BCI_TP4

October 20, 2022

Diplomatura en Ciencia de Datos, Aprendizaje Automático y sus Aplicaciones

Edición 2022

Mentoría: Data Science aplicado a BCI

Grupo 2

Integrantes: Gastón Briozzo, Pablo Ventura

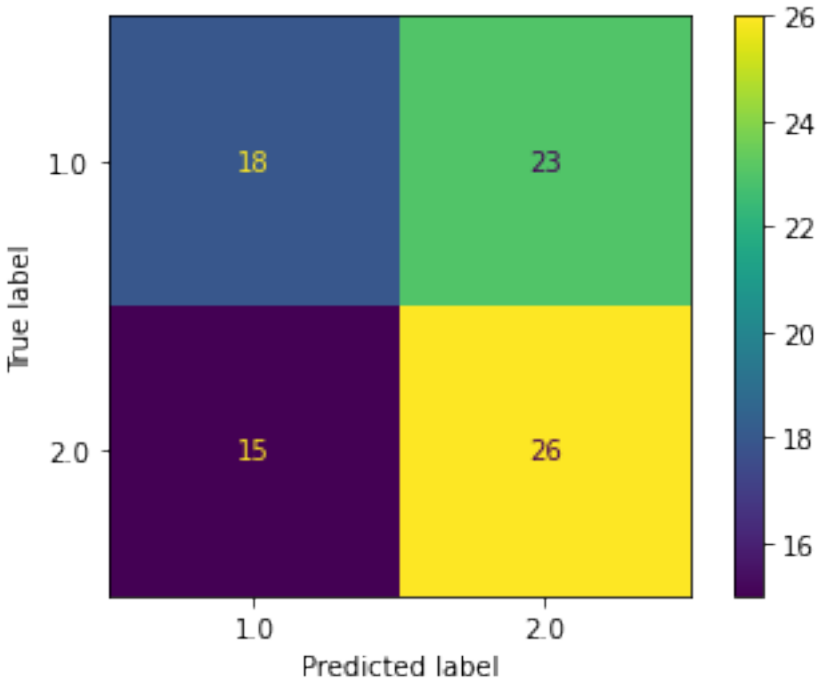
Profesor de Práctico: Juan Manuel Lopez

A) Benchmarking y desarrollo del algoritmo evaluador:

a) Utilice una clasificación aleatoria de los ejemplos para utilizar como benchmark de los resultados posteriores. Este benchmark representa el peor de los desempeños de clasificación posibles.

Emplearemos el clasificador aleatorio DummyClassifier de sklearn como benchmark en orden de sentar una base para la mínima presición aceptable.

Dado que las clases estan bien balanceadas, la accuracy será métrica suficiente.



	precision	recall	f1-score	support
1.0	0.46	0.39	0.42	41
2.0	0.47	0.54	0.50	41
accuracy			0.46	82
macro avg	0.46	0.46	0.46	82
weighted avg	0.46	0.46	0.46	82

b) Evalúe el desempeño/rendimiento de este benchmark bajo las métricas seleccionadas en el apartado anterior. Considere repetir este procedimiento algunas veces para obtener un promedio, máximo, mínimo u otro representante de estos resultados, ya que se trata de un proceso completamente aleatorio.

Haremos una descripción estadística de la precisión del clasificador aleatorio, tomando un total de 100 mediciones

```
Accuracy Mean Value: 0.5007317073170732
Accuracy Min  Value: 0.34146341463414637
Accuracy Max  Value: 0.6585365853658537
Accuracy StaD Value: 0.05456082522380473
```

Vemos que la accuracy promedio del clasificador aleatorio esta en torno al 50%, lo que es de esperarse en un problema binomial equilibrado.

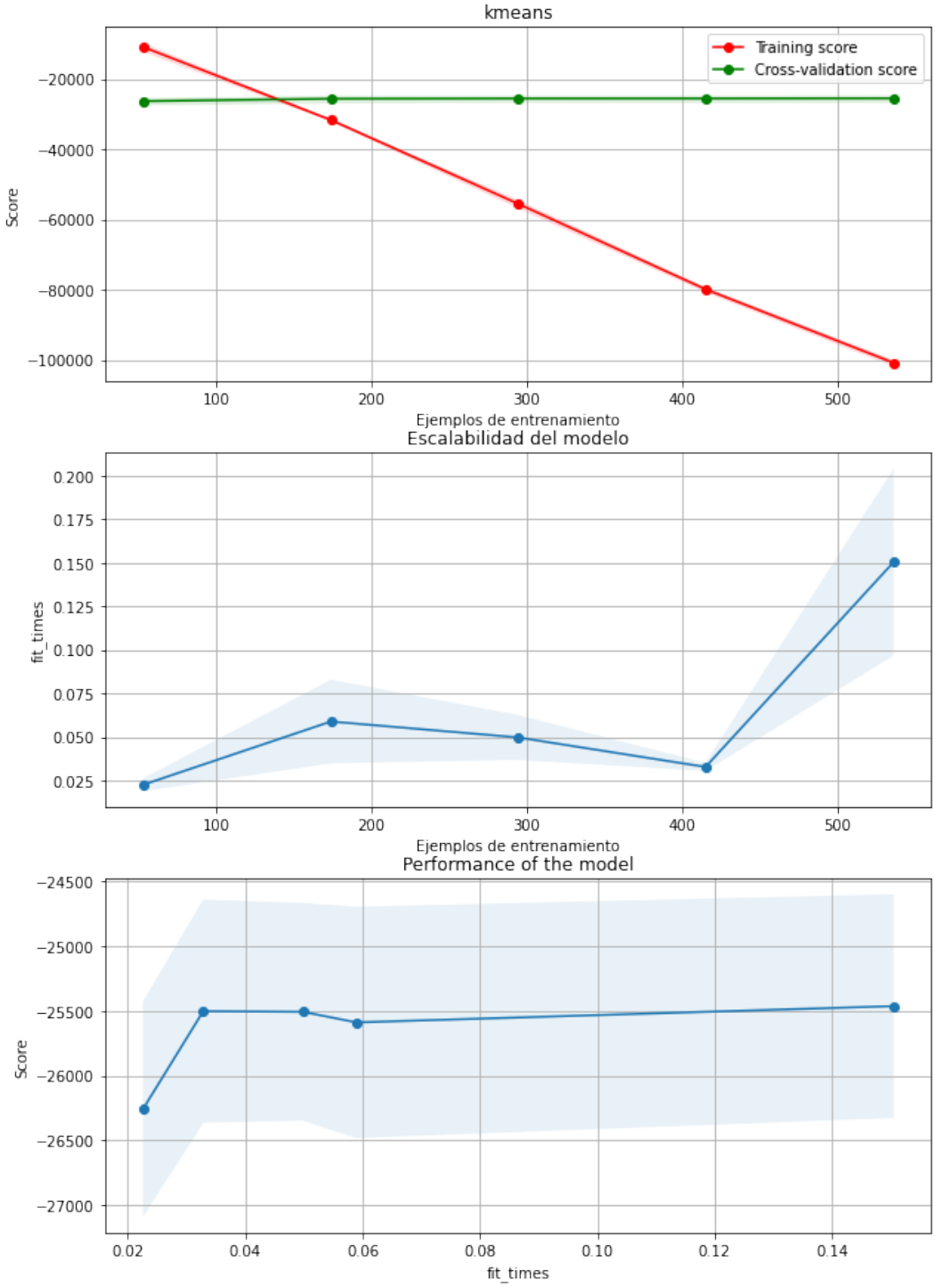
B) Búsqueda a grandes rasgos:

a) Utilicen la mayor cantidad de algoritmos de aprendizaje automático supervisado que puedan (mínimo 3). Creen para cada uno un modelo bajo el paradigma de dicho método y entrénelo con el dataset elegido. Opcional: generar curvas de progreso de métricas y funciones de pérdida a lo largo del entrenamiento.

K-means:

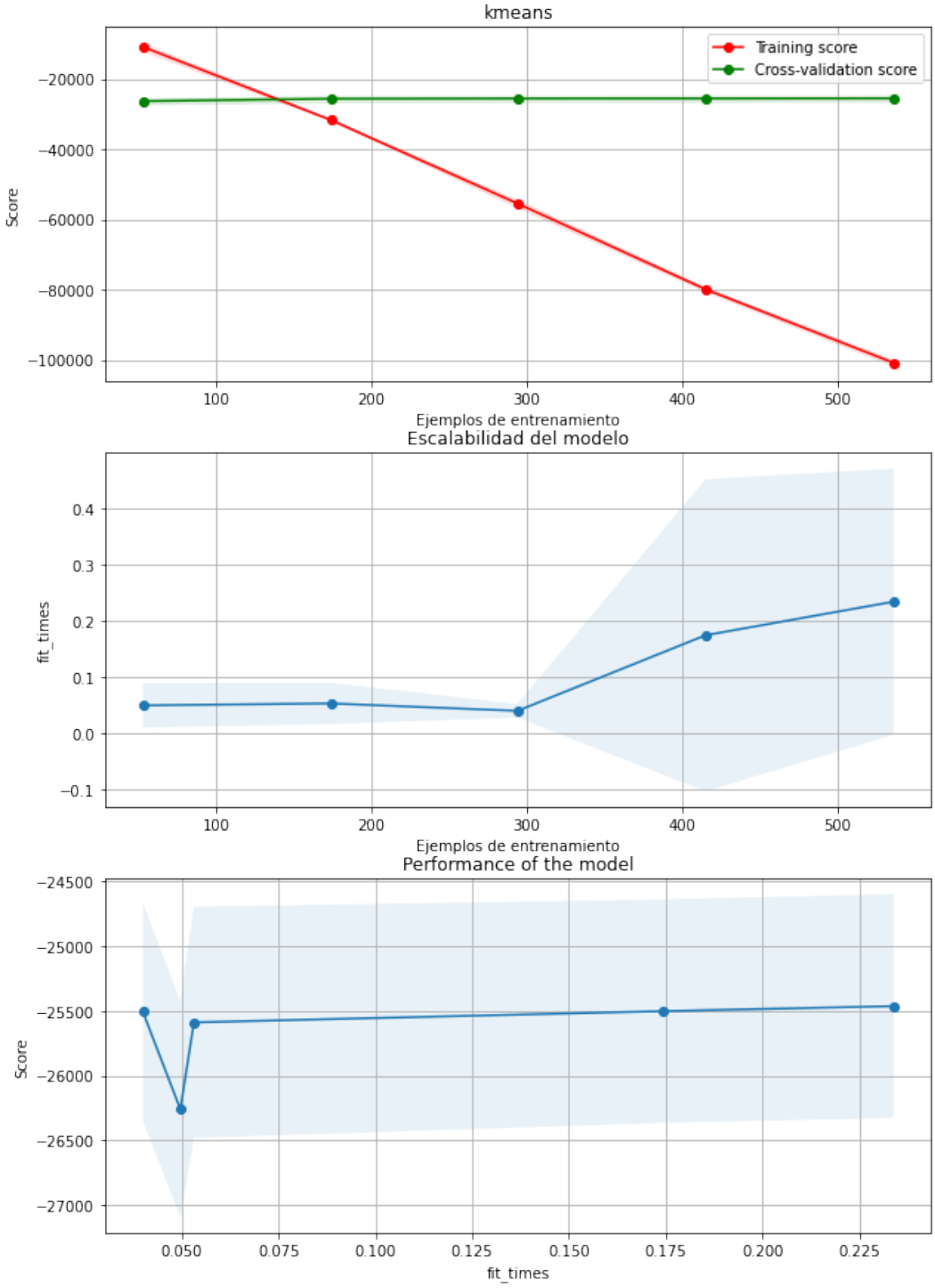
Para este algoritmo probamos variando el parametro "algorithm" con "elkan", "auto" y "full".

elkan:



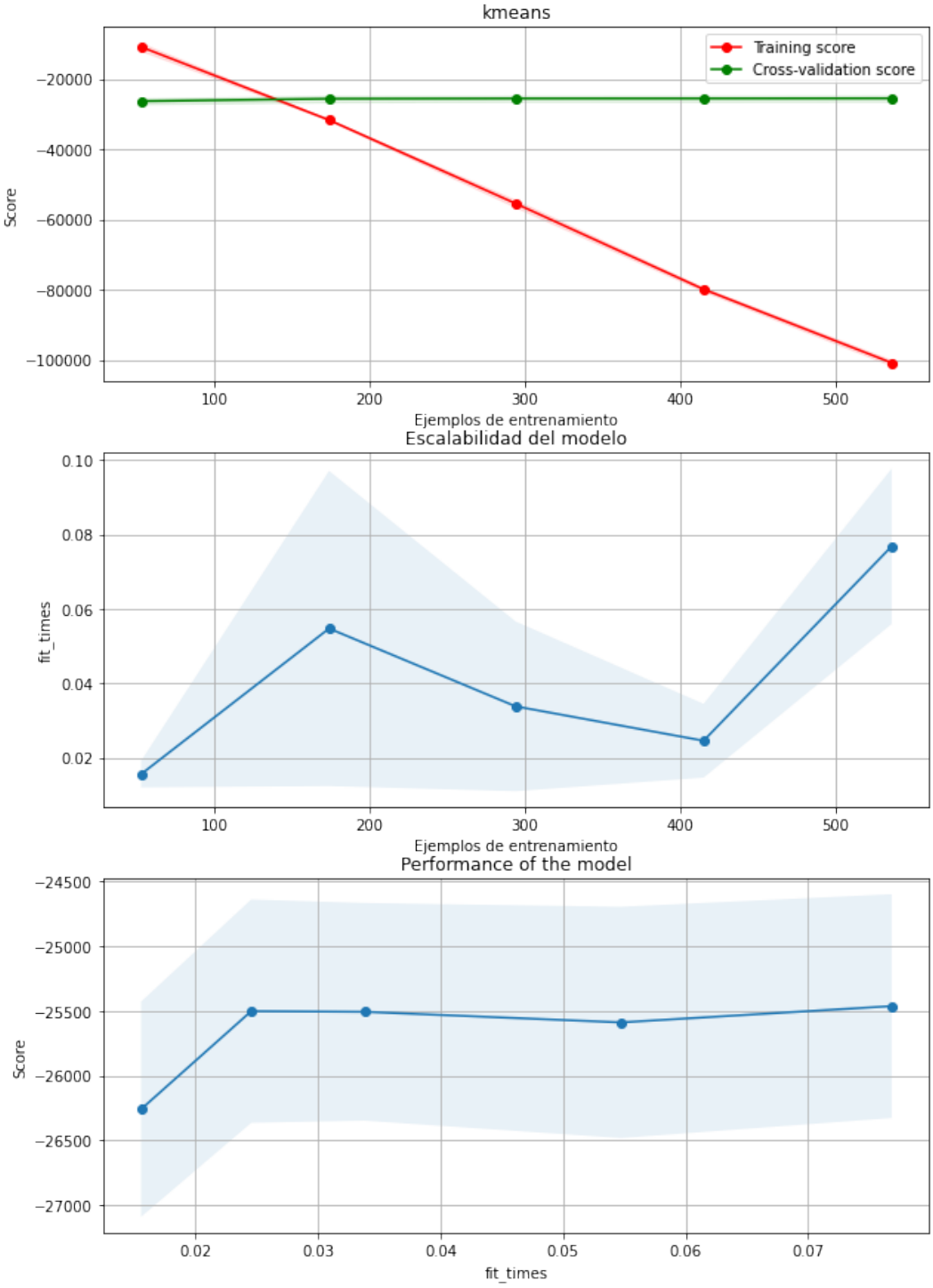
`Algorithm: elkan Accuracy: 0.7317073170731707`

auto:

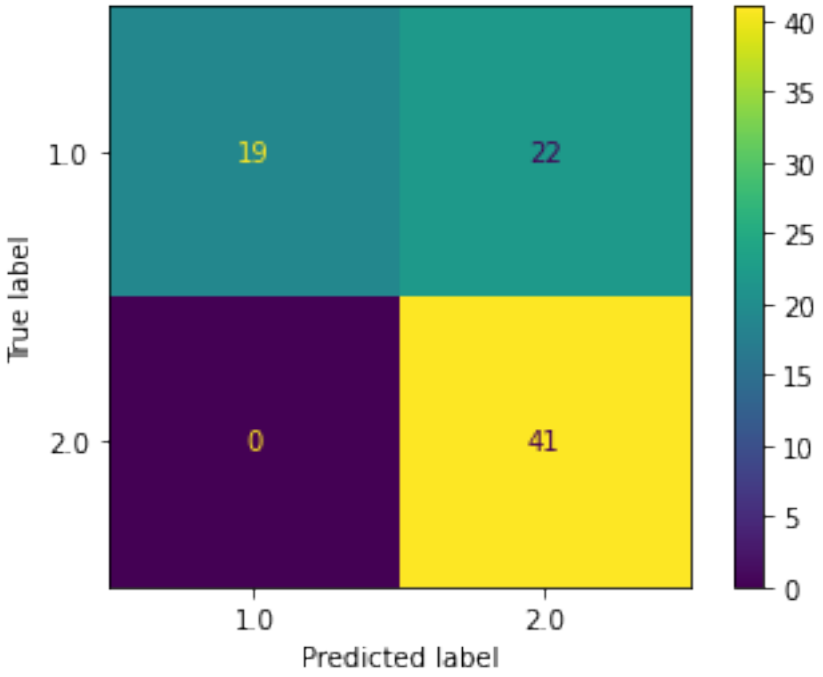


`Algorithm: auto Accuracy: 0.7317073170731707`

full:

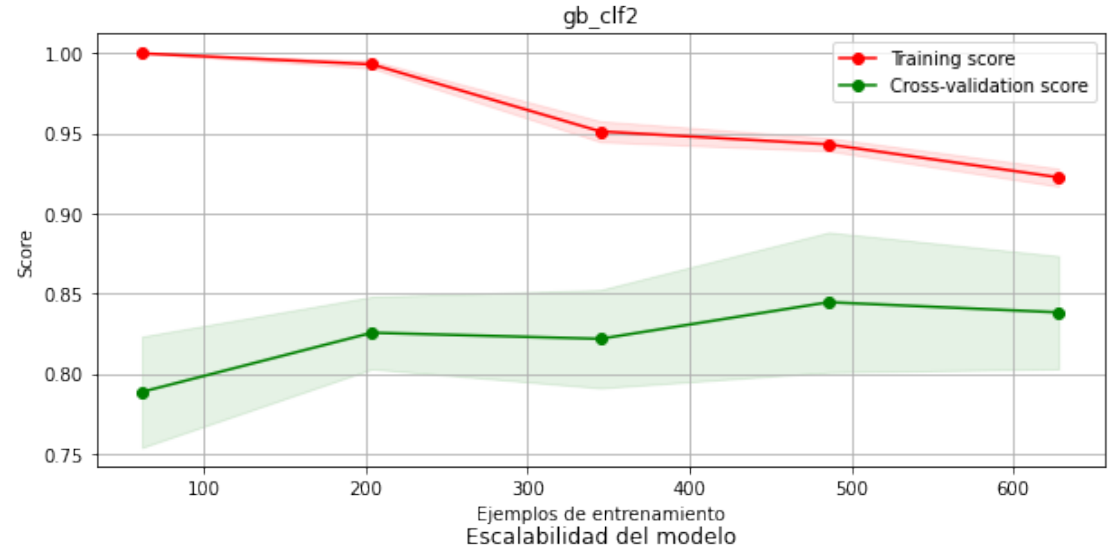


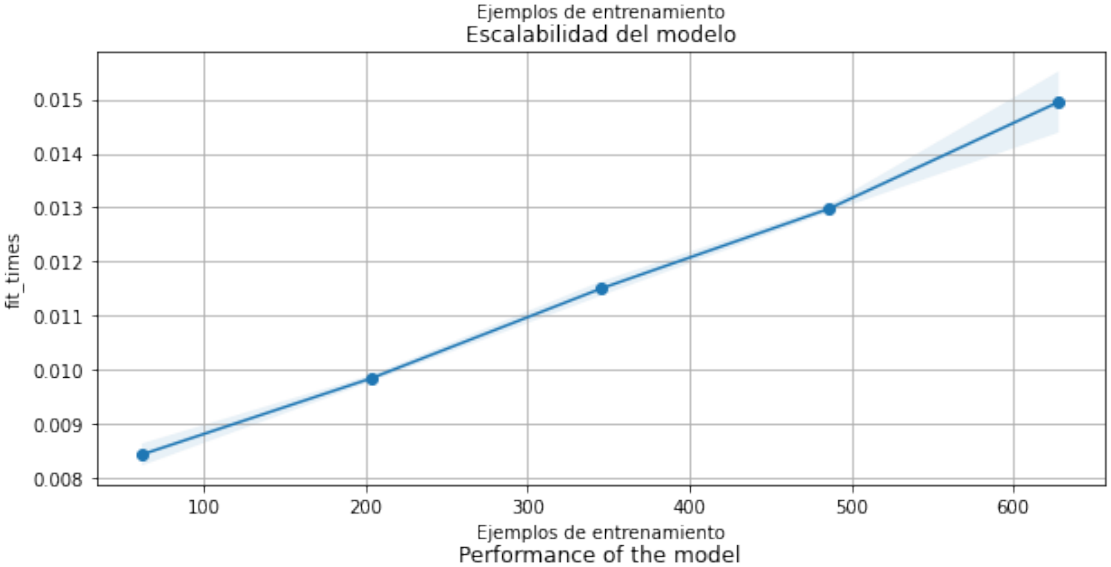
`Algorithm: full Accuracy: 0.7317073170731707`

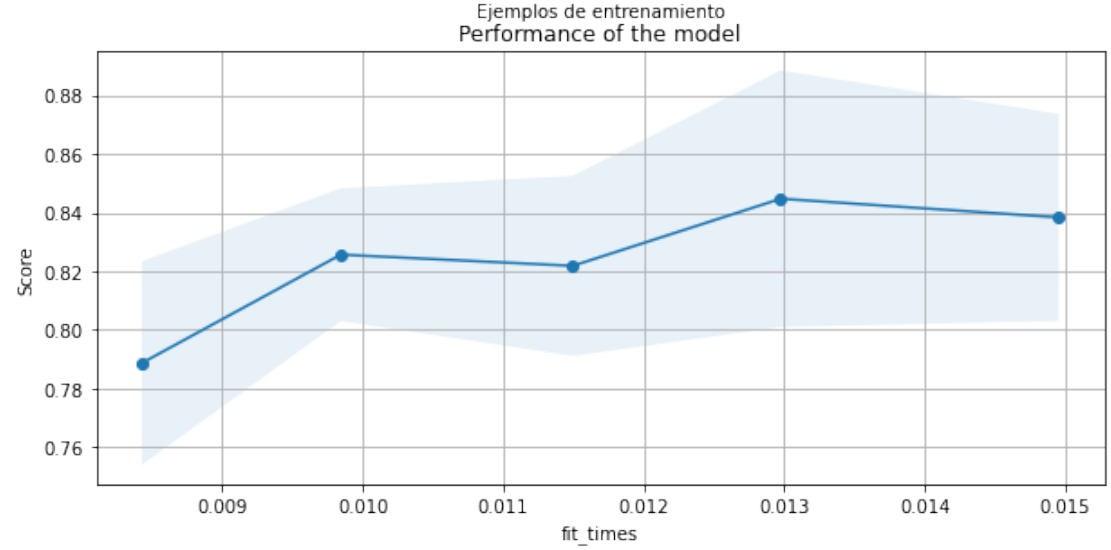


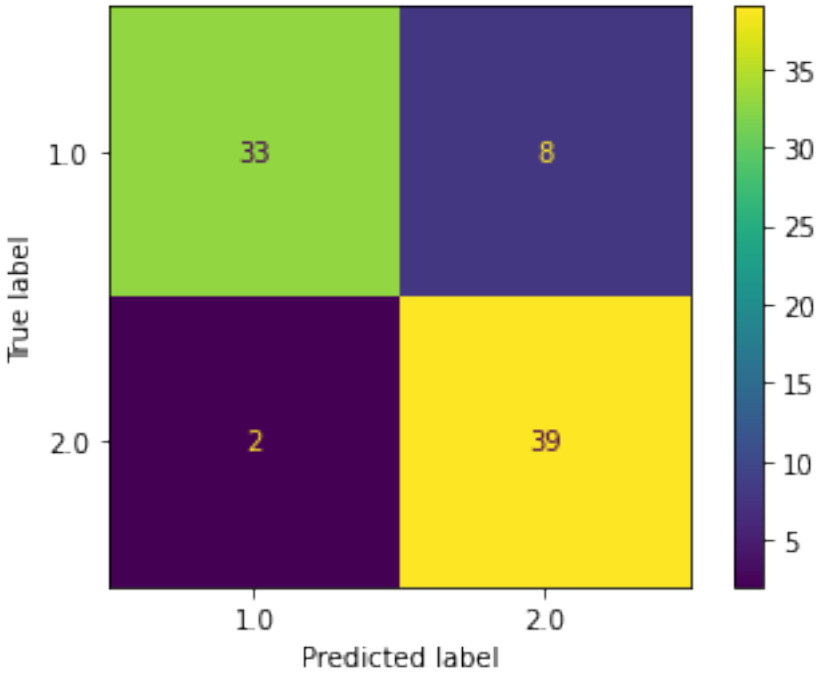
	precision	recall	f1-score	support
1.0	1.00	0.46	0.63	41
2.0	0.65	1.00	0.79	41
accuracy			0.73	82
macro avg	0.83	0.73	0.71	82
weighted avg	0.83	0.73	0.71	82

Random Forest GBDT



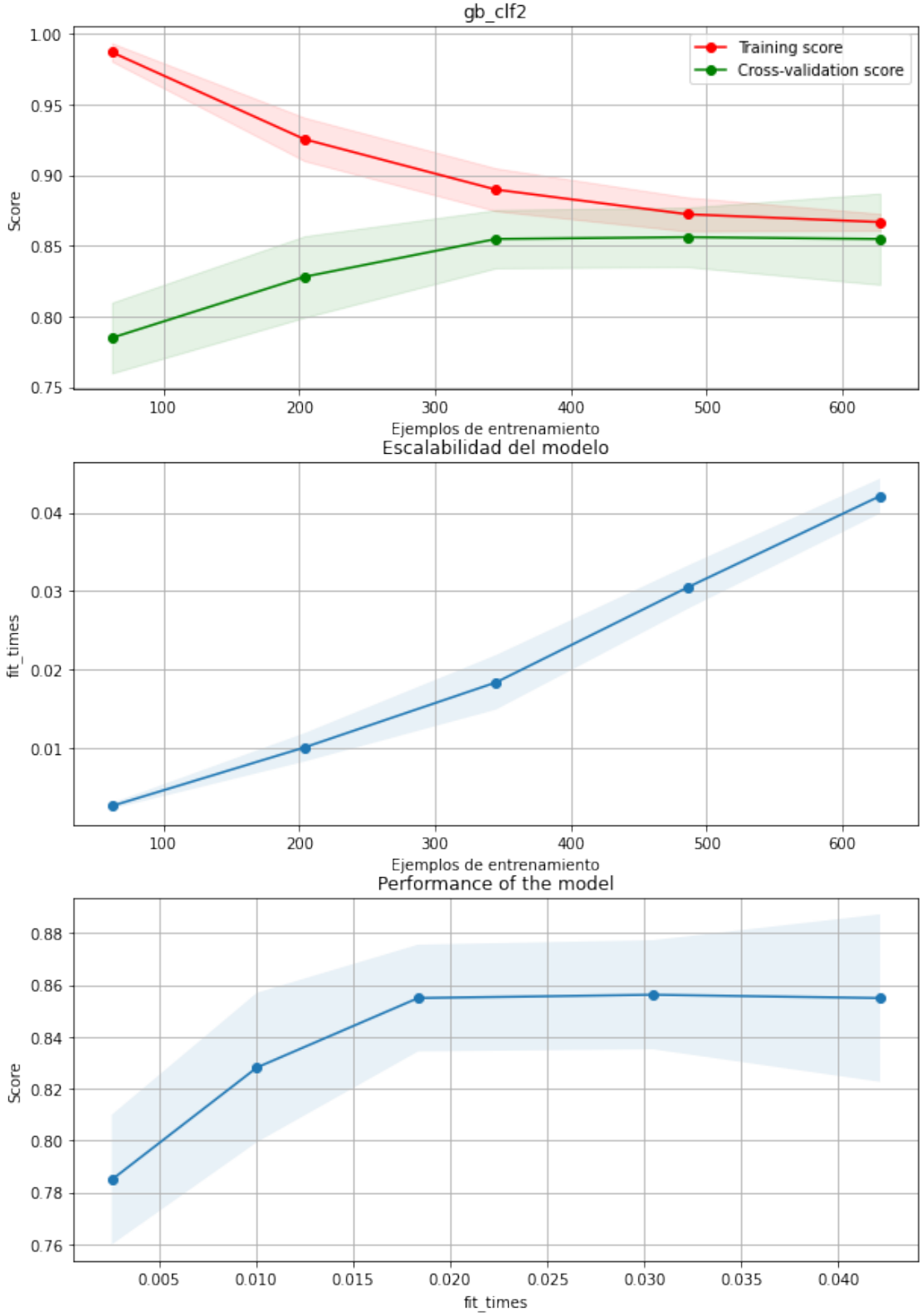


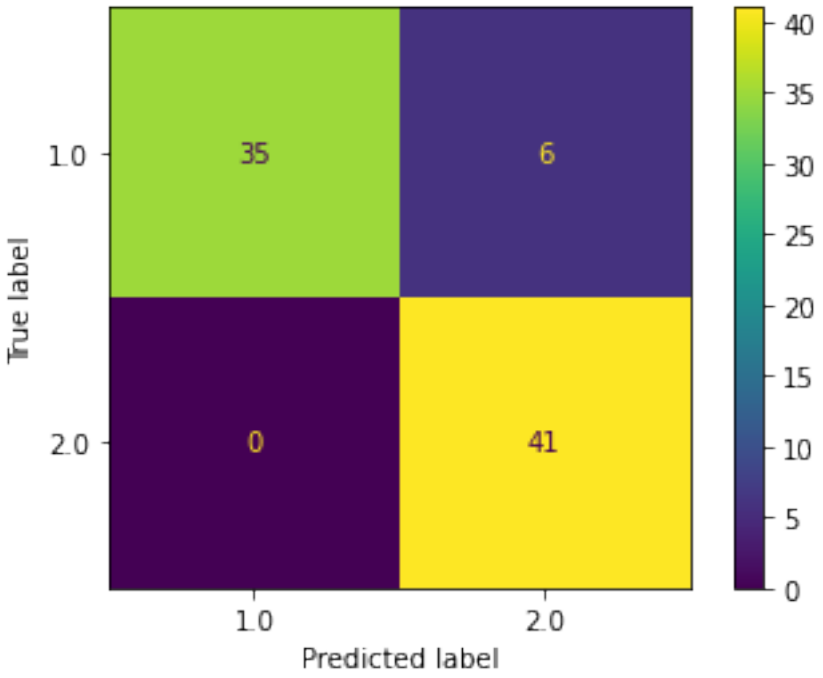




	precision	recall	f1-score	support
1.0	0.94	0.80	0.87	41
2.0	0.83	0.95	0.89	41
accuracy			0.88	82
macro avg	0.89	0.88	0.88	82
weighted avg	0.89	0.88	0.88	82

LinearSVC


gb_clf2
Training score
Cross-validation score
Score
Ejemplos de entrenamiento
Escalabilidad del modelo
fit_times
Ejemplos de entrenamiento
Performance of the model
Score
fit_times



	precision	recall	f1-score	support
1.0	1.00	0.85	0.92	41
2.0	0.87	1.00	0.93	41
accuracy			0.93	82
macro avg	0.94	0.93	0.93	82
weighted avg	0.94	0.93	0.93	82

c) Si encuentra resultados de las métricas analizadas o gráficos de entrenamiento ploteados que resulten destacables -no sólo porque sean valores más óptimos, sino también posibles casos extraños, situaciones de over y underfitting, etc- muéstrelos y analícelos en este inciso.

En particular la curva de aprendizaje de kmeans resulta dificilmente interpretable presentando valores sumamente negativos probablemente debido al hecho de que el algoritmo no sea adecuado para el problema.

C) Búsqueda puntualizada:

a) Con el modelo que presente mejores resultados, lleve a cabo una búsqueda ahora sí más detallista, variando los hiperparámetros y funciones de costo. Si estos métodos permiten variar la cantidad de instancias/épocas de entrenamiento, analice lo que sucede cuando varía las duraciones de entrenamiento.

Hacemos un gridsearch con los siguientes parametros:

```
param_grid = {
                "linearsvc__loss"             : ["hinge", "squared_hinge"],
                'linearsvc__C'                : [0.1, 1, 10],
                'linearsvc__intercept_scaling': [0.1, 1, 10]
            }
```

```
Window Size:  1024  Overlaping Fraction:  0.5  Best Accuracy:  0.86
Mejor Estimador LinearSVC(C=0.1, max_iter=100000, random_state=0, tol=1e-05))])


Window Size:  1024  Overlaping Fraction:  0.25  Best Accuracy:
0.8163265306122449
Mejor Estimador LinearSVC(C=0.1, intercept_scaling=0.1, max_iter=100000,
                          random_state=0, tol=1e-05))])


Window Size:  1024  Overlaping Fraction:  0.125  Best Accuracy:
0.8163265306122449
Mejor Estimador LinearSVC(C=10, intercept_scaling=10, loss='hinge',
                          max_iter=100000, random_state=0, tol=1e-05))])


Window Size:  1200  Overlaping Fraction:  0.5  Best Accuracy:
0.8809523809523809
Mejor Estimador LinearSVC(C=1, loss='hinge', max_iter=100000, random_state=0,
                          tol=1e-05))])


Window Size:  1200  Overlaping Fraction:  0.25  Best Accuracy:
0.9047619047619048
Mejor Estimador LinearSVC(C=10, intercept_scaling=0.1, loss='hinge',
                          max_iter=100000, random_state=0, tol=1e-05))])


Window Size:  1200  Overlaping Fraction:  0.125  Best Accuracy:
0.8154761904761905
Mejor Estimador LinearSVC(C=0.1, loss='hinge', max_iter=100000, random_state=0,
                          tol=1e-05))])


Window Size:  1240  Overlaping Fraction:  0.5  Best Accuracy:  0.9
Mejor Estimador LinearSVC(C=0.1, intercept_scaling=0.1, max_iter=100000,
```

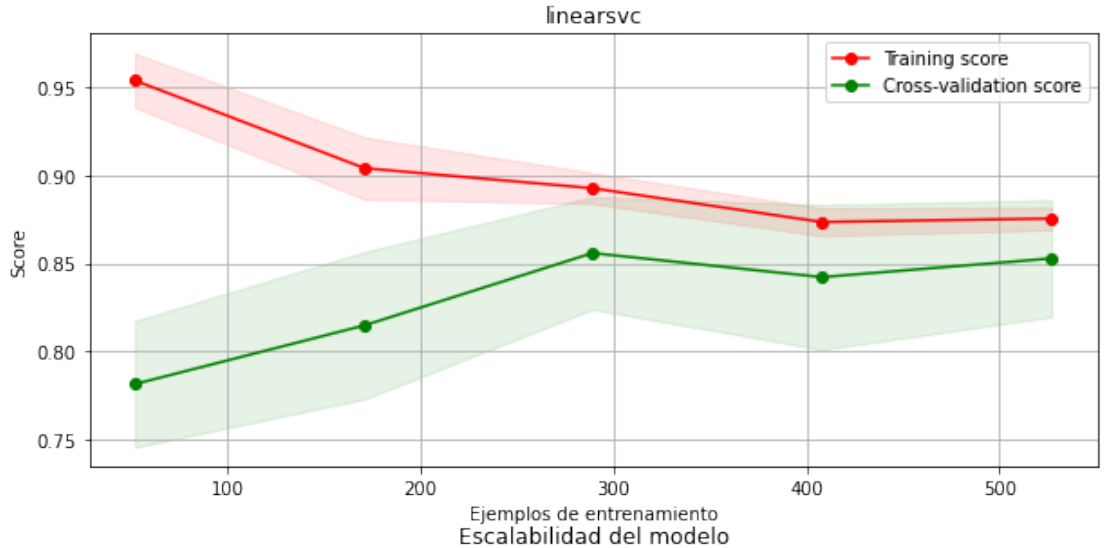
```
                          random_state=0, tol=1e-05))])

Window Size:  1240  Overlaping Fraction:  0.25  Best Accuracy:
0.9390243902439024
Mejor Estimador LinearSVC(C=1, intercept_scaling=0.1, loss='hinge',
                          max_iter=100000, random_state=0, tol=1e-05))])

Window Size:  1240  Overlaping Fraction:  0.125  Best Accuracy:
0.8536585365853658
Mejor Estimador LinearSVC(C=1, intercept_scaling=0.1, loss='hinge',
                          max_iter=100000, random_state=0, tol=1e-05))])

Window Size:  1300  Overlaping Fraction:  0.5  Best Accuracy:
0.8421052631578947
Mejor Estimador LinearSVC(C=0.1, intercept_scaling=0.1, loss='hinge',
                          max_iter=100000, random_state=0, tol=1e-05))])

Window Size:  1300  Overlaping Fraction:  0.25  Best Accuracy:
0.8701298701298701
Mejor Estimador LinearSVC(C=0.1, intercept_scaling=0.1, max_iter=100000,
                          random_state=0, tol=1e-05))])
```

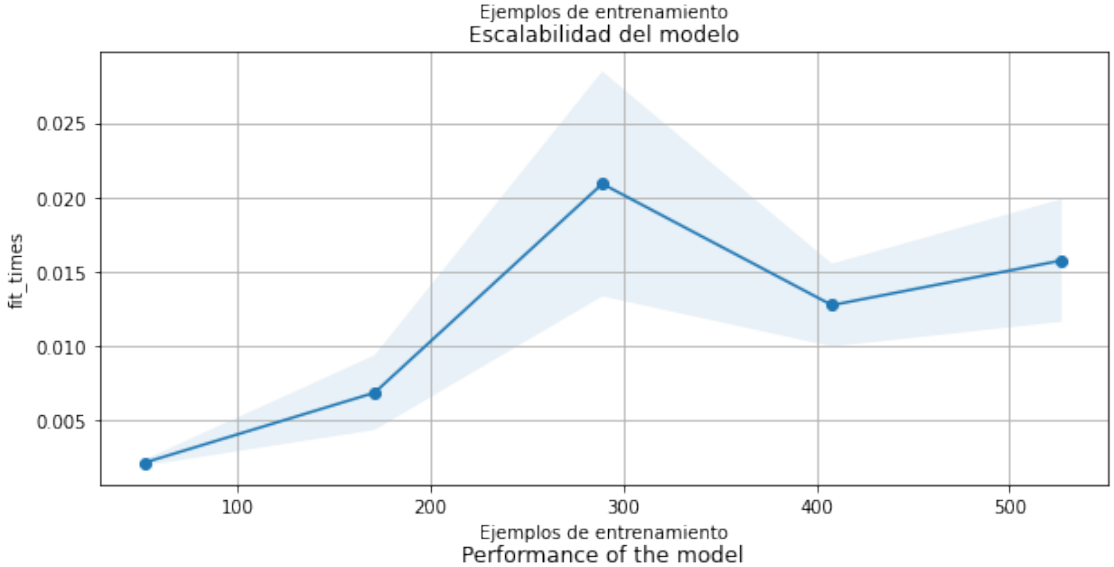
Finalmente el mejor estimador es

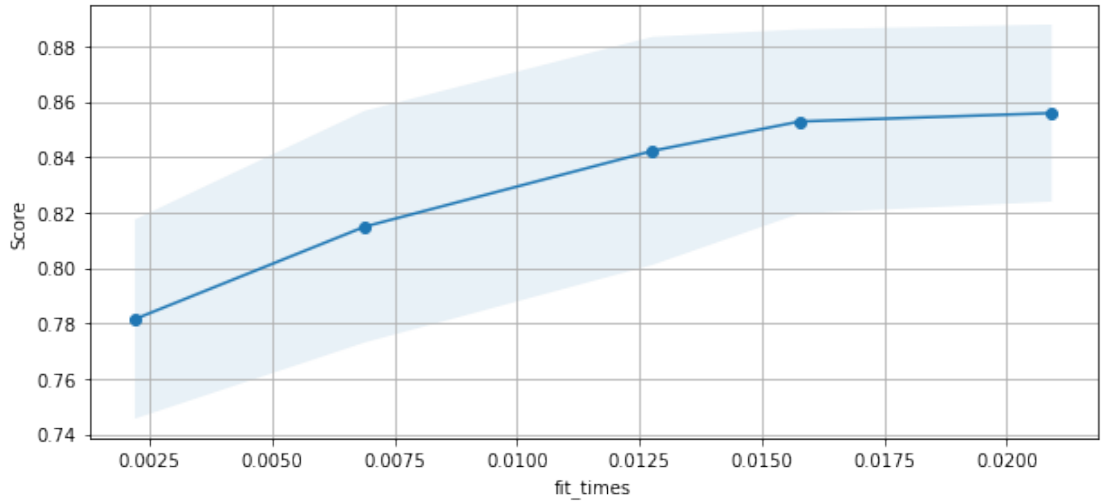
```
 LinearSVC(C=1, intercept_scaling=0.1, loss='hinge',
           max_iter=100000, random_state=0, tol=1e-05))])
```

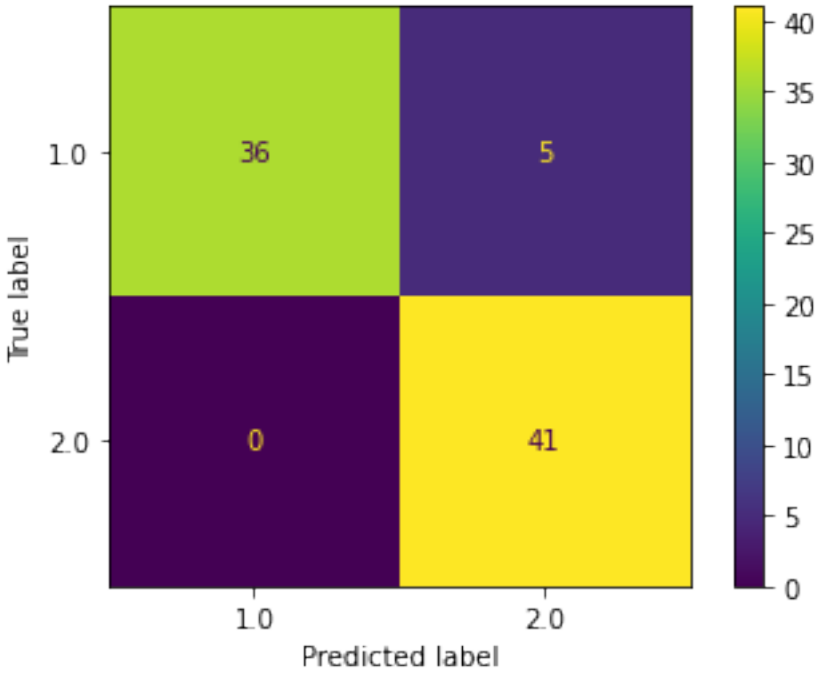
para un Window Size de 1240 y un Overlaping Fraction de 0.25, considerando todo el dataset.

b) Reporte los resultados obtenidos y seleccione el set completo de configuraciones que mejor resuelven, bajo su criterio, nuestro problema de clasificación









	precision	recall	f1-score	support
1.0	1.00	0.88	0.94	41
2.0	0.89	1.00	0.94	41
accuracy			0.94	82
macro avg	0.95	0.94	0.94	82
weighted avg	0.95	0.94	0.94	82