

## Informe TP1: Reservas de Hotel

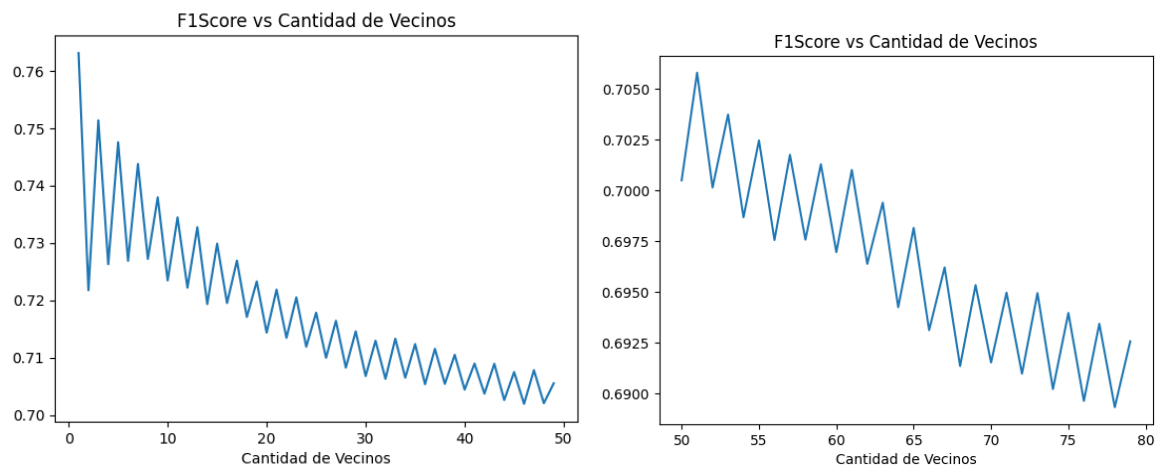
### Integrantes:

Agama Avila, Arely - 105829  
Martinez, Selene Anahi - 100439  
Meichtri, Melany - 102330

### Checkpoint 3:

#### a.KNN

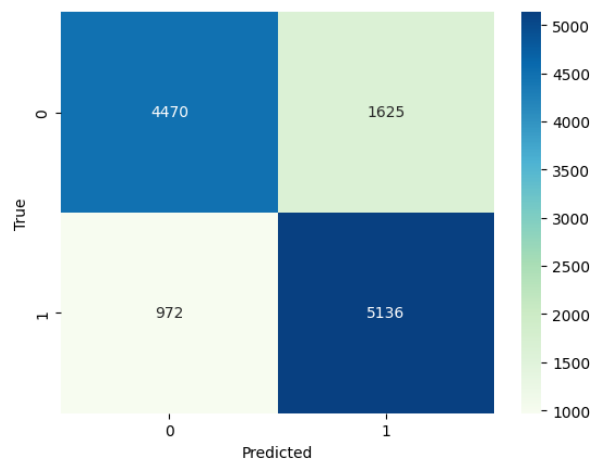
Vemos a relación de f1\_Score para rango de vecinos(0,50) y (50,80)



Concluimos que va disminuyendo el f1\_Score a medida que aumenta la cantidad de vecinos.

Usamos Random Search para buscar los mejores hiperparametros, obteniendo 11 vecinos, distancia:Manhattan y algoritmo: ball tree.

Asi nos quedaria la matriz de confusión:

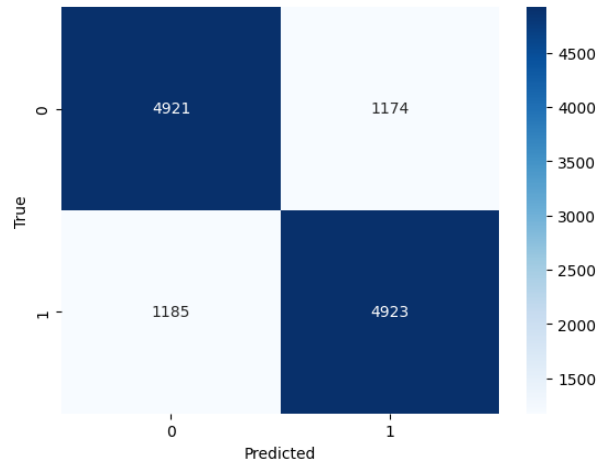


## B. SVM

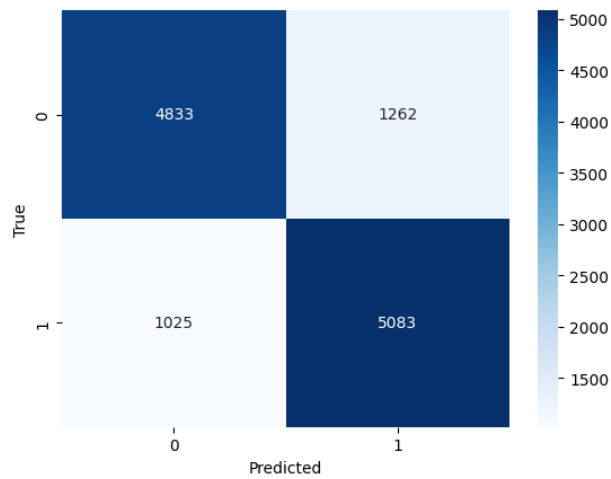
Usando Grid Search para obtener los mejores hiperparametros para cada kernel.

Kernel lineal

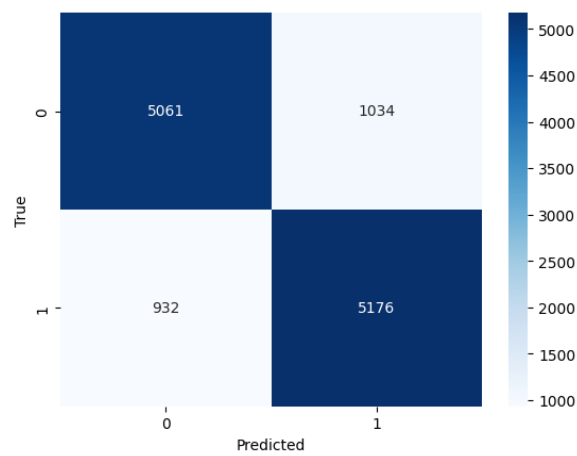
Obtenemos esta matriz de confusión:



Kernel Radial

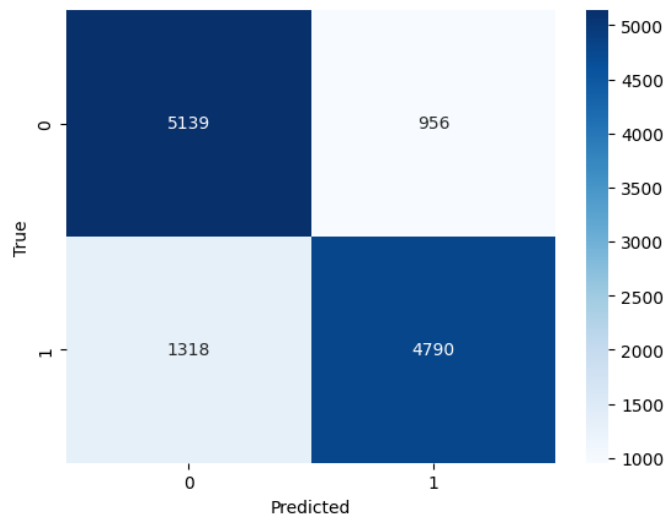


Kernel Polinomial



## D. XG BOOST

Mejoramos el learning rate aumentando lentamente desde 0.00` hasta 0.012 que vimos que era el óptimo basándose en cuánto disminuye std(desviación) y cuanto aumenta el mean(métrica). Así se vuelve más preciso en la predicción.

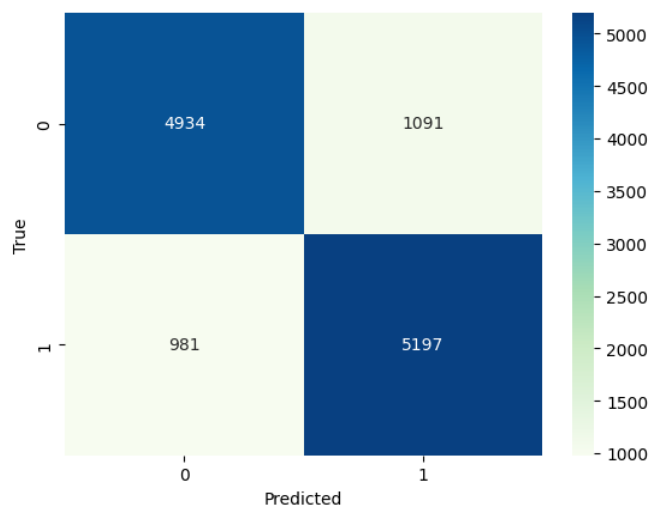


	precision	recall	f1-score	support
0	0.80	0.84	0.82	6095
1	0.83	0.78	0.81	6108
accuracy			0.81	12203
macro avg	0.81	0.81	0.81	12203
weighted avg	0.81	0.81	0.81	12203

## E. Voting Stacking Matriz de Confusión y métricas

Utilizamos los mejores knn, random forest y SVM radial.

```
accuracy: 0.8302056871261165
f1_score: 0.8337879030964223
recall: 0.8412107478148269
precision: 0.8264949109414759
```



### Stacking Matriz de confusión y métricas

Es un método de ensamble que demora mucho más tiempo que el voting pero mejora las métricas, deberíamos aumentar el max\_iter ya que no llegamos al limite según indica el método ejecutado.

```
accuracy: 0.8749487830861263  
f1_score: 0.8758542141230068  
recall: 0.8713175785043703  
precision: 0.8804383382401046
```

