

# Описание формата сравнения.

Апрель, 2017

## Входные данные.

Все эксперименты проводились на следующих датасетах.

Для задач классификации:

Adult, amazon, appet, click, criteo, internet, kdd98, kddchurn, kick, paribas, springleaf, upsel.

Для задач регрессии:

Allstate, bimbo, liberty.

Все датасеты разбивались на обучающую и тестовую части в соотношении 4:1 соответственно. Обозначим их за  $(X_{full\_train}, y_{full\_train})$  и  $(X_{test}, y_{test})$ .

## Предобработка датасета.

Итак, на входе имеется обучающая  $(X_{full\_train}, y_{full\_train})$  и тестовая  $(X_{test}, y_{test})$  выборки, а также список номеров колонок категориальных признаков.

В экспериментах используется 5-фолдовая кросс-валидация. Поэтому  $(X_{full\_train}, y_{full\_train})$  разбивается на 5 подвыборок  $(X_1, y_1), \dots, (X_5, y_5)$ , и из них конструируется 5 выборок вида  $(X_i^{train}, y_i^{train})$ ,  $(X_i^{val}, y_i^{val})$  таким образом, что  $(X_i^{val}, y_i^{val})$  совпадает с  $(X_i, y_i)$ , а  $(X_i^{train}, y_i^{train})$  совпадает с  $\cup_{j \neq i} (X_j, y_j)$ .

Далее, для каждой такой пары, мы предобрабатываем категориальные признаки по следующей схеме.

Пусть имеется обучающая  $(X^{train}, y^{train})$  и валидационная  $(X^{val}, y^{val})$  выборки. Для простоты обозначений будем считать, что все признаки категориальные. Вводим понятие времени в обучающей выборке. На выборках, в которых присутствует признак "время" — упорядочиваем все по нему, если же такого признака нет, то производим случайную перестановку объектов. Считаем, что для задач классификации метки классов принадлежат множеству  $\{0, 1\}$ . Далее, для каждого  $j$ -го признака и  $i$ -го объекта, считаются 2 числа  $a_{ij}$  и  $b_{ij}$ :

$$a_{ij} = \sum_{k=1}^{i-1} [X_{ij}^{train} = X_{kj}^{train}] y_{kj}^{train},$$
$$b_{ij} = \sum_{k=1}^{i-1} [X_{ij}^{train} = X_{kj}^{train}], \text{ где } [\dots] \text{ — индикатор.}$$

Теперь в обучающей выборке категориальные признаки заменяются на числовые по следующей формуле.

**Для задач классификации:**

$$X_{ij}^{train} = \frac{a_{ij} + 1}{b_{ij} + 2}.$$

**Для задач регрессии:**

$$X_{ij}^{train} = \begin{cases} \frac{a_{ij}}{b_{ij}}, & \text{if } b_{ij} \neq 0 \\ 0, & \text{if } b_{ij} = 0 \end{cases}.$$

Далее необходимо заменить категориальные признаки в валидационной выборке. Для этого, для каждого  $j$ -го признака и  $i$ -го объекта так же считаются 2 числа  $c_{ij}$  и  $d_{ij}$ :

$$c_{ij} = \sum_k [X_{ij}^{val} = X_{kj}^{train}] y_{kj}^{train},$$
$$d_{ij} = \sum_k [X_{ij}^{val} = X_{kj}^{train}], \text{ где } [\dots] \text{ — индикатор.}$$

Теперь в валидационной выборке категориальные признаки заменяются на числовые по следующей формуле.

**Для задач классификации:**

$$X_{ij}^{val} = \frac{c_{ij} + 1}{d_{ij} + 2}.$$

**Для задач регрессии:**

$$X_{ij}^{val} = \begin{cases} \frac{c_{ij}}{d_{ij}}, & \text{if } d_{ij} \neq 0 \\ 0, & \text{if } d_{ij} = 0 \end{cases}.$$

Таким образом получилось 5 пар (обучающая и валидационная) выборок, которые содержат только числовые значения.

Далее, для исходных выборок  $(X_{full\_train}, y_{full\_train})$  и  $(X_{test}, y_{test})$ , также заменим категориальные признаки на числовые по той же самой схеме, что и для  $(X^{train}, y^{train})$  и  $(X^{val}, y^{val})$ .

### Сетка параметров.

Параметры подбираются с помощью библиотеки hyperopt. Ниже приведен список подбираемых параметров и распределений, откуда они выбирались для каждого алгоритма:

XGBoost.

- 'eta': Логравномерное распределение  $[e^{-7}, 1]$
- 'max\_depth': Дискретное равномерное распределение  $[2, 10]$
- 'subsample': Равномерное  $[0.5, 1]$
- 'colsample\_bytree': Равномерное  $[0.5, 1]$
- 'colsample\_bylevel': Равномерное  $[0.5, 1]$
- 'min\_child\_weight': Логравномерное распределение  $[e^{-16}, e^5]$
- 'alpha': Смесь:  $0.5 * \text{Вырожденное в } 0 + 0.5 * \text{Логравномерное распределение } [e^{-16}, e^2]$
- 'lambda': Смесь:  $0.5 * \text{Вырожденное в } 0 + 0.5 * \text{Логравномерное распределение } [e^{-16}, e^2]$

LightGBM.

- 'learning\_rate': Логравномерное распределение  $[e^{-7}, 1]$
- 'num\_leaves': Дискретное логравномерное распределение  $[1, e^7]$
- 'feature\_fraction': Равномерное  $[0.5, 1]$
- 'bagging\_fraction': Равномерное  $[0.5, 1]$
- 'min\_sum\_hessian\_in\_leaf': Логравномерное распределение  $[e^{-16}, e^5]$
- 'min\_data\_in\_leaf': Дискретное логравномерное распределение  $[1, e^6]$
- 'lambda\_l1': Смесь:  $0.5 * \text{Вырожденное в } 0 + 0.5 * \text{Логравномерное распределение } [e^{-16}, e^2]$
- 'lambda\_l2': Смесь:  $0.5 * \text{Вырожденное в } 0 + 0.5 * \text{Логравномерное распределение } [e^{-16}, e^2]$
- 'max\_bin': Дискретное логравномерное распределение  $[1, e^{20}]$

### Подбор параметров.

При подборе, в каждом алгоритме выставляется параметр, отвечающий за максимальное число деревьев, равный 2000. Далее, для каждого конкретного набора параметров, в каждом из 5 фолдов, при добавлении очередного дерева в алгоритм, подсчитываются значения метрик на валидационной выборке. В итоге получается 5 2000-мерных векторов, которые усредняются в один вектор, по которому берется аргмаксимум. Полученное число и является оптимальным количеством деревьев для данного набора параметров.

В итоге, производилось 50 итераций подбора параметров и выбирались те параметры, на которых получалась наилучшая метрика LogLoss.

### Итоговый результат.

В итоге, в алгоритме выставляются оптимальные параметры, и запускается обучение на предобработанном  $(X_{full\_train}, y_{full\_train})$ . После этого вычисляется значение метрики LogLoss на предобработанной тестовой выборке  $(X_{test}, y_{test})$ .

### Версии библиотек.

- xgboost (0.6)
- scikit-learn (0.18.1)
- scipy (0.19.0)
- pandas (0.19.2)
- numpy (1.12.1)
- lightgbm (0.1)
- hyperopt (0.0.2)