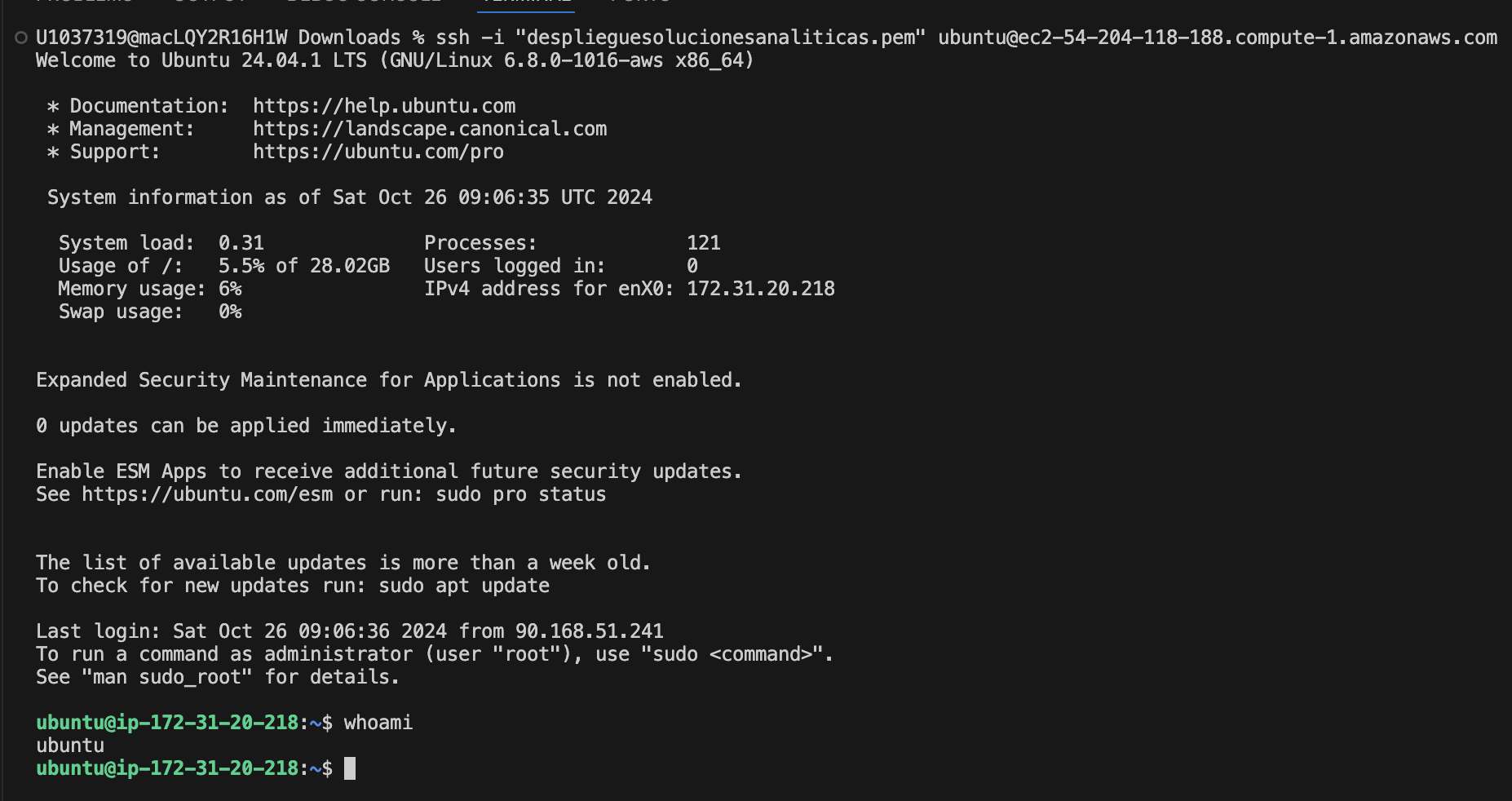
**Instale y pruebe MLflow en una máquina virtual**

* Incluya un pantallazo de la consola de AWS EC2 con la máquina en ejecución en su reporte

A screenshot of a computer

Description automatically generated

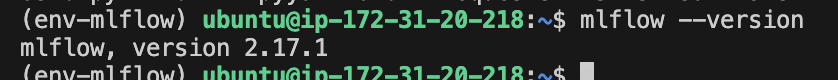
* Incluya en su reporte un screenshot de la conexión a la máquina virtual.



* Incluya en su reporte un screenshot de la instalación de mlfow

A screenshot of a computer

Description automatically generated



* Descargue el archivo mlfow-diab.py que encontrará en Coursera. Estudie con cuidado el código y describa en su reporte qué hace:

El código descarga datos de diabetes, tanto variables independientes como la variable dependiente. Luego reparte los datos en un set de pruebas y train. Seguidamente se mportan las librerías de mlflow y se inicializa el servidor donde va a correr mlflow (<http://localhost:5000>). Luego se registra el experimento y se realiza una ejecución del experimento con los parámetros definidos (n\_estimators = 200, max\_depth = 6, max\_features = 4). Se usa el algoritmo RandomForestRegressor para entrenar el modelo y se realizan las predicciones correspondientes usando el set de datos de train. Luego se registran los parámetros del modelo en mlflow, se registra el modelo. Finalmente se crea la metrica de MSE usando el set de datos de test y se registra su resultado en mlflow.

* En la terminal de la máquina ejecute mlow como servidor y exponga el servicio por el puerto 8050. mlflow server -h 0.0.0.0 -p 8050. Tome un pantallazo de la salida de la terminal e inclúyalo en su reporte.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

* Incluya en su reporte una gráca comparando los dos modelos y justique el comportamiento de acuerdo con los parámetros explorados

Para este caso corri tres modelos con parámetros diferentes cada uno

A screenshot of a computer

Description automatically generated

* Repita este procedimiento con el notebook mlow-mnist que encontrará en Coursera. Incluya en su reporte una gráca comparando los modelos entrenados y justique el comportamiento de acuerdo con los parámetros explorados.

El primer modelo (en golor gris) corresponde al modelo con los parámetros por defecto del Notebook. En el segundo modelo, de color verde, se ajustaron los parámetros de la siguiente manera:  
--batch\_size 512

--epochs 13

--learning\_rate 0.08

--dropout 0.5

A screenshot of a computer

Description automatically generated

Como podemos observar el desempeño del modelo a nivel de accuracy no mejora significativamente en el modelo de color verde.

En este modelo se esta entrenando redes neuronales usando gradiente descendiente para encontrar los parámetros optimos. Se usan las bibliotecas de python Keras y TensorFlow. Los datos que se están usando para el modelo son MNIST, que consiste en imágenes de dígitos escritos a mano. Las imágenes se normalizan para que sus valores estén entre 0 y 1. Se usa una función run\_mlflow para registrar el modelo, los parametros y sus resultados en MLflow para llevar trasabilidad de los experimentos realizados.

**Instale y pruebe MLflow en Databricks**

Incluya en su reporte un screenshot de su cuenta de Databricks A screenshot of a computer

Description automatically generated

* Revise los resultados en Experiments, genere grácas comparativas. Compare los dos modelos en términos de MSE (error cuadrático medio). Incluya en su reporte una gráca comparando los dos modelos y justique el comportamiento de acuerdo con los parámetros explorados.

El primer modelo (color Azul) se ejecuto con los parámetros por defecto configurados en el notebook. El segundo modelo (Color verde) se ejecuto con los parámetros:

n\_estimators = 600

max\_depth = 8

max\_features = 7

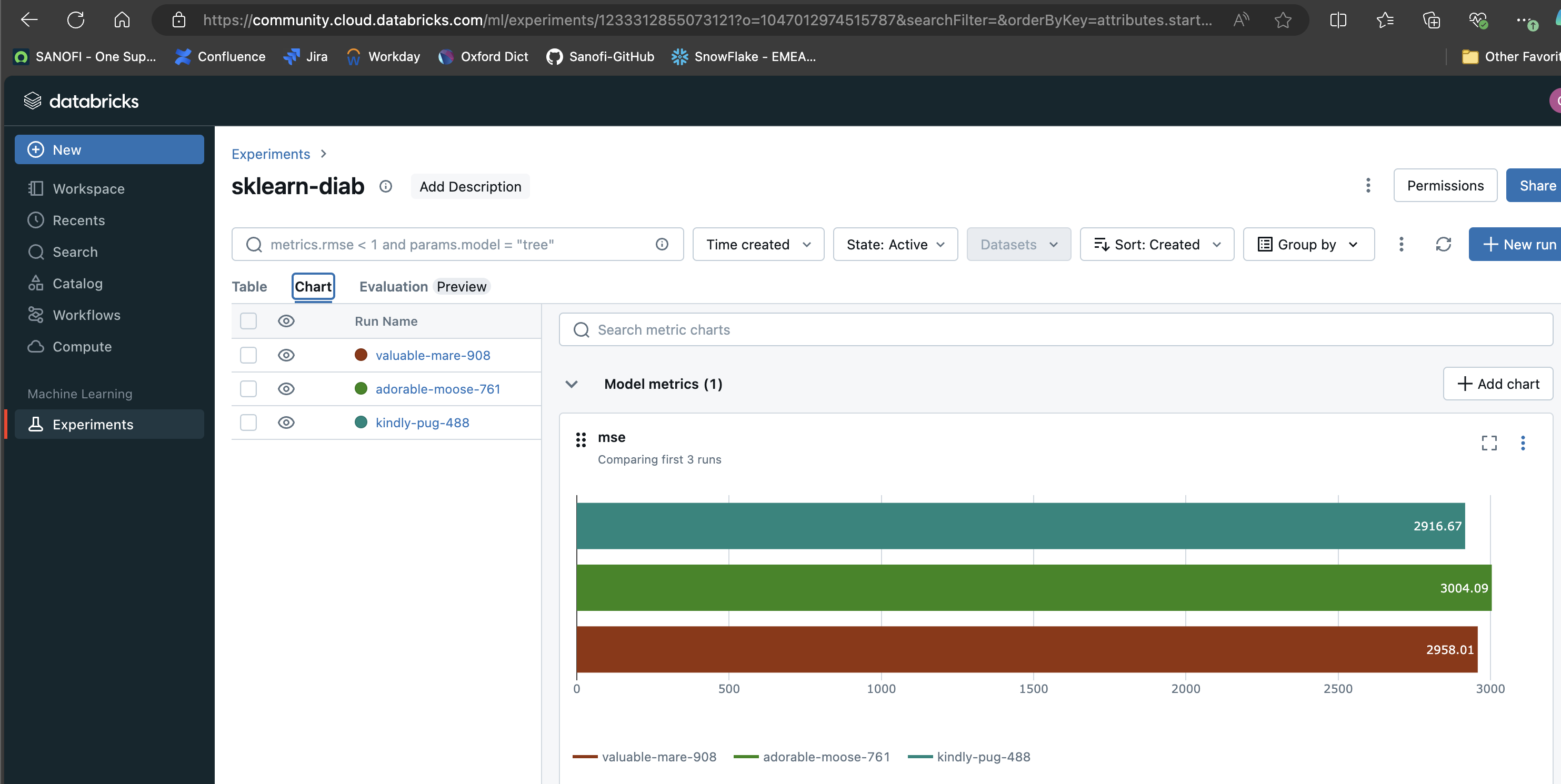
El tercer modelo (color cafe) se ejecuto con los parámetros:

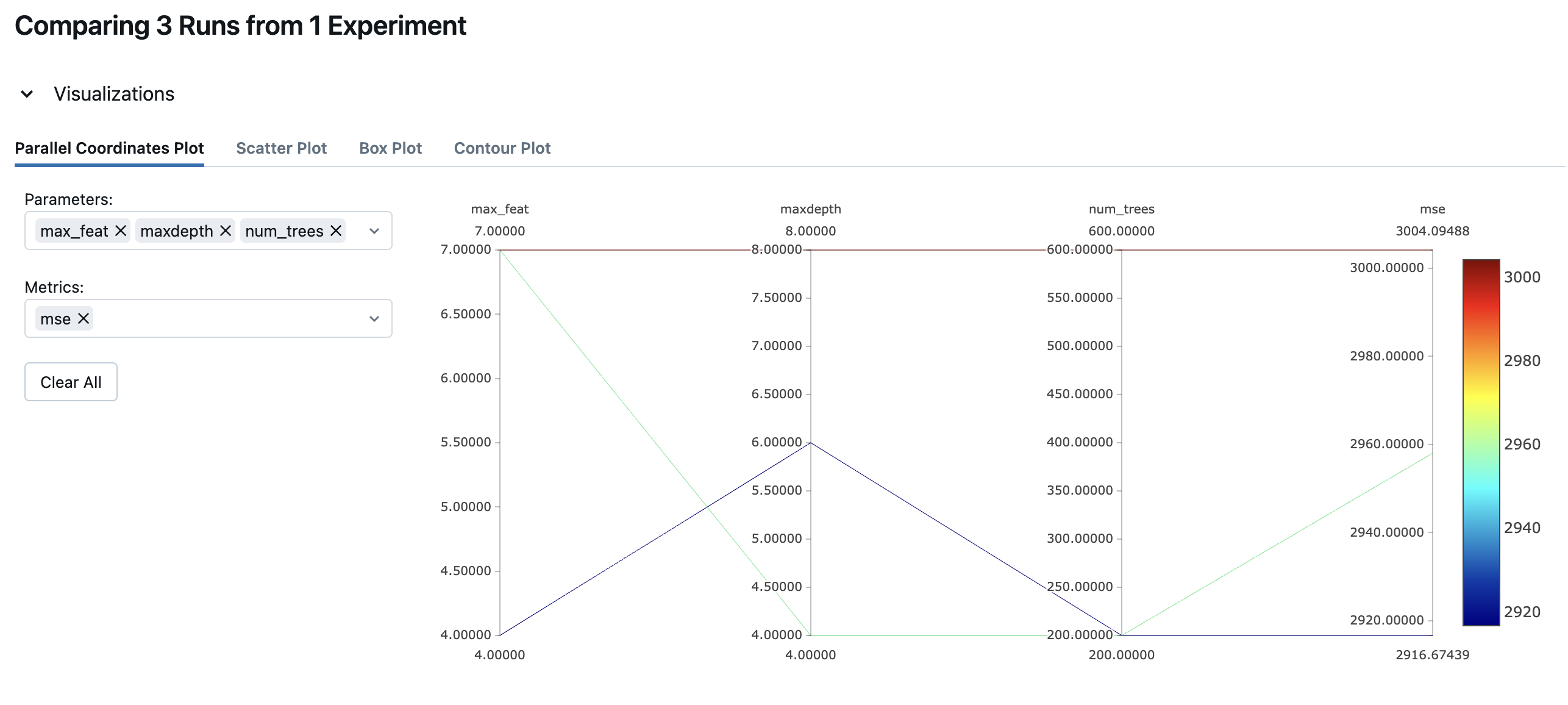
n\_estimators = 200

max\_depth = 4

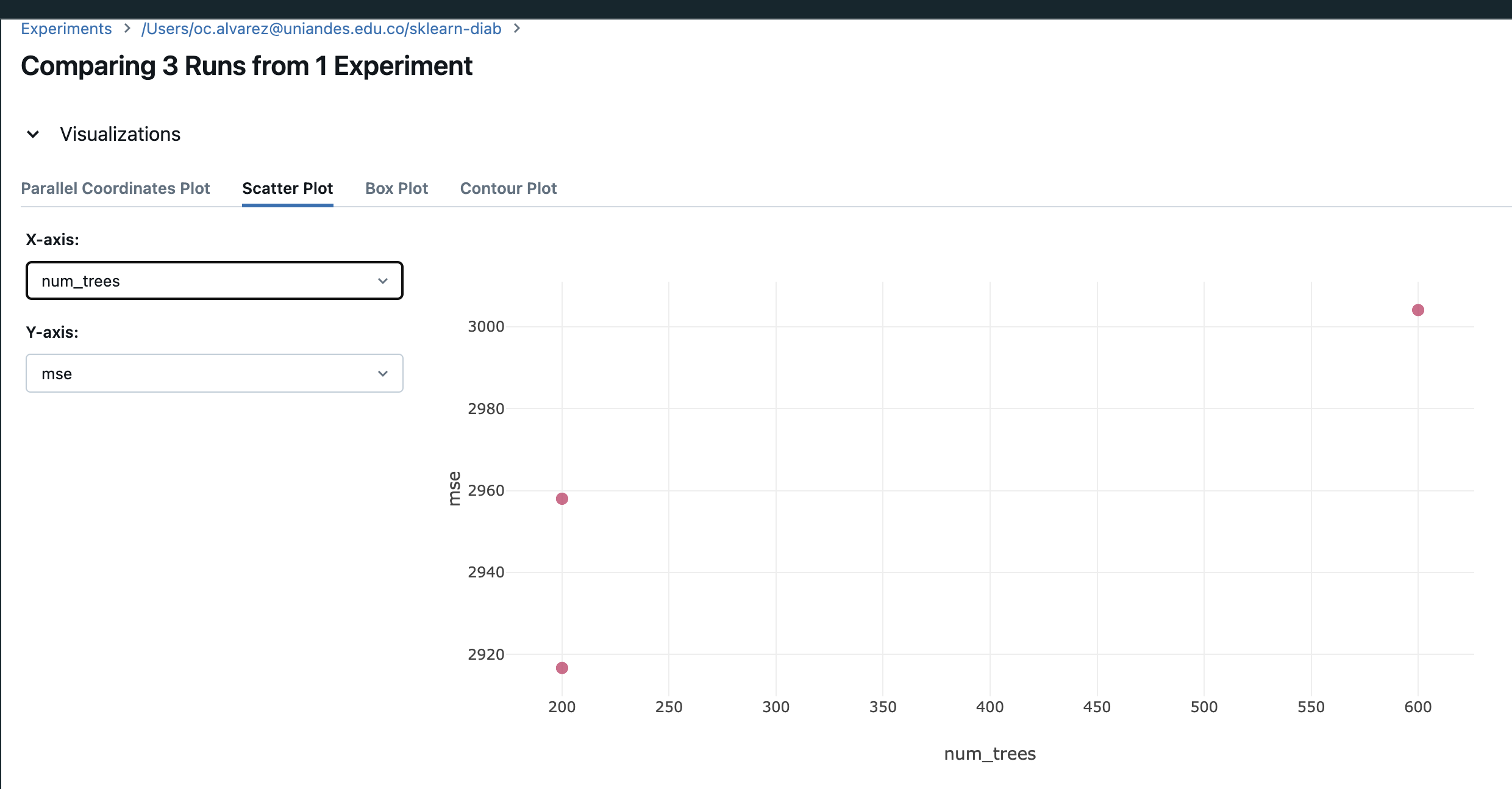
max\_features = 7

Como pdoemos observer en las graficas de MSE, el modelo que tiene mejor desempeño es el original, con un MSE de 2916.67. El segundo modelo tiene un desempeño inferior, muy seguramente por la cantidad de estimadores y la profundiad de los arboles ya que podría generar un sobre ajuste del modelo y al momento de predecir no se obtienen estimaciones precisas.





Podemos evidenciar que cuando el numero de arboles aumento el MSE se impacto negativamente.



* Repita este procedimiento con el notebook mlow-mnist que encontrará en Coursera. Incluya un pantallazo de las grácas en su reporte

En este notebook se esta entrenando redes neuronales usando gradiente descendiente para encontrar los parámetros optimos. Se usan las bibliotecas de python Keras y TensorFlow. Los datos que se están usando para el modelo son MNIST, que consiste en imágenes de dígitos escritos a mano. Las imágenes se normalizan para que sus valores estén entre 0 y 1. Se usa una función run\_mlflow para registrar el modelo, los parametros y sus resultados en MLflow para llevar trasabilidad de los experimentos realizados.

El primer modelo (color verde) se ejecuto con los parámetros por defecto configurados en el notebook. El segundo modelo (Color azul) se ejecuto con los parámetros:

--epochs 8

--learning\_rate 0.1

--num\_hidden\_layers 3

El tercer modelo (color cafe) se ejecuto con los parámetros:

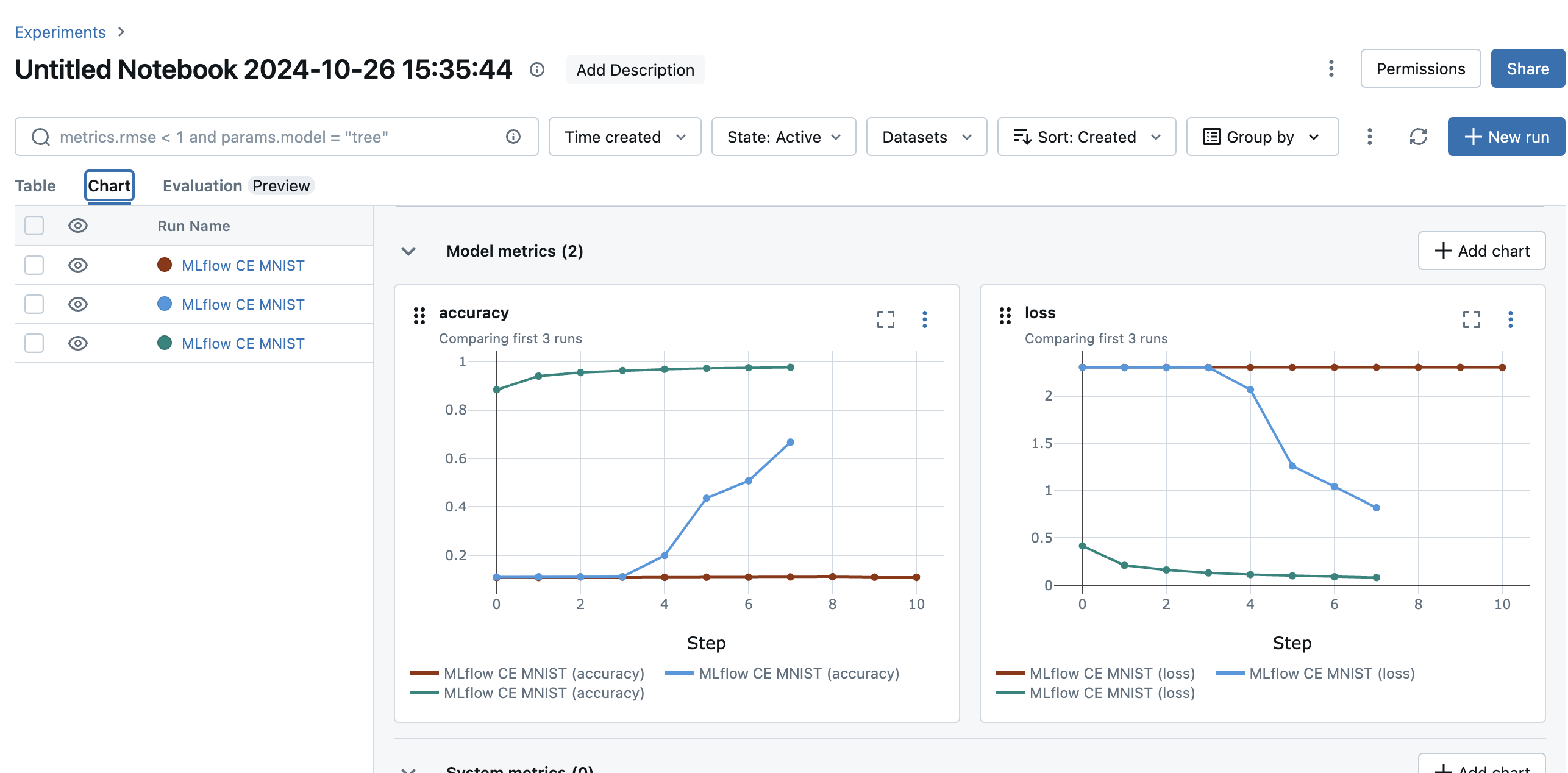
--epochs 11

--learning\_rate 0.2

--num\_hidden\_layers 10

--dropout 0.5

Podemos observar que el modelo que mejor desempeño logror de acuerdo a la metrica de Accuracy es el primero. Logra tener un accuracy mas rápido. El tercer modelo no tiene un desemepeño bueno. Este tercer modelo se configuro con 10 capas ocultas lo que puede estar generando que se aprendan patrones demasiado específicos de los datos de entrenamiento y genere que al realizar estimaciones con datos nuevos su desempeño se malo comparado con los otros modelos.



De igual forma podemos ver que al aumentar el numero de épocas el accuracy del modelo se afecta. Cuando tenemso once épocas el accuracy tiene un impacto negativo alto. El equilibrio del numero de épocas debe considerarse en relación a los otros parámetros que se pueden ajustar del modelo.

A screenshot of a graph

Description automatically generated