LAB - 8

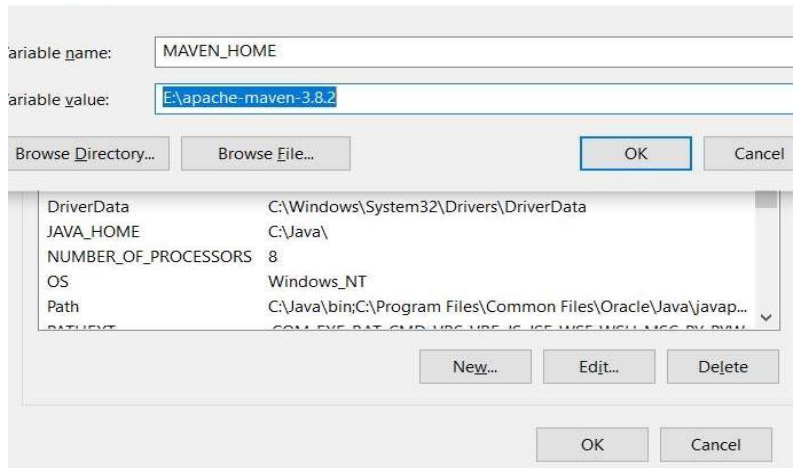
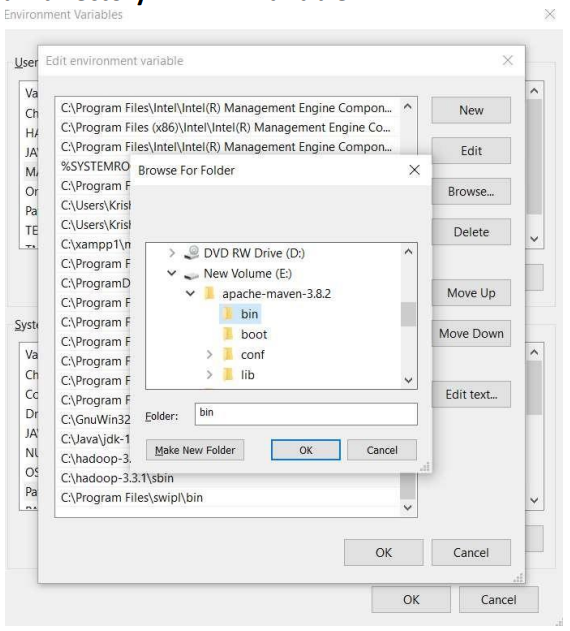
|  |  |
| --- | --- |
| Name | Keval D Gandevia |
| Roll Number | CE046 |
| ID | 19CEUEG017 |
| Subject | Big Data Analytics |

**Aim:** Leveraging machine learning using Mahout like tools.

* **Installing Mahout:**
* Cloning the git repository of the Mahout.



* **Installing Maven:**
* We need to download the maven.
* Then, we need to add environment variables such as MAVEN\_HOME and M2\_HOME.
* Finally, by adding path of the bin directory we are ready to use it.



* **Java program to run the recommendation on the dataset with the help of the mahout library.**

|  |
| --- |
| import java.io.File;  import java.util.List;  import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;  import org.apache.mahout.cf.taste.impl.neighborhood.ThresholdUserNeighborhood;  import org.apache.mahout.cf.taste.impl.recommender.GenericUserBasedRecommender;  import org.apache.mahout.cf.taste.impl.similarity.PearsonCorrelationSimilarity;  import org.apache.mahout.cf.taste.model.DataModel;  import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;  import org.apache.mahout.cf.taste.recommender.RecommendedItem;  import org.apache.mahout.cf.taste.recommender.UserBasedRecommender;  import org.apache.mahout.cf.taste.similarity.UserSimilarity;  public class Recommender  {      public static void main(String args[]){  *try*          {  *//Creating data model*              DataModel datamodel = *new* FileDataModel(*new* File("data")); *//data*  *//Creating UserSimilarity object.*              UserSimilarity usersimilarity = *new* PearsonCorrelationSimilarity(datamodel);    *//Creating UserNeighbourHHood object.*              UserNeighborhood userneighborhood = *new* ThresholdUserNeighborhood(1.0, usersimilarity, datamodel);              System.out.println("User Neighborhood information");    *//Create UserRecomender*              UserBasedRecommender recommender = *new* GenericUserBasedRecommender(datamodel, userneighborhood, usersimilarity);                int i = 2;  *for*(int i = 1; i <= 5; i++)              {                  List<RecommendedItem> recommendations = recommender.recommend(i, 3);                  System.out.println("User ID #"+i);  *for* (RecommendedItem recommendation *:* recommendations)                  {                      System.out.println(recommendation);                  }              }          }  *catch*(Exception e)          {              System.out.println(e.getMessage());          }      }  } |

* **Pom.xml file for dependencies:**

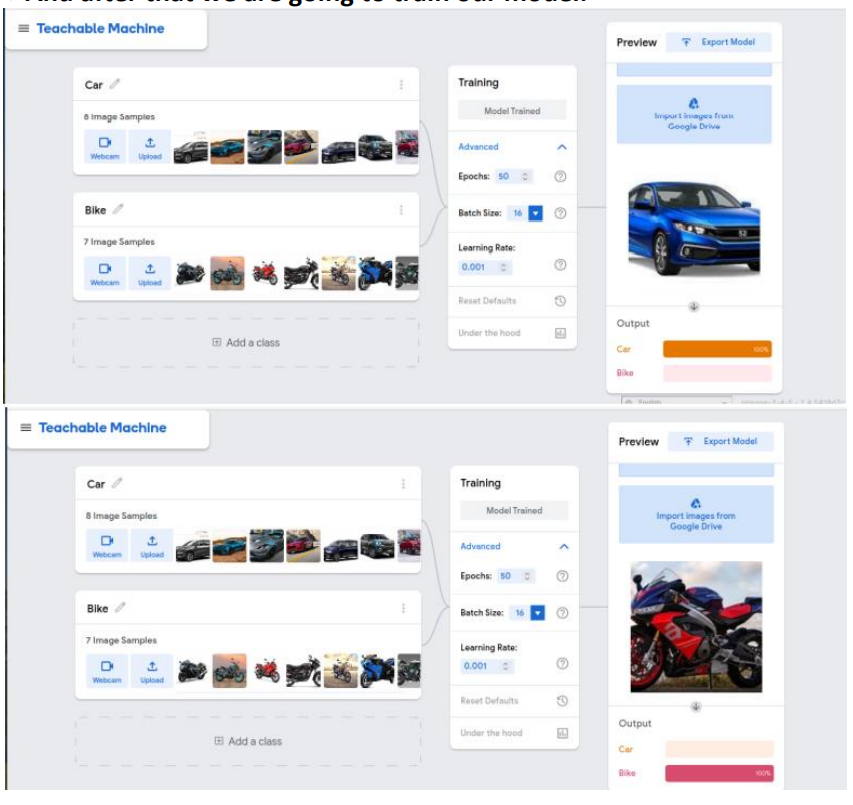
|  |
| --- |
| <project      xmlns=http://maven.apache.org/POM/4.0.0 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">      <modelVersion>4.0.0</modelVersion>      <groupId>Mavens</groupId>      <artifactId>demo-mahout</artifactId>      <version>0.0.1-SNAPSHOT</version>      <name>demo-mahout</name>      <description>Demo project for Mahout Maven Build Demonstration</description>      <properties>          <java.version>1.8</java.version>      </properties>      <build>          <sourceDirectory>src</sourceDirectory>          <outputDirectory>build/classes</outputDirectory>          <plugins>              <plugin>                  <version>3.1.0</version>                  <groupId>org.apache.maven.plugins</groupId>                  <artifactId>maven-jar-plugin</artifactId>zzzzzzzzzzzzzzz                  <configuration>                  <archive>                  <manifest>                  <addClasspath>true</addClasspath>  *<!-- add you main class -->*                  <mainClass>Recommender</mainClass>                  </manifest>                  <manifestFile>src/resources/META-INF/MANIFEST.MF</manifestFile>                  </archive>                  </configuration>                  </plugin>              <plugin>                  <groupId>org.apache.maven.plugins</groupId>                  <artifactId>maven-dependency-plugin</artifactId>                  <executions>                      <execution>                          <id>copy</id>                          <phase>install</phase>                          <goals>                              <goal>copy-dependencies</goal>                          </goals>                          <configuration>                              <outputDirectory> ${project.build.directory}                              </outputDirectory>                          </configuration>                      </execution>                  </executions>              </plugin>                  <plugin>                  <groupId>org.apache.maven.plugins</groupId>                  <artifactId>maven-resources-plugin</artifactId>                  <configuration>                      <encoding>UTF-8</encoding>                  </configuration>                  </plugin>          </plugins>      </build>      <dependencies>          <dependency>              <groupId>org.apache.mahout</groupId>              <artifactId>mahout-core</artifactId>              <version>0.9</version>          </dependency>          <dependency>              <groupId>org.apache.mahout</groupId>              <artifactId>mahout-math</artifactId>              <version>0.13.0</version>          </dependency>          <dependency>              <groupId>org.apache.mahout</groupId>              <artifactId>mahout-integration</artifactId>              <version>0.13.0</version>          </dependency>      </dependencies>  </project> |

* **Mahout:**
* Apache mahout is an open-source project that is mainly used for creating scalable machine learning algorithms.
* It provides different machine learning techniques like preprocessors, clustering, regression, recommenders, distributed linear algebra.
* It also provides the MapReduce but it is deprecated.
* Algorithms of mahout are written on top of the Hadoop, so it works well in distributed environment.
* One of the applications of Mahout is, Twitter uses Mahout for user interest modeling.
* **Performing analytics using machine learning techniques:**
* Machine learning is a data analytics technique that teaches computers to do what comes naturally to humans and animals, learn from experience.
* Machine Learning uses two techniques:
* **Supervised learning:**
* Here we train machine using data that is well labeled which means some data is already tagged with the correct result.
* It learns with labeled data.
* Types: Regression, classification, Decision Trees, etc.
* **Unsupervised learning:**
* It is a training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.
* No training will be given to the machine.
* Platform for performing analytics.
* We need to group our examples into classes that you want the computer to learn.
* Train your model.
* You can export your model for your projects.

**Example:**

Dataset-1: Cars

Dataset-2: Bikes



**Example:**

Dataset-1: Husky

Dataset-2: Golden Retriever

Dataset-3: Pug

