Modèle de Prédiction du Prix de Clôture des Actions

Projet Finance – Préparation du Modèle de Machine Learning

Hamza Berrada

Novembre 2025

1 Objectif du Modèle

L'objectif est de construire un modèle de machine learning permettant de prédire la valeur **Close(t)** d'une action à partir des données historiques disponibles jusqu'à la veille. Le modèle choisi pour débuter est une **régression linéaire** (modèle simple et interprétable).

Le travail s'appuie sur quatre grandes entreprises du secteur de l'intelligence artificielle :

- Nvidia (NVDA)
- Microsoft (MSFT)
- Alphabet / Google (GOOG)
- AMD (AMD)

Les données sont récupérées via Yahoo Finance et stockées au format CSV.

2 Variables et Structure des Données

Chaque fichier CSV contient les colonnes suivantes :

- **Date**
- **Open** (prix d'ouverture)
- **High** (plus haut du jour)
- **Low** (plus bas du jour)
- **Close** (prix de clôture)
- **Volume** (volume échangé)

La variable cible du modèle est :

$$y = \text{Close}(t)$$

Les variables explicatives sont les valeurs des jours précédents :

$$X = \{\text{Open}, \text{High}, \text{Low}, \text{Close}, \text{Volume}\}$$

3 Problème Temporel et Fenêtre Glissante

Pour éviter toute fuite d'information (data leakage), le modèle ne doit utiliser que les données du passé. Nous introduisons une **fenêtre temporelle** de taille w (window size).

Pour prédire le jour t, on utilise les w jours précédents :

$$t-1,\ t-2,\ \ldots,\ t-w$$

Chaque jour fournit 5 variables, donc après **flatten**, le vecteur de features contient :

$$5 \times w \text{ colonnes}$$

Les premières lignes du dataset sont ignorées afin de garantir la disponibilité complète des w jours précédents.

4 Création Automatique des Features par Décalage

Pour chaque fenêtre temporelle, les variables sont générées automatiquement à l'aide de la fonction `shift()` de Pandas.

Exemple pour $w = 3$:

```
Open_t1   = df["Open"].shift(1)
Open_t2   = df["Open"].shift(2)
Open_t3   = df["Open"].shift(3)
```

Cette opération est répétée pour les colonnes High, Low, Close et Volume.

Toutes les colonnes contenant des valeurs NaN (début du dataframe) sont supprimées.

5 Construction du Dataset

Le dataset final est construit selon les étapes suivantes :

1. Charger les données brutes depuis le CSV.
2. Générer les colonnes décalées pour chaque fenêtre w.
3. Appliquer un flatten naturel via la génération de colonnes `shift()`.
4. Supprimer les lignes contenant des valeurs NaN.
5. Définir :

$$X = \{\text{toutes les colonnes décalées}\} \qquad y = \text{Close}(t)$$

6 Fenêtres Testées

Dans une démarche empirique, différentes tailles de fenêtres seront testées :

$$w \in \{1,\ 3,\ 5,\ 7,\ 14,\ 30\}$$

Pour chaque fenêtre :

- construction automatique du dataset
- entraînement d'un modèle de régression linéaire
- mesure de la performance sur un jeu de test
- traçage de la courbe "elbow" pour trouver le meilleur compromis entre performance et complexité

7 Prochaine Étape

La prochaine étape du projet consiste à :

- écrire la fonction Python `prepare_window_data(df, window_size)`
- générer les datasets pour chaque fenêtre
- entraîner la régression linéaire pour chaque fenêtre
- comparer les résultats

Ce document constitue le socle théorique pour implémenter proprement le modèle.