MLiC_HW2

Ali Yesilkanal

November 9, 2017

Part 1

Task: Learn a model that can predict drug response on a per drug basis (by Drug) across the GDSC cell lines

A quick look at the data provided revealed that even though the GDSC data base has over 1000 cell lines and 265 drugs, not all of the cell lines have expression profile. Similarly, not all cell lines with expression profile have drug IC50 information. Therefore, I first wanted to identify drug-cell line pairs that have expression information available for this analysis.

I first filtered the dose response data for the drug-cell line pairs that have AUC values higher than 0.9. After this filtering there was no change in the total number of drugs that have been tested, but each drug had fewer paired cell lines because of the AUC filter than they had had before the filter was applied.

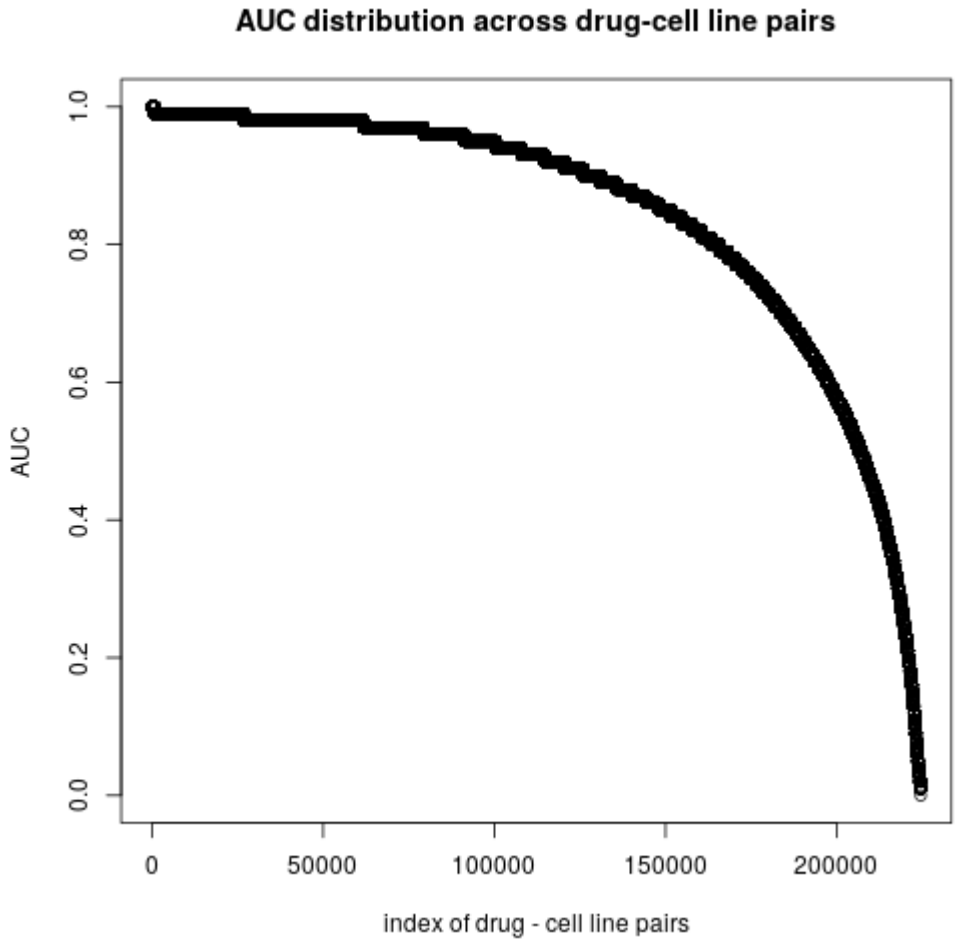

Figure 1: Ranking of AUC for each drug - cell line pair

Then I applied a second filter, where I only selected the drug-cell line pairs where there is expression data for the cell line. The following table shows a few of the drugs and the number of cell lines that they have been tested on as an example (after the two filtering steps).

```
##    DRUG_ID Num_CL_per_Drug
## 1        1             323
## 2     1001             362
## 3     1003              98
## 4     1004              84
## 5     1005             483
## 6     1006             245
## 7     1007             226
## 8     1008             581
## 9     1009             711
## 10    1010             738
```

For this task, I will use gene expression data as the feature matrix, but each feature matrix has ~18,000 features (genes) in this case. In order to reduce the number of features, I looked for genes the expression value of which correlated with the outcome (IC50) of each drug across multiple cell lines, based on spearman correlation. I created a matrix for each individual drug where the first column is the LN_IC50 values across different cell lines and the rest of the columns are the gene expression matrix. Then I selected top 30 genes that correlated highly with the LN_IC50 outcome. The correlations can be negative or positive; only the absolute value of the spearman coefficient has been considered in this feature selection. Each drug has a different set of 30 genes (features). For example, the following matrix shows 5 of rows of the feature matrix for Drug 1.

```
##   COSMIC_ID LN_IC50     CBFB     TAF8    ROBO1    ZNRD1   KHDRBS3   KCNMA1
## 1    683665    2.44 7.381675 4.046006 3.865636 9.077781  6.560701 7.031678
## 2    684055    3.34 7.073084 3.971642 5.308128 8.938284 10.391483 3.688912
## 3    684057    3.57 6.924102 4.049746 4.876562 8.873273 10.617343 3.273957
## 4    684059    3.19 7.645027 4.065640 6.543183 8.816438  9.530215 3.332459
## 5    684062    2.46 8.013019 3.899042 4.560593 8.403494  9.321044 3.460722
##     STAT5B    MRPL16     ELF1     CD38   GAS2L3    RAB2A     MGST3
## 1 5.440448 10.025188 7.735772 9.591007 5.282070 6.448660  8.183073
## 2 3.992091  8.419572 5.446713 2.731800 8.751107 8.099281  9.982433
## 3 4.583204  9.213105 5.002667 2.859110 8.221462 8.153698 11.663702
## 4 4.052993  9.130500 5.459796 3.094018 7.075008 7.721207  9.570292
## 5 4.172765  9.295527 5.458624 3.100430 7.472950 7.779979 11.193158
##      TRBC2     X.14 FAM177A1    TIAL1    CCND3   MAGED2   ANP32B   NEDD4L
## 1 9.942958 6.816912 5.689164 8.939407 4.733044 5.574793 10.71191 4.481254
## 2 3.080555 3.327946 6.606224 8.315006 3.683693 8.522961 11.01311 4.928425
## 3 3.193664 6.200406 6.926580 8.226353 3.818064 9.086599 10.18925 5.453939
## 4 3.176467 5.248458 6.789393 8.811677 3.440980 8.805028 10.80717 5.468753
## 5 3.030396 7.433340 6.391761 8.436430 4.164414 8.758842 10.65421 5.427052
##     MAPK14  KIAA0922  FAM129C   CCDC69    GNA15   ELAVL1   ABHD12     HHEX
## 1 5.164008 10.055993 3.392460 5.744546 5.300199 7.459808 5.402310 3.379929
## 2 4.743144  6.599129 3.078927 3.262694 3.156875 7.762851 5.558390 3.085935
## 3 5.696285  5.684233 3.132494 3.485825 3.079335 7.193856 4.634104 4.006684
## 4 5.322044  7.033263 3.130982 3.448732 3.431294 7.572971 4.804074 3.863366
## 5 5.758023  7.214689 3.049585 3.073087 2.932586 8.393873 5.827125 3.611079
##      PROP1
## 1 3.184891
## 2 2.951848
## 3 3.047507
## 4 3.437736
## 5 2.975095
```

In order to train a model for each drug, I used the Random Forest method with 5 fold cross validation. I used the *ranger* function in the *caret* package. First, I train the models as regression. The following shows the top 10 and bottom 10 models trained for each drug ranked by their r2 scores. The trained models are named “ranger_” followed by the DRUG_ID.

```
##    ranger_index r2_scores
## 1    ranger_182 0.7394169
## 2    ranger_136 0.7185944
## 3    ranger_170 0.6888964
## 4    ranger_200 0.6827337
## 5   ranger_1149 0.6700780
## 6    ranger_268 0.6226356
## 7    ranger_157 0.6059950
## 8   ranger_1012 0.5950195
## 9    ranger_274 0.5536906
## 10   ranger_165 0.5453891
```

```
##     ranger_index   r2_scores
## 256  ranger_1023  0.11924071
## 257    ranger_52  0.11216735
## 258  ranger_1072  0.10221931
## 259   ranger_202  0.10114799
## 260  ranger_1219  0.09616221
## 261  ranger_1042  0.09515936
## 262   ranger_185  0.09383000
## 263  ranger_1091  0.07928832
## 264  ranger_1029  0.05864470
## 265  ranger_1166 -0.35012756
```

It looks the drug response models with the highest r2 score are the ones that were trained on fewer cell lines. But the trained model r2 values are rather random when the model was trained on at least ~45 cell lines. The full list of r2 values of all models and the corresponding number of cell lines the model is trained on can be found in “rf_regres_ranked2.txt” file.

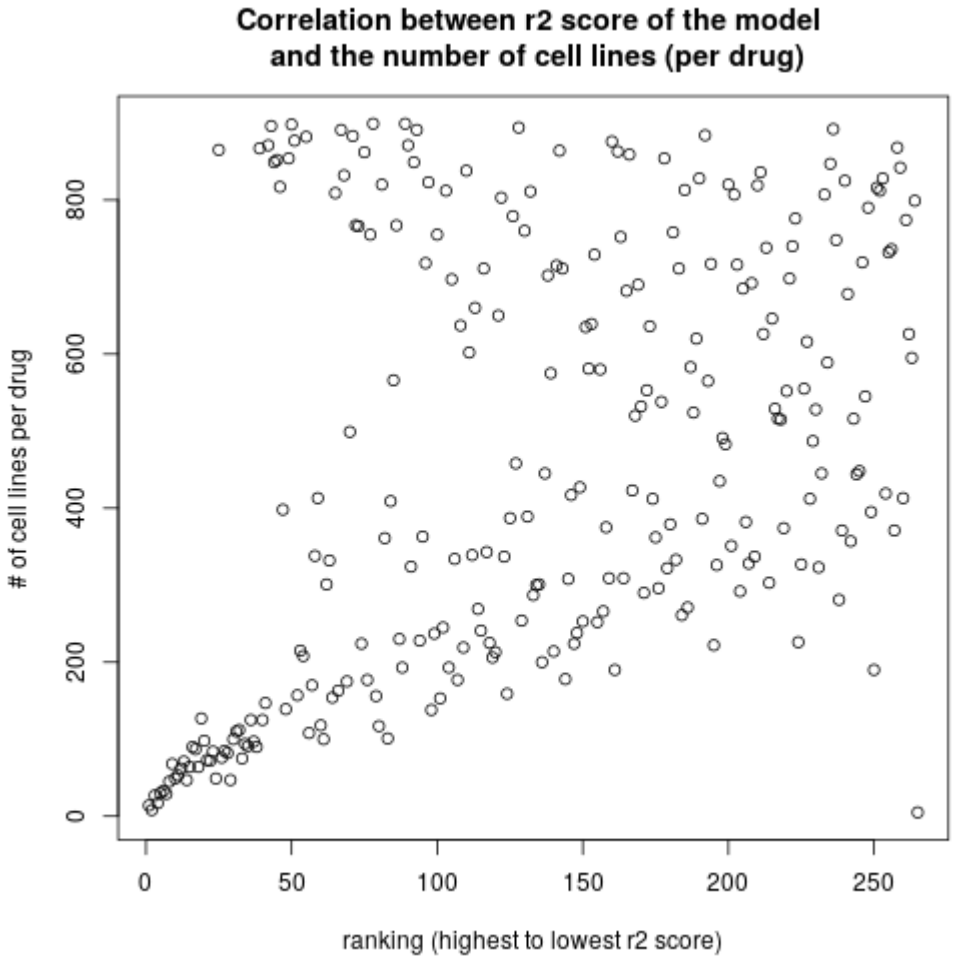

Figure 2: Cell line with the highest r2 score are those which were trained on fewer cell lines

We can also approach the drug response problem as a classification problem, where the cell lines are grouped into “sensitive”, “intermediate”, and “resistant” based on the LN_IC50 values for each drug. This is achieved by z-transforming the LN_IC50 values and ranking the cell lines based on their z-scores. I initially set a threshold of -2/2 for the z-score to assign “sensitve” and “resistant” labels to the cell lines per drug. But this cut-off was too

stringent, and most drugs did not have any sensitive or resistant cell lines due to skewed distribution. This resulted in very unbalanced classes. So I lowered the cut-off of the z-score to -1/1. Cell lines with z-score lower than -1 were deemed “sensitive” to the drug, and those higher than 1 were deemed “resistant”.

After this process, I retrained the random forest models with these three classes as the outcome. I used the same initial 30 features for each drug (I essentially just turned the continuous LN_IC50 distribution into 3 classes within the same feature matrix). The following shows the top 10 and bottom 10 models trained for each drug ranked by their F1 scores. The trained models are named “ranger2_” followed by the DRUG_ID.

```
##    ranger2_index F1_scores
## 1    ranger2_136 0.9230769
## 2    ranger2_157 0.9130435
## 3    ranger2_165 0.8860759
## 4    ranger2_104 0.8648649
## 5     ranger2_83 0.8533333
## 6    ranger2_205 0.8496815
## 7    ranger2_200 0.8461538
## 8    ranger2_312 0.8439153
## 9    ranger2_332 0.8409556
## 10  ranger2_1149 0.8372093
```

```
##     ranger2_index F1_scores
## 256   ranger2_170 0.7317073
## 257   ranger2_235 0.7306502
## 258    ranger2_60 0.7284768
## 259   ranger2_177 0.7280576
## 260     ranger2_5 0.7260726
## 261   ranger2_302 0.7257618
## 262   ranger2_301 0.7153729
## 263    ranger2_41 0.7125000
## 264   ranger2_225 0.7083333
## 265   ranger2_167 0.7000000
```

Classification approach seems more robust to the effect of the number of cell lines the models are trained on.

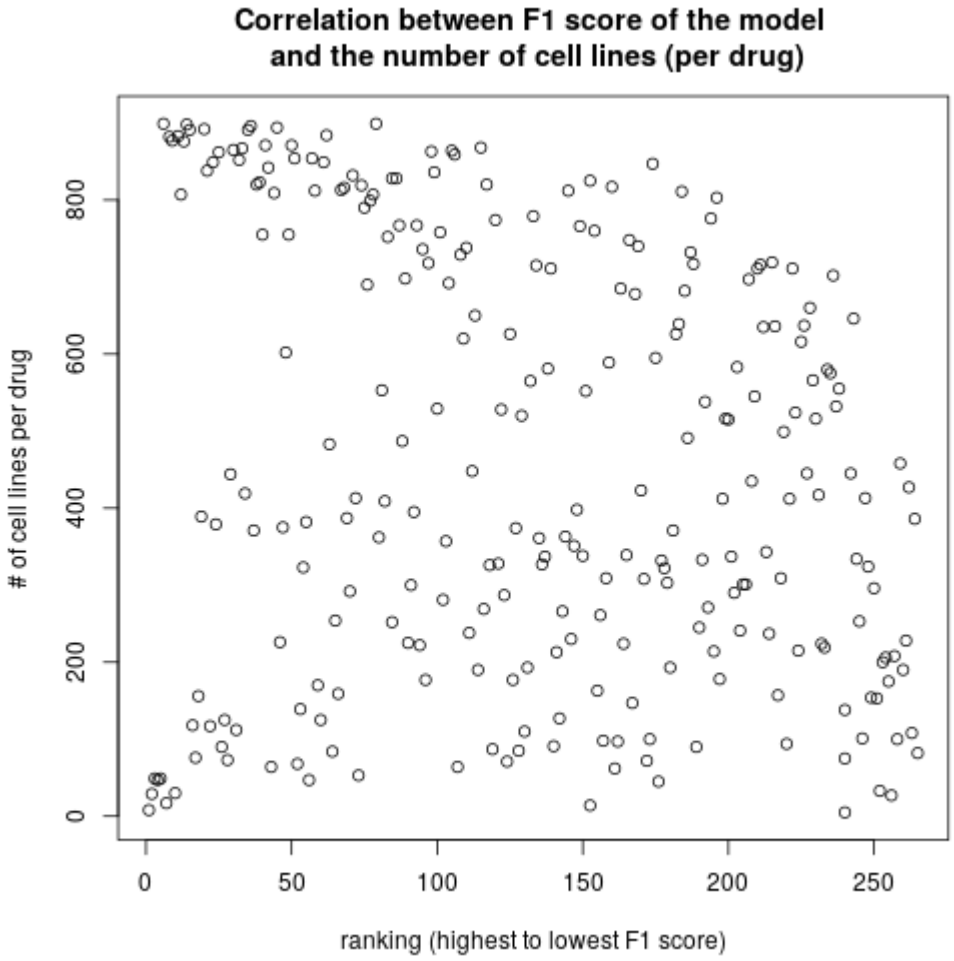

Figure 3: There is no obvious pattern between F1 score distribution and the number of cell lines on which the models were trained

Part 2

Task: Try to improve the results of “by Drug” prediction by adding additional assay types to your model.

The efficacy of the drugs can vary based on the mutational landscape of the cell lines. Therefore, I wanted to incorporate mutational information to the feature matrices. I used the ‘WES_variants.tsv’ file to access the list of mutations in each cell line. I wanted to look at most frequently mutated genes across all cell lines (regardless of the type of the mutation).

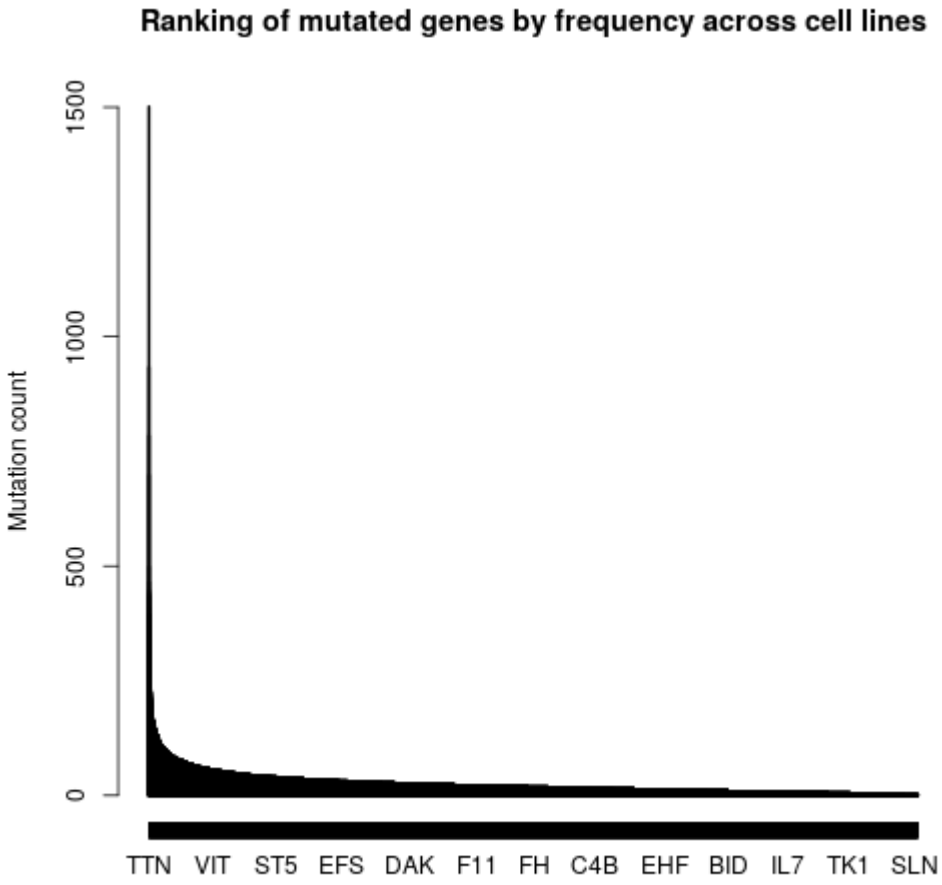

Figure 4: There are 403 genes that are mutated more than 100 times in total across all cell lines

A cut off of 100 counts for mutation frequency results in 403 genes that are most frequently mutated across all cell lines. I used these genes and made a feature matrix for each cell line where the gene took a value of “1” if it is mutated, and “0” if it is not. I limited the number of genes I used for the feature matrix to the top 50. I added these mutational gene features to the original 30 features and made a matrix of 80 features per drug. The following is an example of the new feature matrix for Drug 1 with mutational information.

```
##     COSMIC_class  LN_IC50_z COSMIC_ID LN_IC50     CBFB     TAF8    ROBO1
## 829 intermediate -0.3641781    683665    2.44 7.381675 4.046006 3.865636
## 57  intermediate  0.8630691    684055    3.34 7.073084 3.971642 5.308128
## 599    resistant  1.1766989    684057    3.57 6.924102 4.049746 4.876562
##        ZNRD1   KHDRBS3   KCNMA1   STAT5B    MRPL16     ELF1     CD38
## 829 9.077781  6.560701 7.031678 5.440448 10.025188 7.735772 9.591007
## 57  8.938284 10.391483 3.688912 3.992091  8.419572 5.446713 2.731800
## 599 8.873273 10.617343 3.273957 4.583204  9.213105 5.002667 2.859110
##       GAS2L3    RAB2A     MGST3    TRBC2     X.14 FAM177A1    TIAL1
## 829 5.282070 6.448660  8.183073 9.942958 6.816912 5.689164 8.939407
## 57  8.751107 8.099281  9.982433 3.080555 3.327946 6.606224 8.315006
## 599 8.221462 8.153698 11.663702 3.193664 6.200406 6.926580 8.226353
##        CCND3   MAGED2   ANP32B   NEDD4L   MAPK14  KIAA0922  FAM129C
## 829 4.733044 5.574793 10.71191 4.481254 5.164008 10.055993 3.392460
## 57  3.683693 8.522961 11.01311 4.928425 4.743144  6.599129 3.078927
## 599 3.818064 9.086599 10.18925 5.453939 5.696285  5.684233 3.132494
##       CCDC69    GNA15   ELAVL1   ABHD12     HHEX    PROP1 TTN_mut
## 829 5.744546 5.300199 7.459808 5.402310 3.379929 3.184891       0
## 57  3.262694 3.156875 7.762851 5.558390 3.085935 2.951848       0
## 599 3.485825 3.079335 7.193856 4.634104 4.006684 3.047507       0
##     MUC16_mut TP53_mut MUC4_mut HYDIN_mut OBSCN_mut SYNE1_mut LRP1B_mut
## 829         0        0        1         1         0         0         0
## 57          0        1        1         0         0         0         1
## 599         0        1        0         1         1         0         0
##     USH2A_mut FLG_mut RYR2_mut NEB_mut PCLO_mut CSMD3_mut MLL2_mut
## 829         0       0        0       0        0         1        1
## 57          0       0        0       0        0         0        0
## 599         0       0        0       0        0         0        0
##     CSMD1_mut GPR98_mut MUC5B_mut AHNAK2_mut FAT3_mut ZFHX4_mut HMCN1_mut
## 829         0         0         0          0        0         0         0
## 57          0         0         0          0        0         0         0
## 599         0         0         0          0        0         0         0
##     FAT4_mut XIRP2_mut DST_mut MUC12_mut DNAH17_mut APOB_mut DNAH11_mut
## 829        0         0       0         0          0        1          1
## 57         0         0       0         0          0        0          1
## 599        0         0       0         0          0        0          0
##     DNAH5_mut LRP2_mut RYR3_mut ABCA13_mut DNAH9_mut MUC17_mut PKHD1L1_mut
## 829         1        0        0          0         0         0           0
## 57          0        0        0          0         0         0           0
## 599         0        0        0          0         0         0           0
##     MACF1_mut SYNE2_mut DNAH8_mut RYR1_mut AHNAK_mut PCDH15_mut DNAH10_mut
## 829         0         0         0        0         0          0          0
## 57          0         0         0        1         0          0          0
## 599         0         0         0        0         0          0          0
##     DNAH6_mut RELN_mut DNAH7_mut SPTA1_mut PKHD1_mut PLEC_mut CDH23_mut
## 829         0        0         0         0         0        0         0
## 57          0        0         0         0         0        0         0
## 599         0        0         0         1         0        0         0
```

After making the new feature matrices for each drug, I trained a new random forest model for each drug either as a regression model (described as “ranger3_” series), or as a three-class model (described as “ranger4_” series). Unfortunately, side by side comparison of the r2 and F1 scores demonstrated no improvement on the model upon addition of the new features.

```
##     DRUG_ID r2_score_1 r2_score_2
## 139     182  0.7394169  0.5994950
## 97      136  0.7185944  0.5616976
## 129     170  0.6888964  0.5505240
## 150     200  0.6827337  0.6747121
## 67     1149  0.6700780  0.6161329
## 188     268  0.6226356  0.5794772
## 120     157  0.6059950  0.5312908
## 12     1012  0.5950195  0.5771313
## 193     274  0.5536906  0.5441594
## 125     165  0.5453891  0.5396366
```

```
##     DRUG_ID F1_score_1 F1_score_2
## 97      136  0.9230769  0.9230769
## 120     157  0.9130435  0.9090909
## 125     165  0.8860759  0.8461538
## 37      104  0.8648649  0.8611111
## 258      83  0.8533333  0.8266667
## 155     205  0.8496815  0.8470125
## 150     200  0.8461538  0.8000000
## 225     312  0.8439153  0.8465680
## 232     332  0.8409556  0.8443824
## 67     1149  0.8372093  0.8372093
```

I also tried other regression models such as **boosted generalized linear model (glmboost)** and **k-nearest neighbors (knn)**, but random forest remained the best training method based on the r2 score. Below are two example comparisons of the three methods used to train Drug 1 and Drug 182 models.

```
##
## Call:
## summary.resamples(object = resamps_1)
##
## Models: rf, glmboost, knn
## Number of resamples: 5
##
## MAE
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf       0.4642431 0.4821795 0.5395407 0.5301879 0.5571453 0.6078310    0
## glmboost 0.5141082 0.5232699 0.5357493 0.5348398 0.5380512 0.5630203    0
## knn      0.5511248 0.5579573 0.5696181 0.5654715 0.5726431 0.5760141    0
##
## RMSE
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf       0.5704937 0.6121446 0.6625377 0.6704955 0.7055391 0.8017625    0
## glmboost 0.6475811 0.6480205 0.6997232 0.6827666 0.7002762 0.7182319    0
## knn      0.6970954 0.7133951 0.7149637 0.7177359 0.7249697 0.7382555    0
##
## Rsquared
##                Min.    1st Qu.     Median       Mean   3rd Qu.      Max.
## rf       0.04657079 0.13137216 0.16763489 0.17341068 0.1938671 0.3276085
## glmboost 0.08097540 0.08365524 0.15364580 0.13742592 0.1643758 0.2044774
## knn      0.01982646 0.07812702 0.09663855 0.08751162 0.1142537 0.1287123
##          NA's
## rf          0
## glmboost    0
## knn         0
```

```
##
## Call:
## summary.resamples(object = resamps_182)
##
## Models: rf, glmboost, knn
## Number of resamples: 5
##
## MAE
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf       0.2632232 0.4578254 0.4847107 0.4965171 0.5098599 0.7669661    0
## glmboost 0.3443181 0.6225460 0.7084123 0.6675006 0.7446083 0.9176183    0
## knn      0.3990000 0.4913333 0.6400000 0.7052667 0.7386667 1.2573333    0
##
## RMSE
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## rf       0.2785252 0.5696025 0.6434937 0.5876256 0.6750007 0.7715059    0
## glmboost 0.4210692 0.6983561 0.7476708 0.7355427 0.8348800 0.9757371    0
## knn      0.3990013 0.4975473 0.9000193 0.8750691 1.1113907 1.4673868    0
##
## Rsquared
##                Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## rf       0.03344118 0.9244962 0.9964317 0.7904375 0.9978183    1    0
## glmboost 0.06697424 0.8652658 0.9106281 0.7678282 0.9962729    1    0
## knn      0.04463000 0.3623122 0.5915290 0.5977061 0.9900595    1    0
```

Therefore, for the rest of the assingment (PART 3), I used the initial random forest regression models for each drug.

Part 3

Task: Develop a version of your model that can rank order the drugs for a given cell line

Using *predict()* function in the *caret* package, I predicted the LN_IC50 value of all drugs on all of the cell lines based on my models. Then I compared these predicted values to the experimental values side by side. To make a statistical comparisons between the two ranked lists of LN_IC50 values, I performed Spearman ranked correlation test for each drug. The files containing the comparisons of the predicted vs experimental values for each drug are in the “compare” folder. This folder also has a file called “spearman_summary.txt” which has the rho statistic and the p-value for the overall correlation between predicted and experimental values per drug.

The following is an example of the predicted (ic50_pred) vs. experimental (LN_IC50) IC50 values for the commonly-used breast cancer cell line MDA-MB-231 (COSMIC_ID: 905960)

```
##    DRUG_ID  ic50_pred COSMIC_ID LN_IC50
## 1      197 -2.3850160    905960   -2.11
## 2      283 -0.9329003    905960   -1.28
## 3      208 -0.8837393    905960   -1.58
## 4     1261 -0.4179843    905960   -0.48
## 5     1230  0.1273443    905960    0.26
## 6      312  0.4512700    905960    0.45
## 7     1008  0.8675560    905960    1.09
## 8      155  0.8748993    905960    0.96
## 9     1264  1.2279720    905960    0.83
## 10    1032  1.5465193    905960    1.60
## 11    1010  1.5771300    905960    1.85
## 12     202  1.5810407    905960    1.18
## 13    1026  1.7856283    905960    2.33
## 14    1046  1.8272627    905960    1.56
## 15    1527  1.9323517    905960    1.89
```

```
##
##  Spearman's rank correlation rho
##
## data:  mb231_compare$ic50_pred and mb231_compare$LN_IC50
## S = 10998, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9754282
```

Not all drugs have been tested on MDA-MB-231 cell line (or the drug-cell line pair did not pass the AUC tresholding we performed in Part 1), but since we have a predictive model for each drug, we can predict which other drugs 231 cells are predicted to be sensitive to.

```
##    ranger_index  ic50_pred
## 1   ranger_1007 -2.9408103
## 2    ranger_197 -2.3850160
## 3    ranger_104 -2.0585390
## 4    ranger_201 -1.7403480
## 5   ranger_1494 -1.4483140
## 6    ranger_140 -1.4277190
## 7   ranger_1004 -1.3201760
## 8    ranger_283 -0.9329003
## 9    ranger_208 -0.8837393
## 10  ranger_1016 -0.6524417
## 11  ranger_1003 -0.5735150
## 12  ranger_1261 -0.4179843
## 13  ranger_1057 -0.2226937
## 14  ranger_1031 -0.1016347
## 15     ranger_3  0.0794010
```

Table above demonstrates that MDA-MB-231 cells are predicted to be also sensitive to the drugs 1007, 104, 201, 1494, 140, and 1004.