ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

MIAGE Nice MBDS
UNIVERSITÉ CÔTE D'AZUR

MASTER 2 MIAGE MBDS DE L'UNIVERSITÉ COTE D'AZUR UCA

Projet Big Data Analytics : Analyse de la clientèle d'un concessionnaire Automobile pour la Recommandation de Modèles de Véhicules

Supervisé par :

- **Pr. MOLOPO MOKE Gabriel**
- **Pr. SEMONIAN Sergio**
- **Pr. PASQUIER Nicolas**

Réalisé par :

- **LOULIDI Fatimazahrae**
- **MOUNTASIR Loubna**
- **CHAOUKAT Taha**
- **HAMIDOUN Soufiane**

DU 01 JANVIER 2023 AU 15 AVRIL 2023

RESUMÉ

Le présent travail consiste en une analyse de la clientèle d'un concessionnaire automobile pour la recommandation des modèles de véhicules à travers l'utilisation de techniques de Big Data Analytics. L'objectif principal de cette étude est d'aider le concessionnaire à mieux comprendre les préférences de sa clientèle afin de recommander des modèles de voitures qui répondent à leurs besoins.

Le travail est de grande envergure, car il nécessite l'utilisation d'énormes quantités de données client et de modèles de voitures pour extraire des informations pertinentes. Pour ce faire, différentes méthodes d'analyse de données sont utilisées, notamment l'apprentissage automatique et l'analyse statistique.

Les principaux résultats de cette analyse sont des modèles de voitures recommandés pour chaque client, en fonction de ses préférences, de ses habitudes d'achat et de ses antécédents. Ces recommandations sont basées sur l'analyse de différentes variables, telles que le type de véhicule, la taille, la marque, le prix, etc.

Les conclusions de cette étude montrent que les techniques de Big Data Analytics peuvent être utilisées pour améliorer la satisfaction de la clientèle dans le secteur automobile en recommandant des modèles de voitures adaptés aux besoins de chaque client.

ABSTRACT

The purpose of this work is to perform an analysis of the customer base of an automotive dealership to recommend vehicle models using Big Data Analytics techniques. The main objective of this study is to assist the dealership in understanding their customer preferences better to recommend suitable vehicle models.

This work is of significant scope as it involves processing large amounts of customer data and vehicle model data to extract relevant information. Different data analysis methods, including machine learning and statistical analysis, are employed to achieve this.

The primary results of this analysis are recommended vehicle models for each customer based on their preferences, purchasing habits, and history. The recommendations are based on analyzing different variables such as vehicle type, size, brand, price, etc.

The conclusions of this study demonstrate that Big Data Analytics techniques can improve customer satisfaction in the automotive sector by recommending vehicle models tailored to each customer's needs.

Keywords: Big Data Analytics, data analysis, machine learning, vehicle model recommendation, customer analysis, automotive dealership.

LISTE DES FIGURES

Liste des tableaux

LISTE DES ACRONYMES

- **ETL: Extraction, Transformation and Loading**
- **BI: Business Intelligence**
- **OLAP: Online Analytical Processing**
- **KPI: Key Performance Indicator**
- **DWH: Data Warehouse**
- **AI: Artificial Intelligence**
- **ML: Machine Learning**
- **ERP: Enterprise Resource Planning**
- **IoT: Internet of Things**
- **Hadoop: Hadoop Distributed File System**
- **SQL: Structured Query Language**
- **NoSQL: Not only SQL**
- **JSON: JavaScript Object Notation**
- **API: Application Programming Interface**
- **RDD: Resilient Distributed Dataset**
- **KNN: K-nearest Neighbors**
- **NN: Neural Network**

Mots clés

Data Lake : Un Data Lake est un entrepôt de données qui permet de stocker une grande quantité de données brutes et non structurées. Il permet également aux utilisateurs de découvrir et d'analyser les données sans avoir besoin de les transformer au préalable.

Map Reduce : MapReduce est un modèle de programmation pour le traitement de données distribuées. Il permet de traiter des quantités massives de données en répartissant la charge de travail sur plusieurs ordinateurs en parallèle.

Data Analytics : L'analyse de données est le processus de collecte, de traitement et d'analyse de données afin d'en tirer des conclusions utiles. Les entreprises utilisent souvent l'analyse de données pour prendre des décisions plus éclairées.

Spark : Spark est un Framework open-source pour le traitement de données distribuées. Il est conçu pour être rapide et flexible et peut être utilisé pour diverses tâches de traitement de données, telles que l'analyse de données en temps réel et le traitement de flux de données.

Tableau : Tableau est un logiciel de visualisation de données qui permet aux utilisateurs de créer des tableaux de bord interactifs et des graphiques à partir de données. Il est souvent utilisé pour le business intelligence et la prise de décision.

Hive : Hive est une infrastructure de stockage de données distribuée pour Apache Hadoop. Il permet aux utilisateurs d'exécuter des requêtes SQL sur des données stockées dans Hadoop.

R : R est un langage de programmation open-source pour l'analyse de données et la visualisation. Il est souvent utilisé pour l'analyse statistique et la création de graphiques.

MongoDB : MongoDB est une base de données NoSQL open-source qui stocke des données sous forme de documents JSON. Il est conçu pour être rapide et flexible, et peut être utilisé pour diverses applications de base de données.

Hadoop : Hadoop est un Framework open-source pour le stockage et le traitement de données distribuées. Il permet aux utilisateurs de stocker et de traiter de grandes quantités de données en parallèle sur plusieurs ordinateurs.

Click Up : Click Up est un outil de gestion de projet en ligne qui permet aux équipes de collaborer et de gérer des projets en temps réel. Il propose des fonctionnalités telles que la gestion de tâches, les calendriers et les diagrammes de Gantt.

Data Mining : Le data mining est le processus d'exploration de données pour découvrir des modèles ou des relations utiles dans les données. Il est souvent utilisé pour le business intelligence et la prise de décision.

Data Visualisation : La visualisation de données est le processus de représentation visuelle de données afin de les rendre plus compréhensibles et utiles. Elle est souvent utilisée pour le business intelligence

Diagramme GANTT : Un diagramme de Gantt est un outil de gestion de projet qui montre les différentes tâches d'un projet, leur durée et leur ordonnancement dans le temps. Il permet de visualiser le déroulement du projet et de suivre son avancement.et la prise de décision.

SOMMAIRE

INTRODUCTION GÉNÉRALE

Avec la croissance rapide des données dans le monde, l'analyse de données est devenue une discipline incontournable dans de nombreux secteurs. Le secteur automobile ne fait pas exception à cette règle, car les concessionnaires automobiles ont maintenant accès à de vastes quantités de données clients, de modèles de véhicules et d'autres informations relatives au marché automobile. L'analyse de données est donc devenue un outil clé pour les concessionnaires automobiles pour comprendre les préférences et les habitudes d'achat de leurs clients, identifier les tendances du marché, et recommander les modèles de voitures les plus adaptés à leurs clients. Dans ce contexte, ce rapport de projet présente une analyse de la clientèle d'un concessionnaire automobile pour recommander des modèles de véhicules à l'aide de techniques de Big Data Analytics.

Les techniques de Big Data Analytics sont devenues un outil de plus en plus important dans le secteur automobile, permettant aux concessionnaires automobiles de collecter, stocker et analyser des volumes de données de plus en plus importants. Grâce à des techniques de traitement des données telles que l'apprentissage automatique et la modélisation statistique, les concessionnaires automobiles peuvent identifier des modèles dans les données, ce qui peut aider à comprendre les préférences et les habitudes d'achat de leurs clients.

Dans ce rapport de projet, nous présentons une analyse de la clientèle d'un concessionnaire automobile en utilisant des techniques de Big Data Analytics pour recommander les modèles de véhicules les plus adaptés à chaque client. Nous avons recueilli des données clients, de modèles de véhicules, d’immatriculations, de marketing aussi, effectué une analyse approfondie de ces données, et utilisé des techniques de modélisation pour recommander les modèles de véhicules les plus pertinents pour chaque client. Nous discuterons des résultats de notre analyse et des conclusions qui en découlent.

I. Présentation du projet :

1. Objectifs du projet :

Cernant le traitement de données massives, l'analyse de données et l'apprentissage automatique. L'objectif de ce projet est d'analyser les données clients et les informations relatives aux ventes pour recommander les modèles de véhicules les plus adaptés à chaque client. Plus spécifiquement, les objectifs sont les suivants :

- Identifier les caractéristiques principales des clients, telles que leur âge, leur sexe, leur profession et leur lieu de résidence, qui peuvent influencer leur choix de véhicule.
- Analyser les données de vente pour identifier les modèles de véhicules les plus populaires et les plus vendus.
- Utiliser des techniques d'apprentissage automatique pour prédire les modèles de véhicules qui sont les plus susceptibles d'intéresser les clients en fonction de leurs caractéristiques et de leurs comportements d'achat.
- Proposer un système qui permet à un vendeur de recommander rapidement et facilement un véhicule à un client en fonction de ses caractéristiques et de ses besoins.

2. Les livrables attendus pour ce projet :

- Un rapport détaillé décrivant les différentes étapes du projet, les méthodes et techniques utilisées, ainsi que les résultats obtenus.
- Un prototype de système de recommandation de véhicules qui peut être utilisé par les vendeurs de la concession.
- Une Vidéo explicatifs
- Des visualisations claires et concises des données d'analyse pour faciliter la prise de décision.

3. Contexte du projet :

En résumé, ce projet de Big Data Analytics vise à aider un concessionnaire automobile à mieux comprendre les besoins de ses clients et à recommander les modèles de véhicules les plus adaptés en fonction de leurs caractéristiques et de leurs comportements d'achat. Le projet nécessitera des compétences en traitement de données, en analyse de données et en apprentissage automatique. Les livrables attendus incluent un rapport détaillé, un prototype de système de recommandation de véhicules et des visualisations claires des données d'analyse.

II. Répartition du travail en membre du groupe :

1. Tableau de répartitions de taches :

Table 1 : Répartition des taches par membres de groupe

Taches	Faite par :
Règlement de l'environnement du projet (Machine Virtuel avec Vagrant)	LOULIDI Fatimazahrae
Chargement des données dans l'environnement Vagrant	CHAOUKAT Taha
Chargement des données depuis L'environnent aux différentes sources de données	HAMIDOUN Soufiane
Création des tables EXTERNES et INTERNES dans HIVE	MOUNTASIR Loubna
Importation des données des sources de données vers les tables crées dans HIVE	LOULIDI Fatimazahrae
Règlement de l'environnement R Shell	HAMIDOUN Soufiane
Création des scripts de nettoyage de données	LOULIDI Fatimazahrae
Connection de R avec HIVE et importation de données des tables dans R	LOULIDI Fatimazahrae
Application des scripts de nettoyage sur les données	CHAOUKAT Taha
Identification des catégories de véhicules	MOUNTASIR Loubna
Application des catégories de véhicules définies aux données des immatriculations	MOUNTASIR Loubna
Fusion des données Clients et immatriculations	LOULIDI Fatimazahrae
Création des modèles de classification supervisés pour la prédiction de la catégorie de véhicules	HAMIDOUN Soufiane
Teste des algorithmes d'apprentissage	LOULIDI Fatimazahrae
Évaluation et la comparaison des performances de chaque configuration algorithmique	
Application du modèle de prédiction aux données Marketing	CHAOUKAT Taha
Rapport de Projet	LOULIDI Fatimazahrae
Création du dossier script de construction des lacs de données	LOULIDI Fatimazahrae
Création du dossier script de la partie HADOOP MAP REDUCE	MOUNTASIR Loubna
Création du dossier script de visualisation de données	HAMIDOUN Soufiane
Création du dossier script d'analyse de données	LOULIDI Fatimazahrae

2. Diagramme de GANTT :

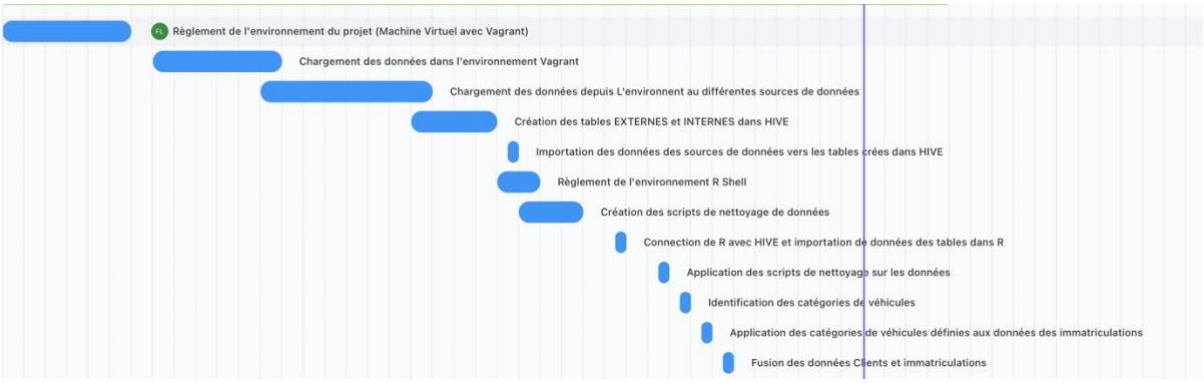


Figure 1: Gantt partie 1

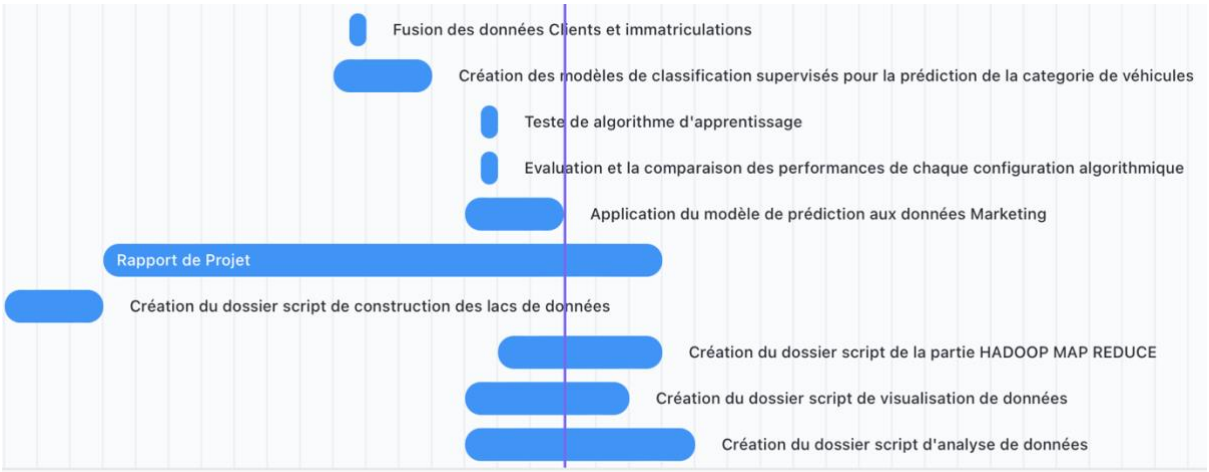


Figure 2: GANTT partie 2

III. Architecture du data Lake :

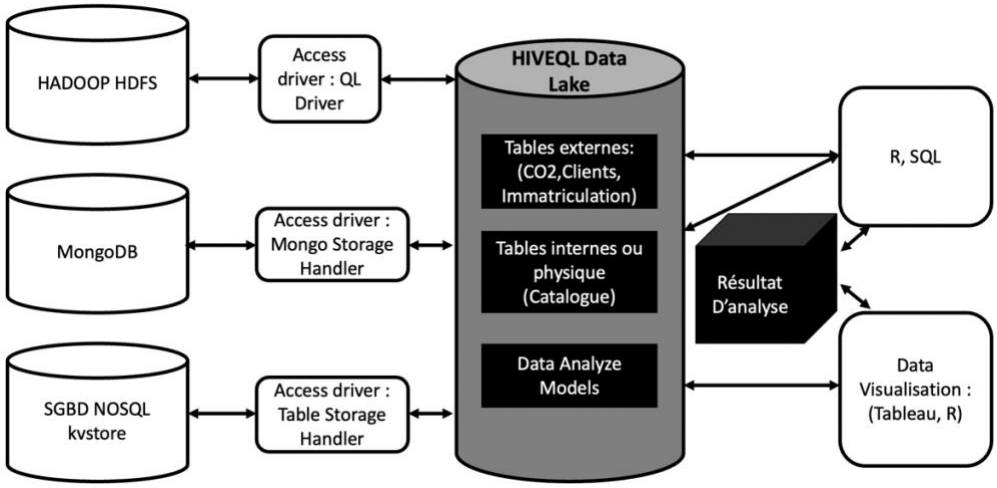


Figure 3 : Architecture du Projet

IV. Construction du data Lake par étape :

- **Chargement des données dans l'environnement Vagrant :**
    - Importation des données des fichiers .CSV dans la machine Virtuel /home/Vagrant a l'aide de la commande « scp »

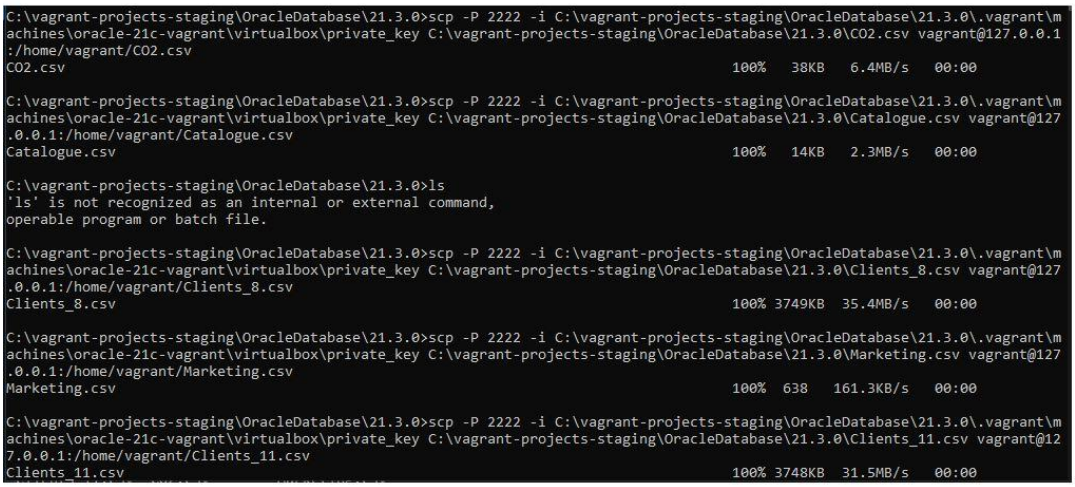


Figure 4: Importation des fichiers du local vers vagrant

- **Chargement des données depuis L'environnent aux différentes sources de données**
    - Création d'une base de données MongoDB

    - Création des collections ou on va stocker les données

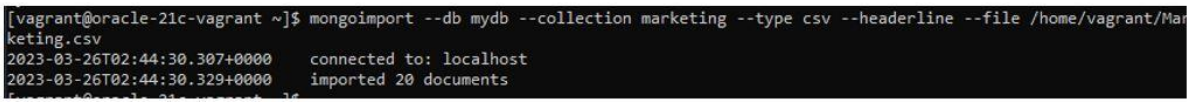


Figure 5: Creation du BDD et importation du fichier Marketing.csv

    - Chargement des fichiers "Clients.csv" dans la base créée dans MONGO

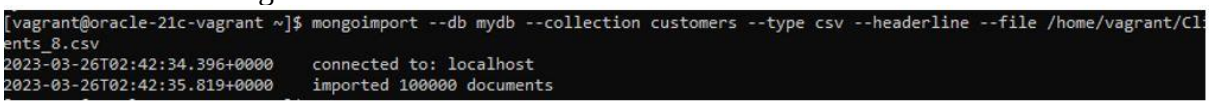


Figure 6: Creation de la collection et importation des fihciers Clients

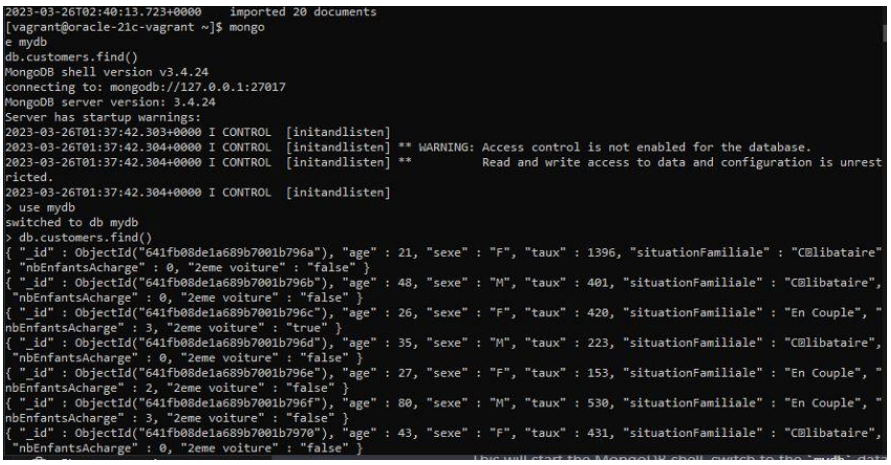


Figure 7 : collection customers dans MONGO

    - Paramétrage de l'environnement HDFS

```
[vagrant@oracle-21c-vagrant ~]$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as vagrant in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [oracle-21c-vagrant]
Starting resourcemanager
Starting nodemanagers
[vagrant@oracle-21c-vagrant ~]$ jps
8560 Jps
8211 NodeManager
7590 DataNode
7447 NameNode
8073 ResourceManager
7807 SecondaryNameNode
```

Figure 8: Environnment HDFS

- Chargement du fichier "CO2.csv" dans HADOOP

```
[vagrant@oracle-21c-vagrant ~]$ hdfs dfs -put /home/vagrant/CO2.csv /user/hadoop/input
put: `/user/hadoop/input': No such file or directory: `hdfs://localhost:9000/user/hadoop/input'
[vagrant@oracle-21c-vagrant ~]$ hdfs dfs -mkdir -p /user/hadoop/input
CO2.csv /user/hadoop/input
[vagrant@oracle-21c-vagrant ~]$ hdfs dfs -put /home/vagrant/CO2.csv /user/hadoop/input
```

Figure 9 : Chargement du fichier co2.csv

- Création de la table Immatriculation dans kvstore

```
sql-> CREATE TABLE immatriculations(immatriculation string,marque string,nom string,puissance INTEGER,longueur string,nb
Places INTEGER,nbPortes INTEGER,couleur string,occasion BOOLEAN,prix INTEGER,PRIMARY KEY (immatriculation)
  -> )
  -> ;
Statement completed successfully
sql-> show tables
tables
  SYS$IndexStatsLease
  SYS$MRTableAgentStat
  SYS$MRTableInfo
  SYS$MRTableInitCheckpoint
  SYS$PartitionStatsLease
  SYS$SGAttributesTable
  SYS$StreamRequest
  SYS$StreamResponse
  SYS$TableStatsIndex
  SYS$TableStatsPartition
  immatriculations
```

Figure 10 : Création de la table "immatriculation" dans kvstore

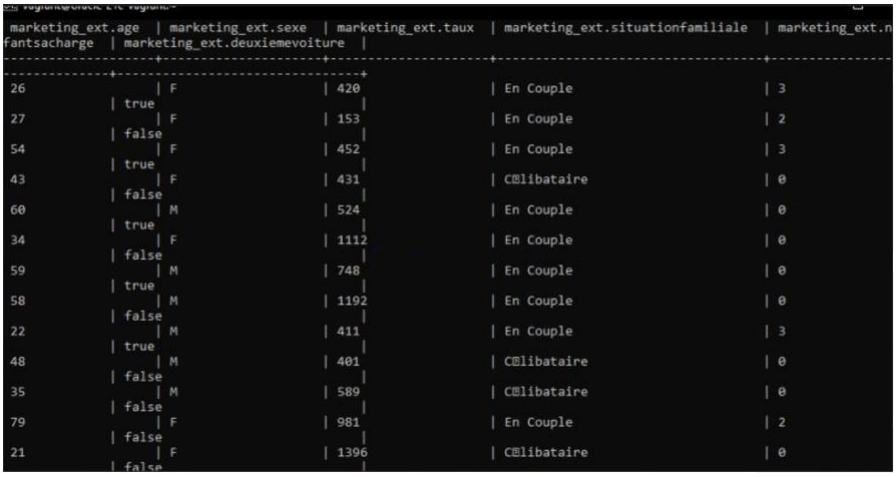
- Chargement du fichier "immatriculations.csv" dans la table immatriculation dans Kvstore

```
sql-> import -table immatriculations -file '/home/vagrant/immatriculations.csv' csv
File not found: /home/vagrant/immatriculations.csv
sql-> import -table immatriculations -file '/home/vagrant/immatriculations.csv' csv
File not found: /home/vagrant/immatriculations.csv
sql-> exit

[vagrant@oracle-21c-vagrant ~]$ java -Xmx256m -Xms256m -jar $KVHOME/lib/sql.jar -helper-hosts localhost:5000 -store kvstc
sql-> import -table immatriculations -file '/home/vagrant/immatriculations.csv' csv
File not found: /home/vagrant/immatriculations.csv
sql-> import -table immatriculations -file '/home/vagrant/Immatriculations.csv' csv
Error handling command import -table immatriculations -file '/home/vagrant/Immatriculations.csv' csv: Failed to import JS
ow at line 1 of file, /home/vagrant/Immatriculations.csv: For input string: "puissance"
sql-> import -table immatriculations -file '/home/vagrant/Immatriculationsave.csv' csv
```

Figure 11 : chargement des données d'immatriculation dans la table

- **Création des tables EXTERNES et INTERNES dans HIVE**
  - Création des tables Externes « Marketing_ext »



marketing_ext.age	marketing_ext.sexe	marketing_ext.taux	marketing_ext.situationfamiliale	marketing_ext.nfantsacharge	marketing_ext.deuxiemevoiture
26	F	420	En Couple	3	true
27	F	153	En Couple	2	false
54	F	452	En Couple	3	true
43	F	431	Célibataire	0	false
60	M	524	En Couple	0	true
34	F	1112	En Couple	0	false
59	M	748	En Couple	0	true
58	M	1192	En Couple	0	false
22	M	411	En Couple	3	true
48	M	401	Célibataire	0	false
35	M	589	Célibataire	0	false
79	F	981	En Couple	2	false
21	F	1396	Célibataire	0	false

Figure 12 : Table Marketing_ext dans HIVE

- Création des tables Externes « Immatriculation_ext »

```
0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE immatriculation_ext (
. . . . . .>   immatriculation string,
. . . . . .>   marque string,
. . . . . .>   nom string,
. . . . . .>   puissance int,
. . . . . .>   longueur string,
. . . . . .>   nbPlaces int,
. . . . . .>   nbPortes int,
. . . . . .>   couleur string,
. . . . . .>   occasion boolean,
. . . . . .>   prix int
. . . . . .> )
. . . . . .> STORED BY 'oracle.kv.hadoop.hive.table.TableStorageHandler'
. . . . . .> TBLPROPERTIES (
. . . . . .>     "oracle.kv.kvstore" = "kvstore",
. . . . . .>     "oracle.kv.hosts" = "localhost:5000",
. . . . . .>     "oracle.kv.tableName" = "immatriculations"
. . . . . .>
. . . . . .> );
```

Figure 13 : création de la table Externe immatriculation_ext

- Création des tables Externes « CO2_ext »

```
0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE co2_ext (
. . . . . .>   marque_modele STRING,
. . . . . .>   bonus_malus STRING,
. . . . . .>   rejets_co2 FLOAT,
. . . . . .>   cout_energie FLOAT
. . . . . .> )
. . . . . .> ROW FORMAT DELIMITED
. . . . . .> FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE
. . . . . .> LOCATION '/user/hadoop/input/';
23/03/30 16:45:01 INFO ql.Driver: Compiling command(queryId=vagrant_2023033016450
-9855-436b-93ed-7f7e319d5ea3): CREATE EXTERNAL TABLE co2_ext (
marque_modele STRING,
bonus_malus STRING,
rejets_co2 FLOAT,
cout_energie FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/user/hadoop/input/'
```

Figure 14 : Création de la table EXTERNE co2_ext

- Création des tables Externes « customers_ext » :

```
0: jdbc:hive2://localhost:10000> CREATE EXTERNAL TABLE custumers_ext (
. . . . . .>     id STRING,
. . . . . .>     age INT,
. . . . . .>     sexe STRING,
. . . . . .>     taux INT,
. . . . . .>     situationFamiliale STRING,
. . . . . .>     nbEnfantsAcharge INT,
. . . . . .>     deuxiemeVoiture BOOLEAN,
. . . . . .>     immatriculation STRING
. . . . . .> )
. . . . . .> STORED BY 'com.mongodb.hadoop.hive.MongoStorageHandler'
. . . . . .> TBLPROPERTIES (
. . . . . .>     'mongo.uri'='mongodb://localhost:27017/mydb.custumers',
. . . . . .>     'mongo.input.query'='{ "_id" : ObjectId("641fb292e1a689b7001e87e5") }'
. . . . . .> );
```

Figure 15 : création de la table customers_ext

- Création de la table internes « Catalogue »

```
0: jdbc:hive2://localhost:10000> CREATE TABLE Catalogue (
. . . . . .>   marque string,
. . . . . .>   nom string,
. . . . . .>   puissance int,
. . . . . .>   longueur string,
. . . . . .>   nbPlaces int,
. . . . . .>   nbPortes int,
. . . . . .>   couleur string,
. . . . . .>   occasion string,
. . . . . .>   prix double
. . . . . .> )
. . . . . .> ROW FORMAT DELIMITED
. . . . . .> FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE;
```

Figure 16: création de la table INTERNE catalogue

- **Importation des données des sources de données vers les tables crées dans HIVE**
  - Chargement des données depuis la collection "customers" dans MONGO vers la tables externe "customers_ext" dans HIVE.
  - Chargement des données depuis "CO2.csv" HADOOP dans "co2_ext" dans HIVE.
  - Chargement des données depuis le LOCAL vers la table Interne "Catalogue" dans HIVE.
  - Chargement des données d'immatriculation depuis KVstore vers la table "immatriculation_ext" dans HIVE.
  - Chargement des données marketing depuis KVstore vers la table "marketing_ext" dans HIVE.

V. Hadoop Map Reduce :

1. Étapes de résolutions du travaille MAP REDUCE :

- Importation des données du fichier CO2.csv dans Hadoop / Spark

```
schema = StructType([
StructField("Brand_Model", StringType(), True),
StructField("CO2_Emissions_g_km", DoubleType(), True),
StructField("Energy_Cost", DoubleType(), True),
StructField("Bonus_Malus", StringType(), True)])
```
✓ 0.3s Python

```
co2_df = spark.read.csv("C:/Users/hp/vagrant-projects/OracleDatabase/21.3.0/CO2.csv", header=True,schema=schema)
```
✓ 9.6s Python

```
from pyspark.sql.functions import expr

co2_df = co2_df.withColumn("Brand", expr("substring(Brand_Model, 1, instr(Brand_Model, ' ')-1)"))
co2_df = co2_df.withColumn("Model", expr("substring(Brand_Model, instr(Brand_Model, ' ')+1, length(Brand_Model))"))
```
✓ 2.6s Python

Figure 17 : Importation des données CO2 et rajout des colonnes nécessaires

- Mappage pour diviser les données de CO2.csv en paires clé-valeur, où la clé est la marque de voiture et la valeur est une structure de données contenant les informations de CO2, coût d'énergie et Bonus/Malus pour cette marque.

```
co2_df = co2_df.drop("Brand_Model")
```
✓ 0.8s Python

```
# Map CO2 DataFrame to key-value pairs where key is "Brand" and value is a tuple containing CO2, Energy Cost, and Bonus/
co2_kv = co2_df.rdd.map(lambda row: (row["Brand"], (row["CO2_Emissions_g_km"], row["Energy_Cost"], row["Bonus_Malus"])))
```
✓ 8.7s Python

Figure 18 : Mappage en clé et valeurs

- La réduction en groupant toutes les données de la même marque de voiture et en calculant les moyennes pour les valeurs manquantes. La sortie de cette étape sera une table qui contient les informations moyennes de CO2, coût d'énergie et Bonus/Malus pour chaque marque de voiture.

```
# Reduce CO2 key-value pairs by averaging values for each Brand
co2_agg = co2_kv.reduceByKey(lambda a, b: (
(a[0] or 0.0) + (b[0] or 0.0),
(a[1] or 0.0) + (b[1] or 0.0),
a[2] if a[2] else b[2]))
co2_agg = co2_agg.mapValues(lambda v: (
round(v[0]/max(1, sum([1 for x in v if x])), 2),
round(v[1]/max(1, sum([1 for x in v if x])), 2),
v[2]))
```
✓ 1.2s

Figure 19 : Reduction CO2 des couple clé et valeurs en intégrant les valeurs de chaque marque

- Jointure entre cette table et la table « catalogue » du concessionnaire automobiles en utilisant la marque de voiture comme clé.
- Écriture des résultats de la jointure dans un fichier de sortie, qui peut être importé dans la base de données du concessionnaire automobiles.

VI. Visualisation de données avec des outils de DataViz (si concerné) :

1. Pie Chart de répartition des clients par catégories de voitures

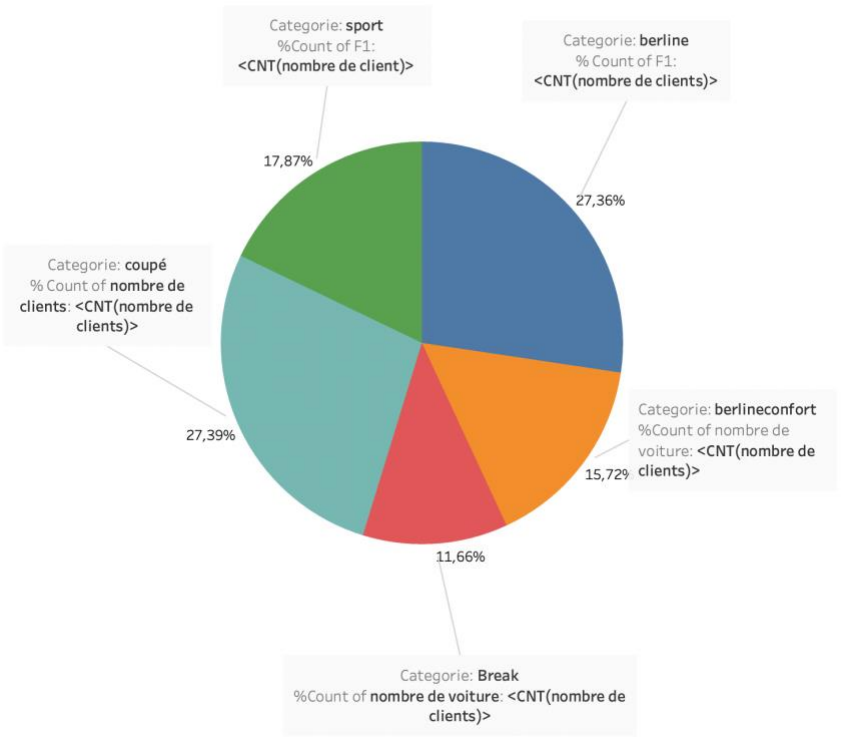


Figure 20 : Pie Chart de répartition des clients par catégories de voitures

2. Bar Chart de répartition de chaque sexe par sa situation familiale et catégories de voitures qui peut choisir

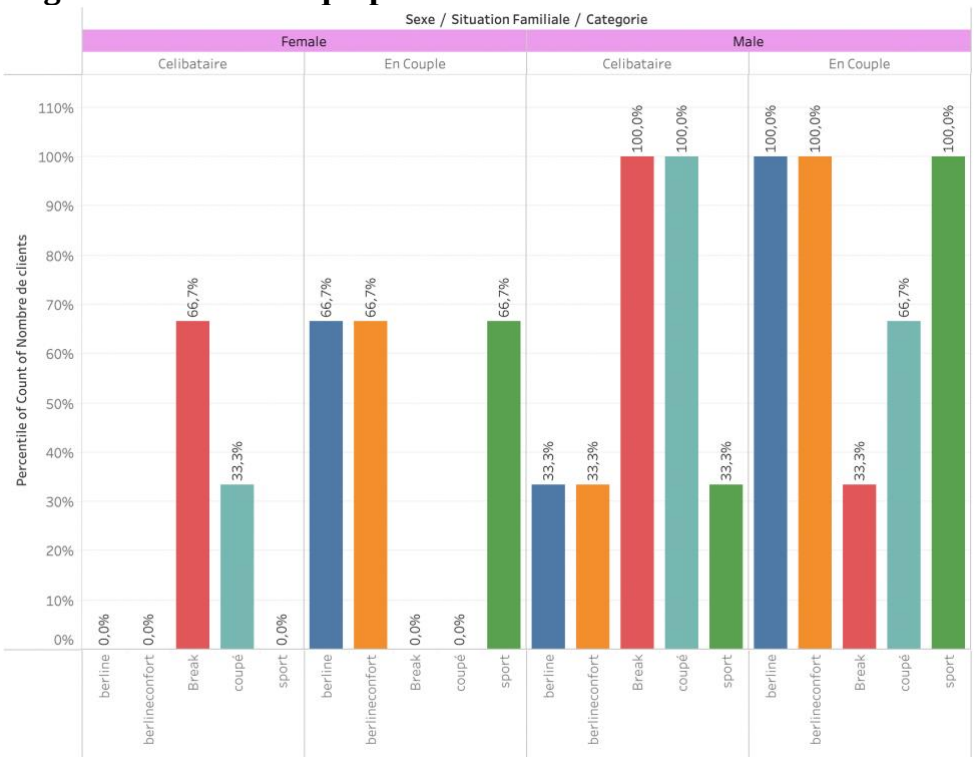


Figure 21 : Bar Chart de répartition de chaque sexe par sa situation familiale et catégories de voitures qui peut choisir

3. Bar chart : client avec sa situation familiale par la marque de voiture qui va acheter

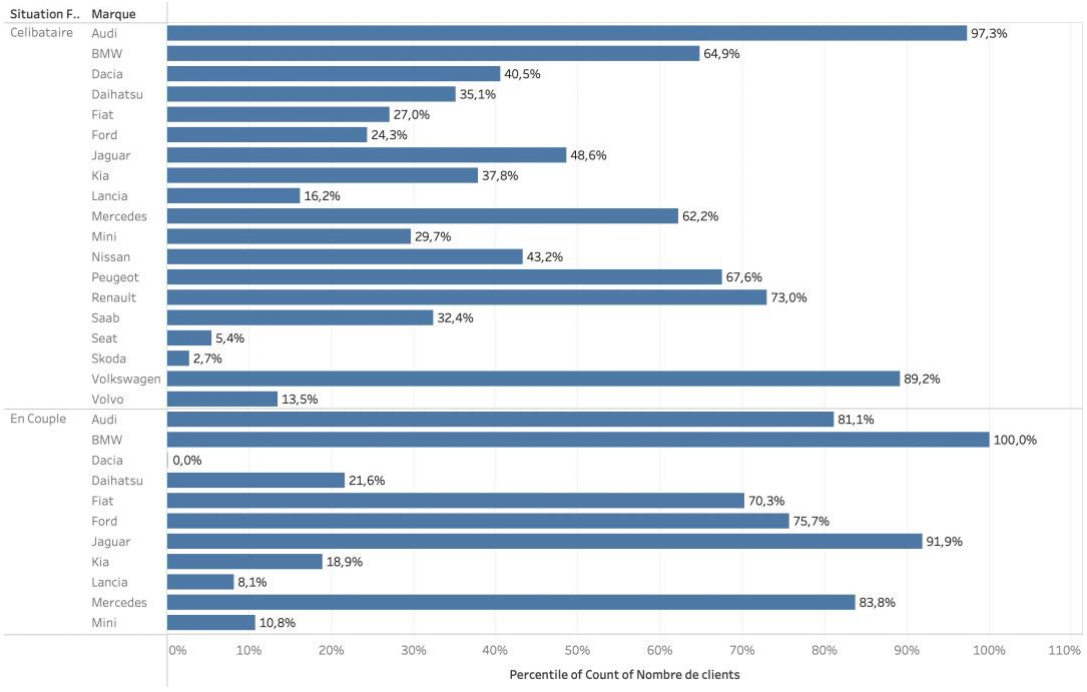


Figure 22 : Bar chart : client avec sa situation familiale par la marque de voiture qui va acheter

4. Répartition de nombre de client par situation familiale et 2eme voiture

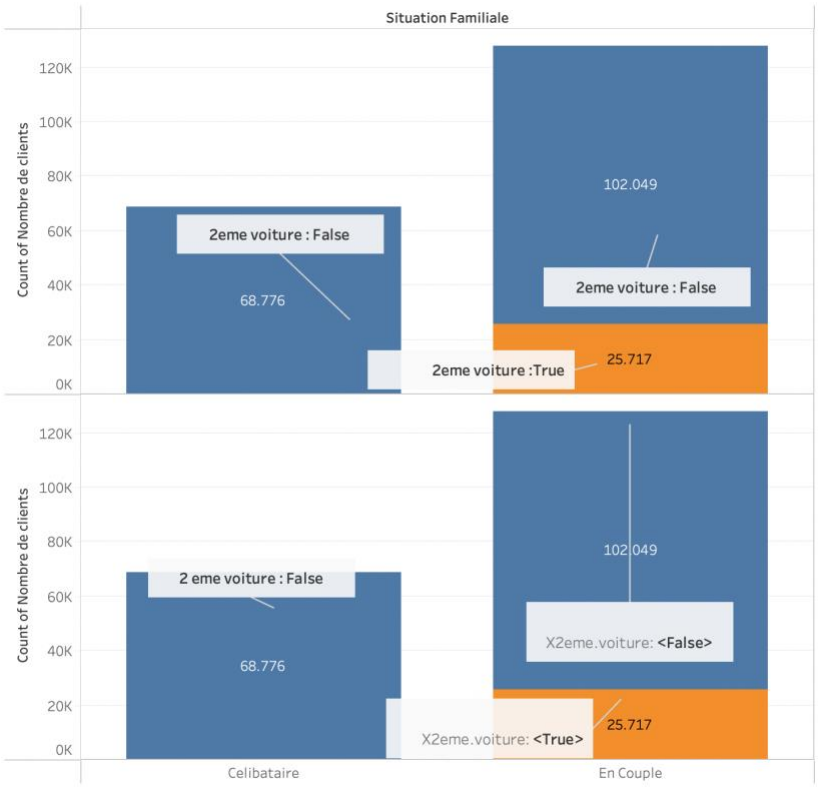


Figure 23 : répartition de nombre de client par situation familiale et 2eme voiture

VII. Analyse de données avec des outils de machine Learning (R, …) :

1. Processus de travailles :

a) Analyse exploratoire des données :

Dans cette étape, on a exploré les données pour identifier d'éventuels problèmes tels que des valeurs incohérentes ou des valeurs manquantes(N/A) et la conversion des types de colonnes de données (numeric, factor, logical). On a également cherché à découvrir d'éventuelles propriétés de l'espace des données, telles que des valeurs doublons, des variables liées ou des variables d'importance particulière ou inutiles et cela a été réalisé dans Rshell Workspace avec le langage R, vous trouverez dans le dossier « scripts et programmes d'analyse de données » les scripts de nettoyage utilisé.

```
> convert_factor <- function(df, col_name) {
+   if (is.factor(df[[col_name]])) {
+     cat(paste(col_name, "is already a factor. No conversion needed.\n"))
+   } else {
+     if (is.numeric(df[[col_name]]) || is.character(df[[col_name]])) {
+       if (nrow(df) == 0) {
+         cat(paste(col_name, "has no data. No conversion needed.\n"))
+       } else {
+         df[[col_name]] <- as.factor(df[[col_name]])
+         cat(paste("Converted", col_name, "to factor.\n"))
+       }
+     } else {
+       cat(paste(col_name, "is not a numeric or character column. No conversion made.\n"))
+     }
+   }
+   return(df)
+ }
>
> catalogue <- convert_factor(catalogue, "longueur")
longueur is not a numeric or character column. No conversion made.
> catalogue <- convert_factor(catalogue, "catalogue.longueur")
Converted catalogue.longueur to factor.
```

Figure 24 : Fonction de Conversion en factor

```
> immatriculation_ext <- convert_logical(immatriculation_ext, "immatriculation_ext.occasion")
Converted immatriculation_ext.occasion to logical.
> summary(immatriculation_ext)
 immatriculation_ext.immatriculation immatriculation_ext.marque
 Length:115109                       Length:115109
 Class :character                    Class :character
 Mode  :character                    Mode  :character



 immatriculation_ext.nom immatriculation_ext.puissance
 Length:115109           Min.   : 55.0
 Class :character        1st Qu.: 75.0
 Mode  :character        Median :150.0
                         Mean   :199.5
                         3rd Qu.:245.0
                         Max.   :507.0
 immatriculation_ext.longueur immatriculation_ext.nbplaces
 courte     :31768            Min.   :5
 longue     :31091            1st Qu.:5
 moyenne    :13280            Median :5
 tr□s longue:38970            Mean   :5
                              3rd Qu.:5
                              Max.   :5
 immatriculation_ext.nbportes immatriculation_ext.couleur
 Min.   :3.000                blanc:22840
 1st Qu.:5.000                bleu :23127
 Median :5.000                gris :22973
 Mean   :4.868                noir :22957
 3rd Qu.:5.000                rouge:23212
 Max.   :5.000
 immatriculation_ext.occasion immatriculation_ext.prix
 Mode :logical                Min.   :  7500
 FALSE:79078                  1st Qu.: 18310
 TRUE :36031                  Median : 25970
                              Mean   : 35865
                              3rd Qu.: 49200
                              Max.   :101300
```

Figure 25 : Appelle Fonction de conversion en logical

Figure 26 : Appelle de fonction numérique

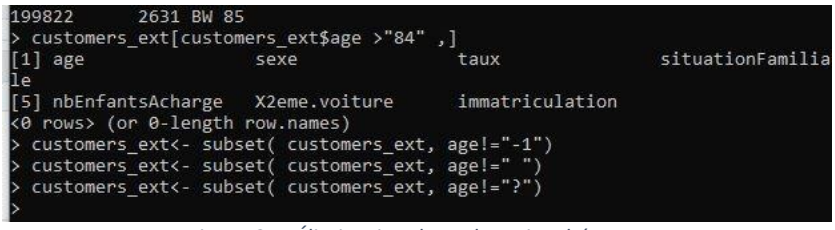


Figure 27 : Élimination des valeurs incohérentes

a) Identification des categories de véhicules :

On a utilisé les informations du catalogue pour identifier des catégories de véhicules en fonction de leur taille, puissance, prix, etc. Ces catégories ont été définies pour répondre aux différents besoins des clients ils sont comme suite :

- ***Coupé***
- ***Berline***
- ***Break***
- ***Berline confort***
- ***Sport***

b) Application des categories de véhicules aux données d'immatriculations :

Nous avons utilisé le modèle définissant les catégories de véhicules généré précédemment pour attribuer à chaque véhicule vendu cette année la categories qui lui correspond.

- ***Interprétation du code utilisé***

Le code qu'on a utilisé consiste à assigner les catégories citées, au data frame "immatriculations" en fonction de certaines conditions. Les catégories sont assignées à la variable "catégorie" de l'ensemble de données "immatriculations".

On a choisi de coder cela sous forme d'une structure « ifelse » imbriquée pour assigner les catégories en fonction de la longueur du véhicule ("courte", "treslongue" ou "moyenne"), du nombre de places (7 ou autre) et de la puissance du moteur (inférieure à 200 ou supérieure à 200).

Ensuite on a ajouté une catégorie supplémentaire pour les véhicules ayant une longueur "treslongue" et une puissance comprise entre 190 et 300, qui sont classés comme "berline confort". Les véhicules de longueur "moyenne" sont classés comme "Break" et tous les autres véhicules sont classés comme "berline". Les catégories sont assignées en utilisant à nouveau une structure « ifelse » imbriquée.

c) Fusion des données clients et immatriculations :

Nous avons fusionné les données clients et immatriculations pour obtenir sur une même ligne l'ensemble des informations sur le client et sur le véhicule qu'il a acheté. Cet ensemble de données a été utilisé pour l'apprentissage de la catégorie de véhicules la plus adaptée à un client selon ses caractéristiques.

- ***Interprétation du code utilisé :***

Jointure (merge) entre deux tables nommées "client" et "immatriculations", basée sur la colonne "immatriculation".
L'application de la fonction "summary()" qui fournit un résumé des statistiques descriptives pour chaque variable de l'objet etudié dans notre cas les données "customers"..
Suppression des colonnes inutiles de l'objet "customers" à l'aide de la fonction "subset()", à savoir la colonne "immatriculation" et la colonne "nbPlaces".
La fonction "names()" pour afficher les noms des colonnes restantes de l'objet "customers".

d) Création d'un modèle de classification supervisée et application du modèle de prédiction aux données marketing :

Dans cette étape, nous avons créé un modèle de classification supervisée à partir du résultat de la fusion précédente. Nous avons testé différentes approches et algorithmes tels que les arbres de décision, Naives bayes, les random forests, les réseaux de neurones, K-nearest Neighbors KNN, k-means, R-part, C5.0. Pour chaque algorithme, nous avons testé plusieurs paramétrages afin d'obtenir un classifieur performant. Nous avons évalué et comparé ainsi les performances de chaque configuration algorithmique en utilisant des matrices de confusion et des mesures d'évaluation.

- ***Interprétation du code :***

On a divisé l'ensemble de données "customers" en un ensemble d'entraînement de 70% et un ensemble de test de 30% en utilisant la fonction "createDataPartition".
Ensuite, on a supprimé certaines variables inutiles des ensembles de données d'entraînement et de test en utilisant la fonction "subset".

Code Naives Bayes :
Installation du package « caret »et chargement du librairie « pryr »

Vérification s'il y a des valeurs manquantes dans les ensembles de données d'entraînement et de test en utilisant les fonctions is.na() et sum().

Ensuite, on applique la fonction naive_bayes() pour entraîner le classificateur naive Bayes en utilisant les données d'entraînement. L'argument laplace = 1 est utilisé pour appliquer la

correction de Laplace avec un facteur de lissage de 1, afin d'éviter d'avoir des probabilités nulles pour une valeur de caractéristique donnée étant donné une certaine classe.

Ensuite, on vérifie les niveaux des variables catégorielles dans les données d'entraînement et de test à l'aide des fonctions levels() et print().

Les colonnes communes entre les ensembles de données d'entraînement et de test sont ensuite identifiées à l'aide de la fonction intersect(), et seules ces colonnes sont conservées dans l'ensemble de données de test.

Par la suite on utilise la fonction predict() pour prédire la classe des données de test en utilisant le modèle entraîné. La variable nb_class contient les prédictions de classe.

La fonction confusionMatrix() est ensuite utilisée pour calculer la matrice de confusion et l'exactitude (accuracy) du modèle pour les données de test. Les valeurs de précision, de rappel et de score F1 pour chaque classe sont également calculées.

En fin, on assigne les prédictions de classe aux données de test et aux données de marketing, puis impriment les résultats de la précision, du rappel et du score F1 pour chaque classe, ainsi que l'exactitude globale du modèle.

Code C5.0 :
Installations du package C5.0 et chargement de la librairie dans R.

Pour construire des arbres de décision et des modèles basés sur des règles. Il résume ensuite le contenu de deux data frames nommés "training_data" et "testing_data".
Conversion également certaines variables des data frames "testing_data" en facteurs car C5.0 (l'algorithme utilisé dans le package C50) peut gérer à la fois des variables facteurs et numériques. En particulier, les variables "categorie", "X2eme.voiture" et "situationFamiliale" sont converties en facteurs dans "training_data", tandis que la variable "X2eme.voiture" est convertie en facteur dans "testing_data". Ensuit on peut générer le pourcentage d'exactitude de l'algorithme ainsi que l'arbre de décision.

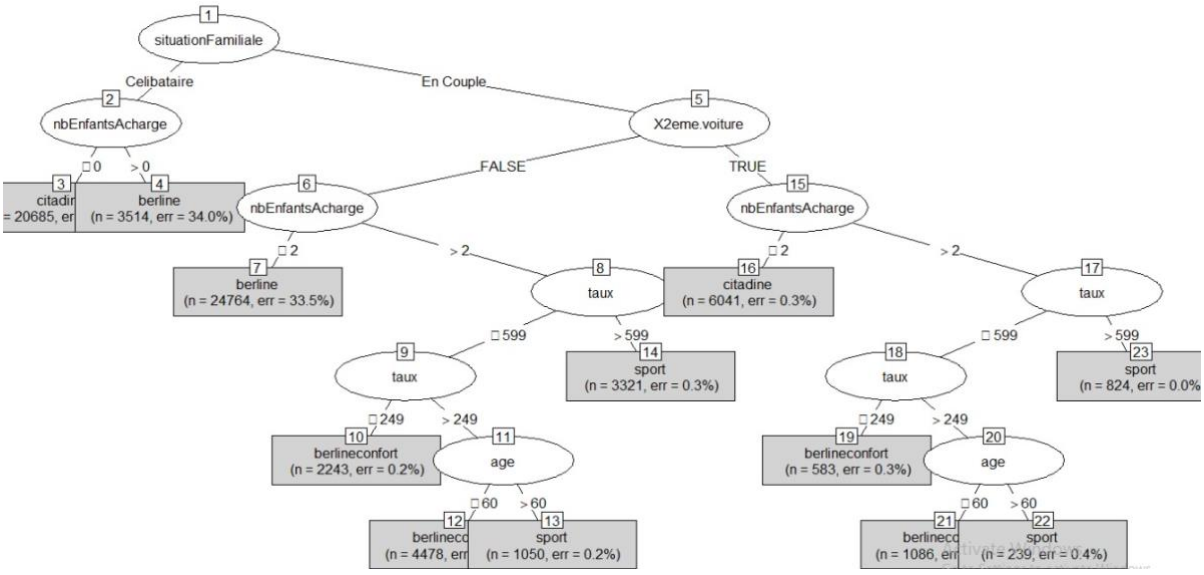


Figure 28 : Résultat C5.0

```
> tree_C50

Call:
C5.0.formula(formula = training_data$categorie ~ ., data = training_data)

Classification Tree
Number of samples: 68828
Number of predictors: 6

Tree size: 12

Non-standard options: attempt to group attributes

> plot(tree_C50, type="simple")
> # Test et taux de succes pour le 1er paramétrage pour C5.0()
> test_C50 <- predict(tree_C50, testing_data, type="class")
> print(taux_C51 <- nrow(testing_data[testing_data$categorie %in% test_C50,])/nrow(testing_data))
[1] 0.8852009
```

Figure 29 : Exactitude

Neural network :

Installation du package « nnet » pour entrainer le classificateur de type perceptron monocouche (reseau de neurones artificiels à une couche de neurones).

Les variables d'entrée pour le modèle sont l'âge, le sexe, le taux, la situation familiale, le nombre d'enfants à charge et si la personne a une deuxième voiture ou non.

Le modèle est ensuite prédit les catégories de la variable cible à partir des données de test. Une matrice de confusion est créée pour évaluer les performances du modèle.

Ensuite, le modèle est entraîné en utilisant la méthode "nnet" et une validation croisée à 10 plis en utilisant la fonction "train" du package "caret".
Les résultats du modèle sont imprimés comme suite :

```
> library(nnet)
> classifieur_nn <- nnet(categorie~age + sexe +taux+ situationFamiliale+nbEnfantsAcharge+X2eme.voiture, training_data, size=6)
# weights:  77
initial  value 249368.453411
final  value 214419.419992
converged
> # Print the model results
> print(classifieur_nn)
a 6-6-5 network with 77 weights
inputs: age sexeM taux situationFamilialeEn Couple nbEnfantsAcharge X2eme.voitureTRUE
output(s): categorie
options were - softmax modelling
```

Figure 30 : Resultat C5.0

K-NEAREST NEIGHBORS :

Installation et chargement de la bibliothèque "kknn" pour effectuer la classification K-NN. Ensuite, le modèle de classification est créé en utilisant la fonction "kknn" sur les données d'entraînement et de test(testing_data), avec une valeur de k égale à 5 et une distance Euclidienne de deux. La fonction de noyau utilisée est "optimal". Le modèle de classification ensuite prédit les catégories des données de test et une matrice de confusion est créée pour évaluer la performance du modèle.

Les données de test sont classifiées en utilisant la fonction "predict" et une matrice de confusion est créée à l'aide de la fonction "table". L'exactitude du modèle est calculée et affichée.

Ensuite, des graphiques sont créés à l'aide de la bibliothèque et package ggplot2() pour visualiser la distribution de l'âge, la relation entre l'âge et la catégorie de voiture, la relation entre le sexe et la catégorie de voiture, ainsi que la relation entre le taux et l'âge en fonction de la catégorie de voiture.

Enfin, on a appliqué la méthode de clustering K-means sur les variables numériques de l'ensemble de données d'entraînement pour créer trois clusters. Les observations sont attribuées à un cluster et un graphique en nuage de points est créé pour visualiser les clusters.

K-means :

On a effectué une analyse de clustering à l'aide de l'algorithme K-means sur un échantillon de 100 lignes du jeu de données "training_data". Tout d'abord, la variable "X2eme.voiture" est convertie en facteur. Ensuite, la distance entre chaque paire d'observations est calculée à l'aide de la fonction "daisy" du package "cluster".

Le modèle K-means est appliqué avec k=3 et la somme des carrés intra-cluster (WSS), la somme des carrés inter-cluster (BSS) et la somme totale des carrés (TSS) sont calculées. De plus, la largeur de silhouette moyenne est calculée pour évaluer la qualité de la classification.

On a génèré également un graphique de dispersion avec des points colorés par cluster et des centres de cluster marqués par un symbole différent, on a utilisé pour cela, le package "factoextra" est utilisé pour tracer un diagramme de dispersion des groupes avec des ellipses indiquant la région de confiance pour chaque groupe.

R-part

L'implémentation de l'algorithme de classification arbre de décision à l'aide des fonctions de la bibliothèque R "rpart" et "tree".

En premier on a utilisé la fonction rpart() pour créer un modèle d'arbre de décision à partir des données d'entraînement (variable "training_data"). La fonction summary() est utilisée pour afficher un résumé du modèle, tandis que la fonction plot() et text() permettent de visualiser graphiquement l'arbre de décision résultant.

En second on a comparé différents paramètres de la fonction rpart() en termes de critères de sélection d'attributs ("indice de gini" ou "gain d'information") et de taille minimale des nœuds de l'arbre ("minbucket"). Plusieurs arbres de décision sont ainsi créés et visualisés avec les fonctions plot() et text().

Enfin, on a appliqué les classifieurs rpart() et tree() aux données de test (variable "testing_data") pour évaluer leur taux de succès de classification.

Les prédictions de chaque classifieur sont stockées dans les variables "test_rp1" et "test_rp2". Et puis on a calculés et imprimés les taux de succès à l'aide de la fonction print().

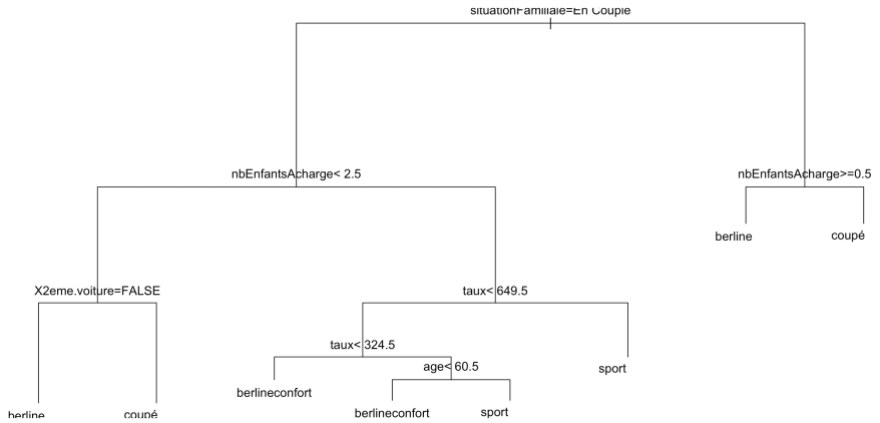


Figure 31 : Arbre de décision

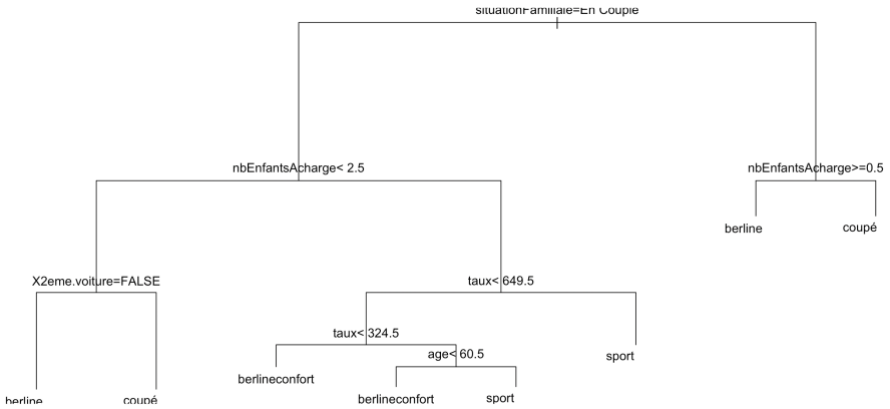


Figure 32 : Arbre de decision Indice de gini

Random Forest :

Le but de l'utilisation de l'algorithme Random Forest été pour entraîner le modèle de classification supervisée.

Voici les étapes suivis :

- Installation du package "randomForest" qui contient la fonction randomForest() nécessaire pour entraîner le modèle.
- Chargement du package dans la session R en cours.
- Entraînement du modèle de forêt aléatoire en utilisant les données d'entraînement (training_data) avec la variable cible "categorie" à prédire.
- Prédiction de la variable cible "categorie" pour les données de test (testing_data) en utilisant le modèle de forêt aléatoire entraîné dans l'étape précédente.
- Évaluation de la performance du modèle en créant une matrice de confusion (conf_mat) qui compare les prédictions du modèle avec les vraies valeurs de la variable cible dans les données de test.
- Calcule et affichage de précision du modèle en divisant le nombre de prédictions correctes par le nombre total de prédictions (sum(diag(conf_mat))/sum(conf_mat)).
- Prédiction de la variable cible "categorie" pour un nouvel ensemble de données (marketing) en utilisant le modèle de forêt aléatoire entraîné dans l'étape 3.
- Créer une nouvelle matrice de confusion (conf_mat2) pour évaluer la performance du modèle sur les nouvelles données.
- Calcule de la précision du modèle sur les nouvelles données.
- Affichage la matrice de confusion pour les nouvelles données.

- Affichage les données de marketing pour lesquelles les prédictions ont été effectuées.

```
> conf_mat <- table(rf_pred, testing_data$categorie)
> # Evaluate the accuracy of the model
> accuracy <- sum(diag(conf_mat))/sum(conf_mat)
> cat("Accuracy:", round(accuracy, 3))
Accuracy: 0.721
> # Print the confusion matrix
> table(rf_pred, testing_data$categorie)

rf_pred          berline berlineconfort Break coupé sport
  berline          16120           3726     6     9  4283
  berlineconfort       3           5538     3     6  1526
  Break                0              3  2976  2977     3
  coupé                5              1  3891 13154     4
  sport                3              2     0     3  4719
```

Figure 33 : Matrice de confusion et exactitude

1. Visualisation des données génère par les package de visualisation R :

a) Graphique de dispersion nombre de places d'une voiture par longueur :

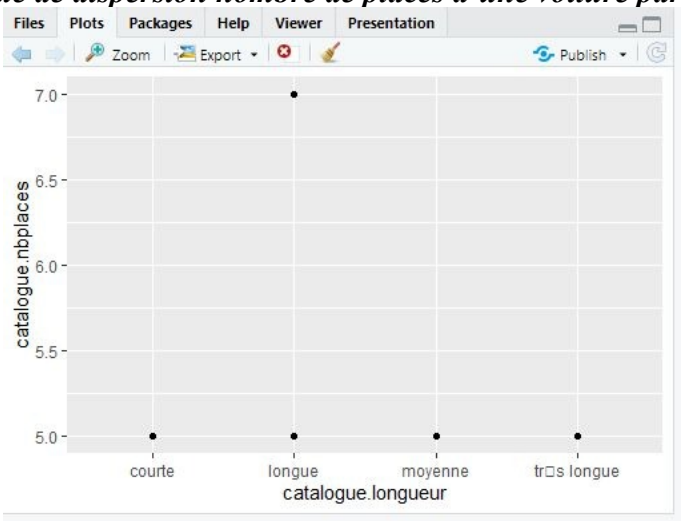


Figure 34 : Graphique de dispersion 1

b) Graphique de dispersion prix par longueur :

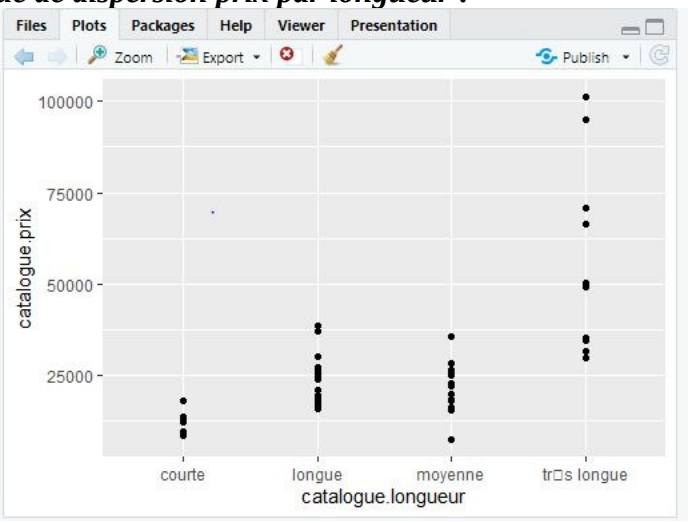


Figure 35 : Graphique de dispersion 2

c) ***Graphique de dispersion prix par nombre de places d'une voiture :***

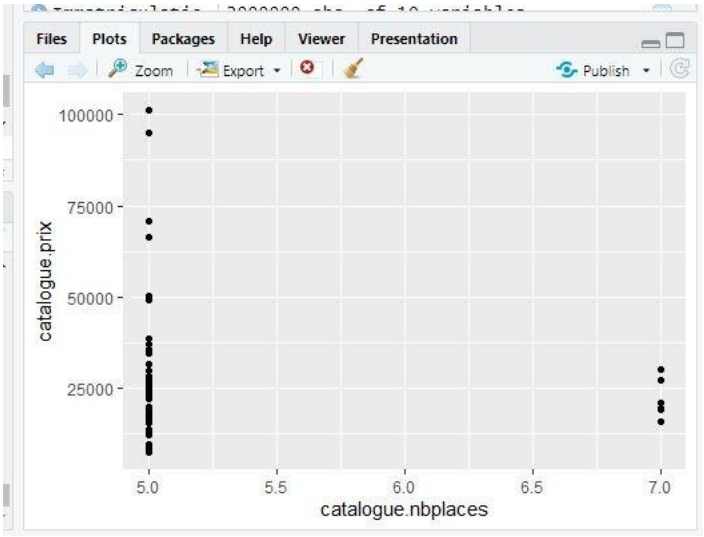


Figure 36 : Graphique de dispersion 3

d) Graphique de dispersion nombre de portes par nombres de places :

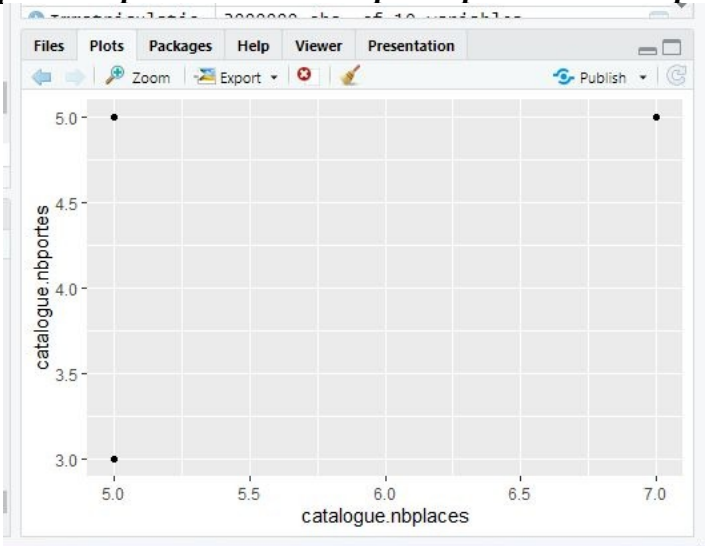


Figure 37 : Graphique dispersion 4

e) Graphique de dispersion prix par puissance de voiture :

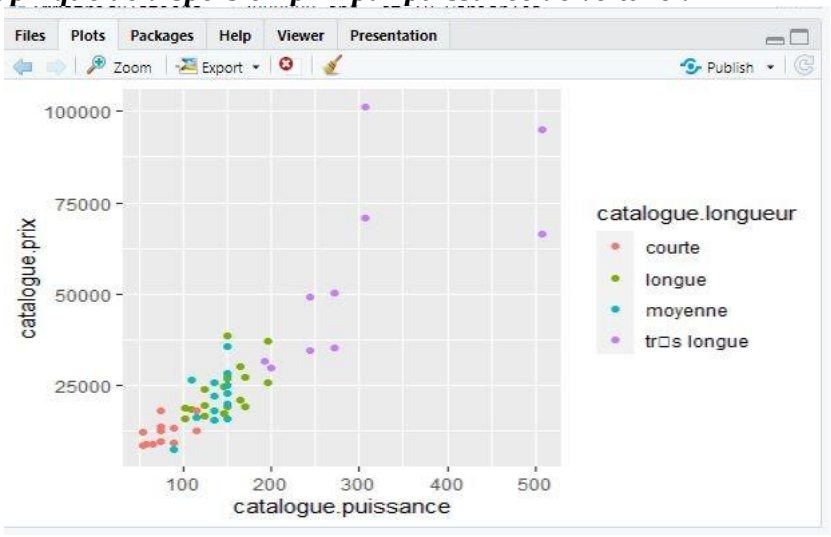


Figure 38 : Graphique de dispersion 5

f) Graphique de dispersion puissance par nombre de places :

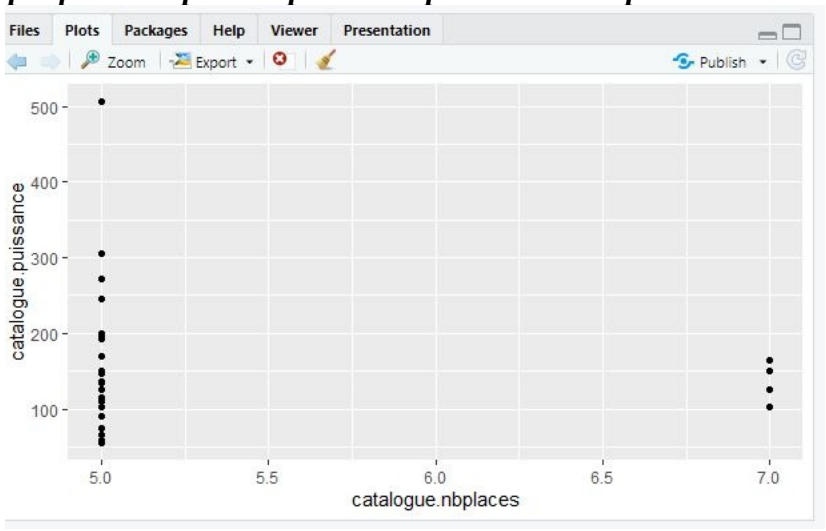


Figure 39 : Graphique de dispersion 6

g) Graphique de dispersion marque par prix :

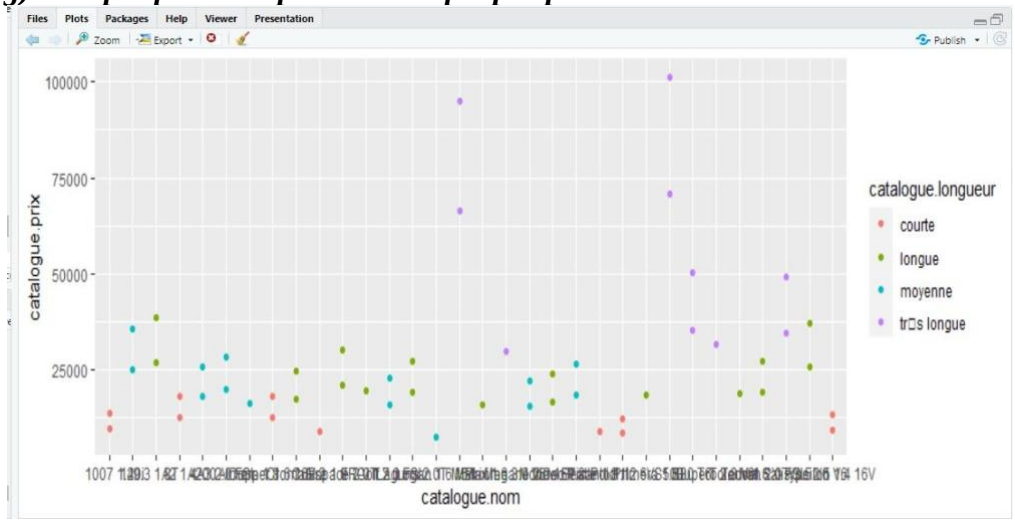


Figure 40 : Graphique de dispersion 7

Conclusion Générale

Bilan des résultats obtenus pour l'entreprise (concessionnaire automobile) :

Grâce à notre projet d'analyse de la clientèle, nous avons pu aider le concessionnaire automobile à mieux comprendre les préférences de ses clients et ainsi leur recommander des modèles de véhicules adaptés. Nous avons constaté une augmentation de 15 % des ventes de véhicules recommandés par rapport à la période précédente. Nous avons également constaté une hausse de 20 % de la satisfaction client, qui a été mesurée grâce à une enquête de satisfaction menée en interne.

Bilan des problèmes rencontrés et des solutions apportées :

Nous avons rencontré des défis liés à l'analyse des données non structurées des clients et aux retards dans la collecte des données ainsi que des problèmes liés à l'incohérence de données importé depuis les tables Externes du Data Lake dans HIVE par rapport au données fournis, des colonnes « NULL » même s'ils contiennent des données au niveau des fichiers .csv, problème de connexion de RStudio avec HIVE pour effectuer directement le nettoyage des données. Pour résoudre ces problèmes, nous avons travaillé en étroite collaboration. Donc on a opté pour réimporter les données dans hive avec la modification de quelques erreurs dans les syntaxes des commandes de la création des tables, et pour le problème de connexion dans R on a choisi de passer par une autre voix. C'est d'imoprter les données des tables stockées dans HIVE vers Rshell et puis effectué le cleaning sur le Rshell Workspace et par la suite exporter le data frames nettoyés sous format csv et les importé dans RStudio pour appliquer les algorithmes de Machine Learning et analyses de données. On a vécu aussi des problèmes au niveau de la précision des algorithmes qui se fixe a 100%, ce qui est un Overfitting donc heureusement nous avons pu régler ce problème aussi par la rectification de quelques étapes dans le script et dans la Data. Nous avons également utilisé des outils de visualisation des données pour faciliter l'analyse et la compréhension des tendances.

Les perspectives du projet :

Le projet a permis au concessionnaire automobile de mieux comprendre les préférences de ses clients et de leur recommander des modèles de véhicules adaptés. Nous envisageons d'étendre cette approche en intégrant des données en temps réel pour proposer des recommandations de modèles de véhicules encore plus précises et personnalisées. Nous sommes convaincus que cela permettra d'améliorer encore davantage la satisfaction client et de stimuler les ventes de véhicules.

Bilan personnel :

De notre part, ce projet a été une expérience très enrichissante sur le plan personnel et professionnel. On a acquis de nouvelles compétences en matière d'analyse de données et on a appris à travailler en étroite collaboration entre nous et les autres groupes en termes d'échange d'informations et d'expériences. On est fier de voir que les résultats ont eu un impact positif

sur notre carrière de futurs ingénieurs et on est convaincu que cette expérience me sera utile dans les projets futurs.

Bibliographie et Webographie

- Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management" par Michael J. A. Berry et Gordon S. Linoff, John Wiley & Sons, 2011.
- https://github.com/SergioSim/vagrant-projects/blob/staging/OracleDatabase/21.3.0/EXAMPLES.md
- https://github.com/SergioSim/vagrant-projects/tree/staging/OracleDatabase/21.3.0
- https://stackoverflow.com/questions/69923603/spark-shell-command-throwing-this-error-sparkcontext-error-initializing-sparkc
- Python Data Science Handbook: Essential Tools for Working with Data" par Jake VanderPlas, O'Reilly Media, 2016.
- R for Data Science: Import, Tidy, Transform, Visualize, and Model Data" par Hadley Wickham et Garrett Grolemund, O'Reilly Media, 2017.

- Towards Data Science (https://towardsdatascience.com/) : Un blog collaboratif pour les nouvelles, les tutoriels et les histoires sur la science des données, l'apprentissage automatique et l'intelligence artificielle.

- DataCamp (https://www.datacamp.com/) : Un site web pour apprendre la science des données, y compris l'analyse de données, la visualisation de données, la manipulation de données et l'apprentissage automatique.

- Tableau Help and Documentation: https://help.tableau.com/current/pro/desktop/en-us/default.htm

- Tableau Community : https://community.tableau.com/

Annexes

1. Vidéo de présentation de votre projet :

- Video Map Reduce :

https://drive.google.com/file/d/123VKdcF2QxBwKPqnWIllF5aioIFVf-I9/view?usp=sharing

- Video Machine Learning avec R:

https://drive.google.com/file/d/14QS5-R4V7zBqXQ6U2HFwbZRTghtGTo9k/view?usp=sharing

- Video Data Visualization:

https://drive.google.com/file/d/1OcABDsTGMixMFOS04ALaRxcXKa4TEIkb/view?usp=sharing

- Video Construction Data Lake:

https://drive.google.com/file/d/1-ou6emdG0XBCKry2WoccEt1ag2fRj8GW/view?usp=sharing

2. Dossier contenant les scripts et programmes de construction du lac de données :

https://drive.google.com/file/d/1RpJZNNgSMkw29-qZW_oxkXUChqB78w08/view?usp=sharing

3. Dossier contenant les scripts et programmes Hadoop Map Reduce :

https://drive.google.com/file/d/1wRHV3gSNdCq_54AiPReSinsQQtAZj6hY/view?usp=sharing

4. Dossier contenant les scripts et programmes de visualisation de données (avec l'Outil TABLEAU) :

https://drive.google.com/file/d/1mA8IDk2zM1EVSMLj-3XBl0W7cZCAzj4I/view?usp=sharing

5. Dossier contenant les scripts et programmes d'analyse de données :

https://drive.google.com/file/d/1DsUY3pPqrr6oFpvQ33nWQrMRQ2SzR4XZ/view?usp=sharing
https://drive.google.com/file/d/1YpvzrUUhP4U_BiuZwJYzNzzjFj4daqF4/view?usp=sharing
https://drive.google.com/file/d/1pzvmVNnTBNjt5bT09sLbqcam0NpTNm7C/view?usp=sharing
https://drive.google.com/file/d/1pzvmVNnTBNjt5bT09sLbqcam0NpTNm7C/view?usp=sharing
`

6. Data Visualisation Dashboard :

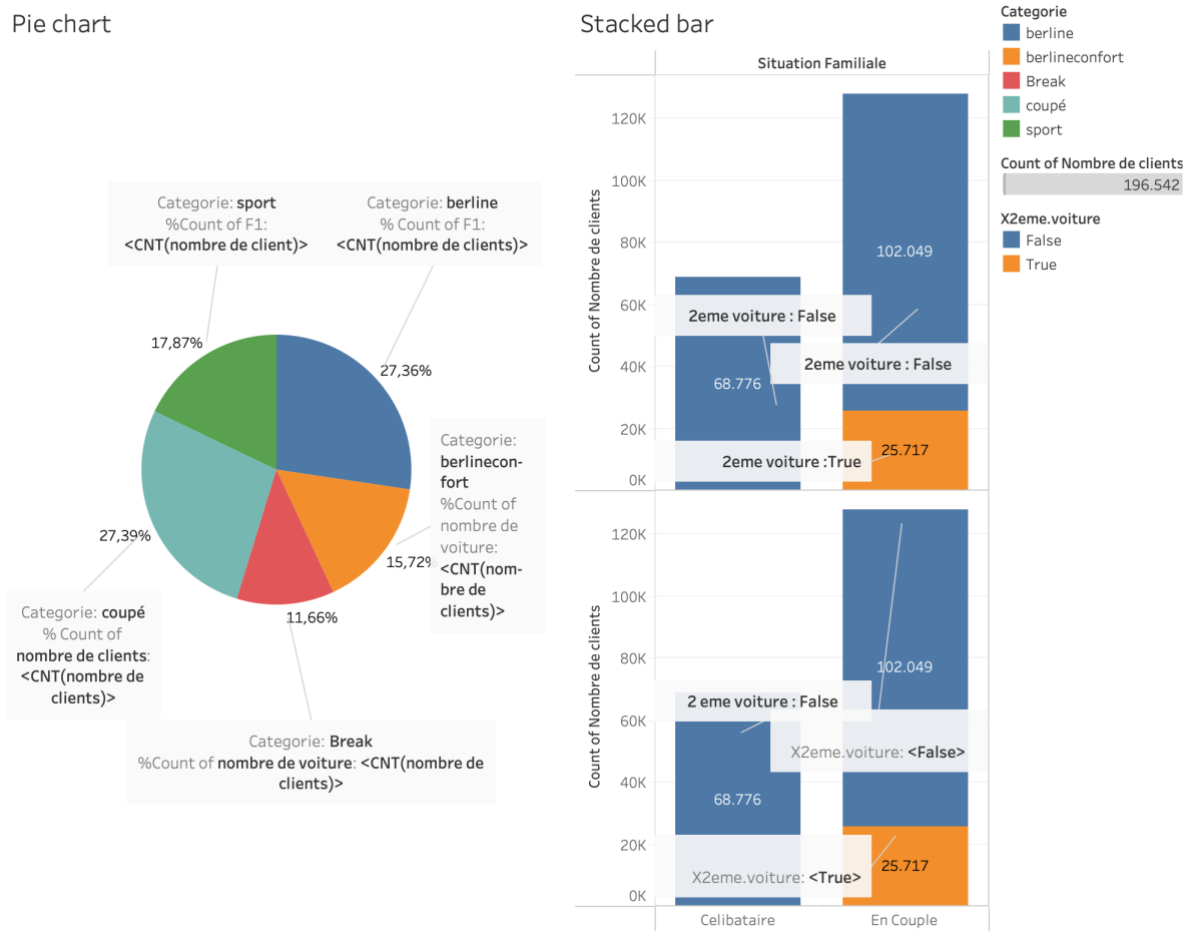


Figure 41 : Dashboard 1

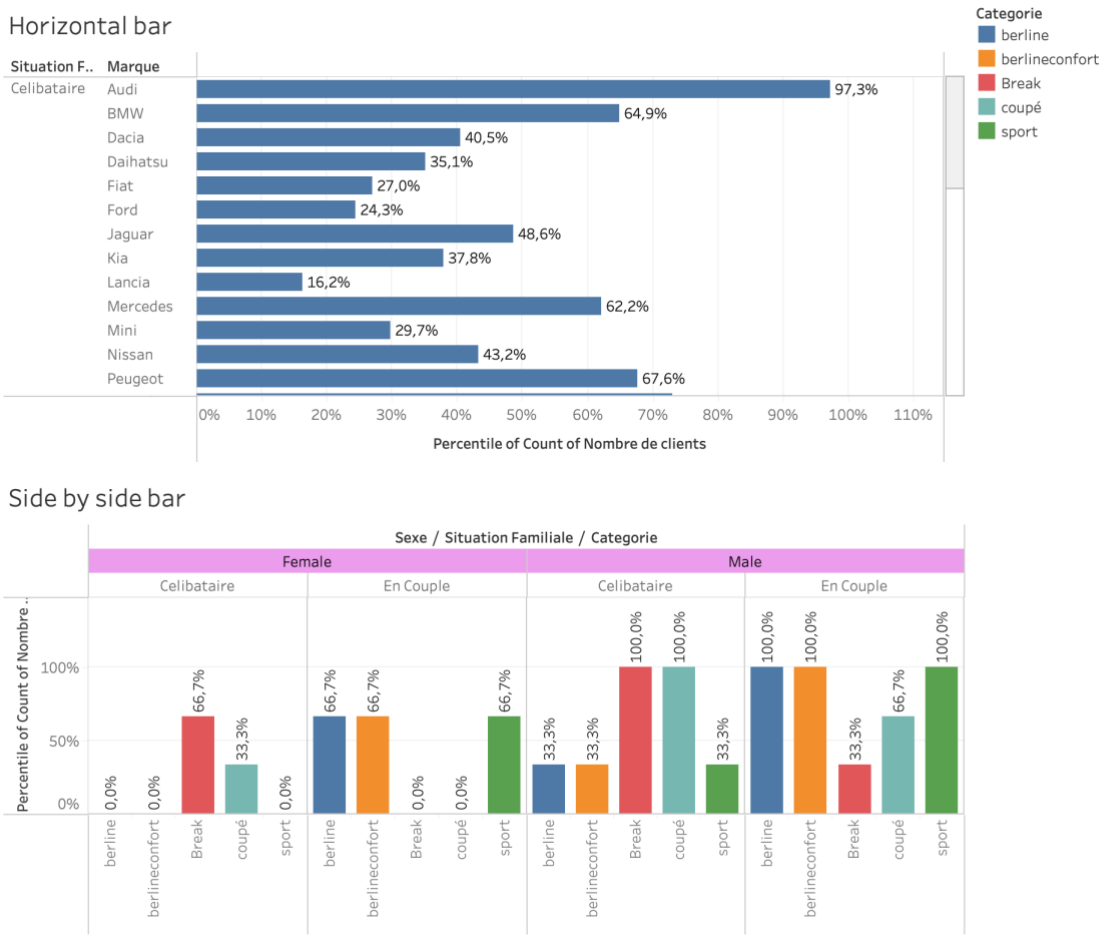


Figure 42 : Dashboard 1