**Instituto Tecnológico y de Estudios Superiores de Monterrey**

***Campus Estado de México***

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

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

TC 3006

Grupo: 101

**Alumno:**

Pablo González de la Parra | A01745096

**Fecha de entrega: 11 de septiembre de 2023**

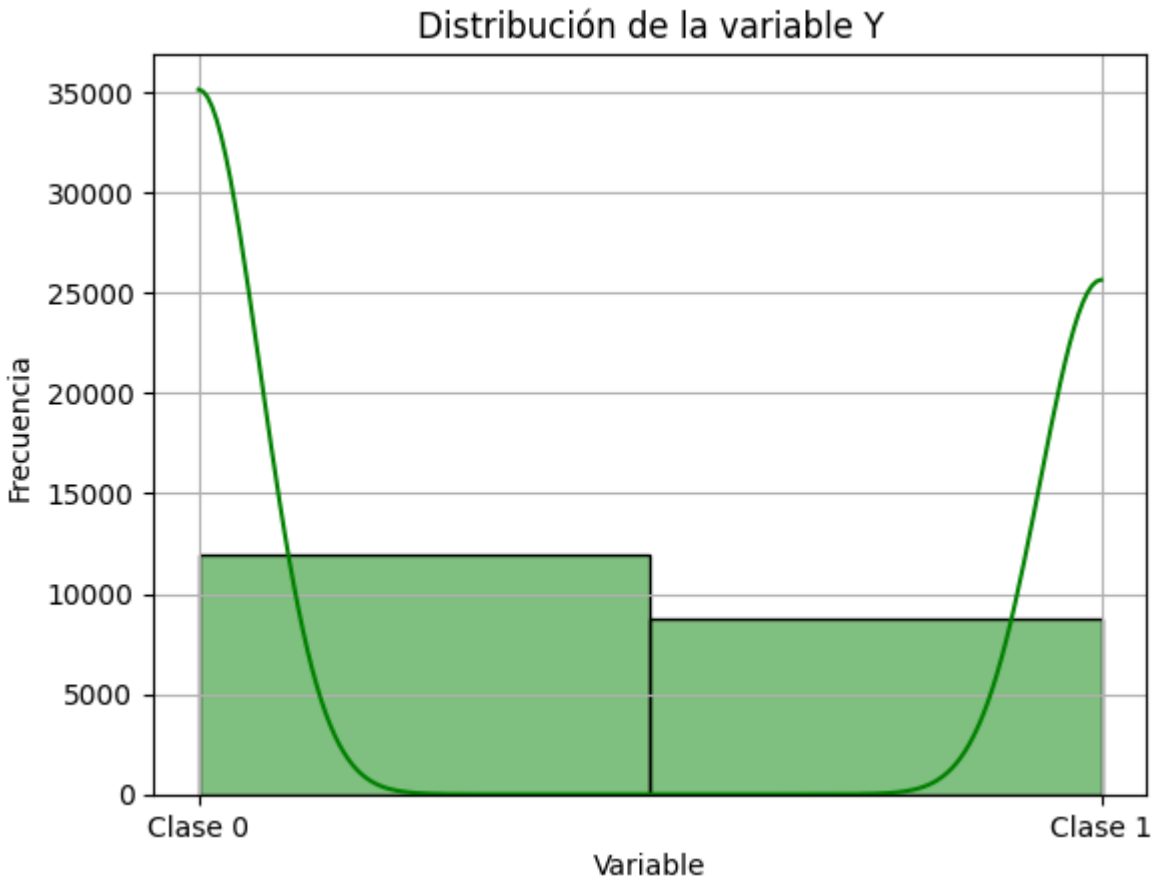
**Introducción**

De acuerdo con las 2 implementaciones que se realizaron en las entregas previas de este módulo, se genera un análisis sobre su desempeño en un set de datos. En este caso se eligió la segunda implementación del algoritmo de ML. Este algoritmo consiste en un modelo de regresión logística (Logistic Regression) construido con un framework / librería existente. En este caso se utilizó la librería de Scikit Learn. De acuerdo con las características de dicha implementación, a continuación se explica el proceso de análisis sobre sus parámetros y resultados.

**Justificación selección del dataset**

El dataset utilizado para la implementación de un algoritmo de Machine Learning (ML) utilizando un framework o librería especializada se eligió debido a diversas características que posee. A continuación se enlistan las características que se consideraron para la elección del dataset. Una de las primeras características que se tomó en cuenta fue tanto el tamaño como la representatividad de los datos. En este caso, el dataset de California Housing contiene aproximadamente 20,000 registros, lo cual es un número considerable de datos. Además, los datos son representativos de la población de California, ya que se obtuvieron de un censo oficial realizado en el año 1990. Por lo tanto, se consideró que el dataset es representativo de la población y que es lo suficientemente grande para entrenar un modelo de ML.

Otra de las características que se tomaron en cuenta fue la calidad de los datos. En este caso, los datos no contienen valores atípicos o ruido excesivo. Además, los datos no contienen valores nulos o faltantes. Por lo tanto, se consideró que los datos son de buena calidad y que no requieren de un proceso de limpieza exhaustivo. Una de las características más relevantes que se consideró antes de utilizar el dataset fue la distribución entre las clases de la variable objetivo. En este caso, de acuerdo al funcionamiento de este tipo de algoritmo de aprendizaje máquina, si una clase (en un problema de clasificación) tiene un mayor número de registros que las demás, el modelo tenderá a predecir esa clase con mayor frecuencia y afectar nuestras métricas de rendimiento. Por lo tanto, se consideró que la distribución entre las clases de la variable objetivo es balanceada, ya que la diferencia entre el número de registros de cada clase es mínima. A continuación se muestra la distribución de las clases de la variable objetivo.



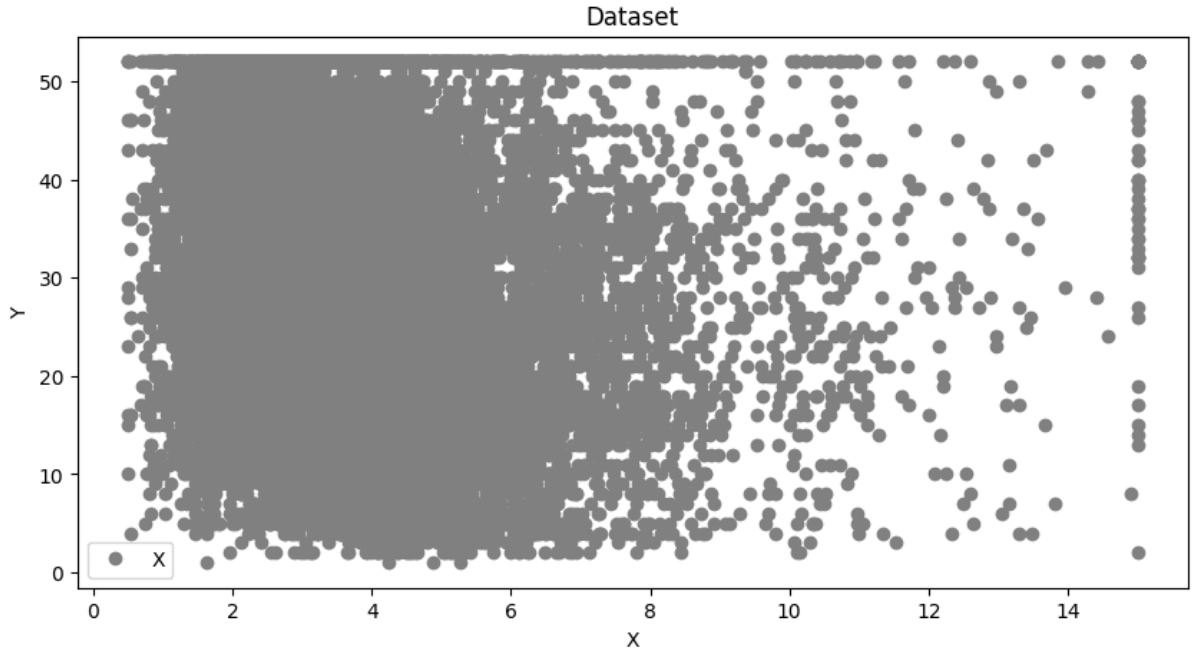

Finalmente debido a que el dataset elegido se encuentra diseñado para un problema de regresión, y no de clasificación, se tuvo que realizar una transformación de los datos para poder utilizarlos en un problema de clasificación. En este caso, se transformó la variable objetivo en una variable categórica, donde se asignó un valor de 1 a los registros que tenían un valor mayor a 2.0 y un valor de 0 a los registros que tenían un valor menor o igual a 2.0. Por lo tanto, se consideró que el dataset es adecuado para el problema de clasificación que se está abordando, ya que la distribución de las variables objetivo no está diseñada para que la precisión de un modelo sea "perfecta", demostrando que el modelo es capaz de generalizar a partir de los datos de entrenamiento a datos que no ha visto previamente.

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

De acuerdo con la implementación del algoritmo, se utilizó un set de datos de 20,640 registros. De estos registros, se separaron 13,209 (80% del 80%) para entrenamiento, 3,303 (20% del 80%) para validación, y 4,128 (20% del 100%) para testing. A continuación se muestra de manera visual la separación de los datos.

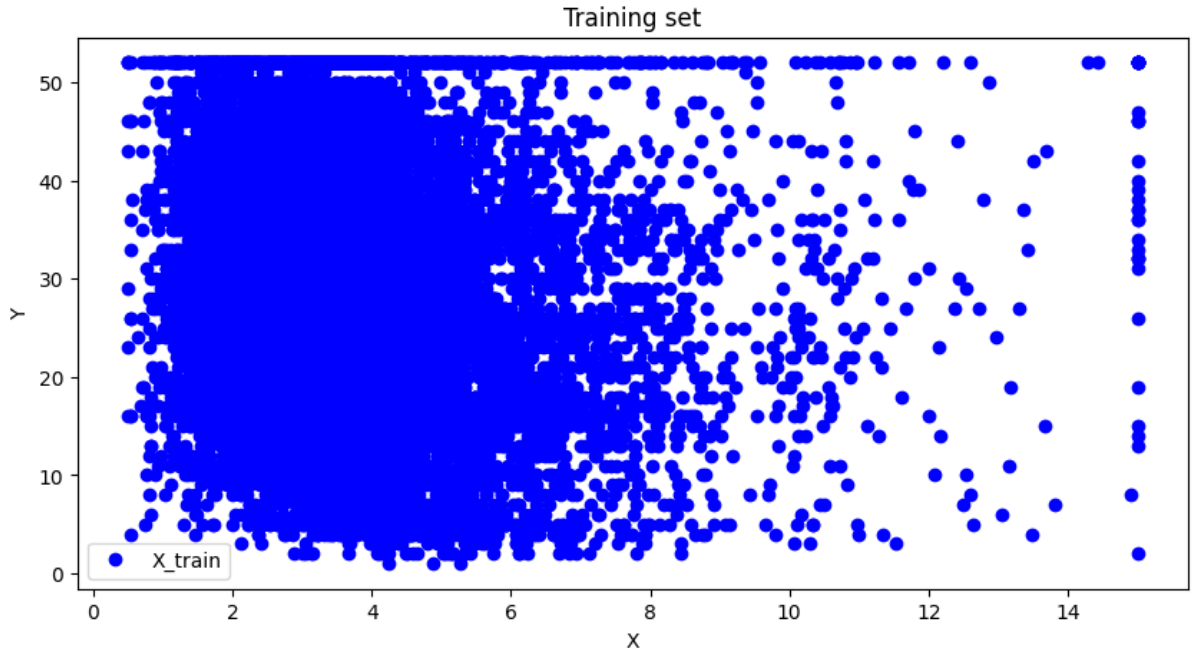
***Dataset***

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 4.7069 | 27.0 | 6.523256 | 1.116279 | 873.0 | 3.383721 | 38.00 | -120.97 |
| 1 | 3.8750 | 15.0 | 5.058406 | 1.075770 | 3359.0 | 2.651144 | 34.10 | -117.87 |
| 2 | 2.8828 | 26.0 | 5.290618 | 1.201373 | 1273.0 | 2.913043 | 33.76 | -117.85 |
| 3 | 5.0000 | 34.0 | 6.474708 | 1.136187 | 705.0 | 2.743191 | 34.35 | -119.74 |
| 4 | 5.0371 | 25.0 | 6.385656 | 1.008119 | 1857.0 | 2.512855 | 38.50 | -121.51 |



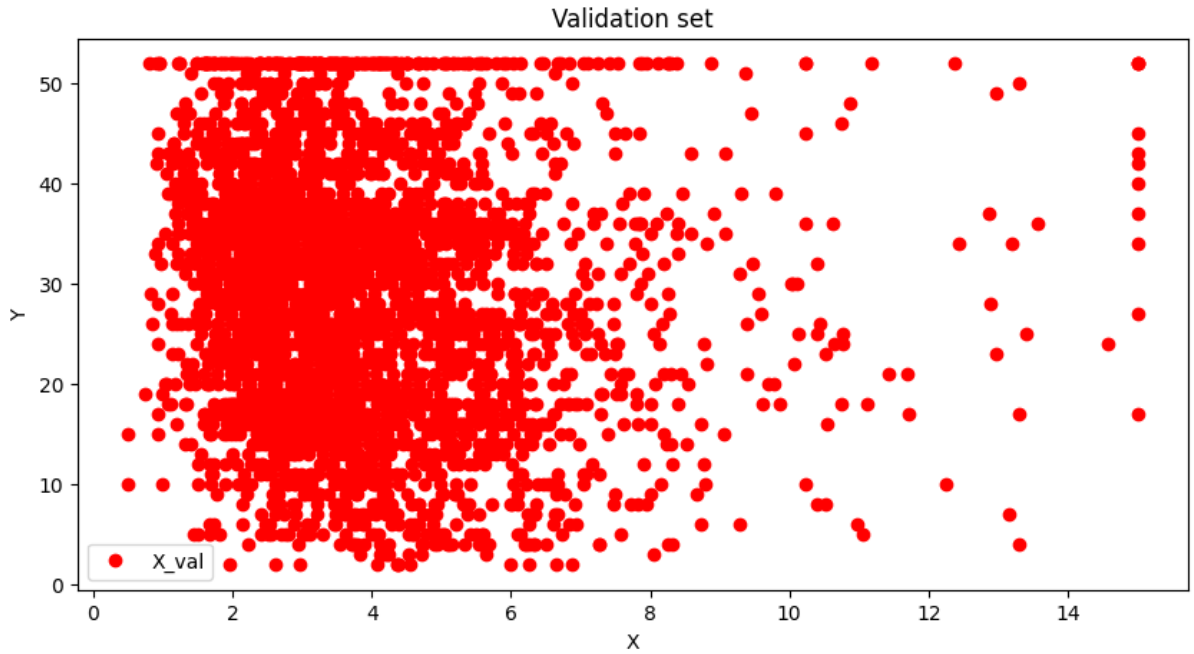

***Dataset de entrenamiento***

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 4.7069 | 27.0 | 6.523256 | 1.116279 | 873.0 | 3.383721 | 38.00 | -120.97 |
| 1 | 3.8750 | 15.0 | 5.058406 | 1.075770 | 3359.0 | 2.651144 | 34.10 | -117.87 |
| 2 | 2.8828 | 26.0 | 5.290618 | 1.201373 | 1273.0 | 2.913043 | 33.76 | -117.85 |
| 3 | 5.0000 | 34.0 | 6.474708 | 1.136187 | 705.0 | 2.743191 | 34.35 | -119.74 |
| 4 | 5.0371 | 25.0 | 6.385656 | 1.008119 | 1857.0 | 2.512855 | 38.50 | -121.51 |

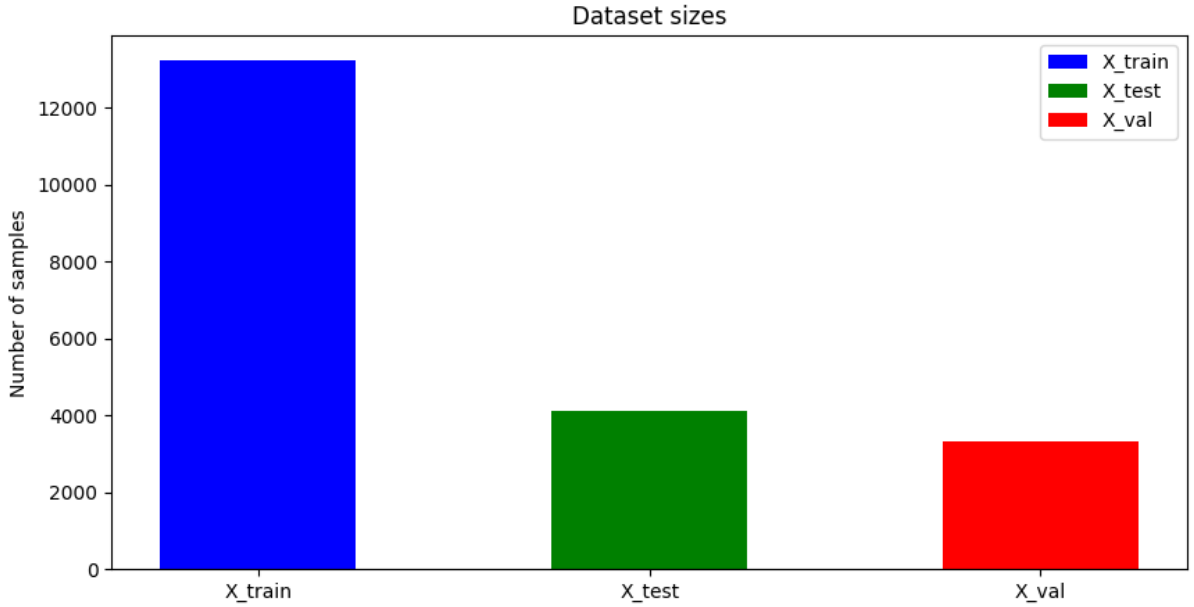



***Dataset de validación***

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 3.5375 | 35.0 | 6.543956 | 1.043956 | 537.0 | 2.950549 | 36.74 | -119.85 |
| 1 | 2.1726 | 23.0 | 4.287179 | 0.987179 | 1060.0 | 2.717949 | 41.02 | -124.16 |
| 2 | 4.2071 | 14.0 | 3.916929 | 1.056639 | 3148.0 | 1.981120 | 33.67 | -117.92 |
| 3 | 4.1727 | 10.0 | 4.101504 | 0.977444 | 517.0 | 1.943609 | 34.15 | -118.45 |
| 4 | 7.1148 | 28.0 | 6.829971 | 1.014409 | 832.0 | 2.397695 | 33.63 | -117.90 |

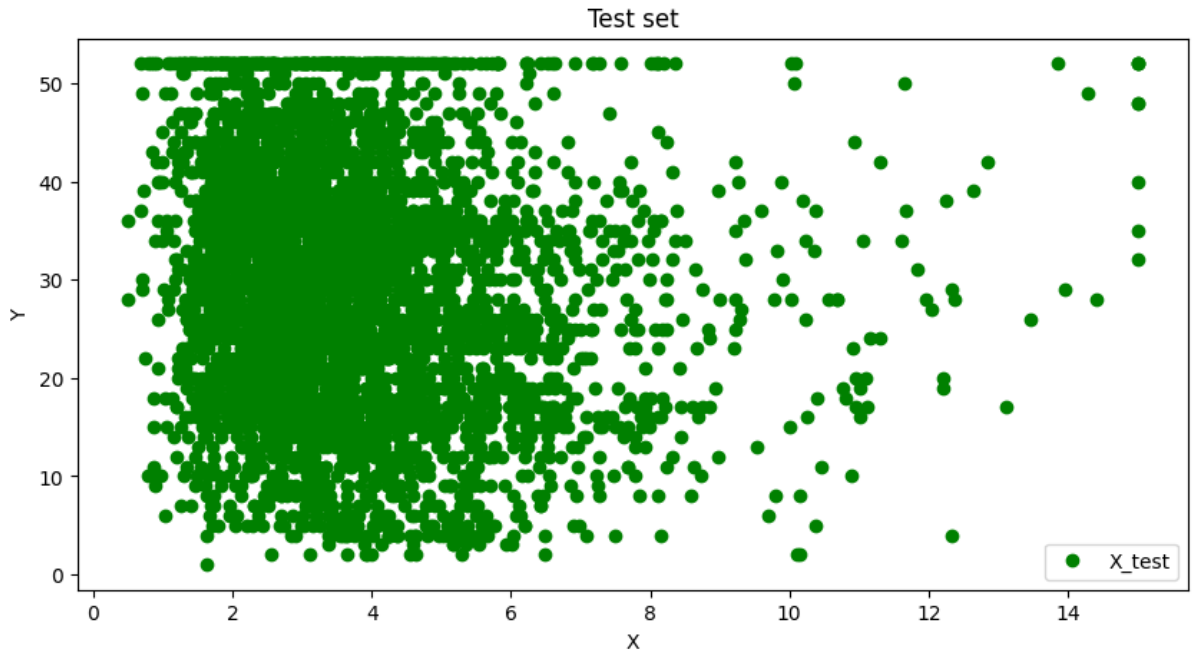


***Comparación de distintos dataset***




***Explicación***

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.6812 | 25.0 | 4.192201 | 1.022284 | 1392.0 | 3.877437 | 36.06 | -119.01 |
| 1 | 2.5313 | 30.0 | 5.039384 | 1.193493 | 1565.0 | 2.679795 | 35.14 | -119.46 |
| 2 | 3.4801 | 52.0 | 3.977155 | 1.185877 | 1310.0 | 1.360332 | 37.80 | -122.44 |
| 3 | 5.7376 | 17.0 | 6.163636 | 1.020202 | 1705.0 | 3.444444 | 34.28 | -118.72 |
| 4 | 3.7250 | 34.0 | 5.492991 | 1.028037 | 1063.0 | 2.483645 | 36.62 | -121.93 |




***Explicación***

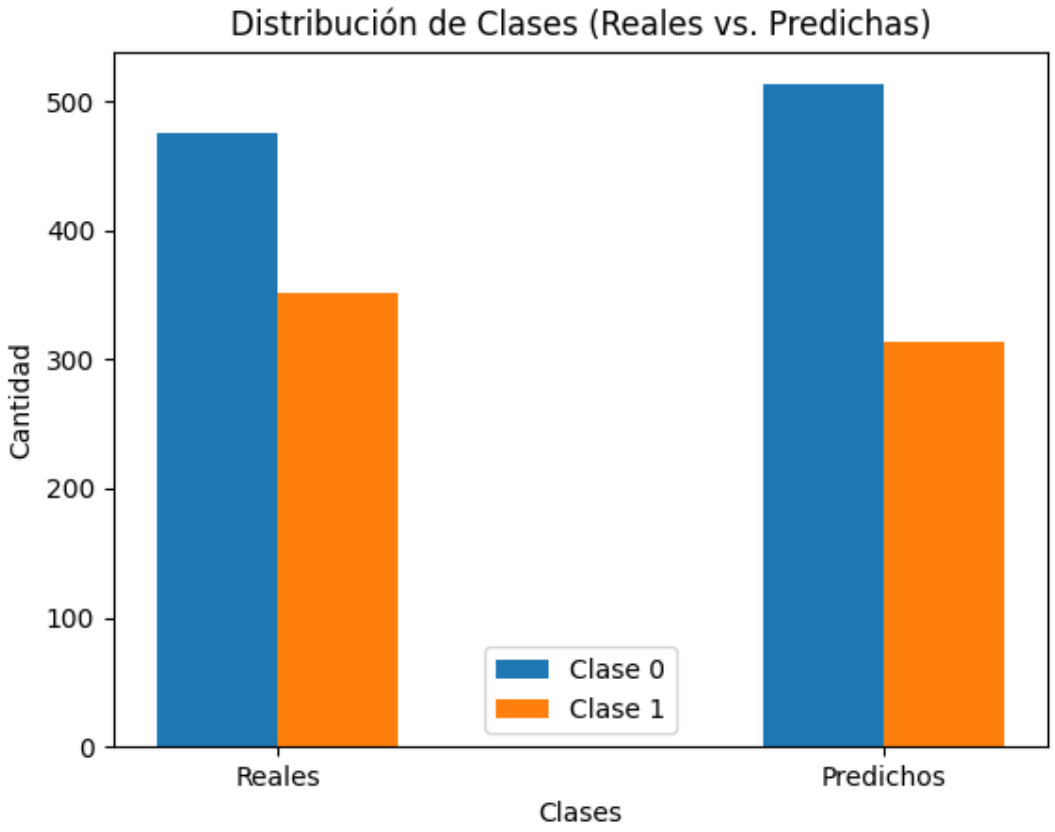
Como se puede observar tanto en las gráficas como en las muestra de los datos en cada uno de los datasets, la separación de los datos se realizó de manera correcta. En este caso, se puede observar que la distribución de los datos en cada uno de los datasets es similar, por lo que se considera que la separación de los datos nos ayuda a evitar el sesgo en el modelo. Aunado a esto, de acuerdo con las gráficas de dispersión, se puede observar que la distribución de los datos en cada uno de los datasets es similar, donde la mayoría toma una gran cantidad de valores, incluso los que parecen ser atípicos.
Esto nos garantiza que el modelo no se sesgó hacia una clase en específico, ya que la distribución de los datos es similar en cada uno de estos. Finalmente, se la separación de los datos en 80% para entrenamiento, 20% para validación y 20% para testing nos garantiza que el modelo no se sesgó hacia una clase en específico, ya que la distribución de los datos es similar en cada uno de estos. La importancia de tener un set de validación, y no solamente un set de testing, es que nos permite evaluar el desempeño del modelo en datos que no ha visto previamente, y que no se utilizaron para entrenar el modelo. Esto evita que el modelo se sobreajuste a los datos de entrenamiento y que no sea capaz de generalizar a datos que no ha visto previamente.

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

***Evidencia sobre el sesgo del modelo***

```
Precisión: 0.81
```

```
Etiquetas reales: 475 de clase 0 y 351 de clase 1
Etiquetas predichas: 513 de clase 0 y 313 de clase 1
Etiquetas reales: 57.51% de clase 0 y 42.49% de clase 1
Etiquetas predichas: 62.11% de clase 0 y 37.89% de clase 1
```




***Justificación***

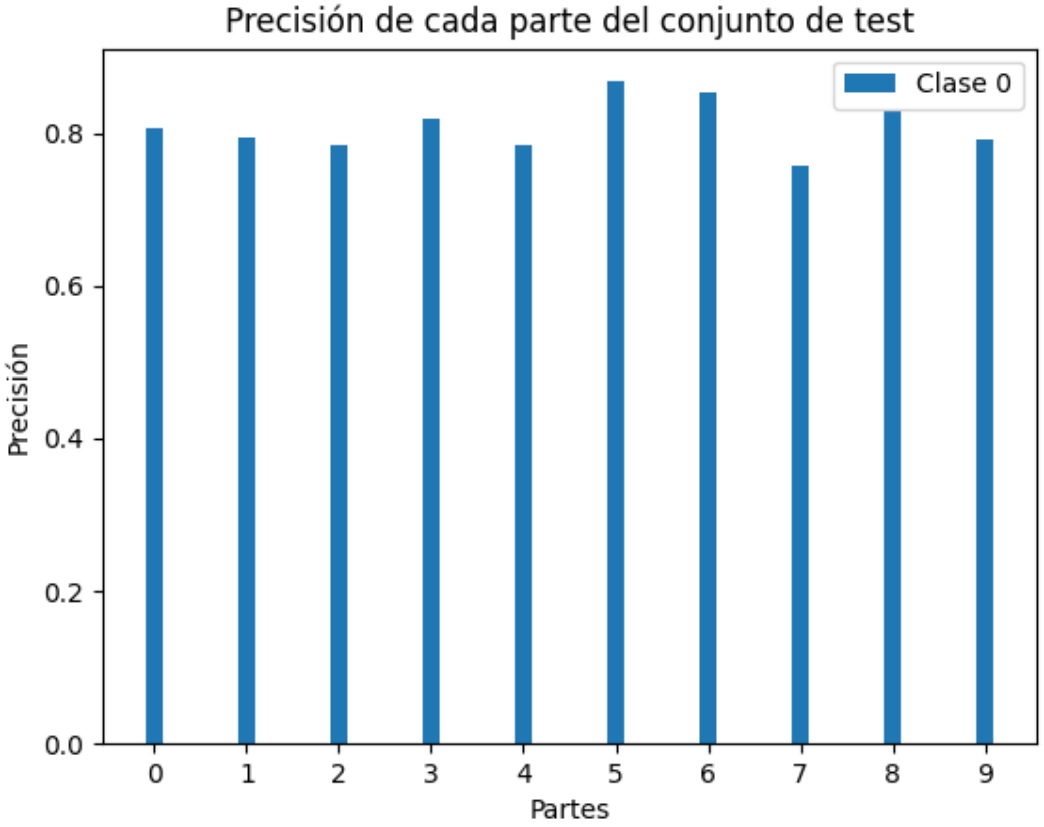
De acuerdo con el cálculo y las gráficas anteriores, La precisión de nuestro modelo entrenado sobre datos nunca antes vistos (test) es de 0.81, lo que nos indica que no existe un sesgo relevante que esté afectando el modelo. Esto se puede comprobar al realizar el cálculo que realiza sobre la predicción de las distintas clases. En este caso, tanto la clase 0 (0) como la clase 1 (1) demuestra que la predicción del modelo genera una distribución similar a la distribución real de los datos.

Esto nos indica que la predicción de dichas clases se realiza de manera exitosa. Esto también demuestra que el modelo no tiene una precisión con un nivel alto debido a que una clase tiene una mayor cantidad de datos que la otra, sino que el modelo es capaz de predecir de manera exitosa ambas clases.

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

***Evidencia de la varianza del modelo***

```
Test scores: [0.8072289156626506, 0.7951807228915663, 0.783132530
1204819, 0.8192771084337349, 0.7831325301204819, 0.86746987951807
23, 0.8536585365853658, 0.7560975609756098, 0.8292682926829268, 0
.7926829268292683]
Mean scores after 10 runs:
Test score: 0.808713
```




***Justificación***

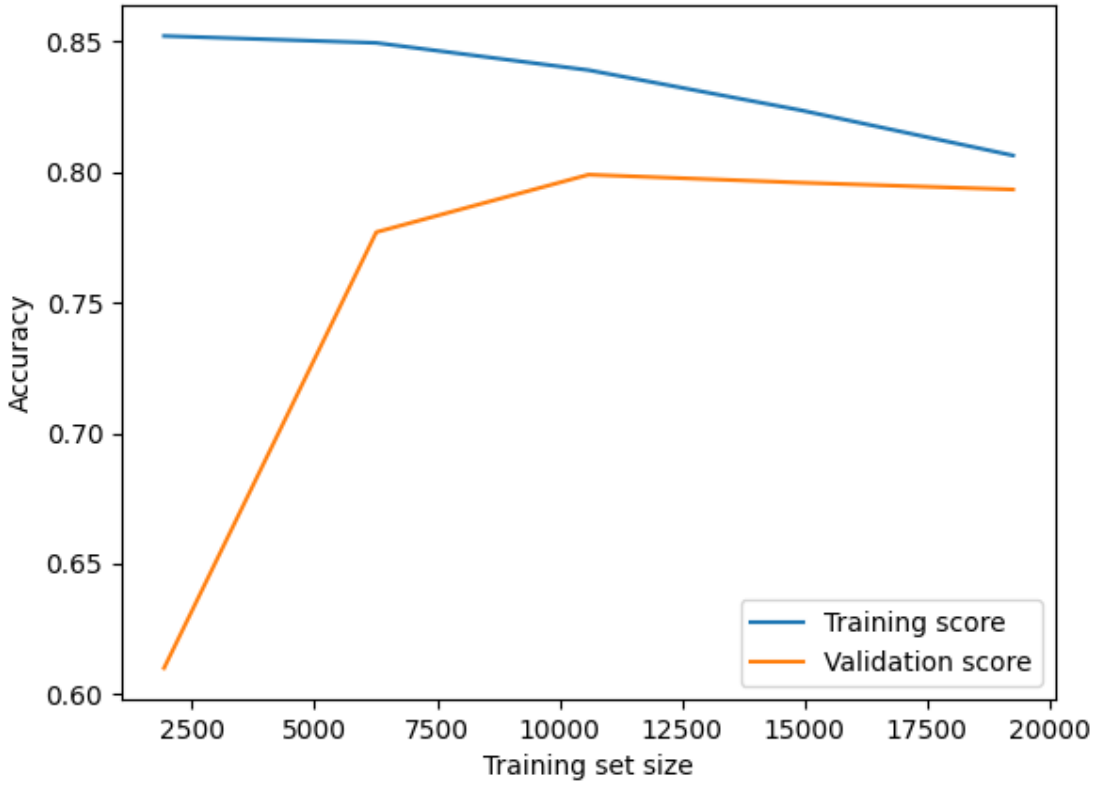
De acuerdo con la definición de la varianza, la varianza es una medida de la variabilidad de los datos de un conjunto de datos. En este caso, la varianza del modelo es baja. Esto se demuestra al dividir el set de testing en 10 partes, mostrando el rendimiento del modelo entrenado en cada una de las partes. Como se puede observar, la varianza del modelo es baja, ya que la precisión del modelo en cada una de las partes es similar. Esto nos indica que el modelo no presenta un sobreajuste (overfitting) ni un subajuste (underfitting). De igual manera, esto se puede demostrar con la gráfica de precisión del modelo en cada una de las partes. En este caso se utiliza esta división de test debido a que como estos datos no han sido vistos por el modelo, se puede observar que el rendimiento generaliza con datos nuevos sin tener una diferencia significativa en la precisión de cada uno de estos segmentos.

**Diagnóstico y explicación el nivel de ajuste del modelo: underfit, fit, overfit**

***Evidencia del ajuste del modelo***

```
Mean scores after 10 runs:
Train score: 0.799721
Validation score: 0.810418
Test score: 0.808717
```

```
Cross-validation scores: [0.81589147 0.79505814 0.82897287 0.8129845  0.78972868 0.78439922
 0.78827519 0.64680233 0.81686047 0.81637597]
Mean cross-validation score: 0.789535
```




***Justificación***

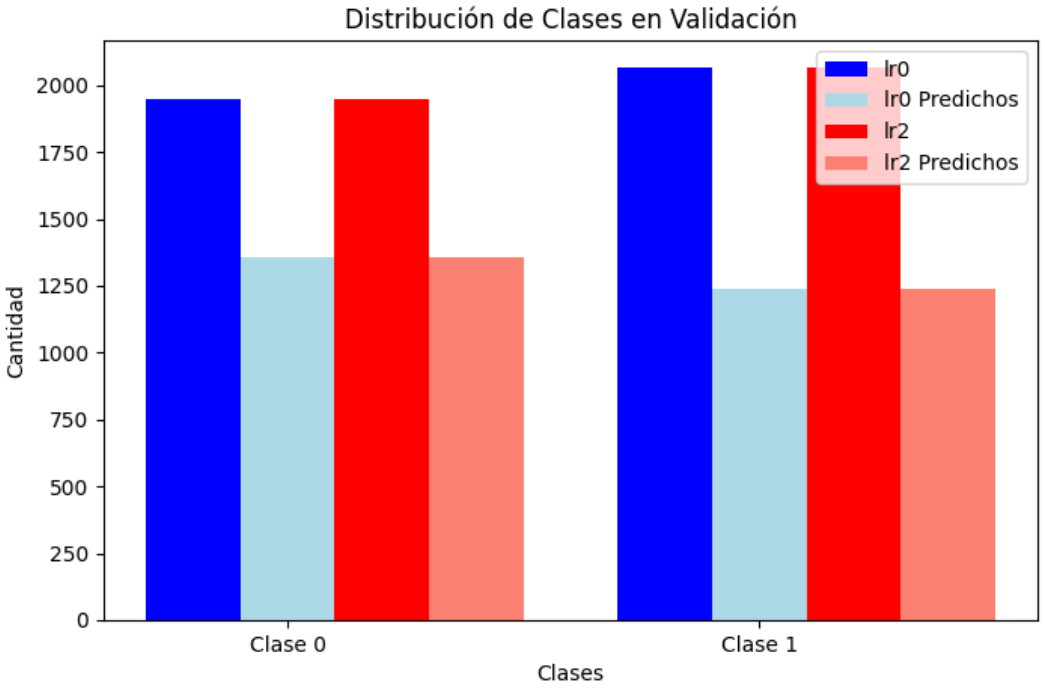
De acuerdo con los resultados tanto de precisión como de la comparación de las gráficas sobre el rendimiento del modelo sobre cada subset de datos, se puede observar que el modelo presenta un nivel de ajuste fit. Esto debido a distintas razones. La primer se puede notar que la calificación de las precisiones a lo largo de las 3 divisiones del dataset (train, validación y testing) se obtienen valores similares, lo que nos indica que el modelo no presenta un sobreajuste (overfitting) ni un subajuste (underfitting), porque si tuviera un sobreajuste, la precisión del modelo en el conjunto de entrenamiento sería mucho mayor que la precisión en el conjunto de validación y si tuviera un subajuste, la precisión del modelo en el conjunto de entrenamiento sería mucho menor que la precisión en el conjunto de validación. Cuando se realiza la función de cross-validation, se puede observar que la precisión del modelo en un promedio de 10 divisiones del dataset es casi del 80%, lo que nos indica que el modelo presenta un nivel de ajuste fit al no tener desviaciones significativas en la precisión del modelo en cada división del dataset.

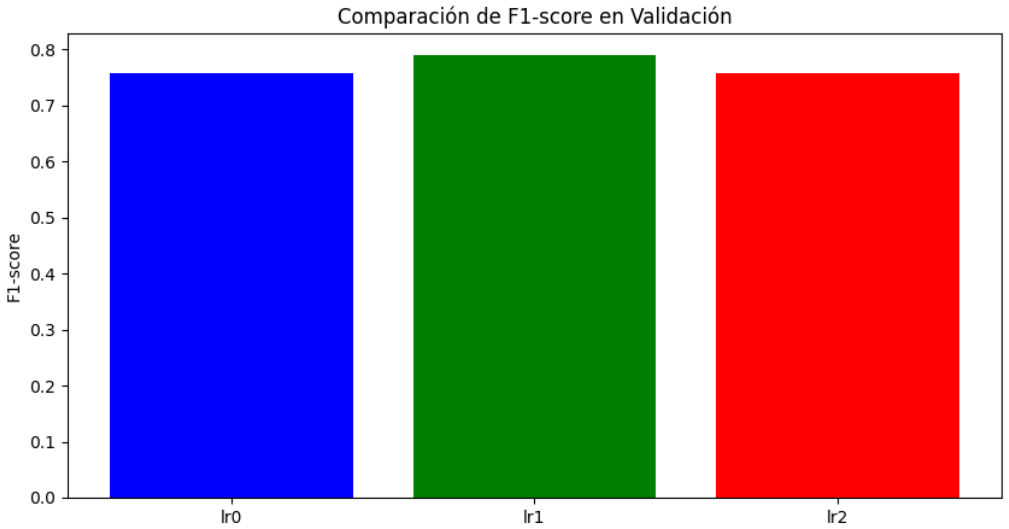
Por otro lado, al observar las gráficas de comparación de las curvas de aprendizaje, se puede notar que el modelo presenta un nivel de ajuste fit, ya que las curvas de aprendizaje convergen a un valor similar, lo que nos indica que el modelo no presenta un sobreajuste (overfitting) ni un subajuste (underfitting), porque si por ejemplo tuviera un sobreajuste, la

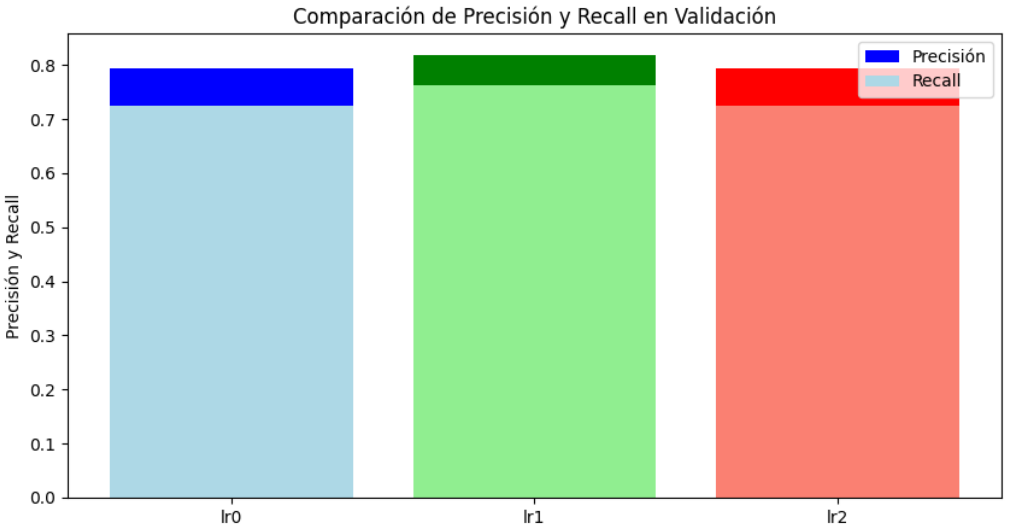
curva de aprendizaje del conjunto de entrenamiento convergerá a un valor mayor que la curva de aprendizaje del conjunto de validación.

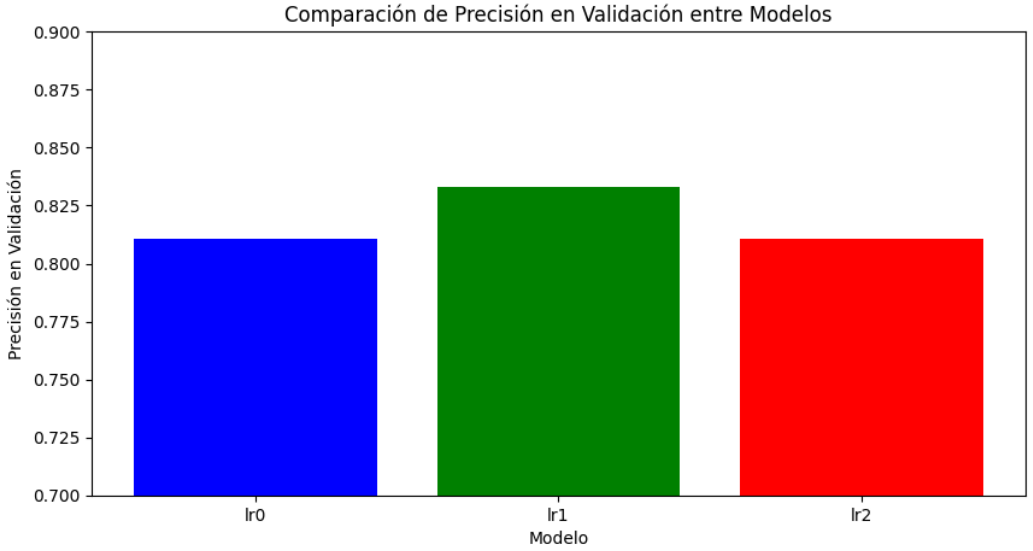
**Uso de técnicas de regularización o ajuste de parámetros para mejorar el desempeño**

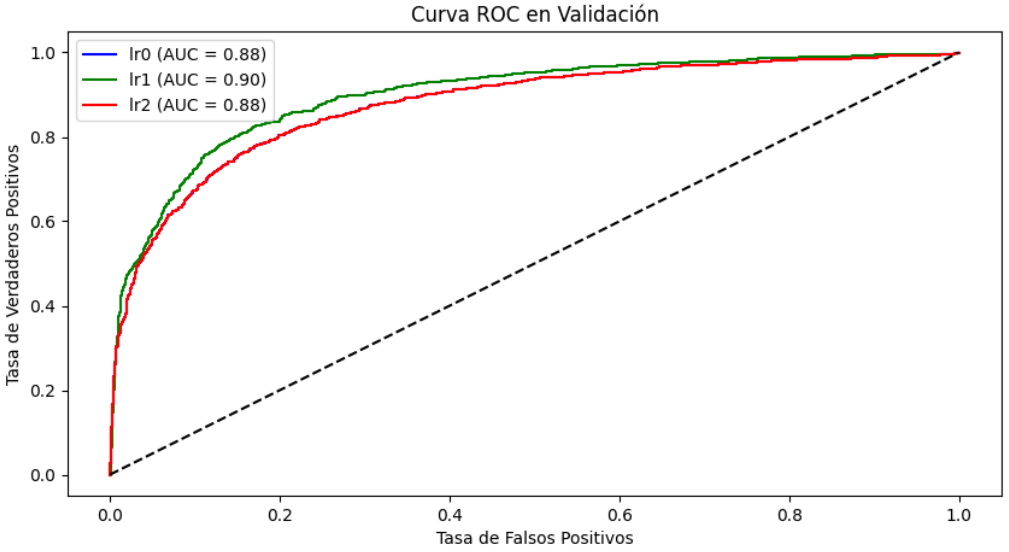
***Comparación de los modelos con los distintas técnicas de regularización***




Comparación de Precisión y Recall en Validación
Precisión
Recall
Precisión y Recall
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
lr0
lr1
lr2


Comparación de Precisión en Validación entre Modelos
Precisión en Validación
0.700
0.725
0.750
0.775
0.800
0.825
0.850
0.875
0.900
lr0
lr1
lr2
Modelo


Curva ROC en Validación
lr0 (AUC = 0.88)
lr1 (AUC = 0.90)
lr2 (AUC = 0.88)
Tasa de Verdaderos Positivos
0.0
0.2
0.4
0.6
0.8
1.0
Tasa de Falsos Positivos

***Justificación***

Aunque el modelo se ajusta bastante bien a los datos de entrenamiento, es importante que podamos predecir los datos que no se han visto antes. De lo contrario, el modelo no será útil para predecir nuevos datos. Esto se conoce como sobreajuste (overfitting) y ocurre cuando el modelo se ajusta demasiado bien a los datos de entrenamiento y no generaliza bien a los datos nuevos. Hay varias formas de mitigar el sobreajuste, como la regularización. A continuación, se muestra la utilización de dos técnicas de regularización: la regularización L1 y la regularización L2 para intentar mejorar el modelo.

Como se puede observar antes de la regularización, el modelo tiene una precisión del 81%, lo que significa que el 81% de las predicciones positivas son correctas. Sin embargo, cuando se observa la gráfica de ROC se puede notar que existe un poco de mejora que se puede realizar para que el incremento de la tasa de verdaderos positivos sea mayor. Esto nos lleva a aplicar la regularización. Al aplicar la regularización L1 (Lasso) se puede observar que la precisión del modelo es ligeramente mejor que la del modelo sin regularización (lr0), lo que indica que la regularización L1 ha mejorado el rendimiento en términos de precisión. Por otro lado, al aplicar la regularización L2 (Ridge) se puede observar que la precisión del modelo es similar a la del modelo sin regularización (lr0), lo que indica que la regularización L2 no ha tenido un impacto significativo en la precisión. Esto nos puede indicar ciertas cosas como las siguientes:

- La regularización L1 (Lasso) tiende a ser útil cuando se sospecha que algunas características son irrelevantes y se desea una selección automática de características.
- La regularización L2 (Ridge) tiende a ser útil cuando se desea evitar el sobreajuste sin necesariamente eliminar características.

Por lo que nuestro dataset, al mejorar con la regularización L1 (Lasso), nos indica que algunas características son irrelevantes y al realizar la selección automática de características, se mejora la precisión del modelo.

En temas más específicos, al tomar la gráfica de precision-recall, el modelo sin regularización tiene un mayor recall que el modelo con regularización L1 (Lasso), lo que indica que el modelo sin regularización tiene un mejor rendimiento en términos de determinar qué tan bueno es el modelo para predecir los positivos. Al igual que en la gráfica de comparación de F1-score en donde en la regularización L1 es mayor, lo que nos indica que el modelo sin regularización tiene un mejor rendimiento en términos de precisión y recall de manera general.