Proyecto final
Data Science - Coderhouse

Presentación ejecutiva

SPACESHIP
TITANIC

Matias Thorel

Contenidos

Abstracto

En este trabajo, intentaremos diseñar y entrenar un modelo, que pueda realziar una predicón binaria o booleana, sobre los pasajeros de la nave que serán transportados a una realidad alterna. Esa es nuestra variable target "Transported: True/False". Para lograr nuestro objetivo, deberemos utilizar algoritmos de clasificación, y poder determinar quienes nuncan llegaran al destino deseado.

Para ello, tenemos el dataset de entrenamiento, que cuenta con mas de 8000 registros, al concer el resultado, haremos el análisis de cada variable comparandolas con el mismo.
Algunas variables por si misma no dicen nada, incluso tienen una cardinalidad alta, pero intentaremos encontrar la relación con el resultado, hacer limpieza de datos, detección de outliers, elegir la mejor estratagia para los registros vacios, crear variables nuevas combinando las existentes.

Luego, tenemos el dataset prueba, una muestra ciega con la cual poner en funcionamiento el modelo, deberiamos obtener la misma accuracy.
Las nuevas variables, calculos y modificaciones que hagamos, intentaré realziarlas meidantes funciones, asi poder reutilizar el código y no tener que escribir de nuevo y arriesgar a que me queden diferencias en ambos dataset.

Contexto comercial

La empresa Spaceship Titanic, se dedica al transportes interplanetario de pasajeros. Para unir los destinos, separados por miles de años luz, las naves, son capaces de crear puentes de Einstein-Rosen, y viajar a través de ellos, llegando así al planeta elegido por los clientes.

El inconveniente con esta tecnología, es que algunos pasajeros son transportados a realidades alternas, para lograr devolverlos a nuestro espacio-tiempo se requiere de misiones extremadamente complejas y por sobre todo las cosas, costosas.

Por lo tanto la empresa necesita encontrar la forma de reducir la perdida de pasjeros, y en consecuencia, aumentar la tasa de clientes transportados exitosamente a los destinos deseados.

Problema comercial

Para intentar descifrar los que, hasta ahora, parece deberse a cuestiones completamente azarosas, es necesario desarollar un modelo supervisado, donde a traves de la observación y anáisis de los datos de viajes pasados, podamos determinar quienes tendran mas probabilidades de llegar a destino y quienes seran transportados a realidades alternas. Cuales son las caracteristicas de los pasajeros, detectar patrones que nos permitan anticiparnos al desenlance inesperado.

Contexto analítico

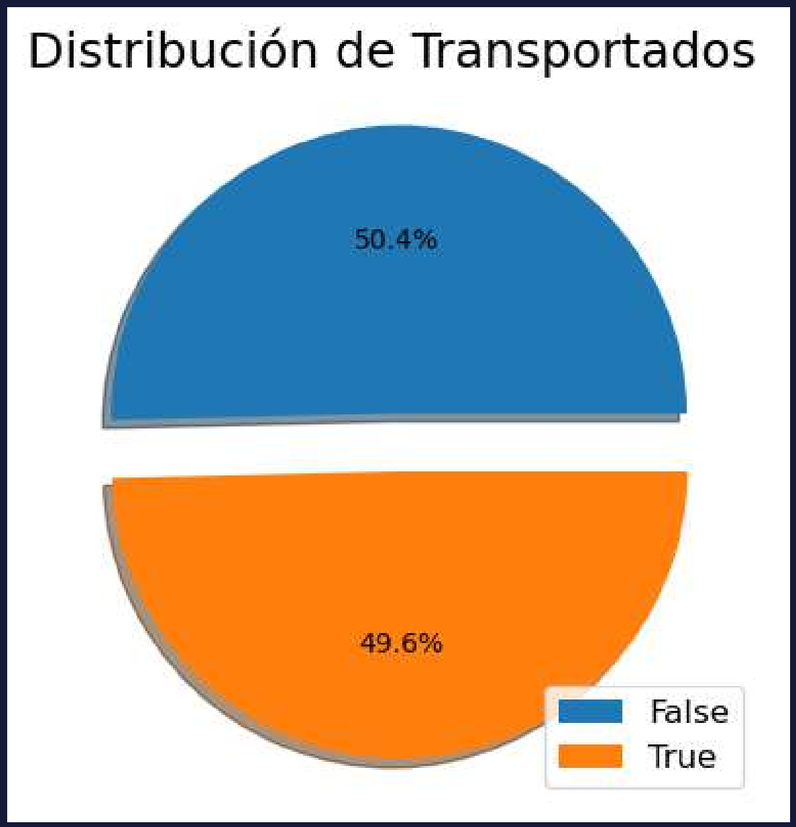
Para el desarrollo del modelo, contamos con las posibles variables que el servicio interplanetiario de Spaceship Titanic ofrece, los plantas de origen y destino, si el pasajero eligió viajar en CryoSleep, en que cabina viajó, los gastos realizados en los amenities de la nave. Luego, el nombre, la edad del pasajero y si es VIP o no.

Análisis exploratorio de datos (EDA)

La primer preguntas que no hacemos es, ¿Como se distribuye nuestra variable target? Es decir, cantos pasajeros son transportados a realidades alterenas y cuantos llegan a destino.

Con este siempre análisis univariado, expresamos en un piechart, la distribución de transportados.

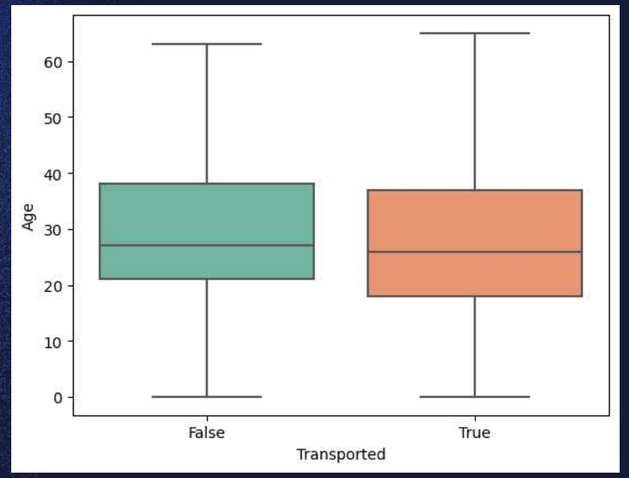
Obtenemos como respuesta, que la misma se encuentra practicamente balanceda.

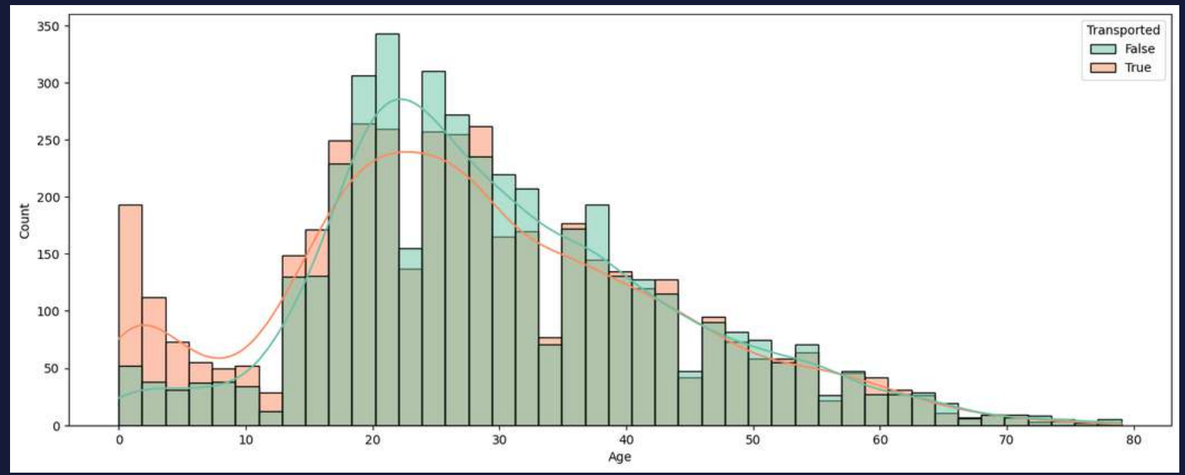


Luego, pensamos en la incidencia que pueda llegar a tener las edad de los pasajeros, en el desarrollo normal del viaje.

Mediante el boxplot e histograma desarollada, determinamos:

- La mayoría de los pasajeros tiene entra 18 y 33, siendo 27 la mediana y casi 29 la media.
- Los menores de 18 son los mas transportados, sobre todo los de 0 años, o recien nacidos.
- De 18 a 33, parecen ser los menos trasportados, y los de mas 33 estan parejos.





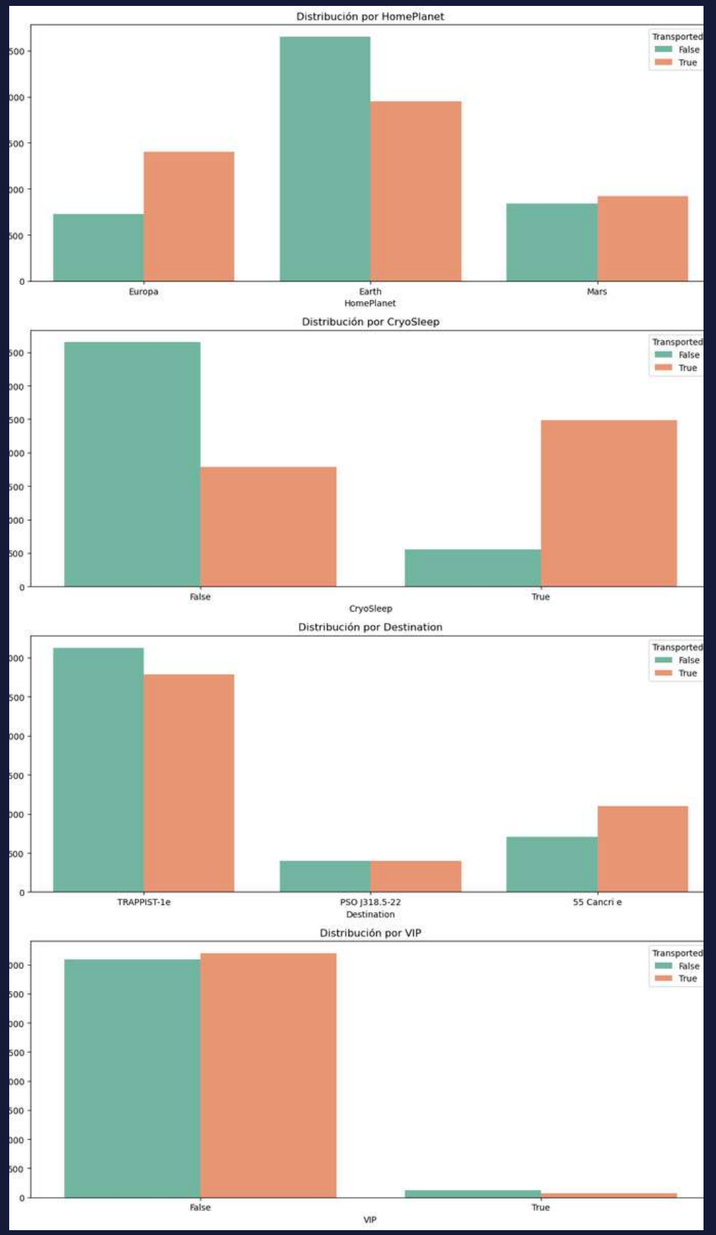
Variables categóricas vs. Target

- En la variable VIP, una categoría es muy dominante sobre la otra, por lo tanto se tomo la decisión de excluir este dato dataset, para evitar sobreajustar el entrenamiento.

- La mayoría de los pasajeros son de la Tierra, pero tiene una tasa menor de transportados a realidades alternas, comparandolos con Europa y Marte que tienen menos pasajeros, pero donde mas de la mitad no llegan a destino.

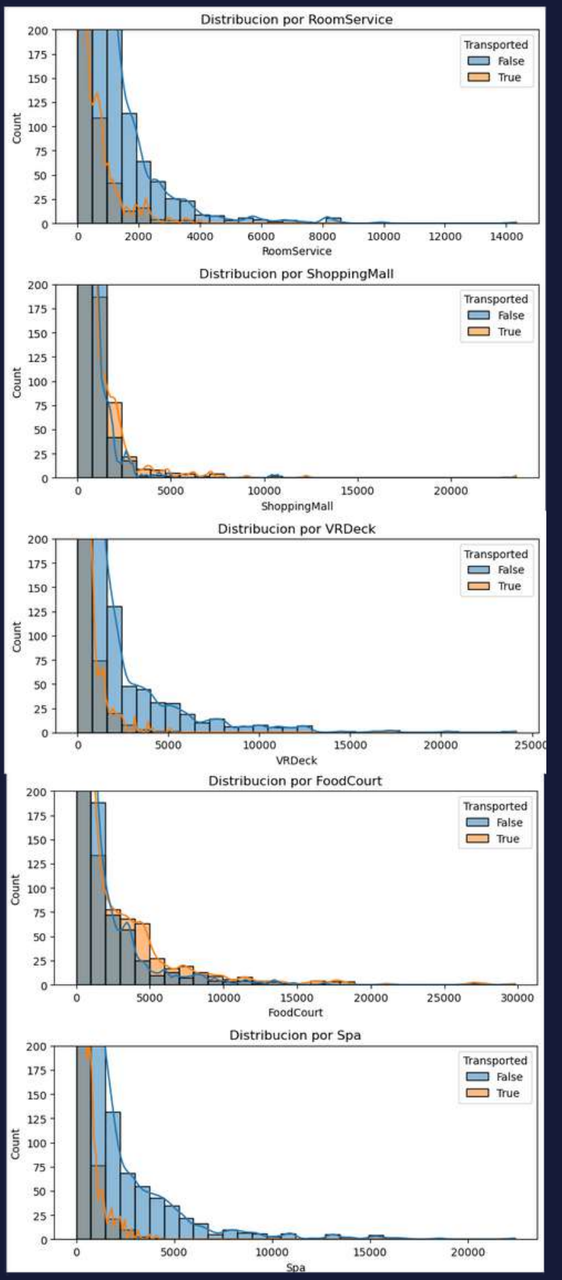
- La mayoría de los pasajeros se dirigen hacia Trappist-1e.

- La mayoría de los que viajan en modo "CryoSleep", fueron transportados a realidades alternas.



Variables de gastos vs. Target

- La mayoría de los pasajeros no gastaron.
- Parece que los pasajeros que tienen menos gastos tienen más probabilidades de ser transportados que los pasajeros que tienen gastos altos.
- ¿Tendrá relación que los que viajan en Cryosleep, al estar dormidos, no pueden gastar? Coincide con que queines viaján así, tambien son los mas transportados a otras realidades.

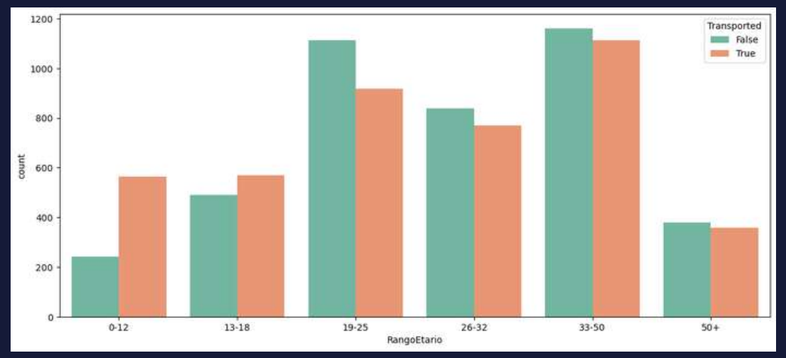


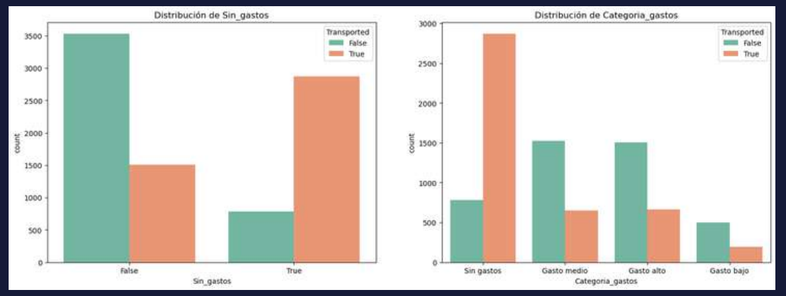
Features Engineering

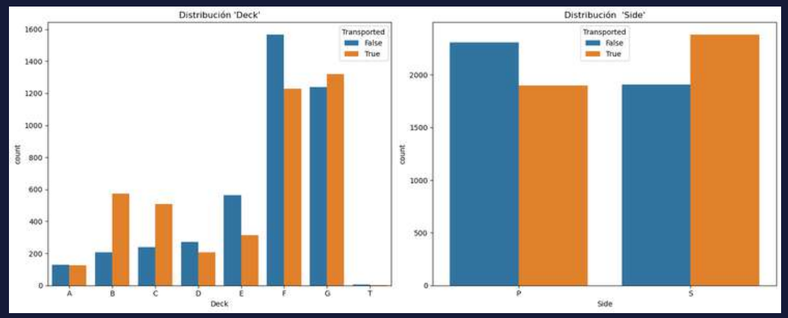
Luego de analizar las variables mencinadas, se diseñaron nuevas a partir de ellas, con el fin de entregar a los algoritmos de machine learning , data que permita mejores predicciones, estas son:

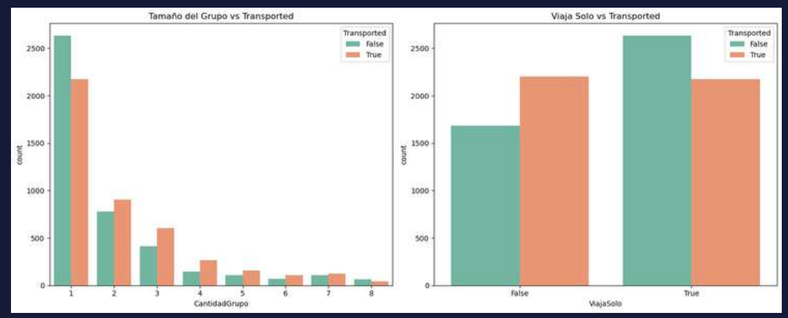
- Rango Etario, segmentamos le edad en rango.
- Gastos Totales, suma de gastos en amenities
- Sin gastos, gasto ¿si o no?

Tambien, se crearon Viaja solo, Tamaño de grupo, Sector de cabina, Lado de cabina









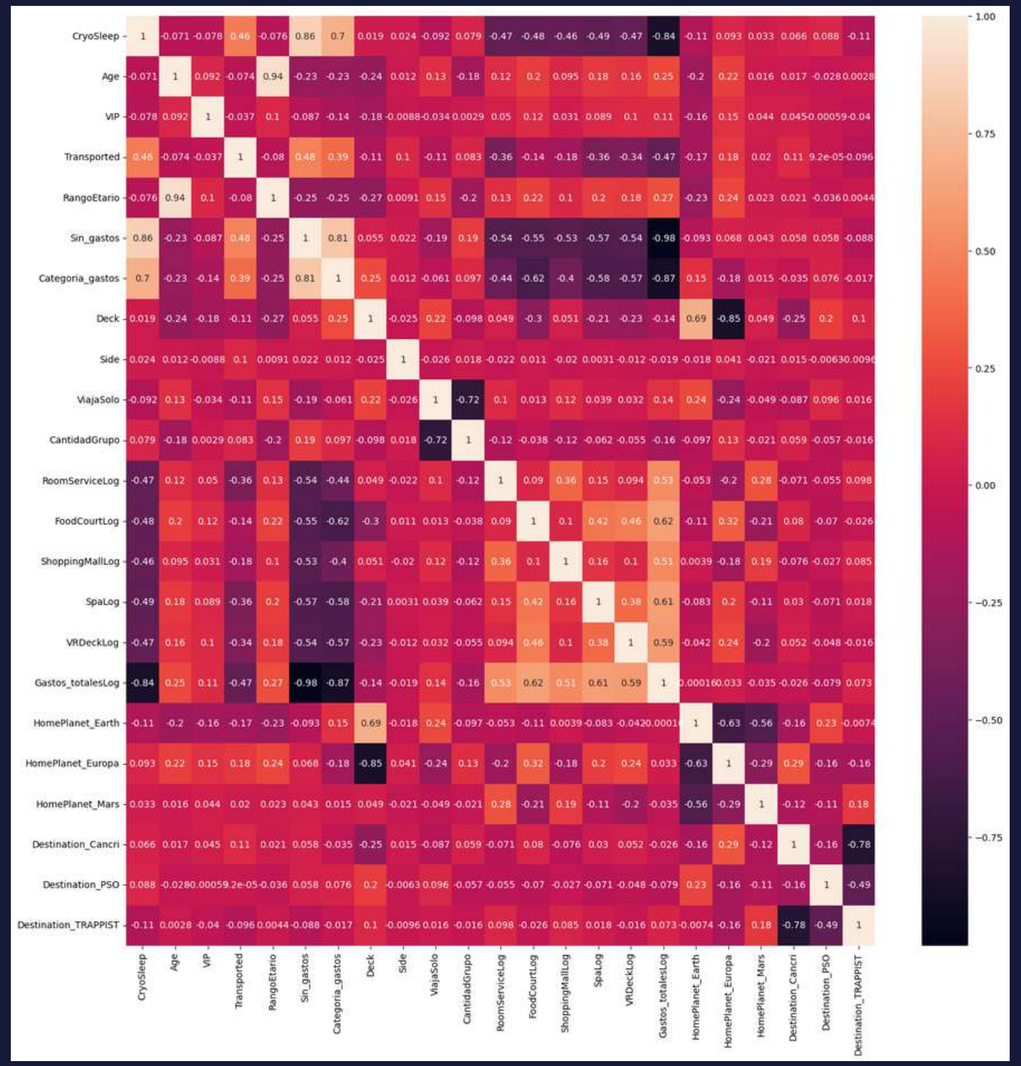
Análisis de nuevas features

- Quienes tienen de 0 a 12 y de 13 a 18 años, son mas proclives a ser transportados.
- 19-25, 26-32 y 33-50 son menos transportdoes.
- Los mayores de 50 estan bastante parejos.
- Podemos ver claramente como los pasajeros sin gastos, son altamente transportados a realidades alternas.
- Los que estan en CryoSleep no tienen gastos, se cumple la hipotesis.
- La mayória de los pasajeros son del Deck F y G.
- En Deck T hay muy pocos pasajeros.
- Pasajeros de cabinas del deck B y C, son altamente transportados.
- Respecto al Side, vemos que los pasajeros estan repartidos en mitades, pero que los del lado "S", son mas transportados que los del P.
- La mayoría de los pasajeros viajan solos
- Queines viajan solos son menos transportados a relidades alternas, que quienes viajan en grupo.

Análisis de correlación

Con este analisis multivariado, buscamos observar la relación que tienen las variables entre sí, ademas de cada una de ellas contra el target.

"Gastos totales" / "Sin_gastos" se muestra como la variable mas importe o de mayor relevancia respecto al taraget, le sigue CryoSleep, que a su vez mantiene una alta correlación con la primera, tal como vismo en el EDA, los que viajan en cryo, no gastan. VIP, támbien como se observo en el análisis exploratorio, parece ser le de menor peso.



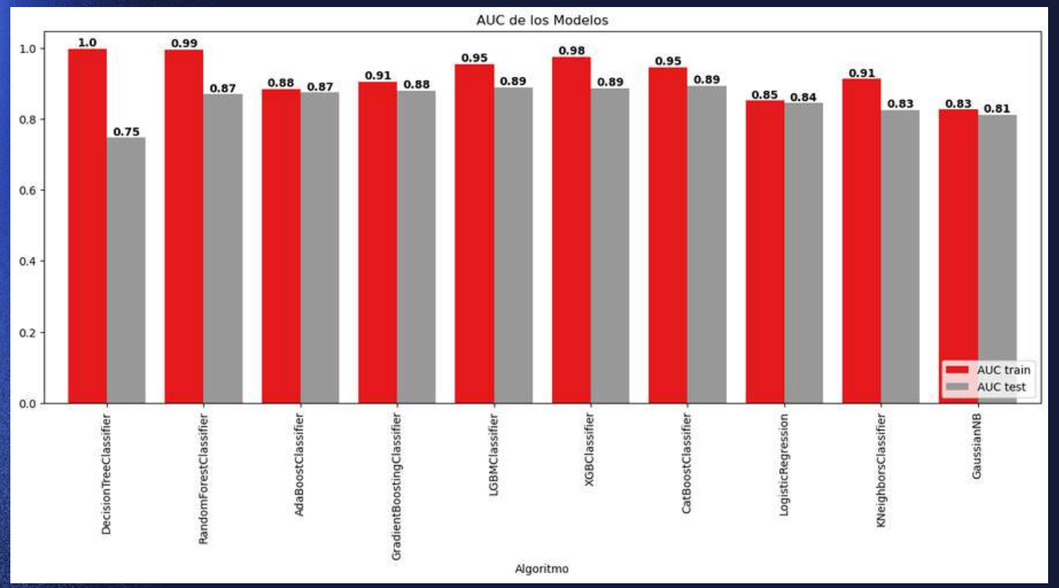
Modelos de Machine Learning

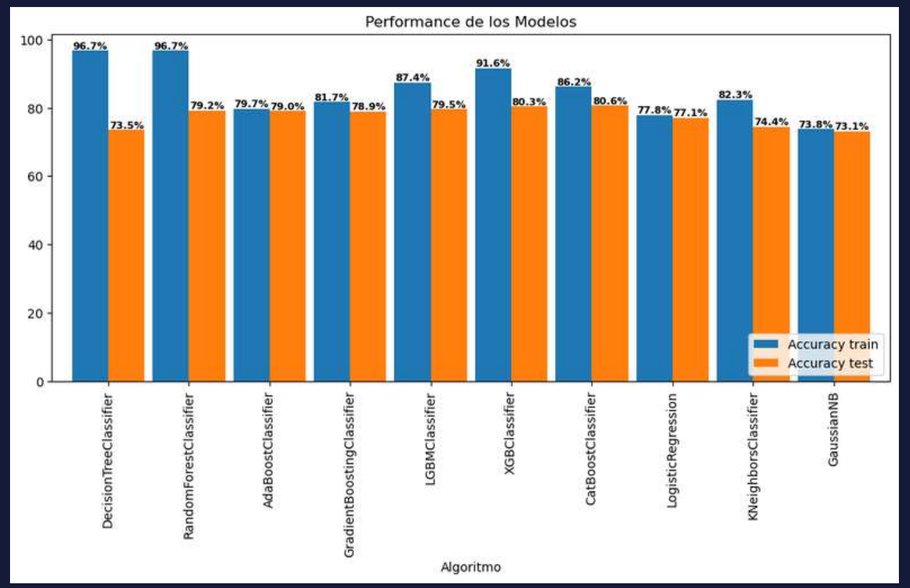
Una vez analizada y procesada toda la data, es decir, features engineering, limpiza de datos, control y manejo de información faltante y outliers, técnicas de encoding y normalización de variables numericas, ya estabamos en condiciones para desarollar distintos modelos de machine learning y poder comparar resultados entre ellos. Elegimos trabajar sobre los algoritmos mas populares de clasificación, es decir que deben predecir por una u otra clase, en este caso True/False, estos son:

- Regresión Logistica
- Decision Tree Classifier
- RandomForest Classifier
- AdaBoost Classifier
- GradientBoosting Classifier
- LGBM Classifier
- XGB Classifier
- CatBoost Classifier
- KNeighbors Classifier
- GaussianNB (Naive Bayes)

Rendimiento de los modelos

	Algoritmo	Accuracy train	Accuracy test	Precision train	Precision test	Recall train	Recall test	F1 train	F1 test	AUC train	AUC test
0	DecisionTreeClassifier	96.740000	73.490000	97.040000	73.240000	96.460000	74.660000	96.750000	73.940000	0.997416	0.747025
1	RandomForestClassifier	96.740000	79.240000	95.680000	80.470000	97.940000	77.630000	96.800000	79.020000	0.994899	0.869547
2	AdaBoostClassifier	79.700000	79.010000	77.430000	76.150000	84.240000	84.930000	80.690000	80.300000	0.884088	0.873701
3	GradientBoostingClassifier	81.670000	78.900000	79.860000	76.650000	85.040000	83.560000	82.370000	79.960000	0.905216	0.879337
4	LGBMClassifier	87.420000	79.530000	85.510000	78.020000	90.320000	82.650000	87.850000	80.270000	0.954677	0.889250
5	XGBClassifier	91.590000	80.330000	89.300000	79.150000	94.630000	82.760000	91.890000	80.920000	0.975494	0.886302
6	CatBoostClassifier	86.190000	80.620000	84.610000	79.140000	88.720000	83.560000	86.620000	81.290000	0.946012	0.893668
7	LogisticRegression	77.850000	77.110000	77.270000	75.980000	79.380000	79.790000	78.310000	77.840000	0.851536	0.844343
8	KNeighborsClassifier	82.330000	74.410000	84.060000	75.620000	80.100000	72.600000	82.030000	74.080000	0.912306	0.825276
9	GaussianNB	73.800000	73.090000	78.770000	77.570000	65.680000	65.530000	71.630000	71.040000	0.826005	0.811592





Optimización de los modelos

- CatBoost Classifier tuvo mejor rendimiento, Accuracy Score y Auc mas alto en test. En train resultaron mayores por lo que hay un poco de sobreajuste a corregir.

- Le siguen XGBoost, LGBM, también RandomForest, aunque este sobre ajustó demasiado en test.

- Los modelos con menor rendimiento fueron GaussianNB y KNN.

- A los algoritmos con mejores resultados, se les aplicaron tecnicas de optimización mediante el ajuste de los hyper-parametros que cada uno de ellos permite. Si bien este metodo no genero un incremento en las metricas sobre el dataframe de testeo, si logro reducir el sobre ajuste que teníamos sobre los datos de entrenamiento.

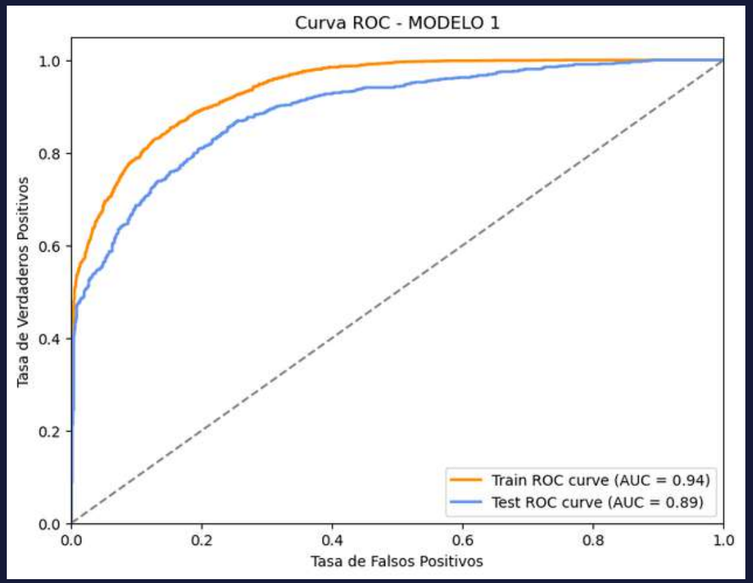
- Finalmente, como dato un plus a lo desarrollado, generamos un stacking model, que combina las predicciones de los mejores algoritmos para obtener una mejor.

Resultados finales

El resultado final del Stacking Model, obtiene 80,51% de Accuracy para Test, y 84%, para train, logrando la mejor relación entre ambos dataframes, para los modelos que superaban el 80%.

Tambien, logro ajustar las Curvas ROC, con un auc de 0.93 para Train y 0.89 para Test.

LGBClassifier, fue el que obtuvo el mejor resultado individual, tras la optimización, con Accuracy de 80,74%



	Algoritmo	Accuracy train	Accuracy test	Precision train	Precision test	Recall train	Recall test	F1 train	F1 test	AUC train	AUC test
10	CatBoostTunning	83.680000	79.930000	81.640000	77.940000	87.210000	83.900000	84.330000	80.810000	0.925971	0.885267
11	LGBMTunning	85.480000	80.740000	83.990000	78.810000	87.920000	84.470000	85.910000	81.540000	0.941680	0.891672
12	XGBTunning	85.390000	80.330000	84.320000	78.710000	87.210000	83.560000	85.740000	81.060000	0.942671	0.890474
13	RandomForestTunning	84.310000	79.300000	83.990000	78.600000	85.070000	80.940000	84.520000	79.750000	0.934524	0.881095
14	Stacking Model	84.830000	80.450000	83.180000	78.760000	87.580000	83.790000	85.320000	81.190000	0.938951	0.891869

Conclusión

- El dataset contenía variables que por si sola, no tenian tanta relevancia, por lo que se crearon nuevas features que permitieron realziar mejores predicciones

- El tratamiento realizado sobre los datos, limpieza, manejo de outliers, trataminto de vacios, y otras transformaciones, permitió no tener que descartar información, pudiendo aprovecar todo el dataset.
- Se utilizaron distitnos algoritmos de clasificación, para obtener las mejores predicciones.

- LGBMClassifier XGBClassifier CatBoostClassifier RandomForestClassifier, fueron los modelos que mejor resultado obtuvieron, pero mostraban sobreajustes.

- Mediante el uso de hyper-parametros, se intentó minimizar los sobreajustes, para finalziar el trabajo ensamblando un Stacking model, que aproveche lo mejor de cada uno de ellos.