СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Я.А. Бекенева

Информационные системы искусственного интеллекта в медицине

Методические рекомендации к практическим работам

СПбГЭТУ «ЛЭТИ», 2021 г.

1 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ НА ТЕМУ «ОБРАБОТКА МЕДИЦИНСКИХ ДАННЫХ»

1.1 Общие положения

1.1.1 Заголовок 3 уровня

Цель работы: освоение принципов преобразования сырых данных, получаемых от источников, к единому формату для дальнейшего хранения и анализа.

Как известно, данные, генерируемые различными источниками, имеют структуру и формат, отличные от принятых в информационных хранилищах. Зачастую данные могут иметь множество различных проблем, такие как пропуски в данных, разный формат хранения, неполнота, неточность, данные в буквальном смысле могут быть сломаны. Нередко для того, чтобы привести данные в требуемый вид, необходимо осуществлять их обработку и очистку.

Еще одной проблемой является различие форматов и структур данных, если используются разные устройства, регистрирующие разные параметры. В таком случае при сборе данных в ИС необходимо осуществить агрегацию данных, что подразумевает не только их объединение, приведение их к унифицированному формату, принятому в данной системе.

1.2 Выполнение работы

Практическое задание состоит из двух этапов.

1.2.1 Выполнение первого этапа

Первый этап — предварительная обработка и очистка данных.

Для выполнения этого задания используется часть данных из набора AACT, набор данных является одинаковым для всех студентов.

В исходной таблице находится информация о критериях включения пациентов в клинические испытания. Указаны минимальный и максимальные возраст пациентов, их пол, а также некоторые критерии

включения/исключения. Особенность данной таблицы заключается в том, что все данные вносили врачи и их ассистенты, следовательно, данные имеют достаточно высокую степень зашумления.

Данные включают в себя следующие атрибуты: id испытания (исследования), минимальный возраст, максимальный возраст, гендер и критерий включения/исключения пациента. Рассмотрим атрибуты, связанные с возрастом, а также диапазон значений этих атрибутов (Рис. 1).

```
       '47 Years', '69 Years', '90 Days', '34 Years', '35 Month
s',
       '71 Years', '31 Days', '16 Months', '36 Months', '23 Month
s',
       '39 Years', '48 Months', '32 Weeks', '5 Weeks', '64 Year
s',
       '42 Years', '365 Days', '36 Weeks', '13 Months', '29 Year
s',
       '78 Months', '85 Years', '49 Years', '43 Years', '11 Month
s',
       '26 Weeks', '22 Weeks', '7 Months', '74 Years', '10 Month
s',
       '68 Years', '40 Months', '33 Weeks', '161 Days', '63 Year
s',
       '11 Weeks', '28 Months', '29 Months', '35 Weeks', '49 Day
s',
       '55 Days', '44 Years', '35 Days', '16 Weeks', '120 Days',
'70 Days',
       '249 Days', '76 Years', '66 Months', '33 Months', '20 Week
s',
       '78 Years', '30 Weeks', '72 Months', '57 Years', '42 Week
s',
       '67 Years', '10 Weeks', '31 Weeks', '32 Months', '39 Month
s',
       '55 Months'], dtype=object)
```

Рис. 1

Из рисунка видно, что атрибут имеет текстовый формат, а значение включает в себя некоторое число и слово, обозначающее год, месяц, неделю или день. Для разных исследований различные единицы записи, то есть в одном исследовании у нас записаны года, в других — недели, в некоторых — дни. Для работы алгоритмов машинного обучения необходимо привести все эти данные к единому виду.

В качестве унифицированной единицы измерения выберем дни. Для этого напишем функцию, которая переводит данные текстовые строки в дни. Для этого будем использовать регулярные выражения. Напишем функцию, которая вычисляет по строке количество дней, которое в ней записано. Для этого используем регулярное выражение, которое преобразует строку на две части: в первую группу попадает число, а во вторую группу — единица измерения. Таким образом, применив функцию get_min_age, для 30 дней мы получим 30, для двух лет — 730, для None получим ноль (Рис. 2). Данную функцию следует применить ко всему столбцу.

```
In [3]: import re
        from datetime import timedelta
```

```
In [ ]: # Пишем функцию, которая вернет python объект timedelta, который может
        def get_min_age(value):
            if pd.isna(value):
                return timedelta(seconds=0)  # Если значение NaN, то значит мин

            pattern = re.compile(r'([0-9]+) (\w+)')
            match = re.match(pattern, value)

            number = match.group(1)  # Берем первую группу - число
            unit = match.group(2)  # Берем вторую группу - единица измерения

            convert_map = {  # будет переводить в дни
                "Years": 365,
                "Months": 30,
                "Weeks": 7,
                "Days": 1
            }
            days = int(number) * convert_map[unit]
            return timedelta(days=days)
```

Рис. 2

Если не учесть пограничные случаи, когда имеется ровно один год либо один день, может возникнуть ошибка. Запишем также в наш словарь (Рис. 3) данные случаи.

```
In [ ]: # Пишем функцию, которая вернет python объект timedelta, который может
        def get_min_age(value):
            if pd.isna(value):
                return timedelta(seconds=0)  # Если значение NaN, то значит мин

            pattern = re.compile(r'([0-9]+) (\w+)')
            match = re.match(pattern, value)

            number = match.group(1)  # Берем первую группу - число
            unit = match.group(2)  # Берем вторую группу - единицы измерения

            convert_map = {  # будет переводить в дни
                "Years": 365,
                "Year": 365,
                "Months": 30,
                "Month": 30,
                "Weeks": 7,
                "Week": 7,
                "Day": 1,
                "Days": 1
            }
            days = int(number) * convert_map[unit]
            return timedelta(days=days)
```

Рис. 3

Теперь атрибуты возраста приведены к единому виду (Рис. 4).

```
49985    6570 days
49986    8030 days
49987       0 days
49988    8030 days
49989    6570 days
49990       0 days
49991    6935 days
49992   18250 days
49993    6570 days
49994   16425 days
49995   18250 days
49996   14600 days
49997    6570 days
49998    6570 days
49999    6570 days
```

Рис. 4

Далее рассмотрим атрибут, обозначающий пол. Рассмотрим, какие значения у нас может принимать столбец gender_description. В качестве значений наиболее часто встречаются слова women, female, and, male, однако присутствуют и другие слова. Необходимо реализовать функцию,

которая будет определять, подходит ли данное испытание для мужчин или не подходит. Для этого также воспользуемся регулярными выражениями (Рис. 5). Следует заметить, что при осуществлении поиска по слову men в данной подстроке, то результат поиска будет также выдавать слово women, и в таком случае результат будет неверным. Регулярное выражение позволит исправить или не допустить такую ошибку. Применив данную функцию к рассматриваемому столбцу, можно определить, идет в записи речь о мужчинах или о женщинах.

```
In [16]: def is_appropriate_to_man(value):
             if pd.isna(value):  # Считаем, что там, где не указано ничего могут
                 return True
             man_patterns = [
                 r'(?!wo)man',  # man , но не начинающийся с wo
                 r'(?!wo)men',
                 r'(?!fe)male',  # аналогично, male, но не female
                 r'(?!fe)males',
                 r'boy',

                 r'prostate',  # Имея некоторое знание о домене, мы можем расшир
                 r'erectile'   # Например, зная, что предстательная железа есть т
             ]
             for pattern in man_patterns:
                 if re.search(pattern, value):
                     return True
             return False

In [ ]: is_for_man = df['gender_description'].apply(is_appropriate_to_man)
        is_for_man
```

Рис. 5

Теперь рассмотрим более детально на поле criteria и то, как оно выглядит. Видим, что оно представляет собой набор текста, не имеющего выраженной структуры, и для выделения нужной информации потребуются методы машинного обучения. Но при этом даже такой текст может быть некорректным: в нем могут быть битые специальные символы либо символы в безопасной форме, либо символы в виде HTML-тегов. Это будет негативно сказываться на работе методов машинного обучения. Следовательно, необходимо скорректировать этот текст. Следует осуществить проверку, встречаются ли в текстах спецсимволы из тегов HTML. Для этого найдем все тексты, которые содержат символ «меньше» (Рис. 6). Как видим, тексты, содержащие данный символ, имеются. Визуализируем их более детально. Также можно увидеть, что некоторые тексты содержат символ «больше», которые записан в виде HTML-тега.

```
5. cerebrovascular or symptomatic peripheral vascular d
isease;

6. heart disease class III or IV NYHA;

7. Estimated glomerular filtration rate (eGFR) ≤60 mL/m
in/1.73m 2 or serum creatinine
&gt; 1.5mg/dL in men or &gt;1.4mg/dL in women

8. Liver function enzymes higher more than two times th
e upper limit

9. Ongoing urinary tract infection

10. drug or alcohol abuse;

11. life expectancy &lt;3 yrs

12. blood pressure &gt;160/100 mmHg

13. Donation of blood to a blood bank, blood transfusio
n, or participation in a clinical
study requiring withdrawal of &gt; 400 mL of blood d
uring the 8 weeks prior to the
enrollment visit and at least 8 weeks thereafter

14. Women of child bearing potential who are unwilling o
```

Рис. 6

Для исправления будет использоваться библиотека ftfy (Рис. 7), которая в автоматическом режиме исправляет основные ошибки с кодировками и с текстами.

```
In [ ]: import ftfy

repaired_text = ftfy.fix_text(text)
print(repaired_text)
```

Рис. 7

Применив данную библиотеку к рассматриваемому тексту (Рис. 8), видим, что символы «больше» и «меньше» скорректированы.

```
12. blood pressure >160/100 mmHg

13. Donation of blood to a blood bank, blood transfusio
n, or participation in a clinical
study requiring withdrawal of > 400 mL of blood duri
ng the 8 weeks prior to the
enrollment visit and at least 8 weeks thereafter

14. Women of child bearing potential who are unwilling o
r unable to use an acceptable
method to avoid pregnancy for the entire study (estr
ogen and/or progesterone
treatment)

15. Women who are pregnant or breastfeeding

16. Patient with a history or current evidence of any co
ndition, therapy, laboratory
abnormality, or other circumstance which, in the opi
nion of the investigator or
coordinator, might pose an unacceptable risk to the
patient or interfere with trial
procedures
```

Рис. 8

Следующее, что необходимо скорректировать, — это Unicode. Unicode является удобным для человека, потому что содержит множество символов, но при этом неудобен для вычислительной машины, так как различные символы могут означать одно и то же, например, символ «1/2» означает

дробь «1/2». Чтобы убрать подобные разночтения, в Unicode можно применять различные методы нормализации текстов. Существует четыре формы нормализации:

NFC – форма нормализации канонической композицией;

NFD – форма нормализации канонической декомпозицией;

NFKC – форма нормализации совместимой композицией;

NFKD – форма нормализации совместимой декомпозицией.

Воспользуемся формой NFKD. Например, применив данную нормализацию к символу «1/2», получим данный символ в корректной форме. Применив данный метод нормализации текстов ко всему тексту, можно решить проблемы, связанные с Unicode-кодировкой.

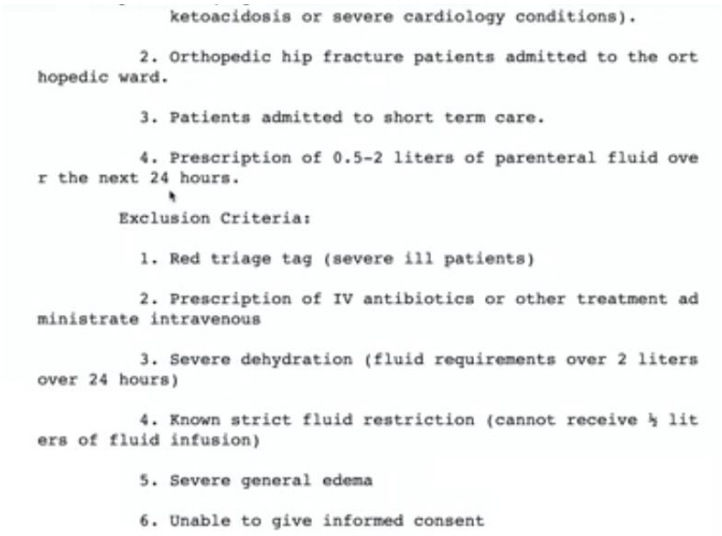

Рис. 9

1.2.2 Выполнение второго этапа

Второй этап – агрегация данных от источников и приведение их к единой структуре.

Задание выполняется каждым студентом самостоятельно. В качестве исходных данных используются наборы данных от нескольких разных источников, которые регистрируют разные параметры одного процесса.

Необходимо:

- проанализировать формат и структуру исходных данных;

- определить требования к дальнейшему использованию данных и характеру анализа;

- определить формат и структуру данных в системе;

- определить необходимые операции, которые необходимо произвести над данными;

- осуществить агрегацию исходных данных, выполнить обработку и очистку, привести данные к установленной структуре.

Требования к отчету

Отчет должен содержать:

- описание исходных данных;

- описание модели данных, хранимых в ИС;

- операции, выполненные для преобразования данных (схема процесса преобразования и описание операций);

- описание результата преобразования данных;

- выводы.

Контрольные вопросы.

1. Какова основная цель обработки данных, получаемых от источников?
2. Назовите существующие виды источников данных по отношению к ИС. Приведите примеры.
3. Приведите примеры источников данных для медицинских ИС.
4. Назовите основные операции, выполняемые при преобразовании сырых данных.
5. В чем смысл очистки данных?

2 РАЗРАБОТКА ИНФОРМАЦИОННОГО ХРАНИЛИЩА ДЛЯ МЕДИЦИНСКОЙ ИС

2.1 Общие положения

Цель работы: освоение методологических подходов проектирования информационных хранилищ для ИС.

Хранилище данных — предметно-ориентированный, интегрированный, неизменчивый, поддерживающий хронологию набор данных, организованный для целей поддержки принятия решений.

В основе концепции ХД лежит идея разделения данных, используемых для оперативной обработки и для решения задач анализа. Это позволяет применять структуры данных, которые удовлетворяют требованиям их хранения с учетом использования в OLTP-системах и системах анализа. Такое разделение позволяет оптимизировать как структуры данных оперативного хранения (оперативные БД, файлы, электронные таблицы и т. п.) для выполнения операций ввода, модификации, удаления и поиска, так и структуры данных, используемые для анализа (для выполнения аналитических запросов). В СППР эти два типа данных называются соответственно оперативными источниками данных (ОИД) и хранилищем данных.

Хранилище данных характеризуется большим объемом редко изменяемой информации. Хранилище данных обменивается информацией с другими хранилищами; как правило, речь идет о пакетной загрузке. Здесь решаются задачи обработки и хранения больших объемов информации (VLDB, Very Large Databases). Хранилище данных формирует сводное представление обо всех данных информационной системы.

В проектировании хранилищ данных важное место занимают интерфейсы обмена данными и интерфейсы конвертации данных. Следует четко выделить те системы, которые поставляют данные в хранилище, и те, что забирают данные из хранилища.

Внутренняя структура хранилища рассчитана на выполнение сложных запросов. Как правило, хранилище данных обслуживается выделенным сервером, а нередко - параллельным сервером баз данных. Хранилища данных используются для прогнозирования развития бизнеса. Они оснащаются средствами аналитической обработки данных (OLAP, On-Line Analytical Processing). На данный момент существуют специально ориентированные на OLAP версии серверов баз данных.

ПО хранилища баз данных состоит из следующих компонентов (рис. 2):

- ПО источников данных.

Здесь формируются входные пакеты данных, которые в состоянии интерпретировать конвертор и загрузчик данных. Это ПО входит в состав внешних систем (если повезло) или разрабатывается специально; в последнем случае основная его задача - достать данные из внешней системы.

- Загрузочная секция (отвечает и за трансформацию данных).

ПО принимает входные пакеты данных в необработанном формате, проверяет целостность пакета (например, контрольную сумму и т.п.), выполняет первичную обработку пакета (например, расшифровку), а также переводит данные во внутренний формат. Теперь они готовы для передачи в хранилище. Данные загрузочной секции хранятся во внутреннем формате системы, обеспечивающей пересылку данных.

- ПО трансформации данных.

Здесь обеспечиваются пересылка данных в хранилище и логика обработки данных. Данные проходит проверку на корректность, переводятся в нужный формат и интегрируются в хранилище.

- Собственно хранилище данных.

Преимущественно это один или несколько мощных параллельных серверов баз данных, которые обеспечивают обработку информации в хранилище. Само хранилище может быть распределенной среды (распределенная база данных) или трехуровневой среды (централизованная или распределенная база данных и сервер приложений). Собственно база данных хранилища редко представляет собой объединение рабочих баз данных. База данных не обязательно должна быть реляционной. Здесь могут с успехом использоваться расширения, предоставляемые современными СУБД, например Spatial Data-расширения, которые поддерживают многомерные данные.

- Интерфейс клиента.

ПО этого уровня обеспечивает взаимодействие приложений-клиентов, запрашивающих информацию из хранилища. Здесь запрещены операции модификации данных.

Все данные в ХД делятся на три основные категории (рис. 2.5):

детальные данные;

агрегированные данные;

метаданные.

2.2 Порядок выполнения

В рамках выполнения практической работы необходимо:

- определить архитектуру информационного хранилища;
- определить структуру данных информационного хранилища;
- осуществить преобразование предварительно обработанных данных для их приведения к выбранной структуре;
- описать логическую структуру метаданных хранилища данных.

Требования к отчету

Отчет должен содержать:

- описание выбранной архитектуры информационного хранилища;
- описание структуры данных в информационном хранилище;
- описание метаданных;
- выполненные операции по преобразованию данных;
- выводы.

Контрольные вопросы

1. Что такое хранилище данных? В каком виде данные размещаются в хранилище?
2. Какие архитектуры информационных хранилищ существуют?
3. Назовите компоненты ПО хранилищ баз данных.
4. Какие виды данных, размещенных в хранилище, существуют?
5. Что такое метаданные, для чего они используются?

3 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ НА ТЕМУ «ПОДВЕДЕНИЕ ИТОГОВ КУРСА».

3.1 Общие положения

Цель работы: общая презентация результатов практических работ, выполненных в рамках курса.

В рамках данного курса каждое практическое занятие представляет собой изучение особенностей разработки определенного компонента информационной системы.

Итоговое практическое занятие посвящено интеграции полученных результатов и представлению разработанной в рамках курса медицинской информационной системы.

3.2 Требования к отчетным материалам

Презентация, представляемая на итоговом занятии, включает в себя:

- описание источников данных и генерируемых ими данных;
- описание информационного хранилища;
- операции ETL;
- выполняемые виды анализа данных;
- заключение.

Контрольные вопросы.

1. Какова цель создания информационной системы для медицинского учреждения?
2. Для решения каких задач уместно использовать информационные и информационно-аналитические системы?
3. Какие виды анализа используются в информационно-аналитических системах?
4. Каковы условия превращения данных в знания?
5. В чем содержание организации и осуществления эксплуатации ИАС?