

Linear Regression Model Integrated with Website using FastAPI

Name: Barsha Baibhabhi

Roll No: 22053416

1. Overview

This project implements a Linear Regression model using scikit-learn, integrates it with a FastAPI backend, and provides a simple frontend for user interaction. The dataset used is the Diabetes dataset from `sklearn.datasets`. The model is trained to predict diabetes progression based on ten numerical input features.

Files in the Project

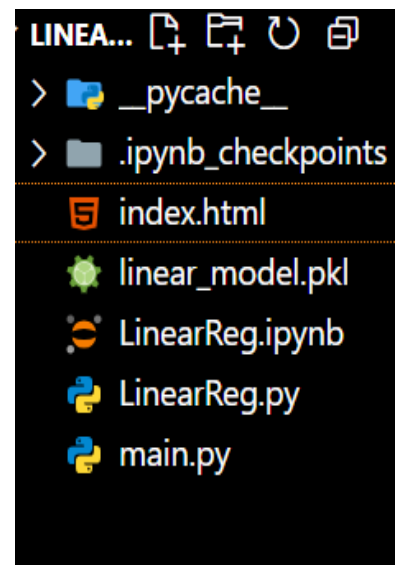
- **LinearReg.ipynb**- Jupyter Notebook for training the model and saving it as a pickle file.

- **LinearReg.py** - Python script version of the Jupyter Notebook.

PS

```
C:\Users\KIIT\Documents\AD22053416\LinearRegression3416>  
jupyter nbconvert --to python LinearReg.ipynb
```

- **linear_model.pkl** - Saved model file.
- **main.py** - FastAPI backend to load the model and serve predictions.
- **index.html** - Frontend with inline CSS & JavaScript for user interaction.



2. Installation & Setup

Prerequisites

Ensure you have Python installed. Install required dependencies:

```
pip install fastapi uvicorn scikit-learn numpy pandas jupyter
```

Starting FastAPI Server

Run the FastAPI server with:

```
uvicorn main:app --host 127.0.0.1 --port 8010 --reload
```

3. Training the Model & Generating Pickle File

RunLinearReg.ipynb to:

1. Load the Diabetes dataset.
2. Train a Linear Regression model.
3. Save the trained model as `linear_model.pkl`.

4. FastAPI Backend (main.py)

The FastAPI backend loads the saved model and provides an endpoint for predictions.

Code Snippet

```
with open("linear_model.pkl", "rb") as f:  
    model = pickle.load(f)
```

```
app = FastAPI()
```

```
class InputData(BaseModel):  
    features: list[float]
```

```
@app.post("/predict/")  
def predict(data: InputData):  
    X_new = np.array(data.features).reshape(1, -1)  
    prediction = model.predict(X_new)  
    return {"prediction": prediction.tolist()}
```

5. Frontend (index.html) with Inline CSS & JavaScript

The frontend allows users to enter feature values and fetch predictions from the FastAPI backend.

Code Snippet

```
async function predict() {  
  
    let inputValues = document.getElementById("features").value;  
  
    let featureArray = inputValues.split(",").map(Number);  
  
    let response = await fetch("http://127.0.0.1:8010/predict/", {  
  
        method: "POST",  
  
        headers: { "Content-Type": "application/json" },  
  
        body: JSON.stringify({ features: featureArray })  
  
    });  
  
    let data = await response.json();  
  
    document.getElementById("result").innerText = "Prediction: " +  
data.prediction; }
```

6. Testing the Integration

1. Run the FastAPI server: `uvicorn main:app --host 127.0.0.1 --port 8010 --reload`

2. Open index.html in a browser.

Linear Regression Predictor for Diabetes

Enter feature values:

Prediction: 113.28831970766086

3. Enter 10 feature values (comma-separated) and click 'Predict'.
 4. The predicted value will be displayed.
-

7. Conclusion

This project demonstrates a simple integration of a Linear Regression model with a FastAPI backend and a basic HTML frontend. The model is trained on the Diabetes dataset and can predict diabetes progression based on input features.
