Proyecto 2: Aprendizaje No Supervisado

Andres Saldaña Rodriguez
A01721193@tec.mx
Tecnológico de Monterrey
Ingeniería en Ciencias de Datos y Matemáticas
Monterrey, Nuevo León, México

Lester Santiago García Zavala
A01721128@tec.mx
Tecnológico de Monterrey
Ingeniería en Ciencias de Datos y Matemáticas
Monterrey, Nuevo León, México

Antonio Jose Zarate Lozano
A01720990@tec.mx
Tecnológico de Monterrey
Ingeniería en Ciencias de Datos y Matemáticas
Monterrey, Nuevo León, México

ABSTRACT

En este artículo nos encontramos con la implementación de un modelo de aprendizaje no supervisado para poder crear grupos y clusterizar datos de pacientes que pueden o no tener diabetes con sus respectivos indicadores de salud. Lo que se hizo fue analizar la base de datos de diabetes. Se elaboraron los modelos y se hicieron los segmentos de los datos en base al método de K-Means. Durante este artículo se verá el contexto del problema, la metodología usada, los conceptos importantes. Al igual que, los resultados de tanto el análisis exploratorio de la base de datos como de los modelos en sí, con sus debidas conclusiones.

ACM Reference Format:
Andres Saldaña Rodriguez, Lester Santiago García Zavala, and Antonio Jose Zarate Lozano. 2022. Proyecto 2: Aprendizaje No Supervisado. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCCIÓN

En este artículo nos encontramos con un reporte del proyecto 2 de aprendizaje no supervisado. En este artículo se revisara el dataset de diabetes, se planteará un problema y una solución usando aprendizaje no supervisado.

El aprendizaje no supervisado es un tipo de aprendizaje de machine learning en el cual un modelo tratará de procesar datos sin etiquetas, datos en bruto para poder después con el algoritmo escogido asignarle una etiqueta en base a patrones. [4]

En este artículo nos encontramos con la problemática que gira al rededor de un dataset del diabetes. Por lo tanto, nos encontramos con más de 700 pacientes que buscaremos agrupar en base a sus diferentes atributos para poder ver si los grupos que salen al agrupar pacientes similares y forma 2 grupos basado en los patrones para la gente que tiene o no diabetes. Después será ver si los grupos que el modelo asignó en base a esos atributos, se parecen a los datos reales de las pacientes que tienen y que no tienen diabetes.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Por lo tanto, se tratará de agrupar los datos de las pacientes en base a los atributos del dataset para que agrupe los que comparten similitudes. La ventaja del dataset es que se podrá comparar los datos originales con las agrupaciones que crea el modelo, los datos originales siendo las pacientes con diabetes, y compararlo con como el modelo los agrupa en base a los atributos previamente mencionados.

Previamente en el 2018 se realizó un estudio en el cual la finalidad última era crear un modelo que usando clustering pueda analizar los atributos que muestran pacientes de diabetes para poderlos agrupar, y de hecho Miguel Fuentes López y Lucas Segarra Fernández (los autores) usan sklearn para realizar las agrupaciones usando k-means (página 29). [1] El objetivo del trabajo mencionado era agrupar pacientes con similitudes para poder monitorearlos de forma adecuada.

1.1 Contexto

La diabetes es una enfermedad crónica, la cual los afectados pierden la capacidad de regular los niveles de glucosa en su sangre. Esto se debe a que pierden en control parcial o total de su producción de insulina. La insulina es una hormona, que se produce en nuestro cuerpo naturalmente. La insulina ayuda a que la glucosa entre nuestras células nos ayude a producir energía. Es como si la insulina fuera una llave que abre una puerta hacia nuestras células, dejando pasar a la glucosa y ayudándonos a producir energía. [5]

Existen 2 tipos de personas diabéticas. Las personas diabéticas tipo 1, las cuales no pueden producir nada de insulina, es decir no tiene llaves para dejar pasar a la glucosa. Las personas diabéticas tipo 2 sí producen insulina, pero esta no funciona de manera adecuada. Es decir es como si el cuerpo produjera llaves pero estas estuvieran dobladas y no pudieran abrir las puertas. También existen distintas variaciones de diabetes tipo 2, pero estos ya son más específicos y poco comunes.

El exceso de glucosa que se encuentra en la sangre a causa del déficit de insulina, puede causar muchos problemas. Algunos pueden ser leves, como sentirse cansado, estar sediento, perder peso, etc. Sin embargo, tiene complicaciones más severas que pueden llevar a afectar el corazón, los riñones, los ojos, pies, etc. Estas complicaciones pueden causar la muerte.

El "dataset" que estaremos utilizando fue recavado de mujeres de "Prima" India, y las personas que se encuentran dentro del dataset son todas mujeres, las cuales todas eran mayores a 21 años.

2 CONCEPTOS PREVIOS Y SU DESCRIPCIÓN

- **Análisis exploratorio de datos:** El análisis exploratorio de datos consiste en mostrar los estadísticos de cada atributo de la base de datos.
  - **Conteo de datos:** Cantidad de instancias de cada atributo.
  - **Promedio o Media:** Se suman todos todos los valores y se divide en el conteo.
  - **Desviación estándar:** Cuanto varían los datos de la media.
  - **Mínimo:** Valor mínimo.
  - **Máximo:** Valor máximo.
  - **1er Cuartil:** Determinado por el valor que indica que los valores menores a ese pertenecen al 25 por ciento de los datos.
  - **2o Cuartil:** Determinado por el valor que indica que los valores menores a ese pertenecen al 50 por ciento de los datos.
  - **3er Cuartil:** Determinado por el valor que indica que los valores menores a ese pertenecen al 75 por ciento de los datos.
- **Correlación:** Indica que tan fuerte es una tendencia lineal entre 2 variables. También si es positiva o negativa.
- **Cluster:** Agrupación de datos.
- **K-Means:** Método de aprendizaje no supervisado para clusterización y agrupación de datos. Funciona a partir de centroides los cuales son puntos iniciales de referencia al centro de los datos. Después se obtiene la distancia euclidiana de cada punto con los centroides. La distancia mínima de los datos al centroide será la que se tomará como referencia para nombrar y asignar el cluster a ese dato. [3]
- **Silhouette Score:** Indica el número óptimo de clusters. [2]

3 METODOLOGÍA

3.1 Descripción general del dataset

Nuestro "dataset" descargado de "Kaggle" es acerca de la enfermedad del diabetes. Cabe mencionar que la información de nuestro "dataset" es solamente de mujeres mayores a 21 años. Dicho "dataset" contiene la columna binaria llamada "outcome" la cual sirve como la clase ya que nos indica si la paciente tiene o no diabetes y 8 atributos con 768 instancias. Encontramos atributos que servirán para el proyecto del aprendizaje no supervisado elaborar modelos de inteligencia artificial que nos puedan predecir con alta exactitud si la paciente tiene o no diabetes.

3.2 Describir cada uno de los atributos

En nuestro "dataset" encontramos 8 atributos de los cuales creemos que todos son de vital importancia para realizar el proceso de aprendizaje no supervisado ya que nos aportan información importante para determinar porque la encuestada tiene o no diabetes.

Los 8 atributos que utilizaremos son los siguientes: Pregnancies el cual nos indica la cantidad de embarazos que ha tenido la encuestada. Glucose para determinar la concentración de glucosa en la sangre de la paciente. BloodPressure el cual nos proporciona la presión arterial. SkinThickness este atributo nos indica el grosor de la piel del tríceps. Insulin el cual nos da los niveles de insulina de las pacientes. BMI es el índice de masa corporal el cual nos puede proporcionar si tienen obesidad o no. DiabetesPedigreeFunction nos indica la probabilidad de tener diabetes dependiendo de su árbol familiar. El último atributo es Age el cual es la edad que tiene cada una de las encuestadas y nos ayudará a determinar en que rango de edad la diabetes es más común.

3.3 ¿Qué conocimiento queremos extraer?

Con el dataset mencionado se busca extraer agrupación de los datos en base de los atributos para poder clasificar las pacientes como diabetes o no diabetes en base a los atributos de BloodPressure, BMI, Diabetes Function, Age, BMI y Glucose. Con estos se generaran 2 clusters, después se comparará con los datos originales que ya tienen un cluster predefinido como el atributo de Outcome que determina si alguien tiene diabetes o no.

3.4 Proceso a seguir para la extracción del conocimiento.

(1) Investigar la base de datos en Kaggle.
(2) Descargar la base de datos.
(3) Definir la problemática a solucionar.
(4) Cargar las librerías necesarias.
(5) Cargar la base de datos en una libreta de Jupyter.
(6) Realizar análisis exploratorio de datos para conocer la base de datos.
(7) Definir las variables que entraran al modelo.
(8) Probar los modelos.
(9) Usar el Silhouette Score para ver el número óptimo de clusters.
(10) Modificar el modelo.
(11) Observar los resultados.
(12) Comparar con lso grupos que venían desde el dataset.
(13) Concluir.

3.5 Metodología CRISP en el proyecto

La metodología CRISP-DM, es el modelo analítico más usado dentro del área de la minería de datos. Ya que explica los pasos a seguir, desde que se llega con una organización o empresa, hasta que se entrega un modelo de "machine learning" o inteligencia artificial.

Se empieza con el entendimiento de los datos, y el entendimiento de la empresa. Es decir que tenemos que lograr comprender cómo es que opera la empresa, cuál es el trabajo que hace,y como los datos que nos están proporcionando se relacionan a esto. Esta etapa es muy importante, y de esta depende que podamos establecer objetivos y metas claras y alcanzables. En pocas palabras les da contexto a los datos.

Luego pasamos a la fase de preparación de datos, en esta fase es una de las más tardadas, ya que nos tenemos que asegurar de limpiar los valores nulos que encontremos, analizar la distribución de los datos, ver que tan balanceados están, eliminar datos que no nos vayan a servir, escalar los datos, etc. Básicamente nos tenemos que asegurar de que los datos que nos dio la empresa sean aptos para los modelos que vayamos a usar en los siguientes pasos. Aquí

tiene mucho que ver el concepto "garbage in, garbage out", es decir que si le metes basura a los modelos, estos van a modelar basura. Con esto en mente es que debemos de explorar y asegurarnos de que los datos que estaremos usando en los modelos sean de la mejor calidad posible.

Finalmente pasamos a la fase de la modelación de los datos. Ahora si los datos que estuvimos limpiando y preparando anteriormente se pasan a los modelos. Se entrenan con estos datos, y luego los probamos para poder evaluar el desempeño de los modelos.

Finalmente tenemos la fase de la entrega del modelo. En esta fase se puede entregar algún tipo de aplicación al cliente, o algo similar, para que ellos puedan utilizar el modelo cuantas veces quieran, con nuevos datos. Es muy importante estar seguros que el modelo pase todas las métricas establecidas en la evaluación y que cumpla con el propósito original antes de entregárselo al cliente. Después se le puede hacer mantenimiento o incluso seguir mejorándolo, pero esto ya no es parte de la metodología CRISP-DM.

4 ANÁLISIS EXPLORATORIO DE DATOS

Antes de poder realizar los modelos tuvimos que realizar el EDA de nuestra base de datos para poder conocer la base de datos y todo lo que contiene. Primeramente nos dimos cuenta como se muestra en la siguiente figura que los datos no están balanceados ya que podemos encontrar que 500 de las pacientes no tuvieron diabetes mientras que 268 si tuvieron. También encontramos que todos los datos son numéricos por lo que sera posible utilizarlos en nuestros modelos.

```
In [3]: data.describe()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

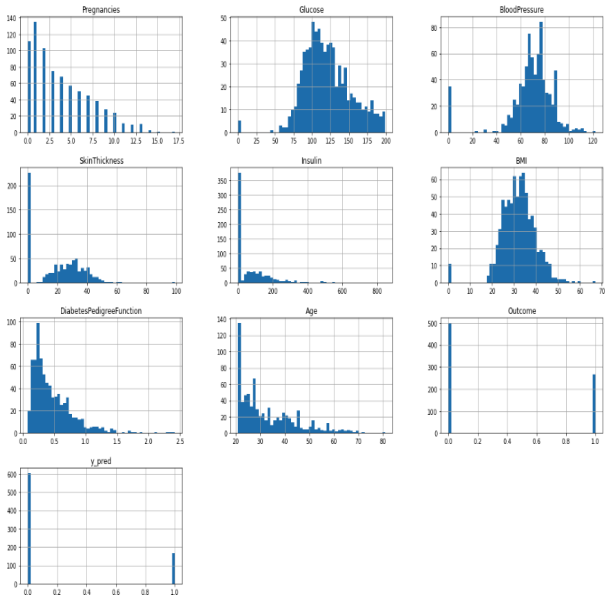
```
In [22]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
 9   y_pred                    768 non-null    int32
dtypes: float64(2), int32(1), int64(7)
memory usage: 57.1 KB
```

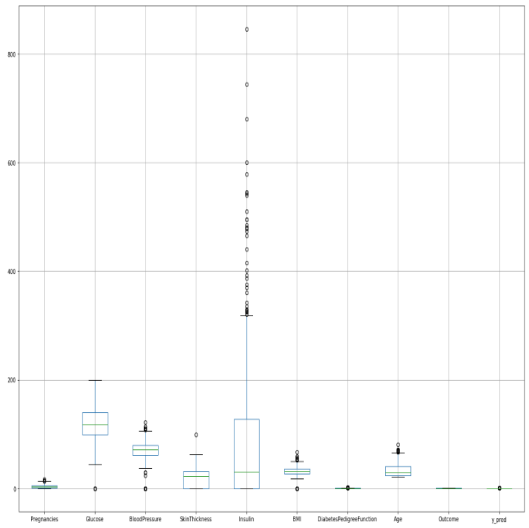
```
In [23]: data["Outcome"].value_counts()
```

```
Out[23]: 0    500
         1    268
         Name: Outcome, dtype: int64
```

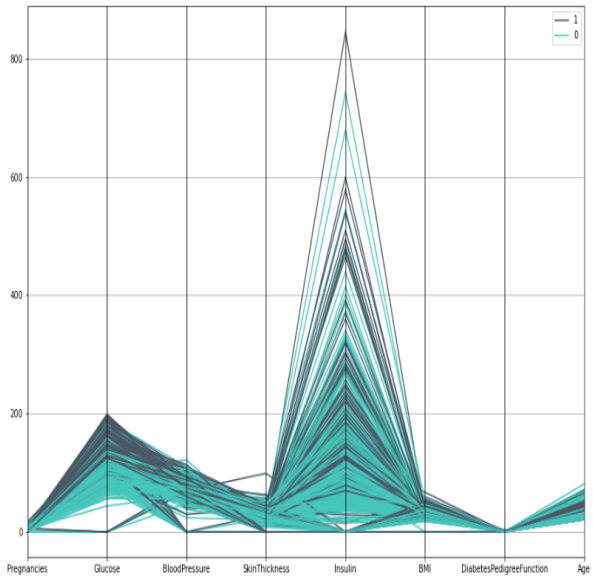
Después de lo anterior realizamos gráficos de nuestra data para poder observar como se comportaban y como estaban distribuidos sus valores. En los histogramas podemos observar que en muchos de los atributos los datos no están ajustados de muy buena manera ya que la mayoría de sus datos se encuentran a la izquierda de la gráfica. Por otro lado en los atributos de BloodPressure, BMI y Glucose los datos se ajustan de buena manera pues su promedio se encuentra cerca de la mediana además de estar distribuidos normalmente.



Lo mencionado anteriormente también puede ser observado utilizando la gráfica de boxplot en donde claramente se ve como algunos de los datos tienen muchos valores atípicos además de un rango muy grande de datos. Esto se debe principalmente a que muchos de sus datos se encuentran en valores muy pequeños. Además en dichos gráficos también se puede observar que las cajas las cuales son creadas a partir de la diferencia entre el primer y tercer cuartil están muy pequeñas y esto se debe a lo que se menciono con anterioridad de los valores de sus datos. Cabe mencionar que podemos observar una cantidad inmensa de datos atípicos en la insulina por lo que puede llegar a ser una variable engañosa al realizar nuestro modelo.

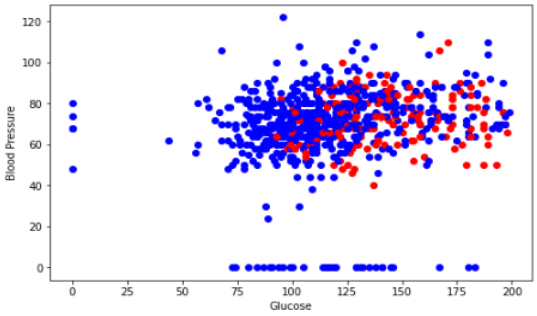


Por ultimo creamos un parallel coordinates plot para poder comparar nuestras variables y encontrar la relación que tiene cada una de ellas con la diabetes. Gracias a dicho gráfico encontramos que la mayoría de las pacientes que cuentan con valores muy altos Glucose y tienen un BMI alto padecían de diabetes. Además algo que nos sorprendió mucho fue la edad de las pacientes que padecían de diabetes pues la mayoría de ellas se encontraban cerca de la mediana de los datos que era alrededor de 29 años.

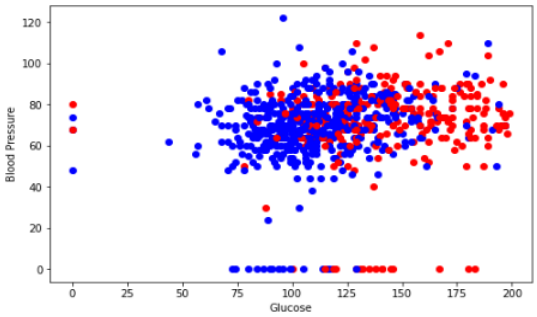


Glucose and Blood Pressure

```
plt.figure(figsize= (8,5))
colors = {0: "blue", 1: "red"}
for i in range(len(data2)):
    plt.scatter(data2[i][1], data2[i][2], color=colors[y_pred
for i in range(len(kmeans.cluster_centers_)):
    plt.scatter(kmeans.cluster_centers_[i][1], kmeans.cluster
plt.xlabel("Glucose")
plt.ylabel("Blood Pressure")
plt.show()
#prediccion
```




```
plt.figure(figsize= (8,5))
colors = {0: "blue", 1: "red"}
for i in range(len(data2)):
    plt.scatter(data2[i][1], data2[i][2], color=colors[data['
plt.xlabel("Glucose")
plt.ylabel("Blood Pressure")
plt.show()
#original
```



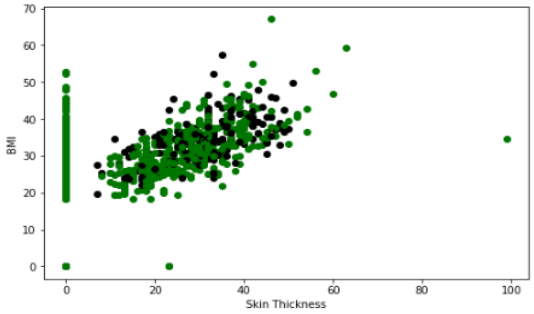
En esta primera imagen podemos ver la presión sanguínea comparada con la glucosa. La gráfica de arriba esta coloreada de acuerdo a K-means y la gráfica de abajo esta coloreada de acuerdo a la columna original del dataset. Podemos ver un coloreo similar en ambas, sin embargo notamos diferencias en los "outliers" que se encuentran en la parte sur y oeste, ya que el algoritmo de Kmenas no logra diferenciarlos muy bien.

5 RESULTADOS

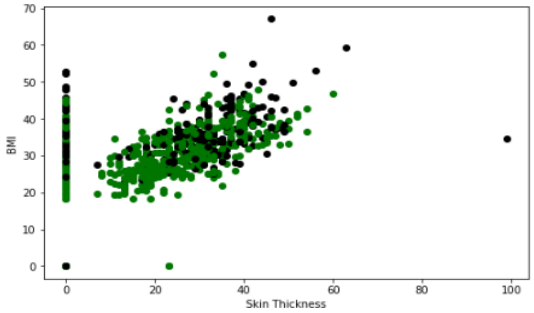
Tras haber entrenado el modelo, decidimos que la mejor manera de medir que estuviéramos cumpliendo nuestro objetivo, fue al comparar gráficas. Es decir, graficamos distintos pares de columnas en un gráfico de dispersión, y luego coloreamos los puntos de acuerdo a la clasificación que resulto de nuestro algoritmo Kmeans, y esto lo comparamos con la clasificación original, que ya venia con el "dataset".

Skin Thickness and BMI

```
plt.figure(figsize= (8,5))
colors = {0: "green", 1: "black"}
for i in range(len(data2)):
    plt.scatter(data2[i][3], data2[i][5], color=colors[y_pred
for i in range(len(kmeans.cluster_centers_)):
    plt.scatter(kmeans.cluster_centers_[i][3], kmeans.cluster
plt.xlabel("Skin Thickness")
plt.ylabel("BMI")
plt.show()
#prediccion
```



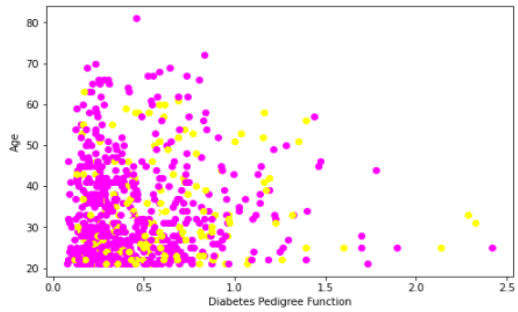

```
plt.figure(figsize= (8,5))
colors = {0: "green", 1: "black"}
for i in range(len(data2)):
    plt.scatter(data2[i][3], data2[i][5], color=colors[data["
plt.xlabel("Skin Thickness")
plt.ylabel("BMI")
plt.show()
#original
```



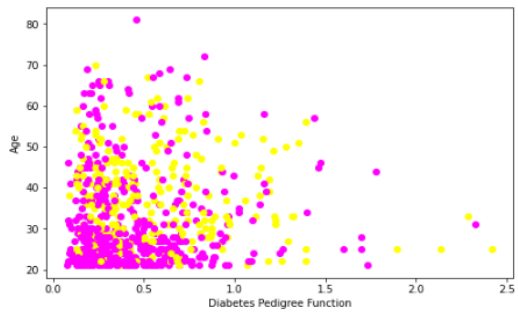
En esta otra gráfica de grosura de la piel con índice de masa corporal, podemos ver las mismas tendencias. EL k menas batallo un poco con los "outliers" a la izquierda de la gráfica, sin embargo a simple vista pereciera ver que hizo un buen trabajo con el conjuntó de datos que se forma en medio. Las partes verdes son muy parecidas.

Diabetes Pedigree Function and Age

```
plt.figure(figsize= (8,5))
colors = {0: "magenta", 1: "yellow"}
for i in range(len(data2)):
    plt.scatter(data2[i][6], data2[i][7], color=colors[y_pred[
for i in range(len(kmeans.cluster_centers_)):
    plt.scatter(kmeans.cluster_centers_[i][6], kmeans.cluster_
plt.xlabel("Diabetes Pedigree Function")
plt.ylabel("Age")
plt.show()
#prediccion
```



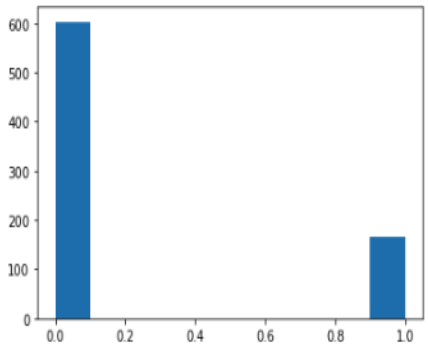

```
plt.figure(figsize= (8,5))
colors = {0: "magenta", 1: "yellow"}
for i in range(len(data2)):
    plt.scatter(data2[i][6], data2[i][7], color=colors[data["O
plt.xlabel("Diabetes Pedigree Function")
plt.ylabel("Age")
plt.show()
#original
```



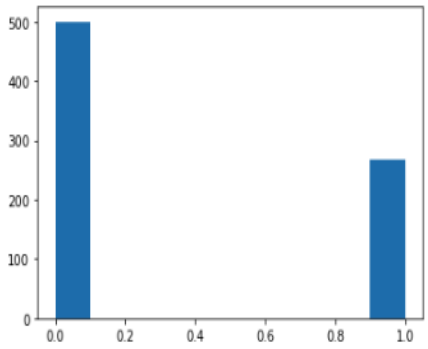
Esta ultima imagen de la funcion de diabetes contra edad, es en la que se puede ver una diferencia más grande entre entre las clasificaciones de Kmeans y las originales. En la gráfica de abajo de las clasificaciones originales podemos ver muchos más puntos amarillos, mientras que en la gráfica arriba de kmeans vemos predominantemente puntos rosas. Sin embargo siguen reteniendo cierta similitud, solo que no es tanta como las gráficas anteriores.

Es importante mencionar que en el dataset original había alrededor de 500 0s o personas que no eran diabéticas y 250 1´s es decir personas que si eran diabéticas. Mientras que de a cuerdo con las clasificaciones del kmeans, había alrededor de 600s 0s y 150 1s, aunque esto no indica necesariamente que un grupo sea de diabético y no diabéticos. Solo nos muestra la magnitud de los grupos con características similares que hizo kmeans.

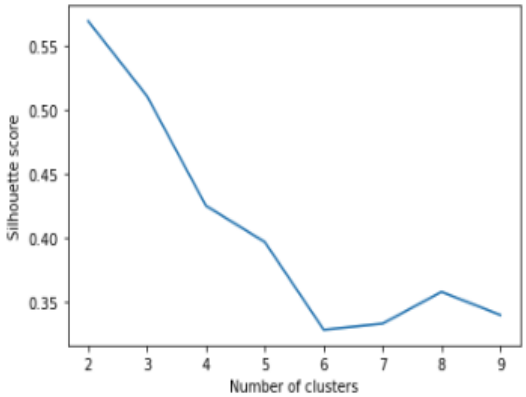
```
In [17]: plt.hist(data["y_pred"])
Out[17]: (array([603.,   0.,   0.,   0.,   0.,   0.,   0
          array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
          <BarContainer object of 10 artists>)
```




```
In [18]: plt.hist(data["Outcome"])
Out[18]: (array([500.,   0.,   0.,   0.,   0.,   0.,   0
          array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
          <BarContainer object of 10 artists>)
```



Por ultimo checamos que hayamos usado el valor correcto de k, o el óptimo, para el algoritmo Kmeans utilizando el "silhouette score". El cuando cuando nos graficamos nos dice que se obtienen los mejores resultados cuando k = 2.



6 CONCLUSIONES

Considerando en análisis visual y la interpretación de las gráficas mencionadas arriba, creo que podemos decir que fue un éxito el algoritmo de aprendizaje kmeans en su tarea de agrupar personas similares. Aunque no tenemos una presentación exacto "D, porque usamos 8 columnas para la formación de "clusters", mediante las gráficas de pares de atributos logramos identificar algunas de los atributos los cuales fueron más efectivos para la clasificación "correcta" de los grupos. Estas siendo la glucosa, la presión sanguínea, la grosura de la piel y el índice de masa corporal.

Cabe mencionar que aunque pudiéramos hacer una matriz de confusión ya que contamos con los datos originales de predicción correctos, no lo haremos, porque este no es el enfoque del proyecto. Simplemente cumplimos con el agrupamiento de los dos grupos, los cuales sospechamos que sean de personas diabéticas y no diabáticas, sospechas las cuales levantamos con l¡nuestro análisis visual de las gráficas.

También es importante decir que este proyecto se podría seguir estudiando dado más tiempo y recursos, ya que existen más combinaciones de gráficas que igual nos demuestren cosas nuevas y nos hagan pensar diferentes. Sin embargo, esta es solo una sugerencia para una investigación a futuro.

6.1 Reflexiones

6.1.1 ***Andrés Saldaña.*** - Aunque desde un principio entendiera cual era en concepto la función de los algoritmos de aprendizaje no supervisado, realmente me costo aplicarlos en la vida real. NO porque fueran difíciles de modelar, si no porque no entendía cual era en análisis que se les estaba haciendo, ni que era lo que representaban los clusters que se formaban, tampoco entendía muy bien la multidimensional de los datos. Sin embargo después de muchas pruebas y errores, finalmente logre comprender a estos algoritmos. Aunque si puede hacer una predicción de 2 dimensionas de un datset que solo tenga 2 columnas, nosotros no buscábamos eso, y eso por esto que me revolví tanto. Me costo comprender las implicaciones de la multidimensional del dataset, y como analizarlo. Sin embargo finalmente entendí que la mejor manera de verla era con los colores, e intentar hacer hipótesis y predicciones de que era lo que significaban. También fue muy útil, tener los resultados reales de las personas que eran y no diabéticas, ya que esto nos dio una comparativa muy útil para ver si íbamos por un buen camino o no. Fue interesante ver en nuestro dataset anterior como un variable tenia casi todo el peso al momento de hacer los clusters, y como las gráficas se veían afectadas por estos, ya que había un alinea que las recortaba muy abruptamente. De todas manera siento que equivocarnos nos ayudo a aprender y comprender mejor la idea del proyecto, y al final acamamos muy contentos con nuestro resultado.

6.1.2 ***Lester García.*** - Al principio de este proyecto realmente no entendía la diferencia entre el aprendizaje supervisado y no supervisado pero al ir realizando el proyecto al igual que con la ayuda de mis compañeros poco a poco me empezó a quedar más claro. Me di cuenta que en el caso de este proyecto que era sobre aprendizaje no supervisado era a partir de datos no etiquetados

para agruparlos para encontrar grupos similares en este caso para predecir la diabetes a partir de los diferentes atributos. Además al contar con los datos reales de diabetes y no diabetes pudimos comparar nuestras respuestas para saber que tan cerca estaban con la realidad de los datos. Además de poder utilizar nuestros modelos para que si llegan datos de una nueva paciente podamos agruparlo y determinar si tiene o no diabetes dependiendo en este caso a partir de su BMI, skin Thickness, Glucose, blood pressure y age ya que fueron las que se utilizaron para crear las gráficas. Cabe mencionar que fue un largo y tardado proyecto ya que al anteriormente contar con un dataset de valores binarios no nos era posible realizar los clusters como debían de ser por lo que se tuvo que cambiar de dataset para poder realizar el proyecto de manera correcta. Esta equivocación me ayudo a entender el aprendizaje no supervisado ya que claramente el modelo de K-means no puede determinarse a partir de un solo atributo sino de dos o más por lo que no solo pude realizar el proyecto con éxito sino también pude comprender sobre la aplicación de dicho aprendizaje.

6.1.3 ***Antonio Zárate***. - Con este proyecto de aprendizaje no supervisado pude ver la diferencia entre los 2 tipos de aprendizajes vistos en clase. Muchas veces pensamos que machine learning tiene que tener datos para entrenar el modelo que ya tengan etiquetas, pero en el mundo real la mayoría de los datos no tienen etiquetas. Por lo tanto, hacer este proyecto me ayudo a darme cuenta de un ejemplo de lo que son los datos en verdad, número continuos que al agruparse se puede encontrar un patrón, en este caso que con los atributos se creen 2 grupos y trate de agrupar diabetes y no diabetes basado en los patrones que observó el K-Means de la presión, el BMI, la glucosa y la insulina. Además pude observar el valor de los modelos de agrupación en el sector de la salud, porque en mi caso lo he usado en el sector de la economía con análisis de elasticidades para agrupar territorios que comparten similitud de volumen y de elasticidad para segmentar precio. Pero el poder ver el K-Means aplicado a otro sector se me hizo súper valioso. Además que pude complementar mis habilidades como científico de datos. Esto al agregar el conocimiento de como funcionan los modelos de clustering del aprendizaje no supervisado a mis habilidades.

REFERENCES

[1] Fuentes López, M., and Segarra Fernández, L. Identificación de patrones de glucemia por tramos de cuatro horas en diabéticos tipo i mediante monitorización continua de glucosa y técnicas estadísticas, 2018.

[2] Shahapure, K. R., and Nicholas, C. Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* (2020), IEEE, pp. 747–748.

[3] Sinaga, K. P., and Yang, M.-S. Unsupervised k-means clustering algorithm. *IEEE access 8* (2020), 80716–80727.

[4] Team, D. S., and Team, D. S. Aprendizaje no supervisado - aprendizaje automático, Dec 2020.

[5] UK, D. Diabetes: The basics.