Step-by-Step Process

1. Setting up my AWS environment

- I started by logging into my AWS root account then created an IAM user with proper permissions to access Amazon Personalize and Amazon S3.
- I set up Multi-Factor Authentication (MFA) and created an IAM role with AmazonPersonalizeFullAccess.
- I also created an S3 bucket named: “habiba-personalize-bucket” which I used to upload my dataset.

2. Preparing the Dataset

- I used the MovieLens 100K dataset as a real-world example of user-item interactions.
- From the downloaded files, I extracted the “u.data” file and converted it into a CSV format called “interactions.csv”.
- To simplify the testing and minimize training time during the assessment, I sampled 1000 rows from the full dataset.
- I made sure it included the required columns:
  user_id, item_id, timestamp, event_type
- Since the original file didn’t include an “event_type”, I added a column and filled it with a constant value "watch" to meet amazon’s schema requirements
- I uploaded the finalized CSV file to my S3 bucket.

3. Creating Resources in Amazon Personalize

a. Dataset Group

- I created a new dataset group named: movie-recommender-habiba.

b. Schema

- I defined the following schema for my interaction data:

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {"name": "user_id", "type": "string"},
    {"name": "item_id", "type": "string"},
    {"name": "timestamp", "type": "long"},
    {"name": "event_type", "type": "string"}
  ],
  "version": "1.0"
}
```

c. Dataset & Import

- I linked the S3 file with the dataset group.
- To avoid permission issues, I created an IAM role and attached the required bucket policy to my S3 bucket.
- Once everything was connected, the data was successfully imported into Amazon Personalize.

4. Training the Recommender

- Since I was working with movie data, I selected the "Video on Demand" domain and chose the use case "Top picks for you"
- I named my recommender: “top_picks_recommender”
- I left the default values for most advanced settings, such as exploration factor (0.3) and minimum recommendation requests (1)
- Once created, I waited for the model to finish training.

5. Generating Recommendations

- After the recommender was ready, I used the Amazon Personalize console to generate recommendations.
- I tested the recommender by entering a sample user_id.
- The recommender returned a list of 25 item IDs along with a score for each — showing how relevant that item was for that user.

Example Output (for User ID: 102)

Rank	Item ID	Score
1	234	0.0086189
2	274	0.0064724
3	25	0.0064259
4	117	0.0057421
5	328	0.0054693
…	…	…
25	758	0.0033

- This means the model recommended item 234 as the most relevant for user 102 based on their past interactions.

Example Output (for User ID: 286)

Rank	Item ID	Score
1	234	0.0092406
2	25	0.0070455
3	274	0.0069612
4	117	0.0057895
5	217	0.0055899
…	…	…
25	407	0.0034857

- This means the model recommended item 234 as the most relevant for user 286 based on their past interactions.

Example Output (for User ID: 119)

Rank	Item ID	Score
1	520	0.0092081
2	234	0.0090202
3	274	0.0069438
4	25	0.0066536
5	117	0.0057587
…	…	…
25	443	0.0032683

- This means the model recommended item 520 as the most relevant for user 119 based on their past interactions.