CS 584- Theory and Application of Data Mining Spring 2019
HW 2 – Drug Activity Prediction

Name: Akshaya Damodaran (G01129364) and Parnavi Tamhankar (G01164687) Mar 11, 2019
Registered Name on Miner: trytry77
Rank: 3
F1 Score 0.80

Problem

To develop a predictive model that can determine whether a given drug is active (1) or not (0) by using feature selection/reduction techniques and experimenting with different classification models.

Initial Steps

Firstly, all drug data are fetched from the training dataset path, their labels are extracted (Active (1) or Inactive (0)) into a list, and the drug data is collected in a DataFrame, with each row of the DataFrame representing the data for one drug as a string.

Feature Extraction using CountVectorizer and reduction using TruncatedSVD

- For this problem, feature extraction is done by using the CountVectorizer. For this problem, instead of treating the many numbers that appear in the dataset for each drug as integers, we treat them as words/strings. We can do so because, the numbers are not categorized into any kind columns.
  Applying CountVectorizer on the training and test data converts each collection into a matrix of token counts, producing a sparse representation of the counts. We do not specify any vocabulary and so, the vocabulary is determined from the collection of drug data.
- Following vectorization using CountVectorizer, we use apply TruncatedSVD on the vectorized training and test data. This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). Contrary to PCA, this estimator does not center the data before computing the singular value decomposition. This means it can work with sparse matrices efficiently which is why we use this method of dimensionality reduction on the sparse term count matrices returned by the CountVectorizer.

Program Flow Outline

- The training dataset (data for 800 drugs) is first loaded and put into a dataframe.
- A list of 'labels' is created for each of the drugs in the training dataset (Active or Inactive).
- Feature Extraction and dimensionality reduction using CountVectorizer and TruncatedSVD respectively is performed on the train data.
- Since the train data set is skewed (i.e. 722 samples of Inactive drugs and 72 samples of Active drugs), we apply SMOTE over sampler on the reduced train data which samples using Synthetic Minority Oversampling Technique i.e. synthetically creating minority samples using svm technique.
- Then the test data set is also put into a dataframe and feature extraction and dimensionality reduction are performed as explained above on the test data as well.

- We then use the Decision Tree Classifier with different weights given to class 0 (Inactive) and class 1 (Active). A label 1 is given more weight (1.5) as this class is lesser in comparison to class 0.
- Next, cross validation is performed with number of folds set to 10 and F1-score is computed to measure accuracy.
- Then, the class labels are predicted for the test data.
- Finally, the predictions list is written to a csv file called 'format.csv'.
- F1-Score on Training data:

```
Report on Decision Tree
              precision    recall  f1-score   support

          -1       0.95      0.93      0.94       722
           1       0.88      0.91      0.90       432

   micro avg       0.92      0.92      0.92      1154
   macro avg       0.92      0.92      0.92      1154
weighted avg       0.92      0.92      0.92      1154


Cross validation scores:
0.9246167027417026
```

Methodology of Choosing Approach

- The train data had around 100,000 features and the distributions were highly sparse. Therefore, we used two methods of dimensionality reduction – SparsePCA and TruncatedSVD. We saw that SparsePCA took a lot of time and accuracy was poorer in comparison with TruncatedSVD.
- The train data was also imbalanced in nature with class 1 (Active) not well represented and training with a very small number of samples of the minority class would result in weak classification. Hence, we used the SMOTE over sampler which performs oversampling by creating synthetic samples of the minority class 0.
- For the classifiers, we tried SVM, KNN, Random Forest and Decision Tree.

Report on different methodologies used –

No.	Method	Accuracy on Test Data
1.	CountVectorizer, SparsePCA, SMOTE, DecisionTree	0.48
2.	**CountVectorizer, TruncatedSVD,** SMOTE, **DecisionTree**	**0.80**
	RESULTS ON IMBALANCED DATA	
3.	CountVectorizer , PCA, KNN	0.45
4.	CountVectorizer, Sparse PCA, KNN	0.68
5.	CountVectorizer, PCA, SVM	0: 0.82 1: 0.26
6.	CountVectorizer, PCA, Random Forest	0.65