Instituto Politécnico Nacional

Escuela Superior de Cómputo

Extraordinario

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

Indicaciones para el Examen Extraordinario de la Unidad de Aprendizaje de Minería de Datos

Examen Extraordinario de la Unidad de Aprendizaje de Minería de Datos

Periodo escolar: 2022-1

Grupos: 3CM10 y 3CV19

Las actividades se realizan de forma individual.

El examen extraordinario de este semestre escolar considera cuatro actividades: un examen teórico, dos videos expositivos y un reporte, las indicaciones se presentan en este documento.

1. **Examen teórico**

Se realizará el lunes 20 de diciembre, en el horario de clase. Incluirá todos los temas considerados en el curso.

2. **Dos Videos expositivos**

Lo realizarán de forma individual, con el contenido que se indica a continuación:

- **Primer Video:**
  Sintetice y presente una breve explicación de todos los temas del marco teórico considerados en el curso. Realizar una presentación en power point.
- **Segundo Video:**
  Presentar el análisis de un conjunto de datos mediante las técnicas de minería de datos, indicadas en el proyecto final. NO se permite que presente el proyecto final, ya que se realizó en equipo.
  Consulte la guía del reporte, debido a que es la estructura de este video.

3. **Un reporte**

El reporte tiene la estructura que estuvimos trabajando en el proyecto final.

Por favor atender los siguientes aspectos:

- El alumno debe aparecer en los videos. No usar modificadores de voz
- Cada video debe tener una duración de 25 a 30 minutos
- Realizar las citas del material bibliográfico que consulte. NO se permite que sólo copien y peguen los materiales que yo les he proporcionado
- Presente el reporte en formato pdf
- En el reporte, agregar el enlace de la carpeta en donde están los videos, verifique que se pueda acceder a ellos antes de que envíe el reporte
- Oficialmente, la fecha límite para la entrega de la actividad 2 (videos) y 3 (reporte) es el miércoles 22 de diciembre, aunque la plataforma estará activada para recibir sus trabajos de forma extraordinaria hasta el lunes 27 de diciembre.

- *Si tiene alguna duda sobre las especificaciones del examen extraordinario, por favor comunicarse por medio de la plataforma Microsoft Teams, que usamos en el curso.*

1. La estructura del proyecto es la siguiente:

A. CONJUNTO DE DATOS.

Identificar un conjunto de datos con diversos tipos, al que le pueda aplicar técnicas de tratamiento de datos, de un tamaño y estructura similar al que uso en los proyectos de limpieza de datos.

i. Descripción del conjunto de datos: intención, fuente de acceso, autor, fecha de aportación.

ii. Diccionario de datos: número de atributo, nombre, tipo, significado y dominio.

B. PROPUESTA DE TÉCNICAS DE MINERÍA DE DATOS:

Indicaciones para Proyecto Final 2022-1

Técnica	Objetivo	Atributos
Árboles (Tipo)	Construir un modelo para predecir qué fármaco podría ser apropiado para un futuro paciente con la misma enfermedad, según el fármaco administrado.	Age Sex BP Cholesterol Na_to_K Drug
Agrupamiento **(#)**	Desea comprender a los clientes, como quiénes son los clientes objetivo, de modo que se pueda dar sentido al equipo de marketing y planificar la estrategia en consecuencia.	ID Sex Marital status Age Education Income Occupation Settlement size
Regresión lineal **(&)**	El objetivo es poder predecir con precisión los costos del seguro médico, con base en los atributos de la edad y la prima del seguro.	age sex bmi children smoker region charges
Regresión logística	Intentar predecir si las personas compraron un producto publicitado, utilizando el método de regresión logística.	User ID Gender Age EstimatedSalary Purchased
Reglas de asociación **(*)**	Sumergirse y buscar si hay alguna diferencia o correlación entre las canastas. Dado que el marco de datos ya está tabulado como un marco de datos activo, de inmediato usaremos el conjunto de datos que se analizará con el algoritmo APRIORI.	# Apple Bread Butter Cheese Corn Dill Eggs Ice cream Kidney Beans

		Milk Nutmeg Onion Sugar Unicorn Yogurt chocolate
Análisis de componentes principales	El objetivo de este método es reorientar los datos para que una multitud de variables originales se puedan resumir con relativamente pocos factores o componentes que capturen la máxima información posible de las variables originales.	GNP.deflator GNP Unemployed Armed.Forces Population Employed

(#) Agregar las métricas de la silueta y el estadístico Pseudo-F para analizar los elementos de los grupos creados.
(&) En la regresión lineal agregar gráfica de residuales, prueba de normalidad y prueba de significancia estadística.
(*) En Reglas de Asociación, puede crear un catálogo de oferta de productos, integrando imágenes. No debe ser el que realizó en el proyecto de tema.

C. ASPECTOS A CONSIDERAR

La estructura de la investigación es la siguiente (reporte y video): En el reporte: Crear una portada para cada técnica, mencionando el nombre que corresponda.
Para cada una de las técnicas de minería de datos debe agregar los aspectos indicados a continuación.

i. Introducción.

- Desarrollar el marco teórico del tema
- Agregar una breve descripción del estudio que se realizó en esta sección, así como los resultados encontrados.
- Intención de la aplicación de la técnica
- Justificación de la técnica a aplicar

ii. Diccionario de datos. Fracción del conjunto de datos que utilizó en cada técnica, considerando: nombre de la variable, tipo de dato (nominal, ordinal, numérico discreto o continuo), dominio de valores.

iii. Resultados. Medidas de estimación y valoración de la obtención y resultados encontrados.

a. Diagrama generado.
b. Medidas obtenidas
c. Descripción de las características de los resultados generados.
d. Tipo de muestra que utilizó para prueba y entrenamiento.

iv. Análisis de los resultados. Describir los resultados encontrados proporcionando la explicación de los mismos. Así como las conclusiones que obtenga. Las métricas de evaluación deben corresponder a la técnica aplicada.
v. Anexo. Indicar el nombre del archivo que contiene el código correspondiente, ubicado en la carpeta compartida

3. Anexo A. Agregue lo que considere adecuado.

Ponderación
Examen - 30%
Video teoría - 20%
Video ejemplo - 20%
Reporte - 30%

Instituto Politécnico Nacional

Escuela Superior de Cómputo

ÁRBOLES

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

- # INTRODUCCIÓN
  - ## MARCO TEÓRICO

Árbol ID3

El algoritmo ID3 es un tipo de árbol de decisión, se basa en el principio de la navaja de Occam, es decir, hacer todo lo posible con menos cosas.

Como sabemos construir un árbol de decisión permite que se explique cada instancia de la secuencia de entrada de la manera más compacta posible a partir de una tabla de inducción.

Principalmente el árbol ID3 favorece indirectamente a aquellos atributos con muchos valores, los cuales no tienen que ser los más útiles. Además cabe resaltar que examina todos los atributos y escoge el de máxima ganancia, forma la ramificación y usa el mismo proceso recursivamente para formar sub-árboles a partir de los nodos generados.

Lo que destacar que este árbol puede ser aplicable sólo a problemas de clasificación y diagnóstico, además determinan las variables que aportan información relevante para la solución del problema.

  - ## BREVE DESCRIPCIÓN DEL ESTUDIO QUE SE REALIZÓ

Lo que se realizó fue llevar a cabo la implementación de un árbol de decisión ID3, y aplicando el criterio de gini.

  - ## INTENCIÓN DE LA APLICACIÓN DE LA TÉCNICA

Considerando el conjunto de datos, vamos a construir un modelo para predecir qué fármaco podría ser apropiado para un futuro paciente con la misma enfermedad, según el fármaco administrado.

  - ## JUSTIFICACIÓN DE LA TÉCNICA A APLICAR

Se utilizó el árbol de decisión ID3, ya que selecciona el atributo que subdivide los ejemplos de la mejor manera.

- # DICCIONARIO DE DATOS
  - ## Intención:

Imagine que es un investigador médico que recopila datos para un estudio. Ha recopilado datos sobre un conjunto de pacientes, todos los cuales padecían la misma enfermedad. Durante el curso de su tratamiento, cada paciente respondió a uno de los 5 medicamentos, Medicamento A, Medicamento B, Medicamento c, Medicamento x & y.

Parte de su trabajo es construir un modelo para descubrir qué fármaco podría ser apropiado para un futuro paciente con la misma enfermedad. Las características de este conjunto de datos son la edad, el sexo, la presión arterial y el colesterol de los pacientes, y el objetivo es el fármaco al que respondió cada paciente.

Es una muestra de clasificador multiclase, y puede usar la parte de entrenamiento del conjunto de datos para construir un árbol de decisiones y luego usarlo para predecir la clase de un paciente desconocido o para recetar un medicamento a un paciente nuevo.

Fuente de datos: IBM

  - ## Fuente de acceso:

https://www.kaggle.com/pablomgomez21/drugs-a-b-c-x-y-for-decision-trees

  - ## Autor:

Sources: IBM Developer Skills Network
Collection methodology: AUTHOR: Saeed Aghabozorgi

- **Fecha de aportación:**

Última actualización: 2021-06-26
Fecha de creación: 2021-06-26

Nombre	Significado	Tipo	Dominio
Age	Edad	Numérico	[15 - 74]
Sex	Sexo	Categórico	M F
BP	Presión sangre	Categórico	Low Normal High
Cholesterol	Colesterol	Categórico	Normal High
Na_to_K	Sodio a Potasio	Numérico	[6.269 - 38.247]
Drug	Medicamento	Categórico	drugA drugB drugC drugX drugY

RESULTADOS

- **DIAGRAMA GENERADO**

- **MEDIDAS OBTENIDAS**

Predicción

```
pt = dt.predict(x_test)
print(pt[0:5])
print(y_test[0:20])
```

```
['drugY' 'drugX' 'drugX' 'drugX' 'drugX']
40     drugY
51     drugX
139    drugX
197    drugX
170    drugX
82     drugC
183    drugY
46     drugA
70     drugB
100    drugA
179    drugY
83     drugA
25     drugY
190    drugY
159    drugX
173    drugY
95     drugX
3      drugX
41     drugB
58     drugX
Name: Drug, dtype: object
```

Medición de la precisión

```
[192] print("Accuracy of the decision tree: ", metrics.accuracy_score(y_test,pt))

      Accuracy of the decision tree:  0.9833333333333333
```

- **DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS RESULTADOS GENERADOS**

Debido a que utilizamos el criterio de Gini, obtuvimos la precisión máxima del Árbol de decisión ID3: **0.98333**.

- **TIPO DE MUESTRA QUE UTILIZÓ PARA PRUEBA Y ENTRENAMIENTO**

Para la etapa de entrenamiento y prueba se utilizó la función train_test_split(), la cual recibió como parámetros: test_size=0.3, random_state=3, los cuales se encargan de elegir de forma aleatoria los datos para ambas etapas; Se utilizó el mismo dataset, y este consta de 201 filas de datos.

• ANÁLISIS DE RESULTADOS

- **DESCRIBIR LOS RESULTADOS ENCONTRADOS PROPORCIONANDO LA EXPLICACIÓN DE ESTOS**

En la etapa de Predicción, nuestra variable objetivo es Drug, y nos arrojó con bastante precisión la predicción de qué fármaco podría ser apropiado para un futuro paciente con la misma enfermedad, según el fármaco administrado.

En la etapa de Medición de la Precisión del árbol de decisión ID3, se calculó la precisión de este con un valor de: 0.98333, lo que significa que cuenta con gran precisión, ya que el valor máximo es 1.

- **CONLUSIONES**

Los resultados del algoritmo implementado en Python nos permiten conocer y predecir a que pacientes es recomendable administrar cierto medicamento, dependiendo de los síntomas y características que presente, con un precisión del 98%.

ANEXO

- **NOMBREL DEL ARCHIVO**

Dataset: drug200.csv
Código: 1_Arboles.ipynb

Instituto Politécnico Nacional

Escuela Superior de Cómputo

CLUSTERING / AGRUPAMIENTO

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

- # INTRODUCCIÓN
  - ## MARCO TEÓRICO

El agrupamiento o clustering es una técnica de minería de datos, y consiste en la división de los datos en grupos de objetos similares. Cuando se representa la información obtenida a través de clusters se pierden algunos detalles de los datos, pero a la vez se simplifica dicha información.
También, es similar a la clasificación, excepto que los grupos no son predefinidos y su objetivo es particionar o segmentar un conjunto de datos o individuos en grupos que pueden ser disjuntos o no.
Nos brinda una ayuda la cual consiste en la clasificación automática tiene por objetivo reconocer grupos de individuos homogéneos, de tal forma que los grupos queden bien separados y diferenciados.
Una de las diferentes tareas que tiene la minería de datos, es el agrupamiento o también llamado clustering, la cual pudimos tomar como una herramienta de clasificación o separación de los datos, de igual manera nos servirá durante del desarrollo de este proyecto.
Psuedo F describe la relación entre la varianza entre conglomerados y la varianza dentro del conglomerado. Si Psuedo F está disminuyendo, eso significa que la varianza dentro del conglomerado aumenta o permanece estática (denominador) o la varianza entre conglomerados está disminuyendo (numerador).
La varianza dentro de un clúster realmente mide qué tan ajustados encajan sus clústeres. Cuanto mayor es el número, más disperso está el grupo, menor es el número, más enfocado está el grupo. La varianza entre conglomerados mide qué tan separados están los conglomerados entre sí.
El objetivo de K-means es minimizar la varianza dentro de los conglomerados (maximizando necesariamente la varianza entre los conglomerados). Entonces, la forma en que puede interpretar esto es: a medida que aumenta el número de grupos, aumenta la varianza dentro del grupo, lo que hace que los grupos reales sean más dispersos / menos compactos y, por lo tanto, menos efectivos (y potencialmente más cercanos a otros grupos).
Dicho esto, todas sus interpretaciones son posibles. Pero antes de seguir adelante y descartar k-medias, debe intentar observar el método del codo (gráfico de número de conglomerados frente a la varianza entre grupos dividido por la varianza total del grupo); si no hay un codo en el gráfico, suele ser una buena señal de que k-means no proporcionará resultados útiles (o al menos esa es mi prueba de fuego).

  - ## BREVE DESCRIPCIÓN DEL ESTUDIO QUE SE REALIZÓ

Se aplicó el método de k-means, el cual es un algoritmo simple de aprendizaje automático no supervisado que agrupa los datos en un número específico (k) de clústeres. ... El método del codo ejecuta la agrupación de k-medias en el conjunto de datos para un rango de valores para k (digamos de 1 a 10) y luego, para cada valor de k, calcula una puntuación promedio para todos los grupos.

  - ## INTENCIÓN DE LA APLICACIÓN DE LA TÉCNICA

Esto se hizo con el fin de agrupar los 2000 datos de los consumidores, tomando en cuenta su edad y los ingresos.

  - ## JUSTIFICACIÓN DE LA TÉCNICA A APLICAR

Se aplicó esta técnica debido a su alto nivel de precisión, y su fácil configuración en el lenguaje de Python.

- # DICCIONARIO DE DATOS
  - **Intención:**

La segmentación de clientes es la subdivisión de un mercado en grupos de clientes discretos que comparten características similares. La segmentación de clientes puede ser un medio poderoso para identificar las necesidades insatisfechas de los clientes. Al utilizar los datos anteriores, las empresas pueden superar a la competencia mediante el desarrollo de productos y servicios especialmente atractivos. Tiene un centro comercial de supermercado y, a través de las tarjetas de membresía, tiene algunos datos básicos sobre sus clientes, como ID de cliente, edad, sexo, ingresos anuales y puntaje de gastos. Desea comprender a los clientes, como quiénes son los clientes objetivo, de modo que se pueda dar sentido al equipo de marketing y planificar la estrategia en consecuencia.

- **Fuente de acceso:**

https://www.kaggle.com/dev0914sharma/customer-clustering

- **Autor:**

Dev Sharma

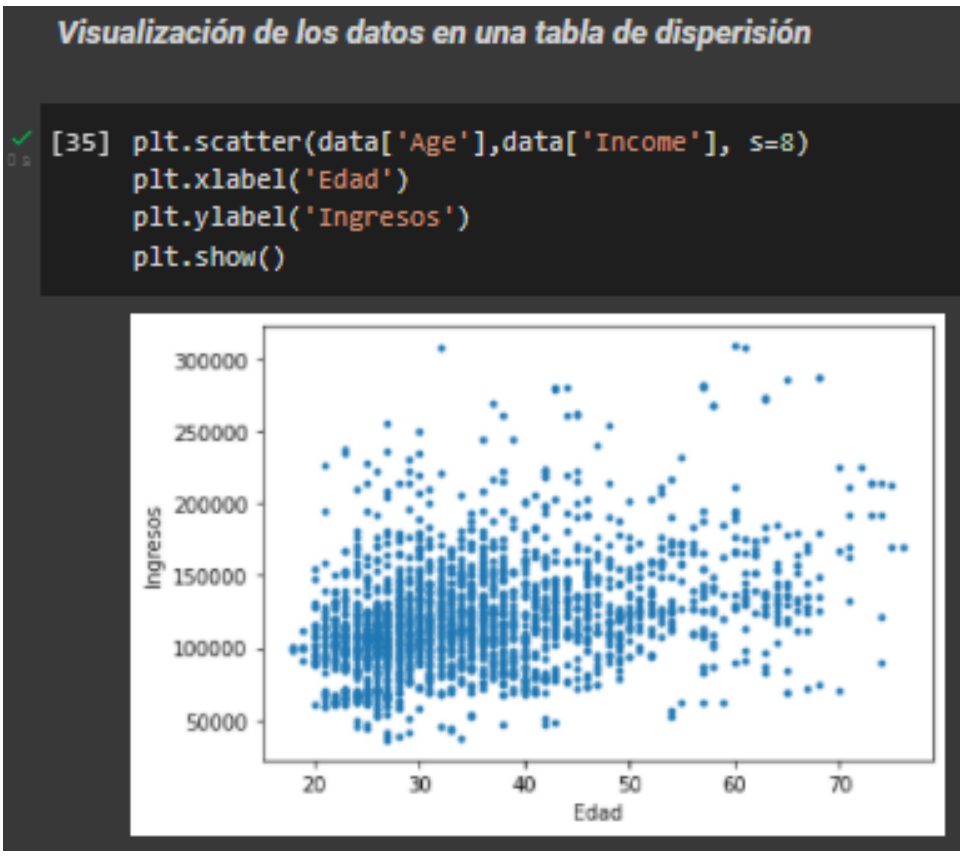
- **Fecha de aportación:**

Última actualización: 2021-05-07
Fecha de creación: 2021-05-07

Nombre	Significado	Tipo	Dominio
ID	▪ Muestra un identificador único de un cliente.	Numérico	Real
Sex	▪ 0: macho ▪ 1: hembra	Categórico	{0,1}
Marital status	▪ 0: soltero ▪ 1: No soltero (divorciado / separado / casado / viudo)	Categórico	{0,1}
Age	▪ 18 Valor mínimo (la edad más baja observada en el conjunto de datos) ▪ 76 Valor máximo (la edad más alta observada en el conjunto de datos)	Numérico	Real
Education	▪ 0 otro / desconocido ▪ 1 escuela secundaria ▪ 2 universidad ▪ 3 escuela de posgrado	Categórico	{0,1,2,3}
Income	▪ 35832 Valor mínimo (el ingreso más bajo observado en el conjunto de datos) ▪ 309364 Valor máximo (el ingreso más alto observado	Numérico	Real

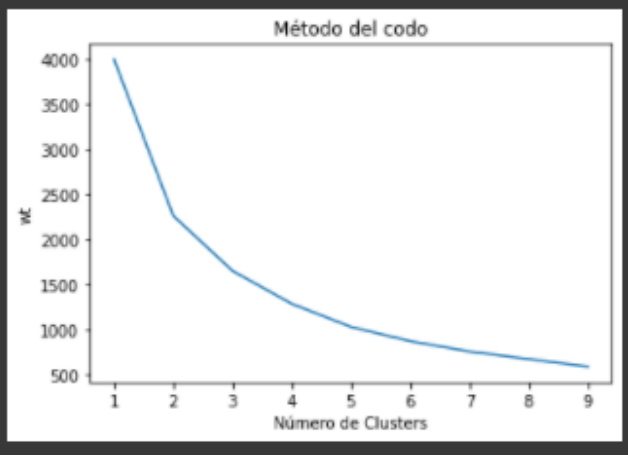
	en el conjunto de datos)		
Occupation	▪ 0 desempleados / no calificados ▪ 1 empleado / funcionario calificado ▪ 2 gerencia / autónomo / empleado altamente calificado / funcionario	Categórico	{0,1,2}
Settlement size	▪ 0 ciudad pequeña ▪ 1 ciudad mediana ▪ 2 gran ciudad	Categórico	{0,1,2}

- **RESULTADOS**
  - **DIAGRAMA GENERADO**



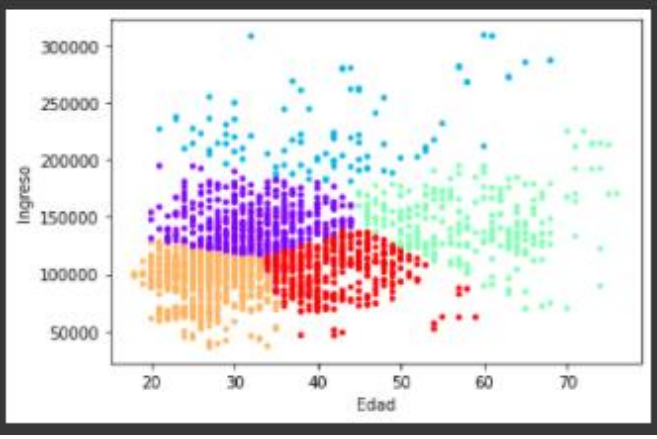
Visualización de la gráfica del Método del codo

```
ms=StandardScaler()
stand=ms.fit_transform(x)

wt=[]
for i in range(1, 10):
    kmeans=KMeans(n_clusters=i,init= 'k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(stand)
    wt.append(kmeans.inertia_)
plt.plot(range(1, 10),wt)
plt.title("Método del codo")
plt.xlabel("Número de Clusters")
plt.ylabel("wt")
plt.show()
```

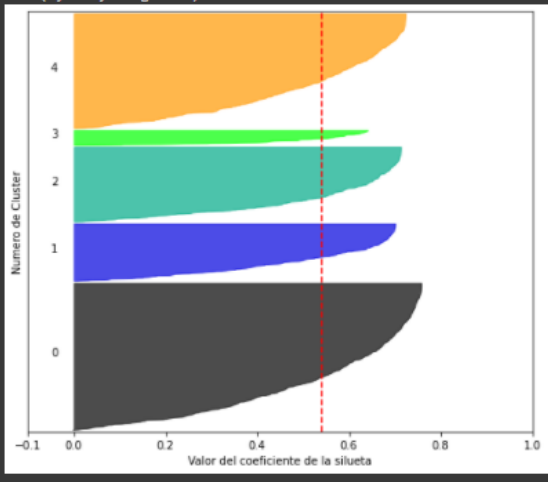


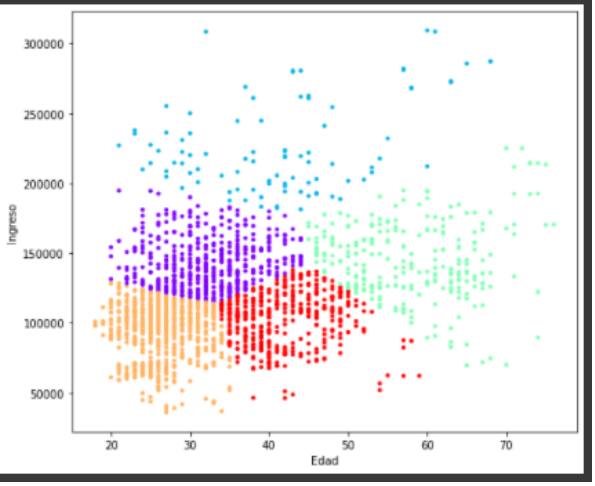
Visualización del proceso de Agrupamiento

```
plt.scatter(new_data['Age'],new_data["Income"],c=new_data['cluster'],cmap='rainbow', s=8)
plt.xlabel('Edad')
plt.ylabel('Ingreso')
plt.show()
```



```
For n_clusters = 5
The average silhouette_score is =  0.5401520680025738
Text(0, 0.5, 'Ingreso')
```

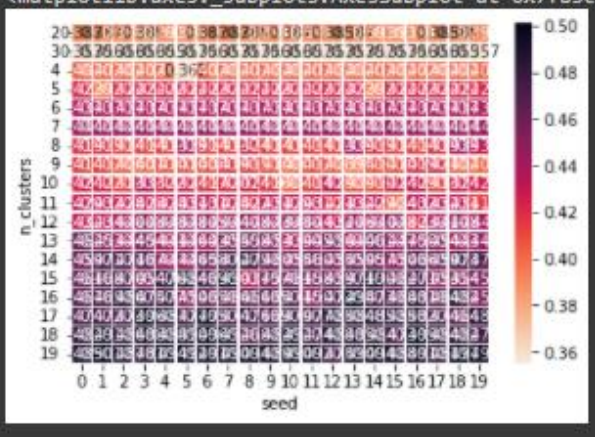




Análisis estadístico Pseudo-F

```
clusters_range=range(2, 20)
random_range  =range(0, 20)
results=[]

for c in clusters_range:
    for r in random_range:
        clusterer=KMeans(n_clusters=c, random_state=r)
        cluster_labels=clusterer.fit_predict(scaled)
        silhouette_avg=silhouette_score(scaled, cluster_labels)
        results.append([c,r,silhouette_avg])

result  =pd.DataFrame(results, columns=["n_clusters","seed","silhouette_score"])
pivot_km=pd.pivot_table(result, index="n_clusters", columns="seed",values="silhouette_score")

plt.figure()
sns.heatmap(pivot_km, annot=True, linewidths=.5, fmt='.3f', cmap=sns.cm.rocket_r)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f83c5957850>
```



- **MEDIDAS OBTENIDAS**

Configuración del clasificador

```
kmeans=KMeans(n_clusters=5,random_state=0)
kmeans.fit(stand)

prediction=kmeans.fit_predict(stand)
prediction
```

```
array([2, 0, 4, ..., 3, 3, 3], dtype=int32)
```

Agregamos una columna para la predicción del Clustering

```
[39] new_data=data.copy()
     new_data['cluster']=prediction
     new_data.head()
```

	ID	Sex	Marital status	Age	Education	Income	Occupation	Settlement size	cluster
0	100000001	0	0	67	2	124670	1	2	2
1	100000002	1	1	22	1	150773	1	2	0
2	100000003	0	0	49	1	89210	0	0	4
3	100000004	0	0	45	1	171565	1	1	2
4	100000005	0	0	53	1	149031	1	1	2

- **DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS RESULTADOS GENERADOS**

El proceso de agrupamiento arrojó 5 grupos que comparten características en base a su edad e ingresos, los cuales se calcularon aplicando el método del codo, éste ejecuta la agrupación de k-means en el conjunto de datos para un rango de valores para k (digamos de 1 a 10) y luego, para cada valor de k, calcula una puntuación promedio para todos los grupos.

- **TIPO DE MUESTRA QUE UTILIZÓ PARA PRUEBA Y ENTRENAMIENTO**

Para la etapa de entrenamiento y prueba se utilizó la función KMeans(), la cual recibió como parámetros: n_clusters=5, random_state=0, los cuales se encargan de elegir de forma aleatoria los datos para ambas etapas, con base a los 5 clusters generados; Se utilizó el mismo dataset, y este consta de 2000 filas de datos.

• ANÁLISIS DE RESULTADOS

- **DESCRIBIR LOS RESULTADOS ENCONTRADOS PROPORCIONANDO LA EXPLICACIÓN DE ESTOS**

Se utilizó la función kmeans.fit_predict(stand) para predecir los grupos a los que va a pertenecer cada consumidor, el cual nos arroja un arreglo con 2000 posiciones, las mismas que pertenecen a cada fila. Por último se agrega la columna de Cluster, para poder visualizar los grupos en una tabla de dispersión.

- **CONLUSIONES**

Vimos cómo es posible implementar tanto el método de agrupamiento por K-Means como el Método del codo en un conjunto de datos bidimensionales, y cómo estos algoritmos son capaces entonces de

encontrar las semejanzas intrínsecas de los datos y producir clases que, en efecto, reproducen lo que puede observarse de manera intuitiva a partir de la gráfica de dispersión.

- **ANEXO**
  - **NOMBREL DEL ARCHIVO**

Dataset: segmentation data.csv
Código: 2 - Clustering.ipynb

Instituto Politécnico Nacional

Escuela Superior de Cómputo

REGRESION LINEAL

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

- # INTRODUCCIÓN
  - ## MARCO TEÓRICO

La regresión lineal es una técnica de modelado estadístico que se emplea para describir una variable de respuesta continua como una función de una o varias variables predictoras. Puede ayudar a comprender y predecir el comportamiento de sistemas complejos o a analizar datos experimentales, financieros y biológicos.
Esta forma de análisis estima los coeficientes de la ecuación lineal, involucrando una o más variables independientes que mejor predicen el valor de la variable dependiente.
La regresión lineal se ajusta a una línea recta o a una superficie que minimiza las discrepancias entre los valores de salida previstos y reales. Hay calculadoras de regresión lineal simple que utilizan el método de "mínimos cuadrados" para determinar la línea que mejor se ajusta para un conjunto de datos pareados.
Los modelos de regresión lineal son relativamente sencillos y proporcionan una fórmula matemática fácil de interpretar que puede generar predicciones.
También puede utilizarse para proporcionar mejores perspectivas mediante el descubrimiento de patrones y relaciones que sus colegas ya pueden haber visto y pensado que habían entendido.

  - ## BREVE DESCRIPCIÓN DEL ESTUDIO QUE SE REALIZÓ

Se llevó a cabo el análisis de los datos y posteriormente se realizó regresión lineal sobre el dataset.

  - ## INTENCIÓN DE LA APLICACIÓN DE LA TÉCNICA

La intención de la aplicación de la técnica de regresión lineal fue para poder predecir con precisión los costos del seguro.

  - ## JUSTIFICACIÓN DE LA TÉCNICA A APLICAR

La aplicación de la técnica de regresión lineal se justifica debido a la naturaleza del dataset, y es que buscamos predecir una variable dependiente (Seguro médico) utilizando los datos de los usuarios; Y precisamente la regresión lineal es ideal para atacar este tipo de problemas y poder calcular un modelo matemático fácil de interpretar.

- # DICCIONARIO DE DATOS
  - ## Intención:

Machine Learning with R de Brett Lantz es un libro que ofrece una introducción al aprendizaje automático usando R. Hasta donde yo sé, Packt Publishing no hace que sus conjuntos de datos estén disponibles en línea a menos que usted compre el libro y cree una cuenta de usuario que puede ser una problema si está sacando el libro de la biblioteca o pidiéndole prestado el libro a un amigo. Todos estos conjuntos de datos son de dominio público, pero simplemente es necesario limpiarlos y recodificarlos para que coincidan con el formato del libro.

  - ## Fuente de acceso:

https://www.kaggle.com/mirichoi0218/insurance

  - ## Autor:

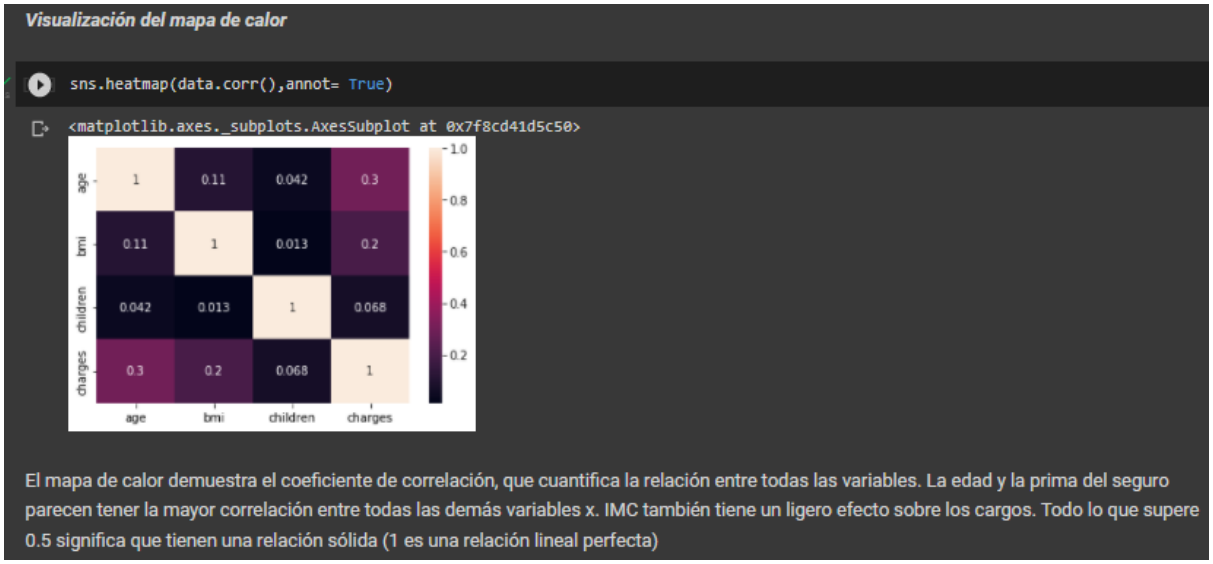
Miri Choi

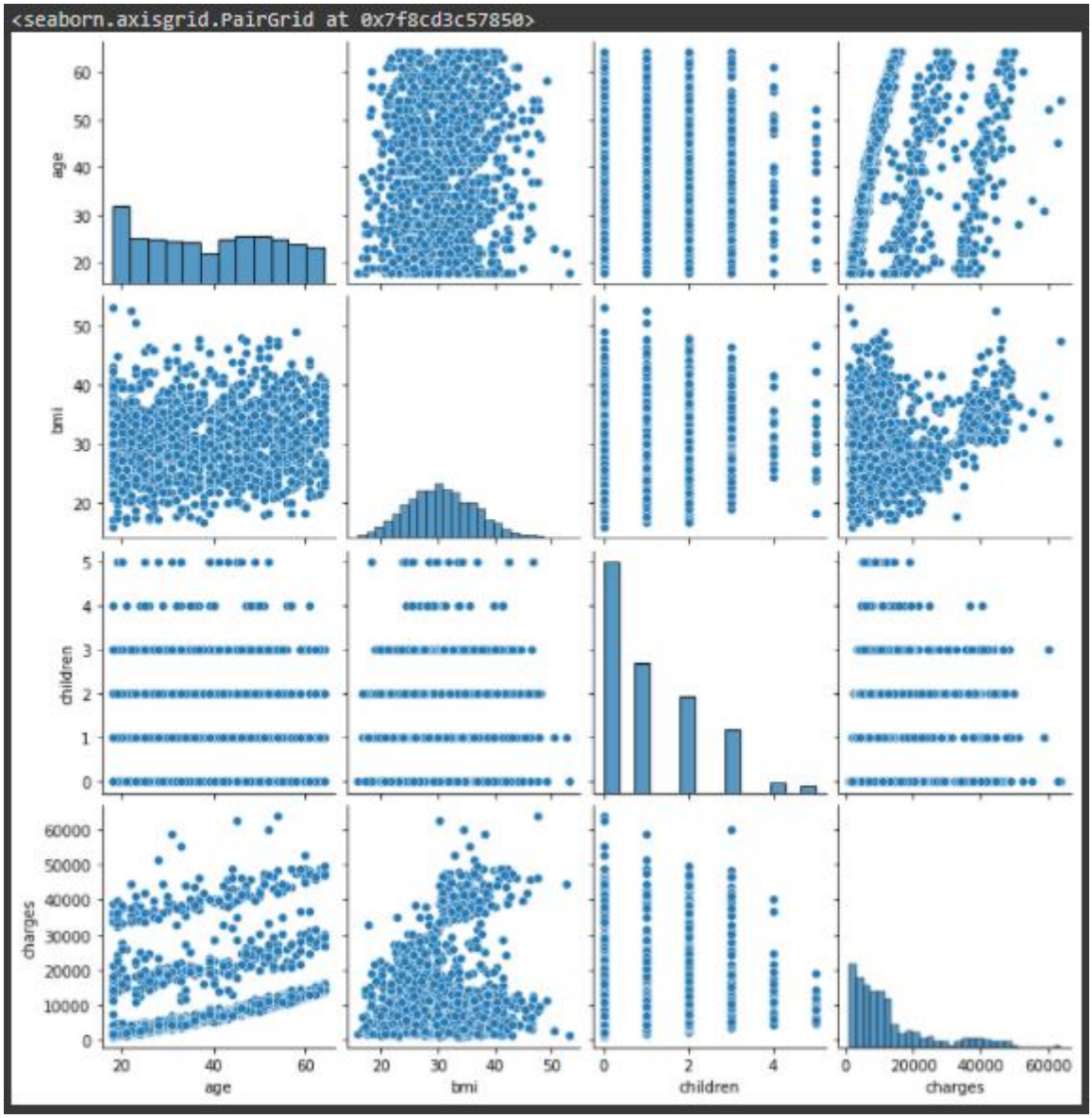
  - ## Fecha de aportación:

Última actualización: 2018-02-20
Fecha de creación: 2018-02-20

Nombre	Significado	Tipo	Dominio
age	Edad	Numérico	(18 – 64)
sex	Sexo	Categórico	{male, female}
bmi	IMC: Índice de masa Corporal	Numérico	(15.96 - 53.13)
children	Hijos	Categórico	{0, 1, 2, 3, 4, 5}
smoker	Fumador	Categórico	{yes, no}
region	Región donde vive el asegurado	Categórico	northeast northwest southeast southwest
charges	Prima del seguro	Numérico	(1121.8739 - 63770.428)

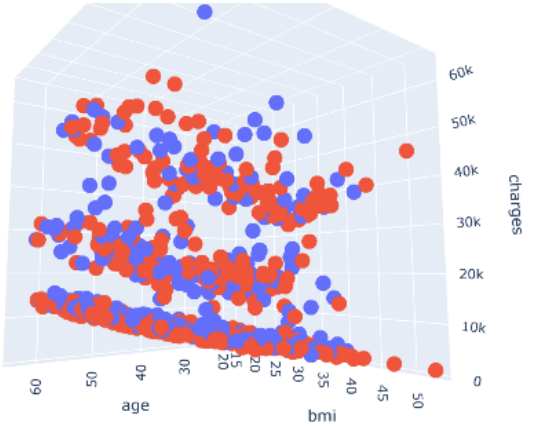
- **RESULTADOS**
  - **DIAGRAMA GENERADO**




<seaborn.axisgrid.PairGrid at 0x7f8cd3c57850>
age
bmi
children
charges
age
bmi
children
charges

Visualización de la gráfica de dispersión 3D para ver la relación entre Edad, IMC y Prima del seguro

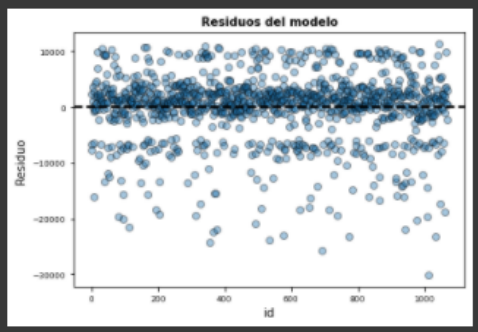
```
fig = px.scatter_3d(data, x='age', y='bmi', z='charges', color= 'sex')
fig.show()
```



A medida que aumenta la edad, las prima de seguro también aumentan. Esto verifica el coeficiente de correlación de 0.3 entre edad y cargos. IMC con cargos muestra la misma relación pero más débil.

Gráfica de residuales

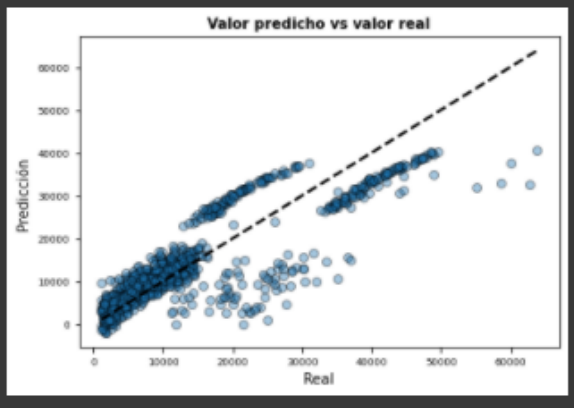
```
modelo  = sm.OLS(endog=y_train, exog=X_train,)
modelo  = modelo.fit()

#y_train = y_train.flatten()
prediccion_train = modelo.predict(exog = X_train)
residuos_train   = prediccion_train - y_train

plt.scatter(list(range(len(y_train))), residuos_train,edgecolors=(0, 0, 0), alpha = 0.4)
plt.axhline(y = 0, linestyle = '--', color = 'black', lw=2)
plt.title('Residuos del modelo', fontsize = 10, fontweight = "bold")
plt.xlabel('id')
plt.ylabel('Residuo')
plt.tick_params(labelsize = 7)
```



Los residuos no parecen distribuirse de forma aleatoria en torno a cero, sin mantener aproximadamente la misma variabilidad a lo largo del eje X. Este patrón apunta a una falta de homocedasticidad y de distribución normal.

Valor predicho vs valor real

```
[81] plt.scatter(y_train, prediccion_train, edgecolors=(0, 0, 0), alpha = 0.4)
     plt.plot([y_train.min(), y_train.max()], [y_train.min(), y_train.max()], 'k--', color = 'black', lw=2)
     plt.title('Valor predicho vs valor real', fontsize = 10, fontweight = "bold")
     plt.xlabel('Real')
     plt.ylabel('Predicción')
     plt.tick_params(labelsize = 7)
```



- **MEDIDAS OBTENIDAS**

Construyendo modelo de regresión

```
[75] ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1,4,5])], remainder='passthrough')
     X = np.array(ct.fit_transform(X))
```

Codificar las variables categóricas

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Entrena, prueba dividida 80/20

```
[86] regressor = LinearRegression()
     regressor.fit(X_train, y_train)

     LinearRegression()
```

Modelo de ajuste

```
[78] y_pred = regressor.predict(X_test)
     math.sqrt(mean_squared_error(y_test, y_pred))

     5641.626558850191
```

Usando el error cuadrático medio de la raíz, podemos determinar la precisión de este modelo (cuán preciso puede predecir este modelo). Según el resultado de la regresión, podemos ver que la raíz del error cuadrático medio es 5641,62. RMSE es la desviación estándar de la variación aleatoria (errores de predicción). Por lo tanto, podemos esperar que este modelo prediga los cargos dentro de una desviación estándar de 5641.62.

```
[79] r2_score(y_test, y_pred)
     print(1 - (1-regressor.score(X, y))*(len(y)-1)/(len(y)-X.shape[1]-1))

     0.7486872962661957
```

Podemos ver que el R2 es 0,799. Tenemos una regla de oro que cuando R2 es mayor que 0.7, indica que es un buen modelo. R2 es un número de variación sistemática sobre la variación total. Esto significa que el 79,9% de las variaciones son sistemáticas. Si queremos saber cuál es el porcentaje de variación aleatoria, podemos dejar 100-79,9 = 25,31%. Por tanto, el 20,1% son variaciones aleatorias, que es un número pequeño. Sin embargo, para evaluar la Regresión multivariable, necesitamos saber que al sumar x variables, R2 siempre aumenta. Por lo tanto, R2 ajustado proporciona una comparación de "manzanas con manzanas" de los modelos, que es 0,7469. Según la regla de oro, este sigue siendo un modelo muy razonable.

- **DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS RESULTADOS GENERADOS**

Podemos ver que el R2 es 0,799. Tenemos una regla de oro que cuando R2 es mayor que 0.7, indica que es un buen modelo. R2 es un número de variación sistemática sobre la variación total. Esto significa que el 79,9% de las variaciones son sistemáticas. Si queremos saber cuál es el porcentaje de variación aleatoria, podemos dejar 100-79,9 = 25,31%. Por tanto, el 20,1% son variaciones aleatorias, que es un número pequeño. Sin embargo, para evaluar la Regresión multivariable, necesitamos saber que al sumar x variables, R2 siempre aumenta. Por lo tanto, R2 ajustado proporciona una comparación de "manzanas con manzanas" de los modelos, que es 0,7469. Según la regla de oro, este sigue siendo un modelo muy razonable.

- **TIPO DE MUESTRA QUE UTILIZÓ PARA PRUEBA Y ENTRENAMIENTO**

Se utilizó la función train_test_split() con los paramámetros test_size = 0.2, random_state = 0, para la etapa de pruebas y entrenamiento se seleccionó de forma aleatoria las filas a utilizar para cada una.

• ANÁLISIS DE RESULTADOS

- **DESCRIBIR LOS RESULTADOS ENCONTRADOS PROPORCIONANDO LA EXPLICACIÓN DE ESTOS**

Usando el error cuadrático medio de la raíz, podemos determinar la precisión de este modelo (cuán preciso puede predecir este modelo). Según el resultado de la regresión, podemos ver que la raíz del error cuadrático medio es 5641,62. RMSE es la desviación estándar de la variación aleatoria (errores de predicción). Por lo tanto, podemos esperar que este modelo prediga los cargos dentro de una desviación estándar de 5641.62.

- **CONLUSIONES**

El análisis de Regresión Lineal nos da a entender que a medida que aumenta la edad, las prima de seguro también aumentan. Esto verifica el coeficiente de correlación de 0.3 entre edad y cargos. IMC con cargos muestra la misma relación pero más débil.

Además, los residuos no parecen distribuirse de forma aleatoria en torno a cero, sin mantener aproximadamente la misma variabilidad a lo largo del eje X. Este patrón apunta a una falta de homocedasticidad y de distribución normal.

Una vez hecha la prueba de normalidad, tanto Normalidad de los residuos Shapiro-Wilk test como Normalidad de los residuos D'Agostino's K-squared test; Ambos test muestran claras evidencias para rechazar la hipótesis de que los datos se distribuyen de forma normal (p-value << 0.01).

Dicho lo anterior, cuando no se cumple la condición de normalidad, estos valores no son fiables. Una mejor aproximación es recurrir a un test de permutación. Para ello, se simula la hipótesis nula de "no asociación entre la variable respuesta y todos predictores", intercambiando aleatoriamente la variable respuesta entre las observaciones.

- **ANEXO**
  - **NOMBREL DEL ARCHIVO**

Dataset: insurance.csv
Código: 3 - RegresionLineal.ipynb

Instituto Politécnico Nacional

Escuela Superior de Cómputo

REGRESIÓN LOGÍSTICA

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

- # INTRODUCCIÓN
  - ## MARCO TEÓRICO

La Regresión Logística es una técnica estadística multivariante que nos permite estimar la relación existente entre una variable dependiente no métrica, en particular dicotómica y un conjunto de variables independientes métricas o no métricas.

El Análisis de Regresión Logística tiene la misma estrategia que el Análisis de Regresión Lineal Múltiple, el cual se diferencia esencialmente del Análisis de Regresión Logística por que la variable dependiente es métrica; en la práctica el uso de ambas técnicas tiene mucha semejanza, aunque sus enfoques matemáticos son diferentes.

La variable dependiente o respuesta no es continua, sino discreta (generalmente toma valores 1,0). Las variables explicativas pueden ser cuantitativas o cualitativas; y la ecuación del modelo no es una función lineal de partida, sino exponencial; si bien, por sencilla transformación logarítmica, puede finalmente presentarse como una función lineal.

Así pues el modelo será útil en frecuentes situaciones prácticas de investigación en que la respuesta puede tomar únicamente dos valores: 1, presencia (con probabilidad p); y 0, ausencia (con probabilidad 1-p).

El modelo será de utilidad puesto que, muchas veces, el perfil de variables puede estar formado por caracteres cuantitativos y cualitativos; y se pretende hacer participar a todos ellos en una única ecuación conjunta.

  - ## BREVE DESCRIPCIÓN DEL ESTUDIO QUE SE REALIZÓ

Primeramente se realizó lo que es un análisis y visualización de los datos, para posteriormente llevar a cabo el modelo de predicción, y finalmente terminar con un análisis del desempeño de la predicción.

  - ## INTENCIÓN DE LA APLICACIÓN DE LA TÉCNICA

Intentar predecir si las personas compraron un producto publicitado, utilizando el método de regresión logística.

  - ## JUSTIFICACIÓN DE LA TÉCNICA A APLICAR

Debido a que buscamos encontrar un modelo de predicción de compra, es que la regresión logística nos funciona a la perfección para generar el modelo buscado, y además, nos otorga una precisión del 86%.

- # DICCIONARIO DE DATOS
  - ### Intención:

Este conjunto de datos incluye si las personas compran un producto en función de su edad, sexo y salario anual estimado. Intentaremos predecir si lo compraron utilizando el método de regresión logística.

  - ### Fuente de acceso:

https://www.kaggle.com/dragonheir/logistic-regression

  - ### Autor:

Ananya Nayan

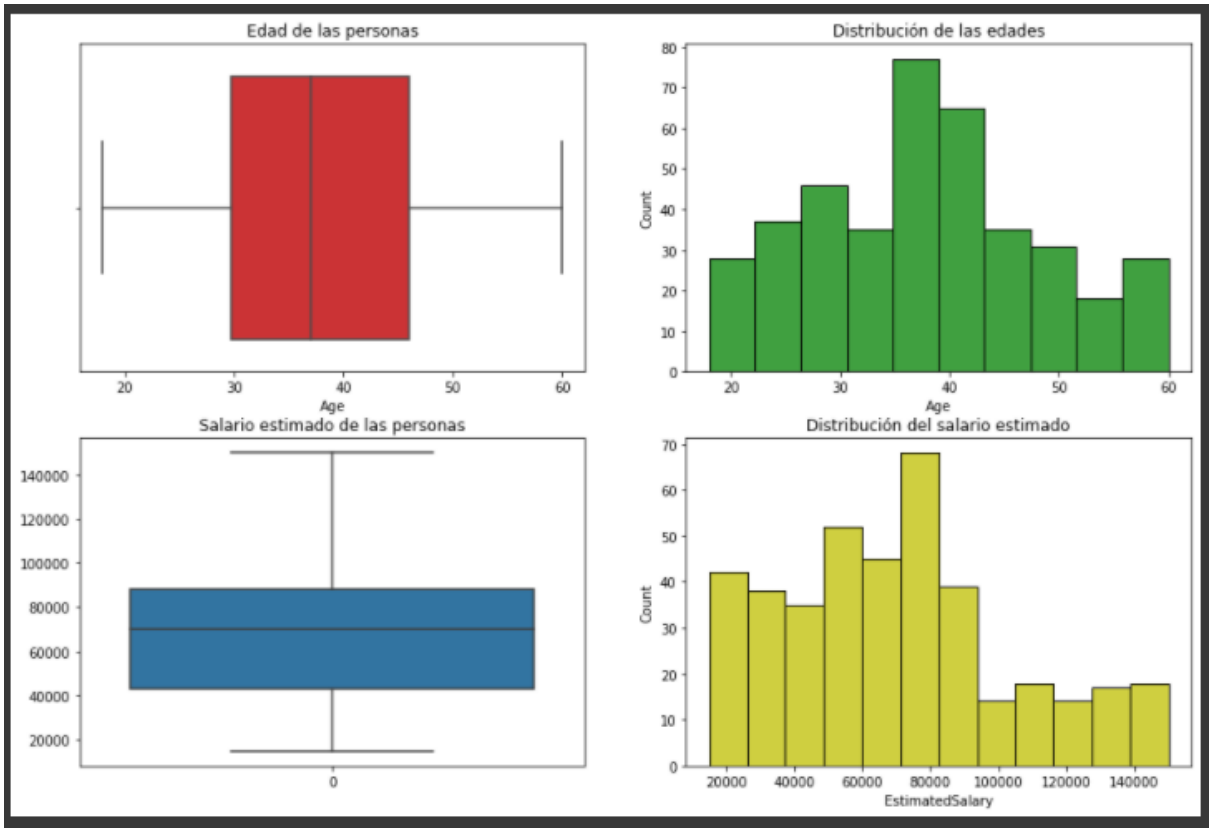
  - ### Fecha de aportación:

Última actualización: 2017-12-24

Fecha de creación: 2017-12-24

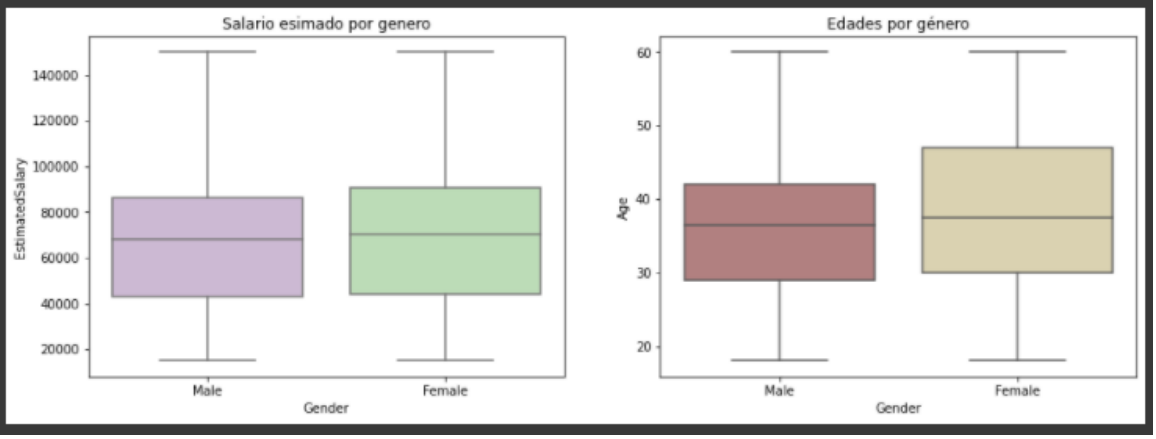
Nombre	Significado	Tipo	Dominio
User ID	Identificador de usuario	Numérico	¿?
Gender	Género	Categórico	[15,000 – 150,000]
Age	Edad	Numérico	[18 - 60]
EstimatedSalary	Salario estimado	Numérico	¿?
Purchased	Adquirido	Categórico	{0, 1}

- **RESULTADOS**
  - **DIAGRAMA GENERADO**



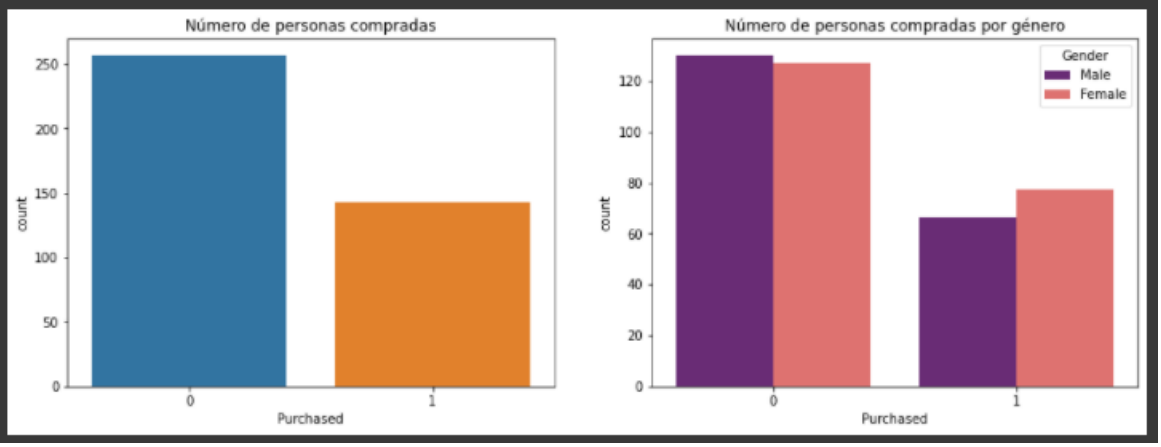
Diagramas de caja y bigote

```
[69] fig, axes = plt.subplots(1, 2, figsize=(15,5))
     sns.boxplot(ax=axes[0],x=df['Gender'], y=df['EstimatedSalary'], palette="PRGn")
     axes[0].set_title('Salario esimado por genero')

     sns.boxplot(ax=axes[1],x=df['Gender'], y=df['Age'], palette="pink")
     axes[1].set_title('Edades por género')
     plt.show()
```



Histogramas

```
fig ,axes = plt.subplots(1,2, figsize=(15,5))
sns.countplot(ax=axes[0],x='Purchased',data=df)
axes[0].set_title('Número de personas compradas')
sns.countplot(ax=axes[1],x='Purchased',hue='Gender',data=df,palette="magma")
axes[1].set_title('Número de personas compradas por género')
plt.show()
```

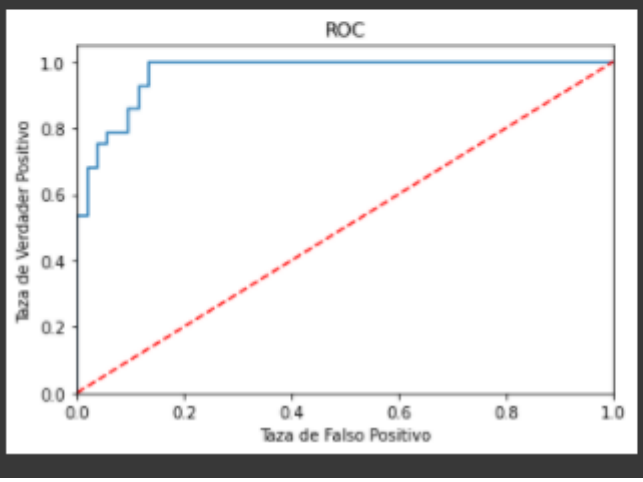


Mapa de calor

```
[72] f,ax = plt.subplots(figsize=(6, 5))

     sns.heatmap(df.corr(), annot=True, linewidths=0.5,linecolor="red", fmt= '.1f',cmap='viridis',ax=ax)
     plt.show()
```

Área bajo la curva

```
model_roc_auc = roc_auc_score(y_test, model.predict(X_test))

fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='AUC (area = %0.2f)' % model_roc_auc)
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim(([0.0, 1.0]))
plt.ylim(([0.0, 1.05]))
plt.xlabel('Taza de Falso Positivo')
plt.ylabel('Taza de Verdader Positivo')
plt.title('ROC')
plt.show()
```



- **MEDIDAS OBTENIDAS**

Escaliento de los datos

```
scaler = StandardScaler()
d_scaled = scaler.fit_transform(X)
data_scaled1 = pd.DataFrame(d_scaled)
data_scaled1.head()
```

	0	1	2
0	-1.020204	-1.781797	-1.490046
1	-1.020204	-0.253587	-1.460681
2	0.980196	-1.113206	-0.785290
3	0.980196	-1.017692	-0.374182
4	-1.020204	-1.781797	0.183751

```
[76] X_train,X_test,y_train,y_test = train_test_split(d_scaled,y,test_size=0.20,random_state=42)

     model = LogisticRegression(C=0.1,max_iter = 500)
     model.fit(X_train,y_train)

     y_pred = model.predict(X_test)

     # y = B + W*X1...
     print(f'Coeficiente de peso : {model.coef_}')
     print(f'Bias : {model.intercept_}')
```

```
Coeficiente de peso : [[-0.06210386  1.39043467  0.79306064]]
Bias : [-0.8686142]
```

Precisión

```
[77] print(f'Precisión de la prueba: {model.score(X_test,y_test)}')
     print(f'Precisión del entrenamiento: {model.score(X_train,y_train)}')
```

```
Precisión de la prueba: 0.8625
Precisión del entrenamiento: 0.8125
```

Reporte de clasificación

```
[78] print(classification_report(y_test, y_pred))
```

```
                precision    recall  f1-score   support

             0       0.85      0.96      0.90        52
             1       0.90      0.68      0.78        28

      accuracy                           0.86        80
     macro avg       0.88      0.82      0.84        80
  weighted avg       0.87      0.86      0.86        80
```

Matriz de confusión

```
[79] df = pd.DataFrame(confusion_matrix(y_test, y_pred),
                       columns = ['Positivo predictivo', 'Negativo predictivo'],
                       index=['Positivo verdadero', 'Negativo Verdadero'])
     df
     #plt.plot_confusion_matrix(y_test,y_pred,figsize=(7,7))
     #plt.show()
```

	Positivo predictivo	Negativo predictivo
Positivo verdadero	50	2
Negativo Verdadero	9	19

Medidas de la predicción

```
print("Exactitud:", accuracy_score(y_test,y_pred))
print("Precisión:", precision_score(y_test, y_pred, ))
print("Llamada:", recall_score(y_test,y_pred))
print("F1 Puntaje:", f1_score(y_test,y_pred))
```

```
Exactitud: 0.8625
Precisión: 0.9047619047619048
Llamada: 0.6785714285714286
F1 Puntaje: 0.7755102040816326
```

Finalmente, podemos utilizar el método SMOTE, que es un proceso de aumento de datos, para aumentar la tasa de aprendizaje.

```
[82] sm = SMOTE(random_state = 2)
     X_train_res, y_train_res = sm.fit_resample(X_train, y_train.ravel())

     clf = LogisticRegression()
     model_res = clf.fit(X_train_res, y_train_res)

     print(f'Prueba de exactitud {model_res.score(X_test,y_test)}')
```

```
Prueba de exactitud 0.9
```

Comparación del tamaño del conjunto de entrenamiento para el modelo

```
print(f'Originalmente: {X_train.shape}')
print(f'Aplicando el método SMOTE: {X_train_res.shape}')
```

```
Originalmente: (320, 3)
Aplicando el método SMOTE: (410, 3)
```

- **DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS RESULTADOS GENERADOS**

Podemos ver en las capturas que algunas de las propiedades del modelo de predicción son las siguientes:

- Exactitud: 0.8625
- Precisión: 0.9047619047619048
- Llamada: 0.6785714285714286
- F1 Puntaje: 0.7755102040816326

Además, si comparamos el tamaño de los conjuntos de entrenamiento que usamos originalmente y después con el método de SMOTE, podemos ver que es mayor el segundo:

- Originalmente: (320, 3)
- Aplicando el método SMOTE: (410, 3)

- **TIPO DE MUESTRA QUE UTILIZÓ PARA PRUEBA Y ENTRENAMIENTO**

Se hizo uso de las función train_test_split(), y recibió como parámetros los valores de test_size=0.20,random_state=42), lo que significa que parte del conjunto se utilizó para el entrenamiento y otra para la prueba del predictor, claramente de forma puramente aleatoria.

• ANÁLISIS DE RESULTADOS

- **DESCRIBIR LOS RESULTADOS ENCONTRADOS PROPORCIONANDO LA EXPLICACIÓN DE ESTOS**

Los resultados del cálculo de un modelo predictivo para saber si una persona adquirió un producto anunciado, utilizando como referencia su edad, su salario estimado y su género, no dieron lo siguientes valores:

- Coeficiente de peso : [[-0.06210386 1.39043467 0.79306064]]
- Bias : [-0.8686142]

Esto significa que si generamos una función usando estos coeficientes con la forma:

$$y = B + W * x1..$$

Fácilmente podremos predecir si una usuario va a comprar un producto que le sea anunciado, con una seguridad del 86%.

- **CONLUSIONES**

Pueden existir altos valores de riesgo relativo u junto a modelos no ajustados o bajos valores del coeficiente de determinación o ambos al mismo tiempo. Como consecuencia, si el profesional no tiene el cuidado al momento de mostrar la información, puede llegar a tener información dudosa. Sin mencionar que la forma y cambio en la selección de los parámetros es clave para lograr un nivel superior de precisión del modelo predictivo.

ANEXO

- **NOMBREL DEL ARCHIVO**

Dataset: Social_Network_Ads.csv
Código: 4 - RegresionLogistica.ipynb

Instituto Politécnico Nacional

Escuela Superior de Cómputo

REGLAS DE ASOCIACIÓN

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

- # INTRODUCCIÓN
  - ## MARCO TEÓRICO

Objetivo: Encontrar asociaciones o correlaciones entre los elementos u objetos de bases de datos transaccionales, relacionales o data-warehouses.
Las reglas de asociación tienen diversas aplicaciones como:

- Soporte para la toma de decisiones
- Diagnóstico y predicción de alarmas en ´telecomunicaciones
- Análisis de información de ventas
- Distribución de mercancías en tiendas
- Segmentación de clientes con base en patrones de compra

Son parecidas a las reglas de clasificación.
Se encuentran también usando un procedimiento de covering, sin embargo, en el lado derecho de las reglas, puede aparecer cualquier par o pares atributo-valor.
Para encontrar este tipo de reglas se debe de considerar cada posible combinación de pares atributo-valor del lado derecho.
Para posteriormente poderlas usando:

- Cobertura: número de instancias predichas correctamente
- Precisión: proporción del número de instancias a las cuales aplica la regla

  - ## BREVE DESCRIPCIÓN DEL ESTUDIO QUE SE REALIZÓ

Se llevó a cabo la creación de un catálogo de ofertas de productos en base a un análisis de canasta.

  - ## INTENCIÓN DE LA APLICACIÓN DE LA TÉCNICA

Sumergirse y buscar si hay alguna diferencia o correlación entre las canastas. Dado que el marco de datos ya está tabulado como un marco de datos activo, de inmediato usaremos el conjunto de datos que se analizará con el algoritmo APRIORI.

  - ## JUSTIFICACIÓN DE LA TÉCNICA A APLICAR

Debido a que buscamos generar las reglas de asociación, que nos van a permitir crear un catálogo de ofertas, es que aplicamos el algoritmo A PRIORI para alcanzar un nivel máximo de Soporte y Confianza.

- # DICCIONARIO DE DATOS
  - ### Intención:

Análisis de la cesta de mercado El algoritmo a priori lo dieron R. Agrawal y R. Srikant en 1994 para encontrar conjuntos de elementos frecuentes en un conjunto de datos para la regla de asociación booleana. El nombre del algoritmo es Apriori porque utiliza el conocimiento previo de las propiedades frecuentes del conjunto de elementos. Aplicamos un enfoque iterativo o búsqueda por niveles donde se utilizan k conjuntos de elementos frecuentes para encontrar k + 1 conjuntos de elementos.

  - ### Fuente de acceso:

https://www.kaggle.com/ahmtcnbs/datasets-for-appiori

  - ### Autor:

AHMET BAŞ

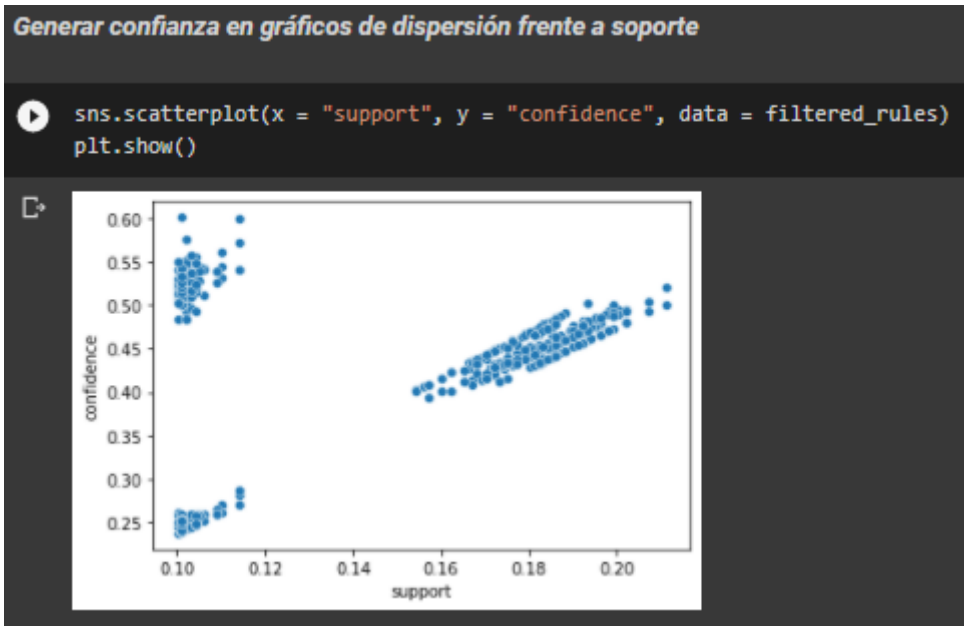
- **Fecha de aportación:**

Última actualización: 2021-05-28
Fecha de creación: 2021-05-28

Nombre	Significado	Tipo	Dominio
#	Identificador	Numérico	[0 - 998]
Apple	manzana	Nominal	{true, false}
Bread	Pan de molde	Nominal	{true, false}
Butter	Manteca	Nominal	{true, false}
Cheese	Queso	Nominal	{true, false}
Corn	Maíz	Nominal	{true, false}
Dill	eneldo	Nominal	{true, false}
Eggs	Huevos	Nominal	{true, false}
Ice cream	Helado	Nominal	{true, false}
Kidney Beans	Frijoles	Nominal	{true, false}
Milk	Leche	Nominal	{true, false}
Nutmeg	Nuez moscada	Nominal	{true, false}
Onion	Cebolla	Nominal	{true, false}
Sugar	Azúcar	Nominal	{true, false}
Unicorn	Unicornio	Nominal	{true, false}
Yogurt	Yogur	Nominal	{true, false}
chocolate	chocolate	Nominal	{true, false}

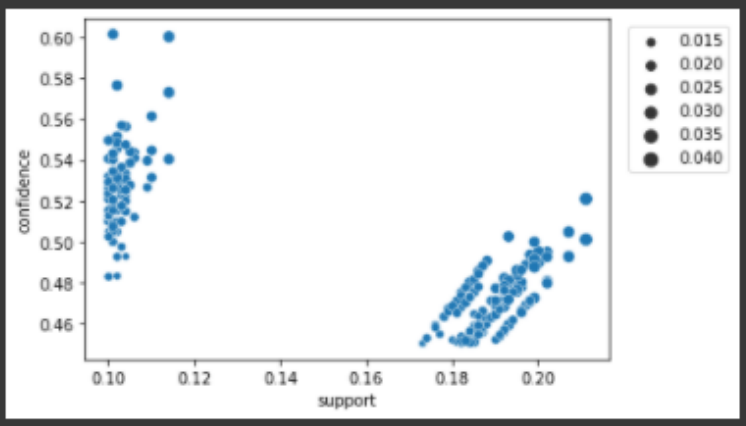
RESULTADOS

DIAGRAMA GENERADO



Generar confianza en gráficos de dispersión frente a soporte

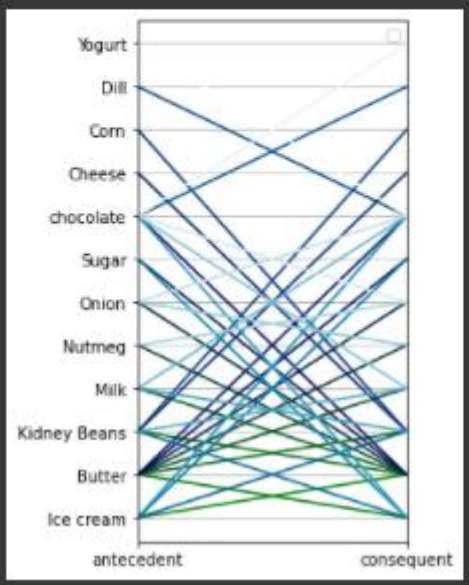
```
sns.scatterplot(x = "support", y = "confidence", size= 'leverage',data = filtered_rules)
plt.legend(bbox_to_anchor= (1.02, 1), loc='upper left',)
plt.show()
```



Agregar texto en negrita otra regla adicional donde admita más de 0.2 para un conjunto de elementos dado

Generar gráfico de coordenadas paralelas

```
plt.figure(figsize=(3,6))
parallel_coordinates(coords, 'rule',colormap = 'ocean')
plt.legend([])
plt.show()
```



- **MEDIDAS OBTENIDAS**

Calculo de todas las reglas de asociación para frecuent_itemsets

```
[21] rules = association_rules(frequent_itemsets,
                               metric = 'support',
                               min_threshold=0.1)
```

```
[22] filtered_rules = rules[(rules['antecedent support'] > 0.02)&
                            (rules['consequent support'] >0.01) &
                            (rules['confidence'] > 0.2) &
                            (rules['lift'] > 1.0)]
```

```
filtered_rules.sort_values('confidence',ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
402	(Unicorn, Dill)	(chocolate)	0.168168	0.421421	0.101101	0.601190	1.426578	0.030231	1.450764
390	(Milk, Dill)	(chocolate)	0.190190	0.421421	0.114114	0.600000	1.423753	0.033964	1.446446
326	(Dill, Cheese)	(Onion)	0.177177	0.403403	0.102102	0.576271	1.428523	0.030628	1.407968
392	(Dill, chocolate)	(Milk)	0.199199	0.405405	0.114114	0.572864	1.413065	0.033358	1.392051
258	(Ice cream, Kidney Beans)	(Butter)	0.196196	0.420420	0.110110	0.561224	1.334913	0.027625	1.320902
...	...	...	...	...	...	...	...	...	...
323	(Butter)	(Nutmeg, Yogurt)	0.420420	0.192192	0.100100	0.238095	1.238839	0.019299	1.060248
287	(Butter)	(Unicorn, Ice cream)	0.420420	0.185185	0.100100	0.238095	1.285714	0.022244	1.069444
371	(Yogurt)	(Corn, Kidney Beans)	0.420420	0.195195	0.100100	0.238095	1.219780	0.018036	1.056306
321	(Yogurt)	(Nutmeg, Butter)	0.420420	0.198198	0.100100	0.238095	1.201299	0.016774	1.052365
376	(chocolate)	(Corn, Kidney Beans)	0.421421	0.195195	0.100100	0.237530	1.216883	0.017841	1.055523

436 rows × 9 columns

Agregar texto en negrita otra regla adicional donde admita más de 0.2 para un conjunto de elementos dado

```
[27] filtered_rules = rules[(rules['antecedent support'] > 0.02)&
                            (rules['consequent support'] >0.01) &
                            (rules['confidence'] > 0.45) &
                            (rules['lift'] > 1.0)&
                            (rules['support']>0.195)]
     filtered_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
66	(Ice cream)	(Butter)	0.410410	0.420420	0.207207	0.504878	1.200889	0.034662	1.170579
67	(Butter)	(Ice cream)	0.420420	0.410410	0.207207	0.492857	1.200889	0.034662	1.162571
68	(Kidney Beans)	(Butter)	0.408408	0.420420	0.202202	0.495098	1.177626	0.030499	1.147905
69	(Butter)	(Kidney Beans)	0.420420	0.408408	0.202202	0.480952	1.177626	0.030499	1.139764
70	(Milk)	(Butter)	0.405405	0.420420	0.198198	0.488889	1.162857	0.027757	1.133960
71	(Butter)	(Milk)	0.420420	0.405405	0.198198	0.471429	1.162857	0.027757	1.124909
72	(Nutmeg)	(Butter)	0.401401	0.420420	0.198198	0.493766	1.174457	0.029441	1.144884
73	(Butter)	(Nutmeg)	0.420420	0.401401	0.198198	0.471429	1.174457	0.029441	1.132484
74	(Onion)	(Butter)	0.403403	0.420420	0.197197	0.488834	1.162726	0.027598	1.133838
75	(Butter)	(Onion)	0.420420	0.403403	0.197197	0.469048	1.162726	0.027598	1.123635
76	(Sugar)	(Butter)	0.409409	0.420420	0.196196	0.479218	1.139853	0.024072	1.112902
77	(Butter)	(Sugar)	0.420420	0.409409	0.196196	0.466667	1.139853	0.024072	1.107357
82	(chocolate)	(Butter)	0.421421	0.420420	0.202202	0.479810	1.141262	0.025028	1.114169
83	(Butter)	(chocolate)	0.420420	0.421421	0.202202	0.480952	1.141262	0.025028	1.114693
92	(Cheese)	(Kidney Beans)	0.404404	0.408408	0.200200	0.495050	1.212143	0.035038	1.171583

- **DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS RESULTADOS GENERADOS**

Para la tabla con los conjuntos de confianza se generaron 436 combinaciones de entre todos los productos seleccionados, en donde se aprecia todas las características posibles:

- Antecedentes
- Consecuentes

- Soporte del antecedente
- Soporte del consecuente
- Soporte
- Confianza
- Lift
- Apalancamiento
- Convicción

Breve introducción a los antecedentes del algoritmo Apriori. El algoritmo asume que cualquier subconjunto de un conjunto de elementos frecuentes debe ser frecuente. Digamos que en nuestros casos, donde {manzana, unicornio, yogur} es frecuente, entonces {manzana, yogur} es frecuente. Mientras que {manzana, unicornio} no es frecuente, entonces {manzana, unicornio, yogur} no es frecuente.

- APOYO = Una forma simple de controlar la complejidad es imponer una restricción que dichas reglas deben aplicar a un porcentaje mínimo de los datos. CONFIANZA = La probabilidad de que B ocurra cuando A; es p (B | A), que en asociación minera.
- LIFT = la co-ocurrencia de A y B es la probabilidad de que realmente veamos los dos juntos, en comparación con la probabilidad de que los veamos juntos si no estuvieran relacionados (independientes) entre sí.
- APALANCAMIENTO = alternativa es mirar la diferencia entre estas cantidades en lugar de su relación.
- CONVICCIÓN = medida para determinar la dirección de la regla. A diferencia de la elevación, la convicción es sensible a la dirección de la regla.

Solo el apoyo y la confianza como parámetro pueden ser engañosos para los artículos que son demasiado comunes / populares en la canasta. Es más probable que los artículos populares formen parte de la misma canasta solo porque son populares en lugar de cualquier otra cosa.
Establecemos el soporte mínimo en 0.06, el número máximo que se analiza en la canasta es 3. Estamos haciendo la primera poda y vemos qué obtenemos del resultado

o TIPO DE MUESTRA QUE UTILIZÓ PARA PRUEBA Y ENTRENAMIENTO

Se utilizó todo el conjunto de datos para la etapa de entrenamiento y la etapa de prueba, pero ya con las reglas de asociación generadas.

• ANÁLISIS DE RESULTADOS

o DESCRIBIR LOS RESULTADOS ENCONTRADOS PROPORCIONANDO LA EXPLICACIÓN DE ESTOS

Por ejemplo, la mantequilla se puede usar como venta cruzada con otros productos, también actúa como algo a ofrecer con antecedentes que es bajo. Así, es más probable que los clientes las compren si la mantequilla se les ofrece a un precio más económico si compran los antecedentes que se vendieron menos en una tienda.

o CONLUSIONES

Las reglas de asociación son bastante útiles a la hora de realizar un análisis de canasta, en este caso de productos de consumo regular.
Gracias a que nos generan una asociación entre los diferentes productos, nos es posible proponer estrategias de venta y acomodo de artículos dentro de una tienda.

• ANEXO

- **NOMBREL DEL ARCHIVO**

Dataset: basket_analysis.csv
Código: 5 - ReglasAsociacion.ipynb

Instituto Politécnico Nacional

Escuela Superior de Cómputo

ANÁLISIS DE COMPONENTES PRINCIPALES

Materia:

Data Mining

Grupo:

3CV19

Profesor:

Ocampo Botello Fabiola

Integrantes:

Castro Cruces Jorge Eduardo

Fecha:

Lunes, 27 de diciembre de 2021

- # INTRODUCCIÓN
  - ## MARCO TEÓRICO

Las relaciones se pueden interpretar como una medida del fenómeno bajo distintos puntos de vista. En un proceso estadístico que cuenta con un gran número de variables es difícil visualizar sus conexiones, al considerar muchas variables tendremos un número mayor de combinaciones representando los coeficientes de correlación.

Es importante reducir el número de variables para desechar información redundante y optimizar el proceso. El Análisis de Componentes Principales (ACP) propone la transformación a un nuevo conjunto sintético de variables (los componentes principales), que no están correlacionados y se encuentran ordenados de tal forma que los primeros conservan la mayor parte de la variación presente en todas las variables originales.

La técnica de ACP fue desarrollada por Pearson (1901) para luego ser retomada por Hotelling (1933) y posteriormente ser implementada con el impulso de las computadoras.

Componentes Principales:

El análisis de componentes principales (ACP) es una técnica estadística multivariante de simplificación, que permite transformar un conjunto de variables originales correlacionadas entre sí, en un conjunto sintético de variables no correlacionadas denominados factores o componentes principales.

En esta transformación no se establecen jerarquías entre variables y se elimina la información repetida (Jolliffe, 1986). Las nuevas variables son combinaciones linealmente independientes de las variables originales, ordenadas de acuerdo con la representación de dispersión respecto a la nube total de información recogida en las muestras.

  - ## BREVE DESCRIPCIÓN DEL ESTUDIO QUE SE REALIZÓ

Se aplicó el análisis de componentes principales para corregir el problema de la multicolinealidad

  - ## INTENCIÓN DE LA APLICACIÓN DE LA TÉCNICA

El análisis de componentes principales son métodos de reducción de datos que se utilizan para volver a expresar datos multivariados con menos dimensiones.

  - ## JUSTIFICACIÓN DE LA TÉCNICA A APLICAR

El objetivo de este método es reorientar los datos para que una multitud de variables originales se puedan resumir con relativamente pocos factores o componentes que capturen la máxima información posible de las variables originales.

- # DICCIONARIO DE DATOS
  - ## Intención:

Un conjunto de datos macroeconómicos que proporciona un ejemplo bien conocido de regresión altamente colineal. Este marco de datos consta de 6 variables económicas, observadas anualmente desde 1947-62:

1) defaltor del PNB defaltor de precios implícito del PNB (producto nacional bruto)
2) desempleados no. de desempleados
3) Fuerzas Armadas no. de personas en las fuerzas armadas
4) población población 'no institucionalizada'> = 14 años de edad
5) empleada número de personas empleadas.
6) Producto Nacional Bruto

- **Fuente de acceso:**

https://www.kaggle.com/dheeraj07/principal-component-analysis

- **Autor:**

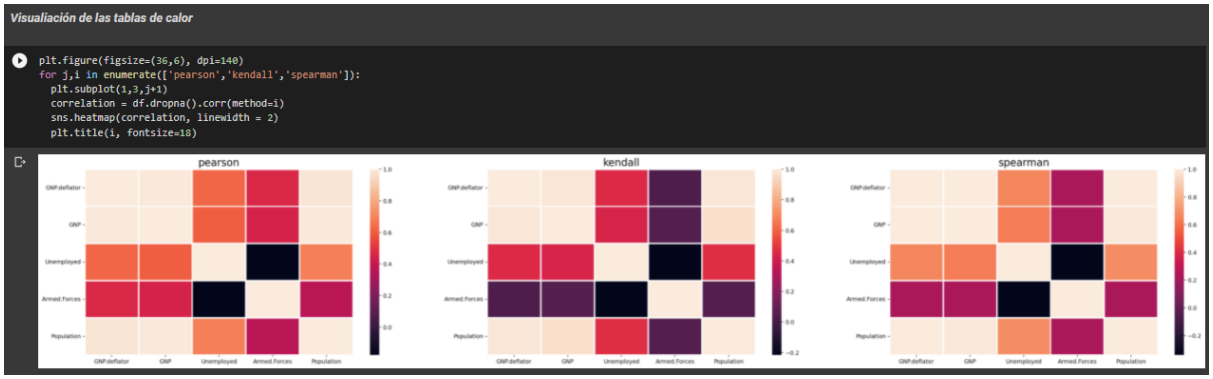
sopara0705

- **Fecha de aportación:**

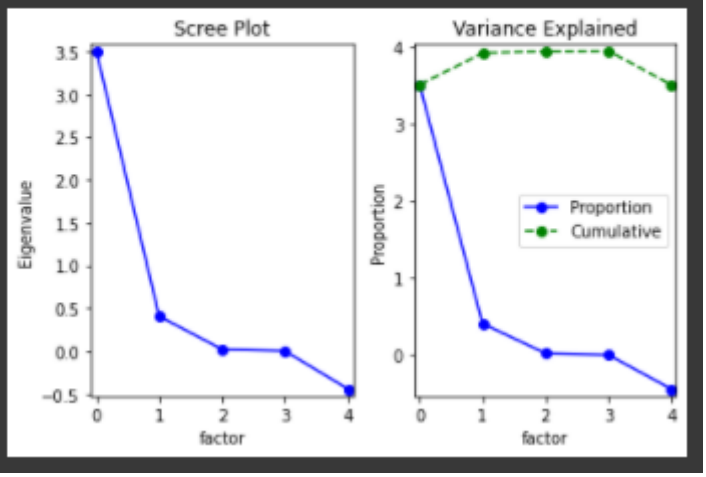
Última actualización: 2021-04-01
Fecha de creación: 2021-04-01

Nombre	Significado	Tipo	Dominio
GNP.deflator	defaltor del PNB defaltor de precios implícito del PNB (producto nacional bruto)	Numérico	[83 – 116.9]
GNP	Producto Nacional Bruto	Numérico	[234.289 - 554.894]
Unemployed	desempleados no. de desempleados	Numérico	[187 - 480.6]
Armed.Forces	Fuerzas Armadas no. de personas en las fuerzas armadas	Numérico	[145.6 - 359.4]
Population	población población 'no institucionalizada'> = 14 años de edad	Numérico	[107.608 - 130.081]
Employed	empleada número de personas empleadas.	Numérico	[60.171 - 70.551]

- **RESULTADOS**
  - **DIAGRAMA GENERADO**



```
model = Factor(df).fit()
model.plot_scree()
plt.show()
```




```
sns.heatmap(X_factors.corr())
plt.show()
```

- **MEDIDAS OBTENIDAS**

Visualiación de la tabla de correlación

```
[25] correlation = df.corr()
     correlation
```

	GNP.deflator	GNP	Unemployed	Armed.Forces	Population
GNP.deflator	1.000000	0.991589	0.620633	0.464744	0.979163
GNP	0.991589	1.000000	0.604261	0.446437	0.991090
Unemployed	0.620633	0.604261	1.000000	-0.177421	0.686552
Armed.Forces	0.464744	0.446437	-0.177421	1.000000	0.364416
Population	0.979163	0.991090	0.686552	0.364416	1.000000

Visualiación del resumen del modelo

```
model.summary()
```

```
/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py:1535: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=16
  "anyway, n=%i" % int(n))
```

OLS Regression Results

Dep. Variable:	Employed	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.981
Method:	Least Squares	F-statistic:	156.4
Date:	Mon, 27 Dec 2021	Prob (F-statistic):	3.70e-09
Time:	07:56:09	Log-Likelihood:	-7.3072
No. Observations:	16	AIC:	26.61
Df Residuals:	10	BIC:	31.25
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	92.4613	35.169	2.629	0.025	14.099	170.823
GNP.deflator	-0.0485	0.132	-0.366	0.722	-0.343	0.246
GNP	0.0720	0.032	2.269	0.047	0.001	0.143
Unemployed	-0.0040	0.004	-0.921	0.379	-0.014	0.006
Armed.Forces	-0.0056	0.003	-1.975	0.077	-0.012	0.001
Population	-0.4035	0.330	-1.222	0.250	-1.139	0.332

Omnibus:	1.572	Durbin-Watson:	1.248
Prob(Omnibus):	0.456	Jarque-Bera (JB):	0.642
Skew:	0.489	Prob(JB):	0.725
Kurtosis:	3.079	Cond. No.	1.74e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.74e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

```
[33] pc = PCA(df,
              ncomp=2,
              standardize=True,
              demean=True,
              normalize=False,
              gls=False,
              weights=None,
              missing=None)
```

```
[34] df_comp = pc.loadings.T
     df_comp
```

	GNP.deflator	GNP	Unemployed	Armed.Forces	Population
comp_0	0.521013	0.519909	0.365806	0.229642	0.521240
comp_1	-0.058090	-0.053455	0.595323	-0.798315	0.045299

```
X_factors = pc.factors
X_factors
```

	comp_0	comp_1
0	-3.204945	0.776652
1	-2.773855	0.877361
2	-2.104761	1.590765
3	-1.929308	1.318500
4	-1.279159	-1.276178
5	-0.893386	-1.985051
6	-0.649108	-1.973110
7	0.098944	-0.612634
8	0.059434	-0.716200
9	0.352178	-0.565773
10	0.827489	-0.443585
11	1.723864	0.891107
12	1.749443	0.399017
13	2.126404	0.515196
14	2.851089	1.021920
15	3.045675	0.182014

```
[36] correlation = X_factors.corr()
     correlation
```

	comp_0	comp_1
comp_0	1.000000e+00	-1.145391e-16
comp_1	-1.145391e-16	1.000000e+00

```
model = sm.OLS(Y,X_pca).fit()
model.summary()
```

```
/usr/local/lib/python3.7/dist-packages/scipy/stats/stats.py:1535: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=16
  "anyway, n=%i" % int(n))
```

OLS Regression Results

Dep. Variable:	Employed	R-squared:	0.919
Model:	OLS	Adj. R-squared:	0.906
Method:	Least Squares	F-statistic:	73.66
Date:	Mon, 27 Dec 2021	Prob (F-statistic):	8.10e-08
Time:	07:56:10	Log-Likelihood:	-22.188
No. Observations:	16	AIC:	50.38
Df Residuals:	13	BIC:	52.69
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	65.3170	0.269	243.206	0.000	64.737	65.897
comp_0	1.7019	0.141	12.040	0.000	1.397	2.007
comp_1	-0.3802	0.248	-1.535	0.149	-0.915	0.155

Omnibus:	0.244	Durbin-Watson:	1.933
Prob(Omnibus):	0.885	Jarque-Bera (JB):	0.401
Skew:	-0.211	Prob(JB):	0.818
Kurtosis:	2.349	Cond. No.	1.90

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

- **DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS RESULTADOS GENERADOS**

Dado que, haremos PCA en los datos para reducir las dimensiones, sigamos adelante y eliminemos la variable objetivo "employed".

```
[29] X = sm.add_constant(X)
```

Esto agrega el término constante beta0 a la regresión lineal múltiple.

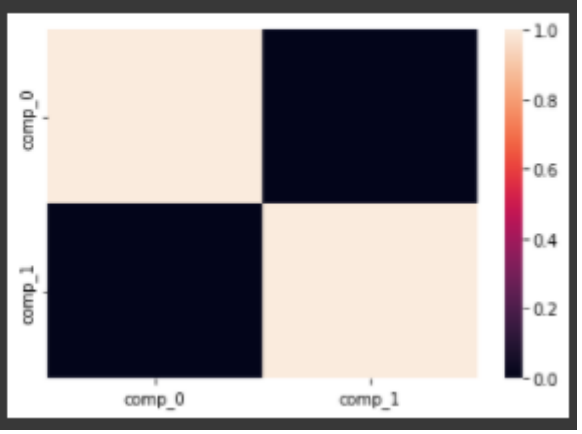
```
model.summary()
```

Notas:
[1] Los errores estándar suponen que la matriz de covarianza de los errores está correctamente especificada.
[2] El número de condición es grande, 1.74e + 05. Esto podría indicar que hay fuerte multicolinealidad u otros problemas numéricos.
Aquí podemos ver que tenemos una situación de "R ^ 2 alta pero pocas razones t significativas, lo que claramente nos dice que tenemos una situación de multicolinealidad y esto está violando nuestros supuestos de" No multicolinealidad "de regresión MCO.

ANÁLISIS DE RESULTADOS

- **DESCRIBIR LOS RESULTADOS ENCONTRADOS PROPORCIONANDO LA EXPLICACIÓN DE ESTOS**

```
sns.heatmap(X_factors.corr())
plt.show()
```



Podemos ver que al utilizar la técnica PCA hemos eliminado por completo el problema de la multicolinealidad.

- **CONLUSIONES**

El análisis de componentes principales son métodos de reducción de datos que se utilizan para volver a expresar datos multivariados con menos dimensiones. El objetivo de este método es reorientar los datos para que una multitud de variables originales se puedan resumir con relativamente pocos factores o componentes que capturen la máxima información posible de las variables originales.

ANEXO

- **NOMBREL DEL ARCHIVO**

Dataset: Longley.csv
Código: 6 - AnalisisComponentesPrincipales.ipynb

- ## Referencias:

http://150.185.9.18/fondo_editorial/images/PDF/CSF/Los%20rboles%20de%20Decisin%2004.pdf
https://www.ecured.cu/Clustering
http://mate.dm.uba.ar/~meszre/apunte_regresion_lineal_szretter.pdfmate.dm.uba.ar
Logistic Regression Calculating Pagestatpages.info
Reglas de Asociacióncccc.inaoep.mx
Análisis de Componentes Principales (ACP) | Software estadístico Excelwww.xlstat.com