Usage Guide

First Program (Training and Saving Models)

Purpose: This script simulates rocket launches, generates synthetic data, and trains machine learning models to predict launch success and maximum altitude.

1. **Setup**:
    - Ensure you have the following libraries installed:
        - `numpy`: For numerical operations.
        - `pandas`: For data handling and manipulation.
        - `matplotlib`: For visualizing results.
        - `scikit-learn`: For machine learning models and data preprocessing.
        - `joblib`: For saving and loading models.

    You can install them using pip if needed:
    ```
    pip install numpy pandas matplotlib scikit-learn joblib
    ```

2. **Input Parameters**:
    - When you run the script, it will ask you to input two values:
        - **Number of Test Samples**: This determines how many synthetic rocket launches will be simulated. Enter a valid integer (e.g., 1000).
        - **Test Size**: This is the proportion of the dataset to be used as the test set for model evaluation. Enter it as a decimal (e.g., 0.8 for 80% training and 20% testing).
        - **Constants:** Change the constants to fit the rocket size you want to train for

          Example:
          ```
          G = 9.81  # Gravitational acceleration (m/s^2), Earth's gravity

          RHO = 1.225  # Air density at sea level in kg/m^3

          A = 0.1  # Cross-sectional area of the rocket (m^2)

          DT = 0.1  # Time step for simulation (seconds)

          T_MAX = 100  # Maximum simulation time (seconds)

          BURN_TIME = 50  # Duration of the rocket engine burn time (seconds)
          ```

3. **Simulation**:
    - The script generates synthetic data by simulating rocket flights with random parameters:
        - Mass: Randomly selected between 500 kg and 1500 kg.
        - Thrust: Randomly selected between 10,000 N and 20,000 N.
        - Wind Speed: Randomly selected between 0 m/s and 30 m/s.
        - Temperature: Randomly selected between 270 K and 310 K.
        - Drag Coefficient: Randomly selected between 0.5 and 1.0.
    - For each combination of these parameters, the script runs a simulation to determine if the rocket launch is successful (achieves an altitude > 10 km) and records the maximum altitude reached (apogee).
4. **Training Models**:
    - The script trains two machine learning models:
        - **Random Forest Classifier**: Predicts the success or failure of a launch (1 for success, 0 for failure).
        - **Random Forest Regressor**: Predicts the maximum altitude (apogee) achieved by the rocket.
5. These models are trained on the synthetic dataset generated in the previous step.
6. **Saving Models**:
    - After training, the script saves the models and the feature scaler using `joblib`. The models and scaler are saved as:
        - `random_forest_classifier.pkl`: The trained classification model.
        - `random_forest_regressor.pkl`: The trained regression model.
        - `scaler.pkl`: The scaler used to standardize input features.
    - These saved models and scaler can be loaded in another script for prediction purposes.
7. **Visualization**:
    - The script generates two plots:
        - **Predicted vs Actual Apogee**: A scatter plot to compare the predicted and actual maximum altitudes (apogee).
        - **Launch Success Distribution**: A bar chart showing the number of successful vs failed launches.

Second Program (Prediction for New Rocket)

Purpose: This script uses the trained models to predict the launch success and maximum altitude (apogee) of a new rocket based on its input parameters.

1. **Setup**:
    - Ensure you have the following libraries installed:
        - `joblib`: To load the saved models and scaler.
        - `pandas`: To handle input data.

    You can install them using pip:
    ```
    pip install joblib pandas
    ```

2. **Loading Models and Scaler**:
    - The script loads the saved models and scaler from the previous script:
        - **Scaler** (`scaler.pkl`): This is used to scale the input features (rocket parameters) to match the training data.
        - **Random Forest Classifier** (`random_forest_classifier.pkl`): Used to predict whether the rocket launch will be successful.
        - **Random Forest Regressor** (`random_forest_regressor.pkl`): Used to predict the maximum altitude (apogee) the rocket will reach.
3. **Input Data**:
    - The script expects input parameters for a new rocket launch. These parameters should be provided in the form of a dictionary with the following keys:
        - `'Mass'`: The rocket's mass in kilograms (e.g., 1200).
        - `'Thrust'`: The thrust of the rocket in Newtons (e.g., 18000).
        - `'Wind Speed'`: The wind speed in meters per second (e.g., 15).
        - `'Temperature'`: The ambient temperature in Kelvin (e.g., 290).
        - `'Drag Coefficient'`: The aerodynamic drag coefficient (e.g., 0.7).

      Example:

```
new_rocket_data = {

'Mass': [1200],  # Mass in kg

'Thrust': [18000],  # Thrust in N

'Wind Speed': [15],  # Wind Speed in m/s

'Temperature': [290],  # Temperature in K

'Drag Coefficient': [0.7]  # Drag Coefficient

}
```

Scaling Input Data:

- The input data is converted into a pandas DataFrame and scaled using the loaded scaler (`scaler.transform(new_rocket)`), ensuring it matches the format of the training data.

4. **Making Predictions**:

- **Success Prediction**: The script uses the trained classifier to predict whether the rocket launch will be successful (1 for success, 0 for failure).
- **Apogee Prediction**: The script uses the trained regressor to predict the rocket's maximum altitude (apogee) in meters.

5. **Output**:
    - The script outputs:
        - A message indicating whether the rocket launch is predicted to be successful or not.
        - The predicted maximum altitude (apogee) in meters.

Example output:

```
New Rocket Launch Successful

Predicted Apogee: 9500.00 meters
```

Summary:

1. **First Script**: Generates synthetic rocket data, simulates launches, trains machine learning models, and saves them for future use.
2. **Second Script**: Loads the saved models and scaler, accepts new rocket parameters, scales the data, and predicts the success and maximum altitude of the launch.

By following this guide, you can simulate and predict rocket launches for various configurations using machine learning models.