Software Engineering Group Projects - Team Agreements

What is the overall goal of your project?

TheTradeDesk (TTD), a media buying platform, have tasked us with creating a brand sentiment tracking system, by extracting the referenced brand and sentiment from public news articles, for a list of selected top brands.

Specifically, we are developing a fully automated pipeline on AWS to extract brand sentiments from monthly CommonCrawl data dumps, a database for tracking the evolution of sentiment for each brand of interest as time series data, and a dashboard for interactive visualisation.

The pipeline will likely consist of three main stages: text content extraction from webpages, content classification to brands, and sentiment detection. Each of these stages may involve an ML task.

Team composition: why did you choose the team members you did? Do you have a good balance of skills? Have you allocated any specific roles to members?

Every member of the team is communicative, passionate, and competent. Further, we are all interested in related topics, and fit this project perfectly since all of us are taking Deep Learning, and most of us are taking Natural Language Processing. Our team is versatile and can shift roles depending on the needs of the project but have chosen to specialize on certain aspects, which are:

- Mario and Louis will oversee deploying the application to AWS. Both are interested in cloud deployment and scalability and Louis has experience working with AWS on a previous industrial software project.
- Max will focus on article preprocessing and database design, since he has experience with both from previous jobs in start-ups.
- Charlize, Maksym and Matthew will develop the python modules for brand identification and sentiment analysis. Initially these will use a simple or pretrained model but will eventually involve implementing our own from research and the NLP course.
- Maksym will oversee the development the analytics webapp in the final few sprints.

How many hours per week have you each committed to working on the project? Obviously, this may vary a bit from week to week, but it is good to set mutual expectations for how much work you will do.

Each member should spend a minimum of 4 hours per week on the project. It is understandable that coursework deadlines may take priority over the project at points in the term. In the Easter break, members should not have any other major commitments, so they can work as many hours as are necessary to finish the project.

Are you adopting a specific software engineering method or process in your team (e.g. Scrum, XP, Kanban)? What does this mean in practical terms for your team and the way that you will work?

We have decided to adopt the Scrum methodology. There are two main reasons for this:

1. As we will be working closely with an industrial partner, we are in a fortunate position of being able to receive feedback regularly from the product owners. TTD use Scrum to manage their projects; as they are the most experienced team members, it will benefit us to use the same methodology as them so they can advise when needed.
2. As group members who have worked in start-ups before can testify, Scrum is the easiest methodology to follow. Investing in the more rigorous engineering practices of Extreme Programming would be a nice-to-have with more time, and it is likely that we will incorporate or draw from its principles for code review and testing. However, a lightweight methodology like Scrum can get us up and running quickest.

As aforementioned, the project pipeline divides broadly into three interdependent challenges which can be worked on in parallel: content extraction, brand identification, sentiment analysis. A fourth challenge is to scale and deploy software on AWS. These challenges lend themselves well to the sprint format of Scrum – we can consider milestones for each one individually or in combination to guide our workflows for each sprint. The number of milestones, complexity of corresponding tasks and skillsets of team members will allow us to divide our resources as necessary. For example, four members will work on developing a basic python package as a sketch for our automation pipeline for the end of sprint 1. Meanwhile, two members will learn about the technologies needed to scale our program.

As project lead, Matthew will act as Scrum Master. He will monitor team progress and stay on top of roadblocks obstructing development. Jiefei and Christopher at TTD will act as product owners. They will collaborate with the team to ensure product development aligns with their vision, help the Scrum Master prioritize tasks and evaluate progress.

What (if any) technology will you use for a) source control, b) tracking the progress of work in your project, c) communication amongst team members, d) storing and sharing documents (bulleted list)

a. ***GitHub***: all software code will be pushed to GitHub. An organization is in place.
b. ***Trello***: project management will be organised on Trello. Kanban boards are a useful visualisation tool and simple to keep up to date by all team members.
c. ***Zoom, Microsoft Teams, WhatsApp***: TTD's preferred communication method is Zoom. Internally, we will use Teams and WhatsApp to coordinate work and meet.
d. ***GitHub, OneDrive***: GitHub will be our primary location for storing project documents to allow TTD to access materials. Non-essential internal documents can be stored in OneDrive via Teams.

Have you set up a schedule for regular team meetings? When will they be?

- Internal team meeting: once a week: Monday 7-8pm
- Meet with supervisors: once a week in the early stage: Wednesday 2-3pm; In the later stage (e.g. after the first sprint), meet every other week.

Have you set any team rules? For example: Do all members need to attend meetings / working sessions? When is it acceptable to miss one? Have you set any rules around code quality, testing, code review etc.? (brief summary - a few paragraphs)

To make sure that everyone is aware of the overall progress of the project, the team will meet once a week for a brief progress update. If someone cannot attend the weekly meeting, they should provide a brief written progress update instead, which will be discussed by the other team members during the meeting. We will also aim to schedule any ad hoc meetings on weekdays only, between 9am and 7pm.

The team will use PEP-8 when writing python, as it is the general coding standard (if we need to write in other languages, we will agree upon the coding standard beforehand), to make code reviews easier and minimise integration problems. The code should be as functional as possible, ideally less than 14 lines per function (strictly less than 20), and each function should have a docstring describing its function, written using Google's python style guide. The linter Flake8 should be used to ensure that the code meets PEP-8 guidelines.

To ensure the software is robust, all team members will write unit tests for the code they develop. Apart from the first sprint (since development is mostly experimental), members should write these tests before they begin coding (i.e., test-driven development). This will help keep the codebase as simple as possible.

When a member is developing software, they should push all their updates to a separate branch from master. When they have finished and their tests are passing, members will open a pull request to merge into master. Another team member will then review the code and tests and accept the pull request if they are happy with them. This will be the main mechanism for peer testing.

All teams will complete a simple peer-review exercise at various points during the project. You will need to say whether each team member's contribution, matches, exceeds, or falls short of the team's expectations. What will be the criteria by which you judge each other's contribution to the project? (one or two paragraphs)

As Trello is being used to continuously organise and monitor the progress of the project, including timesheets for each member of the team, it will be easy to determine everyone's contribution, based on:

- The user stories they have completed during the sprint.
- The number of hours of work they have completed.
- The quality of their code and tests.

At the end of each sprint, we will have a retrospective to consider everyone's contributions, discuss how well the sprint went and what improvements can be made. These will align with the checkpoint peer assessments and will help decide how we mark each member.

If a member falls short of expectations, we will discuss how they can improve, e.g., by increasing the quality of their code/tests, or by balancing everyone's workload better. If they are consistently not putting in the expected effort, then it will be discussed with the supervisor.

Initial Plans and Risk Analysis

Give a brief overview of your initial project plan. This will likely change as the project develops but set out a plan at the beginning and you can adapt it later as you go through the iterations. (up to one page)

During the spring term, the project will be split into 4 sprints, 2 weeks apart (so they align with the course checkpoints). In each sprint, there will be multiple epics focusing on various parts of the product. This will allow us to avoid bottlenecks in the development process by making these epics as independent as possible. At the end of the sprint, we will present each completed epic to Jiefei and Christopher to get feedback on, which will be its user-acceptance test. Broadly, the epics for each sprint will be:

Sprint 1:

- Develop a basic python package for identifying all the brands in an article and the sentiment for each of them.
- Deploy a test Spark/MLflow application locally and then on AWS, to learn how these technologies are used.

Sprint 2:

- Filter down the articles to be used for identifying brand sentiment, based on e.g.:
    - Popularity of the news site and the number of articles available.
    - The article format on CommonCrawl.
- Port the code of the python package to a new Spark/MLflow application and deploy it on AWS without a database. Set up the pipelines for pushing updates to AWS.
- Research and improve different sections of the python package, in particular:
    - Extractors for parsing articles (based on how the dataset has been reduced).
    - Models and datasets for brand identification.
    - Models and datasets for brand sentiment.

Sprint 3:

- Design and deploy a database for storing the brand sentiment on AWS.
- Design a simple webapp locally that displays time series data for a specific company (where the data is hardcoded into the webapp).
- Continue to research improvements on the python package.

Sprint 4:

- Update the AWS application with the improvements from the previous sprint and connect it up to the new database.
- Deploy the webapp on AWS and connect it up to the database.

By this point, we will have a minimum viable product. Therefore, during the Easter holidays, we will focus primarily on completing the report and making improvements to the app, including features beyond the MVP. This will be one single epic, but with frequent team meetings to ensure all issues are resolved.

We have decided to use Apache Spark and MLflow for our machine learning pipeline, since TTD use these tools the most for their ML applications. Therefore, the system will be easier for them to use after the handover. We will also mainly use Python for this system since it is everyone's preferred language. We have not decided on the database or the front-end language/framework since this will depend on how we implement the rest of the system.

What are the main risks to the success of this project? These might be technical or non-technical. **How will you mitigate these risks?** (up to one page - perhaps a list or table)

Risk	Type	Mitigation strategy
Getting distracted by expansion goals without having an MVP. If we start working on expansion goals without completing the MVP first, it is possible that we do not complete either of them on time. This is because things can take longer time than expected to complete or unexpected problems could arise.	Time Management	E.g. Define MVP deliverables clearly before starting We could follow the agile methodology to mitigate this risk. This consists in gradually implementing small pieces of software that add value to the product and getting feedback on that before moving on.
Misjudging the complexity/time to complete sections of the project	Time Management	Use scrum poker to decide on the complexity of a task, this should avoid the cognitive bias of anchoring. Moreover, we aim to complete the simplest version of the project as quickly as possible, so it will clear how improvements will integrate into the architecture.

Signatures

Alessandra Russo

Louis Manestar

Matthew Collins

Max Greenwood

Mario Lavina Martinez

Charlize Yang

Maksym Tymchenko