CS212
Final Project

Team Members: Bo Feng
Yifan Liu
Jiaming Li
Chen Qin
Jialun Cao
Mengyu Ji
Qihang Guan
Bo Zhang

0. Hypotheses

As for the data association in Git, we select four related parameters: the number of code submission times of the programmer; The average length of comments submitted by the programmer; How long the programmer has been working in the kernel (from the time the code was originally committed to the time it was last committed);The number of fixes the programmer has.

- **Hypothesis 1:** *The more code is committed, the more fixes the programmer makes to the kernel.*

The number of program submissions is directly related to the programmer's contribution to the kernel. So we have a good reason to believe that the number of code commits is highly correlated with the number of fixes it has.

- **Hypothesis 2:** *The higher the average length of the comment, the more times the programmer fixes the kernel.*

Comments are a brief introduction to code after it has been committed on Git. In general, the longer the comment length, the more complex the submitted code, the more functional and feasible the code, and the higher the probability of a fix for the kernel. Therefore, the average length of the annotation is also a dimension that can be used to determine the number of fixes.

- **Hypothesis 3:** *The longer the programmer works in the kernel, the more fixes the programmer makes to the kernel.*

As a rule of thumb, the longer you stay in a job, the more you know about it. As a programmer, the longer you work in the kernel, the more you understand how the kernel works, the more you understand the problems that the kernel is prone to, and the easier it is to fix the kernel. Thus, a programmer's time at the kernel is also a measure of his or her fix to the kernel.

1. Requirement

1.1. Data source:

All data of this project are from the data about Linux kernel on Git crawled by the crawler. We used the method of random sampling to select several versions from many versions, and then carried out sampling. First, 2000 data were selected for model construction. This data includes the author's name, the number of commits, the author's development time, and the commit details. The author's development time is extracted and subtracted by a specialized crawler to obtain the specific working days.

1.2. Data filtering and cleaning:

There are many redundant and wrong data, as well as duplicate and scrambled data in the crawled data. This will filter out the data and get clean, pure data.

1.3. Data processing:

We classify programmers by name. Take a programmer and count the number of times he has committed code, the average length of comments submitted code, the amount of time worked in the kernel, and the total number of fixes in the detail record (retrieved by tag keyword).Repeat multiple times to get different sets of data. Thus, each column of data in the sample includes: [programmer name, number of times the code was submitted, average length of comments submitted to the code, time worked in the kernel, total number of fixes.

1.4. Establishment of regression model:

The sorted data are grouped to establish a 3-yuan regression model. The regression model is established by taking the total fix number of each programmer as the dependent variable, the number of times the programmer commits code, the average length of comments committed code, and the working time in the kernel as the dependent variable.

1.5. Establishment of decision tree:

After the regression model is established, a decision tree is also established to test and improve the overall prediction effect. We first calculated the information entropy of each independent variable, arranged from small to large, determined each parent node and leaf node of the decision tree, and finally established a three-layer decision tree.

1.6. Repeat operation to improve the regression model:

After the establishment of the original model, we re-selected different versions in Git, expanded the data to 20,000, 40,000 and 100,000, and screened and cleaned them one by one. Finally, each group of data is introduced into the regression model of the previous version in turn. When errors are found, the regression model is optimized and improved. Eventually, the error was limited to 5.4%. At the same time, the information is put into the decision tree for analysis and comparison, and the corresponding results are obtained.

1.7. Result Analysis:

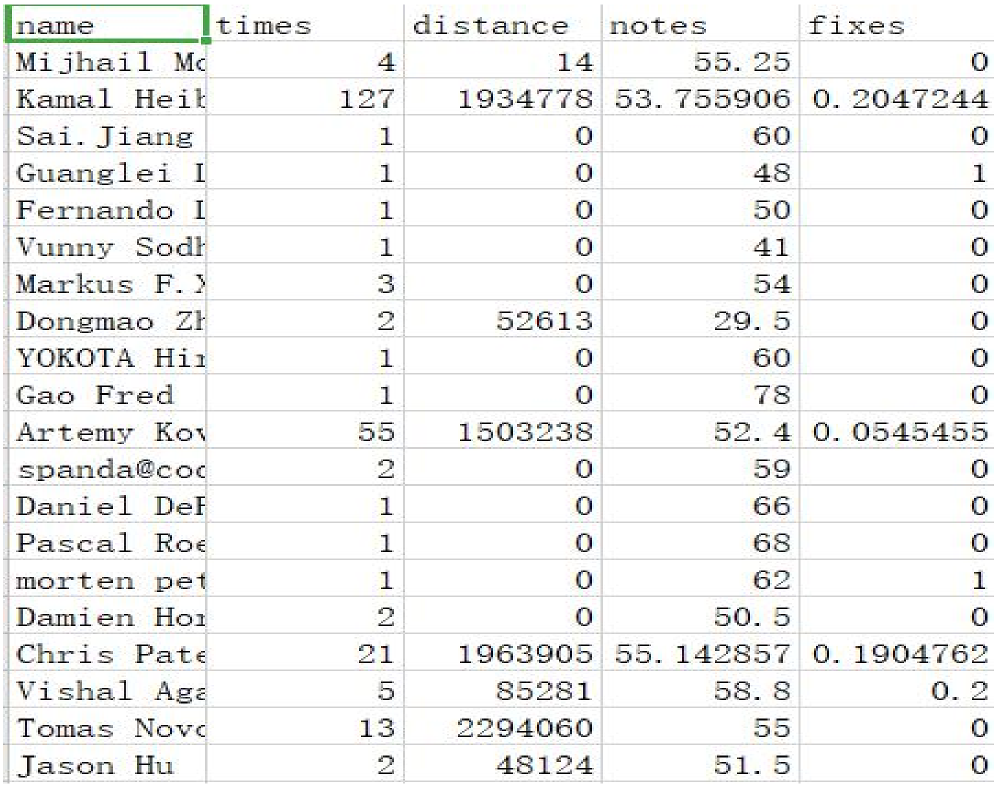
Two groups of analysis data were finally obtained: (1) The more accurate FIX value predicted by the regression model. (2)The length of the programmer's FIX predicted by the decision tree. Finally, we first put the programmers who need to be analyzed into the decision tree to analyze the most basic fix length, and then put it into the regression model to draw a more specific conclusion.

2. Code structure analysis

There are three parts: Data collect, Data process, Data analysis. There are four ”.py” files: data_collect.py, data_process.py, decision_tree.py, regression.py. data_process.py needs to use result.csv made by data_collect.py.

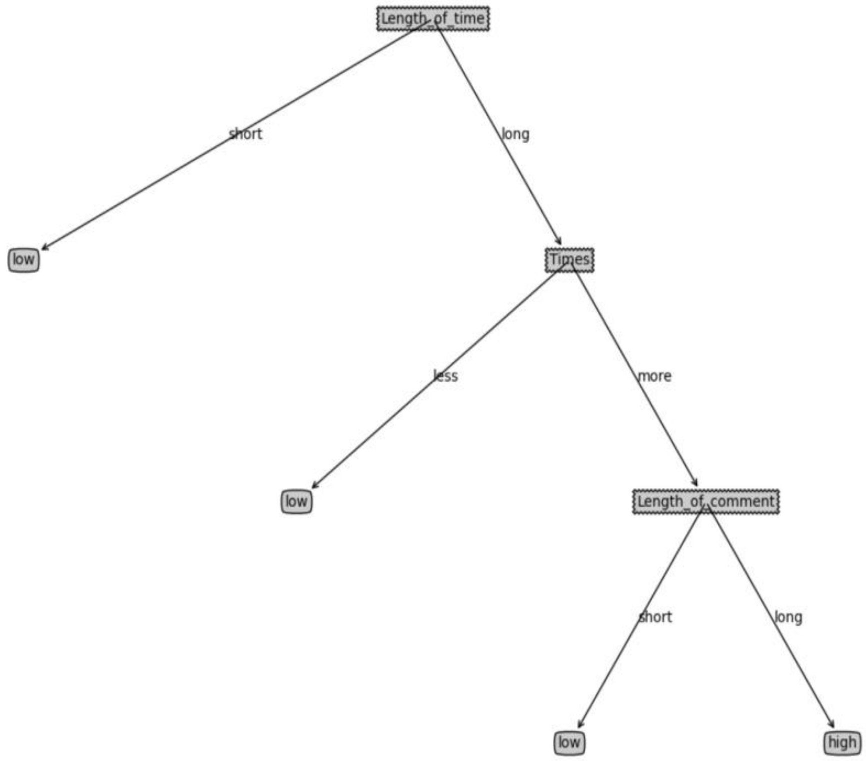
3. Result display

Data collection(about 2,000 rows):



name	times	distance	notes	fixes
Mijhail Mc	4	14	55.25	0
Kamal Heik	127	1934778	53.755906	0.2047244
Sai.Jiang	1	0	60	0
Guanglei I	1	0	48	1
Fernando I	1	0	50	0
Vunny Sodh	1	0	41	0
Markus F.X	3	0	54	0
Dongmao Zh	2	52613	29.5	0
YOKOTA Hir	1	0	60	0
Gao Fred	1	0	78	0
Artemy Kov	55	1503238	52.4	0.0545455
spanda@coc	2	0	59	0
Daniel DeF	1	0	66	0
Pascal Roe	1	0	68	0
morten pet	1	0	62	1
Damien Hor	2	0	50.5	0
Chris Pate	21	1963905	55.142857	0.1904762
Vishal Aga	5	85281	58.8	0.2
Tomas Novc	13	2294060	55	0
Jason Hu	2	48124	51.5	0

Desicion tree:



4. Conclusion the argument

After collecting the Linux kernel data on the git, cleaning filter, modeling analysis and a series of operations, finally draw the conclusion: the number of a developer committing, the average length of code comments, the kernel of working length are “little” positively correlated with fix quantity of kernel. However, after calculating the regression, it is not good, R^2 is so small even less than 0.01.

In our opinion, ”dirty data“ is still in data set but we have no idea or good standard to clean it. The relationship between data may not the linear but we haven’t studied the mathematical knowledge about the non-linear regression.