Analyse de la Clientèle d'un Concessionnaire Automobile pour la Recommandation de Modèles de Véhicules

TPA

Durée du projet

19 Octobre 2023 - 31 décembre 2023

Github

Encadrants pédagogiques

PASQUIER Nicolas
MOPOLO Gabriel
SIMONIAN Sergio
WINKLER Marco

MBDS 2023 - AGLAE - BONE - CHIAPPE - LAPSHINA - VINCENT

Résumé

Le projet présenté visait à améliorer la compréhension et la gestion des données dans le secteur automobile, en mettant l'accent sur l'analyse des tendances et des préférences des clients. La nature de ce travail était principalement axée sur l'analyse de données et le Machine Learning, avec une application spécifique aux catégories de voitures.

L'envergure du travail a englobé plusieurs aspects du traitement des données, allant du nettoyage initial des données à leur visualisation avancée. Cela a inclus la manipulation de grandes quantités de données et l'application de techniques sophistiquées pour extraire des informations pertinentes.

Pour atteindre ces objectifs, nous avons utilisé diverses méthodes. Les techniques de nettoyage et de préparation des données ont été cruciales, en particulier pour le traitement des caractéristiques catégorielles. Nous avons également mis en œuvre des modèles de Machine Learning, tels que RandomForestClassifier, pour classer les différentes catégories de voitures. Des outils de visualisation comme Grafana ont été utilisés pour représenter les données de manière intuitive et compréhensible.

Les principaux résultats ont été impressionnants, avec des modèles de Machine Learning atteignant des précisions de 76% et 78% dans la classification des catégories de voitures. Ces résultats démontrent l'efficacité des méthodes utilisées et leur applicabilité dans le domaine de l'analyse de données automobiles.

En conclusion, ce travail a abouti à des avancées significatives dans la compréhension des données du secteur automobile. Il a permis d'améliorer les décisions stratégiques et marketing en fournissant des analyses approfondies et des prévisions précises. Les résultats indiquent également des perspectives prometteuses pour l'application future de ces méthodes dans des scénarios marketing plus larges.

Mots clés:

- Analyse de données
- Secteur automobile
- Machine Learning
- Visualisation des données
- RandomForestClassifier
- Grafana
- Classification des catégories de voitures
- Nettoyage des données
- Préparation des données
- Tendances du marché automobile

Abstract

This project was aimed at enhancing the understanding and management of data in the automotive sector, focusing on analyzing trends and customer preferences. The nature of this work was primarily centered around data analysis and machine learning, with a specific application to car categories.

The scope of the work encompassed various facets of data handling, ranging from initial data cleansing to advanced data visualization. This involved handling large volumes of data and applying sophisticated techniques to extract relevant insights.

To achieve these objectives, a range of methods were employed. Data cleansing and preparation techniques were pivotal, especially for dealing with categorical features. Machine learning models, such as the RandomForestClassifier, were implemented to classify different car categories. Visualization tools like Grafana were used for an intuitive and comprehensible representation of data.

The main results were impressive, with machine learning models achieving accuracies of 76% and 78% in car category classification. These outcomes demonstrate the efficacy of the methods used and their applicability in the automotive data analysis realm.

In conclusion, the work led to significant advancements in understanding automotive sector data. It facilitated improved strategic and marketing decisions by providing in-depth analyses and accurate forecasts. The findings also indicate promising prospects for the future application of these methods in broader marketing scenarios.

Keywords:
- Data Analysis
- Automotive Sector
- Machine Learning
- Data Visualization
- RandomForestClassifier
- Grafana
- Car Category Classification
- Data Cleansing
- Data Preparation
- Automotive Market Trends

Liste des figures

Liste des acronymes

- API : Application Programming Interface
- BDA/DL : Big Data Analytics/Data Lake
- BI : Business Intelligence
- CSV : Comma-Separated Values
- D3.js : Bibliothèque JavaScript pour la création de visualisations interactives
- DataBus : Bus de données
- Datalake : Lac de données (Data Lake en anglais)
- DOM - Document Object Model
- ETL : Extract, Transform, Load
- F1-Score - Une mesure d'efficacité pour les modèles de classification en Machine Learning
- Grafana : Plateforme de visualisation de données
- HDFS : Hadoop Distributed File System
- JSON : JavaScript Object Notation
- K-Means - Algorithme de clustering en K moyennes
- ML - Machine Learning
- MLP - Multi-Layer Perceptron
- MongoDB : MongoDB (Système de gestion de base de données orientée documents)
- Nats JetStream : Système de messagerie et de streaming en temps réel
- NoSQL : Not Only SQL (Non-relationnel)
- PK - Primary Key (Clé Primaire)
- PostgreSQL - Souvent abrégé en "Postgres", mais ce n'est pas un acronyme.
- PostgreSQL : Système de gestion de base de données relationnelle
- Python : Programmation Orientée Objet
- Redis - Remote Dictionary Server
- Redis : Système de gestion de base de données clé-valeur en mémoire
- SKLearn : Scikit-learn (une bibliothèque de machine learning en Python)
- Spark : Apache Spark (un moteur de traitement des données en cluster)
- SQL - Structured Query Language
- SVM - Support Vector Machine
- Upsert - Opération combinée de mise à jour (update) et d'insertion (insert)

Plan du document

1. Introduction générale

Dans un monde où la technologie évolue à un rythme sans précédent, l'industrie automobile se trouve à l'intersection de plusieurs transformations majeures. Le Big Data est devenu un outil indispensable pour comprendre et répondre aux besoins changeants des clients. Ce rapport présente notre projet réalisé dans le cadre du cours de M2 MIAGE-MBDS, intitulé "Analyse de la Clientèle d'un Concessionnaire Automobile pour la Recommandation de Modèles de Véhicules". Ce projet vise à exploiter les données clients et véhicules pour fournir des recommandations personnalisées et améliorer l'expérience client.

Le concessionnaire automobile, notre client, nous a fourni un accès à l'ensemble de données, incluant son catalogue de véhicules, les informations sur les immatriculations de l'année en cours, et un fichier client détaillé. Notre objectif principal est de développer un système capable d'analyser ces données pour recommander le véhicule le plus adapté aux besoins spécifiques de chaque client. Cette tâche implique non seulement une compréhension approfondie des préférences et comportements des clients mais aussi une analyse technique des caractéristiques des véhicules.

Pour relever ce défi, nous avons adopté une architecture Big Data Analytics/Data Lake (BDA/DL) en utilisant des technologies telles que Python avec Spark pour le traitement de données, PostgreSQL pour la gestion des données SQL, et MongoDB et Redis pour les bases de données NoSQL. Un bus de données Nats Jetstream a été intégré pour faciliter l'échange de données en temps réel entre les différentes composantes de notre système.

Ce rapport détaille notre approche pour la mise en œuvre de ce projet, en commençant par la conception de l'architecture, le choix des technologies, la gestion et la visualisation des données, jusqu'à l'analyse approfondie des données pour générer des recommandations pertinentes. Nous discuterons également des défis rencontrés et des solutions adoptées pour surmonter ces obstacles, ainsi que des résultats obtenus et de leur impact potentiel sur l'activité du concessionnaire automobile.

Ce rapport se structure en plusieurs parties clés, débutant par une présentation détaillée du contexte et des objectifs du projet, suivie d'une exploration approfondie de l'architecture et des technologies utilisées, puis d'une analyse des données et des méthodologies employées, pour finalement conclure sur les résultats obtenus et les perspectives futures.

2. Présentation du projet

Le projet se distingue par son caractère innovant, s'inscrivant à la croisée des chemins entre la technologie Big Data et le secteur automobile en constante évolution. L'utilisation de techniques avancées d'analyse de données pour personnaliser les recommandations de véhicules représente une avancée significative dans la manière dont les concessionnaires interagissent avec leurs clients. Ce projet va au-delà de la simple vente de véhicules. Il vise à transformer l'expérience client en utilisant des insights basés sur des données pour répondre de manière proactive et précise aux besoins individuels des clients.

Dans le contexte métier actuel, les concessionnaires automobiles sont confrontés à une concurrence accrue et à des clients de plus en plus informés et exigeants. L'industrie automobile, en pleine mutation avec l'émergence de nouvelles technologies et de

préoccupations environnementales, requiert une approche plus ciblée et personnalisée de la vente. Notre projet s'inscrit dans ce contexte en offrant une solution qui permet au concessionnaire de mieux comprendre ses clients et de leur proposer des véhicules qui correspondent précisément à leurs attentes et besoins, en se basant sur une analyse approfondie des données clients et véhicules.

La criticité de ce projet réside dans sa capacité à transformer les données en insights actionnables qui peuvent directement influencer les décisions de vente et la satisfaction client. Les principaux enjeux incluent l'amélioration de l'expérience client, l'augmentation des ventes par des recommandations personnalisées, et l'optimisation de la gestion des stocks de véhicules.

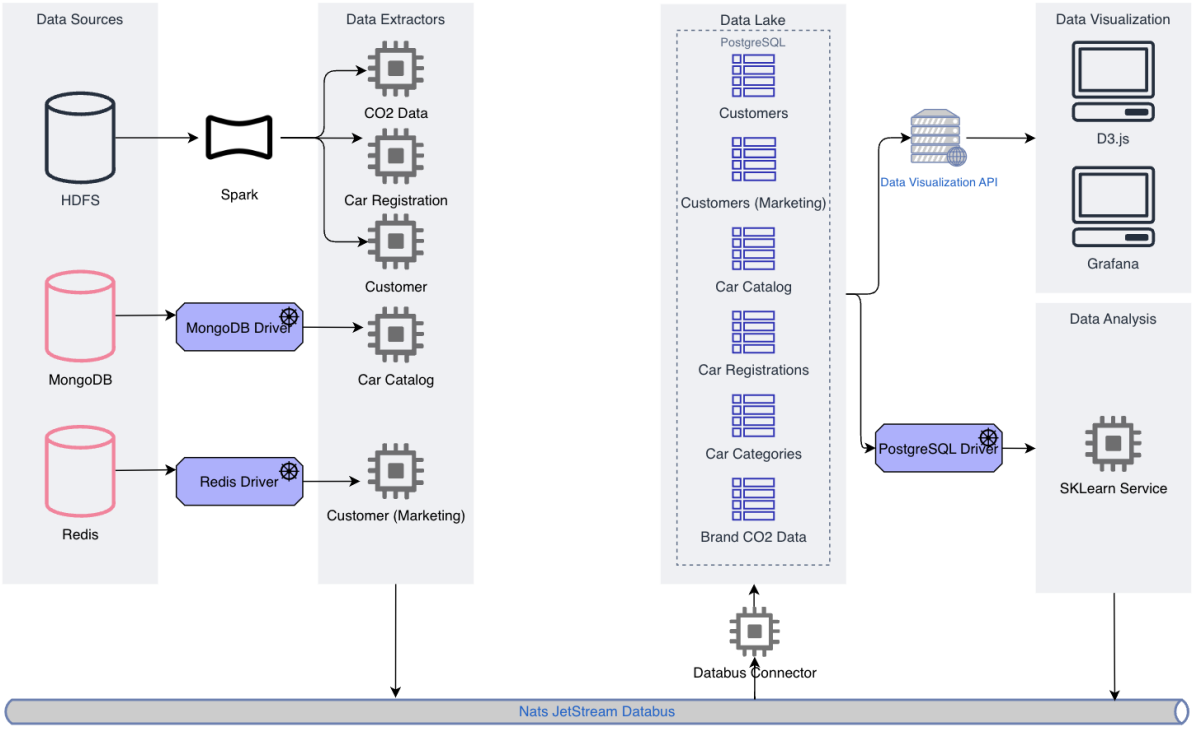
Cependant, le projet comporte également des risques significatifs. Le premier est la qualité des données : des données inexactes ou incomplètes peuvent mener à des recommandations erronées. Un autre risque est lié à la technologie : la dépendance à des systèmes complexes de Big Data nécessite une expertise technique élevée et pose des défis en termes de maintenance et de mise à jour. Enfin, il y a un risque de confidentialité et de sécurité des données, crucial dans le traitement des informations personnelles des clients.

3. Répartition du travail en membre du groupe

Le projet "Analyse de la Clientèle d'un Concessionnaire Automobile pour la Recommandation de Modèles de Véhicules" est une collaboration étroite entre les membres de notre équipe, chacun apportant son expertise pour réaliser les différents aspects du projet. Voici comment les responsabilités ont été réparties :

- Yehoudi et Valeria - Data Analysis / Data Visualization : Yehoudi et Valeria se concentrent sur l'analyse des données et la visualisation. Ils utilisent des outils pour interpréter les données clients et véhicules, et développent des visualisations interactives pour présenter les insights de manière compréhensible.
- Ernesto - Spark / Data Treatment / Data Analysis / Data Visualization : Ernesto joue un rôle dans le traitement des données en utilisant Spark, un outil puissant pour le traitement du BigData. Il contribue également à l'analyse des données et à la création de visualisations, en s'assurant que les insights extraits sont précis et pertinents.
- Mike - Data Treatment / Databus Management / Scripts d'Automatisation / Infrastructure Management : Mike est responsable du traitement des données présentent sur MongoDB et Redis, la gestion du bus de données et la mise en place du connecteur de bus de données, son intégration aux modules existants et développe des scripts d'automatisation pour assurer un flux de données fluide et efficace entre les différentes sources de données et le data lake.
- Sébastien - Database Management / Datalake API / Data Visualisation : Sébastien gère la base de données SQL (PostgreSQL) et NoSQL (MongoDB et Redis). Il est chargé de structurer les tables de données et de mettre en place l'API pour faciliter l'accès et la manipulation des données par les autres membres de l'équipe. Il devra mettre en œuvre le grafana et créer toutes les requêtes SQL.

4. Architecture du data lake



Architecture du data lake

Notre architecture est divisés en 6 modules : Data Source, Data Extractor, Data Lake, Data Visualization, Data Analysis et Databus :

Data Source

- HDFS : Le système de fichiers distribué Hadoop, utilisé pour le stockage de données volumineuses.
- MongoDB : Une base de données NoSQL orientée documents, souvent utilisée pour stocker des données semi-structurées ou non structurées.
- Redis : Un magasin de structure de données en mémoire, utilisé pour les bases de données, la mise en cache et le courtage de messages.

Data Extractor

- CO2 Data : Extracteur responsable du traitement des données de **CO2.csv** et qui utilise Spark pour le traitement des données
- Car Registration : Extracteur qui traite les données d' **Immatriculations.csv** et qui utilise Spark.
- Customer : Extracteur traitant les CSVs **Clients_xx.csv** et qui utilise Spark pour cela.
- Car Catalog : Extracteur qui traite les véhicules du catalogue depuis les lignes du fichier **Catalogue.csv** chargé dans MongoDB. L'extracteur est connecté via le Driver MongoDB et traite les données via la pipeline Aggregate.
- Customer (Marketing) : Extracteur traitant les données provenant du CSV **Marketing.csv.** Chaque ligne est convertie en objet JSON et stockée sous forme d'objet dans la BDD Redis.

Data Lake

- C'est un entrepôt centralisé qui permet de stocker toutes les données structurées après le traitement de nos extracteurs.
- PostgreSQL : Une base de données relationnelle utilisée ici pour organiser les données dans le Data Lake.
- Les données sont structurées en différentes tables (Clients, Clients (Marketing), Catalogue de voitures, Immatriculations de voitures, Catégories de voitures, Données CO2 des marques).

Data Visualization

- Des outils comme D3.js et Grafana sont utilisés pour créer des représentations graphiques des données. Ils sont alimentés par une API de visualisation de données.

Data Analysis

- SKLearn Service : Ce service utilise la bibliothèque de machine learning Scikit-learn pour effectuer des analyses de données, comme la classification, la régression, le clustering, etc.

Databus

- Permet la communication entre les différents composants de notre architecture. Un databus nous permet l'ajout de nouveaux composants sans affecter les autres composants.
- Nats JetStream est utilisé comme bus de données. Il permet la transmission en temps réel des événements (ajout et mise à jour des données à effectuer dans le data lake) de manière fiable et sécurisée (persistance des messages non distribués, chiffrement TLS).
- Databus Connector : Permet la récupération des événements du databus et effectue les opérations pour l'ajout ou la mise à jour des données.

5. Construction du data lake par étape

a. Collecte des Données

Ensemble des programmes et scripts disponibles ici : Github.
Interface d'administration de la BDD disponible ici : PGADMIN (user: prof, password: BigDataMBDS)

Processus de Chargement des Données

Le processus de collecte des données dans notre architecture de Data Lake commence par le chargement des données à l'aide du programme Python **data_upload**. Ce programme est configuré pour interagir avec diverses sources de données, notamment HDFS, Redis et MongoDB, en utilisant des pilotes spécifiques. Ce programme est hautement configurable et de nouveaux pilotes de stockage peuvent être ajoutés.

Configuration :

```
{
```

```
  "drivers": {
    "HDFS": {"url": "hdfs://135.181.84.87:9000"},
    "Redis": {"host": "135.181.84.87", "port": 6379, "password": "",
"db": 0},
    "MongoDB": {"connection_string":
"mongodb://135.181.84.87:27017/bigdatambds"}
  },
  "files": [
    {"path": "files/CO2.csv", "target": {"type": "HDFS", "path":
"/data_wharehouse/CO2.csv"}},
    {"path": "files/Marketing.csv", "target": {"type": "Redis", "key":
"Marketing"}},
    {"path": "files/Catalogue.csv", "target": {"type": "MongoDB",
"collection": "Catalogue"}}
  ]
}
```

Méthodologie de Chargement

Les fichiers CSV sont chargés depuis leur emplacement local vers des destinations spécifiques dans HDFS ou Redis.

Chaque fichier CSV a un chemin de destination défini, assurant ainsi une organisation structurée des données dans le système de fichiers distribué.

Traitement Batch

Les données sont traitées en batch, ce qui signifie que les données sont chargées par lots à des intervalles réguliers plutôt qu'en temps réel.

b. Extraction et Traitement des données

Ensemble des programmes et scripts disponibles ici : Github.

Le cœur du traitement des données dans notre architecture de Data Lake est orchestré par le programme **data_treater**, qui s'exécute toutes les cinq minutes. Ce programme est responsable de l'exécution de cinq fonctions clés, chacune dédiée au traitement d'un type de données spécifique. Ces fonctions utilisent des technologies comme MongoDB, Redis et Apache Spark pour transformer et préparer les données pour un stockage et une analyse ultérieurs.

Traitement des Données du Catalogue avec MongoDB

La fonction **mongo_treat_car_catalog** est conçue pour traiter les données du fichier **Catalogue.csv**. Elle utilise MongoDB (aggregate) pour exécuter une série d'opérations de transformation sur les données du catalogue de voitures. Ces transformations incluent la standardisation des noms de marques et de modèles, la conversion des attributs de voitures comme la puissance, la longueur, la capacité de sièges, le nombre de portes, et la couleur. Par exemple, les marques sont converties en majuscules, et des cas spécifiques comme "Hyunda." sont corrigés en "Hyundai". De même, les longueurs de voitures sont

catégorisées en termes tels que "short", "medium", "long", et "very_long". Ces transformations garantissent une uniformité et une précision accrues dans les données du catalogue.

Traitement des Données Marketing avec Redis

La fonction **redis_treat_marketing** traite les données du fichier **Marketing.csv**, stockées dans Redis. Cette étape implique la conversion des données textuelles en formats standardisés. Par exemple, les statuts matrimoniaux et les genres sont mappés à des termes uniformes comme "single", "married", "M", et "F". Des validations supplémentaires sont effectuées pour s'assurer de la cohérence des données, telles que la vérification de l'âge (doit être supérieur ou égal à 18 ans) et la conversion des taux de dette en valeurs numériques. Ces transformations sont cruciales pour garantir que les données marketing sont fiables et prêtes pour des analyses plus approfondies.

Traitement des Données d'Immatriculation avec Spark

La fonction **spark_treat_inmatriculation** s'attaque aux données d'immatriculation. Apache Spark est utilisé pour lire, nettoyer et transformer ces données. Les étapes de traitement incluent l'élimination des valeurs nulles, la correction des noms de marques, et la combinaison des colonnes "marque" et "nom" en une seule. De plus, des transformations sont appliquées pour standardiser les attributs tels que la couleur et la longueur des voitures. Par exemple, les couleurs sont mappées à des termes standardisés comme "white", "blue", "grey", etc. Ces transformations sont essentielles pour assurer l'uniformité et la facilité d'analyse des données d'immatriculation.

Traitement des Données CO2 avec Spark

La fonction **spark_treat_co2** est responsable du traitement des données de CO2. Cette étape implique la lecture et le nettoyage des données, la suppression des colonnes inutiles, et la transformation des données pour obtenir des informations claires sur les émissions de CO2, les coûts énergétiques, et les bonus/malus. Des opérations telles que la conversion des colonnes en types de données flottants et la normalisation des noms de marques et de modèles sont effectuées pour garantir la précision et l'utilité des données pour des analyses environnementales.

Traitement des Données Client avec Spark

Enfin, la fonction **spark_treat_client** s'occupe des données clients issues des fichiers **Clients_11.csv** et **Clients_19.csv**. Cette fonction fusionne les deux ensembles de données, applique un nettoyage pour éliminer les valeurs aberrantes ou manquantes, et transforme les données pour normaliser les informations sur les clients. Les transformations incluent la standardisation des statuts matrimoniaux et des genres, ainsi que la conversion des données numériques telles que l'âge, le taux de dette, et le nombre d'enfants à charge. Cette étape est cruciale pour assurer que les données clients sont prêtes pour des analyses démographiques et comportementales.

Intégration de Nats Jetstream dans le Traitement des Données

Après l'exécution des fonctions de traitement des données par le programme **data_treater**, les résultats sont publiés sur Nats Jetstream. Cette étape est cruciale car elle permet la transmission en temps réel des données traitées vers d'autres composants de l'architecture du Data Lake, facilitant ainsi une intégration fluide et une réactivité accrue du système.

Nats Jetstream joue un rôle central dans la gestion des flux de données au sein de notre architecture. Il agit comme un système de messagerie distribué qui assure la livraison fiable et ordonnée des messages (données traitées) aux différents services et composants

du Data Lake. En utilisant Nats Jetstream, nous bénéficions d'une communication asynchrone et non bloquante entre les processus de traitement des données et les autres parties de notre système, comme les bases de données et les services d'analyse.

c. Stockage dans le Data Lake

Ensemble des programmes et scripts disponibles ici : Github.

i. Structure de notre base de données

Le stockage des données dans le Data Lake est structuré autour de plusieurs tables dans une base de données PostgreSQL. Chaque table est conçue pour stocker des données spécifiques avec des colonnes bien définies pour garantir la cohérence et l'intégrité des données. Voici une description détaillée de chaque table et de ses colonnes :

Table **catalog_car_category** : Cette table stocke les différentes catégories de voitures. Chaque catégorie est identifiée par un ID unique et possède un nom descriptif. Elle sert à classifier les voitures dans le catalogue selon des caractéristiques communes.

Colonne	Type	Description
id	INTEGER (PK)	Identifiant primaire de la catégorie, allant de 0 à n.
name	VARCHAR(32)	Nom de la catégorie de voiture.

Table **catalog_car** : La table principale pour les informations détaillées sur chaque voiture. Elle inclut des données telles que la marque, le modèle, la puissance, la longueur, la capacité de sièges, le nombre de portes, la couleur, le statut d'occasion, et le prix. Elle est liée à la table **catalog_car_category** et intègre des données sur les émissions de CO2 et les coûts énergétiques.

Colonne	Type	Description
id	SERIAL (PK)	Identifiant primaire de la voiture.
category_id	INTEGER	Référence à l'identifiant de la catégorie de voiture.
brand	VARCHAR(32)	Marque de la voiture.
name	VARCHAR(255)	Nom du modèle de la voiture.
power	INTEGER	Puissance de la voiture, avec contrainte de valeur.
length	catalog_car_length	Longueur de la voiture, type énuméré. 'short', 'medium', 'long', 'very_long'
seating_capacity	INTEGER	Capacité de sièges, avec contrainte de valeur. [2;5]
number_doors	INTEGER	Nombre de portes, avec contrainte de valeur. [3;7]
color	catalog_car_color	Couleur de la voiture, type énuméré. 'white', 'blue', 'grey', 'black', 'red'

used	BOOLEAN	Indique si la voiture est d'occasion.
price	NUMERIC	Prix de la voiture, avec contrainte de valeur. [5000; 150000]
bonus_malus	NUMERIC	Bonus ou malus écologique associé à la voiture, avec contrainte de valeur. Peut-être nul si aucune information sur le constructeur.
co2_emissions	NUMERIC	Émissions de CO2 de la voiture. Peut-être nul si aucune information sur le constructeur.
energy_cost	NUMERIC	Coût énergétique de la voiture.

Contrainte Unique : Une contrainte unique est définie sur la combinaison des colonnes (brand, name, power, length, seating_capacity, number_doors, color, used). Cette contrainte assure qu'il n'y a pas de doublons pour une combinaison spécifique de caractéristiques de voiture.

Table **customer_markting_analysis_data** : Cette table fait le lien entre les données marketing des clients et les voitures du catalogue. Elle associe chaque client (identifié dans les données marketing) à des catégories de voitures et à des modèles spécifiques, facilitant ainsi l'analyse des préférences et comportements d'achat des clients.

Colonne	Type	Description
customer_marketing_id	INTEGER (PK)	Référence à l'identifiant du client dans la table marketing.
catalog_car_category_id	INTEGER	Référence à l'identifiant de la catégorie de voiture.
catalog_car_id	INTEGER	Référence à l'identifiant de la voiture dans le catalogue.

Table **brand_co2_emissions** : Contient des informations spécifiques sur les émissions de CO2, les coûts énergétiques, et les bonus/malus écologiques pour chaque marque et modèle de voiture. Cette table est essentielle pour les analyses environnementales et la conformité réglementaire.

Colonne	Type	Description
brand	VARCHAR(32)	Marque de la voiture.
car_name	VARCHAR(255)	Nom du modèle de la voiture.
bonus_malus	NUMERIC	Bonus ou malus écologique associé à la voiture, avec contrainte de valeur. [-15000; 15000]

co2_emissions	NUMERIC	Émissions de CO2 de la voiture, avec contrainte de valeur. [0; 500]
energy_cost	NUMERIC	Coût énergétique de la voiture, avec contrainte de valeur. [0; 2000]

La combinaison (brand, car_name) est utilisée comme clé primaire et est donc indexée.

Table **customer_car_registration** : Gère les informations d'immatriculation des voitures associées aux clients. Chaque enregistrement lie un identifiant d'immatriculation à un modèle de voiture spécifique dans le catalogue, permettant de suivre quel client possède quelle voiture.

Colonne	Type	Description
registration_id	VARCHAR(16) (PK)	Identifiant d'immatriculation de la voiture.
car_id	INTEGER	Référence à l'identifiant de la voiture dans le catalogue.

Table **customer** : Stocke des informations détaillées sur les clients, y compris l'âge, le genre, le taux d'endettement, le revenu, le statut matrimonial, le nombre d'enfants à charge, et la possession d'une deuxième voiture. Cette table est cruciale pour l'analyse démographique et comportementale des clients.

customer_id	SERIAL (PK)	Identifiant primaire du client.
age	INTEGER	Âge du client, avec contrainte de valeur. [18;100]
gender	customer_gender	Genre du client, type énuméré. ‘M’, ‘F’
debt_rate	NUMERIC	Taux d'endettement du client, avec contrainte de valeur. [0; +INF[
income	NUMERIC	Revenu du client, calculé à partir du taux d'endettement. debt_rate / 0.3
marital_status	catalog_car_length	Statut matrimonial du client, type énuméré. 'single', 'couple', 'married', 'divorced', 'widowed'
dependent_children	INTEGER	Nombre d'enfants à charge, avec contrainte de valeur. [0; +INF[
has_second_car	BOOLEAN	Indique si le client possède une deuxième voiture.
car_registration_id	VARCHAR(16)	Référence à l'identifiant d'immatriculation

		actuel de la voiture du client.

Une contrainte unique est appliquée sur la combinaison (age, gender, debt_rate, marital_status, dependent_children, has_second_car), assurant l'unicité des profils clients.

Table **customer_marketing** : Similaire à la table customer, mais spécifiquement axée sur les données collectées à des fins marketing. Elle contient des informations démographiques et financières sur les clients, utilisées pour des analyses marketing ciblées et des études de marché.

customer_id	SERIAL (PK)	Identifiant primaire du client.
age	INTEGER	Âge du client, avec contrainte de valeur. [18;100]
gender	customer_gender	Genre du client, type énuméré. ‘M’, ‘F’
debt_rate	NUMERIC	Taux d'endettement du client, avec contrainte de valeur. [0; +INF[
income	NUMERIC	Revenu du client, calculé à partir du taux d'endettement. debt_rate / 0.3
marital_status	catalog_car_length	Statut matrimonial du client, type énuméré. 'single', 'couple', 'married', 'divorced', 'widowed'
dependent_children	INTEGER	Nombre d'enfants à charge, avec contrainte de valeur. [0; +INF[
has_second_car	BOOLEAN	Indique si le client possède une deuxième voiture.

Une contrainte unique est appliquée sur la combinaison (age, gender, debt_rate, marital_status, dependent_children, has_second_car) pour garantir l'unicité des données marketing des clients.

ii. Calcul des données CO2

La table **brand_co2_emissions** joue un rôle crucial dans le suivi des émissions de CO2 et des aspects environnementaux liés aux véhicules. Pour maintenir l'exactitude et la pertinence des données dans cette table et dans d'autres tables liées, des vues, triggers et fonctions sont utilisés. Ces mécanismes automatisent la mise à jour des informations relatives aux émissions de CO2 des véhicules dans le Data Lake.

Vue **brand_co2_average**

- **But** : Cette vue est conçue pour fournir un aperçu des émissions moyennes de CO2, des coûts énergétiques moyens, et des bonus/malus moyens pour chaque marque de voiture.
- **Fonctionnement** : La vue calcule les moyennes des colonnes **co2_emissions**, **energy_cost**, et **bonus_malus** pour chaque marque présente dans la table **brand_co2_emissions**. Elle regroupe les données par marque et effectue les calculs moyens correspondants.

- **Utilisation** : Cette vue nous est utile pour agréger les données de CO2 par constructeur (voir fonctions ci-dessous). De plus, elle permet aux analystes et aux décideurs d'obtenir rapidement des informations sur les performances environnementales moyennes des marques de voitures.

Trigger **update_catalog_car_co2_emissions**

- **But** : Ce trigger est conçu pour mettre à jour automatiquement les données des véhicules dans la table **catalog_car** lorsqu'une nouvelle entrée est insérée ou mise à jour dans la table **brand_co2_emissions**.
- **Fonctionnement** : Lorsqu'un enregistrement est ajouté ou modifié dans **brand_co2_emissions**, ce trigger déclenche une fonction qui calcule et met à jour les valeurs moyennes des émissions de CO2, des coûts énergétiques, et des bonus/malus pour la marque concernée dans la table **catalog_car**.

Trigger **update_catalog_car_co2_emissions_from_brand_co2_emissions**

- **But** : Ce trigger est conçu pour maintenir la cohérence des données entre les tables **brand_co2_emissions** et **catalog_car**.
- **Fonctionnement** : Après chaque insertion ou mise à jour dans **brand_co2_emissions**, ce trigger invoque une fonction qui met à jour les enregistrements correspondants dans **catalog_car**, en s'assurant que les informations sur les émissions de CO2 et les aspects énergétiques sont à jour et cohérentes.

Fonction **update_catalog_car_co2_emissions**

- **Description** : Cette fonction est appelée par le trigger mentionné ci-dessus. Elle récupère les valeurs moyennes des émissions de CO2, des coûts énergétiques, et des bonus/malus pour une marque donnée et met à jour ces valeurs pour tous les véhicules de cette marque dans la table **catalog_car**.
- **Mécanisme** : La fonction effectue une requête sur la vue **brand_co2_average**, qui calcule les moyennes des émissions de CO2, des coûts énergétiques, et des bonus/malus pour chaque marque. Ces moyennes sont ensuite appliquées aux enregistrements correspondants dans **catalog_car**.

Fonction **update_catalog_car_co2_emissions_from_brand_co2_emissions**

- **Description** : Cette fonction est déclenchée par le trigger associé pour mettre à jour les données des véhicules dans **catalog_car** en fonction des changements dans **brand_co2_emissions**.
- **Mécanisme** : Elle met à jour les enregistrements dans **catalog_car** avec les dernières valeurs moyennes d'émissions de CO2, de coûts énergétiques, et de bonus/malus pour chaque marque, assurant ainsi que les données reflètent les informations les plus récentes et précises.

Cet ensemble constitue la structure fondamentale du Data Lake, permettant un stockage organisé et efficace des données. Les contraintes et les types spécifiques garantissent l'intégrité et la validité des données stockées, facilitant ainsi leur utilisation pour des analyses ultérieures.

d. Data Visualization et Data Analysis

i. Accès aux données via API

Ensemble des programmes et scripts disponibles ici : Github.

L'API de notre Data Lake, construite avec le framework Go gin, joue un rôle crucial dans la facilitation de l'accès aux données. Elle est conçue pour être à la fois flexible et intuitive, permettant de récupérer des données de manière efficace.

Endpoints de notre API

- GET /metrics : Fournit des métriques sur les différentes tables de la base de données. Pour chaque table, elle renvoie le nombre total de lignes.
- GET /:table : Permet de récupérer des données d'une table spécifique. Supporte la pagination avec les paramètres page et size. Renvoie les données sous forme d'agrégats JSON.
- GET /visualization/:table : Offre une fonctionnalité de requête avancée pour une table spécifique. Permet l'utilisation de filtres (where), de tri (sortby et orderby), de limitation (limit) et de conditions (having). Renvoie les données sous forme d'agrégats JSON.
- POST /query : Permet aux utilisateurs d'exécuter des requêtes SQL personnalisées. Les requêtes sont envoyées au format JSON et les résultats sont renvoyés sous forme d'agrégats JSON.

ii. Visualisation des Données

Ensemble des programmes et scripts disponibles ici : Github.

La visualisation des données est un aspect clé de notre architecture, et pour cela, nous utilisons D3.js. Cet outil permet de créer des visualisations de données interactives et détaillées, adaptées à l'exploration et à la compréhension des tendances complexes dans les données. D3.js est particulièrement efficace pour manipuler le DOM et créer des graphiques personnalisés qui mettent en lumière des insights spécifiques des données.

En complément, Grafana est utilisé pour des visualisations supplémentaires, notamment pour créer des tableaux de bord interactifs et des visualisations en temps réel. Grafana excelle dans la présentation des métriques clés et des tendances de manière intuitive, ce qui est essentiel pour la surveillance et l'analyse rapide des données.

iii. Analyse des données

Ensemble des programmes et scripts disponibles ici : Github.

Dans notre projet, l'analyse des données est une étape essentielle qui permet de transformer les informations brutes en insights actionnables et stratégiques.

Préparation et Traitement des Données

Nous commençons par préparer et traiter les données clients et voitures. À l'aide de scripts Python spécifiques, nous effectuons des opérations telles que le remplacement de valeurs, l'encodage de labels, et la suppression de colonnes inutiles. Ces scripts, comme

customers_treater.py et **cars_treater.py**, sont conçus pour nettoyer et structurer les données, les rendant ainsi prêtes pour l'analyse.

Catégorisation et Analyse des Voitures

Ensuite, nous catégorisons les voitures en utilisant l'algorithme K-Means. Ce processus, détaillé dans **categories_creator.py**, implique la sélection de caractéristiques pertinentes telles que les émissions de CO2 et le coût énergétique. Nous utilisons un graphique en coude pour déterminer le nombre optimal de clusters. Chaque voiture est alors assignée à une catégorie spécifique, et ces informations sont publiées sur le bus de données Nats Jetstream.

Modélisation et Prédiction

Pour l'analyse des données clients, nous utilisons un modèle Bayesian Gaussian Mixture, comme décrit dans **bayesian_gaussian_mixture.py**. Ce modèle est entraîné sur les données préparées et évalué pour sa précision. Les prédictions générées sont ensuite utilisées pour enrichir notre compréhension des préférences et comportements des clients.

Publication et Utilisation des Résultats

Les résultats de ces analyses, y compris les prédictions des modèles de machine learning, sont publiés sur Nats Jetstream. Cela nous permet d'intégrer facilement ces insights dans d'autres aspects de notre projet. Par exemple, les prédictions issues des modèles sont utilisées pour enrichir les données marketing, offrant ainsi une perspective plus profonde sur les préférences des clients.

iv. Gestion du Databus

Ensemble des programmes et scripts disponibles ici : Github.

La gestion efficace du flux de données est un élément crucial de notre architecture de Data Lake, et c'est là que le Databus joue un rôle fondamental. Le Databus, en utilisant Nats Jetstream, sert de colonne vertébrale pour la communication et la distribution des données entre les différents composants de notre système.

Publication des Données Traitées

- DATASOURCE.update et DATASOURCE.upsert :
  - Les données traitées, issues des scripts de traitement des données clients et voitures, sont publiées sur deux subjects principaux : **DATASOURCE.update** et **DATASOURCE.upsert**.
  - Ces subjects sont utilisés pour indiquer les opérations à effectuer sur les données : soit une mise à jour (**update**) des enregistrements existants, soit un ajout (**upsert**) qui combine insertion et mise à jour.
  - Chaque événement publié sur ces subjects contient des informations clés telles que **table** (la table dans laquelle les données doivent être ajoutées ou mises à jour) et **tablePK** (la clé primaire à utiliser pour ces opérations).
- DATAANALYSYS.upsert :
  - Les résultats issus des analyses de machine learning, tels que les prédictions ou les classifications, sont publiés sur le subject DATAANALYSYS.upsert.
  - Ce processus garantit que les insights générés par les modèles de machine learning sont correctement intégrés dans le Data Lake, permettant ainsi leur utilisation dans des applications ultérieures, telles que la personnalisation des offres ou l'amélioration des stratégies marketing.

- Comme pour les données traitées, les événements publiés contiennent des informations sur la table cible et la tablePK correspondante, assurant ainsi une intégration précise et cohérente des données.

La gestion du Databus via Nats Jetstream offre plusieurs avantages clés :

- **Fiabilité** : La persistance des messages et la gestion robuste des erreurs garantissent que les données sont transmises de manière fiable et sécurisée.
- **Flexibilité** : La capacité de gérer différents types d'opérations (comme update et upsert) et de cibler des tables spécifiques offre une grande flexibilité dans la gestion des données.
- **Intégration** : La structure claire des messages, avec des informations spécifiques sur les tables et les clés primaires, facilite l'intégration des données dans diverses parties de notre architecture.

6. Hive/Spark (Ernesto)

a. Introduction Spark

Dans le contexte du traitement de données massives, l'introduction aux technologies Spark est essentielle. Spark est un framework open source qui permet de traiter rapidement et de manière distribuée des volumes massifs de données sur des clusters informatiques.

b. Présentation et traitement du fichier CO2.csv

Le fichier CO2.csv nous fournit des informations pertinentes pour l'entraînement des modèles de Machine Learning. Cependant, un nettoyage préalable à son utilisation est nécessaire. Ses informations jouent un rôle important dans la tâche de classification.

Transformation de CO2

Chargement des Données : Lecture du fichier CO2.csv dans un DataFrame Spark.

Élimination des Valeurs Null : Suppression des lignes contenant des valeurs nulles.

Suppression de Colonnes Inutiles : Retrait de la première colonne inutile "_c0".

Renommage de Colonnes : Renommage de la colonne "Cout enerie" en "Cout energie".

Manipulation des Chaînes de Caractères : Pour la colonne 'Bonus / Malus', suppression de tous les caractères après le signe € (conservation des caractères avant le signe €). Manipulations similaires pour d'autres colonnes visant à supprimer des caractères spécifiques.

Création de Nouvelles Colonnes : Création d'une colonne "Marque" à partir de "Marque / Modele", en conservant tout avant l'espace. Création d'une colonne "Modele" à partir de "Marque / Modele", en conservant tout après l'espace.

Suppression de colonnes : Suppression de la colonne "Marque / Modele" après la création des nouvelles colonnes.

Conversion de Colonnes au Format Float : Conversion des colonnes "Rejets CO2 g/km", "Bonus / Malus", et "Cout energie" au format float.

c. Résultats après traitement

Après traitement, les données sont converties en objets JSON structurés. Chaque objet contient des informations essentielles telles que la marque, le modèle, le bonus/malus, les émissions de CO2, et le coût énergétique. Par exemple, une ligne du fichier **CO2.csv** pourrait être transformée en l'objet JSON suivant :

```
{
  "brand": "AUDI",
  "car_name": "E-TRON SPORTBACK 55 QUATTRO",
  "bonus_malus": -6000,
  "co2_emissions": 0.0,
  "energy_cost": 319.0
}
```

d. Intégration des résultats

Ces objets JSON sont ensuite publiés sur le bus de données Nats Jetstream, conduisant à leur stockage dans la base de données, spécifiquement dans la table **brand_co2_emissions**. Des triggers et des fonctions dans la base de données assurent que les valeurs correctes sont appliquées aux entrées correspondantes dans le catalogue de véhicules, garantissant ainsi que les informations sur les émissions de CO2 et les coûts énergétiques sont à jour et précises.

En conclusion, l'utilisation de Spark pour le traitement du fichier **CO2.csv** permet non seulement un nettoyage et une structuration efficaces des données, mais facilite également leur intégration dans notre système de données, assurant la disponibilité d'informations précises pour des analyses ultérieures et des décisions basées sur les données.

7. Technique de Visualisation

a. D3.JS

Ensemble des programmes et scripts disponibles ici : Github.
Démonstration vidéo disponible ici : *Vidéo*.
Site disponible ici : Dashboard.

i. Introduction

La visualisation des données est un élément essentiel dans notre projet, jouant un rôle crucial dans la manière dont nous interprétons et partageons des insights complexes tirés de nos analyses. Dans un contexte où les données sont abondantes et souvent multidimensionnelles, une visualisation efficace nous permet de présenter ces informations de manière compréhensible et engageante. Elle facilite la détection rapide de tendances, de modèles et d'anomalies, aidant les décideurs à comprendre les nuances et à prendre des décisions éclairées.

D3.js, joue un rôle pivot dans la création de visualisations de données interactives et sophistiquées. C'est une bibliothèque JavaScript puissante et flexible qui permet de manipuler des documents basés sur des données. D3 nous permet de lier des données

complexes à un Document Object Model (DOM), et d'appliquer des transformations basées sur les données pour créer des visualisations riches et interactives. Sa force réside dans sa capacité à donner vie aux données, rendant les interactions avec les visualisations non seulement informatives mais aussi intuitives. Dans notre projet, l'utilisation de D3.js nous aide à créer des graphiques détaillés, ce qui améliore significativement notre capacité à communiquer des insights complexes de manière simple et efficace.

ii. Chaîne de Traitement des données

1. *Chargement des données* :

Cette étape consiste à récupérer les données depuis la base de données via une requête API. Cela constitue la base sur laquelle toutes les analyses et visualisations seront construites.

2. *Identification et préparation des attributs* :

Après avoir obtenu les données, l'accent est mis sur l'examen de leur structure pour identifier les variables clés. Cette analyse est cruciale pour déterminer les caractéristiques pertinentes à visualiser.

3. *Agrégation des données* :

Dans cette phase, les données sont regroupées selon des caractéristiques communes, telles que la marque de la voiture. Des métriques agrégées sont ensuite calculées pour chaque groupe. Par exemple, la moyenne du prix et de la puissance, ainsi que le décompte du nombre de clients par marque de voiture.

4. *Accumulation des données* :

Cette étape implique de compter le nombre de voitures par marque et par genre (Féminin/Masculin). Cela offre une perspective sur la répartition des genres pour chaque marque, enrichissant ainsi la compréhension des tendances et préférences des consommateurs.

5. *Transformation et Filtrage* :

Les données sont transformées pour s'adapter à des formats spécifiques requis par les différents types de graphiques. Cela inclut l'extraction d'attributs spécifiques, comme les émissions de CO2 et le coût énergétique, pour chaque combinaison unique de marque et de modèle de voiture. En parallèle, un filtrage est appliqué pour exclure les données incomplètes ou les valeurs nulles, garantissant ainsi la précision et la fiabilité des visualisations.

En résumé, chaque étape du processus contribue à transformer des données brutes en visualisations significatives, rendant l'ensemble des informations non seulement visuellement attrayantes mais aussi facilement compréhensibles pour l'utilisateur.

iii. Utilisateurs ciblés

Les utilisateurs ciblés sont les vendeurs de la concession automobile, les responsables marketing et toute personne cherchant à comprendre les préférences des clients en matière de véhicules.

Vendeurs de la concession automobile. Besoins et attentes : Ils ont besoin d'informations instantanées et faciles à interpréter pendant les interactions avec les clients. Ils recherchent des données précises sur les modèles de voitures qui pourraient intéresser un client en fonction de son profil (âge, revenu, situation familiale, etc.). Ils veulent pouvoir recommander rapidement des véhicules correspondant aux préférences des clients.

Responsable marketing. Besoins et attentes : Ce groupe cherche à comprendre les tendances générales de l'achat de véhicules, les préférences des différents segments de clientèle, et les caractéristiques des modèles populaires. Ils ont besoin de données agrégées pour orienter les campagnes marketing et cibler efficacement les clients potentiels.

Analystes de données/Gestionnaires. Besoins et attentes : Ils cherchent à plonger plus en profondeur dans les données pour découvrir des relations complexes entre les caractéristiques des véhicules et les comportements d'achat des clients. Ils ont besoin d'outils de visualisation plus avancés pour explorer, analyser et présenter des insights approfondis pour guider les décisions stratégiques de la concession.

Clients potentiels (utilisateurs indirects). Besoins et attentes : Ils cherchent à acheter un véhicule en tenant compte de certaines informations par exemple : les modèles populaires, les caractéristiques clés et les tendances actuelles du marché. Ils pourraient utiliser ces visualisations pour orienter leurs choix ou prendre des décisions éclairées lors de l'achat d'une voiture.

iv. Objectifs de Visualisation et Tâches Utilisateurs

Objectifs de la visualisation :

Aider les vendeurs : Permettre aux vendeurs d'évaluer rapidement les types de véhicules susceptibles d'intéresser les clients visitant la concession.

Assistance au marketing : Envoyer des documentations précises sur les véhicules appropriés aux clients sélectionnés par le service marketing.

Tâches utilisateurs :

Pour les vendeurs : Identifier rapidement le type de véhicule approprié en quelques secondes.

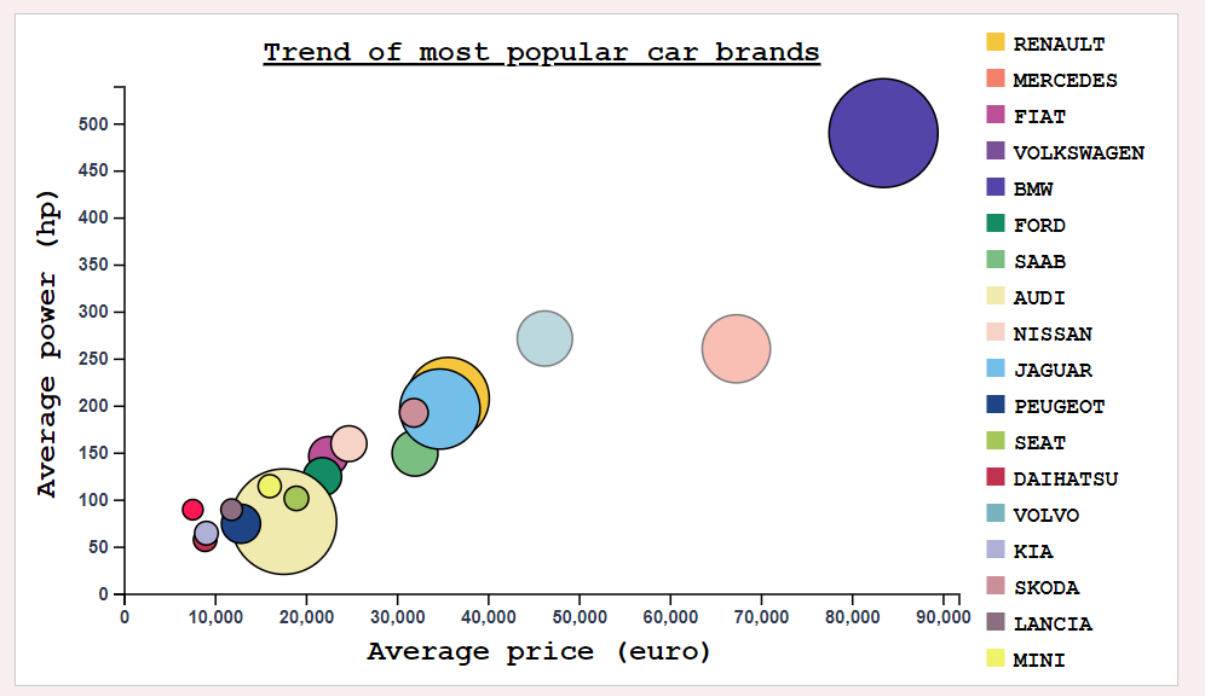
Pour le service marketing : Sélectionner des clients et leur envoyer des documentations précises sur les véhicules qui pourraient les intéresser.

v. Développement des Techniques de Visualisation

Afin de visualiser nos données nous avons choisi trois types de visualisation différents:

1) **Bubble-chart (Diagramme à bulles) - Moyenne Puissance - Prix moyen :**
   Utilisation de trois attributs principaux - la marque de la voiture, la puissance moyenne et le prix moyen. La taille des bulles représente le nombre de clients intéressés par chaque modèle de voiture.

   Les bulles peuvent être utilisées pour représenter différents modèles de voitures, avec la taille proportionnelle au nombre de clients intéressés. Cette visualisation permet aux utilisateurs d'identifier visuellement les modèles les plus populaires en fonction de la puissance et du prix.

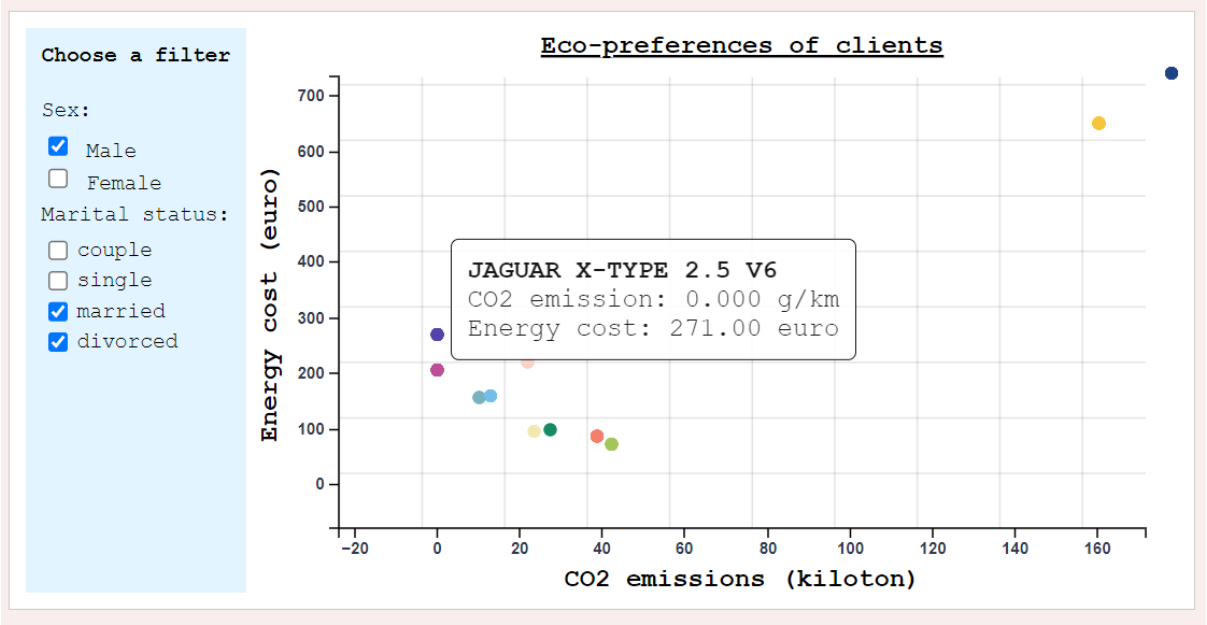


Scatter Plot des “Tendances des marques de voitures les plus populaires”

2) **Scatter plot avec zoom - Émissions CO2 - Coût énergétique :**

   Six attributs principaux - les émissions de CO2 et le coût énergétique des voitures, avec le nom de la marque et du modèle pour le tooltip et le sexe avec la sitiation familiale pour le filtrage.

   Cette visualisation offre la possibilité de zoomer pour explorer en détail une région spécifique du graphique. De plus, l'interaction inclut un tooltip qui affiche des informations détaillées (nom de la marque et du modèle, les valeurs d'émissions CO2 et du coût énergétique) lorsque l'utilisateur survole une donnée spécifique. Il est également possible de filtrer les données par sexe et statut marital pour une analyse plus ciblée.

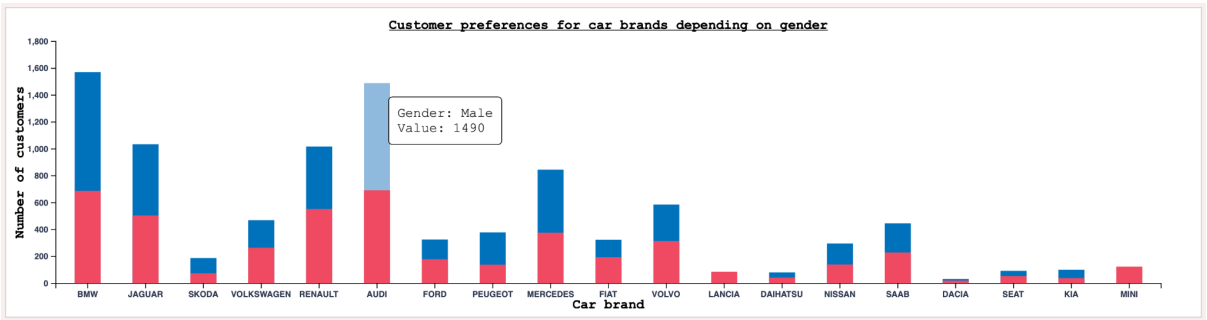


Scatter Plot des "Eco-Preferences of clients"

3) Stack-Bar Chart (Diagramme à barres empilées) - Marque-Sexe :

Les attributs incluent la marque de la voiture, le genre et le comptage des voitures pour chaque marque et chaque sexe.

Cette visualisation empile les barres pour chaque marque, différenciant la répartition des véhicules par genre. L'interaction permet de sélectionner une marque spécifique pour analyser la répartition des sexes. Cette technique offre une comparaison visuelle de la répartition des voitures par marque et par sexe.



Bar plot des “Preferences clients pour les marques de voitures par genre”

vi. Niveaux de Visualisation

Vue globale ou d'ensemble («overview») : Une représentation générale des données qui permet à l'utilisateur d'avoir une idée globale de l'ensemble des informations sans rentrer dans les détails.

Par exemple, le bubble-chart sur la moyenne de la puissance et du prix moyen donne une vision globale des modèles de voitures les plus populaires en fonction de ces

deux attributs. Les bulles représentent les différents modèles avec leur taille indiquant la popularité générale, mais sans afficher de détails spécifiques au survol.

Cette vue globale permet aux utilisateurs de rapidement identifier les tendances dominantes sans avoir à se plonger dans des données spécifiques.

Vue détaillée ou spécifique : Une représentation plus approfondie, permettant à l'utilisateur d'explorer des détails spécifiques ou des informations plus ciblées à partir de la vue globale. Cette vue offre un contexte plus détaillé ou une analyse approfondie des données.

Par exemple, le scatter plot avec la possibilité de zoomer sur les émissions de CO2 et le coût énergétique fournit une vue plus détaillée lorsque l'utilisateur souhaite examiner de plus près une région spécifique du graphique. Le tooltip qui affiche le nom de la marque et du modèle permet également d'accéder à des informations spécifiques au survol.

De même, le stack-bar chart sur la répartition des voitures par marque et par sexe offre une vue plus détaillée lorsqu'une marque spécifique est sélectionnée pour analyser plus en profondeur la répartition par genre.

vii. Conclusion

Les techniques de visualisation développées présentent un fort potentiel pour répondre aux besoins des vendeurs, des responsables marketing et des analystes au sein de la concession automobile. Leur conception variée et leur capacité à offrir à la fois une vue globale et des analyses approfondies promettent d'apporter des réponses précieuses.

Bien que ces visualisations n'aient pas encore été présentées aux utilisateurs, leur conception en fonction des objectifs du projet et des besoins spécifiques des différents acteurs suggère un fort impact potentiel. Les retours d'utilisateurs anticipés pourraient aider à affiner ces visualisations pour répondre encore mieux aux attentes.

Des améliorations futures pourraient inclure des fonctionnalités supplémentaires pour une exploration plus approfondie des données (le filtrage plus avancé par exemple par revenus ou le taux d'endettement) ou des ajustements basés sur des retours d'utilisateurs réels.

En somme, bien que les visualisations n'aient pas encore été déployées, leur conception répond aux besoins identifiés et promet d'apporter des réponses précieuses une fois présentées aux utilisateurs.

b. Grafana (Bonus)

Ensemble des programmes et scripts disponibles ici : Github.
Démonstration vidéo disponible ici : Vidéo.
Site disponible ici : Dashboard. (capture d'écran)

i. Introduction

La visualisation des données est d'une importance essentielle dans notre projet d'analyse de données pour le secteur automobile, permettant une compréhension intuitive des tendances et une communication efficace des résultats. Grafana, en tant que plateforme de visualisation open source, joue un rôle clé en offrant une flexibilité totale pour créer des tableaux de bord interactifs. Grâce à ses fonctionnalités d'exploration des données, d'alertes en temps réel et à sa polyvalence pour intégrer différentes sources de données, Grafana permet une prise de décision informée et une détection rapide des anomalies, contribuant ainsi à maximiser l'impact des insights issus de l'analyse de données.

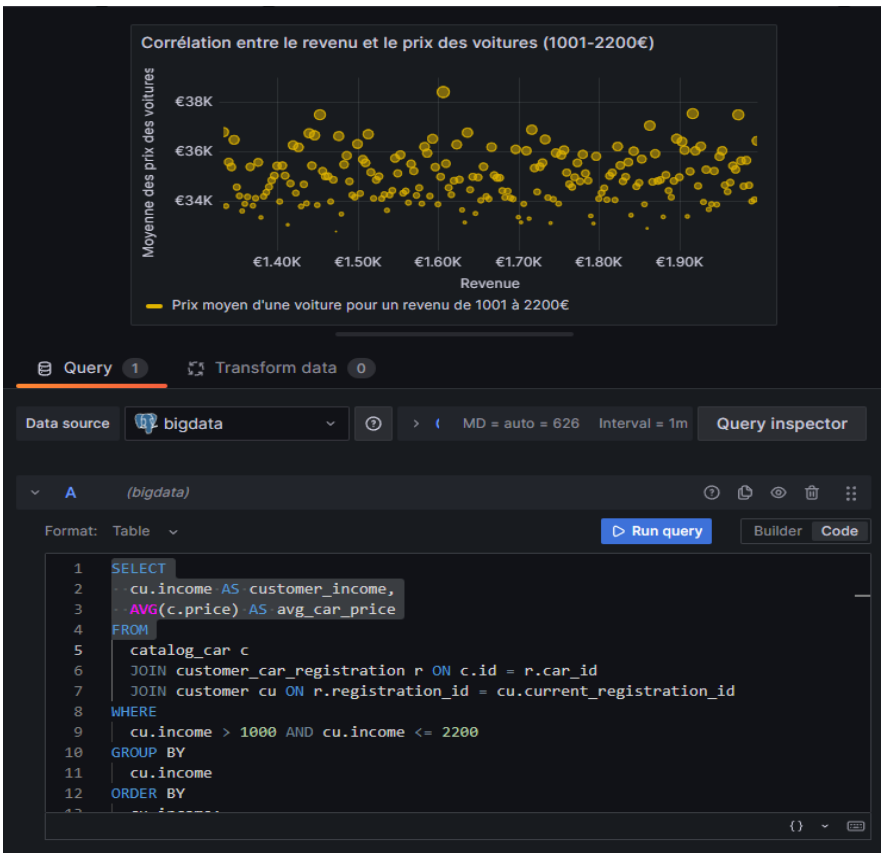
ii. Chaîne de Traitement des Données

Le processus de transformation des données brutes en visualisations dans Grafana s'appuie sur un data source PostgreSQL configuré. Les étapes incluent l'extraction des données via des requêtes SQL, le nettoyage pour éliminer les anomalies, la transformation pour l'agrégation et la préparation des données, puis la création de tableaux de bord dans Grafana. Les requêtes SQL jouent un rôle central, facilitant la manipulation des données brutes pour une représentation visuelle significative dans Grafana, offrant ainsi un moyen structuré d'obtenir des insights pertinents à partir des données stockées dans la base PostgreSQL.

iii. Développement des Techniques de Visualisation

Technique de Visualisation 1: Corrélation Revenu-Prix (Scatter Plot)

Un Scatter Plot illustre la corrélation entre le revenu des clients (plus de 2200€) et le prix moyen des voitures. Chaque point représente un groupe de clients avec le revenu sur l'axe des x et le prix moyen des voitures sur l'axe des y.



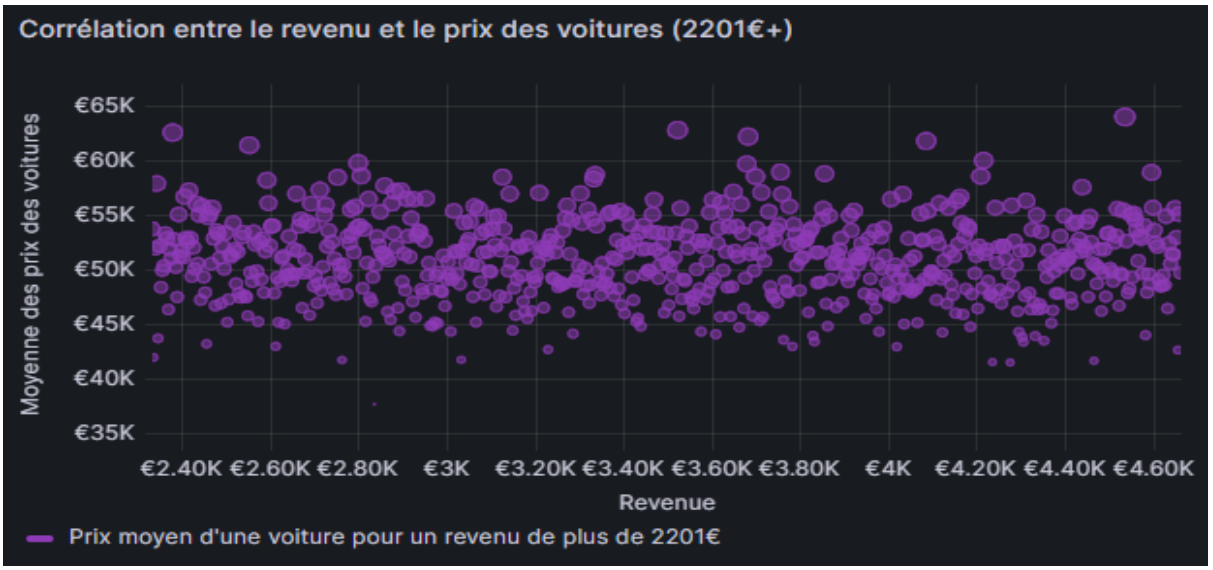
Exemple de visualisation pour “Corrélation entre le revenu et le prix des voitures (1001-2200€)”

Attributs:

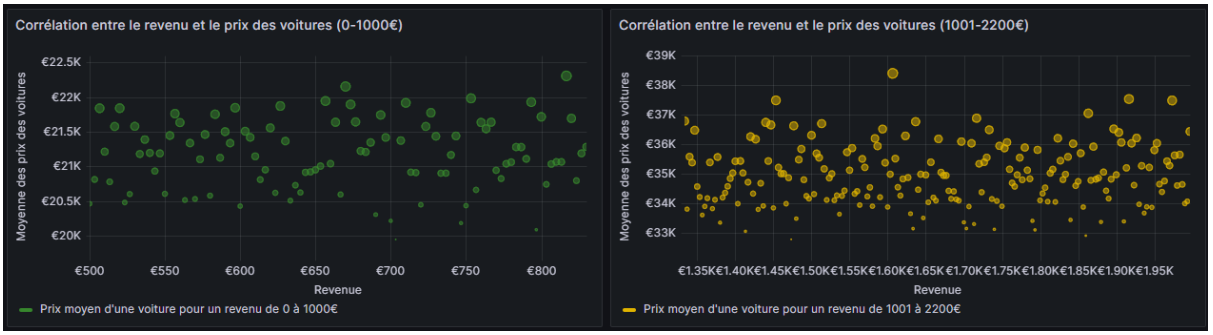
- `customer_income`: Revenu du client.
- `avg_car_price`: Prix moyen des voitures pour chaque groupe de revenu.

Interaction avec les données:

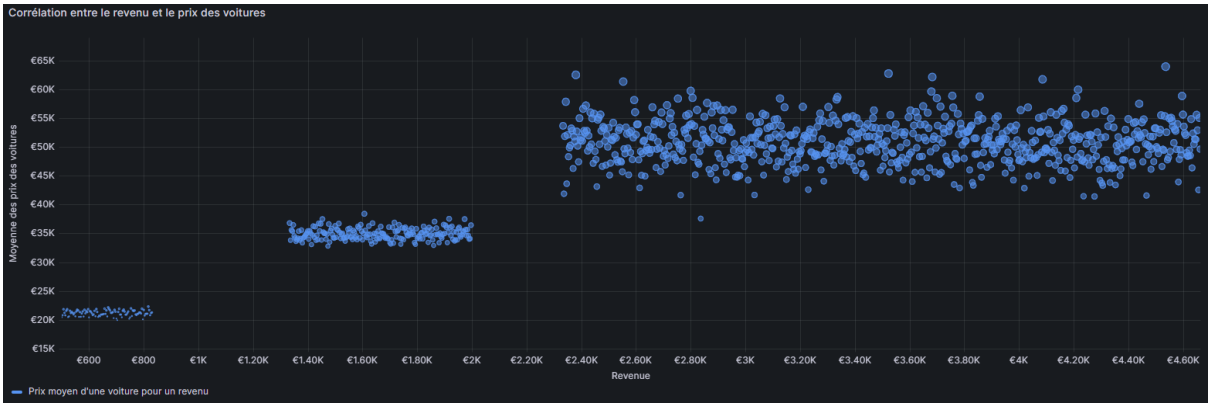
- Navigation: Exploration des tranches de revenu pour voir l'impact sur le prix des voitures.
- Sélection: Mise en évidence d'une tranche de revenu spécifique pour la comparaison (0 à 1000€, 1001 à 2200€ & +2201€).
- Filtres: Filtrage par intervalle de revenu pour affiner la visualisation.



Scatter de plot de la “Corrélation entre le revenu et le prix des voitures (2201€+)”



Les autres tranches de la “Corrélation entre le revenu et le prix des voitures”



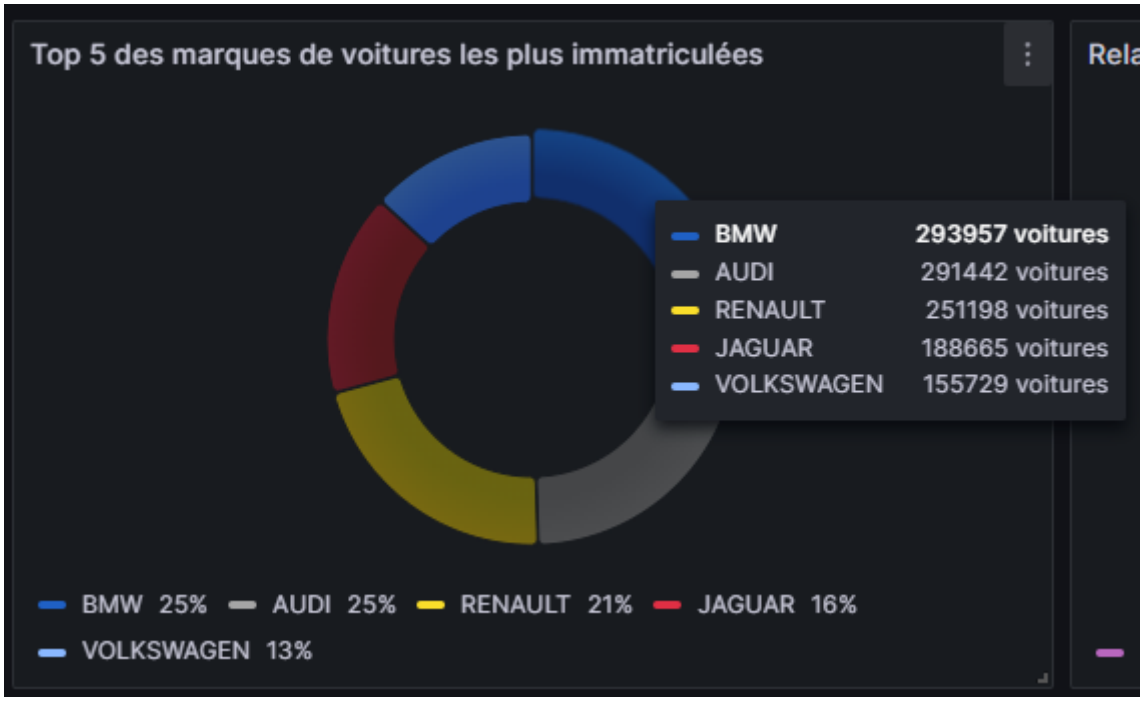
Démonstration de la lisibilité en tranches

Technique de Visualisation 2: Top 5 des Marques Immatriculées (Pie Chart)

Un Pie Chart présente les cinq marques de voitures les plus immatriculées. Chaque segment représente une marque, la taille du segment étant proportionnelle au nombre total d'immatriculations.

Attributs:
- `brand`: Marque de la voiture.
- `registration_count`: Nombre total d'immatriculations.

Interaction avec les données:
- Navigation: Identification rapide des marques les plus populaires.
- Sélection: Choix d'une marque pour plus d'informations.
- Filtres: Filtrage possible par critères spécifiques, comme le nombre minimum d'immatriculations.



Pie chart du “Top 5 des marques de voitures les plus immatriculées”

Technique de Visualisation 3: Revenu et Âge par Marque (Bar Chart)

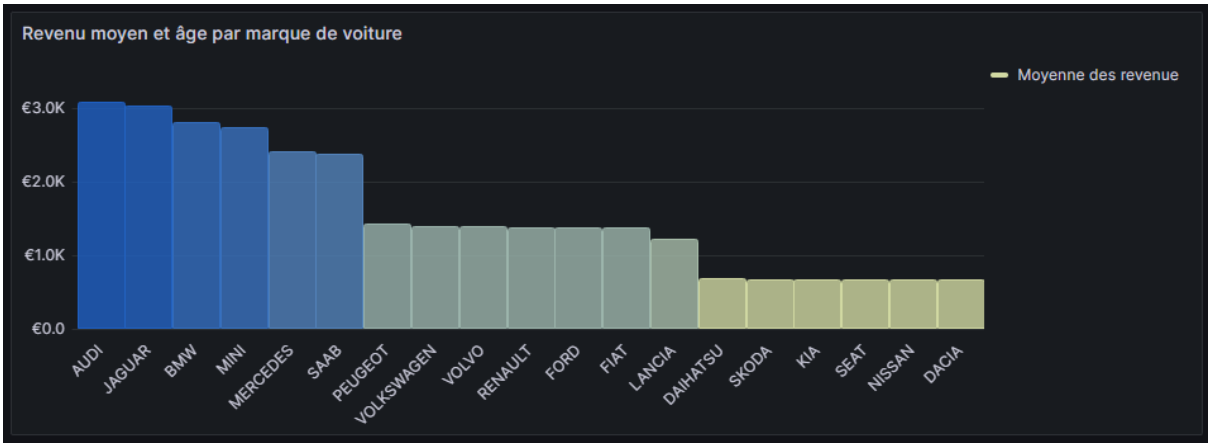
Un Bar Chart représente le revenu moyen et l'âge des clients pour chaque marque de voiture. Chaque barre comporte deux segments, un pour le revenu moyen et l'autre pour l'âge.

Attributs:
- `brand`: Marque de la voiture.
- `avg_income`: Revenu moyen des clients de la marque.
- `avg_age`: Âge moyen des clients de la marque.

Interaction avec les données:
- Navigation: Identification des marques liées à des revenus plus élevés ou des clients plus jeunes.

- Sélection: Choix d'une marque pour plus d'informations.
- Filtres: Filtrage possible par critères comme le revenu minimum ou l'âge minimum.

Ces visualisations offrent une exploration interactive des données, facilitant la compréhension des tendances liées aux revenus, aux marques de voitures et aux caractéristiques des clients.



Bar chart du “Revenu moyen et âge par marque de voiture”

iv. Niveaux de Visualisation

Pie Chart - Vision Globale

- Objectif: Identifier rapidement les marques de voitures les plus populaires.
- Exemple: Le Pie Chart montre que la marque A est la plus immatriculée, tandis que les marques B, C, D et E complètent le top 5.
- Implémentation: Aucune interaction complexe nécessaire, une simple observation suffit.

Scatter Plot - Contexte Détaillé

- Objectif: Explorer la corrélation spécifique entre le revenu et le prix des voitures.
- Exemple: En sélectionnant une tranche de revenu spécifique, l'utilisateur peut voir comment les prix des voitures varient dans cette tranche.
- Implémentation: Interactivité avec des points de données individuels, possibilité de filtrer par intervalle de revenu.

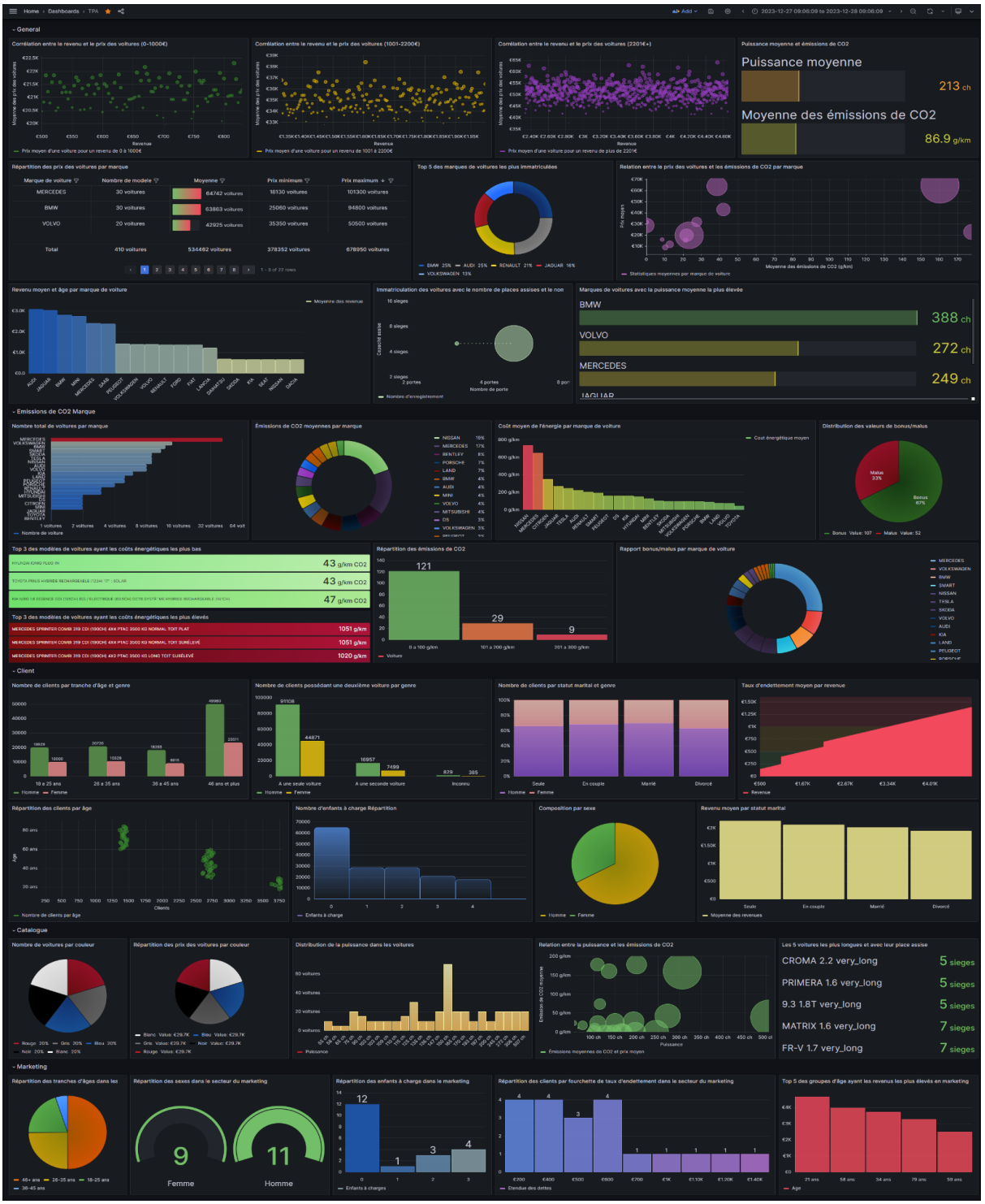
Bar Chart - Vision Globale et Contexte Détaillé

- Objectif: Comprendre la relation entre le revenu moyen et l'âge des clients pour chaque marque.
- Exemple: Le Bar Chart offre une vue globale des différences entre les marques en termes de revenu moyen et d'âge. En sélectionnant une barre, l'utilisateur peut se plonger dans les détails de cette marque spécifique.
- Implémentation: Les barres fournissent une vision globale, tandis que la sélection offre un contexte détaillé pour une marque spécifique.

v. Chargement de Données Indépendantes

Grafana démontre une flexibilité et une évolutivité significatives en chargeant des ensembles de données indépendants de l'application. En utilisant PostgreSQL comme source de données principale et en effectuant des requêtes SQL directes dans Grafana, l'outil offre une flexibilité totale pour interroger et visualiser les données. Sa capacité à intégrer diverses sources de données, à créer des requêtes ad hoc, et à évoluer avec des volumes importants de données garantit une adaptabilité continue aux besoins changeants du concessionnaire, tout en permettant une exploration dynamique et une gestion optimale des performances.

vi. Présentation du prototype Grafana



Vue globale du prototype Grafana

Le prototype Grafana créé pour l'exploration des données offre une interface interactive permettant aux utilisateurs d'analyser les ventes de voitures sous différents angles. Le premier tableau de bord présente une vue d'ensemble des ventes, tandis que le deuxième se concentre sur le profil client. Le troisième se penche sur l'analyse des émissions de CO2. Les graphiques interactifs, tels que les diagrammes de dispersion, les

tableaux récapitulatifs et les diagrammes à secteurs, offrent une exploration détaillée. Les utilisateurs peuvent filtrer les données, comparer visuellement des catégories spécifiques, et explorer les détails en cliquant sur des éléments spécifiques. En résumé, ce prototype facilite l'analyse approfondie des données et la prise de décisions éclairées dans le domaine de la vente de voitures.

vii. Conclusion

Les techniques de visualisation mises en œuvre dans notre projet d'analyse de données pour le secteur automobile, via Grafana, ont considérablement enrichi la compréhension des tendances et des relations clés. L'utilisation de Scatter Plots pour explorer la corrélation entre le revenu et le prix des voitures offre une vision détaillée, tandis que les Pie Charts présentent de manière succincte les marques les plus immatriculées. Les Bar Charts, quant à eux, fournissent une vue globale et détaillée des revenus moyens et de l'âge des clients par marque. Ces visualisations, intégrées dans un prototype Grafana interactif, facilitent une analyse approfondie des ventes de voitures. Les retours initiaux des utilisateurs mettent en avant la facilité d'exploration et la clarté des insights obtenus. De plus, l'extension des capacités de visualisation pour inclure des analyses prédictives pourrait fournir des perspectives encore plus puissantes pour les décisions futures dans le domaine de la vente de voitures.

8. Analyse de données avec des outils de machine learning

Ensemble des programmes et scripts disponibles ici : Github.
Démonstration vidéo disponible ici : Vidéo.

a. Introduction

Dans notre projet, nous adoptons le Machine Learning comme un outil fondamental pour analyser et interpréter les données de manière approfondie. Cette technologie nous permet de mieux comprendre et prédire les tendances du marché et les préférences des clients. Le ML nous aide à analyser les données des clients pour offrir une expérience plus personnalisée, en suggérant des véhicules et des services adaptés à leurs besoins et préférences individuels. On l'utilise aussi pour affiner nos stratégies marketing, en ciblant les campagnes sur les segments de clients les plus réceptifs.

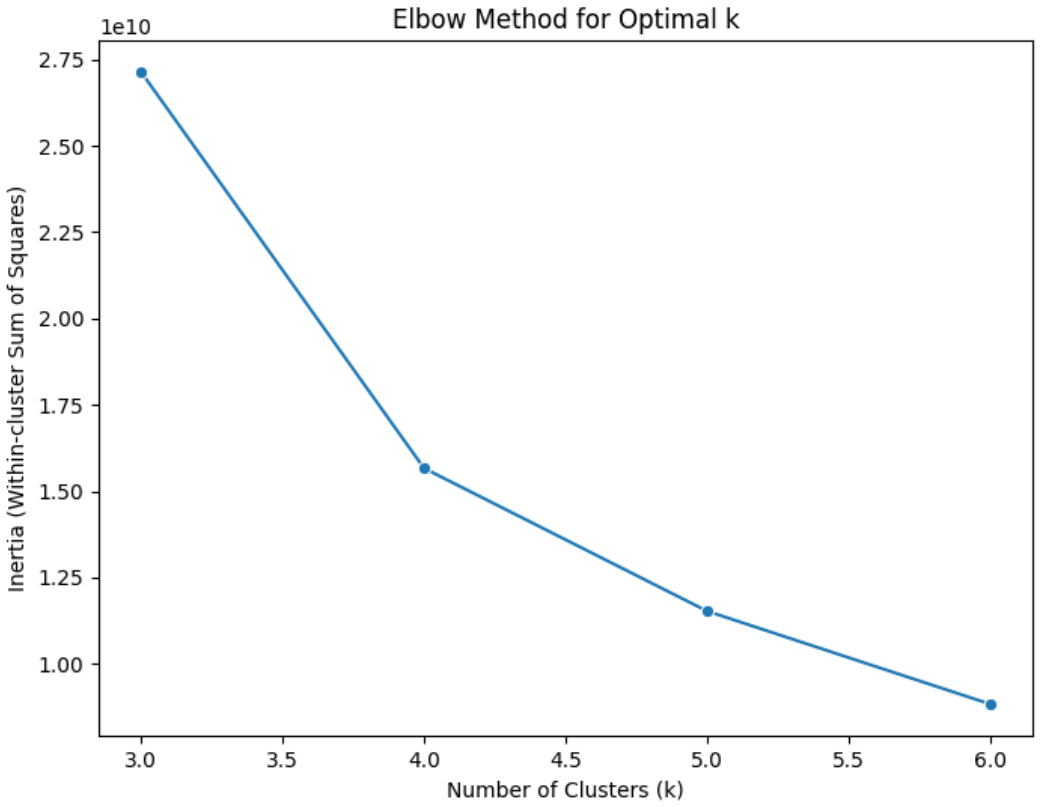
b. Techniques et Outils Utilisés

Pour résoudre le problème de classification des données, une approche basée sur l'utilisation de modèles de classification a été adoptée. Différentes techniques ont été explorées et plusieurs modèles ont été entraînés pour évaluer leur performance sur les données spécifiques du problème.

- MLP Classifier (Réseau de Neurones)
- Random Forest
- Bayesian-Gaussian Mixture

- Mean Shift
- Support Vector Machine (SVM)

Suite à l'exploration de ces différentes techniques avec l'aide de scikit-learn (sklearn pour Python), l'algorithme Random Forest a émergé comme le choix optimal pour la classification des données. La méthode de K-means de scikit-learn a été utilisée pour identifier le nombre optimal de clusters (6 dans notre cas), améliorant ainsi la précision globale du modèle.



c. Processus d'Analyse de Données

Nettoyage des Données :

Cette procédure est exécutée avant d'alimenter la base de données. Donc, il n'y a pas besoin de la refaire. Elle visait à identifier et traiter les valeurs manquantes, les valeurs aberrantes, et d'autres anomalies pour garantir la fiabilité des données utilisées pour l'entraînement des modèles.

Encodage des Données :

Le fichier ***'customers_treater.py'*** a été développé spécifiquement pour gérer l'encodage des données. Il a pris en charge la conversion des caractéristiques catégorielles en format numérique afin de rendre les données compatibles avec les algorithmes d'apprentissage automatique. L'encodage est essentiel pour permettre aux modèles de comprendre et de traiter les informations de manière efficace.

d. Modèles et Connaissances Générés

L'utilisation du modèle ***RandomForestClassifier*** dans le cadre de notre objectif de classification des catégories de voitures, que ce soit pour prédire le modèle spécifique de la voiture ou sa catégorie générale. Ce choix d'algorithme s'appuie sur la puissance des ensembles d'arbres de décision pour prendre des décisions de classification robustes.

Pour optimiser les performances du modèle, nous avons réalisé une recherche approfondie des hyperparamètres à l'aide de ***GridSearchCV***. Les hyperparamètres clés, tels que le nombre d'estimateurs, la profondeur maximale de l'arbre, le nombre minimum d'échantillons requis pour la division, et le nombre minimum d'échantillons dans une feuille, ont été finement ajustés pour garantir une configuration optimale du modèle.

Cette approche a permis de créer un modèle RandomForestClassifier adapté à la complexité des données, capable de généraliser efficacement aux différentes catégories de voitures.

e. Résultats et Interprétation

Les résultats ont une efficacité significative dans la classification des voitures, que ce soit pour prédire le modèle spécifique d'une voiture ou sa catégorie globale. Le modèle visant à prédire le modèle spécifique affiche une précision de 76%, tandis que celui pour la catégorie atteint une précision de 78%.

Notre modèle de prédiction de catégorie est subtilement meilleur en précision, cela veut dire que nous avons moins de faux positifs que de faux négatifs. Lorsque notre modèle classe une voiture dans une catégorie particulière, il a tendance à avoir raison avec une fréquence plus élevée, minimisant ainsi les cas où une voiture est incorrectement attribuée à une catégorie.

	precision	recall	f1-score	support
0	0.87	0.90	0.88	11345
1	0.78	0.69	0.74	4009
2	0.72	0.74	0.73	5485
3	0.55	0.52	0.53	2877
4	0.55	0.41	0.47	2579
5	0.63	0.70	0.66	5908

Notre modèle de prédiction de voiture spécifique se trouve dans le même cas, présentant une précision légèrement meilleure.

Classification Report:

	precision	recall	f1-score	support
0	0.91	0.90	0.90	4834
1	0.94	0.56	0.70	115
2	0.51	0.68	0.59	417
3	0.82	0.82	0.82	4665
4	0.53	0.51	0.52	181
5	0.59	0.67	0.63	298
6	0.54	0.70	0.61	1269
7	0.59	0.66	0.62	1178
8	0.77	0.82	0.80	3256
9	0.64	0.64	0.64	338
10	0.75	0.62	0.68	219
11	0.68	0.92	0.78	688
12	0.97	0.84	0.90	1970
13	0.96	0.62	0.75	284
14	0.61	0.60	0.61	187
15	0.69	0.65	0.67	530
16	0.52	0.61	0.56	323
17	0.68	0.66	0.67	1323
18	0.58	0.64	0.61	1194
19	0.71	0.57	0.63	552
20	0.77	0.62	0.69	1923
21	0.83	0.96	0.89	1555
22	0.58	0.63	0.60	339
23	0.77	0.67	0.72	557
24	0.61	0.61	0.61	522
25	0.79	0.56	0.65	99
26	0.65	0.65	0.65	1366
27	0.78	0.60	0.68	2025
accuracy			0.76	32207
macro avg	0.71	0.68	0.68	32207
weighted avg	0.76	0.76	0.76	32207

Nous sommes également capables de donner une importance aux features. Cela peut-être important si le client ne sent pas satisfait de la recommandation que l'on lui donne pour générer une autre.

```
---------- Feature Importance Analysis ----------
Top 5 Important Features:
              Feature  Importance
2  dependent_children    0.272659
1           debt_rate    0.202995
5              income    0.191351
6      marital_status    0.143330
4      has_second_car    0.137766
```

f. Application aux Clients Sélectionnés par le Service Marketing

Les modèles ont été appliqués aux clients dans le fichier marketing. Le script **results.py** se charge de cette tâche. Une catégorie et une voiture sont attribuées à chaque client, comme vous pouvez le constater sur l'image.

	Age	Debt_rate	Dependent_ch	Gender	Has_second_c	Income	Marital_status	Car_category_	Car_id
	25	971	2	0	0	3236.67	2	2	233
	55	455	2	1	0	1516.67	2	3	56
	25	162	1	0	0	540	2	2	100
	24	886	0	0	0	2953.33	1	0	311
	22	722	2	1	0	2406.67	2	1	281
	67	1237	1	1	1	4123.33	2	0	312
	44	924	1	0	0	3080	2	2	238
	81	922	0	1	0	3073.33	2	1	201

L'efficacité (F1-Score) de nos modèles peut être interprétée comme la capacité de nos algorithmes à formuler des recommandations pertinentes qui suscitent l'intérêt des clients et les incitent à effectuer un achat.

g. Conclusion et Perspectives Futures

Analyse des résultats

L'analyse des résultats a montré que nos modèles sont capables de formuler des recommandations pertinentes. L'utilisation de métriques telles que la précision et le rappel nous a permis d'identifier les forces et les faiblesses de nos modèles, afin de les améliorer de manière continue.

Impact sur la personnalisation des offres

L'intégration de ces modèles dans des scénarios marketing concrets pourrait avoir un impact significatif sur la personnalisation des offres pour les clients. Cela permettrait de proposer une expérience client plus adaptée aux besoins et aux intérêts de chacun, ce qui augmenterait la probabilité d'achat et la satisfaction des clients.

Impact sur les ventes

Bien que nous ne disposions pas de données de ventes pour évaluer directement l'impact financier, les performances solides de nos modèles suggèrent qu'ils ont le potentiel d'influencer positivement les décisions des clients, et donc les ventes potentielles.

9. Conclusion générale

Le projet a conduit à des avancées significatives dans la compréhension et la gestion des données relatives au secteur automobile. Nos techniques de visualisation, en particulier celles intégrées dans Grafana, ont permis une analyse approfondie et intuitive des tendances et des relations clés. Les modèles de Machine Learning, notamment le RandomForestClassifier, ont démontré une efficacité remarquable dans la classification des catégories de voitures, avec des précisions respectives de 76% et 78% pour les modèles spécifiques et les catégories de voitures. Ces résultats suggèrent un potentiel significatif pour influencer positivement les décisions des clients et, par conséquent, les ventes potentielles.

Nous avons dû surmonter divers défis, notamment en ce qui concerne le traitement et l'analyse des données. Des techniques avancées ont été utilisées pour le nettoyage des données et leur préparation pour le Machine Learning, y compris l'encodage des caractéristiques catégorielles. Les ajustements des hyper-paramètres des modèles ont été une étape cruciale pour optimiser les performances et la précision des prédictions.

L'accent sera mis sur l'amélioration continue des modèles existants et l'exploration de nouvelles techniques de Machine Learning. L'intégration de ces modèles dans des scénarios marketing concrets pourrait avoir un impact significatif sur la personnalisation des offres pour les clients, augmentant ainsi la probabilité d'achat et la satisfaction des clients. De plus, l'extension des capacités de visualisation pour inclure des analyses prédictives pourrait fournir des perspectives encore plus puissantes pour les décisions futures.

Le projet a été une opportunité exceptionnelle de développement professionnel et personnel. Travailler sur des données complexes et mettre en œuvre des solutions innovantes a renforcé nos compétences en analyse de données et en Machine Learning. La collaboration au sein de l'équipe a permis un échange de connaissances et une croissance collective, renforçant notre capacité à relever des défis complexes et à fournir des solutions efficaces.

En résumé, ce projet a non seulement apporté des avantages tangibles à l'entreprise en termes de compréhension des données et d'efficacité des ventes, mais a également contribué à notre croissance en tant qu'analystes de données et professionnels du secteur automobile.

10. Références et Bibliographie

- Apache Spark : Un moteur de traitement de données unifié pour le traitement par lots et en temps réel.
- Apache Hadoop : Un cadre logiciel pour le stockage distribué et le traitement de grandes quantités de données sur des clusters informatiques.
- Hadoop Installation on Linux Systems par Keegan Fernandes : Un guide pour installer Hadoop sur des systèmes Linux.
- Setting Up a Spark Standalone Cluster on Docker in Layman Terms par Marin Aglić : Instructions pour configurer un cluster Spark en mode autonome sur Docker.
- PostgreSQL : Un système de gestion de base de données relationnelle et objet.
- How to Run PostgreSQL and PgAdmin Using Docker par Raouf Makhlouf : Un tutoriel pour exécuter PostgreSQL et PgAdmin à l'aide de Docker.
- D3 by Observable. JavaScript library for bespoke data visualization.

11. Annexes

a. Vidéo de présentation de votre projet

- CHIAPPE : Data Lake - Présentation
- BONE : Spark - Démo
- BONE : Data Analysis - Présentation & Démo
- AGLAE : Data Visualization (Grafana) - Vidéo
- AGLAE : Data Visualization (API) - Vidéo
- VINCENT & LAPSHINA : Data Visualization (D3.js) - Vidéo

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

- SQL Files : Ici
- Data Uploader : Ici
- Déploiement de l'infrastructure : Ici
- Page administration de la BDD : Ici (user: prof, password: BigDataMBDS)

c. Dossier contenant les scripts et programmes Hadoop Map Reduce

- Data Treater : Ici
- Data Treater (Spark only) : Ici

d. Dossier contenant les scripts et programmes de visualisation de données

- D3.Js : Ici (hébergé ici)

- Grafana : Ici (hébergé ici)

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

- Data Analyzer & Classifier : Ici
- RF Classifier Model : Ici
- Spectral RF Model : Ici
- Spectral RF Model (Optimized) : Ici

MERCI
Bonne Année !