

Научно-исследовательский журнал «Modern Economy Success»

<https://mes-journal.ru>

2025, № 2 / 2025, Iss. 2 <https://mes-journal.ru/archives/category/publications>

Научная статья / Original article

Шифр научной специальности: 5.2.3. Региональная и отраслевая экономика (экономические науки)

УДК 339.138



<sup>1</sup> Коновалик Е.А.,  
<sup>1</sup> EPAM Systems

***К вопросу о применении Python и больших данных в задачах оптимизации  
инфлюенс-маркетинга компании: разработка индекса полезности инфлюенсера***

**Аннотация:** целью исследования является разработка авторского подхода к оптимизации инфлюенс-маркетинга на основе авторского индекса полезности блогера (BVI) и современных технологий анализа больших данных (Python, LLM).

**Методы:** в исследовании используются методы сбора и обработки больших объёмов данных из социальных сетей посредством Python (Pandas, API-запросы), алгоритмы многопоточного и асинхронного анализа данных, а также функции больших языковых моделей (LLM) для структурирования и интерпретации полученных результатов. Кроме них применялись общенаучные методы, сравнительный, описательный и обобщающий анализ.

**Результаты (Findings):** предложен индекс полезности блогера (BVI), который позволяет выявлять наиболее релевантных инфлюенсеров с учётом пересечения их аудиторий и целевой аудитории компании. Разработан программный код на Python, который автоматически рассчитывает BVI, формирует базу потенциально эффективных каналов и позволяет оценивать варианты сотрудничества. Показано, что использование больших данных, LLM и современных цифровых технологий для аналитики в инфлюенс-маркетинге обеспечивают расширение возможностей в поддержке принятия решений.

**Выводы:** полученные данные подтверждают эффективность разработанного индекса BVI и подходов к его вычислению, т.к. позволяет улучшить процесс для выбора блогеров, существенно сократить время- и трудо- затраты в инфлюенс-маркетинге. Предложенный подход с применением больших данных, Python и LLM расширяет возможности маркетинговых исследований, поскольку служит базисом для принятия экономически обоснованных решений с последующим ростом эффективности маркетинговых кампаний.

**Ключевые слова:** инфлюенс-маркетинг, индекс полезности блогера (BVI), аналитика, нейросети, большие данные, большие языковые модели (LLM), Python, оптимизация маркетинга, аудитория

**Для цитирования:** Коновалик Е.А. К вопросу о применении больших данных в задачах оптимизации инфлюенс-маркетинга компании: разработка индекса полезности инфлюенсера // Modern Economy Success. 2025. № 2. С. 360 – 377.

Поступила в редакцию: 11 декабря 2024 г.; Approved after reviewing: 8 февраля, 2025; Принята к публикации: 11 марта 2025 г.

<sup>1</sup> Yauheni Kanavalik,  
<sup>1</sup> EPAM Systems

***On the application of big data in optimizing a company's influencer  
marketing: development of an influencer utility index***

**Abstract:** the *purpose* of this study is to develop an original approach to optimizing influencer marketing based on the author's Blogger Utility Index (BVI) and modern data analysis technologies (Python, LLM).

**Methods:** the study employs methods for collecting and processing large volumes of data from social networks using Python (Pandas, API requests), algorithms for multi-threaded and asynchronous data analysis, as well as the

functions of large language models (LLM) for structuring and interpreting the obtained results. In addition, general scientific methods, including comparative, descriptive, and summarizing analysis, were applied.

**Findings:** the study proposes the Blogger Utility Index (BVI), which enables the identification of the most relevant influencers by considering the intersection of their audiences with the company's target audience. A Python-based software tool has been developed to automatically calculate BVI, compile a database of potentially effective channels, and assess collaboration options. The study demonstrates that the use of LLM and modern digital technologies for influencer marketing analytics enhances decision-making capabilities.

**Conclusions:** the obtained results confirm the effectiveness of the developed BVI index and the approaches to its calculation, as it improves the influencer selection process while significantly reducing time and labor costs in influencer marketing. The proposed approach, utilizing Python and LLM, expands the possibilities of marketing research by providing a foundation for making economically justified decisions, ultimately increasing the effectiveness of marketing campaigns.

**Keywords:** influencer marketing, Blogger Utility Index (BVI), analytics, neural networks, large language models (LLM), Python, marketing optimization, audience

**For citation:** Yauheni Kanavalik On the application of large language models (LLM) in optimizing a company's influencer marketing: development of an influencer utility index. Modern Economy Success. 2025. 2. P. 360 – 377.

The article was submitted: December 11, 2024; Approved after reviewing: February 8, 2025; Approved after reviewing: March 11, 2025

### Введение

В условиях динамичного развития рынка и существенного усложнения маркетинговых коммуникаций наибольшую ценность для бизнеса при принятии решений представляет аналитика. Аналитика в современном бизнесе – поддерживающий стратегическое развитие инструмент, идеи применения которого базируются на использовании комплекса современных технологий, предоставляющих возможность обрабатывать огромные объемы данных и выявлять различные закономерности, оценивать эффективность бизнес-процессов, а также многое другое. Вместе с тем, беспрецедентные темпы роста рынка искусственного интеллекта (ИИ) и появление различных его конфигураций (типовые алгоритмы, нейросети, чат-боты, большие языковые модели – LLM и др.), вкупе с растущим значением технологий больших данных, обусловили трансформацию практики и подходов к проведению аналитики. В частности, экстраполируя все вышесказанное на специфику маркетинговых исследований, все большее значение уделяется вопросам синергии преимуществ человека, потенциала его креативности и идей со средствами оптимизации, представленными интеллектуальной аналитикой и различными алгоритмическими решениями. Стоит признать, что

маркетинг представленные трансформационные процессы затронули в первую очередь – сегодня прослеживаются растущие роль и значение современных цифровых технологий в аналитике и маркетинге как таковом. Поскольку ценность представленных инструментов и технологий не вызывает сомнений, научно-теоретический и практический интерес представляет разработка прикладных решений для поддержки и повышения эффективности маркетинговых задач компании (поскольку современный рынок ограничен на предмет их доступности и открытого обоснования принципов работы).

Актуальным направлением современного маркетинга является инфлюенс-маркетинг, объемы и динамика которого указывают на стремительную популяризацию инфлюенс-маркетинга в бизнесе (рис. 1). Инфлюенс-маркетинг предполагает проведение работы с инфлюенсерами – блогерами, известными личностями, аккумулирующими вокруг себя определенную целевую аудиторию [1, 3, 6]. Именно аудитория инфлюенсера представляет интерес с точки зрения компании и её маркетинга, т.к. влияние на подписчиков и их привлечение способствуют достижению искомых целей маркетинговой активности.

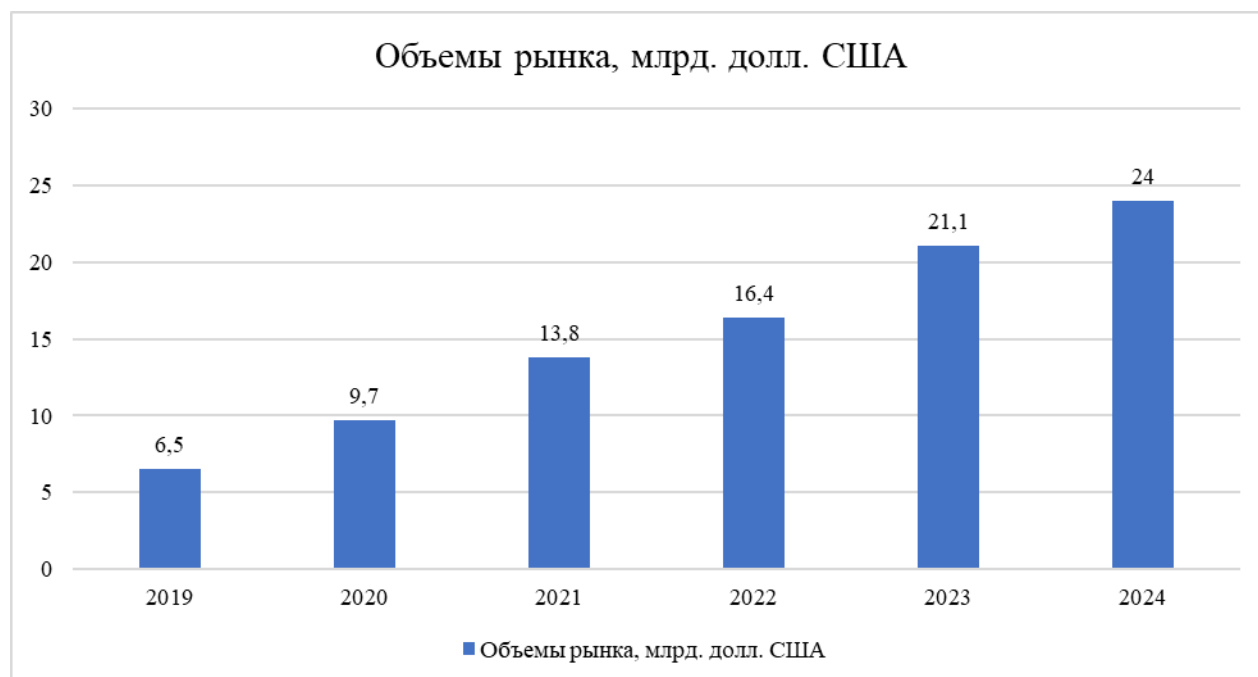


Рис. 1. Объемы и динамика рынка инфлюенс-маркетинга в мире, составлено по данным [1, 9].

Fig. 1. Volumes and dynamics of the global influencer marketing market, compiled based on [1, 9].

Опираясь на представленный рис. 1, заметим, что инфлюенс-маркетинг, действительно, выступает продуктивным каналом и способом продвижения. Тем не менее, в научной литературе выделяются не только преимущества и явные достоинства инфлюенс-маркетинга, но и его недостатки [3, 4, 6 и др.]. Ключевым из них становится сложность выбора подходящего компании инфлюенсера, от которого, фактически, зависит конечная экономическая эффективность и результаты маркетинга. Заявленное обстоятельство усугубляется тем, что компании требуется осуществлять контроль инфлюенсеров, проводить постоянную аналитику их работы, согласовывать любые действия и совершенствовать маркетинг, оперируя принципами стратегического партнерства, взаимного уважения, а также доверия (которое, тем не менее, подкрепляется контрактными обязательствами). Эффективность инфлюенс-маркетинга зависит от целой системы факторов-условий, и ключевое из них – правильный выбор инфлюенсеров, за которым последует более слаженная и скоординированная работа, устремленная на достижение конкретных целевых индикаторов.

Совмещая описанные ранее направления и тенденции популяризации технологий больших данных, нейросетей и LLM в современном маркетинге с идеями организации инфлюенс-маркетинга, отметим, что до сих пор открытыми остаются вопросы совместного применения указанных инструментов-каналов маркетинга для повышения эффективности и результативности в бизнесе –

достижения преимуществ оптимизации средствами технологий. Заметим, что инфлюенс-маркетинг может быть не только оптимизирован средствами современных цифровых технологий, но и выведен на определенную планку автоматизации, что невозможно в отсутствие выработки концептуальных подходов к оптимизации инфлюенс-маркетинга компании через разработку индекса полезности инфлюенсера и поддержку аналитики средствами цифровых технологий и больших данных. В комплексе все вышеперечисленное обуславливает объект и предмет настоящего исследования, отражающие инфлюенс-маркетинг и процессы его оптимизации средствами больших данных, Python и LLM соответственно.

#### Материалы и методы исследований

Исследование основано на открытых публикациях и статистических данных, в которых затрагиваются фундаментальные идеи, преимущества, проблемы и подходы к организации инфлюенс-маркетинга как в целом, так и с задействованием оптимизационных преимуществ больших данных, Python и LLM.

Исследование проводилось в несколько этапов. В ходе первого этапа была проработана общая концепция исследования, проведен анализ научной литературы по теме и выделены основные тезисы и обобщения на предмет оптимизации инфлюенс-маркетинга средствами больших данных в связке с Python и LLM. Преимущественно задействовали описательные, сравнительные и обобщающие методы, посредством которых при анали-

зе научных статей [3, 4, 6 и др.] могут быть выделены следующие преимущества-способы оптимизации инфлюенс-маркетинга через большие данные, Python и LLM, а именно: поддержка аналитики за счет предоставления прикладных инструментов проведения маркетинговых исследований, применение генеративной функции с созданием контента при работе с инфлюенсером, автоматизация, основанная на оценке работы инфлюенсеров и сравнительном выборе наиболее эффективных партнеров на основе больших данных.

Также в ходе первого этапа особое внимание уделялось литературе, посвященной вопросам применения LLM для работы с большими данными; на основании ряда исследований [2, 5 и др.] можно сделать выводы о том, что LLM и большие данные становятся продуктивным средством аналитики и поддержки работы, функции которого позволяют собирать и обрабатывать информацию, представленную в различных вариациях (естественный язык, числовые данные и др.), классифицировать её, обобщать, вычленять отдельные смыслы, осуществлять интерактивный интеллектуальный поиск и т.п.

Кроме того, отдельное внимание уделялось исследованию современных решений, позволяющих выполнять схожие или идентичные запросы в аналитике инфлюенс-маркетинга средствами больших данных, LLM, ИИ или иных нейросетевых алгоритмов. В числе перспективных технологий видятся: SOMONITOR, предназначенный для проведения комплексного анализа маркетинговых данных с использованием объяснимого ИИ и LLM – разработан А. Farseev, Q. Yang, M. Ongpin, I. Gossoudarev, Y.-Y. Chu-Farseeva, S. Nikolenko (функции алгоритма, которые выделяются авторами: прогнозирование кликабельности, изучение востребованного контента, оптимизация прибыли и др.) [7]; INFLECT-DGNN, принципы работы которого основаны на использовании динамических графовых нейронных сетей для прогнозирования влияния пользователей на аудиторию – разработан Е. Tiukhova, Е. Penaloza, М. Óskarsdóttir, В. Baesens, М. Snoeck, С. Bravo [11]; SAGraph – граф рекламных кампаний, собранный из данных платформы Weibo, предназначен для определения ключевых инфлюенсеров и изучения контента в социальных сетях – представлен в исследовании Х. Zhang, Х. Chen, Y. Liu, J. Wang, Z. Hu, R. Yan. [12]; Influencer – AI-инструмент для создания промо-контента, позволяющий создавать рекламные посты, заголовки, изображения, контекстно-зависимые рекомендации, возможности кастомизации в интерактивном формате – описан Х. Liu, А. Sun, Р. An, Т. Ma, J. Zhao [10]; а также приме-

нение маркетинговых запросов с существующими LLM-решениями, которые позволяют обрабатывать множество данных и работать над усилением маркетинга, что описано Y. Gao, S.K. Arava, Y. Li, J.W. Snyder Jr. [8].

Учитывая существующие алгоритмы и их функционально-целевое предназначение, а также выделенные особенности, проблемы и специфику инфлюенс-маркетинга, были поставлены общие задачи последующего проведения исследования.

В частности, на втором этапе проводимого исследования с учетом выделенных положений и проведенного анализа был разработан авторский Индекс полезности блогера (BVI – Blogger Value Index), сочетающий в себе соотношение количества подписчиков и общее количество подписчиков целевого канала с применением корректирующего коэффициента. Данный коэффициент был предложен впервые и выступил результатом деятельности на базе компании ruzhanyfarm.by – Ружанская Ферма; работа с индексом позволила отбирать эффективных инфлюенсеров для организации рекламных кампаний.

В ходе третьего этапа предложенный авторский индекс BVI был оптимизирован и выведен на уровень автоматического применения с использованием больших данных, средств Python и LLM; была осуществлена апробация и вычисление индекса, на основании чего сделаны выводы о степени его применимости и полезности в инфлюенс-маркетинге.

Таким образом, целью исследования выступила разработка индекса полезности инфлюенсера и раскрытие возможных способов оптимизации задач инфлюенс-маркетинга с использованием данного индекса в связке с инструментами современных цифровых технологий (большие данные, Python и LLM).

### Результаты и обсуждения

Перенасыщение рынка маркетинговыми предложениями и растущая ценность инфлюенс-маркетинга в совокупности с стремительными темпами интеграции технологий больших данных и LLM во все сферы человеческой жизни обусловили потребность в разработке интегрально объединяющего данные факторы инструмента оптимизации инфлюенс-маркетинга. В качестве такого инструмента выступил авторский Blogger Value Index, предназначенный для оценки полезности блогера (лидера мнений, владельца профиля в социальной сети) для бизнеса с точки зрения соответствия аудитории компании, обращенной к конкретному блогу (инфлюенсеру). Сегодня общераспространенным трендом работы с инфлюенсерами выступает ориентация на более малые сегменты

инфлюенсеров – нано- и микро- инфлюенсеров, подписчики которых отличаются большей вовлеченностью, лояльностью и тесными взаимоотношениями с инфлюенсером [3; 4]. Как итог, эффективность рекламы и таких инфлюенсеров оказывается несколько выше.

Тем не менее, открытым остается вопрос поиска инфлюенсеров, что в целом остается одной из наиболее время- и трудо- затратных частей инфлюенс-маркетинга. И именно в целях решения данной задачи, а также последующей оптимизации и повышения эффективности, был разработан упомянутый индекс BVI.

Blogger Value Index – специальный коэффициент, предназначенный для оценки полезности блогера для предприятия, высчитывается на базе аудитории, читающей (подписанной на) инфлюенсера. Индекс построен на соотношении следующих переменных:

$T$  – общее количество подписчиков целевого профиля;

$I_i$  – количество подписчиков, которые одновременно следят за конкурентом и целевым профилем;

$n$  – количество конкурентов, которые анализируются;

На основании соотношения представленных значений был выведен BVI, расчет которого проводится по следующей формуле: (1)

$$BVI_{index} = \frac{\sum_{i=1}^n I_i}{T} \quad (1)$$

Опираясь на представленную формулу, отметим, что значение  $I_i$  определяется пересечением двух множеств: (2)

$$I_i = |S_i \cap T| \quad (2)$$

где:

$S_i$  – множество подписчиков конкурента  $i$ ;

$T$  – общее количество подписчиков целевого канала.

В целях упрощения работы с данными приме-

няется корректирующий коэффициент; его введение не обязательно, однако улучшает процесс анализа. В качестве значения такого коэффициента применяется  $K = 10000$ , где  $K$  – принятое количество подписчиков для инфлюенсера.

Таким образом, с учетом объединения представленных формул и параметров формируется следующая формула BVI: (3)

$$BVI_{index} = \frac{\sum_{i=1}^n |S_i \cap T|}{T} \times K \quad (3)$$

где:

$T$  – общее количество подписчиков целевого профиля;

$I_i$  – количество подписчиков, которые одновременно следят за конкурентом и целевым профилем;

$n$  – количество конкурентов, которые анализируются;

$K$  – принятое среднее количество подписчиков инфлюенсера.

Отметим, что для расчета значений предложенного авторского коэффициента BVI важно собирать реальные данные на предприятии.

Сбор данных возможно оптимизировать за счет использования средств современных цифровых технологий, а именно через разработку программного кода средствами Python и Pandas + Instagram\* API (\*принадлежит компании Meta, признанной экстремистской и запрещенной на территории Российской Федерации). На базе предприятия Ружанская Ферма был проведен необходимый анализ и сбор данных. Для сбора данных были использованы 10 конкурентов Ружанской Фермы, совокупный объем подписчиков которых составил 35 тыс. чел.

Общий процесс анализа с применением Python и Pandas + Instagram\* API (\*принадлежит компании Meta, признанной экстремистской и запрещенной на территории Российской Федерации) раскрыт на рис. 2:

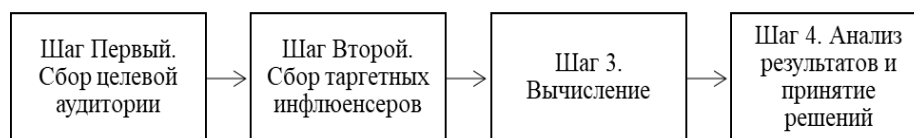


Рис. 2. Алгоритм использования BVI средствами современных технологий и программирования.

Fig. 2. Algorithm for using BVI through modern technology and programming.

Раскроем содержание каждого шага более подробно.

Шаг Первый. Сбор целевой аудитории. В ходе данного этапа средствами Python формируется уникальный ID каждого подписчика каждого из конкурентов. Таким образом множество  $S_i$  наполняется, что потребуется в дальнейшей работе.

Библиотеки для дальнейшего использования импортируются.

Далее объявляется функция, которая на основе Rapid API выгрузит публичную информацию о подписчиках для данного конкурента  $i$ .

Объявляется функция `get_followers`. Функция принимает имя пользователя конкурента, и используя стандартный `http`, клиент `python3` скачивает публичную информацию о подписчиках этого

клиента, после чего возвращает назад. Подписчиков может быть достаточно много, в связи с чем был добавлен `tqdm` в целях визуализации прогресса (рис. 3).

```
def get_followers(username, amount_per_request=1000, pages_limit=10):
    conn = http.client.HTTPSConnection("api_host")

    rapid_api_headers = {
        'x-rapidapi-key': "unique_key",
        'x-rapidapi-host': "api_host"
    }

    followers = []
    pagination_token = None

    with tqdm(total=pages_limit, desc=f"Fetching Pages for {username}") as progress_bar:
        for page in range(pages_limit):
            # Build the URL with a pagination token if available
            if pagination_token:
                url = f"/v1/followers?username_or_id_or_url={username}&amount={amount_per_request}&pagination_token={pagination_token}"
            else:
                url = f"/v1/followers?username_or_id_or_url={username}&amount={amount_per_request}"

            # Make the request
            conn.request("GET", url, headers=rapid_api_headers)
            res = conn.getresponse()
            data = res.read()

            # Parse the response JSON
            parsed_data = json.loads(data)

            # Append the followers from the current page
            if "data" in parsed_data and "items" in parsed_data["data"]:
                followers.extend(parsed_data["data"]["items"])

            # Check if there's a next page
            pagination_token = parsed_data.get("pagination_token")
            if not pagination_token:
                break # No more pages to fetch

            progress_bar.update(1)

    conn.close()

    return {
        "followers": followers,
        "followers_number": len(followers),
        "metadata": {
            "requests_made": page + 1
        }
    }
```

Рис. 3. Python-скрипт для получения подписчиков пользователя из API с использованием HTTP-запросов.

Fig. 3. Python script for retrieving user followers from the API using HTTP requests.

Чтобы получить информацию для каждого из  $i = 10$  конкурентов, используется следующая функция (рис. 4):



```
def merge_followers(accounts):
    all_followers = set() # Use a set to automatically handle unique
    usernames

    for account in accounts:
        try:
            followers_data = get_followers(account, pages_limit=10)

            for follower in followers_data['followers']:
                # Extract the username from each follower
                all_followers.add(follower['username'])

        except Exception as e:
            print(f"Error processing {account}: {e}")

    return list(all_followers)

accounts = ["competitor_username_1", "competitor_username_2",
            "competitor_username_n", "competitor_username_10"]

unique_followers = merge_followers(accounts)
```

Рис. 4. Программный код для сбора уникальных подписчиков из нескольких аккаунтов с помощью API-запросов, разработано автором.

Fig. 4 Program code for collecting unique followers from multiple accounts using API requests.

После `unique_followers` будет содержать около 35000 уникальных имен пользователей, которые подписаны на одного из конкурентов, что и отражает искомое множество – целевую аудиторию компании.

Шаг второй. Сбор таргетных инфлюенсеров.

Множество  $T$  собирается из множества всех подписок каждого уникального пользователя целевой аудитории  $S$ . Согласно математическому преобразованию: (4)

$$T = \bigcup_{u \in S} F(u) \quad (4)$$

где:

$u$  – уникальный пользователь;

$F(u)$  – подписки конкретного пользователя  $u$ ;

$\bigcup_{u \in S}$  – объединение множеств.

Так, для каждого пользователя  $u \in S$  будет взято множество его подписок  $F(u)$ . Все множества  $F(u)$  будут объединены для каждого  $u$  в  $S$ , что позволит получить результирующее множество  $T$ , в котором содержатся уникальные подписки всех пользователей целевой аудитории.

Объявляем функцию для обработки каждого конкретного пользователя  $u$ . Функция ниже (рис. 5) принимает уникальный ID пользователя и несколько необязательных параметров для оптимизации собственной работы. Используя API, она идет по всем подпискам уникального пользователя, в некоторых случаях, когда подписок особенно много, используется механизм постраничного итерирования, чтобы получить максимально полный список. После того, как функция завершила свою работу, она возвращает список наружу, который фактически является подмножеством  $T$  для  $u$ .

```
def get_followings(username, amount_per_request=1000, pages_limit=2):
    conn = http.client.HTTPSConnection("api_host")

    rapid_api_headers = {
        'x-rapidapi-key': "api_key",
        'x-rapidapi-host': "api_host"
    }

    get_followings = []
    pagination_token = None

    followings = []

    for page in range(pages_limit):
        # Build the URL with a pagination token if available
        if pagination_token:
            url = f"/v1/following?username_or_id_or_url={username}&amount={amount_per_request}&pagination_token={pagination_token}"
        else:
            url = f"/v1/following?username_or_id_or_url={username}&amount={amount_per_request}"

        # Make the request
        conn.request("GET", url, headers=rapid_api_headers)
        res = conn.getresponse()
        data = res.read()

        # Parse the response JSON
        parsed_data = json.loads(data)

        # Append the followers from the current page
        if "data" in parsed_data and "items" in parsed_data["data"]:
            followings.extend(parsed_data["data"]["items"])

        # Check if there's a next page
        pagination_token = parsed_data.get("pagination_token")
        if not pagination_token:
            break # No more pages to fetch

    conn.close()

    return {
        "followings": followings,
        "followings_number": len(followings),
        "metadata": {
            "requests_made": page + 1
        }
    }
```

Рис. 5. Программный код для получения списка подписок (тех, на кого подписан пользователь) из API с постраничным запросом.

Fig. 5. Program code for retrieving a list of subscriptions (accounts a user follows) from the API with pagination.

Используя Python список всех полученных подмножеств объединяется в одно множество T. При невысокой скорости работы Rapid API ско-

рость работы целесообразно оптимизировать с использованием параллельных вычислений (рис. 6):



```
from concurrent.futures import ThreadPoolExecutor, as_completed
from tqdm import tqdm

user_followings = {}

max_users = 40000
users_to_process = unique_followers[:max_users]

def fetch_followings(username):
    try:
        followings_data = get_followings(username)
        return username, followings_data["followings"]
    except Exception as e:
        print(f"Error processing {username}: {e}")
        return username, []

# Limit the number of simultaneous requests to 40
max_workers = 40

with ThreadPoolExecutor(max_workers=max_workers) as executor:
    # Submit all tasks to the executor
    futures = {executor.submit(fetch_followings, username): username for
                username in users_to_process}

    # Use tqdm for progress tracking
    for future in tqdm(as_completed(futures), total=len(futures),
                      desc="Processing unique followers"):
        username, followings = future.result()
        user_followings[username] = followings
        print(f"Processed {username}: {len(followings)} followings")

print(f"\nCompleted processing {len(user_followings)} unique followers.")
```

Рис. 6. Программный код для многопоточного сбора подписок большого числа пользователей (до 40 тыс. чел.) с использованием параллельных вычислений.

Fig. 6. Program code for multithreaded collection of subscriptions from a large number of users (up to 40,000 people) using parallel computing.

Впоследствии полученные данные обрабатываются для более удобной работы и загружаются в

библиотеку для работы с большими данными pandas (рис. 7):

```
import pandas as pd

data = []

# Iterate through the dictionary
for user, followings in user_followings.items():
    for following in followings:
        try:
            data.append({'user': user, 'follows': following['username']})
        except (KeyError, TypeError):
            print(f"Skipping invalid entry for user {user}: {following}")

# Create the DataFrame for later analysis
df = pd.DataFrame(data)
```

Рис. 7. Программный код для создания DataFrame в Pandas на основе данных о подписках пользователей.

Fig 7. Program code for creating a Pandas DataFrame based on user subscription data.

Опираясь на приведенное множество T, был проведен беглый анализ, что позволило выявить, на что подписана аудитория (рис. 8):



Рис. 8. Каналы, на которые подписана аудитория бизнеса.

Fig. 8. Channels followed by the business audience.

Как можно заметить, наибольшие по масштабу каналы выступили лидерами рейтинга ввиду своей размерности; именно в целях нивелирования данного недостатка применяется BVI, который позволяет нивелировать размер аудитории каналов и вывести «вверх» именно те каналы, в которых превалирует доля целевой аудитории бизнеса S.

Шаг третий. Вычисление BVI. После сбора всех необходимых данных вычисляется индекс. Для данной задачи объявляются функции для работы с сохраненным файлом ранее проведенной работы (рис. 9):

```
def save_df_to_csv(df, file_path):
    try:
        df.to_csv(file_path, index=False)
        print(f"DataFrame saved to {file_path}")
    except Exception as e:
        print(f"Error: {e}")

def load_df_from_csv(file_path):
    try:
        df = pd.read_csv(file_path)
        print(f"DataFrame loaded from {file_path}")
        return df
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        return None
    except pd.errors.ParserError:
        print(f"Error: Could not parse the CSV file '{file_path}'. Check the file format.")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")
        return None
```

Рис. 9. Функции для работы с CSV-файлами – сохранение DataFrame в CSV и загрузка DataFrame из CSV.  
Fig. 9. Functions for working with CSV files – saving a DataFrame to CSV and loading a DataFrame from CSV.

Впоследствии необходимые файлы загружаются с диска (рис. 10). В целях улучшения выборки из числа были исключены каналы, на которые

подписано недостаточное количество целевой аудитории бизнеса; каналы с аудиторией менее 30 человек из 30000 были выведены из выборки.

```
import pandas as pd

file_path = '/content/drive/.../audience.csv'

df = load_df_from_csv(file_path)
```

Рис. 10. Программный код для загрузки DataFrame из CSV-файла, разработано автором.  
Fig. 10. Program code for loading a DataFrame from a CSV file, developed by the author

Объявляется функция для получения информации о каждом из каналов. Чтобы компенсировать

нестабильность API, добавляются перезапросы, и ожидание (рис. 11):

```
import http.client
import json

def get_instagram_info(username, retries=3, delay=2):
    conn = http.client.HTTPSConnection("host_api")
    headers = {
        'x-rapidapi-key': "api_key",
        'x-rapidapi-host': "host_api"
    }

    for attempt in range(1, retries + 1):
        try:
            if attempt > 1:
                print(f"Attempt {attempt} to fetch data for username: {username}")

            conn.request("GET", f"/v1/info?username_or_id_or_url={username}&include_about=true", headers=headers)
            res = conn.getresponse()
            data = res.read()

            decoded_data = data.decode("utf-8") # Decode the bytes object
            to a string
            json_data = json.loads(decoded_data) # Parse the JSON string
            into a Python dictionary

            # Check if an error exists in the response
            if 'error' in json_data.get('data', {}):
                print(f"Error in response: {json_data['data']['error']}")
                raise ValueError("API returned an error.")

            if 'data' in json_data:
                return json_data['data'] # Return the parsed JSON data

            return None # Return None if the 'data' field is missing

        except json.JSONDecodeError:
            print("Error: Invalid JSON response from the API.")
        except ValueError as ve:
            print(f"Error: {ve}")
        except Exception as e:
            print(f"Unexpected error: {e}")

        # If not the last attempt, wait before retrying
        if attempt < retries:
            time.sleep(delay)

    print("Failed to fetch data after multiple attempts.")

    return None
```

Рис. 11. Программный код для сбора информации о пользователе через API.  
Fig. 11. Program code for retrieving user information via API.

Далее выполняется следующая функция для каждого из каналов, который попал в множество каналов T, и на который подписано по крайней мере 30 человек из 30000 полученных пользователей множества S (рис. 12):

```
def process_row(row):
    username = row['name']

    # Check if the username is empty or None
    if not username:
        return {"error": f"Username is empty or None for row: {row}"}

    try:
        info = get_instagram_info(username)
        if info:
            # Combine existing row data with the Instagram data
            return {**row, **info}
        else:
            return {"error": f"Error retrieving information for {username}"}
    except Exception as e:
        return {"error": f"An error occurred processing {username}: {str(e)}"}

from concurrent.futures import ThreadPoolExecutor, as_completed
from tqdm import tqdm
import time

# Create an empty list to store the results
results = []

with ThreadPoolExecutor(max_workers=3) as executor:
    # Submit tasks for parallel execution
    futures = {executor.submit(process_row, row.to_dict()): idx for idx, row in df.iterrows()}

    # Use tqdm to display progress
    for future in tqdm(as_completed(futures), total=len(futures), desc="Processing rows"):
        result = future.result()
        if result:
            results.append(result)

# Save results to a JSON file
output_file = "/content/drive/_/influences_raw_data.json"
with open(output_file, "w") as f:
    json.dump(results, f, indent=4)

print(f"Results saved to {output_file}")
```

Рис. 12. Программный код для асинхронного получения информации о пользователях.  
Fig. 12. Program code for asynchronous retrieval of user information.

Асинхронное вычисление, представленное на рис. 12, применялось в целях повышения скорости работы API, что повысило стабильность ответов API. Также используется tqdm, что позволяет мо-

ниторить ход работы процесса выполнения программы. Полезной практикой становится проверка количества ошибок при выполнении процесса (рис. 13):

```
errors = 0;
length = 0;

for item in data:
    length += 1;

    if 'error' in item:
        errors += 1;

print(f"{errors} of errors out of {len(data)} entities. {errors / len(data) * 100}%")
```

Рис. 13. Программный код для оценки ошибок при выполнении процесса.

Fig. 13. Program code for evaluating errors during process execution.

На основании извлеченных данных было получено 18 ошибок из 4139 запросов, или 0,4348% ошибок, что достаточно допустимо с учетом вы-

борки; было потеряно 0,4% данных, что не должно сказаться на результатах анализа.

```
import pandas as pd
import json

def load_json_to_dataframe(file_path):
    try:
        with open(file_path, 'r') as f:
            data = json.load(f)
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        return None
    except json.JSONDecodeError:
        print(f"Error: Invalid JSON format in '{file_path}'.")
        return None

    # Extract required columns and handle missing keys gracefully
    rows = []
    for item in data:
        if 'error' in item:
            continue
        else:
            row = {}
            row['name'] = item.get('name', None)

            if 'about' in item and isinstance(item['about'], dict):
                row['about.username'] = item['about'].get('username', None)
                row['about.country'] = item['about'].get('country', None)
            else:
                row['about.username'] = None
                row['about.country'] = None

            row['full_name'] = item.get('full_name', None)
            row['audience'] = item.get('audience', None)
            row['url'] = item.get('url', None)
            row['follower_count'] = item.get('follower_count', None)
            row['account_type'] = item.get('account_type', None)
            row['is_business'] = item.get('is_business', None)

            row['biography'] = item.get('biography', None)
            row['page_name'] = item.get('page_name', None)

            rows.append(row)

    df = pd.DataFrame(rows)
    return df

file_path = "/content/drive/_/influences_raw_data.json"
df = load_json_to_dataframe(file_path)
```

Рис. 14. Программный код для подготовки информации об инфлюенсерах для дальнейших вычислений.

Fig. 14. Program code for preparing influencer information for further calculations.



После подготовки данных осуществляется расчет данных (рис. 15) и значения индекса сортируются по каналам так, чтобы определить каналы с наибольшей целевой аудиторией. Таким образом

будут выделены самые эффективные каналы продвижения среди рассматриваемой группы нано-инфлюенсеров. Полученные данные выгружаются (рис. 16).



```
df['bv_index'] = df['audience'] / df['follower_count'] * 10000
```

Рис. 15. Программный код для вычисления BVI для каждого инфлюенсера.

Fig. 15. Program code for calculating BVI for each influencer.



```
import pandas as pd
from google.colab import files

if df is not None: # Check if the DataFrame was loaded successfully
    # Save the DataFrame to a CSV file
    csv_file_name = "/content/drive/.../influences_raw_data.csv"
    df.to_csv(csv_file_name, index=False)

    # Download the CSV file
    files.download(csv_file_name)
else:
    print("DataFrame is empty or could not be loaded. Cannot download CSV file.")
```

Рис. 16. Программный код для сохранения DataFrame в CSV-файл.

Fig. 16. Program code for saving a DataFrame to a CSV file.

Впоследствии собранные данные обрабатываются через программы excel или google doc, для удобной визуализации. В случае проводимого исследования сформировалась таблица из 4 тыс. ка-

налов, которые представляют интерес для сотрудничества в контексте инфлюенс-маркетинга. Данные каналы были отсортированы по убыванию BVI (рис. 17):



	A	C	D	E	F	G	J	L	M
1	name	about.count	full_name	audience	url	follower_count	bv_index	classificatio	category
2	fan		МЯС		30	<a href="https://www.instagram.com/fan_mяс">https://www.instagram.com/fan_mяс</a>	698	429.7994269	Pet Food Retail
3	dog		Дог Л		31	<a href="https://www.instagram.com/dog_l">https://www.instagram.com/dog_l</a>	722	429.3628809	Business
4	bfc	Belarus	БФКЕ		45	<a href="https://www.instagram.com/bfc_бфке">https://www.instagram.com/bfc_бфке</a>	1108	406.1371841	Dog Sports Federation
5	pris	Belarus	Лакон		38	<a href="https://www.instagram.com/pris_лак">https://www.instagram.com/pris_лак</a>	958	396.6597077	Business
6	lakc		НАТУ	ТАКО	34	<a href="https://www.instagram.com/lakc_нату">https://www.instagram.com/lakc_нату</a>	927	366.7745415	Business
7	spe		ОШЕ		30	<a href="https://www.instagram.com/spe_оше">https://www.instagram.com/spe_оше</a>	831	361.0108303	Business
8	jum		Jump		33	<a href="https://www.instagram.com/jum_jump">https://www.instagram.com/jum_jump</a>	944	349.5762712	Business
9	zoo	Belarus	ZOO	ЗИН	52	<a href="https://www.instagram.com/zoo_zin">https://www.instagram.com/zoo_zin</a>	1570	331.2101911	Business
10	spo	Belarus	КУ		38	<a href="https://www.instagram.com/spo_ku">https://www.instagram.com/spo_ku</a>	1185	320.6751055	Business
11	hap	Belarus	ЛАКО	лару	142	<a href="https://www.instagram.com/hap_лако">https://www.instagram.com/hap_лако</a>	4462	318.2429404	Business
12	toto	Belarus	Toto	Минск	46	<a href="https://www.instagram.com/toto_minsk">https://www.instagram.com/toto_minsk</a>	1462	314.6374829	Business
13	fron	Belarus	Нату		51	<a href="https://www.instagram.com/fron_нату">https://www.instagram.com/fron_нату</a>	1623	314.232902	Business
14	pan	Belarus	"Пан	ак и к	55	<a href="https://www.instagram.com/pan_пан">https://www.instagram.com/pan_пан</a>	1804	304.8780488	Business
15	dog	Belarus	КИНС		90	<a href="https://www.instagram.com/dog_kins">https://www.instagram.com/dog_kins</a>	3001	299.9000333	Business
16	parl	Ukraine	Party		61	<a href="https://www.instagram.com/parl_party">https://www.instagram.com/parl_party</a>	2077	293.6928262	Business
17	pho	Belarus	ФОТО	звени	31	<a href="https://www.instagram.com/pho_foto">https://www.instagram.com/pho_foto</a>	1077	287.8365831	Personal
18	per	Belarus	Анат		111	<a href="https://www.instagram.com/per_anat">https://www.instagram.com/per_anat</a>	3950	281.0126582	Business
19	nati	Belarus	Нату	Минск	32	<a href="https://www.instagram.com/nati_нату">https://www.instagram.com/nati_нату</a>	1139	280.9482002	Business
20	tp.d	Belarus	Нату		100	<a href="https://www.instagram.com/tp.d_нату">https://www.instagram.com/tp.d_нату</a>	3560	280.8988764	Business
21	bra	Belarus	ШЛЕ		88	<a href="https://www.instagram.com/bra_шле">https://www.instagram.com/bra_шле</a>	3147	279.631395	Business
22	zoo	Belarus	Зоом		35	<a href="https://www.instagram.com/zoo_zoom">https://www.instagram.com/zoo_zoom</a>	1252	279.5527157	Business
23	iam	Belarus	Пр	й мед	31	<a href="https://www.instagram.com/iam_пр">https://www.instagram.com/iam_пр</a>	1110	279.2792793	Business
24	reh	Belarus	Реаб	нск	97	<a href="https://www.instagram.com/reh_реаб">https://www.instagram.com/reh_реаб</a>	3492	277.7777778	Business
25	can	Belarus	САМ		35	<a href="https://www.instagram.com/can_сам">https://www.instagram.com/can_сам</a>	1276	274.2946708	Business
26	plat	Belarus	Platin		65	<a href="https://www.instagram.com/plat_platin">https://www.instagram.com/plat_platin</a>	2418	268.8172043	Business
27	pet	Belarus	Итан	ь	31	<a href="https://www.instagram.com/pet_итан">https://www.instagram.com/pet_итан</a>	1159	267.4719586	Business
28	deli	Belarus	Лакон		34	<a href="https://www.instagram.com/deli_лак">https://www.instagram.com/deli_лак</a>	1293	262.9543697	Business
29	bob	Belarus	НАТУ	САК	82	<a href="https://www.instagram.com/bob_нату">https://www.instagram.com/bob_нату</a>	3127	262.2321714	Business
30	cryp	Belarus	Crypt		36	<a href="https://www.instagram.com/cryp_crypto">https://www.instagram.com/cryp_crypto</a>	1386	259.7402597	Business
31	tuzi	Belarus	Оде		58	<a href="https://www.instagram.com/tuzi_оде">https://www.instagram.com/tuzi_оде</a>	2280	254.3859649	Business
32	sfer	Belarus	Ошей		93	<a href="https://www.instagram.com/sfer_ошей">https://www.instagram.com/sfer_ошей</a>	3670	253.4059946	Business
33	gav	Belarus	Шлей	ZEED	32	<a href="https://www.instagram.com/gav_шлей">https://www.instagram.com/gav_шлей</a>	1264	253.164557	Business
34	only	Belarus	Реаб	нск +	58	<a href="https://www.instagram.com/only_реаб">https://www.instagram.com/only_реаб</a>	2329	249.0339201	Business
35	nati	Belarus	NATU		53	<a href="https://www.instagram.com/nati_natu">https://www.instagram.com/nati_natu</a>	2145	247.0862471	Business
36	hvo	Belarus	ОДЕ	СК   Б	39	<a href="https://www.instagram.com/hvo_оде">https://www.instagram.com/hvo_оде</a>	1583	246.3676563	Business
37	tam	Belarus	Приц		47	<a href="https://www.instagram.com/tam_приц">https://www.instagram.com/tam_приц</a>	1909	246.2022001	Business
38	wel	Belarus	«W	к	72	<a href="https://www.instagram.com/wel_w">https://www.instagram.com/wel_w</a>	3010	239.2026578	Business
39	hol	Belarus	«ХОЛ	К	81	<a href="https://www.instagram.com/hol_хол">https://www.instagram.com/hol_хол</a>	3391	238.8675907	Business
40	fluff	Belarus	Анат	Пово	221	<a href="https://www.instagram.com/fluff_anat">https://www.instagram.com/fluff_anat</a>	9258	238.7124649	Business
41	diks	Belarus	Diks		33	<a href="https://www.instagram.com/diks_diks">https://www.instagram.com/diks_diks</a>	1401	235.5460385	Business
42	bes	Belarus	ЛЕП	ОБ   М	50	<a href="https://www.instagram.com/bes_леп">https://www.instagram.com/bes_леп</a>	2123	235.5157796	Business
43	dog	Belarus	DO		37	<a href="https://www.instagram.com/dog_do">https://www.instagram.com/dog_do</a>	1599	231.3946216	Media
44	alle	Belarus	ALLE	К	48	<a href="https://www.instagram.com/alle_alle">https://www.instagram.com/alle_alle</a>	2130	225.3521127	Business
45	dog	Belarus	Счас		31	<a href="https://www.instagram.com/dog_счас">https://www.instagram.com/dog_счас</a>	1388	223.3429395	Business
46	valt	Belarus	Валт	розни	86	<a href="https://www.instagram.com/valt_валт">https://www.instagram.com/valt_валт</a>	3903	220.3433256	Business
47	cha	Belarus	Ошей	сла	101	<a href="https://www.instagram.com/cha_ошей">https://www.instagram.com/cha_ошей</a>	4589	220.0915232	Business
48	nos	Poland	НОУ		33	<a href="https://www.instagram.com/nos_nou">https://www.instagram.com/nos_nou</a>	1507	218.9781022	Business
49	anir	Belarus	Проф		36	<a href="https://www.instagram.com/anir_prof">https://www.instagram.com/anir_prof</a>	1644	218.9781022	Business
50	titbi	Belarus	ТІТБІ		31	<a href="https://www.instagram.com/titbi_titbi">https://www.instagram.com/titbi_titbi</a>	1416	218.9265537	Business
51	rulz	Belarus	Нату	омце	40	<a href="https://www.instagram.com/rulz_нату">https://www.instagram.com/rulz_нату</a>	1828	218.8183807	Business
52	pek	Belarus	Мног	х	46	<a href="https://www.instagram.com/pek_мног">https://www.instagram.com/pek_мног</a>	2109	218.1128497	Business
53	pro	Belarus	PROF	л, CE	37	<a href="https://www.instagram.com/pro_prof">https://www.instagram.com/pro_prof</a>	1706	216.8815944	Business
54	loft	Belarus	Миск		49	<a href="https://www.instagram.com/loft_misk">https://www.instagram.com/loft_misk</a>	2273	215.5741311	Business
55	bee	Belarus	Beez		77	<a href="https://www.instagram.com/bee_beez">https://www.instagram.com/bee_beez</a>	3640	211.5384615	Business
56	kotc	Belarus	Корм		305	<a href="https://www.instagram.com/kotc_korm">https://www.instagram.com/kotc_korm</a>	14445	211.1457252	Business

Рис. 17. Фрагмент исходной таблицы-базы данных.  
Fig. 17. Fragment of the original database table.

В целях обработки данной таблицы рекомендуется использование LLM – использовался специально сформулированный запрос в OpenAI API; в

целях исключения искажения данных применялся структурированный ответ (рис. 18).

```
from openai import OpenAI
from pydantic import BaseModel, RootModel
from typing import List

API_EKY = 'api-key'

class AccountClassificationResult(BaseModel):
    username: str
    classification: str
    category: str
    explanation: str

class AccountClassificationResultList(BaseModel):
    results: List[AccountClassificationResult] =
RootModel(AccountClassificationResult)

client = OpenAI(
    api_key=API_EKY
)

def process_batch(batch):
    """Simulates processing a batch using the client function."""
    # Replace this with your actual client call logic
    response = client.beta.chat.completions.parse(
        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": prompt_template},
            {"role": "user", "content": batch.to_csv(index=False)},
        ],
        response_format=AccountClassificationResultList
    )
    return response

from tqdm import tqdm
import json

merged_results = []

batch_size = 100

for start in tqdm(range(0, len(df), batch_size), desc="Processing
batches"):
    end = start + batch_size
    batch = df.iloc[start:end] # Extract batch

    # Process the batch
    response = process_batch(batch)

    # Merge the response with the results
    merged_results.extend(json.loads(response.choices[0].message.content)
['results'])
```

Рис. 18. Структурированный ответ в LLM (OpenAI API).

Fig. 18. Structured response in LLM (OpenAI API).

Опираясь на приведенные и полученные результаты, отметим выявленные закономерности и продуктивные возможности предложенного индекса:

Во-первых, предложенный индекс BVI позволяет извлекать достаточно релевантные результаты; большинство выявленных инфлюенсеров (нано-инфлюенсеры) с высоким значением индекса представляют реальный интерес для сотрудничества компаний.

Во-вторых, были выявлены достаточно вариативные страны, в которых может размещаться эффективная реклама (блогеры из Беларуси, Литвы, России, которых читает целевая аудитория из Беларуси).

В-третьих, были выделены неочевидные ниши для продвижения; в их числе тарологи и астрологи,

на которых подписана целевая аудитория канала, что позволяет расширять возможные направления продвижения и охватывать более вариативные сегменты целевой аудитории.

В-четвертых, удалось выяснить, что «очевидные» каналы продвижения продукции компании (каналы от лица собак, каналы питомников и т. п.) пользуются практически нулевым вниманием со стороны целевой аудитории бизнеса, что обосновывает отсутствие перспектив партнерства с данными инфлюенсерами.

Все изложенное доказывает выделенные положения о ценности и преимуществах оптимизации инфлюенс-маркетинга с использованием больших данных, Python и LLM (современных цифровых технологий (программных средств)) и доказывает, что в дальнейшем имеет место расширение и

апробация функциональных возможностей разработанного программного кода.

### Выводы

Таким образом, проведенное исследование вносит вклад и расширяет существующие представления об интеграции аналитических и цифровых технологий в бизнес-процессах современных компаний. На примере инфлюенс-маркетинга (как востребованного и популяризирующегося инструмента продвижения) доказывается необходимость проработки процедур выбора инфлюенсеров, их последующего контроля и оценки эффективности взаимодействия. Разработанный авторский индекс BVI (Blogger Value Index), в сочетании со средствами технологий больших данных, LLM и программными механизмами на базе Python, позволяют оптимизировать данную задачу и вывести её на новый уровень организации, что потенциально обеспечивает повышение эффективности маркетинга и расширяет потенциал реализации креативной функции при принятии решений.

Разработанный индекс обладает достаточно обширным спектром преимуществ, а именно 1) позволяет выявлять наиболее релевантных ин-

флюенсеров; 2) сокращает (существенно) время, затрачиваемое на поиск инфлюенсера и его аналитику; 3) обеспечивает расширение географии и охвата подписчиков – потенциальных потребителей продукции; 4) позволяет увеличивать эффективность инвестиций в маркетинг как следствие интеллектуальной аналитической поддержки принимаемых решений.

Итак, с точки зрения маркетинга, ценность полученных результатов исследования и предложенного авторского подхода определяются возможностью поиска дополнительных точек роста для бизнеса через оптимизацию современными технологиями. Таким образом, обоснованные и сформированные на основе аналитики больших данных решения в инфлюенс-маркетинге позволяют компаниям вырабатывать более жизнеспособные и результирующие стратегии. Перспективы дальнейших исследований связаны с обоснованием влияния такой аналитики с применением предложенного авторского индекса BVI на эффективность и возвратность инвестиций при организации инфлюенс-маркетинга, что найдет отражение в будущих работах автора.

### Список источников

1. АРИР. Инфлюенс-маркетинг [Электронный ресурс]. URL: [https://interactivead.ru/wp-content/uploads/2024/10/arir\\_inflyuens\\_marketing\\_2024\\_stranicy.pdf](https://interactivead.ru/wp-content/uploads/2024/10/arir_inflyuens_marketing_2024_stranicy.pdf) (дата обращения: 26.10.2024)
2. Ашихмин Е.Г., Левченко В.В., Селеткова Г.И. Опыт применения больших языковых моделей для анализа социологических данных, полученных в результате интервью о восприятии студентами предпринимательской деятельности // Цифровая социология. 2024. Т. 7. № 3. С. 4 – 14.
3. Писарева Е.В. Инфлюенс-маркетинг как инструмент эффективной коммуникации в цифровой среде // Beneficium. 2023. № 4 (49). С. 85 – 91.
4. Терещенко Д.А. Роль контент-маркетинга и инфлюенс-маркетинга в повышении эффективности отдела продаж // Практический маркетинг. 2024. № 8. С. 52 – 54.
5. Филиппов В.Н. Трансформация информационно-коммуникационных стратегий коммерческих структур в период постпандемии и экономической турбулентности // Вопросы теории и практики журналистики. 2021. Т. 10. № 2. С. 351 – 365.
6. Чибисова Е.А., Кулюдина А.С. Инфлюенс-маркетинг: преимущества, риски, рекомендации // Российская школа связей с общественностью. 2024. № 35. С. 136 – 151.
7. Farseev A., Yang Q., Ongpin M., Gossoudarev I., Chu-Farseeva Y.-Y., Nikolenko S. SOMONITOR: Combining Explainable AI & Large Language Models for Marketing Analytics [Электронный ресурс] // arXiv. 2024. DOI: 10.48550/arXiv.2407.13117
8. Gao Y., Arava S.K., Li Y., Snyder J.W. Jr. Improving the Capabilities of Large Language Model Based Marketing Analytics Copilots With Semantic Search And Fine-Tuning // International Journal on Cybernetics & Informatics (IJCI). 2024. Vol. 13. № 2. P. 15 – 31. DOI: 10.5121/ijci.2024.130202
9. Influencer marketing market size worldwide from 2016 to 2024 [Электронный ресурс]. URL: <https://www.statista.com/statistics/1092819/global-influencer-market-size/> (дата обращения: 26.10.2024)
10. Liu X., Sun A., An P., Ma T., Zhao J. Influencer: Empowering Everyday Users in Creating Promotional Posts via AI-infused Exploration and Customization [Электронный ресурс] // arXiv. 2024. DOI: 10.48550/arXiv.2407.14928

11. Tiukhova E., Penaloza E., Óskarsdóttir M., Baesens B., Snoeck M., Bravo C. INFLECT-DGNN: Influencer Prediction With Dynamic Graph Neural Networks // IEEE Access. 2024. Vol. 12. P. 115026 – 115041. DOI: 10.1109/ACCESS.2024.3443533
12. Zhang X., Chen X., Liu Y., Wang J., Hu Z., Yan R. SAGraph: A Large-scale Text-Rich Social Graph Dataset for Advertising Campaigns [Электронный ресурс] // arXiv. 2024. DOI: 10.48550/arXiv.2403.1510 [Jelektronnyj resurs] // arXiv. 2024. DOI: 10.48550/arXiv.2403.15105

### References

1. ARIR. Influence Marketing [Electronic resource]. URL: [https://interactivead.ru/wp-content/uploads/2024/10/arir\\_inflyuens\\_marketing\\_2024\\_stranicy.pdf](https://interactivead.ru/wp-content/uploads/2024/10/arir_inflyuens_marketing_2024_stranicy.pdf) (date accessed: 10/26/2024)
2. Ashikhmin E.G., Levchenko V.V., Seletkova G.I. Experience of using large language models for the analysis of sociological data obtained as a result of interviews on students' perception of entrepreneurial activity. Digital sociology. 2024. Vol. 7. No. 3. P. 4 – 14.
3. Pisareva E.V. Influence Marketing as a Tool for Effective Communication in the Digital Environment. Beneficium. 2023. No. 4 (49). P. 85 – 91.
4. Tereshchenko D.A. The Role of Content Marketing and Influence Marketing in Improving the Efficiency of the Sales Department. Practical Marketing. 2024. No. 8. P. 52 – 54.
5. Filippov V.N. Transformation of Information and Communication Strategies of Commercial Structures in the Post-Pandemic and Economic Turbulence Period. Issues of Theory and Practice of Journalism. 2021. Vol. 10. No. 2. P. 351 – 365.
6. Chibisova E.A., Kulyudina A.S. Influence Marketing: Benefits, Risks, Recommendations. Russian School of Public Relations. 2024. No. 35. P. 136 – 151.
7. Farseev A., Yang Q., Ongpin M., Gossoudarev I., Chu-Farseeva Y.-Y., Nikolenko S. SOMONITOR: Combining Explainable AI & Large Language Models for Marketing Analytics [Electronic resource]. arXiv. 2024. DOI: 10.48550/arXiv.2407.13117
8. Gao Y., Arava S.K., Li Y., Snyder J.W. Jr. Improving the Capabilities of Large Language Model Based Marketing Analytics Copilots With Semantic Search And Fine-Tuning. International Journal on Cybernetics & Informatics (IJCI). 2024. Vol. 13. No. 2. P. 15 – 31. DOI: 10.5121/ijci.2024.130202
9. Influencer marketing market size worldwide from 2016 to 2024 [Electronic resource]. URL: <https://www.statista.com/statistics/1092819/global-influencer-market-size/> (accessed: 26.10.2024)
10. Liu X., Sun A., An P., Ma T., Zhao J. Influencer: Empowering Everyday Users in Creating Promotional Posts via AI-infused Exploration and Customization [Electronic resource]. arXiv. 2024. DOI: 10.48550/arXiv.2407.14928
11. Tiukhova E., Penaloza E., Óskarsdóttir M., Baesens B., Snoeck M., Bravo C. INFLECT-DGNN: Influencer Prediction With Dynamic Graph Neural Networks. IEEE Access. 2024. Vol. 12. P. 115026 – 115041. DOI: 10.1109/ACCESS.2024.3443533
12. Zhang X., Chen X., Liu Y., Wang J., Hu Z., Yan R. SAGraph: A Large-scale Text-Rich Social Graph Dataset for Advertising Campaigns [Electronic resource]. arXiv. 2024. DOI: 10.48550/arXiv.2403.1510. [Jelektronnyj resurs]. arXiv. 2024. DOI: 10.48550/arXiv.2403.15105

### Информация об авторе

Коновалик Е.А., архитектор Программного Обеспечения, EPAM Systems, г. Сан-Хосе, ул. Санта-Клара, д. 181, кв. 1203, Калифорния, США. Orcid ID: <https://orcid.org/0009-0004-2354-4614>, [yauheni.kanavalik@gmail.com](mailto:yauheni.kanavalik@gmail.com), +1 562 476 16 44

© Коновалик Е.А., 2025