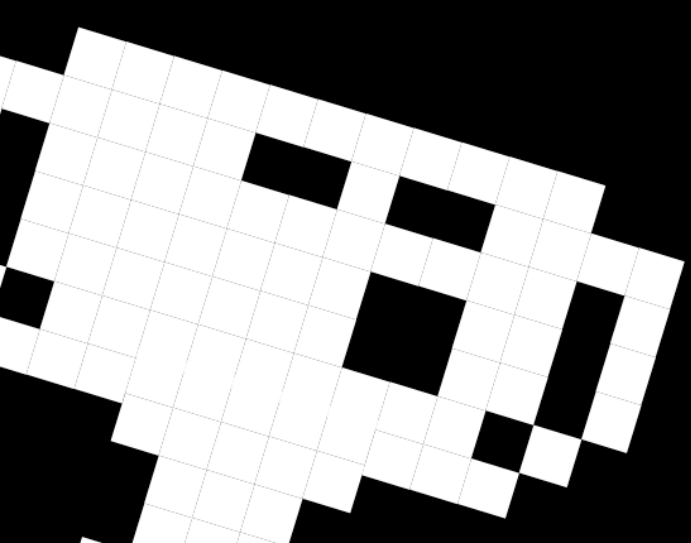
Хакатон

25 декабря
››› 25 января

Постановка задачи

ТРЕК 2



Краткое описание задачи

Рекламные кампании в Telegram требуют точного прогнозирования бюджета. Задача команды — разработать модель машинного обучения, которая на основе исторических данных будет предсказывать количество просмотров (охват) рекламного объявления, используя в качестве основных входных параметров CPM (стоимость за 1000 показов) и идентификатор канала размещения.

Полное описание задачи

Проблематика

Планирование рекламных бюджетов в Telegram часто происходит "вслепую", без точного понимания, какой охват будет достигнут при заданном CPM на конкретном канале. Это приводит к неэффективному распределению бюджета и сложностям в оценке KPI.

Продуктовая цель

Создать работающий прототип ML-модели и веб-сервис (API, без интерфейса), который позволяет менеджерам по рекламе и маркетологам оценивать потенциальный охват объявления до его запуска. Это позволит оптимизировать бюджет кампаний и повысить эффективность медиапланирования.

Концепция решения

Участникам предоставляется исторический датасет в формате CSV, содержащий агрегированную статистику по рекламным объявлениям в Telegram за два года. На основе этих данных необходимо построить модель машинного обучения, которая будет прогнозировать ключевой показатель эффективности — количество просмотров (VIEWS).

Процесс работы модели

Обучение: модель обучается на исторических данных, где каждое наблюдение содержит:

- AD_ID — уникальный идентификатор объявления.
- CPM — стоимость за тысячу показов (основной финансовый параметр).
- VIEWS — целевая переменная (количество просмотров).
- CLICKS и ACTIONS — количество кликов по объявлению и целевых действий заложенных в объявление.
- CHANNEL_NAME — канал размещения (категориальный признак).
- DATE — дата размещения (для учета временных трендов и сезонности).

Прогноз: после обучения модель должна выполнять прогноз по трем входным параметрам

- CPM (стоимость за тысячу показов).
- Дата размещения.
- Канал размещения.

Задача участников — провести исследовательский анализ данных (EDA), создать и обучить модель регрессии, которая максимально точно предсказывает VIEWS на основе заданных параметров. Разрешено обогащать данные через внешние API аналитики Telegram-каналов (например, TgStat) для получения дополнительных характеристик каналов.

Форма решения

Веб-сервис с REST API endpoint, принимающий входные параметры и возвращающий прогноз. Без необходимости разработки интерфейса системы, тестирование и оценка будет проводиться с помощью автоматизированной системы проверки.

Теоретическое пояснение: Как работает прогнозирование охватов в digital-рекламе

Маркетинговый контекст задачи

В digital-маркетинге, особенно в контексте рекламы в социальных сетях и мессенджерах, эффективное планирование бюджета это критически важная задача. Рекламодатели стремятся максимизировать охват аудитории при заданном бюджете. Ключевым финансовым параметром здесь выступает CPM (Cost Per Mille — стоимость за тысячу показов).

Почему CPM, канал и дата это ключевые параметры для прогноза?

1. CPM - это прямая "цена видимости". Чем выше CPM, тем более ценным (или конкурентным) считается показ в системе аукциона рекламной платформы. Исторически между CPM и охватом существует нелинейная зависимость: после определенного порога увеличение CPM не дает пропорционального роста просмотров. Модель должна выявить эту зависимость для каждого канала.

2. Канал размещения (CHANNEL_NAME) - это, по сути, портрет аудитории. У каждого Telegram-канала есть свои ключевые характеристики, невидимые в исходных данных, но критически важные для прогноза:

- **Количество подписчиков** (охват потенциальной аудитории).
- **ER (Engagement Rate)** — вовлеченность аудитории, которая влияет на виральность контента и органический прирост просмотров.
- **Тематика канала и ЦА** (целевая аудитория): финансовый канал и развлекательный будут иметь разную конверсию из показа в просмотр.
- **Историческая «цена» аудитории канала.** В том числе для получения этих признаков разрешено использовать внешние API (TgStat).

3. Дата размещения (DATE) — фактор временных трендов и сезонности. Активность аудитории в Telegram неоднородна:

- **День недели и время суток** (вечером активность выше).
- **Сезонность** (праздники, дни зарплат, конец месяца).
- **Общественные или рыночные события,** вызывающие всплеск активности в тематических каналах.

Как модель машинного обучения создает прогноз?

Модель, обученная на исторических данных, по сути, автоматически обнаруживает сложные, неочевидные для человека взаимосвязи между этими параметрами. Она отвечает на вопросы к примеру вот такие:
«На сколько просмотров в среднем можно рассчитывать на канале X в пятницу при CPM = Y?»
«Как изменится охват, если повысить ставку CPM на 20%?»
«На каком из двух каналов с одинаковой тематикой наш бюджет даст больший охват в »

Бизнес-ценность решения

Итоговая модель - это инструмент для data-driven медиапланирования. Она позволяет:

- Оптимизировать бюджет: Распределять средства между каналами не на основе интуиции, а на основе предсказанной отдачи.
- Вести переговоры: Обоснованно обсуждать стоимость размещения с владельцами каналов.
- Прогнозировать KPI: Заранее оценивать, каких показателей охвата и, косвенно, кликов (через историческую конверсию VIEWS → CLICKS) можно достичь.
- Снижать риски: Избегать неэффективных вложений в каналы, где высокая стоимость CPM не конвертируется в адекватный охват.

Требования к решению

Требования к модели и данным

- Проведение EDA, очистка и предобработка исторических данных.
- Разработка и обучение модели машинного обучения (регрессии) для прогнозирования VIEWS.
- Использование в качестве основных входных признаков: CPM,CHANNEL_NAME,DATE.
- Разрешено обогащать данные с помощью внешних API (статистика каналов).
- Основная метрика качества: сравнение реальных («VIEWS») и предсказанных моделью значений на тестовой выборке.

Требования к реализации

- Код проекта на Python с использованием библиотек для анализа и ML (к примеру pandas, numpy, scikit-learn, xgboost, catboost и т.д.).
- Веб-сервис (API endpoint), развернутый локально или в облаке, но имеющий доступ из открытой сети для проведения проверки.
- Endpoint должен принимать входные параметры (например, в формате JSON: {"cpm": float, "channel": str, “data”: str} и возвращать прогноз в формате {"predicted_views": int}.
- Четкое описание выбранного подхода, использованных признаков и метрик качества модели.

Требования к проекту

- Полный код решения представленный в Git-репозиторий на GitHub
- README.md с инструкцией по запуску модели и API.
- requirements.txt со списком зависимостей при необходимости.
- Скриншоты при необходимости и для наглядности или описание рабочего процесса.

Технические требования

Общие требования

- Язык программирования: Python.
- Библиотеки: любые открытые библиотеки для анализа данных и ML (к примеру pandas, numpy, scikit-learn, lightgbm, catboost и др.).
- Для веб-сервиса (API): на ваше усмотрение
- СУБД: технология и вообще наличие на ваше усмотрение.

Требования к реализации

- Рабочий прототип, развернутый локально или в облаке, но имеющий доступ из открытой сети для проведения проверки.
- Доступ к API endpoint должен быть обеспечен в течение 10 рабочих дней после завершения соревнования.
- Репозиторий на GitHub с полным кодом, инструкцией по запуску и скриншотами/описанием работы.

Дополнительные условия

- Разрешено использование только открытых API сторонних сервисов аналитики (TgStat, TgMaps и аналоги).
- Использование открытых предобученных моделей и библиотек разрешено.