Pronóstico de inventario, Aplicando Aprendizaje por Refuerzo en una Empresa Comercializadora de Productos Veterinarios.

Diego Marcel Fernandez Álvarez, Kevin Gutierrez Paredes, Moises Meza Rodriguez, Stephanny Gabriela Sanchez Bautista ,Jovani Eleuterio Quispe Quispe
Universidad Nacional de Ingeniería
Lima, Perú
diego.fernandez.a@uni.pe
kevin.gutierrez.p@uni.pe
Moises.meza.r@uni.pe
stephanny.sanchez.b@uni.pe
jovani.quispe.q@uni.pe

***Abstract*— El proyecto de aprendizaje por refuerzo tiene como objetivo abordar el pronóstico de inventario, considerando el doble desafío de minimizar el stock en inventario y maximizar las ganancias en un entorno de empresa Comercializadora de productos Veterinarios utilizando técnicas de aprendizaje por refuerzo. Mediante simulación, se desarrolló y probó un conjunto de agentes inteligentes capaces de tomar decisiones en tiempo real para optimizar la gestión de inventarios. Nuestros hallazgos demuestran reducciones significativas en la gestión de Inventarios y aumentos sustanciales en las ganancias, destacando el potencial del aprendizaje reforzado para mejorar la eficiencia operativa y la sostenibilidad dentro del sector Comercial.**

I. INTRODUCCIÓN

La empresa Comercial del Sector Veterinario, actualmente tiene deficiencia en su proceso de Compras, para el abastecimiento correcto de la demanda de ventas para atender los pedidos de los clientes, para lo cual se enfrentan a una compleja paradoja. Por un lado, su enfoque en la satisfacción del cliente, la amplia variedad de productos y la disponibilidad constante en los almacenes, genera un exceso de inventario. Por otro lado, este exceso se traduce en un desperdicio significativo de Líneas de productos con fecha de vencimiento, con implicaciones Financieras para la empresa alarmantes.

Anualmente, más del 10% de los productos con fecha de vencimiento se pierden, lo que representa una pérdida para la empresa en la línea de Vacunas.

En este contexto, la empresa tiene un problema que necesita implementar una propuesta de mejora para afrontar esta pérdida por controles deficientes en línea de producto con fecha de vencimiento. Pequeños cambios en sus procesos pueden tener un impacto significativo. Sin embargo, la necesidad de maximizar ganancias en un entorno competitivo añade una capa de complejidad a la problemática.

II. MARCO TEÓRICO

Agentes de diferencia temporal (TD) : Los agentes de diferencia temporal (TD) son algoritmos de aprendizaje por refuerzo que combinan lo mejor de la programación dinámica y los métodos de Monte Carlo. Estos agentes aprenden a tomar decisiones óptimas en un entorno al actualizar sus estimaciones de valor en cada paso, basándose en la diferencia entre la recompensa obtenida y la estimación previa. A diferencia de los métodos de Monte Carlo que esperan completar una secuencia completa de acciones, los agentes TD pueden aprender de forma más eficiente al realizar actualizaciones incrementales. Esta capacidad de aprender de forma continua y adaptarse a nuevos entornos los convierte en una herramienta valiosa en diversas aplicaciones, desde juegos hasta robótica.

Aprendizaje fuera de políticas utilizando Q-learning: el agente de Q-learning selecciona acciones en función del valor Q máximo estimado para cada par estado-acción y actualiza sus valores Q utilizando el error de diferencia temporal entre las recompensas observadas y predichas. El agente equilibra la exploración y la explotación a través de una política épsilon-codiciosa, donde elige acciones aleatorias con una probabilidad épsilon y, en caso contrario, selecciona la acción con el valor Q más alto.

Aprendizaje según la política utilizando Expected Sarsa: Expected Sarsa es un algoritmo de aprendizaje de diferencias temporales según la política que estima los valores de acción utilizando el valor esperado del siguiente par estado-acción bajo la política actual. Este enfoque permite al agente aprender directamente de sus acciones mientras considera las recompensas futuras esperadas, lo que conduce a un aprendizaje más estable.

Agente Deep Q-Network (DQN): el agente DQN emplea una red neuronal profunda, específicamente una arquitectura completamente conectada, para aproximar la función de valor Q. Al aprovechar las redes neuronales, el agente es capaz de aprender patrones complejos y tomar decisiones informadas basadas en los estados observados. A través de la capacitación, el agente de DQN adapta su función de política para optimizar las decisiones de reabastecimiento de inventario, con el objetivo en última instancia de maximizar las ganancias y minimizar el desperdicio y las ventas perdidas. El agente pudo ajustar con éxito la decisión de reabastecimiento ajustando la función de política con una red neuronal completamente conectada.

Agente Base: un agente base se utiliza como punto de referencia para comparar el rendimiento de diferentes políticas y otros agentes.

III. SITUACIÓN PROBLEMÁTICA

La empresa Comercial del Sector Veterinario se encuentra en una situación problemática compleja debido a la deficiencia en su proceso de compras, la cual genera un exceso de inventario y, a su vez, un desperdicio significativo de productos con fecha de vencimiento, especialmente en la línea de vacunas.

La paradoja radica en:

- **Necesidad de satisfacer al cliente:** La empresa busca mantener una amplia variedad de productos y asegurar su disponibilidad constante para cumplir con la demanda y garantizar la satisfacción del cliente.
- **Exceso de inventario:** Esta estrategia, sin embargo, provoca un sobreabastecimiento y acumulación de productos en los almacenes.
- **Pérdida financiera por productos vencidos:** El exceso de inventario, junto con la falta de controles eficientes, lleva a que un porcentaje considerable de productos (más del 10% de las vacunas) caduquen, generando pérdidas financieras alarmantes.

IV. METODOLOGÍA

Este proyecto desarrolló un modelo para la gestión de inventarios en el sector minorista, aprovechando el aprendizaje por refuerzo para tomar decisiones óptimas de reabastecimiento que satisfacen la demanda del cliente y minimizan los costos operativos

Se diseñó un entorno de simulación que replica las dinámicas de inventario de una empresa comercializadora de productos veterinarios. Este entorno incluyó parámetros como capacidad de almacenamiento, costos de mantenimiento, precios de compra y venta, tiempos de vencimiento de productos y una demanda modelada mediante una distribución de Poisson con parámetro λ. La simulación se centró en un producto con vida útil limitada, representativo de la línea de vacunas.

Para el desarrollo del modelo, se implementó un marco de trabajo que se muestra en la Fig 1. Este marco de trabajo empieza con la **definición de parámetros**, donde se establecen las características del entorno, como capacidad de inventario, costos operativos, demanda y horizonte temporal. Luego, se realiza la **inicialización del entorno**, que simula dinámicas reales de inventario, incluyendo el seguimiento de productos, costos de desperdicio y cálculo de recompensas en función de las decisiones del agente. Finalmente, en la etapa de **evaluación de políticas**, se prueban y comparan estrategias de reabastecimiento, analizando métricas como ganancias, desperdicio y satisfacción de la demanda, con el objetivo de identificar la política más eficiente y equilibrada para optimizar el sistema.

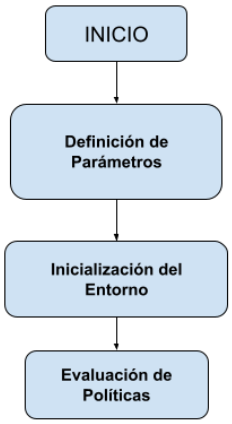


Fig 1.- Diagrama de trabajo para implementar un sistema de aprendizaje por refuerzo aplicado a la gestión de inventarios.

4.1 DEFINICIÓN DE PARÁMETROS.-

State Space : en nuestro proyecto, el espacio de estados consta de tuplas que representan la cantidad de productos en stock y el tiempo más cercano que queda hasta el vencimiento. El tamaño de este espacio estatal está determinado por la capacidad de la empresa y el tiempo de caducidad de los productos. Como sabemos, el tiempo de caducidad puede variar mucho entre productos: hay productos como las vacunas, que tienen una vida útil increíblemente corta en comparación con los productos de otras líneas de venta que pueden durar años sin abrir. En el presente trabajo, la simulación se realiza para un producto, donde en cada paso de tiempo hay una demanda regida por una distribución de Poisson con parámetro λ.

Action Space: las acciones en nuestro proyecto son la cantidad para reabastecer en cada espacio de tiempo, dependiendo de la configuración del entorno, varía de cero a capacidad máxima de stock.

Rewards: en cada paso del tiempo, la recompensa recibida por el agente tiene cuatro componentes:

Utility $(SP - C) * d$	***Waste Penalty*** $WC * UW$
Maintenance Cost $MC * US$	***Unavailability Penalty*** $SP * DNC$ $DNC = max(AS + Action - d, 0)$

Fig 2.- Fórmulas que componen la función de recompensa del modelo de aprendizaje por refuerzo aplicado a la gestión de inventarios.

Para facilitar la comprensión de las fórmulas, en la tabla 1 se muestran las definiciones de las variables y sus símbolos de identificación. Por otro lado, la fórmula para hallar la recompensa se detalla en la ecuación Eq1.

Tabla 1.- Definición de Variables.

Variable	Descripción	Símbolo
Demand	Demanda distribuida de Poisson	d
Utility	Representa las ganancias generadas por la venta de productos	U
Waste Penalty	Calcula el costo asociado a los productos que no fueron vendidos y expiraron.	WP
Unavailability Penalty	Representa las pérdidas económicas por no satisfacer la demanda.	UP
Actual Stock	Stock Actual	AS
Selling Price	precio de venta unitario	SP
Cost	Costo unitario del producto	C
Waste Cost	Costo por unidad desperdiciada	WC
Units Wasted	Número de unidades desperdiciadas	UW
Maintenance Cost	Costo de almacenamiento por unidad	MC
Units in Stock	Unidades en Stock	US
Demand Not Covered	Demanda insatisfecha	DNC
Total Reward	Recompensa total obtenida por el agente en cada paso de tiempo.	R

Los parámetros configurables son: SP, C, WC, MC y λ parámetros de la distribución de la demanda.

Total reward = U- WP - MC - UP ……(Eq1)

4.2 INICIALIZACIÓN DEL ENTORNO

Procedimientos de paso de tiempo:

- **Stock**: si la acción tomada es reabastecer "n" unidades de un producto, la cantidad de stock real aumenta y el seguimiento del tiempo de vencimiento de estas nuevas unidades se registra en una lista de diccionarios (expiration_data) del entorno.

- **Unidades desperdiciadas**: cada elemento de datos de vencimiento se actualiza disminuyendo el tiempo de vencimiento en uno. Las UW (unidades desperdiciadas) en cada paso de tiempo son aquellas cuyo tiempo de vencimiento es menor que cero.

- **Demanda no cubierta**: en cada paso del tiempo, si la demanda no está cubierta por el stock real y la acción (cantidad de reabastecimiento), se aplica una penalización.

4.3 EVALUACIÓN DE POLÍTICAS

La simulación se llevó a cabo en Google Colab, utilizando una GPU Tesla T4 para garantizar un procesamiento eficiente. Los agentes de aprendizaje por refuerzo fueron evaluados a lo largo de 1000 episodios, cada uno con un horizonte temporal de 100 pasos. Durante este proceso, se analizaron métricas clave como la recompensa acumulada, el desperdicio total y el porcentaje de demanda satisfecha. Estas métricas permiten evaluar el desempeño de las políticas implementadas en diversos escenarios.

El principal objetivo de los agentes fue maximizar las ganancias totales, considerando no solo las ventas realizadas, sino también los costos asociados al desperdicio de productos y a las demandas no cubiertas. Este enfoque busca un equilibrio entre la eficiencia operativa y la sostenibilidad financiera del sistema.

La política del **agente base** fue diseñada como una referencia para comparar el desempeño de otras políticas. Su estrategia consistió en monitorear el nivel de inventario y el tiempo de vencimiento de los productos. Si la cantidad disponible caía por debajo de la mitad de la capacidad máxima del inventario o si el tiempo de vencimiento estaba próximo, el agente reabastecía hasta alcanzar la mitad de la capacidad máxima. En caso contrario, optaba por no realizar ningún reabastecimiento para evitar acumulaciones innecesarias que pudieran conducir a un desperdicio elevado.

En la **evaluación de políticas**, se compararon las decisiones y resultados del agente base con las políticas aprendidas por los agentes de aprendizaje por refuerzo. Este análisis permitió identificar fortalezas y debilidades en cada enfoque, destacando cómo las políticas aprendidas lograban un mejor equilibrio entre la satisfacción de la demanda y la minimización de costos operativos. La comparación entre políticas incluyó visualizaciones de recompensas acumuladas y mapas de calor que reflejaron la eficiencia de las decisiones en diferentes combinaciones de estados.

V. CONFIGURACIÓN DEL PROYECTO

Entorno del proyecto:

- **Entorno de simulación:** configurar espacios de estados, acciones y recompensas incluyendo la demanda regida por una distribución de Poisson y los parámetros como costos de mantenimiento, almacenaje, vencimiento de productos.

- **Implementación**
  - **Espacio de estados**
    - Stock disponible (AS) y tiempo hasta el vencimiento.
    - Parámetros como demanda (λ) y capacidad máxima de inventario.
  - **Espacio de acciones:** Cantidad a abastecer.
  - **Parámetros configurables:** establecer valores iniciales para SP (precio de venta), C (costo unitario), WC (costo de desperdicio), MC (costo de almacenamiento) y λ(Demanda).

Implementación del modelo:

El modelo cuenta con algoritmos como Q-Learning, Expected Sarsa y Deep Q-Network (DQN). Cada uno tiene métodos para actualizar políticas y tomar decisiones basadas en el estado del inventario.

- **Implementación:**
  1. **Q-Learning:** Implementar una tabla Q en el cual se actualizarán los valores para cada combinación de estado-acción. Se usará una política greedy para explorar y explotar.
  2. **Expected Sara:** Calcular los valores esperados de acción basados en política actual. Utilizar la información para ajustar las decisiones.
  3. **DQN:** Configurar una red neuronal completamente conectada para aproximar la función Q. Este algoritmo entrega la red utilizando datos simulados, ajustando las políticas con el tiempo.

Preparación de los Datos:

Los datos claves del proyecto son; demanda histórica, costos y tasas de desperdicios.

- **Implementación:**
  - **Recolección de Datos:** Extraer del sistema los datos de venta, inventarios y productos vencidos de la empresa.
  - **Estructuración:** Crear formatos tabulares con columnas como fecha, stock, demanda diaria, productos vencidos, costos, y otros.
  - **Normalización:** Ajusta los valores para que se encuentren dentro de los rangos para ser utilizados por los modelos de aprendizaje.

Infraestructura tecnológica:

El proyecto usó Google Colab con GPU T4 para entrenar modelos.

- **Hardware**

  Configurar el entorno de google colab u otro entorno de GPU compatible.

- **Software**

  Se utilizan bibliotecas como PyTorch o TensorFlow. Asimismo se instalan paquetes adicionales para simulaciones, como gym o numpy.

Pruebas y simulaciones:

Las simulaciones se realizaron para un solo producto con demanda regida por Poisson.

- **Implementación:**
  - Configurar un entorno de simulación que imite el comportamiento real de la veterinaria.
  - Realizar pruebas con el producto específico (vacuna con vida útil corta).
  - Analizar resultados en términos de: Reducción de productos vencidos, incremento de ganancias y capacidad de cubrir demandas.

Capacitación del personal

Es importante la compresión de las métricas como “reward” y decisiones basadas en políticas es clave.

- **Implementación**:
  - Diseño de manuales o talleres para explicar cómo interpretar las decisiones del modelo.
  - Entrenar al equipo en herramientas como dashboard para monitorear el desempeño del inventario.

Monitoreo y mejora continua

El uso de agentes como Q-Learning o DQN sugiere un proceso iterativo de aprendizaje y ajuste.

- **Implementación:**
  - **Recopilar datos reales:** alimentar el modelo con datos nuevos de inventario y demanda.
  - **Actualizar los parámetros:** ajustar las configuraciones iniciales basándose en el rendimiento observado.
  - **Monitorea métricas clave:** revisar los costos por productos desperdiciados, venta no cubierta y ganancia total.

Las características clave del ambiente incluyen	
Capacity: Stock Maximo que puede tener la empresa.	**Buying Price**: Precio al que se compran los productos .
Selling Price: Precio al que se venden los productos.	**Weekly Demand**: Demanda esperada de productos por semana.
Expiration Time: Vida útil de los productos en días.	**Max Time**: Horizonte temporal máximo para la simulación. .
Gamma: Factor de descuento para recompensas futuras.	**Expiration Cost**: Costo asociado con productos vencidos.
Max Loss: Pérdida máxima aceptable para la empresa	**Maintenance Cost**: Costo asociado con el mantenimiento del inventario. .

VI. DESARROLLO

Para el desarrollo del proyecto, se trabajó en el entorno de google Colab.

La presente investigación detalla los resultados del proyecto de aprendizaje por refuerzo alcanzados:

A. Aprendizaje por Refuerzo

Recompensas:

- Se logra un incremento en las recompensas acumuladas conforme a las simulaciones realizadas. En consecuencia, se refleja un aprendizaje progresivo y adaptable al entorno en el que se encuentra.
- Los algoritmos Epsilon - Greedy, demostraron mejoras continuas en los promedios finales.

Políticas aprendidas:

- Por un lado, se evidenció que durante estados de bajo inventario o tiempos de caducidad altos los agentes toman decisiones agresivas para el abastecimiento.
- Por otro lado, los agentes toman estrategias más simples cuando se mantiene un alto inventario con tiempos de caducidad cercanos.

Desperdicios:

- Las simulaciones iniciales evidencian que los desperdicios eran constantes mientras que el modelo iba avanzando esto se iba reduciendo.
- Se evidencio que durante las simulaciones los

Agentes como Expected Sarsa mantenían niveles de desperdicio bajos.

-

B. Q-learning vs Expected Sarsa

Q-learning:

- Toma decisiones más arriesgadas pero con un enfoque más optimista durante las simulaciones.
- Las simulaciones demostraron que con mayor frecuencia se sobreabestica lo que provocó mayor desperdicio; sin embargo, logró un alto índice de recompensas.

Expected Sarsa:

- Las simulaciones evidenciaron un comportamiento más consevador; en consecuencia, priorizo la estabilidad y la reducción de desperdicio.
- Demostró mayor eficiencia en entornos más restrictivos.

C. Agente Greedy y Epsilon Greedy

Agente Greedy:

- Se evidenció niveles bajos de eficiencia en escenarios complejos y restrictivos ocasionando mayores niveles de desperdicio.
- Las limitaciones de los escenarios demostró su incapacidad para tomar mejores decisiones a pesar de su competitividad inicial.

Agente Epsilon- Greedy:

- En diferentes escenarios logró un desempeño óptimo mediante el cambio de la demanda y los tiempos de caducidad.
- Este agente resaltó su adaptabilidad mediante la exploración y explotación; en consecuencia, logró mejores recompensas en el promedio final.

-

D. Agente Deep Q-Network (DQN)

Recompensas:

- Demostró un incremento significativo durante las recompensas acumuladas. En este caso, se evidenció una mejor adaptabilidad en entornos más simples donde; por ejemplo, el agente Greedy fallo.

Estabilidad:

- La cantidad de episodios de entrenamiento fueron importantes para que el agente DQN, lograra estabilidad en entornos complejos.

Desempeño:

- El equilibrio de reabastecimiento y costos asociados permitió reducir el desperdicio.
- Se desempeñó mejor que otros agentes basados en metodología tabular en escenarios con alta variabilidad.

Limitaciones:

- El tiempo de entrenamiento es mayor debido a su complejidad y la optimización de la red neuronal.
- Es sensible a la configuración de los parámetros.

F. Impacto de los parámetros

- Los tiempos cortos en caducidad aumentaron la complejidad del problema durante las simulaciones. En este caso, el agente Expected Sarsa fue el más eficiente en este escenario.
- Los parámetros del factor descuento y la tasa de aprendizaje (“gamma y step size”) afectaron el rendimiento de los agentes.

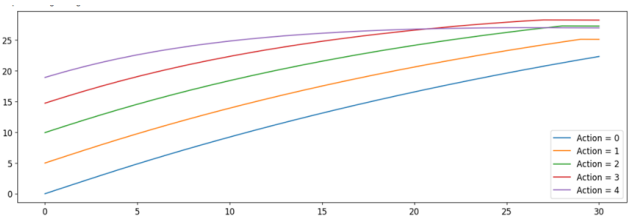
VII. RESULTADOS

el proyecto se desarrolló para evaluar el resultado de 6 tipos de agentes, con una misma situación problemática, para lo cual después de haber realizado la codificación, pruebas y entrenamiento del algoritmo, se obtuvo los siguiente resultado con la siguiente configuración de los parámetros:

						Considerando Costo de Mantenimiento de Inventario		Sin Costo de Mantenimiento de Inventario				
		Basic	Greedy	Eps-Gredy	Fixed Agent	Q-Learning	Expected Sarsa	Q-Learning	Expected Sarsa	Q-Learning	Expected Sarsa	Lambda Sarsa
	num_runs		200	200	10	10	10	30	30	10	10	10
	num_steps		1000	1000								
	num_episodes				100	1000	1000	100	100	300	300	300
Environmentals	capacity	30			30	30	30	30	30	30	30	30
	maintenance_cost	0.01			0.01	0.01	0.01	0	0	0.01	0.01	0.01
	expiration_time	30			10	10	10	10	10	10	10	10
	buying_price	0.5			0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	selling_price	1			1	1	1	1	1	1	1	1
	expiration_cost	1			1	1	1	0	0	1	1	1
	weekly_demand	5			5	5	5	5	5	5	5	5
	gamma	0.925			0.925	0.925	0.925	0.925	0.925	0.925	0.925	0.925
	max_time	100			365	365	365	100	100	365	365	365
	initial_stock				30	30	30	30	30	30	30	30
	max_loss				200	200	200	100	100	1000	1000	1000
Agent Info	num_actions	31		31	31	31	31	31	31	31	31	31
	num_states	930		930	341	341	341	341	341	341	341	341
	expiration	30		30	10	10	10	10	10	10	10	10
	epsilon	0.1		0.25	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	step_size	0.5		0.5	0.8	0.8	0.8	0.5	0.5	0.8	0.8	0.8
	discount	1		1	1	1	1	1	1	1	1	1

resultados por cada tipo de agente y escenario.

6.1 - Básico

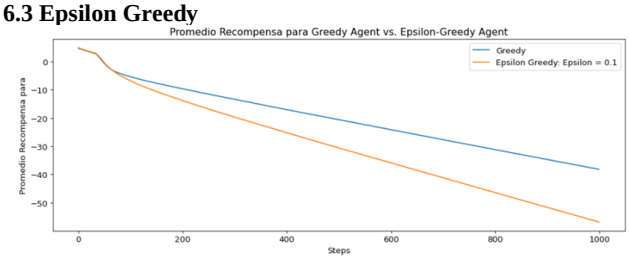


este gráfico nos ayuda a entender cómo el número de productos comprados y el tiempo de expiración restante afectan las ganancias esperadas. La mejor política dependerá de las condiciones específicas del problema, como la demanda, los costos y el tiempo de expiración del producto.

6.2 Greedy

El gráfico indica que, si bien la estrategia voraz puede parecer prometedora al principio, no es una estrategia óptima para maximizar la recompensa a largo plazo en este entorno. Se necesitaría un agente con una estrategia más sofisticada (como la exploración) para lograr un mejor rendimiento.

6.3 Epsilon Greedy



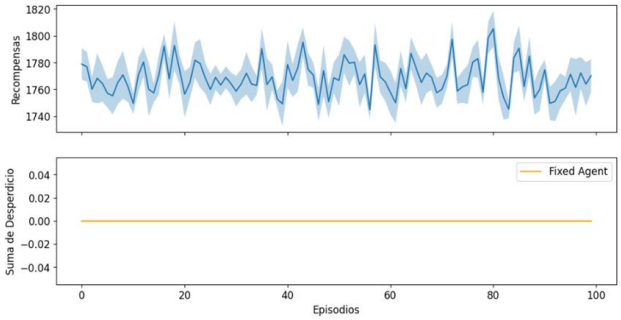
El gráfico muestra la comparación del rendimiento, medido en promedio de recompensa, entre dos agentes (Greedy y Epsilon Greedy) a lo largo de un número determinado de pasos en un entorno de aprendizaje por refuerzo.

Análisis de las curvas:

- **Greedy:** Este agente sigue una estrategia codiciosa, es decir, siempre elige la acción que cree que le dará la mayor recompensa inmediata. Inicialmente, puede obtener mejores resultados, pero su rendimiento se estanca e incluso disminuye a largo plazo. Esto se debe a que no explora otras acciones que podrían conducir a recompensas mayores en el futuro.
- **Epsilon Greedy:** Este agente equilibra la explotación (elegir la mejor acción conocida) con la exploración (probar acciones aleatorias). El parámetro "epsilon" controla la probabilidad de exploración. En este caso, con un epsilon de 0.1, el agente explora el 10% de las veces y explota el 90%. Aunque su rendimiento inicial puede ser menor, a largo plazo logra un mejor rendimiento promedio que el agente Greedy. Esto se debe a que la exploración

En este escenario particular, el agente Epsilon Greedy con un epsilon de 0.1 demuestra un mejor rendimiento a largo plazo que el agente Greedy. Esto resalta la importancia de equilibrar la explotación y la exploración en el aprendizaje por refuerzo para lograr un rendimiento óptimo.

6.4 Fixed Agent



El gráfico muestra el rendimiento del "Fixed Agent" en la tarea de gestión de inventario a lo largo de los episodios de entrenamiento.

Gráfico superior (Recompensas):

- **Tendencia general:** La línea azul representa la recompensa promedio obtenida por el agente en cada episodio. A pesar de algunas fluctuaciones, no se observa una clara tendencia ascendente o descendente, lo que sugiere que el agente no está aprendiendo a mejorar su rendimiento significativamente a lo largo del tiempo. Esto es esperable dado que se trata de un agente con una política fija.
- **Variabilidad:** El área sombreada alrededor de la línea azul representa el intervalo de confianza de las recompensas. La amplitud de esta área indica la variabilidad en el rendimiento del agente entre diferentes ejecuciones. Una alta variabilidad sugiere que el rendimiento del agente es sensible a las condiciones iniciales o a la aleatoriedad del entorno.

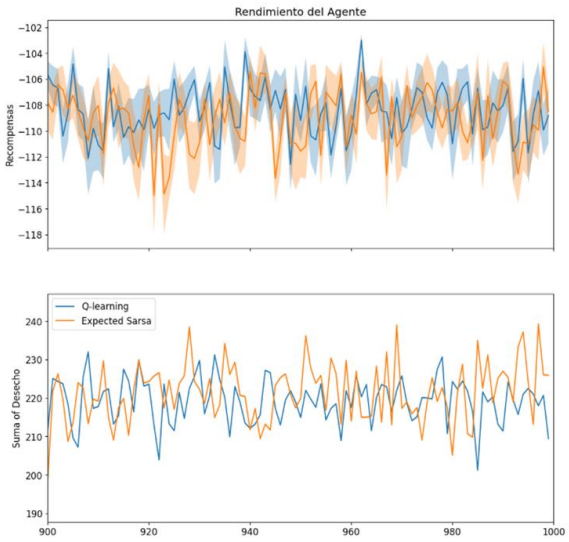
Gráfico inferior (Suma de Desperdicio):

Nivel de desperdicio: La línea naranja representa la cantidad promedio de productos desperdiciados por el agente en cada episodio. En este caso, la línea se mantiene prácticamente constante y cercana a cero, lo que indica que el agente es muy eficiente en la gestión del inventario y evita el desperdicio de productos.

En general, el gráfico sugiere que el "Fixed Agent" tiene un rendimiento consistente en términos de recompensas, aunque no muestra una mejora a lo largo del tiempo. Además, es muy eficiente en la prevención de desperdicios.

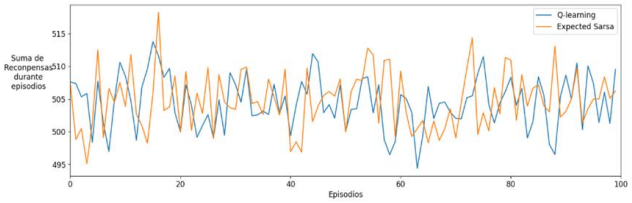
Es importante tener en cuenta que este análisis se basa en un solo agente con una política fija. Para obtener una imagen más completa del problema de gestión de inventario, sería necesario comparar el rendimiento de diferentes agentes con diferentes políticas, incluyendo agentes que puedan aprender y adaptarse al entorno.

6.5 QLearning – Sarsa



Estos gráficos sugieren una compensación entre maximizar las recompensas y minimizar el desperdicio. El agente que logra mayores recompensas (azul) parece hacerlo a costa de generar más desperdicio. Esto podría deberse a una estrategia más agresiva o exploratoria que conduce a más productos caducados. El otro agente (naranja) es más conservador, lo que resulta en menores recompensas pero también menos desperdicio.

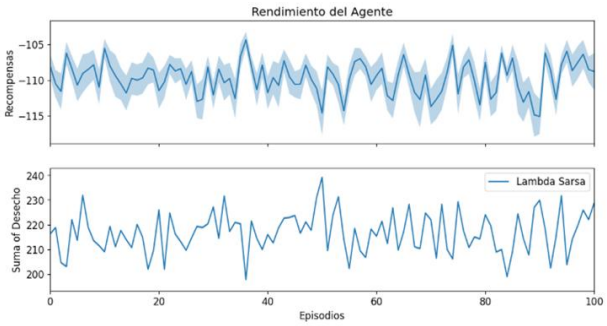
6.6 Q-learning - Expected Sarsa



Este gráfico muestra la suma de recompensas obtenidas por dos algoritmos de aprendizaje por refuerzo, Q-learning y Expected Sarsa, a lo largo de los episodios de entrenamiento.

En general, este gráfico indica que ambos algoritmos, Q-learning y Expected Sarsa, logran aprender y obtener recompensas en el problema de gestión de inventario. Aunque Q-learning parece tener un rendimiento ligeramente superior en este caso particular, la diferencia no es muy grande. Es importante recordar que este es solo un experimento y que los resultados podrían variar si se cambian los parámetros o el entorno.

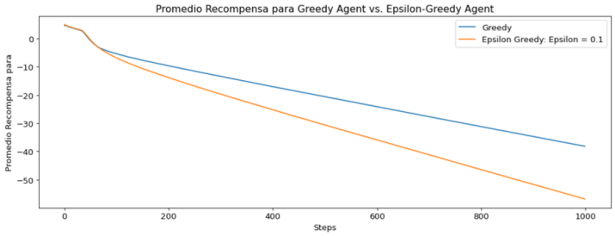
6.7 Lambda Sarsa



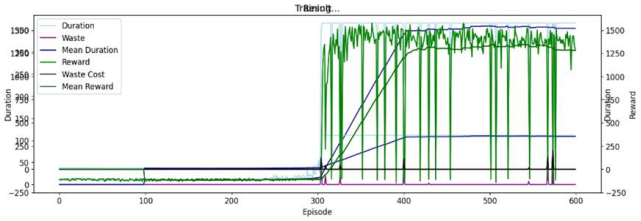
El gráfico sugiere que el agente de aprendizaje por refuerzo está aprendiendo con éxito a mejorar su rendimiento en la tarea, aumentando las recompensas y disminuyendo el desperdicio. Sin embargo, el rendimiento no mejora indefinidamente y parece estabilizarse, lo que podría deberse a limitaciones del agente, del entorno o de la configuración del experimento.

VIII. CONCLUSIONES

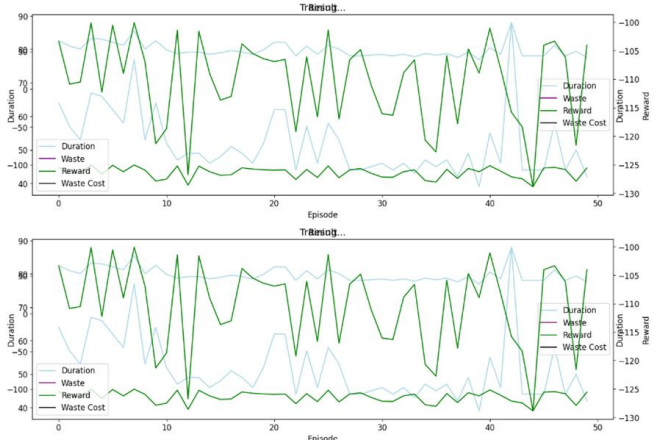
Las conclusiones del presente trabajo, indican que el uso de los agentes Greedy , Eps Greedy, el resultado no fue el óptimo esperado a la situación problemática planteada, donde el rendimiento de ambos agentes se ve afectado por la complejidad del entorno y configuración de los parámetros del agente(los factores de descuento) y la selección del valor de epsilon, en el agente Epsilon Greedy es crucial. Un valor muy alto puede llevar a una exploración excesiva y un rendimiento deficiente, mientras que un valor muy bajo puede resultar en una explotación excesiva y una falta de adaptación al entorno



a diferencia de los agentes de tipo Q-Learning , Expected Sarsa y Lambda Sarsa, el cual tiene resultado tangible tanto en el proceso de aprendizaje



En conclusión, con los resultados obtenidos por cada ambiente y agente,s se tiene las siguientes conclusiones:



Este gráfico parece mostrar el rendimiento de un agente de aprendizaje por refuerzo a lo largo de 50 episodios. Hay varias métricas trazadas, lo que nos da una visión completa de cómo el agente está aprendiendo y qué tan bien se está desempeñando.

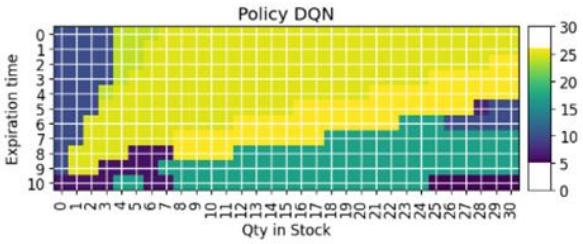
Métricas:

- **Duración**: Representa cuánto dura cada episodio. Un aumento en la duración puede indicar que el agente está explorando más o que las tareas se están volviendo más complejas.
- **Desperdicio**: Probablemente se refiere a la cantidad de recursos desperdiciados o utilizados eficientemente por el agente. Una disminución en el desperdicio indica un aprendizaje y una optimización de recursos.
- **Recompensa**: Es la señal principal que utiliza el agente para aprender. Un aumento en la recompensa significa que el agente está mejorando su rendimiento en la tarea.
- **Costo de desperdicio:** Podría ser una penalización asociada con el desperdicio, que afecta la recompensa general.

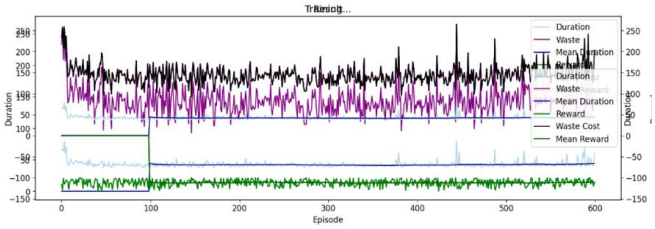
Análisis:

- **Tendencia general:** A lo largo de los episodios, parece haber una tendencia general a la disminución del desperdicio y un aumento en la recompensa. Esto sugiere que el agente está aprendiendo a realizar la tarea de manera más eficiente y a lograr mejores resultados.
- **Variabilidad:** Hay fluctuaciones en todas las métricas. Esto es normal en el aprendizaje por refuerzo, ya que el agente explora diferentes acciones y estrategias. Las fluctuaciones pueden disminuir a medida que el agente converge hacia una política óptima.
- **Relación entre métricas:** Parece haber una correlación entre el desperdicio y la recompensa. Cuando el desperdicio disminuye, la recompensa tiende a aumentar. Esto tiene sentido, ya que un menor desperdicio generalmente implica un mejor uso de los recursos y, por lo tanto, un mayor rendimiento.
- **Duración:** La duración de los episodios parece variar sin un patrón claro. Esto puede depender de la naturaleza de la tarea y de cómo el agente está explorando el entorno.

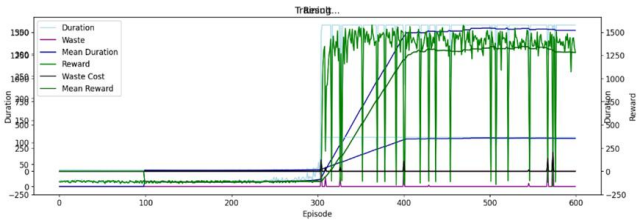
Por lo que se puede concluir que ambos gráficos, parece que siguen patrones similares en cuanto a la disminución del desperdicio y el aumento de la recompensa. Sin embargo, existen algunas diferencias en la magnitud de los cambios y en la variabilidad de las métricas. Esto puede deberse a diferentes condiciones iniciales, parámetros del agente o variaciones aleatorias en el proceso de aprendizaje.



El gráfico proporciona una visualización de la política aprendida por un agente DQN, lo que ayuda a comprender cómo el agente toma decisiones en diferentes situaciones. Sin embargo, se necesita más contexto para realizar un análisis más profundo de la política y su efectividad.



El gráfico muestra un agente de aprendizaje por refuerzo que parece estar aprendiendo y mejorando su rendimiento en la tarea a lo largo de 600 episodios. Sin embargo, un análisis más preciso requeriría más información sobre el problema y la configuración del experimento



El gráfico muestra que el agente está aprendiendo con éxito. Está logrando reducir el desperdicio, aumentar la recompensa y, en general, mejorar su rendimiento en la tarea.

IX. REFERENCIAS

- [1] Nina Deliu et al., "Artificial Intelligence-based Decision support Systems for precision and Digital Health,", arXiv:2407.16062v1, 2024
- [2] I. Franco et al., "Sales time series analytics using deep Q-Learning", arXiv:2201.02058v1, 2022.
- [3] Xinlin Wang et al., "A reinforcement learning-based online learning strategy for real-time short-term load forecasting", https://doi.org/10.1016/j.energy.2024.132344, 2024.

- [4] H. van Hasselt et al., "Deep Reinforcement Learning with Double Q-learning", arXiv:1509.06461v3, 2015.
- [5] D. Arumugam et al., "Satisficing Exploration for Deep Reinforcement Learning", arXiv:2407.12185v1, 2024.
- [6] D. Abel et al., "Three Dogmas of Reinforcement Learning", arXiv:2407.10583v1, 2024.

X. CONTRIBUCION DE CADA INTEGRANTE

Nombre	Contribución
Diego Fernandez Álvarez	Código y Pruebas
Kevin Gutierrez Paredes	Resultados
Moises Meza Rodriguez	Introducción, situación problemática, metodología
Stephanny Sanchez Bautista	Comparativo de modelos
Jovani Quispe Quispe	Conclusiones

XI. Anexos

- Código del Proyecto Colab:

https://colab.research.google.com/drive/1C2K-0hUU7D6EbJL-K0MxwOwEIgZ-GdgI?authuser=1#scrollTo=2RIt9s-WEq0r

- GitHub

https://github.com/DFAUNIIA/TrabajoFinal_RL_Grupo3