ECMA 31330 - Problem Set 5

Bruno Aravena M.

Due date: March 6th 3.30pm

Instructions: For every assignment, you will be asked to submit a document where you write up your answers and insert the figures/ tables requested for the assignment.You will also be required to submit the code where you code your simulations and produce your answers, figures, tables, etc. You can write your code in any programming language you want, but I will be submitting solutions and presenting examples in TA sessions in R.

This means you will submit a .pdf where you answer the questions and include your final figures, and also a script of code (e.g. .R) where I can see your work. Even if you work with a document that integrates code and answers (e.g. R Markdown or Jupyter Notebook I would insist in you also sending a .pdf version of it.

1 K-Means

In this problem set you will use different methods to predict a binary variable. We will be working with the Wage dataset from the textbook. Once again, In R these can be accessed through the ISLR2 package. Details and explanation for each dataset can be found clicking here. If you are not using R, the .csv files can be found here. The goal will be to predict the variable `jobclass`.

1. The variable `jobclass` takes 2 values ("Information" and "Industrial"): create an indicator variable that takes the value of 1 when the job is of class "Information".

2. Take a random sample of 20% that indicates whether an observation will be on training and test. We will be using the same split across methods: Each method will be trained on the training dataset and we will compare its performance in the test data set.

3. K-means relies on numeric data. We will use `age`, `logwage` and `education_years`. Note that `education_years` does not exist in our dataset: Propose and implement a mapping between the categorical variable `education` and `education_years`.

4. Create a version of your data that contains your outcome variable and a standardized version of `age`, `logwage` and `education_years`. **Important: your standardization must include both training and test observations.**

5. Run k-means with k = 8 on your training dataset.

6. You now have 8 groups or clusters. Compute the fraction of observations for which your outcome variable is equal to 1 on each group. If this fraction is above 0.5 (i.e. most of the observations in that group have the job class "Information") we will "predict" that every observation in that group has a job of the class "Information".

7. Would your assignment of clusters into binary predictions change if you instead used the cluster coordinate in the outcome variable to determine the prediction associated to each cluster? Why ? Why not ?

8. Using this prediction compute a training set (i.e. within-sample) classification accuracy (number of observations correctly classified / total of observations).

9. We now want to predict a job class for each observation in the test set. We have a model with 8 clusters and a prediction for each. However, since the test observations were not included in the k-means, we a priori do not know to which cluster they belong to. Considering only the numeric variables(`age`, `logwage` and `education_years`), compute the sum of squared distances and match each test observation to the centroid that minimizes this sum. *(Note: this is equivalent to running k-means in the test set, when already fixing the centroids.)*

10. Compute a test set classification accuracy (number of observations correctly classified / total of observations).

2 Neural Networks

1. Reuse the standarized version of your data from Question 1.4 and Train a Neural Network to predict the indicator variable for job class with the following architecture:
   Input layer: 3 features (`age`, `logwage` and `education_years`)
   First hidden layer: 16 neurons with ReLU activation.
   Second hidden layer: 16 neurons with ReLU activation.
   Output layer: 1 neuron (for regression).
   Hint: You might want to check the code posted for NN with binary classification

2. Report the classification accuracy for the training dataset

3. Report the classification accuracy for the test dataset.

3 Regression Trees

1. Run a regression tree that predicts the outcome variable based on `age`, `logwage` and `education_years` with 8 leaves or final nodes. Set control parameters to achieve this (e.g. max depth = 3).
2. Plot this tree.
3. Report the classification accuracy for the training dataset
4. Report the classification accuracy for the test dataset.
5. Run a regression tree that predicts the outcome variable based on all the data but that is still constrained to a maximum of 8 leaves or final nodes. Set control parameters to achieve this (e.g. max depth = 3).
6. Plot this new tree.
7. Report the classification accuracy for the training dataset
8. Report the classification accuracy for the test dataset.

4 Compare all methods

1. Create a table where you compare the accuracy in the training and test datasets across the 4 methods employed to classify (k-means, NN, tree, tree with more data).
2. Comment on your results. Did we expect some approaches in theory to perform better than others for this task? Did they perform as expected?
3. (*Optional / Extra Credit*): The problem set was mostly oriented at giving you exposure to these methods. However so much more can be done. Implement 1 or 2 modifications or new approaches that might outperform these benchmark examples. Meaningful attempts and/or meaningful improvements will be considered for extra credit.