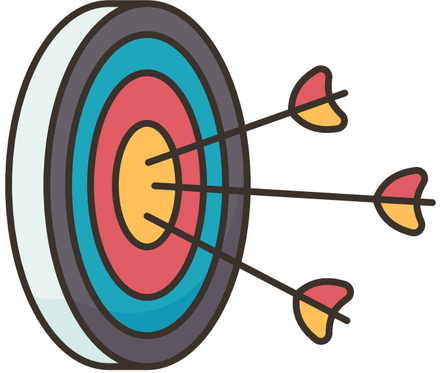
IMPLEMENTACIÓN DE ALGORITMOS DE MACHINE LEARNING DISTRIBUIDOS

Objetivo general: Desarrollar e implementar algoritmos de machine learning distribuidos utilizando técnicas de paralelismo y herramientas como Dask y PySpark, optimizando su rendimiento y escalabilidad.

OBJETIVOS ESPECÍFICOS



OBJETIVO 1

Desarrollar un conjunto de algoritmos de machine learning que incluyan enfoques de ensamblado, visión computacional y clustering.

OBJETIVO 2

Preparar conjuntos de datos adecuados para el entrenamiento y evaluación de los algoritmos.

OBJETIVO 3

Implementar técnicas de paralelismo y distribución para los algoritmos de machine learning desarrollados.

OBJETIVO 4

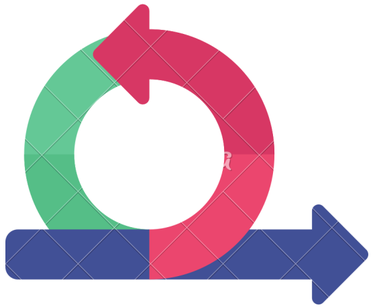
Utilizar Dask y PySpark para manejar grandes volúmenes de datos y mejorar la escalabilidad de los algoritmos.

OBJETIVO 5

Optimizar los algoritmos de machine learning para mejorar su rendimiento y eficiencia.

OBJETIVO 6

Preparar y presentar los resultados del proyecto..



METODOLOGÍA

ENFOQUE ÁGIL Y ESTRUCTURA DE SPRINTS

- Se desarrollo 3 Sprint de aproximadamente de 4 semanas
- Seguimiento del progreso y resolución de impedimentos
- **Revisión de Sprint**
  - Evaluación del trabajo completado al final de cada sprint
- **Retrospectiva de Sprint**
  - Reflexión sobre lo que funcionó bien y lo que se puede mejorar.

HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

- **Control de Versiones:** GitHub
- **Manejo de datos y mejorar escalabilidad:** Dask y PySpark
- **Identificación de cuello de botella:** cProfile
- **Optimización y evaluación de algoritmos:** Grid search, random search y Monitoreo.
- **Desarrollo y Pruebas:** Visual Studio Code o Google Colab

SPRINT 1

OBJETIVOS:

- **Desarrollar un conjunto de algoritmos** de machine learning que incluyan enfoques de ensamblado, visión computacional y clustering.
- **Preparar conjuntos de datos** adecuados para el entrenamiento y evaluación de los algoritmos.

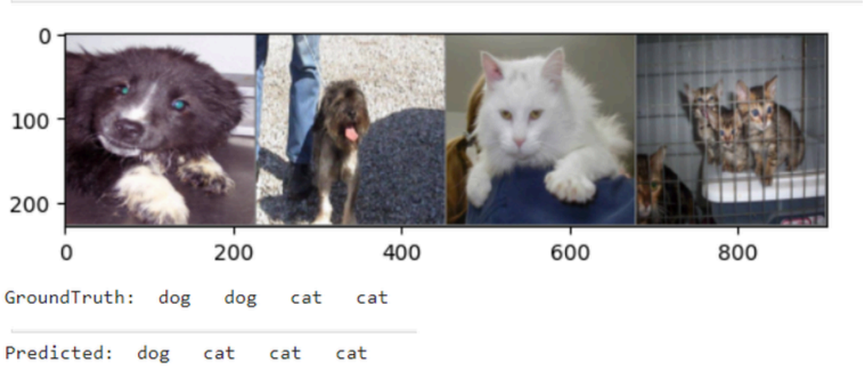
LOGROS:

- Se logro desarrollar un conjuntos de algoritmos
- Preparación de conjuntos de datos.

Algoritmos de ensamble como Random Forest, Gradient Boosting, y XGBoost.

Algoritmos de clustering como K-means, DBSCAN y Agglomerative Clustering.

Algoritmos de visión computacional como CNN con PyTorch.



- **Se seleccionó los siguientes conjuntos de datos:**
  - Datos nutricionales del menú de Burger King
  - Diabetes
  - Imágenes de perros y gatos

SPRINT 2

OBJETIVOS:

1. **Implementar técnicas de paralelismo y distribución** para los algoritmos de machine learning desarrollados.
2. **Utilizar Dask y PySpark** para manejar grandes volúmenes de datos y mejorar la escalabilidad de los algoritmos.

LOGROS:

- Se logro aplicar las técnicas de paralelismo y distribución mediante la utilización de Dask y PySpark.

- **Dask:** Utilizar Dask para manejar y analizar grandes volúmenes de datos en paralelo, aprovechando la capacidad de múltiples núcleos de CPU.
- **PySpark:** Utilizar PySpark para distribuir las tareas de procesamiento y entrenamiento de modelos en un clúster, mejorando la escalabilidad y el rendimiento.

- **Se aplicó estas herramientas para algunos algoritmos que son los siguientes:**
  - K-means
  - XGBoost
  - CNN

SPRINT 3

OBJETIVOS:

1. **Optimizar los algoritmos de machine learning** para mejorar su rendimiento y eficiencia.
2. Preparar y presentar los resultados del proyecto.

LOGROS:

- Se logró optimizar los algoritmos, asimismo, se aplico técnicas para mejorar la eficiencia del programa.

- **Para la optimización de los algoritmos se hizo uso**:
  - Grid search, random search.

- Asimismo, para la verificación de la eficiencia del programa se aplicó un Monitoreo tanto del **tiempo de ejecución y el uso de recursos.**

- Además, de **Perfilado de código** con la herramienta cProfile para la identificación de cuello de botella.

Se implemento para los algoritmos de K-means y XGBoost obteniendo los siguientes hiperparametros:

```
{'init': 'k-means++', 'max_iter': 100, 'n_clusters': 2, 'n_init': 10}
```

```
{'eta': 1, 'max_depth': 3, 'objective': 'binary:logistic', 'subsample': 0.5}
```

RESULTADOS

• FUNCIONALIDADES DESARROLLADAS

Algoritmos Implementados con aplicación con Dask y PySpark:

- XGBoost
- Clustering K-Means

Optimización de Modelos

- Reducción del tiempo de entrenamiento
- Aumento de la precisión del modelo, además, las métricas de silhouette_score y davies_bouldin_score

• RESULTADOS DE PRUEBAS Y ANÁLISIS DE RENDIMIENTO

Resultados de las métricas:

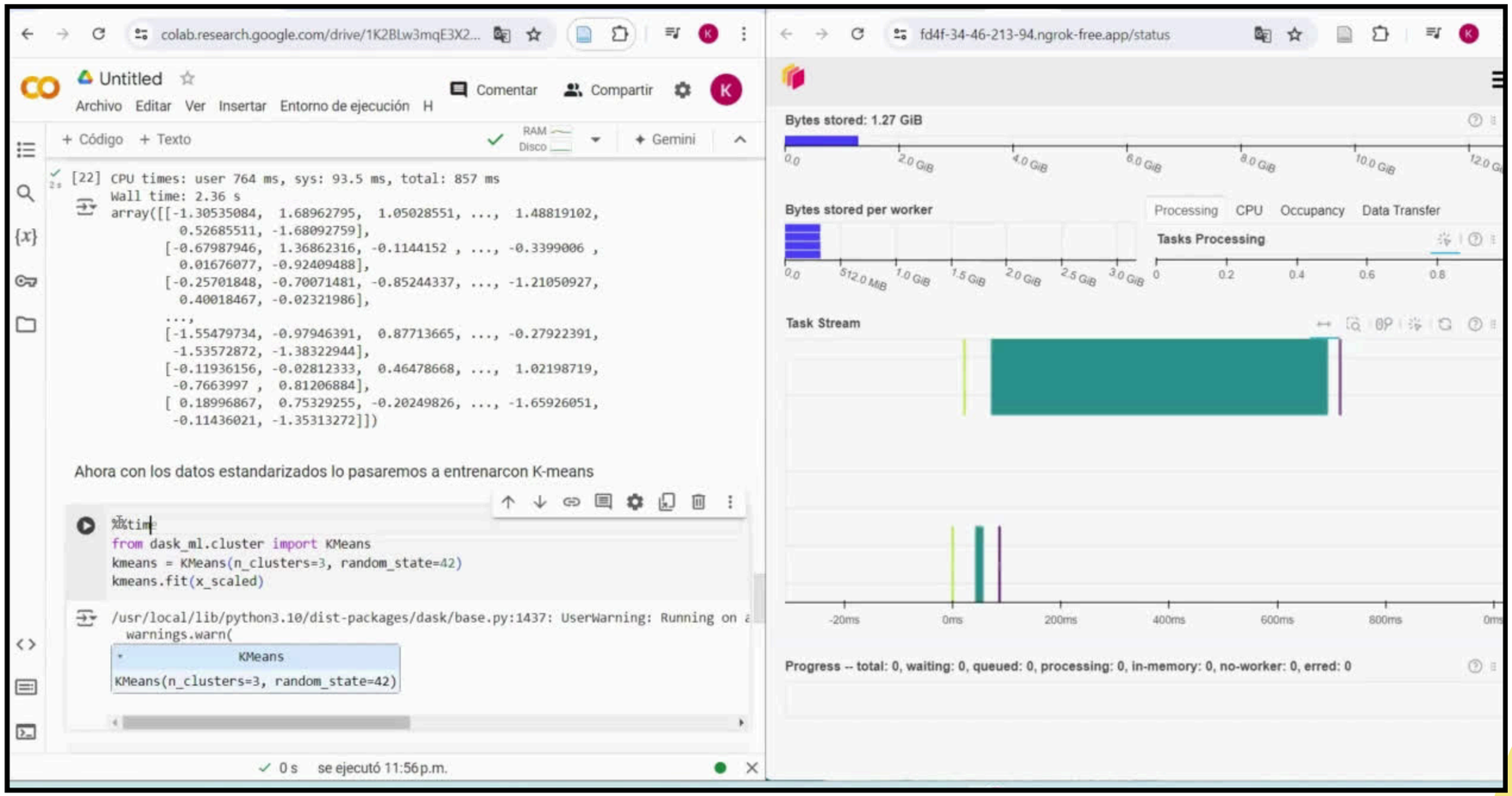
Modelos	Paralelización y distribuida	Con optimización
XGBoost	Accuracy de 74.50%.	Accuracy de 75.84%
K-Means	silhouette_score 0.425 davies_bouldin_ core = 0.656	silhouette_score =0.5292 davies_bouldin_score = 0.68

Resultados del rendimiento:

Modelos	Paralelización y distribuida	Con optimización
XGBoost	Tiempo de ejecución: 4.92 segundos. CPU uso: 100.0% Memoria uso: 13.5%	Tiempo de ejecución: 2.48 segundos. CPU uso: 98.5% Memoria uso: 26.2%
K-Means	Tiempo de ejecución en Dask: 0.54 segundos. CPU uso: 54.8% . Memory uso: 13.1%	Tiempo de ejecución en Dask: 0.03 segundos. CPU uso: 44.2%. Memory uso: 22.9%

DEMOSTRACIÓN EN VIVO

- VIDEO/GIF DE LA FUNCIONALIDAD EN ACCIÓN



ANÁLISIS Y EVALUACIÓN

Lecciones aprendidas

- Se implementaron y evaluaron modelos y se documentaron los resultados.
- La integración de Dask y PySpark en el entorno distribuido.
- Optimizar los algoritmos de machine learning y mejoras en el rendimiento

Desafíos y soluciones

- Problemas de compatibilidad con ciertas versiones de bibliotecas de dask_ml_xgboots.
  - Se resuelve actualizando a versiones compatibles se logra instalando dask_xgboots y xgboots.
- En la integración de Dask y PySpark el tiempo de ejecución era 4.92 segundos.
  - Se aplicó un balanceo de tareas y se obtuvo un tiempo de 2.48 segundos.

CONCLUSIÓN Y FUTURO TRABAJO

Resumen de los logros

Evaluación y Preparación de Modelos

- Implementamos y evaluamos modelos, documentando los resultados.
- Los datos están listos para su uso en aplicaciones futuras.

Integración de Tecnologías Distribuidas

- Integramos Dask y PySpark.

Optimización de Algoritmos

- Mejoramos la eficiencia de los algoritmos de machine learning.

POSIBLES MEJORAS Y EXPANSIONES FUTURAS

- Implementación de más algoritmos con dask y PySpark.
- Aplicar con más cantidad de datos.
- Optimizar con más técnicas para tener un mejor rendimiento.