Modelado Predictivo

Práctica 3. *Regresión lineal de datos y gradiente descendente*

Objetivo. Introducir al alumno a la tarea de regresión mediante el uso de modelos lineales.

Habilidades para la industria. Manejo de entornos virtuales, desarrollo de servicios web, test unitarios, configuración básica de servidores.

Instrucciones.

1. Este paso no es necesario, pero es sugerido. Active su ambiente virtual creado en la práctica anterior. (**Tip**: Utilice los comandos *conda create* y *conda activate*).

2. Instale los paquetes de FastAPI y Uvicorn dentro del ambiente virtual. (**Tip**: Utilice los comandos sugeridos aqui https://fastapi.tiangolo.com/#installation). Si lo desea puede utilizar Flask, que también es un API para desarrollo de servicios web o cualquier otra de su preferencia.

3. Para todo requerimiento, en la carpeta que sea la raiz de su proyecto deberá incluir un archivo de requerimientos llamado *requirements.txt* (https://pip.pypa.io/en/stable/reference/requirements-file-format/) que será utilizado para lainstalación de requerimientos enlos unittests.

4. Siguiendo la guía oficial del framework/API seleccionado (Para FastAPI puedeir a https://fastapi.tiangolo.com/tutorial/first-steps/), desarrolle un servicio web, con autenticación, con los siguientes endpoints:

   - Primer endpoint

Método	POST
Endpoint	/linear/regression/train
Entrada y salida	application/json
Parámetros	str data_url: Liga de descarga de archivo. Será la liga de un CSV. El valor a predecir vendrá contenida en la columna ***prediction***
	float learning_rate: Tasa de aprendizaje utilizada para el algoritmo de gradiente descendente estocástico.
	int iterations: Número de iteraciones que se ejecutará el algoritmo de gradiente descendente estocástico.
Funcionalidad	Desde la URL recibida como parámetro, descargue el archivo y utilizando el algoritmo de gradiente descendente estocástico entrene un modelo de regresión lineal. El número de características puede variar, por lo que es requerido que haga la detección del número de características para seleccionar la dimensionalidad correcta del modelo.
Respuesta	List[float]: model_weights: Los pesos del modelo entrenado.

   - Segundo endpoint

Método	POST
Endpoint	/linear/regression/sgd/predict
Entrada y salida	application/json
Parámetros	List[float] model_weights: Pesos utilizados para realizar la predicción.
	List[List[float]] input_data: Datos que serán requeridos predecir.
Funcionalidad	Utilizando los pesos recibidos como parámetro, realice la predicción del valor para los datos recibidos como parámetro.
Respuesta	List[float]: prediction: Lista de valores asignados a cada uno de los puntos de datos recibidos como parámetros, en el mismo orden que como fueron recibidos.

5. La autenticación debe ser de tipo OAuth2 usando token Bearer, que acepte como usuario «TestUser» y contraseña «&TestUserPassword123.» bajo el endpoint **/token** (si usa FastAPI refiera al siguiente enlace https://fastapi.tiangolo.com/tutorial/security/). Los endpoints relacionados a la regresión únicamente podrán ser consumidos si el usuario inicio sesión anteriormente.

6. De acuerdo a las validaciones que considere necesarias, los códigos HTTP de respuesta pueden ser 2XX o 4XX (usualmente), pero puede revisar el uso de códigos de errores en la liga https://en.wikipedia.org/wiki/List_of_HTTP_status_codes. Su servicio web **NO** debe responder en ningun caso un error de tipo 5XX.

7. Considere realizar tests unitarios para probar múltiples casuisticas de sus endpoints.

8. Dado que tiene la libertad de elegir el framework/API de su preferencia, es necesario poder inicializar el servidor de manera «genérica», para ello, en la raiz de su proyecto incluya un script llamado *start.py* que permita inicializar el servidor que expone su servicio web.

9. Su sevidor debe aceptar las peticiones de manera local (en la dirección IP 127.0.0.1) y en el puerto 8125.

10. Cree un repositorio en git y suba el proyecto. El repositorio no necesariamente debe ser para esta práctica específica, puede crear un repositorio para la materia. Para la evaluación se permitirá considerar el directorio raiz del proyecto como un subdirectorio del repositorio.

11. Para la entrega de la práctica, debe enviar un correo a la dirección **uriel.personal.systems@gmail.com**, el ausnto del correo no es relevante, pero el cuerpo del correo debe contener la siguiente información:

    ID=MPB22P3

    GIT_URL=<La URL de su repositorio en donde se encuentra el proyecto>

    GIT_SUBFOLDER=<La ruta dentro del repositorio en donde se encuentra el proyecto>

    GIT_BRANCH=<La rama en la que se encuentra el proyecto>

    El correo y el repositorio debe cumplir con los siguientes criterios:

    - El correo debe venir de su correo registrado en el SAES o el proporcionado para la materia pues será validado contra una lista de correos permitidos.

    - La URL del git debe ser el formato en clonación de HTTPS y debe poder clonarse sin reestricción (debe ser público).

    - Los requerimientos deben poder instalarse desde la ruta proporcionada en GIT_SUBFOLDER dentro del archivo *requirements.txt*.

    - El servidor debe poder inicializarse desde la ruta proporcionada en GIT_SUBFOLDER.

    - Los resultados de los tests serán notificados al mismo correo electrónico de donde fue enviado el correo y desde la misma dirección a donde fue enviado.