

1. Create docker network


- `docker network create cifar10-network`


2. MongoDB docker container


- `docker pull mongo`
- `docker run -d --name mongodb -p 27017:27017 --network cifar10-network mongo`

3. Backend Service

- Go to the
/UCD_ML_Application_Project/Model_train_microservice
directory available in the backend Dockerfile.
- `docker build -t cifar-backend-service .`
- `docker run -d --name cifar-backend-service -p 8000:8000 --network cifar10-network cifar-backend-service`

UCD_ML_Application_Project / Model_train_microservice / app / db.py 

 Lakshan-Jayaweera285 [add comments](#)

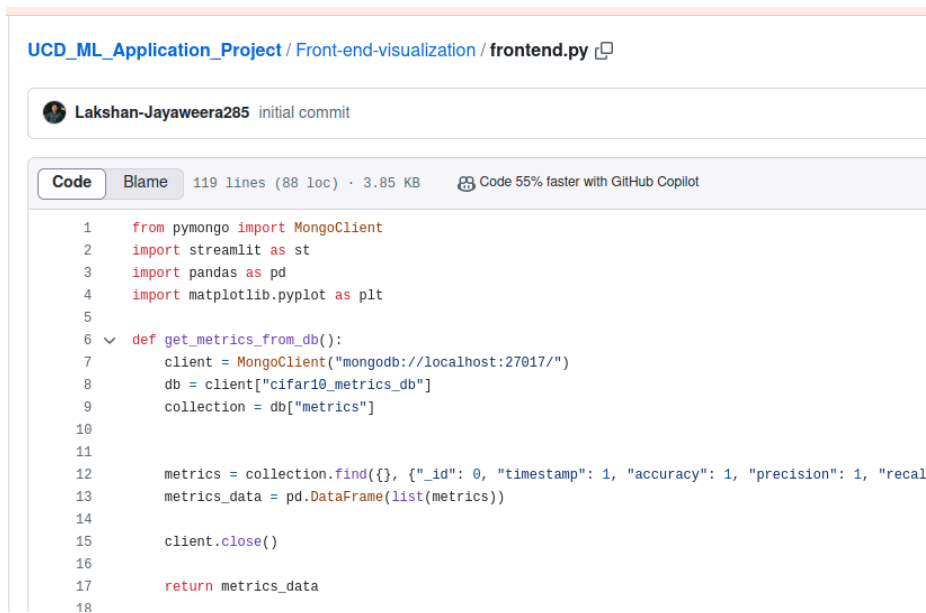
Code Blame 81 lines (72 loc) · 2.74 KB  Code 55% faster with GitHub Copilot

```
1  from pymongo import MongoClient
2  from datetime import datetime
3
4  DB_NAME = "cifar10_metrics_db"
5  COLLECTION_NAME = "metrics"
6  COLLECTION_TRAINING_DETAILS = "training_details"
7  client = MongoClient('mongodb://localhost:27017/')
8  db = client[DB_NAME]
9  collection = db[COLLECTION_NAME]
10 training_details_collection = db[COLLECTION_TRAINING_DETAILS]
11
12 def init_db():
13     pass
14
15 def add_metric(accuracy: float, f1_score: float, precision: float, recall: float):
16     metric = {
17         "timestamp": datetime.now().isoformat(),
18         "accuracy": accuracy.
```

- Change the source code in app/db.py mongo client to
“mongodb://mongodb:27017”

4. Frontend Service

- Go to the /UCD_ML_Application_Project/Front-end-visualization/ directory available in the frontend Dockerfile.
- `docker build -t cifar-frontend-service .`
- `docker run -d --name cifar-frontend-service -p 8501:8501 --network cifar10-network cifar-frontend-service`



The screenshot shows a GitHub repository for 'UCD_ML_Application_Project' with the file 'Front-end-visualization / frontend.py' selected. The commit is by 'Lakshan-Jayaweera285' and is an 'initial commit'. The file statistics show 119 lines (88 loc) and 3.85 KB. A badge indicates 'Code 55% faster with GitHub Copilot'. The code is a Python script that imports pymongo, streamlit, pandas, and matplotlib. It defines a function 'get_metrics_from_db()' which connects to a MongoDB instance at 'mongodb://localhost:27017/' and retrieves metrics from a collection named 'metrics'. The metrics are then converted into a pandas DataFrame and returned.

```
1 from pymongo import MongoClient
2 import streamlit as st
3 import pandas as pd
4 import matplotlib.pyplot as plt
5
6 def get_metrics_from_db():
7     client = MongoClient("mongodb://localhost:27017/")
8     db = client["cifar10_metrics_db"]
9     collection = db["metrics"]
10
11
12     metrics = collection.find({}, {"_id": 0, "timestamp": 1, "accuracy": 1, "precision": 1, "recall": 1})
13     metrics_data = pd.DataFrame(list(metrics))
14
15     client.close()
16
17     return metrics_data
18
```

- Change the source code in frontend.py mongo client to "mongodb://mongodb:27017"