Big Data Mining Techniques (M161)
Winter Semester 2021-2022

Deadline: Last Day Before Exam Period
Assignment for teams of 2 students

Goal

The purpose of the project is to familiarize you with the basic steps of the process followed for applying data mining techniques, namely: collection, preprocessing / cleaning, conversion, application of data mining techniques and evaluation. Implementation will be done in the Python programming language using the SciKit Learn tool. The project consists of three (3) tasks related to categorization and nearest neighbors. Three (3) separate competitions have been created for the requirements of the job on the Kaggle platform. You will need to sign up in the Kaggle platform using your academic email (STUDENT_ID@di.uoa.gr) and upload the output files with the predictions. The Kaggle platform provides you with 42 hours of GPU usage if you want to speed up your calculations with neural networks. Pay special attention to the report because the work is first graded by the quality of the documentation.

Requirement 1: Text classification

Description

The requirement is related to text classification of news articles. The data are organized in CSV files whose fields are separated by the '|' character. There are two files:

1. train_set.csv (111795 items): This file will be used to train your algorithms and contains the following fields:
    a. Id: A unique number for the article
    b. Title: The title of the article
    c. Content: The content of the article
    d. Label: The category to which the article belongs
    e. Target: Numerical representation of label

2. test_set.csv (47912 items): This file will be used to predict new (unseen) data. It contains the same fields as in the training file except from the Label field. You will be asked to predict this field using classification algorithms.

There are 4 categories of articles and they are presented in the table below.

Business
Entertainment
Health
Technology

Question 1: Get to know the Data: WordCloud

You are required to create a word cloud for each article category. That is, a word cloud for the “Business” category, one for the “Entertainment” category, etc. For creating a word cloud you will use all the articles in each category. The purpose of word cloud is to provide a general description of the category. An example of a word cloud is shown in the following image. You can use any Python library you want to create the word cloud. **Your report should include an image for each category and a very brief description.**

Question 2: Classify the Test Set

In this part of the assignment you need to build a machine learning model that is capable of identifying the category of an article using text. The goal is to familiarize yourself with the scikit-learn library. For the purposes of this exercise a Kaggle competition is launched. The deliverable of this question is a brief description of the following is necessary:

- Preprocessing you performed
- The models you training
- The features you used
- The parameters you selected

Important: 1. Use the K-Nearest Neighbor classifier as your classification approach (using the Jaccard coefficient as described in the following question).

2. You need to predict the ‘Target’ variable and not the ‘Label’. A sample submission is present in Kaggle.

When you have prepared your classification model, go to the Kaggle Competition and submit your solution.

Kaggle Competition: https://www.kaggle.com/c/bigdata-2022-part-classification/data

Note: It is necessary to upload your solution in Kaggle in order to receive the full mark for this question.

Output Files

You should use your classifier and create the file classifications.csv containing the test set csv predictions. The file format is CSV and is shown below:

Id	Predicted
1	0
2	4
...	…

Hints

- Use the data sketch library
- (extra credit) A very good solution can be achieved using other classical machine learning classification approaches.
- Experiment with the Keras library.
- Experiment with the Hugging Face Transformers library.
- Use Google Colab.

Requirement 2: Nearest Neighbor Search with Locality Sensitive Hashing

Question 1: Nearest Neighbor Search without and with Locality Sensitive Hashing

Description

In this question you will be given a train set file with small texts. Every text is an IMDB review. You will also be given a test set in the same format. The purpose of requirement 2 is to speed up the K-NN classification (where K=15) method using the LSH technique.

You will compare the brute-force method, where each document in the test-set is compared to each document in the train-set, with the approach that we use LSH first to identify candidate pairs of one train-set document and one test-set document where the similarity is expected to be more than a threshold (start with a threshold value of τ=0.8). So in the LSH case you will only compute the actual similarity between two documents if the expected similarity is above the threshold.

You have to consider the following metric for finding the most similar documents: **Jaccard Similarity**. For the LSH implementation use the Min-Hash LSH family and set the number of permutations to {16,32,64}

Evaluation Results

You need to evaluate the performance of the LSH algorithm and you should report:

1. The total LSH Index Creation Time (BuildTime)
2. The total time it took to answer all the test set questions. (QueryTime)
3. TotalTime: BuildTime + QueryTime
4. The fraction of the true K-most similar documents (that is, the ones that the brute force method returns) that the LSH method also returns.

In your report should include a table as follows:

Type	**BuildTime**	**QueryTime**	**TotalTime**	**fraction of the true K most similar documents that are reported by LSH method as well**	**Parameters (different row for different K or for different number of permutations, etc)**
Brute-Force-Jaccard	0	300	300	100%	-
LSH-Jaccard	100	50	150	80%	Perm=16

...	...	...	...	...	...

Things to Consider:

1. Try to use vectorized operations.
2. You can use available implementations of the LSH families
3. Use http://ekzhu.com/datasketch/lsh.html

Classify The Reviews Using KNN

After you have prepared your report regarding the time performance of the LSH algorithm you are expected to use your LSH-KNN algorithm in order to classify the sentiment of the reviews in the test set. Similarly to the Requirement-1, Question-2, a Kaggle competition is also created. You need to predict each entry in the test-set using only the KNN-LSH implementation. More complex ML models should **not** be used.

The Kaggle Competition is available at: https://www.kaggle.com/c/bigdata2022-part-imdb

Note: It is necessary to upload your solution in Kaggle in order to receive the full mark for this question.

Output Files

You should use your LSH model and create the file classifications.csv containing the test set KNN predictions. The file format is CSV and is shown below:

Id	Predicted
1	1
2	0
...	…

Hint:

- Use the data sketch library

Requirement 3: Time Series Similarity

Question a: Dynamic Time Warping

In this question you are required to **implement** the algorithm Dynamic Time Warping(DTW) in order to compute the similarities between time series of different time resolutions. Usage of an existing implementation is not allowed. For the purposes of this task a Kaggle Competition has been created. The competition provides you with a test dataset where each row contains two time series: (a) seq_a and (b) seq_b. The goal is to calculate the distance between the time series using DTW with euclidean distance and to provide an answer to the Kaggle challenge. You are required to also provide your implementation in your report and also to provide the time required in order to estimate all the test set.

The Kaggle competition is available at: https://www.kaggle.com/c/bigdata2022-uoa-part-dtw/

Output Files

You should use your LSH model and create the file dtw.csv containing the test set KNN predictions. The file format is CSV and is shown below:

id	distance
1	0
2	100.0
...	…

Note: It is necessary to upload your solution in Kaggle in order to receive the full mark for this question.

Regarding the deliverables

The folder you deliver should have the name:
Ass1_name1_AM1_name2_AM2.

Kaggle Submission:

- Each team should have a simple account for submitting the solutions with the pattern Surname1_Surname2.
- It is necessary to upload a solution for every Kaggle Competition

The folder should contain:

1. A text with detailed analysis on the experiments you did and the methods you tried in PDF format. Your report should also contain all the tables and plots requested and should not exceed 30 pages. In the report you should include a description of your

experiments and everything you can think of to show what experiments you did, why the specific results of the methods you selected, how these methods work, and commentary on your results. **All tasks will be evaluated on the basis of the detailed documentation and the extent to which the tasks are being implemented.**

2. The requested output files.
3. The source code files.

Datasets

Available at:

- https://www.kaggle.com/c/bigdata-2022-part-classification/ (1b)
- https://www.kaggle.com/c/bigdata2022-part-imdb (2b)
- https://www.kaggle.com/c/bigdata2022-uoa-part-dtw (3a)