

## USER MANUAL OF SOUND TREE

1. Add soundTree jar to your Project
2. Add Neo4j jars and Weka jars to your Project (which are provided you in Jars folder)
3. Import recommendation.main package to your project's java file which uses soundTree methods

Example : **import** recommendation.main.\*;

4. Folder of dataset which is provided by ARGEDOR must include song.txt, map.txt, performer.txt, album.txt and user action log files for creating database. The folder should not include irrelevant files.
5. There are example mains for both creating graph database (Figure 1) and getting recommendations from existing database (Figure 2)
  - For using methods of soundTree, firstly create Recommender class object. Then, methods can be called via this object like in examples.

```
public static void main(String argv[]) throws Exception
{
    String argedor= "C:\\Users\\isinsukatircioglu\\workspace\\SoundTreeProject\\data";
    String workS= "C:\\Users\\isinsukatircioglu\\workspace\\SoundTreeProject\\db";
    Recommender r = new Recommender();

    r.train(argedor, workS);
}
```

Figure 1: Create Graph Database

```
public static void main(String argv[]) throws Exception
{
    Long user = 2073756L;
    String workS= "C:\\Users\\isinsukatircioglu\\workspace\\SoundTreeProject\\db";
    List<Long> results = new ArrayList<Long>();
    Recommender r = new Recommender();
    r.loadWorkspace(workS);
    results.addAll(r.recommend(user)); // results include ids of recommendations
    if(results.size()==0)
    {
        System.out.println("Empty results");
    }
    for(int i=0;i<results.size();i++)
    {
        System.out.println(results.get(i));
    }
}
```

Figure 2: Recommend Songs

6. recommendation.main package of Sound Tree jar includes following methods, methods can be used according to below descriptions:

**public void train(String argedorDataFolderPath, String workspacePath);**

Firstly, train method must be called in order to create neo4j database from Argedor dataset which is in the “argedorDataFolderPath” (first argument of method) path. Database and related files will be created in “workspacePath” (second argument of method) path. This method must be called at least once.

**public void loadWorkspace(String workspacePath);**

If database is already created, loadWorkspace method can be used to determine path of database (workspacePath). Method takes one argument workspacePath which is path of database as a String type. If database is created before, usage of this method is mandatory. If train method is called and database is created at the beginning, this method may not be called.

**public List<Long> recommend(Long userId );**

After creation of Database by train method or determined database path by loadWorkspace, recommend method can be called to get recommendations for the user. This method takes one argument which is userID (as Long type) of intended user. This method returns a list which consists of songIDs of recommendations as long data type.

**Warning:** Creating Graph Database may take a long time according to log file size. Please do not stop generation of database process.