Assignment 4 Report

MING JIANG SIHAN CHEN GENGCONG YAN

mingj | sihanc | gengcong@kth.se

December 10, 2020

Contents

1 Introduction

This project is based on the activity prediction of chemical compounds, which are displayed as SMILEs string. We first check the basic information and distribution of the raw dataset, and add some features based on RDKit library[1] and Morgan fingerprints. Next, we test the baseline of different models on different feature sets, and decide which combination to do the hyper-tuning. We choose the Random Forest model with basic atom features and two kinds of fingerprints(ECFP, FCFP) calculated by RDKit library, and lightGBM with all features to do the further comparison. Finally, our decision is to use Random Forest model, which can get the highest AUC value on the test set, and show the stability on the 5-folds cross-validation. Our estimated_auc is therefore obtained according to the performance of our model on the test data, which is **0.826**.

We run our program in Kaggle kernel, with Python Version of 3.7.6. and Conda Version of 4.9.2. The external packages we used is shown in Table 1.

Table 1: External Packages

Package name	Version	Package name	Version	Package name	Version
numpy	1.18.5	gensim	3.3.3	sklearn	0.0
pandas	1.1.4	matplotlib	3.2.1	mol2vec	0.1
scipy	1.4.1	rdkit	2020.09.2		

2 Data Preparation

The raw dataset consists of 121,374 rows of data, with the ”SMILES” column representing the structure of chemical species using short ASCII strings, and the *label* column named ”ACTIVE”. We loaded the data with *Pandas* to check the basic distribution of the raw dataset, finding out that it’s an imbalanced dataset, in which only around **10%** data are labeled ”ACTIVE”. The distribution is shown as Fig.1.

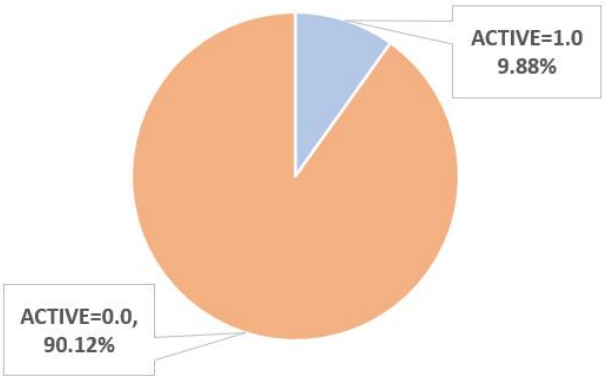

Figure 1: Data Labels Distribution

The imbalanced data need to be noticed during the parameter tuning of model. In addition, it is why we need to use AUC value as the final evaluation index, instead of accuracy.

In order to better evaluate the performance of our model and avoid overfitting, we use kFolds cross-validation, with $k = 5$. Fig. 2 shows the data separation on the original dataset. We firstly split the data into training set (80% of the training data) and test set (20% of the training data), respectively for training the model and testing model performances. And the 80% training set will be further separated into training set and validation set for the cross-validation.

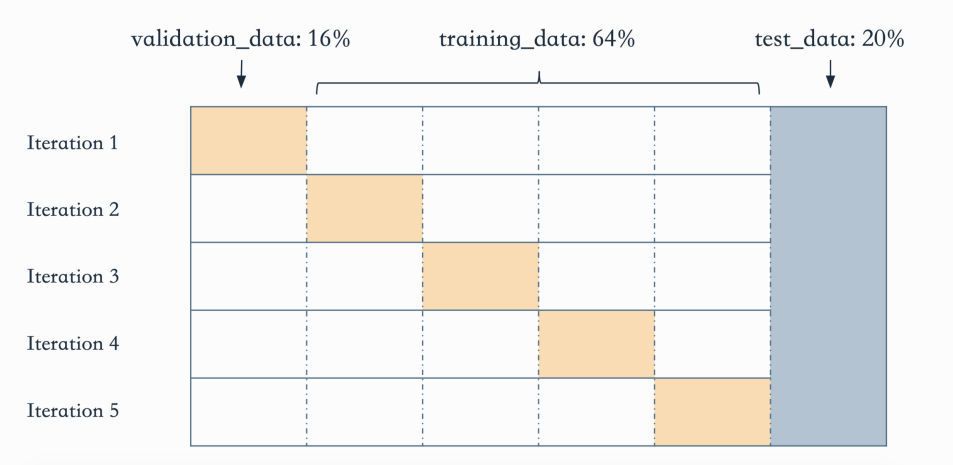

Figure 2: k-folds(k=5) Cross-validation on Dataset

3 Feature Selection

There were some existing researches exploring the prediction of chemical compounds activity, which have provided us several ways to start our work. Ratnawati et. al.[2] used some basic features, like the number of atoms, the Molar mass, the number of rings, number of certain elements (such as Cl, Br), to predict a chemical compounds to be active or not. And the Morgan fingerprints, which learned the features inside a chemical compound based on the structure, could well include the features of atoms, substructures, functional groups, *etc.* and generate a unique identifier. There are three kinds of fingerprints which might be used in the following procedure[3][4].

In this project, we extract and consider four parts of features, the 10 bit basic features (molecular weight, number of heavy atoms and so on), the 124 bit FCFP fingerprint, 124 bit ECFP fingerprint and the 300 bit Mol2Vec vector features, which will be all described in detail in Section 3.1 and 3.2. Different features combinations are chosen for different models, which will be shown in Section 4.

3.1 Basic Chemical Compound Features

Since we all lack professional chemical knowledge on this dataset, Some simple and basic features of the chemical compound are our first choice. These features can provide a quick and easy way to express the fundamentals of a chemical compound. The features we used are below:

- GetNumAtoms( ) - **Number of atom**
- CalcExactMolWt( ) - **Molecular weight**

- fr_ArN( ) - **Number of N functional groups attached to aromatics**
- fr_halogen( ) - **Number of halogens**
- Lipinski.NumAliphaticRings( ) - **Number of aliphatic rings for a molecule**
- Lipinski.NumAromaticRings( ) - **Number of aromatic rings for a molecule**

Features like number of aliphatic carboxylic acids and aliphatic hydroxyl groups are also tested on the dataset. But we decide to drop them after finding that their means are zero.

Some of these basic features, such as *Number of atom*, *Molecular weight*, and so on, are too large compared with the others. As a result, we run imputation, normalization and discretization on training and validation set for these basic atom features.

3.2 Three Kinds of Fingerprints

1. **Extended Connectivity Fingerprints**[3]. An ECFP is defined as the set of all atom identifiers for each radius of perception up to the limit n. As the radius of perception expands, this set includes all identifiers found in both previous iterations and the current one. This family of fingerprints, better known as Morgan fingerprints, is built by applying the Morgan algorithm to a set of user-supplied atom invariants. When generating Morgan fingerprints, the radius of the fingerprint must also be provided. For ECFP-class fingerprints, the atom properties are atomic number, charge, hydrogen count, *etc.*[5]

2. **Functional-Class Fingerprints**. Rather than basing the original atom identifier on CANGEN[6] properties, Feature invariants of FCFP are all based on the presence of a "pharmacophoric" property, which are hydrogen bond acceptor, hydrogen bond donor, negatively ionizable, positively ionizable, aromatic and halogen. At some times this can lead to quite different similarity scores in ECFP and FEFP.

3. **Mol2vec**. Mol2vec is an unsupervised machine learning approach to learn vector representations of molecular substructures[4]. It is a new idea borrowing from natural language processing techniques Like the Word2vec models. Words of close relationship in documents are also in close proximity in the vector space, when they are represented in a form of vector. In the similar way, Mol2vec can learns vector representations of molecular substructures that point in similar directions for chemically related substructures. Compounds, like words, can finally be encoded as vectors by summing the vectors of the individual substructures. After that, these vectors can be fed into machine learning models to train and predict compound properties.

Overall, here we get four kinds of features, containing *basic atom features*, and three kinds of fingerprints as *ECFP*, *FCFP*, and *mol2vec*. Next, after the imputation, normalization and discretization on training and validation set for *basic atom features*, we combine them relatively and run several models to get the best combination of feature set and model.

3.3 Feature importance

In addition, we compare the importance of different features on predictions. One approach is to simply get the correlation matrix of the training data to see which feature has the highest correlation coefficient in the ACTIVE column and the found feature is the most important when distinguishing between active and inactive cases. Another approach is to use the functions in InfoBitRanker of rdInfoTheory module to calculate the information gain of the features. The information gain is able to show how significantly a feature can be used to transform a system into a stable state, which means the feature is more useful for classifying on data. We try to keep the relatively important features found by the above two approaches and drop the other less important ones. However, after repeated experiment, it shows that the remained features yield a worse AUC score and it is not an ideal idea to consider dropping the features that are superficially less important. Therefore, we decide to keep the original feature set.

4 Building Models

4.1 Environment and External Packages

The code environment in the group is Kaggle notebook. The external packages we chose are displayed in Table 1. Specifically, RDKit[1] is a collection of cheminformatics and machine-learning software contains several subpackages that may be used to generate features. Through RDKit, we can know the atoms numbers, molecular weight or fingerprints, *etc.* of the chemical compounds. Mol2vec[4] is an unsupervised machine learning approach to learn vector representations of molecular substructures. Here, we use RDKit to get some basic features through the chemical compounds, as well as two kinds of Morgan Fingerprints, while mol2vec provides us another way to get the fingerprint of chemical compounds.

4.2 Baseline

For each of the meaningful feature combinations, we develop different models and compare their performances on training data with the feature combination. We firstly select some simple machine learning models such as Naive Bayes, Decision Tree model and Random Forest model and filter out some models that have relatively bad performances. In the final, we put emphasis on three models, Random Forest, LightGBM and XGBM because they show better AUC results. The baseline of these model in different feature set without tuning is shown as Table 2.

As shown in Table 2, the random forest model with the feature combination of Basic+FCFP+ECFP is the best. Besides, the LightGBM model with the feature combination of Basic+FCFP+ECFP+Mol2Vec overwhelms other combinations of the same model. Therefore, these two models with the corresponding feature combinations are chosen as the two primary models of this project.

Table 2: Baseline: Comparison of Different Models with Different Features

	Random Forest	LightGBM	XGBM
Basic	0.590	0.641	0.627
FCFP	0.804	0.741	0.682
ECFP	0.787	0.740	0.687
Mol2Vec	0.757	0.764	0.726
Basic + FCFP	0.801	0.753	0.694
Basic + ECFP	0.791	0.744	0.696
Basic + Mol2Vec	0.759	0.779	0.729
Basic + FCFP + ECFP	**0.811**	0.772	0.716
Basic + FCFP + ECFP + Mol2Vec	0.767	**0.783**	0.737

5 Hyper-Parameter Optimization

The next step is to adjust the hyper-parameters to optimize the two models. The approach we use is the GridSearchCV function in the *sklearn* package. GridSearchCV is used to apply an exhaustive search over specified parameter values for an estimator.

We emphasize two hyper-parameters, the *max_depth* and *n_estimators* when tuning the Random Forest model. In the experiment, the max_depth of 30 and the estimators number 400 have the highest average AUC score of 0.821 on 5 folds estimation, as shown in Table 3. We choose it as our final parameter combination, because it has a high AUC value on test data and performs stably on the 5-folds cross-validation. Fig. 3 shows the AUC value of Random Forest on the 5-folds cross-validation with the best parameters, together with the *std_test_score* in Table 3, these lines almost overlap, and the variance of the cross-validation results is very small, explaining the stability and no no overfitting of the model to a certain extent.

Table 3: AUC score of random forest model on 5 splits with different hyper-parameters

max depth	**n_estimators**	**split0**	**split1**	**split2**	**split3**	**split4**	**mean score**	**std test score**
30	225	0.8186	0.8071	0.8139	0.8136	0.8182	0.8143	0.004156809
30	300	0.8218	0.8137	0.8160	0.8180	0.8229	0.8185	0.003461563
30	**400**	0.8259	0.8147	0.8171	0.8211	0.8216	**0.8201**	**0.003854793**
40	225	0.8196	0.8098	0.8150	0.8164	0.8190	0.8160	0.003524616
40	300	0.8255	0.8100	0.8159	0.8176	0.8194	0.8177	0.00502906
40	400	0.8269	0.8100	0.8172	0.8192	0.8238	0.8194	0.005813082

We also test the performance on lightGBM with the feature set: Basic, FCFP, ECFP, mol2vec, which contains all the features we have considered. And the best performance on this combination, with the best hyper-parameters combination: learning rate of 0.2, number of leaves of 60, number of estimators (250), subsample (0.8), reg_alpha 0.1 and reg_lambda (0) is 0.805, still lower than that of Random Forest model, as shown in Figure 4. As a result, we choose Random Forest, with the feature set: Basic, FCFP, ECFP, as our final model.

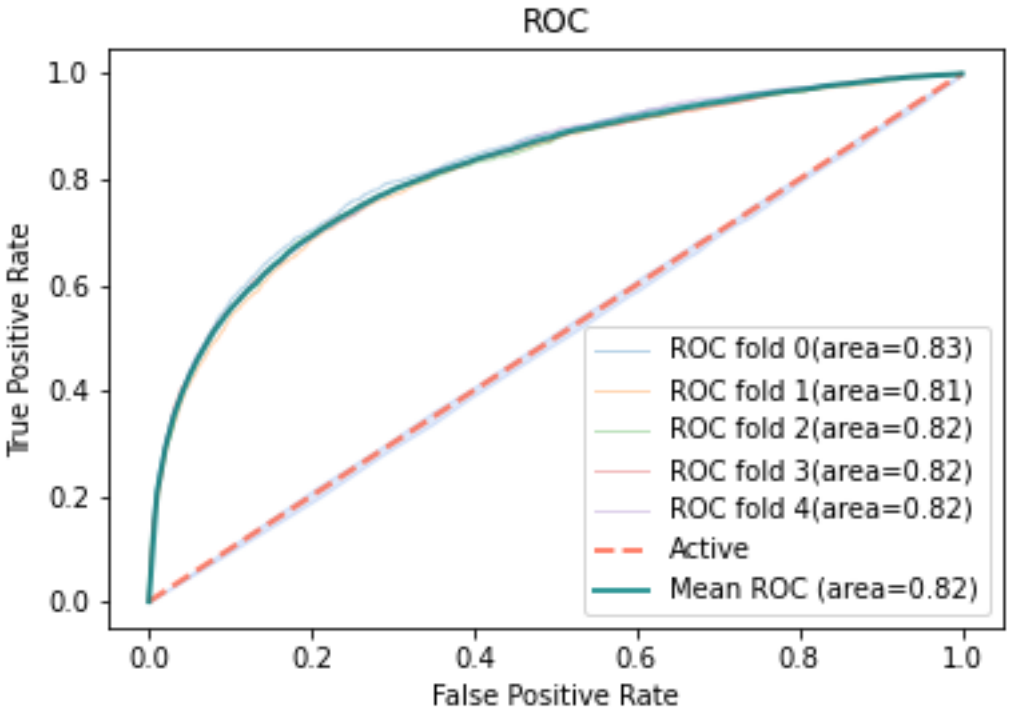

Figure 3: 5-folds Cross-validation on Random Forest

6 Results

In the final, we adopt the Random Forest classifier with 400 estimators and a max depth of 40 as our model. Also we set the hyper-parameter 'class_weight' to be 'balanced_subsample', which may help when the training data is unbalanced. The final estimated AUC on the 20% training data is 0.831. It is a little higher than the AUC scores of 5-fold estimation. The reason may be that in the 5-fold estimation we have 5 splits on the training set which accounts for 80% of the whole training data, which means that there are less training data (64% of the whole training data) for 5-fold estimation so the model may earn a lower AUC score.

7 Future Work

1. The features we chose only trained on single model, we can consider to combine multiple models in prediction at the same time.

2. We can consider the possible prediction based on the data structure of chemical compounds.

3. We can generate the images of chemical compounds' structure base on "SMILES". Taking image resolution and computing power into consideration, we can try to solve the problem in image prediction models .

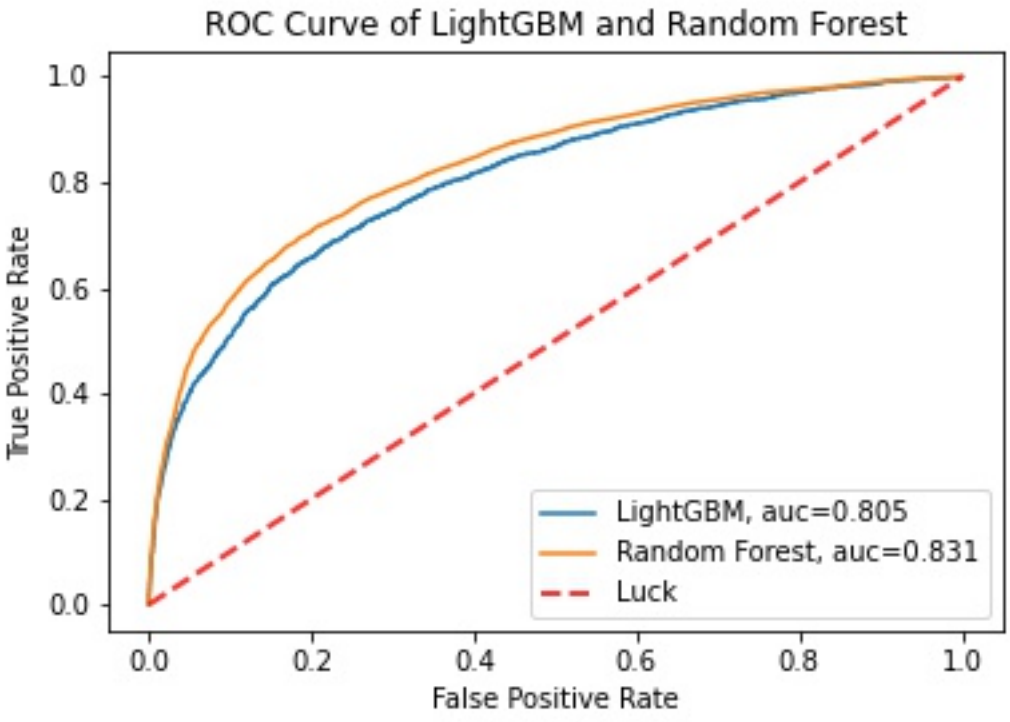

Figure 4: The ROC curve of LightGBM vs Random Forest

References

[1] "Rdkit," https://www.rdkit.org/, accessed December 9, 2020.

[2] D. E. Ratnawati, Marjono, and S. Anam, "Prediction of active compounds from smiles codes using backpropagation algorithm," in *AIP Conference Proceedings*, vol. 2021, no. 1. AIP Publishing LLC, 2018, p. 060009.

[3] "Computing extended connectivity fingerprints," https://depth-first.com/articles/2019/01/11/extended-connectivity-fingerprints/, accessed December 9, 2020.

[4] S. Jaeger, S. Fulle, and S. Turk, "Mol2vec: unsupervised machine learning approach with chemical intuition," *Journal of chemical information and modeling*, vol. 58, no. 1, pp. 27–35, 2018.

[5] "Open source ecfp/fcfp circular fingerprints in cdk," https://cheminf20.org/2014/02/21/open-source-ecfpfcfp-circular-fingerprints-in-cdk/, accessed December 9, 2020.

[6] D. Weininger, A. Weininger, and J. L. Weininger, "Smiles. 2. algorithm for generation of unique smiles notation," *Journal of chemical information and computer sciences*, vol. 29, no. 2, pp. 97–101, 1989.