

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Системное программирование и компьютерные технологии

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка модуля для реализации аффинитивного анализа в среде Matlab.

Студент

Р. Р. Зайнетдинов

(И.О. Фамилия)

(личная подпись)

Руководитель

В. С. Климов

(И.О. Фамилия)

(личная подпись)

Консультанты

Н. В. Яценко

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 20 _____ г.

Тольятти 2018

АННОТАЦИЯ

Тема: «Разработка модуля для реализации аффинитивного анализа в среде Matlab».

В данной бакалаврской работе анализируются реализации алгоритмов искусственного интеллекта в Matlab. В исследовании рассматривается технология аффинитивного анализа и их реализации в среде Matlab.

Структура выпускной квалификационной работы состоит из введения, трех глав, заключения и списка литературы.

Во введении описывается актуальность темы исследования, заключающаяся в отсутствии реализации алгоритма аффинитивного анализа в Matlab. Формулируются цели и задачи исследования.

В первой главе проводится обзор предметной области. Рассматривается реализация алгоритмов искусственного интеллекта в Matlab.

Во второй главе описываются задачи аффинитивного анализа и их решение. Приводится подробный пример работы алгоритма, реализующую задачу аффинитивного анализа.

В третьей главе приведена разработка модуля для реализации алгоритма аффинитивного анализа в среде Matlab и представлен пример работы программы.

Работа состоит из 45 страниц, включая 19 рисунков, 9 таблиц, списка литературы из 20 источников и 2 приложений.

В заключении представлены основные выводы по проделанной работе.

ABSTRACT

The title of the graduation work is the Development of Toolbox for Implementation of Affinity Analysis in the Environment of Matlab.

In this graduation work algorithms of artificial intelligence in the Matlab are analyzes. In the work the technology of the affinity analysis and and their implementation in Matlab. are considered.

The structure of the graduation work consists of an introduction, three parts, a conclusion and a list of references.

In the introduction describes the relevance of the research topic, which is the lack of implementation of the algorithm for affinity analysis in Matlab. The goals and objectives of the study are formulated.

The first part provides an overview of the subject area. The implementation of artificial intelligence algorithms in Matlab is considered.

The second part describes the affinity analysis tasks and their solution. A detailed example of the algorithm that implements the task of affinity analysis.

In the third part the development of Toolbox for the implementation of the algorithm for affinity analysis in Matlab and an example of the program is presented.

The work consists of 45 pages, including 19 figures, 9 tables, the list of literature consists of 20 sources and 2 appendixes.

In conclusion, the main conclusions on the work done are presented..

Оглавление

ВВЕДЕНИЕ	5
-----------------------	----------

ГЛАВА 1 ИСПОЛЬЗОВАНИЕ МАТЛАВ ПРИ МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ	6
1.1 Обзор предметной области	6
1.2 Анализ тулбоксов математической среды разработки Matlab	12
ГЛАВА 2 МАТЕМАТИЧЕСКИЙ АППАРАТ ЗАДАЧИ АФФИНТИВНОГО АНАЛИЗА.....	20
2.1. Описание задачи аффинитивного анализа	20
2.2. Поиск ассоциативных правил	20
2.3. Значимость ассоциативных правил.....	22
2.4 Алгоритмы поиска ассоциативных правил	23
ГЛАВА 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ.....	30
3.1. Программная реализация модуля.....	30
3.2. Тестирование разработанной программы	31
ЗАКЛЮЧЕНИЕ.....	36
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	37
ПРИЛОЖЕНИЕ А - Программный код алгоритма.....	40
ПРИЛОЖЕНИЕ Б - Интерфейс программы.....	43

ВВЕДЕНИЕ

Сегодня существует множество программных продуктов для математического моделирования технических систем. Среди бурно развивающихся систем компьютерной математики, которые ориентированы на численные расчеты, наиболее распространенной системой является Matlab (The MathWorks, США). Из-за большого количества модулей расширения Matlab, поставляемых с программой, эта среда также является самой функциональной из систем компьютерной математики. Однако алгоритмы машинного обучения реализованы лишь частично, например, нет официальной поддержки алгоритмов аффинитивного анализа. В связи с этим актуальность бакалаврской работы заключается в разработке модуля для реализации аффинитивного анализа в математической среде разработки Matlab.

Matlab — это платформа для научных вычислений и высокоуровневого программирования, использующая интерактивную среду, что позволяет выполнять сложные вычислительные задачи более эффективно, чем при использовании языков программирования, таких как C, C++, C# и Java.

Объектом исследования бакалаврской работы является интеллектуальный анализ данных.

Предметом исследования бакалаврской работы является модуль, в котором реализован аффинитивный анализ.

Целью бакалаврской работы является расширение функциональных возможностей Matlab за счёт разработки модуля для реализации аффинитивного анализа.

Задачи, решаемые в ходе работы:

- изучение системы компьютерной математики Matlab;
- исследование модулей для реализации алгоритмов искусственного интеллекта, реализованных в Matlab;
- изучение математического аппарата аффинитивного анализа;
- проектирование и реализация модуля для аффинитивного анализа в Matlab.

ГЛАВА 1 ИСПОЛЬЗОВАНИЕ MATLAB ПРИ МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ

1.1 Обзор предметной области

Среди универсальных программных продуктов для моделирования технических систем, наиболее известным является Matlab. Это одна из самых популярных платформ компьютерной математики, используемых в настоящее время в науке и технике. Данный программный продукт предназначен для численного и аналитического решения различных математических задач и исследования динамических систем, включающих как дискретные, так и непрерывные. Главная особенность Matlab – простота языка программирования, благодаря которому любой пользователь сможет начать работать в этой среде. В отличие от других подобных программ, Matlab получил широкое распространение в инженерной практике.

Matlab используется в математическом анализе, алгебре, статистике, эконометрике, тестировании и систем измерений, финансовом моделировании, вычислительной биологии и др.

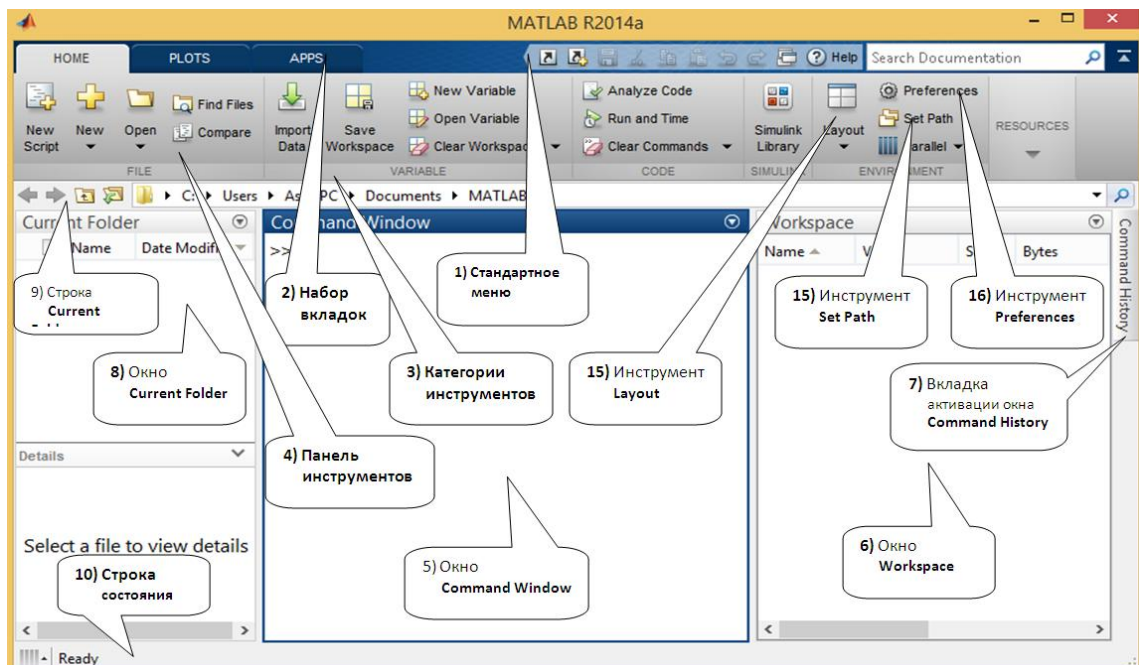


Рисунок 1.1 – Рабочая среда Matlab

Математический пакет Matlab позволяет проводить инженерные расчеты и используется при проведении научных исследований. Используемые в Matlab библиотеки для расчетов оптимизированы для работы с данными в векторном и матричном виде.

Matlab позволяет работать с:

- системами линейных уравнений;
- регрессией и интерполяцией;
- дифференцированием и интегрированием;
- обыкновенными дифференциальными уравнениями и др.

С помощью дополнительных расширений у Matlab имеется функция импорта данных из файлов, (показано на рисунке 1.2).

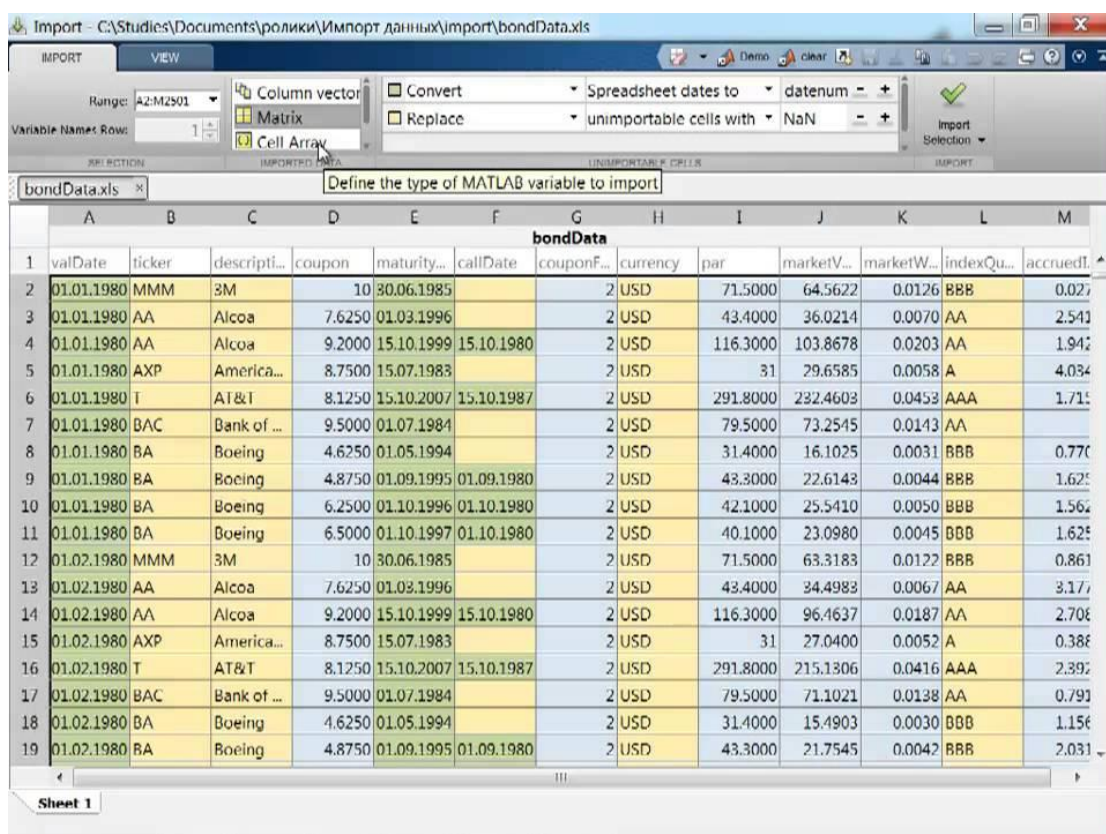


Рисунок 1.2 – Импорт данных в Matlab

Также есть возможность считывание информации из баз данных, внешних устройств и научных приборов (осциллографов). Благодаря Matlab можно анализировать данные, находить и тестировать гипотезы, строить математические модели. Для этого в среде Matlab реализованы следующие функции: сглаживания, фильтрации, свертки и преобразования Фурье.

Математический пакет Matlab предусматривает возможность построения двумерных и трехмерных диаграмм, помимо этого, доступен список функций для визуализации данных. Для каждого типа диаграмм в Matlab есть возможность просмотра примеров. Любой график может быть сохранен как векторных (SVG, PDF), так и растровых (JPG, PNG) форматах изображениях. Кроме того, математический пакет Matlab предоставляет возможность генерировать полные отчеты по расчетам, с сохранением в популярные форматы, такие как DOC, HTML и PDF. Также в отчет можно включить программный код с комментариями, вычислениями и графиками.

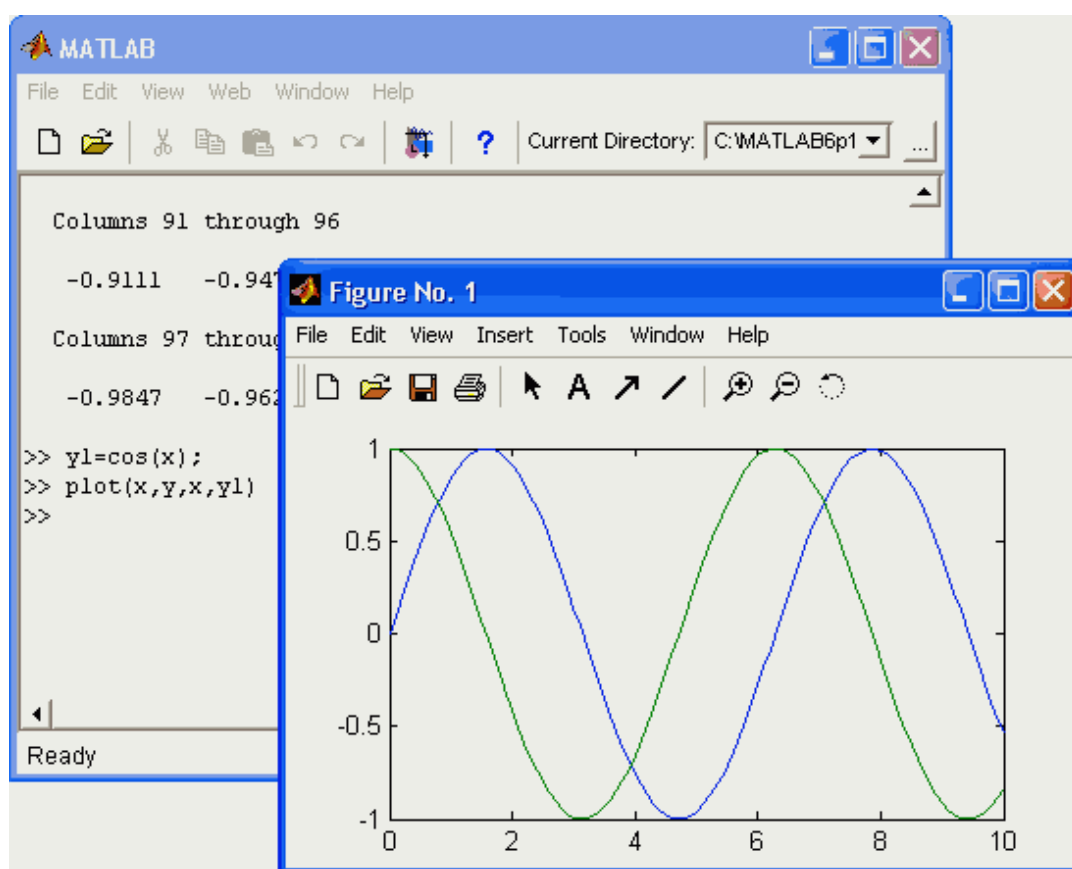


Рисунок 1.3 – Вычисления в Matlab, графическая функция fplot

Математическая среда разработки Matlab имеет высокоуровневый язык программирования и различные инструменты для отладки кода программы. Поддержка операций над матрицами и векторами – уникальность этого языка программирования и также не требуется объявлять переменные и определять их типы вычислений при программировании. В то же время встроенный язык Matlab поддерживает объектно-ориентированные методы программирования,

обработку ошибок и управление потоком. Существует также возможность определения пользовательских типов данных, используя различные структуры.

Математическая среда разработки Matlab состоит из следующих компонентов:

1. Окно командной строки (на англ. Command Window);
2. Редактор кода (на англ. Editor);
3. Анализатор кода (на англ. Code Analyzer);
4. Профилировщик (на англ. Profiler).

Окно командной строки предназначено для просмотра результатов выполнения команд и выполнения интерактивных действий с информацией. Редактор кода содержит несколько приложений, которые дают вам возможность редактировать файлы, написанные на встроенном языке программирования Matlab. Анализатор кода отображает возможные ошибки и проблемы программного кода, а также возможности улучшения кода с помощью сообщений. Профилировщик используется для отслеживания быстродействия кода и его оптимизация.

Matlab имеет поддержку интеграции с самыми распространенными языками программирования. Имеется возможность вставки программного кода на языках программирования таких как C, C++, C#, Java и др., а также, благодаря библиотеке Engine, есть возможность вызова программного кода Matlab, написанные на различных языках программирования.

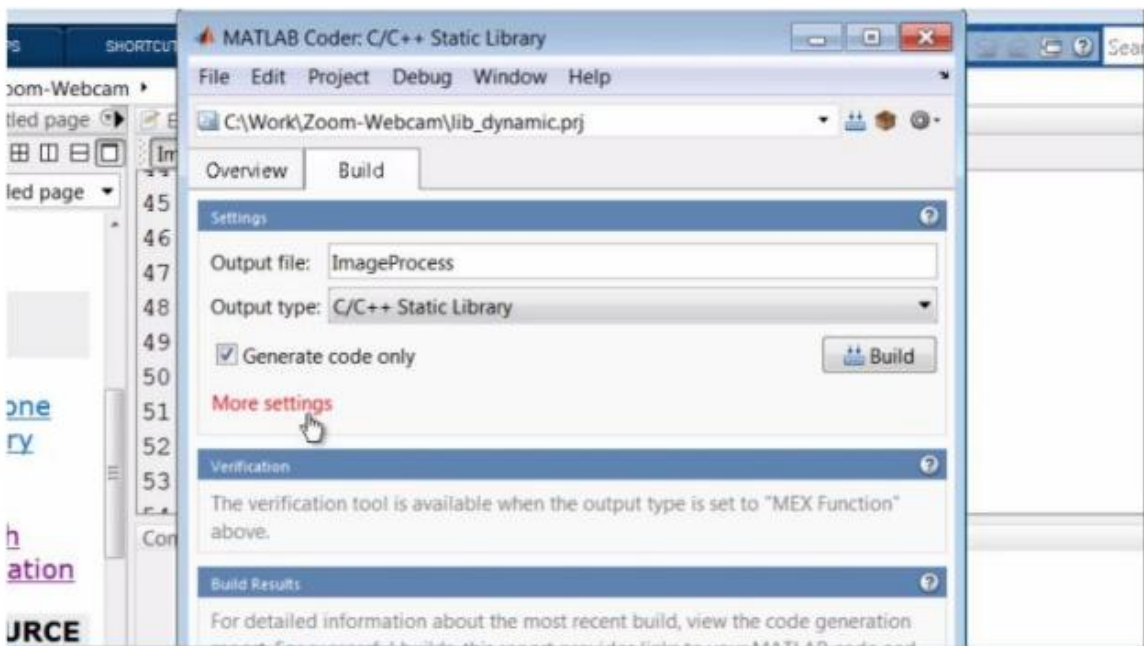


Рисунок 1.4 – Создание автономной библиотеки

```

60
61     emxInit_real_T(&ImageHor, 2);
62
63     /* First compute the horizontal zoom */
64     /* 'ImageProcess:28' ImageHor = zeros(Min, Wout); */
65     ImageIn_idx_0 = ImageIn->size[0];
66     i0 = ImageHor->size[0] * ImageHor->size[1];
67     ImageHor->size[0] = ImageIn_idx_0;
68     emxEnsureCapacity((emxArray_common *)ImageHor, i0, (int32_T)sizeof(real_T));
69     i0 = ImageHor->size[0] * ImageHor->size[1];
70     ImageHor->size[1] = (int32_T)Wout;
71     emxEnsureCapacity((emxArray_common *)ImageHor, i0, (int32_T)sizeof(real_T));
72     loop_ub = ImageIn->size[0] * (int32_T)Wout;
73     for (i0 = 0; i0 < loop_ub; i0++) {
74         ImageHor->data[i0] = 0.0;
75     }
76
77     /* 'ImageProcess:30' for col = 1:Wout */
78     for (ImageIn_idx_0 = 0; ImageIn_idx_0 < (int32_T)Wout; ImageIn_idx_0++) {
79         /* 'ImageProcess:32' cout = col-1; */
80         /* 'ImageProcess:33' cin = cout/Factor+1; */
81         eps = ((1.0 + (real_T)ImageIn_idx_0) - 1.0) / Factor;
82
83         /* 'ImageProcess:35' before = floor(cin); */

```

Рисунок 1.5 – Интеграция программного кода

Математическая среда Matlab использует jit-компилятор и оптимизированную библиотеку матричных вычислений. Благодаря встроенной многопоточности в Matlab, реализуется автоматическое разделение вычислений на различные потоки. Это позволяет добиться преимущества в скорости в системах с несколькими ядрами. В математическом моделировании для параллельной вычислительной обработки информации имеется возможность

использования мощности графических процессоров, а также компьютерных кластеров.

Математическая среда Matlab предоставляет удобные средства для разработки и развертывания приложений. Благодаря этому, на сервере среды разработки Matlab, именуемую Mathwork, реализуются публикации разработанного алгоритма или приложения, который имеет общий доступ другим людям, работающим в Matlab. Не менее важным является то, что в Matlab существует свой компилятор под названием Matlab Compiler. Он позволяет поделится приложением Matlab, а также позволяет запускать приложения без установленной среды Matlab.

В Matlab имеются специальные средства разработки графического интерфейса – Graphical User Interface Development Environment (GUIDE).

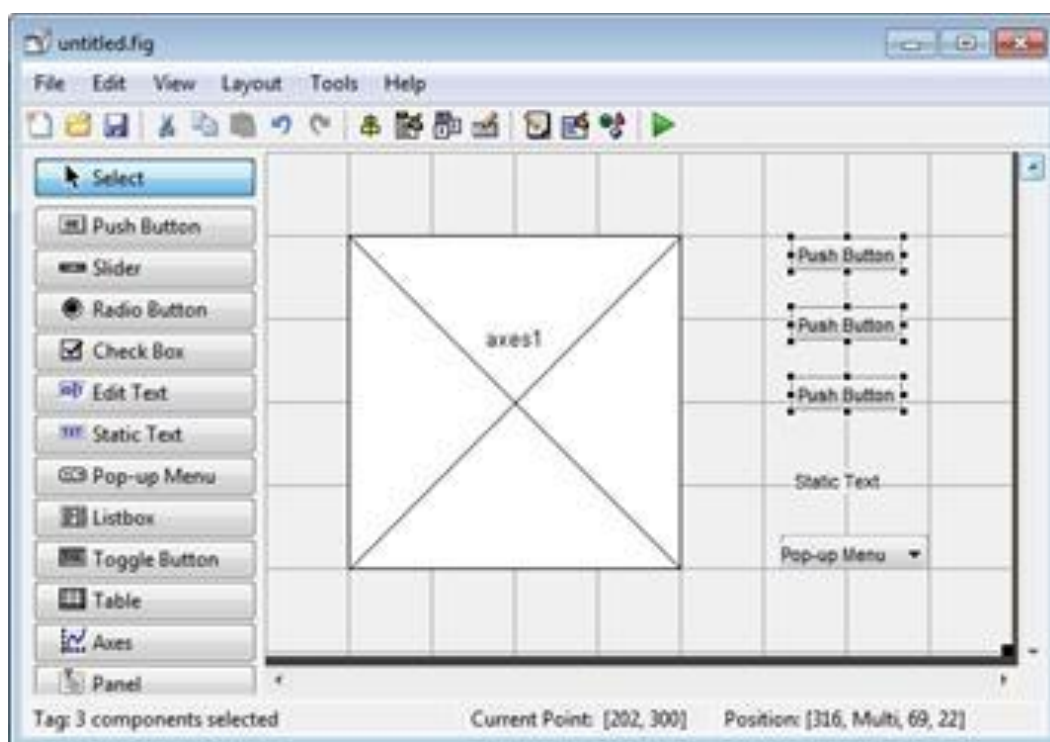


Рисунок 1.6 – Графический интерфейс GUIDE

GUIDE – это интерактивная графическая система для проектирования и создания графических пользовательских интерфейсов. Она обеспечивает гибкость системного конструктора и позволяет проектировать графический интерфейс приложения. GUIDE поддерживает некоторые элементы управления, такие как: списки, поля со списком, кнопки, надписи и др.

1.2 Анализ модулей математической среды разработки Matlab

Рассмотрим тулбоксы Matlab, в которых реализуются алгоритмы искусственного интеллекта:

- Optimization Toolbox.
- Fuzzy Logic Toolbox.
- Neural Network Toolbox.
- Statistics and Machine Learning Toolbox.

В Neural Network Toolbox имеются функции и приложения для моделирования сложных нелинейных систем. Такие нелинейные системы обычно сложно описываются уравнениями. Также этот модуль имеет поддержку методов обучения с учителем, нейронными сетями прямого распространения, с радиально-базисными и динамическими сетями. С этим модулем есть возможность создания, обучения и моделирование нейронных сетей.

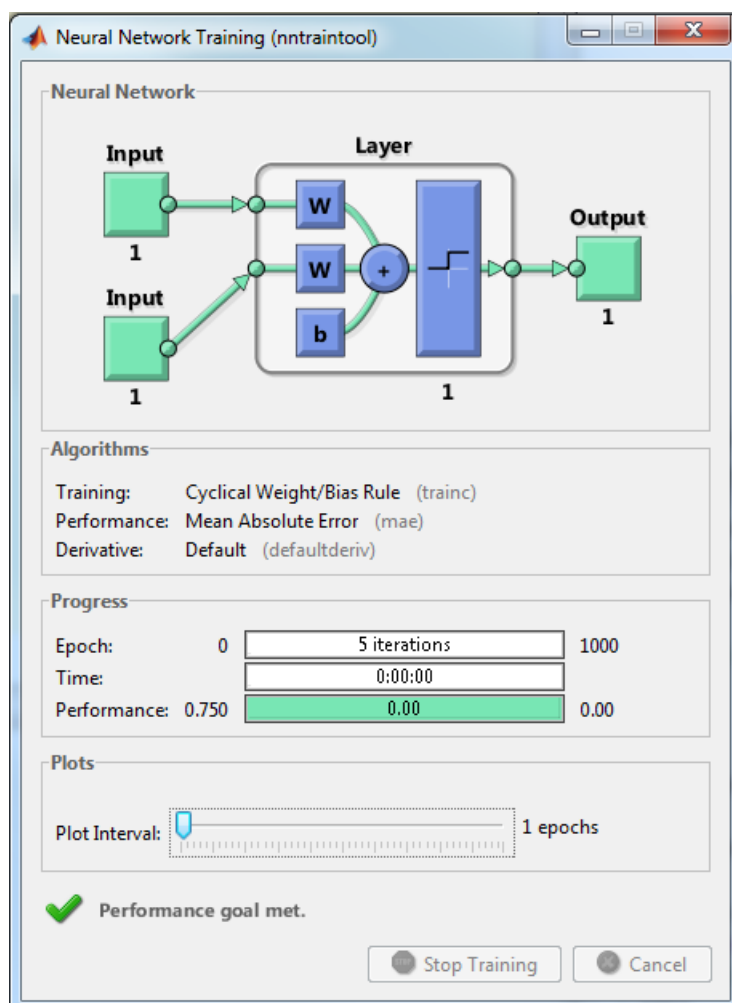


Рисунок 1.7 – Окно модуля с демонстрацией обучения

Neural Network Toolbox так же можно использовать для таких задач, как кластеризация, прогноз временных рядов, аппроксимация данных, распознавание образов, моделирование динамических систем и их управление.

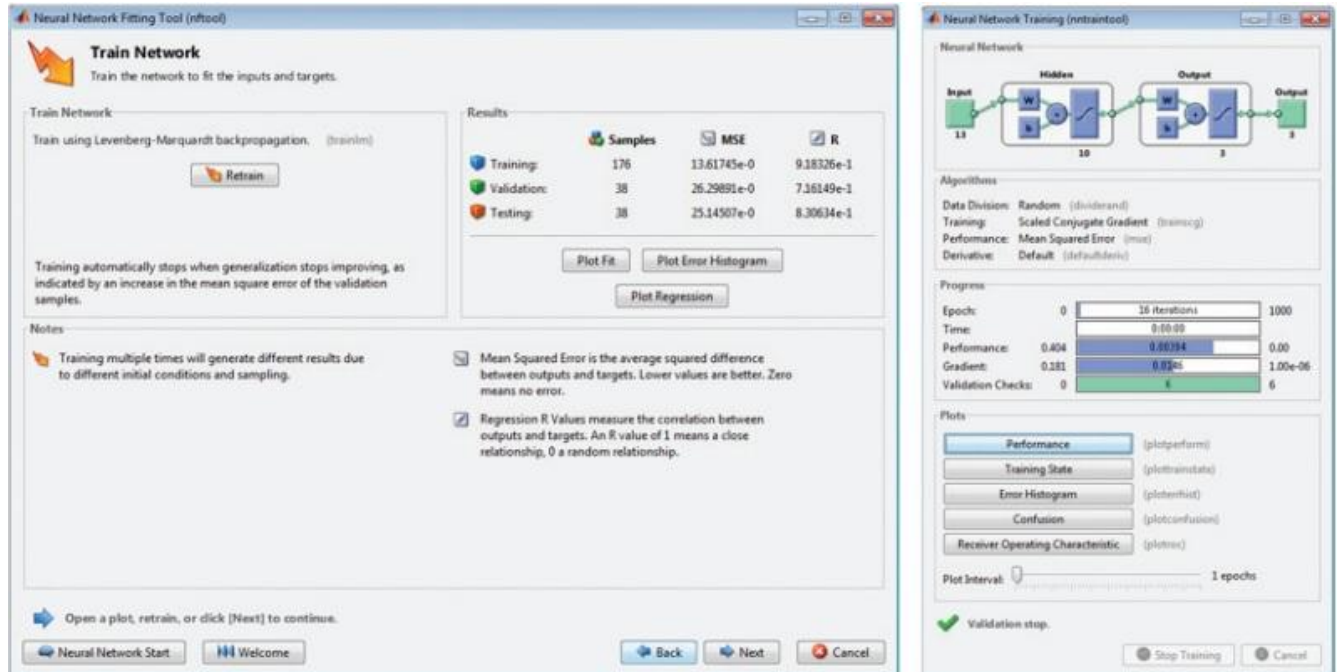


Рисунок 1.8 – Интерфейсы модуля для работы с нейронными сетями

Optimization Toolbox – это библиотека с функциями, расширяющие возможности математической среды Matlab по численным вычислениям и предназначенные для решения задач оптимизации и систем нелинейных уравнений. Optimization Toolbox включает в себя наиболее часто используемые методы минимизации и максимизации. Инструмент включает в себя алгоритмы для стандартных и многомерных задач, что позволяет решать задачи, используя их структуру или разреженность.

Optimization Toolbox можно использовать для таких задач, как решение нелинейных уравнений, аппроксимация, линейное и нелинейное программирование, задачи многокритериальной оптимизации и др.

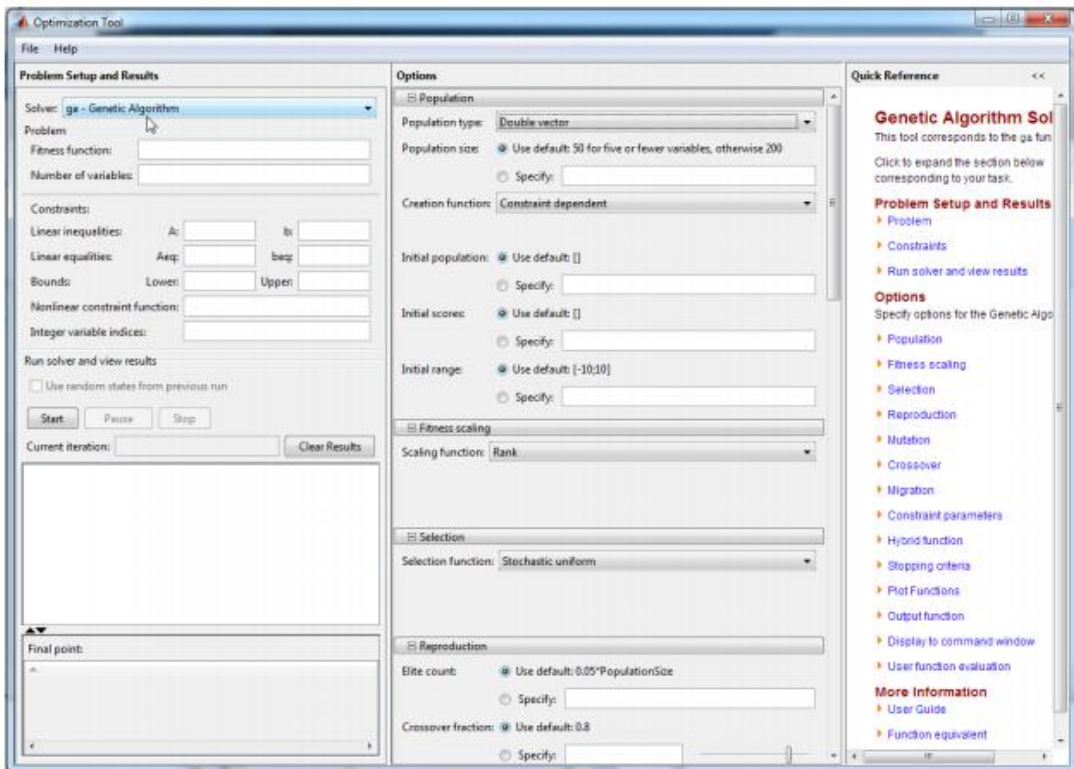


Рисунок 1.9 – Окно модуля Optimization Toolbox

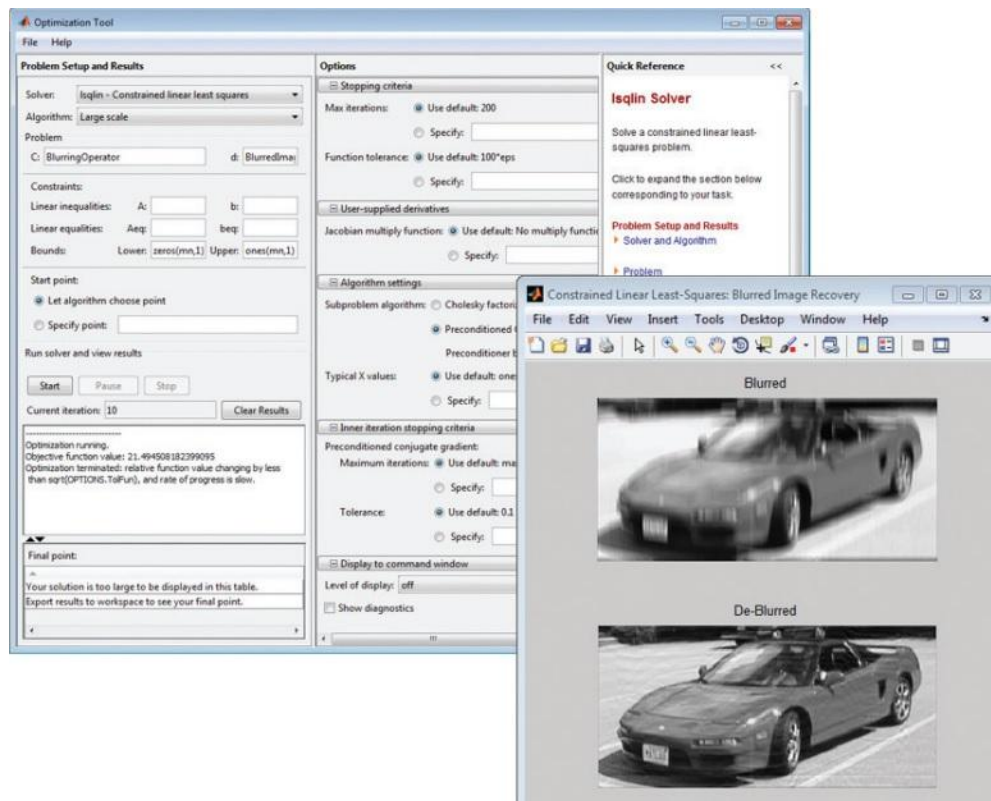


Рисунок 1.10 – Оптимизация изображения с помощью Optimization Toolbox

Доступ к функциям и параметрам инструмента можно получить с помощью командной строки:

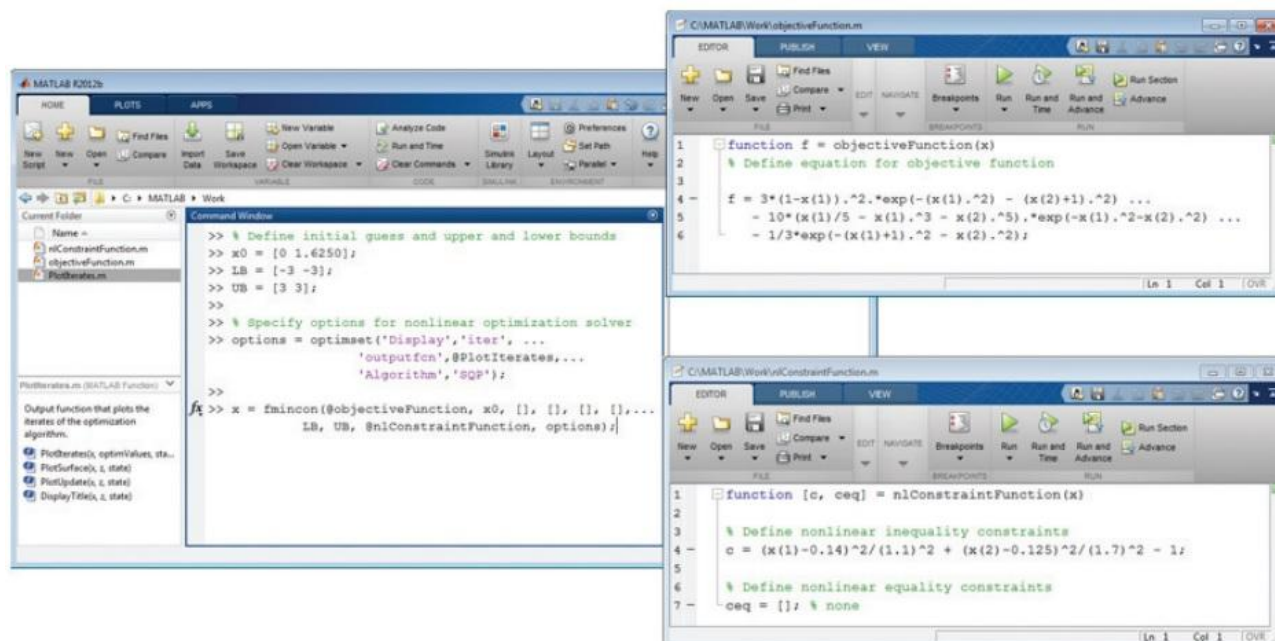


Рисунок 1.11 – Запуск инструмента в командной строке, которая вызывает файл с описанием целевой функции и файл с описанием ограничений

Модуль Statistics and Machine Learning Toolbox представляет собой инструмент, содержащий алгоритмы для организации и приложения для анализа и моделирования данных. Так же он содержит алгоритмы, которые производят анализ многомерных данных. Он предоставляет использование статистических графиков для исследовательского анализа данных, генерацию случайных чисел для метода Монте-Карло и выполнения проверок гипотез. Алгоритмы регрессии и классификации позволяют делать выводы из данных и строить прогнозные модели.

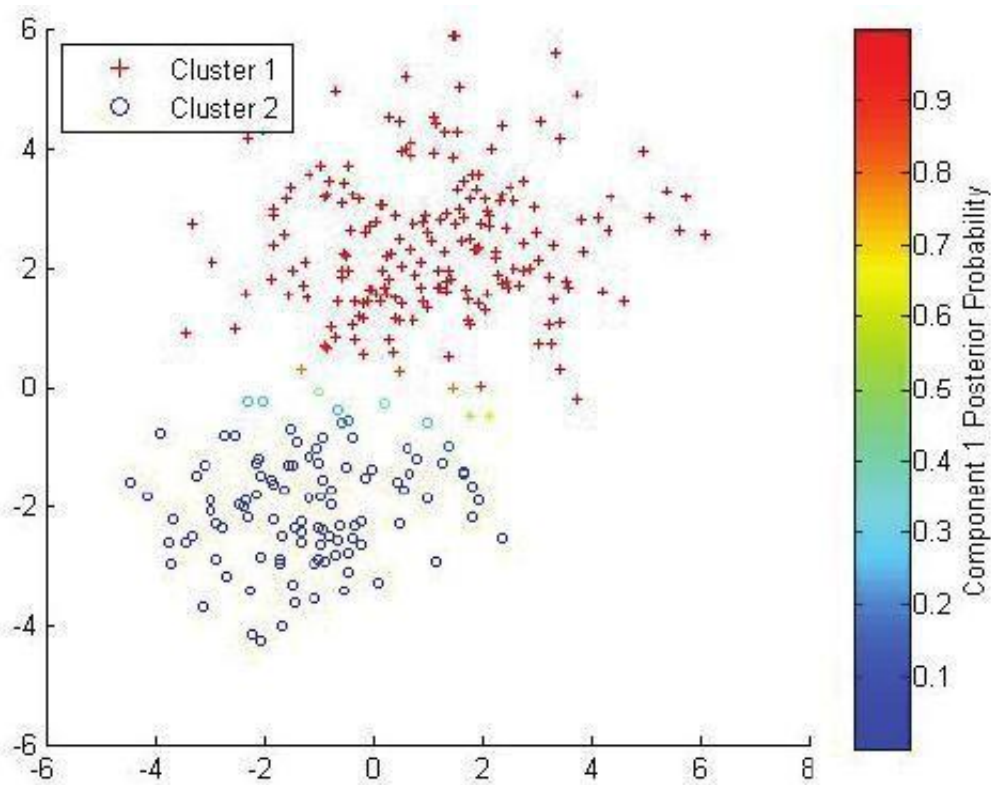


Рисунок 1.12 – Кластеризация с использованием модуля *Statistic and machine learning toolbox*

Для анализа данных *Statistics and Machine Learning Toolbox* обеспечивает ряд функций:

- Пошаговая регрессия.
- Регуляризация и другие методы сокращения размерности, которые позволяют идентифицировать переменные или функции.
- Обеспечивает контролируемые и неконтролируемые алгоритмы машинного обучения, включая метод опорных векторов.
- Реализация расчета распределения вероятностей, связки и смеси нормальных распределений.
- Иерархическая кластеризация Гауссовой модели и скрытых Марковских моделей.

Fuzzy Logic Toolbox предоставляет функции *Matlab*, графические модули и блоки *Simulink* для анализа, проектирования и моделирования различных систем, основанные на нечеткой логике. Модуль помогает конструировать

системы нечеткого вывода благодаря функциям многих распространенных методов, в том числе нечеткой кластеризации.

Панель элементов позволяет моделировать сложные системные поведения благодаря простым логическим правилам, и реализовывать их в системе нечеткого вывода. Так же есть возможность использования блоков нечеткого вывода в Simulink и моделирование нечетких систем комплексных моделей всей динамической системы.

Основные функции модуля Fuzzy Logic Toolbox:

- Создание и построение нечетких систем;
- Определение связей функций;
- Регламентирование нечетких правил;
- Оценка и визуализация нечетких систем и др.

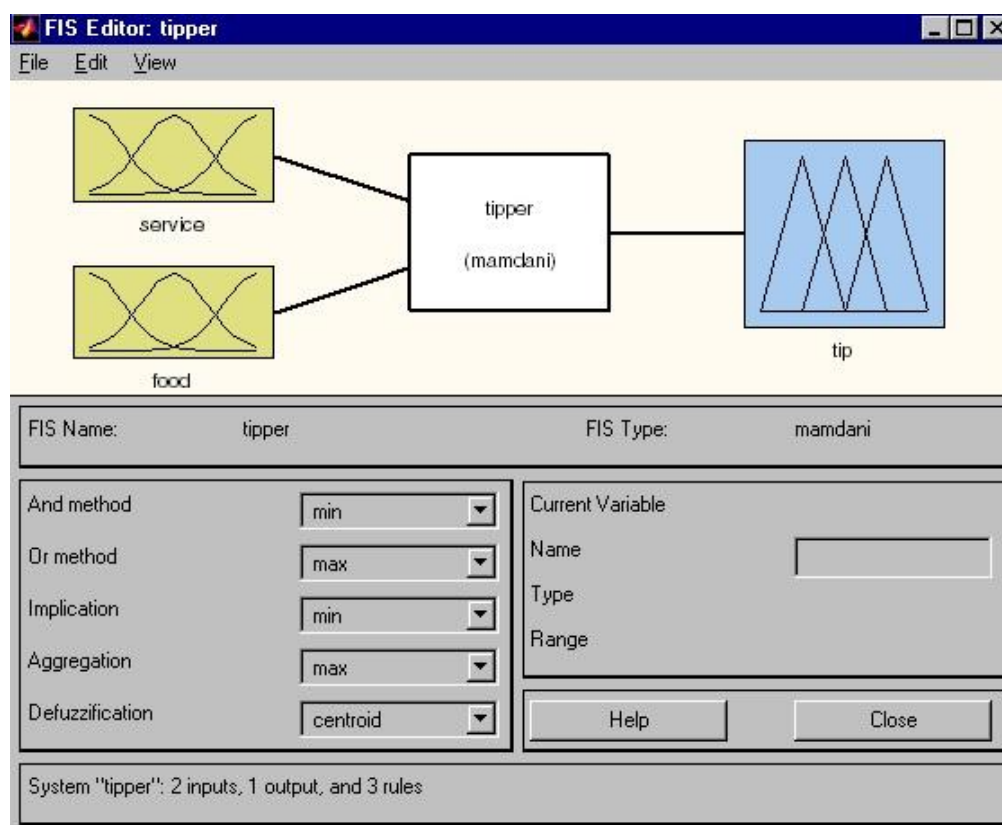


Рисунок 1.13 – Окно графического интерфейса редактора FIS

Наиболее важной частью прикладных программ является приложение Simulink. В настоящее время система Simulink является одной из самых используемых инструментов для численных расчетов и используется в различных областях знаний. Приложение Simulink – это библиотека блоков для многодоменного моделирования и модельного проектирования. Оно поддерживает системное проектирование, автоматическую генерацию кода моделирование и непрерывный мониторинг, в том числе проверку встроенных систем. Simulink предоставляет собственный графический редактор, настраиваемые библиотеки блоков и решатели для моделирования и симуляции. Оно интегрировано с Matlab, что позволяет включать его алгоритмы в модели и экспортировать результаты моделирования для дальнейшего анализа.

Основные особенности приложения Simulink:

- интерактивная графическая среда для построения блок-диаграмм;
- удобные средства построения многоуровневых иерархических многокомпонентных моделей;
- отображение области информации для просмотра результатов моделирования;
- импорт алгоритмов Matlab в модели и др.

Рассмотрим в таблице 1 результаты анализа модулей в Matlab и реализации алгоритмов искусственного интеллекта.

Таблица 1 – Реализации алгоритмов искусственного интеллекта в модулях Matlab

Алгоритм	Решаемые задачи	Toolbox Matlab
нейронные сети	классификация и регрессионный анализ	Neural Network Toolbox
нечеткие системы	создания гибридных интеллектуальных систем.	Fuzzy Logic Toolbox
k-средних	кластеризация	Statistic and machine learning toolbox
c-средних	кластеризация	Fuzzy Logic Toolbox
k-ближайших соседей	классификация и	Statistic and machine

	регрессия	learning toolbox
--	-----------	------------------

Продолжение таблицы 1.

Алгоритм	Решаемые задачи	Toolbox Matlab
генетические алгоритмы	оптимизация функций и задачи компоновки	Optimization toolbox
метод опорных векторов	бинарная классификация и задачи мультиклассификации	Statistic and machine learning toolbox
дерево классификации и регрессии	классификация и регрессия	Statistic and machine learning toolbox
apriori	аффинитивный анализ	отсутствует реализация

Matlab — это платформа для научных вычислений и высокоуровневого программирования, использующая интерактивную среду, что позволяет выполнять сложные вычислительные задачи более эффективно, а также в Matlab встроены дополнительные расширения, благодаря которым реализуются специальные задачи в различных областях применения.

Тем не менее, представленные в таблице модули, реализующие алгоритмы машинного обучения, лишь частично реализованы в Matlab. Исходя из этого можно выявить цель исследования – расширение функциональных возможностей Matlab за счёт разработки модуля для реализации аффинитивного анализа.

Поставленная цель решается путем следующих задач:

- изучение математического аппарата аффинитивного анализа;
- проектирование и реализация модуля для аффинитивного анализа в Matlab.

ГЛАВА 2 МАТЕМАТИЧЕСКИЙ АППАРАТ ЗАДАЧИ АФФИНТИВНОГО АНАЛИЗА

2.1. Описание задач аффинитивного анализа

Аффинитивный анализ (affinity analysis) — один из распространенных методов Data Mining. Само название в переводе означает близость, сходство. Суть этого метода — исследование взаимной связи между событиями, которые происходят одновременно, а целью является обнаружение ассоциаций между разными событиями. Эти описывающие связь между событиями правила называются ассоциативными правилами.

Примеры ассоциативных правил:

- отбор различных комбинаций продуктов, которые часто или никогда не покупаются совместно;
- отбор доли клиентов, которые положительно или отрицательно относятся к обслуживанию в заведении;
- возможность побочных эффектов при совместном приеме различного сочетания лекарств.

Главное понятие в аффинитивном анализе — транзакция. Транзакция — это некоторый набор событий, происходящих совместно. Например, транзакцией является одновременная покупка каких-либо продуктов покупателем в магазине. Аффинитивный анализ позволяет ответить на вопрос: является ли выборка какого-либо одного товара причиной выборки другого подходящего товара? Такие связи основывают ассоциативные правила.

2.2. Поиск ассоциативных правил

Ассоциативные правила количественно описывают связь между наборами предметов, указанных в условии и следствии. Эта связь характеризуется двумя метриками — поддержкой и достоверностью.

В процессе поиска ассоциативных правил необходимо найти все ассоциации, поддержка и достоверность которых выше заданных значений

Для дальнейшего исследования рассмотрим таблицу 2, в которой представлены 5 транзакций и 5 элементов.

Таблица 2 – Пример базы данных транзакций

№ транзакции	Молоко	Хлеб	Масло	Пиво	Ткань
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

Множество $I = \{\text{молоко, хлеб, масло, пиво, ткань}\}$ – множество элементарных событий. В таблице 2 элементы, в записи со значением 1 означает наличие элемента в транзакции, а значение 0 - отсутствие элемента в транзакции. Например, существует такое правило в магазине, что, покупая масло и хлеб, покупатели также берут молоко. Иными словами, $\{\text{масло, хлеб}\} \rightarrow \{\text{молоко}\}$.

Поддержка (Support) — это показатель частоты появления набора элементов в наборе данных. Поддержка ассоциативного правила Q — это число транзакций, которые содержат как условие, так и следствие. Например, для ассоциации $X \rightarrow Y$ можно записать в виде:

$$Q X \rightarrow Y = \frac{P_{X,Y}}{N} \quad (1),$$

где $P_{X,Y}$ – количество транзакций, содержащих X и Y , N – общее количество транзакций. Например, набор элементов $X = \{\text{пиво, ткань}\}$ имеет поддержку $\frac{1}{5} = 0,2$, т.к. транзакция происходит в 20% всех транзакций (1 из 5 транзакций).

Достоверность (Confidence) — это показатель того, как часто ассоциативное правило оказывается верным. Величина доверия $X \rightarrow Y$ в отношении совокупности сделок T это есть доля транзакций, содержащая X и Y . Достоверность рассчитывается по формуле:

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

$$U X \rightarrow Y = \frac{P_{X \cap Y}}{P_X} \quad (2),$$

где $P(X \cap Y)$ – количество транзакций, содержащие условия и следствия X и Y ,
 $P(X)$ – количество транзакций, содержащих только условие X . Например,
 $\{\text{масло, хлеб}\} \rightarrow \{\text{молоко}\}$ имеет достоверность $\frac{0,2}{0,2} = 1$. Это означает, что
 покупатель покупает масло и хлеб, он так же берет молоко.

Если же поддержка и достоверность правила достаточно высоки, можно утверждать, что любая транзакция, включающая условия, будет содержать и следствия.

2.3. Значимость ассоциативных правил

Алгоритмы поиска ассоциативных правил обнаруживают те ассоциации, которые удовлетворяют ограничениям на поддержку и достоверность, заданных пользователем. Часто требуется рассматривать десятки и сотни тысяч ассоциаций, что делает невозможным обработку такого большого объема данных методом перебора всех возможных правил. Очевидно, что желательно сократить количество правил так, чтобы анализировать только самые значимые правила.

Можно выделить объективные и субъективные показатели важности правил. Такие меры, как поддержка и достоверность являются объективными, а субъективными показателями являются лифт и левередж:

Лифт (Lift) — это отношение частоты появления условия в транзакциях к частоте появления условия в целом. Значение лифта больше, чем единица, показывает, что условие чаще появляется в транзакциях, содержащих следствие, чем в остальных. Лифт рассчитывается по формуле:

$$\text{lift } X \Rightarrow Y = \frac{\text{supp } XY}{\text{supp } (X) \cdot \text{supp } (Y)} \quad (5)$$

Например, в таблице 2 лифт $\{\text{молоко, хлеб}\} \Rightarrow \{\text{масло}\} = \frac{0,2}{0,4 \cdot 0,4} = 1,25$.
 Если значение лифта равняется 1, это значит, что события в условии и следствии являются независимыми друг от друга.

Другой мерой значимости является левередж. Левередж (Leverage) — это разность между наблюдаемой частотой, при которой условие и следствие

появляются совместно, и также появления произведения частот условия и следствия по отдельности.

$$\text{Leverage}(X \rightarrow Y) = \text{supp}(X \rightarrow Y) - \text{supp}(X) \cdot \text{supp}(Y) \quad (6)$$

Такие меры, как лифт и левередж, могут использоваться для дальнейшего ограничения набора рассматриваемых ассоциаций путем установления порога значимости, ниже которого ассоциативные правила отбрасываются.

2.4 Алгоритмы поиска ассоциативных правил

Первый алгоритм поиска ассоциативных правил AIS, был разработан в 1993 году. С того момента началось исследование ассоциативных правил и уже в середине 90-х годов прошлого века начали разрабатываться несколько новых алгоритмов. В алгоритме AIS элементы предметного набора создаются и подсчитываются во время сканирования базы данных. Каждая транзакция проверяется на наличие нечастых наборов, выявленных при предыдущем проходе. Соответственно, новые наборы формируются путем расширения имеющихся наборов. Этот алгоритм является устаревшим и самым неэффективным из современных, поскольку генерирует и учитывает слишком много нечастых предметных наборов.

Следующий рассматриваемый алгоритм называется SETM. Для вычисления часто встречающихся предметных наборов, с использованием языка SQL, был разработан этот алгоритм. Как и алгоритм AIS, SETM также формирует элементы во время сканирования базы данных, основываясь на преобразованиях. Алгоритм SETM отделяет формирование элемента от их подсчета, с целью использования стандартной операции объединения языка SQL. Общий недостаток этих двух алгоритмов является излишнее генерирование и подсчет ненужных элементов, которые в результате оказываются нечастыми. В конце концов для улучшения поиска ассоциативных правил был разработан алгоритм Apriori.

Простейший алгоритм поиска ассоциативных правил учитывает все возможные сочетания условий и следствий, оценивает их поддержку и достоверность, а затем исключает все ассоциации, не соответствующие

указанным ограничениям. Число возможных ассоциаций с увеличением числа субъектов возрастает в геометрической прогрессии.

Если в базе данных транзакций имеется k объектов и все ассоциации являются бинарными, необходимо проанализировать $k \cdot 2^{k-1}$ ассоциаций. Поскольку транзакции в базе данных обычно содержат тысячи элементов, вычислительные затраты на поиск ассоциативных правил огромны. По этой причине в процессе генерации ассоциативных правил широко используются методики, позволяющие сократить количество ассоциаций, которые необходимо анализировать. Одной из наиболее распространенных является алгоритм поиска ассоциативных правил Apriori.

Алгоритм Apriori является классическим алгоритмом поиска правил ассоциации интеллектуального анализа набора данных. Для нахождения ассоциаций используется основная идея алгоритма, заключающаяся в использовании итеративного подхода. В основе алгоритма Apriori лежит понятие частого набора, который также можно назвать частым предметным набором, часто встречающимся в множестве. Под частотой понимается простое количество транзакций, в которых содержится данный предметный набор. Тогда частыми наборами будут те из них, которые встречаются чаще, чем в заданном числе транзакций.

Методика поиска ассоциативных правил с использованием частых наборов состоит из трех шагов:

1. Создание предметных наборов.
2. Отсечение лишних элементов. На этом шаге происходит выборка частых наборов по правилу Apriori - любые подмножества частого множества также должны быть частыми. Если предметный набор имеет значение поддержки меньше заданного, то такой предметный набор будет удален.
3. Создание ассоциативных правил. Накопление количества ассоциаций появляется в базе данных.

Чтобы сократить пространство поиска ассоциативных правил, алгоритм Аргіогі использует свойство антимонотонности. Данное полезное свойство позволяет значительно уменьшить пространство поиска ассоциативных правил. Суть свойства заключается в том, что поддержка любого предметного набора элементов не может превышать заданной минимальной поддержки любого из его подмножеств.

Рассмотрим пример работы алгоритма Аргіогі при заданном минимальном значении поддержки равному 3. В таблице 3 приведен вариант базы данных транзакций.

Таблица 3 – База данных 6 транзакций

№ транзакции	Транзакция
1	а, б, в
2	б, г
3	б, а, г, в
4	д, г
5	а, б, в, г
6	е

Сначала происходит формирование одноэлементных кандидатов, как показано в таблице 4.

Таблица 4 – Формирование 1-элементных кандидатов

Предметный набор	Значение поддержки
а	3
б	4
в	4
г	3
д	1
е	1

Затем предметные наборы со значением поддержки меньше 3 отсекаются. В данном случае это наборы д и е, которые имеют поддержку равную 1. Оставшиеся наборы, показанные в таблице 4, считаются часто встречающимися одноэлементными предметными наборами: а, б, в, г. После проделанной операции идет сканирование базы данных и алгоритм повторяется.

Таблица 5 – Часто встречающиеся 1 – элементные наборы

Предметный набор	Значение поддержки
а	3
б	4
в	4
г	3

Далее происходит формирование двухэлементных кандидатов, представленных в таблице 6, подсчет их поддержки и отсечение наборов с уровнем поддержки, меньшим 3. Оставшиеся двухэлементные наборы товаров, считающиеся часто встречающимися двухэлементными наборами аб, ав, бг, принимают участие в дальнейшей работе алгоритма.

Таблица 6 – Формирование 2-элементных кандидатов

Предметный набор	Значение поддержки
аб	3
ав	3
аг	2
бв	2
бг	3
вг	2

Таблица 7 – Часто встречающиеся 2 – элементные наборы

Предметный набор	Значение поддержки
аб	3
ав	3
бг	3

На последнем этапе алгоритм формирует трехэлементные наборы товаров: абв, абг, бвг, авг, подсчитывает их поддержку и отсекает наборы с уровнем поддержки меньшим 3. Набор товаров, представленный в таблице 9, может быть назван часто встречающимся.

Таблица 8 – Формирование 3-элементных кандидатов

Предметный набор	Значение поддержки
абв	3
абг	2
бвг	2
авг	2

Таблица 9 – Часто встречающийся 3 – элементный набор

Предметный набор	Значение поддержки
абв	3

Стоит отметить, благодаря информации об отсеченных элементах на предыдущих этапах алгоритма, алгоритм Apriori уменьшает количество тех наборов, которые никаким образом не смогут стать часто встречающимися. Отсечение предметных наборов происходит на основе предположения о том, что у часто встречающегося набора все подмножества должны быть часто встречающимися. Если в наборе находится подмножество, которое на

предыдущем этапе было определено как нечасто встречающееся, этот набор уже не включается в формирование и подсчет. Так предметные наборы аг, бв, вг были отброшены как нечасто встречающиеся, алгоритм не рассматривал предметные наборы абг, бвг, авг. Поскольку алгоритм Apriori отбросил нечасто встречающиеся наборы, последний этап алгоритма сразу определил предметный набор абв как единственный трехэлементный часто встречающийся набор.

В зависимости от размера самого длинного частного набора алгоритм Apriori просматривает базу данных определенное количество раз. Предложены некоторые разновидности алгоритма Apriori, которые оптимизируют стандартный алгоритм, для уменьшения числа сканирований базы данных и числа предметных наборов. Было предложено два усовершенствованных алгоритма Apriori:

1. AprioriTID. Интересной особенностью этого алгоритма является то, что база данных не используется для подсчета поддержки предметного набора на ряд элементов после первого прохода. Для этого используется кодировка элементов, выполненная в предыдущих проходах. На последующих этапах размер кодированных наборов может быть значительно меньше размера базы данных, что позволяет сэкономить ресурсы.

2. AprioriHybrid. Анализ времени работы алгоритмов Apriori и AprioriTID показывает, что в более ранних проходах Apriori достигает большего успеха, чем AprioriTID, но алгоритм AprioriTID работает лучше Apriori в более последующих проходах. Кроме того, этот алгоритм использует ту же процедуру для формирования предметных наборов. Исходя из этого, алгоритм AprioriHybrid предлагает объединить наилучшие свойства алгоритма Apriori и алгоритма AprioriTID. AprioriHybrid использует алгоритм Apriori в первых проходах и впоследствии переходит к алгоритму AprioriTid. Однако переход с Apriori на AprioriTID требует дополнительных ресурсов.

Таким образом, аффинитивный анализ применяется в задаче поиска ассоциативных правил, которые исследуют взаимные связи между событиями,

происходящие одновременно. Ассоциативные правила описывают связь, характеризующую двумя метриками — поддержкой и достоверностью.

Поддержка показывает, какой процент или количество содержит определенный набор данных. Достоверность показывает, какова вероятность того, что из наличия в транзакции набора условной части правила следует наличие в ней набора заключительной части. Одной из наиболее распространенных является алгоритм поиска ассоциативных правил Apriori.

Методика поиска правил ассоциации состоит из трех шагов:

1. Создание предметных наборов.
2. Отсечение лишних элементов.
3. Создание ассоциативных правил.

Чтобы уменьшить пространство поиска ассоциативных правил, алгоритм Apriori применяет свойство антимонотонности. Для оптимизации поиска ассоциативных правил применяется два усовершенствованных алгоритма Apriori – AprioriTID и AprioriHybrid, однако это требует дополнительных ресурсов.

ГЛАВА 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

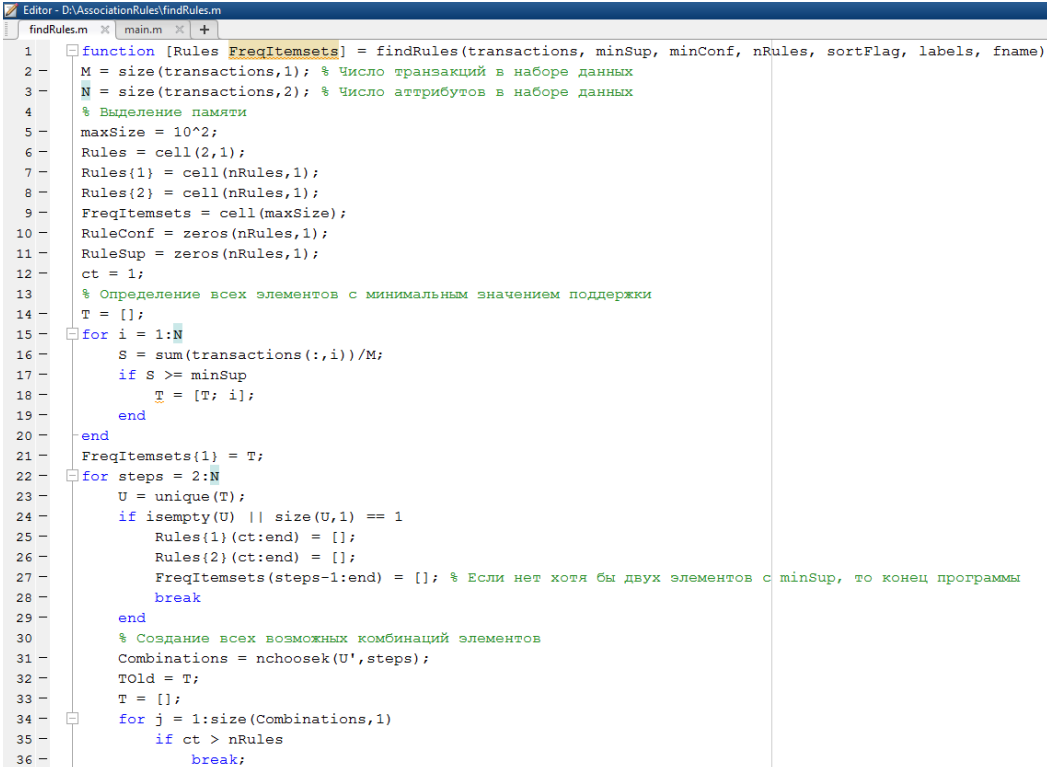
3.1. Программная реализация модуля

В ходе выполнения работы в математической среде на встроенном языке программирования Matlab был разработан модуль, реализующий алгоритм, который находит ассоциативные правила на основе анализа множества транзакций.

Возможности программы:

- реализован интерфейс программы для удобства использования;
- вызов справки;
- по заданным пользователем значениям поддержки и достоверности производится поиск ассоциативных правил;
- вывод результата в окне результата;
- вывод результата в консоли полностью или частично;
- сохранение результатов в файл со специальным расширением dmrg;
- открытие файла с результатами.

Часть программного кода представлена на рисунке 3.1. Полная программная реализация представлена в приложении А.



```
Editor - D:\AssociationRules\findRules.m
findRules.m  main.m  +
1  function [Rules FreqItemsets] = findRules(transactions, minSup, minConf, nRules, sortFlag, labels, fname)
2  M = size(transactions,1); % Число транзакций в наборе данных
3  N = size(transactions,2); % Число атрибутов в наборе данных
4  % Выделение памяти
5  maxSize = 10^2;
6  Rules = cell(2,1);
7  Rules{1} = cell(nRules,1);
8  Rules{2} = cell(nRules,1);
9  FreqItemsets = cell(maxSize);
10 RuleConf = zeros(nRules,1);
11 RuleSup = zeros(nRules,1);
12 ct = 1;
13 % Определение всех элементов с минимальным значением поддержки
14 T = [];
15 for i = 1:N
16     S = sum(transactions(:,i))/M;
17     if S >= minSup
18         T = [T, i];
19     end
20 end
21 FreqItemsets{1} = T;
22 for steps = 2:N
23     U = unique(T);
24     if isempty(U) || size(U,1) == 1
25         Rules{1}(ct:end) = [];
26         Rules{2}(ct:end) = [];
27         FreqItemsets(steps-1:end) = []; % Если нет хотя бы двух элементов с minSup, то конец программы
28         break
29     end
30     % Создание всех возможных комбинаций элементов
31     Combinations = nchoosek(U',steps);
32     Told = T;
33     T = [];
34     for j = 1:size(Combinations,1)
35         if ct > nRules
36             break;
```

Рисунок 3.1 – Программный код алгоритма поиска ассоциативных правил

3.2. Тестирование разработанной программы

Для того чтобы подключить разработанные функции необходимо в командной строке Matlab с помощью команды `addpath` указать путь, где находятся m-файлы с функциями, реализующие алгоритм. Затем с помощью команды `open` происходит открытие m-файлов.

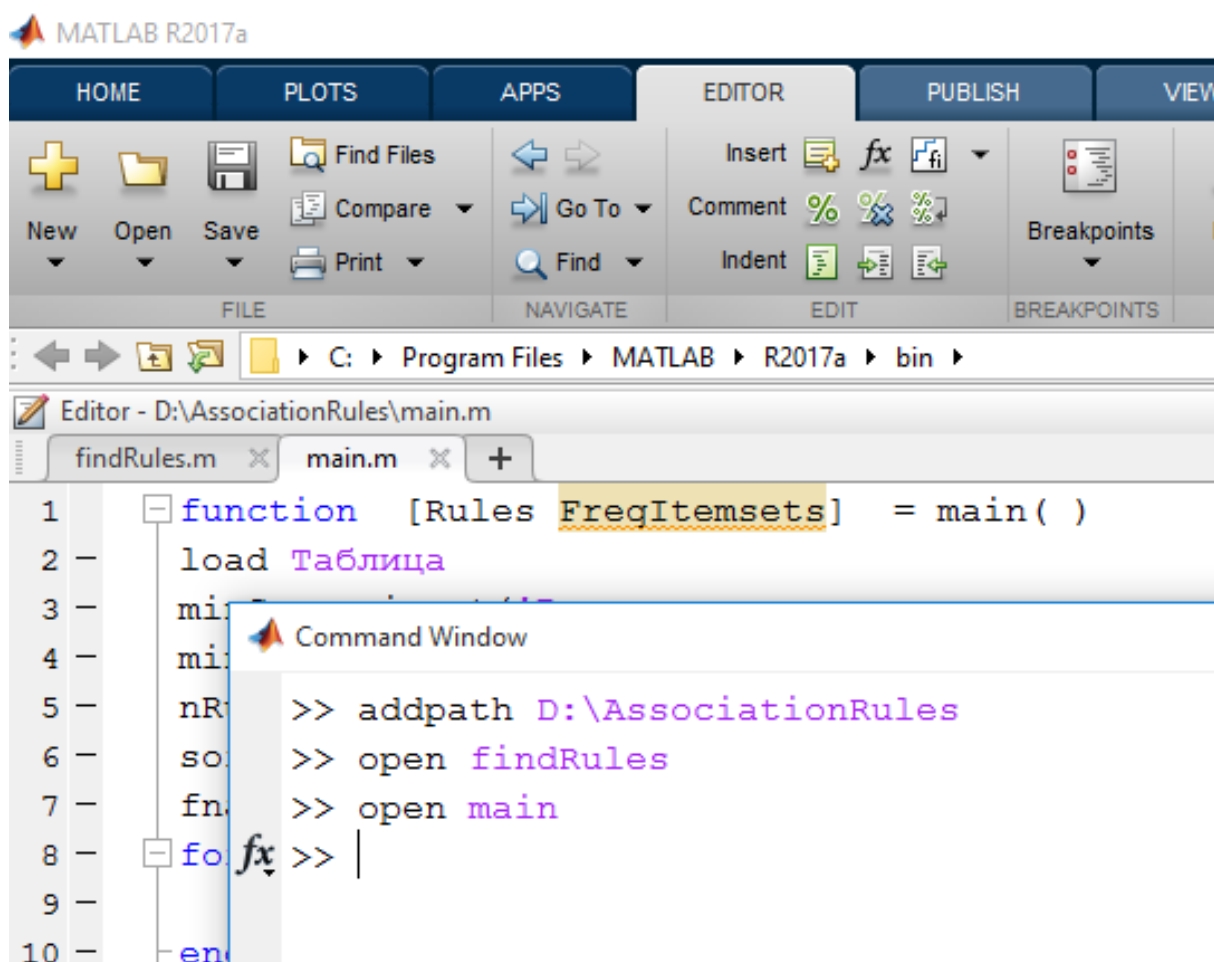


Рисунок 3.2 – Подключение и открытие функций через командную строку Matlab

Чтобы протестировать работу программы, был выбран dataset кондитерских изделий, состоящий из множества транзакций. Для простоты использования программы, был разработан интерфейс, представленный на рисунках 3.3 - 3.6.

На рисунке 3.3 представлено главное окно программы, на котором реализовано сверху меню программы, состоящая из двух вкладок, по середине

расположилась краткая информация, а снизу – две кнопки: тестирование алгоритма и выход из программы.

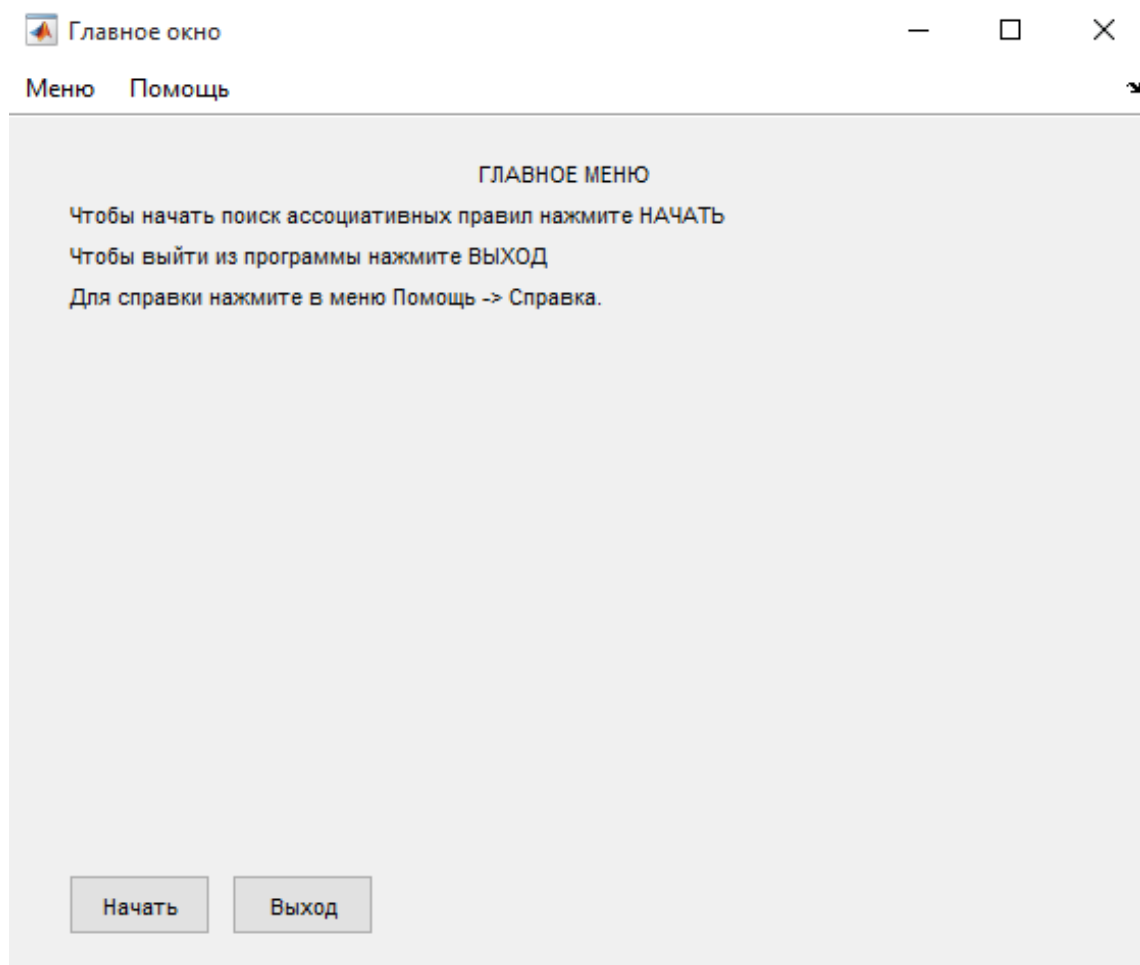


Рисунок 3.3 – Интерфейс программы. Главное окно

При нажатии на кнопку «Начать» открывается окно «Мой модуль», изображенный на рисунке 3.4. Здесь пользователь выбирает файл dataset в формате txt и разделяющий данные в файле знак (точка, запятая, точка с запятой и пробел). Далее необходимо ввести критерии поиска. Значения достоверности и поддержки вводятся от 1 до 100, по желанию можно указать знак %. Так же в программе реализована защита от ввода неверных данных, например если пользователь введет любое значение меньше 0 или больше 100, то программа отобразит ошибку и попросит пользователя ввести числовое значение в диапазоне от 1 до 100.

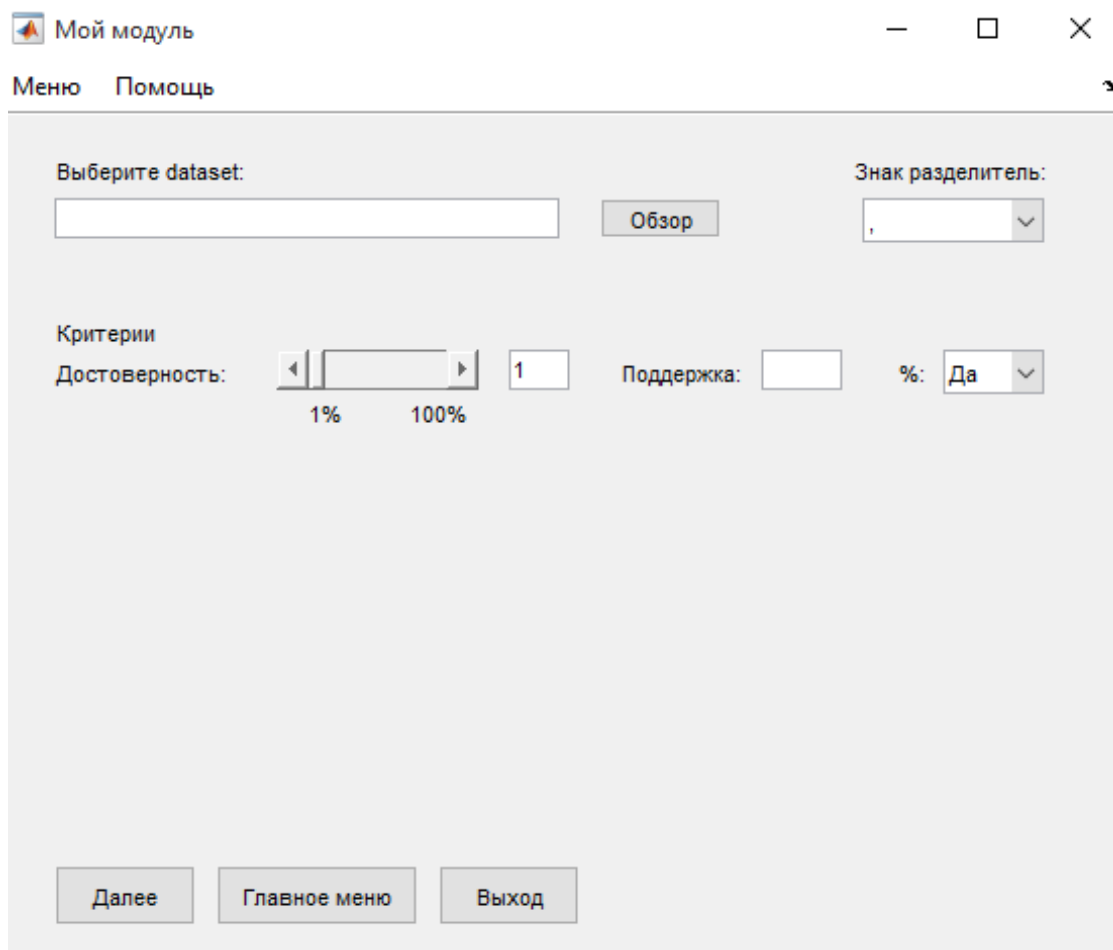


Рисунок 3.4 – Окно выбора файла и критериев поиска

Если же пользователь не знает как пользоваться программой, то для этого реализована справочная информация, которая показана на рисунке 3.5. Чтобы вызвать окно справки нужно в панели меню нажать «Помощь» и далее выбрать «Справка». В левой части находится список разделов. Чтобы отобразить информацию о конкретном разделе, нужно нажать на соответствующий раздел и на кнопку Показать и в правой части отобразится соответствующая информация. При необходимости можно вернуться на предыдущее окно, нажав кнопку «Закреть» на панели инструментов.

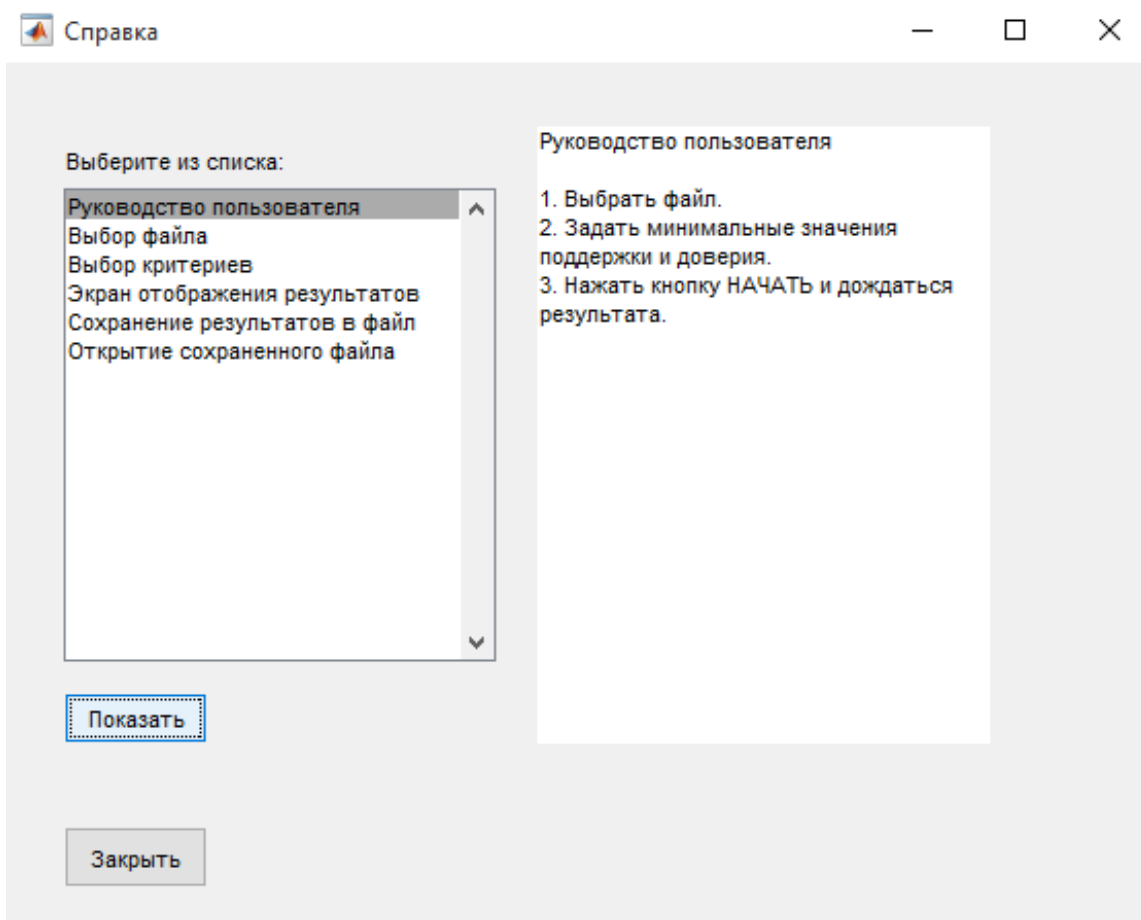


Рисунок 3.5 – Справочное окно программы

Выбрав файл и критерии поиска, при нажатии на клавишу «Далее» в окне «Мой модуль» начнется сама работа алгоритма, заключающая в поиске ассоциативных правил между различными событиями. По окончании работы алгоритма, программа выведет соответствующее сообщение о завершении поиска ассоциативных правил и откроется окно «Результаты», изображенное на рисунке 3.6. В окне «Все правила» представлены все составленные алгоритмом ассоциативные правила и их значения поддержки и достоверности (П и Д – соответственно). Ниже в окне «Элементы» представлены все элементы и их количество совпадений в транзакциях. Справа расположились введенные пользователем критерии поиска, а также сколько было составлено правил из выбранного dataset'a. Чтобы отобразить результаты в консоль, внизу расположена кнопка «Вывести в командную строку». Так же сверху во вкладке «Результаты» есть возможность вывода информации в консоль полностью, либо по 50 правил за нажатие на клавишу.

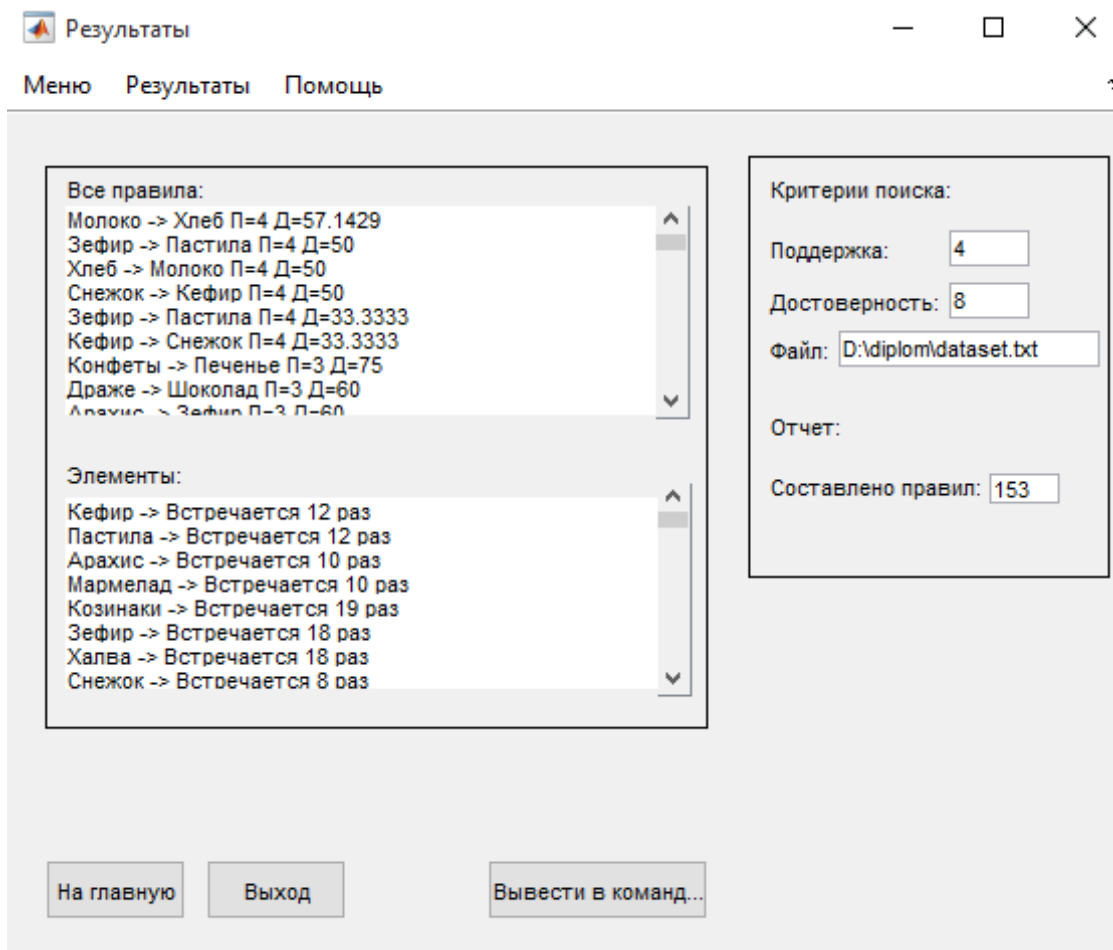


Рисунок 3.6 – Интерфейс программы. Окно вывода результатов
И последняя возможность программы – сохранение результатов в файл с
расширением dmr (data mining results) и открытие этого файла.

В результате тестирования программы получены различные результаты в
зависимости от введенных пользователем значений поддержки и
достоверности.

ЗАКЛЮЧЕНИЕ

Matlab предоставляет удобные средства для разработки алгоритмов на высокоуровневом языке программирования. В нём имеются все необходимые средства интегрированной разработки программного обеспечения, включая отладчик и профайлер.

В данной бакалаврской работе была изучена система компьютерной математики Matlab. Изучены модули, в которых реализованы алгоритмы искусственного интеллекта. Произведен анализ математического аппарата аффинитивного анализа, а также был рассмотрен пример аффинитивного анализа данных.

По итогу проделанной работы, был разработан модуль в математической среде Matlab, реализующий задачу аффинитивного анализа, а также для простоты использования модуля был реализован интерфейс программы.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. López, C. P. MATLAB Control Systems Engineering / César Pérez López. – Apress, 2014. – 163 p.
2. Optimization Toolbox. [Электронный ресурс] URL: https://matlab.ru/products/optimization-toolbox/optimization-toolbox-rus_web.pdf (Дата обращения: 18 февраля 2018 г.).
3. Roşca, D. G., Rădoiu, D. Step-by-step model for the study of the apriori algorithm for predictive analysis [Электронный ресурс] / Daniel Grigore Roşca, Dumitru Rădoiu. – University of Tîrgu Mureş, Romania, 2015. URL: <http://scientificbulletin.upm.ro/papers/20151/08%20Step%20by%20step%20Radoiu.pdf>. (Дата обращения: 18 февраля 2018 г.).
4. Paluszek, M., Thomas S. MATLAB Machine Learning / Michael Paluszek, Stephanie Thomas. – Apress, 2017. – 326 p.
5. Association rule learning. [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Association_rule_learning#cite_note-apriori-8 (Дата обращения: 2 марта 2018 г.).
6. Araghinejad, Sh., Data-Driven Modeling: Using MATLAB in Water Resources and Environmental Engineering / Shahab Araghinejad. – Springer, 2014. – 290 p.
7. Apriori algorithm [Электронный ресурс] URL: https://en.wikipedia.org/wiki/Apriori_algorithm (Дата обращения: 2 марта 2018 г.).
8. Parry, R. M. OmniSpect: An Open MATLAB - Based Tool for Visualization and Analysis of Matrix-Assisted Laser Desorption/Ionization and Desorption Electrospray Ionization Mass Spectrometry Images / R. Mitchell Parry. – 2013. – №24. – pp. 532-560.

9. Yali. Market Basket Analysis: identifying products that go well together [Электронный ресурс] / Yali. – 2017. URL: <https://discourse.snowplowanalytics.com/t/marketbasketanalysisidentifyingproducts-and-content-that-go-well-together/1332> (Дата обращения: 16 марта 2018 г.).
10. Lynch, S. Dynamical Systems with Applications using MATLAB / Stephen Lynch. – Springer International Publishing, 2014. – 514 p.
11. Dashidondok, Sh. Sh. Formation of Life Quality Indicators System through Search Algorithm of Association Rules [Электронный ресурс] / Dashidondok Sh. Shirapov. – 2016. URL: <http://ceur-ws.org/Vol-1726/paper-02.pdf>. (Дата обращения: 16 марта 2018 г.).
12. Zinchenko, Yu. V. Comparing the Use of the Simoyu Method and the Ident Program in Matlab for Parametric Identification of the Tension-Stabilizing 41 Process in the Winding of a Steamed Filament / Yu. V. Zinchenko, A. N. Timokhin, Yu. D. Rumyantsev // Fibre Chemistry. – 2016. – №47. – pp. 390-412.
13. Walaа M., Ahmed H., Hoda K. Combined Algorithm for Data Mining using Association rules [Электронный ресурс] / Walaа Medhat, Ahmed Hassan, Hoda Korashy. – Ain Shams University, Egypt, 2014. URL: <https://arxiv.org/ftp/arxiv/papers/14/10/1410.1343.pdf>. (Дата обращения: 16 марта 2018 г.).
14. Dori, D. When quantitative meets qualitative: enhancing OPM conceptual systems modeling with MATLAB computational capabilities / Dov Dori, Aharon Renick, Niva Wengrowicz // Research in Engineering Design. – 2016. – №27. – pp. 130-141.
15. MATLAB. Exponenta. MATLAB & Toolboxes. Приложения с GUI [Электронный ресурс]. URL: <http://matlab.exponenta.ru/gui/book2/1.php> (Дата обращения: 16 марта 2018 г.).
16. MATLAB. Exponenta. MATLAB. [Электронный ресурс]. URL: <https://matlab.ru/products/matlab> (Дата обращения: 16 марта 2018 г.).
17. MATLAB. Exponenta. Simulink. [Электронный ресурс]. URL: <https://matlab.ru/products/simulink> (Дата обращения: 16 марта 2018 г.).

18. MATLAB. Exponenta. Разработка алгоритмов. [Электронный ресурс]. URL: <https://matlab.ru/solutions/tech-calc/algo-dev> (Дата обращения: 16 марта 2018 г.).

19. Козлов, А. Н. Интеллектуальные информационные системы [Электронный ресурс] / Козлов Алексей Николаевич - Учебное издание, Россия, 2013 г. URL: <http://portalsga.ru/data/3118.pdf> (Дата обращения: 16 марта 2018 г.).

20. Дьяконов, В. П. MATLAB. Полный самоучитель. / Дьяконов Владимир Павлович – ДМК Пресс, 2012. – 768 с.

ПРИЛОЖЕНИЕ А

Программный код алгоритма

```
function [FinalRules, Rules]=Apriori(T,MinSup,MinConf)
```

```
    % Формирование 1-элементных кандидатов
```

```
    C{1}=cell(1,numel(Items));
```

```
    count{1}=zeros(size(C{1}));
```

```
    for r=1:numel(Items)
```

```
        C{1}{r}=Items(r);
```

```
        for i=1:numel(T)
```

```
            if IsContainedIn(C{1}{r},T{i})
```

```
                count{1}(r)=count{1}(r)+1;
```

```
            end
```

```
        end
```

```
    end
```

```
    L{1}=C{1}(count{1}/numel(T)>=MinSup);
```

```
    k=1;
```

```
    while ~isempty(L{k})
```

```
        b=[];
```

```
        for i=1:numel(L{k})
```

```
            b=union(b,L{k}{i});
```

```
        end
```

```
        b=b(:)';
```

```
        C{k+1}={};
```

```
        for i=1:numel(L{k})
```

```
            A=L{k}{i};
```

```
            for j=1:numel(b);
```

```
                if ~ismember(b(j),A)
```

```
                    New=[A b(j)];
```

```
                    Found=false;
```

```
                    for r=1:numel(C{k+1})
```

```
                        if IsSame(New,C{k+1}{r})
```

```
                            Found=true;
```

```
                            break;
```

```
                        end
```

```
                    end
```

```
                    if ~Found
```

```
                        C{k+1}=[C{k+1} {New}];
```

```
                    end
```

```
                end
```

```
            end
```

```
        end
```

```
        count{k+1}=zeros(size(C{2}));
```

```
        for r=1:numel(C{k+1})
```

```
            for i=1:numel(T)
```

```
                if IsContainedIn(C{k+1}{r},T{i})
```

```
                    count{k+1}(r)=count{k+1}(r)+1;
```

```
                end
```

```
            end
```



```

end

L{k+1}=C{k+1} (count{k+1}/numel(T)>=MinSup);

k=k+1;

end

L=L(1:end-1);

% Выборка ассоциативных правил
Rules={};
Supp=[];
Conf=[];
Lift=[];
for k=2:numel(L)
    for i=1:numel(L{k})

        countL=0;
        for j=1:numel(T)
            if IsContainedIn(L{k}{i},T{j})
                countL=countL+1;
            end
        end

        S=GetNonTrivialSubsets(L{k}{i});
        Q=S(end:-1:1);
        for r=1:numel(S)
            Rules=[Rules; {S{r} Q{r}}];
            countS=0;
            countQ=0;
            for j=1:numel(T)
                if IsContainedIn(S{r},T{j})
                    countS=countS+1;
                end
                if IsContainedIn(Q{r},T{j})
                    countQ=countQ+1;
                end
            end
            Supp=[Supp; countL/numel(T)];
            Conf=[Conf; countL/countS];
            Lift=[Lift; countL/(countS*countQ/numel(T))];
        end
    end
end

% Сортировка
[Conf, SortOrder]=sort(Conf, 'descend');
Supp=Supp(SortOrder);
Lift=Lift(SortOrder);
Rules=Rules(SortOrder,:);
for i=1:size(Rules)

```

```

        Rules{i,3}=Supp(i);
        Rules{i,4}=Conf(i);
        Rules{i,5}=Lift(i);
    end

    FinalRules=Rules(Conf>=MinConf & Lift>=1,:);

end

function b=IsContainedIn(A,B)

    b=all(ismember(A,B));

end

function S=GetNonTrivialSubsets(A)

    n=numel(A);

    S=cell(2^n-2,1);
    for i=1:numel(S)
        f=dec2binvec(i);
        f=[f zeros(1,n-numel(f))];
        S{i}=A(logical(f));
    end

end

end

```

Интерфейс программы

```

clear all;
close all;
warning off;

%Создание главного окна
Start_Window = figure('Name','Главное
окно','NumberTitle','Off','WindowStyle','Normal','menubar','none')
;

%Создание главного меню
FileMenu = uimenu(Start_Window,'Label','&Меню','Accelerator','F');
    FileOpenMenu =
uimenu(FileMenu,'Label','&Открыть','Accelerator','O','Callback','l
oadMiningResults');
    FileExitMenu =
uimenu(FileMenu,'Separator','On','Label','&Выход','Accelerator','x
','Callback','exitProgram');
HelpMenu = uimenu(Start_Window,'Label','&Помощь');
    HelpContentsMenu =
uimenu(HelpMenu,'Label','&Справка','Accelerator','H','Callback','h
elpContents');
    HelpAboutMenu = uimenu(HelpMenu,'Separator','On','Label','&O
программе','Callback','displayAbout');

%Создание кнопок
uicontrol(Start_Window,'Style','pushbutton','String','Начать','Pos
ition',[30,20,70,30],'Callback','MyProgram');
uicontrol(Start_Window,'Style','pushbutton','String','Выход','Posi
tion',[110,20,70,30],'Callback','exitProgram');

%Создания полей и кнопок о файле

uicontrol(Start_Window,'Style','text','Position',[230,380,190,20],
'String','ГЛАВНОЕ МЕНЮ','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','text','Position',[30,360,400,20],
'String','Чтобы начать поиск ассоциативных правил нажмите
НАЧАТЬ','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','text','Position',[30,340,400,20],
'String','Чтобы выйти из программы нажмите
ВЫХОД','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','text','Position',[30,320,400,20],
'String','Для справки нажмите в меню Помощь ->
Справка.','HorizontalAlignment','left');

```

```

clear all;
close all;
warning off;

%Создание окна
Start_Window = figure('Name','Мой
модуль','NumberTitle','Off','WindowStyle','Normal','menubar','none
');

%Создание главного меню
FileMenu = uimenu(Start_Window,'Label','&Меню','Accelerator','F');
FileOpenMenu =
uimenu(FileMenu,'Label','&Открыть','Accelerator','O','Callback','l
oadMiningResults');
FileExitMenu =
uimenu(FileMenu,'Separator','On','Label','&Выход','Accelerator','x
','Callback','exitProgram');
HelpMenu = uimenu(Start_Window,'Label','&Помощь');
HelpContentsMenu =
uimenu(HelpMenu,'Label','&Справка','Accelerator','H','Callback','h
elpContents');
HelpAboutMenu = uimenu(HelpMenu,'Separator','On','Label','&O
программе','Callback','displayAbout');

uicontrol(Start_Window,'Style','text','Position',[30,380,190,20],'
String','Выберите dataset:','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','edit','Position',[30,360,250,20],'
BackgroundColor','white','HorizontalAlignment','left','Tag','edit_
file_name');

uicontrol(Start_Window,'Style','pushbutton','Position',[300,360,60
,20],'String','Обзор','Callback','openDataFile');

uicontrol(Start_Window,'Style','text','Position',[425,350,190,50],
'String','Знак разделитель:','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','popupmenu','Position',[430,350,90,
30],'String','','|; |: |.
|пробел','BackgroundColor','white','Tag','delimiter');

uicontrol(Start_Window,'Style','text','Position',[30,300,110,20],'
String','Критерии','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','text','Position',[30,285,110,15],
'String','Достоверность:','HorizontalAlignment','left');

uicontrol(Start_Window,'Style','slider','Position',[140,285,100,20
],'Value',0,'Callback','alterConfBox','Tag','conf_slider');

uicontrol(Start_Window,'Style','edit','Position',[255,285,30,20],

```

```

String', '1', 'BackgroundColor', 'White', 'HorizontalAlignment', 'left'
, 'Tag', 'conf_edit', 'Callback', 'changeConfSlider');

uicontrol(Start_Window, 'Style', 'text', 'Position', [155,260,130,20],
'String', '1%', 'HorizontalAlignment', 'left');

uicontrol(Start_Window, 'Style', 'text', 'Position', [205,260,130,20],
'String', '100%', 'HorizontalAlignment', 'left');

uicontrol(Start_Window, 'Style', 'text', 'Position', [310,285,110,15],
'String', 'Поддержка:', 'HorizontalAlignment', 'left');

uicontrol(Start_Window, 'Style', 'edit', 'Position', [380,285,40,20], '
BackgroundColor', 'White', 'HorizontalAlignment', 'left', 'Tag', 'suppo
rt_edit');

uicontrol(Start_Window, 'Style', 'text', 'Position', [447,285,15,15], '
String', ':%', 'HorizontalAlignment', 'left');

uicontrol(Start_Window, 'Style', 'popupmenu', 'Position', [470,275,50,
30], 'String', 'Да|Нет', 'BackgroundColor', 'white', 'Tag', 'sup_type');
    %Создание кнопок
uicontrol(Start_Window, 'Style', 'pushbutton', 'String', 'Далее', 'Posi
tion', [30,20,70,30], 'Callback', 'evaluateCriteria');
uicontrol(Start_Window, 'Style', 'pushbutton', 'String', 'Главное
меню', 'Position', [110,20,100,30], 'Callback', 'Menu');
uicontrol(Start_Window, 'Style', 'pushbutton', 'String', 'Выход', 'Posi
tion', [220,20,70,30], 'Callback', 'exitProgram');

```