

Laboratorio 4

Andrés Peña Diaz (201913766)

Laura Isabela Martínez Galindo (202012613)

Juan José Córdoba Vela (201922105)

Responsabilidades:

Creación del pipeline: Juan José Cordoba

Para crear el pipeline, se tuvo que hacer varios pasos antes:

1. Cargar el archivo *university_admission_train.csv*
 2. Eliminar la columna 'Unnamed: 0' del dataset ya que el archivo csv de test no tiene esta columna. Si se deja arroja algunos errores.
 3. Eliminar todas las filas que tengan nulos.
 4. Dividir los datos en train y test.
 5. Crear el ColumnTransformer. Esto se usa para normalizar entre 0 y 1 las columnas. Además, se especifican las columnas que no vamos a usar.
-
1. Crear el Pipeline. Este pipeline se crea con el ColumnTransformer y con un modelo de regresión.
 2. Luego, se ajusta el training data al pipeline creado.
 3. Se predicen datos nuevos para X_train y X_test.
 4. Obtenemos el score tanto del train como del test.
 5. Obtenemos el mean absolut error del train y test pasando como parámetros Y_train o Y_test y las predicciones del paso 3.
 6. Se usa la funcion dump para serializar el modelo
 7. Se usa la funcion load para cargar el archivo y poder usarlo en cualquier aplicación

Construcción del API: Andrés Peña

Se crearon tres archivos: DataModel, Main y PredictionModel.

En el primer archivo mencionado se crea una clase DataModel donde se simboliza el registro de la base de datos.

En Main se crea un API básico y se crea una función que pueda recibir y transformar los datos recibidos en un dataframe para que estos puedan ser recibidos por el modelo.

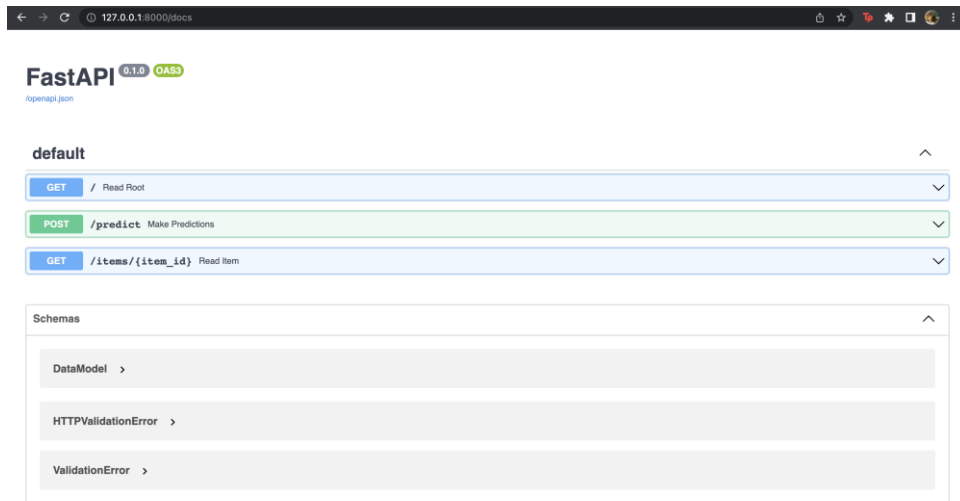
Aquí tambien se realizan predicciones.

En el último archivo, PredictionModel tenemos la clase Model que representa el modelo cargado. Se crea una instancia con base en el modelo joblib obtenido en la creación del pipeline. Finalmente está la funcion make_predictions para realizar predicciones.

Para verificar que el API estaba correctamente creado debiamos ejecutar el siguiente comando en la terminal:

```
vicorn main:app --reload
```

Y debíamos copiar y pegar en un browser: <http://127.0.0.1:8000/docs> y debía aparecer algo así.



Pruebas: Isabela Martinez

Para poder hacer las pruebas, debíamos ir a POST del API, e ingresar los datos de una fila que teníamos en el *university_admission_train.csv* menos el dato de Admission Points y ejecutar la petición.

Prueba 1

Datos reales:

```
[
  {
    "Serial No.": 479, "GRE Score": 327, "TOEFL Score": 113, "University Rating": 4,
    "SOP": 4.0, "LOR ": 2.77, "CGPA": 8.88, "Research": 1, "Admission Points": 84.47}
]
```

Datos para probar:

```
[
  {
    "serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0,
    "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 0}
]
```

No parameters

Request body required application/json

```
{
  "serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0, "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 0
}
```

Execute Clear

Resultado

Code	Details						
200	<p>Response body</p> <pre>{ "resultado": [85.80514953017814] }</pre> <p>Response headers</p> <pre>content-length: 33 content-type: application/json date: Thu, 03 Nov 2022 01:48:11 GMT server: uvicorn</pre>						
<p>Responses</p> <table border="1"> <thead> <tr> <th>Code</th> <th>Description</th> <th>Links</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>Successful Response</td> <td>No links</td> </tr> </tbody> </table> <p>Media type application/json</p>		Code	Description	Links	200	Successful Response	No links
Code	Description	Links					
200	Successful Response	No links					

```
{
  "resultado": [
    85.80514953017814
  ]
}
```

Prueba 2

Datos reales:

```
[
  {"serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0,
  "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 84.47},
  {"serial_no": 446, "gre_score": 301, "toefl_score": 92, "university_rating": 1, "sop": 1.85,
  "lor": 1.5, "cgpa": 7.71, "research": 0, "admission_points": 45.08},
  {"serial_no": 336, "gre_score": 297, "toefl_score": 100, "university_rating": 1, "sop": 2.41,
  "lor": 1.59, "cgpa": 7.89, "research": 0, "admission_points": 47.42}
]
```

Datos para probar

```
[
  {"serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0,
  "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 0},

  {"serial_no": 446, "gre_score": 301, "toefl_score": 92, "university_rating": 1, "sop": 1.85,
  "lor": 1.5, "cgpa": 7.71, "research": 0, "admission_points": 0},

  {"serial_no": 336, "gre_score": 297, "toefl_score": 100, "university_rating": 1, "sop": 2.41,
  "lor": 1.59, "cgpa": 7.89, "research": 0, "admission_points": 0}
]
```

POST /predict Make Predictions

Parameters

No parameters

Request body required

application/json

```
{
  "serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0, "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 0},
  {"serial_no": 446, "gre_score": 301, "toefl_score": 92, "university_rating": 1, "sop": 1.85, "lor": 1.5, "cgpa": 7.71, "research": 0, "admission_points": 0},
  {"serial_no": 336, "gre_score": 297, "toefl_score": 100, "university_rating": 1, "sop": 2.41, "lor": 1.59, "cgpa": 7.89, "research": 0, "admission_points": 0}
}
```

Resultado

Code Details

200

Response body

```
{
  "resultado": [
    85.80514953017814,
    44.46514433014101,
    45.29338367062446
  ]
}
```

Response headers

```
content-length: 69
content-type: application/json
date: Thu, 03 Nov 2022 01:50:23 GMT
server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

```
{
  "resultado": [
    85.80514953017814,
    44.46514433014101,
    45.29338367062446
  ]
}
```

Prueba 3:

Datos reales:

```
[
  {"serial_no": 438, "gre_score": 317, "toefl_score": 71, "university_rating": 1, "sop": 4.15,
  "lor": 3.5, "cgpa": 7.65, "research": 1, "admission_points": 61.0},
  {"serial_no": 477, "gre_score": 298, "toefl_score": 104, "university_rating": 3, "sop": 2.5,
  "lor": 4.75, "cgpa": 8.12, "research": 0, "admission_points": 62.0},
  {"serial_no": 132, "gre_score": 311, "toefl_score": 95, "university_rating": 2, "sop": 1.46,
  "lor": 2.22, "cgpa": 7.87, "research": 0, "admission_points": 47.03}
  {"serial_no": 331, "gre_score": 327, "toefl_score": 84, "university_rating": 3, "sop": 3.5,
  "lor": 3.0, "cgpa": 8.66, "research": 1, "admission_points": 80.0},
  {"serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0,
  "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 84.47}
]
```

Datos para probar

```
[
  {"serial_no": 438, "gre_score": 317, "toefl_score": 71, "university_rating": 1, "sop": 4.15,
  "lor": 3.5, "cgpa": 7.65, "research": 1, "admission_points": 0},
  {"serial_no": 477, "gre_score": 298, "toefl_score": 104, "university_rating": 3, "sop": 2.5,
  "lor": 4.75, "cgpa": 8.12, "research": 0, "admission_points": 0},
  {"serial_no": 132, "gre_score": 311, "toefl_score": 95, "university_rating": 2, "sop": 1.46,
  "lor": 2.22, "cgpa": 7.87, "research": 0, "admission_points": 0}
  {"serial_no": 331, "gre_score": 327, "toefl_score": 84, "university_rating": 3, "sop": 3.5,
  "lor": 3.0, "cgpa": 8.66, "research": 1, "admission_points": 0},
  {"serial_no": 479, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0,
  "lor": 2.77, "cgpa": 8.88, "research": 1, "admission_points": 0}
]
```

The screenshot shows a REST client interface with a green header bar. The method is POST and the endpoint is /predict. The request body is a JSON array of five objects, each representing a student's data. The interface includes a 'Parameters' section (empty), a 'Request body' section with a dropdown set to 'application/json', and 'Cancel' and 'Reset' buttons.

```
POST /predict Make Predictions

Parameters
No parameters

Request body required application/json

[
  {
    "serial_no": 438,
    "gre_score": 317,
    "toefl_score": 71,
    "university_rating": 1,
    "sop": 4.15,
    "lor": 3.5,
    "cgpa": 7.65,
    "research": 1,
    "admission_points": 0
  },
  {
    "serial_no": 477,
    "gre_score": 298,
    "toefl_score": 104,
    "university_rating": 3,
    "sop": 2.5,
    "lor": 4.75,
    "cgpa": 8.12,
    "research": 0,
    "admission_points": 0
  },
  {
    "serial_no": 132,
    "gre_score": 311,
    "toefl_score": 95,
    "university_rating": 2,
    "sop": 1.46,
    "lor": 2.22,
    "cgpa": 7.87,
    "research": 0,
    "admission_points": 0
  },
  {
    "serial_no": 331,
    "gre_score": 327,
    "toefl_score": 84,
    "university_rating": 3,
    "sop": 3.5,
    "lor": 3.0,
    "cgpa": 8.66,
    "research": 1,
    "admission_points": 0
  },
  {
    "serial_no": 479,
    "gre_score": 327,
    "toefl_score": 113,
    "university_rating": 4,
    "sop": 4.0,
    "lor": 2.77,
    "cgpa": 8.88,
    "research": 1,
    "admission_points": 0
  }
]
```

Resultado

Code

Details

200

Response body

```
{
  "resultado": [
    59.19122295099604,
    59.62355286247208,
    52.655096888290075,
    77.83870165562027,
    85.80514953017814
  ]
}
```

Download

Response headers

```
content-length: 106
content-type: application/json
date: Thu, 03 Nov 2022 01:57:21 GMT
server: unicorn
```

Responses

Code	Description	Links
200	Successful Response	No links

```
{
  "resultado": [
    59.19122295099604,
    59.62355286247208,
    52.655096888290075,
    77.83870165562027,
    85.80514953017814
  ]
}
```

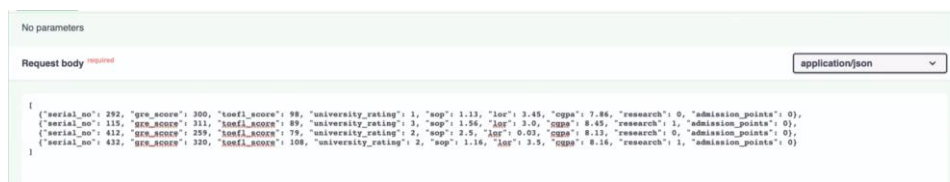
Prueba 4:

Datos reales:

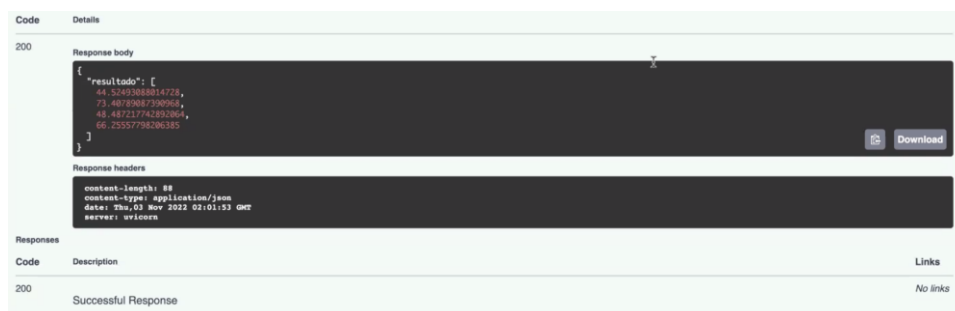
```
[
  {"serial_no": 292, "gre_score": 300, "toefl_score": 98, "university_rating": 1, "sop": 1.13,
  "lor": 3.45, "cgpa": 7.86, "research": 0, "admission_points": 50.5},
  {"serial_no": 115, "gre_score": 311, "toefl_score": 89, "university_rating": 3, "sop": 1.56,
  "lor": 3.0, "cgpa": 8.45, "research": 1, "admission_points": 59.0},
  {"serial_no": 412, "gre_score": 259, "toefl_score": 79, "university_rating": 2, "sop": 2.5,
  "lor": 0.03, "cgpa": 8.13, "research": 0, "admission_points": 56.0},
  {"serial_no": 432, "gre_score": 320, "toefl_score": 108, "university_rating": 2, "sop": 1.16,
  "lor": 3.5, "cgpa": 8.16, "research": 1, "admission_points": 73.0}
]
```

Datos a probar:

```
[
  {"serial_no": 292, "gre_score": 300, "toefl_score": 98, "university_rating": 1, "sop": 1.13,
  "lor": 3.45, "cgpa": 7.86, "research": 0, "admission_points": 0},
  {"serial_no": 115, "gre_score": 311, "toefl_score": 89, "university_rating": 3, "sop": 1.56,
  "lor": 3.0, "cgpa": 8.45, "research": 1, "admission_points": 0},
  {"serial_no": 412, "gre_score": 259, "toefl_score": 79, "university_rating": 2, "sop": 2.5,
  "lor": 0.03, "cgpa": 8.13, "research": 0, "admission_points": 0},
  {"serial_no": 432, "gre_score": 320, "toefl_score": 108, "university_rating": 2, "sop": 1.16,
  "lor": 3.5, "cgpa": 8.16, "research": 1, "admission_points": 0}
]
```



Resultado



```
{
  "resultado": [
    44.52493088014728,
    73.40789087390968,
    48.487217742892064,
    66.25557798206385
  ]
}
```

Prueba 5:

Datos reales:

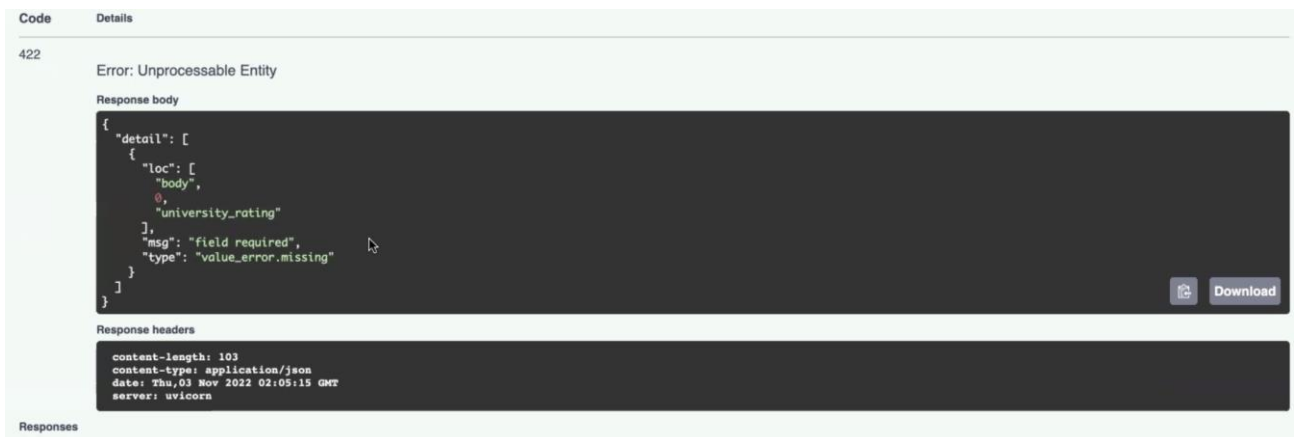
```
[
  {"serial_no": 292, "gre_score": 300, "toefl_score": 98, "university_rating": 1, "sop": 1.13,
  "lor": 3.45, "cgpa": 7.86, "research": 0, "admission_points": 50.5}
]
```

Datos a probar:

```
[
  {"serial_no": 292, "gre_score": 300, "toefl_score": 98, "University_rating": 1, "sop": 1.13,
  "lor": 3.45, "cgpa": 7.86, "research": 0, "admission_points": 50.5}
]
```



Resultado



```
{
  "detail": [
    {
      "loc": [
        "body",
        0,
        "university_rating"
      ],
      "msg": "field required",
```



```

    "type": "value_error.missing"
  }
]
}

```

Prueba de el retrain:

Request body required

application/json

```

{
  "serial_no": 380, "gre_score": 292, "toefl_score": 105, "university_rating": 2, "sop": 2.0, "lor": 2.91, "cgsa": 7.83, "research": 0, "admission_points": 46.06}, {"serial_no": 90, "gre_score": 316, "toefl_score": 85, "university_rating": 4, "sop": 0.24, "lor": 0.23, "cgsa": 8.76, "research": 1, "admission_points": 74.0}, {"serial_no": 345, "gre_score": 264, "toefl_score": 115, "university_rating": 2, "sop": 1.5, "lor": 2.0, "cgsa": 7.34, "research": 0, "admission_points": 47.0}, {"serial_no": 420, "gre_score": 292, "toefl_score": 100, "university_rating": 3, "sop": 2.36, "lor": 2.76, "cgsa": 7.48, "research": 0, "admission_points": 47.08}, {"serial_no": 377, "gre_score": 297, "toefl_score": 96, "university_rating": 2, "sop": 0.73, "lor": 2.3, "cgsa": 7.43, "research": 0, "admission_points": 34.0}, {"serial_no": 379, "gre_score": 303, "toefl_score": 71, "university_rating": 1, "sop": 2.1, "lor": 2.5, "cgsa": 7.65, "research": 0, "admission_points": 56.0}, {"serial_no": 178, "gre_score": 319, "toefl_score": 113, "university_rating": 3, "sop": 3.5, "lor": 2.83, "cgsa": 7.46, "research": 0, "admission_points": 82.0}, {"serial_no": 37, "gre_score": 321, "toefl_score": 110, "university_rating": 5, "sop": 4.42, "lor": 4.82, "cgsa": 9.24, "research": 1, "admission_points": 89.47}, {"serial_no": 297, "gre_score": 313, "toefl_score": 98, "university_rating": 1, "sop": 1.63, "lor": 1.67, "cgsa": 7.82, "research": 0, "admission_points": 50.5}, {"serial_no": 426, "gre_score": 323, "toefl_score": 111, "university_rating": 5, "sop": 2.15, "lor": 5.0, "cgsa": 9.86, "research": 1, "admission_points": 138.0}, {"serial_no": 492, "gre_score": 297, "toefl_score": 99, "university_rating": 4, "sop": 2.86, "lor": 1.99, "cgsa": 7.81, "research": 0, "admission_points": 54.0}, {"serial_no": 288, "gre_score": 324, "toefl_score": 97, "university_rating": 5, "sop": 4.63, "lor": 4.5, "cgsa": 9.08, "research": 1, "admission_points": 89.0}, {"serial_no": 59, "gre_score": 300, "toefl_score": 99, "university_rating": 1, "sop": 4.32, "lor": 2.0, "cgsa": 6.8, "research": 1, "admission_points": 36.0}, {"serial_no": 227, "gre_score": 306, "toefl_score": 110, "university_rating": 2, "sop": 3.5, "lor": 4.0, "cgsa": 8.45, "research": 0, "admission_points": 63.0}, {"serial_no": 95, "gre_score": 302, "toefl_score": 111, "university_rating": 2, "sop": 4.43, "lor": 3.0, "cgsa": 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, "cgsa": 9.11, "research": 1, "admission_points": 89.0}, {"serial_no": 107, "gre_score": 313, "toefl_score": 114, "university_rating": 3, "sop": 2.85, "lor": 2.71, "cgsa": 7.7, "research": 0, "admission_points": 39.6}, {"serial_no": 35, "gre_score": 331, "toefl_score": 114, "university_rating": 4, "sop": 4.4, "lor": 1.35, "cgsa": 9.74, "research": 0, "admission_points": 90.36}, {"serial_no": 119, "gre_score": 296, "toefl_score": 99, "university_rating": 2, "sop": 3.0, "lor": 3.5, "cgsa": 7.28, "research": 0, "admission_points": 47.0}, {"serial_no": 59, "gre_score": 299, "toefl_score": 98, "university_rating": 1, "sop": 3.0, "lor": 6.82, "research": 1, "admission_points": 40.03}, {"serial_no": 33, "gre_score": 334, "toefl_score": 100, "university_rating": 4, "sop": 3.0, "lor": 0.18, "cgsa": 9.4, "research": 1, "admission_points": 91.0}, {"serial_no": 89, "gre_score": 286, "toefl_score": 113, "university_rating": 2, "sop": 3.0, "lor": 2.67, "cgsa": 7.77, "research": 1, "admission_points": 66.0}, {"serial_no": 172, "gre_score": 334, "toefl_score": 113, "university_rating": 5, "sop": 4.0, "lor": 4.5, "cgsa": 6.58, "research": 1, "admission_points": 89.0}, {"

```

Execute

Clear

```

{"RMSE-train": 113.66835435009195, "MAE-train": 7.9714492090990525, "R^2-train": 0.7058791918933558, "RMSE-test": 117.7401704907249, "MAE-test": 8.226726016409662, "R^2-test": 0.6475298355832665}

```

Responses

Curl

```

curl -X 'POST' \
  'http://127.0.0.1:8000/retrain' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{"serial_no": 470, "gre_score": 327, "toefl_score": 113, "university_rating": 4, "sop": 4.0, "lor": 2.77, "cgsa": 8.88, "research": 1, "admission_points": 84.47}, {"serial_no": 440}'

```

Request URL

http://127.0.0.1:8000/retrain

Server response

Code

Details

200

Response body

```

{"RMSE-train": 113.66835435009195, "MAE-train": 7.9714492090990525, "R^2-train": 0.7058791918933558, "RMSE-test": 117.7401704907249, "MAE-test": 8.226726016409662, "R^2-test": 0.6475298355832665}

```

Response headers

```

content-length: 242
content-type: application/json
date: Thu, 03 Nov 2022 02:10:47 GMT
server: uvicorn

```

Responses

Code

Description

Links

200

Successful Response

No links

Media type

application/json

Análisis de las pruebas:

Las tres primeras pruebas fueron exitosas ya que el dato a predecir (Admission Points) fue muy cercano al dato real. Por ejemplo, en la primera prueba el dato real es de 84.47 y el resultado de la prueba fue de 85.8.

Por otro lado, la prueba número 4, contiene resultados incoherentes ya que poseen un margen de error mucho más alta. Pongamos el caso de:

```
{"serial_no": 115, "gre_score": 311, "toefl_score": 89, "university_rating": 3, "sop": 1.56, "lor": 3.0, "cgpa": 8.45, "research": 1, "admission_points": 59.0}
```

Y el resultado de la prueba fue de 73.40789087390968.

Para la prueba 5 el error fue gracias a que se ingresó mal la prueba, pues se puso University_rating y debíamos escribir university_rating ya que en el DataModel especificamos que la variable debía ir sin minúsculas.

En la prueba de retrain es donde re-entrenamos el modelo y obtenemos las métricas de desempeño del modelo.

Estrategias de mitigación

En general, cuando se trabaja con datos que son tomados de Usuarios siempre existe la posibilidad de incurrir en un error debido a que a veces se presentan en diferentes formatos y los modelos trabajan con uno específico. Es decir, una columna que indica un puntaje podría ser "Ciento trece" pero en realidad se busca que el dato suministrado haya sido el número "113", no obstante, el significado es el mismo. Por lo tanto, para mitigar incoherencias y errores de predicción se trata de ser lo más específico a la hora de pedir inputs de cara a un modelo, además se recomienda que se contemplen los casos más frecuentes en los que se incurre, dependiendo del tipo de dato.