ECE PARIS · LYON
ÉCOLE D'INGÉNIEURS

Rapport – Airline Passenger satisfaction

Sofiane EL FARTASS
Module : Machine learning II
Enseignant : Mouadh YAGOUBI

INTRODUCTION

Le défi du challenge Kaggle «*Airline Passenger satisfaction* », implique la création d'un modèle de machine learning capables de prédire le niveau de satisfaction des passagers à partir de données relatives à leur expérience de vol.

L'objectif était d'utiliser un algorithme de deep learning pour exploiter la complexité des données et capturer les relations non linéaires entre les caractéristiques des vols et la satisfaction des passagers. Pour ce faire, nous avons exploré différentes architectures de réseaux neuronaux profonds et ajusté les hyperparamètres pour obtenir un modèle performant.

Ce rapport présente de manière concise la méthodologie adoptée afin de résoudre au mieux ce challenge.

Lien du challenge : https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

Score obtenu :

```
812/812 [==============================] - 1s 1ms/step - loss: 0.0934 - accuracy: 0.9602
Test accuracy: 0.9601555466651917
Test loss: 0.09338445961475372
```

ANALYSE DES DONNÉES

La première étape fut l'analyse des données, afin de comprendre les informations contenues dans ce dataset. Tout d'abord, nous avons commencé par explorer la structure globale du jeu de données en examinant sa forme à l'aide de la fonction ***train.shape***, qui nous a donné le nombre de lignes et de colonnes. Cette première étape nous a permis d'avoir une idée de l'étendue de nos données et la volumétrie du dataset.

Ensuite, nous avons utilisé **train.head()** pour afficher les premières lignes du jeu de données. Cela nous a fourni un aperçu des variables disponibles et de leurs valeurs. Cette exploration préliminaire nous a permis d'identifier les caractéristiques principales de notre ensemble de données et de commencer à formuler des hypothèses sur les relations entre ces caractéristiques et la satisfaction des passagers.

Pour obtenir des informations plus détaillées sur les types de données et les valeurs manquantes éventuelles, nous avons utilisé ***train.info()***. Cette fonction nous a fourni des informations sur les types de données de chaque colonne ainsi que le nombre de valeurs non nulles. Cette étape était essentielle pour préparer nos données à être utilisées dans nos analyses ultérieures.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Unnamed: 0                        103904 non-null  int64
 1   id                                103904 non-null  int64
 2   Gender                            103904 non-null  object
 3   Customer Type                     103904 non-null  object
 4   Age                               103904 non-null  int64
 5   Type of Travel                    103904 non-null  object
 6   Class                             103904 non-null  object
 7   Flight Distance                   103904 non-null  int64
 8   Inflight wifi service             103904 non-null  int64
 9   Departure/Arrival time convenient 103904 non-null  int64
 10  Ease of Online booking            103904 non-null  int64
 11  Gate location                     103904 non-null  int64
 12  Food and drink                    103904 non-null  int64
 13  Online boarding                   103904 non-null  int64
 14  Seat comfort                      103904 non-null  int64
 15  Inflight entertainment            103904 non-null  int64
 16  On-board service                  103904 non-null  int64
 17  Leg room service                  103904 non-null  int64
 18  Baggage handling                  103904 non-null  int64
 19  Checkin service                   103904 non-null  int64
...
 23  Arrival Delay in Minutes          103594 non-null  float64
 24  satisfaction                      103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
```

Figure 1 : Output train.info()

Enfin, pour explorer les relations entre les variables et identifier les corrélations potentielles, nous avons calculé **la matrice de corrélation** à l'aide de la méthode de **Spearman**. Cette matrice de corrélation nous a permis de quantifier les relations linéaires entre les différentes variables de notre ensemble de données. En visualisant cette matrice de corrélation à l'aide d'un heatmap, nous avons pu identifier les relations les plus importantes et les plus significatives pour notre modèle de prédiction de la satisfaction des passagers aériens.

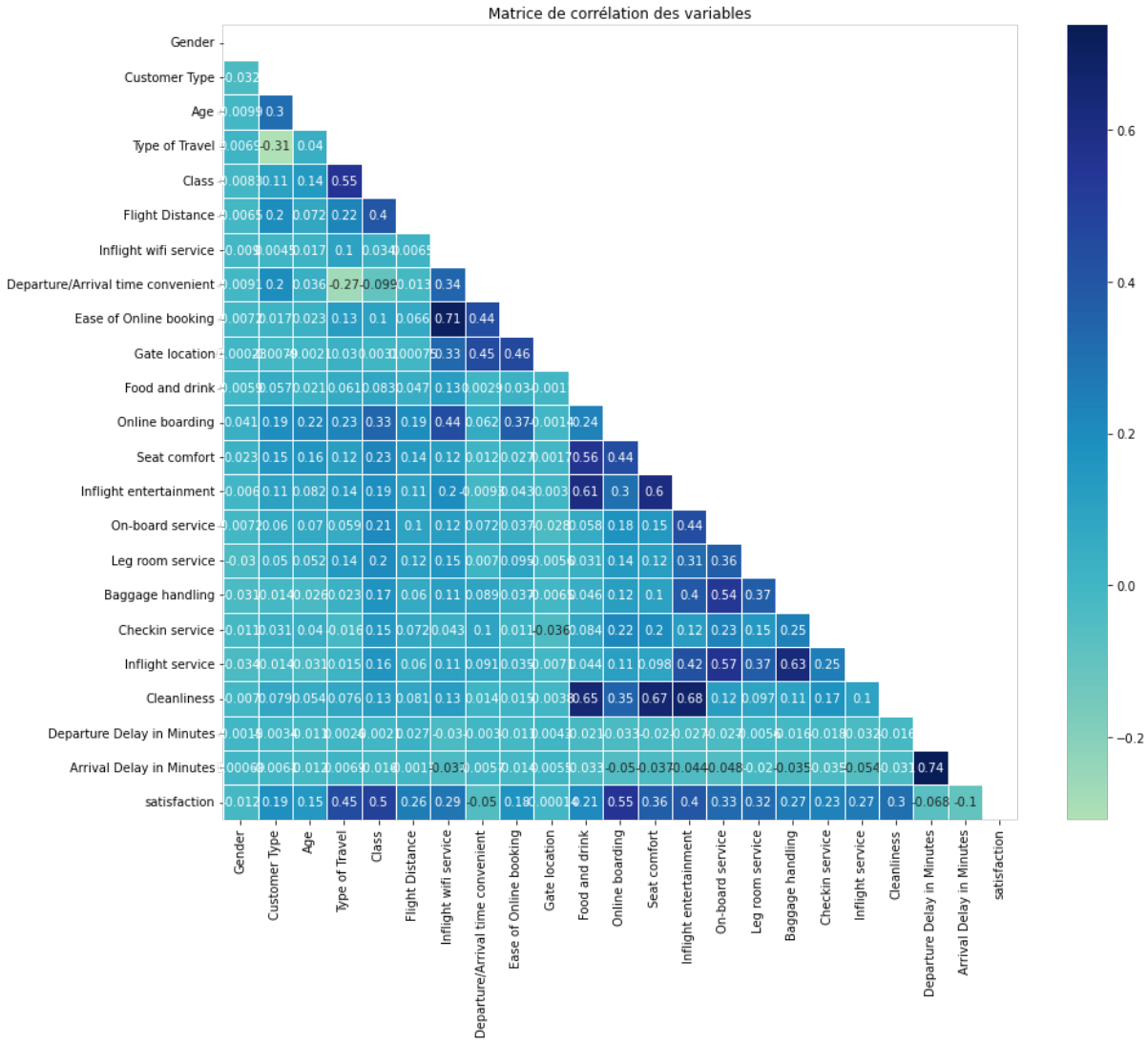


Figure 2 : Matrice de corrélation

PRÉ-PROCESSING DES DONNEES

Dans cette phase de prétraitement des données, nous avons mis en œuvre plusieurs étapes visant à préparer nos données pour l'entraînement d'un modèle de prédiction de satisfaction des passagers aériens. La fonction ***process_data()*** est utilisée pour appliquer ces transformations sur nos ensembles de données d'entraînement et de test.

Tout d'abord, nous avons supprimé les colonnes inutiles '***Unnamed**: 0*' et '***id***' qui ne contribuent pas à la prédiction de la satisfaction des passagers et pourraient introduire du bruit dans notre modèle.

Ensuite, nous avons converti les variables catégorielles en variables numériques à l'aide de techniques d'encodage. Par exemple, nous avons remplacé les valeurs des variables '***Gender***',

'***Customer Type***', '***Type of Travel***', '***Class***' et '***satisfaction***' par des valeurs numériques correspondantes. Cela permet à notre modèle de machine learning de traiter ces variables de manière appropriée.

Enfin, nous avons traité les valeurs manquantes dans la colonne ***'Arrival Delay in Minutes***' en les remplaçant par la médiane de cette colonne. Cela garantit que notre modèle ne soit pas affecté par les valeurs manquantes dans cette caractéristique cruciale.

CONSTRUCTION DU MODELE

Nous avons élaboré un réseau neuronal séquentiel pour notre tâche de prédiction de la satisfaction des passagers aériens. Le modèle comprend quatre couches Dense, dont les tailles respectives sont de 128, 64 et 32 neurones, activées par la fonction d'activation ReLU, favorisant ainsi la non-linéarité et la complexité du modèle.
Pour atténuer le risque de surapprentissage, des couches de dropout sont insérées entre chaque couche Dense, avec un taux de dropout de 0.5, désactivant aléatoirement la moitié des neurones pendant l'entraînement.

La dernière couche est constituée d'un seul neurone activé par la fonction d'activation sigmoïde, adaptée à la tâche de classification binaire de la satisfaction des passagers. Pour optimiser le modèle, nous avons compilé celui-ci en utilisant l'optimiseur **Adam** avec une learning rate de 0.001, une fonction de perte **binaire_crossentropy** et la métrique **accuracy** pour évaluer les performances du modèle.

- Adam : Un algorithme d'optimisation qui ajuste les poids du modèle pendant l'entraînement en utilisant des estimations adaptatives du taux d'apprentissage pour chaque paramètre.

- Binary Crossentropy : Une fonction de perte utilisée dans les problèmes de classification binaire pour mesurer la divergence entre les probabilités prédites et les vérités terrain.

- Métrique accuracy : Une mesure de performance qui calcule le pourcentage de prédictions correctes par rapport à l'ensemble des prédictions effectuées par le modèle.

Le modèle a été entraîné sur les données d'entraînement pendant 20 epochs, avec une taille de batch de 64. Les données de validation ont été utilisées pour surveiller les performances du modèle et ajuster les poids afin d'obtenir la meilleure généralisation possible.

EVALUATION DES PERFORMANCES

Nous avons évalué la performance du modèle sur les données de test en utilisant le modèle préalablement entraîné. Les données de test ont été prétraitées en utilisant le même **scaler** que pour les données d'entraînement. Nous avons ensuite évalué le modèle en termes de

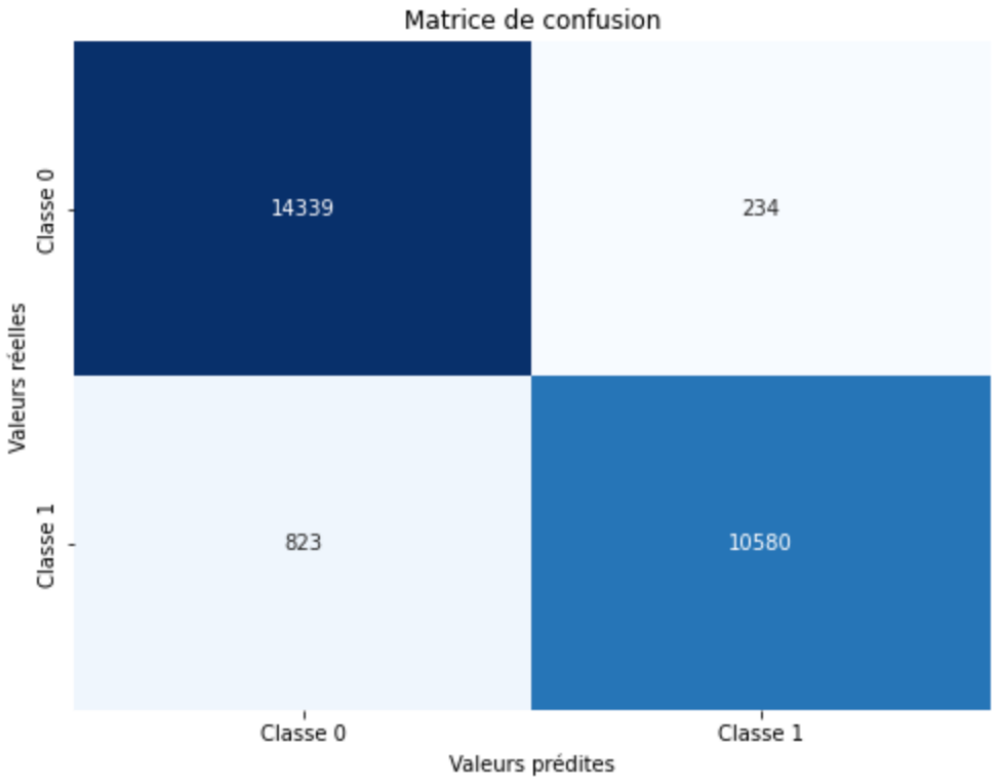
perte et de précision en utilisant la méthode ***evaluate()*** du modèle, qui a renvoyé un score de perte de 0.0934 et une précision de 0.9602.

Pour évaluer plus en détail les performances du modèle, nous avons effectué des prédictions sur les données de test en utilisant la méthode ***predict()***. Les prédictions ont été converties en classes binaires (0 ou 1) en utilisant un seuil de 0.5. Nous avons ensuite généré un rapport de classification pour examiner la précision, le rappel, le score F1 et le support pour chaque classe.

Le rapport de classification indique que le modèle a obtenu :

- **Une précision** de 0.95 pour la classe 0 (**insatisfait**) et de 0.97 pour la classe 1 (**satisfait**)
- **Un rappel** de 0.98 pour la classe 0 et de 0.94 pour la classe 1.
- **Un score F1**, qui représente l'équilibre entre précision et rappel, est de 0.96 pour la classe 0 et de 0.95 pour la classe 1.
- Enfin, **l'accuracy** globale du modèle sur les données de test est de 0.96, ce qui indique une bonne performance globale du modèle dans la prédiction de la satisfaction des passagers aériens.

Voici la matrice de confusion associée :



CONCLUSION

Ce rapport a présenté une approche méthodique pour prédire la satisfaction des passagers aériens en utilisant le deep learning. À travers **une analyse exploratoire**, **un prétraitement des données** et **la construction d'un modèle de réseau neuronal**, nous avons démontré une efficacité prometteuse dans la prédiction de la satisfaction des passagers. Cette méthodologie ouvre la voie à des améliorations futures pour optimiser davantage les performances du modèle.