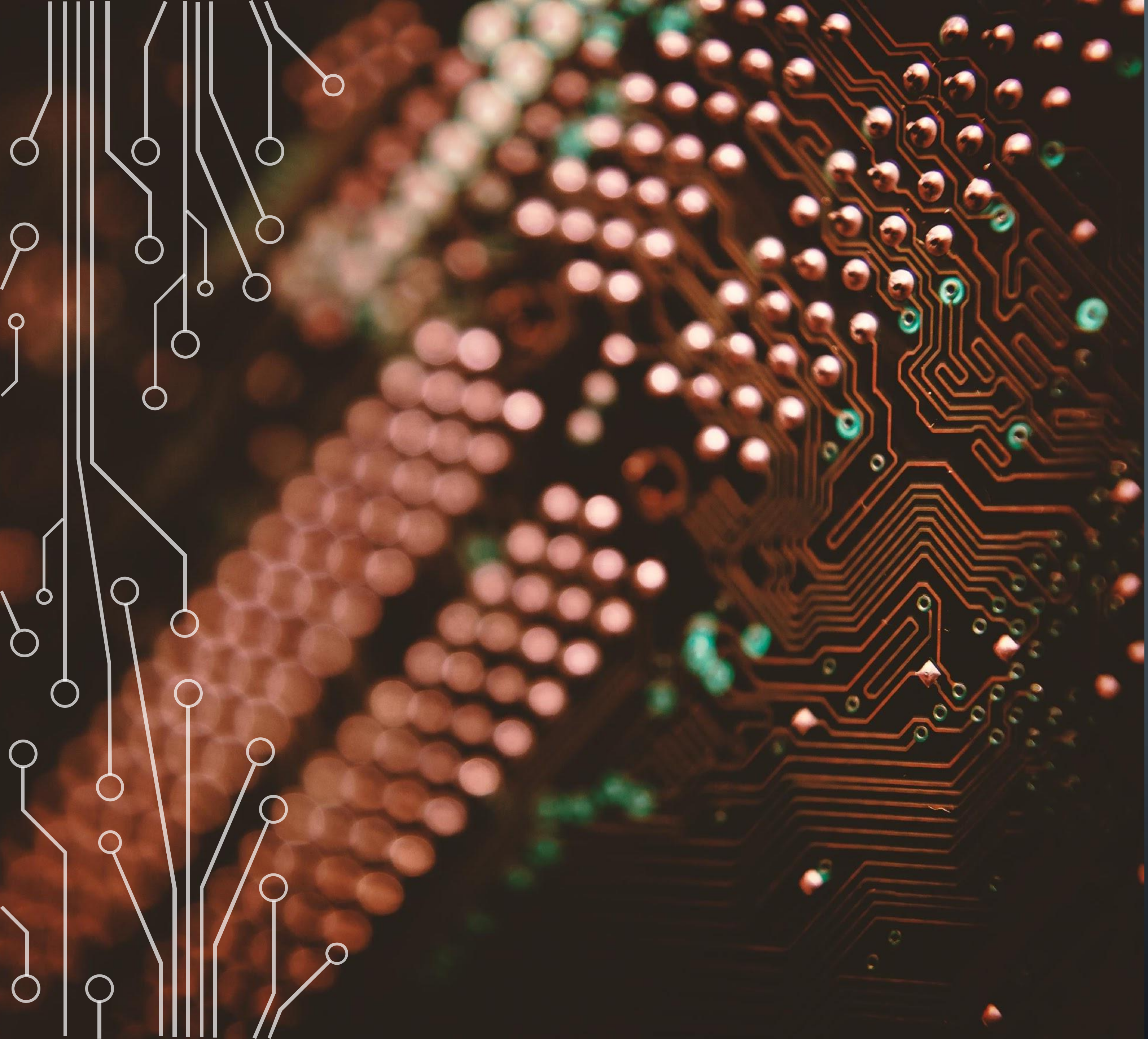
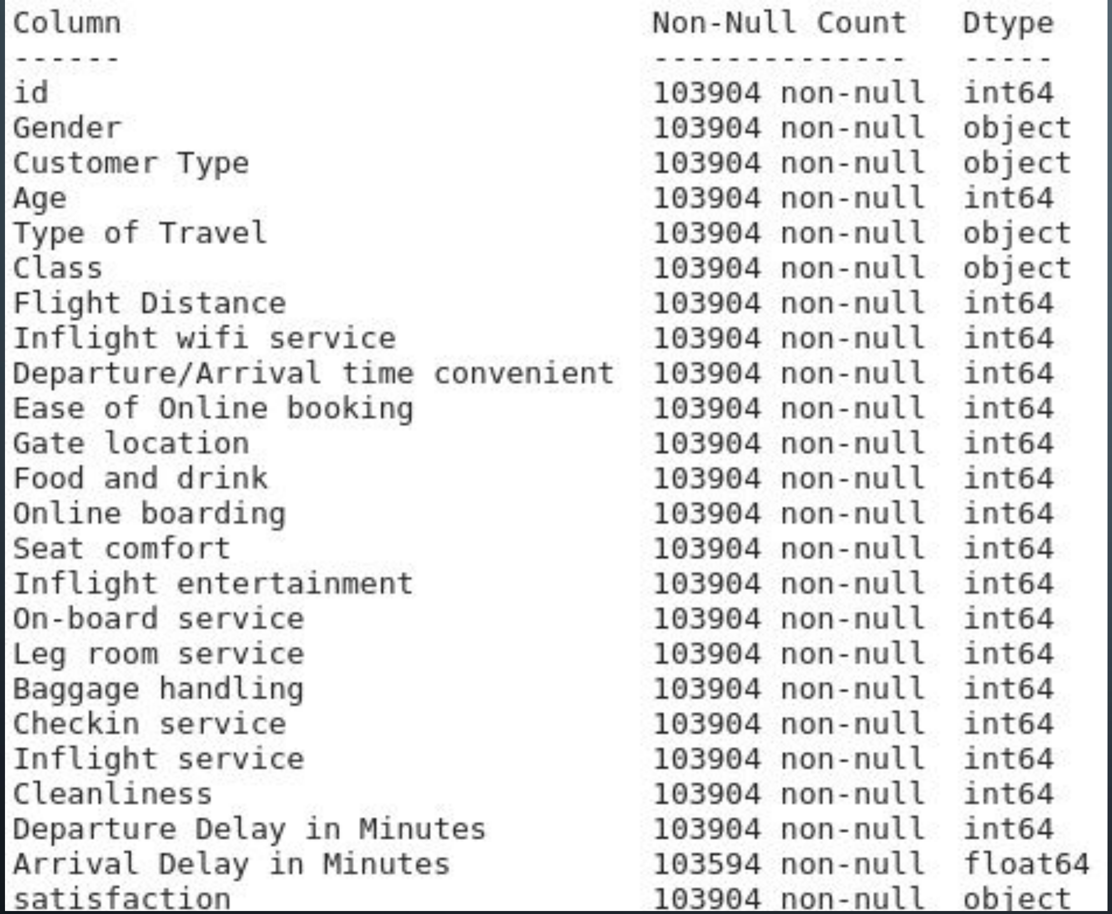
AIRLINE PASSENGER SATISFACTION



PLAN

- Présentation du dataset
- Exploration des données
- Pre-processing
- Choix du modèle
- Optimisation

PRESENTATION DU DATASET



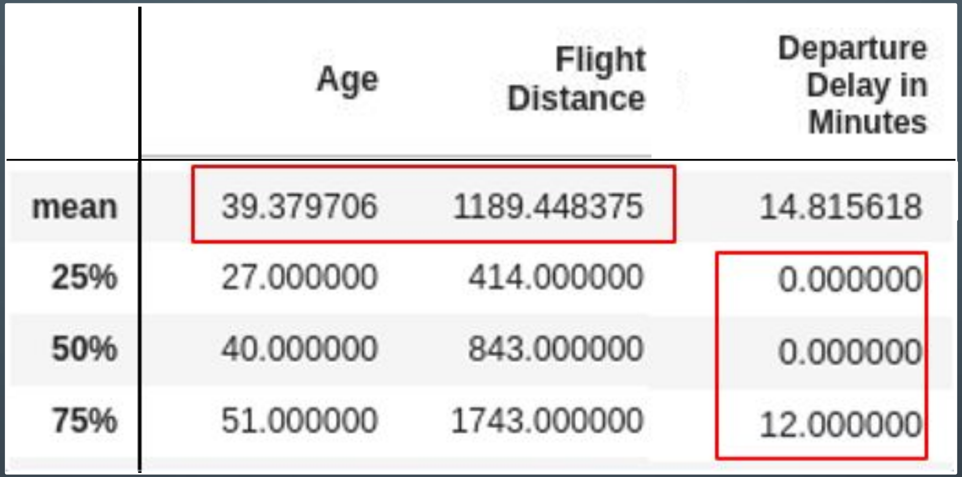
```
Column                             Non-Null Count   Dtype
------                             --------------   -----
id                                 103904 non-null  int64
Gender                             103904 non-null  object
Customer Type                      103904 non-null  object
Age                                103904 non-null  int64
Type of Travel                     103904 non-null  object
Class                              103904 non-null  object
Flight Distance                    103904 non-null  int64
Inflight wifi service              103904 non-null  int64
Departure/Arrival time convenient  103904 non-null  int64
Ease of Online booking             103904 non-null  int64
Gate location                      103904 non-null  int64
Food and drink                     103904 non-null  int64
Online boarding                    103904 non-null  int64
Seat comfort                       103904 non-null  int64
Inflight entertainment             103904 non-null  int64
On-board service                   103904 non-null  int64
Leg room service                   103904 non-null  int64
Baggage handling                   103904 non-null  int64
Checkin service                    103904 non-null  int64
Inflight service                   103904 non-null  int64
Cleanliness                        103904 non-null  int64
Departure Delay in Minutes         103904 non-null  int64
Arrival Delay in Minutes           103594 non-null  float64
satisfaction                       103904 non-null  object
```

- 24 features
- Taille du dataset: 103 904 données

PROBLÉMATIQUE

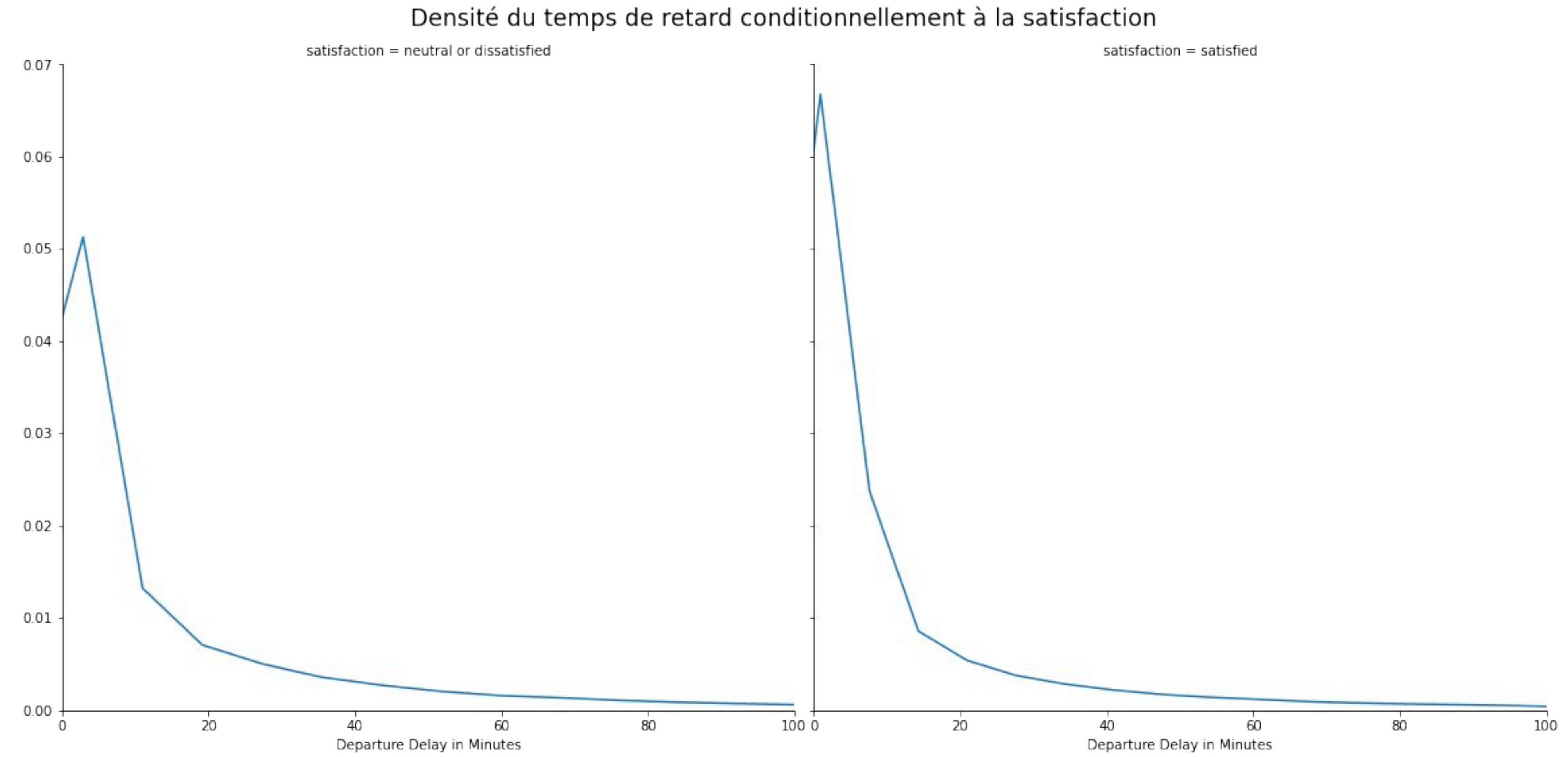
- Prédire la satisfaction d’un client lors d’un vol de la compagnie à partir des données de notre dataset
- Classification
- Recherche du meilleur accuracy

EXPLORATION DES DONNÉES (1)

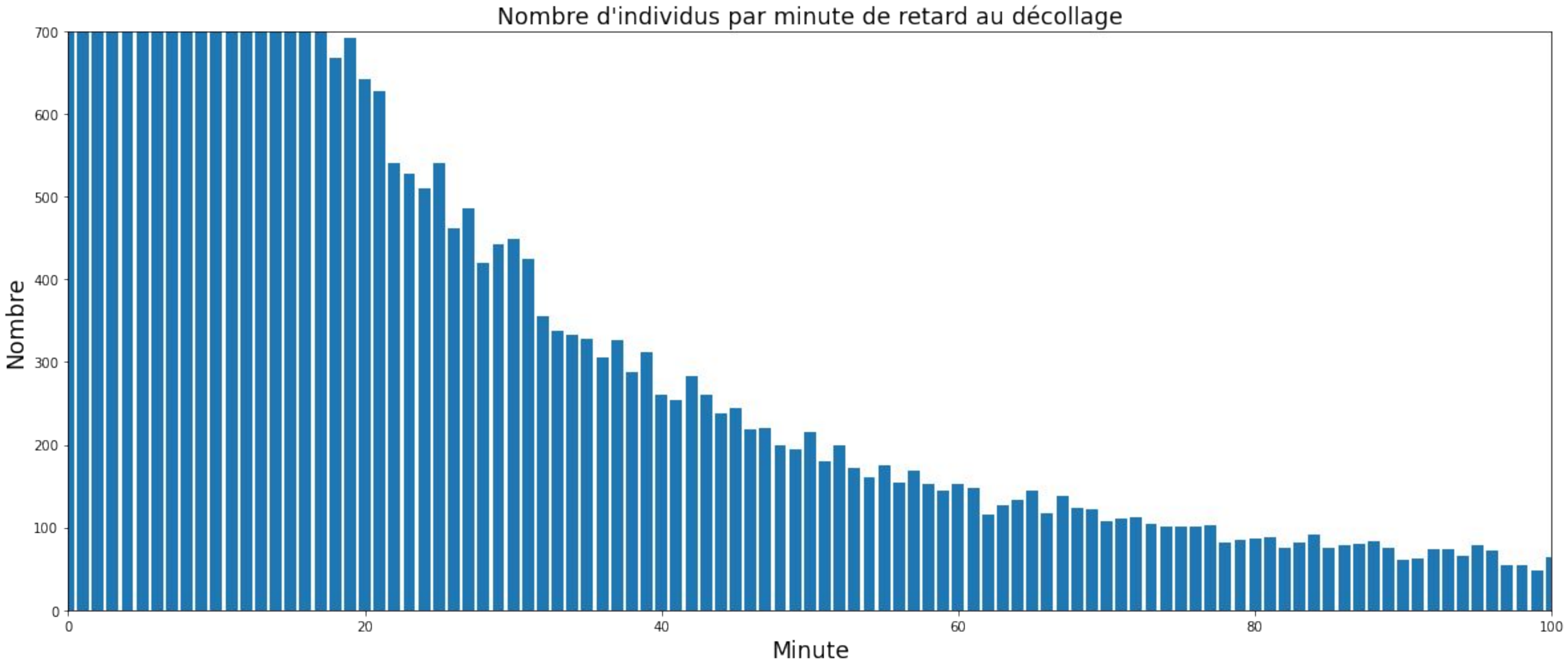


	Age	Flight Distance	Departure Delay in Minutes
mean	39.379706	1189.448375	14.815618
25%	27.000000	414.000000	0.000000
50%	40.000000	843.000000	0.000000
75%	51.000000	1743.000000	12.000000

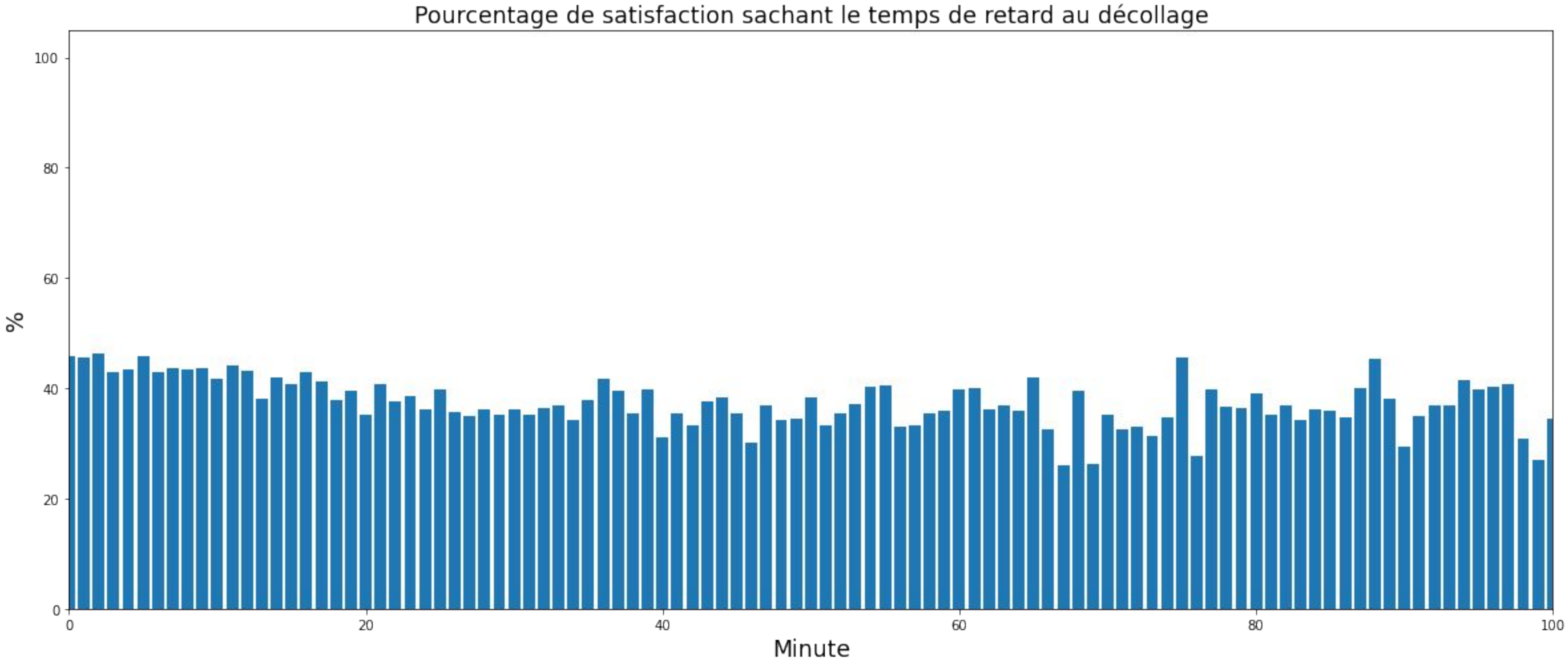
EXPLORATION DES DONNÉES (2)



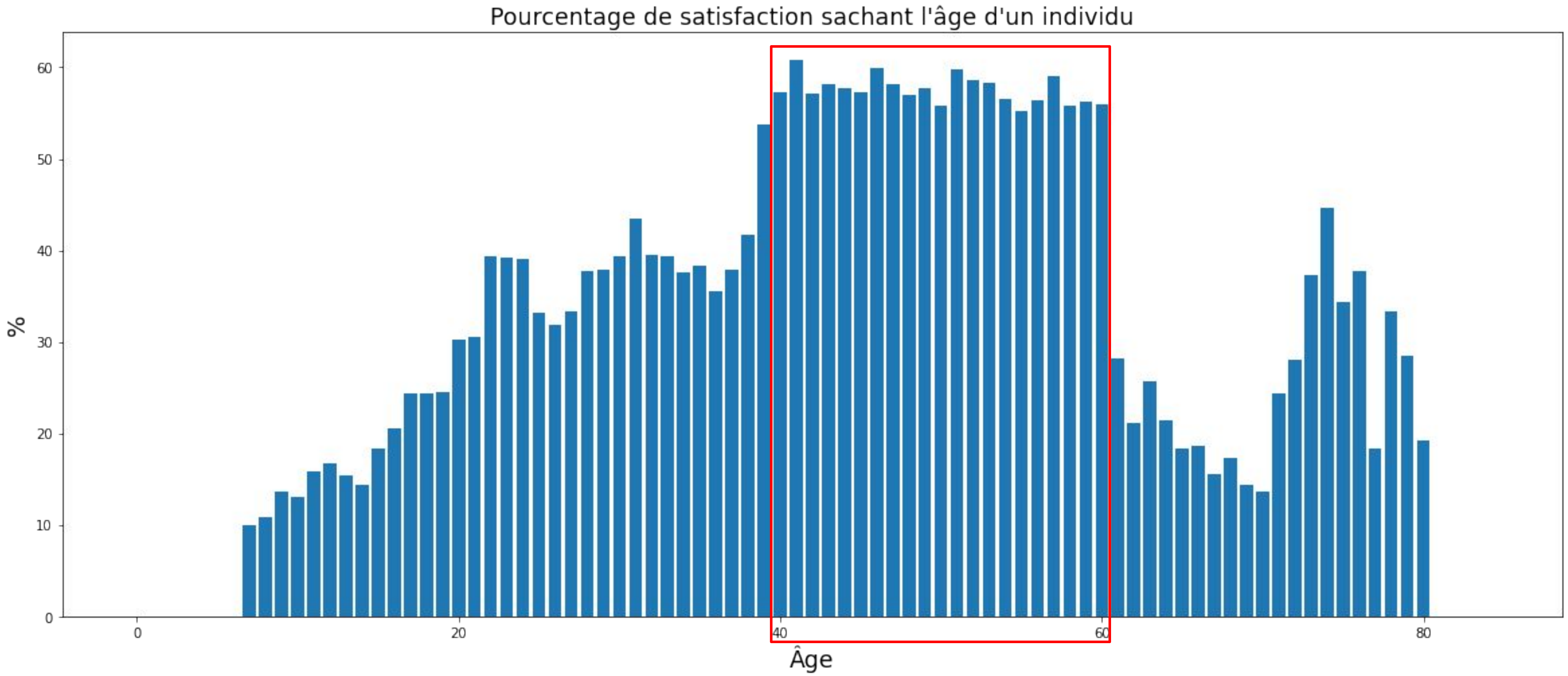
EXPLORATION DES DONNÉES (3)



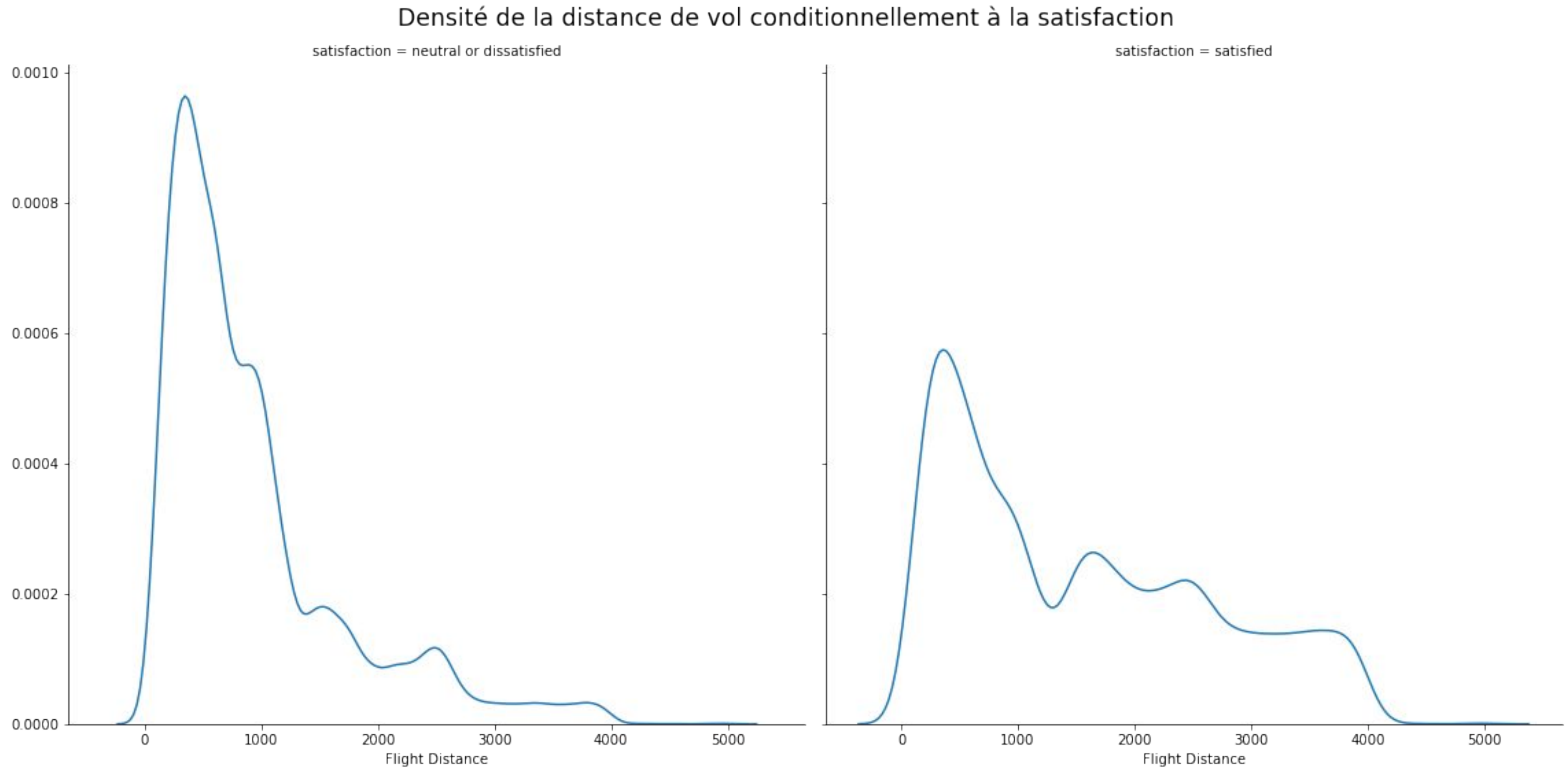
EXPLORATION DES DONNÉES (4)



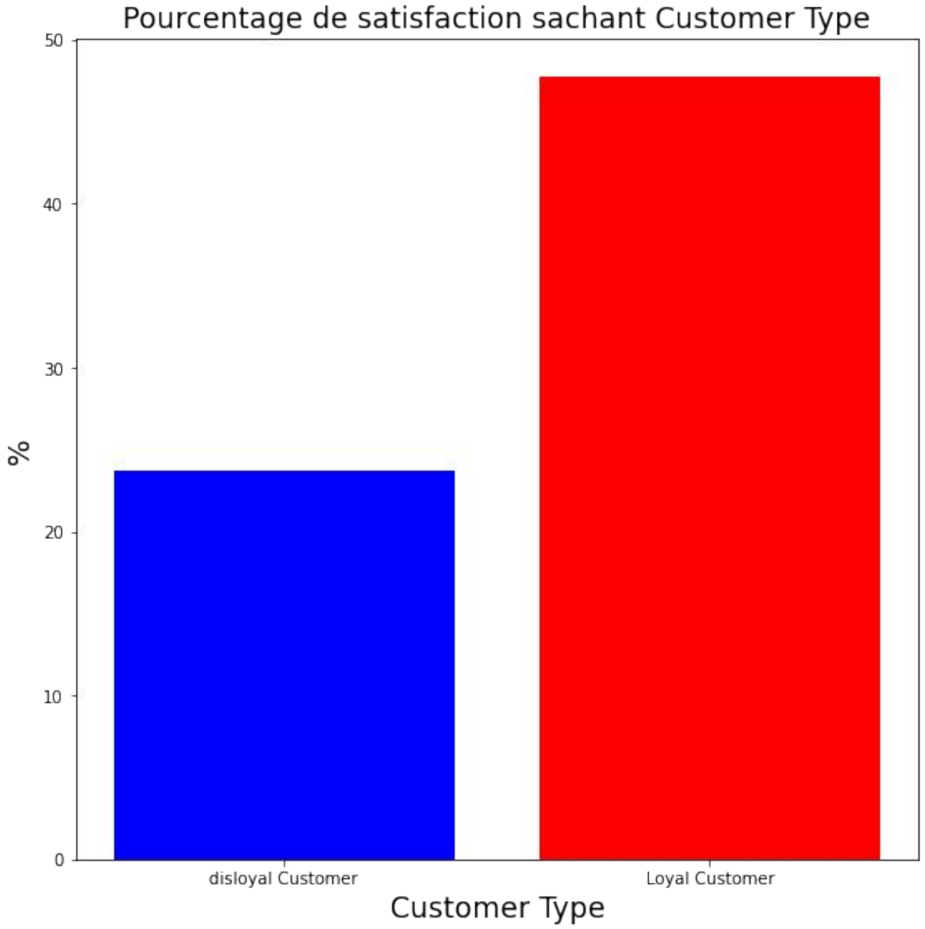
EXPLORATION DES DONNÉES (5)

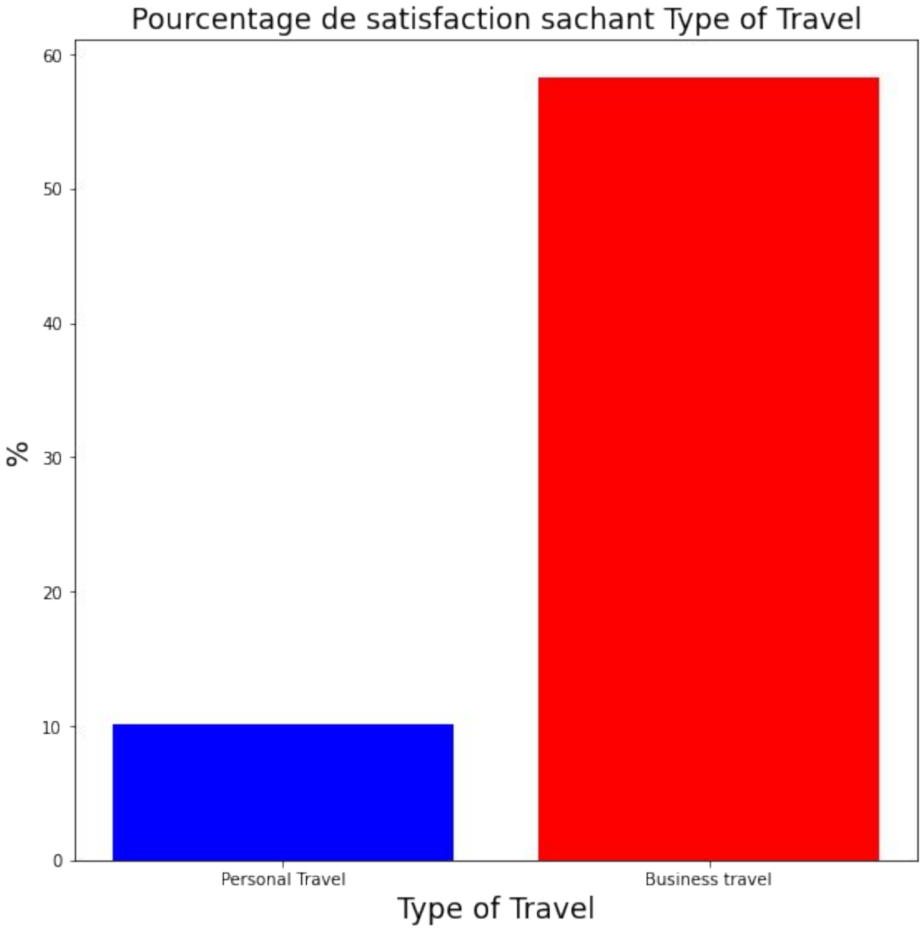


EXPLORATION DES DONNÉES (6)

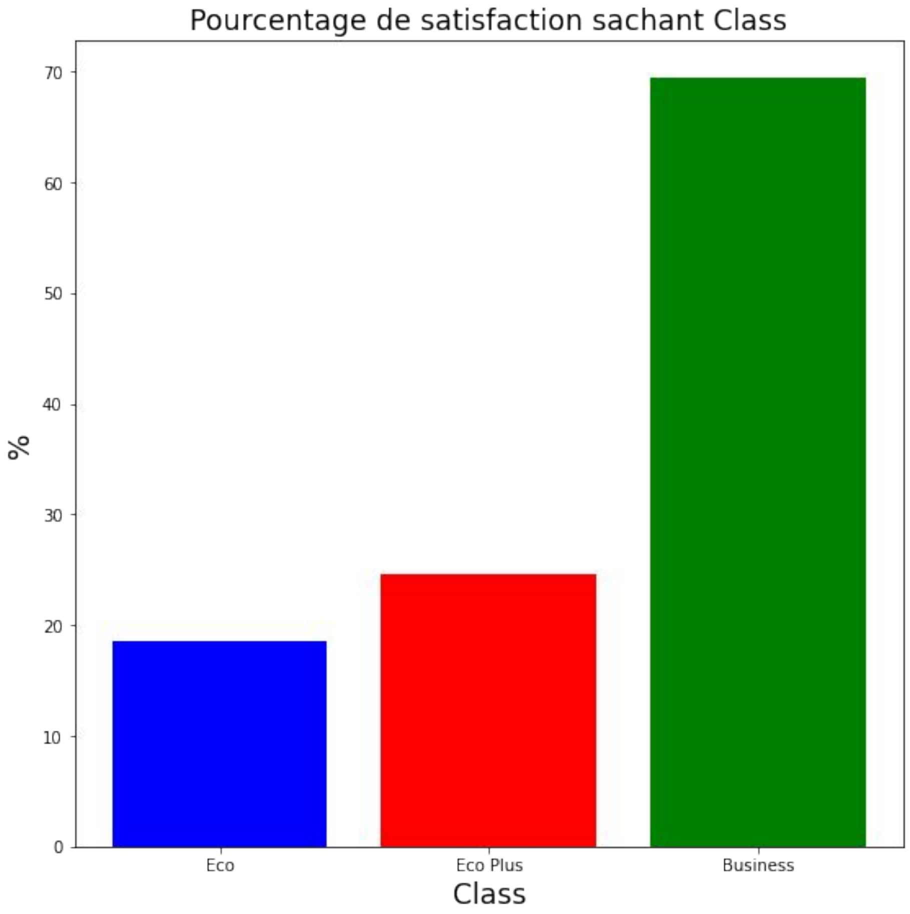


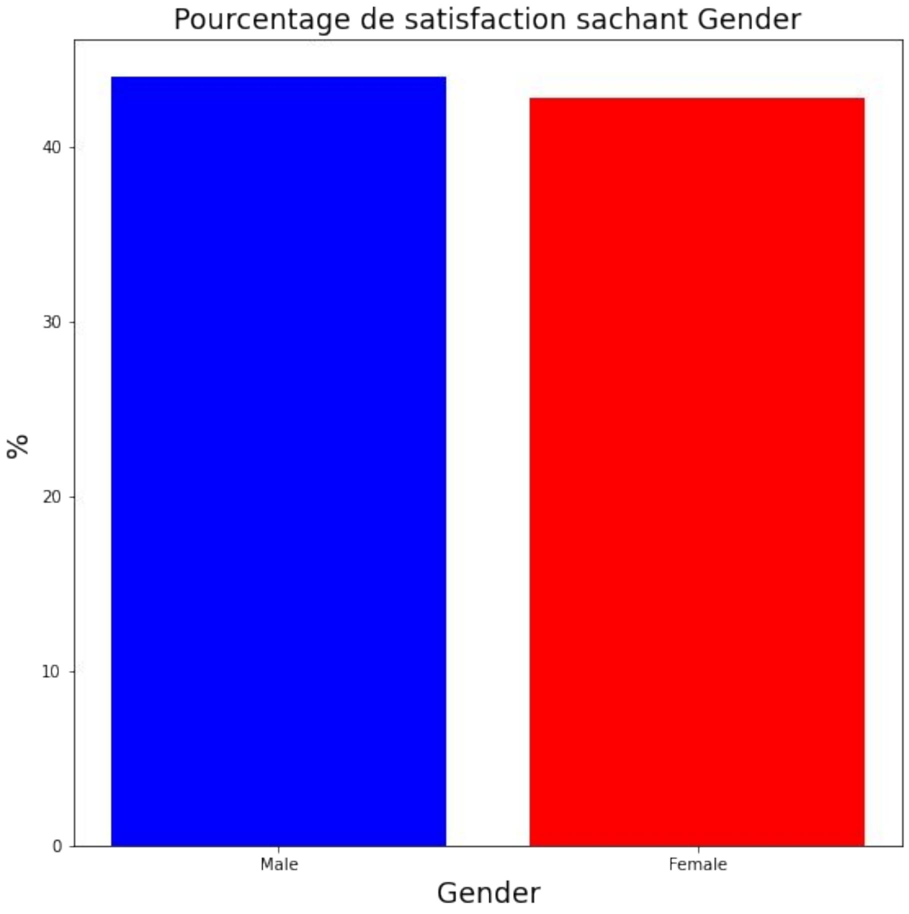
EXPLORATION DES DONNÉES (7)



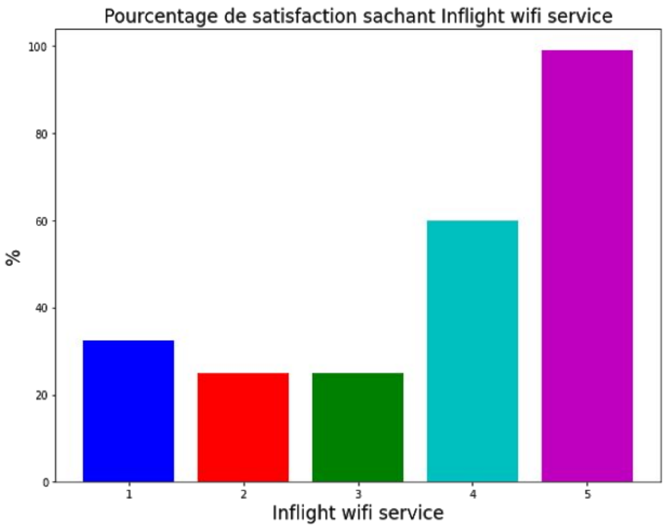


EXPLORATION DES DONNÉES (8)

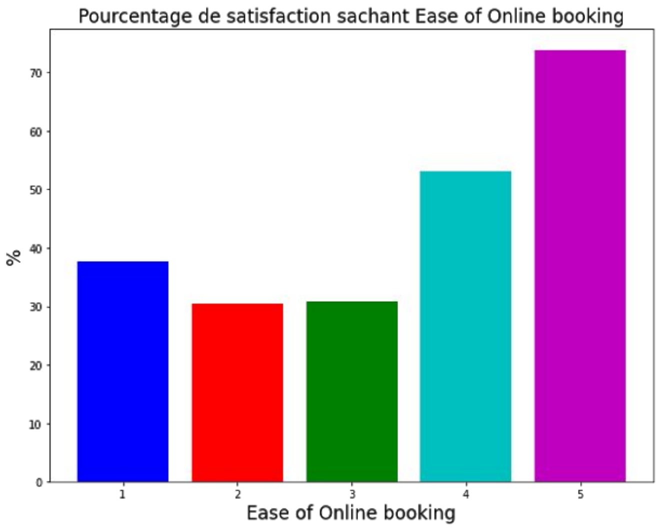

Pourcentage de satisfaction sachant Class


Pourcentage de satisfaction sachant Gender

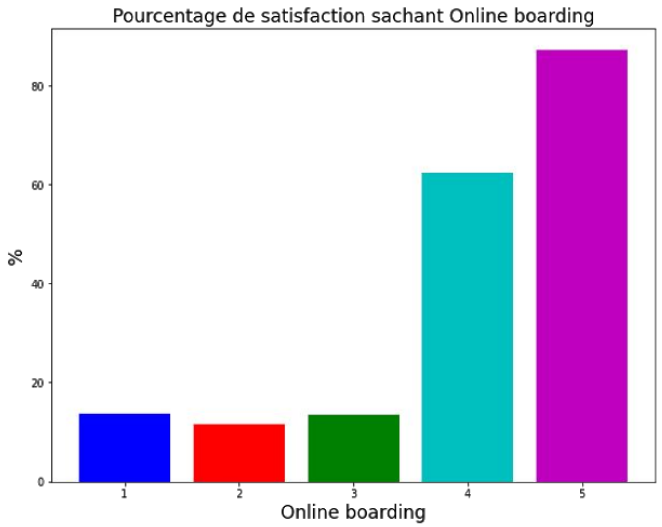
EXPLORATION DES DONNÉES (9)



Pourcentage de satisfaction sachant Inflight wifi service

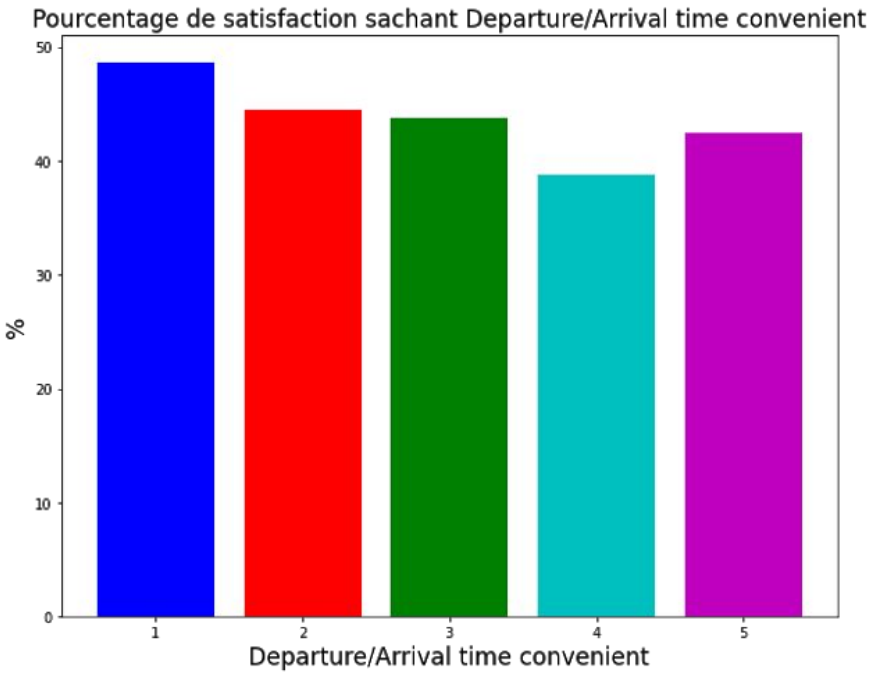


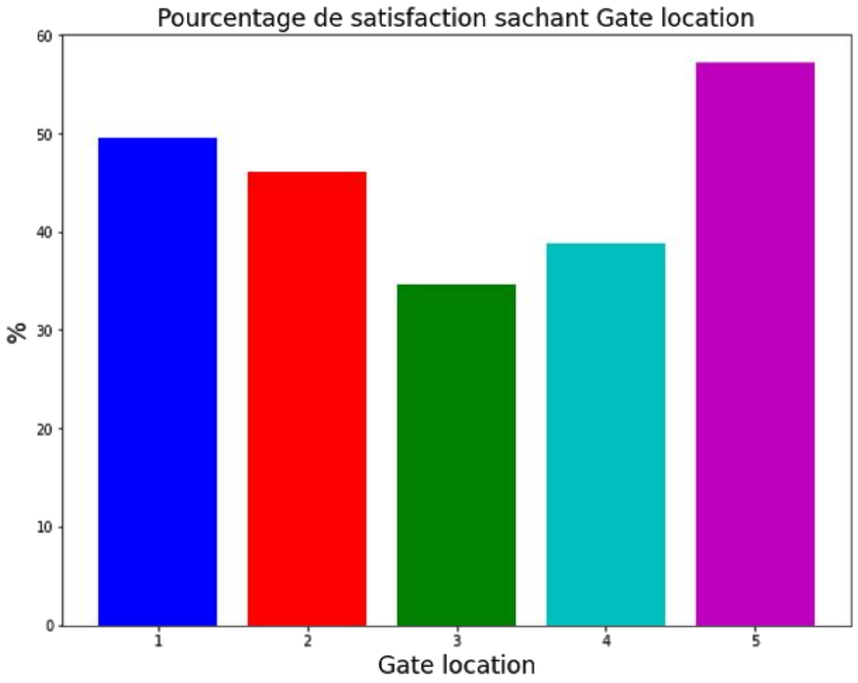
Pourcentage de satisfaction sachant Ease of Online booking

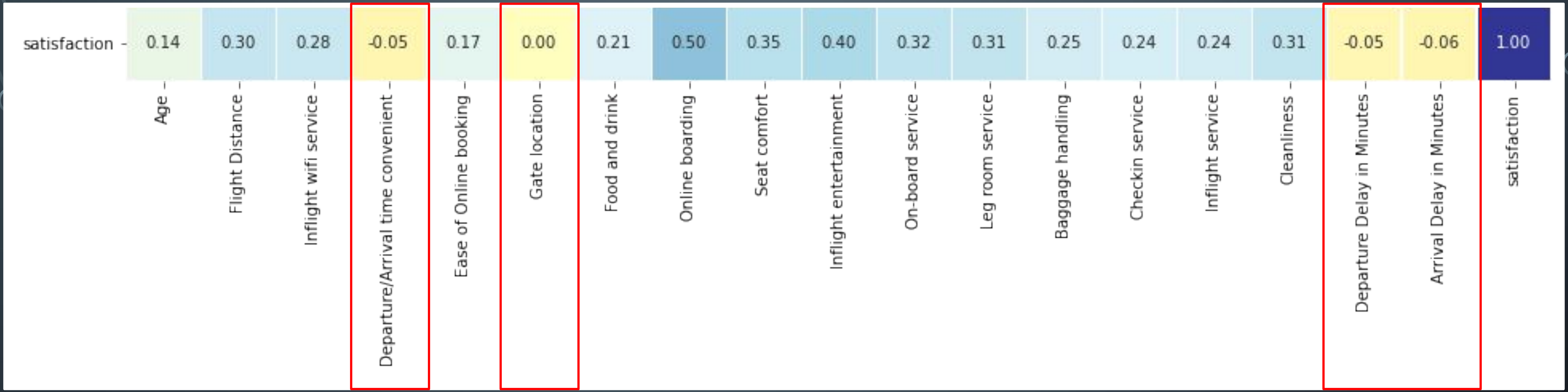


Pourcentage de satisfaction sachant Online boarding

EXPLORATION DES DONNÉES (10)


Pourcentage de satisfaction sachant Departure/Arrival time convenient

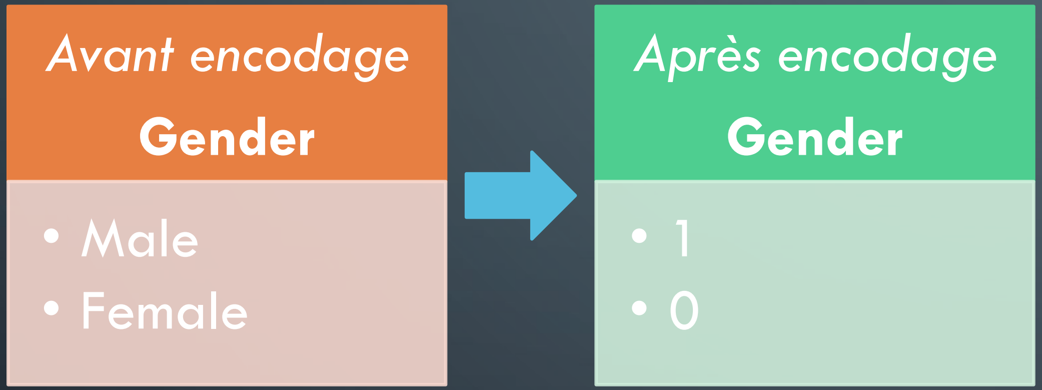

Pourcentage de satisfaction sachant Gate location



FEATURE SELECTION

Features enlevés car coefficient de corrélation bas (<0,1)

PRE-PROCESSING



Avant encodage

Gender

- Male
- Female

Après encodage

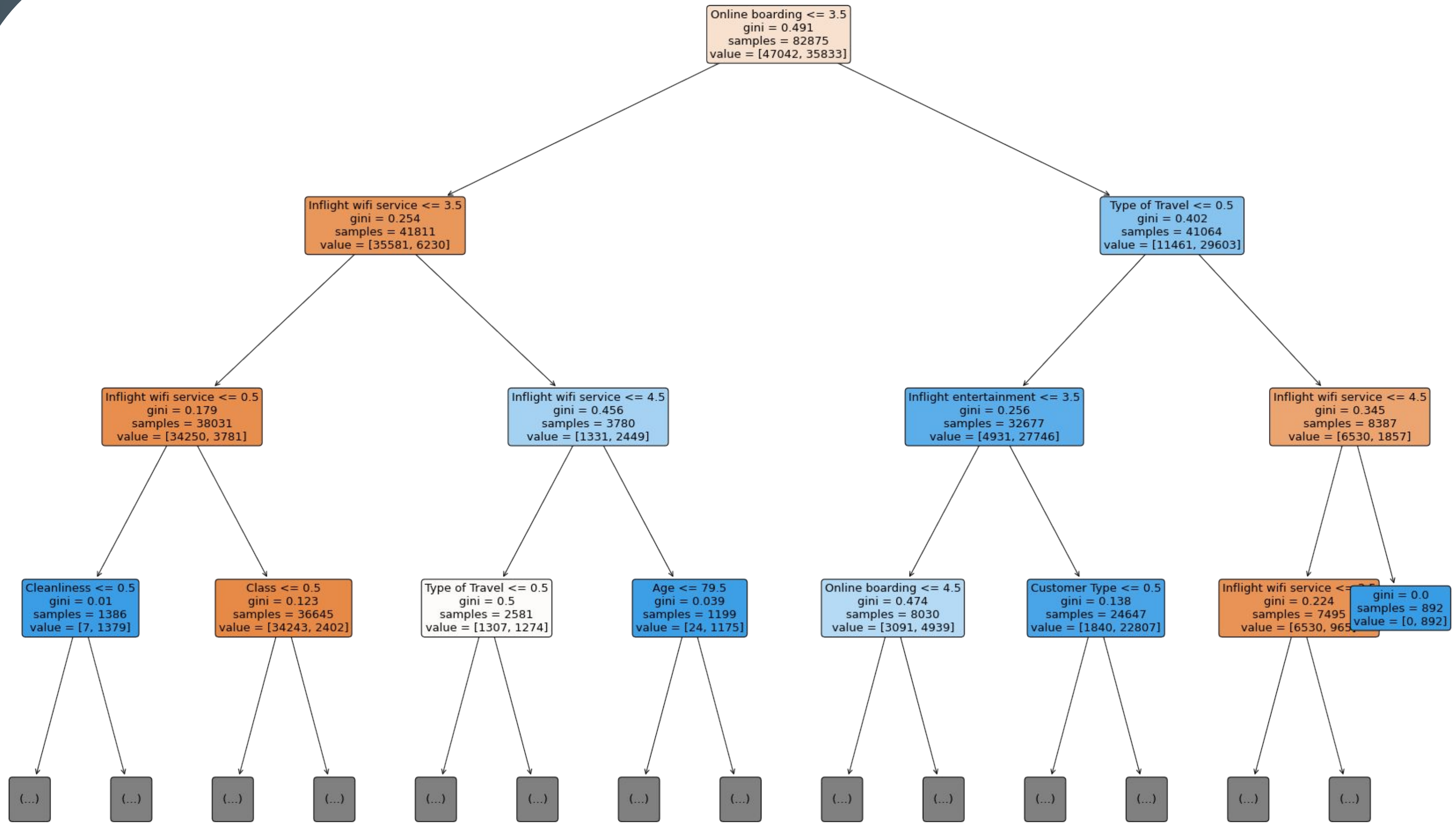
Gender

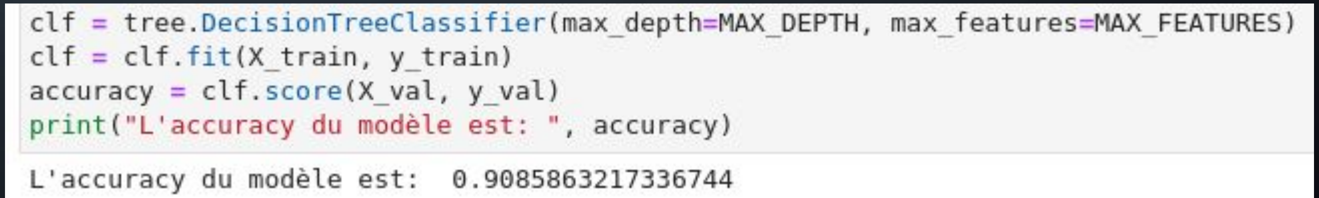
- 1
- 0

- ***Algorithme***: Label Encoder
- ***Fit***: Regarde les catégories qui sont là, associe un nombre à une catégorie (commence toujours par 0)
- ***Transform***: Retourne le tableau de catégories encodées

CHOIX DU MODÈLE

DECISION TREE

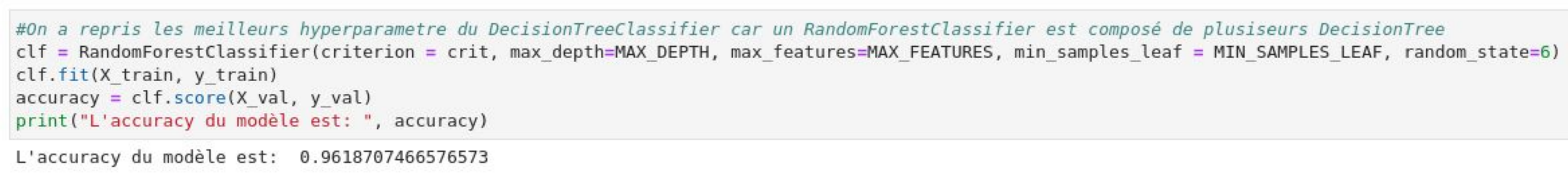




```
clf = tree.DecisionTreeClassifier(max_depth=MAX_DEPTH, max_features=MAX_FEATURES)
clf = clf.fit(X_train, y_train)
accuracy = clf.score(X_val, y_val)
print("L'accuracy du modèle est: ", accuracy)
```

```
L'accuracy du modèle est:  0.9085863217336744
```

RANDOM FOREST



```
#On a repris les meilleurs hyperparametre du DecisionTreeClassifier car un RandomForestClassifier est composé de plusiseurs DecisionTree
clf = RandomForestClassifier(criterion = crit, max_depth=MAX_DEPTH, max_features=MAX_FEATURES, min_samples_leaf = MIN_SAMPLES_LEAF, random_state=6)
clf.fit(X_train, y_train)
accuracy = clf.score(X_val, y_val)
print("L'accuracy du modèle est: ", accuracy)
```

```
L'accuracy du modèle est:  0.9618707466576573
```

OPTIMISATION

GridSearchCV

Decision Tree

Hyperparamètres à optimiser

- max_depth (range 1 à 20)
- criterion (gini ou entropy)
- min_sample_leaf (1 à 10)

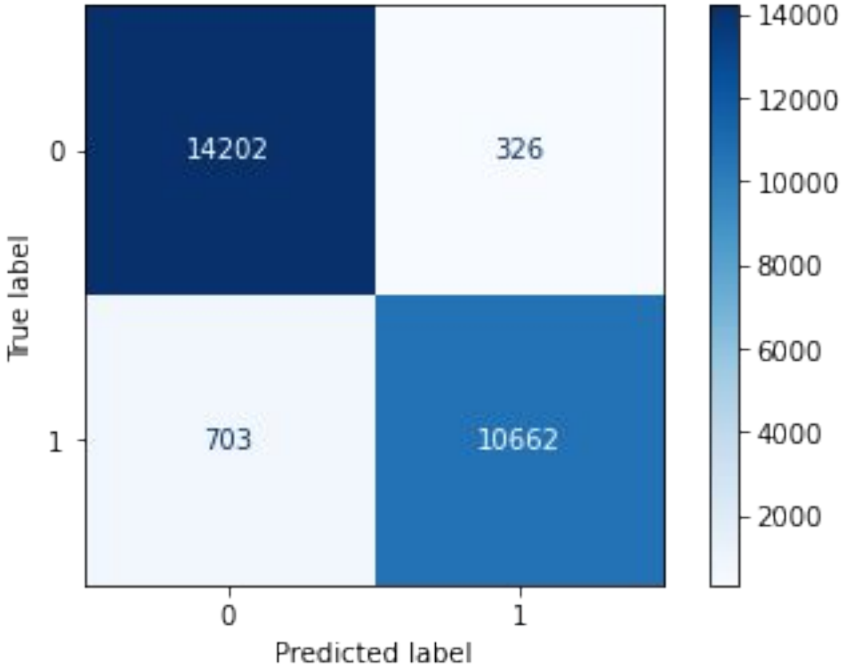
Random Forest

Hyperparamètres à optimiser

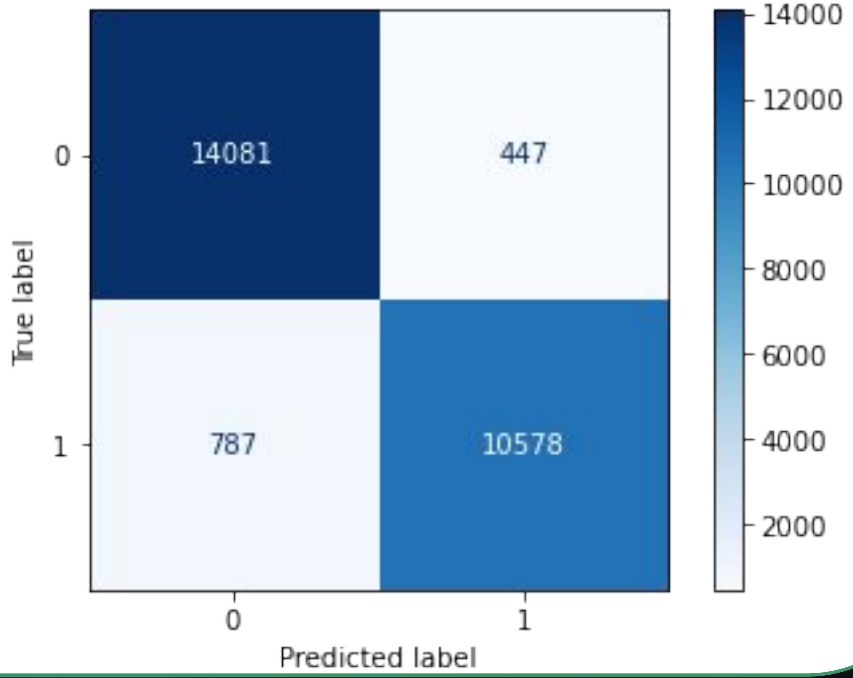
- n_estimators (10 à 1000)

VISUALISATION DES RESULTATS

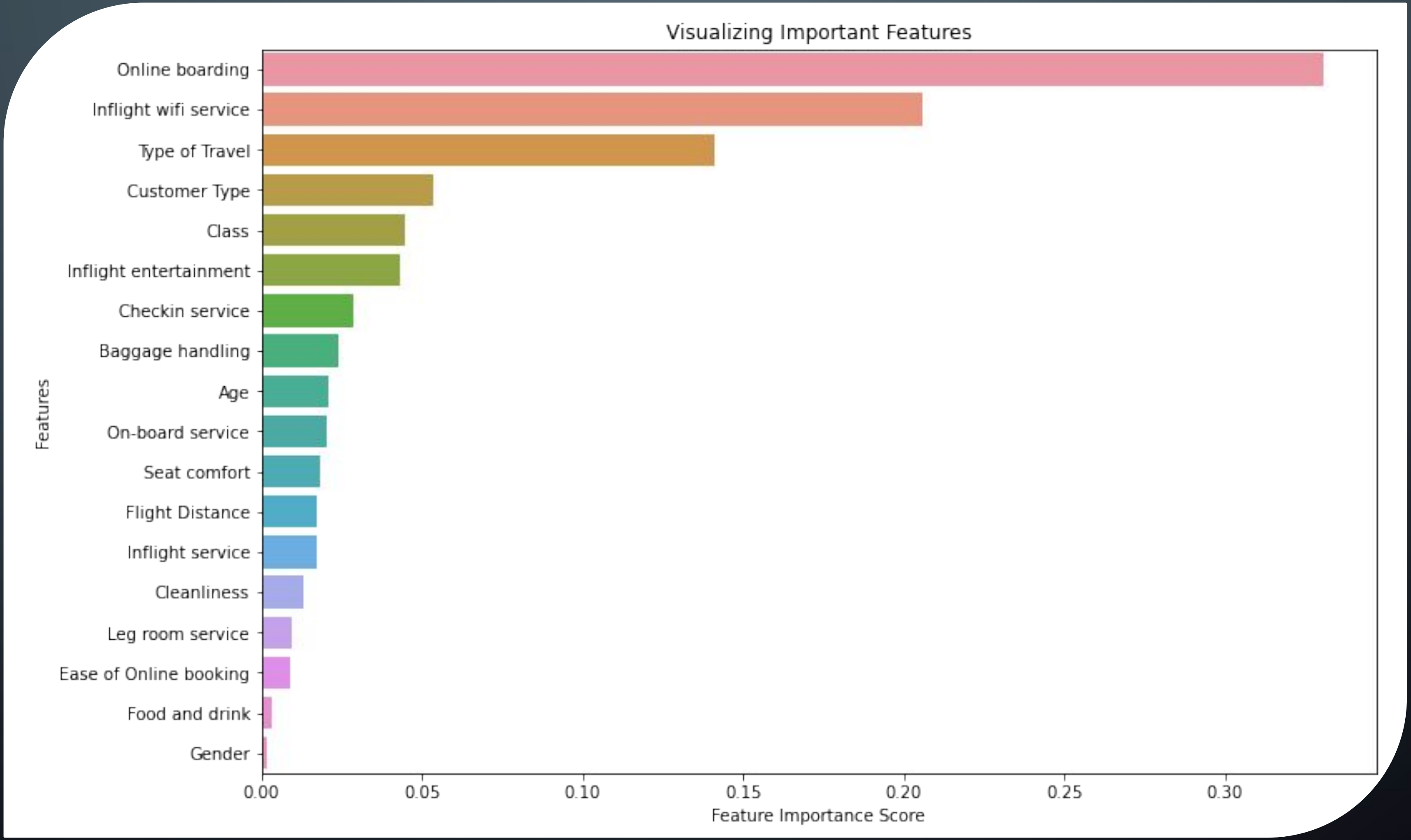
Matrice de confusion du Random Forest



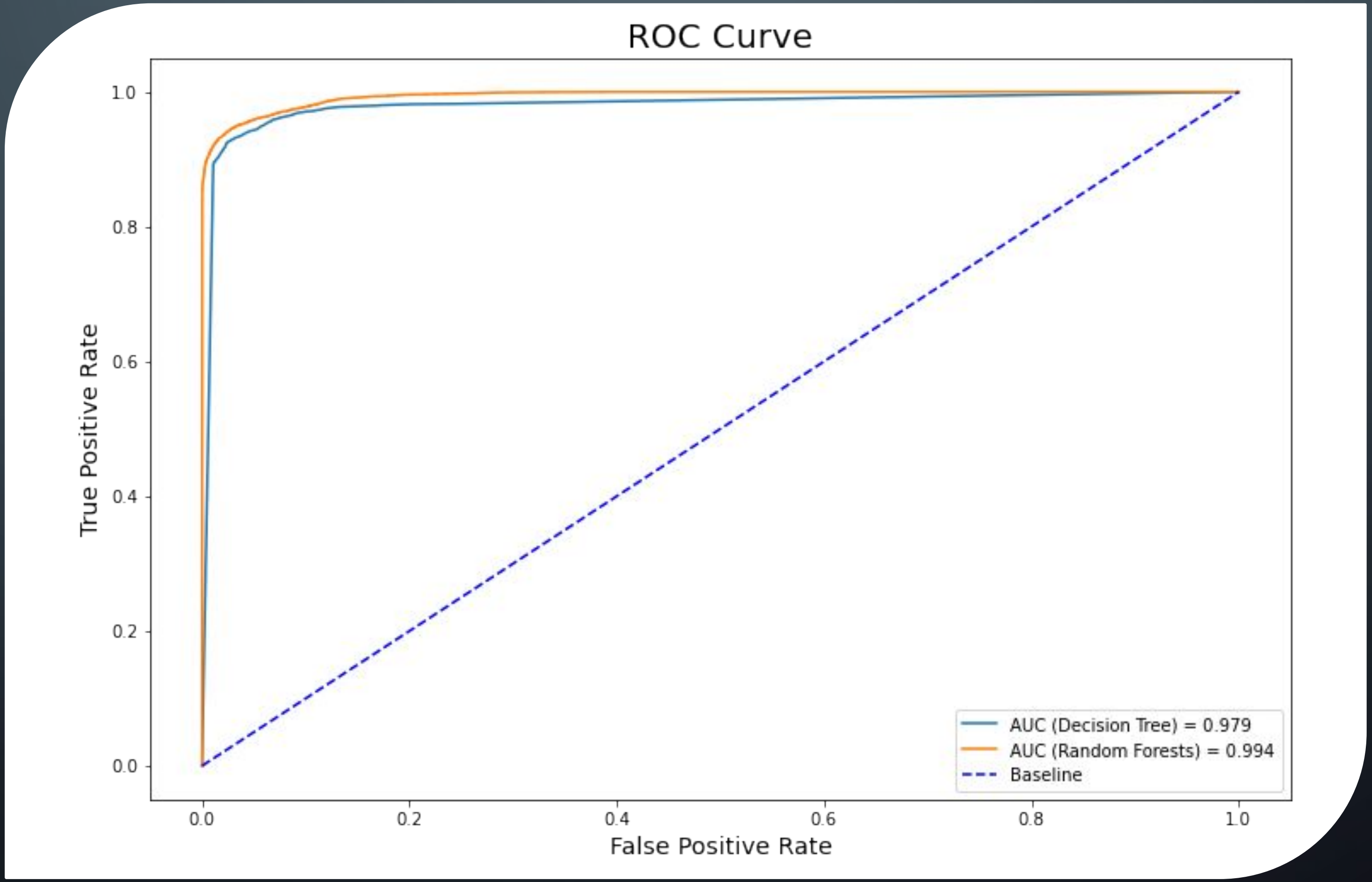
Matrice de confusion du Decision Tree Classifier



IMPORTANCE DES FEATURES



COMPARAISON DES PERFORMANCES AVEC LES COURBES ROC



CONCLUSION