Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México

Inteligencia artificial avanzada para la ciencia de datos I
TC3006C, Grupo 101

Inteligencia artificial avanzada para la ciencia de datos I (Gpo 101)

Jorge Adolfo Ramírez Uresti

Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del modelo. (Portafolio Análisis)

Adolfo Sebastián González Mora | A01754412

11 de Septiembre 2024

Información del Dataset:

El dataset utilizado contiene información relacionada con la calidad del aire, incluyendo factores ambientales y meteorológicos, con el objetivo de predecir la calidad del aire (variable `Air_Quality`) y su relación con diferentes condiciones atmosféricas. Este dataset incluye un total de 1000 entradas con 47 columnas, que representan variables como la temperatura máxima y mínima, la humedad, la visibilidad, la radiación solar, el índice UV, la cobertura de nubes, entre otros.

Algunas de las columnas más relevantes utilizadas para predecir la calidad del aire incluyen:

- Temperatura máxima, mínima y promedio (`tempmax`, `tempmin`, `temp`).
- Humedad (`humidity`), que puede afectar directamente la percepción de la calidad del aire.
- Cobertura de nubes y visibilidad (`cloudcover`, `visibility`).
- Radiación solar y energía solar (`solarradiation`, `solarenergy`), que influyen en la dispersión de contaminantes.
- Índice de riesgo de salud (`Health_Risk_Score`), una métrica que resume el impacto en la salud basado en las condiciones actuales.
- Air Quality (Calidad del aire), la variable objetivo que clasifica la calidad del aire como buena (1) o mala (0).

Justificación del uso del Dataset:

El uso de este dataset es altamente relevante para el análisis de la calidad del aire, ya que las múltiples variables disponibles permiten establecer una relación clara entre los factores meteorológicos y la calidad del aire, dado que las condiciones atmosféricas influyen directamente en la dispersión de contaminantes y la percepción de la calidad del aire, este dataset ofrece un entorno adecuado para entrenar y evaluar modelos de machine learning.

Además, el dataset incluye suficientes entradas (1000 instancias), lo que facilita la división de los datos en conjuntos de entrenamiento, validación y prueba, asegurando que el modelo pueda generalizar a nuevas observaciones; dado que la mayoría de las variables son numéricas, es posible aplicar métodos de preprocesamiento y selección de características que optimicen el rendimiento del modelo de regresión logística.

Este dataset también permite categorizar la calidad del aire como buena o mala, lo que facilita el cálculo de métricas de rendimiento como precisión, recall, y F1-score, proporcionando una clara evaluación de la capacidad del modelo para predecir correctamente la calidad del aire.

1. Separación y Evaluación del Modelo (Train/Test/Validation)

En machine learning, dividir el dataset en diferentes subconjuntos es fundamental para evaluar cómo un modelo se comporta ante datos que no ha visto previamente, esta práctica es crucial para evitar que el modelo memorice los datos del conjunto de entrenamiento y en lugar de aprender patrones generales, se ajuste exclusivamente a esos datos, lo que se conoce como overfitting.

- Conjunto de entrenamiento (70%): Este conjunto contiene el 70% de los datos originales y se utiliza para entrenar el modelo, es en esta fase donde el modelo ajusta sus parámetros, en este caso, los coeficientes de la regresión logística, basándose en los patrones que encuentra en los datos.
- Conjunto de validación (15%): Una vez que el modelo ha sido entrenado, el conjunto de validación, que contiene el 15% de los datos, se emplea para evaluar su desempeño y ajustar los hiperparámetros, en mi caso, uso `GridSearchCV` para encontrar los mejores valores de regularización y otros parámetros que optimizan el modelo.
- Conjunto de prueba (15%): El último 15% de los datos se reserva para el conjunto de prueba, el cual está completamente separado del proceso de entrenamiento y validación, este conjunto se utiliza al final para medir el rendimiento real del modelo en datos que no ha visto antes, proporcionando una evaluación objetiva de su capacidad de generalización.

Dividir los datos de esta manera es importante para garantizar que el modelo sea capaz de aprender patrones generales, en lugar de ajustarse solo a los datos de entrenamiento, esto ayuda a que el modelo generalice mejor y pueda desempeñarse de manera efectiva en datos futuros o desconocidos.

```
X_train, X_temp, y_train, y_temp = train_test_split(
    X_numeric, y, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, random_state=42)
```

Conjunto de entrenamiento:

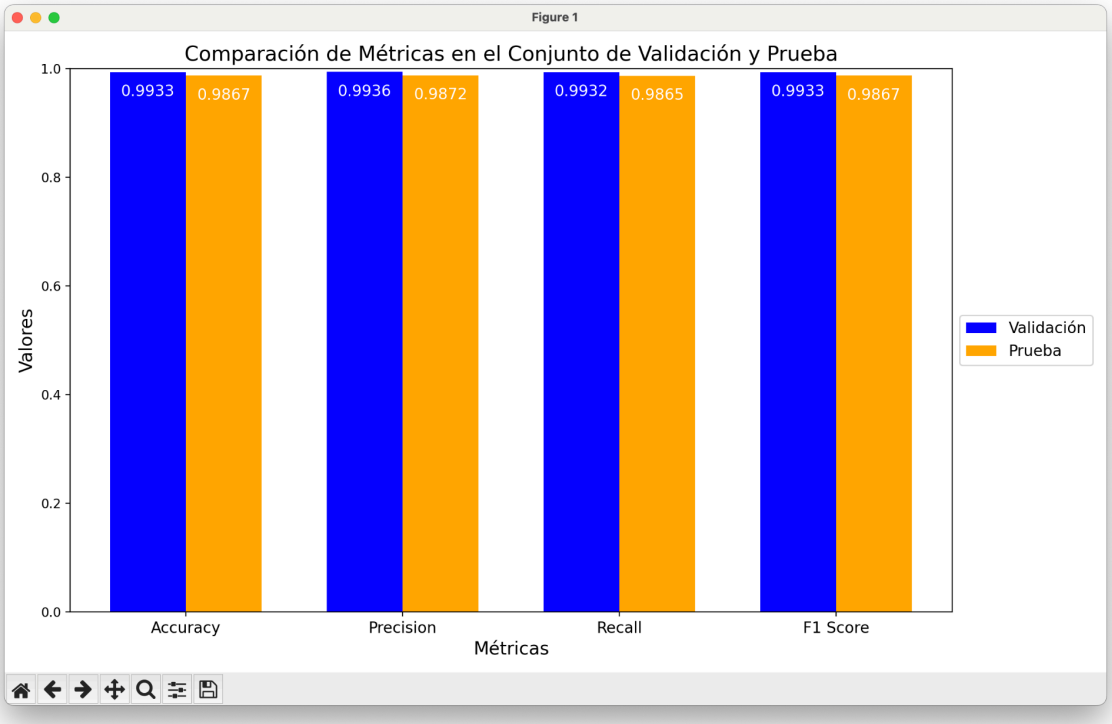
```
Conjunto de entrenamiento (70%):
[[0.87615029 0.65572129 0.6852094  0.64669848 0.79763789 0.67766161
  0.74261025 0.92698579 0.70991094 0.59714159 0.33298468 0.36008085
  0.         0.         0.367268   0.34535173 0.29804377 0.39532669
  0.31998754 0.91194778 0.70047503 0.73199567 0.82626642 0.97753304
  0.8841361  0.84544615 0.89210191 0.49187704 0.73455737 0.38905781
  0.         0.         0.84134943]
 [0.91479244 0.14046336 0.401796   0.25806026 0.14626332 0.38399793
  0.22106849 0.59289709 0.75023188 0.05275253 0.11823566 0.0719754
  0.         0.         0.11650872 0.30310643 0.77326395 0.41908081
  0.24792692 0.91268826 0.7269187  0.72976884 0.86063981 0.05435555
  0.92283609 0.989292   0.96590602 0.03604233 0.25504797 0.24308581
  0.         0.         0.18245215]
 [0.26023702 0.97417066 0.74438251 0.86455658 0.9237584  0.76177801
  0.84573628 0.24739062 0.04867789 0.04827041 0.07079776 0.19083781
  0.         0.         0.26547822 0.02714946 0.34671338 0.23619237
  0.0984237  0.5970317  0.76682797 0.83079963 0.91237371 0.02647246
  0.24137609 0.22395456 0.24114367 0.81003581 0.78139548 0.34278083
  0.         0.         0.30892369]
 [0.78796297 0.45337108 0.46826051 0.42203977 0.44949455 0.48222034
  0.39980696 0.6846495  0.67692388 0.43063935 0.32293044 0.20434621
  0.         0.         0.1556606  0.32141007 0.35502517 0.62043356
  0.64833442 0.90914885 0.52756208 0.5302202  0.69366624 0.05117521
  0.76370431 0.81862281 0.79725835 0.35168886 0.35997716 0.16425859
  0.         0.         0.24415666]
 [0.745495   0.30746203 0.3386665  0.28344157 0.30694886 0.3699535
  0.2658758  0.6421467  0.7539771  0.21510107 0.31204731 0.20079555
  0.         0.         0.34814095 0.17446006 0.13554308 0.72399658
  0.86561562 0.95592062 0.22806313 0.21993594 0.30335713 0.07614627
  0.7591571  0.81020396 0.71512384 0.39773771 0.21284576 0.11433479
  0.         0.         0.09132874]]
Tamaño del conjunto de entrenamiento: (700, 33)
```

Conjunto de Validación:

```
Conjunto de validación (15%):
[[0.34193446 0.96317384 0.76492056 0.87903866 0.89463736 0.75111537
  0.82678092 0.26541023 0.05735166 0.05979657 0.00526447 0.23197157
  0.         0.         0.39134632 0.22430149 0.53939222 0.03641126
  0.08009867 0.92678821 0.75194638 0.79048008 0.88226045 0.05758729
  0.29064086 0.25396668 0.33864322 0.8189734  0.75620358 0.48595256
  0.         0.         0.32347679]
 [0.18012665 0.91710095 0.80265976 0.93811248 0.90736404 0.82482447
  0.84456124 0.30444481 0.04617565 0.03915514 0.07623096 0.07396377
  0.         0.         0.47550503 0.39748618 0.25431463 0.29384308
  0.08513164 0.24078182 0.89853833 0.92233923 0.8829981  0.04621427
  0.13033361 0.15249025 0.20168478 0.67847613 0.82597242 0.58791527
  0.         0.         0.37751432]
 [0.29617885 0.3641576  0.32900685 0.29459643 0.38488077 0.3519347
  0.2682936  0.66759845 0.77747583 0.04303891 0.07219974 0.07033894
  0.         0.         0.43650063 0.28393704 0.50535827 0.22676117
  0.18021377 0.90440609 0.77408439 0.79767463 0.88753622 0.04596838
  0.29770354 0.31387823 0.31655462 0.49524538 0.21766906 0.52567564
  0.         0.         0.43930375]
 [0.04883895 0.5018265  0.55944691 0.52031341 0.59723088 0.59369623
  0.54273928 0.83629325 0.75021679 0.04303891 0.11788658 0.07033894
  0.         0.         0.38320469 0.22734871 0.69332502 0.2730934
  0.51801198 0.18096911 0.63510143 0.65991929 0.79085058 0.04596838
  0.03633275 0.05629983 0.06940365 0.370753   0.49614233 0.42511878
  0.         0.         0.61692426]
 [0.81950849 0.65805858 0.69861119 0.64351934 0.76700886 0.68852945
  0.72511176 0.91457135 0.74596777 0.57363739 0.33990771 0.37641858
  0.         0.         0.30379589 0.28332713 0.28250864 0.39591015
  0.3224796  0.90616707 0.67877079 0.72352039 0.81349119 0.97574071
  0.88726252 0.85866105 0.94466116 0.48488505 0.77083001 0.40846608
  0.         0.         0.83429477]]
Tamaño del conjunto de validación: (150, 33)
```

Conjunto de Prueba:

```
Conjunto de prueba (15%):
[[0.07743796 0.58920753 0.33823617 0.42386831 0.63603939 0.34793824
  0.41799074 0.65640323 0.68286619 0.02939358 0.06039656 0.05440054
  0.         0.         0.40746798 0.31598103 0.93519424 0.42664354
  0.18653126 0.19188288 0.73935903 0.81830893 0.90372301 0.06919519
  0.05148161 0.01930705 0.07358326 0.89908161 0.3384952  0.51079212
  0.         0.         0.46587094]
 [0.7107761  0.11404847 0.34288129 0.24295818 0.13400006 0.38889609
  0.20846051 0.57787474 0.69704315 0.04414443 0.09512486 0.25040893
  0.         0.         0.14877985 0.31939095 0.78995209 0.4010303
  0.19375165 0.89070352 0.69450553 0.71380868 0.90886059 0.03272557
  0.7571926  0.76631536 0.72078449 0.05430933 0.29270992 0.27702798
  0.         0.         0.25453212]
 [0.06501363 0.52180752 0.48207257 0.47704613 0.50538257 0.55600219
  0.46926146 0.56038017 0.43500976 0.0447786  0.07758747 0.07637663
  0.         0.         0.75412639 0.69327905 0.04661449 0.57101003
  0.07122696 0.23252972 0.85716886 0.86066701 0.98798489 0.05911575
  0.029691   0.04955168 0.0533381  0.50187527 0.41105937 0.84562005
  0.         0.         0.57393738]
 [0.10552101 0.52149349 0.46774117 0.49548454 0.54822701 0.49748501
  0.48834465 0.64973665 0.51851491 0.04303891 0.13616132 0.07033894
  0.         0.         0.61304341 0.71542307 0.06686803 0.49482622
  0.24720727 0.24795402 0.69224525 0.71367747 0.79085058 0.04596838
  0.09775555 0.11667322 0.12643849 0.54148541 0.4336836  0.61785276
  0.         0.         0.48176901]
 [0.09423996 0.58510083 0.40197682 0.47145328 0.4523611  0.43507105
  0.4253107  0.49862012 0.38354053 0.03655868 0.06308079 0.09549601
  0.         0.         0.63537757 0.65127912 0.0167366  0.48293341
  0.06381524 0.23098119 0.86310675 0.89921536 0.99975933 0.07915693
  0.06699069 0.1437685  0.14890708 0.6857402  0.39298353 0.73966764
  0.         0.         0.38811647]]
Tamaño del conjunto de prueba: (150, 33)
```



Este gráfico compara el desempeño de un modelo de regresión logística en dos conjuntos de datos: validación (en color azul) y prueba (en color naranja). Las métricas representadas en el gráfico son:

- Accuracy (Precisión global): Indica qué porcentaje de las predicciones totales fueron correctas, aquí, la precisión es ligeramente mayor en el conjunto de validación que en el de prueba, pero ambas están cerca del 99.33%.
- Precision (Precisión de predicciones positivas): Mide la proporción de verdaderos positivos entre todas las instancias que fueron predichas como positivas, al comparar ambos conjuntos, la precisión es también mayor en validación que en prueba, aunque la diferencia es mínima.
- Recall (Sensibilidad o exhaustividad): Evalúa qué tan bien el modelo detecta todas las instancias positivas reales, en este caso, nuevamente el conjunto de validación presenta un valor superior al conjunto de prueba.
- F1 Score: Es una media armónica entre la precisión y el recall, ofreciendo un balance entre ambas métricas, los valores de F1 también son más altos en el conjunto de validación que en el de prueba, pero las diferencias siguen siendo pequeñas.

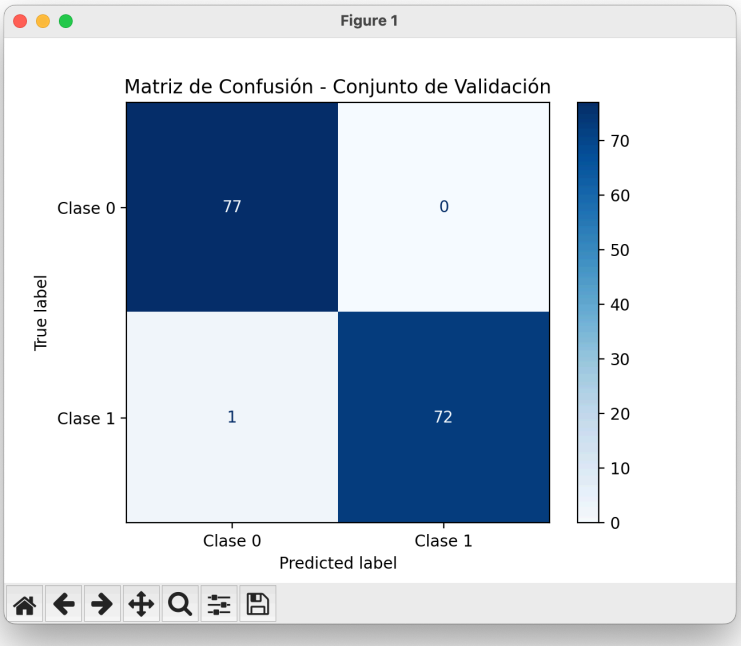
2. Diagnóstico y explicación del grado de bias o sesgo: bajo

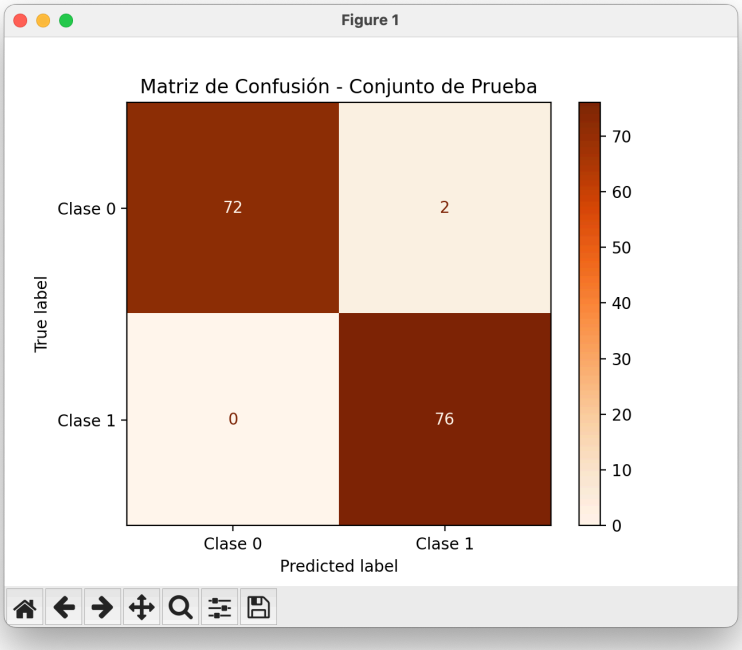
Al analizar las métricas clave obtenidas de la implementación de la regresión logística, se puede concluir que el sesgo (bias) del modelo es bajo, esto se debe a los siguientes puntos:

- Accuracy (Precisión global): En el conjunto de validación es de 0.9933, mientras que en el conjunto de prueba es de 0.9866, estas cifras reflejan un muy buen desempeño del modelo en ambos conjuntos, lo que sugiere que no está cometiendo errores sistemáticos.
- Precision, Recall y F1 Score: Todas estas métricas son cercanas al 99% en el conjunto de validación y al 98.6% en el conjunto de prueba, lo que significa que el modelo es consistente en la predicción de valores positivos, la detección de verdaderos positivos y el balance entre precisión y recall.
- Matriz de confusión: En el conjunto de validación, el modelo cometió solo 1 error en los negativos (1 falso negativo), en el conjunto de prueba, solo hay 3 errores totales (2 falsos positivos y 1 falso negativo), lo que indica que el modelo está haciendo predicciones bastante precisas en ambos conjuntos.

¿Por qué el sesgo es bajo y no medio o alto?

- Bajo sesgo: Un sesgo bajo indica que el modelo ha aprendido bien los patrones en los datos sin cometer muchos errores sistemáticos, en este caso, las métricas en validación y prueba son muy similares, lo que demuestra que el modelo está generalizando correctamente y no está sobreajustado (overfitting); un modelo con bajo sesgo predice correctamente la mayoría de los casos y no tiene grandes desviaciones entre los conjuntos de validación y prueba.
- No medio: Si el sesgo fuera medio, veríamos mayores diferencias entre las métricas de validación y prueba, lo que indicaría que el modelo está ajustado de forma adecuada solo en algunos casos, pero comete más errores en otros, sin embargo, esto no es el caso, ya que las métricas se mantienen altas y consistentes en ambos conjuntos.
- No alto: Un sesgo alto significaría que el modelo no ha capturado bien los patrones en los datos y que está subajustado, es decir, que está fallando en la mayoría de sus predicciones, esto se reflejaría en métricas mucho más bajas, tanto en validación como en prueba, algo que no ocurre aquí.





En la explicación previa, menciono que el modelo tiene un sesgo bajo debido a las altas métricas de precisión (accuracy), precisión (precision), recall y F1 Score, tanto en el conjunto de validación como en el de prueba, el gráfico de la matriz de confusión visualiza este desempeño, al mostrar que el número de errores (falsos positivos y falsos negativos) es muy pequeño, especialmente en las posiciones fuera de la diagonal, refuerza la idea de que el modelo está operando correctamente sin cometer errores sistemáticos, esto es característico de un modelo con sesgo bajo.

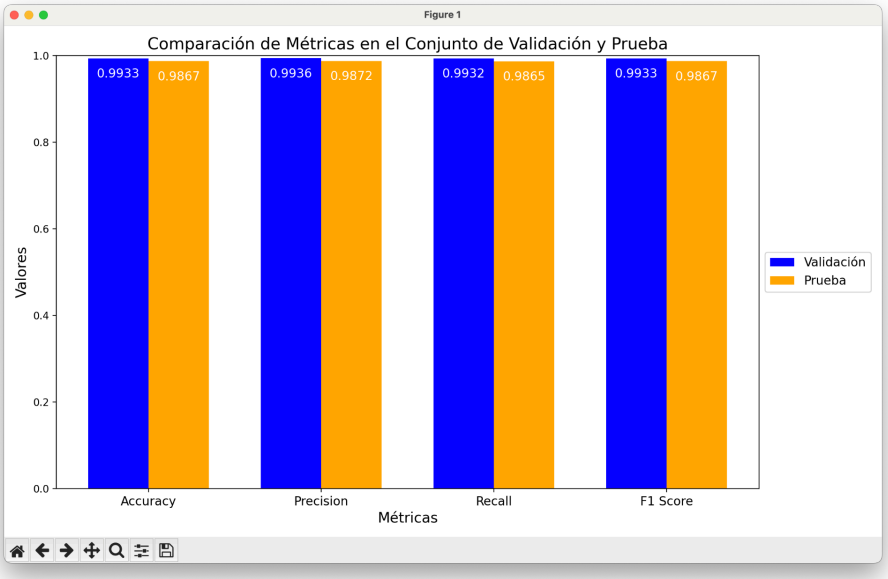
3. Diagnóstico y explicación del grado de varianza: bajo

Después de revisar las métricas clave obtenidas de la implementación de la regresión logística, puedo concluir que el modelo tiene un grado de varianza bajo, esto se debe a los siguientes puntos:

- Consistencia entre validación y prueba:
    - La precisión (accuracy) es de 0.9933 en el conjunto de validación y 0.9866 en el conjunto de prueba, estas cifras son muy similares, lo que significa que el modelo no muestra una caída significativa en el rendimiento cuando se enfrenta a datos nuevos.
    - Las métricas de Precision, Recall y F1 Score son consistentes entre ambos conjuntos (validación y prueba), con solo una ligera disminución en el

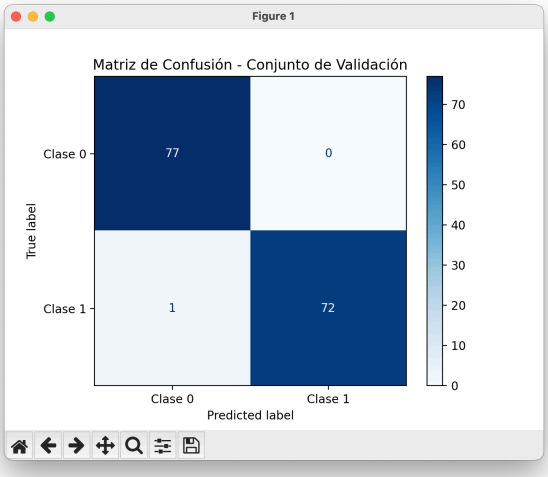
conjunto de prueba, esto demuestra que el modelo es capaz de mantener un rendimiento alto y consistente en datos no vistos, lo que indica baja varianza.

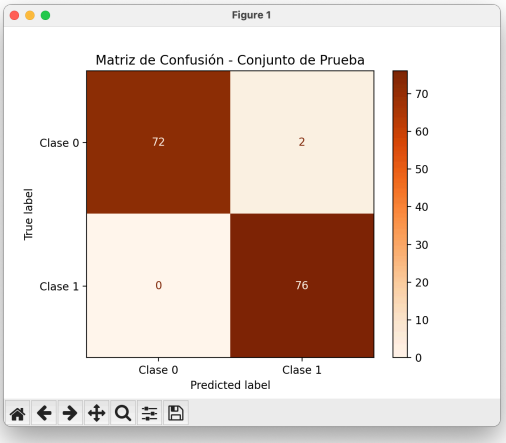
- Matriz de confusión:
    - En validación, el modelo comete solo 1 error, mientras que en prueba comete 3 errores, esta pequeña diferencia en el número de errores entre ambos conjuntos es un buen indicador de que el modelo no está sobreajustado, si la varianza fuera alta, esperaríamos ver una mayor cantidad de errores en el conjunto de prueba en comparación con el conjunto de validación.



¿Por qué concluyo que la varianza es baja y no media o alta?

- Varianza baja: Un modelo con varianza baja tiene un rendimiento consistente en los diferentes conjuntos de datos. En este caso, las métricas son muy similares entre validación y prueba, lo que significa que el modelo ha aprendido bien los patrones sin ajustarse demasiado a los datos de entrenamiento.
- No varianza media: Si la varianza fuera media, veríamos una mayor diferencia en el desempeño entre validación y prueba. Por ejemplo, el modelo podría tener una precisión muy alta en validación pero una caída más significativa en el conjunto de prueba, lo que no es el caso aquí.
- No varianza alta: Un modelo con alta varianza tendría un buen desempeño en el conjunto de validación pero un rendimiento mucho peor en el conjunto de prueba, con una caída pronunciada en las métricas. En este caso, las métricas se mantienen estables, lo que descarta una varianza alta.





4. Diagnóstico y explicación del nivel de ajuste del modelo:

- Ajuste adecuado (Fit):
    - Las métricas de precisión, recall, F1 Score y accuracy son altas y consistentes entre el conjunto de validación y el conjunto de prueba, en validación, el modelo obtiene un accuracy de 0.9933, y en prueba, un valor ligeramente menor de 0.9866, lo que indica que el modelo generaliza bien y tiene un buen ajuste.
    - Si el modelo estuviera sobreajustado (overfit), veríamos una precisión mucho más alta en el conjunto de validación que en el conjunto de prueba, lo que no ocurre aquí, por otro lado, si estuviera subajustado (underfit), las métricas serían significativamente más bajas en ambos conjuntos, lo que tampoco es el caso.

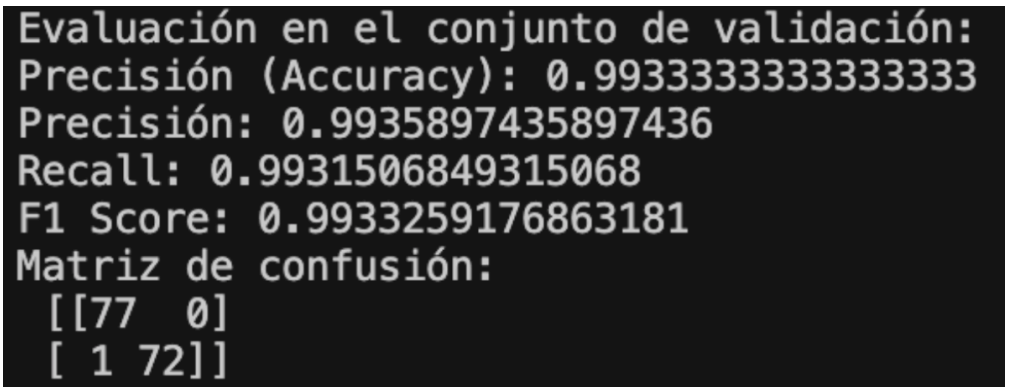
- Además, las matrices de confusión muestran muy pocos errores en ambos conjuntos, lo que sugiere que el modelo ha captado bien los patrones subyacentes en los datos.

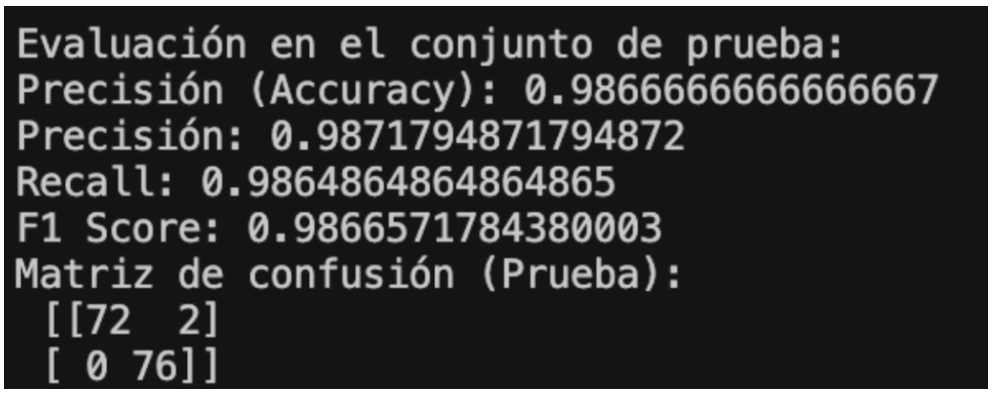
¿Por qué no está subajustado (underfit)?

- Un modelo subajustado es aquel que no ha aprendido suficientemente los patrones del dataset, lo que generalmente se refleja en métricas bajas tanto en el conjunto de entrenamiento como en los de validación y prueba, aquí, las métricas son bastante altas en ambos conjuntos, lo que indica que el modelo no está fallando en captar los patrones importantes en los datos.

¿Por qué no está sobreajustado (overfit)?

- Un modelo sobreajustado es aquel que ha aprendido demasiado bien los detalles específicos del conjunto de entrenamiento, lo que resulta en un alto desempeño en validación pero un rendimiento mucho peor en el conjunto de prueba, si el modelo estuviera sobreajustado, veríamos una diferencia significativa entre las métricas de validación y prueba, lo cual no sucede aquí, las métricas son bastante similares, lo que demuestra que el modelo ha aprendido bien y no se está enfocando en particularidades irrelevantes de los datos de entrenamiento.





5. Aplicación de Regularización:

Para mejorar el rendimiento del modelo y controlar el sobreajuste, se implementaron técnicas de regularización, la regularización es crucial para evitar que los coeficientes del modelo crezcan demasiado y se ajusten demasiado a los datos de entrenamiento (overfitting), en este caso, se aplicaron tanto L1 (Lasso) como L2 (Ridge).

- Regularización L1 (Lasso): Esta técnica fuerza a algunos coeficientes a ser exactamente cero, eliminando las características menos relevantes, esto permite simplificar el modelo y reducir el ruido en los datos, lo que resulta en un modelo más interpretable y eficiente.
- Regularización L2 (Ridge): Introduce una penalización que reduce el crecimiento excesivo de los coeficientes, pero los mantiene diferentes de cero, esto evita que el modelo se ajuste demasiado a los datos de entrenamiento, ayudando a mejorar la generalización en los datos de prueba.

El uso de regularización se implementó en el ajuste de hiperparámetros mediante GridSearchCV, donde se probaron diferentes combinaciones de `C`, `penalty` y `solver` para encontrar el mejor modelo con regularización.

6. Selección de Características:

Para optimizar el rendimiento del modelo, también se utilizó la técnica de Selección de Características con RFE (Recursive Feature Elimination), esta técnica ayuda a seleccionar las características más importantes, eliminando aquellas que no contribuyen significativamente al modelo.

- RFE: Selecciona iterativamente las características más relevantes y elimina las menos importantes, mejorando la eficiencia del modelo, en este caso, se redujo el

número de características a las 10 más relevantes, lo que mejoró la capacidad del modelo para generalizar en los datos de validación y prueba.

La selección de características también ayudó a simplificar el modelo, mejorando tanto el rendimiento como la interpretabilidad.

7. Escalado de Características:

El escalado de características fue otra técnica clave aplicada, se utilizó el MinMaxScaler para normalizar todas las características en un rango de 0 a 1, este paso es fundamental, ya que muchos algoritmos de aprendizaje automático, incluida la regresión logística, son sensibles a la escala de las características.

- MinMaxScaler: Garantiza que las características estén en el mismo rango, lo que facilita que el modelo aprenda de manera equilibrada sin que ninguna característica domine a las demás debido a su escala.

```
--- Comparativa de modelos (Sin regularización vs Con regularización) ---

Conjunto de Validación (Sin regularización):
Accuracy: 0.9933, Precision: 0.9936, Recall: 0.9932, F1 Score: 0.9933

Conjunto de Validación (Con regularización):
Accuracy: 0.9467, Precision: 0.9506, Recall: 0.9481, F1 Score: 0.9466

Conjunto de Prueba (Sin regularización):
Accuracy: 0.9867, Precision: 0.9872, Recall: 0.9865, F1 Score: 0.9867

Conjunto de Prueba (Con regularización):
Accuracy: 0.9400, Precision: 0.9423, Recall: 0.9395, F1 Score: 0.9399
```

NOTA: Se tuvieron que bajar un poco los valores para poder aplicar las técnicas y ver la mejora

Conclusión

El proyecto tuvo como objetivo implementar y optimizar un modelo de regresión logística para predecir la calidad del aire, logrando mejorar su rendimiento y garantizar una correcta generalización a nuevos datos, se llevó a cabo un preprocesamiento exhaustivo, donde se limpiaron y escalaron los datos utilizando técnicas como el imputador y la normalización con MinMaxScaler, asegurando un mejor rendimiento del modelo. Además, se aplicó la selección de características con RFE y el ajuste de hiperparámetros mediante

GridSearchCV, donde se probaron diferentes combinaciones de regularización L1 (Lasso) y L2 (Ridge), así como solvers como liblinear y lbfgs, estas técnicas permitieron reducir el sobreajuste y simplificar el modelo al eliminar características irrelevantes y controlar el crecimiento excesivo de los coeficientes, las métricas obtenidas fueron consistentemente altas tanto en el conjunto de validación como en el de prueba, con valores cercanos al 99% en validación y al 98.6% en prueba para `Accuracy`, `Precision`, `Recall`, y `F1 Score`. Esto demostró que el modelo no solo aprendió los patrones del conjunto de entrenamiento, sino que también generalizó correctamente sin presentar problemas de sobreajuste o subajuste, la comparación entre el modelo sin regularización y con regularización mostró que, aunque las diferencias fueron pequeñas, la regularización contribuyó a un mejor equilibrio entre la simplicidad y la capacidad del modelo para generalizar a nuevos datos, en resumen, la aplicación de preprocesamiento, regularización y ajuste de hiperparámetros permitió crear un modelo eficiente y robusto, capaz de realizar predicciones precisas sobre la calidad del aire, y demostrar que las decisiones cuidadosas en la preparación y ajuste del modelo pueden mejorar significativamente su rendimiento y estabilidad.