Машинное обучение и анализ данных

1 Обучение на размеченных данных

1.1 Постановка задачи

Для обучающей выборки $X = (x_i, y_i)_{i=1}^{l}$ найти такой алгоритм $a \in \mathbb{A}$, на котором будет достигаться минимум функционала ошибки:

$$Q(a, X) \to \min_{a \in \mathbb{A}}.$$

Разброс – дисперсия ответов алгоритма $D(a(x))$.

Смещение – математическое ожидание $E(y - a(x))$.

1.2 Процедура обучения

Анализ данных и визуализация:

- разбиение признаков на вещественные, дискретные, категориальные, бинарные
- анализ пропущенных значений и масштабов признаков
- вещественные и дискретные признаки: гистограммы (DataFrame.hist) и попарные scatterplot (seaborn.pairplot, plt.scatter, Axes.scatter)
- вещественные признаки: попарные корреляции (DataFrame.corr, seaborn.heatmap)
- категориальные признаки: seaborn.countplot, DataFrame.crosstab
- категориальные vs. вещественные: seaborn.boxplot
- баланс классов – countplot целевого признака
- понижение размерности: t-SNE (масштабированные признаки), MDS

- аномалии: OneClassSVM
- восстановление плотностей распределений: seaborn.distplot

Предобработка данных:

- выделение holdout выборки (со стратификацией)
- обработка пропущенных значений (в т.ч. проверить, случайно ли расположение пропусков в матрице, также см. *link*)
- обработка аномалий
- вещественные и дискретные признаки: преобразование, масштабирование
- бинарные признаки: LabelEncoder
- категориальные признаки: dummy-кодирование

Baseline на holdout выборке:

- линейная модель
- дерево решений
- ансамбль: бэггинг, случайный лес, градиентный бустинг
- нейронная сеть
- наивный байес
- kNN
- метод опорных векторов (SVM)

Подбор параметров на кросс-валидации:

- стратификация
- валидация без выбросов: при вычислении метрики, чувствительной к выбросам, не учитываем выбросы (см. *link*)
- учет статистической значимости при сравнении скоров
- подбор параметров по одному (с анализом графиков)
- усреднение моделей с близкими значениями скора
- отложенная выборка – для оценки качества на нетронутых данных

1.3 Метрики качества

Метрики качества регрессии:

- $MSE(a, X) = \frac{1}{l} \sum\limits_{i=1}^{l} (a(x_i) - y_i)^2$ (настраивается на выбросы)
- $MAE(a, X) = \frac{1}{l} \sum\limits_{i=1}^{l} |a(x_i) - y_i|$ (более устойчивая к выбросам)
- $R^2(a, X) = 1 - \frac{\sum_{i=1}^{l}(a(x_i)-y_i)^2}{\sum_{i=1}^{l}(y_i-\bar{y})^2}$

Метрики качества классификации:

- $\text{accuracy}(a, X) = \frac{1}{l} \sum\limits_{i=1}^{l} [a(x_i) = y_i]$ (для разумных алгоритмов должна быть больше доли объектов крупного класса)
- $\text{precision}(a, X) = \frac{TP}{TP+FP}$ (насколько доверять классификатору в случае срабатывания)
- $\text{recall}(a, X) = \frac{TP}{TP+FN}$ (на какой доле объектов первого класса алгоритм срабатывает)
- $F = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$ (при $\beta < 1$ важнее recall)

Метрики качества оценок принадлежности классу:

- AUC PRC – площадь под PR-кривой (изменяется при изменении баланса классов)
- AUC ROC – площадь под ROC-кривой (TPR – ось X, FPR – ось Y)
- GINI $= 2 \cdot$ (AUC ROC) $- 1$

1.4 Важные модели

Логистическая регрессия:

- сигмоида: $\pi(z) = \frac{1}{1+e^{-z}}$
- функция потерь (для меток ± 1): $Q(w) = \sum\limits_{i=1}^{l} \ln(1 + e^{-y_i \langle w, x_i \rangle})$

- оценка вероятностей принадлежности классам: $P\{y|x\} = \pi(y\langle w, x\rangle)$
- масштабирование, регуляризация

Случайный лес:

- композиция n_estimators глубоких решающих деревьев (не переобучается при росте количества деревьев)
- уменьшение корреляции деревьев за счет выбора признака из случайного подмножества размера max_features ($d/3$ для регрессии, $\sqrt{d}$ для классификации) – самый важный параметр (нужно настраивать в первую очередь)
- bootstrap = True

Градиентный бустинг:

- композиция n_estimators неглубоких (max_depth как правило 3-6) решающих деревьев
- каждое следующее дерево исправляет ошибки построенной композиции
- борьба с переобучением: уменьшение размера шага learning_rate (с увеличением количества базовых алгоритмов) и бэггинг (обучение базового алгоритма на случайной подвыборке subsample)
- библиотеки: XGBoost, LightGBM, CatBoost

2 Поиск структуры в данных

2.1 Кластеризация:

- KMeans, MiniBatchKMeans, KMeans++ – выпуклые кластеры примерно одинакового размера
- Bisect Means
- GussianMixture (EM-алгоритм) – задача восстановления плотности (кластеры предполагаются выпуклыми)
- DBSCAN – неравные невыпуклые кластеры, при необходимости можно отсеивать выбросы

- AgglomerativeClustering, Ward (евклидово расстояние) – случай большого числа кластеров
- MeanShift
- AffinityPropagation
- SpectraClustering
- Birch

2.2 Отбор признаков:

- Низкая дисперсия (VarianceThreshold)
- Модуль корреляции (линейная информативность) – вещественные признаки и вещественные ответы (бинарные признаки следует кодировать значениями ± 1)
- Площадь под кривой – для задачи бинарной классификации (сортируем признаки по величине площади под кривой и отбираем лучшие)
- Модуль взаимной информации (mutual information) – дискретный признак и дискретный ответ (в т.ч. многоклассовая классификация)
- Жадное добавление и удаление, алгоритм ADD-DEL – много информативных признаков (RFE, RFECV, SelectFromModel)
- Веса при масштабированных признаках в линейных моделях (в т.ч., L1-регуляризатор для обнуления весов)
- Уменьшение критерия информативности при разбиении по признаку – оценка важности признака при построении деревьев (ExtraTreesClassifier)
- Разность ошибок на out-of-bag выборке – оценка важности признака при построении случайного леса

2.3 Понижение размерности:

- Метод случайных проекций, RandomProjection – новые признаки как линейные комбинации исходных, хорошо работает для текстов
- Метод главных компонент, PCA (максимизация дисперсии при понижении размерности) – матрица объекты-признаки должна быть центрирована
- SVD

3 Построение выводов по данным

3.1 Оценки параметров

Несмещенная оценка: $E\hat{\theta} = \theta$. Примеры: $\bar{X}_n$, S_n^2 для $X \sim N(\mu, \sigma^2)$.

Состоятельная оценка: $\forall\ \varepsilon > 0\ P\{|\hat{\theta} - \theta| > \varepsilon\} \to 0$ при $n \to \infty$. Пример: $\bar{X}_n$ и S_n^2 – состоятельные оценки для EX и DX.

Асимптотически нормальная оценка: $\forall x \in \mathbb{R}\ P\{\frac{\hat{\theta}-\theta}{\sigma(\theta)/\sqrt{n}} \leq x\} \to \Phi(x)$ при $n \to \infty$, где $\Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} e^{-y^2/2} dy$. Из асимптотической нормальности следует состоятельность.

Оценка максимального правдоподобия:

$$\hat{\theta}_{\text{МП}} = \text{argmax}_{\theta \in \Theta} \prod_{i=1}^{n} p_X(x_i|\theta) = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ln p_X(x_i|\theta).$$

Состоятельные и асимптотически нормальные. Примеры: $\bar{X}_n$, $\frac{n-1}{n} S_n^2$ для $X \sim N(\mu, \sigma^2)$.

3.2 Распределения

Нормальное распределение: $X \sim N(\mu, \sigma^2)$, $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, $F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int\limits_{-\infty}^{x} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$.

Распределение χ^2 с k степенями свободы: $X = \sum\limits_{i=1}^{k} X_i^2 \sim \chi_k^2$, где $X_i \sim N(0,1)$ – k независимых случайных величин.

Распределение Стьюдента с ν степенями свободы: $X = \frac{X_1}{\sqrt{X_2/\nu}} \sim St(\nu)$, где $X_1 \sim N(0,1)$, $X_2 \sim \chi_\nu^2$.

Распределение Фишера с числом степеней свободы d_1 и d_2: $X = \frac{X_1/d_1}{X_2/d_2} \sim F(d_1, d_2)$, где $X_1 \sim \chi_{d_1}^2$, $X_2 \sim \chi_{d_2}^2$ – независимые случайные величины.

Пусть $X \sim N(\mu, \sigma^2)$, дана выборка $X^n = (X_1, ..., X_n)$. $\bar{X}_n = \frac{1}{n} \sum\limits_{i=1}^{n} X_i$ – выборочное среднее. $S_n^2 = \frac{1}{n-1} \sum\limits_{i=1}^{n} (X_i - \bar{X}_n)^2$ – выборочная дисперсия.

$\bar{X}_n \sim N(\mu, \frac{\sigma^2}{n})$, $(n-1)\frac{S_n^2}{\sigma^2} \sim \chi_{n-1}^2$, $T = \frac{\bar{X}_n - \mu}{S_n/\sqrt{n}} \sim St(n-1)$.

Пусть $X_1 \sim N(\mu_1, \sigma_1^2)$, $X_2 \sim N(\mu_2, \sigma_2^2)$, даны выборки размеров n_1 и n_2.

$\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \sim F(n_1 - 1, n_2 - 1)$.

3.3 Доверительные интервалы

Квантиль порядка α случайной величины X – величина X_α, такая, что $P(X \leq X_\alpha) \geq \alpha$, $P(X \geq X_\alpha) \geq 1-\alpha$.

Предсказательный интервал порядка $1-\alpha$ случайной величины X – отрезок $[X_{\frac{\alpha}{2}}, X_{1-\frac{\alpha}{2}}]$.

Пусть $X \sim N(\mu, \sigma^2)$, дана выборка $X^n = (X_1, ..., X_n)$. Тогда $\bar{X}_n \sim N(\mu, \frac{\sigma^2}{n})$.

Предсказательный интервал для выборочного среднего: $P(\mu - z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}} \leq \bar{X}_n \leq \mu + z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}) = 1-\alpha$.

Доверительный интервал для среднего: $P(\bar{X}_n - z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X}_n + z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}) = 1-\alpha$.

Для среднего: $\bar{X}_n \pm z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}$ – когда дисперсия известна; $\bar{X}_n \pm t_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}$ – когда дисперсия не известна. statsmodels.stats.weightstats: _zconfint_generic, _tconfint_generic.

Для доли: $\hat{p} \pm z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$; улучшение для случаев, когда доля близка к 0 или 1, – доверительный интервал Уилсона. statsmodels.stats.proportion: proportion_confint, samplesize_confint_proportion (размер выборки для интервала заданной ширины)

Для разности долей: $\hat{p_1} - \hat{p_2} \pm z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n_1} + \frac{\hat{p_2}(1-\hat{p_2})}{n_2}}$; для связанных выборок – используем таблицу сопряжённости.

Для не очень удобных статистик (напр., медиана, отношение долей): извлекаем из выборки с возвращением выборки объема n, для каждой вычисляем статистику, оцениваем эмпирическую функцию распределения (numpy.percentile).

3.4 Проверка гипотез

$X^n = (X_1, ..., X_n), X \sim P$.

$H_0 : P \in \omega$, $H_1 : P \notin \omega$ (ω – семейство распределений).

Статистика $T(X^n) \sim F(x)$ при H_0, $T(X^n) \nsim F(x)$ при H_1.

(T, H_0) – **статистический критерий**.

$p-$**value** – вероятность получить такое же значение статистики (как в эксперименте) или еще более экстремальное при справедливости H_0: $p = P(T \geq t|H_0)$.

Ошибка первого рода – "наказание невиновного". **Ошибка второго рода** – "признание невиновным виноватого".

Мощность статистического критерия – вероятность "наказать виноватого".

Корректный критерий имеет вероятность ошибки I рода не больше, чем α. **Идеальный критерий** – корректный критерий с максимальной мощностью.

Биномиальный критерий для доли: stats.binom_test.

Критерий согласия Пирсона (хи-квадрат): stats.chisquare (подчинена ли наблюдаемая случайная величина теоретическому закону распределения). Статистика: $\chi^2 = \sum_{i=1}^{K} \frac{(n_i - np_i)^2}{np_i} \sim \chi^2_{K-1-m}$, K – число карманов, n_i – наблюдаемые частоты, np_i – ожидаемые частоты (должны превышать 5 для 80% карманов), m – количество параметров, оцененных по выборке.

3.5 Параметрические критерии

Критерии Стьюдента:

- условия применимости: нормальное распределение случайных величин; либо $\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$ и $nS^2 \sim \chi^2_{n-1}$ (stats.probplot(dist='chi2', sparams=(n-1))
- z-критерий: statsmodels.stats.weightstats.ztest (одна или две независимые выборки)
- t-критерий: stats.ttest_1samp (одна выборка), stats.ttest_ind (независимые выборки, выборка с большей дисперсией должна быть не меньшего объема), stats.ttest_rel (зависимые выборки)

Проверка нормальности непрерывного признака:

- stats.probplot(dist='norm') похож на прямую → использовать методы, устойчивые к небольшим отклонениям от нормальности (в т.ч. критерии Стьюдента)
- stats.shapiro отвергает нормальность → не использовать методы, чувствительные к отклонениям от нормальности

Критерии для доли:

- z-критерий: statsmodels.stats.proportion.proportions_ztest (две независимые выборки)

3.6 Непараметрические критерии

Критерий знаков: statsmodels.stats.descriptivestats.sign_test (одна выборка или две связанные выборки).

Ранговые критерии: stats.wilcoxon (одна выборка или две связанные выборки), stats.mannwhitneyu (две независимые выбоки)

3.7 АБ-тесты

Планирование эксперимента:

- выбор экспериментальных метрик: чувствительные для АБ-тестирования и хорошо согласуются с целевыми бизнес-метриками
- репрезентативная экспериментальная группа: стратификация по важным показателям или рандомизация
- устойчивость: не видеть значимых изменений там, где их нет, видеть значимые изменения там, где они есть (АА-тест не показывает значимых изменений в интересующих метриках, иначе – менять дизайн)
- размер групп и длительность эксперимента: фиксируем минимальный размер эффекта и допустимые вероятности ошибок I и II рода (например, 0.05 и 0.2), рассчитываем размер групп исходя из зафиксированных параметров с помощью калькулятора мощности

Проверка гипотез и принятие решений:

- H_0: изменение не повлияло на пользователей
- H_1: изменение повлияло на пользователей

3.8 Корреляции

Корреляция Пирсона: сила линейной взаимосвязи, неустойчива к выбросам.

DataFrame.corr(method='pearson'), scipy.stats.pearsonr.

Корреляция Спирмена: сила монотонной взаимосвязи, устойчива к выбросам.

DataFrame.corr(method='spearman'), scipy.stats.spearmanr.

Корреляция Мэтьюса: сила взаимосвязи между бинарными переменными.

Коэффициент V Крамера: сила взаимосвязи между категориальными переменными, не может быть отрицательным (проверить условия применимости критерия хи-квадрат: $n \geq 40$, $\frac{n_{i+}n_{+j}}{n} < 5$ не более, чем в 20% ячеек).

Значимость корреляции:

- непрерывные величины – (scipy.stats.pearsonr, scipy.stats.spearmanr)
- бинарные и категориальные величины – критерий хи-квадрат, $n \geq 40$, $\frac{n_{i+}n_{+j}}{n} < 5$ не более, чем в 20% ячеек (scipy.stats.chi2_contingency(correction=False), при True критерий более консервативный); когда мало данных – точный критерий Фишера, для таблиц 2×2 (scipy.stats.fisher_exact)

3.9 Множественная проверка гипотез

Поправка Бонферрони: достигаемые уровни значимости сравниваются с $\frac{\alpha}{m}$ – перестраховываемся в отношении ошибок первого рода, совершаем больше ошибок второго рода (мощность снижается).

Метод Холма: нисходящая процедура проверки, $\alpha_i = \frac{\alpha}{m-i+1}$ – всегда мощнее, чем метод Бонферрони. statsmodels.sandbox.stats.multicomp.multipletests(method = 'holm').

Метод Бенджамини-Хохберга: восходящая процедура проверки, $\alpha_i = \frac{\alpha_i}{m}$ – допускается больше ошибок первого рода (доля отвергаемых верных гипотез не более α), критически увеличивается мощность. Важное условие: независимость статистик, которые проверяют гипотезы. statsmodels.sandbox.stats.multicomp.multipletests(method = 'fdr_bh').

3.10 Регрессия

$$E(y|x) = \beta_0 + \sum_{j=1}^{k} \beta_j x_j$$

Total Sum of Squares: $TSS = \sum\limits_{i=1}^{n} (y_i - \bar{y})^2$.

RSS: $RSS = \sum\limits_{i=1}^{n} (y_i - \hat{y}_i)^2 \to 0$ (МНК – минимизация квадратов ошибок).

R^2**:** $R^2 = 1 - \frac{RSS}{TSS}$.

Предположения МНК ($\hat{\beta}$ – несмещенные и состоятельные оценки истинных β):

- истинная модель – линейна: $y = X\beta + \varepsilon$ (анализ графика остатков в зависимости от признака)
- объекты дают независимую выборку наблюдений

- ни один из признаков не является линейной комбинацией других: rank $X = k+1$
- ошибка случайна: $E(\varepsilon|x) = 0$ (проверка гипотезы)

Предположения Гаусса-Маркова (МНК-оценки имеют наименьшую дисперсию в классе всех оценок β, линейных по y):

- предположения МНК
- гомоскедастичность ошибки (дисперсия ошибки не зависит от значений признака): $D(\varepsilon|x) = \sigma^2$ (statsmodels.stats.api.het_breuschpagan)

$$D(\hat{\beta}_j) = \frac{\sigma^2}{TSS_j(1-R_j^2)}$$

Мультиколлинеарность – близкая к линейной зависимость признаков ($R_j^2 \approx 1$).

Предположения о нормальности ошибки (МНК-оценки совпадают с оценками максимального правдоподобия):

- предположения Гаусса-Маркова
- нормальность ошибки: $\varepsilon|x \sim N(0, \sigma^2)$ (stats.shapiro, stats.probplot(dist='norm'))

При выполнении описанных предположений можно строить доверительные интервалы для β_j, доверительный интервал для среднего отклика $E(y|x)$ и предсказательный интервал для значения $y|x$.

4 Прикладные задачи анализа данных

4.1 Временные ряды

Подбор модели в классе ARIMA:

- Визуальный анализ ряда: сезонность, пропуски, выбросы, необходимость стабилизации дисперсии, необходимость исключения из рассмотрения начала ряда
- Стабилизация дисперсии (метод Бокса-Кокса) – при необходимости (scipy.stats.boxcox)
- Выбор порядка дифференцирования (стационарность – критерий Дики-Фуллера, tsa.stattools.adfuller)

- Выбор начальных приближений p, q, P, Q с помощью графиков автокорреляционной (graphics.tsa.plot_acf) и частичной автокорреляционной (graphics.tsa.plot_pacf) функций

- Выбор модели с помощью информационного критерия Акаике (tsa.statespace.SARIMAX, model.aic)

- Анализ остатков (несмещённость, стационарность, неавтокоррелированность) и модификация модели

- Прогнозирование

4.2 Метрики

CPA (cost per acquisition) $= \frac{\text{total advertisement spend}}{\text{number of registered users}}$.

CPI (cost per install) $= \frac{\text{total advertisement spend}}{\text{number of installs}}$.

ROI (return on investment) $= \frac{\text{total revenue} - \text{total cost}}{\text{total cost}}$.

ARPU – average revenue per user.

LTV (lifetime value) $= \frac{\text{ARPU}}{\text{lifetime}}$.

RR (return rate) $= \frac{\text{current number of customers from the original set}}{\text{number of customers at the original set}}$.

CR (churn rate) $= \frac{\text{number of churned customers}}{\text{total number of customers}}$.

4.3 Анализ текстов

Предобработка текста:

- токенизация – разбиение текста на слова

- нормализация – приведение слов к начальной форме: стемминг (стрижка окончаний, не всегда работает) и лемматизация (приведение к нормальной форме по словарю, медленнее стемминга, но работает лучше)

Извлечение признаков из текста:

- счетчики слов (CountVectorizer): признаки – слова (удаляем стоп-слова и редкие слова), значения признаков – доли вхождений слов в документ

- TF-IDF$(d,w) = n_{dw} \log \frac{l}{n_w}$ (TfidfVectorizer), n_{dw} – доля вхождений слова w в документ d, l – общее количество документов, n_w – число документов, в которых w встречается хотя бы раз (после удаления стоп-слов)
- N-граммы (N подряд идущих слов в тексте) и skip-граммы (наборы из N токенов, расстояние между соседними не превышает k), коллокации – для обогащения признакового пространства
- векторное представление слов (word2vec) как способ снизить размерность: похожие слова имеют близкие векторы небольшой размерности d, векторное представление документа – усреднение / сложение векторов слов, входящих в документ

Обучение моделей:

- подготовка выборки: извлечение признаков, отбор признаков (корреляция с целевой переменной, понижение размерности с помощью PCA)
- обучение модели: байесовские методы и линейные модели для больших размерностей, в случае векторных представлений можно использовать и более сложные модели

4.4 Ранжирование и рекомендательные системы

Точность ранжирования:

- $y(q,d) \in \{0,1\}$, y – целевая переменная, q – запрос, d – документ
- $d_q^{(i)}$ – i-й по релевантности документ для запроса q и ранжирующей модели $a(q,d)$
- Precision@$k(q) = \frac{1}{k} \sum\limits_{i=1}^{k} y(q, d_q^{(i)})$ (не учитывает позиции релевантных документов)
- AP@$k(q) = \frac{\sum_{i=1}^{k} y(q,d_q^{(i)}) \text{Precision@}i(q)}{\sum_{i=1}^{k} y(q,d_q^{(i)})}$ (учитывает позиции релевантных документов)
- MAP@$k = \frac{1}{|Q|} \sum\limits_{q \in Q} \text{AP@}k(q)$

DCG:

- $y(q,d) \in \mathbb{R}$, y – целевая переменная, q – запрос, d – документ
- $d_q^{(i)}$ – i-й по релевантности документ для запроса q и ранжирующей модели $a(q,d)$

- $\mathrm{DCG@}k(q) = \sum\limits_{i=1}^{k} \frac{2^{y(q,d_q^{(i)})}-1}{\log(i+1)}$ (учитывает и релевантность, и позицию документа)
- $\mathrm{nDCG@}k(q) = \frac{\mathrm{DCG@}k(q)}{\max \mathrm{DCG@}k(q)}$ (нормировка на значение при идеальном ранжировании)

Методы ранжирования:

- pointwise (поточечный): релевантность $a(q, d)$ оценивается непосредственно для каждого объекта
- pairwise (попарный): минимизация количества дефектных пар, функционал ошибки оценивается сверху гладкой функцией
- listwise (списочный): оптимизация nDCG

Оценка качества рекомендаций (оффлайн):

- точность: $\mathrm{Precision@}k = \frac{\text{купленное из рекомендованного}}{k}$, $\mathrm{AP@}k$ – усредненный по сессиям $\mathrm{Precision@}k$
- полнота: $\mathrm{Recall@}k = \frac{\text{купленное из рекомендованного}}{\text{количество покупок}}$, $\mathrm{AR@}k$ – усредненный по сессиям $\mathrm{Recall@}k$, Взвешенный ценами $\mathrm{Recall@}k = \frac{\text{стоимость купленного из рекомендованного}}{\text{стоимость покупок}}$