Random Forest in Big Data Analysis

Brief by Shen Wang

Introduction

The aim of the project is to use machine learning techniques to train data analysis models to predict the success or fail of start-ups. The technique used in modeling predictions, which I will mainly analyze in this report, is the random forest.

Random forest is a machine learning algorithm that widely used in modeling predictions and big data analysis to deal with classification and regression problems. It runs efficiently on large data sets and can process input without dimensionality reduction. In addition, it has an effective method for estimating missing data. Generally speaking, random forest is an efficient algorithm and it is easy to use in big data analyzation.

Approach

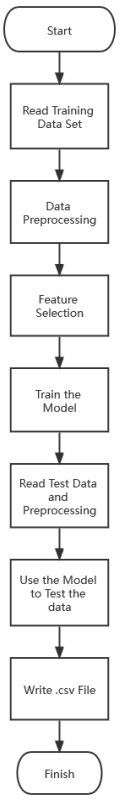

Figure 1. Flow

The work flow of the whole program is as the figure in the left. At the beginning, the program will read the training .xlsx file and preprocess the data to fill the blank cells and translate date type. Then features will be selected by *SelectKBest* and *f_classif* in *sklearn.feature_selection*. After that, the model will be trained by random forest in *sklearn.ensemble*. And the result will be provided by test data. Finally, the result will be written in a .csv file.

Random forest algorithm combines multiple decision trees into a "forest". It repeatedly and randomly selects samples from the original set to train, and then generates decision trees to form a forest. The results of new data are determined by the number of votes in the classification tree. Its essence is an improvement of decision tree algorithm, which merges multiple decision trees together, and the build of each tree depends on the independent extraction of samples.

Results

In the implement of the project, in order to compare the effects of different model training algorithms, I printed the results with the same function parameters.

```
Classifier:  RandomForestClassifier ,  Accuracy: 0.8160377358490566
Classifier:  DecisionTreeClassifier ,  Accuracy: 0.7688679245283019
Classifier:  ExtraTreesClassifier ,  Accuracy: 0.7216981132075472
```

Figure 2. Benchmarks

The three compared algorithms are random forests, decision tree and extra trees respectively. *Accuracy* refers to the accuracy of the training model calculated by the *sklearn.metrics.accuracy_score* function. According to the running results, random forest has the highest accuracy, which is 0.816. The accuracy of decision tree is 0.768, and that of extra trees is 0.721. It is easy to find that random forest is the best of the three algorithms.

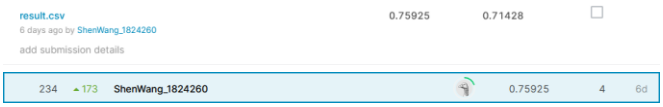

Figure 3. Score

In the Kaggle competition, my score is 0.75925.

Pros/Cons

- Pro: It can process data with high dimensions without feature selection, because feature subsets are selected randomly.
- Pro: Trees are independent during training that the training speed is high and it is easy to parallelize.
- Con: Random forests may be overfitting on some distributed classification or regression problems.
- Con: For the data with different attribute values, it will have an impact on the random forest. The weights produced by the random forest on such data are not credible.

Conclusion

In summary, random forest is a convenient and efficient machine learning algorithm for training data analysis models. It is an improvement of decision tree algorithm and can analyze a large number of random samples in parallel. However, it may overfit on some problems, which need to pay attention to.