Introduction to ML (NPFL054)

Homework #3

Name: Jakub Hejhal

School year: 2nd

Part 1 – Data analysis and feature filtering

a)

The proportion of active and non-active ligands is the same in d1 as in d2 and it is 1:20.

b)

```
Number of discrete features: 104
Number of continuous features: 16
```

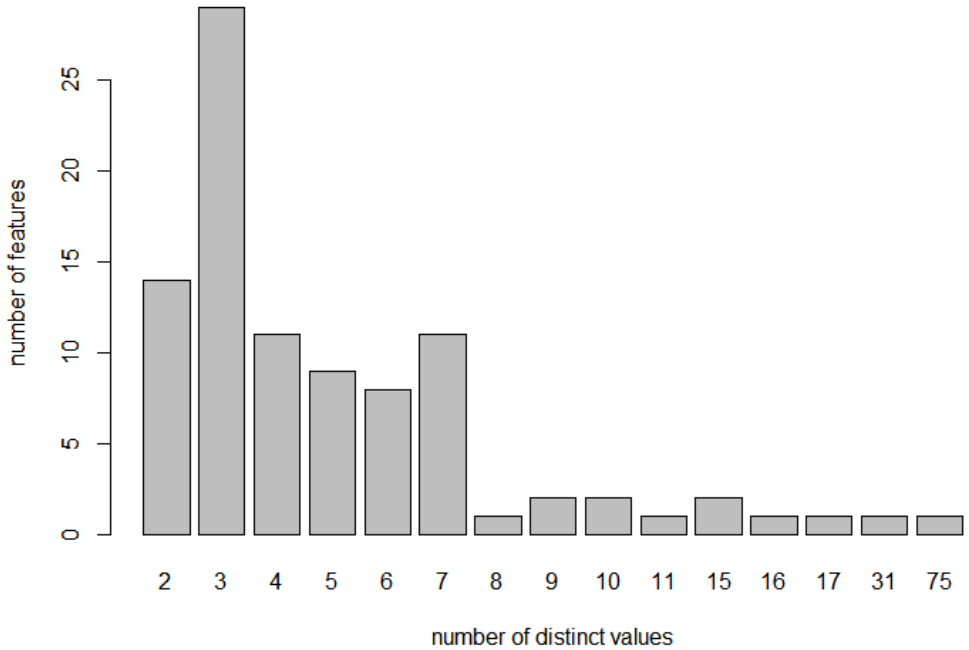
c)

```
Constant features are:
NumRadicalElectrons, fr_azide, fr_benzodiazepine, fr_diazo, fr_epoxide,
fr_isocyan, fr_isothiocyan, fr_nitroso, fr_prisulfonamd, fr_thiocyan

After removing constant features, there are 110 features remaining
```

d)

Number of values	2	3	4	5	6	7	8	9	10	11	15	16	17	31	75
Number of features	14	29	11	9	8	11	1	2	2	1	2	1	1	1	1



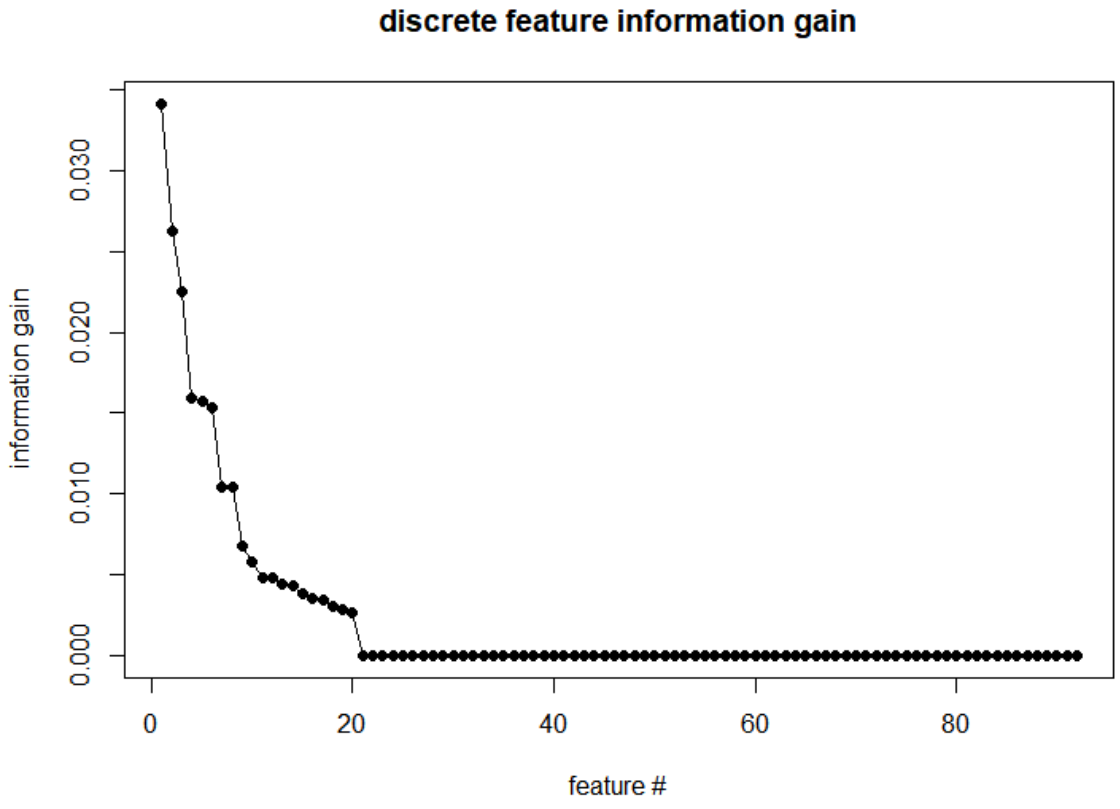
e)

When frt = 4, then only 2 binary features - fr_phos_acid and fr_phos_ester don't satisfy the condition min(fr(A), n-fr(A)) >= frt. After removing these features, there are 108 features in feature vector, from which 92 are discrete.

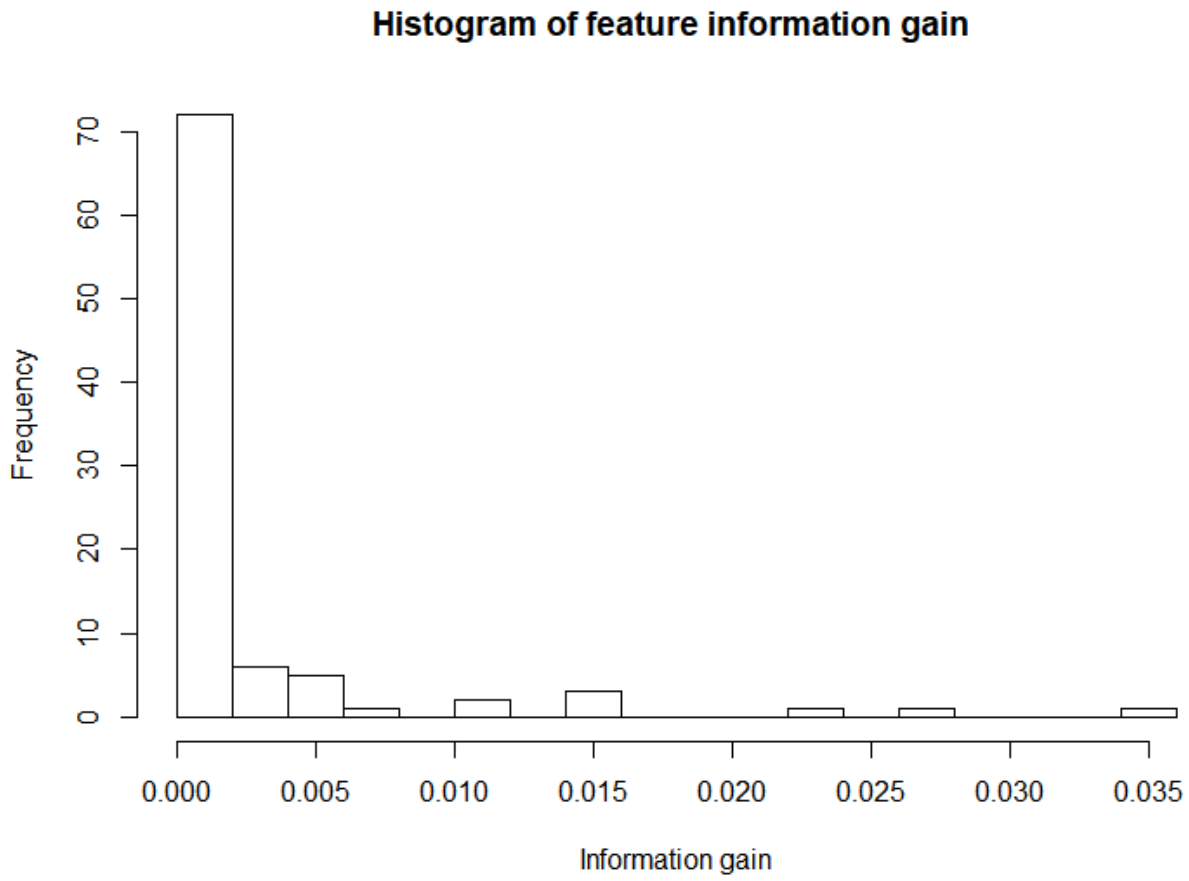
f)

Feature name	Information gain
NumAromaticHeterocycles	0.034153465
fr_Ar_N	0.026327765
NumAromaticRings	0.022578862
fr_furan	0.015928678
fr_NH0	0.015733522
fr_ArN	0.015359575
fr_C_O	0.010410970
fr_C_O_noCOO	0.010410970
RingCount	0.006768526
fr_amide	0.005739577
NumAromaticCarbocycles	0.004811570
fr_benzene	0.004776517
fr_NH1	0.004376109
NumHAcceptors	0.004325674
fr_nitro_arom	0.003799977
fr_nitro	0.003501224
fr_imidazole	0.003413015
fr_NH2	0.002993372
NumHDonors	0.002867073
fr_Al_OH_noTert	0.002604004
NumValenceElectrons	0.000000000
HeavyAtomCount	0.000000000
NHOHCount	0.000000000
NOCount	0.000000000
NumAliphaticCarbocycles	0.000000000
NumAliphaticHeterocycles	0.000000000
NumAliphaticRings	0.000000000
NumHeteroatoms	0.000000000
NumRotatableBonds	0.000000000
NumSaturatedCarbocycles	0.000000000
NumSaturatedHeterocycles	0.000000000
NumSaturatedRings	0.000000000
fr_Al_COO	0.000000000
fr_Al_OH	0.000000000
fr_Ar_COO	0.000000000
fr_Ar_NH	0.000000000
fr_Ar_OH	0.000000000
fr_COO	0.000000000
fr_COO2	0.000000000
fr_C_S	0.000000000
fr_HOCCN	0.000000000
fr_Imine	0.000000000
fr_N_O	0.000000000
fr_Ndealkylation1	0.000000000
fr_Ndealkylation2	0.000000000
fr_Nhpyrrole	0.000000000
fr_SH	0.000000000
fr_aldehyde	0.000000000
fr_alkyl_carbamate	0.000000000
fr_alkyl_halide	0.000000000
fr_allylic_oxid	0.000000000
fr_amidine	0.000000000
fr_aniline	0.000000000
fr_aryl_methyl	0.000000000
fr_azo	0.000000000
fr_barbitur	0.000000000
fr_bicyclic	0.000000000
fr_dihydropyridine	0.000000000
fr_ester	0.000000000

```
fr_ether                  0.000000000
fr_guanido                0.000000000
fr_halogen                0.000000000
fr_hdrzine                0.000000000
fr_hdrzone                0.000000000
fr_imide                  0.000000000
fr_ketone                 0.000000000
fr_ketone_Topliss         0.000000000
fr_lactam                 0.000000000
fr_lactone                0.000000000
fr_methoxy                0.000000000
fr_morpholine             0.000000000
fr_nitrile                0.000000000
fr_nitro_arom_nonortho    0.000000000
fr_oxazole                0.000000000
fr_oxime                  0.000000000
fr_para_hydroxylation     0.000000000
fr_phenol                 0.000000000
fr_phenol_noOrthoHbond    0.000000000
fr_piperdine              0.000000000
fr_piperzine              0.000000000
fr_priamide               0.000000000
fr_pyridine               0.000000000
fr_quatN                  0.000000000
fr_sulfide                0.000000000
fr_sulfonamd              0.000000000
fr_sulfone                0.000000000
fr_term_acetylene         0.000000000
fr_tetrazole              0.000000000
fr_thiazole               0.000000000
fr_thiophene              0.000000000
fr_unbrch_alkane          0.000000000
fr_urea                   0.000000000
```



Histogram showing the distribution of information gain (similar to p.d.f.)



Part 2 – Baseline model for automatic classification

a)

Precision = TP / (TP + FP), therefore precision can be equal to 100% only if FP=0%. As the FPR increses, FP must increase as well (because FP = FPR * N, where N is constant), so the highest possible precision goes down. We know that N = 0.95 and that P=0.05 in our dataset. When FPR is already at 10%, it means that FP = 0.1 * 0.95, so the highest possible precision is when all active ligands are identified correctly, i.e. FN=0, TP=P. When this happens, the precision is then 0.05 / (0.05 + 0.1 * 0.95) = 34.48%

b)

AUC0.1 mean estimate: 0.0857437100248662

Standard deviation: 0.00531527262818578

Confidence interval: 0.0819413930431766, 0.0895460270065557

c)

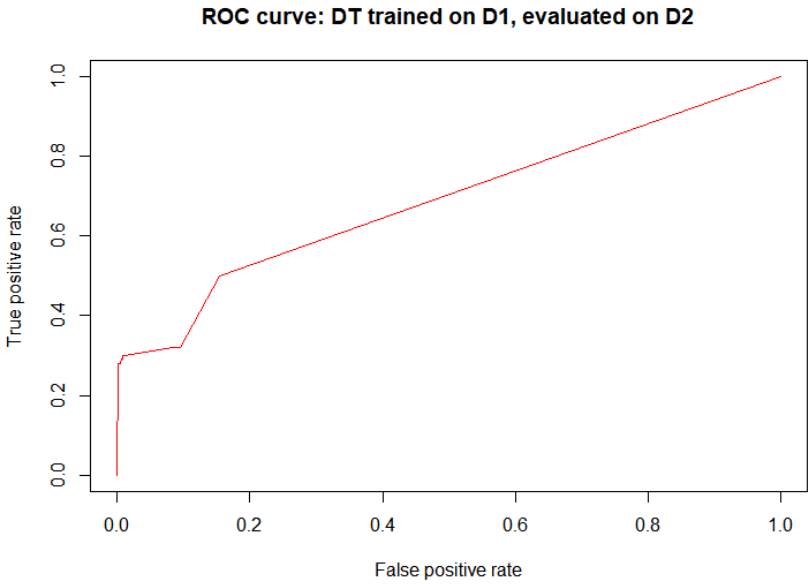
cp	AUC01	Standard dev.	Standard err.
0.3	0.02761	0.02396	0.00758
0.24	0.05181	0.01274	0.00403
0.192	0.05713	0.01699	0.00537
0.1536	0.05303	0.01965	0.00621
0.12288	0.06037	0.02206	0.00698
0.0983	0.06926	0.01039	0.00329
0.07864	0.06853	0.00927	0.00293
0.06291	0.07004	0.00827	0.00262
0.05033	0.06838	0.01041	0.00329
0.04027	0.07044	0.00902	0.00285
0.03221	0.07151	0.00577	0.00182
0.02577	0.0791	0.00795	0.00251
0.02062	0.0828	0.00644	0.00204
0.01649	0.08542	0.00448	0.00142
0.01319	0.0859	0.00415	0.00131
0.01056	0.08633	0.00633	0.002
0.00844	0.08475	0.00604	0.00191
0.00676	0.08694	0.00818	0.00259
0.0054	0.08698	0.00528	0.00167
0.00432	0.08764	0.00252	8e-04
0.00346	0.0881	0.00549	0.00174
0.00277	0.08763	0.00601	0.0019
0.00221	0.08756	0.00513	0.00162
0.00177	0.08685	0.00485	0.00153
0.00142	0.08737	0.00694	0.0022

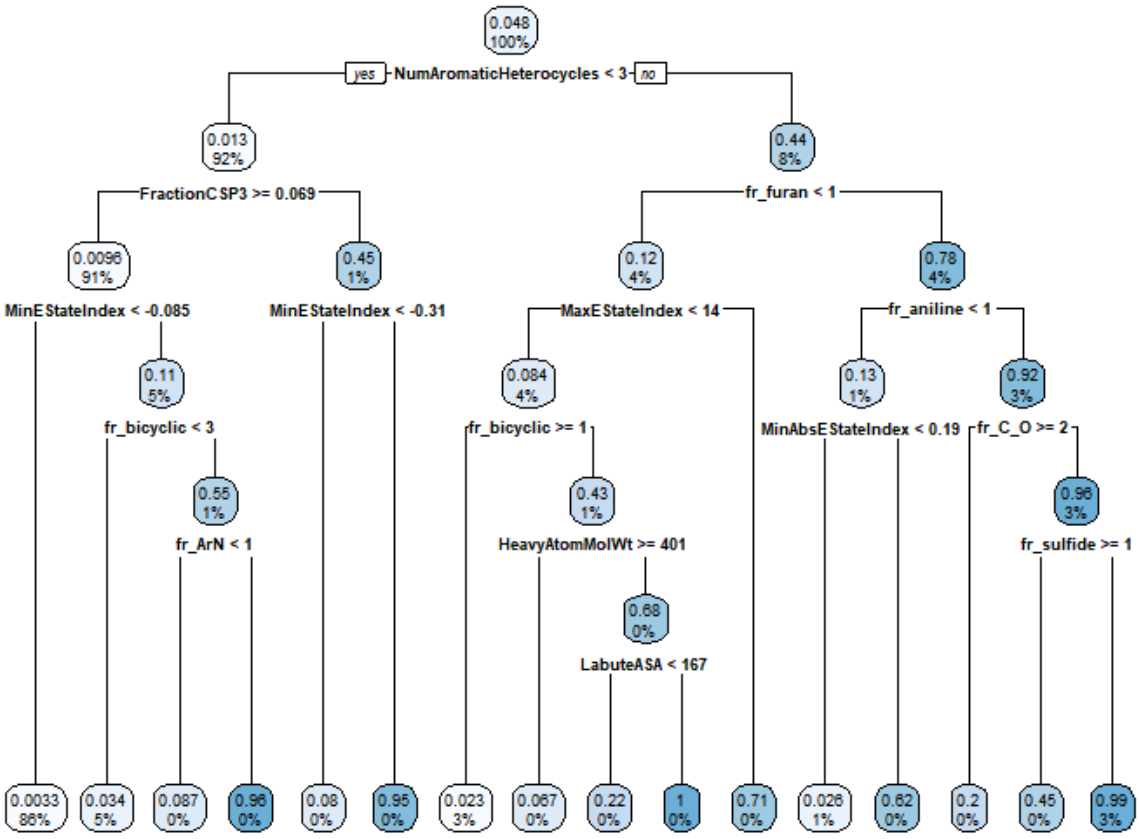
Maximum mean of AUC01 is reached when cp=0.00346, SE=0.00174. Maximum cp, for which the mean of AUC01 is at least max_auc01 – SE = 0.0881 – 0.00174 = 0.08636 is **0.00676**, so this is the optimal cp value.

d)

DT trained on D1 wtih cp=0.00676 gives on D2 AUC_{01} of only 0.030725, which is significantly less than maximum mean AUC_{01} on D1 (0.881). It may be a hint, that the tree is overfitted, or that simple decision tree algorithm is not very suitable for this kind of task and more sofisticated algorithm is required. Furthermore, D1 and D2 aren‘t samples from

identical population and are statistically a bit different, which may be another reason for the low AUC_{01}.

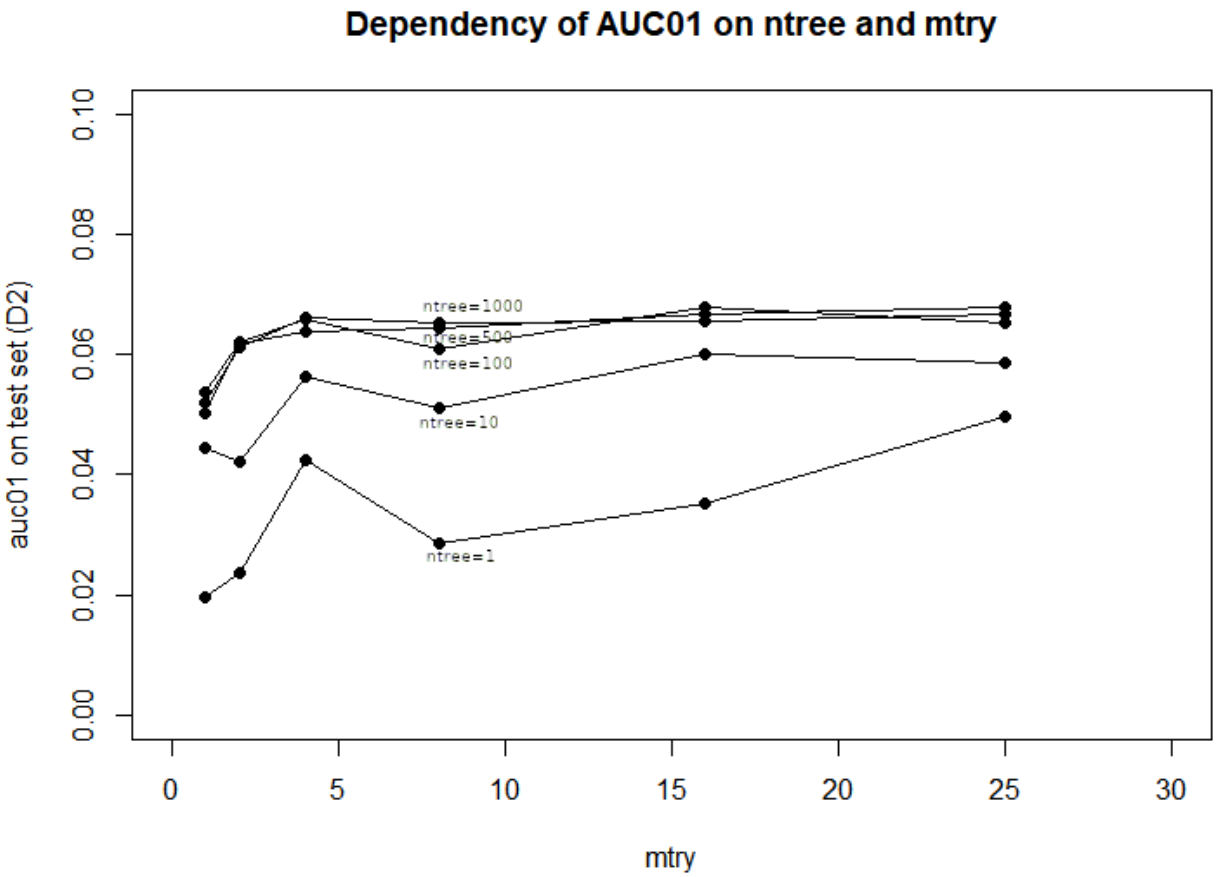




The tree plot shows, that the tree is quite complicated and probably overfitted.

Part 3 – More advanced models

a, b)



Random forest model was trained on D1 using various mtry and ntree parameter values and then evaluated on D2

This graph shows relationship between mtry, ntree and $AUC_{0.1}$**.** Single line represents constant ntree. The bottom outlier represents ntree=1, so it is understandable, that single tree performs poorly. As ntree get larger, the $AUC_{0.1}$ performance is on average better and more importantly, gets more consistent, with many trees in the pool, things tend to average out and the result is model with low variance. When mtry=2, RF performs poorly no matter the ntree, but anything above mtry=4 seem to have very little effect on the performance.

Therefore probably the best will be to stick to the default mtry=10, and because random forests cannot really get overfit, ntree will be as high as possible (something like 2000).

c)

DT trained on D1 in part 2 scored only 0.030725 on D2, whereas all RF trained on D1 with at least 10 trees were above 0.4 when evalued on D2. Altough RFs are much slower than DTs (can be paralelized), they are much more robust, stable and aren‘t prone to overfitting. Therefore RF model is more suitable for this tasks than simple DT.

Part 4 – Experiments with different data sets

Experiment	train set	test set	CV mean $AUC_{0.1}$	CV SD	CV CI	D2 $AUC_{0.1}$
a)	4/5 D1	1/5 D1	0.09638	0.002134	(0.09372, 0.09903)	0.06580
b)	D1 + 1/5 D2	4/5 D2	0.08772	0.001649	(0.08568, 0.08977)	
c)	D1 + 4/5 D2	1/5 D2	0.09368	0.008842	(0.0827, 0.10465)	
d)	4/5 D2	1/5 D2	0.09652	0.002987	(0.09282, 0.10023)	

e)

Results of experiment a) and show that training on 4/5 of D1 and testing on remaining 1/5 of the same data set yields very high $AUC_{0.1}$, but we can see that the model doesn't generalize well to data from different population (D2). The same thing would probably happen if the model from d) would be evaluated on D1.

Experiment b) shows, that when we mix into the training set only a little bit of D2, test results on D2 improve drastically. Somehow, D1 is missing some essential information needed for good performance on D2.

Mixing in even more of D2 to our training set in experiment c) unfortunately makes that test set very small (only 20 active ligands!), which in turn makes CV confidence interval very large, so it's not clear, how good actually the model is, but I would guess it is almost as good as the d) model (when evaluated on D2), while still knowing a lot about D1, which may be very useful, when we do the blind classification.

Part 5 - Final model selection and prediction on the blind test set

b)

I chose the random forest algorithm, rather than the baseline, because it performed obviously much better overall. The hyperparameters chosen were ntree=2000 (as high as possible) and mtry=10, reasons for this exact parameters choice are given in **Part 3** of the report. Than I trained it on all available data (D1 + D2), because even though RF trained only on D2 gave the best results when evaluated on itself (recall experiment 4d), the model in experiment 4c wasn't far off. Because of the small size of the available data (only 400 active ligands), every piece of data is very useful, so the D1+D2 model will be probably more robust than the D2 model (D2 has **only** 100 active ligands). Furthermore, there may be some information about the ligands in D1, that is not in D2, that may turn out to be useful in T blind prediction.

c)

We can estimate the precision on blind test set by empiricaly measuring it on D2, because D2 is close to T data set. The problem is that the model we use in part 5a has already seen all the data during the training phase, so we cannot it to estimate the precision. So we do some kind of a compromise. We will split D2 into 2 parts: test part and train part. test part will have the same number of examples as blind set and the rest of D2 together with D1 will be used for the training. This model is quite similar to the one in 4b.

The precision estimates are:

d.50 precision: 0.92

d.150 precision:0.5

d.250 precision:0.308

# of predicted ligands	precision	TP
50	0.92	46
150	0.5	75
250	0.308	77

Given that there are only 82 active ligands in the test set, the results are quite good. In d.50 case, 46 identified ligands were actually active, in d.150, the model missed only 7 active ligands out of 82, and in the d.250 case, only 5 active ligands weren't selected by the model. Given that the model was trained on some examples from D2 population and given that I can't train it on data from T population, I would expect the precision on T to be a bit lower than the empiricaly measured estimate above.

d)

Recall = TP / P, In our case, P is a constant (in total 82 positives) If we assume the estimates in 5c are accurate, the recall on T set would be around:

# of predicted ligands	recall
50	0.561
150	0.915
250	0.940