Sberbank Data Science Contest

#	Участник	Общий балл	Задача A	Задача B	Задача C
77	mrk-andreev (DMIA)	358.8919	0.896297 (169.4049)	1.575759 (189.4871)	1.647985 (0.0000)
(11%)			14 (2%)	59 (8%)	194 (27%)

Задача A

Дано:

- Выборка пользователей сбербанка
- История транзакций всех пользователей

  [record = {date, mcc_code, tr_type, amount}]
- Пол (М/Ж) для части выборки

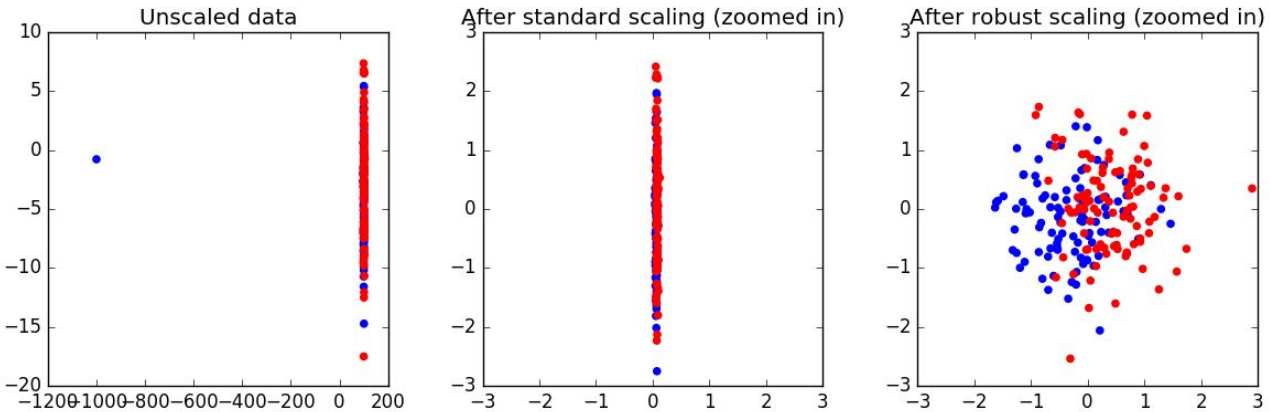
Найти:

- Пол для неразмеченной части выборки
- Метрика AUC-ROC

$$\text{AUC-ROC} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} I[y_i < y_j] I[\tilde{y}_i < \tilde{y}_j]}{\left(\sum_{i=1}^{n} I[y_i = 0]\right)\left(\sum_{i=1}^{n} I[y_i = 1]\right)} \in [0, 1].$$

Что было использовано

- (?) RobustScaler



https://goo.gl/V0SocG

Статистики для групп

- (+) Статистики для групп mcc_code, tr_type по каждому пользователю, следующего вида:

```
F2 = transactions
   .groupby('customer_id').apply(lambda x :
                           x.groupby('mcc_code')['amount'].sum())
         .unstack().fillna(0)
F2.rename(columns=lambda x: 'sum_'+str(x), inplace=True)
```

- Разделим выборку на [amount>0] и [amount<0]
- Вычислим статистики для [∀ amount], [amount>0], [amount<0]

- Вычислим сколько дней пользователь наш клиент: max(day) - min(day)

XGBoost

- Мы получили ~4000 признаков
- Отправим их в XGBoost

```
def apply_model(tr, te, target):
    clf = xgboost.XGBClassifier(seed=0, learning_rate=0.02, max_depth=5,
subsample=0.6815, colsample_bytree=0.701, n_estimators=1000, nthread=4)
    clf.fit(tr, target)
    return clf.predict_proba(te)[:, 1]
```

(++) Смешиваем модели!

Идем на форум и видим:

Скрипт задачи A, скор 0.867786

Автор: Const

Просмотров: 1333

Ответов: 25

Последний комментарий: bernadsky 3 дня назад

Получаем submit товарища @Const и смешиваем его с нашим решением:

```
from scipy.stats import rankdata
r_ans = pd.read_csv('../data/raw/task1_solution_by_const.csv' )
blended_submit = submit_data.copy()
blended_submit['gender'] = rankdata(r_ans['gender'].values) +
rankdata(submit_data['gender'].values)
blended_submit.to_csv('../data/submits/sbm_' + ts + '_blended.csv',
index=False)
```

Задача B

- Используем XGBoost с настроенными параметрами…

```
k = 500
param = {
  'eta' : 0.2/float(k),
  'max_depth' : 5,
  'colsample_bytree' : 0.2,
  'min_child_weight' : 13,
  'gamma' : 14,
  'subsample' : 0.7,
  'objective' : 'reg:linear',
  'eval_metric' : "rmse"
}

clf = xgboost.train(param, dtrain, num_boost_round=100*k)
```

t ~ 2 часа

Спасибо за внимание

https://github.com/mrk-andreev/contest-sdsj/