

Кластеризация пользователей онлайн-магазина методом K-Means с визуализацией в Google Colab

Мандронов Антон Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается практическое применение алгоритма кластеризации K-Means для сегментации клиентской базы онлайн-магазина на основе поведенческих и транзакционных признаков. Разработка и визуализация выполнены в облачной среде Google Colab с использованием библиотек Pandas, Scikit-learn, Matplotlib и Plotly. Проведён анализ структуры исходных данных, включающий предобработку пропусков и масштабирование признаков. Оптимальное число кластеров определено с помощью метода локтя и коэффициента силуэта. Полученные сегменты интерпретированы через усреднённые профили кластеров, для наглядного представления многомерных данных применены методы снижения размерности PCA и t-SNE. Результаты демонстрируют возможность выделения трёх устойчивых групп клиентов, различающихся по частоте покупок, среднему чеку и времени, прошедшему с последнего заказа. Предложенный подход позволяет с минимальными вычислительными затратами получить интерпретируемую сегментацию, пригодную для построения персонализированных маркетинговых стратегий.

Ключевые слова: кластеризация, K-Means, сегментация клиентов, онлайн-магазин, Google Colab, визуализация данных, машинное обучение, PCA, анализ данных.

Clustering of online store customers using K-Means with visualization in Google Colab

Mandronov Anton Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article examines the practical application of the K-Means clustering algorithm for segmenting the customer base of an online store based on behavioral and transactional features. The development and visualization are carried out in the Google Colab cloud environment using Pandas, Scikit-learn, Matplotlib, and Plotly libraries. The structure of the source data is analyzed, including preprocessing of missing values and feature scaling. The optimal number of clusters is determined using the elbow method and silhouette coefficient. The resulting segments are

interpreted through average cluster profiles, and dimensionality reduction techniques such as PCA and t-SNE are applied for visual representation of multidimensional data. The results demonstrate the ability to identify three stable customer groups differing in purchase frequency, average order value, and recency of the last purchase. The proposed approach enables obtaining interpretable segmentation with minimal computational costs, suitable for developing personalized marketing strategies.

Keywords: clustering, K-Means, customer segmentation, online store, Google Colab, data visualization, machine learning, PCA, data analysis.

1 Введение

1.1 Актуальность

Современный рынок электронной коммерции характеризуется высокой конкуренцией и постоянно растущим объёмом накапливаемых данных о поведении пользователей. Эффективное управление взаимоотношениями с клиентами требует перехода от массовых маркетинговых акций к персонализированному взаимодействию, основанному на понимании различий между группами потребителей. Сегментация клиентской базы позволяет оптимизировать рекламные бюджеты, повысить конверсию и удержание клиентов за счёт предложения релевантных товаров и услуг. Среди множества методов сегментации алгоритмы кластеризации, в частности K-Means, занимают особое место благодаря сочетанию простоты реализации, высокой вычислительной эффективности и наглядной интерпретируемости результатов. Использование облачных сред, таких как Google Colab, предоставляет исследователям и практикующим аналитикам доступ к вычислительным ресурсам без необходимости развёртывания локальной инфраструктуры, что делает подобные аналитические решения доступными для широкого круга специалистов.

1.2 Обзор исследований

В работе А. С. Платоновой и М. Н. Рыжковой проводится сравнительный анализ алгоритмов машинного обучения (K-Means, иерархическая кластеризация, DBSCAN) для сегментации пользователей на основе транзакционных данных. Установлено, что метод K-Means в сочетании с RFM-метриками обеспечивает оптимальное соотношение скорости и качества сегментации при обработке больших объёмов информации [1]. В исследовании Е. С. Пискун, Д. В. Нуансенгси и Е. Н. Котько рассматриваются подходы к оптимизации бизнес-процессов в электронной коммерции с использованием методов искусственного интеллекта. Подчёркивается критическая важность предварительной обработки данных, включая логарифмическую трансформацию признаков с высокой асимметрией, для повышения устойчивости моделей машинного обучения [2]. В публикации В. А. Ганявина, С. Ю. Черникова и А. А. Макарова представлена методика RFM-сегментации клиентской базы с последующей классификацией алгоритмами машинного обучения. Показано,

что кластеризация на основе стандартизированных RFM-показателей позволяет выделить однородные группы клиентов и повысить адресность маркетинговых коммуникаций [3]. В работе С. А. Гутника и М. М. Звягинцева исследуется применение методов интеллектуального анализа данных и машинного обучения для повышения эффективности управленческих решений. Акцентируется внимание на значимости визуализации результатов кластеризации с помощью методов снижения размерности (PCA, t-SNE), что существенно облегчает интерпретацию сегментов [4]. В исследовании Н. А. Титовой и В. Ю. Щеколдина рассматривается разработка адресных маркетинговых стратегий на основе модифицированного RFM-анализа. Демонстрируется, что расширение классической RFM-модели дополнительными параметрами позволяет более точно дифференцировать клиентские сегменты и повышает эффективность персонализированных коммуникаций с потребителями [5].

1.3 Цель исследования

Цель исследования — на примере обезличенного набора данных о транзакциях пользователей онлайн-магазина продемонстрировать полный цикл сегментации клиентской базы методом K-Means, реализованный в среде Google Colab.

2 Материалы и методы

В качестве исходных данных использован общедоступный набор данных о розничных транзакциях "Online Retail" (источник: UCI Machine Learning Repository), содержащий сведения о заказах интернет-магазина за период с 01.12.2010 по 09.12.2011. Датасет включает следующие поля: номер счёта (InvoiceNo), код товара (StockCode), описание товара (Description), количество (Quantity), дата и время счёта (InvoiceDate), цена за единицу (UnitPrice), идентификатор клиента (CustomerID), страна (Country).

Анализ и моделирование выполнялись в облачной среде Google Colab с использованием языка Python 3 и библиотек: Pandas (обработка данных), NumPy (численные операции), Scikit-learn (предобработка, кластеризация, метрики качества), Matplotlib и Plotly (визуализация).

Методология исследования включала следующие этапы:

1. Загрузка и очистка данных: удаление записей с пропущенными идентификаторами клиентов, исключение отрицательных значений количества (возвраты) и нулевых цен.

2. Формирование признаков RFM: для каждого уникального клиента рассчитаны показатели Recency (количество дней с момента последней покупки до конечной даты анализа), Frequency (общее число совершённых заказов) и Monetary (суммарная стоимость всех приобретённых товаров).

3. Предобработка признаков: логарифмическое преобразование Monetary и Frequency для уменьшения асимметрии распределений, стандартизация всех признаков методом StandardScaler.

4. Выбор числа кластеров: анализ графика метода локтя (зависимость внутрикластерной суммы квадратов от числа кластеров) и коэффициента силуэта.

5. Кластеризация: применение алгоритма K-Means с оптимальным числом кластеров.

6. Визуализация и интерпретация: построение двумерных проекций с помощью Principal Component Analysis (PCA) и t-distributed Stochastic Neighbor Embedding (t-SNE), анализ усреднённых характеристик каждого кластера.

3 Результаты и обсуждения

Перед началом работы необходимо установить и импортировать все необходимые библиотеки. В Google Colab большинство из них уже предустановлены, однако для гарантии можно выполнить установку Plotly. Затем загружается файл с датасетом (рис. 1).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
import datetime as dt
import warnings
warnings.filterwarnings('ignore')

df = pd.read_excel('Online Retail.xlsx', dtype={'CustomerID': str, 'InvoiceNo': str})
print(f"Исходный размер данных: {df.shape}")
print(df.head())
```

Рис. 1. Импорт библиотек и загрузка данных

После выполнения данного блока в памяти оказывается DataFrame df, содержащий 541909 строк и 8 столбцов (рис. 2).

```
Исходный размер данных: (541909, 8)
InvoiceNo StockCode Description Quantity \
0 536365 85123A WHITE HANGING HEART T-LIGHT HOLDER 6
1 536365 71053 WHITE METAL LANTERN 6
2 536365 84406B CREAM CUPID HEARTS COAT HANGER 8
3 536365 84029G KNITTED UNION FLAG HOT WATER BOTTLE 6
4 536365 84029E RED WOOLLY HOTTIE WHITE HEART. 6

InvoiceDate UnitPrice CustomerID Country
0 2010-12-01 08:26:00 2.55 17850 United Kingdom
1 2010-12-01 08:26:00 3.39 17850 United Kingdom
2 2010-12-01 08:26:00 2.75 17850 United Kingdom
3 2010-12-01 08:26:00 3.39 17850 United Kingdom
4 2010-12-01 08:26:00 3.39 17850 United Kingdom
```

Рис. 2. Вывод первых строк датасета

Удалим записи с отсутствующими идентификаторами клиентов, отрицательным или нулевым количеством товара и нулевой ценой. Также добавим расчётный столбец TotalSum, представляющий стоимость позиции в чеке (рис. 3).

```
df_clean = df.dropna(subset=['CustomerID']).copy()
df_clean['CustomerID'] = df_clean['CustomerID'].astype(int)
df_clean = df_clean[df_clean['Quantity'] > 0]
df_clean = df_clean[df_clean['UnitPrice'] > 0]
df_clean['TotalSum'] = df_clean['Quantity'] * df_clean['UnitPrice']

print(f"Размер данных после очистки: {df_clean.shape}")
print(f"Количество уникальных клиентов: {df_clean['CustomerID'].nunique()}")
```

Размер данных после очистки: (397884, 9)

Количество уникальных клиентов: 4338

Рис. 3. Очистка и предварительная фильтрация данных

Для каждого клиента рассчитываются три ключевые характеристики: Recency (давность последней покупки в днях), Frequency (количество уникальных заказов) и Monetary (общая сумма потраченных средств). Опорной датой для Recency служит день, следующий за максимальной датой в наборе данных (рис. 4).

```
snapshot_date = df_clean['InvoiceDate'].max() + dt.timedelta(days=1)
print(f"Опорная дата для расчета Recency: {snapshot_date}")

rfm = df_clean.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (snapshot_date - x.max()).days,
    'InvoiceNo': 'nunique',
    'TotalSum': 'sum'
}).reset_index()

rfm.columns = ['CustomerID', 'Recency', 'Frequency', 'Monetary']

print(rfm.head())
print(rfm.describe().round(2))
```

Опорная дата для расчета Recency: 2011-12-10 12:50:00

	CustomerID	Recency	Frequency	Monetary
0	12346	326	1	77183.60
1	12347	2	7	4310.00
2	12348	75	4	1797.24
3	12349	19	1	1757.55
4	12350	310	1	334.40
	CustomerID	Recency	Frequency	Monetary
count	4338.00	4338.00	4338.00	4338.00
mean	15300.41	92.54	4.27	2054.27
std	1721.81	100.01	7.70	8989.23
min	12346.00	1.00	1.00	3.75
25%	13813.25	18.00	1.00	307.41
50%	15299.50	51.00	2.00	674.48
75%	16778.75	142.00	5.00	1661.74
max	18287.00	374.00	209.00	280206.02

Рис. 4. Формирование RFM-таблицы

Распределения Monetary и Frequency имеют ярко выраженную правостороннюю асимметрию, что может негативно сказаться на работе K-Means. Для снижения асимметрии применяется натуральное

логарифмирование с добавлением единицы (функция `log1p`). Затем все признаки стандартизируются с помощью `StandardScaler` для приведения к нулевому среднему и единичной дисперсии (рис. 5).

```
rfm_scaled = rfm[['Recency', 'Frequency', 'Monetary']].copy()

rfm_scaled['Frequency_log'] = np.log1p(rfm_scaled['Frequency'])
rfm_scaled['Monetary_log'] = np.log1p(rfm_scaled['Monetary'])

features = ['Recency', 'Frequency_log', 'Monetary_log']

scaler = StandardScaler()
rfm_scaled[features] = scaler.fit_transform(rfm_scaled[features])
X = rfm_scaled[features].values

print("Первые 5 строк после масштабирования:")
print(rfm_scaled[features].head())
```

Первые 5 строк после масштабирования:

	Recency	Frequency_log	Monetary_log
0	2.334574	-0.955214	3.706225
1	-0.905340	1.074425	1.411843
2	-0.175360	0.386304	0.716489
3	-0.735345	-0.955214	0.698739
4	2.174578	-0.955214	-0.618962

Рис. 5. Логарифмирование и стандартизация

Для определения числа кластеров k последовательно обучаются модели K-Means при k от 1 до 10. Для каждого k вычисляются внутрикластерная сумма квадратов (WCSS) и средний коэффициент силуэта (рис. 6). Построенные графики помогают выявить компромиссное значение (рис. 7). Изгиб кривой WCSS и максимум силуэта наблюдаются при $k = 3$.

```
wcss = []
sil_scores = []
K_range = range(2, 11)
for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X)
    wcss.append(kmeans.inertia_)
    sil_scores.append(silhouette_score(X, labels))
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))
ax1.plot(range(1, 11), [KMeans(n_clusters=i, random_state=42, n_init=10).fit(X).inertia_ for i in range(1,11)], 'bo-')
ax1.set_xlabel('Число кластеров (k)')
ax1.set_ylabel('WCSS')
ax1.set_title('Метод локтя')
ax1.grid(True)
ax2.plot(K_range, sil_scores, 'ro-')
ax2.set_xlabel('Число кластеров (k)')
ax2.set_ylabel('Средний коэффициент силуэта')
ax2.set_title('Оценка силуэта')
ax2.grid(True)
plt.tight_layout()
plt.show()
optimal_k = K_range[np.argmax(sil_scores)]
print(f"Оптимальное число кластеров по коэффициенту силуэта: {optimal_k}")
```

Рис. 6. Расчёт WCSS и коэффициента силуэта

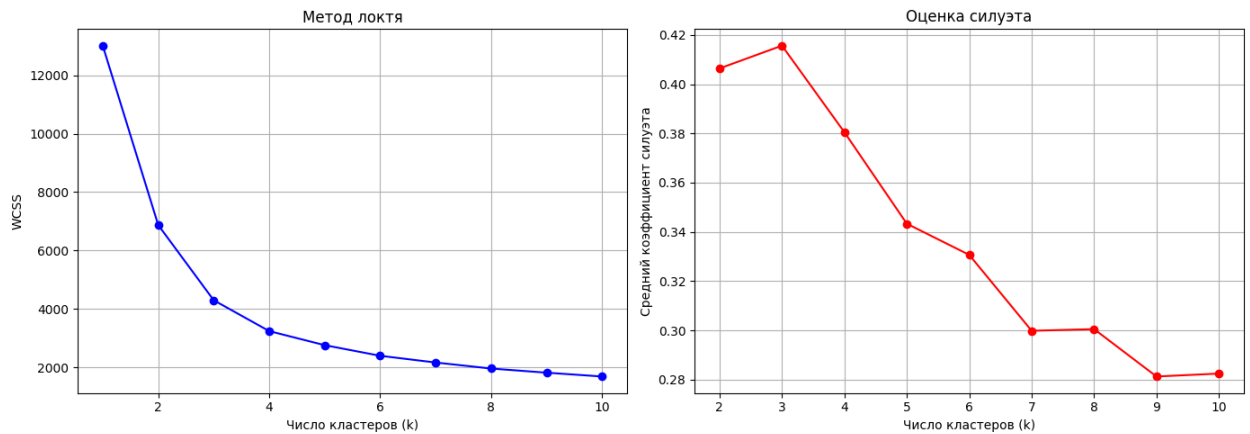


Рис. 7. График метода локтя и коэффициента силуэта

Модель K-Means обучается с выбранным числом кластеров $k = 3$. Метки кластеров добавляются в исходный DataFrame rfm (рис. 8).

```
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
rfm['cluster'] = kmeans.fit_predict(X)

print("Распределение клиентов по кластерам:")
print(rfm['cluster'].value_counts().sort_index())
```

Распределение клиентов по кластерам:

Cluster	count
0	2036
1	1323
2	979

Name: count, dtype: int64

Рис. 8. Обучение K-Means и назначение кластеров

Для наглядного отображения многомерных данных используются алгоритмы PCA и t-SNE. PCA даёт линейную проекцию, сохраняющую глобальную структуру, тогда как t-SNE лучше отображает локальные соседства. Визуализация выполняется с помощью интерактивной библиотеки Plotly (рис. 9).

```

pca = PCA(n_components=2, random_state=42)
pca_result = pca.fit_transform(X)
rfm['PCA1'] = pca_result[:, 0]
rfm['PCA2'] = pca_result[:, 1]

tsne_sample = rfm.sample(min(1000, len(rfm)), random_state=42)
tsne = TSNE(n_components=2, random_state=42, perplexity=30, n_iter=1000)
tsne_result = tsne.fit_transform(X[tsne_sample.index])
tsne_sample['TSNE1'] = tsne_result[:, 0]
tsne_sample['TSNE2'] = tsne_result[:, 1]

fig_pca = px.scatter(rfm, x='PCA1', y='PCA2', color=rfm['Cluster'].astype(str),
                    title='Визуализация кластеров (PCA)',
                    labels={'color': 'Кластер'},
                    opacity=0.7,
                    hover_data=['Recency', 'Frequency', 'Monetary'])
fig_pca.show()

fig_tsne = px.scatter(tsne_sample, x='TSNE1', y='TSNE2', color=tsne_sample['Cluster'].astype(str),
                    title='Визуализация кластеров (t-SNE, n=1000)',
                    labels={'color': 'Кластер'},
                    opacity=0.7,
                    hover_data=['Recency', 'Frequency', 'Monetary'])
fig_tsne.show()

```

Рис. 9. Построение проекций PCA и t-SNE

Полученные графики (рис. 10 и рис. 11) демонстрируют чёткое разделение кластеров, особенно выраженное в проекции t-SNE.

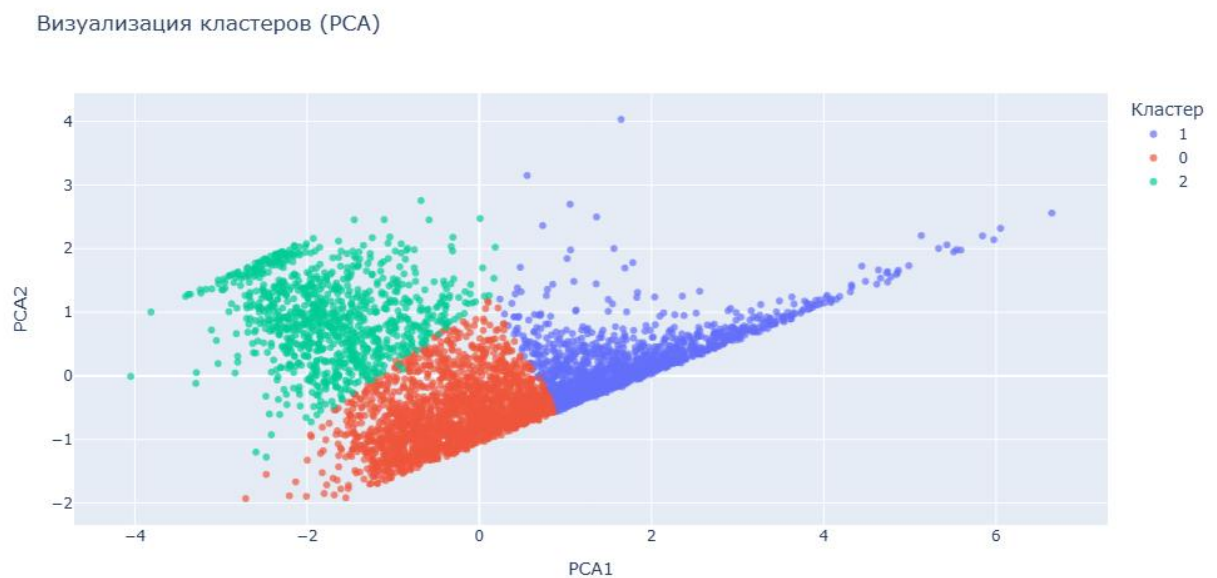


Рис. 10. График проекции PCA

Визуализация кластеров (t-SNE, n=1000)



Рис. 11. График проекции t-SNE

Для содержательной характеристики каждого сегмента вычисляются средние значения исходных (не логарифмированных) RFM-метрик, а также размер кластера в абсолютном и процентном выражении (рис. 12). Результат выполнения кода представлен в таблице (табл. 1).

```
cluster_profile = rfm.groupby('cluster')[['Recency', 'Frequency', 'Monetary']].mean().round(2)
cluster_profile['Count'] = rfm['cluster'].value_counts()
cluster_profile['Percentage'] = (cluster_profile['Count'] / len(rfm) * 100).round(2)

print("Средние значения RFM по кластерам:")
print(cluster_profile)
```

Рис. 12. Расчёт профилей кластеров

Таблица 1 – Средние значения RFM-метрик по кластерам

Кластер	Recency (дней)	Frequency (заказов)	Monetary (y.e.)	Count	Percentage
0	54.86	2.05	615.24	2036	46.93%
1	30.10	9.83	5494.34	1323	30.50%
2	255.27	1.39	398.13	979	22.57%

Анализ таблицы позволяет дать содержательную характеристику каждому сегменту:

- Кластер 0 («Активные новички» или «Регулярные клиенты с низкой частотой»): самая многочисленная группа (46.93%), для которой характерна невысокая, но заметная частота покупок (в среднем 2 заказа) и умеренная суммарная стоимость (615 y.e.). Главная особенность — относительно небольшая давность последней покупки (около 55 дней), что свидетельствует о продолжающемся взаимодействии с магазином. Это ядро клиентской базы, которое совершает покупки время от времени. Маркетинговая стратегия для данной группы должна быть направлена на увеличение частоты транзакций

и среднего чека через персонализированные рекомендации, программы поощрения повторных покупок и кросс-продажи.

- Кластер 1 («VIP-клиенты»): наиболее ценный сегмент (30.50% клиентов), демонстрирующий наилучшие показатели по всем метрикам. Resency минимально (30 дней) — клиенты активны и совершают покупки регулярно. Средняя частота заказов составляет 9.83, а суммарные траты достигают почти 5500 у.е., что в разы превышает показатели других групп. Это лояльные и высокодоходные покупатели. Стратегия взаимодействия: программы лояльности премиум-уровня, эксклюзивные предложения, ранний доступ к новинкам, персональный менеджмент и мероприятия для укрепления долгосрочных отношений.

- Кластер 2 («Спящие клиенты»): наименьшая по численности группа (22.57%) с крайне высоким значением Resency (более 255 дней), что указывает на потерю интереса или переход к конкурентам. Частота покупок минимальна (1.39 заказа в среднем), а суммарные расходы невелики (398 у.е.). Эти клиенты, вероятно, совершили одну-две пробные покупки и прекратили взаимодействие. Для данного сегмента целесообразны реактивационные кампании: специальные предложения с ограниченным сроком действия, персонализированные скидки на товары, которые они ранее просматривали или покупали, а также email-рассылки с напоминанием о бренде.

4 Выводы

Проведённое исследование подтвердило эффективность применения алгоритма K-Means для сегментации клиентской базы онлайн-магазина в условиях ограниченных вычислительных ресурсов, доступных в среде Google Colab. Основные выводы работы:

- Использование RFM-метрик в качестве признакового пространства позволяет получить интерпретируемые сегменты, напрямую связанные с поведенческими паттернами покупателей.

- Предварительное логарифмирование и стандартизация данных являются необходимыми этапами для корректной работы метрического алгоритма K-Means ввиду сильной асимметрии исходных распределений Monetary и Frequency.

- Совместное применение метода локтя и коэффициента силуэта обеспечивает объективный выбор числа кластеров, в данном случае $k=3$.

- Визуализация результатов с помощью PCA и t-SNE значительно упрощает восприятие многомерной структуры данных и служит дополнительным подтверждением качества кластеризации.

- Выделенные сегменты («Активные новички», «VIP-клиенты» и «Спящие клиенты») имеют чёткие количественные и качественные различия, что позволяет сформулировать дифференцированные маркетинговые стратегии для каждой группы.

Таким образом, предложенный подход представляет собой воспроизводимый и масштабируемый аналитический конвейер, который

может быть легко адаптирован под конкретные бизнес-задачи и интегрирован в процессы принятия решений в сфере электронной коммерции.

Библиографический список

1. Платонова А. С., Рыжкова М. Н. Применение машинного обучения для сегментации пользователей в маркетинговых исследованиях // Радиотехнические и телекоммуникационные системы. 2025. № 3. С. 47–53. URL: <https://cyberleninka.ru/article/n/primenenie-mashinnogo-obucheniya-dlya-segmentatsii-polzovateley-v-marketingovyh-issledovaniyah> (дата обращения: 18.04.2026).
2. Пискун Е. С., Нуансенгси Д. В., Котько Е. Н. Оптимизация бизнес-процессов в электронной коммерции с применением методов и алгоритмов искусственного интеллекта // Доклады Белорусского государственного университета информатики и радиоэлектроники. 2024. Т. 22. № 6. С. 103–111. URL: <https://cyberleninka.ru/article/n/optimizatsiya-biznes-protsessov-v-elektronnoy-kommertsii-s-primeneniem-metodov-i-algoritmov-iskusstvennogo-intellekta> (дата обращения: 18.04.2026).
3. Ганявин В. А., Черников С. Ю., Макаров А. А. RFM-сегментация и классификация клиентской базы для группы 1R алгоритмом машинного обучения // Вестник МИРБИС. 2018. № 2. С. 14. URL: https://scholar.archive.org/work/2p5yeszzdrewzopsfob4b7u5se/access/wayback/https://cs.journal-mirbis.ru/-/MR86m_KvRZOtj4iRfmm2YQ/sv/document/1f/75/06/521295/370/2_2018_VM.pdf?1531207289#page=20 (дата обращения: 18.04.2026).
4. Гутник С. А., Звягинцев М. М. Применение методов интеллектуального анализа данных и машинного обучения для повышения эффективности управления в менеджменте // Международный бизнес. 2025. № 3 (13). С. 74–95. URL: <https://ibj.mgimo.ru/jour/article/view/130> (дата обращения: 18.04.2026).
5. Титова Н. А., Щеколдин В. Ю. Разработка адресных маркетинговых стратегий на основе модифицированного RFM-анализа // Практический маркетинг. 2015. № 4 (218). С. 8-16. URL: <https://cyberleninka.ru/article/n/razrabotka-adresnyh-marketingovyh-strategiy-na-osnove-modifitsirovannogo-rfm-analiza> (дата обращения: 18.04.2026).