

Exploring Machine Learning:
The ID3 algorithm
Design Document

Mohammed Ibrahim
Computer Science Department
College of Science, Swansea University
Swansea, SA2 8PP, UK

May 25, 2012

Abstract

This document outlines all of the design principles which we have used for this project and it also discuss different methodologies. With respect to design document I have included more two document in which all the source code of our program has documented using Javadoc and Doxygen

Contents

1	Introduction	3
2	Project Strategy	3
2.1	Methodologies	3
2.2	Documentation Plan	3
2.3	Documentation of source code	3
2.4	Security	4
3	UML Diagrams	4

1 Introduction

The development of Machine learning project has involved several key design decisions. I will be describing all of the design aspects that have guided this project.

2 Project Strategy

2.1 Methodologies

At the initial stage when designing the project we have considered several different development methodologies to base the design of the project. The waterfall model is a solid well established model and has a linear structure that has the potential to produce a good project, so we have decided to implement our project using this model. We have taken each phase of the model and applied to each section of the project.

2.2 Documentation Plan

All documentation created during this project is complete and to the point. At the beginning of the project, requirements and specifications document submitted by giving the concise idea of the project. In the middle of the term the interim document has been given to show the progress of our project. We ensure that extensive documentation is produced to fully convey all of the aspects of the system, including design, testing and user manual. The documents are broken down into small sections to provide sufficient detail at every level; this then in total makes a full structured document. The design document describe the project design in a formal notation by showing and explaining the UML diagram. The two Java programs has been well documented by using the tool called *Doxygen*. The user manual will provide the screen shots of the output with the clear instructions to help user to run the program.

2.3 Documentation of source code

To implement the software we will be using Doxygen and Javadoc tools.

To refer from [2](page 1): Doxygen Documentation : Doxygen is a documentation system which generate multiple programming languages. Example C, C++, Java, Python, PHP. It is a tools which we use to write a software reference documentation. The documentation is written within the code, and is relatively easy to keep up to date. It helps in three ways:

- It can generate an on-line documentation browser in the form of HTML code and off-line reference manual in Latex from a set of documented source files. It also generate output in (Ms-Word) in RTF format, PostScript, hyperlinked PDF, compressed HTML, and Unix man pages. The documentation is extracted directly from the sources, which makes it much easier to keep the documentation consistent with the source code.
- doxygen can be configured to extract the code structure from undocumented source files. Doing this type is very useful find quick in large source distributions.
- One can ‘abuse’ doxygen for creating normal documentation.

Doxygen is developed under Linux and Mac OS X, it is highly portable. It runs on most other Unix flavours as well. It supports a number of outputs formats where HTML is the most popular one.

To refer from [3](page 1): Javadoc : Javadoc is a documentation generator from Sun Microsystems for generating API documentation in HTML format

from Java source code. To refer from [1](page 1): It is a tool that parses the declarations and documentation comments in a set of source files and produces a set of HTML pages describing the classes, interfaces, constructors, methods and field.

2.4 Security

To ensure that the two programs is secured, that is secure from corruption and accidental deletion we have created repository using *GITHUB* and all the documents and source code has been kept in the repository.

3 UML Diagrams

::Default

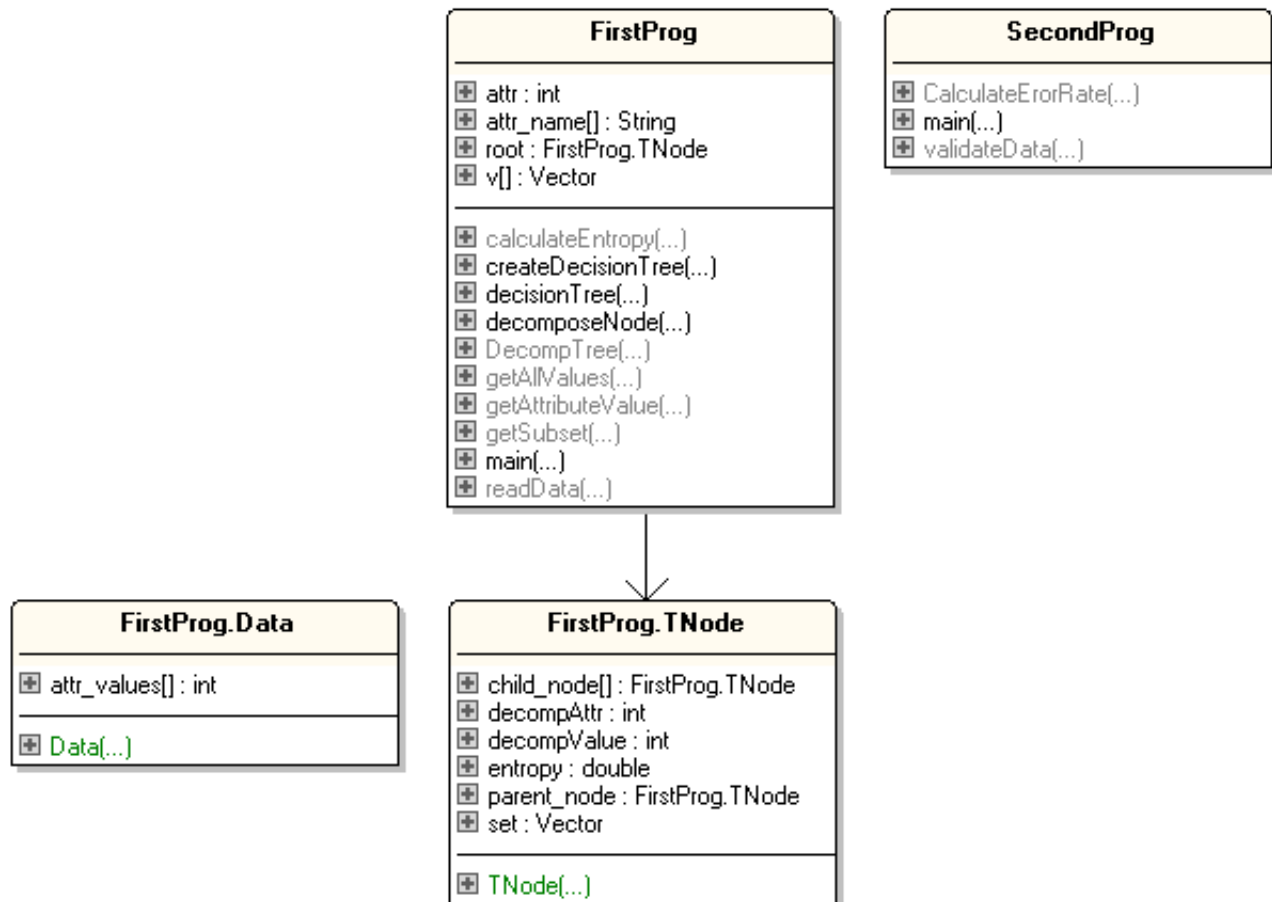


Figure 1: UML Diagram

With this design document I have also included the source code. All the source code are well documented using **Doxygen** and **Javadoc**.

References

- [1] Oracle. Javadoc. <http://docs.oracle.com/javase/1.5.0/docs/guide/javadoc/index.html>, january 2004.
- [2] Dimitri van Heesch. Doxygen documentation. <http://www.stack.nl/~dimitri/doxygen/>, October 1997.
- [3] Wikipedia. Javadoc. <http://en.wikipedia.org/wiki/Javadoc>, january 2012.