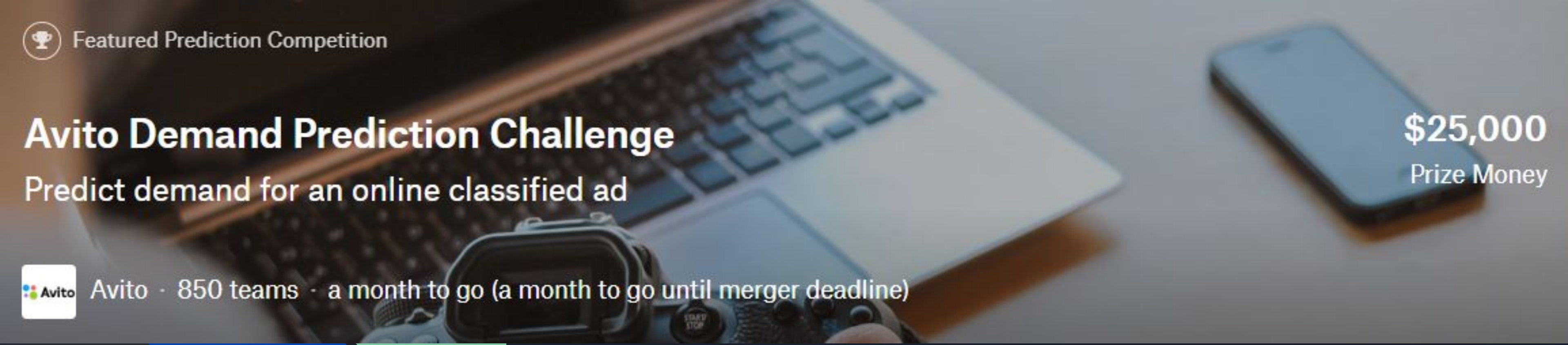
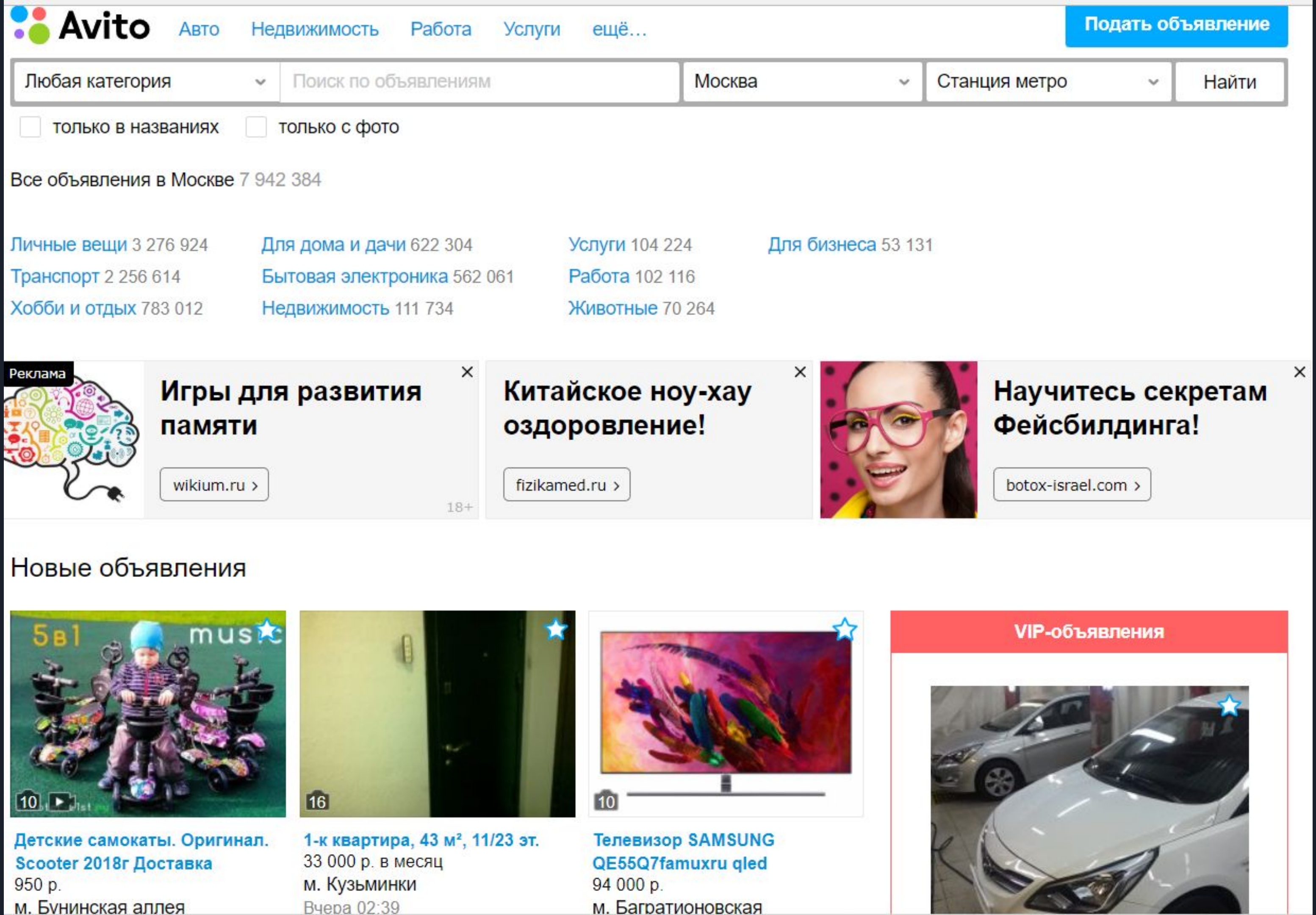


Retail website deal probability predictor

May 16th, 2018
Stephan, Matan, Itai and Ilai


Avito
Авто
Недвижимость
Работа
Услуги
ещё…
Подать объявление
Любая категория
Поиск по объявлениям
Москва
Станция метро
Найти
только в названиях
только с фото
Все объявления в Москве 7 942 384
Личные вещи 3 276 924
Транспорт 2 256 614
Хобби и отдых 783 012
Для дома и дачи 622 304
Бытовая электроника 562 061
Недвижимость 111 734
Услуги 104 224
Работа 102 116
Животные 70 264
Для бизнеса 53 131
Реклама
Игры для развития памяти
wikium.ru ›
18+
Китайское ноу-хау оздоровление!
fizikamed.ru ›
Научитесь секретам Фейсбилдинга!
botox-israel.com ›
Новые объявления
5в1
10
Детские самокаты. Оригинал. Scooter 2018г Доставка
950 р.
м. Бунинская аллея
16
1-к квартира, 43 м², 11/23 эт.
33 000 р. в месяц
м. Кузьминки
Вчера 02:39
10
Телевизор SAMSUNG QE55Q7famuxru qled
94 000 р.
м. Багратионовская
VIP-объявления

Шотландская от Интерчемпиона

29 999 руб.

м. Тушинская
Сегодня 00:31

NY12 Золотые шиншиллы

Цена не указана

м. Тимирязевская
Сегодня 00:24

Британцы с сапфировыми и изумрудными глазами

20 000 руб.

м. Охотный ряд
Вчера 23:56

Уникальные котята F1

150 000 руб.

Компания
м. Библиотека им. Ленина
Сегодня 02:38

Котята девочки Мейн кун

15 000 руб.

Теплый стан
Вчера 23:41

Снежные бенгалы

8 000 руб.

Белорусская
12 мая 13:18

Что такое VIP-объявления?

Все объявления в Москве / Животные / Кошки / Британская

Назад Следующее →

Британцы с сапфировыми и изумрудными глазами

20 000 ₽

№ 1077948413, размещено вчера в 23:56 36 (+20)

Добавить заметку

Показать телефон
8 916 XXX-XX-XX

Написать сообщение

Татьяна
Продавец
На Avito с 12 мая 2018

Адрес
Москва, м. Охотный ряд

Given data - categorical/numerical features

	item_id	user_id	region	city
0	b912c3c6a6ad	e00f8ff2eaf9	Свердловская область	Екатеринбург
1	2dac0150717d	39aeb48f0017	Самарская область	Самара
2	ba83aefab5dc	91e2f88dd6e3	Ростовская область	Ростов-на-Дону
3	02996f1dd2ea	bf5cccea572d	Татарстан	Набережные Челны
4	7c90be56d2ab	ef50846afc0b	Волгоградская область	Волгоград
5	51e0962387f7	bbfad0b1ad0a	Татарстан	Чистополь

price	item_seq_number	activation_date	user_type
400.0	2	2017-03-28	Private
3000.0	19	2017-03-26	Private
4000.0	9	2017-03-20	Private
2200.0	286	2017-03-25	Company
40000.0	3	2017-03-16	Private
1300.0	9	2017-03-28	Private

Given data - Free text

category_name	param_1	param_2	param_3	title	description
аксессуары	одежда			colins	хорошем состоянии.

Given data - Images

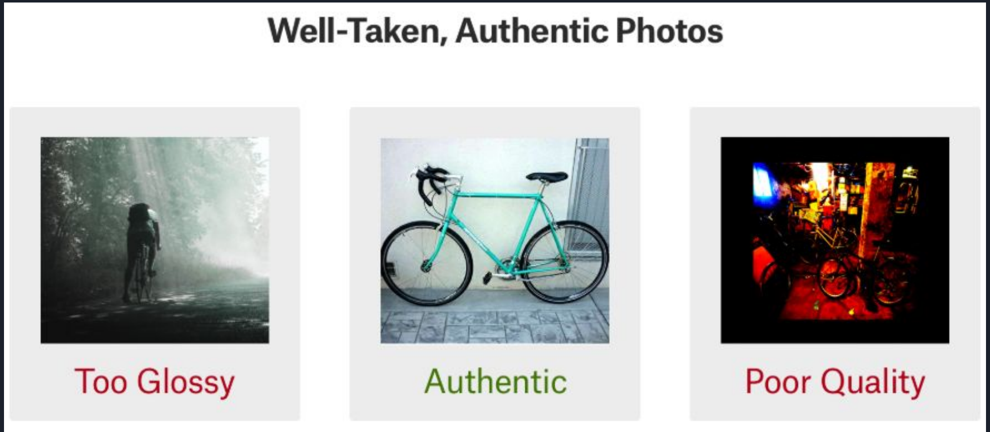


image	image_top_1
d10c7e016e03247a3bf2d13348fe959fe6f436c1caf64c...	1008.0
9c9392cc51a9c81c6eb91eceb8e552171db39d7142700...	692.0
b7f250ee3f39e1fedd77c141f273703f4a9be59db4b48a...	3032.0
6ef97e0725637ea84e3d203e82dadb43ed3cc0a1c8413...	796.0
54a687a3a0fc1d68aed99bdaaf551c5c70b761b16fd0a2...	2264.0
eb6ad1231c59d3dc7e4020e724ffe8e4d302023ddcbb99...	796.0
0330f6ac561f5db1fa8226dd5e7e127b5671d44d075a98...	2823.0
9bab29a519e81c14f4582024adfebd4f11a4ac71d323a6...	567.0
75ce06d1f939a31dfb2af8ac55f08fa998fa336d13ee05...	415.0
54fb8521135fda77a860bfd2fac6bf46867ab7c06796e3...	46.0

Given data - goal: predict **deal probability** based on the **ad parameters, text and images**

deal_probability
0.12789
0.00000
0.43177
0.80323
0.20797
0.80323
0.00000

Feature eng. - Exploration and enhancement of given features

- Found some interesting users and categories (shown in next slides)
- Added boolean features for columns with null values (has_image, has_price etc.) and checked for correlations
- Most of the items have low deal probabilities (about 80% between 0 and 0.2)
- Found strong correlation (linear and monotonic) between image_top_1 and the target feature, especially in items with high deal probability
- Further steps:
    - Data cleaning
    - Explore other data files (train_active, periods_train)
    - Investigate correlations of new features (from NLP, image processing)

In [26]:
```
power_users = new_df[(new_df.mean_deal_probability >= 0.4) & (new_df.user_count > 10) ]
print(power_users.sort_values(by = 'mean_deal_probability' , ascending = False).head(10))
weak_users = new_df[(new_df.mean_deal_probability <=0.05) & (new_df.user_count > 10) ]
print(weak_users.sort_values(by = 'mean_deal_probability' , ascending = True).head(10))
```

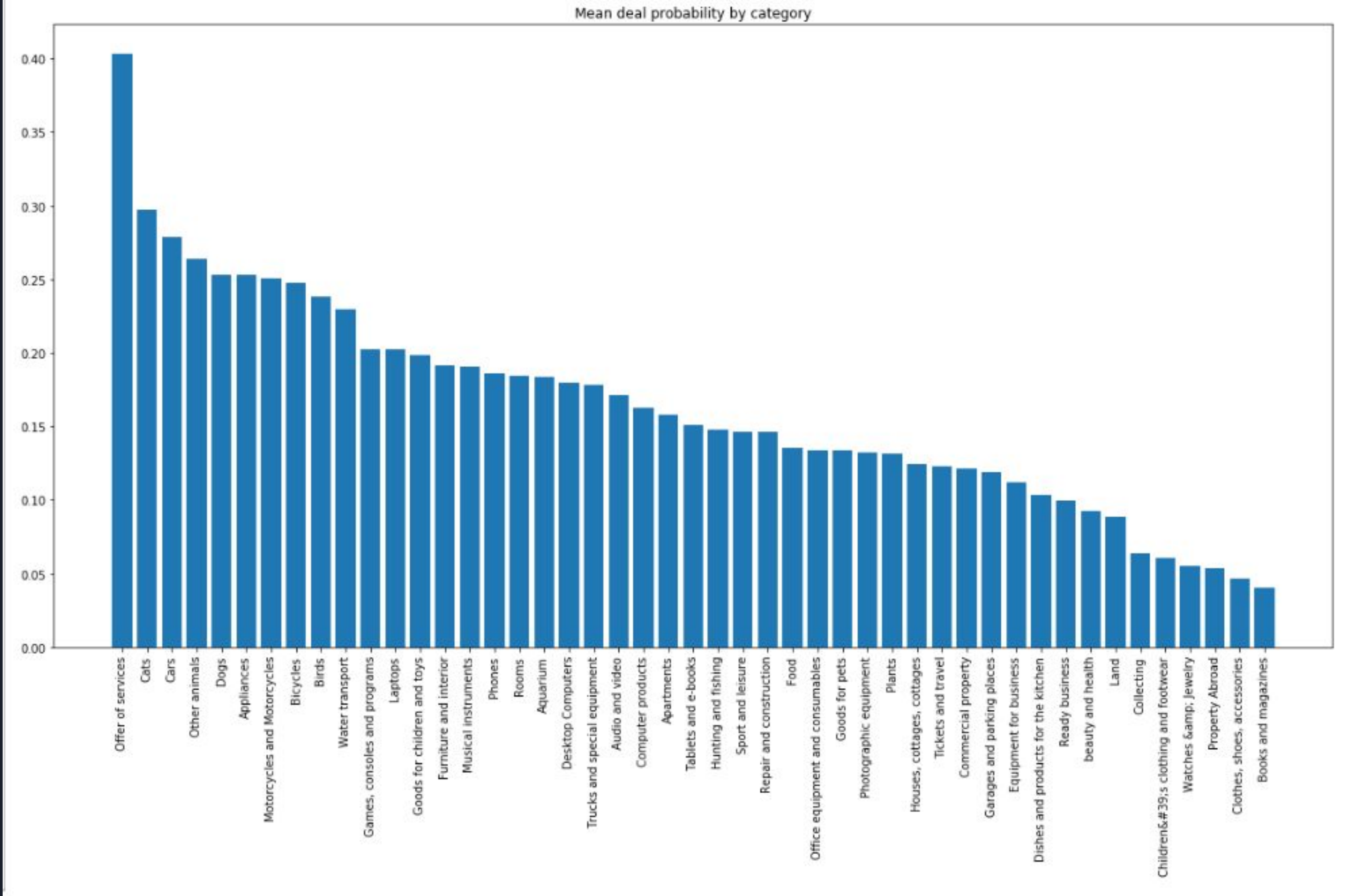
```
              mean_deal_probability  user_count
user_id
1e385206d244               0.847509          11
cc64f7af92b1               0.833261          14
813caaee2df7               0.809858          17
9b3d419e34b5               0.785030          11
55c1e63aa8b0               0.785030          12
389a4c70a84c               0.785029          12
8b07fb855dd3               0.754731          14
9872399c0349               0.741417          18
f3d5d5dd61ec               0.728956          14
d5d20e58d6d6               0.726449          11
              mean_deal_probability  user_count
user_id
4d8ebeb108ec                    0.0          11
61d48ebafbea                    0.0          11
61d4b0ef0694                    0.0          12
61d8ab6b1032                    0.0          11
bd14b235d185                    0.0          16
bd0438832fe1                    0.0          17
6214c3049fc6                    0.0          19
623a9dce3a44                    0.0          16
624fdd8e69e5                    0.0          12
bcff97cc7ecd                    0.0          14
```

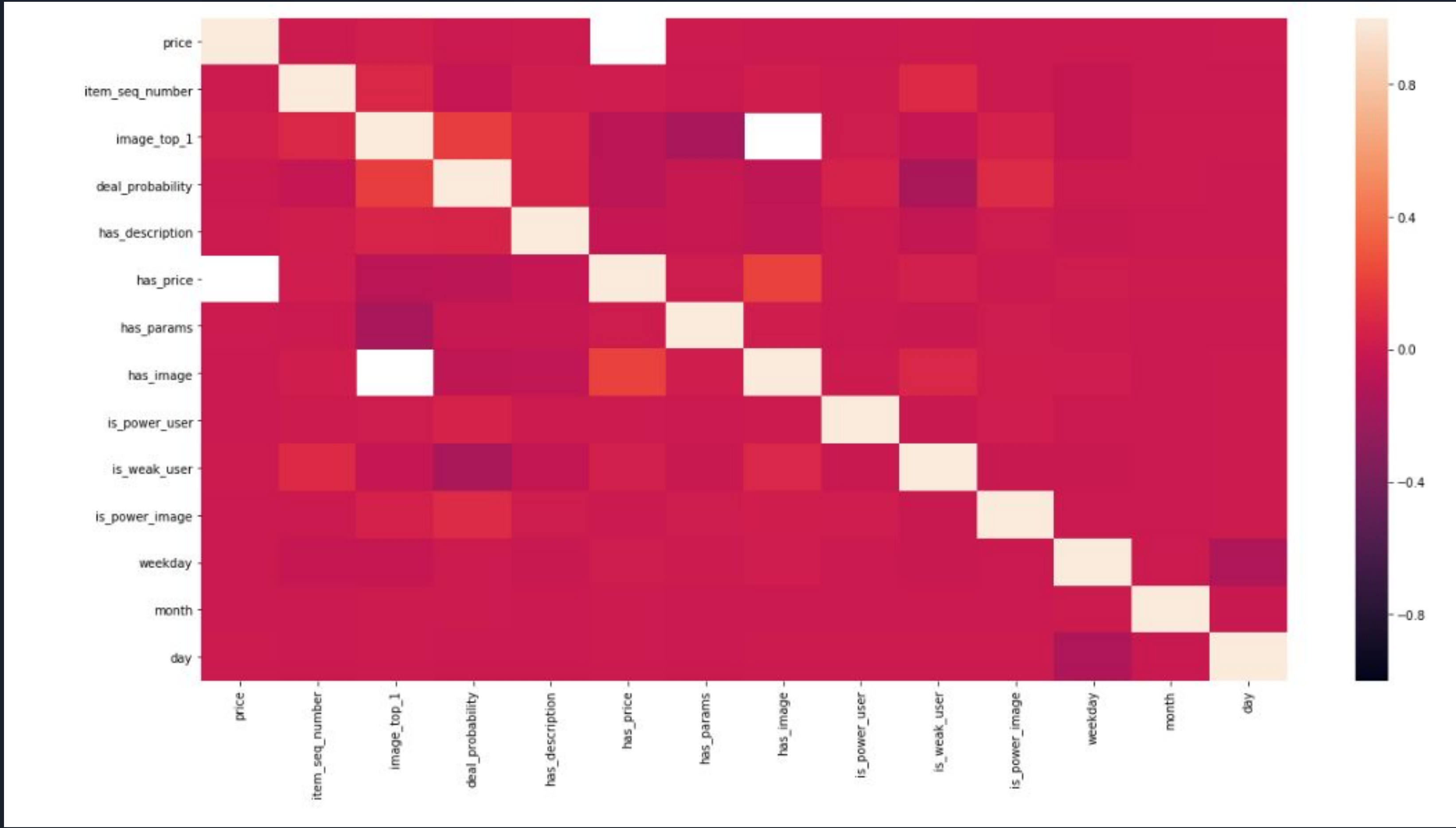
In [27]:
```
# Add a boolean feature for power users and weak users:
index_1 = list(power_users.index)
train['is_power_user'] = np.isin(train['user_id'], index_1)
print(train[train['is_power_user'] == True].shape[0]/ train.shape[0]*100)
```

```
0.15504608147801285
```

In [28]:
```
index_2 = list(weak_users.index)
train['is_weak_user'] = np.isin(train['user_id'], index_2)
print(train[train['is_weak_user'] == True].shape[0]/ train.shape[0]*100)
```

```
9.643254331446085
```


Mean deal probability by category
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Offer of services
Cats
Cars
Other animals
Dogs
Appliances
Motorcycles and Motorcycles
Bicycles
Birds
Water transport
Games, consoles and programs
Laptops
Goods for children and toys
Furniture and interior
Musical instruments
Phones
Rooms
Aquarium
Desktop Computers
Trucks and special equipment
Audio and video
Computer products
Apartments
Tablets and e-books
Hunting and fishing
Sport and leisure
Repair and construction
Food
Office equipment and consumables
Goods for pets
Photographic equipment
Plants
Houses, cottages, cottages
Tickets and travel
Commercial property
Garages and parking places
Equipment for business
Dishes and products for the kitchen
Ready business
beauty and health
Land
Collecting
Children's clothing and footwear
Watches & Jewelry
Property Abroad
Clothes, shoes, accessories
Books and magazines


price
item_seq_number
image_top_1
deal_probability
has_description
has_price
has_params
has_image
is_power_user
is_weak_user
is_power_image
weekday
month
day
0.8
0.4
0.0
−0.4
−0.8

Feature eng. - Images

- Features: size, colorfulness, dominant color, average color
- Image Quality? blurriness/sharpness, luminance
- Classifying features: score and label from different models
- Further steps:
    - Fit the classify label to ad title and description
    - Number of objects present
    - Amount of text present

Feature eng. - Images Examples

```
  img_size img_size_x img_size_y  img_blurriness  img_colorfulness  \
0   172800        360        480      841.013424         29.951734

  img_dominant_color img_dominant_blue img_dominant_green img_dominant_red  \
0    [172, 157, 171]               172                157              171

     img_color_avg  img_blue_avg  img_green_avg  img_red_avg  img_blue_std  \
0  [130, 117, 129]    129.700532     116.838519    128.81044     63.768069

   img_green_std  img_red_std                            Resnet50_score  \
0      63.397314    64.897819  [(cloak, 0.139486), (stole, 0.0892575)]

                                 xception_score  \
0  [(lab_coat, 0.508774), (trench_coat, 0.246874)]

                                 Inception_score
0  [(bow_tie, 0.988496), (trench_coat, 0.00381778)]
```


```
  img_size img_size_x img_size_y  img_blurriness  img_colorfulness  \
0   172800        360        480      170.511577         12.962352

  img_dominant_color img_dominant_blue img_dominant_green img_dominant_red  \
0          [4, 9, 2]                 4                  9                2

  img_color_avg  img_blue_avg  img_green_avg  img_red_avg  img_blue_std  \
0  [43, 49, 45]     43.157176      49.418866    44.597263      72.51686

   img_green_std  img_red_std                                Resnet50_score  \
0      71.163693    72.391195  [(web_site, 0.103203), (monitor, 0.0877461)]

                              xception_score  \
0  [(monitor, 0.409773), (screen, 0.138921)]

                                  Inception_score
0  [(modem, 0.273131), (wine_bottle, 0.0933613)]
```

Feature eng. - Images Examples

```
  img_size img_size_x img_size_y  img_blurriness  img_colorfulness  \
0   172800        480        360       65.503883          4.474525

  img_dominant_color img_dominant_blue img_dominant_green img_dominant_red  \
0       [31, 36, 39]                31                 36               39

  img_color_avg  img_blue_avg  img_green_avg  img_red_avg  img_blue_std  \
0  [24, 29, 32]     23.841748      29.403142    31.503003     10.741228

   img_green_std  img_red_std                                Resnet50_score  \
0      10.866127    11.341529  [(wardrobe, 0.235635), (crate, 0.0949693)]

                             xception_score  \
0  [(binder, 0.204239), (menu, 0.0762859)]

                               Inception_score
0  [(crate, 0.309755), (slide_rule, 0.28785)]
```

Feature eng. - Free text

	price	item_seq_number	image_top_1	deal_probability	title_word_count	description_word_count	merged_params_word
price	1.000000	0.061099	0.035071	-0.010853	0.065333	0.049828	-0.
item_seq_number	0.061099	1.000000	0.093324	-0.036068	0.132158	0.120281	-0.
image_top_1	0.035071	0.093324	1.000000	0.188871	0.247162	0.183789	-0.
deal_probability	-0.010853	-0.036068	0.188871	1.000000	0.017285	-0.001158	-0
title_word_count	0.065333	0.132158	0.247162	0.017285	1.000000	0.308280	-0.
description_word_count	0.049828	0.120281	0.183789	-0.001158	0.308280	1.000000	-0.
merged_params_word_count	-0.020851	-0.056616	-0.557352	-0.116995	-0.166665	-0.162763	1.
description_sentence_count	0.028791	0.125823	0.167183	-0.016130	0.251324	0.854852	-0.
description_words/sentence_ratio	0.017774	-0.008236	0.045569	0.050242	0.088518	0.092528	-0.
title_capital_letters_ratio	-0.033504	-0.055070	0.128419	0.021500	-0.303308	-0.010537	-0.
description_capital_letters_ratio	-0.002779	0.024389	0.087672	0.002238	0.029260	0.038988	-0.
title_non_regular_chars_ratio	0.099775	0.191032	0.189312	0.022199	0.434856	0.183368	-0.
description_non_regular_chars_ratio	0.005346	0.089587	0.083366	-0.011636	0.141690	0.291877	-0.
title_num_of_newrow_char	NaN	NaN	NaN	NaN	NaN	NaN	
description_num_of_newrow_char	0.011555	0.106225	0.159598	-0.024514	0.197092	0.755349	-0.
title_num_adj	0.006099	0.027831	-0.077893	-0.044298	0.291463	0.079471	0.
title_num_nouns	0.044343	0.107833	0.257261	0.014511	0.817634	0.298755	-0.
title_adj_to_len_ration	-0.013958	-0.019843	-0.152274	-0.049400	-0.019432	-0.024772	0.
title_noun_to_len_ration	-0.025251	-0.037511	-0.025477	-0.017093	-0.356624	-0.057076	0.
description_num_adj	0.066113	0.124427	0.137847	-0.001019	0.300760	0.898495	-0.
description_num_nouns	0.050675	0.120833	0.199486	0.004715	0.312669	0.970856	-0.
description_adj_to_len_ration	0.000882	-0.018710	-0.137510	-0.040210	-0.078614	-0.127597	0
description_noun_to_len_ration	0.006111	0.005324	0.061534	0.024627	0.029115	-0.047335	-0.
title_sentiment	-0.010653	0.017676	0.006652	-0.006089	0.070308	0.025541	-0.
description_sentiment	-0.011110	0.031508	0.006608	-0.020147	0.022263	0.104806	-0.

Feature eng. - Text, POS tagging

tagged_title	tagged_description	title_num_adj	title_num_nouns	title_adj_to_len_ration	title_noun_to_len_ration
[(Кокоби, S), ((, NONLEX), (кокон, S), (для, P...	[(Кокон, S), (для, PR), (сна, S), (малыша, S),...	0	3	0.0	1.000000
[(Стойка, S), (для, PR), (Одежды, S)]	[(Стойка, S), (для, PR), (одежды, S), (,, NONL...	0	2	0.0	0.666667
[(Philips, NONLEX), (bluray, NONLEX)]	[(В, PR), (хорошем, A=n), (состоянии, S), (,, ...	0	2	0.0	1.000000
[(Автокресло, S)]	[(Продам, V), (кресло, S), (от0-25кг, S)]	0	1	0.0	
			1	0.0	

The russian tagger rags sentences using the Russian National Corpus tagset:

http://www.ruscorpora.ru/en/corpora-morph.html

Here are some of the most important tags:

- S – noun
- A – adjective
- NUM – numeral
- A-NUM – numeral adjective
- V – verb
- ADV – adverb

description_adj_to_len_ration	description_noun_to_len_ration
0.142857	0.571429
0.000000	0.571429
0.117647	0.470588
0.000000	0.666667
0.000000	0.500000

param_1	param_2	param_3
Постельные принадлежности	NaN	NaN
Другое	NaN	NaN
Видео, DVD и Blu-ray плееры	NaN	NaN
Автомобильные кресла	NaN	NaN
С пробегом	ВАЗ (LADA)	2110
Автомобильные кресла	NaN	NaN
Сантехника и сауна	NaN	NaN

title_first_noun	title_first_noun_stemmed	title_second_noun	title_second_noun_stemmed	title_third_noun	title_third_noun_stemmed	de
Кокон	кокон	сна	сна	малыша	малыш	
Стойка	стойк	одежды	одежд	вешалки	вешалк	
состоянии	состоян	кинотеатр	кинотеатр	смарт	смарт	
кресло	кресл	от0-25кг	от0-25кг	None	None	
вопросы					None	

How many unique (distinct) "param"s we have?

```
print("Num of distinct params: "
      ,len(set(train["param_1"].tolist()
```

```
Num of distinct params:  1229
```

Is the product popular (how many buy this product)?

How many unique (distinct) nouns we have? How many distinct adjs we have?

In [301]:

```
print("Num of distinct nouns: "
      ,len(set(train["title_first_noun"].tolist()) | set(train["titl
print("Num of distinct stemmed nouns: "
      ,len(set(train["title_first_noun_stemmed"].tolist()) | set(tra
print("Num of distinct adjs: "
      ,len(set(train["title_first_adj"].tolist()) | set(train["title
print("Num of distinct stemmed adjs: "
      ,len(set(train["title_first_adj_stemmed"].tolist()) | set(trai
```

```
Num of distinct nouns:  55650
Num of distinct stemmed nouns:  34244
Num of distinct adjs:  36603
Num of distinct stemmed adjs:  17395
```

Feature eng. - Text, Sentiment analysis

title_sentiment	description_sentiment
150000.000000	138477.000000
0.013922	0.196082
0.189638	0.558819
-1.000000	-1.000000
0.000000	0.000000
0.000000	0.000000
0.000000	1.000000
1.000000	1.000000

title_sentiment	description_sentiment
0.0	0.0
0.0	0.0
0.0	0.0
0.0	0.0
0.0	-1.0

Feature eng. - Text, wrap-up and further steps

- Better study basic feature MI to target var:
  Length, word counts, words-to-sentence ratios, caps / non-letter char

- POS Tagging: Nouns, adjs, ratios.

  - What are popular “products”? “Product condition” is description. Look for Adjs/Nouns that increase prob. to buy. Look at their embeddings.

  - Clustering by embeddings. By params.

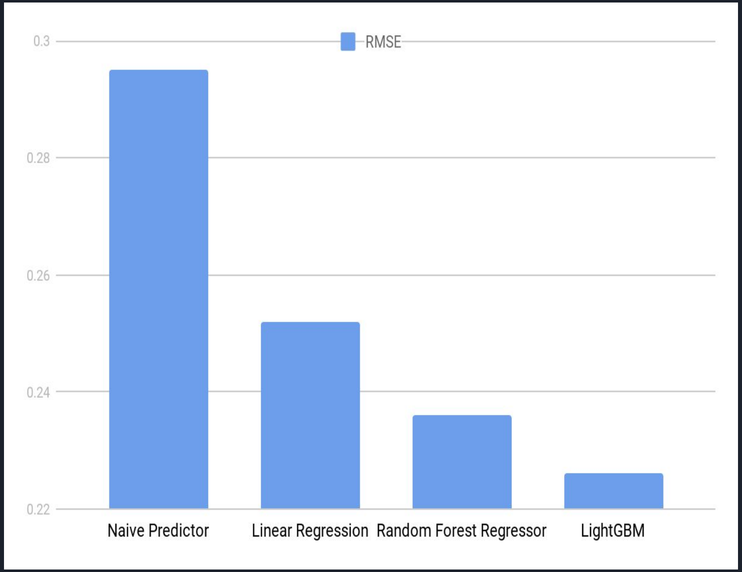
- Further steps:

  - Parsing of description to better identify find central NPs

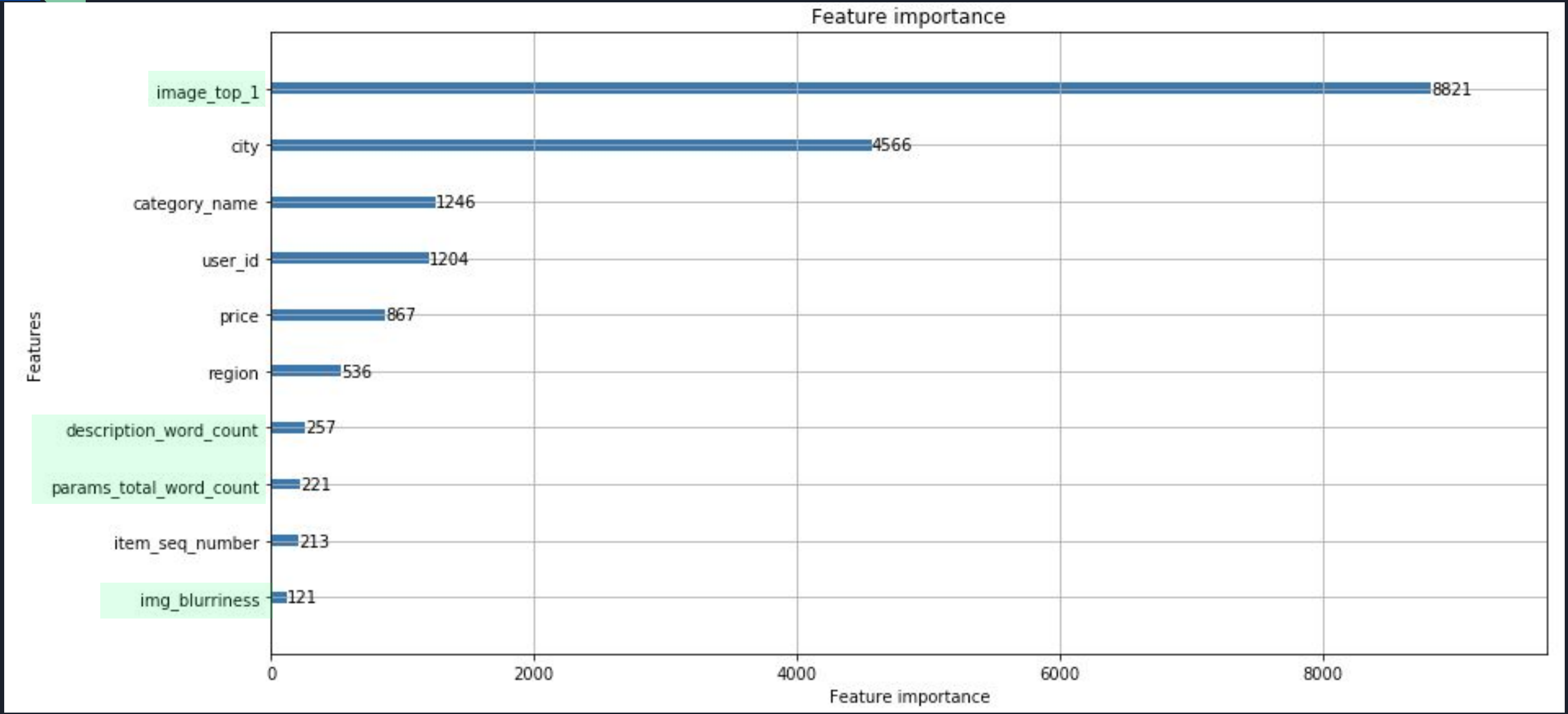
  - Description embedding (LSTM)

Models and evaluation - Initial results

- **Naive Predictor** - predicting 0 always
- **Random forests** - an ensemble learning method that operate by constructing multiple decision trees at training time and outputting the weighted prediction (regression) of the individual trees
- **LightGBM** - A fast, distributed, high performance gradient boosting (we use GBRT) framework based on decision tree algorithms
- Our result puts us in place ~320/~900 (almost ⅓!)



Models and evaluation - Feature Importance



Models and evaluation - Next Steps

- **Hyperparameter Tuning** - LightGBM has 100's of hyper parameters, and some method such as Grid Search for hyperparameter optimization should be applied to refine the performance of the algorithm.
- **Results Drilldown** - LightGBM offers many analysis methods such as displaying the resulting decision trees etc..
- Checking for **overfitting** - Something that easily happens with decision trees.
- **Neural Networks** - Attempting to build some NN architecture to achieve better results.