MARMARA
UNIVERSITY

CSE 3063 Object Oriented System Design

Doç. Dr. Murat Can Ganiz

Department of Computer Engineering

Marmara University

Fall 2020

Project I

Group 20

150319067 - Elif Sünnetçi

150119933 - Nurhande Akyüz

150119686 - Diala Jubeh

150119900 - Emre Okul

150119746 - Alper Dokay

150119630 - Bilal Tan

150119640 - Ayberk Altuntabak

Requirement Analysis Design

Introduction

The world has become a well-designed place thanks to the developments in technology. Many companies or institutions start using technological tools intensively. As the ongoing features are still on the way, people all wonder about the concept of Artificial Intelligence. Artificial intelligence helps people to understand a circumstance, handle an issue or offer a solution to a problem with various options. As the concept implies, the main purpose is to come up with new solutions to problems. One of the fields of Artificial Intelligence is machine learning and it has been widely used in scientific activities and modern research laboratories. Machine Learning requires some strict steps to follow in order to reach a goal with high accuracy. The key is generally to train the models formed by scientists or developers. Here in this paper, we offer a new tool for scientists and developers to be used in preprocessing steps of any machine learning development project.

Purpose

In machine learning, data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it. Today, most practical machine learning models utilize supervised learning, which applies an algorithm to map one input to one output. For supervised learning to work, you need a labeled set of data that the model can learn from to make correct decisions. For instance, Doccano (https://github.com/doccano/doccano) is a nice example as an open source data labeling tool for textual data. Data labeling typically starts by asking humans to make judgements about a given piece of unlabeled data. To put it in a nutshell, our aim is to provide a user interactive console-based program to facilitate the labeling process in the preprocessing phase of any machine learning project. In addition, the ability that each user can relabel the same instance many times so that we can get the consistency of the performance of that specific user.

Intended audience

Labeling texted data is an enormous field that targets many audiences that will employ this sentiment analysis in different manners. Mostly, people who seek to build accurate (Artificial intelligence / Machine learning) projects in which they need to employ and maintain large sets of text data.

Intended use

The project act as a text annotation tool, sooner the annotated texts done will be applied as a basis for many projects like businesses (for instance, recommendation systems), machine translators, smart chatbots, directing scientific research, as well as political parties who need to figure out how they are seen by the public and many more applications,... etc.

Use Case

Use Case: Label a Dataset
Actors: User, System

1. System categorizes the data in the dataset in terms of creating labels and instances.
2. User logs in to the system.
3. System offers the user a list of instances, in which the user can assign labels accordingly.
4. System sends an output related to the dataset's instances that user has labeled, that includes performance Metric and Report for each user (How many datasets assigned to the user and consistency of the performance of user, time spent for labeling), Performance Metric and Report for each dataset (Completeness of the dataset and information about users' contribution) and Performance Metric and Report for each instance. (label assignments and information about users' contribution).

Scope and Scenario

The scope of this project is to develop a Java based object oriented implementation of a Data Labeling System. This project consists of creating a user interactive console-based program in order to form a consistent and accurate textual data labeling system by using Java programming language in an object oriented manner and JIRA to keep track of tasks for each sprint of the project. As a resource for this project, user information and labels will be taken

from a json file and the product of the project will be an user interactive data labelling system. Moreover, checking the user consistency will be an essential part to take the labeling of the user into consideration or not. Developing the project without meeting the team members is a constraint for this project. At the end of the project, requirement analysis document, UML class diagram, UML sequence diagram, and java code will be delivered as artifacts. The project will be completed by January 2020.

Glossary

JIRA: is a software that is developed in order to help teams manage their work. It provides to practice agile methodologies such as scrum.

UML: short for unified modelling language. It is a standardised language which helps developers to keep track of the system. UML is a visualized language which represents the system with diagrams.

Data Labelling: is the process of detecting related tags for the given data instance.
Machine Learning : is one of the branches of Artificial Intelligence. It is an application that gives the ability to learn automatically by using its models and it is able to improve its system by experience without being programmed again. Machine Learning models are trained to find common patterns in data to make some estimations for new data.

Use Case: is a set of cases or scenarios for using a system, tied together by a common user goal. It emphasizes the user's part of view and explains everything in the user's language.

Class: is a modeling tool provided by the programming language for a program or application to use in representing real world objects. Classes on this program are used to define data and user. Class Label's are chosen according to the objects they represent.

Instance: is a variation of an object created from a class. Users can create these object variants by using constructors.

Dataset: is usually designed for analysis rather than constantly updating from different users, so it represents the end of a data collection or a snapshot of a particular time.

Labeling Mechanism: In this program, it is done by manually tagging the data that the user sees in the console application. This mechanism can be developed by gaining artificial intelligence capabilities.

User: is a person using a computer or network service. In this program, the person who classifies the data is called the user.

Consistency: is a performance measure associated with User, that expresses the user's uniformity in labelling.

User interface: a platform for interaction between the user and the system; in order to label instances easily.

User Needs

A software requirement is a capability needed by the user to solve a problem or to achieve an objective. In other words, requirement is a software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation. Therefore, our program first manually identifies important sections of text or tags the text with specific or random labels to generate its training dataset. To give an example, in some cases such as social media applications or websites a comment may be toxic, obscene and insulting at the same time but it may also happen that the comment is non-toxic. In conclusion, we are going to label customer's comments via mechanisms designed by us specifically or randomly.

System Features and Requirements

Functional Features and Requirements

- Users of the program should be able to mark any instances from any dataset according to the labels defined multi-times.
- Datasets should be pre-ready to be used by the users whenever they are called.
- Program should describe a dataset before the user tags them so that users can be informed about the dataset as well.

- Each user can label multiple instances in multiple datasets and the program should be able to go over the inputs and create related outputs.
- There must be different approaches to be used during the program runtime. These mechanisms should be flexible to be differentiated and selected by the user.
- The program should be capable of handling file import in the beginning of the program.
- The program must have the ability to create random users if preferred by the admin.
- The program should output a dataset consisting of instances labeled to be used in the machine learning processes.
- The program should output reports about user performance, instance performance and dataset performance at any time (stop simulation, access reports).
- The program will execute the labeling steps by pre-defined additions such as the dataset and the program's mechanisms.

Non-functional Requirements

- **Performance**

The program's import activities must be done in a short-time period in order to make a strong and powerful software for users. By executing the commanded activities, it should update the users in runtime with ongoing tasks. Logging is another need that should be done to store the trace of any program trial in the back without any performance loss. Moreover, datasets needed to be checked for duplicate entities and redundancy for both import and export operations. Also, the program will provide performance reports that can be accessed at any time to see the performance of users, instances and datasets.

- **Reliability**

Since the data exported by the program will be in use of machine learning models, the program is extensively planned to have various and efficient mechanisms. Users will get the benefit of different approaches with correctly implemented tools.

- **Recoverability**

Users will be able to have the files used by the program accessible and they can revert the changes or export new data files whenever they request from the program to do that.

- **Data Integrity**

Any dataset with the program's format will be available to import to the system. Users will have the privilege of importing any dataset over the program.

- **Interoperability**

The program will apply a tool-based orientation in control of users. Mechanisms will cooperate with both the data and the users so as to provide a diversity among program entities and users.

- **Security**

The datasets and outputs should be executed correctly and prevented loss of any data imported. Additionally, the program must be reachable for only the people having the program itself without any remote connection. Besides, authentication of users will be applied in each login to the system.

- **Maintainability**

If the program faces any problem in use, it should create a log for that and export it to the log file to be handled in next developments. Furthermore, the program should have the design to be extended with both current entities like mechanisms and possible new entities such as machine learning models or deep learning models.

User-interface

The program will be running as a console application. Hence, the scenario will be applied on a console screen. Any operation like data import, data export, labeling processes are needed to be handled over the console if needed.

Not proper implementation of any functional or non-functional requirements would cause dissatisfaction for customers. Therefore, the needs should be revised and applied accordingly.

Domain Model