**1.** **Perceptrón**

El algoritmo de perceptrón es relativamente viejo (fue propuesto en 1958). Su objetivo es encontrar un hiperplano de separación. Es un algoritmo que procesa un dato de entrenamiento a la vez. Solo encuentra la solución si los datos son linealmente separables. Los parámetros de la función son los siguientes:

· ***penalty***, es el término de regularizacion, l1 lasso, l2 ridge, o elasticnet (calculado entre l1 y l2)

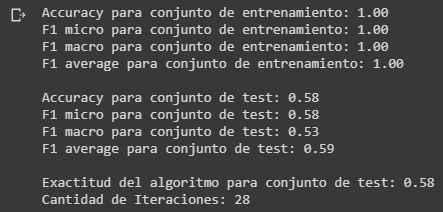
· ***alpha***, multiplica el término de regularizacion (penalty)

· ***max\_iter***, iteraciones máximas. Corresponde al número de veces que el algoritmo va a recorrer todos los puntos de entrenamiento

· ***tol***, Criterio de parada, las iteraciones terminan cuando (loss > previous\_loss - tol)

**1.1.Dataset Aborto No Balanceado**

La exploración manual de parámetros alcanza los mejores valores fijando ***penalty*** = 'l1', ***alpha*** = 0.0001, ***max\_iter*** = 1000 , ***tol*** = 0.001. De esta manera obtenemos las siguientes métricas:

****

Variando el valor de ***penalty***, manteniendo constante el resto de los parámetros, las métricas empeoran levemente en el conjunto de entrenamiento y apenas mejoran en el conjunto de test.

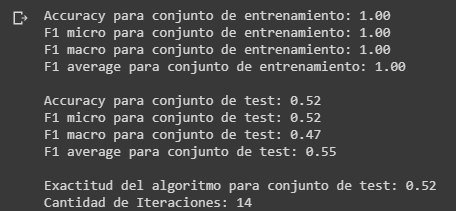
Cuando aumentamos el ***alpha*** (por ejemplo alpha=0.01), el valor de las métricas caen considerablemente.

La cantidad de iteraciones, a pesar de haber fijado ***max\_iter*** = 1000, fueron 28. Por lo tanto, variar este parámetro no afecta los resultados salvo que se fije por debajo de 28.

La variación del parámetro ***tol*** prácticamente no modifica las métricas.

**1.2.Dataset Aborto Balanceado**

La exploración manual de parámetros alcanza los mejores valores fijando ***penalty*** = ‘l2’, ***alpha*** = 0.0001, ***max\_iter*** = 1000 , ***tol*** = 0.001. De esta manera obtenemos las siguientes métricas:

****

Variando el valor de ***penalty***, manteniendo constante el resto de los parámetros, las métricas no cambian en el conjunto de entrenamiento y apenas empeoran en el conjunto de test.

Cuando aumentamos el ***alpha*** (por ejemplo alpha=0.01), el valor de las métricas caen considerablemente.

La cantidad de iteraciones, a pesar de haber fijado ***max\_iter*** = 1000, fueron 14. Por lo tanto, variar este parámetro no afecta los resultados salvo que se fije por debajo de 14.

La variación del parámetro ***tol*** prácticamente no modifica las métricas.

**2.** **K-NN**

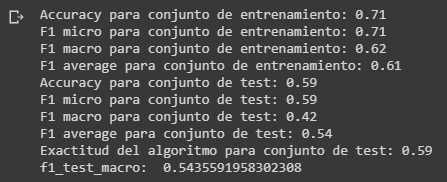
El modelo de clasificación de vecinos más cercanos pertenece a la familia de modelos no paramétricos. En este caso todo lo que el modelo conoce está en los datos de entrenamiento, por lo que podemos considerar como parámetros a estos datos.

El algoritmo define, en primer lugar, la medida de distancia en el espacio de características (***metric***). En segundo lugar define el número de vecinos que vamos a considerar para la clasificación (***n\_neighbors***).

Al ingresar un dato “nuevo”, el modelo calcula la distancia respecto a todos los puntos de entrenamiento ya etiquetados (computacionalmente esto constituye una desventaja en datasets muy grandes). Para tomar una decisión solamente va a considerar los K vecinos más cercanos que se hayan definido previamente. Finalmente el modelo va a clasificar a ese nuevo punto como la clase mayoritaria de sus vecinos más cercanos.

**2.1.Dataset Aborto No Balanceado**

La exploración manual de parámetros alcanza los mejores valores fijando ***n\_neighbors*** = 10 y ***metric*** = 'cosine'. De esta manera obtenemos las siguientes métricas:

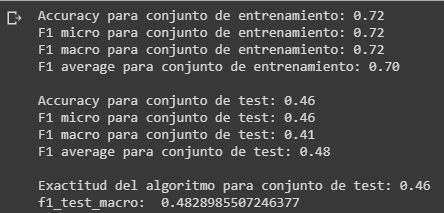


Manteniendo constante ***metric*** = 'cosine', a medida que aumentamos el valor de ***n\_neighbors*** (por ejemplo n\_neighbors = 20) las métricas empeoran en el conjunto de entrenamiento y aumentan en el conjunto de test.

Manteniendo constante ***n\_neighbors*** = 10, cuando fijamos ***metric*** = ‘euclidean’ los valores de las métricas caen tanto en el conjunto de entrenamiento como en el de test. Lo mismo ocurre cuando fijamos ***metric*** = ‘manhattan’.

**2.2.Dataset Aborto Balanceado**

La exploración manual de parámetros alcanza los mejores valores fijando ***n\_neighbors*** = 6 y ***metric*** = 'cosine'. De esta manera obtenemos las siguientes métricas:



Manteniendo constante ***metric*** = 'cosine', a medida que aumentamos el valor de ***n\_neighbors*** (por ejemplo n\_neighbors = 10) las métricas empeoran tanto en el conjunto de entrenamiento como en el conjunto de test.

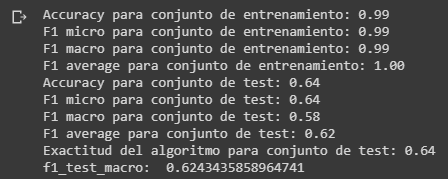
Manteniendo constante ***n\_neighbors*** = 6, cuando fijamos ***metric*** = ‘euclidean’ los valores de las métricas caen tanto en el conjunto de entrenamiento como en el de test. Lo mismo ocurre cuando fijamos ***metric*** = ‘manhattan’.

**3.** **Logistic Regression**

El modelo de regresión logística usa un enfoque probabilístico. Tiene como objetivo dar la probabilidad de que una instancia x sea de una clase y.

**3.1.Dataset Aborto No Balanceado**

La exploración manual de parámetros alcanza los mejores valores fijando ***penalty*** = 'l2', ***C*** = 1*,* ***max\_iter*** = 1000y ***tol*** = 0.5. De esta manera obtenemos las siguientes métricas:

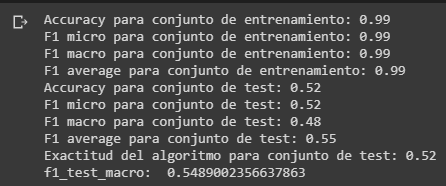


A medida que aumentamos el valor de ***tol*** (por ejemplo, tol=200), manteniendo el resto de los parámetros constantes, vemos que el valor de las métricas caen considerablemente en el conjunto de entrenamiento.

Otro aspecto que notamos es que el algoritmo solo funciona cuando ***penalty*** asume el valor ‘l2’.

**3.2.Dataset Aborto Balanceado**

La exploración manual de parámetros alcanza los mejores valores fijando ***penalty*** = 'l2', ***C*** = 0.4*,* ***max\_iter*** = 1000y ***tol*** = 0.5. De esta manera obtenemos las siguientes métricas:

****

A medida que aumentamos el valor de ***tol*** (por ejemplo, tol=200), manteniendo el resto de los parámetros constantes, vemos que el valor de las métricas caen considerablemente en el conjunto de entrenamiento.

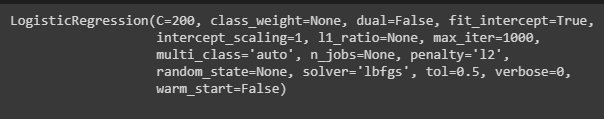
Otro aspecto que notamos es que el algoritmo solo funciona cuando ***penalty*** asume el valor ‘l2’.

**4.** **Mejor modelo**

En la exploración manual de parámetros vemos que el modelo de regresión logística es superior a los otros (perceptrón y K-NN) ya que se obtienen mejores métricas, sobre todo en el conjunto de test. Es importante mencionar que para el dataset no balanceado nos enfocamos en el valor del Accuracy, mientras que en el dataset balanceado usamos el F1\_macro.

Luego de realizar este procedimiento manual, para elegir los mejores parámetros del modelo nos basamos en el Grid Search, dado que generaliza mejor.

En el caso del dataset de Aborto No Balanceado, los mejores parámetros del modelo son los siguientes:



En el dataset de Aborto Balanceado los mejores parámetros que arroja el algoritmo son:

