

Machine Learning et Deep Learning en Data Science

by REDA OUZIDANE

3 avril 2025

1 Introduction

Le Machine Learning et le Deep Learning sont des sous-domaines de l'intelligence artificielle qui permettent d'entraîner des modèles sur des données pour faire des prédictions ou extraire des informations pertinentes.

2 Machine Learning

2.1 Types d'apprentissage

- **Supervisé** : Apprentissage basé sur des données étiquetées. Exemples :
 - **Classification** : Prédire une catégorie (ex : spam ou non-spam).
 - **Régression** : Prédire une valeur continue (ex : prix d'une maison).
- **Non supervisé** : Détecter des motifs dans les données sans étiquettes. Exemples :
 - **Clustering** : Regrouper des données similaires (ex : segmentation client).
 - **Réduction de dimensionnalité** : PCA, t-SNE pour visualiser des données complexes.
- **Apprentissage par renforcement** : Agent prenant des décisions pour maximiser une récompense cumulative (ex : AlphaGo, robots autonomes).

2.2 Prétraitement des Données

- **Traitement des valeurs manquantes** : Suppression, imputation (moyenne, médiane, KNN Imputer).

- **Standardisation** : Transformation des données pour une distribution normale :

$$Z = \frac{X - \mu}{\sigma} \quad (1)$$

- **Normalisation** : Mise à l'échelle des données dans un intervalle défini :

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

- **Encodage des variables catégoriques** :
 - **One-Hot Encoding** : Création de colonnes binaires pour chaque catégorie.
 - **Label Encoding** : Attribution d'un entier unique à chaque catégorie.

2.3 Implémentation en Python

Listing 1 – Chargement des bibliothèques

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Chargement des données
X = pd.read_csv("data.csv")
y = X.pop("target")

# Pr traitement
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, te

# Mod le
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# valuation
print("Accuracy:", accuracy_score(y_test, y_pred))
```

3 Deep Learning

3.1 Introduction

Le Deep Learning repose sur des réseaux de neurones profonds (DNN) pour extraire des représentations complexes à partir de données non structurées.

3.2 Types de Réseaux

- **Perceptron Multicouche (MLP)** : Réseau de neurones dense utilisé pour des tâches de classification et régression.
- **Convolutional Neural Networks (CNN)** : Utilisé pour l'analyse d'images et la vision par ordinateur.
- **Recurrent Neural Networks (RNN, LSTM)** : Modélisation des séquences temporelles et NLP.
- **Transformers** : Architecture avancée pour le NLP (ex : BERT, GPT).

3.3 Implémentation en Python avec TensorFlow

Listing 2 – Réseau de neurones simple avec TensorFlow

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

# Construction du modèle
model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.5),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Compilation
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Entraînement
model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_val, y_val))
```

4 Évaluation et Optimisation

- **Métriques de Performance :**
 - **Accuracy :** Pourcentage de bonnes prédictions.
 - **Précision et Recall :** Utilisés pour des classes déséquilibrées.
 - **Matrice de confusion :** Visualisation des erreurs.
- **Optimisation des Hyperparamètres :** GridSearchCV, Random Search, Bayesian Optimization.
- **Techniques de régularisation :** Dropout, L1/L2 Regularization.

5 Conclusion

Le Machine Learning et le Deep Learning offrent des outils puissants pour l'analyse des données et la prédiction. L'utilisation de bibliothèques comme Scikit-Learn et TensorFlow permet une mise en œuvre efficace et rapide. **By REDA OUZIDANE.**