MTH6101 Introduction to Machine Learning

Laboratory week eight

The intention of this laboratory is to do a split of data and fit and compare three different classifiers. We will use the dataset `Default` that has customer default records for a credit card company. The aim is to predict whether a customer will default or not.

1. Open `RStudio` and install the libraries `ISLR` and `cvTools`.

2. We will first create a set of indices for and 80/20 data split of data that will be used for latter modelling. To this end, load the library `cvTools` then **do remember to set** the random seed to a value of zero before creating the folds. Use for this the command `set.seed`. Now create the folds by running the function `cvFolds` for a split of $n = 10000$ values in $K = 5$ folds. Save the output of this command in the variable `CV`.

3. Briefly examine and describe the contents of your newly created variable `CV`.

4. We will use the **first four** folds (values of `CV$which` of $1, 2, 3, 4$) to **train** the models. The **fifth** fold, (value of `CV$which` equal to 5) will be used to **test** and compare models. To this end, associate the index `CV$subsets[CV$which!=5]` to a variable called `Train`, and `CV$subsets[CV$which==5]` to a variable called `Test`.

5. Load the library `ISLR` and examine the dataset `Default`: note how many values are available, note and do a summary of the variables of the dataset, do a pairs plot of the data. Note that the variable of interest is `default`. In addition, you may want to do an advanced pairs plot with `ggpairs` from the library `GGally`.

6. Using the function `glm` and using the training data (`data = Default[Train,]`), build the following logistic (`family = "binomial"`) models:

   - A model to predict `default` as function of `balance` (`default`$\sim$`balance`), stored in `M1`.
   - A model to predict `default` as function of `balance` and `student` (`default`$\sim$`balance+student`), stored in `M2`.
   - A model to predict `default` as function of all the variables (`default`$\sim$`.`), stored in `M3`.

7. Using the function `predict.glm` and using the test data (`newdata =Default[Test,]`), build predictions (`type="response"`) for each of the models `M1,M2,M3` you just built. Store your predictions in variables `P1,P2,P3`

8. Now we are ready to build confusion matrices for each case. Using the variable `Ytrue<-Default[Test,]$default=="Yes"` and the indicators e.g. `Y1<-P1>0.5`, use the command `table` to build a matrix for each model.

9. For each of models `M1,M2,M3` compute performance measures True Positive Rate TPR $= \frac{\text{TP}}{\text{P}}$ and False Positive Rate FPR $= \frac{\text{FP}}{\text{N}}$.

10. (**Extra**) Using a loop, repeat all the computations you have done so that you effectively do a 5-fold crossvalidation for the three models. The only extra ingredient you need is a variable to keep the TPR and FPR for the different models. Then plot TPR vs. FPR in $[0,1]^2$, coloring by model.