# LearnTrack Big Data Project
Complete Setup, Python Scripts Execution & Visualization Guide

2025–2026

## Contents

## 1 Project Overview

LearnTrack is a polyglot Big Data analytics platform designed to analyze student activities in online learning systems.

The platform provides:

- Real-time engagement scoring
- Multi-region access detection
- Batch analytics at course level
- Interactive dashboards

Technologies used: MongoDB, Redis, Neo4j, Apache Spark, Cassandra, and Grafana.

## 2 Docker Network Creation

All containers communicate through a shared Docker network.

```
docker network create learntrack-net
```

## 3 MongoDB Setup

### 3.1 Run MongoDB Container

```
docker run -d \
  --name mongo \
  --network learntrack-net \
  -p 27017:27017 \
  mongo
```

### 3.2 Insert Sample Data

```
docker exec -it mongo mongosh
```

```
use learntrack

db.activities.insertMany([
  { studentId:"S7", courseId:"C1", action:"page_view", timestamp:new
     Date(), region:"AF" },
  { studentId:"S7", courseId:"C1", action:"page_view", timestamp:new
     Date(), region:"EU" },
  { studentId:"S9", courseId:"C3", action:"quiz_attempt", timestamp:new
      Date(), region:"US" }
])

db.activities.find().pretty()
```

## 4 Redis Setup (Python Script Execution)

### 4.1 Run Redis Container

```
docker run -d \
  --name redis \
  --network learntrack-net \
  -p 6379:6379 \
  redis
```

### 4.2 Execute Redis Python Loader

Redis engagement scores are computed using the Python script `redis_loader.py`, which reads activity data from MongoDB and stores aggregated scores in Redis.

```
docker exec -it spark bash
```

```
python /opt/spark/work-dir/redis_loader.py
```

### 4.3 Verify Redis Scores

```
KEYS student:*:score
GET student:S7:score
MGET student:S7:score student:S9:score
```

## 5 Neo4j Setup (Python Script Execution)

### 5.1 Run Neo4j Container

```
docker run -d \
  --name neo4j \
  --network learntrack-net \
  -p 7474:7474 \
  -p 7687:7687 \
  -e NEO4J_AUTH=neo4j/password \
  neo4j:5
```

### 5.2 Execute Neo4j Python Loader

The graph database is populated using the Python script `neo4j_loader.py`, which loads activities from MongoDB into Neo4j.

```
docker exec -it spark bash
```

```
python /opt/spark/work-dir/neo4j_loader.py
```

### 5.3 Verify Graph in Neo4j Browser

Open:

http://localhost:7474

```
MATCH (s:Student)-[:PERFORMED]->(a:Activity)-[:IN_REGION]->(r:Region)
RETURN s, a, r;
```

### 5.4 Multi-Region Access Detection

```
MATCH (s:Student)-[:PERFORMED]->(a1)-[:IN_REGION]->(r1),
      (s)-[:PERFORMED]->(a2)-[:IN_REGION]->(r2)
WHERE r1.name <> r2.name
  AND duration.between(a1.timestamp, a2.timestamp)
      < duration({minutes:30})
RETURN s.id AS studentId,
       collect(DISTINCT r1.name) + collect(DISTINCT r2.name) AS regions
          ;
```

## 6 Cassandra Setup

### 6.1 Run Cassandra Container

```
docker run -d \
  --name cassandra \
  --network learntrack-net \
  -p 9042:9042 \
  cassandra:4
```

### 6.2 Create Keyspace and Table

```
docker exec -it cassandra cqlsh
```

```
CREATE KEYSPACE learntrack
WITH replication = {'class':'SimpleStrategy','replication_factor':1};

USE learntrack;

CREATE TABLE course_engagement (
  courseid text,
  date date,
  count int,
  PRIMARY KEY (courseid, date)
);
```

## 7 Apache Spark Batch Processing

### 7.1 Run Spark Container

```
docker run -it \
  --name spark \
  --network learntrack-net \
  -p 4040:4040 \
  bitnami/spark:3 \
  bash
```

### 7.2 Execute Spark Job

The Spark job reads activities from MongoDB and writes aggregated engagement results into Cassandra.

```
spark-submit \
  --packages \
  org.mongodb.spark:mongo-spark-connector_2.12:10.3.0,\
  com.datastax.spark:spark-cassandra-connector_2.12:3.5.0 \
  /opt/spark/work-dir/spark_job.py
```

### 7.3 Verify Cassandra Results

```
SELECT * FROM learntrack.course_engagement;
```

## 8 Grafana Visualization

### 8.1 Run Grafana Container

```
docker run -d \
  --name grafana \
  --network learntrack-net \
  -p 3000:3000 \
  grafana/grafana
```

### 8.2 Access Grafana

http://localhost:3000

- Username: admin
- Password: admin

### 8.3 Dashboards

- Redis: student engagement scores
- Cassandra: course-level engagement analytics
- Neo4j: multi-region access detection

## 9 Conclusion

This document presented a complete and consistent setup of the LearnTrack Big Data platform, including Docker orchestration, Python-based data loading, batch analytics with Spark, and real-time visualization using Grafana.