СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc135866922)

[ПРАКТИЧЕСКАЯ РАБОТА №1. ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ 7](#_Toc135866923)

[1.1 Общие сведения 7](#_Toc135866924)

[1.1.1 Список терминов и определений 7](#_Toc135866925)

[1.1.2 Описание бизнес-ролей 8](#_Toc135866926)

[1.2 Требования к системе 8](#_Toc135866927)

[1.2.1 Требования к структуре и функционированию системы 8](#_Toc135866928)

[1.2.2 Требования к численности и квалификации персонала системы 9](#_Toc135866929)

[1.2.3 Показатели назначения 9](#_Toc135866930)

[1.2.4 Требования к надежности 9](#_Toc135866931)

[1.2.5 Требование к безопасности 10](#_Toc135866932)

[1.2.6 Требования к эргономике и технической эстетике 10](#_Toc135866933)

[1.2.7 Требования к транспортабельности для подвижных АС 10](#_Toc135866934)

[1.2.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранения компонентов системы 10](#_Toc135866935)

[1.2.9 Требования к защите информации от несанкционированного доступа 11](#_Toc135866936)

[1.2.10 Требования по сохранности информации при авариях 11](#_Toc135866937)

[1.2.11 Требования к защите от влияния внешних воздействий 11](#_Toc135866938)

[1.2.12 Требования к патентной частоте 11](#_Toc135866939)

[1.2.13 Требования по стандартизации и унификации 11](#_Toc135866940)

[1.2.14 Дополнительные требования 11](#_Toc135866941)

[1.2.15 Требования к функциям (задачам), выполняемым системой 12](#_Toc135866942)

[1.3 Требования к видам обеспечения 12](#_Toc135866943)

[1.3.1 Требования к математическому обеспечению системы 12](#_Toc135866944)

[1.3.2 Требования к информационному обеспечению системы 12](#_Toc135866945)

[1.3.3 Требования к лингвистическому обеспечению системы 13](#_Toc135866946)

[1.3.4 Требования к программному обеспечению системы 13](#_Toc135866947)

[1.3.5 Требования к техническому обеспечению 13](#_Toc135866948)

[1.4 Требования к метрологическому обеспечению 13](#_Toc135866949)

[1.4.1 Требования к организационному обеспечению 13](#_Toc135866950)

[1.4.2 Требования к методическому обеспечению 14](#_Toc135866951)

[ПРАКТИЧЕСКАЯ РАБОТА №2. ПОСТРОЕНИЕ UML-CASE И ОПИСАНИЕ ДИАГРАММЫ 15](#_Toc135866952)

[ПРАКТИЧЕСКАЯ РАБОТА 3. ВЫБОР АРХИТЕКТУРЫ СИСТЕМЫ 17](#_Toc135866953)

[3.1 Общие положения 17](#_Toc135866954)

[3.1.1 Полное наименование системы и ее условное обозначение 17](#_Toc135866955)

[3.1.2 Номер договора (контракта) 17](#_Toc135866956)

[3.1.3 Наименование организаций – Заказчика и Разработчика 17](#_Toc135866957)

[3.1.4 Основания для разработки системы 17](#_Toc135866958)

[3.1.5 Плановые сроки начала и окончания работы по созданию системы 17](#_Toc135866959)

[3.1.6 Источники и порядок финансирования работ 17](#_Toc135866960)

[3.1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы 18](#_Toc135866961)

[3.1.8 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ 18](#_Toc135866962)

[3.1.9 Определения, обозначения и сокращения 19](#_Toc135866963)

[3.1.10 Описание бизнес-ролей 20](#_Toc135866964)

[3.2 Назначение и цели создания системы 20](#_Toc135866965)

[3.2.1 Назначение системы 20](#_Toc135866966)

[3.2.1 Цели создания системы 20](#_Toc135866967)

[3.3 Характеристика объектов автоматизации 21](#_Toc135866968)

[3.3.1 Краткие сведения об объекте автоматизации 21](#_Toc135866969)

[3.3.2 Сведения об условиях эксплуатации объекта автоматизации 21](#_Toc135866970)

[3.4 Требования к системе 21](#_Toc135866971)

[3.4.1 Требования к системе в целом 21](#_Toc135866972)

[3.4.1.2 Требования к структуре и функционированию системы 21](#_Toc135866973)

[3.4.1.3 Требования к численности и квалификации персонала системы 22](#_Toc135866974)

[3.4.1.4 Показатели назначения 22](#_Toc135866975)

[3.4.1.5 Требования к надежности 22](#_Toc135866976)

[3.4.1.6 Требование к безопасности 23](#_Toc135866977)

[3.4.1.7 Требования к эргономике и технической эстетике 23](#_Toc135866978)

[3.4.1.8 Требования к транспортабельности для подвижных АС 23](#_Toc135866979)

[3.4.1.9 Требования к эксплуатации, техническому обслуживанию, ремонту и хранения компонентов системы 23](#_Toc135866980)

[3.4.1.10 Требования к защите информации от несанкционированного доступа 24](#_Toc135866981)

[3.4.1.11 Требования по сохранности информации при авариях 24](#_Toc135866982)

[3.4.1.12 Требования к защите от влияния внешних воздействий 24](#_Toc135866983)

[3.4.1.13 Требования к патентной чистоте 24](#_Toc135866984)

[3.4.1.14 Требования по стандартизации и унификации 24](#_Toc135866985)

[3.4.1.15 Дополнительные требования 24](#_Toc135866986)

[3.4.2 Требования к функциям (задачам), выполняемым системой 25](#_Toc135866987)

[3.4.3 Функциональная структура системы 25](#_Toc135866988)

[3.4.4 Требования к видам обеспечения 26](#_Toc135866989)

[3.4.4.1 Требования к математическому обеспечению системы 26](#_Toc135866990)

[3.4.4.2 Требования к информационному обеспечению системы 26](#_Toc135866991)

[3.4.4.3 Требования к лингвистическому обеспечению системы 26](#_Toc135866992)

[3.4.4.4 Требования к программному обеспечению системы 27](#_Toc135866993)

[3.4.4.5 Требования к техническому обеспечению 27](#_Toc135866994)

[3.4.4.6 Требования к метрологическому обеспечению 27](#_Toc135866995)

[3.4.4.7 Требования к организационному обеспечению 27](#_Toc135866996)

[3.4.4.8 Требования к методическому обеспечению 27](#_Toc135866997)

[3.5 Состав и содержание работ по созданию (развитию) системы 27](#_Toc135866998)

[3.6 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 29](#_Toc135866999)

[3.6.1 Приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ 29](#_Toc135867000)

[3.6.2 Изменения, которые необходимо осуществить в объекте автоматизации 30](#_Toc135867001)

[3.6.3 Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ 30](#_Toc135867002)

[3.6.4 Создание необходимых для функционирования системы подразделений и служб 30](#_Toc135867003)

[3.6.5 Сроки и порядок комплектования штатов и обучения персонала 30](#_Toc135867004)

[3.7 Требования к документированию 30](#_Toc135867005)

[3.8 Источники разработки 31](#_Toc135867006)

[ПРАКТИЧЕСКАЯ РАБОТА № 4. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ МЕТОДОЛОГИИ SADT 32](#_Toc135867007)

[4.1 Цель создания ИС 32](#_Toc135867008)

[4.2 Краткое описание 32](#_Toc135867009)

[4.3 Способ создания ИС 33](#_Toc135867010)

[4.4 Средства создания ИС 33](#_Toc135867011)

[4.5 Проектирование контекстной диаграммы функциональной модели ИС 33](#_Toc135867012)

[4.6 Вывод 34](#_Toc135867013)

[ПРАКТИЧЕСКАЯ РАБОТА № 5. ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ИНФОРМАЦИОННОЙ СИСТЕМЫ В НОТАЦИИ IDEF0 35](#_Toc135867014)

[5.1 Вывод 37](#_Toc135867015)

[ПРАКТИЧЕСКАЯ РАБОТА № 6. Проектирование модели потоков данных в нотации DFD 38](#_Toc135867016)

[6.1 Вывод 39](#_Toc135867017)

[ПРАКТИЧЕСКАЯ РАБОТА № 7. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ И СОЗДАНИЕ ER-ДИАГРАММЫ 40](#_Toc135867018)

[7.1 Цель практической работы 40](#_Toc135867019)

[7.2 Теоретическое введение 40](#_Toc135867020)

[7.3 Задание практической работы: 41](#_Toc135867021)

[7.4 Выполнение практической работы 41](#_Toc135867022)

[7.5 Описание связей: 43](#_Toc135867023)

[7.6 SQL-запросы 43](#_Toc135867024)

[7.7 Выводы по практической работе 44](#_Toc135867025)

[ПРАКТИЧЕСКАЯ РАБОТА № 8. СОЗДАНИЕ ДИАГРАММЫ СОСТОЯНИЙ 46](#_Toc135867026)

[8.1 Вывод 47](#_Toc135867027)

[ПРАКТИЧЕСКАЯ РАБОТА № 9 Описание ЭСЕ 48](#_Toc135867028)

[9.1 Описание ЭСЕ 48](#_Toc135867029)

[9.2 Наполнение системы 48](#_Toc135867030)

[9.3 Математические расчеты 49](#_Toc135867031)

[9.4 Расчет математического ожидания информационного блока системы 49](#_Toc135867032)

[9.5 Расчет дисперсии информационного блока системы 50](#_Toc135867033)

[9.6 Расчет среднеквадратического отклонения 50](#_Toc135867034)

[9.7 Расчет энтропии системы 50](#_Toc135867035)

[9.8 Вывод 51](#_Toc135867036)

[ВЫВОД 52](#_Toc135867037)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 54](#_Toc135867038)

[ГЛОССАРИЙ 56](#_Toc135867039)

[ПРИЛОЖЕНИЕ А 60](#_Toc135867040)

[Список всех элементарных семантических единиц 60](#_Toc135867041)

ВВЕДЕНИЕ

Вводим введение

1. аССОЦИАТИВНЫЕ ПРАВИЛА

1.1 Постановка задачи

Придумать предметную область и на основе выбранной предметной области реализовать алгоритм Apriori.

1.2 Описание алгоритма

Аффинитивный анализ (affinity analysis) – один из распространенных методов Data Mining. Его название происходит от английского слова affinity, которое в переводе означает «близость», «сходство». Цель дангоro метода исследование взаимной связи между событиями, которые происходят совместно. Разновидностью аффинитивного анализа является анализ рыночной корзины, цель которого обнаружить ассоциации между различными событиями, то есть найти правила для количественного описания взаимной связи между двумя или более событиями. Такие правила называются ассоциативными правилами.

Алгоритм Apriori. При практической реализации систем поиска ассоциативных правил используют различные методы, которые позволяют снизить пространство поиска до размеров, обеспечивающих приемлемые вычислительные и временные затраты, например, алгоритм Apriori.

В основе алгоритма Apriori лежит понятие частого набора, который также можно назвать частым предметным набором, часто встречающимся множеством (соответственно, он связан с понятием частоты). Под частотой понимается простое количество транзакций, в которых содержится данный предметный набор. Тогда частыми наборами будут те из них, которые встречаются чаще, чем в заданном числе транзакций.

Примерами приложения ассоциативных правил могут быть следующие задачи:

* + - * выявление наборов товаров, которые в супермаркетах часто покупаются вместе или никогда не покупаются вместе;
      * определение доли клиентов, положительно относящихся к нововведениям в их обслуживании;
      * определение профиля посетителей веб-ресурса;
* определение доли случаев, в которых новое лекарство оказывает опасный побочный эффект.

Следующее важное понятие – предметный набор. Это непустое множество предметов, появившихся в одной транзакции.

Анализ рыночной корзины – это анализ наборов данных для определения комбинаций товаров, связанных между собой, иными словами, производится поиск товаров, присутствие которых в транзакции влияет на вероятность наличия других товаров или комбинаций товаров.

Современные кассовые аппараты в супермаркетах позволяют собирать информацию о покупках, которая может храниться в базе данных. Затем накопленные данные могут использоваться для построения систем поиска ассоциативных правил.

Поддержка ассоциативною правила – это число транзакций, которые содержат каr условие, так и следствие.

Например, для ассоциации 𝐴→𝐵 можно записать:

Достоверность ассоциативного правила 𝐴→𝐵 представляет собой меру точности правила и определяется как отношение количества транзакций, содержащих и условие, и следствие, к количеству транзакций, содержащих только условие:

Если поддержка и достоверность достаточно высоки, можно с большой вероятностью утверждать, что любая будущая транзакция, которая включает условие, будет также содержать и следствие.

Лифт – отношение частоты появления условия в транзакциях, которые также содержат и следствие к частоте появления следствия в целом. Значения лифта большие 1 показывают, что условие чаще появляется в транзакциях, содержащих следствие, чем в остальных. Можно утверждать, что лифт является обобщенной мерой связи двух предметных наборов: при значениях лифта больше 1 связь положительная, при 1 она отсутствует, а при значениях меньше 1 – отрицательная.

Лифт (оригинальное название — интерес) вычисляется следующим образом:

**1.3 Описание предметной области**

Исходные данные из магазина автозапчастей представленные (Рисунок 1.1).

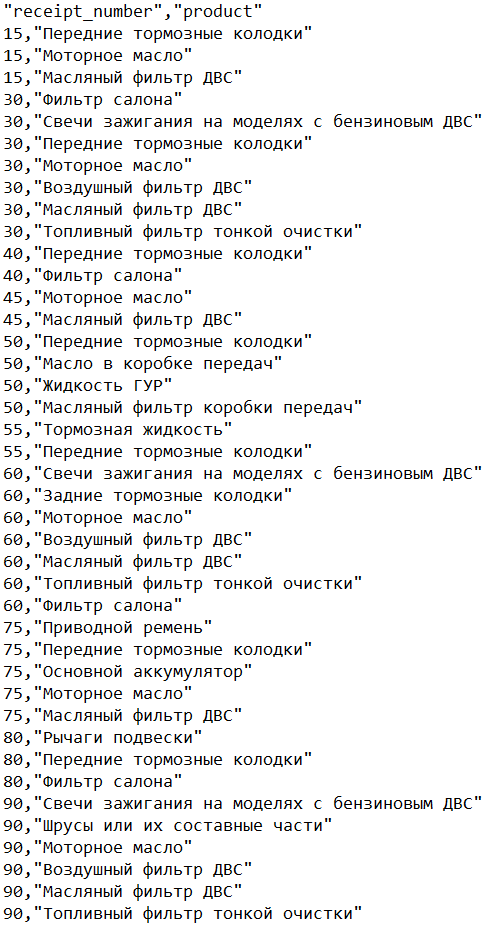


Рисунок 1.1 – Исходные данные

1.4 Ассоциотивные правила в программе deductor

В программе Deductor Визуализатор "Правила" отображает ассоциативные правила в виде списка правил. Этот список представлен таблицей со столбцами: "Номер правила", "Условие", "Следствие", "Поддержка, %", "Поддержка, Количество", "Достоверность", "Лифт" (Рисунок 2).

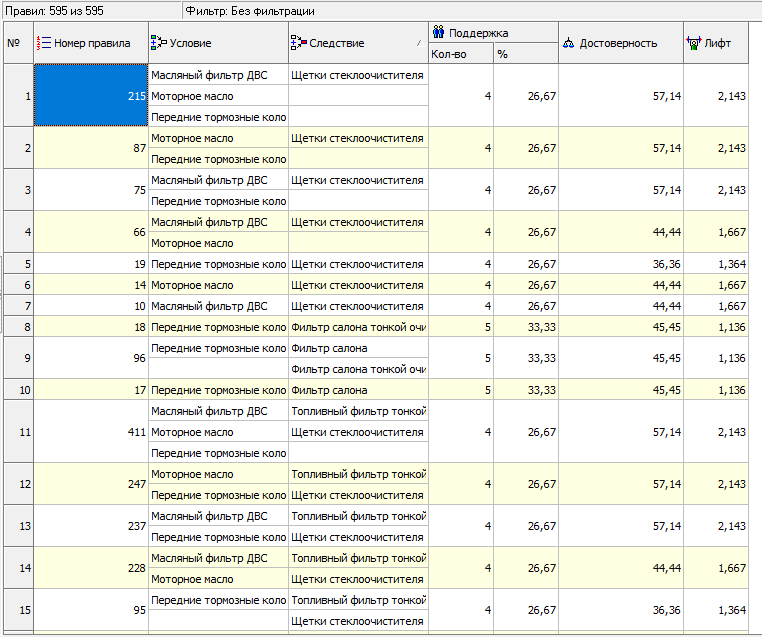


Рисунок 1.2 – Анализ данных в программе Deductor

* 1. Расчёт поддержки

Возьмем ассоциацию передние тормозные колодки и моторное масло. Поскольку количество транзакций, содержащих как передние тормозные колодки, так и моторное масло, равно 7, а общее число транзакций 15, то поддержка данной ассоциации будет:

1.6 Расчёт достоверности

Возьмем ассоциацию передние тормозные колодки и моторное масло.

Поскольку количество транзакций, содержащих только моторное масло (условие), равно 4, то достоверность данной ассоциации будет:

1.7 Расчёт лифта

*Лифт* – отношение частоты появления условия в транзакциях, которые также содержат и следствие к частоте появления следствия в целом.

1.8 Программная реализация

Создадим ассоциативные правила в программной реализации на языке высокого уровня Python [2].

Полученный результат работы программы ассоциативных правил (Рисунок 1.3).



Рисунок 1.3 – Результат работы программы

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сорокин А. Б. Лекции и методические материалы — РТУ МИРЭА, 2023.
2. Python documentation — URL: https://www.python.org/ (Дата обращения: 20.12.2022)
3. Matplotlib — Visualization with Python — URL: https://matplotlib.org/ (Дата обращения: 21.12.2022)
4. Информационные технологии, № 7, 2012. Теоретический и прикладной научно-технический журнал. — М.: Новые технологии, 2012. — 80 с.
5. Настройка весовых матриц ЗСУР регулятора с помощью биоинспирированных алгоритмов оптимизации. Вестник РГРТУ. 2016. № 55. — Рязань: РГРТУ. — С. 131–139.

ВЫВОД

Информационная система, разработанная для web приложения по поиску авиабилетов, представляет собой комплекс программных средств, который обеспечивает автоматизированный процесс покупки билетов. Система должна соответствовать требованиям безопасности, обеспечивать быстрое и удобное использование, а также поддерживать управление данными о клиентах и транзакциях.

Для создания данной системы необходимо провести предварительный анализ требований к проектируемой системе. В процессе работы над проектом необходимо разработать диаграмму прецедентов для одного из классов или прецедентов проектируемой информационной системы. Важным шагом является уточнение требований, созданных на предыдущих этапах, на основании макета информационной системы.

Для проектирования функциональной модели информационной системы необходимо выбрать цель создания ИС, способ и средства создания, а также использовать нотацию IDEF0. На следующем этапе необходимо продолжить проектирование функциональной модели и провести моделирование двух уровней декомпозиции в нотации IDEF0, составив текстовое описание проектируемых модулей на уровнях декомпозиции.

Для выбранного функционального блока нижнего уровня декомпозиции необходимо выполнить декомпозицию в нотации DFD. Для более полного понимания и формализации заданной предметной области необходимо создать модель «сущность – связь» в нотации ERD.

Важным аспектом разработки информационной системы является создание диаграммы состояний проектируемой информационной системы с использованием соответствующей нотации языка UML. Наконец, необходимо закрепить имеющиеся знания о параметрах ИС и изучить методологию расчета требуемых параметров проектируемой информационной системы, а также приобрести навыки анализа и формализованного описания заданной предметной области.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32−2107. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления": введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст: дата введения 2018-01-07. – URL: http://www.consultant.ru/document/cons\_doc\_LAW\_292293/ (дата обращения: 12.04.2023). – Текст: электронный.

2. ГОСТ Р 7.0.100–2018. Национальный стандарт Российской Федерации. Библиографическая запись. Библиографическое описание: общие требования и правила составления: Приказом Федерального агентства по техническому регулированию и метрологии (Росстандартом) от 03 декабря 2018 N 1050-ст: дата введения 2019-01-07– М.: Стандартинформ, 2018. – 128 с.

3. ГОСТ 19.701−90 ЕСПД (ИСО5807−85). Межгосударственный стандарт. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения: Постановлением Государственного комитета СССР по управлению качеством продукции и стандартам N 3294: дата введения 1990-01-01. – М.: Стандартинформ, 2010 – 158 с.

4. ГОСТ 34.602−2020. Межгосударственный стандарт. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы : Приказом Федерального агентства по техническому регулированию и метрологии № 1522-ст от 19 ноября 2021 г.: дата введения 2022-01-01. – М.: Российский институт стандартизации, 2021. – 12 с.

5. Жизненный цикл информационной системы. Этап формирования требований к ИС: сайт. – URL: https://www.sites.google.com/site/metodsybd/blok-5-etapy-ziznennogo-cikla/5-1-etap-formirovania-trebovanij-k-is (дата обращения: 03.04.2023). – Текст: электронный.

6. Жизненный цикл информационной системы. Проектирование ИС: сайт. – URL: https://www.sites.google.com/site/metodsybd/blok-5-etapy-ziznennogo-cikla/5-2-proektirovanie-is (дата обращения: 10.04.2023). – Текст: электронный.

ПРИЛОЖЕНИЯ

Приложение А – Листинг кода для ассоциативных правил

Приложение А

Листинг кода для ассоциативных правил

Листинг А.1 – Используемые библиотеки

import pandas as pd

import numpy as np

import seaborn as sn

import itertools

import time

Листинг А.2 – Файл txt с данными для анализа

"receipt\_number","product"

15,"Передние тормозные колодки"

15,"Моторное масло"

15,"Масляный фильтр ДВС"

30,"Фильтр салона"

30,"Свечи зажигания на моделях с бензиновым ДВС"

30,"Передние тормозные колодки"

30,"Моторное масло"

30,"Воздушный фильтр ДВС"

30,"Масляный фильтр ДВС"

30,"Топливный фильтр тонкой очистки"

40,"Передние тормозные колодки"

40,"Фильтр салона"

45,"Моторное масло"

45,"Масляный фильтр ДВС"

50,"Передние тормозные колодки"

50,"Масло в коробке передач"

50,"Жидкость ГУР"

50,"Масляный фильтр коробки передач"

55,"Тормозная жидкость"

55,"Передние тормозные колодки"

60,"Свечи зажигания на моделях с бензиновым ДВС"

60,"Задние тормозные колодки"

60,"Моторное масло"

60,"Воздушный фильтр ДВС"

60,"Масляный фильтр ДВС"

60,"Топливный фильтр тонкой очистки"

60,"Фильтр салона"

75,"Приводной ремень"

75,"Передние тормозные колодки"

75,"Основной аккумулятор"

75,"Моторное масло"

75,"Масляный фильтр ДВС"

80,"Рычаги подвески"

80,"Передние тормозные колодки"

80,"Фильтр салона"

90,"Свечи зажигания на моделях с бензиновым ДВС"

90,"Шрусы или их составные части"

90,"Моторное масло"

90,"Воздушный фильтр ДВС"

90,"Масляный фильтр ДВС"

90,"Топливный фильтр тонкой очистки"

Листинг А.3 – Функция main

data = pd.read\_csv('static/data//new\_data.csv')  
data.hist()  
unique\_receipts = data.receipt\_number.unique()

print(len(unique\_receipts))

unique\_receipts  
count\_receipt = len(unique\_receipts)  
unique\_products = data['product'].unique()

print(len(unique\_products))

unique\_products  
data\_np = data.to\_numpy()  
group\_products\_receipts = []

for i in unique\_receipts:

micro\_data = []

for j in data\_np:

if j[0] == i:

micro\_data.append(j[1])

group\_products\_receipts.append(micro\_data)

Листинг А.4 – Функция создание всех сочетаний

mass\_group\_products = []

mass\_group\_products\_str = []

count = 1

last\_len = 0

for mass in group\_products\_receipts:

start\_time = time.time()

print('-' \* 25)

print(f'Элемент {count} / {len(group\_products\_receipts)}')

print('Количество элементов =', len(mass))

if len(mass) > 4:

n = 4

else:

n = len(mass)

print(f'Максимум элементов = {n}')

groups = []

for count\_item in range(1, n + 1):

permutation = itertools.permutations(mass, count\_item)

comb\_not\_sort = []

for comb in permutation:

groups.append(list(comb))

for i in range(len(groups)):

for j in range(len(groups)):

if i != j and set(groups[j]).isdisjoint(groups[i]) and set(groups[i]).isdisjoint(groups[j]):

if sum([groups[i], groups[j]], []) not in mass\_group\_products\_str:

mass\_group\_products\_str.append(sum([groups[i], groups[j]], []))

mass\_group\_products.append([groups[i], groups[j]])

print("Время = %s seconds" % (time.time() - start\_time))

print(f"Получено сочетаний {len(mass\_group\_products) - last\_len}")

print(f"Всего элементов = {len(mass\_group\_products)}")

count += 1

last\_len = len(group\_products\_receipts)

print('\n' \* 1)

print(len(mass\_group\_products))

Листинг А.5 – Функция расчёта поддержки

for i in range(len(mass\_group\_products)):

count = 0

for j in group\_products\_receipts:

if set(mass\_group\_products[i][0]).issubset(j) and set(mass\_group\_products[i][1]).issubset(j):

count += 1

if i == 1:

print(f'{count\*100}/{count\_receipt} = {round((count\*100/count\_receipt),2)}')

mass\_group\_products[i].append((round((count\*100/count\_receipt),2)))

Листинг А.6 – Функция расчёта достоверности

for i in range(len(mass\_group\_products)):

count\_one = 0

count\_two = 0

for j in group\_products\_receipts:

if set(mass\_group\_products[i][0]).issubset(j) and set(mass\_group\_products[i][1]).issubset(j):

count\_one += 1

for j in group\_products\_receipts:

if set(mass\_group\_products[i][0]).issubset(j):

count\_two += 1

if i == 1:

print(f'{count\_one}/{count\_two} = {round((count\_one\*100/count\_two),2)}')

if count\_two != 0:

mass\_group\_products[i].append((round((count\_one\*100/count\_two),2)))

else:

mass\_group\_products[i].append(0)

Листинг А.7 – Функция расчёта лифта

for i in range(len(mass\_group\_products)):

count\_one = 0

for j in group\_products\_receipts:

if set(mass\_group\_products[i][1]).issubset(j):

count\_one += 1

if i == 1:

print(f'{count\_one}/{count\_receipt} = {round((count\_one\*100/count\_receipt),2)}')

if (count\_one\*100/count\_receipt) != 0:

mass\_group\_products[i].append((round(mass\_group\_products[i][-1]/(count\_one\*100/count\_receipt),2)))

else:

mass\_group\_products[i].append(0)

Листинг А.8 – Функция получения финальных данных

Dataframe = pd.DataFrame(dataframe, columns =['Suported', 'Reliability', 'Lift'], index=name\_for\_index)

Dataframe = (Dataframe.loc[Dataframe.Suported < 89])

Dataframe = (Dataframe.loc[20 < Dataframe.Suported])

Dataframe = (Dataframe.loc[30 < Dataframe.Reliability])

Dataframe = (Dataframe.loc[Dataframe.Reliability < 70])  
Dataframe.sort\_values('Lift', ascending=False)