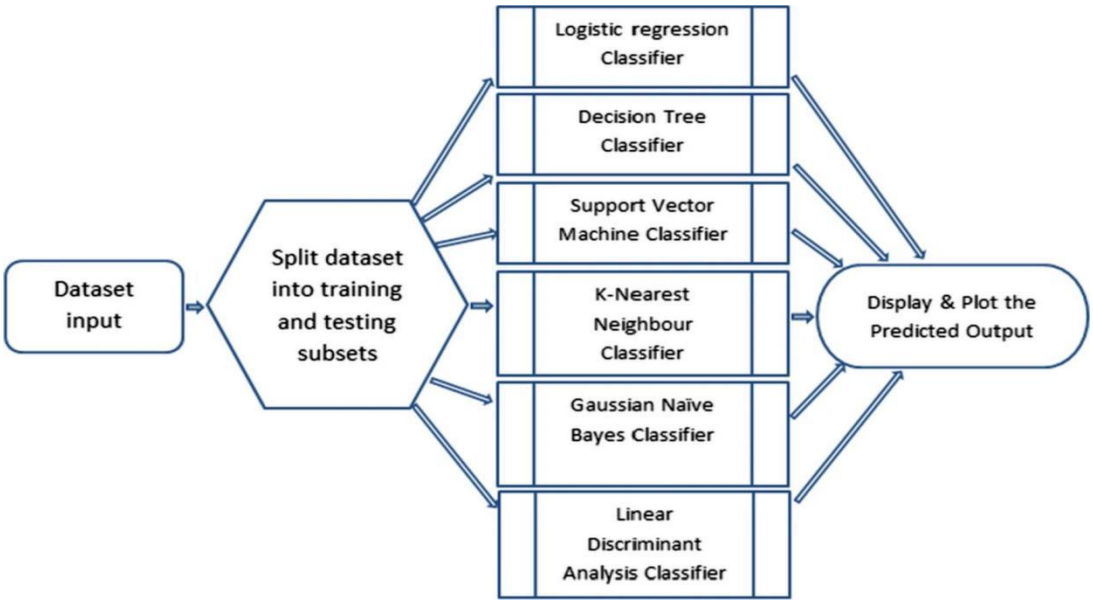
Phishing Site Prediction Using Foundation and Integrated Classifier Techniques with Cross Validation

There are total 12 machine learning classifier algorithms which are categorized into 2 types; integrated classifiers and basic classifiers. All 12 machine learning classifier algorithms use all 30 features to find the best prediction algorithm for predicting phishing websites.

1. Basic Classifiers The basic machine learning classifiers used in this experiment are: first a logistic regression classifier, followed by a Gaussian plain Bayesian classifier, next a decision tree classifier, next a support vector machine classifier, then a K-nearest neighbor classifier, and then a linear discriminant analysis classifier. The flowchart of these basic classifiers is shown below.1. All the individual basic classifiers are depicted next to the flowchart.



Flowchart of the base classifier

2. Basic classifier

2.1 Logical regression classifier

$$p = \frac{1}{{1 + e^{{ - \left（ { + b_{1} x_{1} + b_{2} x_{2} + \cdots + b_{p} x_{p} } \right）}} }}p$$

= \frac{1}{{1 + e^{{ - \left（ { + b_{1} x_{1} + b_{2} x_{2} + \cdots + b_{p} x_{p} } \right）
}} }}

2.2 Decision tree classifier

1 - \mathop \sum \limits_{i = 1}^{C} \left（ {p_{i} } \right）^{2}基尼 = 1 - \mathop \sum \limits_{i = 1}^{C} \ left（ {p_{i} } \right）^{2}

\mathop \sum \limits_{i = 1}^{C} - p_{i} *log_{2} \left（ {p_{i} } \right）熵 = \ mathop \sum \limits_{i = 1}^{C} - p_{i} *log_{2} \left（ {p_{i} } \right）

2.3 Support vector machine classifier（svm）

left[{\frac{1}{n}\mathop \sum \limits_{i = 1}^{n} {\text{max}}\left（ {0,1 - y_{i} \left （ {w \cdot x_{i} - b} \right）} \right）} \right] + \lambda \left|{\left| w \right|}\right|^{2}

\left[{\frac{1}{n}\mathop \sum \limits_{i = 1}^{n} {\text{max}}\left（ {0,1 - y_{i} \left （ {w \cdot x_{i} - b} \right）} \right）} \right] + \lambda \left|{\left| w \right|}\right|^{2}

2.4 Gaussian naive Bayesian classifier

left（ {c|x} \right） = \frac{{P\left（ {x|c} \right）P\left（ c \right）}}{P\left（ x \right）}P\left（ {c|x} \right） = \frac{{P\left （ {x|c} \right）P\left（ c \right）
}}{P\left（ x \right）}

(8)

P\left（ {c|X} \right） = P\left（ {x_{1} |c} \right） \times P\left（ {x_{2} |c} \right）

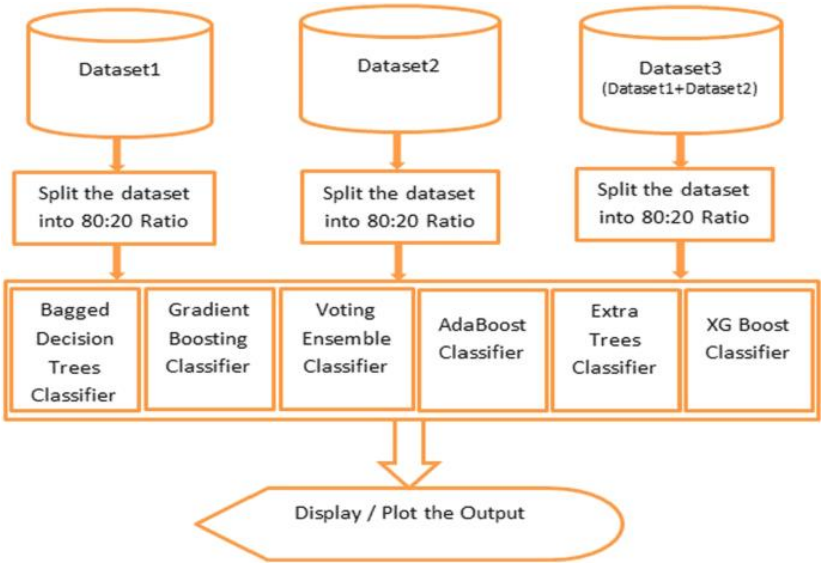
\times \cdots \times P（x_{n} |c） \times P\left（ c \right）P\left（ {c|X} \right）

= P\left（ {x_{1} |c} \right） \times P\left（ {x_{2} |c} \right） \times \cdots \times

P（ x_{n} |c） \times P\left（ c \right）

3. Classifier based on integration

In the next stage, an integration-based machine learning classifier will be used. First, use the bagging classifier. Secondly, Adaboost classifier is used. Next, use the gradient lifting classifier, and then use the voting ensemble classifier. Finally, an additional tree classifier is used, followed by XGBoost classifier. We combine ten-fold cross validation for training, testing models and analyzing data sets. The process diagram of these integration-based classifiers without cross-validation is shown in the following figure. 2. All single classifiers based on integration are described next to the flow chart, whether they are verified or not



Flow chart of integrated classifier

3.1Integrated classifier used

Packed with decision tree (Bagging) classifier

Adaboost classifier

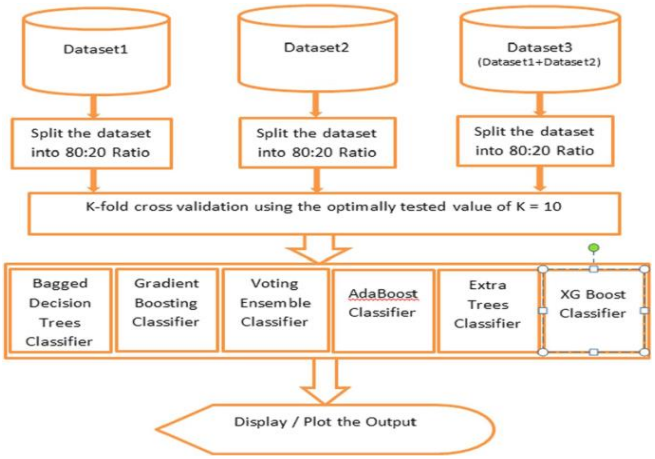
Gradient lifting classifier

Voting integrated classifier

Additional tree classifier

XGBoost classifier

4.Classifying the merged data set（这里他进行了数据集）



The cross-validation process diagram of this process is shown in the following figure.

5.K-fold cross-validation

This is a technique that can improve the retention method. It changes the way we choose data sets for training and testing; Instead, it divides the data set into k subsets and repeats the Holdout method k times. The steps of K-fold cross-validation are as follows.

1.
It breaks our data set into k subsets, which is called folding.

2.
For each fold created from our data set, build a model on the created k-1 fold, and these folds will test the model to check the validity of the model for the k-th fold.

3.
Repeat this process until you test each k by folding it into an independent test set.

4.
Record the results, and then take out the average of the prediction accuracy, and use it as the test index of the model currently being tested or realized.

6. conclusion（参考）

In this experiment, the machine learning classifier is implemented using python code. First, the base classifier has been used, and the result is; Firstly, the accuracy of logistic regression classifier is 93.28%, secondly, the accuracy of Gaussian Naive Bayes classifier is 91.44%, and then the decision tree classifier is used. Its prediction accuracy is 95.87%, while the prediction accuracy of support vector machine classifier is 94.80%. Then, the K nearest neighbor classifier is predicted with 93.43% accuracy. Finally, the linear discriminant analysis classifier produced an accuracy of 92.87%.

In the next step, we tested the first data set. Firstly, the accuracy of bagging classifier is 98.78%. Secondly, Adaboost classifier is used. Its accuracy is 95.91%. Then, using gradient lifting classifier, its prediction accuracy is 97.56%, while the prediction accuracy of voting integrated classifier is 97.15%. The prediction accuracy of extra tree classifier is 99.18%. The accuracy of XGBoost classifier is 99.18%. In the next step, we experimented with a second data set. Firstly, the accuracy of bagging classifier is 98.64%. Secondly, Adaboost classifier is used. Its accuracy is 94.21%. Then, using gradient lifting classifier, its prediction accuracy is 96.47%, while the prediction accuracy of voting integrated classifier is 97.19%. Finally, the prediction accuracy of the extra tree classifier is 98.73%. The accuracy of XGBoost classifier is 98.37%.

Next, we conducted experiments on the merged data sets. Firstly, the accuracy of bagging classifier is 98.51%. Secondly, Adaboost classifier is used; Its accuracy is 92.52%. Then, using gradient lifting classifier, its prediction accuracy is 95.63%, while the prediction accuracy of voting integrated classifier is 96.52%. Next, the extra tree classifier is predicted, and the accuracy rate is 98.59%. Finally, XGBoost classifier produces 98.07% accuracy.

In this study, the results obtained from ensemble-based classifiers with and without ten times cross-validation are compared, and the results clearly show that ensemble classifiers perform better when using cross-validation in classification. Like the cross-validation method, when using python to test the splitting with standard training, the data set is split into as many parts as the number of folds selected, instead of being split into two parts as usual. From this experimental study, it is proved that the classification technology based on integration is superior to the basic classifier. When we carried out this experiment, we changed the data set, and observed that when the number of instances was initially reduced from 2456 to 11,055, the result decreased by about 0.2% to 1.2%. However, when we merge two data sets, although we have the same attributes in the generated data set and the total number of instances increases to 13,511, when cross-validation is not applied, the results will be significantly affected. The maximum degradation observed in Adaboost classifier is about 25%, and the minimum degradation observed in gradient lifting classifier is about 6.9%. Extra Trees and XGBoost classifiers provide the most accurate prediction output; Both of them predict that the output score of the first data set is 99.18%, and the output scores of the second data set are 98.73, 98.37% and 98.07, 98.59%. Both of them use cross validation.

After using different data sets and merging two data sets, the number of instances is significantly increased, and the prediction accuracy of the integrated classifier is not reduced, and it is found that it is far ahead of the basic classifier, because the lowest accuracy score of naive Bayesian classifier prediction is 91.44% in the basic classifier, while the lowest score of the integrated classifier prediction is 95.91% in AdaBoost classifier prediction. Among all the basic classifiers used, the maximum accuracy score of decision tree classifier is 95.92, while the maximum accuracy score of extra tree and XGBoost classifier is 99.18%. However, when the third data set (that is, the merged data set) is classified, XGBoost classifier predicts the best accuracy of 88.71% without K-fold cross-validation, while the best accuracy of ExtraTrees classifier is 98.59% through cross-validation. Therefore, although the data set is larger, ExtraTrees algorithm performs best in cross-validation.