Machine Learning Assignment 3

Hands on with Linear and Logistic Regression and Decision Trees

(30 points)

Problem1—Predicting Income using Logistic Regression and Decision Trees
For this problem, you will be using the Adult dataset from here (https://archive.ics.uci.edu/ml/datasets/adult) and the goal is to use a logistic regression and a decision tree model to predict the binary variable income (>50K or <=50K) based on the other attributes in the dataset. Read the attribute information from here then click on the datafolder and download "adult.data".

1. (0.5 pt) Load adult.data into a dataframe in R. Note that adult.data does not have column names in the first line so you need to set header=FALSE when you read the data then manually set the column names. Inspect the dataset using "str" and "summary" functions. What is the type of each variable in the dataset numeric/continuous or categorical/discrete? For each categorical variable explain whether it is nominal or ordinal.

2. (0.5 pt) There are some missing values in this dataset represented as " ?" (Note: there is a space before ?) . Make sure that all " ?" are converted to NAs. You can do so by setting "na.strings" parameters in "read.csv" to " ?".

3. (1pt) Set the random seed, and split the data to train/test. Use 80% of samples for training and the remaining 20% for testing. You can use "sample" (similar to what we did in slide 37 of week 6 lecture but you need to adjust1000 and 900 to the number of observations in your dataset and the size of the sample) or alternatively, you can use "createDataPartition" method from caret package.

4. (3pt) **Read the section on "Handling Missing Data" in chapter 13 of the textbook Machine Learning with R.** Find which columns/variables in the train and test set have missing values. Then decide about how you want to impute the missing values in these columns. Explain why you chose this imputation approach.

5. (3pt) The variable native-country is sparse, meaning it has too many levels, where some levels occur infrequently. Most machine learning algorithms do not work well with spares data. One-hot-encoding or dummy coding of these variables will increase feature dimensions significantly and typically some preprocessing is required to reduce the number of levels. One approach is to group together the levels which occur infrequently. For instance, one could combine together countries with less than 0.1% occurrence in the data to an "other" category. Another possibility is to use domain knowledge; for instance, combine countries based on their geographic location ("Middle East", "East-Europe", "West-Europe", etc. In a subsequent assignment we will use "feature hashing" which is yet another way to deal with sparse data but for now, please **read the section on Making use of sparse data (remapping sparse categorical data) in chapter 13 of the textbook Machine learning with R.** Then combine some of the infrequent levels of the native-country. You can decide whether you want to combine the levels based on frequency or domain knowledge. Either one is fine for this assignment but preference will be with a choice that would increase the cross validation performance of the ML models you will train subsequently.

6. (3pt) Use appropriate plots and statistic tests to find which variables in the dataset are associated with "income". Remove the variable(s) that are not associated with income.

7. (2pt) Train a logistic regression model on the train data (preprocessed and transformed using above steps) using the glm package and use it to predict "income" for the test data. Note: As explained in the lectures, "predict" method will return predicted probabilities. To convert them to labels, you need to use some threshold (typically set as 50%) and if the predicted probability is greater than 50% you predict income>50K; otherwise predict income<=50K (please review the example in lecture 7.2).

8. (3 pt)Get the cross table between the predicted labels and true labels in the test data and compute the

total error as well as the **precision** and **recall** for both income<=50K and income>50K classes.

9. (3pt) The target variable "income" is imbalanced; the number of adults who make <=50 is three times more than the number of adults who make >50K. Most classification models trained on imbalanced data are biased towards predicting the majority class (income<=50K in this case) and yield a higher classification error on the minority class (income >50K).

   One way to deal with class imbalance problem is to *down-sample* the majority class; meaning randomly sample the observations in the majority class to make it the same size as the minority class.

   The downside of this approach is that for smaller datasets, removing data will result in significant loss of information and lower performance. In Module 12, we will learn about other techniques to deal with data imbalance without removing information, but for this assignment, we use down-sampling in an attempt to address data imbalance.

   Note: Down-sampling should only be done on the training data and the test data should have the original imbalance distribution. You can downsample as follows:

   - Divide your training data into two sets, adults who make <=50K and the ones who make >50K.
   - Suppose that the >50K set has *m* elements. Take a sample of size *m* from the <=50K set.
     - You can use "sample" from the base package to sample the rows or alternatively, you can use the method "sample_n" from dplyr package to directly sample the dataframe
   - Combine the above sample with the >50K set. You can use "rbind" function to combine the rows in two or more dataframes. This will give you a balanced training data with the same observations in >50K and <=50K classes.
   - Re-train the logistic regression model on the balanced training data and evaluate it on the test data. Compare the total error, precision, and recall for the <=50K class and >=50K classes with the previous model. Which model does better at predicting each class?

10. (3pt) Repeat steps 7-9 above but this time, use a C5.0 decision tree model to predict "income" instead of the logistic regression model (use trials=30 for boosting multiple decision trees (see an example in slide 44, module 6) . Compare the logistic regression model with the boosted C5.0 model.

Problem2—Predicting Student Performance

For this problem we are going to use UCI's student performance dataset. The dataset is a recording of student grades in math and language and includes attributes related to student demographics and school related features. Click on the above link, then go to "Data Folder" and download and unzip "student.zip". You will be using student-mat.csv file. The goal is to create a regression model to forecast student final grade in math "G3" based on the other attributes.

11. (0.5pt) Read the dataset into a dataframe. Ensure that you are using a correct delimiter to read the data correctly and set the "sep" option in read.csv accordingly.

12. (2pt) Explore the dataset. More specifically, answer the following questions:
    a. Is there any missing values in the dataset?
    b. Which variables are associated with the target variable G3? To answer this question, use appropriate plots and test statistics based on variable types. You can do this one by one for all variables or write a loop that applies appropriate statistic tests based on variable types. Either approach is fine.
    c. Draw a histogram of the target variable "G3" and interpret it.

13. (0.5 pt) Split the data into train and test. Use 80% of samples for training and 20% of samples for testing.

14. set the random seed: set.seed(123)

15. (2 pt) Use caret package to run 10 fold cross validation using linear regression method on **the train data** to predict the "G3" variable . Print the resulting model to see the cross validation RMSE. In addition, take a summary of the model and interpret the coefficients. Which coefficients are statistically different from zero? What does this mean?

Set the random seed again. We need to do this before each training to ensure we get the same folds in cross validation. Set.seed(123) so we can compare the models using their cross validation RMSE.(2 pts) Use caret and leap packages to run a 10 fold cross validation using step wise linear regression method with backward selection **on the train data**. The train method by default uses maximum of 4 predictors and reports the best models with 1..4 predictors. We need to change this parameter to consider all predictors. **So inside your train function, add the following parameter tuneGrid = data.frame(nvmax = 1:n),** where n is the number of variables you use to predict "G3". Which model (with how many variables or nvmax) has the lowest cross validation RMSE? Take the summary of the final model, which variables are selected in the model with the lowest RMSE?

16. (2pt) Which model does better at predicting G3 based on the cross validation RMSE? Get the predictions of this model for the test data and report RMSE.

What to Turn in:

A **Rnotebooke** consisting of your answers to the question as outlined above for problems 1 and 2 together with R code you used to answer each question

Format of the submission

Your submission must be in two formats:

1. **A .html file which contains the preview of your notebook**. When you click on preview in R studio to preview an R notebook, an html file is created in the same directory as your notebook. You must submit this .html file or your submission will not be graded.

2. An .rmd file which containing your R notebook.

Please do not hesitate to email me if you have any question.