

# Documentación Proyecto Final

## GUI de Minería de Datos: Datanalycs

Guadarrama Ortega César Alejandro

Asignatura: Minería de Datos

Profesor: Dr. Guillermo Gilberto Molero Castillo

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Semestre 2022-2

26 de mayo de 2022

## 1. Objetivo

Crear una GUI de minería de datos intuitiva para el usuario y que ofrezca una gran gama de herramientas y algoritmos para poner en práctica los análisis obtenidos.

## 2. Requerimientos

### Funcionales

- Usar el lenguaje de programación Python para una implementación más sencilla y directa.
- Uso de Streamlit, un API de Python que convierte los scripts de datos en aplicaciones web. No se requiere experiencia en front-end.
- Implementación de algoritmos y análisis de datos (EDA, APC, ACD, Clustering, Regla de asociación, Pronóstico y Clasificación).
- Soporte de datos CSV, modificación en tiempo real.
- Selección de variables a eliminar del Data Frame, si lo necesita el usuario.
- Capacidad de descargar el Data Frame modificado para el uso de otros análisis de datos.
- Interacción del usuario mediante botones, sliders, cajas de texto, etc.
- Barra lateral para carga de datos, botones y cajas de texto.
- Pagina principal, donde se mostrará toda la información.
- Robustes, señalamientos para que el usuario no introduzca datos erróneos para el análisis.
- Implementación de gráficos.

### No funcionales

- La app web debe ser intuitiva con el usuario.
- Diseño minimalista.
- Información en forma de bloques.

## 3. Desarrollo

### Nombre de la app web

El nombre de la aplicación web será “Datanalycs” este se mostrará al inicio de la página web.

## Metodología para el desarrollo

Para la creación de esta app web se está implementando el desarrollo en cascada. Cada uno de los algoritmos o análisis profundos de datos son una meta que se debe cumplir. La creación de un nuevo módulo no afecta directamente al anterior.

Las fases de la metodología en cascada son las siguientes:

- 1. Análisis de requisitos.
- 2. Diseño del sistema.
- 3. Diseño del programa.
- 4. Codificación.
- 5. Pruebas.
- 6. Despliegue del programa.
- 7. Mantenimiento.

## Herramienta de desarrollo

Debido a fallas con la herramienta Tkinter y tiempo de entrega corto se optó por cambiar la herramienta inicial. Para el desarrollo de esta GUI se utilizó Streamlit que es una API web compatible con el lenguaje Python donde su mayor enfoque es facilitar el front-end. La implementación es muy sencilla, solo tenemos que instalar Streamlit vía línea de comandos e importar las librerías correspondientes en nuestro proyecto.



Figura 1: Logo de Streamlit

## Implementación de algoritmos y análisis de datos

En la finalización del proyecto los algoritmos o análisis que se implementaron son los siguientes:

- Componente: Análisis Exploratorio de Datos (EDA)
- Componente: Selección de características (APC o ACD)

- Componente: Clustering (Jerárquico Ascendente y Jerárquico Particional)
- Componente: Reglas de asociación (Apriori)
- Componente: Pronóstico (Árbol de Decisión y Bosques Aleatorios)
- Componente: Clasificación (Árbol de Decisión y Bosques Aleatorios)

Se desplegaron los gráficos, métricas y tablas correspondientes. El usuario hará una selección de que opción quiere desplegar en pantalla y se debe mostrar de forma correcta.

## 4. Manual de usuario

### 4.1. Carga de Datos

Los datos se cargan a través de una barra lateral como se muestra en la figura 2.

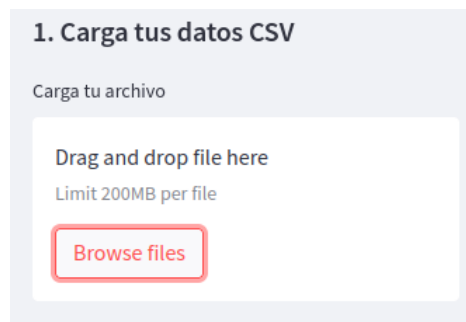


Figura 2: Botón para cargar los datos

Al presionar el botón desplegará la ventana del explorador de archivos de tu sistema operativo para que el usuario haga la selección del archivo con los datos como se muestra en la figura 3.

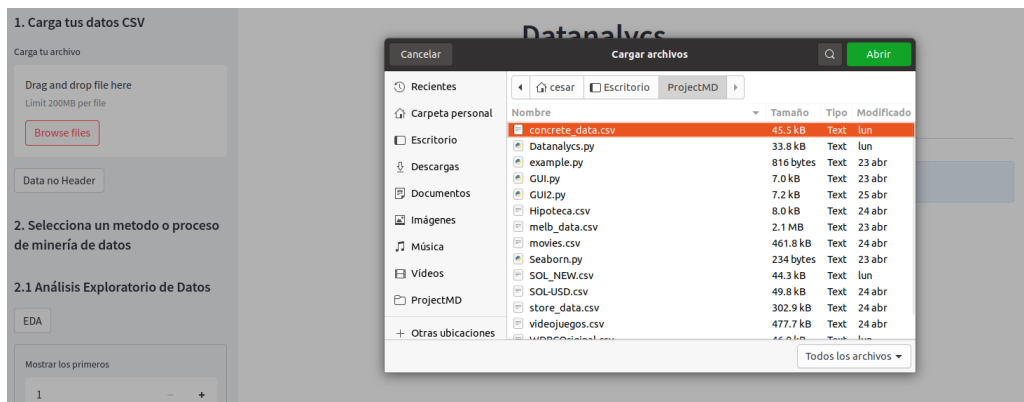


Figura 3: Despliegue de ventana

Seleccionamos los datos y dentro del botón aparecerá el archivo que seleccionamos, podemos hacer el cambio en cualquier momento sin problema. El cambio se muestra en la figura 4.

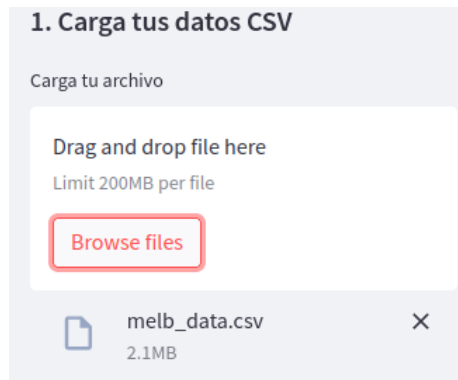


Figura 4: Botón para cargar los datos modificado

Ahora ya podemos trabajar con la data, como se muestra en la figura 5.

## Datanalycs

Esta es una GUI creada con [StreamLit](#)

Credit: Guadarrama Ortega César Alejandro

### Input DataFrame

	Suburb	Address	Rooms	Type	Price	M
0	Abbotsford	85 Turner St	2	h	1,480,000.0000	S
1	Abbotsford	25 Bloomburg St	2	h	1,035,000.0000	S
2	Abbotsford	5 Charles St	3	h	1,465,000.0000	S
3	Abbotsford	40 Federation La	3	h	850,000.0000	P
4	Abbotsford	55a Park St	4	h	1,600,000.0000	V
5	Abbotsford	129 Charles St	2	h	941,000.0000	S
6	Abbotsford	124 Yarra St	3	h	1,876,000.0000	S
7	Abbotsford	98 Charles St	2	h	1,636,000.0000	S
8	Abbotsford	6/241 Nicholson St	1	u	300,000.0000	S
9	Abbotsford	10 Valiant St	2	h	1.097.000.0000	S

Figura 5: Página principal de la aplicación web

## 4.2. Análisis Exploratorio de Datos (EDA)

Para el análisis exploratorio de datos se presiona el botón lateral llamado EDA, también si lo deseamos podemos mostrar los n valores del data frame. Esto se muestra en la figura 6.

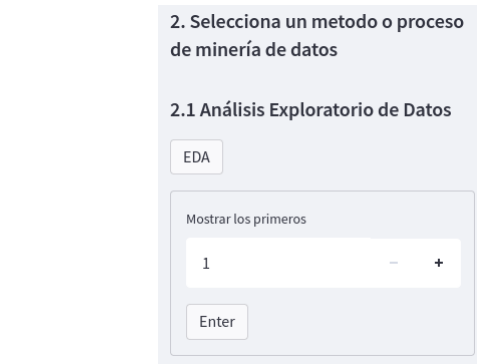


Figura 6: Botón de despliegue de EDA

La página principal cambiara totalmente al EDA con breves descripciones en cada uno de los puntos como se muestra en la figura 7.

## 1. Descripción de la estructura de los datos

### Input 10 Data Head

Es importante mostrar cierta cantidad de datos para visulizar la data.

	Suburb	Address	Rooms	Type	Price	Method	SellerG
0	Abbotsford	85 Turner St	2	h	1,480,000.0000	S	Biggin
1	Abbotsford	25 Bloomburg St	2	h	1,035,000.0000	S	Biggin
2	Abbotsford	5 Charles St	3	h	1,465,000.0000	SP	Biggin
3	Abbotsford	40 Federation La	3	h	850,000.0000	PI	Biggin
4	Abbotsford	55a Park St	4	h	1,600,000.0000	VB	Nelson
5	Abbotsford	129 Charles St	2	h	941,000.0000	S	Jellis
6	Abbotsford	124 Yarra St	3	h	1,876,000.0000	S	Nelson
7	Abbotsford	98 Charles St	2	h	1,636,000.0000	S	Nelson
8	Abbotsford	6/241 Nicholson St	1	u	300,000.0000	S	Biggin
9	Abbotsford	10 Valiant St	2	h	1,097,000.0000	S	Biggin

### Data types

El siguiente despliegue muestra los tipos de datos de las columnas (variables y tipos).

```
Suburb      object
Address     object
Rooms       int64
Type        object
```

Figura 7: Página principal con el despliegue de EDA

Recordando que este punto finaliza con la matriz correlacional como se muestra en la figura 8.

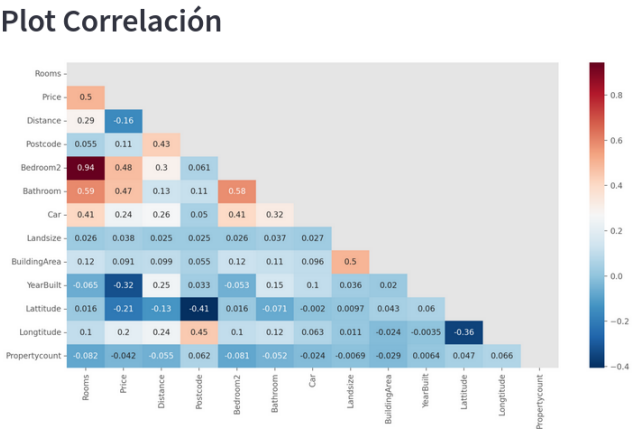


Figura 8: Página principal con a matriz correlacional

4.3. Selección de características (APC o ACD)

El botones de selección de características se encuentra en la barra lateral como se muestra en la figura 9. El botón ACP hace el análisis de componentes principales. El botón ACD hace el análisis de correlación de los datos. La caja de selección de evaluación visual mostrará los gráficos de este apartado. La caja de selección del top de valores, mostrará un despliegue de los valores correlacionales dependiendo de la variable seleccionada.

2.2 Componente: Selección de características:

2.2.1 Análisis de Componentes Principales (ACP)

ACP

2.2.1 Análisis de Correlacional de los Datos (ACD)

ACD

Evaluación visual

Enter

Top 10 valores de correlación

Enter

Figura 9: Botones y cajas de selección de características

Al presionar el botón deseado la página principal cambiará totalmente a la selección de características, si se presiona ACP se mostrará como se observa en la figura 10.

## Análisis de componentes principales (ACP)

1. Se hace una estandarización de los datos
2. Se calcula la matriz de covarianzas o correlaciones.
3. Se calculan los componentes (eigen-vectores) y la varianza (eigen-valores).
4. Se decide el número de componentes principales.
5. Se examina la proporción de relevancias –cargas–

## Estandarización de los datos

### Datos Estandarizados

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda
0	0.6201	0.1047	-1.6990	0.5044	0.6495	0.1959
1	1.0639	-0.1016	-0.7120	-0.5154	0.2592	1.9374
2	0.8912	0.2263	-0.9126	1.6672	1.0803	-0.3791
3	1.2742	1.1289	-1.5786	-1.5590	0.9096	2.1141
4	0.7196	-0.4000	0.0903	0.0273	0.1595	-0.1795
5	0.4367	-0.2232	-1.6107	-1.1356	0.7000	-0.0918
6	1.1146	1.2026	1.0692	-1.2310	0.4625	0.4151

Figura 10: Página principal con despliegue ACP

Si se presiona ACD se mostrará como se observa en la figura 11.

## Análisis correlacional de datos

### Matriz correlacional por el metodo de:

Pearson

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vi
ingresos	1.0000	0.5602	-0.1098	-0.1241	0.7129	
gastos_comunes	0.5602	1.0000	-0.0544	-0.0999	0.2094	
pago_coche	-0.1098	-0.0544	1.0000	0.0106	-0.1933	-
gastos_otros	-0.1241	-0.0999	0.0106	1.0000	-0.0644	-
ahorros	0.7129	0.2094	-0.1933	-0.0644	1.0000	
vivienda	0.6147	0.2048	-0.0946	-0.0546	0.6058	
estado_civil	-0.0426	-0.0572	0.0522	-0.0202	-0.0630	-
hijos	-0.0245	-0.0723	-0.0449	0.1248	0.0014	-
trabajo	-0.0389	-0.0791	0.0189	0.0473	-0.0238	-
comprar	0.4671	0.2002	-0.1965	-0.1103	0.3408	-

Figura 11: Página principal con despliegue ACD

## 4.4. Selección de variables

La selección de variables se hace a través de un apartado en la página principal y se puede descargar una nueva versión de la data como se muestra en la pagina 12.



## Selección de variables

Variables a eliminar

comprar x

```
[  
  "comprar"  
]
```

Nueva matriz de información

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	€
0	6000	1000	0	600	50000	400000	
1	6745	944	123	429	43240	636897	
2	6455	1033	98	795	57463	321779	
3	7098	1278	15	254	54506	660933	
4	6167	863	223	520	41512	348932	
5	5692	911	11	325	50875	360863	
6	6830	1298	345	309	46761	429812	
7	6470	1035	39	782	57439	606291	
8	6251	1250	209	571	50503	291010	
9	6987	1258	252	745	40611	324098	

Press to Download

Figura 12: Apartado de selección de variables

## 4.5. Clustering Jerárquico

El apartado de botones y cajas de selección se muestra en la figura 13. Recordando que para aplicar el algoritmo se hace de forma ascendente. Puedes desplegar el árbol ascendente, escoger el número de clústeres, que clúster específico quieres desplegar y los centroides finales.

### 2.3 Clustering Jerárquico

Clustering Jerarquico

Número de clusters

2 - +

Enter

Qué cluster quieres mostrar

0 - +

Enter

Centroides Clustering Jerárquico

Figura 13: Apartado de Clustering Jerárquico

La aplicación del algoritmo se queda hasta el despliegue del árbol, el usuario tendrá que escoger el número de clústers como se muestra en la figura 14.

## Arbol de clustering

Este arbol sirve para observar la cantidad de clústers que hay en nuestra data, el problema es que se tiene que seleccionar de forma manual

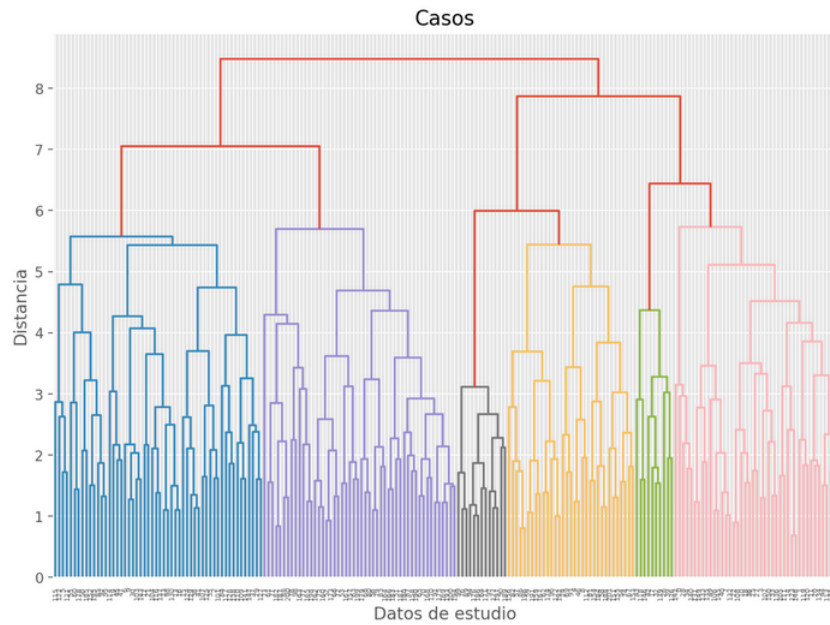


Figura 14: Árbol ascendente final

Finalmente con el último botón podemos observar todos los centroides como se muestra en la figura 15.

## Aplicación del algoritmo: Jerárquico Ascendente

### Centroides finales para el análisis

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda
0	6,061.2364	1,069.3091	181.9455	487.0727	47,917.4000	341,928.98
1	6,562.7561	1,080.7317	196.8537	463.6341	52,877.9512	557,275.78
2	2,937.3784	717.7838	231.8108	530.8378	23,903.4054	280,371.94
3	4,339.3030	1,063.6970	235.1515	515.2424	28,316.4848	291,658.51
4	6,061.1000	866.9000	155.3000	681.1000	61,871.4000	577,582.60
5	4,132.6154	1,188.0000	271.2308	573.2308	21,392.3077	304,409.53
6	2,543.1538	566.5385	252.1538	570.0769	23,715.0769	310,044.84

Figura 15: Centroides finales por el método de Clustering Jerárquico

## 4.6. Clustering Particional

El apartado de botones y cajas de selección se muestra en la figura 16. Recordando que para aplicar el algoritmo se hace de forma ascendente. Puedes desplegar la grafica de los datos marcando el punto de inflexión, escoger que clúster específico quieres desplegar y los centroides finales.



Figura 16: Apartado de Clustering Particional

La aplicación del algoritmo se queda hasta el grafico del punto de inflexión y el etiquetado final de los clústeres, como se muestra en la figura 17.

Grafica con marca

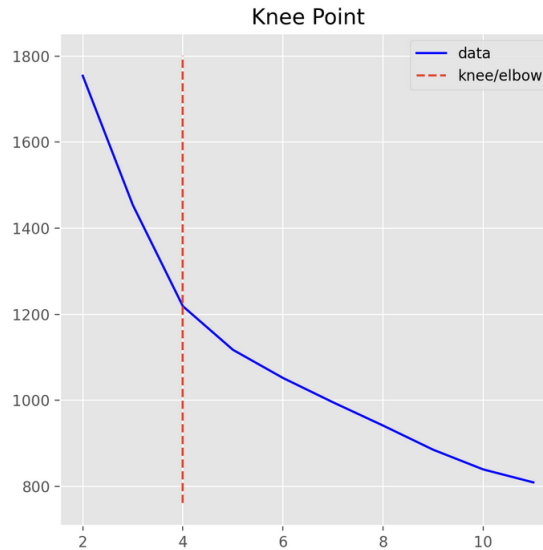


Figura 17: Gráfica del punto de inflexión por el metodo de la rodilla

Finalmente con el último botón podemos observar todos los centroides que se obtuvieron por medio de este método como se muestra en la figura 18.

## Número de elementos por clúster

	clusterP
0	46
1	58
2	50
3	48

## Centroides finales para el análisis

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivien
0	3,831.6957	923.1957	239.9565	530.7391	27,016.0870	296,854.65
1	6,105.6897	1,065.5862	180.8621	491.3793	48,317.4138	347,785.39
2	3,248.1400	840.0400	242.0600	541.8600	23,250.5200	286,621.72
3	6,435.8750	1,041.3958	190.4375	502.2708	54,578.2708	567,889.04

Figura 18: Centroides finales por el método de Clustering Particional

### 4.7. Reglas de Asociación: (Algoritmo Apriori)

Este algoritmo funciona solo con valores discretos y transaccionales. Para comenzar una nueva impresión de reglas de asociación debemos cargar la data correspondiente y presionar el botón de "Data no Header" para crear una matriz sin los cabezales o columnas principales, esto se muestra en la figura 19 y 20.

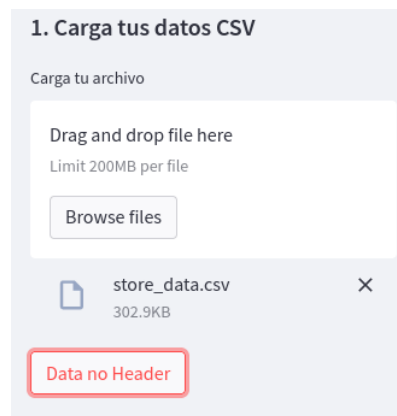


Figura 19: Sección de carga de archivo y modificador de data sin cabezal

## Input DataFrame No Header

	0	1	2	3
0	shrimp	almonds	avocado	vegetables mix
1	burgers	meatballs	eggs	<NA>
2	chutney	<NA>	<NA>	<NA>
3	turkey	avocado	<NA>	<NA>
4	mineral water	milk	energy bar	whole wheat rice
5	low fat yogurt	<NA>	<NA>	<NA>
6	whole wheat pasta	french fries	<NA>	<NA>
7	soup	light cream	shallot	<NA>
8	frozen vegetables	spaghetti	green tea	<NA>
9	french fries	<NA>	<NA>	<NA>

Figura 20: Data frame sin cabezal

Si presionamos el botón de Apriori mostrara la implementación del algoritmo hasta el ploteo del gráfico de consumo o transacción de los elementos individuales como se muestra en la figura 22, también podemos colocar los valores de soporte, confianza y elevación para completar el análisis y finalmente la impresión de reglas individuales o el conjunto de ellas, como se muestra en la figura 21.

### 2.5 Reglas de Asociación Algoritmo Apriori

Apriori

Support

0.01 - +

Confidence

0.10 - +

Lift

1.01 - +

Enter

Impresión de reglas

Ver regla exacta

1 - +

Enter

Figura 21: Sección de reglas de asociación

## Grafico de barras

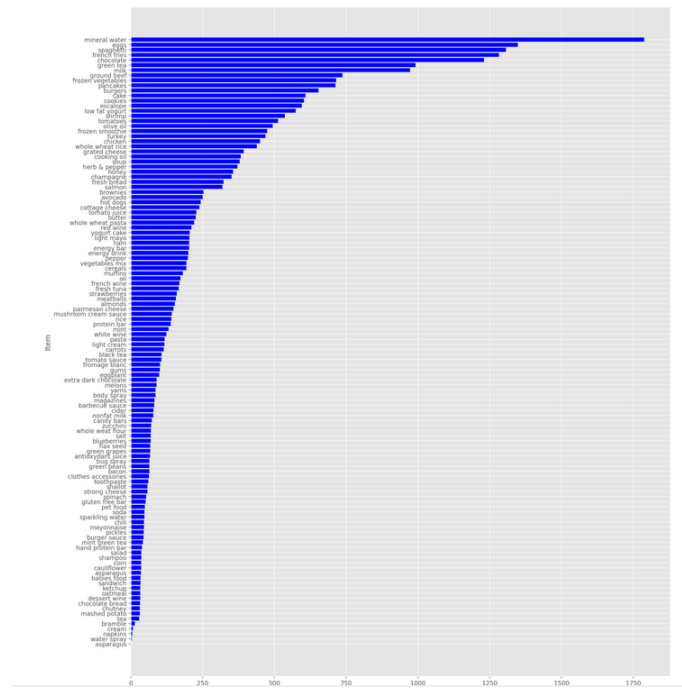


Figura 22: Grafico de barras

Finalmente creamos unas nuevas reglas de asociación con los siguientes valores:

- Support: 0.01
- Confidence: 0.3
- Lift: 2.0

El resultado de reglas se observa en la figura 23, dando como resultado 9 de estas, recordar que se pueden cambiar los valores en cualquier momento para crear nuevas reglas.

**2.5 Reglas de Asociación Algoritmo Apriori**

Apriori

Support  
0.01 - +

Confidence  
0.30 - +

Lift  
2.00 - +

Enter

Impresión de reglas

## Reglas de Asociación

Algoritmo: **Apriori**

Número de reglas

9

Data Frame Rules

Regla: frozenset({'herb & pepper', 'ground beef'})

Soporte: 0.015997866951073192

Confianza: 0.3234501347708895

Lift: 3.2919938411349285

=====

Figura 23: Reglas de asociación con datos introducidos

Si queremos realizar la impresión de una regla individualmente para su análisis solo debemos colocar el numero en la caja de selección del final, un ejemplo de ello se muestra en la figura 24.

Enter

Impresión de reglas

Ver regla exacta

3 - +

Enter

## Reglas de Asociación

Algoritmo: **Apriori**

Regla

3

RelationRecord(items=frozenset({'milk', 'soup'}), support=0.015197973603519531, orde

Figura 24: Impresión de regla individual

## 4.8. Pronóstico: (Arboles de Decisión y Bosques Aleatorios)

### Fuerza del concreto

Tenemos como contexto un conjunto de datos de materiales y procesos de mezcla para obtener la resistencia del concreto.

El objetivo es pronósticar la resistencia del concreto a partir de la cantidad, tiempo y demás atributos de preparación en una mezcla.

Los atributos o variables son las siguientes:

- Id: representa el identificador de la prueba asociada.

- Cement: Cantidad de cemento en kilogramos.
- Blast Furnace Slag: Cantidad de Blast Furnace Slag en kilogramos.
- Fly Ash: Cantidad de ceniza producida.
- Water: Cantidad de agua en litros.
- Super-plasticizer: Rigidez del cemento después del secado.
- Coarse Aggregate: La naturaleza gruesa de las partículas de cemento.
- Fine Aggregate: Finura del cemento.
- Age: Edad o tiempo antes de que necesite reparación.
- Streght: Resistencia del hormigón (/kN) por kiloNewton.

Si precionanamos el botón de Resultados como se observa el la figura 25 obtendremos el score de los dos modelos y observamos que es mejor el de bosques aleatorios con un puntaje de 0.89.

2.6.1 Pronóstico (Árbol de decisión) y (Bosques Aleatorios)

Resultados (Pronóstico) de Árbol de decisión y Bosques Aleatorios

Cement

1 - +

Blast Furnace Slag

1 - +

Fly Ash

0.00 - +

Water

1 - +

Superplasticizer

1 - +

Coarse Aggregate

1 - +

## Pronostico: Árbol de Decición y Bosques Aleatorios

Fuerza del Concreto, resultados del entrenamiento

Score Arbol de Decision: 0.8118877104627771

Score Bosque Aleatorio: 0.8906472437287175

Entrenamiento del modelo realizado, introduce los datos en la barra lateral para hacer un nuevo pronostico

Figura 25: Resultados finales del pronóstico

Hacemos un nuevo pronóstico con los siguientes valores:

- Cement: 148
- Blast Furnace Slag: 132
- Fly Ash: 0.0



- Water: 192
- Super-plasticizer: 10
- Coarse Aggregate: 890
- Fine Aggregate: 596
- Age: 28

El resultado del pronóstico se desplagara en la página principal, como se muestra en la figura 26. Se mostrarán los resultados usando los dos modelos previamente entrenados (Áboles y Bosques). Quedará a decisión del usuario que valor tomará para la práctica real.



Figura 26: Pronóstico realizado

## 4.9. Clasificación: (Arboles de Decisión y Bosques Aleatorios)

### Canal de distribución de Whosale

En contexto es un conjunto de datos de diferentes clientes de un distribuidor mayorista. Incluye el gasto anual en unidades monetarias (M.U.) en diversas categorías de productos.

El objetivo es clasificar el canal de distribución dependiendo de los valores en gastos anuales en diferentes productos y región de distribución.

Los atributos o variables son las siguientes:

- FRESH: gasto anual (u.m.) en productos frescos (Continuo)
- MILK: gasto anual (u.m.) en productos lácteos (Continuo)
- GROCERY: gasto anual (m.u.) en productos de abarrotes (Continuo)
- FROZEN: gasto anual (u.m.) en productos congelados (Continuo)

- DETERGENTS-PAPER: gasto anual (u.m.) en detergentes y productos de papel (Continuo)
- DELICATESSEN: gasto anual (m.u.) en productos y golocinas (Continuo)
- CHANNEL: Canal de clientes - Horeca (Hotel/Restaurante/Cafetería) o Canal Retail (Nominal)
- REGION: Región del cliente – Lisnon, Oporto u otra (Nominal)

Si precionanamos el botón de Resultados como se observa el la figura 27 obtendremos el score de los dos modelos y observamos que es mejor el de bosques aleatorios con un puntaje de 0.897.

2.7 Clasificación: Cadena de distribución de Whosale

2.7.1 Clasificación (Árbol de decisión) y (Bosques Aleatorios)

Resultados (Clasificación) de Árbol de decisión y Bosques Aleatorios

Region

1 - +

Fresh

1 - +

Milk

1 - +

Grocery

1 - +

Frozen

1 - +

## Clasificación: Árbol de Decisión y Bosques Aleatorios

### Canal de distribución de Whosale, resultados del entrenamiento

#### Separación de los dos tipos de canal

#0:Horeca (Hotel/Restaurante/Café)

#1:Retail u Otro

	0
0	298
1	142

Score Arbol de Decision: 0.8522727272727273

Score Bosque Aleatorio: 0.8977272727272727

Figura 27: Resultados finales de la clasificación

Hacemos una nueva clasificación con los siguientes valores:

- REGION: 2 – Otra
- FRESH: 6350
- MILK: 1431
- GROCERY: 4221
- FROZEN: 1038
- DETERGENTS-PAPER: 477
- DELICATESSEN: 1773

El resultado de la clasificación se desplegara en la página principal, como se muestra en la figura 28. Se mostrarán los resultados usando los dos modelos previamente entrenados (Áboles y Bosques). Quedará a decisión del usuario que valor tomará para la práctica real. Si queremos hacer otra clasificación solo tenemos que agregar nuevos valores.



Figura 28: Clasificación realizada

## 5. Video demostrativo

Para finalizar se realizó un video demostrativo del proyecto completo, introduciendo datos de nuestro interés y mostrando cada análisis o algoritmo implementados. El video está en una calidad 1080p a 60fps, se encuentra dentro de la misma carpeta de Drive o a través de este enlace de YouTube: <https://youtu.be/VE6TEmCFn3I>

## 6. Conclusiones

El objetivo del proyecto se cumplió, la GUI de minería de datos funciona de manera satisfactoria. Encontré muchos tropiezos en la realización de este proyecto, cambié de herramienta de desarrollo en un inicio por cuestiones de conocimiento e implementación, pero al final encontré la solución más viable usando Streamlit. Este proyecto implementa el conjunto de todos los conocimientos de la asignatura, donde a través de la inteligencia artificial, entrenamiento de modelos y aplicación de algoritmos podemos realizar análisis precisos para el ambiente práctico.

## 7. Referencias

### Referencias

- [1] Fuente de datos para el pronóstico: <https://www.kaggle.com/datasets/prathamtripathi/regression-with-neural-networking>
- [2] Fuente de datos para la clasificación: <https://archive.ics.uci.edu/ml/datasets/Wholesale+customers>
- [3] Streamlit. (2022). A faster way to build and share data apps.Documentación recuperada de <https://streamlit.io/>
- [4] Colaboradores de Wikipedia. (2022). Desarrollo en cascada. Recuperado de [https://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](https://es.wikipedia.org/wiki/Desarrollo_en_cascada)