UNIVERSIDAD DEL NORTE

Maestría en Ingeniería Electrónica

MODELADO Y SIMULACIÓN DE SISTEMAS

NOMBRE DEL ESTUDIANTE: Deivis de Jesús Martínez Acosta

Código: 7570727

Informe del trabajo final de curso

Título: Análisis del comportamiento de una lotería para la identificación de patrones presentes usando un modelo bayesiano.

Contexto introductorio

El tema del número ganador en una lotería ha sido objeto de estudio desde diversas disciplinas, investigando diferentes aspectos como los comportamientos de los ganadores, la posible relación con fechas o direcciones, entre otros. Sin embargo, existe un área poco explorada en la cual se centra este trabajo, que se basa en el carácter de aleatoriedad conocido como el azar inherente a las loterías.

Este trabajo examina un juego de lotería utilizando un enfoque basado en el teorema de Bayes, con el objetivo de detectar patrones en sus resultados. Según el modelo bayesiano los números con la mayor probabilidad de resultar ganadores estaban en un rango de -8.7 y -9.4. Se utiliza una metodología que puede ser aplicada en otros sistemas aleatorios, como accidentes de tráfico, el comportamiento de enfermedades o la criminalidad, entre otros.

El problema consiste en identificar cuatro grupos de seis números que tengan alta probabilidad de ser resultados ganadores en una lotería. Para ellos se emplea una adaptación de la metodología propuesta en el artículo usado como guía de este trabajo. Se analizan los resultados frente a la frecuencia de ellos y se analiza si existe algún patrón o tendencia de los datos.

https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642018000100019

Para la solución al problema propuesto se identifican los siguientes objetivos:

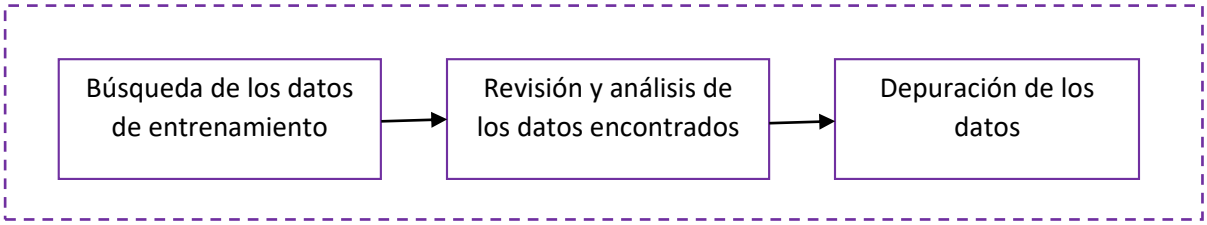
- Gestionar la base de datos de entrenamiento
- Construir y entrenar el sistema Bayesiano
- Generar la base de datos para la validación
- Validar el sistema realizado

Metodología

Seguimos cuatro pasos generales para alcanzar los objetivos: 1. Gestión de los datos de entrenamiento, 2. Construcción y entrenamiento del sistema bayesiano, 3. Generar la base de datos de validación y 4. Validar el sistema realizado.

Se procedió a aplicar cada paso de la metodología describiendo lo realizado.

Gestión de los datos de entrenamiento: Se siguen los pasos como se muestran el siguiente diagrama de bloques.



La búsqueda de los datos se realizó, para tener números de loterías reales, de los cuales se encontró para su libre utilización el archivo IsraeliLottery.csv, se hace la revisión de los mismos identificando que los datos de Game y Date, deben ser eliminados para tener depurado nuestros datos, como se muestra en la imagen 1.

	Game	Date	A	B	C	D	E	F
0	6801	03/09/1968	3	14	18	22	25	33
1	6802	10/09/1968	13	20	23	29	32	34
2	6803	17/09/1968	8	12	26	27	34	38
3	6804	24/09/1968	1	14	17	26	35	39
4	6805	01/10/1968	1	7	8	9	11	30

Imagen 1. Muestra de las 5 primeras filas de la matriz de datos

En la siguiente tabla 1, se muestra la organización de 4047 números granadores de la lotería cuya serie va de la “A” a la “F”, tomando así un valor de k = 6 correspondiente al número de balotas en cada sorteo.

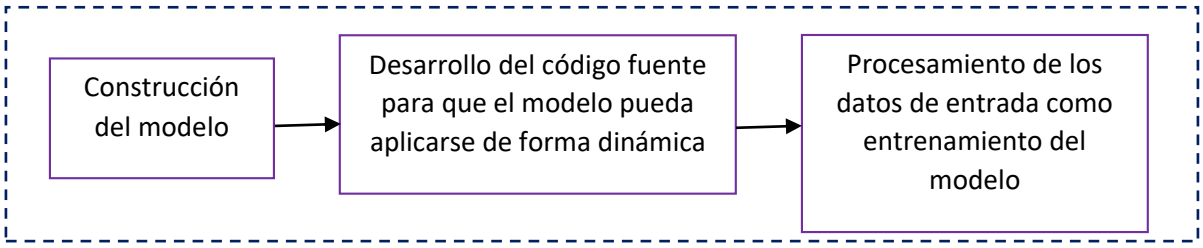
Sorteos	**Resultados**					
	A	B	C	D	E	F
1	3	14	18	22	25	33
2	13	20	23	29	32	34
3	8	12	26	27	34	38
…						

n = 4047	1	6	23	29	33	34

Tabla 1. Organización de los datos de entrenamiento del modelo

Para la **construcción y entrenamiento del sistema bayesiano**, se tomó como base el sistema de ecuaciones propuesto por Duda et al. (2001).

Se obtuvo como resultado el modelo propuesto $P_i(x) = X^t W_i X + w_i^t X + w_{io}$, donde $P_i(x)$ serán cada uno de los valores obtenidos de la evaluación de cada resultado ganador de la lotería usado como los datos de entrada. El valor de X equivale a la matriz con los valores de entrada o de entrenamiento. Cada X_i, es el valor de una fila con los k=6 valores. Para X^t es el valor de la matriz de entrada traspuesta. En esta etapa se realizaron los pasos como se muestra en siguiente diagrama.



El valor de W_i es calculado de la formula $W_i = -\frac{1}{2} \sum_i^{-1}$, donde $\sum_i^{-1}$ es la matriz de covarianza invertida, de los resultados.

En cuanto al valor de w_i^t es la traspuesta de w_i que es $w_i = \sum_i^{-1} u_i$, donde u_i es el vector de medias de la matriz X de entrada.

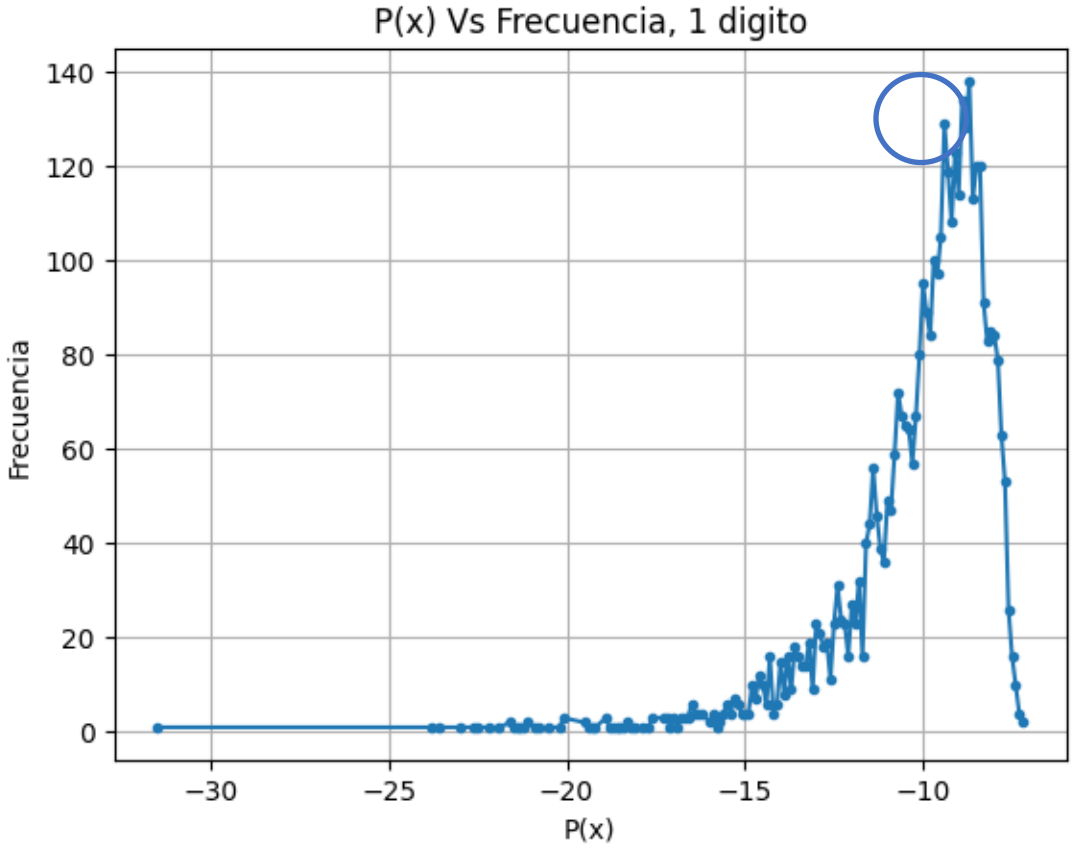
El valor de w_{io} está dado por: $w_{io} = -\frac{1}{2} u_i^t \sum_i^{-1} u_i \ -\frac{1}{2} \ln(|\sum_i|) + \ln(k)$ donde el valor de u_i^t , es la traspuesta del vector de medias y $|\sum_i|$, es el determinante de la matriz de covarianza, k para nuestro caso es 6 la cantidad de números (balotas) de hasta dos cifras que salen en un sorteo; los otros valores fueron descritos anteriormente.

Estos valores fueron aplicados como base de operaciones de los cálculos en el modelo $P_i(x) = X^t W_i X + w_i^t X + w_{io}$, se hizo la implementación en el lenguaje de programación Python y se calculó el valor de P(x) para cada uno de los 4047 sorteos de entrada que tenemos en la matriz X.

Para la **generación de datos de validación** se pueden seguir varias rutas, en este trabajo se tomaron dos: una de ellas es la descrita en el artículo guía (generar la combinatoria de M, k para tener los datos, excluyendo los repetidos y los ya

ganadores) y otra es generando de forma aleatoria estos valores, descartando los repetidos y evaluándolos enseguida. Para la segunda forma que requiere un poco más de tiempo de procesamiento por traer los datos de forma aleatoria y estos pueden coincidir repetidamente. Se generan solo números que se encuentren entre el mínimo y el máximo valor de cada columna, permitiendo que los valores de validación estén en el mismo rango que los valores de entrenamiento.

Encontramos los valores $P_i(x)$, pasando al modelo cada uno de los valores de los resultados de la lotería guardados en la matriz X, calculamos la frecuencia de los valores de $P_i(x)$ redondeados a 1, 2, 3, y 4 dígitos de precisión, obteniendo la gráfica1 para 1 dígito de precisión como también lo hace el artículo citado. donde encontramos una frecuencia de 138 para un valor de -8.7, también -8.9 con una frecuencia de 134, y -9.4 con frecuencia de 129.



Grafica 1. Valores P(X) Vs Frecuencia para 1 dígito de precisión

Las gráficas siguientes son a 2, 3 y 4 dígitos de precisión donde se pierde un poco la visibilidad especifica de la frecuencia, pero muestran el mismo comportamiento. Por ello se obtiene un valor de $P_i(x)$ promediado de los valores encontrados en el modelo entre -8.7004 y -9.3996, dando como resultado -9.05 que es el valor base para hacer la validación con los valores generados con los algoritmos diseñados.

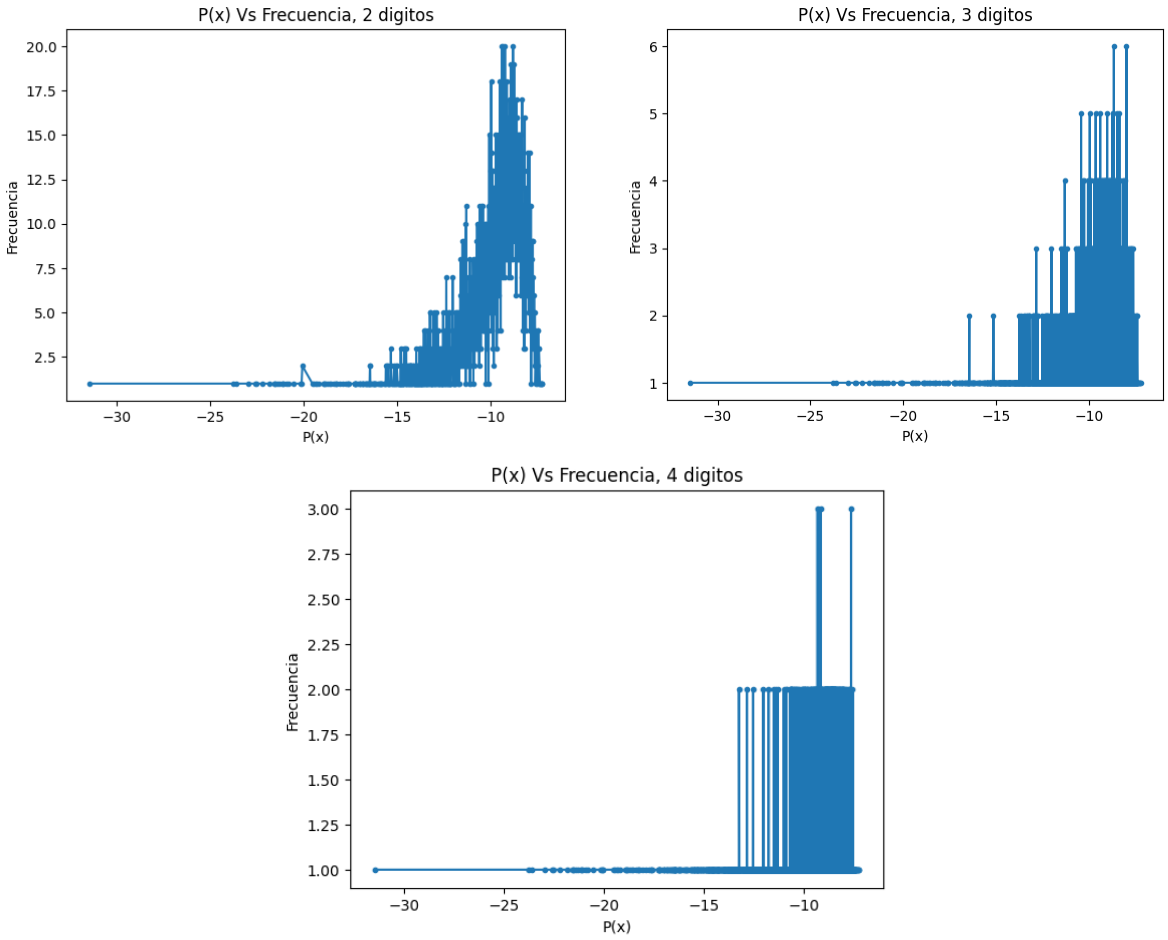


Grafico 2. Valores P(x) Vs Frecuencias para 2, 3 y 4 dígitos de precisión

En la **validación del sistema** se tomaron los datos generados, para el primer caso de generación, se recorren fila a fila los datos y se seleccionan los cuatro valores con mayor aproximación en el resultado P(x) al deseado que para nuestro caso es de -9.05 con un error permitido con valor absoluto de $1x10^{-2}$ y de $1x10^{-5}$, para cada algoritmo según la memoria del computador lo soportaba. Esto permitió tener los siguientes resultados:

Fue generada la descripción de los datos de entrenamiento donde encontramos sus características más relevantes como la cantidad de datos de cada columna verificando que cada una tiene 4047 registros, al igual que el vector de medias o las medias para cada una de las columnas, como también el valor mínimo y máximo presente en cada balota de los diferentes sorteos. Como se muestra en la Imagen 2.

	A	B	C	D	E	F
count	4047.000000	4047.000000	4047.000000	4047.000000	4047.000000	4047.000000
mean	5.698542	11.428713	17.128737	22.948357	28.795404	34.635285
std	4.635402	5.988067	6.698026	6.820599	6.616360	5.910514
min	1.000000	2.000000	3.000000	4.000000	7.000000	13.000000
25%	2.000000	7.000000	12.000000	18.000000	24.000000	31.000000
50%	4.000000	10.000000	17.000000	23.000000	29.000000	35.000000
75%	8.000000	15.000000	22.000000	28.000000	33.000000	38.000000
max	31.000000	37.000000	44.000000	46.000000	48.000000	49.000000

Imagen 2. Características de los datos de entrada contenidos en X

Estos datos son características de los sorteos de una lotería de Israel.

La matriz de covarianza es:

A	B	C	D	E	F
21.487	18.2934	15.5588	11.741	8.44696	5.82058
18.2934	35.8569	29.8012	22.9052	16.6132	11.108
15.5588	29.8012	44.8636	34.7669	25.4465	16.629
11.741	22.9052	34.7669	46.5206	33.8206	22.0125
8.44696	16.6132	25.4465	33.8206	43.7762	29.0645
5.82058	11.108	16.629	22.0125	29.0645	34.9342

La matriz de covarianza invertida es:

A	B	C	D	E	F
0.082352	-0.0407762	-0.00219944	0.000685256	0.00095294	-0.000933232
-0.0407762	0.0824804	-0.0407555	-0.000121579	0.000718456	-0.00055326
-0.00219944	-0.0407555	0.0811377	-0.0394125	-0.00115417	0.000497659
0.000685256	-0.000121579	-0.0394125	0.0781578	-0.0385759	0.00153114
0.00095294	0.000718456	-0.00115417	-0.0385759	0.0817984	-0.0435852
-0.000933232	-0.00055326	0.000497659	0.00153114	-0.0435852	0.064017

En cuanto a los resultados para el primer algoritmo donde se generan los números de las balotas aleatoriamente, se forma el vector de k=6 valores y se evalúan enseguida, se encontraron 4 números con 0.01 de margen de error debido a que la

capacidad de computo exigida es mucho mayor y al intentar tener una mayor precisión se paraba el programa, sin embargo con mayor capacidad de computo esta alternativa no exige procesar todos los posibles números para encontrar los adecuados con un margen de error permitido. Estos números son listados en la siguiente tabla 2:

Índice	A	B	C	D	E	F	P(x)
15454	6	17	26	30	34	43	-9.04983576242006
90828	3	16	23	32	36	39	-9.050828621513695
214120	6	13	16	27	35	44	-9.056288201622184
256381	9	15	26	29	34	33	-9.054465355942845

Tabla 2. Números con aproximación de $1x10^{-2}$ al valor deseado del modelo bayesiano

Para el caso donde se aplicó el algoritmo que genera los valores de las balotas de forma secuencial en una productoria M, k encontramos qué valores cumplen con un nivel de error $1x10^{-5}$ obteniendo como posible números ganadores los detallados en la siguiente tabla. Por capacidad de computo se tomaron los 4 primeros.

Índice	A	B	C	D	E	F	P(x)
6303667	1	4	13	18	24	26	-9.050002825246832
11989725	1	6	14	22	23	35	-9.050009432593232
20067599	1	9	9	17	28	32	-9.049999055255498
32034953	1	13	20	21	28	32	-9.050001256040236

Tabla 3. Números con aproximación de $1x10^{-5}$ al valor deseado del modelo bayesiano

En el análisis y discusión de los resultados podemos destacar dos aspectos importantes como vemos en la tabla 3, el tiempo de procesamiento de los datos para validación y aún la generación de los mismos es extensa, cuando se intentan generar más de $5x10^{7}$ sorteos o posibles combinaciones de 6 valores de balotas, un procesador Intel Pentium 7, con 8GB de memoria RAM y un disco de estado sólido se queda sin capacidad de computo, por ello podemos notar que los números de la tabla 3 siempre la primera balota “A” aparece el número 1, no se alcanzó la capacidad de cómputo para que la combinatoria llegara a la primera balota. Por esta razón se optó por el algoritmo anterior que genera los números de las balotas aleatoriamente, pero se encontró que aún consume mayor recurso.

Si analizamos de forma detenida los datos de la tabla 2 y la tres podemos notar que, aunque el margen de error es mucho mayor en la tabla 2 los números encontrados tienen características aleatorias para cada balota frente a los otros valores. Mientras que en la tabla 3, los valores están en un rango similar y se necesitó procesar aún en el rango de números de hasta 32 millones.

Adjunto a este informe programa en Python, el archivo de los datos de entrenamiento y el libro de google colab para la ejecución del programa al igual todos los archivos estarán compartidos en el siguiente enlace: https://github.com/deivismartinez/modelos