СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Большие данные. Основные понятия |  |
| 1.1 Общие положения | 4 |
| 1.2 Приём данных | 5 |
| 1.3 Сбор данных | 5 |
| 1.4 Анализ данных | 6 |
| 1.5 Представление результатов | 7 |
| 1.6 Облачные технологии для работы с большими данными | 7 |
| 1.7 Поиск закономерностей или Data Mining | 8 |
| 1.8 Искусственный интеллект. Нейросети | 9 |
| 1.9 Технологии обработки больших данных | 10 |
| 2 Разработка информационно-справочного ресурса |  |
| 2.1 Описание программных средств, используемых для создания веб-сайта | 12 |
| 2.2 Проектирование навигационной структуры веб-сайта | 15 |
| 2.3 Разработка графического интерфейса информационно-справочного ресурса | 16 |
| 2.4 Верстка страниц информационно-справочного ресурса | 17 |
| 2.5 Тестирование работы информационно-справочного ресурса |  |
| Заключение |  |
| Список использованной литературы |  |
| Приложение А. Листинг презентации результатов учебной (ознакомительной) практики |  |

ВВЕДЕНИЕ

Целями учебной (ознакомительной) практики является закрепление знаний и умений, приобретенных обучающимися в течение первого учебного года, в результате освоения теоретических знаний, ориентированных на будущую профессиональную подготовку по выбранному профилю, приобретение им практических навыков и компетенций в области информационных систем и технологий.

Задачами учебной практики являются:

* закрепление знаний, полученных при теоретическом обучении, подготовка к изучению последующих профильных дисциплин;
* практическое использование полученных теоретических знаний и практических навыков путем разработки презентации и сайта по теме «Большие данные. Анализ и обработка»;
* совершенствование навыков решения информационных задач на конкретном рабочем месте.

Актуальность данной темы прежде всего связана с особенностью современной жизни человека, в частности с сильным внедрением цифровых технологий: например, крупные государственные организации вынуждены хранить большое число сведений о людях («Госуслуги»).

Таким образом, вытекает потребность в систематизации, анализе, обработки и хранения, так называемых, «больших данных».

Также стоит отметить актуальность данной темы с точки зрения прохождения курса обучения по специальности программиста, поскольку позволит дать более глубокое представление о том, какое программное обеспечение необходимо применять для работы с большим массивом данных.

1 БОЛЬШИЕ ДАННЫЕ. ОСНОВНЫЕ ПОНЯТИЯ

1.1 Общие положения

Исходя из названия, можно сделать вывод, что термин «большие данные» прежде всего означает анализ большого массива сведений. Тем не менее, в действительности, дело не столько в объёме данных, необходимых для обработки, сколько в разнообразном по своей структуре представлении: в виде журналов, сайтов, текстовых документах или машинных кодах. При этом данные постоянно обновляются, что приводит к невозможности использовать традиционные методы анализа информации.

При этом вдобавок к информации, возникающий естественным путём, существуют и данные современных информационных систем, которые не могут быть обработаны в связи с тем, что их анализ требует вычислительных ресурсов, существенно превышающих вычислительные ресурсы самой информационно системы. Для решения данной проблемы применяют Data Mining – поиск закономерностей из кучи пустых данных.

Массивы данных могут быть обработаны посредством следующих подходов [2]:

1. Данные преобразовываются из различных форматов в общий, допускающий выполнение запросов к единообразным данным;

2. Складирование данных в единое хранилище без изменения формата (JSON, XML, CSV), образуя несвязанный набор, называющийся «озером данных» (Data Lake);

3. Обработка потоковых данных посредство специализированных программных приложений (с использованием SQL-подобного синтаксиса).

Таким образом, «большие данные» – это огромные массивы данных или потоки, которые содержат информацию, требующие быстрой интерактивной обработки с целью исследования или проверки гипотез. Для реализации данных задач требуется высокий уровень параллелизма и большой объём оперативной памяти.

Большая российская энциклопедия [1] приводит следующее, но схожее по своему смыслу определение: «Большие данные […] системы хранения и обработки огромных массивов структурированных и неструктурированных [данных](https://bigenc.ru/c/dannye-83e10d) разнообразного состава, часто обновляемых и поступающих из различных источников, обрабатываемых, например, в целях увеличения эффективности бизнеса…».

1.2 Приём данных

Для обработки больших данных используются сложные системы, в которых можно выделить несколько этапов: приём, сбор, анализ данных и представление результатов [2].

Первым этапом является приём данных. Его задача заключается в базовой подготовке сведений с целью приведения данных к единому формату представления.

Исходные же данные определяются источниками информации, например, погодные наблюдения можно проводить разными способами (с помощью наблюдения или датчиков). Для эффективной работы применяется собственный сборщик (Data Crawler).

Затем выполняются преобразования систем измерения и типов, то есть обработка данных содержательно не затрагивает имеющуюся информацию, но может изменить её вид: например, трансформировать данные из полярной системы координат в прямоугольные или привести все значения к системе измерения СГС для проведения расчётов магнитных величин.

1.3 Сбор данных

Сбор данных – второй компонент обработки данных, характеризующийся взаимодействием с системами хранения данных.

Устанавливается точка сбора, где данные снабжаются мета-сведениями (техническая информация) с отправкой в хранилище для дальнейшей обработки. При этом данные, не прошедшие точку сбора, не учитываются.

Причём в зависимости от разной степени структурированности данные обрабатываются по-разному:

- если сведения структурированы, то происходит преобразование информации по заранее известны алгоритмам;

- если сведения полуструктурированы, то потребуется интерпретатор данных (механизм преобразования). Чаще всего такие данные снабжены метаданными для удобной конвертации. Например, использование битов чётности в служебных целях.

- если же сведение не структурированы вовсе, то может потребоваться создание специального софта вкупе с кропотливой ручной работой.

На этом этапе также устанавливается базовая проверка достоверности данных. В частности, отрицательный рост человека не может быть учтён, а скорость тела с ненулевой массой не может превышать скорость света.

Собранные данные далее передаются в системы анализа в режиме реального времени (или постфактум).

1.4 Анализ данных

В отличие от предыдущих этапов, на этапе анализа ценность имеет именно сама информация, содержащаяся в данных. Анализ данных является самой трудоёмкой частью в процессе обработки данных.

Анализ может проводиться как в режиме реального времени, так и в пакетном режиме.

Существует большое количество методик обработки данных: например, запросы и отчётность (используются в СУБД Microsoft Access или LibreOffice Base), реконструкция по математической модели и другие. Методики используют собственные специфические алгоритмы в зависимости от поставленных целей.

Важно также отметить, что использование данных искусственным интеллектом позволяет развивать системы динамического обучения.

По сути анализ данных – различные аналитические механизмы, которые применяют аналитические алгоритмы для управления моделями и идентификации сущностями для получения новой содержательной информации, являющаяся искомой в рамках поставленной задачи.

1.5 Представление результатов

Полученная информация представляется на уровне потребления пользователем, однако имеются некоторые механизмы, позволяющие применять результаты для конкретных практических задач:

- мониторинг данных, то есть отображение в реальном времени процессов обработки информации;

- генерация отчётов и запросов к данным, визуализация полученной информации в виде документов, таблиц или кратких справок;

- трансформация данных и экспорт в другие системы.

1.6 Облачные технологии для работы с большими данными

Долгое время существовала проблема с подбором вычислительных ресурсов, поскольку ранее требовалось либо заключать договор аренды дата-центров, который затем приходилось вместе соединять, либо создавать собственную серверную. Однако, такой подход был очевидно неудобен, поскольку не всегда было известно, на какой срок ставить аренду и как технически поддерживать сервера.

Облачные технологии позволили изменить это тем, что предложили пользователям использовать вычислительные ресурсы ровно на столько, на сколько им нужно:

- ресурсы предоставляются по требованию и таким же образом освобождаются;

- плата начисляется за фактическое время использование ресурсов;

- предоставление и освобождение ресурсов производятся пользователем через веб-портал без всякой бумажной волокиты.

Такие эластичные виртуальные машины позволяют их использовать не только в качестве хранилища, но и основывать на них собственные информационные системы, так называемые, «платформы как сервис» (Platform as a Service).

В настоящее время крупнейшими облачными провайдерами являются Amazon Web Services и Google Cloud. Последний в свою очередь позволяет создавать на собственной базе текстовые документы, презентации и реляционные таблицы.

1.7 Поиск закономерностей или Data Mining

Если у вас есть большой массив неструктурированных данных, то анализ подобной информации может не представяляться возможным, поскольку вычислительных мощностей может просто не хватать для выполнения задачи в разумное время. Отсюда вытекает необходимость систематизации данных, что обеспечит их анализ. Один из распространённых способов – Data Mining.

Рисунок 1 – Методы Data Mining

Основу методов дата-майнинга составляют всевозможные методы классификации и моделирования посредством дерева решений, искусственных нейронных сетей, генетических алгоритмов, эволюционного программирования, ассоциативной памяти и нечёткой логики (рис. 1). Рассмотрим некоторые из них.

Дерево решений – метод принятия решения, в основе которого лежит дерево, на ребрах которого аргументы целевой функции, в листьях – значения, а в прочих узла – признаки, позволяющие отличить случаи. Цель состоит в том, чтобы предсказать решение на основании нескольких переменных на входе.

Генетический алгоритм – метод решения задач, путём случайного подбора различных комбинаций.

Самым распространенным алгоритмом же является корреляционный анализ, при котором выясняется теснота связи между двумя и более переменными.

Метод принятий решения на основе нечёткой логики применяется в случае, если приходится исследовать рассуждения в условиях нечёткости, размытости или схожести с максимально общими рассуждениями.

Вышеописанные методы позволяют принимать решения в условиях отсутствия четких закономерностей.

1.8 Искусственный интеллект. Нейросети

В последнее время свою популярность обретают «нейросети» - искусственные нейронные сети, представляющие собой связку простых процессов, даже в сравнении с компьютерными процессами, каждый из которых выполняют функционал передачи сигнала другим процессорам.

Тем не менее подобная схема способна выполнять весьма сложные задачи.

Нейросети являются частным случаем искусственного интеллекта, а в данный момент приоритетным направлением для развития из-за возможности обучения. Данная способность имеет главное преимущество по сравнению с традиционными алгоритмами, поскольку в процессе самообучения нейронная сеть выявляет сложные зависимости между входными и выходными данными, на основании которых способна сделать обобщение.

1.9 Технологии обработки больших данных

Для работы с Big Data требуется собственный инструментарий, поскольку привычные способы работы с данными посредством классических СУБД непригодны. В частности, это хорошо отражает термин «горизонтальная масштабируемость».

Горизонтальная масштабируемость предполагает разбиение системы на тысячи и более вычислительных узлов без потери производительности.

Именно потеря производительности и не позволяет применять к большим данным классические алгоритмы и инструменты.

Большинство дата-сайентистов используют следующее программное обеспечение (рис. 2): Hadoop, NoSQL и язык R.

Рисунок 2 – Инструментарий Big Data

2 РАЗРАБОТКА ИНФОРМАЦИОННО-СПРАВОЧНОГО РЕСУРСА

2.1 Описание программных средств, используемых для создания веб-сайта

Информационно-справочный ресурс должен быть представлен в виде веб-сайта, содержащий ряд веб-страниц, каждая из которых должна иметь своё информационное значение.

Для реализации поставленных задач необходимо использовать средства верстки веб-страниц.

Рассмотрим один из классических, но и простых способов реализации веб-страницы – HyperText Markup Language (HTML). Дословно «язык гипертекстовой разметки».

Важно отметить, что язык HTML – не язык программирования, как многие ошибочно полагают. Этот язык был выбран на основании прохождения курса в университете, а также по причине его простоты.

Код HTML представляет собой набор элементов HTML, которые по своей сути являются строительными блоками. С помощью этих инструментов для отображения на экране применяют различные конструкции.

Язык HTML – язык тегов. Каждый элемент представляет собой этот самый тег, которые можно условно разделить на два вида по своему синтаксису:

- парные теги, например, <div>…</div> – тег блочной верстки;

- одиночные теги, например, <img href = “…”> – тег вставки изображения.

Любой тег может иметь обязательные или опциональные атрибуты, которые определяют какие-либо свойства. Например, ранее неписанный атрибут href = “…” позволяет указать абсолютный или относительный адрес к изображению.

Регистр написания HTML-код является не существенным, в отличие, например, от XHTML. Многие элементы могут быть вложены друг в друга, например, теги <div> часто вкладывают друг в друга для обеспечения грамотной блочной верстки.

Любая веб-страница прежде, чем отобразиться на экране пользователя должна пройти процесс интерпретации, при котором каждая строчка кода читается последовательно. Строчки кода считываются сверху вниз, из-за чего более поздние по написанию строчки могут перекрывать значение ранних.

Важно отметить, что процесс интерпретации происходит прежде, чем веб-страница полностью загрузится.

Необходимо также сказать и про особенность структуры HTML-кода. Любой код должен начинаться указания DOCTYPE, отвечающего за корректное отображение страницы.

Далее, в соответствии со стандартной структурой, идет <html>, внутри которого и прописывается содержание веб-сайта, ибо вне него браузер не читает информацию.

Вложенные дочерние элементы <head> отвечают за различные метаданные, такие как прикрепление таблицы стилей CSS, о которых речь пойдёт позже, и заголовок страницы <title>, и <body>, внутри которого и находится всё визуальное содержимое.

Таким образом, по своей сути HTML-документ представляет собой дерево тегов (рис. 3).

Рисунок 3 – Схематическое изображение дерева HTML

Однако, это не единственное средство разработки для реализации веб-сайта. Необходимо реализовать также и графический дизайн сайта. Для этого будут использован язык каскадных таблиц стилей CSS – Cascading Style Sheets.

Cascading Style Sheets (CSS) – язык таблицы стилей, который позволяет прикрепить различные стили, например, к HTML-документу, такие как шрифт, цвет, особенности верстки (flex-, или grid-системы) и так далее.

Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств.

Таблицы стилей прежде всего использует для ограждения и отделения описания логической структуры сайта от описания внешнего вида и дизайнерской структуры.

Это условное разделение позволяет увеличить доступность документа для самого программиста вёрстки, а также улучшает гибкость и возможность управления его представлением.

Рассмотрим синтаксис подключения CSS-файла или самих стилей к элементам. Отметим три основных способа:

- <link rel = “stylesheet” href = “…”> – подключение через <link> в теге метаданных;

- <div style = “…”> – подключение отдельно к каждому тегу;

- <style> p {…} </style> – подключение через стили в теге метаданных <head>.

Необходимо также сказать и про синтаксис в самом CSS-файле. Выглядит он следующим образом: селектор {свойство: значение;}, где под селектором подразумевается тег, к которому применяются свойства, например, базовый тег <div> или созданный пользователем класс, идентификатор.

Часто в роли селекторов могут выступать, например, так называемы селекторы псевдоклассов, например, a:hover, то есть при наведении на тег <a> компьютерной мышью будут применяться определённые свойства.

Важно поговорить про два фундаментальных понятия в CSS-верстке. Это наследование и каскадирование. Они тесно связаны между собой.

Рассмотрим первое. Под наследованием понимают передачу ряда свойств от родительского элемента к дочернему.

Например, CSS-стили, прописанные для тега <body>, будут наследоваться всеми элементами внутри него, которые могут наследовать определённые свойства.

Существует теги, неспособные к автоматическому наследованию, для этого предусмотрено свойство inherit – принудительное наследование.

Теперь рассмотрим каскадирование. Под каскадированием следует понимать механизм управления конечным результатом применения CSS-правил в случаях, когда к одному элементу применяются разные таблицы стилей.

Определяется каскадирование следующим образом:

- правило !important, повышающее важность того или иного свойства;

- специфичность, то есть применение свойств по важности селектора;

- порядок, в котором подключены таблицы стилей.

На основании всей вышесказанной теории и будет выполнена работа.

2.2 Проектирование навигационной структуры веб-сайта

Проектирование навигационной структуры информационно-справочного ресурса является самой важной частью в создании сайта, поскольку проект выступит планом структуры веб-сайта. Следование плану позволит упростить выполнение самой работы.

Рассмотрим основные этапы проектирования сайта.

- определение количества веб-страниц, из которых будет состоять веб-сайт;

- определение тематик каждой из веб-страницы;

- описание макета каждой веб-страницы.

В рамках выполнения отчёта по учебной практике необходимо сделать многостраничный веб-сайт. Было принято сделать семь страниц, из которых четыре основных: «Главная», «О нас», «Написать нам», «Статьи»; и три тематических – «Основные положения», «Обработка данных», «Технологии разработки».

Все страницы будут иметь навигационное меню, которое будет менять свой внешний вид в зависимости от размера экрана (на маленьких экранах будет использоваться интерактивное выпадающее меню).

Снизу, как и на всех страницах, будет находиться «подвал» сайта, с описанием авторства и года создания.

Каждая страница будет состоять из одного большого содержательного блока, разбитого на две части: в левой – изображение (подпись), в правой – текст. Однако, страница «Написать нам» будет иметь всего один элемент – способ связи.

Данный макет будет использоваться для всех веб-страниц.

На главной можно будет получить основную информацию о сайте; на странице «О нас» – об авторах; на странице «Написать нам» можно будет связаться с создателями; на странице «Статьи» можно будет получить доступ ко всем статьям на сайте по интерактивным элементам.

2.3 Разработка графического интерфейса информационно-справочного ресурса

Под созданием графического интерфейса информационного-справочного ресурса следует понимать создание графического макета сайта на основе описаний макетов из проектирования информационного ресурса.

Для всех страниц будет использован одинаковый макет (рис. 4). Но для страницы «Написать нам» разделения на два логических раздела не будет, поскольку будет единая форма для заполнения.

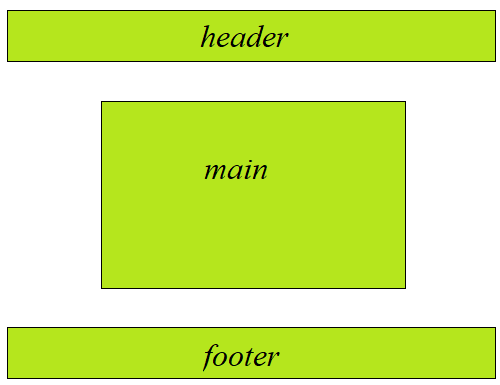


Рисунок 4 – Макет веб-страниц

Для всех веб-страниц будет использован единая «темная тема» из двух RGB-цветов: для фона – (51, 51, 51), для блоков – (0, 0, 0), но с прозрачностью в 50%.

Подобная цветовая схема позволит лучше воспринимать содержание информационно-справочного ресурса.

Цвет всех текстов будет белым, в определённых случаях будет иметь жирный шрифт.

2.4 Верстка страниц информационно-справочного ресурса

Под версткой необходимо понимать техническое создание веб-страниц, а также их содержательную часть.

Для логической реализации создадим три папки: images, pages, styles. (рис. 5).



Рисунок 5 – Логическое разделение реализации веб-сайта

В первую папку поместим все необходимые изображения для верстки сайта, во вторую – HTML-страницы, в третьей – таблицу стилей index.css.