16-791 Applied Data Science – Spring 2018, Section A

Homework #2

Reports are due at 12:00pm (noon) on Monday, February 19.

Please upload your report and code on canvas.

Please remember to properly format all documents as described in this document

Individual work will be strictly enforced!

In this assignment, we will practice model selection with k-Nearest-Neighbors, Logistic Regression, Naïve Bayes' and SVM classifiers. The data for this exercise is wine industry data. Each record represents a sample of a specific wine product, the input attributes include its organoleptic characteristics, and the output denotes the quality class of each wine: {high, low}. The labels have been assigned by human wine tasting experts, and we can treat that information as "ground truth" in this exercise. Your job is to build the best model to predict wine quality from its characteristics, so that the winery could replace costly services of professional sommeliers with your automated alternative, to enable quick and effective quality tracking of their wines at production facilities. They need to know whether such change is feasible, and what extents of inaccuracies may be involved in using your tool.

You will need to download the R package e1071 & if working in Python you will require scikit-learn. Please feel free to use an R markdown (.rmd) file or Jupyter notebooks (.ipynb) file to complete your homeworks.

Part 1:
Please write the following R/Python functions:

- get_pred_logreg(train,test) – Logistic Regression [{suggested implementation with R: glm with binomial (logit) model} , {suggested implementation with Python: sklearn.linear_model: LogisticRegression]
- get_pred_svm(train,test) – SVM [{suggested implementation with R: e1071:svm}, {suggested implementation with Python: sklearn.svm:SVC}]
- get_pred_nb(train,test) – Naïve Bayes' [{suggested implementation with R: e1071:naiveBayes}, {suggested implementation with Python: sklearn.naive_bayes: GaussianNB}]
- get_pred_knn (train,test,k) – k-Nearest-Neighbors [{suggested implementation with R: e1071:knn}, {suggested implementation with Python: sklearn.neighbors: KNeighborsClassifier}

Each of the above functions will accept a training set data frame and a testing set data frame. You may assume that the last column of the data is the output dimension, and the output is binary {0,1}. The above functions return data frames with two columns: prediction and true output.

Part 2:

Please write the following R/Python function:

- do_cv_class(df, num_folds, model_name) – a generic function for k-fold cross-validation for selected classification models model_name={logreg, svm, nb, knn}. Return a dataframe (model_function(predict+actual) & folds)

Note: For the use with knn model, please enable parsing of 'model_name' argument to identify the number of neighbors assuming that 'k' in 'knn' encodes that value (so do_cv_class(df,10,5nn) would execute 10-fold cross-validation on frame 'df' using a 5-Nearest-Neighbor classifier). In addition, please use the simple majority voting scheme with knn models.

Part 3:

Please write an R/Python function called 'get_metrics' specified below:

- Inputs:
    - Prediction data frame. The first column of it contains predicted values, the second column represents true labels (0/1)
    - Cutoff. It is a numeric parameter with the default value of 0.5 that specifies the threshold value of a scoring binary classifier output.
- Output:
    - A data frame with elements (tpr, fpr, acc, precision, recall) (respectively: true positive rate, false positive rate, accuracy, precision, recall).

Part 4:

(a) Using do_cv_class on the attached wine data find the parameter k in the kNN model that gives the best generalization. As the number of neighbors varies, can you identify the ranges of k when the models overfit and underfit data?

(b) Using do_cv_class to fit the three types of parametric models for wine data find the type of the model yielding the highest test-set accuracy. What is the accuracy of the default classifier for this data?

(c) Summarize your findings from sub-tasks (a), (b) and (c). Which model appears to work best on this data?

Submission Guidelines

Your submission should include two files:

1. All R or python code (include the code for generating graphs) should be placed in the file named HW2_YourAndrewID.R (or .rmd) or HW2_YourAndrewID.py (or .ipynb) respectively.
    a. If you submit a .R or .py file, **change the file extension to *.txt in order for it to be acceptable to the TurnitIn system.**
    b. If you submit a python or R notebook (.rmd or .ipynb) **ALSO SUBMIT AN HTML VERSION of the notebook.**
2. A pdf file named HW2_YourAndrewID.pdf containing summaries of obtained results for Part 4.