

Artificial Intelligence and Data Engineering

Data Mining and Machine Learning

**Design and development of the application “DrawIt”**

**Students**

Verena Penner

Charlotte Leonard

Academic Year 2021/2022

# **Table of contents**

[**1.** **Design** 1](#_Toc92833768)

[**2.** **Machine Learning** 2](#_Toc92833769)

[**3.** **Implementation** 3](#_Toc92833770)

[**Bibliography** 4](#_Toc92833771)

# **Design**

* Application „DrawIt“
  + Consists on a drawing game
  + is orientated to the QuickDraw game from Google Creative Lab (and is a kind of rebuild?)
  + Computer asks you to draw a figure and proposes 5 categories as a list
  + User has to draw one of the figures in the graphics editor Microsoft Paint
  + The sketch has to be saved and uploaded to the application by drag and drop
  + The computer automatically tries to guess the right category by using a classification method
  + The computer shows you also the most similar drawing from the database (as a kind of evidence)
  + No possibility to let the computer learn new categories or improve the classifier by the user
  + It is an easy and basic game where computer and user are playing together
* Requirements
  + The application is available for anybody (who has the access to a computer)
  + User:
    - For lonely people
    - For kids to have fun and occupation
    - For people who are interested in drawing
    - For people who are interested in machine learning stuff
  + For the user obligatory: editor Microsoft Paint
  + Self explanatory and userfriendly user interface (every step of the game is well illustrated)
  + Userfriendly regarding to the performance
  + Best accurated classification model is implemented for a high accuracy rate
  + Python obligatory? Internet?
* Use case description
  + Start the application?
  + Upload a drawing (set link? Drag and drop?)
  + Confirm drawing for classifying
  + View a similar drawing of the database
  + Renew the game
* Architecture
  + Code written in the programming language python
    - Main libraries for working with images and machine learning algorithm: xy
  + Client-Side
  + Server-Side

# **Machine Learning**

Short machine learning definition? (what is it, target, mention approaches…)

* Dataset
  + Available on the open platform <https://www.kaggle.com/google/tinyquickdraw> (and <https://github.com/googlecreativelab/quickdraw-dataset>)
  + Is a large dataset from 2018 with over 50 million drawings of users from all over the world
  + Contains in total 345 different categories and the volume is around 28 GB
  + Dataset contains following attributes (dimensionality: 6)
    - Key
    - Word (= category)
    - Countrycode
    - Timestamp
    - Recognized
    - Drawing
  + Dataset available in a raw and a preprocessed format
    - Simplified drawings are online in format of .ndjson, .bin and .npy
  + Dataset can be requested via API
    - Simplified drawings as binary files (.bin)
    - makes the handling of the high amount of image data easier
    - A python library is available
    - Downloads the data files and caches them locally
    - API documentation and useful code examples are available here: <https://quickdraw.readthedocs.io/en/latest/>
  + For our application we use a subset of 5 categories and 500 drawings per category. In this way the complexity is reduced, and the performance enhanced. The categories we have chosen are “xy”, …. The figures are as different as possible such as the classifier is getting worse when the only five chosen categories are quite similiar
* Data pre-processing
  + Format of drawing data different
  + Drawing numeric attribute, category is nominal
* Attribute selection
  + Word and drawing attributes are necessary for the application
  + Using selection method didn’t make sense in regarding to the required features of the application
* Model learning
* Model validation
* Conclusion: model selection?

(different algorithms should be experimented for selecting the most suitable model; analysis using the cross-validation methodology should be carried out)

Still to mention:

* Many strokes, but we work with whole pictures in our application
* Datatypes of attributes
* Simplified data algorithm

# **Implementation**

Conclusion?

API can lead to poor performance

# **Bibliography**

Diagrammtyp

*nstanziiert*