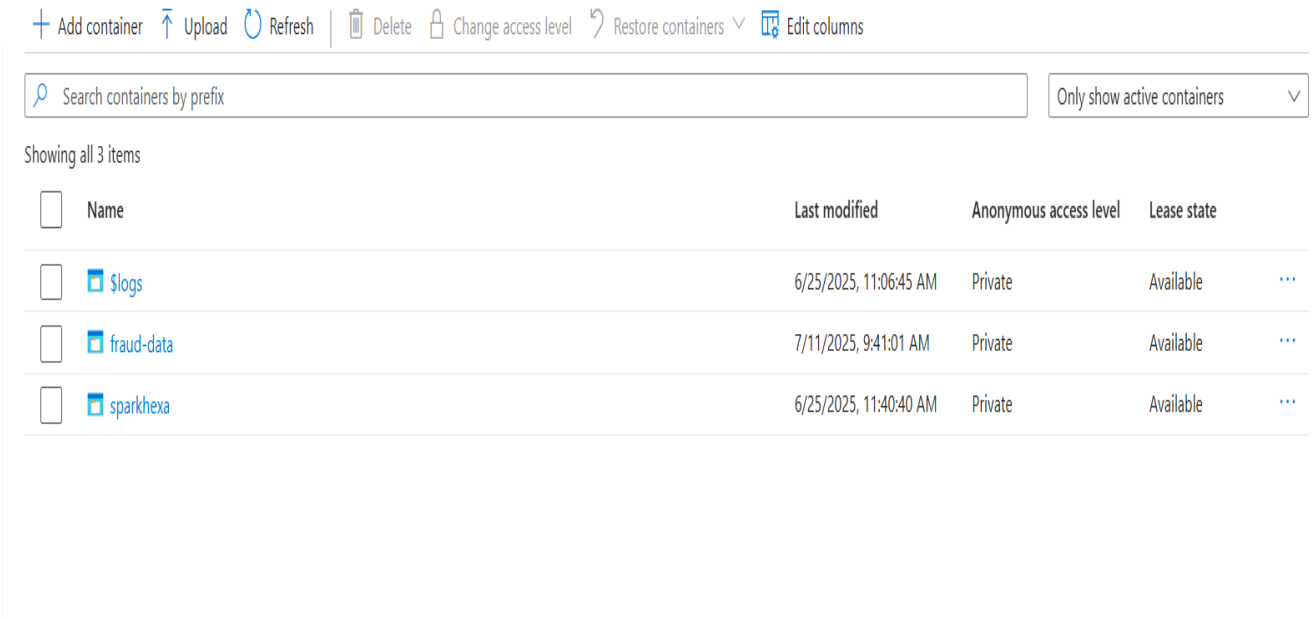
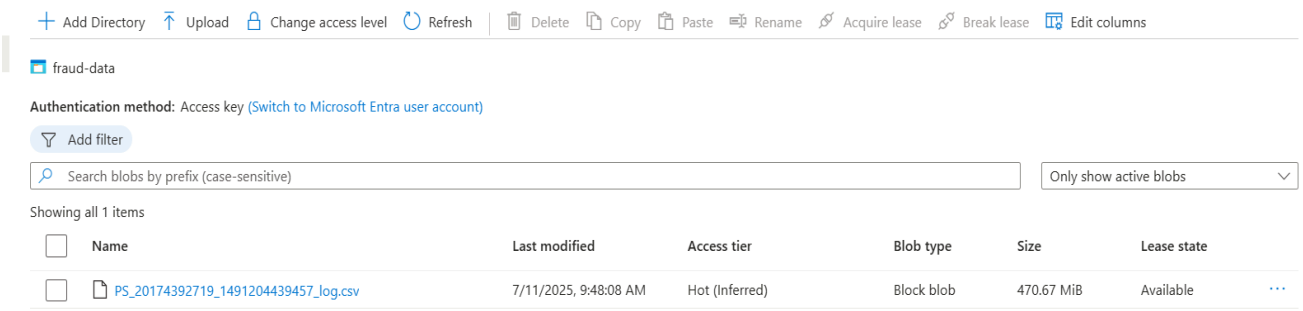
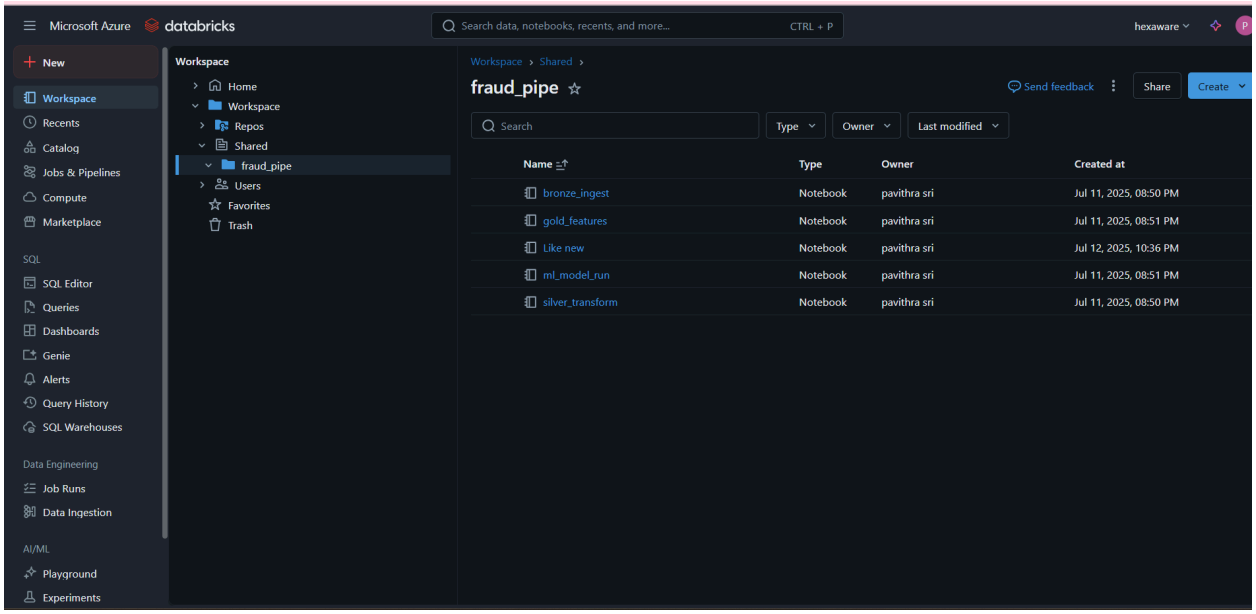
1) Storage Setup

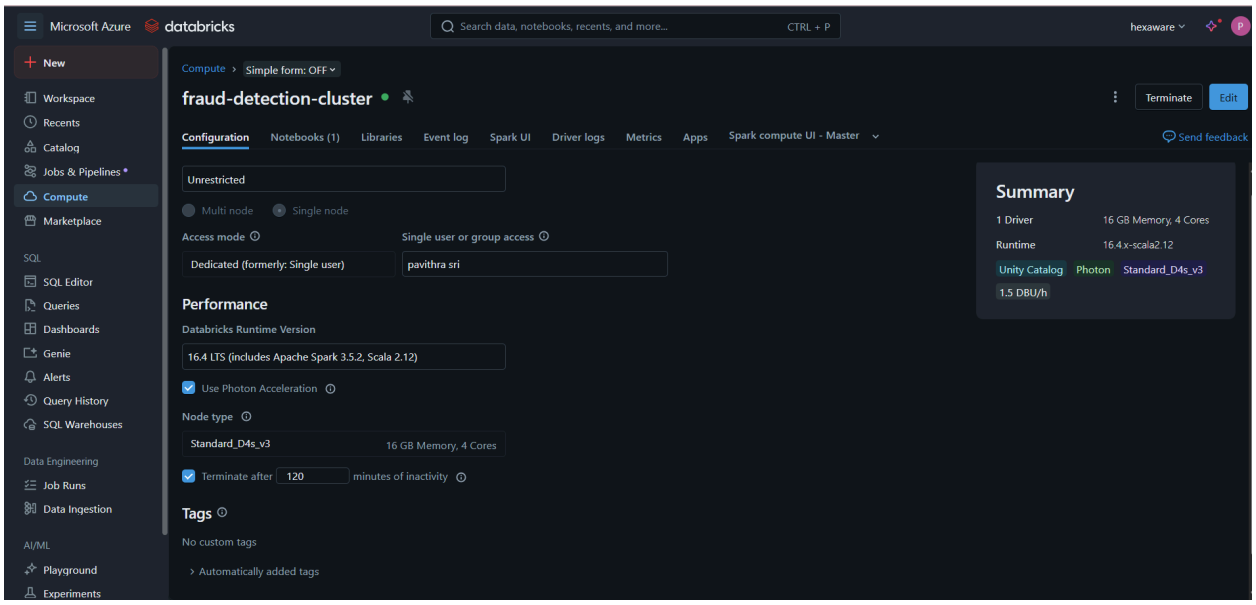




2) Notebooks Overview



3) Cluster in databricks



4) Reading and printing initial bronze table

```
df_bronze: pyspark.sql.dataframe.DataFrame = [step: integer, type: string ... 9 more fields]
 |-- type: string (nullable = true)
 |-- amount: double (nullable = true)
 |-- nameOrig: string (nullable = true)
 |-- oldbalanceOrg: double (nullable = true)
 |-- newbalanceOrig: double (nullable = true)
 |-- nameDest: string (nullable = true)
 |-- oldbalanceDest: double (nullable = true)
 |-- newbalanceDest: double (nullable = true)
 |-- isFraud: integer (nullable = true)
 |-- isFlaggedFraud: integer (nullable = true)

+----+--------+--------+-----------+-------------+--------------+-----------+--------------+--------------+-------+--------------+
|step|    type|  amount|   nameOrig|oldbalanceOrg|newbalanceOrig|   nameDest|oldbalanceDest|newbalanceDest|isFraud|isFlaggedFraud|
+----+--------+--------+-----------+-------------+--------------+-----------+--------------+--------------+-------+--------------+
|   1| PAYMENT| 9839.64|C1231006815|     170136.0|     160296.36|M1979787155|           0.0|           0.0|      0|             0|
|   1| PAYMENT| 1864.28|C1666544295|      21249.0|      19384.72|M2044282225|           0.0|           0.0|      0|             0|
|   1|TRANSFER|   181.0|C1305486145|        181.0|           0.0| C553264065|           0.0|           0.0|      1|             0|
|   1|CASH_OUT|   181.0| C840083671|        181.0|           0.0|  C38997010|       21182.0|           0.0|      1|             0|
|   1| PAYMENT|11668.14|C2048537720|      41554.0|      29885.86|M1230701703|           0.0|           0.0|      0|             0|
+----+--------+--------+-----------+-------------+--------------+-----------+--------------+--------------+-------+--------------+
only showing top 5 rows
```

5) Initial dataframe

```
df_raw: pyspark.sql.dataframe.DataFrame
        step: string
        type: string
        amount: string
        nameOrig: string
        oldbalanceOrg: string
        newbalanceOrig: string
        nameDest: string
        oldbalanceDest: string
        newbalanceDest: string
        isFraud: string
        isFlaggedFraud: string
        _rescued_data: string

<pyspark.sql.streaming.query.StreamingQuery at 0x7f0de0db7890>
```

6) Bronze table:-

Table

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest
1	249	PAYMENT	3119.51	C1515157403	15175.2	12055.69	M880864628	0.0
2	249	PAYMENT	11494.08	C1197653668	12055.69	561.61	M1158991064	0.0
3	249	PAYMENT	14220.43	C1682344014	561.61	0.0	M1461592831	0.0
4	249	PAYMENT	334.8	C89862438	0.0	0.0	M1716146009	0.0
5	249	PAYMENT	13467.2	C907984678	0.0	0.0	M271674048	0.0
6	249	PAYMENT	34657.33	C1279245281	0.0	0.0	M2017138104	0.0
7	249	PAYMENT	4064.65	C991912470	0.0	0.0	M551774431	0.0
8	249	PAYMENT	10756.58	C288501649	0.0	0.0	M830766810	0.0
9	249	PAYMENT	2027.2	C1900786990	0.0	0.0	M1247120968	0.0
10	249	PAYMENT	6208.09	C1243315764	0.0	0.0	M1492892419	0.0
11	249	PAYMENT	497.74	C1949669367	0.0	0.0	M309620983	0.0
12	249	PAYMENT	403.47	C59721299	0.0	0.0	M103401090	0.0
13	249	PAYMENT	1338.56	C1585148831	0.0	0.0	M1866023791	0.0
14	249	PAYMENT	287.29	C385694981	0.0	0.0	M1579738254	0.0

7) Clean and transform the data and saving to the delta table:-

Schema Details History

```
step: string
type: string
amount: string
nameOrig: string
oldbalanceOrg: string
newbalanceOrig: string
nameDest: string
oldbalanceDest: string
newbalanceDest: string
isFraud: string
isFlaggedFraud: string
_rescued_data: string
```

```
df_silver: pyspark.sql.dataframe.DataFrame
    step: string
    type: string
    amount: string
    originator: string
    oldbalanceOrg: string
    newbalanceOrig: string
    receiver: string
    oldbalanceDest: string
    newbalanceDest: string
    _rescued_data: string
    is_fraud: integer
    is_flagged_fraud: integer

<pyspark.sql.streaming.query.StreamingQuery at 0x7f0de0db5910>
```

8) Displaying silver transformed table:-

Table

	amount	originator	oldbalanceOrg	newbalanceOrig	receiver	oldbalanceDest	newbalanceDest
1	3119.51	C1515157403	15175.2	12055.69	M880864628	0.0	0.0
2	11494.08	C1197653668	12055.69	561.61	M1158991064	0.0	0.0
3	14220.43	C1682344014	561.61	0.0	M1461592831	0.0	0.0
4	334.8	C89862438	0.0	0.0	M1716146009	0.0	0.0
5	13467.2	C907984678	0.0	0.0	M271674048	0.0	0.0
6	34657.33	C1279245281	0.0	0.0	M2017138104	0.0	0.0
7	4064.65	C991912470	0.0	0.0	M551774431	0.0	0.0
8	10756.58	C288501649	0.0	0.0	M830766810	0.0	0.0
9	2027.2	C1900786990	0.0	0.0	M1247120968	0.0	0.0
10	6208.09	C1243315764	0.0	0.0	M1492892419	0.0	0.0
11	497.74	C1949669367	0.0	0.0	M309620983	0.0	0.0
12	403.47	C59721299	0.0	0.0	M103401090	0.0	0.0
13	1338.56	C1585148831	0.0	0.0	M1866023791	0.0	0.0
14	287.29	C385694981	0.0	0.0	M1579738254	0.0	0.0

9) Feature engineering - example aggregations and saving it to the gold delta table:-

```
df_gold: pyspark.sql.dataframe.DataFrame
    originator: string
    total_transactions: long
    total_amount_sent: double
    average_amount_sent: double
    fraud_count: long
df_silver: pyspark.sql.dataframe.DataFrame
```

Schema Details History

```
step: string
type: string
amount: string
originator: string
oldbalanceOrg: string
newbalanceOrig: string
receiver: string
oldbalanceDest: string
newbalanceDest: string
_rescued_data: string
is_fraud: integer
is_flagged_fraud: integer
```

10) Displaying gold table:-

Table

	originator	total_transactions	total_amount_sent	average_amount_sent	fraud_count
1	C876714021	1	443010.87	443010.87	0
2	C224938823	1	2959.54	2959.54	0
3	C1676650606	1	34739.91	34739.91	0
4	C289153199	1	371.23	371.23	0
5	C1181048539	1	17411.44	17411.44	0
6	C1503438438	1	196626.77	196626.77	0
7	C765877097	1	63191.19	63191.19	0
8	C477278966	1	287868.3	287868.3	0
9	C326407304	1	250341.94	250341.94	0
10	C1285801989	1	90349.08	90349.08	0
11	C1424608077	1	13656.17	13656.17	0
12	C714358180	1	401340.14	401340.14	0
13	C909652139	1	29606.84	29606.84	0
14	C2124981192	1	313025.88	313025.88	0
15	C17536176	1	203336.28	203336.28	0

11)Gold table sample, schema, and count of records:-

Table

	originator	total_transactions	total_amount_sent	average_amount_sent	fraud_count
1	C876714021	1	443010.87	443010.87	0
2	C224938823	1	2959.54	2959.54	0
3	C1676650606	1	34739.91	34739.91	0
4	C289153199	1	371.23	371.23	0
5	C1181048539	1	17411.44	17411.44	0

5 rows

```
root
 |-- originator: string (nullable = true)
 |-- total_transactions: long (nullable = true)
 |-- total_amount_sent: double (nullable = true)
 |-- average_amount_sent: double (nullable = true)
 |-- fraud_count: long (nullable = true)

Gold Table Record Count: 6347542
```

12) Checking Missing Values in Gold Table

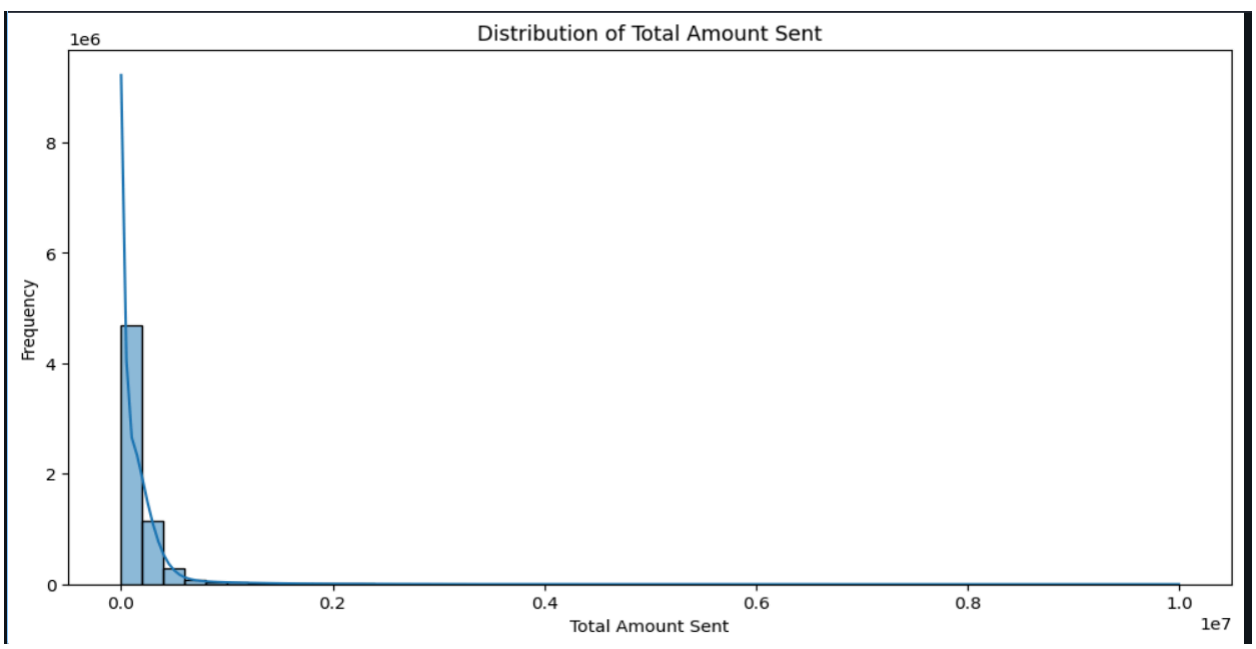
Table

	originator	total_transactions	total_amount_sent	average_amount_sent	fraud_count
1	0	0	0	0	0

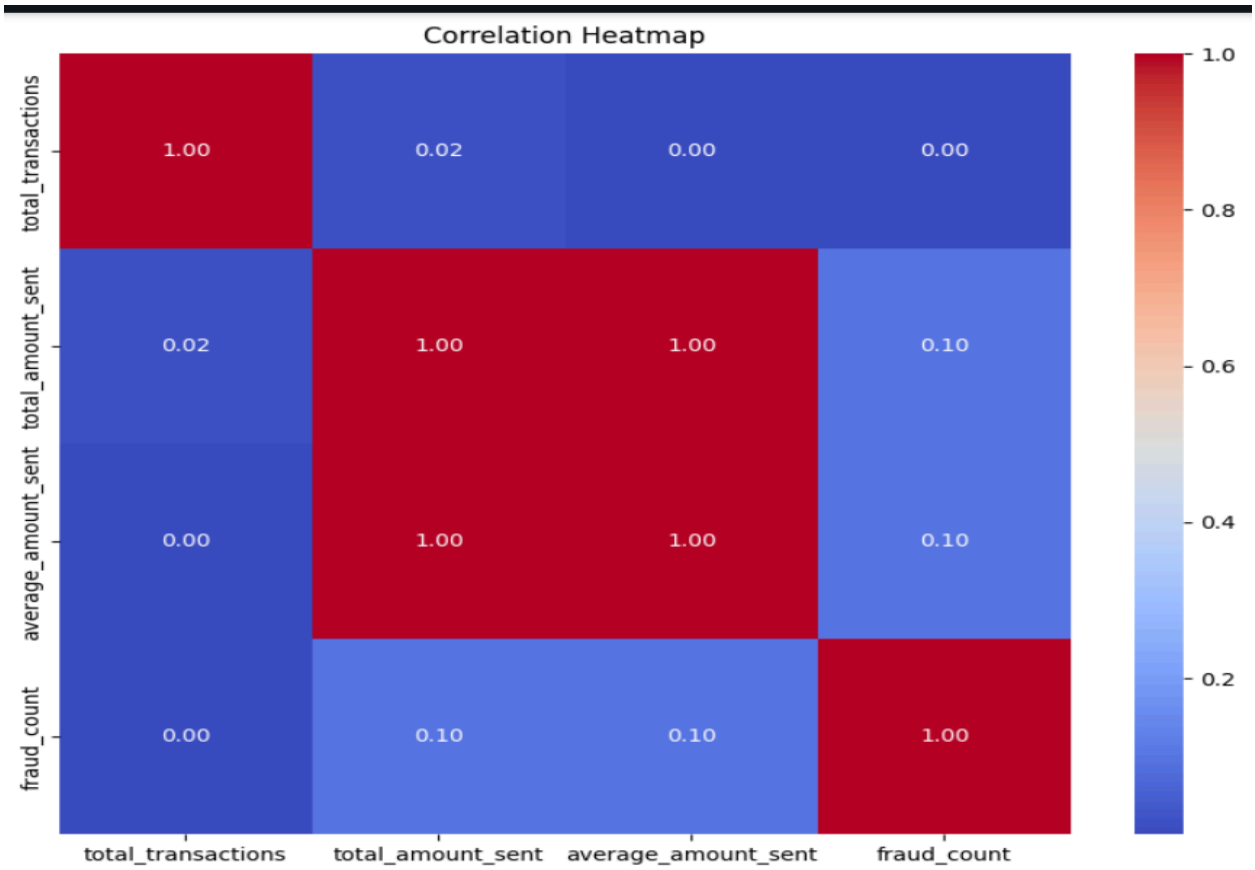
13) Convert to Pandas for sklearn

```
df_pd: pandas.core.frame.DataFrame
    total_transactions: int64
    total_amount_sent: float64
    average_amount_sent: float64
    fraud_count: int64
```

14) Histogram of Total Amount Sent



15) Correlation Heatmap



16) Train model

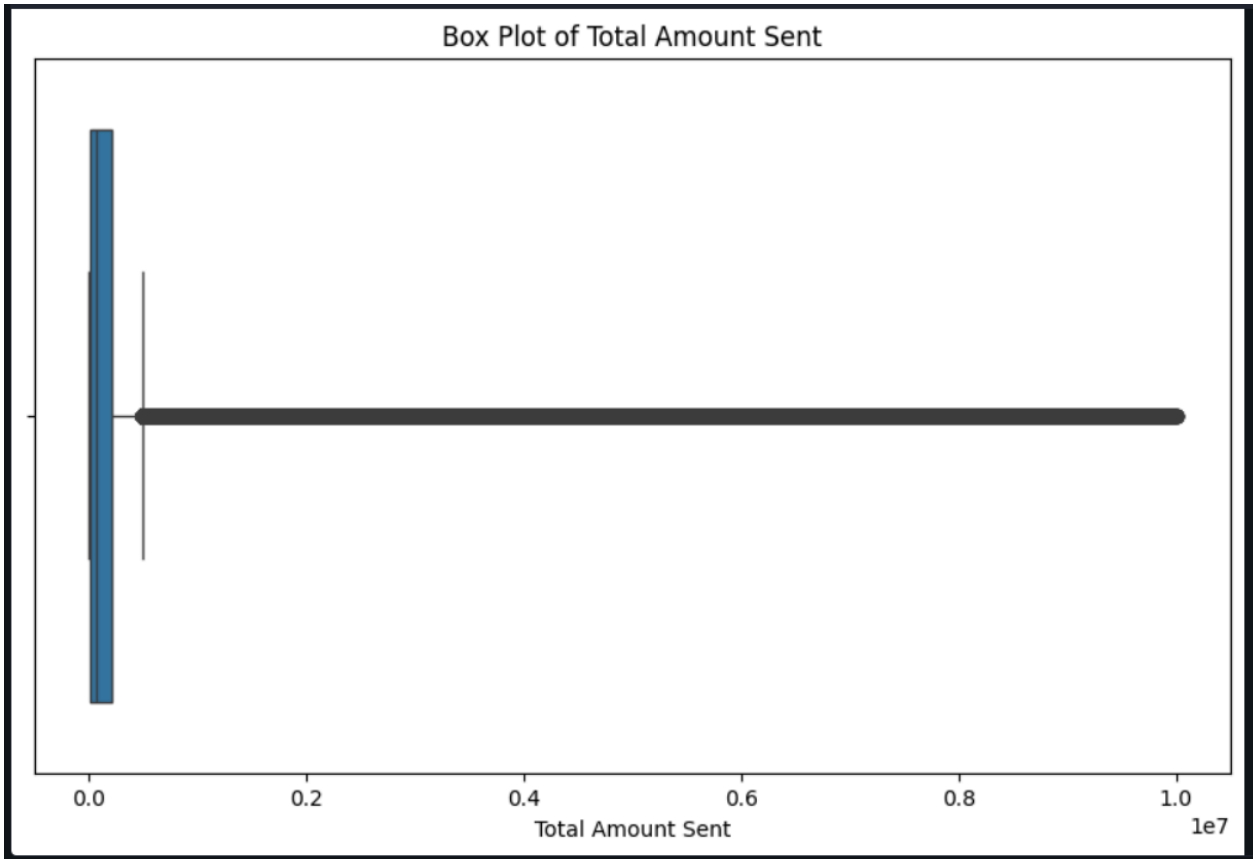
	total_transactions	total_amount_sent	average_amount_sent	fraud_count	anomaly_flag
0	1	5014.17	5014.17	0	0
1	1	15188.56	15188.56	0	0
2	1	9349.98	9349.98	0	0
3	1	35423.27	35423.27	0	0
4	1	71906.86	71906.86	0	0
5	1	80557.71	80557.71	0	0
6	1	7762.46	7762.46	0	0
7	1	8759.81	8759.81	0	0
8	1	405.30	405.30	0	0
9	1	8600.77	8600.77	0	0

17) Save final output with anomaly flag to Delta Lake

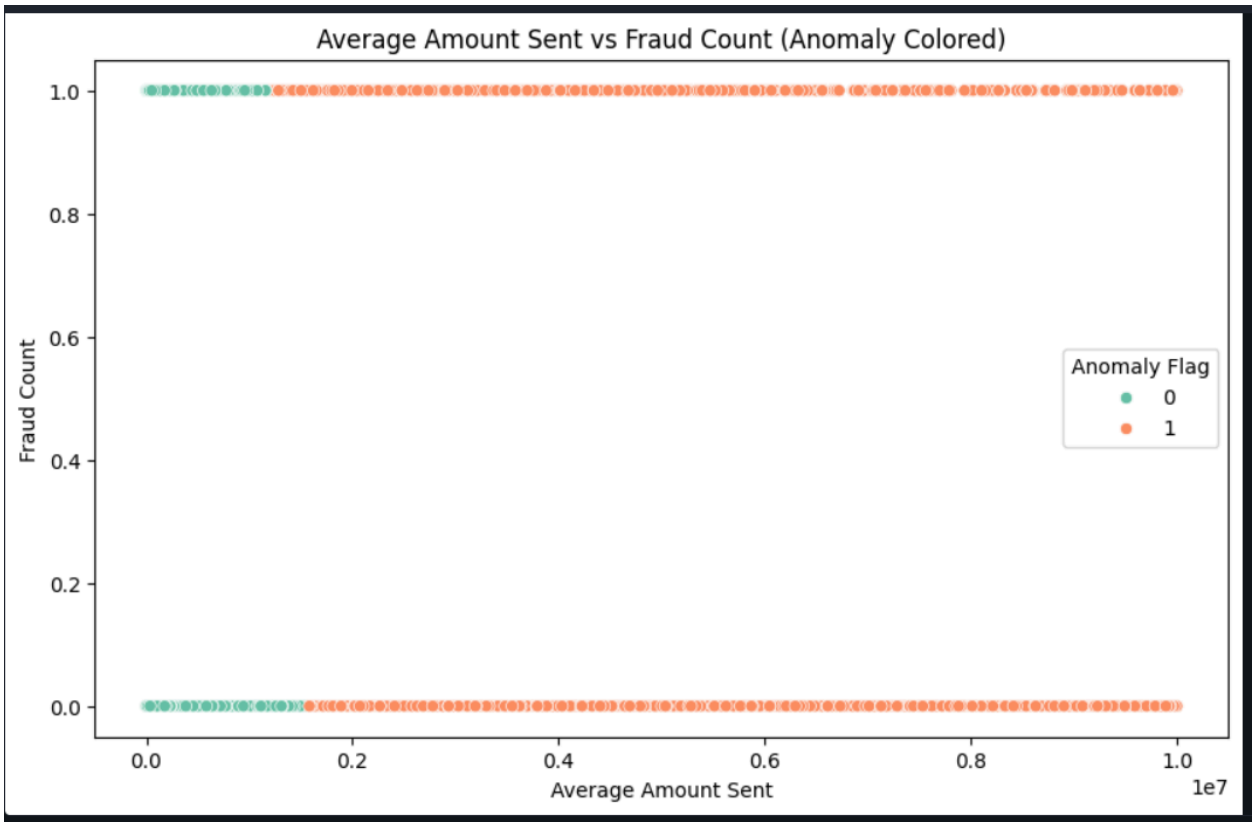
```
anomaly_sdf: pyspark.sql.dataframe.DataFrame = [total_transactions: long, total_amount_sent: double ... 3 more fields]

Anomaly detection results saved to Delta successfully.
```

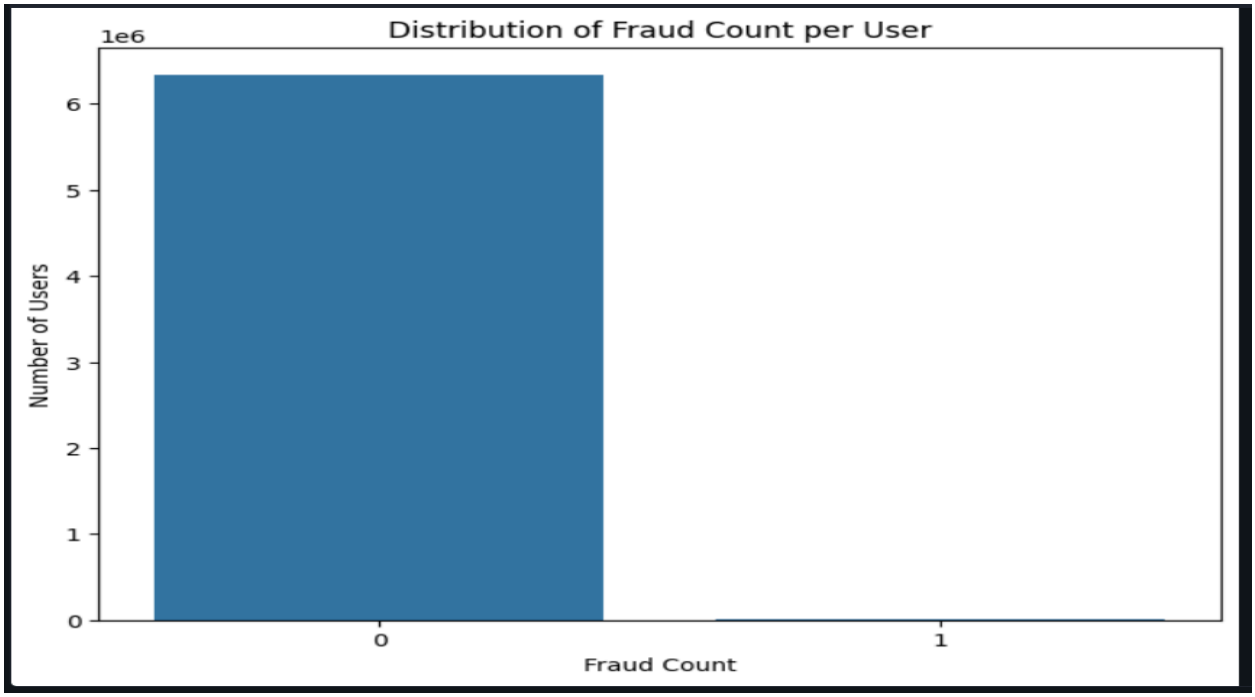
18) Box plot



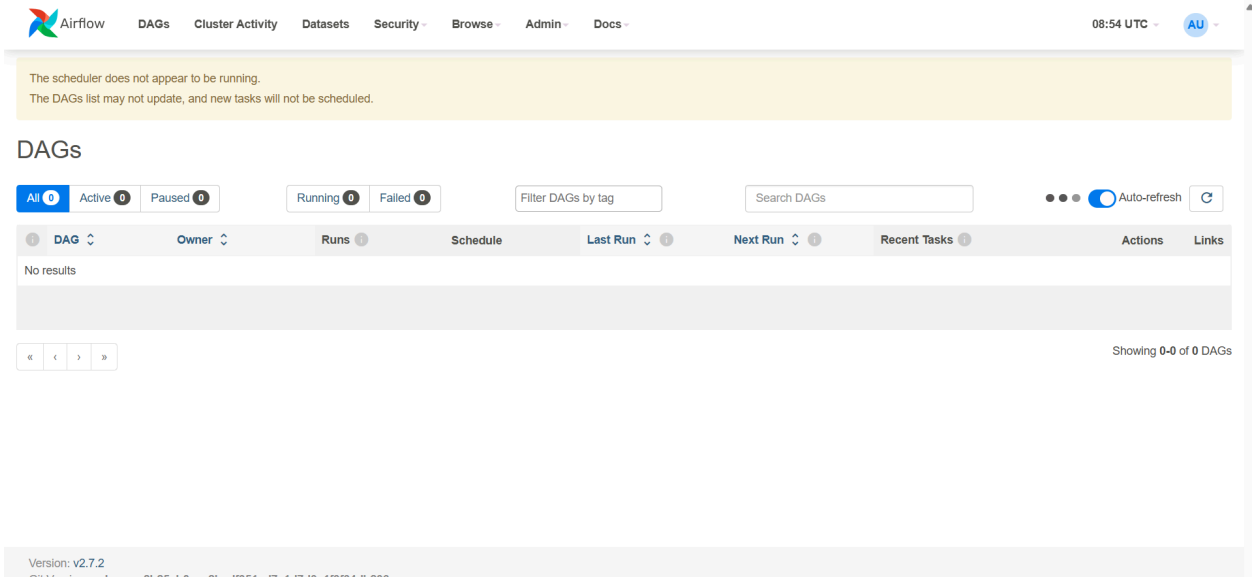
19) Anomaly colored



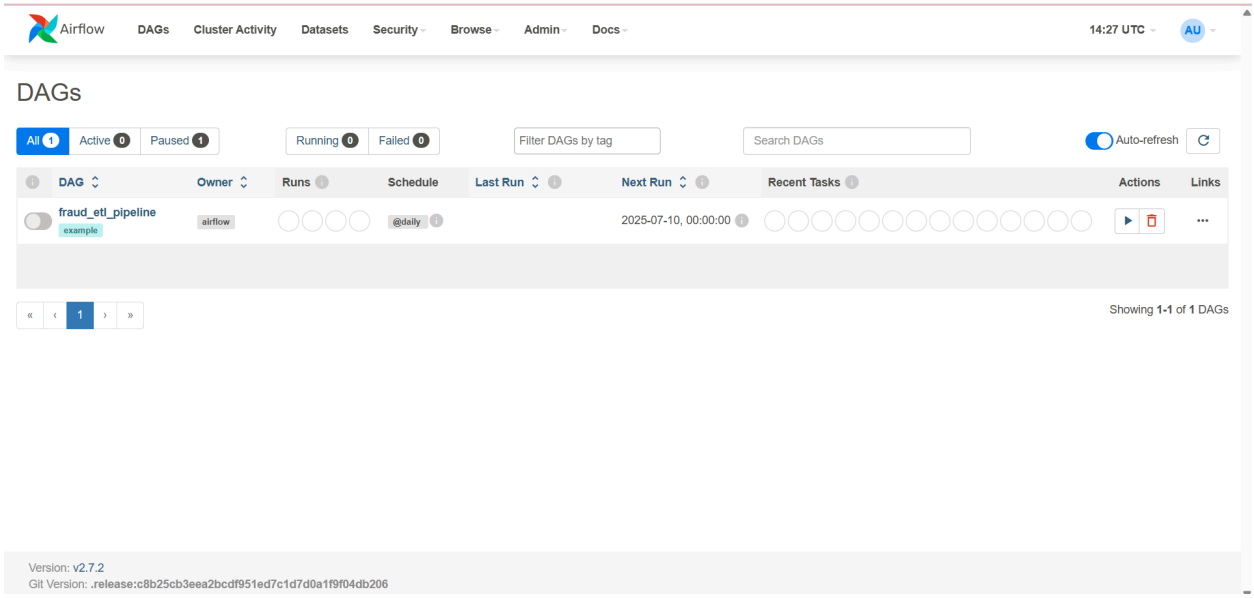
20) Bar graph



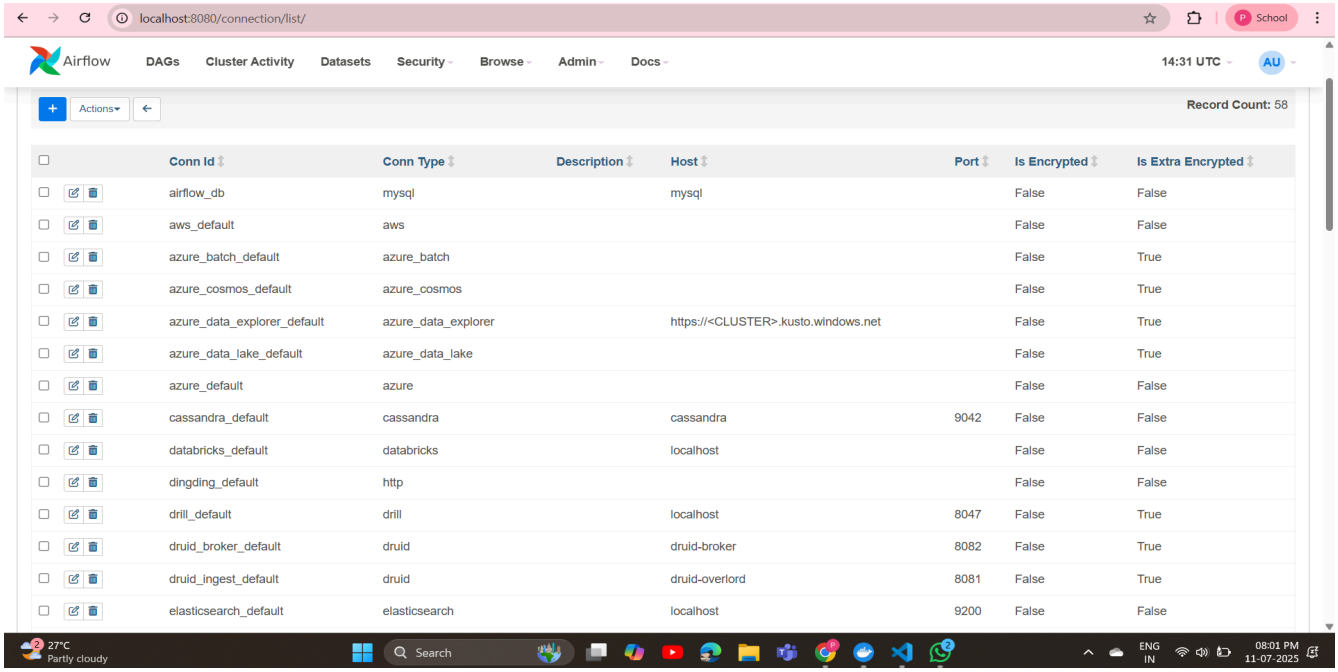
21) Airflow DAG's creation

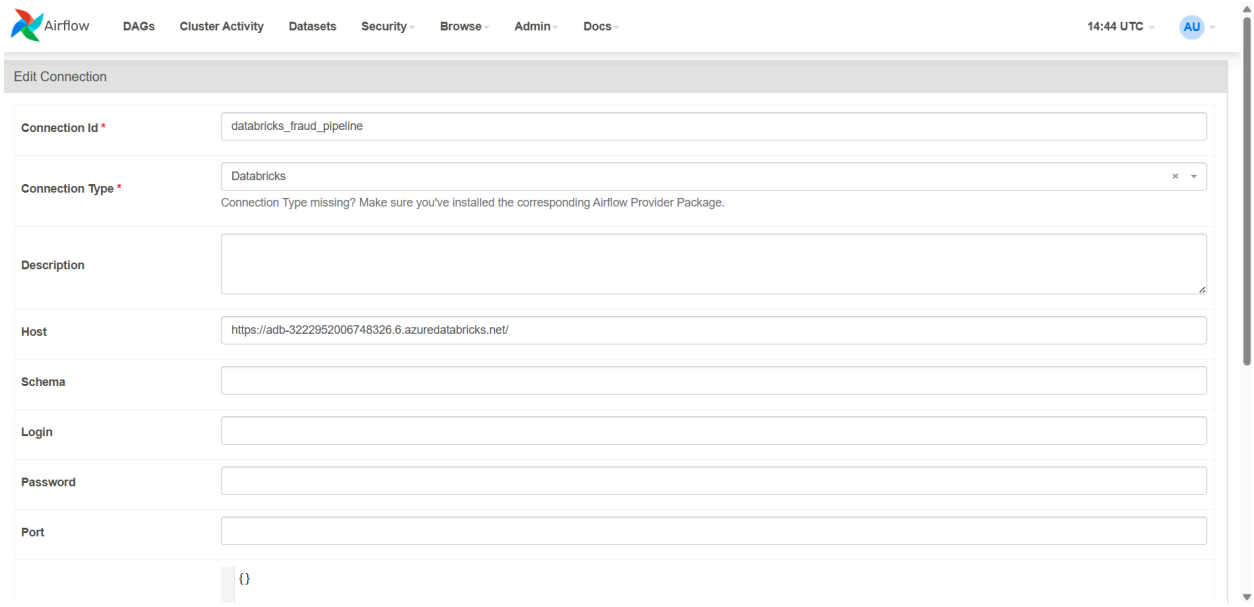


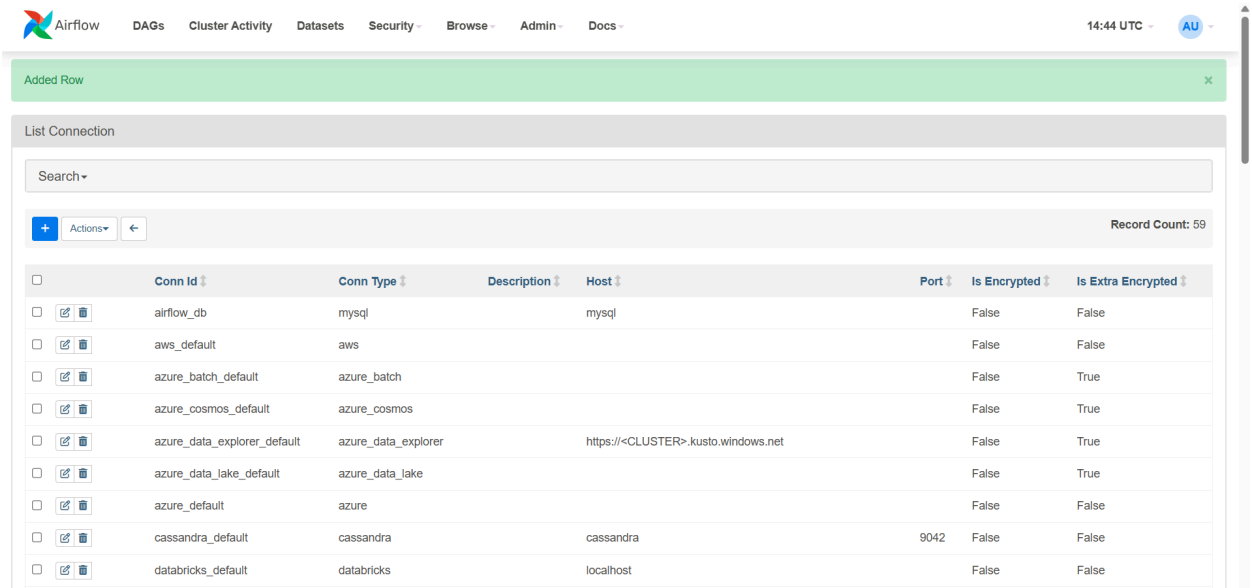
22) Dag created successfully



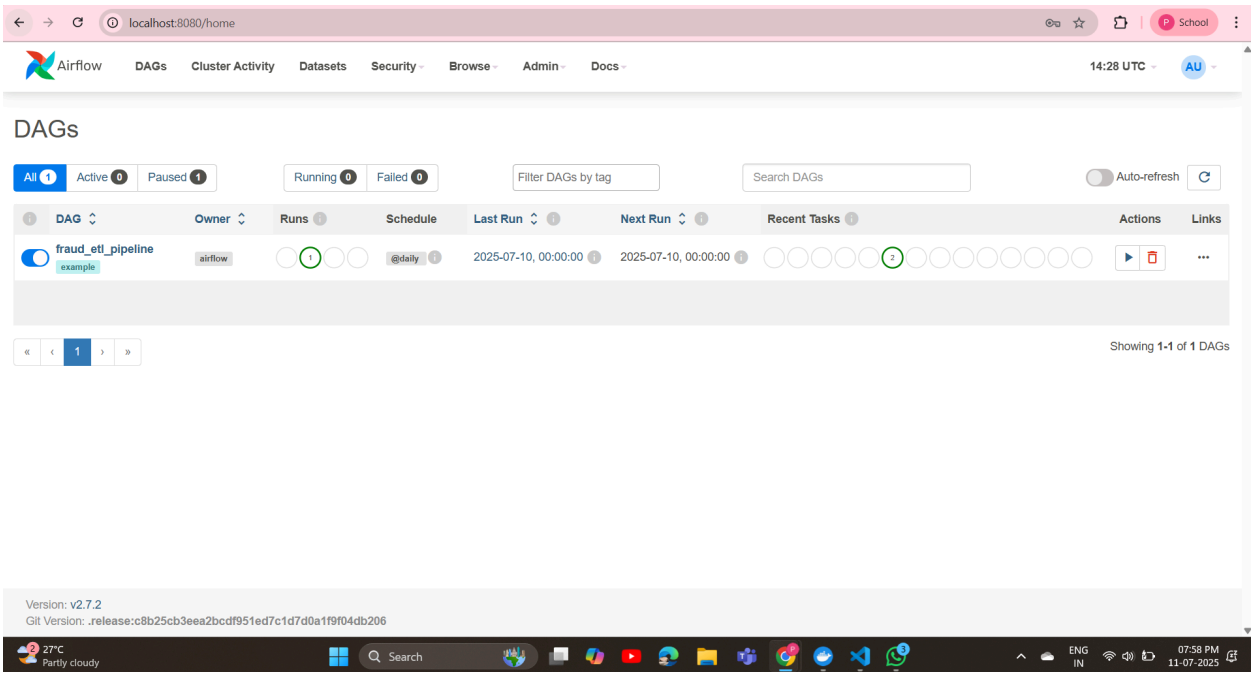
23) Creating DAG schedule



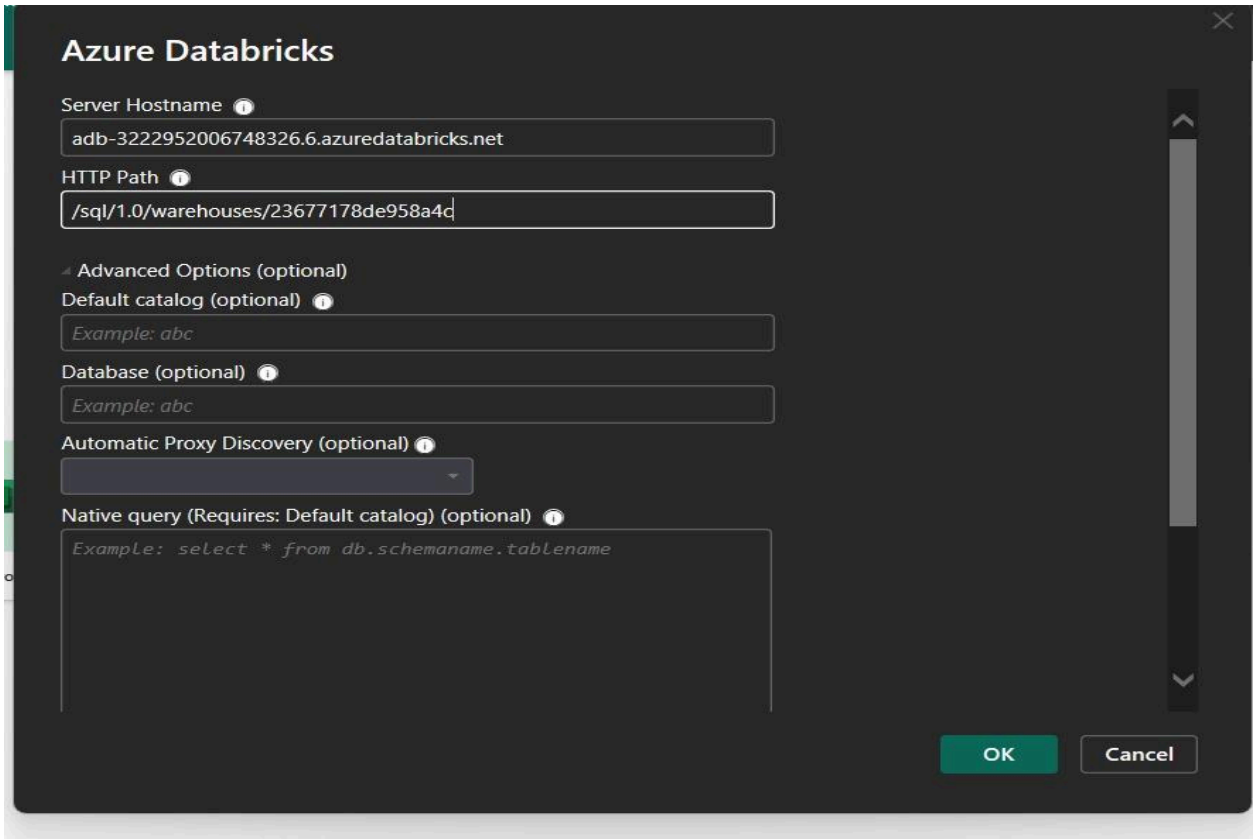




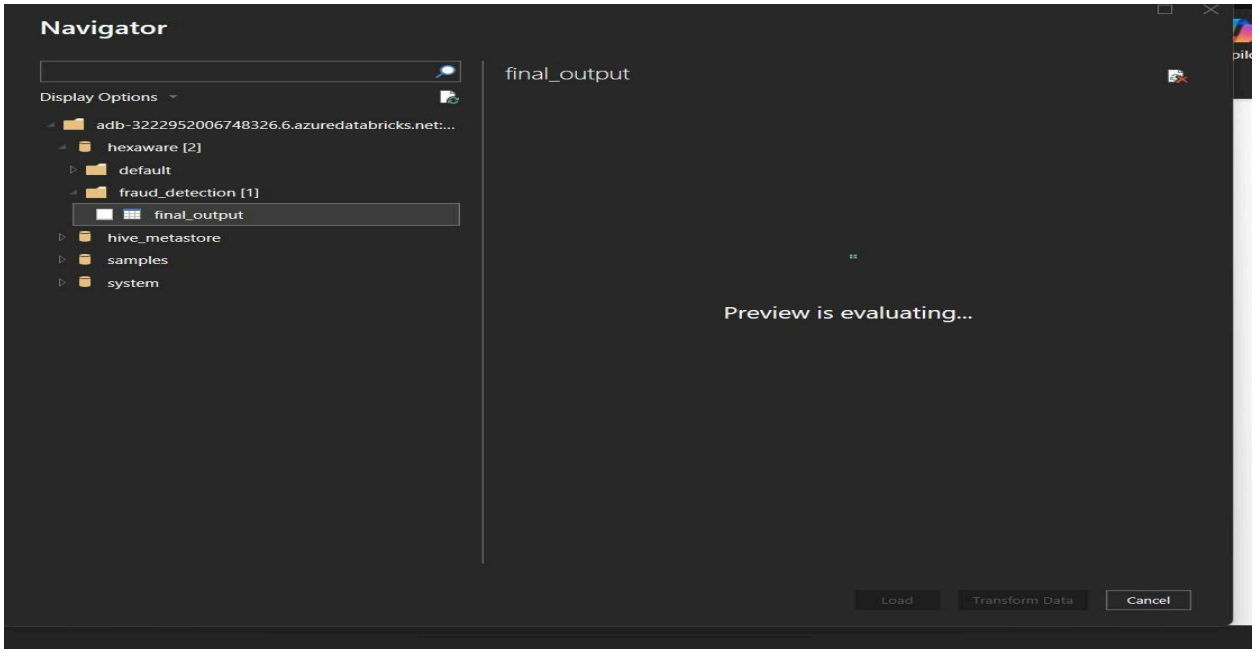
24) DAG schedule created successfully



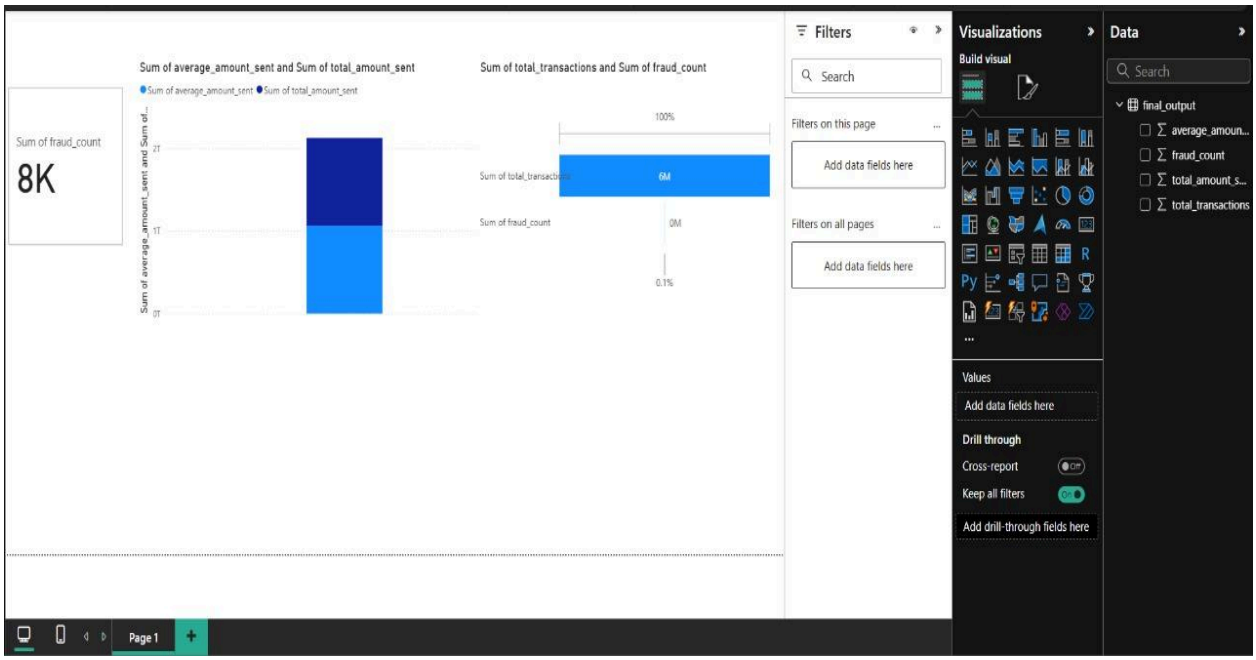
25) Power BI and Azure databricks connection



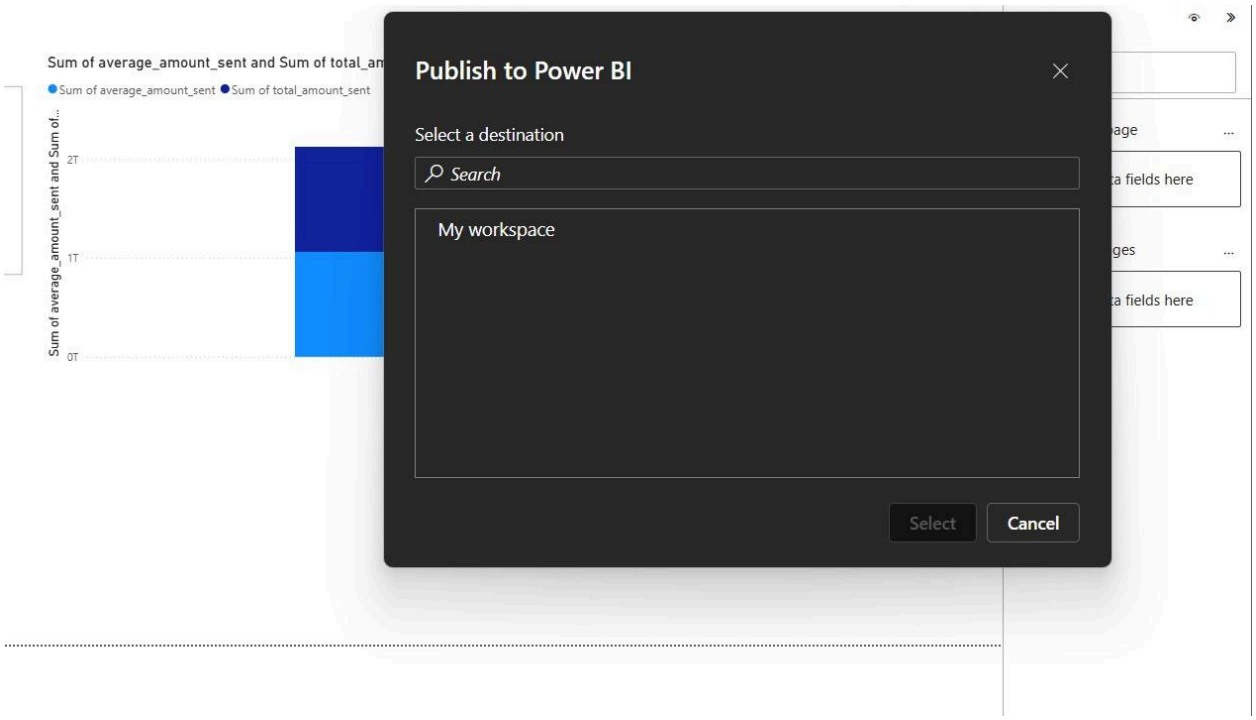
26) Connection string in power BI



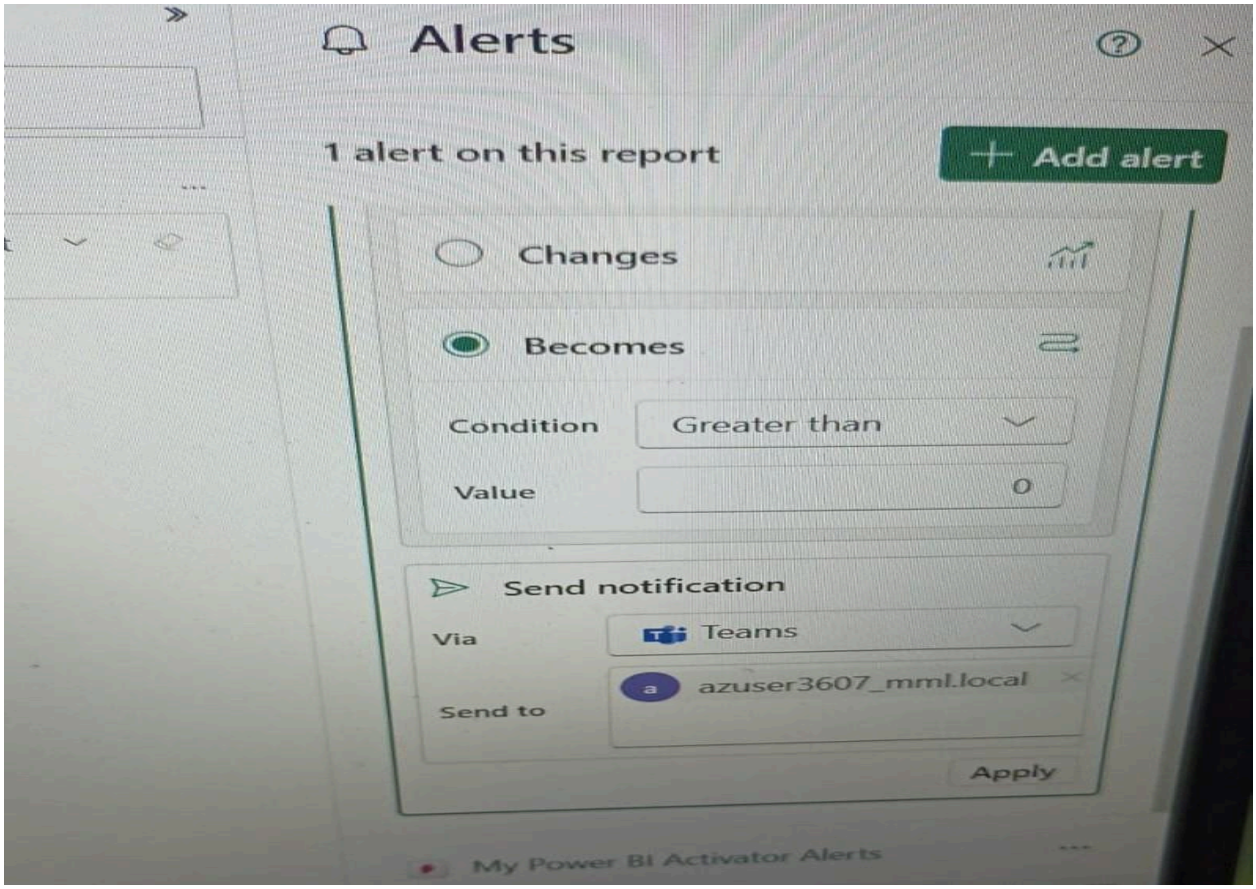
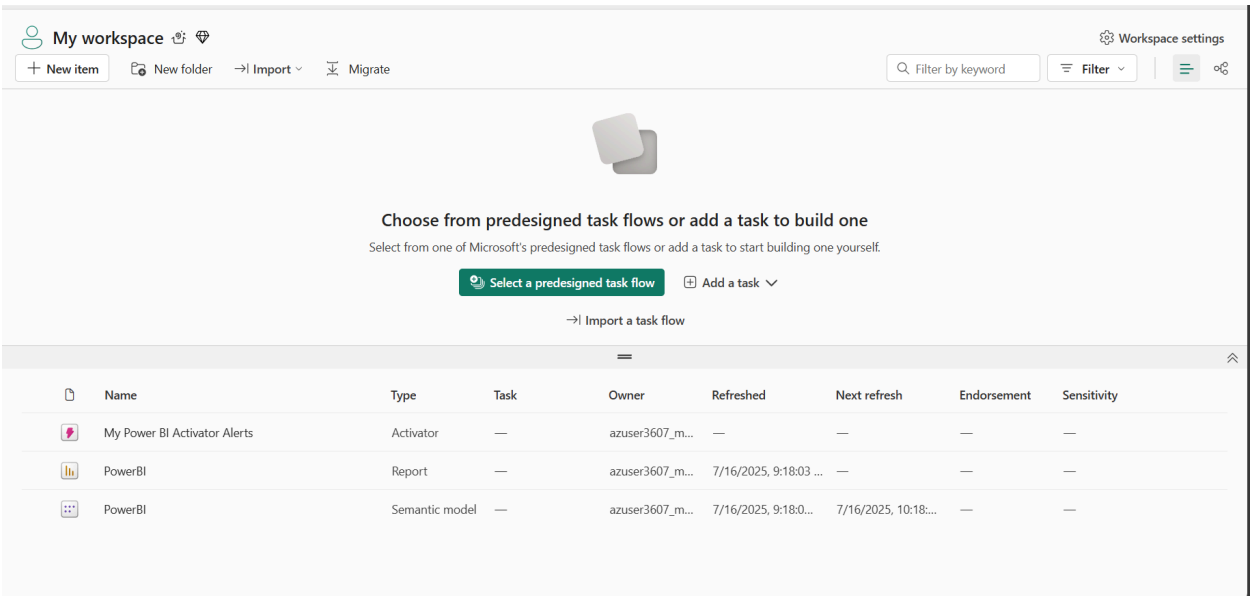
27) Power Bi visualization creation



28) Publishing to our workspace



29) Creating alerts



30) Power BI workspace