COMP2043.GRP Interim Group Report

Team02
Machine Learning Dataset Parsing Tool

Group Members
Boyuan Ma - zy22053 (Team Leader)
Xinyan Li - zy22043
Hao Liu - zy22046
Xinjie Pang - zy22055
Marios Igkiempor - scymi1

Supervisor
Chris Roadknight

December 2018

Contents

1 Description of the Problem

Our team's project is centered around a system that classifies datasets by the best type of machine learning approach to take in order to best analyse the data. It is proposed that the system will be able to parse a dataset, analyse its features and propose a type of learning (supervised, semi-supervised and unsupervised) that can best model the dataset based on the analysed features.

The client intends to use the project as both a prefilter for machine learning and as a teaching aid, and so the program should be able to provide useful information as to how the machine learning approach was derived.

Datasets that the program will analyse will be provided by the client. The possibility of analysing new datasets will be a target that the team will strive for, however it is not essential for the core functionality required by the client.

In essense, the project will take a dataset and tell the user the most optimal machine learning strategy with which to analyse the data.

2 Background Information and Relevant Research

2.1 Machine Learning

The team has to figure out how many machine learning methods would be provided in our website, their respective principles, and how and when to use each one. There is many Test suitability of open source machine learning toolkits such as H2o [1], Weka [2] and various Python and R packages. These machine learning toolkits could be used to analyze different datasets. In this early stage of the project, the team has chosen to use Weka to understand each machine learning method.

3 Requirements Specification

4 Initial Design Ideas

5 Key Implementation Decisions

5.1 Database Design

5.1.1 NoSQL vs SQL

Heterogenous big data benefits from NoSQL for a few reasons. Firstly, there is no predefined schema that the data has to conform to, which is useful for our application as we have to save data from a multitude of different data sources in the same data store. We cannot index all the data efficiently beforehand, and therefore cannot define a large schema that all datasets will conform to.

Even if we could, a large schema would waste a lot of space, as most datasets would only use a small subset of the schema. This would also make searching the database very inefficient, as we would have to search through potentially hundreds of unused fields.

Removing schemas also makes the database faster to query [3]. Because each dataset is stored in one document rather than spread across multiple tables, the program knows exactly where to look for the data set rather than having to search through multiple tables. This helps when we are having to search through hundred of datasets to find the data set we are interested in.

5.1.2 Lack of Relations Between Datasets

The data sets that our program will use contain almost no relations. As such, using a relational database would waste a lot of the functionality associated with relations, and cause a lot of uneccessary overhead. Instead, a non-relational database will just store data sets independently of each other. Querying the database will simply involve returning a JSON object, with no links to other data sets.

5.2 Accepted File Types in the System

When studying Weka, the team found that there are many file types that machine learning tools could use, for instance Arff data files, CSV data files, XRFF data files, amongst others. After some research into the UCI Machine Learning Repository [4], the team decided to use CSV data files to store datasets into the database and to be used by the machine learning tools. This is because CSV files make it convenient for the team to transform the .data files and .name files found on the UCI repository into a single file. CSV files are also easy to store in the NoSQL database. Knowing the format of the data is useful for data pre-processing and analysing.

6 Problems Encountered

7 Time Planning and Project Management

At the very start of the project, time planning was discussed but was not followed or enforced as much as it should have been. As such, a Gantt Chart was devised and is shown in figure 7. This Gantt chart has helped us stick to deadlines and structure our work.

Alongside the Gantt chart, we have a Kanban board which is hosted on Trello.com. Scrum was discussed as an alternative approach, but Kanban was chosen for some reasons outlined below:

- Teams using Kanban can cope with mutable requirements flexibly. Teamwork will continue with changing project environment.
- Formal and informal meetings can be held as frequently as needed and agile working process can be pushed favorably by team communication.
- Due to the main process of the project having been decided, Kanban is a good system as it allows us to vizualise every stage of work and everyones processing.
- For a team with less development experience, it is difficult to deliver an executable program in a short time. Scrum would be difficult to practice because of its demand on techniques and experience.
- Scrum need to stipulate working time for every iteration, which is difficult for teams with less development experience. Kanban only displays stage missions but not time limits.
- Scrum needs to declare a Product Owner, Scrum manager and Team which might cause confusion in an inexperienced team. Kanban can make team members focus on their work and research.

References

[1] "Home." [Online]. Available: https://www.h2o.ai/

[2] "Weka 3: Data mining software in java." [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[3] "Mongodb and mysql compared." [Online]. Available: https://www.mongodb.com/compare/mongodb-mysql

[4] "Uci machine learning repository: Flags data set." [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html