

The ECML/PKDD Discovery Challenge 2016 on Bank Card Usage Analysis – Task 1

Андрей Зимовнов
Yandex Data Factory

Задача и метрика качества

Задача: Предсказать посещения банковских отделений (их 323) для каждого клиента (<https://dms.sztaki.hu/ecml-pkcd-2016>)

Метрика качества: $\frac{1}{2} (\text{cosine@1} + \text{cosine@5})$ усредненное по всем клиентам.

$$\text{cosine@}k = \frac{\sum_{i=1}^k v(p_i) \hat{v}(p_i)}{\sqrt{\sum_j v(j)^2} \sqrt{\sum_{i=1}^k \hat{v}(p_i)^2}}$$

$v(j)$ - истинное количество посещений отделения j

$\hat{v}(j)$ - предсказанное количество посещений отделения j

$p_i, 1 \leq i \leq 5, \hat{v}(p_1) > \dots > \hat{v}(p_5)$ – индексы пяти самых популярных предсказанных отделений

Как устроен сплит данных

Данные о транзакциях, пользователях, посещениях отделений



Данные о клиентах

- **USER_ID** Unique user id
- **AGE_CAT** Age category in 2014. a - -35, b - 36-65, c - 65+
- **LOC_CAT** Location category of the user. a = capital, b = city, c = village
- **INC_CAT** Income category. possible values are a = low, b = medium, c = high, d = no income
- **GEN** Gender. 1 = male, 0 = female
- **LOC_GEO_X** Geo info of user address is rounded to 100m
- **LOC_GEO_Y** Geo info of user address is rounded to 100m
- **TARGET_TASK_2** The date when the user applied for credit card. Only given for training users.
- **C201***- Binary columns for each month. If True, the user has at least one credit card.
- **W201***- Binary columns for each month. If True, the user is categorized as "wealthy" in the system of the bank.

Данные о клиентах - пример

	USER_ID	AGE_CAT	LOC_CAT	INC_CAT	GEN	LOC_GEO_X	LOC_GEO_Y	TARGET_TASK_2	C201401	C201402	...
0	30277	a	c	b	0	857400	334900	2014.04.30	0	0	...
1	99045	b	b	b	0	699400	173300	NaN	0	0	...
2	19239	a	b	d	0	695900	170700	NaN	0	0	...
3	24396	a	b	a	0	585900	78300	NaN	0	0	...
4	111628	b	b	a	0	586000	78200	2015.06.30	1	1	...

Данные о транзакциях

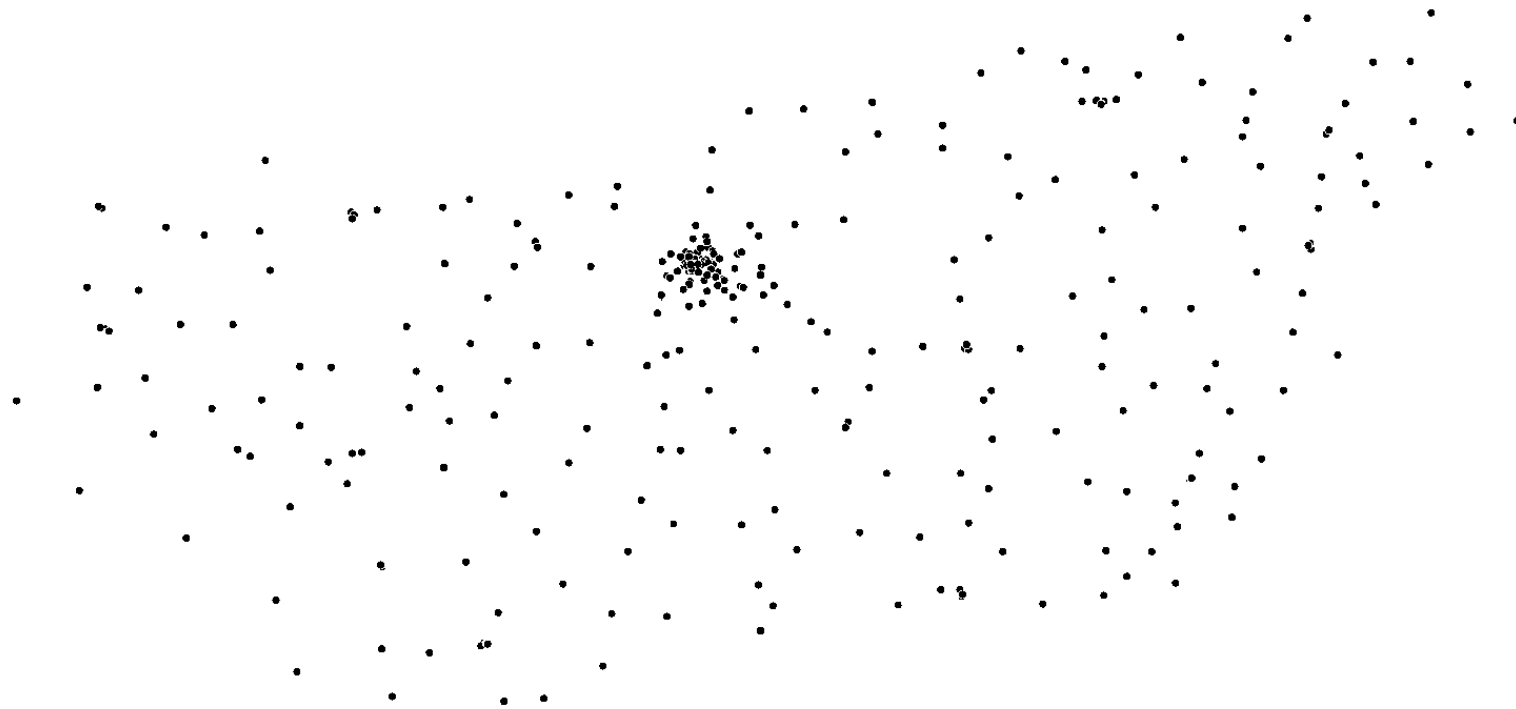
- **USER_ID** Unique user ID
- **POI_ID** Unique shop ID
- **CHANNEL** Type of activity. p = pos, n = webshop, b = branch
- **DATE** Date of activity
- **TIME_CAT** Time rounded to a = 05-11h, b = 12-18h, c = 19 -04h
- **LOC_CAT** Event location category. a = capital, b = city, c = village
- **MC_CAT** Anonymized market category groups. Types are indexed from a ... j
- **CARD_CAT** Credit vs. debit card. c = credit card, d = debit card
- **AMT_CAT** Amount of money spent in three categories. a=low, b=medium, c=high
- **GEO_X** Geolocation information of the event
- **GEO_Y** Geolocation information of the event

Данные о транзакциях - пример

	USER_ID	POI_ID	CHANNEL	DATE	TIME_CAT	LOC_CAT	MC_CAT	CARD_CAT	AMT_CAT	GEO_X	GEO_Y
0	91498	28052	p	2014-01-01	a	b	b	d	b	605119.0	58997.8
1	266177	1759	n	2014-01-01	a	NaN	j	d	b	NaN	NaN
2	202438	897	p	2014-01-01	a	b	b	d	c	698820.0	174757.0
3	109668	19939	p	2014-01-01	a	b	b	d	b	716050.0	271521.0
4	218581	13992	p	2014-01-01	a	a	j	d	c	653242.0	239511.0

Гео-координаты

Венгрия со столицей Будапешт



Клиентские признаки

- Euclidean distances from the user geo location to all of the 323 bank POIs' geo locations.
- C201* features for the first 6 months (binary columns for each month, if True, the user has at least one credit card).
- W201* features for the first 6 months (binary columns for each month, if True, the user is categorized as "wealthy" in the system of the bank).
- Gender, age, income, location categories one-hot encoded vectors.

Транзакционные признаки

- Each transaction is characterized by time category, location category, merchant category, card category and amount category. For any set of above features we can define a vector of counters for all possible values combinations. For instance, let's take into consideration card category and amount category, which can take (c, d, \emptyset) and (a, b, c, \emptyset) values respectively. For them we define counters for any possible values combination $(c, a), (c, b), \dots, (\emptyset, c), (\emptyset, \emptyset)$. For each combination corresponding counter is incremented with every transaction having such values combination. We'll refer to the resulting vector of counters as **C(card, amount)** for the example above.

Транзакционные признаки

- Each transaction also has a geo location, we propose to use it to count how many transactions took place next to the nearest bank POI, i.e. we increment bank POI counter each time it is the nearest to a transaction geo location. This way we acquire a vector **L** of counters of length 323.

Транзакционные признаки

- The resulting vector is then defined as a concatenation of the following vectors for the first 6 months:
 1. C(time)
 2. C(location)
 3. C(merchant)
 4. C(card)
 5. C(amount)
 6. C(merchant, amount)
 7. C(location, merchant)
 8. C(time, location, amount)
 9. C(card, amount)
 10. C(merchant, card, amount)
 11. L

Алгоритм

For the task we trained 323 classifiers on described features that predicted if the user will visit certain bank POI.

Every classifier is gradient boosted decision trees implemented in **xgboost**.

We tuned the number of trees using cross-validation on the training set, the optimal value was around 75 ± 25 , but most of them had 75, so we took 75 trees for each bank POI to avoid overfitting.

The predicted vector contains the probabilities of predictions for every bank POI.

It takes 2 hours on 32 core machine to train all classifiers.

Схема валидации

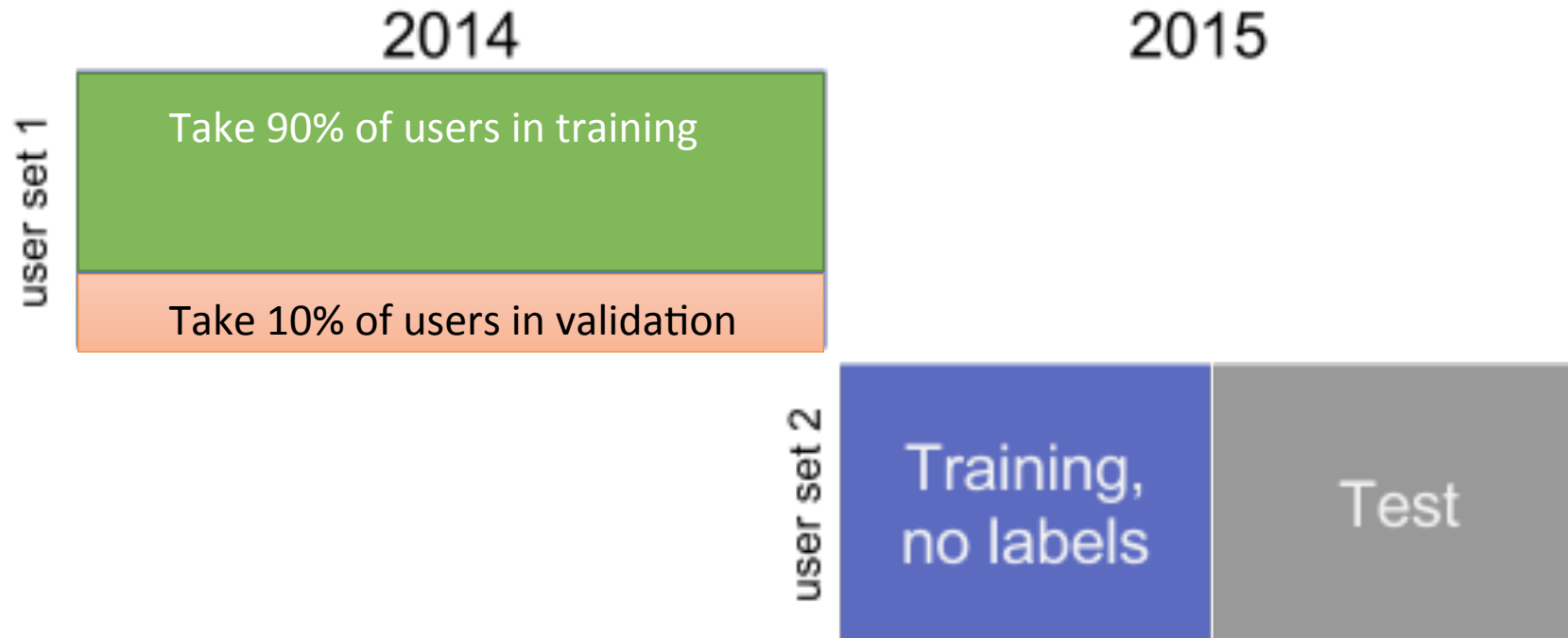


Схема валидации — корреляция с leaderboard

- 0.679382259127 -> LB 0.6818
- 0.680437368829 -> LB 0.6843
- 0.682123092701 -> LB 0.6849
- 0.683509285526 -> LB 0.6851

Результаты Task 1 (26 участников)

Task 1: Predict the bank branches visited by the user.



1st, **0.68659**
Team: ISMLL



2nd, **0.68512**
Team: Ya



3rd, **0.67436**
Team: Cosine Vinny

Результаты Task 2 (39 участников)

Task 2: Upselling prediction



1st, **0.71862**
Team: achm



5th, **0.71479**
Team: TwoBM



2nd, **0.71730**
Team: Cosine Vinny



6th, **0.71386**
Team: MMNF



3rd, **0.71589**
Team: Degrees of Freedom



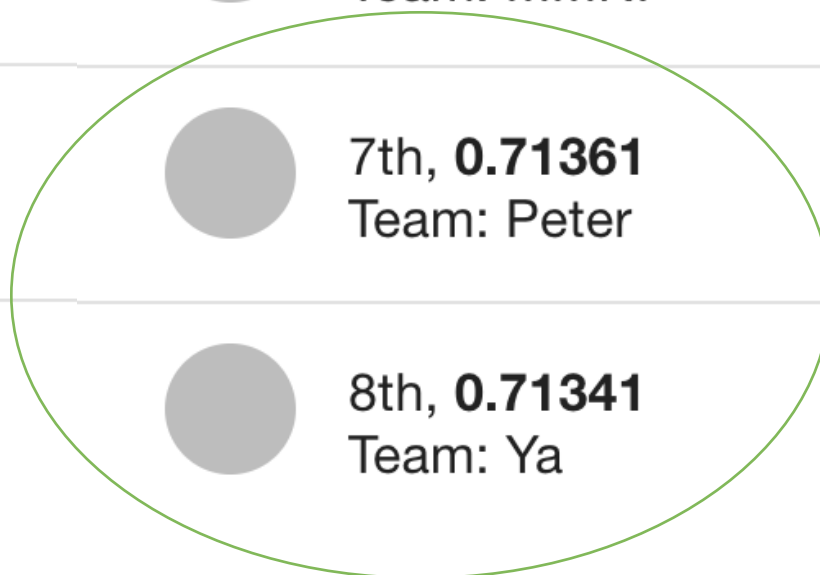
7th, **0.71361**
Team: Peter



4th, **0.71523**
Team: ISMLL



8th, **0.71341**
Team: Ya



Task 2 — Upselling Prediction

Ромов Петр

Yandex Data Factory

Что является целью предсказания?

Таргет: факт оформления кредитной карты пользователем в тестовом периоде

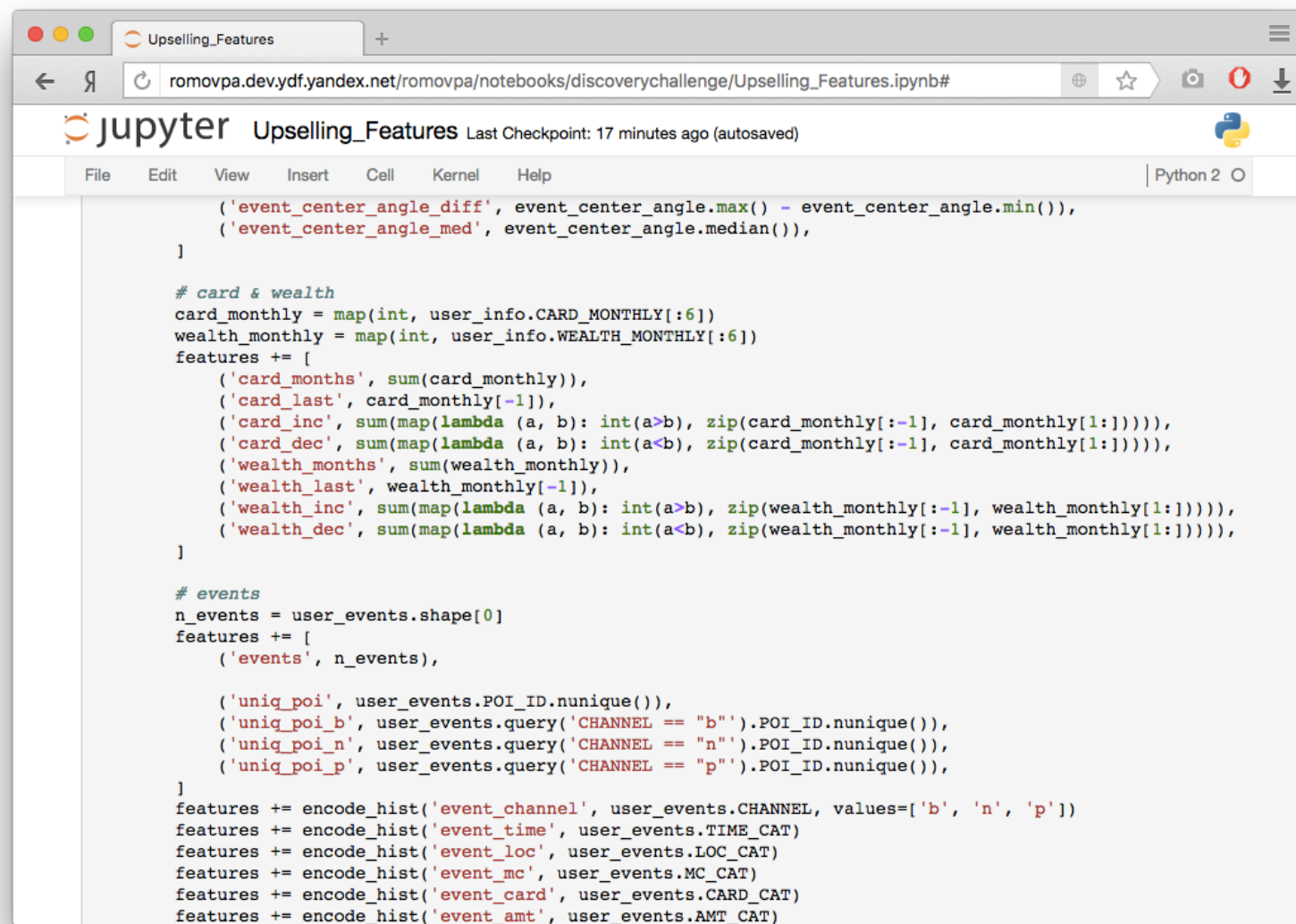
Метрика: ROC-AUC



Вечер #1: Загрузка данных / фичи

- Из анкеты клиента: пол, возрастная категория, тип локации
- Координаты клиента (место проживания)
- Наличие кредитной карты или депозита в первом полугодии
- Из транзакций:
 - Число транзакций: общее, по дням недели
 - Число отделений банка, которыми пользовался клиент
 - Координаты: median, mean, std, quantiles
 - Расстояния точек транзакций до Будапешна: median, mean, std, quantiles
 - Углы точек транзакций относительно Будапешта
 - Число дней / недель / месяцев, когда клиент был активен

Вечер #1: Загрузка данных / фичи



```
    ('event_center_angle_diff', event_center_angle.max() - event_center_angle.min()),
    ('event_center_angle_med', event_center_angle.median()),
]

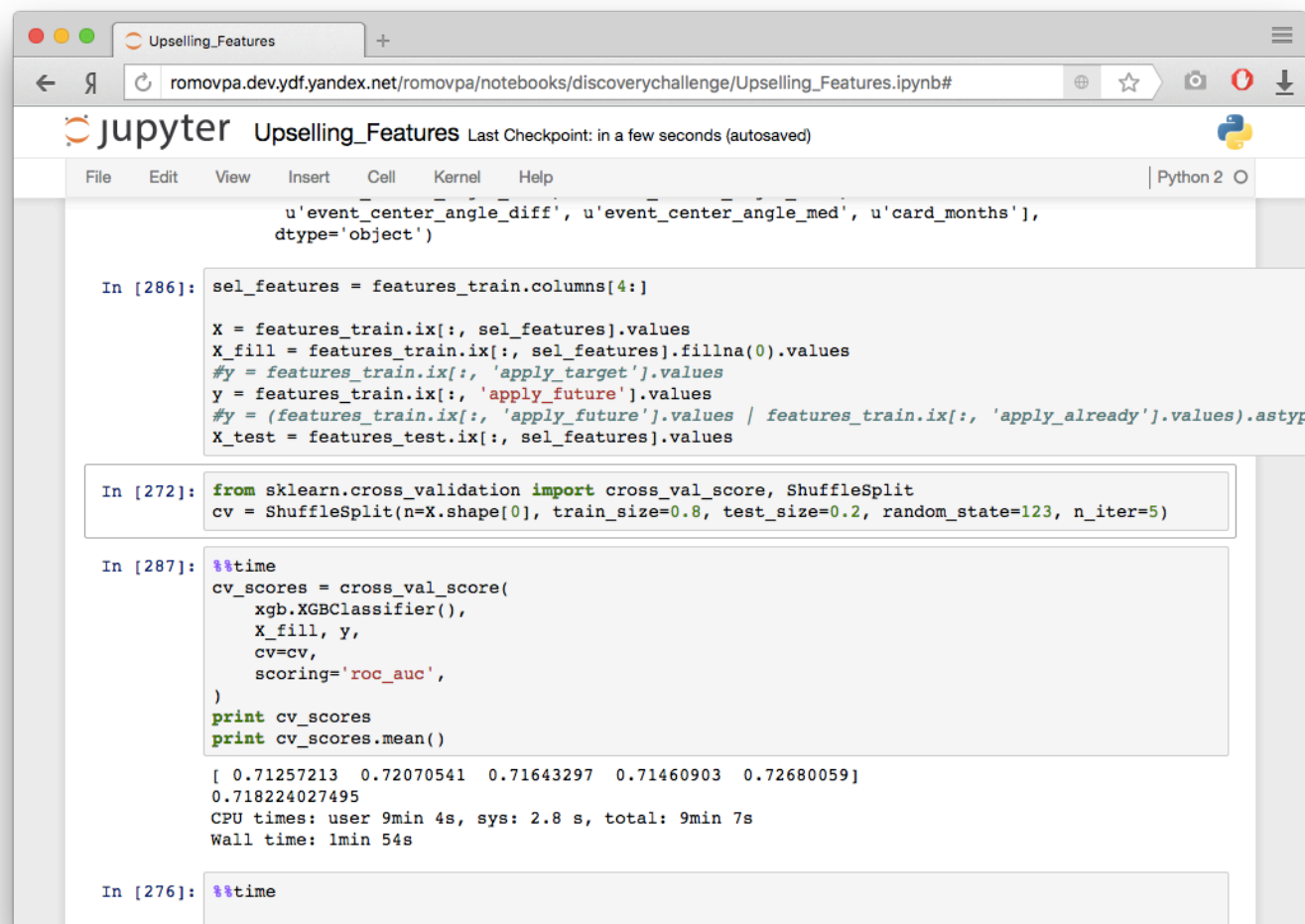
# card & wealth
card_monthly = map(int, user_info.CARD_MONTHLY[:6])
wealth_monthly = map(int, user_info.WEALTH_MONTHLY[:6])
features += [
    ('card_months', sum(card_monthly)),
    ('card_last', card_monthly[-1]),
    ('card_inc', sum(map(lambda (a, b): int(a>b), zip(card_monthly[:-1], card_monthly[1:])))),
    ('card_dec', sum(map(lambda (a, b): int(a<b), zip(card_monthly[:-1], card_monthly[1:])))),
    ('wealth_months', sum(wealth_monthly)),
    ('wealth_last', wealth_monthly[-1]),
    ('wealth_inc', sum(map(lambda (a, b): int(a>b), zip(wealth_monthly[:-1], wealth_monthly[1:])))),
    ('wealth_dec', sum(map(lambda (a, b): int(a<b), zip(wealth_monthly[:-1], wealth_monthly[1:])))),
]

# events
n_events = user_events.shape[0]
features += [
    ('events', n_events),

    ('uniq_poi', user_events.POI_ID.nunique()),
    ('uniq_poi_b', user_events.query('CHANNEL == "b"').POI_ID.nunique()),
    ('uniq_poi_n', user_events.query('CHANNEL == "n"').POI_ID.nunique()),
    ('uniq_poi_p', user_events.query('CHANNEL == "p"').POI_ID.nunique()),
]

features += encode_hist('event_channel', user_events.CHANNEL, values=['b', 'n', 'p'])
features += encode_hist('event_time', user_events.TIME_CAT)
features += encode_hist('event_loc', user_events.LOC_CAT)
features += encode_hist('event_mc', user_events.MC_CAT)
features += encode_hist('event_card', user_events.CARD_CAT)
features += encode_hist('event_amt', user_events.AMT_CAT)
```

Вечер #2: XGBoost и первый сабмишн



The screenshot shows a Jupyter Notebook window titled "Upselling_Features". The browser address bar shows the URL: `romovpa.dev.ydf.yandex.net/romovpa/notebooks/discoverychallenge/Upselling_Features.ipynb#`. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar (Python 2, O). The code is organized into three input cells:

```
u'event_center_angle_diff', u'event_center_angle_med', u'card_months'],
dtype='object')

In [286]: sel_features = features_train.columns[4:]

X = features_train.ix[:, sel_features].values
X_fill = features_train.ix[:, sel_features].fillna(0).values
#y = features_train.ix[:, 'apply_target'].values
y = features_train.ix[:, 'apply_future'].values
#y = (features_train.ix[:, 'apply_future'].values | features_train.ix[:, 'apply_already'].values).astype
X_test = features_test.ix[:, sel_features].values

In [272]: from sklearn.cross_validation import cross_val_score, ShuffleSplit
cv = ShuffleSplit(n=X.shape[0], train_size=0.8, test_size=0.2, random_state=123, n_iter=5)

In [287]: %%time
cv_scores = cross_val_score(
    xgb.XGBClassifier(),
    X_fill, y,
    cv=cv,
    scoring='roc_auc',
)
print cv_scores
print cv_scores.mean()

[ 0.71257213  0.72070541  0.71643297  0.71460903  0.72680059]
0.718224027495
CPU times: user 9min 4s, sys: 2.8 s, total: 9min 7s
Wall time: 1min 54s

In [276]: %%time
```

Результаты Task 2 (39 участников)

Task 2: Upselling prediction



1st, **0.71862**
Team: achm



5th, **0.71479**
Team: TwoBM



2nd, **0.71730**
Team: Cosine Vinny



6th, **0.71386**
Team: MMNF



3rd, **0.71589**
Team: Degrees of Freedom



7th, **0.71361**
Team: Peter



4th, **0.71523**
Team: ISMLL



8th, **0.71341**
Team: Ya

