Projet Stoc_King

Jingcheng Wu, Qinye Shi, Runzhou Xu, Ye Liu

03/2018

Contents

List of Listings

List of Figures

1 Analyse du système et Architecture logiciel

Pour analyser notre système nous avons pris "Capella" pour modélisation graphique des systèmes. Pour simplement présenter notre idée, nous prenons deux graphes *System Architecture Blank* et *Logical Architecture Blank*. Puis nous discutons la faisabilité des algorithmes pour prédire le prix, analyser les nouvelles, et proposer les stratégies.

1.1 Système analyse

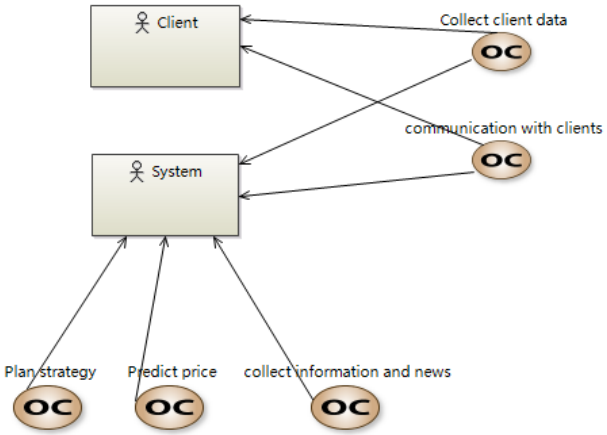

Figure 1: OAB

Pour notre système, les fonctions principales sont *collect client data*, *communication with clients*, *plan strategy*, *predict price*, et *collect information and news*.

1.2 Analyse du système

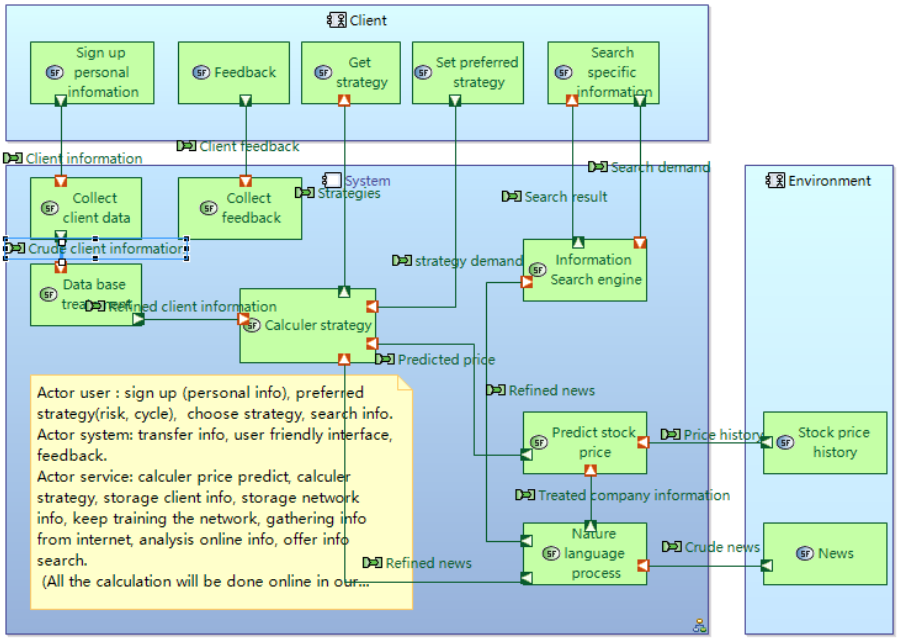

Figure 2: System Architecture Blank

Comme dans cette figure, notre système peut :

- Collectionner l'information du clients et les intégrer dans notre base de données.
- Permettre les clients commenter leur expérience.
- Analyser les nouvelles en utilisant Machine Learning.
- Prédire le prix du stock à partir du nouvelle et l'histoire.
- Proposer quelques stratégies à partir du *Predicted price*, *Refined news*, *strategy demand*, et *Refined client information*.

1.3 Architecture logiciel

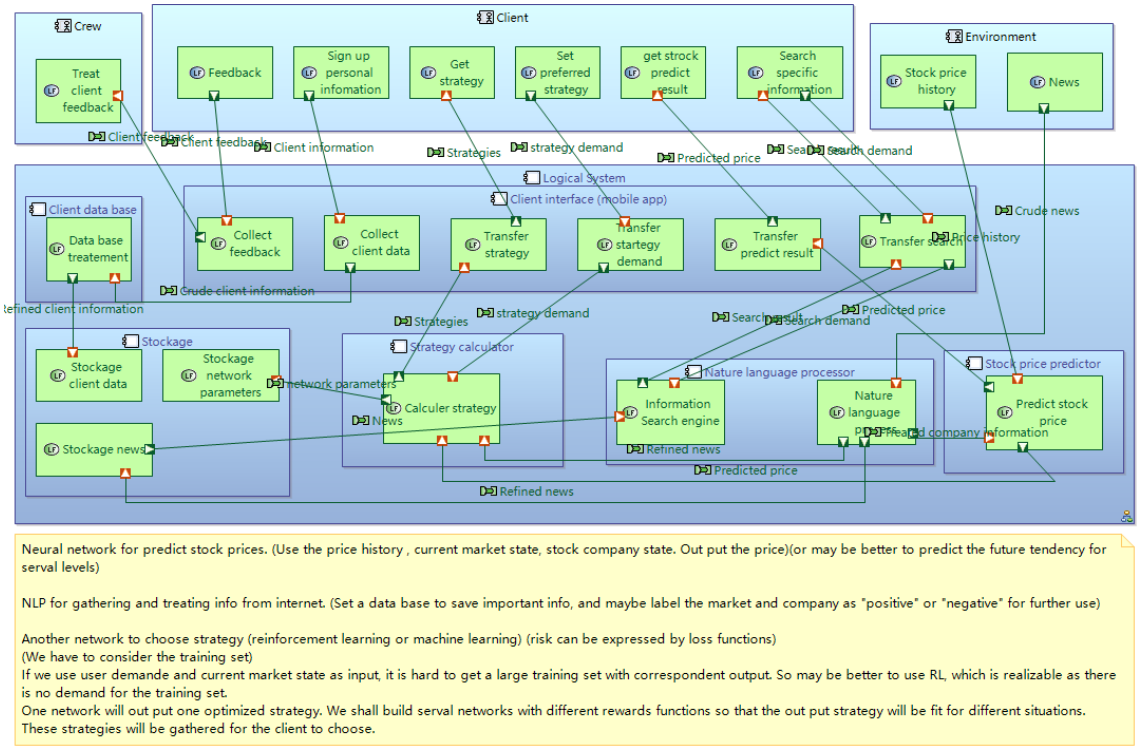

Figure 3: Logical Architecture Blank

Pour des raisons de confidentialité, nous avons décidé de prendre un interface (une application mobile) pour les clients, qui transmet tous les information. Et tous les calcules vont être traité par notre serveur.

1.4 Une chaîne pour demander les stratégies

Dans la figure suivant, nous vous proposons un processus pour demander les stratégies:

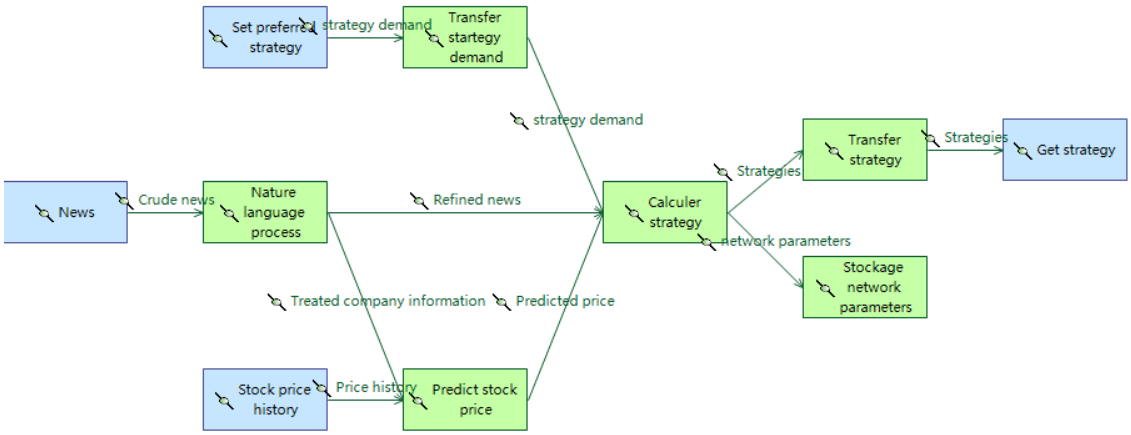

Figure 4: Chaîne de *Ask for strategy*

Les clients peut faire de demande, et notre système va collectionner les information via internet, et traiter les nouvelles et l'histoire du prix du stock. A partie de ces information, notre système peut puis étudier les stratégies plus efficace et profitable.

2 Analyse de faisabilité

Dans cette partie, nous vous présentons la faisabilité de notre projet. Principalement, notre système va être construit sur trois parties.

Premièrement, un partie de Natural Language Processing, qui utilise l'algorithme du RNN et LSTM pour traiter les nouvelles, et retourner un valeur qui signifie l'influence des nouvelles sur le marché.

Ensuite, un modèle de DNN pour prévoir la tendance du prix du stock. Parce que c'est très difficile de prédire le prix exact.

Dernièrement, un modèle de Reinforcement learning comme AlphaGO, qui utilise les nouvelles traité et le tendance du prix, pour proposer quelques stratégies au client.

2.1 Partie de NLP

Cette partie d'algorithme est pour traiter les nouvelles. Nous pouvons utiliser algorithme de RNN(Recurrent neural network) et LSTM(Long short-term memory) pour réaliser cette fonction.

Ensuite, je vais vous plus préciser ces algorithmes.

2.1.1 Classification des nouvelles

En informatique, l'opinion mining (aussi appelé sentiment analysis) est l'analyse des sentiments à partir de sources textuelles dématérialisées sur de grandes quantités de données (big data). [1]

Il y a beaucoup de articles sur ce sujet(par exemple [2]) Et nous avons testé sur la première partie du analyse de sentiment. Nous avons étudier *Label classification*.

	title	tags
0	How to draw a stacked dotplot in R?	[r]
1	mysql select all records where a datetime fiel...	[php, mysql]
2	How to terminate windows phone 8.1 app	[c#]
3	get current time in a specific country via jquery	[javascript, jquery]
4	Configuring Tomcat to Use SSL	[java]

Figure 5: *Exemple de data*

[1] sentiment analyse https://fr.wikipedia.org/wiki/Opinion_mining

[2] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and Trends® in Information Retrieval 2.1–2 (2008): 1-135.

Et puis nous avons testé de classifier ces document.

```
Title:  contenttype application json required rails
True labels:    ruby,ruby-on-rails
Predicted labels:       json,ruby-on-rails
```

Figure 6: *Exemple de Output*

Comme le montre la figure ci-dessous, l'effet global est très bon, ce qui prouve que l'on peut aussi implémenter la classification de l'algorithme. Et si nous utilisons d'autres méthodes d'optimisation, nous pouvons obtenir de meilleurs résultats.

```
DCG@   1: 0.287 | Hits@   1: 0.287
DCG@   5: 0.347 | Hits@   5: 0.400
DCG@  10: 0.363 | Hits@  10: 0.449
DCG@ 100: 0.399 | Hits@ 100: 0.629
DCG@ 500: 0.425 | Hits@ 500: 0.834
DCG@1000: 0.442 | Hits@1000: 1.000
```

Figure 7: Accuracy

Plus tard, nous avons testé l'algorithme pour classer les articles. Les résultats obtenus sont présentés dans la figure ci-dessous: En raison de contraintes de temps et de ressources, nous n'avons effectué que des tests de base et une formation. Enfin, nous pouvons voir que nos résultats sont acceptables.

```
———————— Train set quality: ————————
processed 105778 tokens with 4489 phrases; found: 4528 phrases; correct: 4386.

precision:  96.86%; recall:  97.71%; F1:  97.28

            company: precision:   97.10%; recall:   99.07%; F1:   98.08; predicted:   656
           facility: precision:   94.43%; recall:   97.13%; F1:   95.76; predicted:   323
            geo-loc: precision:   98.11%; recall:   99.10%; F1:   98.60; predicted:  1006
              movie: precision:   94.12%; recall:   94.12%; F1:   94.12; predicted:    68
       musicartist: precision:   95.38%; recall:   97.84%; F1:   96.60; predicted:   238
             other: precision:   95.42%; recall:   96.30%; F1:   95.86; predicted:   764
            person: precision:   98.75%; recall:   98.42%; F1:   98.59; predicted:   883
           product: precision:   97.15%; recall:   96.54%; F1:   96.85; predicted:   316
        sportsteam: precision:   97.69%; recall:   97.24%; F1:   97.46; predicted:   216
            tvshow: precision:   81.03%; recall:   81.03%; F1:   81.03; predicted:    58
```

Figure 8: Résultat de training set

```
———————— Validation set quality: ————————
processed 12836 tokens with 537 phrases; found: 422 phrases; correct: 193.

precision:  45.73%; recall:  35.94%; F1:  40.25

             company: precision:   66.28%; recall:   54.81%; F1:   60.00; predicted:    86
            facility: precision:   41.18%; recall:   41.18%; F1:   41.18; predicted:    34
             geo-loc: precision:   73.42%; recall:   51.33%; F1:   60.42; predicted:    79
               movie: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:     9
         musicartist: precision:   15.38%; recall:   14.29%; F1:   14.81; predicted:    26
               other: precision:   37.50%; recall:   25.93%; F1:   30.66; predicted:    56
              person: precision:   46.03%; recall:   25.89%; F1:   33.14; predicted:    63
             product: precision:   25.00%; recall:   11.76%; F1:   16.00; predicted:    16
          sportsteam: precision:   23.08%; recall:   30.00%; F1:   26.09; predicted:    26
              tvshow: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:    27
```

Figure 9: Résultat de validation set

```
———————— Test set quality: ————————
processed 13258 tokens with 604 phrases; found: 471 phrases; correct: 238.

precision:  50.53%; recall:  39.40%; F1:  44.28

             company: precision:   62.71%; recall:   44.05%; F1:   51.75; predicted:    59
            facility: precision:   45.45%; recall:   42.55%; F1:   43.96; predicted:    44
             geo-loc: precision:   76.32%; recall:   52.73%; F1:   62.37; predicted:   114
               movie: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:     5
         musicartist: precision:    3.12%; recall:    3.70%; F1:    3.39; predicted:    32
               other: precision:   40.86%; recall:   36.89%; F1:   38.78; predicted:    93
              person: precision:   66.67%; recall:   42.31%; F1:   51.76; predicted:    66
             product: precision:   17.65%; recall:   10.71%; F1:   13.33; predicted:    17
          sportsteam: precision:   36.36%; recall:   25.81%; F1:   30.19; predicted:    22
              tvshow: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:    19
```

Figure 10: Résultat de test set

Les résultats sont très satisfaisants.Pour d'autres tests à l'avenir, nous allons tester l'analyse émotionnelle des nouvelles et la classification des nouvelles.

Il y a beaucoup de recherche fondamentale sur Internet qui peut nous guider et je crois en notre capacité. Nous pouvons enfin faire un système d'analyse des nouvelles très parfait.

2.1.2 Sentiment analyse

Ensuite, nous analysons la possibilité d'analyser l'impact des nouvelles. Analyser l'impact d'un événement de nouvelles sur le marché est quelque peu analogue à une analyse de sentiment de nouvelles. Les nouvelles négatives auront un impact négatif sur le marché, tandis que les nouvelles positives auront un impact positif.

Nous pensons donc que nous pouvons d'abord utiliser l'apprentissage de la migration(transfer machine learning) et utiliser le modèle de réseau neuronal qui a déjà entraînées. Nous pouvons analyser davantage les nouvelles (telles que l'impact des mêmes nouvelles sur différentes entreprises, etc.).

Alors, nous regardons la capacité du réseau de neurones à analyser les émotions. Ici, les chercheurs utilisent les données sur IMBD [3]. Il y a 25000 commentaires positives et 25000 commentaires de négatives.

Figure 11: Exemple de commentaire

Pour ce data set:

- Contains at most 30 reviews per movie
- At least 7 stars out of 10: positive (label = 1)
- At most 4 stars out of 10 : negative (label = 0)

Utilisez la méthode du *bag of 1-grams with TF-IDF*. Enfin, sur l'ensemble de test, la précision obtenue était de 88,5%. Voici un vocabulaire appris par les réseaux de neurones, certains montrant des émotions positives (et négatives). [4]

ngram	weight	ngram	weight
great	9.042803	worst	-12.748257
excellent	8.487379	awful	-9.150810
perfect	6.907277	bad	-8.974974
best	6.440972	waste	-8.944854
wonderful	6.237365	boring	-8.340877
Top positive		Top negative	

Figure 12: Top positive et Top négative

Afin de promouvoir notre vision, de nombreux algorithmes et modèles peuvent être utilisés.

- *bag of words of n-grams* Peut obtenir plus d'informations entre les mots.
- 1-D Convolution Neural netword [5] Il y a des résultats plus précis et des vitesses de calcul plus rapides.

[3] IMBD mouvie reviews dataset http://ai.stanford.edu/~amaas/data/sentiment

[4] Ces résultat viennent de https://www.coursera.org/learn/language-processing/lecture/T7fNB/linear-models-for-sentiment-analysis

[5] un exemple ici http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

- Deep learning, *Probabilistic neural language model*, RNN, LSTM, *Attention model*. Le réseau peut se rappeler plus d'informations avant. Et le degré d'attention aux différentes informations changera selon le problème. [6]

2.2 Partie de DNN

Dans cette section, nous utilisons l'apprentissage en profondeur pour tester et analyser les données boursières. Nous utilisons les données de tushare. [7]

2.2.1 Prédire le prix

Premièrement, nous essayons de prédire les prix futurs des actions en fonction des directives de LSTM.

Tout d'abord, nous avons normalisé les données de stock, et les données de jeu d'entraînement qui en résultent sont présentées ci-dessous.

	bt_Close	bt_Volume	bt_close_off_high	bt_volatility	eth_Close	eth_Volume	eth_close_off_high	eth_volatility
690	0.000000	0.000000	-0.560641	0.020292	0.000000	0.000000	-0.418477	0.025040
689	-0.002049	-0.170410	0.250597	0.009641	-0.011498	0.239937	0.965898	0.034913
688	-0.009946	0.092475	-0.173865	0.020827	0.025190	0.978201	-0.317885	0.060792
687	-0.002855	0.060603	-0.474265	0.012649	0.006810	0.680295	-0.057657	0.047943
686	-0.005457	-0.048411	-0.013333	0.010391	0.002270	0.066829	0.697930	0.025236
685	-0.012019	-0.061645	-0.003623	0.012782	0.002991	0.498534	-0.214540	0.026263
684	0.054613	1.413585	-0.951499	0.069045	-0.006349	2.142074	0.681644	0.040587
683	0.043515	0.570968	0.294196	0.032762	0.040890	1.647747	-0.806717	0.055274
682	0.030576	-0.110282	0.814194	0.017094	0.040937	0.098121	-0.411897	0.019021
681	0.031451	-0.007801	-0.919598	0.017758	0.054014	0.896944	-0.938235	0.025266

Figure 13: Training set

Voici notre prédiction de l'ensemble d'entraînement.

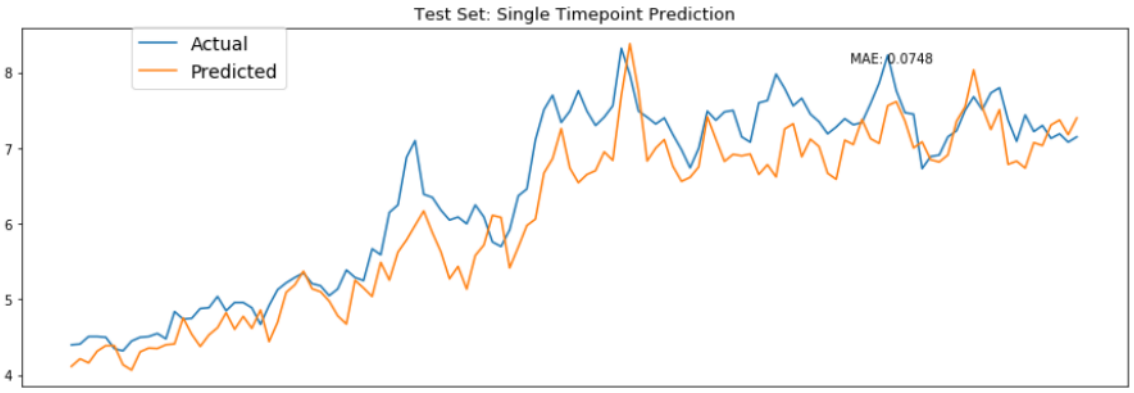

Figure 14: Résultat de test set

Comme le montre la figure, nous avons une bonne prévision des tendances futures. Cependant, certains chercheurs ont émis des doutes sur le fait que la simple utilisation de prédictions aléatoires peut donner de bons résultats.

[6] uses LSTM or GRU and gradient clipping https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[7] Tushare http://tushare.org/

$$PredPrice_t = PredPrice_{t-1} * \epsilon, \epsilon \sim N(\mu, \sigma) \ \& \ PredPrice_0 = Price_0$$

Figure 15: Expression de prédictions aléatoires

Ainsi, nous avons comparé les résultats du LSTM avec ceux prédits par la formule ci-dessus. Après cela, nous avons eu le résultat de leur perte.

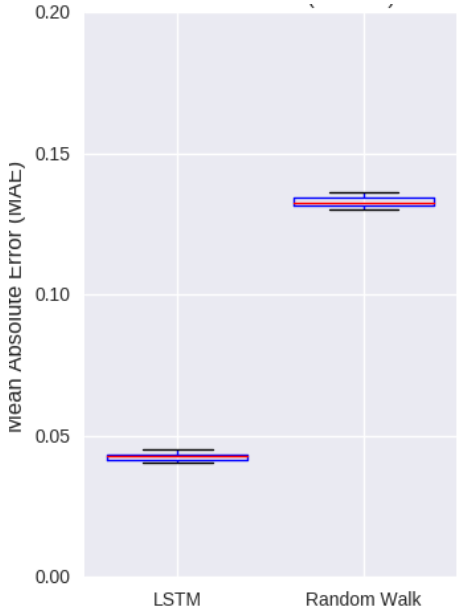

Figure 16: Comparaison de perte

Nous pouvons voir que globalement, LSTM obtient une erreur plus petite. Sa prédiction pour les stocks est utile!

2.2.2 Résultat de Facebook prophet

Après cela, nous avons utilisé l'algorithme de Facebook *Prophet* pour effectuer des tests simples, un algorithme open source développé par facebook et utilisé pour l'algorithme d'analyse de séries chronologiques. [8]

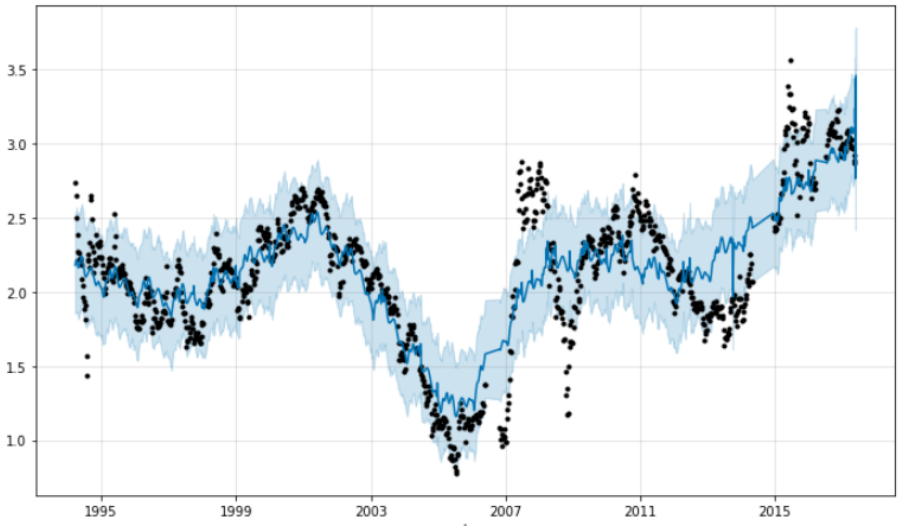

Figure 17: Résultat de Facebook prophet

[8] Facebook prophet https://github.com/facebook/prophet

3 Prédire la tendance

Bien que l'algorithme ci-dessus montre l'effet sur la prévision des prix, selon notre pensée, il est encore irréaliste d'utiliser ces algorithmes comme un guide pour investir dans les choix.

Mais utiliser ces algorithmes pour prédire les tendances futures des prix est beaucoup plus précis. Nous croyons d'abord que nous pouvons limiter nos résultats de prévision à cinq niveaux. (2: hausse rapide, 1: hausse lente, 0: maintien stable, -1: légère baisse, -2: légère baisse)

Nous avons étudié beaucoup de littérature de recherche chinoise dans ce domaine et constaté qu'ils ont tous obtenu de très bons résultats (taux de précision allant jusqu'à 80% pour les prévisions futures des stocks).

En fin de compte, nous supposons que notre système devrait avoir les éléments suivants:

- Input : Les cours historiques des actions. Et le résultat de l'analyse précédente des sentiments (c'est-à-dire, si le développement de l'entreprise était positif ou négatif pendant cette période).
- Output : La prévision de la tendance du cours des actions pour une période de temps (par exemple, un mois) (préliminaire rédigée en cinq niveaux - 2: hausse rapide, 1: hausse lente, 0: maintien stable, -1: légère baisse, -2: légère baisse).
- Normaliser les données, et un peu de pré-traitement. Amélioration des données.
- En utilisant un cadre d'apprentissage en profondeur, plusieurs réseaux neuronaux différents sont utilisés pour prédire le résultat séparément.
- Les algorithmes ou modèles que nous pouvons utiliser sont: RNN, LSTM, Prophet, CNN, etc.
- Agréger les résultats prévisionnels de tous les modèles et enfin les intégrer dans un résultat.

4 RL reinforcement learning

En raison de contraintes de temps, nous n'avons pas encore commencé à tester cette partie, nous avons juste des idées préliminaires.

Afin de mieux comprendre nos idées, les plans suivants utilisent Go pour faire l'analogie. Parce que la réalisation de *Alpha Go* dans le domaine de Go est choquant.[9]

Pour notre cas, l'environnement économique est un environnement quantifié (à l'exception de certaines actualités politiques, pour lesquelles nous utiliserons le traitement du langage naturel pour quantifier).

Pour notre système, les règles sont les règles d'achat et de vente: ces investisseurs normaux suivront ces règles. Cela facilite la construction d'un modèle.Tous les comportements d'achat et de vente peuvent être exprimés en chiffres. Les règles d'achat et de vente sont les mêmes que les règles d'utilisation de Go.

Les récompenses et les punitions pour gagner ou perdre sont plus facilement définies. Nous pouvons mesurer le montant d'argent gagné ou perdu.

De cette façon, nous construisons notre modèle d'apprentissage par renforcement.Considérez les résultats obtenus par *Alpha Go*. Nous sommes très informatifs sur un tel algorithme et il doit donner de bons résultats.

4.1

En utilisant les idées ci-dessus, nous pouvons construire notre modèle.
Formellement, la base du modèle d'apprentissage par renforcement consiste en : [10]

- un ensemble d'états S de l'agent dans l'environnement.
- un ensemble d'actions A que l'agent peut effectuer.
- un ensemble de valeurs scalaires "récompenses" R que l'agent peut obtenir.

Pour notre système, les éléments correspondants sont:

[9] Alpha Go https://deepmind.com/research/alphago/
[10] https://fr.wikipedia.org/wiki/Apprentissage_par_renforcement

- un ensemble des divers types d'investissements que nous détenons.
- un ensemble d'acheter ou vendre une action (ou un fonds, etc).
- Le montant total gagné.

Utilisez le pseudocode suivant pour entraîner notre modèle.

```
On initialise V(s) aleatoirement,
qui est la valeur que l'agent attribuera a chaque etat s.
    On initialise la politique a evaluer.
    On repete (pour chaque episode) :
        On initialise s
                On repete (a chaque pas de temps de l'episode) :
                        a = action donnee par la politique pour s
                        L'agent effectue l'action a; on observe la recompense r
                        et l'etat suivant s'
                        V(s) = V(s) + alpha * [r + gamma * V(s') - V(s)]
                        s = s'
                Jusqu'a ce que s soit terminal
```

5 Plan du notre système

Enfin, notre système est illustré ci-dessous.

Premièrement, un partie de Natural Language Processing, qui utilise l'algorithme du RNN et LSTM pour traiter les nouvelles, et retourner un valeur qui signifie l'influence des nouvelles sur le marché.

Ensuite, un modèle de DNN pour prévoir la tendance du prix du stock. Parce que c'est très difficile de prédire le prix exact.

Dernièrement, un modèle de Reinforcement learning comme AlphaGO, qui utilise les nouvelles traité et le tendance du prix, pour proposer quelques stratégies au client.

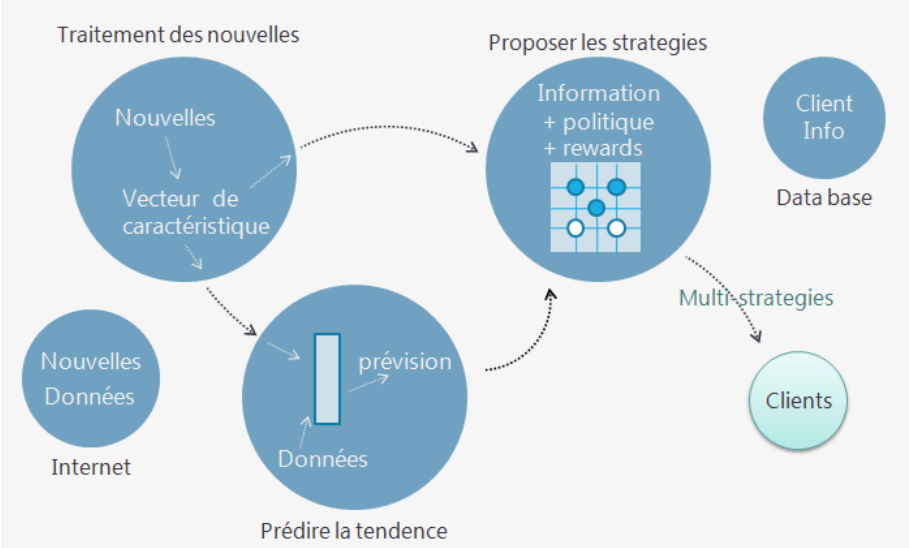

Figure 18: Plan du notre système