Jonh Stiven Arboleda Gallego A00359103
Isabella Ortiz Jaramillo A00358451

1. IDENTIFICACIÓN DEL PROBLEMA

Netflix cuenta con 200 millones de usuario activos aproximadamente, siendo una de las principales compañías de streaming por suscripción y también una de las pioneras en este campo. Como toda compañía Netflix cuenta con ciertas falencias en la precisión a la hora de sugerir películas según nuestro historial, así que se decidió realizar una mejor clasificación de las películas teniendo en cuenta más factores que el género de la película como normalmente lo realiza la plataforma, factores como: la fecha de lanzamiento, el director de la película, el país de origen y el reparto. Las sugerencias con factores diferentes y más específicos se hacen con el fin de que los usuarios reciban recomendaciones más acordes a sus gustos, a lo que frecuentan en sus películas y a lo que buscan ver en el momento.

La base de datos manejada fue elegida de kaggle.com, siendo de libre uso.

2. RESEARCH

REQUERIMIENTOS FUNCIONALES

1. Realizar tablas con la información de películas obtenidas del dataset de películas en la plataforma de Netflix.
2. Realizar 5 gráficos representando información importante para el estudio de datos.
    1. Un gráfico ilustrando la cantidad de películas por años que existen en la plataforma
    2. Un gráfico ilustrando la cantidad de películas por algunos países
    3. Un gráfico que muestra algunos de los directores más conocidos en la plataforma
    4. Un gráfico que ilustre las películas por el género.
    5. Un gráfico ilustrando la duración de las películas por intervalos.
3. Filtrar la información de acuerdo al criterio escogido por el usuario.
    1. Primer criterio: Se filtran las columnas establecidas internamente con un valor de cadena, como lo son: Id de la película, título de la película, fecha de publicación, nombre del director, elenco de la película, país de origen.
    2. Segundo criterio: Se filtran las columnas establecidas internamente con un valor numérico, como lo son: Año de públicas en Netflix, duración en minutos de la película. Estos valores se filtran por medio de intervalos.
    3. Tercer criterio: Se filtran las columnas establecidas internamente con un valor categórico, como lo es: la calificación de la película.
4. Listar los registros y filtrarlos por valores particulares de una columna.
5. Sugerir las películas más adecuadas al usuario de acuerdo a su historial, basándonos en el entrenamiento de un árbol de decisión de implementación propia.

6. Usar el atributo género para el entrenamiento del árbol de decisión, así se hará la clasificación sobre los valores que puede obtener este atributo en nuestro dataset.

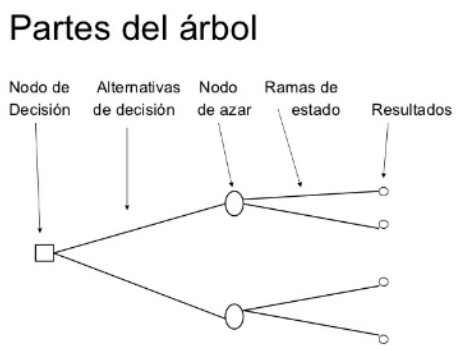
REQUERIMIENTOS NO FUNCIONALES

1. Implementación propia de un árbol de decisión para las sugerencias al usuario.[1]
2. Implementación de un árbol de decisión en base a librerías de C#

Para la solución de este problema específicamente es necesario conocer acerca de diferentes temas, algunas de estos temas pueden ser:

Árboles de decisión

Un árbol de decisiones es una representación esquemática que facilita la toma de decisiones, especialmente cuando existen riesgos, costos, beneficios y múltiples opciones al representar visualmente las diferentes posibilidades que existen ante un escenario; además, de las posibles consecuencias que cada escenario podría traer. Su nombre se da debido al parecido que tiene el esquema con las ramas de un árbol; un árbol de decisiones puede utilizarse en cualquier aspecto de la vida cotidiana, desde decisiones difíciles en la familia, hasta aplicaciones complejas en los negocios y en la inteligencia artificial.[2] [3]



Además, estos árboles deben de cumplir unas reglas[4] específicas para su implementación, las cuales son:

1. Al comienzo del juego se da un nodo inicial que no es apuntado por ninguna flecha, es el único del juego con esta característica.
2. El resto de los nodos del juego son apuntados por una única flecha.
3. De esto se deduce que hay un único camino para llegar del nodo inicial a cada uno de los nodos del juego. No hay varias formas de llegar a la misma solución final, las decisiones son excluyentes.

[1] https://www.kaggle.com/shivamb/netflix-shows
[2] https://profesionistas.org.mx/que-son-y-como-hacer-arboles-de-decisiones/
[3] https://www.gestiondeoperaciones.net/procesos/arbol-de-decision/
[4] https://es.wikipedia.org/wiki/Árbol_de_decisión#Reglas

Machine Learning[5]

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. Veamos cómo funciona. Esta herramienta es de uso común en plataformas de recomendación, plataformas como Netflix, Disney+, Spotify, Google, Microsoft, Amazon, etc.

3. SOLUCIONES CREATIVAS

Para la solución de este punto llevamos a cabo el método de **lluvia de ideas** [6]la cual consiste en seguir 4 pasos para un desarrollo óptimo, dichos pasos son:

- **Suspender el juicio**. Eliminar toda crítica. Cuando brotan las ideas no se permite ningún comentario crítico. Se anotan todas las ideas.

- **Pensar libremente**. Es muy importante la libertad de emisión. Las ideas locas están bien. Las ideas imposibles o inimaginables están bien. De hecho, en cada sesión tendría que haber alguna idea suficientemente disparatada que provocara risa a todo el grupo.

- **La cantidad es importante**. Hace falta concentrarse en generar un gran número de ideas que posteriormente se puedan revisar. Cuanto mayor sea el número de ideas, más fácil es escoger entre ellas.
- **El efecto multiplicador**. Se busca la combinación de ideas y sus mejoras. Además de contribuir con las propias ideas, los participantes pueden sugerir mejoras de las ideas de los demás o conseguir una idea mejor a partir de otras dos. ¿Qué tiene de bueno la idea que han dicho? ¿Qué se puede hacer para mejorarla o para hacerla menos salvaje?

Dicho lo anterior, procedemos a las ideas.

[5]https://cleverdata.io/que-es-machine-learning-big-data/
[6]https://es.wikipedia.org/wiki/Lluvia_de_ideas#:~:text=La%20lluvia%20de%20ideas%2C%20también,originales%20en%20un%20ambiente%20relajado.

1. **Búsqueda manual:** El usuario guarde las películas que más le llamen la atención para verlas en un futuro, cuando desee ver las películas ya guardadas.
2. **Guardar las películas vistas:** Guardar las películas ya vistas por el usuario en una lista, al completar cierta cantidad de películas vistas comenzar a recomendar de acuerdo a su género más visto.
3. **Recomendar a través de sus vistas:** Por medio de árboles de decisión tomar decisiones de recomendaciones en base las películas que busca, teniendo en cuenta el género, los directores, etc.
4. **Utilizar listas de películas predeterminadas:** Tener listas con diversas películas por género, para recomendarlas de acuerdo al género más visto.
5. **Recomendación visual:** Dar diferentes listas de recomendaciones sin base en ningún gusto del usuario, solo mostrando diferentes tipos de películas

- **Soluciones de interfaz:**
    1. Se muestra el árbol de decisión con el formato de guardar carpetas, con formato texto.
    2. Se muestra el árbol de decisión en una forma de PDF.
    3. Se escogen diferentes imágenes árboles de decisión de internet para mostrarlas y que el usuario se haga una idea.
    4. Los datos de las películas se muestran en un DataGridView para que el usuario pueda revisarlas y realizar filtros específicos.
    5. Los datos de las películas se van a ver únicamente en un Excel y el usuario tendrá que abrirlo para ver la información.
    6. Los datos de las películas se muestran de forma resumida con gráficos.

- Se decidió separar las ideas obtenidas entre parte de implementación (BackEnd) y la parte visual (FrontEnd) para poder tener una mejor decisión a la hora de descartar ideas, ya que permite mezclar las diferentes ideas de manera objetiva y enfocada a la mejor solución.

4. PASAR AL DISEÑO PRELIMINAR

- Se descarta la **búsqueda manual**, ya que no cumple con el objetivo del proyecto para brindar una mejor experiencia al usuario.
- Se descarta **guardar películas vistas** ya que puede funcionar después de un tiempo, al inicio será ineficiente para el cliente y tomará algún tiempo antes de poder cumplir su función.
- Se descarta **utilizar listas de películas predeterminadas** ya que la recomendación no será tan acertada si el usuario decide cambiar de género por alguna ocasión.
- Se descartan las opciones de solución visual de mostrar el árbol el árbol en PDF, elegir imágenes de internet y la información únicamente en Excel ya que es ineficiente para el usuario no poder tener información personalizada y de mejor manejo.
- Por último, se decide acoger en conjuntos las ideas de **recomendar a través de sus vistas,** mostrar la información del árbol en formato TXT y la opción de los gráficos conjuntos a mostrar la información de las películas en un DataGridView.
    - La opción acogida de mostrar información resumida en gráficos va de la mano con la opción de mostrar la información de las películas en un DataGridView

- Las opciones que se escogieron fueron tomadas en conjunto y en base a dar una mejor solución al sistema, que le permita al usuario tener una mejor experiencia y cumpliendo con los requerimientos establecidos para el proyecto.

5. Evaluación y selección de la solución.

Para la solución de este proyecto fueron usadas diferentes herramientas como:

- **VisualStudio:** IDE para el desarrollo de aplicaciones basadas en C# el cual permite y soporta .Net, herramienta fundamental para nuestro desarrollo
- **GitHub:** Plataforma de almacenamiento de código en la nube, es especial para poder trabajar en conjunto y desde diferentes lugares y/o dispositivos.
- **VisualParadigm** y **LucidChart:** Permite el desarrollo de la documentación como diagramas de clases, diagramas de secuencia, diagrama de objetos, entre otros.

A continuación la evaluación, selección y desarrollo de la solución:

- En general, la parte visual se decide realizar separando cada función principal del programa por “ventanas” o “pestañas” para una mayor comodidad al usuario, permitiéndole así una mejor ubicación en el programa sin complicaciones.
- Para mostrar la información de las películas y que el usuario pueda verla de una forma cómo se decide usar una interfaz para implementar un DataGridView que cargue los datos desde un archivo de Excel donde se encuentran todas las películas de la plataforma. Así se obtiene una mejor experiencia para el usuario en comodidad y agilidad a la hora de leer los datos.
- Se decide que en la misma interfaz donde se encuentran los datos aparezcan las opciones de filtro para ahorrar agilidad y tiempo a la hora de realizar estos filtros. Así mismo, en el momento de filtrar se hará sobre el DataGridView que ya existe.
- Se decide mostrar la información de las películas resumida en gráficos en una ventana diferente de donde se tiene el DataGridView para mejorar la visibilidad de la aplicación y una mejor lectura de la información presentada en los gráficos.
- Se decide crear una ventana diferente para que el usuario pueda observar el árbol de decisión de forma gráfica.
    - En esta ventana existirá de forma especial la opción de elegir entre el árbol de implementación propia y el árbol basado en librerías.
- El árbol de nuestra implementación cuenta con diferentes clases, las cuales son Tree, Node, Question, Split y Leaf (hereda de Node), implementadas en el modelo.