Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Implementación de una técnica de aprendizaje máquina sin el uso de un framework

TC3006C. Inteligencia artificial avanzada para la ciencia de datos

Módulo 2. Aprendizaje Máquina

Grupo: 101

A01749075. Ameyalli Contreras Sánchez

Prof. Jorge Adolfo Ramírez Uresti

Fecha de entrega: 25 de agosto de 2025

Semestre agosto - diciembre 2025

Índice

1. Dataset empleado.....3

1.1. Datos de entrenamiento.....4

1.1.1. Dataset original.....4

1.1.2. Dataset con submuestreo.....4

1.1.3. Dataset con sobremuestreo.....5

1.2. Datos de prueba.....5

1.1.1. Dataset original.....5

1.1.2. Dataset con submuestreo.....5

1.1.3. Dataset con sobremuestreo.....6

2. Resultados del modelo.....6

2.1. Resultados dataset original.....6

2.2. Resultados dataset con submuestreo.....7

2.3. Resultados dataset con sobremuestreo.....8

3. Análisis y conclusiones.....10

1. Dataset empleado

De la plataforma de Kaggle se descargó el dataset ***Student Performance (Multiple linear regression)*** en el cual se presenta información sobre los hábitos de estudio de distintos estudiantes durante el periodo escolar, tales como horas de estudio previas al examen, sus notas en el examen previo, si son parte de actividades extracurriculares, horas de sueño, los documentos de práctica que revisaron y como variable objetivo su desempeño final.
Se optó por transformar el problema en uno de clasificación, es decir se transformó la variable objetivo en una variable binaria, donde calificaciones finales por debajo de 70 se considera como la clase *"Failed"* = 0, y arriba de esa nota se considera *"Passed"* = 1.

A continuación se presenta un extracto del dataset final empleado para el modelo y su información obtenida con el comando data.info():

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	1	9	1	1
1	4	82	0	4	2	0
2	8	51	1	7	2	0
3	5	52	1	5	2	0
4	7	75	0	8	5	0
...	...	...	...	...	...	...
9995	1	49	1	4	2	0
9996	7	64	1	8	5	0
9997	6	83	1	8	5	1
9998	9	97	1	7	0	1
9999	7	74	0	8	1	0

10000 rows × 6 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Hours Studied                   10000 non-null  int64
 1   Previous Scores                 10000 non-null  int64
 2   Extracurricular Activities      10000 non-null  int64
 3   Sleep Hours                     10000 non-null  int64
 4   Sample Question Papers Practiced  10000 non-null  int64
 5   Performance Index               10000 non-null  int64
dtypes: int64(6)
memory usage: 468.9 KB
```

```
Performance Index
0    7495
1    2505
Name: count, dtype: int64
```

Dado el desbalance de clases dentro del dataset, se realizaron 3 versiones de datasets para la realización de 3 modelos. La primera versión, con el dataset original, el cual cuenta con clases desbalanceadas. En segundo lugar, se realizó un submuestreo, reduciendo la cantidad de datos de la clase mayoritaria al mismo número que de la clase minoritaria, es decir, cada clase contendrá 2505 datos. Y finalmente, para la tercera versión se realizó un sobremuestreo, replicando y añadiendo datos a la clase minoritaria para tener la misma cantidad de datos que

la clase mayoritaria, es decir, cada clase terminará con 7495 datos. Esto con el objetivo de poder comparar los resultados obtenidos con 3 variaciones del mismo dataset.

El data split se realizó de forma manual y aleatoria. A continuación se muestra la información correspondiente a los datos de entrenamiento y de prueba para cada versión mencionada anteriormente.

1.1. Datos de entrenamiento

1.1.1. Dataset original

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	6	92	0	4	9	1
1	3	47	1	7	6	0
2	7	91	0	5	1	1
3	1	93	0	6	2	0
4	4	98	1	7	2	1
...	...	...	...	...	...	...
7995	7	56	0	9	5	0
7996	6	89	1	8	8	1
7997	4	67	1	6	2	0
7998	2	79	1	5	3	0
7999	4	48	0	7	4	0

8000 rows × 6 columns

1.1.2. Dataset con submuestreo

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	1	83	1	4	6	0
1	9	95	0	6	0	1
2	5	85	0	8	5	1
3	7	87	0	5	4	1
4	8	85	1	4	6	1
...	...	...	...	...	...	...
4003	9	75	0	7	4	1
4004	7	90	1	5	7	1
4005	8	83	0	9	3	1
4006	8	95	1	4	6	1
4007	5	86	1	5	1	1

4008 rows × 6 columns

1.1.3. Dataset con sobremuestreo

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	86	1	8	2	1
1	3	59	1	8	5	0
2	4	93	1	8	8	1
3	4	70	0	5	4	0
4	9	67	0	4	9	0
...	...	...	...	...	...	...
11987	1	67	0	9	3	0
11988	6	99	1	8	5	1
11989	7	43	1	4	3	0
11990	2	58	1	9	8	0
11991	8	89	0	7	8	1

11992 rows × 6 columns

1.2. Datos de prueba

1.1.1. Dataset original

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	9	73	1	6	5	0
1	1	52	0	9	1	0
2	6	85	0	4	0	0
3	3	84	1	8	5	0
4	7	81	0	6	8	1
...	...	...	...	...	...	...
1995	5	56	1	8	2	0
1996	1	85	1	6	9	0
1997	1	93	1	8	0	0
1998	8	59	1	4	7	0
1999	3	79	1	6	3	0

2000 rows × 6 columns

1.1.2. Dataset con submuestreo

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	9	67	1	8	8	0
1	1	45	0	7	2	0
2	6	67	0	8	3	0
3	1	97	1	8	0	1
4	9	89	1	7	9	1
...	...	...	...	...	...	...
997	7	58	1	6	4	0
998	8	75	0	8	0	0
999	9	89	0	7	2	1
1000	9	96	0	7	4	1
1001	9	78	1	5	1	1

1002 rows × 6 columns

1.1.3. Dataset con sobremuestreo

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	3	95	1	4	9	1
1	9	40	0	5	5	0
2	5	68	0	6	5	0
3	9	97	1	6	2	1
4	2	54	1	9	5	0
...	...	...	...	...	...	...
2993	4	57	0	4	3	0
2994	9	83	1	8	2	1
2995	5	53	0	5	8	0
2996	7	88	0	8	2	1
2997	1	53	1	6	1	0

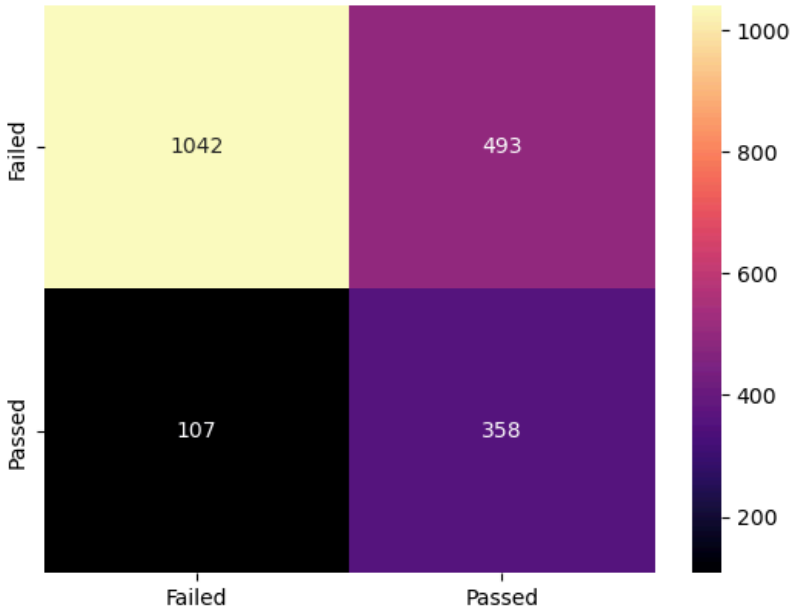
2998 rows × 6 columns

2. Resultados del modelo

En la presente sección se muestran los mejores resultados que se pudieron obtener con el modelo implementado tras mover en repetidas ocasiones los parámetros de learning rate y cantidad de épocas, éstos fueron evaluados utilizando matrices de confusión y reportes de clasificación, de los cuales se tomaron en cuenta diferentes métricas según la versión del dataset empleada. El objetivo principal fue balancear el nivel de error de cada clase, pero enfocándose principalmente en que haya menor cantidad de clasificaciones pertenecientes a la categoría “Passed” = 1 que sean colocadas por equivocación en la categoría “Failed” = 0.

2.1. Resultados dataset original

Matriz de confusión:



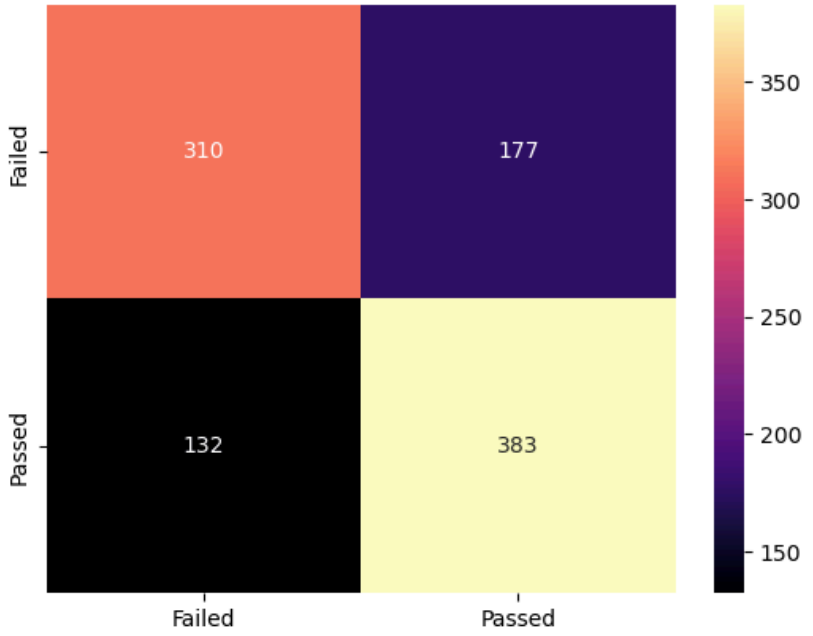
Reporte de clasificación:

	precision	recall	f1-score	support
0	0.91	0.68	0.78	1535
1	0.42	0.77	0.54	465
accuracy			0.70	2000
macro avg	0.66	0.72	0.66	2000
weighted avg	0.79	0.70	0.72	2000

Debido a que en esta versión de dataset, se está ocupando la versión más desbalanceada, es de esperar que los resultados obtenidos no fueran tan buenos, ya que se cuenta con una cantidad muy limitada de ejemplos de la clase "Passed" tanto para entrenamiento como para prueba. Como consecuencia de esto, se observa que al hacer el split de los datos, el conjunto de prueba no cuenta con un balance de las clases (1535 muestras para Failed, 465 muestras para Passed). Por ello, se tomará en cuenta la métrica de F1-Score, ya que es la que considera datasets muy desbalanceados, con lo que se observa que se obtuvo un desempeño de 0.78 para predicción de la clase "Failed", y un desempeño del 0.54 para la clase "Passed". Estos resultados naturalmente no son los que se esperarían idealmente, pero debido al fuerte desbalance, se puede considerar que tiene un desempeño medianamente aceptable. Adicionalmente, se consiguió que la clasificación de datos de Passed tuviera menor cantidad de errores que la clase Failed, esto, a pesar de que no se cuenta con un conjunto balanceado, se refleja en la métrica de recall donde Passed obtuvo un 77% de clasificaciones correctas, mientras que Failed obtuvo un 68%.

2.2. Resultados dataset con submuestreo

Matriz de confusión:



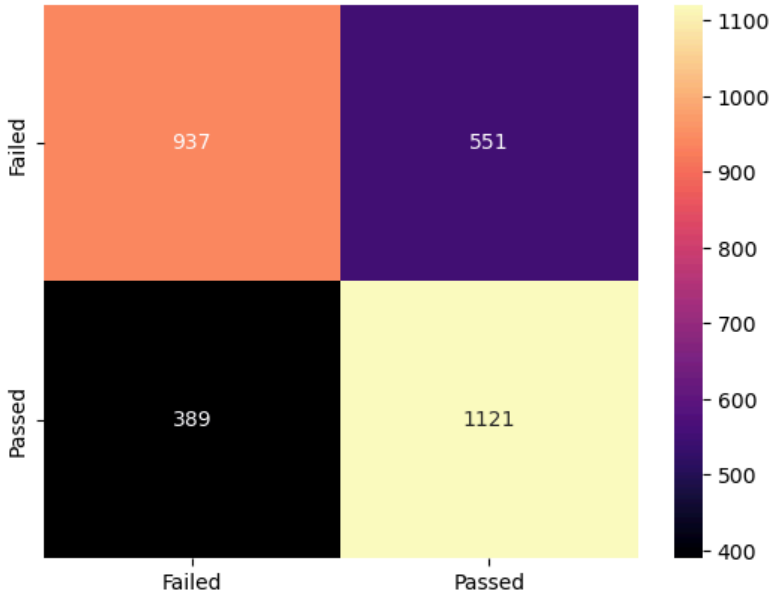
Reporte de clasificación:

	precision	recall	f1-score	support
0	0.70	0.64	0.67	487
1	0.68	0.74	0.71	515
accuracy			0.69	1002
macro avg	0.69	0.69	0.69	1002
weighted avg	0.69	0.69	0.69	1002

Para esta evaluación de desempeño, el dataset ya cuenta con balance en sus clases, por lo que hay la misma cantidad de datos para cada clase en el dataset original, lo que se refleja en la mejora del modelo, ya que esto facilita que tenga varios ejemplos de cada clase, en lugar de desde el principio tener más probabilidad de tomar muestras de una sola clase.
Debido a que los datos de prueba no cuentan con un desbalance muy fuerte (487 muestras de Failed, 515 muestras de Passed), se puede optar por utilizar como métrica de desempeño el accuracy sin tener dudas sobre su validez para este conjunto de datos. Por lo tanto, el accuracy del modelo utilizando el dataset balanceado con submuestreo es de 0.69, lo cual no es un valor ideal, ya que, aproximadamente, entre ¼ y ⅓ de los datos de cada clase los clasifica de manera incorrecta, sin embargo, con el modelo implementado fue el mejor resultado que se obtuvo variando los parámetros de épocas, learning rate y pesos iniciales.
Además, cumple con el objetivo mencionado al principio del presente reporte: que los datos pertenecientes a la clase Passed sean clasificados con menor error que los datos de la clase Failed, esto se puede observar al utilizar la métrica de Recall, la cual indica que los valores pertenecientes a la clase Passed fueron bien clasificados en un 74% mientras que los de la clase Failed en un 64%.

2.3. Resultados dataset con sobremuestreo

Matriz de confusión:



Reporte de clasificación:

	precision	recall	f1-score	support
0	0.71	0.63	0.67	1488
1	0.67	0.74	0.70	1510
accuracy			0.69	2998
macro avg	0.69	0.69	0.69	2998
weighted avg	0.69	0.69	0.69	2998

Dentro de esta evaluación de desempeño, también se contó con un dataset balanceado, pero en esta ocasión con sobremuestreo. Sin embargo, al igual que el conjunto de prueba obtenido del dataset con submuestreo, éste conjunto también se encuentra bastante balanceado, gracias a que tiene la misma probabilidad al seleccionar valores de cada clase (1488 muestras de Failed, 1510 muestras de Passed).

Por consiguiente se puede emplear de manera confiable la métrica de accuracy = 0.69, la cual es igual al del modelo con submuestreo, lo que en cierta forma muestra una consistencia en los resultados del modelo. Sin embargo, como se mencionó anteriormente, lo ideal sería que el valor de accuracy fuera mucho más alto, ya que solo clasificó correctamente alrededor de ¼ y ⅓ de los datos. Sin embargo, tras probar múltiples combinaciones de parámetros, este fue el mejor resultado obtenido que además cumple con el objetivo inicialmente planteado. Esto, se puede observar, con ayuda de la métrica de recall, la cual muestra que los datos de la categoría Passed fueron bien clasificados en un 74% (igual que en el modelo con submuestreo), mientras que los de la categoría Failed fueron bien clasificados en un 63% (ligeramente menor que con el submuestreo).

3. Análisis y conclusiones

Ninguno de los modelos presentados mostró tener un desempeño sobresaliente por sobre los demás. Sin embargo, el modelo que resultó ser ligeramente mejor que los otros fue aquel en el que se empleó el dataset con submuestreo. Esto debido a que se conservan datos previamente existentes en la base de datos, solo se eliminan datos de la clase mayoritaria, aunque esto conlleva una pérdida de información que podría ser valiosa.

Mientras que, para el dataset en el que se usó el sobremuestreo, es importante resaltar que se duplicaron datos ya existentes con el objetivo de tener muchos más datos de la clase minoritaria, pero eso pudo generar una incapacidad para reconocer patrones nuevos, ya que memoriza las instancias existentes, lo que lleva rápidamente a un sobreajuste, por lo que ese modelo deja de ser confiable para fines de evaluación de desempeño.

A manera de conclusión, se sugiere hacer experimentación de parámetros y de seccionamiento de dataset más exhaustiva. Otra alternativa que queda pendiente por probar es aplicar sub y sobremuestreo al dataset, es decir un muestreo híbrido, de forma que no sea exagerada la cantidad de datos que se duplican de la clase minoritaria, pero que tampoco se reduzca la clase mayoritaria tan drásticamente, buscando un equilibrio entre ambos métodos para buscar un mejor desempeño y a la vez disminuir el riesgo de perder demasiada información o introducir ruido o sobreajuste. Además, se puede buscar implementar otra metodología de sobremuestreo como SMOTE para evitar el overfitting y que en vez de duplicados, se generen datos sintéticos y ligeramente distintos de las instancias existentes.