Estudio de víctimas por siniestros viales en la Ciudad Autónoma de Buenos Aires desde 2015 hasta 2018 y clasificación mediante la aplicación de aprendizaje supervisado

JUAN GABRIEL LIMACHI ZUÑAGUA Y DARÍO ADRIÁN ZABALJÁUREGUI[#]

ALUMNI CIENCIA DE DATOS – UTN FRBA – CLUSTER AI

2021

[#]A quien se debiera direccionar toda correspondencia: dzabaljuregui@frba.utn.edu.ar

ABSTRACTO

La seguridad vial en la ciudad es un aspecto importante en el control y prevención del malestar en los habitantes de la misma; año a año se están tomando decisiones de política pública para reducir los siniestros e invertir en nuevas formas de proteger los movimientos seguros. En este análisis partimos hacia la búsqueda de una interpretación definida de los datos de estos últimos años, con un objetivo de ayudar con algoritmos a visualizarla y predecir la clasificación de los siniestros según un parámetro; los resultados cumplen con el primer objetivo, pero no pueden afirmar el último.

PALABRAS CLAVE

Argentina; Buenos Aires; Data; Siniestro; Víctima; Análisis de Datos; Histograma; Correlación; Clasificación; Regularización; Machine Learning; Logistic Regression; KNN:K-Nearest Neighbors; SVM:Support Vector Machines.

01 INTRODUCCIÓN

Se registran en la Ciudad de Buenos Aires un aproximado de 100 víctimas fatales por siniestros viales, un número en crecimiento en esta última década. A partir de la década del 2010, se comenzaron a recopilar datos de sensores o identificadores en todas las áreas de la sociedad civil, y en los años 2016 y 2017 se crearon los Planes de Acción para un Gobierno Abierto, con la creación de múltiples datasets, portales y publicaciones de información con el fin de cumplir una agenda de transparencia y avanzar en la utilización de datos abiertos para llevar a cabo políticas públicas informadas. Dada su reciente implementación y uso, la investigación y aplicación de métodos y sus resultados han sido escasas.

02 OBJETIVO

En este análisis, nuestro objetivo fue cambiar este paradigma y utilizar este acceso a la información para asistir en las tomas de decisión de políticas públicas relacionadas en los Planes de Seguridad Vial de la Ciudad, los cuales buscan reducir los siniestros viales y las víctimas cada año. Particularmente, nos enfocamos en un análisis de los datos disponibles para entender mejor la coyuntura actual, y en la implementación de algoritmos que nos permitan clasificar —los siniestros ocurridos y por suceder— por el sexo *(no se disponía de distinción de género)* de las víctimas; esto último con el objetivo de predecir la ocurrencia de los accidentes involucrando personas masculinas o femeninas, y ver si hay distinciones fuera de lo común entre ellos.

03 DATASET

Buenos Aires Data fue el portal del cual obtuvimos el dataset más actual para la realización de este estudio. El mismo consistía en más de 33000 entradas de siniestros ocurridos entre los años 2015 y 2018 en formato .csv ó .xlsx, con 29 columnas portadoras de distintas características cada una como se puede ver en la Fig. 1 que nos muestra el 'head' de nuestro data-frame. Se contaba con muchas variables del tipo *string* y categóricas, siendo menores las variables numéricas continuas; así y todo, se decidió proseguir con el dataset hasta ver resultados preliminares. A simple vista, se consideró pertinente hacer unas modificaciones iniciales de nuestro dataset original.

FIG. 1

'HEAD' DEL DATASET ORIGINAL

Shape: (33234, 29)

	causa	rol	tipo	sexo	edad	mes	periodo	fecha	hora	lugar_hecho	...
0	homicidio	conductor	moto	NaN	NaN	2.0	2015	2/14/2015	19:00:00	cafayate y severo garcia grande de zequeira	...
1	homicidio	NaN	NaN	NaN	NaN	2.0	2015	2/25/2015	3:00:00	lugones, leopoldo av. y udaondo, guillermo av.	...
2	homicidio	peaton	peaton	femenino	NaN	2.0	2015	2/27/2015	8:00:00	avda jujuy y avda independencia	...

04 Pre-procesamiento

En primer lugar, como parte del pre-procesamiento de datos se hizo una limpieza de los mismos, eliminaron las filas o columnas que tenían valores NaN y/o nulos, y las columnas que se consideraron que no nos iban a servir ni para visualizar los datos ni para ajustar los modelos de clasificación. Las entradas, por consecuencia, bajaron a aproximadamente 30000.

05 Análisis Exploratorio de Datos (E.D.A.)

En lo que respecta a este apartado, después de realizar el pre-procesamiento, estudiamos los datos mas representativos que nos permitiesen alcanzar nuestro obejtivo. Como se dijo anteriormente, muchos eran variables de tipo categóricas, por lo cual se dificultaba que posteriormente pudiésemos implementar los modelos trabajados. Entonces, se procedió a seleccionar aquellas columnas que, en lo posible, nos brindaran datos de tipo continuo. Nuestro dataset quedó de la siguiente manera:

Fig. 2

'HEAD' DEL DATASET PROCESADO

Shape: (30570, 8)

	tipo	sexo	causa	x	y	cantidad_victimas	comuna	edad
6	moto	masculino	homicidio	-58.377362	-34.617451	1	1.0	18.0
7	automovil	masculino	lesiones	-58.469471	-34.629286	1	7.0	26.0
8	moto	masculino	homicidio	-58.528416	-34.650157	1	9.0	24.0

Siendo las variables de columnas con las que trabajamos:

- TIPO: que indica la modalidad del siniestro.
- SEXO: femenino o masculino.
- CAUSA: indica cómo se caratuló el siniestro.
- LATITUD (x) y LONGITUD (y): float
- CANTIDAD DE VICTIMAS: int
- COMUNA: int
- EDAD: int

05.01 Tipo de Siniestro

En un principio, se observó que existen 25 tipos únicos de siniestros. Para el estudio de nuestro caso, procedimos a visualizar cuáles eran los tipos de siniestros más recurrentes a los cuales las personas se encontraban expuestas en el territorio, lo cual quedó plasmado en la Fig. 3 y Fig. 4. Allí se observó que el top de los 5 siniestros más recurrentes, equivalían a cerca del 90% del total. Además, confirmamos que la mayor cantidad de siniestros eran ocasionados por motos, como lo menciona el Banco de Desarrollo de Latinoamérica (CAF).

Fig. 3

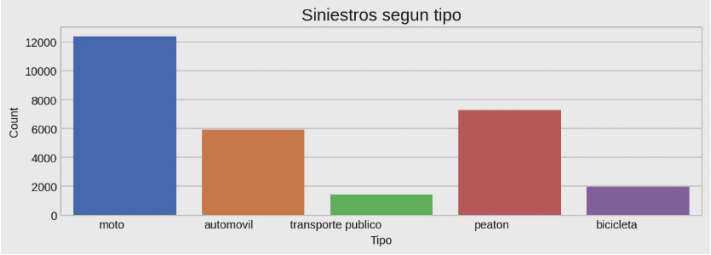
LOS 5 SINIESTROS DE MAYOR FRECUENCIA

Posición	Tipo	Cantidad
1°	Moto	12395
2°	Peatón	7301
3°	Automóvil	5930
4°	Bicicleta	1963
5°	Transporte público	1423

" Las motos son la primera causa de muerte en siniestros de tránsito en niños y adolescentes en América Latina". También apreciamos que los accidentes de peatones superaban a los de automóviles.

Fig. 4

SINIESTROS SEGÚN TIPO



05.02 Sexo de la persona

También se pudo verificar que entre los años 2015 y 2018, la cantidad de personas de sexo masculino que sufrieron siniestros, duplica a las persnas de sexo femenino (19760 hombres siniestrados contra 9252 mujeres).

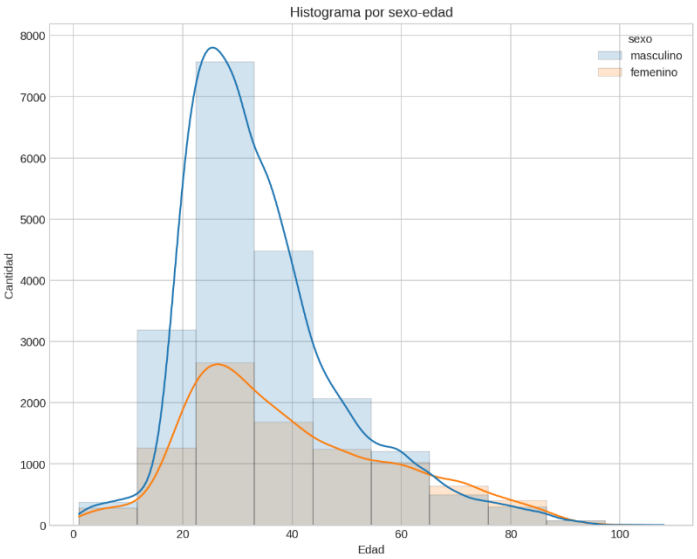
Fig. 5

SINIESTROS SEGÚN SEXO

En cuanto a la edad de las personas, determinamos que la media en ambos casos está entre los 20 y 35 años, lo cual no es casual ya que entre esos años las personas son más activas laboralmente y por ende están más expuestas. Visualizamos las frecuencias en Fig. 6:

FIG. 6

HISTOGRAMA DE FRECUENCIA DE SINIESTROS POR SEXO-EDAD

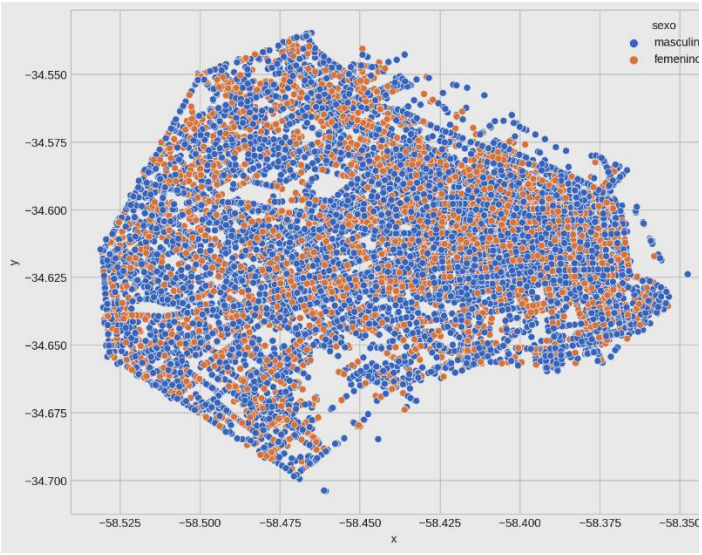


Además, deseamos determinar mediante los datos geográficos con los que contamos, si había alguna concentración de siniestros anómala por sexo en lo que respecta a la CABA; lo realizamos con scatterplot en Fig. 7.

Del gráfico observamos que no había una concentración notoria de siniestros de personas del sexo masculino por sobre femenino, o viceversa.

FIG. 7

SCATTERPLOT DE SINIESTROS SEGÚN SEXO EN EL TERRITORIO



05.03 COMUNAS

Del análisis de las comunas pudimos observar que en las cuales es más frecuente la ocurrencia de siniestros son:

1. Comuna 1
2. Comuna 3
3. Comuna 15
4. Comuna 9
5. Comuna 14

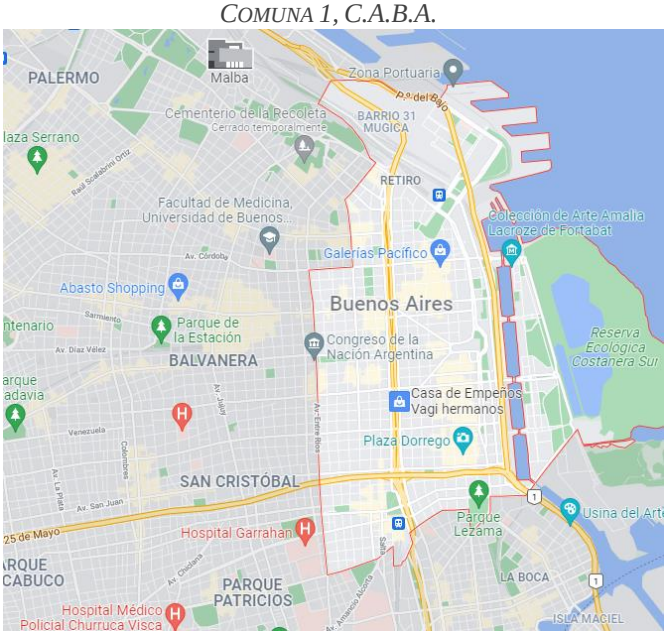
Que justamete coinciden con las áreas de más movimiento en la Ciudad de Buenos Aires, donde la que se destaca en la Fig. 8 sobre las cinco primeras es la *Comuna 1* (Fig. 9), la cual contiene a zonas como el Microcentro porteño y Retiro, zonas altamente transitadas por vehículos y peatones, donde actualmente ya se hace año tras año una reducción del tránsito como política del gobierno local.

FIG. 8

SINIESTROS SEGÚN COMUNA

FIG. 9

COMUNA 1, C.A.B.A.

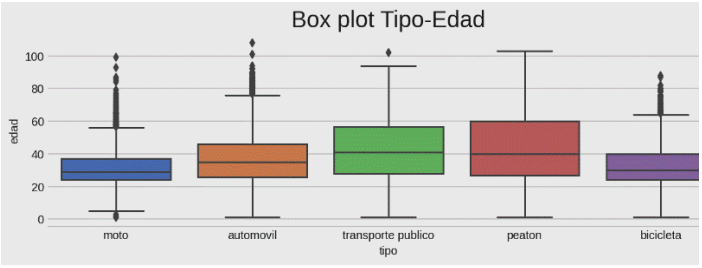


05.04 EDAD

De los datos concernientes a la edad, medianate el uso de un gráfico de cajas en la Fig. 10, se pudo ver que la media de siniestros en moto y bicicleta ronda los 30 años, en lo que respecta a los automóviles estos suben a los 35, en transporte público es de 40 años, al igual que los que involucran a peatones.

FIG. 10

DISTRIBUCIÓN DE EDAD POR TIPOS DE SINIESTROS



05.05 CANTIDAD DE VÍCTIMAS

Con los datos de la media de la edad anteriores tabulados (Fig. 11), obtuvimos las cantidades de víctimas (suma, máximo y mínimo) por tipo de siniestro.

FIG. 11

MEDIA DE EDAD CON CANTIDAD DE VÍCTIMAS POR TIPOS

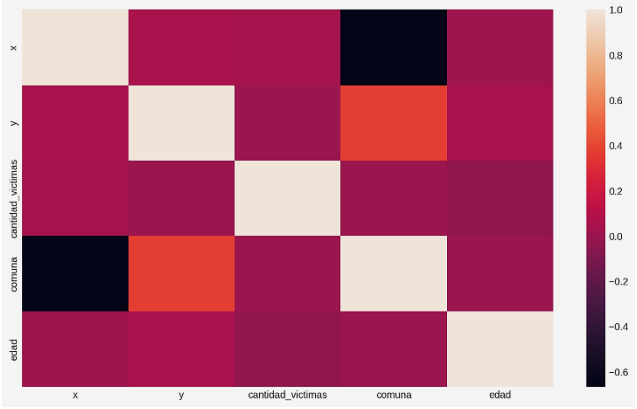
	edad	cantidad_victimas		
	mean	sum	max	min
tipo				
automovil	37.176728	8324	17	1
bicicleta	33.353540	2084	7	1
moto	31.335458	13475	5	1
peaton	43.286399	7773	7	1
transporte publico	42.572031	2810	18	1

05.06 CORRELACIÓN DE VARIABLES

Por último en esta exploración, nos preguntamos sobre la correlación de nuestros datos en el dataframe y visualizamos con un heatmap en Fig. 12.

FIG. 12

CORRELACIÓN DE VARIABLES/CARATERÍSTICAS

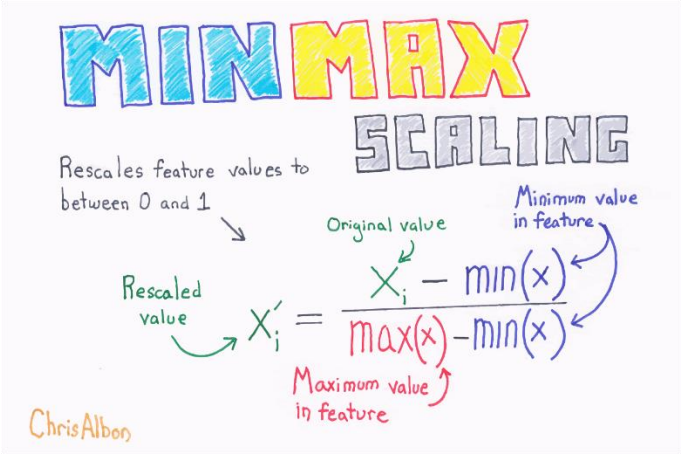


A lo que no encontramos variables muy correlacionadas, excepto de la característica geográfica de Latitud (y) con el número de comuna, si mucho valor práctico.

06 INGENIERÍA DE VARIABLES/CARACTERÍSTICAS

Además de la selección y detalle de nuestros datos realizados hasta ahora, convertimos las columnas con valores del tipo *string* ([‘tipo’, ’causa’]) en variables dummies binarias. También ajustamos mediante `fit_transform` nuesta etiqueta a estimar transformada en y:*array* las cuales queremos que el algoritmo devuelva, mientras que en x:*array* agrupamos las etiquetas estimadoras.

Generamos (tanto x como y) dos sets de entrenamiento y prueba independiente —este último de un 20%—, separados de nuestro dataset final mediante `train_test_split`. Para implementar los modelos explicados posteriormente, primero escalamos los datos con `MinMaxScaler()` ajustados a `X_train`, una función estimadora que transforma las variables escalando cada variable a un rango determinado. escalando y traduciendo cada variable individualmente.



07 MÉTODOS Y MODELOS

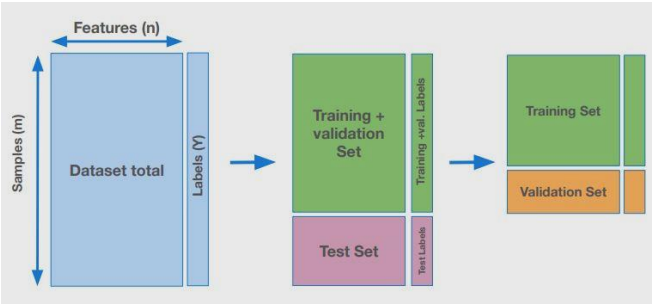
Para trabajar y llegar a nuestro objetivo, utilizamos diversas herramientas de entorno de programación (IDEs), siendo la principal las Jupyter notebooks de Jupyter para el lenguaje de programación *Python*, donde desarrollamos el código para el análisis del dataset tan grande. A su vez, el contexto de aislamiento y la natural necesidad de colaborar en tiempo real, nos llevó a explorar las plataformas surgidas de Google Colaboratory (ahora Colab, ca. 2017) y CoCalc (de Sagemath, ca. 2013), las que nos ayudaron mucho en el trabajo continuo, actualización y seguimiento del software. Contamos dentro de los mismos con las

librerías Numpy (cálculos), Matplotlib (visualización), ScikitLearn (algoritmos), Pandas (gestión de datasets en dataframes). Por último, todo el trabajo se plasmó en una rama de un repositorio en GitHub.

07.01 CLASIFICACIÓN

Con nuestros datos divididos en train y un test de la forma expuesta en Fig. 13, procedemos a realizar una clasificación de la etiqueta seleccionada. Los modelos están caracterizados por parámetros que son aprendidos durante el entrenamiento al ser expuestos a los datos. Adicionalmente, los clasificadores tienen hiper-parámetros que definen la familia de funciones que se pueden aprender. Por medio de una técnica llamada validación cruzada (cross validation) determinamos cuál es la configuración del hiper-parámetro que minimiza el error de clasificación.

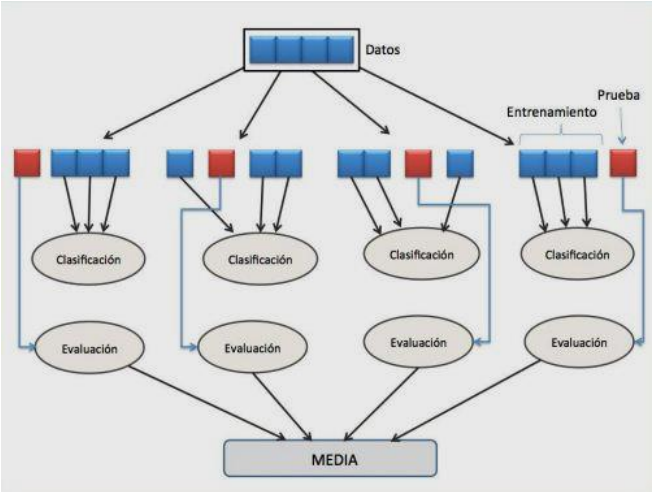
FIG. 13
PROCESO DE OBTENCIÓN DE TRAIN + TEST SETS



El clasificador aprendió la regla de decisión utilizando el train set (samples + labels). Luego, clasificó las muestras de test (sin mirar las labels de test) y se midió la exactitud de clasificación en testeo.

"*Cross validation*" (CV) se realizó con las muestras de entrenamiento. Consistió en dividir nuestro training set en *K folds* (K porciones) e iterar K veces (Fig. 14). En cada iteración, una porción se utilizó como validación independiente y el resto como train. En cada iteración, se entrenó un modelo con train y se evaluaba el resultado de clasificación con validación.

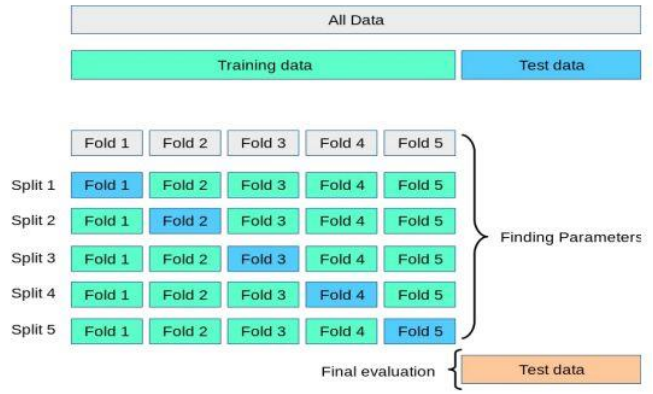
FIG. 14
PROCESO DE VALIDACIÓN CRUZADA



Luego, se realizaba un promedio de la exactitud de clasificación de las K iteraciones. Cross validation sirvió para poder estimar el error estadísticamente. Además, si existiesen varios hiperparametros, a cada uno se estimaba su error por cross validation y se preservaba el hiper-parámetro que menor error promedio de cross validation generara.

El cross validation también nos dio una idea de cómo funciona el modelo propuesto frente a distintas particiones train-valid de los datos. En general, un buen modelo debía "clasificar aceptablemente bien" en todas las particiones; era una manera de "sincerar" si el modelo funcionaba bien en distintos escenarios y no sólo depender de la suerte de nuestra partición.

FIG. 15
GRID SEARCH EN UNA VALIDACIÓN CRUZADA



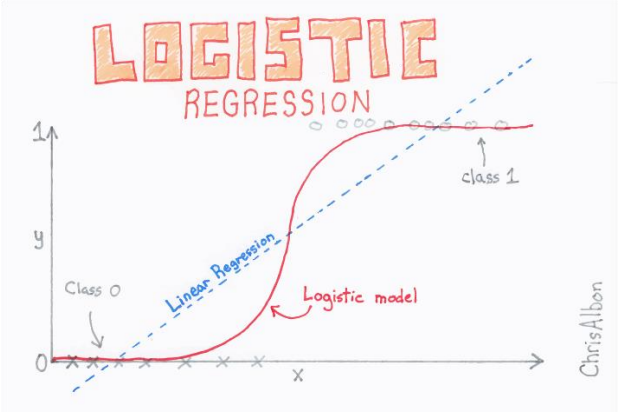
Los modelos de clasificación que utilizamos consistieron de hiper-parámetros que el usuario debio seleccionar. Estos determinaron la regla de decisión y

por ende la performance del modelo. Para saber qué hiper-parámetros seleccionar, lo que hicimos fue generar una lista de los mismos y probar todas las combinaciones posibles de ellos (grid-search, Fig. 15)

En resumen, el *pipeline* del proceso quedó de la forma:

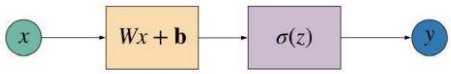
FIG. 16
PIPELINE DEL PROCESO DE CLASIFICACIÓN POR ML

07.01.1 REGRESIÓN LOGÍSTICA



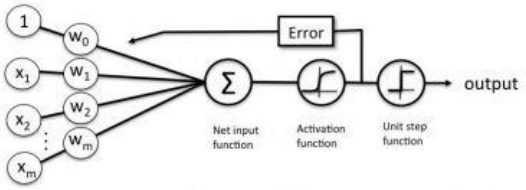
Es un clasificador lineal compuesto por una regresión lineal, precedida de una función activación sigmoide, por lo cual el output es binario y no continuo (Fig. 17). A cada muestra clasificada le asigna una probabilidad de pertenecer a cada clase existente en el problema. Si esta es mayor a cierto threshold (0,5) entonces pertenece a esta clase, de lo contrario viceversa.

FIG. 17
MODELO LINEAL A FUNCIÓN SIGMOID



El regresor logístico aprende de un parámetro interno por cada dimensión del vector de entrada (vector W). Calcula el gradiente del error de clasificación y trata de minimizarlo. Este modelo podría capturar relaciones lineales, si existieran.

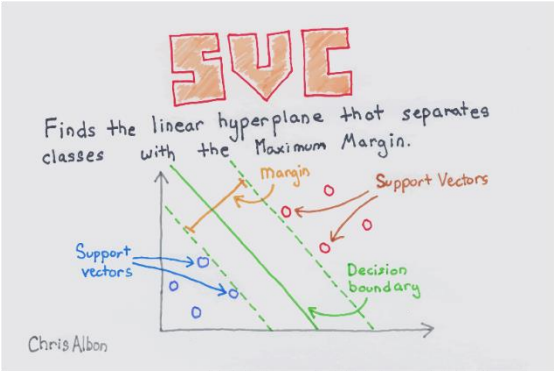
FIG. 18
ESQUEMÁTICA DE CLASIFICADOR DE REGRESIÓN LOGÍSTICA



En nuestro caso, le ingresamos los *costos* y *penalizaciones* para que procese con los *solver*:

```
param_lr = {'C':[1, 10, 100, 1000],
"penalty": ("l1", "l2"),
"solver":("lbfgs", "liblinear")}
```

07.01.2 SVM



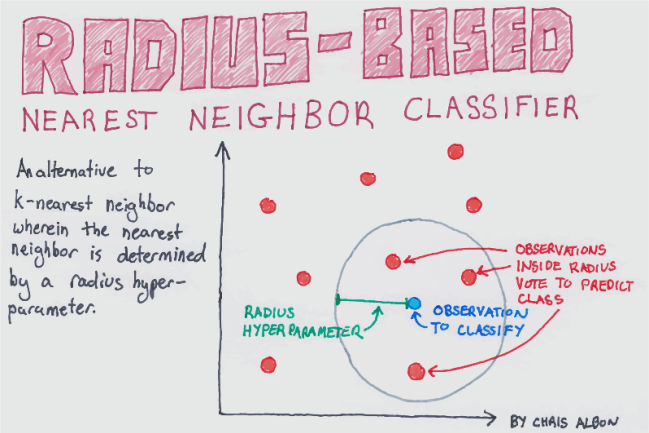
Se trata de un clasificador lineal, el cual busca un hiperplano que maximiza el margen entre las clases. En el caso de que las clases no sean linealmente separables, se acude al *Soft Margin*, un penalizador de muestras mal clasificadas, las cuales se penalizan con un Costo seleccionado por el usuario. Este modelo calcula el mejor hiperplano dentro de las opciones posibles, lidiando con clases superpuestas mediante el *Soft Margin* ya mencionado. Se busca maximizar el margen de los datos generados con el hiperplano, haciendo que la mayor parte de las muestras caigan cerca del plano. El margen separador queda definido por "s" muestras, llamadas *support vectors*.

Los kernels son funciones de similaridad entre muestras. Mapean nuestros datos a un espacio de alta dimensión donde son linealmente separables. Allí en ese nuevo espacio donde son mapeadas las muestras se aplican los productos internos (o similaridad). Cuando usamos SVM, podemos aplicar un kernel para facilitar la clasificación, es decir que el hiperplano estará afectado por el kernel.

En nuestro caso, le ingresamos los *costos*, *gamma* y penalizaciones para que procese con los *kernels*:

```
param_svreg = {'kernel':('linear',
'rbf'), 'C':[1, 10], 'gamma':[0.1,1]}]
```

07.01.3 KNN



Es un modelo de clasificación en el cual un nuevo dato es agrupado según K vecinos más cercanos. Para esto, se calcula la distancia del elemento nuevo a los existentes y se ordenan para seleccionar a qué grupo pertenecen. Uno de los hiper-parámetros del modelo es determinar la cantidad de K vecinos. En el método KNN, se clasifica cada nuevo set de pesos según corresponda los K vecinos más cercanos de una comuna u otra. El cálculo de la distancia que utilizamos es el default, que es la Euclidiana que se corresponde a la fórmula. $d(P1, P2) = \sqrt{(x2 - x1)2 - (y2 - y1)2}$

La ventaja del modelo KNN es que al ser un método no paramétrico, se nutre de la existencia de no-linealidades en los datos, a diferencia del modelo de regresión logística.

En nuestro caso, le ingresamos los *neighbors*:

```
param_neigh = {'n_neighbors':[1, 10,
100, 200]}
```

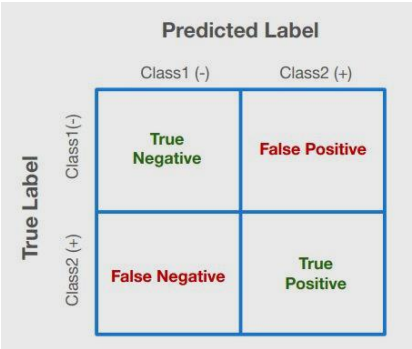
08 Resultados

Para evaluar los resultados obtenidos de la experiencia, recurrimos a la *matriz de confusión* la cual es un elemento para evaluar los resultados de clasificación de ML. Esta nos permitió determinar qué tan bien estuvo clasificando el modelo adoptado, lo cual nos permitió visualizar qué tan precisas eran nuestra salidas (resultado del mejor modelo en Fig. 22).

Como ya se mencionó anteriormente, nuestra *label* (etiqueta) es el sexo de la persona (dividiéndose en siniestro de personas masculinas y femeninas).

Fig. 19

Matriz de confusión



Accuracy = (TN+TP) / Total
Sensitivity (recall) = TP/(TP+FN)
Specificity = TN/(TN+FP)

Fig. 20

Resultados de losmodelos (score/accuracy)

MODELO	ACCURACY
Regresión Logística	71%
SVM	0.1%
KNN	72%

El *accuracy*, que es la métrica que nos permitió evaluar los modelos de clasificación con los que se trabajaron, en nuestro caso nos dio una exactitud del 0.72 o 72% en los datos de entrenamiento (lo que equivale a unas 72 predicciones correctas sobre 100) para el KNN y un 0.71 o 71% con los datos de testeo para el SVM (Fig. 20). Esto nos llevó a la conclusión que el modelo tuvo un desempeño adecuado (aunque no es del todo exacto), en la identificación y clasificación por sexo de la persona siniestradas con los datos aportados.

Además, estudiando los resultado obtenidos de la curva ROC (área bajo la curva, Fig. 21) —el área bajo la curva ROC (AUC) da una idea de cuan bueno es mi clasificador independientemente del accuracy— tuvimos un AUC = 0.73 ó 73%, el cual estaba más cercano a 1 que a 0.5, lo cual nos dijo que para distintos umbrales de clasificación, las clasificación de nuestro modelo seguía siendo buena.

FIG. 21
GRÁFICO ROC (ÁREA BAJO LA CURVA)

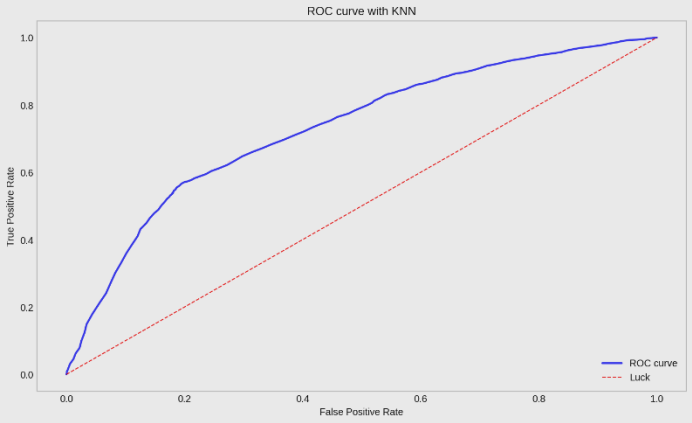


FIG. 22
MATRIZ DE CONFUSIÓN PARA KNN-CLASSIFIER

De la observación de la matriz, se puede apreciar que la diagonal no es sustancialmente más elevada que los restantes cuadrantes, sobre todo en la clasificación de siniestros en personas de sexo "femenino", donde aparentemente se las estaría clasificando un elevado número de casos con la etiqueta "masculinos".

Los errores y el rango se dieron de la siguiente forma:

	Model	**RSME**	**MSE**	**MAE**
0	KNN	0.534361	0.285542	0.285542

09 CONCLUSIONES

De los modelos a los cuales recurrimos en este estudio, se llegó a la conclusión que el KNN era el que mejores resultados nos proporcionaba, siendo el segundo mejor modelo el de Regresión Logística, y con el que obtuvimos peores resultados el de SVM.

Sin embargo, el modelo KNN, si bien nos daba un *accuracy* bastante bueno, seguía teniendo inexactitudes, lo cual se pudo probar en el grafico de la Matriz de Confusióny y en la determinación del error, donde como mencionamos en muchos casos estaría etiquetando a personas de sexo femenino como masculinos, y este error es más acentuado que en el etiquetado de personas masculinas; el error absoluto es mayor al 25%. Esto se puede deber a los datos con los que contamos en nuestro data-set, donde nos encontramos con una gran cantidad de variables de tipo categóricas.

En conclusión, se obtuvo buenos resultados en cuanto la clasificación de siniestros por sexo de la persona, pero también se observó que se puede mejorar el modelo, lo cual se podría lograr realizando un *merging* con otros data-sets que nos brinden más información, y que sobre todo cuenten con datos de tipo numérico y continuo.

Nuestra hipótesis inicial sobre la predicción de clasificación de siniestros, si tomamos un error admisible del 10%, hace que el modelo actualmente estudiado no sea lo suficientemente certero.

Más allá de ese resultado, este análisis nos permitió conocer la caracterización de la seguridad vial en la Ciudad de Buenos Aires con mucha profundidad.

10 REFERENCIAS

[1] *Presentamos el informe de víctimas fatales 2020 y el segundo Plan de Seguridad Vial de la Ciudad*. (2020). Buenos Aires Ciudad - Gobierno de La Ciudad Autónoma de Buenos Aires, from https://www.buenosaires.gob.ar/jefaturadegabinete/movilidad/noticias/presentamos-el-informe-de-victimas-fatales-2020-y-el-segundo

[2] *Gobierno abierto*. (n.d.). Buenos Aires Ciudad - Gobierno de La Ciudad Autónoma de Buenos Aires, from https://www.buenosaires.gob.ar/agendadetransparencia/gobierno-abierto

[3] *Plan de Seguridad Vial*. (n.d.). Buenos Aires Ciudad - Gobierno de La Ciudad Autónoma de Buenos Aires, from https://www.buenosaires.gob.ar/movilidad/plan-de-seguridad-vial

[4] *Buenos Aires Data*. (n.d.). Buenos Aires Data, from https://data.buenosaires.gob.ar/

[5] *Buenos Aires Data*. (n.d.). Buenos Aires Data, from https://data.buenosaires.gob.ar/dataset/victimas-siniestros-viales

[6] *Las motos son la primera causa de muerte en siniestros de tránsito en niños y adolescentes en América Latina* (08 de mayo de 2017). Montevideo, Uruguay. Caf.com, from https://www.caf.com/es/actualidad/noticias/2017/05/las-motos-son-la-primera-causa-de-muerte-en-siniestros-de-transito-en-ninos-y-adolescentes-en-america-latina/

[7] *Apuntes Cluster AI. 2021 - Cátedra de Ciencia de Datos UTN – FRBA* (2021), from https://github.com/clusterai/clusterai_2021

[8] Albon, Chris. *Machine Learning Flashcards*. (n.d.). Machinelearningflashcards.com, from https://machinelearningflashcards.com/

[9] *Elements of Statistical Learning: data mining, inference, and prediction. 2nd Edition.* (2019). Stanford.edu, from https://web.stanford.edu/~hastie/ElemStatLearn/

[10] VanderPlas, J. (2019). *Python Data Science Handbook | Python Data Science Handbook*. Github.io, from https://jakevdp.github.io/PythonDataScienceHandbook

[11] *The Jupyter Notebook — Jupyter Notebook 6.1.3 documentation. (n.d.). Jupyter-Notebook.readthedocs.io*, from https://jupyter-notebook.readthedocs.io/en/stable/

[12] *Google Colaboratory. (n.d.). Colab.research.google.com.*, from https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/01.01-Help-And-Documentation.ipynb

[13] *Quinn, S. (2018, July 23). How Google has crushed it with Colaboratory. Kainos Applied Innovation*, from https://medium.com/kainos-applied-innovation/how-google-has-crushed-it-with-colaboratory-5664b5fb5856

[14] *What is CoCalc? — CoCalc Manual documentation. (n.d.). Doc.cocalc.com.*, from https://doc.cocalc.com/

[15] *CoCalc. (2020, July 24).* Wikipedia, from https://en.wikipedia.org/wiki/CoCalc

[16] *NumPy user guide — NumPy v1.21 Manual*. (n.d.). Numpy.org, from https://numpy.org/doc/stable/user/index.html

[17] *User Guide — pandas 1.3.4 documentation*. (n.d.). Pandas.pydata.org, from https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html

[18] *SciPy — SciPy v1.4.1 Reference Guide*. (2019). Scipy.org, from https://docs.scipy.org/doc/scipy/reference/

[19] *Users guide — Matplotlib 3.5.0 documentation*. (n.d.). Matplotlib.org, from https://matplotlib.org/stable/users/index.html

[20] *User guide and tutorial — seaborn 0.11.2 documentation*. (n.d.). Seaborn.pydata.org. https://seaborn.pydata.org/tutorial.html

[21] *User guide: contents — scikit-learn 0.22.1 documentation*. (2019). Scikit-Learn.org. https://scikit-learn.org/stable/user_guide.html

[22] *Machine Learning 101 – Medium*. (multiple-years' publications). Medium. Retrieved in 2021, from https://medium.com/machine-learning-101/

11 RECONOCIMIENTOS

Finalmente, nos pareció necesario agradecer a los profesores de la materia Ciencia de Datos: Martín Palazzo, Nicolás Aguirre y Santiago Chas por brindar su conocimiento y técnica, y por proveer su tiempo en una asignatura electiva de una universidad pública; también a tod@s l@s ayudantes de la cátedra, y especialmente a Lautaro Rshaid, nuestro mentor.