

Universidad Nacional
de General Sarmiento



Laboratorio de Construcción de Software

TP Inicial - Entrega 3

Alumnos:

Perez Giannina - DNI: 43 729 769

Prieto Lucas - DNI: 43 626 494

2023

Introducción

En la etapa previa a esta, logramos entrenar nuestro modelo de Regresión Logística de tal forma que este alcanzó nuestro objetivo pretendido. Sin embargo, en esta sección nos centraremos en la optimización de nuestro modelo, para poder mejorar aún más su rendimiento y logre realizar predicciones más exactas. Para ello, realizaremos una configuración de parámetros y, por consiguiente, un reentrenamiento de nuestro modelo.

Configuración de parámetros

La configuración de parámetros en un modelo de Machine Learning se refiere a ajustar los valores de los diferentes parámetros que controlan el comportamiento y la eficacia del modelo. En el caso de un modelo de Regresión Logística, hay varios parámetros que se pueden configurar, como la tasa de aprendizaje, el número de iteraciones, el tipo de regularización, etc.

En nuestro modelo, utilizamos la biblioteca Scikit-learn para ajustar los parámetros necesarios, los cuales nos quedaron de la siguiente manera:

```
model = linear_model.LogisticRegression(max_iter=9000,  
C=1.0,penalty='l2', solver='lbfgs')
```

Máximo número de iteraciones (max_iter)

En algunos casos, la regresión logística puede requerir más iteraciones para converger. Si el valor predeterminado no es suficiente, se puede aumentar max_iter para permitir que el algoritmo converja.

Parámetro de regularización (C)

El parámetro C controla la fuerza de la regularización en la regresión logística. Un valor menor de C impone una mayor regularización, lo que puede ser útil para prevenir el sobreajuste, especialmente si existen muchas características. Por otro lado, un valor mayor reduce la regularización.

Tipo de regularización (penalty)

El parámetro penalty determina el tipo de regularización a aplicar, ya sea L1 (valor absoluto de los coeficientes) o L2 (norma euclidiana de los coeficientes). L2 tiende a ser más común y puede ayudar a evitar problemas de multicolinealidad.

Algoritmo de optimización (solver)

El parámetro solver determina el algoritmo de optimización utilizado para ajustar los coeficientes del modelo. lbfgs es adecuado para conjuntos de datos pequeños y se comporta bien en muchas situaciones.

Re-entrenamiento del modelo

Conjunto de datos de entrenamiento y de prueba

El primer paso que llevamos a cabo fue dividir nuestro conjunto de datos en dos partes. Por un lado, separamos el 70% de los datos para ajustar los parámetros del modelo y realizar el entrenamiento de este. Por otro lado, el 30% de los datos restantes los utilizamos a modo de prueba, con el fin de poder realizar una evaluación y saber si el rendimiento de nuestro modelo es bueno frente a datos desconocidos.

Balance de datos con smote

Para poder combatir a los datos desequilibrados, decidimos utilizar Smote. Smote es una técnica que se puede utilizar para el sobremuestreo de datos. Esta técnica crea nuevos ejemplos sintéticos en lugar de realizar un sobremuestreo por sustituciones.

Precisión del modelo

Existen diferentes métricas que se pueden implementar en un modelo para poder evaluar su rendimiento. Con respecto a nuestro entrenamiento, aplicamos la exactitud porque nos pareció la métrica más adecuada.

Ésta es la proporción de predicciones correctas respecto al total de predicciones realizadas. Se calcula como el número de predicciones correctas dividido entre el número total de predicciones.

Resultados

Precisión del modelo: 0.93

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.92	0.94	0.93	778
1	0.94	0.91	0.93	767
accuracy			0.93	1545
macro avg	0.93	0.93	0.93	1545
weighted avg	0.93	0.93	0.93	1545

Conclusiones

Nos pareció adecuado destacar la importancia de realizar un reentrenamiento de un modelo, pues nos permite experimentar diferentes posibilidades para aumentar en grandes escalas el rendimiento con respecto a las predicciones. Como se ve en los resultados, la precisión de nuestro modelo concluyó en una escala notoria.

A modo de cierre, podemos decir que fue interesante realizar la investigación en esta etapa, ya que nos brindó un buen conocimiento acerca del dinamismo que existe dentro de la Inteligencia Artificial y el Machine Learning.

Bibliografía

[¿Cuál es la diferencia entre parámetro e hiperparámetro? - Analytics Lane](#)

[Parámetros de entrenamiento - AWS](#)

[Cómo entrenar ML con Scikit Learn - OpenWebinars](#)

[Smote - Data Science](#)

[SMOTE for Imbalanced Classification with Python -](#)

[MachineLearningMastery.com](#)