```
# Python Code for GHG Emission Monitoring and Reduction in Agriculture


# Dataset: 'df_agriculture.csv'

import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.inspection import plot_partial_dependence
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import GradientBoostingRegressor
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv('df_agriculture.csv')

# Data Preprocessing
df['Date'] = pd.to_datetime(df['Date'])
df = df.set_index('Date')

# Define independent and dependent variables
X = df[['Fertilizer_Usage', 'Livestock_Emissions', 'Tillage_Method',
'Energy_Usage']]
y = df['GHG_Emissions']

# Split dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Question 1: GHG Emission Prediction using Random Forest
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Predictions and evaluation
y_pred = rf_model.predict(X_test)
print(f"R² Score: {r2_score(y_test, y_pred)}")
print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred)}")

# Plot Partial Dependence (feature importance visualization)
plot_partial_dependence(rf_model, X_test, ['Fertilizer_Usage',
'Livestock_Emissions', 'Energy_Usage'])
plt.show()

# Question 2: Identifying Key Emission Sources using SHAP Analysis
# Placeholder: SHAP analysis requires additional library (shap) - not implemented
here

# Question 3: Scenario Analysis for Emission Reduction using Gradient Boosting
gb_model = GradientBoostingRegressor(n_estimators=100, random_state=42)
gb_model.fit(X_train, y_train)

# Simulating impact of precision fertilization (reduce Fertilizer_Usage by 15%)
X_test_reduced_fertilizer = X_test.copy()
X_test_reduced_fertilizer['Fertilizer_Usage'] *= 0.85
```

```python
# Predicting emissions with reduced fertilizer usage
y_pred_reduced_fertilizer = gb_model.predict(X_test_reduced_fertilizer)
print(f"Emissions with Precision Fertilization: {y_pred_reduced_fertilizer.mean()} kg CO₂/ha")

# Question 4: Linear Regression for Energy Usage Optimization
lr_model = LinearRegression()
lr_model.fit(X_train[['Energy_Usage']], y_train)

# Predicting emissions based on energy usage
y_pred_energy_optimized = lr_model.predict(X_test[['Energy_Usage']])
print(f"Energy Usage Impact on GHG Emissions: {y_pred_energy_optimized.mean()} kg CO₂/ha")

# Visualization: Actual vs Predicted Emissions
plt.scatter(y_test, y_pred, color='blue', label='Random Forest')
plt.scatter(y_test, y_pred_reduced_fertilizer, color='green', label='Precision Fertilization')
plt.xlabel("Actual GHG Emissions")
plt.ylabel("Predicted GHG Emissions")
plt.legend()
plt.title("Actual vs Predicted GHG Emissions with Mitigation Strategies")
plt.show()
```

Explanation of the Python Code:

Random Forest for GHG Emission Prediction:
A RandomForestRegressor is used to predict GHG emissions based on fertilizer usage, livestock emissions, tillage method, and energy usage. The model is evaluated using R^2 and MSE.

Partial Dependence Plot:
A partial dependence plot visualizes the impact of features (fertilizer usage, livestock emissions, energy usage) on GHG emissions.

Gradient Boosting for Scenario Analysis:
A GradientBoostingRegressor is used to simulate the impact of precision fertilization by reducing fertilizer usage by 15%. The resulting emissions are predicted and compared to the baseline.

Linear Regression for Energy Usage Optimization:
A LinearRegression model predicts the impact of energy usage on GHG emissions. This can be used to simulate the effect of energy-saving measures on emissions.

This Python code provides a comprehensive framework for measuring, monitoring, and reducing GHG emissions in agriculture using data science techniques.