COMP20008 Project 2

V1.0: September 29, 2020

DUE DATE

The assignment is worth 25 marks, worth (25% of subject grade) **and is due 8:00AM on Wednesday 21st October 2020**. Submission is via the LMS. Late penalty structure is described at the end of this document.

Introduction

This project is designed to give you practice at solving some practical data science tasks. You will need to implement and evaluate a data linkage system and a classification system using sound data science principles. We intentionally do not prescribe the exact steps required as a key part of the assessment is to evaluate your ability to exercise your own judgement in deciding how to solve these practical data science problems, rather than simply implementing a pre-defined algorithm. This is a big part of any real-world data science task, as there are usually many approaches that can be taken and many irregularities in the data that are impossible to exhaustively predefine in a task description. A data scientist needs to be able to select the best approach for resolving these issues and justify it in order for their results to be convincing.

For this project, you will perform a data linkage on two real-world datasets (Part 1) and explore different classification algorithms (Part 2).

The project is to be coded in Python 3. Eight (8) relevant data sets can be downloaded from LMS and are available on Juptyer Hub:

- datasets for Part 1: Data Linkage
  - `abt.csv`
  - `buy.csv`
  - `abt_buy_truth.csv`
  - `abt_small.csv`
  - `buy_small.csv`
  - `abt_buy_truth_small.csv`
- datasets for Part 2: Classification:
  - `life.csv`
  - `world.csv`

Part 1 - Data Linkage (12 marks)

Abt and Buy both have product databases. In order to perform joint market research they need to link the same products in their databases. Therefore the research team has manually linked a subset of the data in order to build a linkage algorithm to automate the remaining items to be linked. This manually linked data is what you will base your work on in this assignment. However the real dataset is unknown to you, as it would be reality and this unknown data is what you will be assessed on.

Naïve data linkage without blocking (4 marks)

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

Task - 1A: Using `abt_small.csv` and `buy_small.csv`, implement the linkage between the two data sets.
Your code for this question is to be contained in a single Python file called `task1a.py` and produce a single csv file `task1a.csv` containing the following two column headings:

```
idAbt,idBuy
```

Each row in the datafile must contain a pair of matched products. For example, if your algorithm only matched product `10102` from the Abt dataset with product `203897877` from the Buy dataset your output `task1a.csv` would be as follows:

```
idAbt, idBuy
10102,203897877
```

The performance is evaluated in terms of *recall* and *precision* and the marks in this section will be awarded based on the two measures of your algorithm.

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

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.

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.

Blocking for efficient data linkage (4 marks)

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

Task - 1B: Implement a blocking method for the linkage of the `abt.csv` and `buy.csv` data sets.

Your code is be contained in a single Python file called `task1b.py` and must produce two csv files `abt_blocks.csv` and `buy_blocks.csv`, each containing the following two column headings:

```
block_key, product_id
```

The `product_id` field corresponds to the `idAbt` and `idBuy` of the `abt.csv` and `buy.csv` files respectively. Each row in the output files matches a product to a block. For example, if your algorithm placed product `10102` from the Abt dataset in blocks with block_keys `x` & `y`, your `abt_blocks.csv` would be as follows:

```
block_key, product_id
x,10102
y,10102
```

A block is uniquely identified by the `block_key`. The same block_key in the two block-files (`abt_blocks.csv` and `buy_blocks.csv`) indicates that the corresponding products co-occur in the same block.
For example, if your algorithm placed the Abt product `10102` in block `x` and placed Buy product `203897877` in block `x`, your `abt_blocks.csv` and `buy_blocks.csv` would be as follows respectively:

```
abt_blocks.csv:
    block_key, product_id
    x,10102

buy_blocks.csv:
    block_key, product_id
    x,203897877
```

The two products co-occur in the same block `x`.

To measure the quality of blocking, we assume that when comparing a pair of records, the pair are always 100% similar and are a match. A pair of records are categorised as follows:

- a record-pair is a true positive if the pair are found in the ground truth set and also the pair co-occur 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 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 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:

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

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

The marks in this section will be awarded based on the pair completeness and reduction ratio of your blocking algorithm.

Note: The time taken to produce your blocking implementation must be linear in the number of items in the dataset. This means that you cannot, for example, compare each product to every other product in the dataset in order to allocate it to a block. Implementations that do so will be severely penalised.

Report on the Data Linkage project(4 marks)

Task - 1C Write a one-page report describing your algorithms and implementations of tasks 1a and 1b. You should discuss:

- How your product comparison works, including your choice of similarity functions and final scoring function and threshold.
- An evaluation of the overall performance of your product comparison and what opportunities exist to improve it.
- How your blocking implementation works.
- An evaluation of the overall performance of your blocking method, how the method relates to the performance measures and what opportunities exist to improve it.

Your report for this task should be contained in a single file called `task1c.pdf` or `task1c.docx`.

Part 2 - Classification (13 marks)

Each year, the World Bank publishes the World Development Indicators which provide high quality and international comparable statistics about global development and the fight against poverty [1]. As data scientists, we wish to understand how the information can be used to predict average lifespan in different countries. To this end, we have provided the `world.csv` file, which contains some of the World Development Indicators for each country and the `life.csv` file containing information about the average lifespan for each country (based on data from the World Health Organization) [2]. Each data file also contains a country name, country code and year as identifiers for each record. These may be used to link the two datasets but should not be considered features.

Comparing Classification Algorithms (3 marks)

Task - 2A Compare the performance of the following 3 classification algorithms: k-NN (k=3 and k=7) and Decision tree (with a maximum depth of 3) on the provided data. You may use sklearn's KNeighborsClassifier and DecisionTreeClassifier functions for this task. To ensure consistency, please ensure that all functions that can be called with a random_state parameter use a random_state of 200. For the k-NN classifier, all parameters other than k should be kept at their defaults. For the Decision tree classifier, all parameters other than the maximum depth and random_state should be kept at their defaults. Use each classification algorithm to predict the class feature `life expectancy at birth(years)` of the data (`Low`, `Medium` and `High` life expectancy) using the remaining features.

Organise and verify your data: Before you begin you should ensure that your dataset is sorted in ascending alphabetical order by country and that any countries not present in both `world.csv` and `life.csv` are discarded.

For each of the algorithms, fit a model with the following processing steps:

- Split the dataset into a training set comprising 70% of the data and a test set comprising the remaining 30% using the `train_test_split` function with a random_state of 200.
- Perform the same imputation and scaling to the training set:
  - For each feature, perform median imputation to impute missing values.
  - Scale each feature by removing the mean and scaling to unit variance.
- Train the classifiers using the training set
- Test the classifiers by applying them to the test set.

Your code must produce a CSV file called `task2a.csv` describing the median used for imputation for each feature, as well as the mean and variance used for scaling, all rounded to three decimal places. The CSV file must have one row corresponding to each feature. The first three lines of the output should be as follows (where x is a number calculated by your program):

```
feature, median, mean, variance
Access to electricity, rural (% of rural population) [EG.ELC.ACCS.RU.ZS], x, x, x
Adjusted savings: particulate emission damage (% of GNI) [NY.ADJ.DPEM.GN.ZS], x, x, x
```

Your code must print the classification accuracy of each classifier to standard output. Your output should look as follows (where the # symbol is replaced by the accuracy of each algorithm, rounded to 3 decimal places):

```
Accuracy of decision tree: #
Accuracy of k-nn (k=3): #
Accuracy of k-nn (k=7): #
```

Your code for this question should be contained in a single Python file called `task2a.py`

Feature Engineering and Selection(6 marks)

Task - 2B This task will focus on k-NN with k=3 (from here on referred to as 3-NN). In order to achieve higher prediction accuracy for 3-NN, one can investigate the use of feature engineering and selection to predict the class feature of the data. Feature generation involves the creation of additional features. Two possible methods are:

- Interaction term pairs. Given a pair of features f_1 and f_2, create a new feature $f_{12} = f_1 \times f_2$. All possible pairs can be considered.
- Clustering labels: apply k-means clustering to the data in `world` and then use the resulting cluster labels as the values for a new feature $f_{clusterlabel}$. You will need to decide how many clusters to use. 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).

An alternative method of performing feature engineering & selection is to use Principal Component Analysis (PCA). The first n principal components can be used as features.

Your task in this question is to evaluate how the above methods for feature engineering and selection affect the prediction accuracy compared to using 3-NN on a subset of the original features in `world`. You should:

- Implement feature engineering using interaction term pairs and clustering labels. This should produce a dataset with 211 features (20 original features, 190 features generated by interaction term pairs and 1 feature generated by clustering). You should (in some principled manner) select 4 features from this dataset and perform 3-NN classification.
- Implement feature engineering and selection via PCA by taking the first four principal components. You should use only these four features to perform 3-NN classification.
- Take the first four features (columns D-G, if the dataset is opened in Excel) from the original dataset as a sample of the original 20 features. Perform 3-NN classification.

Your output for this question should include:

- Any text, numbers, or other numerical data you reference in your report, printed to standard output
- Any graphs or charts as `png` files with the prefix `task2b` (e.g. `task2bgraph1.png`, `task2bgraph2.png`)
- The classification accuracy for the test set for of the three methods in the following format, as the last three lines printed to standard output (where the # symbol is replaced by the accuracy of 3-NN using each feature set, rounded to 3 decimal places):

```
Accuracy of feature engineering: #
Accuracy of PCA: #
Accuracy of first four features: #
```

Note: For Task 2B you do **not** need to use a random_state of 200.

Your code for this question should be contained in a single Python file called `task2b.py`

Your Report (4 marks)

Task - 2C Write a 1-2 page report describing your implementations. You should discuss:

- Which algorithm (decision trees or k-nn) in Task-2A performed better on this dataset? For k-nn, which value of k performed better? Explain your experiments and the results.
- A description of the precise steps you took to perform the analysis in Task-2B.
- The method you used to select the number of clusters for the clustering label feature generation and a justification for the number of clusters you selected.
- The method you used to select four features from the generated dataset of 211 features for your analysis and a justification for this method.
- Which of the three methods investigated in Task-2B produced the best results for classification using 3-NN and why this was the case.
- What other techniques you could implement to improve classification accuracy with this data.
- How reliable you consider the classification model to be.

Your report for this task should be contained in a single file called `task2c.pdf` or `task2c.docx`.

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

Your code should be contained in four Python files, `task1a.py`, `task1b.py`, `task2a.py` and `task2b.py` which are to be submitted to the LMS by the due date. Your two reports `task1c` and `task2c` should be also be submitted to the LMS by the due date.

Other

Extensions and Late Submission Penalties: All requests for extensions must be made by email, sent to Chris Ewin. 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)
- $144 < hourslate$: (25 marks deduction)

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

Silent Policy: A silent policy will take effect 48 hours before this assignment is due. This means that no question about the assignment will be answered, whether it is asked on the

discussion forum or via email. By this stage, you should have a clear idea of what the assignment is about and have figured out any issues that require staff input.

This project is expected to require 30-35 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 Ed forum. 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.

References

[1] The World Bank, "World development indicators," 2016, data retrieved from World Development Indicators, https://databank.worldbank.org/source/world-development-indicators.

[2] World Health Organization, "Life expectancy and healthy life expectancy data by country," 2016, data retrieved from Global Health Observatory data repository, https://apps.who.int/gho/data/node.main.688?lang=en.