Marmara University

Department of Computer Science and Engineering

CSE3063 Object Oriented Software Design

Requirement Analysis Document

150116020 - Muhammet Kürşat Açıkgöz
150116024 - Ahmet Elburuz Gürbüz
150117013 - Mehmet Ali Yüksel
150117018 - Ahmet Önkol
150117023 - Anıl Şenay
150117030 - Beyza Aydoğan
150117072 - Bilgehan Geçici

Due Date: 19/12/2020

Requirement Analysis Document

About Project

In machine learning, data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful informative labels to provide context so that a machine learning model can learn from it.[1] 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. Data labeling typically starts by asking humans to make judgements about a given piece of unlabeled data.[1]

In this purpose, we would like to develop a data labeling mechanism to tag the data which are most commonly in the form of images, videos, audio and text assets with proper, meaningful labels. We also would like to develop a user-friendly interface for increasing efficiency and user experience. At the end, the label mechanism can be used by multiple users and produce multiple labeled data for ready to use advanced processes.

Requirement Specification Vision (Purpose)

This requirement specification system document describes the functions and requirements specified for this Data Labeling Mechanism System. The purpose of this project is to provide a user integrated data labeling specification system which will eventually yield a dataset that can be used for the training of Artificial Intelligence models such as Machine Learning.

By observing the set of instances from a given input set, users will be asked to choose descriptive labels for each instance from a label set and assign it to the instance. This document is intended for both the stakeholders and the developers of the system.

Problem Statement

Data labeling system can be used to label customer comments in an e-commerce web site as positive or negative or this system can be used to label news from online newspaper articles as sports, world, economy, politics, etc. This is known as sentiment classification problem. As a more general explanation; sentiment classification is the automated process of identifying opinions in text and labeling them as positive, negative, or neutral, based on the emotion’s customers express within them.[2]

Scope

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.[3] It is also required for a variety of use cases including computer vision, natural language processing, and speech recognition. The goal of the data labeling system is to increase accuracy. For this purpose, users are asked to label instances.

In the first iteration, random labels will be defined for instances by the system itself, but in subsequent iterations, different users can add different tags to an instance by using various types of labeling mechanisms.

In the second iteration, reporting functionality is added for user performance and labeling operation for a particular dataset. The main idea is to collect statistics for users, compare users in the context of a particular dataset or globally, and calculate metrics for instances in the dataset that are labeled with many users. The resulting reports will give us an idea about the quality of the data labeling and the quality of the users.

System Constraints

Will run as a console application on any device that has Java Runtime Environment installed.

Stakeholders

- Murat Can Ganiz: Customer
- Lokman Altın: Customer
- Muhammet Kürşat Açıkgöz
- Ahmet Elburuz Gürbüz
- Mehmet Ali Yüksel
- Ahmet Önkol
- Anıl Şenay
- Beyza Aydoğan
- Bilgehan Geçici

Glossary of Terms

User: Person who labels the instances.

JSON Files: Json is short for JavaScript Object Notation, and is a way to store information in an organized, easy-to-access manner. In our case, it will hold needed information in the input file and will be written to the corresponding output file.

Configuration File: Configuration files provide the parameters and initial settings for the operating system and some computer applications.

Data: It can be any unprocessed fact, value, text, sound or picture that is not being interpreted and analyzed.

Dataset: Collection of data.

Instance: Set of objects to be labeled by every user.

Assignment: Labeled instances are kept as whole for later usage.

Label: A classifying phrase or name applied to an instance to identify given instance.

Labeled Data: Data that comes with a tag; like a name, a type, or a number.

Log File: A log file is a file that keeps a registry of events, processes, messages and communication between various communicating software applications and the operating system.

Labeling Mechanism: Mechanism that tags instances with labels using an algorithm or in a particular way.

Consistency Check Probability: A user's labeling probability of an instance for a second time in a dataset.

Final Label: If an instance is labeled more than once (by the same user or by different users), it is the most frequently used class label.

Completeness Percentage: Shows what percentage of a dataset is labeled by users or what percentage of instances are labeled.

Consistency Percentage: Indicates how consistent a user behaves when labeled instances. We achieve this value by letting users label instances that they previously labeled.

Standard Deviation: Gives information about whether the user's time spent labeling instance is normal or not.

Entropy: Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information.

Proposed System

1. Functional Requirements

- The data labeling system can be used by multiple users simultaneously.
- The system gets user information as a json file.
- An instance can be labeled by one or more users (possibly with different class labels).
- The data labeling system can use multiple label sets on the same data with respect to the input file.
- The labels of the instances given by each user must be stored in the output file.
- The data labeling system should assign labels to instances randomly for iteration 1 of the project.
- The system can work with different rule based labeling mechanisms with respect to input files.
- The system must print logs to console and in a log file.
- User, Dataset and Instance performance metrics are reported in output.
- In the middle of the execution, the program can be terminated.
- The report is created when execution starts.
- After each assignment, the report and output are updated when execution starts even though the program terminates.
- In the case of termination of the program, it will continue from where it has left.

2. Non-Functional Requirements

Usability

- Project should be user-friendly.

Reliability

- Project must keep the user's data safe.

Performance

- Project must read and implement the input file in a short time.
- Labeling mechanism must work and give output in a reasonable time and format.
- Logging mechanism must not slow down the program.

Supportability

- Project must be platform independent.
- Project should be able to run on any Java based platform.

Implementation

- Project will be implemented in Java.
- Input and output files must be in JSON format.

Use Case Model

Case	Given input set is valid.
Actor	System
Description	The system reads the given input files and handles them.
Condition	• The config and input files must be valid, exist and in JSON format.
Flow of Events	• System takes user information from the files and starts creating related objects.

Case	Input set is not valid or does not exist.
Actor	System
Description	Encountering the situation of invalid or non-existence of input.
Condition	• Either config or input files exist or not exist. • If exists then having an invalid format or invalid content.
Flow of Events	• An error handling occurs. • Log of the error is printed to the console and to an error log file.

Case	Instances are labeled successfully.
Actor	RandomBot
Description	For the first iteration, instances are labeled by the random bot mechanism.
Condition	• The label and instance objects must be valid and exist.
Flow of Events	• The system takes label and instance objects. • The RandomBot assigns the label objects randomly to the instance objects with respect to the maximum number of labels per instance.

Case	CurrentDatasetId is set to “id=1”
Actor	System
Description	The system starts labeling the dataset with the given id number.
Condition	• Given dataset must exist and valid.
Flow of Events	• System takes dataset information from the input file set and starts creating related objects and labeling instances.

Case	Execution stopped.
Actor	System User
Description	System user stops the execution of the program.
Condition	• A valid dataset must be in process of labeling.
Flow of Events	• System user stops the program during execution.

Case	A user labels an instance again.
Actor	RandomBot
Description	RandomBot checks an instance it labeled before and labels again with the percentage of consistency check probability.
Condition	• Consistency check probability must be a positive value.
Flow of Events	• RandomBot turns back and checks an instance it labeled before and labels it either with the same label or a new one.

Project Plan & Deadlines

- Iteration 1- December 5
- Iteration 2- December 19
- Iteration 3- January 2

References

- **[1]**https://aws.amazon.com/sagemaker/groundtruth/what-is-data-labeling/
- **[2]**https://monkeylearn.com/blog/sentiment-classification/#:~:text=For%20sentiment%20classification%20problems%2C%20rule,uncomfortable%2C%20frustrated%2C%20etc).
- **[3]**https://whatis.techtarget.com/definition/data-labeling