolean form of TRUE and FALSE, where 0's refer to negative diagnosis ("0=Blood Donor" and "0s=suspect Blood Donor") and 1's refer to positive diagnosis ("1=Hepatitis", "2=Fibrosis", and "3=Cirrhosis")

4. Use the function table() to determine how many positive observations and negative observations are in this dataset? Since we observe imbalance in the class labels, we will perform downsampling by propensity matching according to *Age* and *Sex*. A potential set of steps to perform this downsampling is as follows:

- Separate the dataframe into two dataframes, one with all positive diagnosis and the other with negative diagnosis.
- Count instances of different combinations of sex and age in the positive diagnosis dataframe.
- For each combination and corresponding count *n*, randomly select *n* instances from the negative diagnosis for that combination of sex and age.
- Combine the positive observations and the negative ones you selected.
- Please set.seed(2021) so that your results are reproducible. Eventually, you should have a dataset with the total number of rows that is twice the number of positive observations, with an almost 50:50 split between the positive and negative observations. It's possible to encounter situations where for example, you have 3 positive observations who are female and 50 years old, but you only have 2 negative observations who are female and 50 years old. In that case, you don't need to match the number of positive and negative observations in that combination of demographics.

5. After imputation, use “ggplot” and “facet_wrap” to plot a grid of histograms with 3 plots per row to explore the data shape and distribution of all the independent variables in this dataset. When you plot, remember to select a reasonable number of bins and add legends and labels when appropriate. Adjust the size of the plot display so that you can see all the facets clearly when you knit. Adjust the figure to the appropriate sizes.

6. The pre-processed dataset needs to be scaled before performing PCA. Please give a brief explanation on why that is the case? Use scale() to standardize the independent variables in this dataset except for sex. Structure a new dataframe that has all the standardized independent variables as well as the binary label column. Hint: you can use as_tibble() function to nicely format the standardized columns into a dataframe.

Principal Component Analysis (PCA)

7. Calculate principal components using function princomp() and print the summary of the results.

8. Plot a scree plot using the screeplot() function.

9. Plot the following two plots and use patchwork/gridExtra to position the two plots side by side:

a. proportion of variance explained over the number of principal components
b. cumulative proportion of variance explained plot over the number of principal components; draw horizontal lines at 88% of variance and 95% variance. Note: please remember to clearly label your plots with titles, axis labels and legends when appropriate. You do not need to put down legends if they are not necessary.

10. What proportions of variance are captured from the first, second and third principal components? How many principal components do you need to describe at least 88% and 95 % of the variance respective?

11. Which are the top 3 variables that contribute the most to the variance captured from PC1? (hint: look at the loadings information)

12. Plot a biplot of the PCA analysis using the biplot() function.

13. Since the biplot is difficult to discern, we usually use the autoplot() function in package “ggfortify” to display a clearer biplot overlaid with scatter plot for the first 2 principal components. Remember to add appropriate titles, labels and coloring to display the plot clearly.

Hierarchical Clustering

14. Calculate a dissimilarity matrix using Euclidean distance and compute hierarchical clustering using the complete linkage method and plot the dendrogram. Use the rect.hclust() function to display dividing the dendrogram into 4 branches.

15. Divide the dendrogram into 4 clusters using cutree() function. Then use table() function and the diagnosis label information to compare the composition (positive vs. negative diagnosis/outcome) of each of the 4 clusters. How would you label each of these four clusters (e.g. cluster 1 is TRUE or FALSE, cluster 2 is …, etc.)?

16. What would be the classification accuracy if you follow the assignment of clusters decided in question 16?

17. Perform the same procedure as laid out in question 15 and 16, but for 10 clusters in this question. Calculate the classification accuracy for 10 clusters.

18. Now try 4 clusters with Ward's linkage method and plot the dendrogram. Then use table() function to view the clustering result. How would you label each of these 4 clusters? What is the classification accuracy in this case?

19. Now try 10 clusters with Ward's linkage method and plot the dendrogram. Then use table() function to view the clustering result. How would you label each of these 10 clusters? What is the classification accuracy in this case?

20. What observations can you make about from the previous 4 attempts at hierarchical clustering? I.e., does the clustering result change using different number of clusters? Does higher number of clusters lead to clustering result that is closer to the actual outcomes? Answer these questions and other relevant observations based on the plots and the classification accuracies you have calculated.

K-Means Clustering

21. Compute k-means clustering on this dataset using the kmeans() function for two clusters. Then use the table() function and the diagnosis label information to compare the composition (TRUE vs. FALSE in the outcome) of each of the 2 clusters (hint: the cluster information from k-means is stored in the $cluster attribute in the k-means result.) What's the clustering classification accuracy?

22. Visualize the clusters using the fiz_cluster() function from factoextra package. You should have the x-axis as PC1 and y-axis as PC, and produce come form of cluster boundaries around the two clusters.

23. While we can adjust the number of clusters in hierarchical clustering to achieve better clustering, we do not do so for k-means. Why do you think that is the case? In this dataset, which method seem to be more appropriate? Give your reasoning briefly, taking into consideration both the accuracies and the different procedures of analysis.

---

kNN

24. We will randomly sample from our dataset to split it into 80:20 (training:test) datasets using convenient tools from caTools package. After you generate train_df and test_df, separate train_df into X_train, and y_train where X_train contains only independent variables and y_train contains on the "diagnosed" label. Perform the same separation for test_df.

- Note: SplitRatio is set to 0.8 to make training set comprised of 80% of the original data
- Note: Set seed to 2021

25. Generate a KNN model using the knn() function. Usek = sqrt(# observations in the training set).

- Learn the syntax of knn() from https://www.rdocumentation.org/packages/class/versions/7.3- (https://www.rdocumentation.org/packages/class/versions/7.3-) 15/topics/knn.
- Note: The dependent variable that you aim to predict should not be included in the training and test data frames that you pass along to knn().
- Note: The labels for the training dataset should be passed separately. Recall that we designed two new vectors for this purpose: y_train and y_test
- It should be clear to you that the output of knn() is a list of the predicted values for the test set you passed.

26. Produce a confusion matrix demonstrates the number of true and false positives, and true and false negatives, that our model predicts. The confusion matrix displays this in a table so that we can clearly see

where the model performs well and where it fails to make the correct prediction. Create a confusion matrix using the CrossTable() function from the package gmodels.

- Set prop.chisq = FALSE so that chi-squared contribution from each cell is ignored. Only the minimum amount of information is needed to answer this question.
- Learn the syntax of the CrossTable() function from https://www.rdocumentation.org/packages/gmodels/versions/2.18.1/topics/CrossTable (https://www.rdocumentation.org/packages/gmodels/versions/2.18.1/topics/CrossTable)

27. How many false positives are there (hint: here, a positive call means that the image contains signs of Diabetic Retinopathy)? Using the definitions that we learned in class, calculate and print accuracy, sensitivity, error rate, and precision. You may choose either to use the information from the printed confusion matrix or to calculate using the equations from lecture slides. However, make sure you show your code, print out the results of the commands, and annotate your code clearly for full credit.

K-fold cross validation with kNN, where K and k have different meanings

28. In the previous kNN model, we divided up our data into 80% training and 20% test. We built the model on the 80% of the training data and tested how well the kNN model performed on the 20% of the test data. Another way that we can evaluate our model, besides holding out 20% of the data for testing, is to use a cross-validation method. Recall from class the K-fold cross validation strategy. The createFolds() function samples observations from the dataset randomly, and the code for which is provided. You are free to change the names in this code to the objects in your own codes. The set.seed function ensures that randomly generated numbers are the same each time so that your answers are consistent. This allows for greater reproducibility, avoiding unnecessary randomness. set.seed(2021) is included when loading the packages. The number 2021 is selected arbitrarily.

```
set.seed(2021)
idx <- createFolds(df_scale$Outcome, k = 5)
sapply(idx, length)
```

Now, we are going to build a knn model again with the more robust cross-validation method to evaluate its output. Train five kNN models using k = 11 neighbors for each of the K= 5 CV folds and compute the error rates of each kNN model on the held-out test data for each fold. The error rate is the number of observations that are classified incorrectly divided by the total number of predictions made. Print the average of the 5 error rates.

29. Plot error rates vs. k values (from knn) for all odd numbers between 5 and 13 (i.e. 5,7,9,11,…,21) using the methods we used in question 10. Which k value gives the minimum average error rate when you perform 5-fold cross validation? Explain what may have caused your initial kNN models from Question 10 to have high error rates and how k-Fold Cross Validation has improved the accuracy rate of your kNN models.