Mechanism of Action (MoA) Prediction

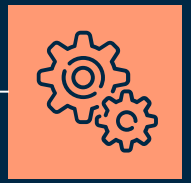
Yian Ding, Haoyu Wang

TABLE OF CONTENTS

01

PROBLEM STATEMENT & EDA

To understand the scope of the problem, we start our analysis with EDA



02

METHODS

Methods we tried to find the best algorithm

03

RESULTS & Discussion

How well are the algorithms and what to do next

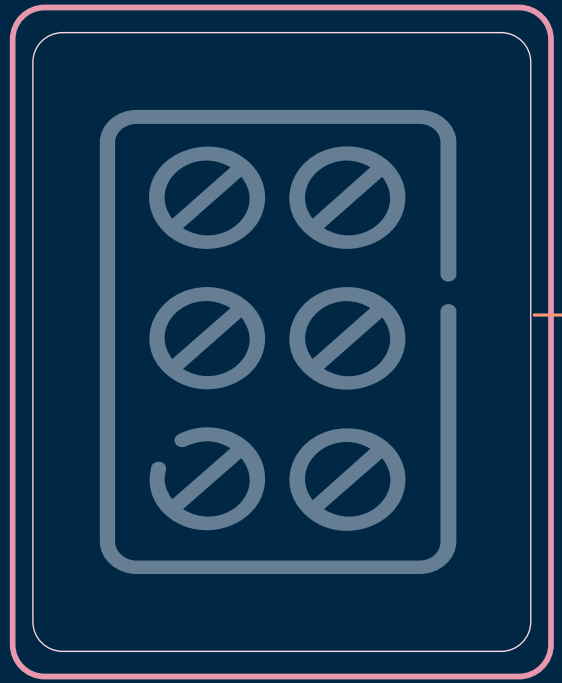
Introduction

Traditional:

The discovery of drugs in the past was usually inspired by function of natural products or experience of traditional remedies, which left us the mechanism of how drugs worked behind.

Now:

MoA (mechanism of actions) as a shorthand to describe the biological activity of a given molecule. One way to learn the MoA of a drug is to analyze and compare the response in treated human cells with MoAs of known drugs by algorithms.



UNDERSTANDING THE PROBLEM

Different types of cells have different responses to drugs.

Analyze all the responses (cell viability data and gene expression data) to different drugs under different treatments.

Find mechanisms of action for the given drugs, e.g. which specific molecular targets to which the drug binds

Help or Speed up the progress of new drug development

train_features.csv - Features for the training set

train_drug.csv - This file contains an anonymous drug_id for the training set only.

train_targets_scored.csv - The binary MoA targets that are scored.

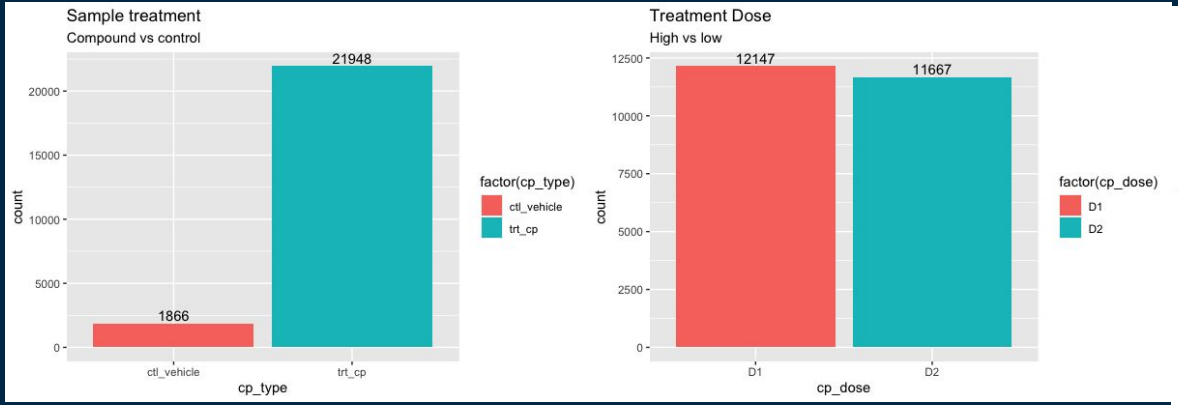
train_targets_nonscored.csv - Some additional binary MoA responses for the training data.

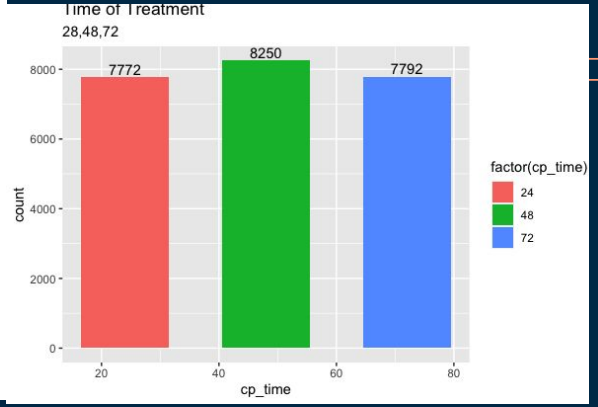
test_features.csv - Features for the test data.

sample_submission.csv - A submission file in the correct format.

EDA Analysis of the features by visualization

Categorical Variables





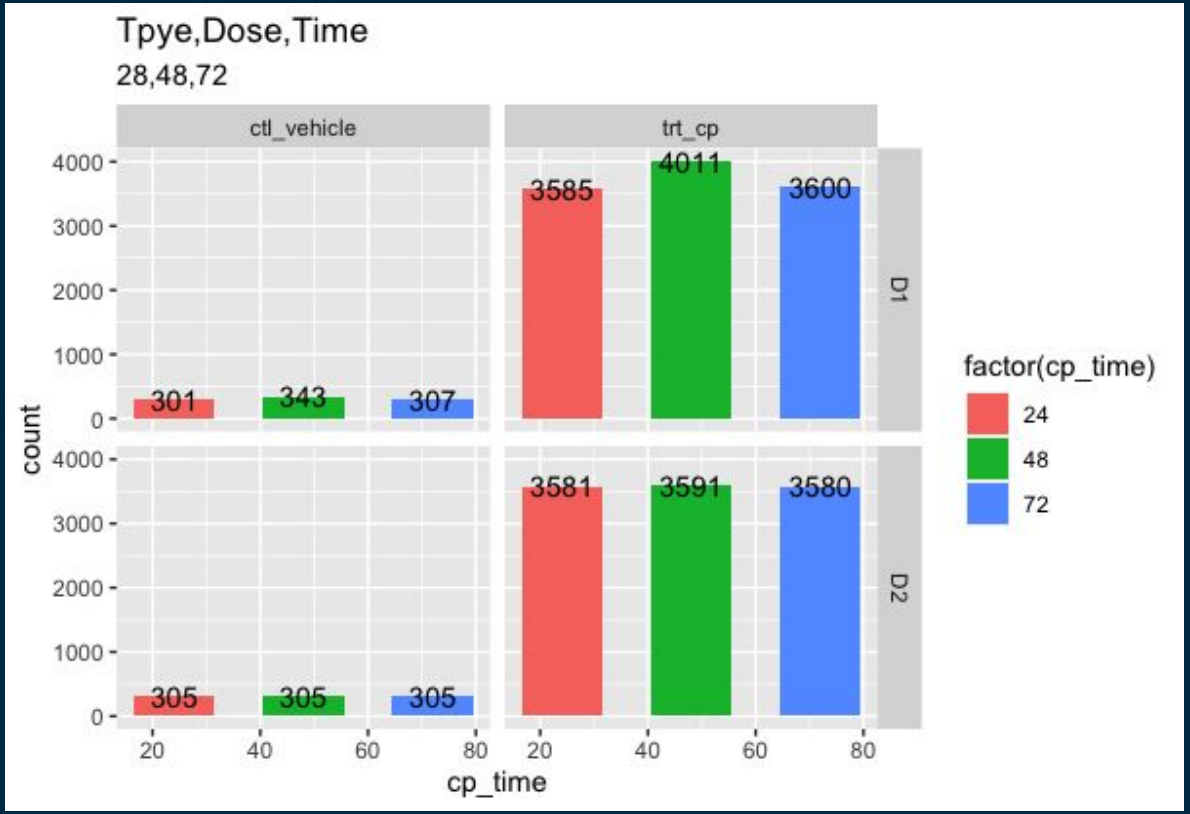
Continuous Variables

Finding the number of cell and gene response features

```
#Find the number of G and C:
#722
Train_list %>%
  select(starts_with("g."))%>%
  ncol()
```

```
#100
Train_list %>%
  select(starts_with("c."))%>%
  ncol()
```

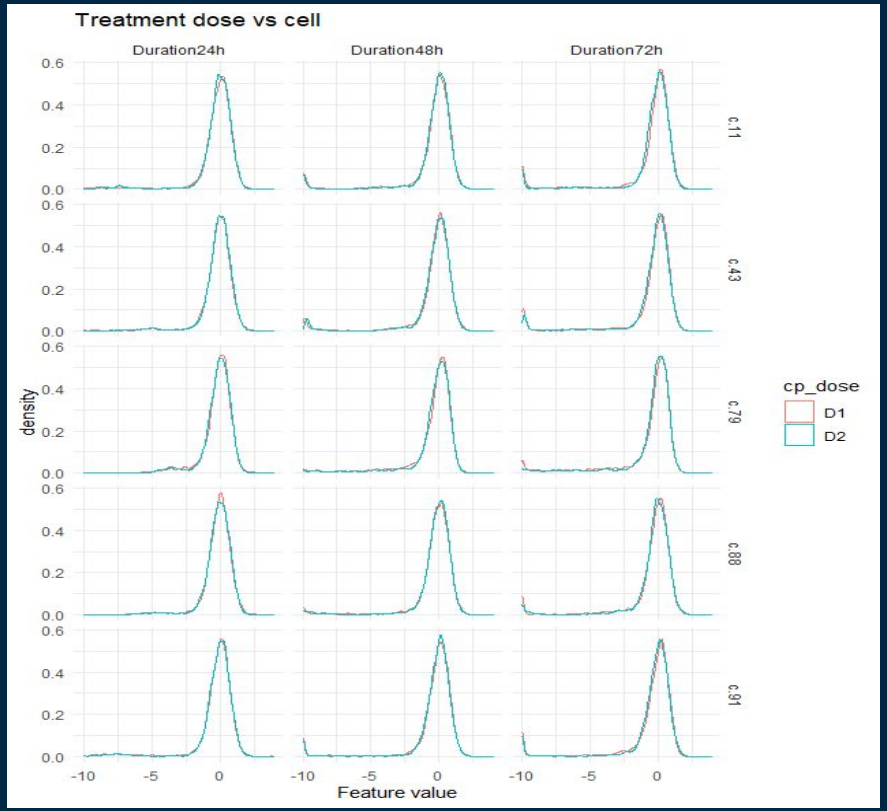
Interaction Visualizations

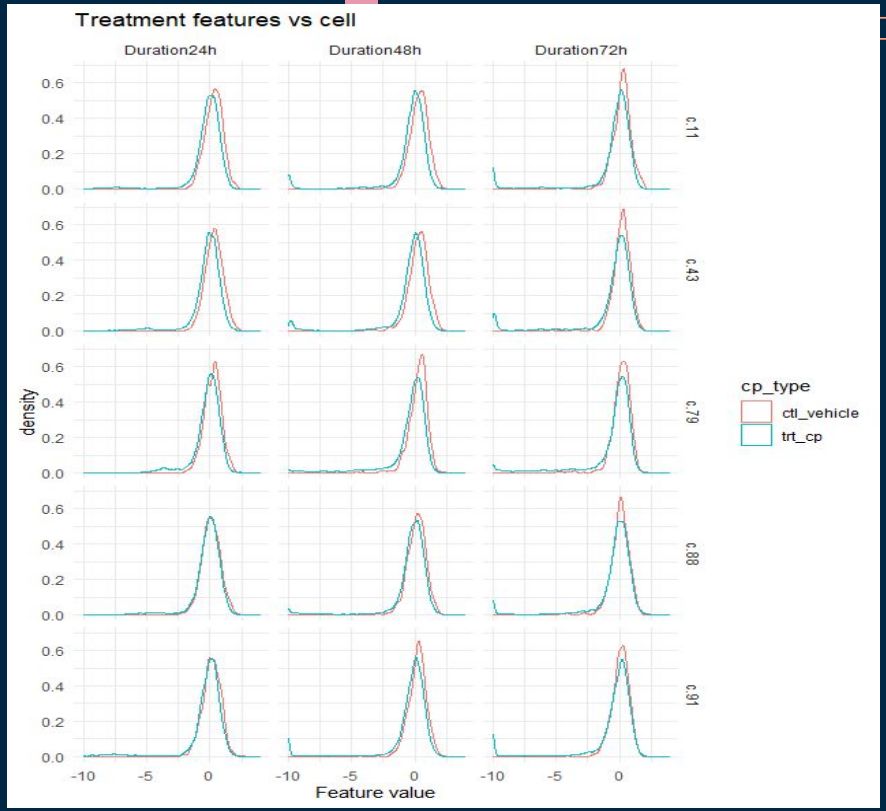


1.Well balanced

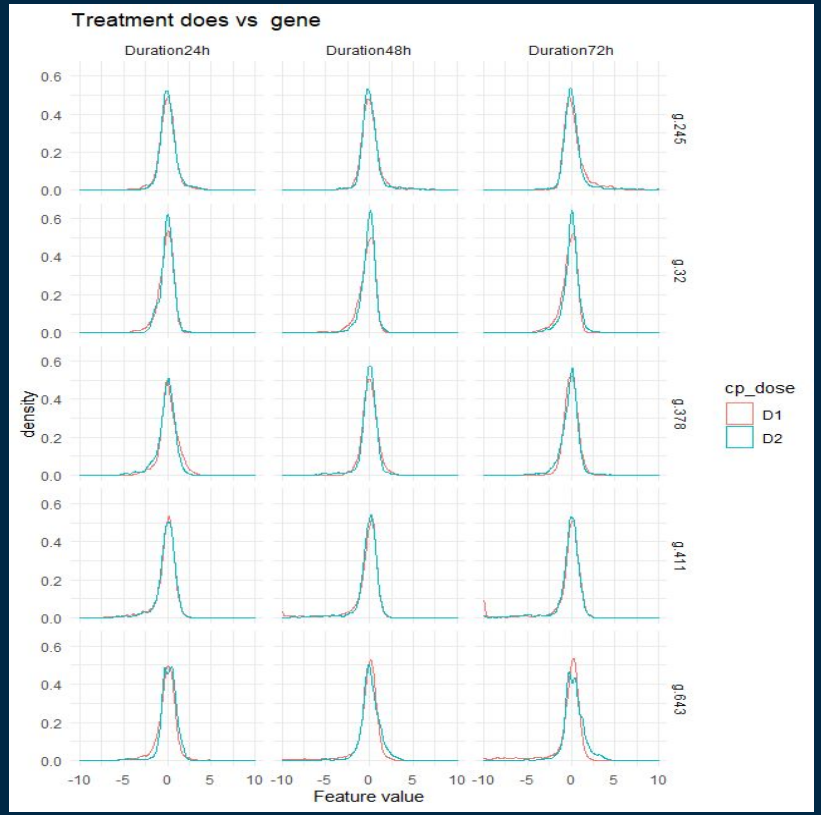
2. 48 hours is a little bit higher than other 2 groups in D1(cp_dose).

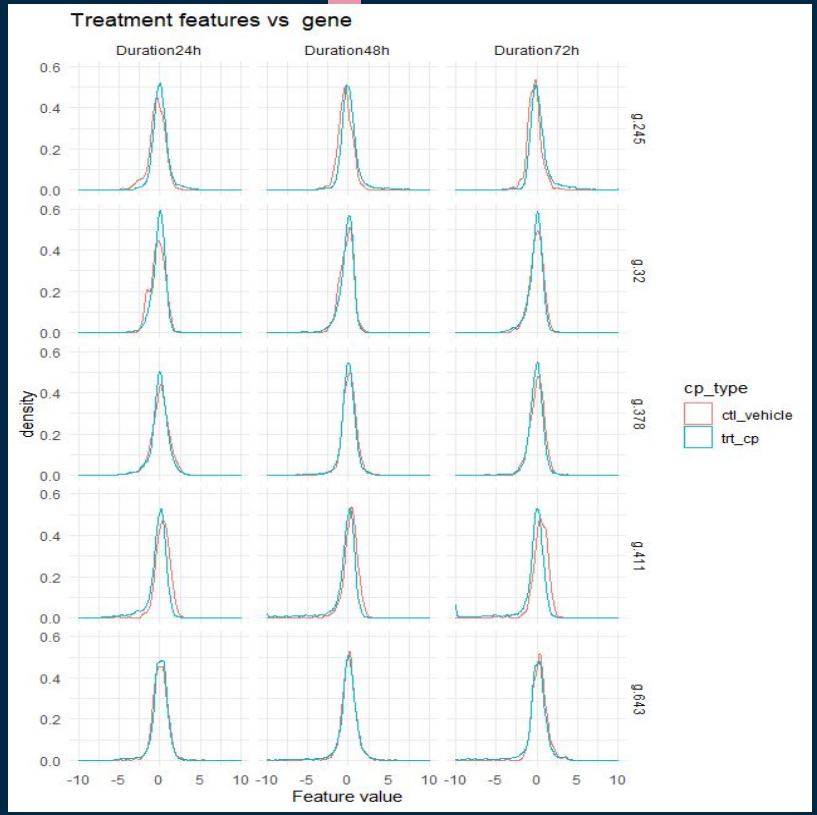
Analysis : cell response



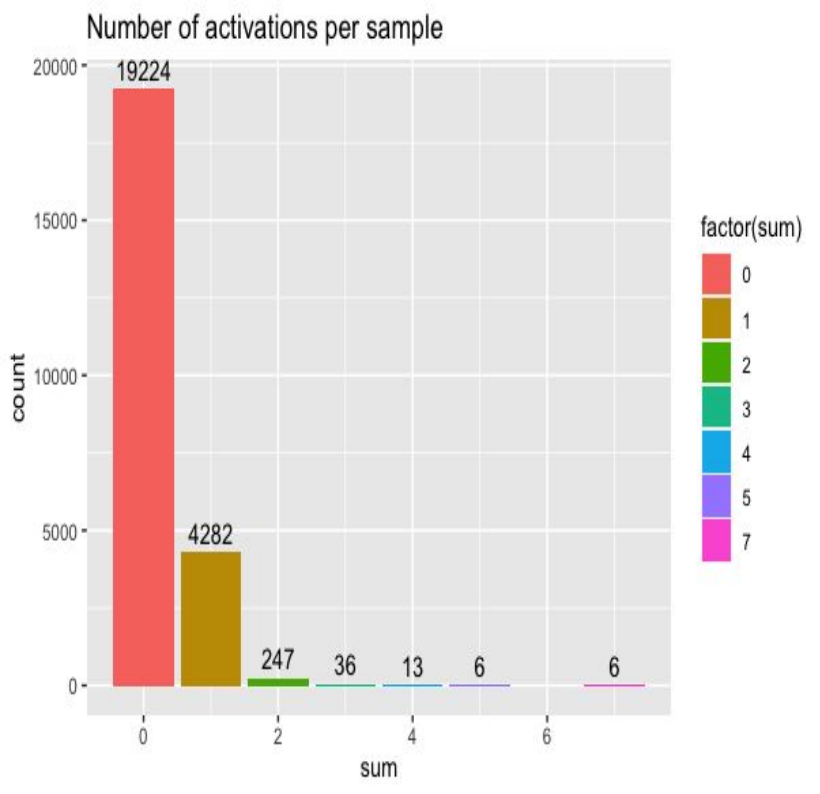


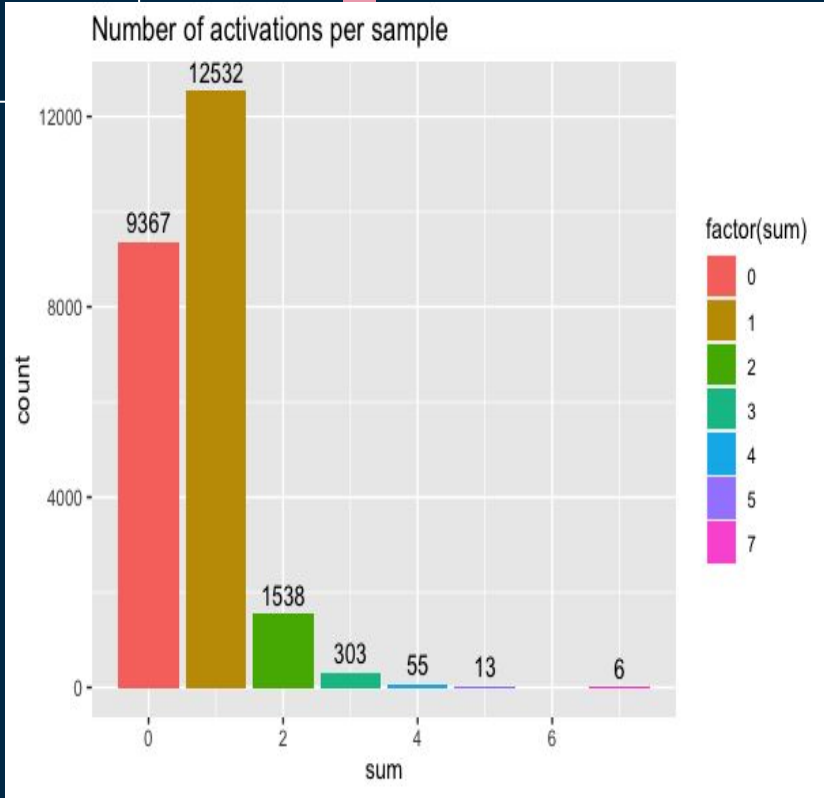
Analysis : gene response



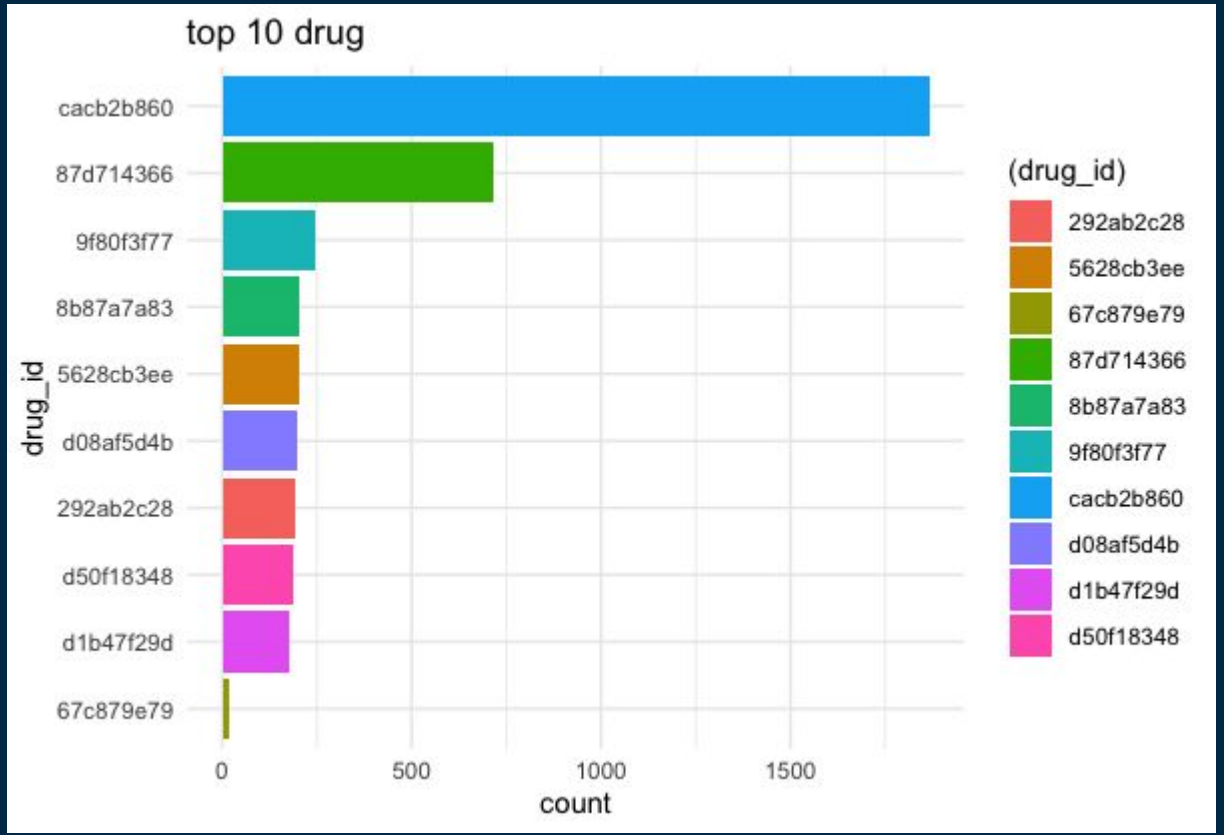


None-scored and scored targets:





Top ten drug ID



Data Preprocessing

- Removing cp_type

```
cp_type
ctl_vehicle         0
trt_cp          16844
dtype: int64
```

- Encoding

```
train['cp_time'] = train['cp_time'].map({24:0, 48:1, 72:2})
train['cp_dose'] = train['cp_dose'].map({'D1':0,'D2':1})
```

- Common stats features

```
df['g_mean'] = df[features_g].mean(axis = 1)
df['g_std'] = df[features_g].std(axis = 1)
df['c_mean'] = df[features_c].mean(axis = 1)
df['c_std'] = df[features_c].std(axis = 1)
df['gc_mean'] = df[features_g + features_c].mean(axis = 1)
df['gc_std'] = df[features_g + features_c].std(axis = 1)
```

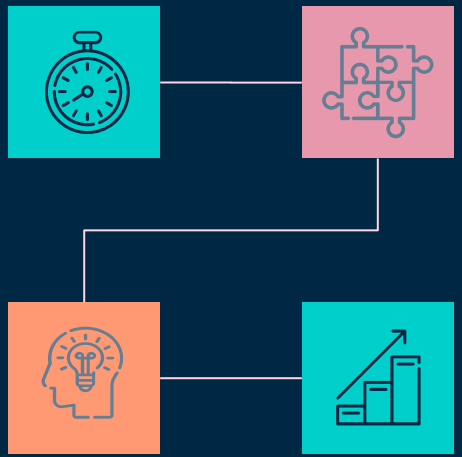
Methods

Principal Component Analysis

Extract principal components from the features

Up/Downsampling and MLSMOTE

Ways to handle class imbalance



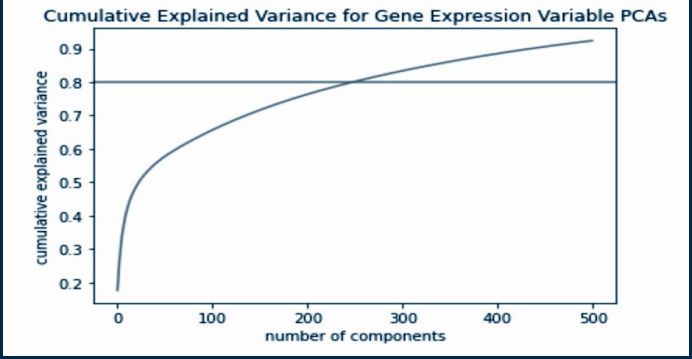
Vanilla Classifiers

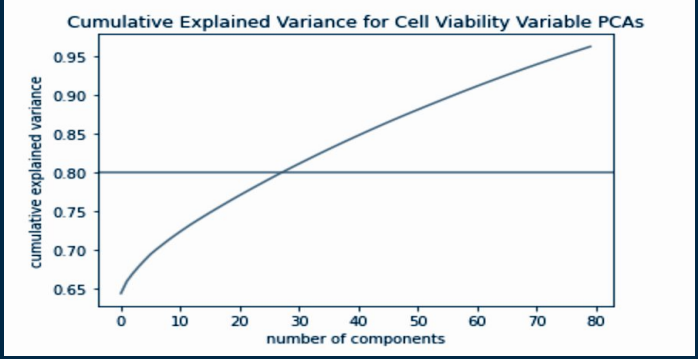
Logistic Regression, KNN and SVM

Logistic Re-Run & Neural Networks

Sensitive to class imbalance

Principal Component Analysis





	cp_time	cp_dose	pca_c1	pca_c2	pca_c3	pca_c4	pca_c5	pca_c6	pca_c7	...	pca_g231	pca_g232	pca_g233	pca_g234
0	0	0	-4.817148	1.062926	-0.589636	-0.575730	-1.081031	-0.548887	-0.472183	...	-0.223639	0.133575	-0.871727	0.101917
1	2	0	-4.997670	-0.215408	-0.167323	0.232766	-0.274137	-0.077560	-0.286073	...	-0.277424	-0.404145	0.019088	-0.091105
2	1	0	0.045162	0.530170	0.417546	-0.039593	0.172192	-0.004675	-0.399574	...	0.663110	0.117897	-0.097634	-1.115766
3	1	0	14.455510	5.272821	-1.244755	3.424090	-2.339055	0.635806	0.409224	...	-1.204965	1.321544	-0.513018	-0.844421
4	2	1	-3.931571	0.596221	0.255306	-0.510133	0.111545	0.150078	0.163462	...	0.256454	1.235980	-0.273024	-0.274744
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
23809	0	1	-3.921035	-0.098002	-0.460143	1.025173	0.441582	0.137145	-0.885680	...	-0.633394	-0.556416	1.185370	-0.453098
23810	0	1	-1.648615	-0.030032	-0.499233	-2.038884	0.518411	0.671420	-0.300262	...	-0.892414	0.609377	0.745017	0.760582
23811	1	1	-5.895690	0.718708	-0.829521	0.271426	-0.187698	-0.408464	-0.029757	...	1.165753	0.014225	0.771122	-0.840293
23812	0	0	-5.389047	0.856159	-0.879516	-0.391346	0.937524	-0.979720	1.128265	...	-0.134373	-0.348500	-0.969733	-0.336896
23813	2	0	20.052371	1.379439	1.238830	0.002209	-0.789444	1.398503	0.354249	...	-0.718479	0.270305	0.180601	0.916041

23814 rŋ9 columns

Vanilla Classifiers

KNN

```
**Processing acat_inhibitor comments...**
Test accuracy is 0.9992365440895788


**Processing acetylcholine_receptor_agonist comments...**
Test accuracy is 0.9912202570301565


**Processing acetylcholine_receptor_antagonist comments...**
Test accuracy is 0.9870212495228401


**Processing acetylcholinesterase_inhibitor comments...**
Test accuracy is 0.9966916910548416
```

OneVsRestClassifier

Logistic Regression

SVM

	Neg Pred	Pos Pred
Neg	7819	4
Pos	15	21

```
**Processing topoisomerase_inhibitor comments...**
Test accuracy is 0.9975823896169996
[[[  21   15]
  [   4 7819]]

 [[7819    4]
  [  15   21]]]
```

Precision: 0.998
Recall:0.999
Specificity:0.583

```
**Processing tubulin_inhibitor comments...**
Test accuracy is 0.9931288968062094
[[[  77   25]
  [  29 7728]]

 [[7728   29]
  [  25   77]]]
```

Precision: 0.996
Recall:0.996
Specificity:0.754

Up/Downsampling and MLSMOTE

	cp_time	cp_dose	pca_c1	pca_c2	pca_c3	pca_c4	pca_c5	pca_c6	pca_c7	pca_c8	pca_c9	pca_c10	pca_c11	pca_c12
0	1.000000	1.000000	3.439526	-0.726461	-0.158161	-0.744945	-0.712607	-0.075173	0.028076	0.508854	-0.475984	0.495147	-1.062269	-0.213855
1	2.000000	1.000000	-3.684235	-1.086398	-2.023943	-0.457292	-0.271369	1.213485	-1.227390	0.855555	0.604156	0.111843	0.139094	0.317084
2	2.000000	1.000000	-4.307215	-0.109936	-1.343393	-0.674654	0.043918	0.531913	0.568216	0.005639	-0.270809	-1.095907	0.932624	-1.042662
3	1.000000	1.000000	-3.903950	0.522946	-0.553465	0.575841	0.006986	0.850557	-0.642691	0.038502	-0.119876	-1.236553	-0.701238	0.268676
4	2.000000	1.000000	0.450329	0.787666	-0.059178	-0.111205	-0.851312	-0.408913	-0.561995	0.312690	0.025304	0.965447	-0.293368	0.556461
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	2.000000	-0.433722	-5.522990	-0.020259	-0.628433	-1.656065	-1.093951	0.406229	1.258172	0.241721	0.017310	-0.435397	-0.184772	0.026642
496	1.804523	1.000000	-11.790293	-2.393449	1.922091	-0.751703	-0.148726	0.579454	0.745568	-0.395502	0.484121	-0.990524	-1.482852	1.884888
497	0.699396	1.300604	-5.959088	0.206346	0.146853	1.348327	0.336444	-0.560043	0.801891	-0.365594	0.111258	0.141591	0.736968	1.448356
498	2.000000	1.968220	14.539649	6.243049	-1.416208	-0.478707	-2.169988	0.979697	0.850433	-1.284779	0.212376	-4.586688	1.148587	-2.647151
499	3.348197	1.000000	-4.357524	0.087526	3.049018	2.724356	0.712479	-0.573096	0.675365	1.728983	0.597251	0.016495	1.360873	2.869278

1105 rows × 268 columns

	5-alpha_reductase_inhibitor	11-beta-hsd1_inhibitor	acat_inhibitor	acetylcholine_receptor_agonist	acetylcholine_receptor_antagonist	acetylcholinesterase_inhibitor
0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...
495	1.0	1.0	0.0	0.0	0.0	0.0
496	1.0	1.0	0.0	0.0	0.0	0.0
497	1.0	1.0	0.0	0.0	0.0	0.0
498	0.0	1.0	0.0	0.0	0.0	1.0
499	1.0	1.0	0.0	0.0	0.0	0.0

1105 rows × 206 columns

MLSMOTE

	cp_time	cp_dose	pca_c1	pca_c2	pca_c3	pca_c4	pca_c5	pca_c6	pca_c7	pca_c8	pca_c9	pca_c10
0	0.000000	0.000000	-4.817148	1.063611	-0.596078	-0.584336	-1.066888	-0.564306	-0.451856	0.129112	0.244541	-0.162973
1	2.000000	0.000000	-4.997670	-0.215591	-0.169544	0.227502	-0.282415	-0.078727	-0.295930	0.403514	-0.024671	-1.626695
2	1.000000	0.000000	0.045162	0.530342	0.413078	-0.044963	0.157633	-0.029527	-0.382621	-0.311135	-0.527955	-0.716281
3	1.000000	0.000000	14.455510	5.272905	-1.236684	3.433563	-2.341201	0.625396	0.441758	1.723473	-0.530681	-0.177071
4	2.000000	1.000000	-3.931571	0.595677	0.257886	-0.513446	0.124833	0.139334	0.140106	-0.378799	-0.708521	-0.063927
...	...	...	...	...	...	...	...	...	...	...	...	...
24914	2.000000	-0.433722	-5.522990	-0.020259	-0.628433	-1.656065	-1.093951	0.406229	1.258172	0.241721	0.017310	-0.435397
24915	1.804523	1.000000	-11.790293	-2.393449	1.922091	-0.751703	-0.148726	0.579454	0.745568	-0.395502	0.484121	-0.990524
24916	0.699396	1.300604	-5.959088	0.206346	0.146853	1.348327	0.336444	-0.560043	0.801891	-0.365594	0.111258	0.141591
24917	2.000000	1.968220	14.539649	6.243049	-1.416208	-0.478707	-2.169988	0.979697	0.850433	-1.284779	0.212376	-4.586688
24918	3.348197	1.000000	-4.357524	0.087526	3.049018	2.724356	0.712479	-0.573096	0.675365	1.728983	0.597251	0.016495

24919 rows × 474 columns

Logistic Regression Re-Run

MLSMOTE dataset

```
**Processing tropomyosin_receptor_kinase_inhibitor comments...**
Test accuracy is 0.9995136186770428
[[[   4    4]
  [   0 8216]]

 [[8216    0]
  [   4    4]]]
```

Precision: 0.999
Recall:1
Specificity:0.5

```
**Processing tubulin_inhibitor comments...**
Test accuracy is 1.0
[[[ 107    0]
  [   0 8117]]

 [[8117    0]
  [   0  107]]]
```

Precision: 1
Recall:1
Specificity:1

Logistic Regression Re-Run

Original dataset

```
**Processing topoisomerase_inhibitor comments...**
Test accuracy is 0.9975823896169996
[[[  21   15]
  [   4 7819]]

 [[7819    4]
  [  15   21]]]
```

```
**Processing tubulin_inhibitor comments...**
Test accuracy is 0.9931288968062094
[[[  77   25]
  [  29 7728]]

 [[7728   29]
  [  25   77]]]
```

MLSMOTE dataset

```
**Processing tropomyosin_receptor_kinase_inhibitor comments...**
Test accuracy is 0.9995136186770428
[[[   4    4]
  [   0 8216]]

 [[8216    0]
  [   4    4]]]
```

```
**Processing tubulin_inhibitor comments...**
Test accuracy is 1.0
[[[ 107    0]
  [   0 8117]]

 [[8117    0]
  [   0  107]]]
```

Overall Precision, Recall and specificity

```
The overall (average) precision, recall and specificity are
0.968269291563848 0.9706157707022307 0.8009708737864077
```

```
0.9998651610739205 1.0 0.9089481103216491
```

Overall Precision, Recall and specificity with holdout data

```
**Processing tropomyosin_receptor_kinase_inhibitor comments...**
Test accuracy is 0.9997993579454254
[[[   5    1]
  [   0 4978]]

 [[4978    0]
  [   1    5]]]
```

```
**Processing tropomyosin_receptor_kinase_inhibitor comments...**
Test accuracy is 0.9995800965777871
[[[   0    2]
  [   0 4761]]

 [[4761    0]
  [   2    0]]]
```

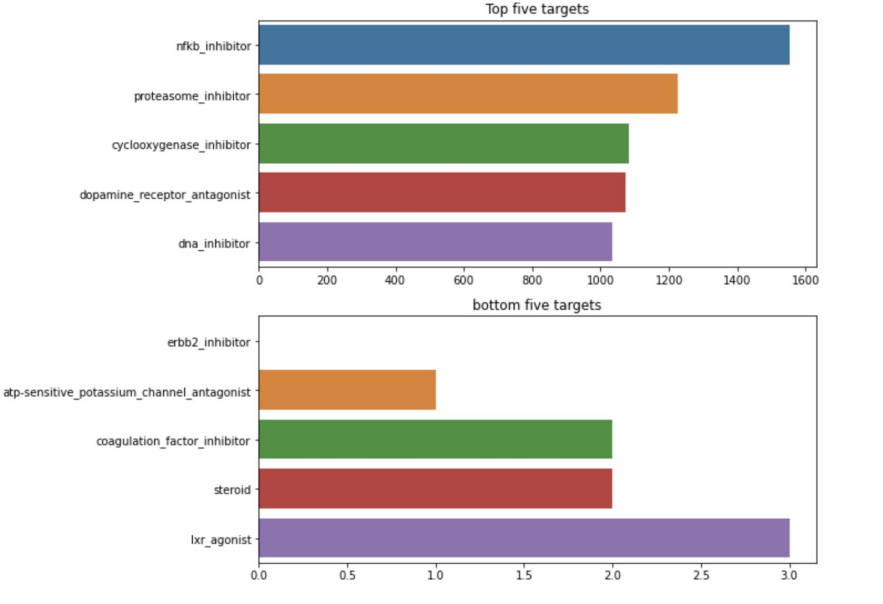
```
**Processing tubulin_inhibitor comments...**
Test accuracy is 1.0
[[[  61    0]
  [   0 4923]]

 [[4923    0]
  [   0   61]]]
```

```
**Processing tubulin_inhibitor comments...**
Test accuracy is 0.9934914969557002
[[[  38   19]
  [  12 4694]]

 [[4694   12]
  [  19   38]]]
```

The overall (average) precision, recall and specificity are
0.968269291563848 0.9706157707022307 0.10861951487738836



Overall Precision, Recall and specificity with holdout data (in the right way)

```
**Processing topoisomerase_inhibitor comments...**
Test accuracy is 0.9958009657778711
[[[  14   12]
  [   8 4729]]

 [[4729    8]
  [  12   14]]]
```

```
0.9682781511166875 0.9705741094108459 0.10412678328579397
```

```
**Processing tubulin_inhibitor comments...**
Test accuracy is 0.9934914969557002
[[[  44   13]
  [  18 4688]]

 [[4688   18]
  [  13   44]]]
```

Neural Networks

```
FOLD: 0, EPOCH: 19, valid_loss: 0.02322150158163692
FOLD: 0, EPOCH: 20, train_loss: 0.01916640222875658
FOLD: 0, EPOCH: 20, valid_loss: 0.02283122806277658
FOLD: 0, EPOCH: 21, train_loss: 0.018382588641238425
FOLD: 0, EPOCH: 21, valid_loss: 0.022782309479745372
FOLD: 0, EPOCH: 22, train_loss: 0.017598671676914493
FOLD: 0, EPOCH: 22, valid_loss: 0.022722477159862007
FOLD: 0, EPOCH: 23, train_loss: 0.017187189388864054
FOLD: 0, EPOCH: 23, valid_loss: 0.022958842538563267
```

	Neg Pred	Pos Pred
	[[7819	4]
	[15	21]]]

Precision: 0.998
Recall:0.999
Specificity:0.583

	Neg Pred	Pos Pred
	[[7728	29]
	[25	77]]]

Precision: 0.996
Recall:0.996
Specificity:0.754

	Neg Pred	Pos Pred
	[7822,	0]
	[11,	26]

Precision: 1
Recall:0.998
Specificity:0.702

	Neg Pred	Pos Pred
	[[7735,	5],
	[12,	107]]

Precision: 0.999
Recall:0.998
Specificity:0.899

AWESOME WORDS

THANKS