Training issue (defect factor analysis)

**Skydisc , Inc**

**Created by Yasunori Kajiwara on 2018/5/6**

**Requirement**  
The file "molding\_A\_20171030.csv" is the manufacturing data on an injection molding machine (plastic manufacturing machine). Quality results are listed in the variable RESULT\_QUALITY. Good is good product, others (ColorUnevenness, Short, Mera) are defective products. It is client's need to analyze which variable affects defective product generation for each type of defective product. Please perform analysis according to the following procedure.

**Language/AI flamework**

* Python3.x
* scikit-learn

**Development environment**

Jupyter notebook

**Submission**

Jupyter notebook file

**Analysis procedure**

1. Read the file "molding\_A\_20171030.csv" and create the dataframe.
2. Count the number of Good, ColorUnevenness, Short, Mera.
3. In the following, we will first analyze variables that have a large influence on the occurrence of ColorUnevenness.

Extract only the data of Good and ColorUnevenness

1. Calculate the number of missing values ​​in each variable.
2. Calculate the basic statistics (average value, standard deviation, median value, maximum value, minimum value, etc.) of each variable with Good and ColorUnevenness as the aggregation axis and output the result as csv file.
3. For each variable, investigate outliers, and if there are outliers, delete that line. Please survey outliers by visualizing the data.
4. For variables whose percentage of missing is over 20%, please consider deleting the variable or complementing the missing and use it in subsequent analysis.  
   （The following is an example of way of consideration）  
   Different colors for Good and ColorUnevenness and draw a histogram of each variable. If there is a variable whose distribution is largely different between Good and ColorUnevenness, we judge that the variable is a variable that distinguishes Good from ColorUnevenness, leaving the deficiency complemented. Alternatively, using a feature selection algorithm, delete variables for which importance is determined to be low.
5. Delete the unnecessary variable, and complement the missing value of the remaining variable with the average value of that variable.
6. It is so-called imbalance data that there is a big difference in the number of data in Good and ColorUnevenness. There are two basic ways to cope with unbalanced data.

One downsamples the one with the larger number of data (Good this time). The other is weighting the objective variable in the machine learning model. We will deal with down sampling below. Thin out Good by random sampling so that the ratio with ColorUnevenness is 7: 3.

1. Please convert Good to 0, ColorUnevenness to 1.
2. Please divide the data into learning data and verification data at a ratio of 7: 3.
3. The model uses a random forest. Is scale conversion (standardization etc.) of variables necessary? Please execute it if necessary. Please state the reason why you think that it is not necessary if you do not need it.
4. Create a random forest model under the following conditions and calculate the importance of the variable.  
   【Model creation condition】  
   Please use GreeSearchCV of scikit-learn for learning data and search parameters by grid search. Evaluate the model by using stratified k-split cross validation and judge the model with the highest AUC as the best model. (The part of k is 3, 5, 10 in general.) 10 if the number of data is sufficient, and k as the number of data decreases)
5. After executing the above, please display the best parameters. Also, please use the best model to verify accuracy with validation data.

In verification of accuracy, please check the following with each of **learning data and verification data**.  
1. Accuracy  
2. ConfusionMatrix

3. Draw a ROC curve and calculate AUC

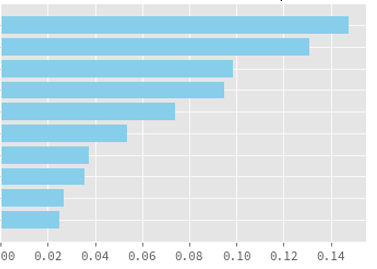
4. Confirm the classification report of Scikit-learn (precision, recall, f1 score)

\* If the accuracy is not sufficient, draw a precision-recall curve and try to consider changing the threshold. For example, if recall is low, recall will rise if you lower the threshold. However, precision goes down. Precision and recall are in a trade-off relationship. By changing the threshold value, precision does not drop so much, sometimes you can greatly improve recall. Drawing the precision-recall curve helps you to judge changes in precision and recall when changing the threshold.

1. Check whether overlearning has occurred because of accuracy evaluation.

If you decide that you are over learning, please try again to suppress excessive learning. When judging that it is not over learning, please state the reason.

1. Find the variable top10, which has a big influence on the occurrence of ColorUnevenness, and visualize it with a graph. The output image is as follows.



VariableA

VariableB

Variable G

Variable H

Variable affecting ColorUnevenness Top 10

Variable importance

Please submit once after completing step ⑯.

In addition, we coped with unbalanced data by downsampling in procedure ⑨, but please deal with other defective items (Short, Mera) by weighting in the model. (If it is RandomForestClassifier of scikit-learn, it can be weighted to the target variable by specifying argument class\_weight)