

Digital Reputation Challenge

2-ое место

Цыпин Артем
22.10.2019

Таблица X1

Большинство переменных являются бинарными.

```
X1.head()
```

	id	1	2	3	4	5	6	7	8	9	...	16	17	18	19	20	21	22	23	24	25
0	3	1	-1.0	-1.0	107.0	255.0	537.0	10.0	41.0	0.0	...	0	0	0	0	0	0	1	0	1	0
1	5	0	0.0	0.0	20.0	0.0	188.0	1.0	25.0	2.0	...	0	0	0	0	0	0	0	0	0	0
2	6	1	0.0	0.0	158.0	155.0	3092.0	3.0	218.0	29.0	...	0	0	0	0	0	0	0	1	0	0
3	8	1	0.0	0.0	102.0	343.0	341.0	0.0	24.0	2.0	...	0	0	0	0	0	0	1	0	0	0
4	10	1	0.0	0.0	1.0	1.0	33.0	0.0	41.0	1.0	...	0	0	0	0	0	0	1	0	1	0

Таблица X1

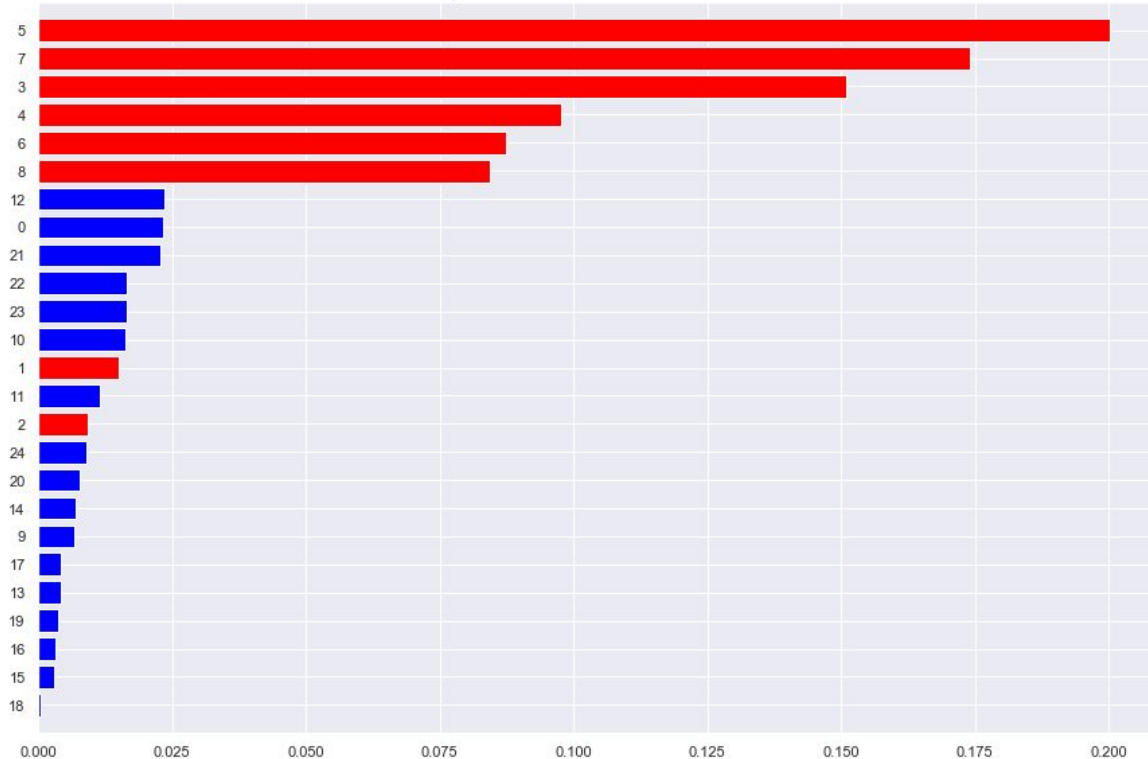
Анализ значений не бинарных переменных таблицы X1.

'2'	5	-2	2	-0.038	0.412	[-2 -1 0 1 2]	[97 109 3662 113 19]
'3'	5	-2	2	-0.020	0.306	[-2 -1 0 1 2]	[52 64 3803 73 8]
'4'	531	0	9967	102.645	359.097	[0 1 2 3 4 5 6 7]	[549 277 198 152 130 125 97 86]
'5'	721	0	10000	184.296	616.335	[0 1 2 3 4 5 9 10]	[2061 92 59 38 33 30 20 19]
'6'	677	0	10000	218.048	505.995	[0 73 68 69 48 35 41 47]	[35 28 28 27 25 25 25 25]
'7'	115	0	1989	8.085	60.907	[0 1 2 3 4 5 6 7]	[1656 691 423 253 185 115 98 77]
'8'	452	0	2409	95.189	133.882	[2 3 11 1 9 5 0 8]	[62 59 57 56 51 50 49 48]
'9'	52	0	1396	2.919	27.434	[0 1 2 3 4 5 6 7]	[1876 836 351 238 137 108 87 63]

Таким образом, таблица X1 состоит из бинарных переменных, счётчиков и двух категориальных переменных.

Важность признаков в X1

Feature importances for class 1. Counters are red



Mean AUC-ROC = 0.5614

По классам есть
небольшой разброс в
значении AUC-ROC.

Score for class 0 = 0.569

Score for class 1 = 0.544

Score for class 2 = 0.583

Score for class 3 = 0.561

Score for class 4 = 0.551

Простой Random Forest на
таблице X1 дает схожие с
бейзлайн решением
организаторов результаты.

Таблица X3

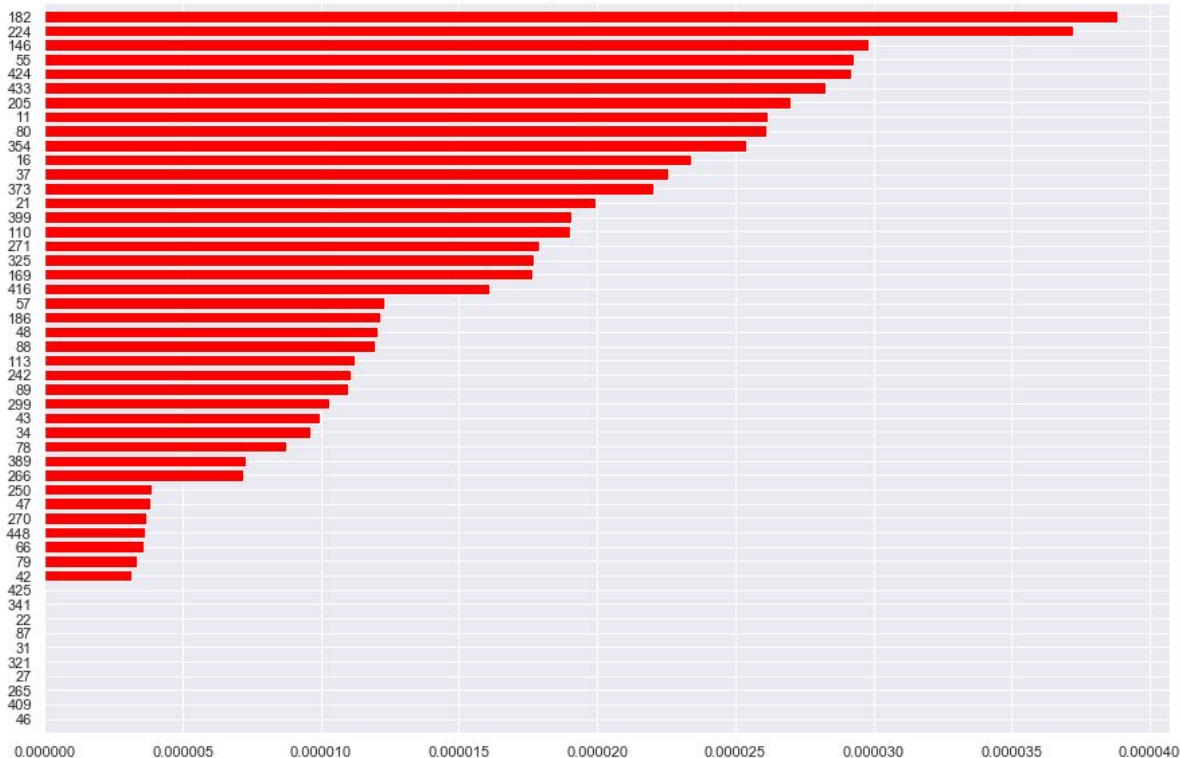
Значения в таблице X3 суммируются в 1 по id.
Предположение: переменные в X3 являются некоторыми категориями ресурсов. Таким образом, значения получаются путем нормализации количества посещений пользователем категории на общее количество посещений пользователя.

X3.head()

	id	1	2	3	4	5	6	7	8	9	...	443	444	445	446	447	448	449	450	451	452
0	3	0.0	0.0	0.00	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.00	0.0	0.0	0.022222	0.0	0.0	0.0	0.000000
1	5	0.0	0.0	0.00	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.00	0.0	0.0	0.029703	0.0	0.0	0.0	0.000000
2	6	0.0	0.0	0.00	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.00	0.0	0.0	0.222222	0.0	0.0	0.0	0.111111
3	8	0.0	0.0	0.02	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.0	0.0	0.02	0.0	0.0	0.060000	0.0	0.0	0.0	0.000000
4	10	0.0	0.0	0.00	0.0	0.055556	0.055556	0.0	0.0	0.0	...	0.0	0.0	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000

Важность признаков в X3

Feature importances for class 2.



Mean AUC-ROC = 0.5049

Score for class 0 = 0.497

Score for class 1 = 0.518

Score for class 2 = 0.507

Score for class 3 = 0.494

Score for class 4 = 0.508

Для большинства таргетов
данные из таблицы X3
бесполезны.

Таблица X2

Матрица с информацией о посещениях ресурсов пользователями.

Самый простой способ получить признаки из X2: подсчитать суммарное количество посещений для пользователей.

Чуть более сложный метод: разбить ресурсы на категории (например по популярности) и подсчитать для каждого пользователя нормированное количество посещений ресурсов каждой категории.

Более сложные методы: матричные разложения для user-item матрицы, индуцированной таблицей X2.

Baseline solution

5 различных LGBM-моделей на различных наборах признаков -- по одной для каждого таргета.

- 1) X1 + resource_counter. Params: learning_rate=0.002, n_estimators=325, max_depth=3.
- 2) X1 + X3 + resource_counter. Params: learning_rate=0.03, n_estimators=650, max_depth=1.
- 3) X1. Params: learning_rate=0.005, n_estimators=800, max_depth=2.
- 4) X1 + resource_counter. Params: learning_rate=0.0033, n_estimators=600, max_depth=2.
- 5) X1. Params: learning_rate=0.011, n_estimators=600, max_depth=1.



Матричные разложения

Вид матрицы X_2 наталкивает на мысль о применении методов неявных матричных разложений для получения вложений пользователей и ресурсов в векторное пространство.

	item 1	item 2	item 3	...	item n
user 1					
user 2					
user 3					
user 4					
user 5					
user 6					
user 7					
user 8					
...					
user n					

\underline{R}

\approx

	feature 1	feature 2
user 1		
user 2		
user 3		
user 4		
user 5		
user 6		
user 7		
user 8		
...		
user n		

\underline{U}

	item 1	item 2	item 3	...	item n
feature 1					
feature 2					

\mathcal{X}

\underline{V}

ALS

Ищем вложения пользователей и ресурсов в векторное пространство, оптимизируя следующий функционал:

$$\min_{y_*, y_*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

ALS

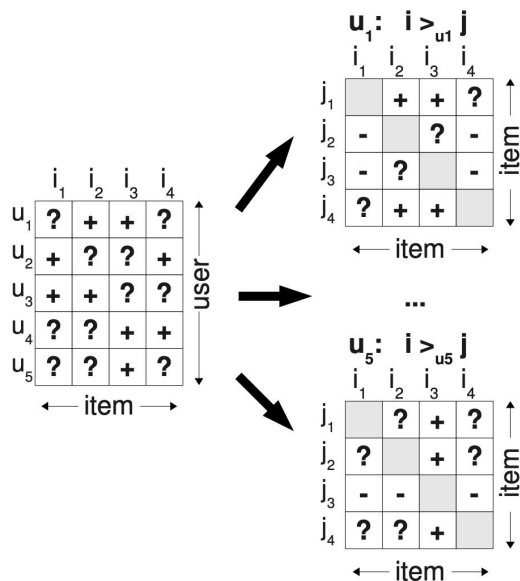
Итеративно оптимизируем представления, используя только ненулевые элементы матрицы:

$$x_u = (Y^T Y + Y^T (C^u - I)Y + \lambda I)^{-1} Y^T C^u p(u)$$

$$y_i = (X^T X + X^T (C^i - I)X + \lambda I)^{-1} X^T C^i p(i)$$

BPR

Алгоритм ALS никак не отличает ресурсы, с которыми пользователь не взаимодействовал, и ресурсы, которые ему принципиально не нравятся.



$$\begin{aligned}
 \text{BPR-OPT} &:= \ln p(\Theta | >_u) \\
 &= \ln p(>_u | \Theta) p(\Theta) \\
 &= \ln \prod_{(u,i,j) \in D_S} \sigma(\hat{x}_{uij}) p(\Theta) \\
 &= \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) + \ln p(\Theta) \\
 &= \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_{\Theta} \|\Theta\|^2
 \end{aligned}$$

Вложение пользователей в векторное пространство

- 1) x_i , полученный с помощью матричного разложения.
- 2) Агрегация всех y_i , посещенных пользователем.
- 3) Конкатенация этих вложений.

Final solution. Model #1

```
ALS_params = {'factors':40, 'iterations':120}
BPR_params = {'factors':350, 'iterations':200}
config = ['als', 'cat', 'cat', 'bpr', 'bpr']
item_user_emb = ['item', {'als':'item', 'bpr':'user'}, {'als':'item', 'bpr':'user'}, 'user', 'user']

dataset_1 = ALS_BPR_Dataset(ALS_params=ALS_params, BPR_params=BPR_params, config=config, item_user_emb=item_user_emb)

dataset_1.fit(X1_url_counter_all, X2_all)

lgbm_params_1 = [{'learning_rate':0.0017, 'n_estimators':550, 'max_depth':4, 'feature_fraction':0.75}] \
    + 4 * [{'learning_rate':0.004, 'n_estimators':760, 'max_depth':3, 'feature_fraction':0.55}]

_ = cross_validate_model(dataset_1, X1.id, Y, lgbm_params_1, use_same_params=False)
```

```
Target 1: mean = 0.6061, std = 0.0246
Target 2: mean = 0.6315, std = 0.0087
Target 3: mean = 0.6294, std = 0.0135
Target 4: mean = 0.6233, std = 0.0124
Target 5: mean = 0.6337, std = 0.0163
All targets: mean = 0.6248, std = 0.0100
```

Final solution. Model #2

```
ALS_params = {'factors':30, 'iterations':60}
BPR_params = {'factors':350, 'iterations':200}
config = ['cat', 'bpr', 'bpr', 'cat', 'bpr']
item_user_emb = [{'als':'user', 'bpr':'user'}, 'user', 'user', {'als':'user', 'bpr':'user'}, 'user']

dataset_2 = ALS_BPR_Dataset(ALS_params=ALS_params, BPR_params=BPR_params, config=config, item_user_emb=item_user_emb)

dataset_2.fit(X1_url_counter_all, X2_all)
```

```
lgbm_params_2 = {'learning_rate':0.004, 'n_estimators':760, 'max_depth':3, 'feature_fraction':0.55}

_ = cross_validate_model(dataset_2, X1.id, Y, lgbm_params_2, use_same_params=True)
```

Target 1: mean = 0.6028, std = 0.0243

Target 2: mean = 0.6407, std = 0.0125

Target 3: mean = 0.6292, std = 0.0167

Target 4: mean = 0.6226, std = 0.0112

Target 5: mean = 0.6359, std = 0.0108

All targets: mean = 0.6262, std = 0.0132

Final solution. Model #3

```
ALS_params = {'factors':40, 'iterations':60}
BPR_params = {'factors':350, 'iterations':170}
config = ['cat', 'bpr', 'bpr', 'bpr', 'cat']
item_user_emb = [{'als':'item', 'bpr':'item'}, 'item', 'item', 'item', {'als':'item', 'bpr':'item'}]

dataset_3 = ALS_BPR_Dataset(ALS_params=ALS_params, BPR_params=BPR_params, config=config, item_user_emb=item_user_emb)

dataset_3.fit(X1_url_counter_all, X2_all)

lgbm_params_3 = {'learning_rate':0.004, 'n_estimators':760, 'max_depth':3, 'feature_fraction':0.55}

_ = cross_validate_model(dataset_3, X1.id, Y, lgbm_params_3, use_same_params=True)

Target 1: mean = 0.6077, std = 0.0270
Target 2: mean = 0.6335, std = 0.0146
Target 3: mean = 0.6253, std = 0.0139
Target 4: mean = 0.6233, std = 0.0184
Target 5: mean = 0.6255, std = 0.0153
All targets: mean = 0.6231, std = 0.0085
```


Final solution

Финальное решение является усреднением трех предыдущих моделей. Модели используют различные методы получения вложений пользователей, в связи с чем дают хорошее качество в ансамбле.

Такая модель даёт первое место на *private* лидерборде. Однако, она не была выбрана в качестве финального решения из-за низкого качества на *public* лидерборде.






ArtemT

10.10.2019 11:54

pub: 0.623703
priv: 0.6281

Leaderboard

Place	Team	Solutions	Award	Score
1	Mamat Shamshiev MMP MSU	28	 Gold	0.6264040
2	Artem Tsypin	38	 Gold	0.6260790
3	Polosataya	53	 Gold	0.6195900



ArtemT

08.10.2019 15:52

pub: 0.623812

priv: 0.626079



ArtemT

10.10.2019 18:03

pub: 0.624452

priv: 0.625472