Assignment

Data Mining Assignment

This assignment represents 100% of the Data Mining module's mark. It is composed of Part 1 which is worth 40 marks, and Part 2 which is worth 60 marks. You can work in a team of 2 students for this assignment. One student per team will be chosen by the team as being the team leader – who will be in charge of coordinating the team's work, and of submitting the assignment in their account on VLE on behalf of all the team.

Note on late submissions process: *Late submissions can be considered for approval by the Computing Department only for good reasons and you need to apply (attaching evidence) to the Computing Student Office. The Department can decide to approve an extension. For more enquiries, and to apply, contact the Student Office (Computing Department, 25 St James, email computing@gold.ac.uk*

PART 1:

This task is based on the Sonar real data seen previously in class. Several objects which can be rock or metal cylinders are scanned on different angles and under different conditions, with sonar signals. 60 measurements are recorded per columns for each object (one record per object) and these are the predictors called A1, A2, ..., A60. The label associated with each record contains the letter "R" if the object is a rock and "M" if it is metal cylinder, and this is the outcome variable called Class.

Two datasets are provided to you: a training dataset in the sonar_train.csv file, and a test dataset in the sonar_test.csv file.

a) You are required to write a Python code implementing the Nearest Neighbour algorithm (not kNN), with the Minkowski distance encountered in lecture of week 1. Your code will read the power q appearing in the Mionkowski distance, and will classify each record from the test dataset based on the training dataset, and then will calculate and display the accuracy, sensitivity, specificity and precision of the predictions on the test dataset. Run your code to produce results first for Manhattan distance and then for Euclidian distance, which are particular cases of Minkowski distance (q=1, and q=2, see lecture week 1).

b) Run your code for the power q as a positive integer number from 1 to 20 and display the accuracy, sensitivity and specificity on the test set in a chart. Which value of q leads to the best accuracy on the test set?

The code, comments, explanations and results will be provided in a Jupyter notebook called Part1.

Note that in this task you are not to apply a library for the nearest neighbour algorithm but you are to code yourself this simple algorithm.

PART 2:

This task is based on a real credit risk data, and is to predict not-credible and credible credit card clients. More precisely, the task is to predict a response variable Y which represents a credit card default payment (Yes = 1, No = 0), using the 23 predictor variables as follows:

X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
X2: Gender (1 = male; 2 = female).
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
X4: Marital status (1 = married; 2 = single; 3 = others).
X5: Age (year).
X6 - X11: History of past payment. One tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

Two datasets are provided to you: a training dataset in the creditdefault_train.csv file, and a test dataset in the creditdefult_test.csv file.

Using Python and any relevant libraries, you are required to build the best predictive model by tuning models using cross validation on the training dataset with each of the following algorithms (seen in class): kNN, decision trees, Random Forest, Bagging, Boosting, and SVM. Out of the models tuned with the above algorithms, select the best model and clearly justify your choice, and evaluate its performance on the test set.

The coding, comments and explanations will be provided in your Python Jupyter notebook called *Part2*, which should include also the results. Moreover, for each algorithm mentioned above, include at least 1 chart in the notebook illustrating how the performance of the models vary when you vary the hyper parameters' values.

What and how to submit:

- A Word file with the complete names and student numbers, and email of all students in the team. The contribution of each student should be clearly stated. The team leader will be indicated as such.
- The Part1 Jupyter notebook
- The Part2 Jupyter notebook

Note regarding work in a team:

- In certain cases as for instance you don't find a team to work with etc, you can opt to work and submit alone. In such a case, do mention it in the Word file above, and consider only point (a) in Part 1, and 3 out of the 6 algorithms mentioned in Part 2 (at your choice).
- If you work in a team of 2 people, Part 1 and Part 2 have to be addressed entirely. A practical option to work in a team is to do Skype meetings with your team mates to collaborate - and sharing your computer screen with them is a very useful feature of Skype (help available at https://support.skype.com/en/faq/FA34895/screen-sharing-in-skype)

creditdefault_test.csv	2 March 2020, 8:52 PM
creditdefault_train.csv	2 March 2020, 8:52 PM
sonar_test.csv	2 March 2020, 8:52 PM
sonar_train.csv	2 March 2020, 8:52 PM

Submission status

This assignment will accept submissions from **Monday, 1 March 2021, 9:00 AM**

Submission status	No attempt
Grading status	Not graded
Due date	Friday, 26 March 2021, 5:00 PM
Time remaining	27 days 4 hours
Last modified	-
Submission comments	Comments (0)

◄ Reading list Jump to... Forum ►