Machine learning use cases

TD1 – feature engineering

Résumé :

Le client nous demande, à partir de données couvrant 6 semaines, de créer une application utilisant l'intelligence artificielle pour prédire le délai d'expédition des commandes des clients. Pour que ce projet soit réalisable, il me semble nécessaire que les données soient représentatives du reste de l'année.

Après avoir exploré le dataset, cela ne semble pas être le cas. En effet, nous observons de fortes augmentations du nombre de commande à traiter lors de la semaine du Black Friday ce qui signifie que des éléments périodiques extérieurs peuvent fortement impacter les délais d'expédition des providers. Pour pouvoir déclarer dans les données tous ces éléments (Noël, périodes de soldes, vacances, jours fériés, publicités…) il faudrait que le client nous les transmette ou avoir les données des expéditions sur au moins un an.

Un autre point qui peut être dérangeant pour l'IA, est qu'il semble y avoir des providers qui ne sont pas continuellement actifs. En effet, en regardant les données, certains providers ont commencé à effectuer des expéditions à partir du 19 novembre (peut-être pour couvrir la période du Black Friday). Cela laisse supposer que d'autres providers qui n'apparaissent pas dans les données sont actifs à d'autres moments de l'année auquel cas le modèle d'IA ne pourrait pas prédire leurs délais d'expédition s'ils n'apparaissent pas dans le set d'entraînement.

Après avoir testé plusieurs feature preprocessing, splits train / test et modèles d'intelligence artificielle sur les données, le meilleur résultat que nous obtenons est de 66% de précision (régression arrondie à l'entier) dans des conditions qui ne sont probablement pas représentatives de la réalité (split train test aléatoire au lieu de chronologique). Je ne pense donc pas qu'il soit pertinent de continuer le projet et de l'amener à l'état de production. Il faudrait des données couvrant une plus grande période et des éclaircissements sur les providers en activité de la part du client pour pouvoir envisager de continuer.

Explication détaillée :

Dataset fourni :

- 6 semaines d'historique des commandes
- 572 841 commandes
- Il est composé de 3 variables :
    - La date et l'heure de la commande
    - L'identifiant du « provider » qui correspond au service de livraison
    - La date d'expédition de la commande

Extrait du dataset :

	datedecreationdecommande	providerservice_id	dateexpe
0	1/11/19 11:40	48	2/11/19 0:00
1	1/11/19 11:40	48	2/11/19 0:00
2	1/11/19 10:24	48	2/11/19 0:00
3	1/11/19 14:24	48	2/11/19 0:00
4	1/11/19 13:28	48	2/11/19 0:00

En condition de production, l'entreprise enverra une mise à jour quotidienne (la nuit) du jeu de données avec les nouvelles commandes et expéditions.

Objectif

L'objectif est de pouvoir prédire en temps réel le nombre de jours écoulés entre la date de commande et d'expédition.

Mise en forme du dataset :

Pour traiter les dates sous formes de valeurs numériques nous extrayons plusieurs variables :

- Le jour de la semaine
- L'heure de la journée
- La minute de l'heure

	provider	date_expe	date_commande	heure_commande	min_commande	gap	jour_expe	jour_commande
0	48	2019-11-02	2019-11-01	11	40	1	5	4
1	48	2019-11-02	2019-11-01	11	40	1	5	4
2	48	2019-11-02	2019-11-01	10	24	1	5	4
3	48	2019-11-02	2019-11-01	14	24	1	5	4
4	48	2019-11-02	2019-11-01	13	28	1	5	4

La variable « gap » correspond au nombre de jours d'écart entre la date de commande et la date d'expédition.

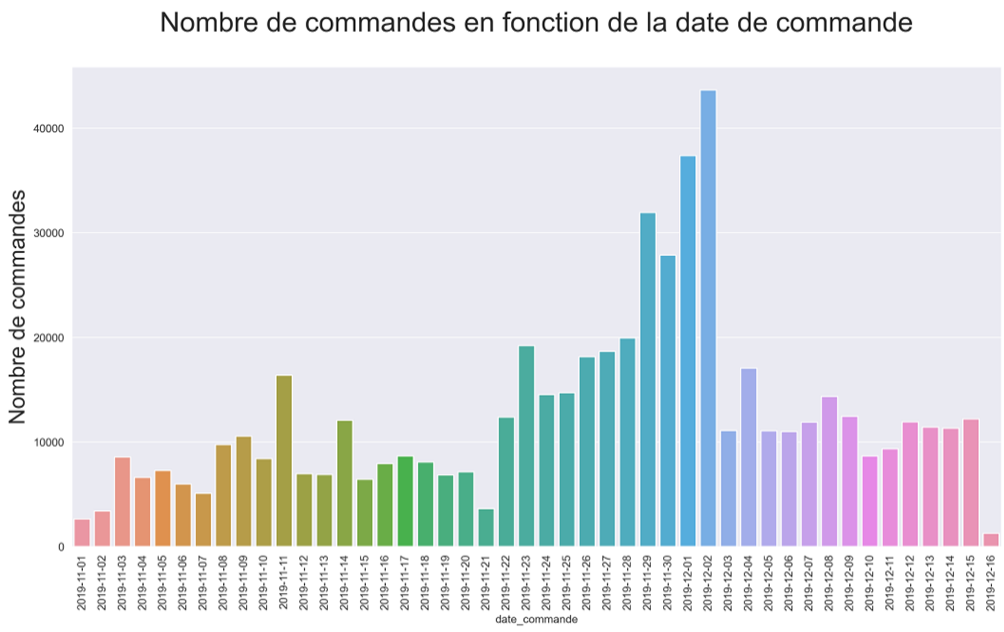
Après mise en forme du dataset , nos obtenons 34 commandes pour lesquelles le nombre de jours entre la date de commande et d'expédition est négatif. Ces résultats sont probablement dus à des erreurs , nous décidons donc de les retirer.

Remarque :

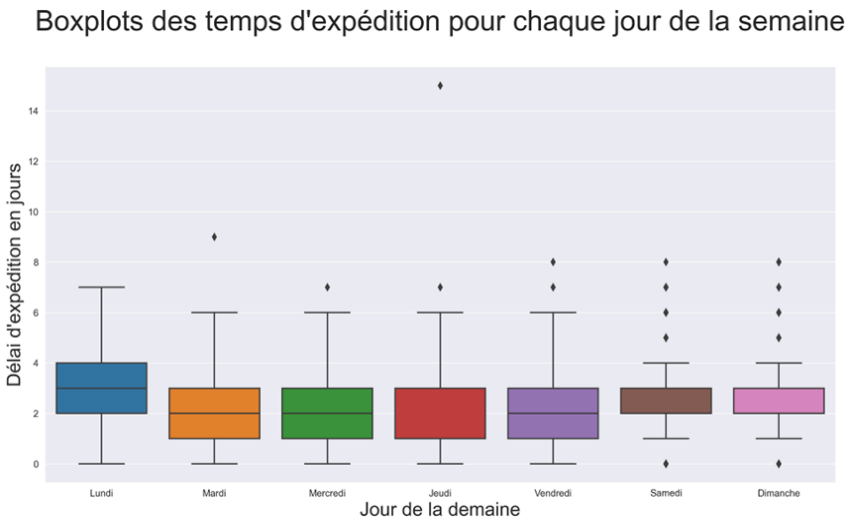
Nous pourrions également extraire le numéro du mois ou de la semaine de l'année. Cependant nous voulons que le modèle d'IA soit capable de prédire toute l'année et nous ne disposons que de 6 semaines de données. En condition de production (à moins de réentraîner le modèle régulièrement), l'algorithme recevrait donc des données qu'il n'a jamais rencontré ce qui pourrait fausser les résultats.

Exploration du dataset :

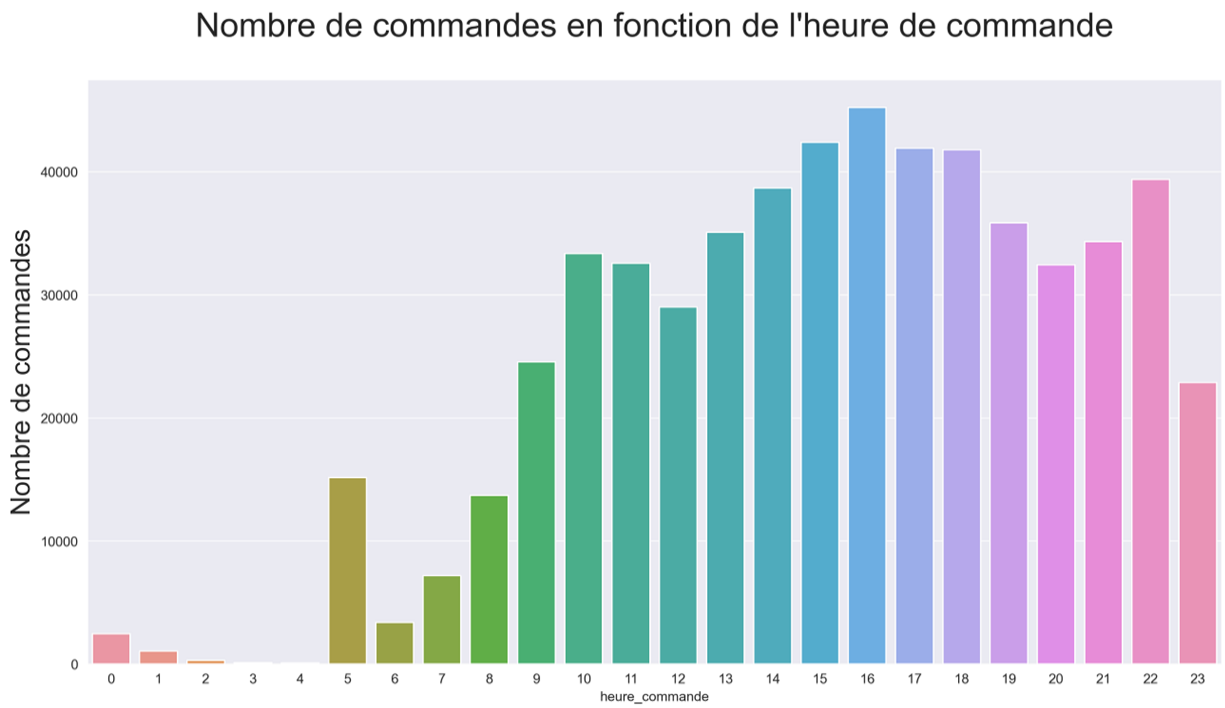
Observation des dates et horaires de commande :



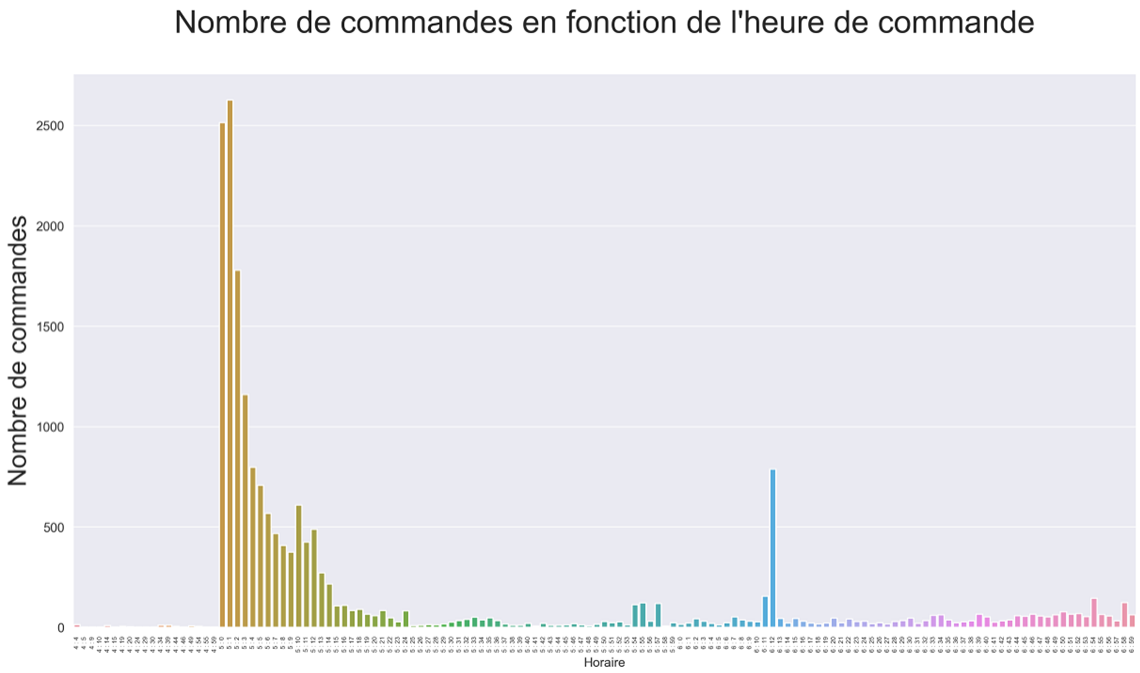
- Nous observons qu'il y a une forte montée des commandes pour la période du black Friday.
- Les valeurs faibles pour les 2 premiers jours et le dernier jour viennent probablement de la manière dont les données ont été récupérées par l'entreprise.



Les délais de livraison semblent généralement plus long pour les commandes effectuées les lundis, samedis et dimanches.

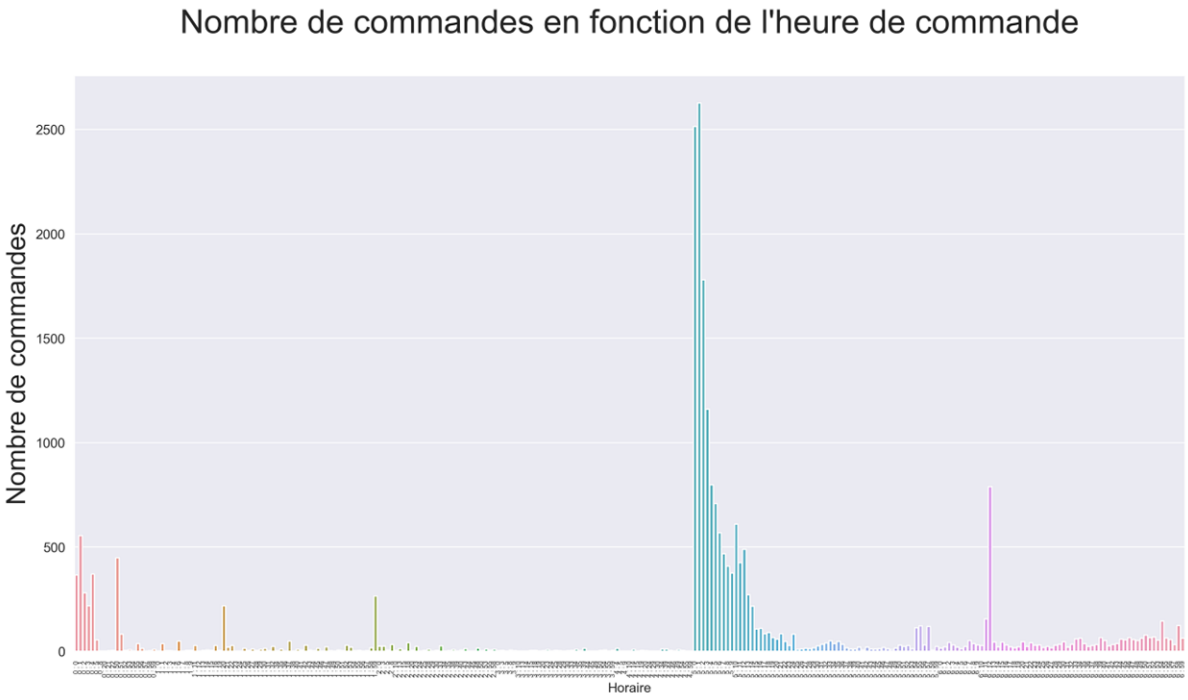


- Nous observons un pic de commandes à 5h du matin ce qui est très contre intuitif.

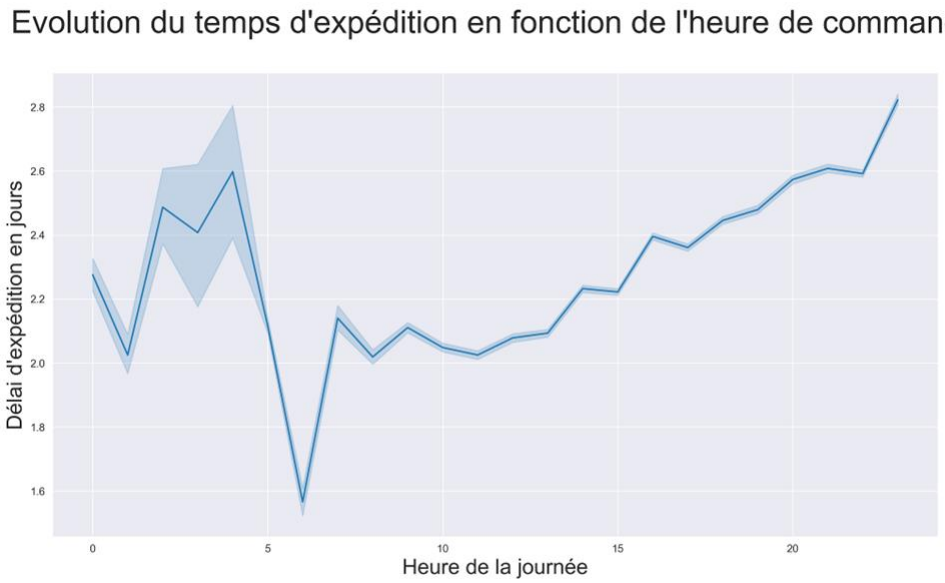


- Pics de commande dans les premières 20 minutes de la 5ème heure et à 6h12
- Difficile de savoir s'il s'agit d'un procédé informatique de l'entreprise (report des commandes effectuées pendant la nuit à 5h du matin) ou d'un comportement client (décalage horaire…).

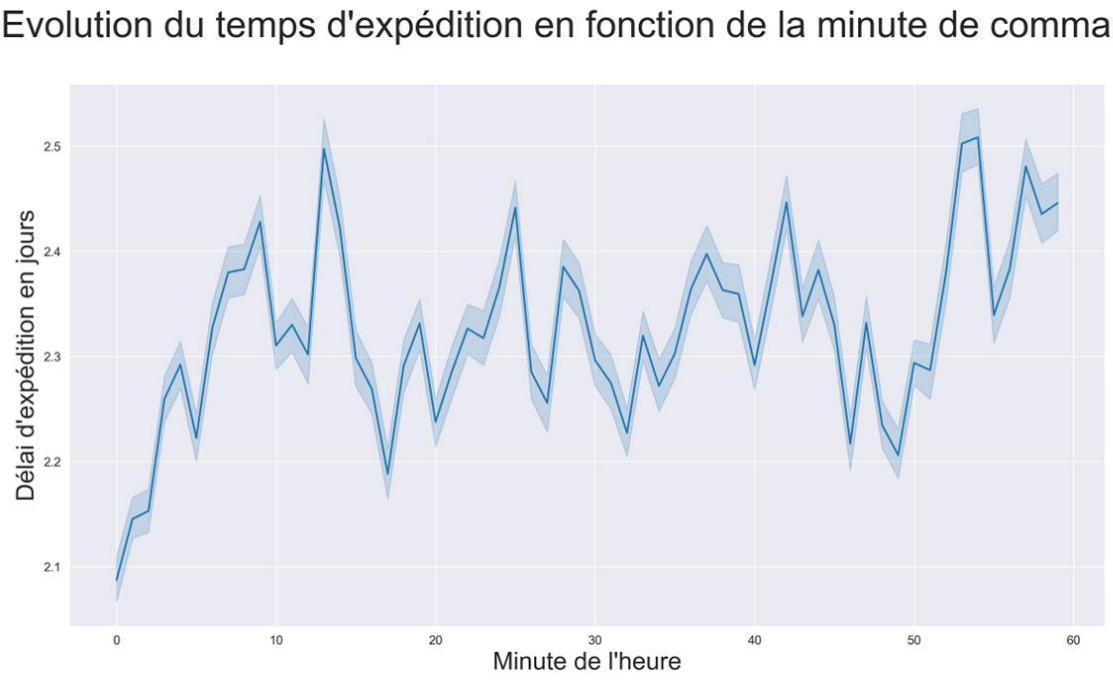
Si l'on agrandit l'intervalle affiché , la première option semble plus plausible :



Nous souhaitons qu'une fois en production , l'IA soit capable d'estimer le délai d'expédition en temps réel donc un report des commandes nocturnes à 5h du matin pourrait fausser les prédictions des clients commandants la nuit.

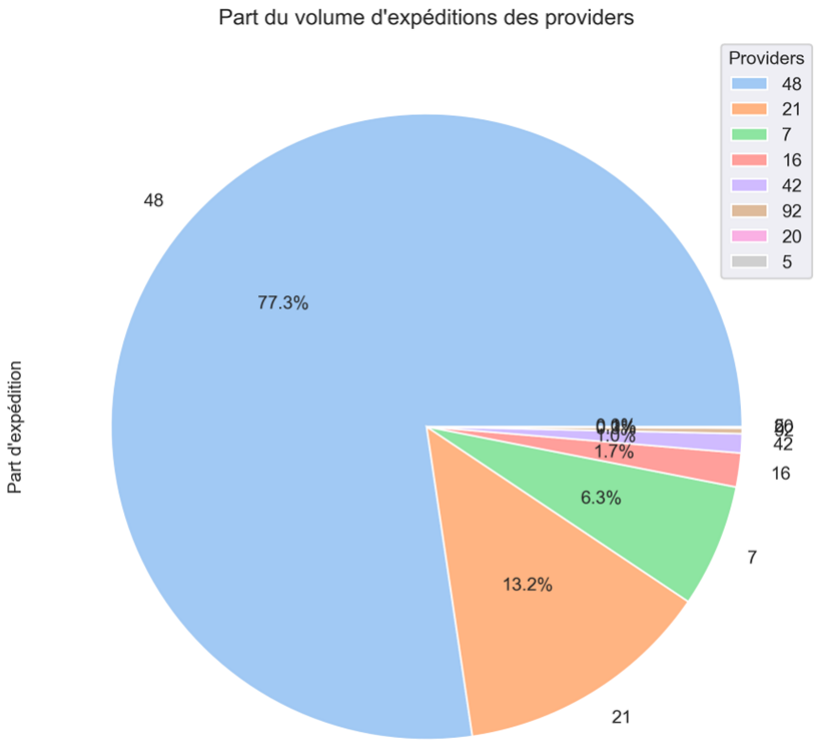


Le délai d'expédition chute fortement pour les commandes effectuées pendant la 6ème heure de la journée puis augmente progressivement entre 7h et 23. Ce délai semble instable entre 1h et 5h, cela peut être dû au peu de commandes effectuées pendant ces horaires.



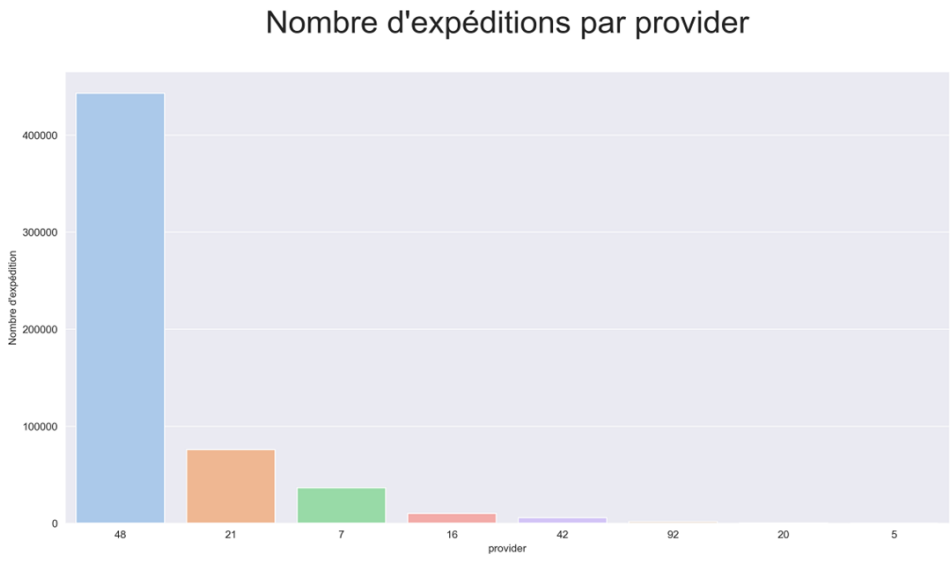
Les résultats semblent assez aléatoires, nous pouvons tout de même remarquer qu'en moyenne les commandes faites dans la première minute sont expédiées plus rapidement.

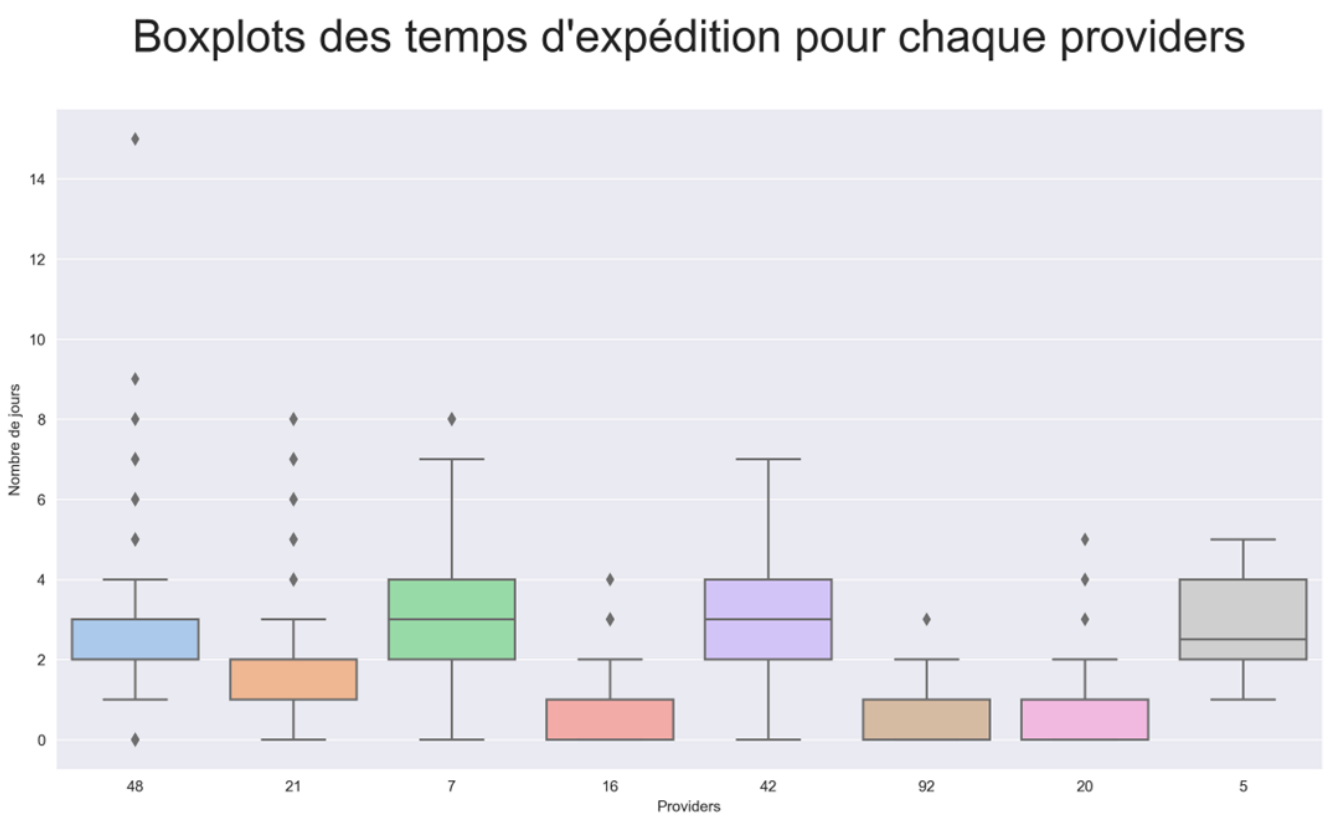
Observation des providers :



Les providers 48 et 21 représentent plus de 90% du nombre de commandes expédiées. Les providers 5, 20 et 92 ont un volume extrêmement faible.

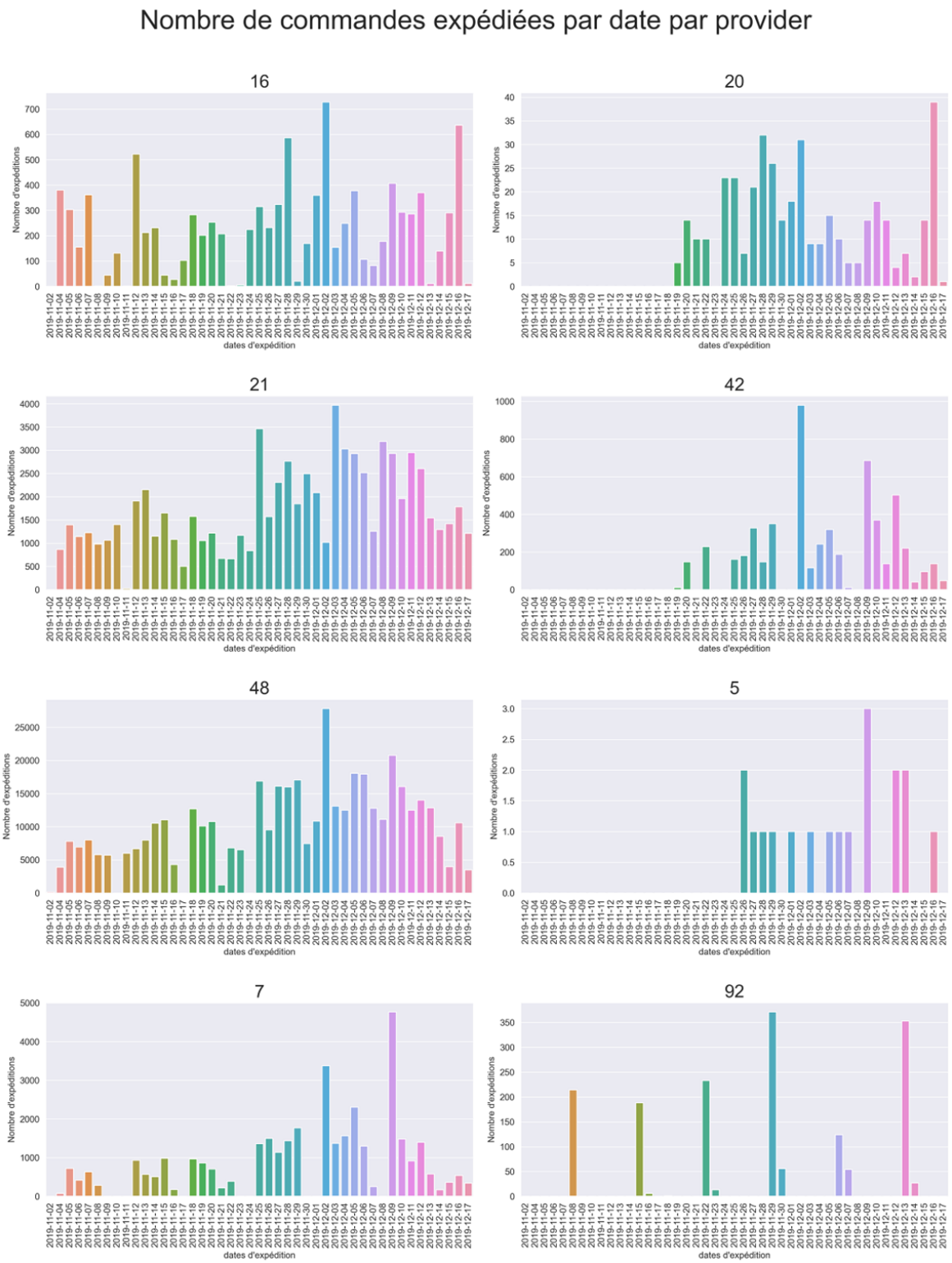
Voici ce que cela représente en nombre de commandes :





Nous observons des délais d’expédition similaires pour les providers 7 et 42 qui sont relativement longs ainsi que pour les providers 16, 92 et 20 qui sont au contraire assez rapides.

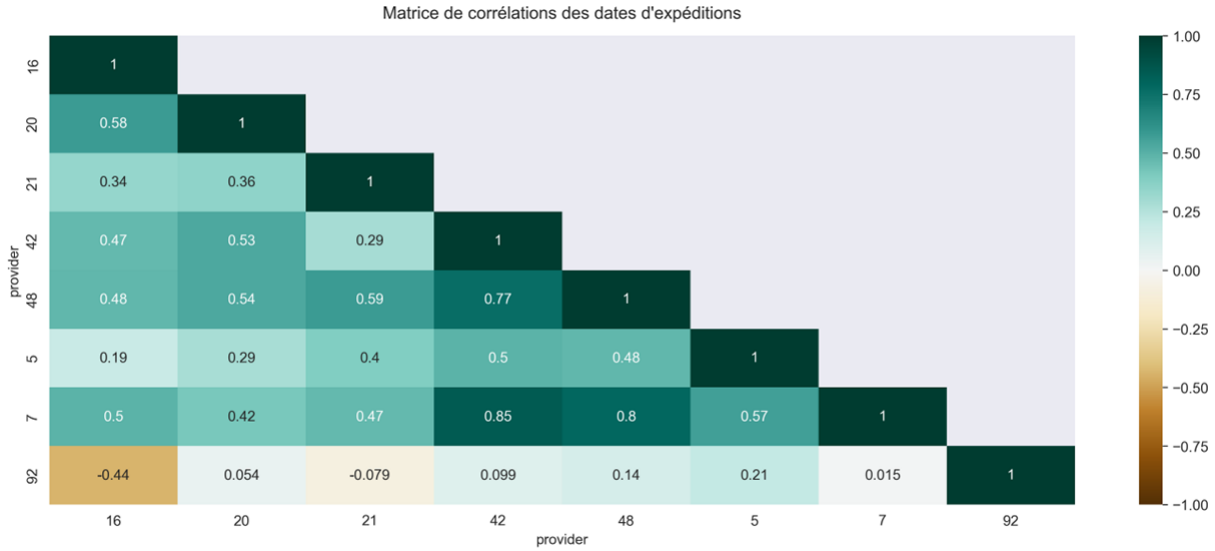
Nous pouvons à présent nous intéresser au nombre de commandes expédiées par jour par chaque providers :



Nous pouvons faire plusieurs observations à partir de ces graphiques :

- Aucune expédition n'apparaît les 2 et 3 novembre 2019 dans la base de données.
    - Cela provient sûrement de la manière dont été rassemblées les données par l'entreprise.
- Le provider 48 qui représente le plus gros volume d'expéditions :
    - N'a pas fait d'expéditions les dimanches 3,10,17 et 24 novembre.
    - Mais en a effectué les dimanches 1er , 8 et 15 décembre.
        - Est-ce qu'il s'agit d'un comportement exceptionnel pour couvrir la période du black Friday et de Noël ?
    - En dehors des dimanches il semble actif pendant toute la période couverte par le dataset.
- Le provider 21 n'a pas effectué d'expéditions le 11 novembre 2019
    - Peut-être car il s'agit d'un jour férié.
- Le provider 7 ne semble pas expédier de commandes les week-ends sauf pour le mois de décembre où il y a quelques expéditions enregistrées les samedis 7 et 14 ainsi que le dimanche 15.
    - Peut-être qu'il y a un changement de comportement du provider en décembre pour couvrir la période de Noël
    - Pas d'expéditions le 11 novembre.
- Le provider 16 semble faire moins voire aucune expédition les vendredis et samedis
- Le provider 42 n'enregistre pas d'expéditions pendant les 16 premières journées du dataset ainsi que les samedis et dimanches.
    - S'il s'agit d'un provider exceptionnel pour couvrir la période du black Friday et de Noël cela pourrait être problématique pour généraliser les prédictions du modèle d'IA en dehors de la période couverte par les données.
- Le provider 92 n'effectue des expéditions que les vendredis et samedis.
- De la même manière que pour le 42 , le provider 16 ne compte aucune expédition pour les 16 premières journées du dataset.
    - Il peut donc également être problématique pour généraliser le modèle d'IA.
- Le provider 5 a un très faible volume de livraison et ses expéditions commencent le 26 novembre.

Nous pouvons également regarder la matrice des corrélations entre les providers. Cette matrice affiche les coefficients de corrélation de Pearson des volumes de commandes expédiés chaque jour pour chaque providers.



Les corrélations positives s'expliquent sûrement par le fait que pendant la période du black Friday le volume de commande à expédier à augmenter en simultané pour tous les providers.

Nous remarquons cependant qu'il y a une corrélations négatives non négligeable entre les providers 16 et 92. Cela est dû au fait que leurs journées d'activités sont opposées. En effet le providers 92 n'expédie que les vendredis et samedis et le 16 expédie tous les jours sauf les vendredis et samedis.

Résumé des interprétations que nous pouvons effectuer sur les providers :

- Les providers 48 et 21 représentent une grande majorité du volume d'expédition.
- La plupart des providers ont des dates pour lesquelles ils n'expédient pas (week-ends, 11 novembre, vendredis…).
- Certains providers ont commencé à effectuer des expéditions uniquement pendant la période du black Friday ce qui peut sous-entendre qu'ils ne sont pas nécessairement actifs le reste de l'année.
- Les providers 16 et 92 se complètent au niveau des dates d'expédition. Ils ont également des délais d'expédition similaires (délais courts). Nous pouvons donc supposé qu'ils correspondent aux livraisons rapides pour lesquelles l'entreprise souhaite assurer un service constant.
- Le provider 5 semble effectuer uniquement de rare commande à l'unité avec un fort délai d'expédition. Il s'agit peut-être d'un service dédié au colis encombrants ou exceptionnels.

Je pense qu'en situation réelle, sans informations et données supplémentaires de la part du client, ce serait une perte de temps d'entraîner un modèle car il y aurait peu de chance qu'il puisse généraliser sur le reste de l'année et la période couverte par le dataset est trop courte pour pouvoir le tester convenablement.

Machine learning :

Par curiosité voici quelques tentatives de machine learning effectuées :

1/Prédiction classique :

Feature preprocessing :

Avant de passer à la prédiction nous calculons des variables pour apporter des informations supplémentaires à l'IA. Ces variables sont :

- Le nombre de commandes en attente pour chaque provider à J-1 (lorsque l'on souhaite effectuer les données les plus récentes dont on dispose sont celles de la veille).
    - Ainsi que le nombre de commandes en attente total.
- Le nombre de commandes envoyées par la provider lors des X derniers jours (X est le délai moyen d'expédition du provider)
    - Ainsi que le nombre de commandes envoyées total lors des X derniers jours.

stock_total	nb_expe_provider	nb_expe_total	48_delay	16_delay	21_delay	7_delay	92_delay	42_delay	20_delay	5_delay	48_nb_expe	16_nb_expe	21_nb_expe	7_nb
18234	285.0	8336	15578	133	1376	1147	0.0	0.0	0.0	0.0	11478	131	2466	
31639	25651.0	33257	24447	410	3354	2669	2.0	734.0	22.0	1.0	42545	323	3878	
24102	15344.0	6195	20673	136	1590	1702	0.0	0.0	0.0	0.0	15344	102	1584	
58781	24527.0	31647	47877	406	3817	5335	0.0	1325.0	19.0	2.0	40507	169	4339	
10626	8014.0	10647	8434	46	1320	658	0.0	151.0	4.0	0.0	18810	2	1333	

Modèles :

Nous utilisons des modèles de régression pour prédire le délai d'expédition puis nous arrondissons ce délai à l'entier pour pouvoir calculer la précision.

Nous définissons 3 scores de précision différents :

- Accuracy 1 :
    - Le pourcentage de bonnes prédictions une fois les valeurs de prédictions arrondies.
- Accuracy 2 :
    - Le pourcentage de bonnes prédictions en regroupant tous les délais de 5 jours ou plus dans une même classe.
- Accuracy 3 :

- Le pourcentage de bons encadrements, lorsque l'on arrondi encadre la prédiction entre l'entier inférieur et supérieur les plus proches (tout en gardant la classe regroupant les délais de plus de 5 jours).
- Les classes sont donc : « Entre 0 et 1 jour », « Entre 1 et 2 jours »… « 5 jours ou plus »

a/Modèle de référence :

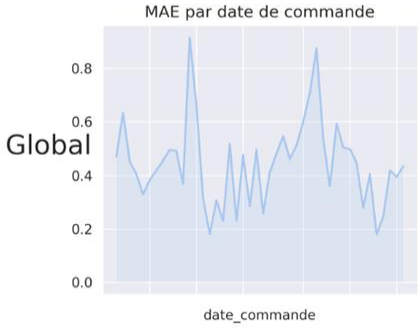
On lance un boosting de gradient avec les hyperparamètres par défaut pour avoir un modèle de référence.

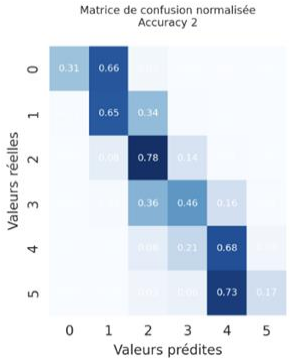
- On utilise un set de test de 25% avec un split train-test aléatoire.
- Pour limiter le nombre de variables que l'on donne au modèle nous n'effectuons pour l'instant pas de « one hot encoding » sur les variables catégoriques (provider, jour_semaine, heure_commande, minute_commande)
- La fonction de scoring utilisée est l'erreur moyenne absolue car c'est celle avec laquelle j'ai eu les meilleurs résultats.

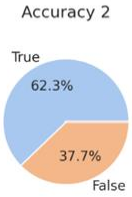
Résultats obtenus :

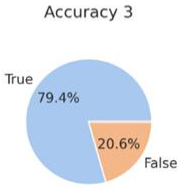
```
Erreur moyenne absolue du modèle : 0.4778 jours
Accuracy 1 : 0.6173
Accuracy 2 : 0.6225
Accuracy 3 : 0.7941
```

a









Le modèle de référence se trompe en moyenne de 0.478 jour et 61.7% des prédictions sont correctes. En utilisant les classes de l'accuracy 3 nous obtenons 79% de bonnes réponses.

2/Modèles avec auto-sklearn :

Pour obtenir un meilleur ensemble de modèle de machine learning nous utilisons la librairie auto-sklearn (auto-ml).

La librairie va tester par elle-même plusieurs modèles (parmi ceux disponibles dans scikit-learn), plusieurs features preprocessing et hyper paramètres pendant un laps de temps précisé afin de retourner le meilleur ensemble qu'elle a réussi construire.

En renseignant dans le dataset les variables catégoriques, la librairie va tester par elle-même de les encoder.

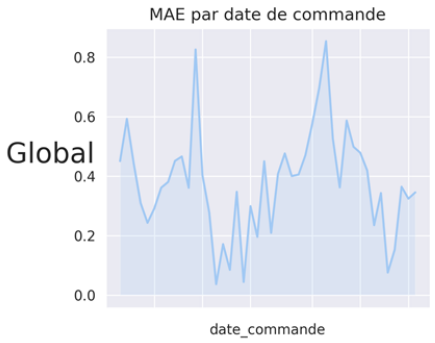
Après une heure d'exécution voici les résultats obtenus :

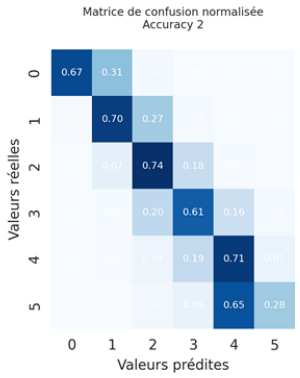
model_id	rank	ensemble_weight	type	cost	duration	train_loss	data_preprocessors	feature_preprocessors
35	1	1.0	gradient_boosting	0.426275	191.562846	0.420373	[]	[no_preprocessing]
20	2	0.0	gradient_boosting	0.430616	232.605614	0.426443	[]	[feature_agglomeration]
30	3	0.0	gradient_boosting	0.454860	113.527705	0.454173	[]	[select_rates_regression]
45	4	0.0	decision_tree	0.496486	39.494354	0.496133	[]	[select_percentile_regression]
37	5	0.0	gradient_boosting	0.496505	191.794844	0.496165	[]	[feature_agglomeration]
6	6	0.0	sgd	0.605582	59.204566	0.605409	[]	[select_rates_regression]
40	7	0.0	decision_tree	0.655639	12.667222	0.655618	[]	[select_percentile_regression]
43	8	0.0	gradient_boosting	0.698761	219.587639	0.694502	[]	[pca]
28	9	0.0	gradient_boosting	0.909802	18.292460	0.909760	[]	[select_rates_regression]
23	10	0.0	gradient_boosting	0.952707	93.545448	0.952684	[]	[polynomial]

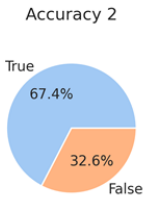
Tableau récapitulant les meilleurs modèles obtenus par auto-sklearn

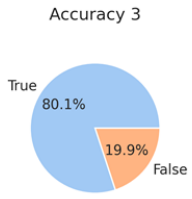
Les modèles les plus performants semblent être « gradient_boosting », «decision_tree » et « sgd ».

```
Erreur moyenne absolue du modèle : 0.4228 jours
Accuracy 1 : 0.6678
Accuracy 2 : 0.6737
Accuracy 3 : 0.8007
```





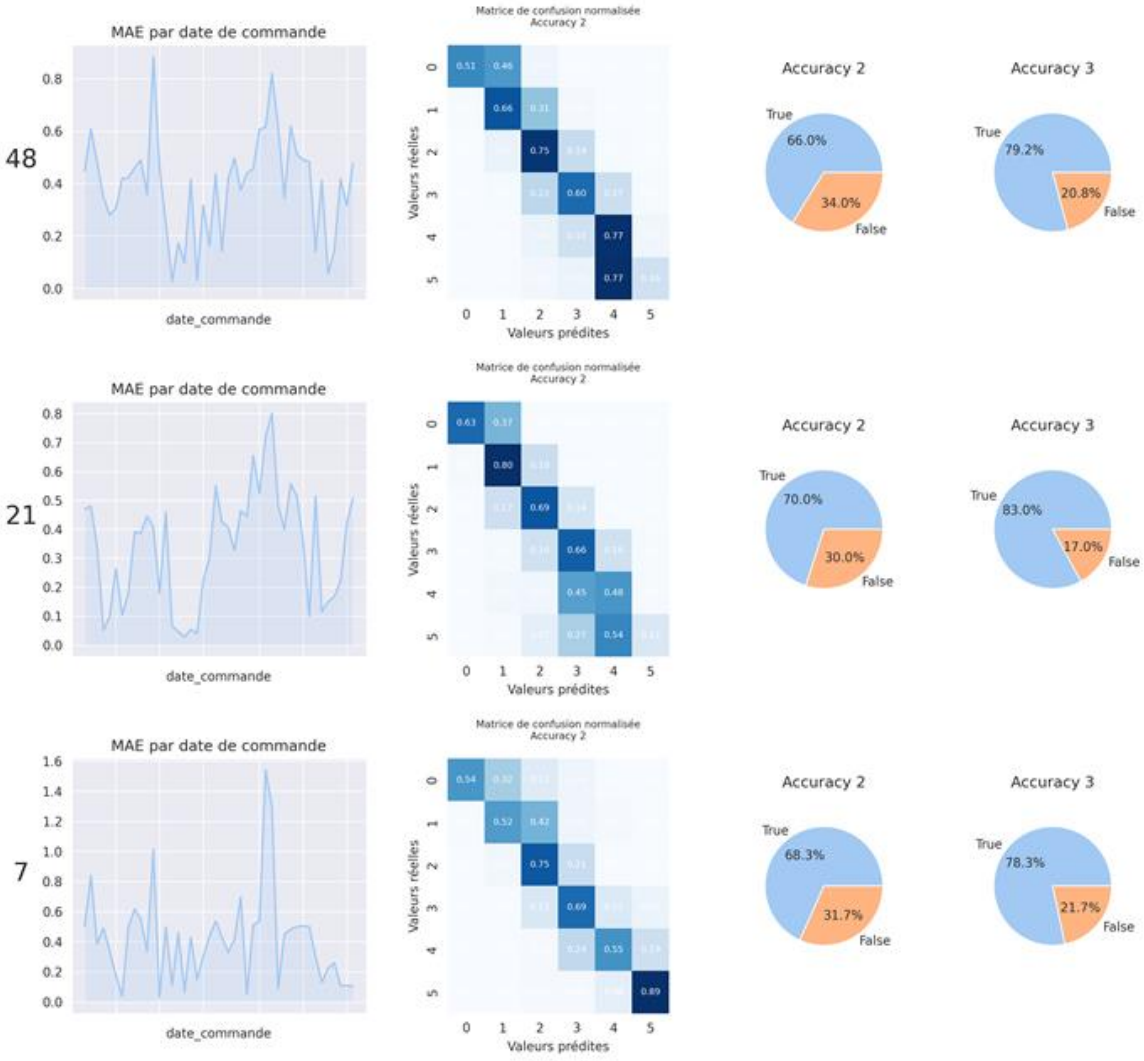




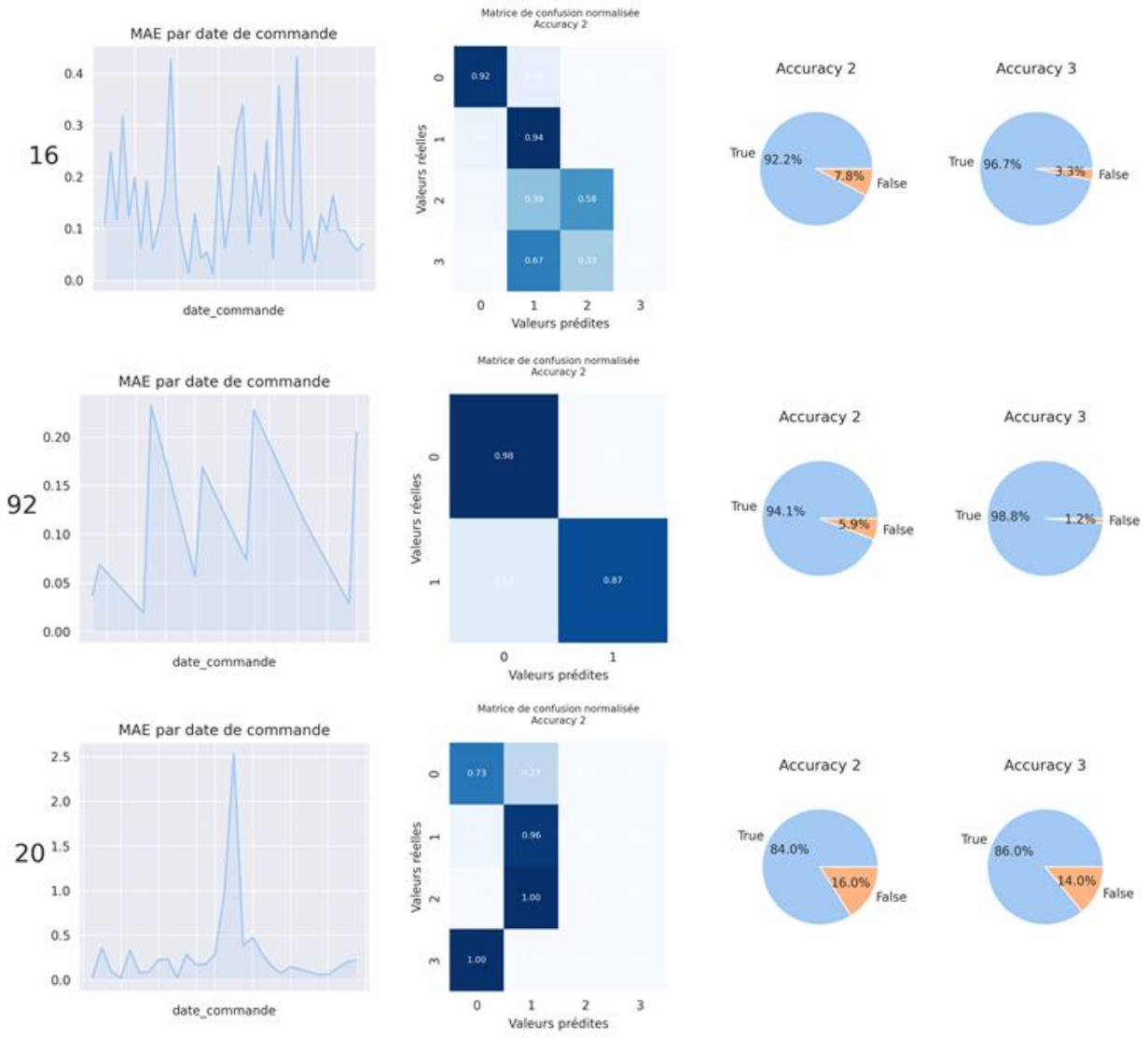
Nous observons une amélioration d'environ 5% par rapport au modèle de référence pour l'accuracy 2.

L'accuracy 3 reste très similaire entre les 2 modèles.

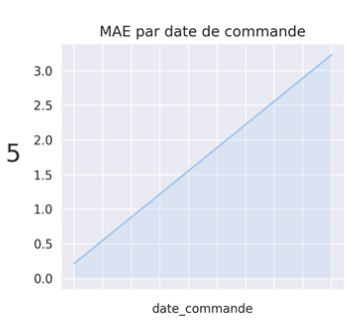
Nous pouvons nous intéresser aux résultats des prédictions pour chaque providers :

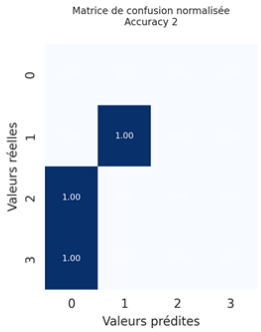


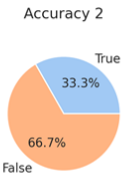
Les trois providers principaux , sont prédits avec des précisions similaires. Le modèle semble avoir des difficultés à prédire les délais d'expéditions supérieurs à 4 jours pour les providers 48 et 21.

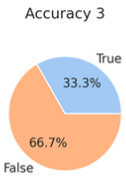


Les providers 16, 92 et 20 que l'on avait qualifiés de livraison express sont mieux prédits que les autres, sûrement en partie grâce à leur régularité et le fait qu'ils n'ont pas expédié de commandes après plus de 3 jours de délai.



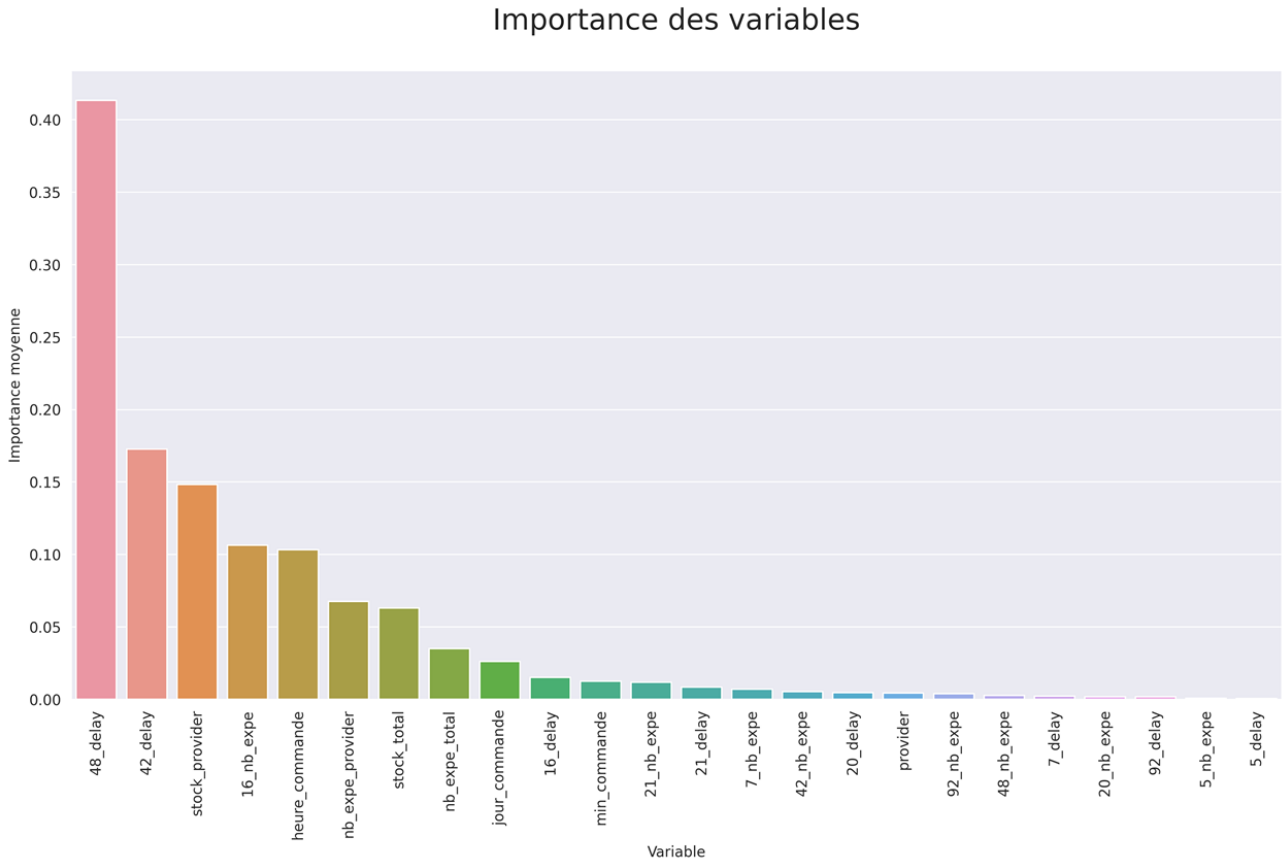






Le modèle n'arrive pas du tout à prédire les délais du provider 5, cela vient sûrement du fait qu'il a très peu de données dans le dataset et que son comportement est différent des autres.

En utilisant la fonction "permutation_importance » de sklearn pour estimer l'importance des variables du modèle, voici le classement que nous obtenons :



Embedding des providers :

Nous avons observé dans la partie exploration que le nombre de providers actifs était variable. Les données dont nous disposons ne couvrent que 6 semaines , il y a donc de bonnes chances que de nouveaux providers entrent en activité pendant le reste de l'année.

Dans cette situation le modèle serait incapable de prédire le comportement d'un provider sur lequel il ne s'est pas entraîné.

C'est pour cela que nous allons essayer de réaliser un embedding des providers. L'objectif de la démarche est d'obtenir de bonnes prédictions de la part du modèle en retirant la colonne contenant les ids des providers et en la remplaçant par d'autres variables qui serviraient à définir les providers.

Ces variables permettant de caractériser les providers sont calculées pour chaque journées à partir des données des jours précédents. Elles sont donc sensées devenir de plus en plus précises avec le temps.

Lorsqu'un nouveau provider entre en activité, les variables seront initialisées avec une valeur par défaut puis, après un certain nombre de jours d'activité du provider elles seront calculées chaque jours automatiquement.

Lors des premiers jours l'IA ne devrait donc pas être précise mais on espère qu'avec le temps et plus de données les précisions s'amélioreront.

Je ne connais pas de méthode pour déterminer les variables à générer, je les choisis donc arbitrairement.

Variables :

- Stock du provider concerné
- Nombre de jours d'activité du provider sur les 15 derniers jours
- Nombre de commandes expédiées par le provider sur les X derniers jours (X est le délai moyen d'expédition du provider)
- Les quartiles du délai d'expédition du provider (min,Q1,med,Q3,max) sur les 15 deniers jours
- Le nombre moyen de commandes expédiées par jour de la semaine sur les 15 derniers jours.
- Un indicateur du type d'expédition du provider
    - 4 classes :
        - Expédition express
        - Expédition rapide
        - Expédition classique
        - Expédition lente
    - Chacune de ces classes correspond à des délais d'expédition différents
    - Il y a une colonne par classe
    - Cette variable doit être saisie à la main lorsqu'un provider rentre en activité afin de donner une indication à l'IA du délai d'expédition. On enregistre alors un 1 dans la classe sélectionné.

- Après 3 jours d'activité du provider on remplace les valeurs de chaque type d'expédition par la part de commandes du provider correspondant à chaque classe.
    - Ex :[0,0,1,0] => [0.05,0.3,0.55,0.1]

Clustering des temps d'expédition pour réaliser les classes :

Nous effectuons un clustering à une dimension sur les délais d'expéditions pour générer 4 classes :

Résultats du clustering :

	nb_commandes	temps_expe_moyen	variance_temps_expe	Type expedition
2	155286	0.906791	0.290726	express
0	201413	2.000000	0.000000	rapide
3	123320	3.000000	0.000000	normale
1	92788	4.461051	0.727220	lente

Les classes sont donc :

- 0 ou 1 jour
- 2 jours
- 3 jours
- 4 jours ou plus

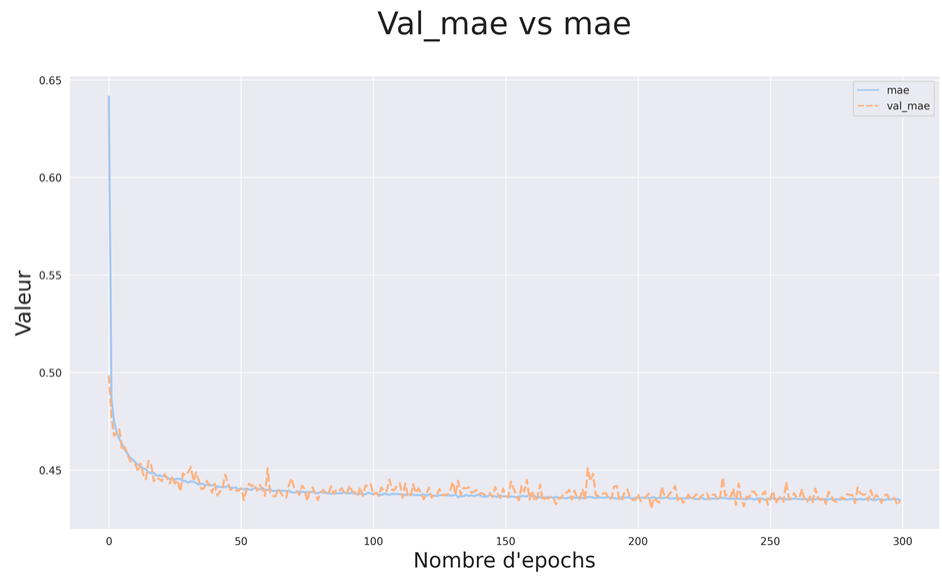
Comme la plupart des variables sont calculées sur les 15 derniers jours on peut également espérer que le modèle puisse s'adapter par lui-même (avec quelques jours de retard) aux fluctuations d'expédition etc.

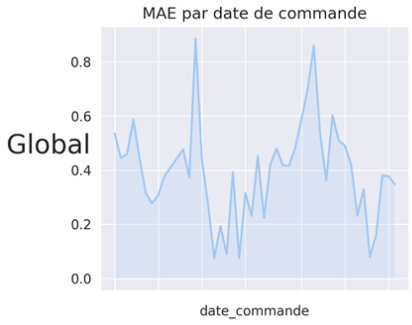
Modèle :

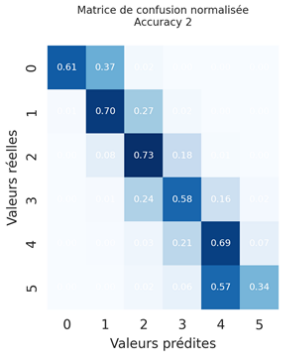
Comme nous avons beaucoup de variables nous utilisons un réseau de neurones en modèle.

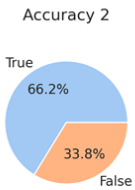
Nous l'entraînons d'abord avec split train test classique.

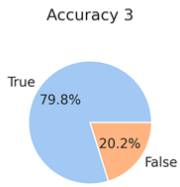
Résultats :











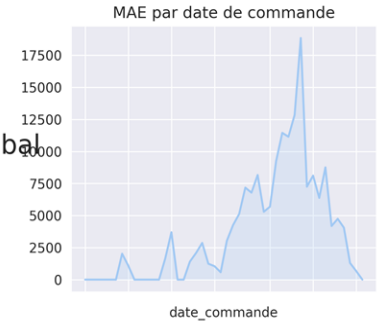

```
Erreur moyenne absolue du modèle : 0.4365 jours
Accuracy 1 : 0.6565
Accuracy 2 : 0.6625
Accuracy 3 : 0.7983
```

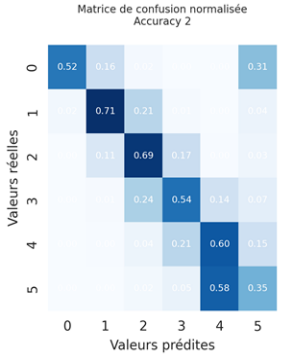
Le modèle n'overfit pas et les résultats obtenus sont similaires à ceux des modèles précédents alors que l'on a enlevé la colonne provider.

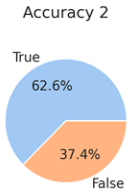
Pour tester que le modèle est capable de prédire les délais d'expédition pour des providers qu'il n'a pas encore rencontré, on l'entraîne cette fois en isolant les providers 42 et 92 dans le set de test.

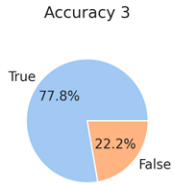
```
Erreur moyenne absolue du modèle : 4984.8001 jours
Accuracy 1 : 0.6107
Accuracy 2 : 0.6256
Accuracy 3 : 0.7778
```

Global



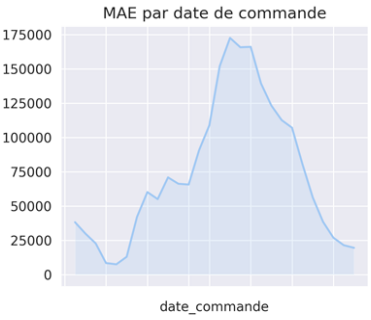


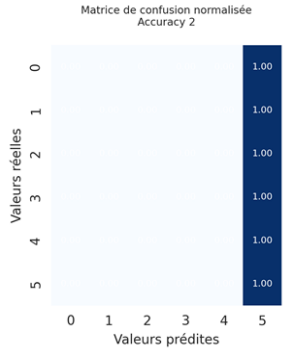


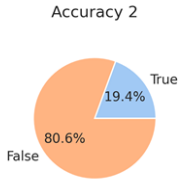


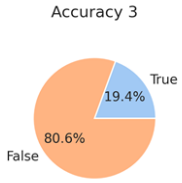
L’accuracy n’a pas beaucoup changée mais la MAE a été multipliée par 10 000.

42

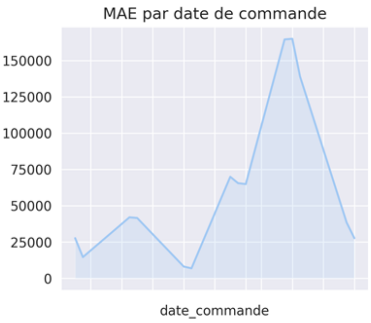


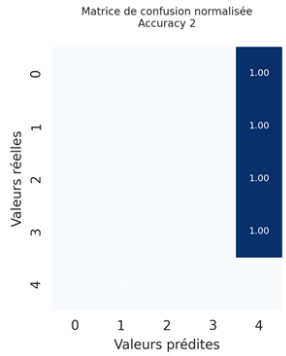


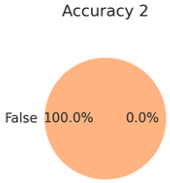


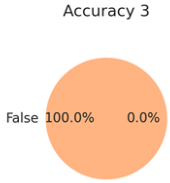


92









Le modèle n’arrive pas du tout à prédire les providers avec lesquels il ne sait pas entraîné c’est donc pour moi une raison supplémentaire de refuser cette mission.