COMP5318 - Machine Learning and Data Mining

Assignment 1

Due: 07 May 2017, 5:00PM

**Summary**

The goal of this assignment is to build a classifier to classify apps from the Apps Market into a set of categories based on their descriptions. The dataset is quite large, so you need to be smart on which method you gonna use and perhaps perform a pre-processing step to reduce the amount of computation. Part of your marks will be a function of the performance of your classifier on the test set.

**1. Data set description**

The dataset is collected from the Apps Market. There are four main ﬁles:

**1. training\_data.csv:**

* There are 20,104 rows; each row corresponds to an app.
* For each row, each column is separated by comma (,). The ﬁrst column is the app’s name, with the remaining columns containing the tf-idf values. The tf-idf values are extracted from words in the description of each app. We have done some pre-processing steps which resulted in 13,626 unique words. If a word is found in the description of an app, it has a tf-idf value (the tf-idf value is not zero). On the other hand, its tf-idf value is equal to zero if the word is not found in the description of the app. More information about tf-idf could be found in <http://en.wikipedia.org/wiki/Tf%E2%80%93idf>
* In summary, data train.txt is a matrix with dimension: 20,104×13,627 (remember the ﬁrst column is the app’s name).

**2. training\_desc.csv:**

* There are 20,104 rows; each row is for an app.
* For each row, each column is separated by comma (,). The ﬁrst column is the app’s name and the second column contains the app’s description.

**3. training\_labels.csv:**

* There are 20,104 rows; each row is for an app.
* For each row, each column is separated by comma (,). The ﬁrst column is the app’s name and the second column is for the label.
* There are 30 unique labels in total, for example Casual, Health and Fitness, etc.

Note that it is not necessary that the same rows of two training ﬁles refer to the same app. Please use the app’s name as a reference.

**4. test\_data.csv:**

* This is a subset of the original data set; we have split the original data set into 90% for training set and 10% for test set (per label). This ﬁle should NOT be used for training the classiﬁer.
* Your code must be able to read the test set, and output a ﬁle “predicted\_labels.csv” in the same data-format as “training\_labels.csv”. Make sure the predictions (classiﬁcation results for the test set) are in the same order as test inputs, i.e. the ﬁrst row of “predicted\_labels.csv” corresponds to the ﬁrst row of “test\_data.csv” and so on).
* The score will be based on how accurate your approach is. We will collect “predicted\_labels.csv” and compare it to the actual labels to get the accuracy of your approach. For further testing purposes, we may use a diﬀerent test set while grading.

**2. Task description**

Each group consists of 2 or 3 students. Your task is to determine / build classiﬁer for the given data set and write a report. The score allocation is as follows:

* Code: max 20 points
* Description: max 80 points

Please see section 5 for the detailed marking scheme. The report and the code are to be submitted to your tutor by the due date.

**2.1 Programming languages and libraries**

You are allowed to use Python3 only.

Although you are allowed to use external libraries for optimization and linear algebraic calculations, you are NOT allowed to use external libraries for basic pre-processing and classiﬁcation. For instance, you are allowed to use scipy.optimize for gradient descent or scipy.linalg.svd for matrix decomposition. However, you are NOT allowed to use sklearn.svm for classiﬁcation (i.e. you have to implement the classiﬁer yourself, if required). If you have any ambiguity whether you can use a particular library or a function, please post on canvas under the “Assignment 1” discussion board.

**2.2 Performance evaluation**

We expect you to have a rigorous performance evaluation and a discussion. To provide an estimate of the performance (precision, recall, F-measure, etc.) of your classiﬁer in the report, you can perform a 10-fold cross validation on the training set provided and average the metrics for each fold.

**3. Instructions to form groups**

You should form groups on your own. Once you have found 1 or 2 other group members, go to Canvas and join a group under “People” tab in Canvas. Select an empty group and join.

* Do not form a group if you do not know any other person who would join you (email Niku and she will allocate group members to you randomly if you don’t have a group).
* Do not join groups that you have not confirmed with other members to be part of.

**4. Instructions to hand in the assignment**

1. The assignment must be handed over in a Google Drive folder. (You need the shareable link for the folder).

2. You should name your folder in this format: “Assignment1\_unikey1\_unikey2\_unikey3” . replace unikey1, unikey2 and unikey3 with the unikeys of your group members. If you are 2 members, you will not have unikey3.

3. You folder must include the following:

i. Assignment 1 ipynb file (a .ipynb ﬁle)

The report should include each member’s details (student ID and name).

ii. Data (a sub-folder)

Including 4 files mentioned in section 1.

iii. output - a file named: “predicted\_labels.csv” -

We will use this ﬁle for grading.

4. Submit the assignment in two steps:

1. Share your assignment folder with **comp5318.students@gmail.com**

2. Submit the link to your Google Drive folder on Canvas to submit the assignment.

5. Your submission should include the report and the code. A plagiarism checker will be used.

6. The report must clearly show

(i) details of your classiﬁer,

(ii) the results from your classiﬁer, including precision and recall results on the training data,

(iii) run-time, and

(iv) hardware and software speciﬁcations of the computer that you used for performance evaluations.

7. A penalty of MINUS 20 points per each day after the due date.

8. Remember, the due date to submit them on to your tutor is 07 May 2017, 5:00PM.

**IMPORTANT: DON'T MODIFY your folder and files after 5:00 pm on 7th of May.**

The timestamp of your final edit (regardless of WHAT kind of edit it is) will be taken as your submission time.

This means that if you submit on 1:00 pm on 7th of May, but edit the file at 5:30 pm to correct a misspelling in your first name, the submission of your assignment is considered at 5:30 pm and you will lose 20% of the marks.

**5. Marking scheme**

|  |  |
| --- | --- |
| Category | Criterion |
| Description [80] | Introduction [5]   * What is the aim of the study? * Why is this study important?   Methods [20]   * Pre-processing (if any) * Classiﬁer   Experiments and results [25]   * Accuracy * Extensive analysis   Discussion [10]   * Meaningful and relevant personal reﬂection   Conclusions and future work [5]   * Meaningful conclusions based on results * Meaningful future work suggested   Presentation [8]   * Academic style, grammatical sentences, no spelling mistakes * Good structure and layout, consistent formatting * Appropriate citation and referencing   Other [7]   * At the discretion of the marker: for impressing the marker, excelling expectation, etc. Examples include fast code, etc. |
| Code [20] | * Code runs and classiﬁes within a feasible time * Well organized, commented and documented |
| Penalties [−] | * Badly written code: [−20] * Not including instructions on how to run your code: [−30] * Late submission: [−20] for each day late |

Note: Marks for each category is indicated in square brackets. The minimum mark for the assignment will be 0 (zero).