

RFM-анализ клиентской активности на основе синтетических данных, сгенерированных с использованием Python

Андрюенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В работе представлена методика проведения RFM-анализа с применением языка программирования Python для сегментации клиентской базы. Синтетические данные были сгенерированы с помощью NumPy и pandas: 500 клиентов, 1000 товаров и 100 000 транзакций за 2024 год с диапазоном цен от 100 до 5000 рублей. Предложен алгоритм расчёта метрик Recency, Frequency и Monetary, реализована их оценка по трехуровневой шкале и классификация клиентов по сегментам. Работа выполнялась в среде Jupyter/Google Colab, что позволило автоматизировать процесс обработки данных и визуализировать результаты анализа. Представленный подход подтверждает целесообразность использования Python в задачах маркетинговой аналитики.

Ключевые слова: RFM-анализ, Python, синтетические данные, сегментация клиентов, обработка данных, визуализация, маркетинговая аналитика, pandas, NumPy

RFM analysis of customer activity based on synthetic data generated using Python

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The paper presents a methodology for conducting RFM analysis using the Python programming language for segmentation of the customer base. Synthetic data was generated using NumPy and pandas: 500 customers, 1,000 products, and 100,000 transactions for 2024 with a price range from 100 to 5,000 rubles. An algorithm for calculating the metrics Recency, Frequency and Monetary is proposed, their assessment on a three-level scale and the classification of customers by segment are implemented. The work was performed in the Jupiter/Google Colab environment, which made it possible to automate the data processing process and visualize the analysis results. The presented approach confirms the expediency of using Python in marketing analytics tasks.

Keywords: RFM analysis, Python, synthetic data, customer segmentation, data processing, visualization, marketing analytics, pandas, NumPy

1 Введение

1.1 Актуальность

Современные компании стремятся не только привлекать новых клиентов, но и эффективно взаимодействовать с существующими. В условиях высокой конкуренции необходимо грамотно распределять ресурсы, фокусируясь на ключевых потребителях. RFM-анализ, как один из самых простых и наглядных способов оценки клиентской ценности, позволяет идентифицировать лояльных, активных и «потерянных» клиентов. Использование Python для проведения такого анализа актуально благодаря его возможностям автоматизации, гибкости в обработке больших объёмов данных и доступности визуализации.

1.2 Обзор исследований

Методика RFM-анализа широко освещается в современной научной литературе и рассматривается как один из ключевых инструментов работы с клиентской базой. Так, в работе А.В. Кривко-Красько RFM-анализ рассматривается как эффективный инструмент оценки клиентской активности и оптимизации бизнес-процессов. Автор подчёркивает его значимость в условиях нестабильной экономической среды и ограниченных ресурсов маркетинга [1]. В исследовании М.Е. Ералевой, С.П. Столбиковой, М.А. Мурзахматова и А.А. Деткова акцент сделан на практической реализации анализа: приводятся примеры написания кода на Python и трактовки полученных результатов сегментации. Это подчёркивает актуальность автоматизированных решений в аналитике [2]. Д.О. Усов в своём исследовании выделяет RFM-анализ как ключевой механизм для запуска таргетированных маркетинговых кампаний. Отмечается, что грамотная сегментация клиентов на основе метрик Recency, Frequency и Monetary способствует повышению отклика и эффективности маркетинговых стратегий [3]. Анализ применения RFM в контексте сегментирования базы также представлен в статье З.Н. Шуклиной, В.А. Уткиной и О.С. Прошиной, где описываются примеры успешного использования методики для выявления групп потребителей с различной степенью лояльности. Подчёркивается возможность адаптации RFM-подхода под особенности различных бизнес-моделей [4].

1.3 Цель исследования

Цель исследования — разработать и протестировать алгоритм RFM-анализа с использованием синтетически сгенерированных данных, реализованный на языке Python, и визуализировать полученные результаты для поддержки принятия решений в области маркетинга.

2 Результаты и обсуждения

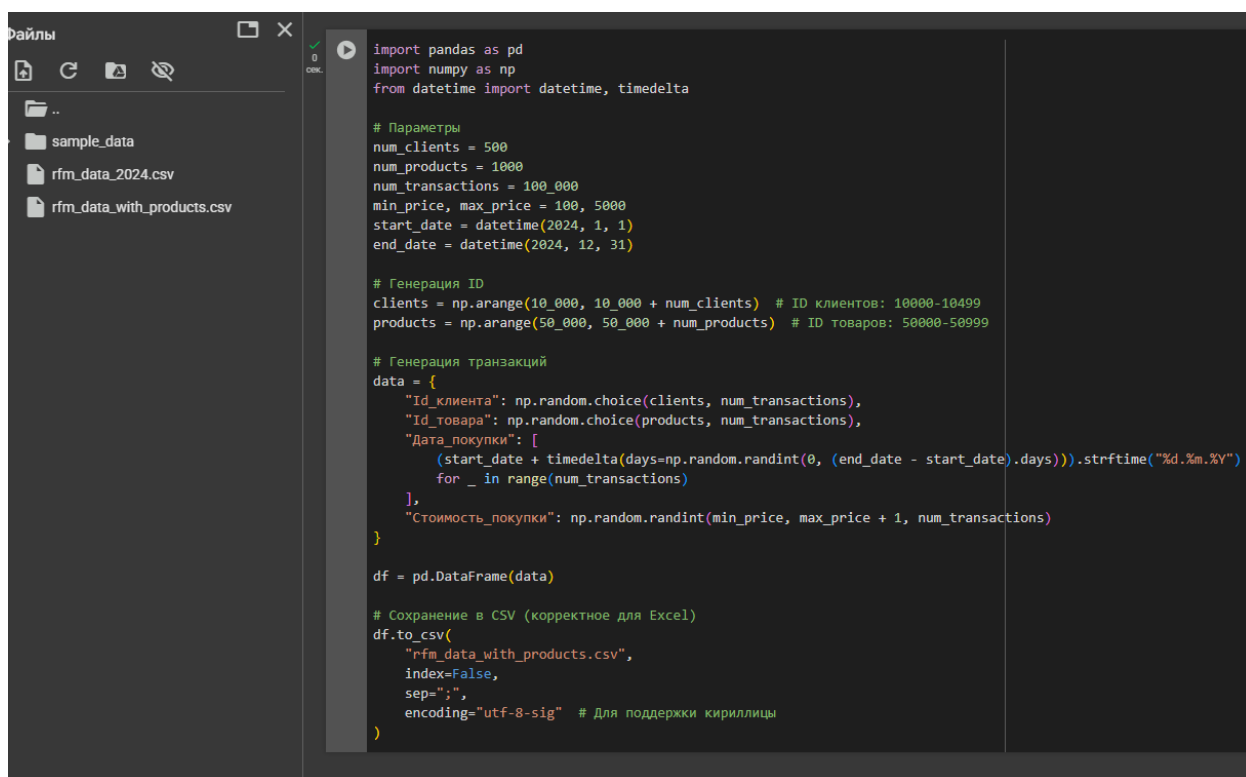
На первом этапе был составлен промт к ChatGPT для генерации кода, формирующего синтетические данные, соответствующие требованиям RFM-анализа. Использовался код с Python-библиотеками pandas, numpy и datetime.

Сгенерированные данные представляют собой таблицу с полями: Id_клиента, Id_товара, Дата_покупки, Стоимость_покупки.

Алгоритм создания данных включал в себя:

- генерацию 500 уникальных клиентов с ID от 10 000 до 10 499;
- выборку из 1000 уникальных товаров с ID от 50 000 до 50 999;
- формирование 100 000 записей о покупках с датами в пределах 2024 года;
- случайное распределение стоимости от 100 до 5000 рублей.

Код был выполнен в среде Google Colab, что позволило быстро протестировать и визуально проконтролировать результаты. После генерации данные были экспортированы в формат CSV с разделителем ; и кодировкой utf-8-sig, обеспечивающей корректное отображение кириллицы при открытии в Microsoft Excel.



```
import pandas as pd
import numpy as np
from datetime import datetime, timedelta

# Параметры
num_clients = 500
num_products = 1000
num_transactions = 100_000
min_price, max_price = 100, 5000
start_date = datetime(2024, 1, 1)
end_date = datetime(2024, 12, 31)

# Генерация ID
clients = np.arange(10_000, 10_000 + num_clients) # ID клиентов: 10000-10499
products = np.arange(50_000, 50_000 + num_products) # ID товаров: 50000-50999

# Генерация транзакций
data = {
    "Id_клиента": np.random.choice(clients, num_transactions),
    "Id_товара": np.random.choice(products, num_transactions),
    "Дата_покупки": [
        (start_date + timedelta(days=np.random.randint(0, (end_date - start_date).days))).strftime("%d.%m.%Y")
        for _ in range(num_transactions)
    ],
    "Стоимость_покупки": np.random.randint(min_price, max_price + 1, num_transactions)
}

df = pd.DataFrame(data)

# Сохранение в CSV (корректное для Excel)
df.to_csv(
    "rfm_data_with_products.csv",
    index=False,
    sep=";",
    encoding="utf-8-sig" # Для поддержки кириллицы
)
```

Рис. 1. Фрагмент кода генерации данных

После выполнения кода в рабочем каталоге появился файл rfm_data_with_products.csv, содержащий сгенерированную таблицу. Файл был открыт в Excel для первичной верификации (рис. 2).

A	B	C	D	E	F	G	H	I	J	K	L
Id_клиент	Дата_поку	Стоимость	Id_товара	всего кли	всего това	всего тра	мин цена	макс цена	начальная	конечная	дата
10139	07.08.2024	774	50748	100000	100000	100000	100	5000	#####	#####	
10333	30.09.2024	547	50080								
10193	06.05.2024	3459	50669								
10413	13.05.2024	3885	50662								
10202	26.02.2024	1736	50708								
10035	20.02.2024	1549	50414								
10082	02.08.2024	3474	50164								
10272	15.02.2024	3744	50831								
10071	20.09.2024	4824	50420								
10462	22.07.2024	3870	50777								
10180	16.01.2024	2088	50271								
10038	28.10.2024	4533	50657								
10126	20.07.2024	3425	50680								

Рис. 2. Фрагмент кода генерации данных

С помощью функции `=СЧЁТЗ(УНИК(A2:A100001))`, вставленной в ячейку E2, было подсчитано количество уникальных значений в столбце Id_клиента. Полученное значение — 500 — соответствует запланированному количеству клиентов. Для подтверждения охвата 1000 товаров была использована формула `=СЧЁТЗ(УНИК(B2:B100001))` в ячейке F2. В результате проверки установлено, что в выборке действительно присутствуют 1000 уникальных товаров.

Для подсчета общего числа строк использована формула `=СЧЁТЗ(A2:A100001)` в ячейке G2. Полученное значение — 100 000 транзакций — подтверждает соответствие сгенерированного объема заданному параметру.

Для оценки минимального и максимального значения в столбце Стоимость_покупки использовались формулы `=МИН(C2:C100001)` и `=МАКС(C2:C100001)`, вставленные в ячейки H2 и I2 соответственно. Проверка показала, что диапазон цен колеблется в пределах от 100 до 5000 рублей, как и требовалось по условиям.

Для определения, охватывают ли даты период строго внутри 2024 года, использовались формулы `=МИН(D2:D100001)` и `=МАКС(D2:D100001)` в ячейках J2 и K2. Результаты показали, что все транзакции попадают в границы от 01.01.2024 до 31.12.2024.

Полученные данные можно считать валидными и готовыми для последующего RFM-анализа. В рамках исследования был написан Python-скрипт, реализующий полный цикл RFM-анализа: от загрузки и предобработки данных до расчёта метрик, присвоения оценок, сегментации клиентов и построения визуализаций. Код включает использование библиотек `pandas`, `numpy`, `matplotlib` и `seaborn`, что обеспечило гибкость анализа и наглядность результатов. На рисунке 3 и 4 представлен фрагмент исходного кода, использованного для анализа клиентской базы.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime

# Загрузка данных
df = pd.read_csv("rfm_data_with_products.csv", sep=";", encoding="utf-8-sig")

# Преобразуем даты в формат datetime
df["Дата_покупки"] = pd.to_datetime(df["Дата_покупки"], format="%d.%m.%Y")

# Последняя дата в данных (можно задать вручную, если нужно)
current_date = df["Дата_покупки"].max() + pd.Timedelta(days=1)

# Расчет RFM-метрик
rfm = df.groupby("Id_клиента").agg({
    "Дата_покупки": lambda x: (current_date - x.max()).days, # Recency
    "Id_товара": "count", # Frequency
    "Стоимость_покупки": "sum" # Monetary
}).reset_index()

rfm.columns = ["Id_клиента", "Recency", "Frequency", "Monetary"]

# Присвоение оценок от 1 до 3 по квантилям
def assign_score(series, reverse=False):
    quantiles = series.quantile([0.33, 0.66]).values
    def score(x):
        if reverse:
            if x <= quantiles[0]: return 3
            elif x <= quantiles[1]: return 2
            else: return 1
        else:
            if x <= quantiles[0]: return 1
            elif x <= quantiles[1]: return 2
            else: return 3
    return series.apply(score)

rfm["R_score"] = assign_score(rfm["Recency"], reverse=True)
rfm["F_score"] = assign_score(rfm["Frequency"])
rfm["M_score"] = assign_score(rfm["Monetary"])

# Формируем строковый RFM-код
rfm["RFM_score"] = rfm["R_score"].astype(str) + rfm["F_score"].astype(str) + rfm["M_score"].astype(str)
```

Рис. 3. Python-скрипт для RFM-анализа

```
# Классификация сегментов
def segment(rfm_code):
    if rfm_code == "333":
        return "Лучшие клиенты"
    elif rfm_code[0] == "3":
        return "Недавние покупатели"
    elif rfm_code[1] == "3":
        return "Часто покупают"
    elif rfm_code[2] == "3":
        return "Большие траты"
    elif rfm_code == "111":
        return "Потерянные клиенты"
    else:
        return "Остальные"

rfm["Сегмент"] = rfm["RFM_score"].apply(segment)

# Анализ сегментов
segment_stats = rfm.groupby("Сегмент").agg({
    "Id_клиента": "count",
    "Recency": ["mean", "median"],
    "Frequency": ["mean", "median"],
    "Monetary": ["mean", "median"]
}).round(1)

# Визуализация
plt.figure(figsize=(10, 6))
sns.countplot(data=rfm, x="Сегмент", order=rfm["Сегмент"].value_counts().index)
plt.title("Распределение клиентов по сегментам")
plt.xticks(rotation=30)
plt.tight_layout()
plt.show()

# Тепловая карта медианных RFM
heatmap_data = rfm.groupby("RFM_score")[["Recency", "Frequency", "Monetary"]].median()
plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data, cmap="YlGnBu", annot=True, fmt=".0f")
plt.title("Медианные значения RFM по кодам")
plt.tight_layout()
plt.show()
```

Рис. 4. Python-скрипт для RFM-анализа

После подтверждения корректности данных был осуществлён основной этап RFM-анализа. Исходный файл `rfm_data_with_products.csv` был загружен в Python-среду, а данные — предварительно преобразованы: дата покупки — в формат `datetime`, а текущей датой анализа установлено значение на один день позже самой поздней покупки.

С помощью агрегирования по идентификатору клиента рассчитаны три ключевые метрики:

- **Recency** — количество дней с момента последней покупки клиента;
- **Frequency** — общее число покупок клиента за весь период;
- **Monetary** — суммарная стоимость всех покупок клиента.

Далее для каждой метрики была реализована процедура присвоения оценки по трехуровневой шкале: значения делились по квантилям 33% и 66%, что позволило получить ранжирование от 1 до 3 (низкий, средний и высокий уровни соответственно). Для **Recency** шкала была инверсирована — чем меньше дней с последней покупки, тем выше оценка.

Объединённые оценки по трем метрикам формировали RFM-код клиента (например, "332", "213" и т. д.), на основе которого производилась

сегментация. Была реализована базовая классификация клиентов на 6 основных групп:

- Лучшие клиенты (код "333") — показывают максимальные значения по всем метрикам;
- Недавние покупатели (начинаются с "3") — совершали покупки совсем недавно;
- Часто покупают (вторая цифра — "3") — отличаются высокой регулярностью;
- Большие траты (третья цифра — "3") — тратят выше среднего;
- Потерянные клиенты (код "111") — низкие значения по всем метрикам;
- Остальные — не попали ни в одну из крайних категорий.

Результаты классификации были представлены на диаграмме (рис. 5). Как видно, наибольшее количество клиентов попало в категории "Недавние покупатели" (более 150 человек) и "Остальные", в то время как "Лучшие клиенты", "Потерянные клиенты" и "Большие траты" встречаются значительно реже. Это типичная ситуация, отражающая правило Парето: меньшинство клиентов приносит основную долю прибыли.

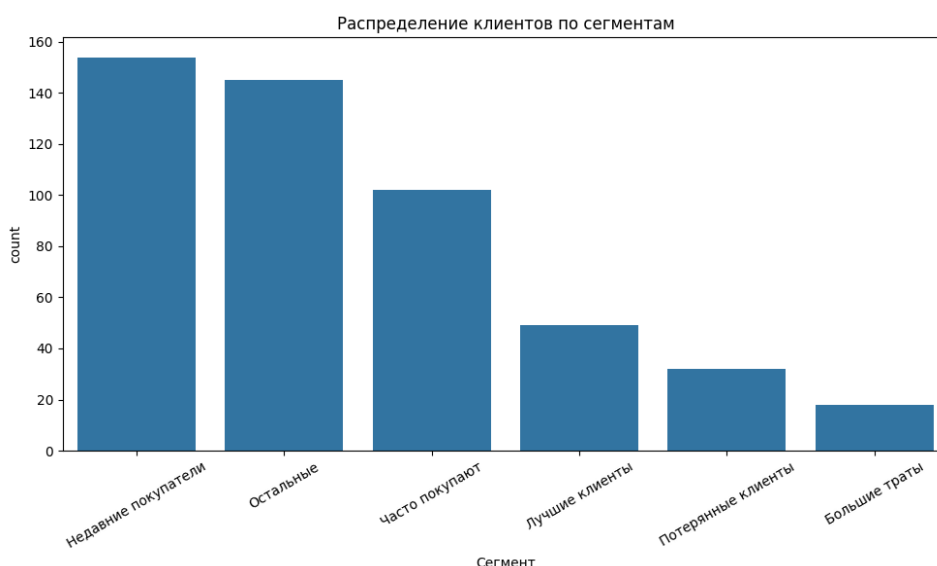


Рис. 5. Распределение клиентов по сегментам

Для более детального анализа была построена тепловая карта медианных значений RFM-метрик по кодам (рис. 6). Тепловая карта позволяет визуально оценить поведение различных RFM-групп. На ней можно выделить, что клиенты с кодом "333" действительно демонстрируют наименьшую давность, наибольшую частоту и максимальные траты. В то время как сегмент "111" обладает противоположными характеристиками.

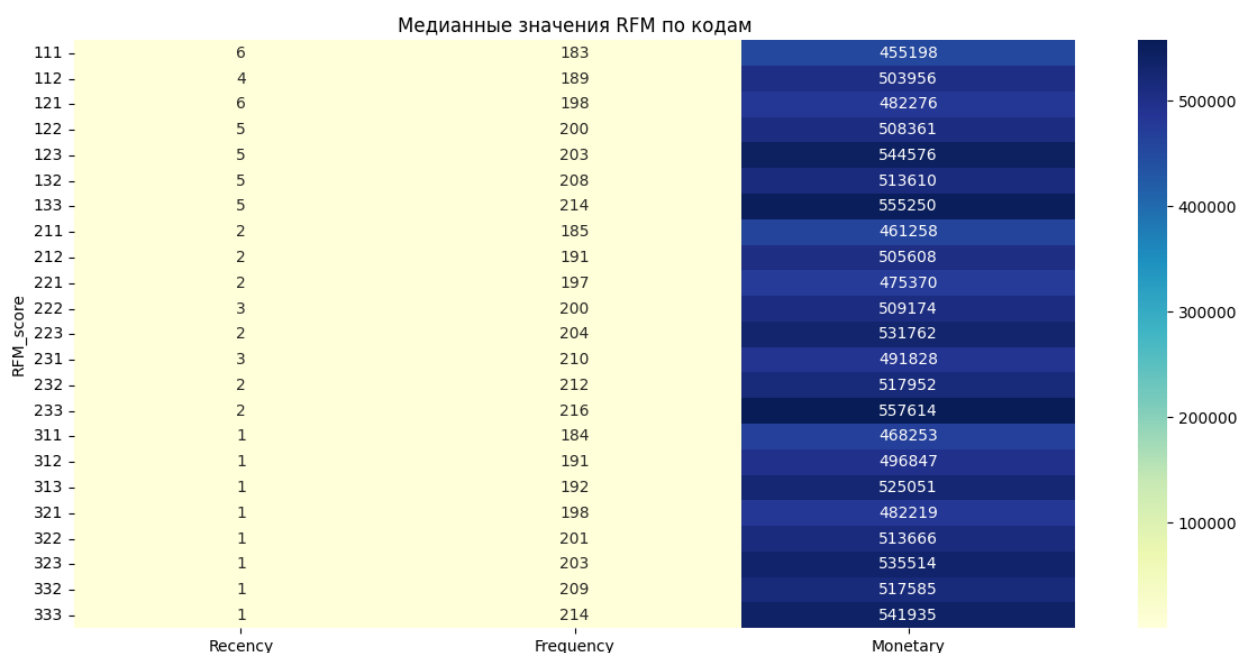


Рис. 6. Медианные значения RFM

Такая визуализация помогает маркетологам быстро идентифицировать приоритетные группы клиентов для ретаргетинга, программ лояльности или акций по возврату утраченной аудитории.

3 Выводы

Проведённое исследование подтвердило эффективность применения метода RFM-анализа для сегментации клиентской базы на основе синтетических данных. Использование языка Python и специализированных библиотек позволило автоматизировать расчёты ключевых метрик (Recency, Frequency, Monetary), провести ранжирование клиентов и сформировать сегменты с учётом их покупательской активности. Созданный скрипт обеспечивает прозрачность аналитического процесса, а визуализация в виде графиков и тепловой карты способствует интуитивному восприятию различий между клиентскими группами. Полученные результаты могут быть применены для построения таргетированных маркетинговых стратегий, повышения удержания клиентов и оптимизации рекламных расходов.

Библиографический список

1. Кривко-Красько А.В. RFM-анализ — инструмент анализа клиентской базы организации // В сб.: Логистические системы и процессы в условиях экономической нестабильности. Материалы V Международной заочной научно-практической конференции. 2017. С. 112–114.
2. Ералева М.Е., Столбикова С.П., Мурзахматов М.А., Детков А.А. Написание кода для RFM-анализа клиентской базы и анализ результатов RFM-сегментации // В сб.: Кибернетика, информатика, аналитика: модели, инструменты, методы. Сборник материалов III международной научно-

- практической конференции студентов, аспирантов и молодых ученых. Донецк, 2024. С. 62–73.
3. Усов Д.О. Механизм использования RFM-анализа для сегментирования клиентской базы компании и проведения маркетинговой кампании // В сб.: Цифровизация общества: состояние, проблемы, перспективы. Материалы VIII ежегодной всероссийской научно-практической конференции. Москва, 2021. С. 136–142.
 4. Шуклина З.Н., Уткина В.А., Прошина О.С. Маркетинговое сегментирование клиентской базы с помощью применения RFM-анализа // Молодой ученый. 2017. № 14 (148). С. 483–488.