Laboratorio 1 Clustering BiciAlpes

Paola Campiño

Jairo Nicolás Gómez

Jesus Felipe Duque

Universidad de los Andes

2023-1

Inteligencia de Negocios

Link repositorio: https://github.com/jf-duque/Lab-1---Clustering

Entendimiento de los Datos

La etapa de entendimiento de los datos es vital porque nos indica si los datos que no están proporcionando son relevantes y suficientes para cumplir con los objetivos del negocio, que en el caso de BiciAlpes es hacer un análisis de la seguridad de las vías para proporcionar información a los usuarios y que estos se sientan más seguros para utilizar los servicios que BiciAlpes suministra.

En primer lugar, se mira la cantidad de filas y columnas que tienen los datos suministrados, que en nuestro caso se hizo una muestra de 10 registros al azar donde se evidencia que hay 15 columnas y 5338 registros. Luego, evidenciamos que al obtener información acerca de estos datos, la mayoría son de tipo int y unos cuantos tipos object. Solo hay dos casos que los datos son tipo float.En general se evidencia que hay completitud de los datos a excepción de aquellas columnas relacionadas con el día de la semana. También, es consistente pues todos los datos en él son precisos y se relacionan de forma coherente entre sí, viendo también que no hay columnas duplicadas

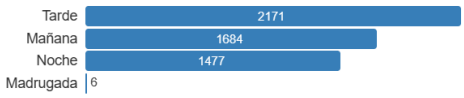
En segundo lugar, para tener un perfilamiento más completo de los datos se hace un análisis estadístico descriptivo de cada una de las columnas para luego mediante un diagrama de caja, poder evidenciar cuales son los datos atípicos para cada criterio y ver la distribución de los datos y ver su variabilidad. Evidenciamos que la mayoría de los datos tienden a tener los datos muy concentrados en el 1er, 2do y 3er cuartil, lo que nos indica que los datos tienden a tener pocos valores atípicos pero muy específicos que no representan la mayoría de los datos. Lo anterior se soporta en el hecho que la desviación estándar de todos los datos no pasa de un 3%, indicando que los datos están más agrupados alrededor de la media. En el caso de speed limit, se ve que hay datos que están muy por encima de la media con valores como 60 y 70 por lo que su desviación estándar es la más alta teniendo un 10%. Es importante mencionar que se hace la separación de tipos de dato entre in y float porque normalmente representan medidas totalmente distintas. Por ejemplo, speed limit se separa y se hace otra grafica aparte porque sus datos son muy diferentes al resto en términos de magnitud.

En tercer lugar, identificamos las variables categóricas y se hace un análisis separado para poder transformarlas y prepararlas para trabajar con ellas en los algoritmos. Se verifica entonces cuales datos son categóricos, los cuales fueron time, Day_of_week y Vehicle_Type. Luego, se verifica cuáles son los registros relacionados con la columna que más se repiten, por ejemplo, en Day_of_week, ver que la mayoría de los accidentes ocurren en día laboral o en Aciddent_Severity cual es tipo de accidente que más se repite el cual fue 3(leve).

Finalmente, se hace un reporte de cada una de las columnas indicando la cantidad de registros que cumplen con algún criterio en específico. A continuación, un ejemplo de lo realizado con cada una de las columnas

Time
Categorical

Distinct	4
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	41.8 KiB



Selección de datos

Para la implementación de cada uno de los algoritmos que se muestran en los siguientes puntos, como grupo acordamos que el propósito de hacer las agrupaciones sería verificar la relación entre las condiciones de pavimento de las vías [Road_Surface_Conditions] con la severidad de los accidentes [Accident_Severity]. De manera que con esta información BiciAlpes podría avisar a sus clientes que tan precavidos deberían ser en caso de encontrar ciertas condiciones de pavimento.

Cabe resaltar que la información de Accident_Severity se muestra con 3 valores:

- 1= Fatal
- 2=Serio
- 3= Leve

y la información de Road_Surface_Conditions se muestra con 3 valores:

- 1= Seca
- 2= Húmedad
- 3= Nieve
- 4 = Hielo
- 5= Inundadas
- 6= Inundadas
- 7= Fango
- -1= null/ missing del rango

Algo importante a resaltar es que los rangos en el análisis de datos se visualizaron que hay ciertos de pavimentos que no tienen registro de accidentes. Como lo son los que tienen inundaciones o fango:

```
In [70]: dt_biciAlpes['Road_Surface_Conditions'].value_counts()
         # se puede ver que la mayoría se da en vias secas.

Out[70]:  1    4260
          2     901
          4      87
         -1      81
          3       8
          5       1
         Name: Road_Surface_Conditions, dtype: int64
```

En cuanto a la severidad de los accidentes si no hay problema en cuanto a la selección de datos:

```
In [66]: dt_biciAlpes['Accident_Severity'].value_counts()
         # se puede ver que la severidad de los incidentes en su mayoría son de tipo 3= l

Out[66]: 3    3462
         2    1781
         1      95
         Name: Accident_Severity, dtype: int64
```

Aunque si es probable que no se hable mucho de accidentes fatales.

Modelamiento

K-means

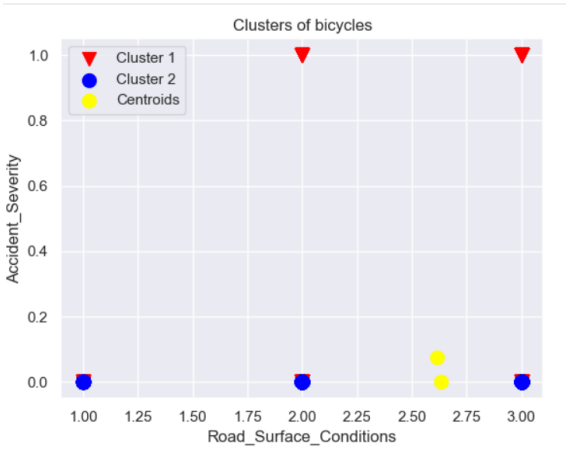
En primer lugar, se empieza haciendo una primera iteración para el modelo k-means con dos variables que son Road_Surface_Conditions y Accident_Severity. Lo anterior para empezar evaluando de a pocos dos segmentos y después si avanzar en el proceso.

El algoritmo k-means es iterativo con una serie de pasos que se repiten hasta que los datos sean coherentes para poder hacer un análisis. Los pasos son:

1. Definir un número de clústeres k para luego redistribuir los datos

2.Aleatoriamente seleccionado puntos de datos k para los centroides.

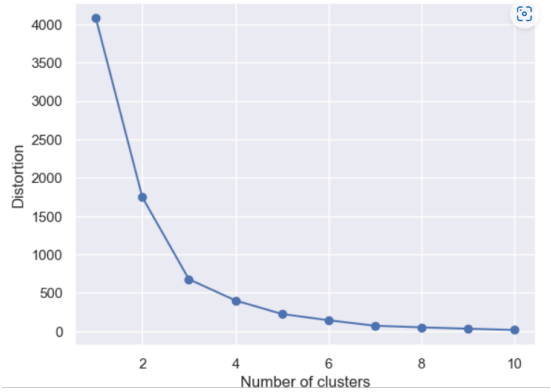
3.Se vuelve a iterar hasta que no exista cambio en los centroides.



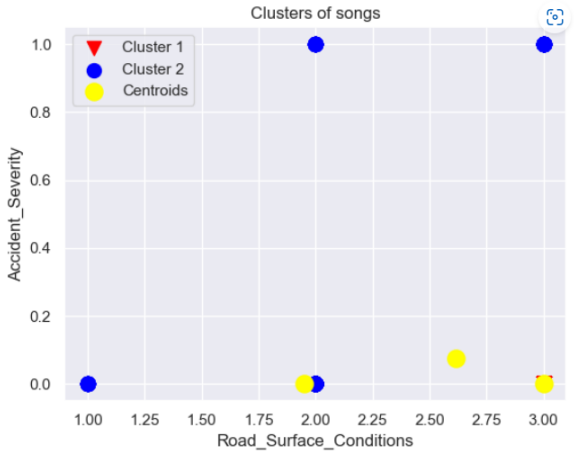
En esta primera iteración, los centroides no representan la ubicación de los clústeres. Por lo anterior hay que volver a iterar.

	Accident_Severity	Road_Surface_Conditions_-1	Road_Surface_Conditions_1	Road_Surface_Conditions_2	Road_Surface_Conditions_3	Road_Surface_Conditions_4
count	5338.000000	5338.000000	5338.000000	5338.000000	5338.000000	5338.000000
mean	0.815380	0.015174	0.798052	0.168790	0.001499	0.016298
std	0.259107	0.122257	0.401491	0.374601	0.038688	0.126632
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.500000	0.000000	1.000000	0.000000	0.000000	0.000000
50%	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
75%	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Se aprecia que los datos al tener una desviación estándar tan baja, su dispersión es muy baja Para evaluar el modelo, y poder hacer un mejor análisis, se hace el método del codo para sacar un k que nos sirva para nuestros datos.

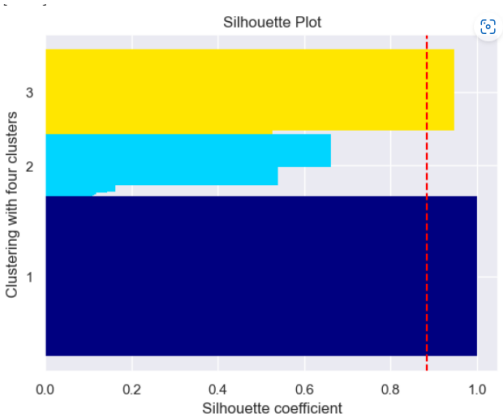


Como resultado no indica que 3 son los cluster ideales. Con lo anterior se vuelve a iterar, pero ahora con tres clusters. Ese seria nuestro hiperparametro indicado



Ahora los centroides están ubicados en los clústeres, es decir, convergen más en esta iteración hacia los centroides. Con la distancia euclídea calculamos la distancia de cada punto al centroide para asignarlos al clúster más cercano.

También se mide la calidad de los clústeres producidos con la silueta para saber si seguimos iterando o no.



Podemos apreciar en esta grafica que los clústeres 1 y 3 los puntos están bien ajustados a sus respectivos clústeres. El 2 indica un valor no tan alto y que los puntos no se encuentran tan bien ajustados, pero no tan bajo para tener que iterar otra vez

DBSCAN

El segundo algoritmo que se utilizó fue DBSCAN que es un algoritmo que define los clústeres a partir de la densidad local. Para la implementación se necesitan 2 parámetros que definirán la forma en la que se agruparan los datos, estos son: épsilon, que determina la distancia minima a la cual deben estar los puntos para considerarse vecinos y el número mínimo de puntos en una vecindad para considerarla un clúster. En este algoritmo existen 3 tipos de puntos:

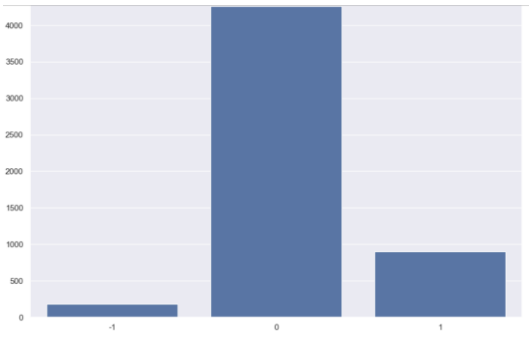
- Punto núcleo: un punto que tiene un número mínimo de puntos a su alrededor con distancia menor o igual a épsilon.
- Punto de borde: un punto que no tiene el número mínimo de puntos a su alrededor para considerarse núcleo, pero es vecino de uno que si lo es.
- Punto de ruido: Cualquier punto que no sea punto nucleo o de borde.

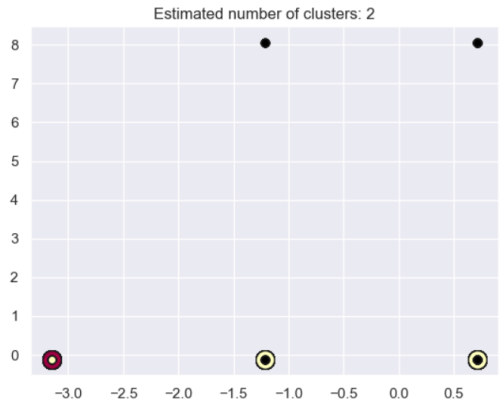
Los pasos del algoritmo serían:

- Tomar un punto arbitrario.
- Si tiene puntos mínimos dentro de E se inicia la formación de clúster, de lo contrario el punto se etiqueta como ruido.
- Si es un punto de núcleo todos los puntos dentro del vecindario E se agregan con su vecindario E si también son puntos núcleo
- Lo anterior se repite hasta que se completa el clúster conectado a la densidad
- El proceso anterior se repite con un nuevo punto que puede ser parte del nuevo clúster o un punto de ruido

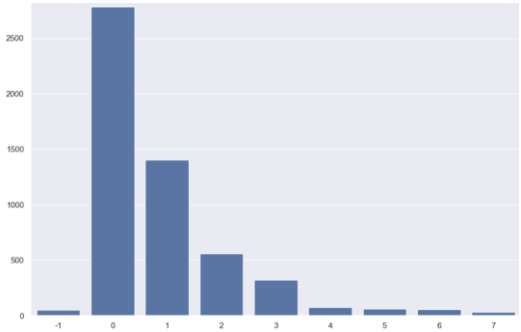
Se hicieron dos iteraciones con preparación de datos usando el oneHotEncoder para las columnas usadas ‘Accident Severity’ y ‘Road surface conditions’. Se obtuvieron las siguientes graficas:

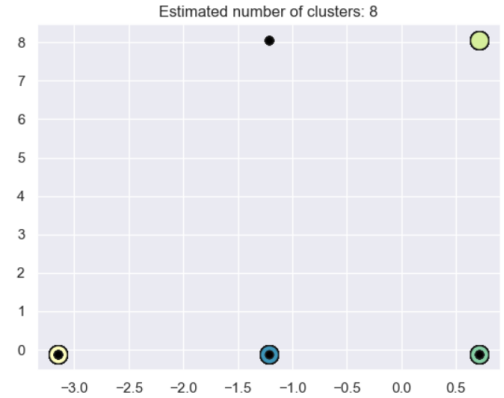
Iteración 1: se obtuvieron dos clústeres y 177 puntos de ruido





Iteración 2: se obtuvieron 8 clústeres y 49 puntos de ruido





A partir de ambas gráficas, notamos una tendencia de los datos a agruparse en el clúster 0 y notamos que, al disminuir el número mínimo de puntos para ser un clúster, se generan muchos más clústeres y por lo tanto quedan muchos menos puntos de ruido o puntos atípicos sin agrupar.

Clustering Spectral

Teoría:

Como tercer algoritmo de selección se optó usar spectral Clustering que es una técnica que tiene como objetivo asignar un grupo a cada dato que no tiene etiquetas al igual a como lo hace kmeans y DBSCAN. Sin embargo, a diferencia de los métodos convencionales spectral clustering, no asume que los cluster son esféricos, de hecho esta técnica es un poco más flexible ya que un cluster podría llegar a tener puntos que puedan llegar a estar más distantes pero conectados. Usualmente este algoritmo tiene sus implementaciones en segmentación de imagen y data mining entre otras aplicaciones.

La implementación de este algoritmo es sencilla por el hecho de usar Spectral Clustering de la libería sklearn, pues tan solo con especificar el número de clústeres y los datos a analizar. En terminos de algoritm este proceso tiene muchas implicaciones. A continuación, voy a presentar los pasos que debe seguir el algoritmo para poder generar las agrupaciones:

1. Crear una matriz de distancias.[1]
2. Transformar la matriz de distancias a una matriz de afinidad. [1]
3. Computar el grado de la matriz D y laplaciano de la matriz (L= D-A). [1]
4. Encontrar los valores propios (eigenvalues) y los vectores propios (eigenvectors) de laplaciano. [1]
5. Con los vectores propios de los K(numero de clusters) valores propios formar una matriz. [1]
6. Normalizar los vectores. [1]
7. Agrupar los datos en un espacio de k dimensiones. [1]

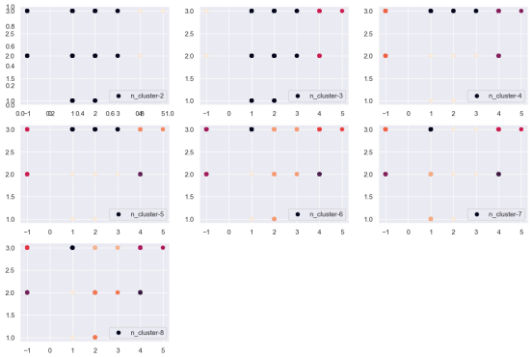
[1] Keerthana V. (2021) What, why and how of Spectral Clustering!, Analytics Vdhya. Recuperdo de : https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering/

Practica:

Para el caso en concreto llegó a hacer 2 modelos uno en el que no tiene datos, pero del que se puede sacar conclusiones importantes.

Modelo 1: Sin preparación de datos

el primero que no tiene preparación de datos. Realmente este modelo no tiene mucho problema si el enfoque es solo identificar patrones por ejemplo el hecho que se den registros de ciertos accidentes de tipo fatal en un tipo de piso en específico.

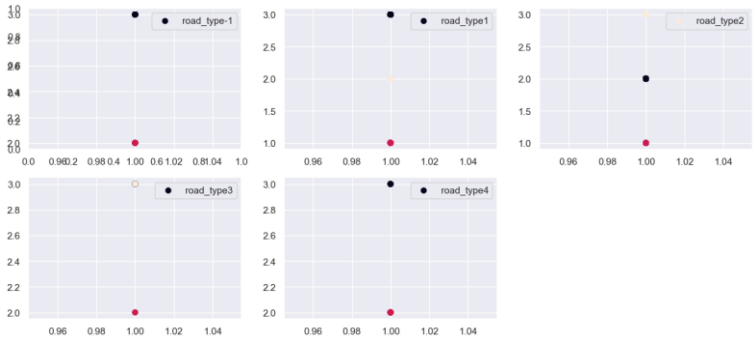


En si los clusters no generan una ayuda al momento de establecer una descripción generalizada de los datos, ya que como se mencionó anteriormente en los números que aparecen en las condiciones de condiciones de pavimento no tienen una jerarquía y como son valores no es muy fácil ver en donde se ubica la mayoría de los registros.

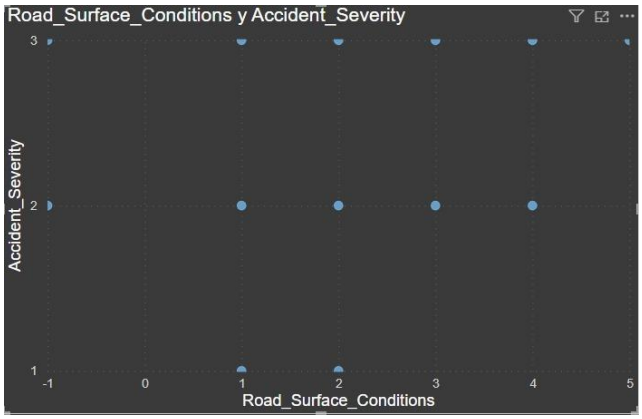
Modelo 2: Preparación de datos

Para la creación del segundo modelo se optó por preparar los datos, cosa que nos llevó a normalizar datos como lo son las condiciones de pavimento con oneHotEncoder usando la librería de pandas y la función get_dummies para así normalizar los datos. Y como acto final se filtrarían solo los valores que son igual a 1, ya que estos valores son aquellos hablan de los accidentes en los pavimentos especificados. De esta manera la selección de clusters sería mucho más sencilla siendo en base a las opciones de que tan grabe es el accidente y de esta manera sacar conclusiones mucho más exactas en cuanto a los peligros de las vías.

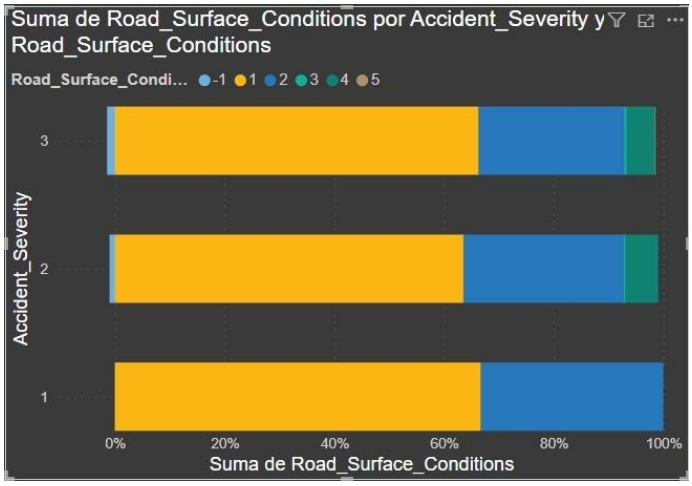
Con la visualización de los datos en base a este modelo fue mucho más efectivo ya que las conclusiones a sacar son mucho más claras, por ejemplo, el hecho de las vías de tipo 3 (nieve) no tienen accidentes de tipo fatal o al igual que lo pasa con las vías de tipo 4 (Hielo).

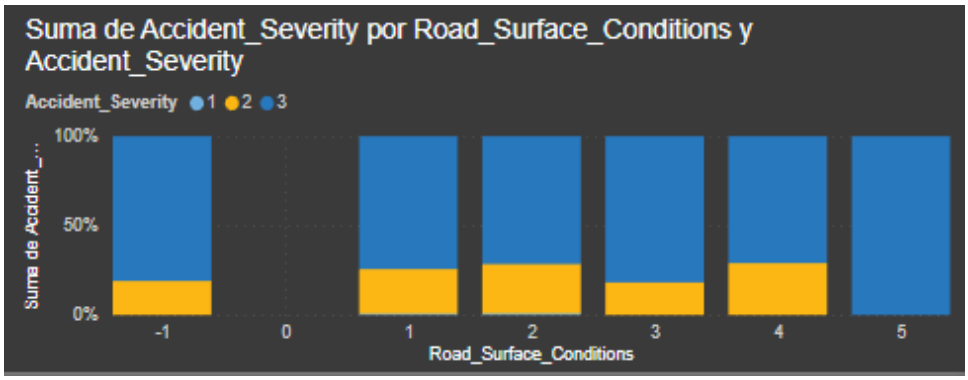


Visualización usando el tablero de control construido



Road_Surface_Conditions	Recuento de Accident_Severity
1	4260
2	901
4	87
-1	81
3	8
5	1
Total	**5338**


Suma de Road_Surface_Conditions por Accident_Severity y Road_Surface_Conditions
Road_Surface_Condi... -1 1 2 3 4 5
Accident_Severity
3
2
1
0%
20%
40%
60%
80%
100%
Suma de Road_Surface_Conditions


Suma de Accident_Severity por Road_Surface_Conditions y Accident_Severity
Accident_Severity 1 2 3
Suma de Accident_...
100%
50%
0%
-1
0
1
2
3
4
5
Road_Surface_Conditions