

Algorithms: Una herramienta de aprendizaje de IA

Rogelio Yael Hernandez Jaimes

No. Cuenta 317276839

rogerlibra01@gmail.com

Facultad de Ingeniería, UNAM

1. Introducción

La Inteligencia Artificial (IA) es una herramienta cuyo uso tanto en la actualidad como en el futuro es y será de suma importancia. A lo largo de los años, esta herramienta ha ido evolucionando y aumentando sus capacidades. La búsqueda de una IA general, es quizás uno de los objetivos más ambiciosos, así como uno de los más difíciles de conseguir. El desarrollo de esta es directamente responsable de las personas que se involucran en su investigación, por lo que los conocimientos de estos requieren de una buena variedad de conocimientos. Ahora bien, para cualquiera de los avances a futuros, se requerirá conocer las bases de la IA, entre tantas los algoritmos de aprendizaje. Con el fin de facilitar el entendimiento de algunos de estos algoritmos, se presenta la aplicación *Algorithms*. El fin de esta aplicación es permitir al usuario poner a prueba algunos de los algoritmos de aprendizaje utilizados en la actualidad para el desarrollo de la IA. Dando algunas observaciones respecto a la aplicación de algunos de estos, como puntos importantes, consideraciones para su implementación, así como muestras gráficas de los datos que se analizan.

El presente trabajo, tratará de una forma un tanto general la implementación de dicha aplicación. Refiriéndose sobre algunas observaciones necesarias en su desarrollo, tales como arquitectura de aplicación, requerimientos, tecnologías utilizadas, observaciones particulares, y algunos otros puntos a utilizar.

2. Requerimientos

Algorithms es una aplicación de aprendizaje, su por lo que su enfoque no solo se centra en la aplicación de algoritmos, sino también en la introducción teórica para cada una, además de observaciones puntuales para su aplicación. Por ejemplo:

- **Características de la colección de datos:** Se busca indicar al usuario los requerimientos básicos de la colección de datos para la aplicación del algoritmo (para la programación) o la forma general de la colección. Por ejemplo: omisión de los encabezados (en la colección), cada transacción en una línea nueva (indicando el factor clave de la transacción, ej. número de usuario), etc.
- **Distribución gráfica de los datos:** Se muestra la distribución de los datos de forma gráfica, esto para conocer de forma general cómo se encuentra y llegar a conclusiones más lógicas.
- **Reducción dimensional:** Consideraciones particulares a seguir para la reducción dimensional de los datos.
- **Normalización y estandarización de datos:** Se recalcará la necesidad de estandarizar o normalizar los datos para diferentes algoritmos.
- **Dispersión de datos:** Gráfica respecto a la dispersión de los datos.

- **Obserbaciones puntuales sobre los algoritmos aplicados:** Algunas consideraciones respecto al costo computacional de los algoritmos, su eficiencia y eficacia y otras consideraciones al seleccionar el algoritmo.

Así pues, respecto a las características funcionales del programa, se tienen algunos puntos a seguir:

- Uso de archivos .csv para operar.
- La colección original de archivos no sufre modificaciones para seguridad.
- Interfaz gráfica de la aplicación: El usuario podrá navegar entre los apartados utilizando un menú de selección desplegable entre los algoritmos.
- Menú principal como punto de partida: El usuario puede ingresar a los diferentes algoritmos a través del index principal de la página.
- Carga de archivos propios del usuario: El usuario podrá cargar su propia colección de datos, indicándole requerimientos para la operación sobre la misma.
- La aplicación muestra los resultados de forma gráfica (utilizando gráficas y elementos para clarificar los resultados del algoritmo).
- La aplicación de los algoritmos no mostrará cada uno de los pasos realizados, sino que se apegará a mostrar resultados y algunos de los puntos importantes para el algoritmo. Adicional a esto, se añadirán las notas mencionadas previamente.
- Respecto a la aplicación de los algoritmos, se mostrarán los resultados adecuados para cada uno (reglas de asociación, matriz de distancia, etcétera). Además, gráficos que ayuden a mostrar los resultados obtenidos de su aplicación.
- El usuario puede indicar qué valores utilizará para los distintos algoritmos. Así mismo, para la reducción de dimensionalidad, el usuario puede escoger libremente las variables a eliminar.
- En la presente aplicación, la página no almacena ninguna colección de datos. Toda información ingresada por el usuario se almacena únicamente de forma temporal, y se elimina al tratar de resimular el algoritmo, o ejecutar cualquier otro.
- La página no contiene ninguna forma de login. La aplicación tiene fines educativos, no profesionales, por lo que no se estima la recaudación de datos o alguna otra información.
- La aplicación no figura la presencia de usuarios base ni superusuarios. La aplicación solo moldea el uso ligero de la aplicación. El usuario no es capaz de incidir directamente sobre el algoritmo, sino que se le permite ingresar algunos determinados parámetros. En el resto de casos, el programa realiza de forma automática la ejecución del algoritmo. Por tanto, no existe riesgo de sistema al operar los algoritmos.
- El usuario es libre de descargar en cualquier momento la página resultante en formato html.

En el producto final, la aplicación incorpora todas las características aquí listadas. Cada algoritmo contiene una introducción teórica para su aplicación. Tales como las aplicaciones para dicho algoritmo, formas de evaluar su exactitud y otros factores.

3. Tecnologías utilizadas

3.1. Arquitectura de aplicación

Para el desarrollo de Algorithms nos basamos en una arquitectura cliente-servidor, "la cual es un modelo de una aplicación distribuida en el cual se basa en dos actores: Uno con rol de proveedor de recursos y otro con rol consultor sobre los recursos".[?] Esta la estructura base. Sin embargo, dada la tecnología de Django, se utiliza internamente una arquitectura vista-controlador (sin modelo pues no se utiliza base de datos).

3.2. Lenguaje de programación

Para la implementación de esta aplicación (backend) se hará uso del lenguaje Python junto a sus librerías *apryori*, *pandas*, *numpy*, *pyplot*, *scipy.spatial*, *sklearn*, *kneed*, *seaborn* [?][?][?]. De esta forma, los algoritmos de aprendizaje se encuentran previamente implementados en las funciones de las bibliotecas listadas previamente. La mayor parte de estas fueron utilizadas a lo largo de las prácticas realizadas durante el semestre en curso.

3.3. Interfaz Gráfica (html,css)

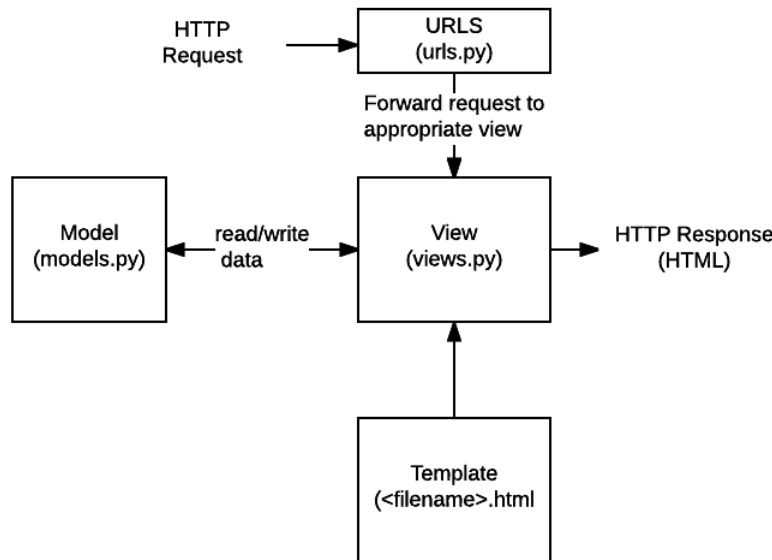
A diferencias de entregas previas, la aplicación fue desarrollada en formato de página web con HTML 5 y CSS. Ambas integradas dentro de Django, herramienta con la cual comunicamos el front-end con el back-end.

En lo que respecta a HTML y CSS, se utilizarán como base para realizar toda lo gráfico mostrado al usuario. HTML es un lenguaje de etiquetas que permite definir la estructura de un archivo que un navegador web interpretará para generar un resultado visible. CSS es un lenguaje de diseño gráfico en cascada (Cascading Style Sheets, CSS) que permite dar una estructura "decorativa." a un lenguaje de etiquetas. De esta forma, CSS permite dar un estilo más visual a la estructura mostrada con HTML, permitiendo así elementos más atractivos para la vista. Además de estos dos, en una proporción pequeña, se utiliza JavaScript, un lenguaje de programación interpretado, esto para mostrar de forma automática algunos elementos resultantes de los algoritmos. [?][?]

3.4. Django

Django es un framework de Python de alto nivel que permite la producción de páginas web con python como base de su backend. Django es un framework publicada en 2005, y desde entonces ha tenido un crecimiento continuo hasta volverse el framework de web genérico.

Django trabaja siguiendo la arquitectura Modelo-Vista-Controlador. La estructura general del framework es la siguiente:



- **URL's:** Django utiliza funciones de visualización separada para cada URL. Utiliza un mapeador URL para procesar las peticiones HTTP a la vista adecuada en la URL de la petición.
- **Vista (view):** Recibe peticiones HTTP y devuelve una respuesta. En las vistas se acceden a los datos necesarios para completar la petición. Es gracias a esta característica que podemos interconectar el frontend con el backend. Cuando hacemos un "salto.^a una URL determinada, Django la procesa a partir de la URL.py para pasar primero por la vista destinada a su procesamiento (definida por el usuario. De esta forma, cuando el usuario final quiera ejecutar algún algoritmo, podemos calcularlo antes de enviar una respuesta gráfica. [?]
- **Modelos:** Estructuras para bases de datos. Pueden ser accedidas y procesadas en las vistas.
- **Templates:** Ficheros de texto que definen la estructura de un fichero. Aquí conectamos HTML. Como se mencionó con anterioridad, omitimos el uso de modelos, por lo que la arquitectura de Django en esta aplicación se restringe a la vista-controlador.

4. Algoritmos de aprendizaje

La presente aplicación realiza la implementación de los siguientes algoritmos de aprendizajes:

- **Apriori:** El algoritmo apriori obtiene las reglas de asociación a partir de un conjunto de transacciones dadas. Para nuestro caso de estudio, dicha colección muestra cada transacción en la misma fila y su contenido es puramente los elementos a evaluar. Para nuestra aplicación, dada su naturaleza de herramienta para el aprendizaje, se incluye una introducción teórica:

Algoritmo a priori:

Este es el primer algoritmo base de los algoritmos de aprendizaje. Se trata de un algoritmo de aprendizaje no supervisado (unsupervised). Un buen ejemplo de la aplicación de este algoritmo son los sistemas de recomendación. Estos sistemas se encuentran en diferentes aplicaciones como Amazon, Youtube, Netflix, entre otras; con sus típicos "videos relacionados", "Películas que podrían gustarte", "Artículos similares", entre otros. Este algoritmo trabaja utilizando las **reglas de asociación**

Reglas de asociación

En lo que refiere a su ejecución, primeramente se solicita al usuario los parámetros a trabajar (elevación, confianza y soporte), además de cargar la colección de datos de extensión .csv con la descripción dada en su nota:

Ingrese los valores

Elevación(Lift):

Confianza(Confidence):

Soporte(Support):

Seleccionar archivo

NOTA: El archivo a seleccionar es de extensión '.csv' para operar. Así mismo, para este algoritmo, se requiere el uso de una colección de datos **SIN CABECERA**. Cada fila cuenta como un registro de la colección, donde cada registro lista en su fila los elementos correspondientes a este. Por ejemplo, en una base de datos de películas, cada registro (fila) lista las películas observadas por una cuenta. De hacer caso omiso a esta estructura el programa operará con errores y los resultados podrían mostrarse erróneos o generar excepciones.

En los resultados el algoritmo muestra la distribución gráfica de los elementos en las transacciones y el punto más importante: las reglas de asociación generadas.

Nº	Item Base	Consecuente	Support
1	frozenset({'Wonder Woman'})	frozenset({'Coco'})	0.005093833780160858
2	frozenset({'Red Sparrow'})	frozenset({'Green Lantern'})	0.005764075067024129
3	frozenset({'Star Wars'})	frozenset({'Green Lantern'})	0.005898123324396783
4	frozenset({'Kung Fu Panda'})	frozenset({'Jumanji'})	0.0160857908847185
5	frozenset({'Jumanji'})	frozenset({'Tomb Rider'})	0.03941018766756032
6	frozenset({'Wonder Woman'})	frozenset({'Jumanji'})	0.005361930294906166
7	frozenset({'The Spy Who Dumped Me'})	frozenset({'Moana'})	0.009919571045576408
8	frozenset({'Thor'})	frozenset({'Moana'})	0.015281501340482574
9	frozenset({'Sicario'})	frozenset({'Ninja Turtles'})	0.005093833780160858
10	frozenset({'Star Wars'})	frozenset({'The Revenant'})	0.005093833780160858

Metricas de Distancia: Si bien no son un algoritmo de aprendizaje, son base funcamental para otros. La colección de datos utilizados poseen N variables cuantificadas. Dependiendo del criterio de distancia (Euclidean, Chevyshev, Mintowski o Manhattan) se determina la distancia que separa a dos elementos. En esta aplicación, además de la introducción teórica previa, se solicita al usuario la siguiente información:

Ingrese los valores

Por favor, indique el tipo de medición a obtener:

- ☐ Distancia Euclidean (euclidean)
- ☐ Distancia de Chebyshev
- ☐ Distancia de Manhattan (cityblock)
- ☐ Distancia de Minkowski

Respecto a la colección de datos, indique el tipo de procesamiento:

- ☐ Datos Estandarizados
- ☐ Datos Normalizados
- ☐ Datos Originales

Cantidad de decimales:

Lambda (solo para Minkowski, en caso contrario, ingrese cualquier valor):

Seleccionar archivo Ninguno archivo selec.

Enviar

NOTA: El archivo a seleccionar es de extensión '.csv' para operar. Así mismo, para este algoritmo, se requiere el uso de una colección de datos **CON CABECERA**. Cada fila cuenta con valores para cada una de las variables indicadas en la cabecera **SIN VALORES NULOS**. De hacer caso omiso, el programa podría mostrar resultados inesperados o terminarse abruptamente.

El usuario escoge libremente la medición a realizar, así como el tipo de procesamiento de datos a ejecutar (normalización, estandarización o utilizar los datos originales), esto para comprobar la utilidad de normalizar o estandarizar los datos. Así mismo, el usuario puede cargar el archivo con las características indicadas por la nota.

En los resultados obtenidos se muestra la colección de datos procesada u original. Además, se muestra la matriz de distancia resultante:

Matriz de distancia (chebyshev)

Sobre esta colección de datos aplicamos las medición de chebyshev y redondeamos a 3 decimales. El resultado de obtener las métricas de distancia es una matriz cuadrada de nxn, donde las filas y columnas corresponden a la transacción indicada. De esta forma, el elemento (1,2) estaría comparando la transacción 1 con la transacción 2. Esta es una matriz simétrica, donde la diagonal compara a la transacción consigo misma, de ahí que la distancia sea 0. Así pues recordemos que la distancia de un punto a otro es simétrica, por lo que es igual $d(A,B)$ que $d(B,A)$, de ahí que la matriz sea también simétrica. Las diferentes formas de medición muestran un valor propio de sí. En ocasiones, la euclidean es menor que la chebyshev y viceversa. Así mismo, existe una diferencia cuando los datos están estandarizados, normalizados o ninguno. Se invita al usuario a experimentar libremente para comprobarlo.

	0	1	2	3	4	5	6	7	8	9	10	11	
0	0.0	2.124	2.395	2.124	1.789	1.64	2.768	2.124	1.677	2.395	2.431	2.395	2
1	2.124	0.0	2.316	2.199	2.199	2.124	2.124	2.199	2.543	2.299	2.372	2.199	2
2	2.395	2.316	0.0	3.226	2.395	2.803	2.898	2.767	2.395	3.28	3.542	2.372	3
3	2.124	2.199	3.226	0.0	2.294	2.932	2.648	3.149	2.719	2.476	2.311	2.395	2
4	1.789	2.199	2.395	2.294	0.0	2.932	1.603	2.124	1.426	2.395	2.181	2.395	2
5	1.64	2.124	2.803	2.932	2.932	0.0	2.68	2.932	2.932	2.199	2.932	2.932	1
6	2.768	2.124	2.898	2.648	1.603	2.68	0.0	2.821	1.562	1.197	2.124	2.174	1
7	2.124	2.199	2.767	3.149	2.124	2.932	2.821	0.0	2.318	3.202	3.465	2.395	3

No se profundiza demasiado en los resultados dado a que en métricas de distancia no hay demasiada información generada por sí mismas.

Clustering: El clústering consiste en agrupar a los elementos de un conjunto de datos de acuerdo a su similitud con otros. En el clustering jerárquico, la agrupación se realiza respecto a dos elementos más cercanos, mientras que en el particional se definen centroides iniciales y se agrupan los elementos de acuerdo a su cercanía con estos. Para cada algoritmo modelamos de forma similar los resultados. Para ambos clusters pedimos como información inicial:

Paso 1: Ingrese los valores

Por favor, indique el tipo de medición a obtener:

- ☐ Distancia Euclídeana(euclidean)
- ☐ Distancia de Chebyshev
- ☐ Distancia de Manhattan (cityblock)

Respecto a la colección de datos, indique el tipo de procesamiento:

- ☐ Datos Estandarizados
- ☐ Datos Normalizados

Respecto a la colección de datos, indique el tipo de procesamiento:

- ☐ Clustering Jerárquico
- ☐ Clustering Particional

Seleccionar archivo Ninguno archivo selec.

Enviar

NOTA: El archivo a seleccionar es de extensión '.csv' para operar. Así mismo, para este algoritmo, se requiere el uso de una colección de datos **CON CABECERA**. Cada fila cuenta con valores para cada una de las variables indicadas en la cabecera, **sin valores nulos**. Así mismo, los valores deben ser cuantificados, en caso de tener valores tipo cadena, se insta al usuario a cuantificarlos. De hacer caso omiso, el programa podría mostrar resultados inesperados o terminarse abruptamente.

Seguido a esto, ambos tipos de clustering pasan por el proceso de selección de características para la reducción de dimensionalidad. En este paso se listan todas las variables y se podrán seleccionar aquellas que se deseen eliminar.

Paso 2: Selección de características

Por favor, indique las variables a descartar:

ingresos <input type="checkbox"/>	gastos_comunes <input type="checkbox"/>	pago_coche <input type="checkbox"/>	gastos_otros <input type="checkbox"/>	ahorros <input type="checkbox"/>
vivienda <input type="checkbox"/>	estado_civil <input type="checkbox"/>	hijos <input type="checkbox"/>	trabajo <input type="checkbox"/>	comprar <input checked="" type="checkbox"/>

Enviar


Tras esto, el clustering jerárquico pasa directo a mostrar resultados. Por su parte, el clustering particional requiere definir el valor de K, para lo cual utilizamos el método del Elbow, que al usuario se presenta de la siguiente forma:

Paso 3: Selección de K-Means

Número de clústers (K):

Enviar

Para la selección de un valor adecuado de K para trabajar el algoritmo se emplea el Elbow método. En la gráfica buscamos el punto de inflexión de la curva. Sin embargo, dadas las ambigüedades de este, indicamos el valor obtenido de forma automática. Es decisión del usuario emplearlo o probar con uno diferente.



Además de la gráfica, se muestra el valor en número estimado de K. El usuario puede escoger libremente el valor de K, aun si no es el estimado.

En cuanto a los resultados. El clustering jerárquico muestra el árbol generado e su desarrollo:



Ambos algoritmos muestran además la lista de los elementos junto con su cluster al que fueron asignados. Además, muestra el total de elementos por cada cluster:

Donde los elementos se agrupan en los distintos clústers como se muestra en ClusterP de la siguiente tabla:

id_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	clusterP
0	0	600	50000	400000	0	2	2	3
123	429	43240	636897	1	3	6	2	2
98	795	57463	321779	2	1	8	2	2
15	254	54506	660933	0	0	3	3	3
223	520	41512	348932	0	0	3	3	3
11	325	50875	360863	1	4	5	2	2
345	309	46761	429812	1	1	5	3	3
39	782	57439	606291	0	0	1	3	3
209	571	50503	291010	0	0	3	3	3

Cada cluster tiene un número dado de elementos, tal como se muestra a continuación:

Cluster	clusterP
0	43
1	56
2	54
3	49

Adicional a esto, se muestran algunos resultados adicionales de menor importancia de acuerdo con cada uno.

Clasificación Logista: La clasificación logística es un tipo de clasificación ideal para la la clasificación binaria, donde una variable solo puede obtener uno de dos valores dados. Para utilizarla, primeramente se debe estimar la regresión lineal. Para su ejecución se solicita al usuario únicamente el tamaño de población para entrenamiento y la colección de datos:

Paso 1: Ingrese los valores

Porcentaje de tamaño de muestra para entrenamiento:

Ninguno archivo selec.

NOTA: El archivo a seleccionar es de extensión '.csv' para operar. Así mismo, para este algoritmo, se requiere el uso de una colección de datos **CON CABECERA**. Cada fila cuenta con valores para cada una de las variables indicadas en la cabecera, **sin valores nulos**. Así pues la variable de clase debe ser **binaria** (solo puede tener dos valores posibles) y debe ser cualitativa (dada por etiqueta). Si la variable está cuantificada, el programa aún así asignará un valor particular a los valores. De hacer caso omiso al resto de advertencias, el programa podría mostrar resultados inesperados o terminarse abruptamente.

Tras esto, la página solicitará realizar la selección de la variable de clase o a estimar:

Paso 2: Selección de la variable de clase

Por favor, indique la variable a utilizar como variable dependiente:

IDNumber ☐ Diagnosis ☒ Radius ☐ Texture ☐ Perimeter ☐ Area ☐

Smoothness ☐ Compactness ☐ Concavity ☐ ConcavePoints ☐ Symmetry ☐

FractalDimension ☐

Esta variable se elimina del listado de variables a escoger. Tras esto, se realiza la selección de características tal cual se observó en clustering, omitiendo la variable de clase. El programa eliminará las variables seleccionadas y de forma automática tomará el resto como la variables predictoras. En los resultados se indica el tipo de cuantificación realizada a la variable de clase:

Resultados obtenidos: Regresión Logística

Archivo: WDBCOriginal.csv

Variable de clase: Diagnosis (M: 0 , B: 1)

Variables predictoras: ['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension']

Entre diferentes elementos, los resultados más destacables de los resultados tenemos la probabilidad de clasificación así como la matriz de clasificación:

Así pues, ponemos a prueba el modelo con la

Elemento	0	1
0	0.2635	0.7365
1	0.3841	0.6159
2	0.1426	0.8574
3	0.2781	0.7219
4	0.1021	0.8979
5	0.0297	0.9703
6	0.1197	0.8803
7	0.0415	0.9585
8	0.0111	0.9889
9	0.0057	0.9943
10	0.2204	0.7796
11	0.6809	0.3191

Exactitud: 0.8947368421052632

Más específicamente nos apegamos al análisis de la matriz de clasificación:

Real/Clasificación	0	1
0	40	7
1	5	62

Árboles de decisión y bosques aleatorios: Los árboles de decisión utilizan nodos de decisión para dividir los elementos de acuerdo al tipo, o al rango de valores de una determinada variable. Al llegar a un nodo hoja, el elemento adquiere por tanto el valor definido en esta para la variable de clase. Los árboles aleatorios es una combinación de n árboles de decisión con configuración propia, disminuyendo la probabilidad de sobreajuste. El usuario puede ingresar los siguientes valores:

Paso 1: Ingrese los valores

Por favor, indique el algoritmo a utilizar:

☐ Arbol de Decision
 ☐ Bosque aleatorio

Respecto a la variable de clase, indique tipo:

☐ Simple (dos valores)
 ☐ Multiple (más de dos valores)

Profundidad máxima:

Elementos mínimos de división de nodo (min_samples_split):

Elementos mínimos por hoja (min_samples_leaf):

Porcentaje de tamaño de muestra para entrenamiento:

Seleccionar archivo
 Ninguno archivo selec.

Enviar

NOTA: El archivo a seleccionar es de extensión '.csv' para operar. Así mismo, para este algoritmo, se requiere el uso de una colección de datos **CON CABECERA**. Cada fila cuenta con valores para cada una de las variables indicadas en la cabecera. De hacer caso omiso, el programa podría mostrar resultados inesperados o terminarse abruptamente.

Notese que podemos escoger entre árboles de decisión y bosques aleatorios. Sumado a esto, se puede escoger si la variable de clase es de tipo binaria o múltiple. Los parámetros adicionales son utilizados para generar los árboles de decisión. En los bosques aleatorios también se aplican estos parámetros. Al igual que en la regresión logística, aquí podremos escoger la variable de clase, la selección de características. Adicionalmente para los bosques aleatorios podemos ingresar el número de árboles a utilizar.

Paso 2: Selección de la variable de clase

Por favor, indique la variable a utilizar como variable dependiente:

Pregnancies

☐

Glucose

☐

BloodPressure

☐

SkinThickness

☐

Insulin

☐

BMI

☐

DiabetesPedigreeFunction

☐

Age

☐

Outcome

☐

Número de árboles (n_estimators):

Enviar

En los resultados de estos algoritmos se encuentra la colección original de datos, la estructura general (en árboles de decisión), curva roc (variables binarias), matriz de clasificación y otros elementos básicos para su análisis.

5. Consideraciones finales

El presente proyecto realiza el desarrollo de una aplicación de aprendizaje básico de algoritmos de aprendizaje. La idea es que una persona inexperta pueda explorar entre estos algoritmos y experimentar con ellos. La idea es dar una herramienta que les permita entender el funcionamiento de los mismos sin llegar a una aplicación profesional. De ahí que no se incluya la idea de superusuarios. Para cada algoritmo implementado se provee una introducción teórica para su estudio. Así mismo, la aplicación añade notas útiles para entender la teoría y los resultados. Como se mencionó, se modificaron las tecnologías utilizadas así como varias requerimientos funcionales como no funcionales.

AI algorithms muestra la teoría abordada a lo largo del semestre. Basada en la información presentada en las diapositivas y prácticas realizadas con el profesor durante todo el semestre. La idea es dar una herramienta similar a la que se tuvo a disposición durante el semestre.

Referencias

- [1] Redes de computadores Arquitectura Cliente-Servidor (2016). Recuperado el 3 de enero del 2023, de: <http://profesores.elo.utfsm.cl/agv/elo322/1s16/projects/reports/Proyecto>
- [2] pandas (2019). User Guide. Recuperado el 24 de octubre de 2022, de: https://pandas.pydata.org/docs/user_guide/index.html
- [3] The SciPy Community (2008). Recuperado el 24 de octubre de 2022, de: <https://docs.scipy.org/doc/scipy/tutorial/index.html#user-guide>
- [4] scikit-learn (2011). Recuperado el 9 de enero de 2023, de: <https://scikit-learn.org/stable/>
- [5] CSS (2020). Recuperado el 9 de enero de 2023, de: <https://developer.mozilla.org/es/docs/Web/CSS>
- [6] JavaScript (2022). Recuperado el 9 de enero de 2023, de: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- [7] Introducción a Django (2022). Recuperado el 9 de enero de 2023, de: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>