|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Referência | |
|  |  | Evento | No Event |
| Predito | Evento | A (TP) | B (FP) |
| No Event | C (FN) | D (TN) |

# Métricas

Sensitivity | Recall = TP/(TP+FN)  
Quantos eventos reais (positivos) foram corretamente identificados.

Specificity = TN/(FP+TN)  
Quantos não eventos (negativos) foram corretamente identificados.

Prevalence = (TP+FN)/(TP+FP+FN+TN)  
Proporção de eventos reais na população (eventos positivos).

PPV (Pos Pred Value) | Precision [1] = (sensitivity \* prevalence)/((sensitivity\*prevalence) + ((1-specificity)\*(1-prevalence))) = TP/(TP+FP)  
Proporção de eventos previstos como positivos que eram realmente positivos.

NPV (Neg Pred Value) | Precison [0] = (specificity \* (1-prevalence))/(((1-sensitivity)\*prevalence) + ((specificity)\*(1-prevalence))) = TN/(FN+TN)  
Proporção de predições negativas que eram realmente negativas.

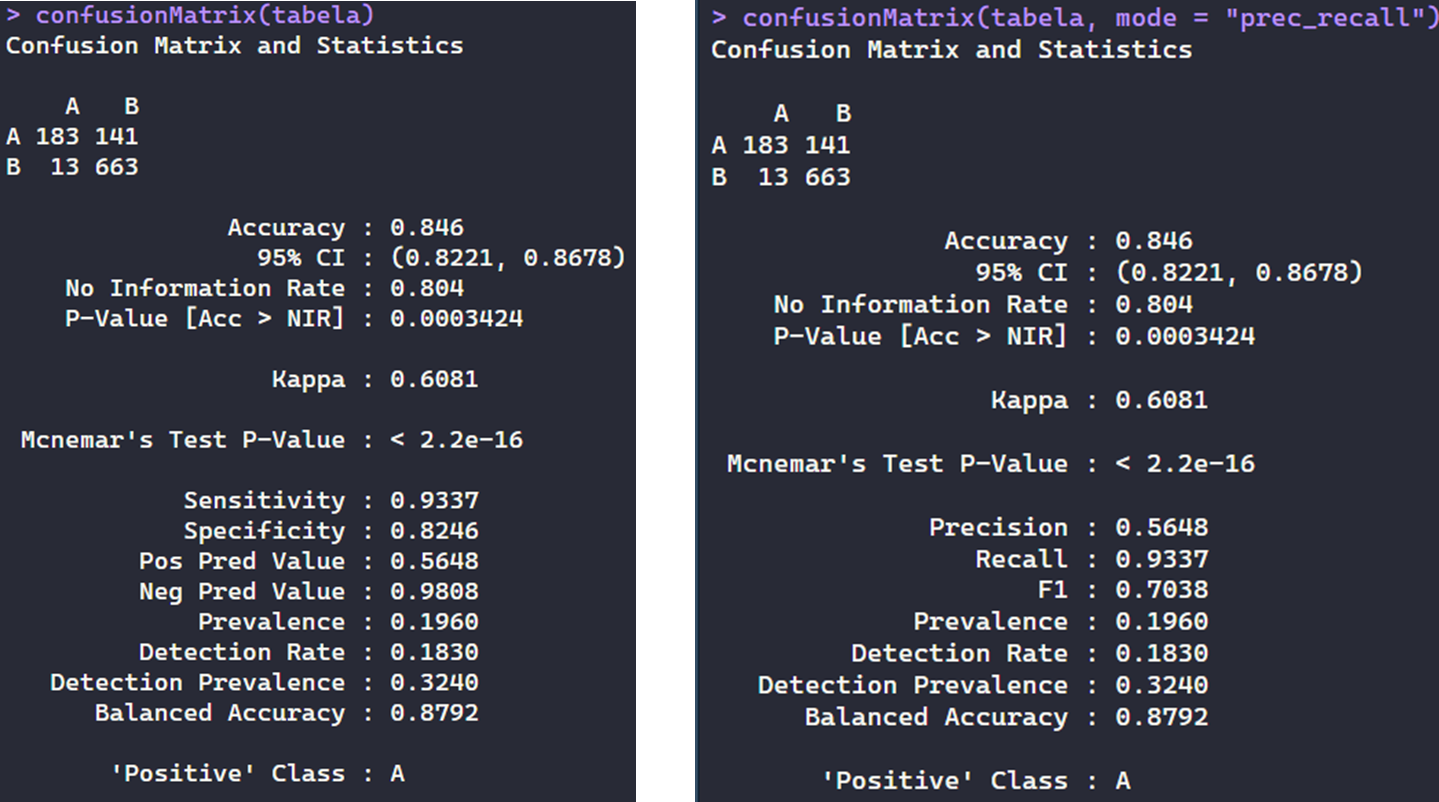
Detection Rate = TP/(TP+FP+FN+TN)  
Proporção de eventos reais detectados em relação ao total da população.

Detection Prevalence = (TP+FP)/(TP+FP+FN+TN)  
Proporção de predições positivas feitas pelo modelo (independentemente de serem corretas ou não).

Balanced Accuracy = (sensitivity+specificity)/2  
Média das taxas de acerto para eventos e não eventos. Usada quando há desequilíbrio entre classes.

F1 = (1+beta^2)\*precision\*recall/((beta^2 \* precision)+recall)  
Média harmônica entre precisão e recall

# Saida no R:



É expresso duas métricas novas

**Kappa**: Mede a concordância entre as predições do modelo e os rótulos reais, corrigindo pela concordância que poderia ocorrer ao acaso. Útil especialmente quando há desbalanceamento entre classes.

**McNemar's Test p-value**: Testa a simetria dos erros do modelo, especificamente se há uma diferença significativa entre os falsos positivos e falsos negativos.

# Saida no Python:

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente