FINAL PROJECT

Course: Mining Massive Datasets

Duration: 06 weeks

I. Formation

- The project is conducted in groups of **04 – 05** students.
- Student groups fulfill the requirements and submit the work according to the instructions below.

II. Requirements

Given datasets in the **datasets** folder, students conduct tasks below.

Data sets	Description
mnist_mini.csv	Hand-written digit images in the MNIST data set. 10000 data rows. Each row has 785 integers. • The first number: digit kind (0, 1, 2, 3, …, 9) • Remaining 784 numbers: pixels of grayscale images (28 x 28).
ratings2k.csv	Product rating data set. The first line is the header. • **index**: row index • **user**: user ID • **item**: product ID • **rating**: rating (0.0-5.0) 2365 remaining lines are data samples.
stockHVN2022.csv	Stock prices of HVN code in HOSE in 2022 (until Nov 18th) The first line is header. • **Ngay**: date

	• **HVN**: price 219 remaining lines are data samples.

a) Task 1 (2.0 points): Clustering

Use **mnist_mini.csv** for this task.

Students use **DataFrame** of **pyspark.sql** to handle data and use **matplotlib.pyplot** to visualize results.

Implement the k-Means algorithm (**pyspark.ml.clustering.KMeans**) with **k = 10**, in which data points at rows **0, 1, 2, 3, 4, 7, 8, 11, 18, 61** are assigned a weight **100** times greater than the others.

For each cluster, measure the average distance from data points to its centroid. Draw a bar chart to visualize the average distances.

Note: organize source code regarding to OOP model.

b) Task 2 (2.0 points): Dimensionality Deduction with SVD

Use **mnist_mini.csv** for this task.

Students use **PySpark** and the **SVD** algorithm to reduce the dimensionality of data points from 784 to 3.

Randomly select 100 processed data points. Use clustering results in task 1 to draw a 3D chart to visualize the distribution of selected points using **matplotlib.pyplot**.

Note: organize source code regarding to OOP model.

c) Task 3 (2.0 points): Recommendation with Collaborative Filtering

Use **ratings2k.csv** for this task.

Split the given data set into **training** and **test** sets with the fraction **7 : 3**.

Use **PySpark** and the **ALS** algorithm to investigate the model performance regarding to **Mean Squared Error (MSE)** and the number of "similar" users in the range of **[10; 20].**

Run inference to visualize model operations.

Draw a bar chart to visualize the correlation between MSE values and the number of "similar" users.

Note: organize source code regarding to OOP model.

d) Task 4 (2.0 points): Stock price regression.

Use **stockHVN2022.csv** for this task.

The problem is to predict the price fluctuation range of the next day given the ones of **k** previous days.

Students use records from **Jan** to **Jun** for the training set and the remaining part for the test set.

With each set, students create a column named “**fluctuation”** to store price fluctuation ranges using the following formula.

Range of date [k] = (Price of date [k] – Price of date [k-1]) / Price of date [k-1]

The first date has a fluctuation range of **0.0%.**

Students then create data frames with two columns,

- **ranges of 5 previous dates**: a vector consisting of the ranges of 5 previous dates.
- **today range**: the range of the next day.

Implement the **Linear Regression** model **(PySpark)** to predict price fluctuation ranges in the training set and then evaluate the model in the test set.

Measure **Mean Square Error** values in the training and test sets.

Use **matplotlib.pyplot** to visualize **Mean Square Error** values in the two sets.

Note: organize source code regarding to OOP model.

e) Task 5 (1.0 point): Multi-class classification

Use **mnist_mini.csv** for this task.

Students implement classifiers using **PySpark.**

- *Input: image vector*
- *Output: category*
- *Loss function: Cross Entropy*
- *Metric: Accuracy.*

Students study and apply common classifiers below.

- Multi-layer Perceptron
  https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier

- Random Forest
  https://spark.apache.org/docs/latest/ml-classification-regression.html#random-forest-classifier
- Linear Support Vector Machine:
  https://spark.apache.org/docs/latest/ml-classification-regression.html#linear-support-vector-machine

Students draw a **twin-bar chart** using **matplotlib.pyplot** to compare accuracies of models in training and test sets.

Note: organize source code regarding to OOP model.

f) Task 6 (1.0 point): Report

- Student groups compose a report.
- **THERE IS NO TEMPLATE. STUDENTS ARANGE CONTENTS IN A LOGICAL STRUCTURE BY YOURSELVES.**
- The report must include below contents
  - Student list: Student ID, Full name, Email, Assigned tasks, Complete percentage.
  - Briefly present approaches to solve tasks, should make use of pseudo code/diagrams.
  - AVOID EMBEDDING RAW SOURCE CODE IN THE PRESENTATION.
  - Study topics are introduced briefly with practical examples.
  - Advantages versus disadvantages
  - A table of complete percentages for each task.
  - References are presented in IEEE format.
- **Format requirements:** avoid using dark background/colorful shapes, students ensure contents are clear enough when printing in grayscale.

III. Submission Instructions

- Create a folder whose name is as

  CK_<Group ID>

- Content:

- **source.ipynb** → source code (remain all cell outputs)
- **source.pdf** → pdf of the notebook
- **report.pdf** → report.

- Compress the folder to a zip file and submit by the deadline.

IV. Policy

- **Student groups submitting late get 0.0 points for each member.**
- **Copying source code on the internet/other students, sharing your work with other groups, etc. cause 0.0 points for all related groups.**
- **If there exist any signs of illegal copying or sharing of the assignment, then extra interviews are conducted to verify student groups' work.**

-- THE END –