

**Министерство образования и науки РФ**

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Казанский (Приволжский) федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ им. Н.И. ЛОБАЧЕВСКОГО**

**КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА**  
Направление: 02.03.01 – «математика и компьютерные науки»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**СБОР И ВИЗУАЛЬНЫЙ АНАЛИЗ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ, С  
ПОМОЩЬЮ ИНСТРУМЕНТОВ PYTHON И POWER BI.**

Студент 4 курса

Группа 05-404

«\_\_» \_\_\_\_\_ 2018 г.

\_\_\_\_\_ Л.И. Захаров

Научный руководитель:

кандидат физ.-мат. наук

«\_\_» \_\_\_\_\_ 2018 г.

\_\_\_\_\_ А.А. Новиков

Заведующий кафедрой мат. анализа КФУ,

доктор физ.-мат. наук, профессор

«\_\_» \_\_\_\_\_ 2018 г.

\_\_\_\_\_ С.Р. Насыров

КАЗАНЬ – 2018

## Содержание

Введение .....	3
1. Визуальный метод анализа данных(Visual Maining) .....	5
1.1 Выполнение визуального анализа данных .....	5
1.2 Характеристики средств визуализации данных .....	8
1.3 Методы визуализации.....	10
1.4 Теоретические основы Power BI.....	14
2. Сбор и визуализация данных на примере нефтеперерабатывающей компании .....	20
2.1 Система работы, сбор и объединение данных .....	20
2.3 Визуализация информации с помощью Power BI .....	21
Заключение .....	23
Список литературы .....	24

## **Введение**

Сбор и анализ больших объемов данных становится все более востребованным направлением. Необходимость поиска закономерностей в больших базах данных усложняет и без того нетривиальную задачу анализа. Такая ситуация особенно характерна для бизнеса, связанного с розничной торговлей, телекоммуникациями, банками, интернетом. В их базах данных аккумулируется огромное количество информации, связанной с транзакциями: чеки, платежи, звонки, логи и т.п.

В настоящей работе рассматривается задача сбора, данных из разных источников, их структурирование, и последующий анализ посредством визуализации.

Для решения данной задачи будем использовать один из языков объектно-ориентированного программирования, в нашем случае этим языком является Python, и многофункциональное гибридное веб-приложение Power BI Desktop.

Целью данной работы является изучение методов анализа данных, а также реализация, на конкретном примере, одного из них. Для достижения поставленной цели необходимо решить ряд задач:

- 1) изучить структуру и метод визуального анализа данных;
- 2) изучить принципы работы Power BI;
- 3) разработать алгоритм для сбора данных;
- 4) реализовать данный алгоритм в программный продукт;
- 5) отобразить полученные результаты в наглядной форме.

В качестве новизны можно отметить предложенный в работе новый вариант работы с большими объемами данных, лежащих в основе системы

Power BI. Это позволяет избавиться от некоторых недостатков, присущих другим системам по работе с данными.

Работа объемом  $n$  листа основного текста состоит из введения, двух частей, заключения, списка используемой литературы, содержащего четырнадцать источников, и приложения.

## **1. Визуальный метод анализа данных(Visual Maining)**

### **1.1 Выполнение визуального анализа данных**

Результаты, получаемые при анализе данных с помощью методов Data Mining, не всегда удобны для восприятия человеком. Во множестве классификационных или ассоциативных правил, в математических формулах человеку достаточно сложно быстро и легко найти новые и полезные знания. Из-за сложности информации это не всегда возможно и в простейших графических видах представления знаний, таких как деревья решений, дейтограммы, двумерные графики и т. П. В связи с этим возникает необходимость в более сложных средствах отображения результатов анализа. К ним относятся средства визуального анализа данных, которые в зарубежной литературе часто называют термином Visual Mining.

Основной идеей визуального анализа данных является представление данных в некоторой визуальной форме, позволяющей человеку погрузиться в данные, работать с их визуальным представлением, понять их суть, сделать выводы и напрямую взаимодействовать с данными.

До недавнего времени визуальный анализ данных для отображения результатов на обычных мониторах использовал только двумерную или очень простую трехмерную графику. Более сложные графические образы отображать в реальном времени было достаточно сложно и дорого. Однако прогресс в области аппаратных средств вывода изображений способствовал и совершенствованию средств визуального анализа данных. В настоящее время существует достаточно большое количество различных видов графических образов, позволяющих представлять результаты анализа в виде, удобном для понимания человеком.

С помощью новых технологий пользователи способны оценивать: большие объекты или маленькие, далеко они находятся или близко. Пользователь в реальном времени может двигаться вокруг объектов или

кластеров объектов и рассматривать их со всех сторон. Это позволяет использовать для анализа естественные человеческие перцепционные навыки в обнаружении неопределенных образцов в визуальном трехмерном представлении данных.

Визуальный анализ данных особенно полезен, когда о самих данных мало известно и цели исследования до конца непонятны. За счет того, что пользователь напрямую работает с данными, представленными в виде визуальных образов, которые он может рассматривать с разных сторон и под любыми углами зрения, в прямом смысле этого слова, он может получить дополнительную информацию, которая поможет ему более четко сформулировать цели исследования.

Таким образом, визуальный анализ данных можно представить как процесс генерации гипотез. При этом сгенерированные гипотезы можно проверить или автоматическими средствами (методами статистического анализа или методами Data Mining), или средствами визуального анализа. Кроме того, прямое вовлечение пользователя в визуальный анализ имеет два основных преимущества перед автоматическими методами:

а) визуальный анализ данных позволяет легко работать с неоднородными и зашумленными данными, в то время как не все автоматические методы могут работать с такими данными и давать удовлетворительные результаты;

б) визуальный анализ данных интуитивно понятен и не требует сложных математических или статистических алгоритмов.

Следствием этих преимуществ является то, что визуальный анализ выполняется быстрее и в некоторых случаях дает лучший результат, чем автоматические методы анализа.

Еще одним важным достоинством визуального анализа является высокая степень конфиденциальности полученных сведений, т. к. они целиком сосредоточены в голове аналитика и не сохраняются даже в оперативной памяти компьютера.

Все перечисленные преимущества позволяют еще больше облегчить работу аналитика и повысить качество получаемых знаний при совместном использовании визуального анализа и методов автоматического анализа. Визуальный анализ данных обычно выполняется в три этапа:

а) беглый анализ — позволяет идентифицировать интересные шаблоны и сфокусироваться на одном или нескольких из них;

б) увеличение и фильтрация — идентифицированные на предыдущем этапе шаблоны отфильтровываются и рассматриваются в большем масштабе;

в) детализация по необходимости — если пользователю нужно получить дополнительную информацию, он может визуализировать более детальные данные.

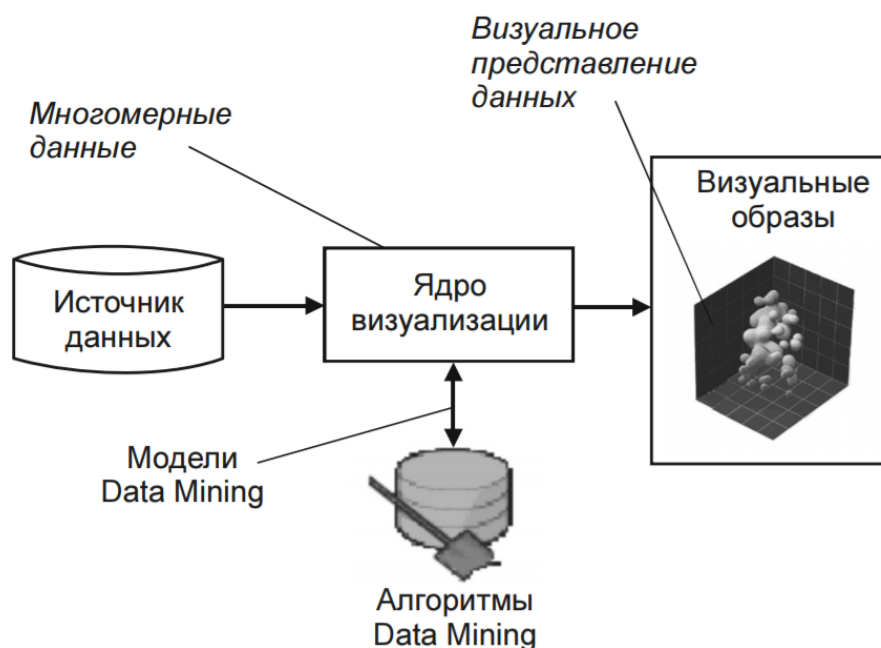


Рис.1 Процесс визуализации данных

Процесс визуализации изображен на рис.1. Так же как и при анализе данных, информация извлекается из некоторого источника, например, из базы данных или из файлов. Затем к ней могут быть применены алгоритмы Data Mining для выявления скрытых закономерностей (классификации, кластеризации и т.п.). Как результаты применения алгоритмов, так и исходные данные подвергаются обработке в ядре визуализации. Основной целью обработки является приведение многомерных данных к такому виду, который можно было бы представить на экране монитора.

## **1.2 Характеристики средств визуализации данных**

Существует достаточно большое количество средств визуализации данных, предоставляющих различные возможности. Для выбора таких средств изучим более подробно три основные характеристики:

- a) характер данных, которые нужно визуализировать с помощью данного средства;
- b) методы визуализации и образы, в виде которых могут быть представлены данные;
- c) возможности взаимодействия с визуальными образами и методами для лучшего анализа данных.

Наборы визуализируемых данных, как и в Data Mining, представляют собой матрицы, в которых ряды являются данными (например, записями об экспериментах, покупки в магазине и т. П.), а колонки — атрибутами данных. При этом данные могут характеризоваться одним или несколькими атрибутами. Кроме того, сами данные могут иметь более сложную структуру: иерархическую, текстовую, графическую и т. П. Таким образом, выделяют следующие виды данных, с которыми могут работать средства визуализации:

- одномерные данные — одномерные массивы, временные ряды и т. П.;
- двумерные данные — точки двумерных графиков, географические координаты и т. П.;



- многомерные данные — финансовые показатели, результаты экспериментов и т. П.;

- тексты и гипертексты — газетные статьи, Web-документы и т. П.;

- иерархические и связанные — структура подчиненности в организации, электронная переписка людей, гиперссылки документов и т. П.;

- алгоритмы и программы — информационные потоки, отладочные операции и т. П.

Для визуализации перечисленных типов данных используются различные визуальные образы и методы их создания. Очевидно, что количество визуальных образов, которыми могут представляться данные, ограничиваются только человеческой фантазией. Основное требование к ним — это наглядность и удобство анализа данных, которые они представляют. Методы визуализации могут быть как самые простые (линейные графики, диаграммы, гистограммы и т. П.), так и более сложные, основанные на сложном математическом аппарате. Кроме того, при визуализации могут использоваться комбинации различных методов. Выделяют следующие типы методов визуализации:

а) стандартные 2D/3D-образы — гистограммы, линейные графики;

б) геометрические преобразования — диаграмма разброса данных, параллельные координаты и т. П.;

в) отображение иконок — линейчатые фигуры (needle icons) и звезды (star icons);

г) методы, ориентированные на пиксели — рекурсивные шаблоны, циклические сегменты и т. П.;

д) иерархические образы — древовидные карты и наложение измерений.

В результате применения методов визуализации будут построены визуальные образы, отражающие данные. Однако этого не всегда бывает достаточно для полного анализа. Пользователь должен иметь возможность работать с образами: видеть их с разных сторон, в разном масштабе и т. П. Для этого у него должны быть соответствующие возможности взаимодействия с образами:

- динамическое проецирование;
- интерактивная фильтрация;
- масштабирование образов;
- интерактивное искажение;
- интерактивное комбинирование.

### 1.3 Методы визуализации

Методы геометрических преобразований:

Основная идея методов геометрических преобразований — визуализировать преобразования и проекции данных в декартовом и в недекартовом геометрических пространствах. К этим методам относятся:

- точки и матрицы;

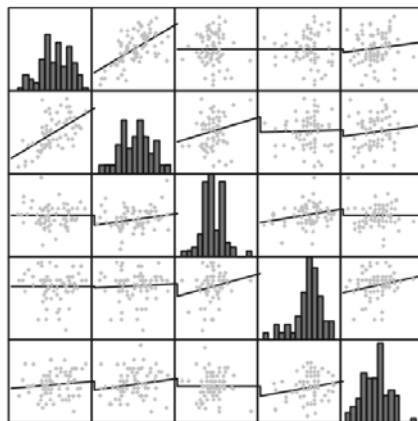
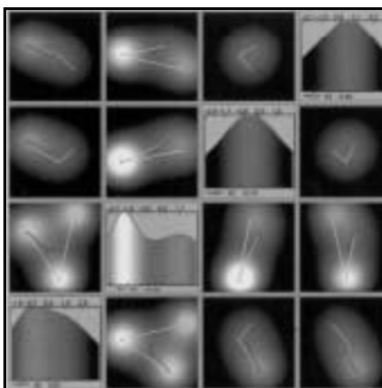


Рис.2 Пример матрицы диаграмм разброса

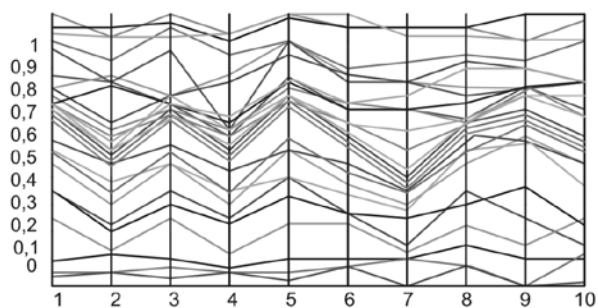
- гипердоли;



*Рис.3 Пример гипердолей*

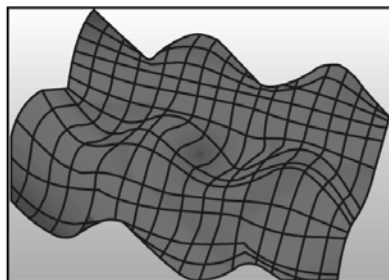
- поверхностные и объемные графики, контуры;

- параллельные координаты;



*Рис.4 Примеры параллельных координат*

- текстуры и растры.



### *Рис.5 Пример отображения текстур*

Методы отображения иконок:

Подход, основанный на отображении иконок, предполагает каждому объекту данных ставить в соответствие некоторую иконку. При этом атрибуты объекта должны отображаться различными визуальными свойствами иконок. Иконки могут комбинироваться в матрицы или графики и, таким образом, предоставляют возможность анализировать все объекты в целом. Использование иконок предполагает следующие методы визуализации:

- линейчатые фигуры;
- «лица Чернова»;
- цветные иконки;
- глифы 1 и др.

Методы, ориентированные на пиксели:

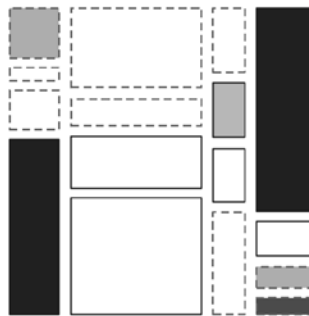
Методы, ориентированные на пиксели, используют для представления каждого элемента данных цветные пиксели. Выделяют следующие подобные методы:

- заполнение пространства;



*Рис.6 Заполнение пространства по спирали*

- рекурсивные шаблоны;
- мозаика.

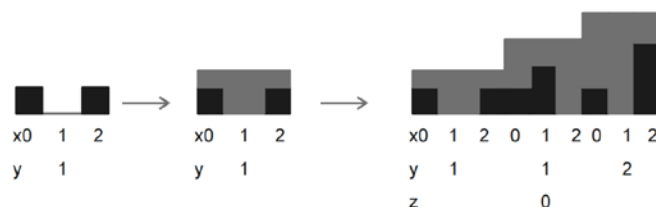


*Рис.7 Пример мозаики*

Методы иерархических образов:

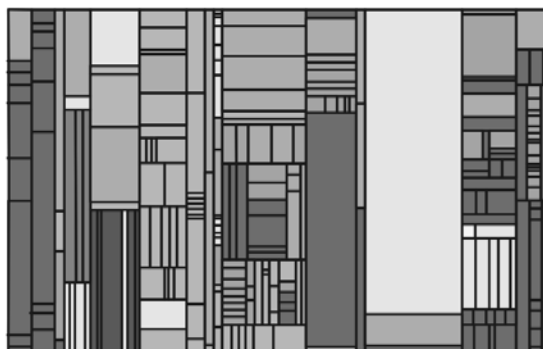
Иерархические образы используются для отображения иерархий и отношений в данных. Они применяются в следующих методах:

- иерархические оси;



*Рис.8 Пример иерархических осей*

- наложение измерений;
- деревья.



## **1.4 Теоретические основы Power BI**

Power BI – это набор продуктов от Microsoft для решения задач бизнес-аналитики. Основное назначение Power BI – это создание визуализаций разнообразных интерактивных данных, полученных из самых разных источников. Первый публичный показ Power BI состоялся в 2014.

В Power BI входят:

- Power BI Desktop (в прошлом Power BI Designer) – приложение под Windows для создания и работы с отчетами. Power BI работает на технологиях, которые также доступны и из MS Excel: Power Query используется для получения данных, PowerPivot используется как быстрая база данных;

- Power BI Service – сервис в интернете, в который можно публиковать pbix файлы напрямую из Power BI Desktop. Сервис поддерживает возможность редактирования загруженных отчетов прямо в интерфейсе. Благодаря сервису загруженные отчеты можно встраивать в отдельные страницы или обновлять по расписанию конкретные отчеты;

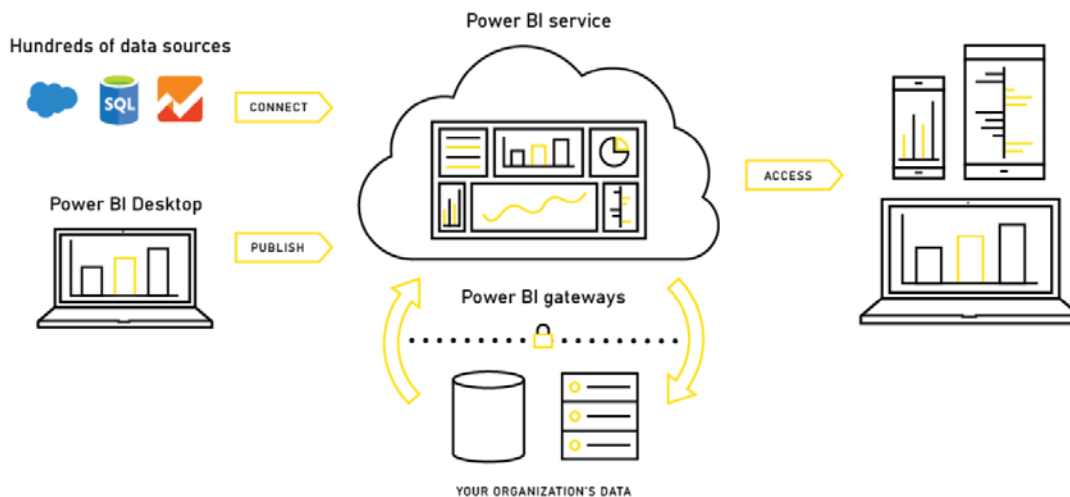
- Power BI mobile – мобильные приложения для iphone и windows phone, с доступом к данным загруженным в облако;

- Power BI Gateway (Personal, Enterprise) – приложения для windows, для регулярной загрузки данных в облако с использованием локального компьютера;

- Power BI Publisher – надстройка над excel для публикации фрагментов таблиц в облако для использования внутри Power BI;

Power BI — достаточно мощная и при этом бесплатная BI платформа. Microsoft вкладывает много средств в развитие этого продукта, в связи с чем

часто выходят обновления, расширяющие ее возможности. Как правило, десктопная версия используется в качестве конструктора, после чего разработанный файл публикуется в Power BI Service.



*Рис.10 Структура Power BI*

В этой работе будет использоваться Power BI desktop, и уже готовый отчет будет публиковаться в Power BI Service. Теперь рассмотрим, из чего состоит, приложение. Power BI — это, по сути, усовершенствованная комбинация некоторых ранее известных продуктов: Data Explorer, GeoFlow, BI Sites и мобильных технологий. Первые две надстройки теперь представлены общественности как Power Query и Power Maps соответственно. Это все можно представить следующим образом:

Power Query — это элемент, используемый для обнаружения и загрузки данных из локального расположения или из интернета. Например, из Википедии. По мере надобности используем PowerPivot для того чтобы форматировать колонки, изменить модель, и т.п. Анализ данных можем производить, используя функции визуализации Power View. Кроме того, используя Power Maps, вы можете визуализировать данные в пространстве, используя масштабируемые 3D карты. Теперь подробнее о каждом элементе.

Power Query раньше была известна как “Data Explorer”. Инструмент позволяет извлекать данные из широкого круга источников. Нативно поддерживаются SQL Server, Oracle DB2, а также каналы Odata, Azure SharePoint, Hadoop и HDInsight, Active Directory и Facebook. Стоит отметить, что если запрос обращен не к файлу, а к папке, то Power Query объединяет данные из всех файлов папки в единый результат. В процессе обновления новые файлы будут приниматься, как новые данные.

Наиболее важная возможность, которую дает Power Query является поиск и импорт данных из веб-ресурсов. Страницы будут показаны в виде табличных данных. Теперь Вы не только имеете доступ к большому количеству общедоступных источников данных, но и можете публиковать собственные источники данных.

Еще один компонент Power BI — Power Map. Power Map наносит данные на 3D карту, которая в последствии может быть увеличена, повернута, анимирована и т.д.

Немного информации про Bi sites.

- Обновление данных. Ахилессова пята использования PowerPivot Office 365 — неспособность планировать обновления данных. В Office 365 обновление нужно было сделать вручную, а результаты загружены повторно. Теперь же ответ на этот вопрос есть: в локальной среде будут создаваться специальные «шлюзы» данных, которые будут управлять обновлением данных и передавать их в Power BI Service, но не наоборот. Это не требует изменения конфигурации брандмауэра или создания сложной инфраструктуры. Таким образом, можно будет настроить автоматическое обновление, причем для каждой отдельной книги;

- Галерея шаблонов. Когда Вы начнете работу, Вы сразу обратите внимание на галерею шаблонов. Здесь не только представлены превью, но и



есть возможность воспользоваться различным функционалом по обработке данных;

- Каталог данных. В равной степени следующая информация может быть использована и в описании Power Query, но так как она неразрывна с BI Sites, мы поместили ее в данный раздел. Каталог данных — это тот сервис, который регистрирует источники данных для Power Query;

- Визуализация. Безусловно, пользователи Power View и раньше видели на что способны инструменты визуализации, но теперь они стали еще совершеннее.

Power BI имеет множество встроенных коннекторов к различным сервисам и базам данных, с помощью которых вы в считанные минуты можете загрузить в программу нужный набор данных из различных источников, связать их между собой и построить консолидированные отчеты и диаграммы. На данный момент все доступные коннекторы можно распределить на четыре группы:

а) Группа “Файл”: Excel, CSV, XML, Текст, JSON, Папка;

б) Группа “База данных”: SQL Server, Access, SQL Server Analysis Service, Oracle, IBM DB2, MySQL, PostgreSQL, Sybase, Teradata, SAP HANA;

в) Группа “Azure”: База данных Microsoft Azure SQL, Microsoft Azure Marketplace, Microsoft Azure HDInsight, Хранилище BLOB-объектов, Табличное хранилище Microsoft Azure, Azure HDInsight Spark, Microsoft Azure DocumentDB, Хранилище озера данных Microsoft Azure;

д) Группа “Другое”: Интернет, Список SharePoint, Канал Odata, Файл Hadoop, Active Directory, Microsoft Exchange, Dynamics CRM online, Facebook, Google Analytics, Объекты Salesforce, Отчеты Salesforce, ODBC, R-скрипт, appFigures, GitHub, MailChimp, Marketo, QuickBook Online, Smartsheets, SQL Sentry, Stripe, SweetIQ, Twilio, Zendesk, Spark, Пустой запрос;

После загрузки данных доступны следующие элементы визуализации:

- линейчатая диаграмма с накоплением;
- гистограмма с накоплением;
- линейчатая диаграмма с группировкой;
- гистограмма с группировкой;
- нормированная линейчатая диаграмма;
- нормированная гистограмма;
- график;
- диаграмма с областями;
- диаграмма с областями с накоплением;
- линейная гистограмма и гистограмма с накоплением;
- линейная гистограмма и гистограмма с группировкой;
- каскадная диаграмма;
- точечная диаграмма;
- круговая диаграмма;
- диаграмма дерева;
- карта;
- таблица;
- матрица;
- заполненная карта;
- воронка;
- датчик;
- многострочная карточка;

- карточка;
- ключевой показатель эффективности;
- срез;
- кольцевой график;
- визуальный элемент r-script (на данный момент включается в параметрах программы).

Таким образом, Power BI является мощным и многофункциональным инструментом, Можно смотреть любые данные — где угодно и как угодно. Все в одном представлении.

## **2. Сбор и визуализация данных на примере нефтеперерабатывающей компании**

### **2.1 Система работы, сбор и объединение данных**

Объектом исследования является нефтеперерабатывающая компания, у которой имеются множество филиалов по всей стране. Для улучшения качества работы компания закупает новое оборудование. Для успешной работы с этим оборудованием поставщики выделяют людей, которые будут обучать сотрудников. Как и говорилось ранее, компания большая, и для руководителей нужен инструмент мониторинга и анализа прогресса прохождения обучения.

Для решения задачи сбора и объединения данных было разработано программное обеспечение в среде JetBrains PyCharm 2018 на языке программирования Python. Для мониторинга прогресса и визуализации информации создан отчет в Power BI.

На данный момент все файлы хранятся в книгах Excel.

Алгоритм сбора данных:

- a) Создание общего файла со списком людей, их должностей, подразделений в которых они работают, табельных номеров, даты приема на работу и т.д;
- b) Объединение первичного файла с файлом, содержащим список курсов;
- c) Назначение на курсы по правилам, которые лежат в отдельном файле.
- d) Отметка персонала, которые уже прошли курс.

Для наглядности подробнее рассмотрим каждый пункт.

В изначальном списке есть просто вакансии на которых еще не назначили работника. Убираем лишние строки. Проверяем полученные данные.

На втором этапе, необходимо соединить несколько таблиц. Все курсы делятся на несколько подразделов. Подразделы находится в отдельных файлах. В общем файле все подразделы выделены разным цветом. Получается сводная таблица, у которой по вертикали Ф.И.О. сотрудника, а по горизонтали секция с общими данными, после которой идут названия курсов. Таблица на данном этапе будет содержать только общую информацию о сотруднике, ячейки с определением на курсы пустые.

На третьем этапе, в общей сводной таблице, для каждого сотрудника нужно определить на какой курс он записан. Для этого используем обозначения “Назначен” и “Рекомендован”. В связи с тем, что количество сотрудников и курсов может быть достаточно большим, существуют правила назначения для каждого класса курсов. Эти правила лежат в отдельных книгах Excel. У каждого сотрудника есть должность, и на основе этой должности программное обеспечение автоматически определяет какой сотрудник записан на тот или иной курс.

На заключительном этапе программа открывает файл, в котором хранится список людей, прошедших те или иные курсы. В общей сводной таблице отметка “Назначен” или “Рекомендован” меняется на “Пройден”.

Таким образом получается одна большая сводная таблица, с которой и будет работать приложение Power BI.

## **2.2 Визуализация информации с помощью Power BI**

Для визуализации информации загружаем данные в приложение Power BI Desktop. Все данные будут разделены на столбцы, с которыми в будущем будет происходить работа. Для получения более наглядной, полезной

информации, можно создавать несколько отчетов. В процессе преобразований данных и составление из них визуальной картины получился отчет следующего вида:

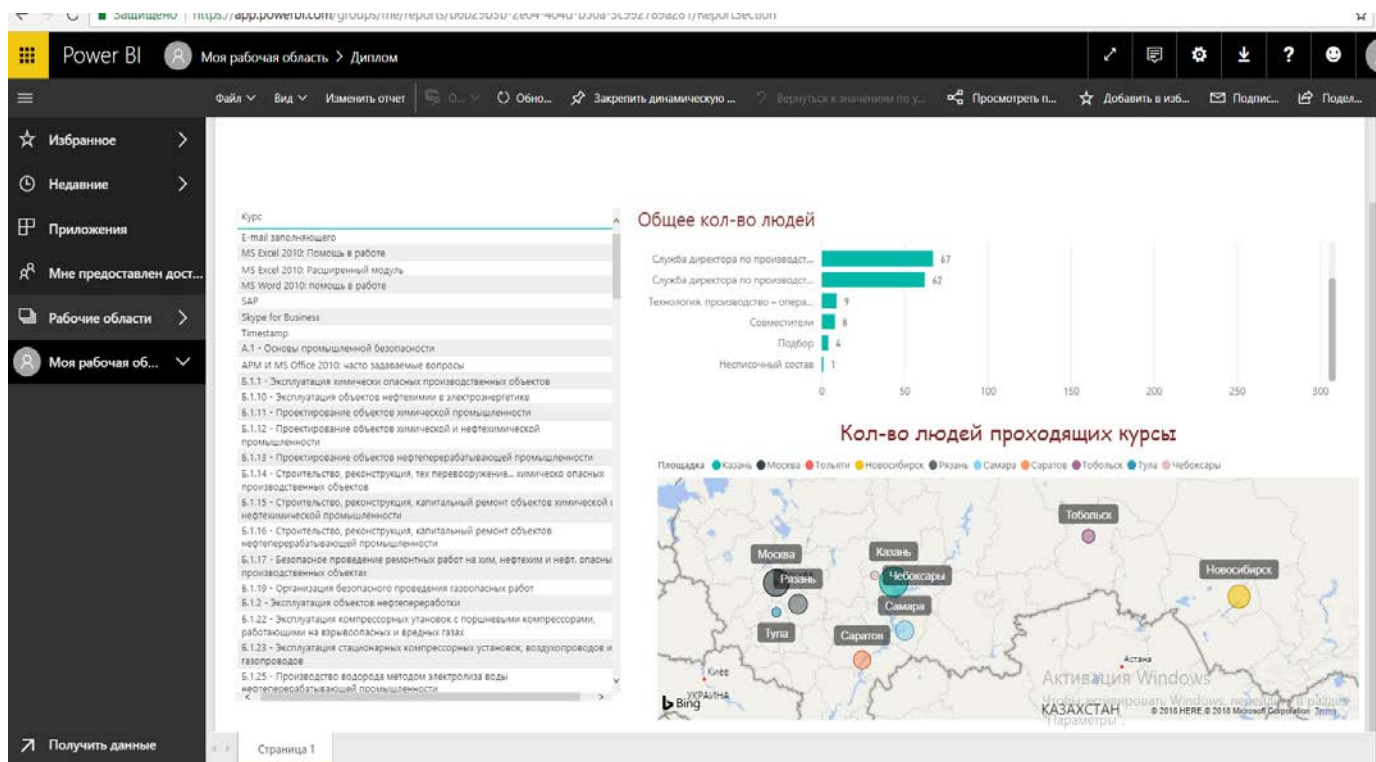


Рис.12 Макет отчета в Power BI

Данный отчет уже загружен в Power BI Service. Этот инструмент дает возможность помещать этот отчет на веб-страницу, в аккаунт на социальных сетях и т.д. Любой человек, имеющий доступ к нему, имеет возможность просмотра и скачивания.

Таким образом, с помощью программного обеспечения на языке Python и набора средств бизнес-аналитики Power BI можно получить готовое решение для бизнеса.

## **Заключение**

В данной выпускной квалификационной работе произведен сбор данных, посредством разработки определенного алгоритма, визуальный анализ и исследование данных для промышленного предприятия. Дан обзор различных видов и форм отображения данных. Реализован алгоритм сбора и обработки информации с разных файлов. Исследован принцип работы сервиса Power BI. Программа для данного проекта, написана на языке Python. Полученные после работы программы данные обрабатываются средством Power BI. Все это актуально в связи с увеличением количества информации и нахождения, отображения, анализа определенных данных из общего объема, для конкретных нужд. Все цели данной работы достигнуты.

## Литература

1. Паклин Н. Б., Орешков В. И. Визуализация данных / Бизнес-аналитика. От данных к знаниям. — 2-е изд. — СПб.: Питер, 2013. — 173 с.
2. Барсегян А.А., Куприянов М.С., Холод И.И. Анализ данных и процессов: Учебное пособие / СПб:БХВ-Петербург, 2009. - 512 с
3. Марк Саммерфилд. Python на практике. / Перевод с английского. — М.: ДМК Пресс, 2014. — 338 с.
4. Сузи Р. А. Язык программирования Python: Учебное пособие. / М.: ИНТУИТ, БИНОМ. Лаборатория знаний, 2006. — 328 с.
5. Маккинли У. Python и анализ данных. / Перевод с английского. — М.: ДМК Пресс, 2015. — 482 с.
6. А. Н. Чаплыгин. Учимся программировать вместе с Питоном. / Учебник. — ревизия 226. — 135 с.
7. Rob Collie. Power Pivot and Power Bi: The Excel User's Guide to Dax, Power Query, Power Bi & Power Pivot in Excel 2010-2016 / 2016г. — 314 с.
8. Сайт Microsoft. Документы о Power BI [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/power-bi/>, свободный.



## Приложения

### Интерфейс программы:

#### 1 Скрипт:

```
import openpyxl
wb = openpyxl.load_workbook(filename="C:\Users\Leo\Desktop\Диплом\1\СМР ОБ
11.05.2018.xlsx")
out = openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/blank.xlsx")
sh = wb['11.05.2018']
output = out['Лист1']
j=2
for i in range(6, 739):
    if not (sh.cell(row=i, column=15).value is None or sh.cell(row=i,
column=15).value == ""):
        print(sh.cell(row = i, column = 15).value)
        if int(sh.cell(row = i, column = 15).value)>10000000:
            output.cell(row=j, column=21).value = sh.cell(row = i, column =
15).value
            name = sh.cell(row=i, column =16).value.split()
            k=2
            for word in name:
                output.cell(row=j, column=k).value = word
                k+=1
            output.cell(row=j,column=9).value = sh.cell(row=i,
column=19).value
            for k in range(1,8):
                output.cell(row=j, column=k+10).value = sh.cell(row=i,
column=k).value
                output.cell(row=j, column=18).value = sh.cell(row=i,
column=10).value
                output.cell(row=j, column=19).value = sh.cell(row=i,
column=8).value
                output.cell(row=j, column=20).value = sh.cell(row=i,
column=9).value
                output.cell(row=j, column=22).value = "Назначен"
                output.cell(row=j, column=23).value = "Назначен"
                output.cell(row=j, column=24).value = "Назначен"
            j+=1
out.save(filename="C:/Abiroy/2d_form/temp/output.xlsx")
```

#### 2 Скрипт:

```
import openpyxl
wb = openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output.xlsx")
sh1 = wb["Лист1"]
wb1 = openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Правила
назначения ОО ОКО ИТ/ИТ правила назначения.xlsx")
sh2 = wb1["Лист1"]
for i in range(2, 608):
    if not (sh1.cell(row=i, column=18).value is None):
        if sh1.cell(row=i, column=18).value == sh2.cell(row=2,
column=1).value:
            for j in range(2, 27):
                sh1.cell(i, column=95+(j-2)).value=sh2.cell(row=2,
column=j).value
```

```

        if not (sh2.cell(row=3, column=1).value.rfind(sh1.cell(row=i,
column=18).value)==-1):
            for j in range(2, 27):
                sh1.cell(i, column=95+(j-2)).value=sh2.cell(row=3,
column=j).value
wb1 = openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Правила
назначения ОО ОКО ИТ/ОО правила назначения.xlsx")
sh2 = wb1["Лист1"]
for i in range(2, 608):
    if not (sh1.cell(row=i, column=18).value is None):
        if sh1.cell(row=i, column=18).value == sh2.cell(row=2,
column=1).value:
            for j in range(2, 12):
                sh1.cell(i, column=85+(j-2)).value=sh2.cell(row=2,
column=j).value
            for k in range(3,11):
                if not (sh1.cell(row=i, column=19).value.rfind(sh2.cell(row=k,
column=1).value)==-1):
                    for j in range(2, 12):
                        sh1.cell(i, column=85+(j-2)).value=sh2.cell(row=k,
column=j).value
                    if not (sh1.cell(row=i, column=19).value.rfind(sh2.cell(row=11,
column=1).value) == -1) and sh1.cell(row=i,
column=14).value.rfind("ППНР/ПНР") == -1 :
                        for j in range(2, 12):
                            sh1.cell(i, column=85 + (j - 2)).value = sh2.cell(row=k,
column=j).value

wb.save(filename="C:/Abiroy/2d_form/temp/output_2.xlsx")

```

### 3 Скрипт:

```

import openpyxl
wb1=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output_2.xlsx")
wb2=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Правила
назначения ОО ОКО ИТ/ОО правила.xlsx")
sh1=wb1["Лист1"]
sh2=wb2["20.04.2018"]
for j in range(2, 608):
    for i in range(5, 740):
        t=True
        if not (sh1.cell(row=j, column=13).value == sh2.cell(row=i,
column=3).value):
            print(sh1.cell(row=j, column=13).value, sh2.cell(row=i,
column=3).value, t)
            t = False
        if not (sh1.cell(row=j, column=14).value == sh2.cell(row=i,
column=4).value):
            print(sh1.cell(row=j, column=14).value, sh2.cell(row=i,
column=4).value, t)
            t = False
        if not (sh1.cell(row=j, column=19).value == sh2.cell(row=i,
column=8).value):
            print(sh1.cell(row=j, column=19).value, sh2.cell(row=i,
column=8).value, t)
            t = False
        if not (sh1.cell(row=j, column=21).value is None) and
(sh1.cell(row=j, column=21).value == sh2.cell(row=i, column=16).value):
            print(sh1.cell(row=j, column=21).value, sh2.cell(row=i,
column=16).value, t)

```

```

        t = True
    print(i, " ", j, " ", t)
    if t:
        for k1 in range(25, 84):
            if not (sh2.cell(row=i, column=k1-2).value is None):
                sh1.cell(row=j, column=k1).value = "Назначен"
wb1.save(filename="C:/Abiroy/2d_form/temp/output_3.xlsx")

```

#### 4 Скрипт:

```

import openpyxl
wb1=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output_3.xlsx")
wb2=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Факт
обучения ОО ОКО ИТ/Чистые данные/Факт пройденных ОО курсов.xlsx")
sh1=wb1["Лист1"]
sh2=wb2["Лист1"]
for j in range(2, 608):
    for i in range(5, 1992):
        if not (sh2.cell(row=i, column=11).value is None):
            str = sh2.cell(row=i, column=11).value.split()
            k1=2
            t = True
            for x in str:
                if (x is None) or (sh1.cell(row=j, column=k1).value is None):
                    t=False
                if t:
                    if sh1.cell(row=j, column=k1).value.rfind(x)==-1 and
x.rfind(sh1.cell(row=j, column=k1).value)==-1:
                        t=False
                    k1+=1
            #if
not(sh2.cell(row=i,column=3).value==sh1.cell(row=j,column=14).value):
            # t=False
            if t:
                print(i, " ", j, " ", t)
                for k1 in range(22, 67):
                    if not (sh2.cell(row=i, column=k1-10).value is None):
                        sh1.cell(row=j, column=k1).value = "Пройден"
wb1.save(filename="C:/Abiroy/2d_form/temp/output_4.xlsx")

```

#### 5 Скрипт:

```

import openpyxl
wb1=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output_4.xlsx")
wb2=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Факт
обучения ОО ОКО ИТ/Чистые данные/ФАКТ пройденных ОКО ИТ курсов.xlsx")
sh1=wb1["Лист1"]
sh2=wb2["Лист1"]
for j in range(2, 608):
    for i in range(2, 85590):
        if not (sh2.cell(row=i, column=1).value is None):
            str = sh2.cell(row=i, column=1).value.split()
            k1 = 2
            t = True
            for x in str:
                if (x is None) or (sh1.cell(row=j, column=k1).value is None):
                    t = False

```

```

        if t:
            if sh1.cell(row=j, column=k1).value.rfind(x) == -1 and
x.rfind(sh1.cell(row=j, column=k1).value) == -1:
                t = False
            k1 += 1
        # if
not(sh2.cell(row=i, column=3).value==sh1.cell(row=j, column=14).value):
        # t=False
        if t:
            print(i, " ", j, " ", t)
            k2 = 22
            flag = True
            while not (sh1.cell(row=1, column=k2).value is None) and
flag:
                if not (sh1.cell(row=1,
column=k2).value.rfind(sh2.cell(row=i, column=3).value) == -1) or not
(sh2.cell(row=i, column=3).value.rfind(sh1.cell(row=1, column=k2).value) == -
1):
                    sh1.cell(row=j, column=k2).value = "Пройден"
                    flag = False
                k2 += 1
            if flag:
                sh1.cell(row=1, column=k2).value=sh2.cell(row=i,
column=3).value
                sh1.cell(row=j, column=k2).value = "Пройден"
wb1.save(filename="C:/Abiroy/2d_form/temp/output_5.xlsx")

```

## 6 Скрипт:

```

import openpyxl
f = open('C:/Abiroy/2d_form/csv/content.txt', 'rb')
wb1=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output_5.xlsx")
sh1=wb1["Лист1"]
str=''
for line in f:
    str+=line.decode('utf8')
list = str.split('\r')
for x in list:
    course = x.split(',')
    if course[2]=="set":
        value = "Назначен"
    elif course[2]=="recomend":
        value = "Рекомендован"
    elif course[2]=="delete":
        value = " "
    elif course[2]=="end":
        value = "Пройден"
    if not (course[3] is None):
        idlist0=course[3].split()[1:-1]
        course_id=idlist0[0]
        for id in idlist0[1:]:
            for j in range(2,608):
                if sh1.cell(row=j, column=21).value==id:
                    k2=22
                    flag = True
                    while not (sh1.cell(row=1, column=k2).value is None) and
flag:
                        if not (sh1.cell(row=1,
column=k2).value.rfind(course_id) == -1) or not (
                            course_id.rfind(sh1.cell(row=1,

```

```

column=k2).value) == -1):
    sh1.cell(row=j, column=k2).value = value
    flag = False
    k2 += 1
    if flag:
        sh1.cell(row=1, column=k2).value = course_id
        sh1.cell(row=j, column=k2).value = value

wb1.save(filename="C:/Abiroy/2d_form/temp/output_6.xlsx")

```

## 7 Скрипт:

```

import openpyxl
wb1=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output_6.xlsx")
wb2=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Факт
обучения ОО ОКО ИТ/Грязные данные/ОБ.xlsx")
sh1=wb1["Лист1"]
sh2=wb2["31.12.2017"]
for j in range(2, 608):
    for i in range(5, 460):
        if not (sh2.cell(row=i, column=12).value is None):
            str = sh2.cell(row=i, column=12).value.split()
            k1=2
            t = True
            for x in str:
                if (x is None) or (sh1.cell(row=j, column=k1).value is None):
                    t=False
                if t:
                    if sh1.cell(row=j, column=k1).value.rfind(x)==-1 and
x.rfind(sh1.cell(row=j, column=k1).value)==-1:
                        t=False
                    k1+=1
            #if
not(sh2.cell(row=i,column=3).value==sh1.cell(row=j,column=14).value):
            # t=False
            if t:
                if not (sh2.cell(row=i, column=15).value is None):
                    list=sh2.cell(row=i, column=15).value.split(",")
                    for x in list:
                        for k2 in range(22, 70):
                            if not ((sh1.cell(row=1,
column=k2).value.split()[0]+".").rfind(x+".")==-1):
                                sh1.cell(row=j, column=k2).value=="Пройден"
                                print(j, k2, "Пройден")
wb1.save(filename="C:/Abiroy/2d_form/temp/output_7.xlsx")

```

## 8 Скрипт:

```

import openpyxl
wb1=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/temp/output_7.xlsx")
wb2=openpyxl.load_workbook(filename="C:/Abiroy/2d_form/tables_input/Факт
обучения ОО ОКО ИТ/Грязные данные/ОО факт Икв 2018.xlsx")
sh1=wb1["Лист1"]
sh2=wb2["Лист1"]
for j in range(2, 608):
    for i in range(5, 977):
        if not (sh2.cell(row=i, column=5).value is None):

```

```

        if sh2.cell(row=i, column=5).value == sh1.cell(row=j,
column=21).value:
            if not (sh2.cell(row=i, column=11).value is None):
                list=sh2.cell(row=i, column=11).value.split(", ")
                for x in list:
                    for k2 in range(22, 70):
                        if not ((sh1.cell(row=1,
column=k2).value.split()[0]+".").rfind(x+".")!=-1):
                            sh1.cell(row=j, column=k2).value=="Пройден"
                            print(j, i, "Пройден", x)
wb1.save(filename="C:/Abiroy/2d_form/temp/Table2.xlsx")

```