**Methodology:**

1. **Data Preprocessing**
   1. Handling Missing Values: Checked if the dataset has any missing values in the data. When a few were found, either imputed them by a more appropriate measure (eg the mean or median if the data is numerical) or deleted if they are rare.
   2. Feature Selection: Chose features for DDoS detection. According to the feature importances graph, the most important features are pkts, bytes, rate, srate and drate. Removed irrelevant or redundant columns, such as Unnamed: 0 (an index column).
   3. Label Encoding: Flgs, state, proto etc. converted all of them into numerical variables. This can be done with the help of the one and same method of one-hot encoding or a Label Encoding.
   4. Data Normalization: Normalized the identified numerical characteristics to a similar range, for instance, bringing them to the 0 to 1 scale, in order to engage all the features to learn in the model in a similar way.
2. **Addressing Class Imbalance:**

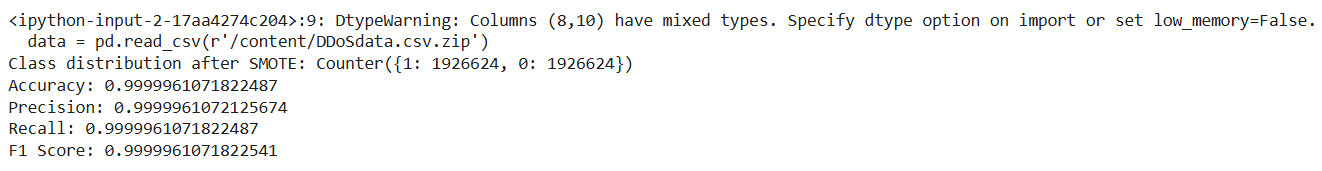
To overcome the problem, Synthetic Minority Over-sampling Technique (SMOTE) was used on the training data set. In SMOTE method, synthetic examples are created from the minority class (normal traffic) with an aim of providing balanced dataset because models perform well for the minority class.

1. **Feature Scaling:**

Standardization: Further, the selected numerical features were normalized to a common scale of mean zero, and the scale of the standard deviation of one. This scaling makes sense in the context of model, as it makes features as balanced as possible during training.

1. **Model Selection and Training:**
   1. Random Forest Classifier: Thus, the Random Forest classification algorithm was used due to its high tolerance to the ratio of classes. This classifier works according to a decision tree so this algorithm is quite good at generalization in respect of different distributions of data.
   2. Training the Model: It was divided into training and testing set as it is common, with 80% of the data set used for training and 20% for testing. The Random Forest model was then used to train the processed version of the training data. It also indicates that while training, the model was capable of identifying features that would differentiate the DDoS traffic from normal traffic.
2. **Feature Importance Analysis:**
   1. Extracting Feature Importance: As it often is with machine learning models, after the training f the Random Forest model, it offered the value of feature importance to the chaos. This analysis is useful for identifying which of the network parameters are most important for differentiating between an attack and normal traffic.
   2. Plotting Feature Importances: Feature importance values were then plotted in a bar chart to allow direct comparison between features and to better help with interpretation.
3. **Model Evaluation**
   1. Confusion Matrix: A confusion matrix was used to measure the performance of the external test set. The matrix in the figure represents True Positives (number of correctly classified DDoS attack), True Negatives (number of correctly classified normal traffic), False Positives and False Negatives. In certain instances, specifically in this case, the model’s performance is good with little errors made in sample classification.
   2. Classification Metrics:
      1. Accuracy: Calculates the rate of accurate predictions as a proportion of total many of predictions.
      2. Precision and Recall: These metrics show that the target class of the model is identifying the DDoS traffic without falsely triggering an alarm (precision) as well as identifying all the true DDoS cases (recall).
      3. F1 Score: One that takes into account both how many of the relevant documents were correctly identified and how many in total were captured.
      4. ROC-AUC Score: Assesses the model’s performance on different classes at different thresholds–providing useful information beyond just a model’s cutoff.
4. **Interpretation:**
   1. Insights from Feature Importances: Upon using the feature importance values, significant conclusions were derived. Data such as pkts, bytes, srate were considered to make a big influence on the decision of a model of the occurrence and identification of DDoS traffic, affirmatively highlighting packet count and data rate.
   2. Confusion Matrix Interpretation: From the confusion matrix results that were obtained it was observed that the model had very high accuracy and very few misclassifications were made and this could imply that the model has good generality to unseen data in this dataset.

**Results:**

****

**Class Distribution after SMOTE**:

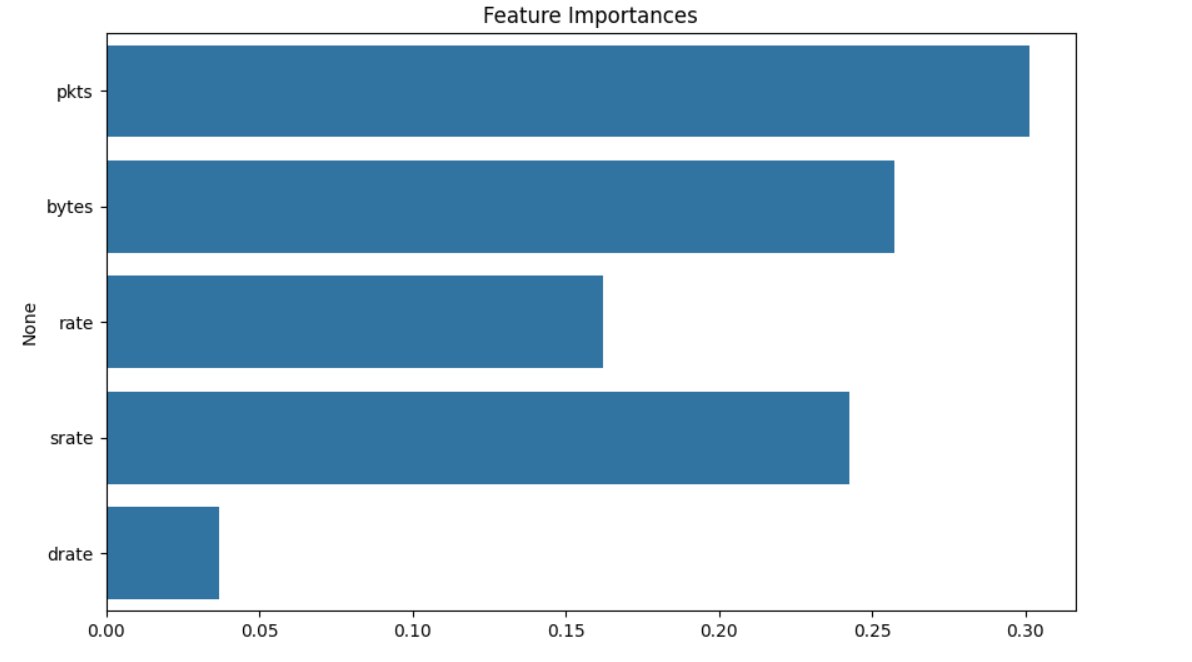
The data has been balanced using SMOTE (Synthetic Minority Over-sampling Technique), which is often used to address class imbalance by synthetically generating samples for the minority class. The resulting class distribution shows a balanced dataset with an equal number of samples for each class: Counter({1: 1926624, 0: 1926624}).

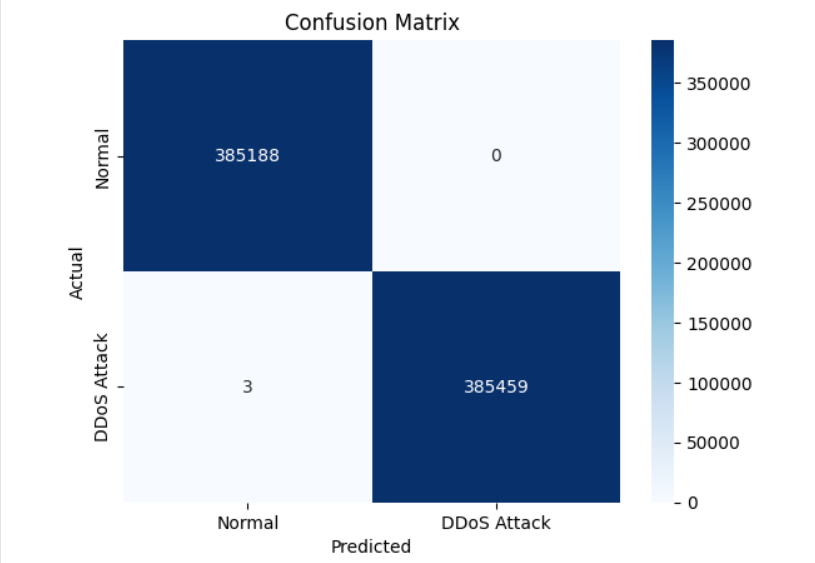
**Accuracy**: 0.999996, which means the model correctly classified nearly 100% of the samples.

**Precision**: 0.999996, indicating that almost all the instances the model classified as positive were true positives.

**Recall**: 0.999996, meaning that the model was able to identify nearly all the actual positive instances.

**F1 Score**: 0.999996, showing a high balance between precision and recall.



This graph shows how important each of the columns in your data set is as observed by the Random Forest model. Feature importance values tell how useful each feature was in aiding the model to correctly predict on whether the record in question is a DDoS attack or normal traffic. Most important features are pkts(packets) and srate (source rate) as they are the top contributors.

This graph is known as the confusion matrix which gives an idea of how well the model performed is in separating “Normal” from “DDoS Attack” traffic.