INDUSTRIALIZACIÓN: API DE FLASK

Eva Arribas

OBJETIVO

Programar una API que permita obtener los datos de mi modelo sobre Attrition además de las predicciones.

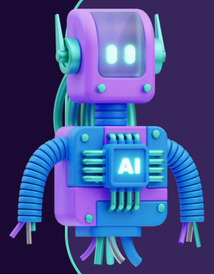
PASOS PREVIOS

Carga de datos del modelo

Cargamos los datos de nuestro modelo sobre el Attrition.

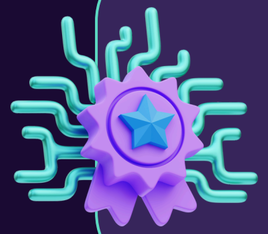
Preprocesamiento de los datos

Unimos los datasets con un merge, eliminamos los valores nulos



Preparación para el modelo

Eliminamos la columna con los datos a predecir y dividimos entre train y test (test_size=0.2, random_state = 42), hacemos el fit y guardamos el modelo en un pickle.

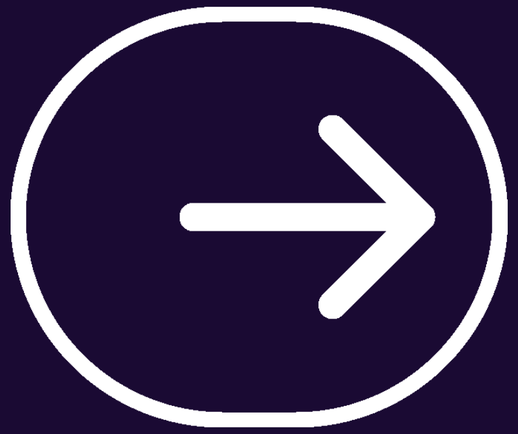


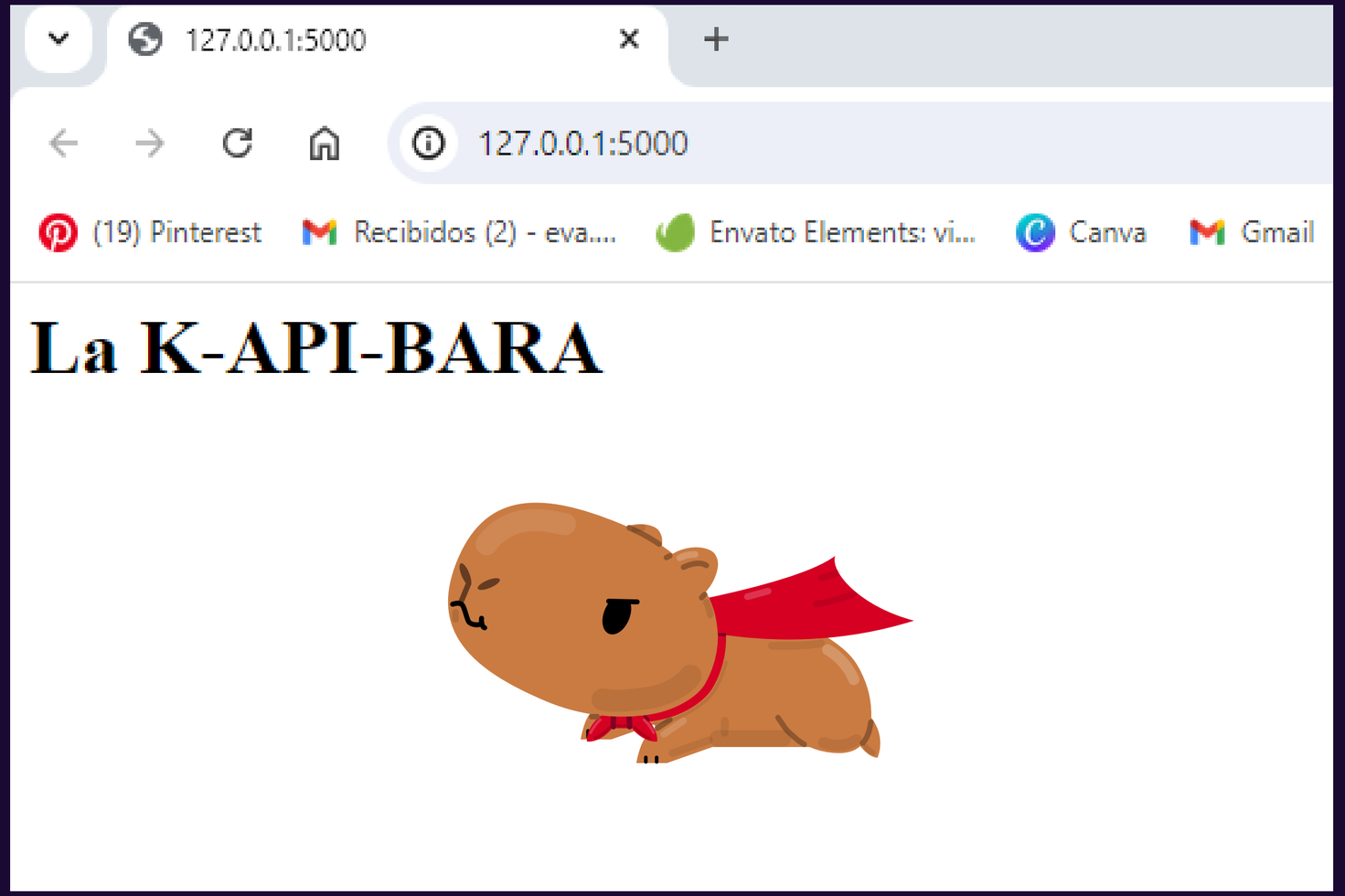
Datos train y test

Guardamos los datos de entrenamiento y de prueba en un json.

API DEFINIDA

```
    app.run(port=5000)
 2.7s
 * Serving Flask app '__main__'
 * Debug mode: off
WARNING: This is a development server.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```





ENDPOINT 1: TODOS LOS DATOS

```python
@app.route('/data', methods=['GET'])
def get_data():
    return data.to_json(orient='records'), 200
```

✓ 0.0s

127.0.0.1:5000/data

(19) Pinterest | Recibidos (2) - eva.... | Envato Elements: vi... | Canva | Gmail

[{"Age":51,"DistanceFromHome":6,"Education":2,"EmployeeCount":1,"Employee
& Development":false,"Department_Sales":true,"EducationField_Life
Sciences":true,"EducationField_Marketing":false,"EducationField_Medical":false,"
Resources":false,"JobRole_Laboratory Technician":false,"JobRole_Manager":false,
Scientist":false,"JobRole_Sales Executive":false,"JobRole_Sales Representative":fa
{"Age":31,"DistanceFromHome":10,"Education":1,"EmployeeCount":1,"Employee
& Development":true,"Department_Sales":false,"EducationField_Life
Sciences":true,"EducationField_Marketing":false,"EducationField_Medical":false,"
Resources":false,"JobRole_Laboratory Technician":false,"JobRole_Manager":false,
Scientist":true,"JobRole_Sales Executive":false,"JobRole_Sales Representative":fal
{"Age":32,"DistanceFromHome":17,"Education":4,"EmployeeCount":1,"Employee
& Development":true,"Department_Sales":false,"EducationField_Life
Sciences":false,"EducationField_Marketing":false,"EducationField_Medical":false,"
Resources":false,"JobRole_Laboratory Technician":false,"JobRole_Manager":false,
Scientist":false,"JobRole_Sales Executive":true,"JobRole_Sales Representative":fal

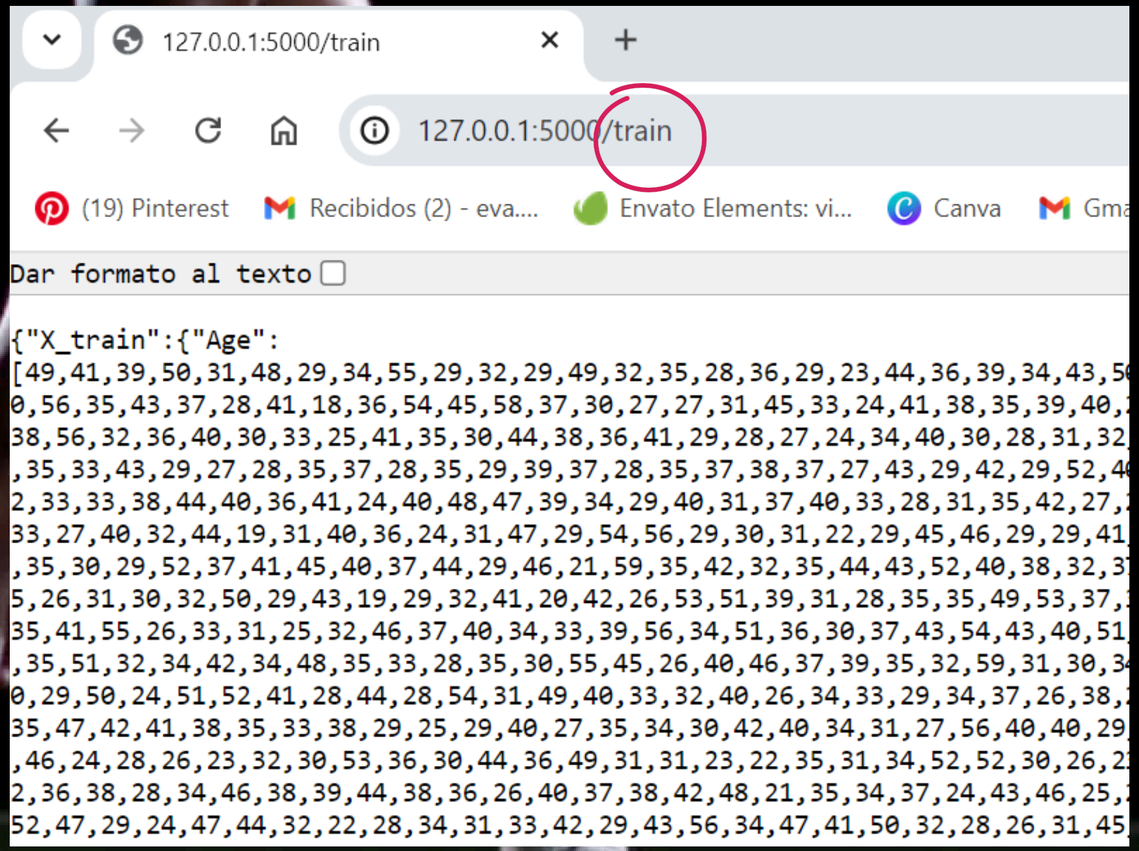
ENDPOINT 2: DATOS DEL TRAIN

```python
# Endpoint para obtener los datos de entrenamiento

def load_train_data():
    global train_data
    with open('train_data.json', 'r') as f:
        train_data = json.load(f)
```

✓ 0.0s

```python
# Endpoint para obtener los datos de entrenamiento

@app.route('/train', methods=['GET'])
def get_train_data():
    global train_data
    if train_data is None:
        return jsonify({"message": "Datos de entrenamiento no disponibles"}), 400

    return jsonify(train_data), 200
```

✓ 0.0s



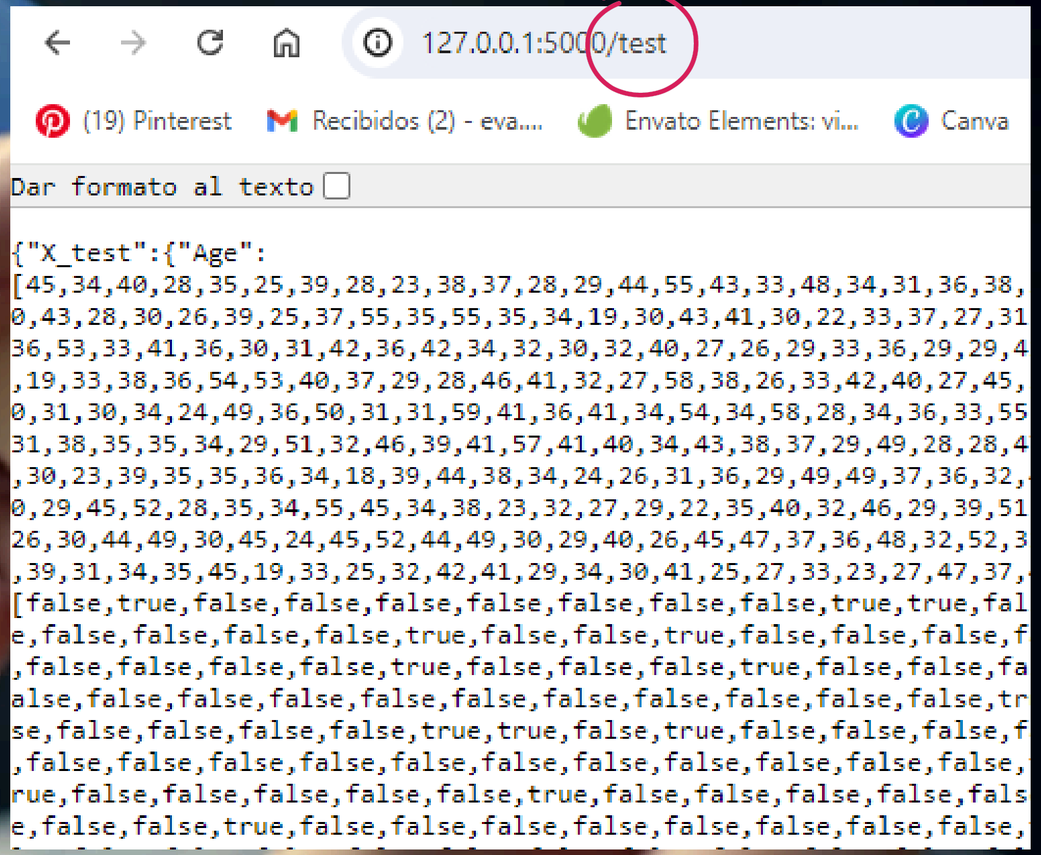
ENDPOINT 3: DATOS DEL TEST

```python
def load_test_data():
    global test_data
    with open('test_data.json', 'r') as f:
        test_data = json.load(f)
```

✓ 0.0s

```python
@app.route('/test', methods=['GET'])
def get_test_data():
    global test_data
    if test_data is None:
        return jsonify({"message": "Datos del test no disponibles"}), 400

    return jsonify(test_data), 200
```

✓ 0.0s



ENDPOINT 4: OBSERVACIÓN N DE MI DATASET

```python
@app.route('/observacion', methods=['GET'])
def obs():
    return jsonify(data.iloc[34].to_dict())
```

127.0.0.1:5000/observacion

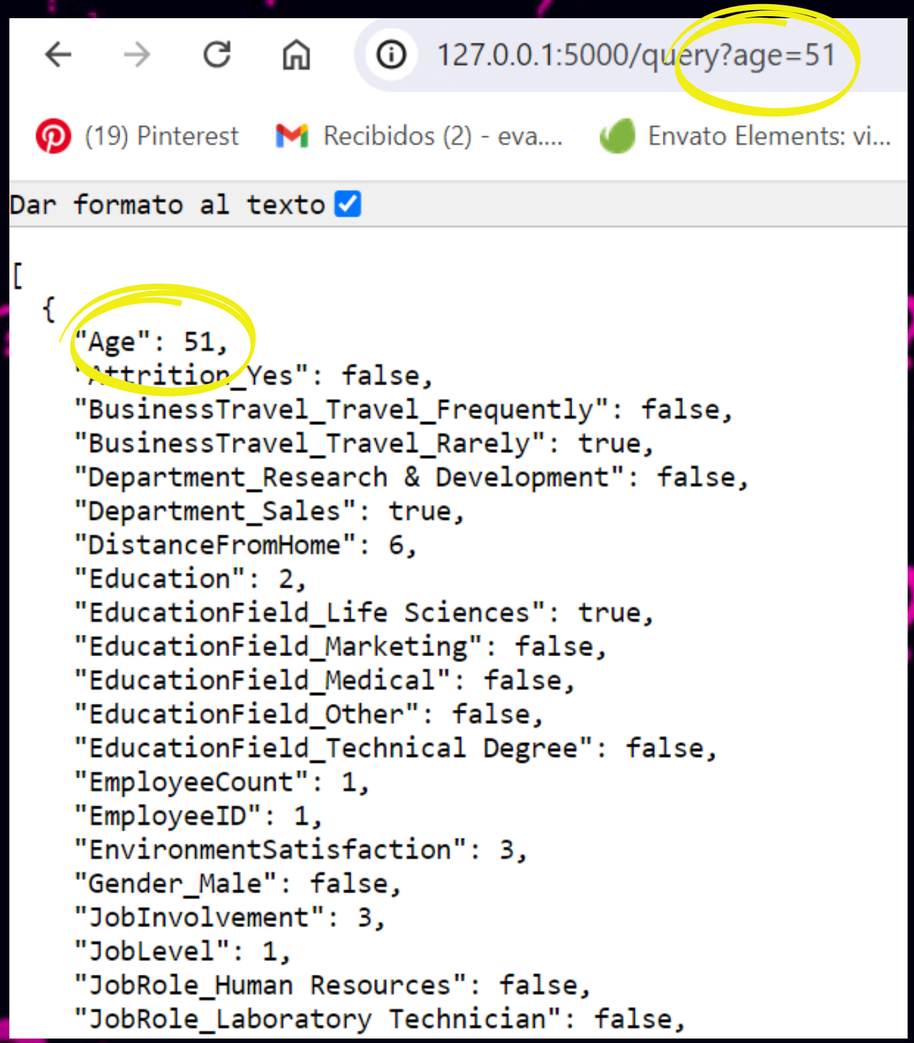
(19) Pinterest | Recibidos (2) - eva.... | Envato Elements: vi... | Canva | Gmail

Dar formato al texto

{"Age":37,"Attrition_Yes":false,"BusinessTravel_Travel_Frequently":true,"Busi
Sciences":false,"EducationField_Marketing":true,"EducationField_Medical":fals
Degree":false,"EmployeeCount":1,"EmployeeID":37,"EnvironmentSatisfaction":2.6
Director":false,"JobRole_Research Director":false,"JobRole_Research Scientist
Representative":false,"JobSatisfaction":4.0,"MaritalStatus_Married":true,"Mar
mesLastYear":3,"WorkLifeBalance":2.0,"YearsAtCompany":4,"YearsSinceLastPromot

ENDPOINT 5: QUERY

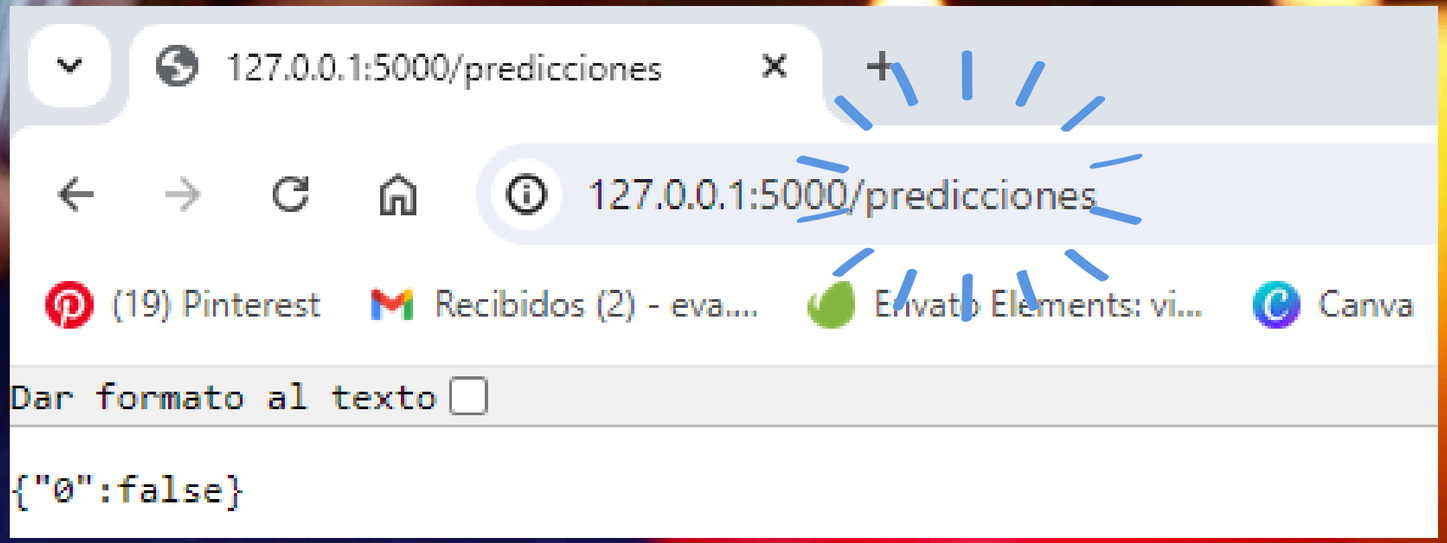
```python
@app.route('/query', methods=['GET'])
def age():
    if 'age' in request.args:
        age = int(request.args['age'])
    else: "No se enuentra una edad válida"

    edad = data[data['Age']==age]
    edad_dict = edad.to_dict(orient= "records")
    return jsonify(edad_dict)
```



ENDPOINT 6: PREDICCIONES

```
df = pd.DataFrame(y_pred)
```

```
@app.route('/predicciones', methods=['GET'])
def preds():
    return jsonify(df.iloc[2].to_dict())
```



GRACIAS