



# Machine learning

## Prévision des mouvements des marchés financiers

FAYZI Houssam



# Plan

- ☐ Comprendre le Machine learning
- ☐ Application du Machine learning
- ☐ Langages
- ☐ Types
- ☐ Implémentation du Machine learning algorithme  
avec R



# C'est quoi ML ?

- L'apprentissage automatique(en anglais machine learning est un type d'intelligence artificielle qui confère aux ordinateurs la capacité d'apprendre sans être explicitement programmés.
- Il consiste en la mise en place d'algorithme ayant pour objectif d'obtenir une analyse prédictive à partir de données, dans un but précis.

# Comprendre le ML



New Data



Machine



Result





## Exemples :

- La voiture autonome de Google
- Classification des emails Gmail
- Moteur de recherche de Google
- La traduction en temps réel de Skype.
- la reconnaissance vocal de Siri d'Apple.
- La reconnaissance facial.



- Les algorithmes de Machine learning utilisent donc nécessairement une phase dite d' apprentissage.
- Les programmes d'apprentissage automatique détectent des schémas dans les données et ajustent leur fonctionnement en conséquence.



# Difference Entre Machine Learning / Data Mining

- **Data Mining** : retraiter les données déjà connues pour en sortir des propriétés et des précisions encore inconnues.
- **Machine Learning** : apprendre aux systèmes à prédire ce que pourrait être le résultat sorti de données encore inconnues à partir de données connues.

# Les Language de Machine learning





# Why R ?

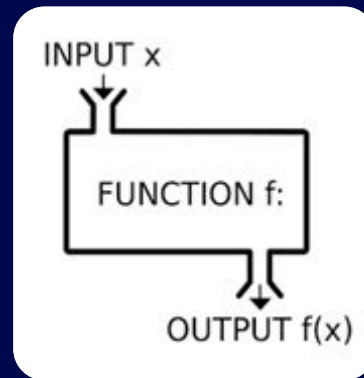
Orienté objet



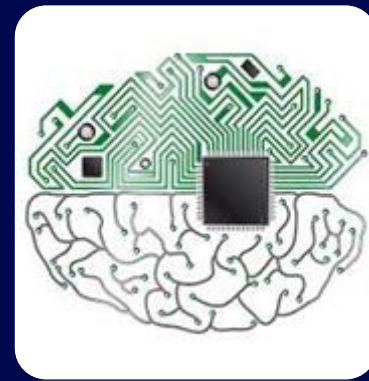
Open source



programmation  
fonctionnelle



Turing complet



# Les étapes du machine learning





# Les Types du ML

Dans ce chapitre, je vais vous présenter les grandes familles d'algorithmes d'apprentissage existantes.

Apprentissage "supervisé" ou "non supervisé"



# Apprentissage supervisé

- Règle de Bayes
- Classification naïve bayésienne
- Régression multivariée
- Régression régularisée
- Protocole d'apprentissage
- Les k plus proches voisins
- Dilemme biais/variance
- Arbre de décision
- Bagging
- Forêt aléatoire
- Perceptron
- Perceptron multicouche
- Les réseaux de neurones
- Deep learning



# Apprentissage supervisé

- K-moyennes
- Cartes auto-organisatrices






# PARTIE II



# **Implémentation du Machine Learning en prédictions des mouvement des marchés financiers**



# Implémentation du Machine Learning en prédictions des mouvements des marchés financiers



## Données utilisés

- Les données quotidiennes historiques du EUR/USD (13/03/2012 → 13/03/2020)
- 2074 lignes
- 4 colonnes



# Explication des données

- Noms et explication des colonnes :
- Date : le jour concerné
- Prix\_debut : le prix à 00:00 (début de la journée)
- Prix\_haut : prix maximum pendant la journée
- Prix\_bas : prix minimum pendant la journée
- Prix\_fin : le prix à la fin de la journée

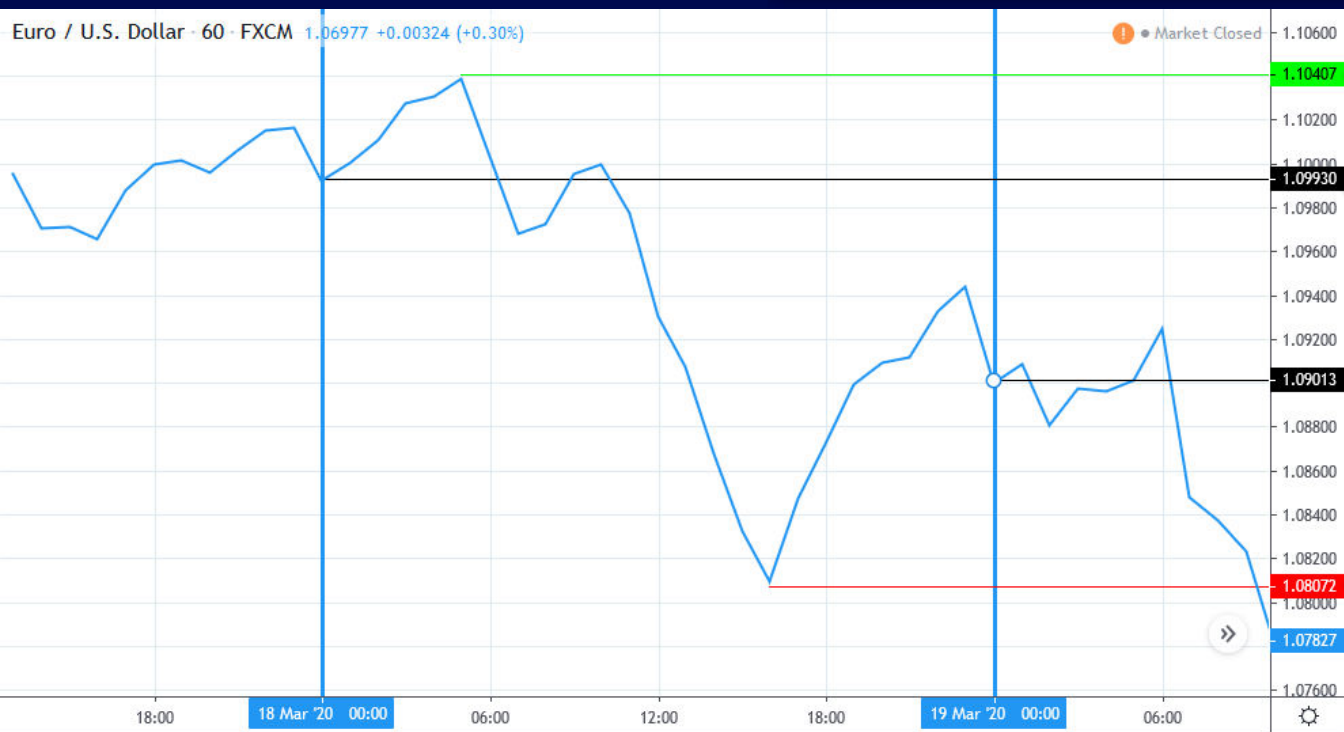




# Terminologie

- EURUSD : le taux d'échange entre l'EURO et le DOLLAR Américain
- Exemple : prix = 1,35000  $\rightarrow$  1 EUR = 1,35 USD
- Le marcher des devises est le plus grand marcher du monde (capital de 5 trillions dollars par jour échangé entre banques, individus)

# Visualisation



Prix plus haut du jour

Prix au début du jour

Prix à la fin du jour

Prix plus bas du jour

Un jour en direction baissière  
(prix\_fin < prix\_debut)  
18 Mars 2020



# **Prédiction du Prix du fin de la journée en utilisant la méthode KNN**

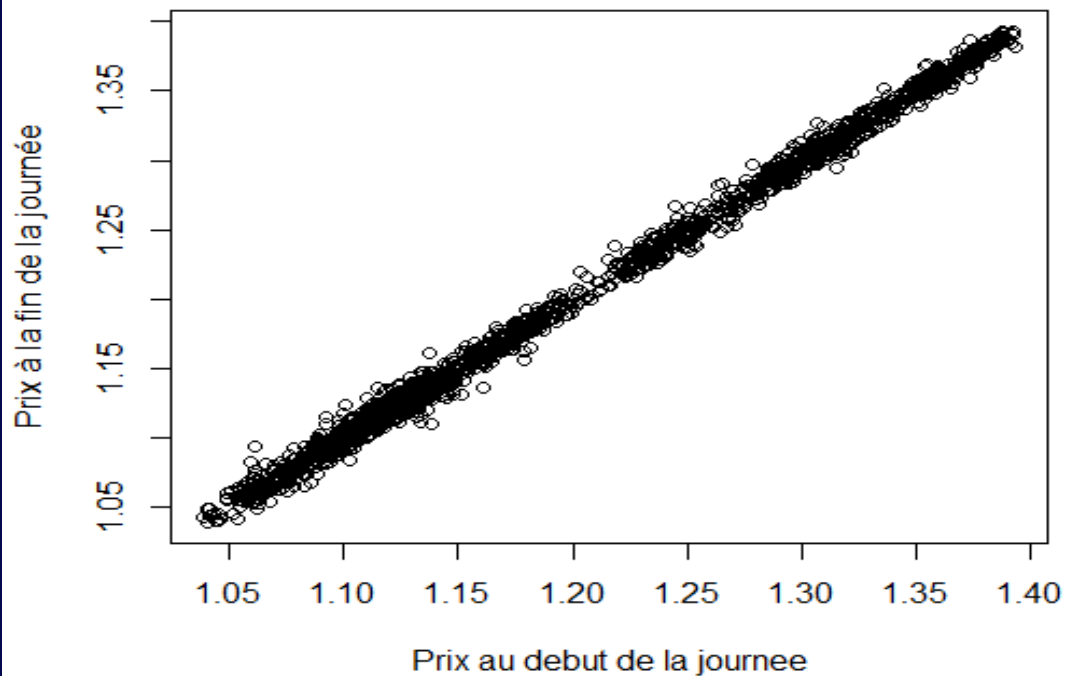


# Chargement des données

```
donnees5<-read.csv("C:/Users/HOUSSAM/Desktop/EURUSD.csv",stringsAsFactors = FALSE)
```

**Importation des données depuis un fichier CSV**

# Visualisation







## Remarques

- On observe une forte corrélation entre le prix au début de la journée et le prix à la fin
- Très petite variance entre les points
- La régression linéaire peut être la meilleure méthode de prédire le future puisqu'on a des point sous forme d'une ligne



# Test de corrélation

- On utilise la fonction `col()` pour déterminer le niveau de corrélation entre les variables :

```
> cor(donnees5$prix_debut, donnees5$prix_fin)  
[1] 0.9982876
```

- 99% de corrélation positive entre le prix de début et le prix de la fin



# Test de corrélation

- On va prendre les colonnes (prix\_debut, prix\_haut, prix\_bas, prix\_fin) pour savoir la corrélation entre eux :

```
> donnees5.subset<-donnees5[,c(1:4)]  
> cor(donnees5.subset)
```

	prix_debut	prix_haut	prix_bas	prix_fin
prix_debut	1.0000000	0.9991635	0.9992184	0.9982876
prix_haut	0.9991635	1.0000000	0.9989152	0.9992314
prix_bas	0.9992184	0.9989152	1.0000000	0.9992202
prix_fin	0.9982876	0.9992314	0.9992202	1.0000000



- Donc notre variable cible (`prix_fin`) est très prédictible



# Normalisation

```
normalize <- function(x) {  
  return ((x - min(x)) / (max(x) - min(x))) }  
  
donnees5.subset.norm<-as.data.frame(lapply(donnees5.subset, normalize))
```





# Training set

- Contient 70% des données choisi par hasard

```
dat.d <- sample(1:nrow(donnees5.subset.norm),  
               size=nrow(donnees5.subset.norm)*0.7,  
               replace = FALSE)  
train <- donnees5.subset.norm[dat.d,] # 70% training data  
train_labels <- donnees5.subset[dat.d,4]
```



## Test set

- Contient 30% des données choisi par hasard :

```
test <- donnees5.subset.norm[-dat.d,] # 30% test data  
test_labels <- donnees5.subset[-dat.d,4]
```

- On a choisi  $k=2$  qui a donné les meilleurs résultats (marge d'erreur minimum)

```
donnees5_pred <- unfactor(knn(train = train, test = test, cl = train_labels, k=2))
```

- La fonction unfactor() empêche la transformations des données en facteurs



# Evaluation du modèle

- Pour évaluer notre modèle on a créé une nouvelle dataset qui va stocker les résultat de knn (prédite) et les résultat réelles (observé)

```
# fusionner `donnees5_pred` et `testLabels`  
fusion <- data.frame(donnees5_pred, testLabels, stringsAsFactors = FALSE)  
  
# noms des colonnes  
names(fusion) <- c("Predite", "Observe")
```



# Evaluation du modèle

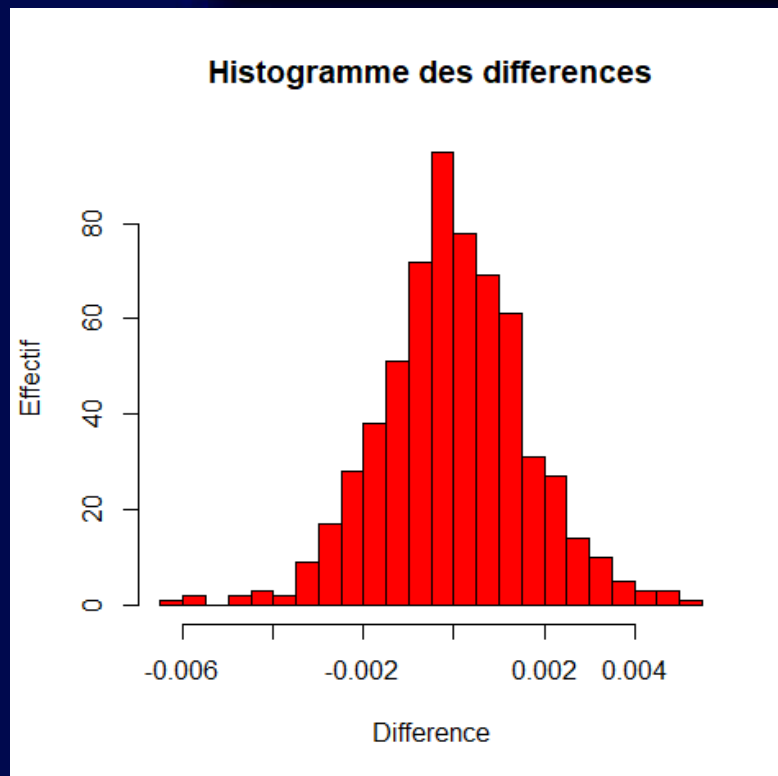
- Ensuite on a calculé la différence entre ces deux variables pour savoir la marge d'erreur

```
> fusion$Difference<-(fusion$Predite-fusion$Observe)
> head(fusion)
```

	Predite	Observe	Difference
1	1.30677	1.30780	-0.00103
2	1.31618	1.31735	-0.00117
3	1.32544	1.32366	0.00178
4	1.32351	1.32118	0.00233
5	1.32566	1.32690	-0.00124
6	1.33462	1.33569	-0.00107



# Evaluation du modèle





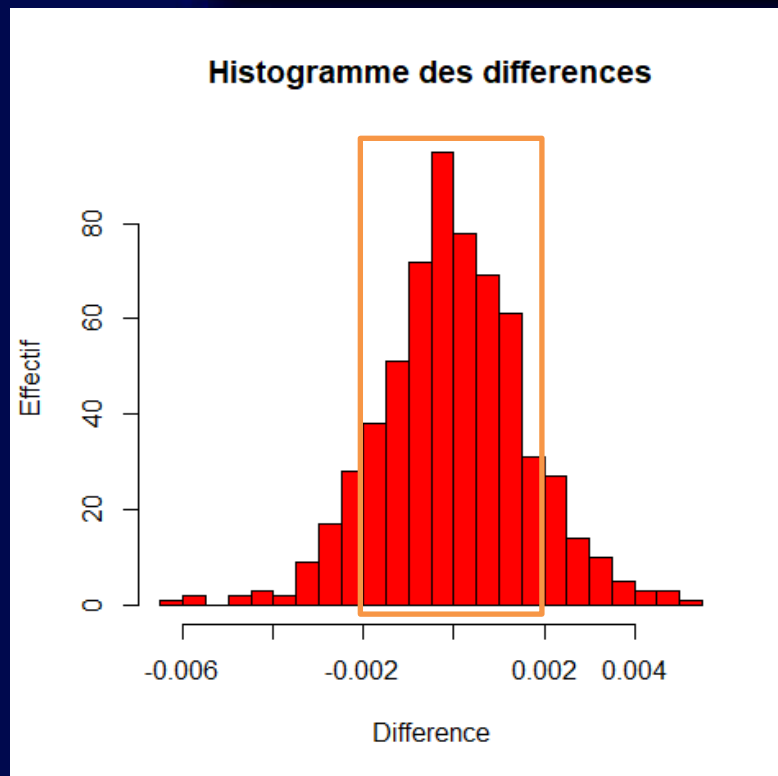


## Remarques

- On observe une distribution normale sur l'histogramme des différence donc on peut avoir un intervalle qui contient la majorité des différences (erreurs)



# Evaluation du modèle





# Evaluation du modèle

- Donc après avoir déterminer graphiquement l'intervalle , on peut dire qu'on a une marge d'erreur de  $\pm 0,002$
- Conclusion du modèle :
- Prix à la fin de la journée = Prix prédit par KNN  $\pm 0,002$



# **Prédiction du Prix du fin de la journée en utilisant la méthode de la régression linéaire**



## Remarques

- On a déjà observé le nuage des point antérieurement et mentionné que cette méthode peut être plus appropriée



# Variables utilisées

- Dans cet exemple on a prédit la prix de la fin par le prix au début :

```
donnees5.subset1<-donnees5[,c(1,4)]  
donnees5.regression<-lm(prix_fin ~ prix_debut,data=donnees5.subset1)  
summary(donnees5.regression)
```



# Interprétation des résultats

Quelques informations sur les résidus , on a une distribution symétrique

```
Call:
lm(formula = prix_fin ~ prix_debut, data = donnees5.subset1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.028232	-0.003414	-0.000016	0.003317	0.032344

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.002355	0.001539	1.53	0.126
prix_debut	0.997994	0.001285	776.63	<2e-16 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.005788 on 2071 degrees of freedom  
Multiple R-squared: 0.9966, Adjusted R-squared: 0.9966  
F-statistic: 6.032e+05 on 1 and 2071 DF, p-value: < 2.2e-16

# Interprétation des résultats

Informations sur les moindres-carrés

$a = 0,997994$

$b = 0,002355$

$Y = 0,997994X + 0,002355$

$p\text{-value} < 2e-16 < 0,05$

Donc le prix\_debut fait une estimation fiable du prix\_fin

```
Call:
lm(formula = prix_fin ~ prix_debut, data = donnees5.subset1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.028232	-0.003414	-0.000016	0.003317	0.032344

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.002355	0.001539	1.53	0.126
prix_debut	0.997994	0.001285	776.63	<2e-16 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.005788 on 2071 degrees of freedom  
Multiple R-squared: 0.9966, Adjusted R-squared: 0.9966  
F-statistic: 6.032e+05 on 1 and 2071 DF, p-value: < 2.2e-16



# Interprétation des résultats

```
Call:
lm(formula = prix_fin ~ prix_debut, data = donnees5.subset1)

Residuals:
    Min       1Q   Median       3Q      Max
-0.028232 -0.003414 -0.000016  0.003317  0.032344

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.002355   0.001539    1.53   0.126
prix_debut   0.997994   0.001285  776.63 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Coefficient de détermination = 0,9966  
F = 6,032e+05

Residual standard error: 0.005788 on 2071 degrees of freedom  
Multiple R-squared: 0.9966, Adjusted R-squared: 0.9966  
F-statistic: 6.032e+05 on 1 and 2071 DF, p-value: < 2.2e-16



# Résultat

- Prix de la fin journée = 0,997994 x Prix début journée + 0,002355



## **Autres méthodes de prédiction**



## Concept

- Cette méthode consiste qu'après chaque  $x$  jours consécutives en une seule direction , on assume que le jour suivant va prendre une direction inverse
- Direction haussière  $\rightarrow \text{prix\_fin} > \text{prix\_debut}$
- Direction baissière  $\rightarrow \text{prix\_fin} < \text{prix\_debut}$





# Méthode 1

- Création d'une nouvelle colonne :
- Direction haussière = 1
- Direction baissière = 0

```
#savoir les directions :  
donnees$direction <- 1  
donnees$direction <- ifelse(donnees$prix_debut > donnees$prix_fin,0,donnees$direction)
```



# Méthode 1

- Création d'une nouvelle colonne pour compter les jours consécutives en même direction :

```
#compter les jours consecutives ayant une seule direction :
donnees$compteur<-1
for (i in 2:nrow(donnees))
{
  if(donnees[i,]$direction == donnees[i-1,]$direction)
  {
    donnees[i,]$compteur<-donnees[i-1,]$compteur+1
  }
}
```



# Méthode 1

- Création d'une nouvelle colonne contenant le direction du jour suivant (observation)

```
#savoir la direction du jour suivant:  
shift<-function(x,n)  
{  
  c(x[-(seq(n))],rep(NA,n))  
}  
donnees$direction_suiv<-shift(donnees$compteur,1)
```

# Méthode 1

3 jours de direction haussière consécutifs

```
> head(donnees[c(2,5,7:9)])
  prix_debut prix_fin direction compteur direction_suiv
1    1.30294  1.30780        1         1             2
2    1.30781  1.31735        1         2             3
3    1.31783  1.32366        1         3             1
4    1.32364  1.32237        0         1             2
5    1.32236  1.32118        0         2             3
6    1.32120  1.31978        0         3             1
```

Le marché va changé la direction le jour suivant

3 jours de direction baissière consécutifs




# Méthode 1

- Ensuite on va calculer les probabilités de continuation/changement de direction :

```
#tableau des evenements :  
table<-table(donnees$compteur,donnees$direction_suiv)  
table  
#probabilites des evenements:  
table.pourcentage<-prop.table(table,1)  
#table de probabilite:  
print.table(local({table.pourcentage[table.pourcentage==0]<-NA;table.pourcentage})))
```

# Méthode 1



	1	2	3	4	5	6	7	8	9
1	0.5092421	0.4907579							
2	0.5084746		0.4915254						
3	0.5363985			0.4636015					
4	0.5916667				0.4083333				
5	0.5918367					0.4081633			
6	0.8000000						0.2000000		
7	0.2500000							0.7500000	
8	0.6666667								0.3333333

Après 6 jours consécutifs en même direction on a une probabilité de 80% pour que le jour suivant va prendre la direction inverse et 20% pour qu'il va continuer en même direction le 7<sup>ème</sup> jour





# Conclusion méthode 1

- Cette méthode simple peut être combinée avec les méthodes de prédiction qu'on a déjà étudié :
- Par exemple on a une marge d'erreur de  $\pm 0,002$  en KNN , on peut réduire cette marge sachant que le prix a une forte probabilité de changer la direction le jour suivant
- On peut aussi éliminer quelques fausses prédictions (prédiction haussière par KNN et prédiction baissière par cette méthode) et compter seulement les prédictions en même direction par ces 2 méthodes pour augmenter et renforcer notre probabilité



# Conclusion méthode 1

- Exemple temps réel :
- L'heure et 00:00 ,sachant le prix du début de la journée à 00:00 est 1,3000 ,on peut prédire le prix de la fin de cette journée par KNN ,supposons qu'il est 1,5000, donc on a prédit que ce jour à une direction haussière puisque  $1,3 < 1,5$  , donc on a décidé d'acheter maintenant à 1,3 et vendre en fin de la journée à 1,5  $\rightarrow$  un profit de 0,2 par chaque unité acheté
- Supposons qu'on est déjà en 7<sup>ème</sup> jour consécutif en direction haussière donc on a une probabilité de 80% que ce jour là va prendre une direction baissière
- $\rightarrow$ Donc on a 2 prédictions opposés  $\rightarrow$  on évite d'acheter ce jour

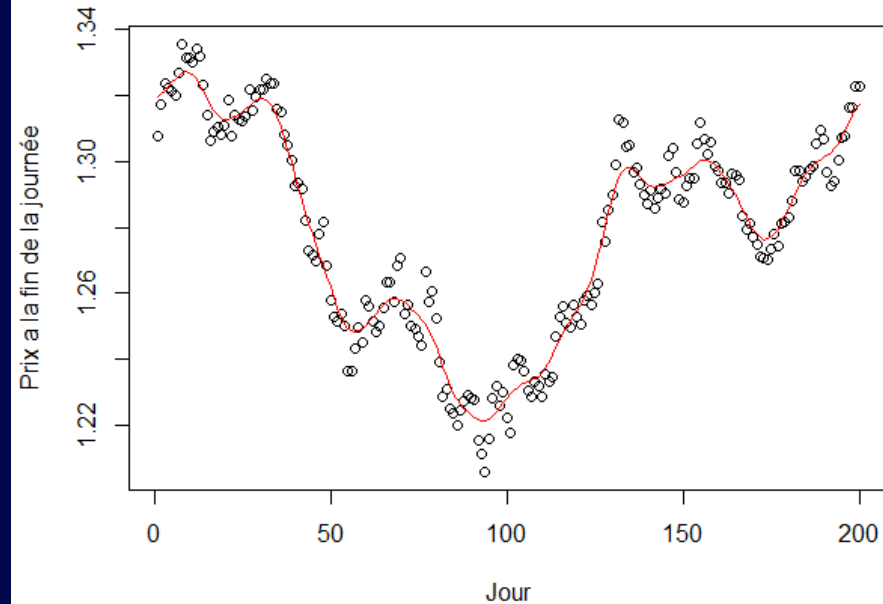


## Méthode 2

- Cette méthode est plus complexe que la première
- On utilise ce qu'on appelle une moyenne mobile et on peut prédire le prix de la fin du jour seulement par savoir  $X$  prix de fin antérieurs par le calcul de la moyenne de  $X$  prix\_fin

## Méthode 2

- Le ligne rouge représente la moyenne mobile
- On peut prédire le prix de la fin de la journée en observant la tendance du moyenne mobile





## Méthode 2

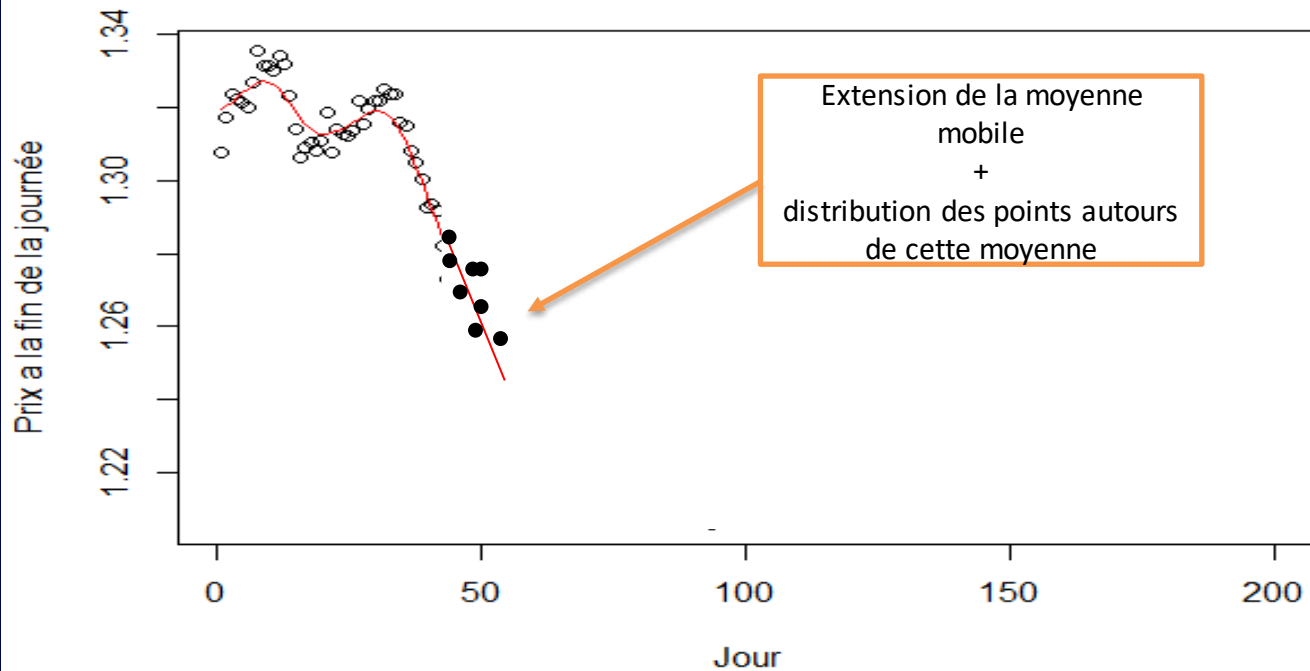
- On a de méthode de prédiction :
- 1- si la direction de la moyenne mobile et haussière → les prix de fin des jours de futurs vont continuer à augmenter jusqu'au changement de direction de la moyenne mobile
- 2- on calcul la moyenne des distances entre les points et la ligne et si un point a une distance beaucoup plus grandes de la moyenne (très loin de la ligne), on dit que le/les points suivant vont retourner vers cette moyenne



# Evaluation

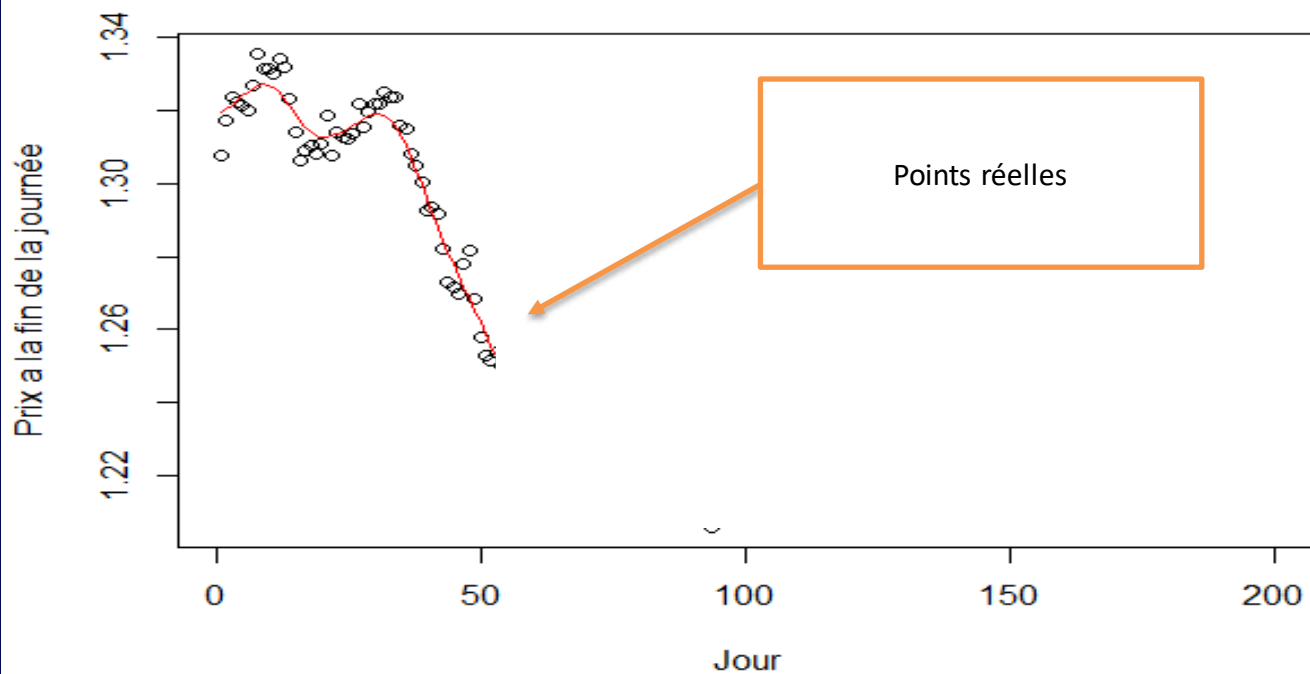
- Ces méthodes sont plus subjectives et plus complexe à évaluer , donc on va utiliser le graphes pour les évaluer

# Evaluation

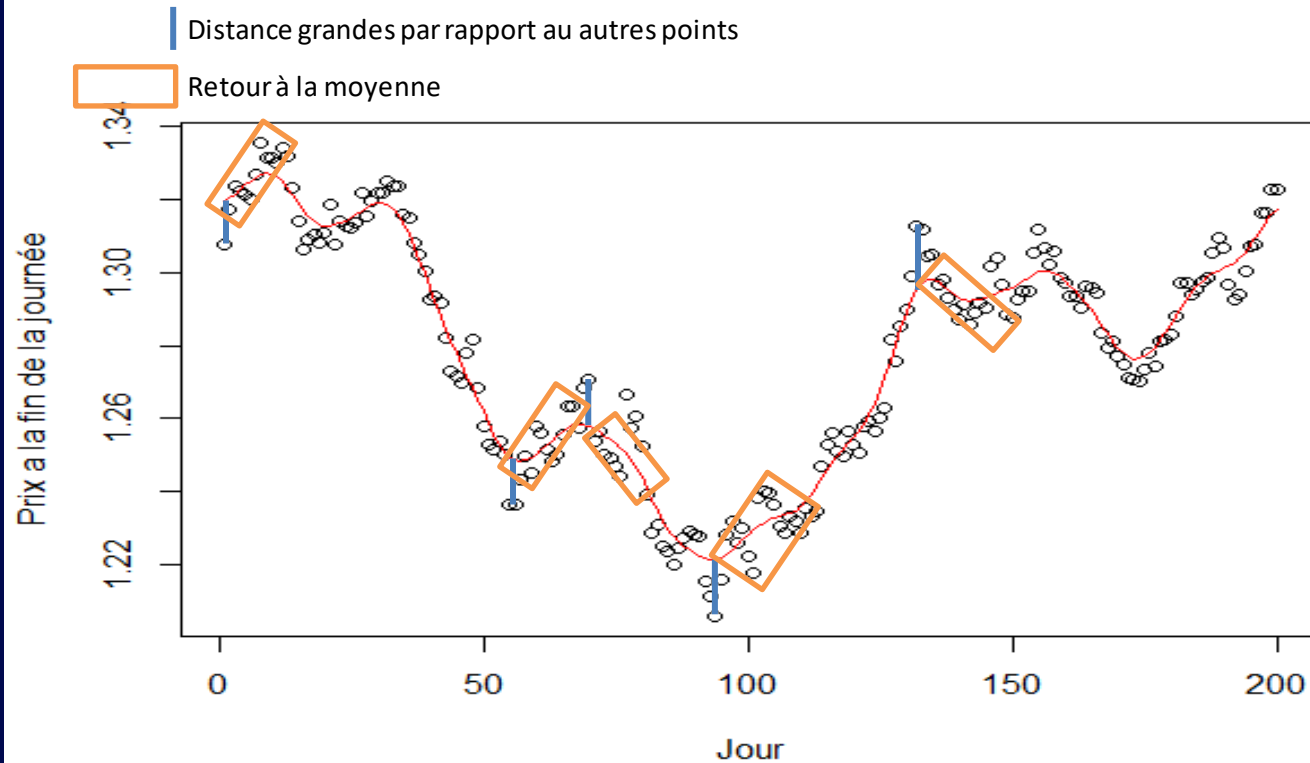




# Evaluation



# Evaluation





# Conclusion

- Machine Learning est un domaine très intéressant utilisé par les grandes banques ,plus que 70% des décisions d'achat/vente dans les bourses est faites par des algorithmes développés en utilisant le concept du Machine Learning

Sources :

- <https://www.tradingview.com/symbols/EURUSD/>
- <https://seekingalpha.com/article/4230982-algo-trading-dominates-80-of-stock-market>