1 Collecting all Permission within a cluster

At first I'm collecting all permissions that all apps of one cluster want. As example the cluster Flashlight:

Permissions

0	**4**	**9**	**10**	**11**	**12**

With the permissions:

ID	**Name**
0	In-App-Purchases
4	Calender
9	Pictures/Media/Files
10	Storage
11	Camera
12	Microphone

2 Collecting the average Permissions

In the next step I'm collecting the *average* permissions. To be exact: As *average* I mean the permissions that at least the half of apps in the cluster wants.

As example: The permissions In-App-Purchases, Storage and Camera

Permissions

0	4	9	**10**	**11**	12

3 Collecting apps and their permissions

Now I'm collecting all the apps of the cluster flashlight and their permissions. If an app has a permission it is marked as 1 (for example) else 0.

	Permissions					
Apps	ID	**0**	4	9	**10**	**11**	12
	14	1	0	0	1	1	1
	42	0	1	1	0	1	0
	145	1	0	0	1	1	0
	465	1	0	1	1	1	1
	1010	0	0	0	1	1	0

Now we have a matrix of every app with their permissions and every line stands for the *badness* of an app.

4 Reducing the badness of the average permissions

In fact, that an app needs some permissions to do whatever the app has to do, I'm now reducing the *badness* of the average permissions inside the cluster.

For example I halved the values:

Permissions

Apps: ID	**0**	4	9	**10**	**11**	12
14	0.5	0	0	0.5	0.5	1
42	0	1	1	0	0.5	0
145	0.5	0	0	0.5	0.5	0
465	0.5	0	1	0.5	0.5	1
1010	0	0	0	0.5	0.5	0

5 Calculating the summary of the badness

At the next step I'm calculating the summary of the *badness* of an app based on the permission it wants.

Permissions

Apps: ID	**0**	4	9	**10**	**11**	12	$\sum$
14	0.5	0	0	0.5	0.5	1	2.5
42	0	1	1	0	0.5	0	2.5
145	0.5	0	0	0.5	0.5	0	1.5
465	0.5	0	1	0.5	0.5	1	3.5
1010	0	0	0	0.5	0.5	0	1.0

Apps: ID	$\sum$
14	2.5
42	2.5
145	1.5
465	3.5
1010	1.0

6 Calculating the percantage of badness

Now I want to now how bad an app is compared with all others inside the same cluster.

At first I define the minimum value of badness as the minimum value that occures in the cluster. Here in this examle it is the value 1.0 of app *1010*

$$min = 1.0 \tag{1}$$

To define the maximum value I'm using the summary of the values of all permissions inside the cluster.

$$\begin{aligned} max &= \sum_{permission} (value_{permission}) \\ &= 6 \end{aligned} \tag{2}$$

Why not the maximum summary of the badness (app 465 with a badness of 3.5)? This would result in some strange results. As example: We have a cluster of round about 20 apps. 19 apps are from the same developer and needs the same permissions. Only one app is from an other developer and needs one permission less. This results in one app marked as green and 19 as red, but this 19 apps just have the *average* permissions (the permissions wich they need to work) and schould marked as yellow.

In the next step I'm calculating the percentage between *min* and *max* for each app. To make the coloring easier in our listing of all apps in a cluster, I'm inverting the value to get 100% as green and 0% as red.

$$p_{app} = 100 - \frac{((sum_{app} - min) * 100.0)}{(max - min)} \tag{3}$$

Apps

ID	p
14	70.0
42	70.0
145	90.0
465	50.0
1010	100.0

The flashlight app *1010* is our *nicest* app in this cluster and app *465* our *worst*.

7 Further improvements

As a further improvement I could replace the start value of 1 for each permission with a value that is based of how *bad* an permission is. For examle: a value between 0 and 1

Permissions

0	4	9	10	11	12
0.1	0.8	0.5	0.7	0.9	0.9