Machine Learning Assignment 3

Hands on with Regression (25 points)

Problem1—Predicting number of college applications

Given the college dataset, the goal of this assignment is to predict the number of applications received ("Apps" variable) using the other variables in the dataset. The variables are:

X: Name of the college
Private : Public/private indicator
- Apps : Number of applications received
- Accept : Number of applicants accepted
- Enroll : Number of new students enrolled
- Top10perc : Proportion of New students from top 10% of high school class
- Top25perc : Proportion of New students from top 25% of high school class
- F.Undergrad : Number of full-time undergraduates
- P.Undergrad : Number of part-time undergraduates
- Outstate : Out-of-state tuition
- Room.Board : Room and board costs
- Books : Estimated book costs
- Personal : Estimated personal spending
- PhD : Percent of faculty with Ph.D.'s
- Terminal : Percent of faculty with terminal degree
- S.F.Ratio : Student/faculty ratio
- perc.alumni : Percent of alumni who donate
- Expend : Instructional expenditure per student
- Grad.Rate : Graduation rate

1. (1p) Download the dataset college.csv and explore its overall structure. Get a summary statistics of each variable. Answer the following questions:
    - How many observation do you have in the data?
    - How many categorical and numeric variables you have in your data?
    - Is there any missing value?
2. (1pt) Remove the first column (the name of the college)
3. (2 pts) Which variables are associated with "Apps". Use "pairs", "cor", and side by side box plot with t.test to answer this question.
4. (1pt) plot the histogram of the number of applications "Apps" variable. Explain what the histogram shows?
5. (1pt) replace the "Top10perc" variable with a factor variable "Elite" with two levels: "Yes" if 50% or more of new students coming from the top 10% of their high school class (that is, if Top10Perc>=50) , and "No" if less than 50% of new students are coming from the top10% of their high school class.
6. (1pt) is the Elite variable you created above associated with the "Apps" variable? Don't just say yes or no, you should explain your answer by plots and test statistics.
7. (1pt) normalize all numerical features (except the "Apps" variable) using z-score standardization using "scale" function
8. set the random seed: set.seed(123)
9. (2 pt) Use caret package to run 10 fold cross validation using linear regression method on all features.

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 meant?

10. 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.

11. (3 pts) Use caret and leap packages to run a 10 fold cross validation using step wise linear regression method with backward selection (i.e., method="leapBackward). 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 1..16 predictors. **So inside your train function, add the following parameter tuneGrid = data.frame(nvmax = 1:16)** . Which model (with how many variables or nvmax) has the lowest RMSE. Take the summary of the final model to see which variables are selected in the model with the lowest RMSE?

12. (2 pt) Split the data into train and test (you can use the first 621 rows for training and rest for testing). use "rpart" function to create a regression tree model from the training data. Get the predictions on testing data and compute the RMSE.

Problem2—Predicting loan default

For this problem, you will use the dataset credit.csv used as an example in week 6 lectures (I attached it to this assignment again for your convenience).

1- Set the random seed, set.seed(123), and split the data to train/test as we did in slide 37 of week 6 lectures.

2- (2pt) Train a logistic regression model on the train data using the glm package and use it to predict the default for the test data

3- (2pt)Compare the predictions with the actual default labels in the test data. What is the false positive rate and false negative rate of your model? How does your logistic regression model compare with the classification models we used in week 6 lectures?

4- (6pt) The outcome variable "default" is imbalanced that is, the number of observations with default=NO is double the number of observations with default=Yes. Most classification models trained on imbalanced data are biased towards predicting the majority class and yield a higher classification error on the minority class. One way to deal with class imbalance problem is to downsample the majority class; meaning randomly sample the observations in the majority class to make it the same size as the minority class. While this approach might work for larger dataset with large enough samples in the minority class, for smaller dataset, it will result in loss of information. Another approach to deal with class imbalance is to use oversampling; meaning generating syntactic samples and add it to the minority class to make it the same size as the majority class. A popular method for generating syntactic samples is called SMOTE (Syntactic Minority Oversampling Technique) which uses KNN to generate syntactic samples for the minority class. Please watch this short Youtube video to understand SMOTE, Then install DMwR package in R and use the SMOTE function to generate syntactic training data for the minority class**. Please note that oversampling or under-sampling should only be applied to the training data and the testing data should be untouched**. Use perc.over=100 argument in the SMOTE function and leave the other parameters as default to create a balanced training data. Then train a logistic regression model on this balanced data and evaluate it on your test data. Compare the False Positive and the False Negative Rates of the model trained on the balanced data to your previous model trained on the imbalance data and discuss the differences.

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.