Guia de instrucción: Modelo de aprendizaje automático en Docker

Isaac Mohamed Laaouaj
Universidad Alfonso X el Sabio
Grado en Computación e Inteligencia Artificial
Curso 2023/24

Actividad 2, Infraestructuras y Servicios en la Nube

Índice

1. Introducción

Construcción y ejecución de un modelo de aprendizaje supervisado muy básico de aprendizaje automático (Machine Learning), a partir de un conjunto de datos (dataset) que representan el carácter (Beningno o Maligno) del tumor cancerígeno detectado en 569 pacientes.

Lo que tratamos de predecir es si el cáncer de los 569 pacientes es Benigno o Maligno, en base a una serie de características definidas en el dataset.

Para ello implementaremos los datos en dos modelos de clasificación mediante la librería scikit-learn, con estas técnicas de aprendizaje automático: Linear Discriminant Analysis y Neural Networks multilayer perceptron.

Finalmente se implementará el modelo en una imagen de docker llamada “docker-ml-model”.

1.1 ¿Qué es Docker?

Docker se trata de un sistema operativo de contenedores, es similar a las máquinas virtuales. Docker permite virtualizar el sistema operativo de un ordenador en “contenedores”.

Dichos contenedores los podemos emplear a la hora de programar y crear aplicaciones, dando la comodidad de implementar, desplegar y ejecutar aplicaciones de manera eficiente y consistente en diferentes entornos.

Cada contenedor es independiente y contiene todo lo necesario para ejecutar una aplicación, incluyendo bibliotecas, dependencias y el propio código.

2. Modelo utilizado en la imagen de Docker

2.1 ¿Qué es el Análisis Discriminante Lineal (LDA)?

El Análisis Discriminante Lineal (LDA) es una técnica estadística utilizada en aprendizaje automático y estadísticas multivariadas.

Su objetivo principal es encontrar la combinación lineal de características que mejor discrimina entre dos o más clases en un conjunto de datos.

2.2 ¿Qué es “Nearest Neighbors (NN)”?

Es un algoritmo de aprendizaje automático que cae en la categoría de métodos de clasificación y regresión basados en instancias. Este enfoque se basa en la premisa de que las instancias similares deben tener etiquetas similares o valores similares.

En el contexto de la clasificación, como en la detección de cáncer mencionada anteriormente, el algoritmo clasifica un nuevo punto de datos basándose en las etiquetas de los puntos de datos más cercanos en el espacio de características.

3. Datos utilizados

Los datos representan el carácter del tumor cancerígeno detectado en 569 pacientes clasificados como benignos (B) o malignos (M). Con estas características que podemos ver gracias a la librería pandas:

```
training = "./data/train.csv"
df_train = pd.read_csv(training)
print(df_train.columns)
```

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')
```

Los datos los hemos dividido en dos, uno destinado al entrenamiento del modelo (train.csv) que conforman el 80% de los datos y otro para realizar pruebas (test.csv) que conforman el 20% de los datos.

Esta distribución de los datos dentro de “train.csv” y “test.csv” se ha hecho de manera aleatoria.

Además tenemos que ajustar nuestro dataset de entrenamiento al modelo de LDA, cambiamos la columna “diagnosis” en valores 0 para los Benignos y 1 en los Malignos:

```
training = "./data/train.csv"
df_train = pd.read_csv(training)
df_train['diagnosis'].replace('M', 1, inplace=True)
df_train['diagnosis'].replace('B', 0, inplace=True)
```

4. Interpretación de los resultados

4.1 LDA (Análisis Discriminante Lineal)

La puntuación LDA es de 0.9701230228471002 es una medida de qué tan bien el modelo LDA se desempeña en el conjunto de datos. Una puntuación cercana a 1 indica un buen rendimiento, por lo tanto tenemos una buena puntuación-

Clasificación LDA: ['M' 'M' 'M' ... 'M' 'B' 'M']. Estas son las predicciones del modelo LDA para cada instancia en el conjunto de datos. 'M' indica una predicción de maligno y 'B' indica una predicción de benigno.

```
LDA score and classification:
0.9701230228471002
['M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'B' 'M' 'M' 'M' 'M'
 'M' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M'
 'M' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M'
 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B'
 'M' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'B'
 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B'
 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B'
 'M' 'M' 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B'
 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M'
 'M' 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B'
 'M' 'M' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'M' 'M'
 'M' 'M' 'B' 'M' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'M' 'M' 'B'
 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'M'
 'B' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B'
 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B'
 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'M'
 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'M' 'B' 'B'
 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B'
 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B'
 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B'
 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M'
 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'M' 'B' 'M'
 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'M'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'M' 'M' 'B']
```

4.2 LDA (Análisis Discriminante Lineal)

Puntuación NN: 0.9226713532513181 se trata de la puntuación de LDA, la puntuación de NN mide qué tan bien el modelo de vecinos más cercanos se desempeña en el conjunto de datos

.

Clasificación NN: ['M' 'M' 'M' ... 'M' 'B' 'B'] son las predicciones del modelo de vecinos más cercanos para cada instancia en el conjunto de datos.

```
NN score and classification:
0.9226713532513181
['M' 'M' 'M' 'B' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'B' 'B' 'M' 'M' 'M'
 'M' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M'
 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M'
 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B'
 'M' 'M' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'B'
 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B'
 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B'
 'M' 'M' 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'M'
 'M' 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'M' 'M' 'B' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'M'
 'M' 'M' 'B' 'M' 'M' 'M' 'M' 'M' 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'B'
 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M'
 'B' 'B' 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B'
 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'M' 'B' 'B' 'B' 'B'
 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B'
 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'M'
 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'M' 'M' 'B'
 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'M' 'B' 'M' 'M' 'B' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'M' 'B' 'B'
 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B'
 'M' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'M' 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'M'
 'B' 'M' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'B'
 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B' 'M' 'B' 'B' 'B' 'B'
 'B' 'M' 'B' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'B' 'M' 'B' 'M'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'B' 'M' 'M' 'B' 'B' 'B' 'M'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'M' 'B' 'M' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B' 'B'
 'B' 'B' 'B' 'B' 'B' 'M' 'M' 'M' 'M' 'M' 'B']
```

5. Instrucciones

5.1 Instrucciones de la construcción de la imagen

Lo hacemos definiendo un “Dockerfile”:

```
# Use base image with mostly all dependencies installed
FROM jupyter/scipy-notebook
LABEL Author="Isaac Mohamed Laaouaj" Email="mlaao@myuax.com"

# Install another dependency using pip
RUN pip install joblib && mkdir models

# Copy data into image
COPY ./data/train.csv ./data/train.csv
COPY ./data/test.csv ./data/test.csv

#Copy model into image
COPY train.py ./train.py
COPY inference.py ./inference.py

# Run training
RUN ["python","train.py"]
```

Ahora mediante el comando docker `docker build -t docker-ml-model -file Dockerfile .` Con esto ya tenemos construida la imagen de docker.

Nos aseguramos que nuestra imagen ha sido creada mediante el comando `docker images`

```
docker-ml-model          latest    4095fde8398b   5 hours ago   4.2GB
```

Ahora le damos a correr mediante el comando `docker run docker-ml-model python “nombre del archivo.py”` y ya tendríamos listo nuestra nuestro docker para ejecutar nuestro modelo.

5.2 Instrucciones del levantamiento e implementación de la imagen “isaac31120/cancer_prediction_docker”

Se puede descargar y aplicar mediante `docker pull isaac31120/cancer_prediction_docker` además esto asegurará que la última versión de la imagen esté disponible localmente en tu sistema.

Y finalmente con este comando puedes ejecutar la imagen de docker en tu ordenador: `docker run --rm -it isaac31120/cancer_prediction_docker`