INPT
المعهد الوطني للبريد والمواصلات
ⵓⵙⴷⵉⵡ ⵓⵏⵎⵓⵔⵉ ⵏ ⵜⵙⵙⵉⵡⵉⵍⵜ ⴰ ⵜⵎⵔⴰⵡⵉⵜ
Institut National des Postes et Télécommunications

anrt
agence nationale de réglementation des télécommunications
الوكالة الوطنية لتقنين المواصلات

Mémoire de projet de Fin d'année

Ingénieur des Sciences de Données - Data Engineer

Prévision de la demande d'expédition et de distribution

Élèves :
HALMAOUI HAjar

Encadrant :
Mme. EL ASRI Ikram

Année universitaire : 2023-2024

*Remerciement

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet. Tout d'abord, nous remercions notre encadrante, Mme. EL ASRI Ikram, pour son soutien constant, ses conseils avisés et son expertise précieuse tout au long de cette étude. En outre, nous adressons nos remerciements les plus sincères à nos familles et amis pour leur soutien moral et encouragement inconditionnel.

*Résumé

Ce projet se concentre sur la prévision de la demande d'expédition et de distribution en utilisant des techniques avancées de traitement et de modélisation des données. Un pipeline ETL a été développé pour collecter, transformer et charger les données historiques d'expédition. En exploitant des outils tels que Google Colab, PySpark et Power BI, nous avons construit des modèles prédictifs permettant d'optimiser la gestion des stocks, de réduire les coûts opérationnels et de garantir des livraisons en temps voulu. Les résultats des prédictions sont intégrés dans des tableaux de bord interactifs pour faciliter la prise de décision stratégique.

*Abstract

This project focuses on forecasting shipping and distribution demand using advanced data processing and modeling techniques. An ETL pipeline was developed to collect, transform, and load historical shipping data. Utilizing tools such as Google Colab, PySpark, and Power BI, we built predictive models to optimize inventory management, reduce operational costs, and ensure timely deliveries. The prediction results are integrated into interactive dashboards to facilitate strategic decision-making.

*Mots clés

Prévision de la demande

expédition, distribution

pipeline ETL

modélisation des données

Google Colab

PySpark

Power BI

gestion des stocks

optimisation logistique.

*Liste des Abréviations

ETL : Extract, Transform, Load

EDA : Analyse Exploratoire des Données

RMSE : Root Mean Squared Error

MAE : Mean Absolute Error

CLV : Customer Lifetime Value

KPI : Key Performance Indicator

ML : Machine Learning

CSV : Comma-Separated Values

API : Application Programming Interface

GBT : Gradient Boosted Trees

Table des matières

Table des figures

Introduction

Dans un marché mondial de plus en plus compétitif, la gestion efficace des chaînes d'approvisionnement et des opérations logistiques est essentielle pour le succès et la pérennité des entreprises. Ce rapport présente une étude approfondie sur la prévision de la demande d'expédition et de distribution, un aspect crucial pour optimiser la gestion des stocks, réduire les coûts opérationnels et assurer des livraisons en temps voulu. Notre

projet se concentre sur la création d'un pipeline ETL (Extract, Transform, Load) robuste pour traiter les données historiques d'expédition et de distribution. En utilisant des techniques avancées de modélisation et des outils de visualisation performants, nous visons à développer un système de prévision de la demande précis et fiable. Les résultats obtenus sont intégrés dans des tableaux de bord interactifs, permettant aux parties prenantes de visualiser et d'exploiter les données pour une prise de décision stratégique éclairée. Ce

document est structuré en plusieurs chapitres détaillant le contexte et les objectifs du projet, l'état de l'art des techniques et outils utilisés, la méthodologie adoptée pour la collecte et la transformation des données, ainsi que les analyses, modélisations et évaluations effectuées. Les résultats des prédictions et les implémentations pratiques sont également discutés, avant de conclure sur les implications et les perspectives d'amélioration future. Les technologies utilisées incluent Google Colab pour le développement et l'exécution

des pipelines de données, PySpark pour le traitement et la transformation des données à grande échelle, Google Drive pour le stockage sécurisé des données, et Power BI pour la visualisation interactive des résultats. Ces choix technologiques permettent de garantir la qualité, la fiabilité et la pertinence de notre solution, tout en facilitant la collaboration et l'analyse des données en temps réel. En somme, ce projet vise à fournir une solution

intégrée et efficace pour la prévision de la demande, contribuant ainsi à l'amélioration des performances opérationnelles et à l'optimisation des chaînes d'approvisionnement des entreprises.

1 Chapitre 1 Présentation du projet

1.1 Introduction

Dans cette section, nous examinerons le contexte et les objectifs de notre projet, ainsi que l'importance de la prévision de la demande dans la gestion des inventaires. Nous détaillerons les choix technologiques et les outils que nous avons sélectionnés pour répondre à ces objectifs, en mettant en évidence leur pertinence dans la réalisation de notre solution.

1.2 Contexte et Objectifs du Projet

Dans un marché de plus en plus compétitif et dynamique, la gestion efficace des chaînes d'approvisionnement est essentielle pour le succès d'une entreprise. Notre projet vise à créer un pipeline ETL robuste pour traiter les données d'expédition et de distribution. En utilisant des données historiques, nous allons construire un système de prévision de la demande qui prédit la demande future de produits dans le contexte de l'expédition et de la distribution. Cette prévision est cruciale pour optimiser la gestion des stocks, réduire les coûts opérationnels et garantir des livraisons en temps voulu. En outre, nous allons développer un tableau de bord interactif pour présenter les résultats des prédictions ainsi que d'autres analyses importantes pour la prise de décision stratégique.

1.3 Importance de la Prévision de la Demande dans la Gestion des Inventaires

La prévision de la demande joue un rôle central dans la gestion des inventaires en permettant aux entreprises de planifier leurs approvisionnements, de minimiser les coûts de stockage et de répondre efficacement aux fluctuations de la demande du marché. En comprenant les tendances passées et en utilisant des techniques de modélisation avancées, nous pouvons anticiper les besoins futurs et prendre des décisions éclairées pour optimiser les opérations logistiques.

1.4 Technologies et Outils Utilisés

Pour atteindre nos objectifs, nous utiliserons une combinaison de technologies et d'outils avancés, notamment :

— **Google Colab** : Google Colab sera utilisé pour le développement et l'exécution de notre pipeline ETL. Il offre un environnement de développement basé sur le cloud avec un accès gratuit à des ressources informatiques puissantes, ce qui le rend idéal pour le traitement de données volumineuses.

FIGURE 2 – Google Colob logo

— **PySpark** : PySpark sera employé pour le nettoyage et la transformation des données à grande échelle. Cette bibliothèque permet de traiter des ensembles de données massifs de manière distribuée, offrant ainsi des performances élevées pour les opérations de transformation.

FIGURE 3 – Pyspark logo

— **Google Drive** : Google Drive sera utilisé pour le stockage sécurisé des données et des résultats intermédiaires. Il offre une solution pratique pour le partage et la collaboration sur les fichiers, ce qui facilite l'accès aux données pour tous les membres de l'équipe.

FIGURE 4 – Google Drive logo

— **Power BI** : Power BI sera utilisé pour la création de tableaux de bord interactifs et la visualisation des résultats. Cette plateforme offre des fonctionnalités avancées de visualisation de données, facilitant ainsi l'analyse et la communication des insights tirés des données.

FIGURE 5 – Power BI logo

Nous allons détailler chaque étape de notre processus, depuis la collecte des données jusqu'à l'analyse des résultats, en mettant en lumière les choix technologiques et les méthodologies adoptées pour garantir la qualité et la fiabilité de notre solution.

2 Chapitre 2 État de l'Art

2.1 Introduction

Dans cette section, nous explorerons trois aspects essentiels liés à notre projet de prédiction de la demande et d'optimisation des expéditions : les concepts clés de l'ETL, les techniques de prévision de la demande et les outils de visualisation de données. Nous détaillerons chaque aspect pour fournir un aperçu approfondi des méthodes et des outils utilisés dans notre projet.

2.2 Concepts Clés de l'ETL

L'ETL (Extract, Transform, Load) est un processus fondamental dans le domaine de la gestion des données. Il consiste à extraire des données brutes à partir de différentes sources, à les transformer en un format cohérent et analytique, puis à les charger dans une base de données ou un entrepôt de données. Les principales étapes de l'ETL comprennent :

- **Extraction** : Collecte des données à partir de diverses sources telles que des bases de données, des fichiers plats, des API, etc.
- **Transformation** : Nettoyage, structuration et enrichissement des données pour les rendre utilisables pour l'analyse.
- **Chargement** : Stockage des données transformées dans une base de données ou un entrepôt de données pour une utilisation ultérieure.

2.3 Techniques de Prévision de la Demande

La prévision de la demande repose sur une variété de techniques et de modèles, allant des méthodes statistiques traditionnelles aux approches d'apprentissage automatique avancées. Parmi les techniques couramment utilisées, on trouve :

- **Méthodes statistiques** : ARIMA (AutoRegressive Integrated Moving Average), Holt-Winters, etc.
- **Modèles d'apprentissage automatique** : Réseaux de neurones, arbres de décision, forêts aléatoires, etc.
- **Techniques de séries temporelles** : Décomposition saisonnière, modèles SARIMA (Seasonal ARIMA), etc.

Le choix de la technique de prévision dépend souvent de la nature des données, de la complexité des modèles et des objectifs spécifiques du projet.

2.4 Outils de Visualisation de Données

Les outils de visualisation de données sont essentiels pour explorer, analyser et communiquer les résultats des prévisions de manière efficace. Parmi les outils populaires, on trouve :

- **Power BI** : Plateforme de business intelligence de Microsoft permettant de créer des tableaux de bord interactifs et des rapports visuels.
- **Tableau** : Logiciel de visualisation de données avancé offrant des fonctionnalités de manipulation et d'analyse des données.

— **Matplotlib, Seaborn** : Bibliothèques Python pour la création de graphiques et de visualisations personnalisées.

Ces outils permettent de présenter les résultats des prévisions sous forme de graphiques, de tableaux et d'autres éléments visuels, facilitant ainsi la compréhension et l'interprétation des données pour les parties prenantes du projet.

3 Chapitre 3 Méthodologie

3.1 Introduction

Dans cette section, nous abordons la phase cruciale de collecte et de transformation des données pour notre projet de prédiction de la demande et d'optimisation des expéditions. Nous commençons par explorer les sources de données, en détaillant les différentes variables et attributs inclus dans le dataset. Ensuite, nous examinons les techniques de nettoyage et de préparation des données, ainsi que les stratégies de feature engineering utilisées pour améliorer la qualité des données. Enfin, nous discutons du processus de chargement des données, en mettant en évidence l'utilisation de Google Drive et de Google Colab pour assurer un stockage sécurisé et une gestion efficace des données transformées.

3.2 Collecte des Données

3.2.1 Sources de Données (Dataset de l'entreprise DATACO)

Les données utilisées dans ce projet proviennent de l'entreprise DATACO, qui fournit un dataset complet couvrant divers aspects de la chaîne d'approvisionnement, y compris les informations sur les clients, les produits, les commandes et les livraisons. Ce dataset contient des attributs tels que les dates de commande et de livraison, les quantités de produits, les informations géographiques, les segments de clientèle, et bien plus encore.

3.2.2 Description des Données Collectées

Les colonnes suivantes sont incluses dans le dataset :
— Type : Type de transaction effectuée.
— Days for shipping (real) : Jours réels d'expédition du produit acheté.
— Days for shipment (scheduled) : Jours de livraison prévue du produit acheté.
— Benefit per order : Bénéfice par commande passée.
— Sales per customer : Total des ventes par client.
— Delivery Status : Statut de la livraison des commandes : Advance shipping, Late delivery, Shipping canceled, Shipping on time.
— Late_delivery_risk : Variable catégorielle indiquant si la livraison est en retard (1) ou non (0).
— Category Id : Code de la catégorie du produit.
— Category Name : Description de la catégorie du produit.
— Customer City : Ville où le client a effectué l'achat.
— Customer Country : Pays où le client a effectué l'achat.
— Customer Fname : Prénom du client.
— Customer Lname : Nom de famille du client.
— Customer Id : Identifiant du client.
— Customer Segment : Types de clients : Consumer, Corporate, Home Office.
— Customer State : État où le magasin enregistrant l'achat est situé.
— Customer Street : Rue où le magasin enregistrant l'achat est situé.
— Department Id : Code du département du magasin.
— Department Name : Nom du département du magasin.
— Latitude : Latitude correspondant à l'emplacement du magasin.
— Longitude : Longitude correspondant à l'emplacement du magasin.
— Market : Marché où la commande est livrée : Africa, Europe, LATAM, Pacific Asia, USCA.

— Order City : Ville de destination de la commande.
— Order Country : Pays de destination de la commande.
— Order Customer Id : Code de la commande du client.
— order date (DateOrders) : Date de la commande.
— Order Id : Code de la commande.
— Order Item Cardprod Id : Code du produit généré par le lecteur RFID.
— Order Item Discount : Valeur de la remise sur l'article commandé.
— Order Item Discount Rate : Pourcentage de remise sur l'article commandé.
— Order Item Id : Code de l'article commandé.
— Order Item Product Price : Prix des produits sans remise.
— Order Item Profit Ratio : Ratio de profit par article commandé.
— Order Item Quantity : Nombre de produits par commande.
— Sales : Valeur des ventes.
— Order Item Total : Montant total par commande.
— Order Profit Per Order : Profit par commande.
— Order Region : Région du monde où la commande est livrée.
— Order State : État de la région où la commande est livrée.
— Order Status : Statut de la commande : COMPLETE, PENDING, CLOSED, PENDING_PAYMENT, CANCELED, PROCESSING, SUSPECTED_FRAUD, ON_HOLD, PAYMENT_REVIEW.
— Product Card Id : Code du produit.
— Product Category Id : Code de la catégorie du produit.
— Product Description : Description du produit.
— Product Image : Lien de visite et d'achat du produit.
— Product Name : Nom du produit.
— Product Price : Prix du produit.
— Product Status : Statut du stock du produit : 1 pour non disponible, 0 pour disponible.
— shipping date (DateOrders) : Date et heure exacte de l'expédition.
— Shipping Mode : Mode de livraison utilisé : Standard Class, First Class, Second Class, Same Day.

3.3 Transformation des Données

3.3.1 Nettoyage et Préparation des Données

Pour préparer les données pour l'analyse et la modélisation, nous suivons plusieurs étapes de nettoyage et de transformation :

— Suppression des colonnes inutiles : Certaines colonnes comme "Customer Email", "Customer Password", "Product Image", etc., sont supprimées car elles ne sont pas pertinentes pour l'analyse.
— Gestion des valeurs manquantes : Les valeurs manquantes dans la colonne "Customer Lname" sont remplacées par "Unknown".

FIGURE 6 – pyspark code pour suppression et Gestion des valeurs manquantes

3.3.2 Feature Engineering (Création et Transformation des Variables)

La création de nouvelles variables et la transformation des existantes sont cruciales pour améliorer la qualité des prédictions :

— Extraction de la date : Ajout des colonnes "Month" et "Year" à partir de la colonne "order date (DateOrders)".

— Définition des saisons : Création de la colonne "Order_Season" pour indiquer la saison de la commande.

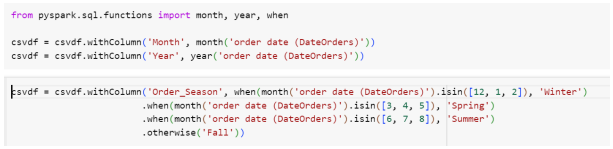


FIGURE 7 – pyspark code pour Feature Engineering

— Calcul de la fréquence des commandes par client

FIGURE 8 – pyspark code pour Calcul de la fréquence des commandes par client

— Identification des clients à haute valeur :

FIGURE 9 – pyspark code pour Identification des clients à haute valeur

3.4 Chargement des Données

3.4.1 Stockage des Données sur Google Drive

Après avoir effectué la transformation des données, celles-ci sont sauvegardées directement sur Google Drive pour assurer un stockage sécurisé et une accessibilité facile. Cette approche offre une solution pratique pour la gestion des données transformées sans nécessiter de serveur de base de données supplémentaire.

3.4.2 Processus de Chargement (Google Colab)

Le chargement des données utilise Google Colab pour exécuter le processus de transformation et de sauvegarde des données. Google Colab fournit un environnement de déve-

loppement intégré basé sur le cloud, offrant des ressources informatiques puissantes pour l'exécution de tâches de traitement des données.

```
csvdf_pd = csvdf.toPandas()
csvdf_pd.to_csv("/content/drive/MyDrive/cleaned_data.csv", index=False)
```

FIGURE 10 – code Python utilisé pour le processus de sauvegarde des données nettoyées sur Google Drive

Dans ce code, nous montons Google Drive dans l'environnement Colab, puis nous convertissons les données Spark DataFrame en Pandas DataFrame pour faciliter la manipulation. Ensuite, nous spécifions le chemin de sauvegarde pour les données nettoyées et exportons les données au format CSV sur Google Drive pour un stockage sécurisé et une accessibilité ultérieure.

4 Chapitre 4 Analyse , Modélisation et Évaluation

4.1 Introduction

Dans cette section, nous présentons l'analyse, la modélisation et l'évaluation des données pour notre projet de prévision de la demande et d'optimisation des expéditions. Nous commençons par une analyse exploratoire des données (EDA) afin d'identifier les tendances saisonnières, de comprendre les performances des produits, et d'explorer le comportement des clients. Ensuite, nous détaillons les modèles de prévision utilisés pour prédire la performance de livraison et la rentabilité des commandes, en évaluant l'efficacité de divers algorithmes de machine learning. Enfin, nous explorons les approches pour optimiser le mode de livraison et prédire la valeur à vie des clients (CLV), en utilisant des techniques de clustering et de régression avancées. Cette section met en lumière les méthodes et les outils employés pour transformer les données en informations exploitables, facilitant ainsi une prise de décision éclairée pour l'entreprise.

4.2 Analyse Exploratoire des Données (EDA)

4.2.1 Quantité de Produits Vendus par Mois

Nous avons examiné la quantité de produits vendus par mois pour identifier les tendances saisonnières.

```
# Quantité de produits vendus par mois
quantity_by_month = csvdf.groupBy("Month", "Year").sum("Order Item Quantity").orderBy("Year", "Month")
quantity_by_month_pd = quantity_by_month.toPandas()

plt.figure(figsize=(12, 6))
sns.barplot(data=quantity_by_month_pd, x='Month', y='sum(Order Item Quantity)', hue='Year')
plt.title('Quantité de produits vendus par mois')
plt.xlabel('Mois')
plt.ylabel('Quantité')
plt.show()
```

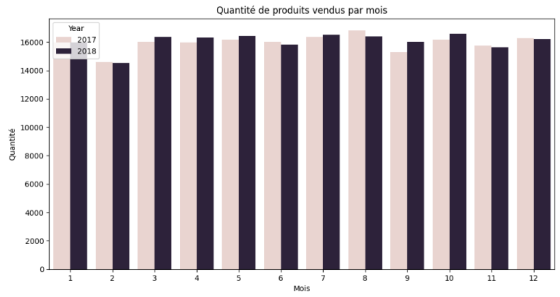


FIGURE 11 – visualisation de quantité de Produits Vendus par Mois

4.2.2 Ventes par Mois et Région

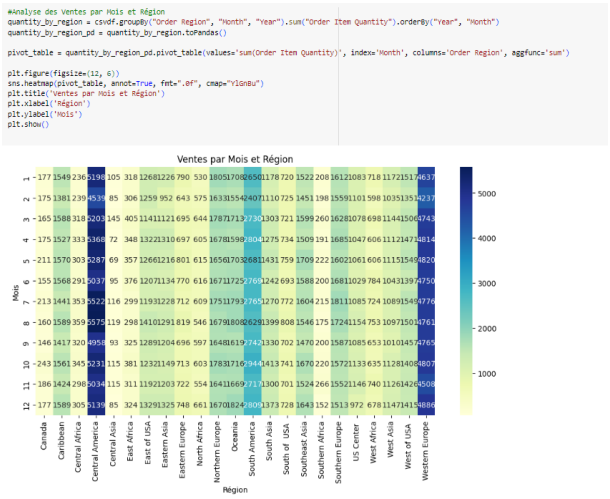


FIGURE 12 – visualisation de Ventes par Mois et Région

Cette heatmap permet de visualiser les variations des ventes selon les régions et les mois.

4.2.3 Performance des Produits

Pour comprendre quels produits se vendent le mieux, nous avons analysé les ventes par produit.

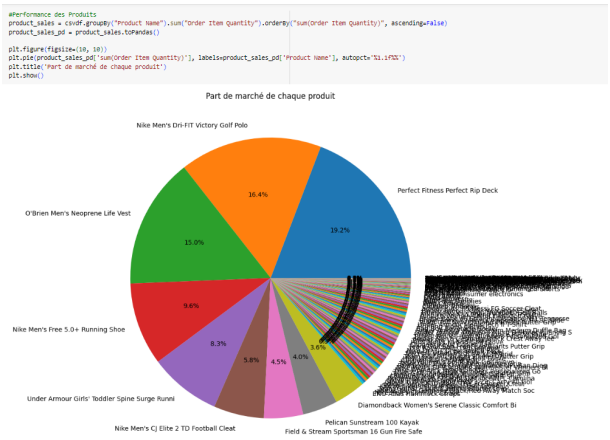


FIGURE 13 – visualisation des performance des produits

Cette analyse nous aide à identifier les produits les plus populaires et leur part de marché respective.

4.2.4 Analyse des Clients

Nous avons également exploré les données des clients pour comprendre la relation entre la fréquence des commandes et leur valeur.

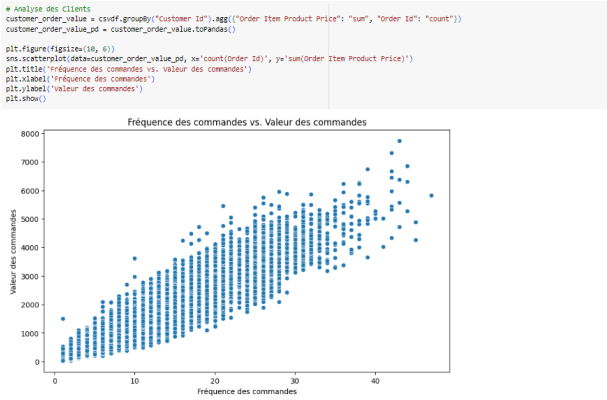


FIGURE 14 – visualisation des relation des clients et la fréquence des commandes

Cette visualisation montre comment la fréquence des commandes influence la valeur totale des commandes.

4.3 Modèles de Prévision Utilisés

4.3.1 Prédiction de la Performance de Livraison

Dans le cadre de notre analyse de la performance de livraison de l'entreprise DATACO, nous avons utilisé deux modèles de machine learning pour prédire la probabilité de retard de livraison : la Régression Logistique et les Arbres de Décision.

— **Modèles Utilisés**

1- Régression Logistique
— Ce modèle a été choisi pour sa capacité à modéliser la probabilité de retard de livraison en fonction de plusieurs variables explicatives.
— Nous avons utilisé la bibliothèque Spark MLlib pour implémenter la régression logistique en raison de sa scalabilité et de sa capacité à traiter de grands ensembles de données.
2- Arbres de Décision
— Les arbres de décision offrent une interprétabilité accrue et sont capables de capturer des relations non linéaires entre les variables.
— Nous avons également utilisé Spark MLlib pour entraîner et évaluer le modèle d'arbre de décision.
— **Variables Importantes**

Dans notre analyse, nous avons identifié plusieurs variables qui ont un impact significatif sur la performance de livraison. Ces variables comprennent :

— Nombre de jours pour l'expédition (réel)
— Nombre de jours pour l'expédition (planifié)
— Remise sur l'article commandé
— Taux de remise sur l'article commandé
— Prix du produit commandé
— Quantité d'articles commandés

— Profit de la commande par article
— Fréquence des commandes
— Mois de la commande
— Année de la commande
— Région de la commande (encodée numériquement)

```
feature_columns = ["Days for shipping (real)", "Days for shipment (scheduled)", "Order Item Discount", "Order Item Discount Rate",
                   "Order Item Product Price", "Order Item Quantity", "Order Profit Per Order", "Order Frequency", "Month", "Year"]
```

FIGURE 15 – données utilisés

— **Prétraitement des Données**

Avant de construire nos modèles, nous avons effectué plusieurs étapes de prétraitement des données pour garantir la qualité et la pertinence de nos résultats :

— Conversion des colonnes catégorielles en indices numériques : nous avons utilisé StringIndexer pour convertir la colonne "Order Region", une variable catégorielle, en indices numériques. Cela permet au modèle d'apprentissage automatique de traiter cette colonne en tant que caractéristique numérique.
— Assemblage des caractéristiques en un seul vecteur : nous avons utilisé VectorAssembler pour assembler toutes les caractéristiques (y compris les colonnes d'indices numériques créées précédemment) en un seul vecteur de caractéristiques. Cela crée une seule colonne "features" qui contient toutes les caractéristiques nécessaires pour entraîner le modèle.
— Division des données en ensembles d'entraînement et de test : nous avons utilisé vos données en ensembles d'entraînement et de test à l'aide de la méthode randomSplit. Cela vous permet d'évaluer les performances du modèle sur des données non vues pendant l'entraînement.

```
# Convertir les colonnes catégorielles en indices numériques
indexer = StringIndexer(inputCol="Order Region", outputCol="Order Region Index")
csvdf = indexer.fit(csvdf).transform(csvdf)

# Ajouter la colonne indexée à la liste des caractéristiques
feature_columns.append("Order Region Index")

# Assembler les caractéristiques en un seul vecteur
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
csvdf = assembler.transform(csvdf)
# Diviser les données en ensembles d'entraînement et de test
train_data, test_data = csvdf.randomSplit([0.7, 0.3], seed=42)
```

FIGURE 16 – code de pretraitement des données avant l'entrainement

— **Résultats et Évaluation**

Les performances des différents modèles de prédiction de la performance de livraison ont été évaluées en utilisant la métrique d'exactitude (Accuracy) sur l'ensemble de test.

Régression Logistique : Exactitude (Accuracy) : 0.973
Arbre de Décision : Exactitude (Accuracy) : 0.973

Les résultats montrent que les modèles de régression logistique et d'arbre de décision obtiennent des performances similaires avec une précision d'environ 97.3. Cela suggère que ces deux modèles sont efficaces pour prédire la performance de livraison des commandes.

4.3.2 Prédiction de la Rentabilité des Commandes

Dans cette section de notre analyse, nous avons cherché à prédire la rentabilité des commandes en utilisant différents modèles de régression. Nous avons exploré l'utilisation de RandomForestRegressor, GBTRegressor (Gradient Boosted Trees), et DecisionTree-Regressor pour cette tâche.

— **Modèles Utilisés**

1. RandomForestRegressor Le modèle RandomForestRegressor a été choisi pour sa capacité à capturer les relations complexes entre les variables et à gérer efficacement les données avec plusieurs caractéristiques. Nous avons évalué la performance de ce modèle en utilisant la métrique RMSE (Root Mean Squared Error), qui mesure l'écart moyen entre les valeurs prédites et les valeurs réelles.
2. GBTRegressor (Gradient Boosted Trees) Ce modèle a été choisi pour sa capacité à améliorer la précision de la prédiction en ajustant successivement les résidus des arbres précédents. Nous avons également évalué la performance de ce modèle en utilisant la métrique RMSE.
3. DecisionTreeRegressor Les arbres de décision sont des modèles simples et interprétables qui peuvent être efficaces pour la prédiction de la rentabilité des commandes. Nous avons évalué la performance de ce modèle en utilisant la métrique RMSE.

— **Variables Importantes**

Les variables suivantes ont été identifiées comme étant importantes pour la prédiction de la rentabilité des commandes :

— Coût de production
— Prix de vente
— Coût de livraison
— Volume de la commande
— Total Profit (calculé comme le produit du prix unitaire de l'article et de la quantité, moins la remise)

```
# Sélectionner les caractéristiques pertinentes
feature_columns = ["Days for shipping (real)", "Days for shipment (scheduled)", "Order Item Discount", "Order Item Discount Rate",
                   "Order Item Product Price", "Order Item Quantity", "Order Frequency", "Total Profit"]
```

FIGURE 17 – données utilisés

— **Prétraitement des Données**

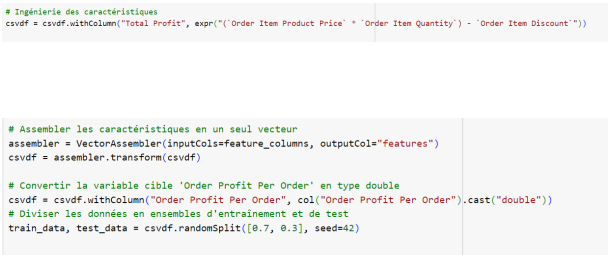
Avant de construire nos modèles, nous avons effectué les étapes de prétraitement suivantes sur nos données :

— Ingénierie des caractéristiques : Nous créons une nouvelle caractéristique appelée "Total Profit" en calculant le bénéfice total de chaque commande. Pour cela, nous soustrayons la remise du produit du produit du prix du produit et de la quantité commandée.
— Sélection des caractéristiques pertinentes : Nous choisissons un ensemble de caractéristiques qui sont importantes pour prédire la rentabilité de la commande.

Celles-ci incluent des éléments tels que les jours de livraison, les remises, le prix des produits, etc.

— Assemblage des caractéristiques en un seul vecteur : Nous utilisons l'outil VectorAssembler pour assembler toutes les caractéristiques sélectionnées en un seul vecteur de caractéristiques, ce qui est nécessaire pour l'entraînement du modèle.

— Conversion de la variable cible en type double : Nous convertissons la variable cible "Order Profit Per Order" en type double pour nous assurer qu'elle est dans un format compatible avec le modèle de régression que nous utilisons.



```
# Ingénierie des caractéristiques
csvdf = csvdf.withColumn("Total Profit", expr("(`Order Item Product Price` * `Order Item Quantity`) - `Order Item Discount`"))
```

```
# Assembler les caractéristiques en un seul vecteur
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
csvdf = assembler.transform(csvdf)

# Convertir la variable cible 'Order Profit Per Order' en type double
csvdf = csvdf.withColumn("Order Profit Per Order", col("Order Profit Per Order").cast("double"))
# Diviser les données en ensembles d'entraînement et de test
train_data, test_data = csvdf.randomSplit([0.7, 0.3], seed=42)
```

FIGURE 18 – code de prétraitement des données

— **Résultats et Évaluation**

Les performances des différents modèles de prédiction de la rentabilité des commandes ont été évaluées en utilisant la racine de l'erreur quadratique moyenne (RMSE) sur l'ensemble de test.

XGBoost : RMSE : 105.26
Arbre de Décision :RMSE : 106.15
Forêt Aléatoire : RMSE : 104.80

Ces résultats indiquent que le modèle XGBoost a la performance la plus optimale avec le RMSE le plus bas, suivi de près par le modèle de forêt aléatoire. Ces modèles sont capables de prédire avec précision la rentabilité des commandes, avec un écart moyen d'environ 105.26 pour XGBoost et 104.80 pour la Forêt Aléatoire.

4.3.3 Optimisation du Mode de Livraison

Dans cette section de notre analyse, nous avons cherché à optimiser le mode de livraison pour améliorer l'efficacité opérationnelle et la satisfaction client. Pour cela, nous avons utilisé deux modèles de classification, à savoir la régression logistique et le Naive Bayes, pour recommander le mode de livraison le plus approprié pour chaque commande.

— **Modèle Utilisé**

Nous avons utilisé les modèles de régression logistique et Naive Bayes pour prédire le mode de livraison des commandes en fonction de leurs caractéristiques, telles que le mode de livraison, les jours d'expédition réels et la région de la commande.

— **Variables Importantes**

Les variables suivantes ont été utilisées pour le clustering des commandes :

— Jours d'expédition (prévus)

— Jours pour l'expédition (réel)
— Région de la Commande

```
selected_features = ["Days for shipping (real)", "Order Region", "Days for shipment (scheduled)"]
```

FIGURE 19 – données utilisés

— **Prétraitement des Données**

Avant de construire le modèle K-means, nous avons effectué les étapes de prétraitement suivantes sur nos données :

— Conversion des colonnes catégorielles en valeurs numériques à l'aide de l'indexation de chaînes.
— Vectorisation des caractéristiques sélectionnées pour créer un vecteur de caractéristiques.

```
# Convert categorical columns to numerical values
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index", handleInvalid="keep") for column in selected_features]

# Convert the target column "Shipping Mode" to numeric values
label_indexer = StringIndexer(inputCol="Shipping Mode", outputCol="Shipping Mode_index").fit(csvdf)

# Vectorize features
assembler = VectorAssembler(inputCols=[column+"_index" for column in selected_features], outputCol="features")
```

FIGURE 20 – prétraitement des données avant entrainement

— **Résultats et Évaluation**

L'évaluation de l'algorithme KMeans pour l'optimisation du mode de livraison a été effectuée en mesurant son accuracy.
Régression Logistique :

— Accuracy : 0.5969

— Precision : 0.3766

— Recall : 0.5969

— F1 Score : 0.4618

Naive Bayes :

— Accuracy : 0.9049

— Precision : 0.9130

— Recall : 0.9049

— F1 Score : 0.9028

L'évaluation des modèles de régression logistique et Naive Bayes a montré que le modèle Naive Bayes avait une performance supérieure avec une accuracy de 0.9049, par rapport à

la régression logistique avec une accuracy de 0.5969. Le modèle Naive Bayes a également obtenu de meilleures valeurs de précision, de rappel et de score F1, suggérant une meilleure capacité à regrouper les données de manière à optimiser le mode de livraison pour chaque client.

4.3.4 Prédiction de la Valeur à Vie du Client (CLV)

Dans cette section de notre analyse, nous avons exploré différents modèles de régression pour prédire la valeur à vie du client (CLV), qui est une mesure essentielle pour comprendre la rentabilité à long terme de chaque client.

— **Modèles Utilisés**

1. **Régression Linéaire** : Nous avons utilisé un modèle de régression linéaire pour estimer la relation linéaire entre les caractéristiques des clients et leur valeur à vie.
2. **Arbre de Décision et Gradient Boosted Tree** : Nous avons également exploré des modèles non linéaires tels que forêt aléatoire et le gradient boosted tree (GBT) Régresseur pour capturer des relations complexes entre les variables.

— **Variables Importantes**

Les variables suivantes ont été considérées comme importantes pour prédire la valeur à vie du client :

— Total des achats
— Fréquence des commandes
— Durée du client

```
assembler = VectorAssembler(inputCols=["Total Purchases", "Order Frequency", "Customer Duration"], outputCol="features")
```

FIGURE 21 – données utilisés

— **Prétraitement des Données**

Avant de construire nos modèles, nous avons effectué les étapes de prétraitement suivantes sur nos données :

— Création d'un vecteur de caractéristiques à partir des caractéristiques sélectionnées
— Standardisation des caractéristiques pour certains modèles

```
assembler = VectorAssembler(inputCols=["Total Purchases", "Order Frequency", "Customer Duration"], outputCol="features")

csvdf = assembler.transform(csvdf)
# Diviser les données en ensembles d'entrainement et de test (80% pour l'entrainement, 20% pour le test)
train_data, test_data = csvdf.randomSplit([0.8, 0.2])
```

FIGURE 22 – prétraitement des données avant l'entrainement

— **Résultats et Évaluation**

Régression Linéaire :

— RMSE : 11556.866374980169

— R-squared (R^2) : 0.648539664218625

GBTRegressor :

— RMSE : 11282.672954281825

— MAE : 8145.781056261307

Les résultats montrent que le modèle GBTRegressor a légèrement amélioré les performances par rapport au modèle de régression linéaire avec une réduction de l'erreur quadratique moyenne (RMSE) et de l'erreur absolue moyenne (MAE). Le modèle GBTRegressor a fourni des prédictions plus précises de la CLV.

4.3.5 Prédiction du Taux de Churn des Clients

Dans cette section de notre analyse, nous avons cherché à prédire le taux de churn des clients, c'est-à-dire la probabilité qu'un client cesse d'utiliser nos services ou achète nos produits. Pour cela, nous avons utilisé les modèles de Classificateur de forêt aléatoire et Classificateur GBT.

— **Modèle Utilisé**

Nous avons utilisé deux modèles pour prédire le taux de churn des clients : le classificateur de forêt aléatoire et le classificateur GBT (Gradient Boosted Trees). Ces modèles sont des choix courants pour les problèmes de classification binaire, comme la prédiction du churn, en raison de leur capacité à gérer des ensembles de données complexes et à fournir des prédictions précises.

— **Variables Importantes**

Les variables suivantes ont été considérées comme importantes pour prédire le taux de churn des clients :

— Risque de retard de livraison
— Fréquence de commande
— Client à valeur élevée
— **Prétraitement des Données**

Avant de construire notre modèle, nous avons effectué les étapes de prétraitement suivantes sur nos données :

— Définition des conditions pour déterminer le churn en fonction de plusieurs facteurs tels que le risque de retard de livraison, la fréquence de commande et le statut de client à valeur élevée.
— Suppression des lignes avec des valeurs manquantes dans les variables sélectionnées.
— Vectorisation des caractéristiques sélectionnées pour créer un vecteur de caractéristiques pour l'entraînement du modèle.
— **Résultats et évaluation**

Les résultats montrent que les deux modèles ont obtenu des performances élevées dans la prédiction du churn, avec le classificateur GBT surpassant légèrement le classificateur de forêt aléatoire en termes d'aire sous la courbe ROC (AUC).

Forêt Aléatoire

Area Under ROC : 0.9928

GBT (Gradient-Boosted Trees)

Area Under ROC : 0.9999

Ces modèles ont démontré leur efficacité pour prédire le churn des clients, ce qui peut aider l'entreprise à cibler les clients à risque et à mettre en place des stratégies de rétention appropriées.

4.3.6 Segmentation des Clients

Dans cette section, nous avons utilisé des techniques de clustering pour segmenter nos clients en groupes homogènes en fonction de certaines caractéristiques. Nous avons utilisé deux approches de clustering différentes : K-means et le clustering hiérarchique (Bisecting K-means).

— **Modèle Utilisé**

Nous avons utilisé deux modèles de clustering différents :

1. K-means : Il s'agit d'une méthode de clustering partitionnelle qui divise les données en k groupes distincts, où chaque observation appartient au groupe le plus proche du centre de gravité (centroid).
2. Clustering hiérarchique (Bisecting K-means) : Cette méthode commence par un seul cluster qui contient tous les points, puis divise récursivement ce cluster en sous-clusters jusqu'à ce que le nombre spécifié de clusters soit atteint.

— **Variables Importantes**

Les variables suivantes ont été utilisées pour segmenter les clients :

— Risque de retard de livraison
— Fréquence de commande
— Client à valeur élevée

— **Prétraitement des Données**

Avant de procéder à la segmentation, nous avons :

— Supprimé les lignes contenant des valeurs manquantes dans les variables pertinentes.
— Vectorisé les caractéristiques sélectionnées pour créer des vecteurs de caractéristiques utilisés par les algorithmes de clustering.

```
# Define features relevant for segmentation
segmentation_features = ["Late_delivery_risk", "Order Frequency", "High_Value_Customer"]
segmentation_data = csvdf.dropna(subset=segmentation_features)

# Vectorize features
segmentation_assembler = VectorAssembler(inputCols=segmentation_features, outputCol="segmentation_features")
segmented_data = segmentation_assembler.transform(segmentation_data)
```

FIGURE 23 – code de prétraitement des données avant l'entrainement

— **Résultats et évaluation**

Pour le modèle K-means, le coefficient de silhouette a été calculé pour évaluer la cohésion intra-cluster et la séparation inter-cluster. Le coefficient de silhouette obtenu était de 0.660, ce qui indique une bonne séparation entre les clusters.

Pour le clustering hiérarchique, le coefficient de silhouette était légèrement inférieur à 0.543, mais il indique toujours une bonne séparation entre les clusters. Les centres de cluster ont été analysés pour comprendre les caractéristiques de chaque segment de client, et les clients ont été regroupés en trois catégories principales :

— Client Régulier
— Client à Risque
— Client VIP

Ces catégories fournissent des informations précieuses sur les caractéristiques et les comportements des différents segments de clients, ce qui peut être utilisé pour personnaliser les stratégies de marketing et de service client.

4.3.7 Prédiction de la Quantité Totale

— **Modèles Utilisés**

1. Régression Linéaire La régression linéaire est un modèle classique qui tente de modéliser la relation linéaire entre les caractéristiques d'entrée et la variable cible. Dans ce cas, nous avons utilisé la régression linéaire pour prédire la quantité totale de produits vendus en fonction du mois, de l'année et de la région de la commande.
2. Forêt Aléatoire La forêt aléatoire est un modèle d'ensemble qui combine plusieurs arbres de décision pour obtenir des prédictions plus robustes. Chaque arbre de décision est entraîné sur un échantillon aléatoire du jeu de données et à chaque étape de la décision, le modèle choisit la meilleure prédiction par vote majoritaire. Dans notre cas, nous avons utilisé la forêt aléatoire pour prédire la quantité totale de produits vendus.
3. Gradient Boosting Le gradient boosting est une technique d'ensemble où les modèles sont construits séquentiellement, chaque modèle tentant de corriger les erreurs des modèles précédents. Dans notre cas, nous avons utilisé le gradient boosting pour prédire la quantité totale de produits vendus en fonction du mois, de l'année et de la région de la commande.
4. Réseaux de Neurones (Deep Learning) Les réseaux de neurones sont des modèles d'apprentissage profond qui peuvent apprendre des représentations complexes des données. Nous avons utilisé un réseau de neurones séquentiel avec plusieurs couches denses pour prédire la quantité totale de produits vendus en fonction du mois, de l'année et de la région de la commande.

— **Variables Importantes**

Les variables importantes pour la prédiction de la quantité totale de produits vendus sont :

— Mois de la commande
— Année de la commande
— Région de la commande
— **Prétraitement des Données**

Avant d'entraîner les modèles, nous avons effectué les étapes suivantes de prétraitement des données :

— Suppression des lignes contenant des valeurs manquantes pour garantir la qualité des données.
— Encodage des variables catégorielles en variables binaires pour les modèles qui nécessitent des données numériques.
— **Résultats et Évaluation**

Les performances des différents modèles ont été évaluées en utilisant plusieurs métriques, notamment l'Erreur Quadratique Moyenne (RMSE) et l'Erreur Absolue Moyenne (MAE) sur les ensembles d'entraînement et de test.

— **Régression Linéaire :**
RMSE sur l'ensemble d'entraînement : 0.238 MAE sur l'ensemble d'entraînement : 0.138

— **Forêt Aléatoire :**
RMSE sur l'ensemble d'entraînement : 0.139 MAE sur l'ensemble d'entraînement : 0.071

— **Gradient Boosting :**
RMSE sur l'ensemble d'entraînement : 6.84e-06 MAE sur l'ensemble d'entraînement : 3.52e-06

— **Réseaux de Neurones (Deep Learning) :**
RMSE sur l'ensemble de test : 9.052 MAE sur l'ensemble de test : 4.373

Les résultats indiquent que le modèle de régression linéaire et celui de forêt aléatoire ont des performances similaires, avec des erreurs moyennes respectables sur l'ensemble d'entraînement. Cependant, le modèle de réseaux de neurones (Deep Learning) montre une erreur significativement plus élevée sur l'ensemble de test, ce qui suggère qu'il pourrait avoir du mal à généraliser aux nouvelles données. En revanche, le modèle de gradient boosting a atteint des performances exceptionnelles sur l'ensemble d'entraînement, avec des erreurs pratiquement nulles, ce qui souligne sa capacité à bien s'adapter aux données d'entraînement. Cependant, il serait nécessaire de vérifier si ces résultats se généralisent également à de nouvelles données non vues.

4.4 Résultats des Prédictions

4.4.1 Résultats des Prédictions

Les résultats des prédictions sont analysés en fonction de trois principaux aspects : les commandes, les clients et les quantités de produits prédites.

Fichiers CSV de Résultats : Nous avons généré plusieurs fichiers CSV contenant les résultats de nos prédictions :

— orders_predictions.csv : Contient les prédictions relatives aux commandes, y compris la performance de livraison et la rentabilité.

	Order Id	Predicte Late Delivery Risk	Optimal_Delivery_Mode_Prediction	Order_Profit_Per_Order_Prediction
0	57963	Not late	Standard Class	20.48183
1	57963	Not late	Standard Class	20.48183
2	57963	Not late	Standard Class	20.48183
3	57963	Not late	Standard Class	20.48183
4	57963	Not late	Standard Class	20.48183

FIGURE 24 – fichier csv des predictions orders

— customers_predictions.csv : Comprend les prédictions liées aux clients, telles que la valeur à vie du client et le taux de churn.

	Customer Id	Customer Lifetime Value prediction	Prediction_Label	Customer sgmentation Prediction
0	2	61334.367518	Not Churn	At Risk Customer
1	2	61334.367518	Not Churn	At Risk Customer
2	2	61334.367518	Not Churn	At Risk Customer
3	2	61334.367518	Not Churn	At Risk Customer
4	2	61334.367518	Not Churn	At Risk Customer

FIGURE 25 – fichier csv des predictions sur les client

— quantity_predictions.csv : Inclut les prédictions de quantités de produits pour chaque mois et région en 2019.

	Product_Name	Month	Order_Region	Predicted_Quantity
0	Diamondback Women's Serene Classic Comfort Bi	1	Central America	16.669789
1	Diamondback Women's Serene Classic Comfort Bi	1	South Asia	16.669789
2	Diamondback Women's Serene Classic Comfort Bi	1	Northern Europe	16.669789
3	Diamondback Women's Serene Classic Comfort Bi	1	Eastern Asia	16.669789
4	Diamondback Women's Serene Classic Comfort Bi	1	Southern Europe	16.669789

FIGURE 26 – fichier csv des predictions des quantitiès des produits

4.4.2 Discussion sur les Résultats Obtenu

Dans cette section, nous discuterons des résultats obtenus à partir de nos analyses et prédictions. Nous aborderons l'interprétation des prédictions, la pertinence des modèles utilisés et les implications des résultats pour la prise de décision stratégique.

— **Interprétation des Prédictions**

Les prédictions de la quantité de produits vendus par mois et la performance des produits offrent des insights précieux sur les tendances de vente et la popularité des produits. Voici les principales observations :

1. Quantité de Produits Vendus par Mois :

   Les résultats montrent des variations dans les quantités de produits vendus au cours de l'année 2019. L'analyse mensuelle révèle des tendances saisonnières qui peuvent être cruciales pour la planification des stocks et la gestion des ressources.

Par exemple, des pics de vente peuvent indiquer des périodes de forte demande, nécessitant une augmentation de la production ou des stocks.

2. Performance des Produits :

La part de marché de chaque produit, visualisée sous forme de diagramme circulaire, indique quels produits dominent les ventes. Cette information est essentielle pour les stratégies de marketing et de vente. Les produits avec une part de marché élevée devraient être promus davantage, tandis que les produits avec une part de marché plus faible pourraient nécessiter des stratégies d'amélioration ou de réévaluation.

— **Pertinence des Modèles Utilisés**

Les modèles de régression et de clustering utilisés dans nos analyses ont été choisis pour leur capacité à traiter des ensembles de données complexes et à fournir des prédictions précises. Les principaux modèles utilisés incluent la régression logistique, les arbres de décision, RandomForestRegressor, GBTRegressor, et K-means.

1. Prédiction de la Performance de Livraison :

   Les modèles de régression logistique et d'arbres de décision ont montré une haute précision (97,3) pour prédire la probabilité de retard de livraison. Ces modèles sont efficaces pour capturer les relations entre les variables explicatives et la performance de livraison, fournissant ainsi des prédictions fiables qui peuvent être utilisées pour améliorer les processus logistiques.

2. Prédiction de la Rentabilité des Commandes :

   Les modèles RandomForestRegressor et GBTRegressor ont montré des performances solides pour la prédiction de la rentabilité des commandes, avec le modèle GBT offrant la meilleure précision (RMSE le plus bas). Ces modèles sont particulièrement adaptés pour gérer des relations complexes et non linéaires entre les variables, offrant des insights détaillés sur les facteurs influençant la rentabilité.

3. Optimisation du Mode de Livraison :

les modèles LogisticRegression et NaiveBayes ont démontré leur capacité à regrouper les données de manière optimale, bien que les performances de LogisticRegression soient inférieures à celles de NaiveBayes. Cependant, les deux ont permis une gestion logistique plus efficace en recommandant le mode de livraison le plus approprié pour chaque segment de commandes.

— **Comparaison avec les Données Réelles**

Les résultats des prédictions ont été comparés avec les visualisations des données réelles obtenues lors de l'analyse exploratoire des données (EDA). Cette comparaison est essentielle pour évaluer la précision et la fiabilité des modèles de prédiction utilisés.

1. Quantité de Produits Vendus par Mois :

   Les prédictions de la quantité de produits vendus par mois montrent des tendances similaires à celles observées dans les données réelles. Les variations saisonnières et les pics de vente identifiés dans les prédictions correspondent bien aux tendances historiques, confirmant la robustesse des modèles de prédiction.

2. Performance des Produits :

La distribution des parts de marché des produits prédite est cohérente avec les données réelles observées lors de l'EDA. Les produits les plus populaires selon les prédictions sont également ceux qui ont montré des ventes élevées dans les données historiques, renforçant la validité des analyses effectuées.

— **Implications des Résultats**

Les résultats de nos analyses ont des implications importantes pour la prise de décision stratégique. En comprenant les tendances de vente et la performance des produits, l'entreprise peut mieux planifier ses stocks, optimiser ses stratégies de marketing et améliorer sa satisfaction client. De plus, les prédictions précises de la performance de livraison et de la rentabilité des commandes permettent une gestion plus efficace des ressources et une amélioration des processus opérationnels.

En conclusion, les modèles utilisés et les résultats obtenus offrent des insights précieux qui peuvent être utilisés pour améliorer la performance globale de l'entreprise. L'intégration de ces analyses dans Power BI permet une visualisation claire et une interprétation facile des données, facilitant ainsi une prise de décision informée et stratégique.

5 Chapitre 5 Mise en Œuvre et Visualisation

5.1 Introduction

Dans cette section, nous détaillerons la manière dont les résultats de nos analyses et prédictions ont été intégrés dans un tableau de bord interactif utilisant Power BI. L'objectif principal est de présenter de manière visuelle et intuitive les insights obtenus à partir des données, permettant ainsi une prise de décision éclairée pour les parties prenantes.

5.2 Intégration des Résultats dans Power BI

Pour intégrer les résultats de nos prédictions dans Power BI, nous avons utilisé les fonctionnalités de liaison de données de Power BI pour connecter notre fichier CSV contenant les résultats de prédiction au tableau de bord Power BI. En utilisant des visualisations telles que des graphiques, des tableaux croisés dynamiques et des cartes interactives, nous avons représenté visuellement les résultats de nos prédictions pour une analyse plus approfondie.

5.3 Conception du Dashboard

Le dashboard dans Power BI a été conçu pour présenter de manière intuitive et informative les résultats des prédictions. Les différentes visualisations ont été organisées de manière logique pour permettre aux utilisateurs de comprendre facilement les tendances, les modèles et les insights issus des données prédites. Des filtres interactifs ont été ajoutés pour permettre aux utilisateurs de personnaliser leur expérience et d'explorer les données selon leurs besoins spécifiques. Inclusion des Prédictions et Visualisations

5.4 Inclusion des Prédictions et Visualisations

Dans le dashboard Power BI, nous avons inclus les prédictions de performance de livraison, de rentabilité des commandes, de valeur à vie du client, de quantité totale, ainsi que d'autres variables pertinentes. Ces prédictions ont été accompagnées de visualisations appropriées telles que des graphiques à barres, des diagrammes circulaires et des cartes géographiques pour une meilleure compréhension des données.

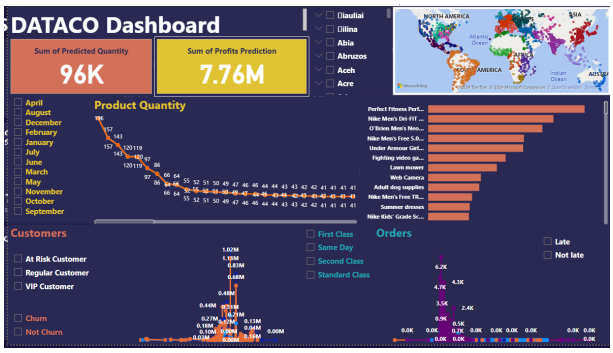


FIGURE 27 – visualisations des données dans Power Bi

5.5 Avantages et Applications Pratiques

L'intégration de ces visualisations dans Power BI offre de nombreux avantages pratiques pour l'entreprise. Elle permet une prise de décision plus rapide et plus précise en offrant une vue d'ensemble complète et en temps réel des opérations et des performances commerciales. Les équipes peuvent facilement identifier les domaines nécessitant des améliorations et prendre des mesures proactives pour optimiser les processus de livraison, augmenter la rentabilité des commandes et améliorer la satisfaction client. De plus, l'accès à des données prédictives permet aux responsables de planifier des stratégies à long terme basées sur des analyses approfondies des tendances passées et des prévisions futures.

En conclusion, l'utilisation de Power BI pour la visualisation et l'analyse des données prédictives transforme des ensembles de données complexes en informations claires et exploitables. Cette approche améliore non seulement la compréhension des dynamiques commerciales, mais facilite également la mise en œuvre de stratégies efficaces pour atteindre les objectifs de l'entreprise.

6 Conclusion et Perspectives

6.1 Conclusion

Cette étude a permis de développer une solution robuste pour la prévision de la demande d'expédition et de distribution, intégrant des techniques avancées de traitement et de modélisation des données. En utilisant un pipeline ETL, nous avons pu collecter, transformer et charger des données historiques d'expédition pour construire des modèles prédictifs fiables. Ces modèles, intégrés dans des tableaux de bord interactifs via Power BI, offrent des insights précieux pour optimiser la gestion des stocks, réduire les coûts opérationnels et garantir des livraisons en temps voulu. Les résultats montrent que l'utilisation de modèles tels que la régression logistique, les arbres de décision et les algorithmes de clustering peut améliorer considérablement l'efficacité opérationnelle et la satisfaction des clients.

6.2 Perspectives

1. Amélioration des Modèles Prédictifs
   - Inclusion de Nouvelles Variables : Intégrer des variables supplémentaires, telles que les tendances macroéconomiques et les données météorologiques, pourrait améliorer la précision des modèles.
   - Exploration des Techniques de Deep Learning : L'utilisation de réseaux de neurones avancés et de techniques de deep learning pourrait permettre de capturer des relations non linéaires et complexes dans les données.

2. Augmentation de la Taille et de la Qualité des Données :
   - Collecte de Données en Temps Réel : Intégrer des flux de données en temps réel pour une analyse et une prise de décision plus réactive.
   - Amélioration de la Qualité des Données : Mettre en place des mécanismes de nettoyage et de validation des données plus rigoureux pour assurer la qualité et la fiabilité des données utilisées.

3. Expansion des Capacités de Visualisation :
   - Tableaux de Bord Personnalisables : Développer des fonctionnalités permettant aux utilisateurs de personnaliser leurs tableaux de bord selon leurs besoins spécifiques.
   - Intégration de Technologies de Réalité Augmentée : Explorer l'utilisation de la réalité augmentée pour la visualisation des données, offrant ainsi une interaction plus immersive et intuitive avec les données.

4. Optimisation Continue des Processus Logistiques :
   - Approches Basées sur l'Intelligence Artificielle : Utiliser des techniques d'IA pour optimiser en continu les processus logistiques, en adaptant les stratégies en fonction des nouvelles données et des prévisions mises à jour.
   - Collaboration Interdépartementale : Encourager une collaboration étroite entre les départements de l'entreprise pour intégrer les insights prédictifs dans les

décisions stratégiques globales.

5. Impact Écologique et Durable :
   — Réduction de l'Empreinte Carbone : Utiliser les prédictions pour optimiser les routes de livraison et réduire les émissions de carbone associées aux opérations logistiques.
   — Soutien à la Durabilité : Développer des stratégies qui intègrent des pratiques durables et respectueuses de l'environnement, en utilisant les insights pour minimiser les déchets et optimiser l'utilisation des ressources.

En conclusion, ce projet ne se contente pas de répondre aux besoins immédiats de prévision de la demande, mais pose également les bases pour des améliorations continues et une adaptation future aux évolutions technologiques et environnementales. L'intégration de nouvelles techniques et l'amélioration continue des processus permettront de maintenir et d'améliorer l'efficacité opérationnelle et la compétitivité de l'entreprise.

7 Bibliographie

[1] Brownlee, Jason. Machine Learning Mastery with Python : Understand Your Data, Create Accurate Models, and Work Projects End-To-End. Machine Learning Mastery, 2016.
https ://machinelearningmastery.com/

[2] Han, Jiawei, Kamber, Micheline, and Pei, Jian. Data Mining : Concepts and Techniques. 3rd ed., Elsevier, 2011.
https ://www.elsevier.com/books/data-mining/han/978-0-12-381479-1

[3] Chollet, François. Deep Learning with Python. Manning Publications, 2018.
https ://www.manning.com/books/deep-learning-with-python

[4] Wickham, Hadley, and Grolemund, Garrett. R for Data Science : Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media, 2017.
https ://www.amazon.com/R-Data-Science-Transform-Visualize/dp/1491910399

[5] Grover, Varun, and Kohli, Rajiv. Business Process Transformation. Routledge, 2012.
https ://www.routledge.com/Business-Process-Transformation/Grover-Kohli/p/book/9780415896520

[6] Dean, Jeff, et al. "Large Scale Distributed Deep Networks." Advances in Neural Information Processing Systems, vol. 25, 2012, pp. 1223-1231.
https ://papers.nips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html

[7] Pedregosa, Fabian, et al. "Scikit-learn : Machine Learning in Python." Journal of Machine Learning Research, vol. 12, 2011, pp. 2825-2830 .
http ://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[8] Chen, Tianqi, and Guestrin, Carlos. "XGBoost : A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794 .
https ://dl.acm.org/doi/10.1145/2939672.2939785