

Plateforme d'Analyse & de Prédiction Automobile

*Une solution intelligente pour piloter les décisions
commerciales automobiles.*

Le Problème

Les concessions et vendeurs automobiles manquent d'outils simples pour :

- Comparer rapidement des véhicules
- Analyser les prospects
- Prédire l'intention d'achat
- Exploiter efficacement leurs données internes

➡ **Résultat : perte de prospects et décisions moins efficaces.**

L'Opportunité

- Le secteur automobile génère des volumes de données énormes
- Les équipes commerciales n'ont pas d'outil adapté et intuitif
- Les solutions actuelles sont complexes, chères ou inexistantes

 **Une vraie opportunité pour un outil simple, moderne et prédictif.**

Notre Solution

Une plateforme Web qui permet :

- Comparaison instantanée des véhicules
- Analyse comportementale des prospects
- Modèle prédictif d'intention d'achat
- Pipeline de données automatisé
- Déploiement en un clic via Docker

 **Une solution complète, rapide, et déjà opérationnelle.**

Fonctionnalités clés

1. Comparateur de véhicules

Analyse instantanée des caractéristiques, prix, options.

2. Analyse des prospects

Âge, réduction appliquée, campagnes marketing vues.

3. Prédiction d'achat

Modèle ML intégré → score clair et interprétable.

Structure du projet

PROJET_STREAMLIT_LINUX/

— app.py	→ application Streamlit
— deploy.sh	→ script de lancement (Docker / app)
— Dockerfile	→ image Docker de l'application
— mosef_marketing.csv	→ jeu de données local
— .env	→ variables de configuration
— .gitignore	→ fichiers à ignorer par Git
— pyproject.toml	→ configuration Poetry / dépendances
— poetry.lock	→ versions figées des paquets
— README.md	→ documentation du projet
— data/	→ (dossier des données, ex: data.csv)
— data_collector/	
— bin/	
— get_data.sh	→ script de récupération des données
— run.sh	→ script d'exécution (optionnel)
— conf/	
— collector.conf	→ configuration du data collector

Pipeline de données (automatisé)

Étapes automatisées :

- Le script `get_data.sh` télécharge les données depuis Google Drive
- Le fichier `data.csv` est placé dans `/data`
- L'application charge automatiquement les données à son démarrage
- La visualisation + prédiction sont mises à jour instantanément

 **Pas de manipulations manuelles → flux fiable et reproductible.**

Comment lancer l'application ?

1. Récupération des données

```
bash data_collector/bin/get_data.sh
```

2. Lancement de l'application

```
bash deploy.sh
```

3. Accès à l'interface

👉 Ouvrir : <http://localhost:8501>

➡ L'utilisateur n'a rien à installer : tout tourne dans Docker.

Technologie utilisée

- **Streamlit** → interface web propre et moderne
 - **Python / Machine Learning** → modèle prédictif intégré
 - **Docker** → déploiement en un clic
 - **Pipeline automatisé** → ingestion des données Google Drive
 - **Architecture modulaire** → facile à maintenir et à faire évoluer
- ➡ **Un choix technologique adapté à la vitesse, la simplicité et la scalabilité.**

Valeur Business

Pour les concessions :

- Décisions plus rapides
- Meilleur ciblage commercial
- Augmentation du taux de conversion
- Gain de temps dans l'analyse
- Outil simple & sans formation

 **Impact direct sur les ventes.**

Conclusion

- Le besoin est réel
- La solution existe
- Le produit est déjà fonctionnel
- L'impact business est clair
- La vision est ambitieuse

→ Une opportunité d'investissement solide.