COMP2043.GRP Interim Group Report

Team02
Machine Learning Dataset Parsing Tool

Group Members

Boyuan Ma - Team Leader
Xinyan Li
Hao Liu
Xinjie Pang
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

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[1]. 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.

6 Problems Encountered

7 Time Planning and Project Management

References

[1] “Mongodb and mysql compared.” [Online]. Available: https://www.mongodb.com/compare/mongodb-mysql