Integrantes del grupo:

Sebastián Barrera A00358271
Santiago Hurtado A00362570
Miguel Sarasti A00364978
Sebastián Morales A00365920

Objetivos

Terminales del Curso

OT1. Desarrollar programas de computador para solucionar problemas de ingeniería de mediana complejidad que involucren el análisis de volúmenes significativos de datos.
OT2. Evaluar los resultados del análisis de datos utilizando teoría de probabilidades e inferencia estadística.
OT3. Evaluar programas de computador utilizando técnicas de análisis de algoritmos y estructuras de datos.
OT4. Sustentar apropiadamente los resultados de cada una de las etapas del proceso de desarrollo de soluciones de ingeniería de forma escrita y oral.

Descripción del problema

Una reconocida clínica de cardiología en Cali necesita saber si sus pacientes son propensos a sufrir un ataque cardíaco, pero dado a que este proceso suele tomar mucho tiempo, el cual puede ser vital para el paciente, decidieron que lo mejor era contratar un grupo de expertos en programación para saber si es posible automatizar este proceso. Para ello decidieron reunir un historial de datos de los pacientes a las que se les ha hecho pruebas para saber si eran o no propensos a sufrir ataques cardiacos, los datos del paciente que incluía este historial eran la edad, el género, el tipo de dolor de pecho (angina típica, angina atípica, dolor no anginoso, asintomático), presión arterial, el nivel de colesterol, el nivel de azúcar en la sangre, si ha sufrido una angina (dolor en el pecho), los resultados de un electrocardiograma en reposo, la frecuencia cardiaca máxima y el resultado de si es propenso o no a sufrir ataques cardiacos. La clínica ordenadamente ha recopilado esta información en un dataset y ha contratado al mejor equipo de ingenieros de software de la ciudad de Cali para este trabajo.

Aparte de mostrar el resultado la clínica también solicitó que el programa permitiera visualizar todos los datos en una tabla los cuales pueden ser filtrados por el tipo de angina, también se necesita que se grafiquen algunos datos para que sea más fácil de visualizar y también comprender los factores más importantes del dataset, estos gráficos deben ser: la cantidad de mujeres y hombres que son propensos a sufrir ataques, los pacientes que han tenido angina inducida por ejercicio contra los que no, rangos de edad de los pacientes, los diferentes tipos de dolor en el pecho y por último un gráfico sobre el nivel de colesterol de los pacientes que son propensos contra los que no a sufrir ataques al corazón.

Dado a que no son muy buenos con la tecnología, les piden que los datos de los pacientes los puedan ingresar de dos formas, la primera es mediante una base de datos con el mismo formato que la base presentada anteriormente pero sin la columna de la solución, y la otra opción es en la que puedan ir agregando cada registro manualmente de manera intuitiva y cuando esté listo poner a correr el programa y este les muestre por pantalla el resultado pero que también les permite almacenarlo en el ordenador.

Base de datos de tomada: https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

Requerimientos funcionales

El sistema debe estar en la capacidad de:

- Predecir con los datos de un paciente, si es probable que esta sufra un ataque cardíaco o no, con los datos tomados del dataset, a su vez mediante un árbol de decisión propio implementado por el equipo de trabajo o uno utilizando una librería de c# sobre árboles de decisión.

- Visualizar un gráfico de reportes de los datos que muestre:
    - la cantidad de mujeres y hombres que son propensos a sufrir un ataque al corazón.
    - Los pacientes que han tenido una angina inducida por ejercicio contra los que no han sufrido dicho padecimiento.
    - Los diferentes rangos de edad de los pacientes
    - Un gráfico de pastel que muestre la cantidad de los pacientes que han tenido un dolor en el pecho, clasificando esto por su tipo.
    - Un gráfico de líneas que indiquen los niveles de colesterol de cada paciente.

- Cargar y mostrar el dataset en una tabla de datos donde se muestre al usuario la información correspondiente.
- Escribir un archivo que sea de un tipo adecuado para mostrar la solución del problema, además que se muestra en la interfaz del programa.
- Una interfaz intuitiva para el manejo del programa, esta debe permitir desarrollar al máximo las funcionalidades del sistema, para su buen uso y correcto funcionamiento.
- El sistema debe estar en la capacidad de realizar las siguientes funciones:
    - Agregar registros de pacientes con datos como: edad, el género, el tipo de dolor de pecho (angina típica, angina atípica, dolor no anginoso, asintomático), presión arterial, el nivel de colesterol, el nivel de azúcar en la sangre, si ha sufrido una angina (dolor en el pecho), los resultados de un electrocardiograma en reposo, la frecuencia cardiaca máxima.
    - Borrar registros
    - Actualizar registros
- Deberá tener una opción que permita listar los datos en una tabla y filtrar los registros basados en los campos del dataset, dicho reporte es configurable mediante un combo box que lista todos los campos:
    - Categórico: seleccionar una opción del combo box para identificar, el sexo, los tipos de dolores en el pecho, la angina inducida por ejercicio, resultados del electrocardiograma y el azúcar en la sangre en ayunas.

Filtrar un reporte por paciente donde me muestre los datos organizados por cada atributo, este debe ser mediante un combo box de opciones que me muestre los n valores de cada columna:

Ejemplo:

Sexo: (1) Masculino o (0) femenino

Angina inducida por ejercicio: (1) Si o (0) No

Resultados del electrocardiograma:

- Valor 0: normal
- Valor 1: Presenta anomalía
- Valor 2: Muestra hipertrofia muscular

El nivel del azúcar en ayunas > 120 mg/ dl: (1)Si o (0)No

Dolor en el pecho:

- Valor 1: angina típica
- Valor 2: angina atípica
- Valor 3: dolor no anginoso
- Valor 4: asintomático

- Numérico: un rango de valores para filtrar los datos del dataset por:
  - Edad
  - Nivel de colesterol
  - Presión arterial en reposo
  - La frecuencia máxima cardiaca.

- Cadena: en este caso sería el id del paciente.

Requerimientos no funcionales:

- Ser desarrollado en el lenguaje de programación C#.
- Realizar las interfaces del programa en Windows form.
- El programa al realizar sus operaciones (agregar, borrar, cargar, actualizar) debe ser lo mas eficiente posible empleando un mínimo de tiempo.
- El programa debe de utilizar la Herramienta GMaps para la representación de un mapa de la clínica (extra al proyecto).
- El sistema debe contar con un módulo de ayuda tanto para la especificación de atributos, información del dataset y un pequeño manual o guía para utilizarlo.
- La aplicación debe manejar el idioma inglés.
- El porcentaje de predicción debe ser mayor a 85% para predecir el resultado.

Acerca del conjunto de datos

- Edad: edad del paciente
- Sexo: sexo del paciente
- intercambio: angina inducida por ejercicio (1 = sí; 0 = no)
- ca: número de vasos principales (0-3)
- cp: tipo de dolor de pecho tipo de dolor de pecho
    - Valor 1: angina típica
    - Valor 2: angina atípica
    - Valor 3: dolor no anginoso
    - Valor 4: asintomático
- trtbps: presión arterial en reposo (en mm Hg)
- chol: colestoral en mg / dl obtenido a través del sensor de IMC
- fbs: (azúcar en sangre en ayunas> 120 mg / dl) (1 = verdadero; 0 = falso)
- rest_ecg: resultados electrocardiográficos en reposo
    - Valor 0: normal
    - Valor 1: tener anomalía de la onda ST-T (inversiones de la onda T y / o elevación o depresión del ST> 0,05 mV)
    - Valor 2: muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes
- thalach: frecuencia cardíaca máxima alcanzada
- objetivo: 0 = menos probabilidad de ataque cardíaco 1 = más probabilidad de ataque cardíaco

:

Método de la ingeniería

1. **<u>Identificación del problema:</u>**

Se requiere predecir teniendo en cuenta los datos de un paciente si este es propenso a sufrir un ataque al corazón.

Necesidades:

- Identificar la historia clínica del paciente
- Identificar los datos irrelevantes del dataset
- Predecir una variable objetivo
- Identificar la variable objetivo
- Predecir si el paciente va a tener un ataque al corazón
- Hacer inferencia del dataset
- Manejo de gráficas en base a los datos de dataset
- Generación Reporte de los datos filtrados por categoría, cadena o numérico.
- Opciones y búsqueda del dataset
- Uso de herramientas estadísticas
- Conocer acerca de los ataques cardíacos
- Análisis de experimentos

2. **Recopilación de información:**
   Se realizo una búsqueda de conceptos previos al problema para realizar una búsqueda creativa de soluciones.

Data Set:

Un conjunto de datos es una colección de datos habitualmente tabulada. Un conjunto de datos contiene los valores para cada una de las variables, como por ejemplo la altura y el peso de un objeto, que corresponden a cada miembro del conjunto de datos. Cada uno de estos valores se conoce con el nombre de dato. El conjunto de datos puede incluir datos para uno o más miembros en función de su número de filas. Conjuntos de datos tan grandes que aplicaciones tradicionales de procesamiento de datos no los pueden tratar se llaman big data.

Tomado de: https://es.wikipedia.org/wiki/Conjunto_de_datos

Gráficos estadísticos

Un gráfico estadístico es una representación visual de datos estadísticos, de forma que estos puedan ser interpretados, analizados y entendidos de forma más sencilla.

Pero ¿Qué es un dato estadístico? Según el INEI (Instituto Nacional de Estadística e Informática), es el "valor o característica cuantitativa de un objeto de conocimiento, con referencia de tiempo y espacio".

Los datos estadísticos pueden ser de dos tipos: cuantitativos (cantidades o valores numéricos) y cualitativos (cualidades que no pueden expresarse numéricamente).

Tipos de gráficos:

1. Histograma

El histograma es la herramienta fundamental de la estadística descriptiva. Resume la variable numérica de un modo sencillo y eficaz. Utiliza las famosas tablas de frecuencias. Es un diagrama de barras. La altura de las barras es la frecuencia. Y cada barra se sitúa en su debida clase.

2. Gráfica lineal o gráfico de líneas

Es un tipo de gráfico estadístico donde los valores se representan con un punto y se unen por medio de líneas, con el fin de visualizar una tendencia en el tiempo.

En el eje horizontal se posiciona la variable que indica las unidades de tiempo y en el vertical se introduce la escala de la variable (pueden presentarse varias variables).

¿Cuándo se usa la gráfica lineal? Cuando necesites mostrar las tendencias de una serie de datos de un periodo determinado (minutos, horas, días, semanas, meses o años).

3. Gráfica circular o pastel

Es otro de los gráficos estadísticos más sencillos y usados. También conocido como "gráfico de sectores" o "torta", el gráfico circular, como su mismo nombre lo dice, es representado por un círculo que simboliza la totalidad y se expresa en porcentajes.

¿Cuándo se usa una gráfica circular? Cuando necesites recalcar proporciones de un total y las categorías sean pocas. No es recomendable usarlo cuando tienes muchas variables, pues genera confusión y el resultado sería incomprensible.

4. Gráfica poligonal o polígono de frecuencias

Es un gráfico estadístico conocido también como polígono de frecuencias. A diferencia del histograma (similar a un gráfico de barras), esta gráfica une los vértices superiores de las barras de un diagrama, formando una línea constante e irregular llamada gráfica poligonal.

Se representa en un eje X (horizontal) y un eje en Y (vertical). Establecidas las variables (eje X) y las frecuencias (eje Y) se marcan los puntos, para luego unirlos y formar una línea poligonal.

Tomado de: https://www.crehana.com/co/blog/marketing-digital/conoce-la-importancia-de-usar-graficos-estadisticos-en-tu-empresa/

Ataque al corazón: La mayoría de los ataques cardíacos son provocados por un coágulo que bloquea una de las arterias coronarias. Las arterias coronarias llevan sangre y oxígeno al corazón. Si el flujo sanguíneo se bloquea el corazón sufre por la falta de oxígeno y las células cardíacas mueren.
https://medlineplus.gov/spanish/ency/article/000195.htm

Árboles de decisión: Un árbol de decisión **es un modelo predictivo** que divide el espacio de los predictores agrupando observaciones con valores similares para la variable respuesta o dependiente.

Para dividir el espacio muestral en sub-regiones es preciso aplicar una serie de reglas o decisiones, para que cada sub-región contenga la mayor proporción posible de individuos de una de las poblaciones.

Si una sub-región contiene datos de diferentes clases, se subdivide en regiones más pequeñas hasta fragmentar el espacio en sub-regiones menores que integran datos de la misma clase.

El tipo de problema a resolver dependerá de la variable a predecir:

- Variable dependiente: estaríamos ante un problema de regresión.
- Variable categórica: nos enfrentaremos a un problema de clasificación.

Ventajas y desventajas de los árboles de decisión

Al hacer uso de esta herramienta surgen ventajas e inconvenientes.

Ventajas

- Son fáciles de construir, interpretar y visualizar.
- Selecciona las variables más importantes y en su creación no siempre se hace uso de todos los predictores.
- Si faltan datos no podremos recorrer el árbol hasta un nodo terminal, pero sí podemos hacer predicciones promediando las hojas del sub-árbol que alcancemos.
- No es preciso que se cumplan una serie de supuestos como en la regresión lineal (linealidad, normalidad de los residuos, homogeneidad de la varianza, etc.).
- Sirven tanto para variables dependientes cualitativas como cuantitativas, como para variables predictoras o independientes numéricas y categóricas. Además, no necesita variables *dummys*, aunque a veces mejoran el modelo.
- Permiten relaciones no lineales entre las variables explicativas y la variable dependiente.
- Nos podemos servir de ellos para categorizar variables numéricas.

Desventajas

- Tienden al sobreajuste u *overfitting* de los datos, por lo que el modelo al predecir nuevos casos no estima con el mismo índice de acierto.
- Se ven influenciadas por los *outliers*, creando árboles con ramas muy profundas que no predicen bien para nuevos casos. Se deben eliminar dichos *outliers*.
- No suelen ser muy eficientes con modelos de regresión.
- Crear árboles demasiado complejos puede conllevar que no se adapten bien a los nuevos datos. La complejidad resta capacidad de interpretación.
- Se pueden crear árboles sesgados si una de las clases es más numerosa que otra.
- Se pierde información cuando se utilizan para categorizar una variable numérica continua.

https://www.maximaformacion.es/blog-dat/que-son-los-arboles-de-decision-y-para-que-sirven/#id1

Machine learning:

Es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. **La máquina que realmente aprende es un algoritmo** que revisa los datos y es capaz de predecir comportamientos futuros. Automáticamente, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana. https://cleverdata.io/que-es-machine-learning-big-data/

3. Búsqueda de soluciones creativas

Para la búsqueda de soluciones creativas se determinaron ciertos criterios importantes en la funcionalidad de nuestra aplicación, estos fueron la selección y búsqueda de un dataset, el uso posible de la georreferenciación, el uso de los árboles de decisión, uso de la librería del mapa, la filtración de los datos, además de los gráficos que se van a utilizar en dicho proyecto.
Por otro lado, las técnicas que se usaron para la escogencia de ideas fueron los métodos de lluvia de ideas y los aspectos positivos, negativos e interesantes (PNI).

1. **Dataset:**
    1. Seleccionar un dataset con la información relevante para la predicción de ataques cardíacos en fuentes como UCI Machine Learning Repository, Kaggle, DANE, entre otros.
    2. Buscar un dataset con los pacientes y síntomas para la predicción de ataques cardíacos de algunos hospitales de Colombia.
    3. Organizar un dataset con la información relevante por medio de la combinación de datos de diferentes datos encontrados en diferentes fuentes.

2. **Georreferenciación:**
    1. Buscar un dataset de coordenadas de las ubicaciones de los pacientes
    2. Poner una dirección de la clínica donde estamos realizando el proyecto

3. **Uso de árboles de decisión:**
    1. Crear nuestra propia estructura del árbol decisión para abarcar y resolver correctamente la solución del sistema.
    2. Utilizar una librería de arboles de decisión en c#, para abarcar la solución del sistema.
    3. Utilizar ambas estructuras de predicción de datos, tanto la librería de c#, como nuestra propia estructura para predecir la variable objetivo.

4. **Filtración de datos:**

    Categórico: Filtrar los datos en un combo box por:

    ✓ Sexo
    ✓ Tipo de dolor en el pecho
    ✓ Angina
    ✓ Resultados del electrocardiograma
    ✓ El azúcar en la sangre en ayunas.

Cadena: Permitir la búsqueda por texto de:

✓ Id

Numérico: Filtrar un rango de valores entre:

✓ Edad
✓ Nivel de colesterol
✓ Presión arterial en reposo
✓ La frecuencia máxima cardiaca.

5. Gráficos:

1. Un gráfico de pastel de la cantidad de mujeres y hombres que son propensos a sufrir un ataque al corazón.
2. Un gráfico de barras de los pacientes que han tenido una angina inducida por ejercicio contra los que no han sufrido dicho padecimiento.
3. Un (histograma) con los diferentes rangos de edad de los pacientes
4. Un gráfico de dona/rosquilla que muestre la cantidad de los pacientes que han tenido un dolor en el pecho, clasificando esto por su tipo.
5. Un gráfico de líneas que indiquen los niveles de colesterol de cada paciente.
6. Un gráfico que muestre los pacientes que pueden padecer de un ataque cardiaco.

6. Exportar solución del sistema

1. Exportar la solución con la predicción en un archivo pdf.
2. Exportar la solución del sistema en un archivo de diferentes tipos (pdf, png, jpg, Word, Excel) como lo desee el usuario.
3. Exportar la solución con la predicción de los datos en un archivo csv.

7. Visualización

1. Desplegar la ventana principal donde se encuentren, las diferentes opciones para el filtrado con botones, además otra interfaz que permita abrir otras ventanas con la tabla y gráficos.
2. Representar todo en una sola ventana grande.
3. Visualizar las interfaces paso por paso, ejemplo en una interfaz cargar el archivo, en otra la tabla y sus funciones, y en otra los gráficos.

4. Representar los gráficos en una ventana aparte, donde el usuario escoja el tipo de gráfico que desea ver, adicional a esto la tabla y el mapa se verán en una ventana, donde se permita realizar todos los requerimientos.

8. Lenguaje de codificación:

1. Java: Es un lenguaje que sabemos usar, tiene muchas librerías especiales para inteligencia artificial, esto es bueno para predecir nuestro resultado, además de que tenemos un amplio conocimiento del lenguaje.

2. C#: Es un lenguaje de Microsoft el cual también tiene un amplio portfolio en librerías de IA, todo esto para desarrollar proyectos que abarcan grandes datasets y además, su funcionalidad, su interfaz gráfica es muy intuitiva y fácil de manejar.

3. Python: Es un lenguaje interpretado, especial para el machine learning, este puede ofrecer infinidad de librerías para resolver dichos problemas de IA.

4. DISEÑOS PRELIMINARES

A continuación, se detallarán las ideas validadas y descartadas dando una explicación que especifique detalladamente la idea.