

# RecSys Yelp 2013 Contest: prediction of the user's relation to the business

VLADIMIR NIKULIN\*

Department of Mathematical Methods in Economy,  
Vyatka State University, Kirov, Russia

## Abstract

Business categories, which are available for both train and test parts of the data represent key elements in the proposed recommender system. Using relations between users and businesses, where star ratings are not necessary, we can transfer categories to users (compute sums of categories). Plus, we computed for users averages in the terms of latitudes and longitudes. As a next step, we found expression of any category in the terms of votes and checkins (week-day and time). Further, we calculated expression of any user and business in the terms of votes and checkins, see Sections 2.3 and 2.4. It was noticed that  $\hat{s}(user, business)$  - user's rating (named, also, as a star) of the business, maybe predicted with high quality in the case if we know average stars and numbers of reviews for both input parameters: user and business. In this particular project, we considered prediction of average stars (numbers of reviews are available) for users and businesses, using as an explanatory variables or features 1) locations (latitudes and longitudes), 2) numbers of reviews, 3) categories, 4) votes (processed by two different methods) and 5) checkins. As a predictor we used GBM and random Forest functions in R. Finally, we input all above data for users and businesses into review train and test sets, where we used predictions of the average stars for the test.users and test.businesses. Note that the number of sufficiently frequent categories is 354. Accordingly, it will be too difficult to load and analyse the data in the standard form of the tables. One way to overcome this problem would be transfer all data (including locations) into the sparse format. However, we decided to implement special novel transformation, which is described in Section 2.2. This transformation let us keep all the remaining blocks in a standard form. In order to calculate the final predictions, we used

---

\*Email: vnikulin.uq@gmail.com

homogeneous ensembling [1], where any single GBM-learner was based on about 10% of all data. In line with main computations, we were able to validate our model with CV-passports, see Remark 2.

## 1 Line N1 as an introduction to the proposed recommender system (case of the given averages)

The method described in Abstract represent just a Line N3. Here, we shall explain Line N1 as the most straitforward, but complete component of the proposed method.

Suppose, we would like to calculate  $\hat{s}(u, b)$ , where the pairs  $\{s(u), r(u)\}$  and  $\{s(b), r(b)\}$  are known (that means, both  $u$  and  $b$  are listed in the `train.user` and `train.business`). Then, we can implement the following formula

$$\hat{s}(u, b) = \exp\left(\frac{r(u) \cdot \log(s(u)) + r(b) \cdot \log(s(b))}{r(u) + r(b)}\right). \quad (1)$$

Otherwise, in the cases if  $s(u)$  or  $s(b)$  are known, we shall apply the following formulas

$$\hat{s}(u, b) = s(u); \quad (2a)$$

$$\hat{s}(u, b) = s(b). \quad (2b)$$

Further, suppose that  $s(u)$  or  $s(b)$  are not listed in the `train.user` and `train.business` (unknown), but listed in `test.user` or `test.business`.

Then, we can apply (1), (2a - 2b) with  $\hat{s}(u)$  and  $\hat{s}(b)$  as a replacement of  $s(u)$  and  $s(b)$ .

**Remark 1** *We did not find any single case, when both  $u$  and  $b$  are not listed in `all.user` and `all.business`, where `all.user` is a union of `train.user` and `test.user`, `all.business` is a union of `train.business` and `test.business`.*

Using Remark 1 as a motivation, we can define general formula for calculation of  $\hat{s}(u, b)$ :

$$\hat{s}(u, b) = \exp\left(\frac{r(u) \cdot \log(\tilde{s}(u)) + r(b) \cdot \log(\tilde{s}(b))}{r(u) + r(b)}\right), \quad (3)$$

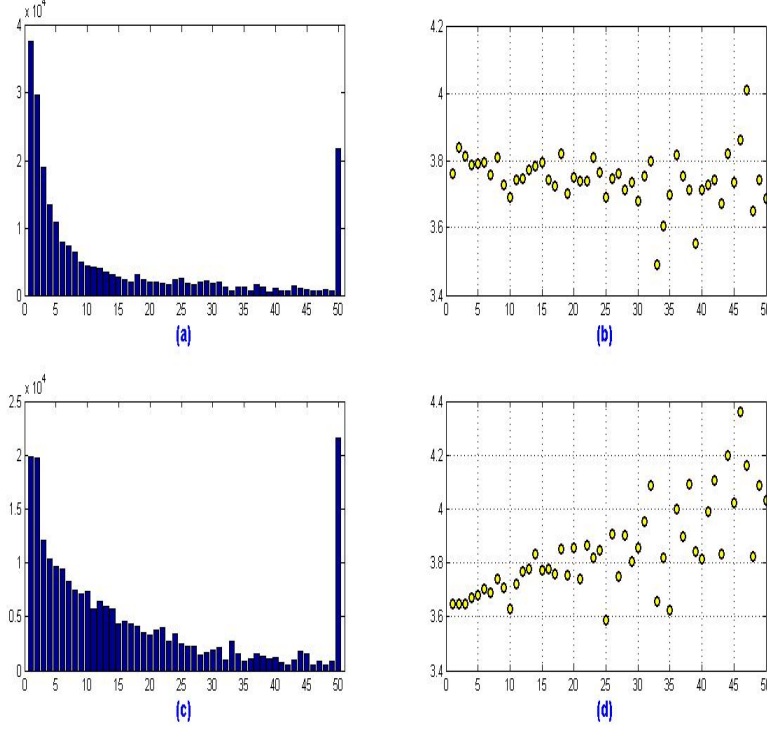


Figure 1: (a) support for users, see Section 1.1; (b) average stars for users; (c) support for businesses; (d) average stars for businesses, where all horizontal axes represent number of occurrences in the training set.

where

$$\tilde{s}(u) = \frac{I(u) \cdot s(u) + \beta \cdot \hat{s}(u)}{I(u) + \beta}; \quad (4a)$$

$$\tilde{s}(b) = \frac{I(b) \cdot s(b) + \beta \cdot \hat{s}(b)}{I(b) + \beta}, \quad (4b)$$

where  $I(u)$  and  $I(b)$  are indicators of the events  $u \in \text{train.user}$  and  $b \in \text{train.bus}$ ,  $\beta = 0.15$  is a regulation parameter.

**Remark 2** We conducted many experiments with homogeneous ensembling as described in [1]. The following results we observed in the terms of CV-passports and training error: 1) review's predictions (see Abstract)  $\{0.957, 0.8738\}$ ,

2) user's predictions  $\{0.94, 0.852\}$ , and 3) business's predictions  $\{0.7838, 0.7049\}$ . As an output, our model produces not only *test.predictions*, but, also, *train.predictions*, which maybe used for smoothing in (4a - 4b).

### 1.1 Line N2 (case of averages, extracted from the review data)

Using *train.review* data we can compute average stars for all involved businesses and users. In addition, we shall compute the numbers of reviews (support):  $\phi(u)$  and  $\phi(b)$ . Further, we can classify users and businesses according to the numbers of reviews. Clearly, average stars, which are based on small numbers of reviews are noisy and should be regularised. On the other hand, the numbers of users and businesses with small number of reviews are large (see Figure 1(a,c)), and the corresponding average stars  $z$  maybe applied as regularisers.

**Remark 3** *It is very interesting to note that users with bigger number of reviews appears to be more conservative (there is a slight decline in value of average stars  $z(\phi(u))$ , see Figure 1(b)). On the other hand, we can see that businesses which are more popular and attract greater attention (means with bigger number of reviews) are gaining higher average stars  $z(\phi(b))$ , see Figure 1(d) (smoothing parameters 3 and 6 were used for users and businesses).*

The following smoothed averages were used as an input in (1)

$$\tilde{s}(u) = (1 - \alpha)\hat{s}(u) + \alpha z(\phi(u)), \quad (5)$$

with smoothing parameter

$$\alpha = \frac{1}{(1 + \phi(u))^\gamma},$$

where  $\gamma = 0.25$  (computation of  $\tilde{s}(b)$  is an identical). Note that  $\hat{s}$  were computed using reviews data, and should not be considered as an identical to  $\hat{s}$  in Section 1.

### 1.2 Computation of the final solution (ensemble of the Lines 1,2,3)

We can link the outcomes of the Lines N1 and N2 according to the formula (1), where numbers of reviews equal to  $r(u) + r(b)$ , where  $r(u)$  and  $r(b)$  are different for the Lines N1 and N2. As a consequence, we shall produce  $ens_{12}$ .

Table 1: Some statistical characteristics of the RecSys 2013 Challenge data.

name	train	test
business	11537	2797
checkin	8282	1796
user	43873	9522
review	229907	36404

Let us denote by  $q_i = r_i(u) + r_i(b)$ ,  $i = 1, 2$ , where  $i = 1$  corresponds to the Line N1, and  $i = 2$  corresponds to the Line N2.

Then, the final solution is calculated according to the formula

$$ens_{\text{final}} = \theta ens_{12} + (1 - \theta) ens_{\text{gbm}}, \quad (6)$$

where

$$\theta = \left( \frac{10}{q_1 + q_2 + 1} \right)^\lambda, \lambda = 0.2,$$

subject to the condition that  $\theta \leq 1$ ,  $ens_{\text{gbm}}$  - heterogeneous ensemble as an outcome of the Line N3.

**Remark 4** *In order to produce  $ens_{\text{gbm}}$  we used about 30 of different homogeneous ensembles, where each of which was based on the different configuration of the database (see Abstract). In particular, we conducted experiments with date (considering test.reviews as a future), gender, city.*

In fact, we considered, also, Line N4 - matrix factorisation via stochastic gradient descent [2], but it did not produce any significant improvement. Also, we considered randomForest as a base learner.

## 2 RecSys 2013 Data

RecSys database<sup>1</sup> was given in JSON format, and includes two parts 1) training with 229907 known reviews (assessments of the businesses by customers), where 1 stands for poor and 5 - for excellent; 2) testing with 36404 unknown reviews (to be predicted). The task was to minimise RMSE - the root mean squared error.

Both datasets are divided into 4 groups (see Table 1).

---

<sup>1</sup><http://www.kaggle.com>

Table 2: Examples of categories for some businesses, where zero means empty space.

Food	Ice Cream + Frozen Yogurt	0
Pet Services	Pet Boarding/Pet Sitting	Pets
Active Life	Yoga	Fitness + Instruction
Hobby Shops	Shopping	Toy Stores
Bars	Nightlife	0
Department Stores	Fashion	Shopping
Pizza	Restaurants	0
Mexican	Restaurants	0
Tanning	Beauty + Spas	0
Auto Repair	Automotive	0
Professional Services	0	0
Greek	Restaurants	0
Food	Donuts	Coffee + Tea
Hotels + Travel	Event Planning + Services	Hotels
Chinese	Restaurants	0
Local Services	Appliances + Repair	Home Services
Auto Repair	Automotive	0
Food	Coffee + Tea	0
Banks + Credit Unions	Financial Services	Mortgage Brokers
Department Stores	Fashion	Shopping
Hotels + Travel	Event Planning + Services	Hotels
American (New)	Restaurants	0
Breakfast + Brunch	Restaurants	0
Tapas/Small Plates	Restaurants	0
Health + Medical	Dentists	General Dentistry
Arts + Entertainment	American (Traditional)	Music Venues

## 2.1 Business Group

We counted 354 sufficiently frequent categories, which were used in train.business not less than 5 times. Other categories were ignored.

**Remark 5** *The categories data are sparse. This property is a very essential: not more than 10 categories are used for the description of any business.*

Further, we formed matrix of features  $\mathcal{A}$  of  $\{11537 \times 354\}$ , where stars (or average business ratings) were used as a target variables and categories were used as a features (binary data, where one stands for present, and zero - for absent).

Using “randomForest” function in R we computed positive importance ratings of different categories (bigger value means greater importance).

Table 3: List of the most important categories, where  $n$  is the number of occurrences in the train.bus set.

Index	Business name	$n$	Rating
73	Restaurants	4503	46.6809
211	Fast Food	386	34.6272
216	Apartments	83	30.4125
138	Real Estate	118	25.6183
28	Mobile Phones	70	24.2056
212	Hotels + Travel	379	17.0649
324	Specialty Food	181	15.8686
24	Beauty + Spas	764	14.3454
301	Hotels	284	13.0453
293	Active Life	525	11.0247
175	Food	1616	10.6413
93	Internet Service Providers	22	10.6279
81	Hair Salons	154	10.6242
160	Nail Salons	242	10.5391
247	Home Services	409	10.0099
232	American (Traditional)	480	9.7582
88	Shopping	1681	9.5534
268	Drugstores	125	9.1458
126	Event Planning + Services	453	9.0172
264	Chiropractors	39	8.5157
11	Car Wash	70	8.4186
29	Performing Arts	54	7.8904
244	Health + Medical	471	7.7370
343	Fitness + Instruction	204	7.3313
166	Professional Services	71	7.2573

Table 4: Categorical data in a novel format (see Section 2.2), where any row corresponds to the particular business, “v11” - number of different categories. Columns from 1 to 10 are sequenced in a decreasing order according to the importance.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11
175	305	305	305	305	305	305	305	305	305	2
278	299	43	43	43	43	43	43	43	43	3
293	343	312	312	312	312	312	312	312	312	3
0	0	0	0	0	0	0	0	0	0	0
167	53	53	53	53	53	53	53	53	53	2
88	213	352	352	352	352	352	352	352	352	3
73	341	341	341	341	341	341	341	341	341	2
73	182	182	182	182	182	182	182	182	182	2
24	111	111	111	111	111	111	111	111	111	2
242	303	303	303	303	303	303	303	303	303	2
166	166	166	166	166	166	166	166	166	166	1
73	260	260	260	260	260	260	260	260	260	2
175	329	47	47	47	47	47	47	47	47	3
126	212	301	301	301	301	301	301	301	301	3
73	98	98	98	98	98	98	98	98	98	2
247	152	192	258	258	258	258	258	258	258	4
242	303	303	303	303	303	303	303	303	303	2
175	329	329	329	329	329	329	329	329	329	2
247	138	65	308	30	30	30	30	30	30	5
88	213	352	352	352	352	352	352	352	352	3
126	212	301	301	301	301	301	301	301	301	3
73	85	85	85	85	85	85	85	85	85	2
73	156	156	156	156	156	156	156	156	156	2
73	73	73	73	73	73	73	73	73	73	1
244	210	70	58	58	58	58	58	58	58	4
73	167	53	232	202	236	236	236	236	236	6
88	213	137	224	224	224	224	224	224	224	4
247	138	216	216	216	216	216	216	216	216	3
126	212	301	301	301	301	301	301	301	301	3
73	286	107	1	1	1	1	1	1	1	4
247	138	30	30	30	30	30	30	30	30	3



Table 5: Original checkin data (15 features), see Section 2.4.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	5	22	24	7	26	0	16	33	8	0	30	0	0	20
0	9	45	36	0	58	0	30	64	0	0	38	0	0	2
2	21	32	42	3	48	0	51	40	6	0	34	0	0	5
0	3	41	23	1	44	0	30	80	2	0	48	0	0	12
4	2	20	14	6	10	0	18	24	9	0	42	0	0	40
0	6	49	30	4	37	0	24	72	4	0	44	0	0	13
0	2	22	19	4	21	0	20	36	5	0	43	0	0	17
0	3	24	18	4	24	0	19	39	4	0	38	0	0	16
0	4	23	17	2	34	0	22	47	2	0	32	0	0	6
0	8	28	22	1	46	0	12	50	1	0	18	0	0	2
0	1	15	7	0	18	0	8	26	0	0	19	0	0	3
0	2	27	15	2	32	0	18	48	2	0	33	0	0	11
1	27	38	42	2	76	0	14	50	3	0	23	0	0	9
4	17	34	35	9	42	0	20	46	9	0	43	0	0	26
0	2	25	20	4	24	0	20	38	3	0	39	0	0	13

Table 6: Dimensional reduction: data of Table 5 after transformation (only 5 features).

9	12	6	4	3
9	6	3	12	4
8	6	4	9	12
9	12	6	3	8
12	15	9	3	8
9	3	12	6	4
12	9	3	6	8
9	12	6	3	8
9	6	12	3	8
9	6	3	4	12
9	12	6	3	8
9	12	6	3	8
6	9	4	3	2
9	12	6	4	3
12	9	3	6	8

## 2.2 Dimensional reduction (novel format for categorical variables)

This section represents one particular component of the proposed method as a main novelty of the whole recommendation system. We replaced ones in the matrix  $\mathcal{A}$  by the corresponding positive rating, zeros were kept intact. After that, we sorted any row in a decreasing order, and replaced positive values by the corresponding indexes (see Table 4). By definition, there are 11 columns in the secondary matrix of indexes  $\mathcal{B}$ , where first column contain indexes of the most important categories, second column contain indexes of the less important categories if available and so on. The maximum number of the different indexes is 10 (see Remark 5), and this number is given in 11th column. In the case if we don't know any categories for this business, all values in the corresponding row of the matrix  $\mathcal{B}$  are set to zero.

## 2.3 Transfer of the votes data to all users and businesses

We have “votes = (useful, funny, cool)” information for train.users, and the task is to transfer this knowledge to the remaining data. As it was described briefly in Abstract, we can find expression of all users in the terms of categories. Then, we can compute matrix  $\mathcal{C}$  of  $[categories, votes]$ , using data from train.users. After that, we can explain all users and businesses in the terms of votes, and compute related secondary features: for example, normalised vectors of votes or sums of votes,  $\log(\text{sums})$  and so on.

## 2.4 Transfer of the checkin data to all users and businesses

The idea to transform checkin data (hour, day of the week, number of counts) to all users and businesses is about the same as in the case of votes, see above Section 2.3. Note that checkin information is not available for all businesses in the both train and test sets.

Firstly, we applied some smoothing. We transferred 7 days of the week to only 3 different categorical values:  $\{1, 2, 2, 3, 3, 3, 1\}$ . Also, we split 24 hours into 5 sub-intervals:  $\{0 : 5, 6 : 9, 10 : 15, 16 : 19, 20 : 23\}$ . The product of 3 and 5 will give us 15 new features. Then, we computed matrix  $\mathcal{D}$  of  $[354, 15]$ , 354 categories and 15 checkins, using data from train and test businesses taking into account the numbers of checkins. After that, we can explain all users and businesses in the terms of checkins, see Table 5. Further, we applied dimensional reduction from 15 to 5 features according to the method similar to Section 2.2.

### 3 Concluding remarks

We note that some business star averages maybe easily collected manually from the Yelp web-site<sup>2</sup>. It is a fairly standard and publicly available procedure, which require no any special knowledge or skills. After that those averages maybe used in the Line N1 of the proposed method. That means, the corresponding businesses will be transferred from test.business to train.business. Consequently, we shall observe some improvement (about 0.004) of the score in the terms of the root mean squared error. We do believe that some limited data warehousing (work with and collection of the real data) as an essential element of the Contest is highly desirable in some cases, as it may help participants to understand the data better. For example, the winner of the PAKDD 2010 data mining Contest<sup>3</sup> used external data, and it was highly rewarded. Generally, it is a good idea to encourage some initiative. In fact, it is far from easy to find out what sort of data to collect, where to find those data, and how to use new data in the model.

### References

- [1] V. Nikulin, A. Bakharia and T.-H. Huang. “On the Evaluation of the Homogeneous Ensembles with CV-passports.” LNCS 7867, Springer, J.Li et al. (Eds.), PAKDD 2013 Workshops, pp. 109 - 120.
- [2] V. Nikulin and T.-H. Huang. (2012) “Unsupervised dimensionality reduction via gradient-based matrix factorization with two learning rates and their automatic updates.” Journal of Machine Learning Research, Workshop and Conference Proceedings, v. 27, pp. 181-195.

---

<sup>2</sup><http://www.yelp.com/phoenix>

<sup>3</sup><http://sede.neurotech.com.br/PAKDD2010/>