

## Laboratorio 4

### Contenido

Laboratorio 4.....	1
Descripción del negocio .....	2
Escenarios de Prueba .....	2
Estrategia de mitigación de errores .....	10
Bono .....	10
Desarrollo de actividades .....	14

### Tabla de ilustraciones

Ilustración 1 Diccionario del negocio .....	2
Ilustración 2 Environment localhost .....	3
Ilustración 3 Get localhost.....	3
Ilustración 4 Post datos correctos.....	4
Ilustración 5 post datos correctos 50.....	5
Ilustración 6 post datos correctos 100.....	5
Ilustración 7 post datos Incoherentes.....	6
Ilustración 8 post datos incoherentes 10.....	6
Ilustración 9 post datos incoherentes 20.....	7
Ilustración 10 post datos incoherentes 50.....	8
Ilustración 11 post tipos datos incorrectos.....	9
Ilustración 12 post tipos datos incorrectos palabras .....	9
Ilustración 13 reentrenamiento .....	10
Ilustración 14 prueba con datos negativos .....	11
Ilustración 15 prueba con datos positivos GRE.....	11
Ilustración 16 prueba con datos positivos TOEFL .....	11
Ilustración 17 prueba con datos positivos University Rating.....	12
Ilustración 18 prueba con datos positivos SOP .....	12
Ilustración 19 prueba con datos positivos LOR.....	12
Ilustración 20 prueba con datos positivos CGPA .....	13
Ilustración 21 prueba con datos positivos Research.....	13
Ilustración 22 log del servidor .....	14

## Descripción del negocio

La Universidad de los Alpes es una institución educativa de alta calidad y una de las más prestigiosas de su región, esta universidad nos contrató para realizar un modelo que permita estimar el puntaje de admisión de un aspirante a maestría.

Para el modelo realizado se utilizaron datos de variables que podemos observar en el siguiente diccionario:

Variable	Descripción
Serial No.	Application id. Unique to each applicant
GRE Score	The Graduate Record Examination. An exam designed to measure overall academic readiness. 0 to 340
TOEFL Score	Score in the Test of English as a Foreign Language obtained by the applicant. 0 to 120
University Rating	Quality of the university where the applicant graduated. 0 to 5
SOP	Statement of Purpose Strength. 0 to 5
LOR	Letter of recommendation Strength. 0 to 5
CPGA	Cumulative Grade Point Average. An score to measure academic performance in bachelor programs. 0 to 10
Research Experience	Does the applicant have research experience?. Either 0 or 1
Admission Points	Points obtained by the applicant in the university Masters Programs admission exam. 0 to 150

*Ilustración 1 Diccionario del negocio*

Después de la realización del modelo de regresión lineal la Universidad de los Alpes nos ha pedido construir un Pipeline y subirlo a un servidor que le permita consumir el modelo mediante solicitudes HTTP.

## Escenarios de Prueba

Para realizar las pruebas primero cargamos el servidor local que tiene la URL <http://127.0.0.1:8000>, estos datos de URL se guardaron en un environment de PostMan para darle mayor facilidad al usuario que quiere realizar las pruebas de nuevo.



Ilustración 2 Environment localhost

Primero solo para probar que el servidor está funcionando vamos a hacer un get.

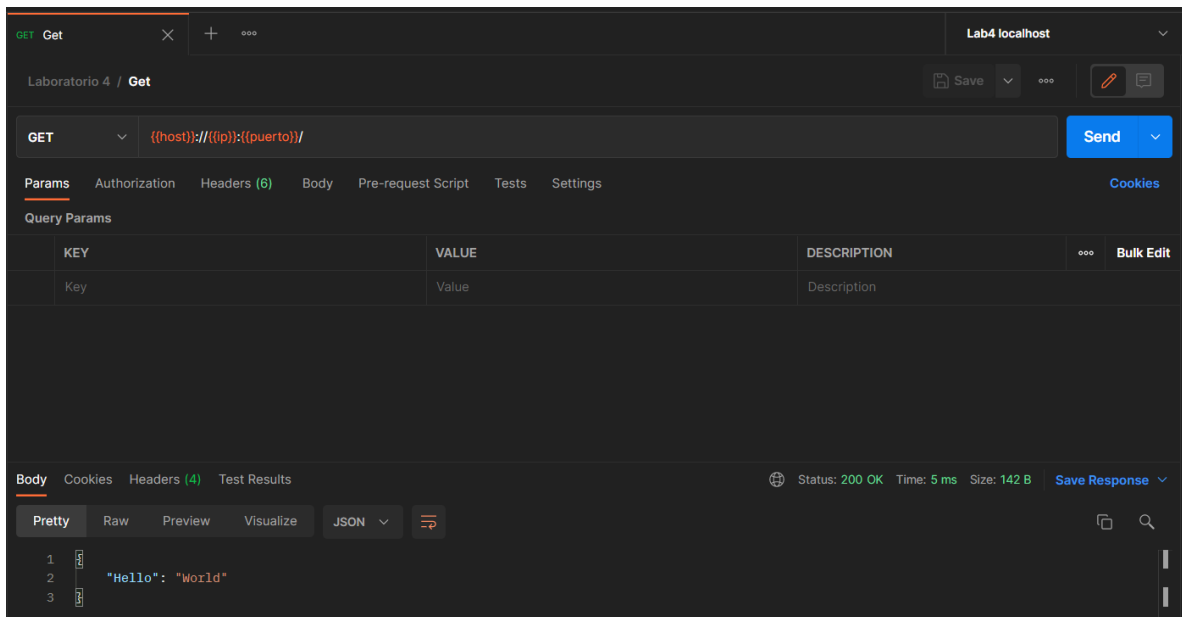
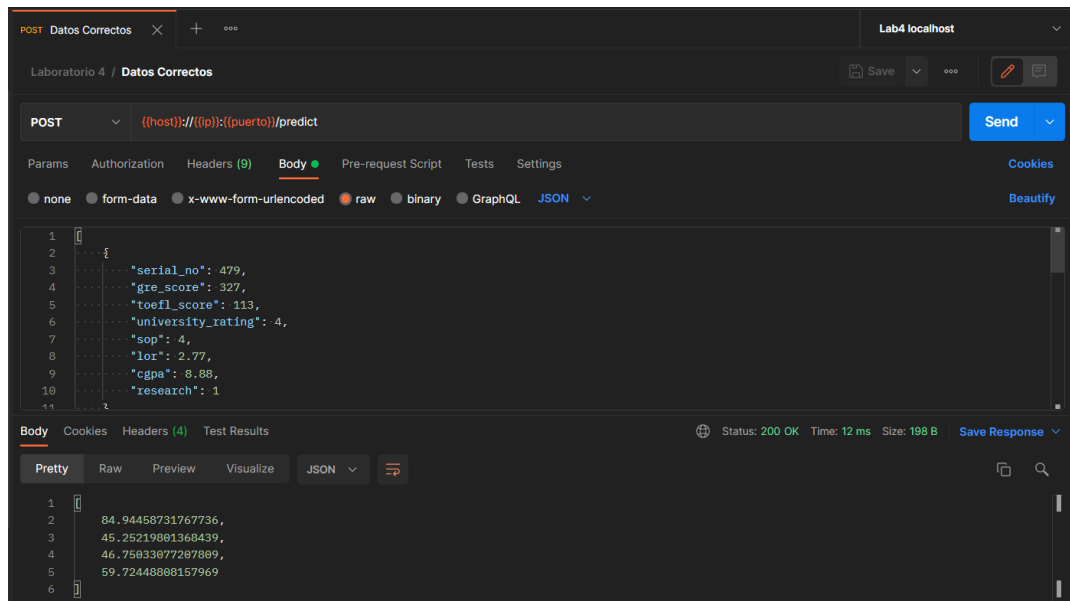


Ilustración 3 Get localhost

Ahora si continuamos con las diferentes pruebas:

1. Se realizó una prueba de post con 4 diccionarios de datos obtenidos directamente del diccionario que se utiliza para en entrenamiento del modelo, de esta manera podemos probar que los resultados tengan sentido porque sabemos los valores originales de la variable objetivo. Para esta prueba se utilizó el archivo JSON de datosCorrectos y se esperan valores similares a 84.47, 45.08, 47.42, 62.



*Ilustración 4 Post datos correctos*

Como podemos observar los valores obtenidos por la petición son muy cercanos a los valores esperados de la variable objetivo, por lo cual podemos observar que el modelo está funcionando correctamente con datos coherentes.

2. Se realizaron 2 pruebas de post con 50 y 100 datos respectivamente, estos datos fueron obtenidos utilizando un random de valores en Python pensando en el rango que se define en el negocio para cada variable (ver Ilustración 1) y a su vez que estos valores sean cercanos a la media de los valores en los datos de entrenamiento. Para esta prueba se utilizaron los archivos JSON de datosCorrectos50 y datosCorrectos100.

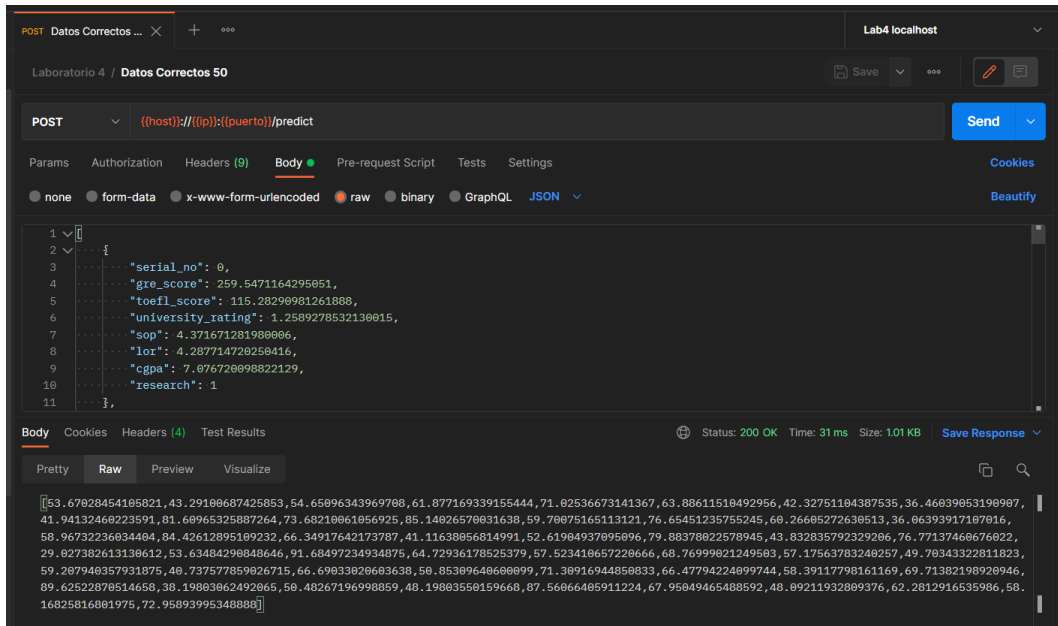


Ilustración 5 post datos correctos 50

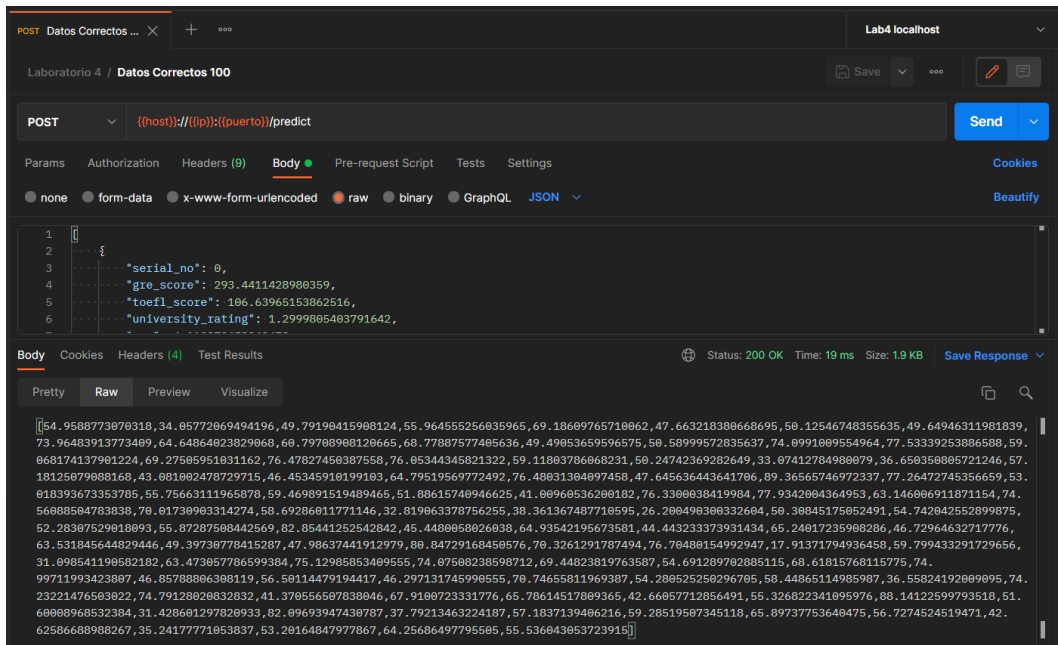


Ilustración 6 post datos correctos 100

Aunque para esta prueba ya no se tienen valores esperados de la variable objetivo podemos verificar que los resultados tienen sentido porque todos los resultados son positivos y están dentro del rango de la variable objetivo (0 a 150).

- Se realizó una prueba de post con un diccionario con todos los valores en 0, esta prueba debería devolver un dato sin sentido. Para esto se utilizó el archivo JSON datosIncoherentes.

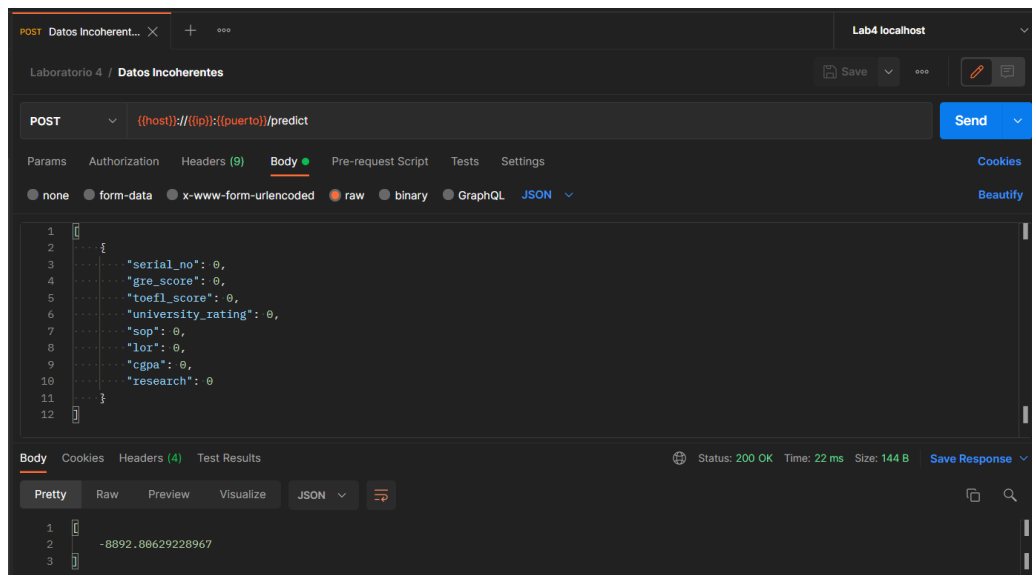


Ilustración 7 post datos Incoherentes

Como podemos observar la prueba devuelve un número negativo muy grande lo cual tiene sentido porque los valores en la prueba son solo 0.

- Se realizó una prueba de post con 10 datos, estos datos fueron obtenidos utilizando un random de valores en Python pensando en utilizar valores en el rango que se define en el negocio para cada variable (ver Ilustración 1) pero que estos valores sean muy pequeños y alejados del promedio. Para esta prueba se utilizó el archivo JSON datosIncoherentes10.

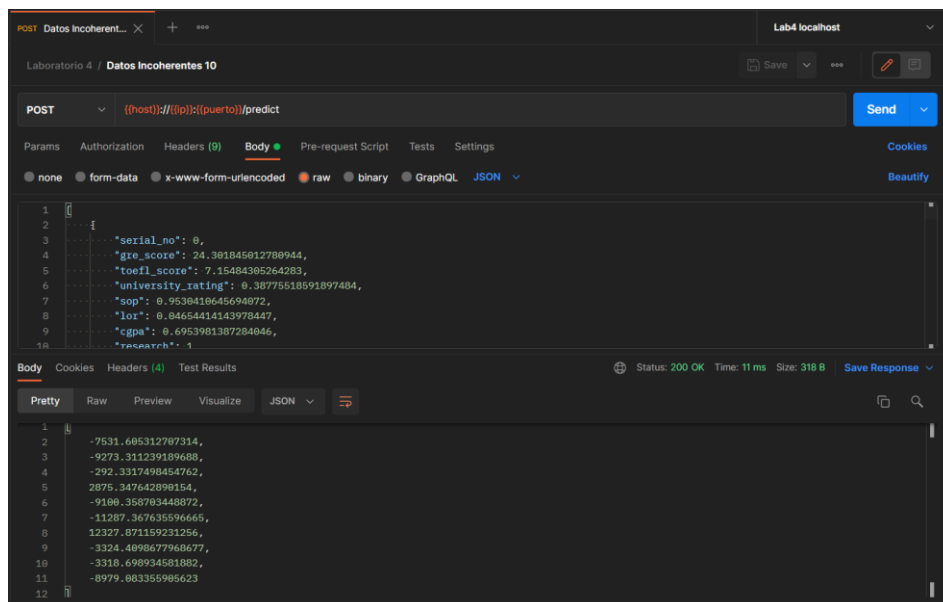


Ilustración 8 post datos incoherentes 10

Para esta prueba podemos observar que los valores, aunque están dentro del rango del negocio son muy pequeños y no tienen sentido para un caso realista de análisis, por esta misma razón los resultados no están dentro del rango de la variable objetivo.

- Se realizó una prueba de post con 20 datos, estos datos fueron obtenidos utilizando un random de valores en Python utilizando valores negativos fuera del rango que se define en el negocio para cada variable (ver Ilustración 1). Para esta prueba se utilizó el archivo JSON datosIncoherentes20.

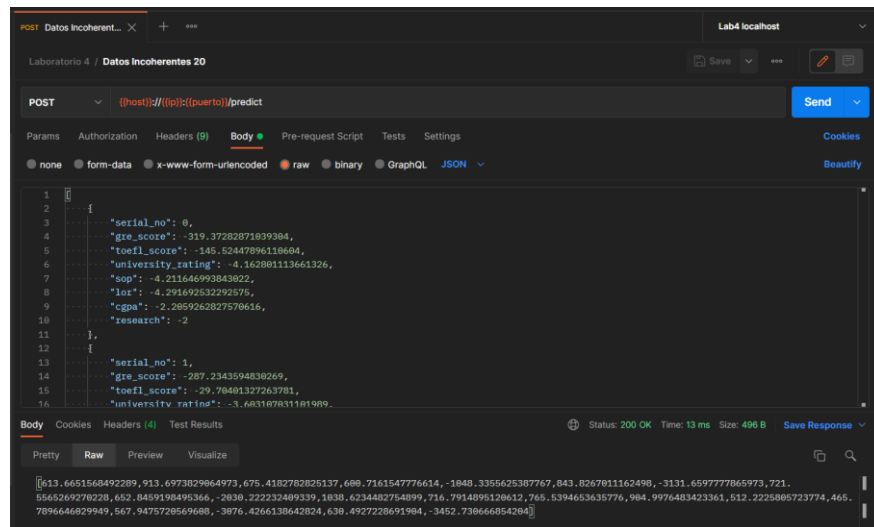


Ilustración 9 post datos incoherentes 20

Para esta prueba podemos observar que como los valores son negativos y están fuera de los rangos del negocio obtenemos resultados que no están en el rango de la variable objetivo.

- Se realizó una prueba de post con 50 datos, estos datos fueron obtenidos utilizando un random de valores en Python utilizando valores positivos fuera del rango que se define en el negocio para cada variable (ver Ilustración 1). Para esta prueba se utilizó el archivo JSON datosIncoherentes50.

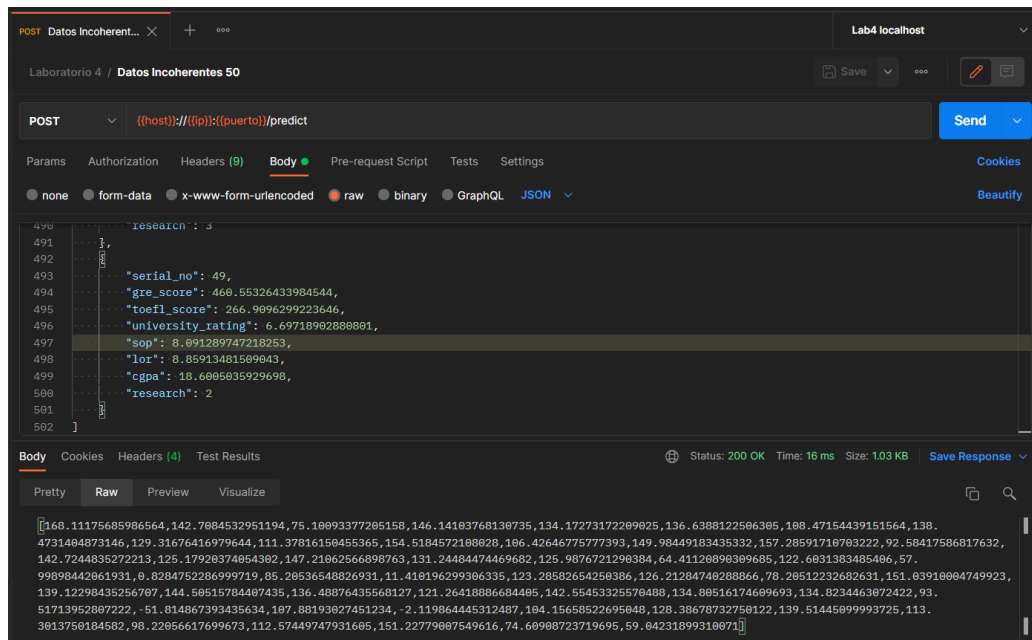


Ilustración 10 post datos incoherentes 50

Para esta prueba podemos observar que los valores resultados en muchos casos están en el rango de la variable objetivo pero son muy altos, esto se debe a que los datos son positivos, son muy altos porque están fuera del rango positivo de las variables y aún no se ha realizado la mitigación de incoherencias para evitar que se ingresen valores fuera de los rangos definidos por el negocio.

7. Se realizó una prueba de post con 10 datos, estos datos fueron obtenidos utilizando un random de valores en Python pensando en el rango que se define en el negocio para cada variable (ver Ilustración 1) y a su vez que estos valores sean cercanos a la media de los valores en los datos de entrenamiento, pero se convirtieron a string todos los valores. Para esta prueba se utilizó el archivo JSON tiposDatosIncorrectos.



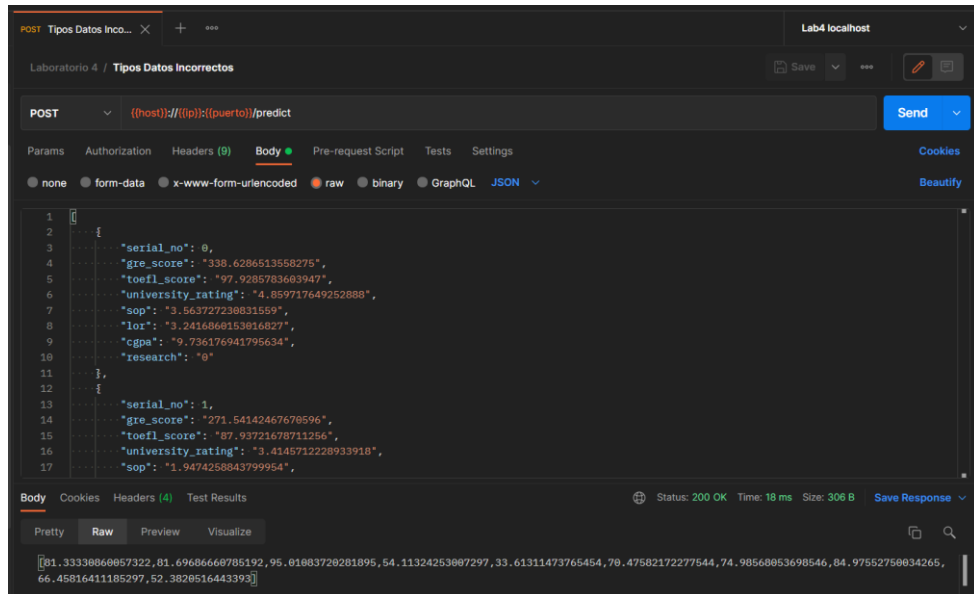


Ilustración 11 post tipos datos incorrectos

En esta prueba podemos observar que aunque los tipos de datos que se entrega al modelo son Strings, este devuelve resultados con sentido y dentro del rango de la variable objetivo porque los valores aunque no tienen el tipo de datos correcto la conversión a String se hizo sobre un valor float correcto.

8. Se realizó una prueba de post con 10 datos, estos datos fueron obtenidos manualmente agregando algunos Strings de palabras en los valores. Para esta prueba se utilizó el archivo JSON tiposDatosIncorrectosPalabras.

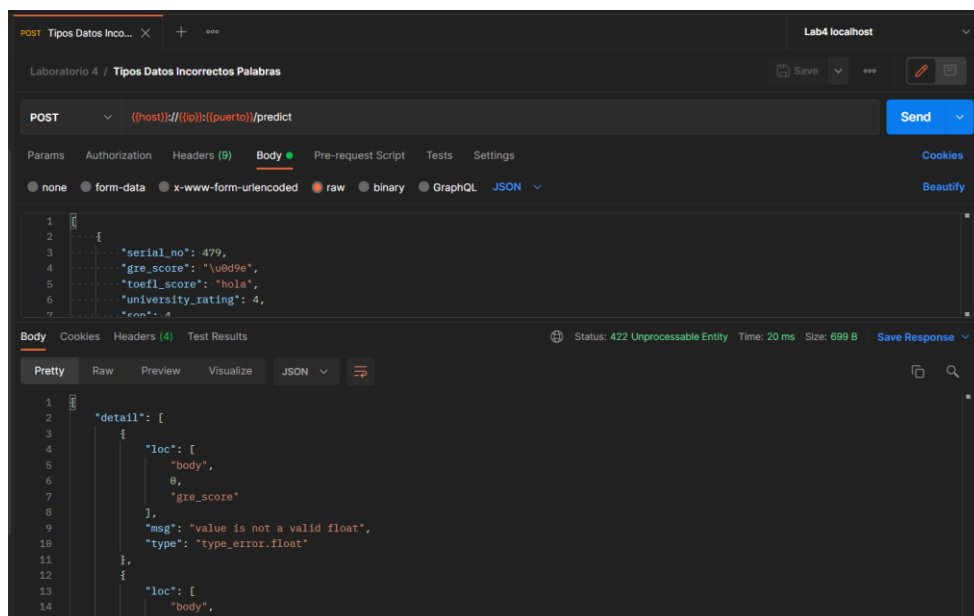


Ilustración 12 post tipos datos incorrectos palabras

Para esta prueba podemos observar que tal y como se esperaba se obtiene un código de error 422 Unprocessable Entity porque se enviaron Strings que además son palabras y el modelo no puede procesar esto.

9. Para la última prueba se utilizó una parte del csv original para realizar un re-entrenamiento del modelo.

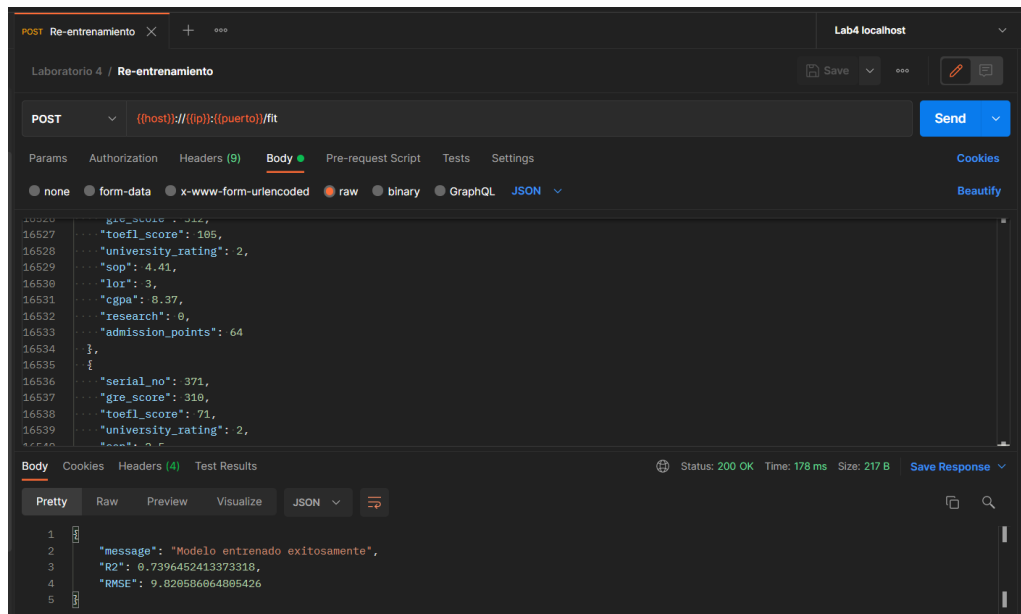


Ilustración 13 reentrenamiento

Para esta prueba podemos observar una respuesta correcta del servidor para el reentrenamiento.

## Estrategia de mitigación de errores

Revisar los inputs del usuario de acuerdo con las reglas de negocio de la Universidad de los Alpes (Revisar ilustración 1). Es decir, revisar cada valor correspondiente y que se encuentre en el rango correcto, mandaríamos un código de error de usuario HTTP para informarle al usuario que la petición que hizo tiene un error, en este caso utilizaríamos el error 400 Bad Request.

## Bono

Para el bono desplegamos un servidor en Compute Engine de Google Cloud Platform. Al servidor se puede acceder por medio de la dirección <http://34.72.86.170/>

En el README del repositorio se encuentran las rutas del servidor para realizar las requests al API.

Se implementó la estrategia de mitigación de errores al API junto con errores HTTP que mostraran en qué se estaba equivocando el usuario al introducir los datos.

Para probar la estrategia de mitigación de errores se utilizaron las pruebas de la Collection que tenían valores fuera de los rangos (negativos y positivos).

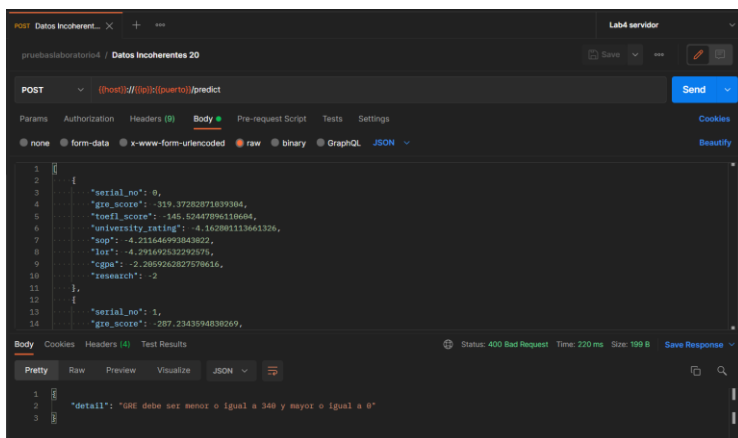


Ilustración 14 prueba con datos negativos

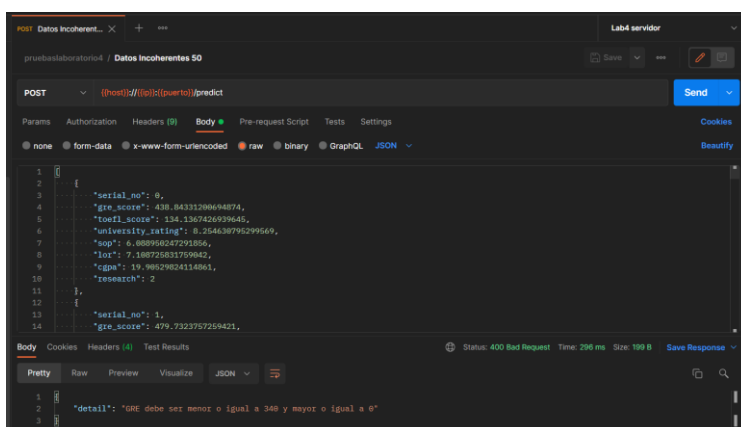


Ilustración 15 prueba con datos positivos GRE

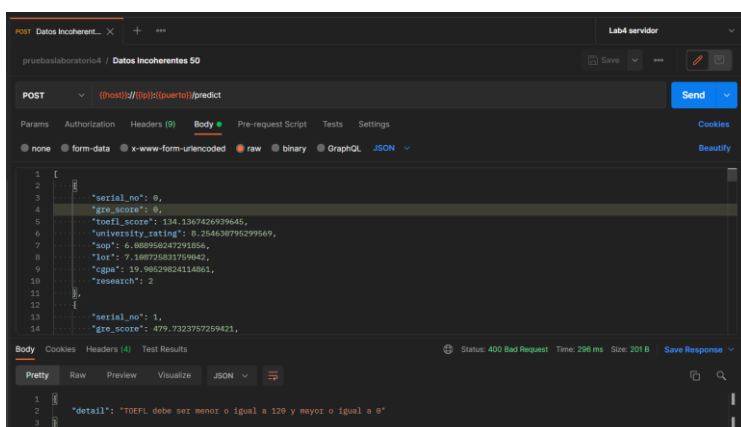


Ilustración 16 prueba con datos positivos TOEFL

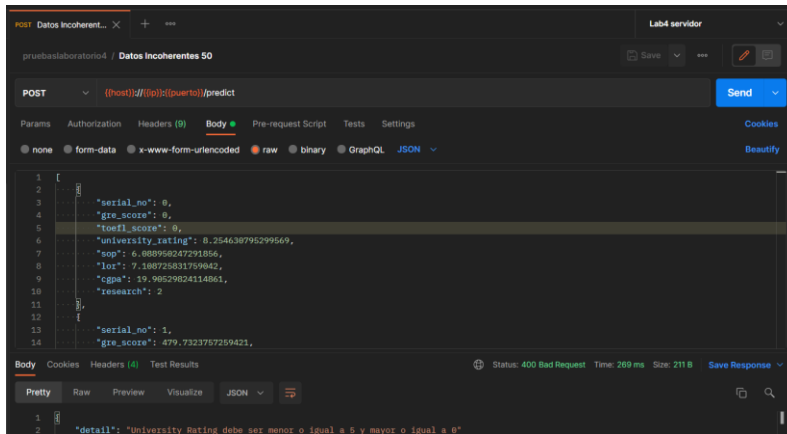


Ilustración 17 prueba con datos positivos University Rating

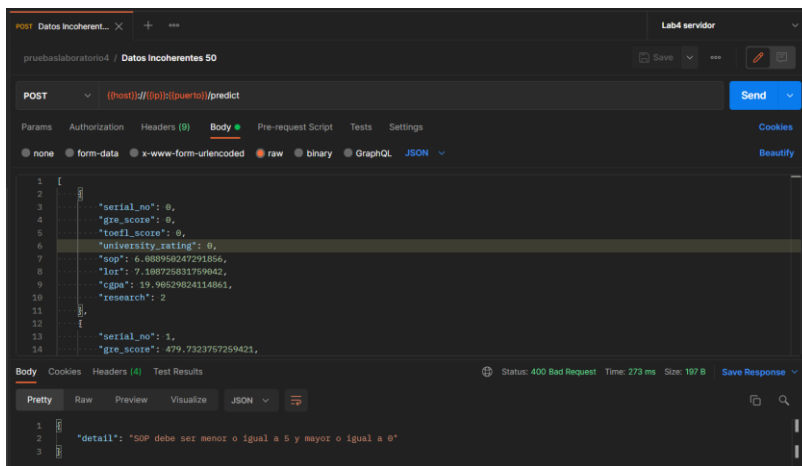


Ilustración 18 prueba con datos positivos SOP

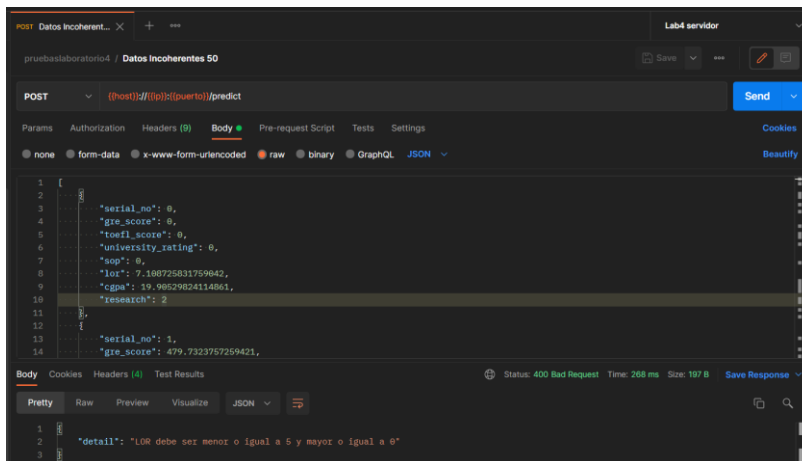
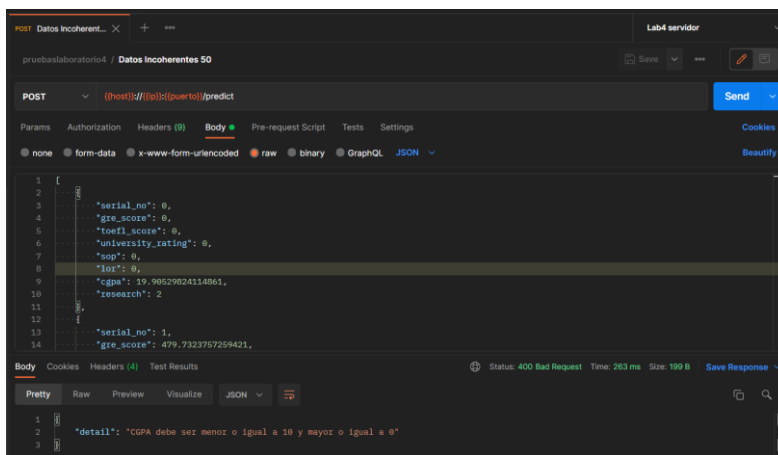
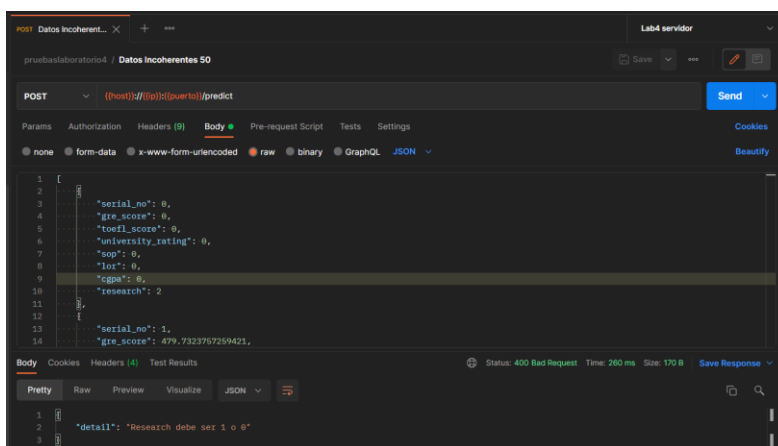


Ilustración 19 prueba con datos positivos LOR



*Ilustración 20 prueba con datos positivos CGPA*



*Ilustración 21 prueba con datos positivos Research*

Tal y como se esperaba después de la estrategia de mitigación estas pruebas no dan un resultado, devuelven un error indicando que los valores están fuera del rango del negocio para cada variable.

Aquí se encuentra una parte del archivo de log del servidor que muestra que se están recibiendo las peticiones y los mensajes de respuesta de acuerdo con si la petición fue hecha de manera correcta o no

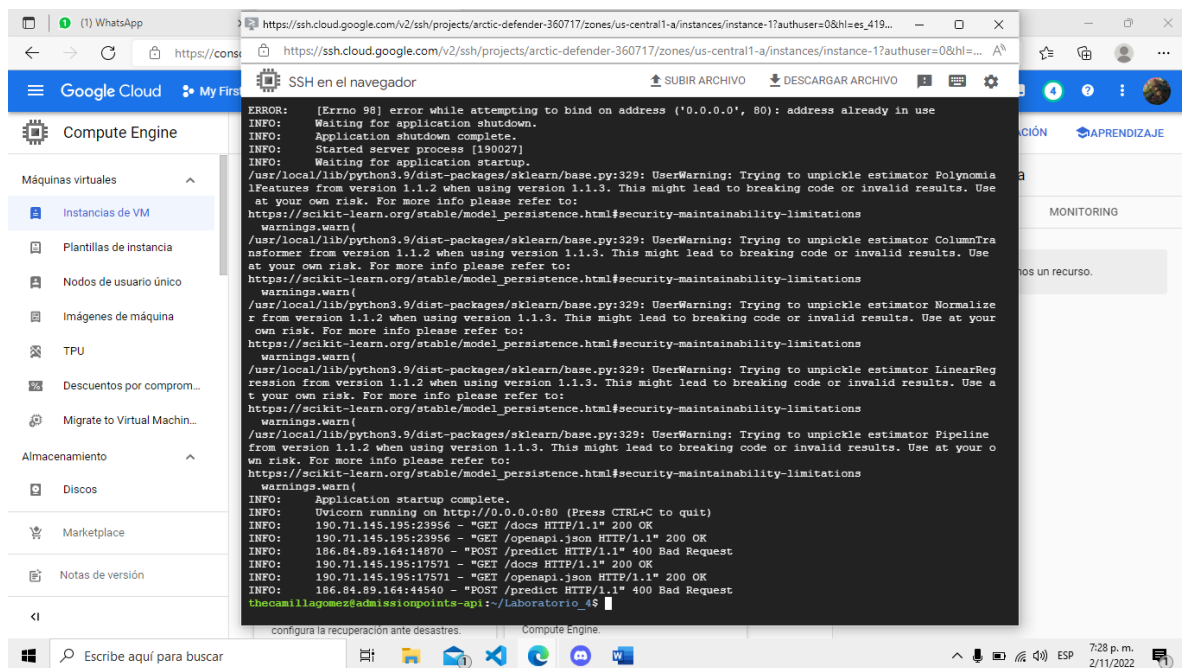


Ilustración 22 log del servidor

## Desarrollo de actividades

Cada integrante del grupo hizo un porcentaje equitativo para cada una de las actividades, junto con el bono.

## Anexos

- En la carpeta assets del repositorio se encuentran los diferentes .json usados para las pruebas y la collection de postman (assets\Lab4localhost.postman\_environment.json) en la que se encuentran todas las pruebas utilizadas para el laboratorio, de la misma manera se encuentran los environments que se pueden usar para postman.
- En la carpeta Pipeline se encuentra el jupyter con la creación del pipeline y en assets se encuentra el pipeline (assets\modelo.joblib).
- En la carpeta principal se encuentra el README con las guías para correr el programa.
- El repositorio del proyecto lo pueden encontrar en la siguiente ruta:

[https://github.com/Gatos-de-negocios-BI/Laboratorio\\_4](https://github.com/Gatos-de-negocios-BI/Laboratorio_4)