

Manual

MRMD3.0: a python tool and webserver for dimensionality reduction
and data visualization through ensemble strategy

Authors: Shida He^{1,2}, Xiucai Ye², Sakurai Tetsuya², Quan Zou^{1*}

Institutions

¹ Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, Zhejiang, China;

² Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan

*corresponding author

Quan Zou: zouquan@nclab.net

GitHub: <https://github.com/heshida01/MRMD3.0>

Webserver: <http://lab.malab.cn/soft/MRMDv3/home.html>

Content

1. Installation

2. Feature rank and Ensemble strategy

3. Input Format of MRMD3.0

4. Command Version

5. Web Server

6. Summary

Reference

1. Installation

MRMD3.0 is an open-source Python-based toolkit, which operates depending on the Python environment (Python Version 3.0 or above). It can be run on multi-OS systems (such as Windows, Mac and Linux operating systems). Before running MRMD3.0, the user should make sure all the following packages are installed:

```
1 category_encoders==2.5.0
2 joblib==1.1.0
3 matplotlib==3.5.2
4 minepy==1.2.6
5 networkx==2.8.4
6 numpy==1.23.1
7 pandas==1.4.3
8 Pillow==9.2.0
9 plotly==5.9.0
10 pydicom==2.3.0
11 scikit_learn==1.1.1
12 scipy==1.8.1
13 seaborn==0.11.2
14 tqdm==4.64.0
```

A faster way to install them is you can install them in MRMD3.0 folder via command:

```
~/MRMD3.0 (master*) » pip install -r requirements.txt
```

2. feature rank and Ensemble methods overview

MRMD3.0 has many built-in feature rank methods, divided into three types: filtering, wrapper and embedded methods. The methods used for ranking feature and combine include PageRank, LeaderRank, HITS and LeaderRank.

2.1 feature rank

In the previous version of MRMD 2.0, we only used seven feature sorting algorithms. We have added 20 methods in the latest MRMD3.0. These methods are mainly from sickit-learn, minepy or our implementation. MRMD3.0 has also optimized the calculation speed, and we have strengthened the parallelism of the code. At the same time, in the experiment, we found that the original mRMR algorithm was too computationally complex. In MRMD3.0, we used a fast mRMR calculation method (<https://github.com/smazzanti/mrmr>), which is mainly explained here.

Table 1. feature rank and ensemble me

type	method	No. variant	Ensemble strategy
Filter	ANOVA	1. Analysis of variance (ANOVA)	PageRank
	Chis Square	2. Chis Square	
	Mutual information	3. Mutual Information (MI) 4. Normalized Mutual Information(NMI) 5. Maximal Information Coefficient (MIC)	
	MRMD1.0	6. MRMD- Euclidean distance 7. MRMD-Cos distance 8. MRMD-Tan Coefficient	LeaderRank
	mRMR	9. F-test correlation difference (FCD) 10. F-test correlation quotient (FCQ)	HITS
Warpper	Recursive feature elimination	11. LogisticRegression, 12. SVM 13. DecisionTreeClassifier	TrustRank
Embedded	Tree feature importanc	14. DecisionTreeClassifier, 15. RandomForestClassifier, 16. GradientBoostingClassifier 17. ExtraTreesClassifier	
	Linear model	18. Lasso 19. Ridge 20. Elasticnet	

2.2 Ensemble strategy

PageRank is an excellent ranking algorithm which is widely used in the ranking task of search engines. MRMD3.0 applies it to the "voting" task, combining the results of each feature ranking algorithm to calculate the feature importance. Besides, MRMD3.0 also has three classic algorithms(HITS, LeaderRank and TrustRank).

PageRank

There are many link relationships between web pages on the Internet. If you can fully use these structures, it can significantly improve the quality of web search. PageRank assumes that multiple web pages will point to a good web page and then calculates the PageRank value by the number of links and degrees of nodes.

HITS

HITS is an algorithm based on a keyword query. The algorithm requires query operations to obtain some web pages as the root set and then expand the web pages that have direct links to the web pages in the root set to the base set.

LeaderRank

The LeaderRank algorithm adds a global node to the original network graph, thereby improving the entire network's connectivity and ensuring the algorithm's convergence speed and robustness. We can also learn from this idea and add a global feature node connected to all features in the directed graph of features.

TrustRank

TrustRank requires human involvement to filter out high-quality torrent pages, but this does not apply and feature ranking because it is difficult for human methods to empirically determine whether a feature is important for a classification task. Therefore, the implementation method of this work is first to select the first few pages provided by PageRank as the features with high importance as seed features and then use this algorithm to rank the features.

3 Input Format of MRMD3.0

MRMD3.0 requires the input of a **2d matrix** with labels, supports multi-label datasets, and belongs to supervised learning. The file format can be CSV, libSVM or Arff.

Tip: label must be placed in the first column in CSV and LibSVM.
The numeric type must be an integer.

Example:

CSV

class	feature1	feature2	feature3	feature4
1	0.130742	0.156566	0.134454	0.188119
1	0.125861	0.1456478	0.745411	0.145791
0	0.145269	0.521744	0.257617	0.346172
0	0.369453	0.216412	0.152212	0.126243

libsvm

1	1:0.130742	2:0.156566	3:0.134454	4:0.188119
1	1:0.125861	2:0.145648	3:0.745411	4:0.145791
0	1:0.145269	2:0.521744	3:0.257617	4:0.346172
0	1:0.369453	2:0.216412	3:0.152212	4:0.126243

Arff

@relation NAME					
@attribute feature1 numeric					
@attribute feature2 numeric	0.130742	0.156566	0.134454	0.188119	1
@attribute feature3 numeric	0.125861	0.1456478	0.745411	0.145791	1
@attribute feature4 numeric	0.145269	0.521744	0.257617	0.346172	0
@attribute class { 0,1 }	0.369453	0.216412	0.152212	0.126243	0
@data					

4 Command Version

MRMD3.0 provides many parameters for users to choose. You can usually get good results with the default parameters by giving the input and output files. Below we will introduce the critical parameters with specific examples.

<https://github.com/heshida01/MRMD3.0>

```
~/MRMD3.0 (master*) > python mrmd3.0.py -h
usage: mrmd3.0.py [-h] [-s S] [-i I] [-e E] [-l L] [-n N] [-t T] [-c {RandomForest,SVM,Bayes}] [-o O] [-p P] [-f F] [-g G] [-r {PageRank,Hits_a,Hits_h,LeaderRank,TrustRank}]

optional arguments:
  -h, --help            show this help message and exit
  -s S, --start S        start index
  -i I, --inputfile I    input file
  -e E, --end E          end index
  -l L, --length L       step length
  -n N, --n_dim N        mrmd3.0 features top n
  -t T, --type_metric T  evaluation metric(f1, acc, recall,precision, auc)
  -c {RandomForest,SVM,Bayes}, --classifier {RandomForest,SVM,Bayes}
                        classifier(RandomForest,SVM,Bayes)
  -o O, --outfile O      output the dimensionality reduction file
  -p P, --picture P      The scatter plots before and after dimension reduction are generated by tsne
  -f F, --topn F         select top n features to chart
  -g G, --config G       the config file for select feature rank methods
  -r {PageRank,Hits_a,Hits_h,LeaderRank,TrustRank}, --rank_method {PageRank,Hits_a,Hits_h,LeaderRank,TrustRank}
                        the rank method for features
```

Basic:

```
~/MRMD3.0 (master*) > python mrmd3.0.py -i test.csv -o out.csv
```

Customize:

1) Select an interval of data for dimensionality reduction

Parameters: -s start index ; -e end index;

Note: Under this parameter, MRMD3.0 will directly select and rank the top N features without performing feature search on the dataset. Usually, start_index(3) and end_index(8) are used together to reduce the dimensionality of a certain range(in this demo 8 - 3) of data features. By default, MRMD3.0 performs dimensionality reduction on all features (start_index = 1 and end_index = -1)

```
~/MRMD3.0 (master*) > python mrmd3.0.py -i test.csv -s 3 -e 8 -o out.csv
```

class	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature13	feature14	feature15	feature16
0	0.1	0.025	0.075	0.125	0.075	0	0.1	0	0.125	0.075	0.05	0.025	0.075	0.025	0.025	0.1
0	0.025	0.025	0.05	0	0	0.125	0.125	0.05	0.05	0.05	0.1	0.025	0.125	0.075	0	0.15
0	0.1	0.025	0.05	0	0	0.025	0.1	0.05	0.025	0.075	0.125	0.125	0.05	0.05	0.075	0.125
0	0.1	0.1	0.025	0.05	0.05	0.05	0.025	0.1	0.05	0	0.025	0.05	0.1	0.075	0.05	0.15
0	0.175	0	0.075	0.125	0	0	0	0.1	0.175	0	0.075	0	0.05	0.1	0.075	0.05
0	0.125	0.05	0.075	0.075	0.025	0	0	0.125	0.05	0.05	0.05	0.05	0.1	0.05	0.075	0.1
0	0.075	0.1	0.05	0.1	0.1	0.1	0.1	0	0.075	0.025	0.05	0	0.075	0.05	0.05	0.075
0	0.075	0.05	0.025	0.125	0.075	0.075	0.025	0.05	0.075	0.1	0.05	0.05	0.025	0.025	0.175	0
0	0	0.025	0.05	0.025	0.05	0.05	0	0.2	0.05	0	0	0.05	0.025	0.2	0.05	0.225
0	0.075	0	0.15	0.025	0.05	0.05	0	0.075	0.1	0.125	0.25	0	0.025	0	0.05	0.025
0	0.1	0.025	0.15	0.025	0.025	0.025	0	0.075	0.175	0	0.15	0.05	0	0.05	0.075	0.075
0	0.05	0.05	0.1	0.025	0.025	0.075	0	0.075	0.15	0.025	0.2	0.025	0	0.025	0.1	0.075
0	0.075	0.025	0.075	0.05	0	0.05	0	0.1	0.1	0.075	0.025	0.1	0.025	0.025	0.2	0.075
0	0.075	0.025	0.05	0.1	0	0	0	0.1	0.175	0	0.075	0.05	0	0.075	0.175	0.1
0	0.05	0.05	0.05	0.075	0	0	0	0.15	0.175	0.025	0	0.025	0	0.075	0.2	0.125
0	0.15	0.075	0.125	0.025	0.05	0	0	0.075	0.1	0.025	0.025	0.1	0.1	0.025	0.1	0.025
0	0	0.025	0.1	0.05	0.025	0.125	0.025	0.075	0.125	0.1	0.05	0.05	0.025	0.025	0.15	0.05
0	0.15	0.025	0.125	0.05	0.075	0.025	0	0.025	0.075	0.075	0.15	0.05	0.05	0	0.075	0.05
0	0.05	0.05	0.15	0.05	0.075	0	0.05	0.025	0.175	0.075	0.15	0.025	0	0.025	0.1	0
0	0.225	0.075	0.15	0	0.025	0	0.05	0.025	0.175	0.025	0.125	0.05	0.025	0	0.05	0
0	0.125	0	0	0.025	0.1	0.025	0.025	0	0.1	0.075	0.075	0.1	0.025	0.05	0.05	0.025

2) Select the top n features

Parameters: -n 100

Default: None

```
~/MRMD3.0 (master*) > python mrmd3.0.py -i test.csv -n 100 -o out.csv
```

Note: which is automatically calculated by default and does not need to be set, on the contrary, you can also set it to 100, it will save the top 100 features.

3) Choose Evaluation Metrics

This parameter is the evaluation metric for cross-validation when using IFS.

```
~/MRMD3.0 (master*) > python mrmd3.0.py -i test.csv -t acc -o out.csv
```

Parameters: -t f1 | acc | recall | precision | AUC

Default f1

Note: When using the IFS method to select the most suitable dimensions, the recommended metrics are as follows: for balanced data, accuracy is recommended; For slightly imbalanced data that require a balance between recall and precision, the F1 score is recommended; If your application requires minimizing false positives (incorrectly predicting positive cases), then you should focus on precision. If your application needs to minimize false negatives (failing to identify actual positive cases), then recall should be the primary focus. and for extremely imbalanced data, The AUC parameter is recommended.

4) Select an ensemble strategy for feature ranking

Parameters: -r PageRank | Hits_a | Hits_h | LeaderRank | TrustRank

Default:PageRank

```
~/MRMD3.0 (master*) > python mrmd3.0.py -i test.csv -r PageRank
```

Note:

PageRank: Calculate the weight of a node based on the number of incoming links (and the weight of the link source) of each node. The advantage is that it is simple and easy to understand.

TrustRank: Some high-quality features are added before MRMD3.0 feature sorting, and these features need to be obtained in advance by other methods. MRMD3.0 uses some features recommended by PageRank by default.

HITS: The HITS algorithm takes into account both incoming and outgoing link characteristics. Which method to choose depends. Hubs focus on pages that are linked to by authoritative pages. authorities are pages with links from the hub.

LeaderRank: It not only considers incoming links (like Pagerank), but also outgoing links and local network structure. A key feature of Leaderrank is that it uses a global "ground node" interconnected with all other nodes to solve the problems of rank leakage and rank pooling. The advantage of LeaderRank is that it can better capture the local influence of nodes in the network.

5) Choose classifiers

This parameter is the classifier for cross-validation when using IFS.

Parameters: -c RandomForest | SVM | Bayes

default: RandomForest

```
~/MRMD3.0 (master*) > python mrmd3.0.py -i test.csv -c bayes -o out.csv
```

Note: The classifier used when using IFS to select the most suitable dimension, choose according to your own needs, and recommend a random forest with stable performance.

6) customize feature selection methods to rank feature

Parameters: -g True | False

Default: False

```
~/MRMD3.0 (master) > python mrmd3.0.py -i test.csv -o out.csv -g true
```

Note: If only one method is used, such as Lasso, MRMD3.0 only uses the Lasso method to rank the features and will no longer execute ensemble algorithms such as PageRank. Path: MRMD3.0/config.py

```
version 3.0
No. variant
1. ANOVA
2. Chis_Square
3. MI
4. NMI
5. MIC
6. MRMD-Eu
7. MRMD-Cos
8. MRMD-Tan
9. FCD
10. FCQ
11. rfe_LogisticRegression,
12. rfe_SVM
13. ref_DecisionTreeClassifier
14. DecisionTreeClassifier,
15. RandomForestClassifier,
16. GradientBoostingClassifier
17. ExtreTreesClassifier
18. Lasso
19. Ridge
20. Elasticnet
====
##DEMO:
##use "ANOVA", "Chis_Square", "MIC" method.
methods = [1]
```

7) Incremental feature selection step

Parameters: -l step length

Default: 1

Note: By default, when using the IFS feature search, each feature is added one by one to search. Here you can increase 1 to make the algorithm execute faster. For example, if it is changed to 3, every 3 features will be searched in groups.

```
~/MRMD3.0 (master*) » python mrmd3.0.py -i test.csv -l 3 -o out.csv
```

Note: By default, when using the IFS feature search, each feature is added one by one to search. Here you can increase 1 to make the algorithm execute faster. For example, if it is changed to 3, every 3 features will be searched in groups.

class	feature322	feature37	feature177	feature22	feature317	feature101	feature77	feature42	feature137	feature226	feature361	feature281	feature22	feature247	features
0	0.4	0.4	0.4	0.2	0.4	0.4	0.2	0.2	0.2	0.4	0.2	0	0.4	0	0.6
0	0.6	0.2	0.2	0.4	0	0.2	0.4	0.2	0.4	0.4	0	0.2	0	0.2	0.2
0	0.6	0	0.8	0.6	0.2	0	0	0.6	0.6	0	0.2	0.2	0.6	0.2	0
0	0.2	0	0	0	0	0.4	0	0	0	0.6	0.4	0.6	0	0	0.6
0	0.2	0.2	0.4	0.6	0.2	0	0.2	0.8	0.4	0	0.4	0.2	0.2	0.4	0
0	0.4	0	0.4	0.4	0.4	0	0.2	0.6	0.6	0	0	0	0.2	0.4	0
0	0.4	0.6	0.2	0.2	0.2	0.2	0.4	0.4	0.2	0	0.4	0.6	0.4	0.2	0.2
0	0.2	0.2	0.2	0	0.2	0	0.4	0	0.2	0.4	0.2	0.2	0.2	0	0
0	0.4	0	0	0.6	0.2	0	0	0.6	0	0.4	0.2	0.4	0	0.2	0
0	0.4	0.4	0.2	0.2	0.4	0.4	0.2	0.2	0	0.4	0	0	0	0.2	0.4
0	0.6	0.2	0.2	0.6	0.2	0.6	0.4	0.4	0.4	0.2	0.2	0	0.4	0.6	0.4
0	0.4	0	0	0.2	0	0	0.2	0.2	0.2	0.2	0.4	0.6	0.2	0.2	0
0	0.2	0	0	0	0	0.8	0	0	0	0.6	0.4	0.6	0.6	0	0.8
0	0.2	0.4	0.4	0.4	0.2	0	0.6	0.2	0.4	0	0.2	0.6	0.2	0.4	0
0	0.2	0.2	0	0	0.2	0.6	0.2	0	0	0	0.6	0.4	0.4	0	0.4
0	0.4	0.2	0	0.2	0.4	0.2	0.2	0.4	0	0.2	0.2	0.2	0.4	0.2	0.4
0	0.4	0.6	0.4	0.2	0.4	0	0.4	0.4	0.4	0	0	0.4	0	0.4	0
0	0.8	0.2	0.2	0.4	0.2	0.2	0.2	0.4	0	0.4	0.4	0	0.8	0.4	0
0	0.2	0.4	0.2	0.2	0.2	0	0.4	0.2	0.4	0.6	0.4	0	0.4	0.4	0
0	0.6	0.2	0.6	0.4	0.4	0.2	0	0.2	0.6	0.4	0	0	0	0.2	0.4
0	0.2	0.4	0.2	0.4	0.4	0.2	0.4	0.6	0.2	0.6	0.4	0.2	0.4	0.2	0
0	0.2	0.2	0	0.4	0.2	0.4	0	0.4	0	0.2	0.6	0.4	0.4	0.2	0.4
0	0.2	0.4	0.4	0.2	0.2	0	0.2	0.4	0.2	0.4	0.6	0.4	0.4	0	0
0	0.8	0.4	0.2	0.4	0	0	0.4	0.4	0.4	0	0	0.2	0.4	1	0
0	0.6	0.2	0	0.6	0.6	0.4	0.4	0.4	0.2	0.2	0	0.4	0	0.4	0
0	0.4	0.4	0.4	0.2	0.4	0	0.4	0.2	0.4	0.2	0.4	0.2	0.6	0	0.2
0	0.4	0.2	0.2	0	0.2	0.4	0.2	0	0.4	0.8	0.4	0	0.4	0.2	0.6
0	0.6	0.4	0.6	0	0.2	0.2	0.4	0	0.8	0	0.2	0.2	0	0.2	0.4

8) select evaluation metric for cross-validation

Parameters: -t f1, acc, recall, precision, auc

```
~/MRMD3.0 (master*) » python mrmd3.0.py -t recall -i test.csv -o out.csv
```

$$\text{acc} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

$$\text{precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

$$\text{recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

$$f1 = \frac{2 * \text{precision} * \text{recall}}{(\text{precision} * \text{recall})}$$

AUC (<https://scikitlearn.org/stable/modules/generated/sklearn.metrics.auc.html>) is the area under the receiver operating characteristic curve from prediction scores.

Note: When using the IFS method to select the most suitable dimensions, the recommended metrics are as follows: for balanced data, accuracy is recommended; For slightly imbalanced data that require a balance between recall and precision, the F1 score is recommended; If your application requires minimizing false positives (incorrectly predicting positive cases), then you should focus on precision. If your

application needs to minimize false negatives (failing to identify actual positive cases), then recall should be the primary focus. and for extremely imbalanced data, The AUC parameter is recommended.

5 Online Webserver

The Webserver (<http://lab.malab.cn/soft/MRMDv3/home.html>) parameter usage is almost the same as the Standalone version. After opening the link, enter the corresponding parameters, upload the file, and wait for the result.

The screenshot shows the MRMD3.0 Webserver interface. The top navigation bar includes links for Home, Webserver, Data demo, Result demo, Code, and Contact. The left sidebar contains the 'Parameters' section with the following fields:

- start_index**: default = 1. The default (1) is equal to The first feature number at the beginning of the data set.
- end_index**: default = -1. The default (-1) is equal to The last feature number at the beginning of the data set.
- length**: default = 1. The default step size is 1.
- dimensions**: default = 1. Default automatic calculation.
- top_n**: default = 15. select top n features to chart.
- feature rank method**: PageRank (dropdown).
- classifier**: RandomForest (dropdown).
- evaluate metric**: f1 score (dropdown).
- Upload Dataset**: (the example file click [here](#))
- Choose File**: No file chosen.
- Start MRMD3.0** button.

The main content area includes:

- Tips**: If your data is too large, we recommend downloading the command-line version of MRMD3.0 from github and using it on a better configured machine.
- Instructions**:
 - start_index**: The first feature of the feature selection interval.
 - end_index**: The last feature of the feature selection interval.
 - step_length**: stride of feature selection.
 - dimensions**: The dimension of a data set after dimensionality reduction.
 - classifier**: RandomForest=1, SVM=2, Bayes=3.
 - rank method**: PageRank = 1, HITS:Authority = 2, HITS:Hub = 3, LeaderRank = 4, TrustRank=5.
 - evaluate metric**: f1_score=1, acc=2, precision=3, recall=4, auc=5.
- File Type**: Supported data file formats are arff, libsvm or csv.
- MRMD3.0**: If you have any problems, please contact me. heshida@tju.edu.cn.

The footer of the page reads: University of TSUKUBA & UESTC.

After running, you will see the following page:

A: The highest metric of MRMD3.0 feature search post-cross-validation

B: Top N feature visualization chart C Score of features after ranking

Webserver Guide:

Parameters:

- **start_index**
The start feature index of the feature selection (default 1)
- **end_index**
The last feature index of the feature selection (default -1)

Note: Usually, `start_index` and `end_index` are used together to reduce the dimensionality of a certain range of data features. By default, MRMD3.0 performs dimensionality reduction on all features (**`start_index = 1` and `end_index = -1`**)

- **length**
stride of feature selection(default 1)

Note: By default, when using the IFS feature search, each feature is added one by one to search. Here you can increase 1 to make the algorithm execute faster. For example, if it is changed to 3, every 3 features will be searched in groups.

- **Top_n**
Save top n features to chart

Note: default 15,include: violin plot, heatmap, stem map)

- **Dimensions**
set the number of features for feature selection.

Note: which is automatically calculated by default and does not need to be set, on the contrary, you can also set it to k, it will save the top k features.

- **Classifier**
Classifier to use when inferring the number of features via IFS
(Randomforest,SVM,Bayes)

Note: The classifier used when using IFS to select the most suitable dimension, choose according to your own needs, and recommend a random forest with stable performance.

- **Rank method**
Various ensemble strategies of MRMD3.0
(PageRank ; HITS:Authority ;HITS:Hub ;LeaderRank ;TrustRank.)

Note:

PageRank: Calculate the weight of a node based on the number of incoming links (and the weight of the link source) of each node. The advantage is that it is simple and easy to understand.

TrustRank: Some high-quality features are added before MRMD3.0 feature sorting, and these features need to be obtained in advance by other methods. MRMD3.0 uses some features recommended by PageRank by default.

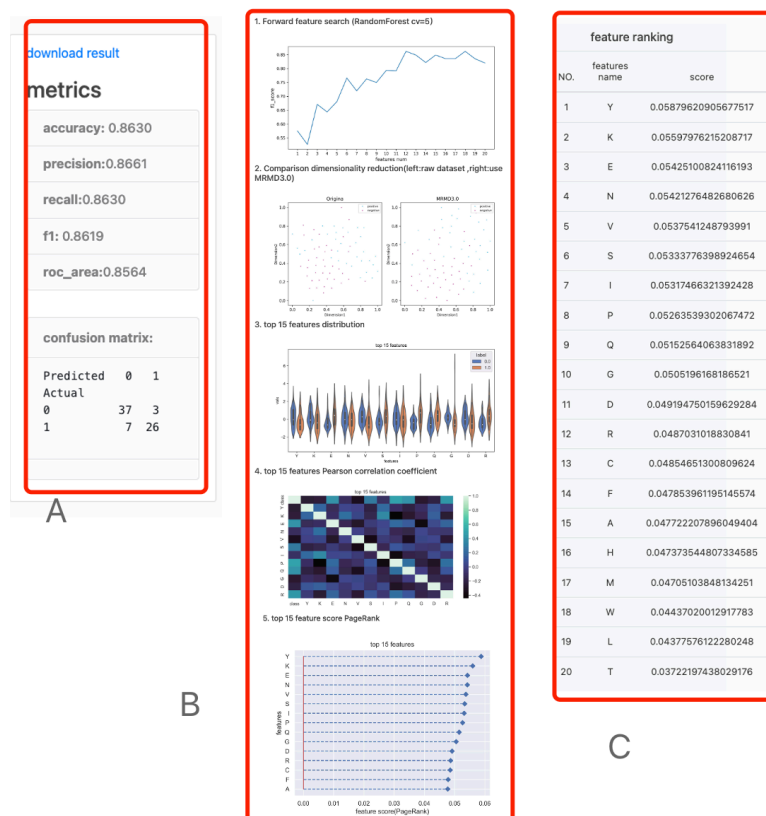
HITS: The HITS algorithm takes into account both incoming and outgoing link characteristics. Which method to choose depends. Hubs focus on pages that are linked to by authoritative pages. authorities are pages with links from the hub.

LeaderRank: It not only considers incoming links (like Pagerank), but also outgoing links and local network structure. A key feature of Leaderrank is that it uses a global "ground node" interconnected with all other nodes to solve the problems of rank leakage and rank pooling. The advantage of LeaderRank is that it can better capture the local influence of nodes in the network.

- **Evaluate metric**
(f1_score, accuracy, precision, recall,auc)

Note: When using the IFS method to select the most suitable dimensions, the recommended metrics are as follows: for balanced data, accuracy is recommended; For slightly imbalanced data that require a balance between recall and precision, the F1 score is recommended; If your application requires minimizing false positives (incorrectly predicting positive cases), then you should focus on precision. If your application needs to minimize false negatives (failing to identify actual positive cases), then recall should be the primary focus. and for extremely imbalanced data, The AUC parameter is recommended.

Demo Result:



6 Summary

MRMD3.0 is a comprehensive Python-based toolkit for dimensionality reduction. We provide 21 feature ranking algorithms and four ensemble strategies. In addition, MRMD3.0 uses the IFS feature search strategy to select features and automatically search for the size of the dimension after dimensionality reduction. In addition, compared with the previous version, an important change of MRMD3.0 is to support a single feature selection method, which is not available in the previous version. Finally, our MRMD3.0 can automatically draw five different types of charts to help users conduct data analysis.

Reference

Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *the Journal of machine Learning research* 12 (2011): 2825-2830.

Ding, Chris, and Hanchuan Peng. "Minimum redundancy feature selection from microarray gene expression data." *Journal of bioinformatics and computational biology* 3.02 (2005): 185-205.

Zhao, Zhenyu, Radhika Anand, and Mallory Wang. "Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform." *2019 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 2019.

Albanese, Davide, et al. "Minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers." *Bioinformatics* 29.3 (2013): 407-408.

Zou, Quan, et al. "A novel features ranking metric with application to scalable visual and bioinformatics data classification." *Neurocomputing* 173 (2016): 346-354.

He, Shida., et al., MRMD2.0: A Python Tool for Machine Learning with Feature Ranking and Reduction. *Current Bioinformatics*, 2020. 15(10): p. 1213-1221.

Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine." *Computer networks and ISDN systems* 30.1-7 (1998): 107-117.

Kleinberg, Jon M. "Hubs, authorities, and communities." *ACM computing surveys (CSUR)* 31.4es (1999): 5-es.

Kleinberg, Jon M. "Authoritative sources in a hyperlinked environment." *Journal of the ACM (JACM)* 46.5 (1999): 604-632.

Lü, Linyuan, et al. "Leaders in social networks, the delicious case." *PloS one* 6.6 (2011): e21202.