# Paso 7. Implementación del diseño

Implementación en el lenguaje de programación C# bajo la plataforma Visual Studio 2017

Lista de tareas a implementar:

* Implementación del modelo con sus diferentes clases, objetos, métodos, relaciones y atributos que ayuden a cumplir los requerimientos establecidos.
* Implementación de estrategias BruteForce, A-priori y F(k-1)xF1
* Implementación de la interfaz gráfica en cuanto a la visualización de clientes por medio de Gmaps
* Implementación de la interfaz gráfica en cuanto a recomendaciones sobre los productos
* Implementación de la interfaz gráfica en cuanto a similitud y recuperación de clientes.
* Implementación de la interfaz gráfica en cuanto a la visualización de graficas

Construcción en el lenguaje C#

Estrategias

public void BruteForceMethod(int tamanho, int numArti)

{

int[] darItemCode = masFrecuentesMetodo(numArti);

brute.generarAsociaciones(tamanho, darItemCode, Ventas,minSuport,minCon);

reporte += brute.Reporte;

}

public void AprioriMethod(int tamanho, int numArti)

{

TamanhoAgrupaciones = tamanho;

NumArticulos = numArticulos;

int[] darItemCode = masFrecuentesMetodo(numArti);

apriori.Generar(tamanho, ventas, darItemCode);

izq = apriori.Implicantes;

der = apriori.Implicados;

List<List<int>> completas = apriori.Completas;

int reglas = izq.Count;

for(int i = 0; i < reglas; i++)

{

ConvertirAsociaciones(izq.ElementAt(i), der.ElementAt(i));

}

foreach (var n in completas)

{

ConvertirAsociacionesCompletas(n);

}

reporte += apriori.Reporte;

}

public void FkMinus1XF1Method(int tamanho, int numArticulos)

{

int[] F1 = masFrecuentesMetodo(numArticulos);

fkmin.FkMinus1XF1(tamanho, F1, Ventas);

reporte += fkmin.Reporte;

}

public List<string> estrategiaFP(List<Venta> ventas)

{

List<string> retorno = new List<string>();

String mensaje = "";

int supCount = (int)(minSuport \* ventas.GroupBy(a => a.DocNum).Count());

algoritmoFP = new FP(supCount);

List<List<Object>> listaRepeticiones= new List<List<Object>>();

List<List<Object>> poda = new List<List<Object>>();

listaRepeticiones = algoritmoFP.repeticionesUnitarias(ventas);

poda = algoritmoFP.objetosCumplen(listaRepeticiones);

algoritmoFP.ordenarMayorMenor(poda);

Arbol tree = new Arbol((supCount/2));

List<string> listaEncontrada = new List<string>();

var x = ventas.GroupBy(n => n.DocNum);

foreach (var m in x)

{

String[] trans = new String[m.Count()];

int contador = 0;

foreach (var s in m)

{

trans[contador] = s.ItemCode;

contador++;

}

for (int i = 0; i < poda.Count; i++)

{

if (trans.Contains(poda[i][0].ToString()))

{

listaEncontrada.Add(poda[i][0].ToString());

}

}

tree.crearArbol(listaEncontrada);

listaEncontrada.Clear();

}

List<string> itemsFrecuentes = new List<string>();

itemsFrecuentes = tree.generarItemsSetsFrecuentes(tree.raiz, itemsFrecuentes);

mensaje += "El mayor item set es \n";

foreach (string item in itemsFrecuentes)

{

// string nombre = busquedaNombreItem(item);

retorno.Add(item);

string nombre = item;

mensaje +=nombre+ "\n";

}

return retorno;

}