Optimal Feature Search to rank crucial factors that influence software development through Predictive Association mining

[1] Eileen Christina Mercy, [2] Dr.T.R.Sivapirya, [3] C. Malarvizhi

[1]QA & E-Learning Manager, [2]Associate Professor, [3]Assistant Professor
[1]Group IT,
[1]GAC Group, City, Singapore

Abstract : Accurately predicting software development effort has been a conundrum that has piqued the interest of researchers ever since computer systems became mainstream. Unlike an assembly line that churns out tangible products, where effort required can be predicted with relative ease, there are a plethora of factors that collectively exert influence on software development effort. There are situational and contextual factors which play a part in determining software development effort. Some of the factors that influence software development effort are industry domain, language used, and development method and resource capability. This study is aimed at improvising on previous research work of correlation of key factors, to add another dimension to accurately rank the key features that influence software development effort. Existing studies have applied correlation, case studies and statistical techniques. The proposed work is a new approach to mine the optimal associations and to rank the most influencing factors. Common associations have been mined from various software development domains to discover the underlying relationship. Feature search is implemented to bring out the dominating factors and has compared with the results of the statistical methods. Using the additional modifiers like clustering and association to arrive at software effort is proven to increase the accuracy of software estimation by more than ninety percent using the same dataset. The proposed method identified the top ranking associations that would enhance the software development process.

***IndexTerms* - Software Development Effort, Clustering, Association, industry domain**

I. INTRODUCTION

Software development is greatly influenced by requirement analysis and design phases irrespective of the domain for which it is being developed. Success of a software product relies on the factors that influence the development stage. Scope, cost and time management are crucial factors in determining the preparation and implementation of the project. Apart from the basic features that influence the software development, the proposed study analyses the kind of relationship that exists between various factors that are in-built in the design and development phase. One such feature is the effort undertaken in the design and implementation of the project. Effort, time and cost estimation play a major role in the continued building of the product. While time and cost are considered as complimentary, effort taken drives the relationship between these factors.

II. RELATED WORK – MACHINE LEARNING IN SOFTWARE DEVELOPMENT

According to Clarke and O'Connor [1], the situational factors that affect the software development process are the nature of the application under development, team size, requirements volatility and personnel experience. Nosheen Qamar et. al. [2] have identified web application size, productivity coefficients and nine different cost drivers - product reliability and complexity, platform difficulty, personnel capability, personnel experience, facilities, schedule, teamwork, process Efficiency and reuse that affect effort estimation.Hanchate and Bichkar[3] have illustrated commonly used Machine Learning techniques such as neural networks, case based reasoning, classification and regression trees, rule induction, genetic algorithm and genetic programming for planning and estimation. Some of the contextual factors affecting Software development are Operational Management, Competence, Company Infrastructure, Organisational Structure, Organizational Culture, Customer, Business Environment, Strategic Management and Knowledge Transfer by Bern et. al. [4]. Factors found to be significant for projecting software development effort are project size, average number of developers working on the project, type of development, Development language,development platform and the use of rapid application development. Jiang And Naude [5].Krishnamoorthy Srinivasan et. al.[6] used machine learning approaches to estimate the software development effort. Decision tree and neural network was used to estimate the effort of the software development. Decision and regression tree found to be the well suited model for estimating the effort. A key finding from Dejaeger, et. al.[7] is that by selecting a subset of highly predictive attributes such as project size, development, and environment related attributes, typically a significant increase in estimation accuracy can be obtained. Applying data mining to software effort estimation in the study conducted by Karna, Vicković and Gotovac[8] has proved that sound results can be gained through the use of data mining within the studied area and generally had a smaller effort estimation error. Machine Learning for Software Development Effort Estimation using Random Forests has been evaluated by Zakrani, Hain and Namir [9] to show that by varying the value of key parameters, the effort estimation model outperforms regression tree models.

III. METHODOLOGY

From a business point of view, accurate software estimation will facilitate business growth and development since IT is ubiquitous to business in the age of Internet 4.0. Companies spend millions on dollars on failed IT projects or projects that ran out of budget due to poor estimation, risk management and project management.

3.1. DATASET

The data were collected from different sources such as promise software engineering repository and zenodo. There were totally eight datasets collected in multiple domains such as Usp-05,China,Usp05-ft,Ebspm,Cocomonasa, Maxwell, Albercht and Nasanumeric. Each dataset contains the features related to software project development. Some features were common in all dataset. The dataset related to the software project management contains the features such as language, effort, cost, etc. Some datasets were related to project management such Nasanumeric. Finding the dominant rules common in all domains and in each domain was done through association mining.

3.2. ***PREPROCESSING AND CLUSTERING***

As the dataset was collected from different domains, it must be preprocessed by using appropriate filters according to the type of data. Some set of features contains numeric values. Two types of filters were used such as discretization and numeric to nominal. For some dataset, discretize was used and for the others numeric to nominal was used. Dataset with less number of features are mined directly with Apriori. Dataset like EBSPM with more than thirty features is clustered to find the similar features. Finding association between similar features may not yield the correct correlation. To avoid duplication of features in association, clustering is applied before association.Overall architecture of the proposed work is shown in Figure 1. Dataset is preprocessed with filters, followed by categorization or clustering.

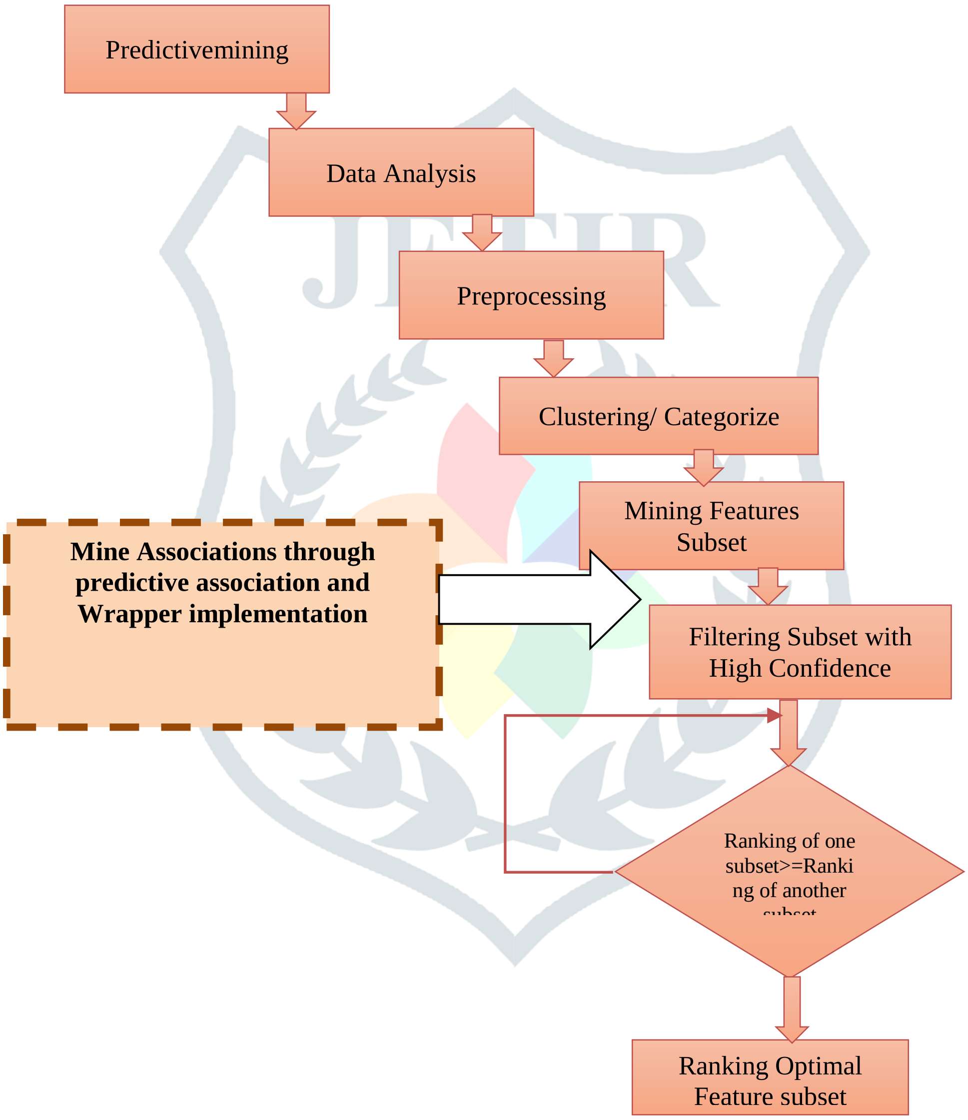

Figure 1. Architecture of Cluster based association mining

The best rules were mined from the dataset and it has been found that the predominant rules were found in the Usp-05 and Usp05-ft dataset. The EBSPM dataset contains the data from the telecommunication and Banking organization. K-Means clustering is used to find the clusters of EBSPM .Rapidminer is used to find the clusters. The clusters are formed according to functional size and Actual_cost_EUR. The resultant dataset is used to generate the frequent itemset in domainwise. The plot of the clustered instances is shown in figure 2.

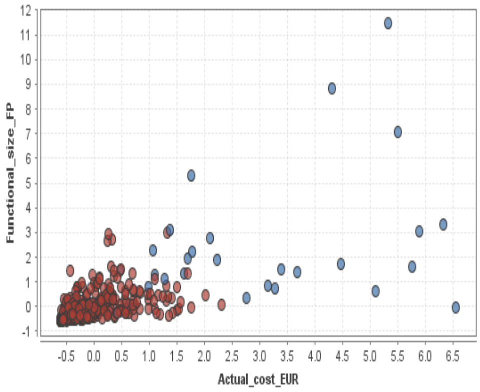

Figure 2: K-Means Clusters of EBSPM

3.3. ASSOCIATION MINING

To mine the association between different features of the dataset apriori and filtered apriori algorithms are implemented. The association was mined as itemset for each dataset and found the best ten rules which were ranked according to the confidence. The minimum confidence was taken as 90% and the support ranges from 10%(Lower Bound) to 100%(Upper Bound). The itemset were generated as one itemset, two itemsets, three itemsets, four itemset and five itemsets (listed in Table 2) that enables ranking of the subsets relevant to redesigning software development.

3.4. OPTIMAL FEATURE SEARCH

The item subsets obtained from the datasets are analysed to identify the predominant factor that greatly affects the core areas of software development. Time, scope and cost management in software development undergo a remarkable change when there is a change in the employee skill set, their experience, their contribution described in terms of effort taken. Many studies such as Menzies, Tim, et al., Finnie, G. R , et. al., have done effort estimation as a key research area. The proposed work digs the factors that are associated along with effort estimation. Hence an association study has been undertaken. After drawing out the relativity, the next step is to rank the features based on their confidence level in which they have contributed to the association. The high confidence itemsets in various domains are compared to identify the common factors that have contributed several times in each group. After detecting the most common occurring contributor, they are ranked based on their level of confidence and support. This is tested again with EBSPM dataset. The pseudocode of the optimal feature search is given below.

Optimal Factors Ranking(N, C, IS)
N- total number of Subset; C- Confidence; IS – Item subset ;
TI- Total number of items in a subset; SI- Selected predominant items

```
AR –set of association rules from each domain
SS- Strong associated subsets filtered
Begin
Step 1.  Identify the number of items in each subset
Step 2.  TI<- Number of items in subset
Step 3. For each domain
                    AR[i] ←Mine association rules for each domain
Step 4.  Filter itemsets from each domain with higher confidence
          For each itemsubset from 1 to N
            For each item in TI:
               If AR(Ci) >  threshold (Average confidence of all
items in a subset)
                    SS(Ci)<- Ci
            End loop
            End loop
Step 5. Prediction of influencing factors
        Compare top feature subsets across domains
            Generate the frequency FSS[i] of SS[i]
        Sort the resultant feature subsets based on frequency
        Rank the associations and categorize to each domain
return Predominant Items SI
End
```

IV RESULTS AND DISCUSSION

The frequent itemset for the dataset with two and three itemsets are drawn as shown in Table 1.

Table 1: Metrics for each dataset

Dataset	Rules with high confidence	Support, Confidence, Number of cycles
China	10	Minimum support: 0.4 (200 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 12
Usp-05	9	Minimum support: 0.25 (51 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 15
Usp-05 ft	8	Minimum support: 0.5 (38 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 10
Albrercht	10	Minimum support: 0.1 (2 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 18
Maxwell	3	Minimum support: 0.7 (43 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 6
Cocomonasa	3	Minimum support: 0.55 (33 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 9
Nasanumeric	9	Minimum support: 0.55 (51 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 9
Ebspm- Banking	10	Minimum support: 0.95 (314 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 1
Ebspm- Telecommunication	10	Minimum support: 0.95 (150 instances) Minimum metric <confidence>: 0.9 Number of cycles performed: 1

The number of itemset for China, Cocomo, Maxwell, Nasanumeric is 3, Albercht is 4, Usp-05 is 5 and Usp05-ft is 6. The number of rules to be generated for each dataset is set according to the number of instances available. Among the rules generated, the best rules with higher confidence were selected. In Banking domain, the following features are associated frequently such as Multi_application_release,Problems_with_external_supplier, Package_of_the_shelf, Pilot_or_proof_of_concept. In Telecommunications, the frequently associated features are Legacy_application,Phased_project,Package_with_customization,Pilot_or_proof_of_concept.

Table 2. Feature subsets retrieved through wrapper subset feature Selection

Dataset	Merit	Feature Subset
cocomo	0.92025	RELY, DATA, TIME,STOR, VIRT, TURN, VEXP, LEXP, MODP, TOOL,LOC
EBSPM	0.63821	Actual_duration_months ,Actual_cost_EUR, Actual_effort_hours, Defects_process, Defects_first_month, Stakeholder_satisfaction_process, Stakeholder Satisfaction - Product
MAXWELL	0.8465	Syear,Har,Dba,T02,T03,T07,T08, T09,T10,T11,T13,T14,Duration,Size,Time
Usp05	Subset 1: 0.507935	FunctPercent, IntComplx, Lang , Tools ,Method
	Subset 2: 0.57299	ID, Tools
USP05-FT	0.67333	ID, DataOut , UFP, ToolExpr,TeamSize
EBSPM BANKING	0.66222	Actual_effort_hours, Migration_project
CHINA	0.78233	ID, Output, Interface, Added,Duration, N_effort
ALBERCHT	0.68212	Output, Inquiry, RawFPcounts

In Cocomo dataset, process complexity, virtual machine experience, main memory constraint, use of software tools and Time are the predominant features for the software development as shown in Table 3(in Appendix). In Nasanumeric dataset, the key factors for the software development are mode, virtual machine experience, language experience, flight or ground system, virtual machine experience and process complexity. The key factors in Maxwell are Source, Database administration and IFC based construction product.

Table 2 shows the features subset mining for each dataset in multiple domains. As the Ebspm dataset contains multiple domains, the feature subsets are mined from multiple domains and from banking domain alone.

In Albertch, input, output and inquiry are the key factors. The common key factors on analyzing the associations of different dataset, it is found that if the project exhibits similar features, the predominant features are extracted in common. These key factors play a vital role for the project manager while managing the project.

The Usp05 dataset yields the two-feature subsets. The subset 1 contains five features and subset 2 contains 2 features. The remaining datasets yielded only one subset with higher confidence. The feature with high confidence in each feature subset was found and the features with low confidence were eliminated from the feature subset for all dataset. The feature with high confidence from each subset were compared and ranked. The similar features with high ranking from each subset from different domains are generated (Table 4 in Appendix). These optimum features aid the software project manager to efficiently manage the software development. The mined optimum features were Effort, Cost, Tools and Language used. The algorithm for the generation of optimal feature search is shown below.

Figure 3 shows the ranking top associations predicted from the software development projects. Effort associated with time tops the list. Language used for development is associated with tools and experience of the developer in the tool. Another notable association is between language and the method of design.

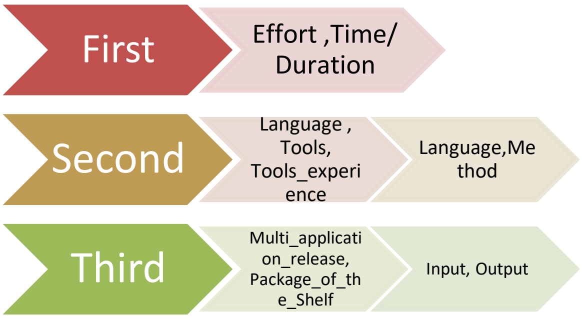

Figure 3. Top ranking associations

V CONCLUSION

As Project Management plays a vital role in organizations, it is necessary to focus on the key factors affecting projects. Our study has ranked the underlying set of generic factors applicable across different industry domains. We have also been able to isolate the specific factors affecting software development applicable to each domain. Since the generic and domain specific factors are available from our study, these rules can be applied on a variety of industry specific project datasets and generic software development datasets for further data mining and study.

This study can be used in the industry to prevent erroneous software estimation by considering factors highlighted specifically for each industry. The generic factors should also be considered in any development project to ensure that the effort estimate is accurate.

In retrospect, since the associations between the factors are listed, the project manager can easily access by which factor, the project has not been completed successfully. This can be extended by extracting the knowledge from the different patterns mined from the different data.

REFERENCES:

[1] Clarke, P., & O'Connor, R. V., 2012. Thesituatonal factors that affect the software development process: Towards a comprehensive reference framework. Information and Software Technology, 54(5), 433-447.
[2] Qamar, N, Batool, F, Zafar, K, 2018. Efficient effort estimation of web based projects using Neuroweb. International Journal of Advanced and Applied Sciences, 5(11), 33=39.
[3] Hanchate D, BIchkar R, 2015. The Machine Learning in Software Project Management – A journey II. Applied Discrete Mathematics and Heuristic Algorithms 1(3):21-47.
[4] Bern, A., Pasi, S. J. A., Nikula, U., &Smolander, K. , 2007. Contextual factors affecting the software development process–an initial view. In 2nd AIS SIGSAND European Symposium on Systems Analysis and Design. Gdansk, Poland, June (Vol. 5).
[5] Jiang, Z., &Naudé, P. , 2007. An examination of the factors influencing software development effort. International Journal of Computer Information and Systems Science and Engineering, 1(3), 182-191.
[6] Srinivasan, Krishnamoorthy, and Douglas Fisher.,1995. Machine learning approaches to estimating software development effort. IEEE Transactions on Software Engineering 21.2 : 126-137.
[7] Dejaeger, Karel & Verbeke, Wouter& Martens, David &Baesens, Bart.,2011. Data Mining Techniques for Software Effort Estimation: A Comparative Study. IEEE Transactions on Software Engineering, 38. 375-397. 10.1109/TSE.2011.55, 2011.
[8] Karna H, Vicković L, GotovacS. , 2019. Application of data mining methods for effort estimation of software projects. Softw: PractExper. 2019;49:171–191.
[9] Zakrani, Abdelali& Mustapha, Hain &Namir, Abdelwahed , 2018. Software Development Effort Estimation Using Random Forests: An Empirical Study and Evaluation. International Journal of Intelligent Engineering and Systems, 11. 300-311. 10.22266/ijies2018.1231.30.
[10] Menzies, T., Chen, Z., Hihn, J., & Lum, K. ,2006. Selecting best practices for effort estimation. IEEE Transactions on Software Engineering, 32(11), 883-895.
[11] Finnie, G. R., Wittig, G. E., &Desharnais, J. M. ,1997. A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models. Journal of systems and software, 39(3), 281-289.

APPENDIX 1:

Table 3.Frequentitemsets with attributes two to six

Dataset	Item(2)	Item(3)	Item(4)	Item(5)& Item(6)
Albercht	Size of set of large itemsetsL(2): 9 Large ItemsetsL(2): Input, Output Input,Inquiry Input, FPAdj	Size of set of large itemsetsL(3): 4 Large ItemsetsL(3): Input, Output, Inquiry Input, Output, FPAdj	Size of set of large itemsetsL(4): 1 Large ItemsetsL(4): Input, Output, Inquiry, FPAdj	
Usp-05	Size of set of large itemsetsL(2): 20 Large ItemsetsL(2): ObjType, AppType ObjType, DBMS ObjType, AppType DataOut,ToolExpr DataOut, DBMS DataOut, AppType	Size of set of large itemsetsL(3): 17 Large ItemsetsL(3): DataOut,ToolExpr, DBMS Lang, Tools, ToolExpr	Size of set of large itemsetsL(4): 9 Large ItemsetsL(4): Lang, Tools, ToolExpr, DBMS	Size of set of large itemsetsL(5): 2 Large ItemsetsL(5): Lang, Tools, ToolExpr, DBMS, Method
Usp05-ft	Size of set of large itemsetsL(2): 17 Large ItemsetsL(2): DataOut, AppType. UFP, AppType. Lang, Tools. Lang, ToolExpr.	Size of set of large itemsetsL(3): 20 Large ItemsetsL(3): Lang, Tools, ToolExpr. Lang, Tools, DBMS.	Size of set of large itemsetsL(4): 15 Large ItemsetsL(4): Lang, Tools, ToolExpr, DBMS.	Size of set of large itemsetsL(5): 6 Large ItemsetsL(5): Lang, Tools, ToolExpr, DBMS, Method Size of set of large itemsetsL(6): 1 Large ItemsetsL(6): Lang,Tools, ToolExpr, DBMS, Method, AppType.
Ebspm-Telecommunication	Size of set of large itemsetsL(2): 32 Large ItemsetsL(2): Phased_project, Migration_project Phased_project, Problems_with_external_supplier	Size of set of large itemsetsL(3): 51 Large ItemsetsL(3): Phased_project, Migration_project, Problems_with_external_supplier Phased_project, Migration_project, Security_related_project.	Size of set of large itemsetsL(4): 44 Large ItemsetsL(4): Phased_project, Migration_project, Problems_with_external_supplier, Package_with_customization. Phased_project, Migration_project, Problems_with_external_supplier, Legacy_application.	Size of set of large itemsetsL(5): 19 Large ItemsetsL(5): Phased_project, Migration_project, Problems_with_external_supplier, Package_with_customization, Legacy_application. Size of set of large itemsetsL(6): 3 Large ItemsetsL(6): Phased_project=0, Problems_with_external_supplier, Security_related_project, Package_with_customization, Legacy_application, Pilot_or_proof_of_concept.
Ebspm-Banking	Size of set of large itemsetsL(2): 13 Large ItemsetsL(2): Multi_application_release, Problems_with_external_supplier Multi_application_release, Security_related_project	Size of set of large itemsetsL(3): 9 Large ItemsetsL(3): Multi_application_release,Problems_with_external_supplier, Package_of_the_shelf Multi_application_release, Problems_with_external_supplier, Many_team_changes_unexperienced_team	Size of set of large itemsetsL(4): 2 Large ItemsetsL(4): Multi_application_release, Problems_with_external_supplier, Package_of_the_shelf, Many_team_changes_unexperienced_team	

Table 4. Top 5 Best Rules for each dataset

Dataset	Top 5 Best Rules
Albercht	1. Output=60 2 ==> Input=40 2 <conf:(1)> lift:(12) lev:(0.08) [1] conv:(1.83) 2. Input=40 2 ==> Output=60 2 <conf:(1)> lift:(12) lev:(0.08) [1] conv:(1.83) 3. Input=40 2 ==> Inquiry=20 2 <conf:(1)> lift:(8) lev:(0.07) [1] conv:(1.75) 4. Input=40 2 ==>FPAdj=1.15 2 <conf:(1)> lift:(8) lev:(0.07) [1] conv:(1.75) 5. Output=17 2 ==> File=5 2 <conf:(1)> lift:(4.8) lev:(0.07) [1] conv:(1.58)
Usp-05	1. ToolExpr=[1,10] 61 ==> DBMS=Oracle 61 <conf:(1)> lift:(1.95) lev:(0.15) [29] conv:(29.75) 2. Lang=Php,_Html,_Sql,_JavaScript 59 ==>ToolExpr=[1,10] 59 <conf:(1)> lift:(3.33) lev:(0.2) [41] conv:(41.27) 3. Lang=Php,_Html,_Sql,_JavaScript 59 ==> DBMS=Oracle 59 <conf:(1)> lift:(1.95) lev:(0.14) [28] conv:(28.77) 4. Method=3_Tier_Architecture 59 ==> Lang=Php,_Html,_Sql,_JavaScript 59 <conf:(1)> lift:(3.44) lev:(0.21) [41] conv:(41.85) 5. Lang=Php,_Html,_Sql,_JavaScript 59 ==> Method=3_Tier_Architecture 59 <conf:(1)> lift:(3.44) lev:(0.21) [41] conv:(41.85)
China	1. Deleted=0 435 ==>Dev.Type=0 435 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 2. Resource=1 346 ==>Dev.Type=0 346 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 3. Deleted=0 Resource=1 297 ==>Dev.Type=0 297 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 4. Interface=0 252 ==>Dev.Type=0 252 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 5. Changed=0 245 ==>Dev.Type=0 245 <conf:(1)> lift:(1) lev:(0) [0] conv:(0
cocomo	1. CPLX=High STOR=Nominal 34 ==> VEXP=Nominal 33 <conf:(0.97)> lift:(1.21) lev:(0.1) [5] conv:(3.4) 2. STOR=Nominal TOOL=Nominal 35 ==> TIME=Nominal 33 <conf:(0.94)> lift:(1.41) lev:(0.16) [9] conv:(3.89) 3. VEXP=Nominal TOOL=Nominal 35 ==> TIME=Nominal 33 <conf:(0.94)> lift:(1.41) lev:(0.16) [9] conv:(3.89) 4. STOR=Nominal 42 ==> VEXP=Nominal 39 <conf:(0.93)> lift:(1.16) lev:(0.09) [5] conv:(2.1) 5. TIME=Nominal 40 ==> STOR=Nominal 37 <conf:(0.93)> lift:(1.32) lev:(0.15) [9] conv:(3)
Maxwell	1. T05=3 43 ==>Ifc=2 43 <conf:(1)> lift:(1.07) lev:(0.04) [2] conv:(2.77) 2. Source=2 54 ==>Ifc=2 53 <conf:(0.98)> lift:(1.05) lev:(0.04) [2] conv:(1.74) 3. Dba=1 Source=2 51 ==>Ifc=2 50 <conf:(0.98)> lift:(1.05) lev:(0.04) [2] conv:(1.65) 4. Source=2 54 ==> Dba=1 51 <conf:(0.94)> lift:(1.01) lev:(0.01) [0] conv:(0.87) 5. Ifc=2 Source=2 53 ==> Dba=1 50 <conf:(0.94)> lift:(1.01) lev:(0.01) [0] conv:(0.85)
nasanumeric	1. mode=semidetached 69 ==>forg=g 69 <conf:(1)> lift:(1.16) lev:(0.1) [9] conv:(9.65) 2. virt=l lexp=h 55 ==>forg=g 55 <conf:(1)> lift:(1.16) lev:(0.08) [7] conv:(7.69) 3. time=n 54 ==>forg=g 54 <conf:(1)> lift:(1.16) lev:(0.08) [7] conv:(7.55) 4. vexp=n 53 ==>forg=g 53 <conf:(1)> lift:(1.16) lev:(0.08) [7] conv:(7.41) 5. mode=semidetached cplx=h 53 ==>forg=g 53 <conf:(1)> lift:(1.16) lev:(0.08) [7] conv:(7.41)
Ebspm - Telecommunication	1. Legacy_application=0 157 ==>Phased_project=0 157 <conf:(1)> lift:(1.01) lev:(0.01) [0] conv:(0.99) 2. Phased_project=0 157 ==>Legacy_application=0 157 <conf:(1)> lift:(1.01) lev:(0.01) [0] conv:(0.99) 3. Package_with_customization=0 156 ==>Phased_project=0 156 <conf:(1)> lift:(1.01) lev:(0.01) [0] conv:(0.99) 4. Pilot_or_proof_of_concept=0 156 ==>Phased_project=0 156 <conf:(1)> lift:(1.01) lev:(0.01) [0] conv:(0.99) 5. Package_with_customization=0 156 ==>Legacy_application=0 156 <conf:(1)> lift:(1.01) lev:(0.01) [0] conv:(0.99)
Ebspm-Banking	1. Package_of_the_shelf=0 331 ==>Multi_application_release=0 331 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 2. Multi_application_release=0 331 ==>Package_of_the_shelf=0 331 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 3. Problems_with_external_supplier=0 325 ==>Multi_application_release=0 325 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 4. Problems_with_external_supplier=0 325 ==>Package_of_the_shelf=0 325 <conf:(1)> lift:(1) lev:(0) [0] conv:(0) 5. Problems_with_external_supplier=0 Package_of_the_shelf=0 325 ==>Multi_application_release=0 325