

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Прикладная математика и информатика»

УДК 004.85

**Отчет об исследовательском проекте на тему:**  
**Оптимизация маркетингового бюджета с использованием ML методов**

**Выполнил студент:**

группы БПМИ223, 3 курса

Сидоров Александр Михайлович

**Принял руководитель проекта:**

Богущий Денис Александрович

Научный сотрудник/Аспирант

Факультет компьютерных наук НИУ ВШЭ

Москва 2025

# Содержание

<b>Аннотация</b>	<b>3</b>
<b>1 Введение</b>	<b>4</b>
<b>2 Обзор литературы</b>	<b>5</b>
<b>3 Данные</b>	<b>6</b>
3.1 Описание датасета . . . . .	6
3.2 Подготовка датасета . . . . .	7
3.2.1 Разбиение на сегменты . . . . .	7
3.2.2 Формирование данных для обучения . . . . .	8
<b>4 Нейронная сеть</b>	<b>9</b>
4.1 Общее описание . . . . .	9
4.2 Реализация нейронной сети . . . . .	10
<b>5 Логистическая регрессия</b>	<b>11</b>
5.1 Аналитическая формула градиента . . . . .	11
5.1.1 Производная логарифмической функции . . . . .	11
5.1.2 Производная сигмоиды . . . . .	12
5.1.3 Производная линейной функции . . . . .	12
5.2 Реализация логистической регрессии . . . . .	13
<b>6 Эксперименты</b>	<b>15</b>
6.1 Метрика качества . . . . .	15
6.2 Логистическая регрессия v1 . . . . .	15
6.3 Логистическая регрессия v2 . . . . .	16
6.4 Нейронная сеть и логистическая регрессия . . . . .	17
<b>7 Заключение</b>	<b>18</b>
7.1 Выводы . . . . .	18
7.2 Перспективы дальнейших исследований . . . . .	18
<b>Список литературы</b>	<b>19</b>

## **Аннотация**

В этом проекте мы разрабатываем и реализовываем методы машинного и глубинного обучения для определения оптимального распределения маркетингового бюджета на промо предложения, чтобы максимизировать эффективность рекламных кампаний. Основная сложность задачи в том, чтобы учесть долгосрочный эффект от маркетинговых стимулов на потребителей, обычно для этого используются различные поправки в оптимизируемую функцию наград.

## **Ключевые слова**

Глубинное обучение, машинное обучение, маркетинг

# 1 Введение

В современном цифровом мире маркетинговые кампании играют ключевую роль в формировании спроса и увеличении продаж. Однако эффективное распределение маркетингового бюджета остается сложной задачей, так как влияние рекламных активностей на пользователей носит как краткосрочный, так и долгосрочный характер. Оптимизация бюджетных затрат требует учета множества факторов, включая потребительское поведение, стоимость привлечения клиентов и возврат на инвестиции.

Традиционные методы анализа маркетинговой эффективности, такие как статистические модели, часто не учитывают сложные нелинейные зависимости между рекламными стимулами и реакцией аудитории. В последние годы методы машинного обучения (ML) становятся все более популярными в этой области, поскольку позволяют находить скрытые закономерности в данных и прогнозировать эффект от различных стратегий распределения бюджета.

Цель данной работы — разработка и реализация ML-моделей для оптимального распределения маркетингового бюджета с учетом долгосрочных и краткосрочных эффектов рекламных кампаний. В рамках исследования рассматриваются существующие подходы к моделированию влияния маркетинговых стимулов, а также разрабатывается метод оптимизации, позволяющий максимизировать эффективность промо-предложений.

В первой части работы представлен обзор существующих методов оценки маркетинговых воздействий. Далее рассматриваются основные принципы построения модели и подходы к ее обучению. В заключительной части проводится экспериментальная проверка предложенной методики и анализ полученных результатов.

## 2 Обзор литературы

Традиционные методы моделирования маркетинговых эффектов часто основываются на эконометрических моделях, таких как линейные и лог-линейные регрессии, модели с постоянной эластичностью спроса и логистические кривые. Однако эти подходы обладают рядом ограничений, таких как слабая адаптивность к изменяющимся рыночным условиям и недостаточная способность учитывать сложные взаимосвязи между маркетинговыми вложениями и потребительским поведением.

Существует двухэтапный подход к решению задачи оптимизации бюджета. *Первый этап* - построение модели для предсказания отклика покупателей на различные стимулы (промокод, скидка, купоны и тд). *Второй этап* - определение необходимого бюджета посредством решения условной задачи оптимизации. Данный подход лежит в основе исследования Alibaba Group [1], в котором представлено прогнозирование рыночной реакции, сочетающее в себе логистическую кривую спроса и нейронные сети.

Несмотря на интерпретируемость и гибкость, двухэтапный подход не способен качественно оценивать длительные эффекты, например, приобретение привычки у покупателя спустя несколько покупок. Для решения данной проблемы может быть использовано обучение с подкреплением (Reinforcement Learning, RL) [3], которое рассматривает задачу оптимизации как динамический процесс принятия решений, где каждое действие имеет влияние не только на текущий, но и на будущий результат.

Еще одним способом решить проблему двухэтапного подхода является создание единого метода, в котором оптимизация бюджета интегрирована в процесс обучения модели [2]. В данном случае вводится одна функция, которая учитывает маркетинговые цели и бюджетные ограничения.

## 3 Данные

Датасетом для проверки качества определения отклика покупателей был выбран Breakfast at the Frat<sup>1</sup>. Он содержит информацию о магазинах, продуктах и продажах за 156 недель. Транзакции объединены по неделям.

### 3.1 Описание датасета

Датасет Breakfast at the Frat содержит в себе 3 таблицы:

- 1 dh Store Lookup - данные о магазинах
- 2 dh Products Lookup - данные о продуктах
- 3 dh Transaction Data - данные о транзакциях за 156 недель, сгруппированные по неделям и магазинам

Магазины бывают 3-х типов:

- 1 VALUE - дешевый сегмент
- 2 MAINSTREAM - средний сегмент
- 3 UPSCALE - дорогой сегмент

Товары разбиты на 7 категорий:

- 1 PRETZELS
- 2 MOUTHWASHES(ANTISEPTIC)
- 3 ALL FAMILY CEREAL
- 4 ADULT CEREAL
- 5 PIZZA/PREMIUM
- 6 KIDS CEREAL
- 7 MOUTHWASH/RINSES AND SPRAYS

---

<sup>1</sup><https://www.dunnhumby.com/careers/engineering/sourcefiles>

## 3.2 Подготовка датасета

### 3.2.1 Разбиение на сегменты

Из-за немногочисленности и схожести по своей сути было принято решение объединить некоторые сегменты, а именно:

- ADULT CEREAL + KIDS CEREAL  
→ CEREAL
- MOUTHWASHES (ANTISEPTIC) + MOUTHWASH/RINSES AND SPRAYS  
→ ANTISEPTIC

Итоговые сегменты определяются полями SUB\_CATEGORY и SEG\_VALUE\_NAME. Их получилось 15 штук.

Было необходимо понять, насколько равномерными и стационарными являются получившиеся сегменты. По графику среднего и стандартного отклонения(3.1) видно, что в данных нет каких-либо аномалий и они являются достаточно стационарными, то есть они не сильно меняются с течением времени. Аналогичный результат можно увидеть на графиках, изображающих продажи конкретной категории в разрезе разных типов магазинов(3.2) - пропорции неизменны. Есть различия в объемах продаж сегментов(3.3), однако это наиболее оптимальный способ разбивки, поскольку он сохраняет информацию о типе сегмента и в то же время минимизирует разницу в объемах.

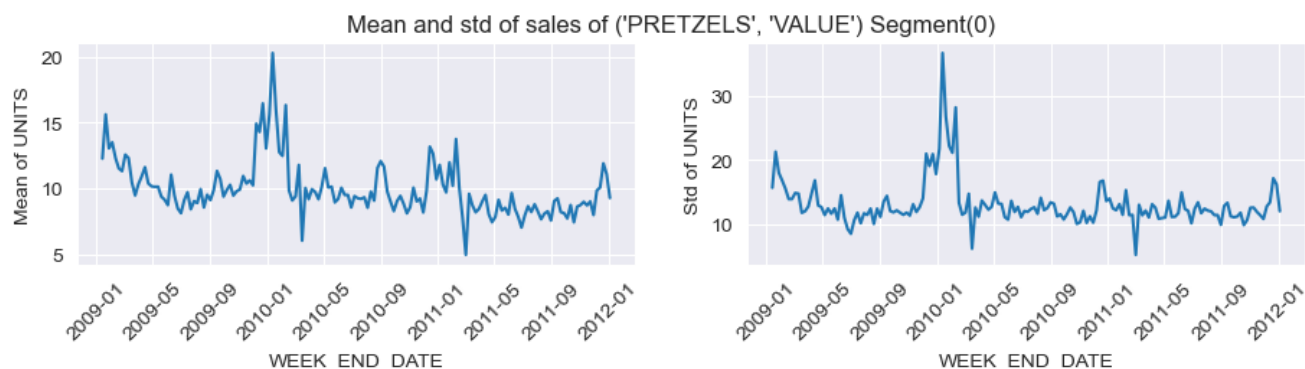


Рис. 3.1: Среднее и стандартное отклонение количества покупок в 0-м сегменте по неделям

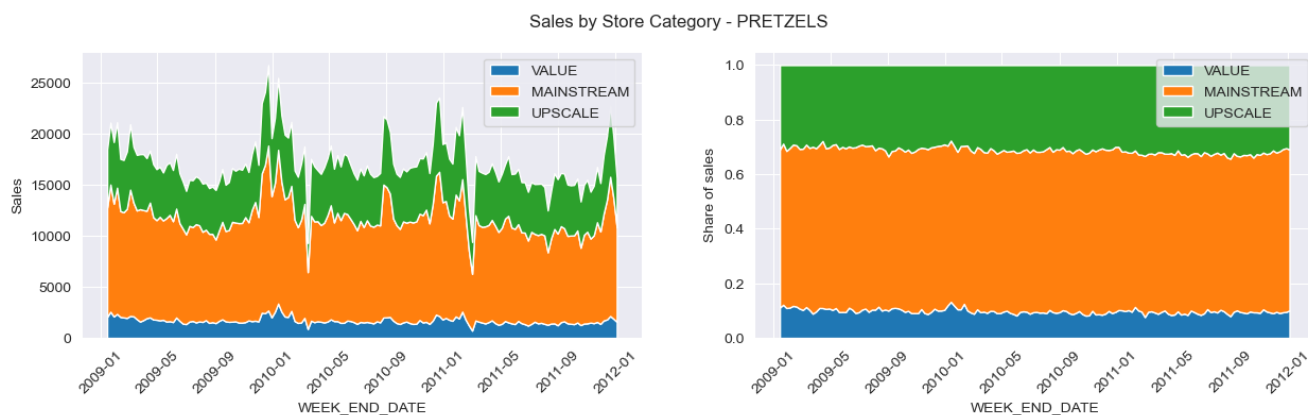


Рис. 3.2: Количество продаж категории "PRETZELS" по типу магазина

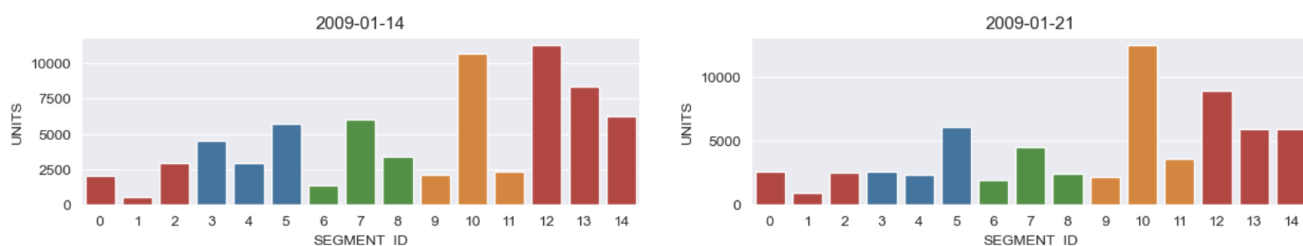


Рис. 3.3: Количество продаж по сегментам

### 3.2.2 Формирование данных для обучения

Пришло время сформировать данные, которые будут использоваться непосредственно в процессе обучения.

Выше было описано формирование 15 сегментов. Данные агрегируем по началу недели, поскольку в изначальном документе они были так разбиты. Дополнительно разбиваем данные по номерам магазинов - это разумно, потому что мы хотим оценить чувствительность к скидкам внутри одного сегмента, и для этого нужно рассматривать все магазины, и в качестве таргета брать то, как они отбирают доли рынка друг у друга.

Итоговый датасет имеет следующий вид(3.4):

	SEGMENT_ID	SEG_VALUE_NAME	SUB_CATEGORY	WEEK_END_DATE	STORE_NUM	SEGMENT_COST	MARKET_SHARE
17097	5	UPSCALE	PRETZELS	2011-11-30	6179	6.936254	0.092471
40970	12	MAINSTREAM	ALL FAMILY CEREAL	2009-02-25	25253	17.577757	0.014625
9494	3	VALUE	CEREAL	2009-09-30	8035	0.000000	0.034711

Рис. 3.4: Итоговый вид датасета

- SEG\_VALUE\_NAME - тип магазина
- SUB\_CATEGORY - подкатегория товара
- SEGMENT\_ID - номер сегмента



- WEEK\_END\_DATE - дата соответствующей недели
- STORE\_NUM - номер магазина
- SEGMENT\_COST - стоимость сегмента
- MARKET\_SHARE - доля рынка сегмента

SEGMENT\_COST - относительная величина скидки на сегмент в конкретном магазине в конкретную неделю

MARKET\_SHARE - доля рынка сегмента в конкретном магазине в конкретную неделю

## 4 Нейронная сеть

### 4.1 Общее описание

Нейронные сети представляют собой класс моделей машинного обучения, структурно напоминающий работу человеческого мозга. Основу составляют нейроны(узлы), группирующиеся в слои и осуществляющие выделение определенных зависимостей. Данный тип моделей широко применяется для решения различных задач и, в частности, для решения задачи предсказания.

Самым простым примером нейронной сети является Полносвязная нейронная сеть(Fully Connected Neural Networks, FCNN). Полносвязная нейронная сеть состоит из последовательности слоев, каждый из которых содержит определенное число нейронов. В полносвязной архитектуре каждый нейрон одного слоя соединяется с каждым нейроном следующего слоя. Структура сети включает три основных типа слоев:

- 1 Входной слой: принимает исходные данные.
- 2 Скрытые слои: выполняют нелинейные преобразования данных с помощью весов и функций активации.
- 3 Выходной слой: генерирует итоговый прогноз или решение задачи.

Каждое соединение между нейронами характеризуется весовым коэффициентом, который настраивается в процессе обучения. Выход каждого нейрона вычисляется как взвешенная сумма входных сигналов с последующим применением функции активации, такой как ReLU (Rectified Linear Unit), сигмоида или гиперболический тангенс.

Обучение сети осуществляется с использованием метода обратного распространения ошибки (Backpropagation) и оптимизационных алгоритмов, таких как стохастический градиентный спуск (SGD) или его усовершенствованные версии (Adam).

## 4.2 Реализация нейронной сети

В данном исследовании была выбрана архитектура нейронной сети, которая состоит из 5 полносвязных слоев с функцией активации ReLU. Для входного слоя контекстные переменные были закодированы с помощью OneHotEncoding.

```
1 class A_Net(nn.Module):
2     def __init__(self, input_dim, hidden_dim=16):
3         super(A_Net, self).__init__()
4         self.fc1 = nn.Linear(input_dim, hidden_dim)
5         self.fc2 = nn.Linear(hidden_dim, hidden_dim)
6         self.fc3 = nn.Linear(hidden_dim, hidden_dim)
7         self.fc4 = nn.Linear(hidden_dim, hidden_dim)
8         self.fc5 = nn.Linear(hidden_dim, hidden_dim)
9         self.fc_out = nn.Linear(hidden_dim, 1)
10
11     def forward(self, x):
12         x = F.relu(self.fc1(x))
13         x = F.relu(self.fc2(x))
14         x = F.relu(self.fc3(x))
15         x = F.relu(self.fc4(x))
16         x = F.relu(self.fc5(x))
17
18         a = torch.sigmoid(self.fc_out(x))
19         return a
```

Листинг 1: Реализация класса нейронной сети

## 5 Логистическая регрессия

### 5.1 Аналитическая формула градиента

В данном случае целевой переменной выступает доля рынка  $q$ :

$$q(c) := \frac{d(c)}{D} = \frac{1}{1 + \exp\{-(a + bc)\}} = \frac{1}{1 + \exp\{-X\theta\}} = \frac{1}{1 + e^{-z}}$$

Для обучение логистической модели необходимо найти градиент отрицательного лог-правдоподобия, имеющий вид:

$$J(\theta) = -\frac{1}{n} [y \cdot \log h(z) + (1 - y) \cdot \log(1 - h(z))]$$

где  $h$  — это сигмоида  $\frac{1}{1+e^{-z}}$ , а  $z$  — линейная функция  $X\theta$ .

Нам нужно найти ее градиент:

$$\nabla_{\theta} J \left( -\frac{1}{n} [y \cdot \log(h(z)) + (1 - y) \cdot \log(1 - h(z))] \right)$$

#### 5.1.1 Производная логарифмической функции

Производная логарифмической функции имеет вид:

$$\frac{d}{dx} \ln(x) = \frac{1}{x}$$

Таким образом, в нашем случае, производная принимает следующий вид (положим  $h(z) = h$ )

$$\begin{aligned} \frac{\partial J}{\partial h} [y \cdot \log(h) + (1 - y) \cdot \log(1 - h)] &= y \cdot \frac{\partial J}{\partial h} [\log(h)] + (1 - y) \cdot \frac{\partial J}{\partial h} [\log(1 - h)] = \\ &= y \cdot \frac{1}{h} + (1 - y) \cdot \frac{1}{(1 - h)} \cdot \frac{\partial J}{\partial h} [1 - h] = \frac{y}{h} + \frac{1 - y}{1 - h} \cdot (0 - 1) = \\ &= \frac{y}{h} + \frac{y - 1}{1 - h} = \frac{y(1 - h) + h(y - 1)}{h(1 - h)} = \frac{y - yh + hy - h}{h(1 - h)} = \frac{y - h}{h(1 - h)} \end{aligned}$$

Вернув  $h = h(z)$ , получим

$$\frac{\partial J}{\partial h} = \frac{y - h(z)}{h(z)(1 - h(z))}$$

### 5.1.2 Производная сигмоиды

Заметим, что

$$\frac{d}{dx} e^x = e^x$$

Перепишем выражение

$$\frac{\partial h}{\partial z} \left[ \frac{1}{1 + e^{-z}} \right] = \frac{\partial h}{\partial z} [(1 + e^{-z})^{-1}]$$

По chain rule

$$\begin{aligned} &= -(1 + e^{-z})^{-2} \cdot (-e^{-z}) = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}} = \frac{1}{1 + e^{-z}} \cdot \frac{(1 + e^{-z}) - 1}{1 + e^{-z}} = \\ &= \frac{1}{1 + e^{-z}} \cdot \left( \frac{1 + e^{-z}}{1 + e^{-z}} - \frac{1}{1 + e^{-z}} \right) = \frac{1}{1 + e^{-z}} \cdot \left( 1 - \frac{1}{1 + e^{-z}} \right) \end{aligned}$$

В терминах предложенной выше нотации получается

$$\frac{\partial h}{\partial z} = h(z)(1 - h(z))$$

### 5.1.3 Производная линейной функции

Наконец найдем производную линейной функции.

$$\frac{\partial z}{\partial \theta} [X\theta] = X$$

Перемножим частные производные и найдем градиент.

$$\begin{aligned} &\nabla_{\theta} J \left( -\frac{1}{n} [y \cdot \log(h(z)) + (1 - y) \cdot \log(1 - h(z))] \right) \\ &= -\frac{1}{n} \left[ \frac{\partial J}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial \theta_j} \right] = -\frac{1}{n} \left[ \frac{y - h(z)}{h(z)(1 - h(z))} \cdot h(z)(1 - h(z)) \cdot X \right] = \end{aligned}$$

$$= -\frac{1}{n}[(y - h(z)) \cdot X] = [X^T \cdot (h(z) - y)] \times \frac{1}{n}$$

Поскольку  $z = X\theta$ ,

$$\nabla_{\theta} J = [X^T \cdot (h(X\theta) - y)] \times \frac{1}{n}$$

## 5.2 Реализация логистической регрессии

Обычно логистическая регрессия используется для решения задачи классификации, но в нашем случае необходимо найти сами коэффициенты, а не просто отнести объект к одному из классов. В связи с этим было принято решение реализовать свой класс логистической регрессии.

```

1 class LogReg():
2     def __init__(self, max_iter=15000, learning_rate=0.01, tol=0.00001):
3         self.thetas = None
4         self.max_iter = max_iter
5         self.learning_rate = learning_rate
6         self.tol = tol
7
8     def fit(self, x, y, a):
9         x, y = x.copy(), y.copy()
10        self.add_ones(x)
11        thetas = np.zeros(x.shape[1])
12        for t in range(1, self.max_iter + 1):
13            y_pred = self.h(x, thetas)
14            grad = self.gradient(x, y, y_pred, n)
15            thetas -= self.learning_rate * grad
16            self.thetas = thetas
17            if self.loss(y, y_pred) < self.tol:
18                break
19
20    def predict(self, x):
21        x = x.copy()
22        self.add_ones(x)
23        z = np.dot(x, self.thetas)
24        probs = np.array([expit(value) for value in z])
25        return probs

```

Листинг 2: Основные функции класса логистической регрессии

```

1  def add_ones(self, x):
2      return x.insert(0, 'x0', np.ones(x.shape[0]))
3
4  def loss(self, y, y_pred):
5      y_one_loss = y * np.log(y_pred + 1e-9)
6      y_zero_loss = (1 - y) * np.log(1 - y_pred + 1e-9)
7      return -np.mean(y_zero_loss + y_one_loss)
8
9  def h(self, x, thetas):
10     z = np.dot(x, thetas)
11     return np.array([expit(value) for value in z])
12
13 def gradient(self, x, y, y_pred, n):
14     return np.dot(x.T, (y_pred - y)) / n

```

Листинг 3: Вспомогательные функции класса логистической регрессии

## 6 Эксперименты

### 6.1 Метрика качества

Для оценки качества моделей была использована метрика *относительной средней абсолютной ошибки* ( $RMAE$ ), которая вычисляется по формуле:

$$RMAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{\sum_{i=1}^N \hat{y}_i},$$

где  $N$  - размер тестовой выборки,  $\hat{y}_i$  - истинное значение,  $y_i$  - предсказанное значение.

Все данные были разбиты на обучающую и тестовую выборки в хронологическом порядке. Также качество замерялось для разных пропорций разбиения: доля обучающей выборки варьировалась от 5 до 95%.

### 6.2 Логистическая регрессия v1

Параметры обучения:

- Максимальное кол-во итераций(max\_iter): 15000
- Скорость обучения(learning\_rate): 0.01
- Критерий остановки(tolerance): 0.0001



Рис. 6.1: Результат обучения логистической регрессии v1

Довольно неплохой результат, однако есть несколько моментов:

- 1 Большая разница между максимальным и минимальным значениями ошибки - около 80 п.п.
- 2 Рост ошибки при увеличении объема обучающей выборки
- 3 В любом случае большая ошибка

### 6.3 Логистическая регрессия v2

Данная версия отличается алгоритмом оптимизации обновления весов - вместо постоянного шага использовался метод Adam.

Параметры обучения:

- Максимальное кол-во итераций(max\_iter): 15000
- Скорость обучения(learning\_rate): 0.01
- Критерий остановки(tolerance): 0.0001
- Параметры Adam:  $\beta_1=0.9$ ,  $\beta_2=0.999$ ,  $\epsilon=1e-8$

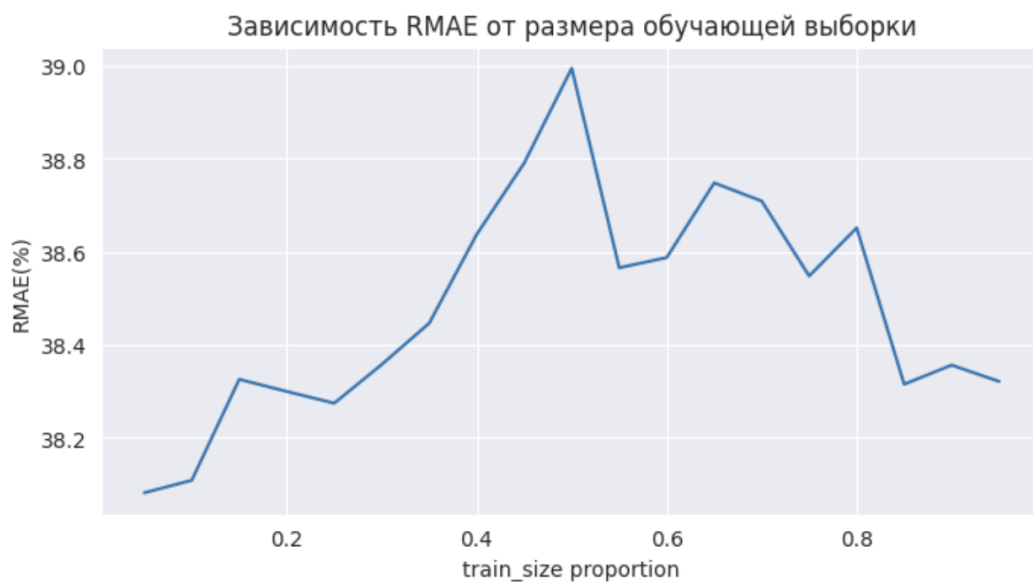


Рис. 6.2: Результат обучения логистической регрессии v2

Отличный результат - все проблемы, присутствующие в прошлой модели были решены.



## 6.4 Нейронная сеть и логистическая регрессия

В данном случае обучение происходило в 2 этапа:

- Обучение нейросети для получения параметра  $a$  ( $a = -2e$ , где  $e$  - выход нейросети)
- Обучение логистической регрессии с фиксированным значением параметра  $a$  и обновлением только параметра  $b$

Данный способ не удалось довести до хорошего качества.

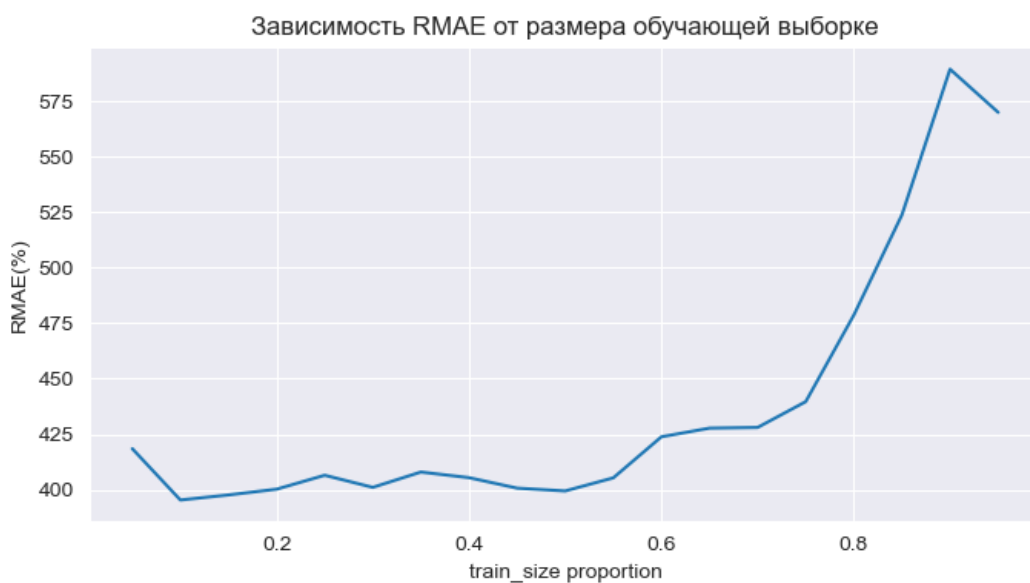


Рис. 6.3: Результат обучения нейронной сети и логистической регрессии

## 7 Заключение

### 7.1 Выводы

В рамках данной курсовой работы была рассмотрена задача оптимизации распределения маркетингового бюджета с использованием методов машинного обучения. Основное внимание было уделено реализации первой части двухэтапного подхода — построению модели прогнозирования рыночного отклика на маркетинговые стимулы.

В процессе работы был проведен обзор традиционных методов оценки маркетинговой эффективности, выявлены их преимущества и ограничения. На основании анализа современных исследований был выбран подход, сочетающий полносвязные нейронные сети и логистические модели спроса, что позволило учесть как нелинейные зависимости, так и сохранить интерпретируемость модели.

### 7.2 Перспективы дальнейших исследований

Основное направление развития проекта связано с реализацией полного цикла оптимизации распределения бюджета. В качестве перспективных направлений можно выделить:

- 1 Переход к обучению с подкреплением (RL): разработка решений, основанных на моделировании задачи как ограниченного процесса принятия решений Маркова (CMDP), что позволит учитывать долгосрочные эффекты маркетинговых акций на поведение пользователей.
- 2 Совместное обучение прогнозной модели и оптимизатора: вместо двухэтапного процесса можно изучить методы end-to-end оптимизации, напрямую оптимизирующие целевую бизнес-метрику под бюджетными ограничениями.

Дальнейшая реализация этих направлений позволит не только завершить построение эффективной системы распределения маркетингового бюджета, но и создаст интеллектуальные решения, способные адаптироваться к изменениям во внешней среде и максимизировать долгосрочную бизнес-ценность рекламных кампаний.

## Список литературы

- [1] Kui Zhao Junhao Hua, Ling Yan, Qi Zhang, Huan Xu и Cheng Yang. “A Unified Framework for Marketing Budget Allocation”. B: *A Unified Framework for Marketing Budget Allocation*. 2019.
- [2] Ziang Yan Shusen Wang Guorui Zhou Jingjian Lin Peng Jiang. “An End-to-End Framework for Marketing Effectiveness Optimization under Budget Constraint”. B: *arXiv preprint, arXiv:2302.0*. (2023).
- [3] Tianchi Cai Jiyan Jiang Wenpeng Zhang Shiji Zhou Xierui Song Li Yu Lihong Gu Xiaodong Zeng Jinjie Gu Guannan Zhang. “Marketing Budget Allocation with Offline Constrained Deep Reinforcement Learning”. B: *Marketing Budget Allocation with Offline Constrained Deep Reinforcement Learning*. 2023.