Quality Assurance Manual

Feedback Analysis Based Product Rating System

Contents

Preface

1) Writing purpose

This manual is a summary of test of **Feedback Analysis Based Product Rating System**, including bugs found and solved, work results, and quality assessment of the current system.

2) Target groups

For developers and testers, this manual is a presentation of work results. The evaluation of whether the execution is consistent with the plan could provide reference for future planning and the analysis of defects can provide suggestions for fixing and preventing bugs. Overall, through the analysis of the test results, we can get the evaluation of software quality and provide experience for the subsequent improvement.

For product managers, this manual can help them clearly understand each stage of product development of completion of the project, the test scope, process and quality situation as well as the risk, the legacy of existing test conclusions, and serves as a reference of a product release launch criteria.

For users, they can clearly understand the current product quality indicators, check whether the product deviates from the requirements and provide suggestions for the optimization of the product.

Test Plan

1) Personnel

The whole develop team and some students in school

2) Equipment

A computer with browsers

3) Progress

Test stage	Test task
unit test	crawler test NLP test outlier detection test UI test
system test	functional test compatibility test
validation test	user acceptance test

Test Result

crawler test result

1. Functional Testing

   The crawler part mainly tested the correctness of the crawler comment regular matching and whether the whole crawler ran smoothly without being blocked. After the completion of the crawler code from the back end, through the regular matching accuracy test at regular intervals (about one month), and modified the code if the test failed and tested again, the back-end crawler part passed the functional test smoothly.

Test Case	Input	Expected Output	Test Output	Result	Description	Importance
Comment number regular match detection	“10,257 total ratings | 2,548 with reviews”	2548	257 (2022.3.13) 2548 (2022.3.13)	- Fail (2022.3.13) -Pass (2022.3.13)	Change the regular matching rule from **"\d+"** to **"\.*[0-9]"** to match correctly	****
Comment number regular match detection	"10,257 total ratings , 2,548 with reviews"	2548	0 (2022.3.25) 2548 (2022.3.25)	- Fail (2022.3.25) -Pass (2022.3.25)	Change the regular matching rule from **"\.*[0-9]"** to **"\s.*[0-9]"** to match correctly	****
Test with 0 comments	URL	0	0 (2022.4.1)	- Pass (2022.4.1)	-	***
Whole crawler test	URL	result_queue	result_queue (2022.4.1)	-Pass (2022.4.1)	-	*****

2. Performance Testing

   The crawler speed was tested by changing the number of threads used by the crawler, and the appropriate number of threads was found within a certain range (the number of threads cannot be increased blindly, because the more threads means that the program is more likely to be blocked by the Anti-crawler mechanism of Amazon) to make the crawler speed as fast as possible. The test results show that the system is less likely to block when the selected thread is 20, and the speed is relatively fastest.

Test Case	Input	Expected Output	Test Output	Result	Description	Importance
Crawler speed performance test	thread _number = 10	Time output as fast as possible	13.194883108139038 s	- Pass (2022.4.2)	-	***
Crawler speed performance test	thread_number = 15	Time output as fast as possible	9.413830995559692 s	- Pass (2022.4.2)	-	***
Crawler speed performance test	thread_number = 20	Time output as fast as possible	5.753036022186279 s	- Pass (2022.4.2)	The crawler speed performance of 20 threads is best with the smallest possibility of crawler blocking	***

NLP test result

1. **Functional Testing**

   We separated the function testing of NLP into three sections: the tokenization section, the polarity valence section and the punctuation emphasis section.

Tokenization Testing

Test ID	Test Case	Input	Expected Output	Actual Output	Result	Note
1	Convert emojis to their textual descriptions	"Generally good, but a little expensive. 😂"	[“Generally good, but a little expensive. face with tears of joy”]	[“Generally good, but a little expensive. face with tears of joy”]	Pass	an emoji matches a textual description in the emoji lexicon
2	Preserve abbreviation	"It's good."	["It's", 'good']	["It's", 'good']	Pass	the quotation mark between letters used for abbreviations will be retained
3	Preserve dollar sign	"The book is cheap. It only costs $3.20."	['The', 'book', 'is', 'cheap', 'It', 'only', 'costs', '3.20']	['The', 'book', 'is', 'cheap', 'It', 'only', 'costs', '3.20']	Fail	"$" will be removed
	Preserve dollar sign	"The book is cheap. It only costs $3.20."	['The', 'book', 'is', 'cheap', 'It', 'only', 'costs', '$:', '3.20']	['The', 'book', 'is', 'cheap', 'It', 'only', 'costs', '$:', '3.20']	Pass	"$" would be treated as a special sign and will be added ": "after it, which world make it to be preserved as a whole token '$:'
4	Leaves contractions and most emoticons	["quite satisfied :)"]	['quite', 'satisfied', ':)']	['quite', 'satisfied', ':)']	Pass	if the length of the stripped string is smaller than two, then it was likely to be emoticon, so we just keep it as the original token. ':)' after strip operation is “, and len('')=0, which is less than 2, so we just keep the initial token ':)'.

Test ID	Test Case	Input	Expected Output	Actual Output	Result	Note
5	Preserve "no" without "."	"I would say no."	['I', 'would', 'say', 'no']	['I', 'would', 'say', 'no.']	Fail	"no." will be preserved as a emoticon, which is not the real case
		"I would say no."	['I', 'would', 'say', 'no']	['I', 'would', 'say', 'no']	Pass	"no" would be preserved as a negative word

Polarity Valence Testing

Test ID	Test Case	Input	Expected valence	Actual valence	Sentiment score	Test Result	Note
1	check positive word	"Our cats thrive well on this dry cat food."	[0, 0, 0, 1.1, 0, 0, 0, 0, 0]	[0, 0, 0, 1.1, 0, 0, 0, 0, 0]	4.35	Pass	the word "well" matches the positive word in the lexicon
2	check increment booster word	"Our cats thrive extremely well on this dry cat food."	[0, 0, 0, 0, 1.1+B_INCR, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 1.393, 0, 0, 0, 0, 0]	4.53	Pass	the booster word "extremely" add B_INCR (= 0.293) to "well"
3	check decrement booster word	"Our cats thrive slightly well on this dry cat food."	[0, 0, 0, 0, 1.1+B_DECR, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0.807, 0, 0, 0, 0, 0]	4.07	Pass	the booster word "slightly" add B_DECR (= -0.293) to "well"
4	check negative word	"awful"	[-2.0]	[-2.0]	0.26	Pass	the word "awful" matches the negative word in the lexicon
5	check decrement booster word	"It's kindof awful"	[0, 0, -2.0-B_DECR]	[0, 0, -1.707]	0.34	Pass	"kindof" will decrease the valence value of "awful"
6	check "kind-of"	"It's kind-of awful"	[0, 0, -2.0-B_DECR]	[0, 0, -1.707]	0.34	Pass	"kind-of" will decrease the valence value of "awful"
7	check booster word contain space	"It's kind of awful"	[0, 0, -2.0-B_DECR]	[0, 0, -1.707]	0.34	Pass	"kind of" will decrease the valence value of "awful"
8	check for "not"	"bad."	[-2.5]	[-2.5]	0.18	Pass	
		"not bad."	[0, -2.5*N_SCALAR]	[0, 1.85]	4.70	Pass	N_SCALAR(= -0.74)
9	check for "without ... doubt"	"good"	[1.9]	[1.9]	3.60	Pass	
		"without any doubt good"	[0, 0, -1.5*N_SCALAR, 1.9]	[0, 0, 1.11, 1.9]	4.87	Pass	the "doubt" would be negated
10	check for ALL CAPS for sentiment word	"This is cool."	[0, 0, 1.3]	[0, 0, 1.3]	4.48	Pass	
		"This is COOL."	[0, 0, 1.3+C_INCR]	[0, 0, 2.033]	4.74	Pass	the valence of "cool" will plus C_INCR(0.733)
11	check for ALL CAPS for booster word	"This is extremely cool."	[0, 0, 0, 1.593]	[0, 0, 0, 1.593]	4.62	Pass	
		"This is EXTREMELY cool."	[0, 0, 0, 2.326]	[0, 0, 0, 1.593+C_INCR]	4.80	Pass	the valence of "cool" after being incremented will then add C_INCR(0.733)
12	check for "no" as negation for an adjacent lexicon item	"no use."	[0.0, 1.9* N_SCALAR]	[0.0, -1.406]	0.46	Pass	no{}
		"no any use."	[-1.2, 0, 1.9* N_SCALAR]	[-1.2, 0, -1.406]	0.17	Pass	no ... {}

Test ID	Test Case	Input	Expected valence	Actual valence	Sentiment score	Test Result	Note
		"no use or value."	[0.0,1.9* N_SCALAR,0, 1.4* N_SCALAR]	[0.0, -1.406, 0, -1.036]	0.19	Pass	no...or{}/no..nor{}
13	"no" as its own stand-alone lexicon item	"I would say no to this product."	[0, 0, 0, -1.2, 0, 0]	[0, 0, 0, -1.2, 0, 0]	0.58	Pass	"no" would be preserved as a negative word
14	check "least"	"It is least good"	[0, 0, 0, 1.9* N_SCALAR]	[0, 0, 0, -1.406]	0.46	Pass	least
15	check special idioms	"the shit."	[0, -2.6]	[0, -2.6]	0.17	Fail	'the' and 'shit' would be separated, and only sentiment word 'shit' would be matched
		"the shit."	[3.0]	[3.0]	4.87	Pass	turn the phrase to be 'the-shit' as a whole idiom, which matches a positive word
16	check for “but”	"This was OK at first but soon went bad."	[0.0, 0, 1.93*0.5, 0, 0, 0, 0, 0, -2.5*1.5]	[0.0, 0, 0.97, 0, 0, 0, 0, 0, -3.75]	0.15	Pass	check for modification in sentiment due to contrastive conjunction 'but', and the sentiment in the left side of ‘but’ would be decreased, while the right side would be increase
17	add check for "only" as NEGATION	"Only problem cannot get my son away from it!"	[0, -1.7*N_SCALAR, 0, 0, 0, 0, 0, 0, 0]	[0, -1.7, 0, 0, 0, 0, 0, 0, 0]	0.27	Fail	"only" does not work
		"Only problem cannot get my son away from it!"	[0, -1.7*N_SCALAR, 0, 0, 0, 0, 0, 0, 0]	[0, 1.258, 0, 0, 0, 0, 0, 0, 0]	4.60	Pass	"only" acts as a negation word
18	Check "much prefer"	"I much prefer my 5.00 LuguLake mouse I got 5 years ago."	[0, -2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	2.50	Fail	"much prefer" has no sentiment effect
		"I much prefer my 5.00 LuguLake mouse I got 5 years ago."	[0, -2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, -2.0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	0.26	Pass	"much prefer" acts as a negative phrase

Punctuation Emphasis Testing

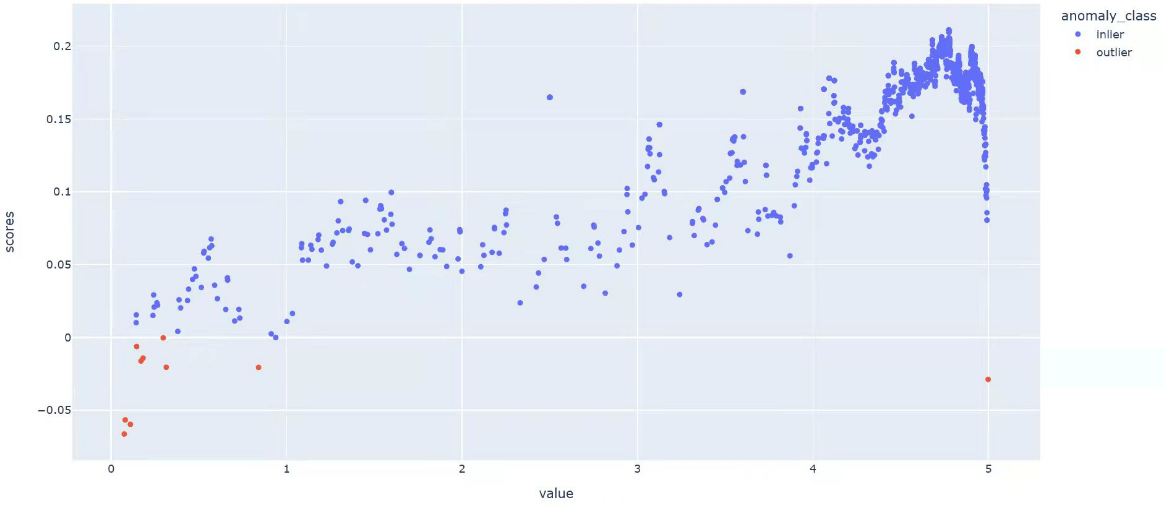
Test Case	Input	Expected sum_s	Actual sum_s	Score	Resul	Description
Check for exclamation point emphasis	"It is good."	1.9	1.9	4.71	Pass	
	"It is good!"	1.9+0.292	2.19	4.77	Pass	add one ep_amplifier
	"It is good!!"	1.9+0.292*2	2.48	4.82	Pass	add two ep_amplifier
	"It is good!!!!"	1.9+0.292*4	3.07	4.88	Pass	add four ep_amplifier
	"It is good!!!!!!!!"	1.9+0.292*4	3.07	4.88	Pass	up to 4 of exclamation points will be counted

2. **Performance Testing**

We tested the accuracy and efficiency of NLP depending on the amazon data set with 400,000 reviews from https://www.kaggle.com/datasets/kritanjalijain/amazon-reviews. In the data set, the first column is the emotional polarity of the comments in data set, with "1" representing negative and "2" representing positive. The third column is the content of the comment. If the predicted score was larger than 2.5, then we considered it to be a positive review. Otherwise, it was a negative review. By comparing the predicted tag with the actual tag in the data set, we got TP (True Positive), FP (False Positive) and count (number of all the reviews) to calculate the result to be 71.13%. And the efficiency is calculated to be 0.00137s for each comment.

outlier detection test result

We used isolation forest to detect the outliers. This is the scatter plot of one thousand comments generated by the isolation forest. Each point on the graph represents a comment, the horizontal coordinate represents the score (0 to 5) of the sentiment analysis for that point, and the vertical coordinate represents the confidence level of that result. Positive confidence represents the probability that the point is not an outlier; the higher the value, the more reliable the comment is. Negative confidence represents the probability that the point is an outlier, and the smaller the value, the less credible the comment is.



As presented in the graph, the more dense the dots are, the higher the credibility of the corresponding comment. The sparser the points, the easier it is to detect outliers.

UI test result

UI test is the assurance for providing users with corresponding access or browsing functions through the functions of the test object. In the testing, the software was tested from the perspective of image display, content display and human-computer interaction.

Test ID	Name	Environment	Process	Expected Results	Actual Results
1	Unified style	google chrome (version 99.0.4844.84)	Open the Amazon and enter the product page, browse products to check the style fitness.	Software information can be clearly displayed. The software style is not abrupt.	No information is weakened. The normal use process of shopping will not be affected by the Add-on.
2	Degree of color coordination	google chrome (version 99.0.4844.84)	Continuously using the add-on for a long time to check whether the color combination of the add-on is uncomfortable.	No discomfort raised by the add-on.	The user has no abnormal feeling.
3	Text correctness	google chrome (version 99.0.4844.84)	Go through the text contents of add-on(including product info window, compare list and guideline) to check the text.	Software information can be clearly displayed. The software style is not abrupt.	No wrong spelling or gramma mistake.
4	Text style	google chrome (version 99.0.4844.84)	Go through the text contents of add-on to check the style of text is unified.	The text size, color and font of the same module should be consistent.	Text of the Add-on follows a unified style.
5	Tips clearness	google chrome (version 99.0.4844.84)	Check whether the tips of the Add-on is clear (by both developer and users).	The tips can be clearly understand without further explaination.	Tips using icons or text are clear. The functions with no textual tips may be comfused before using.

Test ID	Name	Environment	Process	Expected Results	Actual Results
6	Image loading	google chrome (version 99.0.4844.84)	Compare the images shown in Add-on and in local place, check whether they are normally displayed.	Images all displayed with normal clarity, in the correct position.	All images are clear and in right position.
7	Button style	google chrome (version 99.0.4844.84)	Go through all buttons in Add-on, check their width, height, color and style.	All the properties should be unified for the buttons with same functions.	Meet the requirements.
8	Button interaction	google chrome (version 99.0.4844.84)	Click the buttons, hover on the buttons to check the interaction of the buttons.	The interaction of buttons can work normally.	All clickable items have mouseover animation and cursor style changes.
9	Page stretch	google chrome (version 99.0.4844.84)	Zoom in, zoom out, and change the display size of the page.	The Add-on should display all imformation dispite the size of the browser.	Zooming in and out of the browser will not affect the display of the Add-on window. The Add-on window dose not change with the page scale, which may cause the window too huge/small. "Locate product" function works well only when chrome size is not changed.

functional test result

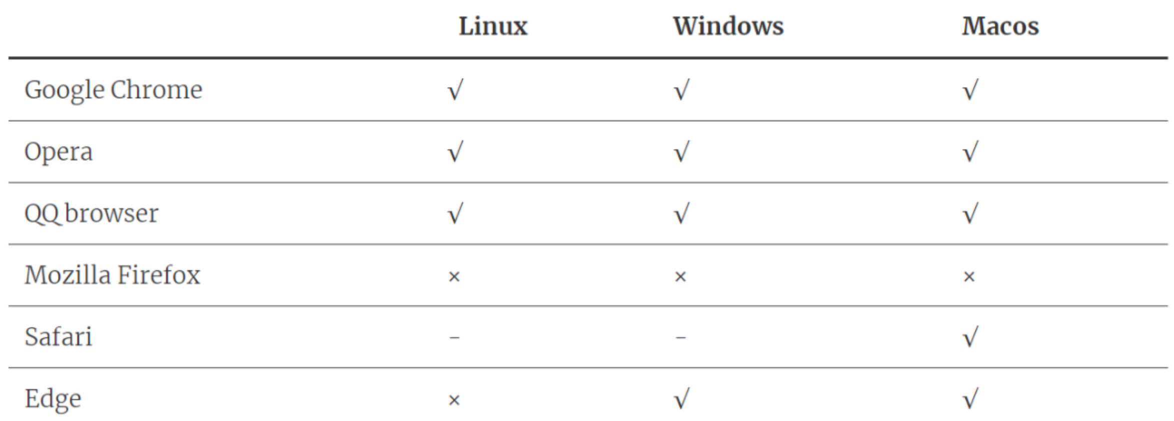
The functional test mainly focuses on what the software can do. It checks whether the software functions meet the requirements of the product requirements specification. In this testing, the program was installed, tested on Google chrome, running on the targeting website https://www.amazon.com/ with language English.

Test ID	Name	Environment	Process	Expected Results	Actual Results
1	Product window display	google chrome (version 99.0.4844.84)	Open the Amazon and enter the product page, move the curson above the products' name. Check the popup window and the information displayed on it.	The popup window shows. Information given in the popup window consistent with commodities.	Popup window displays. Informations shown are all correct. But the display time is slow for some products which have abundant comments.
2	Product Adding	google chrome (version 99.0.4844.84)	Click the add product button to check the function.	The compare list will show when the first product is added.	The function works well.
3	Compare list display	google chrome (version 99.0.4844.84)	Add all the products in webpage to the compare list, check whether their information is displayed correctly.	Product data in list is correct.	The data presented is all correct. For some products without price(no declared in page), 'N/A' is displayed.
4	Compare list sorting	google chrome (version 99.0.4844.84)	Click the compare list "Sort" button, check the sequence of the products. Change the quantity of products in the list and repeat the test.	After sorting, all products in list should rank according to the score from large to small.	The function works well.

Test ID	Name	Environment	Process	Expected Results	Actual Results
5	Compare list navigation	google chrome (version 99.0.4844.84)	Click on the product in compare list, check weather the site is relocated to the correct position.	After clicking, the site should be relocated to the correct position of the clicked product and gives a mark.	The function only works well when the browser is not resized. If the browser is resized, navigation will locate to the wrong position.
6	Software wizard display	google chrome (version 99.0.4844.84)	Check weather the wizard icon is displayed on the browser. Click the icon, check weather the guideline is completed.	Guideline is compelete and open/close, page changing functions work well.	The functions work well. Guideline is complete.
7	Software wizard moving	google chrome (version 99.0.4844.84)	Dragging the wizard icon, check weather it moves with the cursor.	The wizard icon can be dragged to anywhere in the browser.	The wizard icon can be dragged to anywhere in the browser. But it can also be dragged outside the size of site and can not be dragged back.

compatibility test result

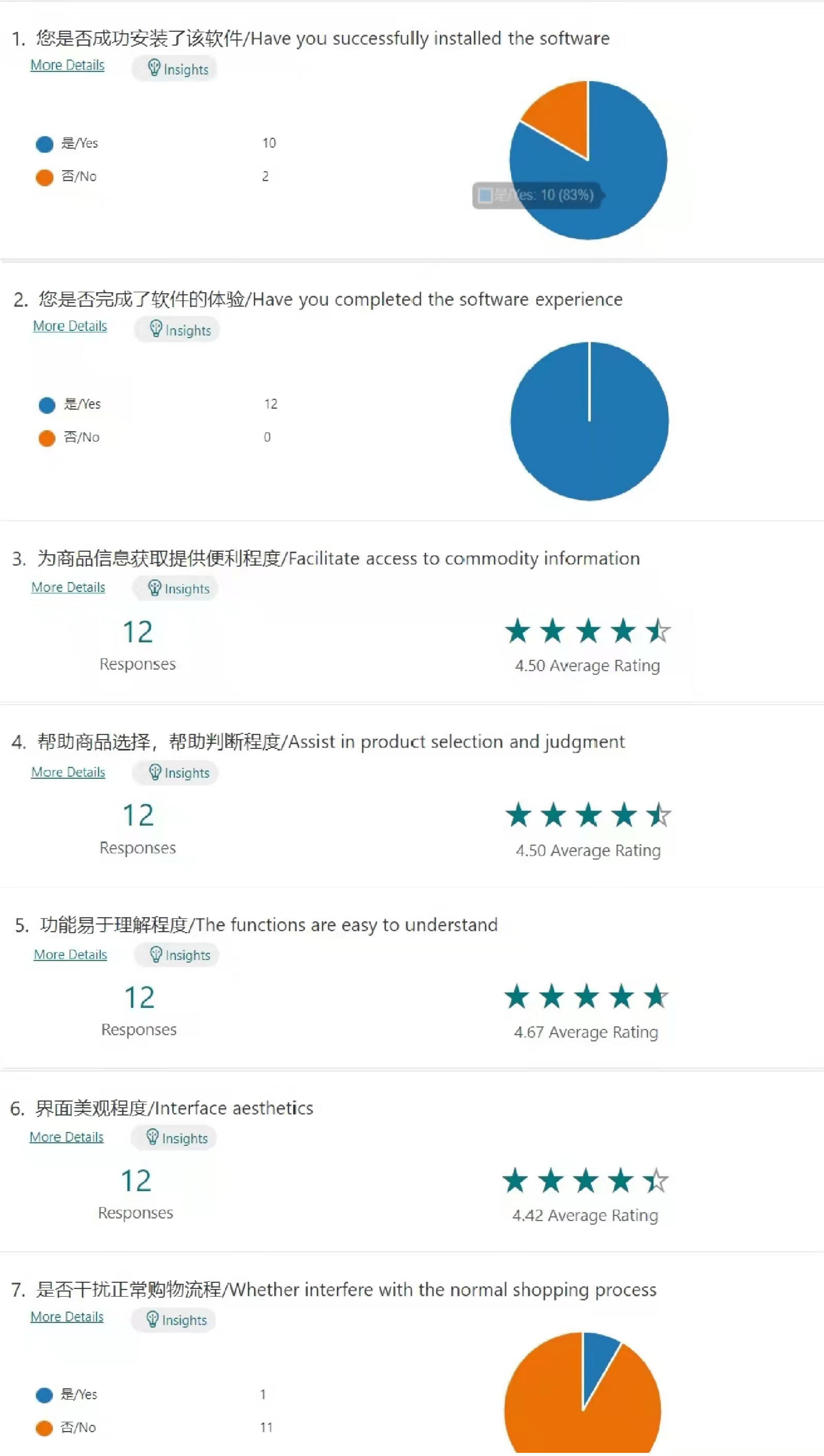
We tested the compatibility of the Chrome extension. The browsers supporting result for the add-on in Windows,Linux and Mac operating systems is shown below.



	Linux	Windows	Macos
Google Chrome	√	√	√
Opera	√	√	√
QQ browser	√	√	√
Mozilla Firefox	×	×	×
Safari	–	–	√
Edge	×	√	√

user acceptance test result

The system was tested by people selected randomly in the university. The questionnaire was given to the tester after they used the system. The result is shown below:



1. 您是否成功安装了该软件/Have you successfully installed the software

More Details　Insights

● 是/Yes	10
● 否/No	2

2. 您是否完成了软件的体验/Have you completed the software experience

More Details　Insights

● 是/Yes	12
● 否/No	0

3. 为商品信息获取提供便利程度/Facilitate access to commodity information

More Details　Insights

12
Responses

★★★★☆
4.50 Average Rating

4. 帮助商品选择，帮助判断程度/Assist in product selection and judgment

More Details　Insights

12
Responses

★★★★☆
4.50 Average Rating

5. 功能易于理解程度/The functions are easy to understand

More Details　Insights

12
Responses

★★★★★
4.67 Average Rating

6. 界面美观程度/Interface aesthetics

More Details　Insights

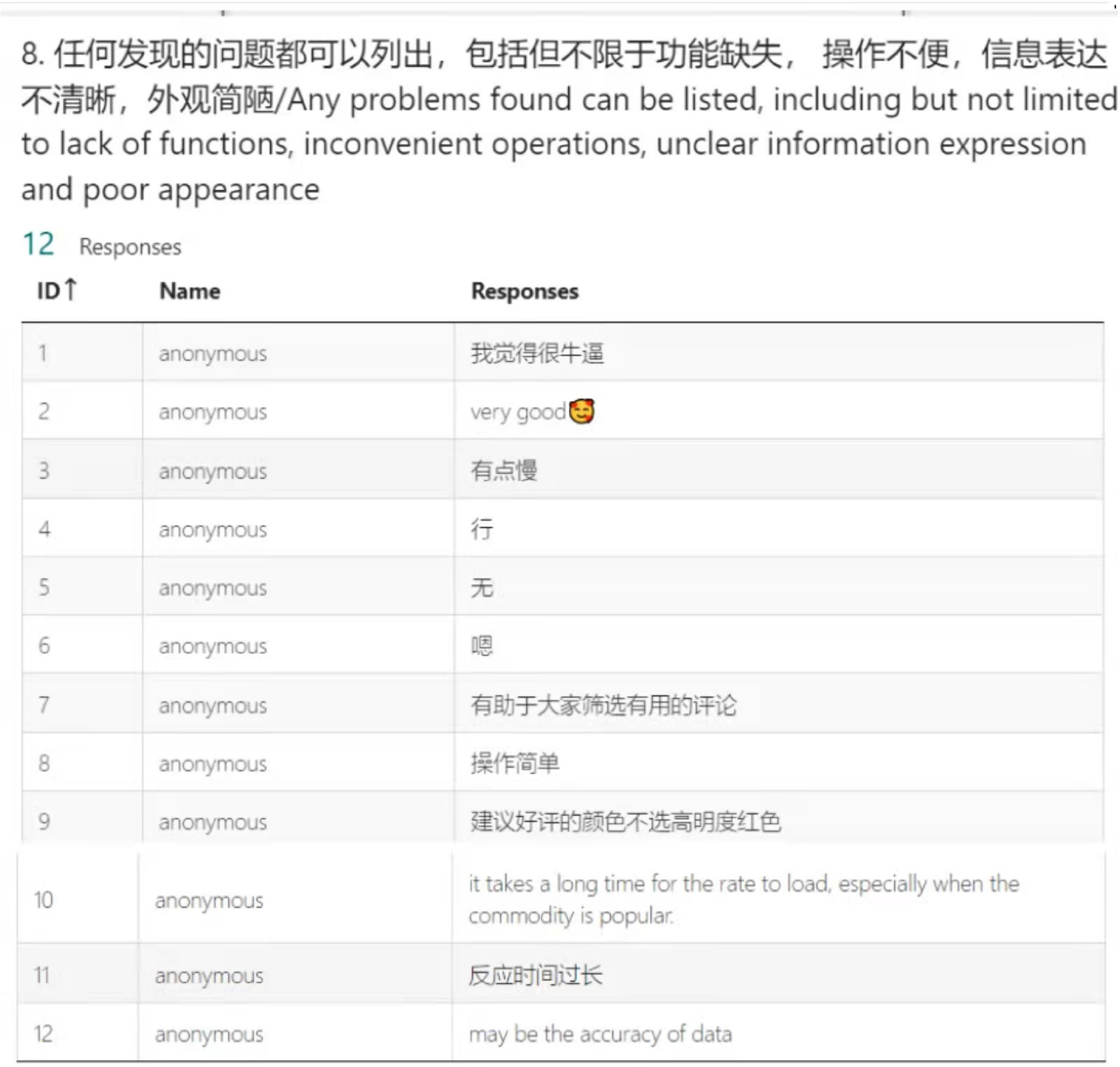
12
Responses

★★★★☆
4.42 Average Rating

7. 是否干扰正常购物流程/Whether interfere with the normal shopping process

More Details　Insights

● 是/Yes	1
● 否/No	11



8. 任何发现的问题都可以列出，包括但不限于功能缺失， 操作不便，信息表达不清晰，外观简陋/Any problems found can be listed, including but not limited to lack of functions, inconvenient operations, unclear information expression and poor appearance

12 Responses

ID↑	Name	Responses
1	anonymous	我觉得很牛逼
2	anonymous	very good🥰
3	anonymous	有点慢
4	anonymous	行
5	anonymous	无
6	anonymous	嗯
7	anonymous	有助于大家筛选有用的评论
8	anonymous	操作简单
9	anonymous	建议好评的颜色不选高明度红色
10	anonymous	it takes a long time for the rate to load, especially when the commodity is popular.
11	anonymous	反应时间过长
12	anonymous	may be the accuracy of data

Overall, the system provides good user experience according to the result of the questionnaire. Most users gave positive comments, along with some pertinent suggestions. After evaluating the feedback, the team modified part of the content in the system including the color of the pie chart after the testing.