

# Spaceship Titanic Data Science Project

		Por Juan Manuel González Kapnik
--	--	---------------------------------





# Tabla de contenidos

## Introducción al Proyecto

01

- a. Descripción del proyecto
- b. Set de datos
- c. Tecnologías y librerías utilizadas

## Análisis Exploratorio de Datos (*EDA*)

03

- a. Porcentaje inicial de transportados
- b. Distribuciones
  - i. Edades
  - ii. Gastos
  - iii. Datos Categoricos

## Entendimiento Básico de los Datos

02

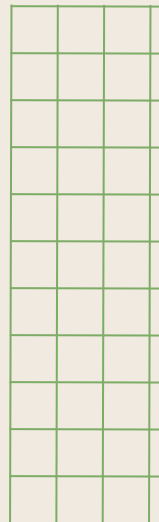
- a. Diferencia entre los diferentes set de datos
- b. Búsqueda de valores faltantes (*missing values*)
- c. Cardinalidades

## *Feature Engineering*

04

Nuevas categorías en base al

- a. *ID*
- b. Cabina
- c. Edad
- d. Gastos





# Tabla de contenidos

## Pre-Procesamiento de los Datos

05

- a. Manejo de *missing values*
- b. Transformación logarítmica
- c. *Hot & Label Encoding*

## Modelo para Datos No Escalables

07

- a. Arbol de decision
- b. *Random Forest*
- c. *Ada Boost*
- d. *Gradient Boost*
- e. *XGB*
- f. *CatBoost*

## Modelo para Datos Escalables

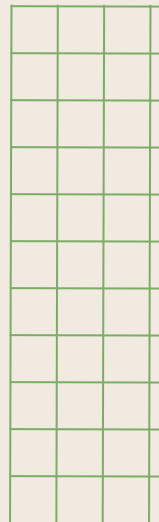
06

- a. Regresion Logistica
- b. *K-Neighbors*
- c. *SVM*
- d. *Naive Bayes*

## Comparación de Modelos

08

- a. Los más efectivos





01

# Introducción al Proyecto





# Descripción del proyecto

La nave espacial Titanic colisionó con una anomalía del espacio-tiempo oculta en una nube de polvo. Por desgracia, casi la mitad de los pasajeros fueron transportados a una dimensión alternativa. Para ayudar a recuperar a los pasajeros perdidos, se le reta a predecir qué pasajeros fueron transportados por la anomalía utilizando los registros recuperados del sistema informático dañado de la nave espacial.





# Set de datos

## *train.csv*

Registros personales de aproximadamente dos tercios (~8700) de los pasajeros, que se utilizarán como datos de entrenamiento.

- *PassengerId* - Un Id único para cada pasajero. Cada Id tiene la forma gggg\_pp donde gggg indica el grupo con el que viaja el pasajero y pp es su número dentro del grupo
- *HomePlanet* - El planeta del que partió el pasajero
- *CryoSleep* - Indica si el pasajero ha decidido permanecer en animación suspendida durante el viaje
- *Cabin* - El número del camarote en el que se aloja el pasajero. Adopta la forma cubierta/num/lado, donde lado puede ser P para babor o S para estribor
- *Destination* - El planeta al que desembarcó el pasajero
- *Age* - Edad del pasajero
- *VIP* - Si el pasajero ha pagado por un servicio VIP especial durante el viaje.
- *RoomService, FoodCourt, ShoppingMall, Spa, VRDeck* - Cantidad que el pasajero ha facturado en cada uno de los muchos servicios de lujo
- *Name* - Nombre y apellido del pasajero
- *Transported* - Si el pasajero fue transportado a otra dimensión





# Set de datos

*test.csv*

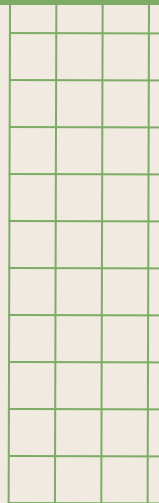
Registros personales para el tercio restante (~4300) de los pasajeros, que se utilizarán como datos de prueba. La tarea consiste en predecir el valor de Transportado para los pasajeros de este conjunto.





# Tecnologías y librerías utilizadas

- *Jupyter Lab* - Editor de código fuente
- *Jupyter Notebook* - Cuaderno como entorno de *Python*
- *Python3* - Lenguaje de programación
- Librerías
  - *Pandas*
  - *Numpy*
  - *Matplotlib.pyplot*
  - *Seaborn*
  - *Missingno*
  - *Sklearn*
  - *Lightgbm*
  - *Xgboost*
  - *Catboost*



02

# Entendimiento Básico de los Datos



# Diferencia principal entre ambos data set

El dataset de entrenamiento contiene la columna de transportados que intentaremos predecir en el set de testing:

## **Transported**

---

False

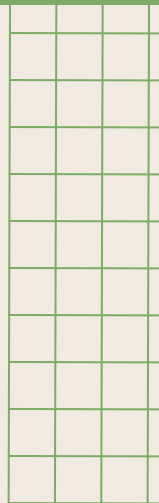
True

False

False

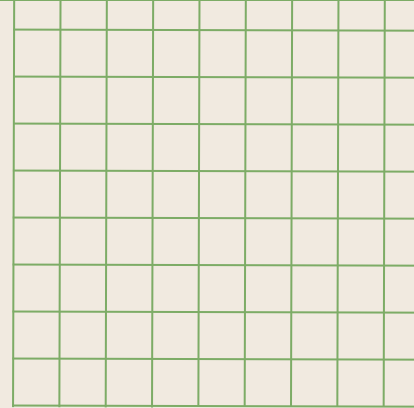
True

...



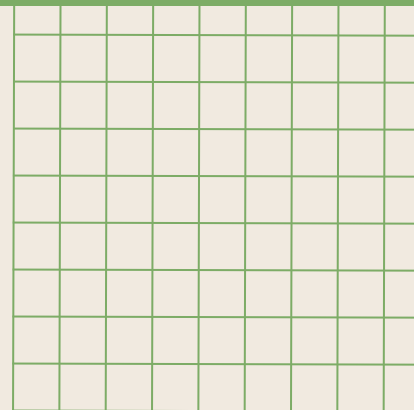
# Disposición inicial: valores perdidos

train.csv	No. of Missing Values	% of Missing Values
PassengerId	0	0
HomePlanet	201	4.7
CryoSleep	217	5.07
Cabin	199	4.65
Destination	182	4.26
Age	179	4.19
VIP	203	4.75
RoomService	181	4.23
FoodCourt	183	4.28
ShoppingMall	208	4.86
Spa	183	4.28
VRDeck	188	4.4
Name	200	4.68
Transported	0	0



# Disposición inicial: valores perdidos

test.csv	Number of Missing Values	% of Missing Values
PassengerId	0	0
HomePlanet	87	2.03
CryoSleep	93	2.17
Cabin	100	2.34
Destination	92	2.15
Age	91	2.13
VIP	93	2.17
RoomService	82	1.92
FoodCourt	106	2.48
ShoppingMall	98	2.29
Spa	101	2.36
VRDeck	80	1.87
Name	94	2.2



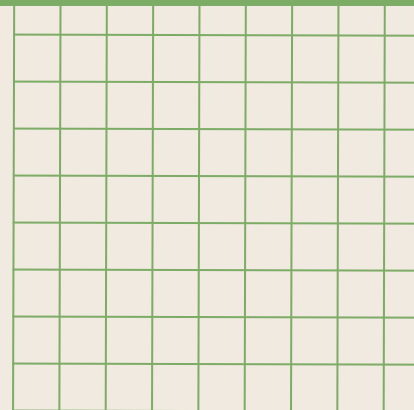
1. ¿La cantidad es alta?
2. ¿Qué podemos hacer al respecto?



# Disposición inicial: Cardinalidades

train.csv	Cardinality
PassengerId	8693
HomePlanet	3
CryoSleep	2
Cabin	6560
Destination	3
VIP	2
Name	8473

test.csv	Cardinality
PassengerId	4277
HomePlanet	3
CryoSleep	2
Cabin	3265
Destination	3
VIP	2
Name	4176



1. ¿La cantidad es alta?
2. ¿Qué podemos hacer al respecto?



# 03

## Análisis Exploratorio de Datos (*EDA*)

Dentro del set de entrenamiento

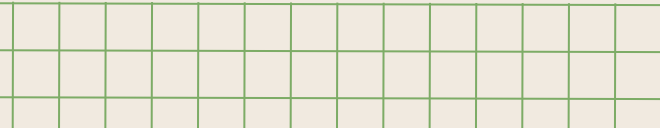


# 50.36%

## De las personas fueron transportadas

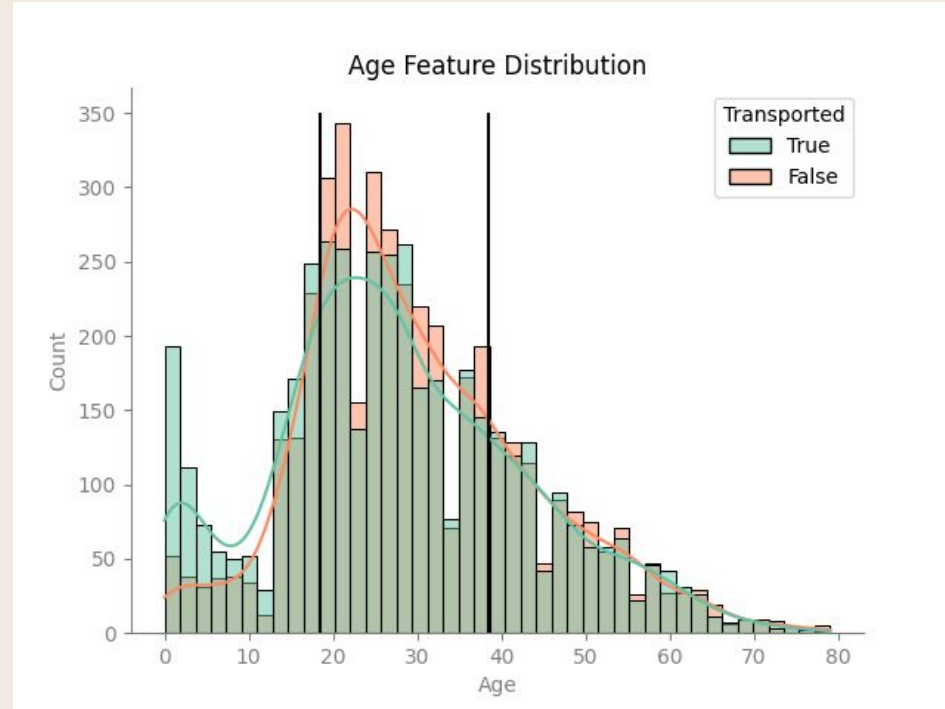


1. ¿Esta balanceado?
2. ¿Necesitamos hacer over o under sampling?



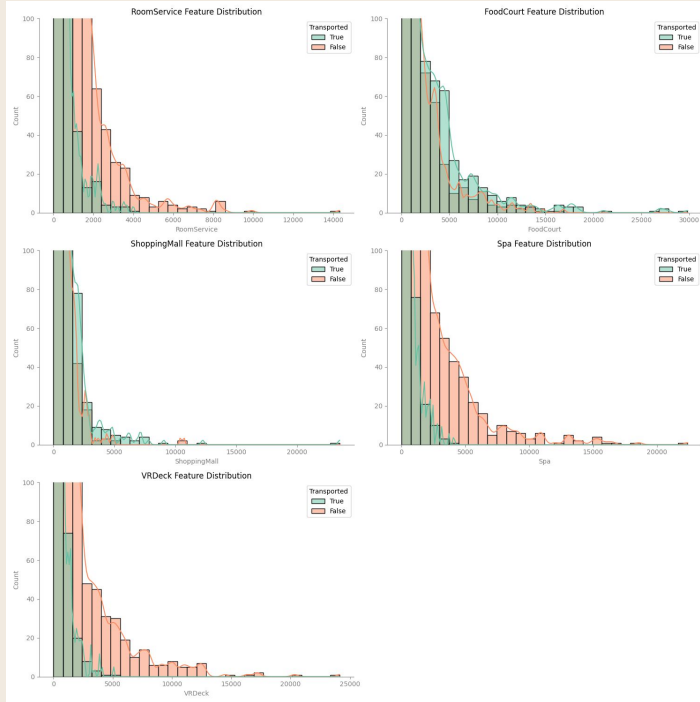


# Distribución de Edades



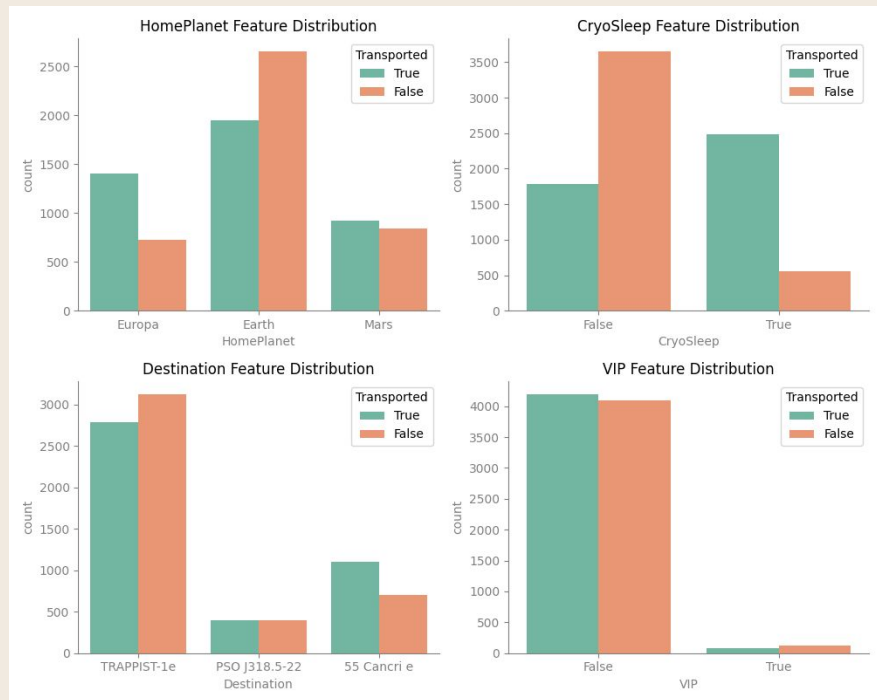
- La mayoría de pasajeros son de edades entre 18 y 32 años
- Pasajeros de entre 0 y 18 años fueron altamente transportados, especialmente recién nacidos
- Pasajeros entre 18 y 38 años fueron menos transportados
- Pasajeros +38 fueron transportados equitativamente

# Distribución de Gastos



- La mayoría de pasajeros no gastaron dinero
- RoomService, Spa y VRDeck tienen distribuciones similares
- Todas las distribuciones presentan asimetría estadística negativa. Por lo tanto, las transformaremos en distribuciones normales mediante la transformación logarítmica
- Aquellos pasajeros que gastaron menos tendieron a ser más transportados. Por lo tanto, podemos generar una nueva categoría que dicte si el pasajero gastó o no
- Podemos generar una categoría que indique el total gastado para cada pasajero, y dividirlo en diferentes tipos de gastos como bajos, medios y altos

# Distribución de Datos Categóricos



- **HomePlanet.** La mayoría de pasajeros vienen de la tierra (mayormente no transportados). Pasajeros de Marte fueron igualmente transportados, y pasajeros de Europa fueron mayormente transportados
- **CryoSleep.** Aquellos que no fueron criogenizados no fueron mayormente transportados, y aquellos que fueron criogenizados fueron mayormente transportados
- **Destination.** La mayoría tuvieron destino a Trappist-1e (mayormente no transportados). Aquellos con destino 55 Cancri e fueron mayormente transportados. Aquellos con destino PSO J318.5-22 fueron equitativamente transportados
- **VIP.** La mayoría de pasajeros no fueron VIP, pero tanto en aquellos que fueron como los que no tienen una distribución equitativa

# 04

## ***Feature Engineering***



Dentro del set de entrenamiento



# ¿Cómo realizamos *feature engineering* en el ID del pasajero?



Cada pasajero tiene un ID con la forma *gggg\_pp* donde:

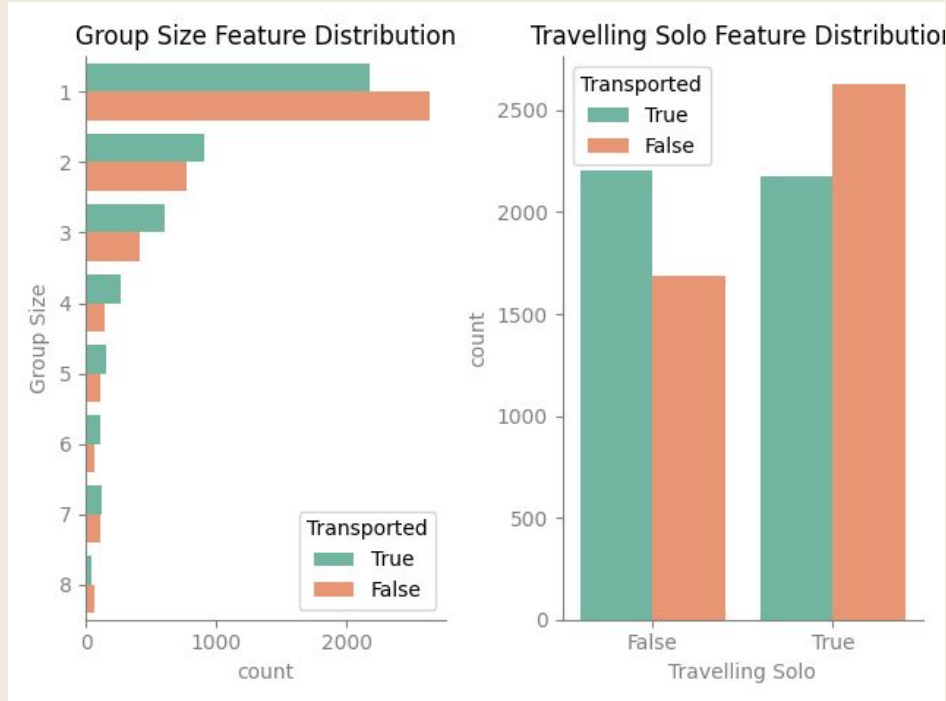
- *gggg* indica el grupo con el que viaja
- *pp* es el número de persona del grupo

Podemos generar dos nuevas categorías: una que indique el número de personas por grupo, y otra que indique si el pasajero está viajando solo

train.csv	PassengerId
0	0001_01
1	0002_01
2	0003_01
3	0003_02
4	0004_01



# Distribución de Tamaños de Grupos y Solo



- Según los tamaños de los grupos, la mayoría de los pasajeros viajaron solos (mayormente no transportados)
- Aquellos que viajaron con más pasajeros tendieron a ser más transportados

# ¿Cómo realizamos *feature engineering* en la cabina del pasajero?



Cada pasajero tiene un ID con la forma  
*cubierta/numero/lado* donde:

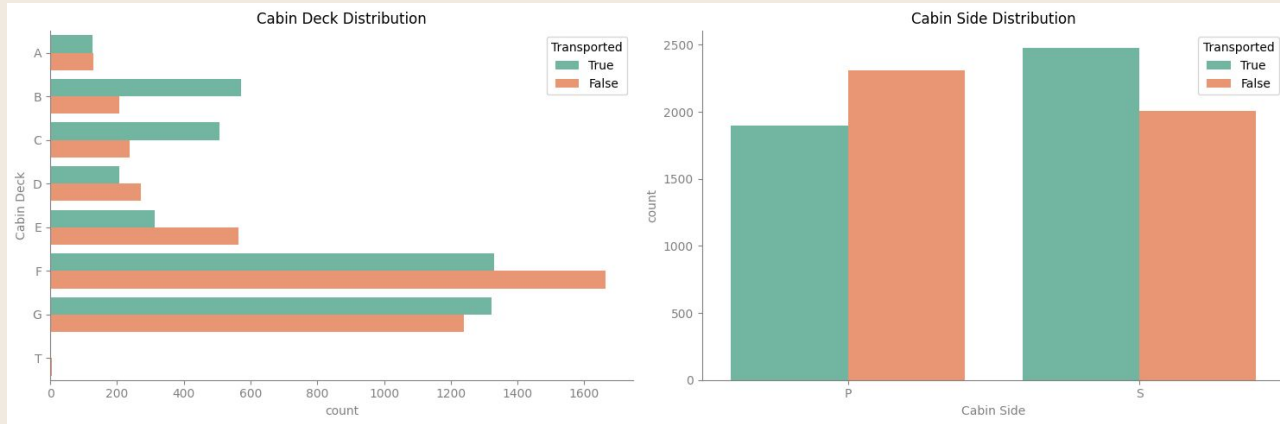
- *cubierta* indica la localización
- *número* de la cubierta
- *lado* puede ser *P* en caso de puerto, y *S* para estribor (*starboard*)

Podemos generar tres nuevas categorías, una para cada ítem

train.csv	Cabin
0	B/O/P
1	F/O/S
2	A/O/S
3	A/O/S
4	F/I/S



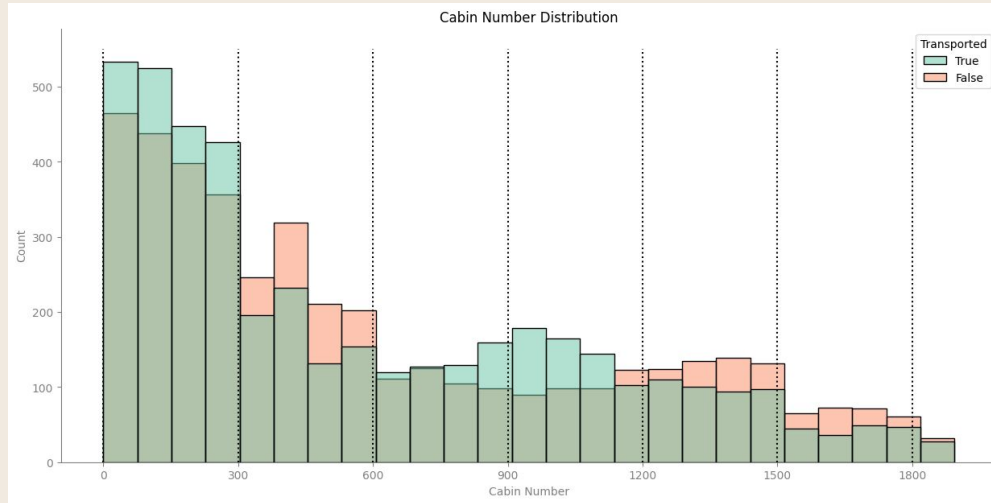
# Distribución de Cubiertas y Lados



- De las cabinas observamos que la mayoría de pasajeros son de las cabinas *F* (mayormente no transportados) y *G* (equitativamente transportados). En la cabina *A* fueron equitativamente transportados. En la cabina *B* y *C* fueron mayormente transportados. En la cabina *D* y *E* fueron mayormente no transportados. Muy pocos pasajeros en la cabina *T*
- Casi la misma cantidad de pasajeros en el lado *P* (mayormente no transportados) y *S* (mayormente transportados)



# Distribución de Números de Cabina

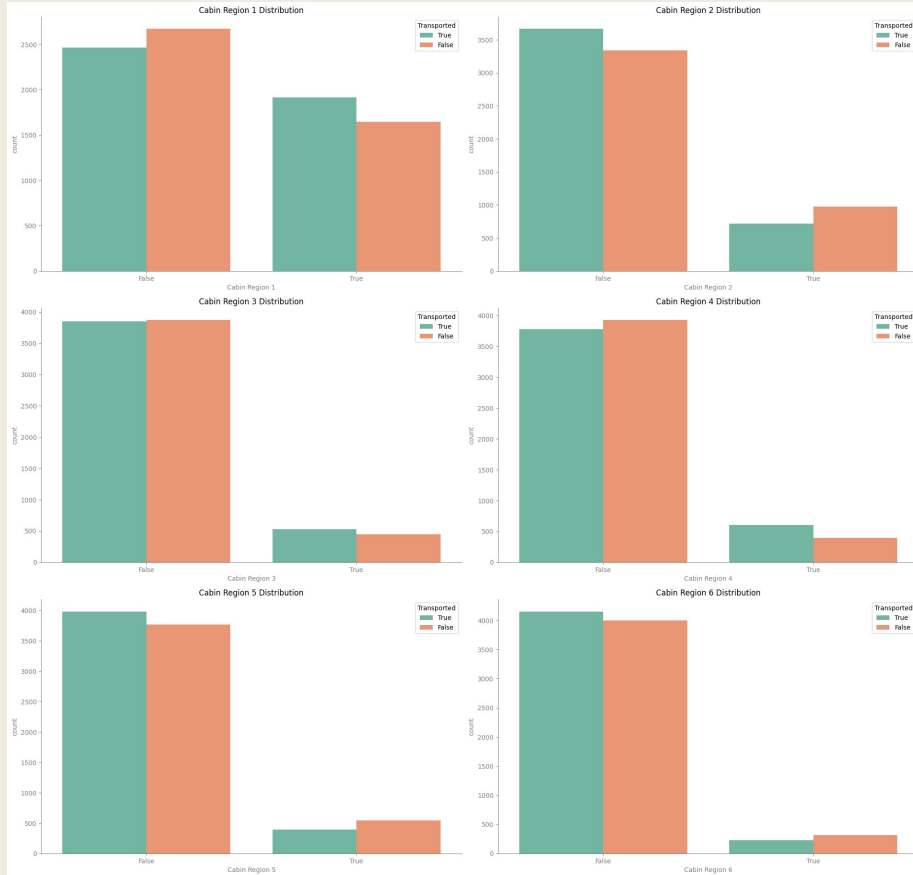


- Según el histograma, cada ~300 hay un cambio entre mayormente transportados y no. Por lo tanto, podemos generar 6 regiones que indiquen en qué número se encuentra el pasajero

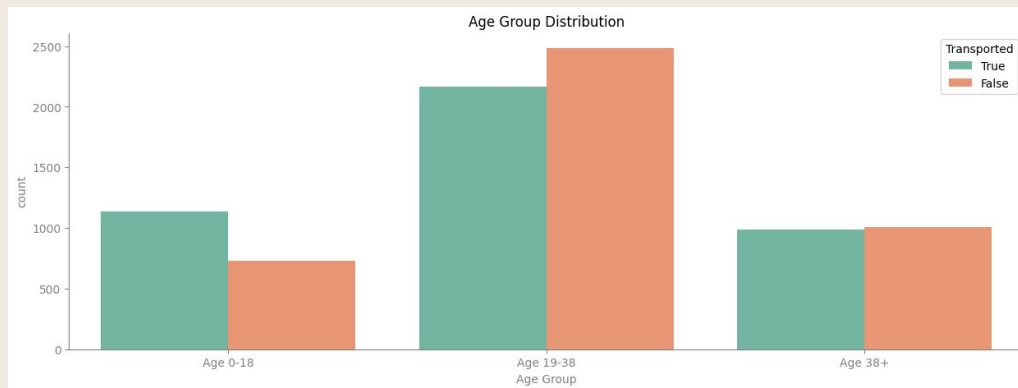


# Distribución de Regiones

- La mayoría se encuentran en la región 1, y fueron los más transportados
- En las regiones 2, 5 y 6 fueron mayormente no transportados
- En las regiones 3 y 4 fueron mayormente transportados (ademas de la region 1)

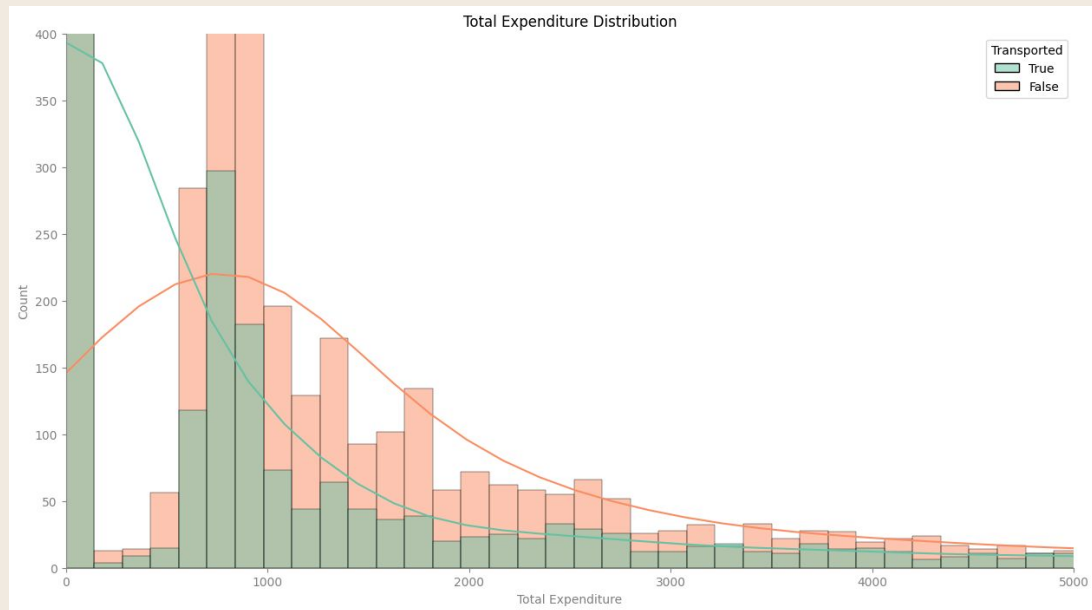


# Distribución de Grupos de Edades



- La mayoría de pasajeros son de edades entre 18 y 32 años
- Pasajeros de entre 0 y 18 años fueron altamente transportados, especialmente recién nacidos
- Pasajeros entre 18 y 38 años fueron menos transportados
- Pasajeros +38 fueron transportados equitativamente

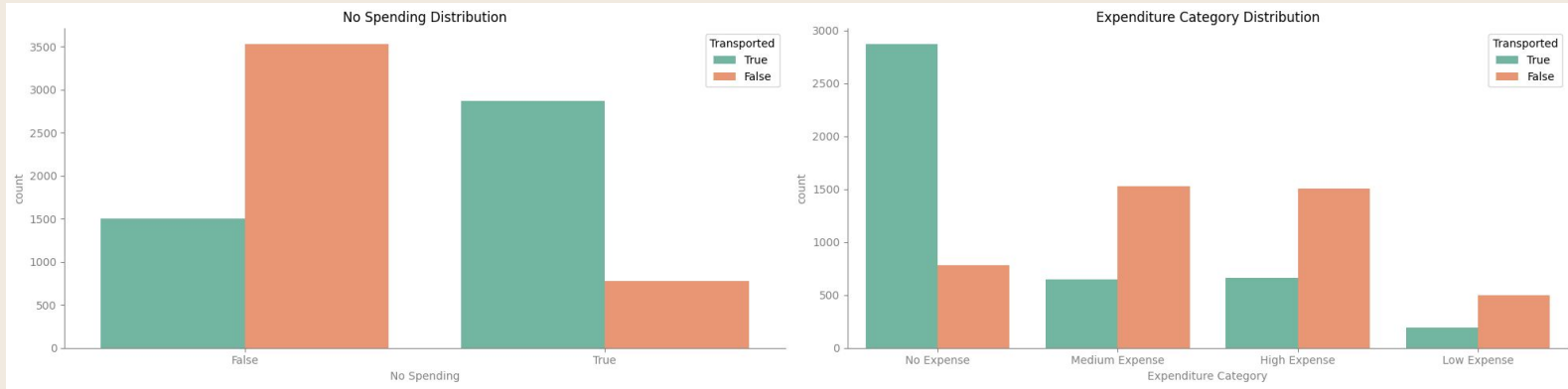
# Distribución de Gastos Totales



- Aquellos pasajeros que no gastaron fueron mayormente transportados, y aquellos que gastaron fueron mayormente no transportados

train.csv	Percentil
25%	0
50% (mediana)	716
75%	1441

# Distribución de No gastos y de Categorías de gastos



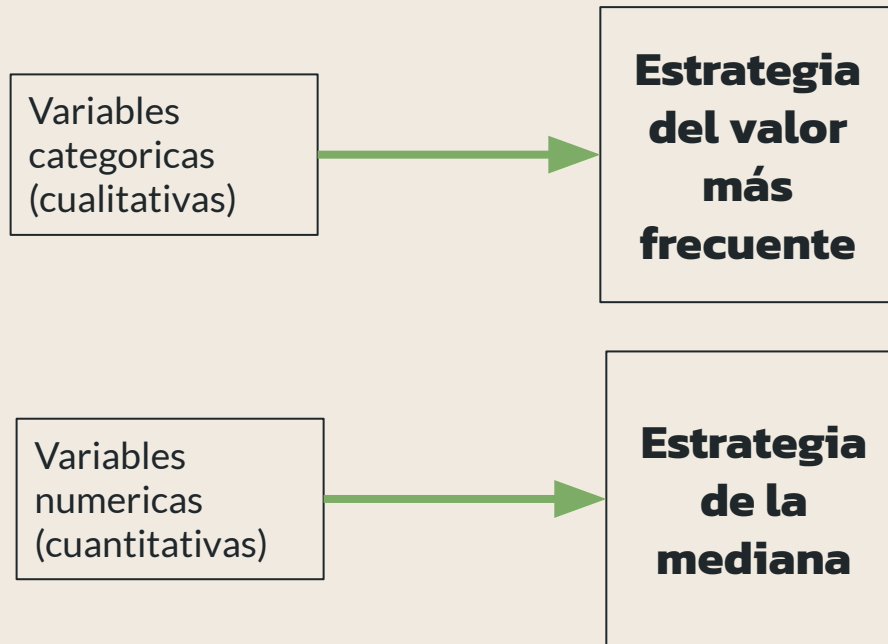
05

# Pre-Procesamiento de los Datos



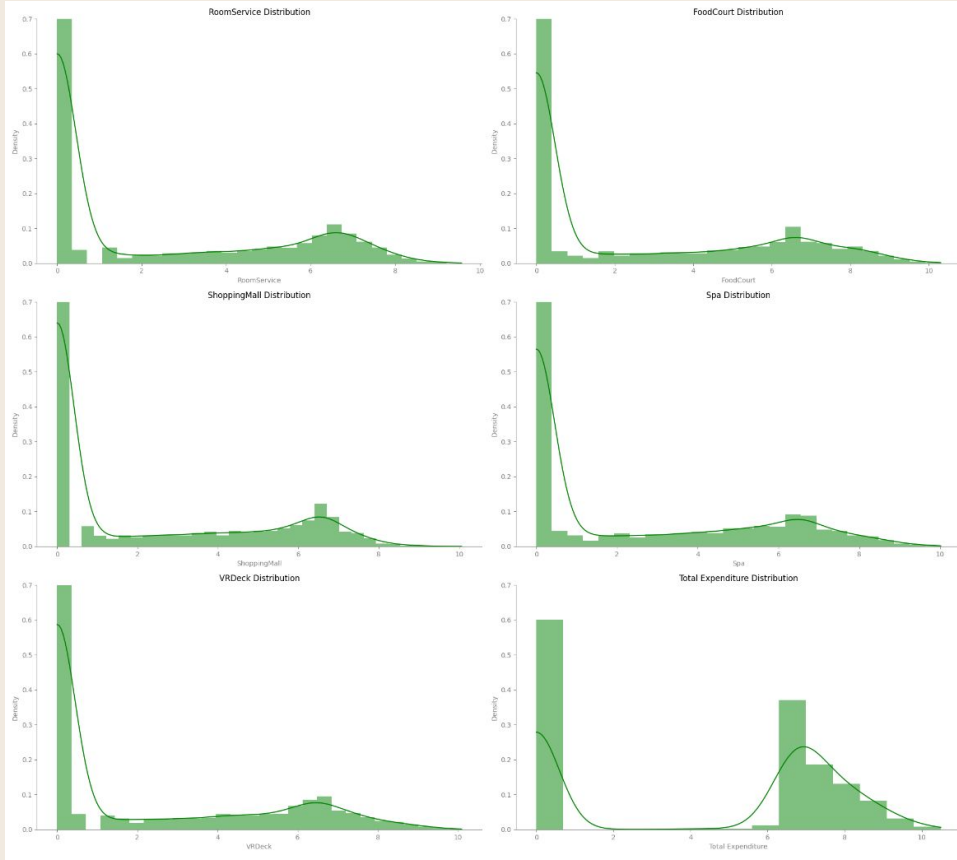


# Manejo de missing values





# Distribución de Gastos (transformación logarítmica)







# Encoding para variables categoricas

Variables  
categoricas  
nominales  
(clasificación)



**Hot  
Encoding**

Variables  
categoricas  
ordinales  
(ordenadas)



**Label  
encoding**

06

# Construcción de Modelos para Datos Escalables



# Resultados del modelo de Regresión Logística

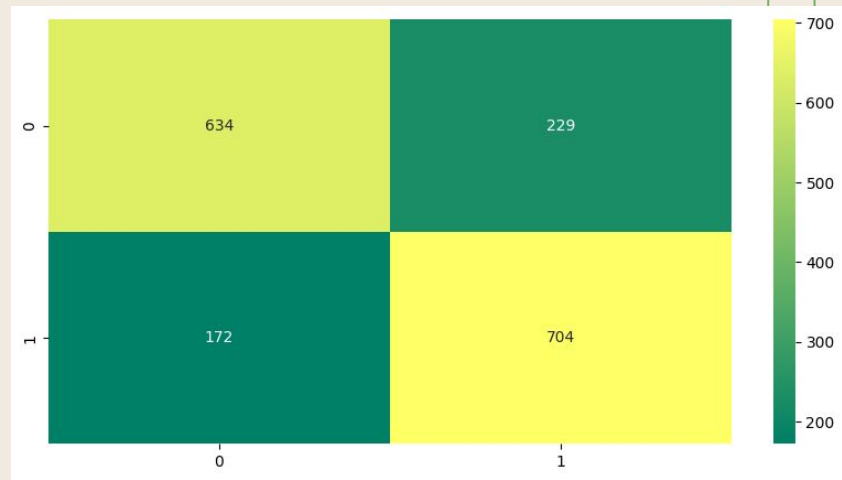
**Precision en Training Data** 77.95

**Precision en Testing Data** 76.94

**Precision** 0.75

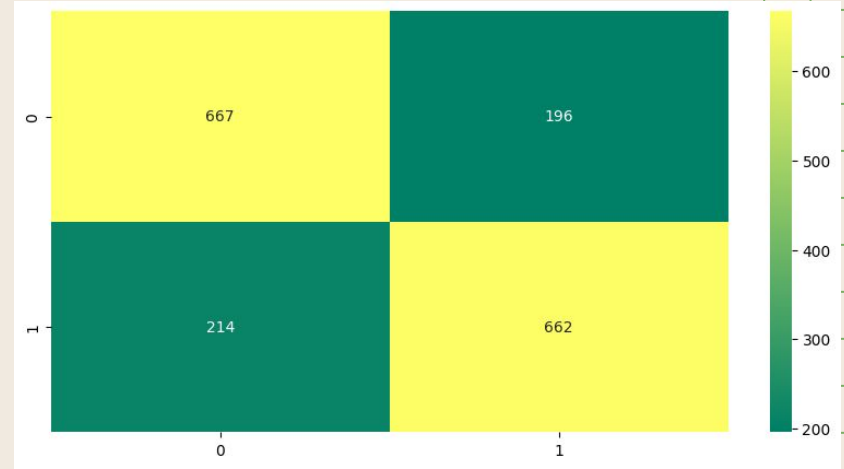
**Recall** 0.80

**F1 Score** 0.78



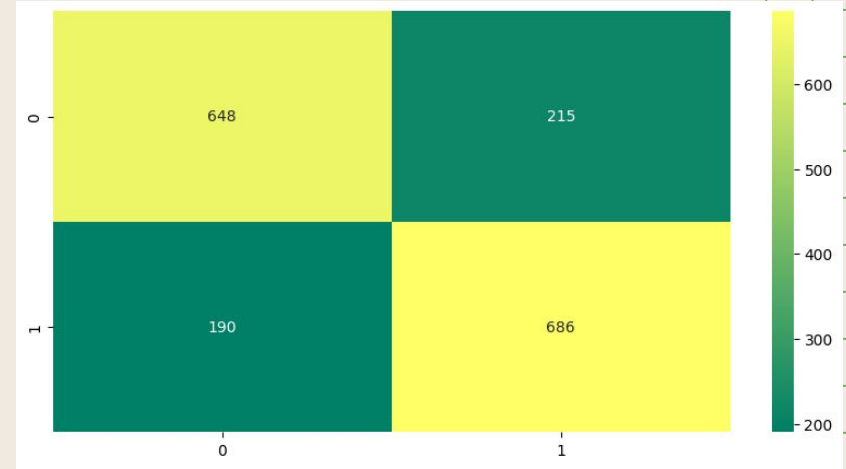
# Resultados del modelo de KNeighborsClassifier

<b>Precision en Training Data</b>	83.29
<b>Precision en Testing Data</b>	76.42
<b>Precision</b>	0.77
<b>Recall</b>	0.75
<b>F1 Score</b>	0.76



# Resultados del modelo de Support Vector Classifier

<b>Precision en Training Data</b>	77.42
<b>Precision en Testing Data</b>	76.71
<b>Precision</b>	0.76
<b>Recall</b>	0.78
<b>F1 Score</b>	0.77



# Resultados del modelo de Naïve Bayes

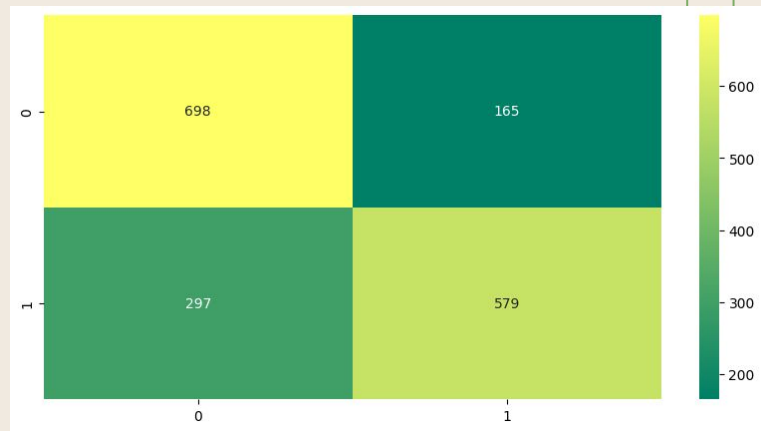
**Precision en Training Data** 73.95

**Precision en Testing Data** 73.45

**Precision** 0.77

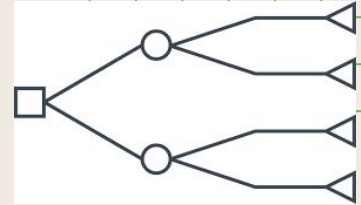
**Recall** 0.66

**F1 Score** 0.71



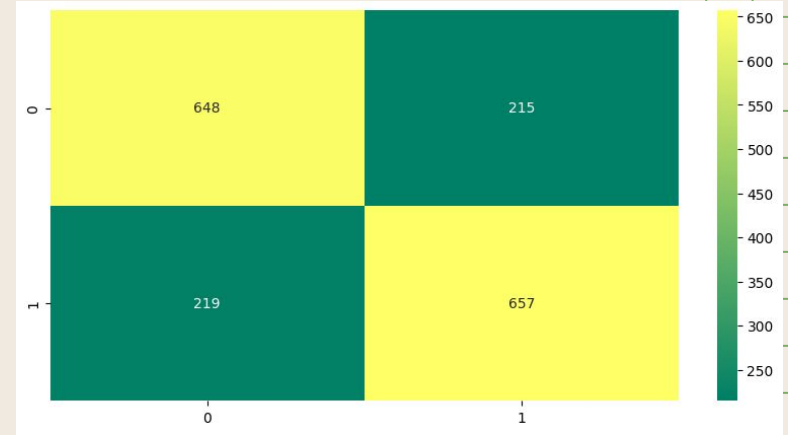
07

# Construcción de Modelos para Datos No Escalables



# Resultados del modelo de Árbol de Decisión

<b>Precision en Training Data</b>	98.47
<b>Precision en Testing Data</b>	75.04
<b>Precision</b>	0.75
<b>Recall</b>	0.75
<b>F1 Score</b>	0.75





# Resultados del modelo de Random Forest Classifier

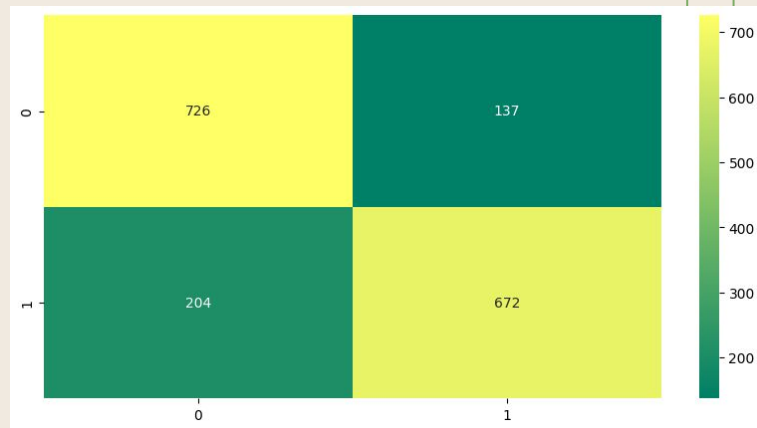
**Precision en Training Data** 98.46

**Precision en Testing Data** 80.39

**Precision** 0.83

**Recall** 0.76

**F1 Score** 0.79



# Resultados del modelo de Adaboost Classifier

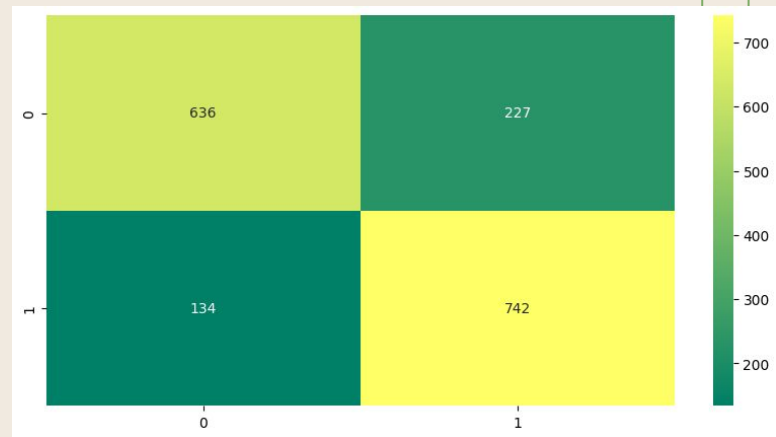
**Precision en Training Data** 79.98

**Precision en Testing Data** 79.24

**Precision** 0.76

**Recall** 0.84

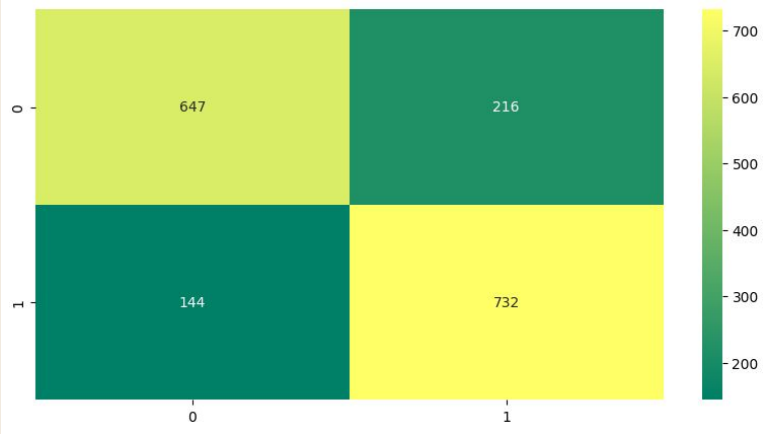
**F1 Score** 0.80





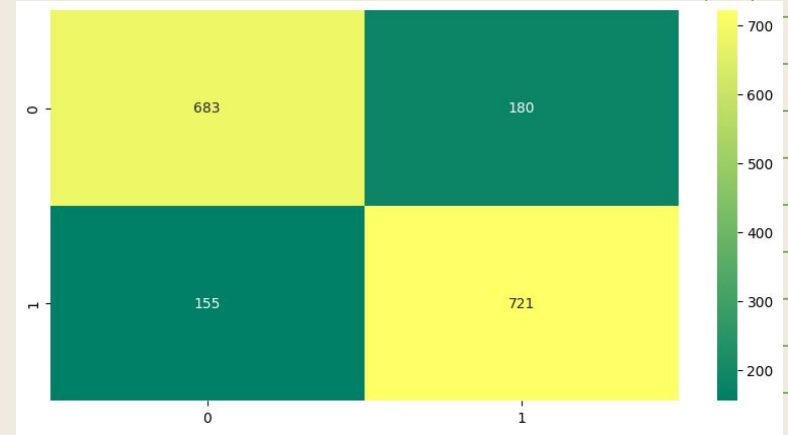
# Resultados del modelo de Gradient Boosting Classifier

Precision en Training Data	82.05
Precision en Testing Data	79.29
Precision	0.77
Recall	0.85
F1 Score	0.80



# Resultados del modelo de XGB Classifier

<b>Precision en Training Data</b>	92.52
<b>Precision en Testing Data</b>	80.73
<b>Precision</b>	0.80
<b>Recall</b>	0.82
<b>F1 Score</b>	0.81



# Resultados del modelo de CatBoost Classifier

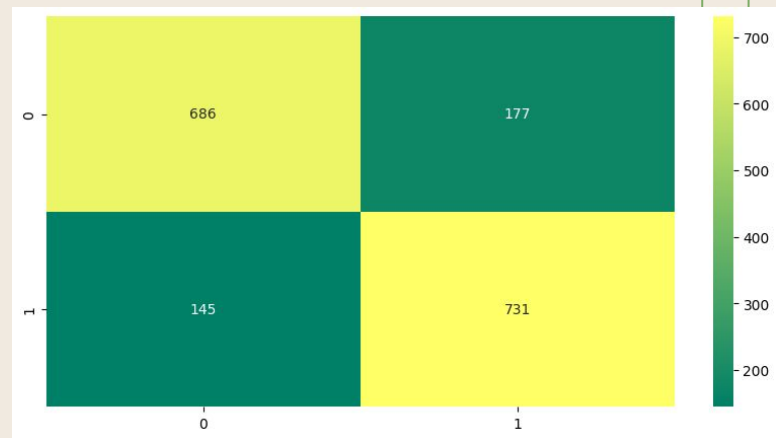
**Precision en Training Data** 87.40

**Precision en Testing Data** 81.48

**Precision** 0.80

**Recall** 0.83

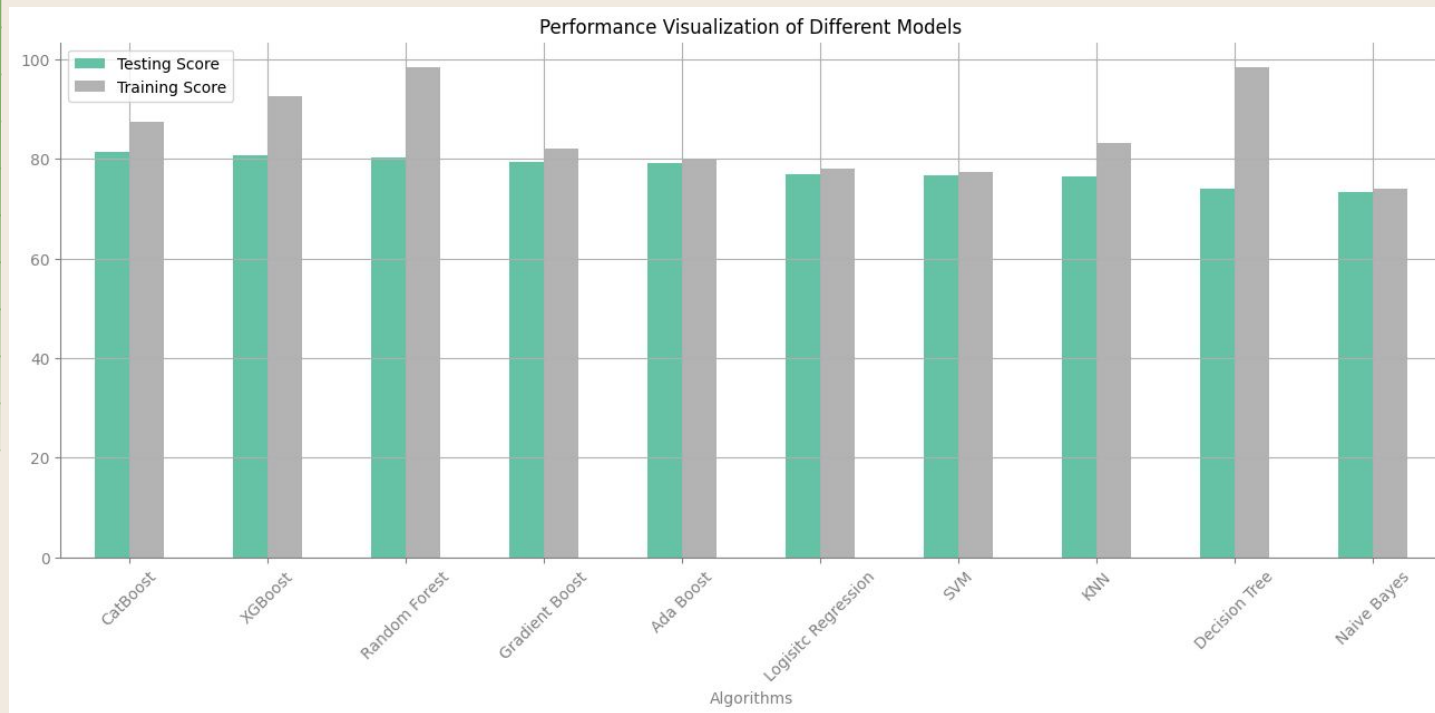
**F1 Score** 0.81



08

# Comparación entre Modelos



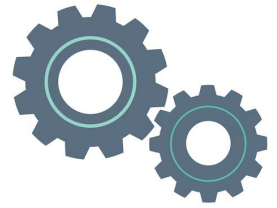


El mejor rendimiento lo presentó Cat Boost con un ~81.48. Sin embargo RandomForest y XGBoost también presentaron buen rendimiento (+80)



09

# Selección de *Hyper Parameters*





# Hyper Parameters en modelo CatBoost



Learning Rate: [0.1, 0.3, 0.5, 0.6, 0.7]  
Random State: [0, 42, 48, 50]  
Depth: [8, 9, 10]  
Iterations: [35, 40, 50]



**Grid Search  
CV  
(cv = 5)**



Learning Rate: 0.3  
Random State: 48  
Depth: 8  
Iterations: 40  
Precision: 0.80

# Hyper Parameters en modelo XGB



Learning Rate: [0.1, 0.3, 0.5, 1.0]  
Random State: [0, 42, 50]  
N Estimators: [50, 100, 150]

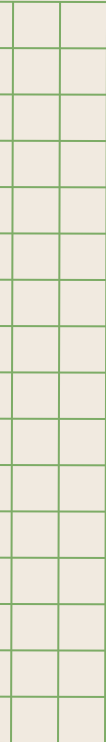


**Grid Search  
CV  
(cv = 5)**



Learning Rate: 0.1  
Random State: 0  
N Estimators: 50  
Score: 0.80

# Hyper Parameters en modelo Random Forest



N Estimators: [100, 300, 500, 550]  
Min Samples Split: [7, 8, 9]  
Max Depth: [10, 11, 12]  
Min Samples Leaf: [4, 5, 6]



**Grid Search  
CV  
(cv = 5)**



N Estimators: 100  
Min Samples Split: 4  
Max Depth: 12  
Min Samples Leaf: 4  
Score: 0.80



# Resultados del modelo de Stacking Classifier

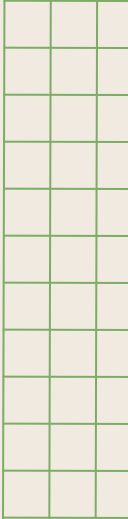
Incluye:

Cat Boost Classifier

XGBoost Classifier

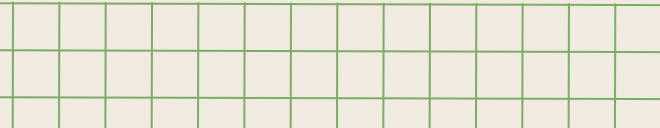
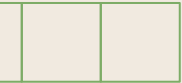
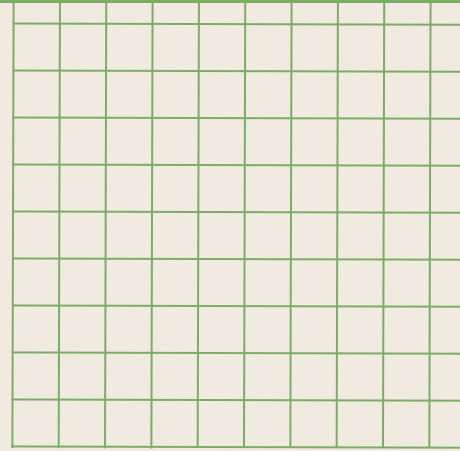
Random Forest Classifier

<b>Precision en Training Data</b>	85.64
<b>Precision en Testing Data</b>	81.08



# 10

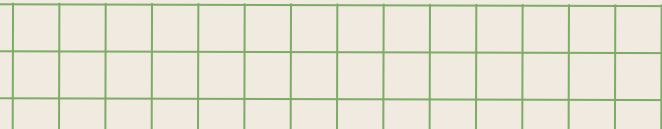
## Prediccion



# 53.10%

## De las personas fueron transportadas

Según la predicción de 81% accuracy



# Gracias por su atencion

## Redes personales:

LinkedIn: Juan Manuel Gonzalez Kapnik

Github: just-juanma

E-mail: [juanmanuelgonzalezkapnik@gmail.com](mailto:juanmanuelgonzalezkapnik@gmail.com)

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

