**1. Data Preprocessing**

**1.1 Data Loading and Column Selection**

* **Dataset Loading:**  
  The dataset is read from a CSV file (globalterrorismdb\_0718dist.csv) using pandas.read\_csv with the 'latin-1' encoding to correctly interpret special characters.
* **Column Selection:**  
  For the classification task (predicting whether an attack is successful), only the following columns are retained:
  + **Temporal & Geographic:** iyear, country\_txt, region\_txt
  + **Attack Details:** attacktype1\_txt, targtype1\_txt, weaptype1\_txt
  + **Casualties:** nkill (number killed) and nwound (number wounded)
  + **Target Variable:** success (binary indicator: 1 for successful attacks, 0 for unsuccessful attacks)  
    For the regression task (predicting total casualties), a new target variable casualties is created as the sum of nkill and nwound.

**1.2 Handling Missing Values**

* **Numeric Columns:**  
  Missing values in nkill and nwound are filled with 0 to ensure that lack of data is treated as zero casualties.
* **Target Variable:**  
  Rows where the target variable (success for classification or casualties for regression) is missing are dropped.

**1.3 Feature Engineering and Encoding**

* **Categorical Encoding:**  
  Columns such as country\_txt, region\_txt, attacktype1\_txt, targtype1\_txt, and weaptype1\_txt are label-encoded. This step converts string labels into numeric values, which is required for most machine learning algorithms.
* **Splitting Features and Target:**
  + **Classification:**  
    The target (y) is defined as the success column (converted to integers), and the features (X) consist of all the remaining selected columns.
  + **Regression:**  
    The target is defined as casualties (the sum of nkill and nwound), and features include columns like iyear, attacktype1\_txt, weaptype1\_txt, targtype1\_txt, and region\_txt.

**1.4 Train-Test Split**

* **Data Partitioning:**  
  The data is split into training and testing sets using an 80/20 ratio, ensuring reproducibility by setting a fixed random\_state (42).

**2. Exploratory Data Analysis (EDA)**

**2.1 Data Inspection**

* **Columns and Sample Data:**  
  The initial columns and a few rows are printed to understand the structure and contents of the dataset.

**2.2 Target Variable Distribution**

* **Classification:**  
  The distribution of the success variable is examined. In earlier runs (from similar notebooks), it was noted that successful attacks greatly outnumber unsuccessful ones, indicating a class imbalance. This imbalance is important when interpreting performance metrics and might necessitate adjustments such as using class weights.

**2.3 Feature Insights (Optional)**

* **Feature Importances:**  
  Post-modeling, feature importance scores are computed. These scores help identify which features contribute most to the prediction. In the classification task, for example, features like targtype1\_txt, iyear, and attacktype1\_txt were found to be the most influential.

**3. Modeling Techniques**

The analysis includes two main modeling approaches: a classification model to predict attack success and a regression model to predict casualties.

**3.1 Classification Model**

**3.1.1 Model Selection and Hyperparameter Tuning**

* **Algorithm:**  
  A Random Forest Classifier is chosen due to its robustness and ability to handle both numerical and categorical data (after encoding).
* **Hyperparameter Grid Search:**  
  The model is tuned using GridSearchCV over a grid of parameters:
  + n\_estimators: [100, 200]
  + max\_depth: [None, 5, 10]
  + min\_samples\_split: [2, 5]  
    The grid search uses 3-fold cross-validation with accuracy as the scoring metric.
* **Outcome:**  
  The best hyperparameters found in one execution were:  
  { 'max\_depth': None, 'min\_samples\_split': 5, 'n\_estimators': 200 }

**3.1.2 Model Evaluation**

* **Test Accuracy:**  
  The optimized model achieved a test accuracy of approximately **93.09%**.
* **Classification Report:**  
  Detailed evaluation metrics include:
  + **Class 0 (Unsuccessful Attacks):**
    - Precision: 0.76
    - Recall: 0.54
    - F1-Score: 0.63
  + **Class 1 (Successful Attacks):**
    - Precision: 0.94
    - Recall: 0.98
    - F1-Score: 0.96
* **F1 Scores:**  
  In addition to the F1-score for the positive class (≈0.962), the macro-averaged F1-score was around 0.796 and the weighted F1-score was about 0.925.
* **Feature Importance:**  
  The model outputs feature importances as follows:
  + targtype1\_txt: 0.208826
  + iyear: 0.197534
  + attacktype1\_txt: 0.190224
  + country\_txt: 0.122186
  + nkill: 0.102945
  + nwound: 0.078312
  + region\_txt: 0.054458
  + weaptype1\_txt: 0.045516

**3.2 Regression Model**

**3.2.1 Model Selection and Hyperparameter Tuning**

* **Objective:**  
  Predict the total number of casualties (i.e., the sum of nkill and nwound).
* **Algorithm:**  
  A Random Forest Regressor is used.
* **Hyperparameter Grid Search:**  
  A similar grid search is conducted with the following parameters:
  + n\_estimators: [100, 200]
  + max\_depth: [None, 10]
  + min\_samples\_split: [2, 5]  
    The grid search is set up with 3-fold cross-validation and uses negative mean squared error as the scoring metric.
* **Outcome:**  
  The best hyperparameters identified for regression were:  
  { 'max\_depth': 10, 'min\_samples\_split': 2, 'n\_estimators': 200 }

**3.2.2 Model Evaluation**

* **Performance Metrics:**  
  On the test set, the regression model achieved:
  + **Root Mean Squared Error (RMSE):** Approximately **33.16**
  + **R² Score:** Approximately **0.614**  
    These metrics indicate that while the model explains around 61.4% of the variance in casualties, there is still room for improvement.

**4. Results and Insights**

**4.1 Classification Task**

* **Overall Performance:**  
  The Random Forest classifier performed well, achieving high accuracy and excellent F1-scores for the dominant class (successful attacks). However, the lower recall for unsuccessful attacks suggests that further tuning or class balancing (e.g., using class\_weight="balanced") might be necessary.
* **Feature Impact:**  
  The computed feature importances indicate that target type (targtype1\_txt), year (iyear), and attack type (attacktype1\_txt) are key drivers in predicting attack success.

**4.2 Regression Task**

* **Predictive Accuracy:**  
  The regression model for predicting total casualties reached an RMSE of about 33.16 and an R² score of 0.614, suggesting a moderate fit. Improving the model might require additional features, more sophisticated feature engineering, or alternative algorithms.

**5. Conclusion and Next Steps**

**5.1 Summary**

* **Workflow:**  
  The analysis covered data cleaning, feature encoding, and both classification and regression modeling.
* **Classification Findings:**  
  The classifier achieves strong performance on the majority class but shows room for improvement in capturing the minority class.
* **Regression Insights:**  
  The regression model shows moderate predictive power, highlighting the complexity of predicting casualties.

**5.2 Future Directions**

* **Address Class Imbalance:**  
  Experiment with oversampling the minority class, undersampling the majority class, or incorporating class weights to improve recall for unsuccessful attacks.
* **Enhanced Feature Engineering:**  
  Explore additional features (e.g., temporal trends, geographic details, or external data) to boost model performance.
* **Model Experimentation:**  
  Consider using more advanced algorithms such as XGBoost or LightGBM, which might capture nonlinear relationships more effectively.
* **Robust Evaluation:**  
  Expand EDA with more visualizations and statistical analyses to better understand feature relationships and model behavior.