Ensemble machine learning based on a voting method using red wine dataset for business education

Motokazu Moritani
Graduate School of Media and Governance
Keio University
5322 Endo, Fujisawa, 2520882 JAPAN
katidoki@sfc.keio.ac.jp

Yoshiyasu Takefuji
Graduate School of Media and Governance
Keio University
takefuji@sfc.keio.ac.jp

***Abstract*— This paper shows the effectiveness of ensemble machine learning based on a voting method. Three machine learning algorithms including Random Forest, Gradient Boosting, LDA (Linear Discriminant Analysis) were individually evaluated using Red-wine dataset. The voting method is to classify wine quality by voting weighted algorithms. Experimental results show that the proposed voting method is better than any individual machine learning methods for the red wine classification problem.**

Keywords— big data, ensemble machine learning, voting method, random forest, gradient boosting, linear discriminant analysis

I. INTRODUCTION

A variety of algorithms have been proposed for prediction. Especially, statistical learning methods such as Ordinary Least Squares (OLS) and Robust Linear Model(RLM), have been used in business areas. However, in recent years, instead of conventional statistical methods, the effectiveness of machine learning has been demonstrated and it has been used in various fields. Among them, ensemble machine learning has been drawing attention without heavy capital investments such as GPU Deep Learning. With the rapid progress of open source machine learning library (sklearn: scikit-learn), they have been used in business fields.

This paper demonstrates the effectiveness of a voting ensemble machine learning method. The voting method classifies red wine qualities by combining results of some statistics and/or machine learning algorithms which are called bases classifiers. In order to evaluate the quality of the algorithms, prediction accuracy using test dataset is used in this paper for measuring goodness-of-fit in classification. For evaluation, by comparing the voting method with each bases classifier, we confirm that the voting method provides higher classification accuracy than individual bases classifier.

This paper details dataset used in Section 2, and the machine learning algorithms. In Section 3, we evaluated and compared the proposed experimental model. Section 4 concludes this paper.

II. THE PROPOSED METHOD

A. *Red wine dataset*

The dataset used in this paper contains "Wine Quality" in Machine Learning Repository of UCI (University of California Irvine) [1]. This dataset was used for research by Cortez et al. In 2009 [2]. Since then, it has been used for model evaluation of machine learning [3~6]. There are 1599 pieces of wine data as the contents of the dataset, each of which has 10 levels of quality evaluation and 11 chemical parameters. Table 1 shows 11 kinds chemical parameters and data statistics (minimum, maximum, and mean value of each parameters) of red-wine dataset.

Table 1 Red-wine data statistics

Chemical Parameter	Min	Max	Mean
Fixed Acidity	4.600	15.90	8.320
Volatile Acidity	0.120	1.580	0.579
Citric Acid	0.000	0.470	0.271
Residual Sugar	0.900	15.50	2.539
Chlorides	0.012	0.611	0.087
Free Sulfur Dioxide	1.000	72.00	15.87
Total Sulfur Dioxide	6.000	289.0	46.47
Density	0.990	1.004	0.997
pH	2.740	4.010	3.311
Sulphates	0.330	2.000	0.658
Alcohol	8.400	14.90	10.42

B. *Proposed methods*

There are many articles using red-wine dataset. Among those papers, an algorithm called "Random Forest" is widely used in recent years. Random Forest is one of machine learning methods called ensemble learning which attracts attention in recent years and its effectiveness has been highly appreciated [7,8]. In this research, several statistics and ensemble learning including Random Forest are compared and the proposed voting method is evaluated.

III. Experimental Result

A. *Dividing dataset*

First of all, to start the experiment, we have to divide the dataset of 1599 pieces of red wine into three datasets for training, for developing, and for testing respectively. The training dataset is used to learn the model, the developing dataset is used for tuning each model, and the test dataset is used for evaluating the proposed models. The ratio of the training dataset, the developing dataset, and the test dataset was set to 0.64: 0.16: 0.2 (1023:256:320). In order to implement the dataset separation, the following scheme is used in Python programming (importing library, loading dataset from csv file, train_test_split dataset):

```
from sklearn.model_selection import train_test_split
import pandas as pd
import
data=pd.read_csv('red-wine.csv')
x=data[["fixed acidity", "volatile acidity", "citric acid", "residual sugar", "chlorides", "free sulfur dioxide", "total sulfur dioxide", "density", "pH", "sulphates", "alcohol"]]
y=data["quality"]
X,x_test,Y,y_test=train_test_split(x,y,test_size=0.2, random_state=0,stratify=y)
x_train,x_dev,y_train,y_dev=train_test_split(X,Y, test_size=0.2,random_state=0,stratify=Y)
```

B. *Choosing Base Classifier*

To run voting classifier, it is necessary to set base classifier to be used. Since base classifier is depending on the dataset, we use training dataset and developing dataset in order to verify the accuracy of algorithms. Results of evaluated accuracy of every algorithm are shown in Table 2.

Table 2 Base Classifier Default Precision

Algorithm	Precision (%)
Linear Discriminant Analysis (LDA)	0.609
Support Vector Machine (SVM)	0.551
K Nearest Neighbors (KNN)	0.516
Gaussian Naïve Bayes (GNB)	0.523
Random Forest (RF)	0.652
Gradient Boosting (GB)	0.645

In this Section, important codes [importing library, clf: classifier, p:predicted output, clf.score(x,y): precision] of each ensemble machine learning are shown as follows:

Linear Discriminant Analysis:

```
from sklearn.discriminant_analysisimport\
LinearDiscriminantAnalysis
clf = LinearDiscriminantAnalysis()
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

Support Vector Machine:

```
from sklearn.svm import SVC
clf = SVC ()
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

K Nearest Neighbors:

```
from sklearn.neighbors import KneighborsClassifier
clf = KNeighborsClassifier()
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

Gaussian Naïve Bayes:

```
from sklearn.naive_bayes import GaussianNB
clf = GaussianNB ()
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

Random Forest:

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

Gradient Boosting:

```
from sklearn.ensemble import GradientBoostingClassifier
clf = GradientBoostingClassifier()
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

Several algorithms (in this case SVM, KNN, RF and GB) can further improve the accuracy of dataset by tuning hyperparameters. In this experiment, we employ an open source library called "Hyperopt" for hyperparameter tuning. The sample source tuning is shown in Table 3. Table 3 shows the difference between before and after tuning the hyper parameter.

Table 3 Hyper Parameter Tuning Result

Algorithm	Before Tuning (%)	After Tuning (%)
SVM	0.551	0.563
KNN	0.516	0.590
RF	0.652	0.699
GB	0.645	0.727

From the result of Table 3, the accuracy of three algorithms (KNN, RF, GB) except SVM is dramatically increased by tuning with the hyper parameter.

C. Voting Method Tuning

From Table 2 and Table 3, it was found that three algorithms including GB, RF and LDA algorithms are suitable for the red-wine dataset. Therefore, these three algorithms are used with the proposed voting method as base classifiers. With base classifier, the voting method needs to be weighted each base classifier. In our experiment, weights of 1 to 10 were set for each, and the optimum weight was searched by using the training dataset and the developing dataset. As a result, it was found that the optimal weighting in this dataset was 5: 2: 2 (RF: GB: LDA) and the accuracy in the development set was 0.707.

D. Implemented Voting method in Python

The Python program for Voting with RF, RB and LDA is described in Fig. 1.

```
import pandas as pd
import numpy as np
from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier as kn
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.discriminant_analysis import \
LinearDiscriminantAnalysis
from sklearn.model_selection import train_test_split
data=pd.read_csv('red-wine.csv')
x=data[["fixed acidity", "volatile acidity", "citric acid",
 "residual sugar", "chlorides", "free sulfur dioxide", "total
 sulfur dioxide", "density", "pH", "sulphates", "alcohol"]]
y=data["quality"]
clf1=RandomForestClassifier(min_samples_leaf=1,
 n_estimators= 243, min_samples_split= 2, random_state= 0,
 criterion="entropy",
 min_weight_fraction_leaf=0.001201368114858965,
 max_features="log2")
clf2=GradientBoostingClassifier(loss="deviance",
 \learning_rate=0.1898731630747871, min_samples_leaf= 1,
 n_estimators=490,min_samples_split=2,random_state=0,
 criterion="friedman_mse",max_features="log2",
 max_depth=47)
clf3=LinearDiscriminantAnalysis()
X,x_test,Y,y_test=train_test_split(x,y,test_size=0.2,\
 random_state=0,stratify=y)
x_train,x_dev,y_train,y_dev=train_test_split(X,Y,
 test_size=0.2,random_state=0,stratify=Y)
clf=VotingClassifier(estimators=[('rf',clf1),('grad',clf2),
 ('lda',clf3)], voting='soft',weights=[5,2,2])
clf.fit(x_train,y_train)
print clf.score(x_dev,y_dev)
```

Fig.1 voting.py

E. Evaluation

At this Section, we compared the accuracy of three base classifiers and voting method using test dataset. Table 4 shows the accuracy of developing dataset and test dataset.

Table 4 Model Evaluation

Algorithm	Developing Dataset (%)	Test Dataset (%)
RF	0.699	0.706
GB	0.727	0.697
LDA	0.609	0.609
Voting	0.707	0.719

In the developing dataset used for hyper parameter tuning, GB has the highest accuracy. In the developing dataset used for hyper parameter tuning, GB has the highest accuracy. On the other hand, the RF with low precision in the developing dataset showed the highest accuracy among the three algorithms in the test data. Most importantly, the voting method improves the accuracy by 1% or more against RF. From these results, it shows that the voting method may yield the best accuracy exceeding the RF.

IV. CONCLUSIONS

This paper justified the effectiveness of the proposed voting method using red wine dataset. Experimental results showed that the accuracy of the proposed voting was by 1% higher than the RF with the highest accuracy among the base classifiers. From this result, the proposed voting method is more accurate than any individual base classifier.

In our experiment, the weighting range of the proposed voting method is from 1 to 10. However, the accuracy can be further improved by widening the range of weighting, by combining different types of base classifiers, and/or by changing the number of combinations. Because, the accuracy of the proposed voting method depends on the accuracy of the individual base classifier. As long as more versatile algorithms over RF can be used, the higher accuracy of the voting method may be achieved.

REFERENCES.

[1] UCI Machine Learning Repository
URL: http://archive.ics.uci.edu/ml/index.php

[2] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
"Modeling wine preferences by data mining from physicochemical properties." In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

[3] Andrii Shalaginov, Katrin Franke
"Multinomial classification of web attacks using improved fuzzy rules learning by Neuro-Fuzzy" International Journal of Hybrid Intelligent Systems, vol. 13, no. 1, pp. 15-26, 2016

[4] Toshiki Sato, Yuichi Takano, Ryuhei Miyashiro
"PIECEWISE-LINEAR APPROXIMATION FOR FEATURE SUBSET SELECTION IN A SEQUENTIAL LOGIT MODEL" Journal of the Operations Research Society of Japan
Vol. 60, No. 1 pp. 1-14, 2017

[5] P. Appalasamy, A. Mustapha, N.D. Rizal, F. Johari and A.F. Mansor "Classification-based Data Mining Approach for Quality Control in Wine Production" Journal of Applied Sciences, 12: 598-601. 2012

[6] Julien-Charles Lévesque, Christian Gagné, Robert Sabourin
"Bayesian hyperparameter optimization for ensemble learning"
UAI'16 Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence Pages 437-446

[7] Y. Takefuji and K. shoji,
"Effectiveness of ensemble machine learning over the conventional multivariable linear regression models,"
Proc. of Hawaii International Conference on Education, 2016

[8] Y. Takefuji, Ensemble methods significantly improve prediction (12 April 2017), eLetter science
URL: http://science.sciencemag.org/content/355/6324/515/tab-e-letters