



APRENTATGE AUTOMÀTIC APLICAT AL BLACKJACK

Pau Amorós Faro
2n Batxillerat B
Bernat Vidal
Treball de Recerca II

Avui en dia, els jocs de casino estan guanyant popularitat gràcies a les plataformes en línia. En conseqüència, el nombre de persones que es pregunten com es pot guanyar a la banca augmenta. El present estudi busca analitzar la viabilitat d'obtenir una estratègia de joc "perfecta" al blackjack i determinar quina és usant mètodes de programació de la família de les intel·ligències artificials. Com a resultat del projecte, s'exposen tres gràfiques, que demostren que es pot crear un agent que aprèn a jugar a blackjack i quina és la millor manera de fer-ho. Els valors en qüestió varen definir-se a partir de l'anàlisi dels resultats de diverses simulacions dels esdeveniments que caracteritzen una partida de blackjack. Després de validar la hipòtesi principal, es conclou que l'ús de l'aprenentatge automàtic és un mètode adient per deduir l'estratègia òptima.

Blackjack / Aprenentatge automàtic / IA / Monte Carlo / Python

Nowadays, casino games are gaining popularity thanks to online platforms. Consequently, the number of people wondering how to beat the house increases. This study aims to analyze the feasibility of obtaining a "perfect" playing strategy in blackjack and determine it using artificial intelligence programming methods. As a result of the project, three graphs are presented, demonstrating that an agent capable of learning to play blackjack can be created and which is the best way to play blackjack. The values in question were defined based on the analysis of the results from various simulations of the events that characterize a game of blackjack. After validating the main hypothesis, it is concluded that the use of machine learning is a suitable method for inferring the optimal strategy.

Blackjack / Reinforcement Learning / AI / Monte Carlo / Python

ÍNDEX

1. INTRODUCCIÓ I HIPÒTESIS DE PARTIDA	2
1.1 METODOLOGIA USADA	3
2. BLACKJACK	4
2.1 NORMES DEL JOC BLACKJACK	5
3. ANÀLISI MATEMÀTICA DEL BLACKJACK	8
3.1 ESTRATÈGIA ÒPTIMA/BÀSICA	9
3.1.2 Demanar una carta més i quedar-se igual.	9
3.1.3 Resum estratègia bàsica	11
3.2 SISTEMES PER A CONTAR CARTES	12
4. HISTÒRIA DE LA INTEL·LIGÈNCIA ARTIFICIAL	15
4.1 ACTUALITAT	16
4.2 FUTUR	16
5. LLENGUATGES O FORMA DE PROGRAMACIÓ	18
5.1 PYTHON	19
5.2 OPEN AI	20
5.3 JUPYTER LAB	21
6. MÈTODES DE PROGRAMACIÓ	22
6.1 MÈTODE DE DECISIONS MARKOV	22
6.2. Q-LEARNING	22
6.3 MÈTODE MONTECARLO	23
7. GENERACIÓ DEL PROGRAMA	25
7.1 PROGRAMA INICIAL	25
7.2 PROGRAMA DEFINITIU	28
8. ANÀLISI DE RESULTATS	30
9. CONCLUSIÓ	31
10. BIBLIOGRAFIA	32
— Sobre el joc blackjack.	32
— Sobre les matemàtiques de jugar al blackjack.	32
— Sobre la intel·ligència artificial.	33
— Sobre llenguatges de programació i metodologies similars.	34
— Per aprendre a programar i analitzar el codi.	35
— Annex	36

1. INTRODUCCIÓ I HIPÒTESIS DE PARTIDA

L'objectiu d'aquest treball és investigar com es pot utilitzar la intel·ligència artificial per a millorar les teves probabilitats de guanyar en un joc de cartes. El joc objecte d'aquest treball és el blackjack. En els últims anys, aquest joc ha esdevingut una de les maneres més directes d'il·lustrar diferents conceptes i teories matemàtiques relacionats amb l'aprenentatge automàtic. Per tal causa, aquest treball planteja diverses estratègies de joc, basades en les propietats de les matemàtiques tradicionals i de diferents estimacions a partir d'anàlisis de la mateixa índole que el mètode Montecarlo o *Q-learning*, que explicaré més endavant en el present treball.

Dintre d'aquest marc i amb els meus coneixements inicials ens hem plantejat diferents hipòtesis de partida que pretenc validar o refutar:

- **H1.** La banca sempre té l'avantatge a l'hora de guanyar al blackjack.
- **H2.** L'aproximació a l'estratègia òptima que ens donarà una IA té una probabilitat més alta de guanyar el joc que l'estratègia bàsica.
- **H3.** Es pot programar una intel·ligència artificial que ens calculi l'estratègia òptima del blackjack.

En definitiva, la motivació del treball és fer un estudi que permeti validar o refutar les hipòtesis de partida. D'altra banda, l'objectiu general del treball, que està directament relacionat amb el joc de cartes anomenat blackjack, rau a programar una intel·ligència artificial perquè plantegi una estratègia de joc *òptima* (en apartats posteriors es donarà context sobre el significat d'aquest adjectiu i les seves connotacions). Aquesta última ens ajudarà a analitzar diferents qüestions i a extreure conclusions. Bàsicament, pensarem sobre el blackjack des d'un punt de vista matemàtic. És a dir, partirem de les següents preguntes:

- Com hauria de jugar per maximitzar la meva expectació, és a dir, les meves probabilitats de guanyar?

-Quina és la màxima expectació?

La resposta a la primera pregunta determina la resposta a la segona i la resposta a la segona determina si matemàticament és interessant jugar.

1.1 METODOLOGIA USADA

Els objectius anteriors comporten diverses tasques que es poden classificar en dos tipus segons la metodologia emprada:

M1. Cerca bibliogràfica, documentació i redacció. La cerca bibliogràfica de documents digitals s'ha fet principalment a través d'internet i també s'han aconseguit llibres en format digital i paper. A l'últim apartat del treball, s'han recopilat els enllaços web d'interès (pàgines informatives, cursos i vídeos) i els documents (llibres). Quant a la redacció del treball, s'ha fet en l'editor *Documents de Google*.

M2. Programació i simulacions per computador. S'han realitzat dos programes informàtics, els quals es poden classificar en dos tipus. El primer (programat en Python) calcula l'expectació màxima que un pot obtenir a una partida de blackjack. A més, a partir d'un nombre de partides, el programa retorna un valor que mostra el tant per cent de partides guanyades, perdudes i empatades. El segon (també programat en Python) simula les diferents estratègies de joc i determina quina és millor. Concretament, he utilitzat les IDE (entorn de desenvolupament integrat) Jupyter i PY Charm.

2. BLACKJACK

Segons *La historia del blackjack (2022)*¹, a partir de l'aparició dels primers jocs de cartes de la mà dels xinesos, va començar a sorgir tota una cultura al seu voltant amb cada vegada un major nombre d'adeptes. Un joc directament relacionat amb aquest fenomen és el blackjack. Es desconeix l'origen exacte del blackjack, encara que es creu que prové de la vint-i-una o, com s'exposa a *History of Blackjack (2022)*², la trenta-una (joc espanyol popular en l'antiguitat on l'objectiu era arribar a trenta-u amb un mínim de tres cartes). Malgrat això, l'origen d'aquest joc és discutit i les primeres referències se situen en l'obra de Miguel de Cervantes anomenada *Rinconete y Cortadillo*. En aquesta novel·la, els dos personatges es dediquen a jugar a la vint-i-una, joc del qual es dedueix que era conegut a la fi del segle XVI.

A diferència de la majoria dels jocs de casino, el blackjack va convertir-se en un èxit sense precedents, pel fet que la possibilitat de contar cartes va suposar un canvi radical a l'hora de competir contra el crupier i augmentar les possibilitats de guanyar. Malgrat que avui en dia ja no es pot fer, com bé s'observa a *Learn How to Play Blackjack*³: “A llarg termini (milers de mans), l'avantatge de la casa és el percentatge de cada aposta que el casino espera mantenir matemàticament. Aquest número pot arribar al 0,5% (...) per a altres jocs (com el blackjack).” La raó per la qual va ser possible realitzar càlculs matemàtics per a superar l'avantatge de la banca va ser que es jugava amb una única baralla de 52 cartes, descartant-se segons anaven sortint i continuant la partida sense realitzar mescles fins a finalitzar-la. D'aquesta manera, el blackjack es va convertir en un joc aritmètic en el qual el que traguessin els jugadors en una mà depenia de les cartes que haguessin aparegut prèviament. Tota aquesta informació, des del meu punt de vista, fa que sigui un objecte d'estudi perfecte per a aquest treball.

¹ Gonzalves, R. (10 de juny del 2022). *La historia del blackjack*. Bestonlinecasino.com. <https://www.bestonlinecasino.com/es/juegos/blackjack/historia/>

² Wintle, S. (27 d'octubre del 2022). *History of Blackjack*. The world of playing cards. <https://www.wopc.co.uk/the-history-of-playing-cards/blackjack/blackjack>

³ Gambling sites. (Abril 2023). *Learn How to Play Blackjack*. <https://www.gamblingsites.org/casino/blackjack/beginners-guide/>

2.1 NORMES DEL JOC BLACKJACK

Donades les circumstàncies del treball en qüestió, és d'una preable importància tenir un coneixement bàsic amb relació a les normes i objectiu del blackjack. A partir de la informació extreta de *Learn to play Blackjack (s.d)*⁴ i *Reglas del blackjack (2015)*⁵, es pot dir que a aquest joc es juga amb una baralla francesa de 52 cartes sense comodins (jòquers) i, depenent del casino, la quantitat de baralles es trobarà entre una i vuit, sent sis el nombre més freqüent. Habitualment, hi ha algunes confusions entre nous jugadors sobre la finalitat del joc en qüestió. Elementalment, quan un juga a blackjack té la meta de guanyar a la banca.

Aconseguir aquest propòsit es pot fer de tres maneres diferents. En primer lloc, traient un valor de mà que sigui superior al del crupier. També, assolint que aquest tregui un valor que sobrepassi el límit de vint-i-u. Finalment, pots guanyar també si obtens un valor de vint-i-u a les teves primeres dues cartes, quan el crupier no ho ha fet. Això no obstant, les maneres descrites a les línies anteriors descriuen un estat directament relacionat amb la victòria, com a tots els jocs, també es pot perdre. A aquest estat arribes quan la teva mà excedeix el valor màxim permès de vint-i-u. Altrament, si el crupier acumula un valor superior al teu que sigui menor o igual a vint-i-u al final de la ronda, perds. Finalment, des del meu punt de vista, considero important mencionar que les cartes dels altres jugadors a la mateixa taula que tu no tenen res a veure amb guanyar la partida, com ocorre a altres jocs molt coneguts com podria ser el pòquer. Per practicitat, es pot resumir en el fet que ets només tu contra la banca.

Al paràgraf anterior s'ha parlat sobre com es pot guanyar o perdre i, si l'hem llegit, ens haurem adonat que la puntuació de les cartes és crucial en aquest joc. Del 2 fins al 10 les cartes tenen el valor que es mostra, per exemple el 2 conta com a dos. El rei, la reina i la sota tenen un valor de deu punts. Finalment, ens trobem amb l'as, el qual és un cas especial. Aquest tindrà el valor d'u o d'onze, segons li convingui al jugador.

⁴ Bicyclecards. (s.d). *Learn to play Blackjack*. <https://bicyclecards.com/how-to-play/blackjack/>

⁵ Abbran juego. (14 de gener del 2015). *Reglas del blackjack*. <https://www.luckia.es/blog/reglas-del-blackjack/>



Figura 1 - Puntuacions de les cartes al blackjack. Jones, C. (s.d). *How to play blackjack*.

<https://www.blackjackapprenticeship.com/how-to-play-blackjack/>

Al blackjack, s'hi juga normalment a una taula semicircular que pot acomodar una quantitat variable de jugadors. Les taules més comunes tenen una capacitat per a set jugadors. El crupier s'asseu darrere la taula on es guarden les fitxes, mentre que els jugadors, a l'altra banda. Aquest joc presenta diverses variants, encara que una ronda estàndard, com es diu a *Cómo jugar Blackjack*(s.d)⁶, segueix les següents passes:

1. El jugador compra fitxes: abans de poder jugar a la taula necessites fitxes.

Aquestes fitxes representen diners i son clau per a avaluar l'èxit de l'estratègia que un segueix.

2. El jugador aposta: al principi de la ronda la primera cosa que es fa és apostar. Als EUA, la majoria de les taules requereixen un mínim de 5 \$ per mà, tot i que aquesta quantitat pot variar.

3. El crupier reparteix les cartes: després de realitzar l'aposta, el crupier repartirà una carta cap amunt a cada jugador i tot seguit una cap avall per a ell mateix. A continuació, repartirà una altra carta cap amunt a tothom (a ell també). És a dir, cada jugador té dues cartes visibles, però el crupier té una visible i una girada.

4. El jugador decideix com jugar: se sumen els valors de les dues cartes que tens i comences amb un valor d'entre quatre i vint-i-u. Si has rebut una carta amb valor deu i un as, significa que tens blackjack. En aquest cas, et paguen una proporció de 3:2 en

⁶ Mejor casino online.mx. (s.d). *Cómo jugar Blackjack*. (s.d). <https://mejorcasinonline.mx/como-jugar-blackjack/>

relació amb la teva aposta, a condició que la banca no tingui blackjack. Si en té, ni perds la teva aposta, ni guanyes res. Ara bé, aquest cas descrit no és massa comú. En conseqüència, hauràs de prendre una de les següents decisions:

4.1. Parar-se. Si les teves primeres dues cartes són acceptables, pots no demanar cap més.

4.2. Tirada. Si vols més cartes per a millorar el teu total, el crupier et donarà més; una cada vegada fins que et passis o et paris. No hi ha límit a la quantitat de cartes que pots demanar. El jugador sempre juga amb totes les cartes descobertes.

4.3. Dobla. Si tens un total que et garanteix avantatge, però necessites una carta addicional, pots doblar la teva aposta inicial i rebre exactament una única carta més.

4.4. Divideix. Quan reps una parella (dues cartes amb el mateix valor) tens la possibilitat de fer una segona aposta i el crupier dividirà les dues cartes en dues mans. En aquest cas jugaràs amb dues mans, en lloc d'una.

4.5. Rendir-se. Si no t'agrada la teva mà inicial, tens l'opció de rendir-te a canvi de la meitat de la teva aposta inicial.

5. El crupier treu la seva mà: si no t'has passat ni t'has rendit, és el moment que la banca jugui la seva mà. Primerament, el crupier començarà girant la seva carta que estava cap avall i calculant el valor que té. Si és més gran o igual a disset, es parerà automàticament. Si no, agafarà cartes addicionals. La banca ha de jugar la seva mà de la mateixa manera cada vegada. Per a la banca, el valor de l'as no varia, no el pot triar com el jugador; les normes de la taula ens indicaran si per la banca l'as té un valor d'1 o d'11.

6. Desemborsament: Una vegada tothom ha acabat el seu torn, una de les següents dues coses haurà passat: la banca s'haurà passat (pagarà una vegada l'aposta a cada mà que estigui en joc) o la banca tindrà un resultat d'entre disset i vint-i-u. En aquest cas, qui tingui un valor més elevat que la banca guanya. Si hi ha un empat, ni es guanya ni es perd.

3. ANÀLISI MATEMÀTICA DEL BLACKJACK

A *Blackjack: the math behind the cards* (2019, pàgina 2)⁷, se'ns explica que donades les característiques del blackjack, és lògic que les matemàtiques tinguin paper fonamental a l'hora de realitzar un estudi sobre aquest. La pregunta que actua com a motor conductor de l'estudi en qüestió és si es pot guanyar o no a la banca. Un símil molt encisador i que, des del meu punt de vista ajuda a aclarir aquesta interrogant, és el que es fa amb la ruleta: es pot guanyar al blackjack, però no a la ruleta? A finals del segle XIX, l'enginyer anglès Joseph Hobson Jagger es va preguntar si les ruletes dels casinos tindrien tendències a causa de defectes de fabricació i desgast dels seus elements. A continuació, els seus col·laboradors varen observar els nombres que sortien en cada ruleta del casino de Montecarlo. La conclusió que va poder extreure va ser que totes estaven en bon estat. Ara bé, hi havia una on no passava això. Mitjançant aquesta informació, va aconseguir obtenir una petita fortuna.

Quan passem de la ruleta al blackjack, contemplem un panorama diferent. La bolla de la ruleta no té memòria. El mateix ocorre, com s'exposa a *The Maths Behind Blackjack* (2017)⁸, si en comptes de trenta-set números tenim els cinquanta-dos naips d'una baralla francesa. Si traiem una carta cada vegada i després la retornem a la baralla (la qual ha de ser mesclada abans de treure'n una altra), estirem davant d'una elecció amb reemplaçament. És a dir, el reemplaçament en llançar una pilota a la ruleta o un parell de daus sobre el tapet verd és implícit (hi està inclòs). Congruentment, tractem amb successos aleatòriament independents. Ara bé, l'estat canvia radicalment si un naip no és retornat a la baralla una vegada mostrat, sinó col·locat en una pila de descart. L'explicació del següent exemple extret de *The mathematics of gambling* (2022)⁹ il·lustra aquest fet. Si ha sortit el 2 de cors, està clar que ja no tornarà a sortir fins que esgotem la baralla sencera, prenguem les cartes de la pila en la qual les hem anat dipositant, les barregem i tornem a treure cartes. Davant aquesta problemàtica, diversos matemàtics varen preguntar-se si es podria guanyar al blackjack tenint en compte que la probabilitat que tant el jugador com el crupier tinguin una determinada puntuació en cada mà que depèn dels naips apareguts prèviament i dipositats

⁷ Blanchard, H. (13 de maig de 2019). *Blackjack: the math behind the cards*. [Tesi doctoral, Louisiana Tech University]. viewcontent.cgi. <https://acortar.link/wKeHSW>

⁸ Casino Maths. (15 de març de 2017). *The Maths Behind Blackjack* [vídeo]. Youtube. <https://www.youtube.com/watch?v=YDBeWh5hgP4>,

⁹ Not only Science!. (25 de gener de 2022). *The mathematics of gambling: how information theory changed blackjack and other games of chance* [vídeo]. Youtube. <https://www.youtube.com/watch?v=d038pCViGpA>

en la pila de descart. Gràcies a un gran nombre de simulacions i hores d'estudi, van concloure que el crupier es veu beneficiat quan hi ha un gran nombre de cartes baixes que no havien sortit i el jugador al contrari.

3.1 ESTRATÈGIA ÒPTIMA/BÀSICA

A *Estudio de la estrategia óptima para el black-jack* (1985, pàgina 97)¹⁰ es menciona la característica principal que ha de tenir l'estratègia òptima, que és l'estratègia que maximitza la mitjana de victòries d'un jugador, o l'expectació d'aquest, jugant una mà contra una baralla completa de cartes. Així doncs, amb un nombre donat de baralles i unes normes establertes només hi pot haver una "estratègia òptima", encara que potser hi ha diferents versions, les quals són lleugerament errònies. És inclús imaginable i probable, que ningú, experts inclosos, sàpiga amb precisió quina és l'estratègia bàsica si busquem que la definició inclogui instruccions sobre com jugar la segona i següents cartes dependentment de quines cartes s'hagin utilitzat a les parts anteriors de les partides. Per exemple, suposem que dividim vuits contra el deu del crupier, passant-nos a la primera mà (la puntuació total de les nostres cartes és superior a 21) i obtenint (8,2,2,2) a la segona. Ràpidament, demanem una carta més o ens quedem amb el que tenim? L'estratègia bàsica, llavors, constitueix un conjunt de normes en relació amb les decisions tenint en compte totes les possibles eleccions que un jugador pot trobar-se, però sense cap referència a les cartes d'altres jugadors o cartes emprades a una ronda prèvia abans de tornar a mesclar la baralla. Aquestes eleccions són: dividir o no dividir, doblar l'aposta o no doblar l'aposta i demanar una carta més o quedar-se com un està. Algunes són bastant evidents, com per exemple, treure una carta quan tens un total de sis, mai demanar una carta més amb vint i no dividir una parella de cinc. Però quin procediment ha de ser usat per a considerar l'acció correcta en casos més marginals?

3.1.2 Demanar una carta més i quedar-se igual.

Per a il·lustrar-ho amb un exemple, extret de *The Theory of Blackjack: The complete Card Counter's Guide to the Casino* (1999, pàgines 13-15)¹¹, considerem l'elecció de si treure una

¹⁰ Departamento de Estadística e Investigación Operativa Facultad Ciencias Matemáticas UCM. (s.d.). *Estudio de la estrategia óptima para el black-jack* [conjunt de dades]. Universidad Complutense de Madrid.

<https://eprints.ucm.es/id/eprint/16268/1/Tejada18.pdf>

¹¹ Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press

carta més o quedar-se amb (T,6)* [La lletra es farà servir al llarg del treball com a un símbol per a qualsevol carta de valor deu] contra el 9 del crupier. Mentre que relativament entre les eleccions més simples d'analitzar, podem veure que la resolució precisa del tema requereix una quantitat extraordinària d'aritmètica (estudi de les relacions o propietats dels nombres naturals). Si ens quedem amb 16, guanyarem o perdrem segons si el crupier es passa o no; no hi haurà empat. En conseqüència, les probabilitats exactes poden ser trobades buscant les 556 seqüències de treure cartes diferents i considerant els camins en relació amb les probabilitats que es donin. Algunes seqüències que passarien són:

<u>Drawing Sequence</u>	<u>Probability</u>
9,2,A,T	$4/49 \times 4/48 \times 15/47$
9,7,6	$4/49 \times 3/48$
9,2,A,A,A,A,8	$4/49 \times 4/48 \times 3/47 \times 2/46 \times 1/45 \times 4/44$

Figura 2 - Càlculs de probabilitats segons les cartes que surten. Pàgina 13 de Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press.

Una vegada elaborat l'anàlisi, trobem que les probabilitats exactes del crupier de passar-se són 0.2304. Posteriorment, ens interessa determinar l'expectació matemàtica associada amb quedar-se com un està. Tenint en compte que guanyem 0.2304 apostes per a cada 0.769 que perdem, se suposa que el jugador perdrà 0.5392 per cada euro esperant que el crupier es passi.

Ara bé, observar el que passa quan es vol una carta més també és necessari per decidir què fer. Tot i que fer aquesta predicció no és tan senzill, car per a cada de les cinc cartes que podem treure sense passar-nos (A,2,3,4,5), les probabilitats de la banca d'assolir diferents totals s'ha de determinar separatament. Per exemple, si el jugador treu un 2 (té un valor total de 18 seguint l'exemple) i es vol quedar així. Quines són les nostres probabilitats condicionades? La solució pot trobar-se jugant la mà del crupier amb un residu de quaranta-vuit cartes. Una vegada s'ha fet això, ja no estem interessats només en les probabilitats que la banca superi el límit establert (21), sinó també cada quant surt 17,18,19,20 i 21. Els resultats es troben a la següent taula.

Player's Cards	17	18	19	20	21	Bust
T,6						.2304
T,6,A	.1259	.1093	.3576	.1076	.0636	.2360
T,6,2	.1248	.1045	.3553	.1265	.0565	.2324
T,6,3	.1244	.1060	.3532	.1265	.0621	.2278
T,6,4	.1257	.1054	.3546	.1243	.0619	.2281
T,6,5	.1252	.1060	.3549	.1256	.0598	.2285

Figura 3 - Taula amb les probabilitats de diversos esdeveniments davant determinats escenaris. Pàgina 14 de Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press.

Aquest exemple mostra la magnitud dels càlculs que es realitzen per a cada escenari possible, els quals s'utilitzen per a determinar una estratègia bàsica. Aquestes operacions i metodologies s'apliquen a tots els aspectes del joc. Un resum és el següent:

3.1.3 Resum estratègia bàsica

A partir de la síntesi de l'estratègia bàsica extreta i contrastada de *Estudio de la estrategia óptima para el black-jack (1985, pàgines 103-104)*¹² i *Blackjack answers (2017)*¹³ establim aquestes directrius bàsiques.

- **Demandar una carta més i quedar-se igual:** mai demanis una carta més amb 17 o més.

Fes-ho quan tinguis un valor d'entre 12-16 contra cartes de valor alt (7, 8, 9, T, A).

- **Treure o no treure cartes amb mà fluixa:** sempre demana una amb 17 i queda igual amb 18 (sempre que no t'enfrontis a 9 o T. Llavors, demana una més.).

- **Dividir parelles:** mai divideixis (4,4), (5,5), o (T,T), però sempre divideix (8,8) i (A,A).

Parteix (9,9) contra els nombres que comprenen els intervals 2-6 i 8-9. Divideix les altres combinacions possibles contra un nombre entre 2-7. Ara bé, a la situació exposada a l'oració anterior, demana una carta més amb (6,6) contra 7, (2,2) i (3,3) contra 2, i (3,3) contra 3.

¹² Departamento de Estadística e Investigación Operativa Facultad Ciencias Matemáticas UCM. (1985). *Estudio de la estrategia óptima para el black-jack* [conjunt de dades]. Universidad Complutense de Madrid. <https://eprints.ucm.es/id/eprint/16268/1/Tejada18.pdf>

¹³ Casino Maths. (15 de març de 2017). *Blackjack answers* [vídeo]. Youtube. <https://www.youtube.com/watch?v=kFjYBMU5q64&t=0s>

- **Doblar aposta (mà fluixa):** dobla quan les teves cartes sumin un total d'entre 13-18 contra 4,5,6. Dobla amb disset contra 2,3. També, amb un valor de 18 contra tres. Per acabar, dobla 19 contra 6.

- **Doblar aposta (mà forta):** sempre dobla amb una puntuació total d'onze. Dobla amb deu, a condició que el crupier tingui un nombre entre 2-6. Finalment, dobla amb vuit contra 5 i 6.

3.2 SISTEMES PER A CONTAR CARTES

El nombre de càlculs matemàtics que poden realitzar-se en relació amb el blackjack són tan extensos que no podrien abastar-se completament a un treball on el límit de pàgines és trenta. Amb això en ment i després de reflexionar, he arribat a la conclusió que, tenint en compte el tema principal d'aquest treball, és d'un gran interès parlar sobre sistemes de contar cartes. He arribat a aquesta resolució, car contar cartes està associat amb un bon resultat a aquest tipus de jocs.

Una manera senzilla d'il·lustrar el ventall de decisions que s'han de tenir en compte quan es pretén contar cartes mencionada a *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino* (1999, pàgines 40-43)¹⁴ és la següent. Imaginem que es reparteix una carta cada vegada d'una baralla, sense tornar a mesclar-la després. Abans que cada carta es giri el jugador té la possibilitat d'apostar que la pròxima carta serà vermella. Amb una baralla sencera, una vegada la primera carta s'ha girat, el que decideix si apostar estirà a una situació avantatjosa la meitat de les vegades. Una estratègia òptima per a contar cartes amb aquestes característiques és òbvia. Això limita l'interès que pot generar l'exemple, motiu pel qual m'agradaria proposar-ne un altre on el jugador no pot distingir els colors (és daltònic). Llavors, un pot imaginar-se diferents mètodes per a obtenir benefici.

Una idea seria buscar un excés de cors sobre les espases a les cartes que no s'han jugat. Amb aquesta condició, el jugador tindria una precisió aproximada a la meitat, malgrat que no sempre tindria avantatge. **L'anomenarem sistema A.**

¹⁴ Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press

Una altra opció és contar els diamants, monitorant la proporció dels diamants a la baralla i apostant quan els diamants siguin més d'un quart de les cartes restants. **L'anomenarem sistema B.**

Finalment, podem fixar-nos en l'equilibri relatiu entre tres grups, com per exemple trèvols, cors i diamants. Tenint en compte que normalment hi ha el doble de cartes vermelles a trèvols, la baralla hauria de ser favorable quan les cartes vermelles que quedin siguin més nombroses que el doble dels trèvols. **L'anomenarem sistema C.**

Aquests tres mètodes per a contar cartes es poden dur a terme assignant valors a les cartes que encara es troben a la baralla, valors els quals serien el contrari en símbols algebraics als nombres contats i contínuament afegits mentre les cartes es treuen de la baralla. Els valors apropiats per als tres sistemes mencionats anteriorment es poden resumir en la següent taula:

<u>System</u>	<u>Spade</u>	<u>Heart</u>	<u>Diamond</u>	<u>Club</u>	<u>Sum of Squares</u>	<u>Correlation</u>
A	-1	1	0	0	2	.707
B	-1	-1	3	-1	12	.577
C	0	1	1	-2	6	.816
Payoffs	-1	1	1	-1	4	1.000

Figura 4 - Valors apropiats per als tres sistemes. Pàgina 41 de Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press.

Durant la meua investigació per a esbrinar com diferents sistemes per a contar cartes serien capaços d'interpretar i analitzar una baralla de blackjack, vaig decidir observar exhaustivament el joc més simple de tots (mencionat i explicar al primer paràgraf). En la meua opinió, l'exemple té dos avantatges. En primer lloc, es podria programar un ordinador per a determinar amb precisió quan es podria guanyar amb cada baralla amb els tres sistemes. De la mateixa forma, saber el color òptim evitant errors de mostra provocats per la realització d'una estratègia basada en els pagaments, perquè s'utilitzarien probabilitats exactes. El segon avantatge és que l'estructura simplista pot fer evident la decisió d'analitzar el més complex joc de blackjack.

Fent servir els resultats del programa realitzat per l'autor del llibre *The Blackjack theory: The Complete Card Counter's Guide to the Casino*, posat en pràctica el 17 de juny del 1974 a les 18:24 h i amb un temps d'execució de 2.94 s com a punt de partida, es mostra que el

coeficient de correlació entre els valors numèrics dels sistemes de contar cartes, explicats anteriorment, i la regulació establerta pels pagaments del blackjack. Això s'aconsegueix dividint la suma dels productes dels valors respectius assignats a cada tipus de carta, per l'arrel quadrada del producte de la suma dels quadrats dels valors del sistema per contar cartes, i la suma dels quadrats dels pagaments. A més, és molt propera a la quantitat relativa guanyada per cada un dels mètodes.

Cards	System C	Payoffs	C ²	P ²	C·P
Spades	0	-1	0	1	0
Hearts	1	1	1	1	1
Diamonds	1	1	1	1	1
Clubs	-2	-1	4	1	2
			6	4	4

$$\text{Correlation} = \frac{\sum C_i P_i}{\sqrt{\sum C_i^2 \cdot \sum P_i^2}} = \frac{4}{\sqrt{6 \cdot 4}} = \frac{4}{\sqrt{24}} = .816$$

Figura 5 - coeficient de correlació entre els valors numèrics dels sistemes de contar cartes explicats anteriorment i la regulació establerta pels pagaments del blackjack. Pàgina 42 de Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press.

Number of Cards Left	A	B	C
9	.676	.569	.768
18	.718	.607	.833
27	.691	.574	.804
36	.719	.573	.842
45	.669	.569	.764

Figura 6 - Quantita relativa guanyada segons els mètodes. Pàgina 42 de Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press.

Amb tot això es podria dir que existeix una dimensió més elevada relacionada directament amb l'anàlisi. Dimensió que defineix una relació de dues variables entre l'eficiència d'un sistema de contar cartes i la seva correlació amb la situació particular donada. El que aprenem de les matemàtiques, llavors, és que l'eficiència està directament relacionada i, en alguns casos igualada, a la correlació entre els valors dels punts del sistema de comptar cartes i les recompenses de les cartes individuals de la situació de blackjack considerada. És a dir, és imprescindible que un sistema de contar cartes funcioni bé si volem obtenir resultats correctes.

4. HISTÒRIA DE LA INTEL·LIGÈNCIA ARTIFICIAL

Segons es resumeix de *What is artificial intelligence: Types, History or Future (2023)*¹⁵. La història de la intel·ligència artificial (IA) va començar en l'antiguitat, amb mites, històries i rumors d'éssers artificials dotats d'intel·ligència o consciència per mestres artesans. Les llavors de la IA moderna van ser plantades per filòsofs que van intentar descriure el procés del pensament humà com la manipulació mecànica dels símbols. Aquest treball va culminar amb la invenció de l'ordinador digital programable en la dècada de 1940, una màquina basada en l'essència abstracta del raonament matemàtic. Aquest dispositiu i les idees que hi havia al darrere van inspirar un grapat de científics a començar a discutir seriosament la possibilitat de construir un cervell electrònic.

El camp de la investigació de la IA, com es diu a *History of Artificial Intelligence (s.d)*¹⁶, es va fundar en un taller celebrat al campus del Dartmouth College (EUA), durant l'estiu de 1956. Els que van assistir es convertiren en els líders de la investigació de la IA durant dècades. Molts d'ells van predir que una màquina tan intel·ligent com un ésser humà existiria en no més d'una generació, i se'ls va donar milions de dòlars per fer realitat aquesta visió.

Un temps més tard, es va fer obvi que els desenvolupadors comercials i els investigadors havien subestimat greument la dificultat del projecte. El 1974, en resposta a les crítiques de James Lighthill i la pressió del congrés, els governs dels Estats Units i el Regne Unit van deixar de finançar la investigació no dirigida sobre la intel·ligència artificial, i el període que va seguir es coneixeria més tard com un "hivern d'IA". Set anys més tard, una iniciativa del govern japonès va inspirar entitats governamentals i la indústria a proporcionar el camp d'investigació milers de milions de dòlars, però a finals dels anys vuitanta els inversors es van desil·lusionar i van retirar el capital de nou.

La inversió i l'interès en la IA va créixer en les primeres dècades del segle XXI, quan l'aprenentatge automàtic es va aplicar amb èxit a molts problemes en el món acadèmic i la

¹⁵ Kelley, K. (9 de març de 2023). *What is artificial intelligence: Types, History or Future*. Simplilearn. <https://acortar.link/C5Q8lm>

¹⁶ Java T point. (s.d.). *History of Artificial Intelligence*. <https://www.javatpoint.com/history-of-artificial-intelligence>.

indústria. Tot això gràcies als nous mètodes, l'aplicació de maquinari d'ordinador potent i la col·lecció d'immensos conjunts de dades.

4.1 ACTUALITAT

No ens hem tornat més hàbils en la programació d'intel·ligències artificials. Així doncs, què ha canviat? Resulta que el límit fonamental de l'emmagatzematge d'ordinadors que ens frenava fa trenta anys ja no és un problema. La Llei de Moore, que estima que la memòria i velocitat dels ordinadors es dupliquen cada any, s'ha començat a complir i, en molts casos, supera les nostres expectatives. Així és precisament com Deep Blue va ser capaç de derrotar Gary Kasparov el 1997 i Alpha Go de Google va ser també capaç de derrotar el campió xinès de Go, Ke Jie l'any 2017.

A l'article *¿Qué es la ia? Conoce la inteligencia artificial? (s.d)*¹⁷ es conclou que ara vivim en l'era del "big data", una època en la qual tenim la capacitat de recollir grans quantitats d'informació massa enutjosa perquè una persona la processi. L'aplicació de la intel·ligència artificial en aquest sentit ja ha estat bastant fructífera en diverses indústries com la tecnologia, la banca, el màrqueting i l'entreteniment. Hem vist que fins i tot, si els algoritmes no milloren gaire, les dades i la informàtica massiva simplement permeten que la intel·ligència artificial aprengui a través de la força bruta. Hi pot haver proves que la llei de Moore està alentint una mica, però l'augment de dades certament no ha perdut cap impuls. Els avenços en ciències de la computació, matemàtiques o neurociència serveixen com a representació conceptual del sostre de la Llei de Moore.

4.2 FUTUR

Què hi ha per al futur? L'autor de *What is Open AI?---Its history and how ChatGPT is changing the world (s.d.)*¹⁸ diu que en el futur immediat, l'habilitat de la IA per comunicar-se amb paraules pot ser considerada com una de les revolucions més rellevants del s. XXI. De fet, ja està en marxa. Exemples d'això són les trucades no realitzades per humans que reben

¹⁷ A OCI. (s.d). *¿Qué es la ia? Conoce la inteligencia artificial*.

<https://www.oracle.com/es/artificial-intelligence/what-is-ai/>

¹⁸ En Taskade. (s.d.). *What is Open AI?---Its history and how ChatGPT is changing the world*.

<https://www.taskade.com/blog/openai-chatgpt-history/#:~:text=OpenAI%20Inc%20was%20founded%20in.3%20Playground%20and%20OpenAI%20API>

una gran quantitat de persones avui en dia. Es podria imaginar interaccionar amb un sistema expert en una conversa fluida, o tenir una conversa en dos idiomes diferents que es tradueixin en temps real. També podem esperar veure cotxes sense conductor en la carretera en els pròxims vint anys (i això és conservador). A llarg termini, l'objectiu és la intel·ligència general, que és una màquina que supera les habilitats cognitives humanes en totes les tasques. Això segueix les línies del robot sensible, que estem acostumats a veure en les pel·lícules. Personalment, em sembla inconcebible que això s'aconsegueixi en els pròxims cinquanta anys. Fins i tot si tenim la capacitat, les qüestions ètiques servirien com una barrera forta contra la distribució d'aquesta. Quan arribi aquest moment, haurem de tenir una conversa seriosa sobre la política de màquines i l'ètica (irònicament, dos temes fonamentalment humans).

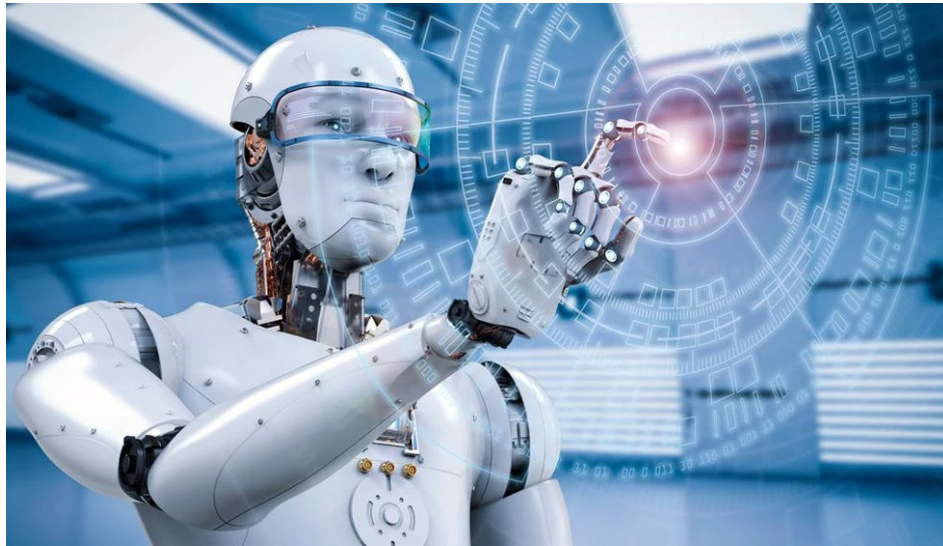


Figura 7 - Esbós d'un robot programat amb una IA. Ikusi (s.d.). Guía básica sobre la inteligencia artificial.

<https://www.ikusi.com/mx/blog/inteligencia-artificial-guia-definitiva/>

5. LLENGUATGES O FORMA DE PROGRAMACIÓ

L'autor de *Lenguaje de programación*¹⁹ estableix la següent definició: “Un llenguatge de programació és un llenguatge formal (és a dir, un llenguatge amb regles estructurals i símbols ben definits) que li proporciona a una persona (...) la capacitat d'escriure (o programar) una sèrie d'instruccions o seqüències d'ordres en forma d'algoritmes.” Aquesta capacitat tindrà, per norma general, la funció d'obtenir un determinat *output* que serà d'utilitat per la realització d'alguna tasca.

Un propòsit destacat dels llenguatges de programació és proporcionar instruccions a un ordinador. Com a tal, els llenguatges de programació difereixen de la majoria de les altres formes d'expressió humana, ja que requereixen un major grau de precisió i completesa. Quan s'utilitza un llenguatge natural per comunicar-se amb altres persones, els autors i parlants humans poden ser ambigus i cometre petits errors, i encara esperen que la seva intenció sigui entesa. No obstant això, els ordinadors fan exactament el que se'ls diu que facin, i no poden entendre el codi que el programador "intentava" escriure. La combinació de la definició del llenguatge, el programa i les entrades del programa ha d'especificar completament el comportament extern que ocorre quan s'executa el codi.

Molts de llenguatges de programació han estat dissenyats des de zero, modificats per satisfer noves necessitats, combinats amb els seus contemporanis i, finalment, han caigut en desús. Tot i que hi ha hagut intents de dissenyar un llenguatge informàtic "universal" que serveixi a tots els propòsits, cap d'ells ha fructificat. La necessitat de diversos llenguatges informàtics sorgeix de la diversitat de contextos en què s'utilitzen. No és el mateix un programa d'un aficionat al codi que necessitarà una empresa amb centenars de programadors.

Una tendència comuna en el desenvolupament de llenguatges de programació ha estat afegir més capacitat per resoldre dificultats utilitzant un nivell més alt d'abstracció. Els primers llenguatges de programació estaven molt lligats al maquinari subjacent de l'ordinador. A mesura que s'han desenvolupat nous llenguatges de programació, s'han afegit

¹⁹ Lenguaje de programación. (2023, 20 setembre). En *Wikipedia*.
https://es.wikipedia.org/w/index.php?title=Lenguaje_de_programaci%C3%B3n&oldid=153876655

característiques que permeten als programadors expressar idees que estan més allunyades de la traducció simple a les instruccions de maquinari subjacents. Com que els programadors estan menys lligats a les necessitats de l'ordinador, els seus programes poden fer més informàtica amb menys esforç del programador. Això els permet escriure més programes en la mateixa quantitat de temps.

5.1 PYTHON

Dintre de l'àmbit de la programació i de les disciplines que sorgeixen d'aquestes, es pot observar com diversos llenguatges han estat creats i desplegats. Concretament, en aquest treball, s'emprarà l'anomenat Python. Personalment, he elegit aquest llenguatge, ja que tenia uns lleugers coneixements d'aquest. També perquè, juntament amb C++, és el més usat quan es duen a terme projectes amb característiques similars al que es discutirà al final del treball.

A *¿Qué es Python?* (s.d.)²⁰, es diu que Python va ser creat per Guido van Rossum i es va començar a implementar l'any 1989. Des del 2003, Python ha estat constantment classificat com un dels deu llenguatges de programació més populars segons l'Índex de Comunitat de Programació TIOBE. A partir d'octubre de 2021, és el llenguatge més popular (per davant de Java, i C). També, va ser seleccionat Llenguatge de Programació de l'Any el 2007, 2010, 2018, i 2020 (l'únic llenguatge que ho ha fet quatre vegades. A més a més, diversos estudis han demostrat que els programadors poden ser més productius en Python que en altres llenguatges de programació. Un estudi realitzat per la companyia de consultoria McKinsey & Company va descobrir que els programadors poden ser fins a cinc vegades més productius en llenguatges de script com Python que en llenguatges convencionals com C i Java, especialment en problemes que impliquen la manipulació de cadenes i la cerca en un diccionari. També s'ha demostrat que Python té un consum de memòria comparable o millor que altres llenguatges de programació, com C o C++. Això fa que Python sigui una opció atractiva per a desenvolupadors que busquen equilibrar la productivitat amb l'eficiència en la gestió de recursos.

²⁰ AWS. (s.d.). *¿Qué es Python?*. <https://aws.amazon.com/es/what-is/python/>

Quant als seus usos, segons *Python® – the language of today and tomorrow* (s.d.)²¹, Python pot servir com a llenguatge de creació de scripts per a aplicacions web, per exemple, a través de *mod.wsgi*. Amb la interfície de passarel·la del servidor web, una API (extracte de codi que permet que diferents aplicacions es comuniquin entre si i comparteixin informació) estàndard ha evolucionat per facilitar aquestes aplicacions. Els marcs web com *Django*, *Pylons*, *Pyramid*, *TurboGears*, *web2py*, *Tornado*, *Flask*, *Bottle* i *Zope* donen suport als desenvolupadors en el disseny i manteniment d'aplicacions complexes.

Una altra informació destacable és que biblioteques com NumPy, SciPy i Matplotlib permeten l'ús efectiu de Python en la informàtica científica, amb biblioteques especialitzades com *Biopython* i *Astropy* que proporcionen funcionalitats específiques de domini. *SageMath* és un sistema d'àlgebra computacional amb una interfície programable en Python: la seva biblioteca cobreix molts aspectes de les matemàtiques incloent-hi àlgebra, combinatòria, matemàtiques numèriques, teoria de nombres i càlcul. Finalment, el llenguatge de programació en qüestió s'utilitza habitualment en projectes d'intel·ligència artificial i d'aprenentatge automàtic amb l'ajuda de biblioteques com *TensorFlow*, *Keras*, *Pytorch* i *scikit-learn*.

5.2 OPEN AI

OpenAI, com es diu a *What is Open AI?---Its history and how ChatGPT is changing the world* (s.d.)²², és una companyia de recerca d'intel·ligència artificial (IA) sense ànim de lucre que té com a objectiu promoure i desenvolupar intel·ligència artificial de tal manera que, segons està escrit a la seva política d'empresa, beneficiï a la humanitat en conjunt. L'organització té com a objectiu col·laborar amb altres institucions i investigadors per fer les seves patents obertes al públic. Per generar molts programes de xarxes neuronals, s'empra la seva llibreria Gym, que és coneguda mundialment.

²¹ Python Institute. (s.d.). *Python® – the language of today and tomorrow*. <https://pythoninstitute.org/about-python#:~:text=Python%20was%20created%20by%20Guido,called%20Monty%20Python's%20Flying%20Circus>

²² Taskade. (s.d.). *What is Open AI?---Its history and how ChatGPT is changing the world*. <https://www.taskade.com/blog/openai-chatgpt-history/#:~:text=OpenAI%20Inc%20was%20founded%20in.3%20Playground%20and%20OpenAI%20API>.

L'objectiu de Gym és estandarditzar la forma en què es defineixen els entorns en les publicacions de recerca de IA, de manera que la recerca publicada es torni més fàcil de reproduir. El projecte pretén proporcionar a l'usuari una interfície simple. A partir de juny de 2017, Gym només es pot usar amb Python. Un dels grans avantatges que proporciona, motiu pel qual és fet servir amb molta freqüència, són ambients completament fets per a realitzar les simulacions. És a dir, t'estalvies d'escriure múltiples línies de codi. Ara bé, un punt desfavorable, i el motiu pel qual no he utilitzat aquesta llibreria, és que et dona molt poca llibertat per interactuar amb components dels mòduls prèviament programats.

5.3 JUPYTER LAB

JupyterLab és un entorn de desenvolupament interactiu basat en la web (IDE) per treballar amb codi, dades i visualització. Proporciona una plataforma flexible per a la informàtica científica i l'anàlisi de dades que poden fer servir investigadors, desenvolupadors, científics de dades i estudiants. JupyterLab és similar al Quadern de Jupyter, però amb una interfície més moderna i rica en característiques, com s'explica a *Jupyter Lab is AWESOME for Data Science (2022)*²³.

Endemés, és un entorn de computació interactiva que suporta una àmplia gamma de llenguatges de programació incloent Python. A més a més, proporciona un entorn de desenvolupament integrat per a aquests llenguatges que inclou un editor de text, una consola per executar codi, un gestor de fitxers i una gamma d'altres eines per treballar amb dades i codi. Una de les característiques clau de JupyterLab és la seva habilitat de treballar amb els quaderns de Jupyter, els quals són documents interactius que combinen codi en directe, text narratiu i visualitzacions. JupyterLab permet als usuaris crear, editar i executar quaderns de Jupyter des del mateix entorn que el seu altre codi i dades. Això fa que sigui fàcil integrar codi, dades i visualitzacions en un únic entorn, que es pot compartir amb altres com un document autònom. L'eina en qüestió també proporciona una gamma d'eines per a la visualització de dades i l'exploració i inclou suport integrat per a biblioteques de visualització populars com *Matplotlib*, *Plotly* i *Bokeh*.

²³ NeuralNine. (23 de maig de 2022). *Jupyter Lab is AWESOME for Data Science* [vídeo]. Youtube. <https://www.youtube.com/watch?v=7wf1HhYQiDg>

6. MÈTODES DE PROGRAMACIÓ

En relació amb la temàtica d'aquest treball, hi ha tres branques/mètodes de programació que són de vital importància. En conseqüència, es tractaran a continuació.

6.1 MÈTODE DE DECISIONS MARKOV

Amb la informació extreta de *An introduction to Markov decision processes (s.d.)*²⁴, es pot dir que un procés de decisió de Markov (MDP) es refereix a un procés estocàstic (el seu comportament intrínsec no és determinista, és a dir, són magnituds aleatòries) de presa de decisions que utilitza un marc matemàtic per modelar la presa de decisions d'un sistema dinàmic (l'estat evoluciona amb el temps). S'usa en escenaris on els resultats són aleatoris o controlats per un prenedor de decisions, que pren decisions seqüencials amb el temps. Els MDP avaluen quines accions ha de prendre el responsable de la presa de decisions tenint en compte l'estat i l'entorn actuals del sistema. Endemés, es basen en variables com l'entorn, les accions de l'agent i recompenses per decidir la propera acció òptima del sistema. Es classifiquen en quatre tipus: finit, infinit, continu o discret, depenent de diversos factors com ara conjunts d'accions, estats disponibles i la freqüència de presa de decisions.

En intel·ligència artificial, s'empren per dissenyar màquines o agents intel·ligents que necessiten funcionar en un entorn on les accions poden produir resultats incerts.

6.2. Q-LEARNING

Prenent com a punt de partida *Introducción al aprendizaje por refuerzo. Parte 3: Q-Learning con redes neuronales, algoritmo DQN* (2020)²⁵, es pot dir que el *Q-learning* és un algoritme d'aprenentatge basat en valors i se centra en l'optimització de la funció de valor segons l'entorn o el problema. La Q en el *Q-learning* representa la qualitat amb la qual el model troba la seva pròxima acció millorant la qualitat. El procés pot ser automàtic i senzill. El model

²⁴ Duke University i Purdue University. (s.d.). *An introduction to Markov decision processes* [conjunt de dades]. <https://www.cs.rice.edu/~vardi/dag01/givan1.pdf>

²⁵ Sanz, M. (3 d'abril de 2020). *Introducción al aprendizaje por refuerzo. Parte 3: Q-Learning con redes neuronales, algoritmo DQN*. Medium. <https://markelsanz14.medium.com/introducci%C3%B3n-al-aprendizaje-por-refuerzo-parte-3-q-learning-con-redes-neuronales-algoritmo-dqn-bfe02b37017f>

emmagatzema tots els valors en una taula, que és la Taula Q. En paraules simples, s'empra el mètode d'aprenentatge per a la millor solució.

- 1. Inicialització: l'agent en jugar el joc per primera vegada no inclourà cap coneixement. Per tant, assumirem que la taula Q és zero.
- 2. Exploració: en aquest pas, l'agent triarà qualsevol dels dos camins possibles. Si l'agent acaba un episodi, recollirà informació de la taula Q. Després, intentarà fer nous camins.
- 3. Mesurar la recompensa: quan l'agent decideix quina acció triar, actua. Això porta a l'agent al següent pas, que és l'estat "S". Cadascuna d'aquestes accions dirigirà a l'agent a diverses puntuacions de recompensa. Per exemple, si l'agent tria l'estat cinc des de l'estat 1, aprendrà basant-se en l'experiència d'aquesta decisió. Ara, l'agent pot triar passar a l'estat 6 o al 9 en funció de l'experiència prèvia i de la possible expectativa de recompensa.
- 4. Actualitzar la taula Q: l'agent calcularà el valor de la recompensa. L'algorisme utilitzarà l'equació Q per a actualitzar el valor en l'Estat "S". A continuació, l'agent aconseguirà valors actualitzats en la taula Q.

L'equació que regeix l'aprenentatge en qüestió i que determina les decisions futures és la següent:

$$\text{New } Q(s,a) = Q(s,a) + \alpha [R(s,a) + \gamma \text{Max}_{a'} Q(s', a') - Q(s,a)]$$

On "s" significa l'estat; "a", l'acció i "R", la recompensa.

6.3 MÈTODE MONTECARLO

Els mètodes de Montecarlo són algorismes computacionals que s'usen per resoldre problemes que són difícils o impossibles d'abordar amb altres mètodes. Aquests mètodes es basen en l'ús de mostres aleatòries repetides per obtenir resultats numèrics. Són molt útils en problemes matemàtics i físics, com ara la simulació de sistemes complexos amb molts graus de llibertat, la resolució d'integrals definides multidimensionals amb condicions de frontera complicades i el càlcul de risc en negocis.

*A Monte Carlo And Off-Policy Methods | Reinforcement Learning Part 3 (2022)*²⁶, es diu que: “per resoldre problemes amb interpretació probabilística, els mètodes de Montecarlo es poden utilitzar per aproximar les integrals descrites pel valor esperat d'una variable aleatòria prenent la mitjana empírica de mostres independents de la variable”. Per això, s'ha de parametritzar la distribució de probabilitat de la variable i fer servir un mostrejador de Montecarlo de cadena de Markov, que genera mostres de la distribució desitjada.

Les simulacions en qüestió, com s'extreu de *Understanding the Markov Decision Process (MDP)*²⁷, requereixen la total realització d'un episodi complet per a poder usar la informació per aprendre. Bàsicament, s'elegeix una política d'acció i es realitza una simulació d'un joc, per exemple, i es calcula la recompensa acumulativa.

És important mencionar que per a jocs amb un arbre de joc colossal, no és recomanable. Per exemple, si s'empra per analitzar una partida d'escacs, es podria haver fet una partida magistral, però una jugada dolenta al final i es consideraria que totes les jugades tenen el mateix impacte en el resultat.

²⁶ Mutual Information. (26 d'octubre de 2022). *Monte Carlo And Off-Policy Methods | Reinforcement Learning Part 3* [vídeo]. Youtube. <https://www.youtube.com/watch?v=bpUszPiWM7o>

²⁷ Jagtap, R. (21 de novembre de 2022). *Understanding the Markov Decision Process (MDP)*. Built in. <https://builtin.com/machine-learning/markov-decision-process>

7. GENERACIÓ DEL PROGRAMA

A continuació es mostra i explica el codi utilitzat per obtenir una estimació de l'estratègia bàsica del blackjack.

7.1 PROGRAMA INICIAL

En primer lloc, s'ha trobat necessari realitzar un programa el qual pertanyi a la branca de les xarxes neuronals, el qual aprengui a jugar a blackjack. Això permetrà demostrar el concepte i fa referència a una de les hipòtesis plantejades al principi del treball: Es pot crear una intel·ligència artificial que amb el pas del temps millori la seva manera de jugar al blackjack? Per culpa de la naturalesa de l'objectiu que es persegueix, no serà necessari elaborar un programa optimitzat a nivells estratosfèrics; ni tampoc dur a terme un nombre d'esdeveniments (episodis de simulació) molt elevat. Amb tot això en ment, s'ha decidit fer servir *Q-Learning*. Més concretament, el laboratori d'informació RL Card i els entorns programats que ofereix. Així doncs, a continuació s'exposa i explica el codi.

```
[1]: import rlc card
    from rlc card.agents import DQNAgent

[3]: env = rlc card.make("blackjack")

[4]: agent = DQNAgent(
    num_actions=env.num_actions,
    state_shape=env.state_shape[0],
    mlp_layers=[64,64],
)

[5]: env.set_agents([agent])
```

Figura 8 - Captura del codi en qüestió feta des del meu ordinador.

En primer lloc, El mòdul rlc card s'importa, el qual proporciona una col·lecció d'entorns i agents predefinits per al reforç d'experiments d'aprenentatge. També, la classe DQNAgent s'importa des del mòdul rlc card.agents. Aquesta classe implementa un agent *Deep Q-Network (DQN)*, explicat anteriorment al treball. A continuació, una instància de l'entorn blackjack es crea utilitzant el mètode make() del mòdul rlc card. L'entorn Blackjack representa el que podria passar a una partida de Blackjack. Posteriorment, es crea una instància de la classe DQNAgent, amb els arguments següents: **num_actions**: el nombre de possibles accions que l'agent pot prendre. En el cas de Blackjack, l'agent pot demanar una carta més o quedar-se igual, de manera que s'estableix a 2. **State_shape**: la forma de la representació de l'estat usada per l'agent. En aquest cas, l'estat és representat per un vector

unidimensional, motiu pel qual es determina com a 0. **Mlp_layers**: una llista que especifica el nombre de neurones en cada capa de la xarxa neuronal utilitzada per l'agent. És a dir, punts pels quals la informació pot divagar. Finalment, l'agent creat s'afegeix a l'entorn blackjack usant el mètode `.setagents()`. Bàsicament, aquesta part del codi estableix una instància de l'entorn Blackjack i un agent Deep Q-Network, que podrà interactuar amb l'entorn. Aquesta configuració es farà servir per entrenar i avaluar el rendiment de l'agent en la tasca Blackjack.

```
[6]: from rlc card.utils import (
      tournament,
      reorganize,
      Logger,
      plot_curve,
    )
```

Figura 9 - Captura del codi en qüestió feta des del meu ordinador.

La següent part del codi importa classes i funcions de la llibreria amb què s'està treballant. En primer lloc, **tournament**, que pren una llista d'agents i un entorn, i simula un torneig on cada agent juga contra l'altre agent d'una manera *round-robin* (tots juguen contra tots). Els resultats del torneig es retornen com un diccionari amb una *key* (forma d'accedir als valors) per a cada agent i valors per a la recompensa total d'aquest últim. També **reorganize**. Una funció que pren una llista de trajectòries, on cada trajectòria és una llista de tuples (estat, acció, recompensa), i les reorganitza en un format que es pot usar per a l'entrenament mitjançant l'aprenentatge. A continuació, **Logger**, que proporciona una manera senzilla de registrar el progrés de l'entrenament a un fitxer o sortida estàndard. Per acabar, **plot_curve**, la qual pren una llista de punts de dades (per exemple les recompenses per episodi) i traça una corba de les dades al llarg del temps.

```
[7]: with Logger("experiments/leduc_holdem_dqn_result/") as logger:
      for episode in range(1000):
          trajectories, payoffs = env.run(is_training=True)
          trajectories = reorganize(trajectories, payoffs)
          for ts in trajectories[0]:
              agent.feed(ts)
          if episode % 50 == 0:
              logger.log_performance(
                  env.timestamp(),
                  tournament(
                      env,
                      10000,
                  )[0]
              )
          csv_path, fig_path = logger.csv_path, logger.fig_path
```

Figura 10 - Captura del codi en qüestió feta des del meu ordinador.

Aquesta és la part més important del programa, ja que conté un bucle d'entrenament per al DQNAgent en l'entorn Leduc Hold'em, amb registre i avaluació de torneigs a intervals regulars. La primera línia crea una instància de la classe Logger i estableix el directori de sortida per als fitxers de registre a «experiments/leduc_holdem_dqn_result/». La declaració “with” assegura que el registre es tanca adequadament quan es completa l'entrenament. A continuació es defineix el començament del bucle d'entrenament, que dura 1000 episodis. Després, s'executa un únic episodi de l'entorn de Leduc Hold'em amb el DQNAgent. L'argument “is_training=True” indica que l'agent hauria d'utilitzar la seva estratègia d'exploració durant aquest episodi. Posteriorment, es pretén reorganitzar les dades de trajectòria en un format que es pot fer servir per a l'entrenament per lots (basat en l'aprenentatge), emprant la funció *reorganize* (en aquest cas és important fer ús de la forma d'escriure americana, car s'està emprant una funció predeterminada per una llibreria pròpia dels EUA. Si no es fa, no funcionaria) explicada anteriorment.

```
[8]: plot_curve(csv_path, fig_path, "DQN")
```

Figura 11 - Captura del codi en qüestió feta des del meu ordinador.

Finalment, la funció “plot_curve(csv,path, fig,path, "DQN")”, que pren tres paràmetres: en primer lloc, **csv_path**: representa el camí a un fitxer CSV (valors separats per comes) que conté les dades rellevants. A continuació, **fig_path**: una cadena que representa el camí per a desar el diagrama generat com un fitxer d'imatge. Finalment, **DQN**. La funció fa ús de la biblioteca *Pandas* per a llegir en les dades CSV de csv_path i extreure els episodis i puntuacions de les columnes importants. A continuació, utilitza *matplotlib* per crear una gràfica de línies dels valors de les puntuacions contra els valors dels episodis. Per acabar, el diagrama es desa com un fitxer d'imatge de la manera indicada per “fig_path”. Aquest diagrama es tractarà al següent apartat. A més, s'ha afegit el codi complet en format text (amb comentaris respectius per a facilitar la comprensió) a un annex, el qual conté també els resultats exactes en valor numèric.

Amb això, podem concloure que es pot programar una xarxa neuronal que mitjançant esdeveniments aprengui a jugar al blackjack. Ara bé, no s'ha tractat la segona hipòtesi del treball: quina és la millor política de joc per a garantir un millor rendiment a l'hora de jugar-hi? Amb això en ment, es considera, analitza i explica una versió millorada i més complexa del codi en anterior, que donarà la política buscada.

7.2 PROGRAMA DEFINITIU

L'*script* en qüestió utilitzarà el mètode Montecarlo (explicat anteriorment al treball) i donarà una aproximació de l'estratègia de joc "òptima", la qual es representarà en manera de gràfica utilitzant la llibreria *Matplotlib*. Per motius d'espai (el programa és relativament extens) es parlarà i explicarà minuciosament les dues parts més rellevants del codi en relació amb la temàtica del treball. Ara bé, a l'annex apareix per complet i s'ha realitzat una explicació detallada. No obstant això, els resultats s'analitzaran també al següent punt del treball.

```
class Estat:
    def __init__(self, jugador_suma, crupier_visible, as_11):
        self.jugador_suma = jugador_suma
        self.crupier_visible = crupier_visible
        self.as_11 = as_11
        self.n_cartamés = 1
        self.n_quedarseigual = 1
        self.Q_cartamés_total = 0
        self.Q_quedarseigual_total = 0
        self.política = QUEDAR_IGUAL

    def actualitza(self, recompensa, acció):
        if acció == QUEDAR_IGUAL:
            self.n_quedarseigual = self.n_quedarseigual + 1
            self.Q_quedarseigual_total = self.Q_quedarseigual_total + recompensa
        else:
            self.n_cartamés = self.n_cartamés + 1
            self.Q_cartamés_total = self.Q_cartamés_total + recompensa

        if self.Q_cartamés_total / float(self.n_cartamés) > self.Q_quedarseigual_total / float(self.n_quedarseigual):
            self.política = DEMANAR_UNA_CARTA_MÉS
        else:
            self.política = QUEDAR_IGUAL
```

Figura 12 - Captura del codi en qüestió feta des del meu ordinador.

En primer lloc, s'analitzarà la classe Estat, que s'empra per representar l'estat d'un joc de blackjack. La classe té dues funcions: "init" i "actualitza". La funció "init" s'usa per inicialitzar la classe i els seus atributs. Aquesta pren tres arguments: **jugador_suma**, **crupier_visible**, i **as_11**, que representen la suma de les cartes del jugador, el valor de la carta visible del crupier i si el jugador té o no un as que es pot comptar com 11. La funció llavors estableix diversos atributs, incloent-hi "n_cartamés", "n_quedarseigual", "Q_cartamés_total", "Q_quedarseigual_total" i "política" (fa referència a la manera de jugar). Aquests atributs s'utilitzen per fer un seguiment de diverses estadístiques durant el joc i aplicar-les a l'equació Q explicada anteriorment.

D'altra banda, la funció "actualitza" es fa ús per actualitzar l'estat del joc. Aquesta pren dos arguments: **recompensa** i **acció**, que representen la recompensa per prendre una acció particular i la mateixa acció (sigui "DEMANAR_UNA_CARTA_MÉS" o "QUEDAR_IGUAL"). Si l'acció és l'última mencionada abans, el mètode actualitza els atributs en qüestió amb la nova recompensa. Al contrari, actualitza els atributs "n_cartamés" i «Q_cartamés_total».

Finalment, la funció compara les recompenses mitjanes d'ambdues accions i si la recompensa esperada per “DEMANAR_UNA_CARTA_MÉS” és més alta que la de “QUEDAR_IGUAL”, actualitza l'atribut de “política” amb “DEMANAR_UNA_CARTA_MÉS”. Si és el contrari, el posa a “QUEDAR_IGUAL”. En resum, l'atribut “política” s'utilitza per determinar la millor acció a prendre basada en l'estat actual del joc.

```
def monteCarloES(num_episodis=5000000):
    estats = [Estat(i, j, 1) for i in range(11, 22) for j in range(1, 11) for l in reversed(range(2))]
    for i in range(0, num_episodis):
        s = random.choice(estats)
        episodi = []
        bj = Blackjack(s.jugador_suma, s.crupier_visible, s.as_11)
        acció = random.randint(0, 1)
        episodi.append([s, acció])
        while True:
            jugador_suma, crupier_visible, as_11, recompensa, joc_finalitzat = bj.torn(acció)
            if joc_finalitzat == False:
                s = estats[determinaEstat(jugador_suma, crupier_visible, as_11)]
                acció = s.política
                episodi.append([s, acció])
            else:
                for e in episodi:
                    e[0].actualitza(recompensa, e[1])
                break
    return estats
```

Figura 13 - Captura del codi en qüestió feta des del meu ordinador.

La segona part més important és la funció anomenada monteCarloES que implementa l'algoritme de Montecarlo per estimar la política òptima per jugar al Blackjack. La funció pren un argument modificable **nombre_episodis** que especifica el nombre d'episodis (jocs) a jugar. Si no es proporciona aquest argument, es fa servir el valor predeterminat de 7500000. La funció primer crea una llista anomenada estats, que conté tots els estats possibles del joc de blackjack. Cada estat està representat per una instància de la classe Estat, que pren tres arguments: **jugador_suma**, **crupier_visible**, i **as_11**. A continuació, la funció entra en un bucle que s'executa per al nombre d'iteracions determinat pel valor de la variable “nombre_episodis”. A cada iteració, se selecciona un estat aleatori de la llista d'estats i es juga un episodi del joc a partir d'aquest estat. La llista d'episodis es fa servir per fer un seguiment de la seqüència d'estats i accions preses durant l'episodi.

L'estat i l'acció s'afegeixen a la llista d'episodis. Ara bé, si el joc ha acabat, la recompensa s'actualitza per cada parell d'accions d'estat a la llista d'episodis fent ús de la funció “actualitza” de la classe Estat. Després de tots els episodis, la llista d'estats que conté les estimacions actualitzades de la política òptima per a cada estat és retornada per la funció.

8. ANÀLISI DE RESULTATS

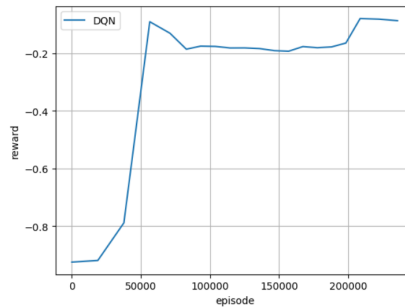


Figura 14 - Gràfic de resultats del primer programa.

Aquesta gràfica representa els resultats del primer programa. Com es pot veure, a mesura que el nombre d'episodis incrementa, la recompensa augmenta. Això demostra que és possible crear un programa que aprengui a jugar al blackjack. Una altra observació és que arriba un punt on la progressió s'estanca i no millora. Motiu pel qual, juntament amb el fet que el punt mort es troba en negatiu, es pot concloure que jugar una quantitat infinita de partides de blackjack és matemàticament "impossible" guanyar a la banca. Ara bé, hi ha una manera de minimitzar pèrdues i, fins i tot jugant poques partides, tenir possibilitats de guanyar. A continuació, els resultats del segon programa.

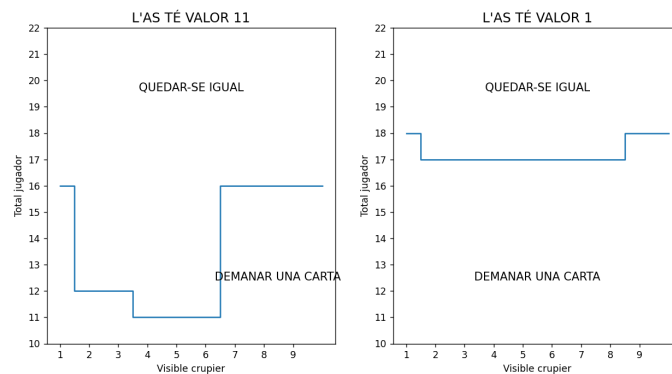


Figura 15 - Gràfic que mostra l'estratègia de joc òptima segons el segon programa.

Aquesta és l'aproximació del segon programa a què seria l'estratègia més bona. Així doncs, es pot realitzar una comparació amb l'estratègia de joc estandarditzada. Arran d'aquesta comparació, es pot determinar que el codi està ben redactat, car l'estructura de l'estratègia és menys o menys igual. Això no obstant, hi ha lleugeres discrepàncies. Per llegir els gràfics es faria el següent. En primer lloc, es miraria el total del jugador. Després, la carta visible del crupier. Si es dona el cas que la intersecció d'aquests dos valors està damunt la línia blava, llavors el jugador ha de quedar-se igual. Si passa el contrari, demanar una carta més.

9. CONCLUSIÓ

En aquest treball he plantejat tres hipòtesis inicials, les quals he analitzat a partir d'informació que he buscat i exposat amb l'objectiu de poder-les corroborar o refutar. Com a resultat del treball, l'algoritme de Montecarlo s'ha utilitzat amb èxit per determinar l'estratègia òptima de joc per a blackjack, tenint en compte diferents combinacions de cartes i la targeta visible del crupier. Mitjançant l'ús de tècniques d'aprenentatge automàtic, l'algoritme ha estat capaç d'aprendre de l'experiència i millorar gradualment el seu rendiment. D'altra banda, l'ús d'eines de visualització de dades, com *matplotlib*, ha permès la representació clara de l'estratègia òptima de joc en diferents escenaris. Aquesta informació pot ser usada pels jugadors per millorar la seva presa de decisions i maximitzar les seves possibilitats de guanyar.

Principalment, aquest treball ha repercutit positivament en el seu autor. És a dir, jo. Perquè no només he perfeccionat i incrementat notablement les nocions sobre programació i informàtica que tenia al principi, sinó també he après nous conceptes i algoritmes (*Q-Learning*, Montecarlo, MKD...). Quant a les hipòtesis i les conclusions a les quals s'ha arribat es podria dir el següent:

Primera hipòtesi: Amb una mostra de partides prou elevada per poder extraure conclusions, la banca sempre tindrà l'avantatge matemàtic a l'hora de guanyar. Ara bé, utilitzant una estratègia òptima es poden minimitzar les pèrdues i esperar “un cop de sort” per sortir guanyant, motiu pel qual és important saber quina és.

Segona hipòtesi: L'aproximació a l'estratègia òptima d'una intel·ligència artificial és molt similar a la prèviament consolidada. Malgrat que no s'ha demostrat si les divergències són pròpies d'errors a l'hora de programar o millores en la manera de jugar. Seria interessant consultar aquest treball d'aquí uns quants anys quan tingui més coneixements per saber quin és el motiu de la diferència.

Tercera hipòtesi: Es valida que es pot crear programa que aprengui a jugar a blackjack i doni una estratègia òptima.

10. BIBLIOGRAFIA

— Sobre el joc blackjack.

Mejor casino online.mx. (s.d). *Cómo jugar Blackjack*. (s.d).

<https://mejorcasinonline.mx/como-jugar-blackjack/>

Bicyclecards. (s.d). *Learn to play Blackjack*. <https://bicyclecards.com/how-to-play/blackjack/>

Gonzalves, R.(10 de juny del 2022). *La historia del blackjack*. Bestonlinecasino.com.

<https://www.bestonlinecasino.com/es/juegos/blackjack/historia/>

Abran juego. (14 de gener del 2015). *Reglas del blackjack*.

<https://www.luckia.es/blog/reglas-del-blackjack/>

Wintle, S. (27 d'octubre del 2022). *History of Blackjack*. The world of playing cards.

<https://www.wopc.co.uk/the-history-of-playing-cards/blackjack/blackjack>

Gambling sites. (Abril 2023). *Learn How to Play Blackjack*.

<https://www.gamblingsites.org/casino/blackjack/beginners-guide/>

Jones, C. (s.d). *How to play blackjack*.

<https://www.blackjackapprenticeship.com/how-to-play-blackjack/>

— Sobre les matemàtiques de jugar al blackjack.

Griffin, P. (1999). *The Theory of Blackjack: The Complete Card Counter's Guide to the Casino*. (5a edició). Huntington Press.

Blanchard, H. (13 de maig de 2019). *Blackjack: the math behind the cards*. [Tesi doctoral, Louisiana Tech University]. viewcontent.cgi.

<https://digitalcommons.latech.edu/cgi/viewcontent.cgi?article=1007&context=mathematics-senior-capstone-papers#:~:text=The%20terminology%20is%20portrayed%20as,first%20two%20cards%20she%20receives.>

Not only Science!. (25 de gener de 2022). *The mathematics of gambling: how information theory changed blackjack and other games of chance* [vídeo]. Youtube.

<https://www.youtube.com/watch?v=d038pCViGpA>

Casino Maths. (15 de març de 2017). *The Maths Behind Blackjack* [vídeo]. Youtube.

<https://www.youtube.com/watch?v=YDBeWh5hqP4>

Casino Maths. (15 de març de 2017). *Blackjack answers* [vídeo]. Youtube.

<https://www.youtube.com/watch?v=kFjYBMU5q64&t=0s>

Departamento de Estadística e Investigación Operativa Facultad Ciencias Matemáticas UCM. (1985). *Estudio de la estrategia óptima para el black-jack* [conjunt de dades].

Universidad Complutense de Madrid.

<https://eprints.ucm.es/id/eprint/16268/1/Tejada18.pdf>

— Sobre la inteligencia artificial.

Kelley, K. (9 de març de 2023). *What is artificial intelligence: Types, History or Future*.

Simplilearn.

<https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/what-is-artificial-intelligence#:~:text=A%20Brief%20History%20of%20Artificial%20Intelligence,-Here's%20a%20brief&text=1956%20%2D%20John%20McCarthy%20coined%20the,just%20a%20list%20of%20instructions.>

Java T point. (s.d.). *History of Artificial Intelligence*.

<https://www.javatpoint.com/history-of-artificial-intelligence>

OCI. (s.d). *¿Qué es la ia? Conoce la inteligencia artificial.*

<https://www.oracle.com/es/artificial-intelligence/what-is-ai/>

Taskade. (s.d.). *What is Open AI?---Its history and how ChatGPT is changing the world.*

<https://www.taskade.com/blog/openai-chatgpt-history/#:~:text=OpenAI%20Inc%20was%20founded%20in.3%20Playground%20and%20OpenAI%20API.>

— Sobre llenguatges de programació i metodologies similars.

Lenguaje de programación. (2023, 20 septiembre). En *Wikipedia*.

https://es.wikipedia.org/w/index.php?title=Lenguaje_de_programaci%C3%B3n&oldid=153876655

NeuralNine. (23 de maig de 2022). *Jupyter Lab is AWESOME for Data Science* [vídeo].

Youtube.

<https://www.youtube.com/watch?v=7wf1HhYQiDg>

AWS. (s.d.). *¿Qué es Python?*

<https://aws.amazon.com/es/what-is/python/>

Python Institute. (s.d.). *Python® – the language of today and tomorrow.*

<https://pythoninstitute.org/about-python#:~:text=Python%20was%20created%20by%20Guido.called%20Monty%20Python's%20Flying%20Circus.>

Proyecto Jupyter. (23 de març de 2023). En *Wikipedia*.

https://es.wikipedia.org/wiki/Proyecto_Jupyter

Jagtap, R. (21 de novembre de 2022). *Understanding the Markov Decision Process (MDP).*

Built in.

<https://builtin.com/machine-learning/markov-decision-process>

Duke University i Purdue University. (s.d.). *An introduction to Markov decision processes* [conjunt de dades].

<https://www.cs.rice.edu/~vardi/dag01/givan1.pdf>

Sanz, M. (3 d'abril de 2020). *Introducción al aprendizaje por refuerzo. Parte 3: Q-Learning con redes neuronales, algoritmo DQN*. Medium.

<https://markelsanz14.medium.com/introducci%C3%B3n-al-aprendizaje-por-refuerzo-parte-3-q-learning-con-redes-neuronales-algoritmo-dqn-bfe02b37017f>

Mutual Information. (26 d'octubre de 2022). *Monte Carlo And Off-Policy Methods | Reinforcement Learning Part 3* [vídeo]. Youtube.

<https://www.youtube.com/watch?v=bpUszPiWM7o>

— Per aprendre a programar i analitzar el codi.

Algorithmen und Datenstrukturen. (7 de gener de 2022). *Monte-Carlo-Methoden für Spielstrategie* [vídeo]. Youtube.

<https://www.youtube.com/watch?v=jdbUBkqtAyc>

Gymnasium Documentation. (s.d.). *Blackjack*.

https://gymnasium.farama.org/environments/toy_text/blackjack/

Bujokas, E. (14 de febrer de 2023). *The Values of Actions in Reinforcement Learning using Q-learning*. Medium.

<https://towardsdatascience.com/the-values-of-actions-in-reinforcement-learning-using-q-learning-cb4b03be5c81>

Edan Meyer. (5 de març de 2020). *Q Learning Tutorial: Testing & Visualizing* [vídeo]. Youtube.

https://www.youtube.com/watch?v=CnkSClZ_75A

Vega, O. (2018). Una introducción a los procesos de decisión markovianos con costo promedio [conjunt de dades]. Miscelánea matemática.

https://miscelaneamatematica.org/download/tbl_articulos.pdf2.a9d7b23c8fcf943b.363630342e706466.pdf

Gymnasium Documentation. (s.d.). *Solving Blackjack with Q-Learning*.

https://gymnasium.farama.org/tutorials/training_agents/blackjack_tutorial/

Sommerville, G. (12 de febrer de 2019). *Winning Blackjack using Machine Learning*. Towards Data Science.

<https://towardsdatascience.com/winning-blackjack-using-machine-learning-681d924f197c>

Gazit, A. (31 de maig de 2019). *Reinforcement-Learning-Agent-Plays-Blackjack-Q-algorithm*. GitHub.

https://github.com/AsafGazit/Reinforcement-Learning-Agent-Plays-Blackjack-Q-algorithm/blob/master/software_agents_submit.pdf

Matplotlib. (s.d.). *Path Tutorial*.

<https://matplotlib.org/stable/tutorials/index.html>

Matplotlib. (s.d.). *origin and extent in imshow*.

https://matplotlib.org/stable/tutorials/intermediate/imshow_extent.html#sphx-glr-tutorials-intermediate-imshow-extent-py

Neral Nine. (5 d'octubre de 2021). Professional 3D Plotting in Matplotlib [vídeo]. YouTube.

<https://www.youtube.com/watch?v=fAztJg9oi7s>

— Annex

Tech with Tim. (19 de septembre de 2022). *Every Python Function Explained* [vídeo]. YouTube.

<https://www.youtube.com/watch?v=NYktbp1WFS8>

Tech with Tim. (7 d'octubre de 2022). *Every Python Function Explained / H-Z* [vídeo]. YouTube.

<https://www.youtube.com/watch?v=wsETdTd1rp8>

Goyal, C. (22 de juliol de 2022). *15 Python Built-in Functions which You Should Know while learning Data Science*. Analytics Vidhya.

<https://www.analyticsvidhya.com/blog/2021/07/15-python-built-in-functions-which-you-should-know-while-learning-data-science/>