HW2 Part 1: Drug activity Prediction

Hsin-Ping Hsu (G01167652)

*Hung-Mao Chen (G01145163)*

* Name registered on miner website: zxcv
* Rank and f1 score: 11/ 0.74
* Approach:

1. Read document and revert the sparse binary matrix: read the original dataset and add the data into an 2D array. Revert both of the training and testing sparse binary matrices; the original dataset contains 100,000 attributes; thus, the size of the training dense matrix is 800 \* 100,000.

Function: read\_data(), get\_max\_dimension(), sparse\_to\_dense (rows,data)

1. Perform PCA for dimension reduction: To reduce the dimension of matrix, use the sklearn.decomposition.PCA by setting the parameter of “tol” (tolerance for singular values) as 0.85. The size of the matrix becomes 800 \* 625 after dimension reduction.

Function: training\_sampling(traindf, testdf,train\_class), pca\_reduction(traindf,testdf)

1. Over sampling to balance the training data by Synthetic Minority Over-sampling (SMOTE): use the [imblearn.over\_sampling.SMOTE](https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html) to perform SMOTE

Function: training\_sampling(traindf, testdf,train\_class)

1. Build several classification models and use cross validation to find the best parameters for the data:

* SVM Classifier: use the [sklearn.svm.svc](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) library to build the SVM classifier. Set the kernel as linear and use cross validation to calculate the f1 score to evaluate SVM classifier.

Function: SVM\_classifier(train\_resampled, class\_resampled,test\_red,test\_class)

* Decision tree classifier: use sklearn.tree.DecisionTreeClassifier to build the decision tree classifier. Also, apply the bagging algorithm to improve the accuracy.

Function: decision\_tree\_classifier(train\_resampled,class\_resampled,test\_red,test\_class)

* Naïve Bayes Classifier: build the Guassian naïve bayes classifier with sklearn.niave\_bayes.GuassianNB.

Function: naive\_bayes\_classifier(train\_resampled,class\_resampled,test\_red,test\_class)

* Adaboost Classifier: use sklearn.ensemble.AdaBoostClassifier to build the model, and apply the RandomSearchCV to find the best parameters for model.

Function: adaboost\_classifier(train\_resampled,class\_resampled,test\_red,test\_class)

* Random Forest classifier: use [sklearn.ensemble.RandomForestClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) library to build the model. Tuning the parameters by performing [RandomizedSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html), use the best parameters to build the random forest classifier. Parameters: criterion, min\_samples\_split, max\_depth, min\_samples\_leaf, max\_leaf\_nodes, n\_estimators.

Function: random\_forest\_classifier(train\_resampled,class\_resampled,test\_red,test\_class)

* Apply cross validation to evaluate the models. The result shows that random forest classifier might be a good model for this dataset.

1. Classifier Evaluation: The result of cross validation shows that Random Forest classifier is one of the best classifiers for this dataset. Therefore, we decided to use random forest classifier for classification.

Function: K\_fold\_validation(traindf, classdf)

Fig. 1

1. Compare the Random Forest Classifier that are built with original imbalanced data and the classifier that are built with balanced data after over sampling.

|  |  |  |
| --- | --- | --- |
| Classifier | Class 0\_f1 score | Class 1\_f1 score |
| imbalanced | 0.951559 | 0. 02711111 |
| balanced | 0. 96457164 | 0.65757912 |

Fig. 2

|  |  |
| --- | --- |
| 一張含有 地圖, 文字 的圖片  自動產生的描述 | 一張含有 地圖, 文字 的圖片  自動產生的描述 |
| Fig.3 Imbalanced ROC | Fig. 4 balanced ROC |

**References:**

1. <https://developerzen.com/data-mining-handling-missing-values-the-database-bd2241882e72>
2. <https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60>
3. <https://towardsdatascience.com/an-approach-to-choosing-the-number-of-components-in-a-principal-component-analysis-pca-3b9f3d6e73fe>
4. [https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over\_sampling.SMOTE.html](%5b4%5d.https:/imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html)
5. [https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html](%5b1%5d.https:/imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html)
6. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.fit>
7. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html#sklearn.ensemble.BaggingClassifier.predict>
8. <http://clopinet.com/isabelle/Projects/NIPS2003/#challenge>
9. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
10. <https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/>
11. <https://datascience.stackexchange.com/questions/15630/train-test-split-after-perform-smote>