**Федеральное государственное образовательное**

**бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ**

**ФЕДЕРАЦИИ»**

**(Финансовый университет)**

**Факультет**

**информационных технологий и анализа больших данных**



**Кафедра «Бизнес-информатика»**

**Расчетно-аналитическая работа по темам:**

«Метод наименьших квадратов», «Многокритериальная оптимизация», «Экспертная оценка», «Системы массового обслуживания»

Студентка группы БИ 20-8:

Смолина Екатерина Алексеевна

Руководитель:

Аксенов Дмитрий Андреевич

**Москва 2022**

**Оглавление**

[1. Метод наименьших квадратов 4](#_Toc105936180)

[1.1.Постановка задачи (физическая модель) 4](#_Toc105936181)

[1.2.Математическая модель 6](#_Toc105936182)

[1.3. Алгоритм решения 7](#_Toc105936183)

[1.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel задачи прогнозирования 7](#_Toc105936184)

[1.3.1.1. Описание входных данных 7](#_Toc105936185)

[1.3.1.2. Описание алгоритма решения 7](#_Toc105936186)

[1.3.1.3. Описание выходных данных 13](#_Toc105936187)

[1.3.2. Алгоритм 2. Решение в Python 13](#_Toc105936188)

[1.3.2.1. Описание входных данных 13](#_Toc105936189)

[1.3.2.2. Описание алгоритма решения 13](#_Toc105936190)

[1.3.2.3. Описание выходных данных 15](#_Toc105936191)

[1.4. Варианты использования алгоритмов 15](#_Toc105936192)

[1.4.1. Варианты использования алгоритма 1 – поиск решений Excel задачи прогнозирования 15](#_Toc105936193)

[1.4.2. Варианты использования алгоритма 2 – Python для аппроксимации линейной и квадратичной функциями 17](#_Toc105936194)

[1.5. Архитектура решения алгоритма на языке программирования Python 21](#_Toc105936195)

[1.6. Тестирование 24](#_Toc105936196)

[1.7. Заключение 29](#_Toc105936197)

[2. Многокритериальная оптимизация 30](#_Toc105936198)

[2.1. Постановка задачи (физическая модель) 30](#_Toc105936199)

[2.2. Математическая модель 32](#_Toc105936200)

[2.3. Алгоритм решения 33](#_Toc105936201)

[2.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel задачи оптимизации 33](#_Toc105936202)

[2.3.1.1. Описание входных данных 33](#_Toc105936203)

[2.3.1.2. Описание алгоритма решения 34](#_Toc105936204)

[2.3.1.3. Описание выходных данных 42](#_Toc105936205)

[2.3.2. Алгоритм 2. Решение в Python 42](#_Toc105936206)

[2.3.2.1. Описание входных данных 42](#_Toc105936207)

[2.3.2.2. Описание алгоритма решения 43](#_Toc105936208)

[2.3.2.3. Описание выходных данных 44](#_Toc105936209)

[2.4. Варианты использования алгоритмов 44](#_Toc105936210)

[2.4.1. Варианты использования алгоритма 1 – поиск решений Excel 44](#_Toc105936211)

[2.4.2. Варианты использования алгоритма 2 – Python 47](#_Toc105936212)

[2.5. Архитектура решения алгоритма на языке программирования Python 57](#_Toc105936213)

[2.6. Тестирование 58](#_Toc105936214)

[2.7 Заключение 72](#_Toc105936215)

[3. Экспертная оценка 73](#_Toc105936216)

[3.1. Постановка задачи (физическая модель) 73](#_Toc105936217)

[3.2.Математическая модель 75](#_Toc105936218)

[3.3. Алгоритм решения 79](#_Toc105936219)

[3.3.1.1. Описание входных данных 79](#_Toc105936220)

[3.3.1.2. Описание алгоритма решения 79](#_Toc105936221)

[3.3.1.3. Описание выходных данных 88](#_Toc105936222)

[3.3.2. Алгоритм 2. Решение в Python 88](#_Toc105936223)

[3.3.2.1. Описание входных данных 88](#_Toc105936224)

[3.3.2.2. Описание алгоритма решения 88](#_Toc105936225)

[3.3.2.3. Описание выходных данных 90](#_Toc105936226)

[3.4. Варианты использования алгоритмов 91](#_Toc105936227)

[3.4.1. Варианты использования алгоритма 1 – поиск решений Excel 91](#_Toc105936228)

[3.4.2. Варианты использования алгоритма 2 – Python 97](#_Toc105936229)

[3.5. Архитектура решения алгоритма на языке программирования Python 104](#_Toc105936230)

[3.6. Тестирование 107](#_Toc105936231)

[3.7 Заключение 117](#_Toc105936232)

[4. Системы массового обслуживания 118](#_Toc105936233)

[4.1. Постановка задачи (физическая модель) 118](#_Toc105936234)

[4.2. Математическая модель 120](#_Toc105936235)

[4.3. Алгоритм решения 123](#_Toc105936236)

[4.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel задачи прогнозирования 123](#_Toc105936237)

[4.3.1.1. Описание входных данных 123](#_Toc105936238)

[4.3.1.2. Описание алгоритма решения 124](#_Toc105936239)

[4.3.1.3. Описание выходных данных 129](#_Toc105936240)

[4.3.2. Алгоритм 2. Решение в Python 130](#_Toc105936241)

[4.3.2.1. Описание входных данных 130](#_Toc105936242)

[4.3.2.2. Описание алгоритма решения 131](#_Toc105936243)

[4.3.2.3. Описание выходных данных 132](#_Toc105936244)

[4.4. Варианты использования алгоритмов 132](#_Toc105936245)

[4.4.1. Варианты использования алгоритма 1 – поиск решений Excel 132](#_Toc105936246)

[4.4.2. Варианты использования алгоритма 2 – Python 135](#_Toc105936247)

[4.5. Архитектура решения алгоритма на языке программирования Python 137](#_Toc105936248)

[4.6. Тестирование 138](#_Toc105936249)

[4.7. Заключение 144](#_Toc105936250)

# **1. Метод наименьших квадратов**

# **1.1.Постановка задачи (физическая модель)**

Наш заказчик – Бракоразводный центр №1, российская компания, занимающаяся оформлением разводов. В данной компании работают юристы, за плечами которых большой опыт, они полностью берут на себя: составление искового заявления, определение подсудности, составление соглашения второго супруга и ходатайства о рассмотрении дела в отсутствие клиента, сбор и оформление всех документов в дело о принятии судом к производству о бракоразводном процессе, представление интересов на неограниченном количестве судебных заседаний до окончательного вынесения решения суда о расторжении брака.

Как мы видим, Бракоразводный центр № 1 предоставляет широкий спектр услуг и успешно обеспечивает полное сопровождения клиента в бракоразводном процессе. В данном центре работает около 10 ведущих специалистов.

После поручения Михаила Мишустина, председателя правительства РФ, в котором требовалось обеспечить условия, чтобы услуги бракоразводных компаний можно было получить в дистанционном формате, и увеличения процента разводов по РФ за последний год, директор выявил необходимость в найме большего количества юристов.

Именно поэтому компания Бракоразводный центр № 1 обратилась к нам с задачей спрогнозировать количество разводов в ближайшие несколько лет, чтобы понять, сколько необходимо нанять новых юристов, чтобы обеспечить оказание услуг на высоком уровне всем клиентам данного центра.

Также нашим заказчиком была предоставлена официальная статистика разводов в России на 1000 человек населения, рисунок 1.1, для проведения наиболее точного прогнозирования.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.1 – Статистические данные количества разводов на 1000 человек населения с 1950 года.

# **1.2.Математическая модель**

Рассмотрим математическую модель Аппроксимации.

– прогноз

(1.1)

(1.2)

Если – аппроксимация

Если – регрессия

(1.3)

(1.4)

# **1.3. Алгоритм решения**

## **1.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel задачи прогнозирования**

### **1.3.1.1. Описание входных данных**

В качестве входных данных мы имеем:

* статистические значения разводов. В качестве значений переменной х, в данном случае, мы взяли перечень годов с 1950 по 2021, и в качестве значений переменной y - количество разводов в России на 1 000 человек населения в зависимости от представленных годов.

### **1.3.1.2. Описание алгоритма решения**

Алгоритм принимает на вход значения переменных по x и y. Далее, используя встроенные функции Excel и формулы, описанные в математической модели, находятся средние значения, истинная зависимость, отклонения, квадратические отклонения, дисперсии, среднеквадратические отклонения, доверительные интервалы, строятся точечные диаграммы для фильтрации и отображения линий тренда и доверительных интервалов, как для изначальных статистических данных, так и для отфильтрованных.

*Подробный пример реализации алгоритма*

Шаг 1: в соответствии с описанием задачи составляем таблицу со значениями x и y. Год мы берем по x и количество разводов по y. (см рис 1.2)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.2 Исходные значения статистики

Шаг 2: находим средние значения по x и y, использую встроенную функцию Excel СРЗНАЧ() (см рис 1.3)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.3 – Поиск средних значений

Шаг 3: далее мы умножаем значения по x и y, возводим значения х в квадрат, растягиваем формулу по ячейкам и по полученные значениям ищем их среднее. Данные действия представлены на Рисунке 1.4:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.4 – Промежуточные расчеты

Шаг 4: ищем значения переменных a и b, по формулам (1.16) и (1.17), чтобы найти решение системы линейных уравнений относительно 𝑎 и 𝑏. (см рис 1.5)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.5 – Поиск значений a и b

Шаг 5: находим тренд по формуле: – истинная зависимость (тренд), как показано на рисунке 1.6.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.6 – Поиск значений тренда

Шаг 6: далее мы находим отклонение и квадратическое отклонение, чтобы найти дисперсию и среднеквадратическое отклонение. (см рис 1.7)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.7 – Отклонение и квадратическое отклонение

Для самопроверки можно использовать встроенную функцию Excel СУММ() и сложить все значения отклонения, и сумма данных значений должна равняться нулю, так как расстояние точек над и под линией тренда должны быть равны. Также необходимо просуммировать и квадратическое отклонение.

Шаг 7: находим дисперсию, по формуле (1.19), и среднеквадратическое отклонение, как корень из дисперсии, как показано на рисунке 1.8.

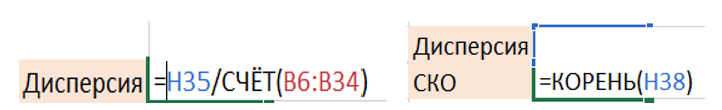


Рисунок 1.8 – Значения дисперсии и среднеквадратического отклонения

Отметим, что поиск дисперсии осуществляется через формулу: СУММ(квадратических отклонений)/СЧЁТ(x). СЧЁТ() также является встроенной функцией Excel, как и СУММ(). Значение среднеквадратического отклонения найден как корень из дисперсии.

Шаг 8: далее мы создаем точечную диаграмму с +- 1,5 сигма для последующей фильтрации данных.

Сначала находим значения +– 1,5 сигма, как показано на рисунке 1.9.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.9 – Значения +– 1,5 сигма

Далее строим точечную диаграмму для фильтрации выпадающих точек и ошибок измерений. Необходимо выбрать источники данных, как это показано на рисунке 1.10.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.10 – Построение точечной диаграммы

Диаграмма построена, теперь мы сможем произвести фильтрацию данных статистики. Как мы видим, некоторые точки выходят за линии сигмы 1,5 (как точка «2002»), они и есть выпадающие.

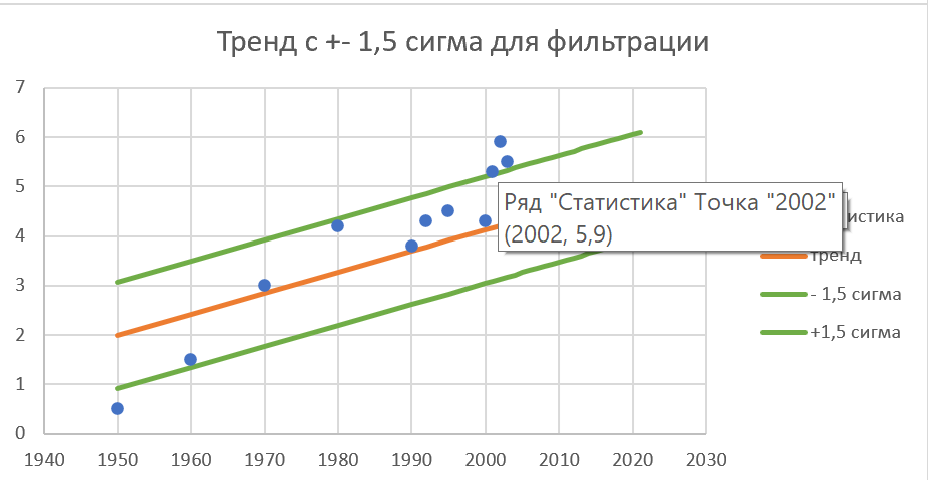


Рисунок 11 – Точечная диаграмма для фильтрации

Шаг 9: производим фильтрацию данных. Через точечную диаграмму производим поиск выпадающих точек и удаляем их. (см. рисунки 1.12 и 1.13)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.12 – Выделенные выпадающие точки

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.13 – Отфильтрованные значения

Шаг 10: после того, как мы получили отфильтрованные значения, мы проделываем заново шаги 2–8 уже на отфильтрованных значениях. Результат представлен на рисунке 1.14.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.14 – Найденные значения для отфильтрованных значений статистики

Шаг 11: далее строим доверительные интервалы для +– 1 сигма, +– 2 сигма и +– 3 сигма. Ищем значения интервалов по формуле: Тренд + 1/2/3\* среднеквадратическое отклонение, как показано на рисунке 1.15

1/2/3 – зависит от выбора сигмы.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.15 – Значения доверительных интервалов

Шаг 12: далее строим по описанному выше алгоритму две точечные диаграммы, первая для отображения доверительных интервалов для отфильтрованных значений, вторая для сравнения линий тренда изначальных и отфильтрованных данных статистики. (см. рисунок 1.16)

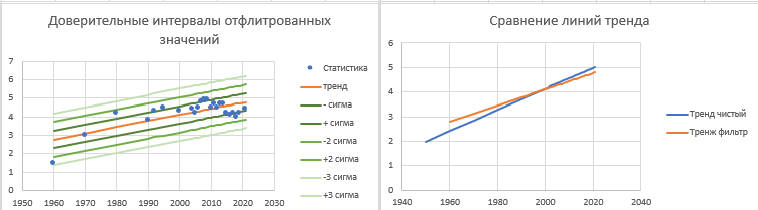


Рисунок 1.16 – Точечные диаграммы доверительных интервалов и сравнения линий тренда

### **1.3.1.3. Описание выходных данных**

После реализации данного алгоритма у нас есть возможность ввести любой год и автоматически будет выведено количество разводов в России на 1 000 человек населения.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.17 – Результаты прогнозирования

## **1.3.2. Алгоритм 2. Решение в Python**

### **1.3.2.1. Описание входных данных**

В качестве входных данных мы имеем:

* Csv файл, содержащий значения переменных x и y. В качестве значений переменной х, в данном случае, мы взяли перечень годов, и в качестве значений переменной y - количество разводов в России на 1 000 человек населения в зависимости от представленных годов.

**1.3.2.2. Описание алгоритма решения**

Алгоритм решения поставленной задачи реализован при помощи двух классов, один из которых (prognoz) отвечает за построение аппроксимации линейной и квадратичной функцией и прогнозирование, а другой (clean\_csv) отвечает за очистку выборки от выбросов. Подробнее данные классы описаны в пункте 1.5.

*Пошаговая реализация алгоритма*

Шаг 1: Пользователь вводит название csv файла, который передается в классы prognoz и clean\_csv

Шаг 2: Класс prognoz формирует переменные со значениями переменных х и у из файла, для дальнейшей их обработки функциями класса

Шаг 3: Функции класса рассчитывают необходимые параметры для построения аппроксимирующей функции: коэффициенты a и b линейной функции, коэффициенты квадратичной функции.

Шаг 4: Рассчитываются параметры для построения доверительных интервалов: стандартное отклонение.

Шаг 5: Построение графиков аппроксимации линейной функцией и аппроксимации квадратичной функцией.

Шаг 6: Построение доверительных интервалов на графике

Шаг 7: Класс clean\_csv вычисляет максимально допустимое значение отклонения тренда на уровне 1,5 sigma

Шаг 8: Точки, выходящие за предел максимально допустимого отклонения, удаляются, очищенная выборка записывается в новый файл csv.

Шаг 9: Строится диаграмма ящика с усами для визуализации выбросов.

Шаг 10: Очищенная выборка снова передается в класс prognoz для построения графиков аппроксимирующей функции уже для очищенной выборки, для возможности детального сравнения фильтрованных и неотфильтрованных данных.

### **1.3.2.3. Описание выходных данных**

Выходными данными алгоритма являются те параметры, которые выберет для вывода пользователь, используя методы реализованных классов. Основными являются:

* Массив точек в формате [fi] (fi – значение аппроксимирующей функции в точках)
* Вид аппроксимирующей функции - в формате строки содержащий каноничную форму аппроксимирующей функции с вычисленными коэффициентами отсортированный в порядке убывания степеней x, для которых определенные для функции коэффициенты отличны от нуля
* Величина дисперсии
* Величина стандартного отклонения
* График аппроксимации линейной функцией
* График аппроксимации квадратичной функцией
* Графики по отфильтрованной выборке
* CSV файл с массивом точек в формате [xi,yi] отфильтрованной выборки

# **1.4. Варианты использования алгоритмов**



## **1.4.1. Варианты использования алгоритма 1 – поиск решений Excel задачи прогнозирования**

Шаг 1: пользователь при необходимости редактирует входные данные и их количество: Год(х), Количество разводов (у). Если необходимы дополнительные строки в таблице, то пользователь должен кликнуть на номер строки на панели слева и выбрать «Вставить». При добавлении новых строк нужно иметь в виду, что также нужно корректировать и формулы, использование которых подробно описано в «3.1.2. Описание алгоритма решения».

Шаг 2: автоматически будет построена точечная диаграмма для фильтрации выпадающих точек и ошибок измерений. Некоторые точки выходят за линии сигмы 1,5 (как точка «2002»), они и есть выпадающие.

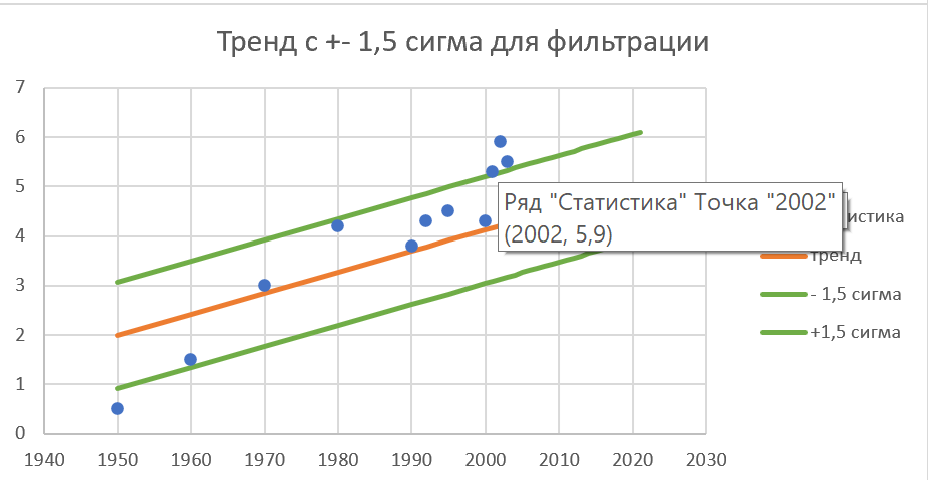


Рисунок 1.18 – Точечная диаграмма для фильтрации

Шаг 3: производим фильтрацию данных. Через точечную диаграмму производим поиск выпадающих точек и удаляем строки с выпадающими значениями, важно отфильтровать всю строку вместе с формулами.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.19 – Выделенные выпадающие точки

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1.20 – Отфильтрованные значения

Шаг 4: будут автоматически построены две точечные диаграммы, первая для отображения доверительных интервалов для отфильтрованных значений, вторая для сравнения линий тренда изначальных и отфильтрованных данных статистики (при изменении размеров выборки, следует проверить наличие ссылок диаграмм на все значения). (см. рисунок 1.21)

Также автоматически будет посчитана дисперсия и среднеквадратическое отклонение.

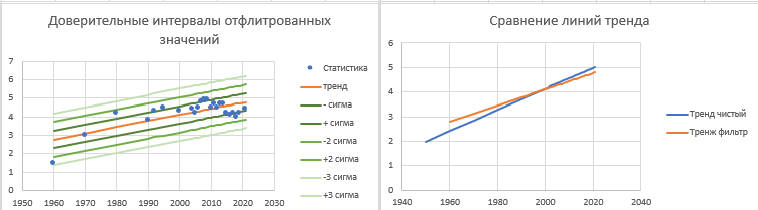


Рисунок 1.21 – Точечные диаграммы доверительных интервалов и сравнения линий тренда

1.4.2. Варианты использования алгоритма 2 – Python для аппроксимации линейной и квадратичной функциями

Пользователь может ввести данные путём загрузки csv файла. Перед выполнением действий для ввода данных необходимо запустить все окна кода с библиотеками для корректной работы программы.

Шаг 1: запустить окно кода «Передача файла в класс». В строке ввода названия csv файла указать его наименование, важно учитывать, что файл csv должен находиться в одной папке с файлом ipynb (кроме папки «Загрузки»)



Рисунок 1.22 – Название файла

Шаг 2: запустить окно кода «Вывод»

В итоге будет выведен результат с массивом точек в формате [fi] (аппроксимированными значениями), формулой линейной и квадратичной аппроксимирующей функции и значением дисперсии. (рис. 1.23)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.23 – Вывод значений

Также автоматически будет выведены графики аппроксимации линейной и квадратичной функций по неотфильтрованной и отфильтрованной выборке (рис. 1.24, 1.25).

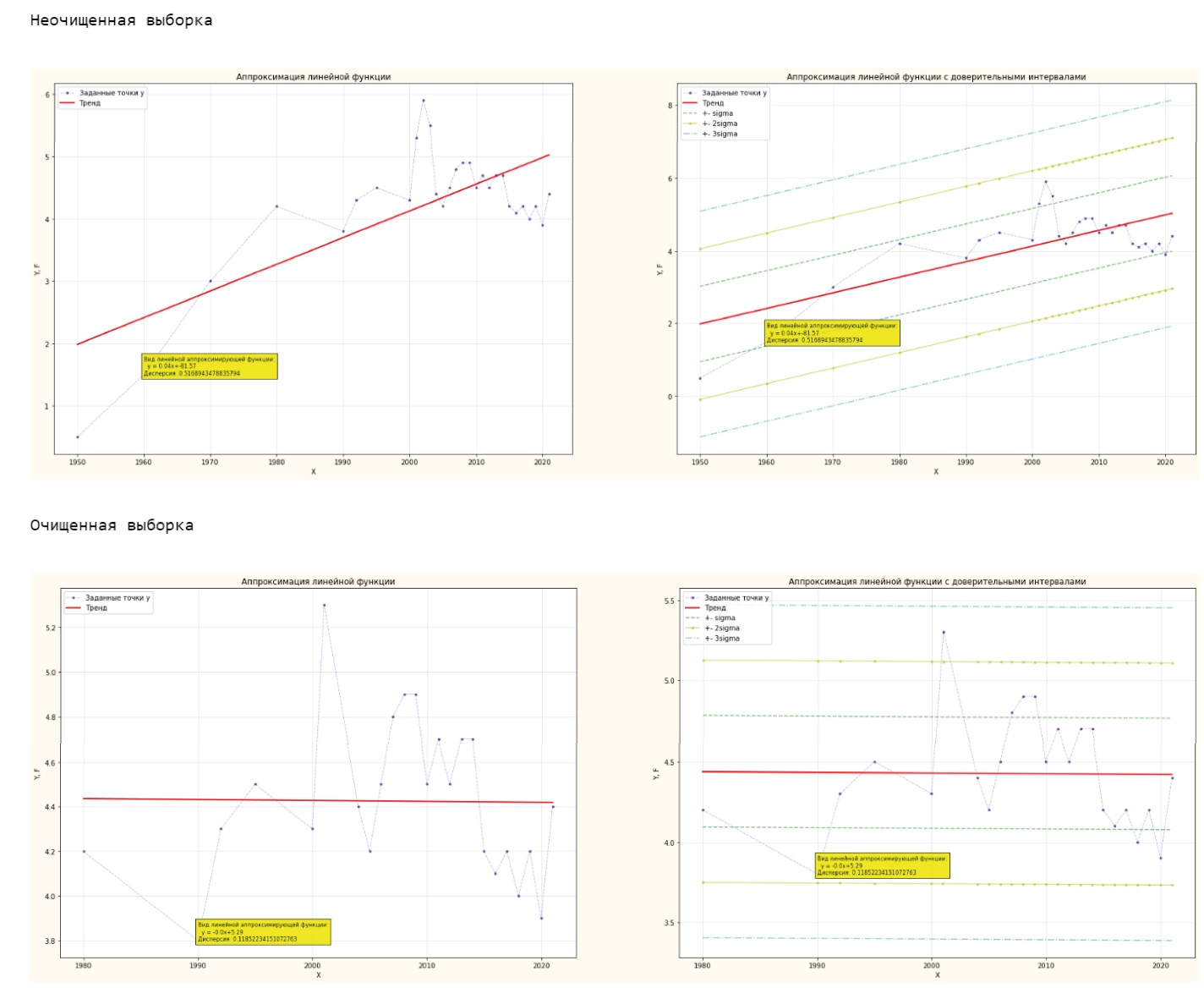


Рисунок 1.24 – графики аппроксимации линейной функции

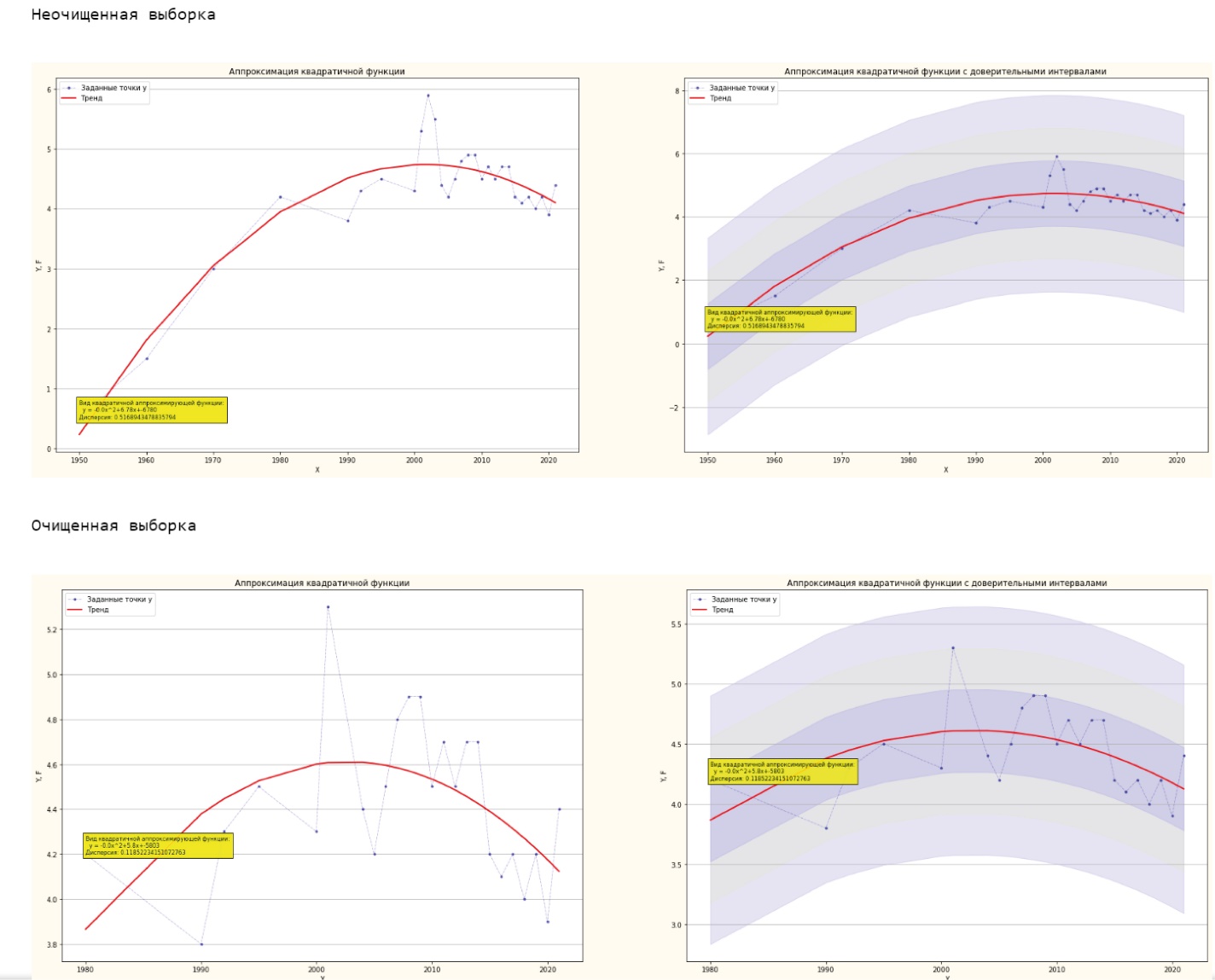


Рисунок 1.25 – графики аппроксимации квадратичной функции

Шаг 3: запустить окно кода «Задание №1.3 Очистка от выбросов». В строке ввода названия csv файла указать его наименование.



Рисунок 1.26 – Название файла

В итоге будет выведены неочищенная и очищенная выборка, разница дисперсий (рис. 1.27, 1.28) и диаграмма размаха (ящик с усами) (рис.1.29).

Изображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описание

Рисунок 1.27 – неочищенная выборка Рисунок 1.28 – очищенная выборка

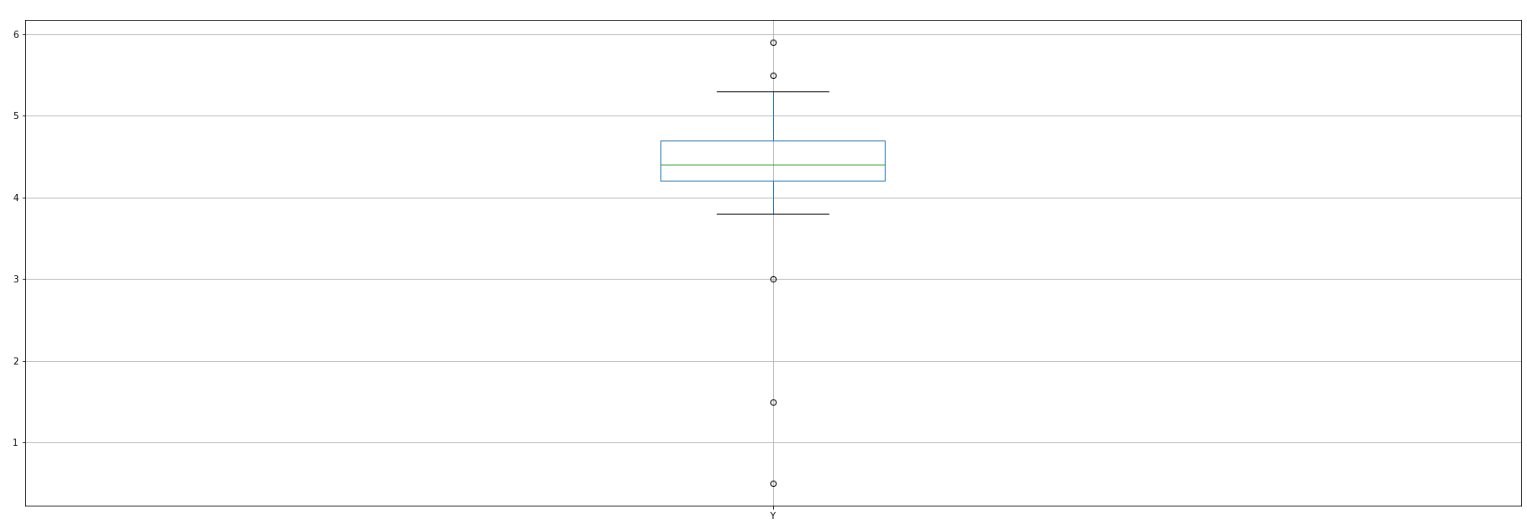


Рисунок 1.29 – ящик с усами

**1.5. Архитектура решения алгоритма на языке программирования Python**

Для реализации всех имеющихся алгоритмов решения мы использовали следующие библиотеки:

* Math — библиотека для использования математических формул;
* NumPy — это библиотека языка Python, добавляющая поддержку больших многомерных массивов (включая матрицы), поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами;
* Scipy — это библиотека Python построенная на базе NumPy и позволяет управлять данными, а также визуализировать их с помощью разных высокоуровневых команд, необходима для функции linprog из пакета scipy.optimize, который содержит в себе множество алгоритмов оптимизации;
* Matplotlib — библиотека для визуализации данных;
* Pandas — библиотека для обработки и анализа структурированных данных;
* [Seaborn](https://seaborn.pydata.org/) — это библиотека для создания статистических графиков на Python. Она основывается на [matplotlib](https://matplotlib.org/) и тесно взаимодействует со [структурами данных pandas](https://livecodestream.dev/post/how-to-work-with-pandas-in-python/);
* Os — модуль, предоставляющий множество функций для работы с операционной системой

Опишем подробнее классы и функции, которые содержатся в коде и реализуют аппроксимацию линейной и квадратичной функциями.

**Prognoz** — класс, созданный для расчёта параметров, а также для визуализации аппроксимации линейной и квадратичной функциями и дальнейшего построения прогноза.

Входные данные класса prognoz:

* Csv-файл, содержащий массивы значения y и x. (Подробнее о формате вводимых данных в csv файл описано в пункте 1.4.2)

Двумя главными атрибутами, внутренними переменными класса являются self.array\_y и self.array\_x, содержащие значения переменной у и переменной х из полученного на входе csv-файла. Значения считываются из файла благодаря методу библиотеки pandas – pd.read\_csv.

Для удобства вызова конкретного параметра была произведена декомпозиция задачи прогнозирования на несколько функций, решающих конкретные подзадачи в классе prognoz.

В основном классе есть функции подсчета:

**sum\_x()** – сумма предикторов х. Принимает на вход внутреннюю переменную self.array\_x, возвращает сумму по данному массиву переменных.

**sum\_y()** – сумма значений y. Принимает на вход внутреннюю переменную self.array\_y, возвращает сумму по данному массиву переменных.

**square\_x()** - сумма квадратов предикторов x. Принимает на вход внутреннюю переменную self.array\_x, возвращает сумму квадратов по данному массиву переменных.

**sum\_xy()** - сумма произведений предикторов x и y. Принимает на вход внутренние переменную self.array\_x и self.array\_y, возвращает сумму произведений по данному массиву переменных.

**var\_a()** - значение a в линейной функции. Принимает на вход значения функций sum\_xy, sum\_x, sum\_y, square\_x, возвращает значение а.

**var\_b()** - значение b в линейной функции. Принимает на вход значения функций sum\_x, sum\_y, var\_a, возвращает значение b.

**approx\_func()** - поиск отклонений, где функция стремится к минимуму для линейной функции

**array\_f()** - аппроксимированные значения для линейной функции. Функция возвращает список аппроксимированных значений.

**function\_str()[0]** – функция возвращает строковое выражение линейной аппроксимирующей функции

**sum\_3\_degree** - сумма предикторов x в 3 степени. Принимает на вход внутреннюю переменную self.array\_x, возвращает сумму по данному массиву переменных, возведенному в третью степень.

**sum\_4\_degree** - сумма предикторов x в 4 степени. Принимает на вход внутреннюю переменную self.array\_x, возвращает сумму по данному массиву переменных, возведенному в четвертую степень.

**sum\_x2\_y** – сумма произведений предикторов x^2 и y. Принимает на вход внутреннюю переменную self.array\_x и self.array\_y, возвращает сумму по данному массиву переменных.

**var\_abc()** - коэффициенты квадратичной функции (a,b,c соответственно)

**array\_square\_f()** - аппроксимированные значения для квадратичной функции

**main\_matrix** - система уравнений

**sigma** - стандартное отклонение массива f (аппроксимирующих значений х). Принимает на вход внутреннюю переменную self.array\_y, возвращает значение стандартного отклонения, рассчитанного как корень из формулы 2 математической модели.

**dispersion** - дисперсия массива f, (аппроксимирующих значений х). Принимает на вход внутреннюю переменную self.array\_y, возвращает дисперсию, рассчитанную из формулы 3 математической модели.

**function\_str()[1]** - строковое выражение аппроксимирующей функции квадратичной

Функции визуализации класса prognoz:

**prog.draw\_board()** - визуализация линейной функции при помощи библиотеки matplotlib

**prog.draw\_board\_f2()** - визуализация квадратичной функции при помощи библиотеки matplotlib

**Clean\_csv** — класс, созданный для фильтрации (удаления выбросов) исходного массива, а также для визуализации аппроксимации линейной и квадратичной функциями и дальнейшего построения прогноза.

# **1.6. Тестирование**

Для тестирования построения аппроксимирующей линейной функции были использованы 3 датасета, тесты производились следующими алгоритмами: Excel и Python. Хотелось бы заметить, что датасеты 1 и 3 не имеют выбросов, поэтому в Таблице 1.1 для данных датасетов результаты представлены для неочищенных выборок.

Таблица 1.1 - Тестирование алгоритмов Excel и Python

|  |  |
| --- | --- |
| Датасет 1 | |
| Неочищенная выборка | |
| Алгоритм MS Excel | Алгоритм Python |
| Дисперсия: 0 | Дисперсия: 0 |
| Коэффициент a: 10,  Коэффициент b: 10 | Функция: y = 10x+10 |
| График: | График: |
| Датасет 2 | |
| Неочищенная выборка | |
| Алгоритм MS Excel | Алгоритм Python |
| Дисперсия: 5.291319 | Дисперсия: 5.291319 |
| Коэффициент a: -0.00664,  Коэффициент b: 50.30494 | Функция: y = -0.1x+50.3 |
| График: | График:  Изображение выглядит как текст, внутренний  Автоматически созданное описание |
| Очищенная выборка | |
| Алгоритм MS Excel | Алгоритм Python |
| Дисперсия: 2.513045 | Дисперсия: 4.89802 |
| Коэффициент a: -0.01892,  Коэффициент b: 51.14235 | Функция: y = -0.1x+50.33 |
| График: | График:  Изображение выглядит как текст, внутренний  Автоматически созданное описание |
| Датасет 3 | |
| Неочищенная выборка | |
| Алгоритм MS Excel | Алгоритм Python |
| Дисперсия: 8.61125 | Дисперсия: 8.61125 |
| Коэффициент a:1.92312E-17,  Коэффициент b: -7.1404E-17 | Функция: y = 0.0x+(-0.0) |
| График: Изображение выглядит как стол  Автоматически созданное описание | График: |

Проанализировав таблицу выше, можно сделать следующие выводы: в неочищенных выборках значения дисперсии идентичны, однако в очищенной выборке датасета 2 значения дисперсии серьезно различаются; есть различия в доверительных интервалах в датасете 1, алгоритм Python построил доверительные интервалы отличные от тренда, хотя дисперсия равна нулю.; хотелось бы отметить, что алгоритм Excel выводит более точные значения, например в коэффициентах в датасете 3 (a = 1.92312E-17, b = -7.1404E-17 в алгоритме Excel, а = 0.0, b = 0.0 в алгоритме Python). Также стоит заметить, что алгоритм Excel выводит коэффициенты линейной функции, а алгоритм Python непосредственно функцию.

Далее было проведено тестирование алгоритмов Python: аппроксимация линейной функции по неочищенной и очищенной выборкам, аппроксимация квадратичной функции по неочищенной и очищенной выборкам. Результаты тестирования представлены в таблице 1.2:

Таблица 1.2 - Тестирование алгоритмов Python

|  |  |
| --- | --- |
| Датасет 1 | |
| Неочищенная выборка | |
| Линейная функция | Квадратичная функция |
| Дисперсия: 0 | Дисперсия: 0 |
| Функция: y = 10x+10 | Функция: y = 0.0x^2+10x+10 |
| График: | График: |
| Датасет 2 | |
| Неочищенная выборка | |
| Линейная функция | Квадратичная функция |
| Дисперсия: 5.291319 | Дисперсия: 5.291319 |
| Функция: y = -0.1x+50.3 | Функция: y = 0.0x^2-0.2x+50 |
| График:  Изображение выглядит как текст, внутренний  Автоматически созданное описание | График: |
| Очищенная выборка | |
| Линейная функция | Квадратичная функция |
| Дисперсия: 4.89802 | Дисперсия: 4.89802 |
| Функция: y = -0.1x+50.33 | Функция: y = 0.0x^2-0.3x+50 |
| График:  Изображение выглядит как текст, внутренний  Автоматически созданное описание | График: |
| Датасет 3 | |
| Неочищенная выборка | |
| Линейная функция | Квадратичная функция |
| Дисперсия: 8.61125 | Дисперсия: 8.61125 |
| Функция: y = 0.0x+(-0.0) | Функция: y = 0.0x^2-0.0x+0 |
| График: | График:  Изображение выглядит как текст, снимок экрана, документ  Автоматически созданное описание |

**Так как есть возможность построения прогнозов для обеих видов аппроксимаций, можно сравнить линии тренда прогнозов. Пример показан на рисунках 1.30 и 1.31 на основном датасете физической модели.**

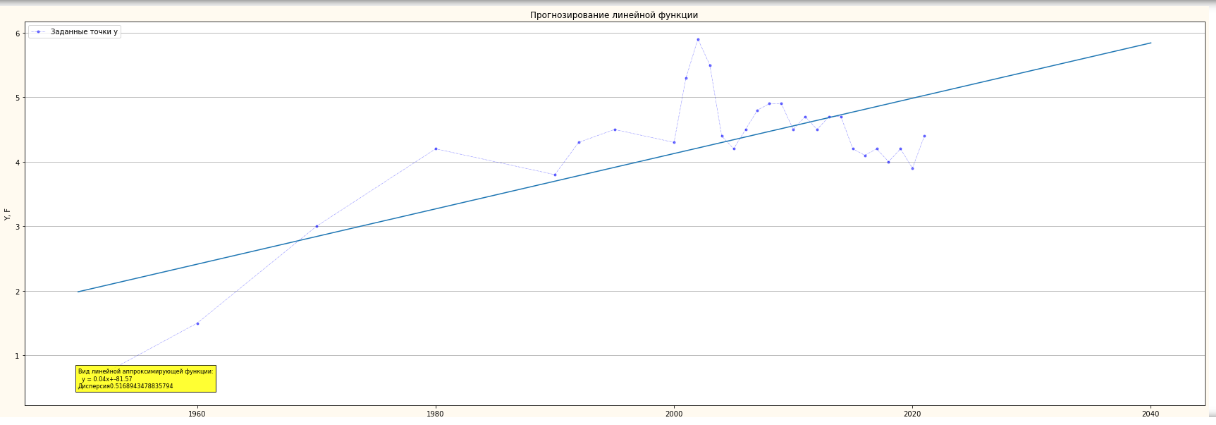
****

Рисунок 1.30 Линейный тренд

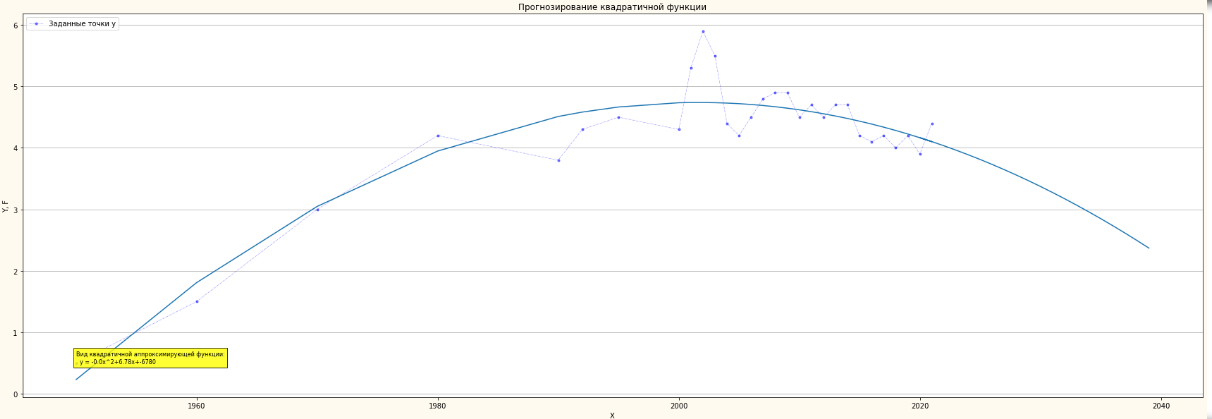


Рисунок 1.31 Квадратичный тренд

Сравнивать нужно значения суммы квадратичных отклонений. (рис 1.32)



**Рисунок 1.32 Значение суммы квадратичных отклонений**

**Значение суммы квадратичных отклонений аппроксимации квадратичной функцией заметно ниже, что говорит о ее более высокой точности по сравнению с аппроксимацией линейной функцией.**

# **1.7. Заключение**

Решение для предоставленной задачи мы реализовали с помощью двух алгоритмов. Первый алгоритм основан на встроенных функциях Excel. Его преимуществами можно считать – простоту использования, редактирования и ввода новых данных, автоматическое обновление формул. Из недостатков – отсутствие пользовательского интерфейса, пользователь самостоятельно должен строить графики и интервалы.

Второй алгоритм, реализованный в Python имеет очевидное преимущество перед первым – строится аппроксимация не только линейной функцией, но и квадратичной функцией. В данном алгоритме удобством является возможность вывести любой требуемый параметр методами реализованных классов, а также построение различных типов графиков, в том числе прогностических. К сожалению, на данном этапе разработки все еще остаются логические ошибки, а также ошибки, связанные с построением графиков. Программа имеет перспективы для расширения функционала, и более детальной доработки, так как можно прописать удобный пользовательский интерфейс, объединить графики аппроксимации с графиками.

Алгоритмы были проверены на трех сформированных датасетах. Выводиться будет только один из них, то есть возможности просмотра альтернативного наименьшего пути нет. Дисперсии, коэффициенты функций рассчитываются одинаково как в Excel, так и в Python. Доверительные интервалы для выборок с выбросами также одинаковы, исключением являются выборки, который не имеют выбросов, так как в этом случае алгоритм Питона некорректно строит доверительные интервалы для них при их отсутствии. Более высокую точность аппроксимации имеет аппроксимация квадратичной функцией, так как величина ее квадратического отклонения меньше.

# **2. Многокритериальная оптимизация**

# **2.1. Постановка задачи (физическая модель)**

Наш заказчик – учебная группа БИ20-8 Финансового университете при Правительстве РФ. Данная учебная группа крайне заинтересована в получении качественного и практико-ориентированного образования.

В текущем семестре преподавателем предмета, «Математические методы принятия решений», является Аксёнов Дмитрий Андреевич. Данный преподаватель также вел у заказчика во втором семестре математику.

Стиль преподнесения информации, установка требований и приведение практических примеров крайне заинтересовало студентов группы БИ20-8, поэтому они обратились в наше агентство, чтобы определить будет ли Аксёнов Дмитрий Андреевич вести другую дисциплину на 3 или 4 курсе. Также нашим заказчиком была предоставлена официальная информация о перечне предметов на 3 и 4 курсе и характеристики данных дисциплин.

Итак, в качестве объектов были перечислены следующие предметы: «Информационные системы управления организацией», «Экономика информационных систем», «Архитектура организации», «Управление информационно-технологическими проектами», «Информационные технологии бизнес аналитики», «Основы управления ИТ-сервисами», «Основы управления информационными технологиями», «Информационная безопасность», «Цифровой бизнес», «Мобильные технологии» и «Основы технологий интернета вещей».

Также заказчик указал следующие критерии: часы семинарских и лекционных занятий, часы самостоятельной работы, полезность дисциплины, сложность, актуальность, количество теоретических проверочных работ и практико-ориентированных.

Все представленные данных от заказчика для проведения дальнейшей оценки приведены на рисунке 2.1.

Изображение выглядит как стол

Автоматически созданное описаниеРисунок 2.1– Представленные данные

# **2.2. Математическая модель**

Рассмотрим математическую модель оптимизации по Парето.

(2.1)

При выполнении ограничений:

*,*

Рассмотрим линейную свёртку двух критериев.

– первый критерий

– второй критерий

Формула линейной свертки для двух данных критериев:

(2.2)

Где a1 и a2 – весовые коэффициенты, отражающие значимость критериев.

В формуле 2.2 знак перед слагаемым зависит от того, оптимизируется критерий к минимуму или же к максимуму. Так как второй критерий стремится к минимуму, его значение умножается на -1. То есть если бы оба критерия стремились к максимуму, формула бы имела следующий вид:

Рассмотрим линейную свёртку для множественных критериев.

(2.3)

При условии, что все критерии стремятся к максимуму

,

В данной работе по физической модели будет использоваться формула линейной свёртки для разноимённых критериев, так как второй критерий стремится к минимуму, формула принимает следующий вид:

(2.4)

То есть критерии, стремящиеся к минимуму, умножаются на -1.

Рассмотрим метод идеальной точки.

(2.5)

Рассмотрим метод контрольных показателей.

(2.6)

где .

# **2.3. Алгоритм решения**

## **2.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel задачи оптимизации**

### **2.3.1.1. Описание входных данных**

В качестве входных данных мы имеем матрицу с множеством критериев, где выделены два основных критерия, их характеристики со значениями, а также список, в котором указано, к чему стремится (к минимуму или максимуму) каждый критерий.

### **2.3.1.2. Описание алгоритма решения**

Алгоритм принимает на вход значения матрицы. Далее, используя формулы Excel, строим оптимизацию по Парето, линейную свёртку двух основных критериев, линейную свёртку по всем критериям, отображаем лепестковые диаграммы, ищем сбалансированное решение методом идеальной точки и методом контрольных показателей.

*Подробный пример реализации алгоритма*

Шаг 1: в соответствии с описанием задачи составляем таблицу данных от заказчика (см рис 2.1).

Шаг 2: выделяем два наиболее значимых критерия и строим точечную диаграмму «Паретооптимальные решения». Находим неулучшаемые значения: точки имеющие минимальные значения относительно сложности и максимальные относительно полезности, при этом в области «улучшения» значений точек нет (область выделена оранжевыми линиями).

В данном случае неулучшаемые значения лежат во второй четверти, поскольку значения по оси абсцисс стремятся к максимуму, а по оси ординат – к минимуму.

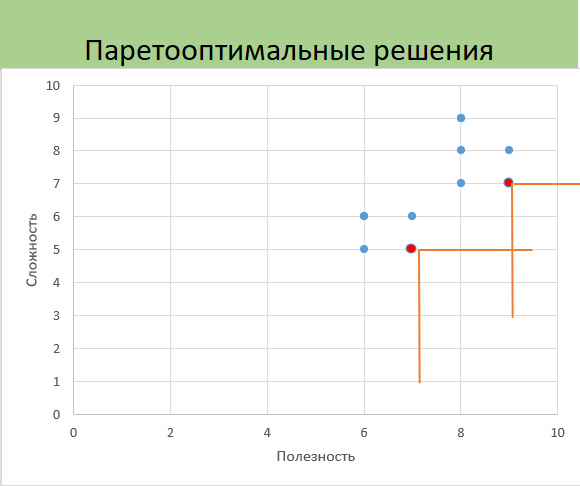


Рисунок 2.2 – Оптимизация по Парето

Шаг 3: строим пустую таблицу данных для линейной свертки основных критериев. Указываем весовые коэффициенты и , где

Изображение выглядит как текст, седзи

Автоматически созданное описание

Рисунок 2.3 – таблица данных линейной свертки

Шаг 4: заполняем таблицу данных для линейной свертки основных критериев значениями, рассчитанными по формуле (2.2) (см рис 2.3).

Где – критерий «полезность дисциплины»

– критерий «сложность дисциплины»

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.4 – линейная свёртка

Шаг 5: ищем максимум по столбцам (=МАКС()) и выделяем соответствующие значения в таблице (см рис 2.4).



Рисунок 2.5 – поиск максимума

Шаг 6: составляем таблицу «Линейная свертка всех критериев», соответствующую таблице с данными от заказчика (см рис 2.1). Вручную определяем значимость каждого критерия по десятибалльной шкале (в столбце, значения которого стремятся к минимуму, присваиваем минус), суммируем все значимости (см рис 2.6) и высчитываем нормализированные деля каждую значимость на сумму значимостей (см рис 2.7).



Рисунок 2.6 – значимость в линейной свёртке



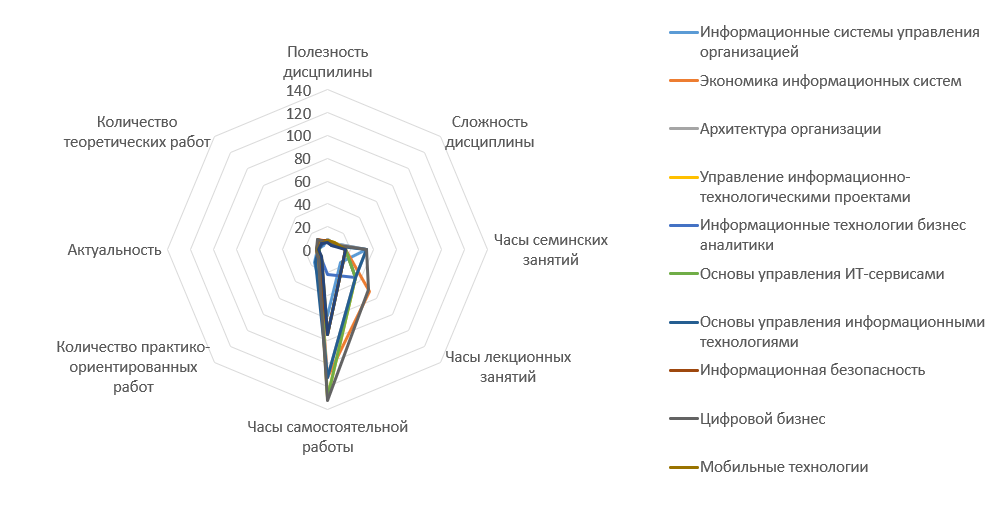
Рисунок 2.7 – нормализированные

Шаг 7: производим свёртку в соответствии с формулой (2.4), находим максимум по столбцу и выделяем соответствующие значения в таблице (см рис 2.8).



Рисунок 2.8 – свёртка

Шаг 8: строим лепестковую диаграмму (см рис 2.9).

Рисунок 2.9 – лепестковая диаграмма

Шаг 9: составляем таблицу «метод идеальной точки», соответствующую таблице с данными от заказчика (см рис 2.1), и высчитываем идеальную точку по каждому столбцу (см рис 2.10), с помощью нахождения максимума и добавления к нему 10%. Для критерия, значения которого стремятся к минимуму, присваиваем значение, которое меньше минимума.

Рисунок 2.10 – нахождение идеальной точки

Шаг 10: строим пустую таблицу данных и заполняем значениями, высчитанными в соответствии с формулой (2.5) без учёта возведения в квадрат.

Где – нормализированная идеальная точка по столбцам

– значение критерия дисциплины

Для критериев, которые стремятся к минимуму, используем обратную дробь.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.11 – формула идеальной точки

Шаг 11: по формуле (2.5) суммируем значения по строкам и возводим в квадрат (см рис 2.12).

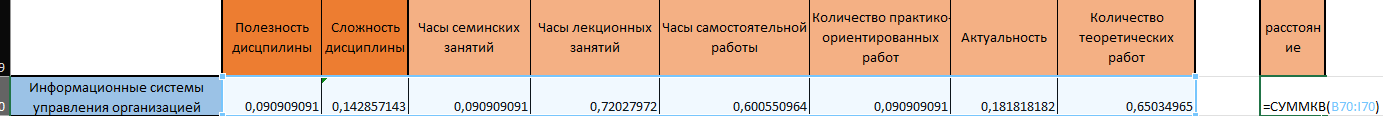


Рисунок 2.12 – поиск расстояния

Находим минимум среди этих значений (=МИН()) и выделяем соответствующие значения в таблице (см рис 2.13).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.13 – минимальная идеальная точка

Шаг 12: строим лепестковую диаграмму (см рис 2.14).

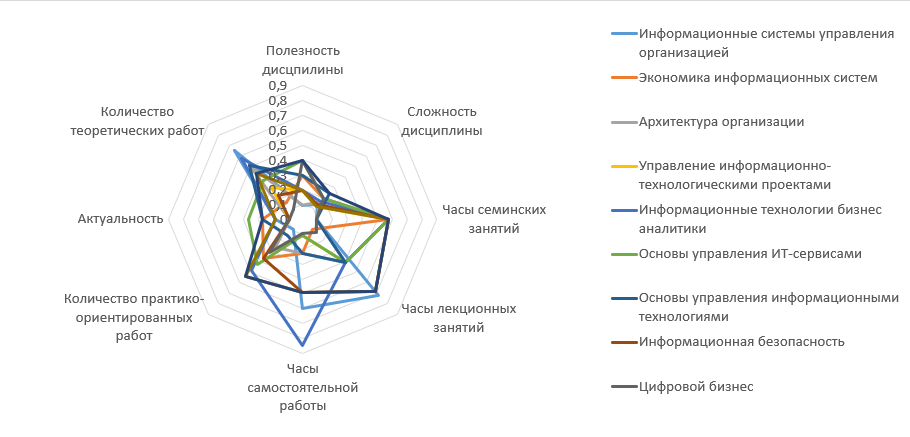


Рисунок 2.14 – лепестковая диаграмма

Шаг 13: составляем таблицу «метод контрольных показателей», соответствующую таблице с данными от заказчика (см рис 2.1), и вручную определяем нижнюю границу каждого критерия (контрольный показатель).

Шаг 14: заполняем пустую таблицу значениями, высчитанными в соответствии с формулой (2.6) (см рис 2.15)

Где – контрольный показатель,

– значение критерия дисциплины.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.15 – метод контрольных показателей

Шаг 15: на вкладке «Главное» в стиле «Условное форматирование» выбираем пункт «Создать правило» и выделяем красным ячейки, содержащие значения меньше единицы (см рис 2.16). Зачёркиваем строки, которые содержат красные ячейки.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.16 – создание правила

Затем находим минимум среди этих значений (=МИН()) и выделяем соответствующие значения в таблице (см рис 2.17).

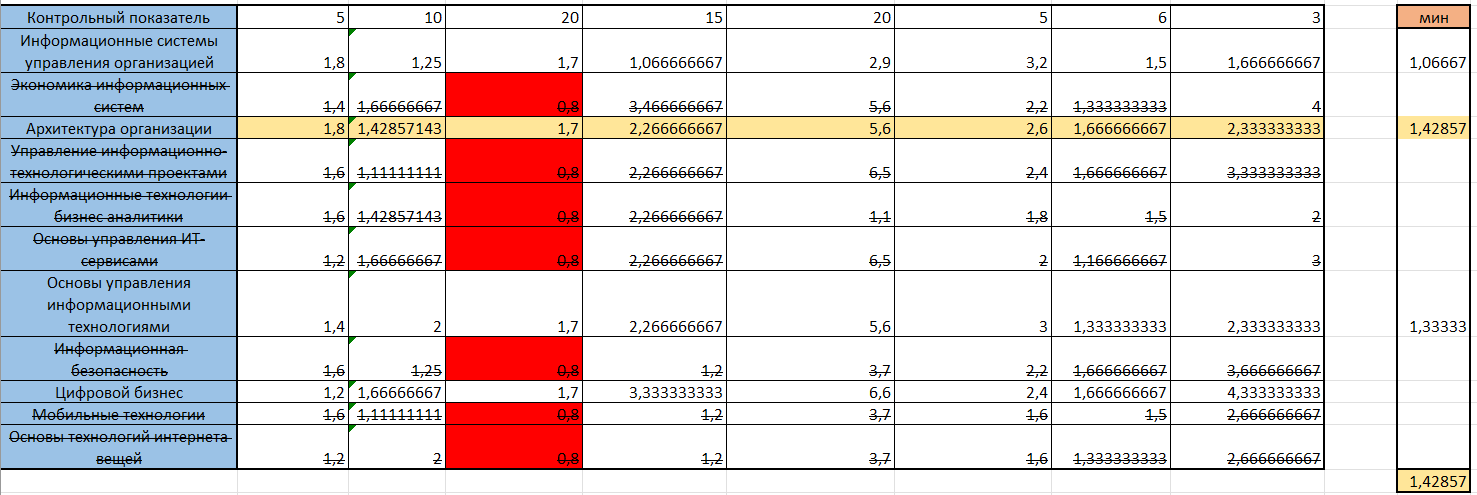


Рисунок 2.17 – таблица «метод контрольных показателей»

Шаг 16: строим лепестковую диаграмму (см рис 2.18).



Рисунок 2.18 – лепестковая диаграмма

### **2.3.1.3. Описание выходных данных**

Выходными данными алгоритма являются:

* Набор пареттооптимальных точек
* Линейная свертка по двум критериям
* Свертка по всем критериям, оптимальная дисциплина по результату свертки
* Лепестковая диаграмма по свертке всех критериев
* Лепестковая диаграмма по методу идеальной точки
* Расстояние каждой дисциплины от идеальной точки
* Оптимальная дисциплина по методу идеальной точки
* Таблица с дисциплинами, прошедшими контрольные показатели
* Лепестковая диаграмма по методу контрольных показателей

## **2.3.2. Алгоритм 2. Решение в Python**

### **2.3.2.1. Описание входных данных**

В качестве входных данных мы имеем:

* Csv файл, содержащий матрицу значений по дисциплинам и соответствующим критериям. Значения, в данном случае, были определены в зависимости от представленных критериев оптимизации.

**2.3.2.2. Описание алгоритма решения**

Алгоритм решения поставленной задачи реализован при помощи двух классов, один из которых (opt) отвечает за расчёты оптимизации методами линейной свертки, идеальной точки и метода контрольных показателей, а другой (RadarAxes) содержится в функции (radar\_factory) и отвечает за построение лепестковых диаграмм. Подробнее данные классы описаны в пункте 5.

*Пошаговая реализация алгоритма*

Шаг 1: Пользователь вводит название csv файла, который передается в класс opt

Шаг 2: Класс opt формирует переменные со значениями переменных х и у из файла, для дальнейшей их обработки функциями класса

Шаг 3: Функция класса for\_two\_criteria рассчитывает линейную свертку по двум критериям определенным пользователем.

Шаг 4: Пользователь вводит с клавиатуры критерии значимости для каждой дисциплины.

Шаг 5: Функция for\_all класса opt рассчитывает линейную свертку по всем критериям и подсвечивает зеленым цветом оптимальную дисциплину.

Шаг 6: Строится лепестковая диаграмма по свертке всех критериев при помощи класса RadarAxes и функции radar\_factory.

Шаг 7: Функцией ideal\_dot рассчитывается идеальная точка для каждого столбца- критерия исходной таблицы как прибавление 10% к максимуму, если критерий стремится к максимуму, и вычитание 10% от минимума, если критерий стремится к минимуму.

Шаг 8: Определяется расстояние каждой дисциплины от идеальной точки и на основании этого находится оптимальная дисциплина.

Шаг 9: Строится лепестковая диаграмма по методу оптимизации идеальной точкой.

Шаг 10: Функция benchmark строит новую матрицу на основании исходной и введенных пользователем контрольных показателей, выводятся только те дисциплины, которые прошли контрольные показатели.

Шаг 11: Строится лепестковая диаграмма по методу лепестковой диаграммы.

### **2.3.2.3. Описание выходных данных**

Выходными данными алгоритма являются:

* Линейная свертка по двум выбранным пользователем критериям
* Свертка по всем критериям, оптимальная дисциплина по результату свертки
* Лепестковая диаграмма по свертке всех критериев
* Лепестковая диаграмма по методу идеальной точки
* Расстояние каждой дисциплины от идеальной точки
* Таблица с дисциплинами, прошедшими контрольные показатели, дисциплины, которые не прошли обозначены значениями «NA»
* Лепестковая диаграмма по методу контрольных показателей

# **2.4. Варианты использования алгоритмов**

## **2.4.1. Варианты использования алгоритма 1 – поиск решений Excel**

Шаг 1: пользователь при необходимости редактирует входные данные, их количество, стремление к максимуму/минимуму и обозначает два самых значимых критерия (рис 2.1). При изменении входных данных нужно иметь в виду, что также нужно корректировать и формулы, использование которых подробно описано в «2.3.1.2. Описание алгоритма решения».

Шаг 2: автоматически будет построена точечная диаграмма «Паретооптимальные решения». Находим неулучшаемые значения: точки имеющие минимальные значения относительно сложности и максимальные относительно полезности, при этом в области «улучшения» значений точек нет (область выделяем оранжевыми линиями прим рис 2.19).

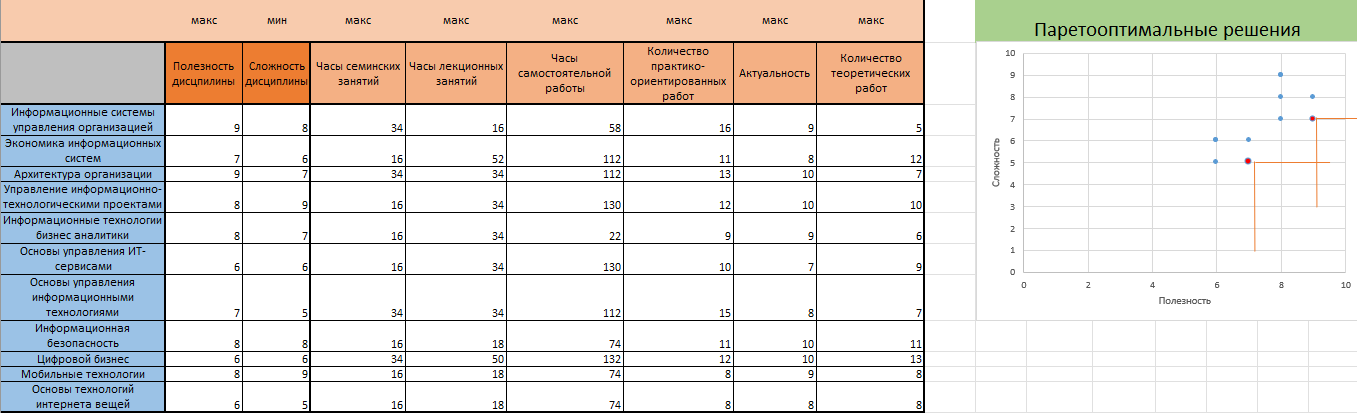


Рисунок 2.19 – входные данные и точечная диаграмма Парето

Шаг 3: автоматически будет высчитана «Линейная свертка для двух основных критериев». Выделяем цветом строки содержащие значения, соответствующие значениям в последней строке «макс», определяя таким образом более подходящие по требованиям предметы (см рис 2.20).



Рисунок 2.20 – Линейная свертка для двух основных критериев

Шаг 4: таблица «Линейная свертка всех критериев» заполняется значениями входных данных (см рис 2.1). Пользователь обозначает важность каждого критерия путём заполнения строки «значимость» числами по десятибалльной шкале. Важно учитывать, что если критерий стремится к минимуму (а не к максимуму), то перед значением следует поставит знак минус (прим «Сложность дисциплины» рис 2.21).



Рисунок 2.21 - Значимость

Шаг 5: автоматически будет высчитан столбец «свёртка». Последнее значение столбца отражает максимальное значение, выделяем цветом строку, содержащую максимальное значение, определяя таким образом более подходящий по требованиям предмет. Лепестковая диаграмма будет высчитана автоматически (см рис 2.22).

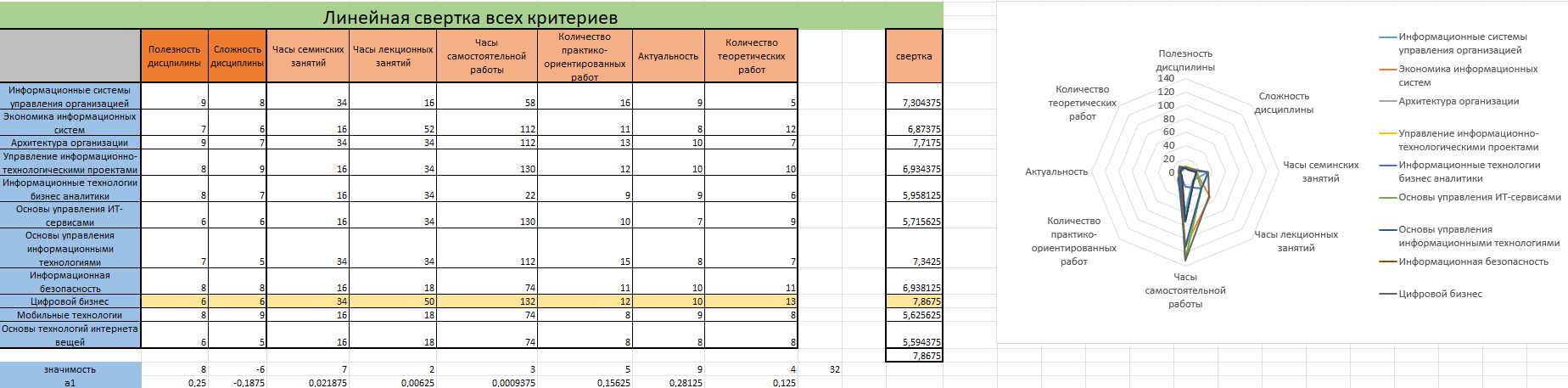


Рисунок 2.22 – линейная свёртка всех критериев

Шаг 6: таблица «Метод идеальной точки» заполняется значениями входных данных (см рис 2.1). Автоматически заполняется строка «идеальная точка», таблица ниже и столбец «Расстояние» (см рис 2.23). Последнее значение столбца отражает минимальное значение, выделяем цветом строку, содержащую минимальное значение, определяя таким образом более подходящий по требованиям предмет. Лепестковая диаграмма будет высчитана автоматически (см рис 2.23).

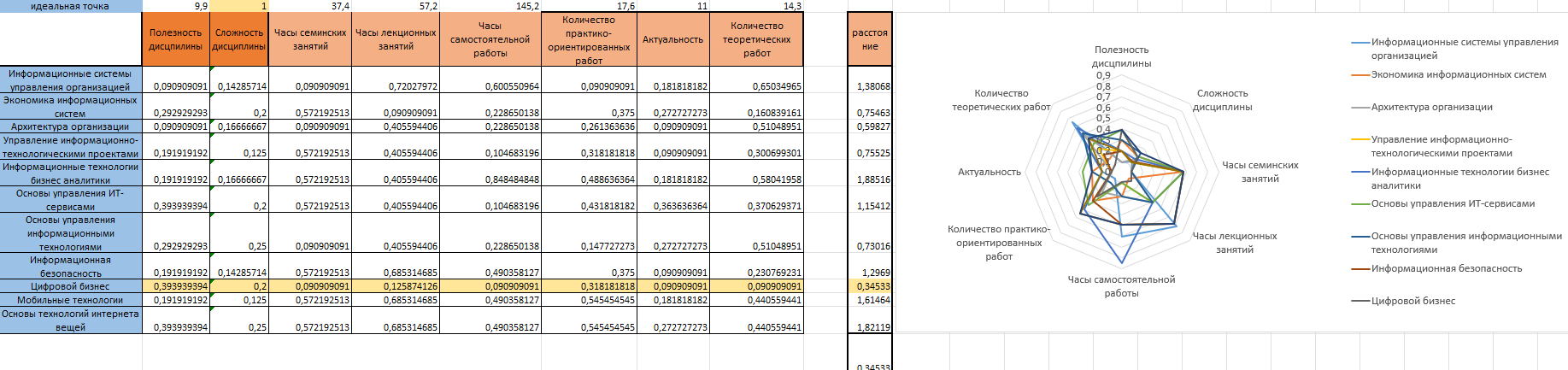


Рисунок 2.23 – метод идеальной точки

Шаг 7: таблица «Метод контрольных показателей» заполняется значениями входных данных (см рис 2.1). Пользователь обозначает нижнюю границу для каждого критерия путём заполнения строки «контрольный показатель» (см рис 2.24).



Рисунок 2.24 – контрольный показатель

Шаг 8: автоматически заполняется таблица ниже (выделяются красным цветом значения меньше единицы) и столбец «Мин» (см рис 2.25). Последнее значение столбца отражает максимальное значение, выделяем цветом строку, содержащую максимальное значение, определяя таким образом более подходящий по требованиям предмет. Зачёркиваем строки, которые содержат красные ячейки. Лепестковая диаграмма будет высчитана автоматически (см рис 2.25).

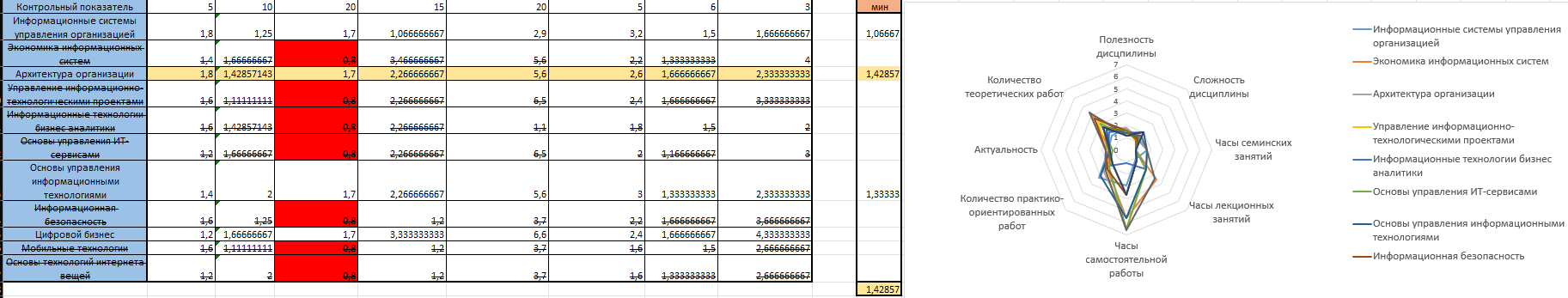


Рисунок 2.25 – Метод контрольных показателей

2.4.2. Варианты использования алгоритма 2 – Python

Пользователь может ввести данные путём загрузки csv файла. Перед выполнением действий для ввода данных необходимо запустить все окна кода с библиотеками для корректной работы программы. Для user-friendly интерфейса необходимо загрузить файл opt.jpg в одну папку с файлом кода ipynb и csv файлом (любую кроме папки «загрузки»).

Шаг 1: запустить два окна с загрузкой библиотек для дальнейшей работы (см рис 2.26)

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.26 – Загрузка библиотек

Шаг 2: запустить третье окно ввода, затем запустить четвёртое, при отсутствии ошибок.

Шаг 3: закрыть всплывающее приветствующее окно.

Шаг 4: после закрытия окна произойдёт переход к консольному вводу данных.

Далее необходимо ввести название csv-файла с данными в формате «name.csv» (см рис 2.27).



Рисунок 2.27 – Ввод имени csv-файла

Необходимо учитывать расположение данных в загружаемом файле, он должен соответствовать шаблону (см рис 2.28).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.28 – Формат csv-файла

Шаг 5: в окне ввода направлений ввести через запятую 0 или 1 в зависимости от того, к чему стремится критерий (количество направлений должно совпадать с количеством критериев). Ввести 0 если критерий стремится к максимуму и 1 если он стремится к минимуму (см рис 2.29).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.29 – Окно ввода направлений через запятую

Шаг 6: далее нужно проверить соответствие номеров и названий критериев. В строке выбора программы предстоит выбрать метод оптимизации (см рис 2.30).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.30 – Проверка критериев

Шаг 7.1.1. если пользователь желает запустить программу линейной свёртки для двух основных критериев, следует ввести цифру 1 в строке выбора программы (см рис 2.32).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.31 – Выбор программы 1

Шаг 7.1.2 в строку ввода через запятую ввести номера двух основных критериев, для которых будет проводиться свёртка (см рис 2.32).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.32 – Ввод основных критериев

Шаг 7.1.3 датафрейм после свёртки данных автоматически будет выведен на экран. Пользователь самостоятельно определяет строки содержащие значения, соответствующие значениям в последней строке «max», определяя таким образом более подходящие по требованиям предметы.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.33 – Датафрейм после свёртки данных

Шаг 7.2.1: если пользователь желает запустить программу линейной свёртки для всех критериев, следует ввести цифру 2 в строке выбора программы (см рис 2.34).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.34 – Выбор программы 2

Шаг 7.2.2: в строку ввода через запятую ввести номера всех критериев, для которых будет проводиться свёртка (количество значимостей критериев в строке должно совпадать с количеством критериев в csv-файле) (см рис 2.35).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.35 – Ввод критериев

Шаг 7.2.3: на экран будет автоматически выведена свёртка данных с выделенным значением оптимальной дисциплины (см рис 2.36) и лепестковая диаграмма с наглядным изображением критериев по каждому предмету (см рис 2.37).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.36 – Свёртка с оптимальной дисциплиной

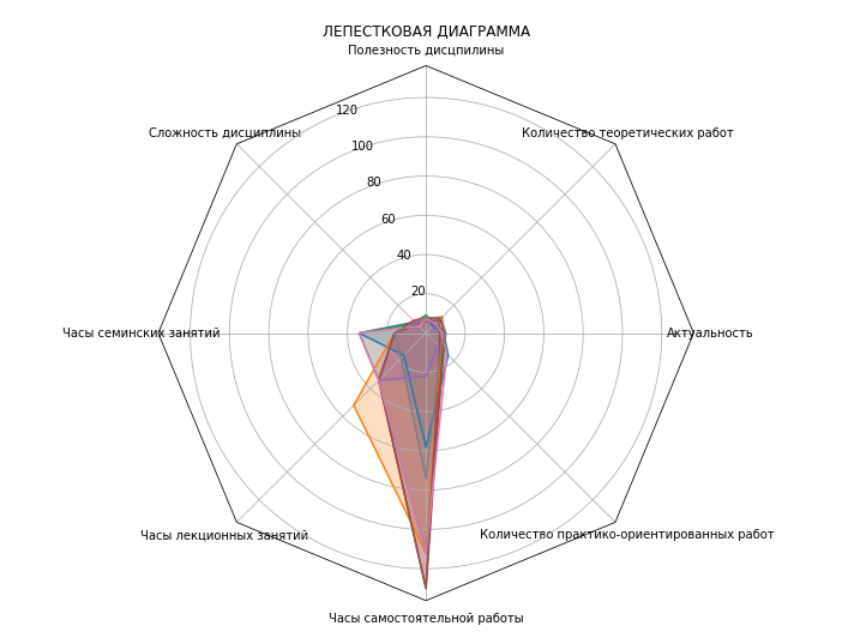


Рисунок 2.37 – Лепестковая диаграмма

Шаг 7.3.1: если пользователь желает запустить программу Метод идеальной точки, следует ввести цифру 3 в строке выбора программы (см рис 2.38).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.38 – Выбор программы 3

Шаг 7.3.2: на экран будет автоматически выведена лепестковая диаграмма с наглядным изображением критериев по каждому предмету (см рис 2.39) и расстояние каждого предмета (см рис 2.40).

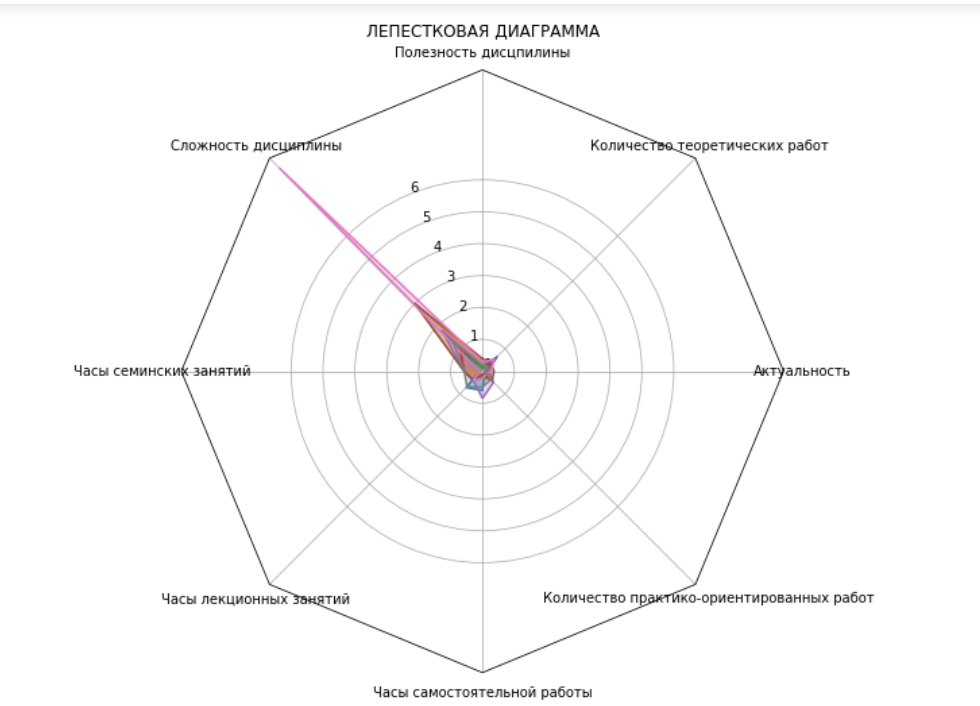


Рисунок 2.39– Лепестковая диаграмма

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.40 - Расстояние каждого предмета

Пользователь самостоятельно определяет строку, содержащую минимальное значение, определяя таким образом более подходящий по требованиям предмет.

Шаг 7.4.1: если пользователь желает запустить программу метод контрольных показателей, следует ввести цифру 4 в строке выбора программы (см рис 2.34).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.41 – Выбор программы 4

Шаг 7.4.2: в строку ввода по очереди ввести контрольные показатели всех критериев (количество показателей автоматически будет совпадать с количеством критериев в csv-файле) (см рис 2.43).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.42 – Ввод контрольного показателя для 8 критерия

Шаг 7.4.3: на экран будет автоматически выведена таблица с предметами, прошедшими контрольные показатели (см рис 2.43) и лепестковая диаграмма с наглядным изображением критериев по каждому предмету (см рис 2.44).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.43 – метод контрольных показателей

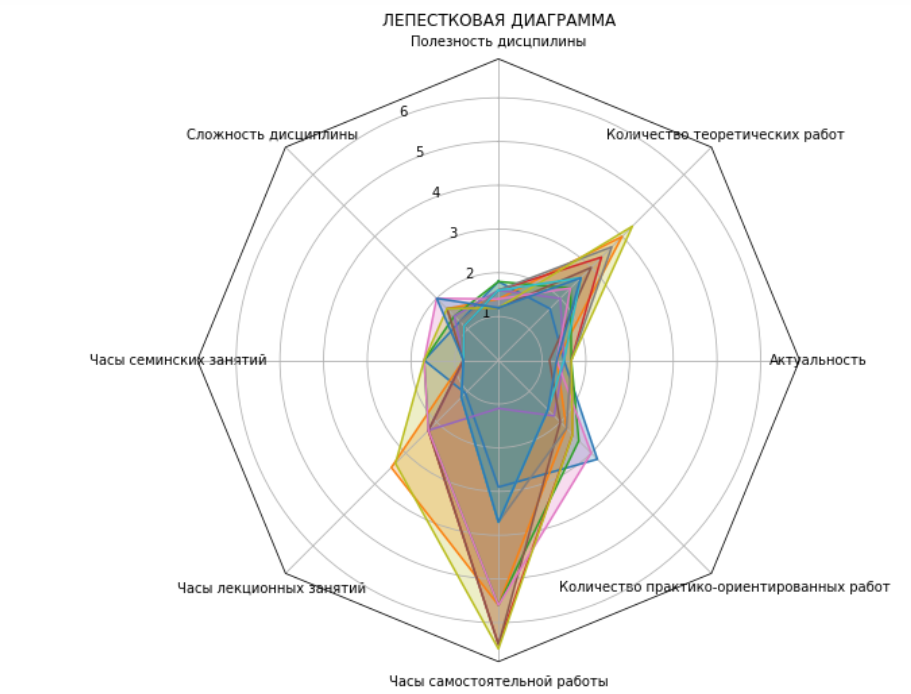


Рисунок 2.44 – лепестковая диаграмма

Пользователь самостоятельно определяет строку, содержащую максимальное значение по столбцу «Расстояние» (см рис 2.43), определяя таким образом более подходящий по требованиям предмет.

Шаг 7.5: если пользователь желает запустить все методы оптимизации, следует ввести цифру 5 в строке выбора программы (см рис 2.45) и по очереди следовать инструкциям, данным выше (см шаг 7.1, шаг 7.2, шаг 7.3 и шаг 7.4).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.45– Выбор программы 5

# **2.**[**5. Архитектура**](file:///C:\Users\79141\Downloads\Vtoroe_tz.docx#_Toc98442922) **решения алгоритма на языке программирования Python**

Для реализации всех имеющихся алгоритмов решения мы использовали следующие библиотеки:

* Pandas — библиотека для обработки и анализа структурированных данных;
* NumPy — это библиотека языка Python, добавляющая поддержку больших многомерных массивов (включая матрицы), поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами;
* Matplotlib — библиотека для визуализации данных;
* Turtle – библиотека графического вывода, построенная на метафоре черепахи

Опишем подробнее классы и функции, которые содержатся в коде и реализуют многокритериальную оптимизацию.

pretty\_table – функция, строящая таблицу-образец заполнения csv-файла с исходными данными

radar\_factory – функция, содержащая класс RadarAxes, предназначенная для построения лепестковых диаграмм

**Opt** — основной класс, содержащий функции многокритериальной оптимизации

Входные данные класса opt:

* Csv-файл, содержащий матрицу значений по дисциплинам и соответствующим критериям. (Подробнее о формате вводимых данных в csv файл описано в пункте 4.2)

Двумя главными атрибутами, внутренними переменными класса являются self.data и self.max\_min, содержащие матрицу из полученного на входе csv-файла и вектор направления оптимизации каждого из критериев. Значения считываются из файла благодаря методу библиотеки pandas – pd.read\_csv.

Для удобства вызова конкретного параметра была произведена декомпозиция задачи многокритериальной оптимизации на несколько функций, решающих конкретные подзадачи в классе opt.

В основном классе есть cледующие функции подсчета:

**criteria\_names()** – функция, извлекающая из исходной таблицы значений названия критериев

**names()** – функция, извлекающая из исходной таблицы значений названия дисциплин

**for\_two\_criteria()** – функция, рассчитывающая линейную свертку по двум критериям, принимает на вход список критериев, по которым требуется оптимизация и переменную, содержащую векторы долей для каждого из двух критериев.

**for\_all()** – функция, рассчитывающая свертку по всем критериям. Принимает на вход вектор значимостей каждого из критериев.

**Ideal\_dot()** – функция, производящая оптимизацию методом идеальной точки.

**benchmark()** – функция, производящая оптимизацию методом контрольных показателей.

# **2.6. Тестирование**

Для тестирования были использованы 5 датасетов, тесты производились следующими алгоритмами: Excel и Python. Хотелось бы заметить, что поиск пареттооптимального решения реализован только в Excel.

Данные первого датасета представлены в таблице 2.1:

Таблица 2.1 Первый датасет

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | макс | мин | макс | макс | макс | макс | макс | макс |
|  | Критерий 1 | Критерий 2 | Критерий 3 | Критерий 4 | Критерий 5 | Критерий 6 | Критерий 7 | Критерий 8 |
| Объект 1 | 9 | 8 | 34 | 16 | 58 | 16 | 9 | 5 |
| Объект 2 | 7 | 6 | 16 | 52 | 112 | 11 | 8 | 12 |
| Объект 3 | 9 | 7 | 34 | 34 | 112 | 13 | 10 | 7 |
| Объект 4 | 8 | 9 | 16 | 34 | 130 | 12 | 10 | 10 |
| Объект 5 | 8 | 7 | 16 | 34 | 22 | 9 | 9 | 6 |
| Объект 6 | 6 | 6 | 16 | 34 | 130 | 10 | 7 | 9 |
| Объект 7 | 7 | 5 | 34 | 34 | 112 | 15 | 8 | 7 |
| Объект 8 | 8 | 8 | 16 | 18 | 74 | 11 | 10 | 11 |
| Объект 9 | 6 | 6 | 34 | 50 | 132 | 12 | 10 | 13 |
| Объект 10 | 8 | 9 | 16 | 18 | 74 | 8 | 9 | 8 |
| Объект 11 | 6 | 5 | 16 | 18 | 74 | 8 | 8 | 8 |
| значимость | 8 | 6 | 7 | 2 | 3 | 5 | 9 | 4 |
| контрольный показатель | 5 | 10 | 20 | 15 | 20 | 5 | 6 | 3 |

В таблице 2.2 представлены результаты по данному датасету:

Таблица 2.2 Полученные результаты датасет 1

|  |  |  |  |
| --- | --- | --- | --- |
| Пареттооптимальные решения | | | |
| Паретооптимальные решения: объект 3, объект 7. | | | |
| Линейная свертка для двух критериев | | | |
| MS Excel | Python | | |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальные показатели: объект 3 и объект 7 |  | | |
| Линейная свертка всех критериев | | | |
| MS Excel | | Python | |
| Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 9 | | Оптимальный показатель: Объект 9 | |
| Метод идеальной точки | | | |
| MS Excel | | Python | |
| Изображение выглядит как текст, стол  Автоматически созданное описание  Оптимальный показатель: Объект 4 | | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 4 | |
| Метод контрольных показателей | | | |
| MS Excel | | | Python |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальный показатель: Объект 3 | | | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 3 |

Данные второго датасета представлены в таблице 2.3:

Таблица 2.3 Второй датасет

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | макс | мин | Макс | макс | макс | макс |
|  | **Критерий 1** | **Критерий 2** | Критерий 3 | Критерий 4 | Критерий 5 | Критерий 6 |
| Объект 1 | 8,5 | 3 | 10 | 10 | 10 | 6 |
| Объект 2 | 10 | 9 | 8 | 8 | 3 | 6 |
| Объект 3 | 7 | 8 | 5 | 8,5 | 7 | 7 |
| Объект 4 | 5 | 6 | 6 | 6,5 | 5 | 8 |
| Объект 5 | 8 | 6 | 8 | 8 | 8 | 7 |
| Объект 6 | 9 | 8 | 8 | 9 | 8 | 7 |
| значимость | 7 | -8 | 9,5 | 10 | 5 | 3 |
| Контрольный показатель | 6 | 10 | 4 | 7 | 3 | 5 |

В таблице 2.4 представлены результаты по данному датасету:

Таблица 2.4 Полученные данные датасет 2

|  |  |  |
| --- | --- | --- |
| Пареттооптимальные решения | | |
| Паретооптимальные решения: объект 1, объект 2, объект 6. | | |
| Линейная свертка для двух критериев | | |
| MS Excel | Python | |
| Изображение выглядит как текст, подключен  Автоматически созданное описание  Оптимальные показатели: объект 1 и объект 2 | Изображение выглядит как текст  Автоматически созданное описание | |
| Линейная свертка всех критериев | | |
| MS Excel | Python | |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальный показатель: Объект 1 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 1 | |
| Метод идеальной точки | | |
| MS Excel | Python | |
| Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 1 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 1 | |
| Метод контрольных показателей | | |
| MS Excel | | Python |
| Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 6 | | Изображение выглядит как стол  Автоматически созданное описание  Оптимальный показатель: Объект 6 |

Данные третьего датасета представлены в таблице 2.5:

Таблица 2.5 Третий датасет

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | макс | макс | Мин | мин | мин |
|  | Критерий 1 | Критерий 2 | Критерий 3 | Критерий 4 | Критерий 5 |
| Объект 1 | 1007 | 1,00 | 300 | 9 | 545 |
| Объект 2 | 1016 | 6,00 | 300 | 8 | 638 |
| Объект 3 | 999 | 5,50 | 300 | 7 | 257 |
| Объект 4 | 1003 | 6,33 | 300 | 6 | 753 |
| Объект 5 | 1011 | 4,20 | 300 | 5 | 694 |
| значимость | 10 | 8 | 6 | 6 | 6 |
| контрольный показатель | 1000 | 2,00 | 400 | 10 | 1000 |

В таблице 2.6 представлены результаты по данному датасету:

Таблица 2.6 Полученные результаты датасет 3

|  |  |
| --- | --- |
| Пареттооптимальные решения | |
| Паретооптимальные решения: объект 2, объект 4. | |
| Линейная свертка для двух критериев | |
| MS Excel | Python |
| Изображение выглядит как текст, кроссворд  Автоматически созданное описание  Оптимальные показатели: объект 2, объект 3, объект 4 | Изображение выглядит как стол  Автоматически созданное описание |
| Линейная свертка всех критериев | |
| MS Excel | Python |
| Изображение выглядит как текст, стол  Автоматически созданное описание  Оптимальный показатель: Объект 3 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 3 |
| Метод идеальной точки | |
| MS Excel | Python |
| Изображение выглядит как текст, стол  Автоматически созданное описание  Оптимальный показатель: Объект 2 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 2 |
| Метод контрольных показателей | |
| MS Excel | Python |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальный показатель: Объект 2 | Оптимальный показатель: Объект 2 |

Данные четвертого датасета представлены в таблице 2.7:

Таблица 2.7 Четвертый датасет

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | макс | макс | макс | макс | макс |
|  | Критерий 1 | Критерий 2 | Критерий 3 | Критерий 4 | Критерий 5 |
| Объект 1 | 704 | 94 | 604 | 6 | 351 |
| Объект 2 | 556 | 721 | 825 | 978 | 356 |
| Объект 3 | 32 | 238 | 810 | 762 | 736 |
| Объект 4 | 634 | 63 | 310 | 545 | 410 |
| Объект 5 | 454 | 113 | 311 | 835 | 707 |
| значимость | 6 | 7 | 8 | 1 | 6 |
| контрольный показатель | 500 | 50 | 4000 | 100 | 3000 |

В таблице 2.8 представлены результаты по данному датасету:

Таблица 2.8 Полученные результаты датасет 4

|  |  |  |
| --- | --- | --- |
| Пареттооптимальные решения | | |
| Паретооптимальные решения: объект 1, объект 2. | | |
| Линейная свертка для двух критериев | | |
| MS Excel | Python | |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальные показатели: объект 1 и объект 2 | Изображение выглядит как стол  Автоматически созданное описание | |
| Линейная свертка всех критериев | | |
| MS Excel | Python | |
| Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 1 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 1 | |
| Метод идеальной точки | | |
| MS Excel | Python | |
| Изображение выглядит как текст, стол  Автоматически созданное описание  Оптимальный показатель: Объект 3 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 3 | |
| Метод контрольных показателей | | |
| MS Excel | | Python |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальный показатель: Объект 1 | | Оптимальный показатель: Объект 1 |

Данные пятого датасета представлены в таблице 2.9:

Таблица 2.9 Пятый датасет

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | макс | мин | макс | мин | макс | мин | макс | мин | макс | мин |
|  | **критерий 1** | **критерий 2** | критерий 3 | критерий 4 | критерий 5 | критерий 6 | критерий 7 | критерий 8 | критерий 9 | критерий 10 |
| объект 1 | 3182,1 | 344 | 398 | 8842 | 6493 | 6390 | 7435 | 1780 | 5567 | 4103 |
| объект 2 | 3869,2 | 1348 | 9989 | 1291 | 5060 | 2598 | 951 | 5345 | 5281 | 5780 |
| объект 3 | 4553,3 | 3193 | 3475 | 657,5 | 2507 | 2083 | 469 | 9040 | 515 | 9902 |
| объект 4 | 4413,4 | 4908 | 7807 | 2256 | 5845 | 3772 | 2126 | 6918 | 4727 | 3572 |
| объект 5 | 2153 | 2067 | 5495 | 8207 | 9215 | 777 | 3935 | 2409 | 4711 | 4346 |
| объект 6 | 861 | 9764 | 8835 | 61 | 3871 | 2611 | 5925 | 2564,4 | 581 | 4047 |
| объект 7 | 8221 | 5614 | 4339 | 4511 | 3329 | 9101 | 8691 | 6646 | 8856 | 7531 |
| объект 8 | 2348 | 8056 | 1007 | 335 | 1253 | 276 | 720 | 2420 | 2728 | 6754 |
| объект 9 | 9592 | 4264 | 4455 | 940 | 3024 | 254 | 7148 | 3218 | 1928 | 9531 |
| объект 10 | 3869,2 | 1348 | 9989 | 1291 | 5060 | 2598 | 951 | 5345 | 5281 | 5780 |
| значимость | 10 | 9 | 7 | 5 | 3 | 6 | 8 | 3 | 5 | 6 |
| контрольный показатель | 3000 | 10000 | 3000 | 5000 | 1000 | 10000 | 200 | 5000 | 1000 | 6000 |

В таблице 2.10 представлены результаты по данному датасету:

Таблица 2.10 Полученные результаты датасет 5

|  |  |  |
| --- | --- | --- |
| Пареттооптимальные решения | | |
| Паретооптимальные решения: объект 1, объект 2, объект 3, объект 9. | | |
| Линейная свертка для двух критериев | | |
| MS Excel | Python | |
| Изображение выглядит как стол  Автоматически созданное описание  Оптимальные показатели: объект 1 и объект 9 | Изображение выглядит как стол  Автоматически созданное описание | |
| Линейная свертка всех критериев | | |
| MS Excel | Python | |
| Изображение выглядит как текст, стол  Автоматически созданное описание  Оптимальный показатель: Объект 9 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 9 | |
| Метод идеальной точки | | |
| MS Excel | Python | |
| Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 7 | Изображение выглядит как текст  Автоматически созданное описание  Оптимальный показатель: Объект 7 | |
| Метод контрольных показателей | | |
| MS Excel | | Python |
| Оптимальный показатель: нет | | Оптимальный показатель: нет |

Проанализировав таблицу, можно сделать следующие выводы: в линейной свертке двух критериев алгоритм Python, в отличие от алгоритма MS Excel, не выводит оптимальные объекты; в датасетах, где имеются значения с плавающей точкой (датасеты 3 и 5), значения свертки всех критериев отличаются по разряду; также хотелось бы отметить, что в методе идеальной точки по третьему датасету значения свертки различаются, однако найденный оптимальный вариант совпадает. Лепестковые диаграммы в Python и MS Excel также имеют схожий вид. За исключением особенностей форматирования, таблицы имеют аналогичный вид.

# **2.7 Заключение**

Решение для предоставленной задачи мы реализовали с помощью двух алгоритмов. Первый алгоритм основан на встроенных функциях Excel. Его преимуществами можно считать – простоту использования, редактирования и ввода новых данных, автоматическое обновление формул. Из недостатков – отсутствие пользовательского интерфейса, пользователь самостоятельно должен строить графики, проводить нормализацию, редактировать формулы в зависимости от направленности оптимизации критерия, что крайне неудобно для пользователя и, как показывает практика, может повлечь за собой ряд вычислительных ошибок.

Второй алгоритм, реализованный в Python имеет очевидное преимущество перед первым –интерфейс, все расчеты и построения проводятся автоматически, за исключением определения контрольных показателей, направленности оптимизации и значимости критериев. В данном алгоритме удобством является удобная загрузка данных, возможность вывести любой требуемый параметр методами реализованных классов. Есть возможность вывода всех способов оптимизации или выбора определенного. Программа имеет перспективы для расширения функционала, и более детальной доработки, так как можно дополнить лепестковые диаграммы легендой, выделить оптимальные решения во всех методах оптимизации, добавить возможность вывода данных, фронтенд.

Алгоритмы были проверены на пяти сформированных датасетах. На датасетах с целыми числами значения линейной свертки, метода идеальной точки и контрольных показателей идентичны. Если в исходной матрице есть числа с плавающей точкой, разрядность значений метода идеальной точки в Python и Excel может разниться, при том, что само число остается тем же, как и оптимальный вариант для выбора. Лепестковые диаграммы в обоих алгоритмах строятся одинаково. Более привлекательным для оптимизации методом для нас стал метод контрольных показателей, который сразу отсеивает точно не подходящие нам дисциплины. Метод свертки по всем критериям удобен, когда мы можем оценить значимость каждого из критериев. Если такая оценка не может быть произведена, стоит выбирать метод идеальной точки, где идеальный показатель можно рассчитать в процентном соотношении от максимума или минимума столбца.

Отвечая на поставленный вопрос технического задания, а именно, какую дисциплину стоит преподавать, можно сделать вывод, что самой часто встречающейся в результатах оптимизации стала дисциплина «Архитектура организации», а на втором месте «Основы управления информационными технологиями». По методу идеальной точки - «Управление информационно-технологическими проектами». В одной из сверток искомым стал результат - «Цифровой бизнес».

# **3. Экспертная оценка**

# **3.1. Постановка задачи (физическая модель)**

Наш заказчик – Минпромторг, федеральный орган исполнительной власти России, осуществляющий функции по выработке государственной политики и нормативно-правовому регулированию в сфере промышленного и оборонно-промышленного комплекса, а также в области развития торговли, также исполняющий функции уполномоченного федерального органа, осуществляющего государственное регулирование внешнеторговой деятельности.

В связи с нынешней политической и экономической ситуацией многие бренды уходят с российского рынка, однако остались права у владельцев франшизы осуществлять экономическую деятельность. В числе одной из основных франшиз стал McDonald’s, американская корпорация, работающая в сфере общественного питания, крупнейшая в мире сеть ресторанов быстрого питания, работающая по системе франчайзинга

Так как многие россияне любят посещать данную сеть ресторанов быстрого питания, уход с рынка McDonald’s стал бы огромным разочарованием для всех его постоянных клиентов. Весной 2022 года компания сообщила, что собирается продать свой бизнес на территории страны, который включает 850 ресторанов быстрого питания, в том числе и те, которые работают по франшизе. Поэтому было принято решения о покупке данной сети, новым владельцем российского бизнеса американской корпорации McDonald’s стал предприниматель из Новокузнецка Александр Говор, которому уже принадлежит 25 заведений этой сети, открытых по франчайзингу.

Однако стоит отметить, что данная сеть ресторанов быстрого питания ввела изменение в меню, так как существует ряд нюансов: авторские права на название и рецептуру соусов, проблема с упаковкой для части позиций, отсутствие поставок и другое.

Поэтому нашему заказчику необходимо определить степень удовлетворенности новым меню клиентов, которые ранее являлись посетителями McDonald’s. Поэтому были необходимо обозначить следующие нюансы в характеристиках, которые важны для клиентов: степень важности соотношения цена и качества и вкуса, помимо хорошей осведомленности старого меню.

Исходя из этого были отобраны следующие позиции нового меню для проведения экспертной оценки:

* Макчикен Премьер
* Филе-о-Фиш
* Двойной Роял
* Роял де Люкс
* Роял

По данным позициям и перечисленным выше характеристикам был проведен экспертный опрос (https://forms.gle/fy7cchstBbUUCBiD9), на основании результатов которого и проводился дальнейший анализ.

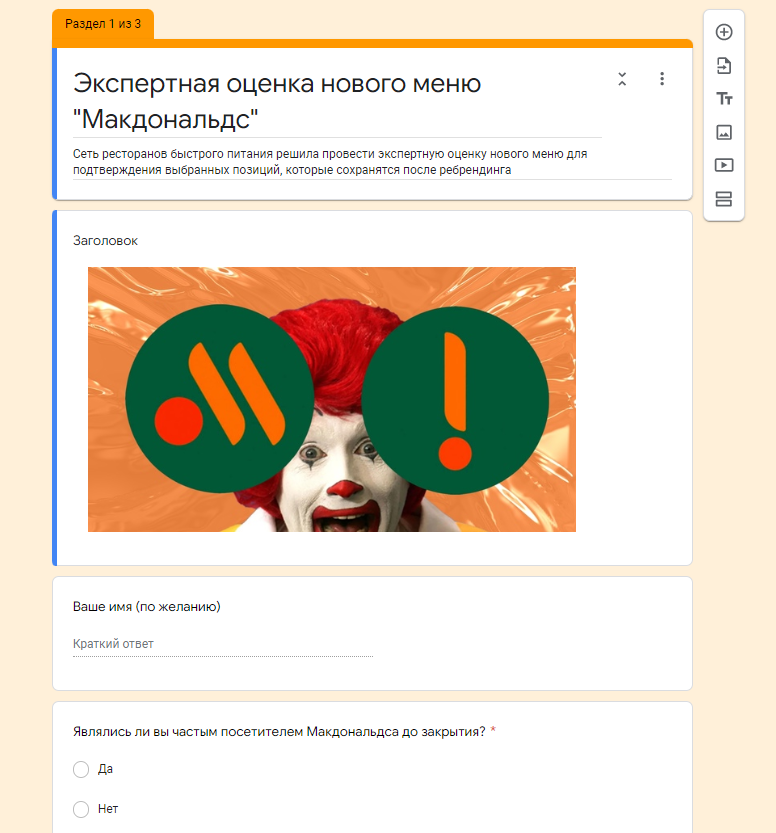


Рисунок 3.1 Форма, проведенного опроса

# **3.2.Математическая модель**

Рассмотрим математическую модель

**Метод средних баллов**

Обработка результатов проводится методом математической статистики:

Среднее количество баллов образца – математическое ожидание выборки:

(3.1)

Данный показатель отражает средний балл, который получил образец с учетом оценок всех экспертов.

Степень рассогласованности мнений экспертов – дисперсия по выборке:

(3.2)

Если дисперсия маленькая – мнения экспертов согласованы, в противном случае, их мнения на счет оцениваемых предметов сильно отличаются. С помощью этого показателя можно определить, насколько правильно подобраны эксперты и верная ли перед ними поставлена задача.

Интервальная оценка баллов – доверительный интервал, например :

(3.3)

(3.4)

Интервальная оценка отражает, насколько сильно разбросаны мнения экспертов, относительно среднего значения. На основе этой информации можно судить, корректно ли в итоге оценены сравниваемые объекты.

**Метод медианных рангов**

Медиана (Me) – это некоторая отметка, делящая ранжированные данные (отсортированные по возрастанию или убыванию) на две равные части. Половина исходных данных меньше этой отметки, а половина – больше.

**Метод средневзвешенных рангов**

Средневзвешенные ранги определяются с учетом компетентности всех экспертов, которые рассчитываются по следующему алгоритму:

1. Определяется среднее арифметическое рангов кандидатов (матожидание):

(3.5)

2. По каждому эксперту рассчитывается степень отклонения его ответа от среднего значения М – то есть дисперсия мнений каждого эксперта:

(3.6)

3. Чем больше дисперсия, тем сильнее мнение эксперта отличается от мнения большинства. Следовательно, его компетентность в данном вопросе можно считать ниже, чем у коллег (исключая догму «согласованности»).

Компетентность эксперта K будет обратно пропорциональна его дисперсии D:

(3.7)

4. Выразим значимость мнения каждого эксперта в процентах (нормируем) и приведем его к виду коэффициента компетентности k:

(3.8)

5. Далее найдем средневзвешенное мнение экспертов относительно кандидатов с учетом коэффициента их компетентности (важности их мнения):

(3.9)

**Бинарные отношения**

Метод попарных сравнений - один из наиболее сложных методов.

Применяется в случаях, когда объекты настолько комплексные и многомерные, что нет возможности от ранжировать их по одному признаку, но есть возможность попарно сравнить их между собой. Каждый эксперт должен сравнить каждую пару вариантов друг с другом. Результат записывается в виде матриц бинарных отношений.

Принцип применения метода, следующий: например, имеется два объекта с условными названиями а и b и метрика [0, 1]. В случае, если объект, а лучше или равен объекту b, метрика приобретает значение 1, в противном случае – значение 0:

(3.10)

(3.11)

Если сравниваемых объектов несколько 𝑎1, 𝑎2, ... , 𝑎𝑛, то результатом их по-парных сравнений будет являться матрица бинарных отношений:

(3.12)

Расстоянием Кемени между бинарными отношениями X и Y называется число, характеризующее количество несовпадающих элементов матриц X и Y:

(3.13)

Медианой Кемени называется такая матрица А, суммарное расстояние Кемени от которой до всех остальных матриц является минимальным:

(3.14)

Найти медиану Кемени можно с помощью задачи нелинейного программирования:  
Исходные данные: матрицы бинарных отношений

Искомая переменная: медиана Кемени А.  
Целевая функция – минимальное суммарное расстояние до всех матриц:

(3.15)

Ограничения:

(3.16)

(3.17)

# **3.3. Алгоритм решения**

**3.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel**

### **3.3.1.1. Описание входных данных**

В качестве входных данных алгоритм принимает:

* Таблица, содержащая список экспертов и соответствующие им результаты по каждому вопросу проведенного опроса, выгруженная в файл Excel

### **3.3.1.2. Описание алгоритма решения**

Алгоритм получает на вход таблицу, содержащую список экспертов и оценки и ранги, полученные в результате опроса. Пользователь форматирует данные, а далее производит математическую обработку экспертных оценок, находя математическое ожидание, дисперсию, среднеквадратичное отклонение, доверительные интервалы и строя диаграмму, также производит математическую обработку ранговых оценок, находя медиану, коэффициенты компетентности и средневзвешенные ранги с учетом коэффициентов компетентности.

*Фильтрация*

Так как выгруженные данные из опроса (рис. 3.2) имеют неудобный для расчетов вид, их надо привести к матрицам продукт-эксперт, но перед этим определить кто из прошедших опрос действительно является экспертом. Для этого были сформулированы два вопроса в форме «Являлись ли вы частым посетителем Макдональдса до закрытия» и «Знакомы ли вы со старым меню Макдональдса».

*Изображение выглядит как текст, карта, внутренний, украшен

Автоматически созданное описание*

Рисунок 3.2 Неотфильтрованные результаты опроса

Очевидно, что люди, которые были частыми клиентами и знакомы с меню – наши главные эксперты, но те, кто не был частым посетителем, однако знаком с меню, также могут считаться экспертами. Люди, которые не были посетителями и не знакомы с меню – отфильтровываются, так как их мнение нельзя считать экспертным, так же, как и люди, которые обозначили, что они являются частыми клиентами, однако они не знакомы с меню, так как это противоречит логике и говорит о недобросовестном прохождении формы.

Итак, чтобы произвести фильтрацию, получается, что надо пройтись по столбцу «Знакомы ли вы с меню» и удалить тех респондентов, которые ответили «нет». После чего удаляются лишние столбцы таблиц и рассчитывается средневзвешенная оценка по каждому из продуктов, чтобы получить матрицу экспертных, затем матрица транспонируется и принимает вид, в котором продукты располагаются по строкам, а эксперты по столбцам.

Аналогично выбираются столбцы с ранговыми оценками по каждому из критериев (вкус, цена-качество) и составляются две отдельные матрицы ранговых оценок.

*Пошаговая реализация алгоритма*

Перед выполнением пользователь должен выгрузить таблицу с результатами опроса из google forms.

Шаг 1: выделяем ячейки с исходными данными и форматируем как таблицу.

Шаг 2: в столбце F "Знакомы ли вы со старым меню?" производим фильтрацию от А до Я

Шаг 3: строки, в которых значения в столбце F являются "Нет" (последние строки), удаляем.

Шаг 4: добавляем новые столбцы согласно количеству оцениваемых продуктов, называя столбцы по наименованию продукта. (рис. 3.3)

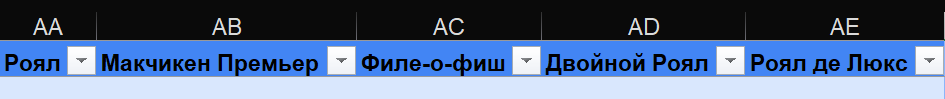


Рисунок 3.3 – Добавление новых столбцов

Шаг 5: далее находим средневзвешенное значение, путем умножения коэффициента значимости критериев вкуса и цены/качества на соответствующую оценку эксперта. Формула для расчета продемонстрирована на Рисунке 3.4:

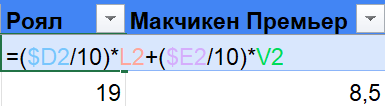


Рисунок 3.4 – Формула для расчета средневзвешенной оценки

Шаг 6: в книге Excel создаем новый лист.

Шаг 7: копируем столбец с именами экспертов и полученные в шаге 4 столбцы с их оценками, а затем вставляем с помощью параметра вставки «Транспонировать» (рис. 3.5)

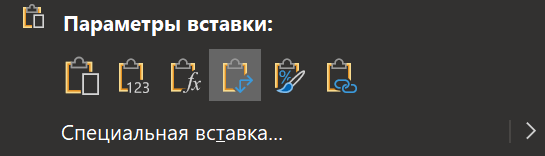


Рисунок 3.5 – Параметр вставки "Транспонировать"

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.6- Полученная в результате транспонирования матрица

Шаг 8: С помощью функции Excel СРЗНАЧ() находим математическое ожидание по каждому объекту оценивания (рис. 3.7).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.7 – Нахождение МО

Шаг 9: с помощью функции Excel ДИСП.Г() находим дисперсию по каждому объекту оценивания (рис. 3.8).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.8 – Нахождение дисперсии

Шаг 10: находим среднеквадратичное отклонение согласно формуле 3.4.

Шаг 11: находим доверительные интервалы согласно формуле 3.3.

Шаг 12: далее строим вспомогательную таблицу (рис. 3.9) для построения диаграммы.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.9 - Вспомогательная таблица

Шаг 13: на основе вспомогательной таблицы строим биржевую диаграмму (рис. 3.10)

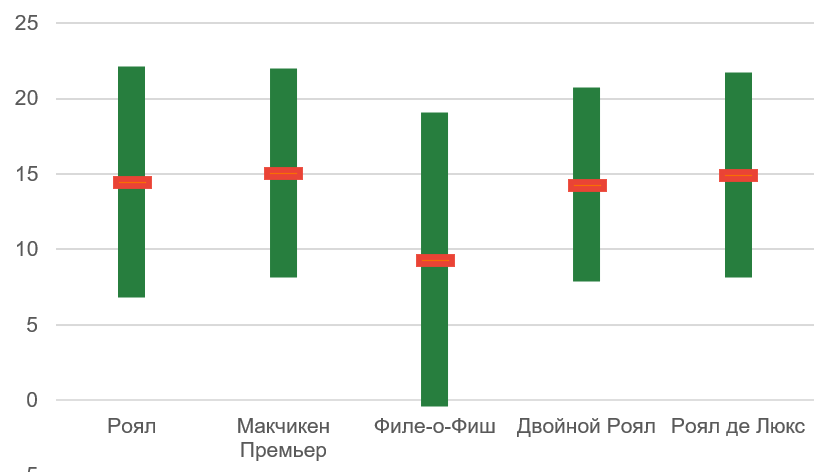


Рисунок 3.10 – Биржевая диаграмма

Шаг 14: С листа с исходными данными копируем данные с ранжированием и вставляем на новый лист, применяя параметр ставки «Транспонирование» (рис. 3.5)

Шаг 15: Далее удаляем слово «место» из каждого значения (рис. 3.11) (Главное > Редактирование > Найти и выделить > Заменить).

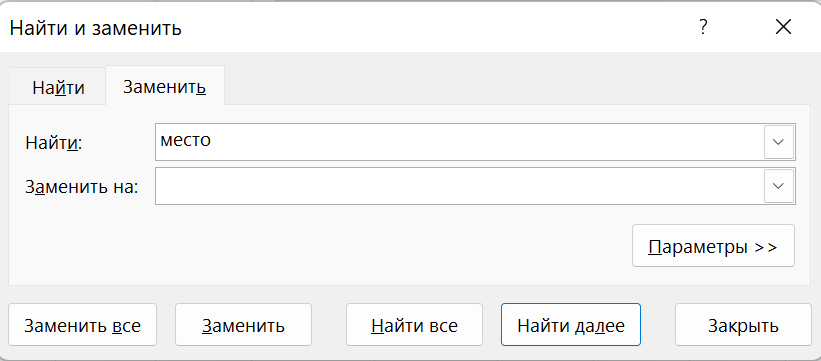


Рисунок 3.11 – Форматирование для ранговых оценок

Шаг 16: находим медиану по каждому объекту оценивания с помощью функции МЕДИАНА().

Шаг 17: для нахождения дисперсии произведем промежуточные расчеты. Найдем математическое ожидание по каждому объекту оценивания с помощью функции СРЗНАЧ()

Шаг 18: С помощью функции СУММКВРАЗН() найдем дисперсию, формула продемонстрирована на Рисунке 3.12:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.12 – Нахождение дисперсии

Шаг 19: далее найдем коэффициент компетентности экспертов, согласно формуле 3.7.

Шаг 20: просуммируем полученные в предыдущем шаге значения с помощью функции СУММ(). Данное действие показано на Рисунке 3.13:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3.13 – Суммирование коэффициентов

Шаг 21: отразим значимость мнения каждого эксперта в процентах, согласно формуле 3.8 (рис. 3.14), а также применим процентный формат (Главная > Число > Процентный формат).



Рисунок 3.14 – Значимость мнения в процентах

Шаг 22: Далее найдем средневзвешенные ранги с учетом коэффициентов компетентности с помощью функции СУММПРОИЗВ() (рис. 3.15).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.15 – Средневзвешенные значения рангов

Шаг 23: для метода бинарных отношений составляем матрицы для каждого эксперта, где сравниваем каждую пару объектов друг с другом. 1, если объект по строке лучше, чем объект по столбцу, и 0 – в противном случае. Составленные матрицы показаны на Рисунке 3.16.



Рисунок 3.16 - Матрицы для бинарных отношений

Шаг 24: далее найдем медиану Кемени. Для этого составим матрицу А, где значения над главной диагональю искомые переменные, под главной диагональю – противоположные транспонированные бинарные значения (рис. 3.17)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.17 - Матрица А

Шаг 25: Для вычисления расстояния Кемени применяем функцию СУММКВРАЗН() (рис. 3.18).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.18 - Расчет расстояний

Шаг 26: создаем целевую функцию с помощью функции Excel СУММ() (рис. 3.19)

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.19 - Создание целевой функции

Шаг 27: применим средство анализа Excel «Поиск решения». Заполняем параметры поиска решения согласно математической модели: ограничением является то, что все переменные могут принимать только бинарные значения, целевая функция стремится к минимуму, задаем ячейки переменных, также выбираем метод решения - «Поиск решения нелинейных задач методом ОПГ». Заполненное окно поиска решения показано на Рисунке 3.20:

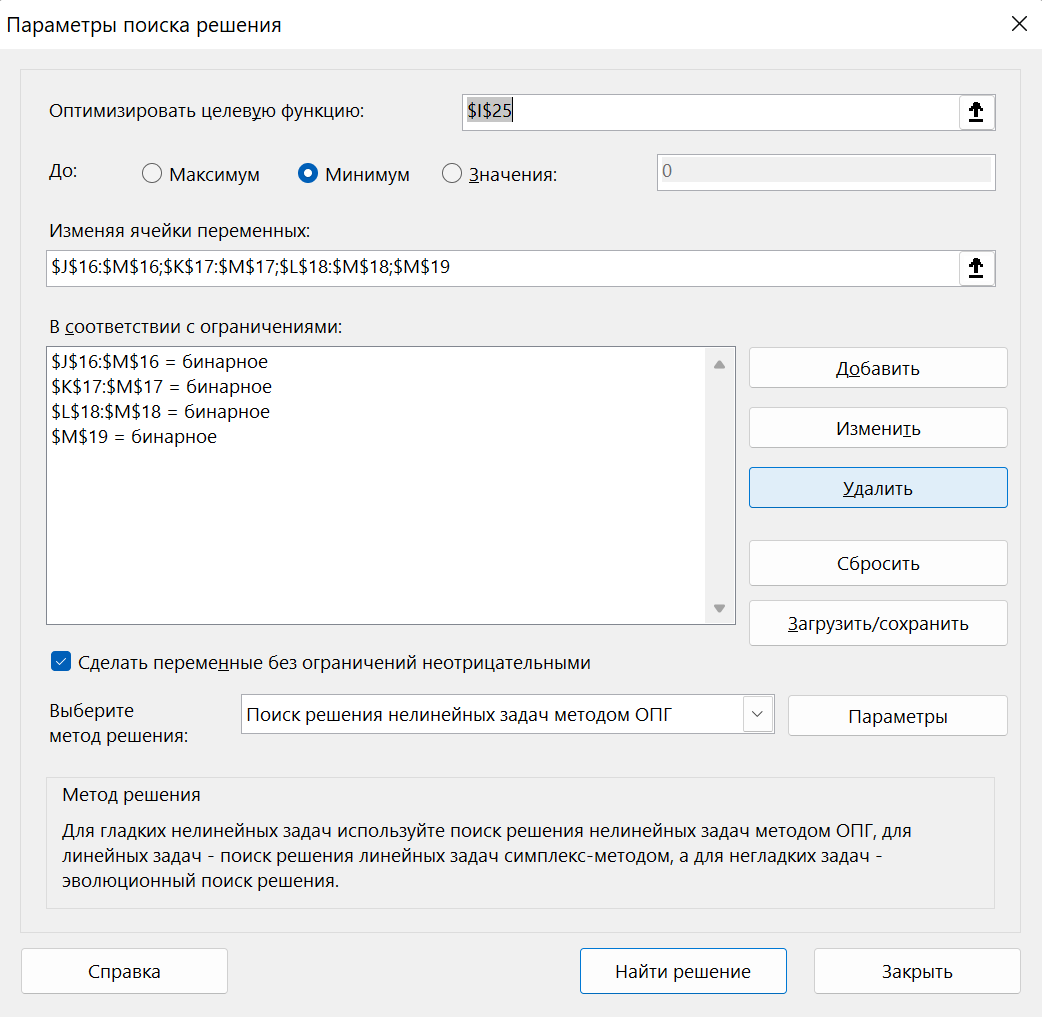


Рисунок 3.20 - Параметры поиска решения

### **3.3.1.3. Описание выходных данных**

Выходными данными алгоритма являются:

*Математическая обработка экспертных оценок*

* Средний балл (матожидание) по каждому продукту;
* Дисперсия по каждому продукту;
* Среднеквадратичное отклонение для каждого продукта;
* Доверительный интервал каждого из продуктов
* Гистограмма вариантов

*Математическая обработка ранговых оценок*

* Медиана рангов
* Массив коэффициентов компетентности экспертов;
* Средневзвешенные ранги с учетом коэффициентов компетентности;

## **3.3.2. Алгоритм 2. Решение в Python**

### **3.3.2.1. Описание входных данных**

В качестве входных данных мы имеем:

* Таблица, содержащая список экспертов и соответствующие им результаты по каждому вопросу проведенного опроса, выгруженная в файл Csv

**3.3.2.2. Описание алгоритма решения**

Алгоритм решения поставленной задачи реализован при помощи класса expert\_evaluations, который содержит функции математической обработки экспертных оценок и ранговых оценок, внутри которого есть функция run(), реализующая консольный ввод данных для пользователя, а также с помощью функции clear\_data, производящей фильтрацию выгруженных данных с опроса. Подробнее данные объекты описаны в пункте 3.5.

*Фильтрация*

Так как выгруженные данные из опроса (рис. 3.21) имеют неудобный для расчетов вид, их надо привести к матрицам продукт-эксперт, но перед этим определить кто из прошедших опрос действительно является экспертом. Для этого были сформулированы два вопроса в форме «Являлись ли вы частым посетителем Макдональдса до закрытия» и «Знакомы ли вы со старым меню Макдональдса».

*Изображение выглядит как текст, карта, внутренний, украшен

Автоматически созданное описание*

Рисунок 3.21 Выгруженные результаты опроса

Очевидно, что люди, которые были частыми клиентами и знакомы с меню – наши главные эксперты, но те, кто не был частым посетителем, однако знаком с меню, также могут считаться экспертами. Люди, которые не были посетителями и не знакомы с меню – отфильтровываются, так как их мнение нельзя считать экспертным, так же, как и люди, которые обозначили, что они являются частыми клиентами, однако они не знакомы с меню, так как это противоречит логике и говорит о недобросовестном прохождении формы.

Итак, чтобы произвести фильтрацию, получается, что надо пройтись по столбцу «Знакомы ли вы с меню» и удалить тех респондентов, которые ответили «нет». После чего удаляются лишние столбцы таблиц и рассчитывается средневзвешенная оценка по каждому из продуктов (подробно о расчете в пункте 3.3.1), после чего матрица экспертных оценок транспонируется, так как продукты должны располагаться по строкам, а эксперты по столбцам.

Затем выбираются столбцы с ранговыми оценками по каждому из критериев (вкус, цена-качество) и составляется две отдельные матрицы ранговых оценок.

*Пошаговая реализация алгоритма*

Шаг 1: Пользователь выбирает какой расчет он хочет произвести, в случае выбора математической обработки экспертных оценок, он вводит название csv файла, который передается в класс expert\_evaluations. (Если речь идет о неотфильтрованных данных опроса, то они передаются в функцию clear\_data).

Шаг 2: Класс формирует переменную с искомой матрицей экспертов и их оценками продуктов.

Шаг 3: Функция класса expert() производит расчеты по математической обработке экспертных оценок (математическое ожидание, дисперсия, среднеквадратичное отклонение, доверительные интервалы).

Шаг 4: Функция expert() строит биржевую диаграмму.

Шаг 5: Далее, при выборе обработки ранговых оценок, пользователь снова вводит название csv файла, но уже содержащим матрицу ранговых оценок, который также передается в класс expert\_evaluations.

Шаг 6: Функция класса rank() рассчитывает медианы по продуктам, коэффициенты компетентности экспертов и средневзвешенные ранги по продуктам.

### **3.3.2.3. Описание выходных данных**

Выходными данными алгоритма являются:

*Математическая обработка экспертных оценок*

* Средний балл (матожидание) по каждому продукту;
* Дисперсия по каждому продукту;
* Среднеквадратичное отклонение для каждого продукта;
* Доверительный интервал каждого из продуктов
* Гистограмма вариантов

*Математическая обработка ранговых оценок*

* Медиана рангов
* Массив коэффициентов компетентности экспертов;
* Средневзвешенные ранги с учетом коэффициентов компетентности;

# **3.4. Варианты использования алгоритмов**

## **3.4.1. Варианты использования алгоритма 1 – поиск решений Excel**

Пользователь вводит данные путём загрузки в Excel из Google Forms. Важно учитывать, что анализ производится для 5 позиций меню и 50 экспертов, если пользователь желает увеличить количество позиций и экспертов, ему стоит обратиться к главе 3.3.1.2. Описание алгоритма решения.

Шаг 1: пользователь выгружает данные опроса на новый лист, выделяет ячейки с исходными данными и форматирует как таблицу (см. рис. 3.22).

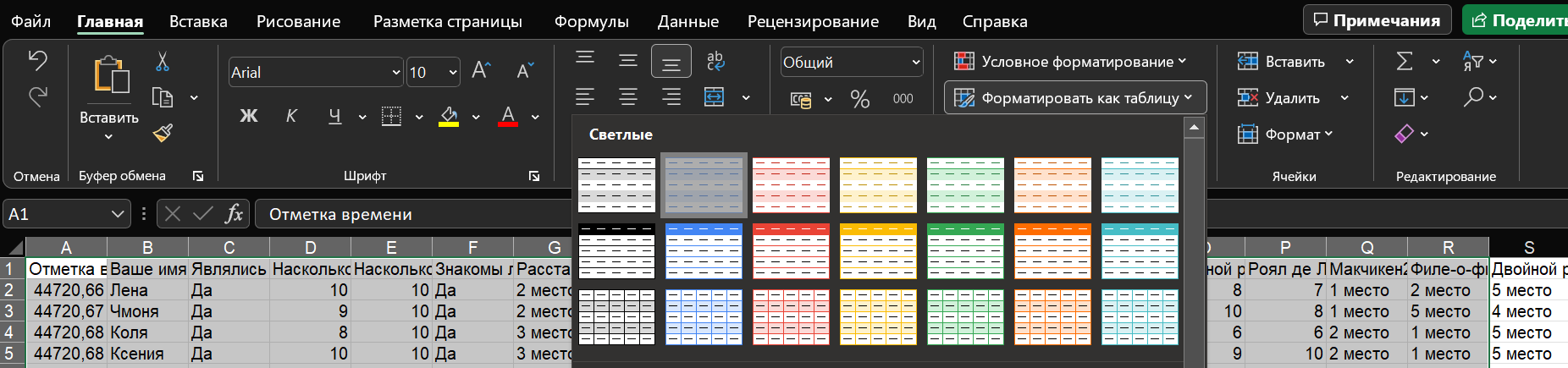


Рисунок 3.22 – Форматирование таблицы

Шаг 2: в столбце F "Знакомы ли вы со старым меню?" пользователь производит фильтрацию от А до Я (см. рис. 3.23).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3.23 – Сортировка от А до Я

Шаг 3: чтобы исключить экспертов не знакомых с меню, пользователь удаляет строки, в которых значения в столбце F являются "Нет" (последние строки). Пользователь оставляет 50 экспертов для анализа.

Шаг 4: пользователь копирует последние 5 столбцов с формулами средневзвешенного значения из листа «Данные» и вставляет на лист с текущими данными (см. рис. 3.24). При необходимости пользователь переименовывает столбцы по наименованию продукта.

Изображение выглядит как текст, табло, снимок экрана

Автоматически созданное описание

Рисунок 3.24 – Вставка столбцов с формулами

Шаг 5: после успешного добавления формул в таблицу, пользователь может удалить лист со старыми данными.

Шаг 6: для дальнейшего анализа пользователь с зажатой клавишей «Ctr» копирует столбец с именами и последние 5 столбцов с формулами, после чего вставляет их в первую таблицу (A1:AY6) на листе «Экспертная оценка» с помощью специальной вставки. В параметрах следует отметить кнопки «значения» и «транспонировать» (см. рис. 3.25).

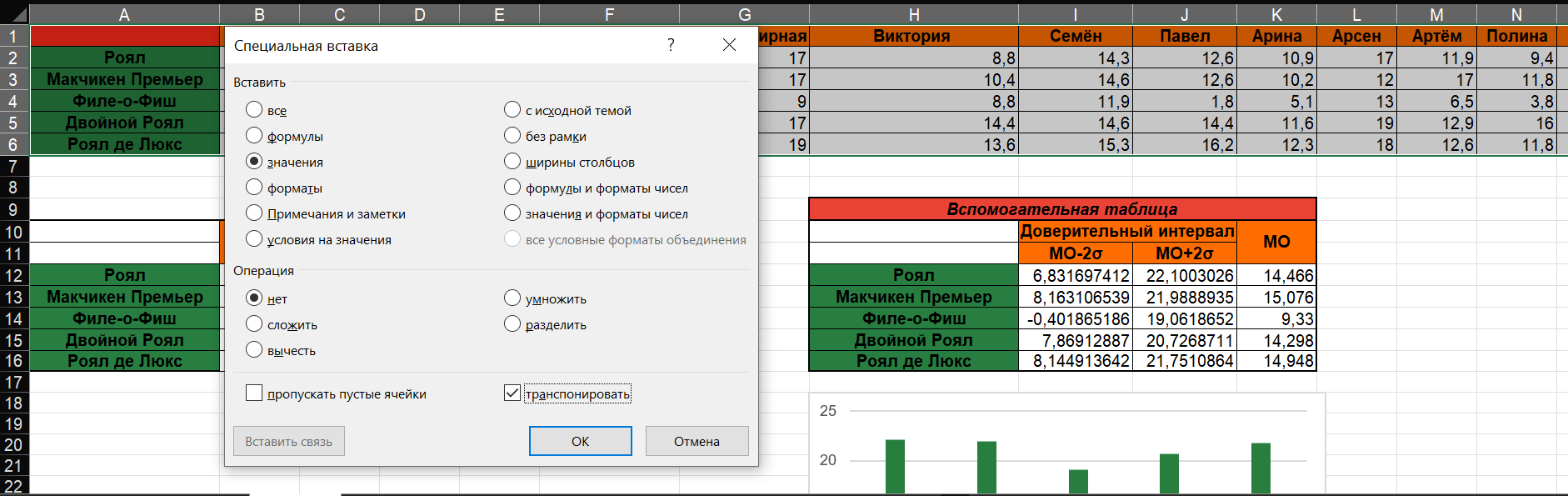


Рисунок 3.25 – Специальная вставка

Шаг 7: В таблице (A10:F16) автоматически будут высчитаны значения математического ожидания, дисперсии, среднеквадратичного отклонения и доверительные интервалы. Также будет высчитана вспомогательная таблица и биржевая диаграмма, построенная на её основе (см. рис. 3.26).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.26 – Построение таблиц со статистическими данными

Шаг 8: с листа с исходными данными пользователь копирует столбец с именами и пять столбцов с ранжированием (со словом «место») и вставляет в первую таблицу на лист «Ранговая оценка», применяя специальную вставку (см. шаг 6 и рис. 3.5)

Шаг 9: пользователь удаляет слово «место» из каждого значения с помощью функции «Найти и заменить» на пути «Главное > Редактирование > Найти и выделить > Заменить» или способом зажатия клавиш Ctrl+ F, во всплывающем окне выбирается «Заменить все» (см. рис. 3.11).



Рисунок 3.27 – Форматирование для ранговых оценок

Шаг 10: во второй таблице (A8:D13) автоматически будут посчитаны медиана, средневзвешенные значения рангов и математическое ожидание (см. рис. 3.28). При необходимости пользователь переименовывает строки по наименованию продукта.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.28 – Вторая таблица

Шаг 11: в третьей таблице (A15:AZ17) автоматически будут посчитаны дисперсия и коэффициенты компетентности (см. рис. 3.29).

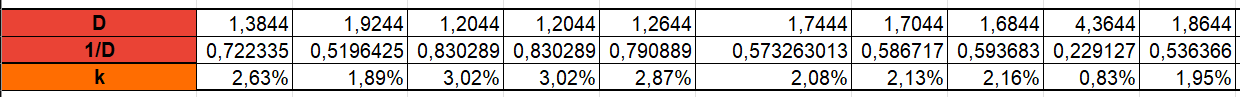


Рисунок 3.29 – Третья таблица

Шаг 12: для метода бинарных отношений пользователь открывает лист «бинарные отношения» и составляет/обновляет матрицы для каждого эксперта (предусмотрено 5 экспертов, если пользователь желает увеличить количество экспертов, ему стоит обратиться к главе 3.3.1.2. Описание алгоритма решения), где сравнивает каждую пару объектов друг с другом.

1 -если объект по строке лучше, чем объект по столбцу;

0 – в противном случае.

Составленные матрицы показаны на Рисунке 3.30.



Рисунок 3.30 - Матрицы для бинарных отношений

Шаг 13: под матрицей 5-го эксперта находится матрица А, где значения над главной диагональю – искомые переменные, под главной диагональю – противоположные транспонированные бинарные значения. Пользователь очищает искомые переменные над главной осью (см. рис. 3.31).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.31 – Матрица А

Шаг 14: для поиска целевой функции пользователь применяет средство анализа Excel «Поиск решения». Параметры поиска решения заранее настроены (см. рис. 3.32). В результате искомые значения над главной диагональю будут найдены, а целевая функция высчитана (см. рис. 3.33).

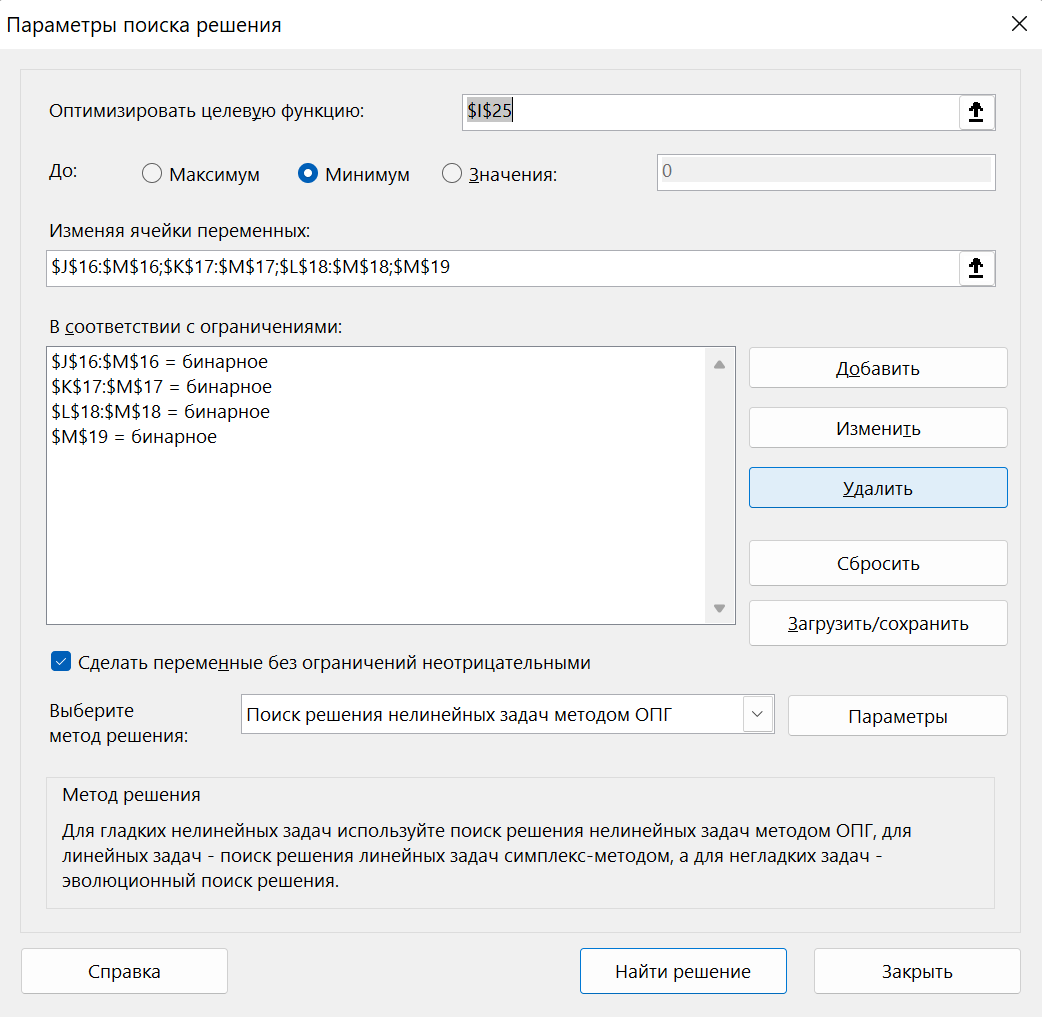


Рисунок 3.32 - Параметры поиска решения

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.33 – Результат применения «Поиска решения»

3.4.2. Варианты использования алгоритма 2 – Python

Пользователь может ввести данные путём загрузки csv файла. Перед выполнением действий для ввода данных необходимо запустить все окна кода с библиотеками для корректной работы программы. Стоит учитывать, что csv-файлы и ipynb-файлы должны находиться в одной папке (любой кроме «Загрузки»).

Шаг 1: пользователь запускает два окна с загрузкой библиотек для дальнейшей работы.

Шаг 2: пользователь запускает третье окно ввода с основным кодом.

Шаг 3: пользователь запускает четвёртое окно ввода с пометкой «Для работы запустить эту часть кода». При правильной работе программы на экран будет выведен выбор метода решения (см рис 3.34), при возникновении ошибок см. шаг 1–2.

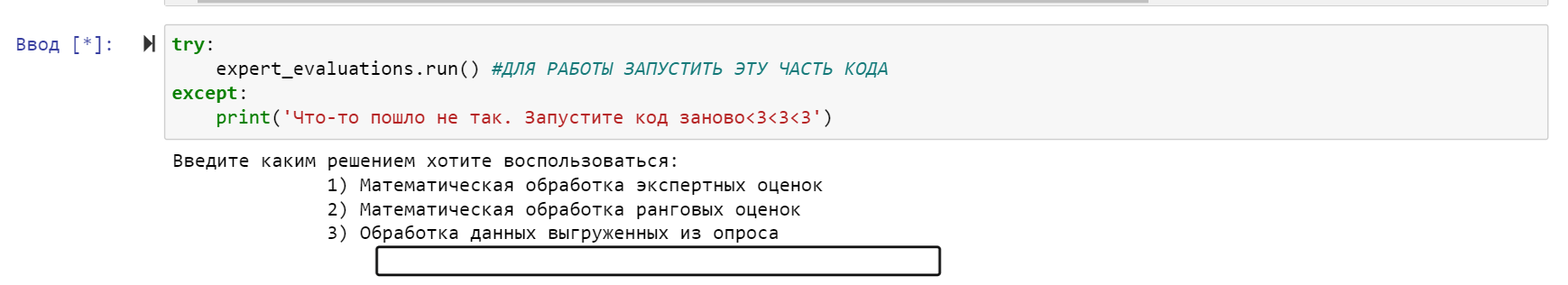


Рисунок 3.34 – запуск программы

Шаг 4.1.1: если пользователь желает запустить программу математической обработки экспертных оценок, следует ввести цифру 1 в строке выбора программы (см. рис. 3.35).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3.35 – Выбор программы 1

Шаг 4.1.2: в окно ввода названия csv-файла пользователь вводит наименование файла с данными в формате «name.csv». Данные в файле должны представлять из себя таблицу транспонированных данных см. Шаг1-6 в 3.4.1. Варианты использования алгоритма 1 – поиск решений Excel.

Изображение выглядит как стол

Автоматически созданное описание

Шаг 4.1.3. после успешной загрузки csv-файла автоматически будут высчитаны значения математического ожидания, дисперсии, среднеквадратичного отклонения и доверительные интервалы (см. рис. 3.36).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.36 – Статистические данные

Также будет высчитана биржевая диаграмма (см. рис. 3.37).

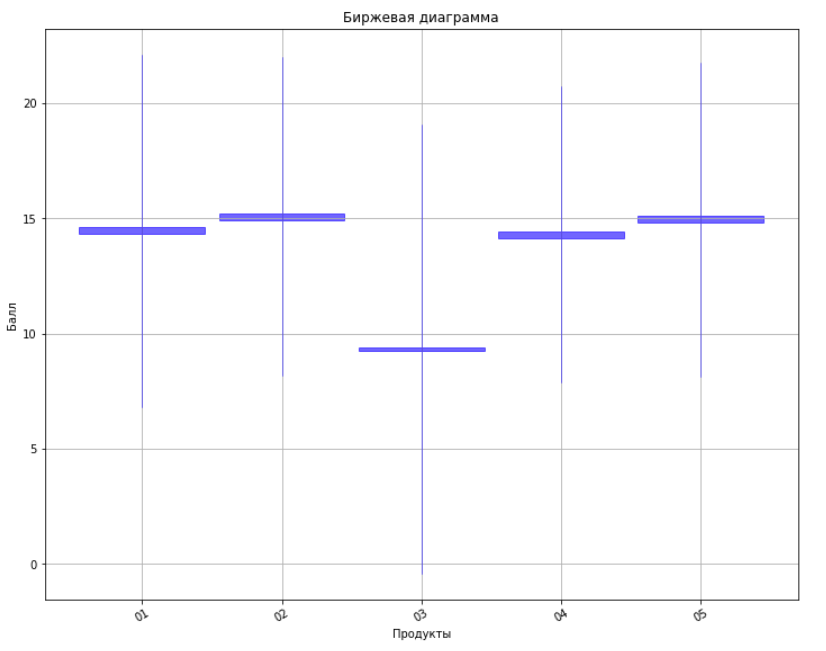


Рисунок 3.37 - Диаграмма

Шаг 4.1.4: пользователь имеет возможность продолжить или завершить работу (см. рис. 3.38). Для этого в окне ввода пользователь должен ввести да, если намерен продолжать или нет, если хочет закончить.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.38 – Выбор действия

Шаг 4.2.1: если пользователь желает запустить программу математической обработки ранговых оценок, следует ввести цифру 2 в строке выбора программы (см. рис. 3.39).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3.39 – Выбор программы

Шаг 4.2.2: в окно ввода названия csv-файла пользователь вводит наименование файла с данными в формате «name.csv». Данные в файле должны представлять из себя таблицу транспонированных данных см. Шаг8-9 в 3.4.1. Варианты использования алгоритма 1 – поиск решений Excel.

Шаг 4.2.3: автоматически будут посчитаны медиана по продуктам, коэффициенты компетентности и средневзвешенные ранги по продуктам (см. рис. 3.40).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.40 – Выходные данные программы математической обработки ранговых оценок

Шаг 4.2.4: пользователь имеет возможность продолжить или завершить работу (см. рис. 3.38 в 4.1.4). Для этого в окне ввода пользователь должен ввести да, если намерен продолжать или нет, если хочет закончить.

Шаг 4.3.1: если пользователь желает обработать данные из опроса, следует ввести цифру 3 в строке выбора программы (см. рис. 3.41).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3.41 – Выбор программы

Шаг 4.3.2: в окно ввода названия csv-файла пользователь вводит наименование файла с данными в формате «name.csv». Данные в файле должны представлять из себя необработанные данные ответов экспертов на опрос.

Шаг 4.3.3: автоматически будет произведена обработка данных и сохранена в следующих файлах csv (см. рис. 3.42).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3.42 – Отсортированные файлы csv

Шаг 4.3.4: на экран будет выведен выбор метода решения для очищенных файлов (см. рис. 3.43).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3.43 – Методы для обработанных файлов

Шаг 4.3.5: при выборе 1 решения будут произведены расчёты на основе файла «mcdonalds\_expert.csv» аналогичные шагам 4.1.1–4.1.4;

При выборе 2 решения будут произведены расчёты на основе файла «flavor.csv» аналогичные шагам 4.2.1–4.2.4;

При выборе 3 решения будут произведены расчёты на основе файла «quality.csv» аналогичные шагам 4.2.1–4.2.4.

# **3.5. Архитектура решения алгоритма на языке программирования Python**

Для реализации всех имеющихся алгоритмов решения мы использовали следующие библиотеки:

* Pandas — библиотека для обработки и анализа структурированных данных;
* NumPy — это библиотека языка Python, добавляющая поддержку больших многомерных массивов (включая матрицы), поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами;
* Matplotlib — библиотека для визуализации данных;
* Tabulate – модуль, который используется для печати табличных данных в красиво отформатированных таблицах.
* Datetime– модуль предоставляет классы для обработки времени и даты разными способами.
* Matplotlib.dates–библиотека, позволяющая преобразовать даты в числовой формат, понятный для Matplotlib для построения биржевой диаграммы

Все вычисления по математической модели производятся при помощи следующего объекта:

**Expert\_evaluations** — основной класс, содержащий функции, реализующие математическую обработку экспертных и ранговых оценок.

Входные данные класса:

* Csv-файл, содержащий матрицу c экспертами и их оценками по каждому из продуктов. (Подробнее о формате вводимых данных в csv файл описано в пункте 3.4.2)

Главным атрибутом, внутренней переменной класса является data, содержащая матрицу из полученного на входе csv-файл. Значения считываются из файла благодаря методу библиотеки pandas – pd.read\_csv.

Для удобства вызова конкретного параметра была произведена декомпозиция ТЗ на несколько функций, решающих конкретные подзадачи в классе Expert\_evaluations.

**expert() –** функция, производящая расчеты по математической оценке экспертных оценок и строящая биржевую диаграмму.

Сначала матрица оценок считывается при помощи метода read\_csv библиотеки pandas и записывается в переменную df.

Для дальнейших вычислений матрица оценок возводится в квадрат (переменная square\_df). Затем рассчитывается математического ожидание по каждому из продуктов с помощью метода mean. rows и записывается циклом for в переменную mean\_list, которая выводится в выходную таблицу в виде столбца «Матожидание».

Дисперсия рассчитывается по формуле, для чего требуется сначала найти сумму квадратов по продукту и квадрат суммы по оценкам продукта. Переменная sum\_squre\_list содержит столбец сумм квадратов, а переменная square\_sum\_list – квадрат суммы. Результат подсчета при помощи цикла for записывается в переменную general\_variance\_list и выводится в выходную таблицу в столбец «Дисперсия по продукту».

Аналогично в переменную standart\_deviation\_general\_list при помощи цикла for записываются значения стандартных отклонений по каждому продукту равных квадратному корню из дисперсии.

Границы доверительных интервалов для каждого продукта записаны в переменные left\_margin (левая граница) и right\_margin(правая граница) все также с помощью цикла for, путем прибавления или убавления удвоенного значения стандартного отклонения.

Далее необходимые переменные (left\_margin, right\_margin, а также переменные, содержащие доверительные интервалы для среднего: left\_mean, right\_mean) передаются в список diagramm\_list, на основе которого строится биржевая диаграмма.

**rank()** – функция, производящая расчеты, по математической оценке, ранговых оценок.

Матрица рангов считывается при помощи метода read\_csv библиотеки pandas и записывается в переменную df.

Затем рассчитывается медиана по каждому из продуктов с помощью метода row.median и записывается циклом for в переменную mediana\_list, которая выводится в выходную таблицу в виде столбца «Медиана по продуктам».

Аналогично рассчитывается математическое ожидание при помощи метода row\_mean и записывается циклом for в переменную mean\_list. Оно потребуется для дальнейшего расчета дисперсии и нахождения коэффициентов компетентности экспертов.

В переменную square\_list записаны квадраты разностей значений матрицы оценок и матожидания, а в переменную sum\_square\_list, соответственно, сумма по списку square\_list.

Рассчитанные дисперсии записывают в список dispersion\_list. После чего находятся обратные дроби для дисперсий (переменная divide\_by\_one\_list) и сумма по ним (переменная sum\_diveded\_std). По итогу рассчитываются средневзвешенные ранги по продуктам путем деления дисперсии на сумму по обратным дробям (переменная weighted\_average\_ranks).

**run()**-функция, отвечающая за конечный консольный вывод для пользователя.

Помимо основного класса была реализована функция фильтрации результатов, проведенного экспертного опроса – **clear\_data**. Можно загрузить таблицу, которую формирует google docks и функция вычленит из нее три матрицы: матрицу экспертных оценок, ранговую матрицу по критерию вкус и ранговую матрицу по критерию цена-качество. Полученные матрицы записываются в три файла csv и сохраняются на компьютер пользователя. Подробнее о принципе, по которому фильтруются выгруженные данные, можно узнать в пункте 3.3.2

# **3.6. Тестирование**

Для тестирования математической обработки экспертных и ранговых оценок были использованы 6 датасетов, полученные в результате опроса экспертов. Первые 5 датасетов – выборки, которые состоят из 10 экспертов. Шестой датасет – генеральная выборка, состоящая 50 экспертов. Также был произведен опрос пяти экспертов, на основании которого был создан 1 датасет по бинарным отношениям.

Следует отметить, что математическая обработка экспертных и ранговых оценок производилась двумя алгоритмами: Excel и Python (табл. 3.1), бинарные отношения только в Excel (табл. 3.2).

Таблица 3.1 - Результаты тестирования экспертных и ранговых оценок

|  |  |  |
| --- | --- | --- |
| Датасет 1 | | |
| MS Excel | Python | |
| Математическая обработка экспертных оценок | | |
| Математическое ожидание:  Изображение выглядит как стол  Автоматически созданное описание | | Математическое ожидание: |
| Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание | | Дисперсия: |
| Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание | | Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание |
| Доверительные интервалы:  Изображение выглядит как стол  Автоматически созданное описание | | Доверительный интервал:  Изображение выглядит как стол  Автоматически созданное описание |
| Диаграмма: | | Диаграмма: |
| Математическая обработка ранговых оценок | | |
| Медиана рангов:  Изображение выглядит как стол  Автоматически созданное описание | | Медиана рангов: |
| Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание | | Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание |
| Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание | | Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание |
| Датасет 2 | | |
| MS Excel | Python | |
| Математическая обработка экспертных оценок | | |
| Математическое ожидание:  Изображение выглядит как стол  Автоматически созданное описание | | Математическое ожидание: |
| Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание | | Дисперсия: |
| Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание | | Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание |
| Доверительные интервалы:  Изображение выглядит как стол  Автоматически созданное описание | | Доверительный интервал:  Изображение выглядит как стол  Автоматически созданное описание |
| Диаграмма: | | Диаграмма: |
| Математическая обработка ранговых оценок | | |
| Медиана рангов:  Изображение выглядит как стол  Автоматически созданное описание | | Медиана рангов: |
| Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание | | Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание |
| Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание | | Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание |
| Датасет 3 | | |
| MS Excel | Python | |
| Математическая обработка экспертных оценок | | |
| Математическое ожидание:  Изображение выглядит как стол  Автоматически созданное описание | | Математическое ожидание: |
| Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание | | Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание |
| Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание | | Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание |
| Доверительные интервалы:  Изображение выглядит как стол  Автоматически созданное описание | | Доверительный интервал:  Изображение выглядит как стол  Автоматически созданное описание |
| Диаграмма: | | Диаграмма: |
| Математическая обработка ранговых оценок | | |
| Медиана рангов:  Изображение выглядит как стол  Автоматически созданное описание | | Медиана рангов: |
| Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание | | Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание |
| Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание | | Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание |
| Датасет 4 | | |
| MS Excel | Python | |
| Математическая обработка экспертных оценок | | |
| Математическое ожидание:  Изображение выглядит как стол  Автоматически созданное описание | | Математическое ожидание: |
| Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание | | Дисперсия: |
| Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание | | Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание |
| Доверительные интервалы:  Изображение выглядит как стол  Автоматически созданное описание | | Доверительный интервал:  Изображение выглядит как стол  Автоматически созданное описание |
| Диаграмма: | | Диаграмма: |
| Математическая обработка ранговых оценок | | |
| Медиана рангов:  Изображение выглядит как стол  Автоматически созданное описание | | Медиана рангов: |
| Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание | | Коэф. Компетентности экспертов:  Изображение выглядит как текст, квитанция, снимок экрана  Автоматически созданное описание |
| Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание | | Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание |
| Датасет 5 | | |
| MS Excel | Python | |
| Математическая обработка экспертных оценок | | |
| Математическое ожидание:  Изображение выглядит как стол  Автоматически созданное описание | | Математическое ожидание: |
| Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание | | Дисперсия: |
| Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание | | Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание |
| Доверительные интервалы:  Изображение выглядит как стол  Автоматически созданное описание | | Доверительный интервал:  Изображение выглядит как стол  Автоматически созданное описание |
| Диаграмма: | | Диаграмма: |
| Математическая обработка ранговых оценок | | |
| Медиана рангов:  Изображение выглядит как стол  Автоматически созданное описание | | Медиана рангов: |
| Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание | | Коэф. Компетентности экспертов:  Изображение выглядит как стол  Автоматически созданное описание |
| Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание | | Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание |
| Датасет 6 | | |
| MS Excel | Python | |
| Математическая обработка экспертных оценок | | |
| Математическое ожидание:  Изображение выглядит как стол  Автоматически созданное описание | | Математическое ожидание: |
| Дисперсия:  Изображение выглядит как стол  Автоматически созданное описание | | Дисперсия: |
| Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание | | Стандартное отклонение:  Изображение выглядит как стол  Автоматически созданное описание |
| Доверительные интервалы:  Изображение выглядит как стол  Автоматически созданное описание | | Доверительный интервал:  Изображение выглядит как стол  Автоматически созданное описание |
| Диаграмма: | | Диаграмма: |
| Математическая обработка ранговых оценок | | |
| Медиана рангов:  Изображение выглядит как стол  Автоматически созданное описание | | Медиана рангов: |
| Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание | | Средневзвешенные ранги:  Изображение выглядит как стол  Автоматически созданное описание |
|  | | |

**Можно видеть, что оба алгоритма выдают абсолютно идентичные значения.**

**По результатам опроса были сделаны следующие выводы: во всех выборках большое значение стандартного отклонения, это означает, что эксперты отвечали достаточно по-разному, выбирая разных лидеров. Например, в датасете 1 лидером является Роял де Люкс, об этом говорит и самое высокое значения математического ожидания, и наименьшие медиана рангов и средневзвешенный ранг , в датасете 2 по тому же принципу лидером стал Двойной Роял, в датасете 5 – Роял. В остальных датасетах, в том числе, в датасете 6, состоящим из 50-ти экспертов, лидером стал Макчикен Премьер. Стоит обратить внимание, что абсолютно во всех датасетах последнее место с явным отставанием занимает Филе-о-Фиш.**

**Далее рассмотрим таблицу 3.2:**

Таблица 3.2 - Математическая обработка бинарных отношений

|  |
| --- |
| **Эксперт 1:**  Изображение выглядит как стол  Автоматически созданное описание  **Эксперт 2:**  Изображение выглядит как стол  Автоматически созданное описание  **Эксперт 3:**  Изображение выглядит как стол  Автоматически созданное описание  **Эксперт 4:**  Изображение выглядит как стол  Автоматически созданное описание  **Эксперт 5:**  Изображение выглядит как стол  Автоматически созданное описание  **Матрица А:**  Изображение выглядит как стол  Автоматически созданное описание  Медиана Кемени: 4 |

**Таким образом, безоговорочным лидером является Макчикен Премьер, поскольку по его строке все единицы, далее идет Роял де Люкс с одним нулем, следом – Роял и Двойной Роял. Замыкает рейтинг, как и в прошлых оценках, Филе-о-Фиш.**

# **3.7 Заключение**

**Решение для предоставленной задачи мы реализовали с помощью двух алгоритмов. Первый алгоритм основан на встроенных функциях Excel. Его преимущества– простота использования, редактирования и ввода новых данных, автоматическое обновление формул, возможность построения бинарных решений. Из недостатков – отсутствие пользовательского интерфейса, пользователь самостоятельно должен строить графики, отфильтровывать данные опроса, редактировать формулы, что крайне время затратно, а также может повлечь за собой ряд вычислительных ошибок в следствие человеческого фактора.**

**Второй алгоритм, реализованный в Python, имеет безоговорочное преимущество перед первым – все расчеты и построения проводятся автоматически, пользователю требуется только предоставить файл для обработки. В данном алгоритме удобством является удобная загрузка данных, возможность вывести любой требуемый параметр методами реализованных классов.** Есть последовательный и грамотно прописанный выбор для пользователя, что именно он хочет делать на каждом шаге реализации кода. Фильтрация производится автоматически, пользователю не нужно вручную форматировать огромную таблицу данных по результатам опроса. Присутствуют такие функции, как сохранение матриц в файл csv, сформированных из неотфильтрованной выборки. **Программа имеет перспективы для расширения функционала, например, написание frontend.**

**Алгоритмы были проверены на пяти сформированных датасетах из генеральной выборки и на самой генеральной выборке из опроса. Значения, полученные в алгоритмах идентичны. Более привлекательным для оценки наших продуктов нам показался метод ранговых оценок, так как в методе экспертной оценки основанной на бальном оценивании достаточно большое стандартное отклонение. Лидерами в зависимости от датасетов становились разные продукты, в первом датасете Роял де Люкс, об этом говорит и самое высокое значения математического ожидания, и наименьшие медиана рангов и средневзвешенный ранг, в датасете 2 по тому же принципу лидером стал Двойной Роял, в датасете 5 – Роял. В остальных датасетах, в том числе, в датасете 6, состоящим из 50-ти экспертов, лидером стал Макчикен Премьер. В отношении этих продуктов, компании Макдональдс приняла верное решение по переносу их в новое меню. Нельзя сказать того же о бургере Филе-о-Фиш который стал самым худшим продуктам по всем методам оценок и по всем датасетам, соответственно, его можно было бы заменить на более востребованную покупателями позицию.**

# **4. Системы массового обслуживания**

# **4.1. Постановка задачи (физическая модель)**

Наш заказчик – «Успокой свои нервы», российская компания, занимающаяся продажей успокоительных настоек для нормализации работы нервной системы и налаживания режима сна.

Для производства успокоительных настоек за основу берутся всем давно известные «народные» компоненты: настойки валерианы, пустырника, боярышника, мяты и димедрол. В комбинации все эти компоненты мягко действуют на центральную нервную систему. Продукты, предоставляемые нашим заказчиком, пользуются высоким спросом, так как при современном ритме жизни с высоким уровнем стресса просто необходимо сохранять спокойствие и следить за своим физическим и эмоциональным состоянием.

Отметим, что целевой аудиторией являются молодежь, в особенности студенты, страдающие посттравматическим стрессовым расстройством после сдачи экзаменов и написания курсовых и выпускных квалификационных работ, а также пожилые люди, так как данный сегмент зачастую принимает одновременно несколько лекарственных препаратов, и как раз настойки нашего заказчика безопасны при совместном приёме с другими лекарственными средствами за исключением препаратов, угнетающих деятельность центральной нервной системы.

Итак, нашим заказчиком была предоставлена следующие официальные данные:

* в качестве временных интервалов выделены: день, вечер и час пик, временные рамки которых соответственно равны с 10 до 15, с 15:00 до 19:00, 7:00 до 10:00 и с 19:00 до 22:00. Отметим, что часы работы непосредственно связаны с концепцией нашего заказчика, поэтому они не работают по ночам, ведь настойка нормализует сон;
* у заказчика в каждом магазине имеется 3 кассы, отметим, что касс самообслуживания не предусмотрено;
* входным поток являются следующие данные: днем за 10 минут обслуживают 4 клиента, вечером – 5 клиентов, в час пик – 6;
* выходным потоком для одной кассы являются данные величины среднего времени обслуживания клиента: днем – 5 минут, вечером – 6 минут, в час пик – 4 минуты.

Наш заказчик – «Успокой свои нервы», обратился к нам для оптимизации работы его точек продаж и получить ответ на следующие вопросы:

1. с какой интенсивностью должно проходить обслуживание, чтобы минимизировать очередь?
2. какова вероятность появления очереди и ее величина?
3. сколько времени требование находится в очереди и каким образом минимизировать его задержку?
4. какова вероятность потери клиента?
5. при каких параметрах системы достигаются минимальные потери прибыли?

# **4.2. Математическая модель**

**Потоки заявок:**

Входной поток заявок, характеризуется скоростью поступления заявок в систему (интенсивностью входного потока заявок). Данный показатель показывает сколько заявок приходит в систему за определенный интервал времени. Если известен средний интервал времени между потоком заявок, то используется 4.2 формула. Если нет, то необходимо определить количество заявок, пришедшее за определенный период времени.

(4.1)

(4.2)

Где N – количество заявок, пришедшие за период – средний интервал времени между поступлением заявок.

Выходной поток заявок, характеризуется скоростью обработки заявок системой (интенсивностью выходного потока заявок от одного прибора). Данный показатель показывает сколько заявок обрабатывает один прибор за единицу времени и рассчитывается по формуле. Если известно среднее время, необходимое на обработку 1 заявки одним прибором, то используется 4.4 формула. Если нет, то необходимо определить количество заявок, обработанное за период времени некоторым количеством приборов.

(4.3)

(4.4)

**Вероятность состояний системы:**

Вероятность того, что в магазине нет покупателей и система бездействует, определяется по следующей формуле:

(4.5)

Для вероятностей наступления состояний очереди используется следующая формула.

(4.6)

В рассмотренных выше формулах – количество касс обслуживания, – максимальная длина очереди.

**Показатель загруженности системы:**

Величина является показателем загруженности системы и дает представление о том, насколько система справляется с потоком клиентов.

(4.7)

• 𝜌 ≪ 𝑛 (показатель загруженности сильно меньше количества касс) – система недогружена, что выгодно для клиента, так как нет очередей, но невыгодна для владельца, так как есть лишние кассы и большой простой по времени;

• 𝜌 < 𝑛 (показатель загруженности меньше количества касс) – система сбалансирована для клиента, так как есть приемлемые очереди и допустимый простой касс;

• 𝜌 ≤ 𝑛 – (показатель загруженности приближен к количеству касс) система сбалансирована для владельца, так как есть большие очереди клиентов, кассы заняты практически полностью;

• 𝜌 > 𝑛 (показатель загруженности больше количества касс) – система перегружена, что выгодно для владельца, так как заявок больше, чем можно об- работать, но невыгодно для клиента по причине бесконечно растущей очереди.

**Показатели для клиентов и владельцев**

Рассмотрим характеристики системы, важные для клиентов. Вероятность отказа в обслуживании показывает вероятность, что новая заявка не помещается в систему (заняты все приборы и заполнена вся очередь – последняя «правая» вероятность):

(4.8)

Вероятность встать в очередь показывает вероятность того, что новая заявка встречает в системе очередь (заняты все приборы и есть место в очереди – сумма всех вероятностей очередей):

(4.9)

Средняя длина очереди показывает среднее количество заявок, ожидающих в очередь:

(4.10)

Среднее время ожидания в очереди рассчитывается по формуле Литтла:

(4.11)

Абсолютная пропускная способность показывает скорость обслуживания заявок (сколько заявок успевает обрабатывается и получает ответ):

(4.12)

Относительная пропускная способность показывает процент обслуженных заявок (какой процент заявок не успевает обрабатывается и получает отказ):

(4.13)

Среднее количество занятых приборов:

(4.14)

Коэффициент простоя показывает процент времени простоя обслуживающих приборов:

(4.15)

# **4.3. Алгоритм решения**

## **4.3.1. Алгоритм 1. Решение с помощью Поиска решения в Excel задачи прогнозирования**

### **4.3.1.1. Описание входных данных**

В качестве входных данных мы имеем:

* временные интервалы: день, вечер и час пик, временные рамки которых соответственно равны с 10:00 до 15:00, с 15:00 до 19:00, 7:00 до 10:00 и с 19:00 до 22:00. Также было отмечено, что часы работы непосредственно связаны с концепцией нашего заказчика, поэтому они не работают по ночам, ведь настойка нормализует сон;
* наличие касс в размере 3, касс самообслуживания не предусмотрено;
* входной поток: днем за 10 минут обслуживают 4 клиента, вечером – 5 клиентов, в час пик – 6;
* выходной поток (для одной кассы) в качестве данных величины среднего времени обслуживания клиента: днем – 5 минут, вечером – 6 минут, в час пик – 4 минуты.
* максимальная длина очереди – 9 человек

### **4.3.1.2. Описание алгоритма решения**

Алгоритм принимает на вход значения входного и выходного потока с временными интервалами. Далее, высчитываем показатели загруженности системы, затем воспроизводим состояние системы, строя граф, учитывая бездействия системы и наличия очереди, затем мы подсчитываем важные характеристики для клиента (такие как: вероятность отказа в обслуживании, вероятность нахождения в очереди, средняя длина очереди, среднее время ожидания в очереди) и для владельца (такие как: абсолютная пропускная способность, относительная пропускная способность, среднее количество занятых касс и коэффициент простоя касс). Отметим, что все необходимые расчеты были выполнены с помощью встроенных функции Excel и формул, описанных в математической модели.

*Подробный пример реализации алгоритма*

Шаг 1: в соответствии с описанием задачи составляем таблицу с входными значениями. (см рис 4.1).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.1 Исходные значения

Шаг 2: как мы видим, на рисунке 4.1 присутствует показатель интенсивности потоков, которая рассчитывается следующим образом (см. рис. 4.2).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.2 – Поиск средних значений

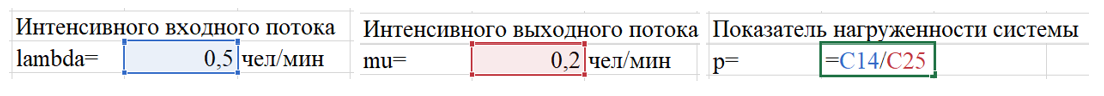
Шаг 3: далее мы вычисляем показатели нагруженности системы в зависимости от временных интервалов (см. рис. 4.3).

Рисунок 4.3 – Показатель нагруженности системы

Шаг 4: ищем показатели для построения графа состояния системы. Для этого делим работу системы на 3 стадии: бездействие, без очереди и с очередью (в зависимости от максимальной длины очереди) (см. рис. 4.4).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.4 – Поиск значений a и b

Шаг 5: находим вероятность бездействия системы для дальнейших подсчётов. (см. рис. 4.5)

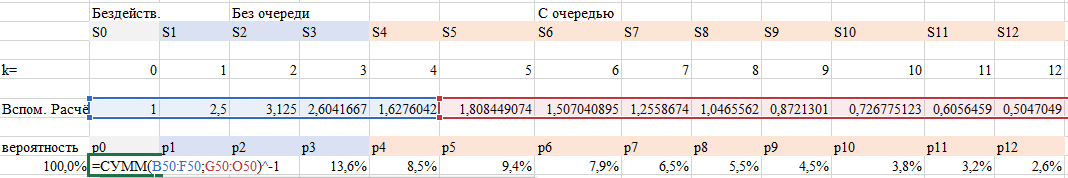


Рисунок 4.5 – Вероятность бездействия системы

Шаг 6: далее мы сможем посчитать вероятность системы с учетом наличия или отсутствия очереди (см. рис. 4.6)

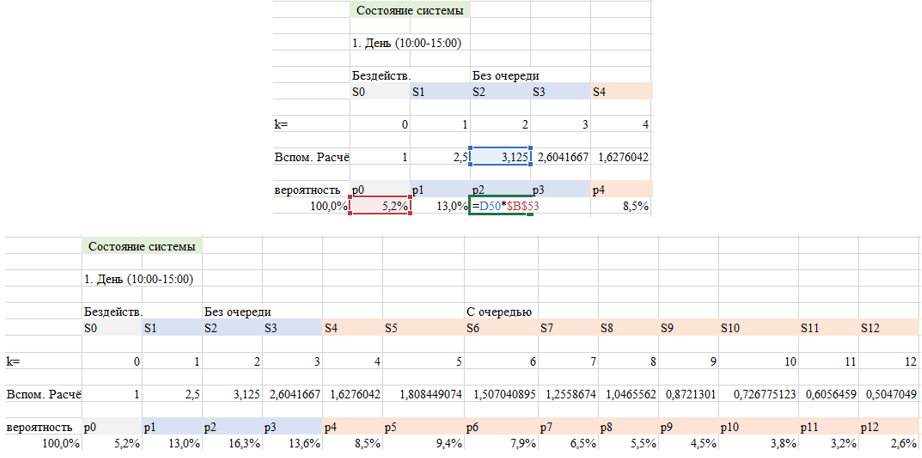


Рисунок 4.6 – Вероятности состояний системы

Как мы видим формула одинаковая для всех 12 стадий, поэтому мы просто растягиваем формулу, как это показано на рисунке 4.6.

Шаг 7: далее для большей наглядности и понимания работы системы строим гистограмму с полученной вероятностью из шага 6.

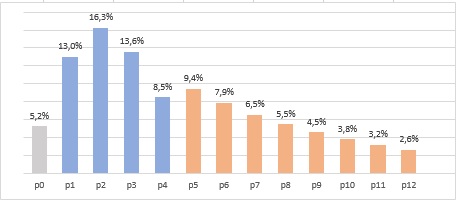


Рисунок 4.7 – Значения дисперсии и среднеквадратического отклонения

Шаг 8: далее мы считаем характеристики, которые крайне значимы для клиента.

1. Вероятность отказа в обслуживании, которая равна последней вероятности стадии состояния системы;

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4.8 – Вероятность отказа в обслуживании

1. Вероятность нахождения в очереди, суммируем вероятности нахождения в очереди за исключением последней, так как при последней вероятности клиент уходит, ничего не приобретая;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.9 – Вероятность нахождения в очереди

1. Среднюю длину очереди находим по показателю нагруженности системы, количеству касс, максимальной длине очереди и вероятности бездействия системы;



Рисунок 4.10 – Средняя длина очереди

1. Среднее время ожидания очереди рассчитываем с помощью полученных выше характеристик для клиента.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.11 – Среднее время ожидания в очереди

Шаг 9: далее мы считаем характеристики, которые имеют значимость для владельца.

1. Абсолютная пропускная способность, интенсивность входного потока умножаем на вероятность отказа в обслуживании, вычтенной из единицы;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.12 – Абсолютная пропускная способность

1. Относительную пропускную способность находим как 1 – вероятность отказа в обслуживании;

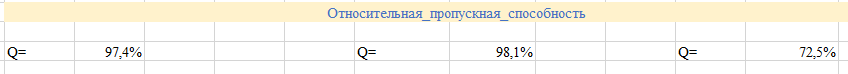


Рисунок 4.13 – Относительная пропускная способность

1. Среднее количество занятых касс находится как отношения абсолютной пропускной способности к интенсивности выходного потока;

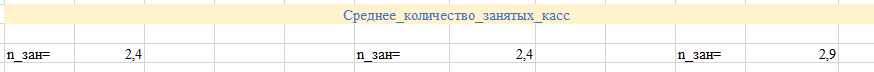


Рисунок 4.14 – Среднее количество занятых касс

1. Коэффициент простоя касс равен отношением между (1- среднее количество занятых касс) и количеством касс.

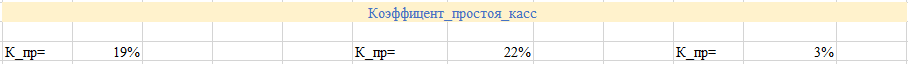


Рисунок 4.15 – Коэффициент простоя касс

Отмечу, что все перечисленные выше шаги выполняются для всех трёх временных интервалов: день, вечер и час пик.

### **4.3.1.3. Описание выходных данных**

После реализации данного алгоритма у нас есть перечень характеристик с показателями как для клиента, так и для владельца. На основании данных характеристик управленцы смогут принять решения для оптимизации работы магазина «Успокой свои нервы». Представлены следующие характеристики:

* Интенсивность входного потока
* Интенсивность выходного потока
* Показатель загруженности системы
* Вероятности нахождения системы в разные промежутки времени
* Вероятность отказа в обслуживании
* Вероятность встать в очередь
* Средняя длина очереди
* Среднее время ожидания в очереди
* Абсолютная пропускная способность
* Относительная пропускная способность
* Среднее количество занятых касс
* Коэффициент простоя

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.16 – Результаты прогнозирования

## **4.3.2. Алгоритм 2. Решение в Python**

### **4.3.2.1. Описание входных данных**

В качестве входных данных мы имеем:

* временные интервалы: день, вечер и час пик, временные рамки которых соответственно равны с 10:00 до 15:00, с 15:00 до 19:00, 7:00 до 10:00 и с 19:00 до 22:00.
* наличие касс в размере 3, касс самообслуживания не предусмотрено;
* входной поток: днем за 10 минут обслуживают 4 клиента, вечером – 5 клиентов, в час пик – 6;
* выходной поток (для одной кассы) в качестве данных величины среднего времени обслуживания клиента: днем – 5 минут, вечером – 6 минут, в час пик – 4 минуты.
* максимальная длина очереди – 9 человек

**4.3.2.2. Описание алгоритма решения**

Алгоритм решения поставленной задачи реализован при помощи функций calculation(), реализующей расчеты характеристик для владельца и для клиента, а также при помощи функции run(), реализующей консольный ввод данных для пользователя. Подробнее данные функции описаны в пункте 4.5.

*Пошаговая реализация алгоритма*

Шаг 1: Пользователь запускает окно ввода и вывода

Шаг 2: Пользователь вводит входные данные: длину очереди, длину интервала, время обработки одной заявки, количество мест обработки и максимальную длину очереди

Шаг 3: Функция calculate() рассчитывает интенсивность входного потока, интенсивность выходного потока, записывает их в соответствующие переменные.

Шаг 4: Функция рассчитывает показатель загруженности системы, записывает их в переменную.

Шаг 5: Функция рассчитывает вероятности нахождения системы в разные промежутки времени и записывает их в таблицу dataframe.

Шаг 6: Функция рассчитывает характеристики для клиента, записывает их в соответствующую переменную.

Шаг 7: calculate() рассчитывает характеристики для владельца, записывает в переменную.

Шаг 8: Все переменные выводятся на экран благодаря функции run().

### **4.3.2.3. Описание выходных данных**

Выходными данными алгоритма являются:

* Интенсивность входного потока
* Интенсивность выходного потока
* Показатель загруженности системы
* Вероятности нахождения системы в разные промежутки времени
* Вероятность отказа в обслуживании
* Вероятность встать в очередь
* Средняя длина очереди
* Среднее время ожидания в очереди
* Абсолютная пропускная способность
* Относительная пропускная способность
* Среднее количество занятых касс
* Коэффициент простоя

# **4.4. Варианты использования алгоритмов**

## **4.4.1. Варианты использования алгоритма 1 – поиск решений Excel**

Шаг 1: пользователь заполняет данные в таблице «Входной поток заявок» для расчёта интенсивности входного потока. Таблица разделена на 3 колонки для 3 ситуаций: день, вечер и час-пик. В каждой колонке пользователь заполняет время (t) и количество человек (N), которое успевают обслужить все кассы за это время.

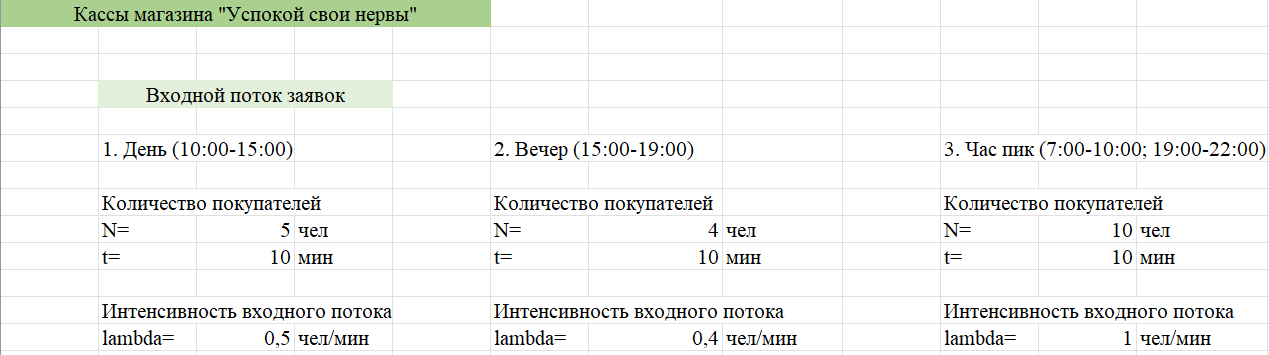


Рисунок 4.17 – Входной поток заявок

Шаг 2: пользователь заполняет величины среднего времени обслуживания клиента днём, вечером и в час-пик в таблице «Выходной поток заявок» для расчёта интенсивности выходного потока.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4.18 – Выходной поток

Шаг 3: в таблице «Загруженность системы» пользователь указывает количество касс и максимальную длину в очереди. Чем ближе значения показателя загруженности и количества касс, тем лучше.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.19 – Загруженность системы

Шаг 4: автоматически считается состояние системы для 3 ситуаций: день, вечер и час-пик. В таблицах для каждой из 3 ситуаций указана вероятность появления в очереди от 0 до 12 человек, при этом диапазон от 0 до 3 означает, что очереди нет, а 12, что клиенту отказывают в обслуживании. Для наглядности данных рассчитываются диаграммы (см. рис 4.20). Пользователь оценивает состояние системы исходя из данных.

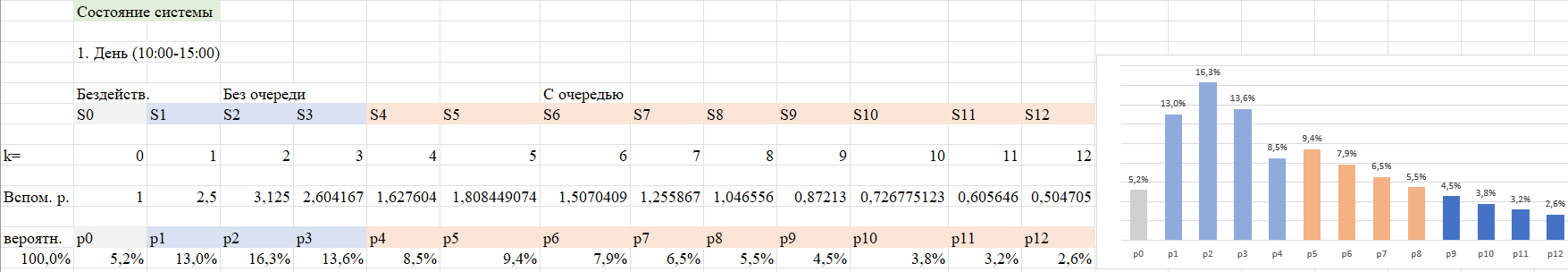


Рисунок 4.20 – Состояние системы днём

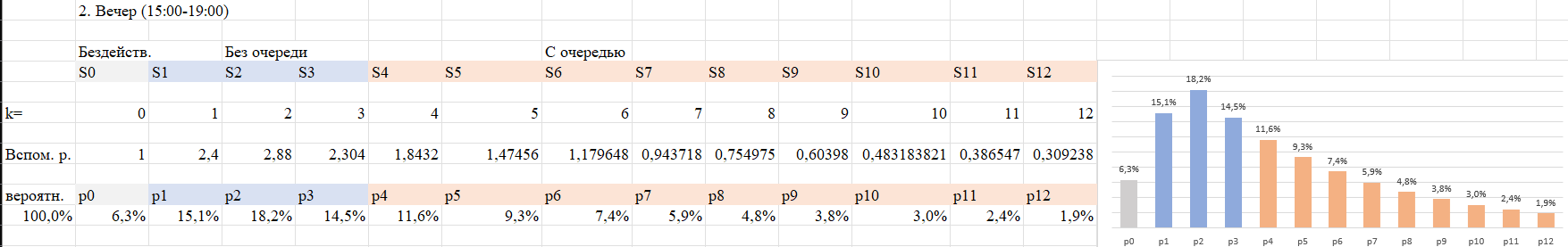


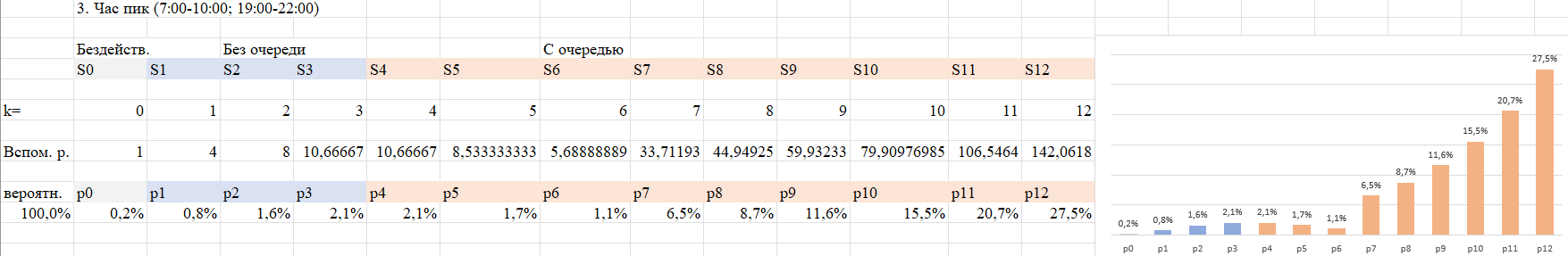
Рисунок 4.21 – Состояние системы вечером

Рисунок 4.22 – Состояние системы в час-пик

Шаг 5: автоматически считаются характеристики для клиента: вероятность отказа в обслуживании, вероятность встать в очередь, средняя длина очереди и среднее время ожидания в очереди. Пользователь оценивает критерии, которые имеют значение для посетителей (см. рис 4.23).

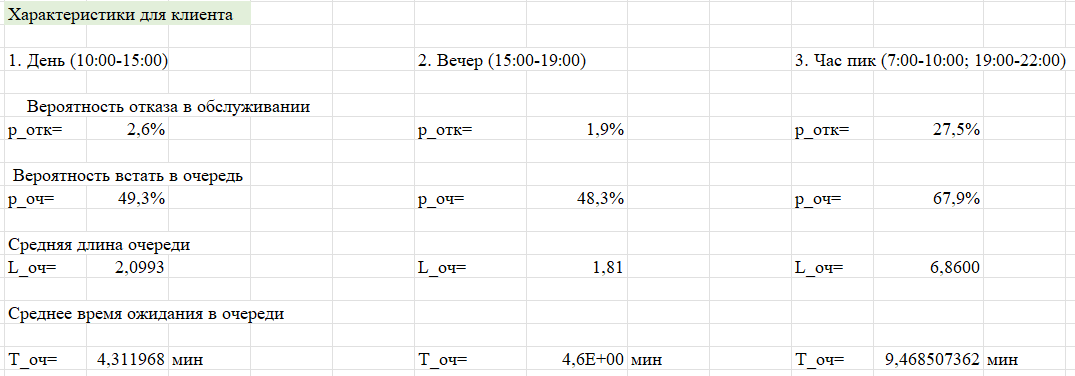


Рисунок 4.23 – Характеристики для клиента

Шаг 6: автоматически считаются характеристики для владельца: абсолютная пропускная способность, относительная пропускная способность, среднее количество занятых касс и коэффициент простоя. Пользователь оценивает критерии, которые имеют значение для посетителей (см. рис 4.24).

Изображение выглядит как текст, небо, внутренний

Автоматически созданное описание

Рисунок 4.24 – Характеристики для владельца

4.4.2. Варианты использования алгоритма 2 – Python

Пользователь может ввести данные путём ручного ввода данных. Перед выполнением действий для ввода данных необходимо запустить все окна кода с библиотеками для корректной работы программы.

Шаг 1: пользователь запускает окно с библиотеками, окно с кодом и окно запуска кода.

Шаг 2: пользователь вводит в строке ввода данных длину очереди (см. рис. 4.25) то есть количество человек, которое успевают обслужить все кассы за время, которое необходимо указать далее.

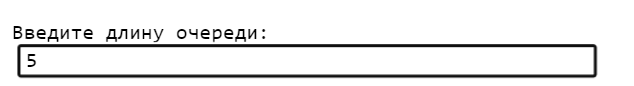


Рисунок 4.25 – Ввод длины очереди

Шаг 3: пользователь вводит в строке ввода данных длину интервала (см. рис. 4.26) то есть время, за которое все кассы успевают обслужить указанное ранее количество человек (данные необходимы для определения интенсивности входного потока).

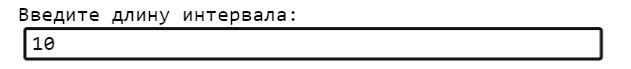


Рисунок 4.26 – Ввод длины интервала

Шаг 4: пользователь вводит в строке ввода данных время обработки одной заявки (см. рис. 4.27) то есть время обслуживания одного человека одной кассой (данные необходимы для определения интенсивности выходного потока и показателя загруженности системы).



Рисунок 4.27 – Ввод времени обработки одной заявки

Шаг 5: пользователь вводит в строке ввода данных количество мест обработки (см. рис. 4.28) то есть количество касс обслуживания.

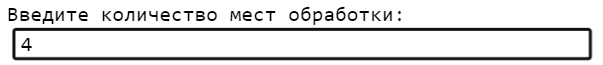


Рисунок 4.28 – Ввод количества мест обработки

Шаг 6: пользователь вводит в строке ввода данных максимальную длину очереди (см. рис. 4.29).

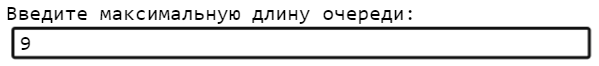


Рисунок 4.29 – Ввод максимальной длины очереди

Шаг 7: автоматически будут высчитаны значения интенсивность входного потока интенсивность выходного потока, показатель загруженности системы, вероятности нахождения системы в разные промежутки времени, вероятность отказа в обслуживании, вероятность встать в очередь, средняя длина очереди, среднее время ожидания в очереди, абсолютная пропускная способность, относительная пропускная способность, среднее количество занятых касс и коэффициент простоя (см. рис. 4.30).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.30 – Выходные данные

Шаг 8: пользователь имеет возможность продолжить или завершить работу (см. рис. 4.31). Для этого в окне ввода пользователь должен ввести да, если намерен продолжать или нет, если хочет закончить.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.31 – Выбор действия

# **4.5. Архитектура решения алгоритма на языке программирования Python**

Для реализации всех имеющихся алгоритмов решения мы использовали следующие библиотеки:

* Pandas — библиотека для обработки и анализа структурированных данных;
* NumPy — это библиотека языка Python, добавляющая поддержку больших многомерных массивов (включая матрицы), поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами;
* Math – библиотека для применения различных математических функций;
* Matplotlib — библиотека для визуализации данных;

Все вычисления по математической модели производятся при помощи следующего объекта:

**Calculate()**— основная функция, реализующая расчеты характеристик для клиента и для владельца

Входные данные функции:

* длина очереди
* длина интервала
* время обработки одной заявки
* количество мест обработки
* максимальная длина очереди

Переменные lambda, mu, p содержат интенсивность входного и выходного потока, показатель загруженности системы. Далее при помощи циклов for рассчитываются вероятности нахождения системы в определенном состоянии и записываются в таблицу pandas dataframe.

После чего строится диаграмма с помощью библиотеки matplotlib по списку вероятностей нахождения системы в определенном состоянии (список p\_list). В заключении рассчитываются оставшиеся характеристики для клиента и для владельца, после чего выводятся с помощью функции run().

# **4.6. Тестирование**

Для тестирования были использованы 3 датасета, тестирование производилось двумя алгоритмами MS Excel и Python.

Результаты представлены в таблице 4.1:

Таблица 4.1 - Результаты тестирования

|  |  |  |
| --- | --- | --- |
| Входные данные | MS Excel | Python |
| Датасет 1 | | |
| Первый период | | |
| Входной поток:  N= 5  t = 10  Выходной поток:  t = 5  Нагруженность системы:  n = 3  m= 9 | Интенсивность входного потока:  lambda = 0,5  Интенсивность выходного потока:  nu = 0,2  Показатель нагруженности системы:  p = 2,5  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 2,6%  Р оч = 50,7%  L оч = 2,0415  Т оч = 4,19  Характеристика для владельца:  А = 0,487  Q = 97,4%  nзан = 2,4  Кпр = 19% | Интенсивность входного потока:  lambda = 0,5  Интенсивность выходного потока:  nu = 0,2  Показатель нагруженности системы:  p = 2,5  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 2,6%  Р оч = 50,7%  L оч = 2,0415  Т оч = 4,19  Характеристика для владельца:  А = 0,487  Q = 97,4%  nзан = 2,4  Кпр = 19% |
| Второй период | | |
| Входной поток:  N= 4  t = 10  Выходной поток:  t = 6  Нагруженность системы:  n = 3  m= 9 | Интенсивность входного потока:  lambda = 0,4  Интенсивность выходного потока:  nu = 0,166  Показатель нагруженности системы:  p = 2,4  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 1,9%  Р оч = 46,3%  L оч = 1,74  Т оч = 4,424  Характеристика для владельца:  А = 0,393  Q = 98,1%  nзан = 2,4  Кпр = 21% | Интенсивность входного потока:  lambda = 0,4  Интенсивность выходного потока:  nu = 0,166  Показатель нагруженности системы:  p = 2,4  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 1,9%  Р оч = 46,3%  L оч = 1,74  Т оч = 4,424  Характеристика для владельца:  А = 0,393  Q = 98,1%  nзан = 2,4  Кпр = 21% |
| Третий период | | |
| N= 10  t = 10  Выходной поток:  t = 4  Нагруженность системы:  n = 3  m= 9 | Интенсивность входного потока:  lambda = 1  Интенсивность выходного потока:  nu = 0,25  Показатель нагруженности системы:  p = 4  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 25,9%  Р оч = 69,8%  L оч = 6,4406  Т оч = 8,6877  Характеристика для владельца:  А = 0,74  Q = 74,1%  nзан = 3  Кпр = 1% | Интенсивность входного потока:  lambda = 1  Интенсивность выходного потока:  nu = 0,25  Показатель нагруженности системы:  p = 4  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 25,9%  Р оч = 69,8%  L оч = 6,4406  Т оч = 8,6877  Характеристика для владельца:  А = 0,74  Q = 74,1%  nзан = 3  Кпр = 1% |
| Датасет 2 | | |
| Первый период | | |
| Входной поток:  N= 5  t = 10  Выходной поток:  t = 3  Нагруженность системы:  n = 4  m= 9 | Интенсивность входного потока:  lambda = 0,5  Интенсивность выходного потока:  mu = 0,333  Показатель нагруженности системы:  p = 1,5  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 0,0%  Р оч = 2,8%  L оч = 0,0447  Т оч = 0,089  Характеристика для владельца:  А = 0,499  Q = 100%  nзан = 1,5  Кпр = 63% | Интенсивность входного потока:  lambda = 0,5  Интенсивность выходного потока:  mu = 0,333  Показатель нагруженности системы:  p = 1,5  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 0,0%  Р оч = 2,8%  L оч = 0,0447  Т оч = 0,089  Характеристика для владельца:  А = 0,499  Q = 100%  nзан = 1,5  Кпр = 63% |
| Второй период | | |
| Входной поток:  N= 3  t = 10  Выходной поток:  t = 1  Нагруженность системы:  n = 1  m= 9 | Интенсивность входного потока:  lambda = 0,3  Интенсивность выходного потока:  mu = 1  Показатель нагруженности системы:  p = 0,3  Гистограмма состояния системы:  Изображение выглядит как стол  Автоматически созданное описание  Характеристика для клиента:  P отк = 0,0%  Р оч = 9%  L оч = 0,13  Т оч = 0,428  Характеристика для владельца:  А = 0,299  Q = 100%  nзан = 0,3  Кпр = 70% | Интенсивность входного потока:  lambda = 0,3  Интенсивность выходного потока:  mu = 1  Показатель нагруженности системы:  p = 0,3  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 0,0%  Р оч = 9%  L оч = 0,13  Т оч = 4,3E-01  Характеристика для владельца:  А = 0,299  Q = 100%  nзан = 0,3  Кпр = 70% |
| Третий период | | |
| N= 10  t = 10  Выходной поток:  t = 5  Нагруженность системы:  n = 6  m= 9 | Интенсивность входного потока:  lambda = 1  Интенсивность выходного потока:  mu = 0,2  Показатель нагруженности системы:  p = 5  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 2,1%  Р оч = 41,5%  L оч = 1,673  Т оч = 1,709  Характеристика для владельца:  А = 0,98  Q = 97,9%  nзан = 4,9  Кпр = 18% | Интенсивность входного потока:  lambda = 1  Интенсивность выходного потока:  mu = 0,2  Показатель нагруженности системы:  p = 5  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 2,1%  Р оч = 41,5%  L оч = 1,673  Т оч = 1,709  Характеристика для владельца:  А = 0,98  Q = 97,9%  nзан = 4,9  Кпр = 18% |
| Датасет 3 | | |
| Первый период | | |
| Входной поток:  N= 10  t = 60  Выходной поток:  t = 20  Нагруженность системы:  n = 4  m= 4 | Интенсивность входного потока:  lambda = 0,166  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 3,33  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 7%  Р оч = 31%  L оч = 0,877  Т оч = 5,671  Характеристика для владельца:  А = 0,155  Q = 93%  nзан = 3,09  Кпр = 23% | Интенсивность входного потока:  lambda = 0,166  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 3,33  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 7%  Р оч = 31%  L оч = 0,877  Т оч = 5,671  Характеристика для владельца:  А = 0,155  Q = 93%  nзан = 3,09 |
| Второй период | | |
| Входной поток:  N= 5  t = 60  Выходной поток:  t = 20  Нагруженность системы:  n = 2  m= 4 | Интенсивность входного потока:  lambda = 0,83  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 1,666  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 8%  Р оч = 38%  L оч = 1,07  Т оч = 14,06  Характеристика для владельца:  А = 0,076  Q = 91%  nзан = 1,52  Кпр = 24% | Интенсивность входного потока:  lambda = 0,83  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 1,666  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 8%  Р оч = 38%  L оч = 1,07  Т оч = 14,06  Характеристика для владельца:  А = 0,076  Q = 91%  nзан = 1,52  Кпр = 24% |
| Третий период | | |
| Входной поток:  N= 12  t = 60  Выходной поток:  t = 20  Нагруженность системы:  n = 5  m= 4 | Интенсивность входного потока:  lambda = 0,2  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 4  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 6%  Р оч = 26%  L оч = 0,711  Т оч = 3,766  Характеристика для владельца:  А = 0,189  Q = 94%  nзан = 3,78  Кпр = 24% | Интенсивность входного потока:  lambda = 0,2  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 4  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 6%  Р оч = 26%  L оч = 0,711  Т оч = 3,766  Характеристика для владельца:  А = 0,189  Q = 94%  nзан = 3,78  Кпр = 24% |
| Четвертый период | | |
| Входной поток:  N= 2  t = 60  Выходной поток:  t = 18  Нагруженность системы:  n = 1  m= 4 | Интенсивность входного потока:  lambda = 0,033  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 0.6  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 3,26%  Р оч = 29,61%  L оч = 0,625  Т оч = 19,41  Характеристика для владельца:  А = 0,32  Q = 96%  nзан = 0,58  Кпр = 42% | Интенсивность входного потока:  lambda = 0,033  Интенсивность выходного потока:  mu = 0,05  Показатель нагруженности системы:  p = 0.6  Гистограмма состояния системы:    Характеристика для клиента:  P отк = 3,26%  Р оч = 29,61%  L оч = 0,625  Т оч = 19,41  Характеристика для владельца:  А = 0,32  Q = 96%  nзан = 0,58  Кпр = 42% |

**Оба алгоритма выдают абсолютно идентичные значения.** Проанализировав таблицу, можно сделать следующие выводы: в отличие от Ecxel, в алгоритме Python автоматически определяется количество показателей для построения графа состояния системы и относительная пропускная способность считается не в процентах, а в десятичных дробях.

# **4.7. Заключение**

**Решение для предоставленной задачи мы реализовали с помощью двух алгоритмов. Первый алгоритм основан на встроенных функциях Excel. Его преимущества– простота использования, редактирования и ввода новых данных. Из недостатков – отсутствие пользовательского интерфейса, пользователь самостоятельно должен строить графики, редактировать формулы, а так как в данном тз они достаточно объемные, это может повлечь за собой ряд ошибок.**

**Второй алгоритм, реализованный в Python имеет преимущество перед первым – все расчеты и построения проводятся автоматически, пользователю требуется передать только входные данные. В данном алгоритме удобством является удобная загрузка данных, наличие пользовательского выбора, отсутствие ручного форматирования диаграмм и таблиц,** автоматически определяется количество показателей для построения графа состояния системы. **Программа имеет перспективы для расширения функционала, например, можно добавить опции выгрузки и загрузки данных в csv, написать интерфейс, улучшить наглядность предоставляемой пользователю информации.**

**Алгоритмы были проверены на трех сформированных датасетах. Значения, полученные в алгоритмах идентичны. В результате анализа магазина (по датасету 1) было обнаружено, что оптимальнее всего в утренние часы и днем использовать три кассы, так как показатель загруженности приближен к количеству касс и система сбалансирована для владельца, так как есть большие очереди клиентов, кассы заняты практически полностью. Для вечернего часа-пик можно дооснастить магазин еще одной кассой, так как система перегружена, что выгодно для владельца, так как заявок больше, чем можно обработать, но невыгодно для клиента по причине бесконечно растущей очереди.**