

Plan du Projet

Niveau Avancé

Durée: 7 semaines

Synthèses: 3 fois par semaine pour contrôle et suivi des progrès

Semaine 1-2 : Modélisation d'un Data Warehouse

Objectif : Créer une infrastructure robuste pour stocker et gérer les données de vente nécessaires à l'analyse prédictive.

Tâches :

- **Jour 1-2 :** Réunion d'équipe pour définir les objectifs et répartir les rôles.
- **Jour 3-4 :** Installation et configuration de l'environnement de travail.
- **Jour 5-7 :** Création des tables de données pour les ventes, les produits, les clients, etc.

Synthèse: Lundi, Mercredi et Vendredi

Semaine 3-4 : Collecte et Préparation des Données

Objectif : Rassembler et nettoyer les données historiques nécessaires à l'analyse.

Tâches :

- **Jour 1-3 :** Génération des données synthétiques
- **Jour 4-5 :** Nettoyage des données (gestion des valeurs manquantes, outliers).
- **Jour 6-7 :** Préparation des données pour l'analyse exploratoire.

Synthèse: Lundi, Mercredi et Vendredi

Semaine 5-6 : Analyse et Modélisation Prédictive

Objectif : Développer des modèles de détection de fraude.

Tâches :

- **Jour 1-3 :** Analyse exploratoire des données pour identifier les tendances et les corrélations.
- **Jour 4-5 :** Sélection et entraînement des modèles de machine learning (régression, séries temporelles).
- **Jour 6-7 :** Évaluation des modèles et sélection du modèle final.

Synthèse: Lundi, Mercredi et Vendredi

Semaine 7 : Validation et Présentation des Résultats

Objectif : valider les résultats obtenus et préparer la présentation finale.

Tâches :

- **Jour 1-2 :** Validation croisée des modèles pour assurer leur robustesse.
- **Jour 3-5 :** Préparation du rapport final avec les résultats, les insights et les recommandations.
- **Jour 6-7 :** Présentation finale aux parties prenantes avec discussion des conclusions et des étapes futures.

Synthèse: Lundi, Mercredi et Vendredi

OUTILS

Collecte et Préparation des Données :

- **Python** : Langage de programmation principal pour le traitement des données et le développement des modèles.
- **Pandas** : Librairie Python pour la manipulation et l'analyse des données.
- **NumPy** : Package Python pour le calcul numérique.
- **SQL** : Pour l'extraction et la gestion des données à partir de bases de données relationnelles.

Analyse Exploratoire des Données (EDA) :

- **Matplotlib et Seaborn** : Bibliothèques Python pour la visualisation des données.
- **Plotly** : Pour créer des visualisations interactives.
- **Jupyter Notebook** : Environnement interactif pour l'analyse et la visualisation des données.

Modélisation Prédictive :

- **Scikit-learn** : Bibliothèque Python pour l'apprentissage machine, y compris la régression et les modèles de classification.
- **TensorFlow ou PyTorch** : Pour les réseaux de neurones et l'apprentissage profond si nécessaire.
- **XGBoost ou LightGBM** : Pour les modèles de boosting.

Validation et Optimisation des Modèles :

Cross-validation : Pour évaluer la performance des modèles.

- **GridSearchCV ou Randomized Search CV** : Pour l'optimisation des hyperparamètres des modèles.
- **Metrics** : Pour évaluer la précision, le rappel, le F1-score, etc RMSE , MAE , R_SCORE.

Visualisation et Présentation des Résultats :

- **Tableau** : Pour créer des tableaux de bord interactifs.

- **Power BI** : Pour la visualisation des données et la création de rapports.
- **Matplotlib, Seaborn, Plotly** : Pour la création de graphiques et de visualisations.

Environnement de Développement et Collaboration :

- **Git et GitHub** : Pour le contrôle de version et la collaboration en équipe.
- **Google Colab** : Plateforme cloud gratuite pour l'exécution de notebooks Jupyter avec accès GPU.
- **VS Code ou PyCharm** : Environnements de développement intégrés (IDE) pour le développement Python.

Infrastructure de Données :

- **PostgreSQL ou MySQL** : Bases de données relationnelles pour le stockage des données.
- **AWS S3 ou Google Cloud Storage** : Pour le stockage des données volumineuses si nécessaire.

Autres Outils Utiles :

- **Docker** : Pour la gestion des conteneurs et l'isolation des environnements.
- **Kubernetes** : Pour l'orchestration des conteneurs dans un environnement de production.