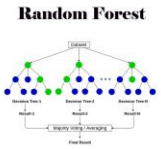
كلية العلوم
FACULTÉ DES SCIENCES

جامعة مولاي إسماعيل
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵎⵓⵍⴰⵢ ⵉⵙⵎⴰⵄⵉⵍ
UNIVERSITÉ MOULAY ISMAÏL

Master

Intelligence Artificielle et Analyse des Données

Système de Prédiction pour Concessionnaire Automobile en Utilisant Random Forest avec Apache Spark et Cassandra



Préparé par :

AHANSAL Salah Eddine
EL KADAH Rachid
EL HAMDI Brahim

Professeur :

Mr. y.oubenaalla

Année universitaire 2024-2025

1. Introduction

Le secteur automobile est en pleine transformation, avec une demande croissante pour des solutions personnalisées et adaptées aux besoins des clients. Les concessionnaires automobiles cherchent à améliorer l'expérience client en proposant des véhicules adaptés à leurs préférences et à leur profil. Ce projet vise à développer un système de prédiction qui utilise des techniques de machine learning pour recommander des catégories de véhicules en fonction des caractéristiques des clients.

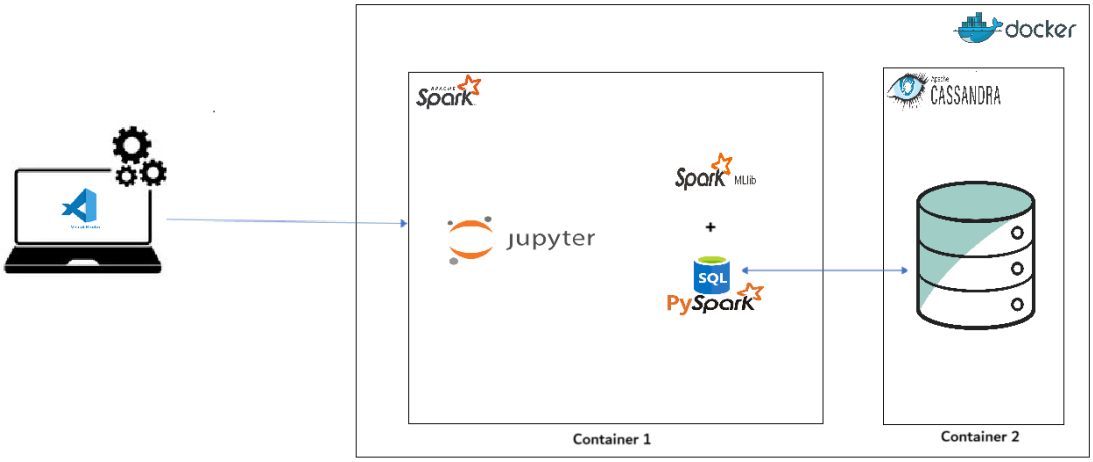
Problématique :

Les concessionnaires disposent de grandes quantités de données sur les clients et les véhicules, mais ces données sont souvent sous-exploitées. L'objectif est de tirer parti de ces données pour prédire les préférences des clients et optimiser les ventes.

Objectifs du Projet :

- Utiliser Apache Spark pour le traitement des données à grande échelle.
- Stocker les données dans Cassandra pour une gestion efficace.
- Explorer et comprendre les données issues des clients, du catalogue et des immatriculations tout en effectuant un nettoyage rigoureux pour garantir leur qualité et leur cohérence.
- Appliquer des techniques de clustering pour segmenter les véhicules.
- Utiliser un modèle Random Forest pour prédire la catégorie de véhicule adaptée à un client.
- Développer une application Streamlit pour interagir avec le modèle.

2.Architecture du Projet et Configuration des Outils



1. Local Machine (VS Code) :

Le projet démarre sur une machine locale, où l'utilisateur utilise **Visual Studio Code (VS Code)** comme environnement de développement. VS Code permet d'écrire, de tester et d'exécuter du code Python et des scripts Shell. L'utilisateur interagit avec les services via des commandes Docker, qui lancent et gèrent les conteneurs nécessaires pour le projet. Les ports exposés par les conteneurs permettent d'accéder aux interfaces utilisateur telles que Jupyter Notebook et Streamlit.

2. Docker et Conteneurs :

Docker est utilisé pour créer des environnements isolés et portables. Deux conteneurs principaux sont configurés :

```yaml
version: '3'
services:

  cassandra:
    image: cassandra:latest
    container_name: cassandra
    ports:
      - "9042:9042"  # Cassandra CQL port
    environment:
      - CASSANDRA_CLUSTER_NAME=MyCluster
    volumes:
      - cassandra_data:/var/lib/cassandra

  spark:
    image: apache/spark:latest
    container_name: spark
    user: root  # Run as root
    ports:
      - "8888:8888"  # Jupyter Notebook
      - "4040:4040"  # Spark UI
      - "8501:8501"  # Streamlit APP
    volumes:
      - ./:/opt/spark/work-dir  # Mount a local directory for notebooks
    depends_on:
      - cassandra
    command: >
      /bin/bash -c "
      apt update &&
      apt install -y python3-pip &&
      pip install jupyter numpy pandas cqlsh matplotlib seaborn streamlit&&
      /opt/spark/bin/pyspark --packages com.datastax.spark:spark-cassandra-connector_2.12:3.2.0 --master local[*] --conf spark.driver.memory=2g --conf spark.executor.memory=2g &&
      jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser --allow-root"
    environment:
      - PYTHONPATH=/opt/spark/python/lib/pyspark.zip:/opt/spark/python/lib/py4j-0.10.9.7-src.zip

volumes:
  cassandra_data:
```

Conteneur Spark :

- Ce conteneur exécute **Apache Spark** pour le traitement distribué des données et **Jupyter Notebook** pour l'analyse interactive.

- Les ports suivants sont exposés :
    - **8888** : Pour accéder à **Jupyter Notebook**.
    - **4040** : Pour accéder à l'interface utilisateur de **Spark**.
    - **8501** : Pour accéder à l'application **Streamlit**.
- Un volume est monté pour partager des fichiers entre la machine locale et le conteneur, permettant de travailler sur des notebooks et des scripts.
- Les dépendances Python nécessaires (numpy, pandas, cqlsh , matplotlib , seaborn , streamlit) sont installées dans le conteneur.

Conteneur Cassandra :

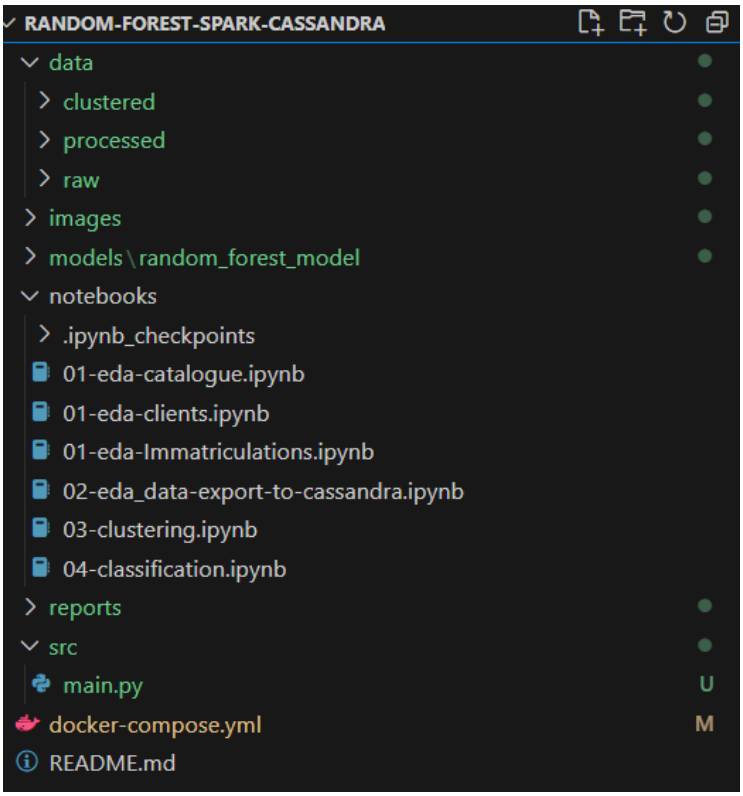
- Ce conteneur exécute **Cassandra**, une base de données NoSQL, pour stocker les données structurées du projet (clients, catalogue, immatriculations).
- Le port **9042** est exposé pour permettre les requêtes **CQL (Cassandra Query Language).**
- Un volume Docker est utilisé pour persister les données de Cassandra, garantissant que les données ne sont pas perdues lors de l'arrêt du conteneur.

3. Flux de Travail :

Le flux de travail du projet est divisé en trois étapes principales :

1. **Local Machine** :
    - L'utilisateur lance les conteneurs Docker avec la commande **docker-compose up**.
    - Il accède à Jupyter Notebook via http://localhost:8888 pour exécuter des analyses de données et des modèles de machine learning.
    - Il utilise l'application Streamlit via http://localhost:8501 pour interagir avec le modèle de prédiction.
2. **Spark Container** :
    - Spark est utilisé pour le prétraitement des données, le clustering (segmentation des véhicules) et la classification (prédiction de la catégorie de véhicule).
    - Le connecteur Spark permet de charger les données depuis Cassandra et de sauvegarder les résultats (clusters, prédictions) dans la base de données.
3. **Cassandra Container :**
    - Cassandra stocke les données brutes et les résultats intermédiaires.
    - Il fournit un accès rapide et scalable aux données pour Spark, garantissant des performances optimales lors du traitement des données.
    -

4. Structure de Projet :



Le projet est organisé en plusieurs dossiers et fichiers pour une gestion claire et modulaire des ressources. Voici une description de chaque composant :

- **data** : Contient les fichiers de données brutes et traitées utilisés dans le projet (par exemple, clients.csv, catalogue.csv, immatriculations.csv).
- **images** : Stocke les images utilisées dans l'application Streamlit pour illustrer les catégories de véhicules (par exemple, familiales.png, luxe.png).
- **models** :Contient les modèles de machine learning sauvegardés (par exemple, random_forest_model).
- **notebooks** :Inclut les notebooks Jupyter utilisés pour l'analyse des données, le clustering et la classification :

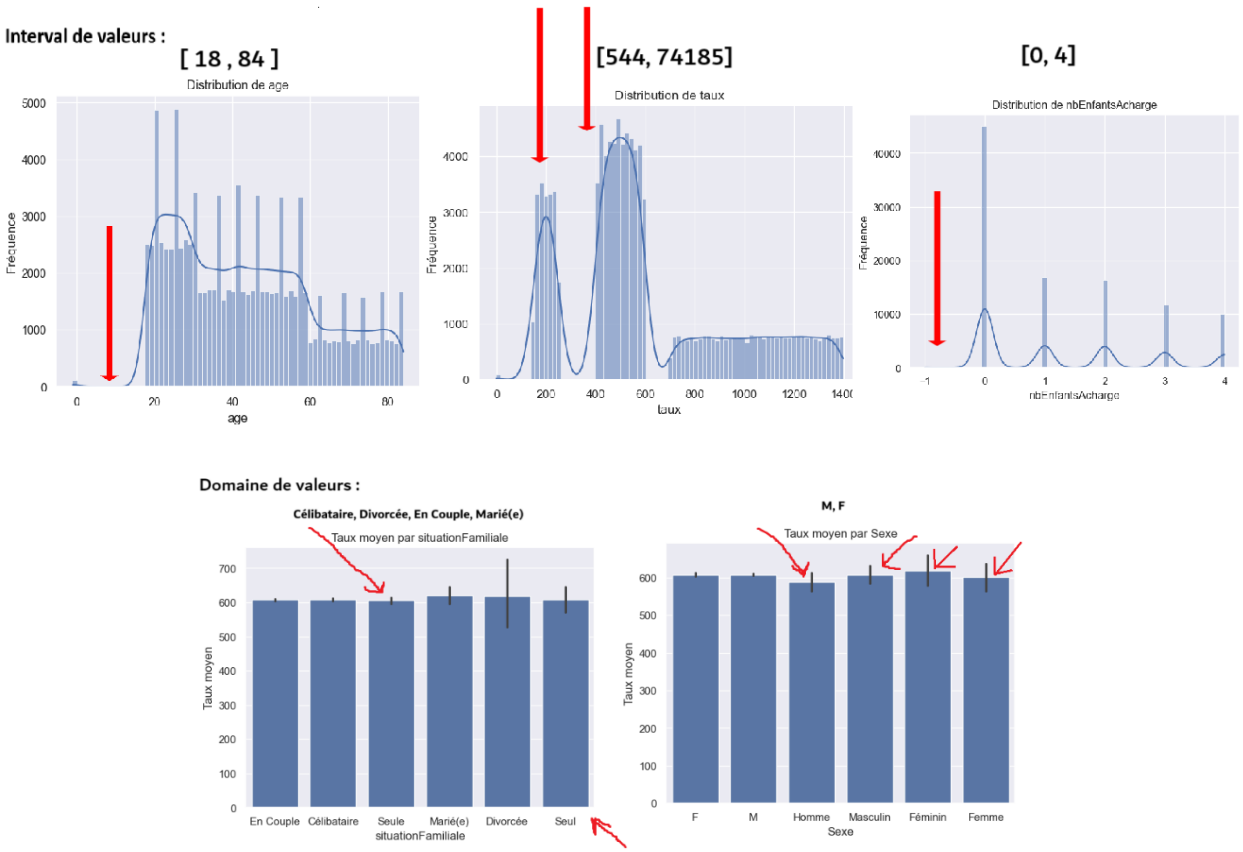
  - ✓ **01-eda-catalogue.ipynb :** Exploration des données du catalogue.
  - ✓ **01-eda-clients.ipynb :** Exploration des données des clients.
  - ✓ **01-eda-immatriculations.ipynb :** Exploration des données des immatriculations.
  - ✓ **02-eda_data-export-to-cassandra.ipynb :** Export des données traitées vers Cassandra.
  - ✓ **03-clustering.ipynb :** Clustering des véhicules.
  - ✓ **04-classification.ipynb :** Classification des clients en fonction de leurs préférences.

- **reports** :Contient les rapports générés pendant le projet (par exemple, des analyses statistiques, des visualisations).
- **src** :Inclut le code source de l'application Streamlit (main.py).
- **docker-compose.yml** :Fichier de configuration Docker pour lancer les conteneurs Spark et Cassandra.
- **README.md** :Documentation du projet, expliquant comment configurer et exécuter le projet.

3.Analyse exploratoire des données

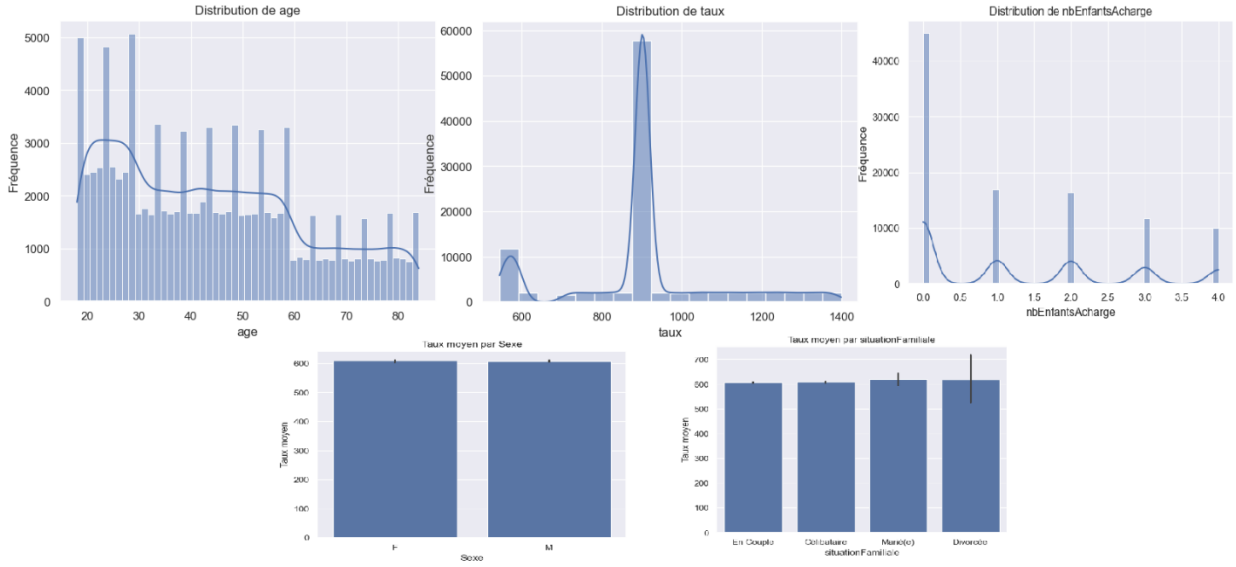
PARTIE 1 : CLIENTS

Tout d'abord, il est nécessaire de **nettoyer les données** en traitant les valeurs manquantes ou incorrectes, comme les erreurs de saisie. Pour gérer les valeurs manquantes, **les colonnes numériques** ont été complétées en remplaçant les valeurs manquantes par **la médiane**, tandis que pour **les colonnes catégorielles**, les valeurs manquantes ont été remplacées par **la modalité** la plus fréquente. Après cette étape, des valeurs incorrectes, situées hors des domaines définis par le dictionnaire de données, ont été identifiées :

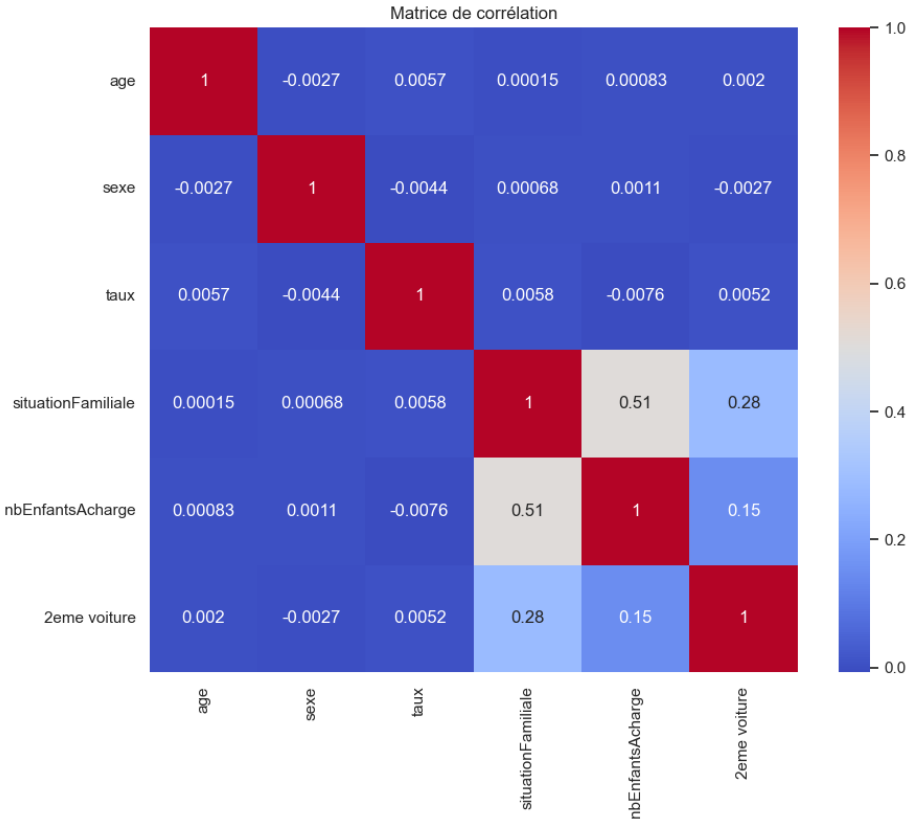


En parallèle du traitement **des valeurs manquantes**, nous avons corrigé les valeurs **incohérentes** ou **incorrectes** dans les données. Les données hors domaine ont été supprimées pour les colonnes telles que ***age*** et ***nbEnfantsAcharge***, car elles représentaient moins de **5%** des enregistrements. Cependant, pour la colonne ***taux***, qui comportait **55,82%** de valeurs incorrectes, une approche différente a été adoptée. **La moyenne** des valeurs valides (respectant les contraintes) a été calculée et utilisée pour remplacer les valeurs incorrectes.

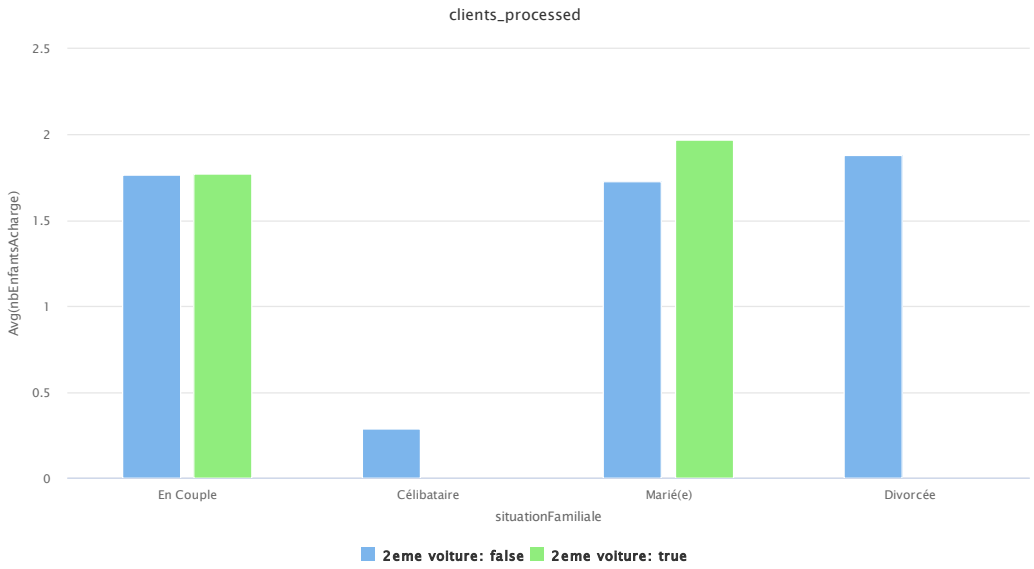
Pour les colonnes catégorielles, nous avons standardisé les données en remplaçant les valeurs par leurs **synonymes** pour garantir une cohérence. Par exemple, dans la colonne ***sexe***, les valeurs comme "Homme" et "Masculin" ont été uniformisées en "M", tandis que "Femme" et "Féminin" ont été converties en "F". De même, dans la colonne ***situationFamiliale***, des valeurs telles que "Seule" et "Seul" ont été remplacées par "Célibataire". Cette étape a permis d'harmoniser les données catégorielles tout en respectant leur sémantique.



Nous avons analysé les corrélations entre les attributs des clients, comme l'âge, le taux, la situation familiale, le nombre d'enfants à charge et la possession d'une deuxième voiture. Cela nous a permis d'identifier les relations importantes entre ces variables :

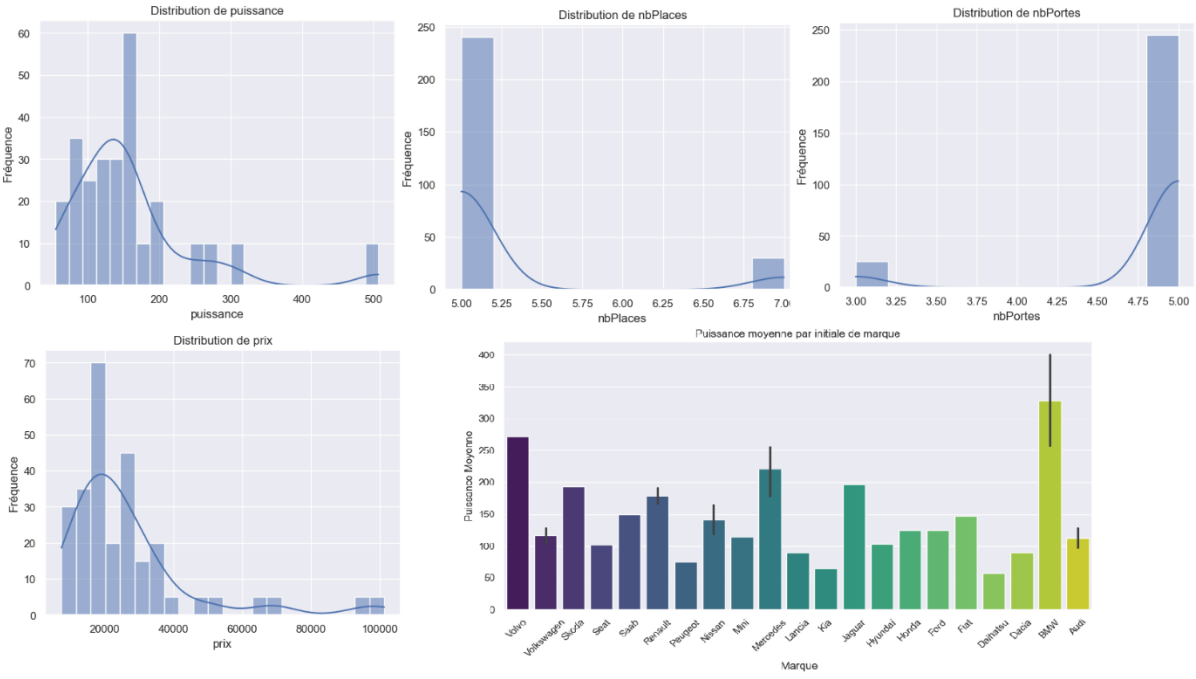


Les corrélations les plus fortes sont observées entre le nombre d'enfants, la situation familiale et la possession d'une deuxième voiture. Après avoir analysé ces colonnes, nous avons constaté que la moyenne du nombre d'enfants est très faible chez les célibataires, tandis que les personnes en couple et mariées sont celles qui possèdent le plus souvent une deuxième voiture, avec également un nombre d'enfants plus élevé.

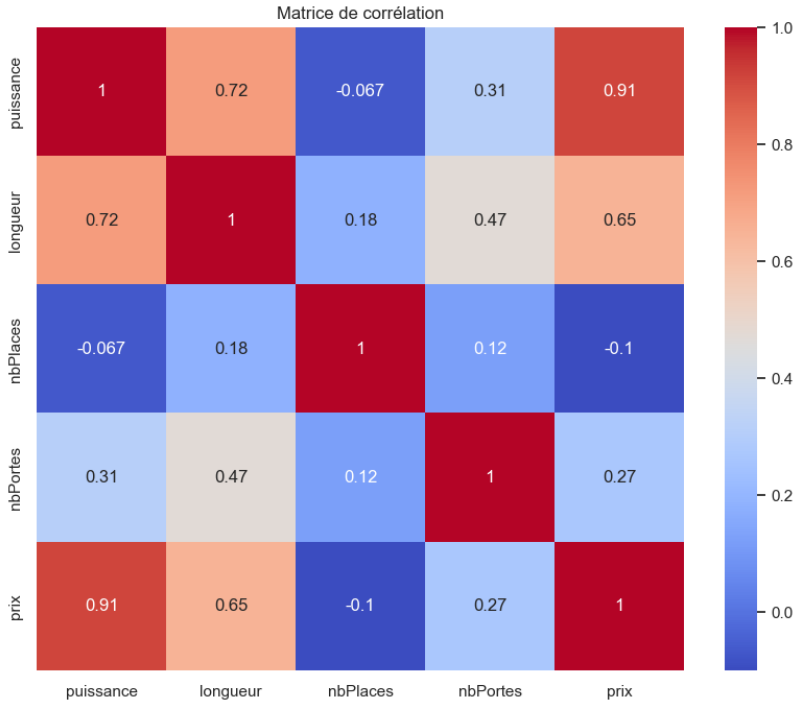


PARTIE 2 : CATALOGUE

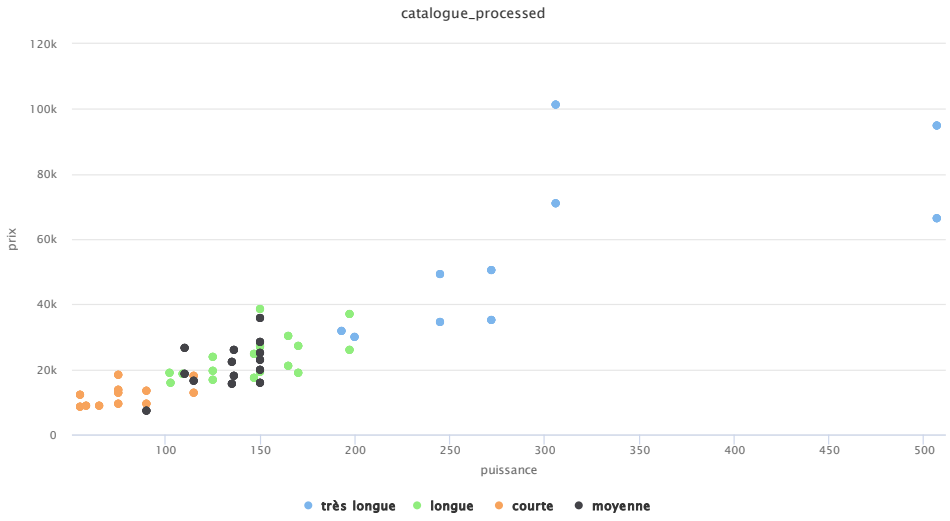
Concernant le catalogue, nous avons constaté que la distribution des données respecte bien les domaines de valeurs définis. Par conséquent, la distribution des attributs est cohérente avec les attentes :



Remarque : À partir de ce point, notre travail s'est concentré exclusivement sur les voitures neuves :



Les corrélations les plus fortes sont observées entre **la puissance**, **le prix** et **la longueur**. Après avoir analysé ces colonnes, nous avons constaté que **l'augmentation** de la puissance s'accompagne d'une **hausse** du prix. De plus, chaque fois que la longueur **augmente**, la puissance et le prix **augmentent** également.

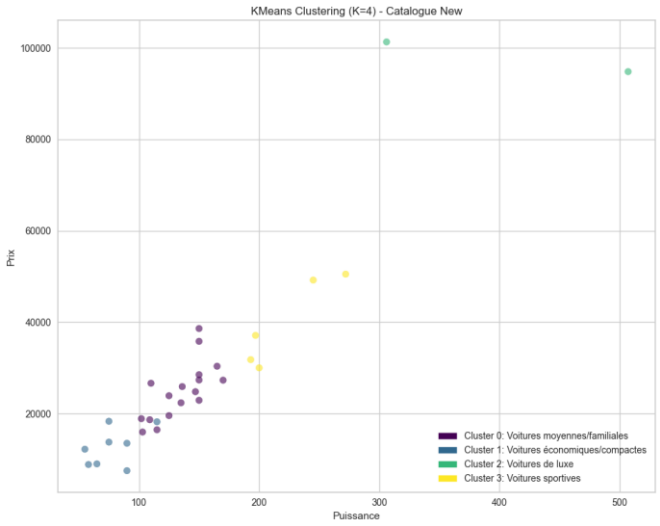


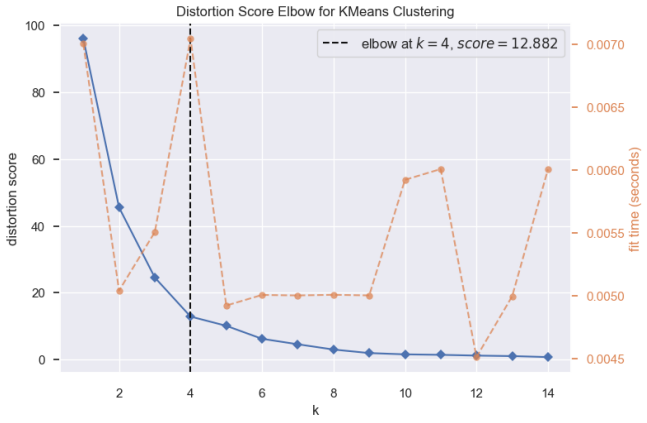
PARTIE 3 : IMMATRICULATIONS

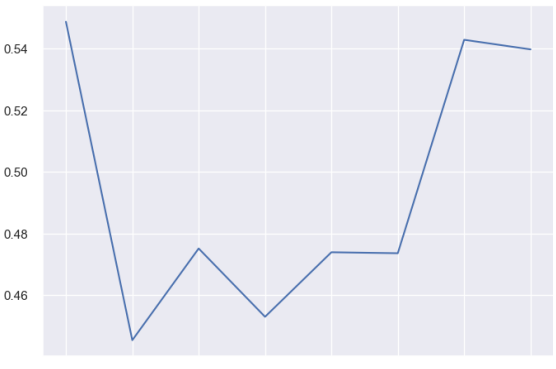
En ce qui concerne les immatriculations, nous avons observé que la distribution des données suit les mêmes règles que celles de la table du catalogue. De plus, la matrice de corrélation est presque identique.

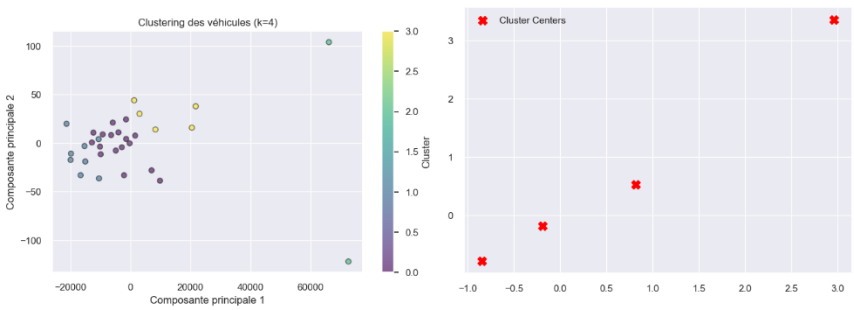
4. *Clustering des données du catalogue*

D'après le graphique de gauche présente **la méthode du coude** utilisée pour déterminer le nombre optimal de clusters (k) dans l'algorithme **KMeans**. Dans le graphique, on observe que la distorsion diminue rapidement pour des valeurs de k entre 2 et 4, ce qui indique une forte amélioration dans la qualité de la modélisation. Cependant, au-delà de **k=4**, la diminution de la distorsion devient beaucoup moins marquée, formant ce qu'on appelle **un coude**, donc **k=4** est un choix optimal en termes de compromis entre la complexité du modèle et la qualité des clusters. En combinant cela avec le graphique de droite, **le coefficient de silhouette** pour **k=4** reste acceptable, bien que k=2 montre un meilleur score de silhouette. Cependant, le compromis entre la réduction de la distorsion et la séparation des clusters suggère que **k=4** peut être le choix optimal dans cette situation.





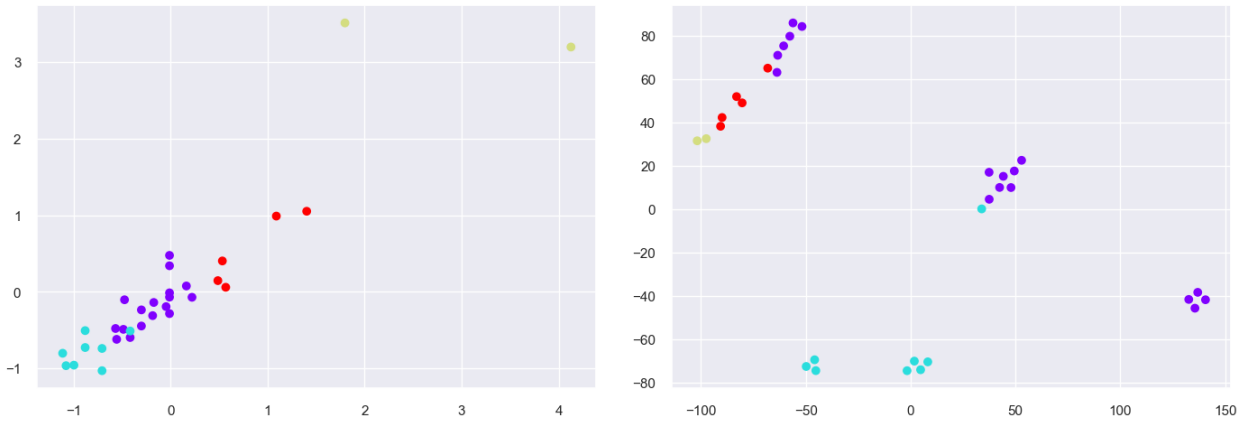




Ces graphiques présentent une analyse visuelle détaillée des résultats du clustering **KMeans** avec k=4, basé uniquement sur les caractéristiques principales : '**puissance**', '**longueur**' et '**prix**'. Les véhicules sont segmentés en quatre groupes distincts, selon les catégories définies :

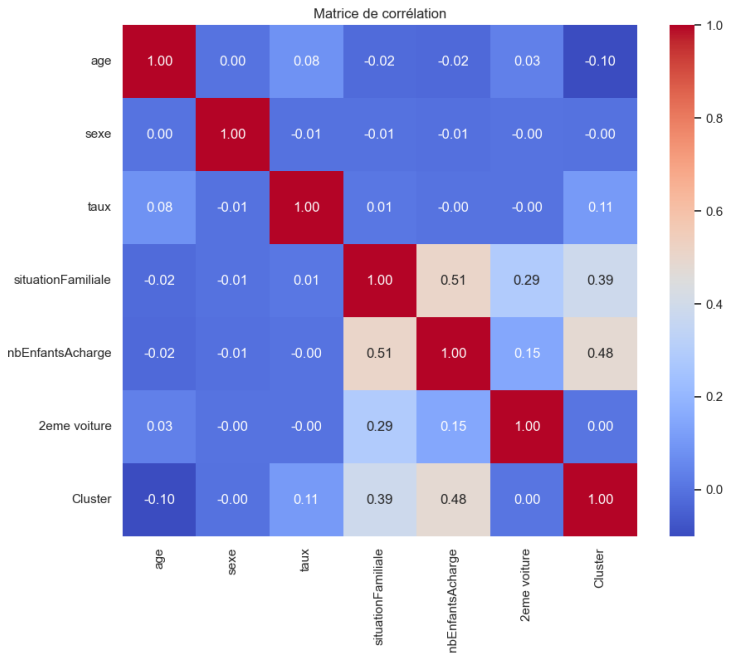
- **Cluster 0 :** Voitures moyennes/familiales,
- **Cluster 1 :** Voitures économiques/compactes,
- **Cluster 2 :** Voitures de luxe,
- **Cluster 3 :** Voitures sportives.

Ces visualisations permettent d'apprécier comment l'algorithme a identifié des patterns significatifs dans les données pour regrouper les véhicules selon leurs attributs spécifiques



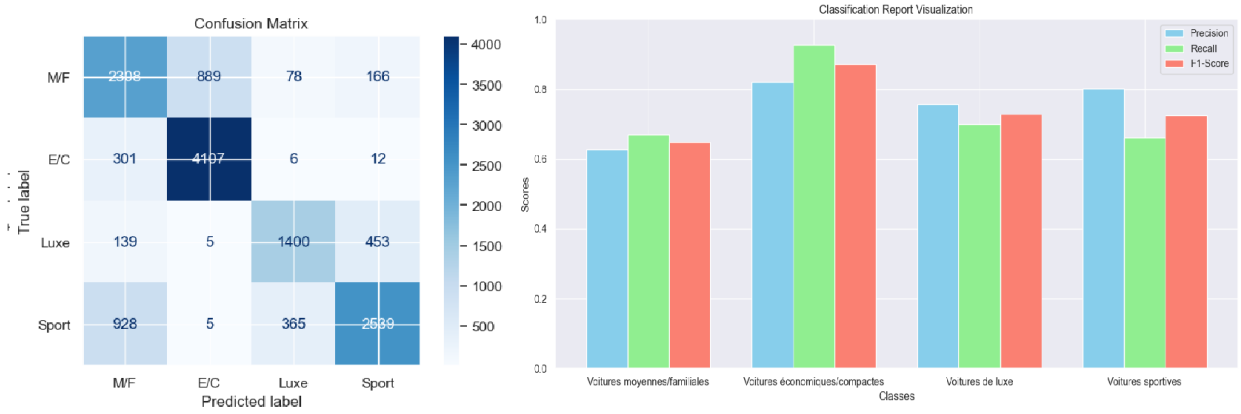
Les deux graphiques montrent que l'algorithme K-Means a bien segmenté les données en quatre clusters distincts. La projection PCA (à gauche) permet de visualiser comment les clusters se forment dans un espace réduit, tandis que la deuxième projection (à droite) montre une séparation encore plus nette des clusters. Ces résultats confirment la capacité de l'algorithme à regrouper les véhicules en fonction de leurs caractéristiques principales.

5. *Classification*



Pour l'analyse de la corrélation dans la nouvelle table résultant de la fusion des données **clients** et **immatriculations**, il est crucial de se concentrer sur les attributs des **clients** et les **clusters**, car ce sont ces informations qui serviront à prédire **la catégorie de véhicule**. Lors de cette **fusion**, les caractéristiques des clients sont associées aux données des véhicules, créant ainsi une base de données plus riche pour effectuer les suggestions. Cependant, l'attribut ***"sexe"*** a été écarté de l'analyse, car sa corrélation avec les clusters de véhicules est jugée **faible**. En se focalisant sur des attributs plus pertinents tels que l'***âge***, le ***taux***, la ***situation familiale*** et la présence d'une ***deuxième voiture***, nous garantissons une meilleure précision dans la prédiction de la catégorie de véhicule.

Le modèle choisi pour la prédiction est le **Random Forest**, qui a montré une excellente performance avec **Accuracy** de **75.57%**



6. Application

Dans cette étude, nous avons conçu une interface utilisateur intuitive pour faciliter le processus de recommandation de véhicules.

La figure ci-dessous illustre cette interface graphique conviviale, pensée pour permettre à l'utilisateur de renseigner efficacement ses informations personnelles. Elle intègre plusieurs champs spécifiques destinés à collecter les données essentielles pour proposer des recommandations personnalisées.

Prédiction de la Catégorie de Véhicule

Entrez les informations du client pour prédire la catégorie de véhicule.

Âge

30

Taux

10000

Situation Familiale

Marié(e)

Nombre d'Enfants à Charge

2

Possède une deuxième voiture

False

Prédire la Catégorie

La catégorie prédite du véhicule est : **Voitures sportives**

Illustration de Voitures sportives