Network Traffic Classification using Logistic Regression

Project Overview:

This machine learning project focuses on classifying network traffic into two categories: benign and malicious. It uses flow-based features derived from network data and applies logistic regression to distinguish between normal and potentially harmful traffic.

Step-by-Step Explanation:

1. Dataset Overview:

The project utilizes two CSV files:

- `webgoat_flow_stats.csv`: Represents malicious network flows.
- `2017-05-02_kali-normal22_flow_stats.csv`: Represents benign network flows.

Both datasets contain flow statistics such as duration, packet sizes, and inter-arrival times. Each flow is labeled as either "Malicious" or "Benign".

2. Preprocessing:

The datasets are loaded using `pandas` and combined into a single dataframe. A new label column `Type` is added to identify the nature of each flow.

Then, throughput-based features are added:

- These are derived by dividing flow-based statistics by the `flowDuration`, for example:
- `flowLengthPerTime = flowLength / flowDuration`

These new features help capture traffic behavior over time.

3. Feature Selection:

From all available features, 39 important ones are selected for training the machine learning model.

These features include statistics like:

- `flowDuration`, `flowLength`, `packetSizeTotal`

- `IATMean`, `IATMax`, `fwdIATMin`, etc.

- All throughput-based features created above

4. Data Cleaning:

A custom function `clean_dataset()` is used to:

- Remove rows with missing or infinite values

- Split the data into features (X) and labels (y)

5. Data Splitting & Scaling:

The data is split into training and test sets using `train_test_split`.

Then, `StandardScaler` is applied to normalize the feature values.

6. Model Training using Logistic Regression:

- A logistic regression classifier is used.

- Hyperparameter tuning is done using `GridSearchCV` with cross-validation.

- Different combinations of:

- `class_weight`, `penalty`, `C`, `fit_intercept`

are tested.

7. Evaluation:

Once the best model is found, it is used to predict the test data.

The following metrics are calculated:

- **Accuracy**: Percentage of correctly predicted labels
- **Confusion Matrix**: Breakdown of True Positives, False Positives, etc.
- **ROC AUC**: Area under the ROC Curve
- **Recall**: True Malicious Detection Rate
- **F1 Score**: Harmonic mean of precision and recall

Confusion Matrix Summary:

The confusion matrix helps in understanding how well the model is able to classify benign and malicious traffic. It shows:

- True Positive: Correctly identified malicious flows
- False Positive: Benign flows wrongly classified as malicious
- True Negative: Correctly identified benign flows
- False Negative: Missed malicious flows

Conclusion:

This project demonstrates an effective pipeline for preprocessing, feature extraction, model training, and evaluation of network traffic classification using Logistic Regression. It provides useful insights into identifying malicious network activities based on flow-level statistics.