Конкурс Avito-2016: Распознавание категории объявления (3 этап)

Павел Блинов

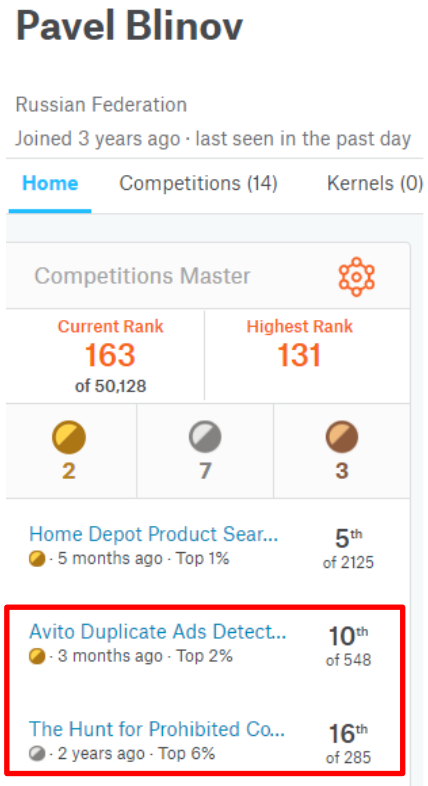
8 октября 2016 г.

План

- About myself
- Соревнование / постановка задачи
- Решение

About myself

www.kaggle.com/pavel1



www.machinelearning.ru/wiki/?title=Avito-2016-2

2е место из 9 на 2 этапе

2е место из 11 на 3 этапе

Соревнование

Этап 1 – 13.06-24.07

Изображения

Этап 2 – 01.08-21.08

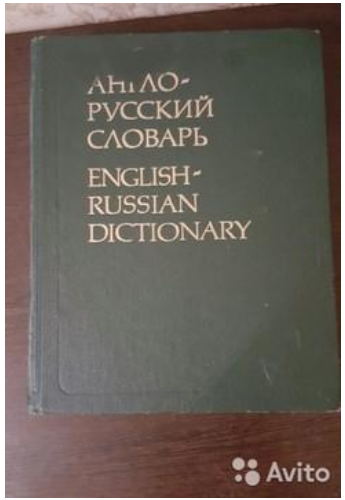
Изображения + заголовки

Этап 3 – 29.08-28.09

Изображения + заголовки + описания + цены

Задача классификации на 194 категории

Данные



Англо-русский словарь

53 000 слов. Автор В.К. Мюллер

150.0

	Train	Test
# объявлений	388 000	194 000
Объём текстовой информации (МБ)	186	92,3
# изображений	1 412 416	
Объём графической информации (ГБ)	~42,8	

Метрика качества

$$Q = \frac{1}{N}\sum_i q_i, \qquad q_i = 0.3 \cdot M_i^1 + 0.3 \cdot M_i^2 + 0.2 \cdot M_i^3 + 0.2 \cdot M_i^4$$

$$M_i^l = \begin{cases} 1, \textit{если категория уровня } l \textit{ угадана верно} \\ 0, \textit{иначе} \end{cases}$$

Actual class	Predicted class	
153	94	$q_i =$
Личные вещи /	Личные вещи /	0.3*1 +
Одежда, обувь /	Одежда, обувь /	0.3*1 +
Женская обувь /	Мужская обувь /	0.2*0 +
Кроссовки	Кроссовки	0.2*0 = 0.6

Особенности

Соревнование:

- Другая платформа (dataring.ru)
- Отсутствие форума
- Только 2 посылки в неделю
- Одно финальное решение
- Малое количество участников
- Многоэтапность

Задача:

- Данные различной природы
- Объём данных
- Сбалансированные данные
- Большое число категорий
- Нестандартная метрика

Вычислительные мощности

- Двухпроцессорный сервер
  (Intel Xeon CPU X5650 @ 2.67GHz × 24)
- 48 ГБ оперативной памяти
- Графическая карта Nvidia Tesla c2075
  (448 CUDA ядер)

Решение

```
# 2-layer model
model1.add(Dense(1839, input_shape=(1792,), init='glorot_normal', activation='relu'))
model1.add(Dropout(0.8))
model1.add(BatchNormalization())
model1.add(Dense(194, init='glorot_normal', activation='softmax'))
...
pred2l = model1.predict_proba(X_test_scaled)

# 3-layer model
model1.add(Dense(1839, input_shape=(1792,), init='glorot_normal', activation='relu'))
model1.add(Dropout(0.8))
model1.add(BatchNormalization())
model1.add(Dense(794, init='glorot_normal', activation='relu'))
model1.add(Dropout(0.5))
model1.add(BatchNormalization())
model1.add(Dense(194, init='glorot_normal', activation='softmax'))
...
pred3l = model1.predict_proba(X_test_scaled)

y_pred = np.argmax(pred2l+pred3l, axis=1)
```

Время на генерацию решения ~ 2 часа

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB

- Количество чисел в заголовке и описании
- Количество четырёхзначных чисел (aka упоминание года?)
- Количество токенов в объединении заголовка и описания
- Длина текста (заголовок + описание)
- Доля символов пунктуации
- Доля латинских символов
- Заголовок начинается с латинского символа?
- Log от цены
- Количество цифр 9 в цене
- Отношение длины заголовка к длине описания
- Текст начинается с цифры
- Текст заканчивается цифрой
- и т.д.

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB

Distributed Representations of Sentences and Documents
https://arxiv.org/pdf/1405.4053v2.pdf

- Заголовок + “ ” + описание
- Стемминг слов с помощью nltk.stem.snowball.RussianStemmer
- Gensim библиотека для построения векторов
- 200 компонент
- 5 эпох, 24 потока, время на обучение ~ 1 час

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB

- Заголовок + “ ” + описание + стемминг RussianStemmer
- groupby("target") -> 194 “тематических текста” -> TfidfVectorizer
- Найдём 400 ключевых слов для каждого “текста” и их веса

Category 38

```
0.607 книг
0.229 переплет
0.208 издательств
0.204 издан
0.143 тверд переплет
0.141 состоян
0.121 автор
0.115 собран сочинен
0.11 литератур
0.108 сочинен
```

Category 46

```
0.574 lg
0.199 телефон
0.181 nexus
0.148 состоян
0.145 телефон lg
0.132 гб
0.118 экра
0.11 android
0.106 lte
0.103 камер
```

- Для каждой категории значение признака: set(ad_text).**intersection**(keywords[categ]) + **sum**(weights) + **bonus** (1 if keywords[categ][0] == ad_text[0] else 0)

- Например, “Собрание сочинений Толстого Л.Н. 14 томов” & category38 (1+1+1) + (0,115+0,095+0,108) + 0 = 3,318

- Итого 194 признака, 24 потока, время получения ~ 3,5 часа

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB

- Заголовок + “ ” + описание + lower
- Выделим символьные триграммы
  - Кондиционер HITACHI RAS-14CH6 -> кон онд нди диц ици цио ион оне нер ер р h hi hit ita tac ach chi hi i r ra ras as- s-1 -14 14c 4ch ch6
- HashingVectorizer на полученных “текстах”
- TruncatedSVD(n_components=400)
- explained_variance_ratio_.sum() == 0.658
- Время получения ~ 15 минут

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB

CV .660
Public .787

- Imagenet v3 https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/image_retraining
- ResNet 50 https://github.com/ry/tensorflow-resnet

- Inception features -> двухслойная и трёхслойная нейросеть
- ResNet features -> двухслойная и трёхслойная нейросеть
- np.hstack(Inception and ResNet features) -> двухслойная нейросеть

```
np.power(inc2, 0.69) + np.power(inc3, .15) + np.power(res2, .97) + np.power(res3, .46) + np.power(inc_res2, .05)+\
np.power(inc2, 0.82) + np.power(inc3, .05) + np.power(res2, .43) + np.power(res3, .83) + np.power(inc_res2, .11)
```

- Hyperopt для подбора показателей степеней
- Итого 194 признака, 24 потока, время получения ~ 120 часов

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB
			CV	.660	.690			
			Public	.787				

- Using pre-trained word embeddings in a Keras model https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html
- Word2vec модель обученная на НКРЯ http://ling.go.mail.ru/misc/dialogue_2015.html#rnc
- Title words only + простая предобработка `title=re.sub(u"[^a-za-я0-9]"," ", title.lower())`
- Итого 194 признака, время получения ~ 2,5 часа

28	200	194	400	194	194	194	194	194
Handmade-признаки	doc2vec признаки	Признаки на основе ключевых слов	3-граммы+ SVD	Image-признаки	CNN for text признаки	tf.idf + LogisticRegression	tf.idf + SGDClassifier	tf.idf + MultinomialNB
			CV	.660	.69	.77	.77	.77
			Public	.787				

- Заголовок + “ ” + описание
- Нормализация словоформ с помощью mystem

```
TfidfVectorizer( min_df=3, use_idf=True, ngram_range=(1,2) )
```

```
[("lr", linear_model.LogisticRegression(C=5.0, random_state=SEED, n_jobs=-1)),
 ("mnb", MultinomialNB(alpha=0.2)),
 ("sgd", linear_model.SGDClassifier(loss='modified_huber', n_iter=11, alpha=0.00001,
                                    l1_ratio=0.15, random_state=SEED, n_jobs=-1, epsilon=0.2))]
```

- Итого 582 признака, время получения ~ 2,5 часа

	CV	Public LB	Примечание
1	.5173	.6798	
2	.6312	.7647	
3	.6609	.7867*	
4	.7396	.8472	
5	~.7925	.8823	
6	~.7945	.8839	
7	~.7977	.8862	
8	~.800?	.8895	
9	~.803?	.???? / .8899*	
10	~.8184	.9011	Та же самая модель что и на втором этапе, только текст дополнен описаниями
11	~.8197	.????	Добавление log(price) как признака
12	~.8230	.9051	Мета признаки tf.idf + LR
13	~.8254	.9082	Мета признаки tf.idf + MNB
14	~.8266	.????	Little fine tuning of NN: add neurons, increase dropout
15	~.8275	.9088	Мета признаки tf.idf + SGD (+features: count of 9s and count of 0s in price)
16	~.9033	.8747	Переобучение by LDA model (BigARTM) (order?!)
17	~.8278	.9092 / .9090*	Добавление BatchNormalization

~ – оценка производилась только по первой части CV

* – использован для отправки

Lessons learned

- Обработка изображений with deep learning technique вполне выполнима
- CNN for text classification
- Stacking крут!
- Стоит тщательнее проверять код на ошибки

Код + описание
bitbucket.org/pavel-blinov/avito2016

Спасибо за внимание!

Вопросы???

Павел Блинов
blinoff.pavel@gmail.com