|  |
| --- |
| **--> Lynda**    **Voici un état des lieux de l’avancement de mon projet final MarketAnalyzerAI Immo :**    **✅ Ce qui a déjà été accompli**   * **Jour 1 (Setup & Organisation)** :   + Mise en place du repo GitHub avec l’arborescence recommandée (src/, data/, notebooks/, streamlit\_app/, etc.).   + Téléchargement et préparation des datasets **Paris** et **Seattle** (InsideAirbnb).   + Script d’ingestion fonctionnel (src/ingest\_core.py) + documentation des sources (DATA\_SOURCES.md). * **Jour 2 (Data Wrangling & EDA)** :   + Nettoyage de base (suppression outliers prix > 2000€/nuit, colonnes inutiles).   + Normalisation initiale (dates → datetime, prix → float, available → 0/1).   + Scripts EDA en place (src/eda.py) : génération d’un rapport **EDA complet (Excel bilingue FR/EN)**, incluant tailles des datasets, % de NA, doublons, stats descriptives, distribution des prix, etc.   + Visualisations exploratoires (histogrammes, boxplots).     **⚠️ Difficultés rencontrées**   * **EDA** : gestion de colonnes fortement manquantes (ex. neighbourhood\_group\_cleansed, calendar\_updated), et difficultés à stabiliser certaines visualisations (occupancy quand calendrier vide). * **Conversion CSV → XLSX** : erreurs récurrentes avec pandas.to\_excel() dues à des colonnes incohérentes ou aux limites Excel (31 caractères pour noms d’onglets). * **Paris** : fichier calendar.csv vide → obligé de travailler avec des proxies pour occupancy\_rate et revenue.     **📅 Prochaines étapes (alignées avec planning initial)**   * **Jour 3 (ML Baseline)** :   + Créer features de base : occupancy\_rate, revenue, sentiment\_score, nb\_amenities, day\_of\_week, month.   + Entraîner un modèle baseline (Régression linéaire + RandomForest).   + Évaluer avec RMSE, R² ; produire confusion matrix (classification binaire : “bien noté” vs “mal noté”). * **Jour 4 (ML Avancé)** :   + Tester XGBoost & LightGBM.   + Comparer performances baseline vs avancés (tableau comparatif metrics).   + Mettre en place un pipeline sklearn (Pipeline + GridSearchCV). * **Jour 5 (Deep Learning)** :   + Construire un MLP simple pour prédiction réservation.   + Tracer learning curves (loss/accuracy). * **Jour 6–9** : poursuivre NLP (sentiment avec BERT), RAG, simulateur, dashboard Streamlit, et finaliser PPT + README.   **Jour 6 – NLP / LLM**   * Entraîner un classifieur de sentiments (TF-IDF + LogisticRegression). * Charger un modèle pré-entraîné (DistilBERT, BERT). * Pipeline complet : tokenisation → embeddings → classification. * Expliquer architectures encodeur/décodeur (BERT vs GPT vs T5). * Slide PPT : comparaison “classique vs Transformers”.     **Jour 7 – RAG & Chatbot**   * Créer pipeline RAG :   + Embeddings (HuggingFace MiniLM ou OpenAI).   + Stockage (FAISS ou Chroma).   + Retrieval + génération avec LLM. * Intégrer dans chatbot Streamlit (zone texte + réponses). * Tester 2–3 questions réelles (ex. “Quel est le prix moyen en été ?”). * Documenter pipeline dans README.     **Jour 8 – Simulateur & Dashboard**   * Créer simulateur “What if…?” (sliders prix/équipements → prédiction revenu). * Intégrer dashboard Streamlit (onglet simulateur + onglet chatbot). * Ajouter visualisations interactives (Plotly/Altair). * Vérifier fonctionnement end-to-end (data → ML/DL → RAG → UI). * Déployer sur Streamlit Cloud et sauvegarder lien public.     **Jour 9 – Finitions & Préparation Jury**   * Finaliser **PPT (template officiel)** avec :   + Intro projet + stack.   + ML (résultats + matrice confusion).   + DL (learning curves).   + NLP/LLM (tokenisation + encodeur/décodeur).   + RAG/Chatbot.   + Simulateur (impact revenu).   + Slide final **Business Model** : cibles, revenus (SaaS/API), différenciation. * Finaliser README (badges, structure claire, liens démo + vidéo Loom). * Répétition **chronométrée** (4 min PPT + 3 min vidéo + 3 min Q/R). * Vérifier démo Streamlit (connexion internet). * Préparer FAQ jury (expliquer confusion matrix, learning curve, choix dataset, limites/biais).     👉 Cette **checklist** couvre :   * **Techniques demandées** (tokenisation, transformers, encodeur/décodeur, matrice de confusion, learning curves). * **Livrables pédagogiques** (README, code commenté, PPT, démo). * **Perspectives business** (business model).       **🔍 Demande de conseils**  Je sens que j’ai pris un peu de retard sur **l’EDA et le cleaning** (plus chronophage que prévu), ce qui pourrait impacter la suite (ML baseline dès demain).  👉 Quels seraient vos conseils d’organisation pour :   * **accélérer** les prochaines étapes tout en gardant un livrable robuste, * **prioriser** les éléments “must have” pour le jury (par rapport aux “nice to have”), * gérer efficacement la partie **Paris (calendar vide)** sans y passer trop de temps, mais sans fragiliser le projet. |

**✅ Checklist Projet Final – MarketAnalyzerAI Immo**

**Jour 1 – Setup & Organisation**

* Créer le repo GitHub (public) avec arborescence : data/, notebooks/, src/, streamlit\_app/, README.md, requirements.txt.
* Rédiger README v1 (nom projet, description courte, objectifs, stack).
* Choisir et télécharger dataset(s) (Airbnb/Kaggle ou simulation).
* Vérifier format des données (.head(), .info(), .describe()).
* Préparer tableau comparatif des datasets candidats (pour PPT).

**Jour 2 – Data Wrangling & EDA**

* Nettoyer données (valeurs manquantes, doublons, outliers).
* Normaliser colonnes (prix, dates → format datetime, encodage catégories).
* Faire EDA : stats descriptives + corrélations (heatmap Seaborn).
* Créer visualisations PPT-ready (histogrammes, boxplots, courbes saisonnalité).
* Documenter le notebook avec commentaires clairs.

**Jour 3 – (Conférence matin) → Après-midi ML Baseline**

* Créer features de base (surface, localisation, saison, prix).
* Entraîner un modèle simple (Régression linéaire ou RandomForest).
* Évaluer : RMSE, R² (régression) ou accuracy (classification simplifiée).
* Générer **matrice de confusion** pour classification binaire simple (bien noté / mal noté).
* Exporter 1 graphique clair pour PPT (“Résultats modèle baseline”).

**Jour 4 – (Conférence matin) → Après-midi ML Avancé**

* Tester modèles plus avancés (XGBoost, LightGBM).
* Comparer avec baseline (tableau metrics).
* Mettre en place un pipeline sklearn (Pipeline + GridSearchCV).
* Tracer **learning curves** pour montrer underfitting/overfitting.
* Documenter choix du “meilleur modèle ML” dans README/notebook.

**Jour 5 – Deep Learning**

* Créer un MLP simple (Keras ou PyTorch) pour prédiction de réservation.
* Tracer **learning curves (loss/accuracy)** pour train/test.
* Expliquer surapprentissage/sous-apprentissage avec exemples visuels.
* Sauvegarder le modèle (.h5 ou .pt).
* Préparer 2 visuels PPT (architecture réseau + courbe loss/accuracy).

**Week-end (entre J5 et J6)**

* Collecter un mini-corpus d’avis voyageurs (ou simuler).
* Nettoyer texte (lowercase, stopwords, ponctuation).
* **Tokenisation obligatoire** (NLTK, spaCy, ou HuggingFace).
* Sauvegarder version tokenisée pour réutilisation J6.
* Ajouter un notebook “text\_preprocessing.ipynb”.

**Jour 6 – NLP / LLM**

* Entraîner un classifieur de sentiments (TF-IDF + LogisticRegression).
* Charger un modèle pré-entraîné (DistilBERT, BERT).
* Pipeline complet : tokenisation → embeddings → classification.
* Expliquer **architectures encodeur/décodeur** (BERT vs GPT vs T5).
* Slide PPT : comparaison “classique vs Transformers”.

**Jour 7 – RAG & Chatbot**

* Créer pipeline RAG :
  + Embeddings (HuggingFace MiniLM ou OpenAI).
  + Stockage (FAISS ou Chroma).
  + Retrieval + génération avec LLM.
* Intégrer dans chatbot Streamlit (zone texte + réponses).
* Tester 2–3 questions réelles (ex. “Quel est le prix moyen en été ?”).
* Documenter pipeline dans README.

**Jour 8 – Simulateur & Dashboard**

* Créer simulateur “What if…?” (sliders prix/équipements → prédiction revenu).
* Intégrer dashboard Streamlit (onglet simulateur + onglet chatbot).
* Ajouter visualisations interactives (Plotly/Altair).
* Vérifier fonctionnement end-to-end (data → ML/DL → RAG → UI).
* Déployer sur **Streamlit Cloud** et sauvegarder lien public.

**Jour 9 – Finitions & Préparation Jury**

* Finaliser **PPT (template officiel)** avec :
  + Intro projet + stack.
  + ML (résultats + matrice confusion).
  + DL (learning curves).
  + NLP/LLM (tokenisation + encodeur/décodeur).
  + RAG/Chatbot.
  + Simulateur (impact revenu).
  + Slide final **Business Model** : cibles, revenus (SaaS/API), différenciation.
* Finaliser README (badges, structure claire, liens démo + vidéo Loom).
* Répétition **chronométrée** (4 min PPT + 3 min vidéo + 3 min Q/R).
* Vérifier démo Streamlit (connexion internet).
* Préparer FAQ jury (expliquer confusion matrix, learning curve, choix dataset, limites/biais).

👉 Cette **checklist** couvre :

* **Techniques demandées** (tokenisation, transformers, encodeur/décodeur, matrice de confusion, learning curves).
* **Livrables pédagogiques** (README, code commenté, PPT, démo).
* **Perspectives business** (business model).