



Machine Learning para Secadoras

Materia:

Inteligencia Artificial Avanzada

Integrantes:

José Antonio Miranda Baños

Campus: Querétaro.

Fecha: 24 de agosto de 2024

Machine Learning Para Secadoras

José Antonio Miranda Baños

IRS, Instituto Tecnológico y de Estudios

Superiores de Monterrey

(Campus QRT)

Qrt, Mx

A01611795@tec.mx

Abstract— This decade (2020) has undoubtedly been the era of Machine Learning; it seems like every company is looking to implement this technology into their products. Machine Learning has proven to be a powerful tool in various industries, from automotive to healthcare. In the context of home appliances, especially dryers, the use of Machine Learning opens up new possibilities to enhance efficiency, optimize performance, and personalize the user experience.

Traditional dryers have evolved in terms of design and functionality, but the integration of Machine Learning algorithms promises to elevate these improvements to a new level. With the ability to analyze large volumes of data in real-time, dryers equipped with Machine Learning can predict usage patterns, adjust drying cycles based on specific load needs, and anticipate potential issues before they occur.

This article explores how Machine Learning can transform dryers by reviewing current applications, technical challenges, and future perspectives. Additionally, I discuss how implementing these technologies can not only increase energy efficiency but also provide greater comfort and garment care. Through a detailed analysis of case studies and recent advances, I aim to offer a comprehensive view of how Machine Learning is redefining the home appliance sector.

Keywords— Machine Learning, Regresión logística, MABE, Secadora

I. INTRODUCCIÓN

Esta década (2020) ha sido definitivamente la década del Machine Learning; pareciera que todas las empresas buscan implementar esta tecnología en cada uno de sus productos. El Machine Learning ha demostrado ser una herramienta poderosa en diversas industrias, desde la automotriz hasta la salud. En el contexto de los electrodomésticos, particularmente en las secadoras, el uso de Machine Learning abre nuevas posibilidades para mejorar la eficiencia, optimizar el rendimiento y personalizar la experiencia del usuario.[1]

Las secadoras tradicionales han evolucionado en términos de diseño y funcionalidad, pero la integración de algoritmos de Machine Learning promete llevar estas mejoras a un nuevo nivel. Con la capacidad de analizar grandes volúmenes de datos en tiempo real, las secadoras equipadas con Machine Learning podrán predecir patrones de uso, ajustar los ciclos de secado según las necesidades específicas de cada carga y anticipar problemas potenciales antes de que ocurran.

De modo que es solo cuestión de tiempo para empezar a ver más productos del tipo electrodoméstico con integraciones de Machine Learning

Este artículo explora cómo el Machine Learning puede transformar las secadoras, revisando aplicaciones actuales, desafíos técnicos y perspectivas futuras. Además, se discute cómo la implementación de estas tecnologías no solo puede incrementar la eficiencia energética, sino también ofrecer una mayor comodidad y cuidado para las prendas. A través de un análisis detallado de estudios de caso y avances recientes, se busca proporcionar una visión integral de cómo el Machine Learning está redefiniendo el sector de los electrodomésticos.

II. DESARROLLO

A. ¿Como funciona tu Secadora?

Aunque la pregunta puede parecer sencilla, entender el funcionamiento de una secadora es crucial para abordar cuestiones más profundas como el daño potencial a la ropa o la eficiencia en el manejo de grandes volúmenes de prendas. Las secadoras, a pesar de su diseño aparentemente simple, involucran una serie de procesos complejos que afectan tanto el resultado del secado como la durabilidad de las prendas.

El funcionamiento de una secadora es relativamente sencillo y se puede entender cómo que queremos separar el agua que quedo después de que la lavadora centrifugara nuestra ropa. La mejor solución para esto es calentarla para evaporar esta agua y extraer el vapor para ir lentamente reduciendo la cantidad de agua en nuestra ropa. Ahora hay muchas otras cosas involucradas que no considere en el resumen rápido de como secar ropa. Por ejemplo, no tomamos en cuenta que la ropa no puede estar junta o sino esta se acumula y esta no se llega a secar, que no podemos hacer girar el tambor muy rápido sino el aire no alcanza a llevarse nada de la humedad y la obstrucción del escape de tu secadora.

Estos y otros varios factores llegan a afectar el ciclo de secado de tu secadora y dañar tu ropa, por lo que no saber usar bien tu secadora afectara la vida útil de tus prendas.

B. Machine Learning para Secadoras

Al comenzar este proyecto, no anticipé la magnitud del impacto que el Machine Learning (ML) podría tener en las secadoras. Inicialmente, podría parecer que no hay necesidad de implementar modelos de ML en un electrodoméstico como una secadora, especialmente si consideramos que, a diferencia de una estufa, no estamos presentes para supervisar su funcionamiento. Sin embargo, al profundizar en el análisis, se vuelve evidente que las secadoras, al ser herramientas "Non-Supervised", se benefician enormemente de la integración de modelos de ML.

1. Definición de "Non-Supervised" y "Supervised"

Non-Supervised: En el contexto de los electrodomésticos, una máquina "Non-Supervised" es aquella que realiza una tarea de manera autónoma sin la necesidad de supervisión continua. Un ejemplo de esto es la secadora. Una vez que introduces la ropa y seleccionas un programa, la secadora completa el ciclo sin intervención adicional. Esta falta de supervisión directa la hace candidata ideal para la integración de ML, que puede optimizar su funcionamiento y mejorar los resultados sin la necesidad de monitoreo constante.

Supervised: En contraste, herramientas "Supervised" como la estufa requieren la presencia del usuario para ajustar parámetros en tiempo real. Dado que el usuario está activamente involucrado en el proceso (como cocinar), no hay una necesidad inmediata de implementar ML para gestionar la operación del aparato, ya que el control manual y la supervisión directa cumplen esa función.

III. IMPLEMENTACIÓN

El desafío es claro: desarrollar la secadora ideal, una máquina que pueda secar eficientemente cualquier tipo de ropa sin importar su naturaleza, al punto de que entregar la ropa doblada sea casi una muestra de vanidad. Para alcanzar este objetivo, nos proponemos utilizar técnicas avanzadas de Machine Learning, específicamente una regresión logística, para identificar y ajustar el ciclo de secado adecuado para cada tipo de prenda.

1. Definiendo la Secadora Ideal

El concepto de la secadora ideal implica una máquina capaz de adaptar su ciclo de secado a las características específicas de cada carga de ropa, asegurando resultados óptimos sin necesidad de intervención manual. Esto requiere un sistema que pueda:

Detectar el Tipo de Prenda: Identificar las características de las diferentes prendas (tipo de tejido, grosor, etc.) para ajustar el ciclo de secado en consecuencia.

Ajustar el Ciclo de Secado: Modificar parámetros como la temperatura y la duración del ciclo según las necesidades específicas de la ropa.

Optimizar el Desempeño: Asegurar que cada ciclo de secado sea eficiente en términos de tiempo y consumo energético, manteniendo la calidad de las prendas.

2. Implementación de la Regresión Logística

Para alcanzar estos objetivos, implementaremos un modelo de regresión logística, que es adecuado para la clasificación binaria y multiclase. Este modelo ayudará a identificar el tipo de ciclo que la secadora debe utilizar basándose en las características de las prendas. Los pasos clave en la implementación son:

Recopilación de Datos: Utilizaremos conjuntos de datos reales proporcionados por la empresa MABE. Estos datos incluyen información detallada sobre diferentes tipos de prendas y los ciclos de secado asociados, permitiendo al modelo aprender las relaciones entre las características de la ropa y los ciclos de secado óptimos.

Preparación de Datos: Los datos se procesarán para asegurar que estén limpios y bien estructurados. Esto incluye la normalización de variables y el manejo de datos faltantes o anomalías.

Entrenamiento del Modelo: Se entrenará la regresión logística utilizando los datos recopilados. El modelo aprenderá a clasificar las prendas en función de sus características y predecir el ciclo de secado más adecuado.

Evaluación del Modelo: Se evaluará la precisión del modelo utilizando métricas de rendimiento como la exactitud, la precisión y la recuperación. Esto ayudará a ajustar los parámetros y mejorar la capacidad del modelo para generalizar a nuevas muestras.

3. Beneficios y Futuro

La implementación de la regresión logística en la secadora no solo promete mejorar la eficiencia del secado y la protección de las prendas, sino que también abre la puerta a futuras mejoras. Con el tiempo, el modelo puede refinarse y ajustarse utilizando datos adicionales y técnicas avanzadas de Machine Learning, como el aprendizaje profundo, para mejorar aún más la precisión y la adaptabilidad de la secadora.

IV. ETL

ETL, Extract Transform Load por sus siglas en inglés, en esta sección explico como pase de varios archivos de Excel donde se registraban los resultados de diversos sensores instalados en una secadora, ah un Data Set que podemos usar para entrenar nuestra red logística.

A. Leyendo el experimento

Como mencioné el raw data me fue entregado en un formato tosco y complicado de leer, Excel tiene sus limitaciones a la hora de manejar grandes volúmenes de datos, sobre todo cuando se hace de manera incorrecta, es por esto que me dieron varios "Libros" de datos, y cada página de estos correspondía a un experimento o prueba realizada con la secadora.

	A	B	C	D	E	F	G	H	I	J	K
1	Time(mn)	RMC	ROB	Energy	Potencia	Filtered	Smooth	Outlet	Inlet	T-A amb	RH amb
2	0	100.0	82	0.2	856.5	0	0	81	85	76.9	46.9
3	0.013	100.0	82	1.8	5477	0	0	81	85	76.8	46.7
4	0.03	100.0	82	3.3	5442.8	0	0	81	85	75.9	46.3
5	0.047	100.0	82	4.6	5374.8	0	0	82	85	75.5	45.8
6	0.063	100.0	82	6.3	5371.2	0	0	81	85	77.3	47.2
7	0.079	100.0	82	7.7	5320.7	0	0	81	85	76.1	46.5
8	0.096	100.0	82	9.2	5247.4	0	0	81	86	76.3	46.6

En total serian alrededor de 2721480 filas, por lo que primero decidí juntar todos estos libros en un solo data frame de Pandas, para poder agilizar cualquier operación que decidiera aplicar a esta información.

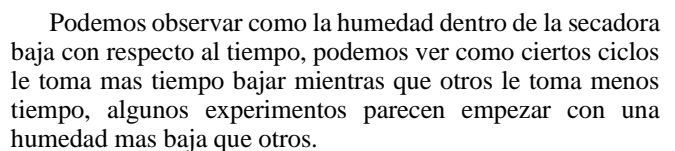
Este código recopila todos los libros, busca las páginas donde haya información de los experimentos, lee la ficha de cada experimento y recupera la masa de ropa y la restricción del flujo de aire del experimento. Después recupera las columnas que contiene la información del experimento, agrega 'Clase' y 'Segmento' que nos permite identificar a que ciclo y prueba de secado pertenece y concatena todo esto a un archivo CSV. Este proceso nos otorga una serie de CSV que podemos analizar de forma más cómoda.

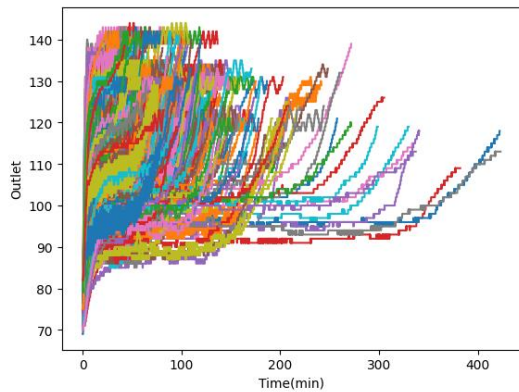
Ahora con los datos extraídos tocara decidir que variables son importantes o no para nuestra regresión logística.

Para poder filtrar mas nuestro Data Set, hay que entender que representa cada variable de este. De esta manera podemos descartar que vale la pena mantener, que puede afectar y que no puede afectar el comportamiento de la secadora con cierto tipo de ropa.

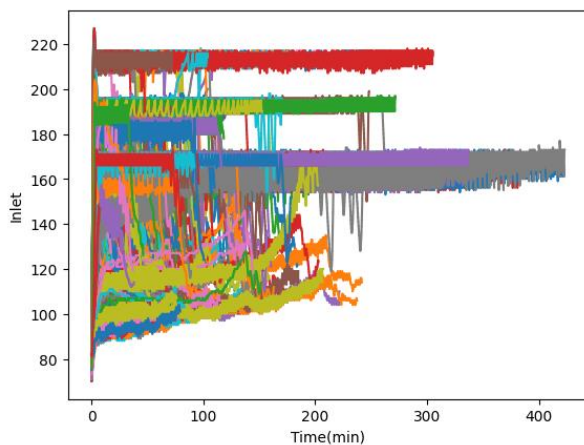
Feature	Explicación
Time(min)	Se refiere al tiempo en minutos.
RMC	La humedad de la ropa, cuando esta llegue a un rango entre 4 y 2 % podemos considerar la ropa como seca.

Tal vez valdría la pena revisar el comportamiento de algunos de estas variables con respecto al tiempo para saber que es lo que realmente sucede con ellos durante el ciclo de secado.

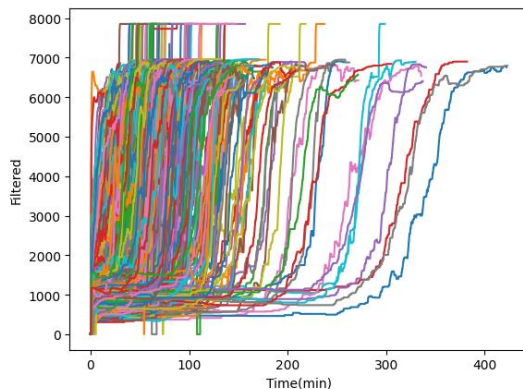




La temperatura de 'Outlet' nos muestra tiende a subir, considerando que mientras la humedad baja, no hay nada que absorba la temperatura de 'Inlet' tiene sentido que esta suba. Vemos como 'Outlet' coincide con 'RMC' en el tiempo, habiendo ciclos que duren más y ciclos que duren menos.



En la grafica de 'Inlet' vemos como se asemeja a un sistema de control amortiguado que se mantiene oscilando, aunque intentan llegar a un Set Point, parece que dependiendo el ciclo y el peso el Set Point y las oscilaciones son diferentes.



Podemos observar como dependiendo del ciclo la señal de 'Filtered' se comporta de forma diferente, durando mas o teniendo una pendiente mucho mas pronunciada, aunque esta algo amontonada parece presentar mas diferencias cuando se trata de ciclos de secado.

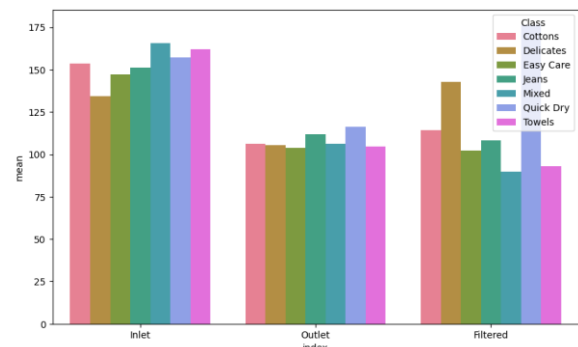
Aunque un poco saturadas estas graficas nos ayudan a comprender de forma general cómo se comportan estas 4 variables. Del RMC nos queda claro que siempre decae con el tiempo, lo que explica el comportamiento de Outlet, mientras menos humedad hay en el ambiente el aire a la salida de la

secadora debería de tener una temperatura más alta. Por otro lado, el comportamiento de Inlet en un inicio parece independiente, puesto que trata de llegar a un setpoint dependiendo el ciclo y viéndose afectada por la masa de ropa, algo que podríamos tomar es su comportamiento una vez llega a su setpoint, ya que los picos se comportan de forma distinta. Por último, tenemos Filtered cuyo parece ser contrario al del RMC, lo que nos indicaría que este aumenta conforme menos humedad se encuentra en la secadora.

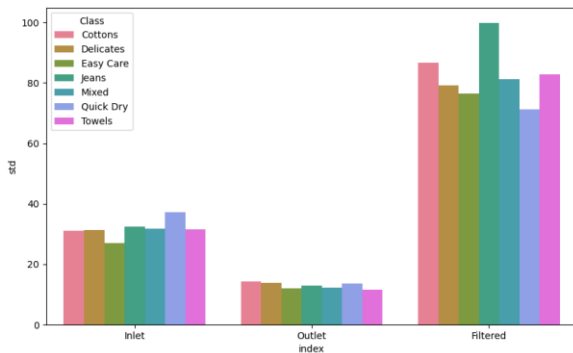
Después de revisar el comportamiento de cada variable, nos podemos dar una idea de que variables nos pueden ayudar a determinar la clase o el tipo de ropa dentro de la secadora. Por lo que conservaremos 'RMC', 'Inlet', 'Outlet', 'Time(min)' y 'Filtered'. Aun ocuparemos Features como 'Weight' o la 'Restricction' para enviar nuestras muestras en rangos específicos. En cuanto a la variables como la 'T-A Amb', 'RH Amb' las descartamos por la poca diferencia que hay entre las muestras y los experimentos de modo que solo sesgarían al modelo al repetirse o tendrían un peso muy bajo. 'Energy' y 'Potenza' las descartamos por recomendación de MABE, nos comentaron que los sensores encargados de medir estas variables realmente no median algo significativo ('la potencia del motor o de algún sistema del cual podamos inferir la clase de ropa).

Ah simple vista no parece que debamos quitar o arreglar alguna parte del Data Set, aunque tuvimos que intervenir en varios Excel para encontrar valores como la restricción o la masa del experimento.

Ahora toca decidir con que conjunto de datos probar nuestra regresión logística. Sería una buena idea encontrar datos que estadísticamente fueran diferentes, por lo que sería buena idea compara las descripciones estadísticas de nuestros experimentos.



Media de temperatura en 'Inlet' y 'Outlet' de los distintos ciclos de secado, se pueden notar diferencias en la mayoría de los ciclos, podemos ver que los ciclos de 'Cottons' y 'Jeans' son muy similares. 'Mixed' es el ciclo con la temperatura media mas alta y 'Delicates' el ciclo con la temperatura media mas baja. 'Filtered' por otro lado parece diferir más en su media aunque 'Mixed' presenta cierta similitud con 'Towels' pero difiere con 'Cottons' mientras que 'Quick Dry' presenta una media más alta y 'Mixed' y 'Towels' la mas baja.



Podemos ver como la desviación estándar para 'Filtered' es muy alta en general, siendo la de Jeans la más alta y 'Quick Dry' la mas baja. Aparte de esto podemos resaltar lo variada que es, volviendo a 'Filtered' una buen Feature de nuestro Data Set. 'Inlet' aquí se ve muy similar, lo que tiene sentido considerando que trata de llegar a un Set Point y mantenerse ahí, pareciéndose 'Cottons', 'Delicates', 'Jeans', 'Mixed' y 'Towels' siendo 'Quick Dry' y 'Easy Care' las mas diferentes del promedio.

Outlet en estos estadísticos es un caso especial, puesto que si bien muestra algo de variación la mayoría se encuentra en un rango de valores muy limitado, esto podría sesgar nuestro modelo de ML, por lo que valdría la pena quitarlo.

La secadora convencional, cuenta con sensores 'Inlet', 'Outlet' y 'Filtered' es por eso que mantendremos estos valores para nuestro modelos de ML, puesto que queremos que este modelo puede ser ocupado para la secadora ideal.

Abajo dejo las descripciones estadísticas de cada ciclo de secado, por si gustas revisarlo, aunque me parece que las graficas te darán una mejor idea de estos datos.

	Cottons		
	Inlet	Outlet	Filtered
count	312120.0	312120.0	312120.0
mean	153.73	106.38	114.27
std	30.98	14.32	86.78
min	70.0	71.0	0.0
25%	126.0	96.0	45.12
50%	162.0	103.0	72.72
75%	171.0	118.0	179.6
max	222.0	144.0	314.04
	Delicates		
	Inlet	Outlet	Filtered
count	259396.0	259396.0	259396.0
mean	134.21	105.36	142.77
std	31.25	13.72	79.24
min	72.0	69.0	0.0
25%	104.0	95.0	67.04
50%	133.0	103.0	131.52
75%	166.0	117.0	215.52

max	187.0	133.0	314.04
	Easy Care		
	Inlet	Outlet	Filtered
count	349441.0	349441.0	349441.0
mean	147.18	103.72	102.15
std	26.86	11.92	76.55
min	72.0	70.0	0.0
25%	124.0	95.0	43.92
50%	156.0	101.0	65.48
75%	167.0	111.0	157.56
max	207.0	137.0	314.04
	Jeans		
	Inlet	Outlet	Filtered
count	621711.0	621711.0	621711.0
mean	151.32	112.01	108.18
std	32.47	12.97	99.92
min	71.0	70.0	0.0
25%	121.0	102.0	26.16
50%	160.0	111.0	57.8
75%	171.0	121.0	204.04
max	225.0	144.0	314.04
	Mixed		
	Inlet	Outlet	Filtered
count	624724.0	624724.0	624724.0
mean	165.61	106.23	89.86
std	31.73	12.17	81.21
min	73.0	71.0	0.0
25%	152.0	100.0	33.24
50%	169.0	103.0	48.32
75%	188.0	112.0	124.4
max	226.0	143.0	314.04
	Quick Dry		
	Inlet	Outlet	Filtered
count	114033.0	114033.0	114033.0
mean	157.04	116.33	176.59
std	37.15	13.47	71.2
min	71.0	70.0	0.0
25%	127.0	106.0	114.88
50%	165.0	118.0	183.88
75%	185.0	128.0	245.6
max	227.0	143.0	275.44
	Towels		
	Inlet	Outlet	Filtered
count	440055.0	440055.0	440055.0
mean	162.02	104.46	93.07
std	31.48	11.57	82.79

min	72.0	70.0	0.0
25%	151.0	98.0	39.2
50%	167.0	102.0	49.0
75%	185.0	110.0	117.88
max	227.0	143.0	314.04

Con estos datos nos podemos dar cuenta de las diferencias y similitudes que hay entre los ciclos de lavado, por ejemplo nos damos cuenta que el ciclo de 'Cottons' es estadísticamente muy similar a 'Jeans' así mismo con 'Mixed' y con 'Towels', por lo que convendría juntarlos dentro de un mismo Data Set como una sola clase, sin embargo hacerlo haría un data set muy grande que seguramente terminaría sesgando el modelo.

Sería una buena idea entrenar un modelo de ML con datos diferentes, para poder clasificarlos mejor. Por lo que proponemos usar el Data Set de 'Jeans' y 'Delicates' puesto que son muy diferentes por lo que podemos esperar buenos resultados de nuestro modelo de clasificación haciendo uso de una regresión logística.

V. REGRESIÓN LOGÍSTICA

Hay que aclarar que la regresión logística nos regresará un valor binario dependiendo la clase que identifique, por eso mismo probaremos haciendo uso de solo dos clases, 'Delicates' y 'Jeans'.

El Data Set que generamos en la parte anterior lo dividiremos primero en 2 partes, 'Train' y 'Test'. De esta forma evitaremos que nuestra regresión nos de valores memorizados y que pueda medirse con valores que simulan la realidad.

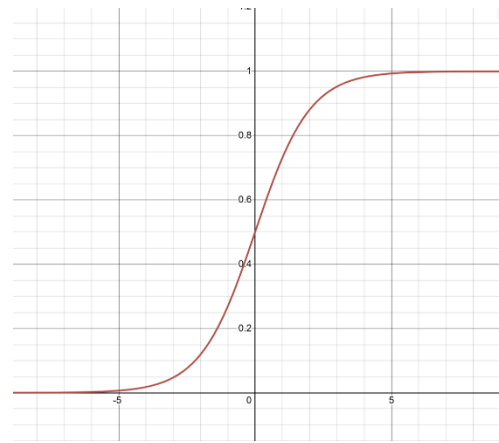
A. Función Hipótesis

Teniendo nuestros Data Sets listos ahora nos centraremos en armar nuestro modelo de Machine Learning. Una regresión Logística que aprenda a discriminar si la carga dentro de nuestra secadora son Jeans o si son Delicados. Lo primero que haremos será definir nuestra función hipótesis.

$$z = \theta_i X_i + b$$

A siempre vista parece una regresión lineal... Es una regresión lineal, donde tomamos las muestras, las multiplicamos por sus respectivos parámetros y le sumamos un bias. Esta función hipótesis será evaluada por una función sigmoide para obtener un resultado binario. Quedando así.

$$S = \frac{1}{1 + e^{-z}}$$



Podemos notar como esta función convierte los resultados de nuestra regresión a un rango de valores entre 0 y 1 convirtiendo la salida en binaria. Aunque se puede dar el caso en que nuestro modelo tenga problemas para identificar las diferencias y nos de valores entre 0 y 1, en nuestro caso sucedió esto, dando un 'Accuracy' de 0, puesto que nunca te dará un valor 0 o 1, sino que se quedan entre esos valores. Para solucionar esto redondeamos las salidas del modelo después de entrenarlo, hacerlo durante el entrenamiento pondrá a oscilar a la función de costo y nunca convergerá o podrías caer en el caso que busque el logaritmo de 0 falle el programa.

B. Descenso por Gradiente

Ya que tenemos nuestra función de hipótesis, inicializada con parámetros aleatorios, calculamos la diferencia entre sus predicciones y las predicciones reales. Ya que nos dimos cuenta que tanto nos equivocamos, sacaremos el gradiente de nuestros parámetros. En nuestro caso lo hacemos de forma matricial para hacerlo de forma rápida.

$$\theta := \theta - \alpha \frac{1}{m} X^T (h_{\theta}(X) - y)$$

Esta ecuación ajusta los parámetros de forma matricial. Donde restamos los parámetros actuales (theta) menos la multiplicación del 'learning rate' (Alpha) entre la cantidad de muestras por el producto punto de la matriz transpuesta por el vector de errores. Moviendo la Theta nueva en la dirección del gradiente negativo, con el objetivo de mejorar el error.

$$b := b - \alpha \frac{1}{m} (h_{\theta}(X) - y)$$

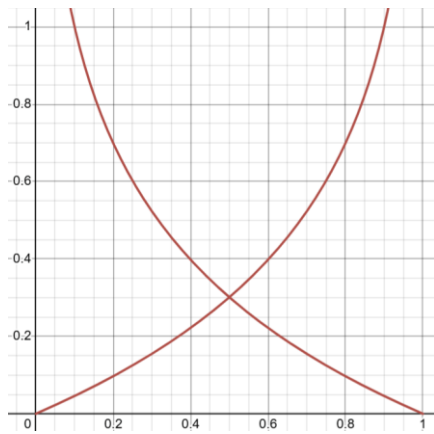
Para actualizar nuestro bias basta que restemos el actual menos el 'learning rate' por el promedio del error, como trabajamos el mismo como un vector no hace falta transformar su forma o algo parecido.

C. Loss Function

Ahora debemos medir el desempeño de nuestro modelo, para esto ocuparemos una función de pérdida. Puesto que para minimizar la función de pérdida debemos ajustar nuestros parámetros y así llegar a predicciones más acertadas. En el caso de nuestro modelo usamos la función de entropía cruzada.

$$L(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Donde 'y' serían nuestros valores reales y 'y_hat' nuestras predicciones. Entender esto será más fácil si vemos la gráfica que se genera de esta función.



Donde el Eje X representa la probabilidad predicha por el modelo y el Eje Y representa la función de pérdida.

El objetivo sería llegar al punto mínimo de esta función compuesta por dos curvas, la curva convexa que comienza en (0,1) y termina en (1,0) y la curva cóncava que comienza en (0,0) y termina en (1,1).

La curva convexa representa cuando el valor verdadero es igual a 1, de modo que mientras más te alejas de la respuesta correcta la entropía es menor, y mientras más te acercas la entropía aumenta.

Del mismo modo la curva cóncava representa la entropía cuando el valor verdadero es 0, de modo que mientras más te acercas a 0 la entropía es mayor y mientras más te alejas la entropía es menor.

Para conseguir un modelo que pueda predecir de manera óptima buscamos que la entropía sea baja. []

D. Accuracy

Con la función de pérdida podemos saber que tan bien o mal está entrenando nuestro modelo. Aunque hay que resaltar que no es una métrica interpretable para eso compararemos los resultados las predicciones de nuestro modelo contra los valores reales, obteniendo así el 'Accuracy'. Este nos brinda una métrica más clara de que tan acertado o no es nuestro modelo.

Para conocer el 'Accuracy' de forma práctica, tomaremos una porción de los datos de Train, una porción pequeña, puede variar dependiendo del tamaño del Data Set. A este subconjunto de Train lo conoceremos como 'Validation'. Con este Data Set probaremos nuestros nuevos parámetros y compararemos la cantidad de aciertos con el número de samples para conocer que tan acertado es el modelo. Aunque de igual manera haremos esta prueba después de entrenar el modelo esta métrica nos servirá para saber si nuestro modelo es útil o si solo está memorizando el Data Set de 'Train'.

Repetiremos todo lo anterior dicho, actualizar parámetros, el valor de la pérdida o la entropía de nuestro modelo y la validación hasta que se cumplan una de dos condiciones.

- Logramos llegar a un mínimo en nuestra función de costo.
- Llegamos al límite de épocas en las que repetiríamos este proceso.

Dependiendo del Data Set, de que tan bien limpiados y del problema que representan los datos, podríamos llegar a un punto mínimo antes de que se terminen las épocas. Aunque siempre cabe la posibilidad de quedarnos atorados con una pérdida muy alta. Esto ocurre principalmente cuando al Data Set le faltan transformaciones o simplemente hay muy poca relación entre los 'Features' y la Y. En todo caso te recomendamos mejorar o modificar tu Data Set.

Ahora que ya tenemos un modelo toca probarlo. Para esto pasaremos el Data Set de 'Test' por la última función de hipótesis que genere nuestro entrenamiento.

Las predicciones que nos genere este modelo son de gran importancia. Comparando el 'Accuracy' de 'Validation' con el que obtengamos de las predicciones de 'Test' podemos caer en alguno de los siguientes casos.

- $Ac_v > Ac_T$ El modelo presenta 'Overfitting'
 - El modelo memorizó los datos de entrenamiento y no predice nada fuera de ese conjunto de datos.
 - Para este caso lo mejor es cambiar la proporción de 'Train' y 'Test' y asegurarse que el escalamiento es igual en ambos Data Sets.
- $Ac_v \leq Ac_T$ Que el 'Accuracy' en 'Test' sea igual o mayor que en 'Validation' es algo deseable, significa que nuestro modelo se desempeña igual o mejor en casos cercanos a la realidad.

E. Diagnostico del modelo

Aunque el 'Accuracy' es nuestra mejor métrica para saber que tan acertado es el modelo no nos habla mucho sobre él. Con esto nos referimos a saber que clases predice mejor nuestro modelo y que tantos datos falsos y verdaderos genera.

Estas métricas nos pueden ayudar a diagnosticar nuestro Data Set y encontrar sobre representación de alguna clase o ayudarnos a darnos cuenta si una clase no presenta relación con los 'Features'.

Para esto hacemos una matriz de confusión, que tiene la siguiente forma.

		Predicted:		
		NO	YES	
Actual:	NO	TN = 50	FP = 10	60
	YES	FN = 5	TP = 100	105
		55	110	

Esta matriz nos permite conocer la proporción de falsos en nuestro modelo, lo que nos permite saber que tan bien clasifica y también ver mejor la distribución de las clases que predice bien y que predice mal. En este ejemplo podemos ver que alrededor de 150 casos los clasifico bien mientras que se equivocó en 15. De 55 'No' que predijo 50 son correctos y 5 son falsos positivos, de 110 'Yes' predijo 100 correctos y 10

son falsos negativos. En este caso podemos darnos cuenta de que el modelo esta un poco cargado de más para 'Yes' por lo que tiene sentido que falle menos identificando esta clase y por lo que confunda más 'No' con esta misma. El detalle esta en la proporción de datos que clasifica de forma errónea.

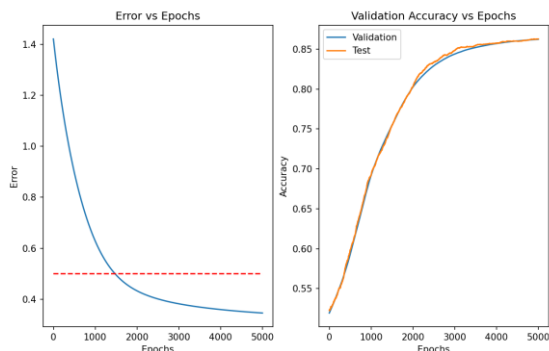
Conociendo la proporción de Falsos Positivos y de Verdaderos Positivos podemos generar una curva ROC. La curva ROC es una herramienta que nos permite visualizar el rendimiento de un modelo de clasificación binaria. Donde cada punto de la curva representa el par de ambas proporciones, correspondiente a un umbral de decisión. De modo que mientras el área bajo la curva sea más cercana a 1 nuestro modelo tendrá un mejor rendimiento, el caso contrario, si el área bajo la curva es cercana a 0.5 indica que nuestro modelo no es mejor que una clasificación aleatoria.

VI. RESULTADOS

Haciendo uso del Data Set que armamos en la parte IV y de un código en Python armado con la descripción de regresión logística mostrada en la parte V conseguimos los siguientes resultados.

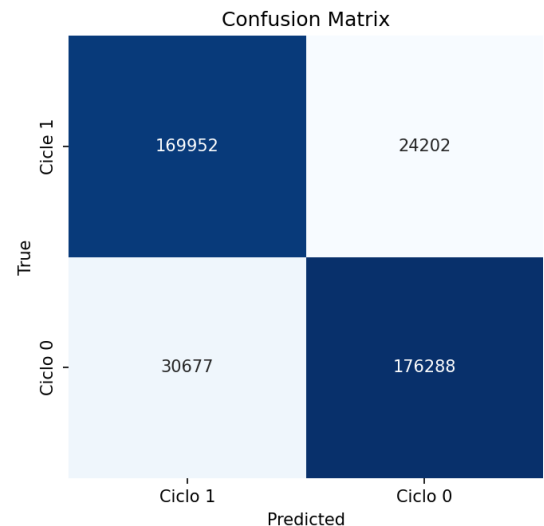
A. Diagnostico de Entrenamiento y Validación

Ocupando el Data Set formado por Jeans y Delicates, con un learning rate de 1e-2 durante 5000 epocas obtenemos los siguientes resultados.



Epoch:	Loss:	Validation Accuracy:
99.10%	34.35%	86.32%

Completando cerca del 99% de épocas, con una pérdida del 34.35% y un 'Accuracy' del 86.32% podríamos decir que nuestro modelo se comporta de forma decente cuando se trata de clasificar el conjunto junto de Validación. Si revisamos la grafica de 'Accuracy' podemos decir que tenemos un buen modelo, que no cae en **OverFitting** ni **UnderFitting** teniendo un alto 'Accuracy' y una delta baja entre las curvas de validación.



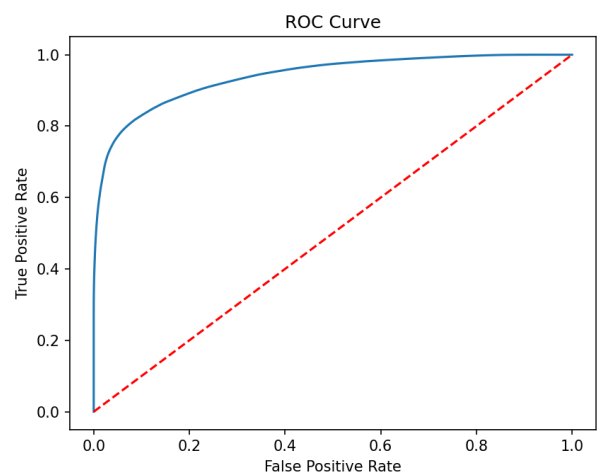
El modelo clasifica correctamente el 87% de las veces

De todas las predicciones positivas realizadas por el modelo, el 85% son correctas.

El modelo identifica correctamente el 88% de los casos positivos reales.

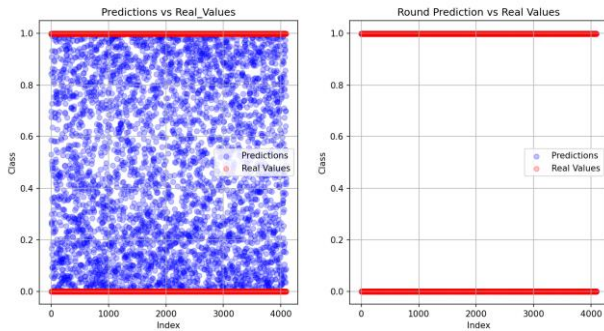
El modelo identifica correctamente el 85% de los casos negativos reales.

Viendo la matriz de confusión podemos decir que el modelo es bastante efectivo para distinguir entre ciclos de secado, tiene un equilibrio bueno entre la precisión y la sensibilidad. Lo cual lo vuelve en un modelo lo suficientemente confiable para dejarlo encargado del cuidado de la ropa.

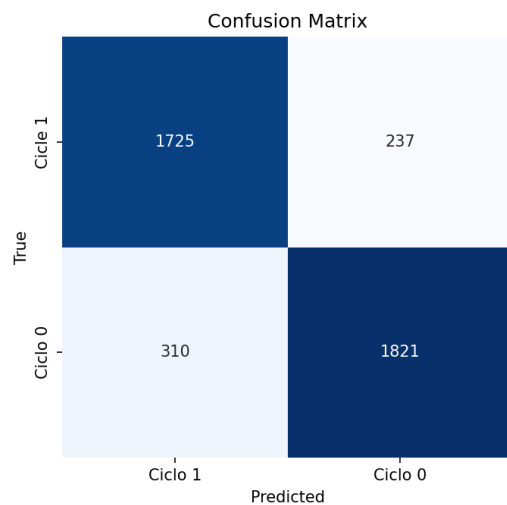


Ahora visualizando la curva ROC con un área bajo la curva de 0.93. Esta es una excelente métrica, significa que tenemos un modelo con alta capacidad discriminativa, lo que significa que en general tendremos pocos falsos.

B. Diagnostico de Prueba



Lo que estas viendo es como se comporta el modelo cuando tratamos de clasificar el Data Set de 'Test'. Logrando un 'Accuracy' de 86.6%, muy parecida al del conjunto de validación.



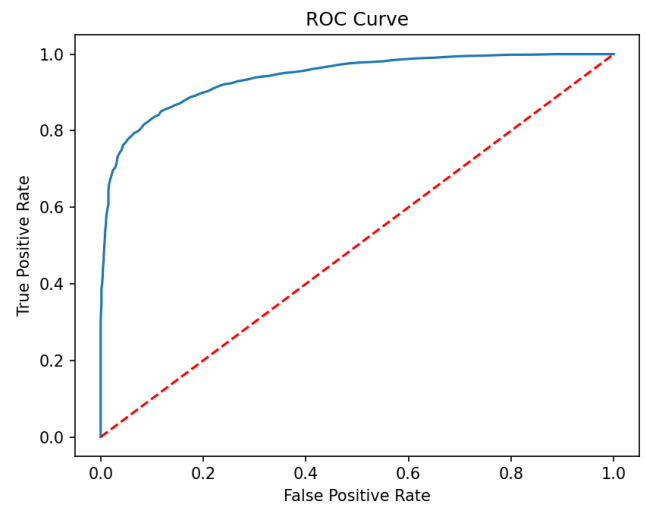
La matriz indica que el modelo tiene una precisión del 87% en el conjunto de prueba.

Alrededor del 85% de las predicciones positivas del modelo son correctas.

El 88% de los casos positivos son reales.

El 85% de los casos negativos son reales.

Nos podemos dar cuenta que los resultados de la matriz son muy parecidos a los resultados obtenidos en la matriz del conjunto de validación.



Por ultimo nos encontramos con la curva ROC la cual a simple vista se nota muy parecida a la anterior. Lo cual es bueno puesto que significa que nuestra modelo no varía mucho a la hora de predecir datos reales.

Teniendo un área bajo la curva de 0.94 tienen un índice de discriminación incluso mejor, al menos para el conjunto de datos correspondiente a 'Test'. Se podría decir que el modelo puede trabajar sin mucho problema con datos fuera de su entrenamiento.

VII. CONCLUSIÓN

Nuestro modelo en general es muy acertado, considerando que ocupa tan solo 3 'Features' del Data Set original esto es muy bueno, puesto que lo vuelve una solución económica al no tener que incluir mas sensores de los que ya se tienen integrados dentro de la secadora. Es verdad que nuestro modelo tiene valores altos debido a que los datos que ocupamos presentaban una alta diferencia. Sin embargo ocupar un conjunto con mas similitudes no hubiera valido la pena, puesto que significaría que el cuidado que se debe de tener a la hora de secar estas prendas sería muy similar.

Con el modelo generado el siguiente paso será comparar este con otros modelos de ML, como pueden ser Redes Neuronales, Arboles de decisión y sus agrupamientos a gran escala. Tenemos especial interés en comparar nuestro modelo con un Bosque de arboles aleatorios, puesto que es un modelo barato de implementar y podría ser una opción mas robusta para modelos de secadoras diseñadas para un mercado mas pequeño y exclusivo.

VIII. IMPLEMENTACIÓN CON FRAME WORKS

Con la intención de encontrar un mejor modelo de ML, vale la pena revisar los modelos de Random Forest y Red Neuronal. Optamos por estas opciones por que son opciones muy efectivas tanto por su fuerza bruta, o por su capacidad de especializarse a problemas de clasificación.

A. Random Forest

El **Random Forest** es un algoritmo de aprendizaje automático que utiliza un conjunto de árboles de decisión para realizar predicciones. Cada árbol en el bosque se entrena con un subconjunto aleatorio de los datos de entrenamiento y realiza una predicción independiente. La predicción final del Random Forest se obtiene promediando (para regresión) o

votando (para clasificación) las predicciones de todos los árboles.[3]

Este método es conocido por su capacidad para mejorar la precisión y reducir el sobreajuste en comparación con un solo árbol de decisión. [4]

B. Neuronal Network

Una **Red Neuronal** es un modelo de aprendizaje automático inspirado en la estructura y funcionamiento de las redes neuronales biológicas en el cerebro humano. Consiste en capas de nodos interconectados, llamados neuronas artificiales, que trabajan juntas para procesar información y aprender patrones a partir de los datos.[5]

Las redes neuronales se utilizan para tareas como la clasificación, la regresión y el reconocimiento de patrones. Cada neurona recibe una entrada, la procesa mediante una función de activación y transmite la salida a las neuronas de la siguiente capa. Este proceso permite a la red aprender y mejorar su rendimiento a medida que se entrena con más datos.[6]

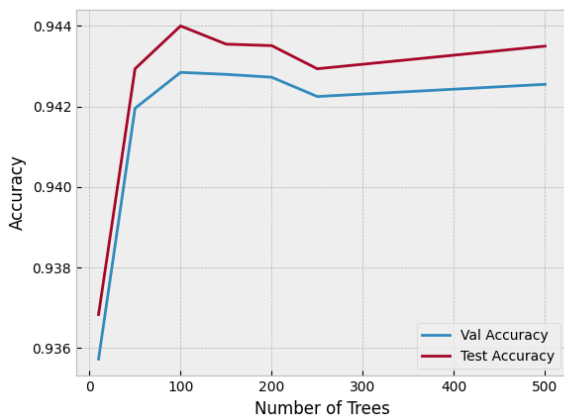
IX. RESULTADOS DE FRAME WORK

Para poder probar los diversos frame works de ML, reutilizaremos nuestro data set ya limpiado y antes de escalar. Esto para ahorrar tiempo, pero vale la pena documentar que se pudo limpiar un Data Set mediante Frame Works.

A. Random Forest

El primer modelo que pondremos a prueba será el **Random Forest**, principalmente. Este siendo el mas barato de los dos nos es de especial interés, así la integración en una secadora no representara un incremento masivo en el precio.

Para poder encontrar el mejor tamaño de bosque decidimos probar varias configuraciones, hasta encontrar la que tenga mejor 'Accuracy'.



Hiperparámetro	Valor	Descripción
Numero de arboles	100	Numero de arboles implementados en nuestro bosque.
Profundidad máxima de cada árbol	8	Limita la profundidad, ayuda a evitar el Overfitting.

Muestras requeridas para dividir un nodo	2	Previene el overfitting y ayuda al árbol a crecer mas.
--	---	--

Considerando que las semillas de los árboles son aleatorias nada nos garantiza que el volver a correr el código nos genere los mismos árboles, pero estos hiperparámetros deberían de generar arboles similares.

La tendencia es clara, mientras más arboles mejora el rendimiento de nuestro árbol, aunque esto puede hacer nuestro algoritmo más caro, en este caso no hay mucha mejoría después de los 100 árboles por bosque.

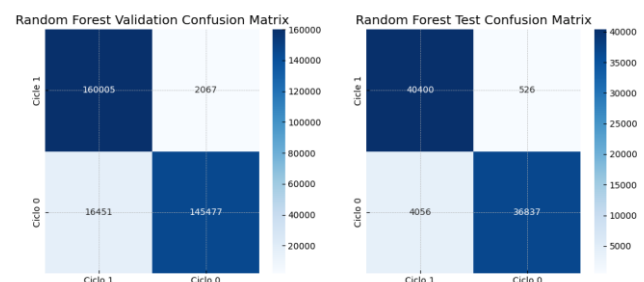
Otro punto para remarcar es que a la hora de armar el árbol debemos poner especial atención en que las deltas de las curvas y al 'Accuracy' que nos arroja entre 'Validation' y 'Test'. Esto nos puede dar un buen indicador de que tan bueno es nuestro modelo en una situación cercana a la realidad y nos puede demostrar si hay **OverFitting** en caso de que el delta entre las curvas sea alto o **UnderFitting** en caso de que el bias es alto y nuestro 'Accuracy' sea bajo.

En este caso no hay un overfitting por que la curva de Test esta sobre la curva de validación, y al tener un alto 'Accuracy' podemos inferir que el bias es bajo.

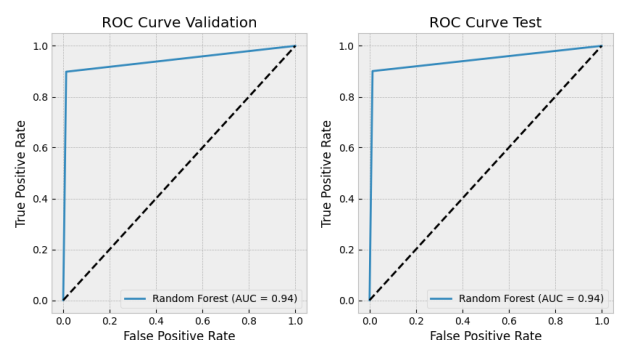
Hacemos uso de **sklearn** y de los siguientes parámetros para nuestro modelo.

Training Random Forest with 100 trees	
Validation Accuracy	0.94
Test Accuracy	0.94

Esto nos indica que el modelo es en extremo muy acertado, muestra una mejoría comparada con nuestro modelo anterior, la regresión logística. Sin embargo, vale la pena revisar las demás métricas que obtuvimos para conocer más acerca del desempeño del modelo.



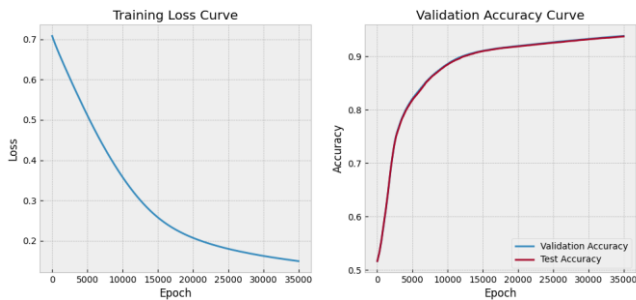
Revisando nuestras matrices de confusión es claro que el 'Random Forest' presenta un mejor desempeño de la regresión logística que implementamos anteriormente.



Viendo las curvas ROC con un área de 0.94 es algo bueno, significa que nuestro clasificador puede discriminar de una forma muy acertada. Aunque la forma nos indica que nuestro modelo tiende a errar mas cuando identifica el ciclo 0 de lo que erra identificando el ciclo 1, que si bien los identifica en su mayoría de forma correcta seria buena idea a la hora de implementarlo en un caso real validar las predicciones del ciclo 0.

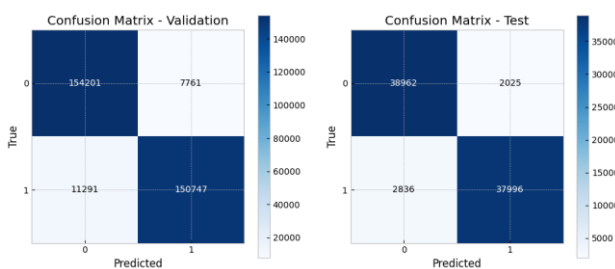
B. Red Neuronal

Ahora que hemos probado el desempeño de los bosques vale la pena ocupar una implementación más robusta y especializada en clasificar.

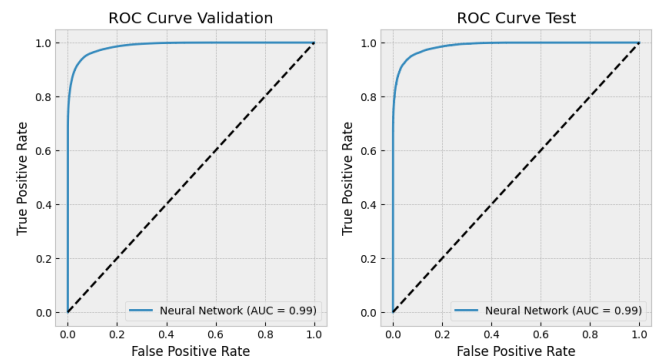


Haciendo uso de una red neuronal podemos ver como la perdida se desploma a comparación de nuestro modelo de regresión logística, esto es de esperarse considerando el poder que tiene una red neuronal. Por otro lado la grafica de 'Accuracy' Podemos ver que el modelo esta perfectamente entrenado para reconocer los valores dentro del conjunto de 'Test' y los de 'Validation' teniendo muy poca diferencia entre ambos.

Training Neuronal Network with 1 hidden layer of 50 neurons	
Validation Accuracy	0.94
Test Accuracy	0.94



Si revisamos la matriz de confusión nos daremos cuenta que el modelo es muy bueno comparado con el modelo de regresión logística. Si comparamos este modelo contra el 'Random Forest' que revisamos anteriormente, pareciera que erramos mas con este, aunque en realidad podríamos decir que este discrimina mejor entre las clases, puesto que no parece que las matrices estén cargadas hacia alguna predicción como pasaba con el modelo anterior.



Viendo la curva ROC nos damos cuenta que el modelo tiene mejor capacidad de discriminación que el modelo anterior, la curva esta menos afilada, lo que significa que el modelo tiene la misma capacidad para predecir cualquiera de los dos ciclos.

X. CONCLUSIÓN

En los análisis realizados hasta este punto, queda claro que la Red Neuronal ha mostrado un rendimiento superior en comparación con los otros modelos evaluados, específicamente en términos de precisión (accuracy) y área bajo la curva (AUC). Este resultado destaca la capacidad de las redes neuronales para manejar la complejidad de los datos y su habilidad para capturar patrones no lineales en las variables. Sin embargo, es importante reconocer que el desempeño actual de la Red Neuronal no implica que los demás modelos, como la regresión logística o los bosques aleatorios, no puedan alcanzar o incluso superar estos resultados si se afinan adecuadamente sus hiperparámetros o se optimiza el preprocesamiento de los datos. Los modelos tradicionales a menudo pueden beneficiarse de un ajuste más meticuloso en su configuración y, en ciertos casos, lograr rendimientos competitivos.

A pesar de la ventaja demostrada por la Red Neuronal en este contexto específico, no se debe concluir de inmediato que este modelo sea la mejor opción para resolver la problemática de la "secadora ideal" que estamos buscando. La selección del modelo de Machine Learning más adecuado no debe basarse únicamente en las métricas de rendimiento actuales, sino también en factores como la interpretabilidad, el costo de implementación, los recursos computacionales necesarios y las necesidades del producto o servicio en cuestión. En muchos casos, un modelo más simple puede ser suficiente y preferible si ofrece ventajas en términos de facilidad de implementación y mantenimiento, sin sacrificar significativamente la precisión.

En este caso particular, propongo una estrategia diferenciada para las secadoras de distintas gamas. Para las secadoras de gama baja, la regresión logística puede ser una opción adecuada, ya que es un modelo interpretable, con bajos costos computacionales y relativamente sencillo de implementar. Su simplicidad lo convierte en una opción robusta para productos en los que la eficiencia y el costo de producción son consideraciones clave. Por otro lado, para las secadoras de gama alta, los bosques aleatorios podrían ser la opción ideal. Aunque son más complejos que la regresión logística, ofrecen una excelente capacidad de generalización, siguen siendo más interpretables que las redes neuronales y requieren un costo de implementación moderado en comparación con las redes neuronales profundas. Además, su capacidad para manejar datos heterogéneos y capturar

interacciones complejas entre las variables podría ser una ventaja en este segmento.

Por último, si bien es cierto que la Red Neuronal podría generar una solución mucho más sofisticada, capaz de discriminar entre todas las clases de manera más precisa y adaptarse mejor a problemas de alta complejidad, también debemos considerar que este tipo de modelo conlleva un mayor costo de desarrollo, requiere más recursos computacionales y puede ser excesivamente complejo para la fase actual del proyecto. En este momento, no parece necesario implementar una solución tan avanzada y costosa, especialmente si los modelos más simples pueden cumplir con los requisitos del problema. A medida que se desarrolle y evolucione la tecnología, quizás más adelante pueda justificarse el uso de redes neuronales, pero por ahora, un enfoque más sencillo parece ser el camino más sensato y eficiente.

REFERENCIAS

- [1] **Fondo Monetario Internacional (FMI)**. (2024, 14 de enero). *La economía mundial transformada por la inteligencia artificial ha de beneficiar a la humanidad. La economía mundial transformada por la inteligencia artificial ha de beneficiar a la humanidad (imf.org)*
- [2] IBM. (s.f.). ¿Qué es la regresión logística? Recuperado de [¿Qué es la regresión logística? | IBM](#)
- [3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- [4] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, 2(3), 18-22. <https://CRAN.R-project.org/doc/Rnews/>
- [5] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. <https://www.deeplearningbook.org/>
- [6] Haykin, S. (2009). *Neural Networks and Learning Machines* (3rd ed.). Prentice Hall. <https://doi.org/10.5555/1530190>