

Module 6 - Homework

Task description

FastAPI: <https://fastapi.tiangolo.com/>

Crea una API que habilite un modelo de predicción de precios de la compra. La API recibirá una petición HTTP con el ID de usuario y responderá con la predicción del precio de la compra. La API necesitará dos componentes: i) el feature store (parquets en `s3://zrive-ds-data/groceries/sampled-datasets/`), y ii) el modelo serializado (`s3://zrive-ds-data/groceries/trained-models/model.joblib`). El código para cargar el modelo y los datos está en `src/basket_model/`. Además, la API deberá loggear en un `.txt` métricas del servicio. El código deberá tener los tests correspondientes.

Componentes de la solución:

Crear una Pull Request (PR) en vuestro repositorio personal en la que resolvais la tarea propuesta. En la descripción de la PR deberás añadir una o varias imágenes con los resultados obtenidos así como una breve descripción con conclusiones o comentarios añadidos.

Los pasos para completar la tarea serán los siguientes:

1. Planificar la API en Excalidraw
2. Descargar los `.parquet` y colocarlos en una carpeta llamada `data/`.
3. Descargar el modelo y colocarlo en una carpeta llamada `bin/`.
4. Utilizar la librería FastAPI para crear una API con los siguientes endpoints:
 - a. GET `/status`: responde con un código 200.
 - b. POST `/predict`: lee el `USER_ID` de la request, busca las features en el histórico, y se las pasa al modelo para realizar la predicción del precio de la cesta. Responde con el precio.
5. Añadir a la API la funcionalidad de escribir métricas en un `.txt`. Las métricas deberán ser tanto del servicio (latencia, número de errores, etc), como del modelo (predicciones, etc).
6. Recordad buenas prácticas: gestión de errores, tests. La gestión de errores es especialmente crítica en un servicio. La principal propiedad de un servicio es la robustez.

Initial code

Crea un script llamado `app.py`. Añade este código para tener una API mínima con FastAPI:

```
import uvicorn
from fastapi import FastAPI

# Create an instance of FastAPI
app = FastAPI()

# Define a route for the root URL ("/")
@app.get("/")
def read_root():
    return {"message": "Hello, World!"}

# This block allows you to run the application using Uvicorn
if __name__ == "__main__":
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

En línea de comandos, podemos levantar la API con la siguiente línea:

```
poetry run python src/module_6/app.py
```

Finalmente, podemos hacer una petición a la API:

```
curl http://localhost:8000
```