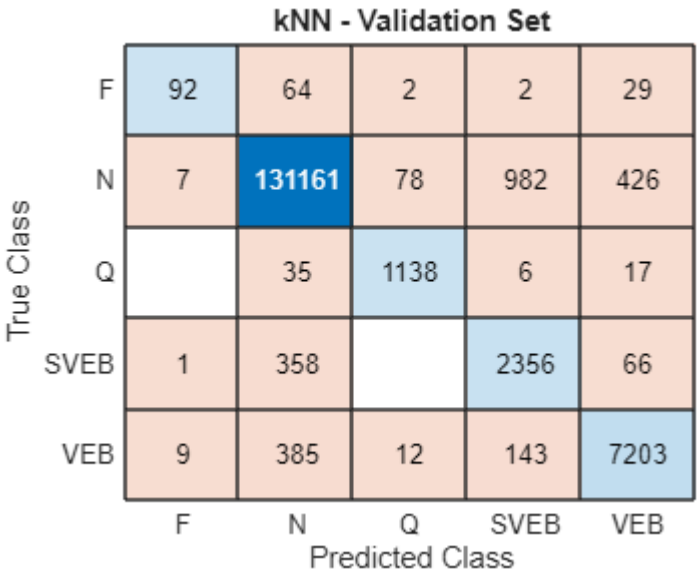
Model Training: kNN

```
rng(42);
knn_model = fitcknn(X_train, Y_train, ...
    'NumNeighbors', 37, ...
    'Distance', 'euclidean', ...
    'Standardize', true, ...
    'ClassNames', unique(Y_train));
```

```
Y_val_pred = predict(knn_model, X_val);

% Ensure same type
Y_val = string(Y_val);
Y_val_pred = string(Y_val_pred);

% Confusion Matrix
figure;
confusionchart(Y_val, Y_val_pred);
title('kNN - Validation Set');
```



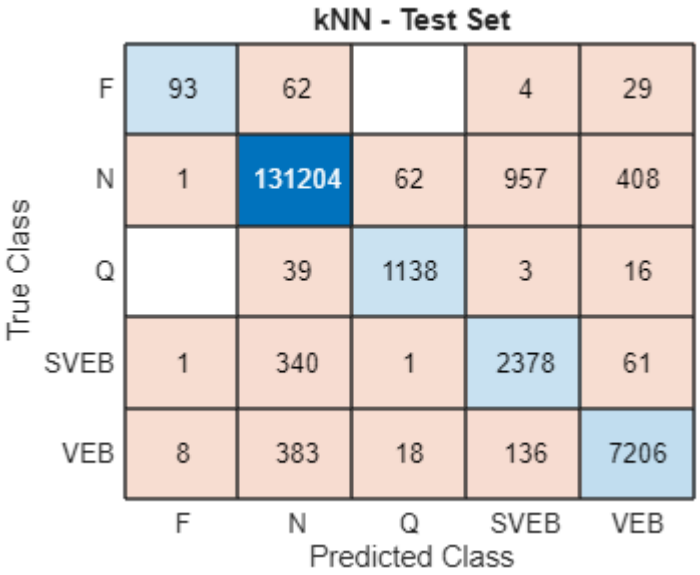

```
% Accuracy
val_accuracy = sum(Y_val_pred == Y_val) / numel(Y_val);
fprintf('kNN Validation Accuracy: %.2f%%\n', val_accuracy * 100);
```

```
kNN Validation Accuracy: 98.19%
```

```
Y_test_pred = predict(knn_model, X_test);

Y_test = string(Y_test);
Y_test_pred = string(Y_test_pred);
```

```
% Confusion Matrix
figure;
confusionchart(Y_test, Y_test_pred);
title('kNN - Test Set');
```




```
% Accuracy
test_accuracy = sum(Y_test_pred == Y_test) / numel(Y_test);
fprintf('kNN Test Accuracy: %.2f%%\n', test_accuracy * 100);
```

```
kNN Test Accuracy: 98.25%
```

```
classes = cellstr(unique([Y_test; Y_test_pred]));
precision = zeros(length(classes), 1);
recall = zeros(length(classes), 1);
f1score = zeros(length(classes), 1);

for i = 1:length(classes)
    class = classes{i};
    TP = sum(Y_test_pred == class & Y_test == class);
    FP = sum(Y_test_pred == class & Y_test ~= class);
    FN = sum(Y_test_pred ~= class & Y_test == class);
    precision(i) = TP / (TP + FP + eps);
    recall(i) = TP / (TP + FN + eps);
    f1score(i) = 2 * (precision(i) * recall(i)) / (precision(i) + recall(i) + eps);
end

metrics_table = table(classes, precision, recall, f1score, ...
    'VariableNames', {'Class', 'Precision', 'Recall', 'F1_Score'});
disp(metrics_table);
```

Class	Precision	Recall	F1_Score

```
    {'F'   }     0.90291     0.49468    0.63918
    {'N'   }     0.99376     0.98923    0.99149
    {'Q'   }     0.93355     0.95151    0.94244
    {'SVEB'}     0.68373     0.85509    0.75987
    {'VEB' }     0.93342     0.92969    0.93155
```

```
macro_f1 = mean(f1score);
fprintf('kNN Macro F1-Score: %.2f\n', macro_f1);
```

```
kNN Macro F1-Score: 0.85
```