Jacqueline Ritter
Nasreen Ahmed
Date – 5th Feb 2020

Subject: Machine Learning
Project Heading: Exploring Feature Selection

In this machine learning project, our aim was to do feature selection on biomedical data, which we found on Kaggle. The features of the data consist of over 20,000 different genes. The values of the features are numerical.

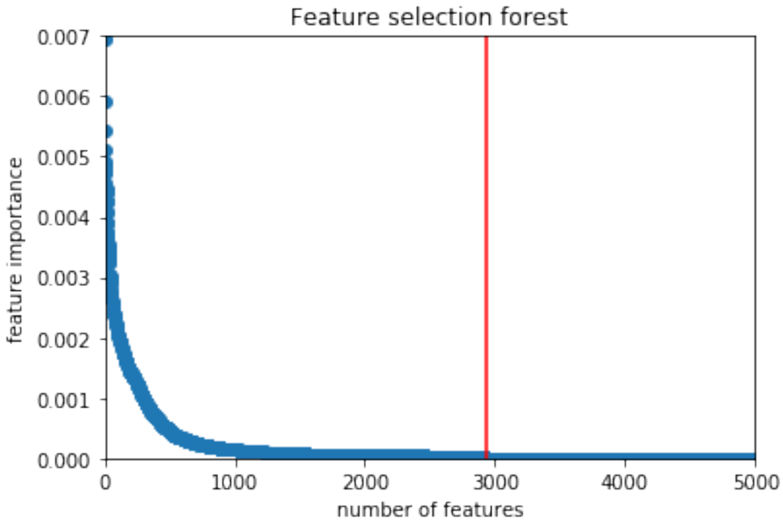
Feature selection approach

There are several feature selection methods, which can be grouped in embedded methods, filter methods and wrapped methods. In our project we focused on embedded methods, even though we also tried some filter methods. Embedded methods are algorithms that have an intrinsic feature selection, like for example random decision trees and random decision forests. Once a random decision tree is built, we can look at the feature score and check which features have been used and which have not. Even better the score also tells us how important the feature is in comparison to the others.

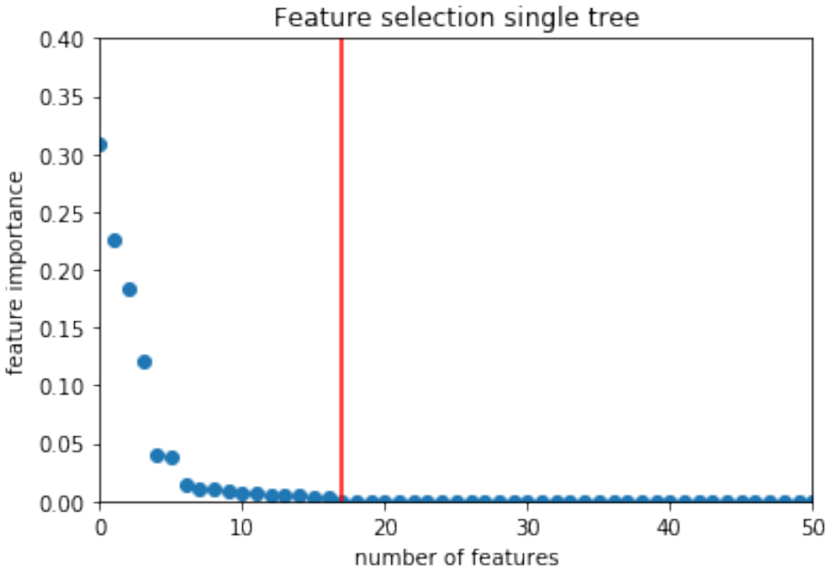
Before we could start to implement a chosen feature selection method, we needed to choose a classifier to test the results we obtained by using feature selection. As we wanted to use random decision trees to select our features, we decided to use the K-nearest neighbors' classifier to compare our results. At the beginning of the code we did a cross-validation to find out how many neighbors we need to use for the whole feature set to obtain the best accuracy score. Once we decided on a suitable feature subset, we will do the same cross-validation to find out the optimal K for the modified problem.

Methodology

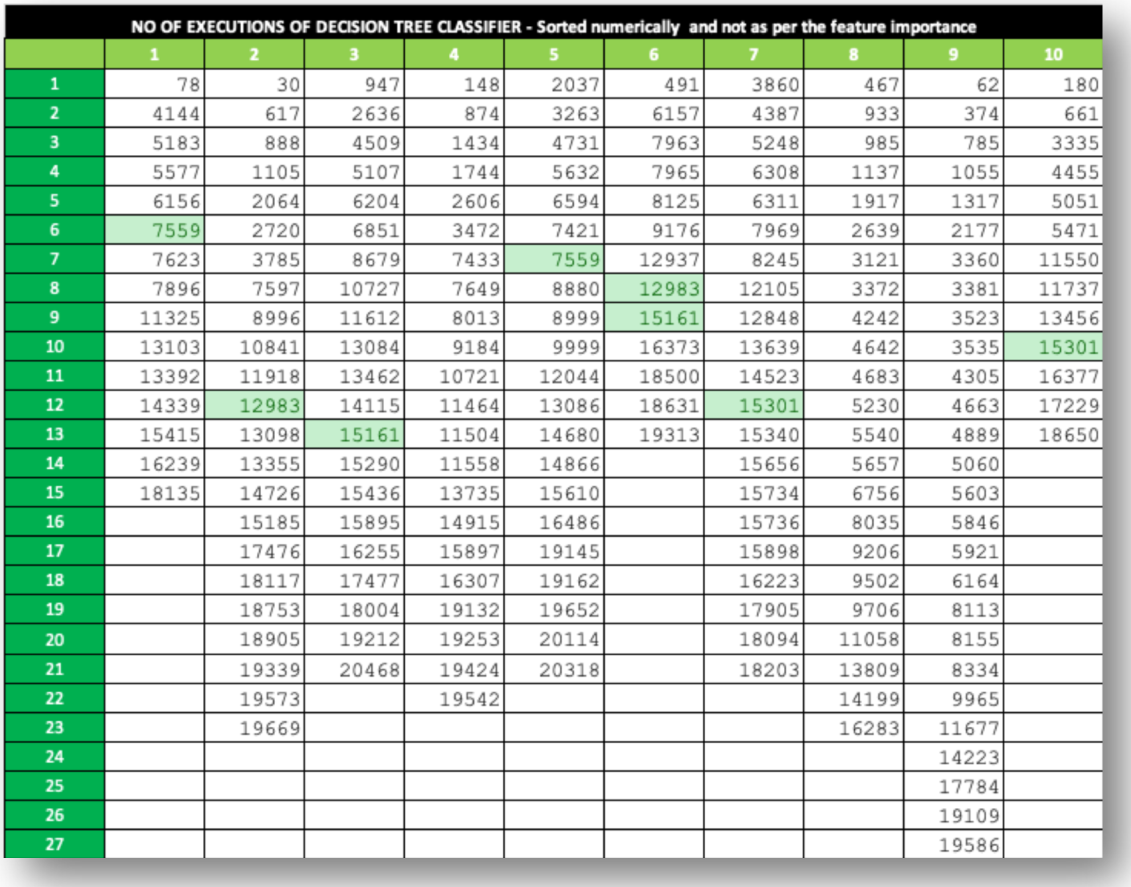
- We are going to evaluate the different results using cross validation and the train, test and validation split.
- Our first attempt was to look at the feature scores of a random decision forest that consists of 100 weak decision trees. Roughly 3000 features have been used in total



However, our aim was to select still fewer features. That is why we decided to look at the features that have been chosen by single, but strong decision trees. On average those classifiers chose about 20 features from the given 20,000 features have not been the same can we still get less than 20 features? K nearest neighbor classifier to test



- As the decision tree follows greedy approach, for every fresh iteration we received a new set of features.

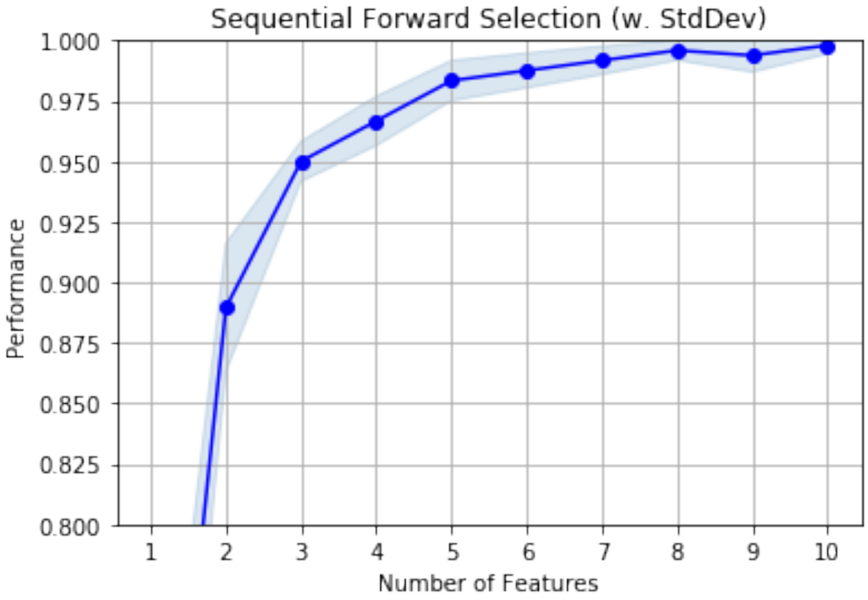


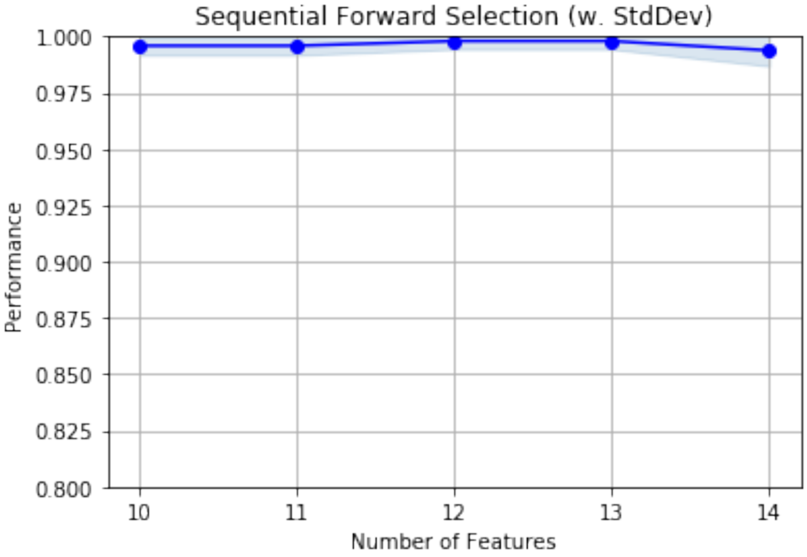
NO OF EXECUTIONS OF DECISION TREE CLASSIFIER - Sorted numerically and not as per the feature importance

	1	2	3	4	5	6	7	8	9	10
1	78	30	947	148	2037	491	3860	467	62	180
2	4144	617	2636	874	3263	6157	4387	933	374	661
3	5183	888	4509	1434	4731	7963	5248	985	785	3335
4	5577	1105	5107	1744	5632	7965	6308	1137	1055	4455
5	6156	2064	6204	2606	6594	8125	6311	1917	1317	5051
6	7559	2720	6851	3472	7421	9176	7969	2639	2177	5471
7	7623	3785	8679	7433	7559	12937	8245	3121	3360	11550
8	7896	7597	10727	7649	8880	12983	12105	3372	3381	11737
9	11325	8996	11612	8013	8999	15161	12848	4242	3523	13456
10	13103	10841	13084	9184	9999	16373	13639	4642	3535	15301
11	13392	11918	13462	10721	12044	18500	14523	4683	4305	16377
12	14339	12983	14115	11464	13086	18631	15301	5230	4663	17229
13	15415	13098	15161	11504	14680	19313	15340	5540	4889	18650
14	16239	13355	15290	11558	14866		15656	5657	5060	
15	18135	14726	15436	13735	15610		15734	6756	5603	
16		15185	15895	14915	16486		15736	8035	5846	
17		17476	16255	15897	19145		15898	9206	5921	
18		18117	17477	16307	19162		16223	9502	6164	
19		18753	18004	19132	19652		17905	9706	8113	
20		18905	19212	19253	20114		18094	11058	8155	
21		19339	20468	19424	20318		18203	13809	8334	
22		19573		19542				14199	9965	
23		19669						16283	11677	
24									14223	
25									17784	
26									19109	
27									19586	

We then executed the code multiple times and then counted the number of times a feature was selected by the tree. We then chose a feature subset consisting of all the features that appeared in more than one tree. (Makes the subset a bit smaller than the average number of selected features by a single tree)

- To further curate the list of features obtained from the feature subset, we implemented the Sequential Forward and Backward Elimination algorithm and found the following results.

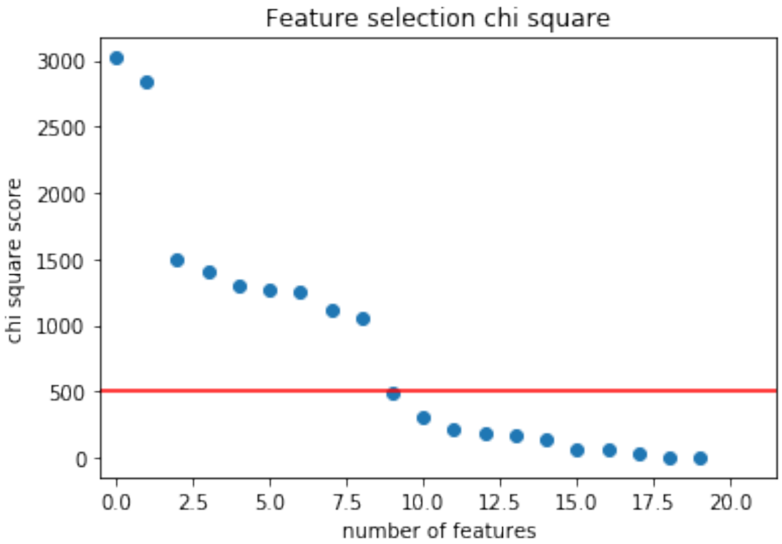




- The final list of 10 features giving accuracy of 0.97 and 0.98 in predicting the right labels.

Apart from the embedded methods, we have also tried implementing few filter methods to curate notable features:

- Chi-square



We can see that the chi squared values are torn apart. That is why we assume it is justified to try to cut the features that have a chi 2 score lower than 500 (there is a gap).

- Pearson Correlation: Implemented in the notebook

Appendix:

Link to Colab Notebook