SIMPLON

MARCO CASION

Résumé

Le résultat obtenu dans le modèle issu du projet MARCO CASION, amène à un R2 de 0.83%, résultat obtenu grâce à l'algorithme RandomForestRegressor.
Cet algorithme fait partie d'une famille d'algorithmes récents et plus ou moins complexes qui parviennent à prédire des résultats cohérents avec un très bon R2 dans une grande majeure partie des situations.
Il devient alors intéressant d'arriver à améliorer ce score…

Thomas Cassagne

E2 – Amélioration d'un modèle – Marco Casion

Table des matières

1 Référentiel des compétences abordées 2

2 Préambule 3

2.1 Résultats obtenus concernant le projet « MARCO CASION » 3

3 Nettoyage approfondi des données 4

4 Feature Engineering 5

4.1 Création de la feature « Brands » 5

4.2 Création de la feature « Seniority » 5

4.3 Création des features « Continent » et « Country » 5

5 Analyse finale – Visualisation 6

6 Corrélation des données 8

7 Normalisation des données 9

8 Création d'un Modèle : V2 9

8.1 Résultats Modèle V1 9

8.2 Résultats Modèle V2 10

9 Conclusion 10

9.1 Tableau récapitulatif des scores 10

9.2 Comparaison des modèles 11

10 Annexes 12

10.1 Annexe 1 : Qu'est-ce qu'un DataFrame ? 12

10.2 Annexe 2 : Normalisation des données 12

10.3 Annexe 3 : MinMaxScaler 12

10.4 Annexe 4 : OneHotEncoder 12

10.5 Annexe 5 : Hyperparamètres 13

10.6 Annexe 6 : Score (MAE, RMSE, R2) 13

1 Référentiel des compétences abordées

Dans le cadre du projet « MARCO CASION » les améliorations envisagées sont relativement nombreuses, c'est ce qui m'a amené à choisir ce sujet pour le travail que je dois produire concernant l'amélioration d'un modèle.

C8. Modifier les paramètres et composants de l'intelligence artificielle afin d'ajuster aux objectifs du projet les capacités fonctionnelles de l'algorithme à l'aide de techniques d'optimisation.

C14. Améliorer l'application d'intelligence artificielle en développant une évolution fonctionnelle pour répondre à un besoin exprimé par un client ou un utilisateur.

2 Préambule

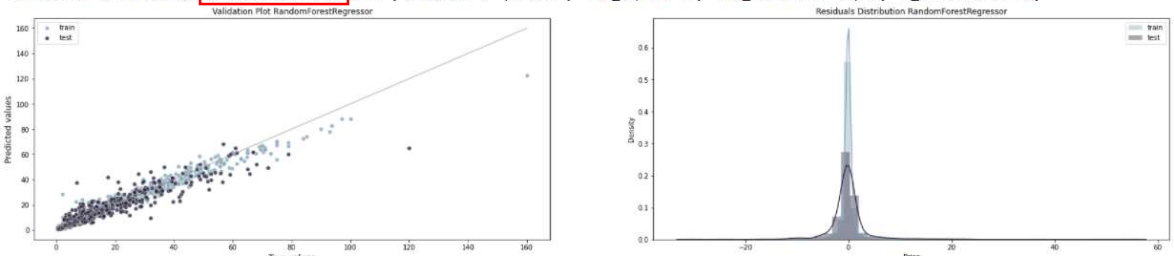
Le résultat obtenu dans le modèle issu du projet MARCO CASION (modèle prédictif de la valeur d'achat d'un véhicule d'occasion sur le marché Indien), amène à un R2 (Voir Annexe 6) de 0.83%, résultat obtenu grâce à l'algorithme **RandomForestRegressor.**

Cet algorithme fait partie d'une famille d'algorithmes récents et plus ou moins complexes (exemple : Xgboost…) qui parviennent à prédire des résultats cohérents avec un très bon R2 dans une grande majeure partie des situations. Il devient alors intéressant d'arriver à **améliorer ce score**. Il faut pour cela bien nettoyer nos données étudier les règles "métier" liées à celle-ci, procéder à du Feature Engineering ... En bref, analyser, comprendre et décortiquer au maximum nos données. 😉

```
Score du jeu TRAIN
MAE:  0.6186984909058257
RMSE:  1.5054327317568839
Median abs err:  0.2732499999999969
R2:  0.9820694671649806

Score du jeu TEST
MAE:  1.6259364978448279
RMSE:  3.69731636443902
Median abs err:  0.7038499999999985
R2:  0.8947465579597457

Score après GridSearchCV
CV Mean:  0.8673514064356065
STD:  0.03281903043838012
Le meilleur score est de: 0.8385841350703653 avec {'criterion': 'poisson', 'max_depth': 80, 'max_features': 'sqrt', 'n_estimators': 50}
```

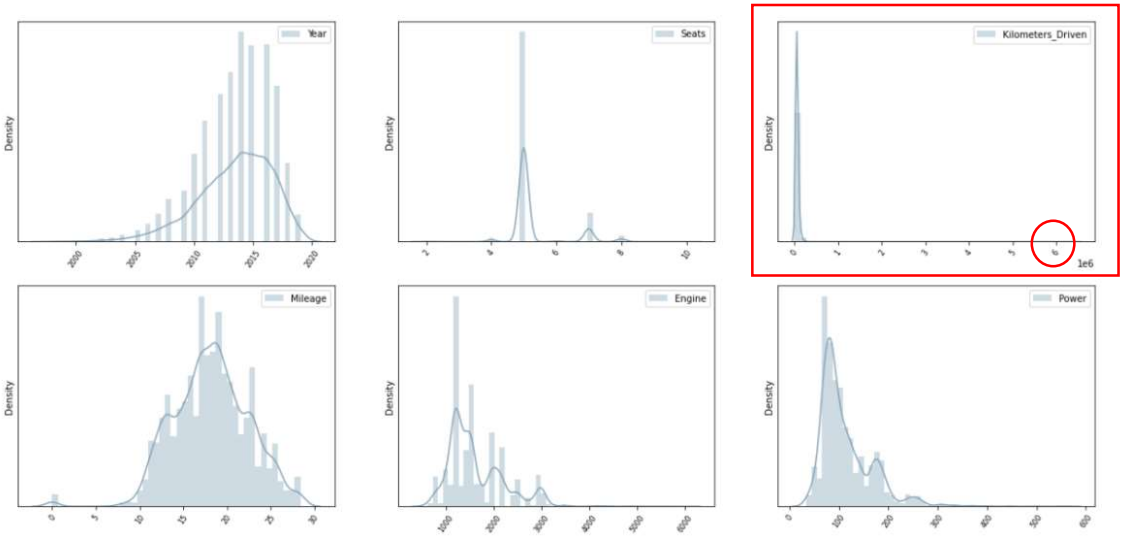


2.1 Résultats obtenus concernant le projet « MARCO CASION »

Résultats de l'algorithme **RandomForestRegressor** au sein du projet « Marco Casion » :

Lors de la première phase du projet « Marco Casion » une visualisation de la distribution des features avait permis d'observer une valeur outlier au sein de la feature Kilometers_Driven. En effet une voiture possédait plus de 6 000 000 de kilomètres.

Schéma N°1 : distribution par critère des données



Cette voiture avait donc été écartée du jeu de donnée fournie par la société. Mais ce n'est pas suffisant !

3 Nettoyage approfondi des données

Afin d'améliorer les distributions des différentes séries du jeu de données, un **Z score** sera appliqué sur celle-ci.

Source :

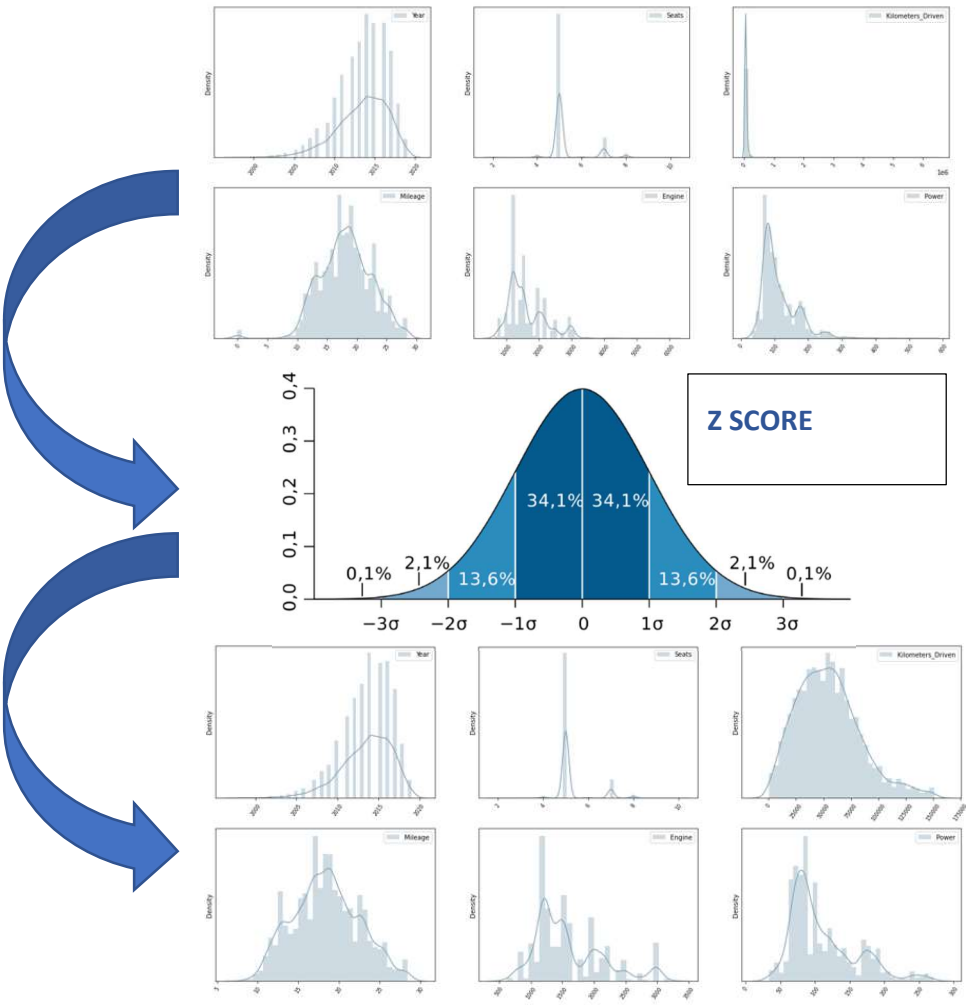
- https://help.tableau.com/current/pro/desktop/fr-fr/calculating_z_scores.htm

En statistiques, le Z score (ou score standard) d'une observation désigne le nombre d'écarts-types qui se trouve au-dessus ou en dessous de la moyenne de la population. Pour calculer un résultat z, vous devez connaître la moyenne de population et l'écart-type de population. Créer une visualisation de score z pour répondre aux questions du type suivant :

- Quel pourcentage de valeurs est-il inférieur à une valeur spécifique ?
- Quelles valeurs peuvent être considérées comme exceptionnelles ?

Le ZSCORE va me permettre d'exclure les valeurs non représentatives au sein des certaines distributions de mon jeu de donnée (Power, Kilometers_Driven, Engine, Mileage).

Schéma N° 2 : évolution des distributions par critère après traitement des données



Grâce au Z score j'ai écarté les valeurs les moins représentées au sein des diverses distributions du jeu de données. Après avoir effectué ce traitement, le jeu de données contient à présent 5603 lignes. Au total 204 lignes « non représentatives » du jeu de données ont été écartées.

4 Feature Engineering

Source :

- https://datascientest.com/feature-engineering

Il est maintenant essentiel de passer par une phase de **Feature Engineering**.

« Le Feature Engineering est un processus qui consiste à transformer les données brutes en caractéristiques représentant plus précisément le problème sous-jacent au modèle prédictif. Pour faire simple, il s'agit d'appliquer une connaissance du domaine pour extraire des représentations analytiques à partir des données brutes et de les préparer pour le Machine Learning.

Il s'agit de la première étape dans le développement d'un modèle de Machine Learning prédictif. Ceci permet d'accroître l'exactitude du modèle sur les nouvelles données inconnues. »

4.1 Création de la feature « Brands »

La feature « Name » du jeu de donnée contient deux éléments distincts :

- Marque de la voiture (exemple Audi)
- Modèle de la voiture (exemple : A4 New 2.0 TDI Multitronic)

	Name	Year	Owner_Type	Seats	Kilometers_Driven	Fuel_Type	Transmission	Mileage	Engine	Power	Price
0	Hyundai Creta 1.6 CRDi SX Option	2015	First	5	41000.0	Diesel	Manual	19.67	1582.0	126.20	12.50
1	Honda Jazz V	2011	First	5	46000.0	Petrol	Manual	18.20	1199.0	88.70	4.50
2	Suzuki Ertiga VDI	2012	First	7	87000.0	Diesel	Manual	20.77	1248.0	88.76	6.00
3	Audi A4 New 2.0 TDI Multitronic	2013	Second	5	40670.0	Diesel	Automatic	15.20	1968.0	140.80	17.74
4	Nissan Micra Diesel XV	2013	First	5	86999.0	Diesel	Manual	23.08	1461.0	63.10	3.50
...	...	...	...	...	...	...	...	...	...	...	...

Je décide alors de splitter ces deux éléments et de créer une nouvelle feature contenant uniquement la marque de la voiture.

Cette nouvelle colonne portera à présent le nom « **Brands** ». Il sera alors possible de compléter les analyses, par exemple par :

- Le rapport de prix entre les différentes marques
- Les marques de voitures les plus vendues en Inde

4.2 Création de la feature « Seniority »

Afin de faciliter l'apprentissage de mon modèle, je décide de simplifier la colonne "**Year**" (allant de 1998 à 2019) en soustrayant l'année actuelle (2022) par l'année de fabrication de la voiture. La feature "Year" devient alors "**Seniority**".

4.3 Création des features « Continent » et « Country »

De plus, après avoir effectué des recherches concernant les continents et pays d'origine des différentes marques de voitures j'ai décidé d'intégrer 2 nouvelles features au DataFrame (Voir annexe1) :

- **Continent** : Continent du siège social de la marque de la voiture
- **Country** : Pays du siège social de la marque de la voiture

Au total **4 nouvelles features** sont ajoutées au DataFrame initial :

Tableau 1 : les données avec les critères ajoutés

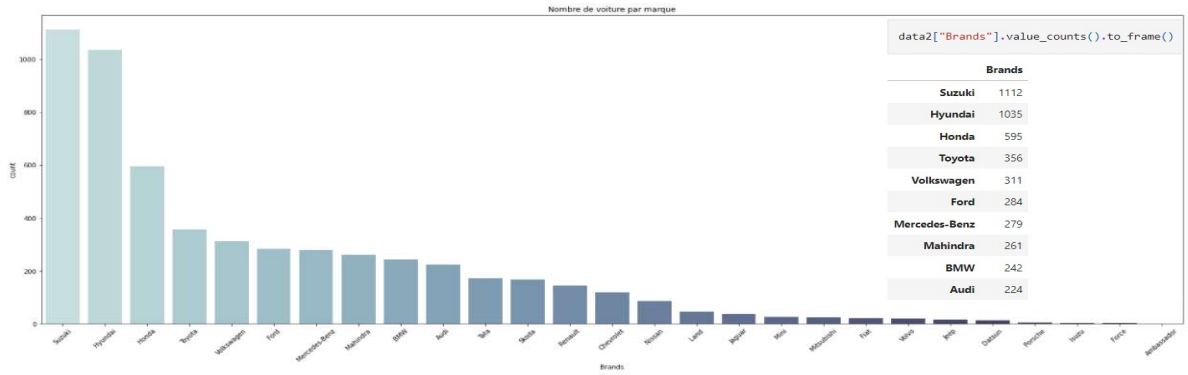
	Brands	Seniority	Owner_Type	Seats	Kilometers_Driven	Fuel_Type	Transmission	Mileage	Engine	Power	Price	Country	Continent
0	Hyundai	6	First	5	41000.0	Diesel	Manual	19.67	1582.0	126.20	12.50	Korea	Asia
1	Honda	10	First	5	46000.0	Petrol	Manual	18.20	1199.0	88.70	4.50	Japan	Asia
2	Suzuki	9	First	7	87000.0	Diesel	Manual	20.77	1248.0	88.76	6.00	Japan	Asia
3	Audi	8	Second	5	40670.0	Diesel	Automatic	15.20	1968.0	140.80	17.74	Germany	Europe
4	Nissan	8	First	5	86999.0	Diesel	Manual	23.08	1461.0	63.10	3.50	Japan	Asia

5 Analyse finale – Visualisation

A partir de ce nouveau DataFrame plusieurs analyses seront effectués :

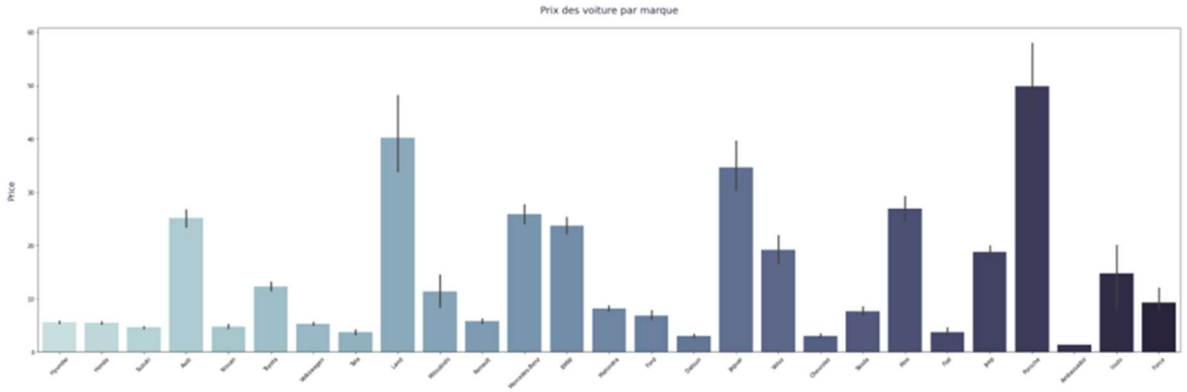
On peut ici observer que les voitures de marque Asiatique Suzuki, Hyundai, Honda et Toyota sont les voitures d'occasion les plus vendues en Inde.

Graphique 1 : nombre de voitures vendues selon la marque

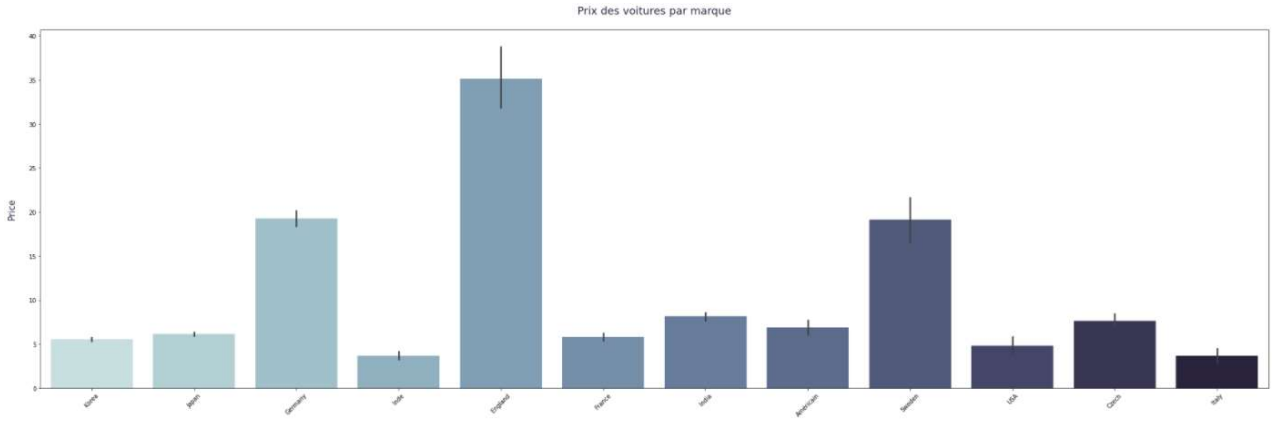


Graphique 2 : Prix des voitures vendues selon la marque

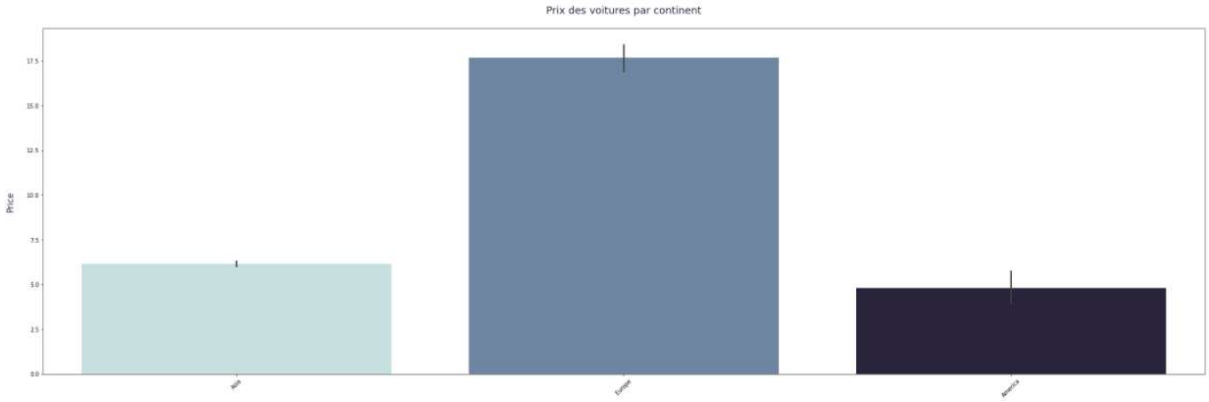
Ce graphique met en évidence que les marques de voiture Porsche, Land Rover et Jaguar sont de loin les plus chères.



Graphique 3 : Prix des voitures vendues selon le pays du siège social de la marque



Graphique 4 : Prix des voitures vendues selon le continent du siège social de la marque

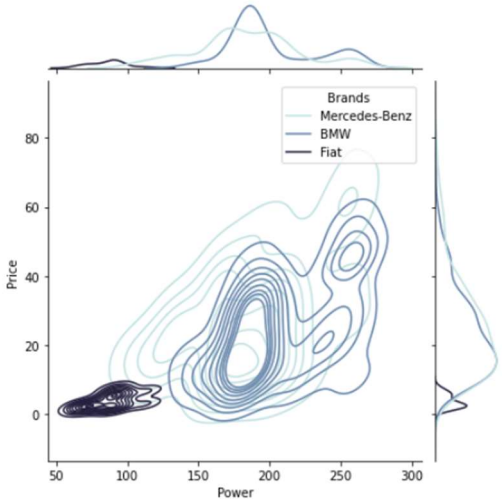


Ces visualisations graphiques permettent de dire qu'en règle générale (et au sein de mon jeu de donnée), **les voitures Européennes sont les plus chères**.

On peut aussi observer que **les marques les plus chères sont Anglaises**. En effet à ce niveau de nettoyage des données il reste 3 marques Anglaise : Jaguar, LandRover et Mini. Ce sont des marques plutôt luxueuses. Ce sont ensuite les marques Allemandes et Suédoises qui sont les plus chères. Ce sont des marques prisées, et orientées sur le marché de l'automobile « haut de gamme » et sportives. Audi, BMW, Mercedes, Porsche et Volvo constituent ces marques Allemandes et Suédoises.

Ce Jointplot permet de mettre en évidence que d'une marque à l'autre, les domaines de prix et de puissance peuvent être complétement distincts

Graphique 5 : prix et puissance du véhicule en fonction de sa marque (ici uniquement Mercedes, BMW et Fiat)



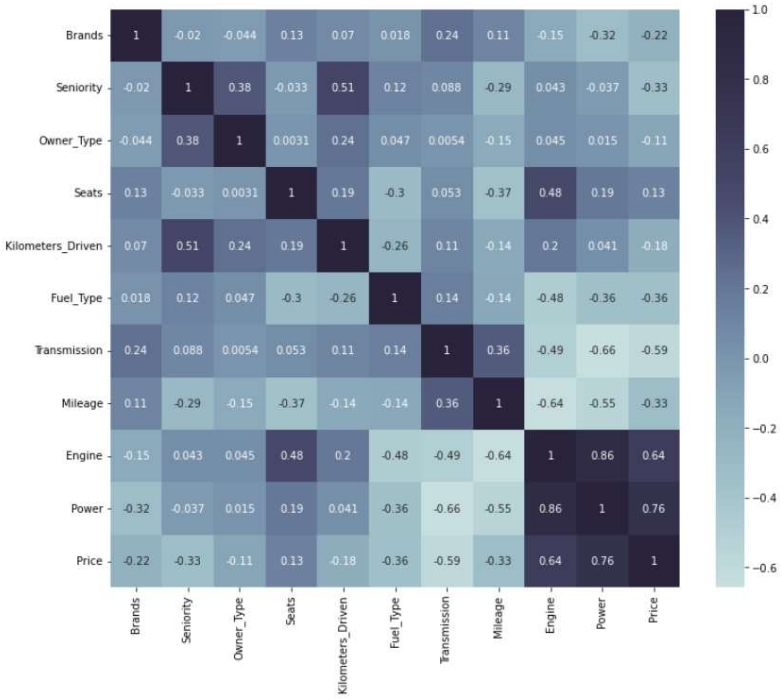
A partir de ce nouveau DataFrame plusieurs tests seront effectués :

- Utilisation de la feature « Seniority »
- Utilisation de la feature « Brands »
- Uniquement 5 et 7 sièges au sein de la colonne « Seats »
- Utilisation de la feature Continent
- Utilisation de la feature Country

Finalement sur l'ensemble des tests réalisés, uniquement les features « **Seniority** » et « **Brands** » s'avéreront concluantes, seront ajoutées aux données et entrainées.

6 Corrélation des données

Tableau 2 : matrice des corrélations



Cette matrice de corrélation me permet d'étudier les différentes relations entre les variables de mon DataFrame :

- Le nombre de chevaux din (**Power**) et la cylindrée (**Engine**) sont étroitement liés : **0.86**
- Plus le nombre de chevaux din (**Power**) est important plus le prix de la voiture (**Price**) est élevé : **0.76**
- La transmission (**manuelle ou automatique**) est fortement liée à la puissance de la voiture (**Power**) : **- 0.66**
- Plus la cylindrée (**Engine**) est importante plus le prix de la voiture (**Price**) est élevé : **0.64**
- Plus le nombre de chevaux din (**Power**) est important plus la consommation de la voiture (**Mileage**) est élevée : - **0.55**
- Plus une voiture est ancienne (**Seniority**) plus son nombre de kilomètres parcourus (**Kilometers_Driven**) est important : **0.51**
- Le nombre de place (**Seats**) a une corrélation liée à la cylindrée (**Engine**) d'une voiture : **0.48** (les voitures 2 places ont de grosses cylindrées)
- La cylindrée (Engine) est aussi étroitement liée au type de carburant de la voiture (**Fuel_Type**) en question : **- 0.48**
- Plus une voiture est ancienne (**Seniority**) plus son nombre de mains (**Owner_Type**) est important : **0.38**
- La transmission a aussi une corrélation non négligeable avec la consommation (**Mileage**) : **0.36**

D'autres relations peuvent aussi être évoquées lors de cette phase d'analyse. Mais celle-ci sont bien moins « marquées » que les relations citées au-dessus.

7 Normalisation des données

En intelligence artificielle il primordial de travailler avec des valeurs contenues sur des échelles similaires ou proches.

Le jeu de donnée possède des features **numériques** et **catégorielles**. Je décide donc de créer un objet « preprocessor » issu de la classe « Preprocessor ».

Ce « preprocessor » a pour but d'appliquer un « MinMaxScaler (Voir annexe 3) » sur les colonnes de type numériques et un « OneHotEncoder (Voir annexe 4) » sur les colonnes de types catégorielles.

Après normalisation (Voir annexe 2) **des données notre DataFrame contient 5603 lignes pour 39 colonnes, comprenant uniquement des valeurs entre 0 et 1.**

Tableau 3 : les données après normalisation

```
DATA shape: (5603, 12)
DATA shape:: (5603, 39)
```

	Seniority	Owner_Type	Seats	Kilometers_Driven	Mileage	Engine	Power	Continent	Brands*Hyundai	Brands*Honda	...	Brands*Fiat	Brands*Jeep	Brands*Porsche	Brands
0	0.190476	0.000000	0.375	0.270697	0.567822	0.372183	0.388137	0.0	0.0	0.0	...	0.0	0.0	0.0	
1	0.380952	0.000000	0.375	0.303847	0.495050	0.223388	0.229929	0.0	0.0	0.0	...	0.0	0.0	0.0	
2	0.333333	0.000000	0.625	0.575678	0.622277	0.242424	0.230182	0.0	0.0	0.0	...	0.0	1.0	0.0	
3	0.285714	0.333333	0.375	0.268509	0.346535	0.522145	0.449732	0.5	0.0	1.0	...	0.0	0.0	0.0	
4	0.285714	0.000000	0.375	0.575672	0.736634	0.325175	0.121925	0.0	0.0	0.0	...	0.0	0.0	0.0	

5 rows × 39 columns

8 Création d'un Modèle : V2

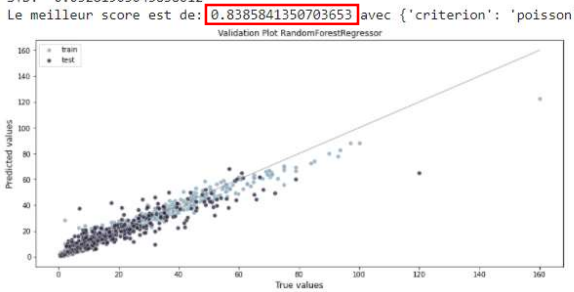
8.1 Résultats Modèle V1

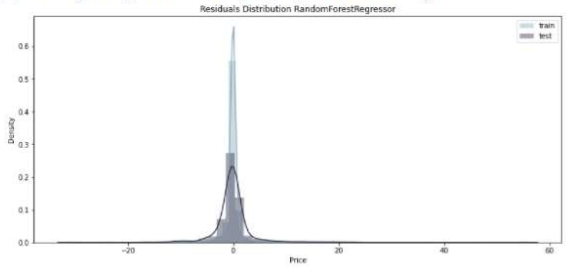
Le résultat obtenu dans le modèle issu du projet MARCO CASION (modèle prédictif de la valeur d'achat d'un véhicule d'occasion sur le marché Indien), amène à un R2 de 0.83%, résultat obtenu grâce à l'algorithme **RandomForestRegressor**.

```
Score du jeu TRAIN
MAE:  0.6186984909058257
RMSE:  1.5054327317568839
Median abs err:  0.2732499999999969
R2:  0.9820694671649806

Score du jeu TEST
MAE:  1.6259364978448279
RMSE:  3.69731636443902
Median abs err:  0.7038499999999985
R2:  0.8947465579597457

Score après GridSearchCV
CV Mean:  0.8673514064356065
STD:  0.03281903043838012
Le meilleur score est de: 0.8385841350703653 avec {'criterion': 'poisson', 'max_depth': 80, 'max_features': 'sqrt', 'n_estimators': 50}
```





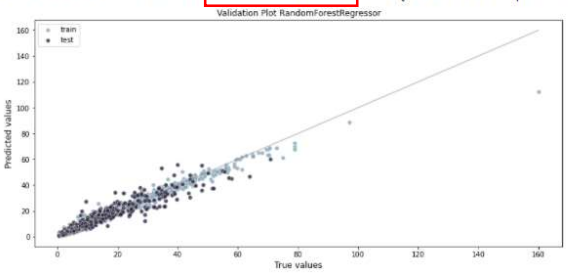
8.2 Résultats Modèle V2

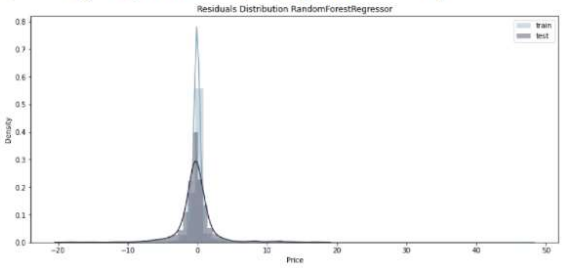
En reprenant la globalité des anciens hyperparamètres (voir Annexe 5) issus de l'algorithme RandomForestRegressor du projet « MARCO CASION », l'ensemble des scores **(MAE, RMSE, R2 : Annexe 11)** est nettement amélioré !

```
Score du jeu TRAIN
MAE:  0.5273076153821634
RMSE:  1.1584698074791957
Median abs err:  0.2542500000000074
R2:  0.9873718004920993

Score du jeu TEST
MAE:  1.295791598204551
RMSE:  2.595567155808776
Median abs err:  0.6578000000000017
R2:  0.9256255833975476

Score après GridSearchCV
CV Mean:  0.904516461915185
STD:  0.029935922857232764
Le meilleur score est de: 0.8938230870690391 avec {'criterion': 'poisson', 'max_depth': 100, 'max_features': 'sqrt', 'n_estimators': 100}
```





9 Conclusion

9.1 Tableau récapitulatif des scores

Modèles	**R2 (en %)**
LinearRegression ***V1***	0.72
RandomForestregression ***V1***	0.83
LinearRegression ***V2***	0.79
RandomForestregression ***V2***	0.89

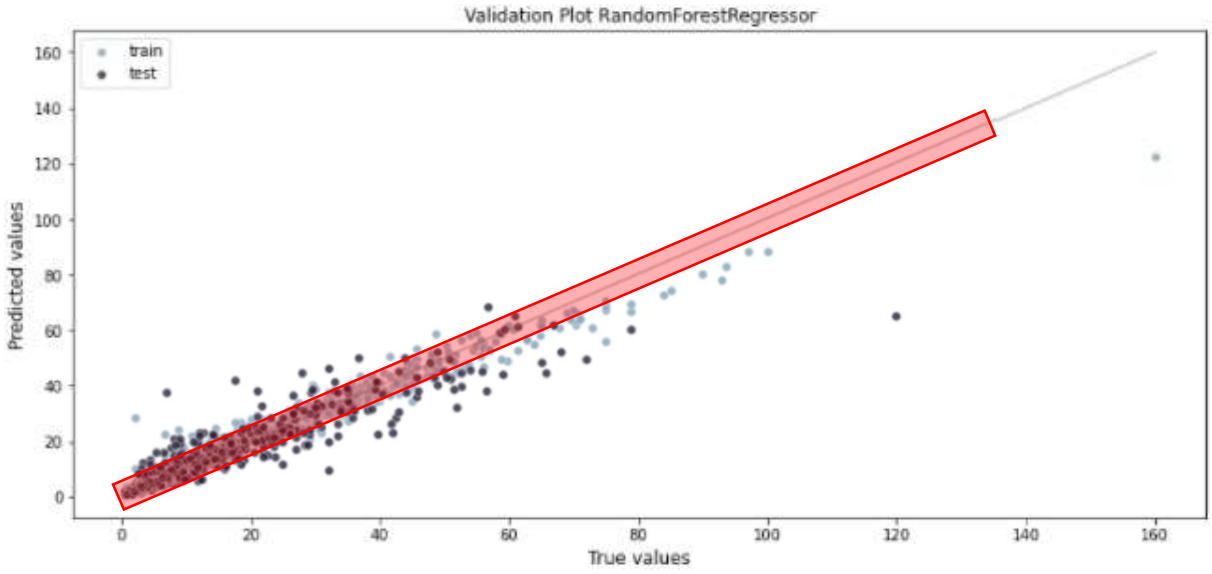
Grâce à un **nettoyage approfondi des données** (notamment à l'aide du Z score), à une analyse poussée (graphiques visuels, matrice de corrélation), aux différentes techniques de **Feature Engineering** employées (simplification des données, création de nouvelles features) et aux **divers tests** effectués, **l'ensemble des divers scores de l'algorithme RandomForestRegressor utilisé lors du projet « MARCO CASION » est nettement amélioré !** Le R2 du nouveau modèle est de **0.89 %.**

Bien entendu ce score n'est pas parfait et reste encore certainement perfectible. Ainsi plusieurs autres critères manquants permettraient probablement d'obtenir de meilleures prédictions :

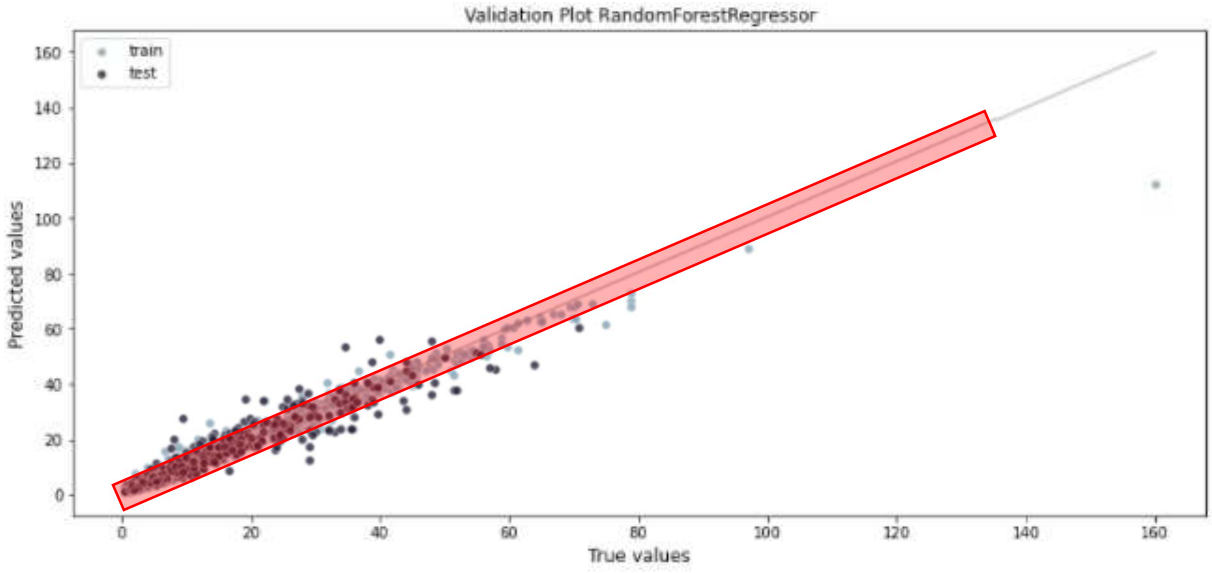
- Un plus grand nombre de données permettrait d'obtenir une prédiction plus fiable sur le périmètre actuel cela pour les catégories de véhicules les moins représentées
- L'ajout de véhicules plus anciens à la base de données
- L'état du véhicule
- Poids du véhicule
- Type de véhicule (citadine, berline, 4X4...)
- Options du véhicule
- Finitions du véhicule

9.2 Comparaison des modèles

Modèle RandomForestRegressor V1 : R2 = 0.83%



Modèle RandomForestRegressor V2 : R2 = 0.89%



Le rectangle rouge qui est positionnée sur les graphiques ci-dessus nous permet de bien visualiser les points pour lesquels l'estimation est moins bonne (points hors du rectangle). **Cette visualisation, en préservant les mêmes échelles, permet de plus facilement mettre en évidence l'amélioration du modèle V2 par rapport au modèle V1.**

10 Annexes

10.1 Annexe 1 : Qu'est-ce qu'un DataFrame ?

Source :

- http://www.xavierdupre.fr/app/ensae_teaching_cs/helpsphinx/notebooks/td1a_cenonce_session_10.html

Un Data Frame est un objet qui est présent dans la plupart des logiciels de traitements de données, c'est une **matrice**, chaque colonne est de même type (nombre, dates, texte), elle peut contenir des valeurs manquantes. On peut considérer chaque colonne comme les variables d'une table.

En bref un DataFrame c'est :

- La structure de données la plus utilisée dans un projet de Data Science
- Un tableau avec des lignes et des colonnes, comme une table SQL ou une feuille de calcul Excel
- Une table, qui peut stocker en mémoire des données dans des formats différents
- Une ou plusieurs Series
- Mutable

Exemple d'un DataFrame :

	state	color	food	age	height	score
Jane	NY	blue	Steak	30	165	4.6
Niko	TX	green	Lamb	2	70	8.3
Aaron	FL	red	Mango	12	120	9.0
Penelope	AL	white	Apple	4	80	3.3
Dean	AK	gray	Cheese	32	180	1.8
Christina	TX	black	Melon	33	172	9.5
Cornelia	TX	red	Beans	69	150	2.2

10.2 Annexe 2 : Normalisation des données

Source :

- https://dataanalyticspost.com/Lexique/normalisation/#:~:text=Normalisation%20%3A%20La%20normalisation%20est%20une,l'application%20de%20certains%20algorithmes.&text=Cette%20m%C3%A9thode%20a%20en%20outre,dans%20la%20fouille%20de%20donn%C3%A9es.

Normalisation : La normalisation est une méthode de prétraitement des données qui permet de réduire la complexité des modèles. C'est également un préalable à l'application de certains algorithmes. ... Cette méthode a en outre de nombreuses applications dans la fouille de données.

10.3 Annexe 3 : MinMaxScaler

Transforme les entités en adaptant chaque entité à une plage donnée. Cet estimateur met à l'échelle et traduit chaque caractéristique individuellement de telle sorte qu'elle se situe dans la plage donnée sur l'ensemble d'apprentissage, par exemple entre zéro et un.

10.4 Annexe 4 : OneHotEncoder

Encode les caractéristiques catégorielles sous la forme d'un tableau numérique unique. L'entrée de ce transformateur doit être un tableau d'entiers ou de chaînes, indiquant les valeurs prises par les caractéristiques catégorielles (discrètes). Les caractéristiques sont encodées à l'aide d'un schéma d'encodage one-hot (alias 'one-of-K' ou 'dummy'). Cela crée une colonne binaire pour chaque catégorie et renvoie une matrice clairsemée ou un tableau dense.

10.5 Annexe 5 : Hyperparamètres

Sources :

- https://fr.wikipedia.org/wiki/Hyperparam%C3%A8tre

Dans l'apprentissage automatique, un hyperparamètre est un paramètre dont la valeur est utilisée pour contrôler le processus d'apprentissage. En revanche, les valeurs des autres paramètres (généralement la pondération de nœuds) sont obtenues par apprentissage.

Un exemple d'hyperparamètre de modèle est la topologie et la taille d'un réseau de neurones. Des exemples d'hyperparamètres d'algorithme sont la vitesse d'apprentissage et la taille des lots.

Les différents hyperparamètres varient en fonction de la nature des algorithmes d'apprentissage, par exemple certains algorithmes d'apprentissage automatique simples (comme la régression des moindres carrés) n'en nécessitent aucun. Compte tenu de ces hyperparamètres, l'algorithme d'apprentissage apprend les paramètres à partir des données. Par exemple, la régression LASSO est un algorithme qui ajoute un hyperparamètre de régularisation à la régression des moindres carrés, qui doit être défini avant d'estimer les paramètres via l'algorithme d'apprentissage.

10.6 Annexe 6 : Score (MAE, RMSE, R2)

MAE : Différence absolue entre les vraies valeurs et les valeurs prédites.

MSE : Moyenne des écarts au carré entre les vraies valeurs et les valeurs prédites.

RMSE : Correspond à la racine carrée du MSE.

Median ABS error : Médiane des différences absolues des erreurs.

CV mean : Moyenne des différents score R2 produits après avoir effectué un GridSearch.

STD : Dispersion des points autour de la moyenne des différentes distributions.