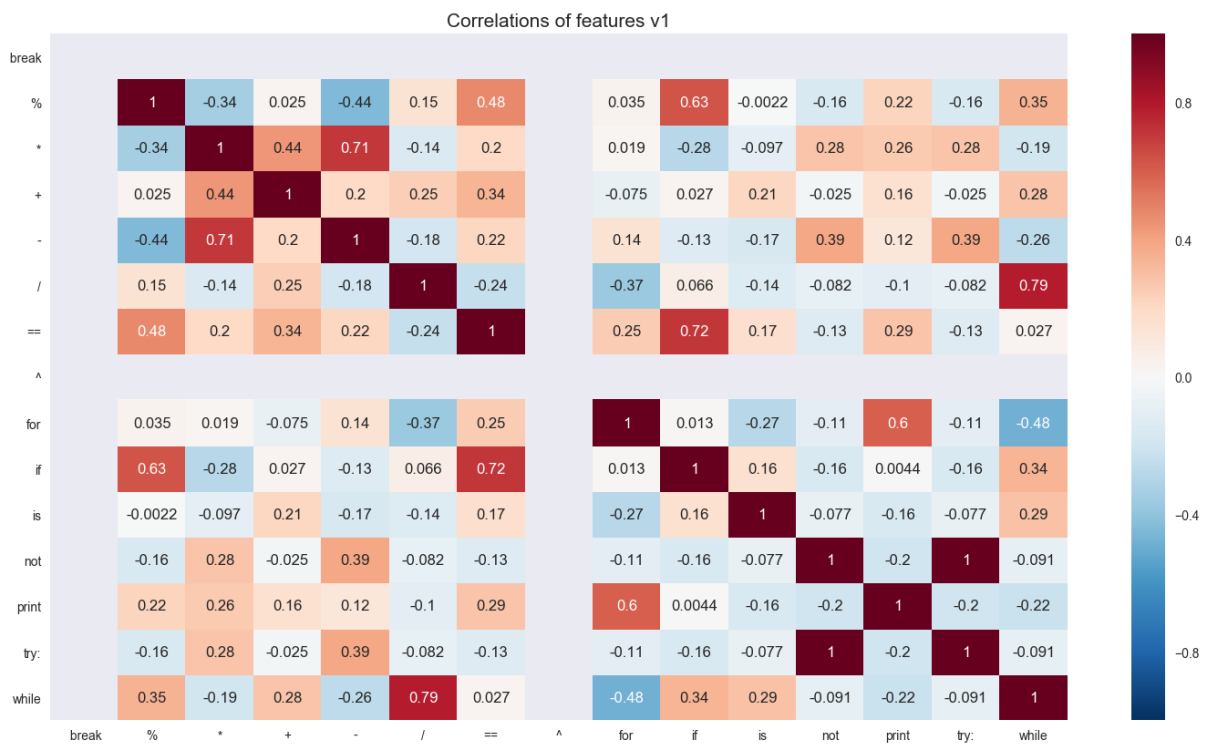
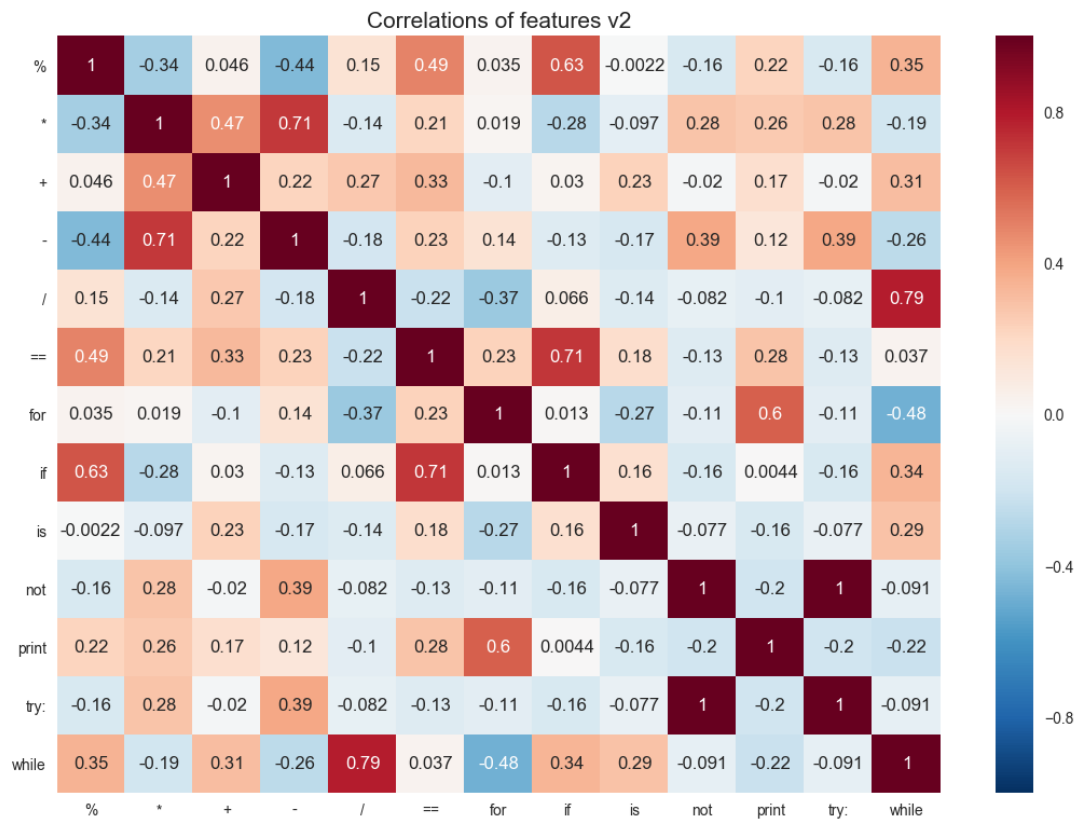


## Decision of what features to use

First try:



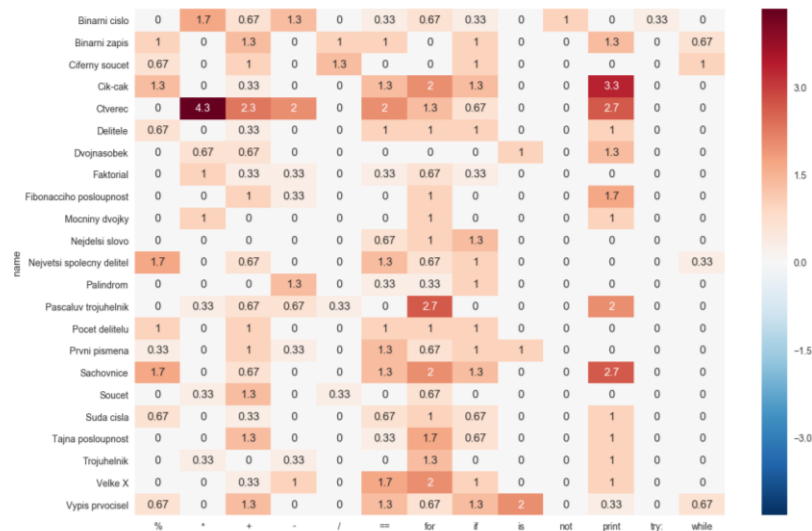
Results: the 'break' and '^' features were not found at all so I decided to remove them.



Next try with removed 'break' and '^' features:

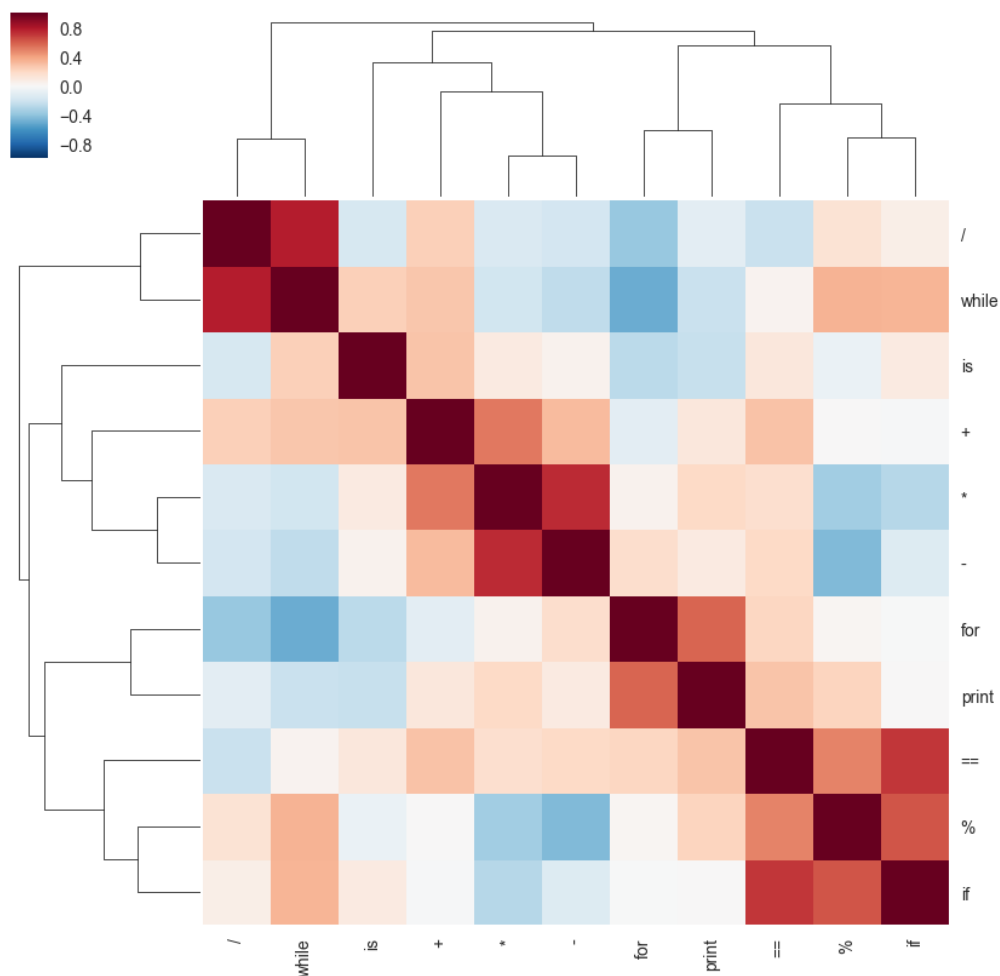
I had noticed that the correlation of 'try' and 'not' is too high so I decided to take a closer look at it.

This is the feature matrix used in the comparison above:



I have noticed that the 'try' and 'not' features have been used only in one item so they have a high correlation. I have decided to remove them as well.

The next correlation looks like this:



Results: We can see that some of the features has a high correlations but I think it is not that high that it must be removed. I think this features are the ones, that I would use.

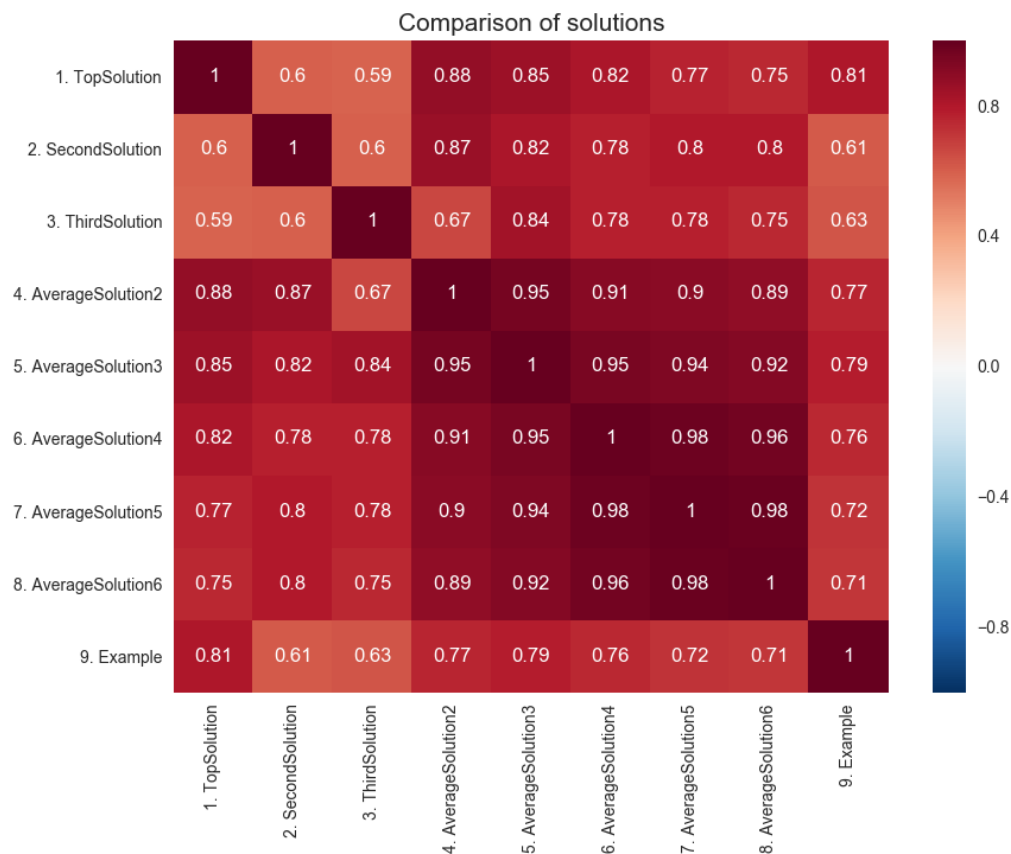
## Decision how to process given solutions

I have come up with two methods of using the user's solutions.

- Use one of the most frequently submitted solutions
- Do average of those solutions.

## Using similarities with all other items

I have created correlations for each of this method and then compared those methods how they



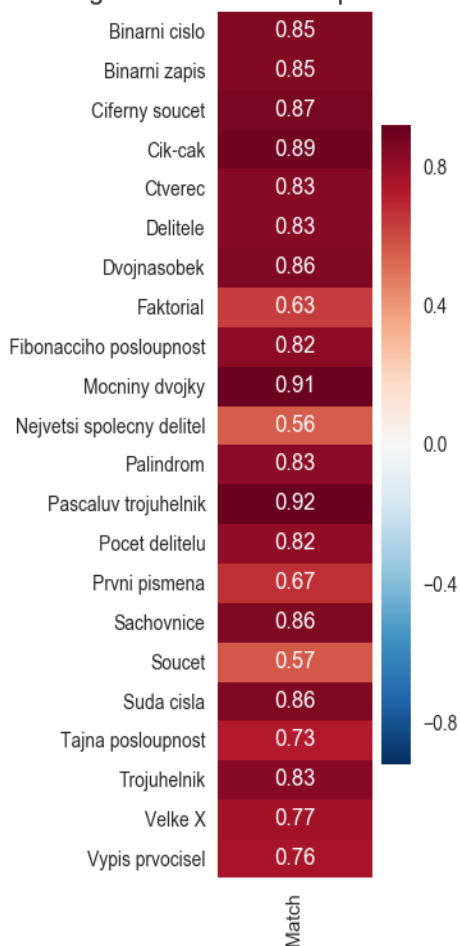
correlate. I have also included the 'Example' which is technique using bag of words on example solutions given by tutor.

From this heatmap we can see that the method using the most frequent solution 'TopSolution' and the method using the average of the three most frequent solutions 'AverageSolution3' are the ones that have the highest correlations with the example solution.

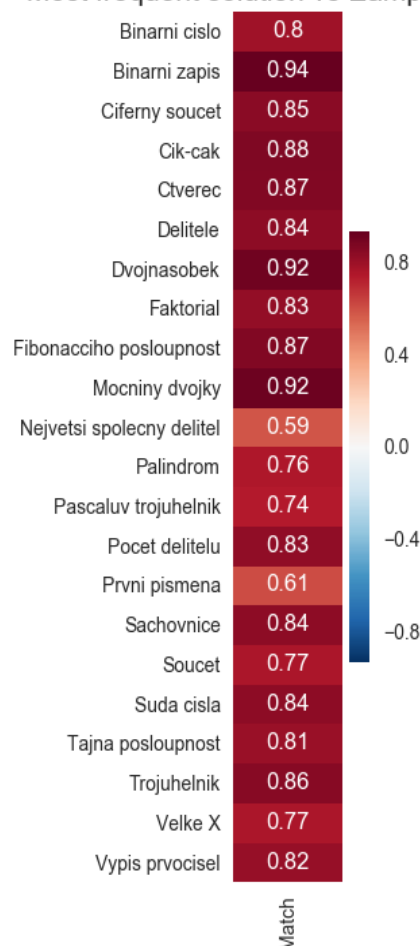
Now I have to decide which of those two methods is better.

I have compared those methods with the example method by each item:

Average3 method vs example solution method



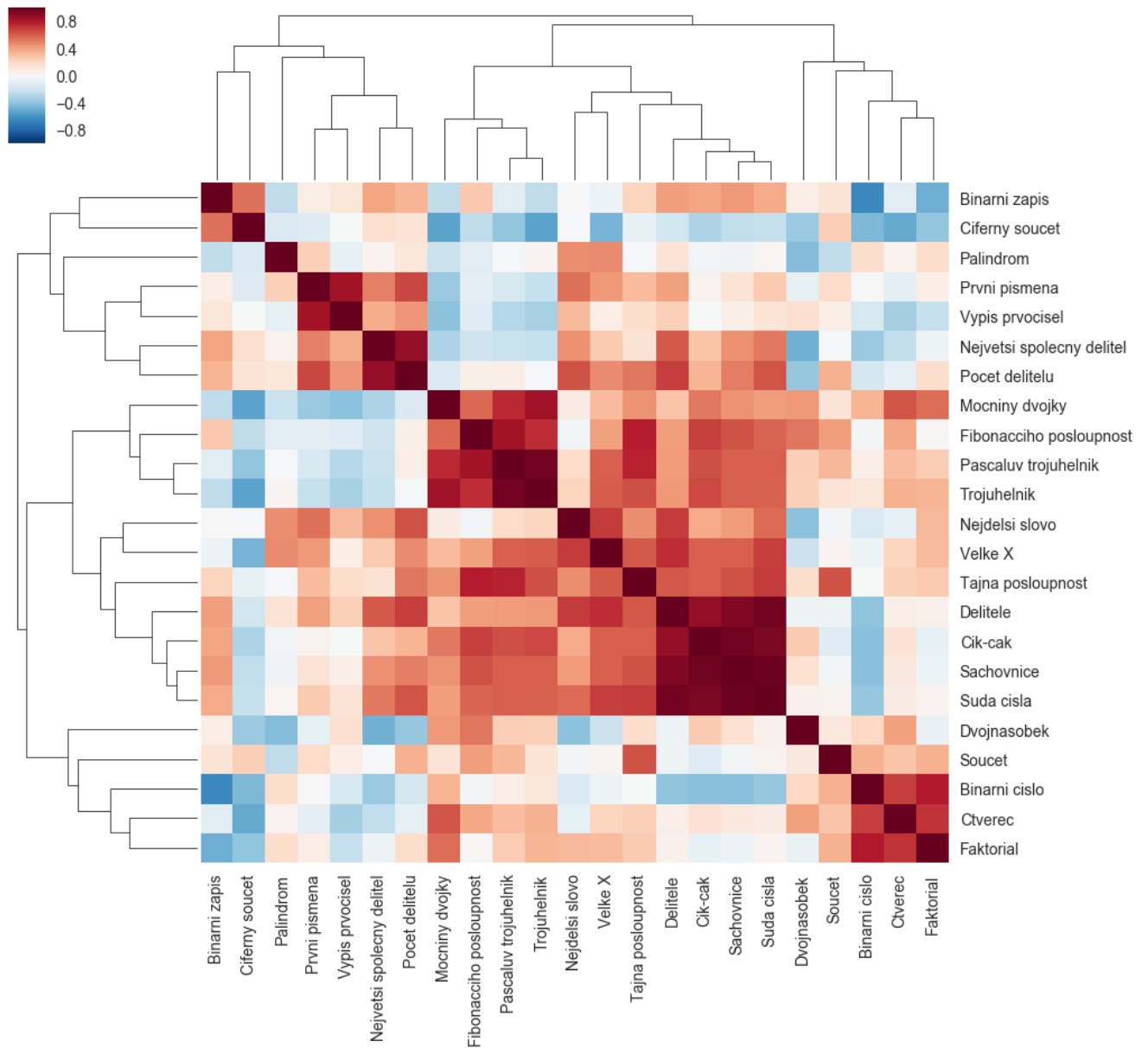
Most frequent solution vs Eample solution



We can see that the method using the most frequent solution has some higher correlations for some of the items but that's only because the most frequent solution was very similar to the example solution. The method using the average has lower average correlation with the example solution method but I think that it is a good think. It means that this method might include some solutions that are different than the example solution.

Decision: I would prefer to use the average solution method because it can do a better job comparing items which might have more than one correct solution.

Here is show the comparison of the items using the method I have chosen in the text above:



This comparison is using the Average3 method and using only those features I have decided to use in the text above.

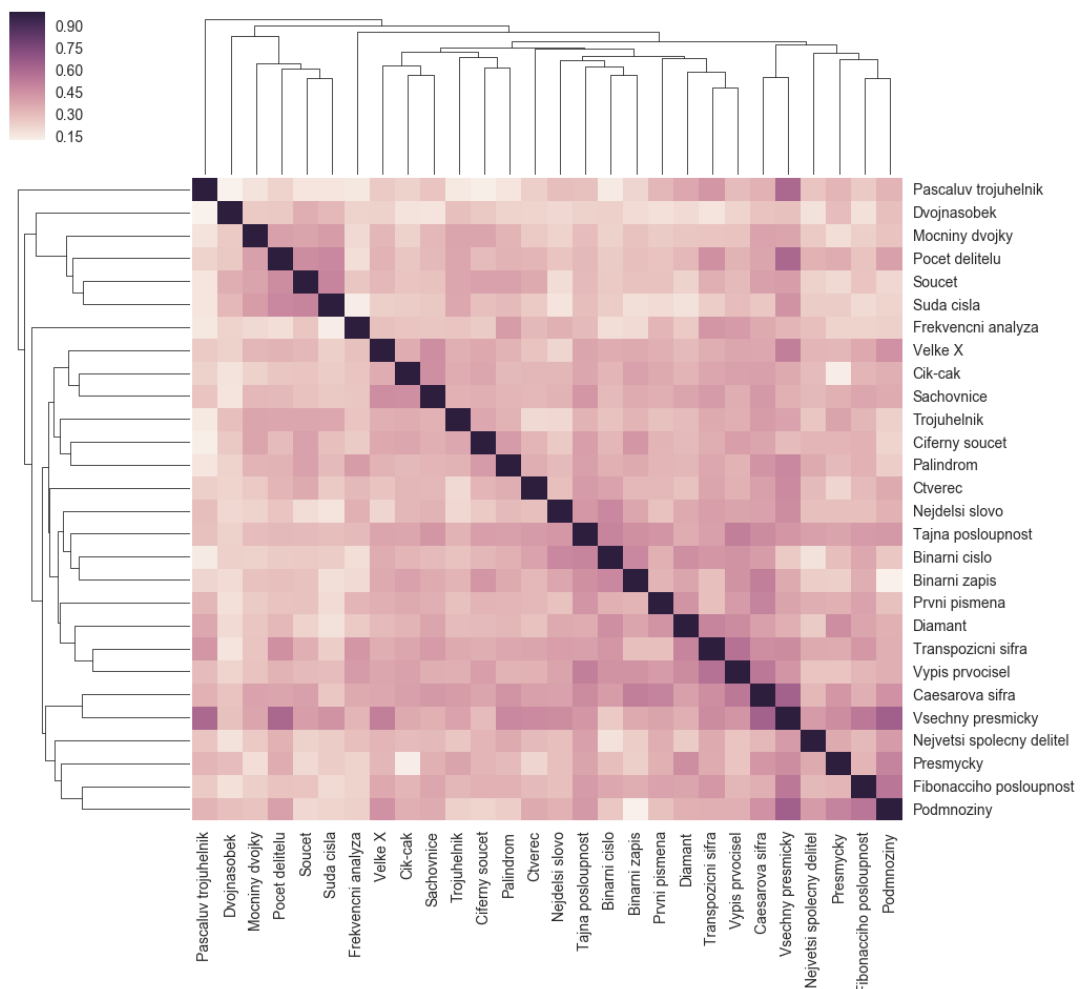
The second method how to compare programming items is comparing the user's performance. We say that two items are similar when the same user needed similar time to solve the problem.

Example:

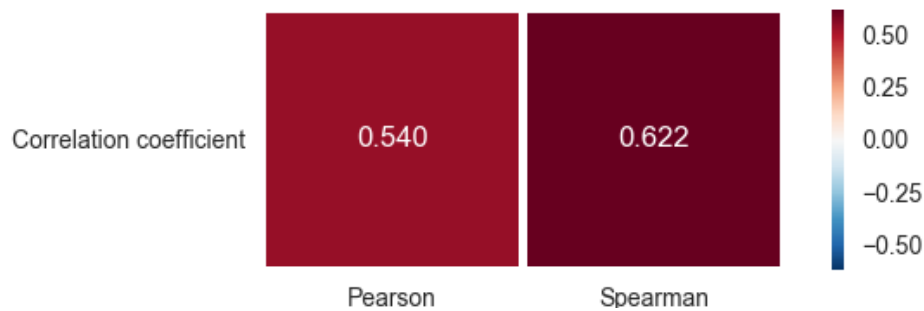
The value 'U22' is user's id. The missing values stands for items that were not solved by this specific user.

We can think of each line in this file like a single feature of the programming items. Then we can use correlation to determine the similar items.

Example of using correlation with spearman coeficient.

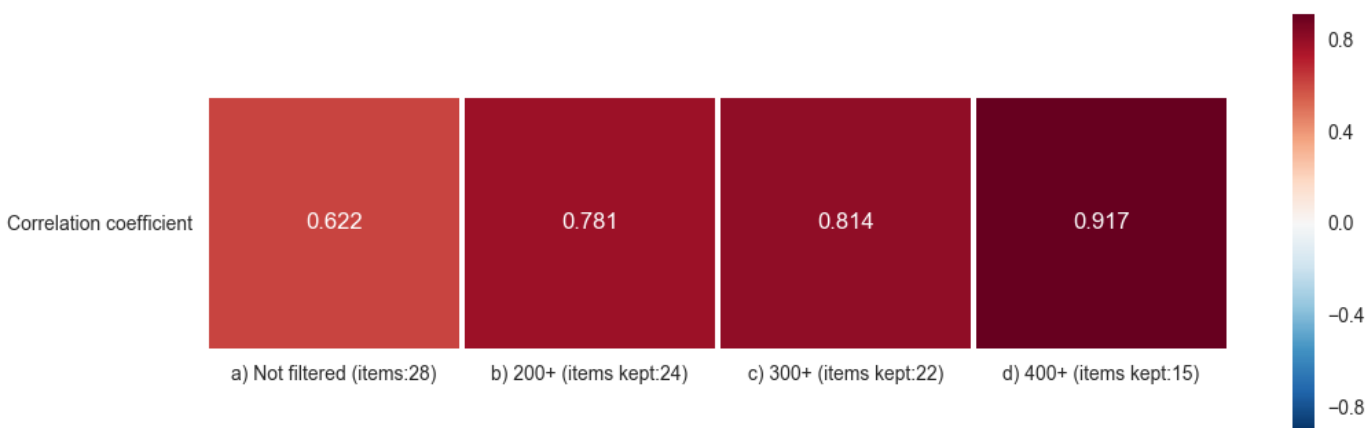


Now we should decide if the amount of data we have is sufficient. We can do this by splitting the data we have in half, calculating correlations of both halves and then comparing those two correlations how they are similar.



We can see that the spearman method gives more stable results. But still the correlation is not very high. We can try to improve this by filtering the data.

The method A is not using any filtration. The method B is filtering out items that were not



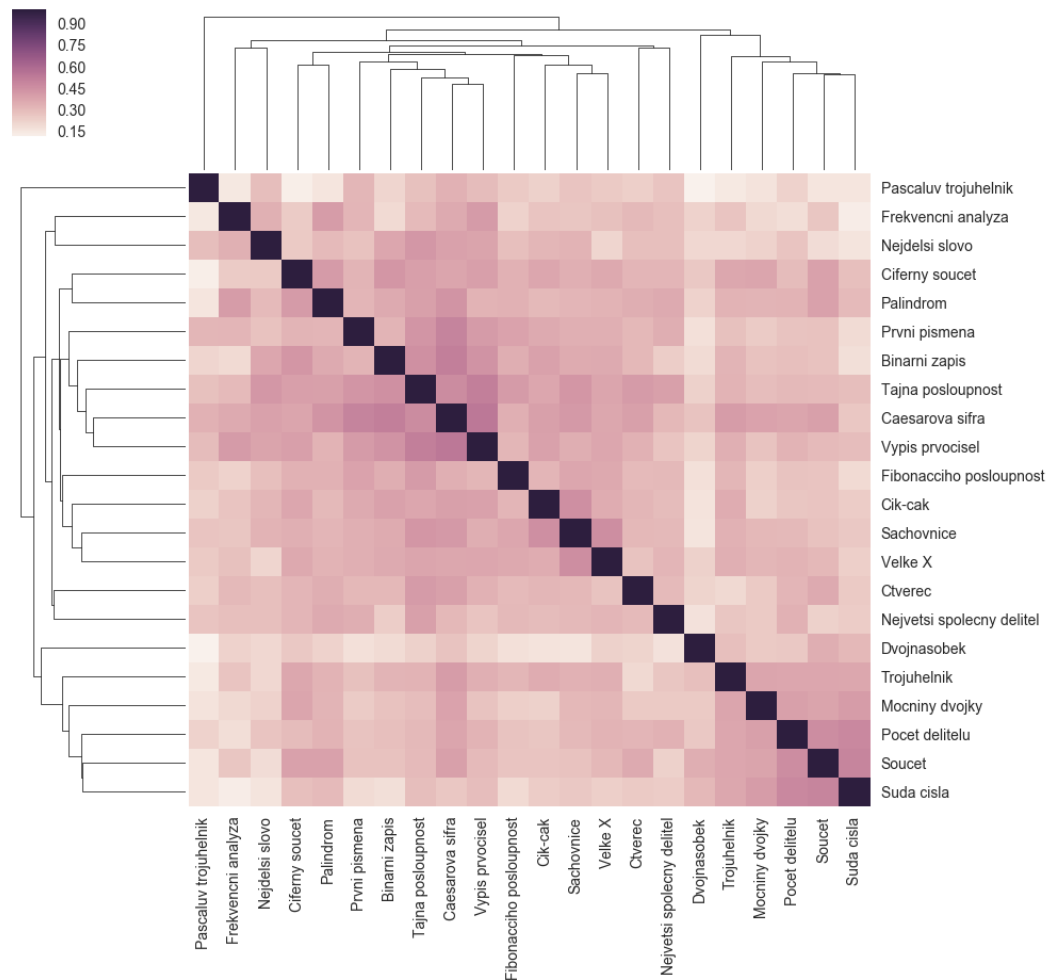
completed by at least 200 users. Method C and D are the same as the method B with the difference of minimal count of submissions.

The data we have contains data about items of different difficulty. The more difficult problems are solved by far less users than the simpler ones. We can filter those difficult items out.

The more items we filter the more the data is stable. But we want to filter out as few items as possible. From the graph above we have decided to choose the method C. It is quite stable and it kept 22 items out of 28.



Here is shown the item similarity matrix using correlation with the spearman correlation coefficient with items that were completed at least 300 times.

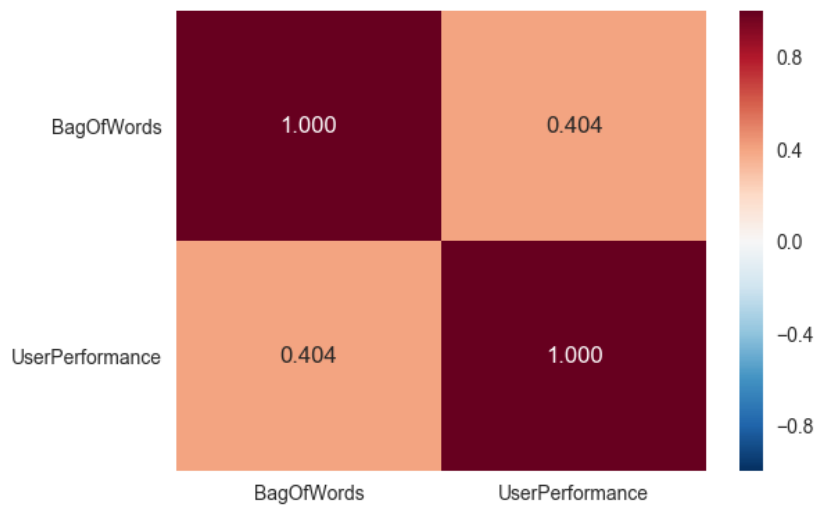


## Bag of word item similarity matrix vs User performance item similarity matrix

Now, we would like to test how the methods of comparing similar items give similar results.

### Comparison using correlation of item similarity matrices

First way how can we compare those methods is to simply calculate the correlation of the item similar matrices. This method is quite straight forward.



From the graph above we can see that the correlation is not that high.

### Comparison using the Top N method

When comparing two methods of comparing the programming items we might not care how differ the whole lists of most similar items. We might want to know how differ the most similar items. So, the second method is based on taking the n count of most similar items from each item similarity matrices and comparing how many items they have in common.

