Project Name (TBD)

Requirements Specification

authors here (TBD)

December 1, 2020

Contents

1 Product Overview

Project Name (TBD) is a web-tool enabling its users to create, edit and manage machine learning models on different sensor data. Project Name (TBD) consists of two primary components:

1. A server application that handles model training and storage, and makes these models available through a REST API. Supports multiple models and multiple users.

2. Client applications to manage and interact with the model, e.g. adding new data points, identifying given data points and setting parameters for model training. A model can be utilized on different clients at the same time. Clients have different capabilites depending on the platform, mobile clients allow input of new data points in form of raw sensor data, while desktop clients allow management features.

Both components together allow for a streamlined and easy-to-use machine learning experience for both the unexperienced and the expert.

2 Purpose

2.1 Core criteria

- The web pages are the graphical user interfaces (GUI) for users.
- The desktop client presents a QR code that when scanned by a mobile client connects the mobile client to the same workspace.
- The QR code contains the link to the workspace and the same link can be used to reaccess the workspace later.
- The desktop client allows management features.
- The desktop client allows creating new labels.
- The mobile web client allows input of new data points in form of raw sensor data.
- The mobile web client allows for selecting the sensors the users wish to use. (e.g accelerometer)
- The mobile web client allows for configuring the recording.
- The mobile web client gives a real time feedback of sensor data to the user.
- The mobile web client sends the collected sensor data to the web server.
- The desktop web client displays the collected samples.
- The desktop web client allows preprocessing and filtering of the data.
- The features which were selected by the user are computed and stored in the database.
- The sent sensor data is stored in the central database in a unified format.
- Users can select a machine learning model they wish to use.
- Users can change the hyperparameters of the machine learning model.
- A model is trained and stored in the database.

- The link to the workspace with the trained model can be used to identify the sensor data on the mobile web client. This identification happens in real time.

2.2 Optional criteria

- Links for previously accessed workspaces can be stored on the browser to ease reaccess.
- Other data capturing devices are supported, e.g. Arduino.
- The workspaces can be protected with a password.
- The desktop web client displays a status sign if a data collecting device is currently connected to the workspace, e.g. a green sign if connected and a red sign otherwise.
- The mobile web client can define triggers if something is detected, e.g. play a sound.
- The desktop web client can show performance metrics to the user. (OPTIONAL ?)
- The user can train multiple models from the same data sample.
- A read-only link to each trained model in a workspace can be generated by the user.
- Workspaces can be deleted together with the related data samples.

2.3 Exclusion criteria

- The mobile web client does not have a QR scanner.
- API calls are not authorized.
- TBD?

3 Usage

3.1 Area of Application

The application is for collecting and labeling sensor data, training a model from the collected data and serving the stored resulting model to the user to classify actions in real time.

3.2 Target Groups

The target group of this application is people that wants to create a machine learning model from mobile sensor Data. That includes:

- People new to machine learning who wants an easy tool to discover machine learning and create models that they can use.
- Experienced engineers that have knowledge of machine learning but need a fast tool to collect data and create a model.

We assume a rather technical audience.

3.3 Operating Conditions

The tool is a web application that can be run on any modern browser. A web browser must be installed on the user device to view the webpage. The application can be used from anywhere with a decent network connection.

Service Duration: 24 hours a day

4 Operating Environment

Database and the server appliance both run on a server, whereas different web browsers may be used for the clients. A Docker image may be provided for the server appliance.

4.1 Software

4.1.1 Server Appliance

- MongoDB support
- Recent versions of Node.js and Python

4.1.2 Client Devices

- A modern web browser (Chrome 87+, Firefox 83+) supporting:
  - Sensor APIs
  - Web Storage API

4.2 Hardware

4.2.1 Server Appliance

The server has to be fast enough to support all clients. This depends on the expected number of clients. Most computations will be done on the server. **TOO ARBITRARY?**

4.2.2 Client Devices

- Video Device

5 Functional Requirements

5.1 Main Functions

These functions must be implemented in order to fulfill the core criteria.

5.1.1 Web Client

The web client supports both desktop and mobile modes. The functionality that will be displayed is determined by the device information of the browser.

Desktop Web Client

/**F010**/ Show a welcome page

/**F020**/ Provide a QR code on the welcome page to create a workspace

/**F030**/ Show the management panel for the created workspace

/**F040**/ Allow creating and renaming labels for the actions to be recorded

/**F050**/ Display the collected data on the management panel

/**F060**/ Enumerate the collected data chronologically on the management panel

/**F070**/ Allow naming of the model on the management panel

/**F080**/ Allow selecting the possible actions on the data on the management panel

/**F090**/ Request the processing of the data according to the selected options

Mobile Web Client

/**F100**/ Show a configuration page with available labels

/**F110**/ Allow configuring the countdown duration until the recording starts

/**F120**/ Allow configuring the recording duration

/**F130**/ Allow configuring each sensor individually

/**F140**/ Show a button to initiate the recording

/**F150**/ Show a countdown page

/**F160**/ Display the configuration on the countdown page

/**F170**/ Show a recording page

/**F180**/ Display the sensor data in real-time as curve graphs

/**F190**/ Show a recording completed page with an option to go to the configuration page for a new recording

5.1.2 Server API

/**F200**/ Generate and serve a link for a new workspace

/**F210**/ Serve workspace information

/**F220**/ Create and rename labels

/**F230**/ Accept data from the mobile client

/**F240**/ Create a model in the workspace

/**F250**/ Rename a model

/**F260**/ Initiate the configured model training

5.1.3 Data Processing

/**F270**/ **TBD, after workshop**

5.2 Extending Functions

/**F280**/ Show the previously visited workspaces on the welcome page

/**F290**/ Make the workspace link password protected

/**F300**/ Delete an existing model in the workspace (Server API)

/**F310**/ Create a read-only link for classifying (Server API)

/**F320**/ **should the desktop client show data being collected in real time?**

6 Data

TBD

6.1 System Data

TBD

6.2 User Data

TBD

6.3 Database

TBD (Data-Warehouse data ?)

7 Nonfunctional Requirements

7.1 Usability

- /N010/ User does not need to register to use the web-tool.
- /N020/ User should be able to record a new sample in less then 3 interactions.
- /N030/ User should be able to train and deploy the machine learning model (/FXXX/) with a single click.

7.2 Swiftness

- **lazy load components?**
- /N040/ Sth about page load speed - might be too restrictive, cannot assess yet.

7.3 Maintainability

- /N050/ Codebase is well documented to ease bug fixes.

7.4 Portability

- /N060/ Desktop web client /FXXX/ should be functioning on browsers newer than Chrome version XX, Firefox version XX, ...
- /N061/ Mobile web client /FXXX/ should be functioning on browsers newer than Chrome version XX, Firefox version XX. ...

7.5 Security

- /N070/ A workspace is inaccessible to other users without the QR code or the link to the workspace.

7.6 Scalability

- /N090/ A maximum of 10.000 ?? workspaces with a maximum of 10 ?? users at the same time is supported.

8 System Models

TBD

8.1 Scenarios

TBD

8.2 Use Cases

TBD

9 User Interface

TBD

9.1 Desktop

TBD

9.2 Mobile

TBD

10 ??Qualitatszielbestimmung??

11 Test Cases and Test Scenarios

TBD

11.1 Test Cases

11.1.1 Main Test Cases

These test cases are necessary for the web-tool to work properly and succeed in serving basic requests.

- /T010/ Access web page (/F010/)
  - /T011/ via Google Chrome,
  - /T012/ via Mozilla Firefox.
- /T020/ Show a QR code.
  - /T021/ Scan the QR code to create a new workspace.
  - /T022/ Scan the QR code to access an already existing workspace.
- /T030/ Access an already existing workspace with a link.
- /T050/
- /T040/ Select which sensors to use for recording.
- /T050/ Set the duration for the countdown.
- /T060/

11.1.2 Extending Test Cases

11.2 Test Scenarios

TBD

12 Development Environment

12.1 Operating Systems

- Windows 10 with WSL1 / WSL2
- Ubuntu 20.04.1 LTS

12.2 IDEs, Editors

- VS Code

12.3 Versioning

- git
- Github for hosting

12.4 Integration

- **(we'll need a CI server, TECO has one?) FIXME**

12.5 Miscallaneous

- LaTeX for documents
- 3D Paint for mockups
- VS Code Live Share for communication
- Discord for communication

And many more to be added later...