Project Title

Final Year Project Backend

Description

Provide a brief description of your project, its purpose, and what it aims to achieve.

Installation

Step-by-step instructions on how to set up the project locally.

```
# Navigate to the project directory

cd backend
```

Create a Virtual Environment

Unix/Linux

```
python3 -m venv venv
source venv/bin/activate
```

Windows

```
python -m venv venv
venv\Scripts\activate
```

Install Dependencies

Install the required dependencies using the following command:

```
pip install -r requirements.txt
```

Project Structure

```
backend/
├── models/
│   └── xgb_classifier.joblib
```

```
├── app/
│   ├── __init__.py       # Initializes the FastAPI app and configures settings
│   ├── main.py           # Contains the API endpoint definitions
│   ├── models.py         # Defines the data models used in the application
│   └── schemas.py        # Pydantic schemas for request and response validation
├── tests/
│   ├── test_main.py      # Tests for the API endpoints
│   └── conftest.py       # Test configurations and fixtures
├── requirements.txt
└── README.md
```

Usage

Instructions and examples on how to use the project.

```
# Start the application
uvicorn app.main:app --reload
```

Testing

For testing purposes in Linux, you can use the following `curl` command:

```
curl -X POST "http://127.0.0.1:8000/predict" \
    -H "Content-Type: application/json" \
    -d '{
        "features": [
            8699, 1.234235046, 3.019740421, -4.304596885, 4.7345489513,
3.624200831,
            -1.357745663, 1.713444988, -0.496358487, -1.28285782, -2.447469255,
            2.101343865, -4.609628391, 1.4674525, -6.079337193, -0.339237373,
            7.581850954, 6.739384385, 9.042493178, -2.721853122, 0.009060836,
            -0.379068307, -0.704181032, -0.656804756, -1.632652957, 1.488901448,
            0.566797273, -0.010016223, 0.146792735, 10000
        ]
    }'
```

This results in:

```
{
    "prediction": 1,
    "probability": 0.7587947249412537
}
```

Model

The project uses an XGBoost classifier trained on the fraud-transcations.txt dataset.

- **Training Script:** Located in `app/models.py`.
- **Model File:** `models/xgb_classifier.joblib`

API Endpoints

POST /predict

- **Description:** Predicts whether a transaction is fraudulent.
- **Request Body:** Transaction details adhering to the `schemas.Transaction` schema.
- **Response:** Prediction result.

Dependencies

All dependencies are listed in `requirements.txt`. Key libraries include:

- **FastAPI:** Web framework for building APIs.
- **Uvicorn:** ASGI server for running FastAPI applications.
- **XGBoost:** Machine learning library for the classifier.
- **Pydantic:** Data validation library used with FastAPI.