Institut national des sciences appliquées de Rouen

INSA de Rouen

Euro-Med, M8

Le Poker est-il seulement un jeu de pur hasard ?

Auteurs :
El Mehdi Kossir
el.kossir@insa-rouen.fr
Hamza Benkhalil
hamza.benkhalil@insa-rouen.fr

Enseignant :
Stéphane Canu
stephane.canu@insa-rouen.fr

23 Juin 2019

Résumé

Nous avons choisi à travers ce rapport d'analyser d'un point de vue statistique, la variante la plus populaire du jeu du poker : le Texas Hold'em.
Notre objectif étant de déterminer si il est possible de prédire les mains des joueurs.
Par le passé les chercheurs portaient plus leur attention aux échecs et au jeu de dames plutôt qu'au poker,comme en 1997 où IBM avait réussi à battre Gary Kasparov le plus grand champion du monde d'échecs à l'époque, par le biais de Deep Blue un superordinateur développé par les chercheurs de l'entreprise américaine.Un autre exemple est celui de l'université de l'Alberta qui a remporté le championnat du monde de dames en 1994 grâce à un programme infaillible. Ces deux événements ont attisé notre curiosité et nous nous sommes donc demandés si il était possible de développer un programme ou une IA capable de battre les meilleurs joueurs de poker.

Table des matières

Chapitre 1

Le jeu du poker

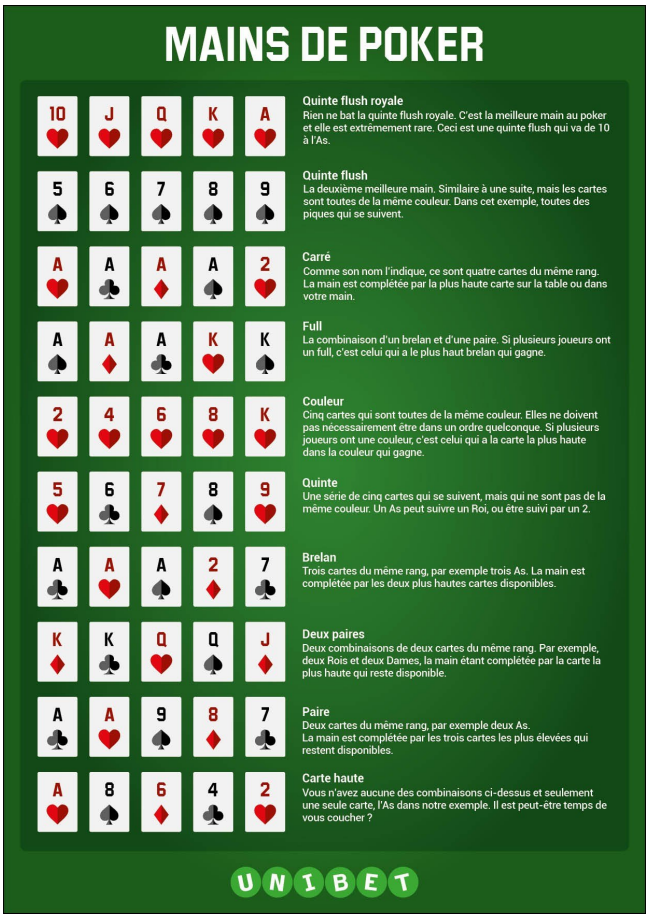
1.1 Introduction

Le poker est un jeu qui se joue avec des cartes standards, c'est aussi un jeu de compétition.
Au poker l'issue du jeu se décide de deux manières différentes :

-A l'abattage, la meilleure main gagne tous les enjeux ('le pot').

-Tous les joueurs sauf un ont renoncé à parier et ont abandonné le jeu. Le dernier gagne le pot directement.

Ainsi, lors d'une compétition, la meilleure main ne gagne pas nécessairement. Par conséquent, bluffer est important au poker. Au poker orthodoxe, le classement des différentes combinaisons de 5 cartes est le suivant :



1.2 Texas Hold'em

Aujourd'hui le poker sort de l'ombre et s'affiche à la lumière comme un véritable sport. Télévision, Internet, presse, nous assistons à une ruée vers le poker. Le poker est un des jeux de cartes les plus populaire joué par plus de 100 millions de joueurs dans le monde. Il y a plusieurs variantes de jeux mais seulement une est universelle et reconnu comme l'ultime test des qualifications et des stratégies du joueur, le Texas Hold'em .

Une main de poker est toujours constituée de 5 cartes. La combi-

naison la plus forte empoche le pot. Au Texas Hold'em vous devez composer la meilleure main possible à l'aide de vos deux cartes et des cinq cartes communes.

Le classement des cartes du plus haut (As) au plus bas (Deux) est le suivant :

Le déroulement d'une partie de Texas Hold'em d'après le site **www.everestpoker.com3** est le suivant :

Le croupier. Les cartes sont distribuées à partir de la position du croupier, qui tourne dans le sens horaire autour de la table après chaque main. Le joueur désigné comme Donneur pour une main donnée est identifié par un marqueur "D" sur la table. Le logiciel traitera automatiquement les cartes au nom du concessionnaire.

Stores. Pour commencer la partie, le joueur à la gauche du donneur place la petite blind (qui est généralement la moitié de la mise minimum) et le joueur à sa gauche place la grande blind (égale à la mise minimum). Ceci met le pot en marche et encourage les autres joueurs à parier. On les appelle "blinds" parce que ces paris sont faits avant qu'une carte ne soit vue.

Le Deal. Chaque joueur reçoit deux cartes, appelées Pocket Cards, qui ne sont visibles que par le joueur qui les détient.

Parier. Après que tous les joueurs ont reçu leurs deux cartes de poche, il y a un tour de mise qui commence par le joueur à la gauche du joueur qui a placé la grosse blind. Selon l'activité antérieure de la main en cours, un joueur peut se coucher, vérifier, suivre, miser, relancer ou relancer dans les limites des limites de la partie et des mises de la table. Un tour d'enchères peut encercler la table plusieurs fois s'il y a des relances et des relances. Le tour se termine lorsque tous les joueurs ont suivi la dernière mise ou se sont couchés.

Le Flop. Vient ensuite le Flop, où 3 cartes sont distribuées face visible dans la zone commune de la table, suivi d'un autre tour de mise.

Le Turn. Puis vient le Tour, où une 4ème carte commune est distribuée face visible à côté du Flop, et un troisième tour de mise a lieu.

La Rivière. La dernière carte commune, appelée Rivière, est placée face visible à côté du Tour et est suivie du dernier tour de mise.

Le Showdown. L'abattage a lieu après la fin des paris. Les autres joueurs comparent les meilleures mains de poker à cinq cartes qu'ils font en utilisant n'importe quelle combinaison de leurs 2 cartes de poche et les 5 cartes communes. Le joueur ayant le rang de main le plus élevé prend le pot. De temps en temps, deux joueurs ou plus auront des mains de rang égal ou la meilleure main possible est composée en utilisant les 5 cartes communes. Dans ces cas les joueurs impliqués se partagent le pot.

Main Suivante. Une fois le pot distribué, le bouton Donneur passe au joueur suivant à gauche (dans le sens des aiguilles d'une montre autour de la table), et les joueurs à gauche du nouveau Donneur posent les blinds pour que la donne pour une nouvelle main puisse commencer.

Chapitre 2

Description des données

2.1 Présentation des données

Notre base de données présente plusieurs enregistrements , chacun d'entre eux désigne un exemple d'une main composée de cinq cartes tirées d'un jeu standard de 52 cartes. Chaque carte est décrite en utilisant deux attributs (couleur et rang). Il y a un attribut de classe qui décrit la main. L'ordre des cartes est important, c'est pourquoi il y a 480 mains possibles de Royal Flush contre 4 (une pour chacune d'elles).

A chaque main nous avons associé un numéro :

0 : Carte haute (Nothing in hand).
1 : Paire (One pair).
2 : Deux paires (Two pairs).
3 : Brelan (Three of a kind).
4 : Quinte (Straight).
5 : Couleur (Flush).
6 : Full (Full house).
7 : Carré (Four of a kind).
8 : Quinte flush (Straight flush).
9 : Quinte flush royal (Royal flush).

Nous disposons aussi d'informations concernant les attributs de chaque carte :

1) **Couleur de la carte** : chiffre de 1 à 4 représentant : Coeur, Pique, Carreau, Trèfle

2) **Rang de la carte** : nombre de 1 à 13 représentant : L'As, 2, 3, ... , Reine, Roi

Il est aussi important de noter que nos données sont distribuées en deux sets : un set d'entraînement et un set de test.

Set d'entraînement :

0 : Nothing in hand, 12493 instances (49.95202% / 50.117739%)
1 : One pair, 10599 instances, (42.37905% / 42.256903%)
2 : Two pairs, 1206 instances, (4.82207% / 4.753902%)
3 : Three of a kind, 513 instances, (2.05118% / 2.112845%)
4 : Straight, 93 instances, (0.37185% / 0.392465%)
5 : Flush, 54 instances, (0.21591% / 0.19654%)
6 : Full house, 36 instances, (0.14394% / 0.144058%)
7 : Four of a kind, 6 instances, (0.02399% / 0.02401%)
8 : Straight flush, 5 instances, (0.01999% / 0.001385%)
9 : Royal flush, 5 instances, (0.01999% / 0.000154%)

Remarque : le premier pourcentage désigne la représentation des mains au sein du set d'entraînement, le second est la probabilité d'une main dans le domaine complet. Nous avons un ensemble de 25010 instances dans un domaine de 311,875,200.

Set de test :

0 : Nothing in hand, 501209 instances,(1.000063)
1 : One pair, 422498 instances,(0.999832)
2 : Two pairs, 47622 instances, (1.001746)
3 : Three of a kind, 21121 instances, (0.999647)
4 : Straight, 3885 instances, (0.989897)
5 : Flush, 1996 instances, (1.015569)
6 : Full house, 1424 instances, (0.988491)
7 : Four of a kind, 230 instances, (0.957934)
8 : Straight flush, 12 instances, (0.866426)

9 : Royal flush, 3 instances, (1.948052)

Remarque : la valeur à l'intérieur des parenthèses désigne la représentation de chaque main au sein du set de test comparé à sa représentation dans le domaine complet. 1 serait donc une représentation parfaite alors qu'un rapport <1 désignerait que la main est sous-représenté par contre si on a un rapport >1 c'est que la main et sur-représenté.
Nous avons un total d'1 million d'instances dans un domaine de 311,875,200.

Quelques statistiques :

Main	#de mains	Probabilité	#de combinaisons
Royal Flush	4	0.00000154	480
Straight Flush	36	0.00001385	4320
Four of a kind	624	0.0002401	74880
Full house	3744	0.00144058	449280
Flush	5108	0.0019654	612960
Straight	10200	0.00392464	1224000
Three of a kind	54912	0.02112845	6589440
Two pairs	123552	0.04753902	14826240
One pair	1098240	0.42256903	131788800
Nothing	1302540	0.50117739	156304800
Total	2598960	1.0	311875200

Remarque : le nombre de combinaisons représente le nombre d'instances dans le domaine.

Chapitre 3

Régression

Nous présentons dans ce chapitre une régression différente de ce que l'on a vu en cours et cela est dû au fait que notre base de données dispose de deux sets de donnés différents : un set d'entraînement et un autre de test.

Une autre difficulté qui se présente à nous est le grand nombre d'attribut pour chacune des cartes constituant une main du joueur.

C'est pour ces raison que nous allons procéder à plusieurs régressions.

3.1 Régression en fonction des données d'entraînement

On cherche à savoir si la main du joueur peut être prédite uniquement par le biais des variables que l'on a à disposition, à savoir la couleur et le rang de chaque carte.

3.1.1 Régression avec la variable Couleur

Du moment que l'on dispose de 5 couleurs pour chaque carte nous avons suivi le modèle suivant :

$$main = A0 + A1.Couleur1 + A2.Couleur2 + A3.Couleur3 + A4.Couleur4 + A5.couleur5$$

Sous forme matricielle cela nous donne :

$$\boldsymbol{z = X.A}$$

Ainsi pour chaque combinaison de 5 cartes nous disposons d'une main particulière avec n = 25010.

A travers le code disponible dans l'annexe nous avons trouvé les résultats suivant :

$$\mathrm{A} = \begin{pmatrix} 0.00575178 \\ -0.0006615 \\ 0.00273885 \\ -0.00104334 \\ -0.00376262 \\ 0.61349938 \end{pmatrix}$$

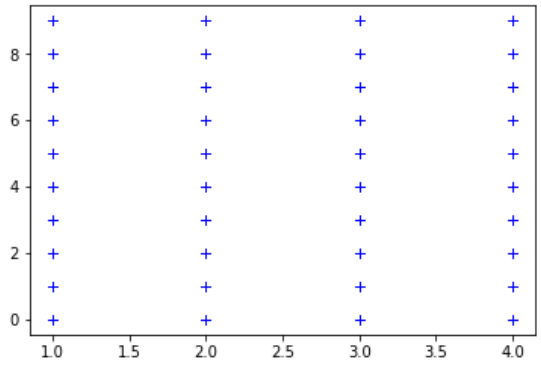


Figure 1 : Visualisation des observations

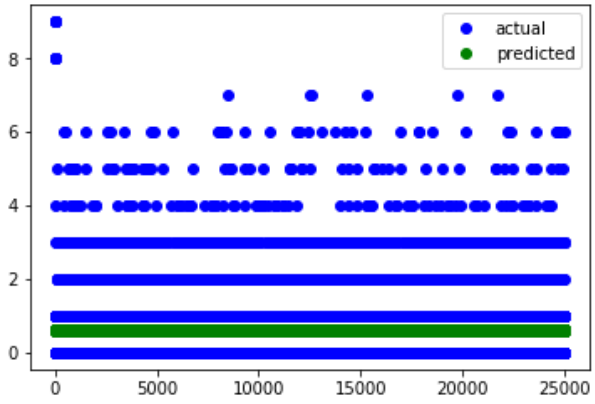


Figure 2 : Le modèle de régression

Le résultat trouvé de R2 est le suivant : 0.00011591523803333725 , ce qui est insuffisant vu que le coefficient de corrélation R2 est éloigné de 1. Nous allons cette fois-ci tenter une régression en ne prenant en compte que la variable du Rang.

3.1.2 Régression avec la variable Rang

$$\textbf{\textit{main = A0 + A1.Rang1 + A2.Rang2 + A3.Rang3 + A4.Rang4 + A5.Rang5}}$$

Nous savons suivi la même méthodologie qu'auparavant et nous avons trouvé les résultats suivant :

$$\mathrm{A} = \begin{pmatrix} 4.61573549e-04 \\ -1.09438694e-03 \\ -1.30307200e-03 \\ 2.10201789e-03 \\ -5.02226861e-04 \\ 6.23642603e-01 \end{pmatrix}$$

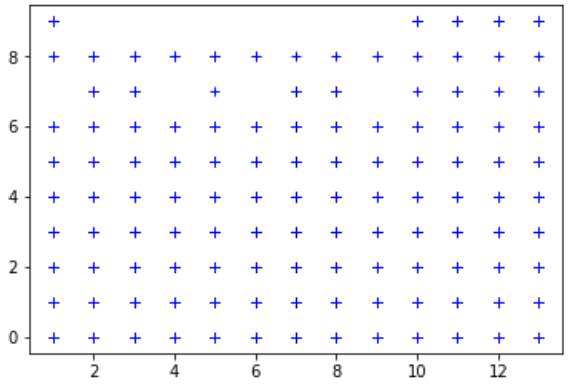


Figure 1 : Visualisation des observations

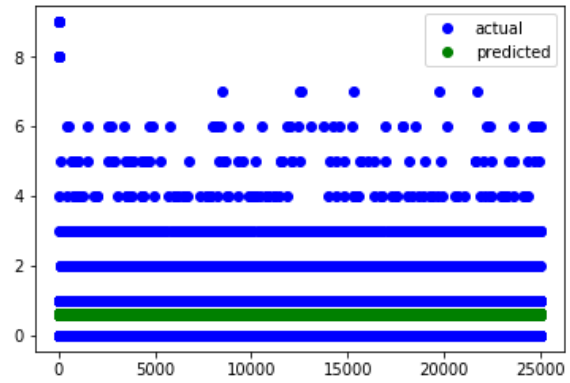


Figure 2 : Le modèle de régression

Le résultat trouvé de : R2=0.0001782345173007016 , ce qui prouve être insuffisant à nouveau, nous allons donc essayer la régression en prenant en compte les deux variables du Rang ainsi que de la Couleur.

3.1.3 Régression avec les deux variables

$$main = A0 + A1.Rang1 + A2.Couleur1 + A3.Rang2 + A4.Couleur3 + A5.Rang3$$

Suivant la méthodologie précédente nous retrouvons les résultats suivant :

$$\mathrm{A} = \begin{pmatrix} 5.73879583e-03 \\ 4.87012504e-04 \\ -7.74183002e-04 \\ -1.10620469e-03 \\ 2.77697145e-03 \\ -1.29880363e-03 \\ -1.02717504e-03 \\ 2.09735258e-03 \\ -3.77728781e-03 \\ -5.12257381e-04 \\ 6.16182280e-01 \end{pmatrix}$$

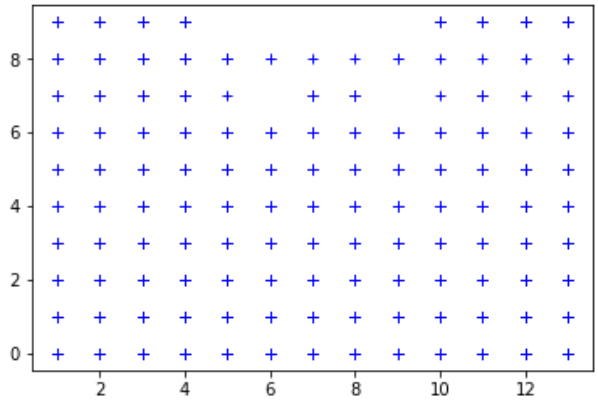


Figure 1 : Visualisation des observations

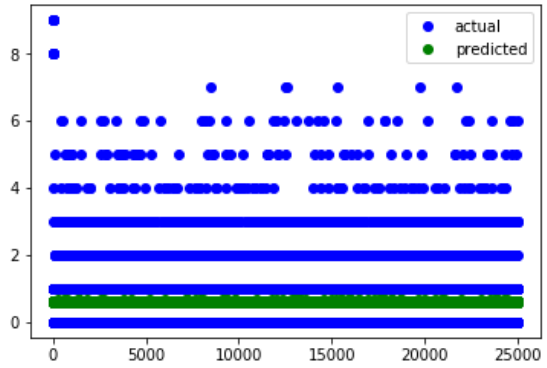


Figure 2 : Le modèle de régression

Le résultat trouvé de : R2=0.0.00029481160185029776 ,cela nous prouve que la régression est imprécise quelque soit la méthode suivie.

3.2 Régression en fonction des données d'entraînement et de test

Nous allons à présent changer de méthodologie, afin d'avoir des résultats plus cohérents en utilisant les données à notre disposition de manière plus efficace.

3.2.1 Distribution des données de test

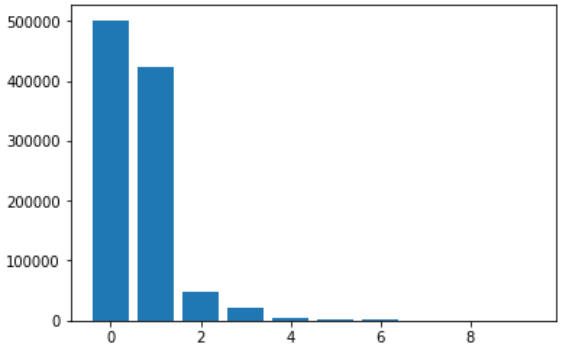


Figure 1 : Fréquences pour chaque main
Remarque : Les numéros désignant les mains sur l'axe de l'abscisse, sont dans l'ordre inverse , par exemple : 0 désigne la main la plus faible

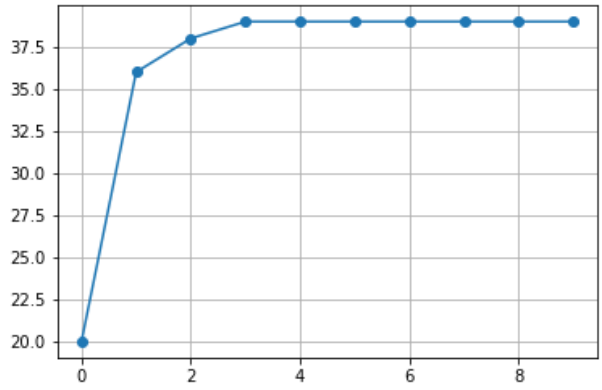


Figure 2 : Fonction de répartition empirique des données de test

3.2.2 Régression

Après une longue recherche nous avons trouvé un ouvrage traitant notre sujet par le biais du Machine Learning, le code sera disponible dans l'annexe.

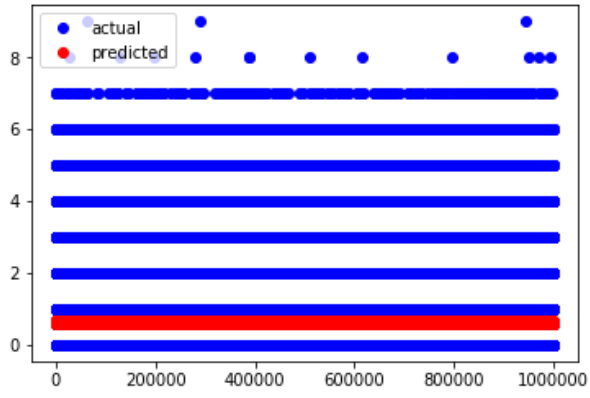


Figure : Modèle de régression en s'appuyant sur du Machine Learning

On remarque que l'on trouve une précision de 42 % ce qui nous dévoile que la régression n'est pas efficace afin de prédire les mains du joueur ce qui nous pousse à chercher des alternatives plus poussées au sein de l'intelligence artificielle.

Chapitre 4

Intelligence artificielle

Lors de ce chapitre nous avons utilisé un ouvrage dont le lien sera disponible dans l'annexe, afin de tenter de trouver une méthode capable de prédire de manière efficace la main d'un joueur en s'appuyant sur les données que nous avons vus et traités précédemment.

Nous avons choisis de présenter que la méthode la plus précise que nous avons pu trouver, le résultat de la prédiction est assez impressionnant.

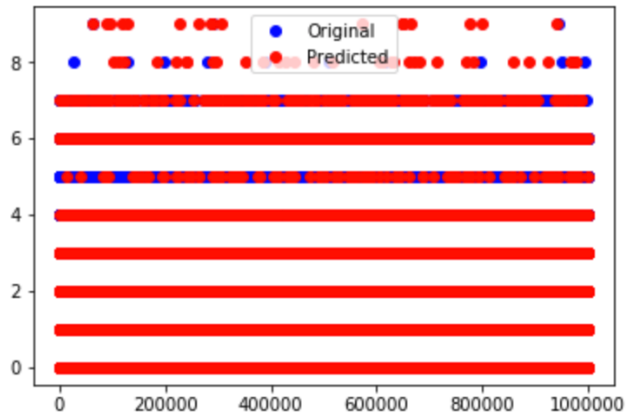
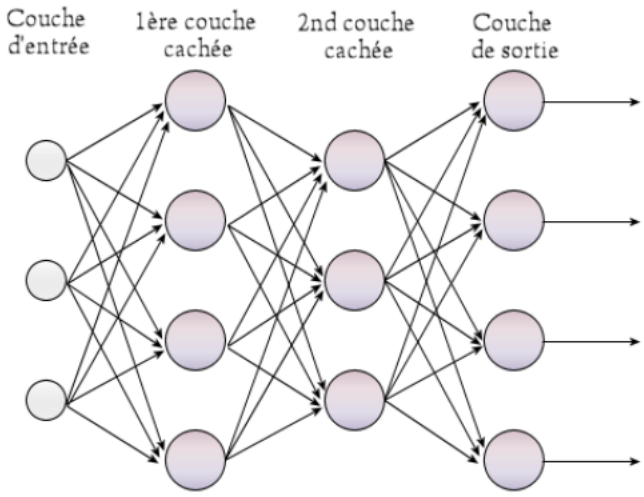


Figure : Modèle de régression en utilisant la méthode du Multi Layer Perceptron

Nous avons trouvé une précision de 95%, cela prouve que c'est la meilleure méthode afin de prédire la main d'un joueur de poker.

Remarques sur le Multi Layer Perceptron :

D'après Wikipedia : Le Multi Layer Perceptron est un type de réseau neuronal artificiel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement, il s'agit donc d'un réseau à propagation directe (feedforward). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global.



Pour plus d'informations sur comment passer d'une régression vers le Multi Layer Perceptron, il existe la documentation suivante :

Multi Layer Perceptron

Chapitre 5

Conclusion

Pour conclure, nous pouvons affirmer à partir de ce dossier et à l'aide de la régression ainsi que de méthodes basées sur du machine learning, qu'il est possible de prédire la main d'un joueur au Poker.Nous avons donc pu répondre à la question de la présence du hasard dans le poker, en montrant qu'en entraînant une machine à reconnaître les différentes combinaisons qu'il serait par la suite possible pour elle de deviner la main d'un joueur.Cela montre que l'apparition d'une IA capable de battre les meilleurs joueurs de Poker est bel et bien loin d'être une simple fantaisie.

De plus, le fait que le choix du thème du dossier soit libre, cela nous a permis de nous pencher sur une question qui nous intéressait fortement. Ce rapport fut aussi bénéfique car il nous a permis d'apprendre de nouvelles choses ainsi que d'approfondir nos connaissances en Python, et cela en mettant en oeuvre ce que l'on a vu en EC de M8.

Chapitre 6

Annexes

Notre programme python : Poker.ipynb

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt,pandas as pd,os
```

In [2]:
```
MesCartes=list()
for index in range(1, 6):
    MesCartes.extend(["Couleur"+str(index), "Rang"+str(index)]) #On a vu dans la description que chaque carte a deux at

MesCartes.append('classe') #la classe correspond a la main du joueur
```

In [3]:
```
train_input=os.path.abspath('../M8_2/Projet/poker-hand-training-true.csv')
test_input=os.path.abspath('../M8_2/Projet/poker-hand-testing.csv')
train_data = pd.read_csv(train_input, names=MesCartes)
test_data = pd.read_csv(test_input, names=MesCartes)
```

In [ ]:
```
train_data
```

In [ ]:
```
test_data
```

In [ ]:
```
#Séparer les classes des cartes
train_y = np.array(train_data['classe'])
train_x = np.array(train_data.drop('classe', 1))

test_y = np.array(test_data['classe'])
test_x = np.array(test_data.drop('classe', 1))
```

In [ ]:
```
test_y
```

In [ ]:
```
test_x
```

In [ ]:
```
My=np.unique(test_y) #Les différentes classes
My
```

In [ ]:
```
_, ni = np.unique(test_y, return_counts=True) #l'effectif de chaque classe
Ni = np.cumsum(ni)
Ni
```

In [ ]:
```
fi = ni/25010  # Fréquences
Fi = Ni/25010  # Fréquences cumulées

print('-------------------------------------------------------\n')
print(' modalites n_i N_i f_i F_i \n')
print('-------------------------------------µ-----------------\n')

for i in range(len(My)):
    print(My[i], ni[i],  Ni[i], fi[i], Fi[i])
```

In [ ]:
```
plt.grid(True)
plt.plot(My, Fi, 'o-')
```

In [ ]:
```
class myConfigs:
    features = 0
    classes = 0


config = myConfigs()
```

In [ ]:
```
config.Type_classementCarte = len(train_data.columns) - 1
config.classes = len(set(train_data['classe']))
```

In [ ]:
```
test_y_1 = list(test_y)
plt.bar(list(range(config.classes)), [test_y_1.count(int(x)) for x in range(config.classes)])
plt.show() #Les Fréquences de chaque classe(main de poker)
```

In [ ]:
```
train_x_1 = np.array(train_data.drop(['Rang1','Rang2','Rang3','Rang4','Rang5','classe'], 1))
train_x_1 #Pour avoir que lestypes de carte
```

In [ ]:
```
train_y.shape
```

In [ ]:
```
plt.plot(train_x_1,train_y,"b+")
plt.show()
```

In [ ]:
```
X = np.concatenate([train_x_1, np.ones((train_y.size, 1))], axis=1)
a = np.linalg.solve(X.T.dot(X), X.T.dot(train_y))
a
```

In [ ]:
```
plt.plot(train_y,'bo', label='actual')
z = X.dot(a)
plt.plot(z,'og',label='predicted')
plt.legend()
plt.show()
```

In [ ]:
```
e=train_y-z
e
```

In [ ]:
```
SCT = np.sum((train_y-np.mean(train_y))**2)
SCM = np.sum((X.dot(a)-np.mean(train_y))**2)
SCR = e.T.dot(e)
SCT, SCM, SCR
```

In [ ]:
```
R2 = 1 - SCR/SCT
R2
```

In [ ]:
```
train_x_2 = np.array(train_data.drop(['Couleur1','Couleur2','Couleur3','Couleur4','Couleur5','classe'], 1))
train_x_2
```

In [ ]:
```
plt.plot(train_x_2,train_y,"b+")
plt.show()
```

In [ ]:
```
X_1 = np.concatenate([train_x_2, np.ones((train_y.size, 1))], axis=1)
a_1 = np.linalg.solve(X_1.T.dot(X_1), X_1.T.dot(train_y))
a_1
```

In [ ]:
```
plt.plot(train_y,'bo', label='actual')
z_1= X_1.dot(a_1)
plt.plot(z_1,'og',label='predicted')
plt.legend()
plt.show()
```

In [ ]:
```
e_1=train_y-z_1
e_1
```

In [ ]:
```
SCT = np.sum((train_y-np.mean(train_y))**2)
SCM = np.sum((X_1.dot(a_1)-np.mean(train_y))**2)
SCR = e_1.T.dot(e_1)
SCT, SCM, SCR
```

```
In [ ]: R2_1 = 1 - SCR/SCT
        R2_1

In [ ]: plt.plot(train_x,train_y,"b+")
        plt.show()

In [ ]: X_2 = np.concatenate([train_x, np.ones((train_y.size, 1))], axis=1)
        a_2 = np.linalg.solve(X_2.T.dot(X_2), X_2.T.dot(train_y))
        a_2

In [ ]: plt.plot(train_y,'bo', label='actual')
        z_2= X_2.dot(a_2)
        plt.plot(z_2,'og',label='predicted')
        plt.legend()
        plt.show()

In [ ]: e_2=train_y-z_2
        e_2

In [ ]: SCT = np.sum((train_y-np.mean(train_y))**2)
        SCM = np.sum((X_2.dot(a_2)-np.mean(train_y))**2)
        SCR = e_2.T.dot(e_2)
        SCT, SCM, SCR

In [ ]: R2_2 = 1 - SCR/SCT
        R2_2
```

Lien vers le projet de machine learning :

Projet IA