Linear Regression Model for Diabetes Progression Prediction

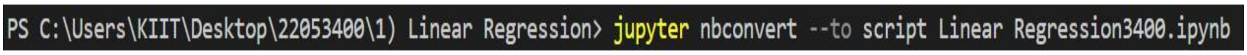
NAME- Ankita Satapathy
ROLL NO- 22053400

1. Introduction

This project implements a simple linear regression model to predict diabetes progression based on BMI values. The model is built using Python and the scikit-learn library, with a custom linear regression class. The application provides an API using FastAPI and a front-end interface for user interaction.

2. Files in the Submission

i. LinearRegression3400.ipynb: Implements a simple linear regression model using gradient descent in Jupyter Notebook and also contains the model training process.
ii. LinearRegression3400.py: Generated in vscode code terminal using command to facilitate backend integration.



```
PS C:\Users\KIIT\Desktop\22053400\1) Linear Regression> jupyter nbconvert --to script Linear Regression3400.ipynb
```

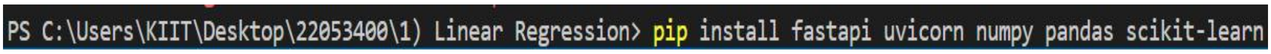
iii. main.py: Defines the FastAPI server to handle prediction requests.
iv. model.pkl: The trained model serialized using pickle.
v. index.html: Front-end UI for user input and displaying predictions.

3. Installation and Setup

i. Prerequisites :- Required Python packages: Numpy, scikit-learn, fastapi, pickle, uvicorn.

- NumPy (*numpy)* is used for numerical computations and matrix operations in model training.
- Scikit-learn (*sklearn*) is used for loading the Diabetes dataset and provides ML utilities.
- FastAPI (*fastapi*) is used to create a web API for serving predictions.
- Pickle (*pickle*) is used to save and load the trained model for reuse.
- Uvicorn (*uvicorn*) is used to run the FastAPI server asynchronously.

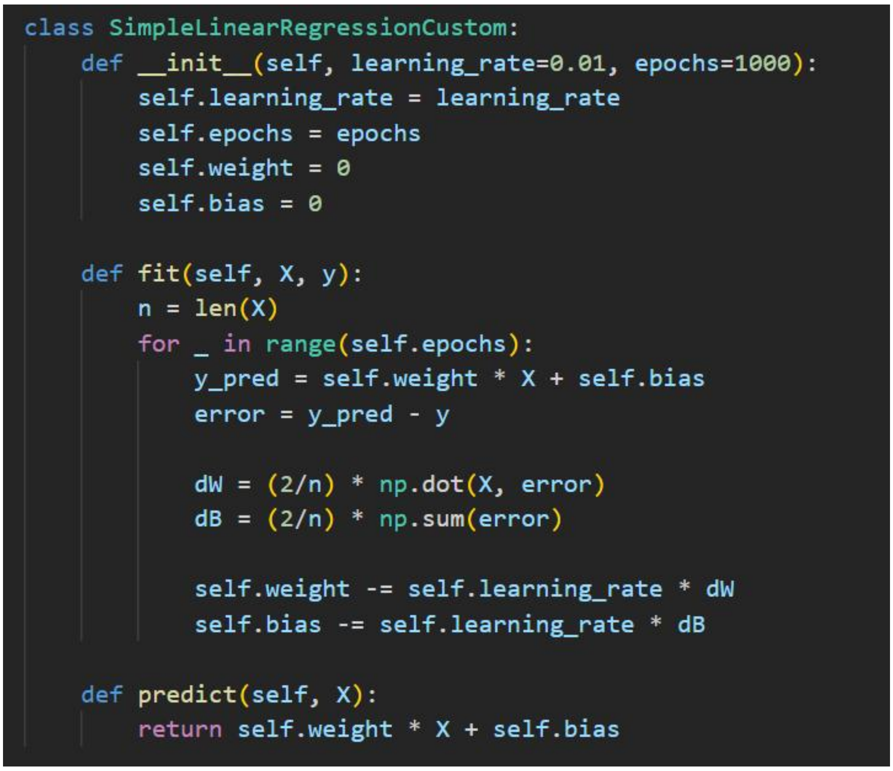
ii. Installing Dependencies:- The following command is used in vscode terminal to install dependencies



```
PS C:\Users\KIIT\Desktop\22053400\1) Linear Regression> pip install fastapi uvicorn numpy pandas scikit-learn
```

4. Model Implementation

The SimpleLinearRegressionCustom class in LinearRegression3400.py implements a basic linear regression model using gradient descent.



```python
class SimpleLinearRegressionCustom:
    def __init__(self, learning_rate=0.01, epochs=1000):
        self.learning_rate = learning_rate
        self.epochs = epochs
        self.weight = 0
        self.bias = 0

    def fit(self, X, y):
        n = len(X)
        for _ in range(self.epochs):
            y_pred = self.weight * X + self.bias
            error = y_pred - y

            dW = (2/n) * np.dot(X, error)
            dB = (2/n) * np.sum(error)

            self.weight -= self.learning_rate * dW
            self.bias -= self.learning_rate * dB

    def predict(self, X):
        return self.weight * X + self.bias
```

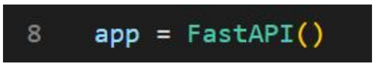
5. Training the Model

- The dataset used is the Diabetes Dataset from sklearn.datasets.
- The model is trained on the BMI feature (X = diabetes.data[:, 2]) to predict the target (y = diabetes.target).
- Training is performed using gradient descent over 1000 epochs.
- After training, the model is saved using pickle:

6. API Implementation

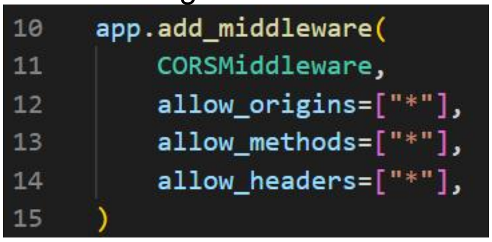
The FastAPI-based backend (main.py) loads the trained model and provides an endpoint to make predictions:

i. API Setup



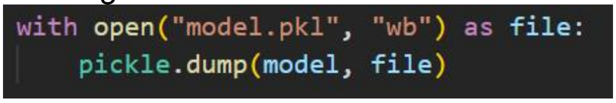
```python
app = FastAPI()
```

ii. CORS Configuration- To allow front-end requests:



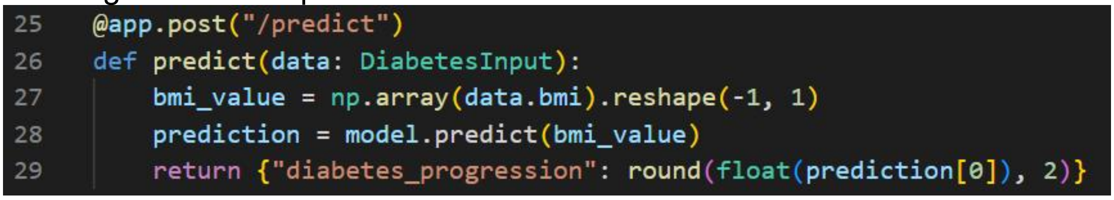
```python
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_methods=["*"],
    allow_headers=["*"],
)
```

iii. Loading the Trained Model



```python
with open("model.pkl", "wb") as file:
    pickle.dump(model, file)
```

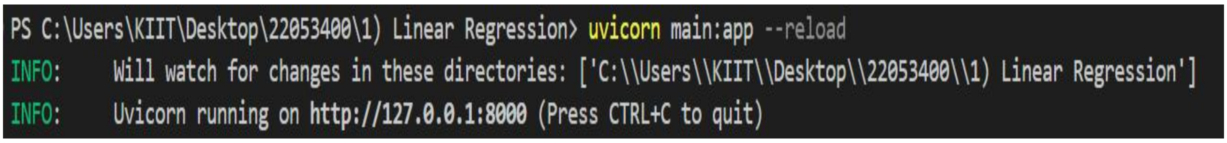
iv. Defining the API Endpoint



```python
@app.post("/predict")
def predict(data: DiabetesInput):
    bmi_value = np.array(data.bmi).reshape(-1, 1)
    prediction = model.predict(bmi_value)
    return {"diabetes_progression": round(float(prediction[0]), 2)}
```

7. Running the Application (FastAPI Server)

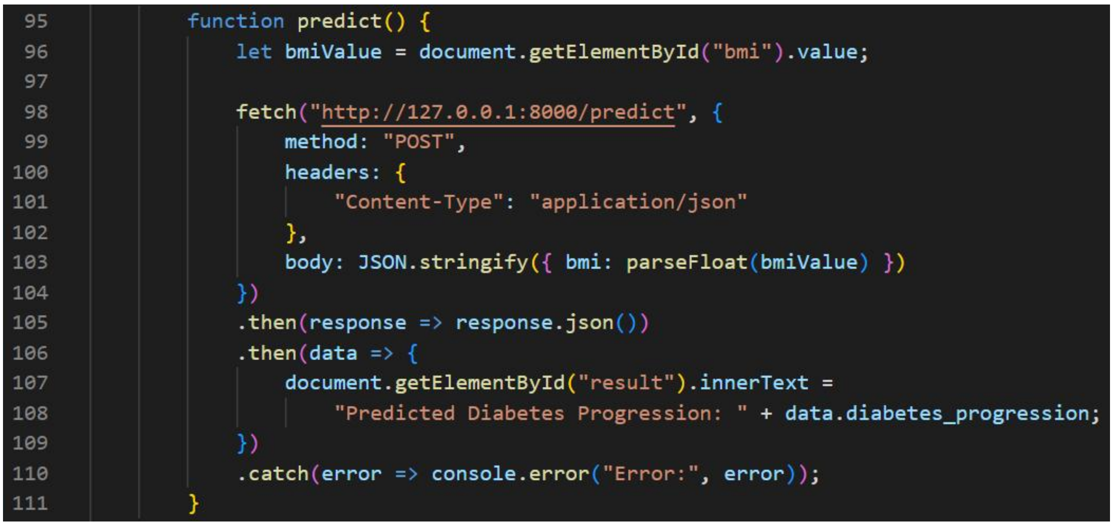
The following command is used in vscode to start the API server:



```
PS C:\Users\KIIT\Desktop\22053400\1) Linear Regression> uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\KIIT\\Desktop\\22053400\\1) Linear Regression']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```

8. Front-End Implementation

The index.html file provides a simple UI for users to input BMI values and get predictions using the API.It has inline CSS and JavaScript. The JavaScript function sends a request to the FastAPI backend:



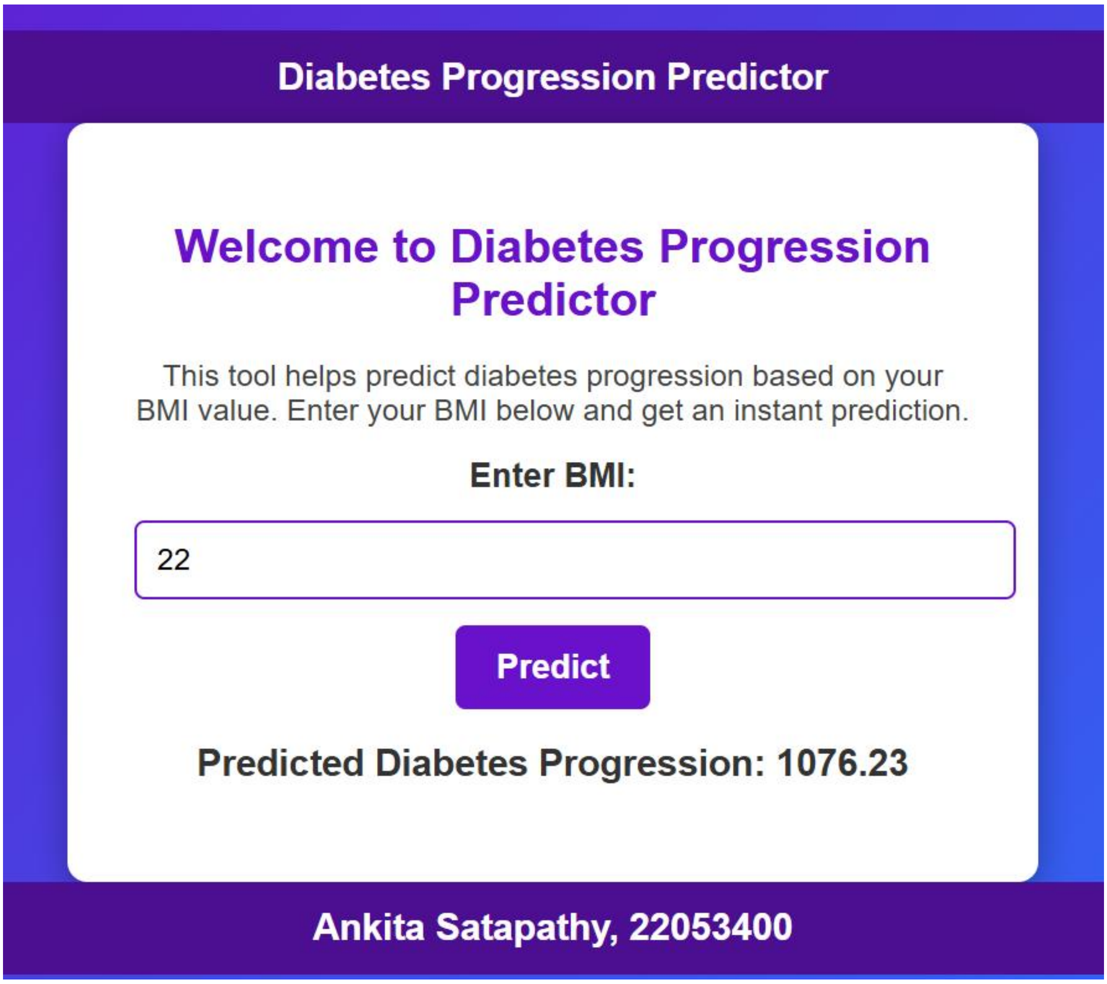
```javascript
        function predict() {
            let bmiValue = document.getElementById("bmi").value;

            fetch("http://127.0.0.1:8000/predict", {
                method: "POST",
                headers: {
                    "Content-Type": "application/json"
                },
                body: JSON.stringify({ bmi: parseFloat(bmiValue) })
            })
            .then(response => response.json())
            .then(data => {
                document.getElementById("result").innerText =
                    "Predicted Diabetes Progression: " + data.diabetes_progression;
            })
            .catch(error => console.error("Error:", error));
        }
```

9. Conclusion

This project successfully implements a simple linear regression model to predict diabetes progression based on BMI. The model is integrated with a FastAPI backend. It utilizes NumPy for numerical computations and Scikit-learn for dataset handling. The trained model is saved using Pickle and deployed via Uvicorn for real-time predictions.

The frontend layout features a centered card with a white background, rounded corners, and a shadow effect. At the top, a bold purple header displays the title "Diabetes Progression Predictor," followed by a welcoming message and a brief description of the tool. Below, a user input field for BMI is provided, along with a "Predict" button, which displays the predicted diabetes progression value underneath upon submission.



Diabetes Progression Predictor

Welcome to Diabetes Progression Predictor

This tool helps predict diabetes progression based on your BMI value. Enter your BMI below and get an instant prediction.

Enter BMI:

22

Predicted Diabetes Progression: 1076.23

Ankita Satapathy, 22053400