

•**Title:**

*Amazon Sales Analytics & Forecasting – End-to-End Data Pipeline*

•**Subtitle:**

*SQL + Power BI + Python Forecasting + Cloud Simulation (Azurite)*

•Name: Tanmay Sharma

•Role: Data Analyst | Specialized in BI, SQL, Forecasting & Cloud Integration

•**LinkedIn:** <http://www.linkedin.com/in/>

# Project Overview

## •Goal:

Build a complete sales analytics solution with:

- Data generation (Python)
- Database (MySQL)
- Visualization (Power BI)
- Forecasting (Prophet)
- Cloud Storage Simulation (Azurite)

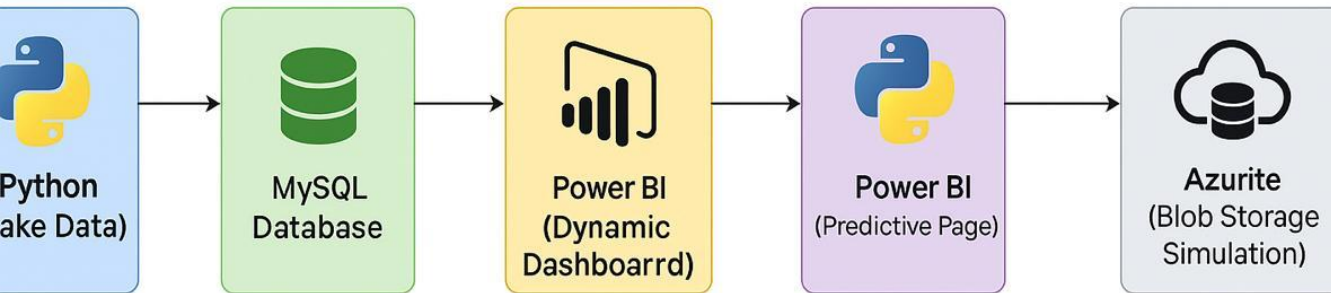
•**Tools Used:** MySQL, Python, Power BI, Azurite, Azure Storage Explorer.

## End-to-End Architecture:

### •Flow:

Python (Fake Data) → MySQL DB → Power BI → Python  
Forecast → Power BI Predictive Page → Azurite (Blob  
Simulation)

## End-to-End Architecture



# Data Generation:

## •Description:

- Python script created **Customers, Products, Orders, Order\_Details.**
- Libraries: Faker, Pandas, Random.

```
Amazon.py > ...
1  import pandas as pd
2  import random
3  from faker import Faker
4  from datetime import datetime
5
6  fake = Faker()
7
8  # Configuration
9  NUM_CUSTOMERS = 2000
10 NUM_PRODUCTS = 500
11 NUM_ORDERS = 10000
12 categories = ["Electronics", "Home", "Fashion", "Beauty", "Sports", "Books"]
13 regions = ["North America", "Europe", "Asia", "South America"]
14 payment_methods = ["Credit Card", "Debit Card", "PayPal", "Net Banking", "UPI"]
15
16 # Generate Customers Table
17 customers = []
18 for i in range(1, NUM_CUSTOMERS + 1):
19     customers.append([
20         i,
21         fake.name(),
22         fake.email(),
23         fake.country()
24     ])
25 customers_df = pd.DataFrame(customers, columns=["CustomerID", "Name", "Email", "Country"])
26
27 # Generate Products Table
28 products = []
29 for i in range(1, NUM_PRODUCTS + 1):
30     category = random.choice(categories)
31     price = round(random.uniform(10, 500), 2)
32     products.append([i, f"{category} Product {i}", category, price])
33 products_df = pd.DataFrame(products, columns=["ProductID", "ProductName", "Category", "Price"])
34
35 # Generate Orders and Order Details
36 orders = []
37 order_details = []
38 order_id = 1
39 for _ in range(NUM_ORDERS):
40     customer_id = random.randint(1, NUM_CUSTOMERS)
41     order_date = fake.date_between(start_date='-2y', end_date='today')
42     region = random.choice(regions)
43     payment_method = random.choice(payment_methods)
44
45     orders.append([order_id, customer_id, order_date, region, payment_method])
```



## Database Setup:

### •Description:

- Created amazon\_sales schema in MySQL.
- Imported generated CSVs via **Table Data Import Wizard**.

Table Data Import

Configure Import Settings

Detected file format: csv

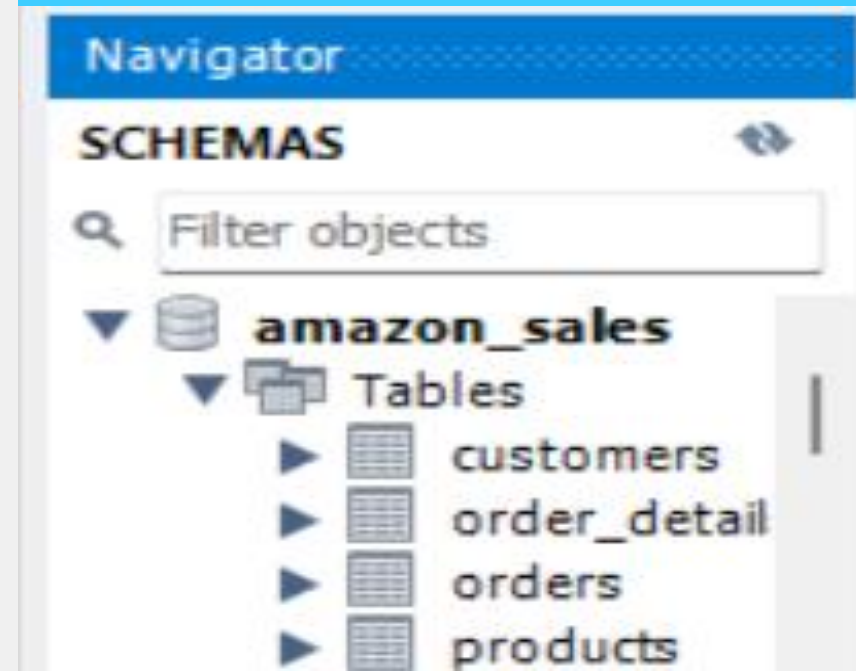
Encoding: utf-8

Columns:

<input checked="" type="checkbox"/> Source Column	Field Type
<input checked="" type="checkbox"/> CustomerID	int
<input checked="" type="checkbox"/> Name	text
<input checked="" type="checkbox"/> Email	text
<input checked="" type="checkbox"/> Country	text

CustomerID	Name	Email	Country
1	Marcus Hop...	sara14@ex...	Korea
2	Autumn Col...	bmendez@...	Swaziland
3	Wendy San...	lesparza@e...	Turks and...
4	Ryan Vega	uward@exa...	Korea
5	David Jaco...	rowestev...	Kenya

< Back Next > Cancel



# SQL Analysis:

## •Queries Performed:

- Monthly Revenue & Profit
- Top Products by Revenue
- Regional Revenue
- Customer Lifetime Value

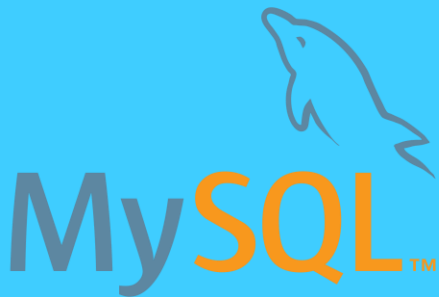


Result Grid					Filter Rows:		Export:									
	Year	Month	Monthly_Revenue	Monthly_Profit		Customer	Total_Spent		ProductName	Total_Revenue		Result Grid	Filter Rows:		Export:	
						Lonnie Smith	26192.62		Beauty Product 374	80519.25			Region	Regional_Revenue	Regional_Profit	
	2023	8	563201.93	112749.19		Carl Ramirez	23853.04		Fashion Product 435	74890.92			South America	3990996.71	804459.71	
	2023	9	616622.27	121715.19		Leslie Davis	22210.32		Electronics Product 174	73153.46			Asia	3889439.32	775746.32	
	2023	10	683420.26	136388.44		Marissa Mack	22146.91		Electronics Product 159	72057.50			Europe	3836225.18	765697.77	
	2023	11	638417.38	129880.25		Caitlin Rice	21817.42		Sports Product 188	71303.82			North America	3798811.63	763227.36	
	2023	12	658462.39	132185.30		Jeanne West	21581.38		Sports Product 190	70011.40						
	2024	1	684749.75	134721.19		Matthew Bell	21211.76		Home Product 216	69416.44						
	2024	2	591255.02	117805.81		Daniel Case	20555.06		Fashion Product 227	68657.40						
	2024	3	626382.53	123657.00		Diane Castro	20485.31		Electronics Product 104	67946.80						
	2024	4	604027.79	121255.85		Haley Wagner	20437.21		Electronics Product 249	66085.04						

# Power BI Integration:

## •Description:

- Connected to MySQL for **dynamic dashboards**.
- Created relationships:
  - Customers → Orders → Order\_Details
  - Products → Order\_Details



### Navigator

Display Options

localhost: amazon\_sales [4]

amazon\_sales.customers

amazon\_sales.order\_details

amazon\_sales.orders

amazon\_sales.products

amazon_sales.products				
ProductID	ProductName	Category	Price	amazon_sales.or
1	Books Product 1	Books	360.28	Table
2	Fashion Product 2	Fashion	119.32	Table
3	Books Product 3	Books	463.1	Table
4	Sports Product 4	Sports	386.82	Table
5	Books Product 5	Books	250.48	Table
6	Electronics Product 6	Electronics	397.99	Table
7	Beauty Product 7	Beauty	237.41	Table
8	Sports Product 8	Sports	196.75	Table
9	Sports Product 9	Sports	59.88	Table
10	Home Product 10	Home	451.43	Table
11	Fashion Product 11	Fashion	128.3	Table
12	Sports Product 12	Sports	164.03	Table
13	Beauty Product 13	Beauty	394.89	Table
14	Electronics Product 14	Electronics	347.32	Table
15	Beauty Product 15	Beauty	295.21	Table
16	Electronics Product 16	Electronics	211.06	Table
17	Electronics Product 17	Electronics	279.45	Table
18	Sports Product 18	Sports	268.78	Table
19	Beauty Product 19	Beauty	353.88	Table
20	Fashion Product 20	Fashion	235.82	Table
21	Books Product 21	Books	448.8	Table
22	Home Product 22	Home	271.36	Table
23	Books Product 23	Books	89.33	Table

Select Related Tables

Load

Transform Data

Cancel

### 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;amazon\_sales
- t:\github\financial report\amazon.scripts\forecast.csv
- t:\github\financial report\ama...on.scripts\monthly\_revenue.csv

Change Source...

Export PBIDS

Edit Permissions...

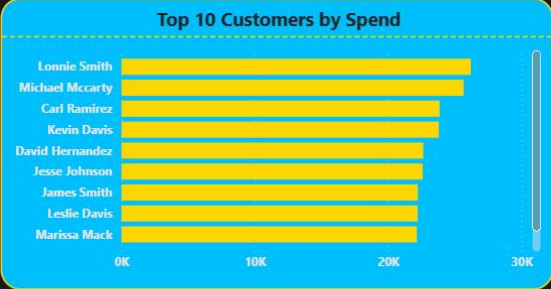
Clear Permissions

Edit Tables

Close

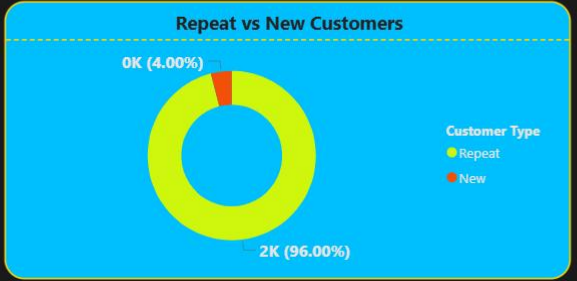


"Amazon Customer Insights Dashboard"

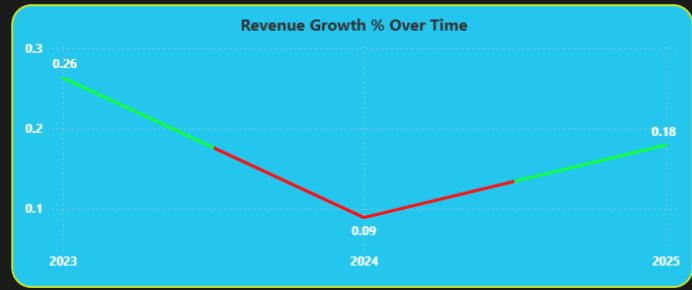


This page answers:

- Who are the top spending customers?
- Where are customers located?
- How many repeat vs new customers exist?

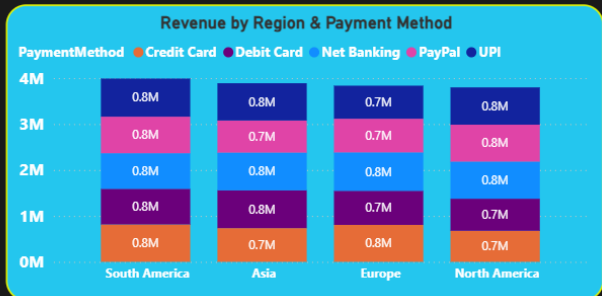


"Amazon Growth & Profitability"

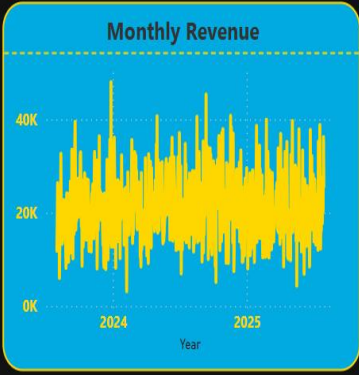
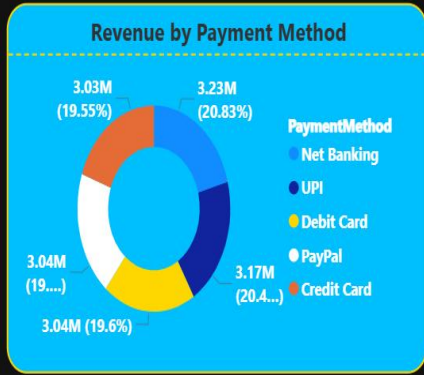


Profit Margin by Region & Category

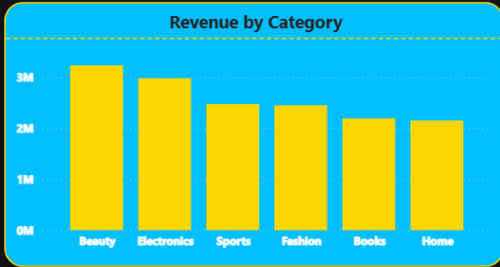
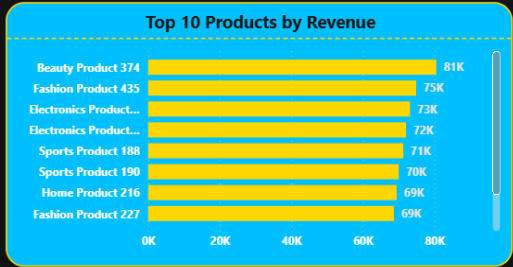
Region	Beauty	Books	Electronics	Fashion	Home	Sports	Total
Asia	0.20	0.20	0.20	0.20	0.20	0.20	0.20
Europe	0.20	0.20	0.20	0.20	0.20	0.20	0.20
North America	0.20	0.20	0.20	0.20	0.20	0.20	0.20
South America	0.20	0.20	0.20	0.20	0.20	0.20	0.20
Total	0.20	0.20	0.20	0.20	0.20	0.20	0.20



"Amazon Sales Performance Summary"

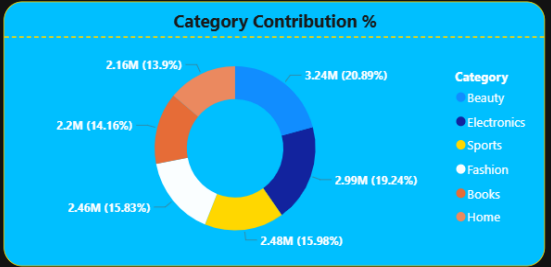


"Amazon Product Performance Analysis"



This page answers:

- Which products are generating the most revenue?
- Which categories contribute the most?



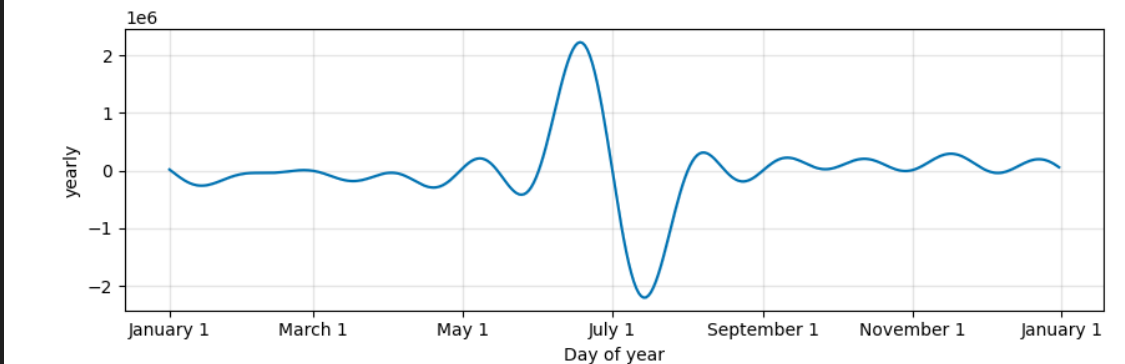
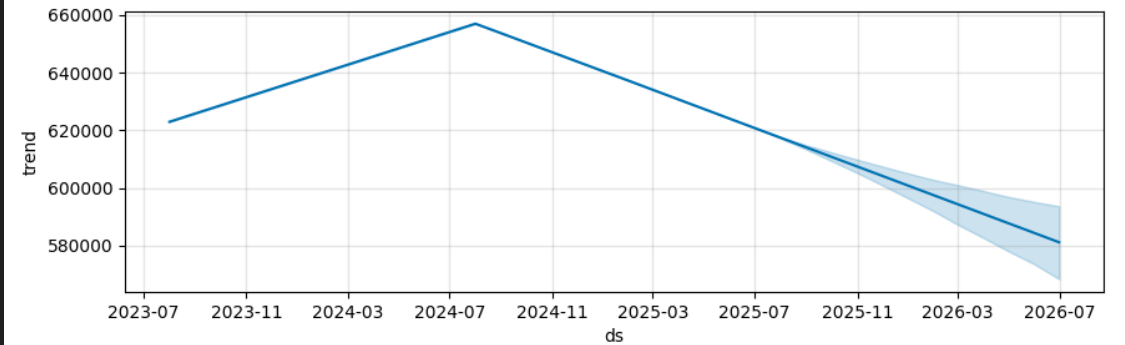


# Predictive Analysis:

## •Description:

- Forecasted next 12 months using **Prophet**.
- Imported forecast.csv into Power BI

```
forecast.py > ...
1 # amazon_forecast.py
2
3 import pandas as pd
4 from prophet import Prophet
5 import matplotlib.pyplot as plt
6
7 # Step 1: Load Historical Data
8 # Ensure your monthly_revenue.csv has columns: Month, Monthly_Revenue
9 df = pd.read_csv("monthly_revenue.csv")
10
11 # Rename columns as required by Prophet
12 df.rename(columns={"Month": "ds", "Monthly_Revenue": "y"}, inplace=True)
13
14 # Convert 'ds' to datetime
15 df['ds'] = pd.to_datetime(df['ds'])
16
17 # Step 2: Initialize Prophet Model
18 model = Prophet(yearly_seasonality=True, daily_seasonality=False)
19 model.fit(df)
20
21 # Step 3: Create Future DataFrame for 12 months
22 future = model.make_future_dataframe(periods=12, freq='M')
23
24 # Step 4: Forecast
25 forecast = model.predict(future)
26
27 # Step 5: Save Forecast to CSV
28 forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].to_csv("forecast.csv", index=False)
29
30 # Step 6: Plot Forecast
31 fig1 = model.plot(forecast)
32 plt.title("Amazon Sales Forecast (Next 12 Months)", fontsize=16)
33 plt.xlabel("Date")
34 plt.ylabel("Revenue")
35 plt.show()
36
37 # Step 7: Plot Components (Trend, Seasonality)
38 fig2 = model.plot_components(forecast)
39 plt.show()
40
41 print("Forecast completed! File saved as forecast.csv")
42
```



## "Predictive Analysis – Future Revenue Forecast"

Next Month "Predicted  
Revenue"

22.86M

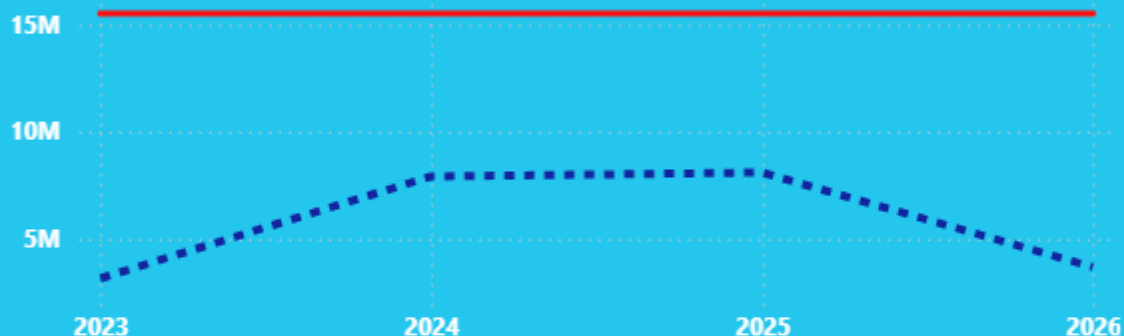
Confidence Range

₹963,446 - ₹990,326

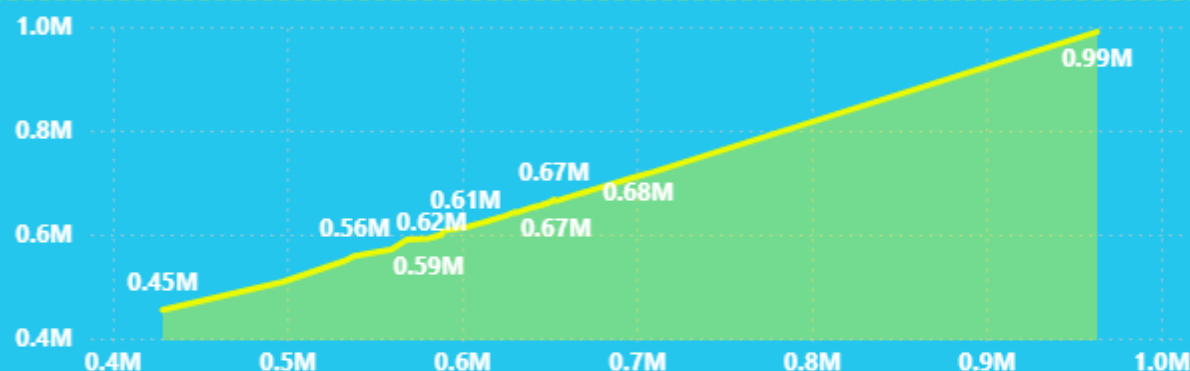
Forecast Growth %

0.37

(Historical + Forecast)



Confidence Interval



Forecast Table

Year	Quarter	Month	Day	Sum of yhat	Sum of yhat_lower	Sum of yhat_upper
2023	Qtr 3	August	1	5,64,882.16	5,59,620.91	5,70,669.32
2023	Qtr 3	September	1	6,17,687.63	6,12,168.24	6,22,976.50
2023	Qtr 4	October	1	6,83,321.61	6,77,669.11	6,88,929.83
2023	Qtr 4	November	1	6,36,861.36	6,31,698.20	6,42,313.70
2023	Qtr 4	December	1	6,55,622.50	6,50,035.89	6,61,401.98
2024	Qtr 1	January	1	6,81,121.15	6,75,401.34	6,86,893.37
2024	Qtr 1	February	1	5,87,010.01	5,81,003.29	5,92,692.94
Total				2,28,58,820.07	2,26,21,724.62	2,30,92,550.68

# Cloud Simulation

## •Description:

- Used **Azurite** for Azure Blob Storage simulation.
- Uploaded all project files via Azure Storage Explorer.

```
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
```

Microsoft Azure Storage Explorer

File Edit View Help

EXPLORER

Search for resources

Collapse all

Refresh all

Quick Access

Emulator & Attached

Storage Accounts

(Attached Containers)

(Emulator - Default Ports) (Key)

Amazon24 (Key)

Blob Containers

amazon

netflix

View all

Queues

Tables

Azurite24 (Key)

Get Started amazon

Upload Download Open Preview New Folder Select All Properties Delete Undelete Manage History Folder Statistics Refresh

Active blobs (default) amazon > Amazon.Scripts

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Ty
Amazon.py	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	text/x-pythc
customers.csv	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	application/
Figure_1.png	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	image/png
Figure_2.png	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	image/png
forecast.py	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	text/x-pythc
forecast.csv	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	application/
monthly_revenue.csv	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	application/
order_details.csv	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	application/
orders.csv	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	application/
products.csv	Hot (inferred)	02-08-2025 15:06	02-08-2025 15:06	Block Blob	application/

Showing 1 to 10 of 10 cached items

Clear completed Clear successful

Transfer of 'T:\GitHub\Financial report\Amazon.Scripts\' to 'devstoreaccount1/amazon/' complete: 10 items transferred (used SAS, discovery completed)

Started at: 02-08-2025 15:06, Duration: 5 seconds

Successfully created blob container 'amazon'

Successfully added new connection.

Copy AzCopy Command to Clipboard

Actions Properties

Node Display Name amazon

URL http://127.0.0.1:10000/devstor

Custom Domain

Type Blob Container

HNS Enabled false

Lease State available

Lease Status unlocked

Public Read Access off

## Key Learnings

- ETL Pipeline Design
- SQL & DAX for BI
- ML Forecasting (Prophet)
- Cloud Readiness with Azurite

## Tagline:

*End-to-End Sales Analytics Project using MAANG standards.*

Link to Github repository: [https://github.com/Tanu272004/Amazon\\_Analysis\\_Project.git](https://github.com/Tanu272004/Amazon_Analysis_Project.git)

Python (Generate Data)



Save CSV → Upload to Azurite (Azure Blob Simulation)



Download from Azurite → Import into MySQL



Power BI → Dynamic Dashboard



Python Forecast (Prophet) → Forecast.csv → Power BI Predictive Page



Screenshots & PPT → GitHub + LinkedIn