ДЕРЕКТЕРДІ ИНТЕЛЛЕКТУАЛДЫ ТАЛДАУ

Дәріс 14

JSON, XML және Excel форматтары

Оспан Ә.Ғ., PhD, доцент-зерттеуші

ҚазҰУ, Жасанды интеллект және Big Data кафедрасы

2026-2027 оқу жылы

Дәрістің мақсаты мен мазмұны

Мақсат және оқу нәтижелері

- Деректер мағынасына қарай сақтау құрылымын таңдау
- Ішкі жазбаларды тексерілетін кестелерге түрлендіру
- Жазу мен қайта оқудан кейінгі сақталуын растау

Дәріс мазмұны

1. **Формат және схема**
2. **JSON**
3. **Тізімдерді қалыпқа келтіру**
4. **XML ағашы**
5. **Excel және алмасуды тексеру**

Формат құрылым мен мақсатқа қарай таңдалады

Формат	Құрылым	Артықшылық	Тексеру
CSV	Бір жазық кесте	Қарапайым алмасу	Бөлгіш, кодтау, типтер
JSON	Нысандар мен массивтер	Ішкі құрылымы бар жазбалар	Кілттер, типтер, тізімдер
XML	Элементтер ағашы	Айқын белгілеу	Тегтер, атрибуттар, атаулар кеңістігі
Excel	Кітап, парақтар, ұяшықтар	Қарау және безендіру	Парақтар, тақырыптар, формулалар

Формат таңдау жазбаның мағынасы мен қолдану тәсілінен туындайды.

Схема мен тексеру сәтті оқудан маңыздырақ

1

Файлды талдау

Таңдалған парсермен синтаксисті тексеру.

2

Схема және типтер

Кілттерді, шектеулерді және айқын түрлендірулерді тексеру.

3

Қалыпқа келтіру және есеп

Жол бірлігін таңдап, талдау орындау.

4

Жазу және қайта оқу

Кілттерді, жазба санын және мәндерді салыстыру.

- Дұрыс синтаксис пәндік деректердің дұрыстығына кепіл емес.
- Мағынаны салыстыру файлдардың сөзбе-сөз сәйкестігінен пайдалырақ.

JSON және Python: негізгі типтердің сәйкестігі

JSON	Оқығаннан кейінгі Python	Нені ескеру керек
object	dict	JSON кілттері әрқашан жолдар
array	list	Элементтердің реті сақталады
string / number	str / int немесе float	Күтілетін типті тексеру
true / false / null	True / False / None	Бос және белгісіз мәндердің мағынасы әртүрлі

load/dump файлмен, loads/dumps жолмен жұмыс істейді.

Жоқ файл орнына толық жергілікті JSON

Python мысалы

```
from pathlib import Path
import json
import pandas as pd

out = Path('work_results')
out.mkdir(exist_ok=True)
recipes = [
    dict(id=101, cuisine='local', ingredients=['rice', 'onion']),
    dict(id=102, cuisine='med', ingredients=['cheese', 'onion']),
    dict(id=103, cuisine='local', ingredients=['rice']),
]
with (out/'recipes_demo.json').open('w', encoding='utf-8') as f:
    json.dump(recipes, f, ensure_ascii=False, allow_nan=False)
```

Жазба келісімі

- id — бірегей бүтін сан; cuisine — жол.
- ingredients — мәтіндік мәндер тізімі.

Қайталанушылық

- Деректер кодта толық берілген.
- Файл жасалған work_results ішіне жазылады.

Ішкі тізім талдау бірлігін таңдауды талап етеді

Ұзын пішін

- Бір жол — «рецепт — ингредиент» жұбы.
- Байланыс жиіліктері мен топтауға ыңғайлы.
- Қайталама жұптар рецепт санын бұрмалауы мүмкін.

Кең пішін

- Бір жол — рецепт; ing_* бағандарында 0/1.
- Индикаторлар қосындысы әртүрлі ингредиенттер санын береді.
- Префикс ингредиенттерді id мен cuisine-нен ажыратады.

Нені санайтыныңызды анықтаңыз: аталып өтулерді, байланыстарды әлде рецептерді.

Тізімді қалыпқа келтіріп, рецепт санын есептеу

Python мысалы

```
with (out/'recipes_demo.json').open(encoding='utf-8') as f:
    loaded = json.load(f)
table = pd.DataFrame(loaded)
long = table.explode('ingredients', ignore_index=True)
pairs = long[['id', 'ingredients']].drop_duplicates()
counts = pairs['ingredients'].value_counts().sort_index()
wide = pd.crosstab(pairs['id'], pairs['ingredients'])
wide = wide.gt(0).astype('int64').add_prefix('ing_')
print(counts.to_dict())
```

Нәтиже: {'cheese': 1, 'onion': 2, 'rice': 2}

Ұзын пішін

- explode әр элементке жеке жол жасайды.
- drop_duplicates әртүрлі рецептерді санауды белгілейді.

Кең пішін

- crosstab id мен ингредиент байланысын құрады.
- gt(0) жиіліктерді индикаторларға айналдырады.

Бір оқу жазбасының үш көрінісі

Көрініс	Бір жол нені білдіреді?	101-рецепт
JSON	Ішкі тізімі бар нысан	ingredients=['rice','onion']
Ұзын кесте	id мен ингредиент байланысы	101-rice; 101-onion
Кең кесте	Индикаторлары бар рецепт	ing_rice=1; ing_onion=1; ing_cheese=0
Ретті сақтау	Бөлек позицияны талап етеді	Индикаторларда рет сақталмайды

Мағына айқын берілгенде ғана жоғалтусыз түрлендіруге болады.

Кері түрлендіру деректер сақталуын тексереді

1

ing_* таңдау

id және cuisine метадеректері ингредиенттер емес.

2

Тізімдерді қалпына келтіру

Индикаторы 1 болатын баған атауларын қосу.

3

Жазу және оқу

Бастапқы JSON схемасын пайдалану.

4

Мағынаны салыстыру

id, cuisine және ингредиенттер жиындарын тексеру.

- Шегіністер мен кілттер реті мәтіндік теңдікті қажет етпейді.
- Ингредиент реті маңызды болса, элемент позициясын сақтаңыз.

XML: түбір, еншілес элементтер және атрибуттар

ElementTree түйіні

- tag — атау; text — мәтіндік мән.
- attrib түйін атрибуттарының сөздігін сақтайды.
- Еншілес элементтер бір түбірлі ағаш құрайды.

Схема бойынша шығару

- findtext еншілес элементті атауы бойынша іздейді.
- int/float мәтінді санға айқын түрлендіреді.
- Ішкі құрылым мен атаулар кеңістігін елемеуге болмайды.

Әр XML-дың бір жазық кесте түріндегі табиғи көрінісі бола бермейді.

Сенімді жергілікті XML мәзірін жасау

Python мысалы

```
import xml.etree.ElementTree as ET
root = ET.Element('menu')
for name, price, weight in [('Салат', 450, 180),
                            ('Күріш', 600, 250)]:
    dish = ET.SubElement(root, 'dish', name=name)
    ET.SubElement(dish, 'price').text = str(price)
    ET.SubElement(dish, 'weight').text = str(weight)
ET.ElementTree(root).write(out/'menu_demo.xml', encoding='utf-8')
parsed = ET.parse(out/'menu_demo.xml').getroot()
menu = pd.DataFrame([
    dict(name=e.get('name'), price=int(e.findtext('price')),
         weight=int(e.findtext('weight'))) for e in parsed])
print(menu)
```

Жасау

- Element түбірді, SubElement еншілес түйіндерді береді.
- name атрибуты мәтіндік price және weight-тен өзгеше.

Оқу

- ET.parse жергілікті файлдың ағашын қайтарады.
- Сандық өрістер схема бойынша int типіне келтіріледі.

Excel: екі парақ және міндетті қайта оқу

Python мысалы

```
book = out / 'format_demo.xlsx'
with pd.ExcelWriter(book, engine='openpyxl') as writer:
    table[['id', 'cuisine']].to_excel(
        writer, sheet_name='recipes', index=False)
    menu.to_excel(writer, sheet_name='menu', index=False)
    sheet = writer.sheets['menu']
    sheet.freeze_panes = 'A2'
    sheet.auto_filter.ref = sheet.dimensions
read_menu = pd.read_excel(
    book, sheet_name='menu', engine='openpyxl')
pd.testing.assert_frame_equal(menu, read_menu)
print('Мәзір деректері сақталды')
```

Нәтиже: Мәзір деректері сақталды

Жазу

- Контекст кітапты сақтауды аяқтайды.
- Әр параққа нақты атау беріледі; index=False.

Тексеру

- read_excel таңдалған парақты оқиды.
- Алмасудан кейін мәндер мен типтерді салыстырамыз.

Инварианттар ненің сақталғанын көрсетеді

Нысан	Инвариант	Тексеру тәсілі
Рецептер	Бірегей id және сол байланыстар	id -> cuisine және ингредиенттер жиыны
Ұзын/кең пішіндер	Әртүрлі ингредиенттер саны	Рецепт бойынша индикаторлар қосындысы
XML мәзірі	Тағам саны, типтер, баға және масса	Қайта оқу және айқын түрлендірулер
Excel	Парақтар мен кестелік мәндер	read_excel және DataFrame салыстыру

Файлды сәтті жазу — тексерудің соңы емес, басы.

Деректер алмасудағы қателер

Талдау

- JSON үшін eval қолданбау.
- Сәтті parse-ті схема тексеруі деп санамау.

Түрлендіру

- Кілттерді, атрибуттарды және маңызды ретті жоғалтпау.
- Рецепт саны мен аталып өту санын шатастырмау.

Сақталу

- Безендіруді өлшем бірліктері деп қабылдамау.
- Қайта оқу мен инварианттарды өткізіп алмау.

Дәрістің негізгі нәтижелері

1. Формат жазба құрылымы мен алмасу мақсатына қарай таңдалады.
2. JSON десериализациядан кейін күтілетін схеманы тексеруді талап етеді.
3. Ұзын және кең пішіндердің жол бірліктері әртүрлі.
4. XML атрибуттары, мәтіні, түйіндері мен реті айқын келісімді талап етеді.
5. Excel және басқа нәтижелер қайта оқу және инварианттармен тексеріледі.

Түсінуді тексеру және өзіндік жұмыс

1. json.load қашан тізім орнына скаляр қайтарады?
2. Ингредиенттердің ұзын және кең кестелерінің жолдары несімен ерекшеленеді?
3. Неліктен әр XML жоғалтусыз бір кестеге айналмайды?
4. sheet_name=None кезінде read_excel не қайтарады?
5. JSON мәтіндері сәйкестігінен деректердің қандай қасиеттері маңыздырақ?

Өзіндік жұмыс

Бөлек тәжірибеге қайталама onion және бос тізім қосыңыз. Өңдеу саясатын беріп, аталып өту санын рецепт санымен салыстырыңыз.

Құжаттама және бастапқы материалдар

Құжаттама

1. Python: JSON

https://docs.python.org/3/library/json.html

2. Python: ElementTree

https://docs.python.org/3/library/xml.etree.elementtree.html

3. pandas: ExcelWriter

https://pandas.pydata.org/docs/reference/api/pandas.ExcelWriter.html

Дәрістің негізі

- Берілген 14-дәрістің JSON, XML және Excel жөніндегі презентациясы
- Берілген 14-дәрістің қосымша презентациясы
- Берілген 14-зертханалық жұмыстың материалы
- Берілген 14-зертханалық жұмыстың жауаптары бар материалы