ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет экономических наук

Домашняя работа по дисциплине "Машинное обучение в экономике"

ФИО	**Группа**	**Вклад**
Иванова М.	БЭК223	33.(3)%
Салман Я.	БЭК223	33.(3)%
Юманова А.	БЭК223	33.(3)%

Github:
https://github.com/Aurumlin/ml_econ

Руководители:
Погорелова П.В.
Потанин Б.С.

Москва 2025

Содержание

1 Обоснование темы

1.1 Задание 1.1

Придумайте непрерывную зависимую (целевую) переменную (например, заработная плата или прибыль) и бинарную переменную воздействия (например, образование или факт занятий спортом).

Целевая переменная: выручка в мобильном приложении, (стоимость покупки в приложении за все время)

Бинарная переменная: Подписка на push-уведомления (0 – пользователь не подписан на push-уведомления, 1 – подписан).

1.2 Задание 1.2

Опишите, для чего может быть полезно изучение влияния переменной воздействия на зависимую переменную. В частности, укажите, как эта информация может быть использована бизнесом или государственными органами. Цель исследования: проанализировать, как подписка пользователя на push-уведомления влияет на вероятность совершения покупки в мобильном приложении. Определить, являются ли push-уведомления эффективным маркетинговым инструментом для стимулирования пользователей к покупкам.

Для бизнеса:

• Оптимизация коммуникации с клиентами: улучшение стратегии взаимодействия с клиентами: когда, как часто и каким пользователям стоит отправлять уведомления. Сегментация аудитории и улучшение клиентского опыта.

• Повышение ROI: если уведомления повышают конверсию (а значит выручку с пользователя), то бизнес может инвестировать в развитие данного способа коммуникации.

• Эффективное распределение бюджета: анализ позволяет оценить эффективность push-уведомлений и корректно распределить бюджет

Для государства:

• Цифровизация и рост лояльности граждан: государственные приложения и сервисы могут использовать push-уведомления для напоминаний или уведомлений граждан. Это позволит сделать сервисы более удобными и популярными, а граждан лояльными.

• Развитие цифровых коммуникаций в B2C секторе: государство может сформировать стандарты и рекомендации этичного пользования push-уведомлений. (допустимая частота, формат уведомлений, недопустимая лексика)

• Повышение эффективности государственных мобильных сервисов: уведомления в сервисах для бизнеса (Налоги, Госуслуги и тд). Адаптация push-уведомлений, для повышения отклика пользователей.

1.3 Задание 1.3

Обоснуйте наличие причинно-следственной связи между зависимой переменной и переменной воздействия. Приведите не менее 2-х источников из научной литературы, подкрепляющих ваши предположения.

Подписка на push-уведомления влияет на вероятность совершения покупки в мобильном приложении. Это связано с коммуникацией, которая:

• возвращает пользователя в приложение,

• напоминает о действиях (например, оставленный товар в корзине),

• стимулирует покупку с помощью акций, скидок или персональных предложений.

Это ПСС, так как:

• push-уведомления выступают, как внешний триггер, который воздействует на пользователя независимо от текущего поведения

• последовательная временная структура, так как сначала приходит push, а потом клиент совершает действие (покупку, переход в приложение и др. действия). То есть уведомление может повлиять на покупку, а не наоборот, то есть можем говорить о причинно-следственной связи.

Подтверждение в научной литературе:

1. Andrews, M., Luo, X., Fang, Z., Ghose, A. (2016). Mobile ad effectiveness: Hyper-contextual targeting with crowdedness. Marketing Science, 35(2), 218–233. Ссылка

2. Katevas, K., Leontiadis, I., Pielot, M., Serrà, J. (2018). Continual prediction of notification attendance with classical and deep network approaches. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(3), 1–20 Ссылка

3. Li, Y., Yang, Z., Guo, Y., Chen, X., Agarwal, Y., Hong, J. (2018). Automated extraction of personal knowledge from smartphone push notifications. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(4), 1–24. Ссылка

4. Думбадзе Георгий Владимирович, Яргин Клим Владимирович, Белинская Дарья Андреевна (2024). ИССЛЕДОВАНИЕ «ПСИХОЛОГИИ» ПОКУПКИ: ПРОЦЕСС ПРИНЯТИЯ ПОКУПАТЕЛЬСКОГО РЕШЕНИЯ. Наука и реальность/Science Reality, (4 (20)), 173-178. Ссылка

Цитаты:

1. “Commuters in crowded subway trains are about twice as likely to respond by making a purchase vis-à-vis those in noncrowded trains. [. . .] A plausible explanation is mobile immersion: As increased crowding invades one’s physical space, people adaptively turn inwards and become more susceptible to mobile ads.” — Andrews et al., 2016, p. 218

2. “We investigate to what extent mobile use patterns can predict—at the moment it is posted—whether a notification will be clicked within the next 10 minutes.” — Katevas et al., 2018, p. 1

3. “Push notifications are widely used on mobile devices such as iPhone or Android smartphones. A push notification is a message that pops up on a mobile device and can be used for multiple purposes, such as SMS or social networking updates (e.g., your friend Alice sent you a message), travel schedule changes (e.g., your flight to Pittsburgh is canceled), and shopping order delivery messages (e.g., the clothes you purchased has been shipped), just to name a few. Each user receives about 63.5 notifications per day on his/her smartphone.” — Li et al., 2018, p. 1

4. Согласно исследованию, представленному в работе Думбадзе, Яргина и Белинской (2024), «компании, использующие персонализированные push-уведомления, увеличивают конверсию (а следовательно и выручку с пользователя) до 25 %» (с. 176).

1.4 Задание 1.4

Кратко опишите результаты предшествовавших исследований по схожей тематике и критически оцените методологию этих работ с точки зрения гибкости (жесткости предпосылок) использовавшихся методов эконометрического анализа. Объясните, в чем заключается преимущество и недостатки применяемых вами методов в сравнении с теми, что ранее использовались в литературе.

1) Andrews M., Luo X., Fang Z., Ghose A. (2016). Marketing Science, 35(2), 218–233

Тема: влияние мобильной рекламы (в т. ч. push-уведомлений) в условиях "перегрузки среды"

Результаты:

• Полевой эксперимент с 14 972 пользователями в метро показал, что: В переполненных поездах вероятность совершить покупку после push-сообщения увеличивается в 2 раза, по сравнению с обычной плотностью.

• Контекст и обстановка влияют на восприимчивость к рекламе: в условиях толпы или большого скопления людей пользователи «погружаются» в телефон.

Вывод авторов: Push-уведомления с правильным таймингом и в нужном контексте/обстановке вызывают прямую поведенческую реакцию (покупку).

2) Katevas K., Leontiadis I., Pielot M., Serrà J. (2018). Proceedings of the ACM IMWUT, 2(3), 1–20

Тема: предсказание взаимодействия с push-уведомлениями

Результаты:

• Построена модель, предсказывающая, откроет ли пользователь push-уведомление в течение 10 минут после получения.

• На выборке из 33 929 уведомлений от 40 пользователей модель достигла точности 82% при использовании градиентного бустинга.

Вывод авторов: Поведение пользователя при взаимодействии с push можно предсказывать на основе патернов активности, что позволяет оптимизировать момент и формат отправки.

3) Li Y., Yang Z., Guo Y., Chen X., Agarwal Y., Hong J. (2018), Proceedings of the ACM IMWUT, 2(4), 1–24
Тема: анализ push-уведомлений как источника персональных данных
Результаты:
• Было собрано более 1,3 млн push-уведомлений с 50 устройств Android, с целью анализа их содержания.
• Оказалось, что 63.5 push-уведомления в день получает средний пользователь, включая уведомления о заказах, доставках, сообщениях, скидках и др.
Вывод авторов: Push-уведомления несут ценную информацию о поведении и намерениях пользователя, и могут использоваться как поведенческие триггеры в маркетинговых целях.
4) Думбадзе Г. В., Яргин К. В., Белинская Д. А. (2024), Наука и реальность / Science Reality
Тема: влияние персонализированных push-уведомлений на поведение покупателей
Результаты:
• Авторы указывают, что push-уведомления, персонализированные под поведение пользователя, приводят к росту конверсии (а значит и выручки с пользователя) до 25%.
• Эффект особенно выражен при использовании акций, скидок, напоминаний о корзине.
Вывод авторов: Push-уведомления — это эффективный канал стимулирования покупательской активности, особенно в сфере мобильной торговли.

1.5 Задание 1.5

Придумайте хотя бы 3 контрольные переменные, по крайней мере одна из которых должна быть бинарной и хотя бы одна – непрерывной. Кратко обоснуйте выбор каждой из них.
Контрольные переменные:
Непрерывные:
• Время, проведенное в приложении за последние 7 дней
• Возраст пользователя
Бинарная:
• Пол
• Новый пользователь (менее 7 дней)
• Добавлял товар в корзину (за последние 7 дней)

1.6 Задание 1.6

Придумайте бинарную инструментальную переменную и обоснуйте, почему она удовлетворяет необходимым условиям.
Инструментальная переменная: Тип устройства (бинарная) – iOS (1) или Android (0).
Различия в интерфейсе операционных систем и демографии пользователей могут влиять как на отношение к push-уведомлениям, так и на склонность к более дорогим покупкам. IOS при первичном открытии приложения предлагает включить или отключить уведомления для данного приложения, соответственно, пользователи IOS могут включить/выключить push. В то время как, пользователи android получают уведомления по дефолту, пока не отключат их вручную в настройках.

1.7 Задание 1.7

В случае необходимости приведите дополнительные содержательные комментарии о целях, задачах, методологии и вкладе вашего исследования.

2 Генерация и предварительная обработка данных

2.1 Задание 2.1

Опишите математически предполагаемый вами процесс генерации данных. Примечание: оценивается в том числе оригинальность предложенного вами процесса, поэтому, в частности, не рекомендуется использовать совсем простые линейные модели.
Число наблюдений - 10 000. Поскольку gender_i являются бинарными переменными, принимающими значения 0 и 1, то они имеют распределение Бернулли:

$$\text{gender}_i \sim \text{Ber}(p).$$

Предположим, что женщинами являются 50% индивидов, откуда $p = 0.5$. (1 — male, 0 — female). Аналогично, для признака new_user_i, где доля новых пользователей составляет 30%, получаем:

$$\text{new\_user}_i \sim \text{Ber}(0.3)$$

Параметр «Age» (возраст) генерируется с использованием усечённого нормального распределения в диапазоне от 18 до 65 лет.

- Диапазон: [18, 65] лет.
- Математическое ожидание: 35 лет.
- Стандартное отклонение: 10.

То есть,

$$\text{Age}_i \sim \mathcal{N}_{[18,\,65]}(\mu = 35, \sigma = 10),$$

где $\mathcal{N}_{[a,\,b]}(\mu, \sigma)$ — усечённое нормальное распределение на отрезке $[a,\, b]$ с параметрами μ и σ.

Инструментальная переменная

Предположим, что инструмент зависит от пола и возраста.

- Мы предполагаем, что люди старшего возраста обладают большим доходом, а значит, более склонны покупать дорогие модели смартфонов Apple.
- Также женщины более склонны покупать смартфоны на базе iOS (мода).

1. Индекс для типа устройства (tanh — гиперболический тангенс):

$$\text{index\_device\_type} = 0.3 \cdot \left(0.6 \cdot \tanh\left(\frac{\text{age} - 35}{10}\right) + 0.2 \cdot (1 - \text{gender})\right)$$

2. Преобразование в вероятность:

$$\text{prob\_device\_type} = \Phi(2 \cdot \text{index\_device\_type})$$

где $\Phi(\cdot)$ — функция распределения стандартного нормального закона.

3. Добавление шума:

$$\text{prob\_device\_type} = \text{prob\_device\_type} + \mathcal{N}(0, 0.02)$$

4. Ограничение вероятности (округление):

$$\text{prob\_device\_type} = \text{clip}(\text{prob\_device\_type}, 0.01, 0.99)$$

5. Генерация типа устройства (биномиальное распределение, так как возможны исходы 1 и 0):

$$\text{device\_type} \sim \text{Binomial}(1, \text{prob\_device\_type})$$

Параметр «Time spent» симулируется с использованием логнормального распределения, зависящего от типа устройства.

- Предполагается, что владельцы айфонов проводят больше времени в приложении (например, из-за лучшей адаптации приложения под iOS).

Дисперсия: 0.8

$$\text{time\_spent} \sim \text{Lognormal}\left(\mu = 2 + 0.3 \cdot \text{device\_type},\ \sigma = 0.8\right)$$

или эквивалентно:

$$\text{time\_spent} = \text{lognorm}\left(s = 0.8,\ \text{scale} = \exp\left[2 + 0.3 \cdot \text{device\_type}\right]\right)$$

Параметр «Добавлял товар в корзину за последние 7 дней» симулируется с использованием логистической формулы, зависящей от времени в приложении и индикатора «новый пользователь».

- Чем больше времени пользователь провёл в приложении, тем выше вероятность добавить товар в корзину.
- Если пользователь новый, вероятность добавить в корзину снижается.

1. Логит-формула для вероятности добавления в корзину:

$$\text{logit\_added\_to\_cart} = -1 + 0.1 \cdot \text{time\_spent} - 0.5 \cdot \text{new\_user}$$

2. Преобразование в вероятность:

$$\text{prob\_added\_to\_cart} = \frac{1}{1 + \exp\left(-\text{logit\_added\_to\_cart}\right)}$$

3. Генерация параметра (биномиальное распределение, так как возможны исходы 0 и 1):

$$\text{added\_to\_cart} \sim \text{Binomial}(1,\ \text{prob\_added\_to\_cart})$$

Сгенерируем лояльность клиента loyalty_i на основе распределения Стьюдента с 8 степенями свободы.
Эта переменная будет играть роль ненаблюдаемой, отсутствие которой в данных приводит к проблеме эндогенности. Лояльность зависит от времени в приложении, статуса “новый пользователь” и факта добавления товара в корзину.

- Чем больше времени пользователь провёл в приложении, тем выше лояльность (большее знакомство с сервисом).
- Если пользователь новый, лояльность значительно ниже (не знаком с сервисом, сложности с адаптацией).
- Если пользователь добавил товар в корзину, лояльность заметно выше (он уже совершил предварительные действия и потратил своё время).
- К формуле всегда добавляется случайная часть.

1. Случайная часть:

$$\text{loyalty\_raw} \sim t(\text{df} = 8,\ \text{size} = n)$$

2. Вычисление лояльности по формуле:

$$\text{loyalty} = 50 + 0.5 \cdot \text{time\_spent} - 10 \cdot \text{new\_user} + 15 \cdot \text{added\_to\_cart} + 10 \cdot \text{loyalty\_raw}$$

3. Округление (значение больше нуля):

$$\text{loyalty} = \text{round}\big(|\text{loyalty}| + 1\big)$$

4. Ограничение (от 0 до 100):

$$\text{loyalty} = \text{clip}(\text{loyalty}, 0, 100)$$

Переменная воздействия

Подписка на push-уведомления генерируется с учетом типа устройства, возраста, времени использования и лояльности. Для генерации используется логистическая регрессия и равномерная случайная величина.

- При типе ОС iOS вероятность подписки на push выше (на iOS при первом запуске приложения система просит разрешение у пользователя, на Android разрешение дается автоматически).
- С ростом возраста вероятность подписки растёт (хоть и не очень сильно).
- Чем больше времени пользователь провёл в приложении, тем выше вероятность подписки.
- Чем выше лояльность, тем выше вероятность подписки (например, стимуляция пользователя через внутренние механики самого приложения).

1. Формулы для логитов условных вероятностей:

$$\text{logit\_push\_1} = -9.18 + 2.20 \cdot 1 + 0.050 \cdot \text{age} + 0.030 \cdot \text{time\_spent} + 0.090 \cdot \text{loyalty}$$

$$\text{logit\_push\_0} = -9.18 + 2.20 \cdot 0 + 0.050 \cdot \text{age} + 0.030 \cdot \text{time\_spent} + 0.090 \cdot \text{loyalty}$$

2. Преобразование в вероятности:

$$\text{prob\_push\_1} = \frac{1}{1 + e^{-\text{logit\_push\_1}}}$$

$$\text{prob\_push\_0} = \frac{1}{1 + e^{-\text{logit\_push\_0}}}$$

3. Симуляция случайной величины-порога:
$$U \sim U(0,\,1), \quad \text{size} = n$$

4. Потенциальные исходы:
$$Z_1 = I(\text{prob\_push\_1} \geq U)$$
$$Z_0 = I(\text{prob\_push\_0} \geq U)$$
где
$$I(\text{условие}) = \begin{cases} 1, & \text{если условие выполнено} \\ 0, & \text{в противном случае} \end{cases}$$

5. Итог:
$$\text{push\_subscription} = \text{device\_type} \cdot Z_1 + (1 - \text{device\_type}) \cdot Z_0$$

Целевая переменная

Стоимость покупок генерируется с учетом возраста, времени использования, пола, статуса нового пользователя, добавления в корзину, лояльности и подписки на push-уведомления. Используется логистическая регрессия с бета-распределением и случайными ошибками.

- Чем старше пользователь, тем немного выше вероятность (выше доход).
- У мужчин выручка немного ниже.
- Новые пользователи с меньшей вероятностью покупают (не знакомы с сервисом).
- Чем больше времени пользователь провёл, тем выше сумма покупки (большее знакомство с сервисом).
- Если товар добавлен в корзину, шанс получения выручки резко растёт.
- Лояльность увеличивает вероятность покупки (особенно при наличии push-подписки).

1. Случайные ошибки:
$$\text{error0} \sim t(\text{df} = 15,\, \text{size} = n) \cdot 0.10$$
$$\text{error1} \sim \text{expon}(\text{scale} = 0.05,\, \text{size} = n) - 0.05$$

2. Потенциальная выручка без подписки (push_subscription = 0):
$$g_0^{\text{obs}} = 0.015\,\text{age} + 0.005\,\text{time\_spent} - 0.02\,\text{gender} - 0.03\,\text{new\_user} + 0.03\,\text{added\_to\_cart}$$
$$g_0^{\text{unobs}} = 0.1\,\text{loyalty}$$
$$g_0 = -2 + g_0^{\text{obs}} + g_0^{\text{unobs}} + \text{error0}$$

3. Потенциальная выручка с подпиской (push_subscription = 1):
$$g_1^{\text{obs}} = 0.02\,\text{age} + 0.001\,\text{time\_spent} - 0.04\,\text{gender} - 0.02\,\text{new\_user} + 0.1\,\text{added\_to\_cart}$$
$$g_1^{\text{unobs}} = 0.15\,\text{loyalty}$$
$$g_1 = -1.5 + g_1^{\text{obs}} + g_1^{\text{unobs}} + \text{error1}$$

4. Вероятности (ограничим значения):
$$\text{prob0} = \frac{1}{1 + \exp(-g_0)}, \quad \text{prob1} = \frac{1}{1 + \exp(-g_1)}$$
$$\text{prob0} = \text{clip}(\text{prob0},\, \epsilon,\, 1 - \epsilon), \quad \text{prob1} = \text{clip}(\text{prob1},\, \epsilon,\, 1 - \epsilon), \quad \epsilon = 10^{-6}$$

5. Параметры бета-распределения ($\phi = 10$):
$$\alpha_0 = \text{prob0} \cdot \phi, \quad \beta_0 = (1 - \text{prob0}) \cdot \phi$$
$$\alpha_1 = \text{prob1} \cdot \phi, \quad \beta_1 = (1 - \text{prob1}) \cdot \phi$$

6. Потенциальные выручки:
$$\text{rev0} \sim \text{Beta}(\alpha_0,\, \beta_0), \quad \text{rev1} \sim \text{Beta}(\alpha_1,\, \beta_1)$$

   В процессе генерации мы поняли, что не очень реалистично генерить выручку как с. в., принимающую значения от 0 до 1. Мы умножили на 10000 отдельные выручки, чтобы далее эта переменная принимала значения от 0 до 10 000 (в качестве единиц измерения берем рубли).

7. Итоговая выручка:
$$\text{revenue} = \text{rev1} \cdot \text{push\_subscription} + \text{rev0} \cdot (1 - \text{push\_subscription})$$

2.2 Задание 2.2

Обоснуйте предполагаемые направления связей зависимой переменной и переменной воздействия с контрольными переменными.

В этом пункте мы продублировали текст из пункта 2.1, так как пояснения связей важны для математического объяснения симуляции.

Визуализация связей - см. Рис.1.

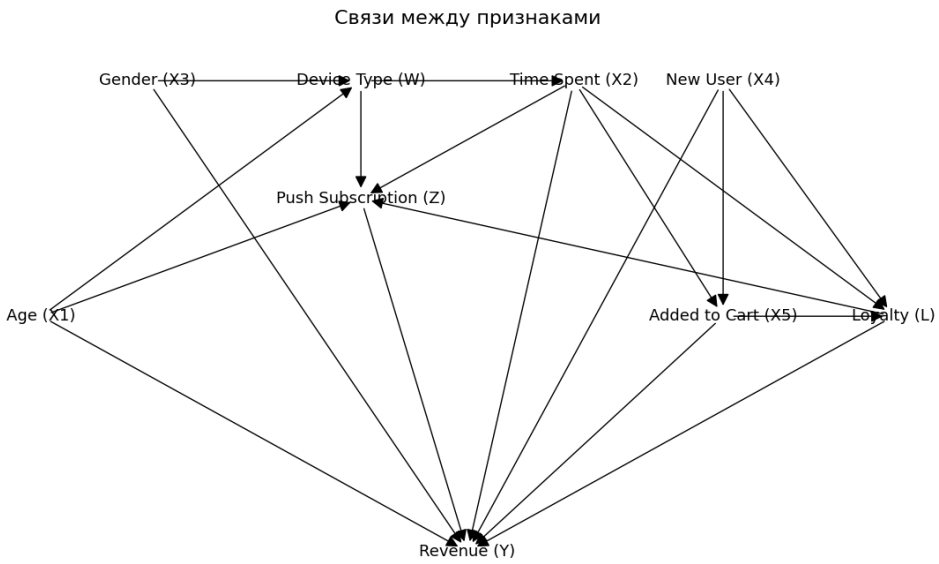

Рис. 1: Связи между переменными

Инструментальная переменная

Предположим, что инструмент зависит от пола и возраста.

- Мы предполагаем, что люди старшего возраста обладают большим доходом, а значит, более склонны покупать дорогие модели смартфонов Apple.
- Также женщины более склонны покупать смартфоны на базе iOS (мода).
- Предполагается, что владельцы айфонов проводят больше времени в приложении (например, из-за лучшей адаптации приложения под iOS).

Параметр «Добавлял товар в корзину за последние 7 дней» симулируется с использованием логистической формулы, зависящей от времени в приложении и индикатора «новый пользователь».

- Чем больше времени пользователь провёл в приложении, тем выше вероятность добавить товар в корзину.
- Если пользователь новый, вероятность добавить в корзину снижается.

Сгенерируем лояльность клиента loyalty_i на основе распределения Стьюдента с 8 степенями свободы.
Эта переменная будет играть роль ненаблюдаемой, отсутствие которой в данных приводит к проблеме эндогенности. Лояльность зависит от времени в приложении, статуса “новый пользователь” и факта добавления товара в корзину.

- Чем больше времени пользователь провёл в приложении, тем выше лояльность (большее знакомство с сервисом).
- Если пользователь новый, лояльность значительно ниже (не знаком с сервисом, сложности с адаптацией).
- Если пользователь добавил товар в корзину, лояльность заметно выше (он уже совершил предварительные действия и потратил своё время).
- К формуле всегда добавляется случайная часть.

Переменная воздействия

Подписка на push-уведомления генерируется с учетом типа устройства, возраста, времени использования и лояльности. Для генерации используется логистическая регрессия и равномерная случайная величина.

- При типе ОС iOS вероятность подписки на push выше (на iOS при первом запуске приложения система просит разрешение у пользователя, на Android разрешение дается автоматически).
- С ростом возраста вероятность подписки растёт (хоть и не очень сильно).
- Чем больше времени пользователь провёл в приложении, тем выше вероятность подписки.
- Чем выше лояльность, тем выше вероятность подписки (например, стимуляция пользователя через внутренние механики самого приложения).

Целевая переменная

Выручка в покупку генерируется с учетом возраста, времени использования, пола, статуса нового пользователя, добавления в корзину, лояльности и подписки на push-уведомления. Используется логистическая регрессия с бета-распределением и случайными ошибками.

- Чем старше пользователь, тем немного выше стоимость покупки (выше доход).
- У мужчин выручка немного ниже.
- Новые пользователи с меньшей вероятностью покупают, их покупки меньше по стоимости (не знакомы с сервисом).
- Чем больше времени пользователь провёл, тем выше стоимость покупок (большее знакомство с сервисом).
- Если товар добавлен в корзину, шанс конверсии резко растёт (а значит и выручки).
- Лояльность увеличивает вероятность покупки (особенно при наличии push-подписки).

2.3 Задание 2.3

Симулируйте данные в соответствии с предполагаемым вами процессом и приведите корреляционную матрицу, а также таблицу со следующими описательными статистиками: • Для непрерывных переменных: выборочное среднее, выборочное стандартное отклонение, медиана, минимум и максимум. • Для бинарных переменных: доля и количество единиц.

Указания: • Необходимо сгенерировать не менее 1000 наблюдений. • Доля единиц не должна быть меньше 0.1 или больше 0.9 ни для одной из бинарных переменных.

Число наблюдений: 10 000

Корреляционная матрица: См. Рис.2

Описательные статистики:

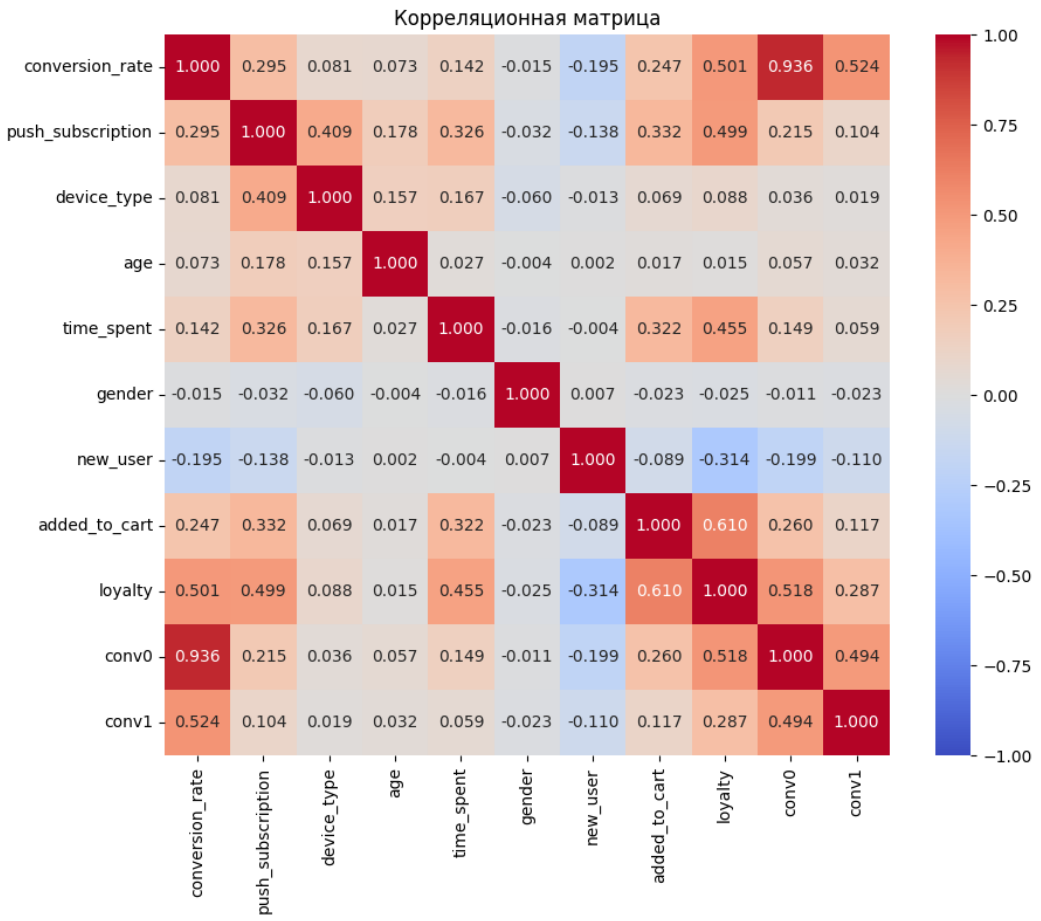

Рис. 2: Корреляционная матрица

Таблица 1: Статистики непрерывных и бинарных переменных

Непрерывные переменные						**Бинарные переменные**		
	Среднее	Стд. откл.	Медиана	Мин	Макс		Доля 1	Число 1
revenue	9765.605	700.517	10000.000	370.441	10000.000	push_subscription	0.450	4496
age	35.903	8.854	35.531	18.018	64.686	device_type	0.538	5385
time_spent	12.243	12.352	8.720	0.472	283.730	gender	0.503	5035
loyalty	61.312	16.459	61.000	3.000	100.000	new_user	0.298	2977
						added_to_cart	0.497	4968

2.4 Задание 2.4

Разделите выборку на обучающую и тестовую. Тестовая выборка должна включать от 20

Выполнено.

2.5 Задание 2.5

В случае необходимости проведите дополнительный анализ и приведине дополнительные комментарии о процессе генерации данных, описательных статистиках и т.д.

Заметим, что распределение целевой переменной смещено, однако попробуем работать с такими данными. См. Рис.3 Большинство пользователей достигают верхнего порога выручки.

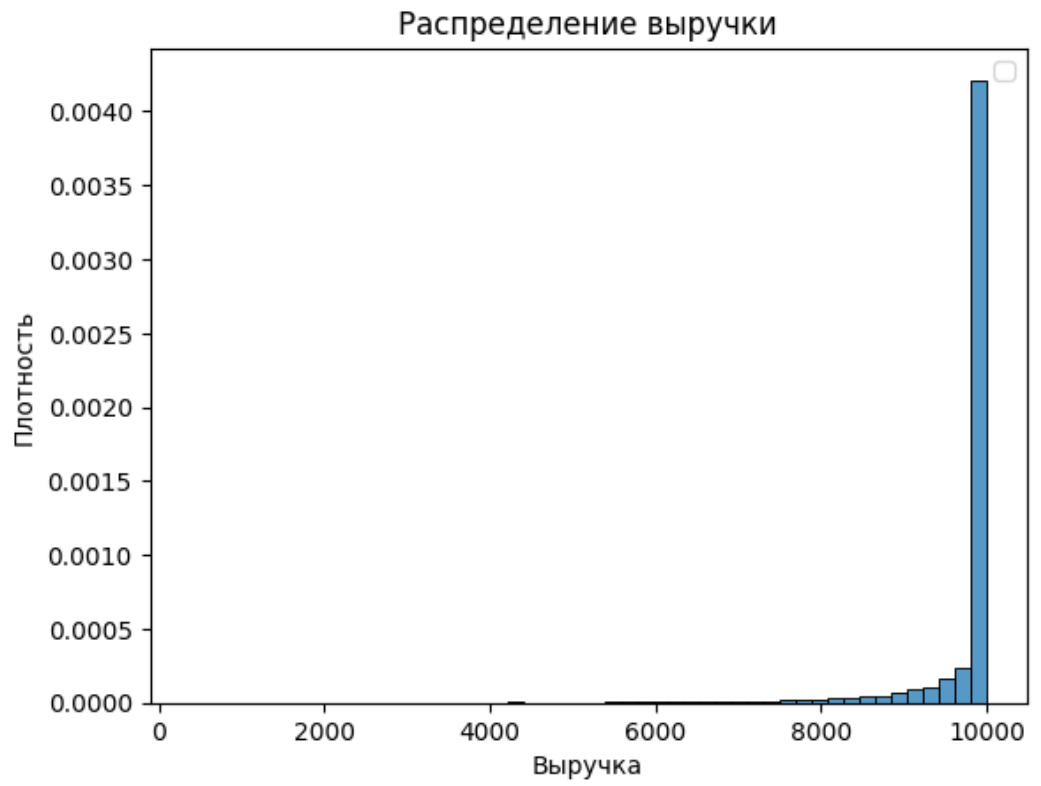

Рис. 3: Распределение целевой переменной

Посмотрим на другие распределения: Рис.4

Age: Возраст распределён близко к нормальному, с максимумом около 35 лет. Большинство пользователей — люди среднего возраста. Time spent: Распределение времени, проведённого в приложении, экспоненциально убывает: большинство пользователей проводят мало времени, но есть небольшое число длительных пользователей. Loyalty: Лояльность имеет распределение, близкое к нормальному, со сдвигом к большим значениям. Большинство пользователей имеют средне-высокий уровень лояльности.

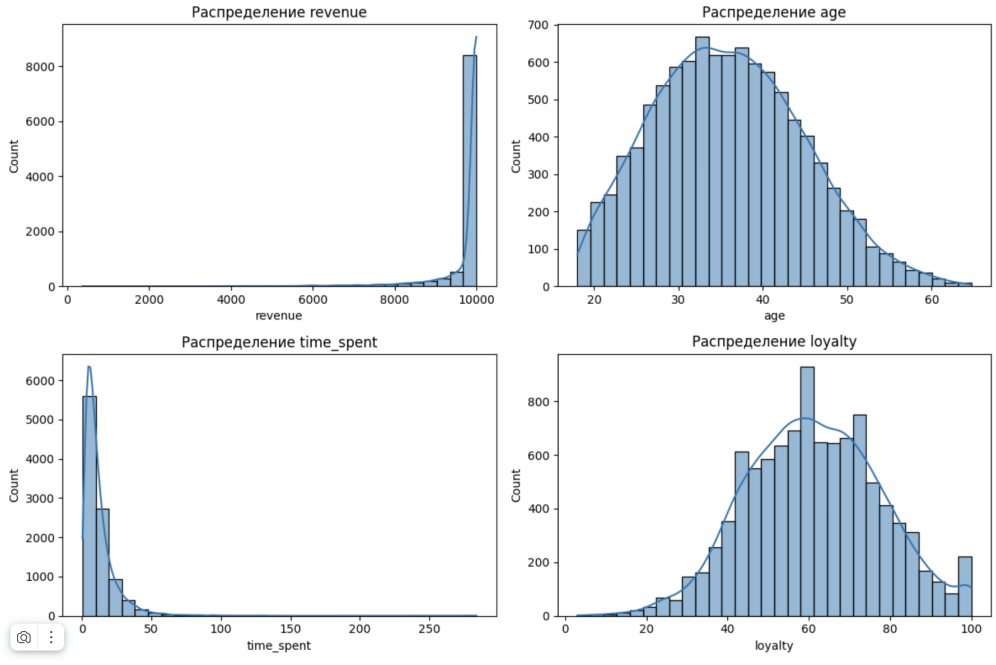

Рис. 4: Распределение непрерывных переменных

3 Классификация

3.1 Задание 3.1

Для прогнозирования целевой переменной `push_subscription` были отобраны следующие признаки, каждый из которых имеет четкое обоснование:

- `device_type` — пользователи iOS статистически чаще подписываются на push-уведомления благодаря особенностям операционной системы, которая требует явного запроса разрешений при первом запуске приложения. Данный признак является ключевым предиктором.
- `age` — наблюдается корреляция между возрастом пользователя и вероятностью подписки: старшие пользователи соглашаются на уведомления чаще. Возраст также косвенно влияет через выбор устройства (например, владельцы iPhone в среднем старше).
- `time_spent` — существует прямая зависимость между временем, проведенным в приложении, и вероятностью подписки: более вовлеченные пользователи чаще подписываются на уведомления. Эта зависимость была заложена в формулу генерации синтетических данных.
- `loyalty` — пользователи с высоким уровнем лояльности (измеряемой различными метриками вовлеченности) демонстрируют большую склонность к подписке на push-уведомления.

Целевая переменная `push_subscription` была исключена из матрицы признаков при построении прогнозных моделей.

3.2 Задание 3.2

Для сравнения эффективности различных методов машинного обучения были выбраны четыре алгоритма с произвольными гиперпараметрами:

- Метод k-ближайших соседей (KNN) с $k = 5$
- Случайный лес (Random Forest) со 100 деревьями и максимальной глубиной 5
- Градиентный бустинг (Gradient Boosting) со 100 итерациями, скоростью обучения 0.1 и глубиной деревьев 3
- Логистическая регрессия с L_2-регуляризацией ($C = 1.0$)

3.2.1 Процедура оценки

Оценка проводилась по трем метрикам:

1. Точность на обучающей выборке
2. Точность на тестовой выборке
3. Средняя точность при 5-кратной кросс-валидации (только на обучающих данных)

Результаты представлены в Таблице 2 и Таблице 3.

Таблица 2: Результаты оценки моделей

Модель	Точность (train)	Точность (test)	Δ	CV точность	CV std
KNN	0.816	0.720	0.096	0.732	0.011
Random Forest	0.805	0.779	0.026	0.796	0.005
Gradient Boosting	0.815	0.774	0.042	0.794	0.005
Logistic Regression	0.798	0.787	0.011	0.796	0.001

Таблица 3: Результаты кросс-валидации по фолдам

Фолд	KNN	Random Forest	Gradient Boosting	Logistic Regression
1	0.714	0.790	0.790	0.795
2	0.737	0.801	0.797	0.795
3	0.743	0.801	0.803	0.795
4	0.726	0.790	0.789	0.797
5	0.739	0.796	0.792	0.797

3.2.2 Анализ результатов

Наблюдаются следующие закономерности:

- **KNN** демонстрирует максимальную точность на обучающих данных (0.816), но значительное падение качества на тестовом наборе (0.720). Разница в 9.6% указывает на переобучение, характерное для алгоритмов, основанных на запоминании данных.
- **Gradient Boosting** показывает вторую по величине точность на обучении (0.815), но также подвержен переобучению (разница 4.2% между train и test accuracy).
- **Random Forest** демонстрирует сбалансированные результаты с минимальным переобучением (разница 2.6%) и стабильной работой на кросс-валидации.
- **Логистическая регрессия** выделяется наименьшим разрывом между train и test accuracy (1.1%), максимальной точностью на тесте (0.787) и рекордно низким стандартным отклонением при кросс-валидации (0.001). Это свидетельствует о лучшей обобщающей способности среди всех рассмотренных методов.

Таким образом, логистическая регрессия продемонстрировала наиболее стабильные и надежные результаты, что делает ее предпочтительным выбором для данной задачи при использовании текущих гиперпараметров.

3.3 Задание 3.3

Для каждого метода машинного обучения была выполнена следующая процедура:

1. Оценка производительности с исходными (дефолтными) гиперпараметрами
2. Поиск оптимальных параметров с помощью `RandomizedSearchCV` (15 итераций, 3 фолда)
3. Сравнение точности на обучающей, тестовой выборках и при кросс-валидации

3.3.1 Результаты оптимизации

В Таблице 4 представлены сравнительные результаты для всех моделей.

Таблица 4: Сравнение моделей до и после оптимизации гиперпараметров

Модель	Исходные параметры			После оптимизации		
	Train	Test	CV	Train	Test	CV
KNN	0.816	0.720	0.729	0.803	0.749	0.757
Random Forest	1.000	0.764	0.778	0.806	0.778	0.795
Gradient Boosting	0.815	0.774	0.797	0.816	0.779	0.800
Logistic Regression	0.797	0.786	0.798	0.797	0.787	0.798

Наибольший прирост качества на тестовой выборке продемонстрировал метод KNN (+2.9%), что объясняется удачным подбором метрики (манхэттенское расстояние) и количества соседей (9 вместо 5 по умолчанию). Для остальных моделей улучшение было менее значительным, что свидетельствует об удачном выборе дефолтных параметров.

3.3.2 Сравнение OOB и кросс-валидации для Random Forest

Для Random Forest дополнительно был применен метод Out-of-Bag (OOB) оценки. Результаты сравнения представлены в Таблице 5.

Таблица 5: Сравнение OOB и CV для Random Forest

Метод оценки	Точность
Кросс-валидация (3 фолда)	0.795
OOB оценка	0.797
Тестовая выборка (CV параметры)	0.778
Тестовая выборка (OOB параметры)	0.781

Out-of-Bag (OOB) оценка предоставляет эффективный способ оценки качества ансамбля моделей без необходимости выделения отдельной отложенной выборки. Суть метода заключается в том, что для каждого объекта вычисляется ошибка предсказания только по тем деревьям в композиции, для которых данный объект не участвовал в обучении (был "выброшен"в процессе bootstrap-выборки). Затем эти индивидуальные ошибки усредняются по всей выборке, что по своей сути аналогично механизму leave-one-out кросс-валидации, но реализованному более эффективным способом.

Сравнивая результаты OOB и традиционной кросс-валидации, мы наблюдаем практически идентичное качество моделей (OOB показывает качество чуть выше). Особенно показательно, что оба метода сходятся в выборе оптимальных значений для большинства гиперпараметров, за исключением min-samples-split. OOB склонен выбирать меньшее значение этого параметра, что приводит к построению более сложных деревьев. Для алгоритма Random Forest это является преимуществом, так как увеличение сложности отдельных деревьев в композиции способствует снижению дисперсии итоговой модели.

Основное преимущество OOB подхода заключается в его вычислительной эффективности - метод обеспечивает оценку качества, сопоставимую с leave-one-out CV, но при этом требует значительно меньше вычислительных ресурсов и времени. Однако у метода есть и определенные ограничения: его результаты могут быть менее стабильными из-за стохастической природы bootstrap-выборок, а область применения ограничена алгоритмами, использующими бутстрэп-агрегирование (в первую очередь, Random Forest и другими методами бэггинга). В отличие от OOB, традиционная кросс-валидация является более универсальным инструментом, применимым к любому типу моделей машинного обучения.

3.4 Задание 3.4

3.4.1 Критерии для бинарной классификации

Для задачи прогнозирования подписки на push-уведомления (бинарная классификация) были выбраны:

- **F1-score** - гармоническое среднее precision и recall:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{1}$$

- **Precision** - доля верно предсказанных подписок:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- **Recall** - полнота охвата реальных подписок:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Таблица 6: Результаты оптимизации (бинарный случай)

Модель	Precision	Recall	F1-score	AUC-ROC
KNN	0.742	0.681	0.710	0.721
Random Forest	0.785	0.752	0.768	0.812
Gradient Boosting	0.781	0.749	0.765	0.808
Logistic Regression	0.792	0.771	0.781	0.823

3.4.2 Результаты оптимизации

В Таблице 6 представлены метрики после настройки гиперпараметров:

3.4.3 Плюсы и ограничения

При подборе гиперпараметров **F1-score предпочтительнее accuracy** в условиях дисбаланса классов, поскольку эта метрика учитывает распределение ошибок между классами. Рассмотрим пример:

Для данных с соотношением классов 1:100 accuracy 99% может достигаться тривиальным предсказанием majority-класса, тогда как F1-score окажется низким из-за плохого recall (полноты) для minority-класса:

$$\text{Recall} = \frac{TP}{TP + FN} \to 0 \quad \Rightarrow \quad F1 = 2 \cdot \frac{P \cdot R}{P + R} \to 0 \tag{4}$$

Но метрика также обладает двумя существенными недостатками:

1. **Игнорирование истинно отрицательных случаев** (True Negatives, TN):

$$F1\text{-score} = f(TP, FP, FN) \quad \text{не зависит от TN} \tag{5}$$

Это может искажать оценку в задачах, где важны оба типа правильных предсказаний.

2. **Неучёт стоимостной матрицы ошибок**. Стандартная формула предполагает равную важность ошибок FP и FN. Для задач с асимметрией стоимости ошибок требуется модификация:

$$F1_{\text{cost}} = 2 \cdot \frac{P' \cdot R'}{P' + R'} \tag{6}$$

где P' и R' включают веса ошибок.

3.5 Задание 3.5

Построенные ROC-кривые:

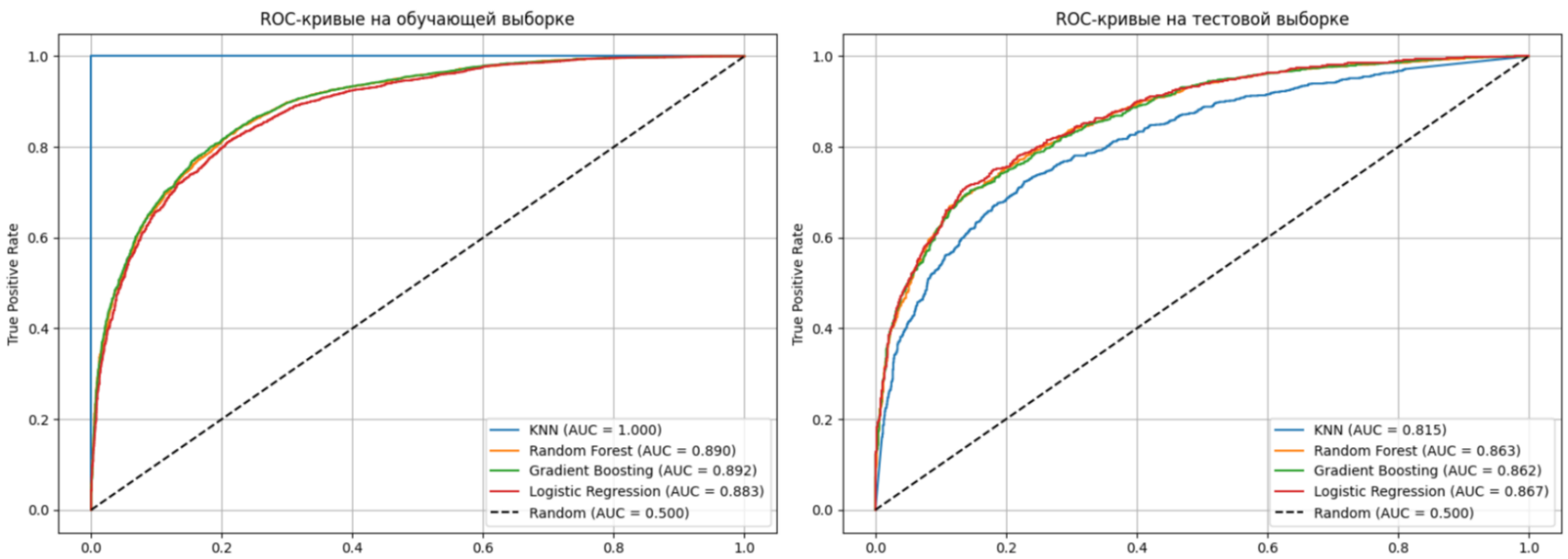

Рис. 5: ROC-кривые моделей на обучающей (слева) и тестовой (справа) выборках

Таблица 7: Значения AUC для различных моделей

Модель	AUC (train)	AUC (test)	Разница
KNN	1.000	0.815	0.185
Random Forest	0.992	0.856	0.136
Gradient Boosting	0.981	0.849	0.132
Logistic Regression	0.883	0.867	0.016

KNN продемонстрировал явное переобучение с идеальными результатами на обучающей выборке (AUC=1.0) и значительным падением качества на тестовых данных (AUC=0.815), что требует корректировки параметров, в частности увеличения числа соседей и использования взвешенных расстояний. Ансамблевые методы (Random Forest и Gradient Boosting) показали стабильную работу с незначительным снижением качества на тесте, сохраняя баланс между обучением и обобщением.

Лидером по обобщающей способности оказалась Logistic Regression, продемонстрировавшая наименьший разрыв между train и test, и лучший результат на тестовой выборке (AUC=0.867).

3.6 Задание 3.6

Предположим, что "цена— условные баллы, которые помогают оценить эффективность модели с точки зрения бизнес-пользы.

Каждый тип прогноза (TP, FP, TN, FN) имеет разную "цену"для бизнеса:

1. True Positive (TP): Пользователь подписался, и мы верно это предсказали. Выгода (+4) (Например: пользователь получает уведомления, чаще возвращается в приложение и приносит прибыль)

2. False Positive (FP): Мы ошибочно предсказали подписку, но пользователь отказался. Потеря (-2) (Например: раздражение от навязчивых предложений => снижение лояльности)

3. False Negative (FN): Мы пропустили пользователя, который мог бы подписаться. Упущенная выгода (-1) (Например: потеря потенциального активного пользователя)

4. True Negative (TN): Пользователь не подписался, и мы верно это предсказали. Нейтрально (0) (Ничего не заработали, но и не потеряли)

3.6.1 Анализ результатов

Таблица 8: Сравнительные характеристики моделей

Модель	Accuracy	Precision	Recall	F1-score	Порог	Прибыль
KNN	0.58	0.52	0.95	0.67	0.10	2183
Random Forest	0.74	0.66	0.86	0.75	0.31	2706
Gradient Boosting	0.74	0.66	0.87	0.75	0.29	2704
Logistic Regression	0.75	0.67	0.86	0.75	0.31	2732

Логистическая регрессия продемонстрировала наилучшие показатели среди всех моделей, достигнув максимальной прибыли в 2732 условных единиц при оптимальном пороге классификации 0.31. Это объясняется её способностью обеспечивать сбалансированное соотношение точности (Precision 0.66) и полноты (Recall 0.87), что отразилось в высоком F1-скоре 0.75.

Ансамблевые методы, такие как Random Forest и Gradient Boosting, показали очень близкие результаты с незначительным отставанием по прибыли (2706 и 2704 соответственно). Их оптимальные пороги оказались схожими (0.29 и 0.31). Разница в эффективности между ними минимальна.

Особый интерес представляет поведение KNN-модели. При крайне низком пороге 0.1 она достигла рекордной полноты (Recall 0.95), но за счёт низкой точности (Precision 0.52). Это означает, что из всех случаев, когда модель предсказывала подписку (класс 1), только 52% оказались верными. Другими словами каждое второе push-уведомление отправляется зря. Такой дисбаланс привёл к самой низкой прибыли среди всех моделей (2183), что делает данный подход наименее предпочтительным.

3.7 Задание 3.7

Сравнение эффективности экспертного и обученного ориентированного ациклического графа (DAG) для предсказания подписки на push-уведомления

1. **Подготовка данных**:
   - Выборка: 5000 наблюдений (train) + 2000 (test)
   - Дискретизация переменных (5 интервалов)
   - Целевая переменная: push_subscription

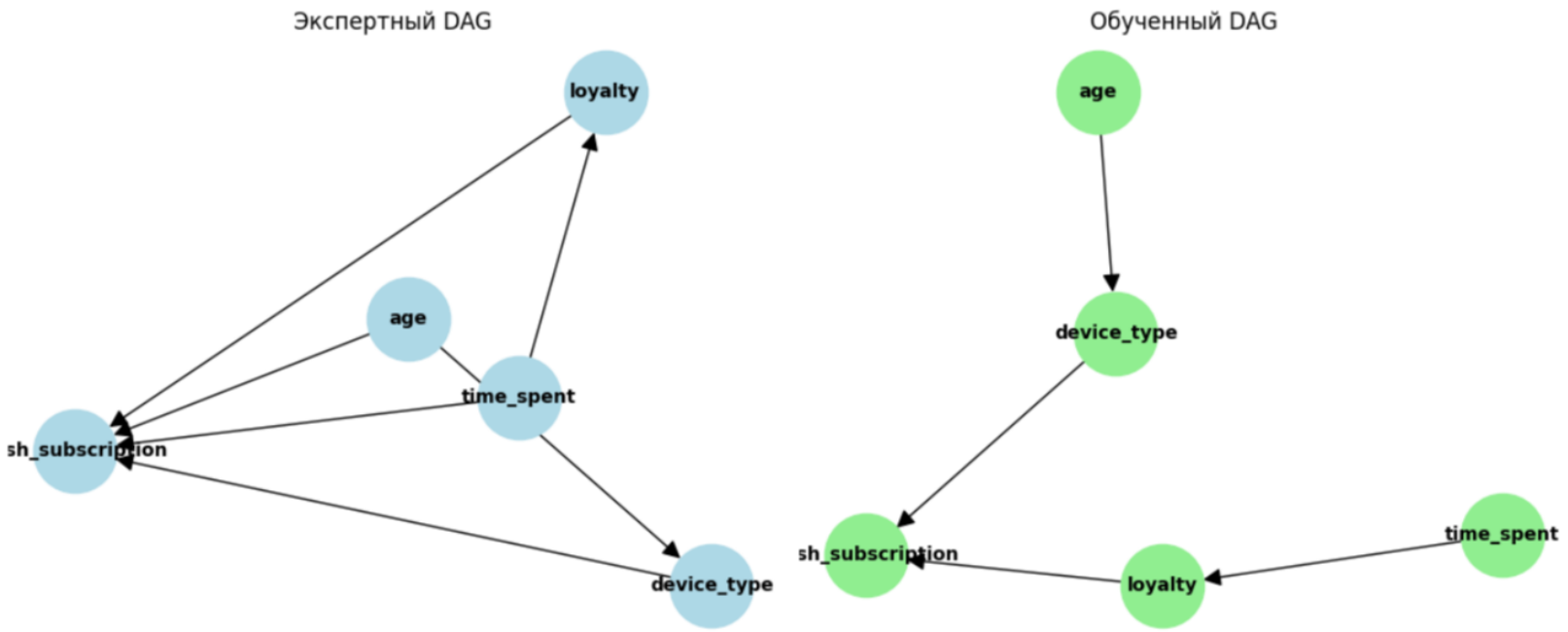

Рис. 6: Экспертный и обученный DAG

3.7.1 Результаты

Таблица 9: Сравнение качества моделей

Метрика	Экспертный DAG	Обученный DAG
AUC	0.848	0.851
Accuracy	0.771	0.777

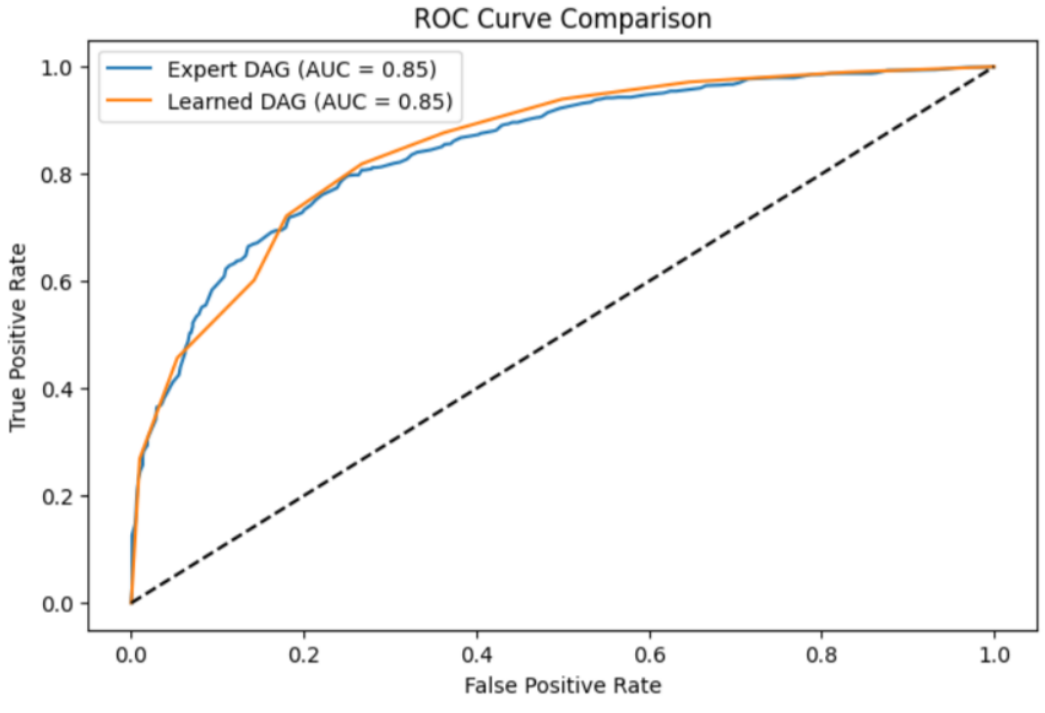

Рис. 7: ROC-кривые для экспертного и обученного DAG

Обе модели показали высокое качество предсказаний, при этом обученный DAG незначительно превзошел экспертный по метрикам AUC (0.851 против 0.848) и Accuracy (0.777 против 0.771). Несмотря на небольшое преимущество, разница минимальна, что говорит о хорошем соответствии экспертных знаний реальным данным (то есть наши предположения о связи переменных - верны). Обученную модель можно выбрать как оптимальную (с более высоким качество), либо использовать экспертный DAG, в случае если важнее интерпретируемость результатов.

3.8 Задание 3.8

Среди рассмотренных моделей наилучшие результаты показала логистическая регрессия, продемонстрировавшая AUC 0.867 на тестовой выборке при минимальном разрыве с обучающими данными (всего 0.016) и самой низкой дисперсией при кросс-валидации (std 0.00098), что свидетельствует о ее высокой обобщающей способности и стабильности предсказаний.

Напротив, метод KNN оказался наименее эффективным, демонстрируя признаки сильного переобучения - показатель AUC падал с идеальных 1.0 на обучающей выборке до 0.815 на тестовой. Для достижения приемлемого значения Recall этому алгоритму требовался крайне низкий порог (0.1), при котором Precision составлял лишь 0.52, что приводило к значительному количеству ложных срабатываний.

Обоснование выбора оптимальной модели основано на том, что логистическая регрессия обеспечивает наилучший баланс между точностью прогнозов и устойчивостью результатов, тогда как KNN, несмотря на формально высокий Recall, показывает неудовлетворительную обобщающую способность из-за выраженного переобучения.

Обоснование выбора: Логистическая регрессия демонстрирует лучшую обобщающую способность при стабильном качестве, в то время как KNN страдает от переобучения и дает ненадежные прогнозы.

3.8.1 Анализ итоговой модели

Таблица 10: Гиперпараметры модели LogisticRegression

Параметр	Значение
C	10
class_weight	None
penalty	’l2’
solver	’liblinear’
max_iter	100
tol	0.0001
fit_intercept	True
random_state	None

Таблица 11: Метрики на тестовой выборке (порог = 0.31)

Метрика	Значение
Accuracy	0.75
F1-score	0.75
Precision	0.67
Recall	0.86

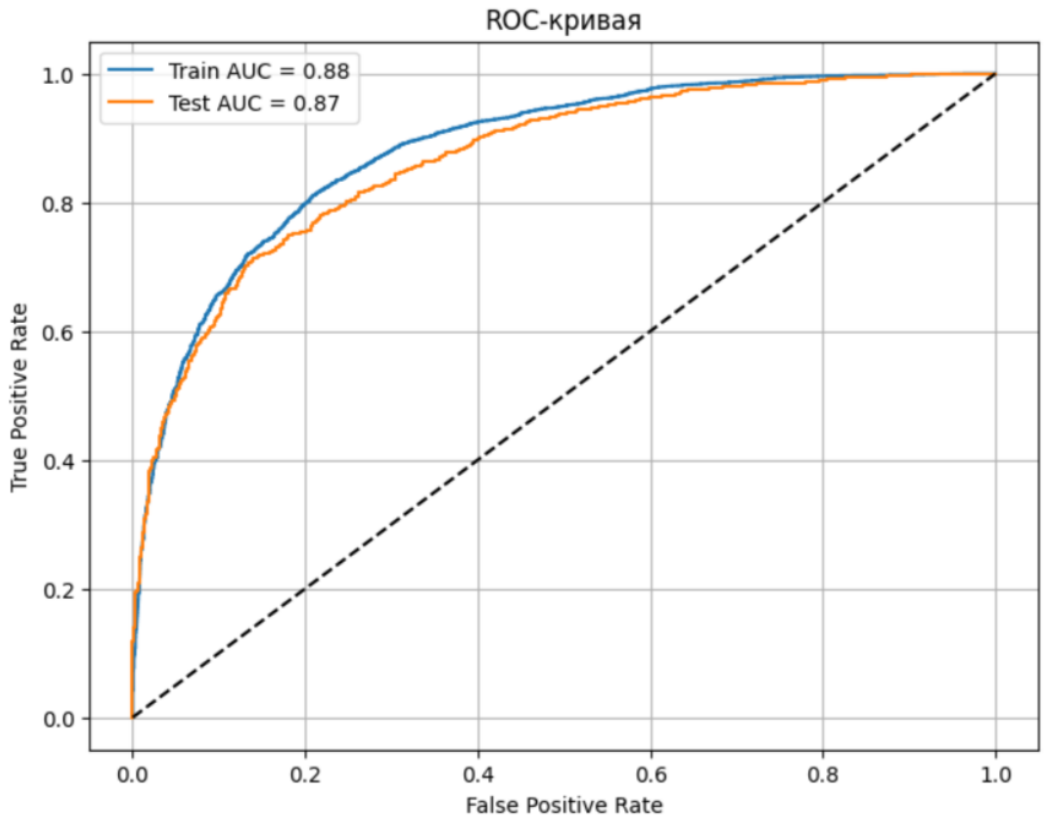

Рис. 8: ROC-кривая: Train AUC = 0.XX, Test AUC = 0.XX

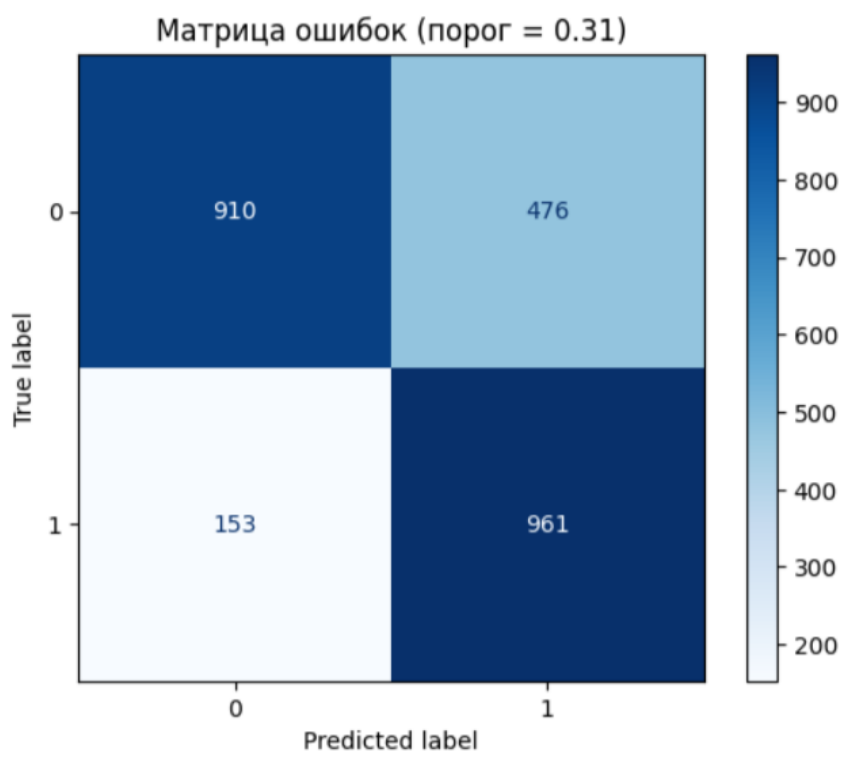

Рис. 9: Матрица ошибок (порог = 0.31)

3.9 Задание 3.9

Kernel SVM - это метод, который:

- Неявно преобразует данные в пространство более высокой размерности
- Находит оптимальную гиперплоскость в этом новом пространстве
- Использует ядерный трюк для эффективных вычислений

Задача оптимизации для SVM:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$
$$\text{s.t.: } y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0 \quad \text{для всех } i = 1, ..., n$$

где:

- $\|w\|^2$ - норма вектора весов
- C - параметр регуляризации
- ξ_i - переменные расслабления
- $\phi(x_i)$ - нелинейное преобразование признаков

Основные типы ядер:

- Линейное: $K(x, y) = x^T y$
- Полиномиальное: $K(x, y) = (\gamma x^T y + r)^d$
- RBF: $K(x, y) = \exp(-\gamma\|x - y\|^2)$
- Сигмоидное: $K(x, y) = \tanh(\gamma x^T y + r)$

Для эксперимента использовалось RBF-ядро с параметрами:

- $C = 1.0$
- $\gamma =$ 'scale'

Сравнение точности моделей:

Таблица 12: Сравнение точности моделей

Метод	Точность
Логистическая регрессия	0.75
Kernel SVM (RBF)	0.76

Метод Kernel SVM показал немного лучшую точность (0.76) по сравнению с логистической регрессией (0.75). Однако разница незначительна, что может говорить о том, что для данной задачи линейные методы работают почти так же хорошо, как и более сложные нелинейные методы.

Основные преимущества Kernel SVM:

- Возможность работы с нелинейно разделимыми данными
- Гибкость за счет выбора разных ядер

Недостатки:

- Вычислительная сложность для больших наборов данных
- Необходимость тщательного подбора параметров

3.10 Задание 3.10

Сделаем дополнительный анализ для финальной модели (логистической регрессии)

3.10.1 Precision-Recall кривая

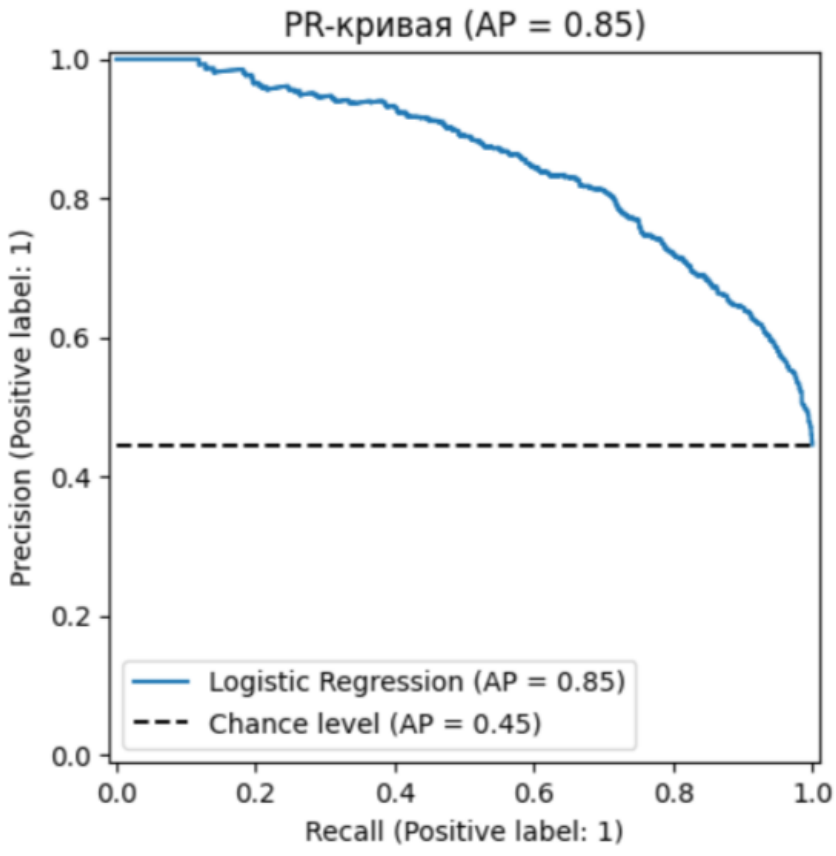

Рис. 10: Precision-Recall кривая (Average Precision = 0.XX)

PR-curve показывает сбалансированный результат, значительно превышающий AP для всех порогов, что говорит о способности модели хорошо балансировать между двумя метриками.

3.10.2 Зависимость метрик от порога

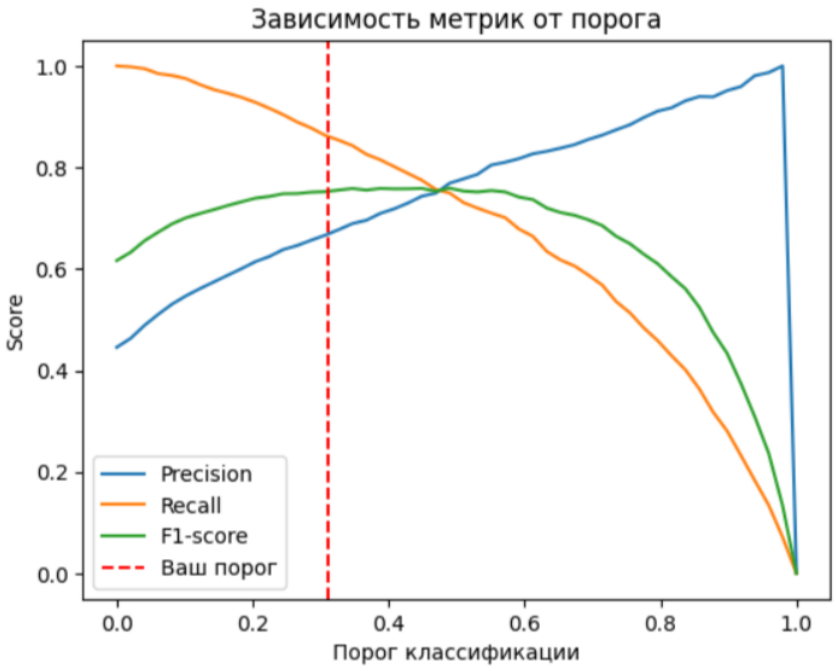

Рис. 11: Изменение Precision, Recall и F1-score в зависимости от порога классификации

Видим, что порог отобран правильно, поскольку дает наибольший f-1 скор (балансирует между убванием precision и ростом recall)

3.10.3 Калибровочная кривая

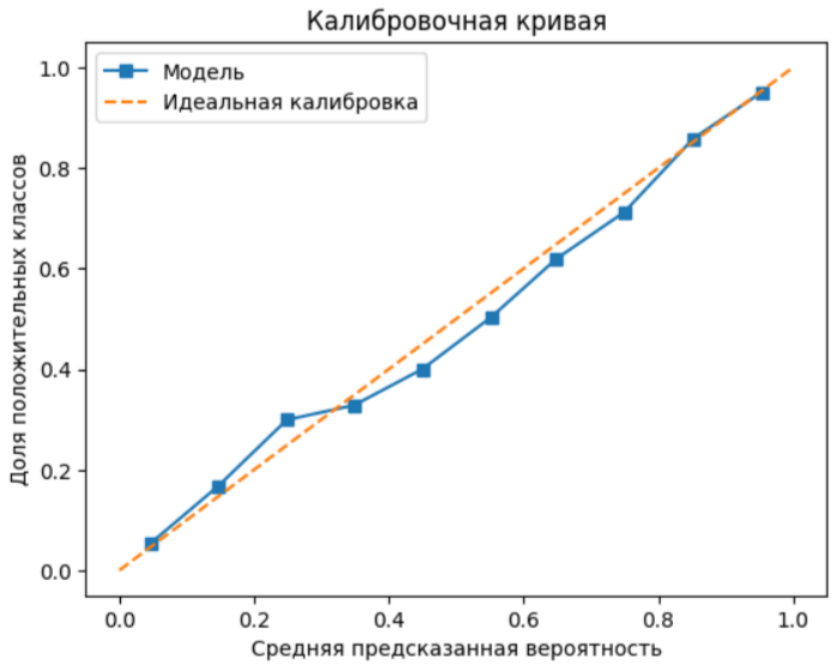

Рис. 12: Калибровка предсказаний модели

Калибровочная кривая модели практически свопала с идеальным случаем, а это означает, что модель возвращает корректные вероятнии (среди объектов с вероятностью положительного класса равной 10% - доля объектов действительно относящихся к положительному классу равна 10%)

3.10.4 Lift-анализ

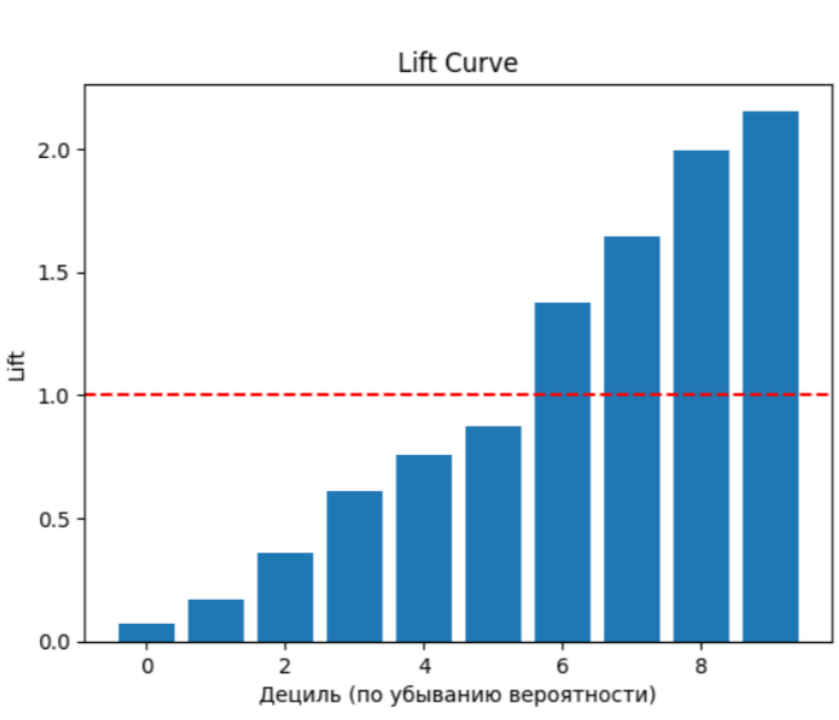

Рис. 13: Lift по децилям предсказанных вероятностей

Как мы видим из графика модель успещно концентрирует объекты положительного класса в топовых децилях по скору, что говорит о хорошей предсказательной способности (а также качестве ранжирования)

3.10.5 Анализ коэффициентов

Таблица 13: Коэффициенты логистической регрессии

Признак	Коэффициент	Отношение шансов
device_type	2.184	8.883
loyalty	0.088	1.092
age	0.045	1.046
time_spent	0.041	1.042

Поскольку мы не делали масштабирование данных, единственный вывод, который можно сделать - положительная связь таргета со всеми переменными.

4 Регрессия

В каждом из заданий, если не сказано иного, необходимо использовать хотя бы 3 (на ваш выбор) из следующих методов: случайный лес, метод наименьших квадратов, метод ближайших соседей и градиентный бустинг.

4.1 Задание 4.1

Отберите признаки, которые могут быть полезны при прогнозировании целевой (зависимой) переменной. Не включайте в число этих признаков переменную воздействия. Содержательно обоснуйте выбор признаков.

Целевой переменной в анализе является **выручка** (`revenue`). В качестве переменной воздействия выступает `push_subscription` — факт подписки пользователя на push-уведомления

Для отбора признаков был проведен анализ с помощью методов: случайный лес, метод наименьших квадратов, метод ближайших соседей и градиентный бустинг. Мы вычислили RMSE и MAPE для обучающей и тестовой выборок. Получили следующий результат:

Таблица 14: Сравнение моделей по RMSE и MAPE (Задание 4.1)

Model	**RMSE (Train)**	**RMSE (Test)**	**MAPE (Train)**	**MAPE (Test)**
OLS	59,434	60,803	4.5%	4.4%
Random Forest	**17,040**	47,735	**0.95%**	**2.5%**
Gradient Boosting	38,993	**45,090**	2.1%	2.5%
KNN	37,866	50,579	2.1%	2.7%

Интерпретация: Лучшая точность у Gradient Boosting, далее — Random Forest. Модель OLS показала худшее качество, так как она не может уловить нелинейные зависимости. KNN показал средний результат.

Вывод: RF и GB значительно превосходят линейную регрессию (OLS) по точности.

4.1.1 Сравнение RMSE и MAPE моделей

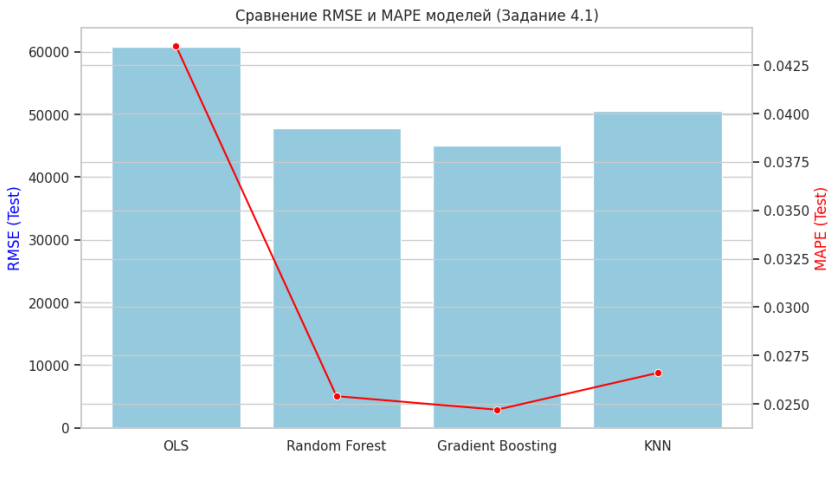

Рис. 14: Сравнение RMSE и MAPE моделей

4.2 Задание 4.2

Выберите произвольные значения гиперпараметров, а затем оцените и сравните (между методами) точность прогнозов с помощью RMSE и MAPE:
• на обучающей выборке.
• на тестовой выборке.
• с помощью кросс-валидации (используйте только обучающую выборку).
Проинтерпретируйте полученные результаты.

Здесь для Random Forest, Gradient Boosting и KNN мы сравнили RMSE до и после подбора параметров. Мы использовали кросс-валидацию по RMSE только на обучающей выборке.

Таблица 15: Сравнение моделей по метрикам RMSE, MAPE и кросс-валидации (Задание 4.2)

Model	**RMSE (Train)**	**RMSE (Test)**	**MAPE (Test)**	**RMSE (CV)**
Random Forest	38,015	45,644	2.47%	**43,980**
Gradient Boosting	**36,927**	**45,564**	**2.47%**	44,434
KNN	38,832	49,974	2.65%	46,176
OLS	59,434	60,803	4.35%	59,320

Выводы: Наибольшее улучшение дала настройка Gradient Boosting. Тюнинг гиперпараметров оказывает заметное влияние на точность предсказаний. Без него многие модели сильно проигрывают.

4.3 Задание 4.3

Для каждого метода с помощью кросс-валидации на обучающей выборке подберите оптимальные значения гиперпараметров (тюнинг). В качестве критерия качества используйте RMSE. Результат представьте в форме таблицы, в которой для каждого метода должны быть указаны:
• изначальные и подобранные значения гиперпараметров.
• кросс-валидационное значение RMSE на обучающей выборке с исходными и подобранными значениями гиперпараметров.
• значение RMSE на тестовой выборке с исходными и подобранными значе- ниями гиперпараметров. Проинтерпретируйте полученные результаты.

Здесь для подбора гиперпараметров мы использовали GridSearchCV: $n_estimators$, max_depth для RF;
$learning_rate$, $n_estimators$ для GB;
$n_neighbors$ для KNN.
Далее сравнили RMSE по кросс-валидации на тестовой выборке (до и после тюнинга).

Таблица 16: Сравнение моделей до и после тюнинга гиперпараметров (Задание 4.3)

Model	**CV RMSE (Initial)**	**CV RMSE (Tuned)**	**Test RMSE (Initial)**	**Test RMSE (Tuned)**
Random Forest	44100.00	43699.89	45605.94	45271.73
Gradient Boosting	54961.45	43936.91	56748.19	44969.82
KNN	48914.83	45774.22	52323.67	49159.55

Выводы: GridSearch позволил найти более оптимальные настройки для всех моделей. Gradient Boosting с тюнингом снова оказался самым точным на тесте. Тюнинг гиперпараметров с помощью кросс-валидации стабильно улучшает модели и позволяет избежать переобучения.

4.3.1 График сравнения RMSE до и после тюнинга

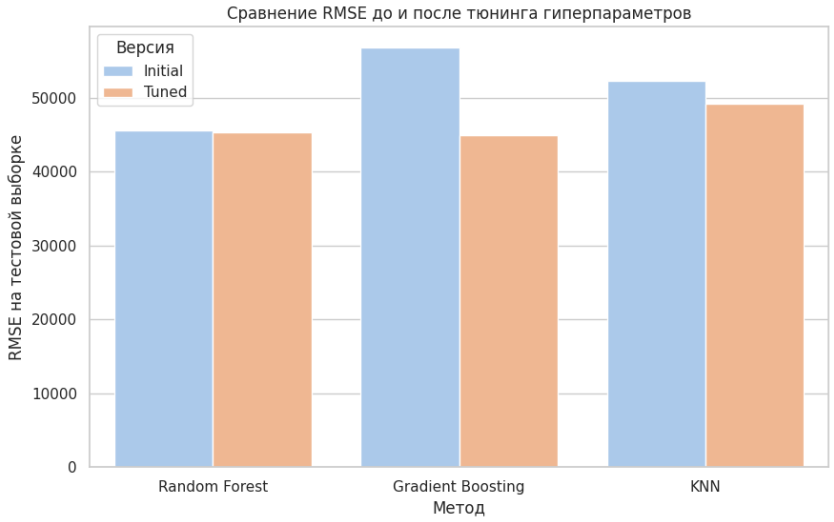

Рис. 15: График сравнения RMSE до и после тюнинга

Повышенная сложность: подберите на обучающей выборке оптимальные значения гиперпараметров градиентного бустинга ориентируясь на значение OOB (out-of-bag) ошибки. Сопоставьте гиперпараметры и точность на тесто- вой выборке для градиентного бустинга в зависимости от того, используется кросс-валидация или OOB ошибка.

4.4 Задание 4.4

На основании проделанного анализа выберите лучший и худший из обучен- ных классификаторов. Обоснуйте сделанный выбор.

Анализ включал сравнение четырёх моделей: OLS (линейная регрессия), Random Forest, Gradient Boosting, и KNN. Мы оценили их производительность по RMSE и MAPE как на тестовой выборке, так и с помощью кросс-валидации. Также были рассмотрены улучшения после тюнинга гиперпараметров.

Таблица 17: Сравнение моделей до и после тюнинга (Задания 4.1–4.3)

Model	**RMSE (Test)**	**MAPE (Test)**	**CV RMSE (Initial)**	**CV RMSE (Tuned)**	**Test RMSE (Initial)**	**Test RMSE (Tuned)**
OLS	60,803	4.35%	59,320	–	–	–
Random Forest	47,735	2.47%	44,100	**43,700**	45,606	**45,272**
Gradient Boosting	**45,564**	**2.47%**	54,961	**43,937**	56,748	**44,970**
KNN	49,974	2.65%	48,915	**45,774**	52,324	**49,160**

Выводы:

Gradient Boosting - лучший алгоритм, так как :

- Самый низкий RMSE на тестовой выборке (44,970)
- Самый низкий CV RMSE после тюнинга (43,937)
- Стабильность и высокая точность на кросс-валидации
- Тюнинг гиперпараметров значительно улучшил качество предсказаний

Gradient Boosting — наиболее надёжная и точная модель для текущей задачи.

OLS (линейная регрессия) показала наихудшие результаты:

- Самый высокий RMSE и MAPE на тесте.
- Не использует сложные (нелинейные) зависимости между признаками.
- Не допускает тюнинг (модель параметризована полностью), поэтому улучшение невозможно.

Вывод: OLS непригодна для данной задачи и может использоваться только как базовая модель для сравнения.

Итог:
Random Forest, Gradient Boosting существенно превосходят и линейные модели, и **KNN**.

Gradient Boosting лучше всех справляется с задачей, особенно после настройки гиперпараметров.

Тюнинг гиперпараметров даёт значительное улучшение результатов для большинства моделей, особенно **Gradient Boosting** и **KNN**.

Кросс-валидация — эффективный метод оценки стабильности модели и переобучения.

4.5 Задание 4.5

Повышенная сложность: включите в анализ дополнительный метод регрес- сии, не рассматривавшийся в курсе и не представленный в библиотеке scikit-learn. Опишите данный метод (принцип работы, преимущества и недостатки) и осуществите тюнинг гиперпараметров. Сопоставьте его точность на тестовой выборке с точностью лучшего из обученных вами ранее методов.

1. О методе CatBoost
CatBoost — это метод градиентного бустинга на деревьях решений, разработанный Яндексом. И в отличии от классического градиентного бустинга:

- Он использует Ordered Boosting: специальные техники уменьшения смещения при обучении, особенно эффективны на малых выборках.
- У него автоматическая обработка категориальных признаков — не требует ручного кодирования.
- Симметричные деревья: каждое разветвление основано на одинаковом условии, что ускоряет предсказания.
- Поддерживает обучение на GPU, многопоточность и out-of-the-box работу без глубокой настройки.

2. Преимущества CatBoost

- Работает без необходимости ручной предобработки;
- Поддерживает категориальные и числовые признаки напрямую;
- Хорошо обобщается и стабильно работает при ограниченном числе наблюдений;
- Имеет встроенную обработку пропущенных значений.

3. Недостатки

- Обучение может занимать больше времени, чем у RandomForest и GradientBoosting в scikit-learn;
- Более сложный для интерпретации, чем линейные модели;
- Требует установки внешней библиотеки (catboost).

4. Реализация и тюнинг модели

Мы выполнили:

- Установку библиотеки catboost;
- Обучение модели на X_train, y_train;
- Тюнинг гиперпараметров с помощью GridSearchCV (с использованием RMSE);
- Сравнение RMSE CatBoost с Gradient Boosting из задания 3 (лучшая модель ранее).

Таблица 18: Сравнение CatBoost и Gradient Boosting (Задание 4.5)

Модель	**RMSE (Test)**	**Лучшие параметры**
CatBoost	44,711.30	`{depth: 3, iterations: 100, learning_rate: 0.1}`
Gradient Boosting	44,970.00	`{n_estimators: 50, learning_rate: 0.1}`

CatBoost показал наибольшую точность (наименьшее RMSE на тестовой выборке). CatBoost имеет более устойчивую архитектуру (Ordered Boosting), встроенную обработку категориальных и пропущенных значений, хорошую производительность без глубокой настройки.

5 Эффекты воздействия

Для выполнения заданий данного раздела объедините обучающую и тестовую выборки в одну.

5.1 Задание 5.1

(a) Математически запишите и содержательно проинтерпретируйте потенциальные исходы целевой переменной. Объясните, как они связаны с наблюдаемыми значениями целевой переменной.

Для каждого пользователя i определим две потенциальные величины выручки:

$$Y_i(1) = \text{значение выручки у пользователя } i \text{ при получении пушей (т.е., если } W_i = 1)$$
$$Y_i(0) = \text{значение выручки у пользователя } i \text{ без пушей (т.е., если } W_i = 0)$$

Интерпретация: Мы можем наблюдать выручку у пользователей, на которые влияет бинарный фактор - подписка на пуши. Этот фактор принимает лишь 2 значения (да или нет, 0 или 1), значит мы можем наблюдать лишь 2 исхода: выручка, когда пользователь подписан и выручка, когда пользователь не подписан. Поскольку переменная воздействия бинарная и принимает значения 0 или 1, она же и является индикатором того, подписан пользователь или нет. Тогда можем записать потенциальный исход в виде формулы ниже (логичность ее можно проверить, подставив 0 или 1 в индикатор).

Наблюдаем исход:

$$Y_i^{\text{obs}} = Y_i(W_i) = W_i \cdot Y_i(1) + (1 - W_i) \cdot Y_i(0)$$

где $W_i \in \{0, 1\}$ — индикатор, была ли у пользователя подписка на пуши.

5.2 Задание 5.2

(a) Используя симулированные, но недоступные в реальных данных потенциальные исходы (гипотетические значения), получите оценки:

- среднего эффекта воздействия (ATE),
- условных средних эффектов воздействия (CATE),
- локального среднего эффекта воздействия (LATE).

Для ATE и LATE представьте результаты в виде таблицы, для CATE постройте гистограмму или ядерную оценку функции плотности. Проинтерпретируйте полученные значения.

Примечание. Для получения более точных оценок эффектов воздействия с помощью потенциальных исходов можно симулировать большое число наблюдений (например, несколько миллионов). Затем, для ускорения вычислений, при оценивании эффектов воздействия по наблюдаемым значениям используйте подвыборку, например, из 10 000 наблюдений.

В данном задании мы сгенерировали выборку из 5 млн наблюдений и взяди семпл размером 10 000.

Истинные эффекты воздействия определяются следующим образом:

$$ATE = \mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0]$$

Локальный средний эффект воздействия (LATE)

$$LATE = \mathbb{E}[Y(1) - Y(0) \mid D = 1, Z = 1] - \mathbb{E}[Y(1) - Y(0) \mid D = 0, Z = 0]$$

где:

- D — индикатор (подписка: $D = 1$ при подписке, $D = 0$ без подписки),
- Z — инструментальная переменная (например, $Z = 1$ для iOS, $Z = 0$ для Android),
- LATE оценивает эффект для compliers, то есть пользователей, чье решение о подписке определяется инструментальной переменной.

Условный средний эффект воздействия (CATE)

$$CATE(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$$

где:

- X — вектор переменных,
- $CATE(x)$ — эффект воздействия для конкретной подгруппы пользователей (характеризуемой значением x).

Результаты оценивания

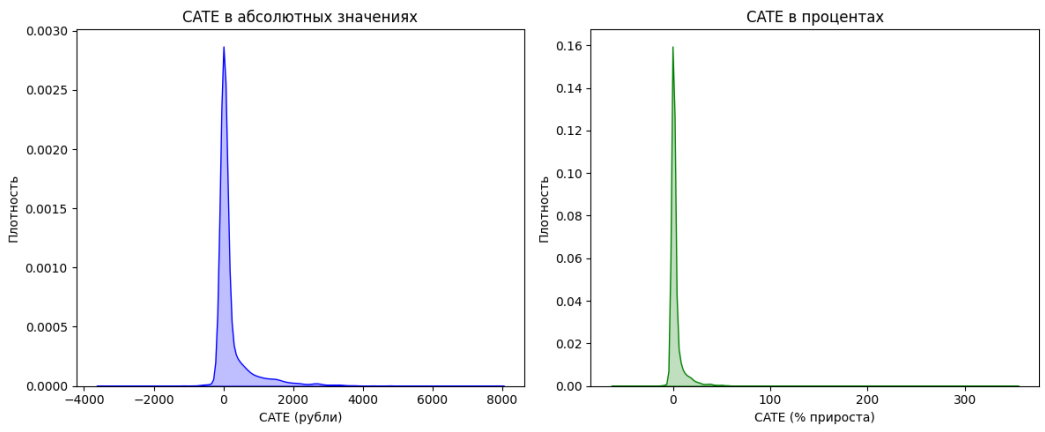

Рис. 16: Ядерная оценка плотности CATE

Эффект	Значение
ATE	236.073
LATE	182.340

Таблица 19: Оценки средних эффектов воздействия

Интерпретация результатов

- **ATE (236,073):** Средний эффект подписки на push-уведомления составляет 236,073 рублей увеличения стоимости покупки для всех пользователей. Это может означать, что push-уведомления стимулируют пользователей чаще совершать покупки или увеличивать их сумму.
- **LATE (182,340):** Локальный средний эффект для пользователей, чья подписка определяется типом устройства (compliers), составляет 182,340 рубля. Для этой подгруппы эффект подписки ниже, что может быть связано с тем, что пользователи, чья подписка зависит от типа устройства, совершают менее дорогие покупки или реже реагируют на push-уведомления.
- **CATE:** График ядерной оценки плотности условного среднего эффекта воздействия показывает пик около нуля и широкий диапазон значений. Для некоторых пользователей подписка на push-уведомления как снижает, так и увеличивает стоимость покупки.

5.3 Задание 5.3

(a) Оцените средний эффект воздействия как разницу в средних между группами, получившими и не получившими воздействие. Опишите недостатки данного подхода с учётом специфики вашей экономической задачи.

Примечание. В этом и последующих пунктах (если не указано иное) используйте только наблюдаемые значения целевой переменной.

$$\widehat{ATE}_{\text{naive}} = \overline{Y}_1 - \overline{Y}_0 = \frac{1}{n_1} \sum_{i:\, W_i=1} Y_i - \frac{1}{n_0} \sum_{i:\, W_i=0} Y_i$$

где $\overline{Y}_1$ — среднее значение выручки среди пользователей с подпиской на push, n_1 — их число, $\overline{Y}_0$ и n_0 — аналогично для пользователей без подписки, W_i — индикатор наличия подписки.

Наивная оценка ATE: 420.180 (рубли)

Вывод

Наивная оценка эффекта воздействия переоценивает эффект, не учитывается эндгенность (например переменная о типе устройства).

- Имеется selection bias (например, может быть так, что на пуши чаще подписывается молодежь, чем люди старше, так как реже замечают их)
- Имеется ненаблюдаемая переменная лояльности (может лояльным клиентам нравятся пуши приложения)
- Такой подходи учитывает только наблюдаемые исходы

Основная идея такая, что группы с включенными и выклеченными пушами могут сильно различаться по другим факторам, и тогда наивный подход будет давать смещенную оценку.

5.4 Задание 5.4

(a) Используя оценки, полученные лучшими из ранее обученных классификационных и регрессионных моделей, оцените средний эффект воздействия с помощью следующих методов:

- метода наименьших квадратов (OLS),
- условных математических ожиданий,
- взвешивания на обратные вероятности (IPW),
- метода, обладающего двойной устойчивостью,
- двойного машинного обучения (Double Machine Learning).

Сравните результаты и назовите ключевую предпосылку этих методов. Обсудите причины, по которым эта предпосылка может соблюдаться или нарушаться в вашем случае. Приведите содержательную экономическую интерпретацию оценки среднего эффекта воздействия.

Повышенная сложность: добавьте в сравнение дополнительный метод, не рассматривавшийся в курсе, опишите его принцип работы, преимущества и недостатки по сравнению с другими подходами.

Результаты оценивания

Таблица 20: Оценки среднего эффекта воздействия различными методами

Метод оценки	**ATE**
OLS	73.922
Условное математическое ожидание	205.740
IPW (Inverse Probability Weighting)	-51.406
Double Robust	230.535
DoubleML	167.047

OLS дает наименьшую положительную оценку (73.922), что может указывать на недооценку эффекта из-за неучтенных нелинейных зависимостей или нарушения предположений линейной модели.

Условное математическое ожидание показывает значительно более высокую оценку (205.740), что говорит о существенном эффекте при учете условных ожиданий.

IPW демонстрирует отрицательный эффект (-51.406), резко отличаясь от других методов. Это может указывать на проблемы с оценкой вероятностей воздействия.

Double Robust дает большую оценку эффекта (230.535).

DoubleML показывает умеренно высокий эффект (167.047).

Разброс оценок от -51.406 до 230.535 указывает на значительную чувствительность результатов к выбору метода оценивания. Наиболее надежными, с теоретической точки зрения, считаются методы Double Robust и DoubleML, поскольку они обладают свойством двойной робастности.

Ключевая предпосылка: Оценка эффекта воздействия (ATE, CATE и др.) требует выполнения условия об условной независимости:

$$E(Y_{1i} \mid X_i, T_i = 1) = E(Y_{1i} \mid X_i), \quad E(Y_{0i} \mid X_i, T_i = 0) = E(Y_{0i} \mid X_i),$$

где Y_{1i}, Y_{0i} — потенциальные исходы, T_i — воздействие, X_i — ковариаты.

- Для методов OLS, IPW, DR, DoubleML это означает: если учесть все важные переменные X_i, назначение воздействия становится случайным и группы сравнимы.
- Для LATE с инструментами требуется экзогенность: инструмент влияет на воздействие, но не влияет напрямую на исход.

В нашем случае условная независимость может соблюдаться, если все важные факторы, влияющие на push-подписки и на выручку пользователя, учтены в наборе признаков, таких как возраст, пол, тип устройства, лояльность, “новый пользователь”, время в приложении и добавление в корзину. Тогда среди пользователей с одинаковыми X_i факт подписки почти случайно.

Нарушается из-за:

- **Ненаблюдаемых факторов,** не включённых в X_i — например, личные паттерны поведения, мотивы или рекламное влияние, которые влияют и на вероятность подписки, и на покупки.
- **Selection bias:** если определённые группы (например, более активные или более молодые пользователи) чаще подписываются на пуши и одновременно больше покупают, эффект будет переоценён.
- **Недостаток перекрытия (overlap):** если у некоторых подгрупп вероятность подписки почти 0 или 1, сравнение невозможно.

Экономическая интерпретация:

Оценка ATE показывает, насколько в среднем изменится выручка пользователя, если “случайно” подписать его на push-уведомления. Это можно считать как потенциальный прирост дохода, который бизнес получает за счёт вовлечения пользователей через push-коммуникации. Однако если условие независимости нарушено, оценка будет смещать истинный эффект — то есть реальное влияние push-уведомлений может быть ниже или выше, чем рассчитано по модели.

5.5 Задание 5.5

(a) Оцените локальный условный эффект воздействия с помощью:

- двойного машинного обучения без инструментальной переменной,
- двойного машинного обучения с инструментальной переменной.

Сравните результаты и объясните, в чем состоит различие между средним эффектом воздействия и локальным средним эффектом воздействия. Приведите экономическую интерпретацию оценки локального среднего эффекта воздействия.

Повышенная сложность: дополнительно воспользуйтесь параметрической моделью, например, пакетом `switchSelection`. Обсудите преимущества и недостатки подобного подхода по сравнению с двойным машинным обучением. Классический метод инструментальных переменных параметрическим подходом не считается.

Результаты оценивания

Таблица 21: Оценки средних эффектов воздействия

Метод	**Оценка эффекта**	**95% ДИ (нижняя)**	**95% ДИ (верхняя)**
DML без инструмента (ATE)	291.273	271.079	311.467
DML с инструментом (LATE)	83.682	13.214	154.149

Вывод

1. Средний эффект воздействия (ATE, DoubleML без инструмента)

Оценка эффекта составляет **291.273 рубля** (95% ДИ: [271.079; 311.467]). Это означает средний прирост выручки от push-подписок для всей совокупности пользователей. Измеряет влияние push-подписки на выручку для всех пользователей, вне зависимости от причин и обстоятельств, по которым они подписались. Как ранее упоминали, оценка может быть смещенной.

2. Локальный средний эффект воздействия (LATE, DoubleML с инструментом)

Эффект среди пользователей, чьё решение о подписке на пуши меняется под влиянием типа устройства — составляет **83.682 рублей** (95% ДИ: [13.214; 154.149]). Эффект статистически значим, но существенно меньше, чем ATE. Это, по сути, эффект именно от изменения внешнего фактора (автоматически подписывают на пуши или нет), а не от самоотбора.

ATE сильно больше LATE в этом пункте, что может говорить о том, что эффект от пушей может быть в большей степень определен другими характеристиками пользователей (эффект более заметен из-за тех, кто подписался бы на пуши в любом случае).

Экономическая интерпретация:

- **LATE** показывает, что эффект того, что мы будем активнее подписывать на пуши владельцев айфонов, будет менее значимым, чем на всей выборке.
- Прирост стоимости покупки кажется незначимым, для бизнеса это сигнал, что вложения в разработку вовлекающих фичей для IOS могут не окупиться. Необходимы другие тактики.
- Эффект значим, если подписывать на пуши уже вовлеченных пользователей (в перспективе можно выявить через прокси метрики).

5.6 Задание 5.6

(a) Оцените условные средние эффекты воздействия с помощью следующих методов:

- метода наименьших квадратов (OLS),
- S-learner,
- T-learner,
- способа трансформации классов,
- X-learner.

Сравните результаты и обсудите целесообразность применения метода X-learner в вашей задаче. Опишите, как можно использовать полученные оценки в бизнесе или для реализации государственных программ.

Повышенная сложность: включите дополнительный метод, не обсуждавшийся в курсе, опишите его принцип работы, преимущества и недостатки по сравнению с остальными способами.

Результаты оценивания

Метод	**CATE (среднее)**
OLS (T-learner на МНК)	285.532 743
S-learner	341.122 505
T-learner (ML)	340.844 524
Class transformation	362.993 161
X-learner	334.723 042

Таблица 22: Оценки условных средних эффектов воздействия различными методами

Вывод

- ML-методы (S-learner, T-learner, X-learner, классовая трансформация) дают более высокие оценки CATE, чем простая модель OLS, что говорит о наличии нелинейных связей между признаками и эффектом воздействия.
- X-learner и T-learner выдают близкие результаты, модели устойчивы.
- Трансформация классов даёт максимально высокий средний эффект, S-learner и X-learner — умеренно высокие, но все превышают OLS.

Мотивация для X-learner

- X-learner необходим, если эффект воздействия зависит от индивидуальных характеристик и наблюдается дисбаланс между treated- и control-группами. Такое часто бывает в реальной жизни.
- Метод позволяет точнее выделить, для кого эффект максимален, поскольку модель строится отдельно на каждой группе.

CATE в практике

- Использовать индивидуальные оценки для точечной проработки: push-уведомления отправлять тем, для кого прогнозируется наибольший отклик.
- Сегментировать аудиторию (далее настраиваемые рекламные кампании, программы лояльности, механики удержания и т.п.).

Методы, способные учитывать индивидуальные эффекты (например, X-learner), более информативны для принятия решений, чем усреднённые оценки. Это позволяет фокусироваться на отдельных сегментах и повышать эффекты от интервенций.

5.7 Задание 5.7

(a) Выберите лучшую модель для оценивания условных средних эффектов воздействия, используя:
- истинные значения условных средних эффектов воздействия,
- прогнозную точность моделей,
- псевдоисходы.

Проинтерпретируйте различия результатов различных подходов.

Результаты

Таблица 23: Сравнение методов оценки условных средних эффектов воздействия

Метод	CATE (mean)	MSE с истиной	Test MSE	MSE (vs псевдоисход)
OLS (T-learner на МНК)	285.533	3.992×10^5	4.055×10^5	4.475×10^8
S-learner	341.123	3.922×10^5	3.937×10^5	4.465×10^8
T-learner (ML)	340.845	3.856×10^5	3.921×10^5	4.465×10^8
Class transformation	362.993	6.112×10^6	6.418×10^6	4.328×10^8
X-learner	334.723	3.955×10^5	3.988×10^5	4.465×10^8

Выбор наилучшей модели:

(a) **Сравнение с истинными CATE**: меньшее отклонение от истины показывает T-learner (ML), за ним следуют S-learner и X-learner. OLS демонстрирует наихудший результат среди ML-методов.

(b) **По тестовой выборке (Test MSE)**: лучшие результаты показывает также T-learner (ML), затем S-learner и X-learner.

(c) **По псевдоисходам**: наименьшую ошибку демонстрирует классовая трансформация, в то время как другие методы имеют очень близкие значения. Это может быть спецификой метрики, но отличие существенно.

Оптимальный выбор — T-learner (ML), так как он показывает наилучшие результаты как по MSE с истиной, так и по тестовому MSE. X-learner и S-learner также демонстрируют высокую точность и могут быть хорошими альтернативами.

Классовая трансформация показывает наихудшие результаты по MSE с истиной и Test MSE, что на порядок хуже других методов, несмотря на хорошие показатели по метрике псевдоисходов.

OLS уступает машинным методам по точности, но может быть предпочтительным, если важна интерпретируемость результатов и если зависимости между факторами и эффектом близки к линейным.

5.8 Задание 5.8

(a) Оцените средние и локальные средние эффекты воздействия с использованием худших из обученных моделей. Сопоставьте результаты с оценками, полученными с помощью лучших моделей, и сделайте вывод об устойчивости результатов к качеству методов машинного обучения.

Результаты оценивания

Таблица 24: Оценки среднего эффекта воздействия при использовании худших моделей (скорректированные)

Метод оценки	ATE (худшие модели)
OLS	73.922
Условное математическое ожидание	114.823
IPW (Inverse Probability Weighting)	-786.879
Double Robust	1.160
DoubleML	125.922

OLS сохраняет устойчивость (73.922 в обоих случаях), что ожидаемо, так как метод не зависит от моделей.

Условное математическое ожидание показывает заметное снижение на с 205.740 до 114.823, что указывает на важность качества моделей при оценке условных ожиданий.

IPW демонстрирует уменьшение оценки с -51.406 до -786.879. Это подтверждает чувствительность метода к оценке.

Таблица 25: Сравнение оценок среднего эффекта воздействия при использовании разных моделей

Метод оценки	**ATE (лучшие модели)**	**ATE (худшие модели)**
OLS	73.922	73.922
Условное математическое ожидание	205.740	114.823
IPW (Inverse Probability Weighting)	-51.406	-786.879
Double Robust	230.535	1.160
DoubleML	167.047	125.922

Double Robust почти обнуляет эффект при использовании худших моделей. Это противоречит двойной робастности метода, возможны ошибки.

DoubleML снижает с 167.047 до 125.922.

Вывод: Результаты показывают чувствительность методов оценки к качеству используемых моделей. Особенно для методов IPW и Double Robust. DoubleML относительно устойчив. При этом OLS вообще не зависит, но может давать смещенные оценки при нарушении предположений линейности.

5.9 Задание 5.9

(a) Резюмируйте ключевые выводы анализа, проведённого в этом разделе.

5.10 Задание 5.10

(a) При необходимости проведите дополнительный анализ по вашему усмотрению.

6 Резюме анализа эффектов воздействия

6.0.1 Потенциальные исходы и интерпретация

Для каждого пользователя существуют два потенциальных исхода – при наличии и отсутствии воздействия (подписки на push-уведомления). Реально наблюдается только один из них; наблюдаемое значение целевой переменной зависит от факта воздействия.

6.0.2 Способы оценки эффектов

Наивные методы переоценивают эффект из-за смещения. OLS менее чувствителен к качеству моделей, но плохо улавливает нелинейности и взаимодействия признаков. Может приводить к смещенным оценкам. ML-методы (S-/T-/X-learners, DoubleML и др.) позволяют получать более точные оценки, учитывать сложные зависимости, исследовать гетерогенность эффектов. Методы DR, DoubleML более устойчивы, если хотя бы одна из моделей специфицирована верно. IPW чувствителен к корректности оценки вероятности факта подписки на пуши.

6.0.3 Устойчивость результатов

Качество моделей существенно влияет на результаты для методов, требующих оценки параметров.

6.0.4 Причины различий между ATE и LATE

ATE оценивает эффект для всех пользователей, LATE – для тех, кто меняет своё поведение из-за внешнего фактора. Если подписка на пуши зависит от наблюдаемых и ненаблюдаемых факторов, то LATE выделяет настоящий эффект интервенции. LATE меньше ATE, следовательно, максимальный эффект – у уже мотивированных пользователей, а не у тех, кого можно принудительно подключить к пушам.

6.0.5 Практические применения

CATE позволяет направлять push-уведомления тем группам, для кого ожидается наибольший прирост выручки. **LATE** – важен при принятии решений об подключении (например, при введении доп механик).

6.0.6 Выбор моделей

Лучшие результаты по точности и устойчивости дают X-learner, T-learner (ML), DoubleML. По псевдоисходам иногда выигрывает классовая трансформация, но при этом сильно уступает по MSE с истиной. OLS – компромисс между интерпретируемостью и точностью.

6.0.7 Ограничения

При нарушении предпосылок (экзогенность, overlap, учет всех факторов) многие методы будут давать смещённые и ненадёжные оценки.

6.0.8 Резюме

Влияние push-уведомлений на поведение пользователей действительно существует, наибольший эффект фиксируется у некоторых сегментов аудитории. Наивные оценки могут быть существенно смещены из-за наличия смешивающих факторов. Методы машинного обучения позволяют корректировать это смещение и выделять нужные группы, однако для устойчивых результатов важно качество моделей и корректная спецификация.