Projet informatique
Exploitation massive des données

Participants :

Antro Comlan-koumi
Sofiane Hamza
Mouhamadou Moustapha Niang

Introduction

Dans ce projet, nous avons utilisé les données historiques de clôture des prix de l'action GOOGLE pour étudier la performance de cette action dans le passé. Nous avons ensuite utilisé ces données pour entraîner un modèle de régression linéaire, qui nous a permis de faire des prévisions sur les rendements futurs de cette action. Enfin, nous avons utilisé ces prévisions pour estimer les prix futurs de l'action, nous avons par exemple estimé le prix de l'action dans les 10 prochains jours.

Récupération de données

Nous avons utilisé la bibliothèque Python "yfinance" pour récupérer les données historiques de clôture des prix de l'action. Pour ce faire, nous avons d'abord importé la bibliothèque, puis nous avons utilisé l'objet "*Ticker*". L'objet "Ticker" peut être créé en passant le symbole de l'action que nous souhaitons qui est "GOOGL" dans notre cas, cet objet récupérera les données, puis en appelant sa méthode "history". Cette méthode télécharge les données historiques et les renvoie dans un objet Pandas DataFrame pour une utilisation ultérieure dans notre analyse et prédiction.

Nous avons utilisé le paramètre période égale à max pour collecter le maximum de données que la bibliothèque peut nous renvoyer.

Vous trouverez dans le fichier excel joint les données utilisées, vous pouvez également les consulter en passant par URL suivant : https://finance.yahoo.com/quote/GOOGL/history?period1=1092960000&period2=1675814400&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true

```
import yfinance as yf
import numpy as np
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression

googl = yf.Ticker("GOOGL")
historical_data = googl.history(period="max")
print("les données historiques du marché de l'action GOOGLE depuis 19/08/2004\n")
print(historical_data)
```

les données historiques du marché de l'action GOOGLE depuis 19/08/2004

Date	Open	High	...	Dividends	Stock Splits
2004-08-19 00:00:00-04:00	2.502503	2.604104	...	0.0	0.0
2004-08-20 00:00:00-04:00	2.527778	2.729730	...	0.0	0.0
2004-08-23 00:00:00-04:00	2.771522	2.839840	...	0.0	0.0
2004-08-24 00:00:00-04:00	2.783784	2.792793	...	0.0	0.0
2004-08-25 00:00:00-04:00	2.626627	2.702703	...	0.0	0.0
...	...	...	...	...	...
2023-02-02 00:00:00-05:00	105.800003	107.849998	...	0.0	0.0
2023-02-03 00:00:00-05:00	102.930000	107.809998	...	0.0	0.0
2023-02-06 00:00:00-05:00	102.400002	104.360001	...	0.0	0.0
2023-02-07 00:00:00-05:00	103.220001	108.180000	...	0.0	0.0
2023-02-08 00:00:00-05:00	102.050003	103.139999	...	0.0	0.0

[4651 rows x 7 columns]

Figure 1 : Capture d'écran du tableau des données historiques du marché de l'action GOOGLE

Lorsque on demande les données historiques de clôture des prix d'une action GO#OGLE avec yfinance, les colonnes renvoyées incluent :

- Date: La date de la valeur de clôture.
- Open: Le prix d'ouverture de l'action pour la date spécifiée.
- High: Le prix le plus élevé atteint par l'action pour la date spécifiée.
- Low: Le prix le plus bas atteint par l'action pour la date spécifiée.
- Close: Le prix de clôture de l'action pour la date spécifiée.
- Adj Close: Le prix de clôture ajusté de l'action pour la date spécifiée, tenue compte de tout dividende ou split d'actions.
- Volume: Le nombre d'actions échangées pour la date spécifiée.

Traitement de données

Pour que notre modèle d'apprentissage automatique fonctionne correctement, il faut éviter qu'il y ait des valeurs manquant "NaN" parmi nos données. Pour cela, la phase de traitement de données est importante avant d'entraîner le modèle.

Pour traiter les valeurs manquantes de vos données, nous avons le choix soit de : Supprimez les lignes contenant des valeurs manquantes, remplir les valeurs manquantes avec une valeur

constante (0, la valeur moyenne de la colonne ..etc), ou de remplir les valeurs manquantes avec la valeur précédente ou suivante.

Nous avons choisi de supprimer les lignes contenant des valeurs manquantes avec la fonction "dropna" de Python.

```
#Suppression des lignes avec des valeurs manquantes
historical_data.dropna(inplace=True)

print("Close : Le prix de clôture de l'action pour la date spécifiée")
print(historical_data[['Close']])
```

```
Close : Le prix de clôture de l'action pour la date spécifiée
                                Close
Date
2004-08-19 00:00:00-04:00    2.511011
2004-08-20 00:00:00-04:00    2.710460
2004-08-23 00:00:00-04:00    2.737738
2004-08-24 00:00:00-04:00    2.624374
2004-08-25 00:00:00-04:00    2.652653
...                               ...
2023-02-02 00:00:00-05:00  107.739998
2023-02-03 00:00:00-05:00  104.779999
2023-02-06 00:00:00-05:00  102.900002
2023-02-07 00:00:00-05:00  107.639999
2023-02-08 00:00:00-05:00   99.370003

[4651 rows x 1 columns]
```

Figure 2 : La colonne Close, Le prix de clôture de l'action pour la date spécifiée

```
# Calculer les rendements du prix de l'action avec la fonction pct_change()
historical_data['returns'] = historical_data['Close'].pct_change()
print("Returns : le rendement du prix de l'action calculé en fonction de 'Close'")
print(historical_data[['returns']] )
```

```
Returns : le rendement du prix de l'action calculé en fonction de 'Close'
                           returns
Date
2004-08-19 00:00:00-04:00       NaN
2004-08-20 00:00:00-04:00  0.079430
2004-08-23 00:00:00-04:00  0.010064
2004-08-24 00:00:00-04:00 -0.041408
2004-08-25 00:00:00-04:00  0.010776
...                             ...
2023-02-02 00:00:00-05:00  0.072787
2023-02-03 00:00:00-05:00 -0.027474
2023-02-06 00:00:00-05:00 -0.017942
2023-02-07 00:00:00-05:00  0.046064
2023-02-08 00:00:00-05:00 -0.076830

[4651 rows x 1 columns]
```

Figure 3 : Le rendement ‘Returns’ du prix de l'action calculé en fonction de 'Close'

Création et entraînement du modèle de prédiction

```
# On divise les données en caractéristiques d'entrée et variable cible
X = historical_data['Close'].values.reshape(-1, 1)
y = historical_data['returns'].values

#On entraîne le modèle de régression linéaire
model = LinearRegression().fit(X, y)
```

"reshape(-1, 1)" est utilisé pour transformer le tableau 1D en un tableau 2D avec une seule colonne. Cela est nécessaire car les algorithmes de scikit-learn attendent que les données d'entrée soient un tableau 2D. L'argument "-1" signifie que le nombre de lignes sera inféré à partir de la longueur des données et du nombre de colonnes. Ce reshaping nous permet d'utiliser les données comme variables d'entrée pour entraîner un modèle d'apprentissage automatique.

Nous avons utilisé un modèle de **régression linéaire** comme modèle de prédiction car il s'agit d'un modèle simple qui est facile à comprendre et à utiliser pour une première approche. De plus, la régression linéaire permet de déterminer la relation linéaire entre deux variables, ce qui est utile pour prédire la variable cible en fonction des variables explicatives dans notre cas les données historiques de clôture des prix.

Prédiction

```
# Obtenir les 10 prochains jours de prix des actions
next_10_days = np.array([i for i in range(int(X[-1, 0]), int(X[-1, 0]+10))]).reshape(-1, 1)
#On utilise le modèle pour faire des prédictions
predictions = model.predict(next_10_days)
print("Prédiction des rendements des 10 prochains jours")
print(predictions)
```

Affichage

```
closing_prices = []
last_close = historical_data['Close'].values[-1]
for r in predictions:
    last_close = last_close * (1 + r)
    closing_prices.append(last_close)

print("Prédiction du prix des actions des 10 prochains jours")
print(closing_prices)
```

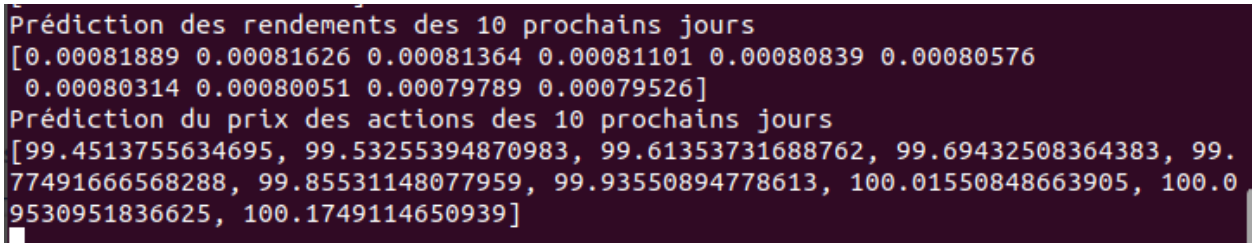

Figure 4 : Prédiction du rendement et du prix des actions des 10 prochains jours

```
#Tracez les prix de clôture prédits
plt.plot(closing_prices)
plt.xlabel('Jour')
plt.ylabel('Prix de clôture')
plt.title('Prix de clôture prédits')
plt.show()
```

Voir la courbe en Annexe.

Glossaire :

yfinance : La bibliothèque yfinance est utilisée pour faciliter la récupération des données financières à partir de Yahoo Finance. Il s'agit d'un module Python qui fournit une interface simple et pratique pour extraire les données financières pour une variété d'actions et d'instruments financiers. Avec yfinance, vous pouvez extraire des données telles que les prix de clôture historiques, les données sur les dividendes, les données sur les splits d'actions, etc. de manière fiable et efficace, sans avoir à naviguer manuellement sur le site web de Yahoo Finance.

Annexe :

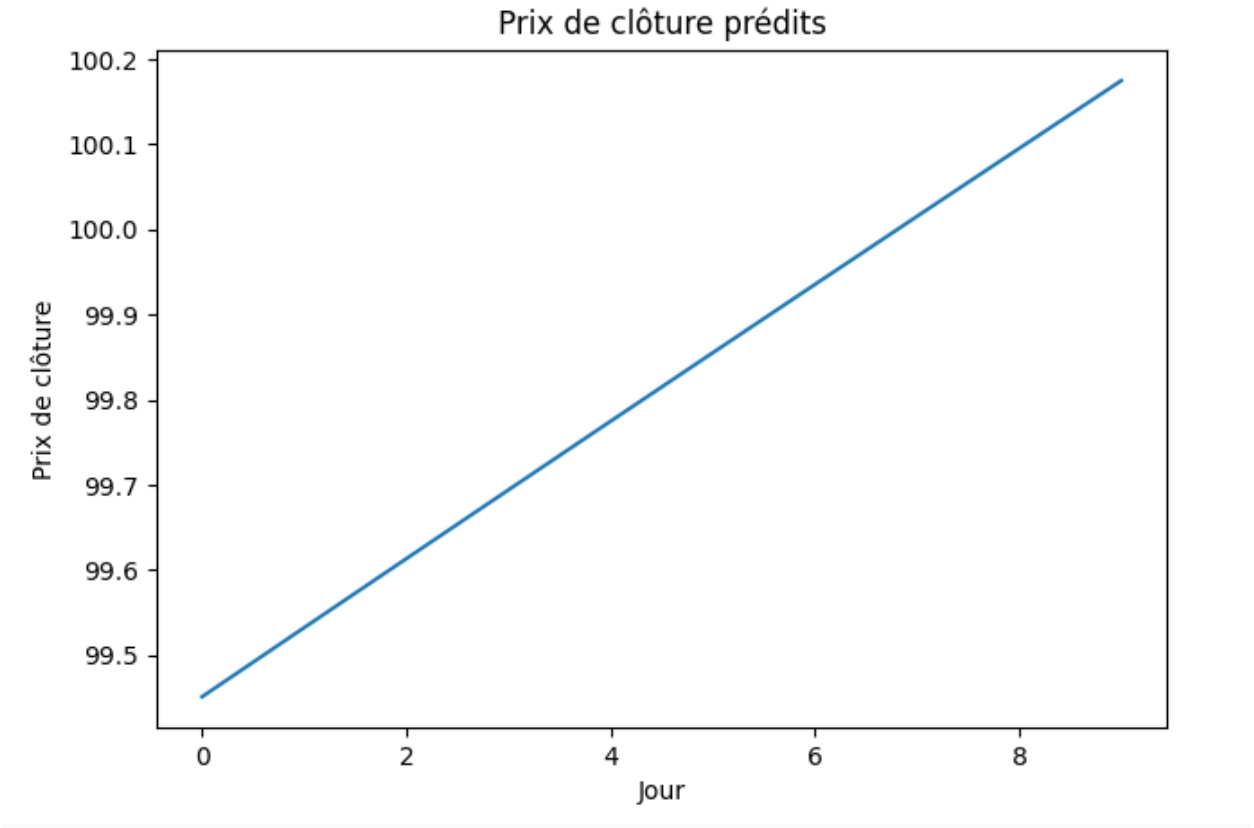

Figure 5 : Prédiction du prix des actions Google des 10 prochains jours