Exercice de recrutement

Introduction :

Je tiens tout d'abord à vous remercier pour cette opportunité et pour cet exercice enrichissant par lequel je vous montrerai mes capacités et ma motivation.
Mon travail se divise sur 3 parties : partie prétraitement des données, partie visualisation des données et partie modélisation, toutes dans le même jupyter notebook “Exercice_recrutement.ipynb”.

1) Prétraitement des données :

Le but de cette étape est de rassembler tous les datasets en un seul qui est “ticket_data.csv”, puisqu’il contient des liens vers tous les autres datasets, pour que je puisse après produire des graphes et faire des visualisations.
Pour ce faire, j’ai utilisé l'opération “merge” de la bibliothèque Pandas de type droite (right merge) pour conserver toutes les lignes du DataFrame “tickets”, tandis que seules les lignes correspondantes du DataFrame de gauche sont conservées.

J'ai ensuite créé de nouvelles variables qui m’aideront par la suite dans l'extraction des informations intéressantes, la visualisation des données et la modélisation.
Les variables sont : “number_middle_stations” (nombre des stations intermédiaires), “number_other_companies” (nombre d’autres compagnies), “trip_duration_minutes” (durée des voyages en minutes), “distance” (distance des trajets) et “distance_range” (classes des durées des trajets).

Après, j’ai essayé d'éliminer les valeurs manquantes. J’ai commencé par la colonne “company_has_wifi” et il s'est avéré que toutes les valeurs manquantes sont de la compagnie “Vatry” donc je ne peux pas utiliser la valeur la plus fréquente, alors je les ai toutes remplies avec la valeur “False”.
C'était le même cas pour les colonnes “company_has_plug”, “company_has_adjustable_seats” et “company_has_bicycle”. Ensuite, pour les colonnes “o_city_population” et “d_city_population” j’ai utilisé une dataset externe (World Cities Database de "simplemaps interactive Maps & Data") pour remplir les populations manquantes, mieux que de remplir avec la valeur moyenne ou médiane, ce qui n'est pas logique.

2) Visualisation des données :

J'ai essayé de varier les informations et les graphiques, donc j’ai commencé par extraire les valeurs min, max et moyenne pour le prix et la durée de voyage :

Les valeurs min, max et moyenne du prix globalement :

	price_in_cents
count	74168.000000
mean	4382.711061
std	3739.325367
min	300.000000
25%	1900.000000
50%	3350.000000
75%	5250.000000
max	38550.000000

Les valeurs min, max et moyenne de la durée du voyage gloabelement :

	trip_duration_minutes
count	74168.000000
mean	424.620793
std	594.981356
min	20.000000
25%	180.000000
50%	290.000000
75%	480.000000
max	29571.000000

Après, j'ai extrait le prix de billets moyen pour chaque type de transport :

Le prix moyen pour chaque type de transport :

	price_in_cents
company_transport_type	
bus	3652.448036
carpooling	2742.171907
train	8506.634793

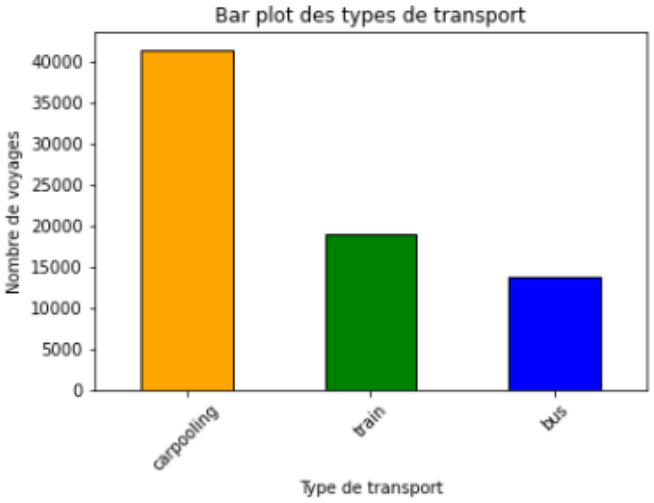
Les valeurs min, max et moyenne de la durée des voyages par trajets :

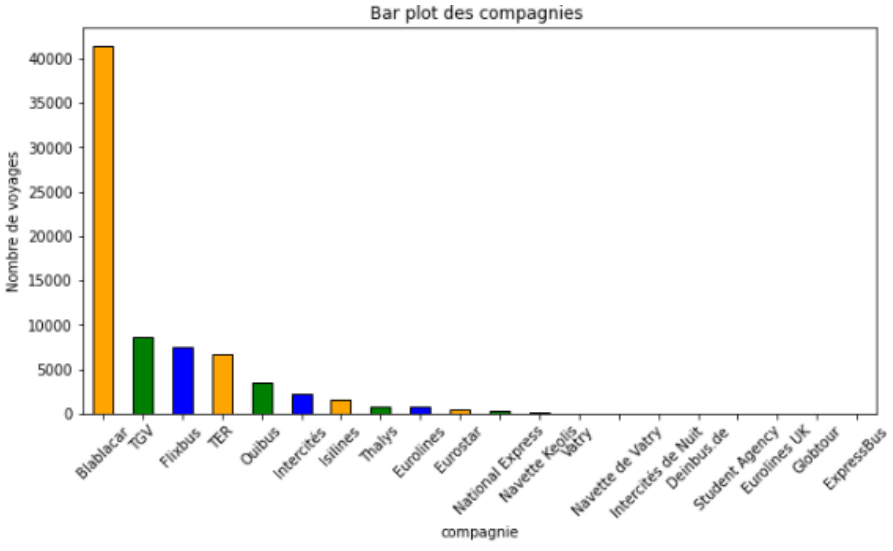
		trip_duration_minutes		
		min	max	mean
o_city_unique_name	d_city_unique_name			
aeroport-paris-vatry-	troyes	1315.0	1315.0	1315.000000
agde	amsterdam	533.0	954.0	618.800000
agen	dijon	744.0	901.0	822.500000
	marseille	336.0	740.0	497.400000
	marseille-aeroport	300.0	480.0	350.000000
...	...	...	...	...
villefranche-sur-cher	bordeaux	190.0	2264.0	700.500000
vitre	nice	593.0	665.0	629.000000
zurich	dijon	490.0	1065.0	777.500000
	liege	350.0	690.0	463.333333
	strasbourg	295.0	635.0	420.000000

Et le prix moyen et la durée moyenne pour chaque type de transport en fonction de la distance du trajet :

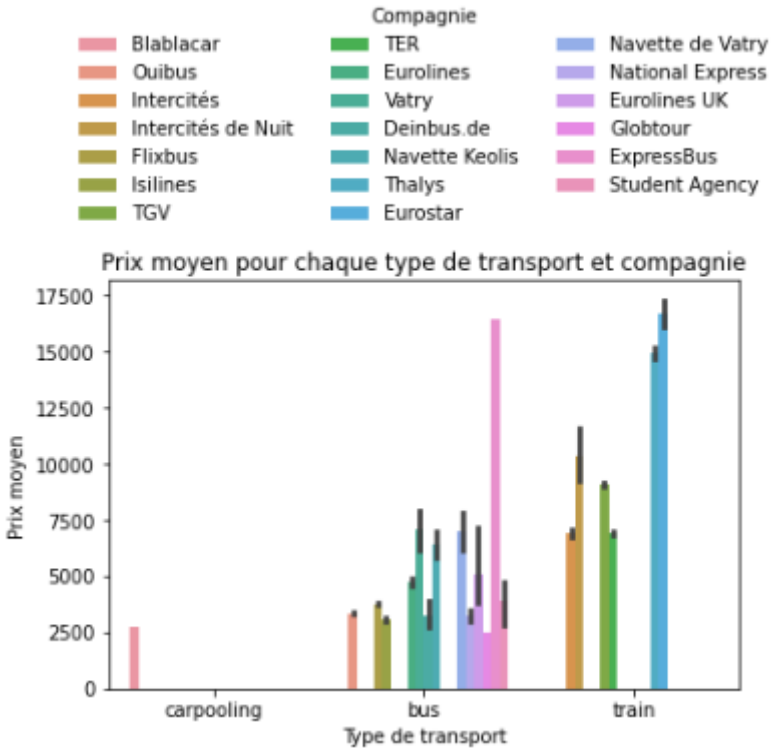
distance_range	company_transport_type	price_in_cents	trip_duration_minutes
short	bus	2642.317625	697.128181
	carpooling	1664.521491	157.204939
	train	6514.602410	397.639759
medium	bus	3435.708698	891.350105
	carpooling	2978.608093	265.533722
	train	8519.020582	435.399306
long	bus	6363.407181	1540.465241
	carpooling	6159.273183	589.649123
	train	14091.783784	707.203604
very long	bus	NaN	NaN
	carpooling	NaN	NaN
	train	NaN	NaN

Pour les graphes, j'ai commencé par afficher la différence entre le nombre des voyages par type de transport et par compagnie :

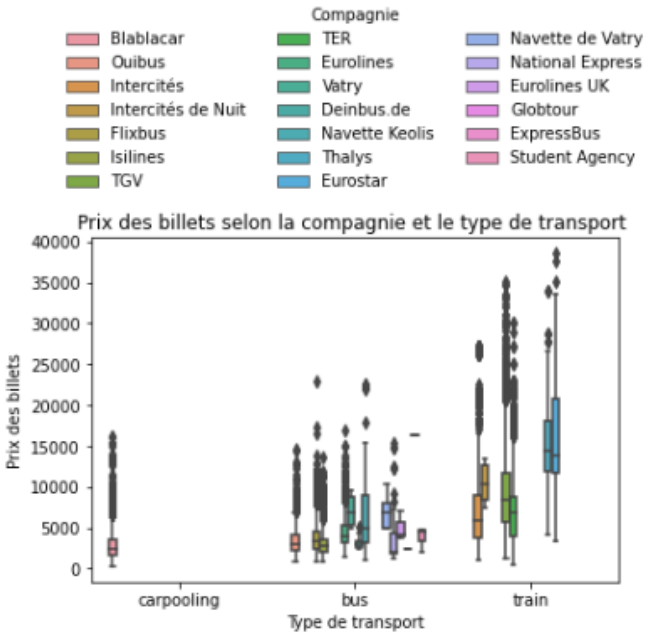




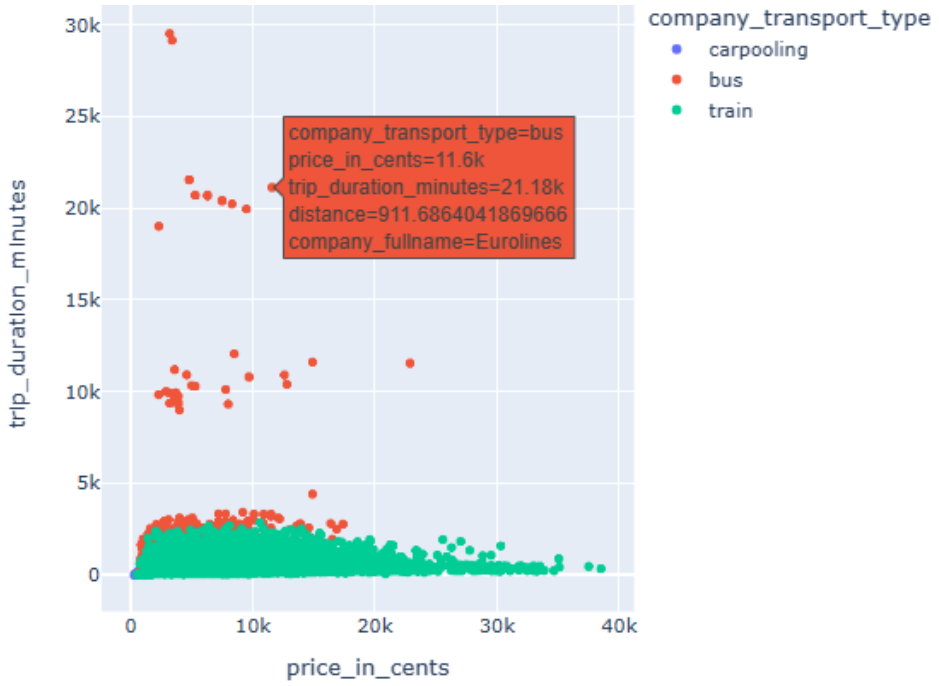
Ce graphe permet d'afficher le prix moyen pour chaque type de transport et sa propre compagnie :



Pour voir les valeurs extrêmes et comprendre la répartition des observations des prix des billets selon le type de transport, j'ai utilisé une boîte à moustache :



Pour une visualisation interactive, j'ai créé un nuage de points sur lesquels nous pouvons passer la souris pour avoir plus d'informations comme le nom de la compagnie :



3) Modélisation :

Pour la partie de modélisation, j'ai sélectionné les colonnes les plus importantes et j'ai transformé celles qui sont catégoriques en numérique. Pour ce faire, j'ai utilisé le Label Encoder car il est utile lorsqu'on a une relation ordinale comme pour le cas de la variable "distance_range" (short, medium, long, very long).

Toutefois, si les catégories n'ont pas de relation ordinale, le One Hot Encoding peut être un meilleur choix, mais, lorsque nous disposons de variables qui possèdent beaucoup de catégories comme pour le cas de la variable "company_fullname", ça peut conduire à une grande augmentation du nombre de caractéristiques.

J'ai procédé à une standardisation avec "Standard Scaler" pour mettre toutes les caractéristiques sur la même échelle .

J'ai testé différents modèles pour les comparer et choisir celui qui offre la meilleure performance en se basant sur l'erreur absolue moyenne (MAE) qui indique une mesure de la qualité de l'ajustement pour les modèles de régression linéaire et le R au carré (R Squared) qui est la moyenne de toutes les erreurs absolues.

J'ai également effectué une recherche de grille (Grid Search) pour la forêt aléatoire afin d'obtenir les meilleurs hyperparamètres.

Comparaison des modèles :

Régression linéaire :

MAE = 1449.91
R au carré = 0.58

Arbre de décision :

MAE = 496.21
R au carré = 0.91

Forêt aléatoire :

MAE = 458.74
R au carré = 0.93

Forêt aléatoire (recherche de grille) :

MAE = 455.18
R au carré = 0.93

Régression linéaire :

MAE = 1449.90
R au carré = 0.58

Boosting de gradient :

MAE = 954.60
R au carré = 0.77

C'est la forêt aléatoire avec la recherche de grille qui a obtenu les meilleurs résultats avec une erreur moyenne de 4.5518 euro pour le prix du billet.

Conclusion :

En conclusion, ce rapport décrit les étapes que j'ai suivies pour travailler sur les différentes données et les résultats que j'ai obtenus, que ce soit au niveau de l'extraction des informations intéressantes ou au niveau des performances des modèles.

Ce fut un exercice enrichissant qui m'a permis d'élargir mes connaissances et m'a donné encore plus envie de travailler au sein de votre startup.

C'est certain que ce travail doit être amélioré et je serais ravi de rejoindre Tictactrip en tant que stagiaire Data Scientist pour pouvoir travailler sur des tâches similaires.