

Agente Pymes

Santiago Santacruz, Duvan Andres Ramirez

Ingeniería de Sistemas, Universidad EAFIT

Medellín, Colombia

ssantacruz@eafit.edu.co

daramirez@eafit.edu.co

Video : <https://bit.ly/3nOtjYo>

Resumen— Las pymes son una gran parte de la economía colombiana. Estas están en todo el país y poco a poco se van generando más microempresas las cuales son una ayuda para las familias colombianas, no obstante con pocas alternativas y mal manejo de presupuesto además de problemas en el inventario estas cierran al corto periodo de abrirlas generando pérdidas. Con base en esto, se considera que se puede crear un agente el cual ayude en las tareas las cuales se puedan automatizar las tareas que ayuden a los usuarios para no tener que cerrar y tener un factor de éxito más grande. Por lo tanto, el objetivo de este ensayo es utilizar la metodología MASINA para la creación de un agente inteligente basado en los modelos definidos y creando así una herramienta que sea capaz de ser usada. Gracias a esto, se pudieron hacer diferentes tareas las cuales se definieron en un diseño del agente y se pudieron obtener diferentes resultados tanto en optimización, incertidumbre, lógica y procesamiento de lenguaje, para que el agente aprendiera y usará. Además de esto se uso una clustering el cual es una tarea adicional la cual el usuario puede ver sus diferentes tipos de categorías gracias al procesamiento del lenguaje.

Keywords: Machine learning, masina, optimización, clustering.

I. INTRODUCCIÓN

El 70% de las microempresas generadas en Colombia no pasan de los cinco años, solo el 29.7% de estas tienen como factor un éxito en el mercado. Estas empresas constituyen un gran porcentaje de personas, las cuales pueden acceder al empleo, gracias a los creadores de estos emprendimientos, y además tener un ingreso fijo. Sin embargo, por los factores de éxito que estas tienen muchas personas quedan sin trabajo en muy poco tiempo [1].

Por otra parte, una de las principales razones por las que las Pymes no sobreviven, es por la falta de sistematización de sus procesos [2]. Esto demuestra la importancia de que las Pymes tengan sistemas que les ayuden con sus procesos desde una etapa muy temprana. Adicionalmente, argumenta que una de las principales causas por las que fracasan las Pymes es por tener una mala gestión de sus inventarios [3].

Con base a esto, la solución que propone nuestro equipo, es poder mejorar el hecho de que los locales tengan una automatización de inventario y ventas, logrando que el porcentaje de éxito de las empresas, que usen nuestra solución, pueda seguir más tiempo en el mercado compitiendo con las grandes empresas.

Esto es gracias a una solución tecnológica la cual usa un agente inteligente, lo cual nos diferencia de los demás productos que hay en el mercado, puesto que se logra que el agente tenga un conocimiento y aprenda de las acciones de los usuarios, también que pueda calcular el óptimo de las utilidades de los productos y además pueda manejar el lenguaje natural haciendo un procesamiento de este.

En este documento, se podrá evidenciar más a detalle la estructura de la solución y la arquitectura que hemos empleado, así como también, la conclusión del producto.

II. MARCO TEÓRICO

El tiempo y la eficiencia son factores claves en una empresa. Para poder sacar el máximo rendimiento a los recursos, es necesario conocer a fondo todos los procesos, pues solo así se podrá identificar qué puntos necesitan mejoras, cuáles funcionan bien y aquellos que hay que eliminar. A continuación, abordamos los conceptos aplicados en la automatización de los procesos de una MiPymes y la aplicación/debate de los mismos.

A. Masina

Al proceso de comprobación que confirma que el software cumple con sus requisitos y aborda las necesidades del usuario se lo conoce como Verificación. Se sabe entonces que la comprobación puede ser un proceso muy complejo, tanto en modo conceptual como en el computacional, debido a la importancia que se tiene para cumplir con los requisitos del proyecto que se esté llevando a cabo. Es por esto que, existen una gran cantidad de metodologías para desarrollar sistemas multiagentes, pero muy pocas de ellas tienen componentes que permitan la verificación de sus diseños.

Por otro lado, la Metodología MASINA (MultiAgent Systems in Automation) surge a fin de adecuar la metodología al modelado de procesos de automatización e integrar la noción inteligente, entre otras cosas. Los agentes diseñados con MASINA deben ser verificados de acuerdo a ciertos criterios de evaluación, para garantizar que el SMA funcione apropiadamente. MASINA ofrece un conjunto de modelos que agrupan los elementos básicos y necesarios para describir al SMA.

Los modelos están basados en la identificación de atributos y sus relaciones; el producto generado consiste en una serie de plantillas que describen al SMA. Básicamente, está constituida por cinco modelos, los cuales describen un SMA: Modelo de agente, Modelo de tareas, Modelo de inteligencia, Modelo de Coordinación y Modelo de comunicación [4].

B. Machine Learning y Clustering

Machine Learning o también conocida como ML es el proceso en el cual los computadores pueden aprender mediante un proceso de uso de datos, el cual hace que aprenda a través del tiempo. En otras palabras, es el proceso en el cual un programa puede mejorar su rendimiento mediante la experiencia. Machine learning hace uso de unos datos los cuales se le conoce como los datos de muestra, y con estos se le hace pasar al programa lo que llamamos anteriormente “experiencia”, con la finalidad de que este pueda hacer predicciones y tomar decisiones sobre la problemática

que se esté intentando resolver.

Las ventajas que se tiene de esta tecnología van desde que es mucho más acertada, ya que el aprendizaje que se tiene con los datos entrena al agente, entonces entre más datos se puedan recolectar el nivel de comprensión y de toma de decisiones es más preciso [5]. El modelo tomado va haciendo validaciones y descartando datos según lo aprendido, lo que hace que aprenda patrones automáticamente mientras que va procesando los datos de manera más rápida a medida que va aprendiendo. Otra gran ventaja de ML es la adaptabilidad que se tiene para poder tratar diferentes tipos de problemas, el agentes que se tienen pueden ser configurados y adaptados según las características y métricas que se elijan, por esto a medida que se va escalando los datos a ser procesados, el modelo no tiene problema en tomar decisiones por lo aprendido anteriormente.

Para poder “enseñarle” al computador o programa esta experiencia existen varias metodologías, entre ellas están aprendizaje supervisado, no supervisado y semisupervisado[6]. El aprendizaje supervisado es muy utilizado y se caracteriza por entrenar al programa o computadora dándole datasets marcados al igual que con algunas respuestas correctas, para que este aprenda a clasificar de manera correcta los datos que se le están entregando, en algunos casos también se utiliza con la intención de que este prediga resultados de manera acertada, algunos ejemplos del uso de este aprendizaje sería la clasificación de imágenes, la detección de spam y análisis predictivo[7]. Aprendizaje no supervisado, método que estamos utilizando en este proyecto, se caracteriza por utilizar algoritmos que analizan y agrupan datos que no están marcados, lo que es impresionante de este aprendizaje es que pareciera que la computadora o programa en serio está aprendiendo solo, ya que en el proceso de agrupar los datos esté va descubriendo patrones o grupos de datos sin la intervención de un humano, su habilidad para encontrar diferencias y similitudes.

Como se mencionó anteriormente el aprendizaje no supervisado se encarga de dividir los datos de entrada en subgrupos, haciendo que los datos que están en cada grupo sean similares entre sí de alguna manera u otra manera. A este proceso se le llama Clustering.

El método de Clustering no siempre es necesario utilizarlo, debido a que no en todos los casos van a dar los mejores resultados y la idea de utilizar este método es crear el mayor impacto posible con los datos que se tienen. La mayoría de casos este es utilizado cuando los datos que se tienen son demasiado extensos y en realidad no tienen ningún orden como tal, esto es debido al trabajo que te quita de encima, como con otras herramientas no supervisadas este coge todos los datos y de manera rápida te los organiza en algo más moldeable y práctico de usar[8]. Un buen Clustering ayuda demasiado al entendimiento de los datos y al pensar cómo trabajar con ellos[9].

III. Diseño del sistema.

En este punto ya vamos a empezar a mirar la construcción del agente MiPymes para poder resolver la problemática ya expresada, para esto nos apoyamos del método masina, como se puede ver en la figura número 1, hacemos un diagrama de conceptualización el cual nos ayuda a revisar y diseñar cómo será este agente y cuales objetivos, tareas, el tipo de agente tendremos que implementar. Con base en esto se obtuvo lo siguiente.

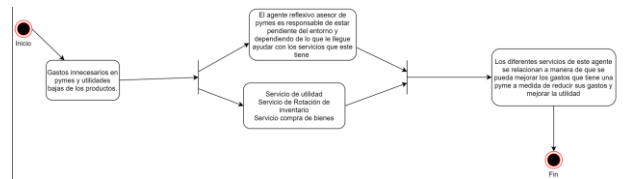


Figura 1. Diagrama de conceptualización

A. Definición del agente

Como primer paso definimos que tipo de agente tendremos y cuáles serán sus características.

- Tipo de agente : El agente es de tipo reactivo simple.
- Objetivo: Aumentar la utilidad neta de la pyme.
- Medida de rendimiento: Reducción de costos.
- Ambiente : Personas.
- Actuadores : Sistema de preguntas.
- Sensores : Teclado, Pantalla.

B. Objetivo del agente

El objetivo del agente asesor de pymes busca aumentar la rentabilidad a partir de estrategias comerciales que permitan una mayor venta de productos. Esto lo hace disponiendo de diferentes servicios que ofrece.

C. Servicios

Dentro de este agente se encontrarán disponibles diferentes servicios para la mejora de utilidad, siendo estos los siguientes:

- Servicio Utilidad: Este se encargará de hacer comparación entre la utilidad que tenga el producto.
- Servicio de Informe de inventario: Se encarga de revisar los productos entrantes y salientes para poder gestionar la venta de estos.
- Servicio de compra de bienes: Informar cuando haya inventario bajo de producto. El agente va a ser de tipo reflexivo porque va a analizar el ambiente

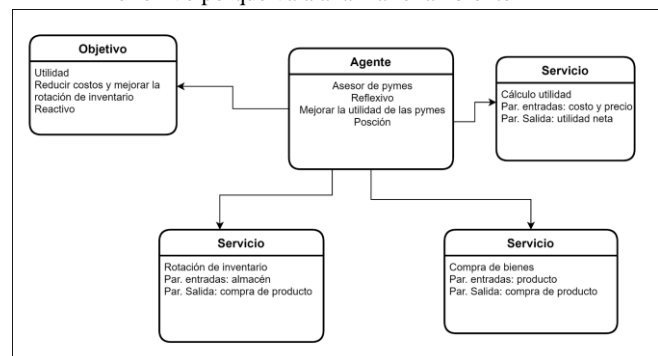


Figura 2. Diagrama de Agentes con sus servicios

En la anterior imagen se puede evidenciar según los modelos de masina como este agente va a tener unos servicios orientados con su construcción y cual es el objetivo específico para este. Dándonos a entender el contexto de este agente y lo que que se quiere realizar en este informe, con esto en cuenta ahora vamos a

revisar las tareas que se tendrán para poder tener estos servicios.

D. Tareas

Estos servicios implementarán las siguientes posibles tareas, las cuales ayudarán con el objetivo general que es ayudar a las pymes con el problema planteado.

- Venta de producto
- Compra de inventario
- Cálculo de utilidad
- Alerta de inventario

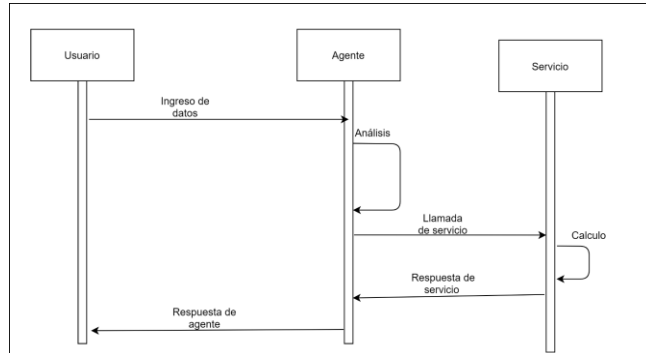


Figura 3. Diagrama de secuencia de servicio

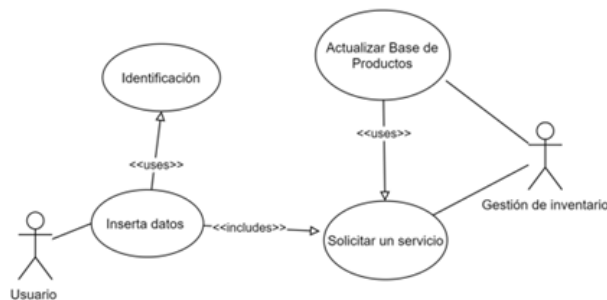


Figura 4. Diagrama de casos de uso

En la figura 3 podemos ver un diagrama el cual nos da como el agente responderá a las reacciones del usuario con él y en general la llamada a los servicios que este tiene. Mientras que en la figura 4 podemos notar los diferentes un caso de uso que el agente puede tener y como este se relaciona con el usuario en cuestión. Teniendo en cuenta esto, ahora revisaremos punto por punto el diseño de cada una de las tareas que se hicieron para este agente junto con sus modelos.

a. Optimización Lineal

Siguiendo con lo planteado, se planteó el primer problema que se llevará a cabo para su resolución, teniendo en cuenta el cálculo de la utilidad total y usando algoritmos de optimización. Para esto se investigó acerca de diferentes algoritmos y se llegó a uno el cual nos da según unos parámetros la utilidad total neta de los diferentes productos en su combinación.

El problema a resolver será calcular la máxima utilidad que se pueda obtener de un listado de artículos que tiene un precio de venta y un costo. Con esto, se tendrán unas restricciones para ser semejante a la realidad, se tiene un presupuesto para la compra de artículos y un presupuesto para guardarlos en la tienda. Además de esto se requerirá que los artículos tengan una ganancia marginal mayor al porcentaje. Con base a esto, el diseño del algoritmo se puede notar en la figura 5 con base en el modelo Masina como el modelo de tarea para este caso.

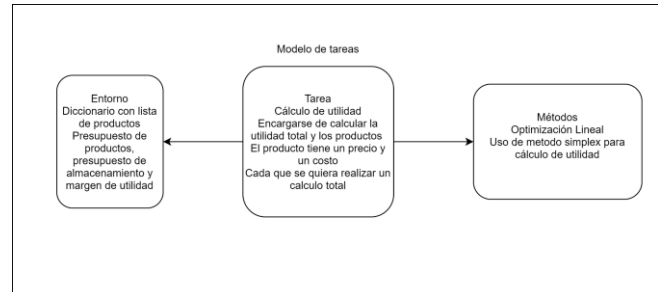


Figura 5. Modelo de tareas

Usando técnicas de optimización matemática para resolver problemas de programación lineal, se busca maximizar la utilidad de la función $f(x)$. Hay diferentes soluciones que se pueden hallar en el sistema pero esta nos permite encontrar la solución máxima que se pueda con las restricciones que se tengan. El sistema de ecuaciones planteado para resolver se puede ver en la figura 7 como un ejemplo de cómo será nuestro sistema de ecuaciones. Función a optimizar:

$$\text{Max}Z = u_1X_1 + u_2X_2 + \dots + u_nX_n$$

Restricciones:

$$c_1X_1 + c_2X_2 + \dots + c_nX_n \leq \text{presupuesto}$$

$$cA_1X_1 + cA_2X_2 + \dots + cA_nX_n \leq \text{presupuesto de almacenamiento}$$

$$1X_1 + 0X_2 + \dots + cAX_n \geq \text{vender mínimo ((el presupuesto)*(1/ \# de productos))}$$

$$0X_1 + 1X_2 + \dots + cAX_n \geq \text{vender mínimo ((el presupuesto)*(1/ \# de productos))}$$

$$\vdots$$

$$\vdots$$

$$0X_1 + 0X_2 + \dots + 1AX_n \geq \text{vender mínimo ((el presupuesto)*(1/ \# de productos))}$$

Figura 6. Optimización máxima utilidad.

La función a optimizar Z busca maximizar la utilidad teniendo en cuenta la utilidad de cada producto. La primera restricción indica que el costo de los productos debe ser menor que el presupuesto. La segunda restricción indica que el costo de almacenamiento de los productos debe ser menor que el presupuesto de almacenamiento. Por último, las demás restricciones indican que la cantidad comprada de cada producto debe ser como mínimo un porcentaje del presupuesto, con el propósito de tener una variedad de productos esto se puede ver en la figura 6.

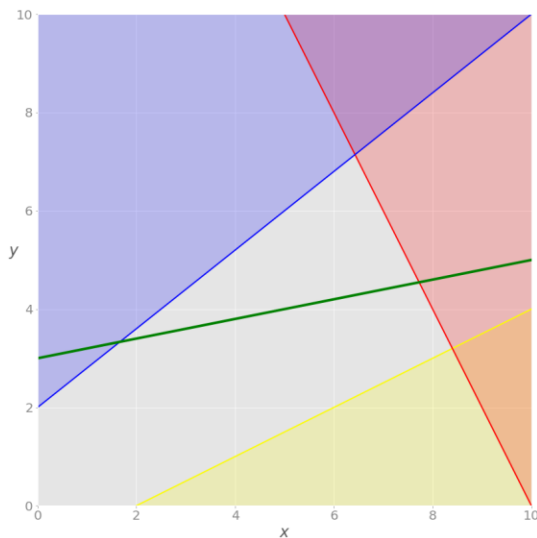


Figura 7. Ejemplo gráfico del método para dos variables

b. Implementación lógica.

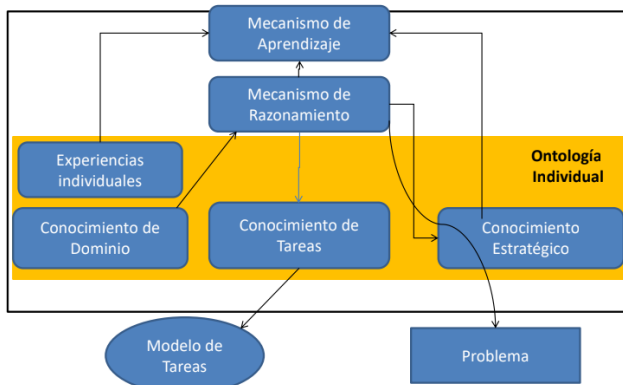


Figura 8. Modelo de inteligencia

La tarea que buscamos solucionar es la de rotación de inventario. Permitir al Agente tener el conocimiento de cuando hace falta un producto, cuando es necesario enviar una alerta por falta de un producto, entre otras funciones. Al tener en cuenta las bases que tenemos de modelamiento como se puede notar en la figura 8, vamos a implementar la lógica en python utilizando pytholog (una librería que nos permite usar prolog), el objetivo de utilizar esta biblioteca es definir un razonamiento simbólico con el uso de lógica imperativa para generar el aprendizaje automático. Usaremos una base de conocimiento, con productos, vendedores y ventas. Además definimos axiomas para realizar las búsquedas que nos permitan tomar decisiones de valor en nuestra MiPymes.

La estructura que realizamos con la librería de pytholog que usa el lenguaje de Prolog es la siguiente:

hechos:

- hay(x,y) -> x siendo el producto, y siendo la cantidad de este.
- vende(x,y,z) -> x siendo el vendedor, y el producto que vende y z la cantidad vendida de este.
- recompensa(x,y) -> x es un vendedor y Y la cantidad total

de productos vendidos.

- vendedor(x) -> x es un vendedor.

reglas:

- vendio(X,Y,Z):- vende(X,Y,Z), vendedor(X).
- hayProducto(X):- hay(X, Y), Y > 0.
- faltaProducto(X):- hay(X, Y), Y < 1.
- alertaProducto(X):- hay(X, Y), Y < 4.
- premiarVendedor(X):- recompensa(X,Y), vendedor(X), Y >= 15.
- castigarVendedor(X):- recompensa(X,Y), vendedor(X), Y <= 1.

En nuestra base de conocimiento los hechos representan las afirmaciones de nuestro conocimiento que tenemos por dado, cuales vendedores tenemos, que productos hay, que ventas se han realizado, cuántos productos ha vendido cada vendedor. Las reglas son las formulaciones lógicas que usamos para llegar a generar conocimiento en nuestro Agente. Cuales productos hacen falta, que productos hay, teniendo en cuenta esto se puede resumir como será el mecanismo de aprendizaje en la tabla 1 explicandonos más sobre como es el tipo de aprendizaje que se llevará a cabo y resumiendo lo que se habló anteriormente.

Mecanismo de aprendizaje	
Nombre	Pylog
Tipo	Supervisado
Tecnica de representacion	Hechos y reglas
Fuente de aprendizaje	La información dada se usa para tener una base de conocimiento el cual es para aprender y probar.

Tabla 1. Mecanismo de aprendizaje

La asignación de los hechos es realizada de manera dinámica en nuestra base de conocimiento, cada nueva afirmación es agregada para ser tomada en cuenta y se actualiza los hechos dependiendo de la afirmación. Si por ejemplo tengo 40 unidades de cheetos y tomas vende 5, entonces habrá 35 unidades de cheetos, se agrega la venta de tomas y se actualiza su recompensa. Esto se puede hacer gracias al mecanismo de razonamiento el cual se puede evidenciar en la tabla 2.

Mecanismo de Razonamiento	
Fuente de informacion	Datos simulados para experimentar
Fuente de alimentación	Predecir los productos, el número de ventas y los productos en alerta que tiene como base.

Técnica de inferencia	Técnica basada en hechos
Lenguaje de representación de conocimiento	OWL
Relación tarea-inferencia (resultado esperado)	Relación de predicción
Estrategias de razonamiento	Deductivo

Tabla 2. Mecanismo de Razonamiento

c. Incertidumbre

La tarea que buscamos solucionar es la mejora de la incertidumbre haciendo uso de la tarea de rotación de inventario. Al permitir que el Agente tenga el conocimiento de su base de datos, se puede calcular cuál es la probabilidad que tiene un producto X de ser vendido dado que este en alerta, así se puede tener en cuenta cuando se deben de restablecer el inventario y no comprar productos de más o que estos se acaben haciendo que haya pérdidas de ingresos. Con base en esto, La teoría de la probabilidad condicional nos permite establecer un modelo para solucionar este problema, esta establece que: dado un B verdadero, ¿cuál es la probabilidad de que A también sea cierto? Esto se puede ver en la figura 8 como está ilustrado lo anterior explicado con probabilidad condicional. Además de esto, el agente se le añade un nuevo modelo de tarea describiendo lo que este hará como se ve en la figura 9.

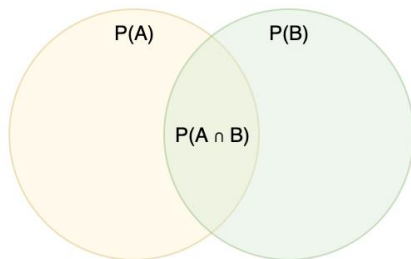


Figura 8. Conjunto de probabilidades

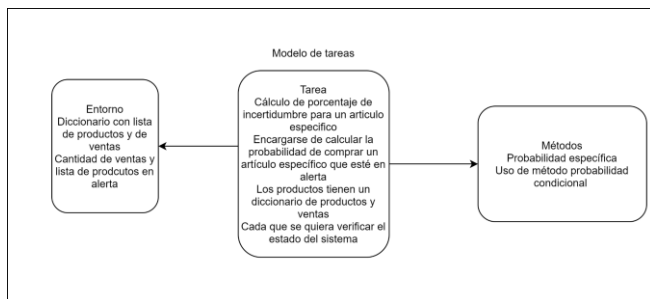


Figura 9. Modelo de tareas probabilidad condicional

la fórmula de probabilidad condicional es $P(A|B) = P(A \cap B) / P(B)$.

partes:

$P(A|B)$ = probabilidad de que ocurra A, dado que B ocurre

$P(A \cap B)$ = probabilidad de que ocurran A y B

$P(B)$ = probabilidad de que ocurra B

Dividimos $P(A \cap B)$ entre $P(B)$ porque queremos excluir la probabilidad de casos que no sean B. Estamos evaluando nuestra probabilidad para estar dentro B.

Dividir por $P(B)$ elimina la probabilidad de que todo no lo sea B. Sobre $C \rightarrow B$.

Para esta iteración usamos la librería de numpy y pandas para el manejo de nuestros datos. El objetivo de utilizar la biblioteca de pandas es poder identificar mejor las relaciones entre las variables, además de las funcionalidades que esta biblioteca nos otorga. Los datos en nuestro Agente entonces se representan de la siguiente manera:

	ganancia	costo	utilidad	...	cantidad	Costoso	Alerta
papas	1000	600	400	...	17	1	1
cheetos	1200	500	700	...	30	1	1
chicles	1500	800	700	...	50	1	0
gol	1100	500	600	...	35	1	1
ponky	700	300	400	...	15	0	1
salsa de tomate	100	20	80	...	4	0	1

Figura 10. Estructura de datos de productos

	ponky	cheetos	papas	salsa de tomate
alertaproducto	5	4	7	1.0
tomas	4	5	13	0.0
juan	4	5	13	0.0
pedro	4	5	13	0.0
alberto	4	5	13	0.0
tulio	0	0	1	1.0

Figura 11. Estructura de datos de Ventas

La fila alerta producto en la figura 10 representa el número de ventas que se hicieron de un producto estando en alerta y la columna Alerta de la figura 11 representa los productos que actualmente están en Alerta. Además de esto, se formuló con el teorema de bayes la cuestión de qué porcentaje es posible de que me compren un producto que esté en alerta dependiendo si es caro. Esto se hizo teniendo en cuenta la siguiente estructura, con base en esto se hizo un modelo de tareas respondiendo el método, la nueva tarea usando el teorema de bayes el uso de este como se puede ver en la figura 12.

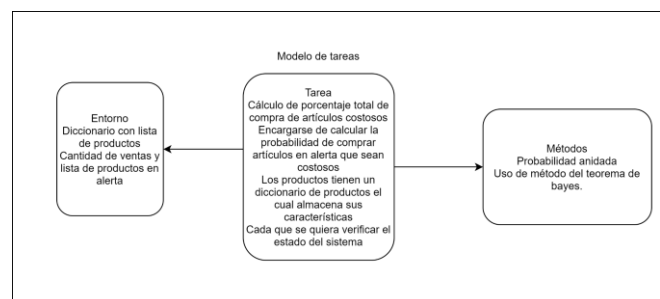


Figura 12. Modelo de tareas probabilidad anidada

d. Aprendizaje del agente

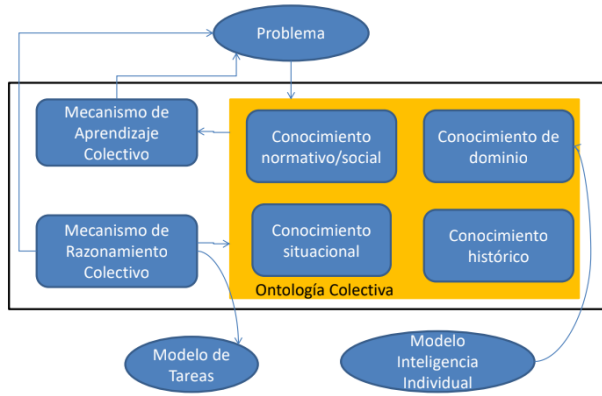


Figura 13. Modelo de inteligencia colectiva.

La tarea que atacamos en esta iteración fue la asociación de patrones para la personalización de productos. Siguiendo con el agente en cuestión, se hizo el modelo de inteligencia colectiva, la cual se puede notar en la figura 13 y además se investigó diferentes tipos de aprendizaje para que este agente conociera. Entre los cuales se estudió el de las tablas de Q y el algoritmo Apriori. Este último nos hizo dar cuenta e impactó de manera en que se puede ver una gran utilidad en el uso de este, la figura 14 nos muestra el pseudocódigo.

```

Apriori(T, ε)
  L1 ← {large 1 – itemsets}
  k ← 2
  while Lk-1 ≠ ∅
    Ck ← {a ∪ {b} | a ∈ Lk-1 ∧ b ∉ a} – {c | {s | s ⊆ c ∧ |s| = k – 1} ⊄ Lk-1}
    for transactions t ∈ T
      Ct ← {c | c ∈ Ck ∧ c ⊆ t}
      for candidates c ∈ Ct
        count[c] ← count[c] + 1
    Lk ← {c | c ∈ Ck ∧ count[c] ≥ ε}
    k ← k + 1
  return ∪k Lk

```

Figura 14. Algoritmo Apriori

Por medio del conocimiento previo que se tenía, se ajustó algunos métodos en las tareas que se hicieron anteriormente haciendo que no solamente se hiciera una transacción por producto, sino que se guardaran que productos iban a venderse en una sola transacción. Esto nos permite encontrar de forma eficiente "conjuntos de ítems frecuentes" los cuales nos ayudan a generar unas reglas de asociación.

Nombre	Algoritmo Apriori
Tipo	Aprendizaje no supervisado
Tecnica de representacion	Reglas de Asociación
Fuente de aprendizaje	Transacciones de la aplicación
Mecanismo de actualización	Refuerzo

Tabla 3. Mecanismo de aprendizaje colectivo

Comercialmente hablando, este algoritmo nos permite ver que tipo de productos son comprados juntos, de esta manera nos permite tenerlo en cuenta para la generación de combos o tener experiencia de usuario el cual podemos poner items cerca para hacer que sean vendidos de manera más frecuente.

e. Procesamiento de lenguaje natural

El objetivo de la tarea ahora es procesar grandes cantidades de productos sin etiquetar para extraer su significado y darles una apropiada categorización. Para esta tarea haremos uso del procesamiento de lenguaje natural (PNL) y K-means, será un método no supervisado. Utilizaremos una base de datos Colombiana de inventario de productos, pues necesitábamos un gran volumen de datos para tener una asertividad alta en la agrupación de las palabras con los clusters. Con base en esto se generó el siguiente modelo de tareas para la representación de la iteración figura 15.

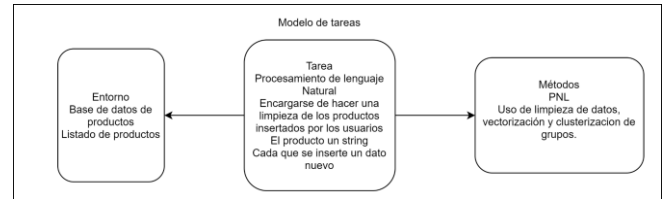


Figura 15. Modelo de tarea PNL

El método que realizaremos consta de los siguientes **pasos**:

- Procesar previamente el texto (los nombres de los alimentos) en palabras limpias para que podamos convertirlo en datos numéricos.
- Vectorización, que es el proceso de convertir palabras en características numéricas para prepararse para el aprendizaje automático.
- Aplicar agrupación de K-means, un algoritmo de aprendizaje automático no supervisado, para agrupar nombres de alimentos con palabras similares.
- Evaluación de la calidad de los conglomerados mediante el etiquetado y la visualización de los conglomerados.

```

0      Limpia tapicería en espuma Binner
1      Limpia tapicería y alfombras Binner
2      Limpiador Dr. Beckmann alfombras 2en1 poder oxi
3      Aceite Bon Aire eléctrico perfume flores
4      Ambientador Aire Varitas Bambú Bonaire 2 Repue...

...
10140      Jabón Carey surtido barra
10141      Jabón Corporal Nosotras Frutos Rojos
10142      Jabón Corporal Nosotras Karite Avena
10143      Jabón Corporal Nosotras Te Verde
10144      Jabón Corporal Nosotras Yogurt

```

Figura 16. Productos en dataframe recién extraídos.

Entonces, primero que todo se hará un preprocesamiento para limpiar los datos de entrada, quitando los caracteres innecesarios, para que los datos sean relevantes en el cluster. En nuestro caso se quitaron palabras de finalización, puntuaciones y números figura 16.


```
[46] matrix[['prod_name', 'limpia', 'ambientador', 'alfombra']]
```

	prod_name	limpia	ambientador	alfombra
0	Limpia tapicería en espuma Binner	1	0	0
1	Limpia tapicería y alfombras Binner	1	0	1
2	Limpiador Dr. Beckmann alfombras 2en1 poder oxi	0	0	1
3	Aceite Bon Aire eléctrico perfume flores	0	0	0
4	Ambientador Aire Varitas Bambú Bonaire 2 Repue...	0	1	0
...
10140	Jabón Carey surtido barra	0	0	0
10141	Jabón Corporal Nosotras Frutos Rojos	0	0	0
10142	Jabón Corporal Nosotras Karite Avena	0	0	0
10143	Jabón Corporal Nosotras Te Verde	0	0	0
10144	Jabón Corporal Nosotras Yogurt	0	0	0

10145 rows x 4 columns

```
0      limpia tapiceria en espuma binner
1      limpia tapiceria y alfombra binner
2      limpiador Dr beckmann alfombra en poder oxi
3      aceit bon air electrico perfum flore
4      ambientador air varita bambu bonair repuesto
...
10140      jabon carey surtido barra
10141      jabon corpor nosotra fruto rojo
10142      jabon corpor nosotra karit avena
10143      jabon corpor nosotra Te verd
10144      jabon corpor nosotra yogurt
Length: 10145, dtype: object
```

Figura 17. Productos tratados según los filtros.

Además de esto, se realizó un procesamiento de las palabras derivadas, esto se refiere a que las palabras parecidas fueran una sola el ejemplo se puede ver en la figura. Un ejemplo de esto sería chocolate, choco, Chocola figura 17. Esto haría que se juntaran en un solo dato y pasarlos a minúsculas por la consistencia de los datos. Después de esto, se removerán los colores para que éstos no afecten la clusterización, ya que si se toma en cuenta los colores puede que genere clasificaciones con más peso basado en esto y no en por producto como tal. Finalmente pasamos de poseer un dataframe a una lista con todos los nombres de los productos filtrados[11].

```
['limpia tapiceria en espuma binner',
'limpia tapiceria y alfombra binner',
'limpiador Dr beckmann alfombra en
poder oxi',...]
```

Después de esta limpieza de datos lo que se hizo fue pasar de texto plano a datos numéricos para poder realizar análisis estadísticos. Para esto se usó la vectorización, hay muchas técnicas para hacerlas las cuales se investigaron e implementaron para saber cual es la mejor con respecto a nuestro código. Entre las que se investigaron esta saco de palabras, TF-IDF (nivel de palabra), TF-IDF (nivel n-gram) y Latent Dirichlet Allocation (LDA). Cada una de estas se implementó en el código para revisar y analizar cómo funcionan y al final tomando en cuenta el criterio de porcentaje de agrupación de palabras consideramos más óptimo y usar bolsa de palabras [7].

Contamos entonces primero la ocurrencia de palabras en un documento. Cada fila, un nombre de alimento, es un documento. El resultado es una matriz que contiene una característica para cada palabra distinta en el texto y el recuento de cada palabra en una fila (o vector) así como sus valores numéricos figura 18.

Para tres columnas obtenemos:

Figura 18. Vectorización bolsa de palabras

Con base en esto, se realizó el método de clusterización. K-means clustering asigna puntos de datos en grupos discretos basados en su similitud o proximidad entre sí. Especificamos el número de conglomerados K y el algoritmo asigna iterativamente cada observación a un conglomerado hasta que las observaciones de cada conglomerado estén lo más cerca posible de su media (o centroide). Para saber y comenzar a conocer cuántos clusters deberíamos tener para reducir procesamiento y tener un buen resultado. Usamos el método de elbow para hacer un test de diferentes valores de K y comparar las distancias de cada dato con el centroide. Como se puede ver en la figura 19.

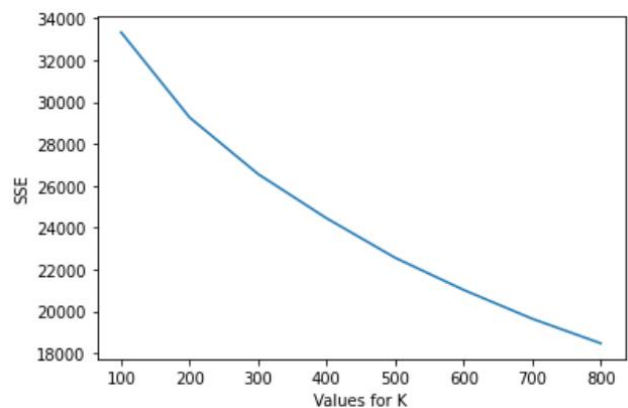


Figura 20. Gráfica de codo para clusters

Con esto ya sabemos que el número óptimo de clusters a realizar es aproximadamente 200-300, figura 21:

	prod_name	cluster
0	Limpia tapicería en espuma Binner	67
1	Limpia tapicería y alfombras Binner	67
2	Limpiador Dr. Beckmann alfombras 2en1 poder oxi	67
3	Aceite Bon Aire eléctrico perfume flores	67
4	Ambientador Aire Varitas Bambú Bonaire 2 Repue...	117
...
10140	Jabón Carey surtido barra	8
10141	Jabón Corporal Nosotras Frutos Rojos	163
10142	Jabón Corporal Nosotras Karite Avena	25
10143	Jabón Corporal Nosotras Te Verde	25
10144	Jabón Corporal Nosotras Yogurt	25

10145 rows x 2 columns

Figura 21. Dataframe con clusters de bolsa de palabras

Nota: Cada fila de nuestra matriz de bolsa de palabras se ha asignado a un grupo entre 0 y 199.

Después realizamos el etiquetado a nuestros grupos con etiquetas legibles. Podemos hacer esto automáticamente recuperando los nombres de las columnas de la matriz que tienen un valor > 0 para cada fila en cada grupo. De esta manera podemos ver las palabras que tienen en común todos los nombres de alimentos en un grupo figura 22.

	prod_name	cluster	label_y
0	Limpia tapicería en espuma Binner	67	
1	Limpia tapicería y alfombras Binner	67	
2	Limpiador Dr. Beckmann alfombras 2en1 poder oxi	67	
3	Aceite Bon Aire eléctrico perfume flores	67	
4	Ambientador Aire Varitas Bambú Bonaire 2 Repue...	117	ambientador
...
10140	Jabón Carey surtido barra	8	barra
10141	Jabón Corporal Nosotras Frutos Rojos	163	
10142	Jabón Corporal Nosotras Karite Avena	25	jabon
10143	Jabón Corporal Nosotras Te Verde	25	jabon
10144	Jabón Corporal Nosotras Yogurt	25	jabon

[10145 rows x 3 columns]

Figura 22. Dataframe con clusters y etiquetas de palabras

Ahora podemos empezar a comprender qué representa cada uno de los grupos. Hay grupos que no tienen palabras comunes en los nombres de sus alimentos, por lo que permanecen sin etiquetar. Una vez con nuestro dataframe con nuestras etiquetas y cluster, podemos realizar el análisis de dimensionamiento del cluster figura 23.

	count of Name
	label_y
	3619
queso	296
vino	289
chocol	188
crema	185
...	...
jamón nico	6
light refresco soka	6
aparato aromatizant car glade repuesto	4
activo brisa disco gel mar mr musculo para sanitario	4
ambientador bolsa canela de en espacio gel glade manzana para pequeno	2

170 rows x 1 columns

Figura 23. Dataframe de dimensionamiento del clúster.

Con $K = 200$ con la matriz de bolsa de palabras, solo 3778 de los 10,145 nombres de alimentos no están etiquetados. En general, el 63% de los nombres de alimentos se agruparon en un grupo limpio. Lo que significa un buen resultado para un primer acercamiento y teniendo en cuenta que redujimos el pool de datos a la mitad por temas de procesamiento. Aun así los nombres de alimentos teóricamente similares deben agruparse porque tienen valores similares para las mismas palabras (y n-gramos)[12].

Posteriormente agregamos una etiqueta adicional de categorización a nuestro dataframe para caracterizar los macro grupos de nuestro cluster figura 25.

	prod_name	...	category
0	Limpia tapicería en espuma Binner	...	Aseo de hogar
1	Limpia tapicería y alfombras Binner	...	Aseo de hogar
2	Limpiador Dr. Beckmann alfombras 2en1 poder oxi	...	Aseo de hogar
3	Aceite Bon Aire eléctrico perfume flores	...	Aseo de hogar
4	Ambientador Aire Varitas Bambú Bonaire 2 Repue...	...	Aseo de hogar
...
10140	Jabón Carey surtido barra	...	Cuidado Personal
10141	Jabón Corporal Nosotras Frutos Rojos	...	Cuidado Personal
10142	Jabón Corporal Nosotras Karite Avena	...	Cuidado Personal
10143	Jabón Corporal Nosotras Te Verde	...	Cuidado Personal
10144	Jabón Corporal Nosotras Yogurt	...	Cuidado Personal

[10145 rows x 3 columns]

Figura 25. Dataframe completo con etiquetas

IV. Experimentos y resultados.

a. Resultados optimización lineal.

Siguiendo con los experimentos para cada una de las tareas planteadas y verificando su funcionamiento en jupyter, ahora realizamos pruebas simulando que el agente estuviera percibiendo el ambiente y realizando las tareas que se formularon en el diseño del sistema. Para esto se inició con la primera función de optimización. Esta función de optimización es utilizada en para que dentro de una pyme pueda tener en cuenta que cantidad de productos los cuales vende pueden generar su máxima ganancia, teniendo en cuenta ámbitos claros como los presupuestos que se tienen. Los resultados que dan el siguiente algoritmo se pueden ver de la siguiente manera :

1. fun: -15791.666666666673
2. x: array([2.77777778, 3.33333333, 2.08333333, 3.33333333, 5.55555556, 83.33333333])

Estos se pueden entonces notar como primero, la máxima utilidad que se tiene de los parámetros que se le pasaron. Por otro lado, se puede ver las cantidades de productos a ser comprados por el gerente de la empresa para poder maximizar sus ganancias. De manera general, el agente utiliza los parámetros que el usuario le pase para poder darle una aproximación más adecuada en cuanto a la ganancia de su utilidad. Esta tipo de información nos ayuda a determinar los tipos de productos que generan mayor utilidad en nuestra MiPymes y nos ayuda a tener un mejor flujo de productos en nuestra rotación de inventario [12].

```

-----
Indique la regla: 5
Calculo de utilidad : Ingrese los siguientes campos

Productos optimos

Presupuesto Al : 2000
Presupuesto :15000
papas: = 4
cheetos: = 5
chicles: = 3
gol: = 5
ponky: = 8
salsa de tomate: = 125

```

Figura 26. Resultados de mercado óptimo.

En la figura 26 se pueden ver los resultados de la optimización, dándonos una respuesta respecto los productos que se tiene en la base de datos junto con los presupuestos dados.

b. Resultados implementación lógica

Siguiendo con el análisis de los experimentos se revisa con unas funciones si la base del conocimiento es funcional, para esto analicemos algunas posibles búsquedas para la siguiente base de conocimiento:

```

hay(papas,17) .
hay(cheetos,30) .
hay(chicles,50) .
hay(gol,35) .
hay(ponky,15) .
hay(salsa de tomate,5) .
vende(tomas,papas,13) .
vende(tomas,cheetos,5) .
vende(tomas,cheetos,5) .
vende(tomas,ponky,4) .
vende(juan,papas,13) .
vende(juan,cheetos,5) .
vende(juan,ponky,4) .
vende(pedro,papas,13) .
vende(pedro,cheetos,5) .
vende(pedro,ponky,4) .
vende(alberto,papas,3) .
vende(alberto,cheetos,5) .
vende(alberto,ponky,4) .
vende(tulio,papas,1) .
recompensa(tomas,37) .

```

```

recompensa(juan,22) .
recompensa(pedro,22) .
recompensa(alberto,12) .
recompensa(tulio,12) .
vendedor(tomas) .
vendedor(juan) .
vendedor(pedro) .
vendedor(alberto) .
vendedor(tulio) .
hayProducto(X):- hay(X, Y), Y > 0.
alertaProducto(X):- hay(X, Y), Y < 4.
faltaProducto(X):- hay(X, Y), Y < 1.
vendio(X,Y,Z):-vende(X,Y,Z),
vendedor(X) .
premiarVendedor(X):- vendedor(X),
recompensa(X,Y), Y >= 15.
castigarVendedor(X):- recompensa(X,Y),
vendedor(X), Y <= 5.

```

?-hayProducto(X)

```

[{'X': 'cheetos'}, {'X': 'chicles'}, {'X': 'gol'}, {'X': 'papas'}, {'X': 'ponky'}, {'X': 'salsadetomate'}]

```

Listamos los productos que tienen unidades mayores a 1. Esta regla nos permite tener un control de inventario y dar una alerta grave sobre los productos faltantes.

?-alertaProducto(X)

```

[{'X': 'No'}, {'X': 'No'}, {'X': 'No'}, {'X': 'No'}, {'X': 'No'}, {'X': 'No'}]

```

Lista los productos y clasifica como Sí o No los productos en los cuales es necesario generar una alerta. La decisión de implementar esta regla es que nos permite tener un control de los productos y poder tomar acciones antes de quedar sin Stock

?-premiarVendedor(X)

```

[{'X': 'alberto'}, {'X': 'juan'}, {'X': 'pedro'}, {'X': 'tomas'}, {'X': 'No'}]

```

?-castigarVendedor(X)

```

{'X': 'No'}, {'X': 'No'}, {'X': 'No'}, {'X': 'No'}, {'X': 'tulio'}]

```

Listamos los vendedores que merecen ser premiados dadas su cantidad de ventas realizada y los vendedores que merecen ser castigados debido a la cantidad de productos vendidos. En este caso todos los vendedores son premiados menos tulio que fue castigado. La decisión de estas reglas es generar un incentivo para que los vendedores generen más ventas y tomar acción sobre aquellos vendedores que no están teniendo un buen rendimiento.

Además de este tipo de consultas también podemos hacer búsquedas específicas como:

```
?-premiarVendedor(tulio)
[ 'No ' ]
```

Esta tarea es de vital importancia para nuestro agente ya que es la encargada de permitir la rotación del inventario y además estar al tanto del registro de todo lo que pasa con nuestros productos, para tomar decisiones de inteligencia.

c. Resultados incertidumbre

Respecto a la incertidumbre, lo primero que tenemos que determinar es la probabilidad que un producto esté en alerta $P(B)$, esto lo hacemos tomando el número de producto en alerta sobre el total de productos. Después determinamos la probabilidad de vender un producto X y que esté en alerta, tomando el número de alertaproducto del producto y dividiéndolo entre la cantidad de alertaproducto de los bienes:

$P(B)$
0.8333333333333334
 $P(A \cap B)$
0.058823529411764705
 $P(A|B)$
7.0588235294117645

Entonces la probabilidad de que un producto X ($X='papas'$ en este caso) sea vendido tal que está en Alerta es del 7% para la figura 27.

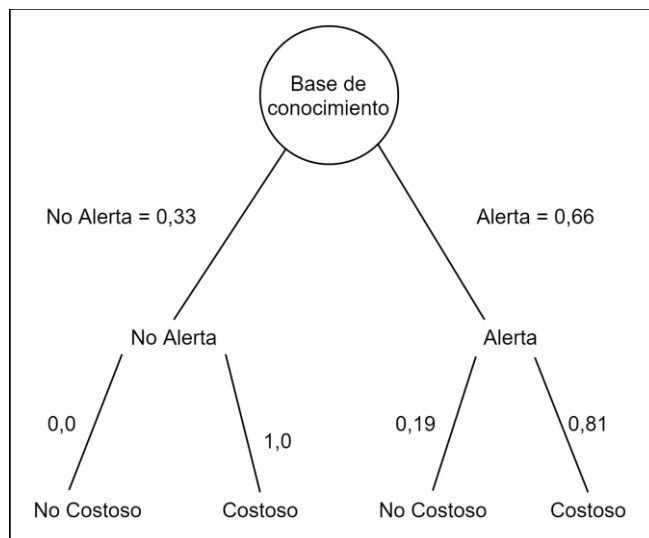


Figura 27. Diagrama Bayes

Con esto en cuenta se pudo calcular en el sistema diferentes tipos de porcentajes anidados, en este caso específico, se quiere calcular el porcentaje de qué probabilidad habría de que

compre un producto en alerta siendo este costoso.

Siendo la formula la siguiente : $P(A|D) = (P(D|A) * P(A))/P(D)$

$P(\text{Alerta}|\text{Costoso})$: Probabilidad de que compren un producto en alerta dado que es costoso.

$P(\text{Costoso}|\text{Alerta})$: Probabilidad de que compren un producto costoso dado que está en alerta.

$P(\text{Costoso})$: Probabilidad de comprar un producto costoso.

d. Resultados aprendizaje del agente

Siguiendo con los experimentos, ahora para la tarea del aprendizaje del agente se siguieron los siguientes pasos para obtener unos resultados. Primero convertimos la lista en un marco de datos con valores booleanos de nuestras ventas. Un True hace significado a una venta y un False a un producto no vendido, como se puede ver en la figura 28.

	Arroz	Cheetos	Papas	Ponky	cheetos	chicle	gol	papas	ponky	salsa de tomate
0	False	True	True	False	False	False	False	False	False	False
1	False	True	True	False	False	False	False	False	False	False
2	True	False	False	False	False	False	False	False	False	False
3	False	False	False	False	True	False	False	True	False	False
4	False	False	False	False	True	False	False	True	False	False

Figura 28. Resultado matriz de venta

Encontramos conjuntos de elementos que ocurren con frecuencia utilizando el algoritmo Apriori. Esto lo logramos gracias a la librería que nos brinda python.

	antecedents	consequents	antecedent support	consequent support	support confidence	lift	leverage	conviction	
0	(Papas)	(Cheetos)	0.2	0.2	0.2	1.0	5.000000	0.16	inf
1	(Cheetos)	(Papas)	0.2	0.2	0.2	1.0	5.000000	0.16	inf
2	(Ponky)	(salsa de tomate)	0.1	0.2	0.1	1.0	5.000000	0.08	inf
3	(ponky)	(cheetos)	0.1	0.3	0.1	1.0	3.333333	0.07	inf

Figura 29. Matriz de datos relacionados.

Este es un ejemplo básico de implementación del aprendizaje de reglas de asociación para el análisis de la venta de productos y dándonos los resultados que aparecen en la figura 29.

e. Resultados visualización y procesamiento de lenguaje natural

Finalmente para la visualización optamos por un mapa de árbol que puede mostrar claramente las etiquetas y tamaños de nuestros grupos dentro de sus respectivas categorías de supermercados. Consideramos usar t-SNE sin embargo, con la gran cantidad de clústeres que tenemos, t-SNE no es el mejor para la legibilidad. Es difícil ver los grupos discretos y su tamaño. Y en 200 grupos, estamos recorriendo mapas de colores varias veces para colorear cada grupo, lo que hace que sea aún más difícil de leer. Es por eso que optamos por un mapa de árbol que puede mostrar claramente las etiquetas y tamaños de nuestros grupos dentro de sus respectivas categorías de supermercados como se puede ver en la figura 26.

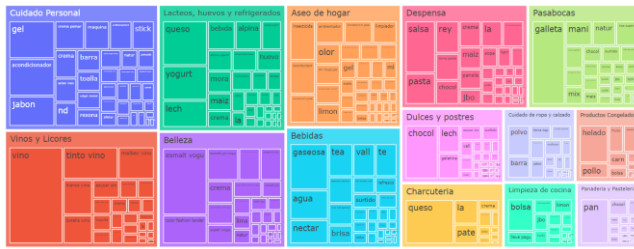


Figura 26. Nombres de alimentos sin etiquetar



Figura 27. Etiquetas específicas de datos

Después de todo el análisis de y la verificación de las clasificaciones, se puede ver que los resultados dan clasificación de los productos en la base de datos. más específicos en la figura 27. Esto ahora nos servirá para llevarlo a producción haciendo una limpieza y clasificar los productos de manera en la cual se tenga un manejo de datos de manera sencilla. Gracias al procesamiento del lenguaje natural se pudieron obtener estos resultados al quitar todas las caracteres innecesarios, esto como se explicó previamente para realizar y mejorar las clasificaciones que se tenían en las tareas anteriores.

Hemos visto lo eficientes que son los clustering de NLP y K-means para organizar texto sin etiquetar. Al igual que con cualquier modelo, no existe un modelo que se ajuste a todos los conjuntos de datos y casos de uso. Pero hay pasos prácticos a seguir y consideraciones para realizar pruebas.

La aplicación de esto en la categorización de productos de comercio electrónico produce muchos beneficios y hay muchas aplicaciones del mundo real: análisis de opiniones de publicaciones y reseñas en redes sociales, búsqueda difusa, recomendaciones de productos, filtrado de correo electrónico.

V. Conclusiones

El presente estudio permite concluir que gracias al método de construcción de agentes por medio de la metodología MASINA,

nos ayuda a tener una base para poder construir desde el inicio y tener un mapa de acción para definir las tareas y cómo estas están relacionadas entre sí. Además de eso, se puede definir la comunicación entre las diferentes tareas y los datos necesarios que necesitan intercambiar para poder ejecutar el agente inteligente, siendo esto importante ya que los datos no son atómicos para cada una de las tareas sino que estos datos son necesarios compartílos y que el agente los use para procesar y dar una respuesta.

Adicionalmente, gracias a los resultados de los experimentos se puede observar un desempeño alto en base a las tareas iniciales las cuales se definieron en el diseño del agente, con base en esto nos resulta satisfactorio usar el agente inteligente. Sin embargo, cabe resaltar que la percepción del agente es por medio del teclado, esto puede mejorar en el estado de que se puede llevar a producción haciendo que el agente perciba el ambiente y reaccione de manera adecuada gracias a la base de conocimiento que se ha generado.

Por último, vale la pena notar que este estudio tiene sus limitaciones, entre ellas la mencionada anteriormente respecto a la percepción del agente y el hecho de que este es un agente en desarrollo mas no en producción, este agente inteligente de MiPymes puede mejorar ayudando a las microempresas a mantenerse en el mercado resolviendo la problemática que se explicó. Los resultados obtenidos de los experimentos se pueden evidenciar cada uno de los resultados de las tareas, dándonos a entender que fue satisfactoria su implementación y además de eso se enfocó en resolver tareas las cuales sirvan al usuario final y no tareas que no se fueran a utilizar. La aplicación de los métodos investigados tuvieron un gran impacto dándonos beneficios de una aplicación robusta la cual servirá para varias microempresas.

REFERENCIAS

- [1] El Espectador. ¿Por qué el 70% de las empresas en Colombia fracasan en los primeros cinco años? El Espectador, 2018. <https://www.elespectador.com/economia/por-que-el-70-de-las-empresas-en-colombia-fracasan-en-los-primeros-5-anos-articulo-820897>
- [2] Rodríguez, J. Mala gestión de inventarios, una de las principales causas de quiebre de las PyMes, 2017. <https://mba.americaeconomia.com/articulos/notas/mala-gestion-de-inventarios-una-de-las-principales-causas-de-quiebre-de-las-pymes>
- [3] TICbeat.. 5 motivos por los que las pymes fracasan, 2016. <https://www.ticbeat.com/empresa-b2b/5-motivos-fracaso-pymes/>
- [4] Jose A. Reina C. Modelo ontológico de verificación de sistemas multiagentes diseñados bajo Masina, 2008. <https://repositorio.unal.edu.co/bitstream/handle/unal/24459/1/0110-18483-1-PB.pdf?sequence=1&isAllowed=y>
- [5] Henrik Brink, Joseph W. Richards and Mark Fetherolf. Real-World Machine Learning. Shelter Island, NY: Manning, 2017.
- [6] Q. Bi, K. Goodman, J. Kaminsky and J. Lessler, "Validate User", *Academic.oup.com*, 2019. [Online]. Available:

<https://academic.oup.com/aje/article/188/12/2222/5567515?login=true>. [Accessed: 12- Ago- 2021].

- [7] "What is Supervised Learning?", *Ibm.com*, 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/supervised-learning>. [Accessed: 12- Ago- 2021].
- [8] "What is Unsupervised Learning?", *Ibm.com*, 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/unsupervised-learning>. [Accessed: 13- Ago- 2021].
- [9] "ML Clustering: When To Use Cluster Analysis, When To Avoid It I Explorium", *Explorium*, 2020. [Online]. Available: <https://www.explorium.ai/blog/clustering-when-you-should-use-it-and-avoid-it/>. [Accessed: 13- Ago- 2021].
- [10] M. Deshmukh and R. Gulhane, *Importance of Clustering in Data Mining*, 7th ed. International Journal of Scientific & Engineering Research, 2016, p. 2.
- [11] Lily W. Clustering Product Names with Python -Part1, 2021.<https://towardsdatascience.com/clustering-product-names-with-python-part-1-f9418f8705c8>.
- [12] Lily W. Clustering Product Names with Python -Part2, 2021. <https://towardsdatascience.com/clustering-product-names-with-python-part-2-648cc54ca2ac>.