

## Title:

***“End-to-End Netflix Analytics Dashboard: Python | Azure | Power BI | MySQL”***

## Content:

- ✓ Your Name: Tanmay Sharma
- ✓ Role Target (Data Analyst / BI Engineer / Data Engineer)
- ✓ Logo strip (Power BI, Python, MySQL, Azure)

- Python 
- Power BI 
- MySQL 
- Azure 



## Slide 2: Project Overview:

Objective: Analyse Netflix data & forecast future trends using BI tools.

Tools Used: Python, Pandas, Faker, Power BI, MySQL, Azurite.

Workflow:

Python → CSV → Azurite → Power BI (Static) → MySQL  
(Dynamic)

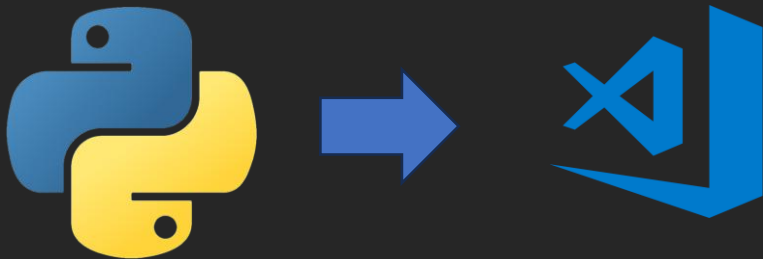
Visual: Flow diagram with arrows showing Python → Azurite → Power BI → MySQL.



## Slide 3: Data Generation (Python)

### •What you did:

- Wrote Python script using Pandas & Faker to generate 10K+ rows of Netflix-like data.
- Saved as CSV for analysis..



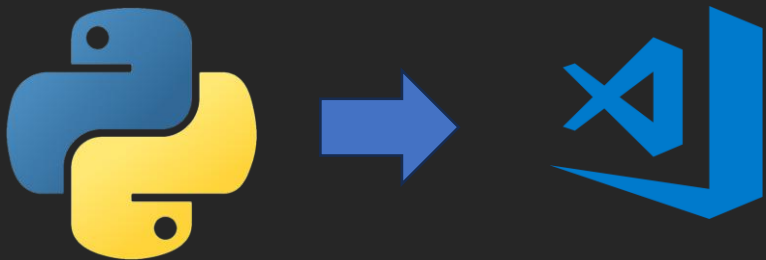
```
Netflix.py > ...
1 import pandas as pd
2 import numpy as np
3 from Netflix2 import Faker
4 import random
5 from sklearn.linear_model import LinearRegression
6 import matplotlib.pyplot as plt
7
8 # Initialize Faker for synthetic data generation
9 fake = Faker()
10
11 # Step 1: Generate synthetic Netflix-like dataset
12 num_rows = 5000
13 years = np.random.randint(2000, 2023, size=num_rows)
14 genres = ['Drama', 'Comedy', 'Action', 'Documentary', 'Thriller', 'Romance', 'Horror', 'Sci-Fi']
15 ratings = ['G', 'PG', 'PG-13', 'R', 'TV-MA']
16 countries = ['United States', 'India', 'United Kingdom', 'Canada', 'Australia', 'Japan']
17
18 data = {
19     'title': [fake.sentence(nb_words=3) for _ in range(num_rows)],
20     'release_year': years,
21     'genre': [random.choice(genres) for _ in range(num_rows)],
22     'country': [random.choice(countries) for _ in range(num_rows)],
23     'rating': [random.choice(ratings) for _ in range(num_rows)]
24 }
25
26 df = pd.DataFrame(data)
27
28 # Save the dataset
29 dataset_path = r'T:\python\JARVIS\netflix_dataset.csv'
30 df.to_csv(dataset_path, index=False)
31 print(f"Netflix dataset created at: {dataset_path}")
32
33 # Step 2: Group data by year for forecasting
34 yearly_data = df.groupby('release_year').size().reset_index(name='count')
35
36 # Prepare data for ML
37 X = yearly_data[['release_year']]
38 y = yearly_data['count']
39
40 # Step 3: Train Linear Regression model
41 model = LinearRegression()
42 model.fit(X, y)
```

```
# Predict next 5 years
future_years = np.array(range(2023, 2028)).reshape(-1, 1)
predictions = model.predict(future_years)

# Create forecast DataFrame
forecast_df = pd.DataFrame({
    'release_year': future_years.flatten(),
    'predicted_titles': predictions.astype(int)
})

# Save forecast
forecast_path = r'T:\python\JARVIS\netflix_forecast.csv'
forecast_df.to_csv(forecast_path, index=False)
print(f"Forecast file created at: {forecast_path}")

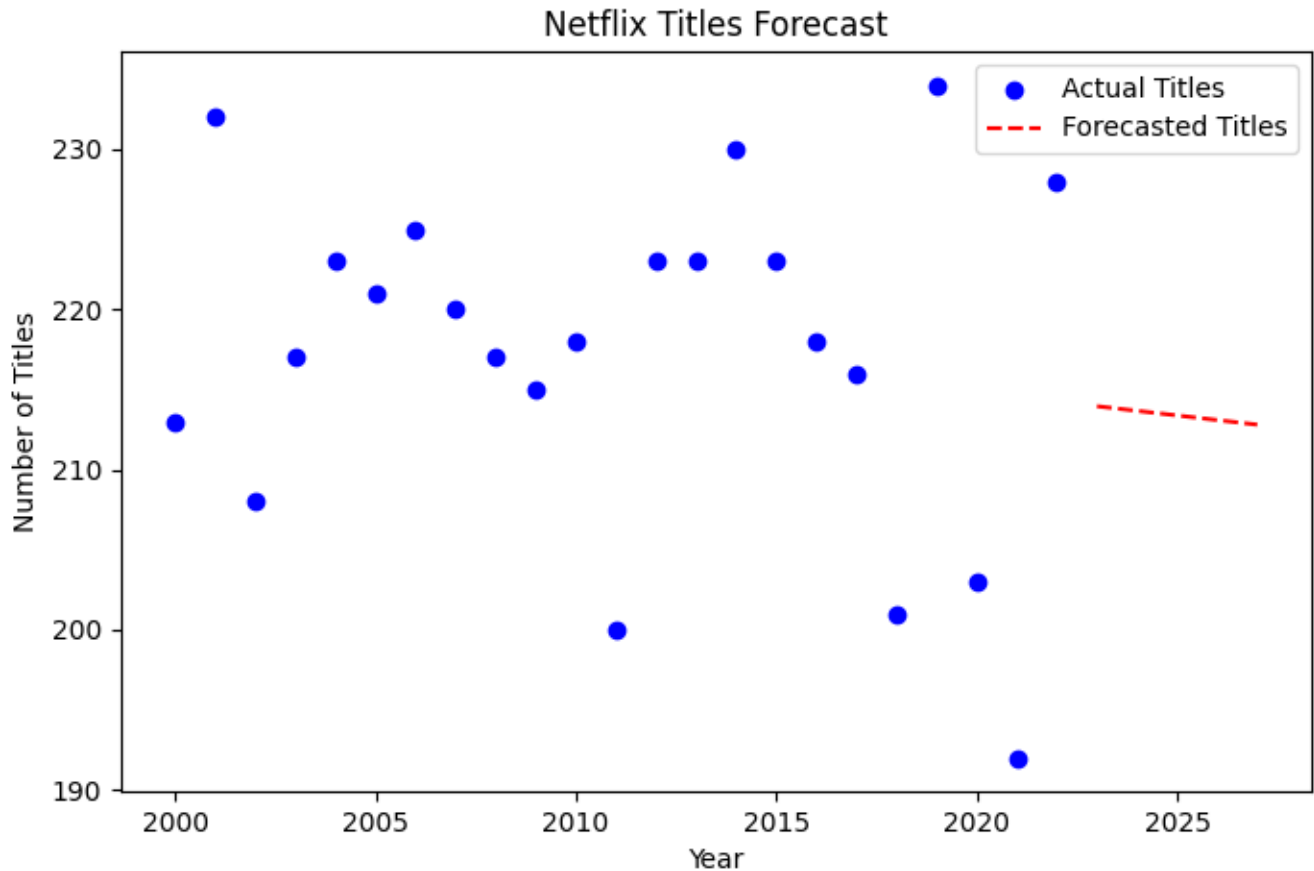
# Step 4: Visualization
plt.figure(figsize=(8, 5))
plt.scatter(X, y, color='blue', label='Actual Titles')
plt.plot(future_years, predictions, color='red', linestyle='--', label='Forecasted Titles')
plt.xlabel('Year')
plt.ylabel('Number of Titles')
plt.title('Netflix Titles Forecast')
plt.legend()
plt.show()
```



# “Forecasting Netflix Titles (Python)”

## Content:

- Technique: **Linear Regression (Scikit-learn)**.
- Predicted growth trend for next 5 years (2023–2027).
- Purpose: Help content strategy & investment planning.



A	B
release_year	predicted_titles
2023	442
2024	442
2025	443
2026	444
2027	444

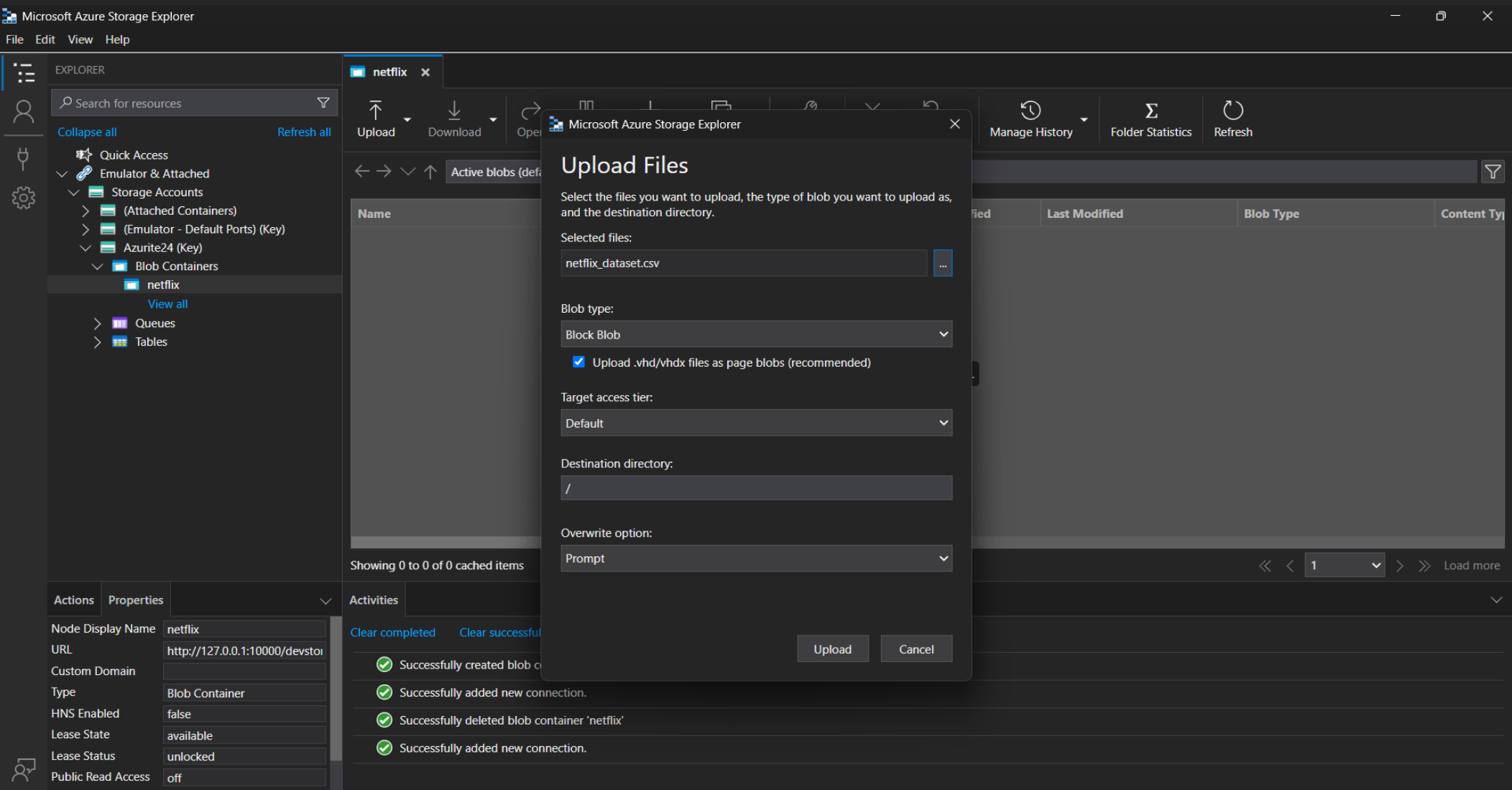
# Azure Storage (Azurite)

•**Purpose:** Simulated cloud storage environment locally.

•**Steps:**

- Uploaded Netflix CSV to Azurite container.
- Connected Power BI to Azurite Blob.

```
Azurite Blob service is starting at http://127.0.0.1:10000
Azurite Blob service is successfully listening at http://127.0.0.1:10000
Azurite Queue service is starting at http://127.0.0.1:10001
Azurite Queue service is successfully listening at http://127.0.0.1:10001
Azurite Table service is starting at http://127.0.0.1:10002
Azurite Table service is successfully listening at http://127.0.0.1:10002
```



EXPLORER

Search for resources

Collapse all

Refresh all

Quick Access

Emulator & Attached

Storage Accounts

(Attached Containers)

(Emulator - Default Ports) (Key)

Azurite24 (Key)

Blob Containers

netflix

View all

Queues

Tables

netflix

Upload

Download

Open

Preview

New Folder

Select All

Properties

Delete

Undelete

Manage History

Folder Statistics

Refresh

Active blobs (default) netflix

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Ty
netflix_dataset.csv	Hot (inferred)	31-07-2025 15:21	31-07-2025 15:21	Block Blob	application/

Showing 1 to 1 of 1 cached items

1

Load more

Actions Properties

Node Display Name

URL

Custom Domain

Type

HNS Enabled

Lease State

Lease Status

Public Read Access

netflix

http://127.0.0.1:10000/devstoi

Blob Container

false

available

unlocked

off

Activities

Clear completed

Clear successful

Transfer of 'T:\python\JARVIS\netflix\_dataset.csv' to 'devstoreaccount1/netflix/' complete: 1 item transferred (used SAS, discovery completed)

Started at: 31-07-2025 15:21, Duration: 4 seconds

Successfully created blob container 'netflix'

Successfully added new connection.

Successfully deleted blob container 'netflix'

Successfully added new connection.

Copy AzCopy Command to Clipboard

# Power BI Dashboard (Static)

## •What you built:

- KPIs: Total Titles, Countries, Popular Genre.
- Visuals: Genre Distribution, Rating Breakdown, Actual vs Forecast Line Chart.



NETFLIX

## Netflix Analytics Dashboard

Total Titles

10K

Number Of Countries

6

Most Popular genre

Action

"Titles After Filters"

23

genre, country, rating

Multiple selections

release\_year

2003

2016

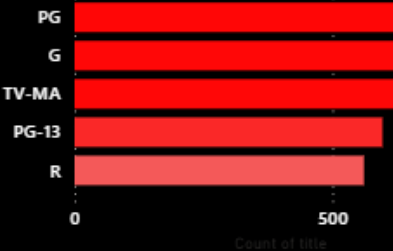


### Actual Vs Forecast

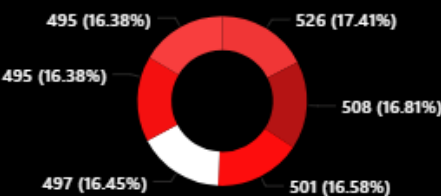
type ● Actual ● Forecast



### Rating Distribution



### Genre Distribution



### Country Wise Titles





# MySQL Integration (Dynamic)

## •What changed:

- Imported Netflix CSV into MySQL using **Import Wizard**.
- Built queries for KPIs and Forecast.
- Connected Power BI to MySQL for **live refresh**.



Result Grid

Filter Rows:

Export:

Wrap Cell Content

	title	release_year	genre	country	rating
▶	Movie_1	2013	Drama	USA	PG-13
	Movie_2	2009	Drama	Canada	TV-MA
	Movie_3	2007	Action	India	TV-MA
	Movie_4	2016	Comedy	India	PG
	Movie_5	2011	Drama	USA	TV-MA
	Movie_6	2011	Comedy	UK	R
	Movie_7	2001	Comedy	India	G
	Movie_8	2014	Documentary	USA	G
	Movie_9	2010	Romance	Australia	PG
	Movie_10	2017	Action	UK	G
	Movie_11	2018	Comedy	Australia	G
	Movie_12	2016	Thriller	Japan	PG-13
	Movie_13	2020	Thriller	UK	PG-13
	Movie_14	2019	Comedy	Canada	PG-13
	Movie_15	2020	Action	UK	G

## Data source settings

Manage settings for data sources that you have connected to using Power BI Desktop.

☒ Data sources in current file ☐ Global permissions

Search data source settings

localhost;netflix\_db

t:\python\jarvis\netflix\_dataset.csv

t:\python\jarvis\netflix\_forecast.csv

## Key Features

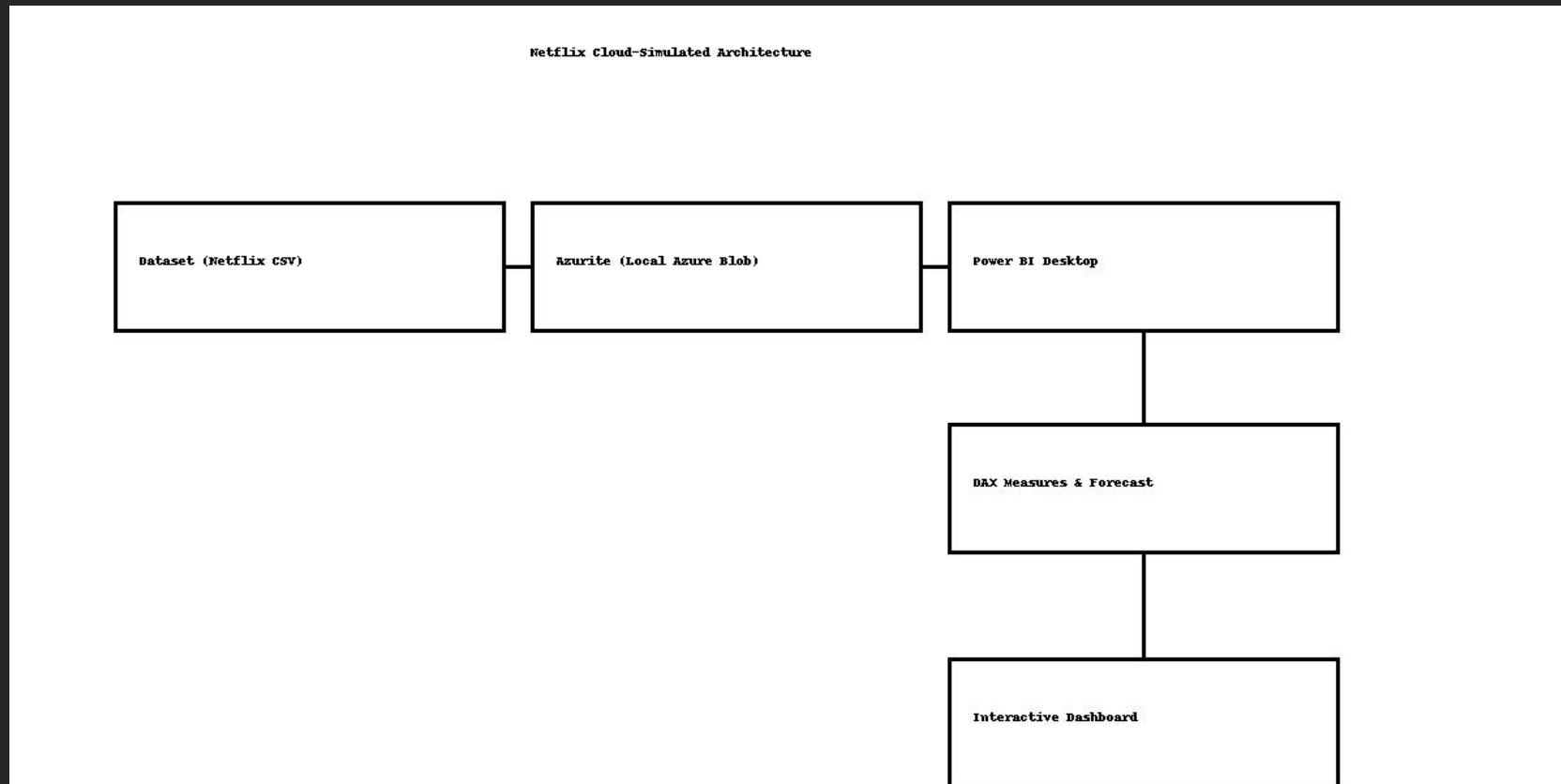
- ✓ End-to-end pipeline
- ✓ Forecasting using Python + Linear Regression
- ✓ Power BI dynamic refresh via MySQL
- ✓ Cloud simulation via Azurite
- ✓ Clean, MAANG-ready dashboard design

## Outcome:

- Real-time Netflix content analysis solution.
- Demonstrated **Data Engineering + BI + Analytics** skills.

## GitHub Link:

[https://github.com/Tanu272004/Netflix\\_analysis\\_bi\\_project](https://github.com/Tanu272004/Netflix_analysis_bi_project)



## Before (Initial State)

- Data stored in **local CSV** → Static, manual updates.
- Dashboard was **limited to historical data** (no real-time refresh).
- Forecasting done separately in Python → **not integrated** with BI tool.
- Cloud simulation not implemented → looked like a basic portfolio project.

## After (Final State)

- Built a **fully automated, end-to-end BI pipeline**:
  - ✓ **Python** for synthetic data generation (10K rows).
  - ✓ **Azulite** to simulate Azure Blob for cloud integration.
  - ✓ **Power BI** dashboard with KPIs, visuals, forecasting view.
  - ✓ **MySQL** as a dynamic backend for live updates & enterprise readiness.
- Added **predictive analytics (Linear Regression)** for forecasting future trends.
- **Refreshable dashboard** → aligns with real-world MAANG data engineering standards.
- Scalable architecture ready for **production environments**

**From a basic static dashboard → to a cloud-simulated, predictive, and database-driven BI solution demonstrating full-stack data skills.**

**NETFLIX**