CSE 3063 OOSD FALL 2020

CSE 3063 JAVA PROJECT: Iterative analysis, design and implementation of a Data Labeling System

Data labeling is the process of assigning one of the several predetermined labels (a.k.a. class labels, categories, tags) to a group of instances (a.k.a. samples, examples, records, documents) via a user interface by human experts. A group of instances are known as a dataset. Labeled dataset is used for the training of Artificial Intelligence models such as Machine Learning models. Please see Doccano (https://github.com/doccano/doccano) as an open source data labeling tool for textual data. You will be developing a similar system but in strictly object oriented manner.

As an example, your data labeling system can be used to label customer comments in a e-commerce web site as Positive or Negative. This is known as sentiment classification problem. In another example the same system can be used to label news from an online newspaper articles as sports, world, economy, politics, other. This is called as news classificaiton problem.

In some datasets an instance may labeled with only a single class label. In other datasets a single instance can be assigned more than one class labels. This will be given as an input along with the dataset and a set of class labels that are available for labeling.

In some datasets the whole text is labeled such as in sentiment classification. In others words or phrases inside an instance (document) can be labeled such as in Named Entity Recognition problem.

Rules and Requirements:

- You will be graded using bell curve. Therefore, all groups are in competition.
- You must use Java
- There will be no databases in this project. You can't use a database system at all. You may store your data in json files.
- You must use Java in an Object Oriented Manner as we introduce in the lectures. However, I want you to go beyond what we teach, do your research on the internet, try to apply best practices and design patterns. You will be grading based on your object oriented modeling.
- You must use iterative and evolutionary development process such as UP, SCRUM, XP. Therefore, you need to start with the core requirements in the first iteration and add more functionality at each subsequent iteration.
- When preparing analysis and design artifacts such as UML diagrams you must follow the instructions exactly and spend more than usual time in analysis and design phases.
- Your business logic must completely be separated from the user interface.
- The main problem is we don't have all the requirements in the beginning. Rest of the requirements will be revealed during next iterations.
- Therefore, you must put special emphasis on the extensibility of your code. This will be evaluated by looking at the change between two iterations. Change will be measured in terms of following metrics. You need to measure and report these at each new iteration.
    - lines of code (LOC) added (previous iterations LOC – this iterations LOC)
    - number of changed classes
    - number of changed methods
    - number of newly added classes
    - number of newly added methods
    - number of newly added attributes
    - ...

  A more detailed document explaining the evaluation criteria will be provided separately.
- It will be a multi-user system.
- A user can label many instances.
- An instance can be labeled by one or more users (possibly with different class labels).
- Your business logic must completely be separated from the user interface.
    - You must implement the data labeling system as a simulation for the first iteration.
    - You will be getting user information as a json file. It will include information such as user names, ids,

    - You will be getting a dataset as a json file. Your system will process the instances in it one by one, and output a json file one by one.
    - The input json file will include meta data such as the set of labels, if the instances can be labeled with a single label or more, if the whole instance is labeled or words or phrases can be labeled, and one or more instances. It will also include user information such as names, ids, user names.
- You will assume that there are several labeling mechanisms.
- A labeling mechanism will take a user, a single instance and a set of class labels as an input, assigns one of these labels to the instance, will return the assigned label or labels associated with the given user. If words or phrases are labeled it will return a set of pairs, each pair including word or phrase and its label(s).
- For the first iteration your labeling mechanism will be random labeling mechanism which will randomly chose one of the labels from the set of labels and assigns it to the instance.
- Your code must support easily pluggable labeling mechanisms. (Hint: A nice place to apply Polymorphism and related design patterns!) For example, I may want you to implement a simple rule based labeling mechanism or a machine learning labeling mechanism, or user interface labeling mechanism in the following iterations
- If we integrate your business logic code to a user interface (think about the user interface labeling mechanism above) it should work with minimal changes. So at the end of the project you will be developing a simple user interface to demonstrate this. You need to report above metrics to prove this.
- The dataset file you will read will include the set of labels, max. number of labels to tag for an instance, and a set of instances. If max. number of labels > 1 instances can have multiple labels. Of course max. number of labels per instance cannot be less than one. Your LabelingMechanism needs to assign one (or more if multilabeled) of these labels to a given instance.
- Your object oriented modeling must support scenarios in which words/terms in a document can also be labeled. It does't mean that you have to implement word or phrase labeling in the first iteration but your design should be extensible that you may be able to add that functionality with min. changes in the following iterations.

After you determined your project groups, you need to

1- open a GitHub account with a meaningful, identifiable name e.g. murat.ganiz or mcganiz

2- open a private GitHub repository using the following naming pattern:
cse3063f20p1[group number]
For group 1, repo name should be: CSE3063F20P1_GRP1
(note: do not open a organization! just open a repo. One of the group members should open the repo and add others as collaborators)

3- add your fellow group members as collaborators to this repo

4- add me (murat.ganiz or mcganiz) and your TA (Lokman) as collaborators to this repo

5- create and empty project with a single java file which prints "hello world" as soon as possible and push it to the repo.

6- Add a readme file, fill it with your group number and list the numbers and names of your group members.

7- Open a folder for each iteration like “iteration1” and store your iteration code in the corresponding folder.

Push (Please see Figure 1 below) as frequently as possible and meaningful! All students in the group must push.

You need to provide following artifacts by the iteration deadlines:

1- A simple Requirement Analysis Document (RAD)
This may include a brief description of the project in a paragraph or two, glossary, list of functional and non-functional requirements, a domain model showing real world objects in your domain along with their simple relations and features. Optionally you may also add a System Sequence Diagram (SSD).
(File name example: CSE3063F20P1_RAD_GRP1 _iteration1.pdf)
This must be pushed into your GitHub repository's corresponding iteration folder by the deadline.

2- Design Class Diagram (DCD)
An UML class diagram showing your domain classes. C
(File name example: CSE3063F20P1_DCD_GRP1_iteration1.pdf)
This must be pushed into your GitHub repository's corresponding iteration folder by the deadline.

3- Design Sequence Diagrams (DSD)
UML Sequence Diagram(s) showing interactions between software objects of your system. (File name example: CSE3063F20P1_DSD_GRP1_iteration1.pdf)
This must be pushed into your GitHub repository's corresponding iteration folder by the deadline.

4- Java Code
This includes classes. Each class must be in a separate .java file named with the name of the class. Please make sure that only the java files, json files and necessary libraries are in the repo. Your repo size should not exceed 10mb. During your demonstrations me or your TA should be able to run your java code easily.
This must be pushed into your GitHub repository's corresponding iteration folder by the deadline.

5- Project Evaluation Metrics
This is a spreadsheet (in excel or open office spreadsheet file) providing your measurements for the metrics given in "CSE3063 Fall2020 Java Project Evaluation Metrics" file. The metrics will be given as columns. Please see the example spreadsheet material under Java Project topic in classroom.

Please make a compressed zip archive of all these 5 artifacts and upload it to Google classroom by the deadline. Best way to do this is to take a snapshot of your GitHub repo and zip it. File size should not exceed 10mb. All students in a group should upload the same file to google classroom
(File name example: CSE3063F20P1_GRP1_iteration1.zip)

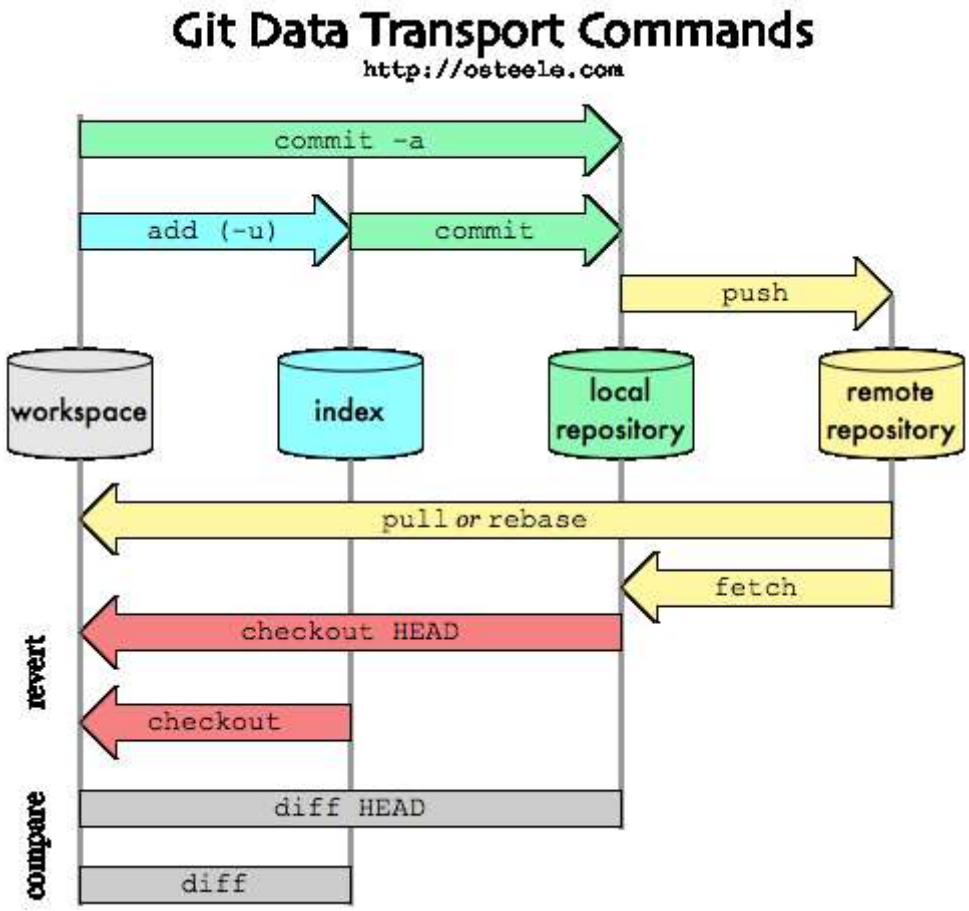

Figure 1: Commit, push, pull in Git