# ENG 4000: Sprint Process Review

**Project:** Disaster Tweets - Real or Not: Natural Language Processing

**Date:** 27/11/2020


Team P

| | |
|---|---|
| Binte Zehra | 215624141 |
| Neena Govindhan | 212137600 |
| Jessie Leung | 215985948 |
| Jonas Laya | 214095715 |
| Paul Sison | 214447510 |

# Table of Contents

# List of Figures



# List of Tables

# 1. Sprint 1

## 1.1. Sprint Planning

**Sprint Planning Zoom Meeting - November 1st**
Since the team is relatively new to working together and is looking to plan the first agile sprint, it is difficult to forecast sprint velocity. Therefore, before planning the initial sprint, the team each filled out the following table and reviewed the first ITPmetrics report before the sprint planning meeting. Doing so helps the team identify both individual and team strengths and weaknesses, as well as analyzes the familiarity of each team member with the project topics, such that a sprint plan can be more effectively created based on these metrics.

The confidence levels used to fill out the pre-sprint plan meeting are as follows:

**1** = No experience
**2** = Limited experience
**3** = Fair experience
**4** = Intermediate
**5** = Expert

| Team member | Machine Learning | NLP | Python | Twitter |
|---|---|---|---|---|
| Binte | 2 | 2 | 3 | 2 |
| Neena | 2 | 2 | 3 | 2 |
| Jessie | 2 | 2 | 3 | 3 |
| Jonas | 2 | 1 | 2 | 2 |
| Paul | 2 | 1 | 2 | 1 |

**Table 1:** Self Perceived Expertise

During the sprint planning meeting, the team discussed project management tools that will be used during the sprint. Reviewing the ITPmetrics report, it was clear that managing the varying schedules of the team would be a challenging task since each team member will be balancing busy work, academic and social responsibilities. It will therefore be fundamental during sprint planning processes to set clear product features to be developed during each sprint, and to accurately review the performance of the team at the end of each sprint.

Furthermore, review of Table 1 finds that as a whole, the team required more experience in many of the fundamental project topics. To overcome, this the following tasks Sprint 1:

- Review the general practices for machine learning regarding natural language processing (e.g. Classifiers = Naive Bayes, Decision Trees, SVC, and classification techniques = Bag of Words, Lemmatization, Stop Words) (**Week 1 of Sprint 1**)
- Review Tweet format and metadata, as well as the Kaggle dataset (**Week 1 of Sprint 1**)
- Implement a first prediction model on the Kaggle dataset and review performance metrics of the classification results (**Week 2 of Sprint 1**)

Given the two week timeline allocated for Sprint 1, the feasibility of accomplishing these tasks was discussed during the Sprint Planning meeting. The team is confident that given clear roles and tasks during Sprint 1, the above mentioned Sprint goals can be completed.

The roles and timeline for Sprint 1 as defined by the team are discussed in following sections, a summary of the sprint progress is discussed, and finally the performance and productivity of Sprint 1 is reviewed and a refinement of the process and tasks for future sprints is examined.

## 1.2. Timeline

**Sprint 1 Duration:** November 2nd - November 16th
**Week 1 of Sprint 1:** November 2nd - November 9th
**Week 2 of Sprint 1:** November 9th - November 16th

## 1.3. Sprint Goal(s)

The sprint goals for the first sprint were as follows:

- Review research papers similar to the project to get familiar with Machine Learning tools and techniques
- Review Tweet format and metadata, as well as, the Kaggle dataset.
- Implement the model using different Machine Learning Classifiers: Naive Bayes, Decision Tree, SVC to find the best one, giving reasonable performance
- Furthermore, implement the model using classification techniques (Bag of Words, Lemmatization, Stop Words) to find better results
- Implement a first prediction model on the Kaggle dataset.

## 1.4. Sprint Backlog

| Sprint Backlog | To-Do (For Sprint 2) | In Progress (For Sprint 2) | Done | Work Estimation |
|---|---|---|---|---|
| **PBI #1:** Familiarize with common ML classifiers and metrics | - | Learn about commonly used deep learning models (LSTM) and study implementations using libraries such as keras. | Learn about common classifiers (Naive Bayes, Decision Tree, Random Forest, SVC) and performance metrics (Recall, precision, F-scores). Study sample code of classifier implementations using libraries such as scikit-learn. | Medium |
| **PBI #2:** Familiarize with NLP approaches and techniques | - | - | Learn about NLP techniques (Bag of words, lemmatization, stop words). Study sample code of implementations of these techniques in existing NLP classification models. | Large |
| **PBI #3:** Familiarize with Tweet format and metadata | - | - | Familiarize with Tweet format and metadata such as followers, retweets, likes, hashtags | Small |
| **PBI #4:** Familiarize with Kaggle test dataset | - | - | Look through Kaggle dataset and familiarize with data columns | Small |
| **PBI #5:** Implement first model to generate predictions on Kaggle dataset | Implement a model using deep learning model (e.g. LSTM). | - | Implement an initial prediction model to generate predictions for the 'test' dataset using common classifiers. Implement a model with cross-validation technique to run on a 'train' dataset. | Large |
| **PBI #6:** Review performanc e of first model | - | - | First model predictions submitted to Kaggle test to review accuracy of predictions against actual labels. Review performance of cross-validated recall, precision and f-scores on 'train' dataset. | Medium |

**Table 2:** Sprint 1 Backlog

## 1.5. Tasks

**PBI #1:**

A study of research papers concerning commonly used classifiers shows some classification models that are effectively used for text classification[1]:

- Non-neural network models: SVM, Naive Bayes, Logistic Regression, Decision Trees, Random Forest (many included in scikit-learn library)
- Neural network models: Long Short-Term Memory, Convolution Neural Networks (many included in keras library)

Furthermore, the metrics that are most commonly used to evaluate the performance of the models use methods of cross-validation (e.g. Stratified K-folds as supplied by the scikit-learn library) on test datasets reviewed by the following performance measures, where *tp* is a true positive (Tweet is classified as a valid Tweet where it is truly valid), *fp* is a false positive (Tweet is classified as valid when it is actually invalid), *fn* is false negative (where Tweet is classified as invalid when it is actually valid)[2]:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$Fscore = \frac{2 * precision * recall}{precision + recall}$$

**Figure 1:** Formulae to calculate Fscore

The team came up with research questions, which helped with coming up with product features. It was useful in terms of structuring the features in an efficient manner. The research questions were divided into four categories, descriptive, diagnostic, predictive, and prescriptive.

1 *Oshikawa, R., J. Qian, and W. Y. Wang. "A Survey on Natural Language Processing for Fake News Detection." Cornell University. https://arxiv.org/abs/1811.00770.*

2 *Shu, K., A. Sliva, S. Wang, J. Tang and H. Liu. "Fake News Detection on Social Media: A Data Mining Perspective." Cornell University. https://arxiv.org/pdf/1708.01967.pdf.*

Example research questions from each category:

***Descriptive (observative)***

**Q1:** What is the relation between the type of disaster (earthquake, wildfire, etc.) and the number of fake tweets?

- NULL HYPOTHESIS (1_1): there is no significant difference between the distribution of the number of tweets for different emergencies.
  -- test: ANOVA, Kruskal-wallis, Chi-Square

***Diagnostic***

**Q8:** What factors prevent first responders to trust machine generated solutions?

- How? --- Qualitative research – interview/survey

***Predictive***

**Q11:** What is the Likelihood of significantly different location of fake tweets if the disaster happens over night?

***Prescriptive***

**Q12:** How do we leverage a machine learning model that can identify authentic disaster tweets into a product/service for use of emergency and disaster response organizations?

## PBI #2:

To address the team's general unfamiliarity with machine learning models, the process of building a model for natural language processing was studied. Upon review of multiple research papers on the topic of NLP and building predictive models, a summary of the NLP process first consists of preprocessing the given dataset into extracted features[3]. Data preprocessing involves techniques such as tokenization, lemmatization, and the use of a stop-words list. Exploring the scikit-learn API finds that built-in methods for these techniques are included (e.g. sklearn.feature_extraction.text.CountVectorizer includes parameters for tokenization and stop-words). These are NLP techniques that will likely be useful in the project's implementation of an NLP-based classification model.

## PBI #3:

Familiarizing with Twitter and structure of tweets. Analyzing tweets in terms of its text, vocabulary, keywords used in it, number of times it has been retweeted, followers/followees relationship. After reviewing various research papers, there were some important takeaways that were useful when analyzing Twitter data:

- Methodologies
  - Analyzing Twitter within hours and days after events
  - Examine how rumors and news are propagated in relation to followers/followees
  - Correlation of key terms in various tweets
- Techniques
  - Feature Extraction

[3] *Oshikawa, R., J. Qian, and W. Y. Wang.*

- Model Construction
- Performance Evaluation
- Coding scheme- for categorizing disasters

- Tools
    - Latent Dirichlet Allocation (LDA) topic modelling
    - Tweet2Vec - Tweet semantic similarity tool
    - Non-neural network models: SVM, Naive Bayes, Logistic Regression, Decision Trees, Random Forest
    - Neural network models: Long Short-Term Memory, Convolution Neural Networks

## PBI #4:

The team analyzed the datasets provided by the Kaggle challenge. The test dataset consists of four columns: id, keyword, location, and text. Similarly, the training dataset consists of the same data column, with the addition of a target data column. The classification task prescribed by the Kaggle challenge is a binary classification problem, wherein the target label indicates the validity of a Tweet given the following labels:

**1**: about a real natural disaster
**0**: not about a natural disaster

By observing the structure of the Kaggle dataset, the primary focus of the initial model was decided to use data from the 'text' column to predict the value of the corresponding 'target' column.

## PBI #5:

For the first sprint, building a simple classification model implemented SVC, Decision Tree, Random Forest, Logistic Regression, and Multinomial NB classifiers using the following approach for building a machine learning model. The training dataset is split using a 10 stratified k-fold method, such that nine folds are used for training and one fold used for testing.

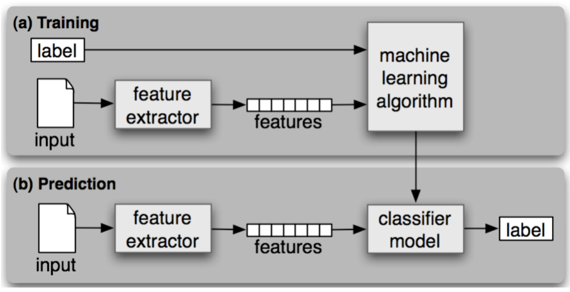


**Figure 2:** Supervised classification process from NLTK[4]

In the feature extraction process, the 'text' data is preprocessed by using techniques such as tokenization, lemmatization and the removal of stop words to compose a Bag of Words in

[4] *"Learn to classify text." NLTK. https://www.nltk.org/book/ch06.html*

the form of a vector. This feature is then trained using a classifier from the scikit-learn library, and outputs prediction labels that are then verified using the ‘target’ column.

**PBI #6:**

| Classifier | Precision | Recall | F1 |
|---|---|---|---|
| **SVC** | | | |
| BOW + Stopwords | 0.70 | 0.70 | 0.68 |
| BOW + Lemmatization + Stopwords | 0.71 | 0.70 | 0.69 |
| **Decision Tree** | | | |
| BOW + Stopwords | 0.57 | 0.58 | 0.57 |
| BOW + Lemmatization + Stopwords | 0.61 | 0.61 | 0.61 |
| **Random Forest** | | | |
| BOW + Stopwords | 0.63 | 0.64 | 0.61 |
| BOW + Lemmatization + Stopwords | 0.67 | 0.66 | 0.64 |
| **Logistic Regression** | | | |
| BOW + Stopwords | 0.65 | 0.66 | 0.65 |
| BOW + Lemmatization + Stopwords | 0.66 | 0.66 | 0.66 |
| **Multinomial NB** | | | |
| BOW + Stopwords | 0.67 | 0.67 | 0.67 |
| BOW + Lemmatization + Stopwords | 0.69 | 0.69 | 0.69 |

**Table 3:** Results of classification techniques for cross-validated training set using 10 stratified k-folds

## 1.6. Project Management Tools

The team used Trello to manage tasks and keep track of product backlog, sprint backlog, tasks that are not started, in progress, and completed.
Jamboard was used during the scrums for reflection and to answer questions ((“What did I do “yesterday” that helped meet the Sprint Goal?”, “What will I do “today” to help meet the Sprint Goal?”, and “Do I see any impediment that prevents me or the team from meeting the Sprint Goal?”).

# 2. Sprint Review

## 2.1. Completeness

The team was able to successfully complete Sprint 1 on time. During week 1 of sprint 1, the team familiarize themselves with different Machine Learning tools and techniques, classifiers (Naive Bayes, Decision Tree, SVC, etc.), and classification techniques (Bag of Words, Lemmatization, Stop words, etc.). Though there is plenty of material available, the team decided to review the basics and begin week 2 of sprint 1. In week 2, the team completed sprint 1 with implementing a first prediction model on the Kaggle dataset and compared results of performance metrics of different classifiers.

### 2.1.1. Burndown Chart

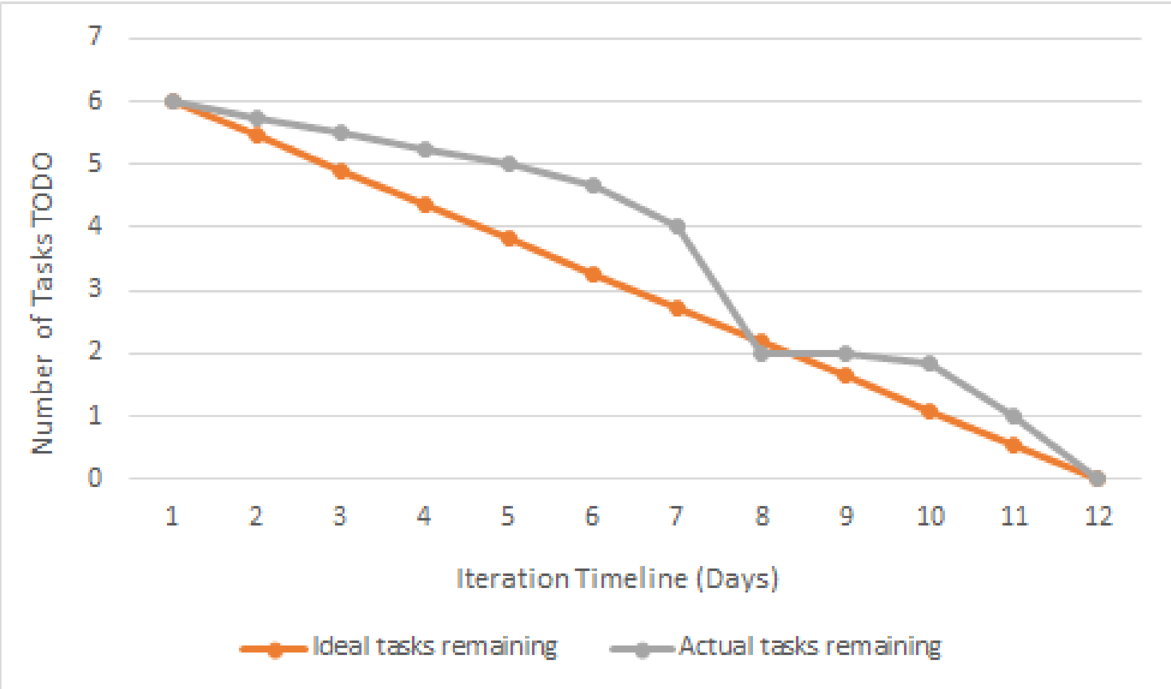


**Figure 3:** Burndown Chart

The goal of this Sprint was mainly to learn about Machine Learning (ML) and Natural Language Processing (NLP) well enough so the team can implement an initial model. Since all the team members are eager to learn about the project topic/domain and everyone wishes to be heavily involved in the development process, the tasks for this Sprint cannot be delegated to only a few members. PBIs 1 and 2 took more time than expected and were the first tasks to be completed. PBIs 3 and 4 were trivial tasks and only took a day to complete.

And due to other course commitments and difficulties in setting up our coding environments, the team was delayed by a few days in implementing the model.

### 2.1.2. Velocity

Having completed 6 PBI tasks in a span of 2 weeks, the team's velocity for this Sprint is 3 PBI tasks per week. A velocity graph will be provided in further reports when the team have completed a few more Sprints.

## 2.2. Roles

For the first sprint the roles were defined as follows:

Jessie Leung: Product Owner/ Development Team
Neena Govindhan: Scrum Master/ Development Team
Binte Zehra: Development Team
Jonas Laya: Development Team
Paul Sison: Development Team

Everyone has a role in the development of the model.

## 2.3. Challenges

The challenges in the first sprint occurred when the team started to code the model. The coding had not been attempted previously, and the team was more focused on planning and research for the problem. However, the knowledge acquired through the research and course work had helped in creating the initial model. With Python, there were some initial hesitations since the team as a whole was a little inexperienced with it, but it was actually much more simple to implement than expected. There was some trouble with managing other courses, so some tasks took longer than expected, but it was good that the sprint backlog was not overloaded so the team managed to get through in time.

# 3. Sprint Retrospective

## 3.1. Happiness Evaluation Metric

To measure the overall team satisfaction levels with Sprint 1, a Happiness graph is used to visually identify the sprint events that team members thought worked effectively towards the product goals. The following scale was used as a metric for the team to measure their 'happiness' levels with each sprint event.

| **Happiness level** | **1** 😫 | **2** ☹️ | **3** 😕 | **4** 🙂 | **5** 😁 |
|---|---|---|---|---|---|
| Criteria | Unhappy with effectiveness of sprint activity, very demotivated | Slightly demotivated, feel that more could have been done with sprint activity | Sprint activity was fair, neutral motivation levels | Generally happy with sprint event, motivation and momentum levels fair | Happy with sprint event, feeling motivated and happy with sprint momentum |

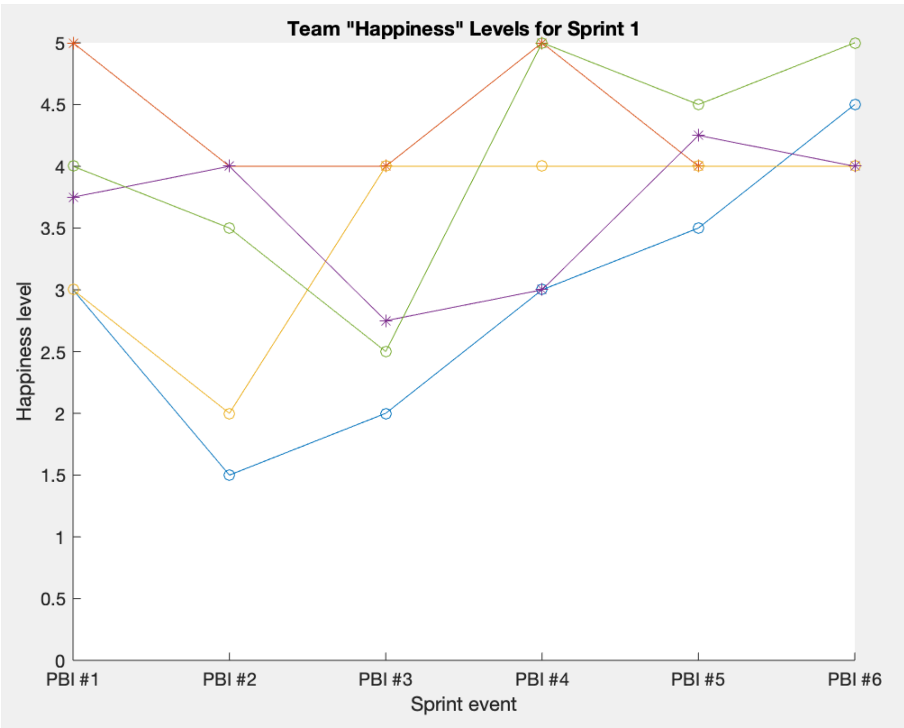


**Figure 4:** Happiness Graph

During the Scrum Master-led Sprint Retrospective, the effectiveness of Sprint 1 was discussed with members of the team. Sprint activities that had generally lower happiness ratings were discussed to address what aspects of the event could have been improved.

Similarly, generally positively rated events were discussed to identify why team members felt that that activity significantly aided the progression of the project. The findings of the retrospective discussions are detailed in the following sections.

## 3.2. Process Related Issues

A discussion of the collaboration process during Sprint 1 resulted in the following actionable procedure changes such that following sprints can be streamlined and more effective.

| Sprint Event | What Went Wrong | What Went Well | Actionable Procedure Objectives |
|---|---|---|---|
| PBI #2 | Access to mass amounts of academic papers felt overwhelming at times, many topics to learn since the team is relatively new to ML. Difficulty understanding how to bridge ML/NLP learning with code implementation. | Reading through many academic papers, found techniques and classifiers that were commonly used throughout most papers. Clear that these were topics that the team should focus on. Helped clarify NLP techniques conceptually. Research papers split up amongst team members and summarized in PowerPoint documents presented amongst team members. | Continue to divide learning workload amongst team members, generate documentation and make accessible amongst members of the team to facilitate overall team learning (documentation available on MS teams, presentation of academic learning such that the entire team benefits). |
| PBI #5 | | Team began learning once the process of implementation began. Concepts that were less clear in previous sprint events (NLP & ML concepts) became easier to understand once in the process of implementing. | Pair academic learning with implementation activities to validate conceptual learning. |
| PBI #6 | | Overall sentiment of progress since results (cross validation scores, testing with Kaggle prediction submissions) helped validate that learning done in previous sprint events was implemented to a satisfactory level. | Continue to generate testable implementations (via cross-validation, Kaggle submissions) and improve on scores from previous models. |

**Table 4:** Process Related Issues

# 4. ITP Metrics Reflection

The ITP Metrics Peer evaluation and Conflict evaluation were completed for Gate 2.

## 4.1. Last Peer Review Reflection

As a team at first we did not know each other very well, but we were able to quickly come together as a group. We are all fairly new to machine learning as a topic and that is our motivation to work on the project. This did initially pose a problem as to where to start since we all were fairly new to the topic. Under the guidance of our supervisor, feedback from peers, our own research and helping each other, we had a better understanding of the problem at hand and have figured out a direction we want to go with the project.

As per our ITP Metrics, Team Dynamic report, as a whole we work well as a team. In terms of Communicate, we have been messaging each other on Whatsapp and having at least one Zoom meeting a week to go over what needs to be done for that week. We keep all of the different files on Microsoft Teams, so that everyone has access to it. To improve our communication we will be using Trello boards more to keep track of the tasks at hand and who will be responsible for it in our scrums.

In terms of Adapt, we are doing well to coordinate between each other to figure out what to do and monitoring what has been done. The thing we need to improve on as a team is to work on our time management. As discussed earlier tools such as Trello, Stormboard, etc. and starting the “daily” scrum meetings will help to effectively plan out our time and pace our goal progression more efficiently.

In the Relate section, as a team we have been good at contributing to work equally and have had a positive environment where no conflicts have arisen as such. To improve on Relate, we should be more prepared in our meeting to be able to have more healthy, fact-driven conflicts, or bring about different perspectives. The objectives of our meeting should be stated ahead of time, so discussion will also be efficiently done. This will be the ScrumMasters role to make sure that this is stated before the scrum meetings.

In Educate, since this is a new topic for all of us, this has been a very important part process so far. We have each done some literature research on the problem to understand how we can expand on the Kaggle project given, and have come together to teach each other the important aspects of the literature research done. We have also done some online tutorials to understand machine learning and natural language processing. We also feel comfortable sharing that information with each other.