

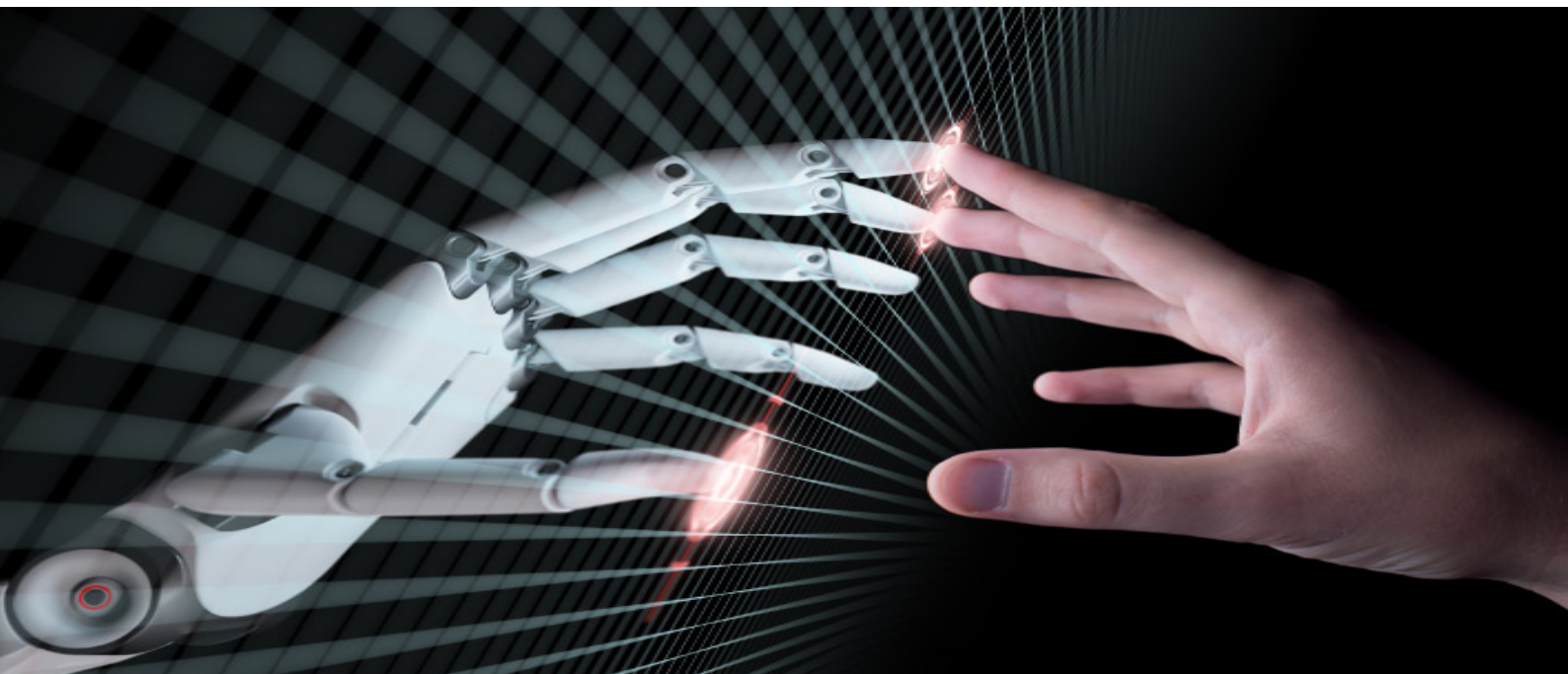


IntelliShow

- Solución en Inteligencia Artificial

Andrés José Basile Álvarez

Proyecto Final Inteligencia Artificial.
Universidad Nacional Autónoma de México
Facultad de Ingeniería
andresbasile123@gmail.com
Github: https://github.com/andresbasilea/Inteligencia_Artificial.git
Diciembre, 2021



Contents

1	Requerimientos	2
2	Sobre el framework utilizado para la creación de la aplicación: Streamlit	2
3	Bibliotecas utilizadas	3
4	Metodología de desarrollo de la aplicación	5
5	Flujo de uso de la aplicación	10
6	Funcionalidad para nuevas predicciones	14
7	Posibles mejoras a futuro	15

1 Requerimientos

Se pide la implementación de una aplicación en donde se pueda demostrar el funcionamiento de ciertos algoritmos de inteligencia artificial. Se tienen que implementar los algoritmos de *métricas de distancia*, *reglas de asociación*, *clustering jerárquico y particional*, *clasificación por regresión logística* y *árboles de decisión para clasificación y pronóstico*.

La implementación consistirá en una página web donde el usuario pueda navegar y seleccionar entre los distintos algoritmos de IA a aplicar, seleccionar el conjunto de datos donde se aplicará el algoritmo y escoger cada uno de los parámetros y características propias de los algoritmos, las cuales fueron definidas al momento de estudiarlos en clase.

La forma de implementar esta solución es libre (se puede usar cualquier framework o lenguaje de programación), siempre y cuando la interfaz para el usuario final sea fácil de utilizar, interactiva y presente los resultados de la manera correcta.

Los algoritmos de *regresión logística* y *árboles de decisión para clasificación y pronóstico* deben de permitir al usuario crear predicciones a partir del ingreso de datos de un elemento de forma manual.

Se debe de permitir descargar los elementos generados por los algoritmos (árbol de decisión como imagen, reporte de validación del modelo y de las nuevas predicciones ingresadas por el usuario, modelos generados, métricas de distancia como archivo .xlsx y demás parámetros que se considere importante que el usuario pueda obtener).

2 Sobre el framework utilizado para la creación de la aplicación: Streamlit

Streamlit es una biblioteca (o framework) de Python de tipo *open-source*, la cual permite la creación de aplicaciones web para machine learning y ciencia de datos de manera sencilla y con una interfaz fácil de usar para el usuario. Para este proyecto, se optó por utilizar este tipo de framework para disminuir los tiempos en los cuales se diseñan y programan las interfaces gráficas.

Como tal, los principales elementos de este framework que utilizamos tuvieron que ver con elementos de selección (*selection boxes*, menús de tipo *dropdown*, ...), botones de descarga y áreas para el ingreso de datos. Una de las bondades de este framework es que permite la impresión rápida de elementos como *DataFrames*, *plots*, entre otros, utilizando la función *write*.

Para ejecutar una aplicación web utilizando streamlit, es tan sencillo como ejecutar el comando *streamlit run scriptName.py*, el cual abrirá una pestaña en el navegador predeterminado del programador y desplegará la aplicación en una dirección de *localhost*.

Otra de las ventajas del framework es que permite trabajar con un flujo de desarrollo similar a metodologías ágiles como *Scrum*, en donde las pruebas al programa se realizan conforme se vaya terminando cada iteración de escritura de código, permitiéndonos observar los errores en el código de forma mucho más temprana que con otras metodologías de desarrollo.

Streamlit también permite la impresión de gráficas con métodos propios del framework. Sin embargo, en el proyecto se utilizaron los métodos de creación de tablas y gráficas existentes en otras bibliotecas de Python (como *matplotlib*).

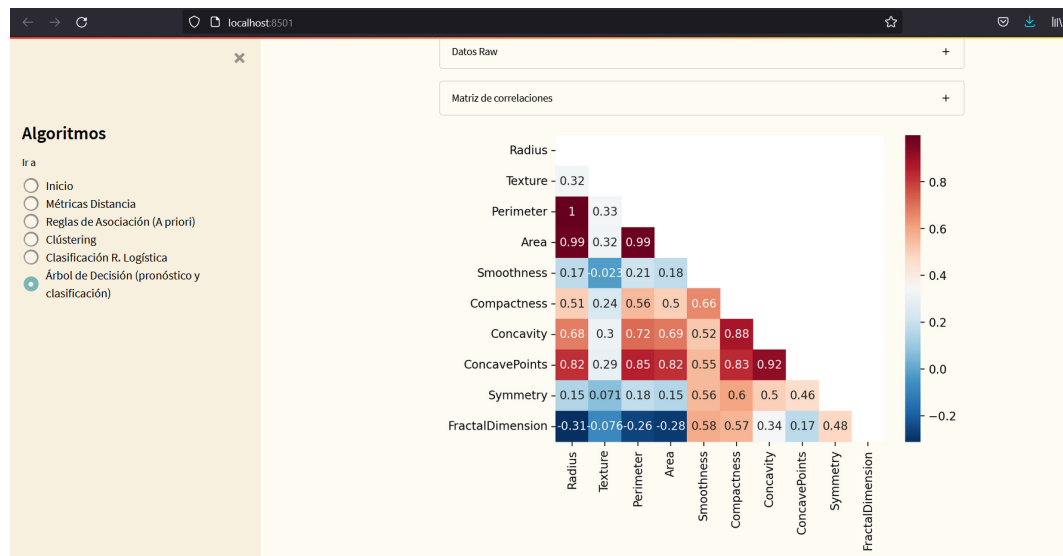
El framework también permite el diseño de interfaces temáticas personalizadas, de las cuales se creó una interfaz sencilla con colores claros que permiten que el usuario alcance a leer los datos de forma sencilla. Además, se puede mejorar la experiencia del usuario mediante el uso de la memoria cache del navegador, guardando algunos elementos durante la sesión en la que el usuario se encuentre utilizando la aplicación (por ejemplo que se guarde el conjunto de datos cargado o los resultados de aplicar uno de los algoritmos).

3 Bibliotecas utilizadas

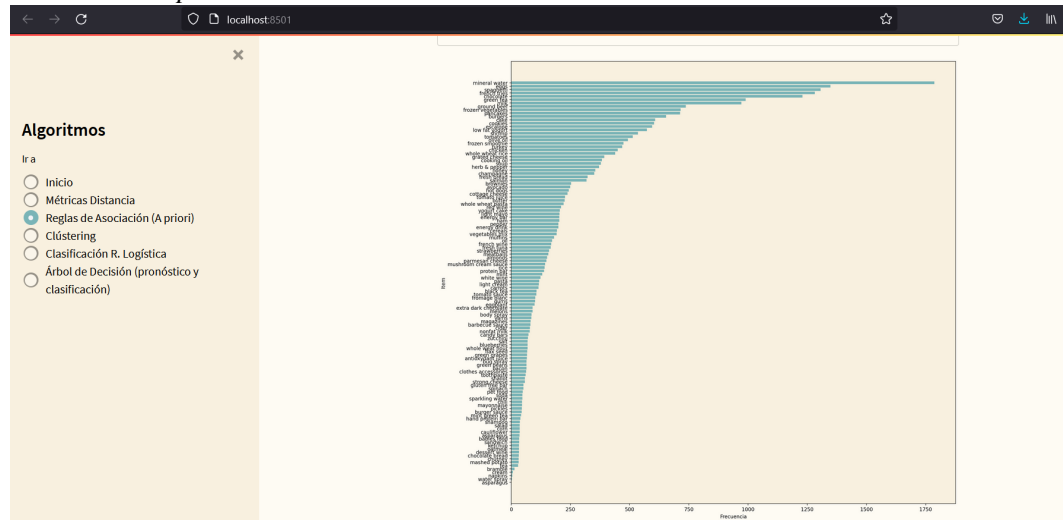
Para el desarrollo de la aplicación, no solamente se utilizaron las funciones propias del framework de desarrollo, sino que también se utilizaron métodos de varias bibliotecas de Python para auxiliarnos en el desarrollo del software. Las bibliotecas más destacadas son: *numpy* (para la operación con matrices y vectores de alta dimensionalidad), *pandas* (para el uso de *DataFrames* en donde trabajaremos con los datos), *matplotlib* para la impresión de gráficas de todo tipo, *sklearn* para implementar algunos algoritmos de machine learning y *seaborn* para la impresión de gráficas de correlación entre los datos.

Ejemplos del uso de estas bibliotecas en nuestra aplicación se muestran a continuación:

Matriz de correlación entre los datos utilizando *seaborn*:



Impresión del histograma con la frecuencia de aparición de ciertos elementos en los datos utilizando *matplotlib*:



4 Metodología de desarrollo de la aplicación

Antes del comienzo del desarrollo de esta aplicación, se tuvo en mente aplicar una metodología de tipo "ágil" en la escritura del código fuente, dividiendo el programa en unidades independientes desarrolladas por separado y en las cuales se pueden hacer pruebas de forma paralela a las iteraciones de escritura de cada uno de los bloques de código.

En el desarrollo, se consideraron los siguientes bloques independientes: *Reglas de asociación, clústering, clasificación por regresión logística y árboles de decisión*, así como una pequeña interfaz de inicio para dar la bienvenida al usuario. En la aplicación, se utilizó una barra lateral para permitirle al usuario seleccionar entre cada uno de los módulos.

La aplicación funciona en una especie de "loop infinito" donde se espera el input del usuario para seleccionar qué bloque desea utilizar. Cuando selecciona uno de los bloques, únicamente se llama a la función correspondiente a la ejecución de todas las características del bloque y se mantiene en loop ese bloque hasta que el usuario seleccione otro. De esta forma, se consigue que cada uno de los bloques se actualice constantemente dependiendo de lo que el usuario ingrese o desee realizar.

El desarrollo de la aplicación se llevó a cabo a lo largo del semestre, conforme se realizaron los algoritmos en clase. Todos los bloques comparten elementos en común como el elemento que permite la subida de un archivo ".csv" o ".xlsx" o el elemento que permite la impresión de los datos "raw", o sea tal cual como se encuentran en el archivo. Para estos elementos en común, primeramente se definieron en uno de los bloques, se hicieron las pruebas unitarias correspondientes (comprobar que el archivo subido sea el que el usuario eligió, comprobar que sólo se puedan subir archivos tipo ".csv" o ".xlsx", comprobar que la impresión de los datos sea correcta, ...) para después integrar estos elementos en todos los demás bloques. A continuación, mostramos la división de la aplicación en los distintos bloques y el acceso a cada uno de ellos desde la barra de navegación lateral.

←

→

↻

🔍

📄

localhost:8501

☆

🔖

🔗

🔍

🔧

🔴

☰

✕

Algoritmos

Ir a

☒ Inicio
 ☐ Métricas Distancia
 ☐ Reglas de Asociación (A priori)
 ☐ Clústering
 ☐ Clasificación R. Logística
 ☐ Árbol de Decisión (pronóstico y clasificación)

IntelliShow

Solución en Inteligencia Artificial

Realizado por Andrés Basile Álvarez



La presente aplicación tiene por objetivo presentar algunos algoritmos de inteligencia artificial. Acceda a la barra lateral dando click a la flecha en la parte superior izquierda para escoger un algoritmo de IA.

←

→

↻

🔍

📄

localhost:8501

☆

🔖

🔗

🔍

🔧

🔴

☰

✕

Algoritmos

Ir a

☐ Inicio
 ☐ Métricas Distancia
 ☐ Reglas de Asociación (A priori)
 ☒ Clústering
 ☐ Clasificación R. Logística
 ☐ Árbol de Decisión (pronóstico y clasificación)

Clústering

Ingrese el archivo deseado

📁

Drag and drop file here

Limit 200MB per file

Browse files

📄

Hipoteca.csv 7.8KB

✕

Usted seleccionó el archivo *Hipoteca.csv* de tipo *application/vnd.ms-excel*

Datos Raw

+

ingresos						
gastos_comunes	0.56					
pago_coche	-0.11	-0.054				
gastos_otros	-0.12	-0.1	0.011			
ahorros	0.71	0.21	-0.19	-0.064		
viviendas	0.61	0.2	-0.085	-0.055	0.61	

6

A continuación, presentamos la matriz de pruebas, donde se comprobó la funcionalidad de los elementos dentro de cada uno de los bloques:

Nombre del proyecto: IntelliShow

Materia: Inteligencia Artificial

Desarrollador: Andrés José Basile Álvarez

Matriz de pruebas

Tipo de prueba	Descripción	Fechas				Actor	Resultado
		10/12/2021	11/12/2021	12/12/2021	13/12/2021	Andrés Basile Álvarez	
Unitaria	Módulo de inicio: comprobar funcionalidad de barra lateral	x	x			x	
Unitaria	Comprobar funcionalidad del elemento general de subida de archivos.	x	x			x	
Unitaria	Comprobar funcionalidad del elemento general de impresión de archivos "raw"	x	x			x	
Unitaria	Bloque métricas de distancia: Comprobar botones radiales para cambio de métrica	x	x			x	
Unitaria	Bloque métricas de distancia: Comprobar impresión de distancias.	x	x			x	
Unitaria	Bloque reglas de asociación: Comprobar impresión de lista de frecuencias.	x	x			x	

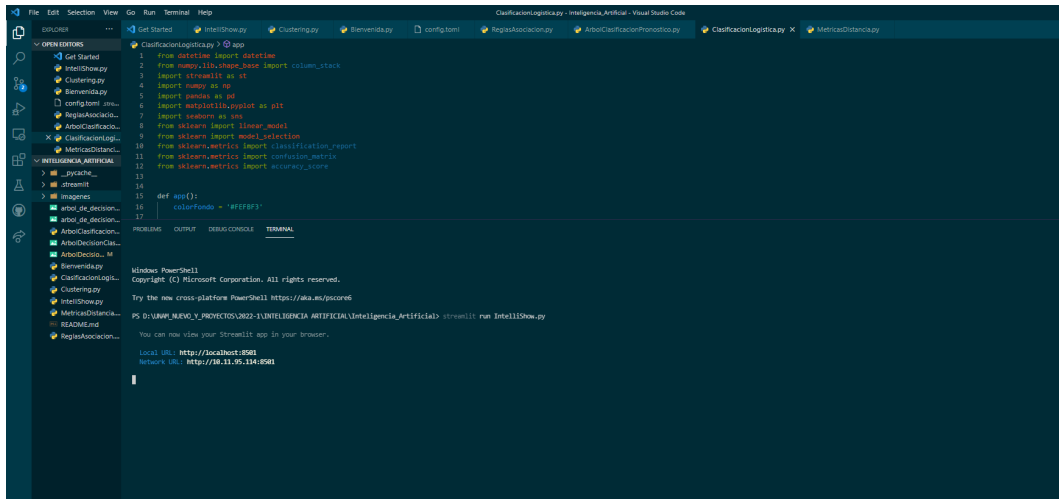
Unitaria	Bloque reglas de asociación: Comprobar funcionalidad de input para "soporte", "confianza" y "elevación".	x	x			x	
Unitaria	Bloque reglas de asociación: Comprobar impresión de reglas de asociación	x	x			x	
Unitaria	Bloque reglas de asociación: Comprobar ingreso de observaciones en cada regla y descarga en archivo.	x	x			x	
Unitaria	Bloque clústering: Comprobar impresión de matriz estandarizada.			x	x	x	
Unitaria	Bloque clústering: Comprobar selección de tipo de clústering.			x	x	x	
Unitaria	Bloque clústering: Comprobar input de número de clústers.			x	x	x	
Unitaria	Bloque clústering: Comprobar impresión de centroides de cada clúster e input de observaciones de cada cluster, así como la descarga de las observaciones.			x	x	x	
Unitaria	Bloque regresión logística: Comprobar impresión de matriz de correlaciones.			x	x	x	
Unitaria	Bloque regresión logística: Comprobar funcionamiento de bloque de selección de características.			x	x	x	

Unitaria	Bloque regresión logística: Comprobar funcionamiento de bloque de selección de variable clase.			x	x	x	
Unitaria	Bloque regresión logística: Comprobar ingreso de variable test_size y sample_input.			x	x	x	
Unitaria	Bloque regresión logística: Comprobar descarga de las nuevas predicciones.			x	x	x	
Unitaria	Bloque árboles: Comprobar selección de tipo de árbol a utilizar			x	x	x	
Unitaria	Bloque árboles: Comprobar impresión de árbol de decisión.			x	x	x	
Unitaria	Bloque árboles: Comprobar impresión de esquema del árbol de decisión.			x	x	x	
Unitaria	Bloque árboles: Comprobar input de nuevas predicciones e impresión en nuevo archivo.			x	x	x	
Unitaria	Bloque árboles: Comprobar descarga de árbol de decisión como imagen y de informe del modelo generado en .txt.			x	x	x	

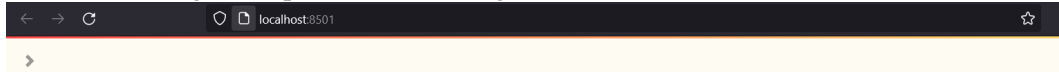
Cabe mencionar que estas pruebas se validaron posteriormente a la finalización de todo el código. No obstante, se realizaron las mismas pruebas conforme se terminaba una iteración de codificación de cada uno de los bloques antes de continuar con los siguientes elementos de un bloque.

5 Flujo de uso de la aplicación

La aplicación es realmente intuitiva y fácil de utilizar por aquel que así lo desee. Como la aplicación se encuentra en una fase previa a producción (algo así como en *User Acceptance Testing (UAT)*), aún no se cuenta con una ubicación web para acceder a la aplicación desde cualquier computadora con acceso a la red. Por ello, se debe de ejecutar la siguiente línea de código desde una terminal, ubicándonos en la carpeta donde está el código fuente:



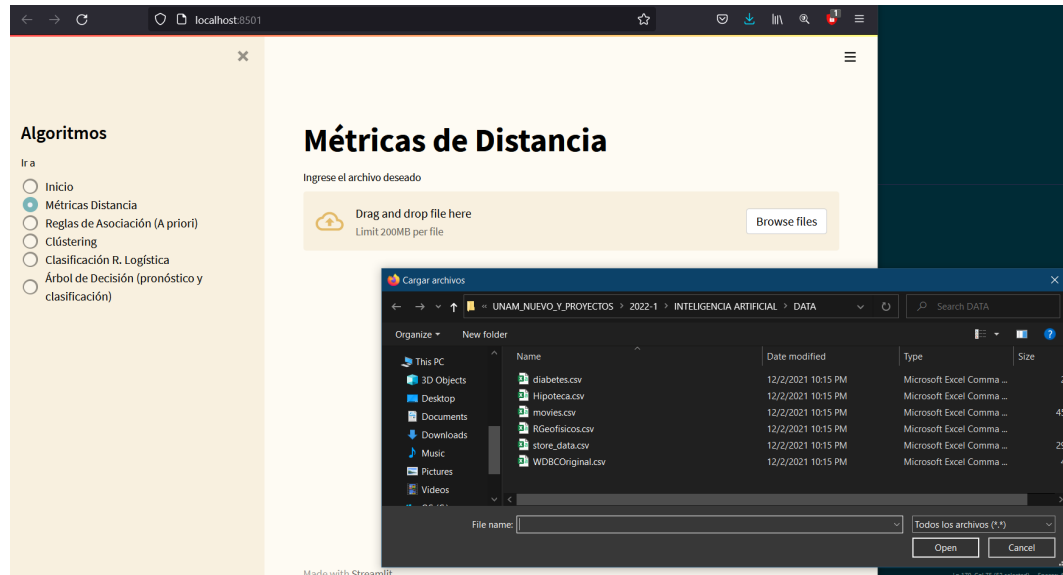
obteniendo la siguiente pestaña en el navegador:



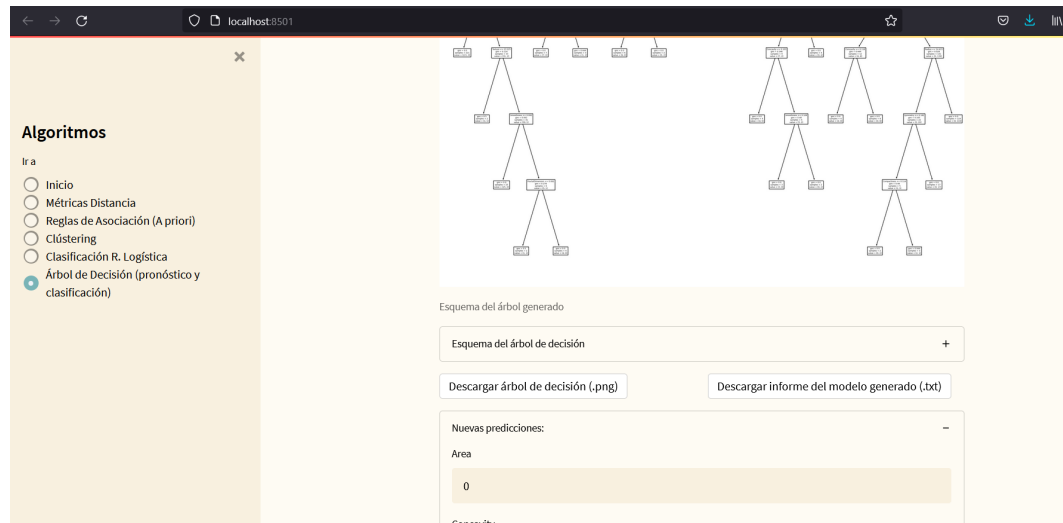
Una vez abierto, se observará la pestaña de inicio (presentada anteriormente en una imagen). En ella, el usuario seleccionará del menú de desplazamiento lateral, el algoritmo que desea utilizar. Una vez seleccionado, todos ellos pedirán la subida de un archivo ".csv" o ".xlsx" para trabajar con él. Una vez subido el archivo en un módulo, automáticamente se cargará en todos los demás.

Hecho lo anterior, se seguirán los procedimientos para el desarrollo de cada uno de los algoritmos. La aplicación irá solicitando los datos conforme sean requeridos y, una vez ingresados, automáticamente se actualizará la aplicación para presentar los últimos resultados. Algunas secciones de los bloques permiten hacer una descarga de un documento donde se muestra un informe sobre lo realizado en el algoritmo o los árboles, gráficas o matrices generados.

A continuación, se muestra el explorador de archivos abierto para hacer la carga de un nuevo archivo en la aplicación. Como se muestra en la imagen, únicamente se despliegan archivos de tipo ".csv" o ".xlsx":



Vemos también los botones para descargar algunos elementos importantes en cada uno de los algoritmos:



Algunos ejemplos de reportes generados (la primera imagen muestra el reporte para los árboles de decisión y la segunda imagen para las reglas de asociación):

```
*InformeArbolClasificación-8.txt - Notepad
File Edit Format View Help

Fecha de generación: 2021-12-10 14:30:31.478175

Esquema generado de: WDBCOriginal.csv
Árbol de decisión: Clasificación
Variables Predictoras:
Area, Concavity, Radius, Compactness, FractalDimension, Smoothness, Symmetry, Perimeter, Texture, ConcavePoints,
Variable Clase:
Diagnosis
|

|-- ConcavePoints <= 0.05
| | |-- Area <= 696.25
| | | |-- ConcavePoints <= 0.05
| | | | |-- Texture <= 22.45
| | | | | |-- class: B
| | | | |-- Texture > 22.45
| | | | | |-- Texture <= 23.23
| | | | | | |-- class: B
| | | | | | |-- Texture > 23.23
| | | | | | |-- Smoothness <= 0.09
| | | | | | | |-- class: B
| | | | | | | |-- Smoothness > 0.09
| | | | | | | |-- FractalDimension <= 0.06
| | | | | | | | |-- class: B
| | | | | | | |-- FractalDimension > 0.06
| | | | | | | | |-- class: B
| | | |-- ConcavePoints > 0.05
| | | | |-- Concavity <= 0.10
| | | | | |-- class: B
| | | | |-- Concavity > 0.10
| | | | | |-- class: M
| |-- Area > 696.25
| | |-- Texture <= 20.74
| | | |-- FractalDimension <= 0.06
```

```
ReglasAsociacion-8.txt - Notepad
File Edit Format View Help
Fecha de generación 2021-12-13 14:44:14.375913

Reglas de asociación generadas para el conjunto: store_data.csv

__Regla__ 1:['burgers', 'eggs']
Soporte: 0.0288
Confianza: 0.33028,
Elevación: 1.83758
Observación sobre la regla: Aquí va la observación 1

__Regla__ 2:['burgers', 'french fries']
Soporte: 0.022
Confianza: 0.25229,
Elevación: 1.47598
Observación sobre la regla: Aquí va la observación 2

__Regla__ 3:['spaghetti', 'burgers']
Soporte: 0.02147
Confianza: 0.24618,
Elevación: 1.41373
Observación sobre la regla: Hola desde la observación 3

__Regla__ 4:['cake', 'mineral water']
Soporte: 0.02747
Confianza: 0.33882,
Elevación: 1.422
Observación sobre la regla: Hola desde la observación 4

__Regla__ 5:['chicken', 'mineral water']
Soporte: 0.0228
```

6 Funcionalidad para nuevas predicciones

Dos de los bloques presentados en la aplicación cuentan con la capacidad para aceptar nuevas predicciones con datos que no necesariamente están en el conjunto de datos. Estos bloques son: Clasificación por regresión logística y árboles de decisión para clasificación y pronóstico. Esta funcionalidad es particularmente útil cuando necesitamos analizar un elemento nuevo que cuenta con los mismos atributos que los elementos del conjunto de datos. Ejemplos de lo anterior podrían ser datos de tumores (donde el doctor ingresará datos de un nuevo tumor [área, perímetro, textura, etc.] para obtener una clasificación de si éste es benigno o maligno, lo cual podría funcionar como un segundo diagnóstico aparte de el del médico) o la clasificación de si un nuevo cliente es apto para un préstamo a partir de datos como sus ingresos o sus deudas.

Esta funcionalidad permitirá a un usuario técnico ingresar valores cuando así lo necesite, además de permitirle descargar dichos resultados en un resultado que contiene los valores ingresados y la predicción (clasificación o pronóstico) como se muestra a continuación:

Nuevas predicciones:

Area

0

Concavity

0

Radius

0

Compactness

0

FractalDimension

0

Smoothness

0

Symmetry

Symmetry

0

Perimeter

0

Texture

0

ConcavePoints

0

B

Descargar nueva predicción

7 Posibles mejoras a futuro

Actualmente, una de las principales debilidades de la implementación es que no se encuentra en una página web accesible desde cualquier dispositivo, sino que se encuentra en un *local-host* accesible únicamente desde donde se esté ejecutando el código. A futuro, se podría establecer un servidor dedicado para ejecutar la aplicación y configurar un dominio web en donde cualquier persona pueda acceder.

Otra mejora podría ser insertar alguna especie de editor para hojas de cálculo, donde se puedan modificar ampliamente los conjuntos de datos antes de aplicar un algoritmo y así obtener los resultados esperados.

Además, se podría modificar la generación de reportes de resultados de los algoritmos para que éstos tengan un formato más legible y con un mejor diseño general. Por ejemplo, se podrían generar las gráficas, tablas y resultados en un archivo presentado como pdf al usuario final.

En general, se cumplieron los objetivos de diseño e implementación de esta aplicación, logrando las funcionalidades requeridas en una aplicación sencilla de utilizar y visualmente agradable.