менеджмента качества. Материалы I международной научно-практической конференции «Информатизация общества», Астана қ., 2004 ж.
4. Куанов Т.Д.,Турганбаев Н.С., Абилкаева Ж.Н., Исмагулова Ф.Е. Organization of standard cubes and algorithms in TOFI technology. // Сборник материалов конференции «Application of Information and Communication Technologies-AICT2014». Астана қ., 2014 ж.
5. Габбасов М.Б. TOFI technology capabilities for data processing and visualization. //Сборник материалов конференции «Application of Information and Communication Technologies-AICT2014». Астана қ., 2014 ж.
6. Об утверждении методики расчетаи нормативов затрат на создание,развитие и сопровождение информационных систем государственных органов.Қазақстан Республикасының Инвестициялар және даму министрлігі(ҚР Президентінің 06.08.2014 ж. N 875 Жарлығымен құрылды)

УДК 519.25

ДЕРЕКТЕРГЕ ТАЛДАУ ЖҮРГІЗУ КЕЗЕҢДЕРІНЕ ШОЛУ ЖӘНЕ WEB SCRAPING

Темирболатова Арайлым Темирболатовна
atemirbolatova@mail.ru
Л.Н.Гумилев атындағы ЕҰУ Математикалық және компьютерлік модельдеу мамандығының 2-ші курс магистранты, Нұр-Сұлтан, Қазақстан
Ғылыми жетекшісі – Г. К. Абдрашева

Деректерді талдауды бірнеше кезеңдерден тұратын процесс ретінде сипаттауға болады, математикалық модель негізінде визуализация және болжам жасау үшін деректер түрлендіріледі және өңделеді. Деректерді талдау - бұл қадамдар тізбегі, олардың әрқайсысы кейінгі кезеңдер үшін маңызды рөл атқарады. Бұл процесс дәйекті, өзара байланысты қадамдар тізбегіне ұқсас:

- проблеманы анықтау;
- мәліметтер алу;
- деректерді дайындау - мәліметтерді тазарту;
- деректерді дайындау - мәліметтерді түрлендіру;
- деректерді зерттеу және визуализациялау;
- болжамды модель;
- үлгіні тексеру, тестілеу;
- орналастыру - нәтижелерді визуализациялау және түсіндіру;

Мәселені анықтау

Деректерді талдау процесі өңделмеген деректерді жинаудан басталады. Ол алдымен анықталып, содан кейін шешілуі керек проблемадан тұрады.

Мұны тек зерттелетін жүйеге: механизмге, қосымшаға немесе тұтастай процеске назар аудара отырып анықтауға болады. Зерттеу жүйенің жұмысын жақсы түсіну үшін жасалуы мүмкін, бірақ оны мінез-құлық қағидаларын түсініп, кейін болжау немесе таңдау жасауды (саналы түрде) ойластырған дұрыс.

Осылайша, міндет осы мәселені шешуге қатысты мәселелерді қарастыру болып табылады, әр түрлі қызығушылықтары бар мамандарды тауып, деректерді талдауға қажет бағдарламалық жасақтаманы орнату қажет.

Деректер алу

Мәселе анықталған кезде, талдаудың бірінші қадамы - деректерді алу. Оларды бір негізгі мақсат үшін таңдау керек - болжамды модель құру. Сондықтан деректерді таңдау сәтті талдау үшін де маңызды сәт болып табылады.

Интернет - бұл деректерді өндіруді бастау үшін жақсы орын. Бірақ олардың көпшілігін алу оңай емес. Барлық деректер файл немесе дерекқор ретінде сақталмайды. Олар HTML файлында немесе басқа форматта болуы мүмкін. Мұнда талдау әдісі құтқаруға келеді. Беттерде нақты HTML тегтерін іздеу арқылы мәліметтер жинауға мүмкіндік береді. Мұндай сәйкестіктер пайда болған кезде арнайы бағдарламалық жасақтама қажетті деректерді шығарады. Іздеу аяқталған кезде сізде талдауға қажет мәліметтер тізімі алынады.

Деректер дайындау

Талдаудағы барлық қадамдардың ішінде деректерді дайындау ең аз проблемалы қадам болып көрінеді, бірақ іс жүзінде деректерді дайындау үшін көп ресурстар мен уақыт қажет. Деректер көбінесе әртүрлі дереккөздерден жиналады, олардың әрқайсысы оны өз формасында немесе форматында ұсына алады. Олар талдау процесіне дайын болуы керек.

Деректерді дайындау келесі процестерді қамтиды:

- алу,
- тазалау
- қалыпқа келтіру,
- оңтайландырылған мәліметтер жиынына айналдыру.

Бұл, әдетте, жобалау кезеңінде жоспарланған осы әдістерге өте ыңғайлы кестелік форма. Көптеген проблемалар жарамсыз, анық емес немесе жоқ мәндер пайда болған кезде, қайталанатын өрістерде немесе жарамды интервалға сәйкес емес деректер туындаған кезде пайда болуы мүмкін.

Деректерді зерттеу / визуализация

Деректерді зерттеу - бұл модельдерді немесе қатынастарды іздеу мақсатында оларды графикалық немесе статистикалық ұсыну арқылы талдау. Визуализация - мұндай модельдерді бөлектеу үшін ең жақсы құрал.

Деректерді зерттеу жиналған ақпараттың түрі мен маңыздылығын түсіну үшін қажет алдын-ала зерттеуден тұрады. Мәселені анықтауда жиналған ақпаратпен бірге категорияға бөлу модельді анықтау үшін деректерді талдаудың қай әдісі қолайлы екенін анықтайды.

Бұл кезең диаграммаларды зерделеумен қатар келесі кезеңдерден тұрады:

- мәліметтерді жалпылау;
- деректерді топтау;
- әр түрлі атрибуттардың өзара байланысын зерттеу;
- заңдылықтар мен тенденцияларды анықтау;
- регрессиялық талдау модельдерін құру;
- классификация модельдерін құру;

Болжалды модель

Деректерді зерделеп болғаннан кейін деректер арасындағы қатынасты кодтайтын математикалық модельді жасау үшін барлық қажетті ақпаратқа ие боласыз. Бұл модельдер зерттелетін жүйені түсіну үшін пайдалы және екі жолмен қолданылады.

Біріншісі — жүйе құратын деректердің мәндері туралы болжамдар. Бұл жағдайда біз регрессиялық модельді қоданамыз .

Екіншісі — жаңа өнімдердің жіктелуі. Бұл қазірдің өзінде жіктеу модельдері немесе кластерлік талдау модельдері. Шын мәнінде, сіз модельдерді нәтиже түріне қарай бөлуге болады:

- Классификациялық модельдер: егер нәтиже сапалы өзгермелі болса.
- Регрессиялық модельдер: егер нәтиже сандық болса.
- Кластерлік модельдер: егер нәтиже сипаттамалық болса.

Бұл модельдерді құрудың қарапайым әдістеріне келесі әдістер жатады:

- сызықтық регрессия
- логистикалық регрессия
- классификация
- дерево решений
- k-ближайших соседей әдісі

Модельді тексеру

Модельді тексеру (валидациялау), яғни тестілеу кезеңі маңызды кезең болып табылады. Бұл бастапқы мәліметтер негізінде модельді сынауға мүмкіндік береді. Бұл өте маңызды, өйткені модель жасаған деректердің шынайылықпен салыстыру арқылы олардың сенімділігін анықтауға мүмкіндік береді. Бірақ бұл жолы сіз талдау үшін пайдаланылған бастапқы деректерді негізге ала аласыз.

Әдетте, үлгіні құру үшін деректерді қолданған кезде, сіз оларды мәліметтер жиынтығы ретінде, ал тексеру үшін – валидация деректері ретінде қабылдайсыз. Осылайша, модель жасаған және жүйе жасаған деректерді салыстыра отырып, қателіктерді бағалауға болады. Әр түрлі деректер жиынтығын қолдана отырып, жасалған модельдің сенімділік шегін бағалаңыз. Дұрыс болжанған мәндер белгілі бір ауқымда ғана сенімді бола алады немесе ескерілетін мәндер ауқымына байланысты сәйкестіктің әртүрлі деңгейлеріне ие болады.

Бұл процесс модельдің тиімділігін сандық бағалауға ғана емес, оны басқалармен салыстыруға да мүмкіндік береді. Бірнеше ұқсас әдістер бар; ең танымал - кросс-валидация (кросс-валидация). Ол жаттығуларды әртүрлі бөліктерге бөлуге негізделген. Олардың әрқайсысы өз кезегінде валидация ретінде пайдаланылады. Қалғанының бәрі жаттығу түрінде. Осылайша сіз біртіндеп жетілдірілетін модель аласыз

Web Scraping

Веб-қию, веб-жинау немесе веб-деректерді алу дегеніміз - бұл веб-сайттардан деректерді жүктеуге, талдауға және құруға арналған деректерді скраптау. Веб-скраб бағдарламалық қамтамасыздандыру гипермәтіндерді жіберу протоколының көмегімен немесе веб-шолғыш арқылы бүкіләлемдік Интернетке қол жеткізе алады.

Оның жұмысының жалпы принципін келесідей түсіндіруге болады: белгілі бір автоматтандырылған код мақсатты сайтқа GET сұрауларын орындайды және жауап ала отырып, HTML құжатын өңдейді, мәліметтерді іздейді және белгіленген форматқа өзгертеді.

Веб-скрепинг, веб-индекстеу, деректерді өндіру, бағалардың онлайн режимінде бақылау және бағаларды салыстыру, өнімдерді шолу (байқауды қарау үшін), жылжымайтын мүлік тізімдерін жинау, ауа-райының мәліметтері үшін қолданылатын қосымшалардың құрамдас бөлігі ретінде мониторинг, веб-сайттың өзгеруін анықтау, зерттеу, интернеттегі бар-жоғын және беделін бақылау, веб-сайттар мен веб-деректерді біріктіру.

Веб-скрепинг арқылы деректерді баптауға арналған көптеген бағдарламалық құралдар бар. Бұл бағдарламалық жасақтама парақтың деректер құрылымын автоматты түрде тануға тырысады немесе веб-скрепинг кодын қолмен жазу қажеттілігін жоятын жазба интерфейсін

немесе деректерды алу мен түрлендіруге қолданылатын кейбір сценарийлік функцияларды және сақтай алатын дерекқор интерфейсін қамтамасыз етуі мүмкін.

Веб-сайтты скраптауға арналған көптеген шешімдер бар. Олардың ішінде:

- API арқылы жұмыс жасайтын немесе веб-интерфейсі бар жеке қызметтер (Embedly, DiffBot және т.б.).
- Әр түрлі бағдарламалау тілдеріндегі ашық бастапқы жобалар (Goose, Scrapy - Python; Goutte - PHP; оқылым, Morph - Ruby).
- Сонымен қатар, өз шешіміңізді жазуға әрқашан мүмкіндік бар. Мысалы, Selenium кітапханасын қолдану (Python бағдарламалау тілі үшін).

Бұл тәжірибеде қалай жұмыс істейді?

Біз GET сұрауы бойынша HTML кодын алу механизмін дайындаймыз. Әрі қарай, біз мақсатты сайттың құрылымын қарастырамыз және бізді қызықтыратын ақпарат бар түйіндерді анықтаймыз. Осыдан кейін біз түйін өңдегішін жасаймыз және деректерді қалыпқа келтірілген түрде көрсетеміз.

Скрипинг жүйесі

Кірісінде біздің жүйе URL мекен-жайын алады, ал шыққан кезде ол қалыпқа келтірілген деректерді қайтарады (мысалы, JSON форматында).

Python-дағы HTTP: Сұраныс кітапханасы

Python коды веб-тазартудың интернеттен деректерді автоматтандырылған түрде алудың негізгі мақсаты. Яғни, Python бағдарламасын қолданып ғаламторды шарлайды. Python бағдарламасы HTTP құрылымын түсіне білуі керек. Бұл сұраныстарды тиімді әрі нақты түрде жүзеге асыратын Python кітапханалары бар.

Selenium - бұл веб-бетті тазартудың қуатты құралы, бастапқыда сайтты автоматты түрде тестілеу арқылы жасалған. Selenium веб-сайтты жүктеу, оның мазмұнын шығару және браузерді пайдалану кезінде пайдаланушының әрекеттеріне ұқсас әрекеттерді орындау үшін браузерлерді автоматтандыру арқылы жұмыс істейді. Сонымен қатар, бұл веб-беттерді тазартудың қуатты құралы. Seleniumді әртүрлі бағдарламалау тілдерінен басқаруға болады, мысалы Java, C #, PHP және, әрине, Python.

Seleniumнің өзінде веб-шолғыш жоқ екеніне назар аудару керек. Оның орнына үшінші тараппен өзара әрекеттесу үшін WebDriver деп аталатын үшінші тараптың интеграциялық бағдарламалық жасақтамасы қажет. Веб-драйверлер Chrome, Firefox, Safari және Internet Explorer сияқты көптеген заманауи шолғыштар үшін бар. Оларды пайдалану кезінде экранда шолғыш терезесі ашылады, және келесідей кодта көрсетілген қадамдар орындалады.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

driver = webdriver.Chrome('chromedriver.exe')
driver.get("https://krisha.kz/prodazha/kvartiry/nur-sultan/? das[who]=1")
```

Осы қысқа кодта не болатынын көрейік:

• Алдымен біз сұрау модулін импорттаймыз. Біз https://krisha.kz/prodazha/kvartiry/nur-sultan/? das[who]=1 веб-беттің мазмұнын аламыз

Берілген URL мекен-жайы бойынша «HTTP GET» бағдарламасын орындау үшін driver.get әдісін қолданамыз

Парақты жүктегеннен кейін HTML элементтерін аламыз. Seleniumнің әдістер тізімі сәл өзгеше көрінеді:

• find_element_by_id
• find_element_by_name
• find_element_by_xpath
• find_element_by_link_text
• find_element_by_partial_link_text
• find_element_by_tag_name
• find_element_by_class_name
• find_element_by_css_selector

-тағы find_all әдісі сияқты, бірінші сәйкес келетін элементтің орнына элементтер тізімін береді (Selenium-дегі WebElement нысандары түрінде ұсынылған).

Сондай-ақ, егер элемент табылмаса, жоғарыда аталған әдістер NoSuchElementException ерекшелігін қалдыратынын ескеру қажет.

Find_element_by_link_text әдісі элементтерді ішкі мәтінге сәйкес таңдайды. Find_element_by_partial_link_text әдісі бірдей, бірақ ішінара ішкі мәтінге сәйкес келеді. Олар көптеген жағдайларда пайдалы, мысалы, мәтіндік құндылыққа негізделген сілтемені табу үшін қолданады.

Find_element_by_name әдісі атрибуттың HTML атауына негізделген элементтерді таңдайды, ал find_element_by_tag_name нақты тег атауын қолданады. Beautiful Soup

Find_element_by_css_selector әдісі Beautiful Soup әдісіне ұқсас, бірақ сенімді CSS таңдау ережелерін талдаумен бірге келеді.

```
import random
class AllDataHouse:
    def __init__(self, getname, getsum, getyear):
        self.getname = getname
        self.getsum = getsum
        self.getyear = getyear
AllName = []
AllSum = []
AllYear = []

AllHouse = []

def datahouse(Name,Sum,Year):
    print("Done")
    indexelement = 0

    data = driver.find_elements_by_css_selector('div[class="a-card__inc"]')

    for getdata in data:
        indexelement = indexelement + 1
```

```
        getname = getdata.find_element_by_css_selector('div[class="a-card__header-left"]').text
        print(getname)
        Name.append(getname)

        time.sleep(random.randrange(4, 6, 3))

        getsum = getdata.find_element_by_css_selector('div[class="a-card__price"]').text
        print(getsum)
        Sum.append(getsum)

        getyear = getdata.find_element_by_css_selector('div[class="a-card__text-preview"]').text
        print(getyear)
        Year.append(getyear)
        time.sleep(random.randrange(4, 6, 3))

        alldatahouse = AllDataHouse(getname, getsum, getyear)
        AllHouse.append(alldatahouse)

    try:
        nextbutton = driver.find_element_by_css_selector('div[class="paginator__btn-text"]')
        if nextbutton:
            nextbutton.click()
            time.sleep(35)
            datahouse(AllName,AllSum,AllYear)
        else:
            return
    except:
        nextbutton = None

    time.sleep(random.randrange(2, 8, 1))
datahouse(AllName,AllSum,AllYear)
print(AllName)
#print(indexelement)
```

Қайтадан бір қадам жасап, не болатынын көрейік: параққа өткеннен кейін біз AllDataHouse классын құрып Name,Sum,Year яғни квартира бөлмесі саны, жылы, ауданы, бағасы бойынша find_elements_by_css_selector әдісін қолданып деректерді аламыз.

Қолданылған әдебиеттер тізімі

1. Бородин А. Н. Элементарный курс теории вероятностей и математической статистики. — СПб.: Лань, 2004. — 256 с.
2. Кибзун А. И., Горяинова Е.Р., Наумов А. В., Сиротин А. Н. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачами. — М.: ФИЗМАТЛИТ, 2002. — 224 с.
3. Айвазян С. А., Мхитарян В. С. Прикладная статистика. Основы эконометрики. Т.1. — М.: ЮНИТИ–ДАНА, 2001. — 656 с.

4. Айвазян С. А. Прикладная статистика. Основы эконометрики. Т.2. — М.: ЮНИТИ–ДАНА, 2001. — 432 с.
5. https://www.youtube.com/watch?v=vmEHCJofslg
6. https://proglib.io/p/web-scraping/
7. https://python-scripts.com/beautifulsoup-html-parsing

UDC 004.386

RESEARCH OF AUTOMATED DEVELOPMENT APPROACHES OF DOMAIN ONTOLOGIES

Anuar Turganbekov
turganbekovanuar@gmail.com

Master student, L. N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan

Supervisor – A. M. Kankenova

Introduction

Starting from the beginning of the Information age, humanity advanced its technologies in multiple domains and number of unstructured and unconnected data has being growing exponentially. Different domain experts and information sources could not follow standards and come to a consensus [1]. In order to solve this significant problem, the concept of the Semantic Web, a hypertext web extension, was introduced. It implies that any data can be shared and integrated within the World Wide Web, having a form understandable by both humans and machines [2, 3].

Ontologies are considered as a further implementation of the Semantic Web concept and a connecting link for information systems with different structures and applications. They define a semantic structure of terms, which describe a data, and relations between those terms. Ontologies also certify that data content is consistent, shared and understandable for both human experts and machines [2, 4].

Manual and automated development approaches exist for constructing ontologies from sets of data. The first approach is considered time and resource consuming, and so most of ontology developers switched to the latter. Automated development of ontologies, also known as ontology learning, is based on methods from information technologies fields such as natural language processing (NLP), machine learning, data mining, information retrieval and knowledge representation. In order for an unstructured data to transform into an ontology, an ontology developer should apply ontology learning methodology in order to process that data and further evaluate developed ontologies [1].

Ontology learning methodology

1. Linguistics methods

Linguistics methods adapt methodology of linguistics onto formal language representation of ontologies and they are used in conjunction with statistical methods and inductive logic programming. Linguistics include pre-processing, terms and concepts extraction, and relations extraction methods categories [1].