



UNIVERSITÉ PARIS NANTERRE

Rapport de Projet

AutoPredict

Membres du projet :

X. Frédéric

R. Yann

R. Jérémy

7 avril 2025

Table des matières

1	Présentation de AutoPredict	2
1.1	Problématique	2
1.2	Solution	2
2	Caractéristiques Principales	3
2.1	Interface Conviviale	3
2.2	Sources Fiables	3
3	Architecture métier	4
3.1	Frontend	4
3.2	Backend	4
3.3	Base de données	4
4	Architecture distribuée	5
4.1	Application Hosting	5
4.2	Database Hosting	5
5	Pratiques de Collaboration et de DevOps	6
5.1	Project Management	6
5.2	Versionnement	6
5.3	Intégration Continue et Déploiement Continu	6
5.4	Maintenabilité du code	6
5.5	Qualité du code	6
6	Partie Data Analytique	7
6.1	Source de données	7
6.2	Nettoyage et manipulation des Données	7

Chapitre 1

Présentation de AutoPredict

1.1 Problématique

Le choix d'une voiture peut s'avérer complexe pour un acheteur, en particulier lorsque de nombreux critères entrent en jeu : budget, type de motorisation, consommation, puissance, style, marque, etc. Les plateformes existantes n'offrent pas toujours une expérience personnalisée ou intuitive pour explorer l'offre de véhicules selon ses préférences réelles. Par ailleurs, les vendeurs ou les analystes souhaitent mieux comprendre les facteurs qui influencent le prix d'un véhicule, et optimiser leur stratégie de vente.

1.2 Solution

AutoPredict est une plateforme web intelligente permettant d'exploiter une base de données automobile pour proposer deux fonctionnalités principales :

- **Recherche assistée de modèles** : à partir d'un budget donné et de certaines préférences (ex. type de transmission, puissance, taille...), l'utilisateur reçoit une liste de véhicules correspondant à ses besoins.
- **Estimation de prix** : à partir des caractéristiques sélectionnées (ex. année, style, consommation...), l'utilisateur obtient une estimation de la fourchette de prix des véhicules correspondants.

Ces fonctionnalités sont rendues possibles grâce à une architecture combinant analyse de données, machine learning, backend intelligent et une interface frontend intuitive.

Chapitre 2

Caractéristiques Principales

2.1 Interface Conviviale

L'interface utilisateur est conçue avec ReactJS pour offrir une navigation fluide et interactive. L'accent est mis sur l'ergonomie, la clarté des résultats et la facilité d'utilisation même pour des utilisateurs non-experts.

2.2 Sources Fiables

Les données utilisées proviennent de datasets publics sur les voitures, incluant des caractéristiques techniques (puissance, consommation, taille, style), économiques (prix, popularité), et temporelles (année de sortie).

Chapitre 3

Architecture métier

3.1 Frontend

Le frontend est développé en ReactJS. Il intègre des composants interactifs permettant à l'utilisateur de :

- Rechercher des modèles de voitures correspondant à ses critères et à son budget
- Estimer la valeur d'un véhicule en fonction de caractéristiques spécifiques
- Visualiser graphiquement les résultats obtenus (filtres, comparateurs, graphiques de prix, etc.)

L'interface dialogue avec le backend via une API REST.

3.2 Backend

Le backend est conçu en Python à l'aide du framework Flask. Il gère :

- L'accès à la base NoSQL contenant les données automobiles
- L'exécution des modèles de machine learning pour la recommandation de véhicules et l'estimation des prix
- L'interface avec le frontend via une API structurée

Les requêtes utilisateurs sont traitées dynamiquement pour retourner des résultats adaptés et rapides.

3.3 Base de données

La base de données utilisée est de type NoSQL, permettant une flexibilité dans la gestion des formats de données hétérogènes typiques du domaine automobile.

Chapitre 4

Architecture distribuée

4.1 Application Hosting

Le projet est conteneurisé afin de faciliter le déploiement, la scalabilité et la portabilité. Docker est utilisé pour packager les composants.

4.2 Database Hosting

La base NoSQL est hébergée dans un environnement compatible cloud. Elle stocke les jeux de données enrichis et traités, accessibles via API.

Chapitre 5

Pratiques de Collaboration et de DevOps

5.1 Project Management

Le projet est géré en équipe de trois membres : Frédéric, Yann et Jérémy. Le suivi des tâches se fait de manière collaborative autour d'outils de gestion agile.

5.2 Versionnement

L'ensemble du code source est versionné via Git, avec des dépôts organisés pour le frontend, le backend et les notebooks d'analyse/ML.

5.3 Intégration Continue et Déploiement Continu

Des pipelines CI/CD seront mis en place pour automatiser les tests, le linting, et le déploiement sur l'environnement de développement.

5.4 Maintenabilité du code

L'utilisation de conteneurs, de frameworks standards (Flask, React) et de pratiques de développement modulaire assure la maintenabilité du projet.

5.5 Qualité du code

Le code est documenté, typé et validé avec des outils de linting et des tests unitaires, notamment sur les scripts de preprocessing et les modèles ML.

Chapitre 6

Partie Data Analytique

6.1 Source de données

Les jeux de données collectés concernent des véhicules avec différentes variables : prix, année, marque, puissance, consommation, popularité, style, etc.

6.2 Nettoyage et manipulation des Données

Les données brutes ont été nettoyées pour supprimer les doublons, combler les valeurs manquantes, et normaliser les unités. Des transformations ont été réalisées sur les variables temporelles et catégorielles.