

Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

**DIPLOME NATIONAL D'INGENIEUR EN INFORMATIQUE**

Université Ibn Khaldoun - UIK



**Projet intégré multidisciplinaire**

*Préparé en vue de valider un module universitaire*

---

**CLASSIFICATION DES IMAGES DE  
FASHION\_MNIST**

---

Réalisé par :

Felah Amen Allah

Hannachi Ayoub

Encadré par :

Mme. Jlassi Amal (UIK)

Mme. Khlifa Nawres (UIK)

**Année Universitaire 2023-2024**

# Remerciements

Au terme de ce travail, aussi modeste soit-il, nous souhaitons exprimer nos sincères remerciements à toutes les personnes qui nous ont soutenus de près ou de loin.

Nous adressons tout particulièrement notre profonde gratitude à Mesdames Jlassi Amal et Khlifa Nawres, nos encadrantes à l'Université Ibn Khaldoun. Leur rigueur, disponibilité, patience, et précieuse expertise ont été pour nous une source d'enrichissement inestimable. Nous les remercions chaleureusement pour le temps qu'elles ont consacré à l'encadrement et au suivi de ce travail, pour les conseils avisés qu'elles nous ont prodigués, ainsi que pour leur engagement lors des nombreuses réunions qui ont marqué les différentes étapes de la réalisation de ce projet.

Enfin, nous souhaitons exprimer nos remerciements les plus sincères aux membres du jury pour l'honneur qu'ils nous font en acceptant de juger ce travail. Leur bienveillance et leurs retours seront pour nous une grande source d'apprentissage et de motivation.

# Résumé

La mode est un secteur dynamique dans lequel la technologie joue un rôle essentiel pour répondre aux attentes croissantes des consommateurs. Grâce au Machine Learning (ML), il est désormais possible de traiter des ensembles de données complexes pour automatiser des tâches telles que la reconnaissance des vêtements, la personnalisation des recommandations et la gestion des stocks. Ces avancées permettent d'améliorer l'efficacité des processus tout en offrant une expérience client plus fluide.

Dans le domaine de la reconnaissance des vêtements, par exemple, le ML permet de classer les articles en fonction de leurs caractéristiques, comme les motifs, les textures ou les couleurs. Cette fonctionnalité est particulièrement utile pour les plateformes de commerce en ligne, où une organisation claire des produits facilite la recherche des utilisateurs. De plus, en analysant les comportements d'achat et les préférences des clients, les systèmes de recommandation basés sur le ML peuvent suggérer des articles compatibles, renforçant ainsi la satisfaction et la fidélisation.

Le ML permet également d'optimiser les chaînes logistiques et la gestion des stocks en prédisant les tendances de la demande. En analysant les données issues des ventes et des médias sociaux, les marques peuvent ajuster leurs collections pour mieux répondre aux attentes des consommateurs. Cette capacité à anticiper les besoins et à rationaliser les processus fait du ML un outil essentiel pour moderniser l'industrie de la mode et maintenir sa compétitivité.

# Abstract

Fashion is a dynamic sector where technology plays a crucial role in meeting the growing demands of consumers. Thanks to machine learning (ML), it is now possible to process complex data sets to automate tasks such as garment recognition, personalised recommendations and inventory management. These advances improve process efficiency while delivering a smoother customer experience.

In the area of apparel recognition, ML enables items to be classified based on their characteristics, such as patterns, textures or colours. This is particularly useful for online shopping platforms, where a clear organisation of products makes it easier for users to find what they're looking for. In addition, by analysing purchasing behaviour and customer preferences, ML-based recommendation systems can suggest compatible items, increasing customer satisfaction and loyalty.

ML also helps optimise supply chains and inventory management by predicting demand trends. By studying data from sales and social media, brands can adapt their collections to better meet consumer expectations. This ability to anticipate needs and streamline processes makes ML a key tool for modernising the fashion industry and maintaining its competitiveness.

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Contexte du projet et mise en évidence</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Présentation du Projet . . . . .	3
1.3 Qu'est-ce-que le mode(fashion) ? . . . . .	4
1.4 L'impact de la Mode sur la Société . . . . .	4
1.5 Les Causes et Facteurs de Risques dans le Secteur de la Mode . . . . .	5
1.6 Objectifs . . . . .	5
1.7 Problématique . . . . .	6
1.8 Conclusion . . . . .	6
<b>2 Description de la dataset et pré-traitement des données</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Les Outils Matériels et logiciels . . . . .	7
2.2.1 Partie logicielle . . . . .	7
2.2.2 Partie matérielle . . . . .	8
2.3 Description de DataSet . . . . .	8
2.3.1 Contenu du Dataset . . . . .	8
2.3.2 Structure du Dataset . . . . .	9
2.3.3 Taille et Variété . . . . .	10
2.4 Prétraitement des données . . . . .	10
2.5 Collection des données . . . . .	11
2.6 Conclusion . . . . .	12

<b>3 Entrainement des modèles et résultats</b>	<b>13</b>
3.1 Introduction . . . . .	13
3.2 Méthodologie . . . . .	13
3.2.1 Algorithmes de Classification . . . . .	13
3.3 1. Évaluation des Modèles avec des Visualisations . . . . .	14
3.3.1 Comparaison des Modèles . . . . .	14
3.3.2 Matrice de Confusion . . . . .	14
3.3.3 Rapport de Classification . . . . .	18
3.3.4 Courbes ROC et AUC . . . . .	19
3.3.5 GridSearch et RandomizedSearch . . . . .	19
3.4 Division 80% train, 20% test et augmentation des données . . . . .	20
3.4.1 Division 80% train 20% test . . . . .	20
3.4.2 Augmentation des données . . . . .	21
3.4.3 Entraînement sur Données Augmentées et Résultats . . . . .	22
3.5 Suppression de la Classe Chemise . . . . .	23
3.5.1 Motivation et Raisons de la Suppression . . . . .	23
3.5.2 Résultats après Suppression . . . . .	23
3.6 Suppression de la Classe Chemise avec Division 80% Train et 20% Test . . . . .	25
3.6.1 Division du Dataset après Suppression . . . . .	25
3.6.2 Résultats après Entraînement . . . . .	25
3.6.3 Analyse des Résultats . . . . .	26
3.6.4 Validation croisée . . . . .	27
3.7 Conclusion . . . . .	28

# Table des figures

1.1	Exemple des vêtements . . . . .	4
2.1	Quelques images de chaque classe . . . . .	9
2.2	Représente le code python utilisé pour importer les bibliothéques . . . . .	10
2.3	Chargement du DataSet Fashion_MNIST . . . . .	11
2.4	Structure des classes du DataSet . . . . .	11
3.1	Matrice de confusion du Logistic Regression . . . . .	15
3.2	Matrice de confusion du KNN . . . . .	15
3.3	Matrice de confusion du Decision Tree . . . . .	16
3.4	Matrice de confusion du SVM . . . . .	16
3.5	Matrice de confusion du Naive Bayes . . . . .	17
3.6	Matrice de confusion du Random Forest . . . . .	17
3.7	Matrice de confusion du XGBoost . . . . .	18
3.8	Courbe ROC du SVM . . . . .	19
3.9	Courbe ROC du XGBoost . . . . .	19
3.10	Division 80% 20% . . . . .	20
3.11	Matrice confusion du svm . . . . .	21
3.12	Matrice confusion du xgboost . . . . .	21
3.13	Illustration des données après augmentation . . . . .	21
3.14	Chemises . . . . .	23
3.15	Autres Chemises . . . . .	23
3.16	Matrice confusion du svm . . . . .	24
3.17	Matrice confusion du xgboost . . . . .	24
3.18	Division data 80 20 . . . . .	25

3.19 Matrice confusion du svm . . . . .	26
3.20 Matrice confusion du xgboost . . . . .	26

# Liste des tableaux

3.1	Comparaison des modèles de classification . . . . .	14
3.2	Classification report SVM et XGBoost . . . . .	18
3.3	Classification Report SVM et XGBoost devision 80 20 . . . . .	22
3.4	Classification Report SVM et XGBoost sans chemises . . . . .	24
3.5	Classification Report SVM, XGBoost . . . . .	26

# Introduction générale

Dans un monde de plus en plus numérique, les méthodes d'apprentissage automatique jouent un rôle central dans des domaines aussi divers que la vision par ordinateur, la reconnaissance vocale et l'analyse de données complexes. La classification d'images est l'une des applications les plus importantes de ces technologies, aidant à résoudre des problèmes tels que la reconnaissance faciale, l'identification de produits et le diagnostic médical.

Le jeu de données Fashion MNIST, une version améliorée du célèbre jeu de données MNIST, propose un ensemble d'images représentant des articles vestimentaires et des accessoires. Conçu pour des tâches de classification, il est largement utilisé comme banc d'essai pour les algorithmes d'apprentissage automatique et les modèles de deep learning.

L'objectif principal de ce projet est de comparer différentes approches de classification sur le Fashion MNIST afin de déterminer celle qui offre les meilleures performances. Plusieurs méthodes sont explorées, allant des modèles classiques tels que les K-Nearest Neighbors (KNN) et les Machines à Vecteurs de Support (SVM) à des techniques avancées comme XGBoost. Ces modèles sont évalués selon divers critères, notamment la précision, la robustesse et la complexité computationnelle.

Le présent rapport propose une analyse détaillée des méthodes utilisées, une discussion approfondie sur leurs performances respectives et des recommandations pour leur utilisation dans des contextes similaires. Par ailleurs, l'efficacité des modèles est également améliorée par des approches complémentaires telles que la réduction de dimensionnalité par analyse en composantes principales.

En combinant des analyses qualitatives et quantitatives, ce travail vise à fournir une base solide

pour mieux comprendre les forces et les limites des différentes approches de classification, tout en explorant des solutions optimales adaptées à des tâches complexes de vision par ordinateur.

Le présent rapport présente notre travail et est structuré comme suit :

- Le chapitre 1 expose Contexte du projet et mise en évidence.
- Le chapitre 2 présente Description de la dataset et pré-traitement des données.
- Le chapitre 3 présente l'entraînement des modèles et résultats.

Nous clôturons le rapport en résumant les principales contributions et en ouvrant quelques perspectives qui peuvent l'enrichir.

# Chapitre 1

## Contexte du projet et mise en évidence

### 1.1 Introduction

La classification d’images est un défi majeur en apprentissage automatique avec de nombreuses applications dans des domaines variés. Ce projet vise à explorer différentes méthodes de classification supervisée appliquées au jeu de données mode MNIST, qui contient 70 000 images représentant des vêtements. À l’aide de modèles classiques tels que SVM, XGBoost, Random Forest, KNN, Decision Tree, Naive Bayes et Logistic Regression, nous comparons leurs performances sur cette tâche. Nous discutons également des techniques de traitement des données et de visualisation avancée pour mieux comprendre et évaluer ces modèles.

### 1.2 Présentation du Projet

L’ensemble de données Fashion MNIST contient des images en niveaux de gris de 28 x 28 pixels, chaque image correspondant à l’une des 10 catégories de produits de mode. Chaque catégorie est bien représentée, ce qui garantit l’équilibre des données. 60 000 images de formation et 10 000 images de test vous permettent de concevoir des modèles robustes et de les évaluer rigoureusement. Utilisez une approche systématique pour prétraiter les données, tester plusieurs algorithmes de classification et ajuster leurs hyperparamètres.

### 1.3 Qu'est-ce-que le mode(fashion) ?

La mode fait référence à un ensemble de tendances et de styles qui influencent les vêtements, les accessoires et l'apparence personnelle. C'est un moyen d'expression individuelle et collective qui reflète les goûts, les valeurs et les préoccupations d'une époque. La mode évolue avec les saisons, les innovations et les influences sociales et culturelles. Il comprend les vêtements, les accessoires, le maquillage et la coiffure. La mode joue également un rôle clé dans l'industrie mondiale, influençant la production, la commercialisation et les choix des consommateurs. C'est un véhicule de statut, de confort et de statut social.



FIGURE 1.1 – Exemple des vêtements

### 1.4 L'impact de la Mode sur la Société

La mode joue un rôle essentiel dans la société contemporaine, influençant non seulement l'apparence mais aussi le comportement et les valeurs culturelles. C'est un véhicule d'expression personnelle, permettant aux individus de s'affirmer et de se démarquer dans un monde standardisé. Outre son rôle esthétique, la mode a des impacts sociaux et économiques importants. Il soutient des milliers d'industries, de la production de tissus à la commercialisation des collections. De plus, la mode a évolué avec l'avènement de la technologie, l'intelligence artificielle et la réalité augmentée transformant désormais l'expérience d'achat et de personnalisation

de vêtements. En conséquence, la mode n'est plus seulement une question d'apparence, mais un moyen de se connecter aux communautés, d'exprimer son identité et de participer aux mouvements sociaux.

## 1.5 Les Causes et Facteurs de Risques dans le Secteur de la Mode

Bien que l'industrie de la mode soit une industrie lucrative, il existe également certains facteurs de risque qui affectent les consommateurs et les fabricants. D'une part, la pression constante des tendances et des saisons entraîne une consommation rapide, ce qui peut conduire à une surproduction et au gaspillage. De plus, la fast fashion crée de graves risques environnementaux en raison de la production de masse, de l'utilisation de matériaux non durables et des conditions de travail des employés dans certaines régions du monde. En outre, le secteur est confronté à des risques économiques, notamment les fluctuations des prix des matières premières, les changements dans les préférences des consommateurs et la concurrence des plateformes de commerce électronique. Face à ces enjeux, il devient crucial pour l'industrie de s'adapter à des pratiques plus responsables, en intégrant la durabilité et en utilisant les avancées technologiques pour anticiper les besoins des consommateurs tout en réduisant l'impact environnemental.

## 1.6 Objectifs

L'objectif principal est de comparer plusieurs modèles classiques d'apprentissage supervisé afin de déterminer quel modèle offre la meilleure précision et robustesse sur cet ensemble de données. Utilisez également des techniques telles que la division de l'ensemble de données (80/20), la validation croisée et l'augmentation des données pour optimiser vos résultats. Ce travail comprend également une analyse approfondie des performances du modèle à l'aide de visualisations telles que des matrices de confusion, des rapports de classification et des courbes ROC/AUC.

## 1.7 Problématique

La mode est une industrie complexe avec une grande variété de produits, ce qui rend la classification difficile. Cette diversité crée des défis pour les plateformes de commerce électronique, où la recherche de biens par les consommateurs devient souvent compliquée. De plus, les recommandations personnalisées restent un problème car les systèmes traditionnels ne sont pas toujours efficaces pour prédire les préférences des utilisateurs.

Pour surmonter ces défis, l'utilisation de méthodes de classification telles que l'apprentissage automatique (ML) et les machines à vecteurs de support (SVM) constituent une solution prometteuse. SVM peut classer les éléments en fonction de caractéristiques telles que la couleur, la texture et le style, ce qui les rend plus faciles à trouver pour les utilisateurs. De plus, cette approche permet la création de systèmes de recommandation personnalisés, améliorant l'expérience d'achat et la fidélisation. Le ML, via les SVM, offre ainsi des outils puissants pour optimiser la gestion des produits et mieux répondre aux attentes des consommateurs dans l'industrie de la mode.

## 1.8 Conclusion

Dans cette étude, nous explorons l'importance croissante de l'industrie de la mode et ses interconnexions avec la technologie, notamment l'utilisation de la reconnaissance d'image et de l'intelligence artificielle pour aider les consommateurs à choisir leurs vêtements. Nous discutons également de divers facteurs et défis auxquels le domaine est confronté, tels que les risques liés à la mode, les préférences personnelles et l'évolution des tendances. L'application de modèles d'apprentissage automatique tels que la technologie d'apprentissage automatique dans la classification des vêtements a ouvert de nouvelles façons d'améliorer l'expérience d'achat en ligne. Le projet met en évidence comment la technologie peut transformer des espaces traditionnels en espaces innovants, faciliter la prise de décision des utilisateurs et offrir de nouvelles dimensions à la mode.

# **Chapitre 2**

## **Description de la dataset et pré-traitement des données**

### **2.1 Introduction**

Ce chapitre présente les principales caractéristiques de l'ensemble de données utilisé et les étapes de prétraitement de base. Ces étapes incluent la normalisation des données, l'équilibrage des classes et l'augmentation des données pour améliorer la qualité des entrées et optimiser les performances du modèle.

### **2.2 Les Outils Matériels et logiciels**

#### **2.2.1 Partie logicielle**

Ce projet utilise Google Colab comme environnement de développement. Il fournit une plate-forme de cloud computing qui permet d'exécuter des notebooks Python et d'accéder à des ressources matérielles puissantes telles que des GPU et des TPU. Cela permet aux algorithmes d'apprentissage automatique et aux modèles d'intelligence artificielle de fonctionner rapidement sur de grandes quantités de données.

## 2.2.2 Partie matérielle

Le matériel utilisé dans ce projet repose principalement sur les ressources de Google Colab. L'infrastructure cloud fournit des GPU et des TPU pour accélérer le traitement des données et la formation des modèles. Ces ressources matérielles sont essentielles à la gestion des tâches de calcul lourdes associées à la formation des modèles d'apprentissage automatique.

## 2.3 Description de DataSet

L'ensemble de données Fashion-MNIST est un ensemble d'images de produits de mode de Zalando. Il est destiné à remplacer directement l'ensemble de données MNIST d'origine pour évaluer les algorithmes d'apprentissage automatique.

### 2.3.1 Contenu du Dataset

#### 2.3.1.1 Images

- Cet ensemble de données contient des images d'articles de mode en niveaux de gris de 28 x 28 pixels.
- Chaque image est représentée par un tableau de 784 pixels, chaque pixel ayant une valeur comprise entre 0 et 255, indiquant l'intensité lumineuse.

#### 2.3.1.2 Annotations et Étiquettes

- Chaque image est étiquetée selon une classe correspondant à un type d'article de mode.
- Les étiquettes sont fournies sous forme de valeurs numériques correspondant à l'une des 10 catégories de vêtements.

#### 2.3.1.3 Classes

- 0 : T-shirt/top
- 1 : Trouser
- 2 : Pullover
- 3 : Dress
- 4 : Coat

- 5 : Sandal
- 6 : Shirt
- 7 : Sneaker
- 8 : Bag
- 9 : Ankle boot

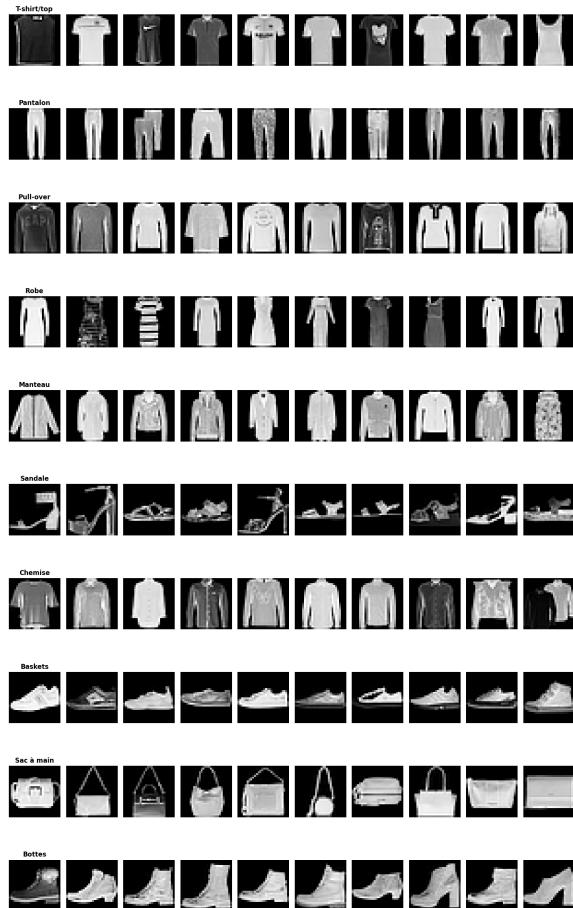


FIGURE 2.1 – Quelques images de chaque classe

### 2.3.2 Structure du Dataset

Le dataset est structuré en deux ensembles principaux :

- Train : Contient 60 000 images d’entraînement.
- Test : Contient 10 000 images de test.

Chaque ligne de ces ensembles représente une image. La première colonne contient l’étiquette (classe), et les 784 autres colonnes contiennent les valeurs des pixel

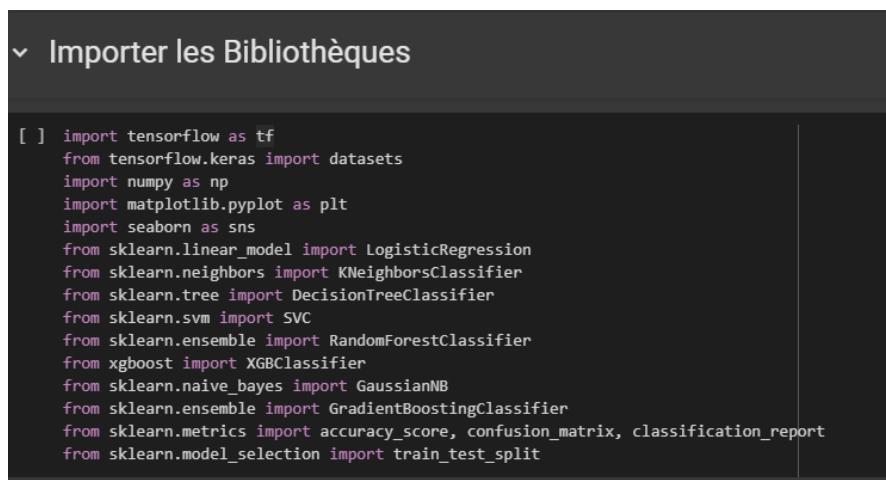
### 2.3.3 Taille et Variété

- Nombre d’images : Le dataset contient un total de 70 000 images (60 000 pour l’entraînement et 10 000 pour le test).
- Résolution : Les images ont une résolution uniforme de 28x28 pixels. Chaque pixel représente l’intensité lumineuse de l’image, ce qui en fait un excellent dataset pour les modèles de classification d’images.

## 2.4 Prétraitement des données

Le processus de préparation des données du jeu de données Fashion MNIST débute par le chargement des images de formation et de validation. Par la suite, les images sont normalisées, ce qui implique que les valeurs des pixels sont divisées par 255.0 afin de les ramener dans une fourchette de 0 à 1, ce qui améliore ainsi la convergence des modèles d’apprentissage. Les images, qui mesurent initialement 28x28, sont transformées en vecteurs de 784 éléments pour être compatibles avec les modèles classiques de machine learning. Cela permet de traiter chaque image comme une entrée vectorielle plutôt qu’une matrice. Pour finir, la structure des données est vérifiée pour s’assurer qu’elles sont prêtes à l’entraînement et à la validation des modèles. Ce traitement préalable assure que les données sont optimisées pour l’apprentissage automatique.

Tous d’abord nous avons importé les librairies :



```

▼ Importer les Bibliothèques

[ ] import tensorflow as tf
from tensorflow.keras import datasets
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import train_test_split

```

FIGURE 2.2 – Représente le code python utilisé pour importer les bibliothèques

- Lecture de DataSet :

```
# Charger le dataset Fashion MNIST
(x_train, y_train), (x_test, y_test) = datasets.fashion_mnist.load_data()

# Normaliser les images (valeurs entre 0 et 1)
x_train, x_test = x_train / 255.0, x_test / 255.0

# Applatiser les images pour les modèles scikit-learn
x_train_flat = x_train.reshape(-1, 28*28)
x_test_flat = x_test.reshape(-1, 28*28)

# Vérification de la forme des données
print(f"x_train shape: {x_train.shape}, x_test shape: {x_test.shape}")

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515    0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880    0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148    0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102    0s 0us/step
x_train shape: (60000, 28, 28), x_test shape: (10000, 28, 28)
```

FIGURE 2.3 – Chargement du DataSet Fashion\_MNIST

- Voici le nombre des fichiers pour chaque classe de DataSet :

```
Nombre d'exemples par classe dans l'ensemble d'entraînement :
Classe   Nombre d'exemples dans l'entraînement
0   T-shirt/top           6000
1   Trouser                6000
2   Pullover               6000
3   Dress                  6000
4   Coat                   6000
5   Sandal                 6000
6   Shirt                  6000
7   Sneaker                6000
8   Bag                    6000
9   Ankle boot             6000

Nombre d'exemples par classe dans l'ensemble de test :
Classe   Nombre d'exemples dans le test
0   T-shirt/top           1000
1   Trouser                1000
2   Pullover               1000
3   Dress                  1000
4   Coat                   1000
5   Sandal                 1000
6   Shirt                  1000
7   Sneaker                1000
8   Bag                    1000
9   Ankle boot             1000

x_train shape: (60000, 28, 28), x_test shape: (10000, 28, 28)
```

FIGURE 2.4 – Structure des classes du DataSet

## 2.5 Collection des données

La collecte des données Fashion MNIST a été réalisée par Zalando en 2017 et comprend 70 000 images d'articles de mode en niveaux de gris, de taille 28x28 pixels. Ces images sont réparties en 60 000 pour l'entraînement et 10 000 pour le test, et sont associées à 10 classes distinctes d'articles comme "t-shirt", "pantalon", ou "robe". Stockées au format PNG, elles as-

surent une compression sans perte. Cet ensemble de données est une référence en apprentissage automatique, notamment pour la classification d'images.

#### 2.5.0.1 Analyse de performance

L'ensemble des métriques sont utilisées pour évaluer les méthodes d'apprentissage automatique. À partir de la matrice de confusion, de nombreuses mesures de performance du modèle peuvent être dérivées, parmi lesquelles la précision est la plus populaire, qui est définie comme : **Accuracy** : correspond à tous les modèles correctement classés divisés par le nombre total de modèles.

$$\text{Accuracy} = \frac{VP + VN}{VP + FP + FN + VN} \quad (2.1)$$

**Précision** : Cela définit l'exactitude du modèle en termes de prédiction.

$$\text{Précision} = \frac{VP}{VP + FP} \quad (2.2)$$

$$\text{Précision} = \frac{VN}{VN + FN} \quad (2.3)$$

**Recall (Sensibilité)** : Cette mesure de performance implique comment différentes valeurs et variables indépendantes affectent une variable dépendante.

$$\text{Recall} = \frac{VP}{VP + FN} \quad (2.4)$$

$$\text{Recall} = \frac{VN}{VN + FP} \quad (2.5)$$

**F1-Score** : Cela traduit l'équilibre entre la précision et le rappel ; c'est la moyenne harmonique de Precision et Recall.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.6)$$

## 2.6 Conclusion

Dans ce chapitre, on a défini les outils matériels et logiciels suite au prétraitement, la collecte des données, ainsi qu'une description de l'ensemble de données et de son contenu.

# Chapitre 3

## Entrainement des modèles et résultats

### 3.1 Introduction

Dans ce chapitre, nous explorons les étapes de base pour former et évaluer un modèle de classification appliquée à un ensemble de données. L'objectif principal est d'identifier le modèle le plus performant en termes de précision et de robustesse, en tenant compte de sa complexité et de son efficacité de calcul.

L'évaluation du modèle est basée sur des mesures clés telles que la précision, le rappel, le score F1 et l'AUC, permettant une analyse détaillée des performances. Des outils tels que les matrices de confusion et les courbes ROC sont utilisés pour identifier les erreurs de classification et comparer la capacité des algorithmes à distinguer les classes. Ces analyses approfondies fournissent une vision claire des forces et des limites de chaque modèle dans différentes conditions.

### 3.2 Méthodologie

#### 3.2.1 Algorithmes de Classification

Les algorithmes suivants ont été testés :

- **SVM (Support Vector Machines)** : efficace pour la séparation des classes avec des marges maximales.
- **XGBoost** : un algorithme d'ensemble basé sur le boosting, connu pour sa performance

et sa vitesse.

- **Random Forest** : construit sur des arbres de décision pour réduire la variance et améliorer la robustesse.
- **KNN (K-Nearest Neighbors)** : un algorithme basé sur la proximité dans l'espace des caractéristiques.
- **Decision Tree** : un modèle simple mais interprétable pour la classification.
- **Naive Bayes** : un modèle simple mais interprétable pour la classification.
- **Logistic Regression** : efficace pour la classification binaire et multi-classes.

## 3.3 1. Évaluation des Modèles avec des Visualisations

### 3.3.1 Comparaison des Modèles

Les modèles XGBoost et SVM ont offert les meilleures performances globales, avec des précisions élevées et des AUC remarquables. Les modèles comme Naive Bayes et Decision Tree montrent des performances inférieures, en particulier dans des cas de données ambiguës ou bruitées.

TABLE 3.1 – Comparaison des modèles de classification

Modèle	Précision (%)	Complexité	Temps d'entraînement	Temps de prédiction
Logistic Regression	84.35	Faible	Faible	Faible
KNN	85.41	Faible	Moyen	Élevé
Decision Tree	78.88	Modérée	Faible	Faible
SVM	88.28	Modérée	Élevé	Modéré
Naive Bayes	58.56	Faible	Très faible	Très faible
Random Forest	87.64	Élevée	Élevé	Modéré
XGBoost	<b>88.35</b>	Élevée	Élevé	Faible

### 3.3.2 Matrice de Confusion

[1]Les matrices de confusion permettent d'analyser les erreurs de classification en identifiant les catégories fréquemment confondues. Par exemple, des erreurs significatives sont observées entre les catégories t-shirts, pulls et chemises, ce qui reflète leur similarité visuelle.

### 3.3.2.1 Logistic regression

La matrice de confusion pour le modèle de régression logistique montre des performances modérées, avec des confusions fréquentes entre certaines classes.

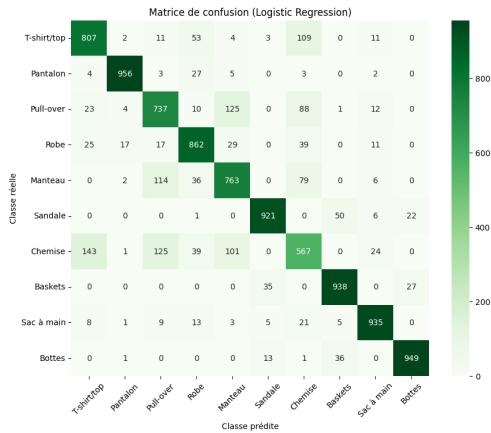


FIGURE 3.1 – Matrice de confusion du Logistic Regression

### 3.3.2.2 K-Nearest Neighbors

La matrice de confusion pour le modèle KNN illustre une précision élevée, mais révèle des confusions entre classes similaires.

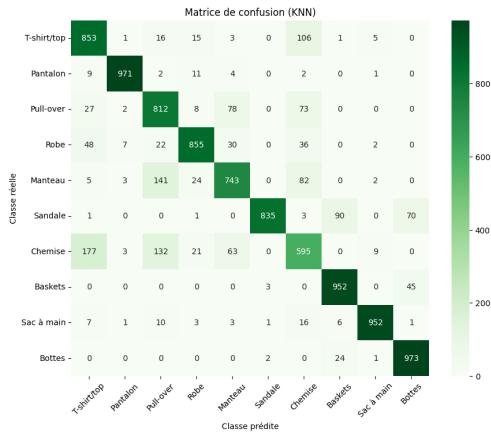


FIGURE 3.2 – Matrice de confusion du KNN

### 3.3.2.3 Decision Tree

Le modèle de l'arbre de décision présente une interprétabilité élevée, mais des performances inférieures dans des classes complexes.

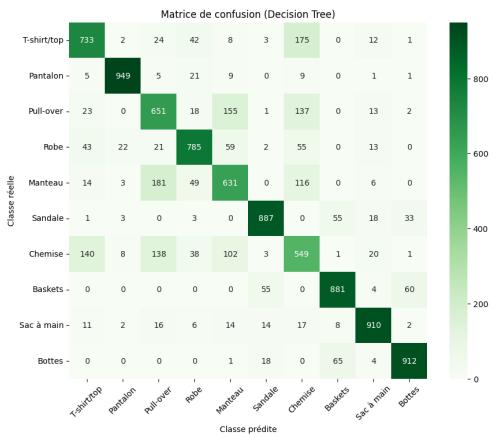


FIGURE 3.3 – Matrice de confusion du Decision Tree

### 3.3.2.4 SVM

Le SVM démontre une capacité supérieure à discriminer entre les classes, comme illustré dans la matrice de confusion.

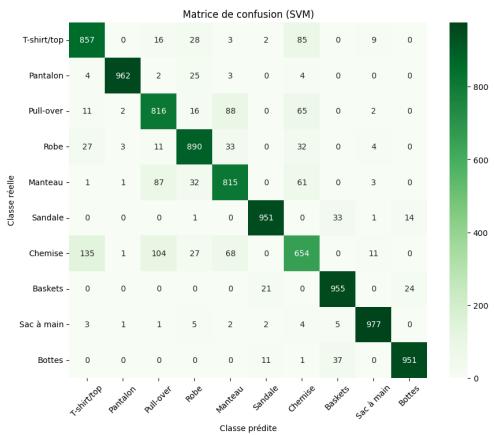


FIGURE 3.4 – Matrice de confusion du SVM

### 3.3.2.5 Naive Bayes

Le modèle Naive Bayes, bien qu'efficace pour certaines classes, montre des limites dans des scénarios complexes.

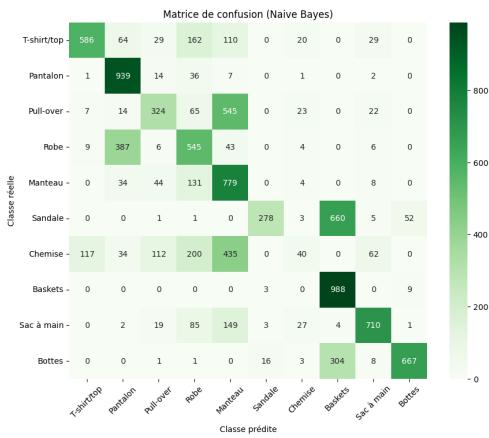


FIGURE 3.5 – Matrice de confusion du Naive Bayes

### 3.3.2.6 Random Forest

Le Random Forest combine plusieurs arbres de décision, améliorant la robustesse globale, comme indiqué dans la matrice de confusion.

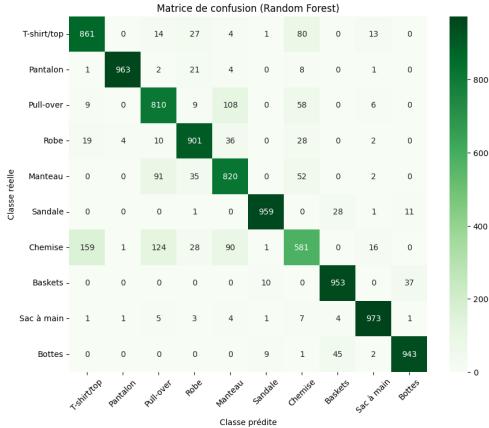


FIGURE 3.6 – Matrice de confusion du Random Forest

### 3.3.2.7 XGBoost

XGBoost offre les meilleures performances globales, avec une matrice de confusion montrant un minimum d'erreurs de classification.

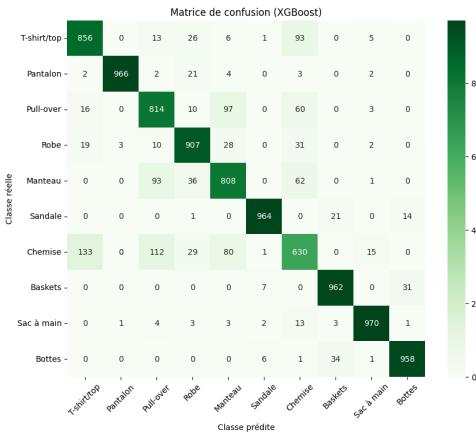


FIGURE 3.7 – Matrice de confusion du XGBoost

### 3.3.3 Rapport de Classification

Les métriques clés, telles que la précision, le rappel, le F1-score, et la précision globale, sont calculées pour chaque modèle. Ces indicateurs permettent de comparer les forces et faiblesses des algorithmes dans différentes conditions. dans ce cas, on va se focaliser sur les deux modèles SVM et XGBoost.

#### 3.3.3.1 SVM et XGBoost

TABLE 3.2 – Classification report SVM et XGBoost

Modele	SVM			XGBoost			
	Classes	Precision	Recall	F1-score	Precision	Recall	F1-score
T-shirt/top	0.83	0.86	0.84	0.83	0.86	0.85	0.85
Trouser	0.99	0.96	0.98	1.00	0.97	0.98	0.98
Pullover	0.79	0.82	0.80	0.78	0.81	0.79	0.79
Dress	0.87	0.89	0.88	0.88	0.91	0.89	0.89
Coat	0.81	0.81	0.81	0.79	0.81	0.80	0.80
Sandal	0.96	0.95	0.96	0.98	0.96	0.97	0.97
Shirt	0.72	0.65	0.69	0.71	0.63	0.67	0.67
Sneaker	0.93	0.95	0.94	0.94	0.96	0.95	0.95
Bag	0.97	0.98	0.97	0.97	0.97	0.97	0.97
Ankle boot	0.96	0.95	0.96	0.95	0.96	0.96	0.96
<b>Accuracy</b>	<b>0.88</b>						

### 3.3.4 Courbes ROC et AUC

Les courbes ROC montrent les compromis entre le taux de vrais positifs et le taux de faux positifs pour différents seuils de classification. L'AUC, ou surface sous la courbe, est utilisée comme indicateur global de performance. Les modèles XGBoost et SVM se distinguent par leurs AUC élevées, démontrant une capacité supérieure de discrimination entre les classes.

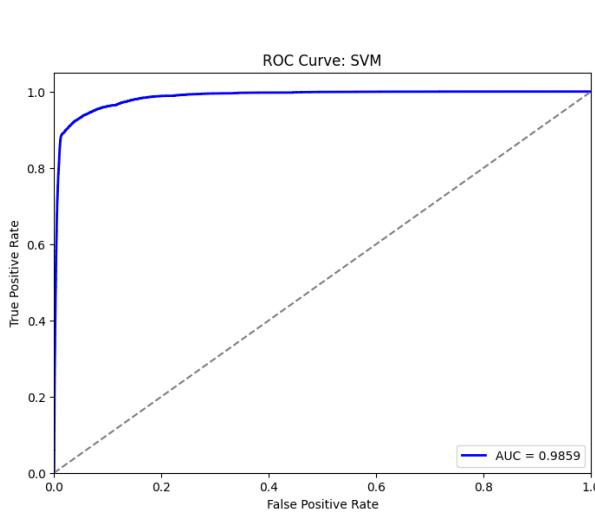


FIGURE 3.8 – Courbe ROC du SVM

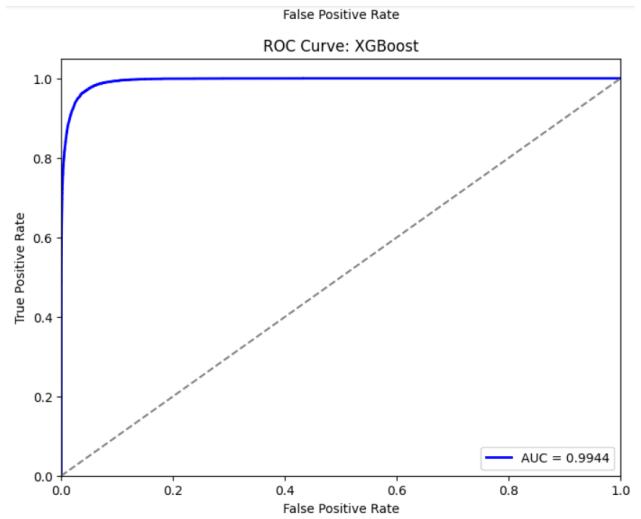


FIGURE 3.9 – Courbe ROC du XGBoost

### 3.3.5 GridSearch et RandomizedSearch

**GridSearchCV** explore systématiquement toutes les combinaisons possibles des hyperparamètres spécifiés, garantissant l'identification des meilleures valeurs, mais peut être coûteux en temps pour de grands espaces de recherche. En revanche, *RandomizedSearchCV* échantillonne de manière aléatoire un nombre défini de combinaisons, offrant une alternative plus rapide tout en conservant de bonnes performances dans l'optimisation des modèles.

#### 3.3.5.1 Optimisation des Hyperparamètres SVM avec GridSearchCV

La recherche des hyperparamètres optimaux d'un modèle SVM avec GridSearchCV améliore ses performances. Pour un noyau RBF, les valeurs  $C = [0.1, 1, 10, 100]$  et  $\gamma = [\text{'scale'}, \text{'auto'}, 0.01, 0.1, 1]$  ont été explorées. La validation croisée a identifié  $C = 10$  et  $\gamma = \text{'scale'}$  comme paramètres idéaux. Ce réglage a conduit à des performances optimales sur les données testées. Cette approche assure une classification fiable pour des problèmes complexes.

### 3.3.5.2 Optimisation des Hyperparamètres XGBoost avec RandomizedSearchCV

Lors de l'optimisation des hyperparamètres d'XGBoost avec *RandomizedSearchCV*, différentes combinaisons ont été testées. Cependant, les paramètres par défaut ont montré de meilleures performances que ceux suggérés par *RandomizedSearchCV*. Après comparaison, il a été décidé de conserver les paramètres par défaut, qui ont offert les meilleurs résultats pour la classification. Cette approche a permis d'optimiser la précision du modèle.

## 3.4 Division 80% train, 20% test et augmentation des données

### 3.4.1 Division 80% train 20% test

Rediviser le jeu de données Fashion MNIST en un ratio de 80% pour l'entraînement et 20% pour le test offre un meilleur équilibre entre les données d'entraînement et de test. Bien que la séparation initiale soit de 85,71% pour l'entraînement et 14,29% pour le test, une répartition 80/20 permet une évaluation plus représentative. Elle maximise également les données disponibles pour l'ajustement des hyperparamètres. Cette méthode est idéale pour intégrer un jeu de validation et mieux refléter les performances sur des données réelles.

```
# Normalize the data to be between 0 and 1
X_train = X_train.astype('float32') / 255.0
X_test = X_test.astype('float32') / 255.0
# Flatten the images (28x28 images become 784-dimensional vectors)
X_train = X_train.reshape(X_train.shape[0], 28*28)
X_test = X_test.reshape(X_test.shape[0], 28*28)
# Combine the training and testing data into one dataset
X_combined = np.concatenate((X_train, X_test), axis=0)
y_combined = np.concatenate((y_train, y_test), axis=0)
# Split the combined dataset into 80% training and 20% testing
x_train_final, x_test_final, y_train_final, y_test_final = train_test_split(
    X_combined, y_combined, test_size=0.2, random_state=42)
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train_final)
x_test_scaled = scaler.transform(x_test_final)
print(f"Final training data size (x_train_final): {x_train_scaled.shape}")
print(f"Final training labels size (y_train_final): {y_train_final.shape}")
print(f"Final test data size (x_test_final): {x_test_scaled.shape}")
print(f"Final test labels size (y_test_final): {y_test_final.shape}")

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515          0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880      1s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148            0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102      0s 0us/step
Final training data size (x_train_final): (56000, 784)
Final training labels size (y_train_final): (56000,)
Final test data size (x_test_final): (14000, 784)
Final test labels size (y_test_final): (14000,)
```

FIGURE 3.10 – Division 80% 20%

Après avoir divisé le dataset en 80% pour l'entraînement et 20% pour le test, les deux modèles ont montré des performances prometteuses, reflétant leur capacité à bien s'adapter au jeu de données initial. Les résultats obtenus sont les suivants :

- **SVM** : 90.94% de précision
- **XGBoost** : 89.14% de précision

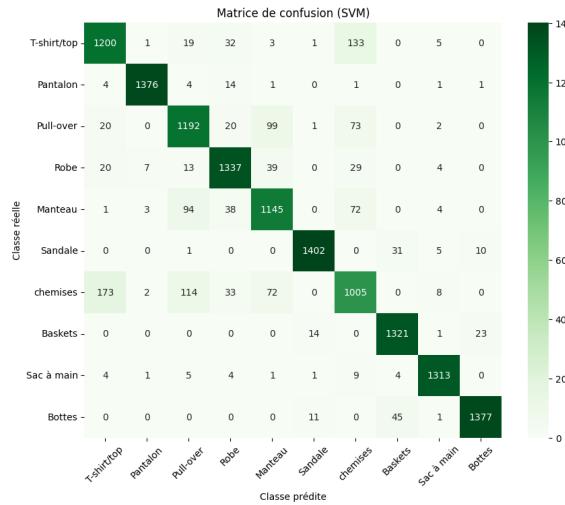


FIGURE 3.11 – Matrice confusion du svm

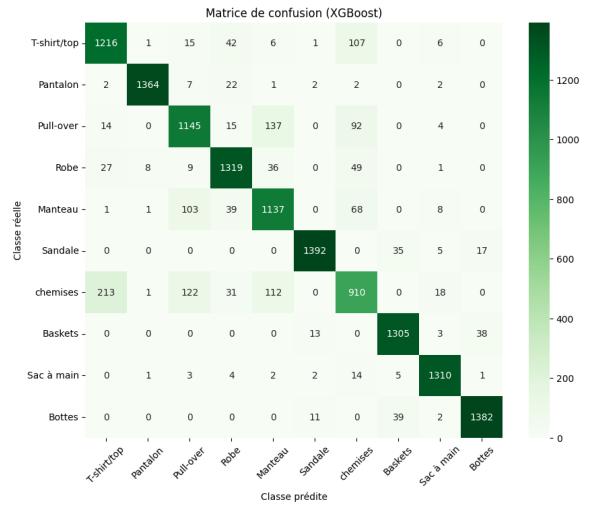


FIGURE 3.12 – Matrice confusion du xgboost

Ces résultats démontrent l'efficacité des deux algorithmes sur des données non augmentées, tout en mettant en lumière les différences subtiles entre les performances respectives des modèles.

### 3.4.2 Augmentation des données

L'augmentation des données a été appliquée dans le but d'améliorer la robustesse et la capacité de généralisation des modèles. Cette technique repose sur des transformations telles que des rotations, des translations, et des changements d'échelle, visant à enrichir artificiellement l'ensemble d'entraînement en augmentant la diversité des données disponibles.



FIGURE 3.13 – Illustration des données après augmentation

TABLE 3.3 – Classification Report SVM et XGBoost devision 80 20

Classe	SVM			XGBoost		
	Precision	Recall	F1-score	Precision	Recall	F1-score
T-shirt/top	0.84	0.86	0.85	0.83	0.87	0.85
Pants	0.99	0.98	0.99	0.99	0.97	0.98
Pullover	0.83	0.85	0.84	0.82	0.81	0.81
Dress	0.90	0.92	0.91	0.90	0.91	0.90
Coat	0.84	0.84	0.84	0.79	0.84	0.82
Sandal	0.98	0.97	0.97	0.98	0.96	0.97
Shirt	0.76	0.71	0.74	0.73	0.65	0.69
Sneaker	0.94	0.97	0.96	0.94	0.96	0.95
Bag	0.98	0.98	0.98	0.96	0.98	0.97
Ankle Boot	0.98	0.96	0.97	0.96	0.96	0.96
<b>Accuracy</b>	<b>0.90</b>	<b>0.90</b>	<b>0.90</b>	<b>0.89</b>	<b>0.89</b>	<b>0.89</b>

Bien que cette méthode soit souvent bénéfique pour réduire les problèmes de surapprentissage, son impact sur ce dataset a mené à une diminution notable de la précision des modèles, comme indiqué ci-dessous.

### 3.4.3 Entrainement sur Données Augmentées et Résultats

Après l'entraînement des modèles sur les données augmentées, une baisse de précision a été observée par rapport à l'approche initiale de division des données en 80% pour l'entraînement et 20% pour le test. Les résultats obtenus sont les suivants :

- **SVM** : 67.00% de précision
- **XGBoost** : 69.00% de précision

Cette diminution peut s'expliquer par le fait que l'augmentation des données, bien qu'elle enrichisse l'ensemble d'entraînement, peut introduire un certain bruit ou des transformations qui éloignent les exemples augmentés des caractéristiques originales du dataset. Cela rend la tâche de classification plus complexe pour les modèles testés, entraînant une légère dégradation des performances.

## 3.5 Suppression de la Classe Chemise

### 3.5.1 Motivation et Raisons de la Suppression

La classe "chemise" a été identifiée comme une source majeure de confusion pour les modèles de classification, en raison de sa similarité visuelle avec d'autres catégories telles que les t-shirts et les pulls. Afin de réduire les erreurs de classification et d'améliorer les performances globales des modèles, cette classe a été retirée du dataset. Cela permet également aux modèles de mieux se concentrer sur les différences entre les autres catégories, améliorant ainsi leur capacité à discriminer les classes restantes.



FIGURE 3.14 – Chemises



FIGURE 3.15 – Autres Chemises

### 3.5.2 Résultats après Suppression

Suite à la suppression de la classe "chemise", les modèles XGBoost et SVM ont été réentraînés sur le dataset modifié. Les performances obtenues sont les suivantes :

- **XGBoost** : 93.24% de précision
- **SVM** : 93.83% de précision

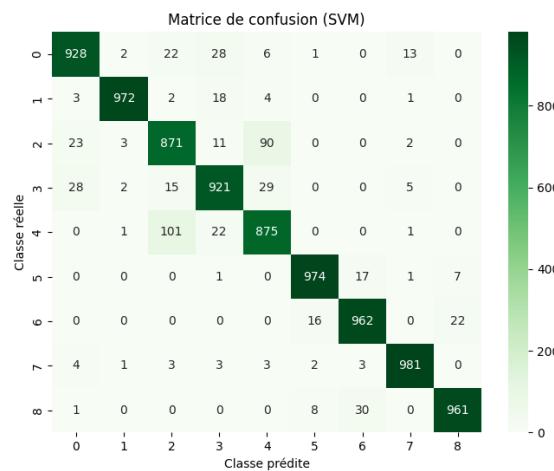


FIGURE 3.16 – Matrice confusion du svm

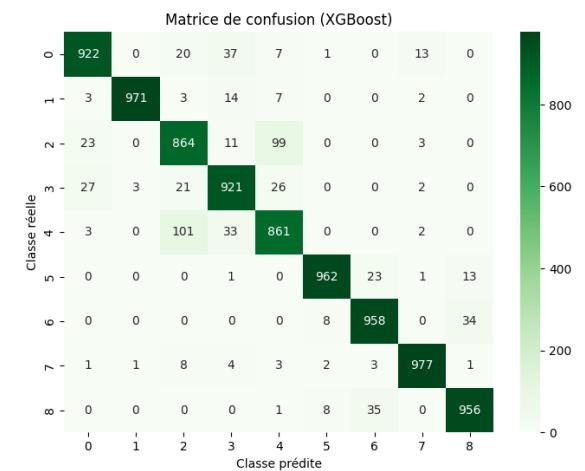


FIGURE 3.17 – Matrice confusion du xgboost

Ces résultats montrent une amélioration notable par rapport aux performances initiales. La suppression de la classe problématique a permis aux modèles de se concentrer davantage sur les caractéristiques distinctives des autres catégories, réduisant les erreurs de classification.

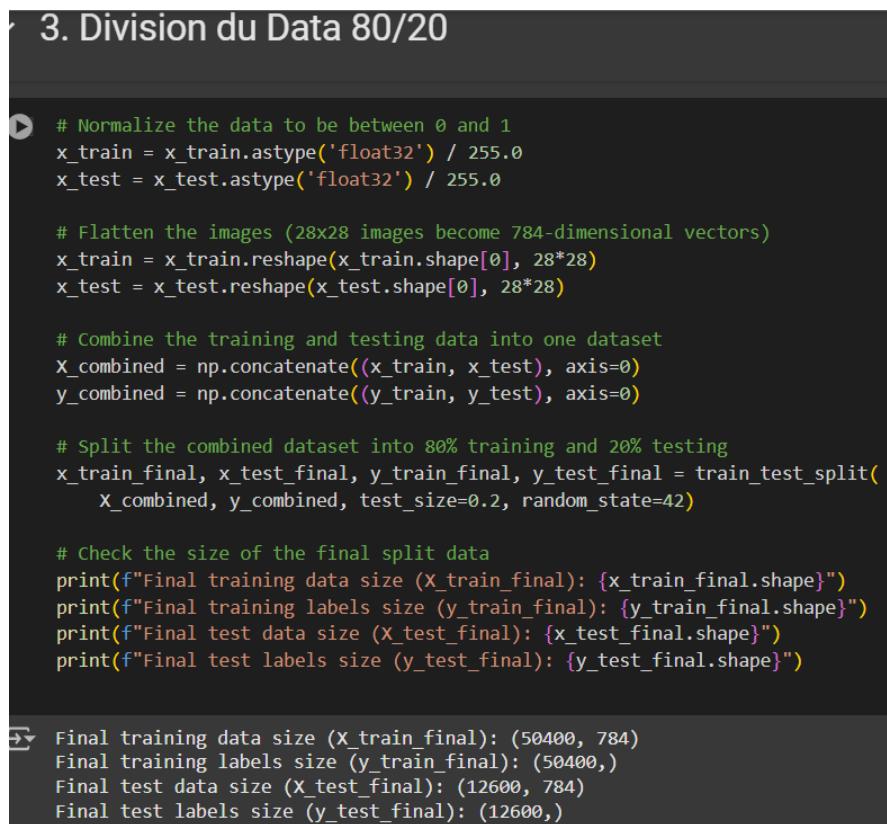
TABLE 3.4 – Classification Report SVM et XGBoost sans chemises

Classe	SVM			XGBoost		
	Precision	Recall	F1-score	Precision	Recall	F1-score
T-shirt/top	0.94	0.93	0.93	0.94	0.92	0.93
Pants	0.99	0.97	0.98	1.00	0.97	0.98
Pullover	0.86	0.87	0.86	0.85	0.86	0.86
Dress	0.92	0.92	0.92	0.90	0.92	0.91
Coat	0.87	0.88	0.87	0.86	0.86	0.86
Sandal	0.97	0.97	0.97	0.98	0.96	0.97
Shirt	0.95	0.96	0.96	0.94	0.96	0.95
Sneaker	0.98	0.98	0.98	0.98	0.98	0.98
Bag	0.97	0.96	0.97	0.95	0.96	0.95
Accuracy	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>0.93</b>	<b>0.93</b>	<b>0.93</b>

## 3.6 Suppression de la Classe Chemise avec Division 80% Train et 20% Test

### 3.6.1 Division du Dataset après Suppression

Après avoir retiré la classe "chemise", le dataset modifié a été divisé en 80% pour l'entraînement et 20% pour le test, conformément à la méthodologie standard. Cette division vise à évaluer la capacité des modèles à généraliser sur un ensemble de test tout en tirant parti de l'absence de la classe "chemise".



```
# Normalize the data to be between 0 and 1
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Flatten the images (28x28 images become 784-dimensional vectors)
x_train = x_train.reshape(x_train.shape[0], 28*28)
x_test = x_test.reshape(x_test.shape[0], 28*28)

# Combine the training and testing data into one dataset
X_combined = np.concatenate((x_train, x_test), axis=0)
y_combined = np.concatenate((y_train, y_test), axis=0)

# Split the combined dataset into 80% training and 20% testing
x_train_final, x_test_final, y_train_final, y_test_final = train_test_split(
    X_combined, y_combined, test_size=0.2, random_state=42)

# Check the size of the final split data
print(f"Final training data size (x_train_final): {x_train_final.shape}")
print(f"Final training labels size (y_train_final): {y_train_final.shape}")
print(f"Final test data size (x_test_final): {x_test_final.shape}")
print(f"Final test labels size (y_test_final): {y_test_final.shape}")

Final training data size (x_train_final): (50400, 784)
Final training labels size (y_train_final): (50400,)
Final test data size (x_test_final): (12600, 784)
Final test labels size (y_test_final): (12600,)
```

FIGURE 3.18 – Division data 80 20

### 3.6.2 Résultats après Entraînement

Les modèles XGBoost et SVM ont été réentraînés sur le dataset modifié avec la division 80%-20%. Les performances obtenues sont les suivantes :

- **XGBoost** : 94.86% de précision
- **SVM** : 94.49% de précision

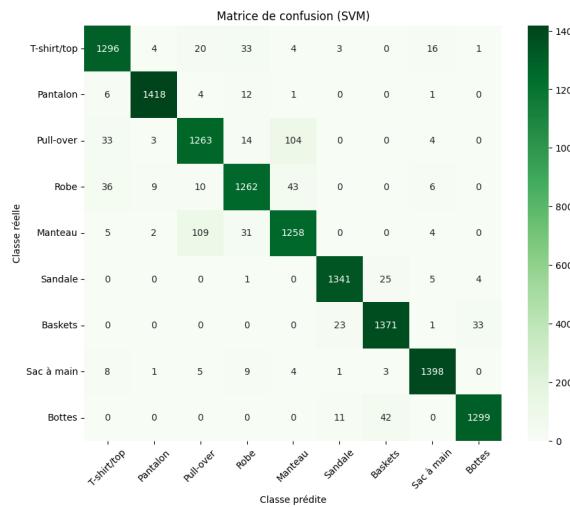


FIGURE 3.19 – Matrice confusion du svm

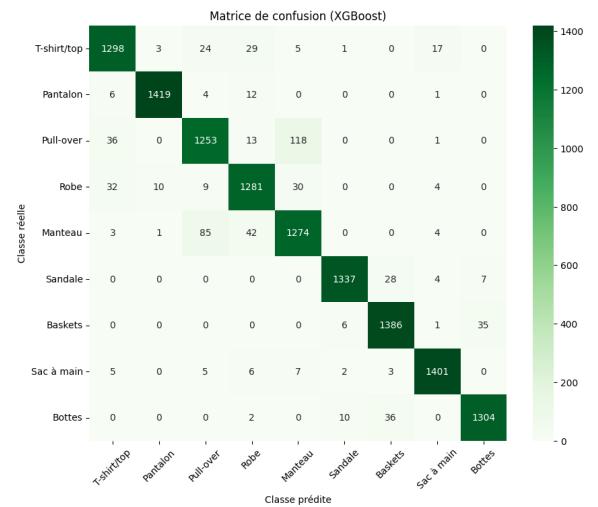


FIGURE 3.20 – Matrice confusion du xgboost

Ces résultats confirment que la combinaison de la suppression de la classe "chemise" et de la division 80%-20% améliore encore davantage la précision des modèles. La division a permis une meilleure répartition des données pour l'entraînement et le test, ce qui a conduit à une généralisation optimale des modèles.

TABLE 3.5 – Classification Report SVM, XGBoost

Classe	SVM			XGBoost		
	Precision	Recall	F1-score	Precision	Recall	F1-score
T-shirt/top	0.94	0.94	0.94	0.94	0.94	0.94
Pants	0.99	0.98	0.99	0.99	0.98	0.99
Pullover	0.90	0.89	0.89	0.91	0.88	0.89
Dress	0.93	0.92	0.93	0.92	0.94	0.93
Coat	0.89	0.89	0.89	0.89	0.90	0.90
Sandal	0.97	0.97	0.97	0.99	0.97	0.98
Shirt	0.95	0.96	0.96	0.95	0.97	0.96
Sneaker	0.97	0.98	0.98	0.98	0.98	0.98
Bag	0.97	0.96	0.97	0.97	0.96	0.97
<b>Accuracy</b>	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>0.95</b>	<b>0.95</b>	<b>0.95</b>

### 3.6.3 Analyse des Résultats

L'amélioration des performances après la suppression de la classe "chemise" et l'application d'une division 80%-20% met en évidence l'importance de la qualité et de la structure des

données. La réduction des sources de confusion a permis aux modèles de mieux capturer les relations entre les autres classes. Le SVM, en particulier, se distingue légèrement par sa précision supérieure à celle de XGBoost, bien que la différence soit marginale. Ces résultats soulignent également l'importance d'une analyse minutieuse des données pour optimiser les performances des algorithmes de classification.

### **3.6.4 Validation croisée**

La validation croisée est une méthode d'évaluation des modèles en apprentissage automatique. Elle consiste à diviser les données en K sous-ensembles, entraînant le modèle K fois, chaque fois sur K-1 sous-ensembles, et en validant sur le sous-ensemble restant. Cette technique permet d'estimer la performance du modèle sur des données non vues et de réduire le surapprentissage. Les résultats sont ensuite moyennés pour obtenir une évaluation plus fiable.

#### **3.6.4.1 validation croisée pour SVM**

Les scores obtenus lors de la validation croisée à 5 plis pour le modèle SVM sont très élevés, variant entre 0.9379 et 0.9428. Cela indique que le modèle présente une performance stable et fiable sur différents sous-ensembles des données. Le score moyen de 0.9399 suggère une bonne capacité de généralisation, avec peu de fluctuation dans les résultats, ce qui est un indicateur positif pour l'efficacité du modèle. Ces résultats montrent que le modèle SVM est performant et bien adapté aux données utilisées pour cette validation croisée.

#### **3.6.4.2 validation croisée pour XGBooost**

Les résultats de la validation croisée à 5 plis pour le modèle XGBoost montrent des performances impressionnantes, avec des scores variant entre 0.9386 et 0.9438. La précision moyenne de 0.9421 est excellente et témoigne de la robustesse du modèle. De plus, l'écart-type des scores étant très faible (0.00199), cela suggère que le modèle maintient une stabilité constante dans sa capacité à généraliser sur différents sous-ensembles des données. Ces résultats confirment que XGBoost est non seulement performant, mais aussi fiable, avec peu de variabilité, ce qui en fait un modèle particulièrement adapté pour des applications nécessitant une haute précision.

## 3.7 Conclusion

En conclusion, ce chapitre a permis de détailler les étapes essentielles de l'entraînement, de l'évaluation et de l'analyse des performances des modèles de classification. À travers une méthodologie rigoureuse, incluant la préparation des données, l'application d'algorithmes variés, l'usage de techniques de visualisation, et l'usage de cross validation, nous avons mis en évidence les modèles les plus adaptés à notre jeu de données. L'évaluation basée sur des métriques quantitatives et des outils visuels a permis d'identifier les forces et faiblesses de chaque approche, offrant une base solide pour orienter les choix futurs. Les résultats obtenus soulignent l'importance d'une évaluation approfondie et d'une analyse critique pour optimiser la précision et la robustesse des systèmes de classification.

# Conclusion générale et perspectives

**Conclusion** Dans ce projet, nous avons exploré la classification des images de vêtements à l'aide du dataset Fashion\_MNIST. Nous avons comparé deux modèles d'apprentissage automatique, SVM et XGBoost, en évaluant leur performance sur différentes configurations, telles que la division du dataset en proportions spécifiques, l'augmentation des données, et la suppression de certaines classes.

Bien que ce sujet soit idéalement adapté pour être traité avec des techniques de deep learning, notamment en raison de la nature pixellisée et peu claire des images du dataset ainsi que de son manque de propreté, nous avons choisi d'utiliser des modèles de machine learning. Malgré cela, nos efforts ont permis d'obtenir des résultats significatifs, surpassant même certains travaux antérieurs qui ont utilisé le même dataset pour des approches similaires. Cette amélioration démontre la robustesse et l'efficacité des modèles que nous avons développés, ainsi que l'importance des étapes de prétraitement et d'optimisation dans les projets de classification d'images.

Ces travaux mettent en avant les opportunités et les défis liés à l'utilisation du machine learning pour des données complexes, tout en ouvrant la voie à des applications plus avancées et innovantes.

**Perspectives** Dans la continuité de ce travail, plusieurs extensions et améliorations peuvent être envisagées :

Réalité augmentée (AR) : L'intégration de la réalité augmentée permettrait de visualiser les vêtements directement sur l'utilisateur à travers une interface immersive. Cette fonctionnalité offrirait une expérience interactive et intuitive, permettant aux utilisateurs d'essayer virtuellement les vêtements sans les porter physiquement.

lement des tenues et de mieux visualiser leur apparence avant un achat.

Proposition d'outfits personnalisés : Une autre évolution majeure serait la création d'un système intelligent capable de recommander des tenues (outfits) en fonction du profil utilisateur, de ses préférences, et des conditions météorologiques. Par exemple :

Par temps pluvieux, le système pourrait suggérer une tenue incluant un imperméable, des bottes et un parapluie. Par temps chaud, il pourrait proposer des vêtements légers et respirants adaptés à la saison.

Ces fonctionnalités nécessiteront l'intégration de modèles de recommandation avancés, alimentés par des données sur les préférences des utilisateurs et les prévisions météorologiques. Cette approche combinera l'intelligence artificielle, le machine learning et des données contextuelles pour enrichir l'expérience utilisateur et rendre le système plus utile et innovant.

En conclusion, ces perspectives positionnent ce projet comme une solution polyvalente et moderne, combinant classification d'images, recommandation personnalisée et immersion via la réalité augmentée. Ces avancées pourraient être particulièrement attractives pour l'industrie de la mode et de la vente en ligne.

# Bibliographie

- @bookbishop2006, title = Pattern Recognition and Machine Learning, author = Christopher M. Bishop, year = 2006, publisher = Springer, address = New York, isbn = 978-0387310732
- @articlebreiman2001, title = Random Forests, author = Leo Breiman, journal = Machine Learning, volume = 45, number = 1, pages = 5–32, year = 2001, doi = 10.1023/A:1010933404324
- @articlechen2016, title = XGBoost : A Scalable Tree Boosting System, author = Tianqi Chen and Carlos Guestrin, booktitle = Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages = 785–794, year = 2016, doi = 10.1145/2939672.2939785
- @articlecortes1995, title = Support-Vector Networks, author = Corinna Cortes and Vladimir Vapnik, journal = Machine Learning, volume = 20, number = 3, pages = 273–297, year = 1995, doi = 10.1007/BF00994018
- @articlecover1967, title = Nearest Neighbor Pattern Classification, author = Thomas M. Cover and Peter E. Hart, journal = IEEE Transactions on Information Theory, volume = 13, number = 1, pages = 21–27, year = 1967, doi = 10.1109/TIT.1967.1053964
- @articlefawcett2006, title = An Introduction to ROC Analysis, author = Tom Fawcett, journal = Pattern Recognition Letters, volume = 27, number = 8, pages = 861–874, year = 2006, doi = 10.1016/j.patrec.2005.10.010
- @bookgeron2019, title = Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems, author = Aurélien Géron, year = 2019, publisher = O'Reilly Media, edition = 2nd, isbn = 978-1492032649
- @bookhosmer2013, title = Applied Logistic Regression, author = David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant, year = 2013, publisher = Wiley, edition = 3rd, isbn = 978-0470582473
- @manualscikit-learn, title = Scikit-Learn : Machine Learning in Python, author = Scikit-Learn

- Developers, year = 2023, url = <https://scikit-learn.org/stable/>, note = Accessed : 2025-01-05
- @articlequinlan1986, title = Induction of Decision Trees, author = J. Ross Quinlan, journal = Machine Learning, volume = 1, number = 1, pages = 81–106, year = 1986, doi = 10.1023/A:1022643204877
- @articlerish2001, title = An Empirical Study of the Naive Bayes Classifier, author = Irina Rish, booktitle = IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, pages = 41–46, year = 2001