Data-intensive Programming

Programming Assignment

Version 1.3

Description

The course has a compulsory programming assignment that is done in groups of two students. Working alone is also ok. Groups are created in Moodle (even if you work alone). Deadline for the grouping is 10th of November.

Code repositories for the groups will be generated by the course staff after the grouping deadline. The repositories are created in https://course-gitlab.tuni.fi/. The data, a skeleton for the assignment and a test suite are published via the gitlab project. The skeleton project is not expected to update often so the course personnel will notify if it is changed.

You are given dataset which, in general, look like this:

a, b, LABEL
7.5459, 9.0568, Fatal
8.4951, 9.2684, Fatal
8.8131, 9.0922, Ok
…

Even if the amount of data is not actually big, you should write your implementation as if it was. Moreover, Spark should be programmed with Scala.

The assignments are evaluated with scale fail or pass. If you fail, you need to fix your submission. The group should commit a working project in the group's repository for submission. The code should be commented at least to some extent.

A basic passed solution implements the tasks 1-4. It is also possible to gain bonus points by implementing the bonus tasks. The points will be added to the sum after the exam has been passed. A possible reason for not getting points from bonus tasks is if the reviewers do not understand the solution due to lack of comments.

A simple test suite is given for simple testing of your solution. You can add own tests to the suite.

Task #1: Basic

The task is to implement k-means clustering with Apache Spark for two-dimensional data. Example data can be found in file data/dataK5D2.csv (ignore the LABEL in this task). The task is to compute cluster means using DataFrames and MLlib. K is given as a parameter. Data for k-means should be scaled but it is not required to scale the cluster centers back to original scale (see Bonus Task #6). See task1 in the assignment.scala file.

Task #2: Three Dimensions

The task is to implement k-means clustering with Apache Spark for three-dimensional data. Example data can be found in file data/dataK5D3.csv (ignore the LABEL in this task). The task is to compute cluster means with DataFrames and MLlib. The number of means (k) is given as a parameter. Remember to scale your data similarly to task 1. See task2 in the assignment.scala file.

Task #3: Using Labels

K-means clustering has been used in medical research. For instance, our example the data that could model some laboratory results of patients and the label could implicate whether she has a fatal condition or not. The labels are Fatal and Ok.

Use two-dimensional data (like in the file data/dataK5D2.csv), map the LABEL column to a numeric scale, cluster in three dimensions (including columns a, b and num(LABEL)) and return only the two-dimensional clusters means for two clusters that are most Fatal. Remember to scale your data similarly to task 1. See task3 in assignment.scala file.

Task #4: Elbow Method

Elbow method is used to find the optimal number of the cluster means. Implement a function (task4) which returns an array of (k, cost) pairs, where k is the number of means and cost is some cost for the clustering.

Bonus Task #1: Functional Style – 0.5 Point

Try to write your code in functional programming style. Use, for example, immutable variables, pure functions, higher-order functions, recursion, and mapping. Avoid looping through data structures. It is very likely that you will get this bonus task just by not making anything stupid.

Bonus Task #2: Efficient Usage of Data Structures – 0.5 Point

Use data structures efficiently. For example:

- Use caching or persisting if it is sensible.
- Consider defining schemas instead of inferring them.
- Avoid unnecessary operations.
- Adjust the amount of shuffle partitions if it is sensible. Reason in comments why or why not to adjust the amount of shuffle partitions.
- When using Datasets, do not use lambdas with higher-order functions so that it leads to extra serialization and deserialization.

Bonus Task #3: Dirty Data – 1 Point

The program should handle dirty or erroneous data somehow. You must reason in the code comments why the data is handled in the way you chose. There must be also a few additional test cases that show some dirty data cases.

Bonus Task #4: Pipeline – 1 Point

Design a machine learning analysis that benefits from a machine learning pipeline. Explain and reason in the code comments. The task can include hyperparameter optimization. Make a few additional tests to show the pipeline running in action.

Bonus Task #5: Visualization – 1 Point

Try to use Scala to make a graph that presents the elbow of Task #4. If it is too difficult to use Scala for some reason (tell why), you can produce the graph using some other programmatic technique. If you use some other technique, also add an image of the graph and the source code that produced the graph to the group's repository in the same folder with the other source code files.

Bonus Task #6 – 1 Point

Scale the cluster centers back to original scale.

Submission

The assignment is submitted by committing the source files into the group's repository and submitting the latest commit hash to Moodle. In addition, if you are aiming for the bonus points, mention which bonus tasks you have done. For example, "BT1, BT2, BT5".

The deadline is 10th of December.