

Análisis de Series de Tiempo

Práctica 2. Eliminación de tendencia por regresión y diferencia

Objetivo. Introducir al alumno al preprocesamiento de series de tiempo.

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 aquí <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 raíz 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 la instalación de requerimientos en los unit tests.
4. Siguiendo la guía oficial del framework/API seleccionado (Para FastAPI puede ir a <https://fastapi.tiangolo.com/tutorial/first-steps/>), desarrolle un servicio web, sin autenticación, con los siguientes endpoints:

- Primer endpoint

Método	POST
Endpoint	/trend/detrending/linear
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
Funcionalidad	Desde la URL recibida como parámetro, descargue el archivo y utilizando la solución analítica de regresión calcule los valores de los coeficientes óptimos. El número de series de tiempo puede variar, por lo que es requerido que haga la detección del número de las mismas (columnas en el CSV menos la columna de predicción) para seleccionar la dimensionalidad correcta del modelo. Adicionalmente, calcule la función de autocorrelación.
Respuesta	List[float]: model_weights: Los valores de los coeficientes óptimos. List[float]: sample_acf: Valores de autocorrelación para lags entre 0 y 50 una vez aplicado el detrending.

- Segundo endpoint

Método	POST
Endpoint	/trend/difference/
Entrada y salida	application/json
Parámetros	List[float] data: Lista que contiene los valores de la serie de tiempo
Funcionalidad	Utilizando los datos recibidos como parámetro, realice la eliminación de tendencia por el método de diferencia y calcule la función de autocorrelación.
Respuesta	List[float]: sample_acf: Valores de autocorrelación para lags entre 0 y 50 una vez aplicada la diferencia.

5. 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.
6. Para estandarizar sus pruebas se recomienda hacer uso de tests unitarios. Esta práctica será evaluada con el uso de los mismos en diferentes casos de uso. Sírvese de guía la siguiente liga <https://docs.python.org/3/library/unittest.html>.
7. 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 raíz de su proyecto incluya un script llamado *start.py* que permita inicializar el servidor que expone su servicio web.
8. Su servidor debe aceptar las peticiones de manera local (en la dirección IP 127.0.0.1) y en el puerto 8125.
9. 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 raíz del proyecto como un subdirectorío del repositorio.
10. Para la entrega de la práctica, debe enviar un correo a la dirección **uriel.personal.systems@gmail.com**, el asunto del correo no es relevante, pero el cuerpo del correo debe contener la siguiente información:

ID=STP2

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 restricción (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.