Sujet de Programmation large échelle PLE 2025-2026

L'objectif de ce projet est de concevoir un système d'analyse de la distribution statistique des *match-ups* dans le jeu vidéo *Clash Royale*. Un *match-up* correspond au choix de l'adversaire effectué par le système. Dans ce jeu, chaque joueur possède un ensemble de cartes et doit en choisir huit pour constituer un « deck » qui lui permettra d'affronter un adversaire disposant également d'un deck de huit cartes. Les joueurs peuvent appartenir à des clans et, selon le mode de jeu, deux joueurs d'un même clan ne peuvent pas s'affronter.

Nous avons enregistré près de 35 000 000 de parties dans notre *master dataset*. Pour chaque partie, nous disposons du deck des joueurs, d'informations sur leur niveau d'expérience, du niveau moyen des cartes de leurs decks et du clan du joueur.

L'objectif est de vérifier que le choix des adversaires est bien aléatoire. Pour cela, nous analyserons pour chaque combat les decks utilisés et vérifierons, pour chaque deck, si la probabilité de rencontrer un autre deck suit une distribution uniforme.

L'analyse peut se faire en considérant les huit cartes du deck des adversaires, mais nous étudierons également les sous-decks des joueurs (c'est-à-dire les combinaisons de 1, 2, 3, 4, 5, 6 ou 7 cartes que l'on peut extraire du deck initial de huit cartes ; on appelle ces sous-decks des archétypes). Certains decks étant plus joués que d'autres, nos analyses statistiques doivent impérativement prendre en compte leur fréquence d'utilisation.

Soit N(d) le nombre de fois que le deck (ie. archétype) d a été joué et Nall le nombre total d'occurrences rencontrées dans le jeu de données. Si le système est aléatoire, le nombre de fois qu'un deck d1 doit rencontrer un deck d2 est égal à N(d1) × N(d2) / Nall. Pour chaque couple de decks (ie. d'archétypes), on peut comparer la valeur observée à cette valeur théorique. Cela nous permet d'obtenir, pour chaque paire de decks, une valeur mesurée et une valeur estimée. Pour vérifier l'indépendance, il suffit ensuite de visualiser ces deux variables à l'aide d'un nuage de points : nous devrions obtenir une droite. La pente de cette droite devrait être de 1 si le système est parfaitement aléatoire et que nos mesures ne sont pas biaisées par d'autres facteurs.

Partie I (Map/Reduce Hadoop): Data Cleaning

Dans cette partie, il vous est demandé, en Hadoop, de nettoyer le jeu de données afin de garantir :

- Que toutes les lignes soient correctement formatées en JSON ;
- Que les decks comportent le bon nombre de cartes (exactement 8 pour chacun des deux adversaires) ;
- Qu'il n'y ait pas de doublons exacts ;
- Que les parties ne soient pas enregistrées plusieurs fois (mêmes joueurs, même date, même round).

L'ordre des joueurs peut varier selon que la partie a été téléchargée en prenant comme source le joueur A ou le joueur B. L'heure de la partie peut varier de quelques secondes pour des parties pourtant identiques vous devrez impérativement assurer ces parties sont bien détectées comme identiques.

Partie II (Nodes & Edges) :

À partir des données nettoyées, nous allons générer deux ensembles :

1- Un ensemble de nœuds D qui contient tous les decks (ou archétypes) rencontrés dans le jeu de données, avec le nombre de fois où ils ont été rencontrés et leur nombre de victoires.
2- Un ensemble d'arêtes qui représente le nombre de fois qu'un deck di a joué contre un autre deck dj, ainsi que le nombre de victoires de di sur dj.

Exemple ensemble de nœuds :
Archetype ;Count ;Win
0001;113176;57058
0002;11704;5606
0003;4816;2660
0004;13048;6689

Exemple ensemble d'arêtes :
Archetype source, Archetype target ; count ; win
0001;0001;44;19
0001;0006;34;17
0001;0007;42;18
0001;000a;171;77
0001;000f;41;22

Partie III (Stats) :

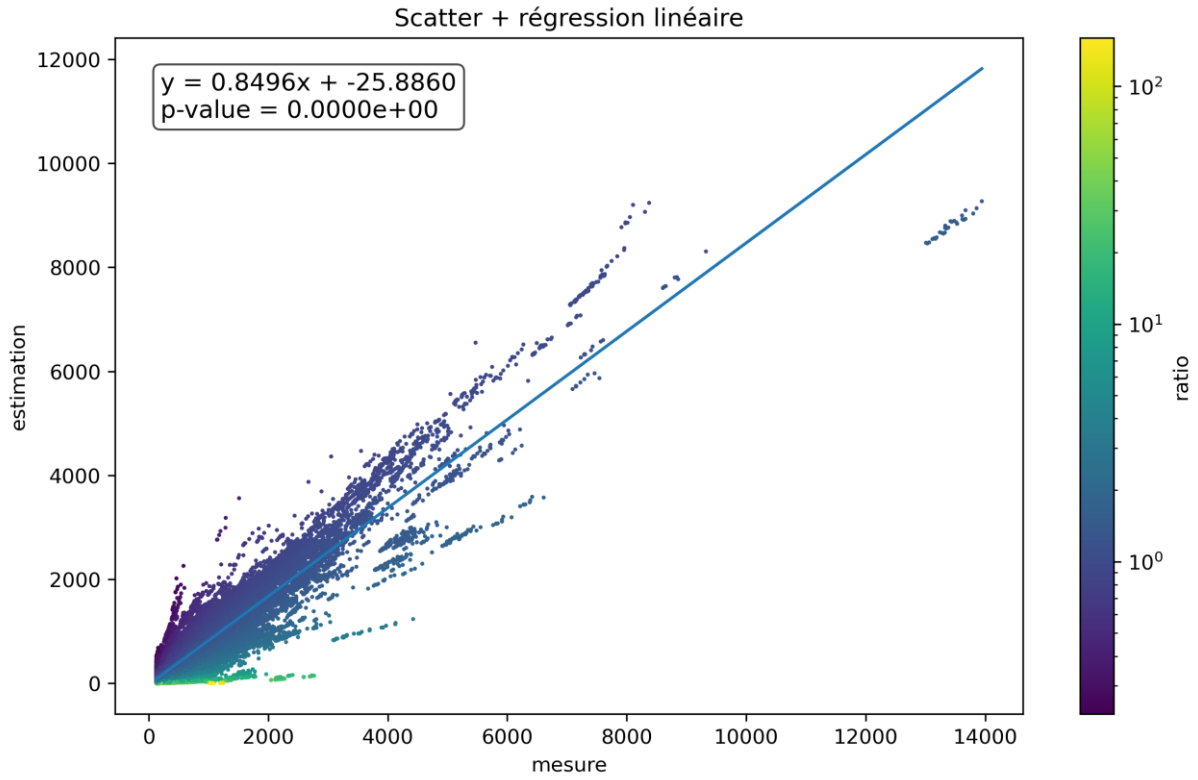
À partir des deux ensembles ci-dessus, nous pouvons générer, pour toutes les arêtes de notre jeu de données, un fichier comportant la valeur mesurée dans les données (résultat de la partie II) et la prévision que nous pouvons faire en fonction des données de départ. La prévision est obtenue en multipliant le nombre de fois où nous avons vu l'archétype source dans les données par le nombre de fois où nous avons vu l'archétype cible dans les données. Il faut ensuite normaliser ce produit en divisant par le nombre total d'archétypes considérés.

Le fichier de résultat sera de la forme suivante :

Archetype source, Archetype target ; count ; win; count source; count target; prevision
0001;0001;44;19;12051,12051; 39.5
0001;0006;34;17;12051,7840; 30.5
0001;0007;42;18;12051,4700; 47

Partie IV (Résultats) :

En analysant le fichier de statistiques sur votre machine, proposez une ou plusieurs analyses des résultats avec les outils statistiques de votre choix. Cette opération n'a pas besoin d'être réalisé avec le framework mapreduce et la plateforme LSD. Par exemple, vous pourrez avec python et mathplotlib créer un nuage de points représentant les valeurs trouvées dans notre jeu de données par rapport aux valeurs attendues, accompagné d'une régression linéaire comme ci-dessous.



Rendu attendu :

1- Le code complet de votre solution en Hadoop MapReduce
2- Un rapport comportant :
   a. L'explication de la solution mise en œuvre (pas besoin de détail/description sur le jeu Clash Royale)
   b. L'analyse théorique des performances attendues
   c. Un benchmark sur les temps d'exécution, les entrées sorties de vos map et de vos reduce etc… en utilisant les jeux de données de différentes tailles disponible sur HDFS.
   d. Une petite analyse statistique des résultats obtenus.
3- Une présentation orale avec slides du contenu du rapport