



**UNIVERSIDAD
DE GRANADA**

BIG DATA I

MÁSTER CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES

CLOUD COMPUTING Y BIG DATA

PRÁCTICA SOBRE CONTENEDORES

Autor

Ignacio Vellido Expósito
ignaciove@correo.ugr.es



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

CURSO 2020-2021

Índice

1. Contenedor para actividades de ciencia de datos basado en R	2
1.1. Descripción	2
1.2. Archivo Dockerfile	2
1.3. Proceso de construcción	3
1.3.1. En hadoop.ugr.es	3
1.3.2. En Azure	4
1.4. Evaluación	7
2. Contenedor para actividades de ciencia de datos basado en Python	9
2.1. Descripción	9
2.2. Archivo Dockerfile	9
2.3. Proceso de construcción	10
2.3.1. En hadoop.ugr.es	10
2.3.2. En Azure	12
2.4. Evaluación	14
3. Contenedor con SGDB MySQL	15
3.1. Descripción	15
3.2. Archivo Dockerfile	15
3.3. Proceso de construcción	15
3.3.1. En local	15
3.3.2. En Azure	16
3.4. Evaluación	20

1. Contenedor para actividades de ciencia de datos basado en R

1.1. Descripción

Contenedor partiendo de una imagen base de Ubuntu al que se le añade R con distintos paquetes de ciencia de datos, concretamente:

- tidyverse
- caret
- RSNNS
- frbs
- FSinR
- forecast

1.2. Archivo Dockerfile

```
1 FROM ubuntu
2 LABEL author="Ignacio Vellido Expósito"
3 ENV http_proxy http://stargate.ugr.es:3128
4
5 # To don't get asked about geographic location (disable dpkg interactivity)
6 ENV DEBIAN_FRONTEND=noninteractive
7
8 # Install R and tidyverse
9 RUN apt-get update && \
10     apt-get install -y r-base \
11         r-cran-tidyverse
12
13 RUN R -e "install.packages(c('caret','RSNNS','frbs','FSinR','forecast'), \
14                             dependencies=TRUE, repos='http://cran.rstudio.com/')"
15
16 # Launch test script
17 COPY testDocker.R /home/testDocker.R
18 RUN cd /home && \
19     Rscript /home/testDocker.R > /home/testOutput.txt
20
21 # Launch R
22 CMD ["R"]
```

Figura 1

El paquete “tidyverse” es necesario instalarlo a través de apt-get para evitar errores. Se incluye el proceso de testeo dentro del dockerfile para agilizar las pruebas, y se concluye indicando el comando por defecto de ejecución del script.

1.3. Proceso de construcción

1.3.1. En hadoop.ugr.es

```

x79056166@hadoop-master:~/cdr$ docker build -t x79056166/cdr .
Sending build context to Docker daemon 4.608kB
Step 1/9 : FROM ubuntu
--> f643c72bc252
Step 2/9 : LABEL author="Ignacio Vellido Expósito"
--> Using cache
--> 7ff8aa28d7ef
Step 3/9 : ENV http_proxy http://stargate.ugr.es:3128
--> Using cache
--> 102733f24444
Step 4/9 : ENV DEBIAN_FRONTEND=noninteractive
--> Using cache
--> d286d2545a08
Step 5/9 : RUN apt-get update && apt-get install -y r-base r-cran-tidyverse
--> Using cache
--> 26af8e09ed55
Step 6/9 : RUN R -e "install.packages(c('caret', 'RSNNS', 'frbs', 'FSinR', 'forecast'), dependencies=TRUE, repos='http://cran.rstudio.com/')"
--> Using cache
--> a86eb51529cb
Step 7/9 : COPY testDocker.R /home/testDocker.R
--> Using cache
--> dc87998e02d4
Step 8/9 : RUN cd /home && Rscript /home/testDocker.R > /home/testOutput.txt
--> Running in 71c60b128894
-- Attaching packages ----- tidyverse 1.3.0 --
v ggplot2 3.2.1 v purrr 0.3.3
v tidbale 2.1.3 v dplyr 0.8.4
v tidyr 1.0.2 v stringr 1.4.0
v readr 1.3.1 v forcats 0.4.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()
Loading required package: Rcpp
Registered S3 method overwritten by 'quantmod':
  method from
as.zoo.data.frame zoo
Loading required package: lattice
Attaching package: 'caret'
The following objects are masked from 'package:RSNNS':
  confusionMatrix, train
The following object is masked from 'package:purrr':
  lift
Saving 7 x 7 in image
Removing intermediate container 71c60b128894
--> a259f0bdc5e6
Step 9/9 : CMD [ "R" ]
--> Running in 2906ca46845f
Removing intermediate container 2906ca46845f
--> 9add484fd46e
Successfully built 9add484fd46e
Successfully tagged x79056166/cdr:latest

```

Figura 2: Construcción de la imagen.

```

x79056166@hadoop-master:~/cdr$ docker run -i -t x79056166/cdr
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

```

Figura 3: Lanzando la imagen.

```
x79056166@hadoop-master:~/cdr$ docker run -i -t x79056166/cdr /bin/bash
root@5ace4be08114:/# ls /home/
Rplots.pdf testDocker.R testGgplot.png testOutput.txt
root@5ace4be08114:/# cat /home/testOutput.txt
k-Nearest Neighbors

150 samples
  4 predictor
  2 classes: 'negative', 'positive'

Pre-processing: centered (4), scaled (4)
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 120, 120, 120, 120, 120, 120, ...
Resampling results across tuning parameters:

  k   ROC      Sens      Spec
  1   0.9400000  0.9600000  0.9200000
  3   0.9701667  0.9666667  0.9266667
  5   0.9895000  0.9766667  0.9400000
  7   0.9923333  0.9766667  0.9066667
  9   0.9918333  0.9800000  0.9066667
  11  0.9908333  0.9733333  0.9133333

ROC was used to select the optimal model using the largest value.
The final value used for the model was k = 7.
```

Figura 4: Contenido de la imagen.

```
x7905616@hadoop-master:~$ ls
CONTAINER_ID IMAGE COMMAND CREATED STATUS PORTS
x7905616@hadoop-master:~$ cd cdython/
608554734907x7905616/cdr "R" 24 seconds ago Up 22 seconds
x7905616@hadoop-master:~$ cd cdython $
0/ftcp quiky_bavering /"docker-entrypoint..." 22 hours ago Up 22 hours
x7905616@hadoop-master:~$ dockerfile requirements.txt testDocker.py
b0d51a5f0bba /"docker-entrypoint..." 22 hours ago Up 22 hours 80/tcp
x7905616@hadoop-master:~$ docker run -i -t x7905616/cdr
9347f3223920 /"docker-entrypoint..." 22 hours ago Up 22 hours 0.0.0.0:8
x7905616@hadoop-master:~$ docker run -i -t x7905616/cdr
b0d51a5f0bba /"docker-entrypoint..." 22 hours ago Up 22 hours 0.0.0.0:8

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

x7905616@hadoop-master:~$ docker ps
CONTAINER_ID IMAGE COMMAND CREATED STATUS PORTS
608554734907x7905616/cdr "R" 24 seconds ago Up 22 seconds
0/ftcp quiky_bavering /"docker-entrypoint..." 22 hours ago Up 22 hours
b0d51a5f0bba /"docker-entrypoint..." 22 hours ago Up 22 hours 80/tcp
9347f3223920 /"docker-entrypoint..." 22 hours ago Up 22 hours 0.0.0.0:8
b0d51a5f0bba /"docker-entrypoint..." 22 hours ago Up 22 hours 0.0.0.0:8
0/ftcp clever_leavitt /"bin/bash" 22 hours ago Up 22 hours
ba237891213d fedora Inspiring_austin /"bin/bash" 22 hours ago Up 22 hours
64c74c50c8bc fedora /"bin/bash" 22 hours ago Up 22 hours
kind_minsky /"bin/bash" 22 hours ago Up 22 hours
29cd732b0508 fedora boring_hodgkin /"bin/bash" 22 hours ago Up 22 hours
5e6d4d12a8e fedora gallant_hodgkin /"bin/bash" 22 hours ago Up 22 hours
31328a3dc12a fedora /"bin/bash" 22 hours ago Up 22 hours
x7905616@hadoop-master:~$
```

Figura 5: Comprobando ejecución.

1.3.2. En Azure

Tal y como está el archivo Dockerfile, si se desplegara en Azure se crearía una “container instance” que cargaría la imagen ejecutando el script, y se cerraría inmediatamente. Para poder acceder al contenedor antes de que se cierre, es necesario lanzar algún servicio que se mantenga en ejecutando/espera. Una opción sería desplegar una app web como Django o Flask, pero por simplicidad lanzaremos el comando indicado en <https://docs.microsoft.com/en-us/azure/container-instances/container-instances-troubleshooting#issues-during-container-group-runtime>

```

1 FROM ubuntu
2 LABEL author="Ignacio Vellido Expósito"
3 # ENV http_proxy http://stargate.ugr.es:3128
4
5 # To don't get asked about geographic location (disable dpkg interactivity)
6 ENV DEBIAN_FRONTEND=noninteractive
7
8 # Install R and tidyverse
9 RUN apt-get update && \
10     apt-get install -y r-base \
11         r-cran-tidyverse
12
13 RUN R -e "install.packages(c('caret','RSNNS','frbs','FSinR','forecast'), \
14     dependencies=TRUE, repos='http://cran.rstudio.com/')"
15
16 # Launch test script
17 COPY testDocker.R /home/testDocker.R
18 RUN cd /home && \
19     Rscript /home/testDocker.R > /home/testOutput.txt
20
21 # For Azure (to keep container alive)
22 CMD [ "tail", "-f", "/dev/null" ]

```

Figura 6: Dockerfile modificado.

Para hacer el despliegue, primeramente se crea un repositorio privado en Docker Hub, y se le cambia el nombre a la imagen de hadoop para adaptarla al repositorio.

```

x79056166@hadoop-master:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID,
ub.docker.com to create one.
Username: ignaciove
Password:
WARNING! Your password will be stored unencrypted in /home/x79056166/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
x79056166@hadoop-master:~$ docker push ignaciove/big_data_i:cdr
The push refers to repository [docker.io/ignaciove/big_data_i]
19c81e3a4cbd: Pushed
6cdcaaf11e6c: Pushed
468c259c5bf8: Pushed
9a63c95bf6c7: Pushed
f6253634dc78: Mounted from danijorq/sghd
9069f84dbbe9: Mounted from danijorq/sghd
bacd3af13903: Mounted from danijorq/sghd
cdr: digest: sha256:193718fa7fd7520e7ce6ba2fa8c79c09283a9693948dea4b293be9ac79d9821a size: 1785

```

Figura 7: Subiendo imagen al repositorio.

TAG			
cdr			
Last pushed 5 minutes ago by ignaciove			
docker pull ignaciove/big_data_i:cdr			
DIGEST	OS/ARCH	LAST PULL	COMPRESSED SIZE
193718fa7fd7	linux/amd64	4 minutes ago	533.23 MB

Figura 8: Imagen en Docker Hub.

Create container instance

Subscription * ⓘ Azure for Students

Resource group * ⓘ Big_Data
[Create new](#)

Container details

Container name * ⓘ cdr ✓

Region * ⓘ (US) East US

Image source * ⓘ
☐ Quickstart images
☐ Azure Container Registry
☒ Docker Hub or other registry

Image type * ⓘ
☐ Public ☒ Private

Image * ⓘ ignaciove/big_data_icdr ✓
 ⓘ If not specified, Docker Hub will be used for the container registry and the latest version of the image will be pulled.

Image registry login server * ⓘ index.docker.io ✓

Image registry user name * ⓘ ignaciove ✓

Image registry password * ⓘ ✓

OS type *
☒ Linux ☐ Windows
 ⓘ This selection must match the OS of the image chosen above.

Size * ⓘ 1 vcpu, 1.5 GiB memory, 0 gpus
[Change size](#)

Figura 9: Desplegando el contenedor en Azure.

✓ Your deployment is complete

Deployment name: Microsoft.ContainerInstances-20210122105732 Start time: 1/22/2021, 11:00:12 AM
 Subscription: Azure for Students Correlation ID: 6852066f-fb96-4972-8f1f-5cdc1adfe3bf
 Resource group: Big_Data

Deployment details (Download)

Resource	Type	Status
✓ cdr	Microsoft.ContainerInstance/containerGroups	OK

Figura 10: Desplegando el contenedor en Azure.

Name	Image	State
cdr	ignaciove/big_data_i:cdr	Running

Events Properties Logs Connect

```

root@SandboxHost-637482940222043830:/# cd /home/
root@SandboxHost-637482940222043830:/home# ls
Rplots.pdf testDocker.R testGgplot.png testOutput.txt
root@SandboxHost-637482940222043830:/home# cat testOutput.txt
k-Nearest Neighbors

150 samples
  4 predictor
  2 classes: 'negative', 'positive'

Pre-processing: centered (4), scaled (4)
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 120, 120, 120, 120, 120, ...
Resampling results across tuning parameters:

   k   ROC      Sens      Spec
1   0.9433333 0.9600000 0.9266667
3   0.9696667 0.9700000 0.9000000
5   0.9870000 0.9733333 0.9000000
7   0.9928333 0.9833333 0.9133333
9   0.9915000 0.9766667 0.9066667
11  0.9916667 0.9800000 0.9066667

```

Figura 11: Contenedor en ejecución.

1.4. Evaluación

Para evaluar el contenedor en ambas plataformas se ha comprobado la salida del scripts. En este script se cargan todas las bibliotecas adicionales instaladas y se aplican operaciones con algunas de ellas.


```

1 # Test libraries
2 library(tidyverse)
3 library(RSNNS)
4 library(frbs)
5 library(FSInR)
6 library(forecast)
7 library(caret)
8
9 # Test ggplot (and tidyverse)
10 ggplot(iris, aes(x=Sepal.Length, y=Petal.Length))
11 ggsave("testGgplot.png")
12
13 # Test caret
14 learn_model <-function(dataset, ctrl, message){
15   model.fit <- caret::train(Class ~ ., data = dataset, method = "knn",
16     |         |         |         |         trControl = ctrl, preProcess = c("center","scale"), metric="ROC",
17     |         |         |         |         tuneGrid = expand.grid(k = c(1,3,5,7,9,11)))
18   model.pred <- predict(model.fit,newdata = dataset)
19   model.cm <- caret::confusionMatrix(model.pred, dataset$Class,positive = "positive")
20   model.probs <- predict(model.fit,newdata = dataset, type="prob")
21
22   return(model.fit)
23 }
24
25 df <- iris
26 df$class <- ifelse(df$Species == "virginica", "positive", "negative") %>% as.factor()
27 df$Species <- NULL
28
29 ctrl <- trainControl(method="repeatedcv",number=5,repeats = 3,
30 |         |         |         |         classProbs=TRUE,summmaryFunction = twoClassSummary)
31 model.raw <- learn_model(df, ctrl, "RAW ")
32
33 print(model.raw)
```

Figura 12: Script de prueba.

2. Contenedor para actividades de ciencia de datos basado en Python

2.1. Descripción

Contenedor partiendo de una imagen base de Ubuntu al que se le añade Python con distintos paquetes de ciencia de datos, concretamente:

- pandas, numpy, scikit-learn
- seaborn, scipy
- matplotlib, xlrd

2.2. Archivo Dockerfile

Para la construcción del archivo Dockerfile se parte de las recomendaciones de https://hub.docker.com/_/python y se adapta para una instalación base de Ubuntu.

```
1 FROM ubuntu
2 LABEL author="Ignacio Vellido Expósito"
3 ENV http_proxy http://stargate.ugr.es:3128
4
5 # To don't get asked about geographic location (disable dpkg interactivity)
6 ENV DEBIAN_FRONTEND=noninteractive
7
8 RUN apt-get update && \
9     apt-get install -y python3 python3-pip
10
11 WORKDIR /usr/src/app
12
13 COPY requirements.txt ./
14
15 RUN pip3 install --upgrade pip && \
16     pip3 install --no-cache-dir -r requirements.txt && \
17     rm requirements.txt
18
19 # Launch test script
20 COPY data /home/data
21 COPY testDocker.py /home/testDocker.py
22 RUN cd /home && \
23     python3 /home/testDocker.py > /home/testOutput.txt
24
25 # Launch Python
26 CMD ["python3"]
```

Figura 13

```
cdpython > requirements.txt
1 pandas
2 scikit-learn
3 seaborn
4 scipy
5 numpy
6 matplotlib
7 xlrd
```

Figura 14: Archivo con los paquetes a instalar.

2.3. Proceso de construcción

2.3.1. En hadoop.ugr.es

```
x79056166@hadoop-master: ~/cdpython
Downloading pip-20.3.3-py2.py3-none-any.whl (1.5 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Not uninstalling pip at /usr/lib/python3/dist-packages, outside environment /usr
    Can't uninstall 'pip'. No files were found to uninstall.
Successfully installed pip-20.3.3
Collecting pandas
  Downloading pandas-1.2.1-cp38-cp38-manylinux1_x86_64.whl (9.7 MB)
Collecting scikit-learn
  Downloading scikit_learn-0.24.1-cp38-cp38-manylinux2010_x86_64.whl (24.9 MB)
Collecting seaborn
  Downloading seaborn-0.11.1-py3-none-any.whl (285 kB)
Collecting scipy
  Downloading scipy-1.6.0-cp38-cp38-manylinux1_x86_64.whl (27.2 MB)
Collecting numpy
  Downloading numpy-1.19.5-cp38-cp38-manylinux2010_x86_64.whl (14.9 MB)
Collecting matplotlib
  Downloading matplotlib-3.3.3-cp38-cp38-manylinux1_x86_64.whl (11.6 MB)
Collecting xlrd
  Downloading xlrd-2.0.1-py2.py3-none-any.whl (96 kB)
Collecting python-dateutil>=2.7.3
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting pytz>=2017.3
  Downloading pytz-2020.5-py2.py3-none-any.whl (510 kB)
Collecting joblib>=0.11
  Downloading joblib-1.0.0-py3-none-any.whl (302 kB)
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-2.1.0-py3-none-any.whl (12 kB)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.3.1-cp38-cp38-manylinux1_x86_64.whl (1.2 MB)
Collecting pillow>=6.2.0
  Downloading Pillow-8.1.0-cp38-cp38-manylinux1_x86_64.whl (2.2 MB)
Collecting cyclo>=0.10
  Downloading cyclo-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting six>=1.5
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: six, python-dateutil, numpy, pytz, pandas, joblib, threadpoolctl, scipy, scikit-learn,
Successfully installed cyclo-0.10.0 joblib-1.0.0 kiwisolver-1.3.1 matplotlib-3.3.3 numpy-1.19.5 pandas-1.2.1 pillow-8
poolctl-2.1.0 xlrd-2.0.1
Removing intermediate container 461265b9d592
--> 27862391ad5c
Step 9/12 : COPY data /home/data
--> f3e12a259f5e
Step 10/12 : COPY testDocker.py /home/testDocker.py
--> 27831d156a43
Step 11/12 : RUN cd /home && python3 /home/testDocker.py > /home/testOutput.txt
--> Running in 219d112adfa7
Removing intermediate container 219d112adfa7
--> a81413558cdf
Step 12/12 : CMD [ "python3" ]
--> Running in fa98956c234e
Removing intermediate container fa98956c234e
--> 8d4efaa4122c
Successfully built 8d4efaa4122c
Successfully tagged x79056166/cdpython:latest
x79056166@hadoop-master:~/cdpython$ docker run -i -t x79056166/cdpython
```

Figura 15: Construcción de la imagen.

```
x79056166@hadoop-master:~/cdpython$ docker run -i -t x79056166/cdpython /bin/bash
root@def12a2da856:/usr/src/app# ls /home/
data figure1.png figure2.png figure3.png figure4.png figure5.png figure6.png testDocker.py testOutput.txt
root@def12a2da856:/usr/src/app#
```

Figura 16: Lanzando la imagen.

```

root@def12a2da856:/usr/src/app# cat /home/testOutput.txt
Datos sin normalizar:
  LBE    LB    AC    FM    UC    ...    Median    Variance    Tendency    E    NSP
1    120.0  120.0  0.0  0.0  0.0  ...    121.0      73.0      1.0  0.0  2.0
2    132.0  132.0  4.0  0.0  4.0  ...    140.0      12.0      0.0  0.0  1.0
3    133.0  133.0  2.0  0.0  5.0  ...    138.0      13.0      0.0  0.0  1.0
4    134.0  134.0  2.0  0.0  6.0  ...    137.0      13.0      1.0  0.0  1.0
5    132.0  132.0  4.0  0.0  5.0  ...    138.0      11.0      1.0  0.0  1.0
...      ...      ...      ...      ...      ...      ...      ...      ...      ...
2122   140.0  140.0  0.0  0.0  6.0  ...    152.0       2.0      0.0  1.0  2.0
2123   140.0  140.0  1.0  0.0  9.0  ...    151.0       3.0      1.0  1.0  2.0
2124   140.0  140.0  1.0  0.0  7.0  ...    152.0       4.0      1.0  1.0  2.0
2125   140.0  140.0  1.0  0.0  9.0  ...    151.0       4.0      1.0  1.0  2.0
2126   142.0  142.0  1.0  1.0  5.0  ...    145.0       1.0      0.0  0.0  1.0

[2126 rows x 25 columns]
-----
Datos normalizados:
  LBE    LB    AC    FM    UC    ...    Mean    Median    Variance    Tendency    E
1    120.0  120.0  0.0  0.0  0.0  ...    137.0    121.0      73.0      1.0  0.0
2    132.0  132.0  4.0  0.0  4.0  ...    136.0    140.0      12.0      0.0  0.0
3    133.0  133.0  2.0  0.0  5.0  ...    135.0    138.0      13.0      0.0  0.0
4    134.0  134.0  2.0  0.0  6.0  ...    134.0    137.0      13.0      1.0  0.0
5    132.0  132.0  4.0  0.0  5.0  ...    136.0    138.0      11.0      1.0  0.0
...      ...      ...      ...      ...      ...      ...      ...      ...      ...
2122   140.0  140.0  0.0  0.0  6.0  ...    150.0    152.0       2.0      0.0  1.0
2123   140.0  140.0  1.0  0.0  9.0  ...    148.0    151.0       3.0      1.0  1.0
2124   140.0  140.0  1.0  0.0  7.0  ...    148.0    152.0       4.0      1.0  1.0
2125   140.0  140.0  1.0  0.0  9.0  ...    147.0    151.0       4.0      1.0  1.0
2126   142.0  142.0  1.0  1.0  5.0  ...    143.0    145.0       1.0      0.0  0.0

[2126 rows x 24 columns]
-----
count    2126.000000    2126.000000    ...    2126.000000    2126.000000
mean     133.303857    133.303857    ...      0.320320      0.033866
std       9.840844      9.840844    ...      0.610829      0.180928
min      106.000000    106.000000    ...     -1.000000      0.000000
25%      126.000000    126.000000    ...      0.000000      0.000000
50%      133.000000    133.000000    ...      0.000000      0.000000
75%      140.000000    140.000000    ...      1.000000      0.000000
max      160.000000    160.000000    ...      1.000000      1.000000

[8 rows x 24 columns]
-----
counts    freqs
categories
N          1655  0.778457
S           295  0.138758
P           176  0.082785
-----
Clasificando con SVM
-----
Mejores hiperparámetros del modelo:
{'C': 100000.0, 'gamma': 0.001, 'kernel': 'poly'}

Mejor score obtenido:
0.8505882352941176

Resultados de la predicción sobre test:
      precision    recall  f1-score   support

   Normal         0.87         0.96         0.91         325
  Suspect         0.86         0.71         0.78          42
 Pathologic        0.53         0.29         0.37          59

```

Figura 17: Contenido de la imagen.

```
tail
x7985616@hadoop-master:~/cdpython$ docker run -i -t x7985616/cdpython
python 3.8.5 (default, Jul 28 2020, 12:59:48)
[GCC 9.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
1e29f1e2c56f	x7985616/cdpython	"python3"	11 seconds ago	Up 11 seconds	

Figura 18: Comprobando ejecución.

2.3.2. En Azure

Los pasos para desplegar la imagen en Azure son los mismos que con el contenedor de R. Es necesario modificar el Dockerfile y subirlo al repositorio, e indicarle a Azure dónde encontrar la imagen.

```
1 FROM ubuntu
2 LABEL author="Ignacio Vellido Expósito"
3 # ENV http_proxy http://stargate.ugr.es:3128
4
5 # To don't get asked about geographic location (disable dpkg interactivity)
6 ENV DEBIAN_FRONTEND=noninteractive
7
8 RUN apt-get update && \
9     apt-get install -y python3 python3-pip
10
11 WORKDIR /usr/src/app
12
13 COPY requirements.txt ./
14
15 RUN pip3 install --upgrade pip && \
16     pip3 install --no-cache-dir -r requirements.txt && \
17     rm requirements.txt
18
19 # Launch test script
20 COPY data /home/data
21 COPY testDocker.py /home/testDocker.py
22 RUN cd /home && \
23     python3 /home/testDocker.py > /home/testOutput.txt
24
25
26 # For Azure (to keep container alive)
27 CMD [ "tail", "-f", "/dev/null" ]
```

Figura 19: Dockerfile desplegado en Azure.

Create container instance

Subscription *

Azure for Students

Resource group *

Big_Data

Create new

Container details

Container name *

cdpython

Region *

(US) East US

Image source *

Quickstart images

Azure Container Registry

☒ Docker Hub or other registry

Image type *

Public

☒ Private

Image *

ignaciove/big_data_icdpthon

If not specified, Docker Hub will be used for the container registry and the latest version of the image will be pulled.

Image registry login server *

index.docker.io

Image registry user name *

ignaciove

Image registry password *

OS type *

☒ Linux

☐ Windows

This selection must match the OS of the image chosen above.

Size *

1 vcpu, 1.5 GiB memory, 0 gpus

Change size

Figura 20: Despliegue del contenedor en Azure.

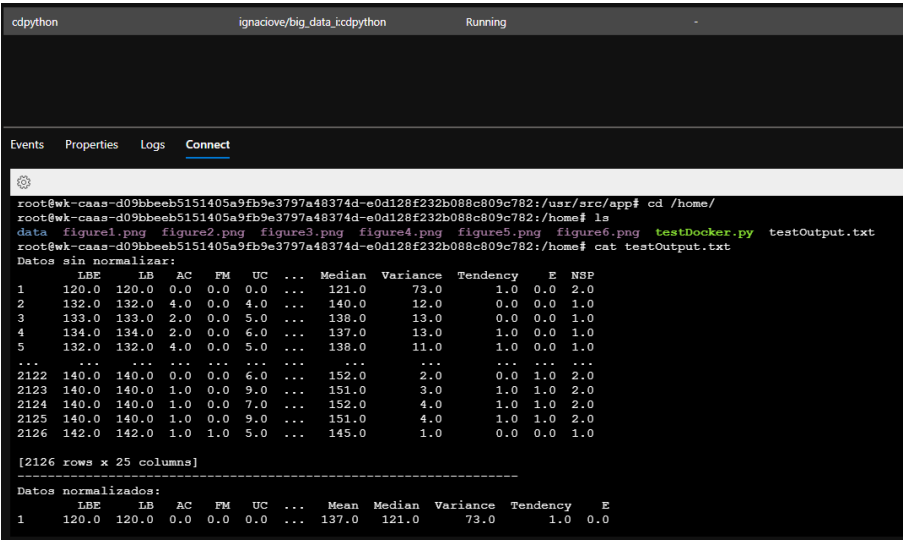


Figura 21: Imagen del contenedor en ejecución.

2.4. Evaluación

Para evaluar el correcto funcionamiento se lanza el siguiente script, que carga los paquetes instalados y realiza un aprendizaje sobre un conjunto de datos con SVM.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  #####
4  # Librerías
5  #####
6
7  import random
8
9  import pandas as pd
10 import numpy as np
11 import matplotlib.pyplot as plt
12 import seaborn as sns
13
14 # Preprocesamiento
15 from sklearn.preprocessing import Normalizer
16 from sklearn.model_selection import train_test_split
17 from sklearn.decomposition import PCA
18
19 # Algoritmos
20 from sklearn.svm import SVC
21
22 from sklearn.model_selection import GridSearchCV
23 from sklearn.model_selection import KFold
24
25 # Evaluación
26 from sklearn.metrics import classification_report, \
27     confusion_matrix, \
28     plot_confusion_matrix
29
30 #####
31 # Lectura
32 #####
33
34 # Semilla con la que se han analizado los resultados
35 random.seed(9999)
36
37 # Cargamos los datos (sheet Raw Data nos es más cómodo que Data)
38 data = pd.read_excel("data/CTG.xls", "Raw Data")
39
40 # Eliminamos las 3 últimas filas que solo contienen valores de máximos y mínimos
41 data = data[:-3]
42
43 # Eliminamos la primera fila que está vacía
44 data = data[1:]
45
46 # Eliminamos las columnas que no contienen información relevante para la
47 # clasificación
48 removed_columns = ["FileName", "Date", "SegFile", "b", "e", "A", "B", "C", "D",
49     "AD", "DE", "LD", "FS", "SUSP", "CLASS"]

```

Figura 22: Parte del script de prueba.

3. Contenedor con SGDB MySQL

3.1. Descripción

Contenedor docker partiendo de una instalación base de MySQL, permitiendo consultas desde el exterior del contenedor.

3.2. Archivo Dockerfile

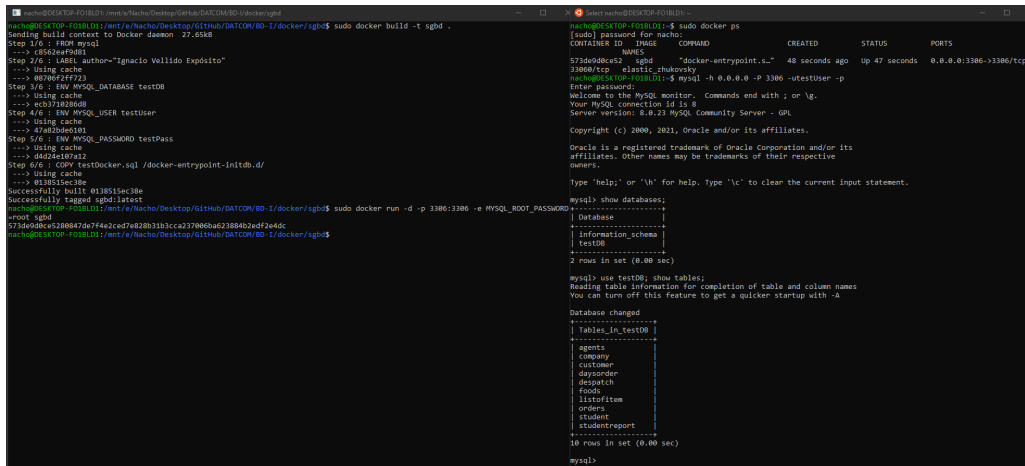
```
1 FROM mysql
2 LABEL author="Ignacio Vellido Expósito"
3
4 # Upload test DB
5 ENV MYSQL_DATABASE testDB
6 ENV MYSQL_USER testUser
7 ENV MYSQL_PASSWORD testPass
8
9 # All scripts in docker-entrypoint-initdb.d/ are automatically
10 # executed during container startup
11 COPY testDocker.sql /docker-entrypoint-initdb.d/
```

Figura 23

Creamos en el archivo Dockerfile un usuario para realizar las pruebas e incluimos una base de datos a partir de un script.

3.3. Proceso de construcción

3.3.1. En local



```
Step 1/6: FROM mysql
Step 2/6: LABEL author="Ignacio Vellido Expósito"
Step 3/6: ENV MYSQL_DATABASE testDB
Step 4/6: ENV MYSQL_USER testUser
Step 5/6: ENV MYSQL_PASSWORD testPass
Step 6/6: COPY testDocker.sql /docker-entrypoint-initdb.d/
Successfully built 018515c38e
Successfully tagged sgdb:latest
nacho@SGDB: /mnt/e/Nacho/Desktop/GitHub/DATCOM/BD-1/docker/sgdb$ sudo docker run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=sgdb sgdb
573d9d8ce5288847d7f462ced7e28b31b3cca237809b6b23884b2ed2e4dc
nacho@SGDB: /mnt/e/Nacho/Desktop/GitHub/DATCOM/BD-1/docker/sgdb$
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| testDB   |
+-----+
2 rows in set (0.00 sec)

mysql> use testDB; show tables;
+-----+
| Tables_in_testDB |
+-----+
| agents            |
| company           |
| customer          |
| daysorder         |
| despatch          |
| food              |
| listofitem        |
| orders            |
| student           |
| studentreport     |
+-----+
10 rows in set (0.00 sec)

mysql>
```

Figura 24: Construcción del contenedor y ejecución desde fuera de él.


```
nacho@DESKTOP-F01BLD1:~$ mysql -h 0.0.0.0 -P 3306 -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sys       |
| testDB    |
+-----+
5 rows in set (0.00 sec)
```

Figura 25: Entrando como usuario root.

3.3.2. En Azure

Una vez más para desplegar el contenedor en Azure subimos la imagen al repositorio y creamos una nueva “container instance”.

Para este contenedor es necesario indicarle a Azure el puerto exportado en la imagen, y modificar el Dockerfile para incluir la contraseña del root (puesto que no se le puede indicar a Azure en el momento de ejecución).

```
1 FROM mysql
2 LABEL author="Ignacio Vellido Expósito"
3
4 # Upload test DB
5 ENV MYSQL_DATABASE testDB
6 ENV MYSQL_USER testUser
7 ENV MYSQL_PASSWORD testPass
8 ENV MYSQL_ROOT_PASSWORD root
9
10 # All scripts in docker-entrypoint-initdb.d/ are automatically
11 # executed during container startup
12 COPY testDocker.sql /docker-entrypoint-initdb.d/
```

Figura 26: Dockerfile modificado.

Create container instance

Subscription * ⓘ Azure for Students

Resource group * ⓘ Big_Data
[Create new](#)

Container details

Container name * ⓘ sgbd ✓

Region * ⓘ (US) East US

Image source * ⓘ
☐ Quickstart images
☐ Azure Container Registry
☒ Docker Hub or other registry

Image type * ⓘ
☐ Public ☒ Private

Image * ⓘ ignaciove/big_data_isgbd ✓
! If not specified, Docker Hub will be used for the container registry and the latest version of the image will be pulled.

Image registry login server * ⓘ index.docker.io ✓

Image registry user name * ⓘ ignaciove ✓

Image registry password * ⓘ ✓

OS type *
☒ Linux ☐ Windows
! This selection must match the OS of the image chosen above.

Size * ⓘ 1 vcpu, 1.5 GiB memory, 0 gpus
[Change size](#)

Figura 27: Creando contenedor.

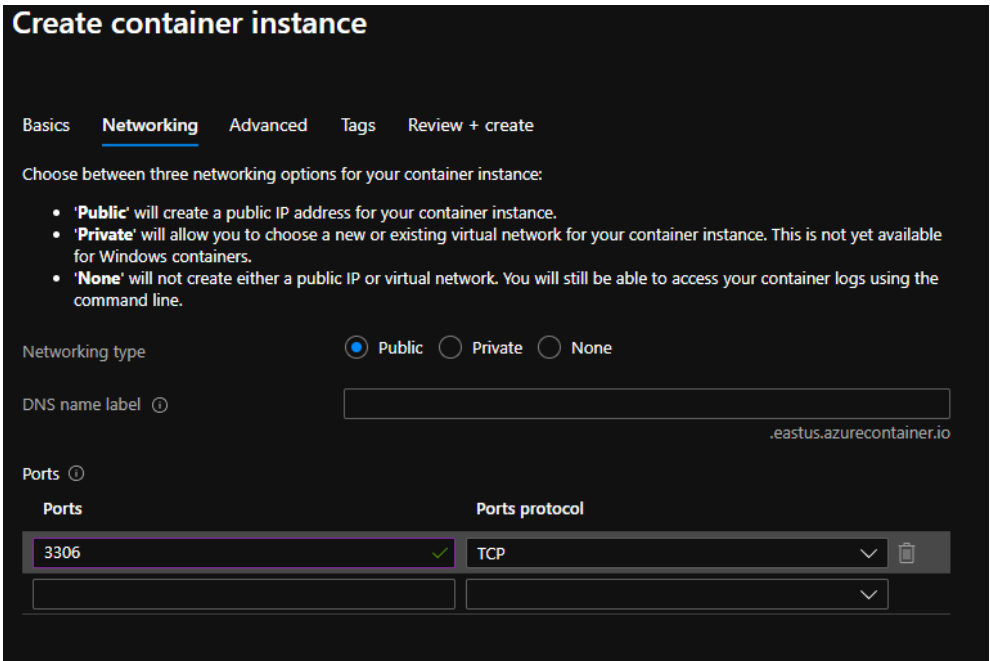


Figura 28: Indicando el puerto exportado.

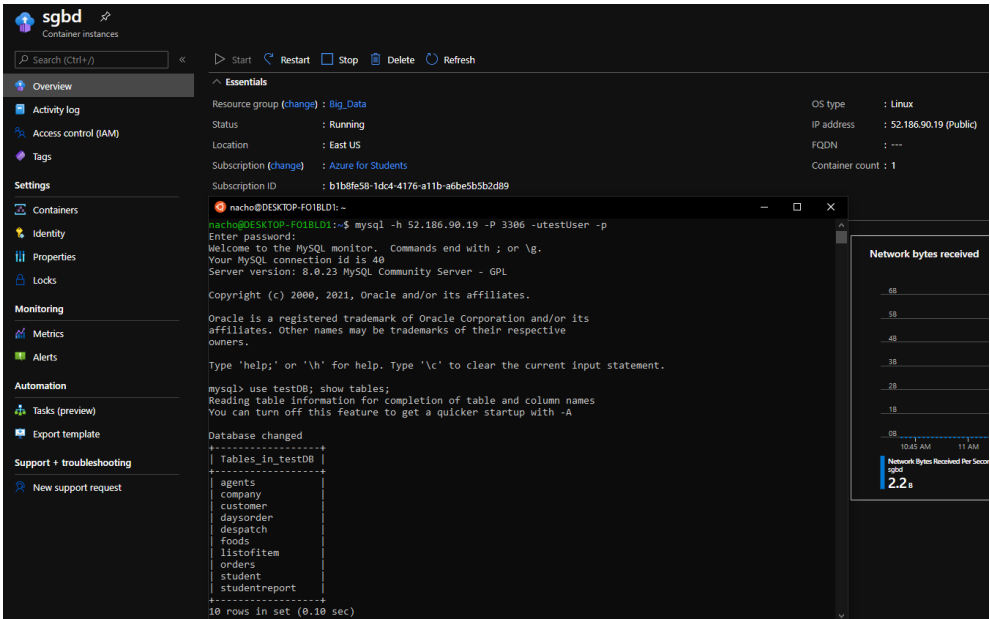


Figura 29: Accediendo a través de la IP pública.

```

sgbd                                ignaciove/big_data_i:sgbd        Running

Events  Properties  Logs  Connect

2021-02-08 10:38:19+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.23-1debian10
2021-02-08 10:38:20+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2021-02-08 10:38:20+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.23-1debian10
2021-02-08 10:38:20+00:00 [Note] [Entrypoint]: Initializing database files
2021-02-08T10:38:20.148276Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.23) in
2021-02-08T10:38:20.154010Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2021-02-08T10:38:24.490190Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2021-02-08T10:38:30.948045Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an
2021-02-08 10:38:41+00:00 [Note] [Entrypoint]: Database files initialized
2021-02-08 10:38:41+00:00 [Note] [Entrypoint]: Starting temporary server
2021-02-08T10:38:41.454955Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.23) st
2021-02-08T10:38:41.520124Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2021-02-08T10:38:42.078165Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2021-02-08T10:38:42.191859Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. So
2021-02-08T10:38:42.481485Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self sign
2021-02-08T10:38:42.481824Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to st
2021-02-08T10:38:42.492359Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-t
2021-02-08T10:38:42.508649Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connect
2021-02-08 10:38:42+00:00 [Note] [Entrypoint]: Temporary server started.
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leap-seconds.list' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
2021-02-08 10:38:45+00:00 [Note] [Entrypoint]: Creating database testDB
2021-02-08 10:38:45+00:00 [Note] [Entrypoint]: Creating user testUser
2021-02-08 10:38:45+00:00 [Note] [Entrypoint]: Giving user testUser access to schema testDB

```

Figura 30: Logs del contenedor.

3.4. Evaluación

La evaluación únicamente se basa en poder acceder al usuario creado en el Dockerfile, corroborando la correcta inclusión de la base de datos de prueba.

```
--
-- Database: `sample`
--
--
-- -----
--
-- Table structure for table `agents`
--
CREATE TABLE IF NOT EXISTS `agents` (
  `AGENT_CODE` varchar(6) NOT NULL DEFAULT '',
  `AGENT_NAME` varchar(40) DEFAULT NULL,
  `WORKING_AREA` varchar(35) DEFAULT NULL,
  `COMMISSION` decimal(10,2) DEFAULT NULL,
  `PHONE_NO` varchar(15) DEFAULT NULL,
  `COUNTRY` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`AGENT_CODE`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
--
-- Dumping data for table `agents`
--
INSERT INTO `agents` (`AGENT_CODE`, `AGENT_NAME`, `WORKING_AREA`, `COMMISSION`, `PHONE_NO`, `COUNTRY`) VALUES
('A007', 'Ramasundar', 'Bangalore', '0.15', '077-25814763', '\r'),
('A003', 'Alex', 'London', '0.13', '075-12458969', '\r'),
('A008', 'Alford', 'New York', '0.12', '044-25874365', '\r'),
('A011', 'Ravi Kumar', 'Bangalore', '0.15', '077-45625874', '\r'),
('A010', 'Santakumar', 'Chennai', '0.14', '007-22388644', '\r'),
('A012', 'Lucida', 'San Jose', '0.12', '044-52981425', '\r'),
('A005', 'Anderson', 'Brisban', '0.13', '045-21447739', '\r'),
('A001', 'Subbarao', 'Bangalore', '0.14', '077-12346674', '\r'),
('A002', 'Mukesh', 'Mumbai', '0.11', '029-12358964', '\r'),
('A006', 'McDen', 'London', '0.15', '078-22255588', '\r'),
('A004', 'Ivan', 'Toronto', '0.15', '008-22544166', '\r'),
('A009', 'Benjamin', 'Hampshair', '0.11', '008-22536178', '\r');
```

Figura 31: Fragmento de la base de datos insertada.