Regression vs Classification:

Regression:

Objective: Predict a continuous value.

Example: Predicting house prices, stock prices, temperature, etc.

Output: A range of values.

Classification:

Objective: Assign an input to a category or class.

Example: Spam detection, image recognition (cats vs. dogs), sentiment analysis.

Output: Discrete classes or labels.

###Code:-

```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Load dataset (you should replace this with your actual dataset)
# For simplicity, let's assume you have a CSV file named 'house_prices.csv'
# with columns like 'sqft', 'num_bedrooms', 'price', etc.
data = pd.read_csv('house_prices.csv')

# Assume 'sqft' and 'num_bedrooms' as features for prediction
features = data[['sqft', 'num_bedrooms']]
target = data['price']

# Split the dataset into training and testing sets
```

```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Create a linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, predictions)
print(f'Mean Squared Error: {mse}')

# Visualize the results (you can modify this based on your data)
plt.scatter(X_test['sqft'], y_test, color='black', label='Actual Prices')
plt.scatter(X_test['sqft'], predictions, color='blue', label='Predicted Prices')
plt.xlabel('Square Footage')
```

```
plt.ylabel('Price')
plt.legend()
plt.show()
```