

# **SESGO DE SUPERVIVENCIA EN EL POKER**

**DESARROLLADO POR ERNESTO MAGOLINI**

# RESEÑA HISTORICA

*Durante la Segunda Guerra Mundial, los Aliados cartografiaron los agujeros de bala en los aviones que fueron alcanzados por el fuego nazi con el objetivo de fortalecerlos. Su enfoque consistía en reforzar las áreas más afectadas por la artillería enemiga para aumentar la resistencia de los aviones frente a futuros ataques.*

*Sin embargo, Abraham Wald llegó a una conclusión diferente que desafiaba el pensamiento convencional. Según su razonamiento, las zonas de mayor impacto representaban el daño que los aviones que lograron regresar a casa pudieron resistir.*

*En lugar de reforzar las áreas dañadas, Wald argumentó que se debía prestar atención a las zonas que no mostraban impactos de bala. Estas áreas eran críticas, ya que los aviones que recibían disparos en esos puntos no lograban sobrevivir.*

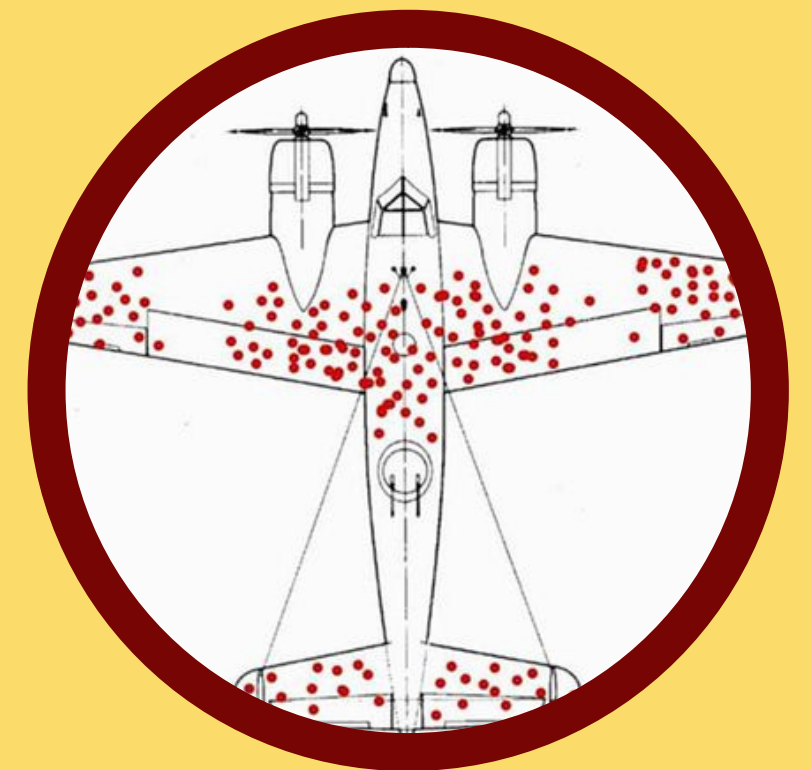
*Fuente: [imagecampus.edu](https://imagecampus.edu)*

# OBJETIVO

*Este fenómeno se conoce como sesgo de supervivencia, que ocurre cuando nos enfocamos en lo que ha sobrevivido en lugar de considerar lo que no lo ha hecho.*

*El enfoque toma la idea de que los datos disponibles solo representan una parte de la imagen completa y que se deben tener en cuenta las muestras faltantes para obtener conclusiones más precisas.*

*Usando esta analogía y aplicándola al poker Texas Holdem este proyecto tiene como objetivo el intentar predecir antes de iniciar cualquier acción si debemos 'abortar la misión' o seguir adelante, o en lenguaje de poker si debemos participar o no de la mano, utilizando la información de los registros históricos de manos jugadas.*



# BASE DE DATOS

*Los datos son obtenidos de una plataforma de póker online, en formato texto y tienen información sobre registros de manos y torneos de póker.*

*El desafío de esta etapa era convertir texto en datos estructurados tipo csv (o tablas) que representen características, para lo cual cree un programa que itera sobre el archivo texto, extraer las 62 variables por cada registro de texto, y devuelve un archivo csv con los registros.  
(conversor texto a csv.ipynb)*

*El siguiente paso fue crear una tabla para visualizar y organizar la base de datos usando herramientas y software del lenguaje de consulta SQL.*

# LIMPIEZA Y TRANSFORMACION DE LOS DATOS

## LIMPIEZA

*Para este proyecto se tomaron en cuenta solo las acciones del jugador antes de que se mostraran las primeras 3 cartas (flop). Además solo se considero a los torneos como como tipo de juego para el análisis.*

*Todas estas condiciones fueron filtradas y organizadas para trabajar solo con información relevante y específica.*

## TRANSFORMACION

*Las variables (o acciones) son de distinta naturaleza, numéricas o asociadas a clases, es por esto que las traducimos a números.*

*Después nivelamos los valores numéricos a la misma escala.*

*Esto pasos son fundamentales para que el modelo pueda interpretar los datos.de la mejor manera*



# MODELO Y PRUEBAS

*El problema que se plantea es el de predecir 3 clases o estados posibles, esto se considera un caso de clasificación.*

*En una fase preliminar de seis modelos probados elegí dos y que usan distintas lógicas (vecino cercanos trabaja por cercanía y árbol de decisión por discriminación basada en condiciones), ya que estos fueron los que tienen mejores métricas.*

*después de esta fase preliminar, se aplicaron diferentes técnicas y evaluaciones para mejorarlo y buscar los mejores resultados.*

*El modelo con mejor desempeño fue Random Forest que es una versión del árbol de decisión donde se entrenan varios de estos modelos en paralelo, usando muestras aleatorias con remplazo de los datos.*

# SOLUCIONES Y MEJORAS

*Considero que hay dos formas de mejorar el algoritmo que van de la mano, por una lado generar mas registros, y en consecuencia agregar una variable que considero fundamental, que son las cartas del jugador, el problema es que hay 1326 combinaciones posibles y la cantidad de registros que tengo es de apenas el doble, esto quiere decir que estadísticamente tengo 2 ejemplos de combinaciones posibles en los registros y el algoritmo no va a aprender de esta variable hasta que tenga la cantidad suficiente de registros.*

# SOLUCIONES Y MEJORAS

*Una técnica que puede verse beneficiada y mejorar considerablemente el modelo es una técnica llamada "smote" que consiste en crear datos sintéticos o dicho de otro modo crear mas ejemplos copiando los que ya tiene para reforzar el aprendizaje. Como se ve en la prueba 5 de ambos modelos es mas equilibrado a la hora de predecir las 3 categorías.*



# CONCLUSION

*El modelo es funcional en un % a la hora de predecir si debemos jugar la mano, pero es muy poco eficiente a la hora de predecir el resultado de una mano en la que vamos a participar.*

*Esto quiere decir que tenemos una herramienta que cumple con la esencia de este proyecto que es intentar identificar que manos debemos evitar jugar, enfocandonos en lo que anteriormente salio mal.*

# RECURSOS

## LENGUAJE DE PROGRAMACION

*SQL / Python / librerias: sklearn, pandas, numpy*

## BASES DE DATOS

*Pokerstars / Sqlite*

## PRESENTACION

*Canva*