



Risk - Less

Rapport Final

PPE-1765-RiskLess

11 Avril 2018

PPE n°1765

BOUAULT Edouard
CHAURAND Benoit
DUCHENE Remy
HAMEL Dorian
MOLANO Luis Carlos
OIKNIN Naomi

Sommaire

I. ABSTRACT	3
1) English	3
2) Français	3
II. PRESENTATION DU PROJET	4
1) Glossaire	4
2) Equipe	5
3) Contexte	5
a. Présentation du sujet	5
b. Etat de l'art	6
4) Objectifs	7
5) Valorisation Innovation Ouverte	8
III. RESULTATS OBTENUS	10
1) Interface	10
2) Base de données	17
3) Algorithmes	18
a. Value at Risk	18
b. Machine Learning	18
c. Volatilité	22
IV. DIFFICULTES RENCONTREES	22
V. PERSPECTIVES D'AVENIR	24
VI. ANNEXE	25

I. Abstract

1) English

67% of novice investors inform themselves before placing a stock market order. The more an investor is informed by the tools at his disposal, the more he minimizes his risk of losing money. The Risk-Less PPE is a project to develop an investment support software.

The goal of the Risk-Less project is to create an interface that will advise its users whether or not to invest. The challenge is to make every effort to create a viable solution with relevant algorithms and a functional software.

To meet this challenge, the project was carried out in several parts. First, we developed the algorithms. For this, we conducted a study to find out which indicators might be of interest to novice investors. It is for this reason that our project brings together basic machine learning algorithms namely linear regression and Random Forest, but also widely used indicators in finance, and therefore easy to study namely volatility and Value at Risk. Secondly, a database and an interface were created. The necessary data was collected for several stocks, namely Peugeot, Airbus and LVMH. To use Risk-Less, you just have to choose the indicator of your choice and then click on the name of the stock that interests you.

2) Français

67 % des investisseurs novices s'informent eux même avant de passer un ordre de bourse. Plus un investisseur s'informe avec les outils à sa disposition et plus il minimise son risque de perdre de l'argent. Le PPE Risk-Less est un projet de développement d'un logiciel d'aide à l'investissement.

L'objectif du projet Risk-Less est de créer une interface qui conseillera à son utilisateur s'il doit ou non investir. L'enjeu est donc de tout mettre en œuvre pour créer une solution viable dotée d'algorithmes pertinents et un logiciel fonctionnel.

Pour répondre à ce challenge, le projet a été réalisée en plusieurs parties. Tout d'abord, nous avons développé les algorithmes. Pour cela, nous avons effectué une étude pour savoir quels indicateurs pourraient intéresser les investisseurs novices. C'est pour cette raison que notre projet regroupe des algorithmes de base en machine learning à savoir la régression linéaire et le Random Forest, mais également des indicateurs très répandus en finance et donc faciles à étudier à savoir la volatilité et la Value At Risk. Dans un deuxième temps une base de données et une interface ont été créées. Les données nécessaires ont été récoltées pour plusieurs actions à savoir Peugeot, Airbus et LVMH. Pour utiliser Risk-Less, il suffit donc de choisir l'indicateur de votre choix puis de cliquer sur le nom de l'action qui vous intéresse.

II. Présentation du Projet

1) Glossaire

Espérance mathématique	Moyenne d'une variable pondérée par sa probabilité d'occurrence. Elle mesure la rentabilité espérée.
Machine Learning	Processus d'apprentissage automatique.
Sous- jacent	Actif financier sur lequel porte un produit dérivé.
ValueAtRisk (VaR)	Pire perte attendue sur un horizon de temps donné pour un certain niveau de confiance (compris entre 0 et 1).
Volatilité	Mesure des amplitudes des variations du cours d'un actif financier. Plus la volatilité d'un actif est élevée et plus l'investissement dans cet actif sera considéré comme risqué car le risque de perte sera important.

2) Equipe



Luis Carlos Molano	Naomi Oiknin	Rémy Duchene	Benoit Chaurand	Dorian Hamel	Edouard Bouault
06 98 13 37 09 luis-carlos.molano@edu.ece.fr	06 61 59 89 66 naomi.oiknin@edu.ece.fr	07 70 61 33 02 remy.duchene@edu.ece.fr	06 43 94 84 12 benoit.chaurand@edu.ece.fr	07 70 28 89 91 borian.hamel@edu.ece.fr	06 66 87 28 65 Eb151539@edu.ece.fr
Systèmes d'information	Ingénierie Financière	Ingénierie Financière	Systèmes d'information	Ingénierie Financière	Systèmes d'information
Chef de projet Implémentation de l'algorithme en python et partie graphique du logiciel	Design, organisation des tâches et développement des algorithmes en Machine Learning	Analyse du risque financier et Responsable Machine Learning, ValueAtRisk	Développement en Python	Analyse du risque financier ,Calcul stochastique et Machine Learning	Développement Python

3) Contexte

a. Présentation du sujet

De manière générale les gens ont peur de prendre des risques et cela est d'autant plus vrai quand de l'argent est en jeu. Ainsi, les gens privilégient la sécurité. Ce qui va à l'encontre de l'investissement en Bourse.

De nombreux jeunes notamment s'intéressent de plus en plus à la bourse car ils y voient un moyen de gagner un peu d'argent en investissant. Cependant il n'est pas toujours aisé de s'en sortir avec la multitude d'informations circulant sur internet.

En matière boursière, la question n'étant pas de savoir s'il faut ou non prendre des risques, notre logiciel permettrait de donner une idée de combien investir.


De manière générale une personne n'y connaissant strictement rien à la bourse commencera tout d'abord par lire des livres et consulter des forums sur le sujet afin de se documenter. Puis elle ouvrira éventuellement un portefeuille virtuel (exemple : Boursorama) qu'elle essayera de faire marcher. Puis, si elle arrive à gagner de l'argent virtuel au bout de quelques mois elle passera ou non à l'étape suivante qui consiste à investir de l'argent réel.

C'est pourquoi nous avons pensé à développer un logiciel qui permettrait aux investisseurs novices (ayant déjà passés les étapes évoquées ci-dessus) d'avoir à disposition un outil supplémentaire par rapport à ceux déjà existants qui leur permettrait d'améliorer leur analyse de la Bourse et réduirait ainsi le risque pris lors de potentiels investissements.

Nous ne prétendons toutefois pas rendre le risque d'investir nul ce qui est impossible.

b. Etat de l'art

	<p>IsoBourse est un puissant logiciel d'aide à la décision boursière qui permet la détection d'opportunités d'investissements à partir d'un modèle mathématique : La Pression IsoBourse</p>
	<p>Aldexia est un logiciel conçu pour aider à détecter de bonnes opportunités d'investissements en bourse.</p> <p>L'Analyse Technique étudie les courbes des actions au fil des jours. Ces dernières sont constituées par les valeurs prises dans le temps par chaque action, faisant des points sur un graphique que l'on relie simplement par des traits.</p>
	<p>WalMaster XE est un logiciel mettant à disposition du particulier les outils et assistance nécessaire à une bonne gestion de ses investissements. Adressé aux débutants comme aux confirmés, le logiciel s'adresse à tout type de profil et ce quel que soit le type de produits boursiers pratiqués : actions, produits dérivés...</p>
	<p>Xl bourse a pour objectif d'aider les investisseurs à déterminer les tendances en cours des principales actions françaises grâce à l'analyse technique. Les informations diffusées sur Xlbourse.fr constituent une aide à la décision pour les investisseurs.</p>
	<p>Botraiders.com est leader des prévisions boursières par intelligence artificielle depuis 2007. C'est un site de bourse avec une approche 100 % mathématiques, qui met à disposition des informations boursières permettant ainsi aux investisseurs de mieux investir en Bourse.</p> <p>Ce site utilise des robots de trading qui analysent les actions de la Bourse de Paris.</p>

	<p>Qtstalker est un logiciel d'analyse technique libre, 100% gratuit et facile à utiliser pour les systèmes à la norme POSIX. De plus il est distribué sous les termes de la licence publique GNU GPL et une communauté de développement active ajoute souvent de nouvelles fonctionnalités.</p>
---	--

Les divers logiciels présentés ci-dessus témoignent de l'attrait des investisseurs pour ces outils permettant d'affiner leurs analyses. La différence fondamentale de notre logiciel par rapport à ceux évoqués est qu'il sera destiné aux investisseurs novices en premier lieu et qu'il sera possible d'accéder au code source afin que d'éventuels utilisateurs expérimentés puissent l'intégrer à d'autres logiciels existants pour parfaire leur analyse.

4) Objectifs

Les critères d'acceptation que nous avons retenus lors de la rédaction du cahier des charges étaient que le logiciel soit en mesure de :

- Pouvoir recenser une base de données d'au minimum 2 actions afin de lancer nos algorithmes sur ces actions
- Mettre à disposition des utilisateurs un indicateur de prix et de risque pour certaines actions
- Calculer l'espérance du prix futur d'une action pour une période fixée adaptée
- Donner une indication aux investisseurs selon un risque moindre (<5%), c'est-à-dire que nos algorithmes devront fournir un résultat avec un intervalle de confiance de 95%.
- Proposer ces services de manière libre et gratuite (open source)

Par ailleurs, nous devons être en mesure de mettre à disposition un forum pour nos utilisateurs. Le système devait être facile à prendre en main, et intuitif d'utilisation.

A la suite du dernier sprint PPE du mois de mars nos objectifs étaient les suivants:

- Trouver davantage de données pour remplir notre base de données et améliorer l'entraînement de nos algorithmes.
- Joindre les bases de données à l'algorithme de VaR afin de rendre la fonctionnalité opérationnelle.
- Récupérer des informations qualitatives notamment en termes d'attentes de potentiels utilisateurs de l'application (mail/phoning éventuellement avec leur accord)
- Améliorer notre interface graphique
- Continuer les phases de tests afin d'avoir une application présentable à la soutenance d'avril

Finalement nous sommes plutôt satisfaits du résultat obtenu. En effet nous pensons avoir respecté les différents objectifs que nous avons fixés lors du lancement du projet. Notre application finale est assez intuitive et simple à prendre en main. Elle permet d'exécuter

différents algorithmes sur des données de trois entreprises différentes à savoir Airbus, LVMH et Peugeot.

Tout d'abord la régression linéaire, puis Ridge puis random Forest. Par ailleurs nous permettons un calcul de la volatilité, du rendement des prix des actions de ces 3 mêmes entreprises ainsi qu'un calcul de la Value at Risk. L'exécution de chaque algorithme affiche à l'utilisateur une indication des différentes informations utiles à sa prise de décision. Par ailleurs les différentes phases de tests ainsi que les résultats obtenus avec chaque algorithme sont proches et cohérentes.

Enfin, l'accès au code source se fait via un lien GitHub situé dans la rubrique Téléchargement de notre forum à l'adresse :

<http://risk-less.forumactif.com/t3-lien-github-pour-telecharger-le-logiciel-risk-less>

5) Valorisation Innovation Ouverte

Pourquoi valoriser en innovation ouverte ?

Nous avons opté pour l'axe de valorisation Open Source.

L'Open Source, en tant que méthode de conception d'un logiciel, permet l'émergence de produits compétitifs sans les besoins en capitaux et main-d'œuvre inhérents aux méthodes traditionnelles de développement des logiciels propriétaires.

Les projets open source sont ouverts à tous dans le sens où n'importe qui peut suggérer des changements dans le code source. Mais ces changements peuvent toujours être rejetés ou annulés.

Notre projet s'inscrivant dans ce domaine et visant à aiguiller des investisseurs novices dans leurs investissements, il nous a paru judicieux de choisir cette valorisation pour notre projet. En effet, le but étant que les futurs utilisateurs puissent remplir notre base de données afin d'avoir plus de données pour entraîner nos différents algorithmes de machine Learning et ainsi avoir des résultats plus représentatifs de la réalité.

Le choix de l'open source s'avérait selon nous être un moyen efficace d'atteindre cet objectif. En effet, l'accès au code source permet à plus de gens de tester notre code et de remonter tout type de faiblesse/bogue de l'application : « Plusieurs têtes valent mieux qu'une ».

Le but principal du projet est donc de donner aux utilisateurs la possibilité d'exécuter le programme, de l'étudier, de l'adapter et de le redistribuer. Les valeurs de l'open source sont en totale adéquation avec ce que nous avons pour but de consolider à travers notre projet à savoir le partage, la coopération et l'entraide. En effet, un logiciel n'existe que par l'utilisation qu'en font ses utilisateurs.

De plus, l'open source est l'avenir de l'innovation et réaliser un projet dans ce cadre nous paraît être une formidable opportunité dans notre cursus. En effet l'ouverture est une des conditions pour innover le mieux et le plus rapidement, ce qui est au cœur du métier d'ingénieur.

Quelle valeur ajoutée est apportée par le projet ?

Le logiciel dispose de divers atouts contribuant à sa valeur ajoutée. En particulier, le langage utilisé à savoir « Python » qui est un des plus répandus et plus utilisé. Ainsi, la contribution au projet par les utilisateurs s'en voit facilitée.

Par ailleurs, il utilise des algorithmes de base en machine Learning à savoir la régression linéaire et le Random Forest. Également des indicateurs très répandus en finance et donc facile à étudier à savoir la volatilité et la Value At Risk.

Ainsi on se place réellement du côté d'un novice et on offre l'opportunité à un public large de réadapter le logiciel. Ce qui a contrario est rarement faisable avec les logiciels existants utilisant des notions trop avancées pour des novices.

Pourquoi créer un projet en Innovation ouverte ?

Les plus grandes entreprises effectuent des projets Open Source. Nous avons choisi de créer un projet en Innovation Ouverte de par sa tendance forte dans le monde professionnel et également car cette valorisation correspondait le mieux au besoin auquel nous souhaitions répondre avec notre solution.

Un projet en Innovation Ouverte est en particulier utile dans le sens où il peut s'appuyer sur une population test (les utilisateurs).

Une des difficultés majeures qui nous a suivi tout au long du projet est notamment que pour exploiter pleinement cet avantage de l'Open Source, il faut avoir un logiciel un minimum fonctionnel. Cependant il nous a fallu du temps pour aboutir à ce que nous avons à l'heure actuelle.

Quel type de distribution pour le code et de diffusion de la connaissance générée par votre projet ?

En ce qui concerne la distribution du code, l'utilisateur le télécharge sur GitHub via un lien disponible dans une section dédiée sur notre forum (rubrique Téléchargement) à l'adresse : <http://risk-less.forumactif.com/>

L'utilisateur peut finalement accéder au code source et utiliser notre logiciel à sa guise. Pour ce qui est de la contribution au code, l'utilisateur peut faire des propositions de modifications et il nous revient de les inclure au code source si ces dernières nous paraissent pertinentes et utiles.

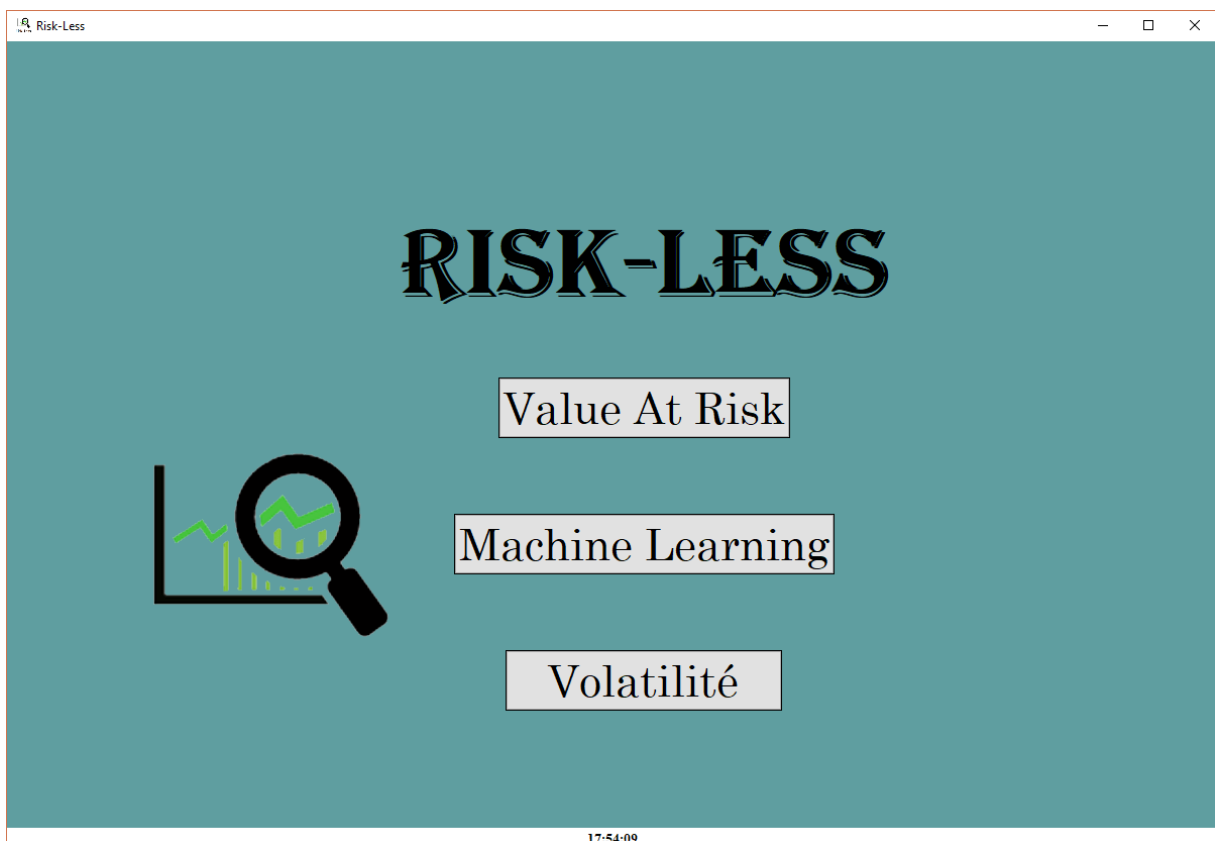
Résultats finaux

III. Résultats obtenus

1) Interface

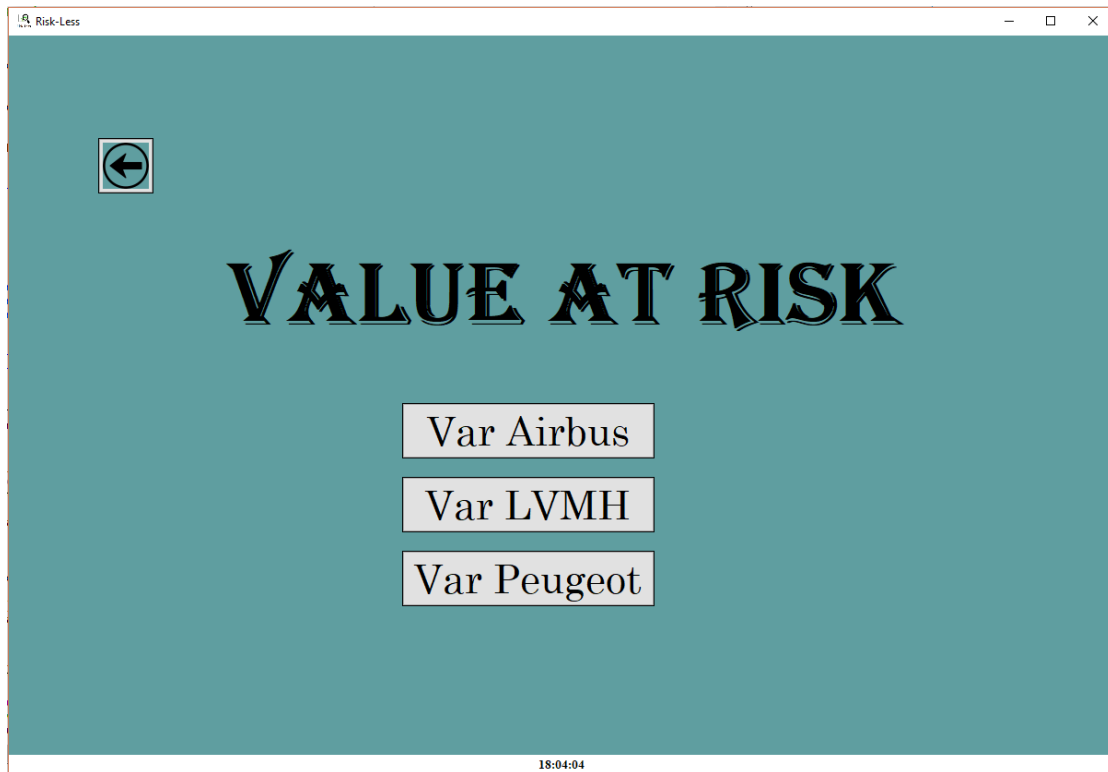
L'interface se doit d'être intuitive et facile d'utilisation. De ce fait nous avons choisis de ne pas surcharger la page et de rester simple. Pour réaliser l'interface nous avons utilisé la bibliothèque Tkinter.

Sur la page d'accueil nous affichons le nom du logiciel en gros, le logo, et trois boutons permettant d'aller sur les 3 différentes grandes parties à savoir : La Value at Risk, le Machine Learning et la volatilité.



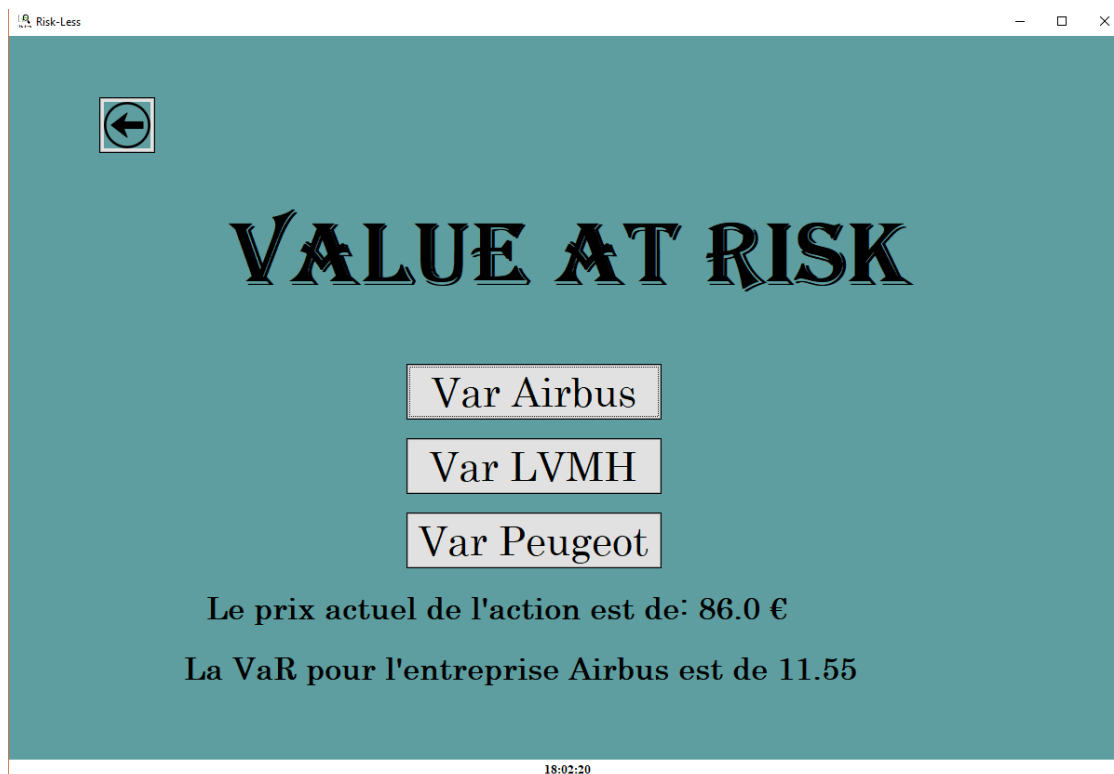
Page d'accueil du logiciel

En allant sur la page de la Value At Risk, nous avons la possibilité de choisir pour quelle entreprise nous voulons la calculer. Nous avons aussi la possibilité de revenir à la page d'accueil en cliquant sur la flèche retour en haut à gauche. Le titre de la section prend la place du nom du logiciel pour tenir au courant l'utilisateur dans quelle partie il se trouve.

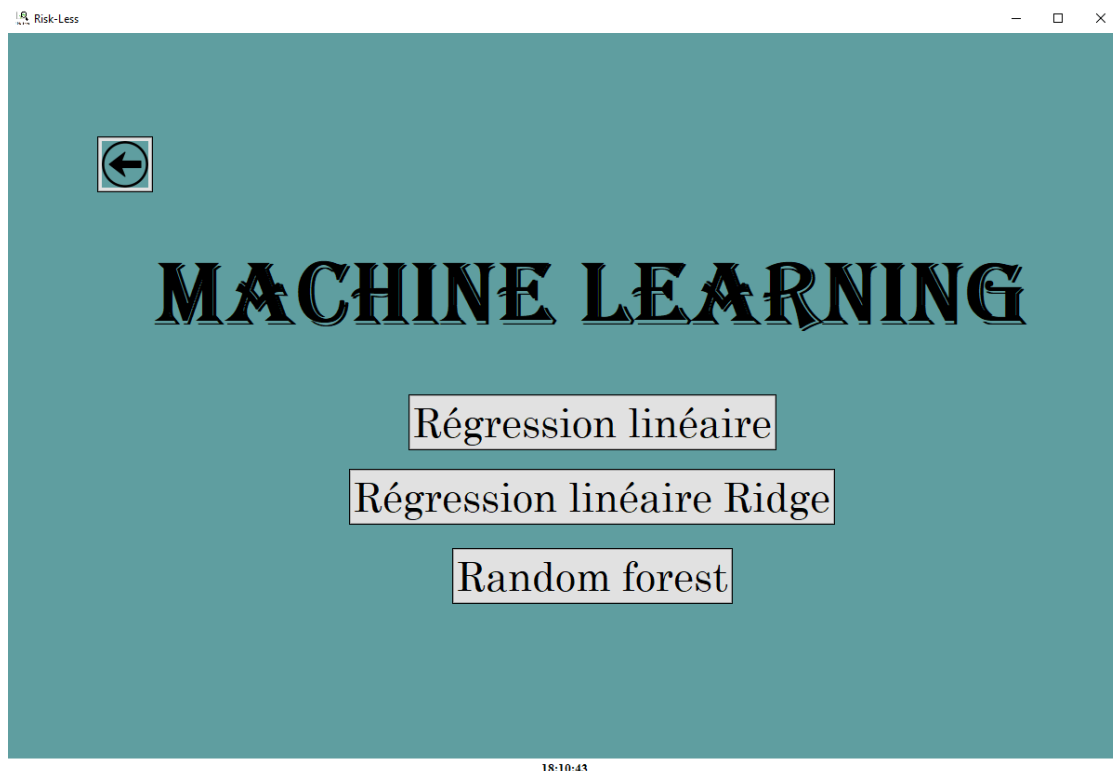


Page de la Value At Risk

En choisissant l'entreprise sur laquelle calculer la Value At Risk, nous obtenons la valeur de celle-ci. Nous affichons également le prix de l'action actuel pour que l'utilisateur se rende mieux compte de cette valeur. En effet VaR de 10 pour une action à 50€ n'a pas le même impact que pour une action à 200€. Pour revenir à l'accueil l'utilisateur peut utiliser la flèche.

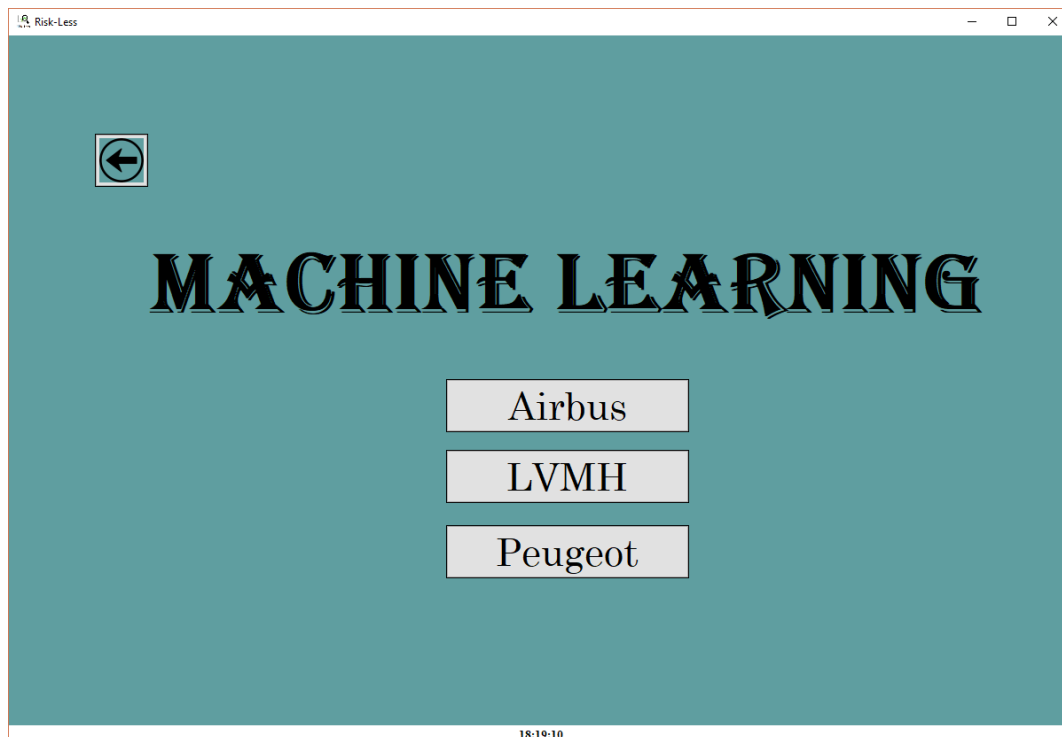


La page suivante est celle du Machine Learning, l'utilisateur a la possibilité de choisir entre trois différents algorithmes : Régression linéaire, régression linéaire Ridge et le random Forest.



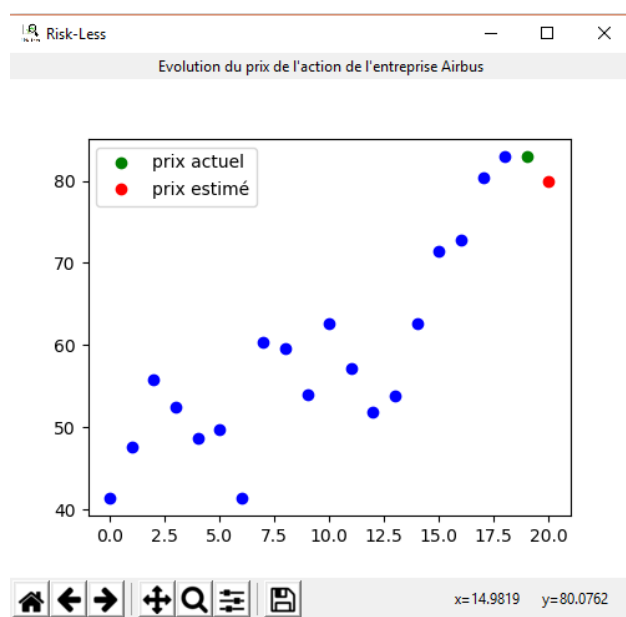
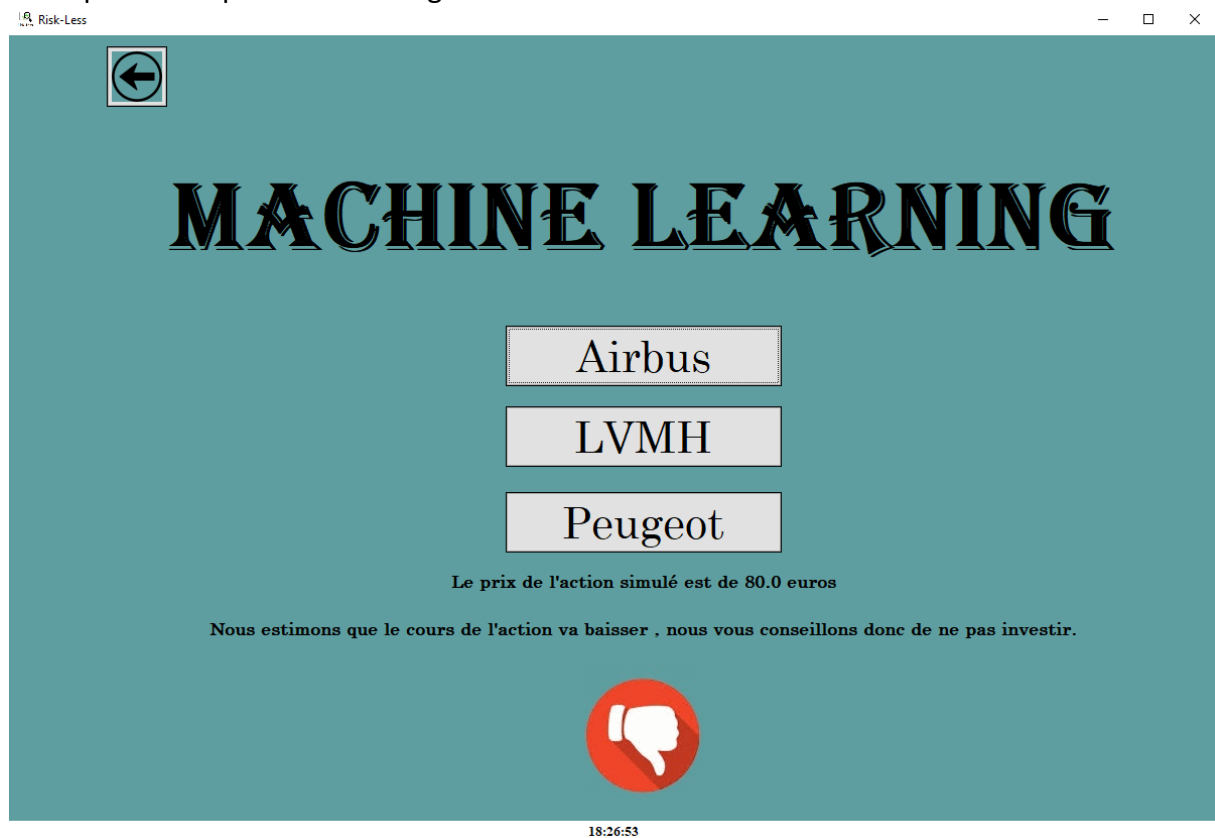
Page Algos Machine Learning

Après avoir fait son choix, il peut de nouveau choisir sur quelle entreprise l'exécuter.



Page Régression Linéaire

L'algorithme renvoie le prix de l'action simulé, le compare avec le prix actuel et conseille l'utilisateur d'investir ou de ne pas investir en fonction du résultat. Pour les algorithmes de Ridge et de Random Forest, nous affichons également le rendement, le gain et le ratio gain/risque. Pour faciliter la visualisation du résultat, un pouce vers le haut de couleur verte s'affiche si le logiciel conseille d'investir, et un pouce rouge vers le bas s'il ne le conseille pas. Le logiciel affiche également un graphe, permettant de visualiser les anciennes valeurs de l'action représentées par des ronds bleus et la valeur actuelle par un rond vert. Enfin, la valeur prédite par l'algorithme est représentée par un rond rouge.



Graphique des prix de l'action de l'entreprise Airbus avec la prévision en rouge

← Risk-Less


←

RIDGE

Airbus

LVMH

Peugeot



Prix futur : 254.25€
Rendement : 1.54%
Gain : 3.85€ / action
Ratio Gain/Risque : 0.28

18:31:27

Page Ridge

← Risk-Less

←

RANDOM FOREST

Airbus

LVMH

Peugeot

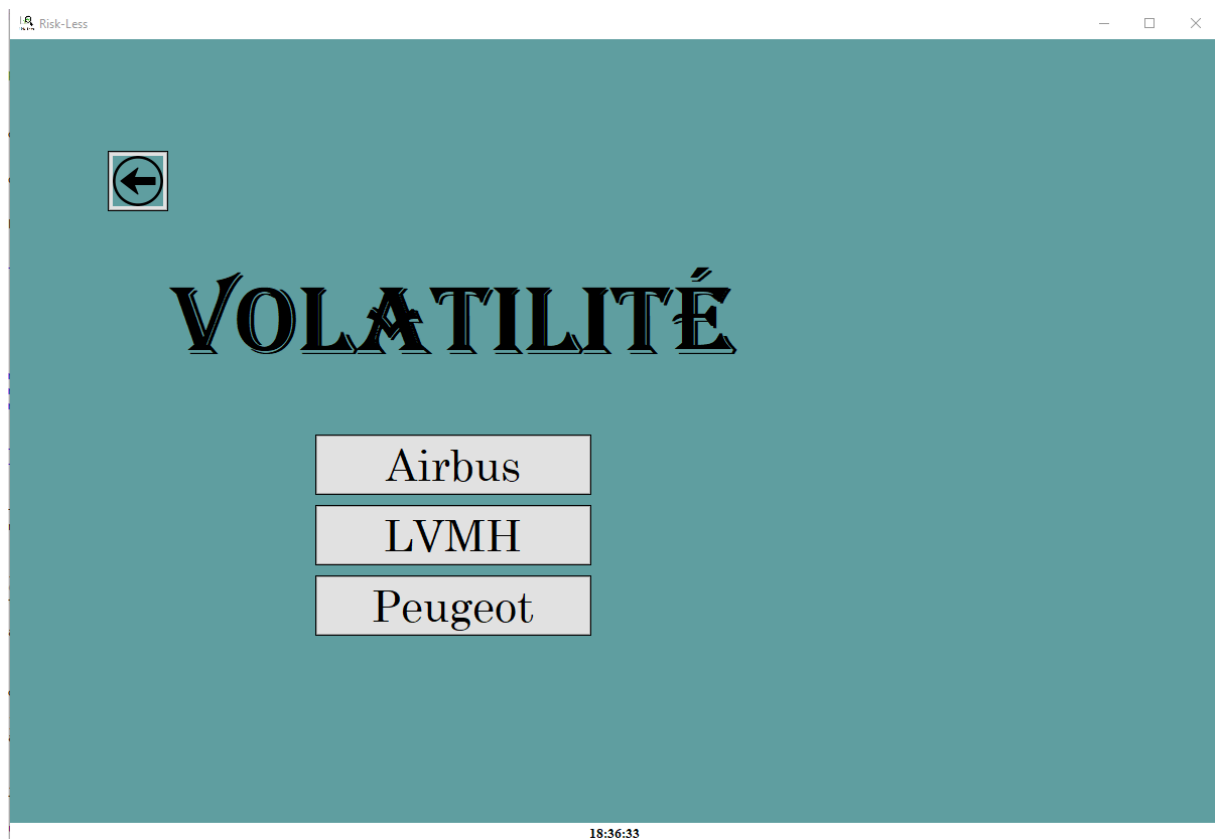


Prix futur : 17.08€
Rendement : 0.77%
Gain : 0.13€ / action
Ratio Gain/Risque : 0.32

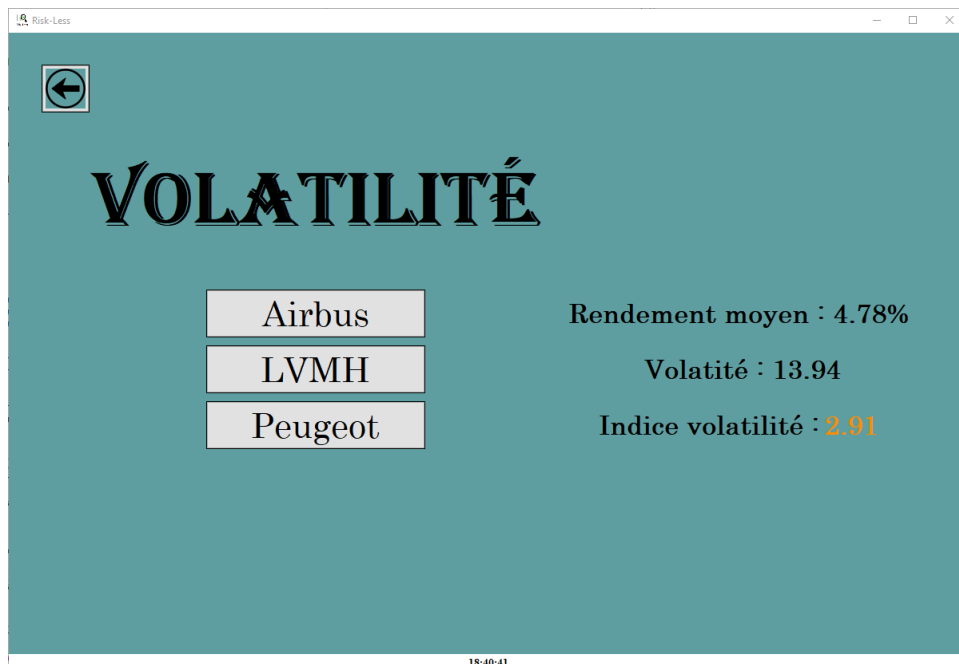
18:37:45

Page Random Forest

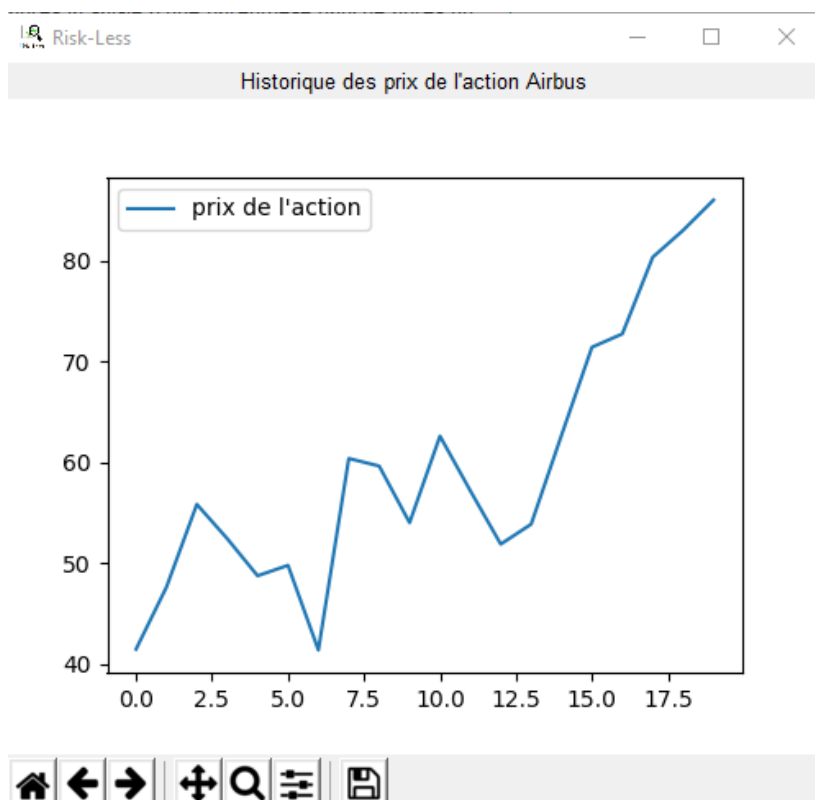
La dernière partie du logiciel est la fonctionnalité correspondante au calcul de la volatilité. Dans cette page, comme pour les deux autres l'utilisateur peut choisir sur quelle entreprise la calculer.



Après avoir choisi l'entreprise, le logiciel affiche le rendement moyen, la volatilité et l'indice de volatilité. Nous obtenons également un graphe affichant l'évolution du prix de l'action.



Page Volatilité



2) Base de données

Pour ce qui est de la base de données, nous avons utilisé la bibliothèque SQLite car elle offre la possibilité d'être directement intégrée au programme au contraire des bases de données traditionnelles du type MySQL ou PostgreSQL qui nécessitent un schéma client-serveur.

Un gros avantage également de cette bibliothèque est que l'on peut l'utiliser sans restriction dans notre projet car elle est dans le domaine public.

De plus, elle est assez facile à prendre en main. Ainsi elle facilite la contribution au code de nos utilisateurs.

Notre base de données regroupe les données passées des actions d'Airbus et de Peugeot. Ces dernières nous servent d'entrée à notre algorithme de Machine Learning. Ainsi l'utilisateur n'a pas à rentrer lui-même les valeurs passées car celles-ci seront automatiquement chargées depuis notre base de données. Pour ce faire nous utilisons deux tables.

La première "entreeX" qui contient les attributs nom_action, acid, benefice, actifs, dividendes et prix_recent dont la clé primaire est composée des attributs nom_action et acid qui est l'identifiant de l'action.

Puis une deuxième table "sortieY" (valeur réelle de l'action) qui contient les attributs nom_action, acid et prix_reel dont la clé primaire est une clé composée de deux clés étrangères nom_action et acid.

En Annexe vous trouverez un aperçu des données de l'entreprise Airbus stockées dans notre base de données.

Nous avons utilisé les valeurs de notre base de données afin d'exécuter notre algorithme de Machine Learning qui a été intégralement implémenté selon le modèle que nous avons proposé dans le Cahier des Charges.

Il se décompose en quatre fonctions qui ,dans l'ordre, calcule les coefficients Téta optimaux afin de réduire au maximum l'erreur de notre régression linéaire (entraînement), affiche l'erreur quadratique entre les valeurs estimées de notre algorithme et les valeurs réelles afin de juger de la précision de notre algorithme (test) , affiche les valeurs prévisionnelles en fonction des valeurs actuelles et enfin affiche les valeurs passées ,actuelles et prévisionnelles du cours de la bourse sur un même graphe afin de mieux visualiser sa variation et conseiller l'utilisateur sur le choix d'investir ou non sur de cette action.

Cette présentation est plus agréable pour un public amateur afin de leur permettre de mieux apprécier les résultats de notre algorithme.

3) Algorithmes

a. Value at Risk

L'algorithme de value at risk a été codé en python. Il a été difficile de trouver de la documentation sur la value at risk avec une simulation de Monte-Carlo. La première étape a été de "créer une action" avec des valeurs test qui allaient être ensuite remplacées par les vraies valeurs de la base de données.

La Value at Risk a la formule suivante :

$$\text{Prix simulé} = \text{Prix actuel} * \text{exponentiel}(\text{rendement} - 0.5 * \text{volatilité}^2) + \text{volatilité} * \text{processus de Wiener} * \sqrt{\text{échéance}}.$$

En général les calculs de value at risk en simulation de Monte Carlo sont effectués sur excel. Il est facile sur excel de générer le processus de Wiener car une fonction intrinsèque à Excel le fait directement. La difficulté a été de générer le processus de Wiener. Pour cela un nombre aléatoire entre 0 et 1 est généré. L'étape suivante est de chercher l'image de ce nombre aléatoire dans la fonction normale inverse. Pour cela nous avons utilisé la fonction suivante : norm.ppf (nombre aléatoire).

Une fois cette le processus de Wiener généré, il suffit d'appliquer la formule ci-dessus en remplaçant les différents termes par les valeurs test mentionnées plus haut.

Enfin la dernière étape a été de relier cet algorithme avec la base de données afin qu'il calcule la value at risk de vraies actions. En effet la value at risk est calculée à partir de la volatilité (calculée par un autre algorithme de Risk-Less) mais aussi à partir des informations sur les actions présentes dans la base de données.

b. Machine Learning

En Machine Learning, on oppose très fréquemment apprentissage supervisé et apprentissage non supervisé.

Bien que les deux types d'apprentissages relèvent de l'intelligence artificielle, l'apprentissage supervisé signifie que l'algorithme est guidé sur la voie de l'apprentissage en ayant connaissance des résultats attendus (output). L'algorithme apprend alors de chaque exemple, avec pour but, d'être capable de généraliser son apprentissage à de nouveaux cas. Dans ce type d'apprentissage il faut donc uniquement chercher les paramètres optimaux.

Au contraire, dans le cas de l'apprentissage non supervisé, l'apprentissage par la machine se fait de façon totalement autonome c'est-à-dire que les données sont communiquées à la machine

sans lui fournir les exemples de résultats attendus en sortie. Dans ce type d'apprentissage il faut chercher les paramètres + l'output (la sortie).

Notre projet utilise des algorithmes d'apprentissages supervisés que nous évoquerons ci-après.

Prévision du cours d'une action avec une Régression linéaire

Nous avons conçu un modèle de Machine Learning pouvant prédire le cours d'une action en fonction des données passées. Pour ce faire nous devons commencer par définir les facteurs influant sur le cours d'une action. Ces facteurs seront alors les entrées X de notre système. La sortie Y de notre système est le cours de l'action. A noter que nous avons intégralement codé la régression linéaire (from scratch).

i. Bases du modèle

Facteurs influant sur le cours d'une action :

X1 – Bénéfice de la société

X2- Actifs de l'entreprise

X3- Dividendes

X4 – Prix le plus récent de l'action

Ces facteurs sont susceptibles de changer au cours du projet au profit de facteurs que nous pourrions trouver plus pertinents par la suite. A ce stade du projet ces facteurs nous semblent assez pertinents et représentatifs de la réalité. Nous définissons ces facteurs comme étant les coordonnées de points X tel que $X = (X_1, \dots, X_4)$. Chaque point est donc caractérisé par ces facteurs.

Estimation de Y noté Y^{\sim} :

Pour un point X_i (dans notre modèle pour chaque point X pris et caractérisé à la date i), on a :

$$Y_i^{\sim} = F(X, \theta) = \theta_0 + \theta_1.X_{i1} + \theta_2.X_{i2} + \theta_3.X_{i3} + \theta_4.X_{i4}$$

$T_i = \{X_{i1} \dots X_{i4}\}$: La date i nous renseigne sur les facteurs du point X_i .

Avec θ les poids de facteurs dans l'expression de Y. Plus le facteur X_n est significatif et déterminant pour la valeur de Y alors plus le poids θ associé sera grand.

Nous allons déterminer θ qui minimise $\frac{1}{2} (\sum |Y_i - Y_i^{\sim}|^2)$ (l'erreur quadratique du système) dans la première phase de notre algorithme de Machine Learning : la phase d'entraînement.

ii. Première étape : la phase d'entraînement

Pour cette étape, l'algorithme doit déterminer les poids θ les plus pertinents pour notre modèle. Pour ce faire, nous avons besoin d'un échantillon de N points $X_n = (X_1, \dots, X_4)$ ainsi que leur Y (réel donc $\neq Y^{\sim}$) associés. Il faut bien comprendre qu'à cette étape nous travaillons

uniquement avec des données passées dont nous connaissons donc X et Y afin de trouver la corrélation entre X, Y et Téta.

Il faut prendre un N très grand car plus N est grand et donc plus nous avons de données réelles et plus le système est précis. Il nous faut donc recueillir une grande base de données. Chaque donnée appelée point sera donc recueillie dans le passé à un temps différent T_i pour avoir des cours d'action différents.

Soit X et Y les matrices suivantes :

$$X = \begin{bmatrix} 1 & X_{11} & \dots & X_{14} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{N1} & \dots & X_{N4} \end{bmatrix} \in \mathbb{R}^{(N,6)} \quad N : \text{nombre de données.}$$

(Exemple : X_{N4} correspond au paramètre X_4 du point (ou donnée) N).

On ajoute une colonne de 1 dans X pour déterminer θ_0

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix} \in \mathbb{R}^{(N,1)} \quad N : \text{nombre de données.}$$

Détermination de θ :

$$\theta = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_N \end{bmatrix} \in \mathbb{R}^{(N+1,1)}$$

$$\theta = (X^T X)^{-1} X^T Y$$

Minimise le plus $\frac{1}{2} (\sum |Y_i - \tilde{Y}_i|^2)$

A la fin de l'entraînement, nous avons donc trouver des poids θ optimisés. Nous avons tous les éléments requis pour déterminer une prévision du cours de l'action futur \tilde{Y} .

iii. Phase de Test.

Nous devons maintenant tester la précision de notre algorithme. Pour ce faire nous comparons la sortie prédite d'une entrée jamais vu au préalable par notre algorithme avec sa sortie réelle afin de déterminer le pourcentage d'erreur entre la prédiction et la réalité. Nous trouvons une erreur généralement comprise entre 3 et 6% ce qui est très concluant.

iv. Phase de Prédiction

Nous connaissons maintenant intégralement notre expression de $\tilde{Y} = F(X, \theta)$ pour calculer le cours de l'action futur en fonction d'un point $X = (X_1, \dots, X_4)$.

Pour prédire le cours de l'action futur, nous récupérons $T_0 = \{X_1, \dots, X_4\}$ qui correspond aux facteurs actuels nous donnant notre point X que l'on injecte dans $\tilde{Y} = F(X, \theta)$. Cela nous donnera donc \tilde{Y} le cours de l'action future en fonction des paramètres relevés actuels et permettra à l'investisseur de connaître le rendement de son éventuel investissement par rapport à sa mise initiale (c'est-à-dire le prix de l'action récent X_4).

Soit G^{\sim} le gain en pourcentage supposé, on a : $G^{\sim} = \frac{Y^{\sim} - X^4}{X^4} \cdot 100$.

Selon le gain supposé, l'investisseur sera libre d'investir ou non en fonction de ses attentes en termes de gain.

Prévision du cours d'une action avec Random Forest

Nous avons décidé d'implémenter un arbre de décisions pour produire une autre prévision en complément de la régression linéaire. L'algorithme sélectionne automatiquement les variables explicatives discriminantes à partir de notre base de données lui permettant ainsi d'extraire des règles logiques de cause à effet qui n'apparaissaient pas initialement et ainsi prédire une sortie associée à l'entrée. Un arbre de régression se construit de manière itérative, en découpant à chaque étape la population en deux sous-ensembles. Le découpage (ou test) s'effectue suivant des règles simples portant sur les variables explicatives, en déterminant la règle optimale qui permet de construire deux populations les plus différenciées en termes de valeurs de la variable à expliquer.

De plus nous avons utilisé la méthode ensembliste qui consiste à coupler différents algorithmes de Machine Learning afin d'en tirer un modèle généralisant beaucoup mieux que les algorithmes initiaux. Ainsi nous avons implémenter un Random Forest c'est-à-dire plusieurs arbres de décisions dont nous faisons la moyenne des sorties. Nous avons eu recours à scikit learn pour implémenter notre algorithme.

Régularisation Ridge + cross-validation

Lorsque l'on utilise une régression linéaire avec la méthode des moindres de carrés, il y a un risque de générer un modèle trop complexe, car correspondant uniquement au jeu de données utilisé lors des tests de régression. Cela s'appelle l'overfitting.

Pour que notre modèle soit généralisable, et soit adapté aux données futures, il doit avoir une complexité modérée, par exemple en réduisant le nombre de variables explicatives.

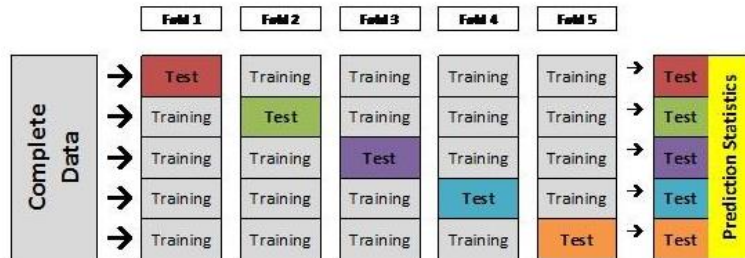
Ridge ajoute donc un terme pénalisant à l'erreur à minimiser, de façon à réduire l'importance des variables explicatives non pertinentes :

$$\mathcal{D}(\beta; \lambda) = \mathcal{D}(\beta) + \lambda \sum_{j=1}^p \beta_j^2$$

Ridge permet donc une régularisation du modèle en réduisant l'overfitting.

Lorsque nous entraînons un modèle en validation simple, nous sommes en outre contraints de séparer le jeu de données entre données d'entraînement (~70%) et données test (~30%). Certaines données sont donc "perdues" pour être testées et ne servent pas à entraîner le modèle.

C'est pourquoi nous avons utilisé la cross-validation, qui consiste à tester notre modèle de multiples fois en changeant de données d'entraînement de données test, puis de faire la moyenne des résultats obtenus.



Cela permet de tirer profit de l'ensemble de nos jeux de données.

c. Volatilité

L'investisseur cherchant quel actif acquérir prête attention d'une part au rendement potentiel offert, et d'autre part au risque que cela implique.

Nous pouvons considérer que plus un actif offre un rendement stable au cours du temps, plus il est sûr. Inversement, plus son rendement a tendance à varier largement, plus le risque est élevé.

La volatilité (σ) indique la dispersion des rendements de l'action par rapport à la moyenne des rendements.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (R_i - R_m)^2}{n}}$$

Avec

R_i le rendement annuel

R_m la moyenne des rendements annuels

n le nombre d'années

Outre le gain potentiel espéré, il est sage de faire attention à ce que la volatilité ne soit pas trop élevée.

IV. Difficultés rencontrées

Naomi Oiknin : Au début du projet, la difficulté a été de tenter de repérer quels indices serait pertinent d'inclure dans le projet. A cette époque je venais de me spécialiser en finance et je n'avais pas vraiment de connaissances ni de recul dans ce domaine. Plus tard dans l'avancement du projet je me suis occupée de l'algorithme de la Value At Risk en simulation de Monte Carlo.

Au moment de coder je me suis rendue compte qu'il y avait seulement très peu de documentation sur le calcul d'un des termes (à savoir le processus de Wiener) en Python. Enfin une dernière difficulté que je pense nous avons tous rencontrée est celle de la valorisation. En effet, il est difficile de créer une communauté open source impliquée alors que le projet n'est pas encore abouti.

Luis Molano : La gestion des événements/ passage d'une page à une autre, tout l'aspect visuel était le plus compliqué à aborder selon moi. Ainsi, la difficulté principale que j'ai pu rencontrer tout au long du projet est le passage du mode console au mode graphique avec tkinter. Cependant je pense avoir réussi à surpasser cette difficulté car le rendu visuel est plutôt correct.

La liaison de la base de données avec l'algorithme de machine Learning fût aussi une étape importante du développement, en plus de trouver des données réelles à exploiter.

Cependant python étant un langage facile à apprendre et couramment utilisé dans la science des données, le nombre de tutoriels sur internet est important et m'a permis de m'améliorer fortement tout au long du projet.

Dorian Hamel : Ayant déjà eu des cours de python ainsi que des connaissances dans d'autres langages, la programmation n'a pas été la partie la plus difficile de mon travail. La tâche la plus ardue était les algorithmes de machine Learning, premièrement comprendre quel algorithme est pertinent dans quel cas et pourquoi en préférer un par rapport à un autre. Ensuite, comprendre les fonctions mathématiques sous-jacentes sur lesquels ces algorithmes fonctionnent.

Benoit Chaurand : La difficulté principale que j'ai rencontré a été l'affichage d'images, et surtout de la position de chaque élément sur la page, pour la rendre la plus cohérente et intuitive possible. En effet Tkinter se différenciant de manière importante de toutes les autres bibliothèques graphiques que j'ai utilisées, il a fallu tout apprendre et découvrir. Heureusement, Tkinter étant massivement utilisé, j'ai pu trouver des tutoriels et forums bien expliqués.

Edouard Bouault : J'ai rencontré au cours du projet deux difficultés principales. La première a été de comprendre l'utilisation de la bibliothèque Tkinter que je n'avais jamais utilisée auparavant, en particulier la conception d'une application avec des frames ainsi que de leur layout.

J'ai aussi eu des difficultés à trouver des données fiables sur la période et aux intervalles voulus pour tester les algorithmes avec de vrais valeurs, avant de suivre la suggestion de M. Kim et de me diriger vers le site Yahoo finance.

Rémy Duchene : Pour ma part la partie la plus difficile de l'implémentation de nos algorithmes de Machine Learning ne résidait pas dans la conception du modèle et son implémentation, les modèles mathématiques associés et notre approche étant assez clair, mais plutôt dans le fait de relier nos algorithmes aux bases de données. Ce fut une étape assez fastidieuse et décisive pour tester véritablement nos algorithmes sur des données réelles et ainsi savoir si notre conception était efficace.

V. Perspectives d'avenir

La première version de notre logiciel étant opérationnelle, nous l'avons partagée de manière open-source via la plateforme GitHub.

Les développeurs intéressés par notre projet seront très bénéfiques, tant pour implémenter de nouvelles fonctionnalités que pour partager leurs idées sur les innovations possibles.

Nous aurons aussi les retours de certains utilisateurs, n'ayant que très peu ou pas de connaissances en informatique. Ceux-ci sont tout aussi importants et pourront s'avérer cruciaux dans les décisions que nous aurons peut-être à prendre.

De notre côté, nous nous sommes déjà intéressés à plusieurs points qu'il serait possible d'améliorer :

- Améliorer la qualité du code
- En ce qui concerne nos algorithmes de machine Learning, un meilleur modèle prévisionnel, prenant en compte davantage de variables explicatives relatives à l'entreprise (notamment sur le bilan financier, les ressources humaines, la concurrence...).
- Augmenter la quantité de données historiques
- Pour l'analyse du risque, prendre en compte plus de paramètres également (ex : données macroéconomiques)
- Permettre aux utilisateurs de créer un compte afin de s'identifier
- Améliorer l'ergonomie de notre interface

VI. Annexe

Ci-dessous vous trouverez un aperçu des données de l'entreprise Airbus stockées dans notre base de données.

DB Browser for SQLite - C:\Users\molant\Desktop\PPEFIN\RiskLess\RiskLess\PPE.db

Fichier Édition Vue Aide

Nouvelle base de données Ouvrir une base de données Enregistrer les modifications Annuler les modifications

Structure de la Base de Données Parcourir les données Éditer les Pragma Exécuter le SQL

Table : entreeX

	nom_action	acid	benefice	actifs	dividendes	prix_recent
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	Airbus	1	719250000	113937000000	0	83
2	Airbus	2	368750000	93311000000	0	39.7
3	Airbus	3	368750000	93311000000	0.6	41.44
4	Airbus	4	368750000	93311000000	0	47.59
5	Airbus	5	368750000	93311000000	0	55.81
6	Airbus	6	587500000	96102000000	0	52.41
7	Airbus	7	587500000	96102000000	0.75	48.72
8	Airbus	8	587500000	96102000000	0	49.76
9	Airbus	12	674500000	106681000000	0	53.99
10	Airbus	13	674500000	106681000000	0	62.58
11	Airbus	14	250000000	111133000000	0	57.15
12	Airbus	15	250000000	111133000000	1.3	51.86
13	Airbus	16	250000000	111133000000	0	53.84
14	Airbus	17	250000000	111133000000	0	62.68
15	Airbus	18	719250000	113937000000	0	71.4

1 - 15 de 20 Aller à : 1

DB Browser for SQLite - C:\Users\molant\Desktop\PPEFIN\RiskLess\RiskLess\PPE.db

Fichier Édition Vue Aide

Nouvelle base de données Ouvrir une base de données Enregistrer les modifications Annuler les modifications

Structure de la Base de Données Parcourir les données Éditer les Pragma Exécuter le SQL

Créer une table Créer un Index Modifier une Table Supprimer la Table

Nom	Type	Schéma
Tables (2)		
entreeX		CREATE TABLE entreeX(nom_action varchar(20) not null, acid int not null, ben
nom_action	varchar (20)	`nom_action` varchar (20) NOT NULL
acid	int	`acid` int NOT NULL
benefice	int	`benefice` int
actifs	int	`actifs` int
dividendes	int	`dividendes` int
prix_recent	decimal (7 , 3)	`prix_recent` decimal (7 , 3) NOT NULL
sortieY		CREATE TABLE sortieY(nom_action varchar(20) not null, acid int not null, prix
nom_action	varchar (20)	`nom_action` varchar (20) NOT NULL
acid	int	`acid` int NOT NULL
prix_réel	int	`prix_réel` int NOT NULL
Index (0)		
Vues (0)		
Déclencheurs (0)		

Table : sortieY

Nouvel Enregistrement

Supprimer l'enregistrement

	nom_action	acid	prix_réel
	Filtre	Filtre	Filtre
1	Airbus	2	41.44
2	Airbus	3	47.59
3	Airbus	4	55.81
4	Airbus	5	52.41
5	Airbus	6	48.72
6	Airbus	7	49.76
7	Airbus	8	41.35
8	Airbus	12	62.58
9	Airbus	13	57.15
10	Airbus	14	51.86
11	Airbus	15	53.84
12	Airbus	16	62.68
13	Airbus	17	71.4
14	Airbus	18	72.72
15	Airbus	19	80.33

1 - 15 de 19

Aller à :

1