UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Aplicación del Aprendizaje Semisupervisado en el descubrimiento de ataques a Sistemas de Recomendación Documentación Técnica

Presentado por Patricia Hernando Fernández
en Universidad de Burgos — 12 de noviembre de 2022
Tutor: Álvar Arnaiz González

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

Planificación por *sprints*

Sprint 1:

- ***Planning meeting***

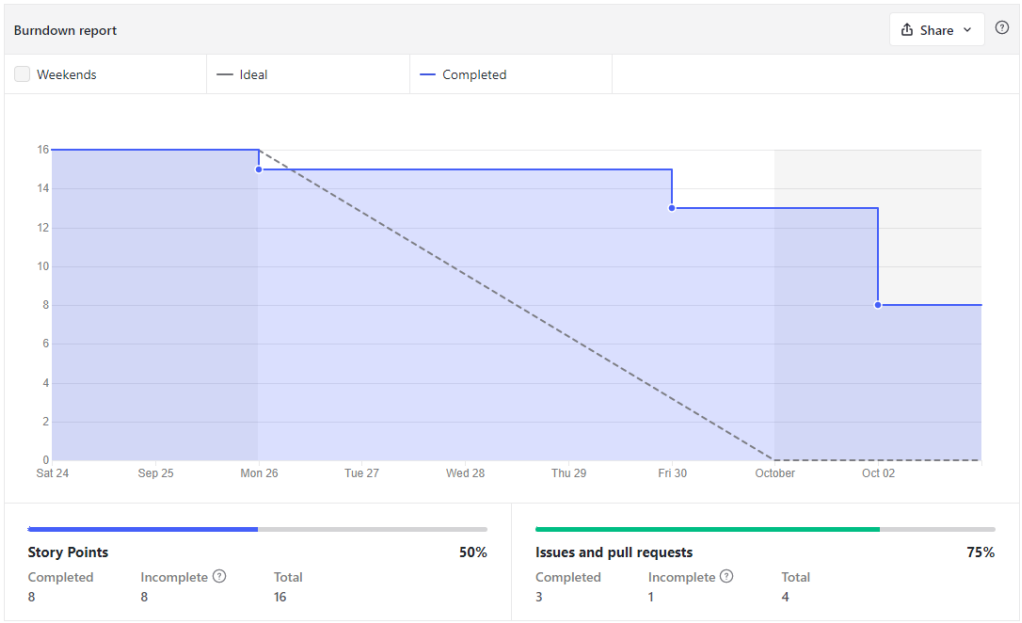
  Durante la reunión se marcaron los siguientes objetivos:

  1. Configuración básica: incluyendo la creación del repositorio, la correcta instalación de ZenHub, la creación de entornos virtuales (miniconda, SKLearn, etc.) y la familiarización con conceptos *scrum*: *milestones, sprints, epics*, etc.
  2. Memoria: comienzo de la redacción incluyendo las secciones de introducción, conceptos teóricos (aprendizaje automático) y trabajo relacionado.
  3. Investigación: búsqueda del código SSADR-CoF y de las bases de datos utilizadas en el paper.
  4. Lectura de papers: Engelen & Hoos [5], García, Triguero & Herrera [3], y Zhou & Duan [6].

- **Marcas temporales** El *sprint* se desarrolló entre el 24 de septiembre de 2022 y el 2 de octubre del 2022.

- ***Burndown Report***

Figura A.1: *Burndown Report Sprint 01*



Como se puede comprobar, no todos los objetivos marcados fueron cumplidos: la estimación del tiempo fue demasiado optimista, además de no contar con el tiempo requerido en solucionar problemas técnicos (LaTeX). Se dejó para próximos sprints la lectura del último paper.

- ***Sprint review meeting*** Durante la reunión se fijaron ciertas correcciones en la memoria (mejorar referencias bibliográficas y la sección de «Trabajos relacionados»), además de la necesidad de introducir una sección teórica de ataques a los sistemas de recomendación.

Sprint 2:

- ***Planning meeting*** Objetivos del siguiente Sprint:

  1. Configuración: debido a la gran cantidad de tiempo invertida en solucionar errores de compilación en LaTeX, se decidió migrar el proyecto a una nueva instalación basada en Debian.
  2. Correcciones: aspectos estilísticos y completar información.
  3. Lectura: Mingdan y Qingshan [2] con el objetivo de introducir una sección teórica de ataques.

4. Memoria: redacción completa de los modelos de ataque en los aspectos teóricos.

- **Marcas temporales** El *sprint* se desarrolló entre el 3 de octubre de 2022 y el 18 de octubre del 2022.
- ***Burndown Report***

Figura A.2: *Burndown Report Sprint 02*

En este *sprint* sí se cumplió con los objetivos marcados. Sin embargo, la estimación de tiempo tampoco fue la adecuada, requiriendo más de lo previsto.

- ***Sprint review meeting*** Durante la reunión se resolvieron dudas acerca de bibliografía, referencias y trabajo previo. Además, se acordó empezar a programar, definiendo así los *issues* desarrollados en el siguiente *sprint*.

Sprint 3:

- ***Planning meeting***

  Durante esta reunión, se decidió empezar a programar el *co-forest*. Para ello, se definieron los siguientes pasos:

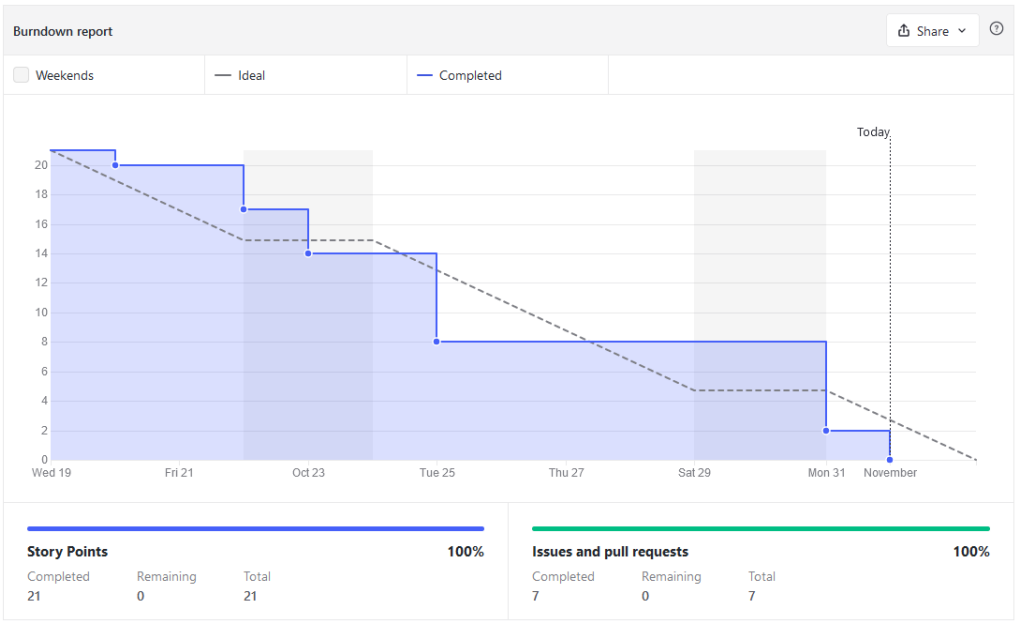
1. Librerías: se acordó aprender a utilizar las librerías más comunes en el *data science*. Entre ellas: MatplotLib y SKLearn. Además, se requirió la correcta configuración del entorno virtual, haciendo que el tiempo dedicado al *issue* fuese mayor de lo estimado (problemas en el `PATH` y con las dependencias).
2. *SKLearn*: aprovechando la correcta documentación de la librería, se decidió repasar los conceptos teóricos básicos, además del manejo de la «interfaz» (métodos comunes). Entre ellos:
   - *Decision trees*
   - *Self training*
   - *Random Forest*
3. Lectura: se concertó la relectura del artículo de Zhou [6] con la intención de comprender el algoritmo y del *paper* «original» del *co-forest* [1]. Durante el proceso de programación, además, se encontró la tesis de Van Engelen [4] y se añadió al conjunto.
4. Documentación: se acordó la corrección de los errores previamente señalados y la inclusión del *sprint* en los anexos.
5. Programación del *Co-Forest*: se programó el pseudocódigo ilustrado en la Tesis de Van Engelen [4], que es muy similar al original [1] pero con algunas diferencias. Inicialmente se intentó usar el *Random Forest* de *SKLearn*, pero se descartó la idea debido a la poca versatilidad que se ofrecía para manejar los *concomitant ensembles*. Se han de corregir ciertos factores, pero se pospondrá hasta la correcta discusión con el tutor.

- **Marcas temporales** El *sprint* se desarrolló entre el 19 de octubre de 2022 y el 2 de noviembre del 2022.
- ***Burndown Report***

  En este *sprint* se cumplió con los objetivos marcados. Nuevamente, la estimación del tiempo fue inferior a la real (se pensaba que se podría depender más de librerías existentes de lo que se pudo en realidad), calculándose un total de aproximadamente 25 horas reales.
- ***Sprint review meeting***

  Durante la revisión del *sprint*, se llegó a la conclusión de que el pseudocódigo podía ser mejor implementado aprovechando ciertas librerías de *Python*. Se comentó cómo mejorar complejidades espaciales y reducir el código. Se fijaron objetivos para las próximas semanas.

Figura A.3: *Burndown Report Sprint 03*



Sprint 4:

- ***Planning meeting***

  Durante la reunión se acordaron los siguientes objetivos:

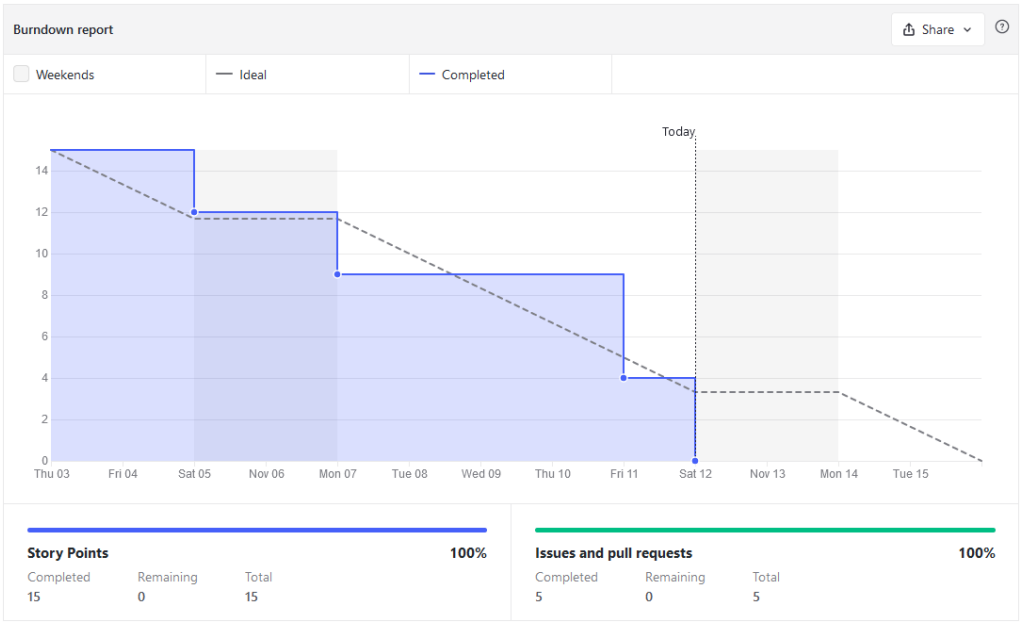
  1. Reimplementación del código: se acordó volver a programar el *co-forest*, esta vez implementando una versión más «*pythoniana*» con el fin de mejorar la complejidad espacial y facilitar la lectura.
  2. Curso de Numpy: se decidió que sería interesante la realización de un curso para aprender a utilizar la librería y aplicarla al código.
  3. Curso de Pandas: aprovechando la relación con el punto anterior, se acordó completar también un curso de esta librería.
  4. Memoria: corregir aspectos anteriores e incluir toda la teoría relacionada con el *co-forest*.

- **Marcas temporales** El *sprint* se desarrolló entre el 3 de noviembre de 2022 y el 15 de noviembre del 2022.

- ***Burndown Report***

  En este *sprint* se completaron los objetivos, aunque quedaron pendientes ciertos aspectos a comentar respecto al código. Destacar que,

Figura A.4: *Burndown Report Sprint 04*



debido a que se terminaron antes de lo planeado los *issues* planificados, se aprovechó para modificar ciertos aspectos pendientes relacionados con la memoria y para probar correctamente el código. Esto hizo que el tiempo real dedicado haya sido ligeramente superior al estimado (más tiempo de documentación).

- ***Sprint review meeting***

Sprint N:

- ***Planning meeting***
- **Marcas temporales**
- ***Burndown Report***
- ***Sprint review meeting***

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

Una muestra de cómo podría ser una tabla de casos de uso:

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

CU-1	**Ejemplo de caso de uso**
Versión	1.0
Autor	Alumno
Requisitos asociados	RF-xx, RF-xx
Descripción	La descripción del CU
Precondición	Precondiciones (podría haber más de una)
Acciones	1. Pasos del CU 2. Pasos del CU (añadir tantos como sean necesarios)
Postcondición	Postcondiciones (podría haber más de una)
Excepciones	Excepciones
Importancia	Alta o Media o Baja...

Tabla B.1: CU-1 Nombre del caso de uso.

Apéndice C

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

Apéndice E

Documentación de usuario

E.1. Introducción

E.2. Requisitos de usuarios

E.3. Instalación

E.4. Manual del usuario

Bibliografía

[1] Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 37(6):1088–1098, 2007.

[2] Si Mingdan and Qingshan Li. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*, 53, 01 2018.

[3] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, 42, 02 2015.

[4] Jesper Van Engelen and Holger Hoos. Semi-supervised ensemble learning. master's thesis., 07 2018.

[5] Jesper Van Engelen and Holger Hoos. A survey on semi-supervised learning. *Machine Learning*, 109, 02 2020.

[6] Quanqiang Zhou and Liangliang Duan. Semi-supervised recommendation attack detection based on co-forest. *Comput. Secur.*, 109(C), oct 2021.