Task Details: Customer Segmentation

You will simulate a dataset with two features: **Annual Income (k$)** and **Spending Score (1-100)**, and train a **K-Means** clustering model (k=5).

Required Files and Steps:

1. Model Training Script (`train_customer_model.py`)

Create this script to perform the following:

- **Simulate Data:** Create 200 random data points for the two required features.
- **Train Model:** Fit a `KMeans` model with $k=5$ clusters to the data.
- **Save:** Save the trained model to `customer_kmeans_model_yourname.pkl`.
- **Save Metadata:** Save the feature names and the trained cluster centers to `customer_metadata.pkl`.

2. Streamlit Application

Create this script to perform the following:

- **Load Resources:** Use `@st.cache_resource` to load both `.pkl` files.
- **Input Widgets:** Use **two** **`st.sidebar.slider`** widgets for the Annual Income and Spending Score features.
- **Prediction:** Use the loaded model to predict the **Cluster ID** for the user's input.
- **Output:**
    - Display the predicted **Cluster ID** using `st.metric()`.
    - Display a **scatter plot** of the original simulated data, clearly highlighting the predicted customer's location on the plot.

Submission

Submit the following items:

1. `customer_kmeans_model_yourname.pkl`.
2. A screenshot of your running Streamlit application.

Homework Part 2 – Evaluating Your Kafka Agents with GEval

In this follow-up assignment, you'll extend your three-agent Kafka system (Planner - Writer - Reviewer) from the previous homework. Keep the same topics—**inbox**, **tasks**, **drafts**, and **final**—and the same message flow, but now add an evaluation stage using **GEval** (from the DeepEval framework) and the **Any model(Feel free to explore models on ollama)** as the judge model.
After your Reviewer posts the approved message to the final topic, build a small evaluation script that consumes the Planner's plan, the Writer's draft, and the Reviewer's final answer (linked by a shared correlation_id) and uses GEval to rate their quality.
Use metrics such as Plan Quality (for the Planner's plan), Helpfulness (for Writer and Reviewer answers), and Final-vs-Draft Improvement (to see whether the Reviewer improved the draft).

Run the evaluator after sending a test question and display the scores for each stage.

Rrefer back to your previous homework for the agent setup, topic creation, and run commands;

your submission should be a report showing the GEval scores and a brief paragraph (≈150 words) reflecting on how automated evaluation helps you measure and improve agent coordination and the code for the Geval part with LLM integration.