CSCM10: Specification

Andy Gray
445348

8/05/20

Swansea University
Prifysgol Abertawe

Abstract

Abstract here

1 Introduction

1.1 Overview of the Problem

1.2 Overview of Proposed Solution

1.3 Aims and Objectives

1.4 Structure of the Document

We will first look into background research. Explaining what machine learning with an introduction to what it is, with going into more detail about the different ML techniques. These include supervised vs unsupervised, classification vs segmentation, data-driven approaches, features and dimensionality. An introduction to the machine learning classifiers proceeds the techniques. The classifiers include logistic regression, decision trees, random forests, SVMs and neural networks. We will also look into the literature of gamification and educational games. We will be looking at their impact, and how they get used, what their purpose is how they are adopted and created. We will then explain our proposed methodology, along with our development intentions for the educational game. Including the proposed tools we plan to use and the project management techniques we intend to implement.

2 Background Research

2.1 Intro to Machine Learning

2.1.1 Supervised vs Unsupervised

2.1.2 Classification vs Segmentation

2.1.3 Data-Driven Approaches

2.1.4 Features and Dimensionality

2.2 Introduction to Machine Learning Classifiers

2.2.1 Logistic Regression

2.2.2 Decision Trees

2.2.3 Random Forests

2.2.4 SVMs

2.2.5 Neural Networks

2.3 Introduction to Gamification and Edu-games

2.3.1 Impact

2.3.2 Usage

2.3.3 Purpose

2.3.4 Adoption

2.3.5 Creation

2.4 Edu-games in Computer Science

2.4.1 Basic Computer Science

2.4.2 Machine Learning

2.5 Summary explaining intent to develop the Edu-game for ML

3 Proposed Methodology

3.1 Introduction to Proposed Application

Our main aim is to create an educational game, that aims to teach the main concepts of machine learning. The application will offer multiple game modes like 'claim the screen', 'pin the point on the decision boundary', to name a few. The application will allow the players to interact with the primary ML model to be able to get an understanding of how the application works. Along with giving the user a video explanation and text of how the algorithm works as well as having links to critical scientific papers to gain a better in-depth understanding.

3.2 Overview of proposed Critical Features

3.2.1 Gameplay

3.2.2 Different players? Single vs Multiplayer?

3.2.3 Different Classifiers?

3.2.4 Educational Component

Explanation of scoring? Explanation of model?

3.3 Overview of proposed optional features

3.3.1 Power-ups?

3.3.2 Alternative game modes?

Underlying function for levels? King of the Hill? Domination?

3.3.3 Human vs AI?

3.3.4 Unlockables?

Code Snippets Publication links

4 Proposed Mock-up prototype images

5 Proposed Implementation Approach

5.0.1 Tools

We plan to make the game using Python 3 [**?**]. We plan to use this language due to the vast amount of packages and libraries available. Python has a vast amount of support within the Data Science and Machine Learning community, and it also has packages available to allow the create games and web applications.

Visual Studio Code (VS Code) will be the chief text editor for implementing the game. Due to the nature of text editor, it will allow us to use all the Python libraries as well as any of the iPython notebook files. iPython notebooks are the primary file type for Jupyter Notebooks whereas VS Code allows user to edit multiple programming languages, which include Python and Jupyter Notebooks, in one place. However, with it being a text editor, it does lack the more advanced features we would expect in an IDE, but these are features we do not need.

Jupyter Notebooks allows the code to be split up into cells, allowing us to be able to test out segments of code, without having to run all of the code in the file, which can be very helpful when trying to develop and implement some modularity features within the application.

We will be using Trello for the kanban tools. "Kanban" is the Japanese word for "visual signal" [**?**]. Using Kanban boards allows us to keep our work visible, this is to allow others to see what it is we are doing, and what is needed to get done. These will allow everyone to see the full picture and keep everyone on the same page.

David Anderson discovered that kanban boards get split into five components: Visual signals, columns, work-in-progress limits, a commitment point, and a delivery point [**?**].

Kanban teams write all their project's work items onto cards, and these are usually one per card. The kanban board gets split into columns, with each column representing an activity which composes the workflow. All the cards change between the workflow until the activity is complete. The column workflow titles can be as simple as to do, in progress and completed. However, David suggests that there should be a work in progress (WIP) limit [**?**]. When a column has reached the limit, of three cards, all

team members get expected to focus on the cards in progress. The WIP limits are critical for exposing bottlenecks in the workflow and maximizing flow. WIP limits give an early warning sign that too much work commissioned. Backlogs of ideas are where the ideas of the team and the customers get placed. The moment an idea gets picked up by a team member and work begins, this gets referred to as the commitment point [**?**]. When the product is finished and is ready for deployment, this stage gets referred to as the delivery point. The overall aim of the kanban is to take a card from the commitment point to delivery point as quick as possible.

5.0.2 Python Libraries

The Python libraries that we will be using are Pygame for game development. [Add a brief description of it here]. This package will be the main one for creating the game's interface and game logic.

For the main implemented machine learning algorithms that are within the game, a package called Scikit-learn will get used. [Add a brief description of it here]. This package will allow us to implement the algorithms [list them here] using the Python language.

6 Proposed Project Management

6.1 Gantt chart

Chart here!!!!

References