Final Project
-
Data Engineering 2015/16

Students:
Temesgen Mehari
Oliver Philipps
Stefan Vikoler

Professors:
Javier Segovia Pérez
Ernestina Menasalvas Ruiz

Universidad Politécnica de Madrid
Facultad de Informática
Campus de Montegancedo
28660 Boadilla del Monte (Madrid)
Spain

Contents

Task I

Business understanding: business goals and datamining goals

I.1 Business objectives:

I.1.1 The Problem

In the last three years the company had to spend more than 20% of the effort investment for many projects with major faults in the system and we are not able to figure out the reasons for these faults. Fortunately we have a data set about our projects of the last years from all our developer teams in south Europe. The data set also includes information about modifications in the system for the bug fixing, the operating system and the developer team. Our boss is really concerned about the major faults and needs information to find a solution. Moreover some clients are not very satisfied with their buggy CMS.
The problem can occur during the development process, which might lead to a minor effect, or also after deployment of the software. This leads to extra ordinary cost in form of person hours per month in the support and development department and the management. The problem can be expressed in Euros for the extra ordinary costs, percentage of the additional costs compared to the costs of the original effort and the number of projects having a major fault.

I.1.2 Goals

The main business goal is to reduce or eliminate the costs for major faults and to increase the productivity. Furthermore we want to increase the quality of our products, which leads to a better customer satisfaction and a better reputation and is the base for future turnover.

I.2 Business success criteria

- Reduction of effort investment for fixing major faults [hours per month or Euros].
- Reduction of bugs reported by customers [%].

I.3 Determine data mining goals

By using logistic regression on the given dataset the goal should be to find a fitting model which describes the influence of different factors that lead to major faults in the software. Therefore the goals for the data model are:

- Find critical factors which lead to major faults to avoid them or reduce their impact.
- Find factors which lead to faultless software.

I.4 Data mining success criteria

- Find meaningful factors based on significant odd ratios of the model.
- R squared value should be higher than 80%.
- The success of the model can be measured if the resulting factors are leading to major faults respectively avoid major faults for future projects.

Task II

Data understanding

The used data for the modeling is in a XLS format and contains 300 instances of software projects from the last 3 years with 9 fields:

1. Project ID [nominal]
2. Country: of the development team [nominal]
3. FAULT: if it was found a major fault [YES,NO]
4. Effort: effort in person/months of the original development [numeric]
5. Module: affected by the changes to fix the errors [nominal]
6. Component: affected by the changes to fix the errors [C1,...,C7]
7. BackEnd: indicating if they were also affected by the changes to fix the errors [YES,NO]
8. FrontEnd: indicating if they were also affected by the changes to fix the errors [YES,NO]
9. OS: operating system related with the errors [Mac,Windows]

All the entries in the dataset are non-empty. 135 of the entries have a major fault (FAULT: YES), which are 45% of all entries and 165 entries have no major fault (FAULT: NO), which are 55% of all entries.

Task III

Data preparation

Some modifications and preparation on the data are required to get useful results by the logistic regression.

III.1 Non-significant data

The column Project ID is a continuous enumeration of the data and therefore not usable or not meaningful for our model.

III.2 Numerics

Logistic regression can not be used on boolean or nominal values. Thus we have to modify all fields except the Effort field.

1. Country: We applied four dummy fields for the four countries (France, Italy, Spain, Portugal) with the boolean values [1,0].
2. Module: We applied four dummy fields for the modules Ads, DDBB, External Payment and Internal Payment with the boolean values [1,0].
3. Component: Also for the 7 components (C1,...,C7) we used 7 dummy fields with the boolean values [1,0].
4. BackEnd: We changed the YES and NO values to 1 and 0.
5. FrontEnd: We changed the YES and NO values to 1 and 0.
6. OS: We applied two dummy fields for the two operating systems Windows and Mac with the boolean values [1,0].

7. FAULT: We changed the YES and NO values to true and false of type boolean

III.3 Training and testing data set

Due to the cross validation in WEKA we do not have to modify the given data set to get training and testing data set.

III.4 Balancing

The given data set is already pretty balanced, 45% and 55%. Therefore we do not have to balance the data.

III.5 Split data fields

We split the field Effort into three fields called lowEffort, mediumEffort and highEffort to also get boolean values [1,0]. The first field contained 1 if the effort for the project was from [0.00, 3.43), the second field from [0.43, 5.06) and the third field from [5.06, 6.70]. We chose these values to get three balanced groups.

Task IV

Modeling

For our model with the prepared data we used the tool WEKA to perform the classification algorithm Logistic Regression with cross validation, which is a class for building and using a multinomial logistic regression model with a ridge estimator. The probability is calculated as follows:

$$\frac{1}{1 + \sum_{j=1}^{k-1} e^{X_i \cdot B_j}}$$

Task V

Evaluation

At the early modeling phase we applied Logistic Regression with cross validation with 10 folds on the whole prepared data.

V.1 Summary

After the first round of our Logistic Regression we already get satisfying results. The model classifies 73.00% of 300 instances correctly and also the root relative squared error is with 88.67% above our data mining goal of 80.00%. In the next cycle of the CRISP-DM we would recommend a stepwise backward regression to reduce the complexity of the model and increase the quality of the model.

Correctly Classified Instances	219	73%
Incorrectly Classified Instances	81	27%
Kappa statistic	0.4497	
Mean absolute error	0.3506	
Root mean squared error	0.4412	
Relative absolute error	70.813%	
Root relative squared error	88.6765%	
Total Number of Instances	300	

Table V.1: Stratified cross-validation summary

V.2 Confusion Matrix

This table[V.2] shows the correctly and incorrectly classified instances. The goal for further work will be to reduce the wrong classified projects, especially

the 34 projects which were classified as false, although they are true. Because we cannot react to a false classified project, which will have a major fault in the end. On the other hand, if we pay more attention to an incorrectly as true classified project, which in the end doesn't have a major fault, we probably do not have to pay more than 20% of effort investment.

a	b	← classified as
88	47	a = true
34	131	b = false

Table V.2: Confusion matrix

V.3 Evaluation of Odds Ratios

V.3.1 Modules and Components

The meaning of this table[V.3] is, effort and bug fixing for errors in these modules and components didn't lead to a major fault. This is especially the case for the variables ExternalPayment and C1.

Variable	**odds ratio**
ExternalPayment	0.1881
C1	0.2531
DDBB	0.5606
InternalPayment	0.5842
C2	0.592
C3	0.6136

Table V.3: Variables and odds decrease the probability for a major fault

The meaning of this table[V.4] is, effort and bug fixing for errors in these modules and components lead to a major fault. This is especially the case for the variables Ads, C7 and C6.

Variable	odds ratio
Ads	12.3564
C7	6.4682
C6	4.3078
C4	2.3418
FrontEnd	1.759
C5	1.2061

Table V.4: variables and odds increase the probability for a major fault

V.3.2 Countries

The development in Italy decreases the probability of major faults, whereas the development in Protugal increases the probability. The development in Spain and France have no high influence.

Variable	odds ratio
Italy	0.7214
Spain	1.0057
France	1.0856
Portugal	1.5764

Table V.5: variables and odds for countries

V.3.3 Operating Systems

The influence of the operating system on major faults is not very significant. Although there is a tendency that Mac increases the probability for major faults, in contrary to Windows, which decreases the probability.

Variable	odds ratio
Windows	0.7729
Mac	1.2939

Table V.6: variables and odds for operating systems

V.3.4 Effort

The table[V.7] shows that projects with lower efforts tend to decrease the probability whereas more complex and projects with higher effort increase

the probability of a major fault. The influence of medium sized projects is not very important.

Variable	**odds ratio**
LowEffort	0.8398
MediumEffort	0.915
HighEffort	1.3115

Table V.7: variables and odds for original development effort

Task VI

Plan deployment

Due to our evaluation we group our plan deployment in following four sections:

1. Modules and Components
2. Countries
3. Operating System
4. Effort

VI.1 Modules and Components

Our model explains us, that the module Ads and the components C7 and C6 are high likely to increase the probability for a major fault. Therefore in the planning phase of future projects we should if possible try to avoid or substitute these factors. Another way is to include internal or external experts during the development process for the problematic modules and components to our project team. In general it would be helpful to offer more trainings for these problems to our employees. Also we should analyse and refactor the concerning implementations in our continuous improvement process. Another possibility is to try to substitute these modules and components by different products or providers.

VI.2 Countries

The model showed that most of our developer teams in the countries do not influence the amount of major faults. But it should be a general aim to

improve our developer teams. For that reason we should offer more training classes. This seems to be the most important for our teams in Portugal. Maybe the experts in Italy can support the teams of other countries in complex projects. That teams can benefit from the Italian knowledge.

VI.3 Operating System

In this section it is not mandatory to introduce actions and processes, because the operating systems do not cause much trouble to our software. Although it would not harm to analyse and refactor our implementations for MAC and enhance our knowledge through trainings and experts.

VI.4 Effort

More complex projects should be split to more smaller projects to reduce the probability of major faults. This could be realized in planning phase of the project.

VI.5 Measurement

For monitoring it is necessary to frequently evaluate new projects to check if these variables are still critical factors for major incidents and evaluate the success of our actions. This can happen on a yearly base. The success can be measured by the new amount in percentage of our major faults. Furthermore it can be evaluated by the occurring support events, customer satisfaction, sold products and the gained new customers. Therefore the company should earn more money and reduce the bug fixing costs.