

Article

Implementación de un algoritmo predictivo para la optimización de la distribución de ambulancias en la ciudad de Bogotá D.C.

Maria Alejandra Saavedra Polanco¹, Sara Consuelo Cobaleda Cruz² and Jaime Alfonso Lemus Casas³

¹ Ingeniero de Sistemas; jaimea-lemusc@unilibre.edu.co

² Ingeniera de Sistemas; sarac-cobaledac@unilibre.edu.co

³ Ingeniera de Sistemas; mariaa-saavedrap@unilibre.edu.co

Received: 15/01/2023; Accepted: 23/03/2023; Published: 30/06/2023

Abstract: This document implements the Support Vector Machine (SVM) algorithm in a .csv data file obtained from the "Bogotá free data" portal, which contained the emergency calls that occurred throughout the month of February. The foregoing was carried out with the objective of predicting how many calls will be made during the day to the emergency line in order to assign the number of ambulances necessary in a certain location in Bogotá.

Keywords: ambulance, emergency call, prediction, algorithm

1. Introducción

La minería de datos es un proceso que consiste en descubrir conocimientos como patrones, asociaciones, cambios, anomalías y estructuras significativas a partir de grandes cantidades de datos [1]. En este documento se busca implementar un algoritmo de predicción denominado Support Vector Machine (SVM), "el cual es un grupo de métodos basados en aprendizaje supervisado, con el objetivo de clasificar, predecir y detectar valores atípicos" [2], con el fin de obtener el número de ambulancias que se necesitan en un día en específico en la ciudad, los datos utilizados serán obtenidos a través de la página de "Datos abiertos Bogotá" [3]

El modelo será entrenado partiendo de la teoría del aprendizaje automático [4] [5], para ello, los pasos a seguir serán: Preparar los datos obtenidos a través de un código que elimina la información que no es importante para el análisis, además de eliminar datos nulos que puedan presentar un error en el análisis, a continuación, se ingresa el conjunto de datos ya preparados al algoritmo para ser entrenado, los datos estarán en formato CSV separados por comas [6].

Segundo, se usará el código de entrenamiento del algoritmo para que adopte un modelo predictivo, este modelo trabaja a través de etiquetas las cuales genera el algoritmo a partir de la agrupación de datos dadas por el archivo de datos, en este paso también se medirá la precisión del algoritmo con datos a los cuales ya se sepa el resultado [7]. El Tercero es ingresar nuevos datos al modelo predictivo, este será para que el modelo evalúe y clasifique a que grupo o etiqueta pertenece el archivo [6].

2. Materiales y Métodos

A continuación, se describirán el proceso llevado a cabo para la realización del proyecto, se especificarán los detalles tomados en cuenta en el desarrollo y se explicarán las decisiones tomadas para el mismo.

1.1 Adquisición y preparación de la data

El primer paso consistió en adquirir la data necesaria para el desarrollo del proyecto, esta se buscó a través de la página de datos libres de la alcaldía de Bogotá con el título “Llamadas de Urgencias y Emergencias que ingresan a través de la línea 123”, y se utilizó específicamente la data de los meses Enero, Febrero y Marzo del año 2022.

A continuación, se realizó en el preprocesamiento de la data, para ello se utilizó la librería pandas de Python, junto con el archivo el archivo CSV obtenido, y se realizó un código para eliminar la información que no es necesaria en el análisis. El proceso que consistió en eliminar 9 de las 10 columnas que había en la data, dando como resultado un data frame de una columna que solo contenía fechas, a continuación, se contó cuántas veces aparecía una misma fecha en el data frame, este proceso da como resultado la cantidad de llamadas que se presentaron en un día específico. Finalmente se colocaron los valores obtenidos en un nuevo data frame de dos columnas, la primera contiene números del 1 al 90, correspondientes a los primeros 90 días del año, y la segunda columna contiene la cantidad de llamadas presentadas en ese día.

	A	B	C	D	E	F	G	H	I	J
1	NUMERO_IN	FECHA_INICIO_DESPLAZAMIENTO-MOVI	CODIGO_LOCALIDAD	LOCALIDAD	EDAD	UNIDAD	GENERO	RED	TIPO_INCIDENTE	PRIORIDAD
2	CRU-0004465	1/02/2022 0:12	4	San Cristóbal	SIN_DATO	SIN_DATO	SIN_DATO	Centro Oriente	Evento Respiratorio	BAJA
3	CRU-0004465	1/02/2022 0:27	11	Suba	77	Años	FEMENINO	Norte	Accidente Cerebro Vascular	CRITCA
4	CRU-0004465	1/02/2022 0:31	7	Bosa	SIN_DATO	SIN_DATO	SIN_DATO	Privada	Herido	ALTA
5	CRU-0004470	1/02/2022 0:36	7	Bosa	29	Años	FEMENINO	Sur Occidente	Patología Ginecobastrica	ALTA
6	CRU-0004470	1/02/2022 0:53	10	Engativ	75	Años	MASCULINO	Norte	Enfermo	ALTA
7	CRU-0004470	1/02/2022 0:55	8	Kennedy	22	Años	MASCULINO	Sur Occidente	Convulsión	ALTA
8	CRU-0004470	1/02/2022 0:57	9	Fontibón	36	Años	MASCULINO	Sur Occidente	Enfermo	ALTA
9	CRU-0004471	1/02/2022 1:05	7	Bosa	25	Años	MASCULINO	Sur	Convulsión	ALTA
10	CRU-0004471	1/02/2022 1:21	14	Los Martires	44	Años	MASCULINO	Sur Occidente	Síntomas Gastrointestinales	ALTA
11	CRU-0004471	1/02/2022 1:38	7	Bosa	59	Años	MASCULINO	Sur	Enfermo	MEDIA
12	CRU-0004471	1/02/2022 1:42	8	Kennedy	SIN_DATO	SIN_DATO	SIN_DATO	Sur	Evento Respiratorio	BAJA
13	CRU-0004471	1/02/2022 1:53	18	Rafael Uribe Uribe	53	Años	FEMENINO	Centro Oriente	Intento de Suicidio	CRITCA
14	CRU-0004472	1/02/2022 1:53	11	Suba	19	Años	FEMENINO	Norte	Herido	ALTA
15	CRU-0004472	1/02/2022 1:55	8	Kennedy	83	Años	FEMENINO	Norte	Herido	ALTA
16	CRU-0004472	1/02/2022 2:18	8	Kennedy	6	Años	MASCULINO	Sur	Enfermo	MEDIA
17	CRU-0004472	1/02/2022 2:22	11	Suba	65	Años	MASCULINO	Norte	Dolor Torácico	CRITCA
18	CRU-0004472	1/02/2022 2:35	7	Bosa	SIN_DATO	SIN_DATO	SIN_DATO	Sur Occidente	Enfermo	MEDIA
19	CRU-0004472	1/02/2022 2:59	18	Rafael Uribe Uribe	SIN_DATO	SIN_DATO	SIN_DATO	Centro Oriente	Evento Respiratorio	ALTA
20	CRU-0004472	1/02/2022 3:45	12	Barrios Unidos	39	Años	MASCULINO	Norte	Herido	ALTA
21	CRU-0004474	1/02/2022 3:58	4	San Cristóbal	SIN_DATO	SIN_DATO	SIN_DATO	Centro Oriente	Inconsciente / Paro cardiorespiratorio	CRITCA
22	CRU-0004474	1/02/2022 4:11	4	San Cristóbal	SIN_DATO	SIN_DATO	SIN_DATO	Centro Oriente	Herido	MEDIA
23	CRU-0004474	1/02/2022 4:20	10	Engativ	53	Años	MASCULINO	Norte	Convulsión	CRITCA

Figura-1. Datos originales de enero de 2022.

Adicionalmente, el código verificó si existía algún dato nulo en el archivo, y se encargó de convertir las fechas en formato dd/mm/aa a únicamente números.

1.2 Algoritmo de aprendizaje

El siguiente paso consistió en seleccionar e implementar un algoritmo de aprendizaje. Debido a que el problema se centra en obtener la cantidad de llamadas de emergencia más probable, para desarrollar este trabajo fue necesario implementar el algoritmo Support Vector Machine (SVM). En primer lugar, se realizó una gráfica de los datos, a continuación, se procedió a dividir la data, de esta forma se utilizaron el 80% de los datos para entrenar el algoritmo, y el otro

20% para verificar la exactitud del modelo. Finalmente, se entrenó el modelo y se compararon los resultados obtenidos de este, con los datos originales.

3. Resultados

A continuación, se presentan los resultados obtenidos a través de todos los procesos. esta sección se dividirá en dos grandes ramas, la primera corresponderá al procesamiento de la data, y la segunda al algoritmo Support Vector Machine

3.1 Data procesada:

El Proceso consistió en eliminar todas las columnas con información que no era relevante para la predicción de la cantidad de llamadas, por tanto, se eliminaron las columnas de: NUMERO_DE_INCIDENTE, LOCALIDAD, EDAD, UNIDAD, GENERO, RED, TIPO_INCIDENTE y PRIORIDAD, a continuación, se obtuvo el número de día de cada uno de los días de cada archivo CSV, finalmente se contó cuántas veces aparece cada número en los registros, de esta manera se obtienen la cantidad de casos sucedidos en un día específico, estas sumatorias se colocaron progresivamente en un nuevo data frame con 90 registros, correspondientes a los primeros 90 días del año. El código utilizado para la preparación de la data se muestra a continuación:

```

1. import pandas as pd
2. Enero = pd.read_csv('Enero.csv')
3. Febrero = pd.read_csv('Febrero.csv')
4. Marzo = pd.read_csv('Marzo.csv')
5. #obtención de los casos
6. #enero
7. Enero = Enero.drop(['NUMERO_INCIDENTE'], axis=1)
8. Enero = Enero.drop(['LOCALIDAD'], axis=1)
   9. Enero = Enero.drop(['EDAD'],
   axis=1) 10.Enero =
Enero.drop(['UNIDAD'], axis=1) 11.Enero
= Enero.drop(['GENERO'], axis=1)
12.Enero = Enero.drop(['RED'], axis=1)
13.Enero = Enero.drop(['TIPO_INCIDENTE'], axis=1)
14.Enero = Enero.drop(['PRIORIDAD'], axis=1)
15.#Separamos el dia de las fechas
16.dataFecha1
=
   Enero['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'].str.split(pat = '/', n
   =1, expand = True)
17.dataFecha1 = dataFecha1.drop([1], axis=1)
18.Enero['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'] =
dataFecha1
19.#febrero

```

```

20.Febrero = Febrero.drop(['NUMERO_INCIDENTE'],
axis=1)21.Febrero = Febrero.drop(['LOCALIDAD'],
axis=1) 22.Febrero = Febrero.drop(['EDAD'], axis=1)
23.Febrero = Febrero.drop(['UNIDAD'], axis=1)
24.Febrero = Febrero.drop(['GENERO'], axis=1)
25.Febrero = Febrero.drop(['RED'], axis=1)
26.Febrero = Febrero.drop(['TIPO_INCIDENTE'],
axis=1)27.Febrero = Febrero.drop(['PRIORIDAD'],
axis=1) 28.#Separamos el dia de las fechas
29.dataFecha2

=
    Febrero['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'].str.split(pat = '-',
n
    = 2, expand = True)
30.dataFecha2 = dataFecha2.drop([0], axis=1)
31.dataFecha2 = dataFecha2.drop([1], axis=1)
32.dataFecha2 = dataFecha2[2].str.split(pat = ' ', n = 1, expand
= True)
33.dataFecha2 = dataFecha2.drop([1], axis=1)
34.Febrero['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'] =
dataFecha2
35.#marzo
36.Marzo = Marzo.drop(['NUMERO_INCIDENTE'],
axis=1)37.Marzo = Marzo.drop(['LOCALIDAD'],
axis=1) 38.Marzo = Marzo.drop(['EDAD'], axis=1)
39.Marzo = Marzo.drop(['UNIDAD'],
axis=1)          40.Marzo =
Marzo.drop(['GENERO'], axis=1) 41.Marzo
= Marzo.drop(['RED'], axis=1)
42.Marzo = Marzo.drop(['TIPO_INCIDENTE'], axis=1)
43.Marzo = Marzo.drop(['PRIORIDAD'], axis=1)
44.Marzo = Marzo.drop(['RECEPCION'], axis=1)
45.#Separamos el dia de las fechas
46.dataFecha3

=
    Marzo['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'].str.split(pat = '-', n
=
    2, expand = True)
47.dataFecha3 = dataFecha3.drop([0], axis=1)
48.dataFecha3 = dataFecha3.drop([1], axis=1)
49.dataFecha3 = dataFecha3[2].str.split(pat = ' ', n = 1, expand
= True)
50.dataFecha3 = dataFecha3.drop([1], axis=1)
51.Marzo['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'] =
dataFecha3

```

```
52.#alistamos la data final
53.Final_Data = pd.DataFrame()
54.Final_Data['Fecha'] = [i+1 for i in range (90)]
55.Final_Data['Casos'] = None
56.#alistar la suma de casos por dia
57.cantidad_total = [0 for i in range(Final_Data.shape[0])]
58.total1 = Enero['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'].tolist()
59.total2 = Febrero['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'].tolist()
60.total3 = Marzo['FECHA_INICIO_DESPLAZAMIENTO-MOVIL'].tolist()
61.#convertimos a enteros
62.total1 = list(map(int,
total1)) 63.total2 =
list(map(int, total2)) 64.total3
= list(map(int, total3))
65.#sumamos e introducimos al dataframe final
66.for i in range(Final_Data.shape[0]):
67.    if i<31:
68.        cantidad_total[i] = total1.count(i+1)
69.        #print(i, i+1)
70.    elif i>30 and i<59:
71.        cantidad_total[i] = total2.count(i-30)
72.        #print(i, i-30)
73.    else:
74.        cantidad_total[i] = total3.count(i-58)
75.        #print(i, i-58)
76.Final_Data['Casos'] = cantidad_total
77.#exportar
78.Final_Data.to_csv('Final_data_extendido.csv
')
```

La data obtenida del proceso de preparación consistió en un archivo CSV de dos columnas y 90 registros, la primera columna corresponde a los días, y la segunda a la suma de llamadas que se presentaron ese día.

	A	B	C
1		Fecha	Casos
2	0	1	456
3	1	2	345
4	2	3	391
5	3	4	372
6	4	5	388
7	5	6	354
8	6	7	348
9	7	8	406
10	8	9	397
11	9	10	363
12	10	11	381
13	11	12	393
14	12	13	405
15	13	14	412
16	14	15	450
17	15	16	413
18	16	17	409
19	17	18	363
20	18	19	412
21	19	20	421
22	20	21	369

Figura-2. Datos obtenidos del procesamiento de imágenes

3.2 Support Vector Machine

El algoritmo utilizado para crear un modelo predictivo es el Support Vector Machine, en primer lugar, se definieron X como la fecha y Y como la cantidad de casos, y se graficaron.

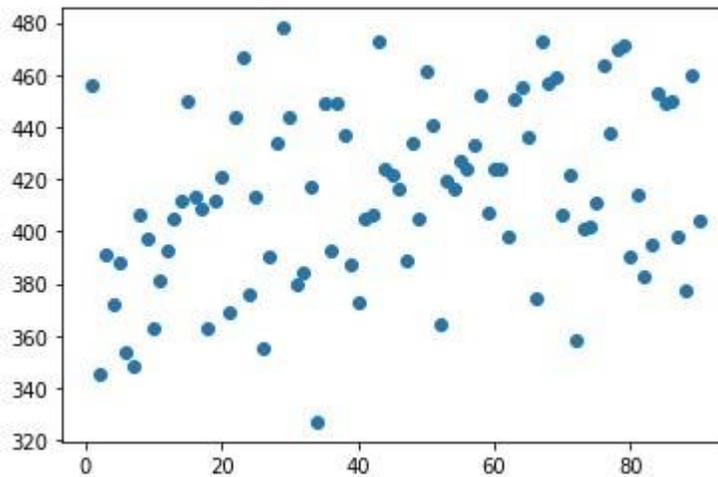


Figura-3. Gráfica de los datos de los primeros 90 días del año

A continuación, se dividió la data de tal manera que se utilizara el 80% de los datos para entrenar el modelo, y el 20% restante para verificar la exactitud de este. Finalmente se creó el modelo a partir de los datos ya reservados y se verificó su exactitud con la librería NumPy, dando como resultado un rango de exactitud de entre el 90% y el 94%. El código utilizado para la obtención del modelo predictivo se muestra a continuación:

```

1. import numpy as np
2. import matplotlib.pyplot as plt
3. import pandas as pd
4. from sklearn import svm
5. data=pd.read_csv("Final_data_extendido.csv")
6. #regresion polinomial
7. #definir x y y
8. x = data["Fecha"].tolist()
   9. y = data["Casos"].tolist()
  10. #pasamos de un list a un
   array
  11. X = np.array(x)
  12. Y = np.array(y)
  13. X= np.reshape(X, [90,1])
  14. #graficamos
  15. plt.scatter(x,y)
  16. plt.show()
  17. from sklearn.model_selection import train_test_split
  18. X_train, X_test, y_train, y_test = train_test_split(X,
   Y,test_size=0.2)

  19. regressor = svm.SVC(decision_function_shape='ovo')
  20. regressor.fit(X_train, y_train)
  21. y_pred = regressor.predict(X_test)
  22. df = pd.DataFrame({"Actual": y_test, "Predicho":
   y_pred})
  23. errors = abs(y_pred - y_test)
  24. mape = 100 * (errors / y_test)
  25. accuracy = 100 - np.mean(mape)
  26. print("Precisión del Modelo: ",round(accuracy, 2))

```

4. Discusión

El presente estudio mostró un alto nivel de precisión en el modelo, al considerar la data sobre los 3 meses se obtuvo un rango de datos muy amplio dando como resultado un buen grado de eficacia. Por lo cual, se logrará determinar en qué fechas será mayor el número de accidentes, esto les facilitará a las empresas prestadoras de servicios un promedio de emergencias a atender en ciertos días.

Este programa puede llegar a fallar si llegase a ocurrir una emergencia la cual se salga de los rangos de lo normal; Se refiere a estas como atentados o en festividades especiales los cuales tienden a tener picos muy altos, por lo general provocados por hechos de intolerancia [8] y por ende el software no lo logrará predecir completamente las emergencias. Sin embargo, el algoritmo

obtendrá un escrúpulo entre 90% - 94% en ocasiones normales. Hay que considerar que las condiciones bajo las cuales se creó este modelo no son constantes, es decir, si sucediera un acontecimiento que diera como resultado una alza en las llamadas de emergencias a futuro, por ejemplo un paro prolongado o una migración masiva, el modelo quedaría desfasado, por otra parte, existen meses en el año que suelen presentar un mayor índice de emergencias que otras, y por tanto el modelo perdería exactitud en esas épocas, dando como resultado una aplicabilidad en un lapso menor de tiempo.

4. Conclusiones

En conclusión el sistema de predicción propuesto a lo largo del documento, el cual utiliza el algoritmo Support Vector Machine (SVM), con el fin de que como su nombre lo dice predecir la cantidad de llamadas a la línea de emergencia en la ciudad de Bogotá, puede ser de gran utilidad para las empresas o instituciones encargadas de prestar el servicio de asistencia en caso de emergencia, debido a que les permite saber con anterioridad una cantidad estimada de las llamadas que recibirán durante cada día del mes en curso o hasta del próximo.

Lo anteriormente mencionado, se debe a las pruebas que se llevaron a cabo, evidenciaron un resultado de precisión, que se encuentra entre el 90% y el 94%, lo que se considera un buen porcentaje, ya que entre más meses la empresa reporte en el algoritmo, esté será más preciso; para las pruebas que se realizaron, se implementó la data de los 3 primeros meses del año 2022, los cuales son correspondientes a enero, febrero y marzo, procesando así un total de 90 días, con sus correspondientes cantidades de llamadas, cabe resaltar, que entre menos picos en los casos presentela data, este será más preciso, es decir, se espera que en el mes de diciembre no sea muy preciso, debido a la cantidad de emergencias que se presentan por las festividades.

References

- [1] «S. Valero, A. Salvador and M. García, "Minería de datos: predicción de la deserción escolar mediante el algoritmo de árboles de decisión y el algoritmo...", Academia , 2005. [En línea],» [En línea].
- [2] «"1.4. Support Vector Machines", scikit-learn, 2022. [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>. [Accessed: 12- May- 2022],» [En línea].
- [3] «Secretaría Distrital de Salud, "Datos Abiertos Bogotá", Datosabiertos.bogota.gov.co, 2022. [Online],» [En línea].
- [4] H. J. Robayo B. y J. S. Arias Ayala, «Desarrollo de un modelo de predicción de casos de deserción universitaria en el área de ciencias de la salud en Colombia,» *Revista Avenir*, vol. 5, nº 1, pp. 24-30, 2021.
- [5] J. Cardona P., A. Currea Meneses y M. Herrera Rodríguez, «Diseño de un modelo de clasificación de la calidad de componentes electrónicos implementando visión artificial a partir de repositorios de imágenes,» *Revista Avenir*, vol. 5, nº 2, pp. 29-44, 2021.
- [6] «Galvaníze, "Predicción de clases y valores numéricos", Help.highbond.com , 2019. [En línea],» [En línea].
- [7] P. Rojas, D. Sandoval y B. Sandoval, «Algoritmo de clasificación de enfermedades en la hoja de café,» *Revista Avenir*, vol. 5, nº 1, pp. 31-39, 2021.
- [8] «El tiempo. "57 % de los homicidios en el país este año han sido por intolerancia". El Tiempo. <https://www.eltiempo.com/justicia/delitos/intolerancia-causa-mas-de-la-mitad-de-las-muertes-violentas-en-colombia-en-2021-567679>,» [En línea].
- [9] «C. Tiempo, "Ambulancias tardan en promedio 35 minutos y 12 segundos en atender emergencias en Bogotá", El Tiempo, 2008. [Online].,» [En línea].



© 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).