

Análisis y reporte sobre el desempeño del modelo.



Curso:

**Inteligencia artificial avanzada
para la ciencia de datos I**

Presenta:

Flavio Ruvalcaba Leija

Fecha:

09-06-2023

Modelo utilizado:

Implementación de regresión logística con el uso de un framework.

Link:

<https://github.com/FlavioRr/InteligenciaArtificialGpo101/tree/main/MachineLearning1>

Puntos a considerar:

- **Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation).**

Con el fin de entrenar el modelo, decidí separar la base de datos `gpa_study_hours` en un 70%, un 15% y un 15%. Mi criterio para esta toma de decisión es el fin de reportar evaluaciones de mi modelo, explico:

Train:

El modelo de regresión logística se entrena en el conjunto de entrenamiento (**df_train**). Esto implica ajustar los parámetros del modelo utilizando estos datos, de manera que el modelo aprenda a hacer predicciones basadas en las características proporcionadas (en este caso, **gpa** y **study_hours (x_train)**) para predecir la variable objetivo (**graduated (y_train)**).

Test:

Una vez que el modelo ha sido entrenado, se evalúa en el conjunto de prueba (**df_test**). Esto proporciona una medida de qué tan bien el modelo se desempeña en datos que no ha visto durante el entrenamiento. La métrica de exactitud (Accuracy) se calcula para medir su rendimiento en este conjunto.

Validation:

Después de evaluar el modelo en el conjunto de test, se utiliza el conjunto de datos de validation (**df_validation**) para obtener una evaluación adicional. Esto puede ayudar a

verificar que el modelo no haya sobreajustado los datos de entrenamiento y que sea capaz de generalizar bien a datos nuevos y no vistos. La métrica de exactitud también se calcula en el conjunto de validación.

Conjunto de Prueba:	Conjunto de Validación:
<ul style="list-style-type: none">• True Positives (TP): 29• True Negatives (TN): 5• False Positives (FP): 1• False Negatives (FN): 0	<ul style="list-style-type: none">• True Positives (TP): 27• True Negatives (TN): 5• False Positives (FP): 2• False Negatives (FN): 2
Accuracy en test: 97.14%	Accuracy en validation: 88.89%

- **Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto**

El sesgo o bias se refiere a un tipo de error sistemático que ocurre cuando un modelo no es capaz de capturar con precisión la relación entre las características de entrada (variables independientes) y la variable de salida (variable dependiente) en un conjunto de datos.

El modelo que se ha presentado tiene un diagnóstico de sesgo medio ya que, con los datos anteriores, podemos interpretar que el conjunto de test, el sesgo es bajo ya que el modelo tiene un buen rendimiento en la detección de casos positivos y la minimización de falsos positivos, sin embargo, en el conjunto de validación, el sesgo es medio-alto debido a la presencia de falsos negativos y falsos positivos, lo que indica que el modelo no está realizando predicciones tan precisas en este conjunto en comparación con el conjunto de prueba.

Sabemos esto ya que en el conjunto de prueba, la ausencia de False Negatives (FN = 0) y un número bajo de False Positives (FP = 1) indican que el modelo tiene un buen rendimiento en la detección de casos positivos y en la minimización de errores de clasificación de casos negativos, lo que sugiere un sesgo bajo y por el otro lado, en el conjunto de validación, la presencia de False Negatives (FN = 2) y False Positives (FP = 2) indica que el modelo está cometiendo errores en la detección de casos positivos y en la clasificación de casos negativos, lo que sugiere un sesgo medio-alto.

- **Diagnóstico y explicación el grado de varianza: bajo medio alto**

La varianza se refiere a cuán sensible es el rendimiento del modelo a las pequeñas variaciones en los datos de entrenamiento.

En el conjunto de prueba, Accuracy es muy alto, sin embargo, en el conjunto de validation aunque la Accuracy es razonablemente alta (88.89%), la presencia de falsos negativos y falsos positivos en el conjunto de validación sugiere una variabilidad en el rendimiento del modelo en datos no vistos.

Esta variabilidad en las métricas entre los conjuntos de prueba y validación indica cierta varianza en el modelo en relación con los diferentes conjuntos de datos.

Por lo tanto, podemos definir que la varianza del modelo es media.

- **Diagnóstico y explicación el nivel de ajuste del modelo: underfitt fitt overfitt**

El nivel de ajuste del modelo se refiere a cuán bien un modelo de aprendizaje automático representa o se adapta a los datos en el conjunto de entrenamiento.

Utilizamos cross-validation para detectar el sobreajuste al evaluar el rendimiento del modelo en diferentes conjuntos de datos de prueba lo que proporciona una evaluación más robusta y confiable del modelo.

Finalmente, con los resultados obtenidos:

1. **Recall: 1.0 (100%):**

- Recall es la proporción de casos positivos que el modelo ha identificado correctamente con respecto al total de casos positivos. Un valor de 1.0 significa que el modelo ha capturado todos los casos positivos en el conjunto de test.

2. **Precision en test: 96.67%:**

- La precisión mide la proporción de casos clasificados como positivos por el modelo que son verdaderos positivos. Un valor del 96.67% indica que la mayoría de las predicciones positivas realizadas por el modelo son correctas.

3. **Accuracy en test: 97.14%:**

- La exactitud (accuracy) es la proporción total de predicciones correctas realizadas por el modelo en el conjunto de test. Un valor del 97.14% muestra un alto nivel de aciertos en general en el conjunto de prueba.

4. Accuracy en validation: 88.89%:

- La exactitud en el conjunto de validation es del 88.89%, lo que indica un buen rendimiento en datos de validación independientes.

5. Precisión promedio en cross-validation (4-fold): 0.96:

- La precisión promedio en validación cruzada es del 96% en promedio. Esto indica un buen rendimiento del modelo en diferentes divisiones del conjunto de entrenamiento durante cross validation.

En general, los resultados sugieren que el modelo tiene un buen ajuste a los datos de prueba y no muestra signos de underfitting u overfitting. Sin embargo, hay una diferencia en el rendimiento entre los conjuntos de prueba y validación, lo que indica cierto grado de variabilidad o sesgo en los datos de validación como establecimos anteriormente.

Basándote en lo encontrado en tu análisis utiliza técnicas de regularización o ajuste de parámetros para mejorar el desempeño de tu modelo y documenta en tu reporte cómo mejoró este.

La técnica de regularización utilizada fue por parte de la librería sklearn “regularización L2”. La regularización L2 agrega una penalización a la función de coste del modelo de regresión logística en función de la suma de los cuadrados de los coeficientes del modelo.

Al agregar la regularización L2 en scikit-learn, configuramos el parámetro **penalty** en "l2" al crear una instancia del modelo **LogisticRegression**.

Antes:

```
Accuracy en test: 97.14%
R² ajustado en test: 0.80
Accuracy en validation: 88.89%
R² ajustado en validation: 0.29
Precisión promedio en cross-validation (4-fold): 0.96 +/- 0.04
```

Después:

```
Accuracy en test: 97.14%  
R2 ajustado en test: 0.80  
Accuracy en validation: 94.44%  
R2 ajustado en validation: 0.65  
Precisión promedio en cross-validation (4-fold): 0.95 +/- 0.06
```

Como podemos observar, con el uso de esta técnica de regularización, el modelo aumento su accuracy de 88.8 % a 94.44% en el conjunto de validación, en cambio al resultado de cross validation, recibimos un mayor promedio.