MLOPS assignment

Diabetes application

Dominic Ho 3MCT
13-12-2022

Inhoud

Chosen dataset

https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset

AI Model

I will keep it simple and use a Logistic regression classifier, with hyperparameter tuning I will have around 77 – 82 accuracy.

About

My fictional company is making an application that speeds up diabetes diagnoses.
A normal blood sugar/glucose test will take a few hours (taking the blood sample and comparing changes after different tests)

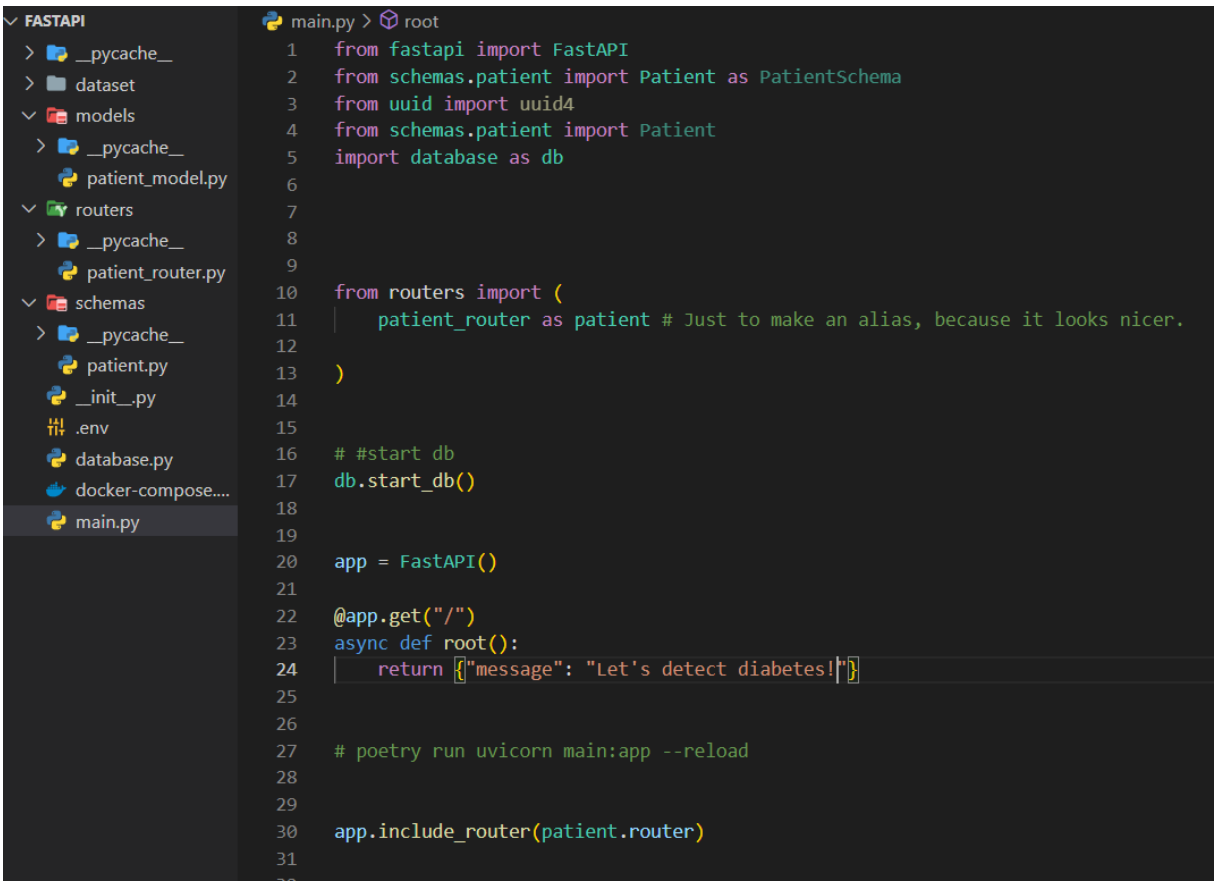
After every test, a blood sample must be sent to the lab which then can take a few days to get the test results back. We want to speed up diagnosis in the lab with our AI model.

(if possible we could also skip the lab part with a mobile application that nurses or doctors can use directly, but I don't know what happens with the samples in the lab. I assume the blood gets analyzed with (expensive) machines)

FastAPI

Environment

I made a new Fastapi **environment** with Poetry



Schema

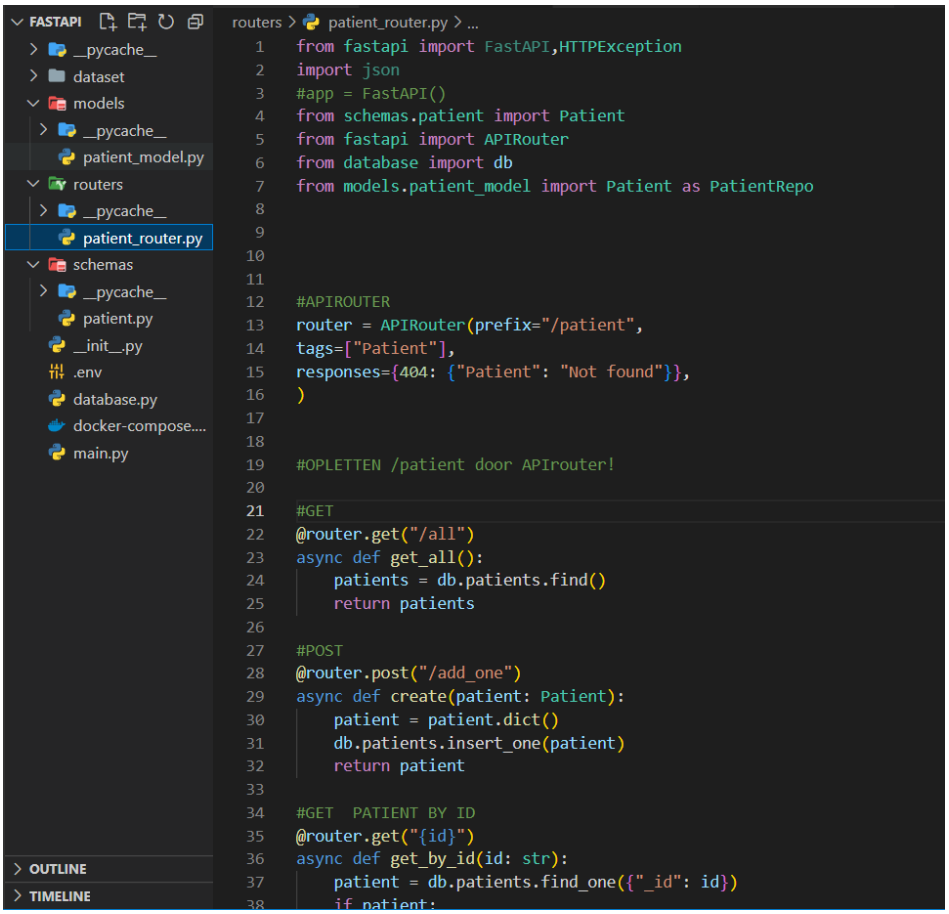
Added a schema **Patient**

schemas > patient.py > Patient > name

```python
# patient.py
from typing import Optional
from pydantic import BaseModel


class Patient(BaseModel):
    # uuid: Optional[str]
    id: str
    name: str
    pregnancies: int
    glucose: int
    bloodpressure: int
    skinthickness: int
    insulin: int
    bmi: float
    dpf: float
    age: int



    class Config:
        orm_mode = True


    def sayHello(self):
        print("Hello, I am " + self.name)


```

Routes

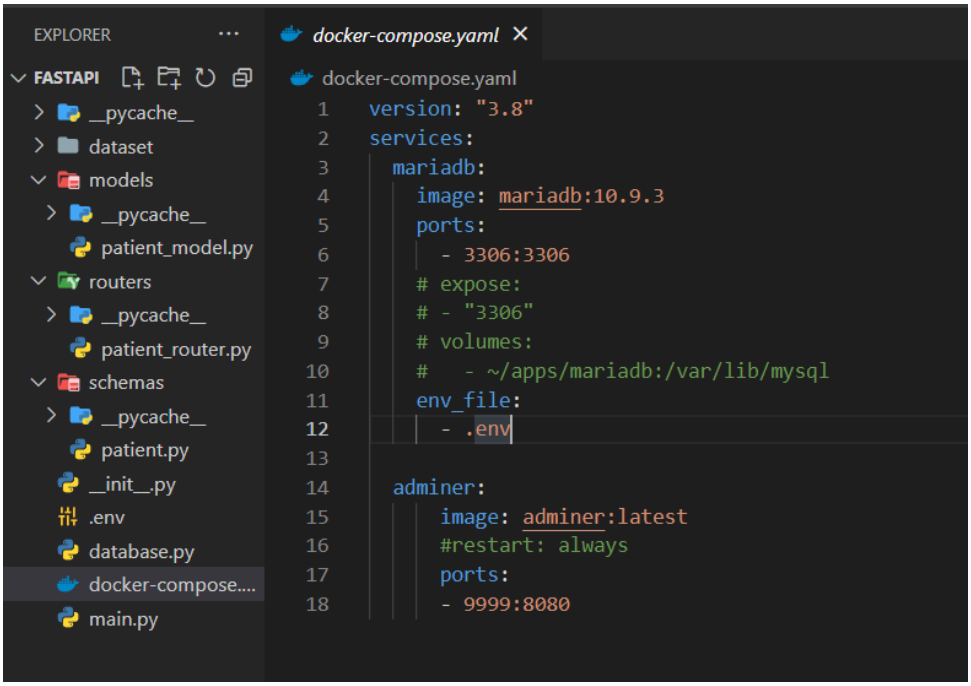
Added **routes** with **API rerouting**

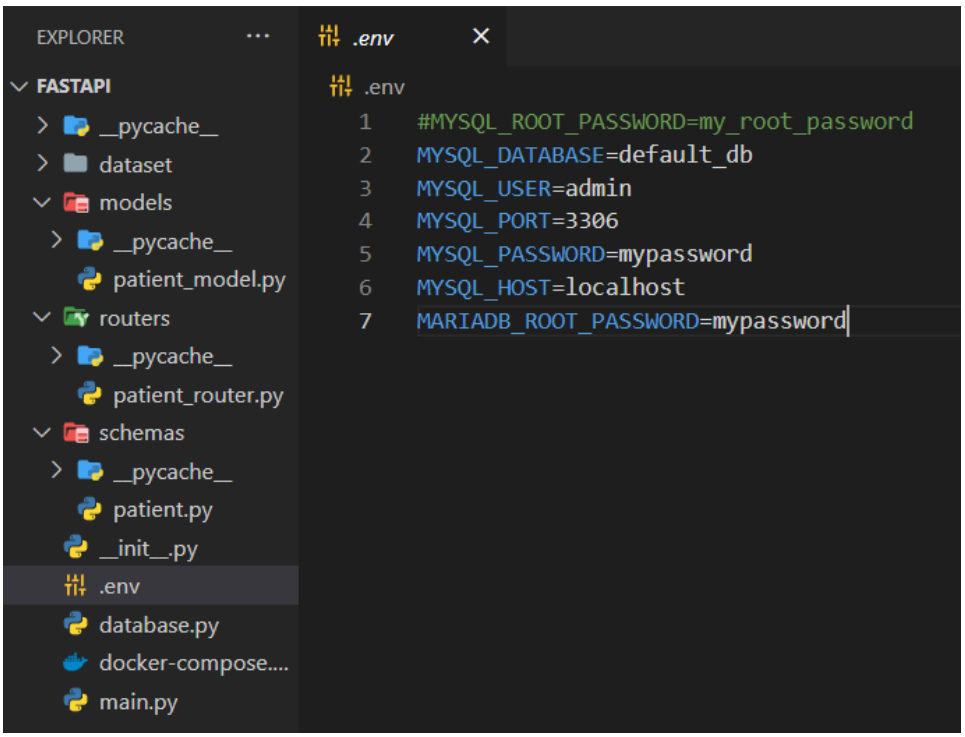


Database with Docker

Setting up **database** with a **docker-compose.yaml** file

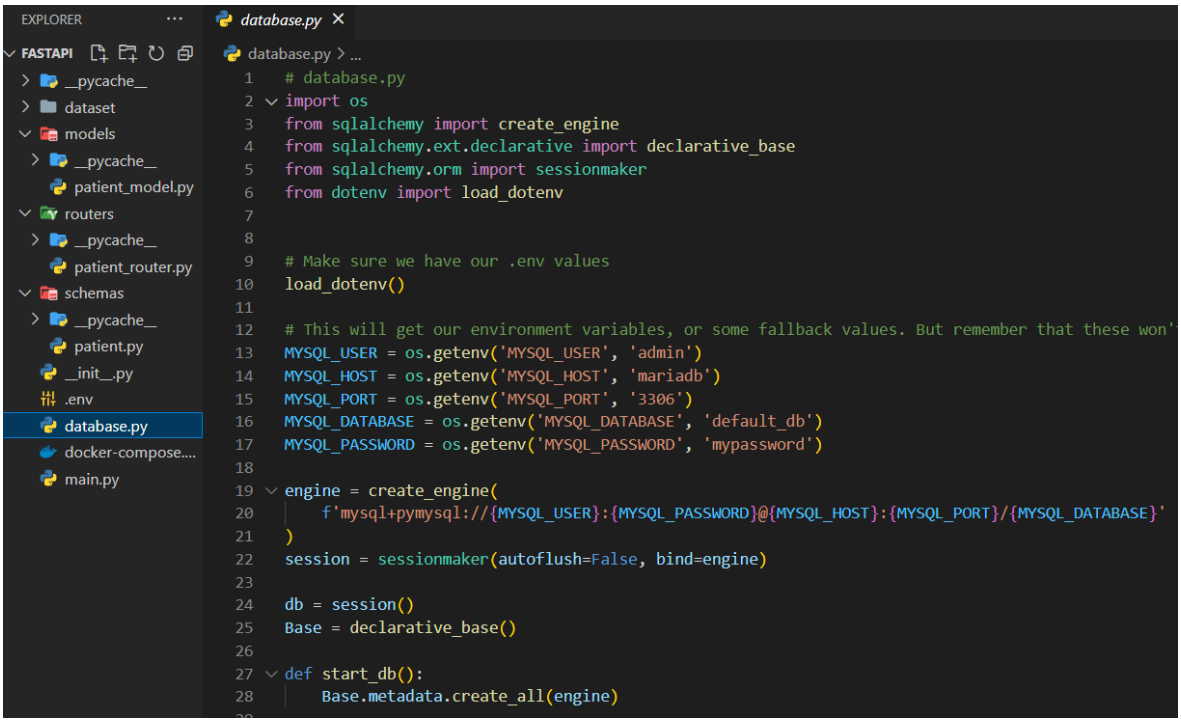
Login to database with **.env** file





Connect database

Connect to database with **database.py**



Database tables

Create database tables with **patient_model**

```python
class Patient(Base):
    __tablename__ = 'patients'

    uuid = Column(String(150),primary_key=True,default=generate_uuid , name="uuid")
    id = Column('id', String(20))
    name = Column('name', String(20))
    pregnancies = Column('pregnancies', Integer)
    glucose = Column('glucose', Integer)
    bloodpressure = Column('bloodpressure', Integer)
    skinthickness = Column('skinthickness', Integer)
    insulin = Column('insulin', Integer)
    bmi = Column('bmi', Integer)
    dpf = Column('dpf', Integer)
    age = Column('age', Integer)
    # outcome = Column('outcome', Integer)

    #init method

    def __init__(self, *,uuid:str = generate_uuid(),id:str = "",name:str = "", pregn

        print("uuid:", uuid)
        print(len(uuid))

        self.uuid = uuid
        self.id = id
        self.name = name
        self.pregnancies = pregnancies
        self.glucose = glucose
        self.bloodpressure = bloodpressure
        self.skinthickness = skinthickness
        self.insulin = insulin
        self.bmi = bmi
        self.dpf = dpf
        self.age = age
```

Register Patient Schema as an ORM-ready object

```python
class Bird(BaseModel):
    # uuid: Optional[str]
    id: str
    name: str
    short: str
    image: str
    recon: list
    food: dict
    see: str



    class Config:
        orm_mode = True

    def sayHello(self):
        print("Hello, I am " + self.name)
```

Repository and queries

Make a repository and add queries from repository

GET http://127.0.0.1:8000/patient/get_all_from_repo Send 200 OK 3.99 ms 357 B

JSON Auth Query Headers 1 Docs Preview Headers 4 Cookies Timeline


```
1 ...
```

```
1 [
2   {
3     "uuid": "09ffd707-a8c7-4a89-967b-303235002461",
4     "pregnancies": 5,
5     "bloodpressure": 1,
6     "insulin": 1,
7     "dpf": 100,
8     "glucose": 1,
9     "id": "jan",
10    "name": "dominic",
11    "skinthickness": 1,
12    "bmi": 30,
13    "age": 30
14  },
15  {
16    "uuid": "184c4126-f75a-4b9f-9a0b-a1c807c32dae",
17    "pregnancies": 5,
18    "bloodpressure": 1,
19    "insulin": 1,
20    "dpf": 100,
21    "glucose": 1,
22    "id": "ben",
23    "name": "ben",
24    "skinthickness": 1,
25    "bmi": 30,
26    "age": 30
27  }
28 ]
```


```
<Patient gert>
<class 'models.patient_model.Patient'> has been added to the database!
INFO:     127.0.0.1:54485 - "POST /patient/add_from_repo HTTP/1.1" 200 OK
INFO:     127.0.0.1:54487 - "GET /patient/get_all_from_repo HTTP/1.1" 200 OK
```

Docker deployment

Docker deployment with **Dockerfile**, we also add a **service** to our docker-compose.yaml

Dockerfile > ...

```
FROM python:3.10-slim-bullseye as python-base

# https://python-poetry.org/docs#ci-recommendations
ENV POETRY_VERSION=1.2.0
ENV POETRY_HOME=/opt/poetry
ENV POETRY_VENV=/opt/poetry-venv

# Tell Poetry where to place its cache and virtual environment
ENV POETRY_CACHE_DIR=/opt/.cache

# Create stage for Poetry installation
FROM python-base as poetry-base

# Creating a virtual environment just for poetry and install it with pip
RUN python3 -m venv $POETRY_VENV \
    && $POETRY_VENV/bin/pip install -U pip setuptools \
    && $POETRY_VENV/bin/pip install poetry==${POETRY_VERSION}

# Create a new stage from the base python image
FROM python-base as example-app
```

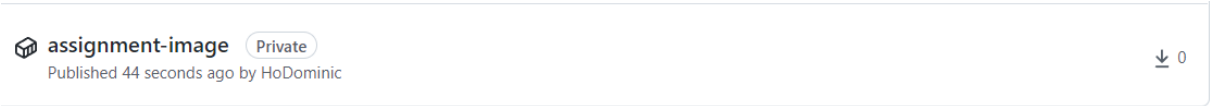
```
api:
    image: api:latest
    build:
      context: .
      dockerfile: Dockerfile
    env_file:
      - .env
    ports:
      - 5000:5000
    depends_on:
      - mariadb
```

Github Container Registry

Generate a token from Github and login with Github account

```
C:\Users\domin>set PAT=[redacted]

C:\Users\domin>echo %PAT%
[redacted]

C:\Users\domin>echo %PAT% | docker login ghcr.io --username HoDominic --password-stdin
Login Succeeded
```

Push image to Github



Kubernetes

To scale out our API accross multiple machines, and get them updated in a perfect manner, we use **Kubernetes**

Create deployment, add service, port-forward and scale it

```
PS C:\Users\domin\OneDrive\Bureaublad\MCT3\ML_OPS\Project_mlops\fastapi-env\fastapi> kubectl get deployment
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app              3/3     3            3           70d
nginx-app-assignment   3/3     3            3           87m
nginx-deployment       2/2     2            2           67d
```

```
PS C:\Users\domin\OneDrive\Bureaublad\MCT3\ML_OPS\Project_mlops\fastapi-env\fastapi> kubectl port-forward service/nginx-http-assignment 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

```
PS C:\Users\domin\OneDrive\Bureaublad\MCT3\ML_OPS\Project_mlops\fastapi-env\fastapi> kubectl get service
NAME                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes             ClusterIP   10.96.0.1       <none>        443/TCP   76d
my-service             ClusterIP   10.99.182.31    <none>        80/TCP    67d
nginx-http             ClusterIP   10.101.210.26   <none>        80/TCP    70d
nginx-http-assignment  ClusterIP   10.105.236.60   <none>        80/TCP    85m
vue-dockerservice      ClusterIP   10.101.120.89   <none>        80/TCP    11d
```

```
PS C:\Users\domin\OneDrive\Bureaublad\MCT3\ML_OPS\Project_mlops\fastapi-env\fastapi> kubectl scale deployment nginx-app-assignment --replicas=10
deployment.apps/nginx-app-assignment scaled
PS C:\Users\domin\OneDrive\Bureaublad\MCT3\ML_OPS\Project_mlops\fastapi-env\fastapi>
```

We need a **cluster-role-binding.yaml** and **service-account.yaml** and a **token** to access the **Kubernetes dashboard**

kubernetes > service-account.yaml

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
```

kubernetes > cluster-role-binding.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin
  namespace: kubernetes-dashboard
```

Secret

Create secret from Docker Registry and deploy API with Kubernetes

kubernetes > deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fastapi-deployment
  namespace: assignment-fastapi-gui
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx-app-assignment
  template:
    metadata:
      labels:
        app: nginx-app-assignment
    spec:
      containers:
      - name: nginx-app-assignment
        image: ghcr.io/hodominic/assignment-image
        ports:
        - containerPort: 80
      imagePullSecrets:
       - name: regcredassignment
```

Deployments

Name	Namespace	Images	Labels
fastapi-deployment	assignment-fastapi-gui	ghcr.io/hodominic/assignment-image	-
nginx-app-assignment	default	nginx	app: nginx-app-assignment