Proyecto Entrega Final

POR:

Tomas Edil Urango Ruiz

Lucas Bustamante

MATERIA:

Introducción a la inteligencia artificial

PROFESOR:

Raul Ramos Pollan

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

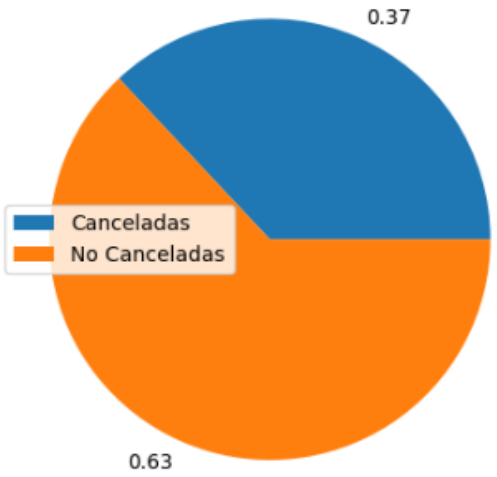
2023

Informe

Introducción

Se tiene interés en averiguar cuántas de sus reservas son canceladas, y poder predecir cuándo se cancelará una reserva, por lo cual con el código desglosamos el porcentaje de reservas canceladas en el hotel, el cual es de un 37%.

```
El porcentaje de reservas canceladas en el hotel es 37.04%
```



La cantidad de reservas que son canceladas es significativa, y por lo tal tener una herramienta con la cual predecir cancelaciones tiene el potencial de ser útil para la hotelera.

Exploración de datos y preprocesamiento.

Se tiene que para los modelos de predicción de las cancelaciones hoteleras se elaboró la exploración de los datos previamente obtenidos, se hizo el procesamiento y con la ejecución de código procedimos a realizar la revisión de cantidad y porcentaje de valores nulos obteniendo datos relevantes en los factores country, agent y company.

country	488	0.408744
market_segment	0	0.000000
distribution_channel	0	0.000000
is_repeated_guest	0	0.000000
previous_cancellations	0	0.000000
previous_bookings_not_canceled	0	0.000000
reserved_room_type	0	0.000000
assigned_room_type	0	0.000000
booking_changes	0	0.000000
deposit_type	0	0.000000
agent	16340	13.686238
company	112593	94.306893

Se revisa qué clase de variable contienen las columnas de valores nulos, para poder llenarlas adecuadamente.

Los valores nulos de la columna "country" pueden reemplazarse arbitrariamente por "desconocido", luego, los demás valores numéricos representan identificaciones, además de hijos, variables que podemos reemplazar con cero razonablemente. usando un heatmap, podemos ver qué columnas presentan correlación con nuestro dato de interés y seguido podemos visualizar la correlación de las variables a continuación

```
text/plain
  is_canceled                      1.000000
  lead_time                        0.293123
  total_of_special_requests        0.234658
  required_car_parking_spaces      0.195498
  booking_changes                  0.144381
  previous_cancellations           0.110133
  is_repeated_guest                0.084793
  company                          0.082995
  adults                           0.060017
  previous_bookings_not_canceled   0.057358
  days_in_waiting_list             0.054186
  adr                              0.047557
  agent                            0.046529
  babies                           0.032491
  stays_in_week_nights             0.024765
  arrival_date_year                0.016660
  arrival_date_week_number         0.008148
  arrival_date_day_of_month        0.006130
  children                         0.004393
  stays_in_weekend_nights          0.001791
  Name: is_canceled, dtype: float64
```

Análogamente, veamos qué columnas numéricas se tiene la porción categórica del data frame, pasamos las fechas a un formato legible por el modelo y luego, se descartan las columnas redundantes.

Para las columnas de variables numéricas, un método de normalización ayuda a que el modelo no le dé prioridad indebida a ciertas columnas, se ve a continuación la varianza de las diferentes columnas sin normalizar, se aplicará un método de normalización logarítmica para aquellas columnas que lo requieren, cambiar el método puede afectar el puntaje del modelo.

	deposit_type	customer_type	reservation_status_year	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	reservation_status_month	reservation_status_day	...	babies	\
0	7	1	...	0	
1	7	1	...	0	
2	7	2	...	0	
3	7	2	...	0	
4	7	3	...	0	

	is_repeated_guest	previous_cancellations	previous_bookings_not_canceled	'
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	

	agent	company	days_in_waiting_list	adr	\
0	0.000000	0.0	0.0	0.000000	
1	0.000000	0.0	0.0	0.000000	
2	0.000000	0.0	0.0	4.330733	
3	5.720312	0.0	0.0	4.330733	
4	5.484797	0.0	0.0	4.595120	

	required_car_parking_spaces	total_of_special_requests
0	0	0
1	0	0
2	0	0
3	0	0
4	0	1

[5 rows x 27 columns]

Finalmente, se obtiene el dataframe "X" preprocesado, y se hacen splits de test y de train. Se puede jugar con el test size para probar en el modelo.

Iteraciones de desarrollo

Importamos los modelos que pensamos que pueden servirnos para el procesamiento de los datos.

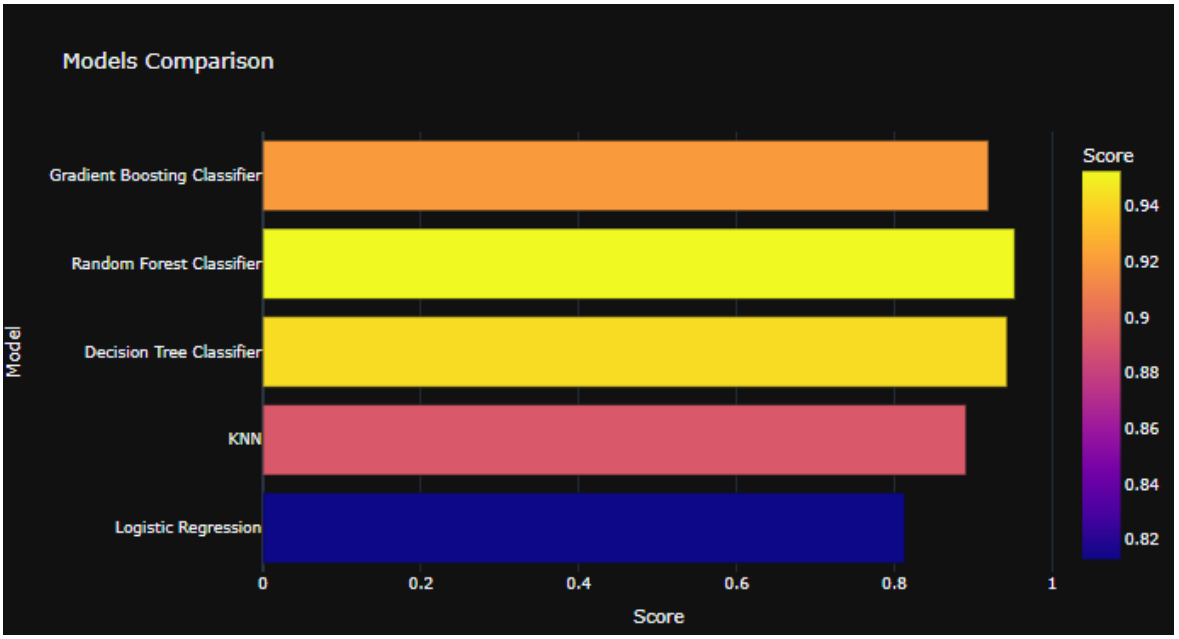
Se probó en su aplicación cual arroja una mayor predicción para nuestro caso, a continuación se muestran los modelos usados en una tabla de comparación donde se muestran desde el que obtuvo mayor a menor puntaje de precisión:

	Model	Score
3	Random Forest Classifier	0.952537
2	Decision Tree Classifier	0.943016
4	Gradient Boosting Classifier	0.919452
1	KNN	0.891001
0	Logistic Regression	0.813161

Una vez obtenida la tabla de comparación de modelos aplicados, se nota que hay varios modelos con muy buena predicción, llegando esta a ser superiores al 90%.

Los principales modelos que destacan, llegando a superar el 94%, son el modelo Decision Tree Classifier y el modelo Random Forest Classifier, con un 94.3% y un 95.2% respectivamente, siendo el modelo Random Forest Classifier con un 95.2% el modelo de preferencia para obtener el mayor puntaje de predicción.

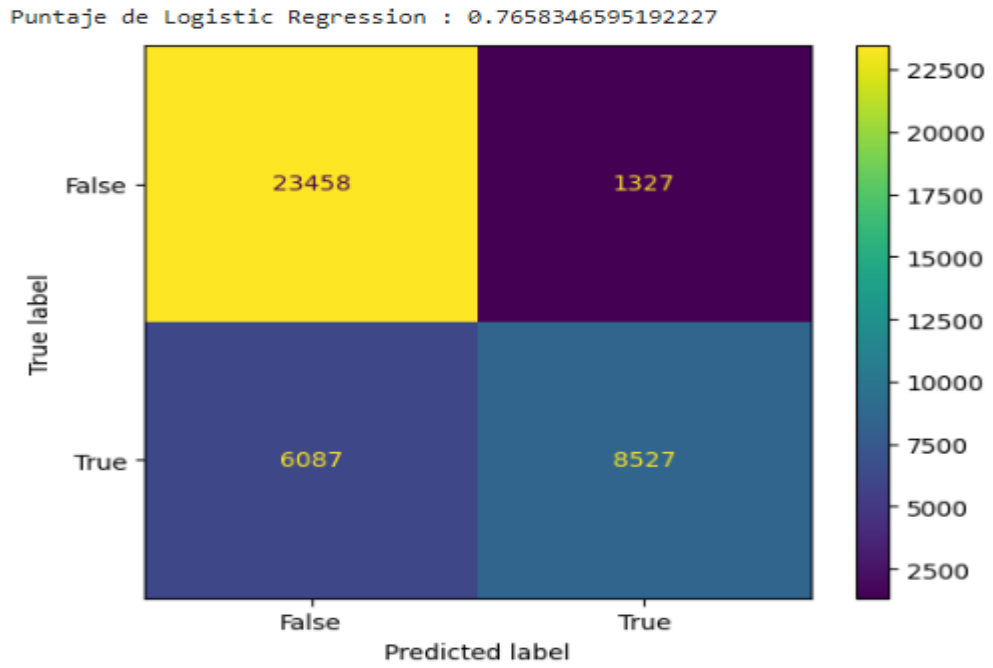
Lo anterior puede ser mejor ilustrado en la siguiente gráfica:



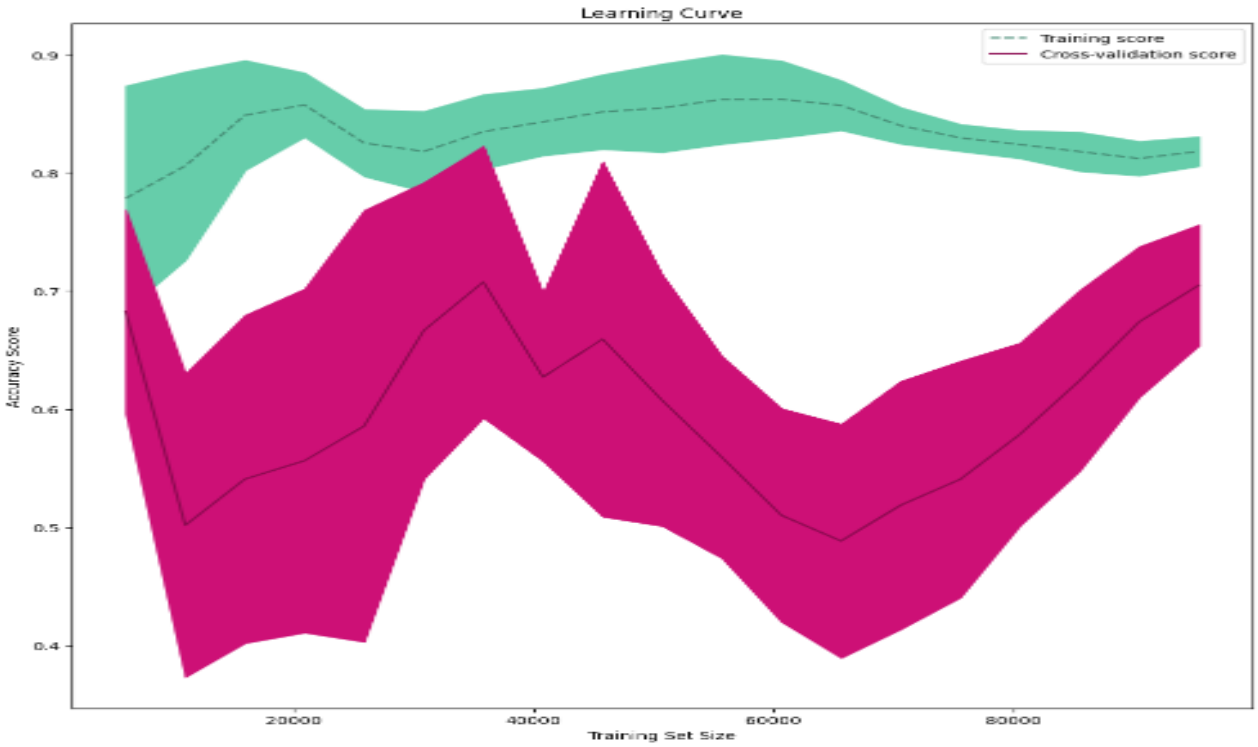
Esta iteración del proyecto nos mostró resultados favorables, pero para obtener métricas más fieles y precisas implementamos métodos de cross-validación.

Confusion Matrix, Puntajes de Cross-Validation y Curvas de Aprendizaje

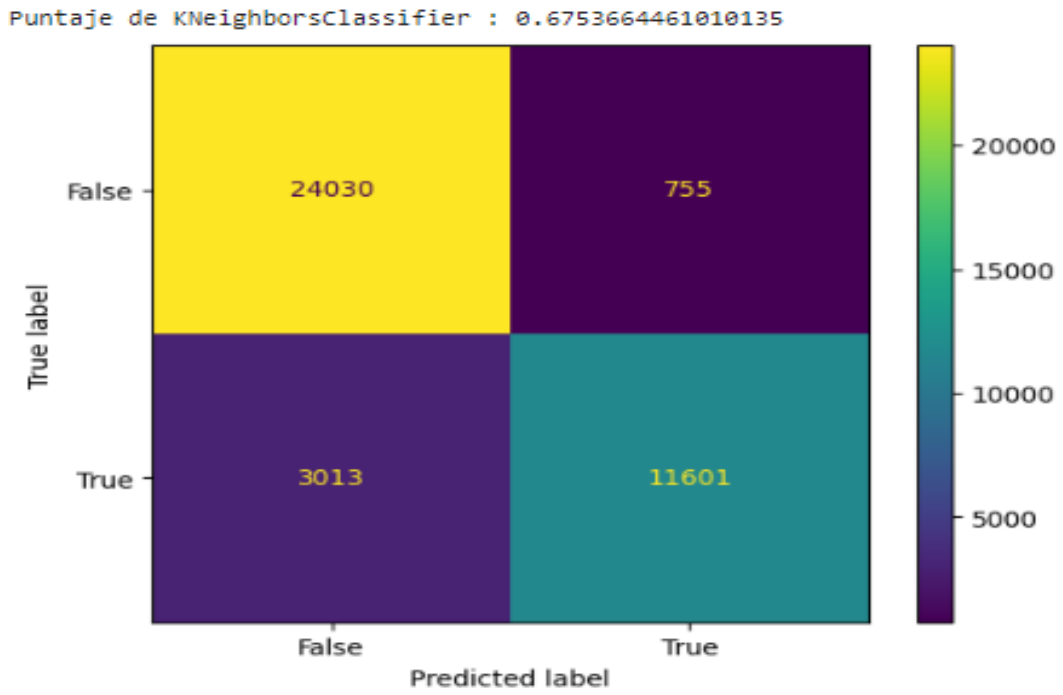
Usando Logistic Regression



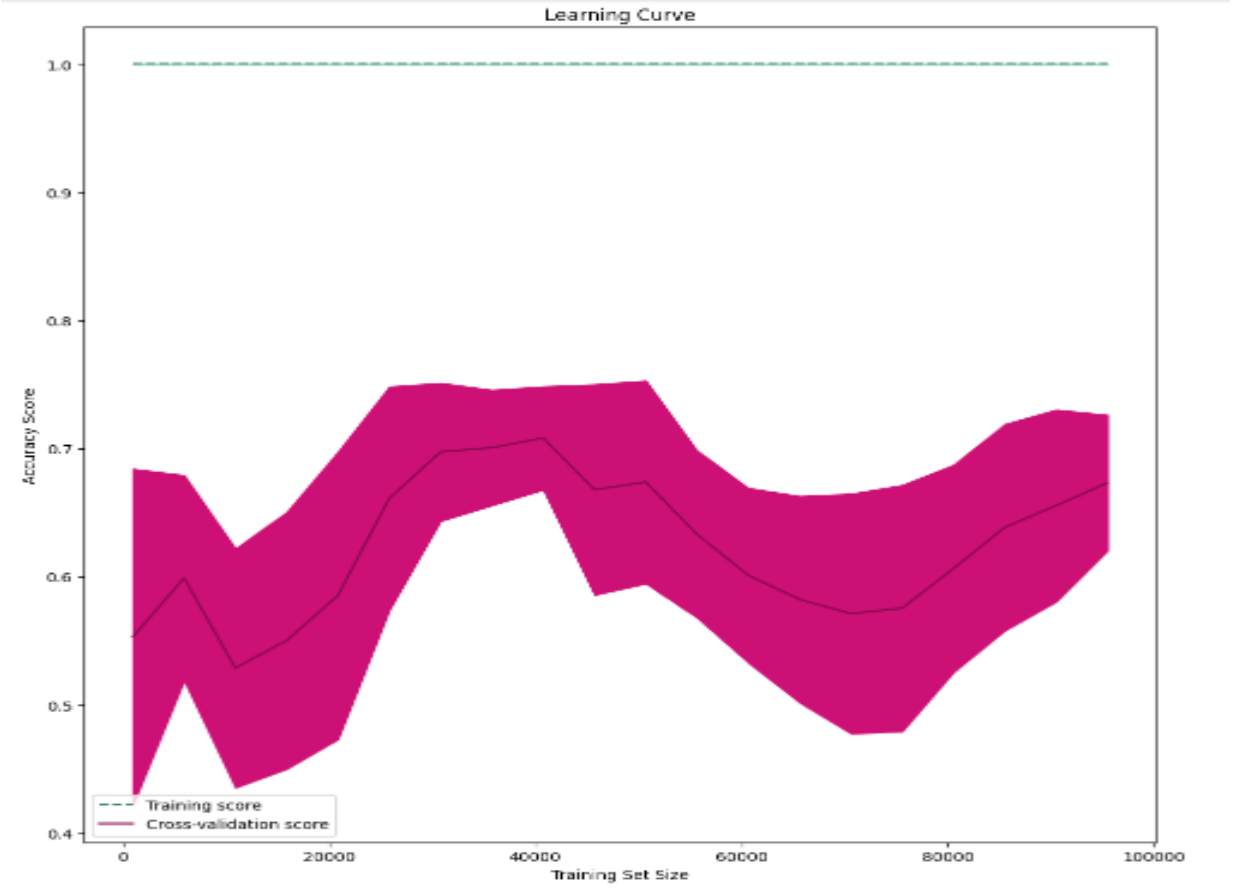
Seguido se grafica una curva de aprendizaje para LogisticRegression.



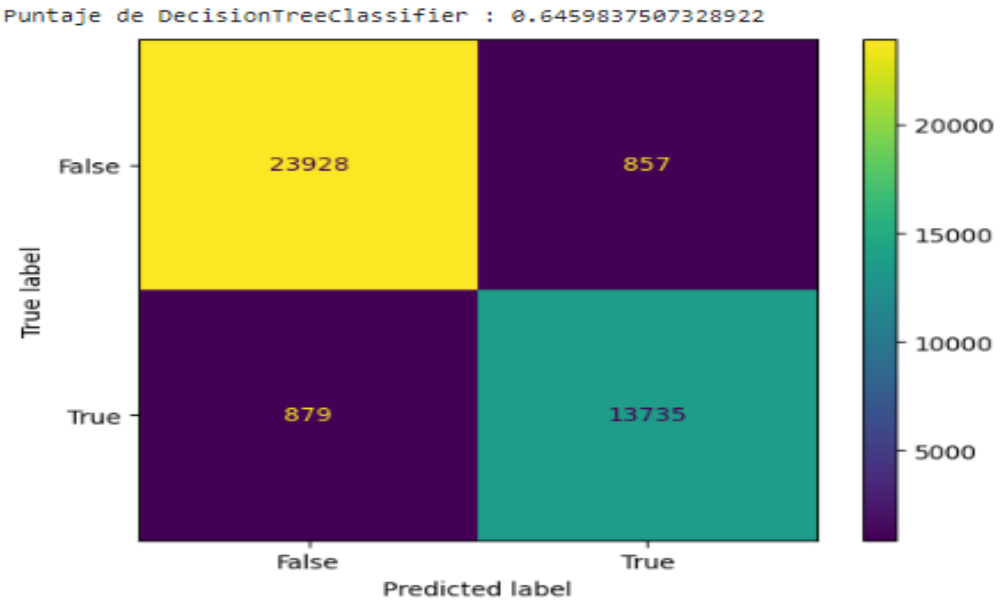
Ahora Usando KNeighborsClassifier.



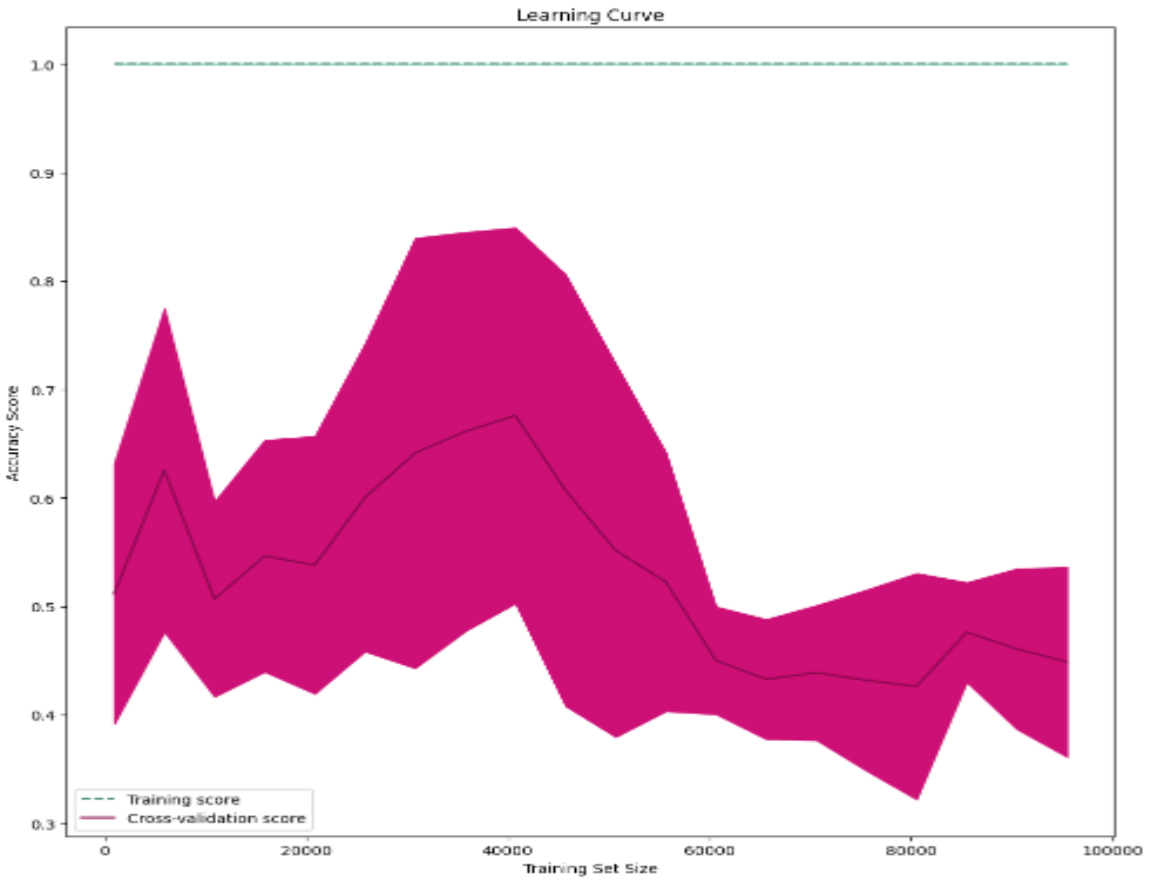
Igualmente se grafica una curva de aprendizaje para KNeighborsClassifier.



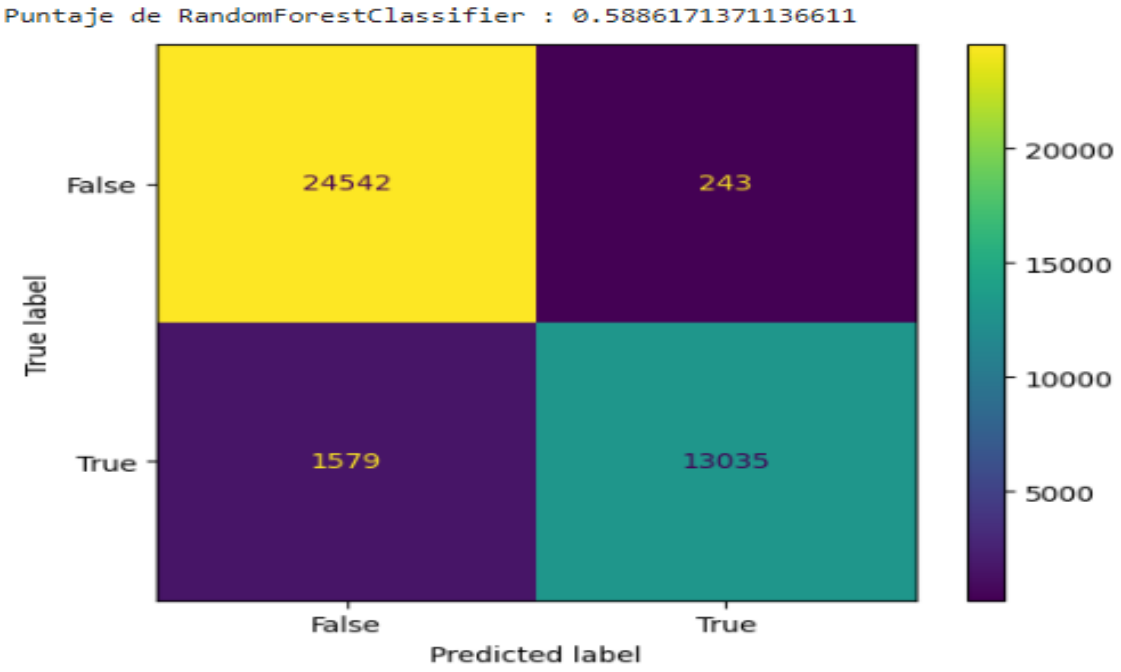
Ahora usando DecisionTreeClassifier.



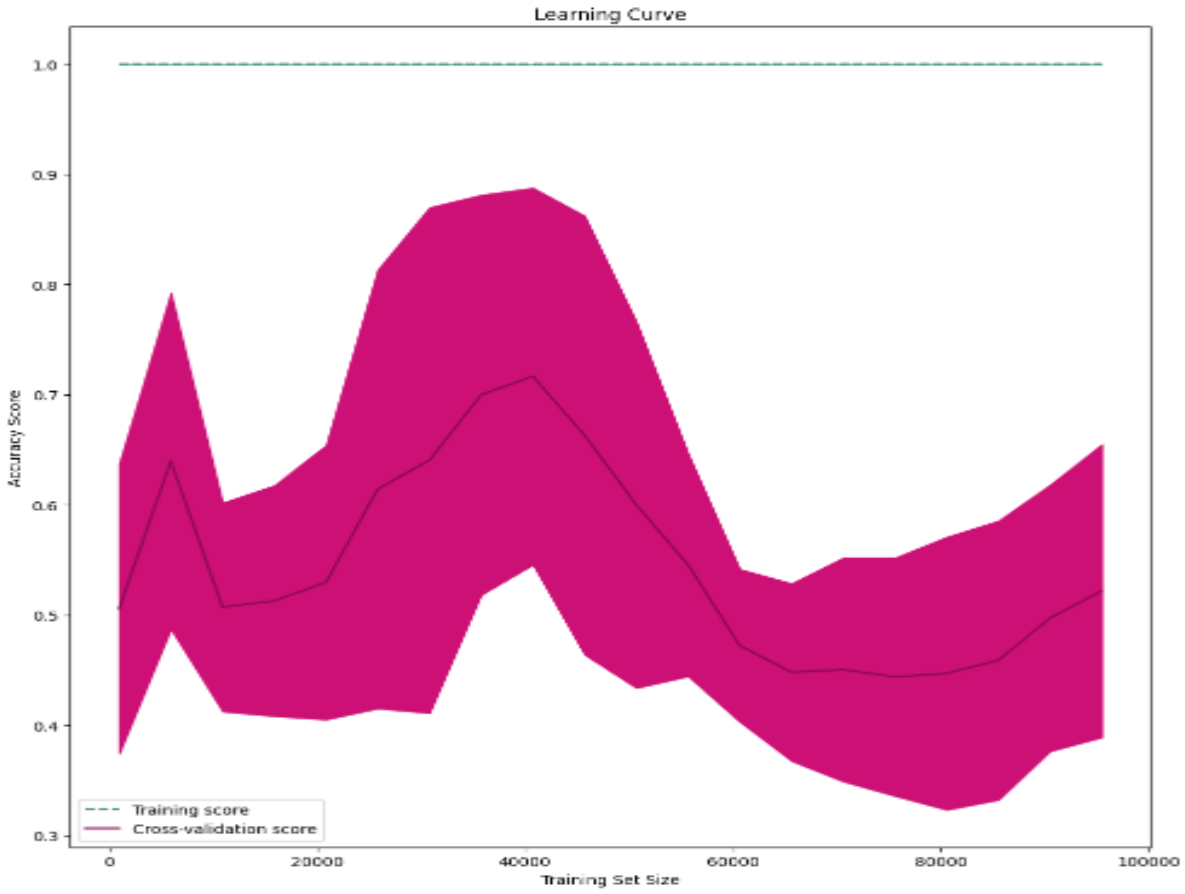
Curva de aprendizaje para DecisionTreeClassifier.



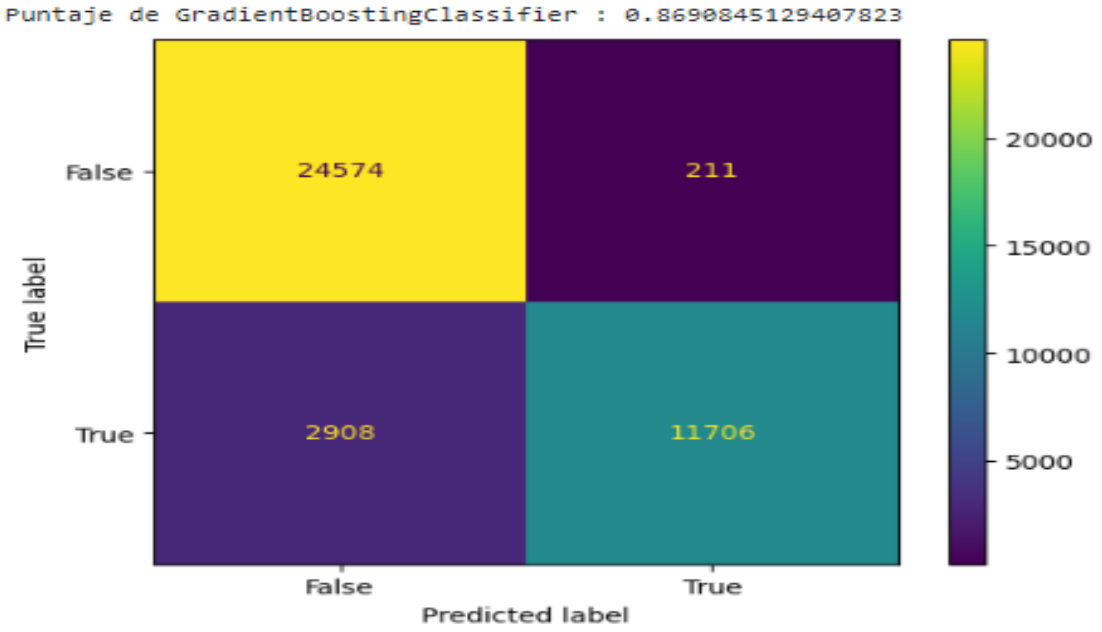
Usando RandomForestClassifier.



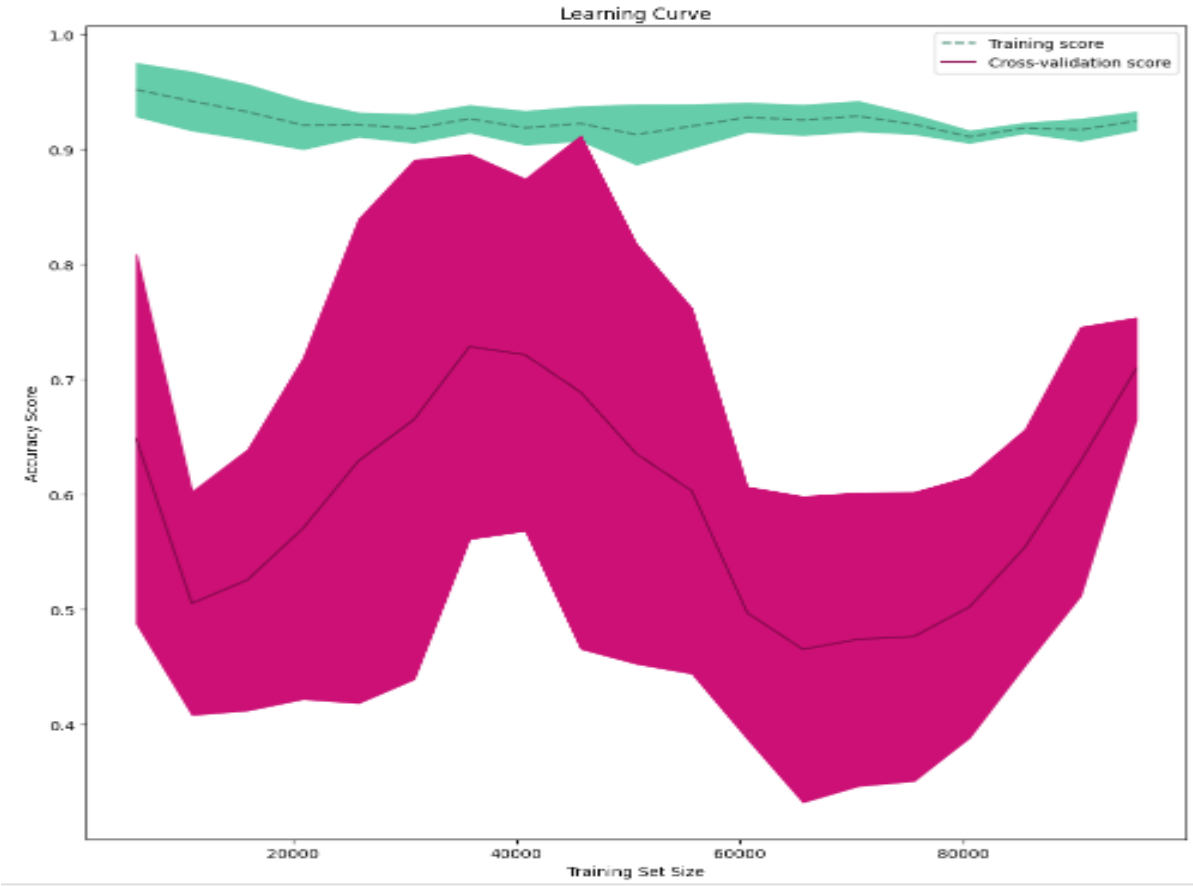
Curva de aprendizaje para RandomForestClassifier.



Usando GradientBoostingClassifier.



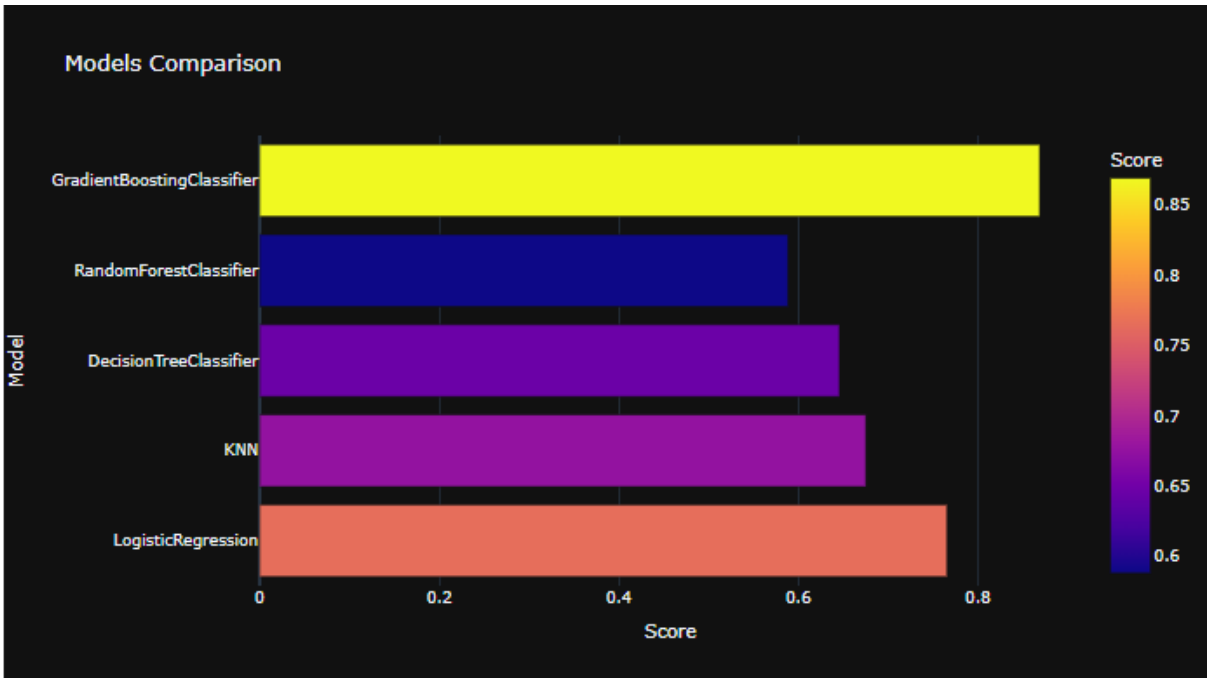
Curva de aprendizaje para GradientBoostingClassifier.



Se compara el desempeño de los diferentes modelos usados. Estos aún pueden refinarse más aún jugando con los distintos parámetros.

	Model	Score
4	GradientBoostingClassifier	0.869085
0	LogisticRegression	0.765835
1	KNN	0.675366
2	DecisionTreeClassifier	0.645984
3	RandomForestClassifier	0.588617

Se puede apreciar mejor la comparación en el desempeño con la siguiente gráfica:



Retos y consideraciones de despliegue.

Usando Logistic Regression, se llevan los hiperparametros óptimos, cross-validating con 10 folds, Inicialmente, se intentó utilizar GridSearchCV y RandomizedSearchCV, pero ambas funciones duraban demasiado tiempo en correr, por lo tal, se encontró hiperparametros iterando, o por prueba y error.

Conclusiones.

Análisis de Curvas de Aprendizaje.

En cuánto a los modelos probados para este problema de ML, los modelos de Kneighbors, DecisionTree y RandomForest mostraron bajo bias (Puntaje de train siempre alto) con mayores niveles de varianza, característicos de un Over-Fitting. LogisticRegression mostró decrecida varianza, aunque aún así una relativamente alta y con un error alto de testeo, además, por ello el modelo aumentó en bias y sufre de Under-Fitting.

Por otro lado, el modelo empleando GradientBoostingClassifier tiene un bias suficientemente bajo, y a mayor nivel de muestreo los puntajes de test y train parecen converger con un puntaje de precisión, indicando lo que sería el modelo más adecuado entre los probados.

Cómo optimizar aún más los modelos empleados.

Inicialmente, se consideró usar GridSearchCV o RandomizedSearchCV para completamente optimizar los hiperparametros de los modelos, los dos factores que limitaron la posibilidad de realizar esto fueron el tiempo necesario para lograrlo, junto con el poder computacional limitado ofrecido por Colab.

Además de esto, podrían alterarse algunas de las decisiones tomadas durante el preprocesamiento y al momento de hacer splits de train-test que potencialmente podrían alterar positivamente el rendimiento de los modelos.