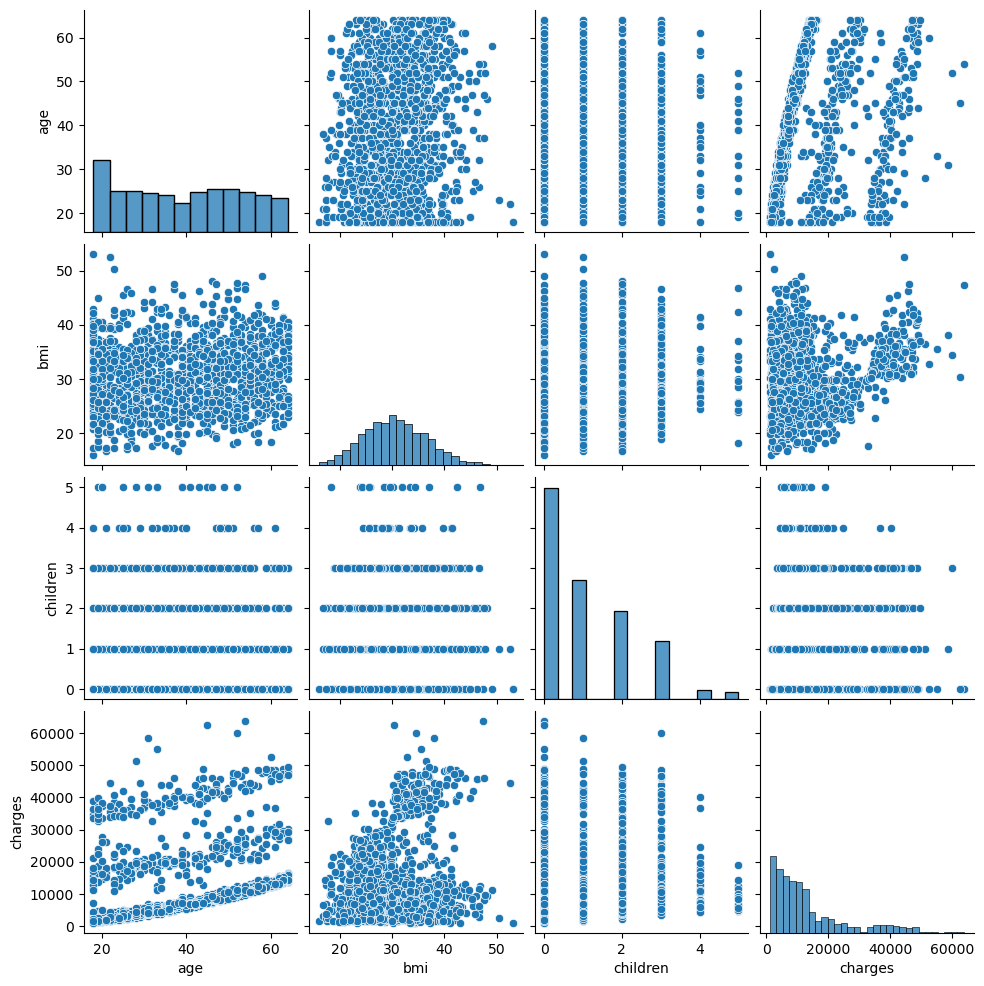
# Import necessary libraries  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error  
  
# Load the dataset  
url = '/Users/kiran/Desktop/insurance.csv'  
data = pd.read\_csv(url)  
  
# Preview the dataset  
data.head()

age sex bmi children smoker region charges  
0 19 female 27.900 0 yes southwest 16884.92400  
1 18 male 33.770 1 no southeast 1725.55230  
2 28 male 33.000 3 no southeast 4449.46200  
3 33 male 22.705 0 no northwest 21984.47061  
4 32 male 28.880 0 no northwest 3866.85520

# Check for missing values  
data.isnull().sum()  
  
# Data summary  
data.describe()  
  
# Visualize the relationships  
sns.pairplot(data)

<seaborn.axisgrid.PairGrid at 0x14b567800>



# Convert categorical variables using one-hot encoding  
data = pd.get\_dummies(data, drop\_first=True)  
  
# Define features (X) and target (y)  
X = data.drop('charges', axis=1)  
y = data['charges']  
  
# Train-test split (80% training, 20% testing)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize and train the model  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
  
# Predict on the test data  
y\_pred = model.predict(X\_test)

# Calculate Mean Squared Error  
mse = mean\_squared\_error(y\_test, y\_pred)  
print(f'Mean Squared Error: {mse}')  
  
# Plot the results  
plt.scatter(y\_test, y\_pred)  
plt.xlabel("True Values")  
plt.ylabel("Predictions")  
plt.title("True Values vs Predictions")  
plt.show()

Mean Squared Error: 33596915.85136147

