

Home Depot Product Search Relevance on Kaggle

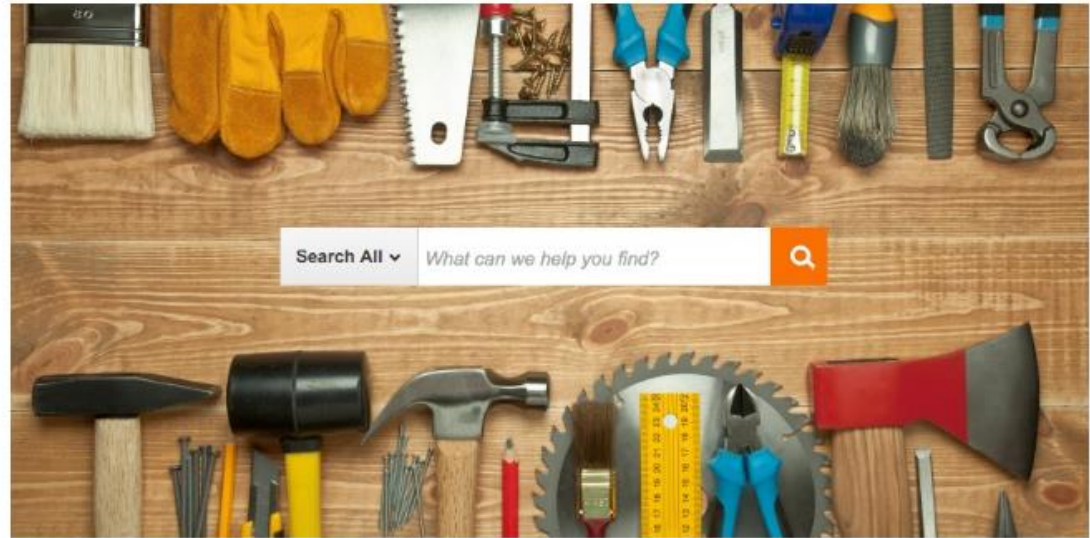
Team Turing Test
3rd place

Igor Buinyi
Kostiantyn Omelianchuk
Chenglong Chen

Presentation for
Kyiv Kaggle Trainings

2016-06-23

Predict the relevance of search results on
homedepot.com



Agenda

1. Kaggle
2. How we started kagglng
3. How we joined this competition
4. Competition: from start to finish
5. Some technical tricks

What is
Kaggle?



- Competitions
- Datasets
- Scripts
- Jobs
- Community

How we
started
kaggling?



August 2015: Read article <https://habrahabr.ru/post/264653/>

Background



Igor Buinyi



Kostiantyn Omelianchuk

- Work together in a computer games studio
- Not gamers :-)
- Knew: statistics, R, MySQL, Excel
- Learned: Python, MongoDB, Hadoop

Why joined
this
competition?



To get into top 10% (~top200 places)

HomeDepot Product Search Relevance:

Mon 18 Jan 2016 – Mon 25 Apr 2016

- 240k (*query,product*) tuples
- 24k unique queries
- 97k unique products
- train/test = 30/70
- Product info:
 - *Title*
 - *Description*
 - *Attributes (many)*
- Relevance:
 - 1 – low relevance
 - 2 – medium relevance
 - 3 – high relevance

Solution from forum

- Text preprocessing

```
def str_stem(s):  
    s = s.lower()  
    s = s.replace(",","")  
    s = re.sub('(?!<=[0-9])[ ]*centimeter[s]*(?=\ |$|\.)', '-cm ', s)  
    s = s.replace("vynal","vinyl")  
    s = stemmer.stem(s)  
    return s
```

```
df_all['search_term_stemmed'] = df_all['search_term'].map(lambda x:str_stem(x))  
df_all['product_title_stemmed'] = df_all['product_title'].map(lambda x:str_stem(x))
```


Solution from forum

- Feature extraction

```
def str_common_word(str1, str2):  
    words, cnt = str1.split(), 0  
    for word in words:  
        if str2.find(word)>=0:  
            cnt+=1  
    return cnt
```

```
df_all['len_of_query'] =  
    df_all['search_term'].map(lambda x: len(x.split())).astype(np.int64)  
df_all['len_of_title'] =  
    df_all['product_title'].map(lambda x: len(x.split())).astype(np.int64)  
  
df_all['word_in_title'] = df_all.apply(lambda x: \  
    str_common_word(x['search_term_stemmed'], x['product_title_stemmed']),axis=1)
```

Solution from forum

- Modelling

```
model = RandomForestRegressor(  
    n_estimators = 500, n_jobs = -1, random_state = 2016, verbose = 1)  
  
model.fit(X_train, y_train)  
model.predict(X_test)
```

Improving the solution

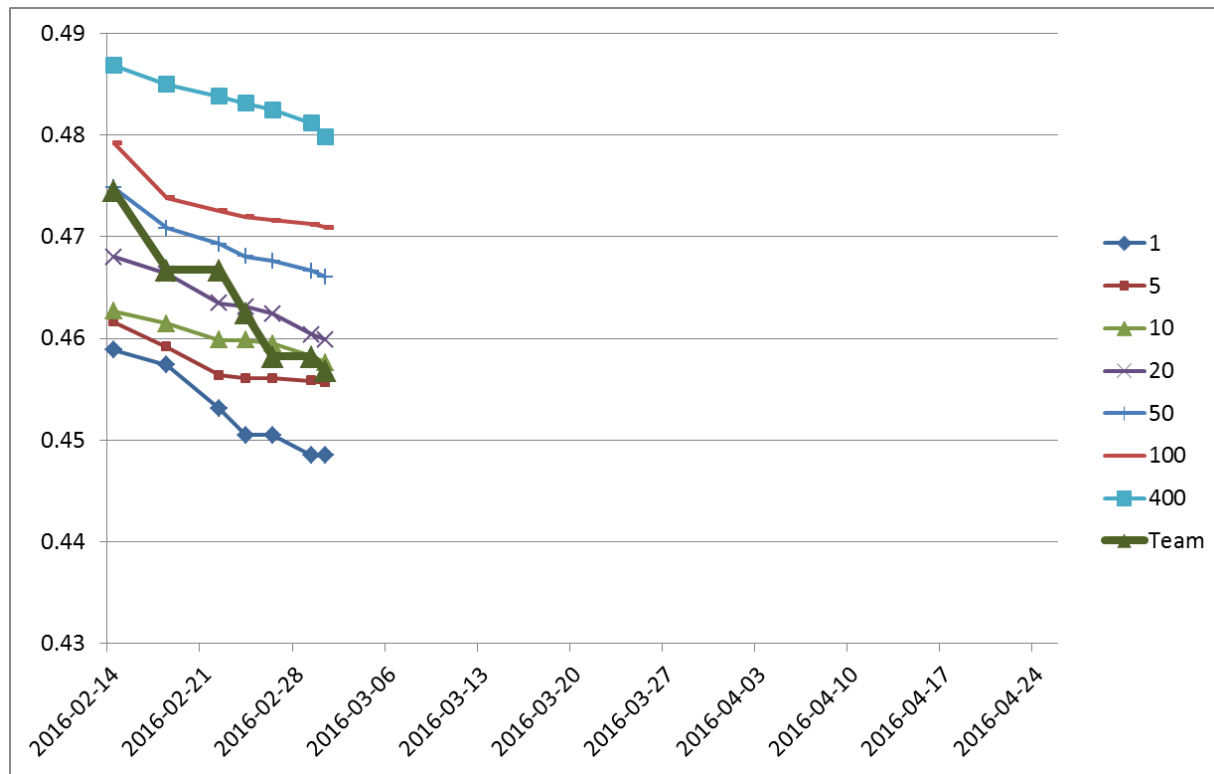
- Model parameters
- New models:
 - classification - **failed**
 - BaggingRegressor & xgboost - **OK**
- Spelling errors -> create dictionary for replacement
- New features
 - tfidf
 - Jaccard & Dice coefficients
 - various levels: word, two words, three words, characters ...
 - brands, materials

Improving the solution

12	↓2	qwerty	0.46144	27	Wed, 03 Feb 2016 11:31:07 (-3.2h)
13	↓2	TSM	0.46158	45	Sat, 30 Jan 2016 01:10:18 (-0.1h)
14	↑8	ekydna	0.46196	46	Tue, 23 Feb 2016 08:29:22
15	↑11	.relevance 👤	0.46243	37	Wed, 24 Feb 2016 06:25:24 (-40.5h)
16	↑3	Turing test 👤	0.46250	54	Wed, 24 Feb 2016 12:25:18
Your Best Entry ↑ You improved on your best score by 0.00077. You just moved up 5 positions on the leaderboard. Tweet this!					
17	↑33	Li Li	0.46270	44	Wed, 24 Feb 2016 06:38:04 (-2.1h)
18	↑16	cavallo	0.46291	10	Tue, 23 Feb 2016 22:49:57
19	↓6	Anthony Bell	0.46294	39	Sun, 21 Feb 2016 05:12:16 (-5d)
20	↑5	rheindata 👤	0.46311	52	Mon, 22 Feb 2016 23:38:11

* Final standing: 0.46250 corresponds to 131th place

Improving the solution



Entering top10

- Vocabulary
- POS tagging
- Important words

*Whirlpool 1.9 cu. ft. Over the Range **Convection Microwave** in Stainless Steel with Sensor Cooking*

- WordNet similarity
- First ensemble

Entering top10

- Vocabulary
- POS tagging
- Important words

Whirlpool 1.9 cu. ft. Over the Range **Convection Microwave** in Stainless Steel with Sensor Cooking

- WordNet similarity
- First ensemble

4	↑3	nhlx5haze	0.454/4	21	Tue, 23 Feb 2016 10:58:25 (-0.1h)
5	↓4	恭喜发财 🧑	0.45605	67	Wed, 24 Feb 2016 05:31:50 (-5.3h)
6	↑7	Jordan Goblet 🧑	0.45636	138	Mon, 22 Feb 2016 15:22:41
7	↓3	Andre Naef	0.45702	42	Fri, 26 Feb 2016 00:36:08
8	↑20	Turing test 🧑	0.45823	59	Fri, 26 Feb 2016 13:17:13
Your Best Entry ↑ Top Ten! You made the top ten by improving your score by 0.00427. You just moved up 13 positions on the leaderboard. Tweet this!					
9	↓4	Vicens Gaitan	0.45834	49	Wed, 24 Feb 2016 13:49:00 (-29.6h)
10	↓1	qianqian	0.45946	30	Tue, 23 Feb 2016 01:46:27 (-35.2h)
11	↑5	Li Li	0.45949	48	Fri, 26 Feb 2016 07:41:45

* Final standing: 0.48823 corresponds to 86th place

Entering top10

- Vocabulary
- POS tagging
- Important words

*Whirlpool 1.9 cu. ft. Over the Range **Convection Microwave** in Stainless Steel with Sensor Cooking*

- WordNet similarity
- First ensemble

Твіти

Твіти й відповіді

Медіа



Igor Buinyi @buinyi · 26 лют.

Whether boosting or bagging, I'm top 10 and #bragging. #kaggle
kaggle.com/c/home-depot-p...



Moving to the 1st place

- Vocabulary
- Improve features
- Word2vec
- Better ensemble

Moving to the 1st place

- Vocabulary
- Improve features
- Word2vec
- Better ensemble



540,000 • 1,468 teams

Home Depot Product Search Relevance

Merger and 1st Submission Deadline

Mon 18 Jan 2016



Mon 25 Apr 2016 (42 days to go)

Dashboard ▼

Public Leaderboard - Home Depot Product Search Relevance

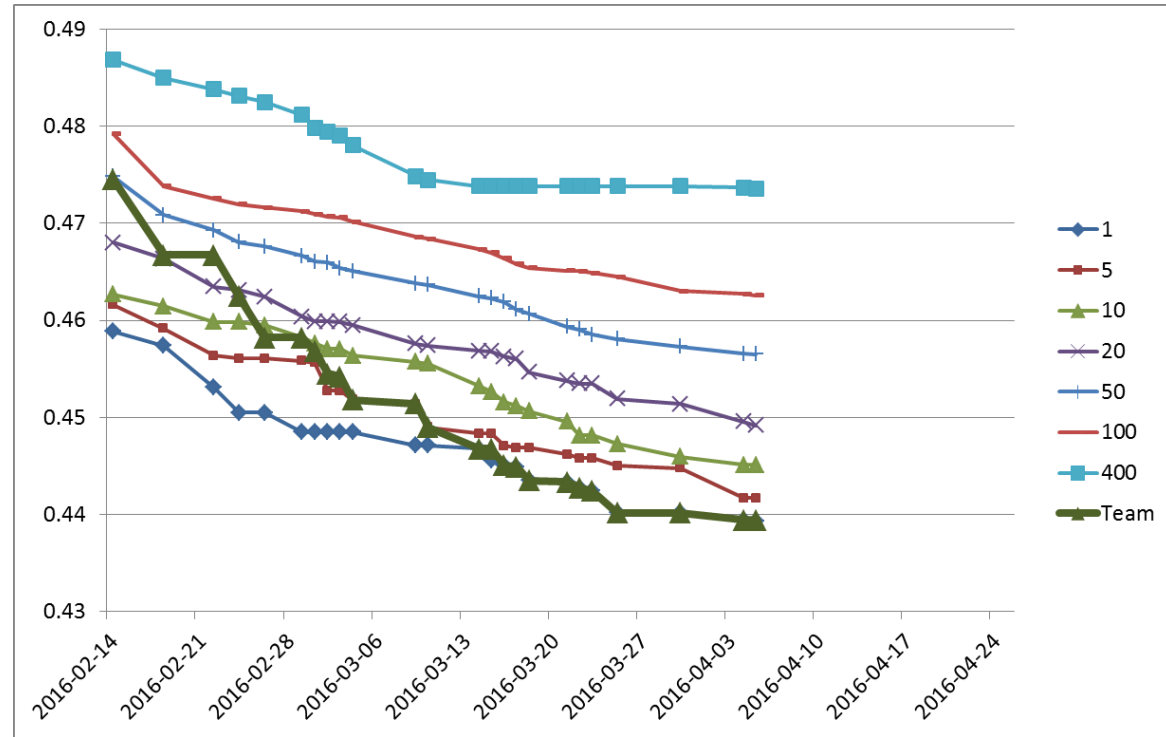
This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

See someone using multiple accounts?
[Let us know.](#)

#	Δ1w	Team Name	Score	Entries	Last Submission UTC (Best - Last Submission)
1	↑4	Turing test 	0.44675	82	Mon, 14 Mar 2016 08:03:16
Your Best Entry ↑ Number One! You jumped into first by improving your score by 0.00220. You just moved up 4 positions on the leaderboard. Tweet this!					
2	↓1	Home Improvement 	0.44688	22	Sun, 13 Mar 2016 21:06:33
3	↓1	Silogram *	0.44690	104	Mon, 14 Mar 2016 06:15:42
4	↓1	ekydna	0.44817	60	Wed, 09 Mar 2016 15:18:05
5	↑1	yang jiao	0.44832	66	Fri, 11 Mar 2016 17:17:40 (-32.8h)
6	↑1	telepathize	0.44997	76	Mon, 14 Mar 2016 06:55:17
7	↑1	nhlx5haze	0.45115	39	Sat, 12 Mar 2016 09:46:49

* Final standing: 0.46675 corresponds to 21th place

Improving the solution



* Final standing: 0.46250 corresponds to 131th place

Clarification of rules

```
import requests
import re
import time
from random import randint

START_SPELL_CHECK = "<span class='spell'>Showing results for</span>"
END_SPELL_CHECK = "<br><span class='spell_orig'>Search instead for"

HTML_Codes = (
    ('"', '&#39;'),
    ('"', '&quot;'),
    ('>', '&gt;'),
    ('<', '&lt;'),
    ('&', '&amp;'),
)

def spell_check(s):
    q = '+'.join(s.split())
    time.sleep( randint(0,2) ) #relax and don't let google be angry
    r = requests.get("https://www.google.co.uk/search?q="+q)
    content = r.text
    start=content.find(START_SPELL_CHECK)
    if ( start > -1 ):
        start = start + len(START_SPELL_CHECK)
        end=content.find(END_SPELL_CHECK)
        search= content[start:end]
        search = re.sub(r'^<[^>]+>', '', search)
        for code in HTML_Codes:
            search = search.replace(code[1], code[0])
        search = search[1:]
    else:
        search = s
    return search ;
```

```
spell_check_dict={
    'steele stake': 'steel stake',
    'gas mowe': 'gas mower',
    'metal plate cover gcfi': 'metal plate cover gfci', ...}
```

Clarification of rules

12



Chenglong Chen

3 months ago

Mon Mar 14 2016 16:57:01 GMT+0200 (Финляндия (зима))

I was thinking to build my own spelling checker using only the data at hand :) This is nice!

[Can the Admins confirm whether we are allowed to use Google spelling correction this way?](#)

[permalink](#)

0



Mattias Fagerlund

3 months ago

Sweet! If I use this, do I need to notify people in the disclosure thread? <https://www.kaggle.com/c/home-depot->

3



William Cukierski

3 months ago

Sat Mar 19 2016 17:14:06 GMT+0200 (Финляндия (зима))

Hey guys, thanks for your patience on response times.

Using Google would indeed be against the rules, which say that "[the winner] represents that he/she/it has the unrestricted right to grant that license." You have neither the code behind nor the rights to Google's secret spelling sauce.

[permalink](#)

1



Silogram

3 months ago

William, how will this be enforced? I assume that all of the winning solutions will have text replacements in the code. How will you know whether these corrections were inspired by Google, or perhaps some other proprietary spell-checker?

Clarification of rules

1



Silogram

3 months ago

Sat Mar 19 2016 22:34:41 GMT+0200 (Финляндия (зима))

Well, now I'm really confused. I'm assuming that something like:

```
s = s.replace('gallons', 'gal')
```

is OK. As is:

```
s = s.replace('thisisatypo', 'this is a typo')
```

And RG seems to be OK with proprietary spell checkers, so I guess someone could put all the search terms in an MS-Word document, capture the spelling corrections and encode them as text replacements. So how is this different from using the Google API to capture text replacements? And if Google really is the one exception, how could it be enforced? Are we simply on the honor system to promise that none of our text replacements come from the Google spell checker?

3



William Cukierski

3 months ago

Mon Mar 21 2016 00:06:28 GMT+0200 (Финляндия (зима))

Hi all,

Our legal rules cover broad expectations, but they can't anticipate and include every technical facet of competitions. As such, there's often an element of interpretation and input, driven by the host, to aim the results in a fair and useful direction. Sometimes a host is most interested in broad ideas and wants to give loose permissions to be creative, sometimes the host wants a production algorithm with very tight restrictions. We know it can be ambiguous and appreciate your persistence in looking for the right way to proceed; the fact that you're in this thread asking these questions is a sign you care about the outcome beyond your personal ambitions.

I again have to defer to RG on specifics of what is allowed, but let me talk through our normal stance on these questions, which is to apply the "new data" test that Igor alludes to: if your algorithm has to predict on new data, does it still work, to roughly the same accuracy, without changing how it works?

Let's say you see the pattern "[0-9]+ lithium" several times in the training set. You suspect this can get a "V" after the number and decide to create a regex that turns "20 lithium" into "20V lithium". This is fine and we generally would not call it external data. You are finding a rule/pattern that generalizes

Preparing for the final race

What's next?

- Do not use our vocabulary
- Build automatic spell checker
- Rewrite code
- Ensure reproducibility
- Query expansion
- Some other improvements

Teaming up



Chenglong Chen

- **1st place in CrowdFlower Search Results Relevance Competition.**

Teaming up



Chenglong Chen

- **1st place in CrowdFlower Search Results Relevance Competition.**

Re: Email from Kaggle User 'KostiaOmelianchuk'

Входящие x



陈成龙 <c.chenglong@gmail.com>

кому: мне ▾



английский ▾



русский ▾

[Перевести сообщение](#)

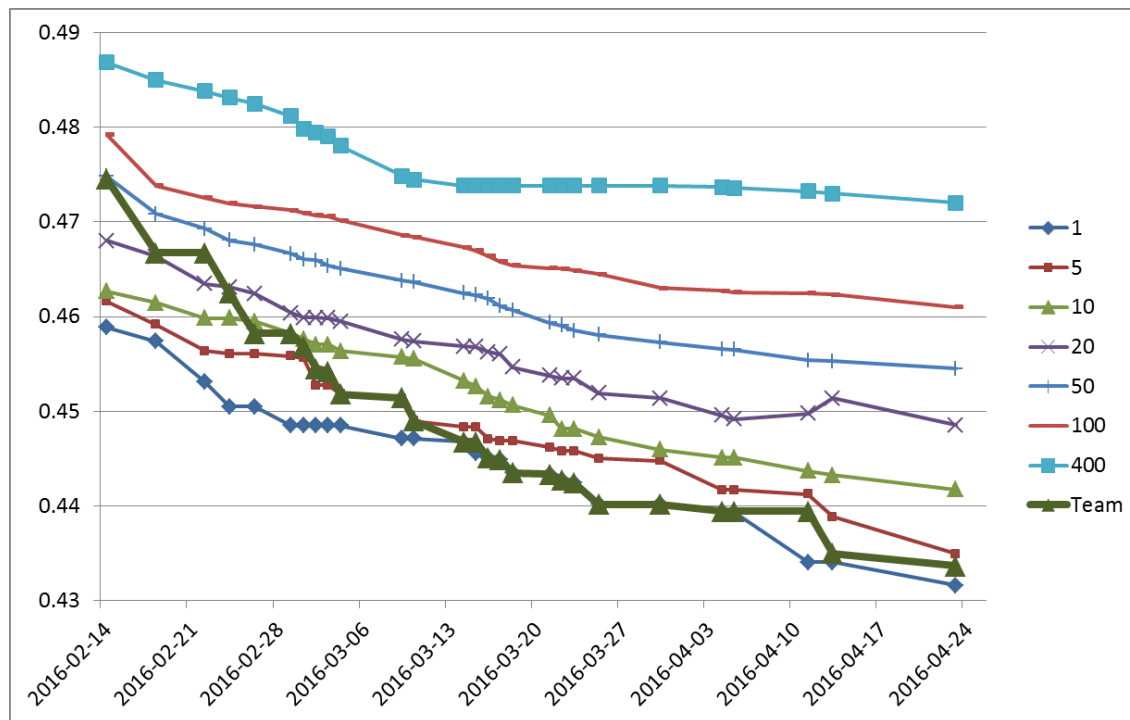
[От](#)

Hi,

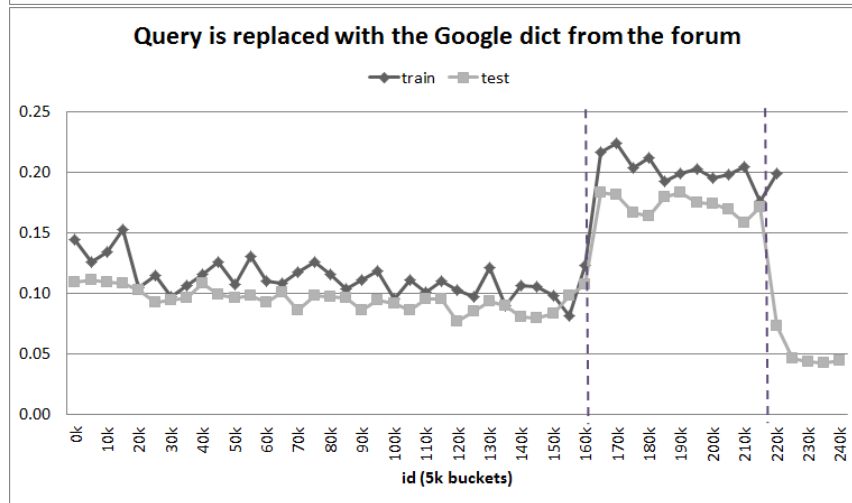
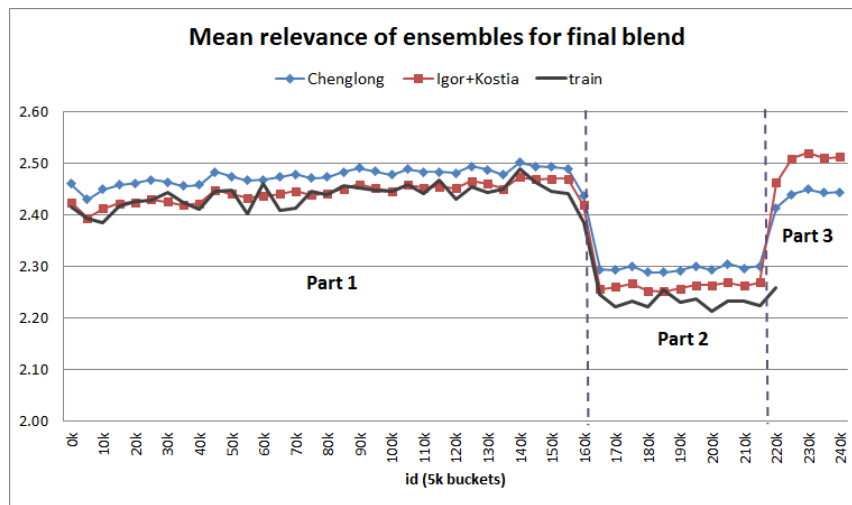
Sure! Just send me a team merge request :) We will discuss in details later.

Chenglong

Standings
a few days
before the
deadline



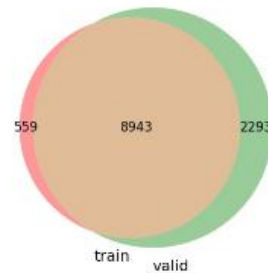
Patterns in the dataset



Cross-validation from Chenglong



(A) ACTUAL SEARCH TERM



(B) NAIVE SPLIT SEARCH TERM



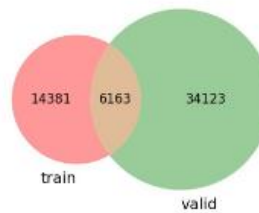
(C) PROPOSED SEARCH TERM



(D) ACTUAL PRODUCT UID



(E) NAIVE SPLIT PRODUCT UID



(F) PROPOSED PRODUCT UID

Competition Finish

- 3rd public -> 4th private
- We would have been in the 2nd place if we had submitted our top public leaderboard model
- The top team was disqualified the next day
- We moved to the 3rd place



What we learned?

- Diligent work is key for building good model
- No 'magic' tricks work (in this competition)
- Winning = skill + luck
- The solution can be simpler!
- Python is good at handling large amounts of data
- We are doing the right thing

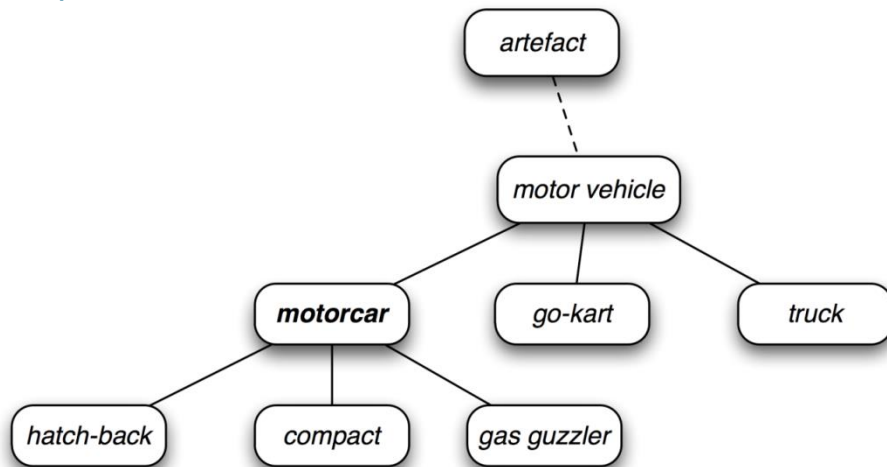
Technical stuff: word2vec

- Word2vec points
 - vector representation of words
 - takes into account the text around the considered word
- Our insights about word2vec
 - model parameters tuning – **not help a lot**
 - different vocabs – **not help a lot**
 - pretrained models – **probably not OK in this case**
 - simple n_similarity between st and pt – **OK**
- Code example:

```
from gensim import models
st = df_all["search_term_stemmed"]
pt = df_all["product_title_stemmed"]
#build vocab
for i in range(len(st)):
    vocab1.append(st[i].split()+pt[i].split()) #...pd[i].split+at[i].split()...
#fit model
model = gensim.models.Word2Vec(vocab1, sg=1, window=10, sample=1e-5, negative=5, size=300)
#calculate similarity
d1.append(st[i].split()[j])
d2.append(pt[i].split()[j])
n_sim_pt.append(model.n_similarity(d1,d2))
```

Technical stuff: WordNet

- WordNet
 - words linked into a hierarchy
 - a collection of synonym words is called lemma or synset
 - more lexical relations (holonyms, entailments, antonyms...)
 - you can calculate distance between words using this hierarchy
- Wordnet similarity
 - path_similarity, lch_similarity, wup_similarity, res_similarity and others
 - try them all



Source: O'Reilly. *NLP with Python*.

Technical stuff: Ensembling basics

- Ensemble: fitting first level models
 - make correct split and fix it (we have not do it :()
 - use different models with different parameters
 - use different samples of your data
 - make sure you can reproduce it!
- Ensemble: greedy selecting
 - you need not use all your models on the second level
 - greedy selecting helps, but might lead to overfitting
- Code example

```
ExtraTreesRegressor(n_estimators = 400),
BaggingRegressor(base_estimator=xgb.XGBRegressor(**xgb_params0), n_estimators=10) ,
RandomForestRegressor(n_estimators=500, max_depth=5, min_samples_leaf=6, max_features=0.9,
min_samples_split=1, n_jobs= -1),
AdaBoostRegressor(base_estimator=None, n_estimators=250, learning_rate=0.03, loss='linear'),
neighbors.KNeighborsRegressor(128, weights="uniform", leaf_size=5),
SVR(kernel='rbf', C=0.2, gamma=0.1),
GradientBoostingRegressor(n_estimators=500, max_depth=6, min_samples_split=1, min_samples_leaf=15,
learning_rate=0.035, loss='ls', random_state=10),
xgb.XGBRegressor(**xgb_params1),
DecisionTreeRegressor(criterion='mse', max_depth=4, min_samples_split=7, min_samples_leaf=30, max_features='sqrt')
```

GOOD LUCK!