CSC 177-01 Data Warehousing and Data Mining (Spring 2018)

Mini-Project 2: Classification

Due: 10:00 am Monday April 2, 2018

Demo Session: class time, Monday April 2, 2018

In this project, you will practice with algorithms for classification.

You will use the file "clinton_trump_tweets.txt" that you used for project 1. The file contains the tweets on Twitter collected during 2016 US presidential election. The file contains tab-separated entries with 14 columns that correspond to the following fields:

Name, ScreenName, UserID, FollowersCount, FriendsCount, Location, Description, CreatedAt, StatusID, Language, Place, RetweetCount, FavoriteCount, Text

The goal is to create a classification model that predicts if a user is a follower of Trump or Clinton.

(20 pts) Remove all retweets first. Remove all users that have less than 20 tweets. For the remaining users, use **all available information in the tweets file that you consider useful** to extract features for classification. You are also encouraged to use any conclusions you draw in project 1 (clustering) to create any features to improve the classification result.

(30 pts) Use ***train_test_split()*** to split data into training and test sets, where 20 percent of the records go to test set. Train three classifiers that we saw in class (Decision Tree, SVM, Logistic Regression). **In your report describe the features that you used for each classifier.**

(20 pts) Perform parameter tuning on k-NN model. Apply **5-fold cross validation on training set** 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 to train your k-NN model. **In your report describe the features that you used for k-NN.**

(10 pts) Plot the F1 score against K value based on the results you achieved from grid search for parameter turning on k-NN.

(20 pts) Using the test set, compute the confusion matrix, the precision, recall and F-measure for (1) Decision Tree, (2) k-NN, (3) SVM, and (4) Logistic Regression. For k-NN, use the best K value

identified from grid search. **Compare their performance and include your conclusions in your report**

Hint:

- 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/

- To reduce the size of intermedia results, you may raise the cleaning threshold from 20 to 30 or 40. Also, if you use ***countVectorizer*** or ***tfidfVectorzier***, set parameters ***max_df, min_df,*** and ***max_features*** appropriately.

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!

Deliverables:

(1) **All your source code** in terms of Python Jupyter notebook or Python source files.
(2) **Your report in PDF format.** In the report, include a section "**Task Division and Project Reflection**", where you should describe the following:
   - Name and Id of each member in your team,
   - who is responsible for which part,
   - and what you have learned from the project as a team.

10 pts will be deducted for missing that section.

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

10:00 am Monday April 2, 2018

NO late submissions will be accepted.

Demo Session:

Each team member must demo your work during the scheduled demo session. Failure to show up in demo session will result in **zero** point for the project.