Big Data Mining Techniques (M161)
Winter Semester 2023-2024

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 and Keras tool. The project consists of three (3) tasks related to categorization, nearest neighbors and duplication detection. Two (2) separate competitions have been created for the requirements of the project 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.

Part 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.

2. test_set.csv (47912 items): This file will be used to predict in 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.

The dataset is openly available at the address:
https://www.kaggle.com/competitions/bigdata2023classification/data.

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

Business
Entertainment
Health
Technology

Question 1.1: 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 1.2: Classification Task

In this question, you should try both classification methods shown below:

- Support Vector Machines (SVM)
- Random Forests

You should use the features below to evaluate the models mentioned before:

- Bag of Words (BoW)
- SVD

Also, you should evaluate and report the performance of every model + feature combination with 5-fold Cross Validation using:

- Accuracy
- Precision
- Recall

Beat the Benchmark

You should select and experiment with any classification algorithm, preprocess steps in order to beat the performance of the best performing model of the previous question. you should justify the methodology that you choose to follow.

Evaluation Results

The report should include the following table with the evaluation of your techniques using the train-set and 5-Fold Cross Validation.

Statistic Measure	SVM (BoW)	Random Forest (BoW)	SVM (SVD)	Random Forest (SVD)	My Method
Accuracy					
Precision					
Recall					

A description of the above results should be included in the report.

Output File

Your code should create the output file "testSet_categories.csv" which will contain the predictions for articles in the test set dataset (the ones where the Label field is not given). You should use your best model. The format of the testSet_categories.csv file, which will contain the categories of articles given in the Test set, is shown below:

Id	Predicted
1	Business
2	Technology
...	

For the file "testSet_categories.csv" the above formatting should be used *strictly* separating the two fields with the comma (",") character and should also have the first line with the two field names (Id and Predicted) followed by your model predictions in the following lines specifying the article Id from the test set and the predicted label.
You will need to upload your file to the Kaggle contest at the address http://www.kaggle.com/competitions/bigdata2023classification**.**

Hint:

1. Because the text files are large see: https://scikit-learn.org/0.15/modules/scaling_strategies.html.
2. Use the Kaggle computing resources if you want to try out complex neural networks. (This is outside the scope of the course).
3. For computing the evaluation metrics: Precision/Recall you will not use the number of examples in each class (Macro).

Part 2: Nearest Neighbor Search with Locality Sensitive Hashing

Question 2.1: 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 a document. You will also be given a test set in the same format.

Similarly to part 1, the data are organized in CSV files whose fields are separated by the '|' character. There are two files:

3. 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.

4. test_set.csv (47912 items): This file will be used to predict in 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.

The dataset is openly available at the address:
https://www.kaggle.com/competitions/bigdata2023classification/data

The purpose of part 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.

Part 3: Nearest Neighbor Search and Duplicate Detection

Question 3.1: De-Duplication with Locality Sensitive Hashing

Description

In this question you will be given a train set file with small texts. Every text is a Quora question. You will also be given a test set in the same format. The purpose of this question is to find how many of the documents in the test-set already exist in the train-set. As duplicate we define document pairs with similarity more than a threshold τ=0.8. You have to search for duplicate documents using the appropriate LSH family in order to reduce the time required for the detection.

The dataset is openly available at the address:
https://www.kaggle.com/competitions/bigdata2023duplicatedetection/data.

You have to do that considering the metrics:

1. Cosine Similarity: Random projection LSH family. Set the parameter K from 1 to 10.
2. Jaccard Similarity: Min-Hash LSH family. Set 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 number of duplicates in the test-set.

In your report should include a table as follows:

Type	BuildTime	QueryTime	TotalTime	#Duplicates	Parameters
Exact-Cosine	0	600	600	1000	-
Exact-Jaccard	0	300	300	1500	-
LSH-Cosine	30	200	230	800	K=2
LSH-Cosine	50	150	200	600	K=3
LSH-Jaccard	100	50	150	900	-

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.

Extra Credit: Question 3.2: Same Question Detection

Description

In this question you should find if questions with similar format ask the same thing. Consider the example:

1. What restaurants should I visit during my holidays trip in **Dublin**?
2. What restaurants should I visit during my holidays in **Athens**?

Clearly, the above two questions share (9) words but the question is not the same. Specifically you will be given pairs of Quora questions and you need to find out if the pair contains questions that ask the same. In other words, you want to create a model that can answer if two questions are ultimately identical. In this question you should experiment with heuristic features that could solve the above problem. **In your report you should describe in detail the similarity features you selected and how you trained your algorithm.**

For this question, there are two files:

1. train_set.csv (283013 pairs): This file will be used to train your algorithms and it contains the following fields:
    a. Id: A unique number for the pair.
    b. Question1: The first question.
    c. Question2: The second question.
    d. IsDuplicate: Column describing whether the pair is a duplicate or not.
2. test_set.csv (121287 items): This file will be used to make predictions for new data. this file contains all fields of the training file except from the IsDuplicate field. You will be asked to predict this field using classification algorithms.

The dataset is openly available at the address:
https://www.kaggle.com/competitions/bigdata2023duplicatedetection/data.

Evaluation Results

You should evaluate the technique using 5-fold Cross Validation and you should report the following metrics: Accuracy, Precision, Recall and F1. **Your report should include a table as the following:**

Method	**Precision**	**Recall**	**F-Measure**	**Accuracy**
Method-1	0,9	0,9	0,9	0,9
Method-2	0,8	0,8	0,8	0,8

A description of the above results should be included in the report.

Output Files

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

Id	Predicted
1	1
2	0
...	

For the file "duplicate_predictions.csv" the above formatting should be used *strictly* separating the two fields with the comma (",") character and should also have the first line with the two field names (Id and Predicted) followed by your model predictions in the following lines specifying the article Id from the test set and the predicted label to indicate whether the documents in the pair with the specified Id are similar or not.
You will need to upload your file to the Kaggle contest at the address http://www.kaggle.com/competitions/bigdata2023duplicatedetection.

Hint:

1. In this question it is important to experiment with different pre-processing techniques for the questions and with different heuristic features.
2. The question pairs were annotated by Quora, and because this task is to some extent subjective, there may be errors in both the train-set and the test-set.
3. For Precision / Recall / F-Measure you will not use the number of examples in each class (Macro-Precision, Macro-Recall, Macro-F-Measure).

Regarding the deliverables

The folder you deliver should have the name:
Ass1_name1_AM1_name2_AM2.

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.