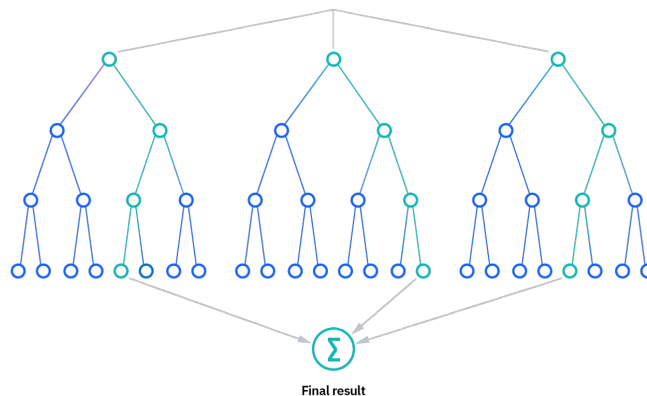


**Instituto Tecnológico y de Estudios
Superiores de Monterrey**
Campus Monterrey

Inteligencia artificial avanzada para la ciencia de datos I
TC3006C.101

**Reporte de Evaluación y Refinamiento de
modelo**



Rodolfo Sandoval Schipper A01720253

Marcelo Márquez A01720588

Arturo Garza Campuzano A00828096

3 sept 2023

Historia de Revisión

Nombre	Fecha	Razón de Modificación	Versión
Rodolfo Sandoval Schipper	3 sept 2023	Revisión.	1
Marcelo Márquez Murillo	3 sept 2023	Revisión.	1
Arturo Garza Campuzano	3 sept 2023	Revisión.	1

Índice general

1. Introducción.....	4
2. Separación de datos.....	4
3. Selección de métricas.....	4
4. Resultados obtenidos.....	4
5. Comparación de resultados.....	4
6. Técnicas de refinamiento.....	4

1. Introducción

La etapa del proyecto que corresponde a este entregable es el proceso de refinamiento de modelos. En este caso, el objetivo primordial es evaluar, comparar y mejorar los resultados de los modelos previamente generados. El proceso de refinamiento es esencial para identificar qué aspectos de nuestro modelo seleccionado (Random Forest Classifier) están funcionando efectivamente y cuáles necesitan ajustes y mejoras.

En este reporte se cubrirán las siguientes partes de este proceso:

- Separación de datos
- Selección de métricas
- Resultados obtenidos
- Comparación de resultados
- Técnicas de refinamiento

Los archivos que estaremos utilizando del ETL son **train_data** y **test_data**.

Ambos archivos se encuentran aquí:

https://drive.google.com/file/d/1T_ZRV0zAYqC3d1am8xCXJ6CqDRyptBw0/view?usp=sharing

https://drive.google.com/file/d/1gJDScqXANL15oZilDhXhkL_fxla-Jrjh/view?usp=sharing

2. Separación de datos

Dentro de la implementación hemos dividido el conjunto de datos que se encuentran en el archivo de **train_data**. La variable objetivo **survived** es lo que queremos predecir en el modelo. Al separarla, garantizamos que el modelo no tenga acceso a esa información que está tratando de predecir durante el entrenamiento. Esto simula la situación para que el modelo utilice los datos para hacer predicciones de datos nuevos o no vistos.

Para evaluar el modelo, necesitamos comparar las predicciones con las variables reales de la variable objetivo “survived”. Al tener un conjunto de datos de prueba con “survived” por separado, podemos medir la precisión, y otras métricas para evaluar el rendimiento del modelo. Sin embargo, también separamos la variable

objetivo para hacer pruebas independientes y generalizar nuevos datos. El hacer predicciones precisas de los datos con de acuerdo a esta variable nos permite utilizar el modelo más a futuro para ingresar nuevos datos y obtener los resultados con de acuerdo a la separación que se implementó en el modelo. A continuación se mostrará el ejemplo de la separación de los datos.

Datos de Entrenamiento X_{train}

PassengerId	Pclass	Age	Sex	Fam	Fare	Embarked
1	3	22	1	1	0	2
2	1	38	0	1	3	1
3	3	26	0	0	1	2
4	1	35	0	1	3	2
5	3	35	1	0	1	2

Datos reales del entrenamiento y_{train}

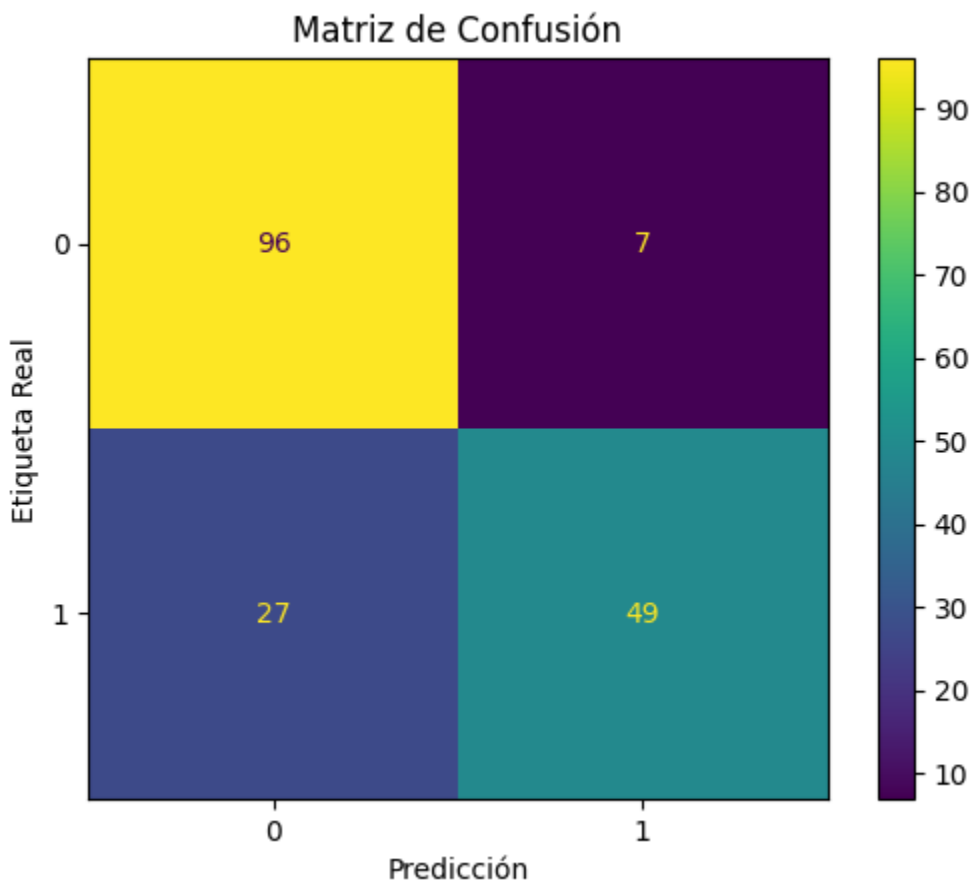
PassengerId	Survived
1	0
2	0
3	1
4	1
5	0

3. Selección de métricas

La evaluación del rendimiento del modelo es crucial para entender su eficacia en la clasificación de los pasajeros como sobrevivientes o no sobrevivientes. Para ello, se han seleccionado varias métricas de evaluación, incluyendo la Matriz de Confusión, Exactitud, Precisión, F1-Score y Recall.

3.1 Matriz de Confusión

Una tabla que muestra la distribución de predicciones y los valores reales del conjunto de datos. En ella, las filas representan las clases reales y las columnas las clases predichas. Los elementos diagonales (Verdaderos Positivos y Verdaderos Negativos) representan las instancias donde las predicciones del modelo coinciden con las clases reales. Los elementos fuera de la diagonal (Falsos Positivos y Falsos Negativos) indican errores. Esta herramienta proporciona una visión integral del rendimiento del modelo al mostrar los VP, VN, FP y FN. Esta métrica es especialmente útil para entender el tipo y la cantidad de errores que el modelo está cometiendo.



3.2 Exactitud

Un valor que representa la proporción de predicciones correctas en relación con el total de predicciones realizadas. Sin embargo, es importante tener en cuenta que una alta exactitud no siempre es indicativa de un buen modelo, especialmente si las

clases están desequilibradas. Por ejemplo, si el 95% de los pasajeros no sobreviven y el modelo predice que nadie sobrevive, aún tendríamos una exactitud del 95%, pero el modelo sería inútil para predecir los sobrevivientes.

$$\text{Exactitud} = \frac{VP + VN}{VP + VN + FP + FN}$$

3.3 Presición

Se centra en las predicciones positivas hechas por el modelo, en sí revisa cuales predicciones las cuales fueron positivas realmente lo son. Esta métrica es especialmente útil cuando un Falso Positivo son más costosos que los Falsos Negativos.

$$\text{Precisión} = \frac{VP}{VP + VN}$$

3.4 Exhaustividad

También conocido como Tasa de Verdaderos Positivos, es otra métrica que se centra en las predicciones positivas pero revisa cuales en verdad fueron positivas, en sí, se pregunta cuántos fueron identificadas correctamente por el modelo. Es útil cuando los FN son más costosos que los FP. En el contexto de la entrega, una exhaustividad alta significa que el modelo es capaz de identificar a la mayoría de los pasajeros que realmente sobrevivieron.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

3.5 F1-Score

Es una métrica que busca el balance entre la precisión y la exhaustividad del modelo. Es especialmente útil cuando una métrica es mucho más alta que la otra, lo que podría llevar a interpretaciones erróneas del rendimiento del modelo. Es más utilizada cuando la distribución de las clases es desequilibrada (200 valores de 1 y 500 valores de 0).

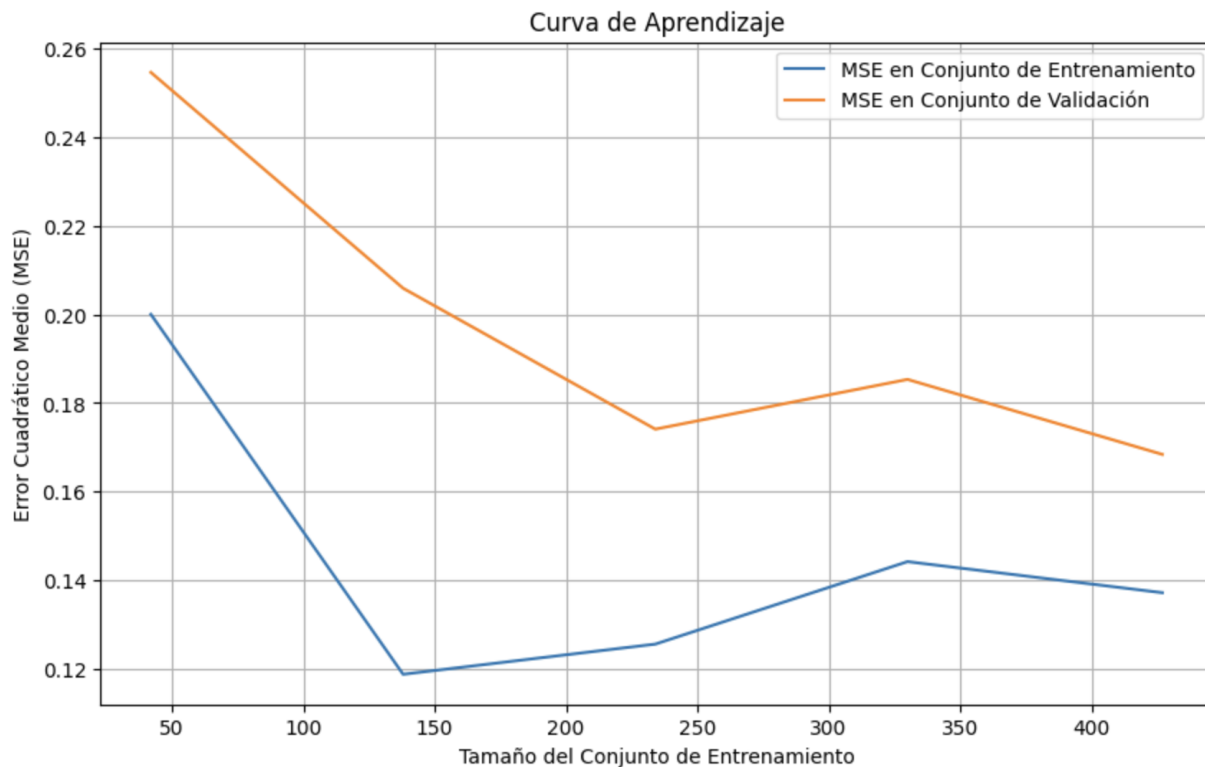
$$F1 - Score = 2 * \frac{\text{Precisión} * \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}}$$

4. Resultados obtenidos

4.1 Curva de aprendizaje

Para observar la curva de aprendizaje se elaboró una gráfica del tamaño del conjunto de entrenamiento contra el error cuadrático medio (MSE) tanto del conjunto de entrenamiento como el conjunto de validación. En la gráfica se observaron las siguientes cosas:

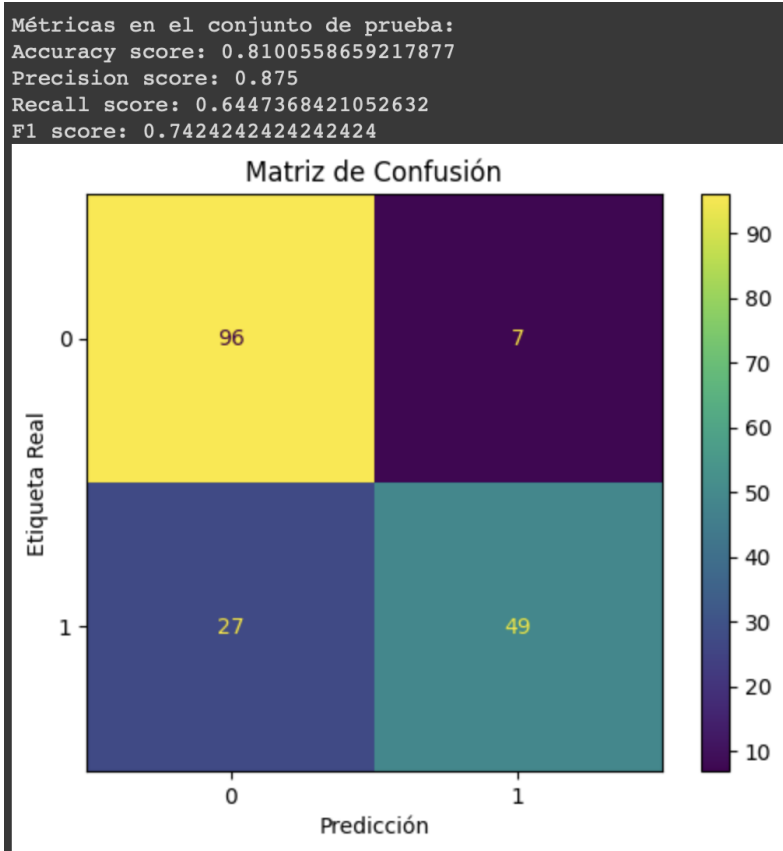
- Las **tasas de error de los conjuntos de entrenamiento y validación** disminuyen a medida que incrementa el tamaño de los conjuntos, lo cual indica que el modelo mejora su capacidad de generalización a medida que se le proporcionan más datos. También se puede observar que estas tasas se estabilizan a medida que el tamaño de los conjuntos aumenta.
- La **brecha entre los conjuntos** de la tasa de error es pequeña (0.025), lo cual es una señal de que el modelo generaliza bien a nuevos datos.
- Ambas curvas se acercan y se mantienen paralelas, lo cual indica que el modelo está aprendiendo de manera efectiva y generalizando bien.



4.2 Resultados con subconjunto de prueba

Las métricas obtenidas probando el modelo con el subconjunto de prueba fueron las siguientes:

1. **Accuracy Score (Exactitud):** el 81% de las predicciones realizadas por el modelo fueron acertadas.
2. **Precision Score (Precisión):** el 87.5% de las veces que el modelo predice una supervivencia tiene razón.
3. **Recall Score (Exhaustividad):** el modelo captura 64.47% de todos los casos de supervivencia presentes en el conjunto de prueba.
4. **F1 score:** con un valor de 0.7424, muestra que el modelo tiene un buen equilibrio entre la precisión y exhaustividad.
5. **Matriz de confusión**
 - Verdaderos Positivos: 96 casos fueron clasificados correctamente como sobrevivientes.
 - Falsos Positivos: 7 casos se clasificaron incorrectamente como sobrevivientes.
 - Falsos Negativos: 27 casos se clasificaron incorrectamente como no sobrevivientes.
 - Verdaderos Negativos: 49 casos fueron clasificados correctamente como no sobrevivientes.



En resumen, estos resultados indican que el modelo es bastante preciso en sus predicciones y el equilibrio entre precisión y exhaustividad se refleja en un sólido puntaje de F1. Sin embargo, su capacidad para identificar a todos los sobrevivientes podría mejorarse.

4.3 Predicciones puntuales

Al final del código para esta entrega se encuentra un ejemplo de predicción puntual. El modelo recibe los datos de un pasajero y regresa si sobrevivió o no.

5. Comparación de resultados

Modelo Anterior	Modelo Refinado
Accuracy score: 0.8044692737430168 Precision score: 0.8153846153846154 Recall score: 0.6973684210526315 F1 score: 0.7517730496453899	Accuracy score: 0.8100558659217877 Precision score: 0.875 Recall score: 0.6447368421052632 F1 score: 0.7424242424242424

Cómo podemos observar, hemos obtenido una configuración que nos ayuda a alcanzar el mejor refinamiento del modelo. Esto se logra aplicando estándares y técnicas de regularización para buscar los mejores hiper parámetros y aplicarlos de acuerdo a nuestro conjunto de datos. También se tomó en cuenta que el modelo y su configuración se debe de ubicar en un punto donde no se implemente un caso de “overfitting” o “underfitting” cómo se puede visualizar en la matriz de confusión. Sin embargo, el modelo utiliza: 'n estimators' cantidad de árboles, 'max depth' profundidad de árbol, 'Min Samples Split' cantidad mínima de muestras para dividir un nodo, y la medición con entropy. El rendimiento óptimo se obtuvo calculando diferentes parámetros dentro de un rango viable para los datos del modelo. Este rango fue probado por iteración para comparar la precisión de cada hiper parámetro y sacar el resultado más alto.

6. Técnicas de refinamiento

Para optimizar el rendimiento del modelo, empleamos un enfoque sistemático de ajuste de hiper parámetros utilizando un diccionario que especifica diversas opciones para la cantidad de árboles, la profundidad máxima de los árboles y el número mínimo de muestras requeridas para dividir un nodo. Este diccionario se integra con la función GridSearchCV, que explora exhaustivamente todas las combinaciones posibles de estos hiper parámetros.

La técnica de validación cruzada se incorpora en este proceso para evaluar la eficacia del modelo en datos no vistos. En nuestro caso, se ha configurado un valor de cinco pliegues, lo que significa que el conjunto de datos se segmenta en cinco partes. Cuatro de estas se emplean para el entrenamiento del modelo, mientras que la quinta se utiliza para la validación. Este procedimiento se repite cinco veces, rotando la partición de validación en cada iteración.

Los hiper parámetros como la profundidad máxima de los árboles y el número mínimo de muestras para dividir un nodo actúan como mecanismos de regularización. Estos parámetros restringen la complejidad del modelo, lo que contribuye a una mejor generalización cuando se enfrenta a nuevos datos.

Al concluir la búsqueda en la cuadrícula, GridSearchCV devuelve los hiper parámetros que han demostrado ofrecer el mejor rendimiento durante las fases de validación cruzada. Estos parámetros óptimos se utilizan entonces para el entrenamiento final del modelo, asegurando así que se ha alcanzado una configuración que maximiza la precisión y robustez del modelo.