

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
|__ data.csv                               → jeu de données principal
|__ deploy.sh                             → script de lancement
|__ Dockerfile                            → image Docker de l'application
|__ mosef_marketing.csv                   → données marketing
|__ poetry.lock                           → versions figées des paquets
|__ pyproject.toml                         → configuration Poetry / dépendances
|__ Presentation.pdf                      → cette présentation
|__ README.md                             → documentation du projet
|__ .env                                   → variables de configuration
|__ .gitignore                            → fichiers à ignorer par Git
|
|__ .venv/                                 → environnement virtuel Python
|__ catboost_info/                         → données du modèle ML
└__ data_collector/
    |__ bin/
    |    |__ get_data.sh                  → récupération des données
    |    |__ run.sh                      → script d'exécution
```

Pipeline de données (automatisé)

Étapes automatisées :

- Le script `get_data.sh` télécharge les données depuis Google Drive
- Le fichier `data.csv` est placé à la racine du projet
- L'application charge automatiquement les données à son démarrage
- La visualisation + prédition 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.**