**Departement mathematiques et informatique**

**Rapport de fin de Module : Data Science Sous Python**

**APPLICATION DE SYNTHESE**

**ANALYSE & VISUALISATION**

**Filière :**

**«INGENIERIE INFORMATIQUE,BIG DATA AND CLOUD COMPUTING »**

**REALISE PAR: DARBI JALILA DATE LIMITE : 20/ 04/2025**

**ENCADRE PAR : M. Mohammed QBADOU**

**Année Universitaire : 2024-2025**



# Introduction

L'analyse des données de ventes est cruciale pour toute entreprise souhaitant comprendre ses performances, identifier les opportunités de croissance et prendre des décisions éclairées. Ce projet vise à appliquer des techniques de science des données en utilisant le langage Python pour explorer et synthétiser des données de ventes.

L’objectif principal de cette étude est d’analyser un ensemble de données de ventes issues d’un fichier CSV, en utilisant le langage Python et la bibliothèque Pandas. À travers cette analyse, nous cherchons à extraire des informations utiles sur les performances commerciales, identifier les produits ou régions les plus rentables, comprendre les tendances temporelles des ventes et émettre des recommandations stratégiques. Cette démarche s’inscrit dans une logique de data-driven decision making, permettant aux décideurs d’optimiser les opérations commerciales.

# Importation et Préparation des Données

L'application est structurée autour des étapes fondamentales du processus d'analyse de données :

* **2.1 Collecte de Données:**
  + Description : La première étape consiste à rassembler les données pertinentes provenant de diverses sources (bases de données, feuilles de calcul, services Web).
  + Importance : L'exactitude et la pertinence des données sont cruciales pour la validité de toute analyse.
* **2.2 Extraction, Transformation et Chargement (ETL) des Données:**
  + Description : Cette étape implique le nettoyage et la transformation des données pour les rendre appropriées à l'analyse.
  + Importance : L'ETL garantit la cohérence, l'exactitude et l'exhaustivité des données, ce qui est essentiel pour obtenir des résultats fiables.
* **2.3 Création d'un Data Frame de Dates:**
  + Description : Création d'une table de dates avec des colonnes pour les dates, les années, les trimestres, les mois, les jours de la semaine, etc.
  + Exigences :
    - Marquée comme table de dates
    - Une colonne de type de données Date (ou datetime)
    - Des valeurs uniques dans la colonne de dates
    - Pas de valeurs vides dans la colonne de dates
    - Couvre des années entières
  + Importance : Essentielle pour l'analyse des séries temporelles.
* **2.4 Création d'un Data Frame des Mesures et Indicateurs:**
  + Description : Utilisation de fonctions d'agrégation et de synthèse (SUM, AVERAGE, COUNT, MIN, MEAN, STD) pour créer un Data Frame avec des mesures et des indicateurs.
  + Implémentation : Conversion des expressions de mesures (initialement en DAX) en Python à l'aide de pandas, numpy et de fonctions lambda.
* **2.5 Développement de Visuels et de Rapports:**

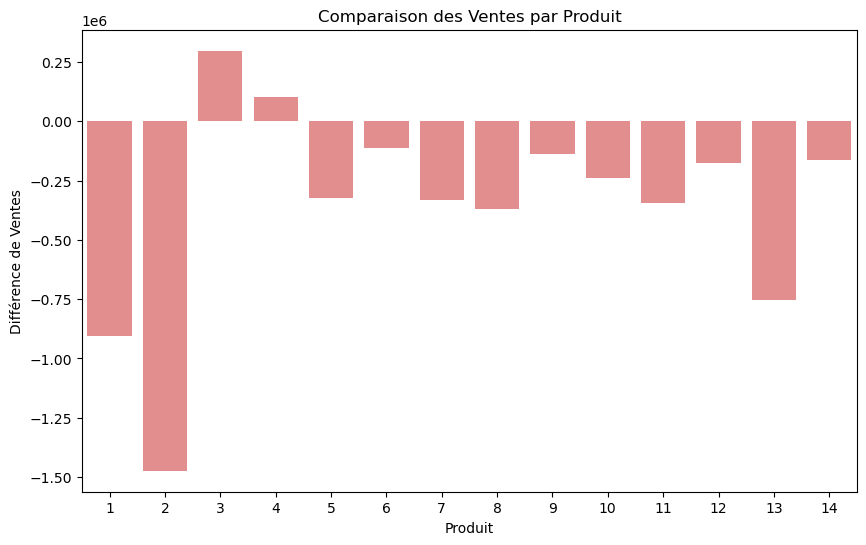
Les visualisations jouent un rôle essentiel dans la communication des résultats de l'analyse. Elles permettent d'identifier rapidement les tendances, les comparaisons et les points saillants des données de ventes.

Description : Utilisation de Matplotlib et Seaborn pour créer des visualisations de données efficaces et des rapports interactifs.

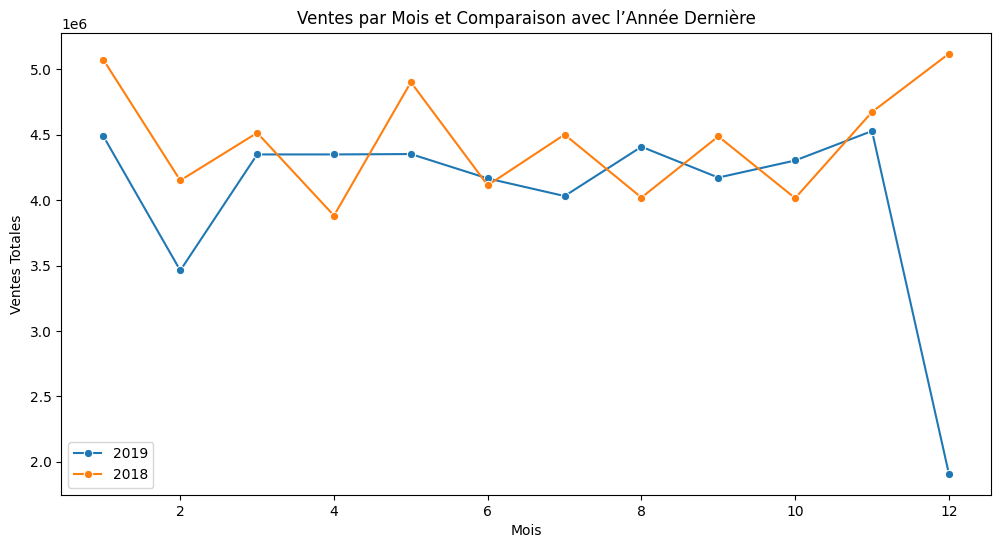
Fonctionnalités :

\*Filtres et fonctionnalités d'extraction pour l'exploration détaillée des données.

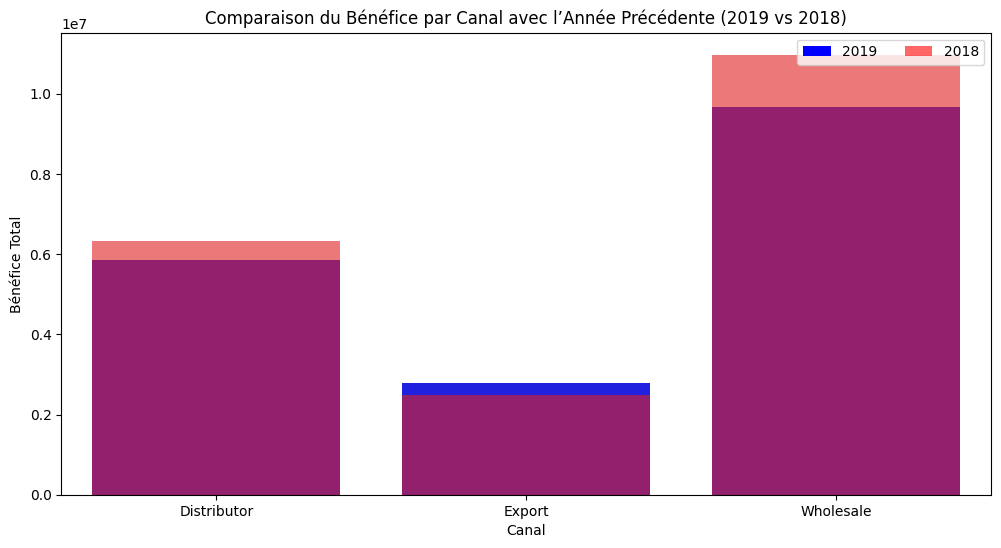
\*Visualisations recommandées :

-Ventes par produit

-Ventes par mois comparées à l'année précédente



-Ventes des 5 premières villes

-Bénéfice par canal comparé à l'année précédente(2018-2019)

-Top 5 des clients par ventes (comparaison avec l'année précédente)

-Bottom 5 des clients par ventes (comparaison avec l'année précédente)

-Cartes interactives pour les ventes, le profit, la marge bénéficiaire et les produits vendus (utilisation de bibliothèques SIG)

# Création de la Table de Dates

 **Importance de la Table de Dates :**

* "L'analyse des données de ventes sur une période donnée nécessite une structure temporelle cohérente. Une table de dates distincte permet de simplifier l'agrégation et le filtrage des données par année, trimestre, mois, etc., et est essentielle pour les comparaisons temporelles comme les variations d'une année à l'autre."

 **Processus de Création :**

* "La table de dates a été créée en utilisant la fonction pd.date\_range de la librairie pandas, générant une séquence de dates continues allant du [première année] au [dernière année], basées sur la plage des dates présentes (ou attendues) dans les données de ventes."
* "Des colonnes supplémentaires ont été extraites de cette séquence de dates, incluant l'année, le trimestre, le mois (numérique et nom), le jour du mois, le jour de la semaine (numérique et nom)."

 **Vérification des Exigences :**

* "La colonne 'Date' de la table de dates a été vérifiée pour s'assurer qu'elle est du type datetime64[ns]."
* "L'unicité des valeurs a été confirmée à l'aide de la fonction nunique()."
* "Le nombre de valeurs nulles a été vérifié avec isnull().sum() et confirmé comme étant nul."
* "Une vérification de l'absence de dates manquantes a été effectuée en comparant la longueur de la table de dates avec le nombre de jours attendu sur la période, tenant compte des années bissextiles
* "Les dates minimales et maximales de la table de dates ([2017] et [2019]) confirment qu'elle couvre des années entières pertinentes pour l'analyse."

# Technologies Utilisées

L'application s'appuie sur des technologies Python clés pour l'analyse de données :

**4.1 pandas : La Fondation de la Manipulation et de l'Analyse de Données**

* **Rôle central :** pandas est une bibliothèque Python extrêmement puissante qui fournit des structures de données de haut niveau et des outils d'analyse conçus pour rendre le travail avec des données "relationnelles" ou "étiquetées" facile et intuitif.
* **DataFrames :** La structure de données fondamentale de pandas est le DataFrame, qui est une table 2D (comme une feuille de calcul) où chaque colonne peut avoir un type de données différent. Cela te permet de stocker et de manipuler efficacement des données hétérogènes.
* **Fonctionnalités clés :**
  + **Lecture et écriture de données :** pandas peut lire et écrire des données dans de nombreux formats différents : CSV, Excel, SQL databases, JSON, etc.
  + **Indexation et sélection de données :** Tu peux facilement sélectionner des lignes, des colonnes ou des sous-ensembles de données en utilisant des étiquettes ou des indices.
  + **Nettoyage des données :** pandas offre des fonctions pour gérer les valeurs manquantes, supprimer les doublons, etc.
  + **Transformation des données :** Tu peux trier, filtrer, regrouper, agréger et restructurer les données.
  + **Combinaison de données :** pandas te permet de fusionner et de joindre des DataFrames de manière similaire aux opérations SQL.
  + **Analyse temporelle :** pandas a d'excellentes fonctionnalités pour travailler avec les séries temporelles.

**4.2 numpy : La Puissance des Opérations Numériques**

* **Calculs numériques :** numpy est la bibliothèque fondamentale pour le calcul numérique en Python. Elle fournit un support pour les tableaux multidimensionnels et les fonctions mathématiques pour opérer sur ces tableaux.
* **Tableaux (ndarrays) :** La structure de données principale de numpy est le ndarray, un tableau N-dimensionnel. Les tableaux numpy sont beaucoup plus efficaces que les listes Python pour stocker et manipuler de grandes quantités de données numériques.
* **Fonctionnalités clés :**
  + **Opérations vectorisées :** numpy permet d'effectuer des opérations sur des tableaux entiers en même temps, ce qui est beaucoup plus rapide que de boucler sur les éléments individuels.
  + **Fonctions mathématiques :** numpy fournit une vaste collection de fonctions mathématiques (trigonométriques, statistiques, etc.).
  + **Algèbre linéaire :** numpy a des fonctions pour les opérations d'algèbre linéaire (produits matriciels, inversions, etc.).
  + **Génération de nombres aléatoires :** numpy peut générer des nombres aléatoires à partir de différentes distributions.

**4.3 Matplotlib et Seaborn : La Visualisation des Données**

* **Matplotlib : La base de la visualisation**
  + Matplotlib est une bibliothèque de traçage 2D en Python qui produit des figures de qualité publication dans une variété de formats imprimables et de environnements interactifs.
  + Elle offre un contrôle considérable sur tous les aspects d'une figure, depuis les lignes et les couleurs jusqu'aux axes et aux étiquettes.
  + Matplotlib est souvent utilisée comme base pour d'autres bibliothèques de visualisation, comme Seaborn.
* **Seaborn : La visualisation statistique**
  + Seaborn est construite sur Matplotlib et fournit une interface de plus haut niveau pour créer des graphiques statistiques attrayants et informatifs.
  + Seaborn simplifie la création de certains types de graphiques (comme les graphiques de distribution, les graphiques de régression, etc.) qui sont complexes à réaliser avec Matplotlib seul.
  + Elle offre également de meilleurs styles et palettes de couleurs par défaut, ce qui rend les visualisations plus esthétiques.
* **Types de visualisations :**
  + **Graphiques linéaires :** Pour montrer les tendances au fil du temps.
  + **Graphiques à barres :** Pour comparer des quantités entre différentes catégories.
  + **Histogrammes :** Pour visualiser la distribution d'une variable.
  + **Diagrammes de dispersion :** Pour explorer les relations entre deux variables.
  + **Box plots (boîtes à moustaches) :** Pour afficher la distribution et les valeurs aberrantes.
  + **Heatmaps (cartes de chaleur) :** Pour visualiser les matrices de corrélation.

**4.4 Bibliothèques SIG (Système d'Information Géographique) : La Dimension Spatiale**

* **Analyse spatiale :** Les bibliothèques SIG en Python (comme geopandas, qui s'appuie sur pandas et shapely) te permettent de travailler avec des données géographiques.
* **Types de données :** Ces bibliothèques peuvent manipuler des données vectorielles (points, lignes, polygones) et des données raster (images satellites, etc.).
* **Fonctionnalités clés :**
  + **Lecture et écriture de formats géospatiaux :** Elles peuvent lire et écrire des formats de fichiers comme Shapefile, GeoJSON, etc.
  + **Opérations géométriques :** Calculer des aires, des périmètres, des distances, des intersections, etc.
  + **Visualisation de données géographiques :** Créer des cartes.
  + **Analyse spatiale :** Effectuer des requêtes spatiales (trouver les points dans un polygone, etc.).

# Conclusion

L’analyse réalisée met en évidence des tendances claires dans la dynamique des ventes de l’entreprise. Grâce à un traitement rigoureux des données, incluant le nettoyage, l’exploration statistique et l’analyse par catégorie, il a été possible d’identifier les produits les plus performants ainsi que les comportements d’achat des clients.

Les résultats obtenus permettent de mieux comprendre la structure des ventes, de cibler les segments porteurs, et d’orienter les décisions futures en matière de stratégie commerciale, de gestion des stocks ou encore de planification marketing.

Ce type d’analyse constitue un outil essentiel pour accompagner la prise de décision basée sur les données. Il ouvre également la voie à des analyses plus avancées telles que la prévision des ventes ou l’étude de la rentabilité par région ou par client, renforçant ainsi la capacité de l’entreprise à anticiper et s’adapter à un marché en constante évolution.