



**Tecnológico
de Monterrey**

8.2 Avance de proyecto 3: Sistema de Recomendación

Análisis de grandes volúmenes de datos

TC4034 grupo 10 Equipo 26

Luis Arturo Dan Fong | A01650672

Eduardo Rodríguez Ramírez | A01794892

Felipe Enrique Vázquez Ruiz | A01638116

09 de Junio de 2024

1. **Revisión** del alcance y objetivos del proyecto.

Después de la revisión de alcances y objetivos llegamos a la conclusión de modificar el cronograma para incluir las últimas actividades, el cronograma resultante es el siguiente:

Etapas	Duración	Inicio	Fin
Investigación	1 día	Mayo 6	Mayo 6
Preparación y preprocesamiento	1 día	Mayo 7	Mayo 7
Análisis exploratorio	2 días	Mayo 8	Mayo 9
Implementación de algoritmo básico	4 días	Mayo 10	Mayo 13
Implementación de algoritmo avanzado	4 días	Mayo 22	Mayo 26
Comparativa de modelos y revisión de avance.	2 días	Junio 8	Junio 9

En cuanto al avance en objetivos, se han cumplido con las fechas establecidas.

2. Realiza la **comparación** de diferentes algoritmos de recomendación de las actividades 4.2 y 6.2, en términos de rendimiento y escalabilidad.

Después de analizar y discutir los resultados, en general concluimos que para este sistema de recomendación, lo mejor sería optar por el algoritmo ALS debido a su escalabilidad y simplicidad de implementación. Sin embargo, si se requiere un modelo más avanzado que pueda capturar relaciones complejas, se podría considerar usar embeddings, pero a la luz de los resultados obtenidos ALS cumple con el objetivo, a continuación se presenta una tabla comparativa de ambos modelos.

Algoritmo	Precision@10	MAP@10	NDCG@10	Escalabilidad	Ventajas	Desventajas
ALS	0.36	0.22	0.24	Altamente escalable ya que es soportado en frameworks como spark.	Escalable a grandes conjuntos de datos	Requiere ajuste de hiperparámetros
					Fácil de implementar en Spark MLlib	Puede sufrir de sparsity (datos escasos)
					Rápido para conjuntos de datos grandes	No captura relaciones no lineales complejas
Embeddings	0.22	0.17	0.22	Moderada ya que requiere más recursos y tiempo de entrenamiento	Captura relaciones complejas entre usuarios y items	Requiere más recursos computacionales
					Puede integrarse con modelos de deep learning	Necesita más tiempo de entrenamiento
					Más flexible para diferentes tipos de datos	Puede sobreajustar si no se regulariza