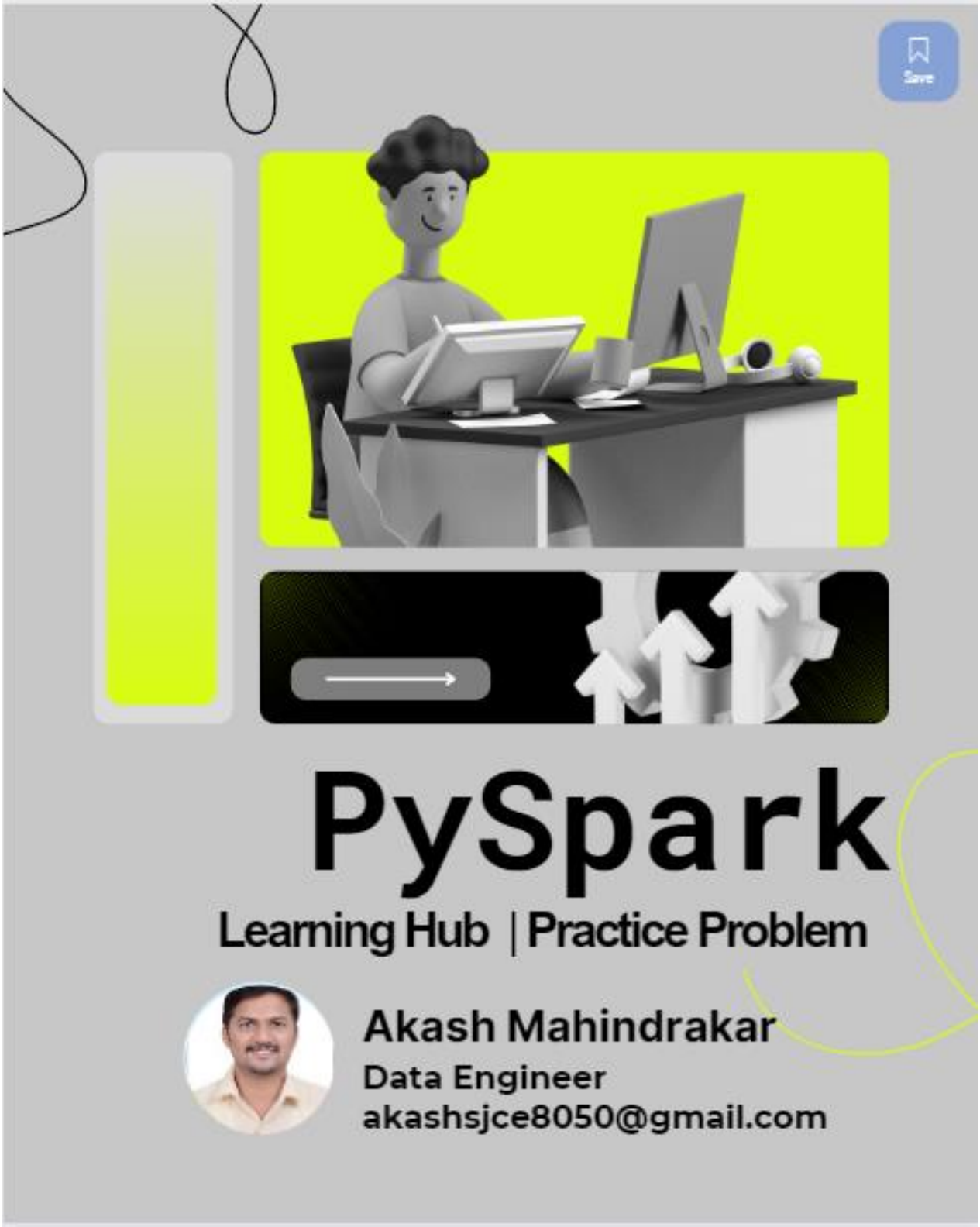
PYSPARK LEARNING HUB



WWW.LINKEDIN.COM/IN/AKASHMAHINDRAKAR

Step - 1 : Problem Statement

Actors and Directors Who Cooperated At Least Three Times

Write a pyspark Program for a report that provides the pairs (actor_id, director_id) where the actor has cooperated with the director at least 3 times.

Difficult Level : EASY

DataFrame:

```
schema = StructType([
        StructField("ActorId",IntegerType(),True),
        StructField("DirectorId",IntegerType(),True),
        StructField("timestamp",IntegerType(),True)
])

data = [
        (1, 1, 0),
        (1, 1, 1),
        (1, 1, 2),
        (1, 2, 3),
        (1, 2, 4),
        (2, 1, 5),
        (2, 1, 6)
]
```

Step - 2 : Identifying The Input Data And Expected

INPUT

INPUT		
ACTOR_ID	DIRECTOR_ID	TIMESTAMP
1	1	0
1	1	1
1	1	2
1	2	3
1	2	4
2	1	5
2	1	6

OUTPUT

OUTPUT	
ACTOR_ID	DIRECTOR_ID
1	1

Step - 3 : Writing the pyspark code to solve

```
# Creating Spark Session

from pyspark.sql import SparkSession
from pyspark.sql.types import StructType,StructField,IntegerType

#creating spark session
spark = SparkSession. \
builder. \
config('spark.shuffle.useOldFetchProtocol', 'true'). \
config('spark.ui.port','0'). \
config("spark.sql.warehouse.dir", "/user/itv008042/warehouse"). \
enableHiveSupport(). \
master('yarn'). \
getOrCreate()


schema = StructType([
        StructField("ActorId",IntegerType(),True),
        StructField("DirectorId",IntegerType(),True),
        StructField("timestamp",IntegerType(),True)
])

data = [
        (1, 1, 0),
        (1, 1, 1),
        (1, 1, 2),
        (1, 2, 3),
        (1, 2, 4),
        (2, 1, 5),
        (2, 1, 6)
]
```

```
df=spark.createDataFrame(data,schema)
df.show()
```

```
+-------+----------+---------+
|ActorId|DirectorId|timestamp|
+-------+----------+---------+
|      1|         1|        0|
|      1|         1|        1|
|      1|         1|        2|
|      1|         2|        3|
|      1|         2|        4|
|      2|         1|        5|
|      2|         1|        6|
+-------+----------+---------+
```

```
df_group=df.groupBy('ActorId','DirectorId').count()
df_group.show()
```

```
+-------+----------+-----+
|ActorId|DirectorId|count|
+-------+----------+-----+
|      1|         2|    2|
|      1|         1|    3|
|      2|         1|    2|
+-------+----------+-----+
```

```
df_group.filter(df_group['count'] >= 3).show()
```

```
+-------+----------+-----+
|ActorId|DirectorId|count|
+-------+----------+-----+
|      1|         1|    3|
+-------+----------+-----+
```

Save

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

Comment

SHARE YOUR THOUGHTS IN COMMENT BELOW

Share