Рекомендательные системы

Look-a-like выделение сегментов пользователей

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Ставим “+”, если все хорошо
“-”, если есть проблемы

Тема вебинара

Look-a-like выделение сегментов пользователей

Елена Позднеева

Team Lead Data Scientist, MegaTex

Как подобрать **персональный** тариф?
Как **удержать/привлечь** клиента?
Как принести компании **финансовый** эффект?
Как повышать **лояльность** клиента?

Контакты:
@frideliya
www.linkedin.com/in/elena-pozdneeva

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в учебной группе **#RecSys-2024-10**
https://t.me/+tr0h3CYc3xs4OTdi

Задаем вопрос
в чат или голосом

Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения

Индивидуально

Время, необходимое
на активность

Пишем в чат

Говорим голосом

Документ

Ответьте себе или
задайте вопрос

Маршрут вебинара

- Что такое look-a-like сегментация
- Оценка качества
- “Матчасть” логистической регрессии
- Деревья решений и ансамбли
- Практика
- Рефлексия

Цели вебинара

К концу занятия вы сможете

1. Формализовать задачу look-a-like
2. Научиться находить похожих пользователей при помощи моделей машинного обучения

Смысл

Зачем вам это уметь

1. Понимать, в каких случаях подходить к решению задач с точки зрения look-a-like
2. Выбирать подходящую модель
3. Научиться разрабатывать модели машинного обучения для задач look-a-like

Строили ли вы модели look-a-like методами машинного обучения?

1 - да
0 - нет

Что такое look-a-like сегментация

Введение

Цель: продать как можно больше слонов

Как: продвигать предложение среди тех, кто с наибольшей вероятностью совершит целевое действие

Подобрать аудиторию экспертно:
- зоопарки
- те, кто покупал других крупных животных…

Если ранее уже продавали и есть покупатели?
- ищем среди новой аудитории тех, кто похож на купивших

Определение и примеры

Look-a-like - подход в моделировании, направленный на поиск аудитории, максимально похожей на целевое множество пользователей

Маркетинг и реклама

Компания использует данные о своих лучших клиентах для создания Look-a-like сегментов. Затем показывает рекламу новым сегментам, чтобы привлечь клиентов с похожими характеристиками.

Электронная коммерция

Интернет-магазины могут рекомендовать товары новым пользователям на основе поведения и предпочтений существующих клиентов с похожими профилями.

Медиа и развлечения

Стриминговые сервисы могут использовать Look-a-like сегментацию для рекомендации фильмов, сериалов или музыкальных треков новым пользователям на основе предпочтений пользователей с похожими интересами.

Финансовые услуги

Банки могут использовать Look-a-like сегментацию для предложения кредитных карт или кредитов новым клиентам, основываясь на характеристиках текущих клиентов с хорошей кредитной историей.

Игровая индустрия

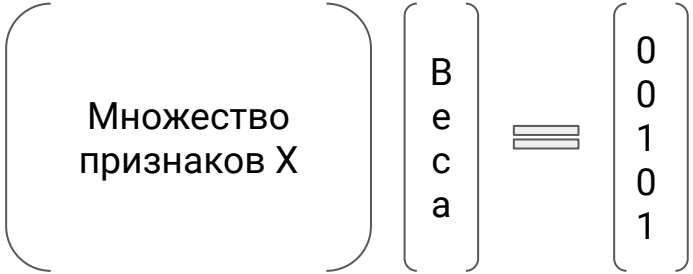
Разработчики игр могут использовать Look-a-like сегментацию для нахождения новых игроков, которые с большей вероятностью будут вовлечены в игру и совершать внутриигровые покупки.

Образование и online-курсы

Платформы могут рекомендовать курсы новым пользователям на основе предпочтений и поведения существующих студентов с аналогичными интересами и обучающими целями.

Формализация

Задача: Обучить модель, которая сможет предсказывать метку **y (0 или 1)** на основе входных данных **X**.



HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	1	101348.88	1
0	1	112542.58	0
1	0	113931.57	1
0	0	93826.63	0
1	1	79084.10	0

Пример из практики

Склонность к приобретению карты лояльности

Задача: Выявить множество клиентов, склонных к приобретению карты лояльности при коммуникации

Целевая переменная:

- Клиенты, которые приобрели карту в течение месяца после коммуникации = класс 1.
- Клиенты, которые не приобрели карту после коммуникации = класс 0.

Признаки:

- профиль клиента (возраст, город)
- агрегаты (кол-во покупок за прошедший месяц, средняя сумма покупок за квартал)

Обучение: бинарная классификация, бустинг с logloss, метрика F1-score

Применение:

- выбрать множество клиентов без карты
- проскорить их моделью
- выбрать топ-N и запустить на них коммуникацию

Методы

- Логистическая регрессия
- Решающее дерево
- Случайный лес
- Бустинг

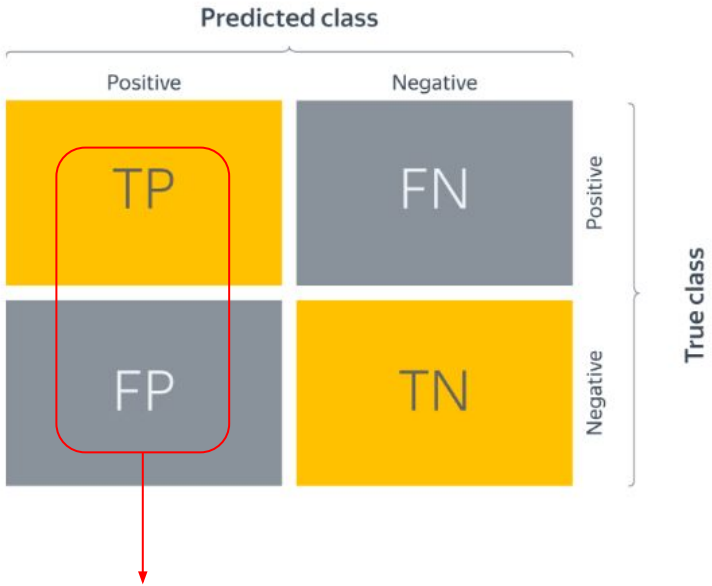
Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Метрики качества

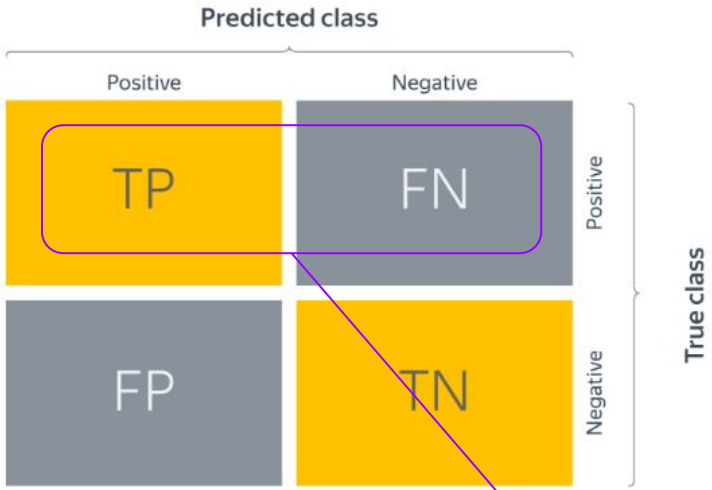
Метрики классификации



Predict	Target	
0	0	TN
0	1	FN
1	0	FP
1	1	TP

$$\text{Precision} = \frac{TP}{TP + FP}$$

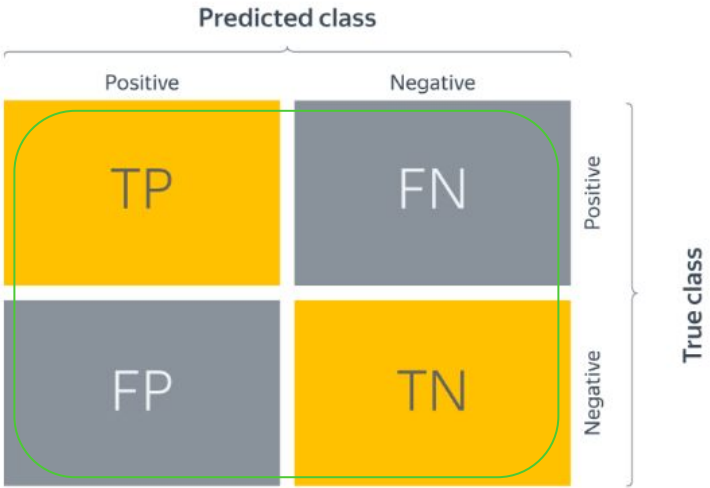
Метрики классификации



Predict	Target	
0	0	TN
0	1	FN
1	0	FP
1	1	TP

$$\text{Recall} = \frac{TP}{TP + FN}$$

Метрики классификации



Predict	Target	
0	0	TN
0	1	FN
1	0	FP
1	1	TP

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Метрики

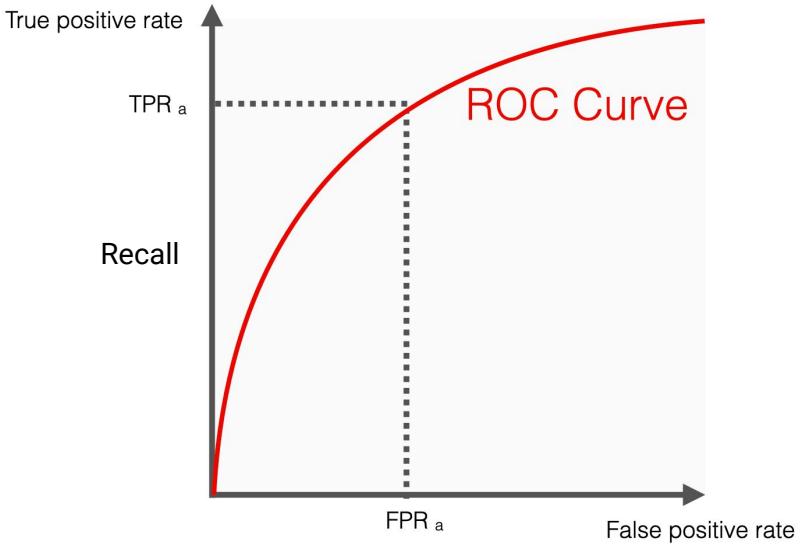
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F_1 = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$F_\beta = (\beta^2 + 1)\frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \beta^2 \cdot \text{Precision}}$$



$$FPR = \frac{FP}{FP + TN}$$

https://habr.com/ru/companies/ods/articles/328372/
и еще немного про ROC_AUC

Когда какую метрику использовать

Метрика	Когда использовать	Пример использования
Precision (Точность)	Когда важна точность положительных предсказаний и стоимость ложных срабатываний высока	Спам-фильтры: минимизация ложных положительных результатов, чтобы легитимные письма не попадали в спам
Recall (Полнота)	Когда важно обнаружение всех положительных случаев и стоимость пропущенных случаев высока	Обнаружение болезней: важно найти все случаи болезни, даже если это означает больше ложных срабатываний
F1-Score	Когда нужен баланс между precision и recall и важно учесть и ложные срабатывания, и пропущенные случаи	Классификация редких событий, где требуется равное внимание к precision и recall (например, обнаружение мошенничества)
Accuracy (Точность)	Когда классы сбалансированы и одинаково важно правильное предсказание как положительных, так и отрицательных случаев	Общее прогнозирование в сбалансированных наборах данных, где нет перевеса одного класса
ROC AUC	Когда нужно оценить модель по всем возможным порогам вероятности и сравнить классификаторы на их способности различать классы	Классификация клиентов по вероятности оттока: оценка модели по всем возможным порогам для принятия решений

Логистическая регрессия

В предыдущих сериях…

Логистическая регрессия - метод статистического анализа, используемый для предсказания бинарного исхода (класса) на основе одной или нескольких независимых переменных.

Математическая модель: $P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n)}}$

$P(Y = 1 \mid X)$ - вероятность положительного исхода

X_i - независимые переменные

β_i - коэффициенты модели показывают, как изменение независимой переменной влияет на вероятность положительного исхода. Положительное значение коэф-та указывает на увеличение вероятности события с увеличением значения переменной

https://education.yandex.ru/handbook/ml/article/linear-models

Математика логистической регрессии

Логистическая регрессия - метод статистического анализа, используемый для предсказания бинарного исхода (класса) на основе одной или нескольких независимых переменных.

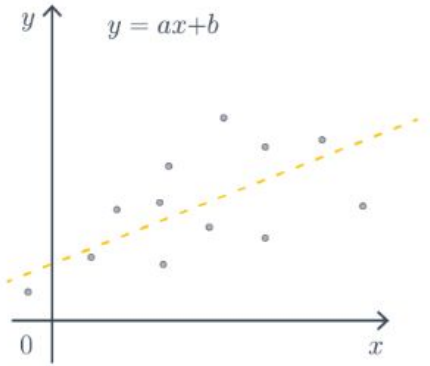
Для линейной регрессии:

$$\hat{f}(x_1, \ldots, x_n) = \mathbf{w}_0 + \mathbf{w}_1 x_1 + \ldots + \mathbf{w}_n x_n$$

$$\mathbf{X} \in \mathbb{R}^{l \times n}, \mathbf{y} \in \mathbb{R}$$

$$\hat{f}(\mathbf{x}) = \sum_{i=0}^{n} \mathbf{w}_i \mathbf{x}_i = \mathbf{x} \cdot \mathbf{w}$$

$$\begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ & \ddots & & \\ & & \ddots & \\ & & & \ddots \\ x_{l1} & x_{l2} & \ldots & x_{ln} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \ldots \\ w_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ \ldots \\ \ldots \\ y_l \end{pmatrix}$$

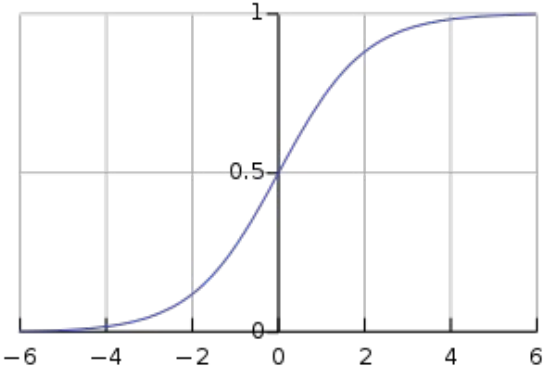


https://education.yandex.ru/handbook/ml/article/linear-models

Математика логистической регрессии

Как перейти к бинарному выходу модели?

1. Предсказываем не 0 и 1, а вероятность принадлежности к классу 1
2. Масштабируем выход линейной модели к интервалу [0; 1]



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

⟹ применяем к линейной модели сигмоиду

$$\hat{f}(\mathbf{x}) = \sigma\left(\sum_{i=0}^{n} \mathbf{w}_i \mathbf{x}_i\right)$$

https://education.yandex.ru/handbook/ml/article/linear-models

Плюсы и минусы

Плюсы	Минусы
Простота и интерпретируемость	Чувствительность к выбросам
Малое количество гиперпараметров	Проблемы с несбалансированными данными
Эффективность на линейно разделимых данных	Невозможность нелинейного разделения
Выходы вероятностей	Требует достаточного количества данных
Меньше склонна к переобучению	Сложности с интерпретацией для большого числа признаков
Устойчивость к мультиколлинеарности благодаря регуляризации	

Вопросы?

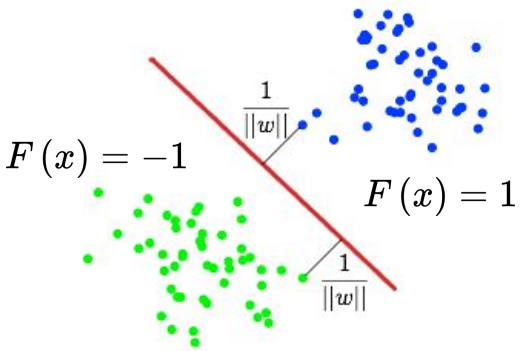
Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

SVM (метод опорных векторов)

Определение

SVM (Support Vector Machine) - это мощный алгоритм машинного обучения, используемый для задач классификации и регрессии. Основная идея заключается в нахождении **оптимальной гиперплоскости**, которая максимально разделяет данные двух классов.



Гиперплоскость в двумерном пространстве.

В многомерном пространстве: $w \cdot x + b = 0$
где **w** вектор весов, **x** вектор признаков,
b смещение

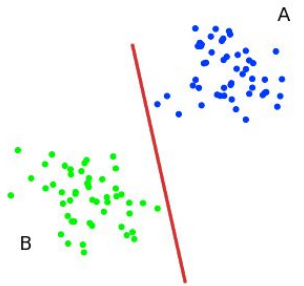
SV (опорные вектора) - векторы, которые являются ближайшими точками к гиперплоскости.

Цель: построить классифицирующую функцию $F(x) = \text{sign}(w \cdot x + b)$

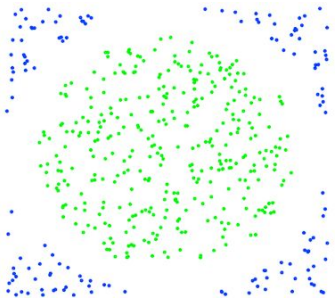
$$\begin{cases} \arg \min\limits_{\mathbf{w},b} ||w||^2, \\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geqslant 1, \quad i = 1, \ldots, m. \end{cases}$$

http://www.ccas.ru/voron/download/SVM.pdf
https://habr.com/ru/articles/105220/

Ядра в SVM



Разделить на классы легко



А здесь?..

Ядра используются для отображения признаков в новое пространство, где классы линейно разделимы.

Новая классифицирующая функция: $F(x) = \text{sign}(w \cdot \varphi(x) + b)$

Ядро: $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$

Полиномиальное: $K(x_i, x_j) = (x_i \cdot x_j + c)^d$

Радиальное базисное: $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$

Сигмоидное: $K(x_i, x_j) = \tanh(\alpha(x_i \cdot x_j) + c)$

Плюсы и минусы

Плюсы	Минусы
Эффективность на малых выборках	Требовательность к вычислительным ресурсам
Гибкость через ядровые методы	Сложность интерпретации
Хорошая обобщающая способность	Чувствительность к выбору ядра и параметров
Устойчивость к переобучению	Чувствительность к шуму и выбросам

Вопросы?

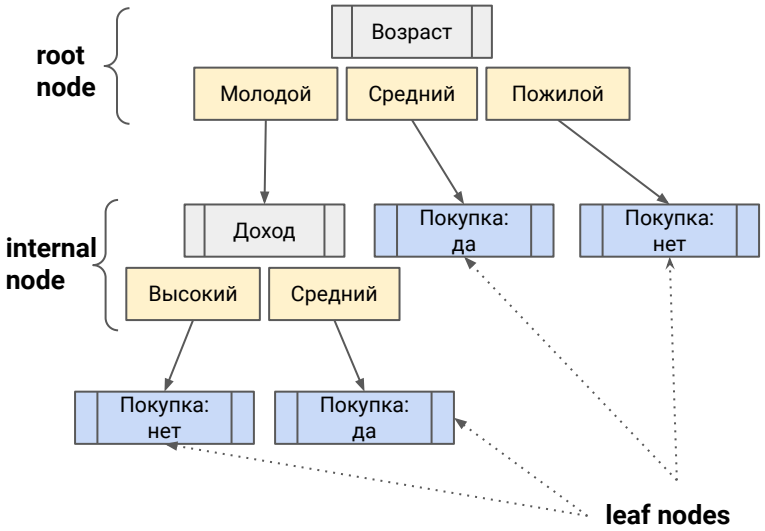
Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Деревья решений и ансамблевые методы

Дерево решений

Дерево решений — это алгоритм машинного обучения, который представляет собой модель в виде дерева, где каждый **внутренний узел соответствует условию на один из признаков**, каждая **ветвь — результат этого условия (правда или ложь)**, а каждый **листовой узел — метка класса или значение**, которое предсказывается.



Параметр	Описание	Влияние
max_depth	Максимальная глубина дерева	Контролирует, насколько глубоко дерево может разветвляться
min_samples_split	Минимальное количество образцов, необходимое для разделения внутреннего узла	Помогает предотвратить создание слишком мелких узлов
min_samples_leaf	Минимальное количество образцов, которое должно быть в листовом узле	Уменьшает переобучение, обеспечивая, что узлы не будут создаваться на основе очень небольших наборов данных
max_features	Максимальное количество признаков, которое будет учитываться при поиске наилучшего разделения	Уменьшает размер дерева и помогает в предотвращении переобучения
criterion	Функция оценки качества разделений	Определяет способ измерения качества разделений

Ансамбли

Ансамбль — набор моделей для решения задачи. Ансамбль обеспечивает в среднем меньшую ошибку по сравнению с одной моделью.

Случайный лес

- состоит из базовых моделей
- базовая модель - решающее дерево
- базовые модели обучаются независимо
- результаты базовых моделей усредняются

Бустинг

- состоит из базовых моделей
- базовая модель - любая
- базовые модели обучаются последовательно, учитывая ошибки и прогнозы с предыдущих шагов

Бустинг

Итоговая модель = линейная комбинация базовых моделей: $a_N(x) = \sum_{k=1}^{N} \alpha_k b_k(x)$

Базовая модель:

$$b_1 = \arg\min L(y, b(x))$$
$$a_1 = b_1$$
$$e_1 = y - b_1(x) \quad - \quad \text{остаток}$$

$L(y, b(x))$
функция потерь базовой модели

$$b_2 = \arg\min L(e_1, b(x))$$
$$a_2 = b_1 + b_2 = \sum_{k=1}^{2} b_k(x)$$
$$e_2 = y - a_2(x) = y - \sum_{k=1}^{2} b_k(x)$$

$$b_{N-1} = \arg\min L(e_{N-2}, b(x))$$
$$a_{N-1} = \sum_{k=1}^{N-1} b_k(x)$$
$$e_{N-1} = y - a_{N-1}(x) = y - \sum_{k=1}^{N-1} b_k(x)$$

$$b_N = \arg\min L(e_{N-1}, b(x))$$
$$a_N(x) = \sum_{k=1}^{N} \alpha_k b_k(x)$$
$$e_N = y - a_N(x) = y - \sum_{k=1}^{N} b_k(x)$$

Сравнение подходов

Критерий	Решающее дерево	Случайный лес	Бустинг
Применение	Простые задачи классификации и регрессии	Задачи с большим количеством признаков и данных	Задачи, требующие высокой точности, с разными типами данных
Интерпретируемость	Высокая	Средняя	Низкая
Преимущества	Простота и интерпретируемость	Высокая точность	Высокая точность
	Не требует нормализации данных	Устойчивость к переобучению и выбросам	Работает хорошо на разных типах данных
Недостатки	Склонность к переобучению	Не всегда эффективен для задач с малым количеством данных	Сложность настройки гиперпараметров
	Плохо справляется с очень большими наборами данных	Меньшая интерпретируемость	

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

LIVE

Цели вебинара

Проверка достижения целей

1. Формализовали задачу look-a-like
2. Разобрались с теорией основных подходов к моделированию
3. Научиться находить похожих пользователей при помощи моделей машинного обучения

Вопросы для проверки

По пройденному материалу всего вебинара

1. Что решает задача look-a-like?
2. Какие метрики бинарной классификации вы знаете?
3. Подходит ли SVM для нелинейной задачи?
4. В чем главное отличие случайного леса от бустинга?

Ключевые тезисы занятия

Подведем итоги

1. Задача look-a-like подразумевает поиск аудитории, максимально похожей на целевое множество пользователей
2. Логистическая регрессия является линейным методом решения задачи бинарной классификации
3. SVM является эффективным методом, когда классы не являются линейно разделимыми
4. Решающее дерево - простой интерпретируемый метод классификации (регрессии)
5. Бустинги являются эффективными моделями для решения задач на табличных данных

Рефлексия

Рефлексия

Будете ли применять на практике то,
что узнали на вебинаре?

Следующий вебинар

14 ноября 2024

Модели Next Best Action

Ссылка на вебинар
будет в ЛК за 15 минут

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Спасибо за внимание!

Приходите на следующие вебинары

Елена Позднеева

Team Lead Data Scientist, МегаТех

@frideliya + чат группы