

Business Case — Smart Product Recommender (Project #4)

1) Contexte

Dans un catalogue e-commerce, l'utilisateur est confronté à un grand nombre de produits. Sans personnalisation, la découverte est lente, ce qui réduit :

- le taux de conversion
- la valeur moyenne du panier
- la satisfaction client (perception de pertinence)

Objectif : proposer un module de recommandations simple et explicable pour améliorer la découverte produit.

2) Problème

- Les utilisateurs ne savent pas quel produit consulter ensuite.
- Le catalogue est large et hétérogène (catégories, prix, usages).
- Sans historique utilisateur (cold start), on doit proposer une solution basée sur le contenu.

3) Solution proposée

Un moteur de recommandation **content-based** alimenté par les métadonnées produits.

Données utilisées

- `name`, `category`, `description`, `tags` (texte)
- `price`, `rating`, `num_reviews` (numérique)

Approche

- Features texte : **TF-IDF** (unigrams + bigrams)
- Features numériques : **standardisation** (StandardScaler)
- Score de similarité : **cosine similarity** sur le vecteur final

Explicabilité

- recommandation = “produits proches” selon les mots/attributs partagés
- filtres = contraintes business (catégorie, prix max)

4) KPI / Mesures (portfolio)

Même sans tracking production, on peut expliquer les KPI visés :

- **CTR recos** : clic sur un produit recommandé
- **Conversion uplift** : taux d'achat après interaction recos
- **Avg order value** : panier moyen

- **Time to discovery** : temps jusqu'à un produit pertinent

Pour la démo :

- on affiche les scores de similarité
- on visualise la distribution des catégories et des prix

5) Hypothèses

- Les métadonnées produits sont suffisamment informatives (description/tags).
- Les produits similaires en contenu sont des alternatives acceptables pour l'utilisateur.
- Les produits avec `rating/num_reviews` élevés sont plus "safe" en recommandation.

6) Risques et limites

- **Cold start utilisateur** : pas d'historique => pas de personnalisation "user-based".
- **Biais de popularité** : `num_reviews` peut sur-favoriser les best-sellers.
- **Qualité des tags** : si les tags sont incomplets, la similarité peut être bruitée.

7) Mini-exercice UX — Filtre catégorie

Pourquoi ce filtre ?

Le filtre catégorie illustre un compromis UX/business :

- **Sans filtre** : plus de diversité (découverte), parfois moins de pertinence perçue.
- **Avec filtre** : recommandations plus cohérentes, mais diversité réduite.

Impact attendu

- si l'utilisateur est en phase "je cherche un produit précis" → filtre utile
- si l'utilisateur est en phase "je découvre" → filtre à éviter / laisser sur "All"

8) Roadmap

- Ajouter une vraie source de données (dataset public ou export catalogue)
- Ajouter un signal comportemental (clics) → hybrid recommender
- A/B test sur CTR/conversion

9) Conclusion

Le dashboard démontre une approche recommandation "pro" :

- pipeline de features
- score de similarité
- contrôles UX (filtres)
- visualisation et storytelling