

Laboratorio 4

Juan Sebastián Pinzón Rocancio

Santiago Pardo Bravo

Daniel Felipe Reyes Ramírez

Link a repositorio en Github:

<https://github.com/G12-NeuralLesses-BusinessIntelligence/Lab4>

Primer endpoint - Predict:

A continuación veremos los escenarios. Los escenarios variarán entre cada uno en la cantidad de elementos que tienen, en sí en la ejecución es exitosa o fallida, y en qué tan correcta es la predicción dados los valores de entrada de los features.

Los escenarios son conjuntos de tamaño n conformados por elementos tomados de forma aleatoria del dataset original usado en el laboratorio 3 y transformado a formato JSON.

Finalmente, cabe aclarar que la variable a predecir es *Admission Points*, y aunque claramente el modelo no la tendrá en cuenta para la predicción ya que es la variable a predecir, la dejaremos dentro del JSON ya que el modelo de todas formas la ignorará pero nos servirá para compararla con la predicción que haga el modelo. En un entorno de producción donde no nos importe comparar la entrada con la predicción simplemente podríamos poner un 0 para llenar la columna. Esto lo realizaremos en el *Escenario 5* a manera de demostración.

Escenario 1.

Escenario tomado del dataset original. Los resultados deberían ser lo más correctos posibles y la ejecución no debería dar errores.

Tamaño del conjunto de prueba: 10 elementos

```
[{"serial_no": "108", "gre_score": "338", "toefl_score": "117", "university_rating": "4", "sop": "3.5", "lor": "2.92", "cgpa": "9.46", "research": "1", "admission_points": "91"}
```

```
,{"serial_no":"298","gre_score":"329","toefl_score":"114","university_rating":"3",
"sop":"3.28","lor":"4.5","cgpa":"9.24","research":"0","admission_points":"87.9
5"},{"serial_no":"498","gre_score":"280","toefl_score":"72","university_rating":
"5","sop":"4.5","lor":"5","cgpa":"9.56","research":"1","admission_points":"93"},
{"serial_no":"300","gre_score":"305","toefl_score":"96","university_rating":"3",
"sop":"4.58","lor":"1.12","cgpa":"8.65","research":"0","admission_points":"71"},
{"serial_no":"255","gre_score":"275","toefl_score":"94","university_rating":"4",
"sop":"4","lor":"5","cgpa":"8.79","research":"0","admission_points":"127.5"},{"s
erial_no":"255","gre_score":"321","toefl_score":"114","university_rating":"4","s
op":"4","lor":"5","cgpa":"9.12","research":"0","admission_points":"85"},{"serial
_no":"415","gre_score":"321","toefl_score":"109","university_rating":"4","sop":"
3.5","lor":"4","cgpa":"8.35","research":"1","admission_points":"72"},{"serial_no
":"286","gre_score":"329","toefl_score":"115","university_rating":"5","sop":"4.1
6","lor":"1.67","cgpa":"6.75","research":"1","admission_points":"90.96"},{"seria
l_no":"1","gre_score":"337","toefl_score":"118","university_rating":"4","sop":"4
.5","lor":"1.63","cgpa":"8.51","research":"1","admission_points":"92"},{"serial_
no":"397","gre_score":"325","toefl_score":"84","university_rating":"3","sop":"3"
,"lor":"3.5","cgpa":"8.79","research":"1","admission_points":"84"}]
```

Output:

```
{
  "results": {
    "0": {
      "0": 89.9064917720235,
      "1": 71.60378717524272,
      "2": 88.55483105633228,
      "3": 62.21592751757917,
      "4": 68.7296567278773,
      "5": 77.25474114588343,
      "6": 83.07313852410638,
      "7": 81.43827369600024,
      "8": 84.88722042849943,
      "9": 76.350711553958
    }
  }
}
```

Ejecución en Postman:

http://127.0.0.1:8000/predict

POST http://127.0.0.1:8000/predict

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

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

```

1 {
2   "serial_no": "108",
3   "gre_score": "338",
4   "toefl_score": "117",
5   "university_rating": "4",
6   "sop": "3.5",
7   "lor": "2.92",
8 }

```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 43 ms Size: 361 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "results": {
3     "0": {
4       "0": 89.9064917720235,
5       "1": 71.60378717524272,
6       "2": 88.55483105633228,

```

Escenario 2.

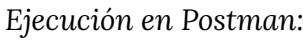
Escenario tomado del dataset original. Los resultados deberían ser lo más correctos posibles y la ejecución no debería dar errores.

Tamaño del conjunto de prueba: 5 elementos

```
[{"serial_no": "397", "gre_score": "325", "toefl_score": "107", "university_rating": "3", "sop": "1.5", "lor": "3.5", "cgpa": "9.11", "research": "1", "admission_points": "84"}, {"serial_no": "41", "gre_score": "313", "toefl_score": "77", "university_rating": "1", "sop": "2.79", "lor": "3.93", "cgpa": "8.08", "research": "0", "admission_points": "50.63"}, {"serial_no": "40", "gre_score": "299", "toefl_score": "88", "university_rating": "2", "sop": "3.33", "lor": "3.05", "cgpa": "7.92", "research": "0", "admission_points": "48.5"}, {"serial_no": "338", "gre_score": "300", "toefl_score": "78", "university_rating": "5", "sop": "5", "lor": "5", "cgpa": "9.47", "research": "1", "admission_points": "94"}, {"serial_no": "447", "gre_score": "299", "toefl_score": "100", "university_rating": "2", "sop": "2", "lor": "2.26", "cgpa": "8.55", "research": "0", "admission_points": "46.52"}]
```

Output:

```
{
  "results": {
    "0": {
      "0": 78.94514658122547,
      "1": 47.81538339189483,
      "2": 52.19662742030322,
      "3": 91.2529873273338,
      "4": 54.622001762822336
    }
  }
}
```



Tamaño del conjunto de prueba: 15 elementos

```
[{"serial_no": "426", "gre_score": "323", "toefl_score": "111", "university_rating": "5", "sop": "2.15", "lor": "5", "cgpa": "9.86", "research": "1", "admission_points": "138"}, {"serial_no": "492", "gre_score": "297", "toefl_score": "99", "university_rating": "4", "sop": "2.86", "lor": "1.99", "cgpa": "7.81", "research": "0", "admission_points": "54"}, {"serial_no": "288", "gre_score": "324", "toefl_score": "97", "university_rating": "5", "sop": "4.63", "lor": "4.5", "cgpa": "9.08", "research": "1", "admission_points": "89"}, {"serial_no": "59", "gre_score": "300", "toefl_score": "99", "university_rating": "1", "sop": "4.32", "lor": "2", "cgpa": "6.8", "research": "1", "admission_points": "36"}, {"serial_no": "227", "gre_score": "306", "toefl_score": "110", "university_rating": "2", "sop": "3.5", "lor": "4", "cgpa": "8.45", "research": "0", "admission_points": "63"}, {"serial_no": "95", "gre_score": "302", "toefl_score": "111", "university_rating": "2", "sop": "4.43", "lor": "3", "cgpa": "7.71", "research": "0", "admission_points": "42.44"}, {"serial_no": "186", "gre_score": "327", "toefl_score": "109", "university_rating": "4", "sop": "4.54", "lor": "2.05", "cgpa": "9.11", "research": "1", "admission_points": "89"}, {"serial_no": "107", "gre_score": "313", "toefl_score": "81", "university_rating": "3", "sop": "2.85", "lor": "2.71", "cgpa": "7.7", "research": "0", "admission_points": "39.6"}, {"serial_no": "100", "gre_score": "305", "toefl_score": "106", "university_rating": "3", "sop": "3.7", "lor": "3.5", "cgpa": "8.5", "research": "0", "admission_points": "50"}]
```

```
l_no":"35","gre_score":"331","toefl_score":"114","university_rating":"4","sop":"4.4","lor":"1.35","cgpa":"9.74","research":"0","admission_points":"90.36"},{"serial_no":"119","gre_score":"296","toefl_score":"99","university_rating":"2","sop":"3","lor":"3.5","cgpa":"7.28","research":"0","admission_points":"47"},{"serial_no":"59","gre_score":"299","toefl_score":"98","university_rating":"1","sop":"3","lor":"2","cgpa":"6.82","research":"1","admission_points":"40.03"},{"serial_no":"33","gre_score":"338","toefl_score":"100","university_rating":"4","sop":"3","lor":"0.18","cgpa":"9.4","research":"1","admission_points":"91"},{"serial_no":"89","gre_score":"286","toefl_score":"113","university_rating":"2","sop":"3","lor":"2.67","cgpa":"7.77","research":"1","admission_points":"66"},{"serial_no":"172","gre_score":"334","toefl_score":"113","university_rating":"5","sop":"4","lor":"4.5","cgpa":"6.58","research":"1","admission_points":"89"},{"serial_no":"414","gre_score":"317","toefl_score":"89","university_rating":"3","sop":"3","lor":"2","cgpa":"7.94","research":"1","admission_points":"49"}]
```

Output:

```
{
  "results": {
    "0": {
      "0": 96.76833860907487,
      "1": 63.48732576439661,
      "2": 93.12867090605667,
      "3": 52.71317084351624,
      "4": 58.465999471439545,
      "5": 54.319720385995495,
      "6": 86.18182812912491,
      "7": 57.630392844469114,
      "8": 78.143577928714,
      "9": 50.05722891160464,
      "10": 51.72802160442677,
      "11": 85.19158000739357,
      "12": 62.881243199966086,
      "13": 83.55575372427563,
      "14": 70.51884101900862
    }
  }
}
```

Ejecución en Postman:

Por lo que, tomaremos del dataset original 5 elementos y cambiaremos los valores de esos features más importantes por valores aleatorios afectando así el resultado.

```
[{"serial_no": "99", "gre_score": "332", "toefl_score": "119", "university_rating": "1", "sop": "4.02", "lor": "3.6", "cgpa": "7.06", "research": "0", "admission_points": "90"}, {"serial_no": "480", "gre_score": "325", "toefl_score": "110", "university_rating": "1", "sop": "2.9", "lor": "4", "cgpa": "3.7", "research": "0", "admission_points": "79"}, {"serial_no": "316", "gre_score": "308", "toefl_score": "104", "university_rating": "5", "sop": "1.6", "lor": "3", "cgpa": "9.45", "research": "1", "admission_points": "65"}, {"serial_no": "182", "gre_score": "305", "toefl_score": "107", "university_rating": "4", "sop": "2.5", "lor": "2.5", "cgpa": "4.65", "research": "0", "admission_points": "71"}, {"serial_no": "171", "gre_score": "312", "toefl_score": "91", "university_rating": "5", "sop": "2.5", "lor": "3.5", "cgpa": "8.47", "research": "1", "admission_points": "68"}]
```

Output:

```
{
  "results": {
    "0": {
      "0": 49.48327255766584,
      "1": 31.746300469670317,
      "2": 90.29413245866955,
      "3": 50.52227582039556,
      "4": 86.02847856649437
    }
  }
}
```

Ejecución en Postman:

The screenshot shows a Postman interface for a POST request to `http://127.0.0.1:8000/predict`. The request body is a JSON array of 5 objects, each representing a student's data. The response is a JSON object with a `results` key, which contains an array of 5 predictions.

Request Body (JSON):

```
[{"serial_no": "99", "gre_score": "332", "toefl_score": "119", "university_rating": "1", "sop": "4.02", "lor": "3.6", "cgpa": "7.06", "research": "0", "admission_points": "90"}, {"serial_no": "480", "gre_score": "325", "toefl_score": "110", "university_rating": "1", "sop": "2.9", "lor": "4", "cgpa": "3.7", "research": "0", "admission_points": "79"}, {"serial_no": "316", "gre_score": "308", "toefl_score": "104", "university_rating": "5", "sop": "1.6", "lor": "3", "cgpa": "9.45", "research": "1", "admission_points": "65"}, {"serial_no": "182", "gre_score": "305", "toefl_score": "107", "university_rating": "4", "sop": "2.5", "lor": "2.5", "cgpa": "4.65", "research": "0", "admission_points": "71"}, {"serial_no": "171", "gre_score": "312", "toefl_score": "91", "university_rating": "5", "sop": "2.5", "lor": "3.5", "cgpa": "8.47", "research": "1", "admission_points": "68"}]
```

Response Body (JSON):

```
{
  "results": {
    "0": {
      "0": 49.48327255766584,
      "1": 31.746300469670317,
      "2": 90.29413245866955,
      "3": 50.52227582039556,
      "4": 86.02847856649437
    }
  }
}
```

En esta ejecución es interesante resaltar la diferencia que causó interferir en los valores de los features más importantes, lo que causó una predicción muy distinta a lo que se esperaba.

En la siguiente tabla podemos ver real vs predicha:

Real	Predicción
90	49.48
79	31.74
65	90.29
71	50.52
68	86.02

Escenario 5.

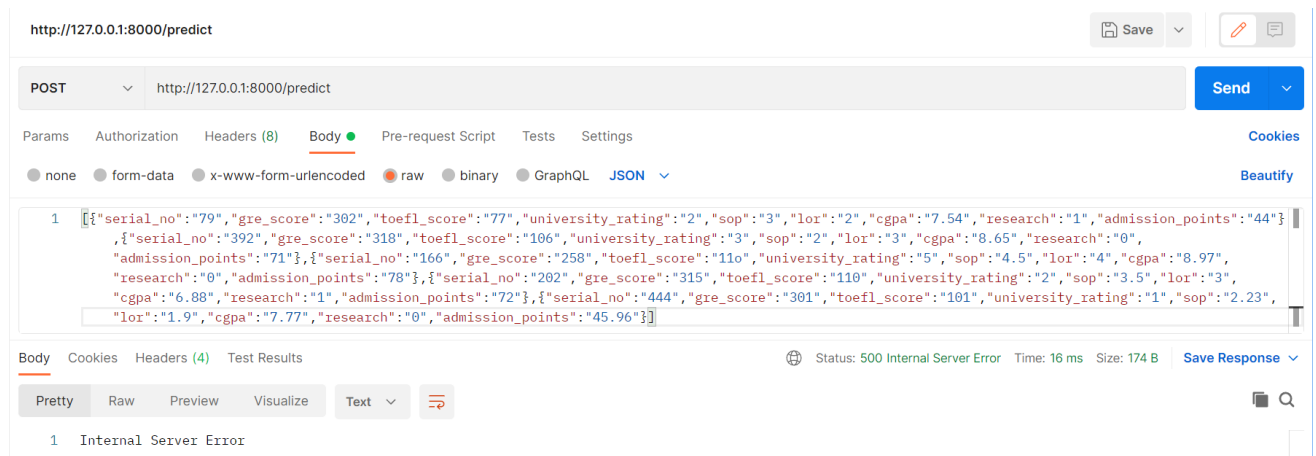
En este escenario provocaremos una ejecución fallida. Para eso, tomaremos 5 elementos del dataset original pero modificaremos uno agregándole un carácter a una de sus columnas, lo que hará que el modelo falle ya que debe recibir solo valores numéricos.

```
[{"serial_no": "79", "gre_score": "302", "toefl_score": "77", "university_rating": "2", "sop": "3", "lor": "2", "cgpa": "7.54", "research": "1", "admission_points": "44"}, {"serial_no": "392", "gre_score": "318", "toefl_score": "106", "university_rating": "3", "sop": "2", "lor": "3", "cgpa": "8.65", "research": "0", "admission_points": "71"}, {"serial_no": "166", "gre_score": "258", "toefl_score": "110", "university_rating": "5", "sop": "4.5", "lor": "4", "cgpa": "8.97", "research": "0", "admission_points": "78"}, {"serial_no": "202", "gre_score": "315", "toefl_score": "110", "university_rating": "2", "sop": "3.5", "lor": "3", "cgpa": "6.88", "research": "1", "admission_points": "72"}, {"serial_no": "444", "gre_score": "301", "toefl_score": "101", "university_rating": "1", "sop": "2.23", "lor": "1.9", "cgpa": "7.77", "research": "0", "admission_points": "45.96"}]
```

Output:

Internal Server Error

Ejecución en Postman:



Mensaje de error recibido en consola del servicio:

```
File "C:\Users\danie\Desktop\Escritorio\Universidad\202220\BI\Lab4\Lab4\.\main.py", line 31, in make_predictions
    dataModels = [json.loads(json.dumps(data), object_hook=lambda d: DataModel(**d))
for data in list_dataModel]
File "C:\Users\danie\Desktop\Escritorio\Universidad\202220\BI\Lab4\Lab4\.\main.py", line 31, in <listcomp>
    dataModels = [json.loads(json.dumps(data), object_hook=lambda d: DataModel(**d))
for data in list_dataModel]
File "C:\Users\danie\AppData\Local\Programs\Python\Python310\lib\json\__init__.py", line 359, in loads
    return cls(**kw).decode(s)
File "C:\Users\danie\AppData\Local\Programs\Python\Python310\lib\json\decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
File "C:\Users\danie\AppData\Local\Programs\Python\Python310\lib\json\decoder.py", line 353, in raw_decode
    obj, end = self.scan_once(s, idx)
File "C:\Users\danie\Desktop\Escritorio\Universidad\202220\BI\Lab4\Lab4\.\main.py", line 31, in <lambda>
    dataModels = [json.loads(json.dumps(data), object_hook=lambda d: DataModel(**d))
for data in list_dataModel]
File "pydantic\main.py", line 342, in pydantic.main.BaseModel.__init__
pydantic.error_wrappers.ValidationError: 1 validation error for DataModel
toefl_score
  value is not a valid float (type=type_error.float)
```

El mensaje es recibido ya que en el DataModel no se aceptan datos de tipo String. Por ejemplo, la columna que lanzó excepción fue `toefl_score`, que está declarada en el DataModel como `float`.

Segundo endpoint - Retraining:

En esta sección vamos a probar 5 escenarios distintos, donde variará la cantidad de muestras para entrenar al modelo y la variable objetivo. Debido a que varios de los

escenarios presentan una gran cantidad de muestras para entrenar el modelo, sólomente mostraremos en el documento el primero como forma de dar a entender el formato que siguen los JSON para el re-entrenamiento.

Escenario 6.

Número de muestras: 10 muestras.

Variable objetivo: Admission Points

```
{
  "Serial
  No.": {
    "0": 479, "1": 446, "2": 336, "3": 20, "4": 432, "5": 112, "6": 55, "7": 59, "8": 123, "9": 477
  },
  "GRE
  Score": {
    "0": 327, "1": 301, "2": 297, "3": 303, "4": 320, "5": 321, "6": 306, "7": 299, "8": 311, "9": 298
  },
  "TOEFL
  Score": {
    "0": 113, "1": 92, "2": 100, "3": 98, "4": 94, "5": 84, "6": 110, "7": 112, "8": 74, "9": 98
  },
  "University
  Rating": {
    "0": 4, "1": 1, "2": 1, "3": 3, "4": 2, "5": 4, "6": 3, "7": 1, "8": 4, "9": 2
  },
  "SOP": {
    "0": 4.0, "1": 1.85, "2": 2.41, "3": 3.5, "4": 1.38, "5": 4.13, "6": 3.0, "7": 0.97, "8": 1.5, "9": 2.23
  },
  "LOR
  ": {
    "0": 2.77, "1": 1.5, "2": 1.59, "3": 3.0, "4": 3.5, "5": 4.0, "6": 3.5, "7": 2.0, "8": 0.53, "9": 3.73
  },
  "CGPA": {
    "0": 8.88, "1": 7.71, "2": 7.89, "3": 8.5, "4": 8.78, "5": 8.68, "6": 8.0, "7": 6.82, "8": 8.36, "9": 7.43
  },
  "Research": {
    "0": 1, "1": 0, "2": 0, "3": 0, "4": 1, "5": 1, "6": 0, "7": 1, "8": 0, "9": 0
  },
  "Admission
  Points": {
    "0": 84.47, "1": 45.08, "2": 47.42, "3": 62.0, "4": 73.0, "5": 69.0, "6": 70.0, "7": 40.03, "8": 57.0, "9": 49.08
  },
  "var_obj": "Admission Points"
}
```

Como podemos evidenciar, las muestras se colocan consecutivamente según la columna y finalmente tenemos un atributo `var_obj` que describe cuál de las columnas será la variable objetivo.

Output:

```
{
  "#Samples": 10,
  "Current var objective": "Admission Points",
  "RMSE": 0.814,
  "MAE": 0.705,
  "R2 score": 0.996
}
```

```
}
```

Ejecución en Postman:

The screenshot shows the Postman interface for a POST request to `http://127.0.0.1:8000/retraining?data_in={"Serial No.":{"0":479,"1":446,"2":336,"3":20,"4":432,"5":112,"6":55,"7":59,"8":123,"9":477}, "GRE Score":{"0":327,"1":301,"2":297,"3":303,"4":320,"5":321,"6":306,"7":299,"8":303,"9":477}}`. The request body is a JSON object with the following structure:

```
{
  "#Samples": 10,
  "Current var objective": "Admission Points",
  "RMSE": 0.814,
  "MAE": 0.705,
  "R2 score": 0.996
}
```

Escenario 7.

Número de muestras: 500 muestras.

Variable objetivo: Admission Points

Output:

```
{
  "#Samples": 500,
  "Current var objective": "Admission Points",
  "RMSE": 106.171,
  "MAE": 7.586,
  "R2 score": 0.721
}
```

Ejecución en Postman:

http://127.0.0.1:8000/retraining?data_in={"Serial No.":{"0":417,"1":275,"2":94,"3":79,"4":393,"5":268,"6":347,"7":177,"8":35,"9":214,"10":101,"11":212,"12":454,"13":39...

POST http://127.0.0.1:8000/retraining?data_in={"Serial No.":{"0":417,"1":275,"2":94,"3":79,"4":393,"5":268,"6":347,"7":177,"8":35,"9":214,"10":101,"11":212,"12":454,"13":398,"14":...

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	data_in	{"Serial No.":{"0":417,"1":275,"2":94,"3":79,"4":393,"5":268,"6":347,"7":177,"8":35,"9":214,"10":101,"11":212,"12":454			
	Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 51 ms Size: 229 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "#Samples": 500,
3   "Current var objective": "Admission Points",
4   "RMSE": 106.171,
5   "MAE": 7.586,
6   "R2 score": 0.721
7 }
```

Escenario 8.

Número de muestras: 20 muestras.

Variable objetivo: GRE Score

Output:

```
{
  "#Samples": 20,
  "Current var objective": "GRE Score",
  "RMSE": 95.567,
  "MAE": 7.372,
  "R2 score": 0.657
}
```

Ejecución en Postman:

http://127.0.0.1:8000/retraining?data_in={"Serial No.":{"0":445,"1":225,"2":382,"3":248,"4":479,"5":285,"6":89,"7":177,"8":96,"9":45,"10":445,"11":54,"12":438,"13":1...

POST http://127.0.0.1:8000/retraining?data_in={"Serial No.":{"0":445,"1":225,"2":382,"3":248,"4":479,"5":285,"6":89,"7":177,"8":96,"9":45,"10":445,"11":54,"12":438,"13":148,"14

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	data_in	{"Serial No.":{"0":445,"1":225,"2":382,"3":248,"4":479,"5":285,"6":89,"7":177,"8":96,"9":45,"10":445,"11":54,"12":438,...			
	Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 28 ms Size: 219 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "#Samples": 20,
3   "Current var objective": "GRE Score",
4   "RMSE": 95.567,
5   "MAE": 7.372,
6   "R2 score": 0.657
7 }
```

Escenario 9.

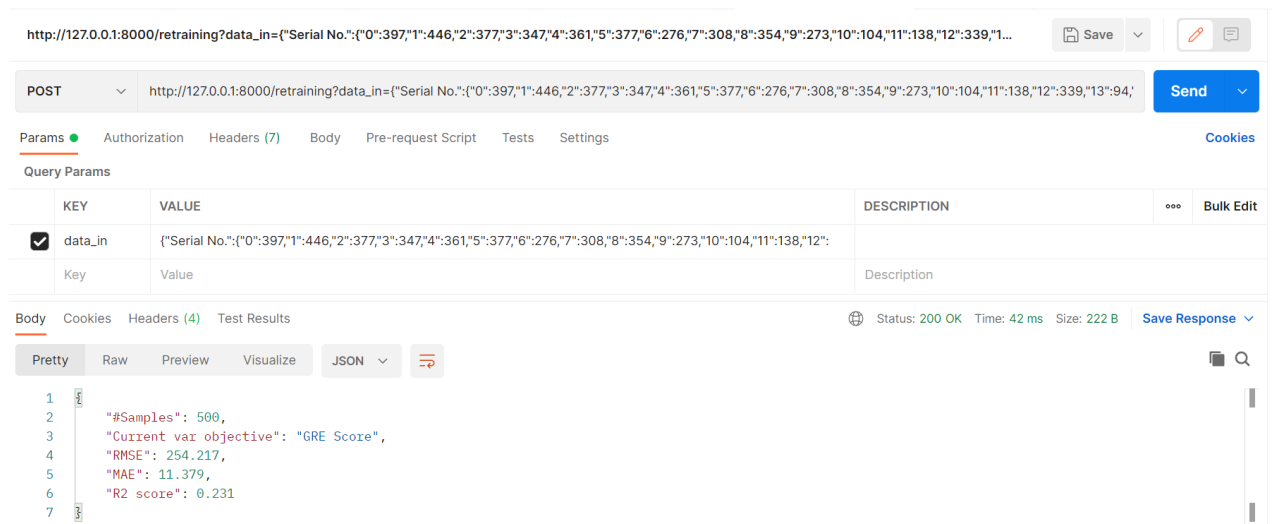
Número de muestras: 500 muestras.

Variable objetivo: GRE Score

Output:

```
{
  "#Samples": 500,
  "Current var objective": "GRE Score",
  "RMSE": 254.217,
  "MAE": 11.379,
  "R2 score": 0.231
}
```

Ejecución en Postman:



Escenario 10.

Número de muestras: 500 muestras.

Variable objetivo: CGPA

Output:

```
{
  "#Samples": 500,
  "Current var objective": "CGPA",
  "RMSE": 0.322,
  "MAE": 0.37,
  "R2 score": 0.492
}
```

Ejecución en Postman:

http://127.0.0.1:8000/retraining?data_in={"Serial No.":{"0":122,"1":118,"2":95,"3":199,"4":457,"5":496,"6":339,"7":313,"8":381,"9":84,"10":94,"11":352,"12":412,"13":211,"14":14}}

Save

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	data_in	{"Serial No.":{"0":122,"1":118,"2":95,"3":199,"4":457,"5":496,"6":339,"7":313,"8":381,"9":84,"10":94,"11":352,"12":412,"13":211,"14":14}}			
	Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 54 ms Size: 213 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "#Samples": 500,
3   "Current var objective": "CGPA",
4   "RMSE": 0.322,
5   "MAE": 0.37,
6   "R2 score": 0.492
7 }
```

Conclusiones:

Los resultados obtenidos son relevantes por que nos permiten darnos cuenta de los siguientes factores:

- En los escenarios de predicción donde se ingresaron las muestras reales pudimos obtener resultados satisfactorios. Donde si bien existían casos en los que la predicción se podía alejar un poco del valor real, el modelo hacía un buen trabajo prediciendo los valores de forma óptima de varias pruebas realizadas.
- El modelo es susceptible a errores siempre sucedan uno de los siguientes escenarios:

En primer lugar, los valores de los features con mayor correlación sean muy desproporcionados o por poco sentido, lo que puede influir en una mala predicción por parte del modelo.

En segundo lugar, como pudimos evidenciar, cuando se ingresa un tipo de dato incorrecto al modelo ocasiona que este tenga un fallo directamente en el programa, lo que detiene su ejecución. Es por eso que es importante que se pueda asegurar la calidad de los datos si se lleva a un ambiente de producción donde deba realizar predicciones de forma regular y no se quiere que el modelo deje de funcionar por una ejecución fallida.

- El segundo endpoint de reentrenamiento nos permite que el modelo pueda volver a ser entrenado durante su montaje en ejecución, lo que significa que no sólo un científico de datos o programar puede hacer dicha tarea, sino que sus

usuarios finales pueden realizar esa tarea de forma sencilla pasándole un dataset al modelo por medio del API así este se puede reentrenar.

Finalmente, para mitigar incoherencias en la predicción y errores de ejecución es importante que si el API es conectado a otro servicio que le provea los datos, tales como un frontend, se asegure desde ese servicio la calidad de los datos. Es decir, que los datos que se ingresan sean coherentes y reales, y para asegurar su correcta ejecución, que los datos respeten su dominio (por ejemplo *University Ranking* debe ser entre 1 y 5) y que los datos sean numéricos sin contener ningún tipo de carácter.