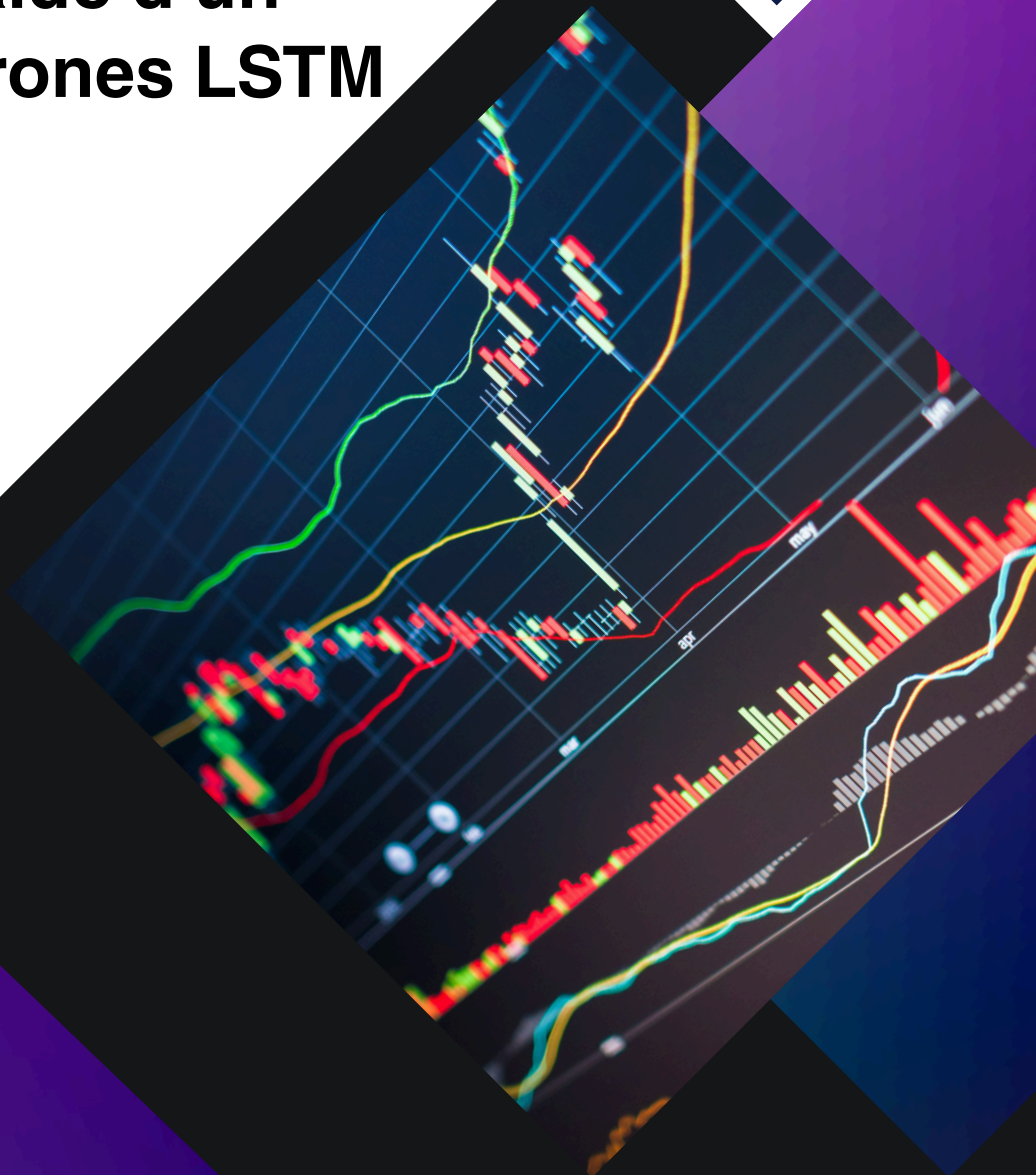


RAPPORT DE PROJET

**Prédiction des prix des
actions à l'aide d'un
réseau de neurones LSTM**



**FAIT PAR :
BOUARFA LAHMAR**

**Supervisé par : M. K.
ELMOUTAOUKIL**

Table des matières

01

Introduction

05

Entraînement du modèle

02

Présentation des données

06

Évaluation et résultats

03

Prétraitement des données

07

Conclusion

04

Architecture du modèle LSTM



Introduction

La prévision des prix boursiers est un enjeu majeur dans le domaine de la finance, tant pour les investisseurs que pour les analystes. Ces prix sont influencés par une multitude de facteurs économiques, politiques et sociaux, rendant leur comportement souvent difficile à modéliser. Ces dernières années, les avancées en intelligence artificielle, et plus particulièrement en apprentissage profond (deep learning), ont ouvert de nouvelles perspectives pour traiter ces données complexes et temporelles.

Dans le cadre de ce projet, nous avons étudié la possibilité de prédire les prix de clôture d'actions à l'aide d'un modèle basé sur les réseaux de neurones LSTM (Long Short-Term Memory), une variante performante des réseaux de neurones récurrents (RNN) adaptée à l'analyse de séries temporelles.

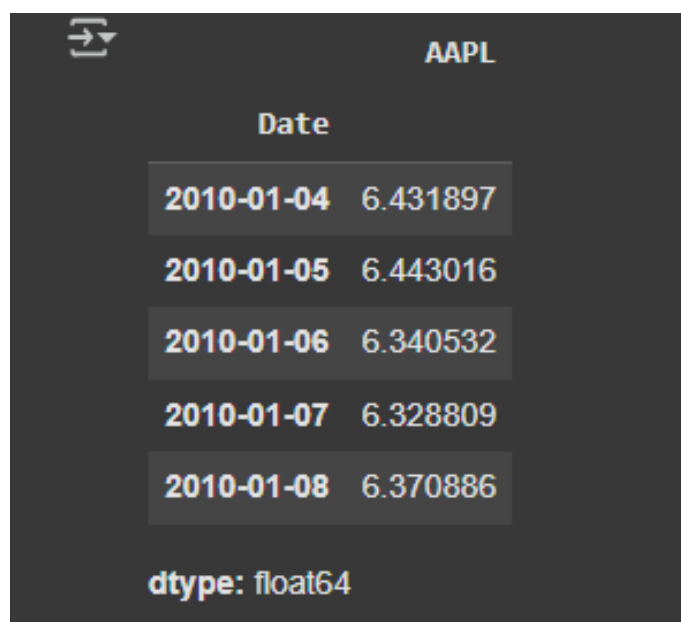
L'analyse a porté sur les données historiques de 19 entreprises technologiques cotées en bourse, parmi lesquelles Amazon (AMZN), Meta (META), Netflix (NFLX), Apple (AAPL), et d'autres géants du secteur. Ces données, extraites de la plateforme Yahoo Finance, ont été exploitées pour entraîner et évaluer des modèles de prédiction.

Ce rapport présente l'ensemble du processus réalisé : de la collecte et préparation des données, à la mise en place du modèle LSTM, jusqu'à l'évaluation des performances et l'interprétation des résultats obtenus. Le but est d'explorer la capacité de ce type de réseau à apprendre les dynamiques boursières et à prédire efficacement les évolutions futures des prix.

Présentation des données

Les données utilisées dans ce projet ont été récupérées via la bibliothèque Python `yfinance`, qui permet un accès rapide et automatisé aux données historiques du marché boursier provenant de Yahoo Finance. Cet outil open source facilite l'extraction des informations essentielles comme les prix d'ouverture, de clôture, les volumes échangés, les plus hauts et plus bas journaliers. Pour ce projet, les données de 19 entreprises technologiques, telles que Amazon, Meta, Netflix, Apple, Google ou encore Microsoft, ont été collectées sur une période allant du 1er janvier 2010 au 1er janvier 2024.

Parmi toutes les colonnes disponibles, seule la valeur de clôture quotidienne ("Close") a été utilisée comme variable cible pour la prédiction. Après téléchargement, les données ont été nettoyées en supprimant les valeurs manquantes ou incohérentes afin d'assurer la qualité des entrées dans le modèle. Le choix du prix de clôture s'explique par sa pertinence pour les analyses techniques, car il reflète la valeur finale d'échange d'une action à la fin d'une séance de marché. Ces séries temporelles ont ensuite été normalisées afin de faciliter l'apprentissage du modèle LSTM.



AAPL	
Date	
2010-01-04	6.431897
2010-01-05	6.443016
2010-01-06	6.340532
2010-01-07	6.328809
2010-01-08	6.370886

`dtype: float64`

Prétraitement des données

Avant d'entraîner le modèle LSTM, un prétraitement rigoureux des données a été effectué. Tout d'abord, les prix de clôture ont été normalisés à l'aide du `MinMaxScaler` de la bibliothèque `scikit-learn`, afin de les transformer dans un intervalle compris entre 0 et 1. Cette étape est cruciale pour améliorer la stabilité de l'apprentissage et éviter que certaines valeurs trop élevées ne dominent les autres. Ensuite, les données ont été organisées en séquences temporelles à l'aide d'une technique de fenêtres glissantes : pour chaque jour à prédire, le modèle prend en entrée les 60 jours précédents, ce qui permet de capturer les dépendances temporelles dans les séries de prix. Enfin, les données ont été divisées en deux ensembles : 80 % pour l'apprentissage du modèle (training set) et 20 % pour l'évaluation (test set), en respectant l'ordre chronologique des données afin de conserver la cohérence temporelle.

```
# Normalisation
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)

# Séquences (fenêtres glissantes)
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:i+seq_length])
        y.append(data[i+seq_length])
    return np.array(X), np.array(y)

SEQ_LEN = 60
X, y = create_sequences(data_scaled, SEQ_LEN)

# Split en train/test
split = int(0.8 * len(X))
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]

# Reshape pour LSTM
X_train = X_train.reshape((X_train.shape[0], SEQ_LEN, 1))
X_test = X_test.reshape((X_test.shape[0], SEQ_LEN, 1))
```


Entraînement du modèle

L'entraînement du modèle s'est appuyé sur une architecture LSTM (Long Short-Term Memory) construite avec la bibliothèque Keras de TensorFlow. Le modèle comprend deux couches LSTM empilées, entrecoupées de couches Dropout pour réduire le risque de surapprentissage, et une couche de sortie dense à un neurone pour prédire la valeur future. L'entraînement a été réalisé sur 50 époques avec un batch size de 32, en utilisant l'optimiseur Adam et la fonction de perte Mean Squared Error (MSE). Un mécanisme d'early stopping a été mis en place pour interrompre l'apprentissage si la performance de validation cessait de s'améliorer pendant 10 époques consécutives. Les courbes de validation ont été analysées pour détecter d'éventuels signes de surapprentissage ; le modèle a montré une bonne capacité de généralisation, avec une convergence stable de la perte d'entraînement et de validation. L'entraînement a été effectué sur un ordinateur personnel avec un processeur standard (CPU), et a nécessité un temps de calcul raisonnable d'environ quelques minutes par entreprise.

```
# Modèle LSTM
model = Sequential([
    LSTM(100, return_sequences=True, input_shape=(SEQ_LEN, 1)),
    Dropout(0.2),
    LSTM(50, return_sequences=False),
    Dropout(0.2),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')

# Early stopping
early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

# Entraînement
model.fit(X_train, y_train, epochs=50, batch_size=32,
        validation_split=0.1, callbacks=[early_stop], verbose=1)

# Prédiction
pred_train = model.predict(X_train)
pred_test = model.predict(X_test)

# Dénormalisation
y_train_inv = scaler.inverse_transform(y_train.reshape(-1, 1))
train_pred_inv = scaler.inverse_transform(pred_train)
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))
test_pred_inv = scaler.inverse_transform(pred_test)

# Erreur MSE
mse_train = mean_squared_error(y_train_inv, train_pred_inv)
mse_test = mean_squared_error(y_test_inv, test_pred_inv)
```

Évaluation et résultats

Après l'entraînement du modèle, une phase d'évaluation a été réalisée pour mesurer ses performances sur les données d'entraînement et de test. Les prédictions sur les données d'entraînement montrent une bonne correspondance avec les valeurs réelles, indiquant que le modèle a bien appris les tendances des séries temporelles sans les mémoriser de manière excessive. Sur les données de test, le modèle a également produit des prédictions relativement proches des valeurs réelles, démontrant sa capacité de généralisation. Pour évaluer quantitativement la qualité des prévisions, l'erreur quadratique moyenne (MSE) a été calculée pour les deux ensembles. Ces erreurs fournissent une mesure de la précision globale du modèle : des valeurs faibles indiquent que les prédictions sont proches des valeurs réelles. Enfin, des graphiques comparatifs ont été générés pour visualiser les résultats : les courbes représentant les prix réels et prédits permettent d'analyser visuellement les écarts, de vérifier la justesse des prévisions et d'identifier d'éventuels décalages temporels ou erreurs sur certaines périodes.



Conclusion

Ce projet a permis d'explorer l'utilisation des réseaux de neurones LSTM pour la prédiction des prix boursiers, en se concentrant sur les données historiques de 19 grandes entreprises technologiques. Grâce à la capacité des LSTM à modéliser les dépendances temporelles dans les séries de données, le modèle a réussi à anticiper les évolutions futures des prix de clôture avec une précision satisfaisante, tant sur les données d'entraînement que sur les données de test.

Le travail a également mis en évidence l'importance d'un prétraitement adapté, notamment la normalisation des données et la création de séquences temporelles, ainsi que l'usage d'outils tels que l'early stopping pour éviter le surapprentissage. Malgré les bons résultats obtenus, les prévisions restent sensibles à la nature volatile et complexe des marchés financiers, et pourraient être améliorées en intégrant des variables supplémentaires ou des modèles plus avancés.

En somme, ce projet illustre le potentiel des techniques d'apprentissage profond appliquées à la finance, tout en soulignant les défis inhérents à la modélisation de données financières complexes.