

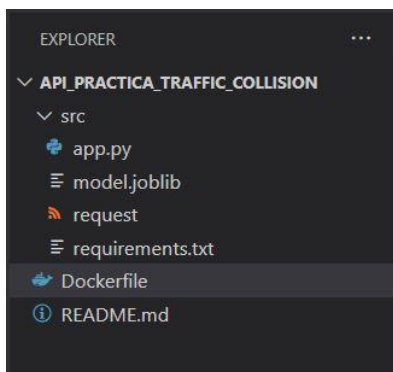
DETALLE CONFIGURACIÓN ENTORNO DOCKER Y API

INTRODUCCIÓN

Para la creación puesta en producción del modelo de machine learning generado a partir del dataset de *Canada Car Accidents 1999-2014* ([Kaggle](#)), se han utilizado los siguientes archivos básicos:

- **Dockerfile:** donde se define la imagen del contenedor Docker y se especifican los comandos que nos permiten crear el entorno de ejecución de la aplicación.
- **App.py:** en este archivo se desarrolla y define el funcionamiento de la aplicación.
- **Requirements.txt:** en él se detallan las librerías que se requieren para el funcionamiento de la aplicación.

Para el correcto funcionamiento de la aplicación y el establecimiento de correspondencias entre los distintos archivos que componen el proyecto, se deben agrupar en un directorio junto con el resto de archivos, que contienen el modelo generado, además de un archivo auxiliar que permite visualizar más fácilmente las consultas que se van a lanzar. Este directorio se muestra en la siguiente imagen:



DETALLE DE LOS ARCHIVOS DE LA APLICACIÓN

A. DOCKERFILE

```
Dockerfile
1  FROM continuumio/anaconda3
2
3  COPY src/requirements.txt /tmp/
4  RUN pip install -r /tmp/requirements.txt
5
6  RUN mkdir /app
7  COPY src /app
8  RUN chmod -R +rwx /app
9
10 WORKDIR /app
11
12 EXPOSE 5000
13
14 CMD ["python", "app.py"]
15
```

Trabajar en un entorno de ejecución dentro de una imagen de un contenedor Docker nos permite como principal ventaja la capacidad de ser ejecutado en cualquier servidor, asegurándonos de su correcto funcionamiento.

Para la creación de la imagen, se ha tomado como referencia la imagen base continuumio/anaconda3, obtenido del sitio web Docker Hub, el cual contiene una distribución potente y flexible de Python que se adecúa al propósito de la aplicación.

Asimismo, se ha especificado el archivo con los requerimientos de librerías de la aplicación 'requirements.txt', lo cual permitirá instalar las mismas, así como el archivo de la aplicación propiamente dicha, donde se procesarán las llamadas a la api y se estimarán las predicciones del modelo correspondientes.

B. REQUIREMENTS

```
src > requirements.txt
1 flask
2 pandas==1.3.4
3 scikit-learn==1.0.1
4 category-encoders==2.3.0
5 lightgbm==3.3.1
6
```

En este archivo se especifican las librerías necesarias. Asimismo, se han especificado las versiones utilizadas en el desarrollo del modelo, para asegurar el correcto funcionamiento de la aplicación en todo momento.

C. APP

Como se ha venido comentando, este archivo recoge el código de desarrollo de la aplicación.

```
1 import flask
2 from joblib import load
3 import pandas as pd
4 import json
5 from werkzeug.exceptions import BadRequest
6
7 app = flask.Flask(__name__)
8
9
10 @app.route('/', methods=['GET'])
11 def home():
12     return ""
13
14 @app.errorhandler(TypeError)
15 def handle_unexpected_error(error):
16     status_code = 400
17     response = {
18         'error': {
19             'type': 'Bad request',
20             'message': 'Parametro invalido los parametros validos son ['C_MTH,C_MDAY,C_HOUR,C_VHS,C_CONF,C_RCFG,C_MTHR,C_RSUR,C_BAUN,C_TRAF,V_TYPE_1,V_TYPE_10,V_TYPE_11,V_TYPE_14,V_TYPE_16,V_TYPE_17,V_T'
21         ]
22     }
23     return json.dumps(response), status_code
24
25
26 @app.errorhandler(BadRequest)
27 def handle_unexpected_error(error):
28     status_code = 400
29     response = {
30         'error': {
31             'type': 'Bad request',
32             'message': error.description
33         }
34     }
35     return json.dumps(response), status_code
36
37
38 @app.route('/predict', methods=['GET'])
39 def predict():
40     input = dict(flask.request.args)
41
42     validations = {
43         'C_MTH': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
44         'C_MDAY': [1, 2, 3, 4, 5, 6, 7],
45         'C_HOUR': [0, 1, 2, 3],
46         'C_VHS': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42],
47         'C_CONF': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
48         'C_RCFG': [1, 2, 3, 4, 5, 6, 7, 8],
49         'C_MTHR': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
50         'C_RSUR': [1, 2, 3, 4, 5, 6, 7],
51         'C_BAUN': [1, 2, 3, 4, 5, 6, 7],
52         'C_TRAF': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
53     }
54     for key,value in input.items():
55         input[key] = [value]
56         if key in validations:
57             if int(value) not in validations[key]:
58                 message = f'El valor de {key} debe estar entre {validations[key]}'
59                 raise BadRequest(description=message)
60
61     default_validation = ['C_MTH','C_MDAY','C_HOUR','C_VHS','C_CONF','C_RCFG','C_MTHR','C_RSUR','C_BAUN','C_TRAF','V_TYPE_1','V_TYPE_10','V_TYPE_11','V_TYPE_14','V_TYPE_16','V_TYPE_17','V_T'
62     defaults = list(set(default_validation) - set(input.keys()))
63     for key in defaults:
64         input[key] = [None]
65
66     print(input,flush=True)
67     input_df = pd.DataFrame(input)
68     model = load('./model.joblib')
69     threshold = 0.85055
70
71     output = {
72         'prediction': 1 if model.predict_proba(input_df)[0][1] > threshold else 0,
73         'probabilidad fatalidad': float(model.predict_proba(input_df)[0][1]),
74         'probabilidad no fatalidad': float(model.predict_proba(input_df)[0][0])
75     }
76     return output
77
78 app.run(host='0.0.0.0', port=5000)
```

Se basa en la librería flask, la cual permite crear la aplicación propiamente dicha. Debido al propósito último de consulta de la misma, se ha definido únicamente el método GET, limitando la acción del usuario a la extracción de información.

Asimismo, se han definido en primer lugar dos validaciones sobre los datos introducidos en la llamada a la api, para establecer dos posibles errores en los datos de entrada. Una para indicar que alguna de las features introducidas no se corresponden con las que se encuentran recogidas en el modelo, y otra que en el caso de las variables categóricas, indica al usuario que ha introducido un valor que no se corresponde con el rango de valores de la feature en cuestión.

Además, se ha definido una tercera validación que se encarga de establecer como 'None' los valores de aquellas variables que no se han introducido en la consulta, evitando problemas en el funcionamiento de la aplicación por falta de datos.

Por último, se carga el modelo y se realiza la predicción con los datos de la consulta, estableciéndose como output la predicción final del modelo, tomando como referencia el threshold óptimo detectado en la construcción del modelo, así como el detalle de la probabilidad que asigna el modelo a la existencia o no existencia de fatalidad en el accidente.

CREACIÓN Y PUBLICACIÓN DE LA IMAGEN

Una vez definida, se ha construido la imagen con la aplicación dentro, mediante el comando **docker build -t mlapi**.

```
(base) C:\Users\Antonio\Documents\Antonio_CUNEF\05_Machine_Learning\API_practica_traffic_collision>docker build -t mlapi .
[+] Building 3.8s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/continuumio/anaconda3:latest
=> [auth] continuumio/anaconda3:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 4.14kB
=> [1/7] FROM docker.io/continuumio/anaconda3@sha256:fd81feb32b49c5381178bf87fef31c6c17ae8114ab7cf95f9511b610a5f18e3
=> CACHED [2/7] COPY src/requirements.txt /tmp/
=> CACHED [3/7] RUN pip install -r /tmp/requirements.txt
=> CACHED [4/7] RUN mkdir /app
=> [5/7] COPY src /app
=> [6/7] RUN chmod -R +rwx /app
=> [7/7] WORKDIR /app
=> exporting to image
=> => exporting layers
=> => writing image sha256:8742b303b1fea20ef328bc59f3d4034ca0e991b2b08626dc1feb2ad5bf743283
=> => naming to docker.io/library/mlapi
```

Para ejecutar al contenedor con la imagen anteriormente creada, se deberá utilizar el comando **docker run -p 5000:5000 mlapi**.

Asimismo, se ha subido la imagen al repositorio dentro de Docker Hub para que sea accesible por cualquier usuario.

```
(base) C:\Users\Antonio\Documents\Antonio_CUNEF\05_Machine_Learning\API_practica_traffic_collision>docker tag mlapi abarred/api_traffic_collision
(base) C:\Users\Antonio\Documents\Antonio_CUNEF\05_Machine_Learning\API_practica_traffic_collision>docker push abarred/api_traffic_collision
Using default tag: latest
The push refers to repository [docker.io/abarred/api_traffic_collision]
5f70bf18a086: Mounted from lhansa/cunefark
6613758afabf: Pushed
81335f87251e: Pushed
fd71583fae31: Pushed
810af0c643be: Pushed
a4fe4c7e64f7: Pushed
189f6ce7a874: Mounted from continuumio/anaconda3
9321ff862abb: Mounted from continuumio/anaconda3
latest: digest: sha256:3526230eaa0383e3df3a894aa4044b5a9a3ca4ad7c97be0f6260f4d03a65ac96 size: 1996
```

Para ello, utilizamos los comandos `docker tag mlapi abarred/api_traffic_collision` y posteriormente `docker push abarred/api_traffic_collision`.

REALIZACIÓN DE LAS CONSULTAS

Para llevar a cabo las llamadas a la api, además de mediante el terminal, se ha utilizado el servicio Postman que ofrece una interfaz amigable tanto a la hora de realizar la consulta como a la hora de visualizar los resultados de la misma.

Para comprobar el correcto funcionamiento de la aplicación se han llevado a cabo las siguientes consultas:

- Una consulta correcta con todos los campos completos

The screenshot shows a Postman interface with a GET request to `http://localhost:5000/predict?C_MNTH=11&C_WDAY=3&C_HOUR=1&C_VEHS=1&C_CONF=4&C_RCFG=1&C_WTHR=2&C_RSUR=5&C_RALN=2&C_TRAF=18&V_TYPE_1=0&V_TYPE_10=0&V_TYPE_11=0&V_TYPE_14=0&V_TYPE_16=0&V_TYPE_17=0&V_TYPE_18=0&V_TYPE_19=0&V_TYPE_20=0&V_TYPE_21=0&V_TYPE_22=0&V_TYPE_23=0&V_TYPE_24=0&V_TYPE_5=0&V_TYPE_6=1&V_TYPE_7=0&V_TYPE_8=0&V_TYPE_9=0&P_SEX_F=1&P_SEX_M=0&P_SAFE=1&P_AGE=48`. The query parameters are listed in a table below:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> C_MNTH	11	
<input checked="" type="checkbox"/> C_WDAY	3	
<input checked="" type="checkbox"/> C_HOUR	1	
<input checked="" type="checkbox"/> C_VEHS	1	
<input checked="" type="checkbox"/> C_CONF	4	
<input checked="" type="checkbox"/> C_RCFG	1	
<input checked="" type="checkbox"/> C_WTHR	2	
<input checked="" type="checkbox"/> C_RSUR	5	
<input checked="" type="checkbox"/> C_RALN	2	
<input checked="" type="checkbox"/> C_TRAF	18	

The response body is shown in JSON format:

```
{
  "prediccion": 0,
  "probabilidad_fatalidad": 0.628142275654321224,
  "probabilidad_no_fatalidad": 0.9718577243456787
}
```

- Una consulta correcta sin incluir alguno de los campos

The screenshot shows a REST client interface with a GET request to `http://localhost:5000/predict?C_MNTH=11&C_WDAY=3&C_HOUR=1&C_CONF=4&C_RCFG=1&C_WTHR=2&C_RSUR=5&C_RALN=2&C_TRAF=18&V_TYPE_1=3&V_TYPE_10=0&V_TYPE_11=0&V_TYPE_14=0&V_TYPE_16=0&V_TYPE_17=0&V_TYPE_20=0&V_TYPE_21=0&V_TYPE_22=0&V_TYPE_23=0&V_TYPE_24=0&V_TYPE_5=0&V_TYPE_6=1&V_TYPE_7=0&V_TYPE_8=0&V_TYPE_9=0&P_SEX_F=1&P_SEX_M=1&P_SAFE=0.23`. The response is a JSON object with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> C_MNTH	11	
<input checked="" type="checkbox"/> C_WDAY	3	
<input checked="" type="checkbox"/> C_HOUR	1	
<input checked="" type="checkbox"/> C_CONF	4	
<input checked="" type="checkbox"/> C_RCFG	1	
<input checked="" type="checkbox"/> C_WTHR	2	
<input checked="" type="checkbox"/> C_RSUR	5	
<input checked="" type="checkbox"/> C_RALN	2	
<input checked="" type="checkbox"/> C_TRAF	18	
<input checked="" type="checkbox"/> V_TYPE_1	3	

The response body is displayed in JSON format:

```
1 {
2   "prediccion": 1,
3   "probabilidad_fatality": 0.3913654174903361,
4   "probabilidad_no_fatality": 0.6086345825096639
5 }
```

- Una consulta incorrecta introduciendo un valor que no se encuentra dentro de los rangos de las variables categóricas

The screenshot shows a REST client interface with a GET request to `http://localhost:5000/predict?C_MNTH=0&C_WDAY=3&C_HOUR=1&C_VEHS=1&C_CONF=4&C_RCFG=1&C_WTHR=2&C_RSUR=5&C_RALN=2&C_TRAF=18&V_TYPE_1=0&V_TYPE_10=0&V_TYPE_11=0&V_TYPE_14=0&V_TYPE_16=0&V_TYPE_20=0&V_TYPE_21=0&V_TYPE_22=0&V_TYPE_23=0&V_TYPE_24=0&V_TYPE_5=0&V_TYPE_6=1&V_TYPE_7=0&V_TYPE_8=0&V_TYPE_9=0&P_SEX_F=1&P_SEX_M=0&P_SAFE=1&P_AGE=48`. The response is a JSON object with an error message:

```
1 {
2   "error": {
3     "type": "Bad request",
4     "message": "El valor de C_MNTH debe estar entre [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]"
5   }
6 }
```

- Una consulta incorrecta introduciendo de manera errónea el nombre de una de las variables

Overview GET http://localhost:5000/... Release Notes + ... No Environment

http://localhost:5000/predict?C_MNTH=11&C_WAY=3&C_HOUR=1&C_VEHS=1&C_CONF=4&C_RCFG=1&C_WTHR=2&C_RSUR=5&C_RALN=2&C_TRAF=18&V_TYPE_1=0&V_TYPE_10=0&V_TYPE_11=0&V_TYPE_14=0&V_TYPE_16=0&V_TYPE_...

GET http://localhost:5000/predict?C_MNTH=11&C_WAY=3&C_HOUR=1&C_VEHS=1&C_CONF=4&C_RCFG=1&C_WTHR=2&C_RSUR=5&C_RALN=2&C_TRAF=18&V_TYPE_1=0&V_TYPE_10=0&V_TYPE_11=0&V_TYPE_14=0&V_TYPE_16=0&V_TYPE_17=0&V_TYPE_18=0&V_TYPE_19=0&V_TYPE_20=0&V_TYPE_21=0&V_TYPE_22=0&V_TYPE_23=0&V_TYPE_24=0&V_TYPE_5=0&V_TYPE_6=1&V_TYPE_7=0&V_TYPE_9=0&P_SEX_F=1&P_SEX_M=0&P_SAFE=1&P_AGE=4

Params Auth Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> C_MNTH	11			
<input checked="" type="checkbox"/> C_WAY	3			
<input checked="" type="checkbox"/> C_HOUR	1			
<input checked="" type="checkbox"/> C_VEHS	1			
<input checked="" type="checkbox"/> C_CONF	4			
<input checked="" type="checkbox"/> C_RCFG	1			
<input checked="" type="checkbox"/> C_WTHR	2			
<input checked="" type="checkbox"/> C_RSUR	5			
<input checked="" type="checkbox"/> C_RALN	2			
<input checked="" type="checkbox"/> C_TRAF	18			
<input checked="" type="checkbox"/> V_TYPE_1	0			

Body Cookies Headers (4) Test Results Status: 400 BAD REQUEST Time: 27 ms Size: 539 B Save Response

Pretty Raw Preview Visualize HTML

```

1 {"error": {"type": "Bad request", "message": "Parametro inválido los parametros validos son
2 [C_MNTH,C_WAY,C_HOUR,C_VEHS,C_CONF,C_RCFG,C_WTHR,C_RSUR,C_RALN,C_TRAF,V_TYPE_1,V_TYPE_10,V_TYPE_11,V_TYPE_14,V_TYPE_16,V_TYPE_17,V_TYPE_18,V_TYPE_19,V_TYPE_20,V_TYPE_21,V_TYPE_22,V_TYPE_23,V_TYPE_24,V_TYPE_5,
V_TYPE_6,V_TYPE_7,V_TYPE_8,V_TYPE_9,P_SEX_F,P_SEX_M,P_SAFE,P_AGE,A_VAGE]"}

```