



Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 22/11/2023 | Aceptado: 14/02/2024 | Publicado: 30/03/2024

Identificadores persistentes:
DOI: 10.48168/innosoft.s15.a157
ARK: [ark:/42411/s15/a157](https://nbn-resolving.org/urn:ark:/42411/s15/a157)
PURL: [42411/s15/a157](https://nbn-resolving.org/urn:purl:42411/s15/a157)

Poker Hand Valuator, IA evaluadora de manos de poker

Poker Hand Valuator, poker hand evaluator AI

Estith Bryan Vargas Quispe¹, Eybert Macedo Pillco², Quispe Ttito Juan Carlos³, Jose Miguel Cano Vilcapaza⁴

¹ Universidad Nacional de San Agustín. Arequipa, Perú. evargasq@unsa.edu.pe

² Universidad Nacional de San Agustín. Arequipa, Perú. emacedop@unsa.edu.pe

³ Universidad Nacional de San Agustín. Arequipa, Perú. jquispett@unsa.edu.pe

⁴ Universidad Nacional de San Agustín. Arequipa, Perú. jcanov@unsa.edu.pe

* Autor para correspondencia: evargasq@unsa.edu.pe

Resumen

Nuestro objetivo es la predicción de manos de poker, la probabilidad de que pueda sacar una buena mano cuando se hace 5 robos de una baraja de poker de 52 cartas, aplicamos la redes neuronales para realizar dicha predicción conjunto con diferentes librerías que ayudan a que el proceso sea más simplificado y los resultados sean más fiables, por lo tanto en el uso de esta metodología logramos obtener un average de 97% en la mayoría de los casos con una desviación de 2.5% lo cual consideramos aceptable debido a la cantidad muy desbalanceada de los datos de este dataset, por lo tanto este método de inteligencia artificial nos sirve para predecir nuevas manos y tomar mejores decisiones conforme te encuentres en una situación del juego.

Palabras clave: dataset, poker hand, poker texas, predicción, red neuronal.

Abstract

Our objective is the prediction of poker hands, the probability that you can draw a good hand when you make 5 steals from a 52-card poker deck. We apply neural networks to make this prediction together with different libraries that help the process is more simplified and the results are more reliable, therefore in the use of this methodology we managed to obtain an average of 97% in most cases with a deviation of 2.5% which we consider acceptable due to the very unbalanced amount of the data from this dataset, therefore this artificial intelligence method helps us predict new hands and make better decisions as you find yourself in a game situation.

Keywords: Dataset, poker hand, poker texas, prediction, neural network.

Introducción

En este trabajo se realizará un algoritmo que se encargue de tomar la decisión según la mayor probabilidad de formar una mejor mano de poker cuando ya se recibieron las cinco cartas iniciales, cabe destacar que existen proyectos que ya realizan trabajos similares como cuando intentan estimar probabilidades en el juego de poker texas que proporcionar a diferentes usuarios un soporte de simulación que permita estimar la probabilidad de ganar una mano[1].

La mayoría de los proyectos de inteligencia artificial referentes al poker son para crear un bot que se encargue de ganar revisando unos que otros parámetros para conseguir su objetivo, entre estos tenemos el bot de [2] que analiza los parámetros de nivel de riesgo, el valor de las jugadas y la apuesta correspondiente, por otro lado otro bot revisa los parámetros [9], adaptar su modo de juego de juegos anteriores.

Uno de los mayores problemas en el poker es la toma de decisiones en la primera mano para cambiar las cartas necesarias según las cartas que te vas a quedar en la mano, esta decisión es fundamental al momento de continuar o no con la partida, por lo que este algoritmo ayuda bastante a un novato e incluso a un profesional en cual seria la mejor decisión cuando se le presente una de las situaciones ya producidas por el algoritmo.

Entonces nos planteamos el método de heurísticas que hacen inferencias deductivas e inductivas, forma y optimizan reglas heurísticas para hacer que las máquinas alcancen resultados similares a una decisión humana[7], otro método de uso también son los algoritmos genéticos[6] los cuales son adaptativos los cuales se emplean para la resolución de problemas de búsqueda y optimización, permiten optimizar funciones numéricas como explica el artículo [5] que utilizar este método para obtener distintos resultados de optimización que sean necesarios para resolver la mejor mano en cierto momento del juego.

Estas decisiones pueden determinar la victoria en una partida importante por lo que aun cuando una de las partes más importantes del poker es el blofeo de lo que uno podría tener en la mano, independientemente de lo que tengan los demás, tomar una buena decisión implica la mayor probabilidad de que salgas librado de un blofeo del oponente.

Entonces nuestro objetivo es que el algoritmo analice la primera mano y escoja las cartas que debe devolver al mazo para que pueda conseguir una mejor combinación, dicho algoritmo tendrá que volver a analizar la nueva mano y guardar la información para futuros análisis de las siguientes manos.

Materiales y métodos o Metodología computacional

Redes neuronales: Las redes neuronales tratan de emular el comportamiento del cerebro humano, caracterizado por el aprendizaje genérico a partir de un conjunto de datos. Estos sistemas imitan esquemáticamente la estructura neuronal del cerebro.[17]

Machine learning: Es una forma de la IA el cual permite a los sistemas aprender de los datos en lugar de aprender de la programación explícita, pero esto no es un proceso sencillo. Un modelo de machine learning es la salida de información que se genera cuando se entrena un algoritmo de machine learning con datos, después del entrenamiento al proporcionar un tipo de modelo, se nos da una salida con el mismo modelo.

Aprendizaje iterativo: Nos permite entrenar modelos con conjuntos de datos antes de ser implementados, algunos de los modelos de Machine Learning están online y son continuos, este proceso iterativo de modelos permite que se logre una optimización en la predicción de datos y que debido al tamaño y la complejidad de los datos, estos pueden ser pasados por alto muy fácilmente por un ser humano.

Aprendizaje Supervisado: comienza típicamente con un conjunto establecido de datos y una cierta comprensión de cómo se clasifican los datos, este aprendizaje pretende encontrar patrones para luego aplicarlos a un proceso de análisis. Un ejemplo claro es la aplicación con base en imágenes y descripciones escritas que distinguen entre millones de animales

Aprendizaje no Supervisado: Se utiliza cuando el problema no requiere una cantidad masiva de datos sin etiquetar, claro ejemplo son las redes sociales las cuales tienen grandes cantidades de datos sin etiquetar, la comprensión detrás de estos datos requieren algoritmos que clasifican los datos con base a los patrones que encuentra. El aprendizaje no supervisado lleva a cabo un proceso iterativo, el cual analiza los datos sin intervención humana.

Poker Hand Data Set:

El data set fue creado el 2007 por Robert Cattral y Franz Oppacher con el objetivo de predecir una buena mano de poker, cada fila de este dataset está compuesta por 5 cartas robadas de una baraja estándar de 52 piezas, cada carta es

descrita usando 2 atributos (palo y número), para un total de 10 variables predictivas. Finalmente tenemos un atributo más el cual se encarga de describir el tipo de mano que se formó al momento del último robo de carta.

El número de instancias o filas que tenemos en el dataset para el entrenamiento es de 25010 filas, mientras que para el testeo tenemos 1 000 000 filas.

Número de atributos predictivos tenemos 10 y un atributo objetivo el cual nos indica el tipo de mano que se formó con los otros atributos.

Tabla 1. Información de los atributos o variables que componen una fila del dataset de Poker Hand

	nombre variable	tipo de atributo	valor de atributo	representa
1	S1	Ordinal	1-4	Corazones, espadas, diamantes, trébol
2	C1	Numérico	1-13	As, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina, rey
3	S2	Ordinal	1-4	Corazones, espadas, diamantes, trébol
4	C2	Numérico	1-13	As, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina, rey
5	S3	Ordinal	1-4	Corazones, espadas, diamantes, trébol
6	C3	Numérico	1-13	As, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina, rey
7	S4	Ordinal	1-4	Corazones, espadas, diamantes, trébol
8	C4	Numérico	1-13	As, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina, rey
9	S5	Ordinal	1-4	Corazones, espadas, diamantes, trébol
10	C5	Numérico	1-13	As, 2, 3, 4, 5, 6, 7, 8, 9, 10, jota, reina, rey
11	Tipo	Ordinal	0-9	revisar tabla 2

A continuación revisamos los valores que puede tomar el atributo Tipo los cuales dependen de los otros atributos que son de las cartas robadas de la baraja.

Tabla 2. Descripción de valores que puede tomar el atributo de tipo según las cartas robadas de la baraja de Poker Hand

valor del atributo	nombre de la mano	descripción
0	nada en mano	no es una mano de poker reconocida
1	un par	un par de rango iguales entre las 5 cartas
2	dos pares	dos pares de rango iguales entre las 5 cartas
3	tres de un tipo	tres cartas iguales de mismo rango entre las 5 cartas
4	escalera	5 cartas secuencialmente numeradas
5	Color	5 cartas de un mismo palo
6	Full	un par y un trío del mismo número respectivamente
7	poker	4 cartas de Ases en una mano de 5 cartas
8	escalera color	5 cartas numeradas secuencialmente de un mismo palo
9	escalera real de color	as, rey, reina, jota, 10 todas de un mismo palo

La distribución de las manos que se aprecian en el dataset según el tipo de mano se ve bastante desbalanceado, en la figura 1 observamos un gráfico en el cual la mano (nada en mano y un par), son bastante abundantes, por lo que ambos componen un aproximado de 92.33% de los datos en el dataset de entrenamiento, aun incluyendo casi todos los valores que pueden tomar las manos menos frecuentes la diferencia es muy notable.

El dataset original sobre el cual trabajaremos tiene la siguiente distribución y tamaño:

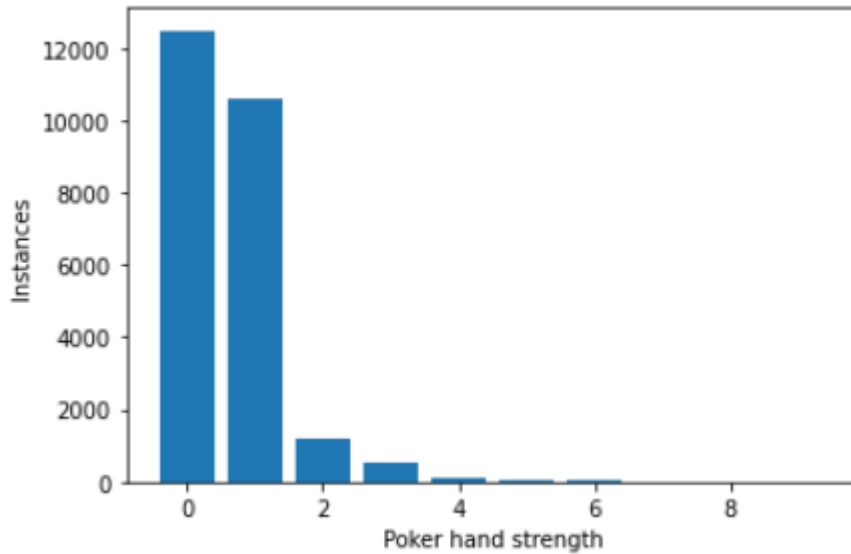


Figura 1. Gráfico de cantidad de manos encontradas en el entrenamiento de la red neuronal.

En la tabla 3 veremos claramente en porcentaje la cantidad de manos que tiene el dataset de entrenamiento con su porcentaje respectivo y la cantidad de manos que tiene el dataset total de manos que tiene el Poker Hand con sus porcentajes.

Tabla 3. Cantidad y porcentajes de combinaciones de tipos de mano en el dataset de entrenamiento y dataset global

valor de atributo	nombre de mano	cantidad instancias de entrenamiento (CIA)	porcentaje CIA	cantidad de instancias global (CIG)	porcentaje CIG
0	nada en mano	12 493	49.95202%	1 302 540	50.1177%
1	un par	10 599	42.37905%	1 098 240	42.2569%
2	dos pares	1 206	4.82207%	123 552	4.7935%
3	tres de un tipo	513	2.05118%	54 912	2.1128%
4	escalera	93	0.37185%	10 200	0.3925%
5	Color	54	0.21591%	5 108	0.1980%

6	Full	36	0.14394%	3 744	0.1441%
7	poker	6	0.02399%	624	0.0240%
8	escalera color	5	0.01999%	36	0.0014%
9	escalera real de color	5	0.01999%	4	0.0002%
total		25 010	99.9999%	2 598 960	100%

Herramientas:

Google Colaboratory: Según el sitio oficial Google Colab [11] es un producto que permite escribir y ejecutar el lenguaje Python de manera online sin costo alguno. En producto es donde ejecuta la mayoría del código para el presente trabajo, principalmente para el entretenimiento del modelo de redes neuronales aplicadas a Poker Hand.

Python: Challenger, Díaz, Becerra. [15] Es un lenguaje de programación que fue desarrollado bajo el concepto de ser libre de uso, ejecución, distribución y modificación.

Tensor Flow: Según el sitio oficial Tensor Flow[16] es una plataforma de extremo a extremo de código abierto para el aprendizaje de desarrollo y entrenamiento de modelos de AI.

Librería Pandas: Librería especializada en el manejo y análisis de estructuras de datos. [13]

Librería Numpy: Según el manual de Python[12] es una librería de python que se encarga de hacer cálculos numéricos y analizar los datos a gran volumen.

Librería Sklearn: Según la UOC[14] Librería que se encarga del procesamiento de un conjunto de datos. Conjunto de rutinas escritas en Python para hacer el análisis predictivo de algoritmos. Esta librería estaba en Numpys, SciPy y Matplotlib.

Procedimientos

Escogimos realizar una red neuronal para detectar la fortaleza de las manos de poker, realizamos varios modelos cada uno mejor que el anterior implementando nuevas técnicas que nos ayuden a mejorar las métricas accuracy, precision y recall, las cuales son indicadores muy utilizados para medir el rendimiento de las redes neuronales según Juba B. [18] sobre todo para aquellas muy grandes.

Data cleaning

Como se observa en la tabla 3 anteriormente vista, el dataset esta muy desbalanceado y vemos que existe filas repetidas, esto se nota en la mano de escalera real de color, segun el dataset global solo existen 4 manos posibles pero en el dataset de entrenamiento observamos que existen 5 manos, por lo cual podemos decir que existe una fila repetida, segun esta observacion podriamos inducir que las demas manos tambien podrian tener filas repetidas por lo cual antes de trabajar con este dataset de entrenamiento debemos hacer tratamiento previo a todo el dataset de entrenamiento.

Para el primer modelo (y los siguientes) se realizó el tratamiento de datos eliminación de valores “missing” con la función **dropna()** de pandas para eliminar las filas que no tengan datos si es que los hubiere, esto lo hacemos como un tratamiento estándar de los datos, igualmente, se eliminan los valores duplicados con la función **df.drop_duplicates(keep='first')** el cual se encarga de eliminar las filas duplicadas que se encuentran en el dataset, manteniendo la primera instancia encontrada y eliminando las otras repetidas.

Ahora nos encontramos con otro problema, antes que nada explicare que una mano de poker por ejemplo escalera real de color que es “As, rey, reina, jota” de mismo palo, 10, es lo mismo en todas sus formas desordenadas en que se robaron de la baraja, por ejemplo su homonimos “rey, As, reina, 10, jota”, “10, rey, As, reina, jota” y sus demas combinaciones del mismo palo, dicho esto la funciones anteriormente mencionadas para eliminacion de filas vacias y eliminacion de duplicados no elimina estos homonimos que a fin de cuentas son la misma mano, por lo tanto desarrollamos una funcion que se encargue de resolver este problema.

Función de mapeo: Esta función se encarga de mapear todas las manos que tengamos en el dataset de entrenamiento, descomponiendo en un array adicional y ordenando los atributos, según encuentre el mapeo en el array de una fila la compara con el mapeo de otras filas, y si dos array son iguales elimina el segundo array encontrado (fila) del dataset véase la figura 2

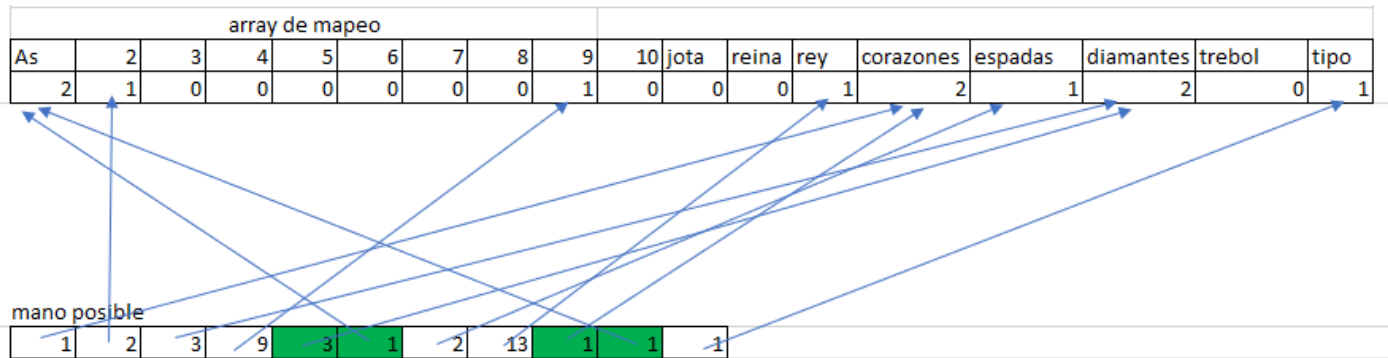


Figura 2. mapeo de manos utilizando un arreglo y contando la cantidad de apariciones.

Observamos que usando el método de transform(df) a realizar un ejemplo para una fila no hace el conteo de cuantas apariciones tenemos de cada palo y cada número en la mano de 5 cartas, por lo que ahora que está ordenado y mapeado tomamos como primer elemento y el las siguiente lecturas y mapeos de otras filas si encuentra alguna fila que sea igual al primer arreglo encontrado podemos deducir que es la misma mano, por lo tanto procedemos a eliminar esa segunda fila del dataset, así eliminamos todas las filas que sean repetidas aun cuando las variables están en desorden.

Utilización de SMOTE:

También se utilizó SMOTE (Synthetic Minority Over-sampling Technique) para la creación de clases sintética, sin embargo no otorgó buenos resultados pues existían clases con cantidad de muestras menores a 5 vecinos, lo cual imposibilitaba la utilización correcta de SMOTE ya que cambia todas las clases a la menor terminando con 124930 muestras.

```
array([9, 9, 9, ..., 9, 9, 9])
```

Figura 3. Clases sintéticas al utilizar (k_neighbors<=4)

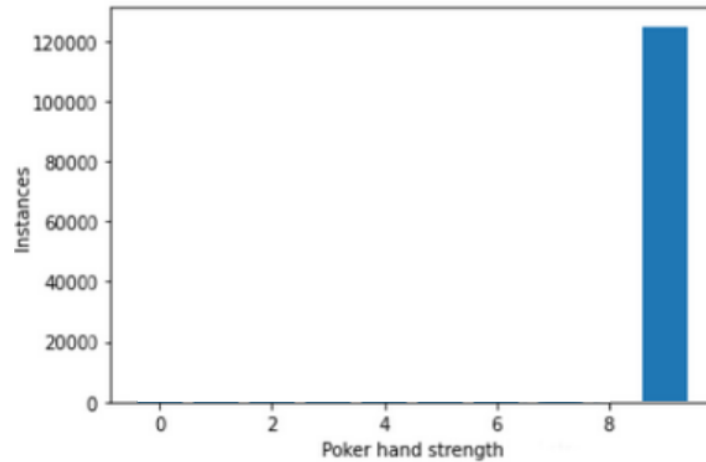


Figura 4. Clases al utilizar SMOTE

Utilización de class weights:

Se utilizó el paquete `compute_class_weight` de la librería `sci-kit learn` para asignar pesos dependiendo de las instancias existentes para cada clase, de esta forma asignando un peso mayor a las clases más raras.

```
[2.00176099e-01 2.35991318e-01 2.07363184e+00 4.87485380e+00  
2.68903226e+01 4.63111111e+01 6.94666667e+01 4.16800000e+02  
5.00160000e+02 5.00160000e+02]
```

Figura 5. Pesos por clase

Resultados y discusión

Primero debemos resaltar la importancia de la conversión o mapeamiento de los datos de entrada, pues esto nos significó un gran aumento en la precisión de todos los modelos que teníamos hasta el momento (20%). Luego de la utilización de los pesos de clase que también ayudó en buena medida a mejorar los resultados obtenidos.

Y finalmente de la utilización de `SCCE` en vez de `CCE`, los resultados son los siguientes:

Al utilizar `SparseCategoricalCrossEntropy` se obtuvo:

Sparse categorical accuracy : 98.26%

```
313/313 [=====] - 1s 3ms/step - loss: 30.0896 - sparse_categorical_accuracy: 0.9826  
[30.08958625793457, 0.9826052188873291]
```

Figura 6. Métricas SCCE

Al utilizar CategoricalCrossEntropy se obtuvo:

Accuracy: 97.73%

Recall: 97.73%

Precision: 97.72%

```
accuracy: 0.9773 - precision_2: 0.9773 - recall_2: 0.9772
```

Figura 7. Métricas CCE

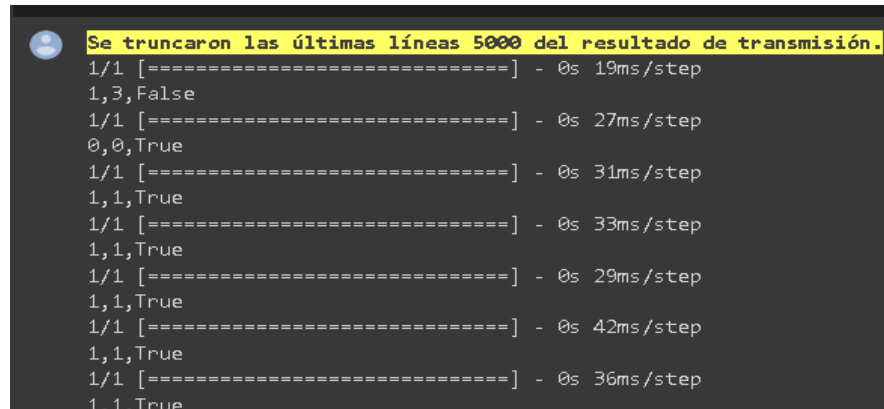
Con lo cual podemos hallar el f1-score :

F1 score

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r}$$

F1: 97.7249997442 %

Finalmente se implementó un método para la visualización de las predicciones junto con los valores esperados en tiempo real, para poder corroborar que el modelo en efecto predice las fortalezas de cada mano particular:



```
Se truncaron las últimas líneas 5000 del resultado de transmisión.
1/1 [=====] - 0s 19ms/step
1,3,False
1/1 [=====] - 0s 27ms/step
0,0,True
1/1 [=====] - 0s 31ms/step
1,1,True
1/1 [=====] - 0s 33ms/step
1,1,True
1/1 [=====] - 0s 29ms/step
1,1,True
1/1 [=====] - 0s 42ms/step
1,1,True
1/1 [=====] - 0s 36ms/step
1,1,True
```

Figura 9. Predicción en tiempo real

Conclusiones

Como conclusiones tenemos que:

Es importante escoger la función de pérdida correcta, pues algunos mostraron mejores resultados que otros al no ser compatibles con el tipo de clase. Un cambio en los parámetros de entrada simple como fue el mapeo de las manos originales puede afectar en gran medida a las métricas obtenidas, obteniendo un 20% de precisión extra (de 60% a 80%), teniendo en cuenta que este dataset tiene una característica especial, que por más que se intente balancear aún conserva un poco del dataset original desbalanceado, eso se puede equilibrar mapeando las manos y equilibrando los pesos según la importancia de las manos que nos encontramos en el dataset. Finalmente con la aplicación de técnicas de balanceo se puede obtener el 20% ,faltante así llegando a altísimos puntajes de precisión , accuracy y recall.

Contribución de autoría

Estith Bryan Vargas Quispe: [Conceptualización](#), [Análisis formal](#), [Investigación](#), [Visualización](#), [Metodología](#), [Software](#), [Validación](#), [Redacción - borrador original](#), [Curación de datos](#), [Escritura, revisión y edición](#). **Eybert Macedo Pillco:** [Conceptualización](#), [Investigación](#), [Visualización](#), [Metodología](#), [Software](#), [Validación](#), [Redacción - borrador original](#), [Curación de datos](#). **Quispe Ttito Juan Carlos:** [Conceptualización](#), [Análisis formal](#), [Investigación](#), [Visualización](#), [Metodología](#), [Software](#), [Redacción - borrador original](#), [Escritura, revisión y edición](#). **Jose Miguel Cano Vilcapaza:** [Conceptualización](#), [Análisis formal](#), [Investigación](#), [Visualización](#), [Metodología](#), [Software](#), [Validación](#).

Referencias

- [1] González, J. (2013, 07). Inteligencia artificial aplicada al poker texas hold'em. UCrea. Retrieved June 22, 2022, from <https://repositorio.unican.es/xmlui/handle/10902/3107>
- [2] Gascón, H., Bordallo, M., & Torres, E. (n.d.). Jugador Ganador de Poker Basado en Inteligencia Artificial. Retrieved June 23, 2022, from <http://www.it.uc3m.es/jvillena/irc/practicass/07-08/IA Poker.pdf>
- [3] Zamora Díez, Fernando. (2021). PokerRun: desarrollo de una aplicación web de póker online. Core.ac.uk. <https://core.ac.uk/display/459228542?source=2>
- [4] Superior, E., Detección, Y., Reconocimiento De Elementos En, M., De, P., Online, M., Jorge, G., Campo, Á., García, M., Ponente, María, J., & Sánchez, M. (2018). UNIVERSIDAD AUTONOMA DE MADRID TRABAJO FIN DE GRADO. https://repositorio.uam.es/bitstream/handle/10486/688137/g%C3%B3mez_campo_manuel_jorge_tfg.pdf?sequence=1
- [5] Profesor, I., Berlanga, A., Jesús, D., Luís, J., & Madrid, G. (n.d.). TRABAJO FIN DE GRADO Título: Diseño y evaluación de una heurística de juego de póker Autor: Jacobo Conrado Pérez-Fajardo Titulación: Grado en Ingeniería. https://e-archivo.uc3m.es/bitstream/handle/10016/16334/TFG_Jacobo_Conrado_Perez_Fajardo.pdf?sequence=1&isAllowed=y
- [6] Marco, G., Miguel, J., & Murillo, L. (2012). Diseño de Estrategias Óptimas en el Póker mediante Algoritmos Genéticos. https://e-archivo.uc3m.es/bitstream/handle/10016/16936/TFG_Gabriel_Marco_Angeles.pdf?sequence=4&isAllowed=y
- [7] Findler, N. V. (1977). Studies in machine cognition using the game of poker. Communications of the ACM, 20(4), 230–245. <https://doi.org/10.1145/359461.363617>
- [8] Catral, R., & Oppacher, F. (2007). Discovering rules in the poker hand dataset. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation - GECCO '07. <https://doi.org/10.1145/1276958.1277329>

- [9] Şirin, V., & Polat, A. (n.d.). A MACHINE LEARNING APPROACH TO THE POKER PLAYING PROBLEM. Retrieved June 23, 2022, from https://users.metu.edu.tr/e163109/Term_Paper_CENG_562.pdf
- [10] Da, B. (2018). Approximating Poker Probabilities with Deep Learning. <https://arxiv.org/pdf/1808.07220.pdf>
- [11] “Colaboración de Google.” <https://research.google.com/colaboratory/faq.html> (accessed Aug. 12, 2022)
- [12] “La librería Numpy | Aprende con Alf.” <https://aprendeconalf.es/docencia/python/manual/numpy/> (accessed Aug. 12, 2022).
- [13] “La librería Pandas | Aprende con Alf.” <https://aprendeconalf.es/docencia/python/manual/pandas/> (accessed Aug. 12, 2022).
- [14] U. O. de Catalunya, “Espacio de recursos de ciencia de datos.” <http://datascience.recursos.uoc.edu/es/preprocesamiento-de-datos-con-sklearn/> (accessed Aug. 12, 2022).
- [15] I. Challenger Pérez, Y. Díaz Ricardo, and R. Becerra García, “El lenguaje de programación Python/The programming language Python,” Rev. Ciencias Holguín, vol. 20, pp. 1–13, 2014.
- [16] “TensorFlow.” <https://www.tensorflow.org/> (accessed Aug. 12, 2022).
- [17] “Las Redes Neuronales Artificiales - Raquel Flórez López, José Miguel Fernández Fernández - Google Libros.” https://books.google.com.pe/books?hl=es&lr=&id=X0uLwi1Ap4QC&oi=fnd&pg=PA11&dq=redes+neuronales+&ots=gONwmsjqZl&sig=AMCEIhaM4AzG4aZctmXAyBjgkE&redir_esc=y#v=onepage&q=redes neuronales&f=false (accessed Aug. 17, 2022).
- [18] Juba, B., & Le, H. S. (2019, July). Precision-recall versus accuracy and the role of large data sets. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4039-4048).