# Predicting the Risk of Enterprise Exit

**Zhenxian Zheng** **Jinjin Lin** **Xiaoyue Xiao**
**17214726** **17214532** **17274695**

## Abstract

*Predicting the risk of enterprise exit is one of the competition questions of data mining competition in CCF2017. The purpose is to model enterprises by analyzing the behavior data of small businesses, and analyze the risk of business failures in the next two years. In this paper, we analyze the match data and carry out feature engineering from data, including data preprocessing and feature processing, which generate a large number of effective features for further training. In model training, we make use of techniques of stacking. A two-level stacking framework is used to predict more accurately and more generally. We also conduct three experiment to validate our stacking models, and results show the effectiveness of our approach. Finally, we rank 9th in 569 teams.*

### 1. Introduction

Traditional enterprise evaluation is based on corporate financial information, borrowing and recording information, to judge the business status and whether it is possible to default [1]. There is no doubt that the evaluation method is more objective and reasonable for the large and medium-sized enterprises with a sound finance and record in the field of traditional bank lending. However, for a large number of small and micro enterprises, not only publicly available real business financial information, nor the public credit information of these enterprises, in the absence of variables, how to use the variable operating conditions weak objective and impartial evaluation of enterprise, is the major problem of the contest problems need to solve [2].

The contest from the more than 20 million enterprises in the country drawn part of the business (after desensitization), to provide business entities in many aspects of behavioral information data. We need to build predictive models of whether the business will run poorly in the future through data mining techniques and machine learning algorithms, and output the risk prediction probability values.

### 2. Background

#### 2.1. Problem Description

Today, data has become a strategic resource and economic asset. Data mining and machine learning methods are used to analyze massive data. Traditional enterprise evaluation mainly based on the enterprise's financial information, credit history information and the possibility of default and other credit information to determine the business status of the company [3]. For financially sound, large and medium-sized enterprises that have records in traditional areas of bank lending, the evaluation method is undoubtedly more objective and reasonable. However, for a large number of small and medium-sized micro-enterprises, they can neither publicly obtain the real financial information of the enterprises nor the public credit information of these enterprises. In the absence of strong variables, how to make use of the weak variables to objectively and justly evaluate the business status of the company is the main problem to be solved.

#### 2.2. Dataset description

In this contest we have two kinds of data. The first one is Enterprise identity information (desensitization) and the behavior data of the enterprise within a certain period of time. This data is the same for both the training set and the evaluation set. The second one is target data. The target value for the business in August 2017 when the operating conditions: closed down 1, normal operation 0.

This table only has training data. Players form their own characteristics and data formats, free to combine the proportion of training and testing data from the data. To protect the data, all data has been sampled and desensitized as necessary.

There are null values or NULL for some columns in the data. There may be duplicates in the records. Participants are advised to deal with them according to the data field description.

### 3. Method

Feature Engineering is the process of transforming raw

data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. The original data in the project contains a great deal of noise, and the expression of information is not concise enough. Therefore, the purpose of the feature engineering is to use a series of engineering activities to express such information in a more efficient way of coding. With the information represented by the feature, the rules contained in the original data are still retained. In addition, the new encoding method also needs to minimize the impact of uncertainty in the original data. Feature engineering plays an important role in the accuracy of the result in this project. Feature engineering mainly includes two parts: data preprocessing and feature processing. The main flow chart is shown in Fig.1.

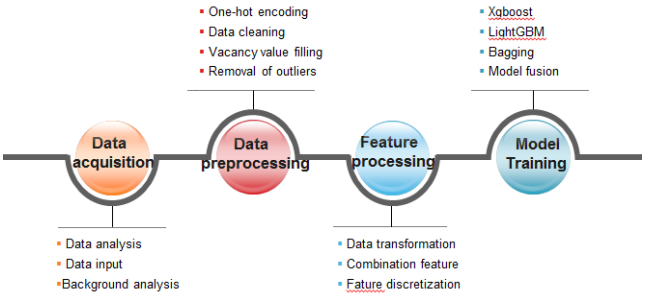


Fig.1: Flow chart of proposed algorithm

### 3.1. Data preprocessing

#### 3.1.1. Data cleaning

In this project, we need to get numerical values, but the original data contain a large number of class features, random codes and Chinese field names, so we need to clean the data and get numerical data that can be trained. For instance, for different types of data, we represent the original meaning of the data by means of different numbers. Only when all the data are converted to numerical data, model training can be predicted**.**

#### 3.1.2. Vacancy value filling

As the enterprise data comes from the real scene, there is a problem of partial information missing. When the class label is missing, the empty value can be removed by the method of ignoring the tuple. Most of this project uses the mean of data and the discrete value to fill the empty value. For example, there are a lot of missing values in each of the 6 indicators of each enterprise, and we usually use 0 to fill it out.

#### 3.1.3. Removal of outliers

The reality of enterprise data is a major problem and the data noise which is usually due to data collection. For instance, there are many discrete points in feature of the registered capital of the enterprise, in this project we mainly detect abnormal points by distance method, the data set and the distance between the point is greater than a certain threshold points as outliers.

#### 3.1.4. One-hot encoding

Qualitative features cannot be directly used: some machine learning algorithms and models can only accept quantitative characteristics input, so we need to transform qualitative features into quantitative features. The simplest way is to specify a quantitative value for each qualitative value, but this way increase the work of the parameter tuning. Usually, the qualitative feature is converted to the quantitative feature by the way of single thermal encoding. Assuming that there are qualitative values of $N$, this feature is extended to $N$ characteristics. When the original characteristic value is $I$ qualitative value, the $I$ extension feature assignment is 1, and the other extension feature assignment is 0. The way of single thermal encoding does not increase the work of parameter adjustment compared to the way of direct designation. For a linear model, the characteristic of using the single thermal code can achieve the nonlinear effect.

| Sample name | ETYPE |
|---|---|
| sample 1 | 6 |
| sample 2 | 7 |
| sample 3 | 8 |
| sample 4 | 16 |
| sample 5 | 17 |

| Sample name | ETYPE 1 | ETYPE 2 | ETYPE 3 | ETYPE4 | ETYPE 5 |
|---|---|---|---|---|---|
| sample 1 | 1 | 0 | 0 | 0 | 0 |
| sample 2 | 0 | 1 | 0 | 0 | 0 |
| sample 3 | 0 | 0 | 1 | 0 | 0 |
| sample 4 | 0 | 0 | 0 | 1 | 0 |
| sample 5 | 0 | 0 | 0 | 0 | 1 |

Fig.2: A case of one-hot encoding

### 3.2. Feature processing

Feature combination creates a subset of new features by combinations of the existing features. The purpose of the feature combination is to increase the available features, improve the stability of the model, and add a large number of training features through the combination of features. This project mainly adopts data transformation, combination feature and feature discretization.

#### 3.2.1. Data transformation

Data transformation is mainly used for the feature of time sequence. Such information cannot be directly used. For example, time characteristics need to extract information such as year and month, so that further analysis can be carried out. Some basic arithmetic operations for different features can also produce a large number of effective features. For example, the polynomial feature used in the project is an effective way. Assuming that there are 3 features $x1$, $x2$, $x3$, the polynomial characteristic of degree 2 can be extended to 9 features.

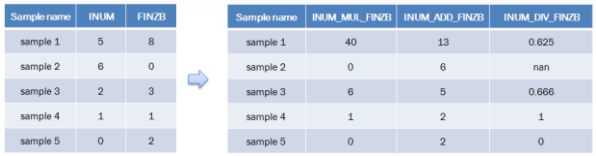


Fig.3: A case of data transformation

### 3.2.2. Combination feature

Combination features are also important for this project. The linear model and decision tree model are not accurately depicted for the nonlinear relationship, and the combination of features can just join the nonlinear expression and enhance the expressive ability of the model. By classifies numerical features by category features, we can get many meaningful features, and increase the stability and accuracy of the models. For example, we can classify the registered capital of an enterprise according to its industry and business type, and calculate the mean and variance of the total capital of every industry, so that we can get statistical significance. And we can count the number of each enterprise's investment and the proportion of its branches in the province and so on.

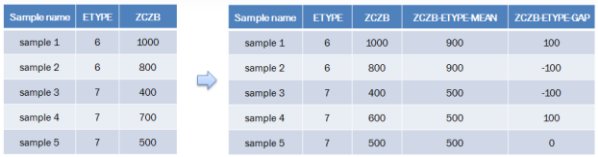


Fig.4: A case of combination feature

### 3.2.3. Feature discretization

Feature discretization is also an effective way of feature processing. In real data, it is not always necessary to get all data partition. By setting threshold, data can be divided, and good results can also be achieved. Usually, the following formula is shown in formula below.

$$x' = \begin{cases} 1 & x > threshold \\ 0 & x \le threshold \end{cases}$$

In order to have a more intuitive representation, table 1 show the features of part of the feature engineering.

Table 1: example of features

| feature abbreviation | type | description |
|---|---|---|
| ZCZB_HY_MEAN | double | ZCZB groupby HY mean() of each entbase |
| ALTER_NUM | int | alter numbers of each entbase |
| ALTBE_RGYEAR_SUM | int | ALTBE sum of each year for each entbase |
| ETYPE_INUM_MIN | double | INUM groupby ETYPE min() |
| ENUM_MUL_INUM | double | ENUM multiply INUM |
| RGYEAR_CLASS | int | RGYEAR divide into 20 class |

### 3.3. Modeling

Stacking is an ensemble learning technique to combine multiple classification models via a meta-classifier. In the standard stacking procedure, the first-level classifiers are fit to the same training set that is used prepare the inputs for the second-level classifier, which may lead to overfitting. Our method, however, uses the concept of cross-validation: the dataset is split into k folds, and in k successive rounds, k-1 folds are used to fit the first level classifier; in each round, the first-level classifiers are then applied to the remaining 1 subset that was not used for model fitting in each iteration. The resulting predictions are then stacked and provided as input data to the second-level classifier. After the training, the first-level classifiers are fit to the entire dataset as illustrated in the figure 5.

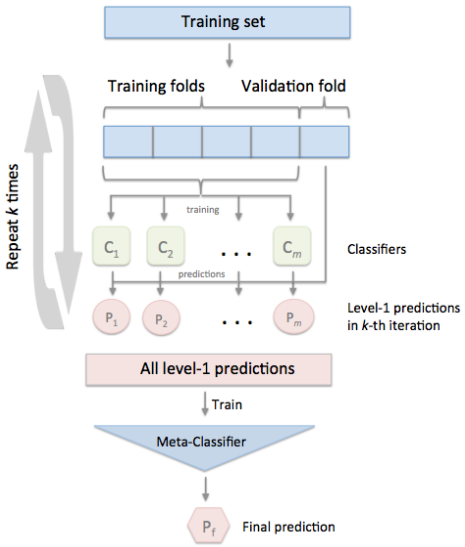


Fig.5: Overview of stacking

So, we want to use techniques of stacking to achieve more effective result. In following subsections, we will talk about which models we choose to be the first level and second level models. Firstly, we will introduce models which we used in first level models. And then, we will present the second models.

### 3.3.1. First level models

Now we are ready to choose some model as the first level models for our stacking framework. There are more than 60 predictive modeling algorithms to choose from. We must understand the type of problem and solution requirement to narrow down to a select few models which we can evaluate. Our problem is a binary classification. We want to identify relationship between output (Exited or not) with other

variables or features (EID, RGYEAR, ZCZB...). We are also performing a category of machine learning which is called supervised learning as we are training our model with a given dataset. With these two criteria - Supervised Learning plus Classification, we can narrow down our choice of models to a few [4]. These include:

- **Decision tree** [5]**:** A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
- **AdaBoost** [6]: AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel Prize for their work.
- **SVM** [7]: A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.
- **Multiple layer perceptron (neural network)** [8]: A multilayer perceptron (MLP) is a class of feedforward artificial neural network.
- **k-NN** [9]: k-nearest neighbor's algorithm (k-NN) is a non-parametric method used for classification and regression. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.
- **Logistic regression** [10]: Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome.

We use the above models as the first level models for our stacking framework. These 6 models guarantee the diversity and generalization of our method, which could offer our predicting model more power.

#### 3.3.2. Second level models

In above subsection, we chose 6 different models as the first level models for our stacking framework. After training of the first level models, the results of these 6 models would be provided as input of the second level models. In this subsection, we also ensure the diversity of our second level models. Except that, the second level models should be more powerful and more explanatory. We choose these 4 models as our second models for stacking framework.

- **Extra trees** [11]: The Extra-Tree method (standing for extremely randomized trees) was proposed with the main objective of further randomizing tree building in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree[15].
- **Random forest** [12]: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- **XGBoost** [13]: XGBoost is an open-source software library which provides the gradient boosting framework for C++, Java, Python, R, and Julia.
- **LightGBM** [14]: LightGBM is a gradient boosting framework that uses tree based learning algorithms. LightGBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise [16].

All these models are tree-based models. There are several distinct advantages of using decision trees in many classification and prediction applications. Decision trees implicitly perform variable screening or feature selection; Decision trees require relatively little effort from users for data preparation; Nonlinear relationships between parameters do not affect tree performance; The best feature of using trees for analytics - easy to interpret and explain to executives. Because of these advantage, tree-based models are an important type of algorithm for predictive modeling machine learning, which is suitable for our second level models.

## 4. Experiment

In this section, we conduct comprehensive experiments to evaluate the effectiveness of our approach. Particularly, we intend to answer the following research questions.

- RQ1: Is there diversity between our first level models?
- RQ2: Is our stacking approach more effective than any other base model?
- RQ3: How would tree depth affects our approach, which we know fact that tree-based models are extremely affected by tree depth? And in this section, the answer will be given.

### 4.1. Evaluation Criterion

The AUC value (AUC's evaluation uses the prediction probability) as the main evaluation criteria, F1 score (F1 score using the prediction results) as a secondary evaluation criterion.

$$F_1 = \frac{2 \times precision \times recall}{(precision \times recall)}$$

The specific rules are: Auxiliary evaluation with F1-score (Precision and Recall) with the same AUC value (reserved for 4 digits after the decimal point).

The score formula is as follows:

$score$ = (AUC takes 4 decimal places) * 10000 + $F_1$

The range of evaluation score is [0, 10001]

### 4.2. Diversity between Base Models

We do pairwise comparison on first level models. The results are provided in figure 6. As we can observe, in most of pairwise comparison the correlation is about 80%. Correlation between decision tree and AdaBoost is highest, 0.91%, which the reason may be that they are all tree-based model. Thus, we can say that, our first level models are be able to be diverse, which would be effective contributors for second level models.

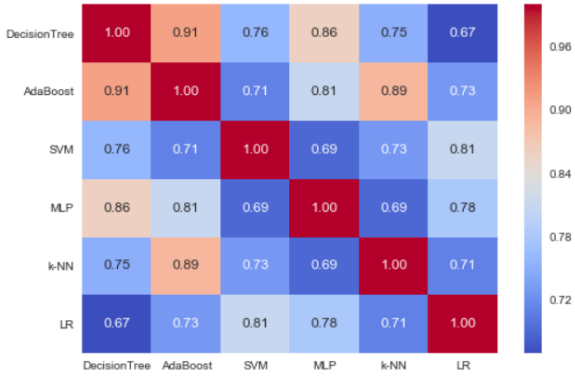


Fig.6: Correlation on first level models

### 4.3. The Effect of Different Learning Models

We also examine the impact of different learning models on the prediction accuracy. We have tried a few popular learning models, including decision tree, AdaBoost, multiple layer perceptron, k-NN, logistic regression, SVM, random forest, XGBoost and LightGBM. The evaluation results in figure 7 show that all the learning models lead to overall good prediction accuracy. In particular, our stacking approach is best effective among all models.

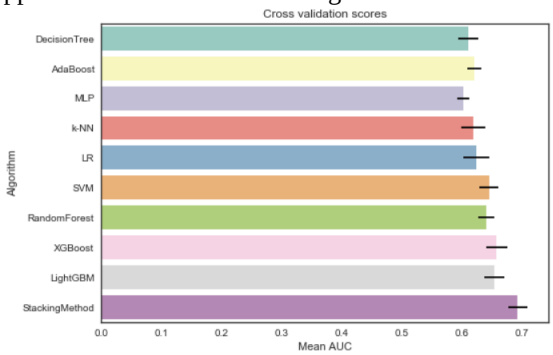


Fig.7: Cross validation scores of all models

### 4.4. The Impact of tree depth

For the number of tree depth, we experimented on a series of tree depth values between 3 to 12, and selected the tree depth that maximizes the AUC score in each model. Figure 8 shows the result. We observe that tree depth equals 8 gives the best AUC scores for all models.

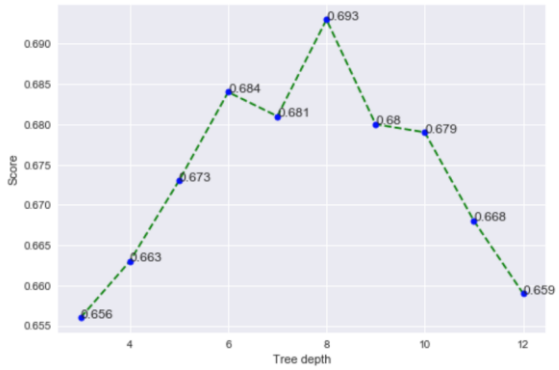


Fig.8: Scores on different tree depth

## 5. Conclusion

In this competition, some important enterprise information can be obtained through the requirements of the competition and the analysis of the data. The key to solving the problem of the predicting the risk of enterprise exit lies in the feature engineering and the model training. What's more, a large number of combination features and effective features ensure that the model can converge to a good value, and model fusion and multiple algorithms improve the accuracy of the result further. A better accuracy will also have some help for future enterprise risk prediction.

## References


[1] Wing C C K. Distribution Reform and Retail Structure in China: An Empirical Analysis of Entries and Exits of Enterprises[J]. Asia Pacific Journal of Marketing & Logistics, 1994, 6(3):3-25.

[2] Shen Q Y, Wang W P, Wang W D. Research on the Evolution of Cluster Scale Based on Trust and Enterprise's Entry and Exit[J]. Chinese Journal of Management Science, 2009.

[3] Datafountain: [EB/OL]http://www.datafountain.cn/#/

[4] competitions/271/intro,2017

[5] Breiman L. Using Iterated Bagging to Debias Regressions[J]. Machine Learning, 2001, 45(3):261-277.

[6] Kohavi, Ron. "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid." KDD. Vol. 96. 1996.

[7] Rätsch, Gunnar, Takashi Onoda, and K-R. Müller. "Soft margins for AdaBoost." Machine learning 42.3 (2001): 287-320.

[8] Joachims, Thorsten. Making large-scale SVM learning practical. No. 1998, 28. Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 1998.

[9] Seiffert, Udo. "Multiple layer perceptron training using genetic algorithms." ESANN. 2001.

[10] Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." IEEE transactions on information theory 13.1 (1967): 21-27.

[11] Wasserman, Stanley, and Philippa Pattison. "Logit models and logistic regressions for social networks: I. An

introduction to Markov graphs andp." Psychometrika 61.3 (1996): 401-425.
[12] Louppe, Gilles, et al. "Understanding variable importances in forests of randomized trees." Advances in neural information processing systems. 2013.
[13] Lindner, Claudia, et al. "Robust and accurate shape model matching using random forest regression-voting." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1862-1874.
[14] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
[15] Ke, Guolin, et al. "LightGBM: A highly efficient gradient boosting decision tree." Advances in Neural Information Processing Systems. 2017.
[16] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. 2016:785-794.
[17] CSDN:introduction of xgboost[EB/OL] http://blog.csdn.net/
[18] a1b2c3d4123456/article/details/52849091,2016
[19] CSDN:lightgbmparameters[EB/OL] http://blog.csdn
[20] .net/niaolianjiulin/article/details/76584785,2015

**Appendices**

### A. Target Data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| TARGET | varchar2(2) | Target label, 1 shut down, 0 normal. |

### B. Evaluation data:

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |

### C. Enterprise basic information data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| PROV | varchar2(4) | Province code |
| RGYEAR | varchar2(4) | The year of the enterprise set up |
| HY | varchar2(8) | Industry category |
| ZCZB | number | Registered capital. RMB: ten thousand yuan. Have been rounded |
| ETYPE | varchar2(8) | Type of enterprise |
| MPNUM | number | The completed identity indicator has been calculated.<br>Empty means 0 |
| INUM | number | |
| FINZB | number | |
| FSTINUM | number | |
| TZINUM | number | |

### D. Alter data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| ALTERNO | varchar2(50) | The code of alter item |
| ALTDATE | date | Date |
| ALTBE | varchar2(4000) | Alter before |
| ALTAF | varchar2(4000) | Alter after |

### E. Branch data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| TYPECODE | varchar2(50) | Branch Id. Branch unique identification |
| IFHOME | varchar2(50) | Is the branch in the same province? 1 for the same province 0 for different provinces |
| B_REYEAR | date | The year of the branch set up |
| B_ENDYEAR | date | The year of the branch close. |

### F. Invest data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| BTEID | varchar2(50) | Invested enterprise ID |
| IFHOME | varchar2(50) | Is the invested enterprise in the same province? 1 for the same province 0 for different provinces |
| BTBL | number | Shareholding ratio |
| BTYEAR | varchar2(50) | The year of the invested enterprise set up |
| BTENDYEAR | varchar2(50) | The year of the invested enterprise set up |

### G. Right data.csv

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| RIGHTTYPE | varchar2(50) | The right type |
| TYPECODE | varchar2(100) | The id of the right |
| ASKDATE | date | Date of Application |
| FBDATE | date | Date of rights publish |

### H. Project data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| TYPECODE | varchar2(100) | Project ID |
| DJDATE | date | Date of winning bid |
| IFHOME | varchar2(50) | Is the project in the same province? 1 for the same province 0 for different provinces |

### I. Lawsuit data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| TYPECODE | varchar2(100) | Law case id |
| LAWDATE | date | Date of the case occurred |
| LAWAMOUNT | number | The amount of the case |

### J. Breakfaith data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| TYPECODE | varchar2(100) | Break faith id |
| FBDATE | date | Date of break faith begins |
| SXENDDATE | date | Date of break faith ends |

### K. Recruit data

| name | type | Description |
|---|---|---|
| EID | varchar2(50) | Enterprise ID, Enterprise unique identification |
| WZCODE | varchar2(50) | The code of Recruitment website |
| POSCODE | varchar2(200) | The code of Recruitment position |
| RECDATE | date | Date |
| PNUM | varchar2(200) | Number of recruits |