



## Mechanisms of Action (MoA) Prediction

班 级 统计 1803 班  
学 号 8201180805  
姓 名 王子正

## Acknowledgement

<https://www.kaggle.com/gunesevitan/mechanisms-of-action-moa-prediction-eda>

<https://www.kaggle.com/isaienkov/mechanisms-of-action-moa-prediction-eda>

<https://medium.com/swlh/drug-discovery-with-neural-networks-a6a68c76bb53>

<https://www.kaggle.com/kushal1506/moa-pytorch-feature-engineering-0-01846>

<https://www.ritchieng.com/machine-learning-dimensionality-reduction-feature-transform/>

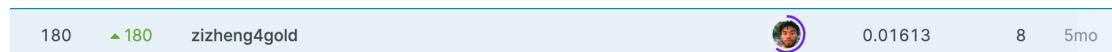
<https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html>

<https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html?highlight=bce#torch.nn.BCELoss>

<https://www.kaggle.com/thehemen/pytorch-transfer-learning-with-k-folds-by-drug-ids>

<https://www.kaggle.com/c/lish-moa/discussion/195195>

## 摘要



<https://www.kaggle.com/zizheng>

本文是 kaggle 竞赛 Mechanisms of Action (MoA) Prediction 比赛总结，这是我第一次参加 kaggle 竞赛，solo 参赛取得了银牌成绩 (180/4373), 参赛过程中参考了许多公开解法，已放在 Acknowledgement 部分。本文通过各种输入特征，如基因表达数据和细胞活力数据预测不同样本(sig\_id)的作用机制(MoA)反应的多个靶标概率，进行探索性数据分析，并采取了 quantile transformer, one-hot encoding, PCA,K-Means、提取统计量特征等方法进行特征工程。建立了三个模型，其中模型一通过 transfer learning 的方法对 pretrained 的 DNN 模型进行微调；模型二为 tabnet；模型三融合 4 个 DNN 模型和一个 Resnet 模型变体。最后融合这三个模型，在测试集得到的 log loss 为 0.016138。

## 目录

<b>1 项目背景</b>	<b>4</b>
1.1 什么是 MoA	4
1.2 如何确定新药物的 MOA	4
<b>2 数据介绍</b>	<b>4</b>
<b>3 探索性数据分析</b>	<b>5</b>
3.1 分类特征	5
3.2 数值特征	6
3.3 目标变量	9
3.4 目标变量与特征	13
3.5 DRUG_ID	15
<b>4 特征工程</b>	<b>15</b>
4.1 统计特征	17
4.2 特征处理总结	18
<b>5 评价标准及优化函数</b>	<b>18</b>
<b>6 模型</b>	<b>19</b>
6.1 训练/验证集划分	19
6.2 模型一	19
6.3 模型二	20
6.4 模型三	21
6.5 模型融合	21
<b>7 总结</b>	<b>23</b>
<b>REFERENCE</b>	<b>23</b>

# 1 项目背景

Connectivity Map 是由麻省理工学院和哈佛大学的 Broad 研究所、哈佛大学创新科学实验室 (LISH) 和美国国立卫生研究院 (NIH) 等共同发起的一个项目。该项目举办了这次 kaggle 竞赛，其目标是通过改进药物成分作用机制 (MoA) 预测算法来推进药物开发。

## 1.1 什么是 MoA

过去，科学家们从天然产品中提取药物，或者从传统疗法中获得灵感。非常常见的药物，如扑热息痛 (paracetamol)，在人们了解其药理作用的生物学机制几十年前就已经投入临床使用。今天，随着更强大技术的出现，药物发现已经从过去的偶然方法转变为基于对疾病潜在生物学机制的理解的更有针对性的模型。在这个新的框架中，科学家们试图识别与疾病相关的蛋白质靶标，并开发一种能够调节该蛋白质靶标的分子。作为对特定分子生物活动的速记，科学家们给分子贴上了一个标签，简称为作用机制(MoA)。

## 1.2 如何确定新药物的 MoA

一种方法是用这种药物处理人类细胞样本，然后用算法分析细胞反应，算法可以在大型基因组数据库中搜索与已知模式的相似性，比如已知 MoA 药物的基因表达库或细胞存活模式库。

# 2 数据介绍

我们需要根据各种输入特征，如基因表达数据和细胞活力数据预测不同样本(sig\_id)的作用机制(MoA)反应的多个靶标概率。

- **train\_features.csv:** 训练集对应的特征数据。特征 g-表示基因表达数据，c-表示细胞活力数据。**cp\_type** 表示使用化合物(cp\_vehicle)或控制扰动(ctrl\_vehicle)处理的样品;控制扰动无 MoA;**cp\_time** 和 **cp\_dose** 表示治疗时间(24、48、72 小时)和剂量(高、低)。
- **train\_drug.csv** 训练集样本的 sig\_id 和 drug\_id 对应关系
- **train\_targets\_scored.csv:** MOA 的二分类标签数据
- **train\_targets\_nonscored.csv:** 额外没有带有标签的 MOA 数据，在测试数据集中不存在
- **test\_features.csv:** 测试数据的特征。需要选手预测测试数据中每一行的每一个 MoA 得分的概率
- **sample\_submission.csv:** 提交文件

### 3 探索性数据分析

训练与测试特征集共占用内存 186.2MB。其中训练数据集共 23814 条数据，876 列；测试数据集共 3982，876 列。训练集与测试集使用的特征完全相同，有 872 个浮点型特性，1 个整型特征（cp\_time），3 个类别型特征。数据集没有缺失值，是很规整的数据集。下面将针对不同类型特征进行分析。

sig_id	cp_type	cp_time	cp_dose	g-0	g-1	g-2	g-3
id_0004d9e33	trt_cp	24	D1	-0.5458	0.1306	-0.5135	0.4408
id_001897cda	trt_cp	72	D1	-0.1829	0.2320	1.2080	-0.4522
id_002429b5b	ctl_vehicle	24	D1	0.1852	-0.1404	-0.3911	0.1310
id_00276f245	trt_cp	24	D2	0.4828	0.1955	0.3825	0.4244
id_0027f1083	trt_cp	48	D1	-0.3979	-1.2680	1.9130	0.2057

#### 3.1 分类特征

共有四个类别型特征，sig\_id、cp\_time、cp\_type 和 cp\_dose，sig\_id 为数据标识不用分析。

绝大多数数据的 cp\_type 都为 trt\_cp，只有 8% 的数据为 ctl\_vehicle(可以理解为对照组)，且数据的相对分布在训练集和测试集上都是相同的。

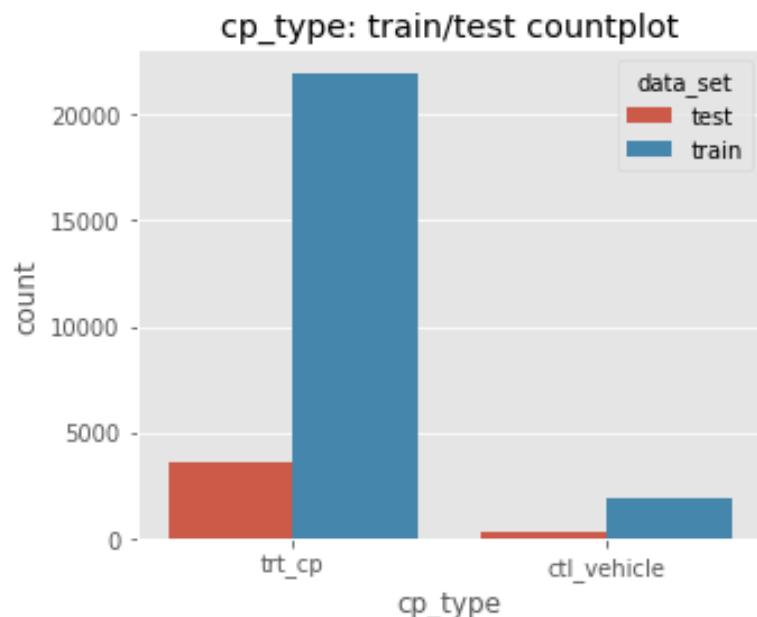


Figure 1

治疗时间(treatment duration)取值为 24, 48 和 72 小时, 分布十分均匀。

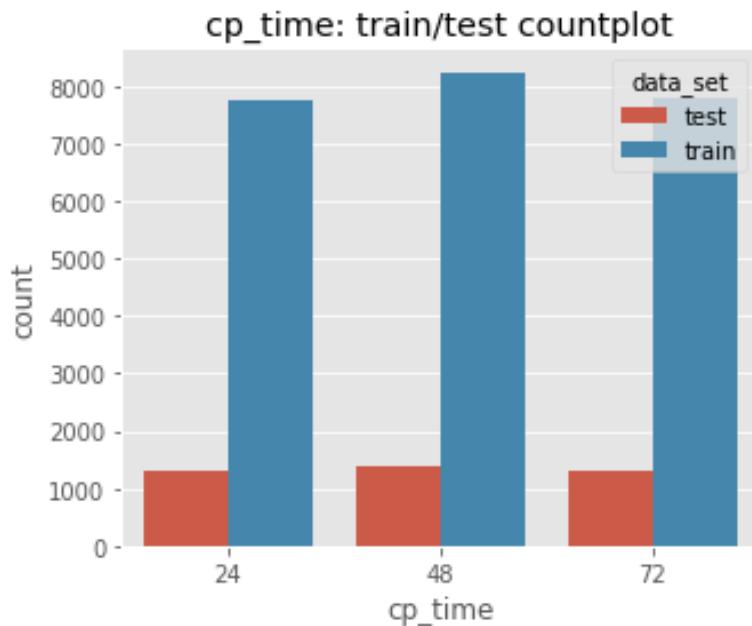


Figure 2

药物剂量也十分均匀。

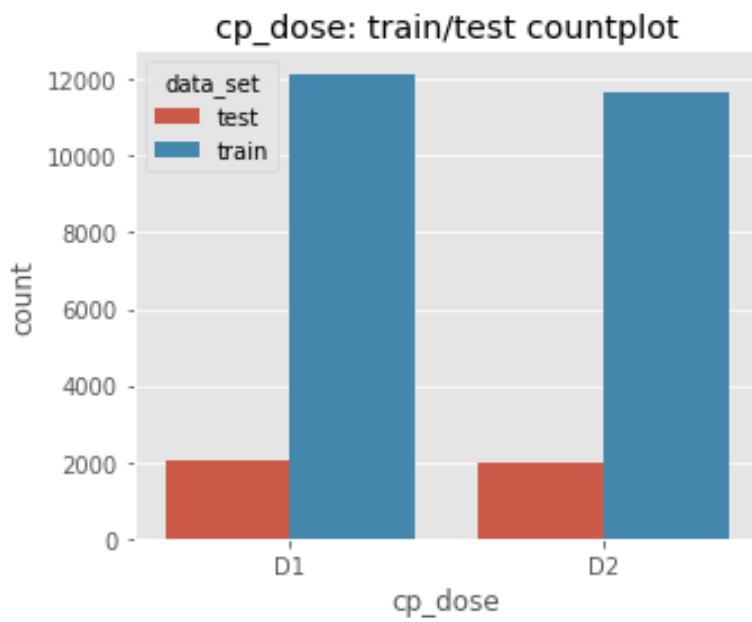


Figure 3

### 3.2 数值特征

数值型特征包括基因表达(gene expression, 数据集中前缀为 g-)和细胞活性(cell viability, 数据集中前缀为 c-), 其中基因表达特征共有 772 个, 细胞活性特征共有 100 个。

### 3.2.1 细胞活性

在细胞群体中总有一些因各种原因而死亡的细胞，总细胞中活细胞所占的百分比叫做细胞活力。数据集中共有 100 个细胞活力特征 (c-0, .....c-99)，每个细胞活力特征都是在相同的细胞基准水平上测得的。

选择 15 个细胞活力特征查看分布，可以看出训练集和测试集的细胞活力集中在均值 0 附近，且在负值部分也有一个集中趋势，整体呈现左偏趋势。

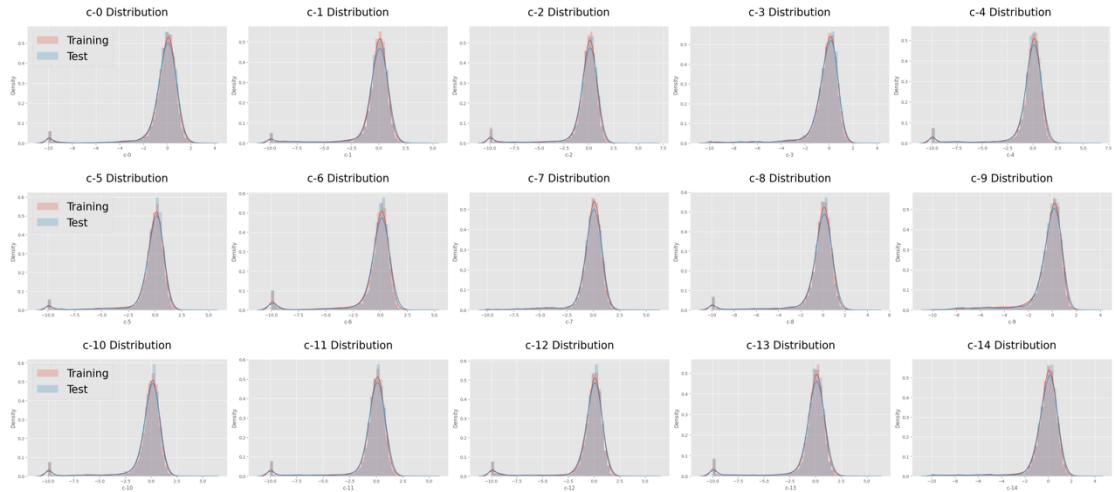


Figure 4

查看不同特征间的相关情况，由热力图颜色深度可以看出特征间相关性是很强的，在特征工程中是很有用的信息。

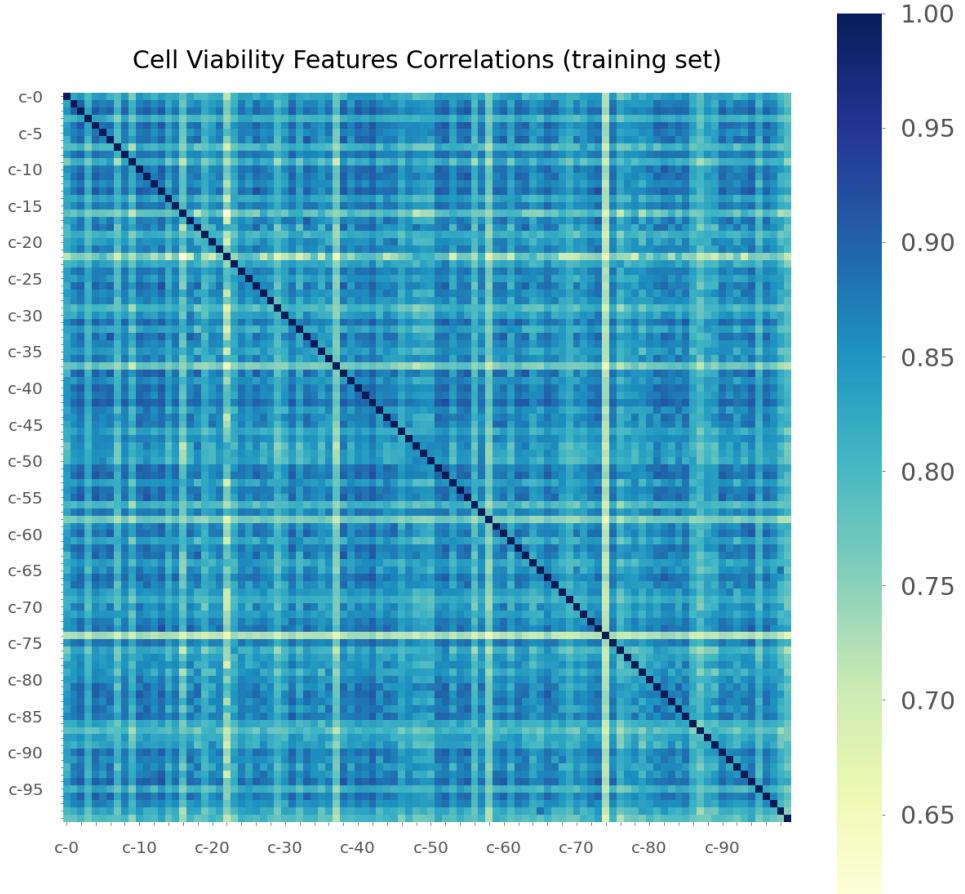


Figure 5

深入分析，选取相关性最高（均大于 0.9）的 20 对细胞活性特征，显然可以利用特征间的相关性构造新的特征，强化特征的预测能力(prediction power)。

	cell 1	cell 2	correlation
0	c-52	c-42	0.924619
2	c-13	c-73	0.923344
4	c-13	c-26	0.921875
6	c-33	c-6	0.914730
8	c-55	c-11	0.914637
10	c-63	c-38	0.914578
12	c-38	c-94	0.914368
14	c-13	c-94	0.914001
16	c-52	c-4	0.913649
18	c-4	c-42	0.913242
20	c-38	c-13	0.912127
22	c-2	c-55	0.911787
24	c-55	c-4	0.911288
26	c-4	c-13	0.911130
28	c-42	c-82	0.910847
30	c-42	c-66	0.910635
32	c-6	c-38	0.910617
34	c-2	c-13	0.910497
36	c-42	c-62	0.910387
38	c-55	c-90	0.910217

### 3.2.2 基因表达

基因表达 (gene expression) 是指将来自基因的遗传信息合成功能性基因产物的过程。数据集中共有 772 个基因表达特征(g-0,.....g-771)。

随机选取 15 个基因表达特征，可以看出相比细胞活性特征分布的同质性，基因表达特征在分布上更有特点，峰度和偏度均有明显不同，在特征工程过程中应该尽量利用到这些特征。

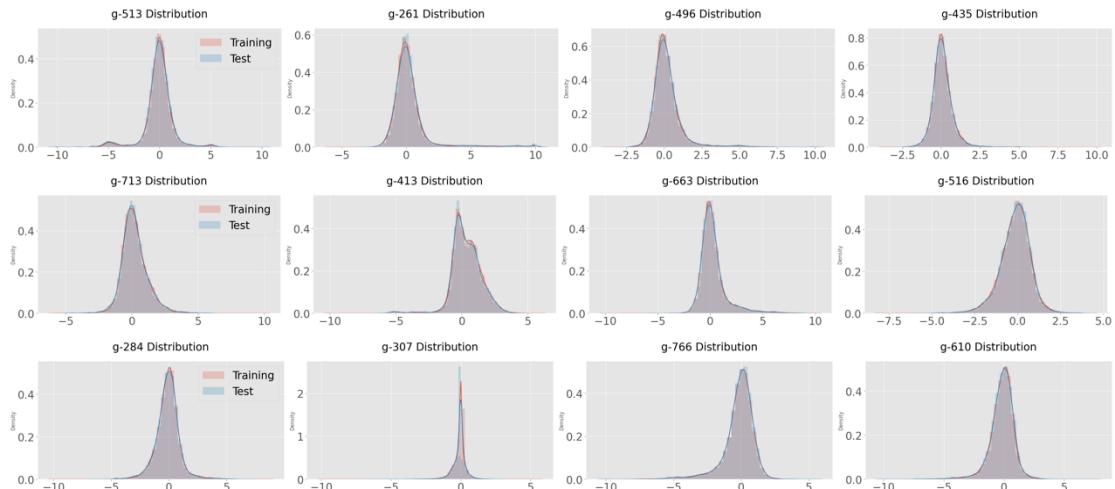


Figure 6

查看基因表达特征间的关系，可以看出绝大多数特征对的相关系数颜色很浅，均在 0.25 左右，相比细胞活性特征可利用性不强。

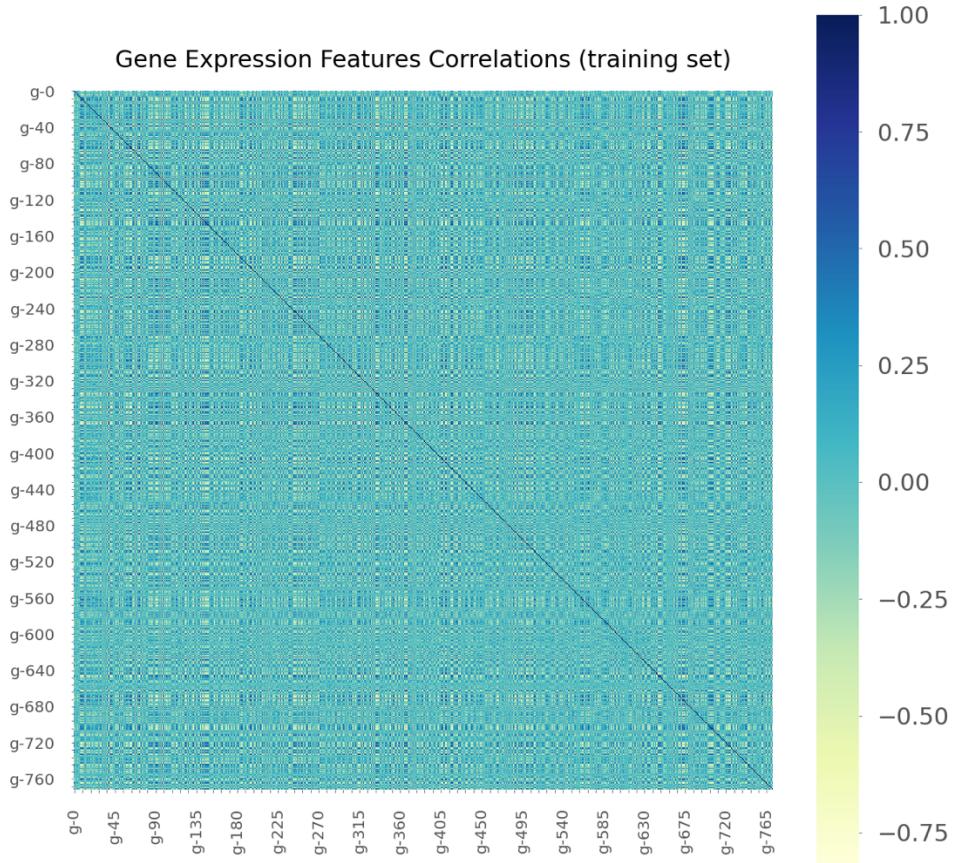


Figure 7

### 3.3 目标变量

目标变量被分为两组，non-scored 和 scored，每种药物在这些目标变量上去值为 0 或 1，这些目标变量是科学家们对每种药物提供的批注。Non-scored 变量只在训练数据中存在，比赛中是以得分变量的预测正确情况作为评价标准，但我们仍然可以利用非得分变量进行探索性分析，特征工程和建模。

得分目标共有 206 个，非得分特征共有 402 个，总共 608 个目标变量。

	sig_id	alpha_reductase_inhibitor	5-hydroxytryptamine_2a_receptor_antagonist	11-beta-hydroxylase_inhibitor	acat_inhibitor	acetylcholine_receptor_agonist	acetylcholine_receptor_antagonist
0	id_000644bb2	0	0	0	0	0	0
1	id_000779bfc	0	0	0	0	0	0
2	id_000a6266a	0	0	0	0	0	0
3	id_0015fd391	0	0	0	0	0	0
4	id_001626bd3	0	0	0	0	0	0

	sig_id	abc_transporter_expression_enhancer	abl_inhibitor	ace_inhibitor	acetylcholine_release_enhancer
0	id_000644bb2	0	0	0	0
1	id_000779bfc	0	0	0	0
2	id_000a6266a	0	0	0	0
3	id_0015fd391	0	0	0	0
4	id_001626bd3	0	0	0	0

查看样本目标变量的取值情况，可以看出对非得分变量和得分变量，每个样本均可取 0 或多类，即这是一个 multi-label 的分类问题，同时样本也可不属于已有变量中的任何一类。大多数情况下，样本被分为 1 类或 0 类，但是在训练集上也有一小部分样本被分为 2,3,4,5 或 7 个不同的标签。但非得分和得分标签分布情况不同，可以看出得分标签被分为 1 类的要比 0 类多，但非得分标签被分为 0 类要比其他情况多，下面将针对得分和非得分标签分别分析。

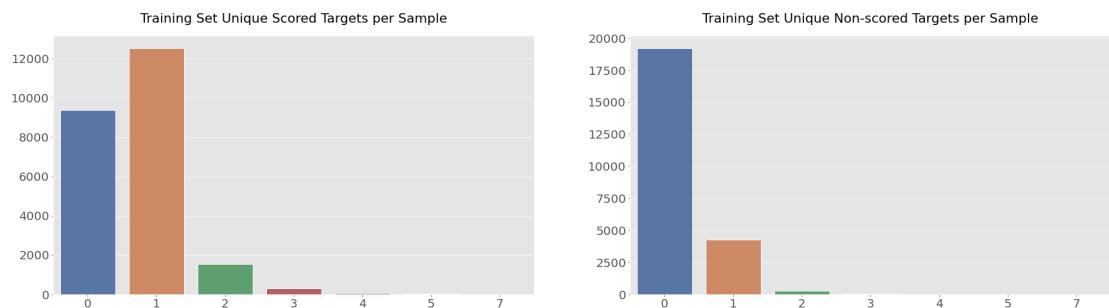


Figure 8

### 3.3.1 得分标签

可以看出，最普遍的得分标签为 nfkb inhibitor、proteasome inhibitor, cyclooxygenase inhibitor, dopamine receptor antagonist, serotonin receptor antagonist 和 dna\_inhibitor，分别有 400 多个样本被标为这些标签。最少见的标签为 atp-sensitive potassium channel antagonist 和 erbb2 inhibitor，每个标签仅有一个样本被归为这些标签。可以预见测试集也会是类似的分布情况。同时还可以看出很多得分标签分类的次数是相同的，这些标签间可能有一定的关系。

### 3.3.2 非得分标签

最普遍的非得分标签为 ace inhibitor, purinergic receptor antagonist, map kinase inhibitor, sterol demethylase inhibitor，分别有 70 多个样本被归为这些标签。同时有 71 个非得分标签没有在任何样本上取得。非得分标签的计数情况和得分标签相差很大，训练集上很多样本并没有归为任意一类。非得分标签计数相同的情况比得分标签更明显，标签间应该有更紧密的关系。

### 3.3.3 目标变量相关情况

大多数得分标签和非得分标签的相关系数都接近于 0，但是从热力图上的红点可以看出还是有一些变量对的相关系数很大（热力图上的白线代表缺失值，并不是相关系数为 1）。

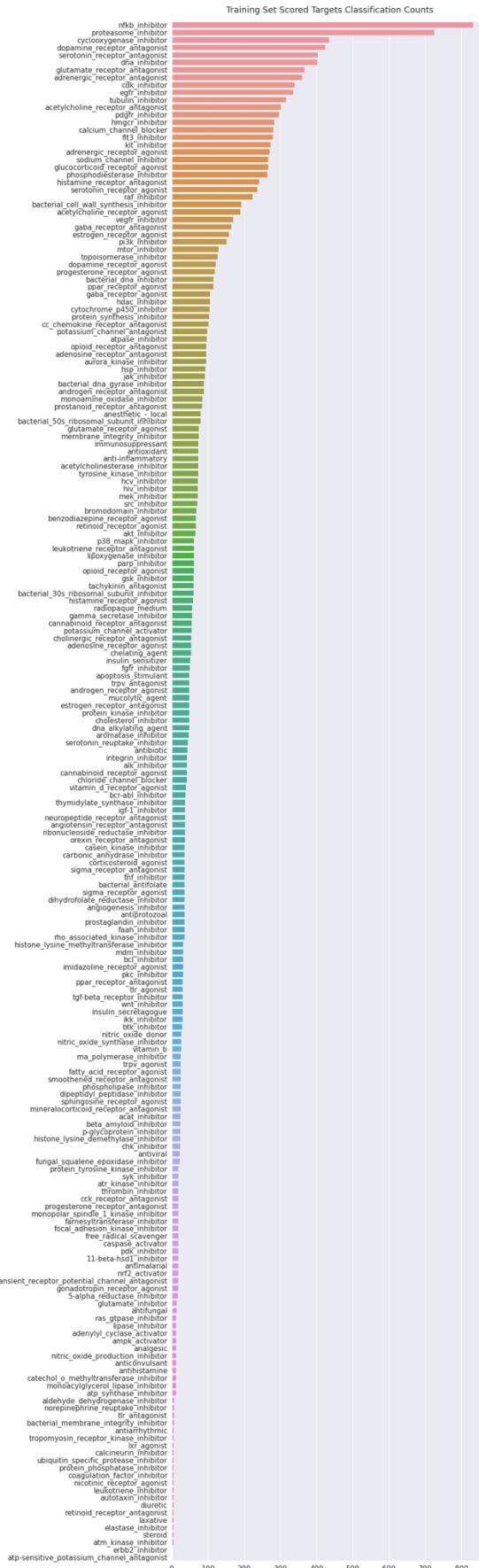


Figure 9

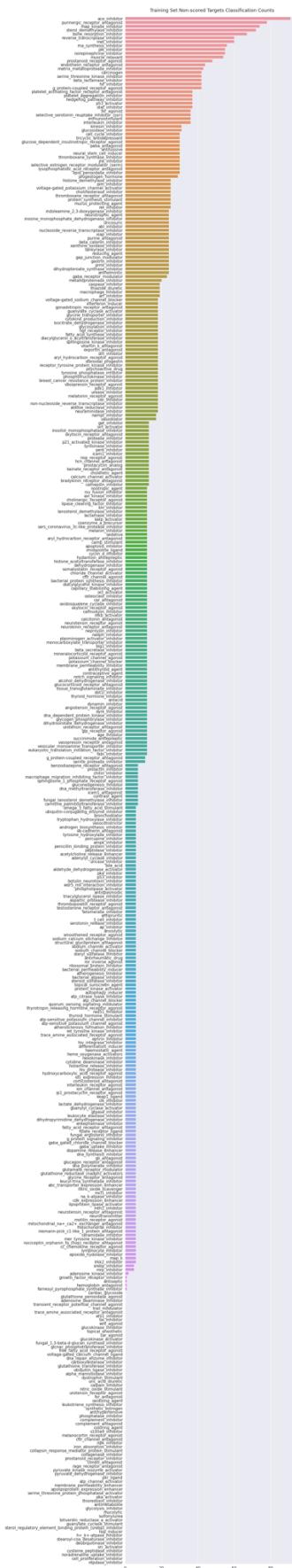


Figure 10

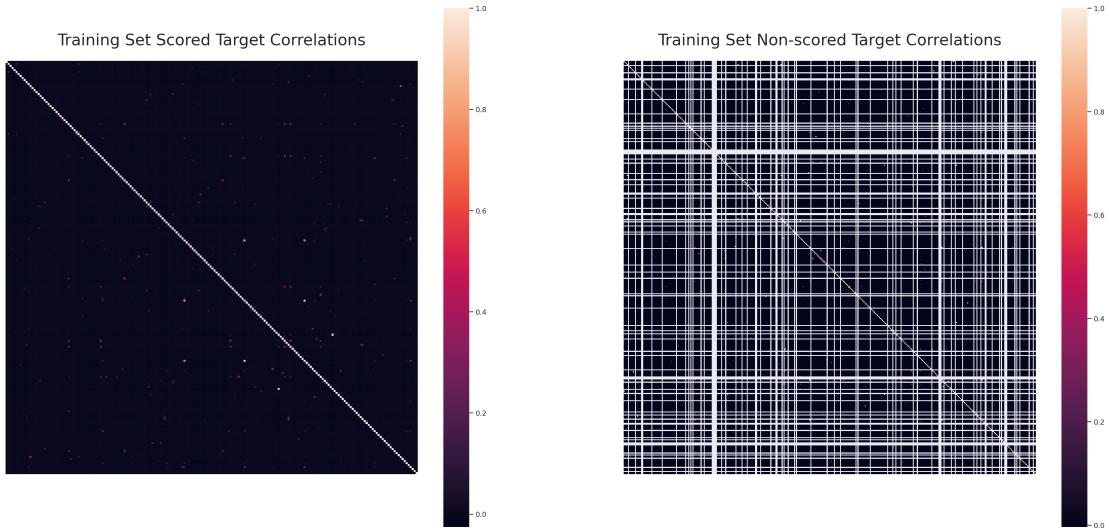


Figure 11

## 3.4 目标变量与特征

### 3.4.1 得分变量与数值型特征

查看每个得分标签与所有的数值特征的相关关系，并找出每个得分标签对应的绝对值最大的相关系数，可以看出有很多得分标签对任意一个数值型特征相关性都很弱，但有少数标签的最大相关系数超过了 0.9。

随机选取部分细胞活性和得分标签绘制散点图，可以看出大多数情况下细胞活性与得分标签间存在正相关关系，但是部分特征与标签存在负相关或无明显关系。

同理绘制基因表达与得分标签散点图，相比细胞活性，基因表达与得分标签的相关性很弱，标签取值为 1 的数据点在水平方向上分布更均匀。所有的基因特征和细胞活性特征在标签取值为 1 时都聚集在 0 附近。基因表达特征取值很大(> 2 或 <-2) 表示药物对当前细胞影响显著，取值接近于 0 意味着药物对细胞没有显著影响，可以忽略不计。

同理可以对非得分标签以及分类特征的相关情况进行分析，可查看 Acknowledgement，这里不再赘述。

	5-	11-beta-			
train_features	alpha_reductase_inhibitor	hsd1_inhibitor	acat_inhibitor	acetylcholine_receptor_agonist	acetylcholine_receptor_antagonist
g-0	-0.008317	-0.011513	0.003049	-0.019100	-0.034001
g-1	-0.004291	-0.004084	-0.000265	-0.005629	0.004920
g-2	0.000719	-0.002585	-0.004516	-0.004385	-0.016947
g-3	-0.008268	-0.002384	0.001146	-0.007750	-0.010446
g-4	-0.003799	0.002661	0.006287	-0.010288	-0.010128
...	...	...	...	...	...
c-95	0.010113	0.004202	0.003092	0.022601	0.029189
c-96	0.008570	0.008131	0.010796	0.019768	0.029630
c-97	0.009440	0.003168	0.009347	0.021624	0.026226
c-98	0.013253	0.010387	0.010546	0.030946	0.035763
c-99	0.008512	0.007679	0.013209	0.021847	0.027331

Best Correlation with train features for every scored target

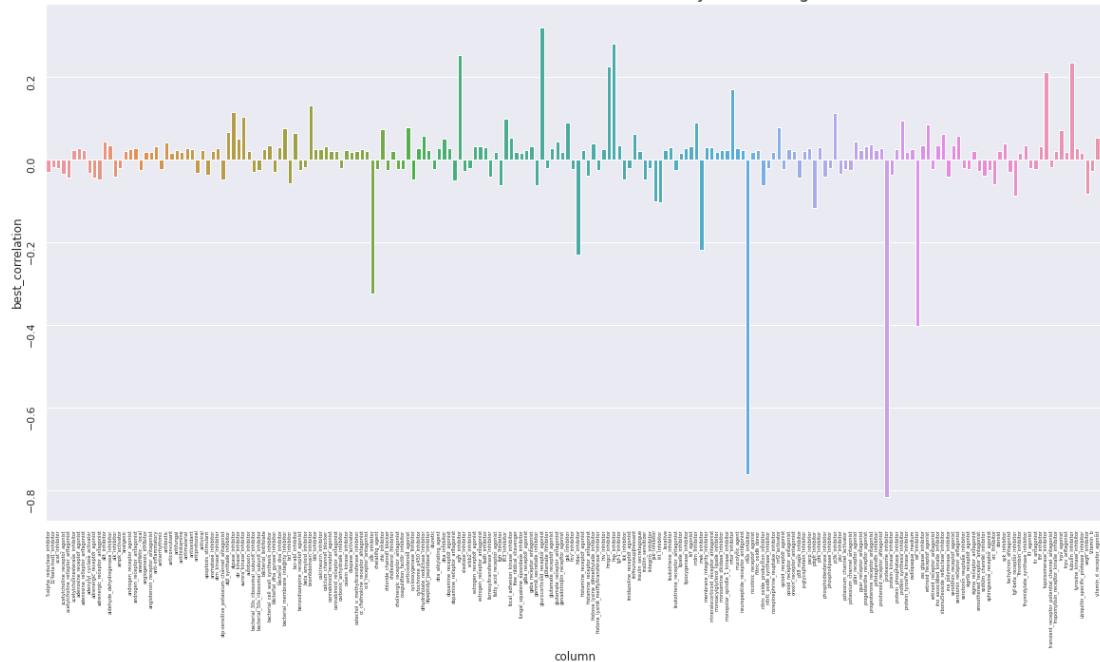


Figure 12

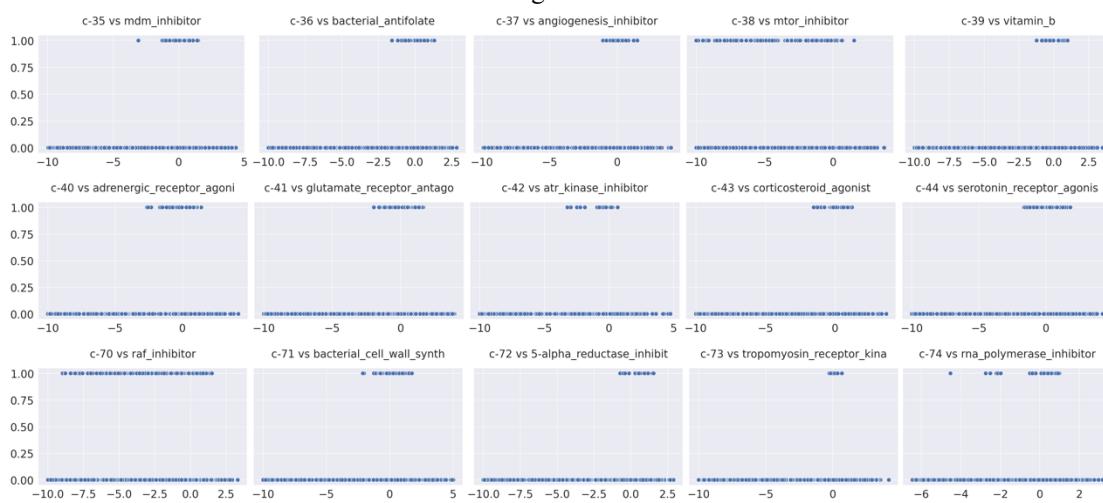


Figure 13 c-target

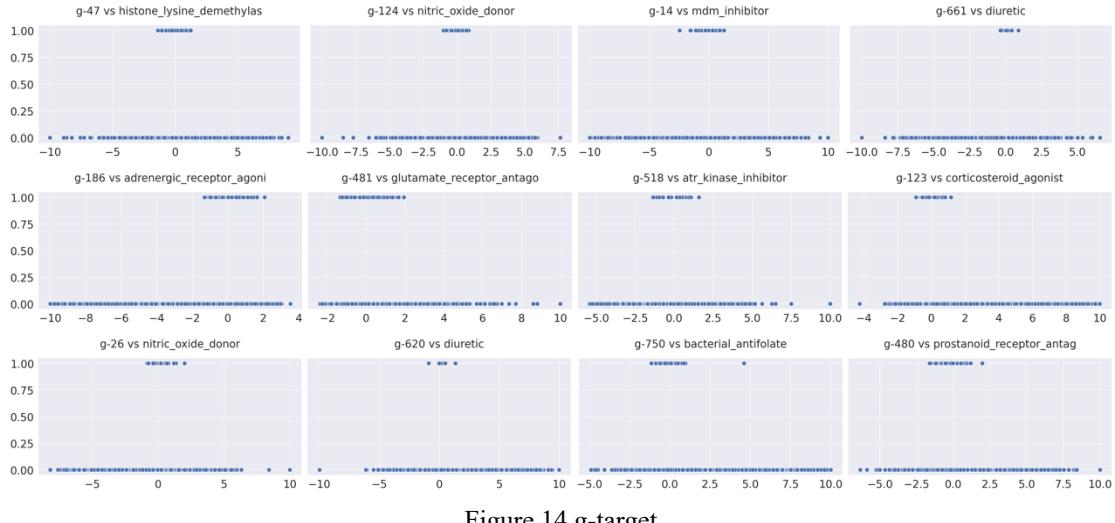


Figure 14 g-target

### 3.5 drug\_id

举办方在比赛临近结束时加入了新的 drug\_id 数据, 只在训练数据集中存在, 通过查看 drug\_id 的技术情况, 可以看出绝大多数 drug\_id 在数据集中出现了 18 次及以下, 且集中在 6 次左右。

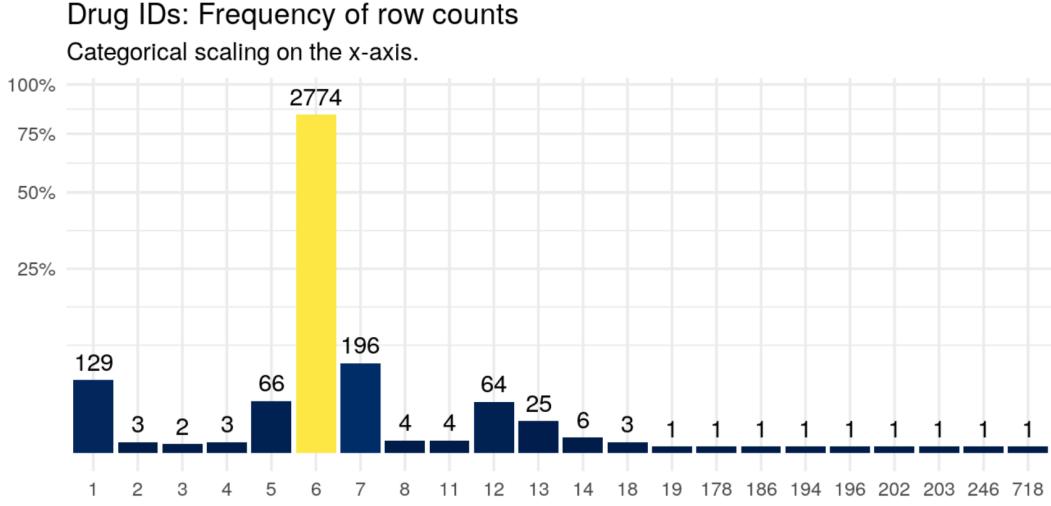


Figure 15

## 4 特征工程

由于生物实验数据获得原理的特殊性, 绝大多数实验数据都使用 quantile normalization 将特征缩放在同样的分布或范围下。Quantile normalization 通过执行一个秩转换能够使异常的分布平滑化, 并且能够比缩放更少地受到离群值的影响。但是, 它会使特征间及特征内的关联和距离失真。QuantileTransformer 函数提供了一个基于分位数函数的无参数转换, 将数据映射到了零到一的均匀分布或高斯分布上。这种 scaling 方法在 kaggle 竞赛上十分常见, 并且通过实验证明了对特定的数据, quantile normalization 效果比其他 scaling 方法更好, 本文中即对

基因表达特征和细胞活性都使用了该方法，转换后的分布如图，可以看出数据分布为高斯形状，但仍然可以看出基本的统计特征。算法原理可见：

<https://academic.oup.com/bioinformatics/article/19/2/185/372664>

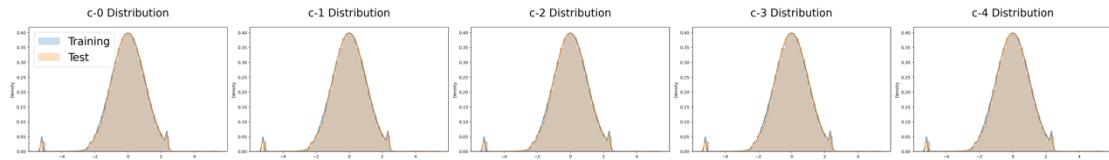


Figure 16

主成分方法是一种常用的降维方法，每一个主成分都寻求在最大程度代表原变量，并与其他主成分无关。但在本文采取主成分方法分别对基因表达和细胞活性特征提取主成分，并将这些主成分通过聚类方法分类标签，以标签作为新的分类特征（这一步应该是有 data leakage 的，但由于测试集和训练集在特征上分布情况大致相同，影响不是很大；且该方法如果在训练和测试集上分别进行，主成分和聚类标签的意义在两个集合上不同，作为新的特征也没有可解释性）。

通过尝试，对基因表达特征选取 600 个主成分，累计方差贡献率 0.9486；对细胞活性特征选取 50 个主成分，累计方差贡献率 0.8166，经过 PCA 的数据情况如下：

	pca_G-0	pca_G-1	pca_G-2	pca_G-3	pca_G-4	pca_G-5	pca_G-6	pca_G-7	pca_G-8	pca_G-9	...	pca_C-40	pca_C-41	pca_C-42	pca_C-43
0	-5.509135	3.934672	9.358451	-7.911861	4.834325	0.834668	3.446985	1.625308	0.909537	2.112349	...	1.000458	0.067321	0.017445	-0.68932
1	-4.899750	3.891428	-11.376574	5.777669	0.924188	0.258293	1.041124	-0.395364	5.401568	1.449645	...	-0.095356	0.709016	-0.517945	-2.04136
2	1.258620	-7.242684	-5.448532	-0.802672	0.922407	3.698523	-1.905278	2.625305	-4.668135	0.999668	...	1.339214	0.401715	0.085583	-0.89143
3	11.514493	-8.717475	-4.263702	-5.765282	-7.029854	-2.680401	-2.229116	6.561454	-2.617044	-3.505768	...	-0.463129	-0.002843	-0.561360	0.07859
4	-6.541055	-2.337753	-10.742705	-4.184780	-8.086348	-8.202837	-4.266611	-3.182038	-1.694953	0.846413	...	1.269987	1.596501	-1.739129	0.27318

对提取的主成分进行 K-Means 方法聚类，采取 Elbow Method 确定聚类数，可以看出，肘点为 5，即将主成分聚为 5 类。再用 PCA 方法降聚类后的主成分降为二维平面可视化。

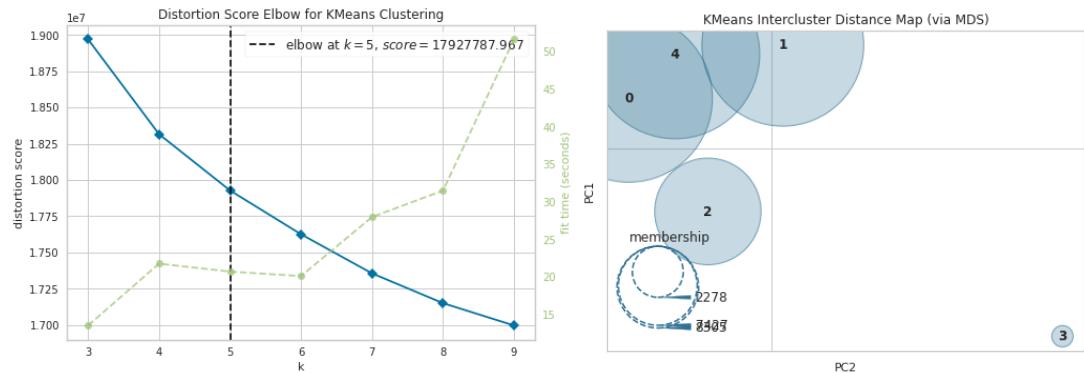


Figure 17

同理也可以直接用原特征进行聚类，这里根据方差阈值过滤掉方差在 0.85 以下的特征，对 0.85 以上的特征进行聚类，这些高方差的特征更具有 prediction power，聚类后的效果也更明显。同上根据肘部法则分别对基因表达特征和细胞活性特征进行 K-Means 聚类，对基因表达特征采取聚类数为 22，细胞活性特征聚类数为 4。

## 4.1 统计特征

①由 3.2.1，将相关系数最强的 20 对细胞活性特征相乘获得组合特征，比如 c-52\*c-42=c52\_c42。

```
df['c52_c42'] = df['c-52'] * df['c-42']
df['c13_c73'] = df['c-13'] * df['c-73']
df['c26_c13'] = df['c-23'] * df['c-13']
df['c33_c6'] = df['c-33'] * df['c-6']
df['c11_c55'] = df['c-11'] * df['c-55']
df['c38_c63'] = df['c-38'] * df['c-63']
df['c38_c94'] = df['c-38'] * df['c-94']
df['c13_c94'] = df['c-13'] * df['c-94']
df['c4_c52'] = df['c-4'] * df['c-52']
df['c4_c42'] = df['c-4'] * df['c-42']
df['c13_c38'] = df['c-13'] * df['c-38']
df['c55_c2'] = df['c-55'] * df['c-2']
df['c55_c4'] = df['c-55'] * df['c-4']
df['c4_c13'] = df['c-4'] * df['c-13']
df['c82_c42'] = df['c-82'] * df['c-42']
df['c66_c42'] = df['c-66'] * df['c-42']
df['c6_c38'] = df['c-6'] * df['c-38']
df['c2_c13'] = df['c-2'] * df['c-13']
df['c62_c42'] = df['c-62'] * df['c-42']
df['c90_c55'] = df['c-90'] * df['c-55']
```

②将所有细胞活性特征乘方，得到新特征。

```
for feature in features_c:
    df[f'{feature}_squared'] = df[feature] ** 2
```

③将 kaggle 讨论区获得的 Resnet 模型中预测能力很强的部分基因表达特征乘方得到新特征。

```
gsquarecols=['g-574', 'g-211', 'g-216', 'g-0', 'g-255',
             'g-577', 'g-153', 'g-389', 'g-60', 'g-370',
             'g-248', 'g-167', 'g-203', 'g-177', 'g-301',
             'g-332', 'g-517', 'g-6', 'g-744', 'g-224',
             'g-162', 'g-3', 'g-736', 'g-486', 'g-283',
             'g-22', 'g-359', 'g-361', 'g-440', 'g-335',
             'g-106', 'g-307', 'g-745', 'g-146', 'g-416',
             'g-298', 'g-666', 'g-91', 'g-17', 'g-549',
             'g-145', 'g-157', 'g-768', 'g-568', 'g-396']

for feature in gsquarecols:
    df[f'{feature}_squared'] = df[feature] ** 2
```

④以每个样本为粒度，分别获得基因表达特征、细胞活性特征以及两特征总体的统计量，包括总和、标准差、偏度和峰度。

```

for df in train, test:
    df['g_sum'] = df[features_g].sum(axis = 1)
    df['g_mean'] = df[features_g].mean(axis = 1)
    df['g_std'] = df[features_g].std(axis = 1)
    df['g_kurt'] = df[features_g].kurtosis(axis = 1)
    df['g_skew'] = df[features_g].skew(axis = 1)
    df['c_sum'] = df[features_c].sum(axis = 1)
    df['c_mean'] = df[features_c].mean(axis = 1)
    df['c_std'] = df[features_c].std(axis = 1)
    df['c_kurt'] = df[features_c].kurtosis(axis = 1)
    df['c_skew'] = df[features_c].skew(axis = 1)
    df['gc_sum'] = df[features_g + features_c].sum(axis = 1)
    df['gc_mean'] = df[features_g + features_c].mean(axis = 1)
    df['gc_std'] = df[features_g + features_c].std(axis = 1)
    df['gc_kurt'] = df[features_g + features_c].kurtosis(axis = 1)
    df['gc_skew'] = df[features_g + features_c].skew(axis = 1)

```

## 4.2 特征处理总结

完成以上特征处理后将 cp\_type 为控制组取值 ctl\_vehicle 的样本去掉，并删除 cp\_type 列。

原特征	特征处理(按顺序)	新特征	特征数
c-0 to c-99	QuantileTransformer	c-0 to c-99	100
g-0 to g-771	QuantileTransformer	g-0 to g-771	772
c,g-	PCA+K-Means	clusters_pca_0 to clusters_pca_4	5
c-(Variance 0.85+)	K-Means	clusters_c_0 to clusters_c_3	4
g-(Variance 0.85+)	K-Means	clusters_g_0 to clusters_g_21	22
c-	Square	c_squared	100
g-574,g-211.....	Square	g_squared	45
c-52&c-42.....	Multiply	c52_c42.....	20
c-,g-,c-&g-	Sum,mean,std,kurt,skew	NA	15
cp_type	Drop ctl_vehicle		
cp_time,cp_dose	One-hot encoding		

表格 1 特征工程

## 5 评价标准及优化函数

从 EDA 过程中可以看出，这是一个 Multi-label 多标签分类问题，且每个样本可以被分为 0 类或多类标签，举办方确定的评价标准为 log loss。对每一行，需要预测每个样本被分为某一标签的概率。若有 N 行 M 个得分标签，就有 N\*M 个预测。

$$\text{log loss} = -\frac{1}{M} \sum_{m=1}^M \frac{1}{N} \sum_{i=1}^N [y_{i,m} \log(\hat{y}_{i,m}) + (1 - y_{i,m}) \log(1 - \hat{y}_{i,m})]$$

- N 为行数( $i = 1, 2, \dots, N$ )
- M 为目标变量数( $m=1, 2, \dots, M$ )
- $\hat{y}_{i,m}$  为第  $i$  行第  $m$  列的预测概率

- $y_{i,m}$  为第  $i$  行第  $m$  列的真实值（1 或 0）
- $\log$  为自然对数

该损失函数在 pytorch 中可以使用 BCEWITHLOGITSLOSS 函数直接实现。

但在训练集上如果直接使用 log loss 作为目标函数进行训练，有可能会出现过拟合的情况，尤其是在本问题中数据量相对较少，模型很容易对预测结果过于自信，导致在测试集上的表现变差，因此考虑对得分目标进行 label smoothing (label smoothing 是对真实值操作的，与预测结果无关)。针对 multi-label 的二分类问题，平滑公式如下：

$$y^{LS} = y(1 - \alpha) + 0.5\alpha$$

$\alpha$  为平滑指数，取值范围为 [0,1]，可以看出当  $\alpha = 0$  时即为原标签， $\alpha = 1$  时恒为 0.5。可以看出相比未平滑的标签只能取值为 0 或 1 的 one-hot encoding，label smoothing 缩小预测值与真实值之间的“差距”，防止模型在训练过程中过于自信，导致通用能力（在测试集上预测）变差。

## 6 模型

本文使用了三个模型，将每个模型的预测结果加权平均得到最终预测。

### 6.1 训练/验证集划分

这是一个多标签的分类问题，为确保划分集合时每个标签在集合中的分布情况相同，需采用分层方法得到验证集，同时多标签问题需采用 iterative stratification 算法（算法原理可见下文 P149，在 python 中有函数实现）。

[https://link.springer.com/content/pdf/10.1007%2F978-3-642-23808-6\\_10.pdf](https://link.springer.com/content/pdf/10.1007%2F978-3-642-23808-6_10.pdf)

同时在竞赛过程中讨论区出现了根据 drug\_id 在训练集的出现频数再分层的方法，分别对 drug\_id 出现次数大于 18 次和小于等于 18 次进行分层，确保每一层出现次数多的 drug\_id 和出现次数少的 drug\_id 分布情况都相同。以此分为 7 折，每折都均匀地有 7 层。训练 7 个随机种子，每个随机种子训练 7 折，对每个种子每折得到的预测结果进行简单评价，得到模型的最终预测结果。

<https://www.kaggle.com/c/lsh-moa/discussion/195195>

### 6.2 模型一

模型一 Architecture 如下，采取 Adam 优化器，超参设置可参考代码。在训练过程中使用了 transfer learning 的 gradual unfreezing 技术，即先对得分和非得分目标变量预先训练好一个表现最好的模型，然后只针对得分变量，从最后一层开始逐渐解冻模型，每次 epoch 内微调所有已解冻的层；然后解冻下一个较低的层，再微调；以上重复；直到微调所有的层，直到在最后一次迭代中收敛。最终得到模型一的 cv log loss 为 0.015621。

Layer	BN	Dropout	Activation	Size(in,out)
input				
Layer 1	✓		Leaky ReLU	#feature,1500
Layer 2	✓	0.5	Leaky ReLU	1500,1250
Layer 3	✓	0.35	Leaky ReLU	1250,1000
Layer 4	✓	0.3	Leaky ReLU	1000,750
Layer 5	✓	0.25	Weight_norm	750,#target
output				

表格 2 模型一结构

### 6.3 模型二

模型二使用了 TabNet (<https://arxiv.org/pdf/1908.07442.pdf>) 这篇论文提出的 TabNet 是一种针对于表格数据的神经网络，它通过类似于加性模型的顺序注意力机制（sequential attention mechanism）实现了 instance-wise 的特征选择，还通过 encoder-decoder 框架实现了自监督学习，从而将树模型的可解释性与 DNN 的表征能力很好地结合到了一起，相信这种兼具两者优点的模型将会成为数据挖掘竞赛中的一大利器，也对未来的研究提供了一个很好的思路。模型的结构如下图，encoder 部分可以由多个 step 顺序构成（当然 step 越多越容易 overfitting）。feature transformer 模块用于特征计算，包含两个部分，一部分仅在一个 step 实现，另一部分对每个 step 共享，同时加入 skip connection 加快训练。Attentive transformer 模块则用于计算当前 step 的 Mask 层（每个 Mask 层都包括上一个 step 得到的信息），可以理解为每个 step 特征的局部重要性，通过 Mask 层与当前 step 输出进行乘法并对每个 step 加和即可得到 feature attributes 即特征的全局重要性。每个 step 的输出加和后经过 FC 层即可得到 encoder 部分的最终输出。具体的算法原理可参考论文，项目中使用了 pytorch 封装的 tabnet 模块，超参可见代码（只使用了一个 step），得到 CV log loss=0.015814。

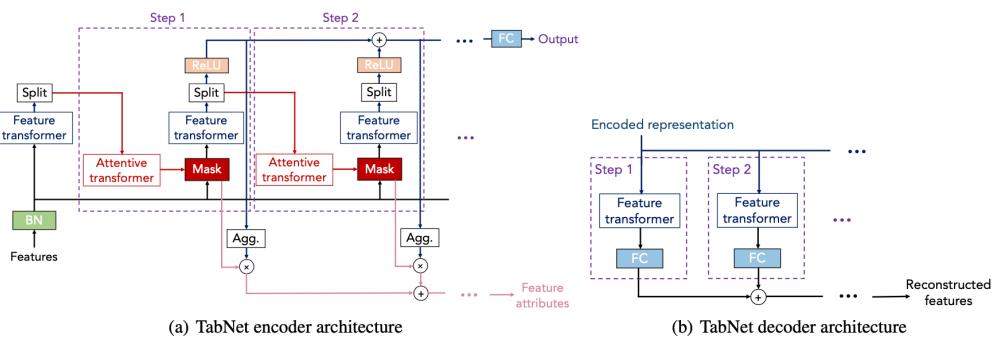


Figure 18

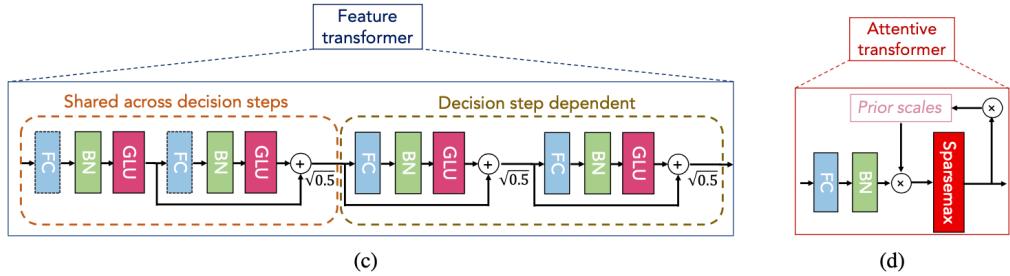


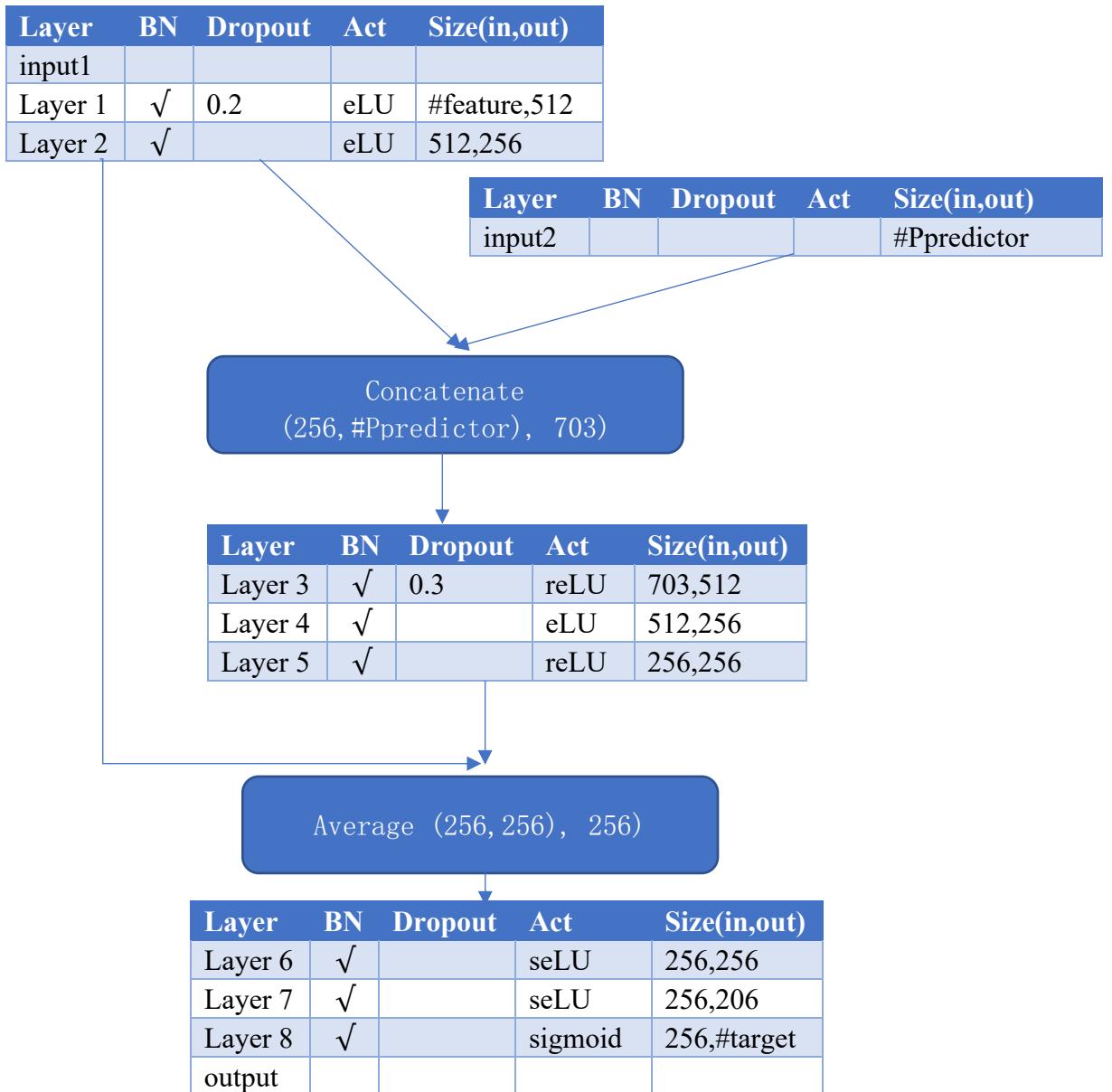
Figure 19

## 6.4 模型三

模型三融合了 5 个模型，一个为 Resnet 的变体模型（Resnet 的 input 分为两路，一路为全部 feature，另一路为 fork 来的 powerful predictor），其它为简单的 5 层、4 层、3 层、2 层 DNN 模型。下面给出 Resnet 变体和 5 层 DNN 的 architecture，其余模型结构和 5 层 DNN 类似。最后的 CV log loss 为 0.015693。

## 6.5 模型融合

以 0.32, 0.34, 0.34 的权重对三个模型预测概率加权，最终得到 Private Leaderboard log loss 为 0.016138。



表格 3 Resnet 结构

Layer	BN	Dropout	Act	Size(in,out)
input1				
Layer 1	✓	0.4	ReLU	#feature,2560
Layer 2	✓	0.4	ReLU	2560,2048
Layer 3	✓	0.4	ReLU	2048,1524
Layer 4	✓	0.4	ReLU	1524,1012
Layer 5	✓	0.4	ReLU	1012,780
output	✓	0.2	ReLU	780,#target

表格 4 5 layer DNN 结构

## 7 总结

虽然第一次参赛就取得了银牌（180/4373），但这次比赛学习到的新知识显然要比一枚银牌更宝贵；特征工程不仅仅是简单地处理缺失值，one hot encoding 甚至 frequency encoding，我们通过对数据的探索性分析提取更为有效的信息，这次比赛也锻炼了我的代码能力，通过写类将代码规范化，也练习了 pytorch 和 tensorflow 的常用功能；接触了全新的 tabnet 模型，在对表格数据建模时不再只能单一地使用 xgboost, lightgbm, catboost 等树模型的集成方法，可以将神经网络的各种架构融入到表格数据的建模中。

整理了一些排名靠前的解法，下面分模块总结 winning solution 的创新点。

### ① 特征工程

- Log scaler; 先按样本归一化再使用 quantiletransformer;
- t-SNE 降维
- 先对测试集通过 1 层的 NN 预测非得分变量，然后将非得分变量作为特征建模
- 去掉在实验组和对照组上差异较小的特征
- 用原始数据做 data augmentation

### ② 模型创新

- 模型融合使用算法估计权数分配
- 使用 DeepInsight 模型将结构数据转化为图像数据
- 使用 CNN 作为主要模型
- 注意力机制的 DNN 模型，在训练过程中随 epoch 增加 batch size

## Reference

- [1] Lamb J, et al. The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*. 2006/9/29. 313(5795):1929-35, (2006).
- [2] B.M. Bolstad, R.A Irizarry, M. Åstrand, T.P. Speed, A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics*, Volume 19, Issue 2, 22 January 2003, Pages 185–193, <https://doi.org/10.1093/bioinformatics/19.2.185>
- [3] Sechidis K., Tsoumakas G., Vlahavas I. (2011) On the Stratification of Multi-label Data. In: Gunopulos D., Hofmann T., Malerba D., Vazirgiannis M. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2011. Lecture Notes in Computer Science, vol 6913. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-23808-6\\_10](https://doi.org/10.1007/978-3-642-23808-6_10)
- [4] [arXiv:1908.07442](https://arxiv.org/abs/1908.07442) [cs.LG]
- [5] [arXiv:1906.02629](https://arxiv.org/abs/1906.02629) [cs.LG]