Proyecto MLOps

Integrantes:

Christian Andrés Delgado Cod. 2244244

Luis Felipe Campo Cod. 2243804

Repositorio proyecto y dataset: [felipkmpo/proyectomlops]

Dataset:

Se utilizó conjunto de datos de imágenes de cáncer de piel, el cual cuenta con dos clases, “maligno” y “benigno”, conjunto de datos disponible en kaggle.

Link dataset kaggle: [https://www.kaggle.com/datasets/rm1000/skin-cancer-isic-images]

Después del preprocesamiento de los datos, se dividió el dataset de la siguiente manera:

```
Shape of x_train: (2307, 100, 100, 3)
Shape of x_test: (891, 100, 100, 3)
Shape of x_val: (99, 100, 100, 3)
```

Observación: Se desarrollaron 3 experimentos, cada uno con parámetros diferentes que permitieron obtener métricas independientes y únicas.

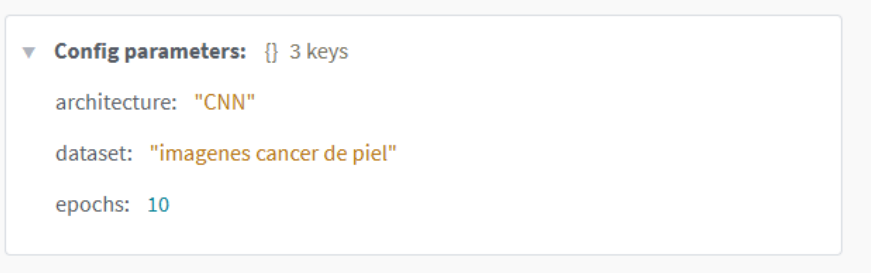
DESARROLLO EXPERIMENTOS

EXPERIMENTO 1

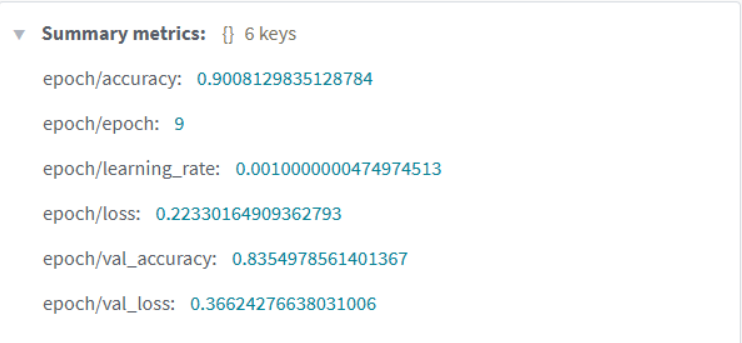
WANDB

Parámetros

- **Épocas 10**



Resumen de métricas



Gráficas Wandb

El primer gráfico, representa la **precisión del modelo** en el conjunto de entrenamiento a medida que avanza cada época, este experimento tiene 10 épocas parametrizadas, la precisión comienza alrededor del 0.7 en la primera época y se incrementa gradualmente

hasta alcanzar aproximadamente 0.87 en la última época; en conclusión el modelo está aprendiendo a clasificar mejor los datos de entrenamiento.

En el segundo gráfico identificamos la **precisión en el conjunto de validación**, aquí se mide el desempeño del modelo en datos que no ha visto durante el entrenamiento, en este gráfico la precisión fluctúa entre épocas, alcanzando un valor de aproximadamente 0.82 en la sexta época, estas fluctuaciones pueden ser indicativas de sobreajuste.

El tercer gráfico nos muestra **la pérdida en el conjunto de validación,** esto nos indica que tan bien o mal se están realizando las predicciones; en este experimento la pérdida comienza alrededor del 0.55 y disminuye con las épocas, en general una pérdida más baja indica mejor rendimiento.

MLFLOW

Resumen métricas

Metrics (6)

Search metrics

Metric	Value
val_accuracy	0.8354978561401367
validation_accuracy	0.8354978561401367
validation_loss	0.36624276638031006
val_loss	0.36624276638031006
loss	0.22330164909362793
accuracy	0.9008129835128784

Gráficas MLFLOW

El primer gráfico muestra la **pérdida en el conjunto de validación,** la tendencia a disminuir en las últimas épocas indica que el modelo está generalizando mejor en el conjunto de validación.

El segundo gráfico muestra la **pérdida en el conjunto de entrenamiento,** la pérdida en entrenamiento comienza alta, desciende de manera constante hasta alcanzar 0.2 en la última época, esta disminución constante sugiere que el modelo está aprendiendo bien los patrones en los datos de entrenamiento.

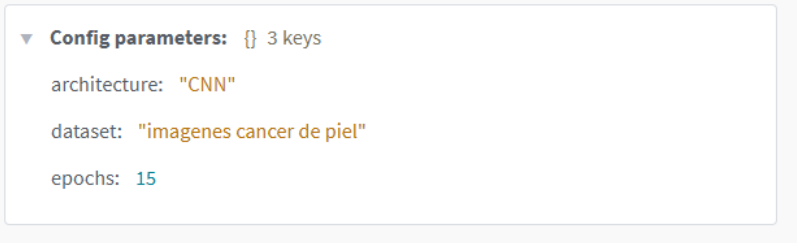
La tercera gráfica muestra la **precisión en el conjunto de entrenamiento,** la precisión empieza alrededor de 0.7 y aumenta hasta superar el 0.9 en la última época, esto es una señal que el modelo está aprendiendo y ajustándose a los datos de entrenamiento.

EXPERIMENTO 2

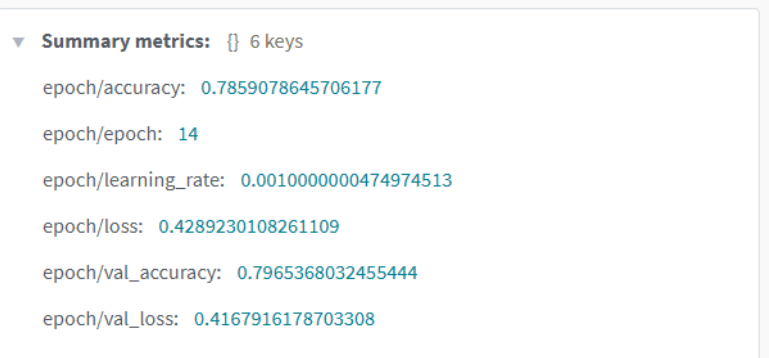
WANDB

Parámetros

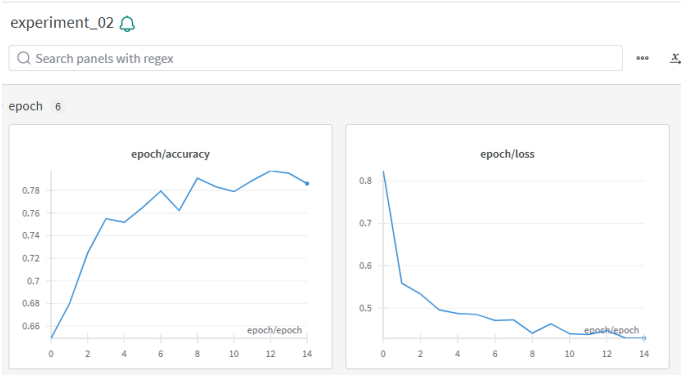
- Se aumentó épocas a 15
- Se implementó dropout 0.5



Resumen de métricas



Gráficas Wandb



La primera gráfica nos muestra la relación entre épocas y precisión, esta gráfica sugiere que el modelo está aprendiendo a clasificar los datos correctamente a medida que se expone a más ejemplos.

La segunda gráfica relacionada con la pérdida y las épocas, identificamos una disminución en la pérdida signo de que el modelo está aprendiendo, sin embargo el aumento en la pérdida alrededor de la época 10 podría significar un ajuste inadecuado de hiperparametros.

MLFLOW

Resumen métricas

Metrics (6)

Search metrics

Metric	Value
val_accuracy	0.7965368032455444
validation_accuracy	0.7965368032455444
validation_loss	0.4167916178703308
val_loss	0.4167916178703308
loss	0.42892301082611084
accuracy	0.7859078645706177

Gráficas MLFLOW

El primer gráfico muestra la **pérdida en el conjunto de validación,** la tendencia a disminuir en las últimas épocas indica que el modelo está generalizando mejor en el conjunto de validación.

El segundo gráfico muestra la **pérdida en el conjunto de entrenamiento,** esta mide que tan bien el modelo se ajusta a los datos de entrenamiento, al disminuir indica que el modelo está aprendiendo a representar los patrones presentes en los datos de entrenamiento.

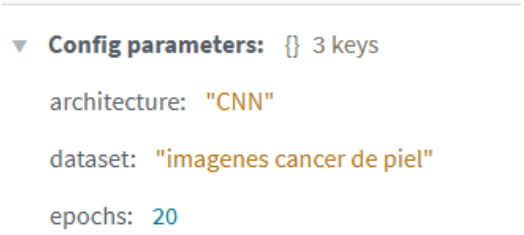
La tercera gráfica muestra la **precisión en el conjunto de entrenamiento,** la precisión mide el porcentaje de predicciones correcta que realiza el modelo, si esta aumenta quiere decir que el modelo está cada vez más seguro de sus predicciones y está clasificando correctamente una mayor proporción de los datos.

EXPERIMENTO 3

WANDB

Parámetros

- Se aumentó épocas a 20
- Se eliminó dropout



Resumen métricas

Summary metrics: {} 6 keys

epoch/accuracy: 0.9647696614265442

epoch/epoch: 19

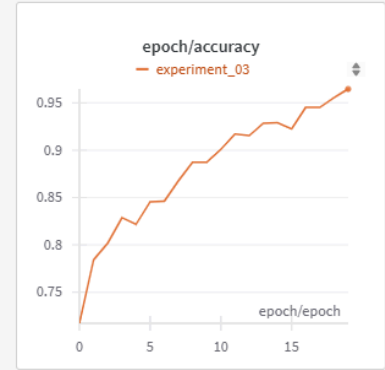
epoch/learning_rate: 0.0010000000474974513

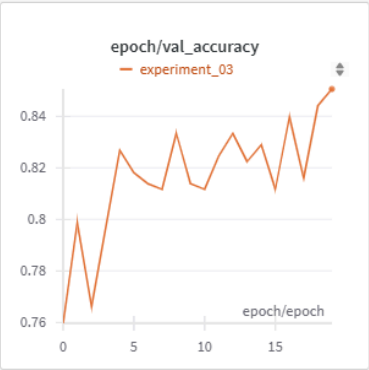
epoch/loss: 0.12069535255432128

epoch/val_accuracy: 0.850649356842041

epoch/val_loss: 0.4227392077445984

GRÁFICAS WANDB





El primer gráfico representa la **precisión del modelo** en el conjunto de entrenamiento a medida que avanza cada época, este experimento tiene 20 épocas parametrizadas, la precisión comienza alrededor del 0.7 en la primera época y se incrementa gradualmente hasta alcanzar aproximadamente 0.96 en la última época; en conclusión, el modelo está aprendiendo a clasificar mejor los datos de entrenamiento, este ha sido la mejor precisión.

En el segundo gráfico identificamos la **precisión en el conjunto de validación**, se identifica que la precisión aumenta pero con algunas fluctuaciones, esto sugiere que el modelo está generalizando bien a nuevos datos, aunque podría haber un ligero sobreajuste hacia el final del entrenamiento.

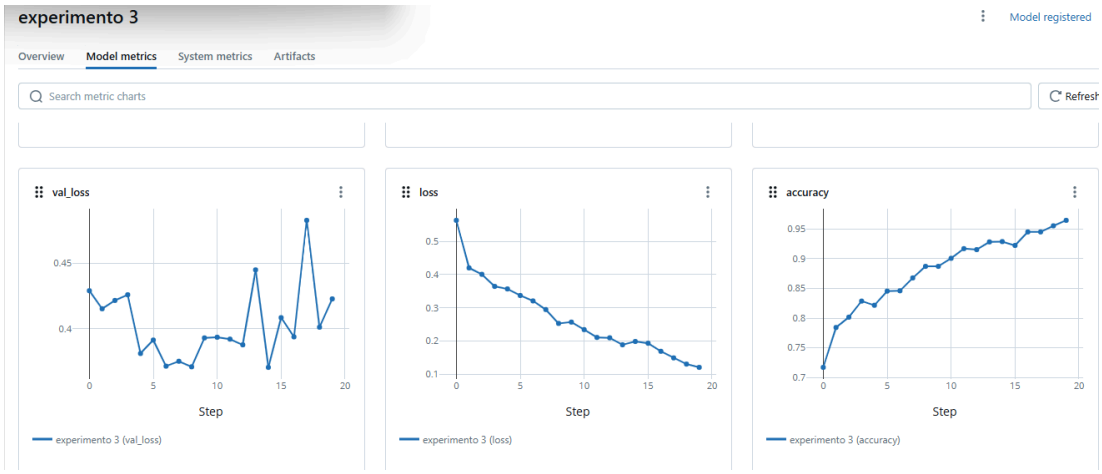
El tercer gráfico nos muestra **la pérdida en el conjunto de validación,** la pérdida disminuye al principio, lo que indica que el modelo está aprendiendo, luego comienza a aumentar ligeramente, esto podría ser un signo de sobreajuste, ya que se puede estar especializando en los datos de entrenamiento y no llegará a generalizar bien datos nuevos.

MLFLOW

Resumen métricas

Metric	Value
val_accuracy	0.850649356842041
validation_accuracy	0.850649356842041
validation_loss	0.4227392077445984
val_loss	0.4227392077445984
loss	0.12069535255432129
accuracy	0.9647696614265442

Gráficas MLFLOW



El primer gráfico muestra la **pérdida en el conjunto de validación,** la tendencia a disminuir en las últimas épocas indica que el modelo está generalizando mejor en el conjunto de validación.

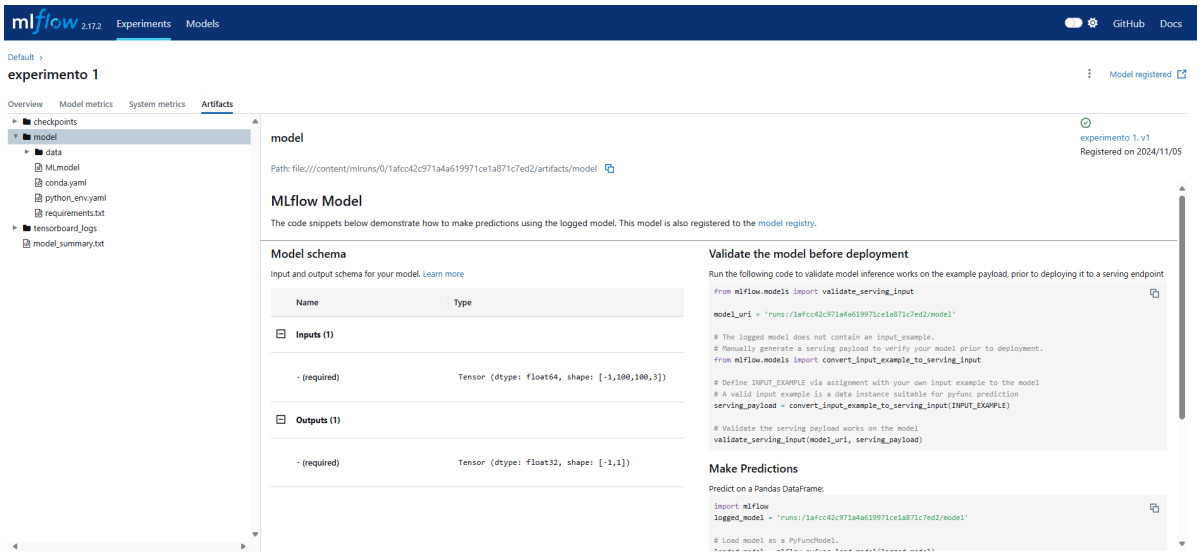
El segundo gráfico muestra la **pérdida en el conjunto de entrenamiento,** esta disminuye de manera suave y constante a los largo del entrenamiento, esto indica que el modelo está aprendiendo a representar los patrones presentes en los datos de entrenamiento.

La tercer gráfica muestra la **precisión en el conjunto de entrenamiento,** la precisión mide el porcentaje de predicciones correctas que realiza el modelo, en este experimento el modelo muestra el valor más alto de precisión, por encima del 90%, sin embargo no se debe descuidar el sobreajuste para validar que el modelo esté generalizando bien.

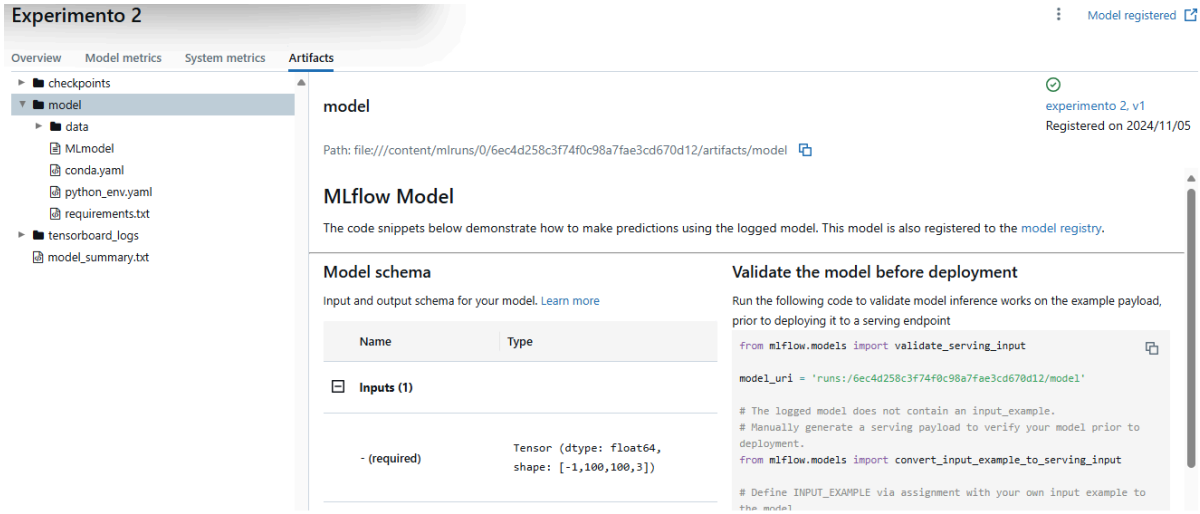
REGISTRO MODELOS PLATAFORMA MLFLOW

La implementación del registro de modelos se ejecuta directamente desde el código, antes del entrenamiento o compilación de modelo de aprendizaje mediante la instrucción "mlflow.tensorflow.autolog()" la cual está dejando guardadas todas la metricas, parámetros y el modelo directamente en la plataforma mlflow.

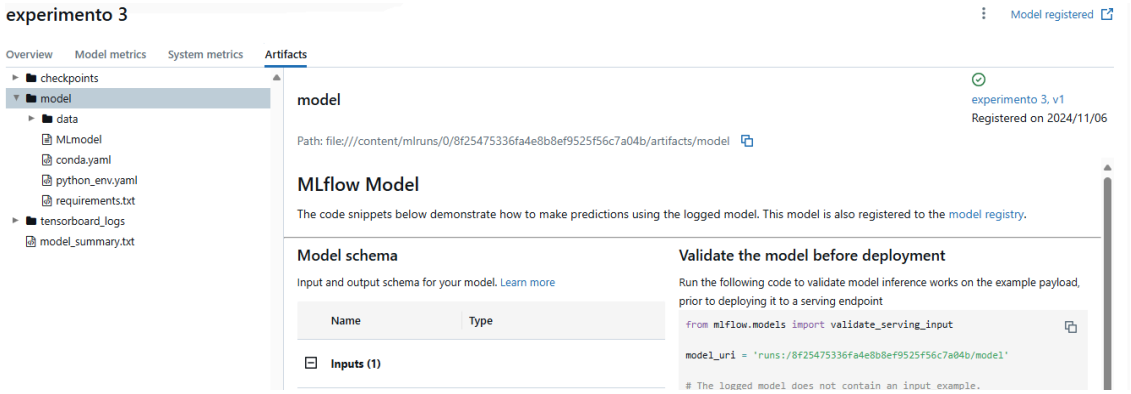
Registro modelo experimento 1



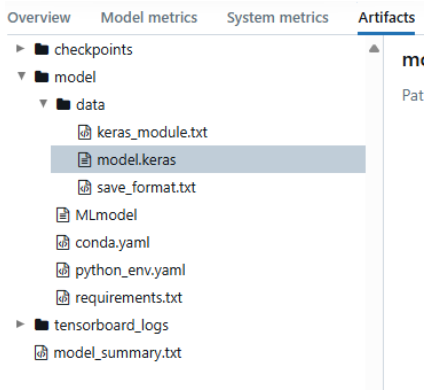
Registro modelo experimento 2



Registro modelo experimento 3



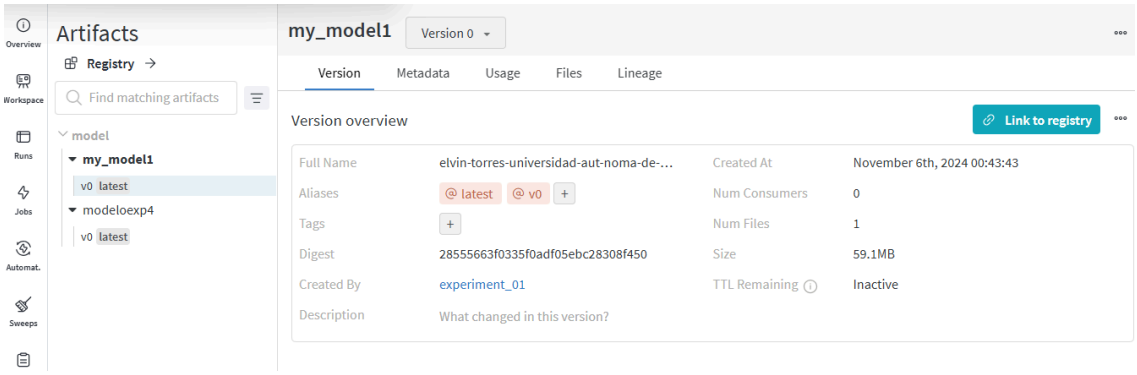
Observación: Se registra modelo del experimento en plataforma mlflow en el apartado “artifacts”, si se requiere utilizar se pueden utilizar para algún despliegue o implementar predicción se utilizarán los comandos indicados o se puede descargar el archivo del modelo “model.keras”.



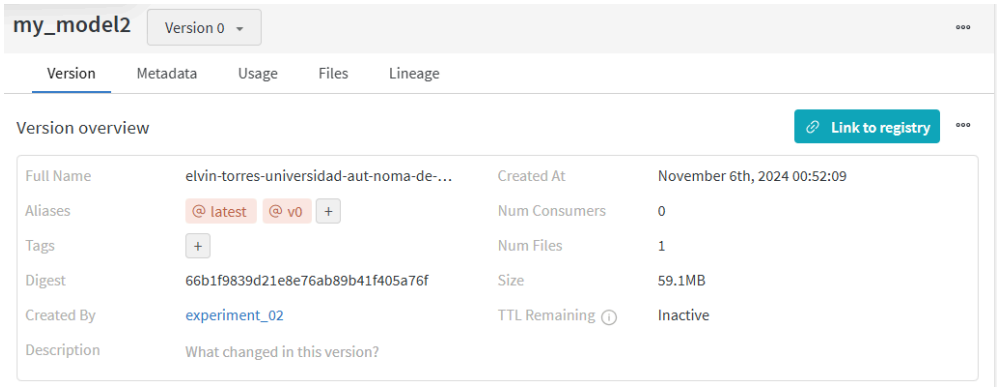
REGISTRO MODELOS PLATAFORMA WANDB

El registro de los modelos en la plataforma wandb se realiza directamente en el desarrollo del código, se implementa después del entrenamiento del modelo, mediante la instrucción “model.save("my_model.h5") “, este queda almacenado en el menú “artifacts” de la plataforma wandb.

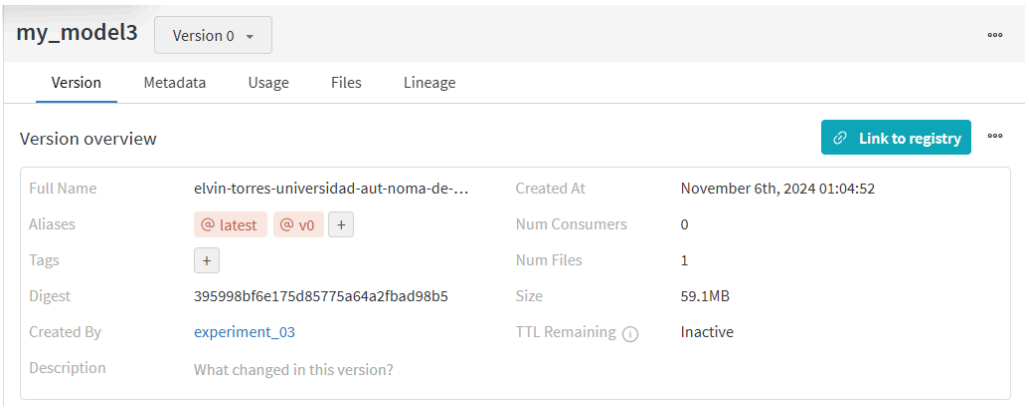
Modelo Experimento 1



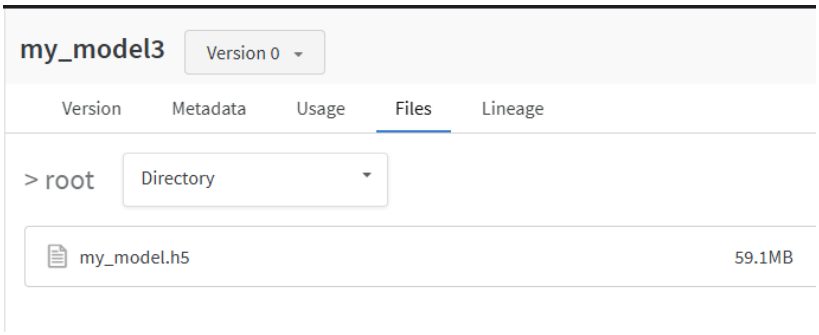
Modelo Experimento 2



Modelo Experimento 3



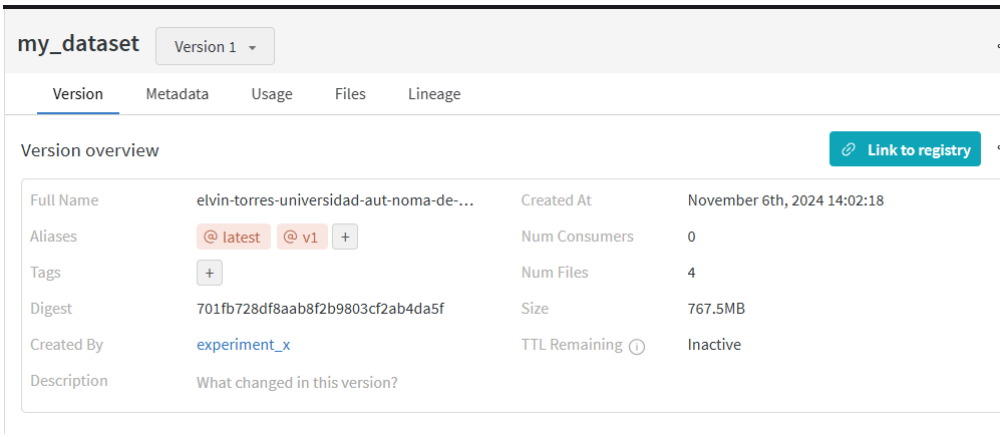
Observación: en la pestaña “usage” encontraremos los comandos para utilizar nuestro modelo en el proyecto que necesitemos mediante api o si se requiere desde la pestaña “file” podremos descargar el modelo formato h5.

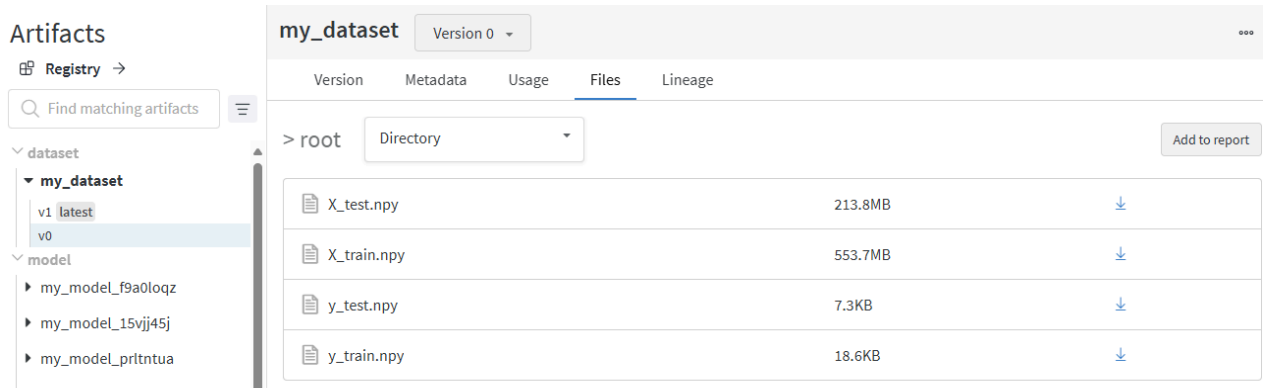

my_model3
Version 0
Version
Metadata
Usage
Files
Lineage
> root
Directory
my_model.h5
59.1MB

REGISTRO DATASET ENTRENAMIENTO Y PRUEBAS PLATAFORMA WANDB

Para el registro del conjunto de datos de pruebas y entrenamiento utilizaremos el comando dataset_artifact = wandb.Artifact("my_dataset", type="dataset"), el cual se implementará dentro del código desarrollado, y se ubicará en justo después del wandb.init con el objetivo que todo el conjunto de datos quede registrado antes de compilación o entrenamiento del modelo requerido.

El dataset lo podremos identificar dentro de la opción artifacts, donde se nos listaran los datasets y modelos disponibles.





REGISTRO DATASET ENTRENAMIENTO Y PRUEBAS PLATAFORMA MLFLOW

Para el registro del conjunto de datos de pruebas y entrenamiento en mlf implementaremos el registro del artefacto en mlflow, para esto implementaremos el código necesario dentro del bloque with mlflow.start_run().

Para el uso de artefacto tipo dataset, crearemos una carpeta temporal y en esta guardaremos arreglos tipo numpy con nuestros conjuntos de datos.

Cuando se termine de guardar los conjuntos de datos dentro de los arreglos se implementará la instrucción “mlflow.log_artifacts(temp_dir, artifact_path="dataset")” con

el fin de registrar el o los conjuntos de datos.

experimento_3