

Дәріс 3. Деректерді визуализациялау

Оқытушы: Сарсембаева Т.С.



Дәрістің мақсаты

Деректерді визуализациялаудың негізгі тұжырымдамасымен және оның деректерді талдау процесіндегі маңыздылығымен таныстыру. Содан кейін, Python-дағы визуализацияның негізгі кітапханасы ретінде Matplotlib-тің негіздерін, оның Pyplot модулін және негізгі функцияларын (plot, title, legend, grid, show) меңгерту көзделеді.

Сондай-ақ, дәріс Matplotlib және Seaborn кітапханаларының артықшылықтары мен кемшіліктерін салыстыруға және тиімді графикалық бейнелеудің негізгі принциптерін (айқындық, қарапайымдылық, салыстырмалылық) түсіндіруге бағытталған.

Негізгі сұрақтар

1. Деректерді визуализациялау дегеніміз не және оның мақсаты қандай?
2. Matplotlib дегеніміз не және оның Pyplot модулі қалай жұмыс істейді?
3. Matplotlib және Seaborn кітапханаларының негізгі айырмашылықтары неде?
4. Тиімді деректерді бейнелеудің негізгі принциптері қандай?

Қысқаша мазмұны (тезистер)

Деректерді визуализациялау

Деректерді визуализациялау — бұл ақпаратты графикалық түрде, яғни жай көзбен түсінуге ыңғайлы етіп ұсыну процесі. Оның негізгі мақсаты - күрделі деректерді адам миы тез қабылдай алатын пішінге түрлендіру. Визуализация деректер мен білім арасындағы "көпір" болып табылады. Ол деректер арасындағы қатынастар мен тенденцияларды, сондай-ақ аномалияларды (шеткі мәндерді) анық байқауға және шешім қабылдау үдерісін жеделдетуге көмектеседі.

Matplotlib кітапханасы

Matplotlib — бұл Python деректер визуализациясы үшін ең кең таралған және жиі пайдаланылатын кітапхана. Оның ең танымал ерекшелігі - икемділігі; көптеген параметрлерді (түстер, сызық стильдері, белгілер) баптауға болады. Pyplot – бұл барлық графиканы жасау үшін қолданылатын 'негізгі' модуль. Ол MATLAB интерфейсіне ұқсас командаларға негізделеді.

Негізгі функциялары: `plt.plot()` (деректер негізінде сызық сызады), `plt.title()`, `plt.xlabel()`, `plt.ylabel()` (тақырып пен ось атауларын орнатады), `plt.legend()` (аңыз қосады), `plt.grid(True)` (тор сызықтарын көрсетеді) және `plt.show()` (графикті экранға шығарады). Matplotlib әртүрлі графиктерді қолдайды: сызықтық, бағаналы, нүктелік, дөңгелек, гистограммалар, `boxplot`, контурлық және т.б.

Matplotlib және Seaborn

Python-да визуализация үшін кеңінен қолданылатын екі негізгі кітапхана – Matplotlib және Seaborn. Matplotlib – бұл негіз бен икемділік. Ол графиктерді нөлден салуға мүмкіндік береді және толық бақылауды қамтамасыз етеді. Кемшілігі: әдемі график үшін өте көп код жазу керек және безендіруді қолмен жасау қажет.

Seaborn – стилистика және статистикалық тұрғыдан графиктер құру үшін қолданылады. Бұл Matplotlib негізінде жасалған жоғары деңгейдегі кітапхана. Seaborn Matplotlib-тің қиын параметрлерін автоматты түрде өзіне алып, тартымды және түсінікті көріністер жасауға мүмкіндік береді. Ол статистикалық мүмкіндіктермен (мысалы, `regplot`, `boxplot`) бірге келеді және тікелей Pandas DataFrame-пен жұмыс істей алады.

Деректерді визуализациялау

Деректерді визуализациялау - бұл ақпаратты графикалық түрде, яғни жай көзбен түсінуге ыңғайлы етіп ұсыну процесі. Оның негізгі мақсаты - күрделі деректерді, сандар және кестелермен көрсетілген деректерді адам миы тез қабылдай алатын пішінге түрлендіру. Басқа сөзбен айтқанда, визуализация деректер мен білім арасындағы "көпір" болып табылады.

Зерттеу деректері жиі көлемді және құрылымдық жағынан күрделі болып келеді. Егер біз оларды тек қағаз үстінде немесе кестеде салыстырсақ, олардың бір-бірімен көп ортақ нәрсесі жоқ сияқты көрінеді. Басқа жағынан, бұл өрнектер графиктер және диаграммалар арқылы көрнекі және концептуалды түрде түсірмеледі. Мысалы, уақыттық қатары сызықтық диаграмма көрсеткіштің мәнінің уақыт аралығында қалай өсетінін немесе төмендейтінін көрсете алады, ал шашыраңқы диаграмма екі айнымалы арасындағы қолданылғаннан қазіргі қатынасты шұғыл анықтай алады.

Визуализация қазір деректерді талдау үдерісінің соңғы маңызды кезеңі ретінде кеңінен танылды. Бұл кез келген аналитикалық қорытынды сапасы тек есептеудің дәлдігіне ғана емес, сонымен қатар нәтижелерді дәл жеткізе білу мүмкіндігіне де байланысты болғандықтан.

Қатынастар мен тенденциялар

Деректер арасындағы қатынастар мен тенденцияларды анық байқай алады

Аномалияларды анықтау

Аномалияларды және экстремалды (шеткі) мәндерді анықтай алады

Оңай оқу

Оқуға оңай стильмен теңдестірілген шолуды ұсынады

Шешім қабылдау

Шешім қабылдау үдерісін жеделдетеді

Сонымен қатар, визуализацияның психологиялық аспектісі бар: адамдар визуал ақпаратты басқа модальдылықтарға қарағанда тезірек және анығырақ түсінеді. Нәтижесінде, инфографикалар, интерактивті бақылау тақталары және визуалды есептер қазіргі ғылымдар мен бизнестің құрылымының бөлігіне айналды.

Осылайша, деректер визуализациясы тек «нәтижелерді көрсету тәсілі» ғана емес, сонымен қатар аналитикалық ойлау тәсілі, ашық байланыс тілі және ақпаратты ұсынуға арналған тиімді құрал болып табылады.

Matplotlib

Matplotlib — бұл Python деректер визуализациясы үшін ең кең таралған және жиі пайдаланылатын кітапхана. Ол ғылым немесе бизнес болсын, кез келген есептеу кеңістігіндегі кез келген диаграмманы жасауға мүмкіндік береді. Бұл кітапхананы қарапайым сызықты графикадан бастап терең көпөлшемді болжау деректерін визуализациялауға дейін жасау үшін пайдалана аламыз. Оның ең танымал ерекшелігі икемділігі, көптеген параметрлер бар: түстер, сызық стильдері, белгілер, тақырыптар.

Pyplot модулі

Pyplot: бұл барлық графиканы жасау үшін қолданылатын 'негізгі' модуль. Ол графикаларды интерактивті етуге арналған төмен деңгейлі құралдар жиынтығын ұсынады. Бұл пакет MATLAB интерфейсіне ұқсас командаларға негізделеді және бірнеше код жолдары арқылы деректерді тез алдын ала қарау мүмкіндігі бар.

Төмендегі қосымша жолдар біз қалаған визуализацияны күрделі визуалды мәліметтерді (мысалы, осьтерді үйлестіру, сурет өлшемдерін орнату немесе координаталық жүйелерді конфигурациялау) өзіміз өңдемей-ақ жасайтын сенімді кодты қамтиды. Pyplot модулі визуализацияларды жасауға арналған 20 мыңдық көзқарасты көрсетіп отыр – бұл әрқашан бәрін егжей-тегжейлі түсіндіріп беру қажет емес суретші.

```
import matplotlib.pyplot as plt

# Мәліметтер
x = [0, 1, 2, 3, 4]
y = [0, 2, 4, 6, 8]

# График салу
plt.plot(x, y, label='y = 2x', color='red', marker='s', linestyle='-')

# Қосымша параметрлер
plt.title('Pyplot арқылы салынған график')
plt.xlabel('X осі')
plt.ylabel('Y осі')
plt.legend()
plt.grid(True)
plt.show() # Нәтижені көрсету
```

- **plt.plot()** — деректер негізінде сызық сызады
- **plt.title(), plt.xlabel(), plt.ylabel()** — графиктің тақырыбы мен ось атауларын орнатады
- **plt.legend()** — аңыз (legend) қосады
- **plt.grid(True)** — тор сызықтарын көрсетеді
- **plt.show()** — графикті экранға шығарады



Бір графикте бірнеше сызық

Кей жағдайда бірнеше деректердер жиынтығын бір графикте салыстыру қажет болады.

```
x = [0, 1, 2, 3, 4]
y1 = [n**2 for n in x]
y2 = [n**3 for n in x]
```

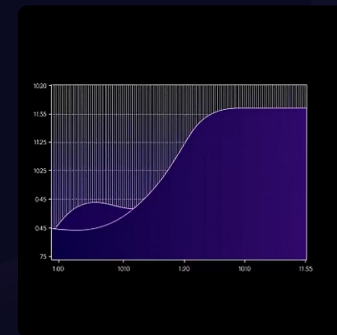
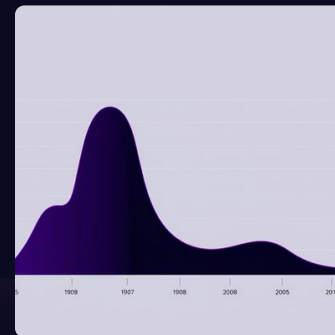
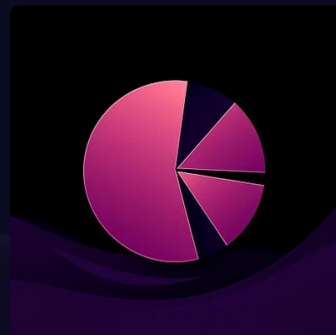
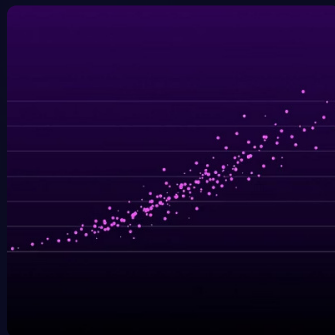
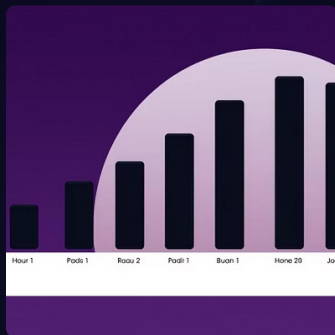
```
plt.plot(x, y1, label='y = x^2', color='green')
plt.plot(x, y2, label='y = x^3', color='orange')
plt.legend()
plt.title('Бір графикте бірнеше сызық')
plt.show()
```

plt.legend() функциясы әр жолдың нені білдіретінін көрсетеді.

Matplotlib графиктерінің түрлері

Matplotlib әр түрлі форматтағы визуализацияны қолдайды:

1. Сызықтық графиктер – трендтерді (тенденцияларды) көрсету үшін
2. Бағаналы диаграммалар - категориаларды салыстыру үшін
3. Нүктелік диаграммалар - корреляцияларды анықтау үшін
4. Дөңгелек диаграммалар - үлесті көрсету үшін
5. Гистограммалар - деректердің таралуын бейнелеу үшін
6. Boxplot және Violin plot - деректердің статистикалық таралуын көрсету үшін
7. Сабақ-жапырақ диаграммалары (Stem plots) – дискретті деректерді көрсету үшін
8. Градиент өрістері (Quiver plots) - векторлық өрістерді немесе бағытталған шамаларды көрсету үшін
9. Контурлық графиктер (Contour plots) - екі өлшемді кеңістіктегі функция мәндерінің деңгей сызықтарын көрсету үшін
10. Спектрлік диаграммалар (Spectral plots) - сигналдардың жиілік сипаттамаларын көрсету үшін



Matplotlib және Seaborn

Python тілінде мәліметтерді визуализациялау үшін кеңінен қолданылатын екі кітапхана бар, олар - Matplotlib және Seaborn. Екеуінің де графикалық шығысы бар, алайда қолдану және функция деңгейі әртүрлі.

Matplotlib

Негіз бен икемділік. Графиктерді нөлден салуға мүмкіндік береді және толық бақылау береді. Әрбір ось, түс, белгі және атау жеке түрде баптай аламыз.

Артықшылықтары:

- Кез келген үлгідегі графиктерді қолдайды
- Толығымен икемді - әр элемент жеке өңделеді
- NumPy және Pandas-пен жақсы біріктірілген

Кемшілігі:

- Өте көп код, безендіру қолмен жасалуы керек

Seaborn

Стилистика және статистикалық тұрғыда графиктер құру үшін қолданылады. Бұл Matplotlib негізінде жасалған жоғары деңгейдегі кітапхана. Seaborn мәліметтерді талдау және статистикалық графиктерді оңай жасауға арналған.

Артықшылықтары:

- Стилдендірілген графиктер автоматты түрде құрылады
- Статистикалық мүмкіндіктермен бірге келеді
- Тікелей Pandas DataFrame-пен жұмыс істейді
- Классикалық түстер палитралары мен тақырыптарды қолдайды

```
import matplotlib.pyplot as plt
import seaborn as sns

# Seaborn стиль орнату
sns.set(style="whitegrid")

# Үлгілік деректер жиынтығын жүктеу
tips = sns.load_dataset("tips")

# Нүктелік диаграмма (scatter plot)
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="day", palette="coolwarm")

# Тақырып пен белгілер
plt.title("Seaborn арқылы визуализация", fontsize=18)
plt.xlabel("total_bill")
plt.ylabel("tip")
plt.show()
```

Бейнелеудің негізгі принциптері

1

Мағыналық айқындық

Визуализацияның әрбір элементі белгілі бір мағынаға ие болуы керек. Түстерін, белгілерін, масштабтарын кездейсоқ таңдауға болмайды — олар деректердің мазмұнын дәл көрсетуі керек.

2

Қарапайымдылық және минимализм

Артық элементтер мен күрделі фондар ақпаратты қабылдауды қиындатады. Графиктің негізгі идеясы бір қарағанда түсінікті болуы керек.

3

Салыстырмалылық

Визуализация көрерменге деректер арасындағы айырмашылықты салыстыруға мүмкіндік беруі керек. Ол үшін бірдей масштабтарын, пропорцияларын және түс диапазондарын сақтау қажет.

4

Түс пен контрастты тиімді қолдану

Түс - визуализацияның ең қуатты байланыс құралы, бірақ оны шамадан тыс пайдалану ақпаратты шатастырады.

5

Контекст және аннотация

Кез-келген график түсіндірмесіз (legend, title, label) ақпаратсыз болып саналады. Сондықтан визуализацияда қосымша мәтіндік түсініктемелер болуы керек.

6

Пропорция және масштаб

Деректердің графикалық көрінісі олардың сандық қатынасына сәйкес келуі керек. Егер бейнелеуде масштаб бұрмаланса, көрермен қате қорытынды жасай алады.

7

Статистикалық дәлдік

Визуализация әрқашан деректердің нақты мазмұнын көрсетуі керек. Қате біріктіру немесе сзу нәтижелерді түсіндіруді бұрмалайды.

8

Эстетикалық тепе-теңдік

Жақсы визуализация эстетика мен ақпараттылықтың үйлесімінен тұрады. Құрылымдық тепе-теңдік (туралау, интервал, қаріп өлшемі) графикті кәсіби және оқуды жеңілдетеді.

Дәрісті меңгеруге арналған бақылау сұрақтары:

1. Деректерді визуализациялау дегеніміз не және оның шешім қабылдау процесіндегі рөлі қандай?
2. `matplotlib.pyplot` модулінің негізгі міндеті қандай?
3. Matplotlib-те `plt.plot()`, `plt.title()`, `plt.legend()` және `plt.show()` функциялары қандай қызмет атқарады?
4. Matplotlib пен Seaborn-ның негізгі айырмашылығы неде? Статистикалық графиктер үшін қайсысы ыңғайлырақ?
5. Тиімді визуализацияның негізгі 3 принципін атаңыз (мысалы, "Қарапайымдылық және минимализм").

Әдебиеттер тізімі:

Nie, S. (2024). Review on Data Visualisation Techniques. Applied and Computational Engineering. DOI: 10.54254/2755-2721/99/20251765

Data Visualization Techniques: Traditional Data to Big Data. In Big Data Analytics and Computing. Springer, 2021. DOI: 10.1007/978-981-15-2282-6_4

Нафлик, К. (2015). Данные: визуализируй, расскажи, используй. — Москва: Альпина Пабlishер.

IBM. Python for Data Science, AI & Development. — Онлайн курс, Coursera платформасында. Қолжетімді: <https://www.coursera.org/learn/python-for-applied-data-science-ai?>

OTUS. Matplotlib: от А до Я. — OTUS ресми блогы.
Қолжетімді: <https://otus.ru/journal/matplotlib-ot-a-do-ya/>