

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ  
ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ  
ИНТЕРНЕТ-МАГАЗИНА**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы  
направления 09.03.04 — Программная инженерия  
факультета КНиИТ  
Волкова Игоря Алексеевича

Научный руководитель \_\_\_\_\_ М. И. Сафончик  
старший преподаватель \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ С. В. Миронов  
к. ф.-м. н. доцент \_\_\_\_\_

## **СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>1 Обзор процесса интеллектуального анализа данных .....</b>	<b>5</b>
<b>1.1 Классификация задач интеллектуального анализа данных .....</b>	<b>5</b>
<b>1.1.1 Задачи классификации и регрессии .....</b>	<b>5</b>
<b>1.1.2 Задача прогнозирования временных рядов .....</b>	<b>6</b>
<b>1.1.3 Задача поиска ассоциативных правил .....</b>	<b>6</b>
<b>1.1.4 Задача кластеризации .....</b>	<b>7</b>
<b>1.2 Основные элементы концепции складирования данных .....</b>	<b>7</b>
<b>2 Проектирование модели интеллектуального анализа данных .....</b>	<b>8</b>
<b>2.1 Проектирование хранилища данных интернет-магазина .....</b>	<b>8</b>
<b>2.2 Разработка ETL процесса для загрузки данных в хранилище .....</b>	<b>8</b>
<b>2.3 Создание модели прогнозирования количества продаж .....</b>	<b>9</b>
<b>3 Разработка мобильного приложения .....</b>	<b>10</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>13</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>14</b>

## **ВВЕДЕНИЕ**

В современном мире интернет-технологий владельцам бизнеса приходится сталкиваться с огромными объемами данных. Ручная обработка этих данных является невозможной, в связи с чем возникает необходимость в использовании аналитических продуктов. Все популярные аналитические продукты представлены либо для настольных компьютеров, либо являются платными, в связи с чем возникает задача разработки мобильного клиента для аналитической обработки данных.

Целью настоящей выпускной квалификационной работы является разработка мобильного приложения для анализа данных интернет-магазина средствами SQL Server Analysis Services. Для достижения данной цели поставлены следующие задачи:

- рассмотреть процесс интеллектуального анализа данных;
- спроектировать реляционное хранилище данных для интернет-магазина;
- организовать процесс ETL для переноса исходных данных в хранилище;
- построить модель прогнозирования продаж товаров по категориям;
- рассмотреть средства разработки мобильных приложений;
- рассмотреть основные концепции JavaScript библиотеки React Native;
- изучить компоненты библиотеки ADOMD.NET, предоставляющие доступ к средствам интеллектуального анализа данных SSAS;
- разработать графический интерфейс пользователя для взаимодействия с моделями интеллектуального анализа данных SSAS.

Для разработки приложения были использованы следующие инструменты:

- Microsoft SQL Server Developer Edition;
- Microsoft SQL Server Analysis Services;
- Microsoft SQL Server Management Studio;
- React Native — библиотека для разработки кроссплатформенных мобильных приложений;
- Expo — платформа, позволяющая «на лету» обновлять приложения React Native после внесения изменений;
- ASP.NET — C# фреймворк для создания веб-сервисов;
- ADOMD.NET — библиотека для платформы .NET, предназначенная для взаимодействия с Analysis Services.

**Структура и объем работы.** Бакалаврская работа состоит из введения, трёх разделов, заключения, списка использованных источников и трёх приложений. Общий объем работы — 53 страницы, из них 46 страниц — основное содержание, включая 20 рисунков. Список использованных источников содержит 20 наименований.

Первый раздел «Обзор процесса интеллектуального анализа данных» содержит обзор задач Data Mining, а также обоснование необходимости использования хранилища данных как структуры хранения данных для анализа.

Второй раздел «Проектирование модели интеллектуального анализа данных» содержит описание процесса проектирования хранилища данных, построения ETL процесса для переноса данных из источника в хранилище, а также пример проектирования модели прогнозирования количества продаж, базирующейся на основе алгоритма временных рядов.

Третий раздел «Разработка мобильного приложения» содержит обзор средств разработки мобильных приложений, обзор основных концепций библиотек React Native и Redux, а также описание процесса разработки мобильного клиента для взаимодействия со средствами интеллектуального анализа данных SQL Server.

# **1 Обзор процесса интеллектуального анализа данных**

## **1.1 Классификация задач интеллектуального анализа данных**

Целью интеллектуального анализа данных (англ. Data Mining) является обнаружение неявных закономерностей в наборах данных. Развитие Data Mining как научного направления стало активно происходить в 90-х годах XX века и было обусловлено широким распространением технологий автоматизированной обработки информации и накоплением в компьютерных системах больших объемов данных. Несмотря на то, что в то время уже существовали способы получения нужной информации из базы данных, во многих случаях этого было уже недостаточно. Возникла потребность поиска взаимосвязей между отдельными событиями среди больших объемов данных, для чего понадобились методы математической статистики, теории искусственного интеллекта и ряда других областей.

Приведем классическое определение интеллектуального анализа данных, введенное одним из основателей данного направления Григорием Пятецким-Шапиро [1, с. 68]: Data Mining — исследование и обнаружение машиной в сырых данных скрытых знаний, которые ранее не были известны, нетривиальны, практически полезны и доступны для интерпретации.

Основными задачами, решаемыми средствами Data Mining, являются такие задачи, как классификация, регрессия, поиск ассоциативных правил и кластеризация [2]. Рассмотрим эти задачи подробнее.

### **1.1.1 Задачи классификации и регрессии**

Задача классификации сводится к определению класса объекта по его характеристикам. Стоит отметить, что множество классов, к которым можно отнести объект, должно быть известно заранее. В качестве примера можно привести задачу классификации клиентов банка по их кредитоспособности. Другая распространенная задача классификации — распознавание образов цифр (10 классов).

Задача классификации рассматривается как задача определения значения одного из параметров объекта (зависимый параметр) на основании значений других параметров (независимые параметры). Если значениями независимых и зависимых переменных являются действительные числа, то такая задача называется задачей регрессии. Задача классификации и регрессии решается в два

этапа. На первом этапе выделяется обучающая выборка — данные, для которых известны значения зависимых и независимых переменных. На основании этой выборки строится модель определения значения зависимой переменной, часто называемой функцией классификации или регрессии.

На втором этапе построенная модель применяется к анализируемым объектам (объектам, для которых неизвестно значение зависимой переменной).

### 1.1.2 Задача прогнозирования временных рядов

Частным случаем задачи классификации является задача прогнозирования временных рядов [1, с. 136]. Временным рядом называется последовательность событий, упорядоченных по времени их наблюдения. События фиксируются через равные интервалы времени  $T$  и представляются в виде последовательности  $\{e_1, e_2, \dots, e_n\}$ , где  $e_i$  — событие в момент времени  $t_i$ ,  $i = 1, 2, \dots, n$ ,  $n$  — общее количество событий.

Событие может характеризоваться несколькими атрибутами:

$$e_i = \{x_1^i, x_2^i, \dots, x_m^i\}$$

где  $x_j^i$  —  $j$ -й атрибут, характеризующий событие в момент времени  $t_i$ . Если один из этих атрибутов может быть определен из значений других атрибутов в текущий или предыдущие моменты времени, то такой атрибут является зависимым. Атрибуты, через которые можно выразить зависимый атрибут, называются независимыми.

Таким образом, задачу прогнозирования временного ряда можно сформулировать следующим образом. Пусть дан временной ряд  $\{e_1, e_2, \dots, e_n\}$ . Требуется на его основании определить значение  $e_{n+k}$  при  $k > 0$ .

### 1.1.3 Задача поиска ассоциативных правил

Задача поиска ассоциативных правил заключается в определении часто встречающихся наборов объектов в большом множестве таких наборов. Эта задача является частным случаем задачи классификации [3].

При анализе часто вызывает интерес последовательность происходящих событий. При обнаружении закономерностей в таких последовательностях можно с некоторой долей вероятности предсказывать появление событий в будущем, что позволяет принимать более правильные решения. Такая задача

является разновидностью задачи поиска ассоциативных правил и называется секвенциальным анализом. Основным отличием задачи секвенциального анализа от поиска ассоциативных правил является установление отношения порядка между исследуемыми наборами.

#### 1.1.4 Задача кластеризации

Задача кластеризации состоит в разделении исследуемого множества объектов на группы схожих объектов [1, с. 158–163]. Такие группы называются кластерами. Кластеризация отличается от классификации отсутствием необходимости иметь выделенную зависимую переменную. Кроме этого, в задаче кластеризации искомое множество кластеров не является заранее определенным.

### 1.2 Основные элементы концепции складирования данных

Прежде чем переходить к анализу данных, необходимо сначала привести их к виду, удобному для решения задач анализа. Предприятия в подавляющем большинстве случаев используют для обработки транзакций OLTP базы данных. OLTP базы данных предназначены, как следует из их названия, для оперативной обработки транзакций — в таких базах должны быстро выполняться операции вставки, обновления, удаления, а также небольшие запросы на выборку. Использование оперативных баз ускоряет работу системы, делая её более удобной для пользователей. Информация в таких базах всегда (а точнее, между транзакциями) является согласованной, что, на первый взгляд, позволяет получать актуальные отчеты в любой момент времени. Однако, такие базы оказываются непригодными для задач анализа данных по ряду причин, основной из которых является неприемлемо долгое время выполнения аналитического запроса.

Таким образом, при решении задач анализа данных возникает необходимость использования альтернативной структуры хранения информации. Подходящей структурой является хранилище данных — предметно-ориентированная, интегрированная, неизменяемая и поддерживающая хронологию электронная коллекция данных для обеспечения процесса принятия решений [4].

## **2 Проектирование модели интеллектуального анализа данных**

### **2.1 Проектирование хранилища данных интернет-магазина**

Для анализа был использован датасет Global Superstore, содержащий в себе данные о продажах товаров для офиса. Искомый датасет находится в свободном доступе и предоставляется компанией Tableau [5].

Прежде чем переходить к анализу данных, представленных в этом датасете, необходимо поместить их в хранилище данных. Так, на основании полей, содержащихся в датасете, было спроектировано хранилище данных типа «звезда». Разработанное хранилище содержит единственную таблицу фактов, в которой накапливается информация о совершенных заказах. Товары и их категории, клиенты, информация о географическом положении, тип доставки и дата являются размерностями данного хранилища.

### **2.2 Разработка ETL процесса для загрузки данных в хранилище**

После проектирования хранилища необходимо загрузить в него данные из датасета. Процесс загрузки данных в хранилище представлен тремя основными шагами:

1. Извлечение данных из внешних источников.
2. Трансформация и очистка данных с целью соответствия их бизнес-модели.
3. Непосредственная загрузка преобразованных данных в хранилище.

Процесс загрузки данных в хранилище подобным образом имеет название ETL, что расшифровывается как «Extract, Transform, Load» — извлечение, преобразование, загрузка [6].

Для создания ETL процесса были использованы службы интеграции SQL Server (SSIS — SQL Server Integration Services) [7]. Визуальный интерфейс к SSIS встроен в среду разработки Microsoft Visual Studio. Пакет SSIS представляет собой несколько задач, выполняемых последовательно. Задачи потока данных используются для перемещения данных из источника в приёмник. Так, для загрузки данных в таблицы размерности, применяется следующий алгоритм.

1. Прочитать исходные данные.
2. Сгруппировать данные по необходимым атрибутам с целью получения уникальных строк.

### 3. Загрузить данные в хранилище.

Для загрузки данных в таблицу фактов необходимо найти соответствующие значения внешних ключей. Для этого применяется преобразование «Уточняющий запрос» [8], которое позволяет соединить данные в потоке с некоторой таблицей по определенным столбцам, после чего выдать в качестве результата строки, для которых нашлось совпадение.

## 2.3 Создание модели прогнозирования количества продаж

Для прогнозирования количества продаж товаров по категориям была разработана модель интеллектуального анализа данных, основанная на алгоритме временных рядов [9]. Разработанная модель использует данные, полученные из представления OrderByCategoryTimeSeries, спроектированного на основе хранилища. В качестве входных столбцов модели были заданы столбцы year и month. Ключевым столбцом модели является столбец name; ключом времени является столбец id. Модель прогнозирует значения столбца quantity. График полученного временного ряда можно просмотреть при помощи средств SQL Server Data Tools [10].

### **3 Разработка мобильного приложения**

В настоящее время существует большое количество различных технологий, позволяющих разрабатывать мобильные приложения. Так, для разработки под ОС Android, можно использовать непосредственно средства платформы, предоставляемые Android SDK. В таком случае разработка приложения производится при помощи IDE Android Studio на языках Java или Kotlin. Аналогично, для разработки под iOS, используются средства, предоставляемые платформой iOS SDK, а разработка производится в IDE XCode на языках Swift или ObjectiveC.

В ходе выполнения работы был дополнен API для доступа к данным SQL Server Analysis Services, разработанный ранее в рамках курсовой работы. Для доступа к службам Data Mining использовались различные классы библиотеки ADOMD.NET. Так, для получения результатов запроса использовался класс AdomdDataReader [11]. В API были добавлены методы, позволяющие получить доступ к списку моделей Data Mining, выполнять запросы, получать результаты хранимых запросов.

API разрабатывался при помощи фреймворка ASP.NET. Для передачи данных использовался протокол HTTP. Для определения параметров HTTP запроса, передаваемых непосредственно в строке запроса, применяется атрибут [FromQuery] [12]. В случае, если не был передан список наименований требуемых столбцов, данный метод возвращает результат DMX-запроса [13] на получение обучающей выборки модели со всеми столбцами с указанным количеством строк. Иначе, возвращается результат аналогичного запроса, но с указанными столбцами.

Помимо этого, при помощи библиотек React Native [14] и Redux [15] был разработан мобильный интерфейс приложения MobileDataMining. Помимо React Native, в проекте использовался ряд дополнительных модулей Node.js:

Разработанное приложение позволяет осуществлять следующие действия:

- просматривать список моделей интеллектуального анализа данных;
- просматривать информацию о выбранной модели (название, алгоритм, входные и прогнозируемые столбцы);
- выполнять простейшие DMX-запросы;
- просматривать графики временных рядов;

- выполнять сохраненные в базе данных запросы;
- просматривать результаты запросов в виде таблиц и графиков;
- axios — модуль для отправки HTTP-запросов;
- prop-types — модуль для задания компонентам определенных типов пропсов;
- immer — модуль для получения нового экземпляра состояния без изменения старого;
- react-native-chart-kit — модуль для построения графиков в React Native [16];
- react-native-elements — модуль, содержащий в себе различные готовые компоненты (кнопки, слайдеры, флагги и т.д.) [17];
- react-native-table-component — модуль для отображения таблиц в React Native [18];
- redux — библиотека для управления состоянием.
- react-redux — модуль, соединяющий React и Redux.
- redux-thunk — модуль, позволяющий отправлять в хранилище Redux асинхронные действия;
- react-navigation [19] — модуль, позволяющий осуществлять навигацию между экранами в React Native.

Поскольку приложение разрабатывалось при помощи React Native с целью достижения кроссплатформенности, оно одинаково работает как на устройствах под управлением iOS, так и на устройствах с OS Android.

Поскольку для управления состоянием приложения в проекте используется библиотека Redux, была создана соответствующая структура проекта. Так, вся логика, связанная с работой с Redux, была помещена в папку store. Данная папка содержит в себе папки actions и reducers, содержащие в себе файлы с действиями и редьюсерами соответственно.

В папке actions содержится файл actionTypes.js, содержащий в себе определение типов возможных действий:

```

1 export const SET_DATA_MINING_MODEL_LIST = 'SET_DATA_MINING_MODEL_LIST';

2

3 export const SET_SELECTED_MODEL_BY_ID = 'SET_SELECTED_MODEL_BY_ID';
4 export const SET_SELECTED_MODEL_COLUMNS = 'SET_SELECTED_MODEL_COLUMNS';
5 export const CLEAR_SELECTED_MODEL = 'CLEAR_SELECTED_MODEL';
6 export const SET_MODEL_FETCHING_ERROR = 'SET_MODEL_FETCHING_ERROR';

```

```
7
8 export const SET_CASES_QUERY_RESULT = 'SET_CASES_QUERY_RESULT';
9 export const CLEAR_CASES_QUERY_RESULT = 'CLEAR_CASES_QUERY_RESULT';
10 export const SET_QUERY_ERROR = 'SET_QUERY_ERROR';
11
12 export const SET_TIME_SERIES_CHART_DATA = 'SET_TIME_SERIES_CHART_DATA';
13 export const CLEAR_TIME_SERIES_CHART_DATA = 'CLEAR_TIME_SERIES_CHART_DATA';
```

Стоит отметить, что типы действий в программе необязательно задавать в виде строковых констант. Однако, подход с использованием именованных констант значительно упрощает процесс разработки [20], поскольку, например, исключается возможность опечатки при наборе строкового значения (в случае же ошибки в названии переменной проект не скомпилируется).

Помимо этого, папка `actions` содержит в себе следующие файлы с определением действий над состоянием:

- `dmModelActions.js` — содержит определение действий над частью состояния, касающейся работы с моделями интеллектуального анализа данных;
- `dmxQueryActions.js` — содержит определение действий над частью состояния, касающейся работы с DMX-запросами и их результатами.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения бакалаврской работы было разработано мобильное приложение, позволяющее взаимодействовать со средствами интеллектуального анализа данных SQL Server Analysis Services. Разработанное приложение является кроссплатформенным благодаря использованию библиотеки React Native. Оно позволяет получать доступ к моделям интеллектуального анализа данных, развернутых на сервере SSAS. Для примера демонстрации работы приложения было разработано реляционное хранилище данных для интернет-магазина продажи офисной техники. На основе данных, расположенных в хранилище, была построена модель прогнозирования временного ряда.

Помимо разработанного приложения, был дополнен и улучшен существующий проект API доступа к средствам SQL Server Analysis Services. В частности, были добавлены методы получения информации о моделях интеллектуального анализа данных и исполнения произвольного DMX-запроса.

Для разработки приложения применялись такие технологии, как библиотека React Native и фреймворк ASP.NET. Для управления состоянием приложения была использована библиотека Redux. Сочетание React Native и Redux позволило создать удобное и эффективное мобильное приложение.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Барсегян, А. А. Анализ данных и процессов / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс. — 3 изд. — Санкт-Петербург: БХВ-Петербург, 2009.*
- 2 *Дядичев, В. В. Задачи и методы интеллектуального анализа данных / В. В. Дядичев, Е. В. Ромашка, Т. В. Голуб // Геополитика и экодинамика регионов. — 2015. — Т. 1, № 3. — С. 23–29.*
- 3 *MacLennan, J. Data Mining with Microsoft SQL Server 2008 / J. MacLennan, T. ZhaoHui, B. Crivat. — Indianapolis: Wiley Publishing, 2009. — Pp. 359–370.*
- 4 *Immon, W. H. Building the Data Warehouse / W. H. Immon. — Indianapolis: Wiley Computer Publishing, 2002. — Pp. 21–29.*
- 5 Tableau [Электронный ресурс]. — URL: <https://www.tableau.com/why-tableau/what-is-tableau> (Дата обращения 03.05.2021). Загл. с экрана. Яз. англ.
- 6 *Kimball, R. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data / R. Kimball, J. Caserta. — 3 edition. — Indianapolis: Wiley Publishing, 2004. — Pp. 3–28.*
- 7 SQL Server Integration Services [Электронный ресурс]. — URL: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15> (Дата обращения 05.05.2021). Загл. с экрана. Яз. англ.
- 8 Lookup Transformation [Электронный ресурс]. — URL: <https://docs.microsoft.com/en-us/sql/integration-services/data-flow/transformations/lookup-transformation?view=sql-server-ver15> (Дата обращения 15.05.2021). Загл. с экрана. Яз. англ.
- 9 Microsoft Time Series Algorithm [Электронный ресурс]. — URL: <https://docs.microsoft.com/en-us/analysis-services/data-mining/microsoft-time-series-algorithm?view=asallproducts-allversions> (Дата обращения 07.05.2021). Загл. с экрана. Яз. англ.
- 10 SQL Server Data Tools [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/sql/ssdt/sql-server-data-tools?>

- `view=sql-server-ver15` (Дата обращения 03.05.2021). Загл. с экрана. Яз. англ.
- 11 AdomdDataReader Class [Электронный ресурс]. – URL: <https://docs.microsoft.com/en-us/dotnet/api/microsoft.analysisservices.adomdclient.adomddatareader?view=analysisservices-dotnet> (Дата обращения 05.05.2021). Загл. с экрана. Яз. англ.
  - 12 Model Binding in ASP.NET Core [Электронный ресурс]. – URL: <https://docs.microsoft.com/en-us/aspnet/core/mvc/models/model-binding?view=aspnetcore-5.0> (Дата обращения 24.05.2021). Загл. с экрана. Яз. англ.
  - 13 *Tennick, A.* Practical DMX Queries for Microsoft SQL Server Analysis Services 2008 / A. Tennick. – McGraw-Hill, 2009. – Pp. 25–40.
  - 14 React Native [Электронный ресурс]. – URL: <https://reactnative.dev/docs/getting-started> (Дата обращения 03.05.2021). Загл. с экрана. Яз. англ.
  - 15 Redux: A Predictable State Container for JS Apps [Электронный ресурс]. – URL: <https://redux.js.org/> (Дата обращения 03.05.2021). Загл. с экрана. Яз. англ.
  - 16 react-native-chart-kit [Электронный ресурс]. – URL: <https://www.npmjs.com/package/react-native-chart-kit/v/2.6.1> (Дата обращения 14.05.2021). Загл. с экрана. Яз. англ.
  - 17 React Native Elements [Электронный ресурс]. – URL: <https://reactnativeelements.com/docs/> (Дата обращения 14.05.2021). Загл. с экрана. Яз. англ.
  - 18 React Native Table Component [Электронный ресурс]. – URL: <https://www.npmjs.com/package/react-native-table-component> (Дата обращения 14.05.2021). Загл. с экрана. Яз. англ.
  - 19 Stack Navigator [Электронный ресурс]. – URL: <https://reactnavigation.org/docs/stack-navigator> (Дата обращения 14.05.2021). Загл. с экрана. Яз. англ.
  - 20 Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – Санкт-Петербург: БХВ, 2021. – С. 122–129.