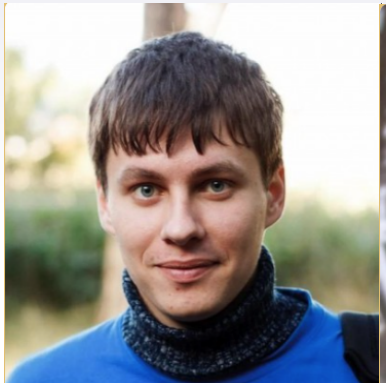
Crowdflower
Competition Analysis

Result



Mikhail Trofimov

Stanislav Semenov

Dmitry Altukhov

In the money | Gold | Silver | Bronze

#	△pub	Team Name	Score	Entries	Last
1	▲ 2	Chenglong Chen	0.72189	160	2y
2	▲ 4	Mikhail & Stanislav & Dmi...	0.71871	83	2y
3	▼ 2	Quartet	0.71861	279	2y
4	▲ 1	Shize & Shail & Phil	0.71802	252	2y
5	▲ 8	I love Phở Bò	0.71700	48	2y
6	▼ 2	Gzs_iceberg	0.71681	122	2y
7	▲ 1	YDM	0.71374	283	2y
8	▲ 10	A & A & G	0.71297	229	2y
9	▲ 7	ë	0.71265	96	2y
10	▲ 4	Alexander D'yakonov (PZ...	0.71262	93	2y

Agenda

We will discuss:

- Problem formulation
- Data
- Metric
- Basic solution
- Advanced features and tricks

Problem formulation

CrowdFlower

https://www.kaggle.com/c/crowdflower-search-relevance

Assessors' UI

Query

Understand Intent:

The search term is:

laptop lenovo

Google Search for laptop lenovo

Result

Product Title:

Lenovo ThinkPadT420 Intel Corei5 2.5GHz 4GB 750GB 14in Wi-Fi DVDRW CAM Windows 7 Professional (64-bit) (Refurbished)

$354.99

Result

Product Image:

Product Page

How well does this result match the query?

Off Topic

- The intent of the query was not matched
- The results are irrelevant to the search query

Acceptable

- The intent of the query is poorly matched
- The result is somewhat related to the query, but it not a good match

Good

- Matches most of the query intent - or the most important part of the query.
- Technically, all parts of the intent are satisfied but result doesn't provide a full, clear and complete answer to the search.

Excellent

- The query intent is clearly satisfied. This is exactly the product I was looking for
- Result is high quality
- Specifics of the Query appear in the Result

Data

```
pd.read_csv('./data/train.csv')[:4]
```

	id	query	product_title	product_description	median_relevance	relevance_variance
0	1	bridal shower decorations	Accent Pillow with Heart Design - Red/Black	Red satin accent pillow embroidered with a hea...	1	0.000
1	2	led christmas lights	Set of 10 Battery Operated Multi LED Train Chr...	Set of 10 Battery Operated Train Christmas Lig...	4	0.000
2	4	projector	ViewSonic Pro8200 DLP Multimedia Projector	NaN	4	0.471
3	5	wine rack	Concept Housewares WR-44526 Solid-Wood Ceiling...	Like a silent and sturdy tree, the Southern En...	4	0.000

Metric

- Quadratic weighted kappa
- Typical value range: from 0 (random) to 1 (complete agreement)
- May go below 0

In order to understand this metric, let's see how to calculate it.

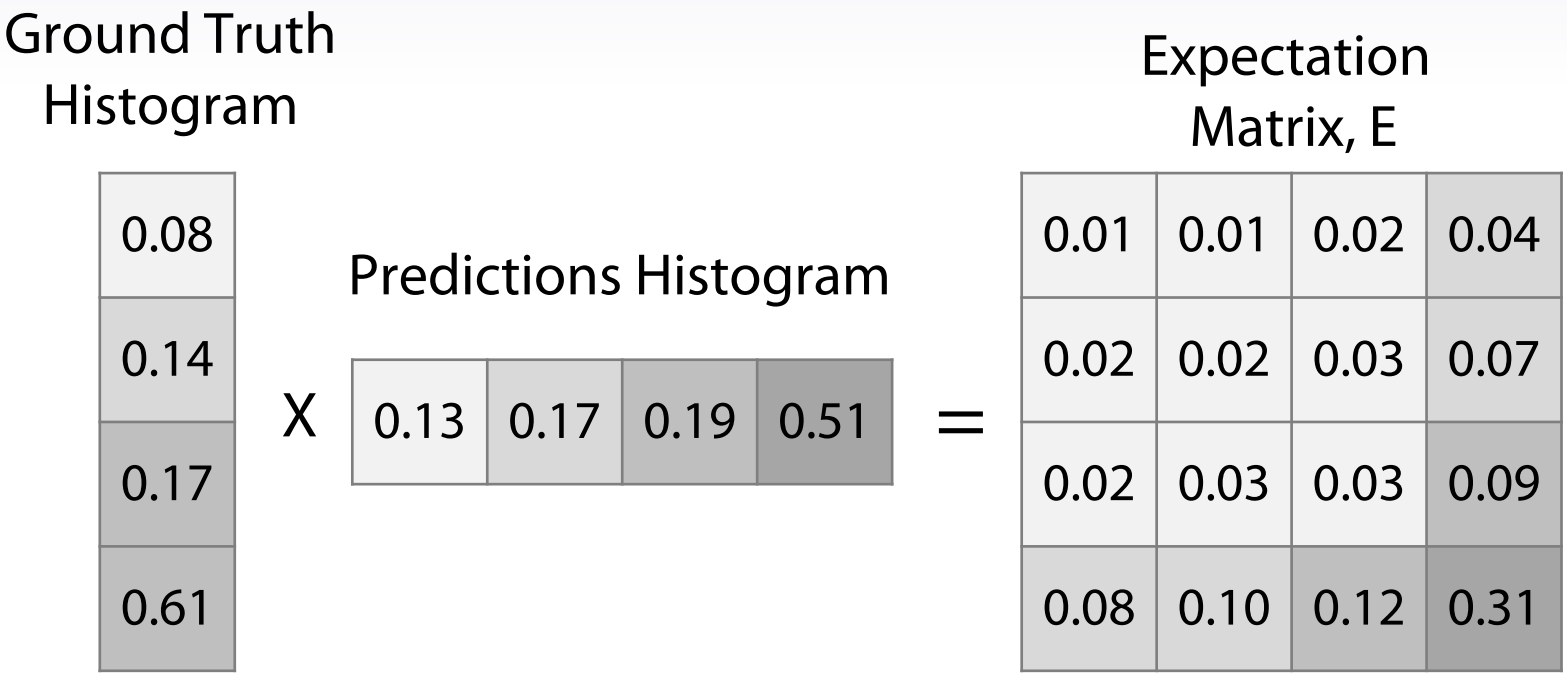
Quadratic Weighted Kappa: C

Normalized
Confusion Matrix, C

0.07	0.00	0.01	0.00
0.01	0.12	0.01	0.01
0.01	0.01	0.15	0.01
0.04	0.03	0.04	0.50

Ratings have N=4 possible values

Quadratic Weighted Kappa: E



Quadratic Weighted Kappa: W

Wight Matrix, W

0.00	0.11	0.44	1.00
0.11	0.00	0.11	0.44
0.44	0.11	0.00	0.11
1.00	0.44	0.11	0.00

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2}$$

Quadratic Weighted Kappa

Normalized Confusion Matrix, C

0.07	0.00	0.01	0.00
0.01	0.12	0.01	0.01
0.01	0.01	0.15	0.01
0.04	0.03	0.04	0.50

Expectation Matrix, E

0.01	0.01	0.02	0.04
0.02	0.02	0.03	0.07
0.02	0.03	0.03	0.09
0.08	0.10	0.12	0.31

Wight Matrix, W

0.00	0.11	0.44	1.00
0.11	0.00	0.11	0.44
0.44	0.11	0.00	0.11
1.00	0.44	0.11	0.00

$$k = 1 - \frac{\sum_{i,j}(\mathbf{W_{i,j}} * \mathbf{C_{i,j}})}{\sum_{i,j}(\mathbf{W_{i,j}} * \mathbf{E_{i,j}})}$$

Our solution

Main points:

- Text features
- Extending of queries
- Per-query models
- Sample weighting
- Bumper features
- Ensemble
- Kappa optimization

Text features: similarities

We have 3 text fields - a **query**, a **title** and a **description**.

That is, for (**query**, **title**) and (**query**, **description**) pairs we calculated:

- The number of matching words,
- Cosine distance between TF-IDF representations
- Distance between the average word2vec vectors
- Levenshtein distance

Text features: symbolic n-grams

text: Once upon a time

Once upon a time	‘Once_’
Once upon a time	‘nce_u’
Once upon a time	‘ce_up’
Once upon a time	‘e_upo’

and so on…

3 important points about data

- Queries are very short
- Number of unique queries is 261
- Queries are the same in train and test

Extending of queries

```
data[['query_original', 'query_extended']]
```

	query_original	query_extended
0	bridal shower decorations	shower bridal banner person decor mr 192 30 36 new
1	led christmas lights	light led christma white wire set green warm multi foot
2	projector	projector led home hdmi theater hd dlp lcd 1080p multimedia
3	wine rack	wine rack bottl wood wall black storag glass mount metal
4	light bulb	light bulb led pack watt white ge 4 a19 great
5	oakley polarized radar	oakley polar radar sunglass s path men len pitch replac
6	boyfriend jeans	boyfriend jean s women crop mossimo destroy distress fade glori
7	screen protector samsung	screen samsung protector galaxi insten note glass zagg s4 temper
8	pots and pans set	set cookwar piec pot pan cook stainless steel 10 non
9	waffle maker	waffl maker belgian black oster classic rotari sandwich duraceram chef
10	oakley radar	oakley radar iridium replac size pitch lens rang len vr28

Per-query models

```
train_df['query'].nunique()
```

```
261
```

```
test_df['query'].nunique()
```

```
261
```

```
len(set(train_df['query'].unique()).intersection(\
    set(train_df['query'].unique())))
```

```
261
```

Sets of queries in the train and the test are the same!

We can split our model into 261 subtasks.

Sample weighting

- We are given variance of scores for each (**query**, **product**) pair
- Large variance means low confidence about label

Sample weighting

- We are given variance of scores for each (**query**, **product**) pair
- Large variance means low confidence about label

1. Simple heuristic: $w = \frac{1}{1 + \text{var}}$

Sample weighting

- We are given variance of scores for each (**query**, **product**) pair
- Large variance means low confidence about label

1. Simple heuristic: $w = \frac{1}{1 + \mathrm{var}}$

2. Restore individual scores using median and variance statistics:
   It's possible to restore scores assuming that there are 3 raters. Example: if median=3.0 and variance=0.66 then scores probably are [2,3,4]

Sample weighting

- We are given variance of scores for each (**query**, **product**) pair
- Large variance means low confidence about label

1. Simple heuristic: $w = \frac{1}{1 + \mathrm{var}}$

   Worked quite good in this case

2. Restore individual scores using median and variance statistics:

   It's possible to restore scores assuming that there are 3 raters. Example: if median=3.0 and variance=0.66 then scores probably are [2,3,4]

   3x data, slow down training, almost no gain

Bumper features

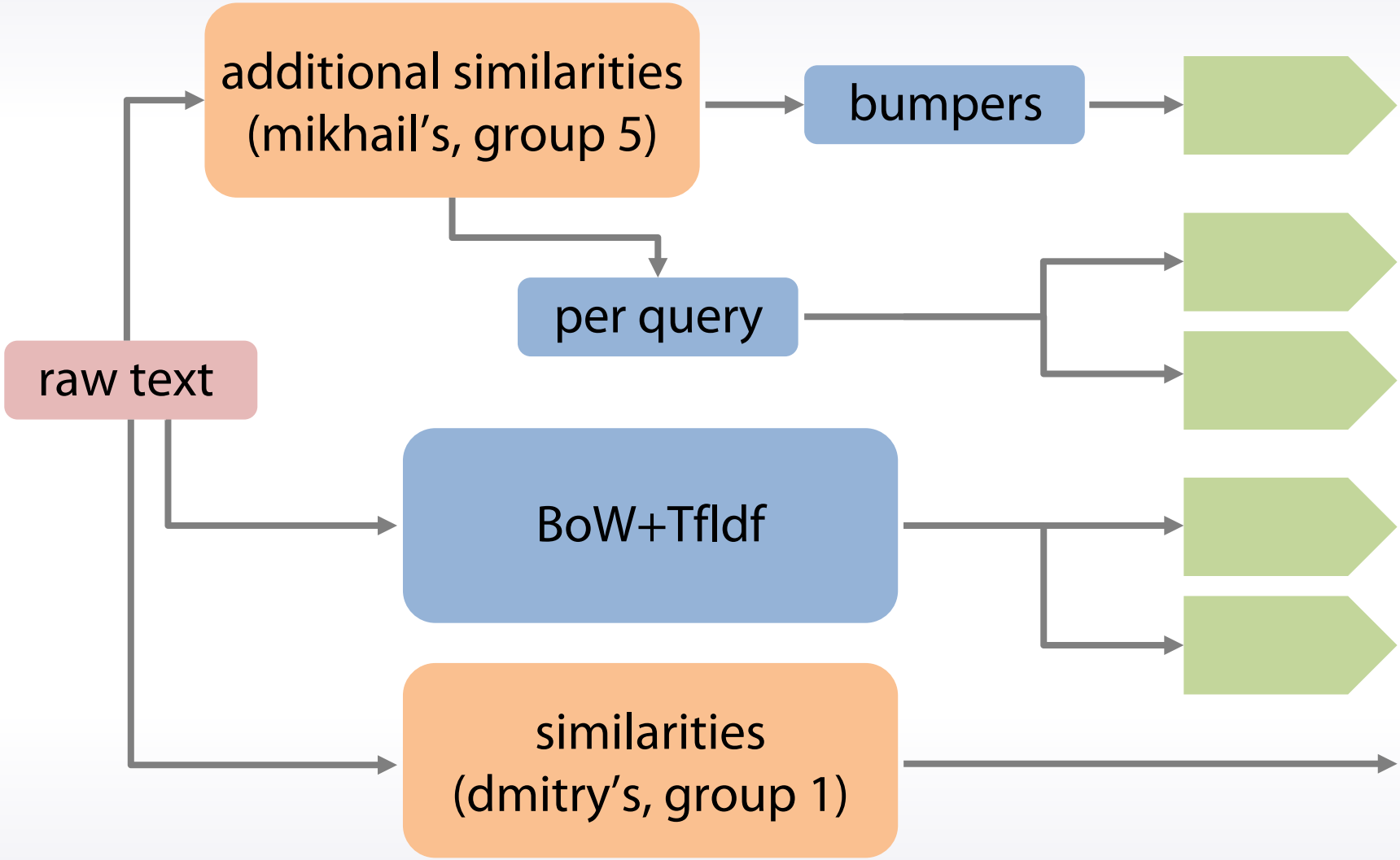
3 artificially created binary tasks used as features:

"Is it true that the target class number greater than **1**?"

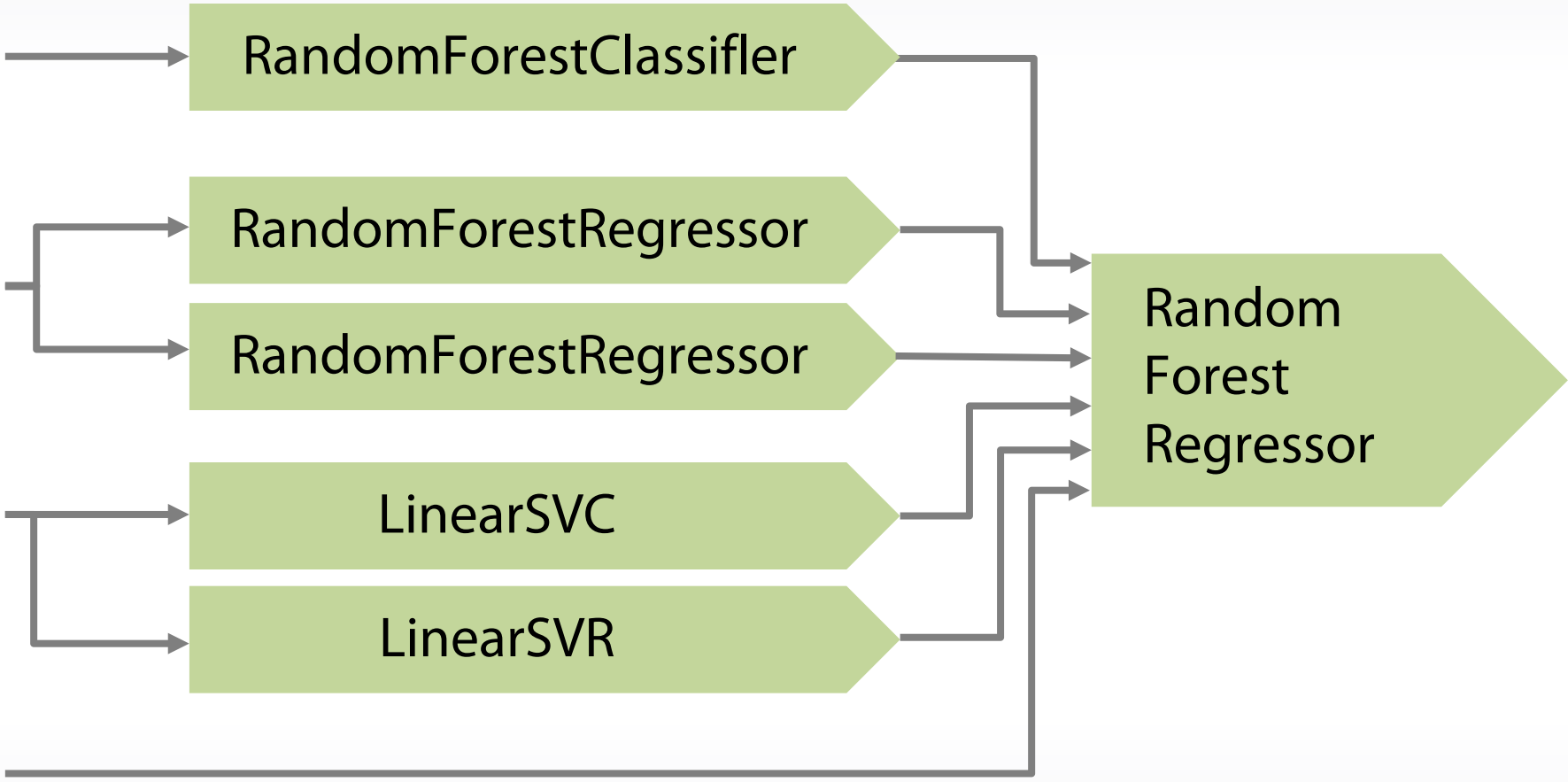
"Is it true that the target class number greater than **2**?"

"Is it true that the target class number greater than **3**?"

Ensemble


additional similarities
(mikhail's, group 5)
bumpers
per query
raw text
BoW+Tfldf
similarities
(dmitry's, group 1)

Ensemble


RandomForestClassifler
RandomForestRegressor
RandomForestRegressor
LinearSVC
LinearSVR
Random
Forest
Regressor

Kappa optimization

- Metric has properties of both classification and regression
- We consider task as regression
- Needed a way to turn real-valued predictions into classes
- We varied thresholds for every class and select the best ones

This significantly increased our score.

Conclusion

- Most important points of our solution:
- Symbolic n-grams
- Expansion of queries
- Optimization of thresholds for Kappa