

Samuel Nicolas Rodriguez Celis – Edward David Diaz Fontecha – Andrés Leonardo Bayona Latorre

Nota: Los escenarios de prueba se realizan con la herramienta postman y unos datos en formato JSON. Los archivos de las peticiones y de los registros usados estarán disponibles en el repositorio para que sean consultados, en este documento por organización se incluyen los links que redirigen a los archivos en texto plano.

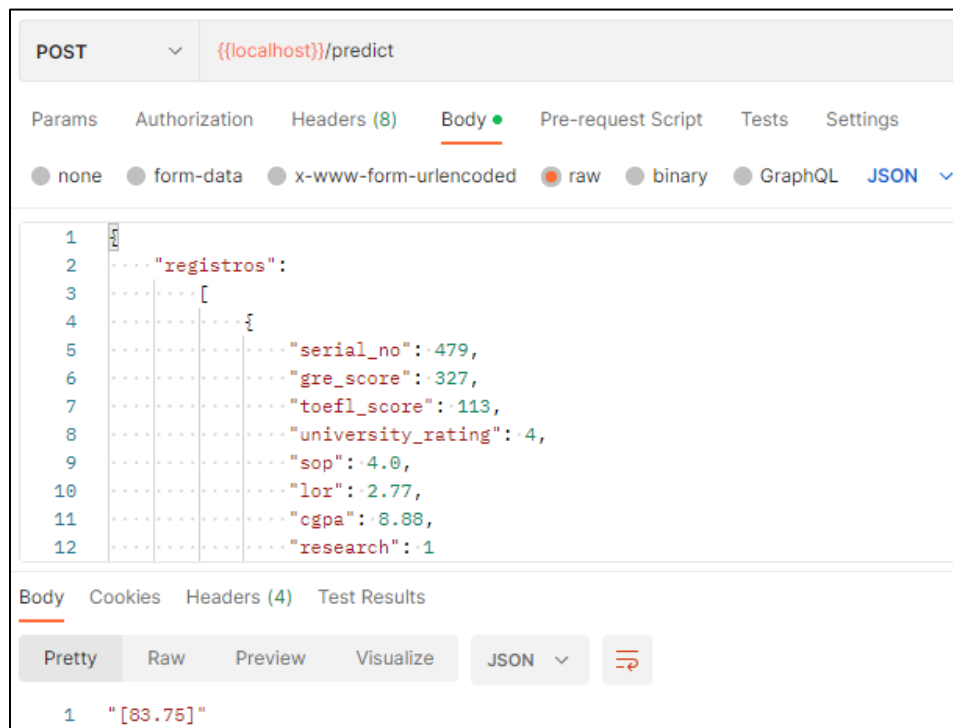
Escenarios de prueba

1. Una única predicción

- Datos usados

https://github.com/snrodriguezc/Lab4_GR19/blob/main/Data/predict_esc1.json

- Resultados



serial_no	Esperado	Obtenido
479	84.47	83.75

- Conclusión

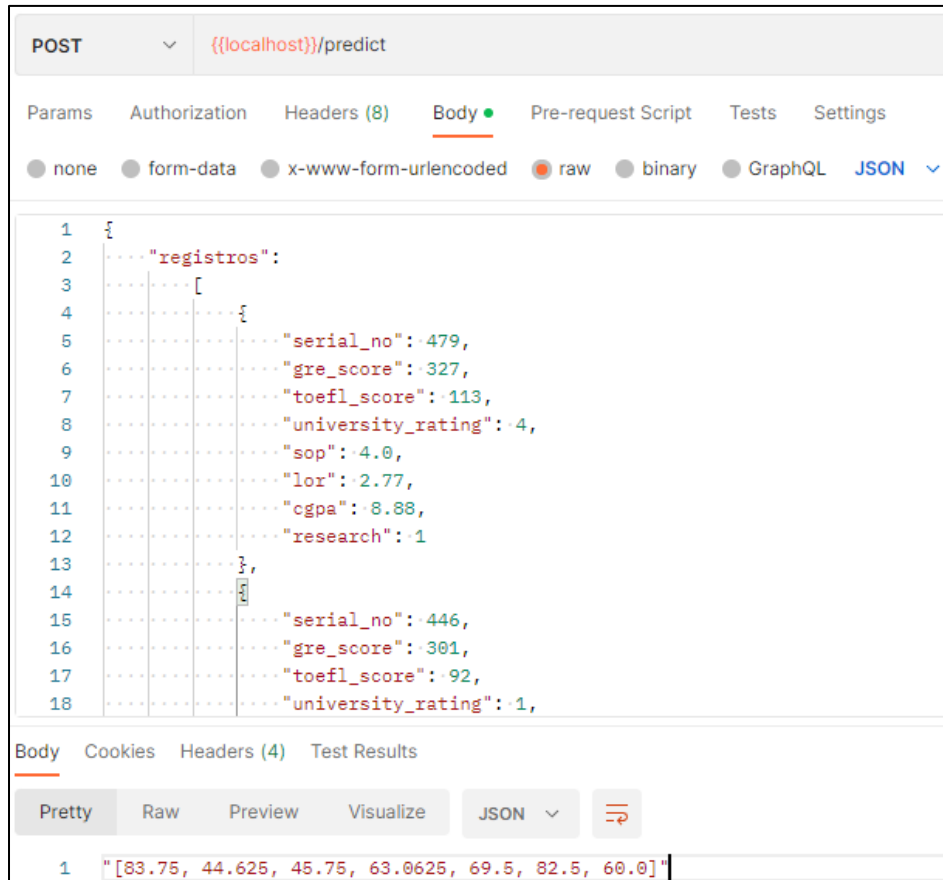
Primero que todo, vemos que para una única predicción que cuenta con valores en todos los atributos la ejecución es exitosa. Además, la predicción se asemeja bastante al valor esperado.

2. Varias predicciones

- Datos usados

https://github.com/snrodriguezc/Lab4_GR19/blob/main/Data/predict_esc2.json

- Resultados



```

1  {
2    "registros":
3      [
4        {
5          "serial_no": 479,
6          "gre_score": 327,
7          "toefl_score": 113,
8          "university_rating": 4,
9          "sop": 4.0,
10         "lor": 2.77,
11         "cgpa": 8.88,
12         "research": 1
13       },
14       {
15         "serial_no": 446,
16         "gre_score": 301,
17         "toefl_score": 92,
18         "university_rating": 1,
19       }
20     ]
21   }
  
```

```

1  "[83.75, 44.625, 45.75, 63.0625, 69.5, 82.5, 60.0]"
  
```

serial_no	Esperado	Obtenido
479	84.47	83.75
446	45.08	44.625
336	47.42	45.75
20	62	63.0625
432	73	69.5
112	69	82.5
55	70	60

- Conclusión

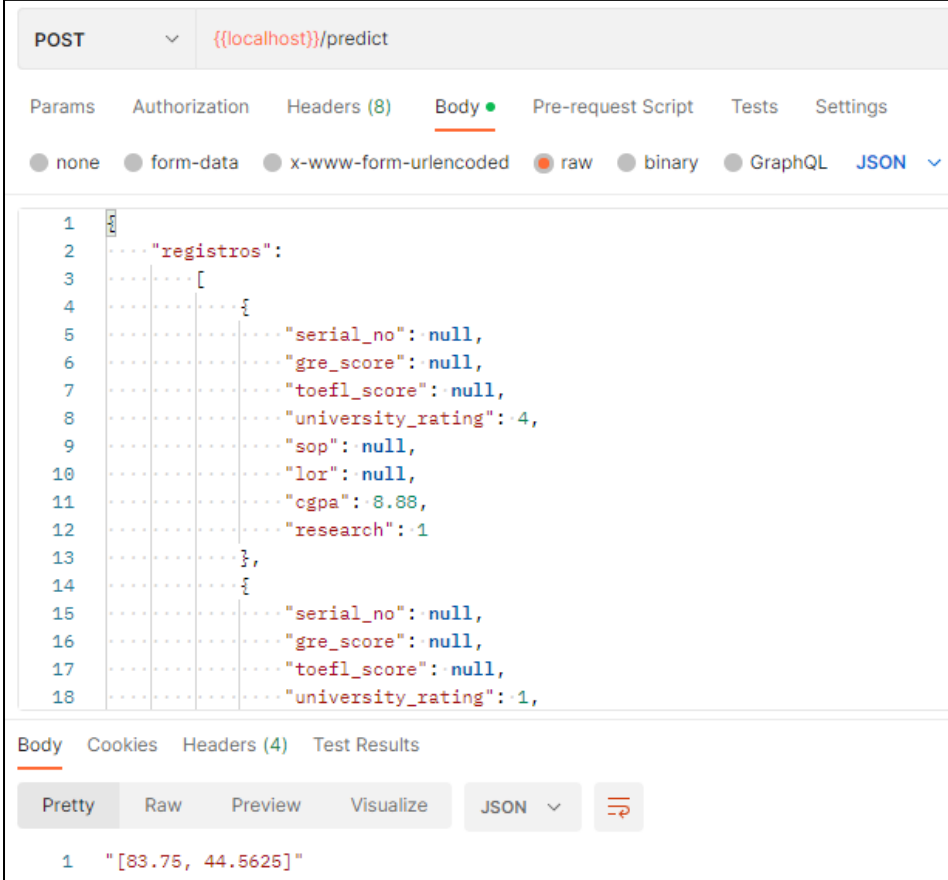
En los datos usados contamos con un total de 7 registros, y en el JSON de respuesta vemos una lista con exactamente 7 predicciones (misma cardinalidad). Así, se prueba que se pueden solicitar tantas predicciones simultaneas como se desee. En cuanto a los resultados tenemos algunos bastante precisos como el de serial 446 o el de serial 20, sin embargo, algunos como el de serial 112 se aleja un poco más del esperado.

3. Predicciones con nulos en variables no relevantes

- Datos usados

https://github.com/snrodriguezc/Lab4_GR19/blob/main/Data/predict_esc3.json

- Resultado



```

1 POST {{localhost}}/predict
2
3 Params Authorization Headers (8) Body Pre-request Script Tests Settings
4
5 none form-data x-www-form-urlencoded raw binary GraphQL JSON
6
7 1
8 2 ..... "registros":
9 3 ..... [
10 4 ..... {
11 5 ..... "serial_no": null,
12 6 ..... "gre_score": null,
13 7 ..... "toefl_score": null,
14 8 ..... "university_rating": 4,
15 9 ..... "sop": null,
16 10 ..... "lor": null,
17 11 ..... "cgpa": 8.88,
18 12 ..... "research": 1
19 13 ..... },
20 14 ..... {
21 15 ..... "serial_no": null,
22 16 ..... "gre_score": null,
23 17 ..... "toefl_score": null,
24 18 ..... "university_rating": 1,
25
26 Body Cookies Headers (4) Test Results
27
28 Pretty Raw Preview Visualize JSON
29
30 1 "[83.75, 44.5625]"
  
```

serial_no	Esperado	Obtenido
479	84.47	83.75
446	45.08	44.625

- Conclusión

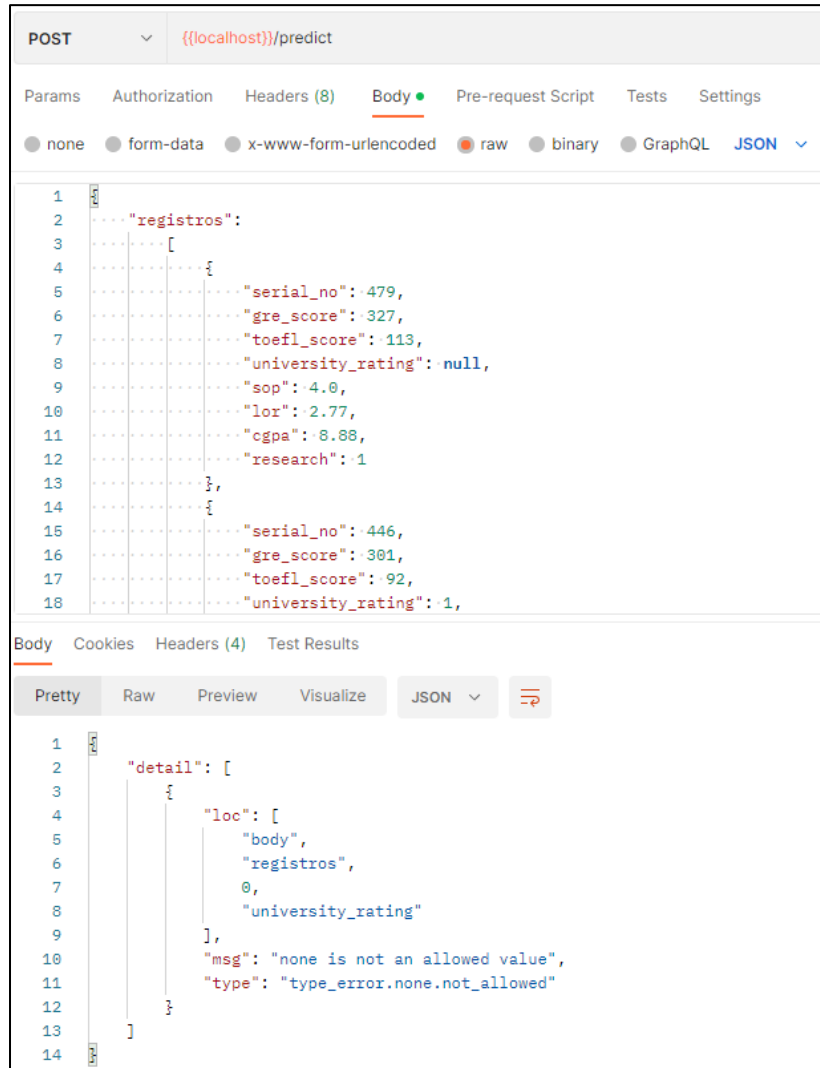
El resultado de esta prueba es similar al del anterior, ya que comprobamos que las predicciones simultaneas son permitidas y los resultados son relativamente coherentes. En este caso aunque tengamos valores nulos, están sobre variables que el modelo no utiliza puesto que en el laboratorio #3 se determinó que no proveían mucha información.

4. Predicciones con nulos en variables relevantes

- Datos usados

https://github.com/snrodriguezc/Lab4_GR19/blob/main/Data/predict_esc4.json

- Resultado



```

POST {{localhost}}/predict

{
  "registros": [
    {
      "serial_no": 479,
      "gre_score": 327,
      "toefl_score": 113,
      "university_rating": null,
      "sop": 4.0,
      "lor": 2.77,
      "cgpa": 8.88,
      "research": 1
    },
    {
      "serial_no": 446,
      "gre_score": 301,
      "toefl_score": 92,
      "university_rating": 1,
    }
  ]
}
  
```

```

{
  "detail": [
    {
      "loc": [
        "body",
        "registros",
        0,
        "university_rating"
      ],
      "msg": "none is not an allowed value",
      "type": "type_error.none.not_allowed"
    }
  ]
}
  
```

- Conclusión

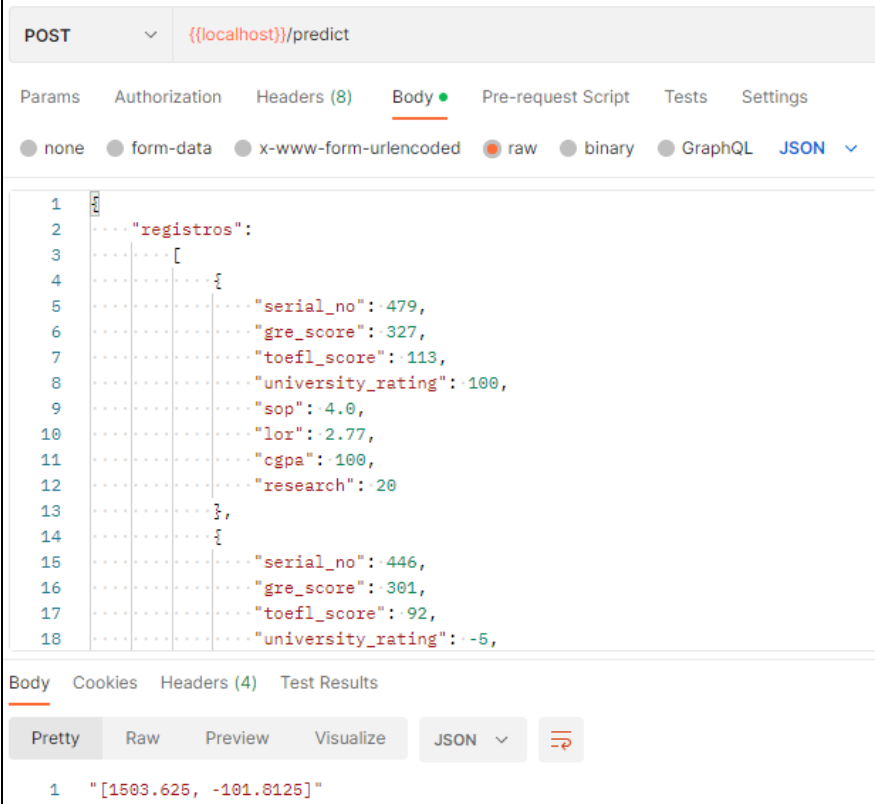
En este caso sucede algo diferente, y es que al tener valores nulos en las variables que “importan” para hacer las predicciones ocurre un error en la ejecución, el cual de cierta manera fue inducido en el desarrollo del API porque se decidió no permitir aquellas entradas con nulos en estos atributos. Así logramos evitar la generación de predicciones completamente incoherentes, que en una aplicación real podrían traer problemas al cliente.

5. Predicciones con valores extremos o atípicos

- Datos usados

https://github.com/snrodriquezc/Lab4_GR19/blob/main/Data/predict_esc5.json

- Resultado



POST {{localhost}}/predict

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2  "registros":
3  [
4    {
5      "serial_no": 479,
6      "gre_score": 327,
7      "toefl_score": 113,
8      "university_rating": 100,
9      "sop": 4.0,
10     "lor": 2.77,
11     "cgpa": 100,
12     "research": 20
13   },
14   {
15     "serial_no": 446,
16     "gre_score": 301,
17     "toefl_score": 92,
18     "university_rating": -5,

```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```

1  "[1503.625, -101.8125]"

```

- Conclusión

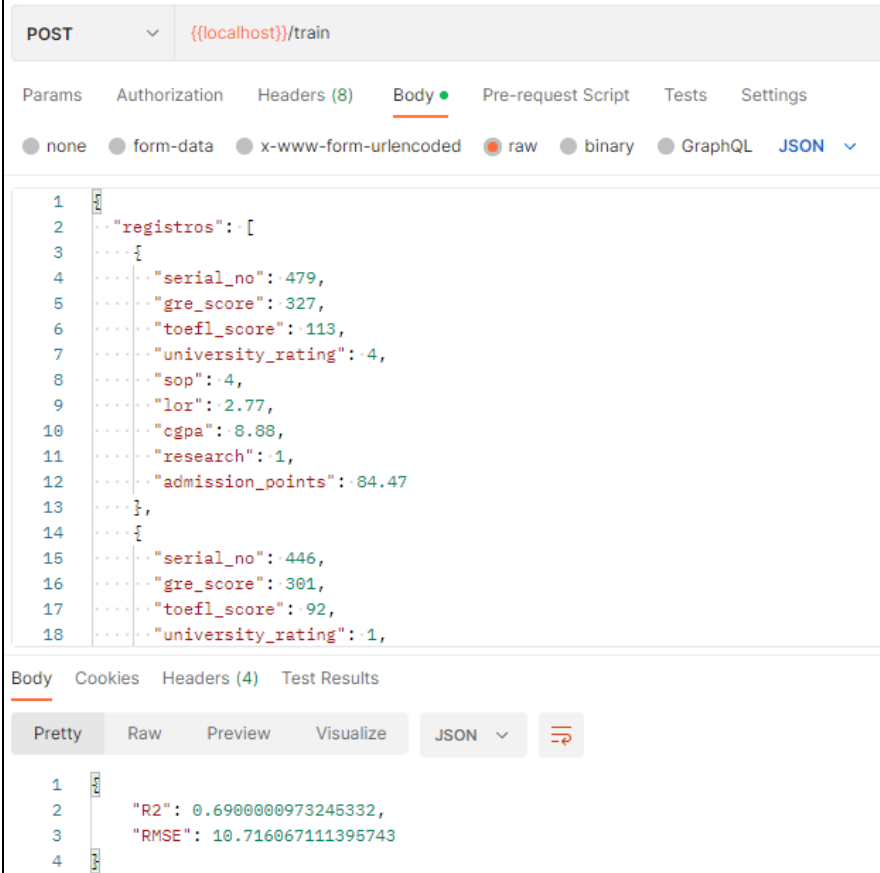
Esta ultima prueba sobre las predicciones se hace con el fin de obtener valores incoherentes, y como vemos en la imagen es posible obtenerlos. Esto se logró ingresando registros con valores atípicos y demasiado alejados de los rangos establecidos por el negocio. Lo cual debería evitarse y será una situación a tener en cuenta en la estrategia que será explicada al final del documento.

6. Re-entrenamiento del modelo

- Datos usados

https://github.com/snrodriquezc/Lab4_GR19/blob/main/Data/university_admission_train.json

- Resultado



POST `{{localhost}}/train`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

```

1  {
2    "registros": [
3      {
4        "serial_no": 479,
5        "gre_score": 327,
6        "toefl_score": 113,
7        "university_rating": 4,
8        "sop": 4,
9        "lor": 2.77,
10       "cgpa": 8.88,
11       "research": 1,
12       "admission_points": 84.47
13     },
14     {
15       "serial_no": 446,
16       "gre_score": 301,
17       "toefl_score": 92,
18       "university_rating": 1,

```

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "R2": 0.6900000973245332,
3    "RMSE": 10.716067111395743
4  }

```

- Conclusión

El reentrenamiento del modelo se realiza con éxito y la respuesta contiene 2 métricas sobre el modelo construido: R^2 y $RMSE$. En este caso para los datos escogidos (ver link arriba), se obtuvo un R^2 de 0.69 y un $RMSE$ de 10.72. De esta manera, se comprueba que siguiendo el formato en que están los registros en el archivo .JSON es posible reentrenar el modelo, generando uno nuevo que pueda ser usado para generar nuevas predicciones.

Estrategia para mitigar incoherencias y errores de ejecución

Las predicciones fueron en su mayoría coherentes mientras se utilizaron registros “normales” en la entrada, con valores no nulos en las columnas más relevantes y con valores dentro de los rangos establecidos por el negocio; claramente no es un modelo perfecto y aun con todo esto en algunas ocasiones hace predicciones un poco alejadas de lo esperado. Para mitigar esto se debe trabajar en la creación de un mejor pipeline, donde se incluyan transformaciones personalizadas mucho más profundas o incluso puede llegar a ser necesario evaluar el uso de otro modelo o algoritmo.

En cuanto a errores en la ejecución o detalles para tener en cuenta, tenemos que se debe tratar de mejor manera la interacción con el cliente, ya que en algún caso no comprobaba que los datos de entrada estuvieran dentro de los rangos o que fueran nulos. Así, determinamos que sería útil crear condicionales que generen excepciones mucho más informativas y fáciles de interpretar por el cliente para que no cometa estos “errores” al ingresar los datos.