**NOTES:**

1. **There are 2 assignments to solve.**
2. **Unless you are a PhD student, you have to submit at least one of the 2 assignments.**
3. **If you are a PhD student, you need to aim towards submitting both of the assignments before the deadline.**

**Lab Assignment 1: Topic Modeling with Wikipedia Articles**

**Objective:**

**In this assignment, you will:**

1. **Scrape Wikipedia articles related to Vietnam.**
2. **Preprocess the text using spaCy.**
3. **Apply LDA to extract 3 topics.**
4. **Visualise the topics using pyLDAvis and interpret the results.**

**🔹 Task 1: Install Required Libraries**

**Before you start, ensure you have the required Python libraries installed:**

**bash**

**!pip install wikipedia-api spacy gensim pyLDAvis matplotlib**

**python -m spacy download en\_core\_web\_sm**

**🔹 Task 2: Understanding Topic Modeling**

* **LDA (Latent Dirichlet Allocation) is a technique for discovering hidden topics in text.**
* **Each document is assumed to be a mixture of topics, and each topic is a mixture of words.**
* **pyLDAvis is a tool for interpreting topic modeling results visually.**

**🔹 Task 3: Fetch Wikipedia Articles**

**You will extract text from four Wikipedia pages:**

1. **Vietnam (General country information)**
2. **Vietnamese language (Linguistic aspects)**
3. **Ho Chi Minh (Vietnamese leader)**
4. **Vietnam War (Major historical conflict)**

**Instructions:**

* **Use the Wikipedia API to fetch the text of these articles.**
* **Ensure you include a user agent in your API request (Wikipedia blocks requests without one).**
* **Handle cases where an article does not exist.**

**🔹 Task 4: Text Preprocessing with spaCy**

* **Convert the text to lowercase.**
* **Remove stopwords (common words like "the", "is", "and").**
* **Tokenise the text into words.**
* **Lemmatise words (convert them to their root form, e.g., "running" → "run").**

**🔹 Task 5: Create & Train an LDA Model**

* **Convert the processed text into a bag-of-words representation.**
* **Create a dictionary mapping words to unique IDs.**
* **Train an LDA model with 3 topics.**
* **Tune parameters (e.g., number of passes) to improve results.**

**🔹 Task 6: Visualise & Interpret Topics**

* **Use pyLDAvis to visualise the extracted topics.**
* **Save the visualisation as an HTML file and open it in a browser.**

**Questions to Answer:**

1. **What words define each topic?**
2. **How do the Wikipedia articles influence topic distribution?**
3. **Do any topics overlap? Why?**
4. **How does adjusting the relevance slider (λ) in pyLDAvis change the words?**
5. **What insights do you gain from this topic modeling approach?**

**🔹 Bonus (Optional)**

**Try modifying the assignment:  
✔ Increase num\_topics to 5 – How does this change the results?  
✔ Use different Wikipedia articles (e.g., "Hanoi", "French Indochina").  
✔ Apply bigram/trigram modeling to capture multi-word phrases.**

1. **DO NOT SUBMIT YET, Make sure you solve Assignment 2 below**
2. **If you run out of time, make sure you at least submit assignment 1.**
3. **PhD students, should aim to submit both assignments.**

**Expected output:**

A screenshot of a graph

AI-generated content may be incorrect.

**Explaining the pyLDAvis GUI for LDA Topic Modeling**

The **pyLDAvis** interface helps visualise topics extracted by **Latent Dirichlet Allocation (LDA)**. Here’s a breakdown of the key components, using the word **"Vietnam"** as an example.

**1️ Left Panel: Intertopic Distance Map**

* This shows the **relationship between topics** in **2D space**.
* Each **circle** represents a **topic**:
  + **Size of the circle**: The proportion of documents assigned to that topic.
  + **Distance between circles**: Similar topics are **closer together**, while different topics are **far apart**.

🛠 **How to Use:**

* Click on a topic to **view its most relevant words** on the right.
* Large circles indicate **dominant topics** in the dataset.

🔍 **Example:**

* The **red circle (Topic 1)** is the **largest**, meaning it covers **most of the text**.
* This suggests that the **Vietnam War is a major theme** in the Wikipedia articles.

**2️ Right Panel: Top-30 Most Relevant Terms for Selected Topic**

* This **bar chart** shows the **most relevant words** for the selected topic.
* **Two bars for each word**:
  + **Red bar**: Frequency of the word **inside the selected topic**.
  + **Blue bar**: The **overall frequency** of the word in the whole dataset.

🔍 **Example (Word: "Vietnam")**

* "Vietnam" has a **long red bar**, meaning it appears **very frequently in this topic**.
* The **blue bar is also long**, meaning it is **a very common word overall**.
* Since **Vietnam dominates the topic**, this topic likely **discusses the Vietnam War or Vietnamese history**.

**3️ Slider: Adjusting Relevance Metric (λ)**

* **Controls the balance** between:
  + **λ = 1** → Shows words that are **most frequent in this topic**.
  + **λ = 0** → Shows words that are **rare overall but uniquely important to this topic**.

🛠 **How to Use:**

* Move the slider **left** (λ → 0): See words that **uniquely define this topic**.
* Move the slider **right** (λ → 1): See words that are **most common in this topic**.

🔍 **Example (If you lower λ)**

* Some **less frequent but important words** like **"battle"**, **"treaty"**, or **"conflict"** might appear, revealing deeper insights.

**4️ Marginal Topic Distribution (Bottom-Left)**

* This shows how **dominant each topic is** in the entire dataset.
* Larger circles = **More documents assigned to that topic**.

🔍 **Example:**

* The **largest topic in the visualization (~52%)** suggests that most of the text discusses **Vietnam War history**.

**Lab Assignment 2: Dimensionality Reduction & Digit Comparison Using PCA**

**Objective:**

In this assignment, you will:

1. **Load the Digits dataset** and explore its structure.
2. **Apply Principal Component Analysis (PCA)** to reduce dimensionality.
3. **Visualise the transformed data** in 2D.
4. **Compare similar-looking digits** and identify common patterns.

**🔹 Task 1: Install Required Libraries**

Before you start, ensure you have the required Python libraries installed:

pip install numpy matplotlib scikit-learn

**🔹 Task 2: Understanding PCA**

* **Principal Component Analysis (PCA)** is a technique that reduces high-dimensional data while preserving the most important patterns.
* The **Digits dataset** consists of **8×8 pixel images** (64D).
* PCA will **compress** this 64D data into **2D or 3D**, allowing us to **visualise digit similarities**.

**🔹 Task 3: Load & Explore the Digits Dataset**

* Load the dataset from **scikit-learn**.
* Each digit is represented as **a 64-dimensional vector** (8×8 pixels).
* Plot a few sample images to understand the dataset.

**🔹 Task 4: Apply PCA for Dimensionality Reduction**

* Reduce the data from **64D to 2D** using PCA.
* Plot the transformed data in a **scatter plot**, colouring points by digit labels.
* Identify clusters and how digits group together.

**🔹 Task 5: Compare Similar-Looking Digits**

* Find which digits **overlap** in the PCA-reduced space.
* Identify **common misclassifications** (e.g., **1 vs. 7**, **3 vs. 8**).
* Compute distances between digit centroids to determine the **most visually similar numbers**.

**Questions to Answer:**

1. **Which numbers cluster together in PCA space?**
2. **Are there any digits that overlap significantly? Why?**
3. **What numbers do people often write similarly? (e.g., 1 and 7, 3 and 8, etc.)**
4. **Does reducing dimensions to 2D capture enough information? Would 3D be better?**
5. **Can PCA be useful in handwriting recognition? Why or why not?**

**If you have completed assignments 1 and 2 for this lab. Zip them together and submit.**