Machine Learning and Neural Computing: Data Classification Coursework

Date to be handed in: by 23:30 30/03/2023

Introduction

The data you have been given is extracted from the Audit Data dataset [1]. The Archive of this dataset can be found at `https://archive.ics.uci.edu/ml/datasets/Audit+Data`. Your task is to understand the data structure using the principal component analysis (PCA) and build a support vector machine (SVM) classification model to predict the fraudulent firm based on the present and historical risk factors.

You have been given two data files, which can be obtained from the module site on Canvas. One includes the training set, named *training_audit_data.csv*, and the other, *test_audit_data.csv*, consists of the test data. Each row of the files except for the heading row is one firm's information provided by ten features. The last column of the files, headed as *R*isk, is the class label indicating the firms that resort to high risk of unfair practices.

The training set you have been given consists of 391 instances, 202 labeled as 1, meaning *h*igh risk, and 189 labeled as 0, *low or no risk*. This dataset can be treated as a balanced dataset. The test set contains 376 instances. You can assume that the data is of satisfactory quality and requires no preprocessing/data cleansing other than normalisation.

The ten features of each file are:

- *S*ector_score: Historical risk score value of the sector.
- *L*OCATION_ID: Unique ID of the city/province.
- *P*ARA_A: Discrepancy found in the planned expenditure of inspection and summary report.
- *P*ARA_B: Discrepancy found in the unplanned expenditure of inspection and summary report.
- *T*OTAL: Total discrepancies found in other reports.
- *n*umbers: Historical discrepancy score.
- *M*oney_Value: Amount of money involved in misstatements in the past audits.

- *D*istrict: Historical risk score of a district in the last ten years.
- *LOSS_SCORE*: The score of loss suffered by the firm last year.
- *H*istory: Average historical loss suffered by the firm in the last ten years.

You will use Support Vector Machines (SVMs) to classify the data. The type of SVM you need to use is the C-SVC (Cost-Support Vector Classifier), and the kernel functions you should use are the Gaussian radial basis function (RBF) and the linear kernel function, respectively.

Software Required

For this coursework, you will need to write your Python code (in version 3 and above) in the Jupyter Notebook. You can use functions from the following packages: Numpy, Pandas, Matplotlib, Seaborn, and Sklearn. Your practical session notes should be very useful - these are all available on *Canvas*.

Tasks

1. Task 1 - Data Exploration (13 marks)
   In this task, you need to use Principal Component Analysis (PCA) to understand the characteristics of the datasets.

   (a) Use Pandas to load both the training set and the test set (1 mark). (Let's denote this original training set as training set (I).)

   (b) Plot two scatter plots of 'TOTAL' against 'Money_Value' of the training set. One for the whole range of values, and the other shows values of 'TOTAL' to $[0, 20]$ and 'Money_Value' to $[0, 3]$ only. Use the function: *subplot* to put these two subplots in one figure. You need to label the data using different colours in the picture according to its class and set the feature text for the x-axis and y-axis, separately (4 marks).
   Hint: examples on how to use pyplot.subplot*,* pyplot.xlim*, and* pyplot.ylim *in* matplotlib *can be found from following links, respectively*:
   `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.subplot.html`,
   `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlim.html`, and
   `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylim.html`.

   (c) Normalise the training set and the test set using **StandardScaler()** (Hint: the parameters should come from the training set only) (2 marks).

   (d) Perform a PCA analysis on the scaled training set and plot the scree plot to report variances captured by each principal component (3 marks).

   (e) Plot projections of the training set (I) in the projection space constructed using the first principal component (PC1) and the second principal component (PC2) obtained from the training set (I). You need to label the data using different colours in the picture according to its class and set the label for the x-axis and y-axis, separately (2 marks).

(f) Obtain projections of the test set by projecting the scaled test data on the same PCA space produced by the training set in Task 1 (d) (1 mark).

2. Task 2 (3 marks)

   (a) Divide the training dataset into a smaller training set (II) and a validation set using the **train_test_split** function and report the number of points in each set. Usually, we use 20%-30% of the total data points in the whole training set as the validation data. It is your choice on how to set the exact ratio (2 marks).

   (b) Normalise both the training set (II) and the validation set (Hint: the parameters should come from the training set (II) only)(1 mark).

3. Task 3 - SVM Classification (9 marks)

   (a) Basic task (5 marks)

      i. Choosing the most suitable parameters (3 marks)
      When using the C-SVC SVM, you have been given the following combinations:
      - Gaussian radial basis kernel:
        - $C = 1, \gamma$=10,
        - $C = 1, \gamma$=0.5,
        - $C = 5, \gamma$=10.
      - Linear kernel:
        - $C = 0.1$.

      You should train an SVM model for each combination from the four combinations and then test it on the normalised validation set. The accuracy rate for each combination on the validation set should be reported. Finally, you need to select the best combination of parameters and report your result.

      ii. SVN classification (2 marks)
      You should now be in a position to further test your model with the selected parameters by classifying the test data. With the normalised whole training set (I) as the input, you will need to train an SVM model with the suitable parameter values discovered in Task 3 (a) i. When the classification model is built you will then need to use it to classify the normalised test set, and report the confusion matrix.

   (b) Advanced task - SVM classification with features reduced using PCA(4 marks)

      i. Looking at the scree plot which you have produced in Task 1 (d), how many principal components (PCs) you would like to use to do feature reduction? Explain the reason (1 mark).

      ii. Reduce features for both the normalised training set (I) and the normalised test set using the PCA result from Task 1 with the number of principal components you have decided to use (1 mark).

      iii. Do the classification using an SVM with parameter values selected in Task 3 (a) (2 marks).
      - Normalise the training set and the test set after the feature reduction.

- Train an SVM model on the training set with reduced features.
- Test the model on the corresponding test set that is the one with reduced features, and report the classification result on the test set by showing the confusion matrix.

4. Task 4 - Summarize your findings and write your conclusions in critical thinking. For example, what can you see from those generated figures? Which model gives a better classification result: the one trained on the original features or the one trained on the reduced features? Is this what you expected? Why? You need to provide evidence to support the reasons you give. (3 marks).

What to Submit

- The deliverable for this coursework includes
  - an experimental report with no more than ten pages, including appendix and less than 1500 words (Please use a single column format. Font size should be set to 11 or 12 point, and the line spacing should be set to 1.5 lines or single) in the PDF format.
  - a Jupyter Notebook including all code you have written for this coursework.

  You must submit both files. The experimental report in PDF format will receive the plagiarism review via *Turnitin*; the Jupyter Notebook will be used to check whether the code works. No marks will be awarded if only one of the files is submitted.

- The structure of the experimental report
  - For each task, you may write a section to show the key code, the figures, if there are any, and the results you obtain. However, showing the splitted training, validation, and test sets is unnecessary. All submitted screenshots should be clear and readable.
  - It is not necessary to write a literature review in the report.

Please name both submissions using your student ID. For example, 17000000.pdf and 17000000.ipynb.

Overall, there are **2 marks** for presentation and clarity of the submitted report. Note that you must do this coursework individually. You need to submit your coursework via Canvas to the assignment portal: Data Classification.

Please note that the ′Turnitin Submission for Data Classification′ portal is for you to obtain a text-matching similarity report to improve your academic writing as necessary. You can submit your work to Turnitin once. Please do not submit the work (that is your final submission) that you would like to be marked there.

References

[1] Hooda, Nishtha, S. B., and Rana, P. S. Fraudulent firm classification: A case study of an external audit. *Applied Artificial Intelligence 32* (2018), 48–64.