

DM 1 – PREVISION DES COURS BOURSIERS (ACTIONS) DU CAC 40

ETUDE DES ACTIONS DE L'ENTREPRISE AIR FRANCE LE 25
NOVEMBRE DE 10H A 12H

ETUDE : QUELS OUTILS PEUT-ON METTRE EN PLACE POUR GUIDER LES DECISIONS
D'UN INVESTISSEUR ?

TABLE DES MATIERES

| | | |
|----|---|---|
| 1. | INTRODUCTION | 2 |
| 2. | QUELS OUTILS PEUT-ON METTRE EN PLACE POUR GUIDER LES DECISIONS D'UN INVESTISSEUR? | 2 |
| | ETAPE 1 : CREATION D'UN CRAWLER | 2 |
| | ETAPE 2 : PREDICTION DES ACTIONS DE L'ENTREPRISE AU TEMPS T+1 | 2 |
| | ETAPE 3 : PREDICTION A N'IMPORTE QUEL TEMPS T | 3 |
| 3. | CONCLUSION | 3 |
| 4. | ANNEXES | 4 |

1. INTRODUCTION

Prévoir l'évolution des actions en bourse est quelque chose de compliqué et il est facile de perdre de l'argent. C'est pourquoi la prévision des cours boursiers est si importante pour les investisseurs.

L'étude que nous réalisons a pour données les actions de l'entreprise Air France du 25 novembre 2020 de 10h à 12h ([figure1](#)).

Au regard de ces données, nous pouvons nous demander : quels outils peut-on mettre en place pour guider les décisions d'un investisseur ?

Pour répondre à cette question, nous procéderons à la création de différents algorithmes, de différents tests statistiques, ainsi qu'à l'interprétation de leurs résultats. Nous nous appuierons sur des illustrations graphiques.

2. QUELS OUTILS PEUT-ON METTRE EN PLACE POUR GUIDER LES DECISIONS D'UN INVESTISSEUR?

ETAPE 1 : CREATION D'UN CRAWLER

Nous avons créé un programme qui récupère les actions d'une entreprise au cours du temps, sur le site boursorama.com. Pour cette étude, nous avons choisi de récolter 61 actions, chacune récupérée à 2 minutes d'intervalle pendant une période de deux heures ([figure1](#)). Nous remarquons que le cours des actions augmente au cours du temps. Cependant, nous ne pouvons pas affirmer que ces valeurs vont continuer d'augmenter.

ETAPE 2 : PREDICTION DES ACTIONS DE L'ENTREPRISE AU TEMPS $T+1$

Nous avons mis en place un algorithme qui prédit le cours des actions 2 minutes après leur arrivée en bourse ([figure2](#)). Nous pouvons donc estimer les futures valeurs de la bourse et les comparer avec les valeurs réelles ([figure3](#)). Cette modélisation semble pertinente ([figure4](#)).

Comme l'algorithme se base à chaque fois sur une valeur réelle de la bourse, les prédictions obtenues ne dépendent finalement que de ces valeurs réelles de la bourse au temps t . Cela permet d'éviter des erreurs. Cependant, cela constitue également la limite de ce modèle, car il est nécessaire de connaître une valeur de la bourse pour prédire la suivante.

Grâce à une extrapolation mathématique ([figure5](#)), nous pouvons aller plus loin que notre prédiction vue en [figure2](#). Désormais, nous pouvons prédire les futures valeurs de la bourse, même pour des valeurs que nous n'avions pas recueillies à l'aide de notre crawler.

Gagner de l'argent en bourse se fait sur le long terme. Nous nous sommes servis de la prédiction vue en [figure5](#) pour créer un modèle de prédiction des futures valeurs de la bourse. Nous pouvons ainsi obtenir les valeurs prédites de la bourse par le modèle pour n'importe quel temps choisi ([figure6](#)). Ce modèle permet de s'affranchir de la connaissance du cours de l'action à un temps t pour prédire la valeur suivante (excepté pour la première valeur qui permet de lancer l'algorithme) : ici, l'investisseur peut retrouver facilement les prédictions des cours de l'action rien qu'en se basant sur un temps choisi.

Nous pouvons observer l'évolution de ces actions au cours du temps pour permettre aux investisseurs de prendre les bonnes décisions ([figure7](#)).

Mais si ce modèle gagne en facilité d'utilisation, il perd un peu en fiabilité car chaque valeur prédite pour un temps $t+1$ se base désormais sur la valeur prédite pour le temps t , et non plus sur la valeur réelle de la bourse au temps t .

3. CONCLUSION

En conclusion, nous avons pu créer un algorithme qui nous permet de prédire une valeur de la bourse à n'importe quel temps t dans le futur.

Nous serions par exemple en mesure de connaître le cours de la bourse de l'entreprise dans 2 mois, et ainsi décider s'il est intéressant d'investir ou s'il est préférable de s'en abstenir.

Cependant, il est probable que si nous avions récolté des données sur un intervalle de temps plus long (des mois, des années), nous aurions construit un modèle de prédiction plus fiable.

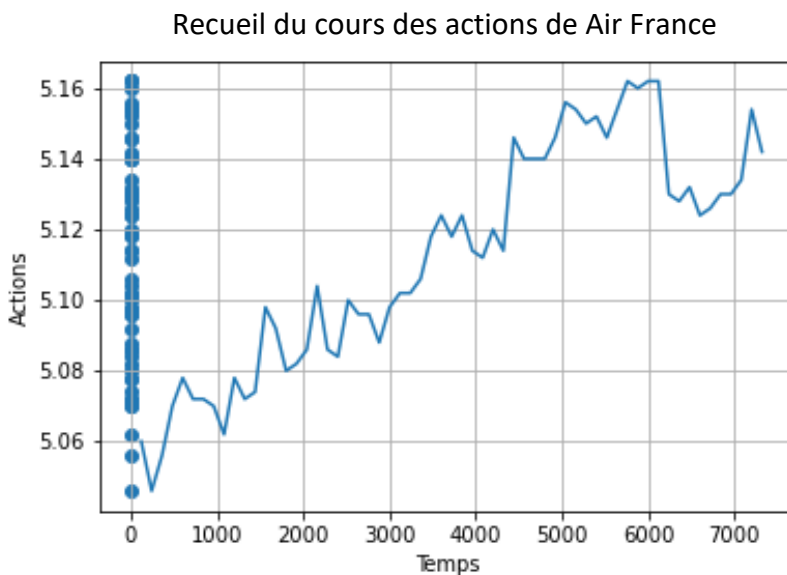


Figure 1 (données des actions d’Air France pendant une période de 2 heures)

Nous avons recueilli les actions en bourse de l’entreprise Air France le 25 novembre 2020 de 10 h à midi. Chacune de ces valeurs a été prise à 2 minutes d’intervalle. Nous obtenons ainsi le graphique des actions en fonction du temps (exprimé en secondes). Nous remarquons sur le graphique que ces valeurs fluctuent beaucoup et que cela serait très difficile visuellement de prédire une future valeur des actions.

```
[5.064901341720569, 5.051855431388075, 5.0611739387684285, 5.074219849100922, 5.081674655005204,
5.076083550576993, 5.076083550576993, 5.074219849100922, 5.06676504319664, 5.081674655005204,
5.076083550576993, 5.077947252053063, 5.10031166976591, 5.094720565337698, 5.0835383564812755,
5.085402057957346, 5.089129460909487, 5.105902774194122, 5.089129460909487, 5.087265759433416,
5.10217537124198, 5.09844796828984, 5.09844796828984, 5.0909931623855575, 5.10031166976591,
5.104039072718051, 5.104039072718051, 5.107766475670192, 5.118948684526616, 5.124539788954827,
5.118948684526616, 5.124539788954827, 5.115221281574475, 5.113357580098405, 5.120812386002687,
5.115221281574475, 5.145040505191604, 5.139449400763392, 5.139449400763392, 5.139449400763392,
5.145040505191604, 5.154359012571956, 5.152495311095886, 5.148767908143745, 5.150631609619816,
5.145040505191604, 5.152495311095886, 5.159950117000168, 5.158086415524098, 5.159950117000168,
5.159950117000168, 5.130130893383039, 5.128267191906969, 5.131994594859109, 5.124539788954827,
5.126403490430898, 5.130130893383039, 5.130130893383039, 5.13385829633518, 5.152495311095886,
5.141313102239463]
```

Figure 2(prédiction des valeurs recueillies)

On définit au préalable le temps $t+1$, ici égal à 2 minutes.

Nous obtenons la prédiction au temps $t+1$ des valeurs récupérées par le crawler. Nous pouvons ainsi les comparer aux valeurs réelles. Nous obtenons ici des valeurs correspondant aux prédictions des cours de l’action, basées sur les valeurs réelles 2 minutes plus tôt. On prédit la valeur que prendra le cours de l’action deux minutes après la sortie de la valeur réelle.

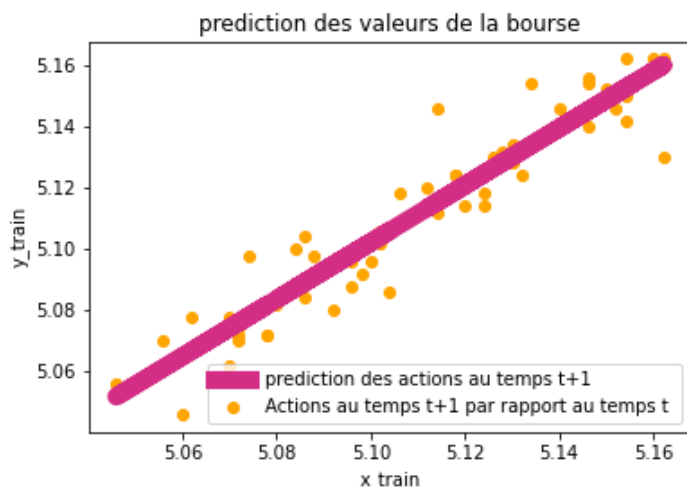


Figure 3(comparaison des valeurs prédites au temps t+1 par rapport aux valeurs réelles)

Ce graphique est la représentation de y_{train} (l'action au temps t+1) en fonction de x_{train} (l'action au temps t), auquel nous avons rajouté la prédiction issue de notre modèle.

On remarque sur ce graphique que les valeurs réelles des actions au temps t+1 (en orange) semblent proches de la droite de régression (notre prédiction en rouge). Nous pouvons donc émettre l'hypothèse que nous avons un bon modèle et que celui-ci va nous être utile pour prédire les actions de la bourse.

0.896680965965527

Figure 4(R carré)

Nous obtenons la valeur R carré qui représente le score de notre régression linéaire. On remarque que le R carré est très proche de 1 donc notre modèle est idéal pour la prédiction.

[0.93185074]
0.3497366072620167

Figure 5(coefficients de la droite de régression)

Nous obtenons les coefficients de la droite de régression avec $a=0,93185074$ et $b=0,34973$. Nous connaissons donc l'équation de cette droite qui est $y = a * x + b$ avec a et b définis ci-dessus. Nous pouvons ainsi nous servir de ces coefficients pour prédire le cours de l'action au temps t+1 de n'importe quelle valeur de l'action au temps t. Il suffit de savoir la valeur d'une action x et de trouver y grâce à l'équation ci-dessus.

```

Console 1/A
In [14]: list
Out[14]:
[array([5.06490134]),
 array([5.06946866]),
 array([5.07372472]),
 array([5.07769073]),
 array([5.08138646]),
 array([5.08483033]),
 array([5.08803951]),
 array([5.09102998]),
 array([5.09381665]),
 array([5.09641341]),
 array([5.09883321]),
 array([5.10108809]),
 array([5.10318931]),
 array([5.10514733]),
 array([5.10697192]),
 array([5.10867216]),
 array([5.11025653]),
 array([5.11173292]),
 array([5.11310871]),
 array([5.11439073]),
 array([5.11558538]),
 array([5.11669862]),
 array([5.11773599]),

```

Figure 6(liste de toutes les prédictions)

Grâce aux coefficients de notre droite de régression (figure5), nous avons pu prédire pour toute valeur au temps t son action au temps t+1. Nous avons créé un algorithme qui se base sur les prédictions précédentes afin de trouver les prédictions futures. Voici quelques petits exemples pour mieux comprendre :

```

test_temps_t_plus_1=predict(test_temps_t)
test_temps_t_plus_2=predict(test_temps_t_plus_1)
test_temps_t_plus_3=predict(test_temps_t_plus_2)

```

Cela nous permet donc de prédire le cours des actions à n'importe quel moment éloigné dans le temps.

On rappelle que $t+1 = 2\text{min}$. Si l'on souhaite par exemple connaître le cours de l'action au temps $t+30$ soit 60 min après notre première valeur (donc à 11h), on tape :

```

In [16]: list[30]
Out[16]: array([5.12385594])

```

On obtient ainsi la valeur prédite pour 11h, égale à 5.1238, calculée grâce aux anciennes prédictions.

Elle est différente de la prédiction au temps $t+1$ que l'on obtenait uniquement grâce à la valeur réelle au temps t. En effet, nous avons récupéré au temps $t+29$ (soit à 10h58) une valeur de 5,124. Lorsqu'on fait sa prédiction avec la méthode vue en figure2 on obtient :

```

In [17]: predict(5.124)
Out[17]: array([5.12453979])

```

On remarque que les deux prédictions ne sont pas identiques, mais que la première peut être calculée pour n'importe quel temps t dans le futur et qu'elle reste proche de la deuxième prédiction.

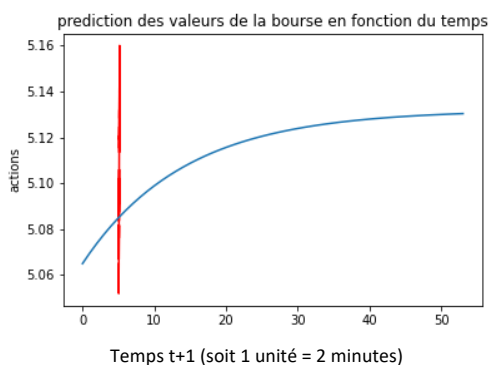


Figure 7(prédiction en fonction du temps)

Nous pouvons donc regarder graphiquement la prédiction des actions en fonction du temps $t+1$, tracée en bleu. On constate que les actions augmentent au cours du temps mais que plus le temps augmente et moins les actions augmentent rapidement. Cela est dû au fait que nous avons obtenu $a=0,931$ comme coefficient de la régression. Les actions risquent même de baisser par la suite car si on se base sur notre modèle, plus l'action est grande et plus elle va diminuer au temps $t+1$, dû aux coefficients de régression.

Exemple : $\text{predict}(5)=5,008$. On remarque que la valeur augmente légèrement.

predict(6)=5,94. On remarque que la valeur diminue légèrement.

Il ne faut pas uniquement se fier au graphique en se disant que si les actions augmentent au cours du temps, alors elles vont continuer à augmenter, mais bien s'appuyer sur le modèle de prédiction.

Figure 8 (Code Python)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Nov 21 17:40:49 2020

@author: angoulvent
"""
import requests
import numpy as np
from bs4 import BeautifulSoup
import time
import matplotlib.pyplot as plt
from scipy import stats

def crawler_Air_France(nb_act,int_tps):

    data = np.array([range(1,nb_act+1),range(1,nb_act+1)], dtype = float)

    for i in range(0,nb_act):

        req = requests.get("https://www.boursorama.com/bourse/actions/cotations/")
        soup = BeautifulSoup(req.text, "lxml")

        table = soup.find("table", class_="c-table c-table--generic c-table--generic c-shadow-overflow__table-fixed-column
c-table-top-flop")

        row = table.find_all("tr")

        row_Air_France = row[8].find_all("span")
        action_Air_France = row_Air_France[0].text.split("<")

        print(float(action_Air_France[0]))

        data[0,i] = float(action_Air_France[0])
        data[1,i] = (i+1)*int_tps
        time.time()

        if i == nb_act:
            break

        time.sleep(int_tps)

    return data

nb_act=61
int_tps=120
z= crawler_Air_France(61,120)
```



```

plt.title("recueil du cours des actions de Air France")
plt.xlabel("Temps")
plt.ylabel("Actions")
plt.plot(z[1:],z[0,])

#partie 1 Collecte et preparation des données

test=z[0,]

x_train=np.linspace(0,0,nb_act-1)
y_train=np.linspace(0,0,nb_act-1)

for i in range(0,nb_act-1):
    x_train[i]=test[i]
    y_train[i]=test[i+1]

#partie 2 Entrainement du modele
from sklearn import linear_model
x_train=x_train.reshape(nb_act-1,1)

reg = linear_model.LinearRegression()
reg.fit(x_train, y_train)

axes = plt.axes()
axes.grid() # dessiner une grille pour une meilleur lisibilité du graphe
plt.scatter(x_train,y_train)
plt.show()

slope=reg.coef_
print(reg.coef_)

intercept=reg.intercept_
print(reg.intercept_)

#cours_t_plus_1 = slope * cours_t _ intercept

#partie 3 Prediction
print(test)

preds = []

for x in test:
    preds.append(reg.predict([[x]]))

preds = [i[0] for i in preds]
print(preds)

# partie 4 Evaluation du modele

v=reg.score(x_train, y_train)
print(v)

plt.scatter(x_train,y_train,c="orange",label='Actions au temps t+1 par rapport au temps t ')
plt.legend()

```

```

plt.plot(x_train,reg.predict(x_train),"-r",color="#D42D85", linewidth=10, label='prediction des actions au temps t+1')
plt.legend()
plt.xlabel("x_train")
plt.ylabel("y_train")
plt.title ("prediction des valeurs de la bourse")
plt.show()

def predict(x):
    return slope * x + intercept

fitLine = predict(x_train)
plt.plot(x_train, fitLine, c='r')

print(predict(12.98))
print(predict(20.3))
print(predict(13.10))
print(predict(12.8))
print(predict(17.2))
print(predict(18.997))

test_temps_t=test[0]
print(test_temps_t)

test_temps_t_plus_1=predict(test_temps_t)
print(test_temps_t_plus_1)

test_temps_t_plus_2=predict(test_temps_t_plus_1)
print(test_temps_t_plus_2)

test_temps_t_plus_3=predict(test_temps_t_plus_2)
print(test_temps_t_plus_3)

uold=test[0]
list=[]
unew=0
n=54
for i in range (n) :
    unew= predict(uold)
    list.append(unew)
    uold=unew

list2=range(54)
plt.plot(list2,list)
plt.xlabel("temps")
plt.ylabel("actions")
plt.title ("prediction des valeurs de la bourse en fonction du temps")

```