GENEHARMONY

File Structure

- **`app.py`:** Contains the Flask application with endpoints for prediction and recommendations.
- **`training.py`:** Script for training the XGBoost model and preparing the dataset.
- **`nutrigenomics_dataset.csv`:** The dataset used for training the model.

Installation

1. Clone this repository:
2. Install required packages:

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

3. Train the model (if not already trained):

```
python training.py
```

4. Run the application:

```
python app.py
```

The application will run on `http://127.0.0.1:5000`.

Endpoints

1. Predict Nutrient

Endpoint: `/predict`

Method: POST

Description: Predicts the nutrient based on user input.

Request Format:

```
{
  "Age": 30,
  "Gender": "Male",
  "Weight (kg)": 70,
  "Height (cm)": 175,
  "Ethnicity": "Asian",
  "Health_Conditions": "None",
  "Lifestyle_Factors": "Moderately Active, Non-Smoker",
  "Dietary_Preferences": "Vegan",
  "SNP_ID": "rs12345",
  "Genotype": "AA"
}
```

Response Format:

```
{
  "prediction":"Iron",
  "status":"success"
}
```

Example Command in Postman:

1. Select POST method.

2. Set the URL to http://127.0.0.1:5000/predict.
3. Add the JSON payload in the Body section (raw format).

2. Get Recommendations

Endpoint: /recommend

Method: POST

Description: Provides dietary recommendations based on the predicted nutrient.

Request Format:

```
{
  "Age": 30,
  "Gender": "Male",
  "Weight (kg)": 70,
  "Height (cm)": 175,
  "Ethnicity": "Asian",
  "Health_Conditions": "None",
  "Lifestyle_Factors": "Moderately Active, Non-Smoker",
  "Dietary_Preferences": "Vegan",
  "SNP_ID": "rs12345",
  "Genotype": "AA"
}
```

Response Format:

```
{
 "prediction":"Iron",
 "recommendation":"Consume lean meats, beans, and spinach for iron-
                  rich nutrition.",
 "status":"success"

}
```

Example Command in Postman:

1. Select POST method.
2. Set the URL to `http://127.0.0.1:5000/recommend`.
3. Add the JSON payload in the Body section (raw format).

Example Input Data

Use the structure of the training dataset (`nutrigenomics_dataset.csv`) as a guide for formatting the input JSON. Make sure to include all relevant features.

Dependencies

- Flask
- Flask-CORS
- XGBoost
- Scikit-learn
- Pandas
- Numpy
- Joblib

Notes

- **Feature Columns:** The input data should match the structure of the dataset used for training. Missing columns will be filled with `0` during prediction.
- **Dataset:** Ensure `nutrigenomics_dataset.csv` is in the same directory or update the path in `training.py`.