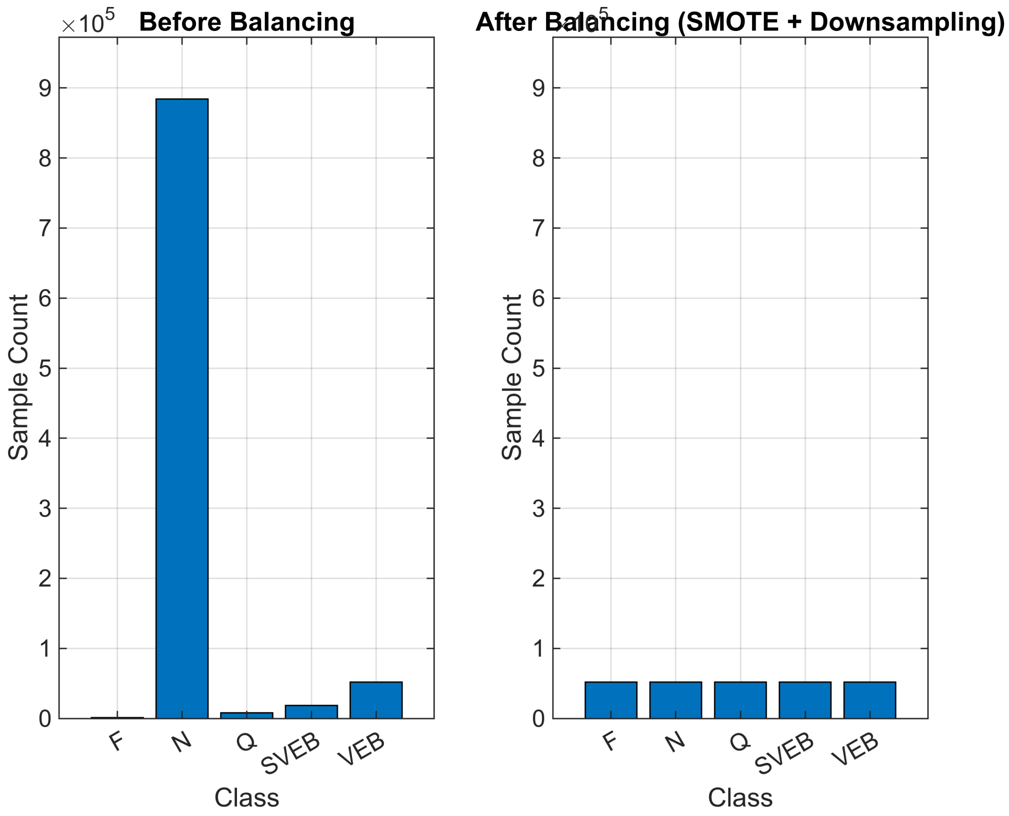
SMOTE

```
data = T_normalized;

% Separate features and labels
original_features = data{:, 3:end};
original_labels = string(data{:, 2});
```

```
% original_features = all features (full dataset)
% original_labels = all class labels

target_classes = ["N","VEB","SVEB", "Q", "F"];
target_count = 51669;  % Target count based on minority class (VEB)

% Custom SMOTE function picks a random set of 51669 samples from the normal
% class with oversampling the other classes to match the minority class
[X_full_bal, Y_full_bal] = customSMOTE_to_target_all(original_features,
original_labels, target_classes, target_count);
```

```
% Before Balancing
original_labels = string(original_labels);
[unique_labels_before, ~, idx_before] = unique(original_labels);
counts_before = accumarray(idx_before, 1);

% After Balancing
Y_full_bal = string(Y_full_bal);  % Balanced labels from SMOTE
[unique_labels_after, ~, idx_after] = unique(Y_full_bal);
counts_after = accumarray(idx_after, 1);

% Plotting
figure;
subplot(1, 2, 1)
bar(categorical(unique_labels_before), counts_before)
title('Before Balancing')
ylabel('Sample Count')
xlabel('Class')
ylim([0, max([counts_before; counts_after]) * 1.1])
grid on

subplot(1, 2, 2)
bar(categorical(unique_labels_after), counts_after)
title('After Balancing (SMOTE + Downsampling)')
ylabel('Sample Count')
xlabel('Class')
ylim([0, max([counts_before; counts_after]) * 1.1])
grid on
```



```
% Converts to string arrays for consistency
original_labels = string(original_labels);    % Before balancing
Y_full_bal = string(Y_full_bal);              % After balancing (output from
SMOTE)

% Counts original labels
[unique_before, ~, idx_b] = unique(original_labels);
count_before = accumarray(idx_b, 1);

% Counts balanced labels
[unique_after, ~, idx_a] = unique(Y_full_bal);
count_after = accumarray(idx_a, 1);

% Ensure label order matches
[all_classes, ia_b, ib_a] = union(unique_before, unique_after);
all_counts_before = zeros(length(all_classes), 1);
all_counts_after = zeros(length(all_classes), 1);

% Fill counts
[~, loc_b] = ismember(all_classes, unique_before);
[~, loc_a] = ismember(all_classes, unique_after);
all_counts_before(loc_b > 0) = count_before(loc_b(loc_b > 0));
all_counts_after(loc_a > 0) = count_after(loc_a(loc_a > 0));

% Create and display table
T = table(all_classes, all_counts_before, all_counts_after, ...
    'VariableNames', {'Class', 'Before_SMOTE', 'After_SMOTE'});
```

```
disp(T)
```

```
    Class      Before_SMOTE    After_SMOTE
    ______     ____________    ___________

    "F"                1256       51669
    "N"          8.8421e+05       51669
    "Q"                7975       51669
    "SVEB"            18540       51669
    "VEB"             51669       51669
```