MODELADO PARA PRECICIÓN DE VENTAS FARMACEUTICAS ATC

Liliana Ochoa Echeverri

liliana.ochoae@udea.edu.co

Universidad de Antioquia

(11 noviembre de 2022)

Introducción

La clasificación ATC (Clasificación Anatómica, Terapéutica, Química) es un índice de sustancias farmacológicas y medicamentos, organizados según grupos terapéuticos. Este sistema fue instituido por la OMS (Organización Mundial de la Salud) y ha sido adoptado principalmente en Europa, pero también en algunos otros países (como Colombia) [1]. El código recoge el sistema u órgano sobre el que actúa, el efecto farmacológico, las indicaciones terapéuticas y la estructura química del fármaco, de allí la importancia de estudiar el índice de ventas de sus ocho diferentes categorías con el propósito de lograr obtener datos que indiquen el mayor consumo por órgano afectado ya que el ATC, recoge el sistema u órgano sobre el que actúa. En base a esto, se desea predecir si la cantidad vendida y exportada desde el sistema de punto de vista en el individuo farmacia es óptima y suficiente para la demanda que tendrá.

Gracias al uso de los algoritmos de clustering podemos clasificar o agrupar los datos según diferentes parámetros o condiciones, lo cual nos permite ampliar el panorama de los datos obtenidos en una base de datos construyendo modelos que logren ampliar el espectro para realizar análisis más profundos de la información, por tanto, se decide utilizar el método de agrupamiento por medio del algoritmo de K-Means ya que, a partir de este y através el modelo de clasificación lineal kernel, se puede optimizar una respuesta de la precisión de ventas de fármacos por cada categoría.

En este caso, para la realización del modelo y la prueba se decidió utilizar el lenguaje Python por medio de Google Colab.

Exploración descriptiva del dataset

Para esta investigación, se seleccionó un dataset de Kaggle “Datos de Ventas Farmacéuticas”, este conjunto de registros, contiene de 600 000 datos transaccionales recopilados en 6 años (período 2014-2019), en cuatro archivos csv, que son salesdaily, saleshourly, salesmonthly, salesweekly, que

indican la fecha, la hora de la venta, la marca del medicamento farmacéutico y la cantidad vendida, exportados desde el sistema de punto de venta en la farmacia.

El grupo seleccionado de medicamentos del conjunto de datos (57 medicamentos) se clasifica en las siguientes categorías del sistema de ATC:

• M01AB: productos antiinflamatorios y antirreumáticos, no esteroides, derivados del ácido acético y sustancias relacionadas
• M01AE: productos antiinflamatorios y antirreumáticos, no esteroides, derivados del ácido propiónico
• N02BA: otros analgésicos y antipiréticos, ácido salicílico y derivados
• N02BE/B - Otros analgésicos y antipiréticos, Pirazolonas y Anilidas
• N05B - Medicamentos psicolépticos, Medicamentos ansiolíticos
• N05C: drogas psicolépticas, hipnóticas y sedantes
• R03: medicamentos para enfermedades obstructivas de las vías respiratorias
• R06: antihistamínicos para uso sistémico

El dataset a estudiar, contiene 4 archivos con datos por horas, diarios, semanalesy anuales. contenidos en 44 columnas donde se podrán predecir mejor las estadísticas según su predicción.

Iteraciones de desarrollo

Procesado de datos

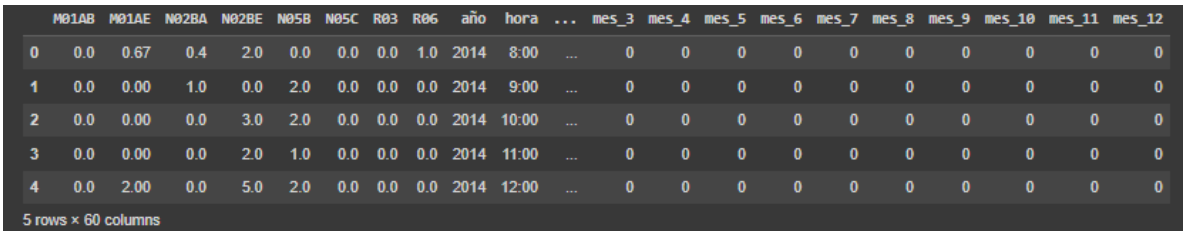
Para poder realizar el análisis de la información inicial, se accedió a una base de datos donde se tiene la lista de ventas farmaceuticas con su clasificación ATC, dento de esa data, se observaron a parte de las 8 clasificaciones de los fármacos, la fecha, la hora y el nombre del día de la semana.

datum	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	Hour	Weekday Name
1/02/2014	0	3.67	3.4	32.4	7	0	0	2	248	Thursday
1/03/2014	8	4	4.4	50.6	16	0	20	4	276	Friday
1/04/2014	2	1	6.5	61.85	10	0	9	1	276	Saturday
1/05/2014	4	3	7	41.1	8	0	3	0	276	Sunday
1/06/2014	5	1	4.5	21.7	16	2	6	2	276	Monday
1/07/2014	0	0	0	0	0	0	0	0	276	Tuesday
1/08/2014	5.33	3	10.5	26.4	19	1	10	0	276	Wednesday
1/09/2014	7	1.68	8	25	16	0	3	2	276	Thursday
1/10/2014	5	2	2	53.3	15	2	0	2	276	Friday
1/11/2014	5	4.34	10.4	52.3	14	0	1	0.2	276	Saturday
1/12/2014	2	0.66	2.5	12	8	0	1	1	276	Sunday
1/13/2014	7.34	7.66	6.2	52	9	0	7	1	276	Monday
1/14/2014	6	1.33	12.3	33.7	6	1	0	2	276	Tuesday
1/15/2014	4	2.34	5	26.7	12	2	3	3	276	Wednesday
1/16/2014	6	2	4.3	28.3	19	1	5	0	276	Thursday
1/17/2014	2	3.68	8.3	20.4	15	0	6	3	276	Friday
1/18/2014	1	5.33	5.8	43.2	15	4	7	2	276	Saturday
1/19/2014	4.33	4	4	14.1	4	0	1	1	276	Sunday
1/20/2014	6	3.34	3.3	11.9	18	2	12	3	276	Monday
1/21/2014	2	3.34	4	42	15	2	0	1	276	Tuesday
1/22/2014	7	3	7	18.1	10	0	5	2	276	Wednesday
1/23/2014	4	4.67	5.2	26.2	2	3	1	3	276	Thursday
1/24/2014	4.67	7.34	7	19.6	13	1	0	0	276	Friday
1/25/2014	6	9	5	37.8	18	0	5	1	276	Saturday
1/26/2014	4.33	1.68	0	24	4	0	0	0	276	Sunday
1/27/2014	4.34	4.67	1.2	23.6	17	2	1	3	276	Monday
1/28/2014	6	3.34	2.8	13.38	22	5	1	5	276	Tuesday
1/29/2014	5.33	4	2	14	10	1	1	2	276	Wednesday
1/30/2014	3.02	1.34	2.4	25.5	7	1	3	2	276	Thursday
1/31/2014	1	2.68	7.1	26.9	9	0	1	0	276	Friday
2/01/2014	4.33	4.32	5	43	13	1	14	0	276	Saturday
2/02/2014	7	3	0.2	13.5	6	2	8	0	276	Sunday
2/03/2014	5	1	8.5	32.4	16	1	1	0	276	Monday
2/04/2014	1.33	3	7	30.6	8	1	17	2	276	Tuesday
2/05/2014	3	4.02	6.2	32.4	15	1	1	1	276	Wednesday

Figura 1. Dataset antes de las transformaciones categóricas

Sin embargo, para poder continuar con el análisis de datos es necesario etiquetar la base de datos, creando una columna para cada valor distinto de la fecha (día, mes,año) para qué exista en la característica y para cada registro marcar con un 1 la columna a la que pertenezca dicho registro y

dejar las demás con 0. Ayudando así a qué la base de datos tenga una mejor lectura de la información.



	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	año	hora	...	mes_3	mes_4	mes_5	mes_6	mes_7	mes_8	mes_9	mes_10	mes_11	mes_12
0	0.0	0.67	0.4	2.0	0.0	0.0	0.0	1.0	2014	8:00	...	0	0	0	0	0	0	0	0	0	0
1	0.0	0.00	1.0	0.0	2.0	0.0	0.0	0.0	2014	9:00	...	0	0	0	0	0	0	0	0	0	0
2	0.0	0.00	0.0	3.0	2.0	0.0	0.0	0.0	2014	10:00	...	0	0	0	0	0	0	0	0	0	0
3	0.0	0.00	0.0	2.0	1.0	0.0	0.0	0.0	2014	11:00	...	0	0	0	0	0	0	0	0	0	0
4	0.0	2.00	0.0	5.0	2.0	0.0	0.0	0.0	2014	12:00	...	0	0	0	0	0	0	0	0	0	0

5 rows × 60 columns

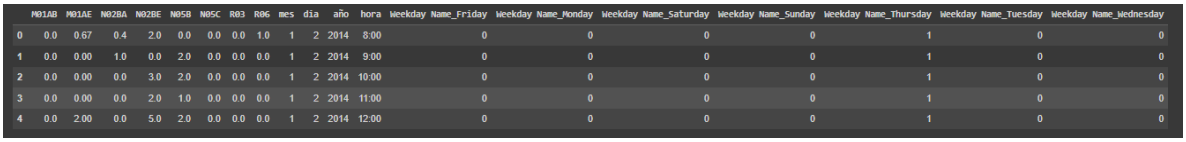
Figura 2. Dataset de transformaciones categóricas

Para comenzar con el preprocesamiento de la base de datos se importan las librerías necesarias.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Figura 3. Dataset preprocesamiento de la base de datos se importan las librerías necesarias

Para el manejo de las variables no cuantitativas usamos el proceso de One Hot Encoding, el cual genera una cantidad n de columnas en donde se les asigna el valor de 0 y 1 que representa el uso de la categoría. Podemos acceder a este proceso por medio de la librería Panda.



	M01AB	M01AE	N02BA	N02BE	N05B	N05C	R03	R06	mes	dia	año	hora	Weekday Name_Friday	Weekday Name_Monday	Weekday Name_Saturday	Weekday Name_Sunday	Weekday Name_Thursday	Weekday Name_Tuesday	Weekday Name_Wednesday
0	0.0	0.67	0.4	2.0	0.0	0.0	0.0	1.0	1	2	2014	8:00	0	0	0	0	1	0	0
1	0.0	0.00	1.0	0.0	2.0	0.0	0.0	0.0	1	2	2014	9:00	0	0	0	0	1	0	0
2	0.0	0.00	0.0	3.0	2.0	0.0	0.0	0.0	1	2	2014	10:00	0	0	0	0	1	0	0
3	0.0	0.00	0.0	2.0	1.0	0.0	0.0	0.0	1	2	2014	11:00	0	0	0	0	1	0	0
4	0.0	2.00	0.0	5.0	2.0	0.0	0.0	0.0	1	2	2014	12:00	0	0	0	0	1	0	0

Figura 3. Dataset una vez aplicadas las transformaciones de categoría

Cómo se puede apreciar en la imagen anterior, ahora se tienen 7 columnas llamadas:

- Weekday Name_Friday
- Weekday Name_Monday
- Weekday Name_Saturday
- Weekday Name_Sunday
- Weekday Name_Thursday
- Weekday Name_Tuesday
- Weekday Name_Wednesday

En las cuales habrá un 1 en la fila donde antes estaba el valor correspondiente de la gravedad del accidente. Este mismo procedimiento se repite con las variables

- dia
- mes
- año
- hora

Obteniendo el siguiente resultado

Modelo no supervisado

Como se mencionó en la introducción, se decidió trabaja con el agrupamiento de K-means ya que es el algoritmo de agrupamiento más utilizado, el cual está basado en centroides e intenta optimizar la varianza de los puntos de datos dentro de un grupo.

Después de realizar varias pruebas de clusteres, se decide aplicar un total de 6 a 8 clusters debido a que es el que mejor hace un agrupamiento de los datos dada la densidad de población presentada en el conjunto de datos.

Este procedimiento se realiza en cada una de las bases de los archivos csv que se encuentran en el dataset.

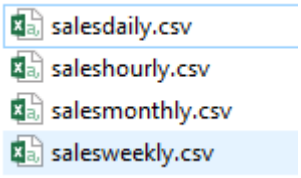


Figura 4. Bases de datos de Kaggle

Al aplicardel algoritmo k-means de 6 a 8 clusteres se muestran los siguientes resultados por base de datos:

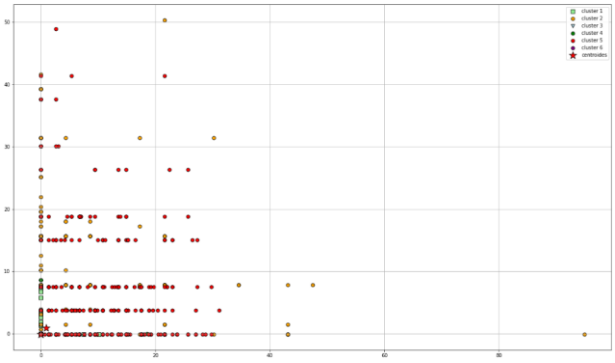


Figura 5. Clusteres en saleshourly

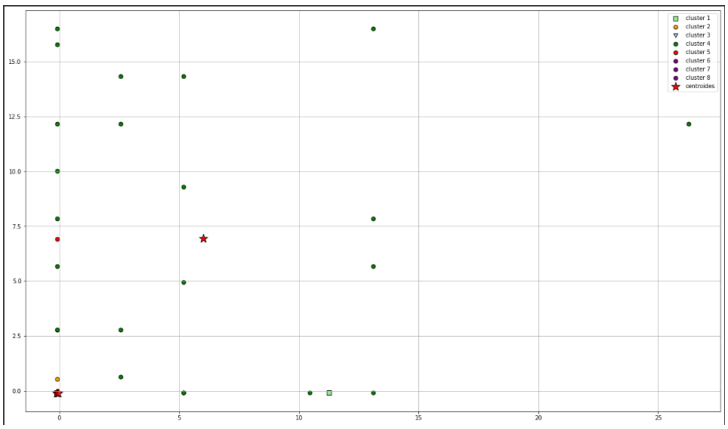


Figura 6. Clusteres en salesdaily

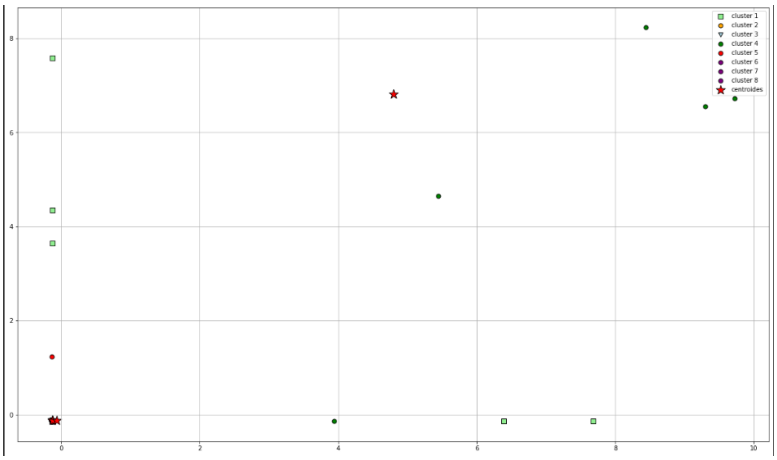


Figura 7. Clusteres en salesweekly

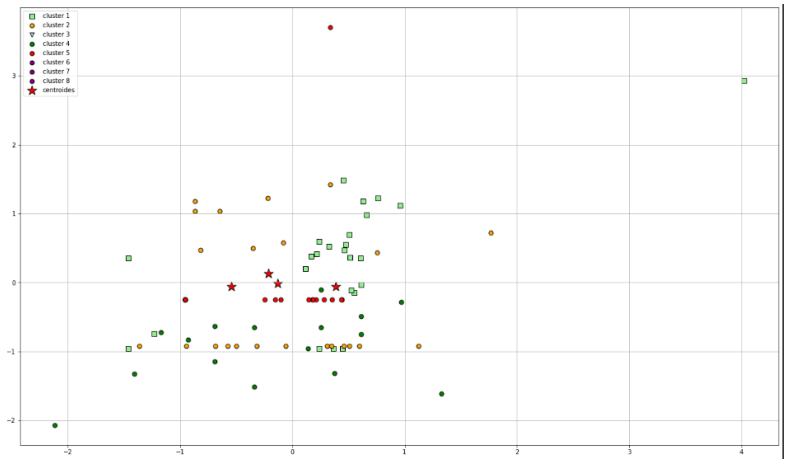


Figura 8. Clusteres en salesmonthly

Análisis preliminar del modelo no supervisado a través de la métrica de evaluación de Elbow

El método de Elbow consiste básicamente en verificar la evolución de la suma de los cuadrados del error para varios valores de K y verificar cual es el que brinda un mejor agrupamiento.

- Saleshourly

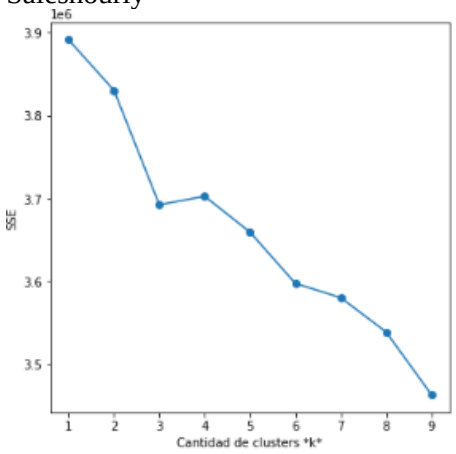


Figura 9. Gráfica de codo

Al realizar un análisis se concluye que la asociación más adecuada es la que está entre los clústeres 2 y 3 ya que se observa mayor homogeneidad de los datos respecto al centroide (ubicación real imaginaria que representa el centro del grupo) en donde los datos se pueden comparar de 3 formas diferentes.

- Salesdaily

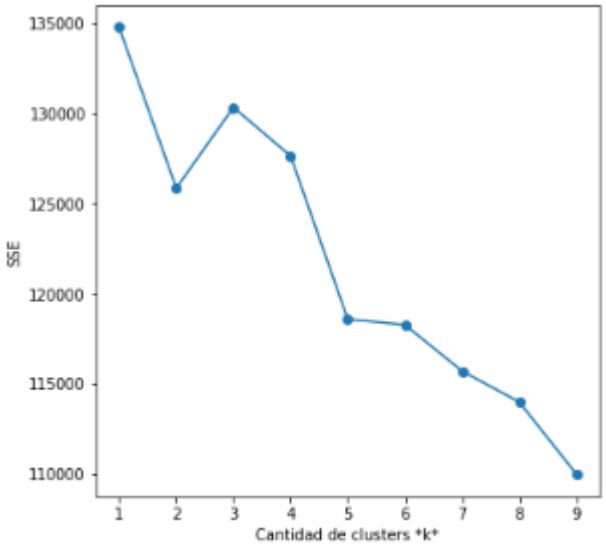


Figura 10. Gráfica de codo

Al realizar un análisis se concluye que la asociación más adecuada es la que está entre los clústeres 1 y 2 ya que se observa mayor homogeneidad de los datos respecto al centroide (ubicación real imaginaria que representa el centro del grupo) en donde los datos se pueden comparar de 2 formas diferentes.

- Salesweekly

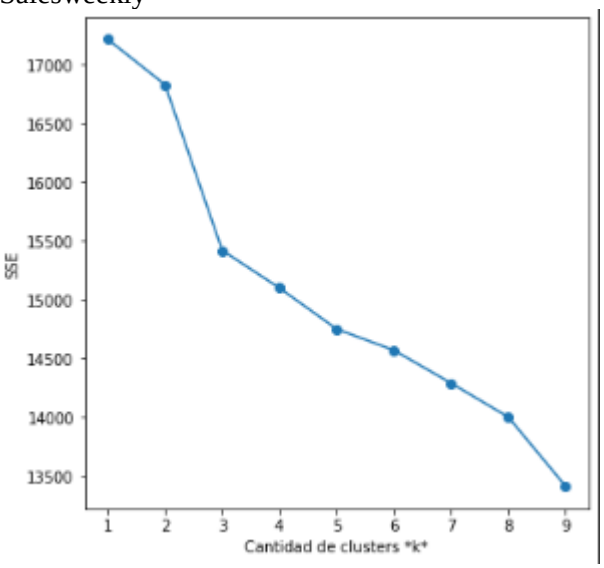


Figura 11. Gráfica de codo

Al realizar un análisis se concluye que la asociación más adecuada es la que está entre los clústeres 2 y 3 ya que se observa mayor homogeneidad de los datos respecto al centroide (ubicación real imaginaria que representa el centro del grupo) en donde los datos se pueden comparar de 3 formas diferentes.

- Salesmonthly

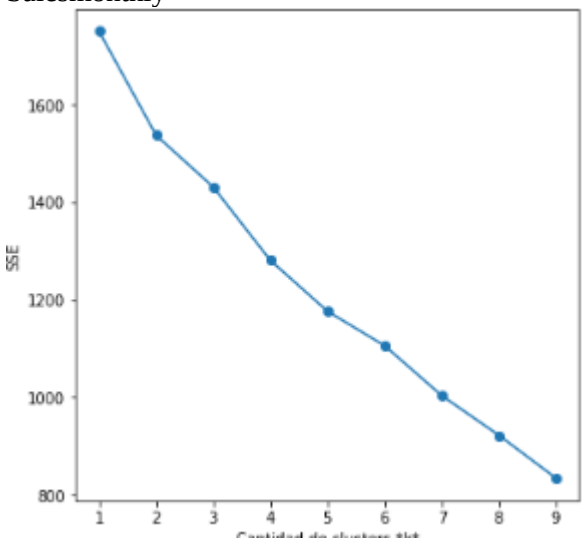


Figura 12. Gráfica de codo

Al realizar un análisis se concluye que la asociación más adecuada es la que está entre los clústeres 1 y 2 ya que se observa mayor homogeneidad de los datos respecto al centroide (ubicación real imaginaria que representa el centro del grupo) en donde los datos se pueden comparar de 2 formas diferentes.

Modelo supervisado

Para este modelo se utilizó clasificador lineal kernel más conocido como máquina de vector de soporte tiene como como tarea las relaciones entre los conjuntos de datos; el kernel lineal es utilizado como un producto normal entre dos observaciones dadas y, el producto entre estos dos vectores es la suma de la multiplicación de cada par de valores de entrada.

Resultado preliminar del modelo no supervisado

En este caso se usan las librerías “sklearn.ensemble” y “sklearn.model_selection”. Permitiendo evaluar la capacidad predictiva de un modelo y comprobar cómo se acercan sus pronósticos a los verdaderos valores de la variable respuesta. Para realizar una cuantificación correcta, se necesita disponer de un conjunto de observaciones de las que se conozca dicha variable. Con esta finalidad se dividen los datos en un conjunto de entrenamiento y un conjunto de prueba, esta segmentación se realiza de forma aleatoria.

```
[ ] X_entrenamiento, X_prueba, y_entrenamiento, y_prueba = train_test_split(datos_Bd_variables, clases, random_state=1)

[ ] print("Tamaño de los datos de entrenamiento = ", X_entrenamiento.shape)
    print("Tamaño de los datos de prueba = ", X_prueba.shape)
    print("Tamaño del vector de clases de entrenamiento = " , y_entrenamiento.shape)
    print("Tamaño del vector de clases de prueba = ", y_prueba.shape)

    Tamaño de los datos de entrenamiento =  (52, 25)
    Tamaño de los datos de prueba =  (18, 25)
    Tamaño del vector de clases de entrenamiento =  (52,)
    Tamaño del vector de clases de prueba =  (18,)

[ ] # Cargar librerias
    from sklearn.ensemble import AdaBoostClassifier
    # Importar Clasificador de Vector de Soporte
    from sklearn.svm import SVC
    # Importar métricas scikit-learn para calculos exactos
    from sklearn import metrics
```

Figura 13. Uso de librerías

A continuación, se muestra cómo se desarrolla el algoritmo AdaBoost

```
# Crear clasificador base
svc = SVC(probability=True, kernel='linear')

# Crear objeto de clasificación AdaBoost
abc = AdaBoostClassifier(n_estimators=2500,learning_rate=0.001)

# Clasificador de entrenamiento AdaBoost
model = abc.fit(X_entrenamiento, y_entrenamiento)

# Predicción de la respuesta para la bd de prueba
y_pred = model.predict(X_prueba)
```

Figura 14. Desarrollo del algoritmo AdaBoost

En la [figura 14] se puede observar el código del modelo y el criterio establecido, así mismo, se logran validar los resultados arrojados.

Se crea un objeto de clasificación AdaBoost, definiendo unos parámetros por defecto qué en este caso se ajusta a 50 estimadores como número máximo en los que finaliza el impulso, el cual se puede reajustar buscando que la exactitud del modelo se acerque a 1. Este proceso itera hasta que los datos de entrenamiento completos se ajustan sin ningún error o hasta que se alcanza el número máximo de estimadores especificado.

El parámetro learning_rate contribuye a los pesos de los estudiantes débiles, se utiliza 1 como valor predeterminado, es decir, la tasa de aprendizaje por definición determina el impacto de cada árbol en la salida y los parámetros controlan la magnitud del impacto. [3] [4]

Posteriormente se agrega el método abc.fit(X_entrenamiento, y_entrenamiento) permite realizar un análisis a partir de las muestras de entrenamiento obtenidas de la división de los datos.

El método predict(X_prueba) permite predecir el resultado del conjunto de datos x para obtener el porcentaje de exactitud utilizando los datos de prueba y el valor de predicción.

Resultados, métricas y curvas de aprendizaje

- **Precisión de** AdaBoost en saleshourly

```
# Exactitud del modelo, qué tan correcto es el clasificador?
print("Accuracy:",metrics.accuracy_score(y_prueba, y_pred))
```

```
Accuracy: 0.6170347502572627
```

Figura 15. Accuracy saleshourly

El conjunto AdaBoost con hiperparámetros predeterminados logra una precisión de clasificación de alrededor del 62 % en este conjunto de datos de prueba.

- **Precisión de** AdaBoost en saleshdaily

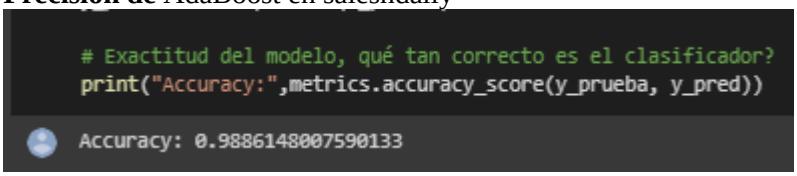


Figura 16. Accuracy salesdaily

El conjunto AdaBoost con hiperparámetros predeterminados logra una precisión de clasificación de alrededor del 99 % en este conjunto de datos de prueba.

- **Precisión de** AdaBoost en salesweekly

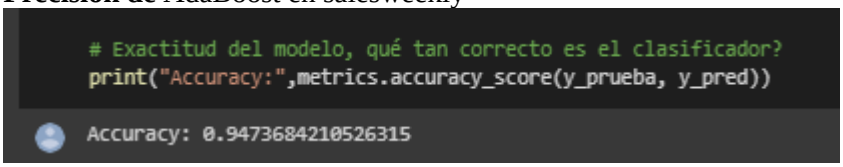


Figura 17. Accuracy salesweekly

El conjunto AdaBoost con hiperparámetros predeterminados logra una precisión de clasificación de alrededor del 95 % en este conjunto de datos de prueba.

- **Precisión de** AdaBoost en salesmonthly

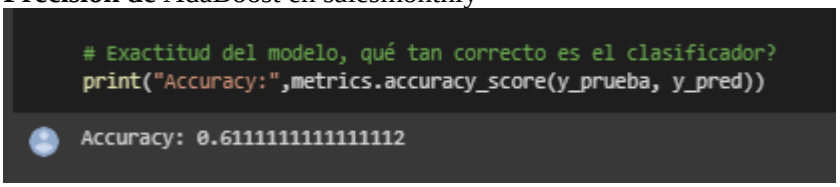


Figura 18. Accuracy salesmonthly

El conjunto AdaBoost con hiperparámetros predeterminados logra una precisión de clasificación de alrededor del 61 % en este conjunto de datos de prueba.

Discusión y Conclusiones

La evaluación de precisión tanto por horas como por meses es impreciso, ya que se comienza con una tasa de aprendizaje en el AdaBoost muy baja de 0.001, haciendo que el modelo sea más estable y, por lo tanto, se podrá controlar la variación en su conjunto de entrenamiento/prueba. Así mismo, tras realizar el ajuste de número de estimadores se puede ver una mayor exactitud del modelo, es decir, a mayor número de estimadores y menor tasa de aprendizaje se llega a un valor óptimo (cercano a 1), para encontrar la métrica de precisión con los datos verdaderos valor que está muy alejado de los datos por horas y por meses Mientras que si se observala evaluación de precisión de los datos por días y semanas se observan valores muy cercanos a 1 lo que garanbtiza altapresisión en la toma de datos.

Con una determinación adecuada y un análisis de los datos representativos se logran alcanzar los resultados usando diferentes herramientas como el One Hot Encoding, el algoritmo de clustering y el algorimo de AdaBoost apoyándonos en la validación de los datos con gráficas y con los diferentes métodos para analizar los resultados y así obtener el óptimo, en nuestro caso el k-means y el accuracy

Con el proceso de recolección, limpieza y segmentación adecuada se logró la construcción de unas fuentes de datos limpias para exploración. Las fuentes de datos procesadas dan pie a la búsqueda de patrones y comportamientos comunes que permitirán caracterizar los accidentes y comprender el fenómeno de ventade fármacos

Referencias Bibliográficas

[1] inec.gob.pa, «CONCEPTOS Y DEFINICIONES,»

[2]M. M. H. RODRIGUEZ, «APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS,» 2015.

[3] A. Navlani, «AdaBoost Classifier in Python,» 20 Noviembre 2018. [En línea]. Available: https://www.datacamp.com/community/tutorials/adaboost-classifier-python. [Último acceso: 19 octubre 2022].

[4] p. clic, «[Aprendizaje automático] Resumen de las ventajas y desventajas de cada método de aprendizaje integrado,» [En línea]. Available: https://programmerclick.com/article/18251647130/. [Último acceso: 19 octubre 2022].