**Run a training script as a command job in Azure Machine Learning**

Learn how to convert your code to a script and run it as a command job in Azure Machine Learning.

**Learning objectives**

In this module, you'll learn how to:

* Convert a notebook to a script.
* Test scripts in a terminal.
* Run a script as a command job.
* Use parameters in a command job.

-- **Conversión de un Notebook a un Script**

Cuando uses notebooks para la experimentación y el desarrollo, es posible que necesites convertir el notebook en un script. Alternativamente, podrías optar por trabajar directamente con scripts. Aquí hay algunas recomendaciones para crear un script listo para producción:

1. **Eliminar Código No Esencial**:
   * Los notebooks permiten explorar datos rápidamente mediante comandos como print() y df.describe(). Sin embargo, al crear un script para automatización, debes evitar incluir código que no sea necesario para el flujo principal.
   * Es crucial eliminar cualquier código no esencial para reducir costos y tiempo de computación.
2. **Refactorizar el Código en Funciones**:
   * Organiza tu código en funciones reutilizables, lo que mejora la legibilidad y facilita las pruebas.
3. **Probar el Script en la Terminal**:
   * Ejecuta el script en una terminal para asegurarte de que funcione como se espera y para verificar que las salidas sean correctas.

Con estos pasos, tendrás un script más limpio y eficiente, listo para producción y que puede ser ejecutado como un trabajo por comandos en Azure Machine Learning.

**Refactorización de Código y Pruebas de Scripts**

Cuando uses código en procesos empresariales, es fundamental que sea fácil de leer y mantener. Una forma común de lograr esto es mediante el uso de funciones. Esto no solo mejora la legibilidad, sino que también permite realizar pruebas más fácilmente en partes específicas del código.

**Refactorización del Código en Funciones**

Por ejemplo, si en un notebook tienes el siguiente código para leer y dividir datos:

# leer y visualizar los datos

print("Leyendo datos...")

df = pd.read\_csv('diabetes.csv')

df.head()

# dividir datos

print("Dividiendo datos...")

X, y = df[['Pregnancies', 'PlasmaGlucose', 'DiastolicBloodPressure', 'TricepsThickness', 'SerumInsulin', 'BMI', 'DiabetesPedigree', 'Age']].values, df['Diabetic'].values

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=0)

Puedes refactorizar el código en dos funciones: una para leer los datos y otra para dividirlos. Un ejemplo de código refactorizado podría ser:

def main(csv\_file):

# leer datos

df = get\_data(csv\_file)

# dividir datos

X\_train, X\_test, y\_train, y\_test = split\_data(df)

# función que lee los datos

def get\_data(path):

df = pd.read\_csv(path)

return df

# función que divide los datos

def split\_data(df):

X, y = df[['Pregnancies', 'PlasmaGlucose', 'DiastolicBloodPressure', 'TricepsThickness', 'SerumInsulin','BMI', 'DiabetesPedigree', 'Age']].values, df['Diabetic'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=0)

return X\_train, X\_test, y\_train, y\_test

**Probar tu Script**

Antes de usar los scripts en entornos de producción, es importante verificar que funcionen como se espera. Una forma sencilla de probar tu script es ejecutarlo en una terminal.

En el espacio de trabajo de Azure Machine Learning, puedes abrir un script en la página de Notebooks y elegir guardarlo y ejecutarlo en la terminal. Alternativamente, puedes navegar directamente a la terminal de la instancia de computación.

Para ejecutar un script de Python llamado train.py, puedes usar el siguiente comando:

Interfaz de usuario gráfica

Descripción generada automáticamenteLos resultados de las declaraciones de impresión aparecerán en la terminal, y cualquier posible error también se mostrará allí.

Este enfoque asegura que tu código sea más modular, lo que facilita su mantenimiento y pruebas, y prepara tu script para la integración en pipelines de automatización.

-- **Ejecutar un Script como un Trabajo de Comando en Azure Machine Learning**

Cuando tienes un script para entrenar un modelo de aprendizaje automático, puedes ejecutarlo como un trabajo de comando en Azure Machine Learning.

**Configurar y Enviar un Trabajo de Comando**

Para ejecutar un script como un trabajo de comando, debes configurarlo y enviarlo. Al usar el SDK de Python (v2), puedes utilizar la función command para configurar el trabajo. Necesitarás especificar valores para los siguientes parámetros:

* **code**: La carpeta que contiene el script a ejecutar.
* **command**: Especifica qué archivo ejecutar.
* **environment**: Los paquetes necesarios que se instalarán en la instancia de computación antes de ejecutar el comando.
* **compute**: La instancia de computación a utilizar para ejecutar el comando.
* **display\_name**: El nombre del trabajo individual.
* **experiment\_name**: El nombre del experimento al que pertenece el trabajo.

Para configurar un trabajo de comando que ejecute un archivo llamado train.py en el clúster de computación llamado aml-cluster, puedes usar el siguiente código:

Texto

Descripción generada automáticamente

Una vez que tu trabajo esté configurado, puedes enviarlo, lo que iniciará el trabajo y ejecutará el script:

Texto

Descripción generada automáticamente con confianza baja

Puedes monitorear y revisar el trabajo en el estudio de Azure Machine Learning. Todos los trabajos con el mismo nombre de experimento se agruparán bajo el mismo experimento. Puedes encontrar un trabajo individual utilizando el nombre de visualización especificado.

Todos los insumos y salidas de un trabajo de comando se rastrean. Puedes revisar qué comando especificaste, qué instancia de computación se utilizó y qué entorno se empleó para ejecutar el script en la instancia de computación especificada.

-- **Uso de Parámetros en un Trabajo de Comando en Azure Machine Learning**

Puedes aumentar la flexibilidad de tus scripts utilizando parámetros. Esto es especialmente útil si deseas entrenar un modelo de aprendizaje automático en diferentes conjuntos de datos o con varios valores de hiperparámetros.

**Trabajar con Argumentos de Script**

Texto

Descripción generada automáticamentePara usar parámetros en un script, debes utilizar una biblioteca como argparse para leer los argumentos pasados al script y asignarlos a variables. Por ejemplo, el siguiente script lee un argumento llamado training\_data, que especifica la ruta a los datos de entrenamiento:

Texto

Descripción generada automáticamente

Cualquier parámetro que esperes debe estar definido en el script. Puedes especificar qué tipo de valor esperas para cada parámetro y si deseas establecer un valor predeterminado.

**Pasar Argumentos a un Script**

Para pasar valores de parámetro a un script, debes proporcionar el valor del argumento en el comando. Por ejemplo, al ejecutar un script en la terminal, usarías el siguiente comando:

python train.py --training\_data diabetes.csv

En este ejemplo, diabetes.csv es un archivo local. Alternativamente, podrías especificar la ruta a un activo de datos creado en el espacio de trabajo de Azure Machine Learning.

Del mismo modo, cuando deseas pasar un valor de parámetro a un script que quieres ejecutar como un trabajo de comando, especificarías los valores en el comando:

Texto

Descripción generada automáticamente