

Использование библиотеки mlxtend для создания ассоциативных правил*Анишкова Анастасия Сергеевна**Приамурский государственный университет имени Шолом-Алейхема**Студент***Аннотация**

Цель данного исследования — использование библиотеки mlxtend для создания ассоциативных правил с целью выявления скрытых закономерностей и взаимосвязей между элементами в наборе данных. Работа выполнялась в облачной среде Google Colab с использованием языка Python, что обеспечило удобство и гибкость в проведении экспериментов. Библиотека mlxtend (Machine Learning Extensions) предоставляет широкий набор инструментов для анализа данных и решения задач машинного обучения, включая ансамблирование моделей, выбор признаков, визуализацию результатов и работу с ассоциативными правилами, такими как алгоритм Apriori. Это особенно актуально для задач анализа рыночных корзин и построения рекомендательных систем. Кроме того, mlxtend предлагает инструменты для исследовательской деятельности, такие как анализ графиков обучения и проверка статистических гипотез, что делает её полезной как в академических исследованиях, так и в промышленных приложениях.

Ключевые слова: Google Colab, mlxtend, ассоциативные правила, Python.

Using the mlxtend library to create associative rules*Anishkova Anastasia Sergeevna**Sholom Aleichem Priamurskiy State University**Student***Abstract**

The purpose of this study is to use the mlxtend library to create associative rules in order to identify hidden patterns and relationships between elements in a dataset. The work was performed in the Google Colab cloud environment using Python, which provided convenience and flexibility in conducting experiments. The mlxtend library (Machine Learning Extensions) provides a wide range of tools for data analysis and solving machine learning problems, including model ensembling, feature selection, visualization of results, and working with associative rules such as the Apriori algorithm. This is especially important for the tasks of analyzing market baskets and building recommendation systems. In addition, mlxtend offers research tools such as learning graph analysis and statistical hypothesis testing, which makes it useful in both academic research and industrial applications.

Key words: Google Colab, mlxtend, associative rules, Python.

1 Введение

1.1 Актуальность

Актуальность исследования использования библиотеки mlxtend для создания ассоциативных правил заключается в её практической значимости для анализа данных, простоте использования и широком спектре применений. Такое исследование может внести вклад в развитие методов анализа данных, а также помочь специалистам выбрать наиболее подходящие инструменты для решения своих задач.

1.2 Обзор исследований

В. Ю. Ильичев, С. Д. Кондратьева описали основы использования комбинаторных алгоритмов для анализа товарных рынков. На основе применения библиотеки машинного обучения MLxtend и прочих методов языка Python разработана программа для выявления наиболее часто встречающихся сочетаний товаров в потребительской корзине [1]. Посвятили работу созданию программы, прогнозирующей выбор покупателей в магазинах. Написанный программный код позволяет проанализировать потребительские корзины и на основе обработанных данных осуществить подсчет вероятности выбора того или иного товара и сформулировать прогнозы по единичным продуктам для корректировки плана закупок. Разработанная схема показала свою эффективность, а именно точный и корректный для данного этапа результат. Она может быть полезна как покупателю, получающему персональное предложение, так и продавцу для увеличения объема продаж и повышения доверия клиентов А. И. Соловьева, А. А. Рихтер [2]. А.П. Трещалин, А.А. Шеменев, А.В. Родин, Д.В. Чурбанов рассмотрели задачу классификации орбитальных космических объектов по их спектральным характеристикам. Получен и проведен анализ спектров и фазовых характеристик широко используемых в космической промышленности материалов. Методом главных компонент удалось понизить размерность исходного признакового пространства и разделить спектры различных материалов в новом пространстве. Аналогичный результат получен с помощью кластерного анализа. Была сформулирована и численно решена задача нахождения оптимальных параметров фотометрической системы. Итогом является алгоритм определения параметров фотометрической системы и таблица результатов работы этого алгоритма для систем из 2, 3 и 5 световых фильтров [3]. Классификацию методов машинного обучения в зависимости от способа обучения, рассмотрены основные алгоритмы машинного обучения и область применения. Особое внимание уделено обучению с подкреплением на основе Марковских процессов принятия решений привел Н. Г. Полетаева [4].

1.3 Цель исследования

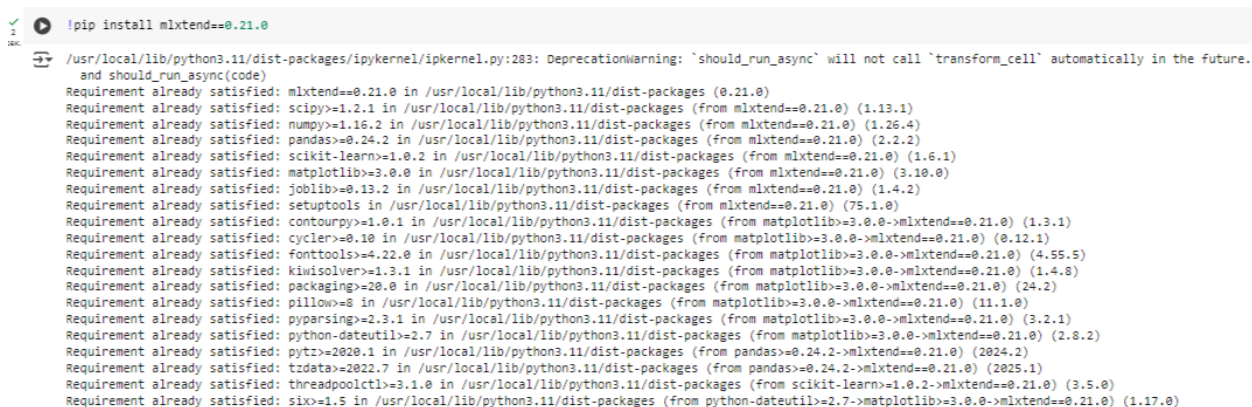
Цель данного исследования — использование библиотеки mlxtend для создания ассоциативных правил с целью выявления скрытых закономерностей и взаимосвязей между элементами в наборе данных.

2 Материалы и методы

В данном исследовании используется среда Google Colab и язык программирования Python.

3 Результаты

Чтобы использовать библиотеку mlxtend (Machine Learning Extensions) в Python, её необходимо предварительно установить. (рис.1).



```

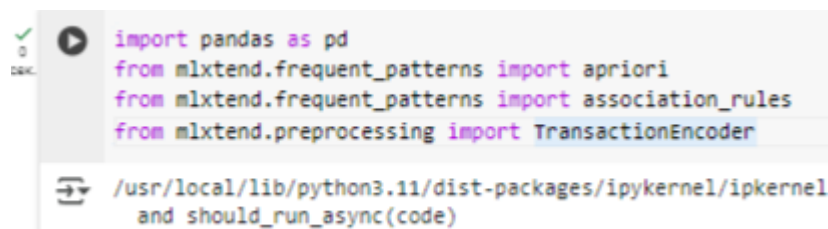
✓ 2
!pip install mlxtend==0.21.0

/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future.
and should_run_async(code)
Requirement already satisfied: mlxtend==0.21.0 in /usr/local/lib/python3.11/dist-packages (0.21.0)
Requirement already satisfied: scipy>=1.2.1 in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (1.13.1)
Requirement already satisfied: numpy>=1.16.2 in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (1.26.4)
Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (2.2.2)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (1.6.1)
Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (3.10.0)
Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (1.4.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from mlxtend==0.21.0) (75.1.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (4.55.5)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0.0->mlxtend==0.21.0) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.24.2->mlxtend==0.21.0) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=0.24.2->mlxtend==0.21.0) (2025.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.0.2->mlxtend==0.21.0) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend==0.21.0) (1.17.0)

```

Рисунок 1. Установка библиотеки

После установки библиотеки mlxtend необходимо импортировать необходимые модули и функции для работы с данными и анализа ассоциативных правил (рис.2): библиотека pandas используется для работы с таблицами и структурированными данными, модуль mlxtend.frequent_patterns позволяет искать частые наборы с помощью алгоритма Apriori и генерировать ассоциативные правила, а модуль mlxtend.preprocessing преобразует данные в формат, подходящий для анализа.



```

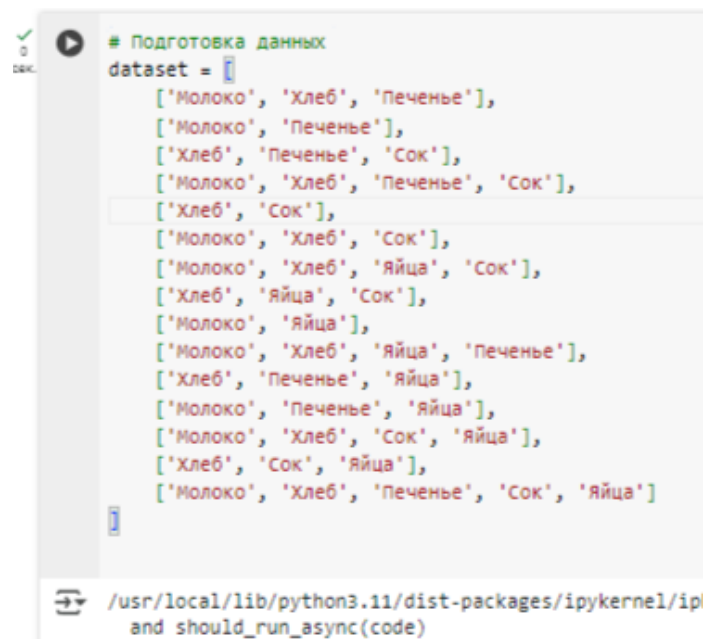
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder

/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel
and should_run_async(code)

```

Рисунок 2. Импортирование библиотек

Для подготовки данных к дальнейшему анализу создадим набор данных (рис.3), представляющий собой список транзакций, где каждая транзакция — это список товаров, купленных вместе.



```

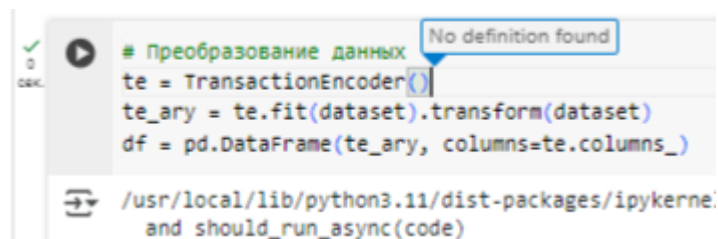
# Подготовка данных
dataset = [
    ['Молоко', 'Хлеб', 'Печенье'],
    ['Молоко', 'Печенье'],
    ['Хлеб', 'Печенье', 'Сок'],
    ['Молоко', 'Хлеб', 'Печенье', 'Сок'],
    ['Хлеб', 'Сок'],
    ['Молоко', 'Хлеб', 'Сок'],
    ['Молоко', 'Хлеб', 'Яйца', 'Сок'],
    ['Хлеб', 'Яйца', 'Сок'],
    ['Молоко', 'Яйца'],
    ['Молоко', 'Хлеб', 'Яйца', 'Печенье'],
    ['Хлеб', 'Печенье', 'Яйца'],
    ['Молоко', 'Печенье', 'Яйца'],
    ['Молоко', 'Хлеб', 'Сок', 'Яйца'],
    ['Хлеб', 'Сок', 'Яйца'],
    ['Молоко', 'Хлеб', 'Печенье', 'Сок', 'Яйца']
]

```

/usr/local/lib/python3.11/dist-packages/ipykernel/ipk... and should_run_async(code)

Рисунок 3. Подготовка данных

Выполняет преобразование данных из списка транзакций в бинарный формат (рис.4), который подходит для анализа с помощью алгоритмов, таких как Apriori.



```

# Преобразование данных
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)

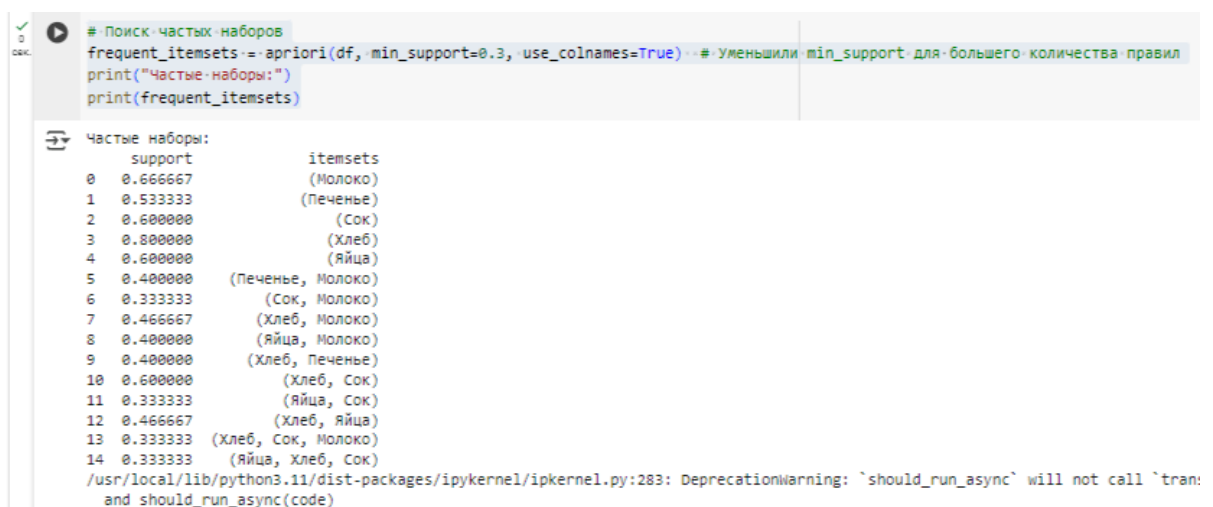
```

No definition found

/usr/local/lib/python3.11/dist-packages/ipykerne... and should_run_async(code)

Рисунок 4. Преобразование данных

Выполним поиск частых наборов товаров с помощью алгоритма Apriori (рис.5).



```

# Поиск частых наборов
frequent_itemsets = apriori(df, min_support=0.3, use_colnames=True)
print("Частые наборы:")
print(frequent_itemsets)

```

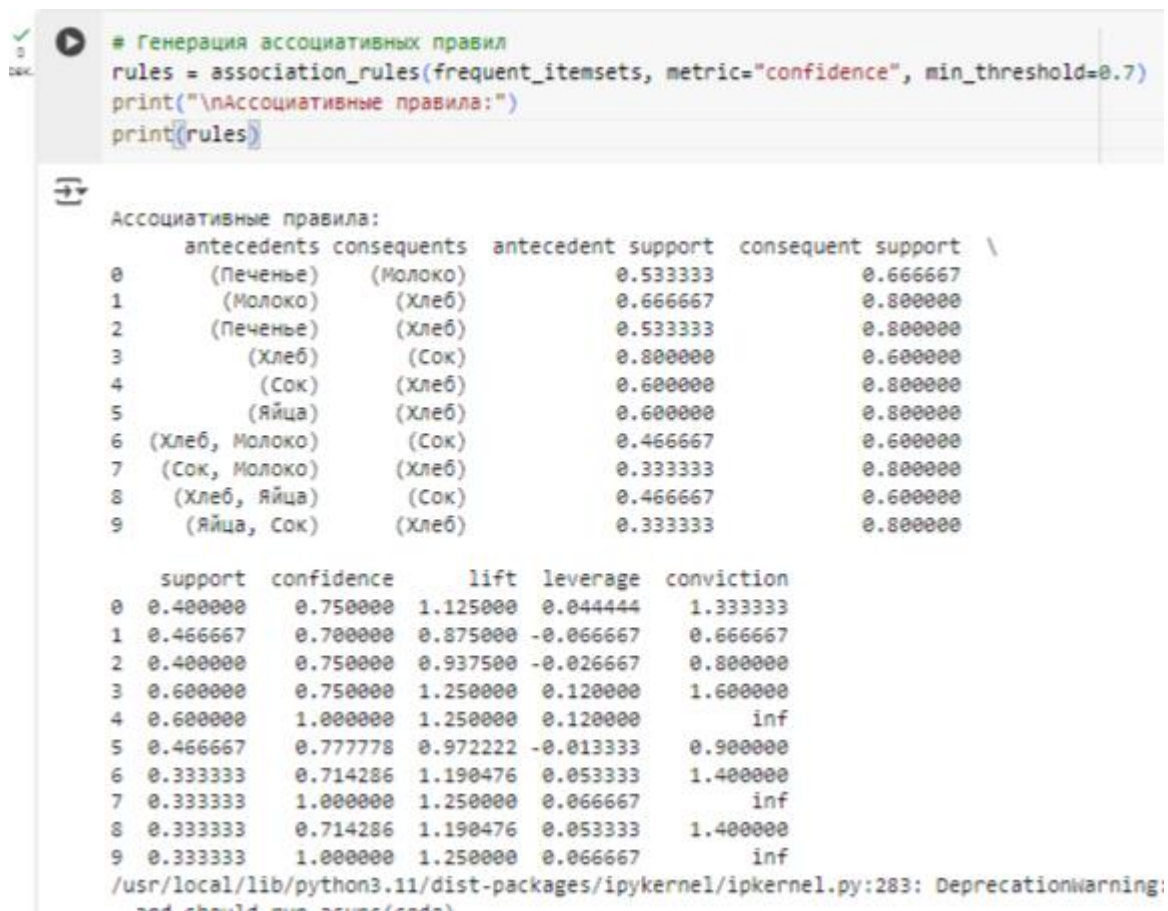
Частые наборы:

support	itemsets
0.666667	(Молоко)
0.533333	(Печенье)
0.600000	(Сок)
0.800000	(Хлеб)
0.600000	(Яйца)
0.400000	(Печенье, Молоко)
0.333333	(Сок, Молоко)
0.466667	(Хлеб, Молоко)
0.400000	(Яйца, Молоко)
0.400000	(Хлеб, Печенье)
0.600000	(Хлеб, Сок)
0.333333	(Яйца, Сок)
0.466667	(Хлеб, Яйца)
0.333333	(Хлеб, Сок, Молоко)
0.333333	(Яйца, Хлеб, Сок)

/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should\_run\_async` will not call `tran... and should_run_async(code)

Рисунок 5. Поиск частых наборов

Сгенерируем ассоциативные правила на основе частых наборов элементов (itemsets) (рис.6).



```
# Генерация ассоциативных правил
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
print("\nАссоциативные правила:")
print(rules)
```

Ассоциативные правила:

	antecedents	consequents	antecedent support	consequent support
0	(Печенье)	(Молоко)	0.533333	0.666667
1	(Молоко)	(Хлеб)	0.666667	0.800000
2	(Печенье)	(Хлеб)	0.533333	0.800000
3	(Хлеб)	(Сок)	0.800000	0.600000
4	(Сок)	(Хлеб)	0.600000	0.800000
5	(Яйца)	(Хлеб)	0.600000	0.800000
6	(Хлеб, Молоко)	(Сок)	0.466667	0.600000
7	(Сок, Молоко)	(Хлеб)	0.333333	0.800000
8	(Хлеб, Яйца)	(Сок)	0.466667	0.600000
9	(Яйца, Сок)	(Хлеб)	0.333333	0.800000

	support	confidence	lift	leverage	conviction
0	0.400000	0.750000	1.125000	0.044444	1.333333
1	0.466667	0.700000	0.875000	-0.066667	0.666667
2	0.400000	0.750000	0.937500	-0.026667	0.800000
3	0.600000	0.750000	1.250000	0.120000	1.600000
4	0.600000	1.000000	1.250000	0.120000	inf
5	0.466667	0.777778	0.972222	-0.013333	0.900000
6	0.333333	0.714286	1.190476	0.053333	1.400000
7	0.333333	1.000000	1.250000	0.066667	inf
8	0.333333	0.714286	1.190476	0.053333	1.400000
9	0.333333	1.000000	1.250000	0.066667	inf

/usr/local/lib/python3.11/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `add_stderr_output` is deprecated and will be removed in a future version of the library. Please use `add_stderr_output` instead.

Рисунок 6. Генерация ассоциативных правил

Визуализируем (рис.7) ассоциативные правила с использованием библиотек `matplotlib` и `seaborn`, это позволяет наглядно представить результаты анализа и упрощает их интерпретацию. С помощью графиков можно быстро оценить распределение правил по ключевым метрикам, таким как поддержка (support) и уверенность (confidence), а также выявить наиболее значимые закономерности в данных.

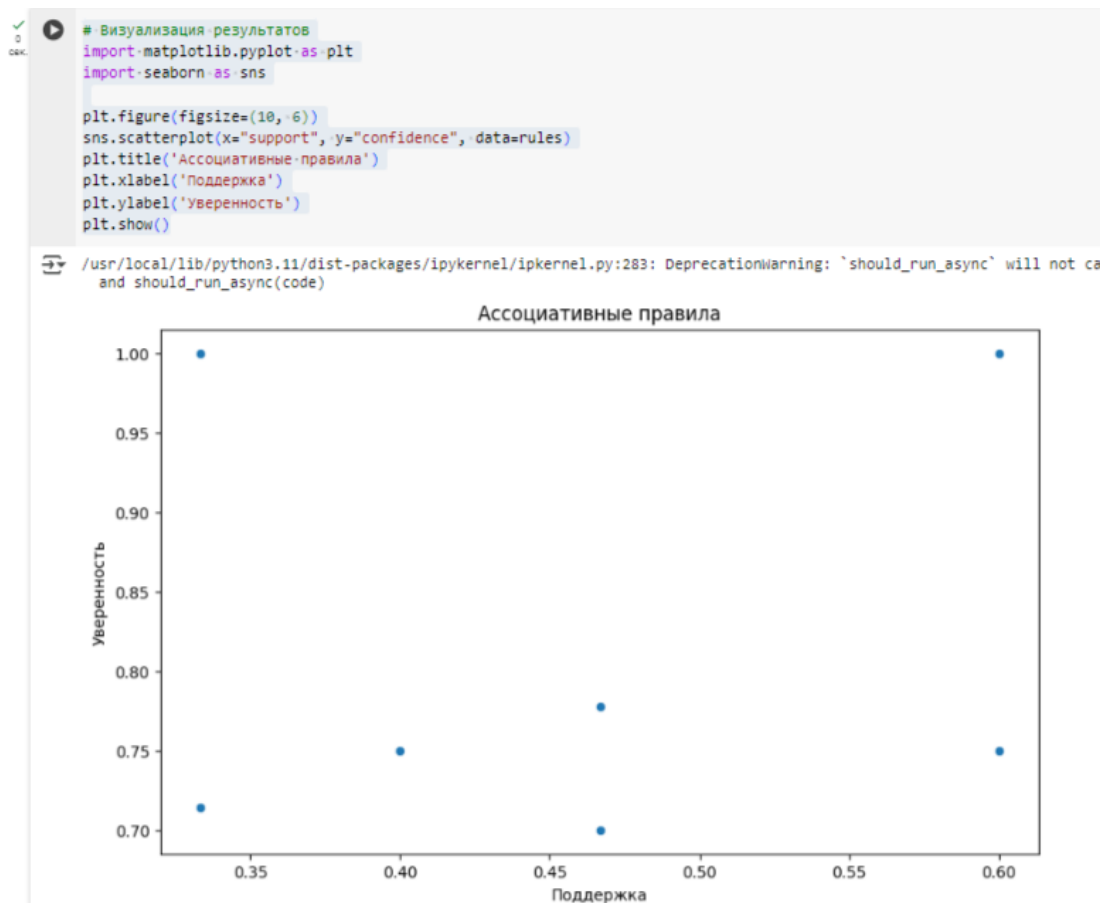


Рисунок 7. Визуализация

Наиболее сильные правила с confidence = 1.0 (100% уверенность) включают Сок→ХлебСок→Хлеб, Сок, Молоко→ХлебСок, Молоко→Хлеб и Яйца, Сок→ХлебЯйца, Сок→Хлеб. Эти правила показывают, что при покупке Сока или его комбинаций с Молоком или Яйцами с вероятностью 100% также покупается Хлеб. Правила с высокой поддержкой, такие как Хлеб→СокХлеб→Сок и Сок→ХлебСок→Хлеб, подтверждают сильную связь между этими продуктами. Значения lift > 1 (например, Хлеб→СокХлеб→Сок, lift = 1.25) указывают на положительную зависимость.

Рекомендации включают размещение Хлеба и Сока рядом на полках, создание акций на комбинации продуктов (например, Сок и Молоко) и анализ правил с низкой уверенностью (например, Молоко→ХлебМолоко→Хлеб) для улучшения их совместного продвижения. Эти выводы помогут оптимизировать ассортимент и повысить продажи.

4 Вывод

Анализ ассоциативных правил выявил несколько значимых закономерностей, которые могут быть использованы для оптимизации ассортимента, планирования маркетинговых акций и улучшения взаимодействия с клиентами. Наиболее сильные правила связаны с продуктами Хлеб и Сок, а также их комбинациями с другими товарами. Эти правила могут стать основой для принятия решений в розничной торговле.

Библиографический список

1. Ильичев В. Ю., Кондратьева С. Д. Использование библиотеки машинного обучения MLxtend для анализа статистики продаж //Системный администратор. 2024. С. 136-140.
2. Соловьева А. И., Рихтер А. А. Анализ магазинных чеков для прогнозирования клиентских предпочтений //Аллея науки. 2019. №. 5 (32). С. 1038.
3. Трещалин А. П. Классификация орбитальных объектов по спектральной информации //Труды Московского физико-технического института. 2020. №. 3 (47). С. 123-140.
4. Полетаева Н. Г. Классификация систем машинного обучения //Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки. 2020. №. 1. С. 5-22.