



Projet Algorithmique: Prévision des cours boursiers des entreprises du CAC 40.

PAR GNAGO YANNICK.

ENSEIGNANT: ABDOUL GAMBO.

SOMMAIRE

OBJECTIF:

- LE PROCESSUS DE COLLECTE DE DONNÉES
- LA MISE EN PLACE D'ALGORITHME DE PRÉDICTION

1-La regression linéaire.2-La régression polynomiale.3-La Moyenne Mobile.



III LA CONCLUSION ET LES PERSPECTIVES

RÉFÉRENCES

Objectif:

L'objectif est de mettre en place un outil d'aide à la décision permettant de guider les décisions d'un investisseur en bourse. Nous avons choisi une entreprise du CAC 40, plus précisément Stellantis ¹. Notre travail se divisera donc en 3 étapes : La création d'un crawler, la mise en place d'algorithme de prédiction et la conclusion.

I- Le processus de collecte de données : Le crawler Voir le script crawler.py

Le crawler est un algorithme capable de collecter des données sur des sites web. Dans notre cas, nous avons récupérer 200 valeurs de la bourse Stellantis en temps réel à intervalle de 5 secondes. Nous nous sommes servis de la bibliothèque BeautifulSoup qui nous a permet de récupérer la balise en référence avec la valeur de Stellantis dans le code HTML ² de la page Web; ensuite à travers la boucle 'for' nous réitérons 200 fois cette opération à intervalle de 5 secondes; puis grâce à la bibliothèque Pandas nous avons créé un dataframe contenant en colonnes : le temps et la valeur de Stellantis. Après la création du dataframe, nous avons fait quelques modifications afin de rendre notre Dataframe exploitable, nous avons convertir d'abord la colonne des valeurs en nombre vu qu'elle était en caractère Object à cause de l'extraction avec BeautifulSoup et pour finir nous avons enregistrer le dataframe en fichier csv pour qu'il soit modifier automatiquement dans le fichier à chaque lancement de l'algorithme.

```
the crawler
from pylab import *
from numpy import *
from numpy import ftpack
import pandas s pd
import time
import pandas s pd
import time
import date ime
from bs4 import beautifulSoup

##collecter du contenu du lien URL
page=requests.get(url2)
page content

##utiliser beautifulSoup pour afficher
soup=BeautifulSoup(page.content, "html.parser")

##oution pour l'extraction de la valeur de bourse
soup-find_all('div',class_="c-faceplate_body")[0].find('span').text

def val():
    page=requests.get(url2)
    soup=BeautifulSoup(page.content, "html.parser")
    valeur=soup.find_all('div',class_="c-faceplate_body")[0].find('span').text

tab=[] #creation du dataframe
for step in range(1,3):
    time.sleep(1)
    time.sleep(1)
    time.sleep(3)
    time.sleep(3)
    time.sleep(1)
    df = pd.Dataframe(tab)

## Modification de la data
df.info()
df.dtypes
## Modification de la data
df.info()
df.dtyces
## Save to csv
pd.read_csv('data', index=False, encoding='utf-8') #save en csv
pd.read_csv('data'), index=False, encoding='utf-8') #save en csv
pd.read_csv('data', index=False, encoding='utf-8') #save en csv
```

FIGURE 1 – Le script du crawler

II-La mise en place d'algorithme de prédiction

De façon générale, la mise en place de l'algorithme de prédiction ne se fera pas sur toutes les 200 valeurs boursières récupérées, nous avons donc diviser la donnée en 2 parties, la partie apprentissage c'est-à-dire ("app" de t=1 à t=132) qui nous permettra d'établir un modèle prédictif et le test ("test" de t=133 à t=199) qui nous permettra de faire des vérifications avec le modèle prédictif. Au total, Nous avons pu développer 3 modèles prédictifs que nous essayerons d'expliquer au mieux dans la suite.

1. La regression linéaire[4]

On cherche ici la droite qui colle le mieux au nuage de points afin de faire des prédictions futures. Notre script comprend 8 parties :

^{1.} Stellantis est un groupe automobile multinational fondé le 16 janvier 2021 résultant de la fusion du groupe PSA et de Fiat Chrysler Automobiles.

^{2.} HTML: HyperText Markup Language

Les bibliothèques : L'insertion de toutes les bibliothèques nécessaires à notre script en particulier sklearn.model_selection, sklearn.linear_model et sklearn.metrics.

L'importation du data : Nous avons importé les données enregistrées sous forme csv dans le crawler et nous avons créé des vecteurs (x pour le temps, t pour le pas et y pour la valeur).

Le nuage de points : On constate que les valeurs ont une tendance décroissante dans l'ensemble.

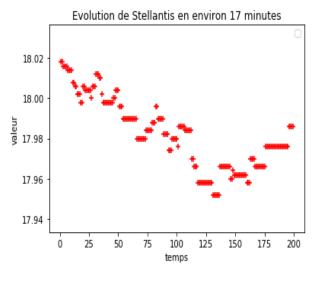


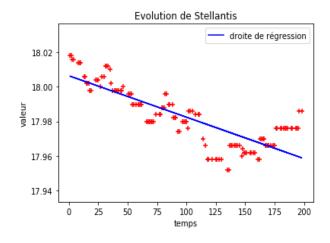
Figure 2

La division du data : il s'agit de la création du modèle d'apprentissage tel que définit précédemment grâce à train test split.

La construction du modèle : l'application de la régression linéaire sur les valeurs d'apprentissage.

La prédiction : Il s'agit de prédire des valeurs futures suivant le modèle précédemment définit par les valeurs d'apprentissage.

La visualisation du résultat : On constate que la droite de régression ne décrit pas parfaitement le nuage de points car il y'a certaines valeurs qui ne sont pas pris en compte, elle décrit l'évolution décroissante des valeurs mais vers la fin le nuage a tendance à remonter.



Evaluation du modèle : Il s'agit ici de calculer les métriques afin d'évaluer la qualité d'un estimateur de cette régression.

• Calcul du MAE³ et du MSE⁴

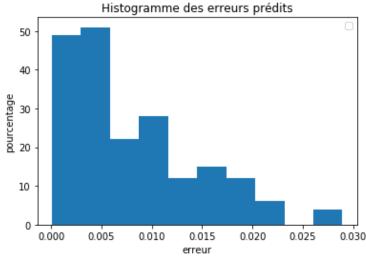
On a: MAE = 0.00805 et MSE = 0.0103

L'erreur entre les valeurs et les prédictions par ce modèle semble être petit. Maintenant essayons de calculer approximativement l'erreur de cette prédiction.

L'erreur absolue moyen

On a : $median_abs_err = 0.00579$ c'est à dire que l'erreur moyen de prédiction de la régression linéaire est d'environ 0.00579.

• L'histogramme des erreurs.



Il y'a donc environ 50% des valeurs prédites qui ont une erreur proche de 0; C'est-à-dire que la moitié des prédictions semblent correctes.

• Calcul de R^{25}

Il évalue la performance du modèle par rapport au niveau de variation présent dans les données.

On a : $R^2 = 0.648 \Longrightarrow$ Ce modèle prédictif décrit 64% des variations des valeurs de stellantis récupérées.

Voir le script reg_linear.py

2. La régression polynomiale[5]

On cherche dans ce cas la courbe qui correspond au mieux au nuage de points. Le principe est basé sur l'interpolation polynomiale c'est-à-dire on cherche le polynôme qui t'interpole au mieux tout le nuage de points.

Les premiers étapes de cette méthode sont pareilles à la précédente. Après l'affichage du nuage de points , on a :

La construction du modèle de régression polynomiale : On définit notre polynôme qui collera au mieux au nuage en faisant varier le degré de ce polynôme.

La prédiction : il s'agit de trouver les valeurs futures suivant le modèle précèdent des valeurs d'apprentissage.

La visualisation du résultat : On constate que ce graphe suit au mieux la tendance du nuage de points avec un polynôme de degré 9, mais il y'a toujours des valeurs qui ne sont pas pris en compte.

^{3.} MAE: l'erreur absolue moyenne

^{4.} MSE : l'erreur quadratique moyenne

^{5.} \mathbb{R}^2 : Le coefficient de corrélation de Pearson

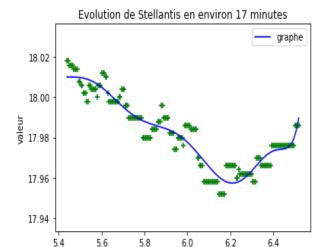


FIGURE 3

temps

Evaluation du modèle :

Calcul du MAE et du MSE

On a: MAE = 0.00446 et MSE = 0.00539

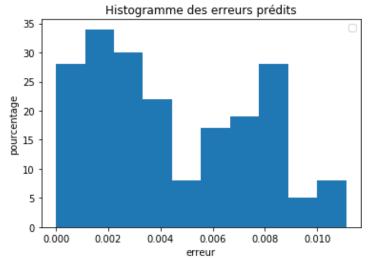
L'erreur entre les valeurs et les prédictions par ce modèle semble être petit. Maintenant essayons de calculer approximativement l'erreur de cette prédiction.

le12+1.63697e18

• L'erreur absolue moyen

On a : $median_abs_err = 0.00366$; c'est à dire que l'erreur moyen de prédiction de la régression polynomiale est d'environ 0.00366.

L'histogramme des erreurs.



Il y'a donc environ 35% des valeurs prédites qui ont une erreur proche de 0.

• Calcul de \mathbb{R}^2

On a : $R^2 = 0.903 \Longrightarrow$ Ce modèle prédictif décrit 90% des variations des valeurs de stellantis.

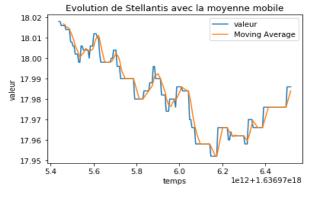
Voir le script reg poly.py

3. Moyenne Mobile[6]

Il s'agit d'un même principe mais cette fois ci en se servant des moyennes mobiles. Elle consiste à calculer la moyenne des données à des intervalles spécifiques.

Après les bibliothèques et la lecture du fichier csv, les étapes suivantes sont :

La visualisation : On constate que cette ligne a une tendance décroissante et elle est très volatile. Voir la ligne bleu FIGURE 4



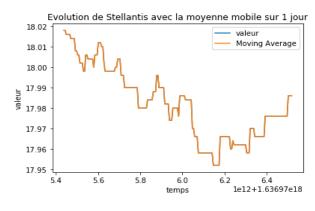


FIGURE 4 FIGURE 5

Le calcul et affichage de la moyenne mobile: Il s'agit ici de calculer la moyenne mobile sur un nombre fenêtre (5 jours) fixé et de représenter cette moyenne afin de la comparer à la ligne précédente. On constate que le graphe de la moyenne mobile a une forme un plus arrondi et qu'elle suit au mieux la tendance mais pas pour toutes les valeurs Voir la ligne orange FIGURE 4.

On remarque aussi que le graphe des moyennes mobiles colle parfaitement à celui des valeurs sur un (1) jour, ce qui semble d'autant normal vu que nos données ont été recueillis en 17 minutes Voir FIGURE 5.

Evaluation du modèle :

Calcul du MAE et du MSE

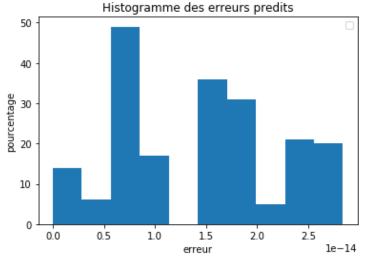
On a: MAE = 1.41e - 14 MSE = 1.63e - 14

L'erreur entre les valeurs et les prédictions par ce modèle est très négligeable. Maintenant essayons de calculer approximativement l'erreur de cette prédiction.

• L'erreur absolue moyen

On a : $median_abs_err = 1.42e - 14$ c'est à dire que l'erreur moyen de prédiction de la moyenne mobile est faible, soit environ 1.42e - 14. Cela prouve que moyenne mobile est un meilleur modèle prédictif comparé aux régressions.

• L'histogramme des erreurs.



Il y'a donc environ 15% des valeurs prédites qui ont une erreur proche de 0.

• Calcul de R^2 On a : $R^2 = 1.0 \Longrightarrow$ Ce modèle prédictif décrit 100% des variations des valeurs de stellantis.

Voir le script reg moy mobile.py

^{6.} La moyenne mobile est une moyenne statistique utilisée pour analyser des séries temporelles, en supprimant les fluctuations transitoires de façon à en souligner les tendances à plus long terme.

III-La conclusion et les perspectives

Ce projet nous a permis d'utiliser le logiciel Python afin d'évaluer les prédictions sur des données directement recueillis sur une page Web. Nous pouvons noter qu'il est très instructif d'un point de vue informatique, statistique et économique. Au niveau de résultat, nous pourrions dire que la régression linéaire montre bien la tendance décroissante de Stellantis en cette période, mais il y'a assez de valeurs aberrantes soit 64% des variations de Stellantis sont représentés par la régression linéaire. Quant à la régression polynomiale qui permet de représenter un peu plus mieux la tendance, nous avons 90% de ces variations représentent bien Stellantis en cette période, ce qui est d'autant mieux pour la prédiction des investisseurs et enfin pour la moyenne mobile, nous avons pu voir que cette méthode représente 100% des variations de Stellantis à condition d'avoir bien choisir le nombre de jours(ici dans notre cas window=1) et que l'erreur de prédiction est d'autant très négligeable avec la méthode des moyennes mobiles.

En somme, pour pouvoir conseiller les investisseurs qui essaient de se positionner sur la valeur de Stellantis en cette période, nous les recommanderons la méthode prédictive des moyennes mobiles sur une fenêtre d'un jour vu qu'elle représente 100% des variations de Stellantis et suit parfaitement sa tendance sur les 17 minutes. Alors comme Stellantis est assez volatile et de plus on arrive à voir que cette tendance commence à croître vers la fin, nous ne pourrons que conseiller nos investisseurs à court terme de prendre une position d'achat (tendance croissante).

À vrai dire, les méthodes prédictives ne sont jamais à 100% idéale vu qu'il y'a tellement de facteurs qui peuvent fausser des prédictions boursières comme des imprévus de la vie qui peuvent agir indirectement sur l'évolution boursières par exemple le krach boursier 2021 dû à la pandémie COVID ou même l'interview du footballeur Cristiano Ronaldo pendant l'euro 2021 qui a fait chuter la valeur Coca-Cola.

Il faut aussi noter que notre travail n'est qu'une approximation simpliste de la prédiction boursière, car pour pouvoir mieux conseiller les investisseurs, il faudrait utiliser certaines méthodes prédictives bien plus appropriées et compliquées afin de trouver les meilleurs indicateurs.

Références

- [1] Analyse de données en Python, de Wes McKinney.
- [2] Python pour le data scientist : des bases du langage au machine learning, de Emmanuel Jakobowicz.
- [3] scikit-learn
- [4] Comment mettre en œuvre une régression linéaire avec python?
- [5] Régression polynomiale avec Python
- [6] Moyenne mobile pour NumPy Array en Python