

ЛАБОРАТОРНА РОБОТА 5

Тема: Управління розмірністю даних

Під час аналізу даних часто зустрічаються ситуації, коли в базі даних міститься велика кількість змінних. Навіть коли початкова кількість змінних невелика, цей набір швидко розширюється на етапі підготовки даних, де створюються нові похідні змінні. У таких ситуаціях імовірно, що підмножини змінних сильно корелюють одна з одною. Включення змінних з високою кореляцією до моделі класифікації чи прогнозування або включення змінних, які не мають відношення до результату, що цікавлять, може призвести до зменшення точності та надійності прогнозу. Велика кількість змінних також створює обчислювальні проблеми для деяких контрольованих і неконтрольованих алгоритмів. Під час розгортання моделі зайві змінні можуть збільшити витрати за рахунок збору та обробки цих змінних. Тому для ефективної роботи алгоритмів аналізу даних часто є необхідним зменшення розміру набору даних, який представляється кількістю змінних. Цей процес є частиною фази пілотного (прототипного) аналізу даних і виконується перед розгортанням моделі. Існує кілька підходів щодо зменшення розмірності даних, серед яких слід виділити:

- включення знання предметної області для видалення або об'єднання категорій;
- використання підсумків даних для виявлення перекриття інформації між змінними і видалення або комбінування зайвих змінних або категорій;
- використання для модифікації даних таких методів, як трансформація категорійних змінних у числові змінні;
- використання автоматизованих методів скорочення, таких як аналіз головних компонентів (PCA), де створюється новий набір змінних, які є середньозваженими вихідними змінними.

Ці нові змінні некорельовані і їх невелика підмножина зазвичай містить більшу частину їх об'єднаної інформації, тому можна зменшити розмірність, використовуючи лише підмножину нових змінних. Зменшення розмірності часто називають вибором факторів або виділенням властивостей.

Хоча аналіз даних віддає перевагу автоматизованим методам, а не знанням предметної області, на першому кроці дослідження даних важливо переконатися, що виміряні змінні відповідають поставленій задачі. Інтеграція експертних знань через обговорення з постачальником даних або користувачем, ймовірно, призведе до кращих результатів. Практичні міркування включають: Які змінні є найважливішими для поставленої задачі, а які, швидше за все, будуть марними? Які змінні можуть містити велику кількість помилок? Які змінні будуть доступні для вимірювання і скільки коштуватиме їх вимірювання у майбутньому, якщо аналіз буде повторений? Які змінні фактично можна виміряти до того, як настане результат? Наприклад, якщо необхідно передбачити коштовність закриття поточного онлайн-аукціону, то не можна використовувати кількість ставок як прогноз, оскільки це не буде відомо до закриття аукціону.

Пакет `pandas` пропонує кілька методів, які допомагають узагальнювати дані. Метод `DataFrame.describe()` дає огляд усього набору змінних у даних. Методи `mean()`, `std()`, `min()`, `max()`, `median()` і `len()` є корисними для вивчення характеристик кожної змінної. Вони дають інформацію про діапазон та тип значень, які приймає змінна. Статистика мінімальних та максимальних значень можна використовувати для виявлення екстремальних значень, які можуть бути помилками. Середнє та медіана дають розуміння центральних значень певної змінної, а велике відхилення між ними також вказує на якісь перекося. Стандартне відхилення дає знання того, наскільки розсіяно дані відносно їх середнього значення. Також є корисними і інші характеристики. Наприклад, комбінація `.isnull().sum()` дає кількість нульових значень.

За допомогою наведеного нижче коду створимо таблицю із статистикою для файлу `BostonHousing.csv`:

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn import preprocessing
import matplotlib.pyplot as plt
```

```
housing_df = pd.read_csv('BostonHousing.csv')
housing_df.head(9)
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
3	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
4	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
5	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
6	0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
7	0.09378	12.5	7.87	0	0.524	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7
8	0.62976	0.0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21.0	396.90	8.26	20.4

```
## додавання CAT_MEDV до housing_df з використанням MEDV
housing_df["cat_medv"] = [1 if medv>=30.0 else 0 for medv in housing_df.medv]
housing_df
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	cat_medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0	0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6	0
2	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	1
3	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2	1
4	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9	0
...
328	0.17783	0.0	9.69	0	0.585	5.569	73.5	2.3999	6	391	19.2	395.77	15.10	17.5	0
329	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67	22.4	0
330	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6	0
331	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9	0
332	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88	11.9	0

```

# Обчислення середнього, стандартного відхилення, мінімального, максимального, медіани,
# довжини та відсутніх значень для CRIM
print('Mean : ', housing_df.crim.mean())
print('Std. dev : ', housing_df.crim.std())
print('Min : ', housing_df.crim.min())
print('Max : ', housing_df.crim.max())
print('Median : ', housing_df.crim.median())
print('Length : ', len(housing_df.crim))
print('Number of missing values : ', housing_df.crim.isnull().sum())
# Обчислення середнього, стандартного відхилення, мінімального, максимального, медіани,
# довжини та відсутніх значень для усіх змінних
pd.DataFrame({'mean': housing_df.mean(),
             'sd': housing_df.std(),
             'min': housing_df.min(),
             'max': housing_df.max(),
             'median': housing_df.median(),
             'length': len(housing_df),
             'miss.val': housing_df.isnull().sum(),
             })

```

```

Mean : 3.360341471471471
Std. dev : 7.352271836781107
Min : 0.00632
Max : 73.5341
Median : 0.26169
Length : 333
Number of missing values : 0

```

	mean	sd	min	max	median	length	miss.val
crim	3.360341	7.352272	0.00632	73.5341	0.26169	333	0
zn	10.689189	22.674762	0.00000	100.0000	0.00000	333	0
indus	11.293483	6.998123	0.74000	27.7400	9.90000	333	0
chas	0.060060	0.237956	0.00000	1.0000	0.00000	333	0
nox	0.557144	0.114955	0.38500	0.8710	0.53800	333	0
rm	6.265619	0.703952	3.56100	8.7250	6.20200	333	0
age	68.226426	28.133344	6.00000	100.0000	76.70000	333	0
dis	3.709934	1.981123	1.12960	10.7103	3.09230	333	0
rad	9.633634	8.742174	1.00000	24.0000	5.00000	333	0
tax	409.279279	170.841988	188.00000	711.0000	330.00000	333	0
ptratio	18.448048	2.151821	12.60000	21.2000	19.00000	333	0
black	359.466096	86.584567	3.50000	396.9000	392.05000	333	0
lstat	12.515435	7.067781	1.73000	37.9700	10.97000	333	0
medv	22.768769	9.173468	5.00000	50.0000	21.60000	333	0
cat_medv	0.174174	0.379830	0.00000	1.0000	0.00000	333	0

Як видно із результатів - різні змінні мають занадто різні діапазони значень. Середнє значення змінної CRIM (а також кількох інших), набагато більше медіани, що вказує на проблеми праворуч. Також важливим є те, що жодна змінна не має пропущених значень. Також, ймовірно, немає ознак екстремальних значень, які могли виникнути через помилки введення.

Для числових змінних можна обчислити повну матрицю кореляцій між кожною парою змінних, використовуючи метод `pandas corr()`.

```
housing_df.corr().round(2)
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	cat_medv
crim	1.00	-0.21	0.42	-0.04	0.46	-0.31	0.38	-0.40	0.67	0.62	0.31	-0.48	0.53	-0.41	-0.16
zn	-0.21	1.00	-0.52	-0.02	-0.50	0.33	-0.54	0.64	-0.30	-0.31	-0.38	0.17	-0.39	0.34	0.36
indus	0.42	-0.52	1.00	0.04	0.75	-0.44	0.64	-0.70	0.57	0.71	0.39	-0.34	0.61	-0.47	-0.39
chas	-0.04	-0.02	0.04	1.00	0.08	0.11	0.07	-0.08	0.01	-0.02	-0.13	0.06	-0.05	0.20	0.12
nox	0.46	-0.50	0.75	0.08	1.00	-0.34	0.74	-0.77	0.61	0.67	0.19	-0.37	0.60	-0.41	-0.24
rm	-0.31	0.33	-0.44	0.11	-0.34	1.00	-0.25	0.27	-0.27	-0.36	-0.37	0.16	-0.62	0.69	0.64
age	0.38	-0.54	0.64	0.07	0.74	-0.25	1.00	-0.76	0.45	0.51	0.26	-0.27	0.59	-0.36	-0.19
dis	-0.40	0.64	-0.70	-0.08	-0.77	0.27	-0.76	1.00	-0.48	-0.53	-0.23	0.28	-0.51	0.25	0.14
rad	0.67	-0.30	0.57	0.01	0.61	-0.27	0.45	-0.48	1.00	0.90	0.47	-0.41	0.48	-0.35	-0.19
tax	0.62	-0.31	0.71	-0.02	0.67	-0.36	0.51	-0.53	0.90	1.00	0.47	-0.41	0.54	-0.45	-0.28
ptratio	0.31	-0.38	0.39	-0.13	0.19	-0.37	0.26	-0.23	0.47	0.47	1.00	-0.16	0.37	-0.48	-0.43
black	-0.48	0.17	-0.34	0.06	-0.37	0.16	-0.27	0.28	-0.41	-0.41	-0.16	1.00	-0.36	0.34	0.15
lstat	0.53	-0.39	0.61	-0.05	0.60	-0.62	0.59	-0.51	0.48	0.54	0.37	-0.36	1.00	-0.74	-0.46
medv	-0.41	0.34	-0.47	0.20	-0.41	0.69	-0.36	0.25	-0.35	-0.45	-0.48	0.34	-0.74	1.00	0.80
cat_medv	-0.16	0.36	-0.39	0.12	-0.24	0.64	-0.19	0.14	-0.19	-0.28	-0.43	0.15	-0.46	0.80	1.00

Таблиця показує кореляційну матрицю для змінних BostonHousing. Більшість кореляцій низькі, а багато – негативні.

Іншим корисним підходом для дослідження даних є агрегація за однією або кількома змінними. Для агрегації за однією змінною можна використовувати метод `pandas value_counts()`:

```
housing_df.chas.value_counts()
```

```
0    313
1     20
Name: chas, dtype: int64
```

У наведеному прикладі показано кількість кварталів, що знаходяться на межі з річкою, порівняно з тими, що не межують з нею - змінна CHAS вибирається як змінна групування. За результатами виявляється, що більшість районів (313 з 333) не межують з річкою.

Метод `groupby()` можна використовувати для агрегування за однією або кількома змінними та для обчислення діапазону підсумкових статистичних даних (число, середнє, медіана тощо). Для категоріальних змінних можна отримати розбивку записів за комбінацією категорій.

Наприклад, нижче обчислюється середнє значення MEDV за CHAS та RM. Числова змінна RM (середня кількість кімнат на житло) спочатку групується у контейнери розміру 1 (0–1, 1–2 тощо).

```
# Створюємо контейнери розміру 1 для змінної за допомогою методу pd.cut().
# За замовчуванням метод створює категоріальну змінну.
housing_df['RM_bin'] = pd.cut(housing_df.rm, range(0, 10), labels=False)
```

```
# Обчислення середнього значення MEDV за RM і CHAS.
# Спочатку виконується групування методом groupby,
# потім обмежується аналіз до MEDV і визначається
# середнє значення для кожної групи.
housing_df.groupby(['RM_bin', 'chas'])['medv'].mean()
```

```
RM_bin  chas
3      0    25.300000
4      0    16.390000
5      0    17.450000
       1    26.500000
6      0    22.011243
       1    24.209091
7      0    35.889286
       1    45.300000
8      0    45.500000
       1    50.000000
Name: medv, dtype: float64
```

Отриманий результат показує, що в наборі даних немає околиць з такими комбінаціями, як наявність поблизу будинку річки і щоб в ньому було у середньому 3 кімнати.

Іншим корисним методом є `pivot_table()` у пакеті `pandas`, який дозволяє створювати зведені таблиці шляхом зміни форми даних за допомогою агрегованих змінних за заданим вибором. Наприклад, нижче обчислюється середнє значення MEDV за CHAS і RM і результат представлено у вигляді зведеної таблиці:

```
# використовується pivot_table() для зміни форми даних і створення зведеної таблиці
pd.pivot_table(housing_df, values='medv', index=['RM_bin'], columns=['chas'],
               aggfunc=np.mean, margins=True)
```

	chas	0	1	All
RM_bin				
3	25.300000	NaN	25.300000	
4	16.390000	NaN	16.390000	
5	17.450000	26.500000	17.812000	
6	22.011243	24.209091	22.145556	
7	35.889286	45.300000	37.065625	
8	45.500000	50.000000	46.000000	
All	22.295527	30.175000	22.768769	

У задачах класифікації, де мета полягає в тому, щоб знайти предиктори, які розрізняють два класи, правильним дослідницьким кроком є створення підсумків для кожного класу. Це може

допомогти у виявленні корисних предикторів, які відображають певне розділення між двома класами. Отриманні дані є корисними практично для будь-якої задачі аналізу даних.

Один із простих способів знайти надмірність даних полягає на побудові кореляційної матриці. Вона показує всі парні кореляції між змінними. Пари, які мають дуже сильну (позитивну чи негативну) кореляцію, містять багато збігів інформації і є кандидатами для скорочення даних шляхом видалення однієї з цих змінних. Видалення змінних, які сильно корелюють з іншими, корисно для уникнення проблем мультиколінеарності, які можуть виникнути в різних моделях. Під мультиколінеарністю розуміється наявність двох або більше предикторів, які мають однаковий лінійний зв'язок зі змінною результату.

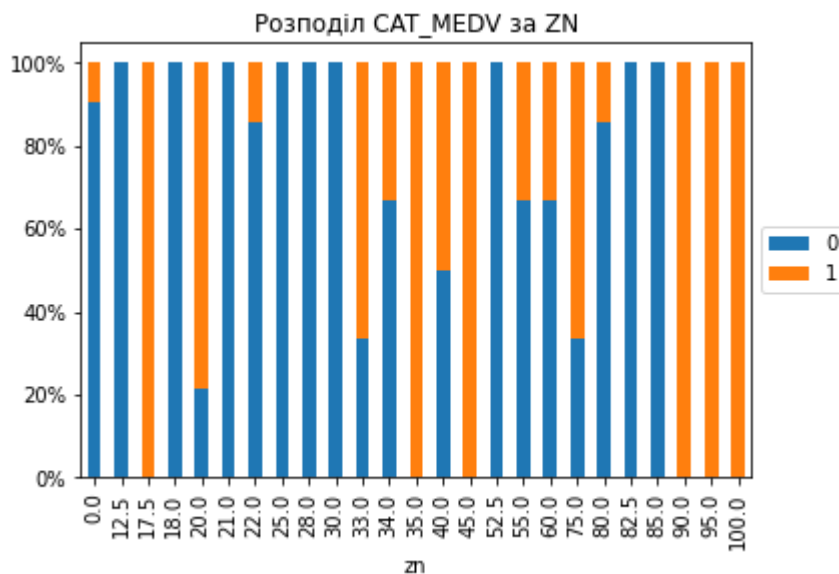
Кореляційний аналіз також може застосовуватись для виявлення дублювання змінних у даних. Іноді одна і та ж змінна випадково з'являється в наборі даних (під іншою назвою), оскільки набір даних було об'єднано з кількох джерел, одне й те саме явище вимірюється в різних одиницях тощо. Використання теплових карт таблиці кореляції (лаб.роб. 4) може спростити завдання виявлення сильних кореляцій.

Якщо категоріальна змінна має багато категорій, то ймовірно це може викликати певні проблеми при аналізі, що приведуть до суттєвого зростання розмірності набору даних. Один із способів впоратися з цим – зменшити кількість категорій, об'єднавши близькі чи подібні категорії. Поєднання категорій часто вимагає включення експертних знань. Для цієї задачі корисними є зведені таблиці, які дозволяють вивчити розміри різних категорій і поведінку змінної у кожній категорії. Як правило, категорії, що містять занадто мало спостережень, є кандидатами для поєднання з іншими категоріями. У задачах класифікації з категоріальною змінною результату зведена таблиця, що має розбивку на класи результатів, може допомогти визначити категорії, які на класи не розділяються. Ці категорії є кандидатами на включення до категорії «інше». Приклад цього можна показати на побудові розподілу змінної CAT_MEDV за категоріями, що утворено на основі змінної ZN:

```
# метод crosstab використовується для створення перехресної таблиці
# двох змінних
tbl = pd.crosstab(housing_df.cat_medv, housing_df.zn)
# конвертація значень у співвідношення
propTbl = tbl / tbl.sum()
```

```
import matplotlib.ticker as mticker
# plot the ratios in a stacked bar chart
ax = propTbl.transpose().plot(kind='bar', stacked=True)
ticks_loc = ax.get_yticks().tolist()
#ax.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax.set_yticklabels(['{:, .0%}'.format(x) for x in ticks_loc])
plt.title('Розподіл CAT MEDV за ZN')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()
```

Як результат отримуємо наступну діаграму:



Із діаграми видно, що розподіл CAT_MEDV є ідентичним для ZN = 17.5, 90, 95 і 100, де всі райони мають значення CAT_MEDV = 1. Тому вказані чотири категорії, ймовірно, можна об'єднати в одну категорію. Аналогічно можна об'єднати категорії ZN, де всі райони мають значення CAT_MEDV = 0. Подальша комбінація, ймовірно, також можлива для подібних стовпчиків на діаграмі.

Практичні завдання

1. Виконайте приклади, наведені у теоретичній частині лабораторної роботи.
2. У наданих файлах (див. лаб. роб. 2) розташовано статистичні дані Футбольної Прем'єр Ліги (FPL): успіхи кожного гравця за певний сезон у Лізі. Назва файлу відповідає певному сезону. Вміст файлу поділяється на колонки:
 - first_name — ім'я гравця
 - second_name — прізвище гравця
 - goals_scored - загальна кількість забитих голів за цей сезон
 - assists - загальна кількість гольових передач - присуджується гравцю з команди забиття воріт, який робить остаточний пас до того, як забити гол, включаючи автоголи.
 - total_points - загальна сума балів, зароблених у цьому сезоні
 - minutes - загальна кількість зіграних хвилин цього сезону
 - goals_conceded - загальна кількість голів, пропущених командою, поки гравець був на полі
 - creativity - творчість, оцінює ефективність гравців з точки зору створення можливостей для оцінки голів для інших гравців. Частина індексу ICT
 - influence - вплив, оцінює вплив гравця на матч, враховуючи дії, які можуть прямо чи побічно вплинути на результат матчу. Частина індексу ICT.
 - threat - загроза, вимірює гравців, які, швидше за все, заб'ють голи. Частина індексу ICT
 - bonus - троє найкращих гравців у кожному матчі відповідно до BPS отримають додаткові бонусні бали - 3 бали будуть нараховані гравцеві з найвищою оцінкою, 2 - другому кращому та 1 - третьому.
 - bps - система бонусних балів (BPS) використовує ряд статистичних даних для створення оцінки BPS для кожного гравця. Троє найкращих гравців у кожному матчі отримають бонусні бали.
 - ict_index - статистичний індекс, розроблений спеціально для оцінки гравця як активу FPL, що поєднує показники впливу, творчості та загроз.

- `clean_sheets` - загальна кількість чистих аркушів - присуджується гравцям, які не пропустили гол і зіграли принаймні 60 хвилин.
 - `red_cards` - кількість отриманих за сезон червоних карток.
 - `yellow_cards` - кількість отриманих за сезон жовтих карток.
3. Виконайте обчислення середнього, стандартного відхилення, мінімального, максимального, медіани, довжини та відсутніх значень для усіх змінних.
 4. Для числових змінних виконайте обчислення повної матриці кореляцій між кожною парою змінних.
 5. Виконайте агрегацію за змінною `goals_scored`.
 6. Створіть контейнери розміру 1 для змінної `goals_scored`. Виконайте обчислення середнього значення “minutes” за “goals_scored” та “assists”.
 7. Створіть зведену таблицю для змінних п.6.
 8. Побудуйте розподіл змінної “minutes” за категоріями, що утворено на основі змінної “goals_scored”.
 9. Підготуйте звіт з виконання практичних завдань.