CSC 177-01 Data Warehousing and Data Mining (Spring 2019)

Mini-Project 3: Classification

Due: 2:00 pm Friday March 29, 2019

Demo Session: class time, Friday March 29, 2019

1. Classification of Twitter Users

In this project you will use the **original data** you used in Project 1. You will practice with algorithms for classification. **The goal of this project is to create classification models that predict if a user is a follower of Trump or Clinton**. In the file "clinton_trump_user_classes.txt", we have the ground truth "class" membership for each user id in the data. Class 0 corresponds to Trump followers, while class 1 corresponds to Clinton followers.

Task 1.1 (10 pts): Remove all retweets first. Remove all users that have less than 20 tweets. You may want to keep the entire tweet content, including hashtags/handles. For the remaining users, use **all available information in the dataset that you consider useful to create features** for classification (such as ***Location, Description, Place***). You are also encouraged to use any conclusions you draw in Project 2 (clustering) to create any features to improve the classification result. **Use *train_test_split()* to split data into training and test sets, where 20 percent of the records go to test set.**

Task 1.2 (20 pts): Train Decision Tree, SVM, Logistic Regression, and Neural Networks. In your report describe the features that you used for each classifier.

Task 1.3 (20 pts): Train k-NN model. In your report describe the features that you used for k-NN. **Perform parameter tuning on k-NN model. Apply 5-fold cross validation** and use **grid search** to find the best K value for k-NN model. Set scoring metric to *F1 score (F-measure)*. Use the best K value identified from grid search to train your k-NN model. **Plot the F1 score against K value** based on the results you achieved from grid search.

Task 1.4 (20 pts): Using the test set, compute the confusion matrix, the precision, recall and F-measure for (1) Decision Tree, (2) SVM, (3) Logistic Regression, (4) Neural Networks, and (5) k-NN. For k-NN model, use the best K value identified from grid search. **Compare their performance and include your conclusions in your report**

2. Grading Breakdown

Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

Implementation	**70 pts**
Your report	**20 pts**
In-class demo	**10 pts**

3. Deliverables:

(1) All your source code/figure in Python Jupyter notebook.

(2) Your report in PDF format, with the name and id of each team member, course title, assignment id, and due date on the first page. As for length, I would expect a **report with more than one page**. Your report should include the following sections (but not limited to):

1. **Problem Statement**
2. **Methodology**
3. **Experimental Results and Analysis**
4. **Task Division and Project Reflection**

In the section "**Task Division and Project Reflection**", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

10 pts will be deducted for missing the section of task division and project reflection.

Notice: please don't send me the original dataset. ☺

All the files must be submitted **by team leader** on Canvas before

2:00 pm Friday March 29, 2019

NO late submissions will be accepted.

4. Demo Session:

Each team member must demo your work during the scheduled demo session. Each team have **three minutes** to demo your work in class. The following is how you should allocate your time:

- Findings/results (1 minute)
- Task division (1 minute)
- Challenges encountered and what you have learned from the project (1 minutes)

Failure to show up in demo session will result in **zero** point for the project.

5. Teaming:

Students must work in teams of 2 or 3 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions. So you should choose partners carefully!

Hint:

- If you experience low memory issues when using *tfidfVectorzier()*, adjust the parameters ***max_df, min_df,*** and ***max_features*** appropriately. ***max_features*** should not be too small, i.e., at least 1000 or 2000.

- Pandas supports high performance SQL join operations. To create feature matrix (X) and response vector (y), you may want to use function ***pd.merge()*** to merge (or to say, join) two dataframes based on values in one particular column. See an example on *how to merge two dataframes along the subject_id value* here:

  https://chrisalbon.com/python/data_wrangling/pandas_join_merge_dataframe/