Examen Módulo II

Diplomado en Ciencia de Datos

Juan Manuel Zapata López

Introducción

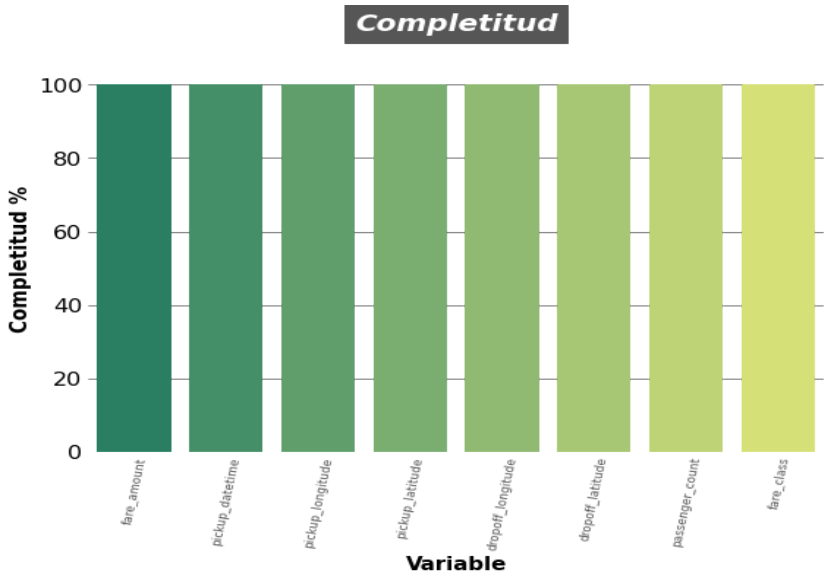
Se cuenta con una tabla que contiene registros de viajes realizados en taxi, con el monto de la tarifa, tipo de tarifa, coordenadas de inicio y fin y número de pasajeros. El objetivo es aplicar tratamientos para analizar la tabla, convertirla en una tabla analítica de datos y finalmente correr modelos de regresión y clasificación.

Calidad de datos

- Se verificó si existían registros duplicados y no se encontraron
- La variable pickup_datetime se convirtió a tipo datetime
- Se hizó un describe de la tabla donde se encontró que había tarifas negativas, latitudes y longitudes fuera de rango, número de pasajeros cero o muy grande (208). Los registros con esas características fueron eliminados.
- Se encontró que la mayoría de coordenadas pertenecen a New York por lo cual se filtraron a esa región.

Completitud

No se encontraron missings en la tabla:



Ingeniería de variables

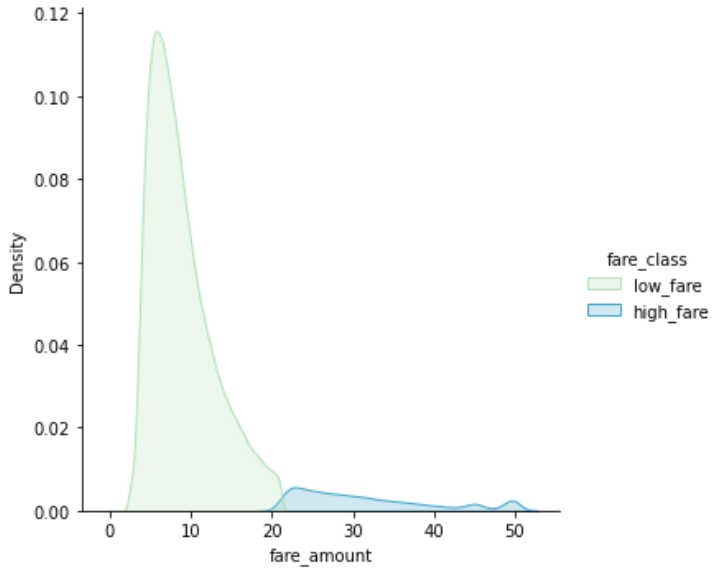
1. Dado que se cuenta con las coordenadas de inicio y término del viaje, se calculó la distancia entre ambos puntos de distintas formas:
    - Distancia Haversine que considera la esfericidad de la Tierra
    - Distancia de Manhattan útil en problemas de movilidad
    - Distancia de manejo, con la api de google maps, esta última se terminó descartando porque en 1 hora sólo había sido calculada para el 6% de la tabla, pero el código quedó comentado

2. A partir de la fecha del viaje, se obtuvieron las siguientes variables:
    - Año
    - Mes
    - Día
    - Hora
    - Día de la semana
    - Variable dummy que indica si es fin de semana

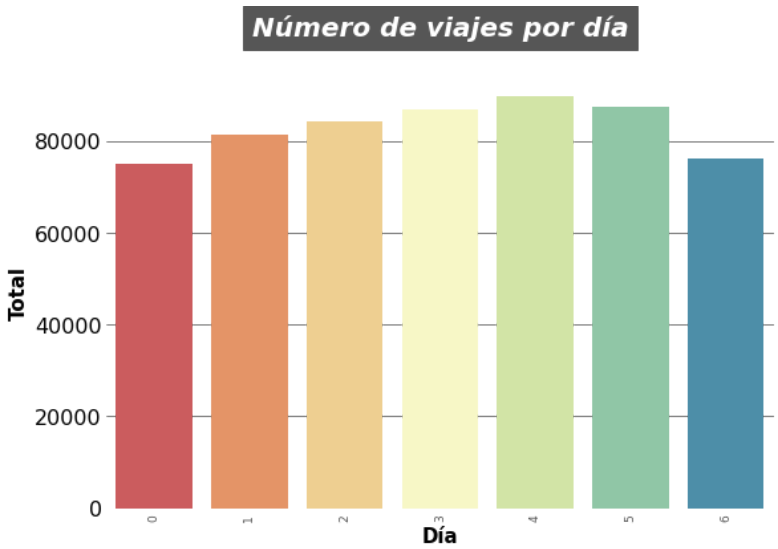
Los registros cuya distancia era cero, también fueron eliminados.

Análisis Exploratorio

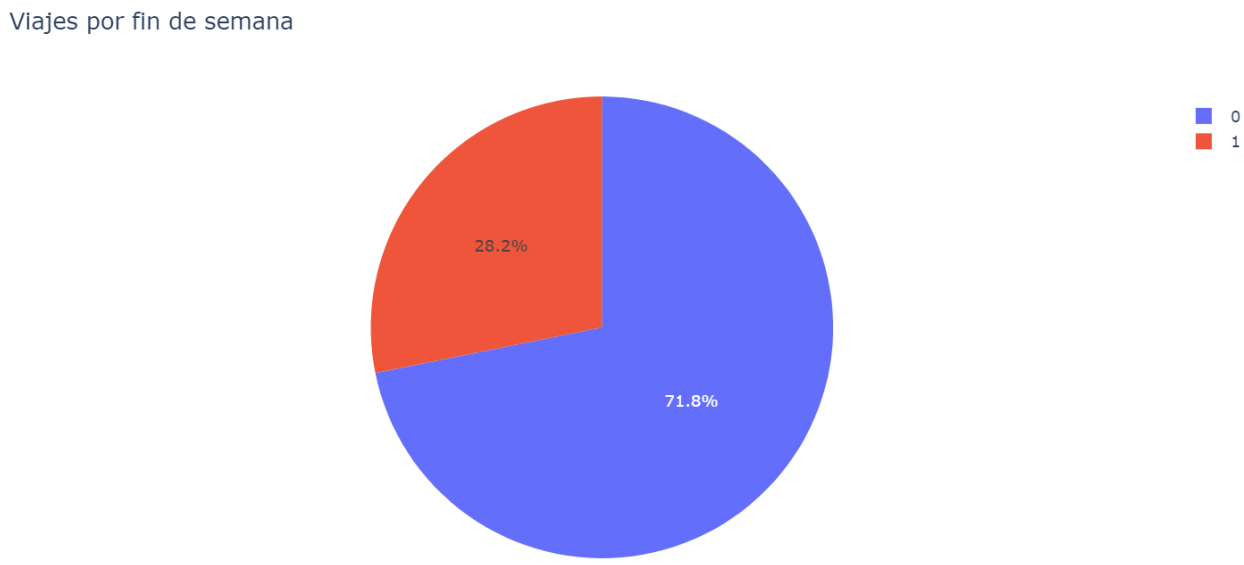
Las tarifas catalogadas como altas son aquellas que tienen un monto mayor a 20 dólares y las tarifas catalogadas como bajas, suelen ser menores respecto al mismo monto:



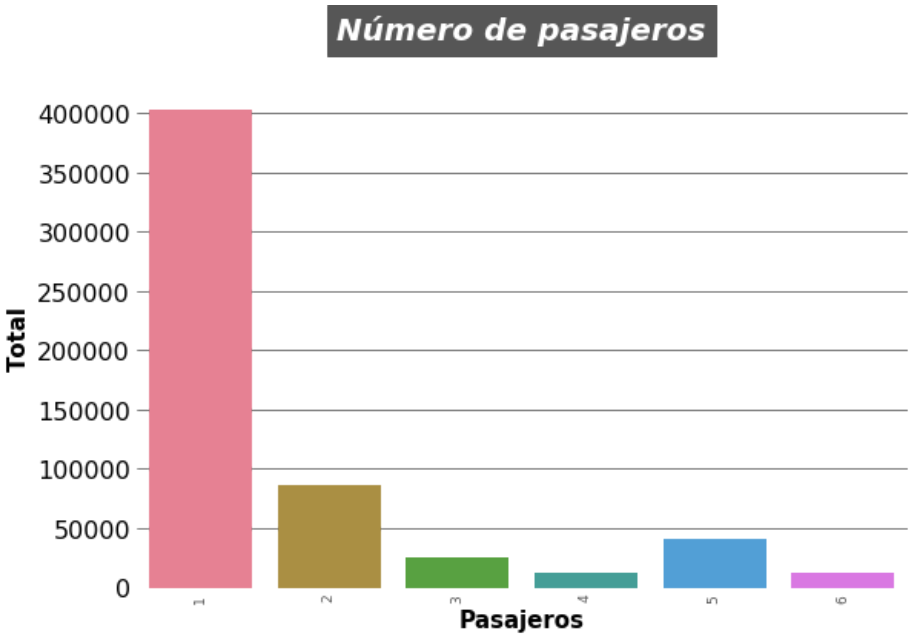
Al graficar el número de viajes por día, notamos que alrededor del viernes suele haber más viajes. (Día 0 = lunes, 1 = martes …). El día que menos viajes hay es domingo.



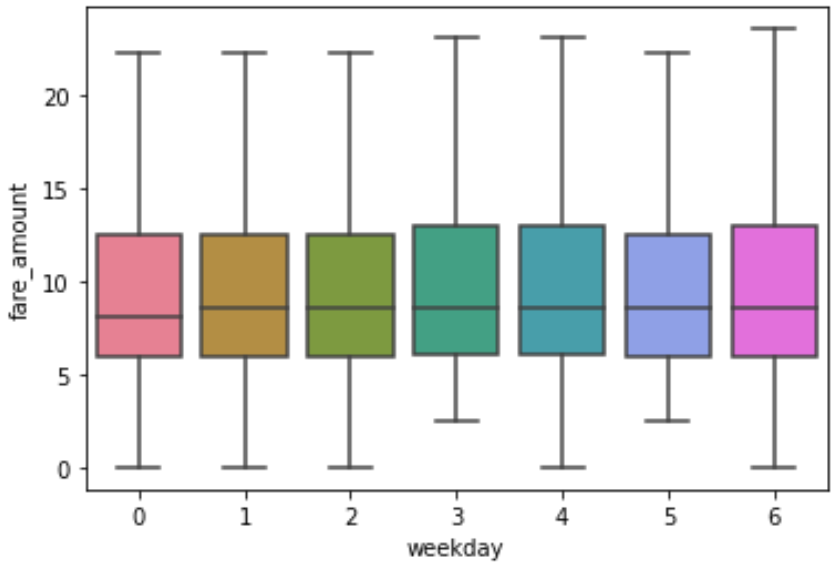
Aunado al hecho anterior, si graficas la distribución de viajes dependiendo de si ocurrieron en fin de semana o no, percibimos que sólo el 28.2% ocurren en sábado o domingo.



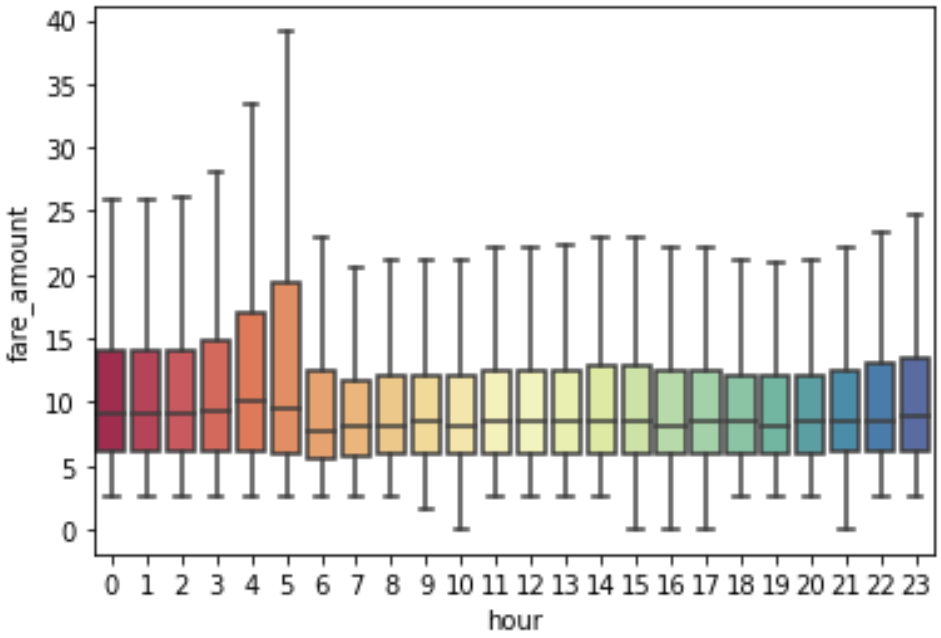
Al graficar el número de viajes, dependiendo del número de pasajeros, podemos notar que la mayoría de viajes sólo transportan a una única persona y que máximo se transportan hasta 6 personas.



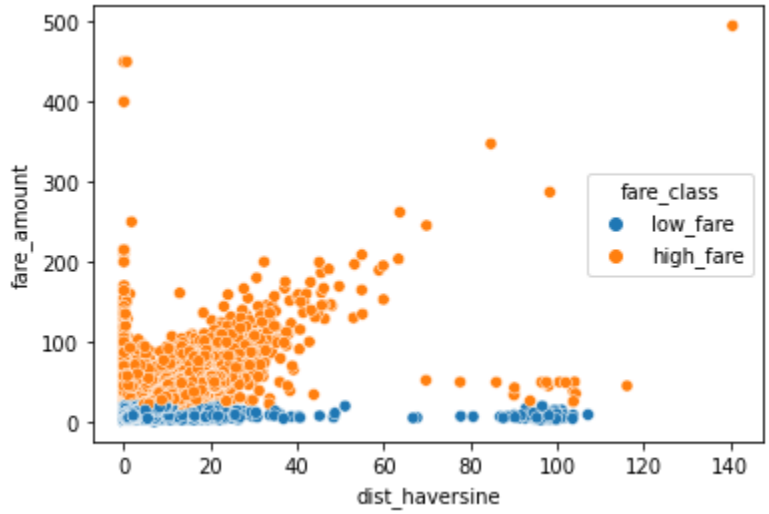
El monto de la tarifa se comporta de forma similar todos los días, con una media alrededor de 7.5 dólares y máximo 25 dólares (sin considerar outliers).

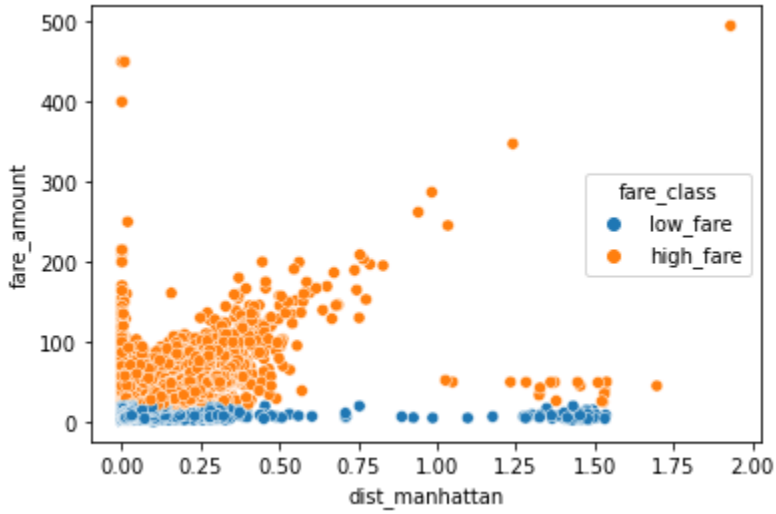


Las tarifas suelen ser más caras en las primeras horas del día (de las 00:00 a las 05:00) y también aumentan un poco en la noche (después de las 21:00).



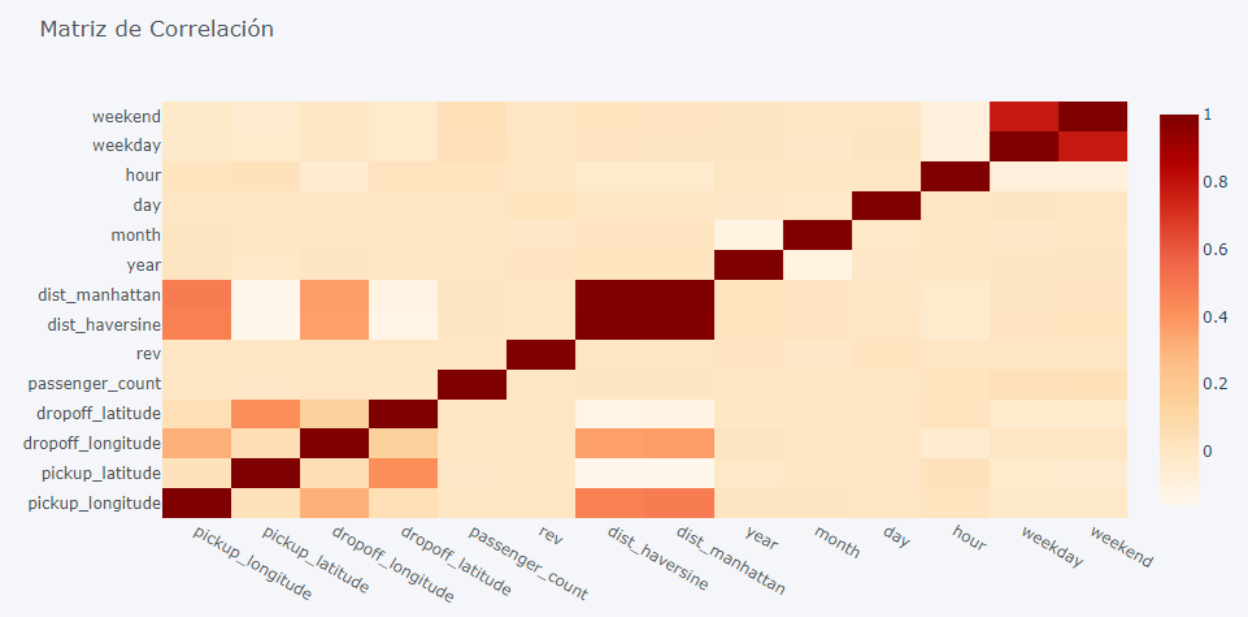
Al realizar scatterplots de las distancias calculadas contra la tarifa, notamos que ambas distancias se comportan de forma muy similar.





Reducción de dimensiones

1. No se encontraron variables con varianza insignificante
2. Para las variables explicativas en ambos tipos de modelo, se encontraron las siguientes correlaciones:



3. Se calculó la correlación con la variable objetivo continua

	fare_amount
fare_amount	1.000000
dist_haversine	0.854027
dist_manhattan	0.841568
year	0.149564
dropoff_longitude	0.112477
dropoff_latitude	0.089743
pickup_latitude	0.078922
pickup_longitude	0.044396
month	0.029501
passenger_count	0.022326
weekday	0.012922
hour	0.005725
weekend	0.002313
day	0.000126

Por los análisis anteriores, se decidió eliminar las variables dist_manhattan, weekday, dropoff_latitude,pickup_latitude, pickup_longitude, dropoff_longitude y day.

4. Para evitar problemas de multicolinealidad, se calculó el VIF de las variables restantes y se decidió eliminar a year:

	variables	VIF
2	year	10.844630
4	hour	5.351588
3	month	4.336257
0	passenger_count	2.681282
5	weekend	1.406931
1	dist_haversine	1.002838

Detección de Outliers

Para este apartado, se intentó eliminar outliers de forma univariada para las variables de la TAD, pero se eliminaban muchos registros de “tarifa alta”, por lo cual se optó por usar la metodología multivariada de local outlier factor, con lo que se eliminaron 11,587 registros (menos del 2% del total) y también se logró mantener el balanceo de la variable fare_class.

De todos los puntos anteriores, se obtuvo una TAD lista para correr tanto el problema de regresión como el de clasificación, es decir, las ingenierías y procesos para ambos algoritmos fueron los mismos. También es importante mencionar que se hizo random search en ambos casos para la obtención de hiperparametros.

Modelo de regresión

Para poder correr los algoritmos más pesados de este tipo, se tuvo que tomar 100,000 registros al azar de la TAD, se corrieron los siguientes modelos: Regresión lineal, regresión lasso, regresión ridge, red elástica, árbol de decisión y adaboost.

Finalmente, se decidió modelar con el árbol de decisión, pues tuvo las mejores métricas y también tenemos interpretabilidad.

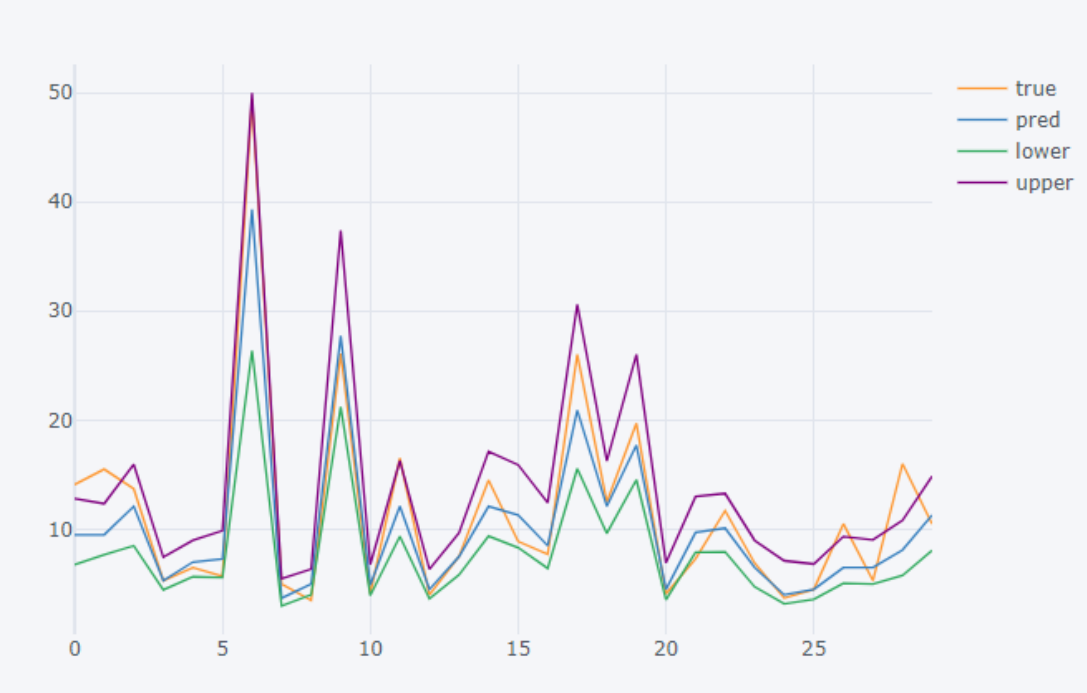
Metricas del train:

MAE	MSE	RMSE	R2	R2_ADJ
2.044754	17.298396	4.159134	0.809339	1.00001

Metricas del test:

MAE	MSE	RMSE	R2	R2_ADJ
2.286411	24.035978	4.90265	0.731126	1.000054

Para la estabilidad se tomaron sólo a los primeros 30 registros, para poder visualizar el modelo, los datos reales y las regresiones cuantilicas al .05% y .95%:



Modelo de Clasificación

En este caso, con la ayuda de una pc gamer se logró correr los modelos con toda la TAD. Se comenzó verificando el balanceo de la variable a predecir:

```
y_train.value_counts(1)
```

executed in 15ms, finished 16:24:14 2021-04-19

```
0    0.90591
1    0.09409
Name: fare_class, dtype: float64
```

```
y_test.value_counts(1)
```

executed in 15ms, finished 16:24:14 2021-04-19

```
0    0.905916
1    0.094084
Name: fare_class, dtype: float64
```

Como se aprecia, la clase positiva (high_fare) representa menos del 10% de la información presente en el dataset, inicialmente se pensó en aplicar técnicas de resampling para mejorar el balanceo; pero al correr los modelos sin aplicar esto último, se obtuvieron buenas métricas y se decidió no rebalancear.

Los modelos probados son k - vecinos más cercanos, árbol de decisión, random forest y adaboost. En este caso todos los algoritmos fueron muy buenos y sus métricas son muy parecidas. El modelo con la mejor estabilidad es KNN, sin embargo, no se eligió como el mejor modelo por el tiempo que demora en hacer predicciones.

El modelo elegido como mejor es, al igual que en regresión, un árbol de decisión, las métricas obtenidas son las siguientes:

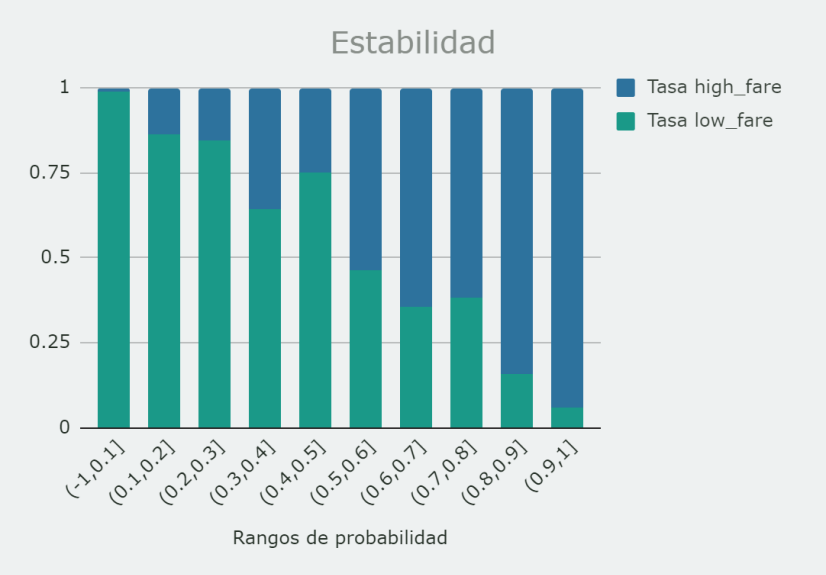
En train:

Accuracy	Precision	Recall	F1	TPR	FPR	AUC
0.966888	0.877518	0.753217	0.81063	0.753217	0.010919	0.972567

En test:

Accuracy	Precision	Recall	F1	TPR	FPR	AUC
0.966094	0.87014	0.751825	0.806667	0.751825	0.011653	0.969897

Y la estabilidad del modelo se ve as así:



Conclusiones

- Al tener grandes volúmenes de información, correr modelos se vuelve un reto en equipos de cómputo convencionales, para este tipo de problemas es bueno usar equipos profesionales o tomar sólo una porción de los datos.
- Aunque los árboles de decisión suelen ser algoritmos que sobreajustan, en algunas ocasiones no lo hacen y son buenos para elegir como mejor modelo por su sencillez e interpretabilidad.
- El balanceo de clases no siempre es necesario, siempre y cuando se tengan suficientes ejemplos.