Bimbo Inventory Demand

Винокуров Никита

22 октября 2016 г.

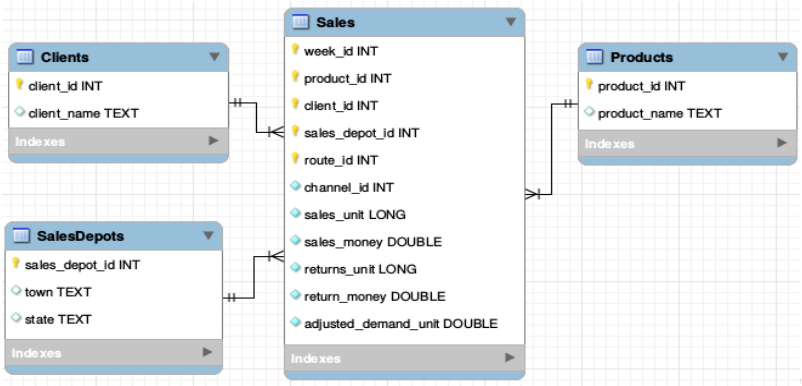
Оглавление

1. Задача, данные, результаты
2. Решения
   - Counters features
   - Linear featearus
   - Categorical embedding features

Постановка задачи и данные

Задача

Предсказать скорректированный спрос товаров для клиентов



Постановка задачи и данные

Данные: Train - 74180464(7 недель). Test - 6999251(2 недели). Объем - 3.25G

Метрика: RMSLE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(log(\widehat{y}_i + 1) - log(y_i + 1))^2}$

Особенности:

- Возвраты в 7% случаев
- Разные интервалы измерений
- Новые клиенты(1-2%) и продукты
- Уменьшение теста
- Public/Private split
- Оптимизация метрики
- Высокая кардинальность

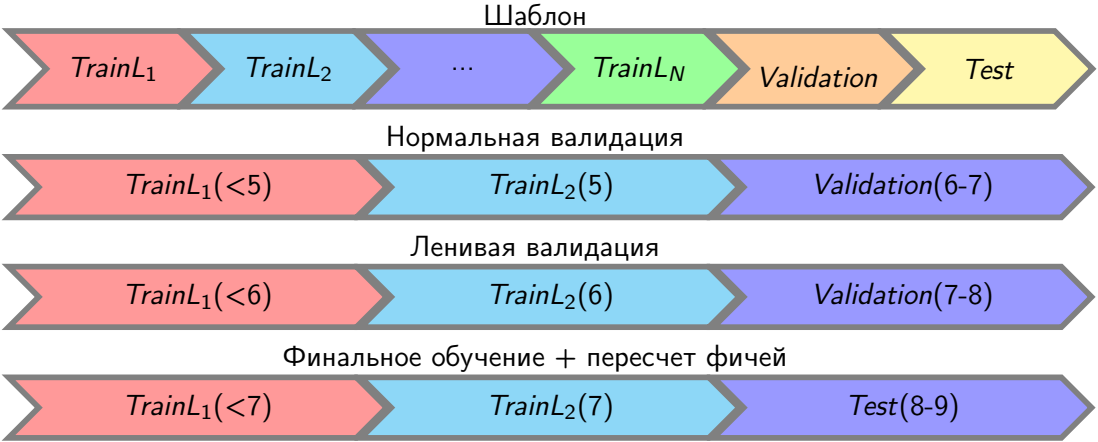
Статистика категорий:

Data type	UniqueCount
Products	1833
Clients	890267
Routes	3620
Canals	9
SalesDepots	552
Towns	260
States	33

Финальные результаты

#	Δrank	Team Name ‡ model uploaded * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑1	The Slippery Appraisals ‡ * • Stanislav Semenov • Alexander Larko • Dmitry Larko • Bohdan Pavlyshenko • Silogram	0.44260	233	Tue, 30 Aug 2016 23:14:15 (-0.4h)
2	↑1	Clustifier & Alex & Andrey ‡ * • clustifier • Alexander Ryzhkov • Andrey Kiryasov	0.44355	124	Tue, 30 Aug 2016 17:16:46 (-7h)
3	↓2	Team Mystic ‡ * • Little Boat • rcarson	0.44409	84	Tue, 30 Aug 2016 22:11:26 (-0.4h)
4	—	Gilberto & Regis • Gilberto Titericz Junior • Regis A. Ely	0.44469	123	Tue, 30 Aug 2016 23:54:48 (-0.2h)
5	—	**Beaver48**	**0.44625**	**38**	**Thu, 25 Aug 2016 07:35:57 (-26.3h)**
6	↑1	--- bimbos below this line ---	0.44729	28	Sat, 02 Jul 2016 08:23:18 (-15.9h)
7	↑14	NimaShahbazi	0.44860	31	Tue, 30 Aug 2016 22:11:08 (-0h)
8	↑10	LazyKnight	0.44921	32	Tue, 30 Aug 2016 18:46:54 (-3.5d)
9	↑13	MeanBimbo	0.44929	18	Tue, 30 Aug 2016 09:50:36 (-34.8h)
10	↓4	n_m	0.44942	14	Tue, 30 Aug 2016 13:44:20 (-0.3h)

Кросвалидация и тестирование



Counters features

Обозначения: $\{x_i^j, y_i\}$ - обучающая выборка

$C_{j_1,j_2,..,j_N}(l_1,l_2,...,l_N) = \{i, x_i^{j_1} = l_1, ..., x_i^{j_N} = l_N\}$, где $j_1, ..., j_N$ - множество фичей, а $l_1, ..., l_N$ - их значения

$C_k(x_i) = C_{j_1,j_2,..,j_N}(x_i^{j_1},x_i^{j_2},...,x_i^{j_N})$, где $k = \{j_1, j_2, .., j_N\}$

Пример:

Index	WeekId	ProductId	ClientId	RouteId	SalesDepotId
1	1	1	1	4	7
2	1	2	2	5	8
3	2	2	2	6	9

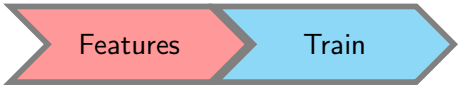
j_1 - ProductId, j_2 - ClientId, k - {ProductId, ClientId}, $C_{j_1,j_2}(1,1)$ - {1}, $C_k(x_3)$ - {2, 3}

Counters features

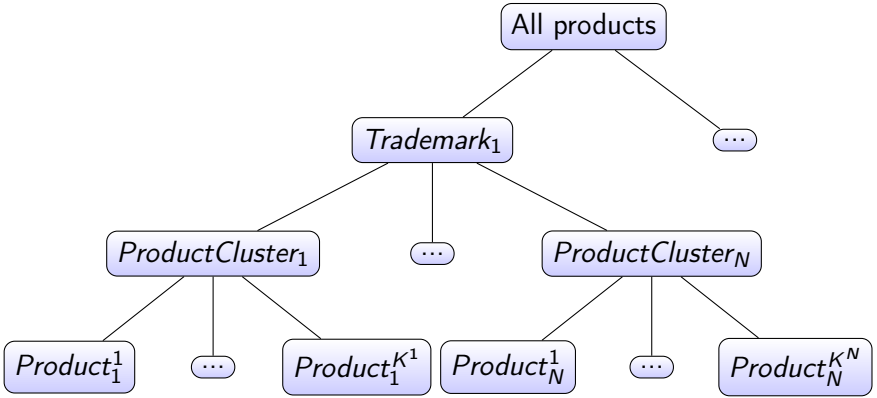
Counters: Любые статистики по подмножествам категориальных данных

- $count(C_k(x_i)) = |C_k(x_i))|$, где k пробегает все возможные подможества категориальных данных
- $mean(C_k(x_i)) = \frac{\sum\limits_{i \in C_k(x_i)} y_i}{|C_k(x_i))|}$
- $median(C_k(x_i)) = median(y_i, i \in C_k(x_i))$
- $mean(C_k(x_i))$ if $C_k(x_i) \neq \emptyset$ else $mean(C_{k_1}(x_i))$, где $C_{k_1}(x_i) \subset C_k(x_i)$
- $mean(C_k(x_i)) + \frac{\alpha}{|C_k(x_i))|+\alpha} * (mean(C_{k_1}(x_i)) - mean(C_k(x_i)))$, где $k1 \subset k$
- ...

Замечания:



Counters features



Counters features

Algorithm	LBRank	Public	Private
GBRT + BaseCounters	39	0.443	0.45703
GBRT + HierCounters	20	0.439	0.45325

Замечания:

- Кластерный GridSearch
- Пересчет фичей
- Глубокие деревья
- NA
- Ensembling не помогает

Linear* featearus

Кодирование признаков:

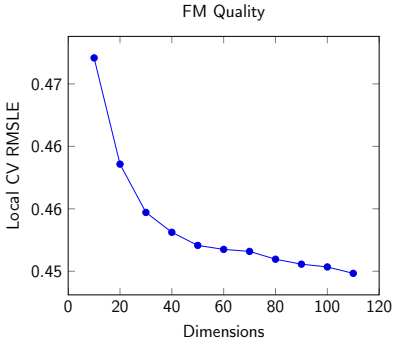
- **OneHot**
- FeatureHashing

ML Алгоритмы:

- LinReg
- **Factorization Machine**

Замечания:

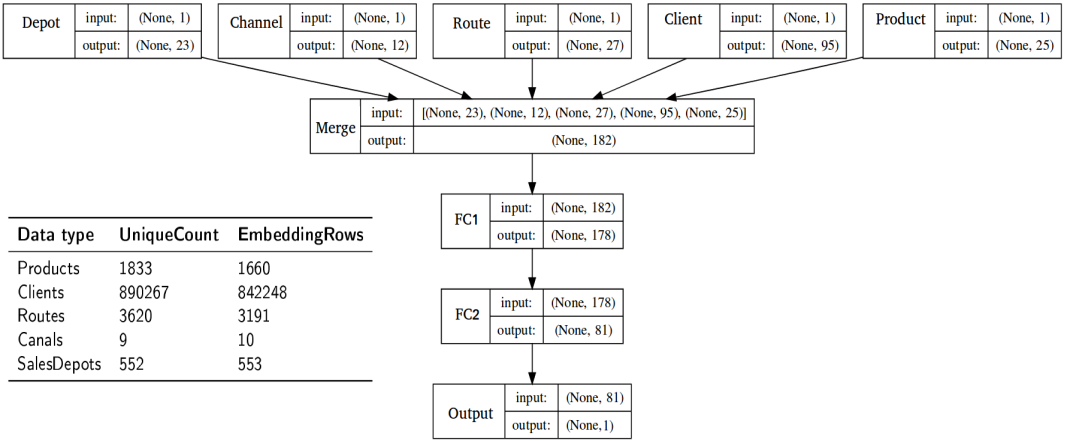
- Кластерный GridSearch
- Factorization Machine MCMC
- Per-parameter updates
- FTRL



Linear* featearus

Algorithm	LBRank	Public	Private
GBRT + BaseCounters	39	0.443	0.45703
GBRT + HierCounters	20	0.439	0.45325
FM + AllData	37	0.442	0.45676
GBRT + HierCounters + FMPred	15	0.436	0.45042

Categorical embedding



Data type	UniqueCount	EmbeddingRows
Products	1833	1660
Clients	890267	842248
Routes	3620	3191
Canals	9	10
SalesDepots	552	553

Categorical embedding

Algorithm	LBRank	Public	Private
GBRT + BaseCounters	39	0.443	0.45703
GBRT + HierCounters	20	0.439	0.45325
FM + AllData	37	0.442	0.45676
GBRT + (HierCounters + FMPred)	15	0.436	0.45042
NeuralNet + (Last3WeekData + BestCountersFromGBRT)	48	0.444	0.45890
GBRT + (HierCounters + CategEmbedding)	5	0.434	0.44720
GBRT + (HierCounters + FMPred + CategEmbedding)	5	0.432	0.44625

Замечания:

- RandomSearch
- Per-parameter updates
- TrashEmbedding

Другие идеи

- Несколько целевых переменных
- Noisy counters не работают
- Решение '0' места

Спасибо!