ОБРОБКА ПРИРОДНОЇ МОВИ (NLP) У PYTHON (ТЕМАТИЧНЕ МОДЕЛЮВАННЯ) ЛЕКЦІЯ 4

- Завдання тематичного моделювання: захоплювати семантичну інформацію за межами окремих слів; виявляти приховані теми чи теми в документах;
- відповідно анотувати документи; використовувати анотації для управління, узагальнення, пошуку та рекомендування вмісту.

- У машинному навчанні та обробки природної мови, тематична модель являє собою тип статистичної моделі для виявлення абстрактних «тем», що зустрічаються в колекції документів.

- Тематичне моделювання — це часто використовуваний інструмент інтелектуального аналізу тексту для виявлення прихованих семантичних структур в тілі тексту та дозволяє нам ефективно аналізувати великі обсяги текстів, об’єднуючи документи в кластери.

- Великий обсяг текстових даних практично не помічений, що означає, що ми не зможемо застосувати наші попередні підходи до контрольованого навчання, тому що ці моделі машинного навчання насправді будуть залежати від історичних даних. У вас не буде зручною мітки, прикріпленою до текстового набору даних, наприклад позитивною або негативною. Замість цього у вас може бути безліч ярликів, таких як різні категорії, які публікуються в газетній статті. Отже нам належить спробувати виявити ці ярлики за допомогою тематичного моделювання.

ЕВОЛЮЦІЯ МОДЕЛЕЙ ВІД МІШКІВ СЛІВ ДО ПРИХОВАНИХ ТЕМ:

Модель	Рік	Опис
Векторна модель	1975	Представлення колекції документів векторами одного спільного для всієї колекції векторного простору
Латентно-семантичний аналіз	1988	Дозволяє аналізувати взаємозв'язок між набором документів і термінами, які в них зустрічаються, шляхом створення набору понять.
Ймовірний латентно-семантичний аналіз	1999	Заснований на змішаному розкладанні, отриманому з моделі прихованих класів
Прихований розподіл Діріхле	2003	Додає генеративний процес для документів: трирівнева ієрархічна, байєсівська модель

- Прихований розподіл Діріхле (LDA-Latent Dirichlet Allocation)— найпопулярніший метод моделювання тем, тож будемо використовувати його. LDA — це метод матричної факторизації. У векторному просторі будь-який корпус (збори документів) може бути представлений як матриця документ-термін.

- Тут слід мати на увазі дуже важливу ідею: насправді дуже складно оцінити неконтрольоване навчання, ефективність моделі навчання, тому що ми насправді не знали правильну тему або правильну відповідь. Все, що ми знаємо, це те, що документи, згруповані разом, поділяють якісь схожі тематичні ідеї. Користувач повинен визначити, що насправді являють ці теми.

ПІДГОТОВКА ДАНИХ

- Встановимо параметри для векторізатора: max_df=.2 означає ігнорувати терміни, що зустрічаються більш ніж у 20% документів; max_df = 3 означає ігнорувати терміни, що зустрічаються більш ніж у 3 документах; max_features — це розмір випадкових підмножин функцій, які слід враховувати при поділі вузла. Таким чином, це по суті позбавлення від термінів, які дійсно використовуються в багатьох документах. Так що зазвичай непогано викинути кілька дійсно загальних слів, а також викидати випадкові слова, можливо, навіть помилки або орфографічні помилки.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(max_df=.2, min_df=3, max_features=2000)
doc_train_matrix = vectorizer.fit_transform(train_docs['normalized'])
words = vectorizer.get_feature_names()
doc_train_matrix

topic_labels = ['Тема {}'.format(i) for i in range(1, 9)]

from sklearn.decomposition import LatentDirichletAllocation
lda = LatentDirichletAllocation(n_components=8, n_jobs=-1,
max_iter=500, learning_method='batch', evaluate_every=5, verbose=1,
random_state=42)

#Досліджуйте теми та розподіл слів
topics_count = lda.components_
print(topics_count.shape)
topics_count[:8]
```

- Отже, є два основних припущення, які ми збираємося зробити, щоб насправді застосувати LDA для тематичного моделювання. По-перше, в документах схожої тематики використовуються схожі групи слів. І це досить розумне припущення, тому що в основному йдеться, що якщо у вас є різні документи охоплюють схожу тему, наприклад, набір документів по темі бізнесу або економіки, які в кінцевому підсумку вони повинні використовувати схожі слова, такі як гроші, ціна, ринкові акції ... Ще одне припущення, яке ми збираємося зробити, полягає в тому, що приховані теми можна знайти, скориставшись пошуком роботи по групах слів, які часто зустрічаються разом в документах по всьому корпусу.І ми дійсно можемо думати про ці дві припущеннях математично.

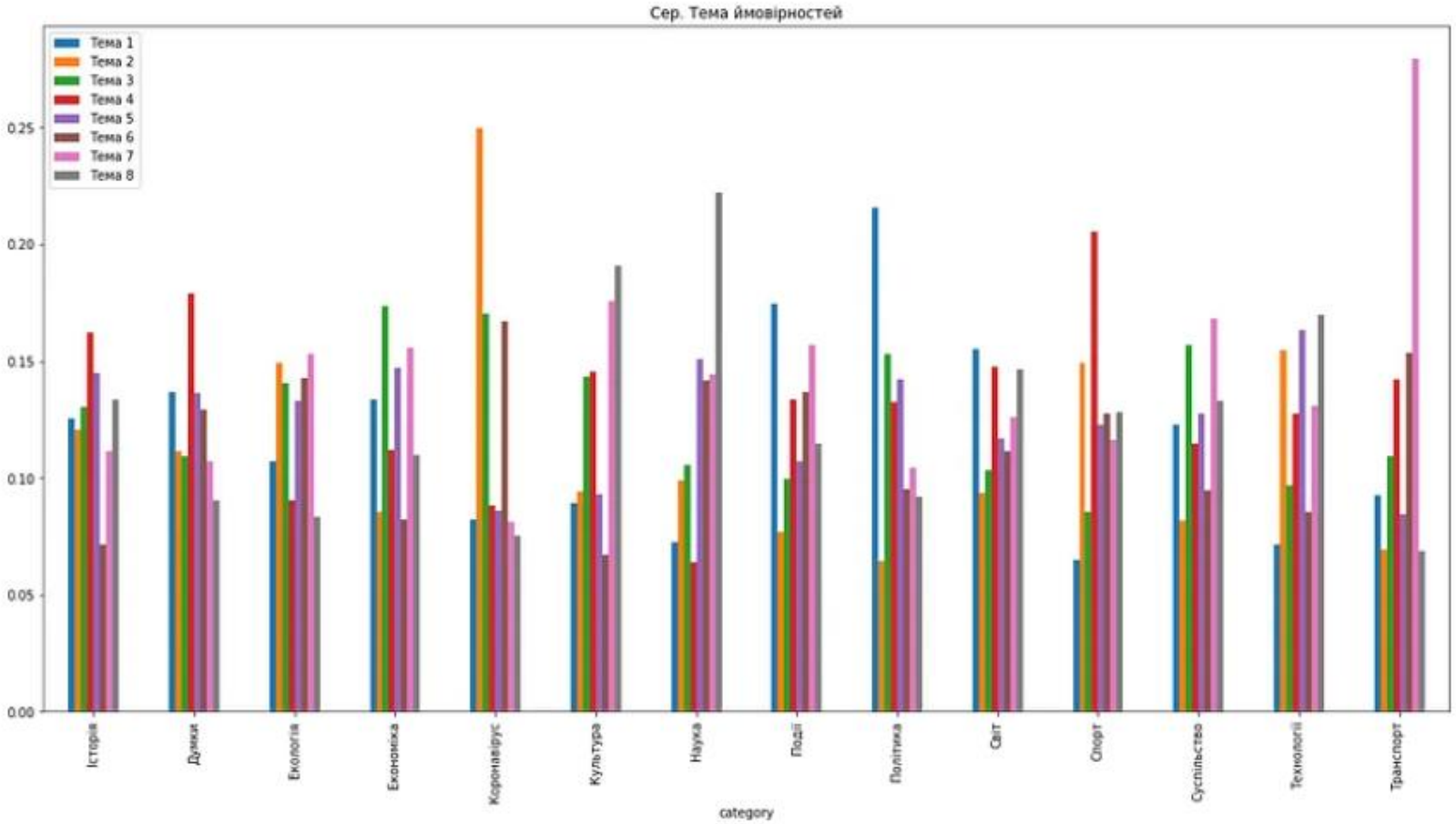
- Ми можемо змоделювати ці припущення наступним чином. Ми можемо сказати, що документи — це імовірнісні розподілу за деякими прихованим темам, а потім самі теми є імовірнісні розподілу слів. Ми можемо уявити, що будь-який конкретний документ матиме розподіл ймовірностей по заданій кількості прихованих тим, тому припустимо, що ми вирішили, що існує вісім прихованих тим в різних документах, тоді будь-який конкретний документ буде мати можливість приналежності до кожної теми.

МОЖЕМО ПОГЛЯНУТИ НА НАЙПОПУЛЯРНІШІ СЛОВА У ТЕМАХ:

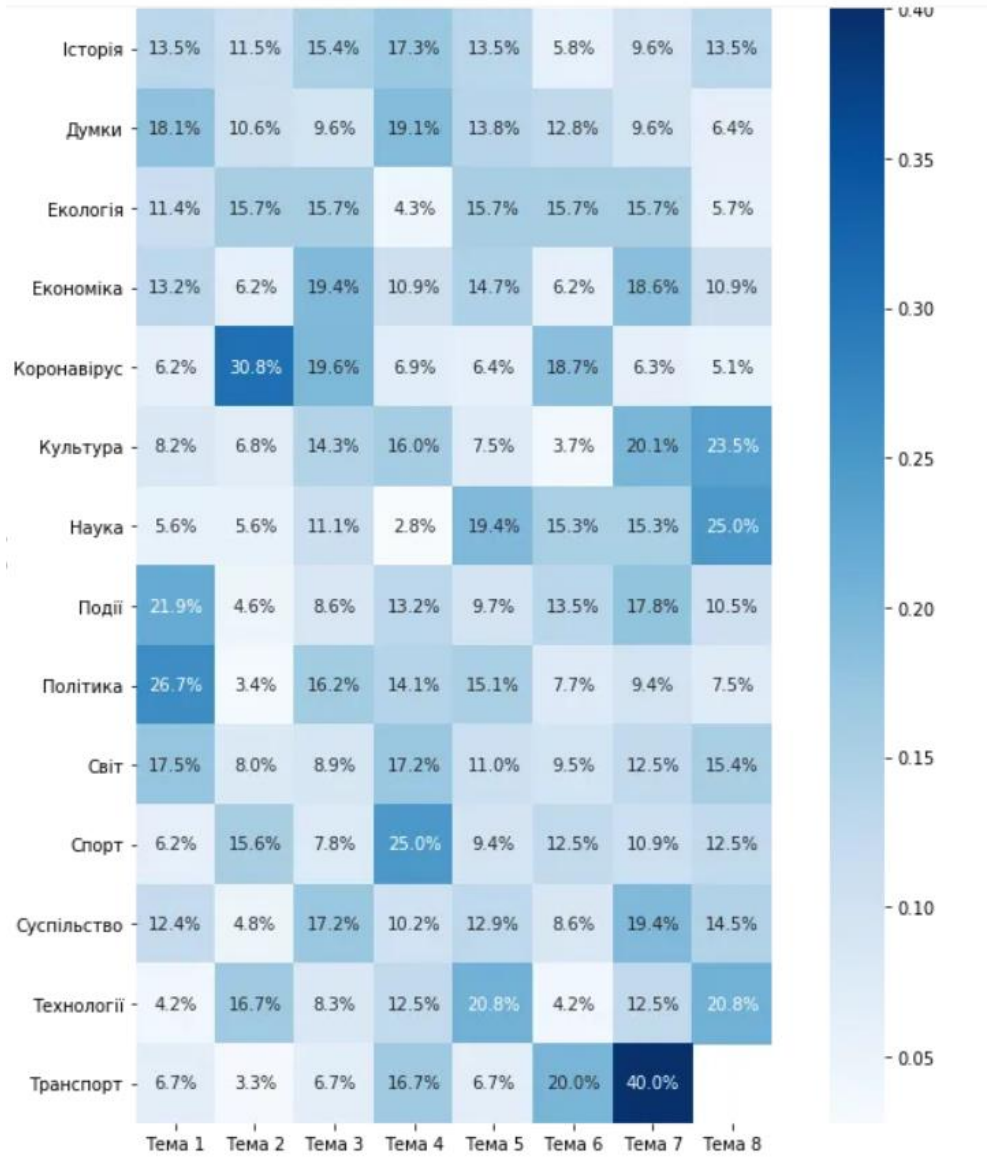
```
n_words = 12
top_words = {}
for topic, words_ in topics.items():
    top_words[topic] = words_.nlargest(n_words).index.tolist()
pd.DataFrame(top_words)
```

	Тема 1	Тема 2	Тема 3	Тема 4	Тема 5	Тема 6	Тема 7	Тема 8
0	день	випадок	карантин	рад	та	вакцина	рок	байден
1	щодо	доба	червоний	сша	сша	covid	фото	міжнародний
2	зеленський	19	зона	карантин	китай	19	штат	премія
3	санкція	covid	та	рік	підписати	щеплення	український	оголосити
4	рнбо	новий	карантинний	німеччина	угода	отримати	локдаун	створити
5	вихідний	виявити	моз	померти	кількість	мільйон	явитися	сша
6	рішення	зафіксувати	львов	український	співпраця	вакцинація	сполучений	працювати
7	росія	коронавірус	франція	2020	запровадити	світ	фільм	ім
8	проти	кий	уряд	загиблий	єс	доза	показати	ракета
9	вважати	вооз	країна	північний	кулеба	понад	чоловік	обмеження
10	та	підтвердити	вакцинація	голова	час	astrazeneca	белт	відкрити
11	мова	тисяча	2021	посол	український	коронавірус	версиин	супутник

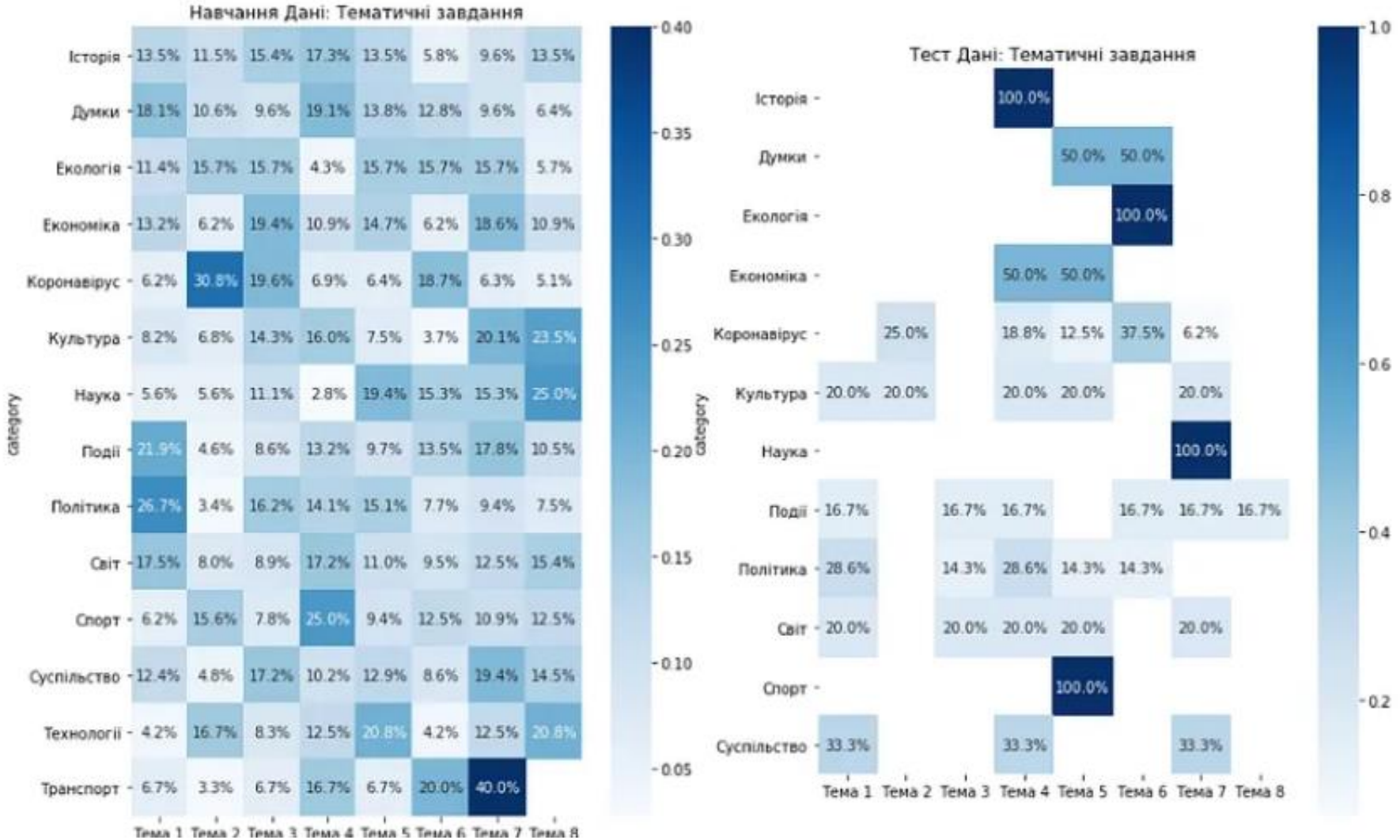
ТЕПЕР МОЖНА МОГЛЯНУТИ ЯК ПЕРЕТИНАЮТЬСЯ ТЕМИ І КАТЕГОРІЇ:



БІЛЬШІСТЬ ПОНЯТЬ ТЕМ МАЄ ВІДНОСНО ЧІТКИЙ ЗВ’ЯЗОК З ЯКОЮСЬ ІНШОЮ ТЕМОЮ.



- Отже, давайте переглянемо оцінку з тестовим набором. При використанні інформації про гіпотезу у тестовому наборі, а також вибір з цієї теми забезпечує найвищу якість. Таким чином, ми можемо призначити критику цієї теми, а потім подивитися, який вихідний текст.



ОСЬ ОЗНАКИ ТОГО, ЩО МИ БАЧИМО ЗОВСІМ ІНШУ ТЕМУ. ДАВАЙТЕ ПОДИВИМОСЯ НА НИХ В ДІЇ, ВИБРАВШИ КІЛЬКА ВИПАДКІВ.

```
train_eval = pd.DataFrame(data=lda.transform(doc_train_matrix),
 columns=topic_labels, index=train_docs.category)
test_eval = pd.DataFrame(data=lda.transform(doc_test_matrix),
 columns=topic_labels, index=test_docs.category)

#Перегляньте неправильно класифіковані статті
test_assignments = test_eval.groupby(level='category').idxmax(
    axis=1).reset_index(-1,
drop=True).to_frame('predicted').reset_index()
test_assignments['title'] = test_docs.title.values
test_assignments.head()
```

	category	predicted	title
0	Історія	Тема 4	Байден прокоментував голосування у справі про ...
1	Думки	Тема 6	Україна безкоштовно отримає 16 мільйонів доз в...
2	Думки	Тема 5	Україна розпочне щеплення вакциною Pfizer, на...
3	Екологія	Тема 6	На Львівщині впроваджують проєкт з аеромедично...
4	Економіка	Тема 4	Колишнього голову правління "ПриватБанку&...

```
misclassified = test_assignments[(test_assignments.category ==
‘Політика’) & (
 test_assignments.predicted == ‘Тема 1’)]
misclassified.title
38        Головна ялинка Харкова засяє як Ейфелева вежа
40    Проведення Олімпійських ігор дасть світові над...
```

ОЦІНКА ТА ВІЗУАЛІЗАЦІЯ РЕЗУЛЬТАТІВ

- pyLDAvis — інтерактивний інструмент, який дозволяє досліджувати значення об’єктів і взаємозв’язків. Так ви зможете візуалізувати великі робочі відносини, а також зрозуміти, як теми пов’язані з кожним з них.

```
!pip install pyLDAvis
import pyLDAvis.sklearn
lda_viz = pyLDAvis.sklearn.prepare(lda, doc_train_matrix, vectorizer,
mds='tsne')
pyLDAvis.display(lda_viz)
```