ITBA
Instituto Tecnológicc
de Buenos Aires

82.05 - Análisis Predictivo

Examen Final

Inés Murtagh

ITBA

Trabajo Final

Análisis Predictivo

Airline Passenger Satisfaction

Encuesta de satisfacción de pasajeros de una línea aérea.

Objetivo

El objetivo o meta de este proyecto es guiar a una compañía aérea a determinar los factores importantes que influyen en la satisfacción del cliente o pasajero de la aerolínea.

Hipótesis

¿Se puede predecir la satisfacción de un pasajero?

¿Existe un patrón, en función de las calificaciones otorgadas por los pasajeros, que refleje la experiencia general del cliente y su satisfacción?

Análisis Exploratorio

base de datos

Airline Passenger Satisfaction

#	id	Gender	Customer ...	Age	Type of Tr...	Class	Flight Dist...	Inflight wif...	Departure...	Ease of On...	satisfaction
0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3	neutral or dissatisfied
1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	3	neutral or dissatisfied
2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2	satisfied
3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	5	neutral or dissatisfied
4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	3	satisfied
5	111157	Female	Loyal Customer	26	Personal Travel	Eco	1180	3	4	2	neutral or dissatisfied
6	82113	Male	Loyal Customer	47	Personal Travel	Eco	1276	2	4	2	neutral or dissatisfied
7	96462	Female	Loyal Customer	52	Business travel	Business	2035	4	3	4	satisfied
8	79485	Female	Loyal Customer	41	Business travel	Business	853	1	2	2	neutral or dissatisfied
9	65725	Male	disloyal Customer	20	Business travel	Eco	1061	3	3	3	neutral or dissatisfied
10	34991	Female	disloyal Customer	24	Business travel	Eco	1182	4	5	5	neutral or dissatisfied
11	51412	Female	Loyal Customer	12	Personal Travel	Eco Plus	308	2	4	2	neutral or dissatisfied
12	98628	Male	Loyal Customer	53	Business travel	Eco	834	1	4	4	neutral or dissatisfied
13	83502	Male	Loyal Customer	33	Personal Travel	Eco	946	4	2	4	
					Personal Travel	Eco	453	3	2	3	target

Cantidad de Filas y columnas: (103904, 24)

Análisis Exploratorio de datos

Análisis Exploratorio

base de datos

Variables

Género: Género de los pasajeros (Femenino, Masculino)
Edad: La edad real de los pasajeros.
Tipo de cliente: el tipo de cliente (cliente fiel, cliente desleal)
Tipo de Viaje: Propósito del vuelo de los pasajeros (Viaje Personal, Viaje de Negocios)
Clase: Clase de viaje en el avión de los pasajeros (Business, Eco, Eco Plus)
Distancia de vuelo: la distancia de vuelo de este viaje
Retraso de salida en minutos: Minutos de retraso en la salida
Retraso de llegada en minutos: Minutos de retraso en la llegada
Satisfacción: Nivel de satisfacción de la aerolínea (Satisfacción, neutral o insatisfacción)

+ Total Score
+ Total Score %
+ Average Rating

Calificaciones (nivel de satisfacción):

Embarque en línea: Nivel de satisfacción del embarque en línea
Comodidad del asiento: Nivel de satisfacción de Confort del asiento
Entretenimiento a bordo: Nivel de satisfacción del entretenimiento a bordo
Servicio a bordo: Nivel de satisfacción del servicio a bordo
Servicio de sala de piernas: Nivel de satisfacción del servicio de sala de piernas
Manejo de equipaje: Nivel de satisfacción del manejo de equipaje
Servicio de Check-in: Nivel de satisfacción del servicio de Check-in
Servicio a bordo: Nivel de satisfacción del servicio a bordo
Servicio wifi a bordo: Nivel de satisfacción del servicio wifi a bordo
Limpieza: Nivel de satisfacción de Limpieza
Hora de salida/llegada conveniente: Nivel de satisfacción de la hora de salida/llegada
Facilidad de reserva en línea: Nivel de satisfacción de la reserva en línea
Ubicación de la puerta: nivel de satisfacción de la ubicación de la puerta
Alimentos y bebidas: Nivel de satisfacción de Alimentos y bebidas

Análisis Exploratorio

variable target

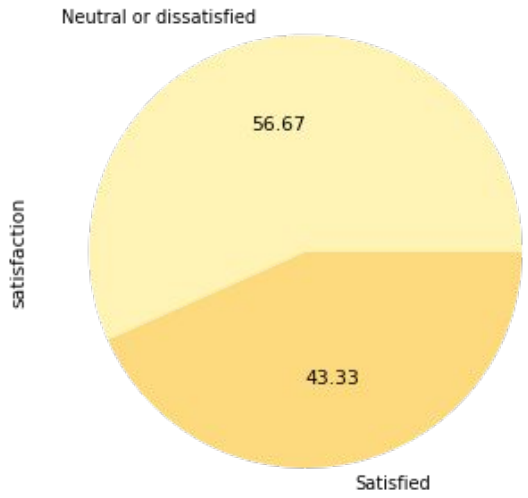
satisfaction
neutral or dissatisfied
neutral or dissatisfied
satisfied
neutral or dissatisfied
satisfied
neutral or dissatisfied
neutral or dissatisfied
satisfied
neutral or dissatisfied
neutral or dissatisfied
neutral or dissatisfied
neutral or dissatisfied
neutral or dissatisfied

target

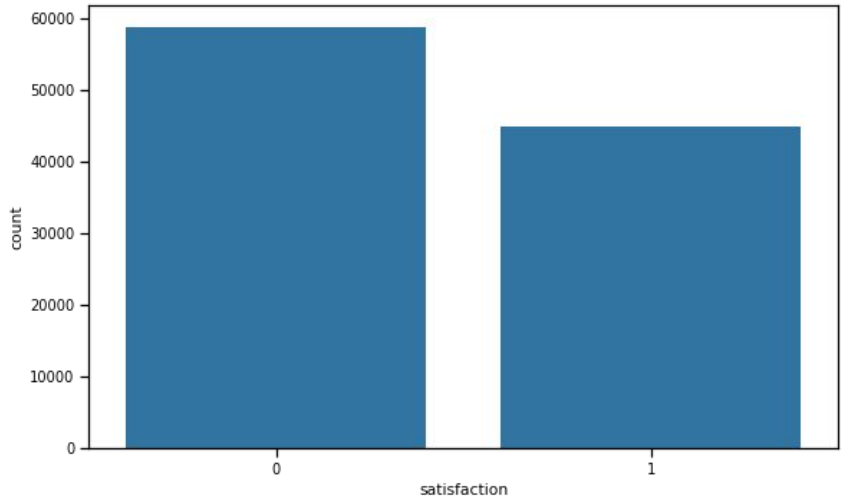
Objetivo: predecir la satisfacción de un pasajero del vuelo

Variable Target: satisfaction

Modelo: clasificación




```
neutral or dissatisfied    58879
satisfied                  45025
Name: satisfaction, dtype: int64
```



Missings

Análisis Exploratorio

missings

```
id                                     0
Gender                                 0
Customer Type                          0
Age                                    0
Type of Travel                         0
Class                                  0
Flight Distance                        0
Inflight wifi service                  0
Departure/Arrival time convenient      0
Ease of Online booking                 0
Gate location                          0
Food and drink                         0
Online boarding                        0
Seat comfort                           0
Inflight entertainment                 0
On-board service                       0
Leg room service                       0
Baggage handling                       0
Checkin service                        0
Inflight service                       0
Cleanliness                            0
Departure Delay in Minutes             0
Arrival Delay in Minutes             310
satisfaction                           0
dtype: int64
```

supuestos:

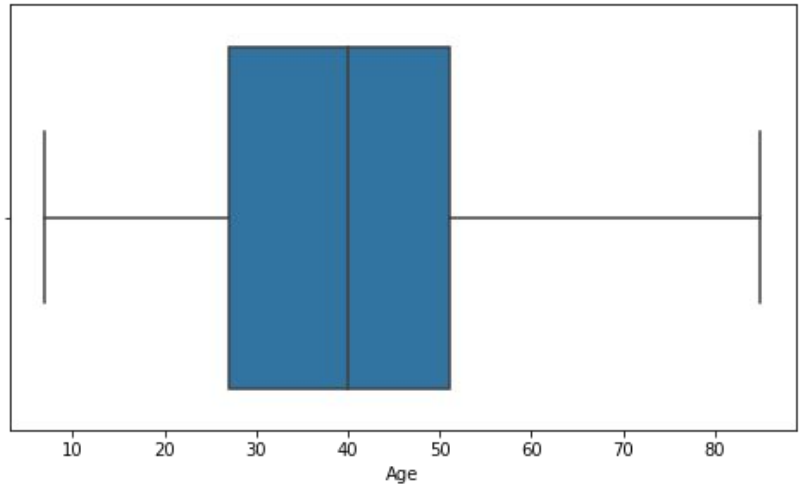
‘Arrival delay in Minutes’: si el valor es nulo, se toma como supuesto que no se retrasó el aterrizaje para ese mismo vuelo.

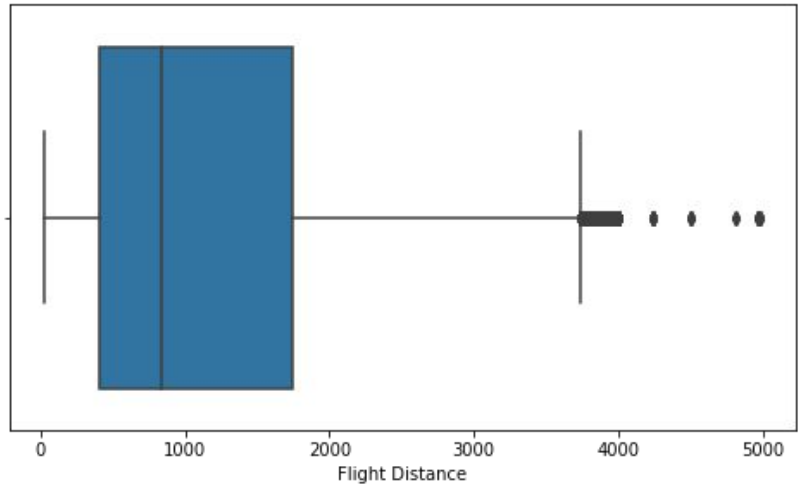
Para los valores nulos, se asignó el número 0

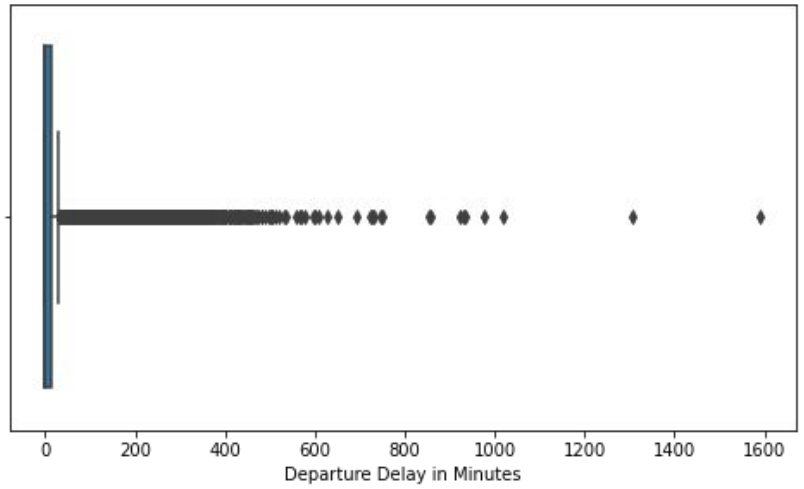
Outliers

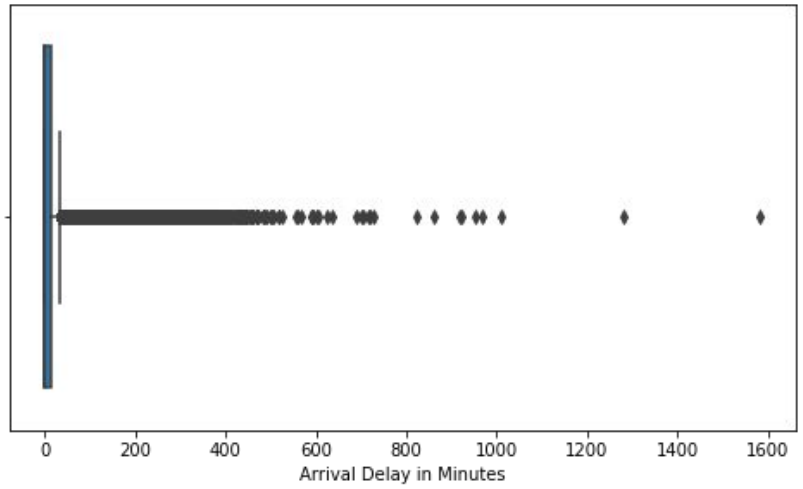
Análisis Exploratorio

outliers





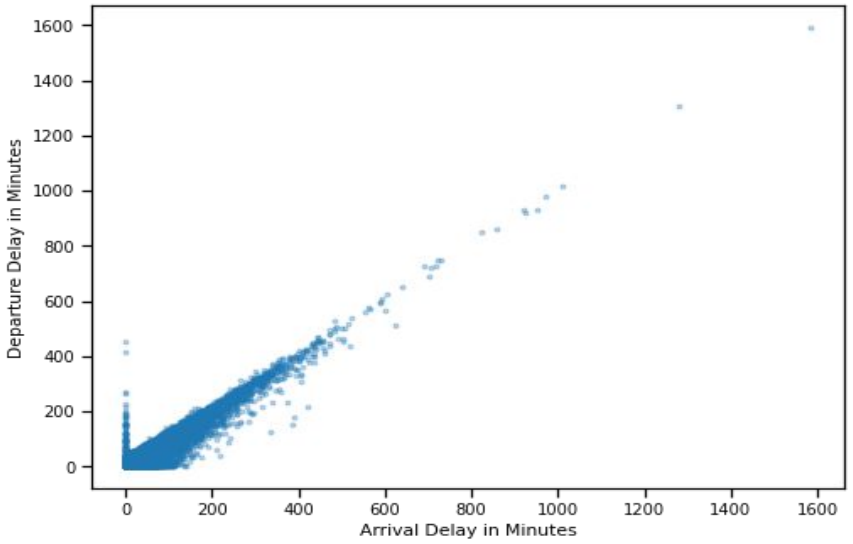


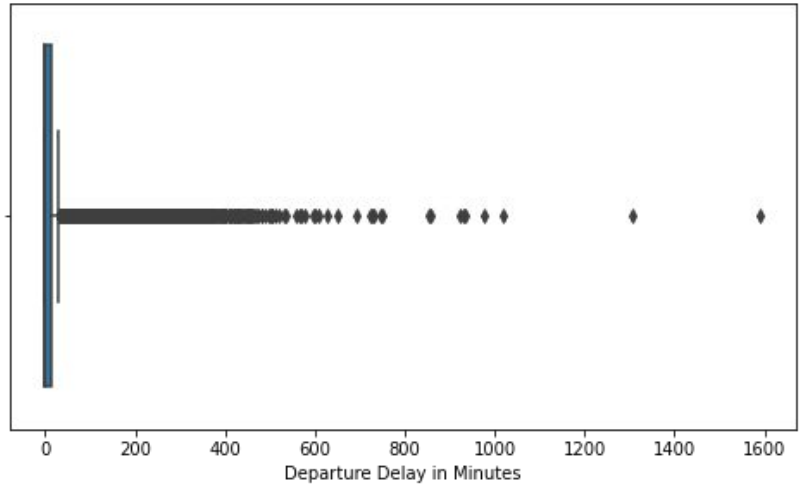


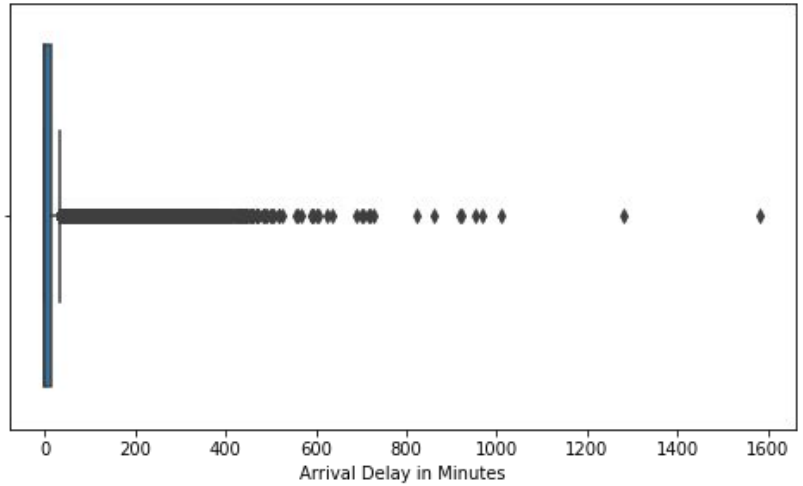
Análisis Exploratorio

outliers

	Departure Delay in Minutes	Arrival Delay in Minutes
count	103904.000000	103904.000000
mean	14.815618	15.133392
std	38.230901	38.649776
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	12.000000	13.000000
max	1592.000000	1584.000000



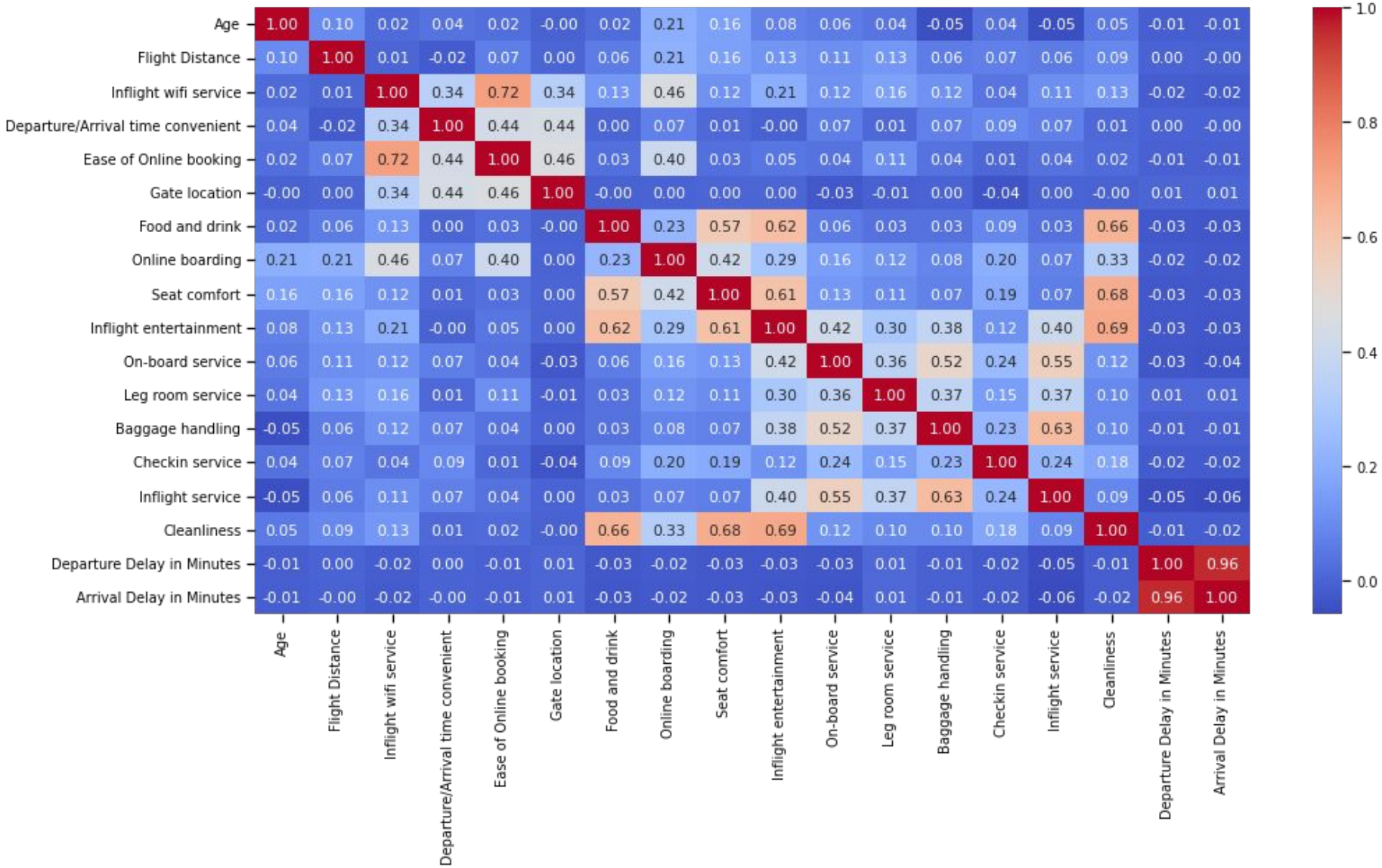




Correlación

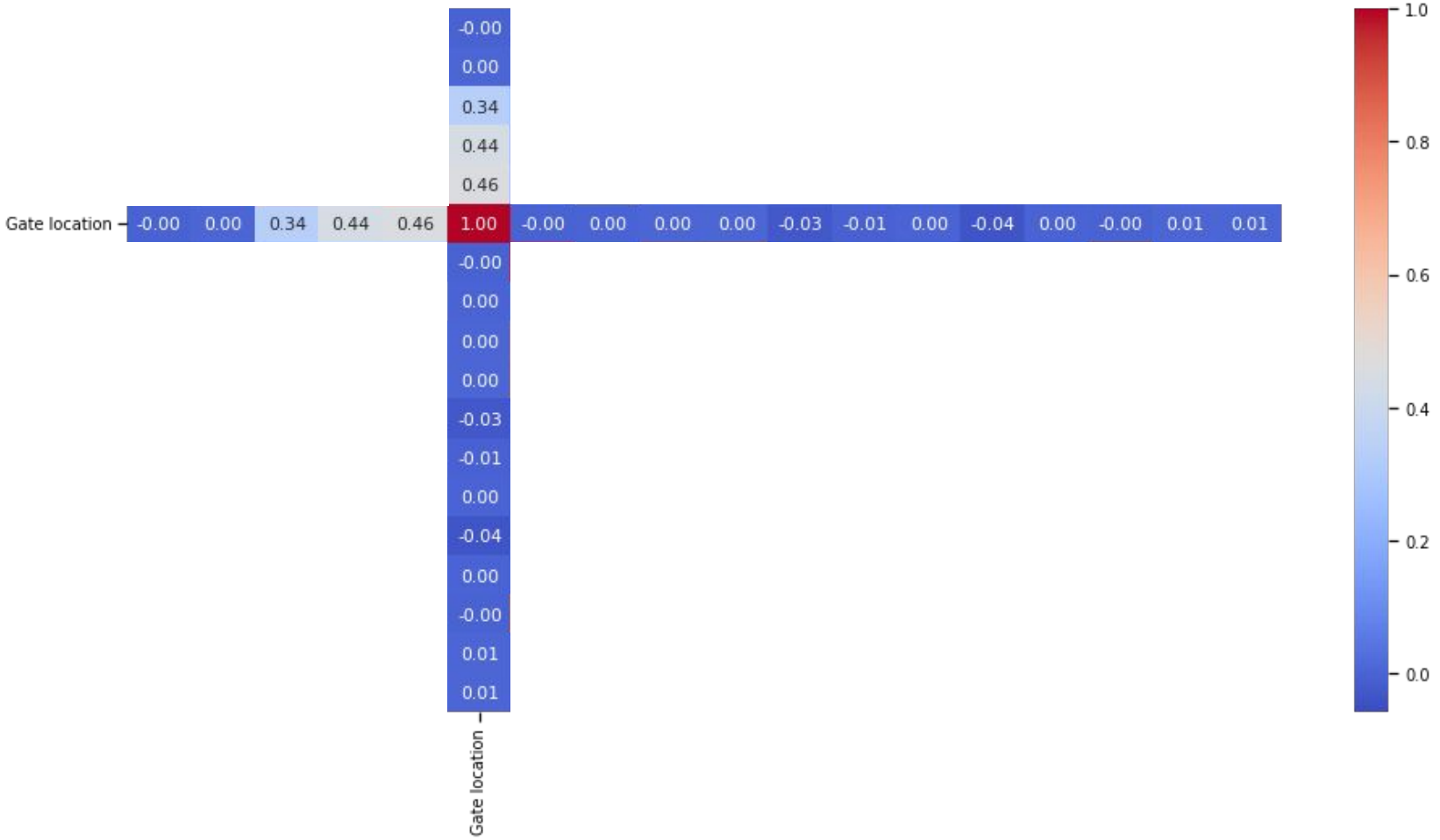
Análisis Exploratorio

correlación



Análisis Exploratorio

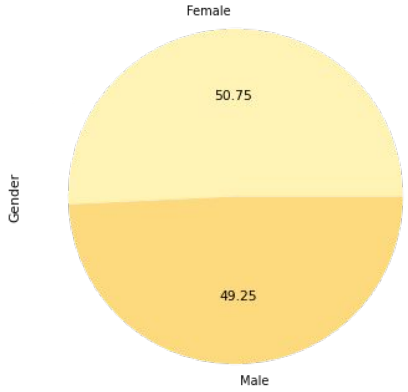
correlación

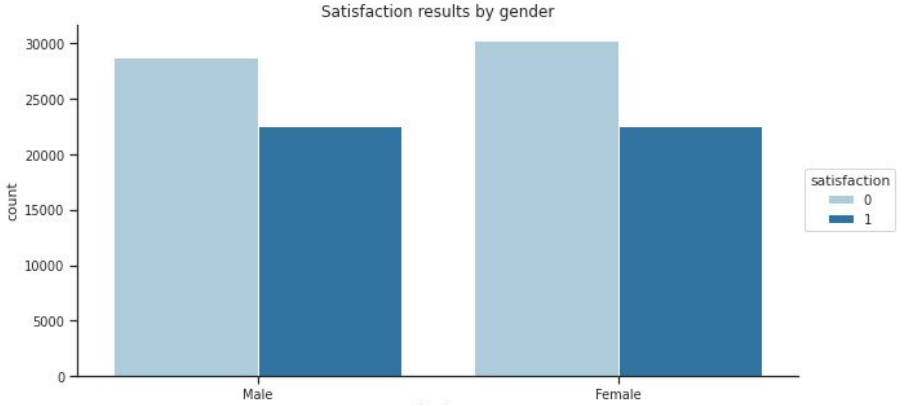


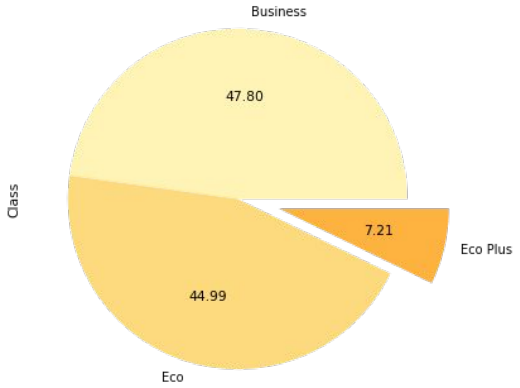
Distribución de las variables y visualizaciones

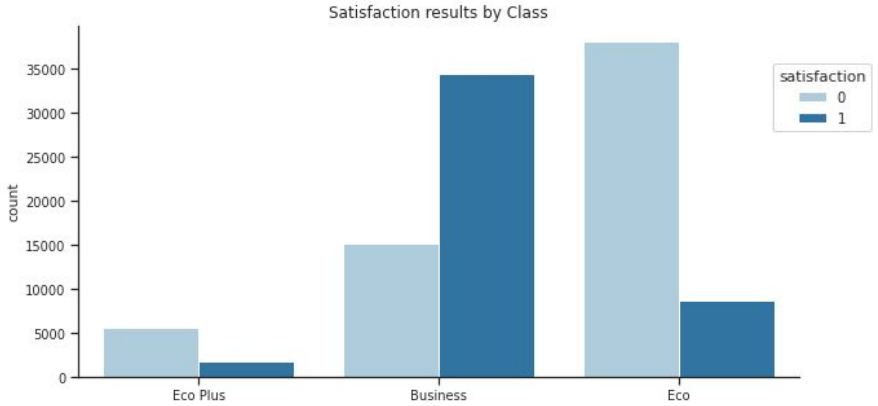
Visualizaciones

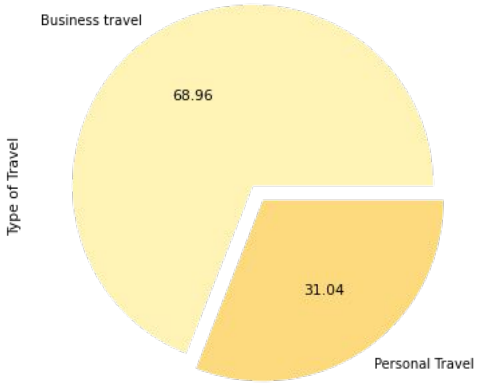
satisfacción por categoría

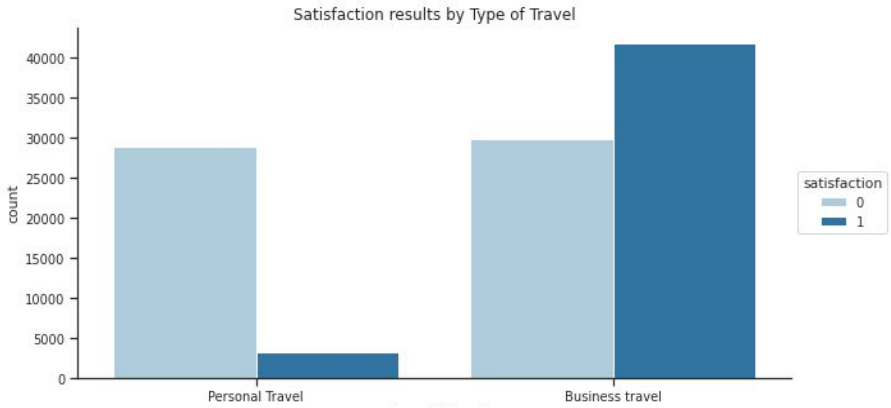






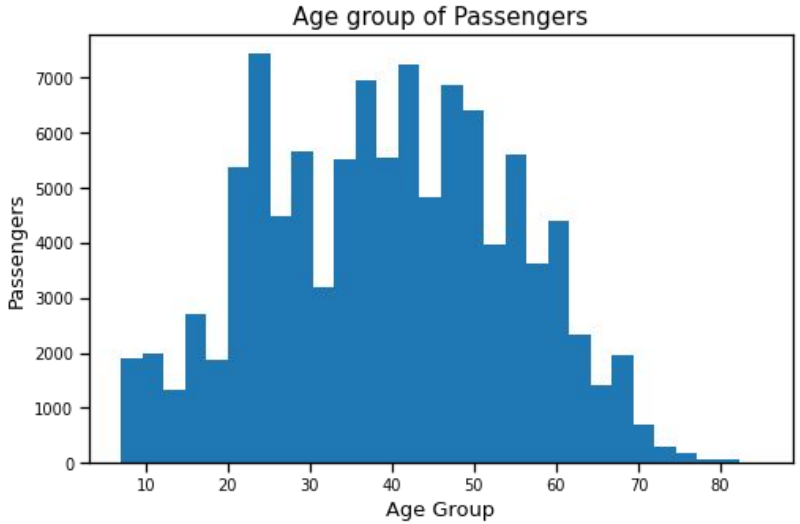


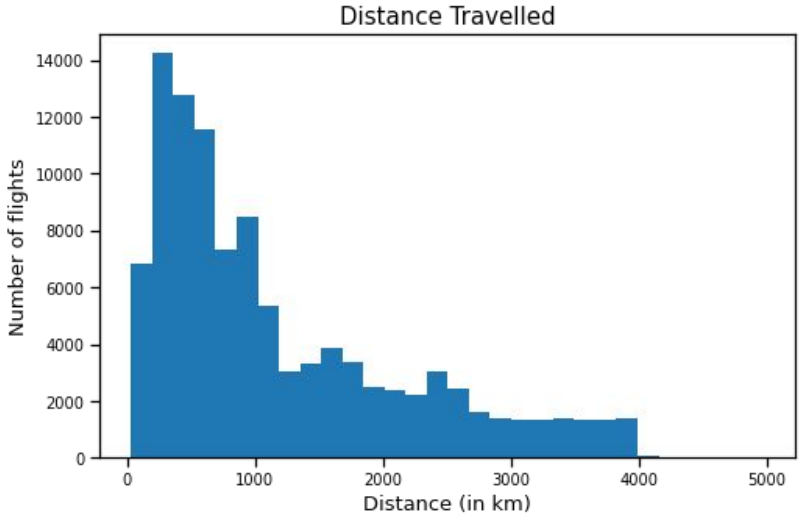


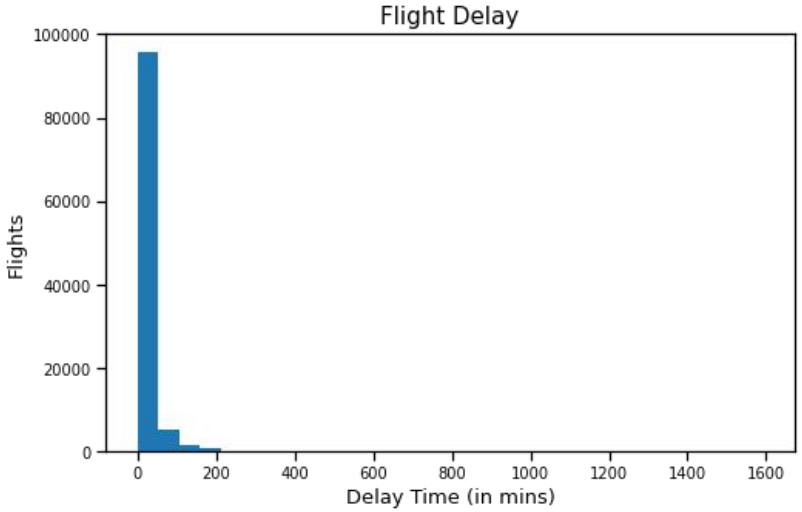


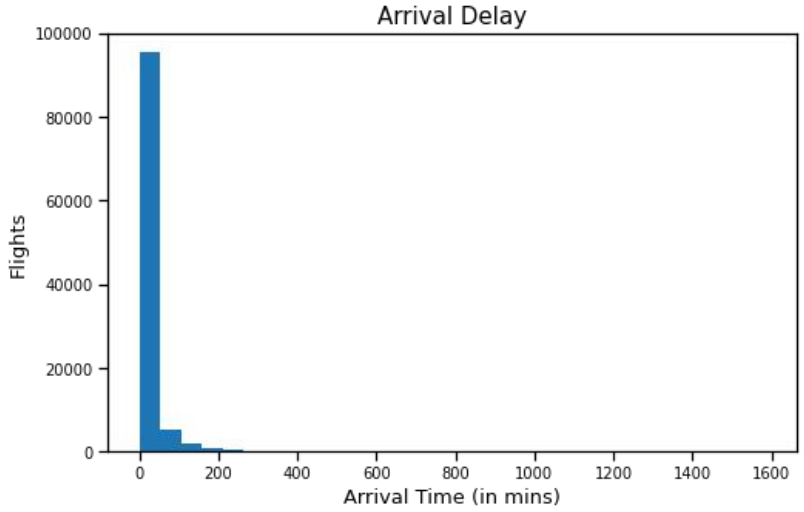
Distribución de las variables

Histogramas

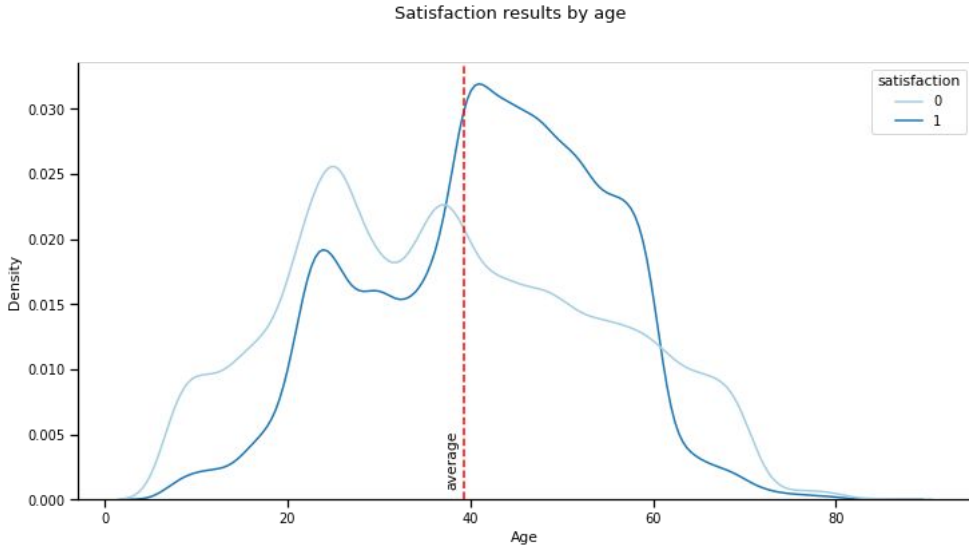


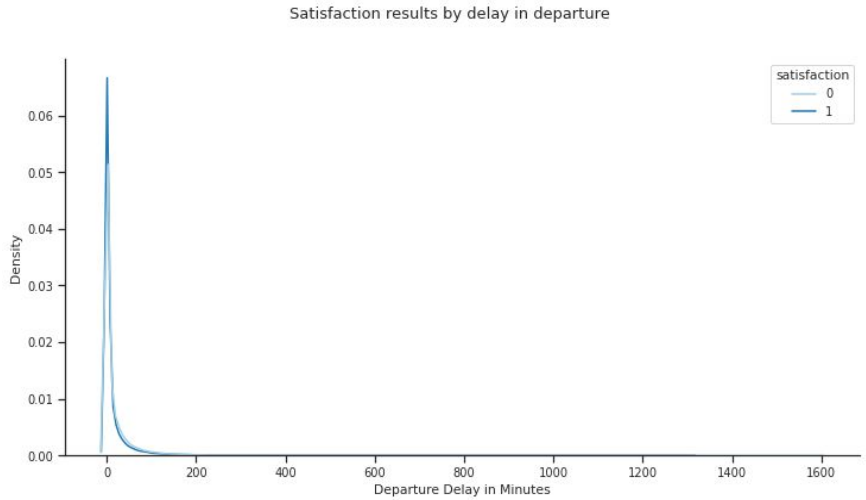


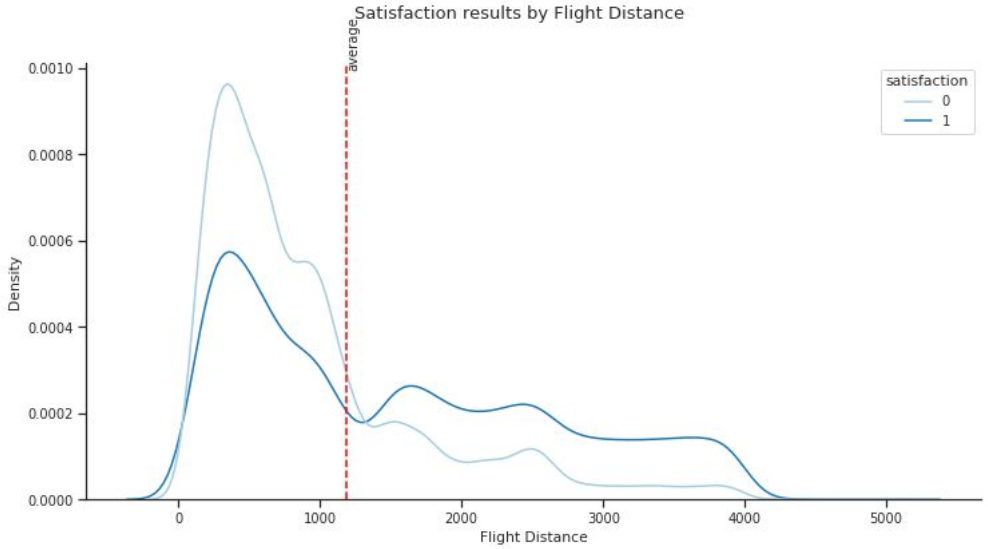


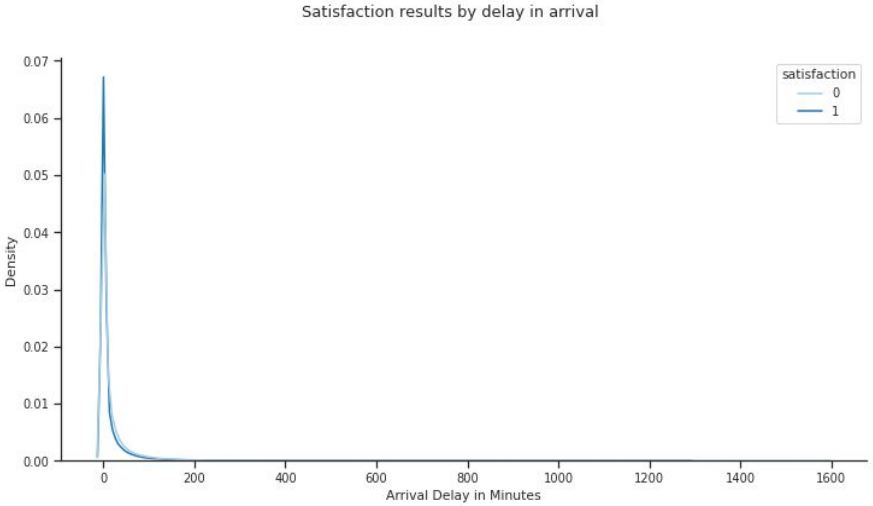


Distribución de las variables
por nivel de satisfacción









Modelos Utilizados

Encoding

Variables categóricas

One-Hot Encoding

variable: **‘Customer Type’** (loyal or disloyal), **‘Type of Travel’** (business or personal)

```
# Creamos las variables binarias
dummies = pd.get_dummies(df['Columna'])

# Añadimos las variables binarias al DataFrame
df = pd.concat([df, dummies], axis = 1)
```

Ordinal Encoding

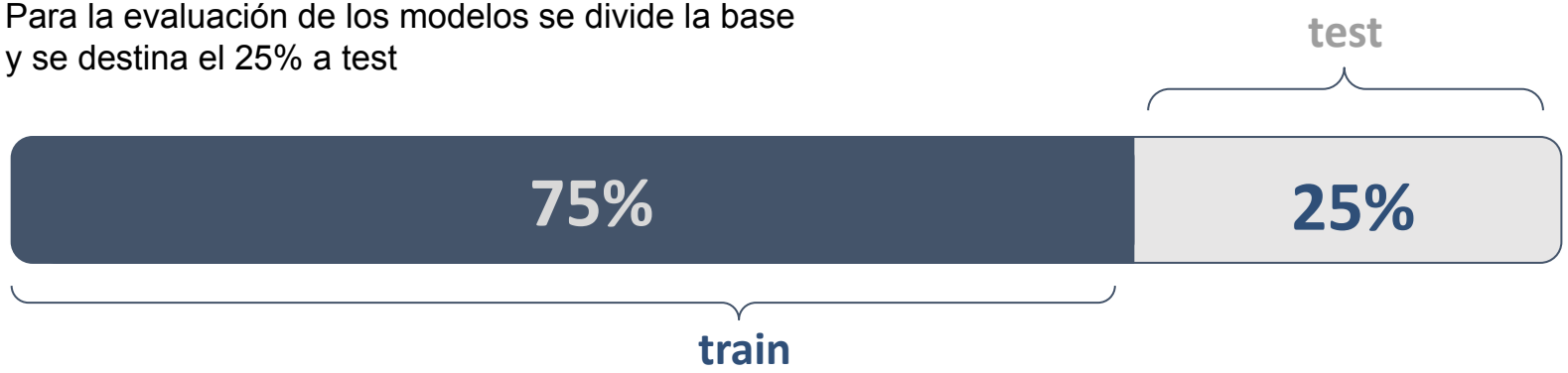
variable: **‘Class’** (eco, eco plus, business)

```
encoder = OrdinalEncoder(categories=[['Eco Plus', 'Business', 'Eco']])

# Ajustamos el codificador con la variable class y la transformamos
encoder.fit(df[["Class"]])
df["Class-encoded"] = encoder.transform(df[["Class"]])
```

Partición de la base

train y test

Para la evaluación de los modelos se divide la base
y se destina el 25% a test




```
Cantidad de registros en train: 77928
Cantidad de registros en test: 25976
```

```python
from sklearn.model_selection import train_test_split

columnas = ['Age', 'Flight Distance', 'Inflight wifi service', 'Departure/Arrival time convenient', 'Ease of Online booking', 'Food and drink', 'Online boarding',
            'Seat comfort', 'Inflight entertainment', 'On-board service', 'Leg room service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness',
            'Total Score %', 'Arrival Delay in Minutes', 'Loyal Customer', 'disloyal Customer', 'Business travel', 'Personal Travel', 'Class-encoded']

X = df[columnas]
y = df['satisfaction']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 21)
```

```python
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Comparación de modelos

Modelos utilizados

comparación

Modelos Probados

para clasificación

1. DecisionTreeClassifier( )
2. RandomForestClassifier( )
3. Extra Tree Classifier( )
4. KNeighborsClassifier( )
5. AdaBoostClassifier( )
6. CatBoostClassifier( )
7. LGBMClassifier( )

Modelos utilizados

comparación

DecisionTreeClassifier():

0.6903605981590883

RandomForestClassifier()

0.8092596187241394

ExtraTreesClassifier()

0.7993775101547487

KNeighborsClassifier()

0.09633607193237459

AdaBoostClassifier()

0.6393814666503261

XGBClassifier()

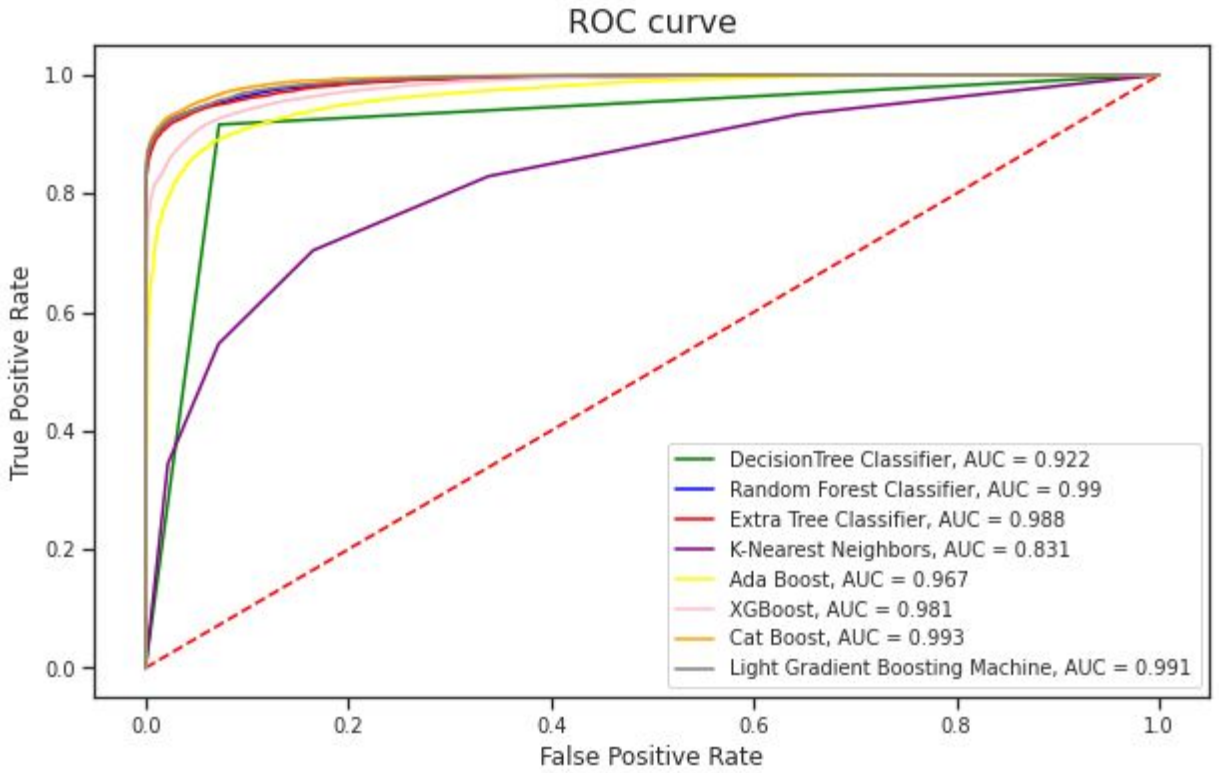
0.7146737224171132

CatBoostClassifier()

0.8229063400818697

LGBMClassifier()

0.8098870541888628

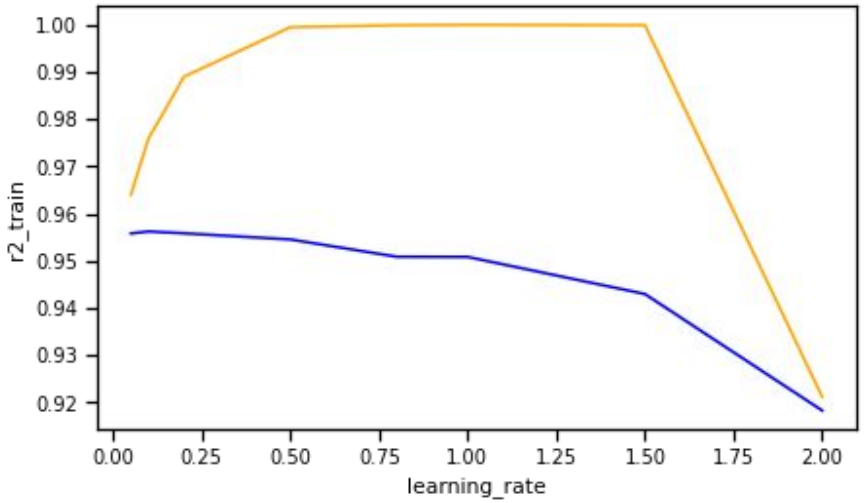


Ajuste de hiperparametros

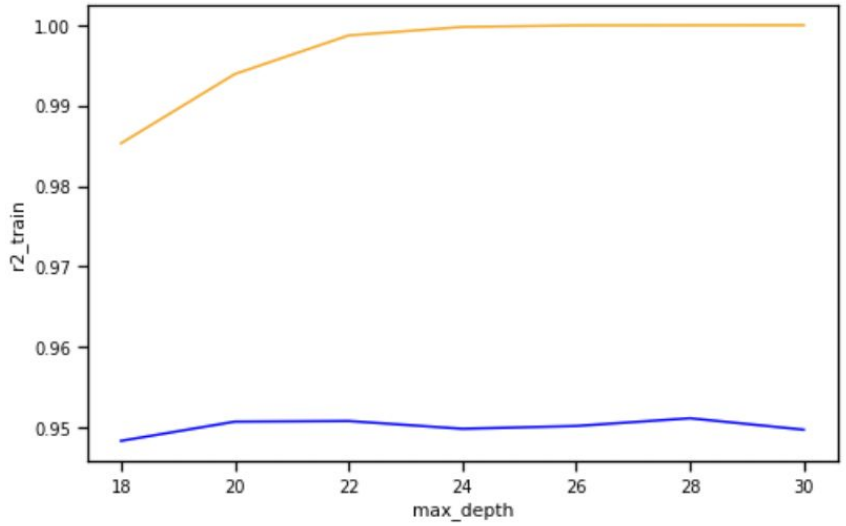
Ajuste de hiperparámetros

learning_rate y max_depth

```
fila = []
lista = [0.1, 0.2, 0.5, 0.7, 0.8, 1, 1.5, 2.0]

for i in lista :
  cb = CatBoostClassifier(learning_rate = i)
  cb.fit(X_train,y_train)
  train = cb.score(X_train, y_train)
  test = cb.score(X_test, y_test)
  fila.append([i, train, test])

scores = pd.DataFrame(fila, columns=["learning_rate", "r2_train", "r2_test"])

fig, ax = plt.subplots(figsize=(7, 4))
ax = sns.lineplot(scores, x="learning_rate", y="r2_train", color = 'orange')
ax = sns.lineplot(scores, x="learning_rate", y="r2_test", color = 'blue')
```



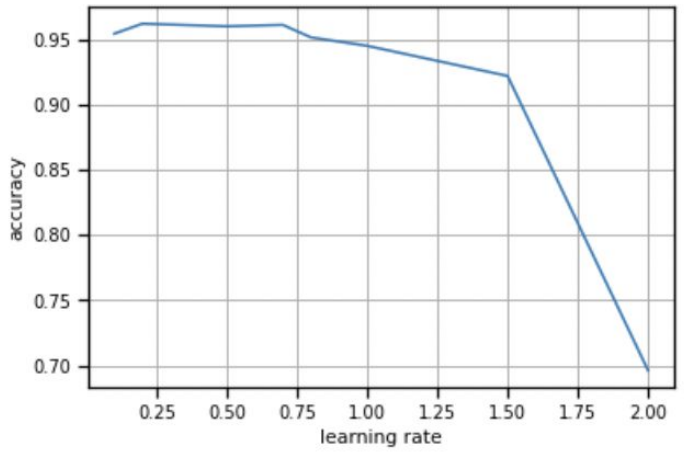

```
fila = []

for i in range(18, 31, 2):
  extratree = ExtraTreesClassifier(bootstrap = False, max_depth = i)
  extratree.fit(X_train,y_train)
  train = extratree.score(X_train, y_train)
  test = extratree.score(X_test, y_test)
  fila.append([i, train, test])

scores = pd.DataFrame(fila, columns=["max_depth","r2_train","r2_test"])

fig, ax = plt.subplots(figsize=(8, 5))
ax = sns.lineplot(data=scores, x="max_depth", y="r2_train", color = 'orange')
ax = sns.lineplot(data=scores, x="max_depth", y="r2_test", color = 'blue')
```



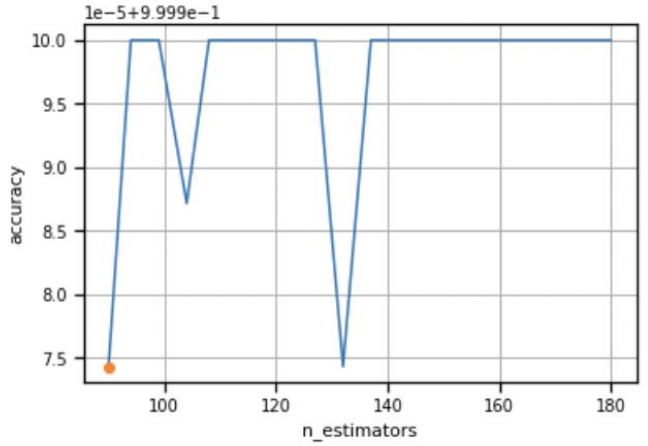
Ajuste de hiperparámetros

learning_rate, n_estimators y depth

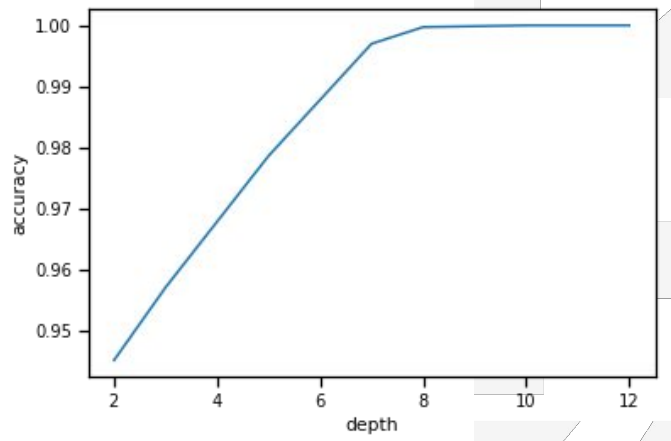
```python
# optimizando learning_rate
lista = [0.1, 0.2, 0.5, 0.7, 0.8, 1, 1.5, 2.0]
accuracy = []

for x in lista:
    lgbmc = LGBMClassifier(learning_rate = x)
    lgbmc.fit(X_train, y_train)
    accuracy.append(lgbmc.score(X_train, y_train))

plt.plot(lista, accuracy, color = '#1e77b4')
plt.grid()
plt.xlabel("learning rate")
plt.ylabel("accuracy")
```




```python
# optimizando n_estimators
lista = range(90, 180, 20)
accuracy = []

for x in lista:
    rf = RandomForestClassifier(n_estimators = x)
    rf.fit(X_train, y_train)
    accuracy.append(rf.score(X_train, y_train))

plt.plot(lista, accuracy)
plt.plot(lista[0], accuracy[0], marker='o')
plt.grid()
plt.ylabel('accuracy')
plt.xlabel('n_estimators')
```

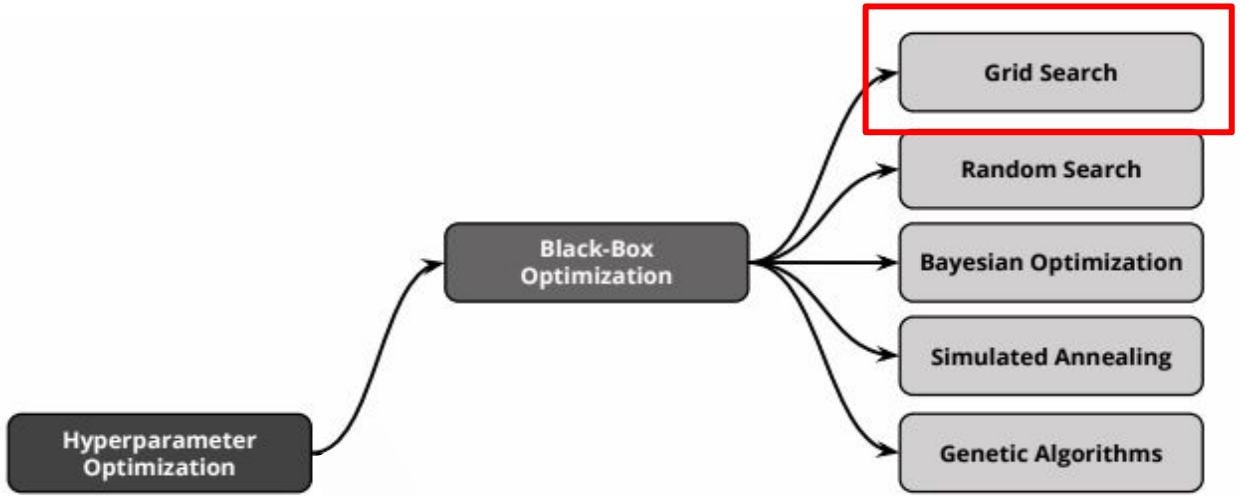



```python
# optimizando depth
lista = [2, 3, 5, 7, 8, 10, 12]
accuracy = []

for x in lista:
    cb = CatBoostClassifier(learning_rate = 0.2, depth = x)
    cb.fit(X_train, y_train)
    accuracy.append(cb.score(X_train, y_train))

plt.plot(lista, accuracy, color = '#1e77b4')
plt.xlabel("depth")
plt.ylabel("accuracy")
```



Ajuste de hiperparámetros

Grid Search



Grid Search: recorre todas las combinaciones de hiperparametros posibles y elegir la mejor.

```
parametros = {'iterations': [100, 200, 300, 500, 800],
              'depth': [8],
              'learning_rate': [0.2],
              'l2_leaf_reg': [1, 2, 4, 5, 8]}

modelo = CatBoostClassifier()
grid = GridSearchCV(estimator = modelo, param_grid = parametros, cv = 3, n_jobs = -1)
grid.fit(X_train, y_train)
```

```
[421] grid.best_params_

      {'depth': 8, 'iterations': 500, 'l2_leaf_reg': 5, 'learning_rate': 0.2}
```

Ajuste de hiperparámetros

modelos

```
[199] extra = ExtraTreesClassifier(bootstrap = False, max_depth= 28, max_features = 10, min_samples_split = 3, n_estimators = 400)
      extra.fit(X_train, y_train)
```

```
ExtraTreesClassifier(max_depth=28, max_features=10, min_samples_split=3,
                     n_estimators=400)
```

```
[202] # {'max_depth': 15, 'n_estimators': 325, 'num_leaves': 24, 'objective': 'binary'}
      lgbm = LGBMClassifier(boosting_type='gbdt', max_depth = 15, learning_rate = 0.2, n_estimators = 325, num_leaves = 24, objective = 'binary')
      lgbm.fit(X_train, y_train)
```

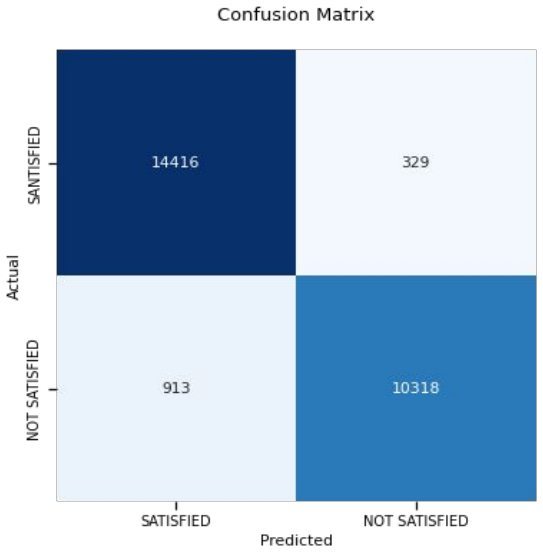
```
LGBMClassifier(learning_rate=0.2, max_depth=15, n_estimators=325, num_leaves=24,
               objective='binary')
```

```
[193] # 'bootstrap': True, 'max_depth': 40, 'max_features': 5, 'min_samples_leaf': 3, 'min_samples_split': 8
      rf = RandomForestClassifier(bootstrap = True, max_depth = 40, max_features = 5, min_samples_leaf = 3, min_samples_split = 8, n_estimators = 120)
      rf.fit(X_train, y_train)
```

```
[46] # {'depth': 8, 'iterations': 500, 'l2_leaf_reg': 5, 'learning_rate': 0.2}
     catboost = CatBoostClassifier(depth = 8, iterations = 500, learning_rate = 0.2, l2_leaf_reg = 5)
     catboost.fit(X_train, y_train)
```

Modelos utilizados

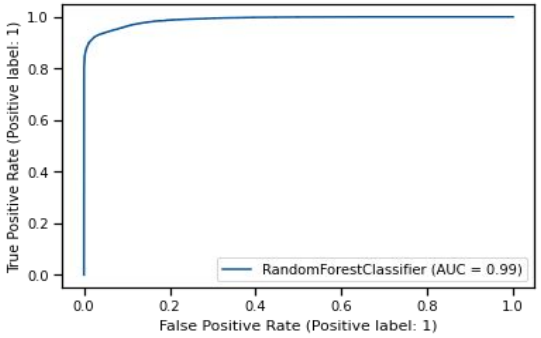
Random Forest



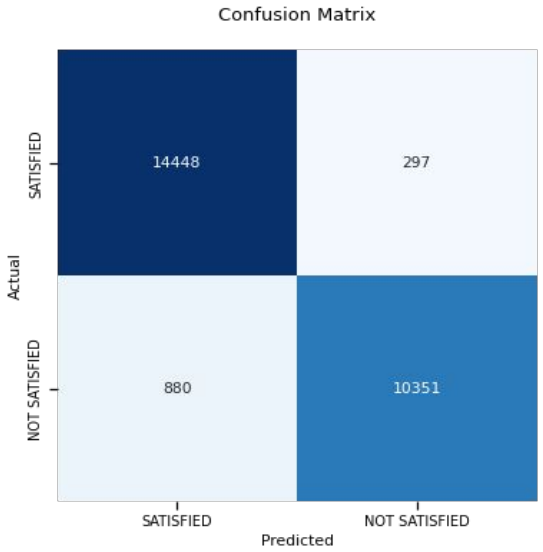

```
Accuracy:  0.9516091777024946
Precision:  0.967385192127460 2
Recall:  0.919063306918351
F1:  0.9426053604858226
```

```
[ ]  roc_auc_score(y_test, y_pred)
```

```
0.9483626391480081
```



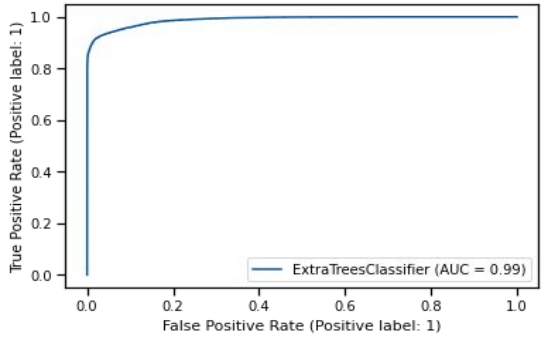
Extra Trees



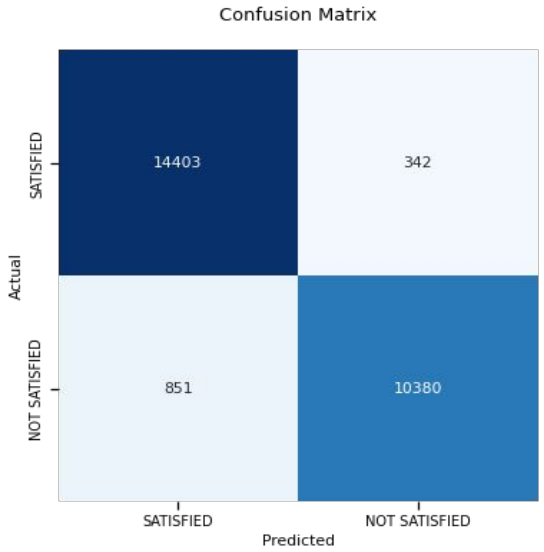

```
Accuracy:  0.9539574992300586
Precision:  0.9698473639853918
Recall:  0.9221796812394266
F1:  0.9454130534002738
```

```
[ ]  roc_auc_score(y_test, y_pred)
```

```
0.9501708850415511
```



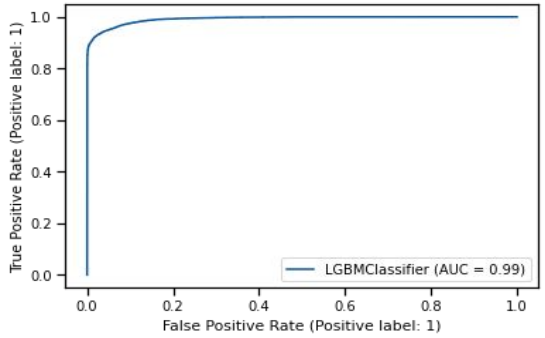
Light Gradient Boosting Machine



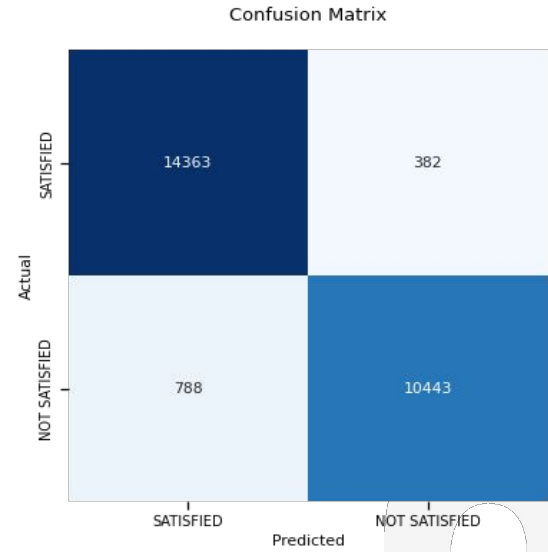

```
Accuracy:  0.9557668617185094
Precision:  0.9689302325581395
Recall:  0.9274329979520969
F1:  0.9477275829125154
```

```
[ ]  roc_auc_score(y_test, y_pred)
```

```
0.9523906257987003
```



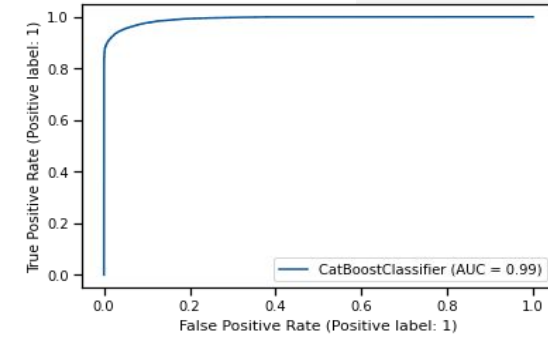
Cat Boost




```
Accuracy:  0.9567292885740684
Precision:  0.9664051684356253
Recall:  0.9323301575995013
F1:  0.9490619051935103
```

```
[ ]  roc_auc_score(y_test, y_pred)
```

```
0.9538219116244371
```



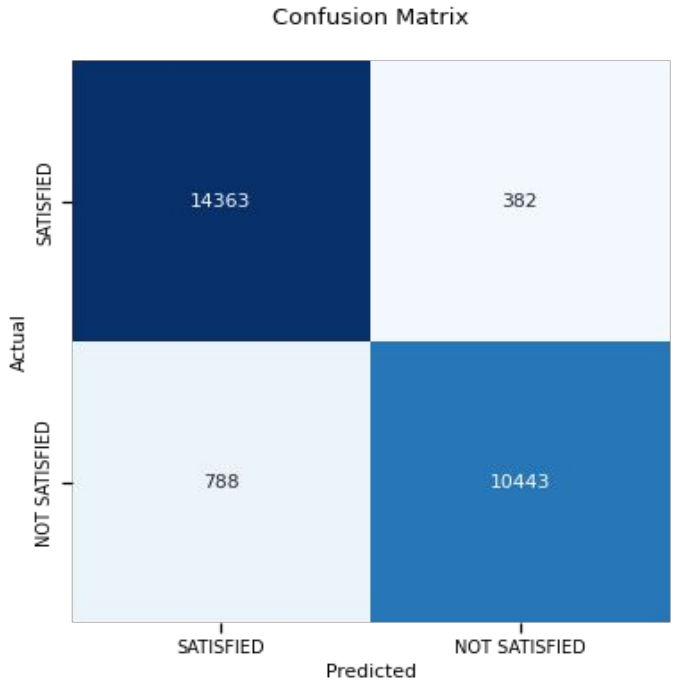
Modelo Ganador

Modelos ganador

Cat boost

CatBoostClassifier()

CatBoost resuelve las características categóricas mediante una alternativa impulsada por permutación en comparación con el algoritmo clásico.



Classification Report

	precision	recall	f1-score	support
0	0.96	0.98	0.97	14745
1	0.97	0.95	0.96	11231
accuracy			0.96	25976
macro avg	0.96	0.96	0.96	25976
weighted avg	0.96	0.96	0.96	25976

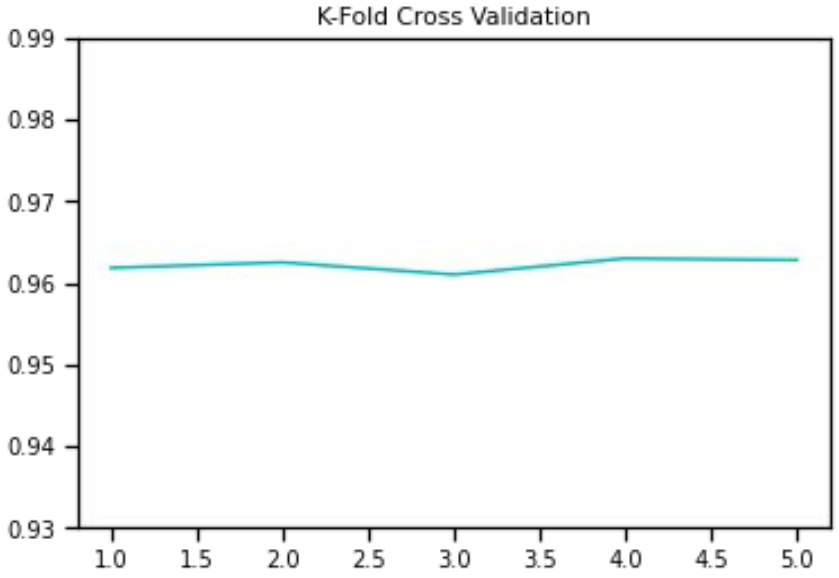
Accuracy: 0.9632737911918694
Precision: 0.9681151498587957
Recall: 0.9462202831448668
F1: 0.9570425072046108

Modelo Ganador

k-Fold Cross Validation

catboost = CatBoostClassifier(depth = 8, iterations = 500, learning_rate = 0.2, l2_leaf_reg = 5)

```
Cross Validation Scores:  [0.96184014 0.96251383 0.96102209 0.96299504 0.96280077]
Average CV Score:  0.962234375638997
Number of CV Scores used in Average:  5
```



Conclusiones

Conclusiones

Objetivo

El objetivo o meta de este proyecto es guiar a una compañía aérea a determinar los factores importantes que influyen en la satisfacción del cliente o pasajero de la aerolínea.

Hipótesis

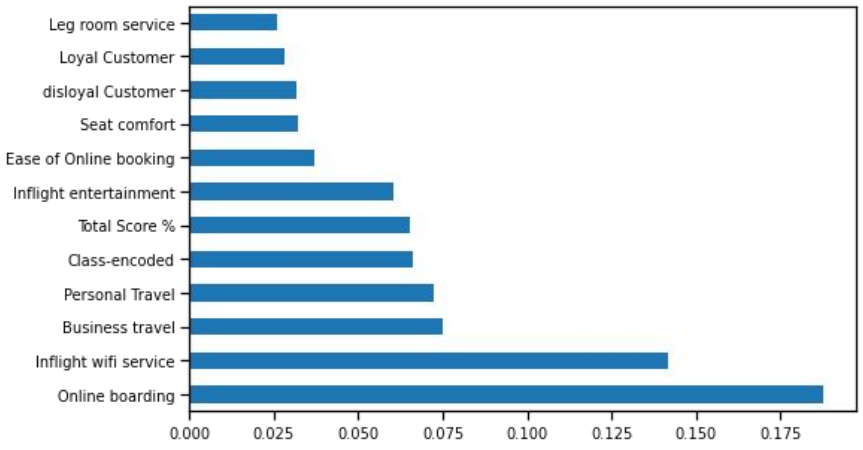
¿Se puede predecir la satisfacción de un pasajero?

¿Existe un patrón, en función de las calificaciones otorgadas por los pasajeros, que refleje la experiencia general del cliente y su satisfacción?

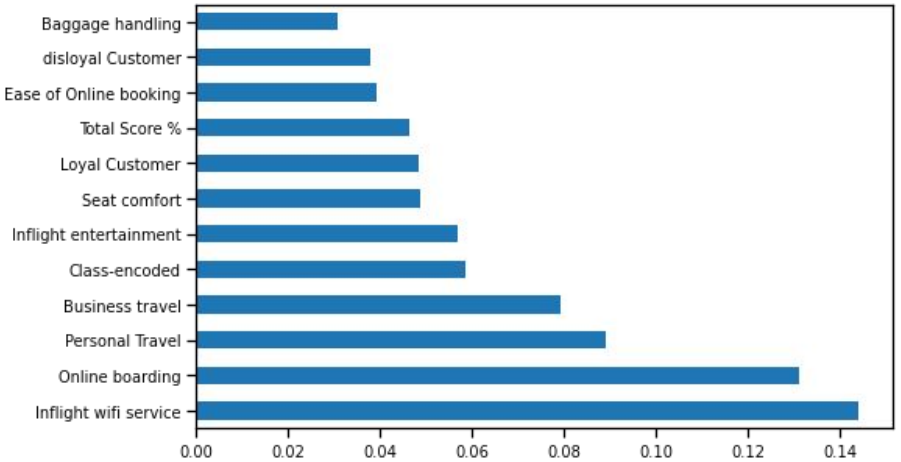
Modelos utilizados

feature importance

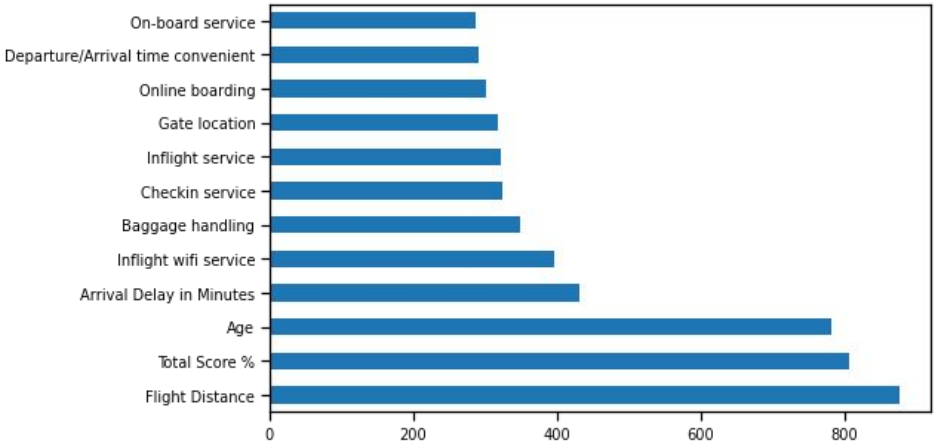
Random Forest



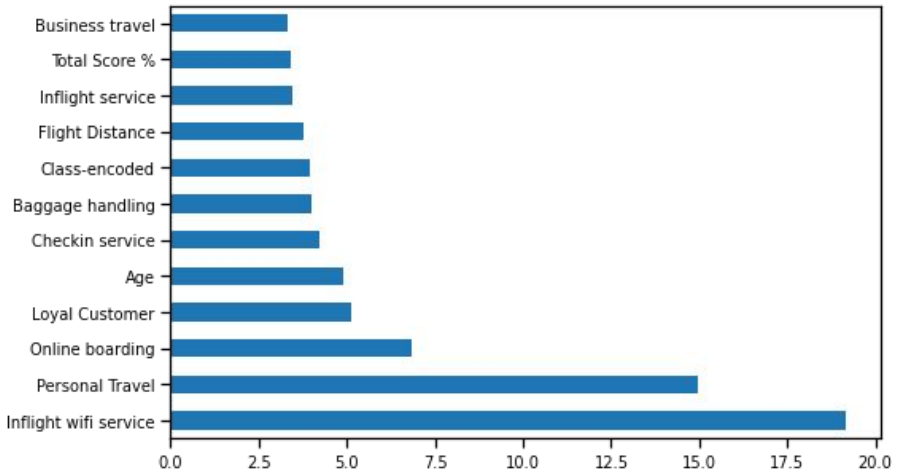
Extra Trees



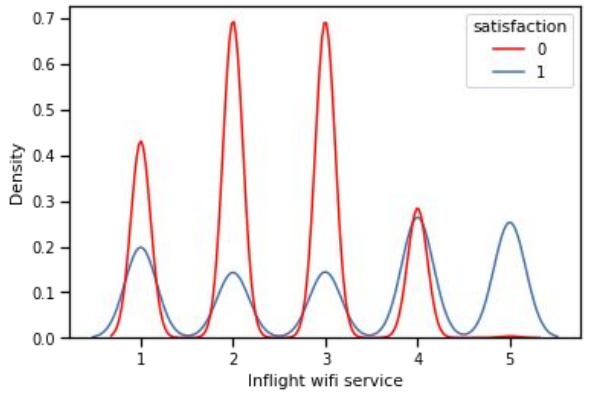
Light Gradient Boosting Machine

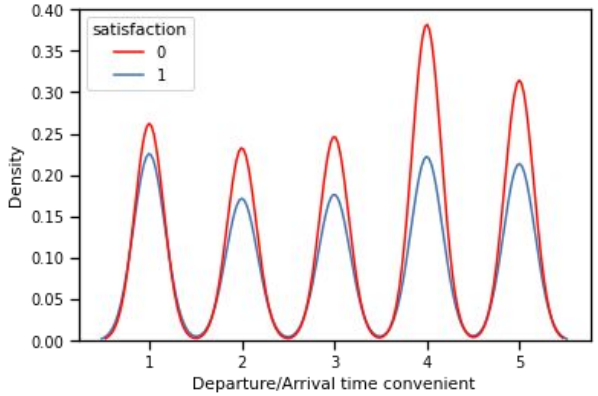


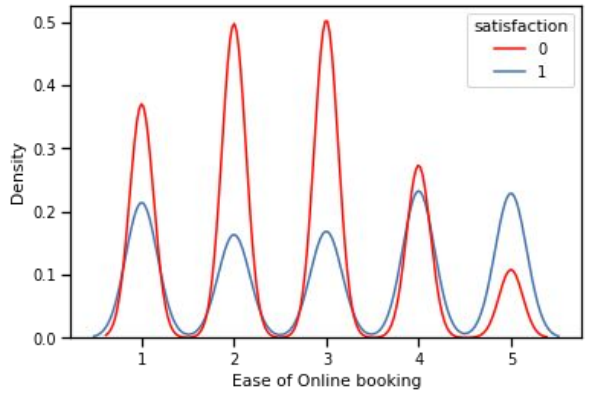
Cat Boost

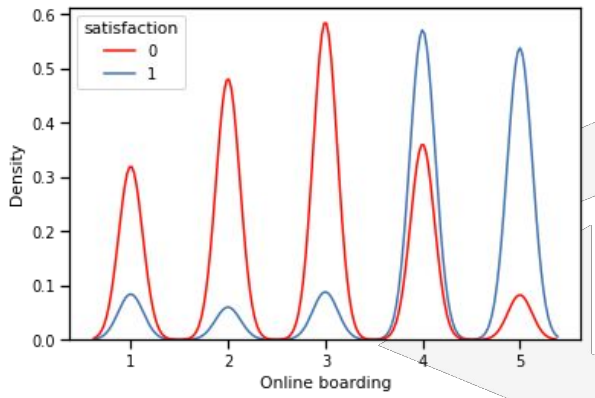


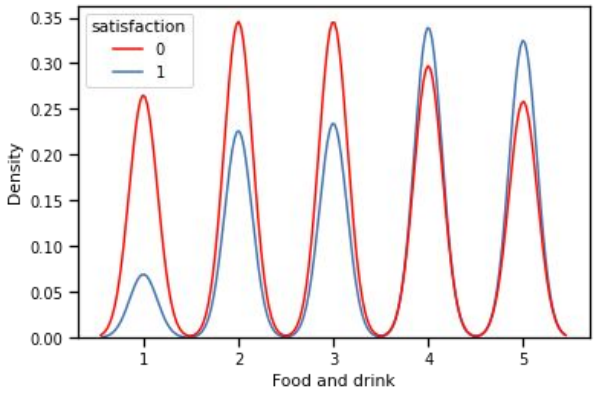
Distribución de las variables

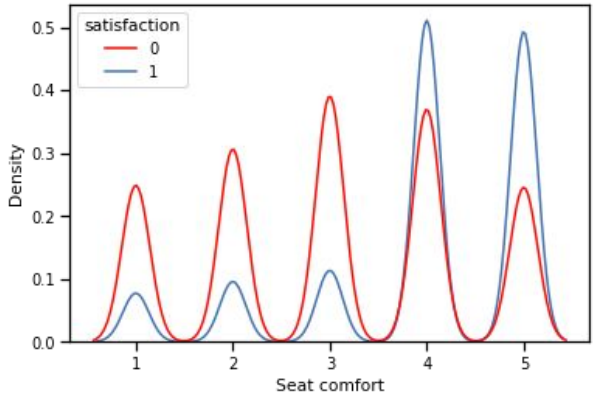


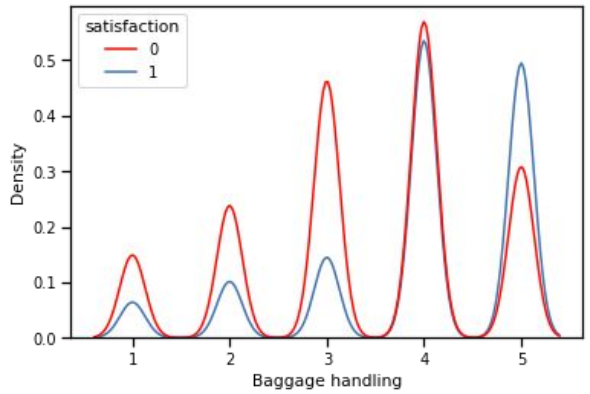


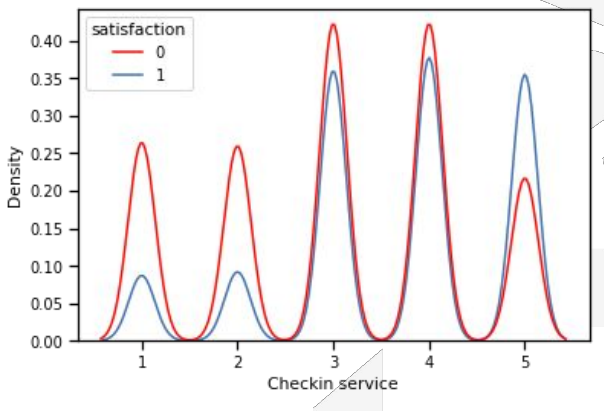


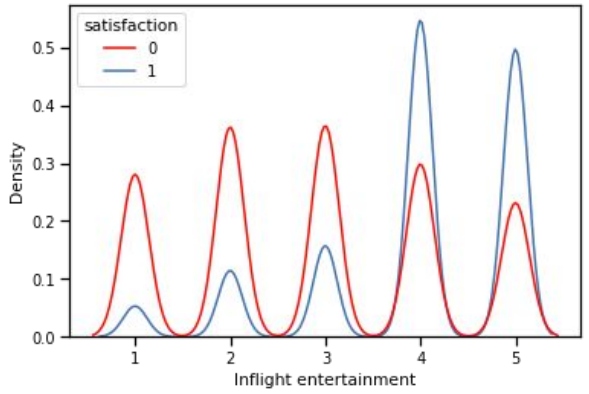


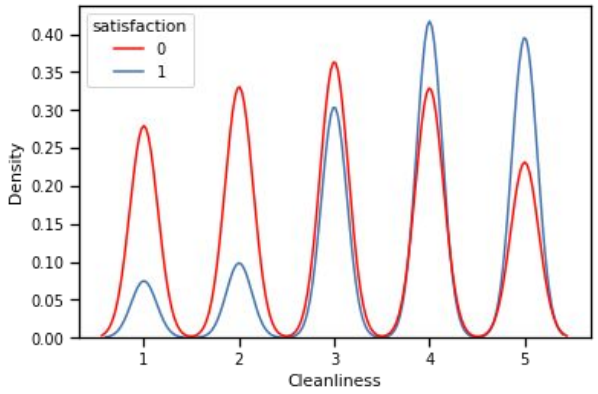


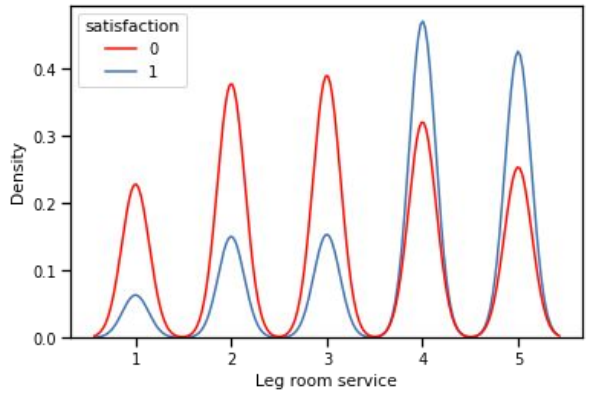


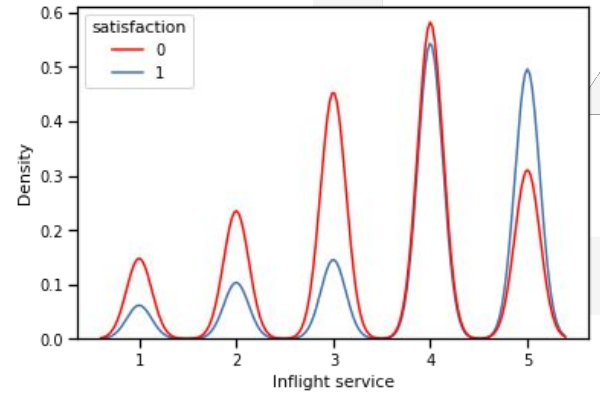






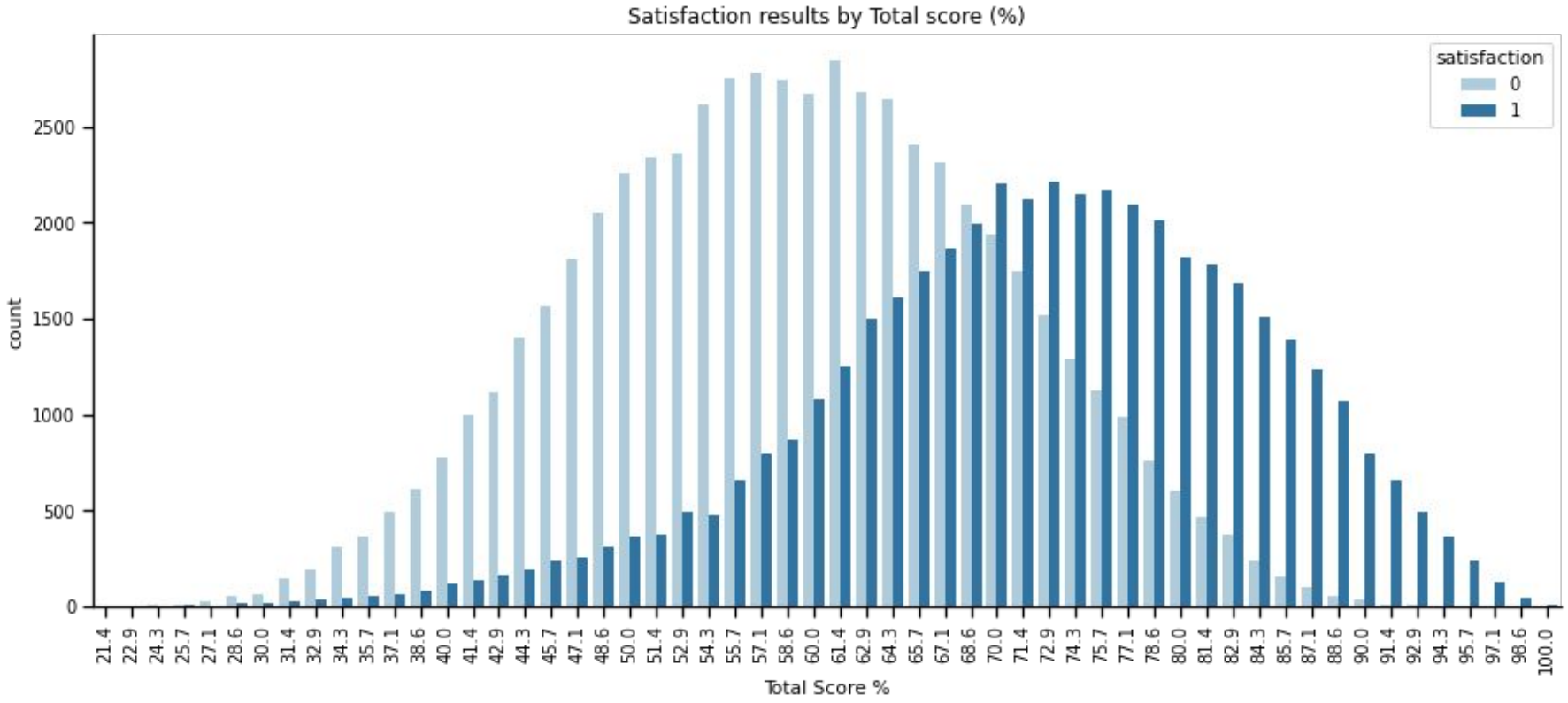






Distribución de las variables

por nivel de satisfacción



ITBA
Instituto Tecnológicc
de Buenos Aires

¡Muchas gracias!