# Introduction

This document provides step-by-step instructions to fully reproduce all the findings reported in the manuscript titled **"An Integrated Framework for Scalable Group Movement Pattern Mining in Massive Trajectory Data"**. Adhering to open science best practices, this guide is designed to ensure complete research reproducibility.

Using the provided code and data, you can regenerate all experimental results from Chapters 5 and 6 of the manuscript, including:

* **Tables:** 7 to 14
* **Figures:** 5 to 8

This guide enables you to regenerate all experimental results by walking you through the entire process: environment setup, repository structure, data processing, modeling, and the final generation of tables and figures.

# System Requirements & Setup

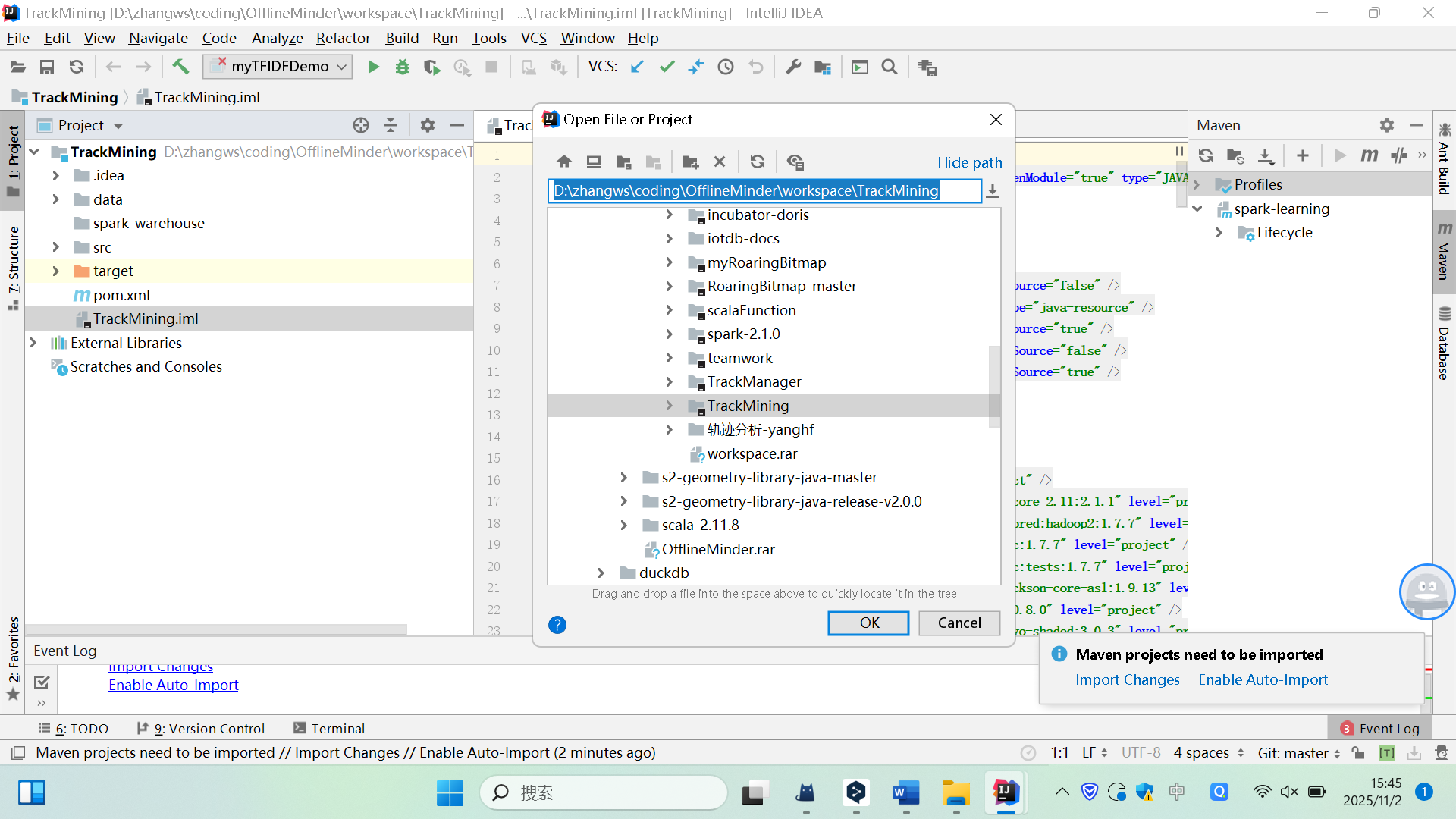
## Software Requirements

* **Operating System:** Tested on Windows 11. Should be compatible with any system supporting the software listed below.
* **SDK:** Scala2.11.8、Java1.8.0+、
* **Build Tool:** Apache Maven 3.3.9+
* **IDE:** IntelliJ IDEA 2019+
* **Database:** MySQL 5.7.11+

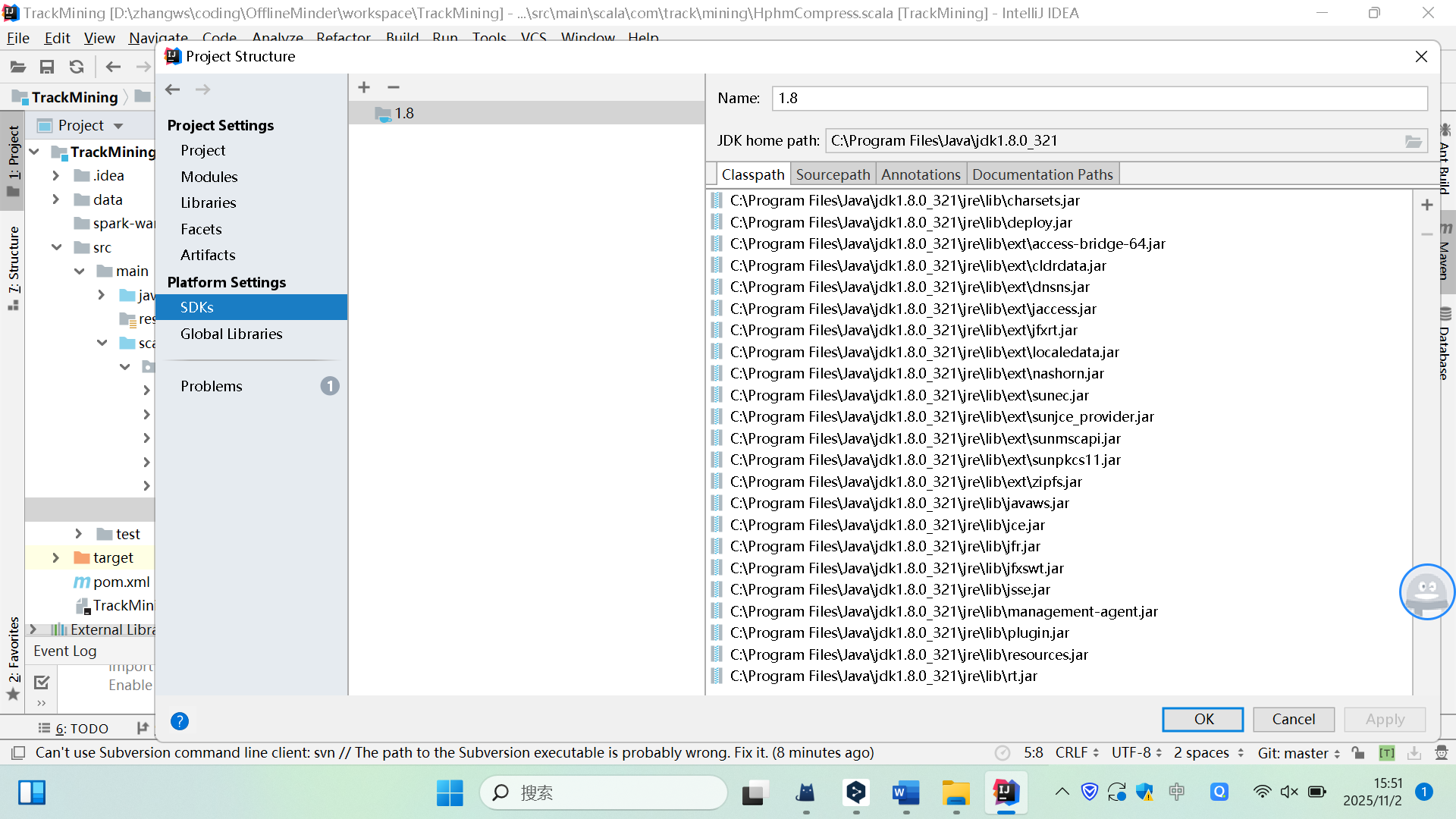
## Environment Configuration

First, ensure the software listed in Section 2.1 is installed, then follow these steps to configure your development environment.

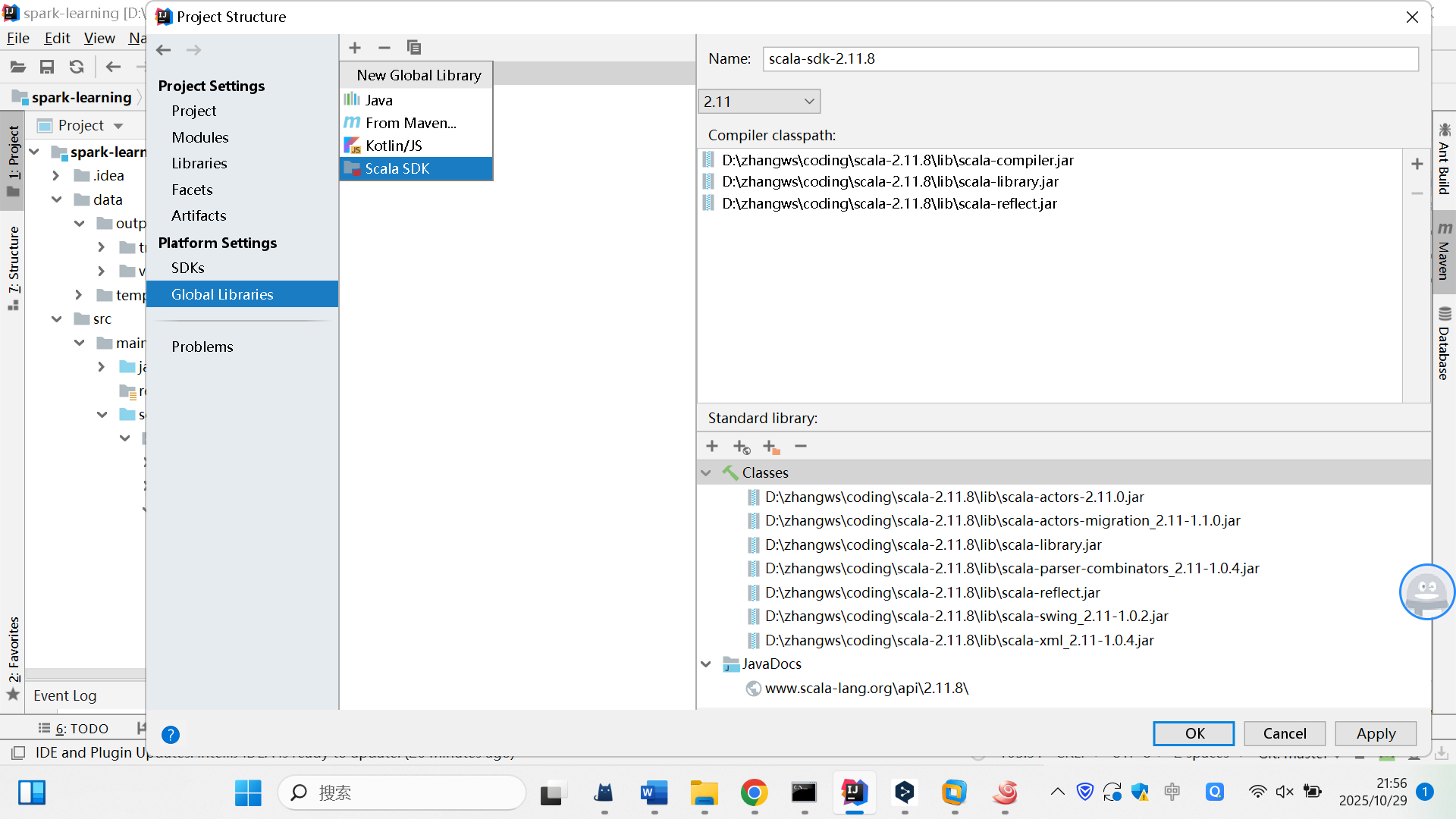
1. **Open Project in IntelliJ IDEA**
   * Launch IntelliJ IDEA.
   * Go to **File → Open → Select project directory**
   * Select the project's root directory (e.g., TrackMining).
   * Choose "Open as Project"



1. **Configure Java SDK:**
   * If SDK is not detected automatically, configure the Java SDK.
   * Navigate to **File → Project Structure →SDKs**
   * Select or add the JDK home path for Java 1.8.



❸ select project directory

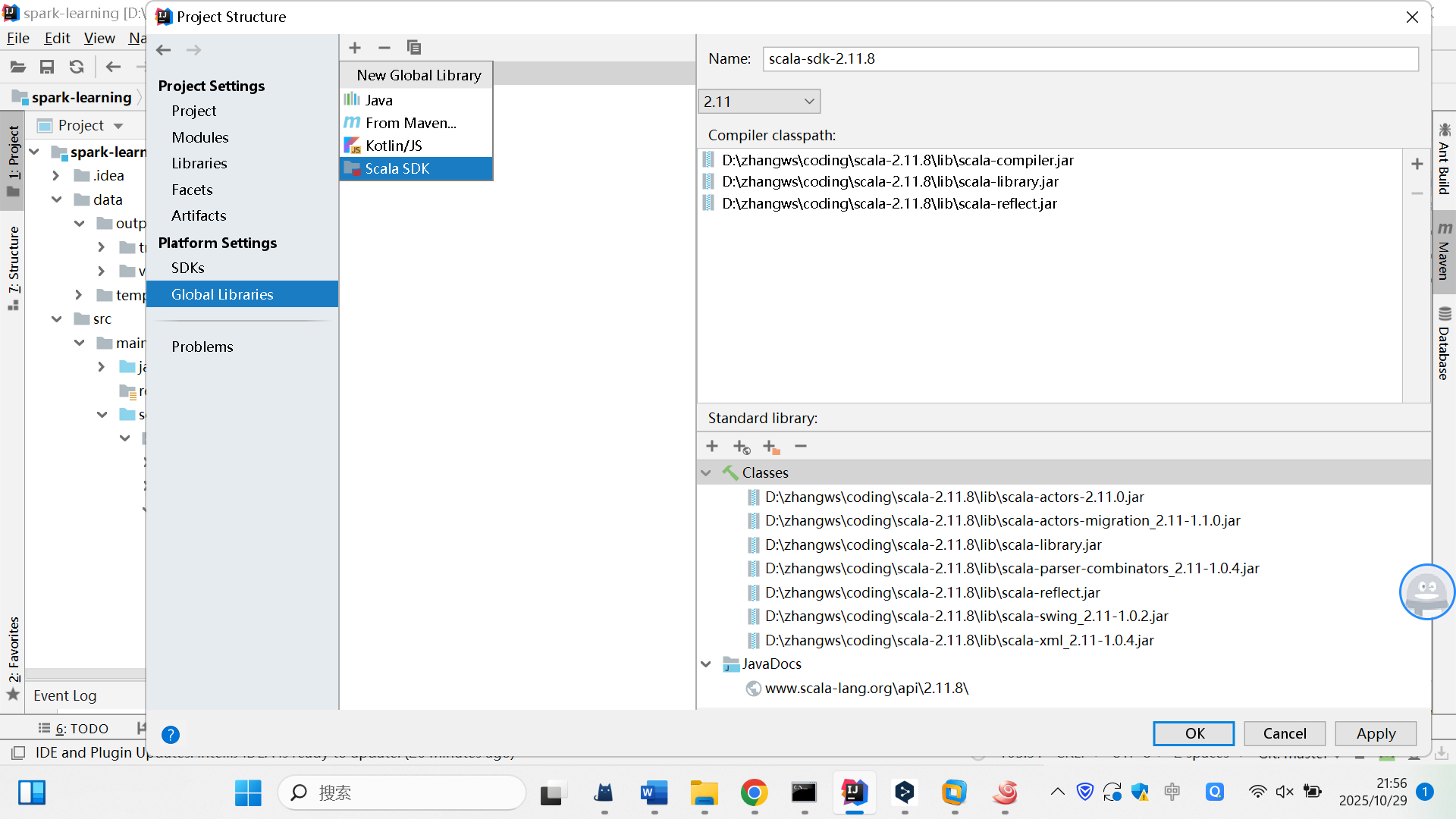
❶

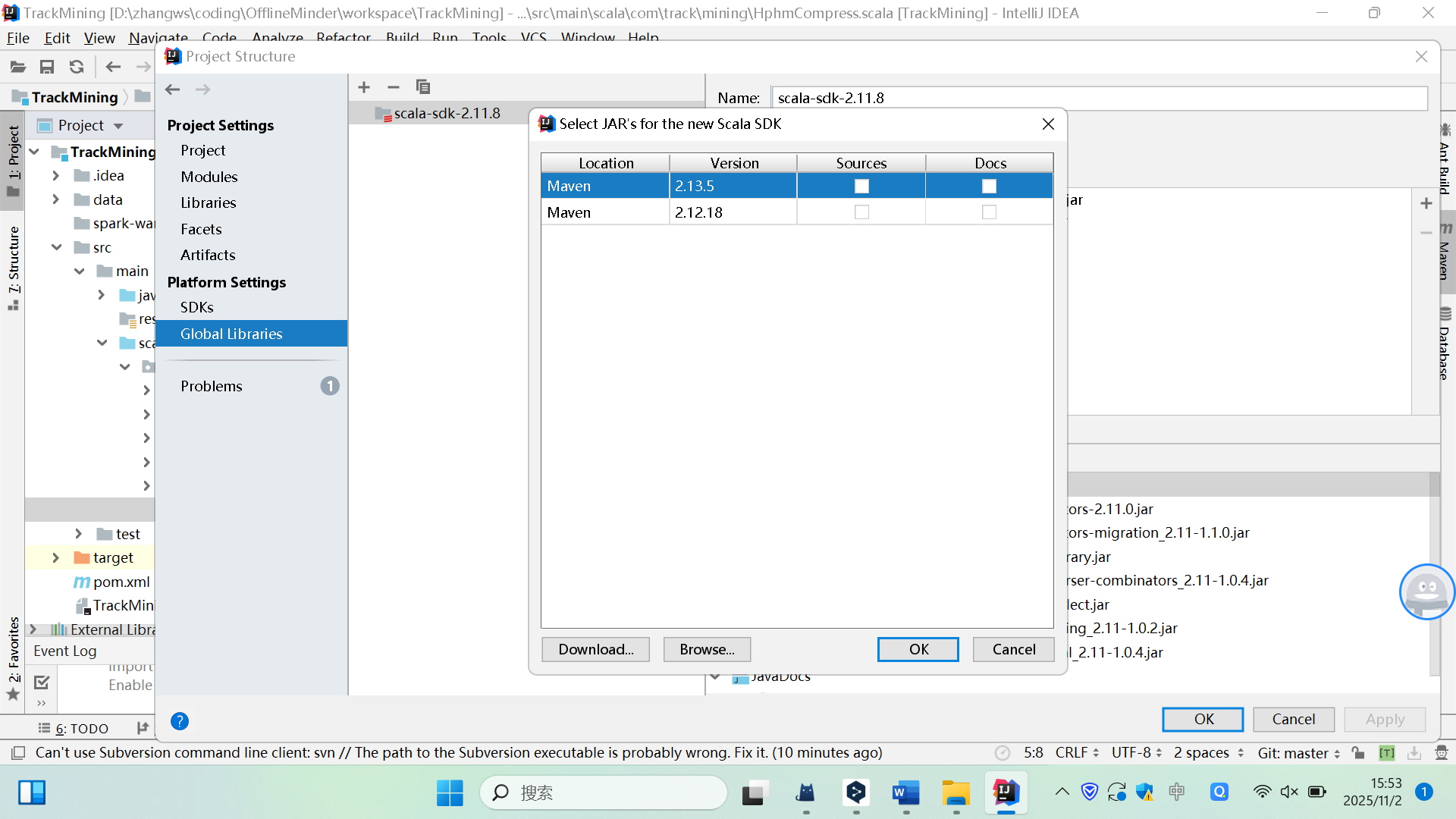
❷

1. **SCALA SDK Configuration**
   * If the Scala SDK id not detected automatically, configure it manually.
   * Navigate to **File → Project Structure... → Global Libraries**
   * Click the **+** button and select **Scala SDK**.
   * Click **“Browse” and** select the installation directory for Scala 2.11.8.
   * Right-click to bring up the context menu, select **"Add to Modules"**.
   * Choose the corresponding module (default: **"TrackMining"**).

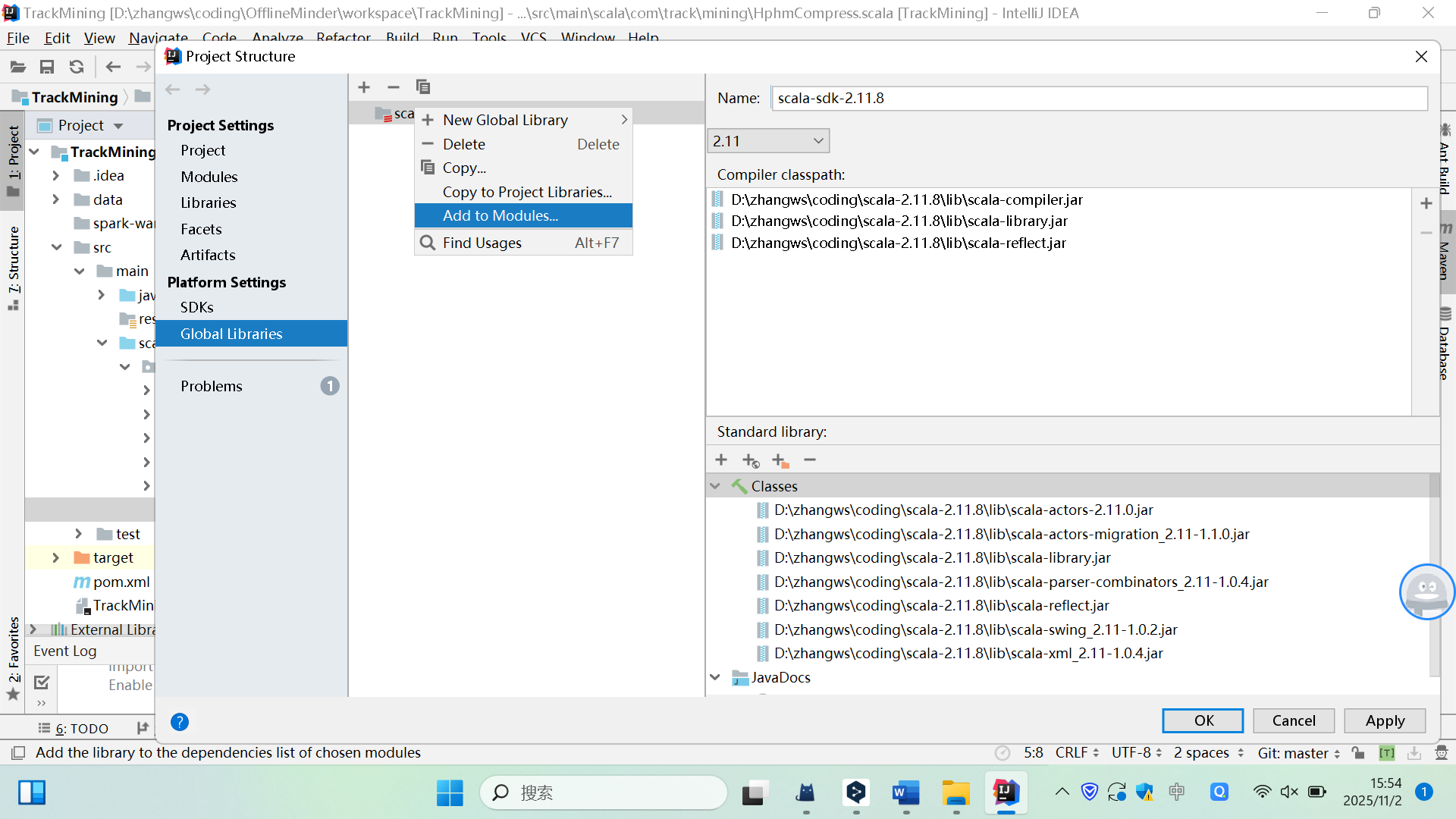


❷ Click +

❶



❸ select directory for Scala 2.11.8.



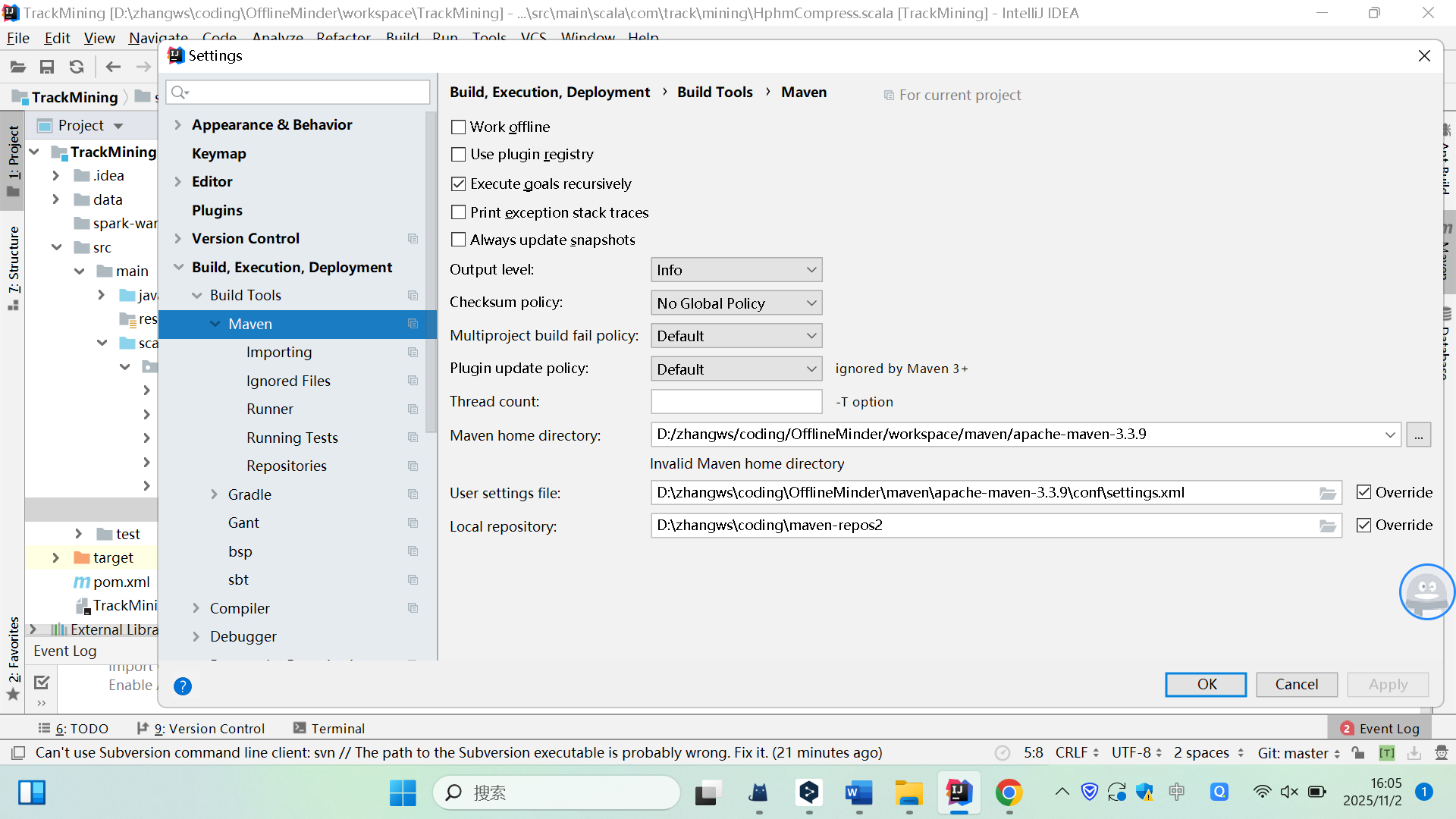
❹ Right-click to bring up the context menu

❺ Select the "Add to Modules" option

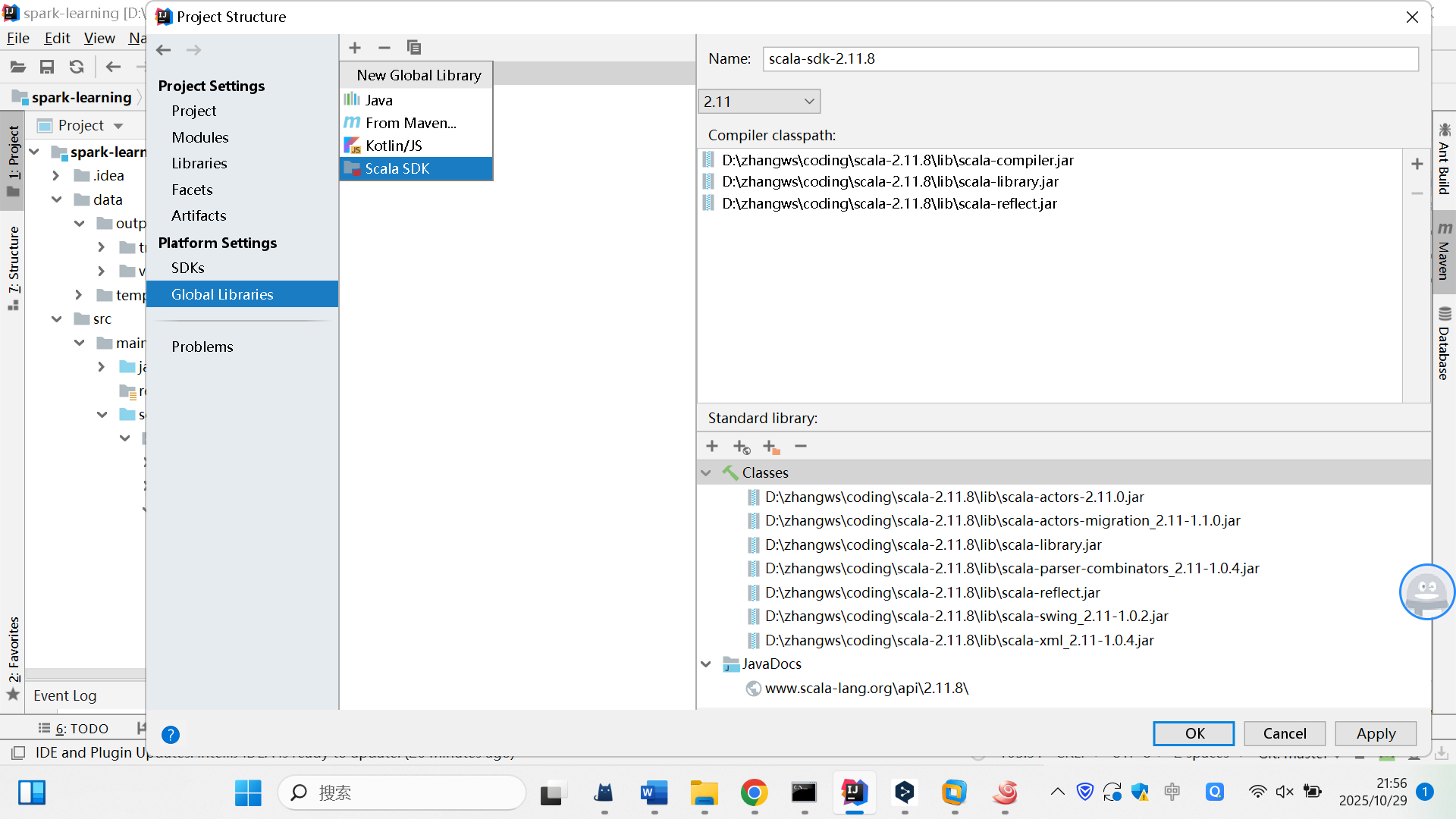


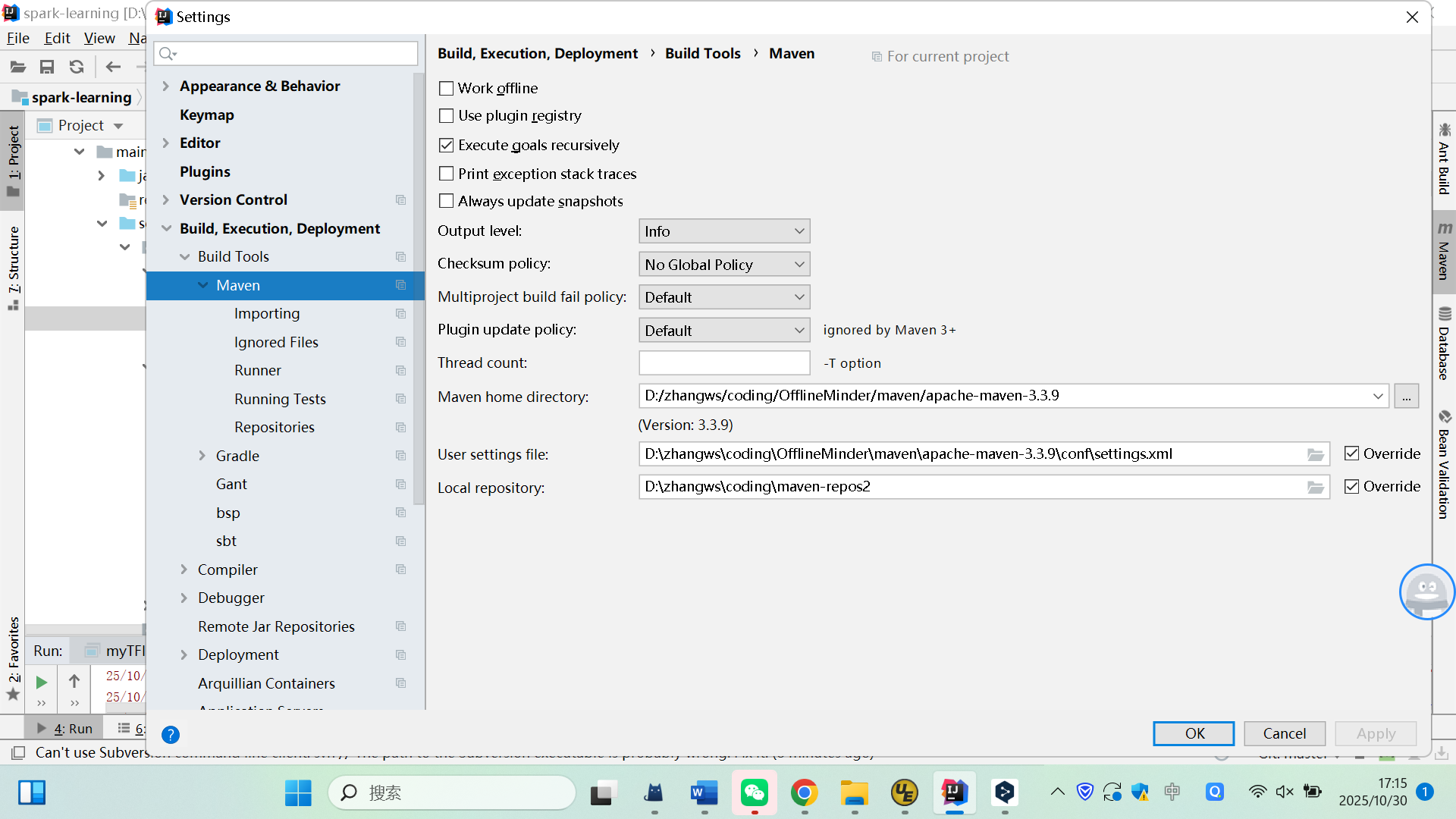
❻ select the corresponding module (the default is "TrackMining")

1. **Configure Maven Local Repository:**
   * Go to **File → Settings (or Preferences on macOS) → Build, Execution, Deployment → Build Tools → Maven**.
   * Set the **"Maven home directory"** path.
   * Set the **"User settings file"** to ${MAVEN\_HOME}/conf/settings.xml.
   * Set the **"Local repository "** to your local Maven repository.
   * In the global ***settings.xml*** file, ensure the <localRepository> tag points to the correct path of your local Maven repository (e.g., maven-repos2).



❷ select maven home directory

❶

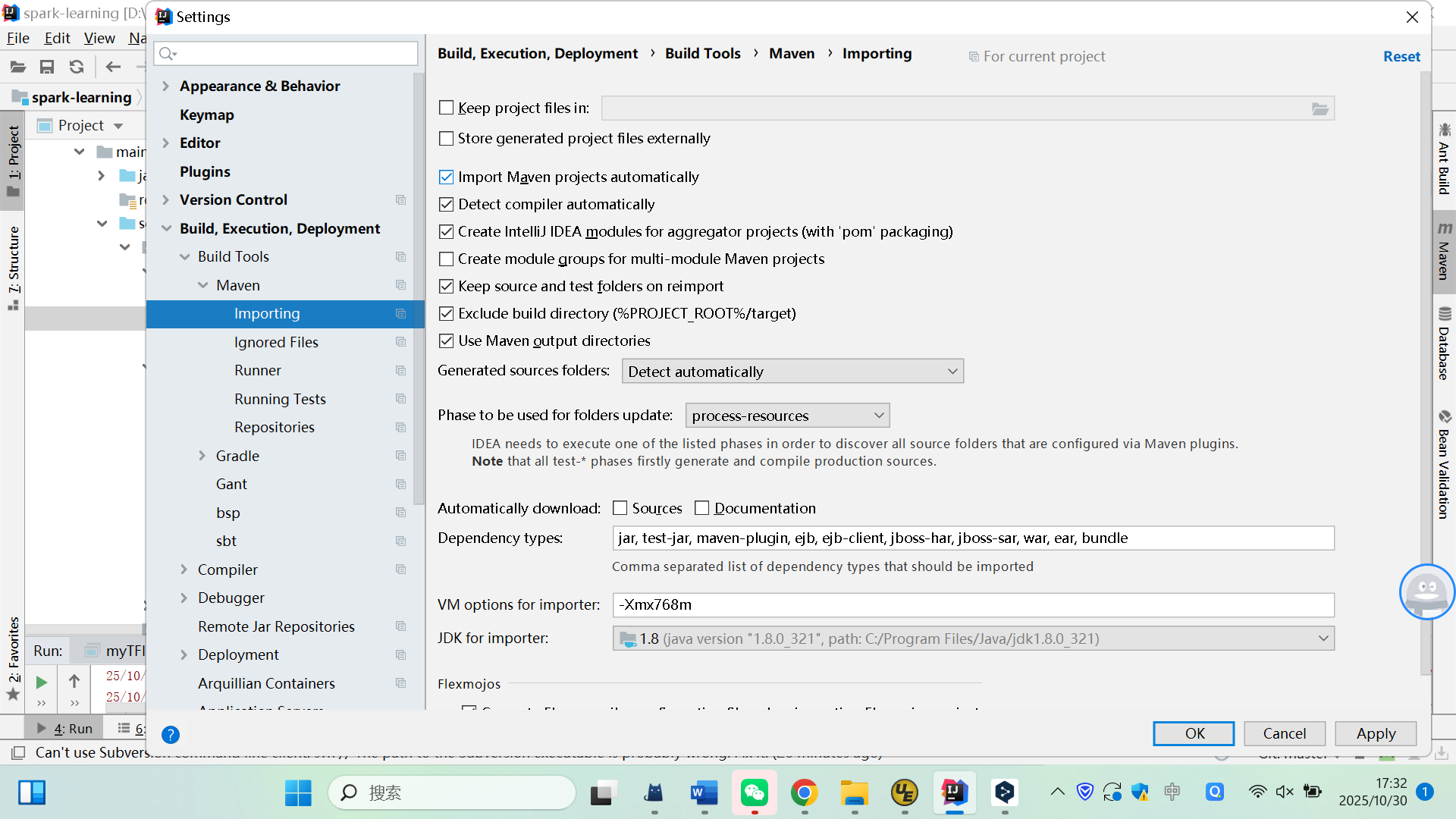


❸ select maven’s settings.xml file, ${MAVEN\_HOME}/conf/settings.xml

❹select your local repository(maven-repos2) directory



Ensure <localRepository> tag points to the correct path of your local Maven repository (maven-repos2).



❻ config the JDK

❺select auto import

# Repository Structure

All file paths in the code are **relative paths**. Please maintain the following repository structure after downloading the project.

TrackMining/

│

├── data/

│ ├── input/

│ │ ├── tb\_s\_event\_wgs\_2025-11-03.zip # Input data file

│ ├── output/ # Output results corresponding to paper chapters

│ │ ├── Chapter5.2

│ │ │ ├── figures/

│ │ │ ├── tables/

│ │ ├── Chapter5.3

│ │ │ ├── figures/

│ │ │ ├── tables/

│ │ ├── Chapter5.4

│ │ │ ├── figures/

│ │ │ ├── tables/

│ │ └── Chapter6

│ │ │ ├── figures/

│ │ │ ├── tables/

├── src/

│ ├──main

│ │ ├──scala

│ │ │ ├── com.track.mining

│ │ │ │ ├── algorithm/ # Experiment algorithms

│ │ │ │ ├── common/ # Common utility methods

│ │ │ │ ├── database/ # MySQL database-related methods

│ │ │ │ ├── trackmanager/ # Core method entry point

│ │ │ │ │ ├── DataWarehouse.scala # Save result to text file

│ │ │ │ │ ├── TrackDataCompress.scala # Code for Chapter 5.2

│ │ │ │ │ ├── TrackDataLSH.scala # Code for Chapter 5.4 & Chapter 6

│ │ │ │ │ ├── TrackDataProcess.scala # Simulated data processing

│ │ │ │ │ ├── TrackDataSimilar.scala # Code for Chapter 5.3

│ │ │ │ │ ├── TrackPatternsDetection.scala # Code for Chapter 6

│ │ │ │ │ ├── TrackView.scala # Core method entry point

│

└── pom.xml # Project dependencies configuration

# Data Description and Import

This section describes how to import the source data into a MySQL database.

* **Source Data File:** data/import/*tb\_s\_event\_wgs\_2025-11-03.z\**.
* **Steps:**

1. **Create the MySQL Database:**
2. **Create the MySQL Database:**

* In your system terminal, log into MySQL:

mysql -u [username] -p

(Example: mysql -u woolsam -p )

* Execute the following commands in the MySQL terminal:

CREATE DATABASE IF NOT EXISTS TrackManager DEFAULT CHARSET utf8 COLLATE utf8\_bin;

* (Optional Grant Command - Review for security/privacy before sharing)

grant all on \*.\* to woolsam identified by "password@123";

1. **Import data:**

Execute the following command in your system terminal (replace ***${PROJECT\_HOME}*** with the actual path to your project):

--创建数据库,执行以下语句

mysql -u [usernum] (eg: woolsam) -p TrackMananger < **${PROJECT\_HOME}**/data/input/ tb\_s\_event\_wgs\_2025-11-03.sql

1. **Verify the Import:**

Run the following queries in the MySQL terminal to confirm successful data import:

-- Check sample data

SELECT \* FROM TrackManager.tb\_s\_event\_wgs LIMIT 10;

-- Confirm the number of devices in the Simulated Dataset

SELECT COUNT(DISTINCT device\_id) FROM TrackManager.tb\_s\_event\_wgs WHERE device\_id LIKE 'test%';

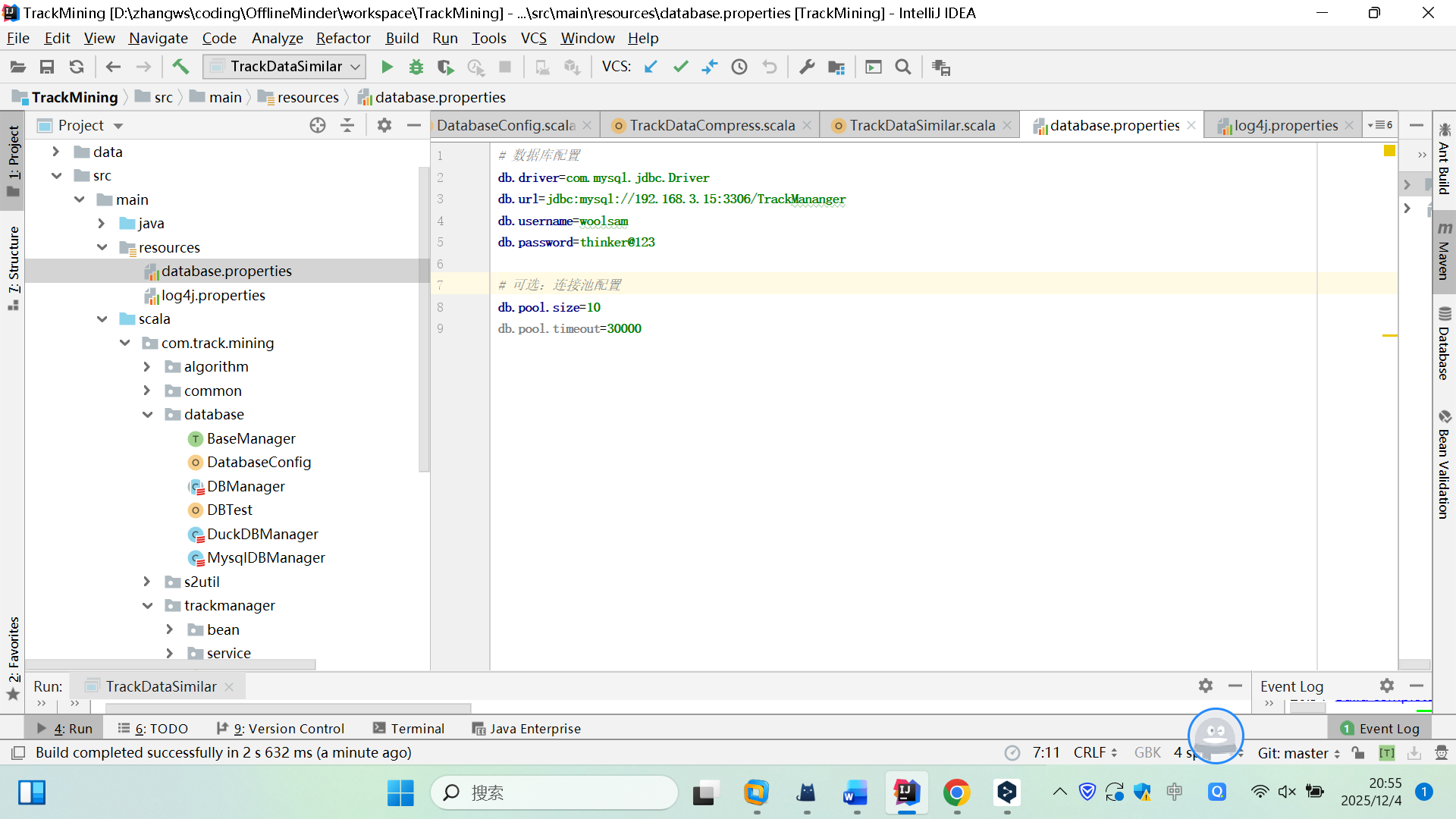
-- Confirm the number of devices in the Elderly Volunteer Dataset

SELECT COUNT(DISTINCT device\_id) FROM TrackManager.tb\_s\_event\_wgs WHERE device\_id REGEXP '[0-9]+$';

# Instructions for analysis

## Basic Configuration

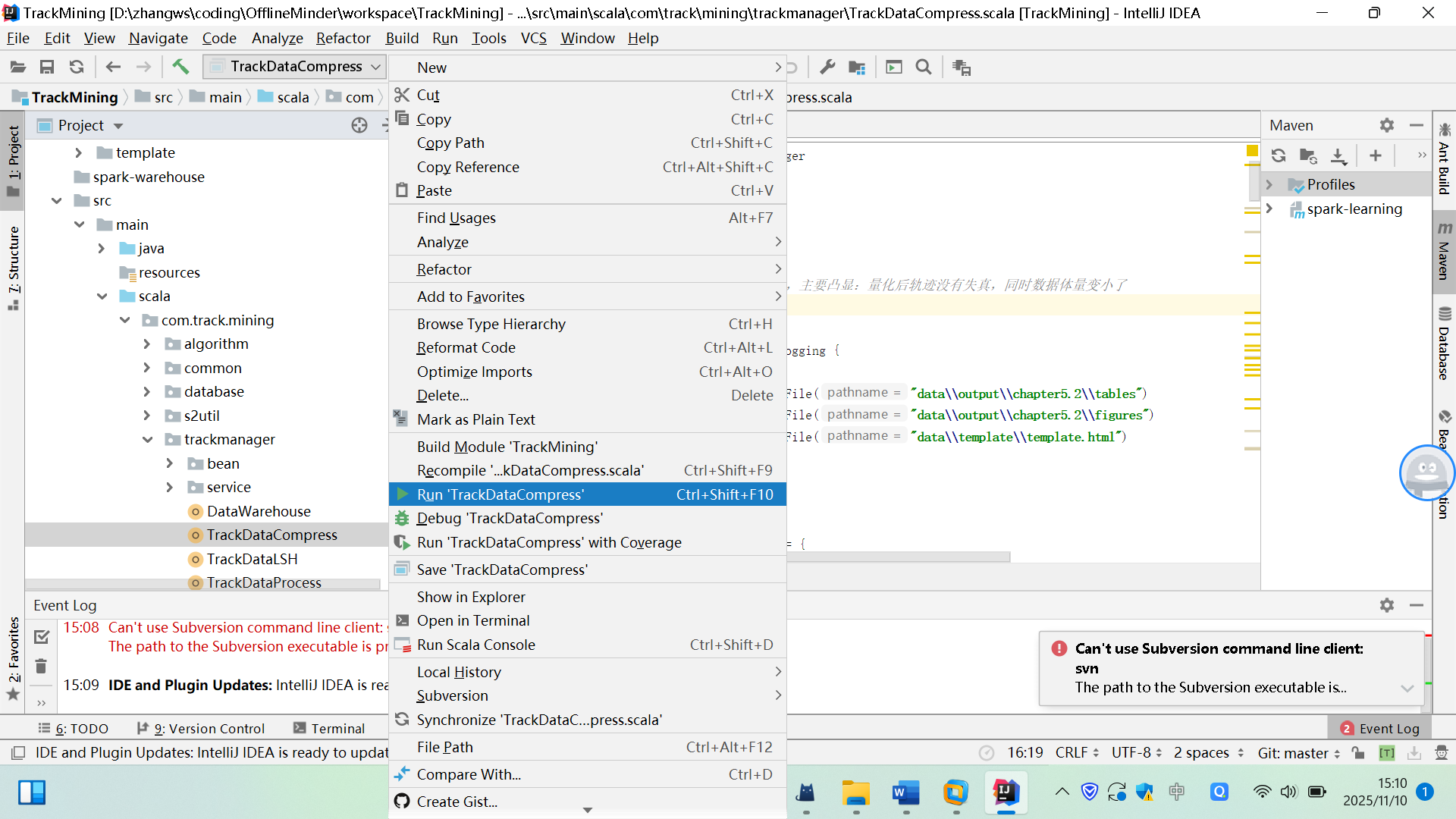
* Launch IntelliJ IDEA and open the ***TrackMining*** project.
* **Modify JDBC Configuration:** Update the database connection settings (URL, username, password) in the ***database.properties*** file within the ***src/main/resources*** directory to match your local MySQL setup.



 ❶ select the “database.properties”

## Generate Chapter5.2 Analysis (Tables and Figures)

* **Code to Run:** com.track.mining.trackmanager.TrackDataCompress.scala
* **Description:** This code generates the results for Tables 7 and Figure 5 in Chapter 5.2.
* **How to Run:**
  + Right-click the *TrackDataCompress.scala* file in the project explorer and click "Run ‘TrackDataCompress‘ " menu.



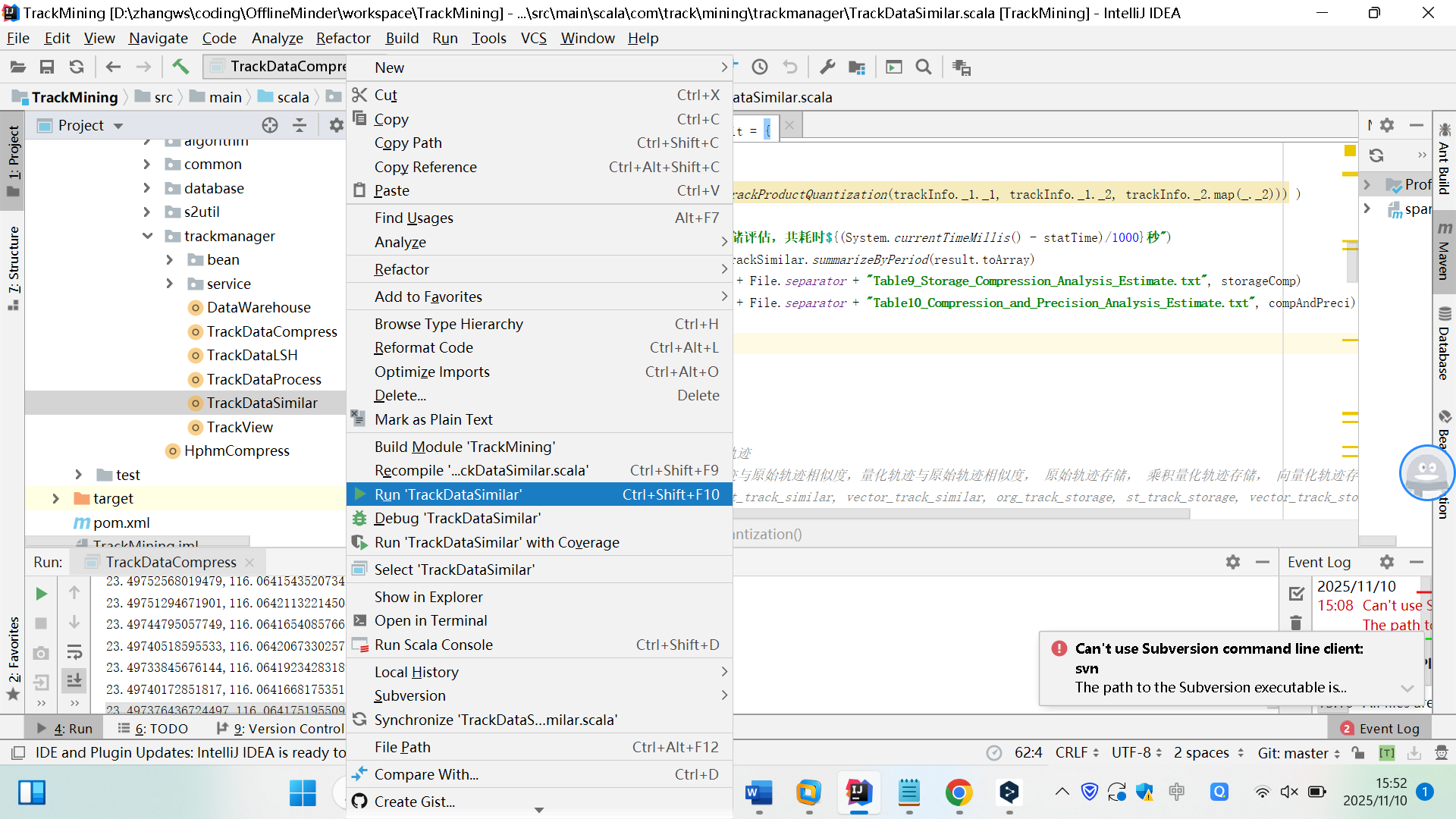
 ❶ select the “TrackDataCompress.scala” class

 ❷ click Run ‘TrackDataCompress’

* **Expected Output:**
  + output/chapter5.2/figures/Driving/\*\_orig.html
  + output/chapter5.2/figures/Driving/\*\_quantify.html
  + output/chapter5.2/figures/Subway/\*\_orig.html
  + output/chapter5.2/figures/Subway/\*\_quantify.html
  + output/chapter5.2/figures/Walking/\*\_orig.html
  + output/chapter5.2/figures/Walking/\*\_quantify.html
  + output/chapter5.2/tables/Table7\_\*
  + ...

## Generate Chapter5.3 Analysis (Tables and Figures)

* **Code to Run:** com.track.mining.trackmanager.TrackDataSimilar.scala
* **Description:** This code generates the results for Tables 9 and 10 in Chapter 5.3.
* **How to Run:**
  + Right-click the *TrackDataSimilar.scala* file in the project explorer and click "Run ‘TrackDataSimilar‘ " menu.



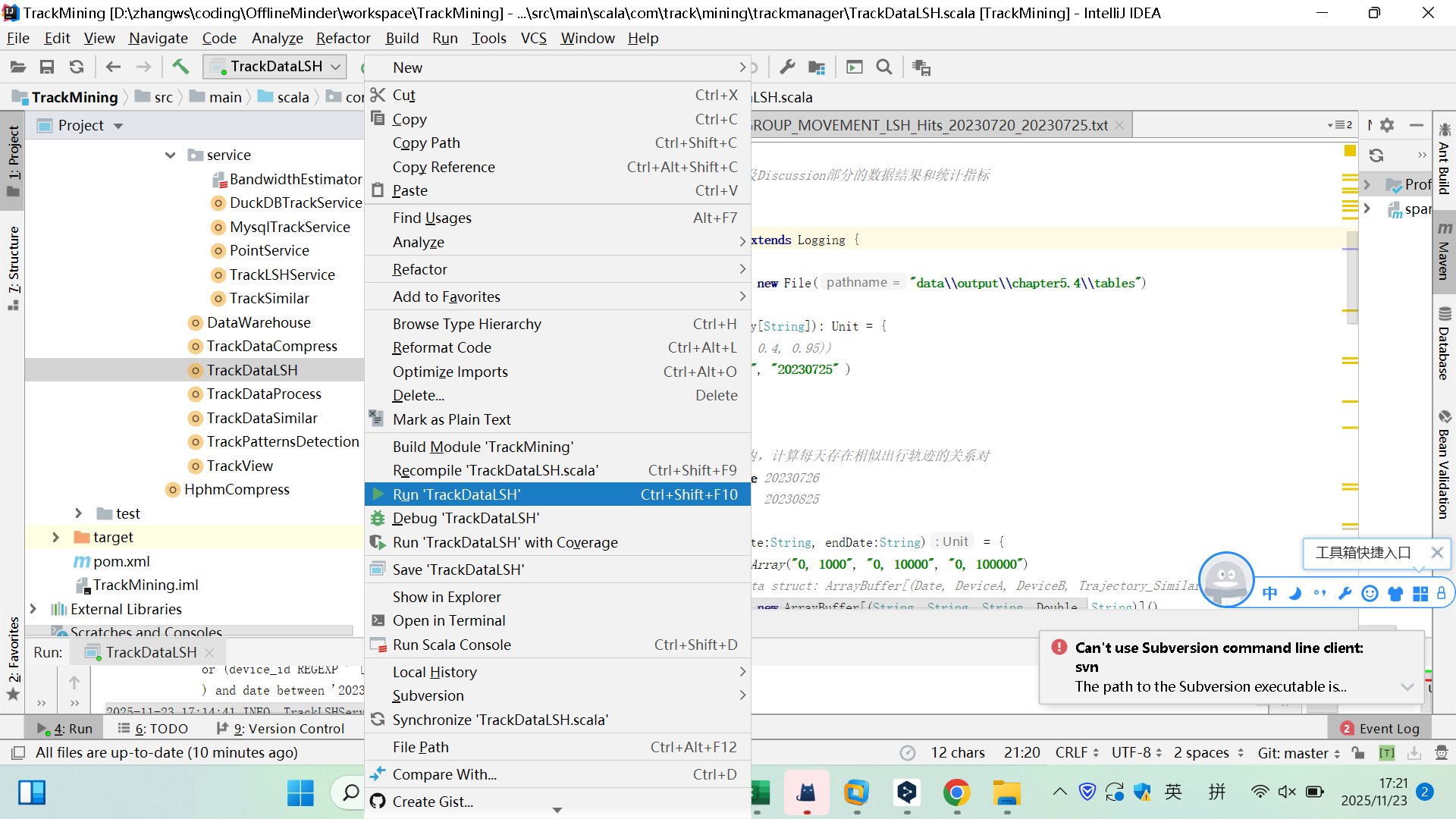
 ❷ click Run ‘TrackDataSimilar

 ❶ select the “TrackDataSimilar.scala” class

* **Expected Output:**
  + output/chapter5.3/tables/Table9\_\*.txt
  + output/chapter5.3/tables/Table10\_\*.txt
  + ...

## Generate Chapter5.4 Analysis (Tables and Figures)

* **Code to Run:** com.track.mining.trackmanager.TrackDataLSH.scala
* **Description:** This code generates the results for Tables 11 and 12 in Chapter 5.4.
* **How to Run:**
  + Right-click the *TrackDataLSH.scala* file in the project explorer and click "Run ‘TrackDataLSH‘ " menu.



❷ click Run ‘TrackDataLSH‘

❶ select the “TrackDataLSH.scala” class

* **Warning: Running this experiment requires significant time and computer memory. If you find the execution time excessive or encounter failures, please adjust your code configuration.**
  + Modify the configuration of the `filterTags` parameter in the `TrackDataLSH.scala` class, Adjusted the third threshold from 50,000 to 20,000:

***Original configuration****: Array(“0, 1000”, “0, 10000”, “0, 50000”)*

***New configuration****: Array(“0, 1000”, “0, 10000”, “0, 20000”)*

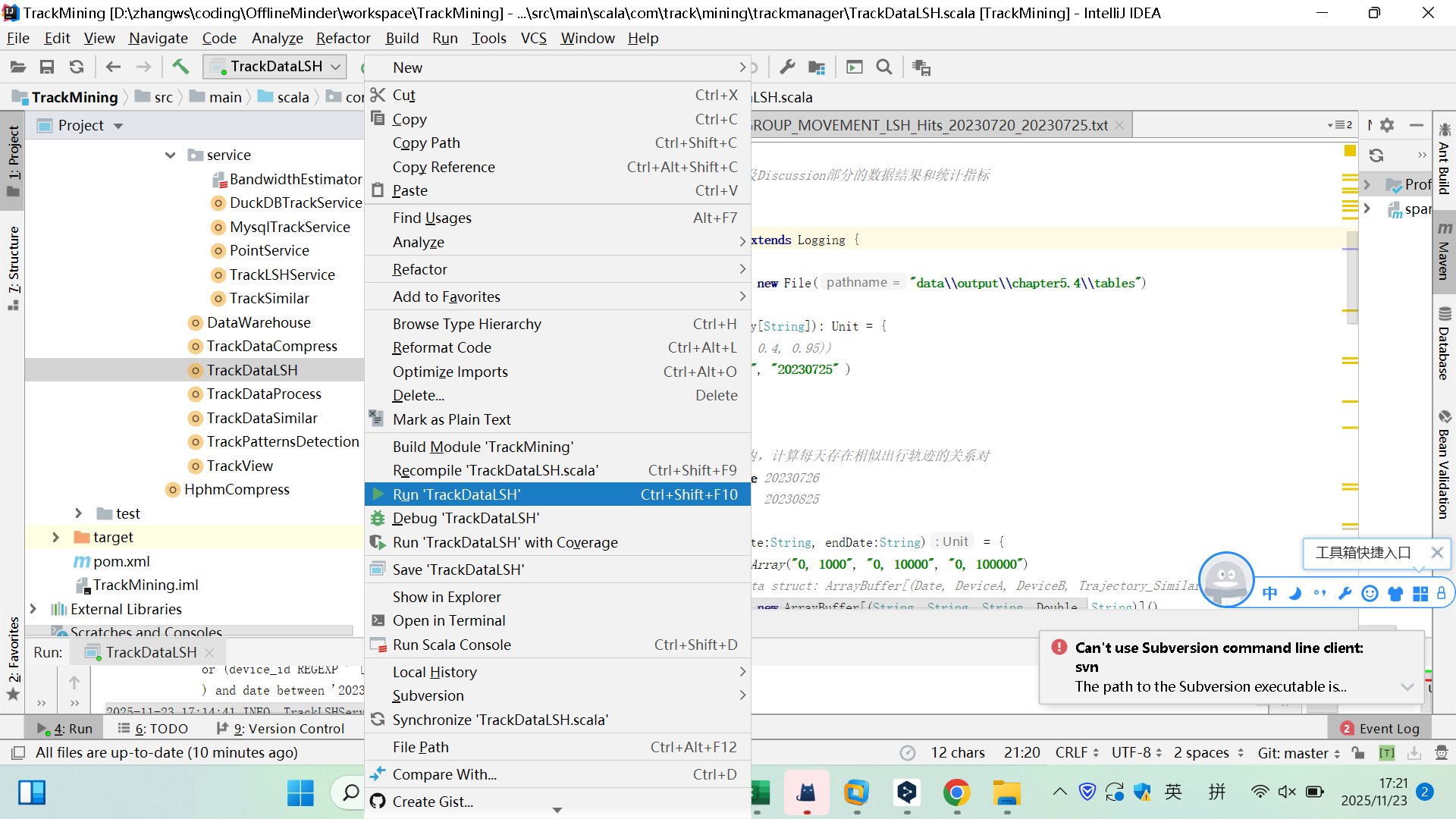


**Modify code configuration**

* **Expected Output:**
  + output/chapter5.4/tables/Table11\_\*.txt
  + output/chapter5.4/tables/Table12\_\*.txt
  + ...

## Generate Chapter6 Analysis (Tables and Figures)

* **Code to Run:** com.track.mining.trackmanager.TrackPatternsDetection.scala
* **Description:** This code generates the results for Tables 13 and 14, Figure 7 and 8 in Chapter 6.
* **How to Run Progress:**
  + Right-click the *TrackDataLSH.scala* file in the project explorer and click "Run ‘TrackDataLSH‘ " menu.



❷ click Run ‘TrackDataLSH‘

❶ select the “TrackDataLSH.scala” class

* **Expected Output:**
  + output/chapter6/tables/Table13\_\*.txt
  + output/chapter6/tables/Table14\_\*.txt
  + output/chapter6/figures/Figures7\_\*.html
  + output/chapter6/figures/Figures8\_\*.html
  + ...

# Troubleshooting

* **"Access denied for user 'XXX'@'IP'" error:** Ensure the configuration for **db.url, db.username, and db.password** in the ***database.properties*** file matches your local MySQL credentials.