Luis Alejandro Azúa Urrutia
21/02/2026

Reporte de resultados:

Modelo de Regresión Lineal para identificar enfermedades cardiovasculares.

Presentado por:

Luis Alejandro Azua Urrutia

Informe de hallazgos

Objetivos del reporte:

Presentar resultados de experimentación y ajustes del Modelo de Regresión Lineal utilizando varias técnicas de regularización, balanceo, ajustes de parámetros e hiperparámetros, justificando hallazgos con resultados numéricos obtenidos con métricas de precisión, accuracy, f1-score y recall.

Presentar el mejor modelo obtenido para futura implementación en el proyecto. Con esta información se busca mejorar y optimizar las campañas de salud pública.

Conjunto de datos y variables:

El conjunto de datos “heart_2020_cleaned.csv” es una base de datos del Censo Nacional de Población y Vivienda 2020 del INEGI. Consta de 3119795 (filas) y 18 variables (columnas) relacionadas con el estilo de vida y salud de la población. En esta se muestra “HeartDisease” (variable objetivo) y otros factores que podrían tener correlación con su presencia.

Luis Alejandro Azúa Urrutia
21/02/2026

Principales hallazgos desglosados:

Las **métricas** utilizadas a para **evaluar** cada uno de los modelos son las siguientes:

Accuracy (Exactitud): Porcentaje total de respuestas correctas. Nos dice que tan seguido acierta el modelo en sus predicciones, pero por si solo podría darnos una idea errónea. Si nuestros datos no están balanceados, como es nuestro caso, el modelo podría acertar casi siempre si dijera que nadie está enfermo.

Recall (Sensibilidad: Con esto medimos el porcentaje de casos de gente enferma que el modelo detecto. Es por esto que esta métrica es la más importante para nuestro modelo.

Precision (Precisión): Mide cuantos casos que denomino "en riesgo" realmente padecen la enfermedad. Es mejor tener un numero bajo de precisión, pero un recall más alto en este caso, pues enviar personas a revisiones preventivas es mejor opción que darle información falsa.

F1-score: Promedio de Recall y Precision. Nos permite definir si el modelo está bien equilibrado.

Modelo logístico simple:

Para comenzar la experimentación, procesamos los datos para que al modelo le fuera más sencillo leerlos. Utilizando los datos procesados con anterioridad, entrenamos un modelo de Regresión logística. Utilizándolo como base obtuvimos resultados que nos ayudaron a entender los ajustes y métodos que tendríamos que implementar más adelante.

Los resultados obtenidos fueron los siguientes:

```
---Evaluación conjunto de entrenamiento---
              precision    recall  f1-score   support

           0       0.92      0.99      0.96    234055
           1       0.55      0.11      0.18     21781

    accuracy                           0.92    255836
   macro avg       0.74      0.55      0.57    255836
weighted avg       0.89      0.92      0.89    255836

---Evaluación conjunto de prueba---
              precision    recall  f1-score   support

           0       0.92      0.99      0.95     58367
           1       0.54      0.10      0.17      5592

    accuracy                           0.91     63959
   macro avg       0.73      0.55      0.56     63959
weighted avg       0.89      0.91      0.89     63959

---Accuracy score---
0.9138666958520302
```

Como se puede observar, el modelo logístico base mostro una exactitud de 91.3%, lo que a simple vista parece ser muy bueno, sin embargo, mirando también su recall de prueba, el .10 nos está diciendo que no es capaz de detectar 90 de cada 100 personas con riesgo de enfermedad. Siendo nuestro objetivo principal identificar a las personas enfermas, los datos están muy alejados de la funcionalidad que se espera del modelo. Esto se debe principalmente a que los datos con los que ha sido entrenado presentan un nivel muy alto de desbalance, pues la mayoría de los encuestados no presentan enfermedades cardiacas. Es precisamente este desbalance el que buscamos combatir en los siguientes experimentos, para obtener los resultados deseados.

Luis Alejandro Azúa Urrutia
21/02/2026

Modelo logístico + Regularization:

Nuevamente utilizamos los mismos datos procesados previamente, los conjuntos de entrenamiento y prueba son los mismos, y es un modelo de regresión logística. Donde realizamos el cambio fue en la calibración de hiperparámetros, un proceso que se realiza antes de entrenar al modelo con los datos. En este caso definimos C con varios valores (1, 0.5, 0.1, 0.01, 0.001, 0.0001) y usamos el esquema KFold, buscando optimizar la métrica f1-score. El mejor valor resulto ser 1, por lo que el modelo no necesita ser más regularizado.

Dicho esto, a pesar de probar con todos los valores y llegar al mejor (C=1), el modelo con regularización alcanzo un F1-score de .18 demostrándonos que ajustar los hiperparámetros no será suficiente para compensar el desbalance de los datos originales.

```
---Evaluación conjunto de entrenamiento---
              precision    recall  f1-score   support

           0       0.92      0.99      0.96    234055
           1       0.55      0.11      0.18     21781

    accuracy                           0.92    255836
   macro avg       0.74      0.55      0.57    255836
weighted avg       0.89      0.92      0.89    255836

---Evaluación conjunto de prueba---
              precision    recall  f1-score   support

           0       0.92      0.99      0.95     58367
           1       0.54      0.10      0.17      5592

    accuracy                           0.91     63959
   macro avg       0.73      0.55      0.56     63959
weighted avg       0.89      0.91      0.89     63959

---Accuracy score---
0.9138666958520302
```

LogisticRegression

▼ Parameters

penalty	'l2'
dual	False
tol	0.0001
C	1
fit_intercept	True
intercept_scaling	1
class_weight	'balanced'
random_state	None
solver	'lbfgs'
max_iter	1000
multi_class	'deprecated'
verbose	0
warm_start	False
n_jobs	None
l1_ratio	None

Modelo logístico + Balanceo:

Posteriormente retomamos el experimento desde el procesamiento de datos. En lugar de utilizar GridSearch, esta vez solo implementamos el valor que más funciono (C=1) y utilizamos una técnica de balanceo llamada Weighted Class Logistic Regression. Lo que hace es castigar al modelo con más severidad cuando falla en identificar a una persona enferma, dándole mucha más importancia a esta clase.

Los resultados de este balanceo son los más drásticos si los comparamos con el modelo básico y el de regularización que probamos anteriormente. Podemos apreciar que el Recall, nuestra métrica de mayor importancia si consideramos el contexto del proyecto, aumento hasta un .78, esto significa que, de cada 100

pacientes, el modelo fue capaz de detectar a 78 de ellos. Esta es la más grande e importante mejoría en todo el experimento.

```
output de: balancing_model_heart_disease.py
---Evaluación conjunto de entrenamiento balanceado con Weighted Class Logistic Regression---
              precision    recall  f1-score   support

           0       0.97      0.75      0.85    234055
           1       0.22      0.78      0.35     21781

    accuracy                           0.75    255836
   macro avg       0.60      0.76      0.60    255836
weighted avg       0.91      0.75      0.80    255836

---Evaluación conjunto de prueba balanceado---
              precision    recall  f1-score   support

           0       0.97      0.75      0.84     58367
           1       0.23      0.78      0.35      5592

    accuracy                           0.75     63959
   macro avg       0.60      0.76      0.60     63959
weighted avg       0.91      0.75      0.80     63959

---Accuracy score---
0.7485107647086414
```

Si revisamos las demás métricas podemos notar como la métrica de Accuracy disminuyo bastante comparada con el modelo anterior. Ahora nos muestra que su accuracy bajo a un 75%, cuando el modelo anterior estaba en un 91%. Es por esto que esta métrica por si sola nos puede dar una idea falsa. El modelo anterior tenía un nivel muy alto, pero era incapaz de encontrar pacientes con la enfermedad. En cambio, este modelo tiene un valor predictivo real que puede ser aplicado en este contexto.

Modelo logístico + Balanceo SMOTE:

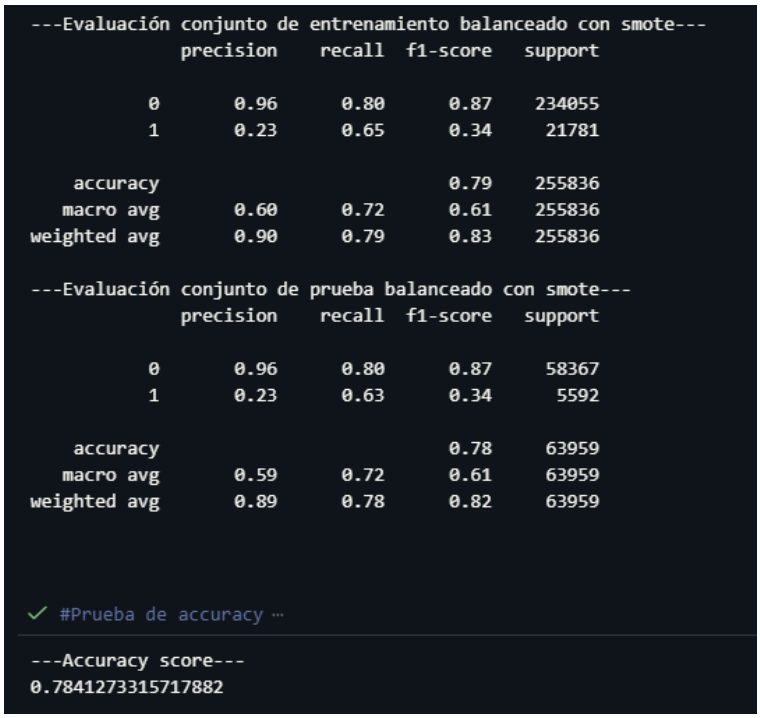
Para realizar otras pruebas, implementamos otro método de balanceo llamado SMOTE. Tuvimos que partir desde el procesamiento de datos, pues este método no es compatible con Weighted Class Logistic Regression. En realidad, ambas técnicas de balanceo quieren cumplir el mismo propósito, pero lo hacen de una manera diferente.

El método SMOTE es capaz de crear datos nuevos artificialmente a partir de datos reales. De esta forma es capaz de balancear los datos de personas enfermas y personas sanas igualando sus cantidades, rellenando la clase más pequeña con pacientes sintéticos.

A pesar de que el modelo puede usar datos artificiales para su entrenamiento, la evaluación del modelo sigue siendo real y honesta, pues los datos de prueba que se utilizan son completamente reales, con su desbalance original.

Para finalizar el estudio implementamos un análisis multivariado de correlación, utilizando un Heatmap, con el fin de identificar como cada variable es capaz de afectar al corazón, así como la interacción que se presenta entre ellas.

Luis Alejandro Azúa Urrutia
21/02/2026



```
---Evaluación conjunto de entrenamiento balanceado con smote---
              precision    recall  f1-score   support

           0       0.96      0.80      0.87    234055
           1       0.23      0.65      0.34     21781

    accuracy                           0.79    255836
   macro avg       0.60      0.72      0.61    255836
weighted avg       0.90      0.79      0.83    255836

---Evaluación conjunto de prueba balanceado con smote---
              precision    recall  f1-score   support

           0       0.96      0.80      0.87     58367
           1       0.23      0.63      0.34      5592

    accuracy                           0.78     63959
   macro avg       0.59      0.72      0.61     63959
weighted avg       0.89      0.78      0.82     63959
```

✓ #Prueba de accuracy …

```
---Accuracy score---
0.7841273315717882
```

En conclusión:

Los modelos de balanceo presentaron la mayor mejoría, apegándose no solo a buenos resultados numéricos, sino a un gran nivel de aplicabilidad al detectar casos reales de pacientes con enfermedad.

El modelo que presento mejores resultados en el proyecto fue el modelo logístico mas el balanceo con Weighted Class Logistic Regression.

El modelo es capaz de detectar al 78% de los pacientes en riesgo, mientras que su contraparte SMOTE detecta a un 63%. Considerando que el contexto implica pacientes que necesitan atención médica, esta diferencia presenta el más importante hallazgo para la implementación del modelo. Ambos modelos presentan una precisión casi idéntica de un .23, esto significa que ambos tienen la misma cantidad de falsas alarmas, en las que se manda a pacientes sanos a hacerse más análisis.

Por último, como el primer método resulto ser el más eficiente, podemos tener por seguro que la mejor manera de manejar el desbalance del dataset, es ajustar la importancia matemática que a generar casos artificiales.