COMP20008 Project 2

V1.0: September 6, 2019

DUE DATE

The assignment is worth 20 marks, worth (20% of subject grade) **and is due 11:59PM on Thursday 26 September 2019**. Submission is via the LMS. Please ensure you get a submission receipt via email. If you don't receive a receipt via email, this means your submission has not been received and hence cannot be marked. Late penalty structure is described at the end of this document.

This project may be done individually or in groups of two or three. Groups must be registered by Friday 13th September using the following form: https://forms.gle/Q9TeXZchamfPQVhJ6, or we will assume you are completing the project individually.

Introduction

For this project, you will perform a data linkage on two real-world datasets and explore different classification algorithms.

The project is to be coded in Python 3. Seven (7) relevant data sets can be downloaded from LMS:

- datasets for Data Linkage
  - `amazon.csv`
  - `google.csv`
  - `amazon_google_truth.csv`
  - `amazon_small.csv`
  - `google_small.csv`
  - `amazon_google_truth_small.csv`
- dataset for Classification: `all_yeast.csv`

Part 1 - Data Linkage (7 marks)

Amazon and Google both have product databases. They would like to link the same products in order to perform joint market research.

Naïve data linkage without blocking (4 marks)

For this part, data linkage without blocking is performed on two smaller data sets: `amazon_small.csv` and `google_small.csv`.

The Tasks: Using `amazon_small.csv` and `google_small.csv`, implement the linkage between the two data sets and measure its performance. Comment on your choice of similarity functions, method of deriving a final score, and the threshold for determining if a pair is a match.

The performance is evaluated in terms of *recall* and *precision*. Ground truth (true matches) are given in `amazon_google_truth_small.csv`.

$$\begin{array}{lcl} recall & = & tp/(tp+fn) \\ precision & = & tp/(tp+fp) \end{array}$$

where tp (true-positive) is the number of true positive pairs, fp the number of false positive pairs, tn the number of true negatives, and fn the number of false negative pairs. The four numbers should sum up to the total number of all possible pairs from the two datasets. $(n = fp + fn + tp + tn)$

Note: The python package `textdistance` implements many similarity functions for strings (https://pypi.org/project/textdistance/). You can use this package for the similarity calculations for strings. You may also choose to implement your own similarity functions.

Your output for this question should include:

1. Your implemented code that generates matches and the two performance measures. (**3 marks**)
2. A discussion on the linkage method and its performance. Comments should include the choices of similarity functions and final scoring function and threshold and also include comments on overall performance of your method. (**1 mark**)

Blocking for efficient data linkage (3 marks)

Blocking is a method to reduce the computational cost for record linkage.

The Tasks: Implement a blocking method for the linkage of the `amazon.csv` and `google.csv` data sets and report on your proposed method and the quality of the results of the blocking. Ground truths are given in `amazon_google_truth.csv`.

To measure the quality of blocking, we assume that when comparing a pair of records, we can always determine if the pair are a match with 100% accuracy. With this assumption, note the following:

- a record-pair is a true positive if the pair is found in the ground truth set and also the pair are in the same block.

- a record-pair is a false positive if the pair co-occur in some block but are not found in the ground truth set.
- a record-pair is a false negative if the pair do not co-occur not in any block but are found in the ground truth set
- a record-pair is a true negative if the pair do not co-occur not in any block and are also not found in the ground truth set.

Then, the quality of blocking can be evaluated using the following two measures:

$$\begin{aligned} \text{Pair completeness } (PC) &= tp/(tp+fn) \\ \text{Reduction ratio } (RR) &= 1-(tp+fp)/n \end{aligned}$$

where n is the total number of all possible record pairs from the two data sets. ($n = fp + fn + tp + tn$)

Your output for this question should include

1. Your implemented code that generates blocks and the two quality measures. (**2 marks**)
2. A discussion on the blocking method and its quality. Comments should include the choices of blocking method and how the method relates to the two quality measures. (**1 mark**)

Part 2 - Classification (12 marks)

A biological sciences research team is interesting in prediciting the cellular localization sites of proteins in yeast, based on particular attributes. In particular, they wish to separate cytoplasm proteins (CYT) from other proteins (non-CYT). They have collected some (somewhat noisy) data from their current yeast samples. The dataset includes a Class column specifying whether a particular sample relates to a CYT or non-CYT protein, along with other features related to the sample. As data scientists, we wish to help them build and assess a classifier for performing this task.

Pre-processing (2 marks)

- **Impute missing values:** Use replacement by mean and replacement by median to fill in missing values. Display the min, median, max, mean and standard deviation for the data with imputations. Justify which imputation method is more suitable.
- **Scale the features:** Explore the use of two transformation techniques (mean centering and standardisation) to scale all attributes after median imputation, and compare their effects. Display the min, median, max, mean and standard deviation for the both types of data scaling.

Comparing Classification Algorithms (3 marks)

We wish to compare the performance of the following 3 classification algorithms: k-NN (k=5 and k=10) and Decision tree algorithms on the all_yeast dataset.

Use the data, all_yeast, as transformed in your pre-processing (mean centered, median imputed). Train the classifiers using 2/3 of the data and test the classifiers by applying them to the remaining 1/3 of the data. Use each classification algorithm to predict the Class feature of the data (CYT or non-CYT) using the first 8 features (mcg-nuc).

What are the algorithms' accuracies on the test data? Which algorithm performed best on this dataset? Explain your experiments and the results.

Feature Engineering (7 marks)

In order to achieve higher prediction accuracy for k-NN, one can investigate the use of feature generation and selection to predict the Class feature of the data. Feature generation involves the creation of additional features. Two methods are

- Interaction term pairs. Given a pair of features f_1 and f_2, create a new feature $f_{12} = f_1 \times f_2$ or by $f_{12} = f_1 \div f_2$. All possible pairs can be considered.
- Clustering labels: apply k-means clustering to the data in all_yeast data and then use the resulting cluster labels as the values for a new feature $f_{clusterlabel}$. At test time, a label for a testing instance can be created by assigning it to its nearest cluster.

Given a set of N features (the original features plus generated features), feature selection involves selecting a smaller set of n features (n < N). Here one computes the mutual information between each feature and the class label (on the training data), sorts the features from highest to lowest mutual information value, and then retains the top n features from this ranking, to use for classification with k-NN.

Your task in this question is to evaluate whether the above methods for feature generation and selection, can deliver a boost in prediction accuracy compared to using k-NN just on the original features in all_yeast. You should:

- Implement the above two methods for feature generation. You should experiment with different parameter values, including k and different numbers of generated features.
- Implement feature selection using mutual information. Again you should experiment with different parameter values, such as how many features to select.

Your output for this question should include

1. Your implemented code that generates accuracies or plots for classification using interaction term pairs (**3 marks**)
2. Your implemented code that generates accuracies or plots for classification using clustering labels (**1 mark**)
3. A discussion (2-3 paragraphs) including:

- Which parameter values you selected and why. (**1 mark**)
- Whether feature selection+generation with interaction term paris can deliver an accuracy boost, based on your evidence from part 1. (**1 mark**)
- Whether feature generation with clustering labels can deliver an accuracy boost, based on your evidence from part 2. (**1 mark**)

Marking scheme

Correctness (19 marks): For each of the questions, a mark will be allocated for level of correctness (does it provide the right answer, is the logic right), according to the number in parentheses next to each question. Note that your code should work for any data input formatted in the same way as `all_yeast.csv`, `amazon.csv`, `google.csv`, and `amazon_google_truth.csv`. E.g. If a random sample was taken from `all_yeast.csv:`, your code should provide a correct answer if this was instead used as the input.

Correctness will also take into account the readability and labelling provided for any plots and figures (plots should include title of the plot, labels/scale on axes, names of axes, and legends for colours symbols where appropriate).

Coding style (1 mark): A Mark will be allocated for coding style. In particular the following aspects will be considered:

- Formatting of code (e.g. use of indentation and overall readability for a human)
- Code modularity and flexibility. Use of functions or loops where appropriate, to avoid highly redundant or excessively verbose definitions of code.
- Use of Python library functions (you should avoid reinventing logic if a library function can be used instead)
- Code commenting and clarity of logic. You should provide comments about the logic of your code for each question, so that it can be easily understood by the marker.

Resources

The following are some useful resources, for refreshing your knowledge of Python, and for learning about functionality of pandas.

- Python tutorial
- Python beginner reference
- pandas 10min tutorial
- Official pandas documentation
- Official mathplotlib tutorials
- Python pandas Tutorial by Tutorialspoint

- pandas: A Complete Introduction by Learn Data Sci
- pandas quick reference sheet
- sklearn library reference
- NumPy library reference
- Textdistance library reference
- Python Data Analytics by Fabio Nelli (available via University of Melbourne sign on)

Submission Instructions

All code and answers to discussion questions should be contained within a single Jupyter Notebook file, which is to be submitted to the LMS by the due date. The first line of your file should clearly state the full names, login names and Student IDs of each member of your group. If you are in a group, please ensure only **one** student submits on behalf of the group.

Other

Extensions and Late Submission Penalties: All requests for extensions must be made by email, sent to the head tutor Chris Ewin and CC'd to the lecturer Pauline Lin. If requesting an extension due to illness, please attach a medical certificate or other supporting evidence. All extension requests must be made at least 24 hours before the due date. Late submissions without an approved extension will attract the following penalties

- $0 < hourslate <= 24$ (2 marks deduction)
- $24 < hourslate <= 48$ (4 marks deduction)
- $48 < hourslate <= 72$: (6 marks deduction)
- $72 < hourslate <= 96$: (8 marks deduction)
- $96 < hourslate <= 120$: (10 marks deduction)
- $120 < hourslate <= 144$: (12 marks deduction)
- $144 < hourslate$: (20 marks deduction)

where $hourslate$ is the elapsed time in hours (or fractions of hours).

This project is expected to require 20-25 hours work.

Academic Honesty

You are expected to follow the academic honesty guidelines on the University website `https://academichonesty.unimelb.edu.au`

Further Information

A project discussion forum has also been created on the subject LMS. Please use this in the first instance if you have questions, since it will allow discussion and responses to be seen by everyone. The teaching team will support the discussions on the forum until 24 hours prior to the project deadline. There will also be a list of frequently asked questions on the project page.