**Metodologías de gestión y diseño de proyectos Big Data**



**2024**

**Actividad práctica2**

Autor: **Maritza Alexandra Pinta**

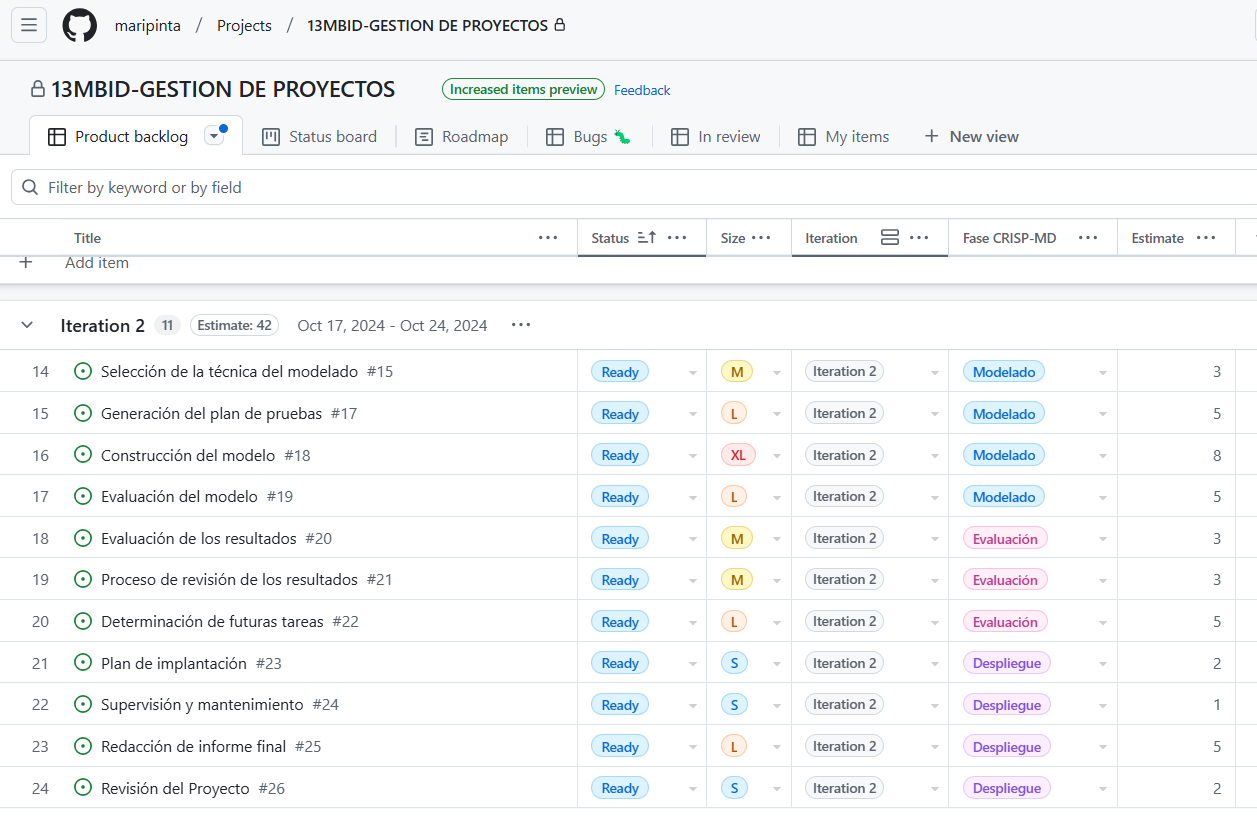
# **ACTIVIDAD PRACTICA 2**

Autor: **MARITZA ALEXANDRA PINTA**

**INICIO DEL SPRINT 2**

Se ha definido la 2da. Iteración del proyecto en ejecución, para esta acción se ha utilizado nuevamente la plataforma GitHub Projects en el mismo proyecto de la iteración anterior. En este caso, las fases de la metodología CRISP-DM abarcadas son:

* Modelado
* Evaluación
* Despliegue



## **[D] MODELADO**

**Selección de la técnica de modelado**

De acuerdo con los objetivos del proyecto, se proponen en primera instancia las siguientes técnicas de modelado para obtener el producto solicitado:

* Métodos de la familia TDIDT (Árboles de Decisión)
* Métodos de regresión
* Redes Neuronales
* Métodos de ensamblado

**Generación del plan de pruebas**

Para el plan de pruebas se ha determinado realizar las siguientes acciones:

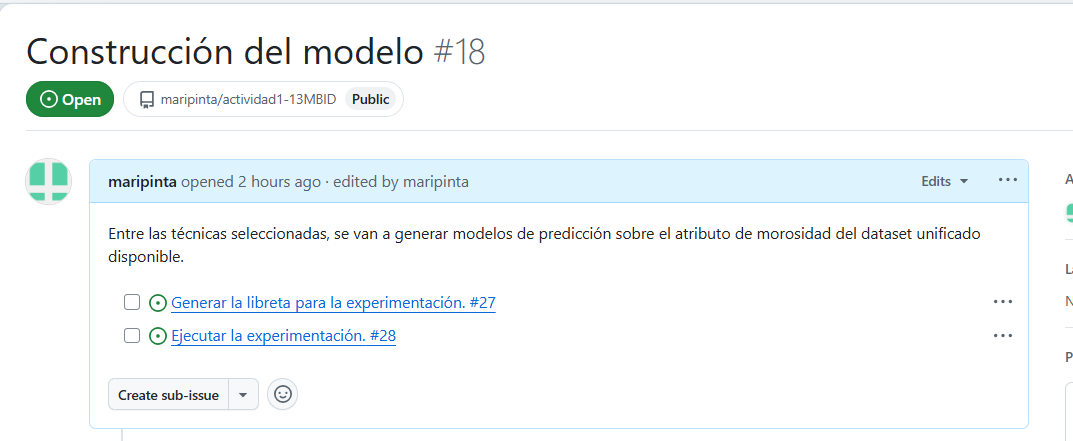
1. Dividir el conjunto de datos disponible en las fracciones por defecto para construir los datasets de entrenamiento y prueba requeridos.

2. Datos de entrenamiento: 75%

3. Datos de prueba: 25%

4. Se van a ejecutar tres (3) instancias de prueba a través de las que se buscará optimizar los resultados obtenidos a fin de lograr cumplir con los objetivos del proyecto en términos de efectividad (80%). En cada instancia de prueba se deberán registrar:

* La técnica utilizada
* Los parámetros de la ejecución realizada
* Evaluar los resultados obtenidos
* A medida que avancen las instancias de prueba, podrán filtrarse (remover) aquellas técnicas que han presentado un rendimiento más bajo que el promedio.
* Tota la experimentación deberá ser registrada mediante la librería mlflow.



**Construcción del modelo**

En esta actividad se van a utilizar diferentes librerías de Python para obtener de forma simple, modelos de predicción sobre el entorno de trabajo definido previamente. El repositorio de código donde se encuentra esta experimentación es:

<https://github.com/users/maripinta/projects/1>

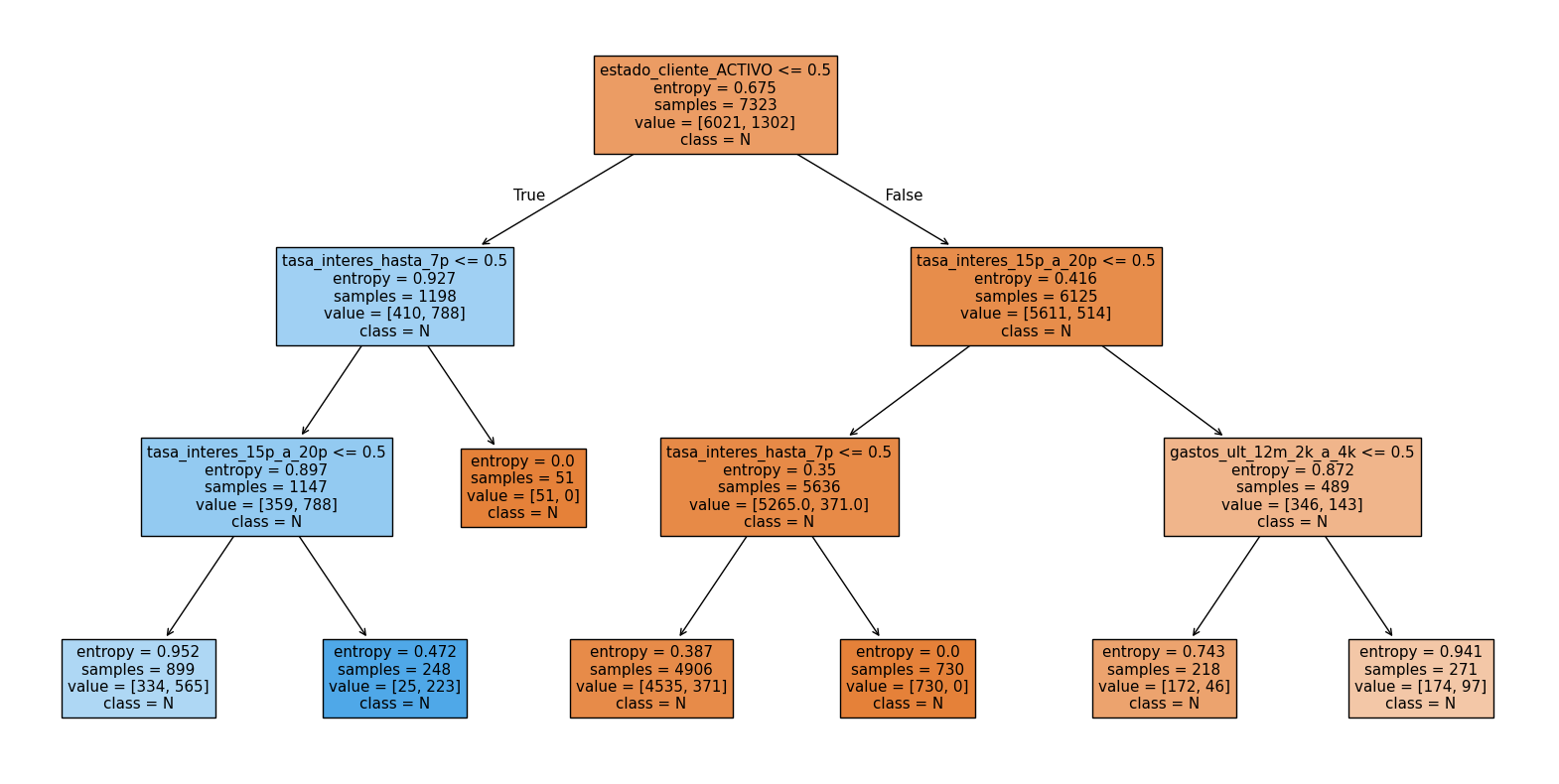
**Evaluación del modelo**

Se documenta en esta sección la ejecución de cada instancia de prueba con las técnicas empleadas y sus resultados obtenidos:

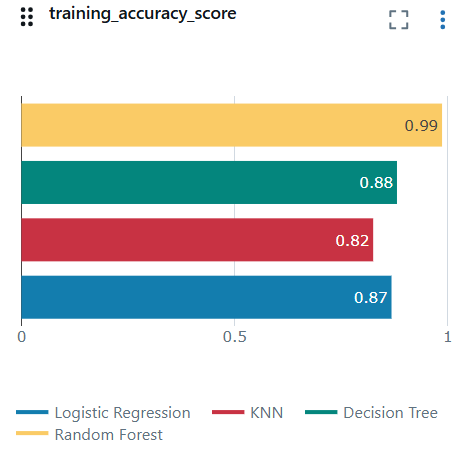
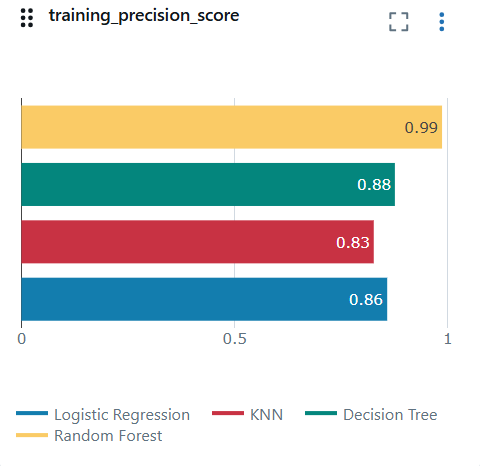
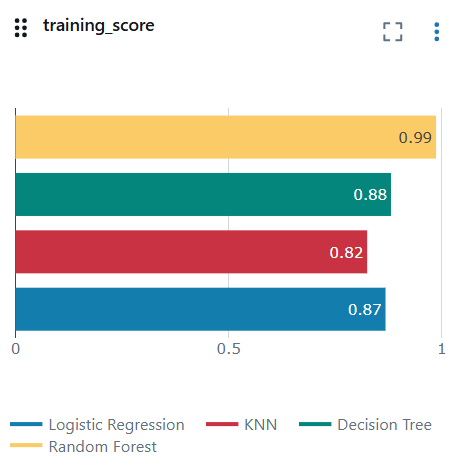
**Prueba No. 1**

|  |  |  |
| --- | --- | --- |
| **Técnica utilizada** | **Parametrización** | **Resultados obtenidos** |
| Regresión logística  LogisticRegression | C 1.0  class\_weight None  dual False  fit\_intercept True  intercept\_scaling 1  l1\_ratio None  max\_iter 100  multi\_class deprecated  n\_jobs None  penalty l2  **random\_state 42**  **solver liblinear**  tol 0.0001  verbose 0  warm\_start False | Rendimiento obtenido: 0.8665028665028665  Matriz de confusión: |
| KNN  KNeighborsClassifier | **Algorithm ball\_tree**  **leaf\_size 25**  metric minkowski  metric\_params None  n\_jobs None  **n\_neighbors 50**  p 2  weights uniform | Rendimiento obtenido: 0.8251433251433251  Matriz de confusión: |
| Árboles de Decisión  DecisionTreeClassifier | ccp\_alpha 0.0  class\_weight None  **criterion entropy**  **max\_depth 3**  max\_features None  max\_leaf\_nodes None  min\_impurity\_decrease 0.0  min\_samples\_leaf 1  **min\_samples\_split 10**  min\_weight\_fraction\_leaf 0.0  monotonic\_cst None  random\_state 42  splitter best | Rendimiento obtenido: 0.8791973791973792  Matriz de confusión: |
| Método de ensamblado de modelos:  RandomForestClassifier | Bootstrap True  ccp\_alpha 0.0  class\_weight None  criterion gini  max\_depth None  max\_features sqrt  max\_leaf\_nodes None  max\_samples None  min\_impurity\_decrease 0.0  min\_samples\_leaf 1  min\_samples\_split 2  min\_weight\_fraction\_leaf 0.0  monotonic\_cst None  **n\_estimators 10**  n\_jobs None  oob\_score False  random\_state 42  verbose 0  warm\_start False | Rendimiento obtenido: 0.8841113841113841  Matriz de confusión: |

[1] Árbol obtenido:



Comparación entre los resultados de esta instancia de pruebas (se obtiene desde mlflow):

**Prueba No. 2**

Dados los resultados de la prueba 1, se determina los siguientes cambios:

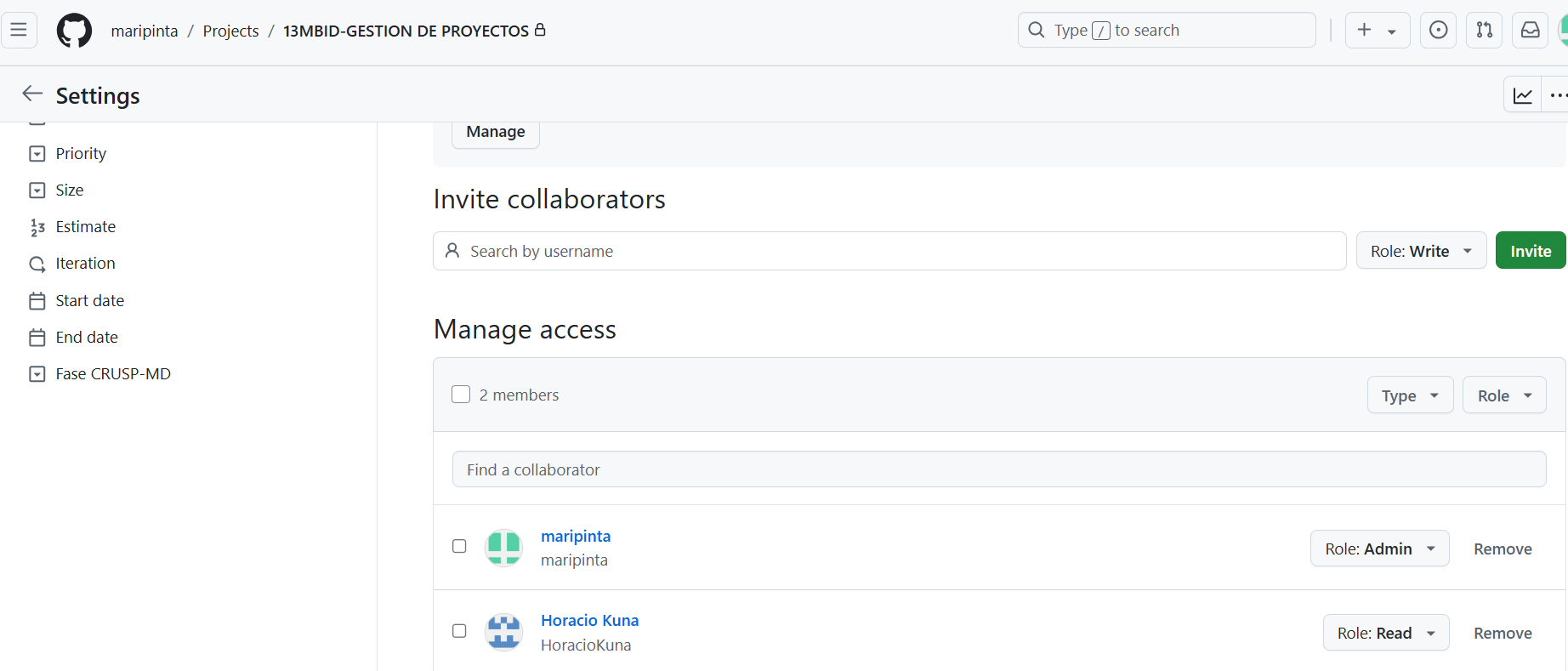
* Reemplazar la técnica KNN por otra dado su bajo rendimiento, con XGBoost porque puede mejorar la precisión y generalización.
* Ajustar los parámetros de las técnicas restantes para buscar optimizar sus resultados:

|  |
| --- |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **Técnica utilizada** | **Parametrización** | **Resultados obtenidos** |
| Árboles de Decisión  DecisionTreeClassifier | ccp\_alpha 0.0  class\_weight None  **criterion gini**  **max\_depth 5**  max\_features None  max\_leaf\_nodes None  min\_impurity\_decrease 0.0  min\_samples\_leaf 1  **min\_samples\_split 5**  min\_weight\_fraction\_leaf 0.0  monotonic\_cst None  random\_state 42  splitter best | Rendimiento obtenido: 0.8783783783783784  Matriz de confusión: |
| KNN  KNeighborsClassifier |  | Rendimiento obtenido:  Matriz de confusión: |
| Regresión logística  LogisticRegression |  | Rendimiento obtenido:  Matriz de confusión: |
| Método de ensamblado de modelos:  RandomForestClassifier |  | Rendimiento obtenido:  Matriz de confusión: |

## **[E] EVALUACION**

**[F] DESPLIEGUE / IMPLEMENTACIÓN**



Repositorio publico

