Dataset: https://www.kaggle.com/datasets/shivamb/netflix-shows

---

40 Beginner-Friendly Coding + DAG + Airflow Questions Using the Netflix Kaggle Dataset

---

SECTION 1 — Basic Python ETL on the Netflix Dataset

1. Write Python code to load `netflix_titles.csv` into a pandas DataFrame.
2. Print the first 10 rows of the dataset.
3. Write code to check for missing values in each column.
4. Remove all rows where `title` is missing.
5. Convert all titles in the `title` column to lowercase.
6. Filter only rows where `type == "Movie"`.
7. Count how many shows were released in 2020.
8. Write code to remove duplicate titles from the dataset.
9. Extract only the columns: `title`, `type`, `country`, `release_year`.
10. Save the cleaned dataset to `cleaned_netflix.csv`.

SECTION 2 — Python Functions for ETL (Used Later in Airflow DAG)

11. Write a function `extract()` that reads `netflix_titles.csv` and returns a DataFrame.
12. Write a function `transform(df)` that drops rows with missing `country`.

13. Write a function that filters movies produced in Kenya.
14. Write a function that counts how many TV Shows came from India.
15. Write a function `load(df, path)` that saves the transformed DataFrame to a CSV.
16. Combine `extract → transform → load` into a single Python script.
17. Write Python code that checks if the dataset file exists before running extract.
18. Write code that logs the number of rows before and after transformation.
19. Write a function that creates a summary dictionary:
    `{"total_movies": X, "total_shows": Y}`
20. Write a function that finds the **top 5 countries** with the most Netflix titles.

SECTION 3 — Airflow DAG Basics Using the Netflix Dataset

21. Create an Airflow DAG called `"netflix_etl_dag"` scheduled to run daily.
22. Add a task that prints `"Starting Netflix ETL"`.
23. Add a PythonOperator task that runs your `extract()` function.
24. Add another PythonOperator task that runs your `transform()` function.
25. Add a task that loads the final cleaned data into `cleaned_netflix.csv`.
26. Set dependencies so the order is:
    start → extract → transform → load
27. Add default args with `retries=1` and `retry_delay=5 minutes`.
28. Add a FileSensor that waits for `netflix_titles.csv` before extract runs.
29. Add a BashOperator that prints the cleaned dataset row count.
30. Make the DAG send an email if extraction fails.

SECTION 4 — Slightly More Advanced Airflow DAG + ETL

31. Modify the DAG so it only transforms rows where `release_year > 2015`.
32. Add a task that generates a summary JSON file with movie counts by country.
33. Add a BranchPythonOperator that checks:

- If the dataset has > 5000 rows → go to `transform`
- Else → go to a task called `"skip_transform"`

34. Add a task that uploads the cleaned CSV to a folder called `/processed`.
35. Add a sensor that waits for a directory `/data/` to exist.
36. Add a DAG run parameter (Airflow Variable) for `min_year` and use it inside transform.
37. Create a failure callback function that logs `"Netflix DAG failed!"`.
38. Add a task that deletes temporary files after loading.
39. Add a Python task that finds the **top 10 directors** with the most titles.
40. Add a final task `"notify_done"` that prints `"Netflix ETL Completed"` and make all tasks flow into it.
