

Notes et précisions - Projet Python

Emilien JEMELEN, M1 2AD, année 2020-2021

1 Cadre du problème: définition d'un "*surebet*" et implications descriptives

Un joueur de paris en ligne aimerait gagner à coup sûr un montant net π_i lorsque l'issue i du pari survient, et ce pour toutes les issues (qui doivent former un système complet d'événements). En d'autres termes il ne veut pas d'aléa.

Les cotes $c_i > 1$ ainsi que les profits $\pi_i > 0$ ("*souhaités*" par le joueur) sont donc supposés exogènes, fixés respectivement par le joueur et par le bookmaker. Les mises sur chaque issue i du pari, notées m_i , et la mise totale, notée m , vérifient alors le système suivant:

$$\left\{ \begin{array}{l} \forall i \in \llbracket 1..n \rrbracket, m_i \cdot c_i - m = \pi_i \\ m = \sum_{i \in \llbracket 1..n \rrbracket} m_i \end{array} \right\} \iff \left\{ \begin{array}{l} \forall i, m_i = \frac{\pi_i + m}{c_i} \\ m = \sum \frac{\pi_i + m}{c_i} \end{array} \right\} \iff \left\{ \begin{array}{l} \forall i, m_i = \frac{\pi_i + m}{c_i} \\ m \cdot (1 - \sum \frac{1}{c_i}) = \sum \frac{\pi_i}{c_i} \end{array} \right.$$

Le membre de droite de la seconde ligne étant strictement positif (on a supposé que $\forall i, \pi_i > 0$), on doit avoir à gauche:

$$1 - \sum \frac{1}{c_i} > 0 \iff 1 > \sum \frac{1}{c_i} \iff \frac{1}{\sum \frac{1}{c_i}} > 1$$

2 Introduction de la notion de *taux de retour joueur* (TRJ)

On pose $\text{TRJ} = \frac{1}{\sum \frac{1}{c_i}}$

Si la condition $\text{TRJ} > 1$ est vérifiée alors on peut poursuivre la résolution.

NB: c'est un cas limite qui permet de s'abstraire de l'aléa et de fixer des profits positifs pour le joueur. En termes financiers c'est une opportunité d'arbitrage.

Sous cette condition, on a:

$$\left\{ \begin{array}{l} \forall i, m_i = \frac{\pi_i + m}{c_i} \\ m \cdot (1 - \frac{1}{TRJ}) = \sum \frac{\pi_i}{c_i} \end{array} \right\} \iff \left\{ \begin{array}{l} m = \frac{TRJ}{TRJ-1} \sum \frac{\pi_i}{c_i} \\ \forall i, m_i = \frac{\pi_i + \frac{TRJ}{TRJ-1} \cdot \sum \frac{\pi_k}{c_k}}{c_i} \end{array} \right.$$

3 Cas particulier: si $\forall i, \pi_i = \pi$

C'est-à-dire que le joueur veut faire π profit sur le pari généralement sans tenir compte des issues, on a (en reprenant le dernier système et en remplaçant π_i par π):

$$m = \frac{1}{TRJ-1} \cdot \pi \iff \pi = (TRJ - 1) \cdot m$$

Remarque n°1:

On comprend donc bien ici l'idée de *retour joueur*. En effet, si l'on a 1 euro à miser, que l'on mise cet euro de manière à obtenir le même profit quelle que soit l'issue du pari, alors on obtiendra TRJ euros en retour de la part de la loterie. Cela reste vrai si $TRJ < 1$, auquel cas on perdrait assurément de l'argent, puisque l'on ferait $TRJ - 1$ euro < 0 de profit.

Remarque n°2:

Cette situation est plus crédible que de souhaiter un profit distinct par issue, sauf pour des raisons de capacité à miser: on voit en effet à la 1.2. du dernier système que m varie linéairement selon le profit souhaité π_i avec une pente proportionnelle à $\frac{1}{c_i}$, donc en dérivant par rapport à π_i il apparaît que le coût d'une unité supplémentaire de profit sur l'issue i est inversement proportionnel à c_i . Autrement dit, plus c_i est grand, plus le coût d'une unité supplémentaire de profit sur l'issue i est faible. Il peut donc y avoir un choix à effectuer en fonction des ressources disponibles du joueur.

4 Objectif du projet

La vérification de la condition $TRJ > 1$ ne pourrait se faire qu'en croisant les cotes de différents sites. Même en faisant cela, cette condition a peu de chances

d'être jamais vérifiée en pratique: il semble évident que des algorithmes bien plus rapides que le mien doivent faire les mêmes calculs en permanence chez les bookmakers. Aussi, sur les plateformes, l'ajustement des cotes doit vraisemblablement se faire après calcul du TRJ croisé entre les sites pour un même pari.

Le but du projet est de collecter toutes les données de résultats de football sur un grand nombre de sites de paris en ligne accessibles depuis la France, d'en faire un DataFrame et de voir si de telles opportunités existent.

5 Pour la partie de code, voir le notebook du projet

<https://github.com/EmilienJemelen/Surebet-finder>