## **FASE 4: TRANSICIÓN DE LA FORMULACIÓN DE IDEAS A LOS DISEÑOS PRELIMINARES**

### **Descarte de ideas no factibles**

|  |  |
| --- | --- |
| *Idea no factible* | Justificación |
| *1.b. Técnica CRISP-DM (Cross Industry Standard Process for Data Mining).* | Debido a la naturaleza del proyecto, esta metodología para el desarrollo de proyectos excede los alcances del mismo. Exigiendo un mayor esfuerzo para obtener resultados que se pueden alcanzar con otra metodología más simple. Es por esto, que se descarta la idea de utilizar la técnica CRISP-DM. |
| *2.a. Modelo de Identificación.* | No va acorde a los objetivos que busca lograr la solución, ya que sólo evidenciar la existencia de objetos, eventos o actividades en un conjunto de datos no es suficiente para generar una solución al problema planteado. |
|  |  |
| *3.a Uso de grafos para ver qué producto está asociado con cuál en el entorno de ventas.* | Esta idea de presentación se descarta debido a su complejidad al momento de implementar la visualización gráfica de un grafo. Además, no es una alternativa que sea muy clara para representar varias asociaciones. |

Después de realizar el descarte de ideas no factibles, se tienen las siguientes alternativas:

1. Metodologías para el desarrollo de proyectos en Minería de Datos:
   1. Técnica Proceso de Generación de Conocimiento o KDD (*Knowledge Discovery in DataBases*).
2. Modelos (y algoritmos respectivos) para la Minería de Datos:
   1. Agrupación
      1. Algoritmo de Clústeres K-Means.
      2. Algoritmo de Clústeres K-Medoids.
      3. Autómatas finitos.
   2. Asociación
      1. Algoritmo de Fuerza Bruta.
      2. Algoritmo Apriori
      3. Algoritmo Partition
   3. Predicción
      1. Algoritmo de Bayes naive.
      2. Algoritmo de Regresión logística.
3. Presentación (gráfica) de resultados:
   1. Uso de histogramas para mostrar la frecuencia de los itemsets.
   2. Uso de tablas para mostrar las asociaciones encontradas.

**Diseños preliminares**

**Técnica Proceso de Generación de Conocimiento o KDD (*Knowledge Discovery in DataBases*).**

Este proceso surte cuatro pasos para la generación de conocimiento. Estas etapas pueden ser recursivas, es decir, que se retorna a ellas una y otra vez (proceso iterativo) a medida que se obtienen resultados preliminares que requieren replantear las variables iniciales.

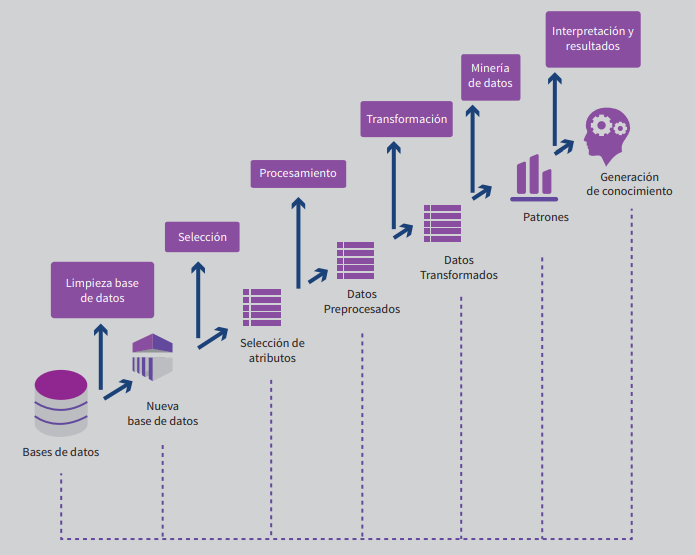


Figura . Proceso de KDD

Las etapas del proceso son:

1. Selección de los datos
2. Pre procesamiento de datos
3. Selección de características
4. Minería de Datos
5. Interpretación y Resultados

|  |  |  |
| --- | --- | --- |
| Etapas | | Descripción |
| **Selección de los datos** | | Consiste en la recolección y preparación de los datos. En este proceso se comprende la problemática asociada a la base de datos y se establecen objetivos. Y se identifican las variables que serán consideradas para la construcción del modelo de minería de datos |
| **Pre-procesamiento de datos** | Integración de datos | Se analiza si la base de datos requiere incluir o integrar información o variables que reposan en otras bases de datos, y que será relevante para el modelo de minería de datos |
| Reconocimiento y limpieza | Se depura el conjunto de datos respecto a valores atípicos, faltantes y erróneos (eliminación de ruido e inconsistencias). |
| **Selección de características** | Exploración y limpieza de datos | Aplicando técnicas de análisis exploratorio de datos se busca identificar la distribución de los datos, simetría, pruebas de normalidad y correlaciones existentes entre los datos. En esta etapa es útil el análisis descriptivo del conjunto de datos (clustering y segmentación, escalamiento, reglas de asociación y dependencia, reducción de la dimensión), identificación de datos nulos, ruido y outliers, así como el uso de matrices de correlación (si las variables son numéricas), diagramas (barras, histogramas, caja y bigotes), entre otras técnicas adecuadas de muestreo |
| Transformación | Se estandariza o normaliza la información (colocarla en los mismos términos de formato y forma). |
| Reducción de datos | Se disminuye el tamaño de los datos mediante la eliminación de características redundantes. |
| **Minería de Datos** | | Se puede definir como un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos, que a su vez, facilita la toma de decisiones y emplea técnicas de aprendizaje supervisado y no-supervisado. |
| **Interpretación y Resultados** | | Se analizan los resultados de los patrones obtenidos en la fase de MD, mediante técnicas de visualización y de representación, con el fin de generar conocimiento que aporte mayor valor a los datos. En esta fase se evalúan los resultados con los expertos y, si es necesario, se retorna a las fases anteriores para una nueva iteración |

Tabla . Proceso KDD

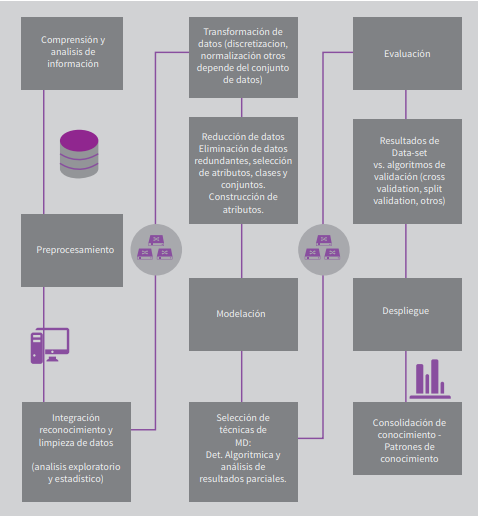
****

Imagen. Proceso KDD

|  |  |
| --- | --- |
| **Técnica de Agrupación utilizando el Algoritmo de Clústeres K-Means.** | **Técnica de Agrupación utilizando el Algoritmo de Clústeres K-Medoids.** |
| El análisis de conglomerados o Clustering, es una técnica que permite analizar y examinar datos que no se encuentran etiquetados, formando conjuntos de grupos a partir de su similitud. | |
| **Pseudocódigo** | |
|  |  |
| **Clústeres K-Means.** | **Clústeres K-Medoids.** |
| Ambos algoritmos intentan minimizar la distancia entre los puntos del mismo grupo y un punto particular que es el centro de ese grupo. | |
|  |  |
| **Similitudes** | |
| * La variación dentro del clúster disminuye con cada iteración del algoritmo. * El algoritmo siempre converge, depende en los centros iniciales. Para cualquiera de los algoritmos, uno debe ejecutarlo varias veces con diferentes inicios * La agrupación final depende de los centros iniciales del clúster. Diferentes comienzos dan como resultado diferentes agrupaciones finales. | |
| **Diferencias** | |
| * Devuelve centros que son promedios de puntos de datos. | * Generalmente devuelve un mayor valor de * Computacionalmente más difícil (debido al paso 2: calcular el medoid es más difícil que calcular el promedio) * Tiene la propiedad (potencialmente importante) de que los centros se encuentran entre los puntos de datos * Se basa en el cálculo de centroides (o medoides) minimizando la distancia absoluta entre los puntos y el centroide seleccionado, en lugar de minimizar la distancia cuadrada. |

|  |  |  |
| --- | --- | --- |
| **Técnica de Asociación utilizando el Algoritmo de Fuerza Bruta.** | | |
| Es una técnica de búsqueda también llamada búsqueda exhaustiva que consiste en observar todos los posibles candidatos a una solución. | | |
| **Pseudocódigo** | | |
|  | | |
| **Complejidad** | | |
| Complejidad temporal | | O(NMw) , M = |
| Complejidad computacional | | Dado un número únido d de item.   * Número total de itemsets: * Número total de posibles asociaciones: |
| **Estrategias para mejorar la generación de itemsets frecuentes** | | |
| * Reducir el número de candidatos (M)   + Búsqueda completa: .   + Utilizar técnicas de poda para reducir M. * Reducir el número transacciones (N)   + A medida que aumenta el tamaño de los itemsets, reducir N. * Reducir el número de comparaciones (NM)   + Usar estructuras de datos eficientes para almacenar los candidatos o transacciones. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Técnica de Asociación utilizando el Algoritmo Apriori.** | | | |
| **Pseudocódigo** | | | |
|  | | | |
| Diagramas de flujo | | | |
|  | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Técnica de Asociación utilizando el Algoritmo Partition.** | | | | |
| **Pseudocódigo** | | | | |
|  | | | | |
| **Características de Implementación** | | | | |
| * Este algoritmo recorre la base de datos sólo dos veces. La primera vez, cada partición es minada independientemente para encontrar todos los conjuntos de ítems frecuentes en la partición y luego se mezclan éstos para generar el conjunto de los conjuntos de ítems candidatos. Muchos de éstos pueden ser falsos positivos, pero ninguno falso negativo. * En la segunda pasada se cuenta la ocurrencia de cada candidato, aquellos cuyo soporte es mayor que el mínimo soporte especificado se retienen como conjuntos frecuentes. * Emplea el mecanismo de intersección entre conjuntos para determinar el soporte de dichos conjuntos, en este caso cada ítem en una partición mantiene la lista de los identificadores de las transacciones que contienen a dicho ítem. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Técnica de Predicción utilizando el Algoritmo de Bayes Naive.** | | | | |
| Naive Bayes es una técnica de clasificación y predicción que construye modelos que predicen la probabilidad de posibles resultados. Naive Bayes utiliza datos históricos para encontrar asociaciones y relaciones y hacer predicciones. | | | | |
| **Pseudocódigo** | | | | |
|  | | | | |
| **Ventajas** | | | | |
| * Muy simple, fácil de implementar y rápido. * Si el supuesto de independencia condicional NB se mantiene, entonces convergerá más rápido que los modelos discriminatorios como la regresión logística. * Incluso si la suposición de NB no se cumple, funciona bien en la práctica. * Necesita menos datos de entrenamiento. * Altamente escalable. Se escala linealmente con el número de predictores y puntos de datos. * Se puede utilizar para problemas de clasificación binarios y de múltiples clases de vidrio. * Puede hacer predicciones probabilísticas. * Maneja datos continuos y discretos. * No es sensible a las características irrelevantes. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Técnica de Predicción utilizando el Algoritmo de Regresión logística** | | | | |
| **Ventajas** | | | | **Desventajas** |
| * Es una técnica muy utilizada porque es muy eficiente. * No requiere demasiados recursos informáticos. * Es muy fácil de interpretar. * No requiere que las funciones de entrada se amplíen. * No requiere ningún ajuste. * Es fácil de regularizar. * Produce probabilidades pronosticadas bien calibradas. * Funciona mejor cuando elimina atributos que no están relacionados con la variable de salida, así como atributos que son muy similares (correlacionados) entre sí. * Es muy eficiente de entrenar. * También es una buena línea de base que puede usar para medir el rendimiento de otros Algoritmos más complejos. | | | | * No puede resolver problemas no lineales ya que su superficie de decisión es lineal. * Cuenta con una alta dependencia de una correcta presentación de sus datos. Esto significa que la regresión logística no es una herramienta útil a menos que ya haya identificado todas las variables independientes importantes. * Dado que su resultado es discreto, sólo puede predecir un resultado categórico. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Técnica de Predicción utilizando el Algoritmo de Regresión logística** | | | | |
| Un autómata finito es un modelo computacional que realiza un proceso de forma automática para producir una salida desde cierta entrada. Para implementar un análisis de datos usando este método, la idea será crear un autómata finito que compute automáticamente un proceso que siempre lleve a analizar unos datos de entrada. | | | | |
| **Definición** | | | | |
| Un autómata finito es una 5-tupla que se compone de la siguiente manera:  (*Q*, Σ, *q0*, δ, *F*), donde:  - *Q* es el conjunto finito de estados por el que está compuesto el autómata.  - Σ es el alfabeto con el que actúa (dominio).  - *q0* pertenece a *Q.*  - δ es la función que define las transiciones entre estados.  - *F* es el conjunto que contiene a los estados del autómata en los que termina (estados de aceptación). | | | | |
| **Tabla de Transición** | | | | |
|  | | | | |
| **Tipos** | | | | |
| **Autómatas Finitos Deterministas**  Es un autómata finito que además es un sistema determinista; es decir, para cada estado *q* ∈ *Q* en que se encuentre el autómata, y con cualquier símbolo *s* ∈ Σ del alfabeto (dominio) leído, existe siempre una transición posible de un estado a otro. Por ejemplo, el autómata a mano izquierda es un AFD. | | | | **Autómatas Finitos No Deterministas**  Es un autómata finito que, a diferencia de los autómatas finitos deterministas (AFD), posee al menos un estado *q* ∈ *Q*, tal que para un símbolo *s* ∈ Σ del alfabeto, exista más de una transición posible. |
| Se concluye que lo más apropiado para nuestra solución es implementar un autómata que automatice un procedimiento que analice los datos que se tienen, sería un Autómata finito determinista. En efecto, este permite tener todos los posibles resultados para cualquier tipo de dato que entre (símbolo), a diferencia del AFN. | | | | |
| **Implementación** | | | | |
| Se tiene un conjunto (C) de datos. El estado inicial (podría ser q1 en el ejemplo) del autómata representaría el subconjunto (S0) de datos con el que comienzo a analizar.  En seguida entra otro subconjunto de datos (S1) tal que S1 pertenece a C, es decir los subconjuntos son los datos que voy a ir analizando. El AFD, que por el momento se encuentra en el estado inicial, decide que procedimiento realizar dependiendo de la clasificación que tenga S2. (esta clasificación estaría representada por los símbolos posibles que tenga el autómata). En el diagrama a continuación estos serían los símbolos: 0 y 1.  Se continuaría de esta forma hasta que se llegue a un estado final. En el diagrama sería: q0    Por ejemplo. En el caso hipotético de que se tenga el primer subconjunto S0 (q1) y entre otro subconjunto de datos S1 de tipo 1, entonces el autómata procedería a realizar las operaciones de análisis que lleven directamente al estado final q0. En el cual la información que se tiene se puede considerar un análisis relevante. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Presentación de resultados mediante el uso de histogramas para mostrar la frecuencia de los itemsets.** | | | | |
| **Diseño** | | | | |
|  | | | | |
| Se utilizan los componentes de Visual Studio para crear histogramas con la frecuencia de los itemsets de un tamaño dado. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Presentación de resultados mediante el uso de tablas para mostrar las asociaciones encontradas.** | | | | |
| **Diseño** | | | | |
| 1. **Pasillo de comidas - Leche, Huevos, Pan** 2. **Pasillo de licores - Licor, Red/Blush Vino, Cerveza, Soda** 3. **Pasillo de desayuno - Cereal Yogurt, Arroz, Avena** | | | | |