75.99 Trabajo profesional

Facultad de ingeniería - UBA

Comparación de lenguajes de programación para la práctica de la ciencia de datos

Fecha de presentación: 06/10/2023

1. Títulos

Comparación de lenguajes de programación para la práctica de la ciencia de datos. Análisis de características salientes de cada uno de los lenguajes y sus ecosistemas. Diseño y desarrollo de benchmarks para definir ventajas, desventajas y contextos de uso.

2. Integrantes

Alumnos: Ayelén Bibiloni Lombardi, Alan Rinaldi, Cristian Salas, Juan Patricio Marshall.
Emails: abibiloni@fi.uba.ar, arinaldi@fi.uba.ar, csalas@fi.uba.ar, jmarshall@fi.uba.ar

Tutora: Rosa Wachenchauzer.
Co tutor: Gaston Martinez.
Emails: rositaw@fi.uba.ar, gaston.martinez.90@gmail.com

3. Resumen

Se llevará a cabo un análisis del potencial de diversos lenguajes de programación para su utilización en el ámbito de la ciencia de datos. Para esto se realizará una evaluación comparativa buscando abarcar distintos aspectos y posibles casos de uso que pertenecen al objeto de estudio. Los lenguajes a su vez serán seleccionados considerando el estado actual del mercado y las futuras tendencias que se puedan avistar.

El proyecto se dividirá en dos etapas: la primera con foco en tareas de análisis de datos, en las cuales se puede evaluar el impacto de las implementaciones en el manejo de recursos del ecosistema (lenguaje de programación + biblioteca). En la segunda etapa, se centrará en tareas de aprendizaje automático, las cuales son computacionalmente intensivas, altamente paralelizables y sumamente restrictivas.
Para el desarrollo se seleccionarán y/o confeccionarán diversos conjuntos de datos para poder evaluar el comportamiento de distintos lenguajes sobre cada uno de ellos, haciendo hincapié en las diferentes características que cada uno tiene y cómo las propiedades de los conjuntos de datos influyen en la performance.

4. Palabras clave

Lenguajes de programación, bibliotecas, marco de trabajo, análisis de datos, aprendizaje automático, evaluación comparativa, conjunto de datos, tipos de datos, manejo de recursos, métricas, Python, Scala, R, Spark, Pandas, Koalas, Julia

5. Abstract

We will carry out an analysis of the potential of various programming languages for their use in the field of data science. For this, multiple benchmarks will be developed, seeking to cover different aspects and available use cases that belong to the object of study. The languages will be selected considering the present state of the market and future trends that could arise.
The project will be split into two stages: the first one will focus on data analysis, finding out how each language and tool manages resources and load management. Later in the second stage, we will focus

on machine learning algorithms where we will analyze the performance of each algorithm, the resources used and the accuracy each of the languages/frameworks shows.
For this development, various datasets will be selected in a way to allow us to evaluate the behavior of each language, emphasizing the different characteristics each one presents. Likewise, an analysis on the impact the size of the dataset has on each tool/language will be made.

6. Keywords

Programming Languages, libraries, framework, data analysis, machine learning, benchmarking, dataset, data types, resource management, metrics, Python, Scala, R, Spark, Pandas, Koalas, Julia

7. Introducción

El propósito de este trabajo profesional es generar una evaluación comparativa de los lenguajes de programación más utilizados en la actualidad para la ciencia de datos, evaluar las diferentes herramientas con las que cada uno de estos cuenta y poder facilitar la elección de los mismos en base a casos de usos particulares. A su vez, todas las herramientas desarrolladas para lograr este objetivo van a poder ser útiles para futuros proyectos que extiendan el alcance de este proyecto.Esto presentará una posible solución al problema que se encuentra hoy en el mercado a la hora de emplear un estándar de comparación entre las herramientas disponibles para el ámbito de la ciencia de datos.

8. Estado del Arte

La ciencia de datos data desde hace aproximadamente medio siglo y actualmente desempeña un papel fundamental en casi todos los campos. Las necesidades de poder procesar, analizar y visualizar grandes volúmenes de información cada vez se encuentran más presentes dentro de cualquier empresa para poder observar, comparar y tomar decisiones de la forma más acertada posible.
Actualmente existen cada vez más herramientas y estudios para la práctica de la ciencia de datos pero entendemos que aún no existe una estandarización concretamente definida para determinar las fortalezas y debilidades de los lenguajes de programación utilizados para esto[1][2][3]. Entendemos también que existe una fuerte dificultad para lograrlo, ya que los avances en nuevas tecnologías y las herramientas que se generan diariamente hacen que los análisis queden obsoletos en poco tiempo [4].
Debido a esta realidad, nuestro objetivo es poder generar un análisis exhaustivo de las últimas herramientas del mercado para que la comunidad pueda contar con la información actualizada y a la vez dejar un estándar de comparación para futuras adaptaciones a las nuevas tecnologías que el mercado traiga.

9. Problema detectado y/o faltante

En la actualidad no existe un benchmark estándar que nos permita determinar qué herramienta utilizar para la resolución de tareas específicas a la ciencia de datos. Esto ocurre debido a que su utilidad puede variar significativamente en función a la naturaleza de los datos y los objetivos del análisis. [5]

10. Solución propuesta

Se propone una evaluación comparativa sobre distintos lenguajes de programación, utilizando múltiples set de datos variando su tamaño y tipo. La solución se compondrá de dos etapas: el análisis de datos y el aprendizaje automático.

Análisis de datos:
En esta primera etapa se evaluará la capacidad de cada lenguaje de programación para la carga de datos, su manipulación y visualización. Se analizará el comportamiento (manejo de memoria, tiempos, etc) de cada lenguaje y framework utilizado en base al tipo de dato predominante en el set de datos y al tamaño de dicho set.

Aprendizaje automático:
En esta segunda etapa se centrará en el comportamiento de los distintos lenguajes y bibliotecas al momento de resolver problemas propios del ámbito de machine learning. Analizaremos cómo los diferentes lenguajes de programación facilitan la implementación de dichos algoritmos, su precisión, rendimiento y escalabilidad.

11. Experimentación y/o validación

Para la experimentación se tomarán entre 3 y 5 lenguajes de programación con una o más herramientas y/o frameworks para ciencia de datos y la generación de los benchmarks. Los sets de datos a utilizar serán variados en base a sus características, la naturaleza de los datos y las dimensiones de los mismos, generando así una una comparativa amplia entre las distintas herramientas en el foco de estudio. Del mismo modo, se generarán los análisis correspondientes para los algoritmos utilizados para el aprendizaje automático.
Para la validación se ejecutarán múltiples simulaciones para los casos ya mencionados de modo que el resultado sea el más preciso posible. Se generarán visualizaciones acordes a estas comparaciones para que sea fácil de entender los puntos fuertes y débiles de cada lenguaje/framework en cada escenario.

12. Plan de actividades

Metodología de Trabajo

Para este proyecto de análisis de lenguajes orientados a la ciencia de datos, se opta por una variación de la metodología Kanban [6], englobando en sprints ciertos subconjuntos de tareas con un mismo tópico. Las tareas se definirán en conjunto, considerando que cada lenguaje/framework contendrá el mismo set de subtareas. Cada uno de los integrantes podrá tomar cualquiera de las tareas se encuentran en el backlog, ya que estas terminan siendo independientes entre sí. Adicionalmente, el equipo mantendrá reuniones periódicas dos veces por semana para trabajar sincrónicamente y tener un punto de encuentro para la sincronización del equipo o el planteo de bloqueantes o riesgos que en el trabajo pudieran surgir. Estos encuentros serán de al menos dos horas cada uno.
Los encuentros con la tutora y el co-tutor ocurrirán quincenalmente con una duración de al menos una hora para notificar avances, bloqueos o inconvenientes.

Entregables

Durante el desarrollo del proyecto se generarán dos entregables, uno para cada fin de cuatrimestre

1. Benchmark para análisis de datos
2. Benchmark para aprendizaje estadístico

En ambos casos se entregará la documentación de cada análisis realizado, por lenguaje y set de datos, con sus respectivos resultados. Adicionalmente se entregará el código fuente del testbench.

Hitos de avance

Dada la planificación del cronograma de trabajo definido consideramos cada sprint como un hito de avance en sí mismo. Como se menciona, cada uno de los sprints está pensado para entregar un valor genuino al objetivo final de cada módulo.

Gestión de Riesgos

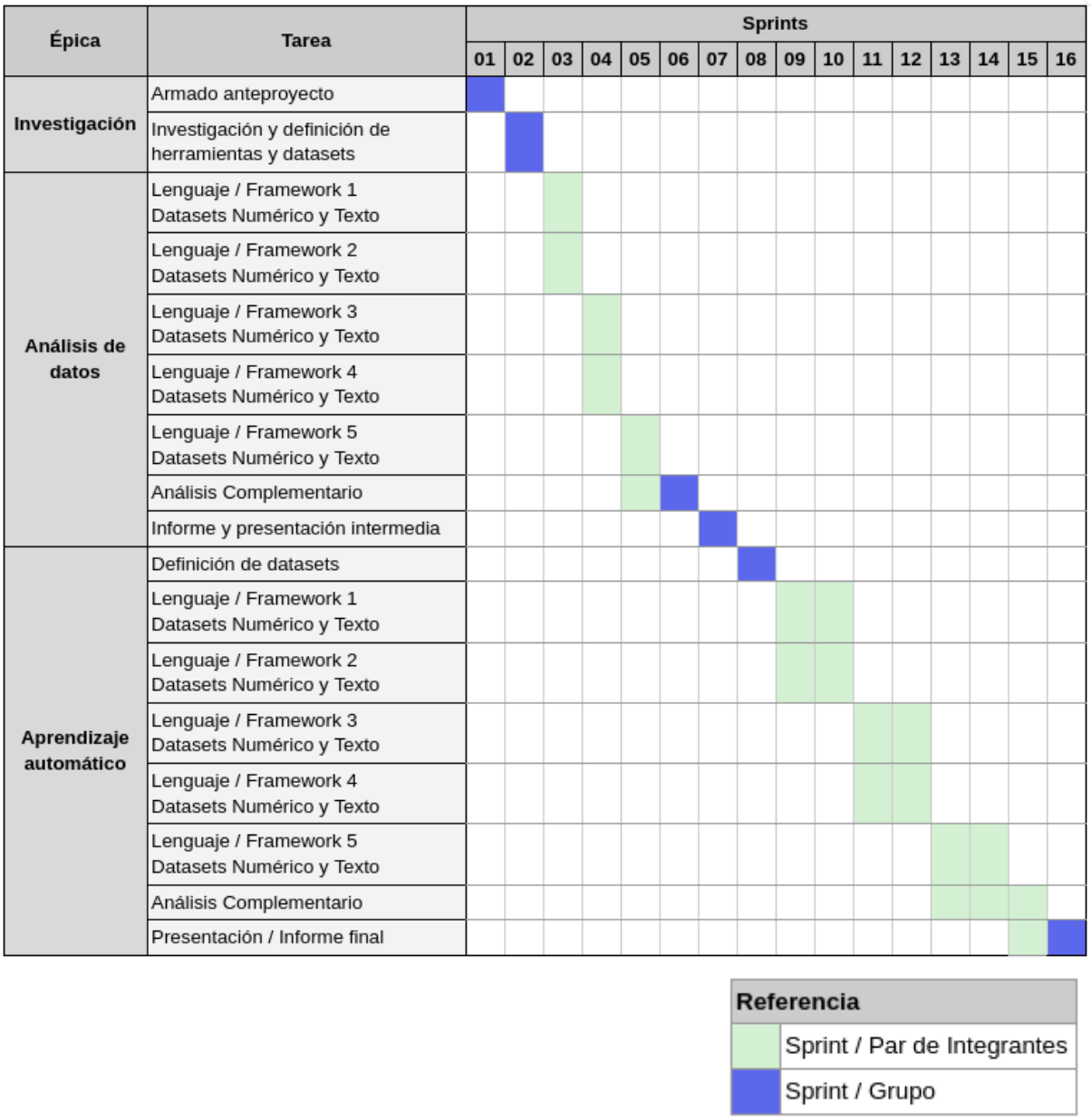
La identificación y gestión de riesgos se llevará a cabo de manera continua a medida que se avance en el proyecto. Se reserva un espacio dentro de cada sprint para desarrollar planes de mitigación.

Estimación y cronograma de entregables

Para cumplir con las normativas de la asignatura se dividió el proyecto en 16 sprints de 2 semanas cada uno, de los cuales se considera una disposición semanal de cada uno de los integrantes de 12hs, generando así, en cada hito un total de 24 horas-persona.

Como se mencionó anteriormente, el trabajo se divide en dos grandes grupos: Análisis de datos y aprendizaje automático. Se estima el desarrollo de cada módulo en aproximadamente 8 sprints (un cuatrimestre), pudiendo realizar una entrega parcial a mitad del cuatrimestre sobre el primer entregable.

Se subdividieron los módulos en tareas para poder realizar entregas de valor a lo largo de cada sprint. A continuación se ilustra una gráfica de las tareas programadas por sprint y las asignaciones que tiene cada una. Estas asignaciones pueden estar divididas en dos, por par de estudiantes o por grupo completo, teniendo en cuenta la dificultad de las tareas o la necesidad del trabajo en conjunto de cada una de ellas.



13. Referencias

[1] Alex Watson, Deepigha Shree Vittal Babu, and Suprio Ray: Sanzu: A Data Science Benchmark (2017)
[2] Linh-Nga Tran: Improving PySpark Performance with Cross-Language Optimization(2021)
[3] Jeyan Thiyagalingam, Mallikarjun Shankar, Geoffrey Fox and Tony Hey: Scientific machine learning benchmarks (2022)
[4] Angelo Mozzillo: Maximizing Efficiency in Existing Data Preparation Pipelines (2023)
[5] Todor Ivanov, Tilmann Rabl, Meikel Poess, Anna Queralt, John Poelman, Nicolas Poggi and Jeffrey Buell: Big data benchmark compendium (2016)
[6] Muhammad Ovais Ahmad, Jouni Markkula and Markku Oivo: Kanban in Software Development: A Systematic Literature Review (2013)