DIPLOMATURA EN CIENCIA DE DATOS, APRENDIZAJE AUTOMÁTICO Y SUS APLICACIONES.
COHORTE 2022 VIRTUAL

M08 – VÍCTIMAS DE INCIDENTES VIALES

Presentación final - Aprendizaje Automático

Mentora: Isabel Mejía

Grupo formado por:
* Giovine – Spitale - Venchiarutti

Aprendizaje Supervisado

Predecir la Gravedad de la Víctima: Heridos o Muertos por accidentes

Balanceo de los datos

```
Personas Heridas: Gravedad_victima    210131
dtype: int64
Personas Fallecidas: Gravedad_victima    1724
dtype: int64
```

- Notamos una gran diferencia entre víctimas heridas y muertas, los que podría afectar nuestra predicción
- Utilizamos la función RandomOverSampler, agrega registros de forma aleatoria para dejar una muestra simétrica

Selección de variables

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 211855 entries, 0 to 232869
Data columns (total 10 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Clase_incidente  211855 non-null  object
 1   Sexo             211855 non-null  object
 2   Edad             211855 non-null  float64
 3   Condicion        211855 non-null  object
 4   Mes              211855 non-null  object
 5   Dia              211855 non-null  object
 6   Num_dia          211855 non-null  float64
 7   Año              211855 non-null  int64
 8   Comuna           211855 non-null  object
 9   Barrio           211855 non-null  object
dtypes: float64(2), int64(1), object(7)
memory usage: 25.8+ MB
```

Preparación de los datos

- Eliminación de los registros nulos restantes, cerca del 9% del data set.
- Separación y codificación con GetDummies de la variable **Gravedad_victima** para usarla de dato objetivo.
- Codificación de las variables categóricas con LabelEncoder
- Estandarizar los datos con StandardScaler, lo que es bastante usual para modelos de aprendizaje supervisado

Separación de los datos

- Dividimos con la función Train_Test_Split el data set balanceado en tres partes
- Entrenamiento y Testeo para nuestro análisis predictivo aplicando diversos modelos
- Validación para determinar la efectividad del modelo elegido

Planteos de Modelos de Predicción

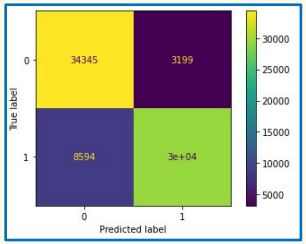
Super Vector Machine

- Ajuste con Kernel función **sigmoide**

```
Accuracy train sigmoid: 54.35%
Accuracy test sigmoid: 54.41%
              precision    recall  f1-score   support

           0       0.54      0.54      0.54     37544
           1       0.55      0.55      0.55     38103

    accuracy                           0.54     75647
   macro avg       0.54      0.54      0.54     75647
weighted avg       0.54      0.54      0.54     75647
```

- Ajuste con Kernel función **rbf**

```
Accuracy train rbf: 84.43%
Accuracy test rbf: 84.41%
              precision    recall  f1-score   support

           0       0.80      0.91      0.85     37544
           1       0.90      0.77      0.83     38103

    accuracy                           0.84     75647
   macro avg       0.85      0.84      0.84     75647
weighted avg       0.85      0.84      0.84     75647
```



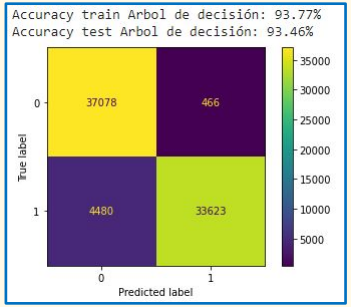
- Ajuste con Kernel función **poly**

```
Accuracy train poly: 70.04%
Accuracy test poly: 69.67%
              precision    recall  f1-score   support

           0       0.65      0.83      0.73     37544
           1       0.77      0.57      0.65     38103

    accuracy                           0.70     75647
   macro avg       0.71      0.70      0.69     75647
weighted avg       0.71      0.70      0.69     75647
```

Decision Tree ✔

- Ajuste de Hiper parámetros con la función GridSearchCV pero superando los 20 Depth llegando a 50/100 llega a responder muy bien a los datos de entrenamiento/validación(overfitting)

```
Accuracy train Arbol de decisión: 93.77%
Accuracy test Arbol de decisión: 93.46%
```



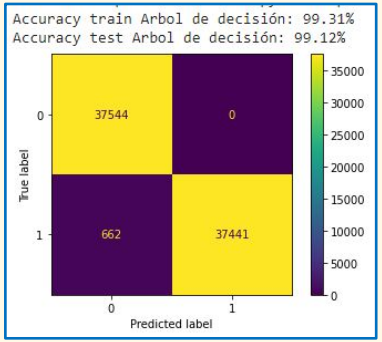

```
              precision    recall  f1-score   support

           0       0.89      0.99      0.94     37544
           1       0.99      0.88      0.93     38103

    accuracy                           0.93     75647
   macro avg       0.94      0.94      0.93     75647
weighted avg       0.94      0.93      0.93     75647
```

- Probamos realizar validación cruzada con la separación del data frame en 5 instancias aleatorias para saber si podíamos mejorar el % de precisión, pero sacamos un valor similar al anterior

```
DecisionTreeClassifier(max_depth=20, splitter='random')
Precisión test fold 0: 94.60
Precisión test fold 1: 93.64
Precisión test fold 2: 91.62
Precisión test fold 3: 94.08
Precisión test fold 4: 93.40
Avg. accuracy = 93.46801770679394
```

Planteos de Modelos de Predicción

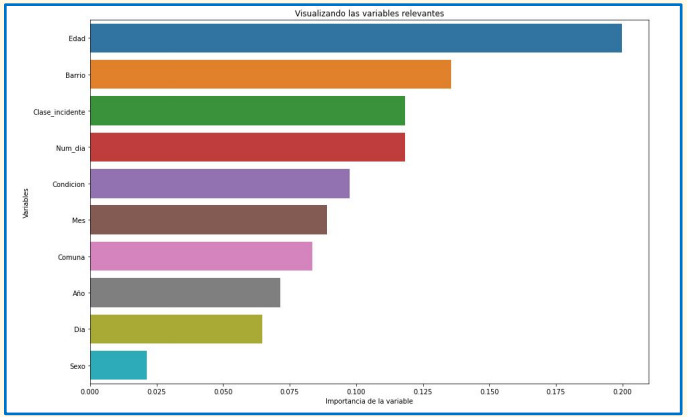
Random Forrest



- Aplicamos el modelo con 50 estimadores y teníamos un nivel de predicción muy alta pero luego con 20 también obtuvimos el mismo nivel de predicción.

	precision	recall	f1-score	support
0	0.98	1.00	0.99	37544
1	1.00	0.98	0.99	38103
accuracy			0.99	75647
macro avg	0.99	0.99	0.99	75647
weighted avg	0.99	0.99	0.99	75647

- Análisis de la relevancia de Variables que forman el data set de testeo



Edad	0.199862
Barrio	0.135774
Clase_incidente	0.118392
Num_dia	0.118385
Condicion	0.097566
Mes	0.089111
Comuna	0.083472
Año	0.071357
Dia	0.064641
Sexo	0.021440
dtype: float64	

Boosting

- Realizamos la separación del data frame en 5 instancias aleatorias de datos y probamos dos modelos de Booting

- Porcentaje promedio de precisión con modelo **XGBC**

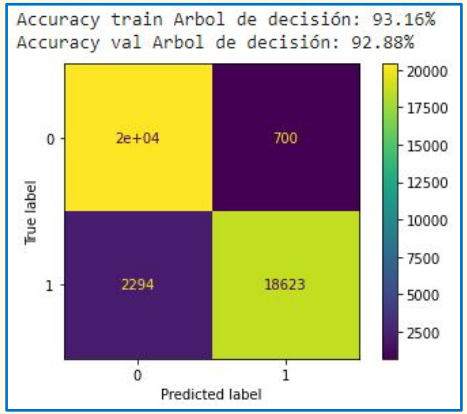
```
XGBC

Avg. accuracy = 71.74239028122223
```

- Porcentaje promedio de precisión con modelo **XGBRFC**

```
XGBRFC

Avg. accuracy = 70.46628356318439
```

Testeo de los Mejores Modelos de Predicción

Decision Tree

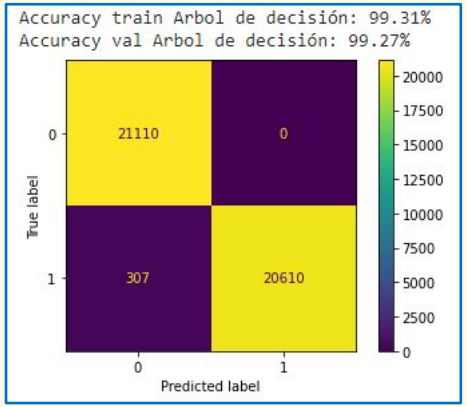


- Se definieron los dos mejores modelos por el % de precisión alcanzado usando el data set de test, con 42027 registros.
- En ambos casos se bajó la cantidad de niveles en árboles de decisión a 20 depth

- Revisando la cantidad de registros, podemos ver que este modelo logra un 93% de predicción

	precision	recall	f1-score	support
0	0.90	0.97	0.93	21110
1	0.96	0.89	0.93	20917
accuracy			0.93	42027
macro avg	0.93	0.93	0.93	42027
weighted avg	0.93	0.93	0.93	42027

Random Forrest

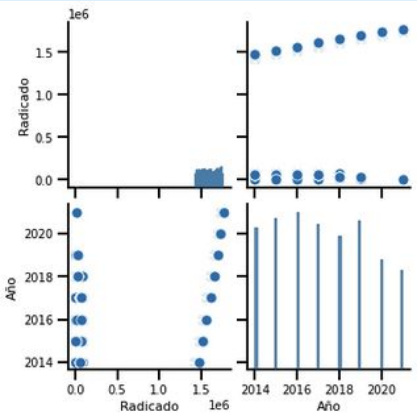


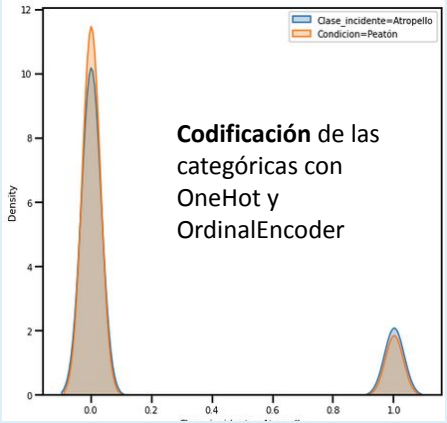
- El % alcanzado de predicción es de 100%. En casos de Muertos la predicción es completa y tenemos un pequeño margen de error para detectar los Heridos.

	precision	recall	f1-score	support
0	0.99	1.00	0.99	21110
1	1.00	0.99	0.99	20917
accuracy			0.99	42027
macro avg	0.99	0.99	0.99	42027
weighted avg	0.99	0.99	0.99	42027

Aprendizaje No Supervisado

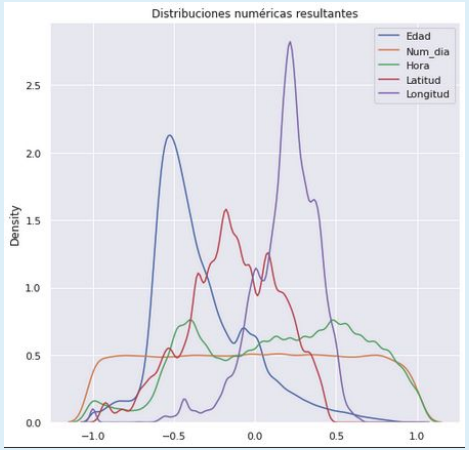
Análisis de correlación de todas las variables y selección





Codificación de las categóricas con OneHot y OrdinalEncoder

Escalado del dataset

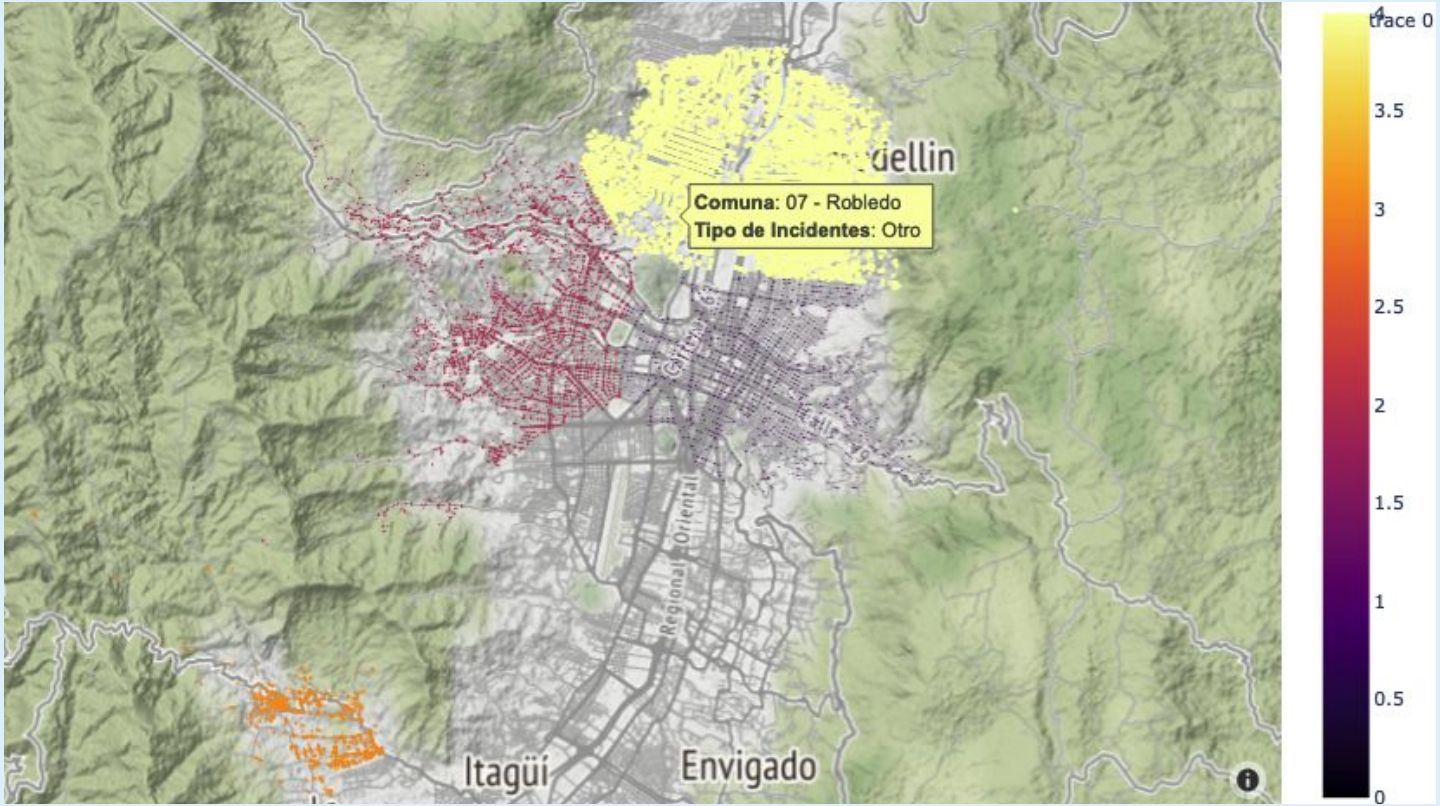


- Útil para al menos PCA, KNN, K-Means que usa distancia L2 o Eucleadiana, DB-Scan y GMM en muchos casos
- Usando las coordenadas no hace falta aplicar alguna normalización

Vemos que existe correlación bastante positiva entre:

- `Radicado` y `Año` (0.68)
- `Clase_incidente=Atropello` y `Condicion=Peatón` (0.8)
- `Barrio=Altavista Sector Central` y `Comuna=70 - Corregimiento de Altavista` (0.85)
- `Barrio=Cabecera San Antonio de Prado` y `Comuna=80 - Corregimiento de San Antonio de Prado` (0.85)

Mezcla de Gaussianas con Latitud y Longitud en subpoblación de personas heridas



El **cluster naranja** representa a la comuna 80 - Corregimiento de San Antonio de Prado.

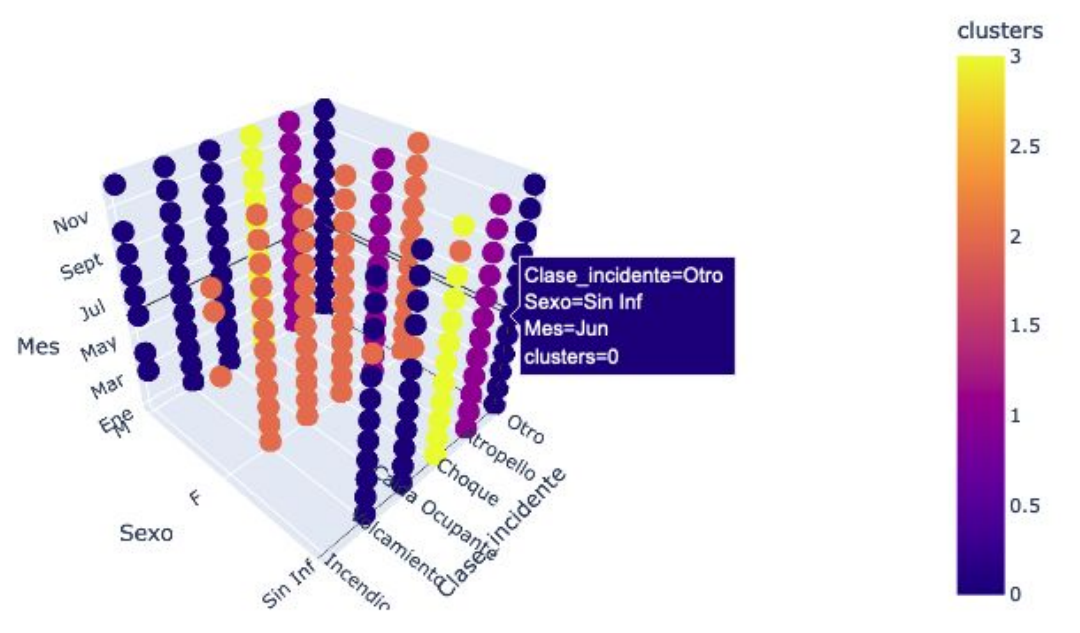
En los clusters restantes se podrían ver que pertenecen a cada uno distinto grupo de comunas.

Las clases de incidente, al menos a simple vista se ven presentadas en todos los clusters, salvo por Incendio que recordamos que tenía muy pocas ocurrencias.

N = 5

K-Means sin Gravedad de víctima

- Método del Codo -> **K=4**



- Choque está representado por el cluster 3.
- Incendio está representado por el cluster 0 y tiene mayoría en sexo Masculino seguido de sexo Femenino.

El **cluster 0:** se corresponde con el sexo Masculino y Sin Inf, para las clases de incidente Incendio, Volcamiento, Ocupante y Otro, para todos los meses salvo Mayo en Sin Inf para Volcamiento y Incendio.

El **cluster 1** se representa por Atropello para todos los sexos y meses.

El **cluster 2** se representa por sexo Femenino para todas las clases de incidentes menos Atropello, y para Sin Inf en Volcamiento, Caida_Ocupante y Choque.

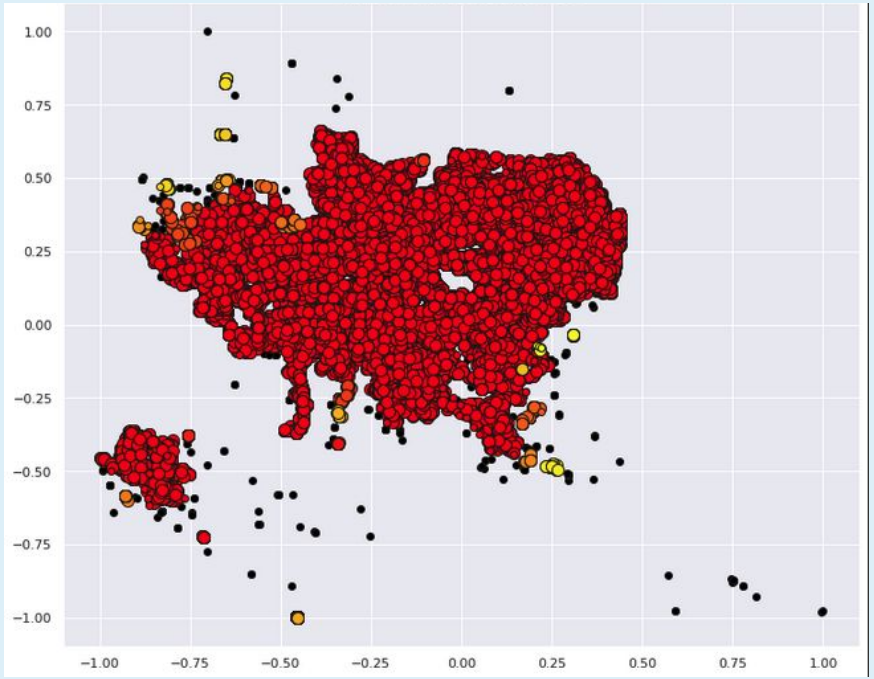
El **cluster 3** se forma en Choque para los sexos Femenino y Sin Inf en todos los meses salvo septiembre en el caso de Sin Inf.

DB-Scan con Latitud y Longitud

```
Cantidad de clusters estimados: 25
Cantidad de puntos con ruido estimados: 265
Coeficiente de Silhouette: -0.324
```

- Epsilon 0.2
- Mínimo de Vecinos 10

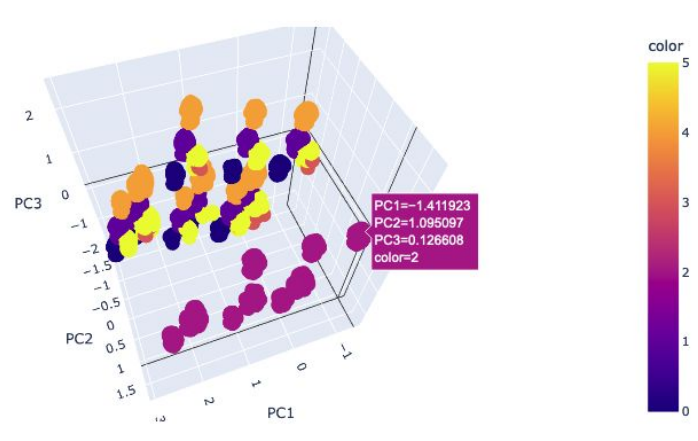
KNN para calcular el Epsilon.
Dio entre 0.01 y 0.03



Si bien el **Coeficiente de Silueta** es más cercano al 0 que al -1, es negativo, por lo cual concluimos, observando además el gráfico que se ha asignado de forma incorrecta ejemplos a los clusters.

Embedding PCA

- n=4 componentes



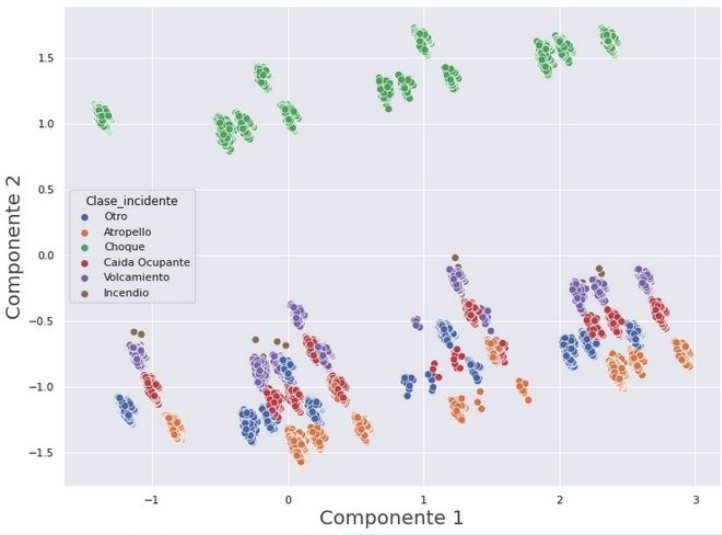
Primeras 3 CPs por Clase de incidente

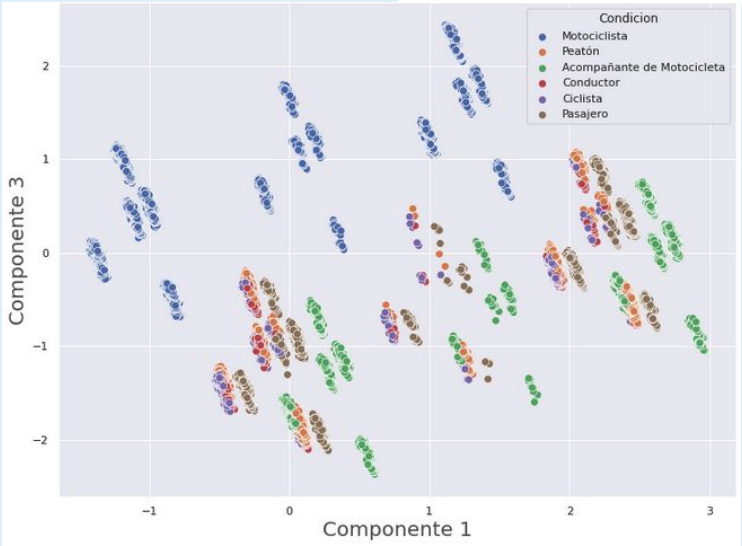
Clara separación de un grupo que en el gráfico de la CP1 y CP2 se ve con **Choque**.

Clase de Incidente CP1 y CP2

las víctimas de **Choque** bastante diferenciadas del resto del tipo de incidentes, siendo **Atropello** la más disímil a esta. El resto bastante agrupadas.

Incendio bastante cercana a **Volcamiento**. Ituímos que puede deberse a ser una consecuencia del **Volcamiento**.



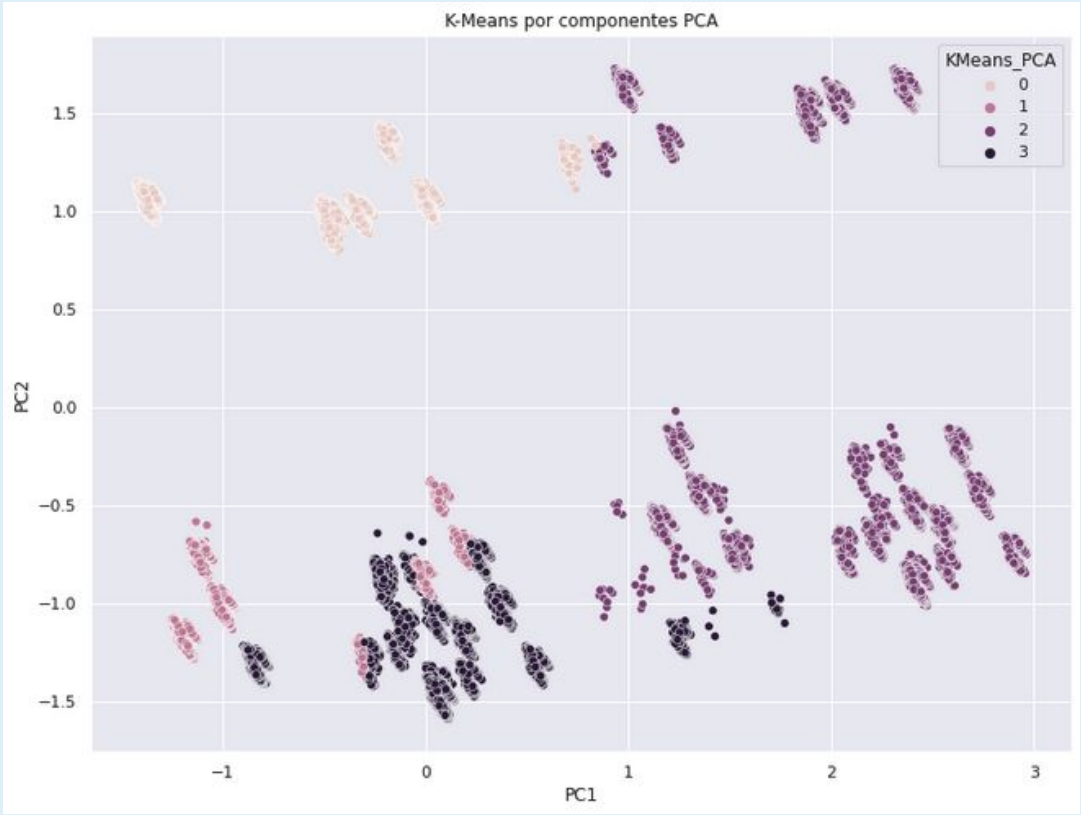


Condicion CP1 y CP3

Separación de **Motociclista** con el resto.

K-Means con PCA

- Método del Codo -> K=4



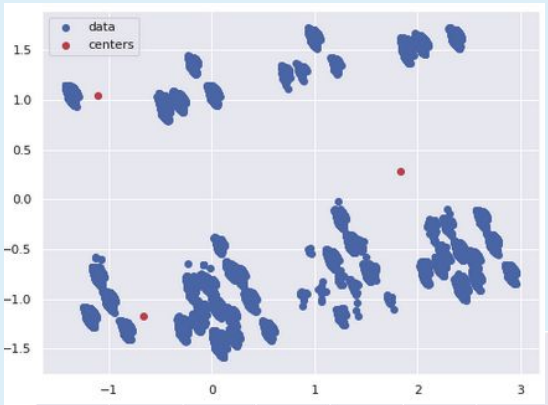
El **cluster 0** parte superior izquierda de CP1 y CP2 y no se mezcla con el resto de los clusters. En visualizaciones de PCA inciales, representa en el caso de Clase_incidente a la mitad de **Choque** y en Sexo al **Masculino**.

El **cluster 1** está bastante mezclado con el cluster 3 y muy alejado del resto.

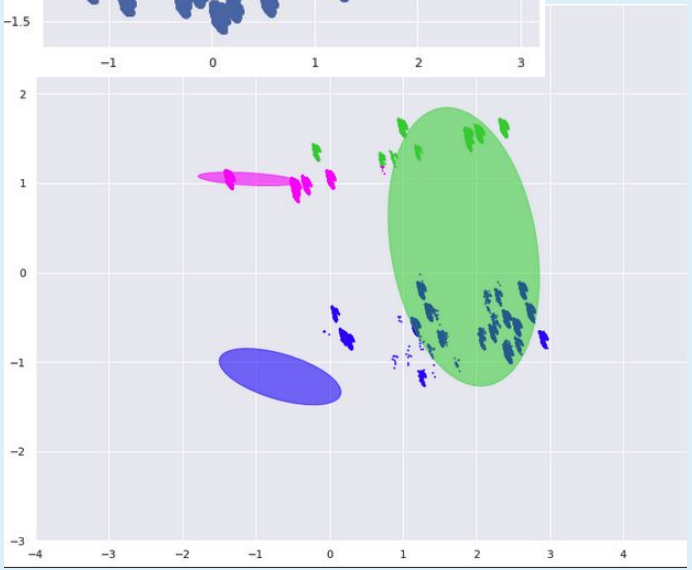
El **cluster 2** bastante separado del resto de clusters hacia la derecha. En visualizaciones de PCA en 2D anteriores, concluimos que representa a la mitad de Choque y los sexos Femenino y Sin Inf.

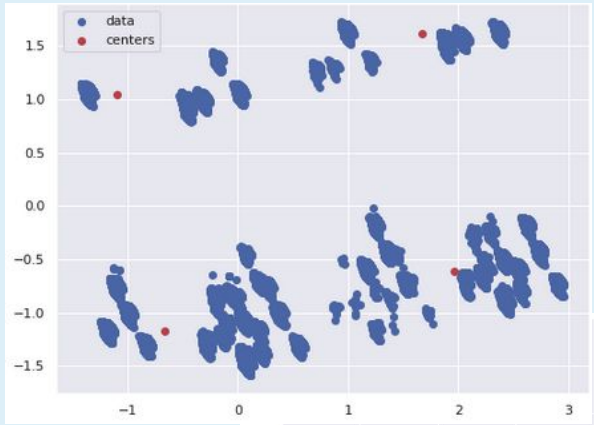
El **cluster 3** se define mayormente en la parte inferior de las componentes, por lo tanto **no** abarca a Choque ni al sexo Femenino.

Mezcla de Gaussianas con PCA

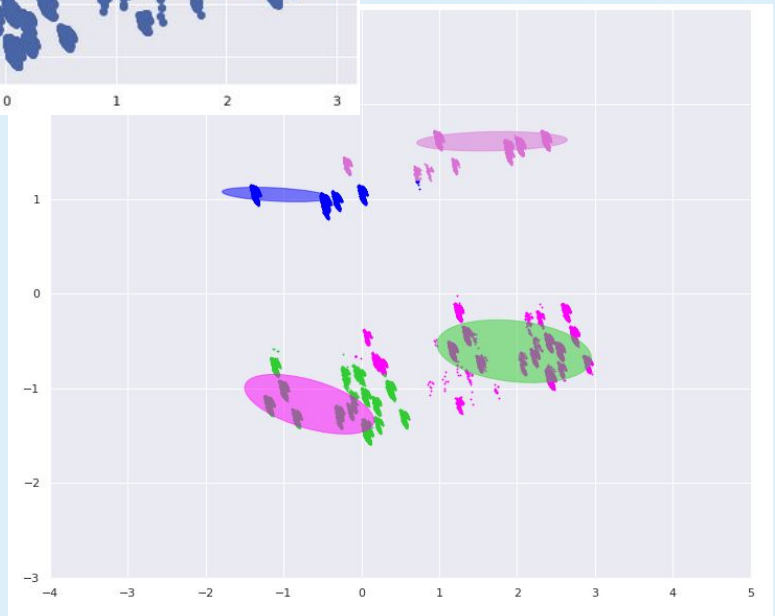


- N = 3



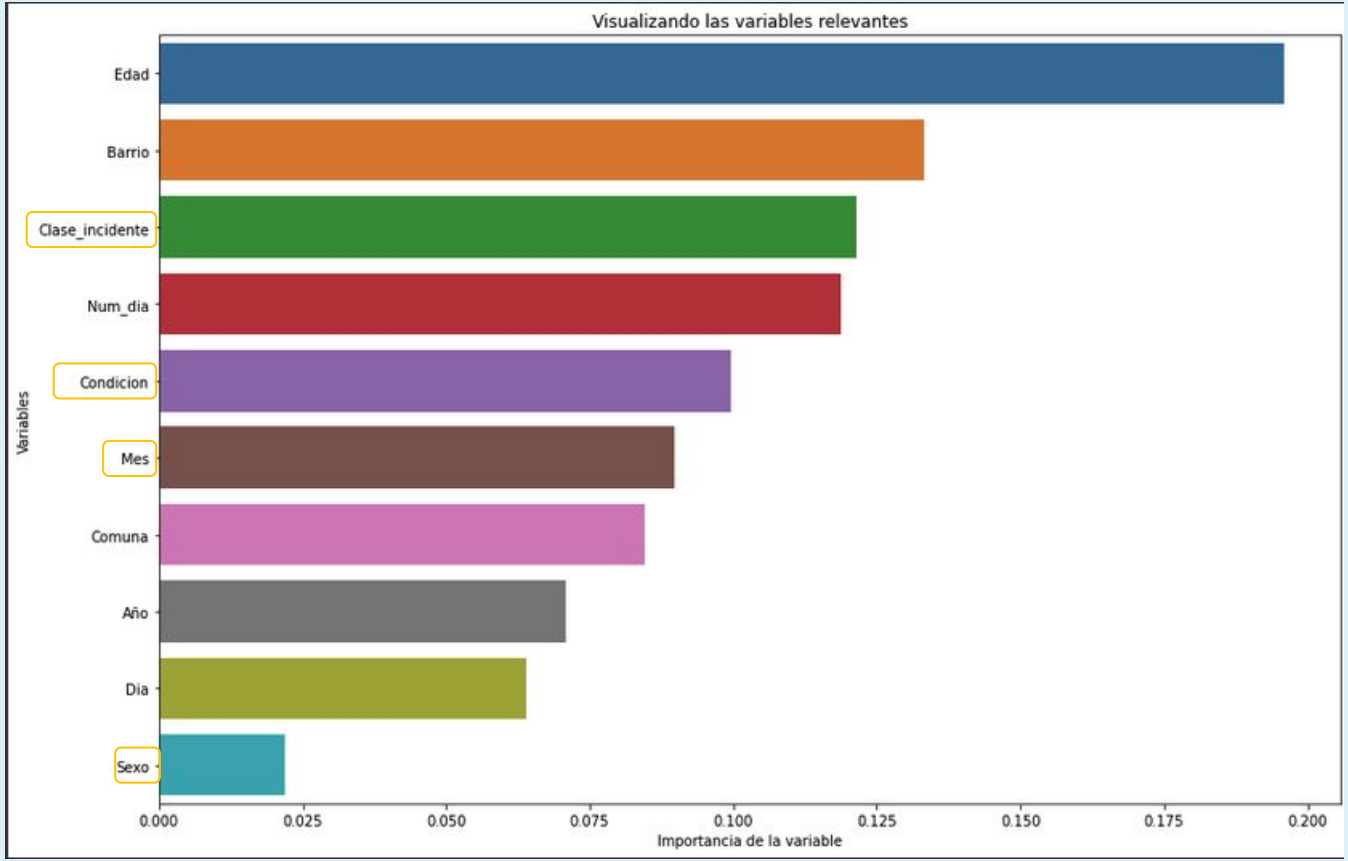


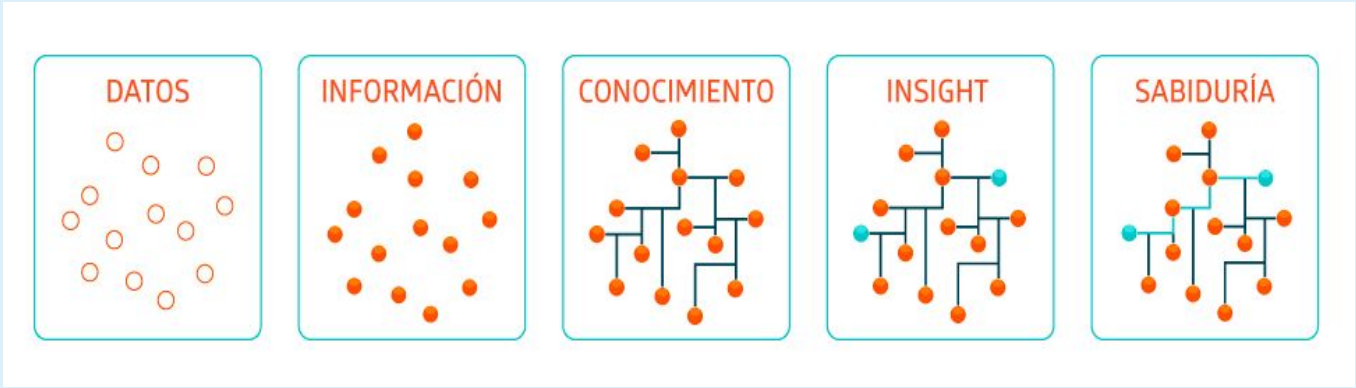
- N = 4



Comparando los gráficos de las gaussianas con N = 3 y N = 4, concluimos que el parámetro de más adecuado es N = 4.

Podemos concluir que los modelos que mejor se adaptaron a nuestro dataset de los solicitados fueron Mezcla de Gaussians y K-Means.





Trabajo completo de AS
Trabajo completo de ANS

Muchas Gracias.

Carina
Gustavo
Candela.