Министерство науки и высшего образованияРоссийской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра «Электронные вычислительные машины»

Отчет по практическому занятию №5

на тему

«Ассоциативные правила»

по дисциплине  
**«Машинное обучение»**

Выполнили:

Студенты группы №140

Бригада 7

Сафаров Д. А.

Тимохин Е. С.

Проверила:  
ас. Панина И.С.

**Цель работы**

Изучить методы построения ассоциативных правил.

**Задание**

1. Выполнить пример из практической части.

2. Выполнить задания, соответствующие варианту:

1) Увеличить значение поддержки на 0.003.

2) Уменьшить значение достоверности на 0.05.

3) Увеличить значение лифта на 0.8.

После каждого шага фиксировать изменения.

3. Сравнить результаты, полученные в примере и в выбранном варианте, и на основе анализа сделать выводы.

**Практическая часть**

Код программы представлен в приложении А.

0) На рисунках 1-3 представлен результат работы программы без изменений.

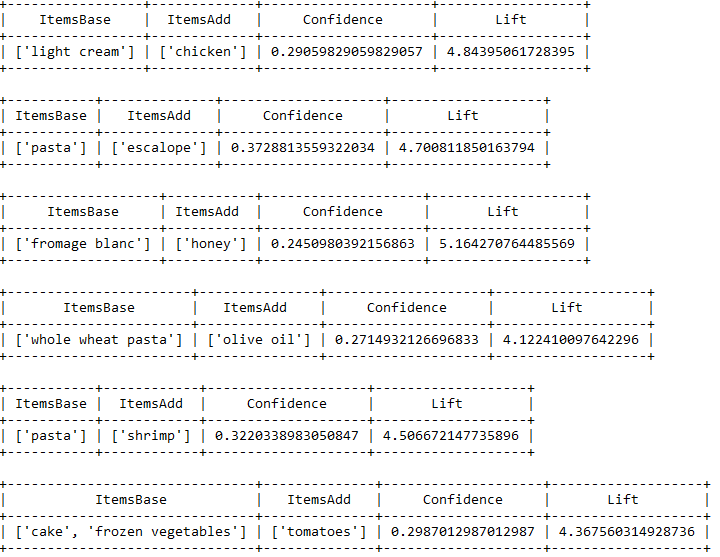


Рисунок 1 – Результат примера, часть 1

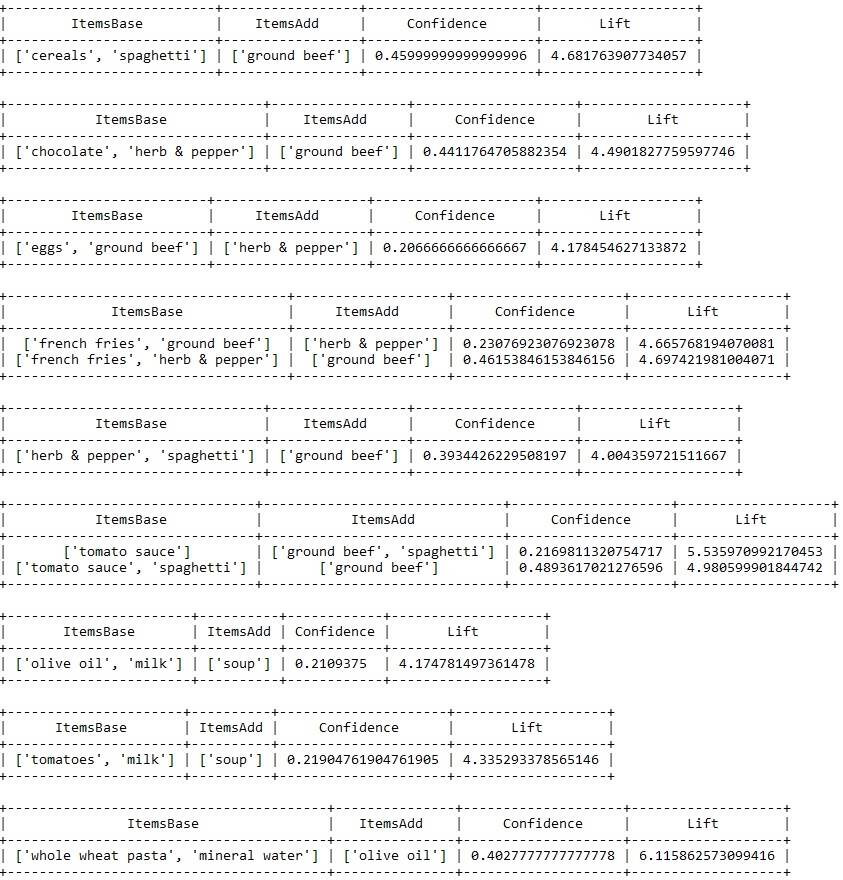
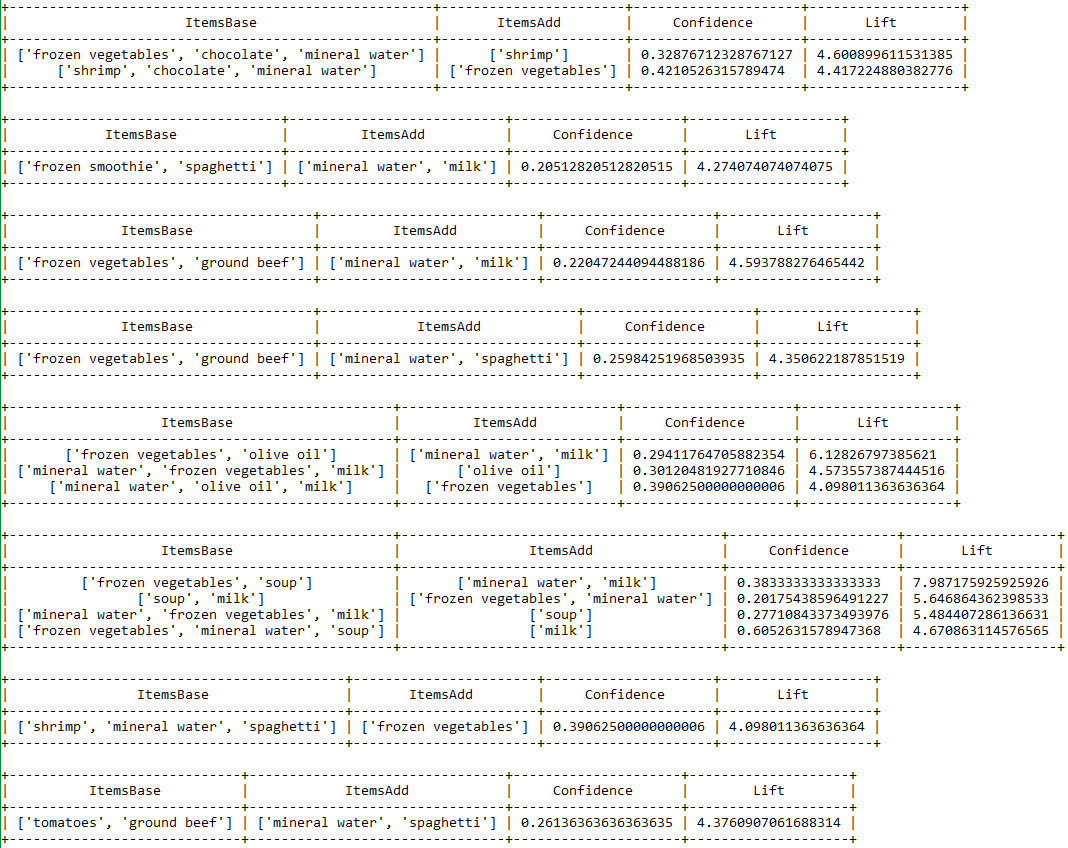
 Рисунок 2 – Результат примера, часть 2

Рисунок 3 – Результат примера, часть 3

1) На рисунке 4 изображён результат работы программы после  
увеличения значения поддержки на 0.003. Результатов сравнительно меньше, и они более логичные.

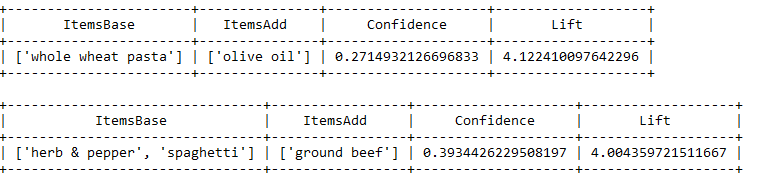


Рисунок 3 – Результат первого действия

2) На рисунках 5-7 изображён результат работы программы после уменьшения значения достоверности на 0.05. Результатов больше, чем до изменения.

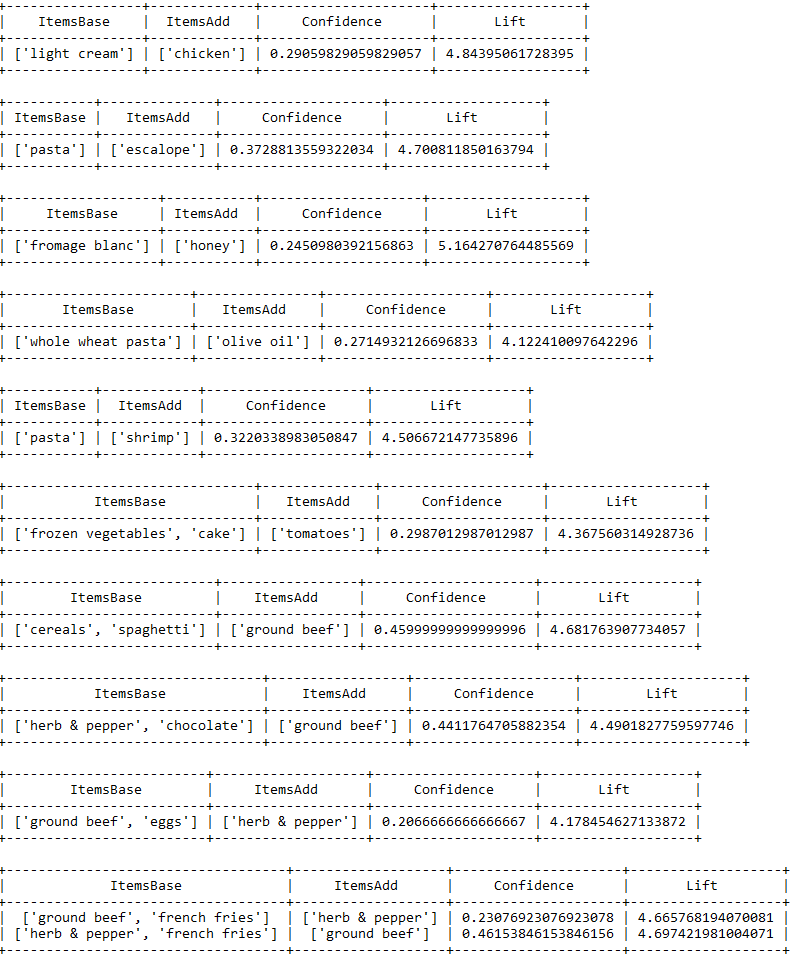
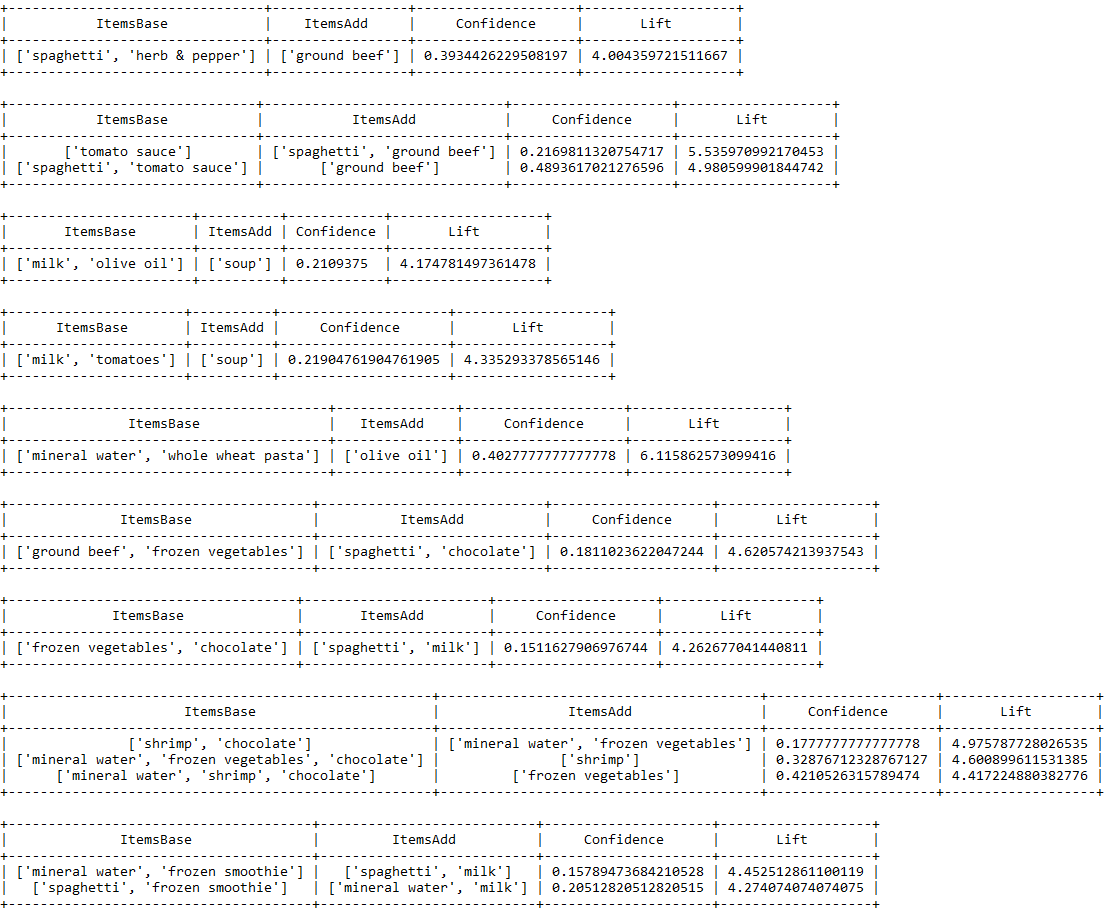


Рисунок 5 – Результат действия 2, часть 1

 Рисунок 6 – Результат действия 2, часть 2

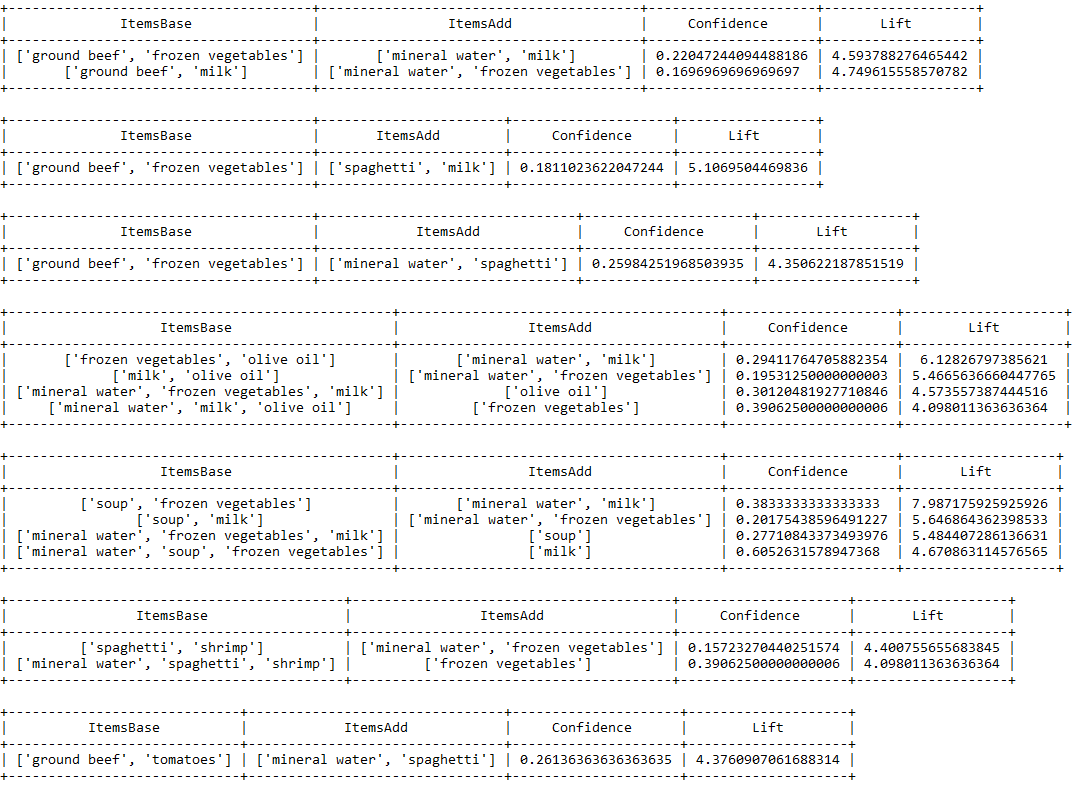


Рисунок 7 – Результат действия 2, часть 3

3) На рисунке 8 изображён результат работы программы после увеличения значения лифта на 0.8. Результатов вновь заметно меньше и нет нелогичных строк.

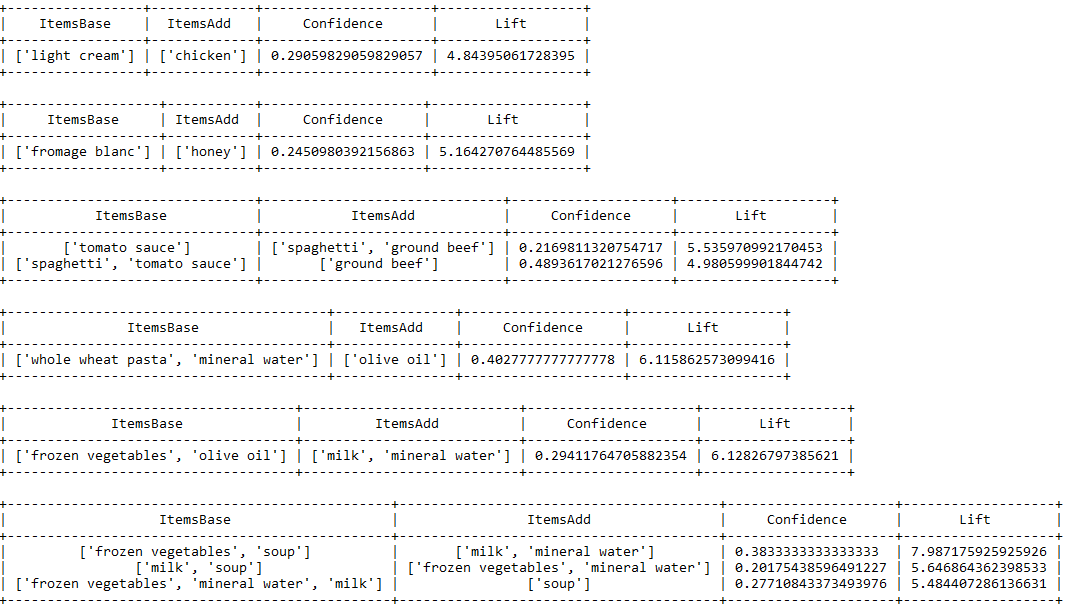


Рисунок 8 – Результат действия 3

Приложение А. Текст программы

import pandas as pd

from io import StringIO

from apyori import apriori as apyori\_apriori

import requests

from tabulate import tabulate

# Загружаем данные из URL

url = "https://raw.githubusercontent.com/adivyas99/Market-Basket-Optimization/master/Market\_Basket.csv"

response = requests.get(url)

if response.status\_code != 200:

print("Не удалось загрузить данные. Код статуса:", response.status\_code)

exit(-1)

# Читаем данные с использованием pandas

data = pd.read\_csv(StringIO(response.text), header=None)

# Заполняем пропуски с использованием ffill()

data.ffill(axis=1, inplace=True)

transactions = []

for i in range(0, len(data)):

transactions.append([str(data.values[i, j]) for j in range(0, len(data.columns))])

# Используем библиотеку apyori для анализа Apriori

result = list(apyori\_apriori(transactions, min\_support=0.003, min\_confidence=0.2, min\_lift=4, min\_length=2))

# Кастомная функция для сериализации

def serialize\_record(record):

return {

'Items': list(record.items),

'Support': record.support,

'OrderedStatistics': [{

'ItemsBase': list(ordered\_stat.items\_base),

'ItemsAdd': list(ordered\_stat.items\_add),

'Confidence': ordered\_stat.confidence,

'Lift': ordered\_stat.lift

} for ordered\_stat in record.ordered\_statistics]

}

# Преобразование столбца 'OrderedStatistics' с помощью пользовательской функции

data\_df = pd.DataFrame([serialize\_record(record) for record in result])

# Отобразим результат в терминале с использованием tabulate

for \_, row in data\_df.iterrows():

print(tabulate(row['OrderedStatistics'], headers='keys', tablefmt='pretty'))

print()