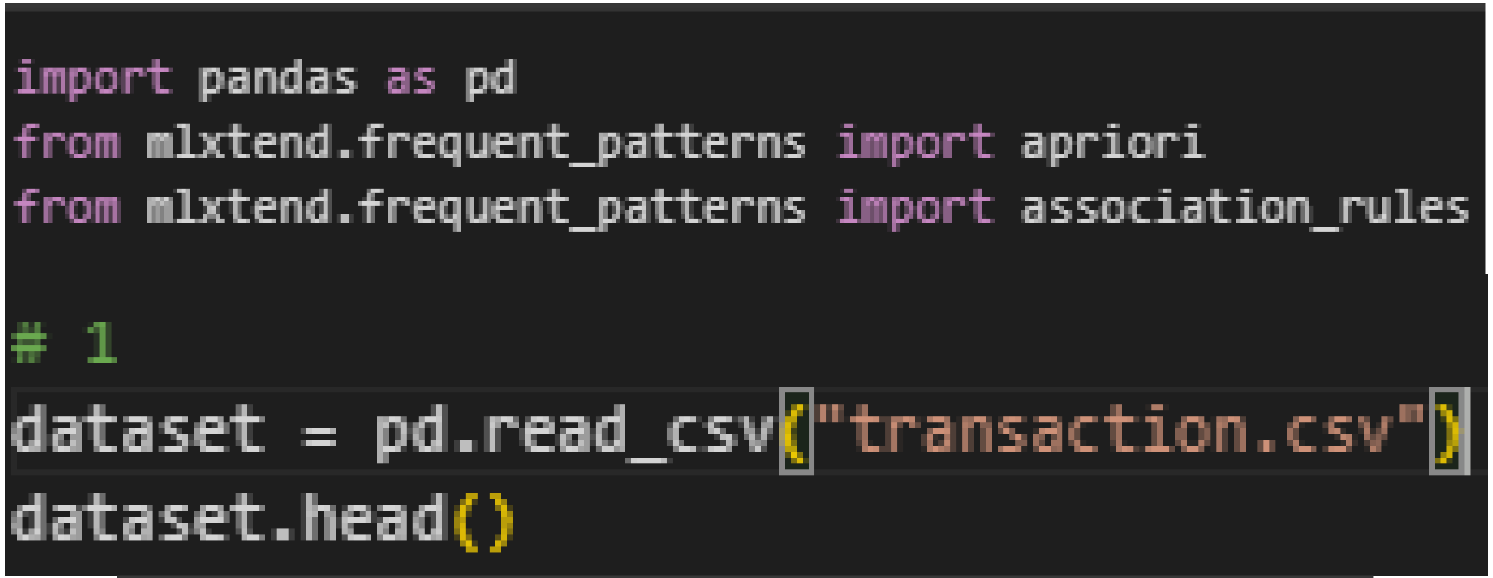
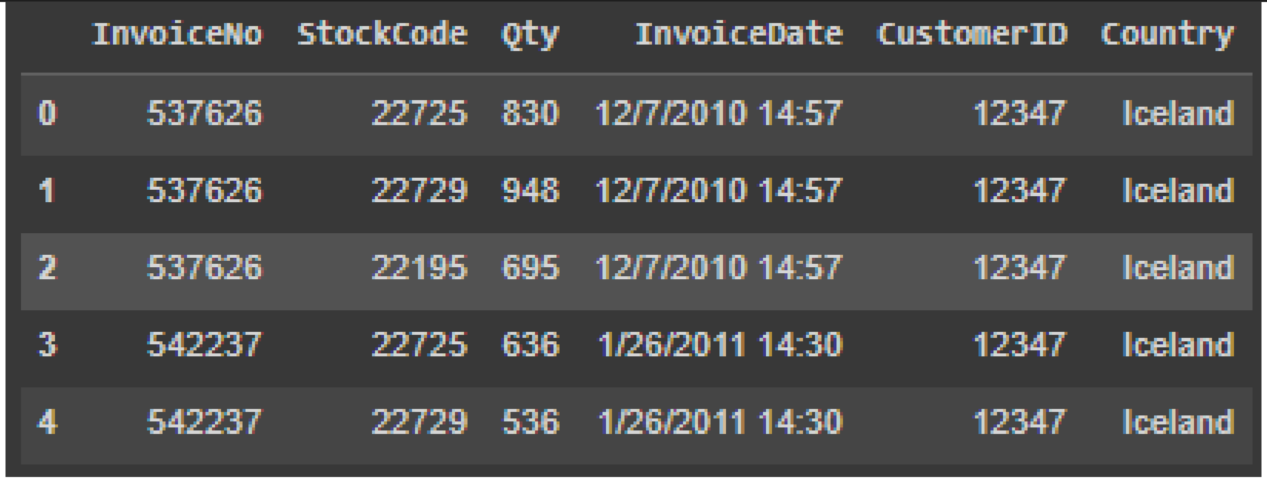
Praktikum Data Mining Minggu Ke-10

Faris Saifullah
3124640034



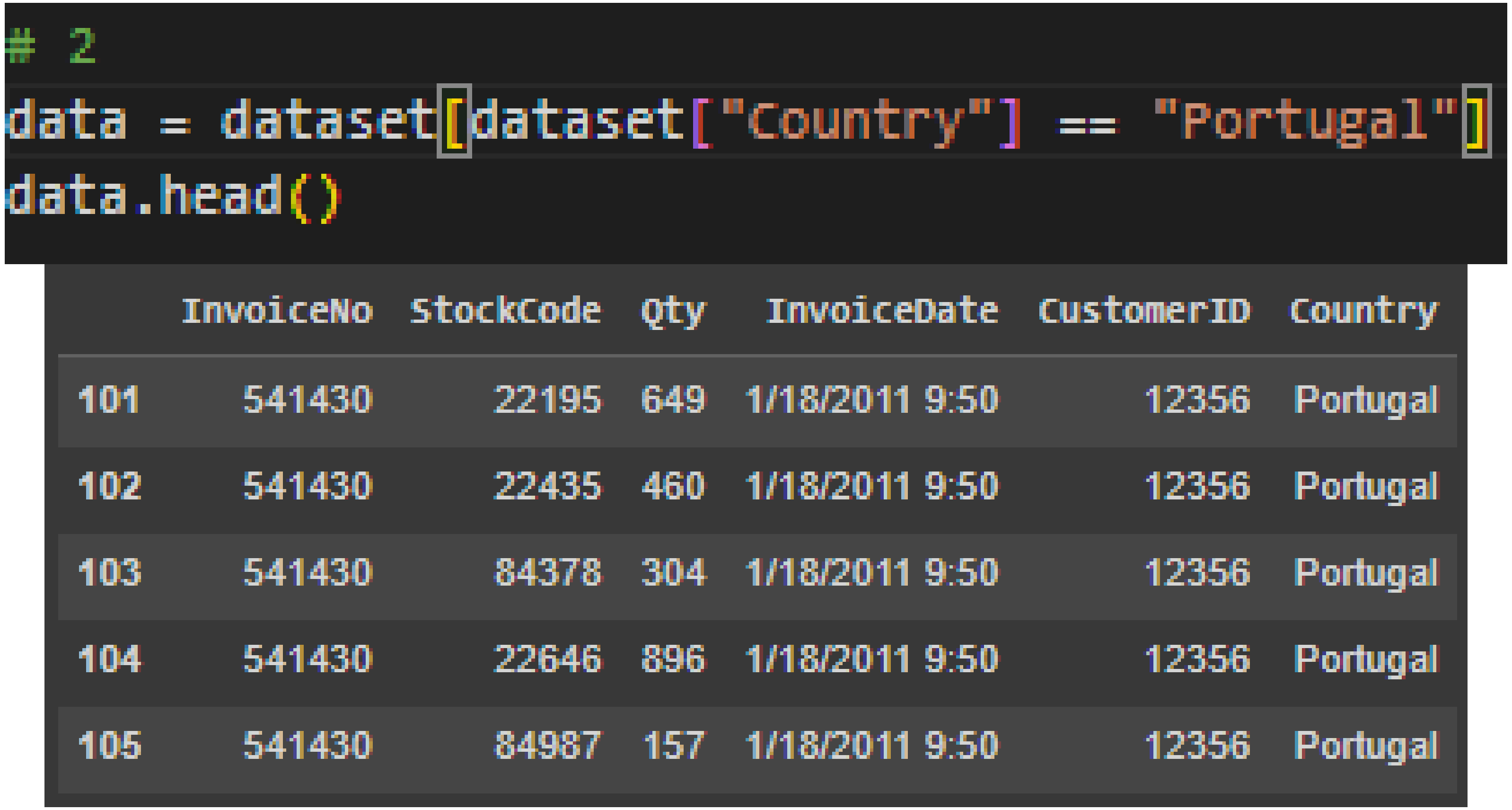
```python
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# 1
dataset = pd.read_csv("transaction.csv")
dataset.head()
```



	InvoiceNo	StockCode	Qty	InvoiceDate	CustomerID	Country
0	537626	22725	830	12/7/2010 14:57	12347	Iceland
1	537626	22729	948	12/7/2010 14:57	12347	Iceland
2	537626	22195	695	12/7/2010 14:57	12347	Iceland
3	542237	22725	636	1/26/2011 14:30	12347	Iceland
4	542237	22729	536	1/26/2011 14:30	12347	Iceland

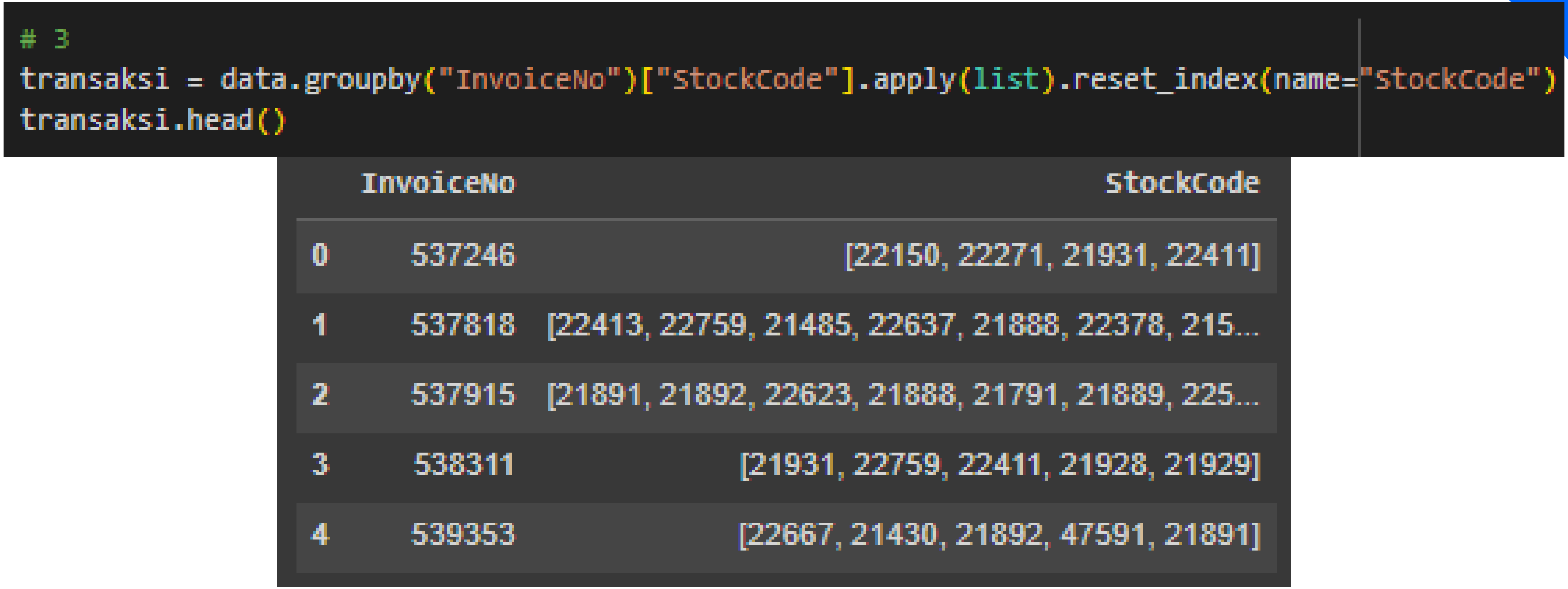
Analisa :
menampilkan data yang ada pada file transaction.csv



```python
# 2
data = dataset[dataset["Country"] == "Portugal"]
data.head()
```

	InvoiceNo	StockCode	Qty	InvoiceDate	CustomerID	Country
101	541430	22195	649	1/18/2011 9:50	12356	Portugal
102	541430	22435	460	1/18/2011 9:50	12356	Portugal
103	541430	84378	304	1/18/2011 9:50	12356	Portugal
104	541430	22646	896	1/18/2011 9:50	12356	Portugal
105	541430	84987	157	1/18/2011 9:50	12356	Portugal

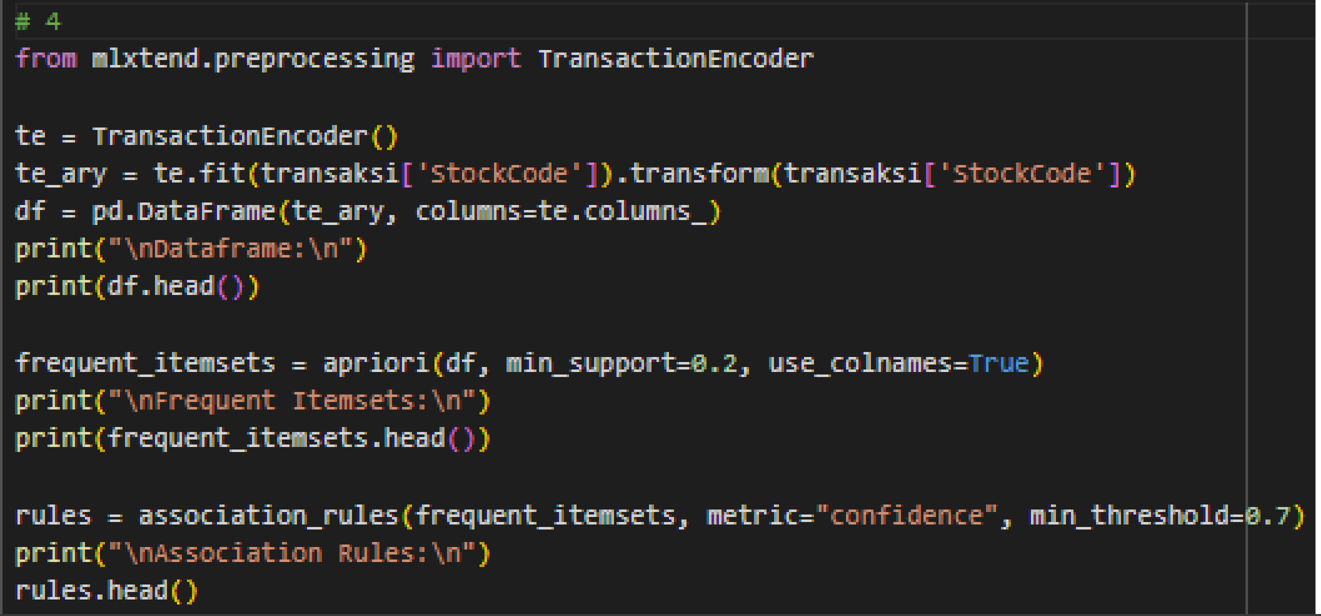
Analisa :
menampilkan dataset negara Portugal pada file
file transaction.csv



```python
# 3
transaksi = data.groupby("InvoiceNo")["StockCode"].apply(list).reset_index(name="StockCode")
transaksi.head()
```

	InvoiceNo	StockCode
0	537246	[22150, 22271, 21931, 22411]
1	537818	[22413, 22759, 21485, 22637, 21888, 22378, 215...
2	537915	[21891, 21892, 22623, 21888, 21791, 21889, 225...
3	538311	[21931, 22759, 22411, 21928, 21929]
4	539353	[22667, 21430, 21892, 47591, 21891]

Analisa :
menampilkan data banyaknya kode StockCode dari data pada setiap transaksi (1 kode InvoiceNo = 1 transaksi)

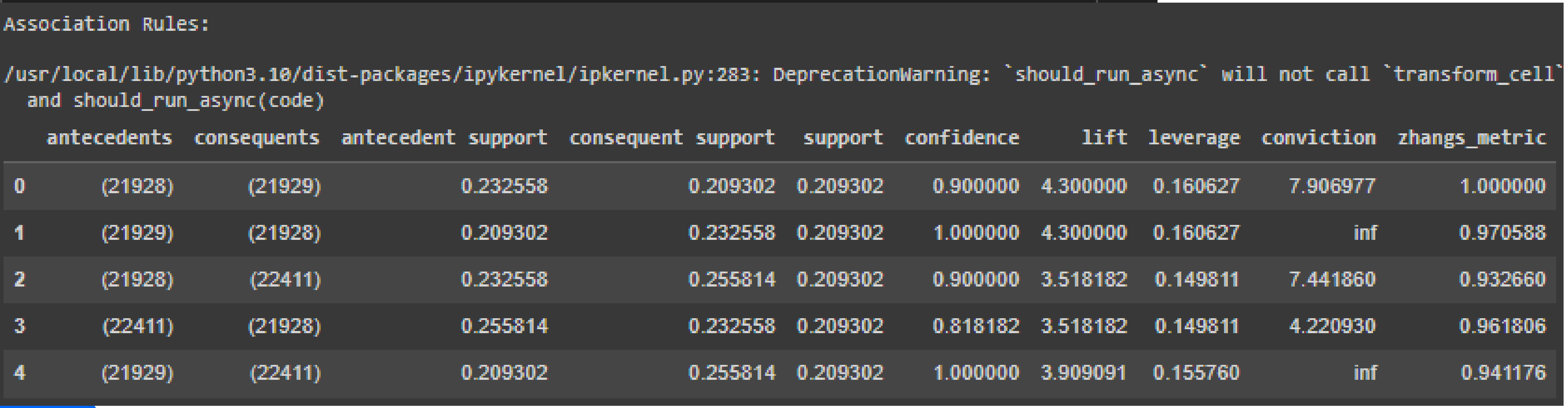


```
# 4
from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()
te_ary = te.fit(transaksi['StockCode']).transform(transaksi['StockCode'])
df = pd.DataFrame(te_ary, columns=te.columns_)
print("\nDataframe:\n")
print(df.head())

frequent_itemsets = apriori(df, min_support=0.2, use_colnames=True)
print("\nFrequent Itemsets:\n")
print(frequent_itemsets.head())

rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
print("\nAssociation Rules:\n")
rules.head()
```



```
Association Rules:

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell`
  and should_run_async(code)
   antecedents  consequents  antecedent support  consequent support   support  confidence      lift  leverage  conviction  zhangs_metric
0      (21928)      (21929)            0.232558            0.209302  0.209302    0.900000  4.300000  0.160627    7.906977       1.000000
1      (21929)      (21928)            0.209302            0.232558  0.209302    1.000000  4.300000  0.160627         inf       0.970588
2      (21928)      (22411)            0.232558            0.255814  0.209302    0.900000  3.518182  0.149811    7.441860       0.932660
3      (22411)      (21928)            0.255814            0.232558  0.209302    0.818182  3.518182  0.149811    4.220930       0.961806
4      (21929)      (22411)            0.209302            0.255814  0.209302    1.000000  3.909091  0.155760         inf       0.941176
```

Analisa : menampilkan association rule pada transaksi dengan minimum support=0.2 dan minimum confidence=0.7