Compte rendu du projet

Phase de prétraitement

- Nous commençons par supprimer les colonnes vide car elle ne constitue que 2.7% du jeu de donnée total. Ce que je considère comme négligeable.

On remarque que la colonne couple contient plus d'une donnée sous ce format particulier (par exemple 173Nm@ 4000rpm).

On distingue 2 unités Nm et rpm et d'autre fois on à kgm à la place de Nm

D'autre fois encore on a une plage de valeurs en rpm.

- Pour résoudre ce problème je divise la colonne couple en 2 parties couple_nm et couple_rpm
- Et je fais la moyenne pour les plages de valeurs de rpm.
- Je converti les Kgm en Nm

couple
173Nm@ 4000rpm
343Nm@ 1400-3400rpm
24 KGM at 1900-2750 RPM

couple_nm	couple_rpm
173.00	4000.0
343.00	2400.0
235.44	2325.0

- Conversion de la toute la colonne consommation en kmpl

- Je supprime les unités de toutes les paramètre numérique (consommation, moteur et puissance)

Analyse du jeu de données

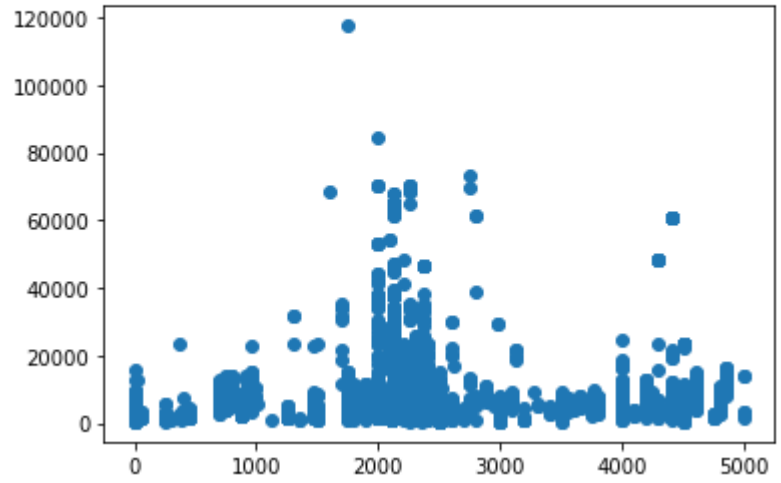
- Je calcule les coefficients de corrélation pour savoir quelles variables garder

Correlation des variables numériques

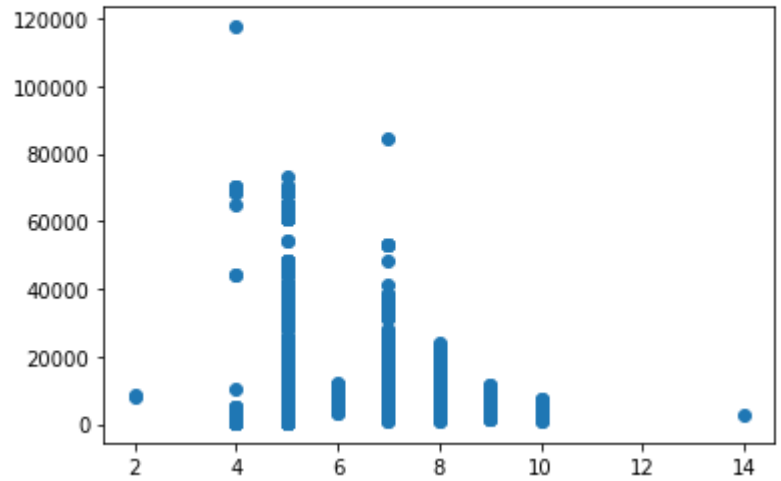
```
In [15]: df.corr().loc["prix_de_vente"]

Out[15]: annee            0.411999
         prix_de_vente    1.000000
         kilometrage     -0.222332
         consommation    -0.129107
         moteur           0.456639
         puissance        0.750211
         nb_sieges        0.040712
         couple_nm        0.623208
         couple_rpm       0.012049
         Name: prix_de_vente, dtype: float64
```

- Doit ont garder les colonnes couple_rpm et nb_siège ? finalement oui après avoir vu leurs distributions par rapport aux prix de ventes

```
plt.scatter(df['couple_rpm'], df['prix_de_vente'])
plt.show()
```




```
plt.scatter(df['nb_sieges'], df['prix_de_vente'])
plt.show()
```



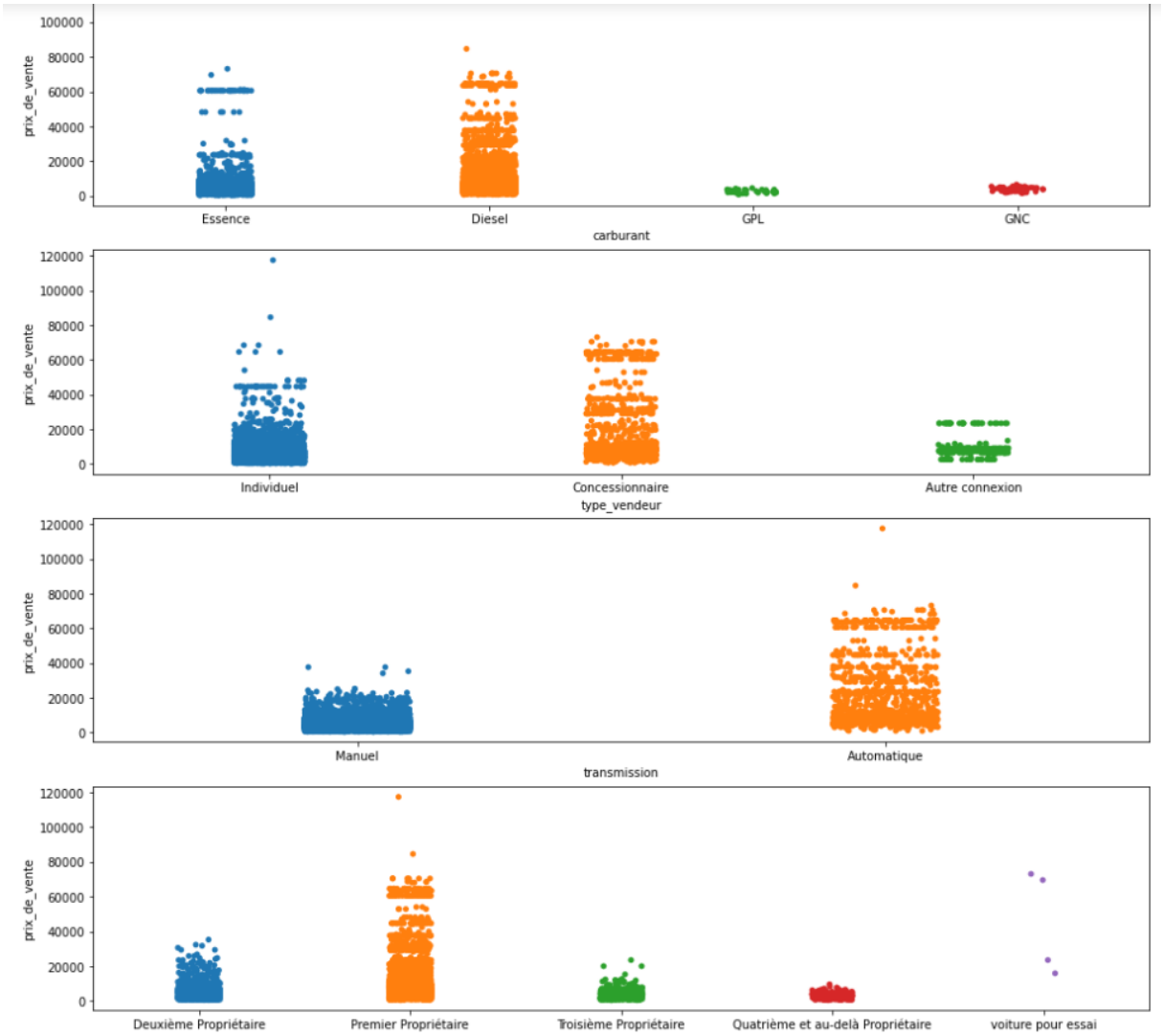
- Je garde ces variables car je ne suis pas sur qu'elles n'affectent pas le prix de la voiture

- Je passe au traitement des variables non numériques

- Je remarque qu'il y'a trop de nom de voitures distinctes (1965) ce qui rend ce paramètre inutilisable en état

- On va donc se limiter à la marque de la voiture uniquement

- On garde aussi les autres variables non numériques car elle semble affecter le prix de la voiture.



- On fait un encodage avec les variables non numérique de façon à pouvoir s'en servir pour la prédiction.

Phase d'apprentissage

- J'utilise 70% du data set pour l'entrainement et 30% pour le test (c'est avec cette répartition que j'ai eu la plus grande précision avec l'un des modèles que j'ai choisi.

Réglage des modèles
4 modèles pour ce problème :
Linear regression
Decision tree regressor
Random forest regressor
XGB Regressor

Mésure d'évaluation utilisés
Que j'évalue selon les mesures suivantes : Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, Accuracy on Training set, Accuracy on Test set.

Comparaison entre différents modèles :

```
                Linear regression Error Table
Mean Absolute Error      :  3285.2574076282435
Mean Squared  Error      :  29782692.15178709
Root Mean Squared  Error :  5457.35211909467
Accuracy on Training set : 0.6716258179335516
Accuracy on Test set : 0.6462980665287942

                Decision Tree Regressor Error Table
Mean Absolute Error      :  886.0440977517106
Root Mean Squared  Error :  1644.0289232030227
Accuracy on Training set :  0.9996050491773518
Accuracy on Test set :  0.9679009345038411

                Random Forest Regressor Error Table
Mean Absolute Error      :  760.2022556400735
Root Mean Squared  Error :  1412.6696809394177
Accuracy on Training set :  0.9945424773977174
Accuracy on Test set :  0.976299651611947

               XGBRegressor Error Table
Mean Absolute Error      :  731.5305033890919
Root Mean Squared  Error :  1382.0036049175396
Accuracy on Training set :  0.9957716377805639
Accuracy on Test set :  0.9773174522830911
```

Meilleur modèle

Le meilleur modèle est XGB Regressor
Un tracé pour comparer la prédiction du meilleur modèle avec les valeurs réelles