CODERHOUSE

Proyecto: Diabetes

Realizado por: Florencia Paz Ponce

Curso: Data Science

Comisión: 48610

CODERHOUSE

Indice:

Introducción 3
Dataset 4
Objetivo 5
Motivación y Audiencia 6
Contexto Comercial e Hipótesis 7
Problemática Comercial 8
Data Wrangling 9
Procesamiento de Datos 10
Conclusiones (Gráficos) 11
Machine Learning 17
Métricas 20
Conclusiones (Métricas) 22
Cross Validation 23
Evaluación: Grilla de Hiperparámetros 24

CODERHOUSE

Introducción:

El dataset elegido trata sobre la ***diabetes***, una enfermedad crónica que afecta la forma en que el cuerpo convierte los alimentos en energía.

El cuerpo descompone la mayor parte de los alimentos que come en azúcar y los libera en el torrente sanguíneo.

CODERHOUSE

Dataset:

Campos y Descripciones:

Edad: Categoría de edad de 13 niveles 0= 0-18 1 = 18-24 9 = 60-64 13 = 80 o más

Sexo: M (masculino) / F (femenino)

Colesterol_Alto: Si la persona tiene o no colesterol alto

Chequeo_de_Colesterol: Si en los últimos 5 años la persona se hizo o no chequeo del colesterol

Masa_Muscular: BMI (el indice de masa muscular de la persona)

Fumador: Si la persona en su vida fumó al menos 100 cigarrillos en su vida (5 cajas aproximadamente)

Problemas_de_corazón: Si la pesona tuvo enfermedad coronaria (CHD) o infarto de miocardio (IM)

Actividad_Física: Si la persona realizó actividad física o no en los ultimos 30 dias.

Consumo_de_frutas: Si consume 1 o mas frutas por dia

Consumo_de_vegetales: Si consume 1 o mas vegetales por dia

Alto_Consumo_de_Alcohol: Se considera consumo alto de alochol si (hombres adultos >=14 tragos por semana y mujeres adultas >=7 tragos por semana)

Salud: Como se consideran en la escala de salud las personas escala (1-5) 1 = excelente 2 = muy buena 3 = buena 4 = regular 5 = mala

Mala_Salud_Mental/Dias dias en los que la persona considera que no se encuentra bien mentalmente.

Dias_enfermo días de enfermedad o lesión física en los últimos 30 días (escala 1-30)

Dificultad_para_caminar Si la persona siente que tiene difultad al caminar o subir escaleras.

Derrame_Cerebral Si la persona sufrió un derrame cerebral o no.

Hipertensión: Si la persona es o no hipertensa

Diabetes: Si la persona tiene o no diabetes

CODERHOUSE

Objetivo:

Descubrir cuales son los factores que están contribuyendo a que las personas contraigan la enfermedad de diabetes.

El objetivo es poder analizar y predecir, en base a la calidad de vida que llevan distintas personas, cuales son aquellos que tienen mayor probabilidad de tener diabetes y asi poder prevenirlo.

CODERHOUSE

Motivación y Audicencia:

Elegí este dataset porque me parece que es una enfermedad que tienen muchas personas y considero que es interesante buscar una manera de mejorar la vida de estos pacientes teniendo en cuenta cuales son los factores que hacen que dicha enfermedad se agrave.

CODERHOUSE

Contexto Comercial e Hipótesis:

Soy una científica de datos de un hospital muy reconocido, el cual está teniendo muchos casos de pacientes enfermos que tienen determinadas patologías y diferentes calidades de vida pero el factor en común que encontraron los médicos es que muchos de ellos manifiestan la enfermedad de Diabetes.

El hospital plantea la ***hipótesis*** de que esto se debe a malos hábitos, alimentación y otros factores de salud que perjudican al paciente por lo tanto deciden estudiar estos casos para ver cual es la mejor manera para ayudar a los pacientes.

CODERHOUSE

Problemática Comercial:

Basados en la información que nos proporciona el dataset:

¿Cuales de las patologías que poseen los pacientes están relacionadas con la enfermedad de diabetes?

Si realizamos actividad física y comemos saludablemente ¿disminuye la probabilidad de tener Diabetes?

¿Cuales son los rangos etarios más afectados?

CODERHOUSE

Data Wrangling:

Limpieza del dataset:

1. **Verificación de nulos:** No contenía

2. **Verificación de duplicados:** No contenía

3. **Cambio de nombre de las columnas realizado anteriormente con el fin de que sea mas fácil identificarlas visualmente:**
Se realizó el cambio ya que los nombres de las columnas y los datos se encontraban en inglés y los pasé a español.

4. **Cambio de las filas del dataset de "texto" a "numérico" para poder para poder realizar posteriormente nuestro arbol de decisión.** (Se encuentra en forma mas detallada en la diapositiva “Procesamiento de Datos”)

CODERHOUSE

Procesamiento de Datos:

1. Reemplacé los valores de "Ninguno" en las columnas Dias_Enfermo y Mala_Salud_Mental/Dias por “0” porque para poder generar el modelo de ML necesito tener todos los valores en forma numérica

2. Una vez realizado el primer paso, cambié toda la tabla de texto a números

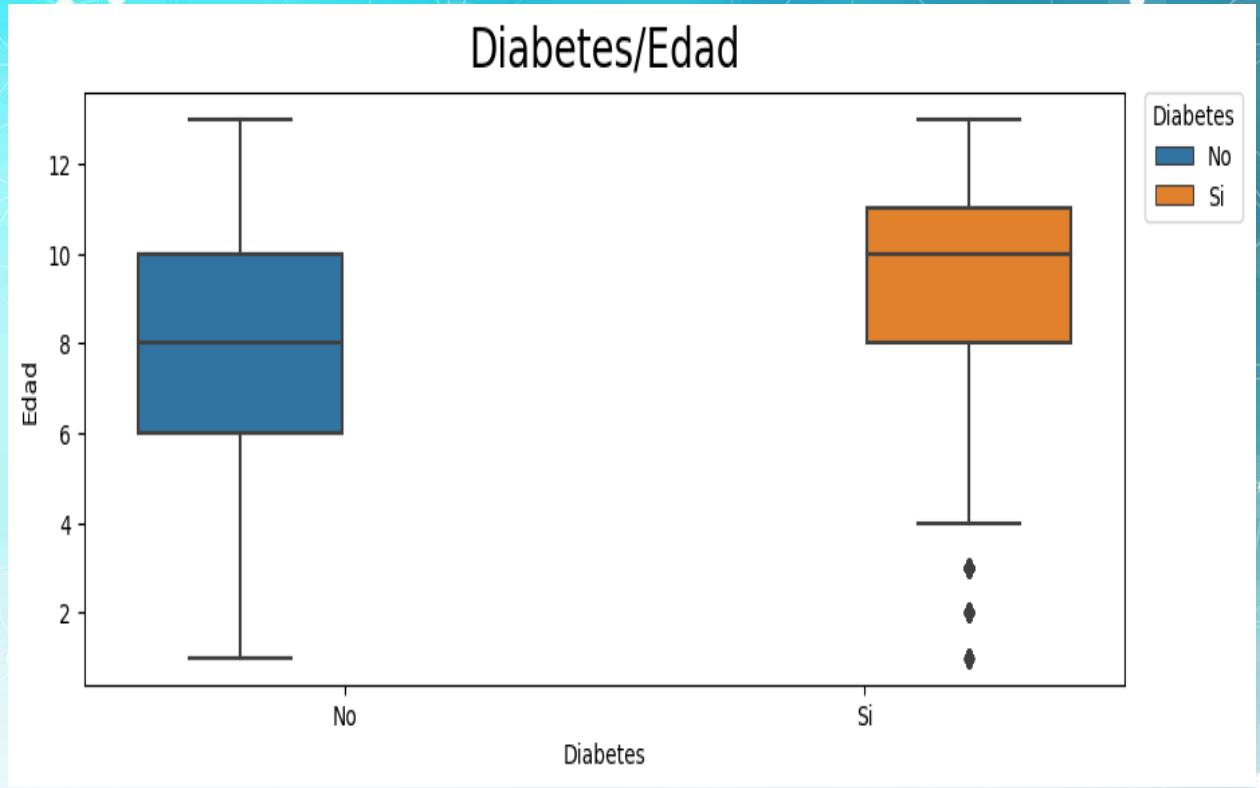
3. Y por último antes de comenzar a realizar el modelo de ML eliminé las columnas que no necesitaba para este proceso.

CODERHOUSE

Conclusiones (Gráficos):

Según los graficos realizados hemos visto que la **edad** explica parte de la relación con la enfermedad, ya que hay un porcentaje mayor que corresponde a las personas adultas.

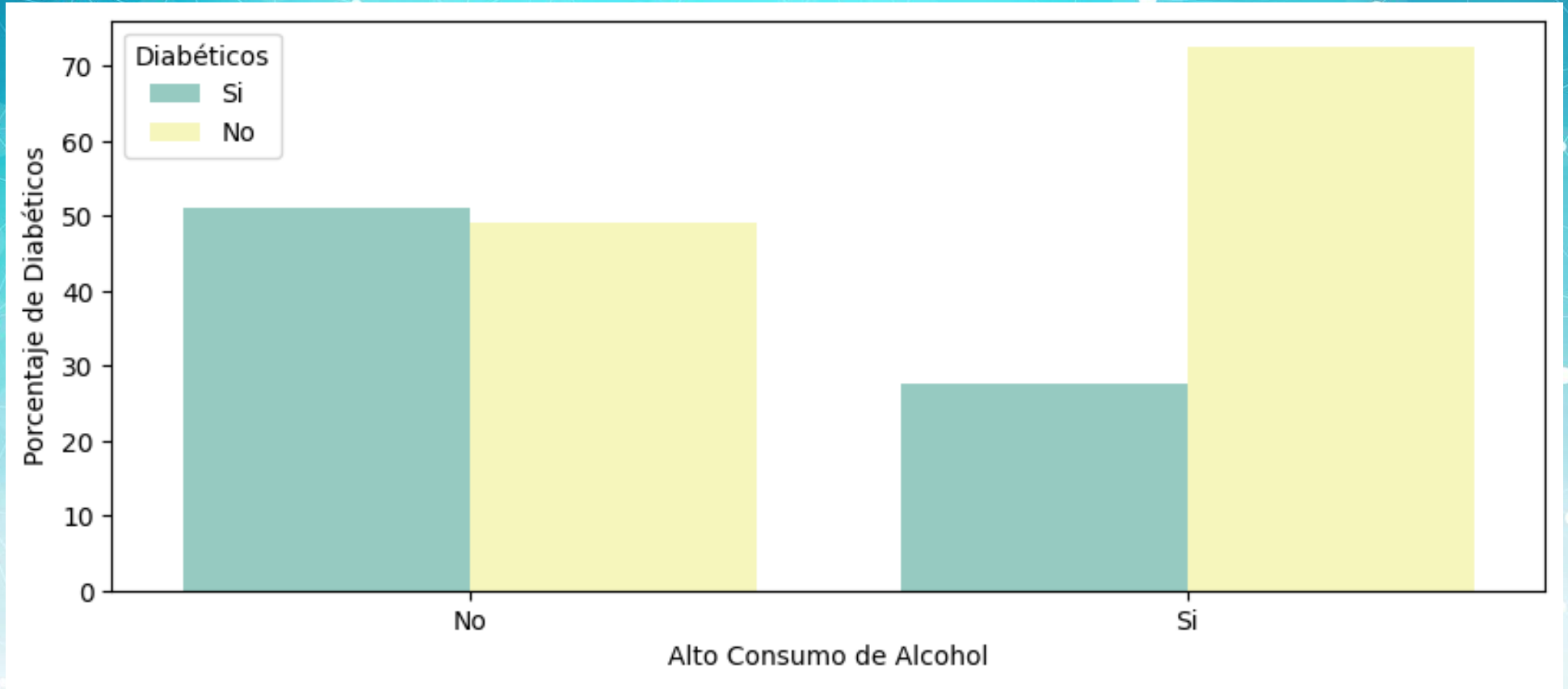
Con respecto a los rangos etarios que están mas afectados por la enfermedad de diabetes se encuentran: entre los 8 y 11 puntos dentro de nuestra categoría etaria, sobre todo entre los puntos 10 y 11. Aproximadamente entrarían las edades entre los 45 y los 70 años.



CODERHOUSE

Pude identificar que:

"Alto consumo de alcohol“
no contribuye necesariamente a que la persona tenga o no Diabetes

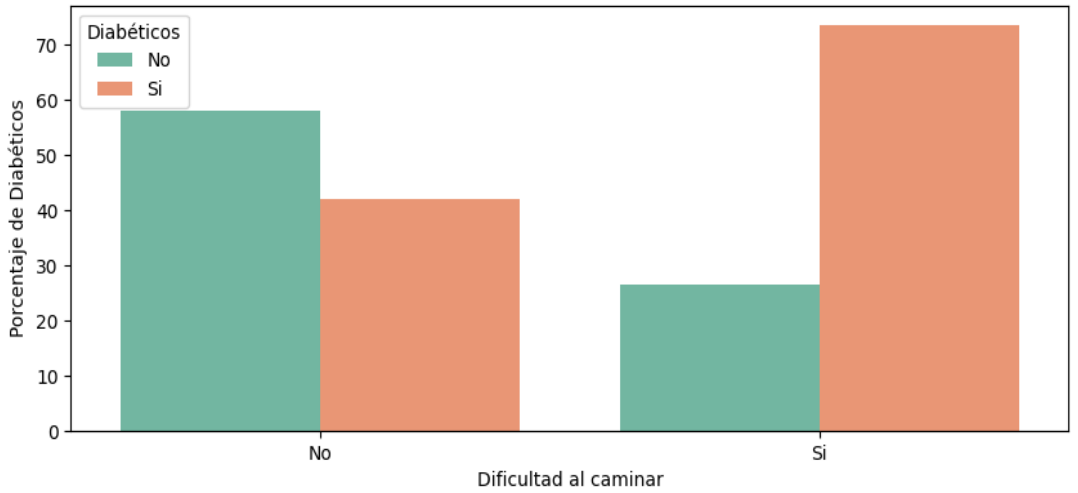


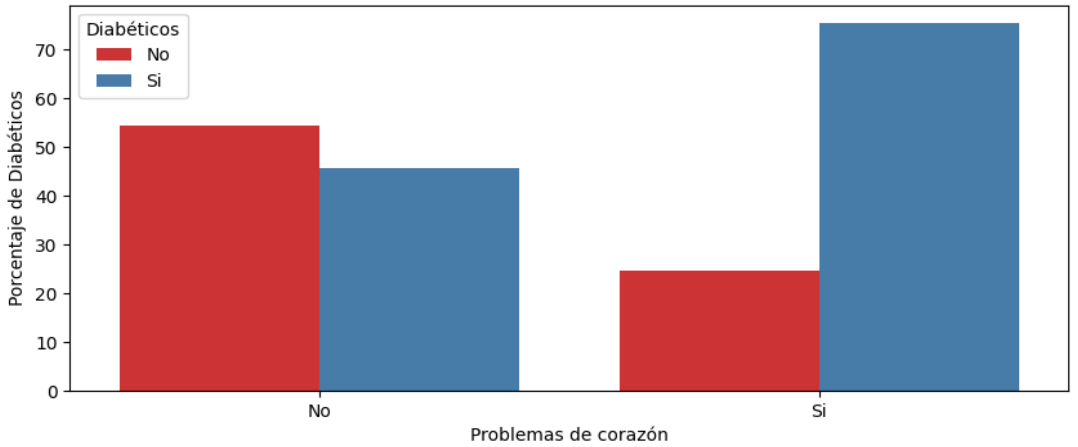
CODERHOUSE

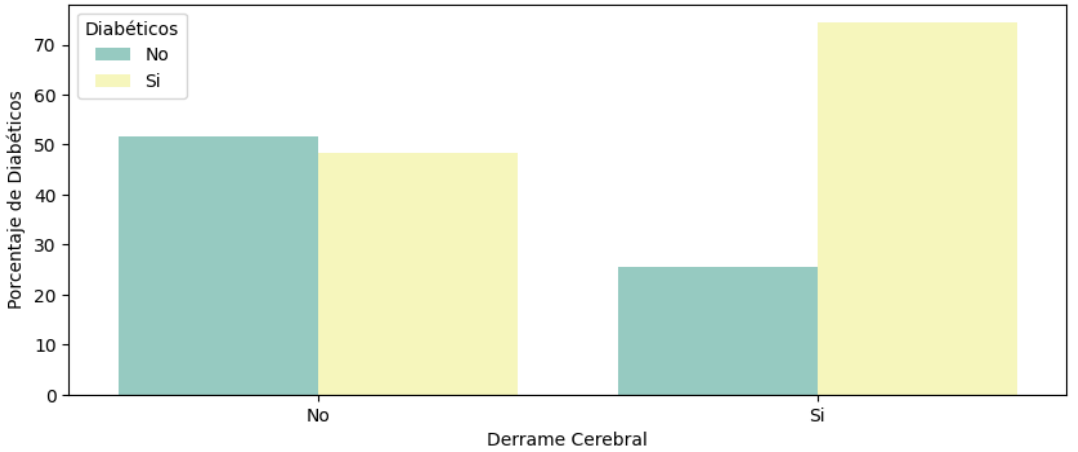
Pude notar que estas patologías de carácter peligroso:

"Derrame Cerebral"
"Dificultad para caminar“ y
“Problemas de Corazón”

si repercuten y contribuyen a la posibilidad de que la persona tenga la enfermedad de Diabetes.





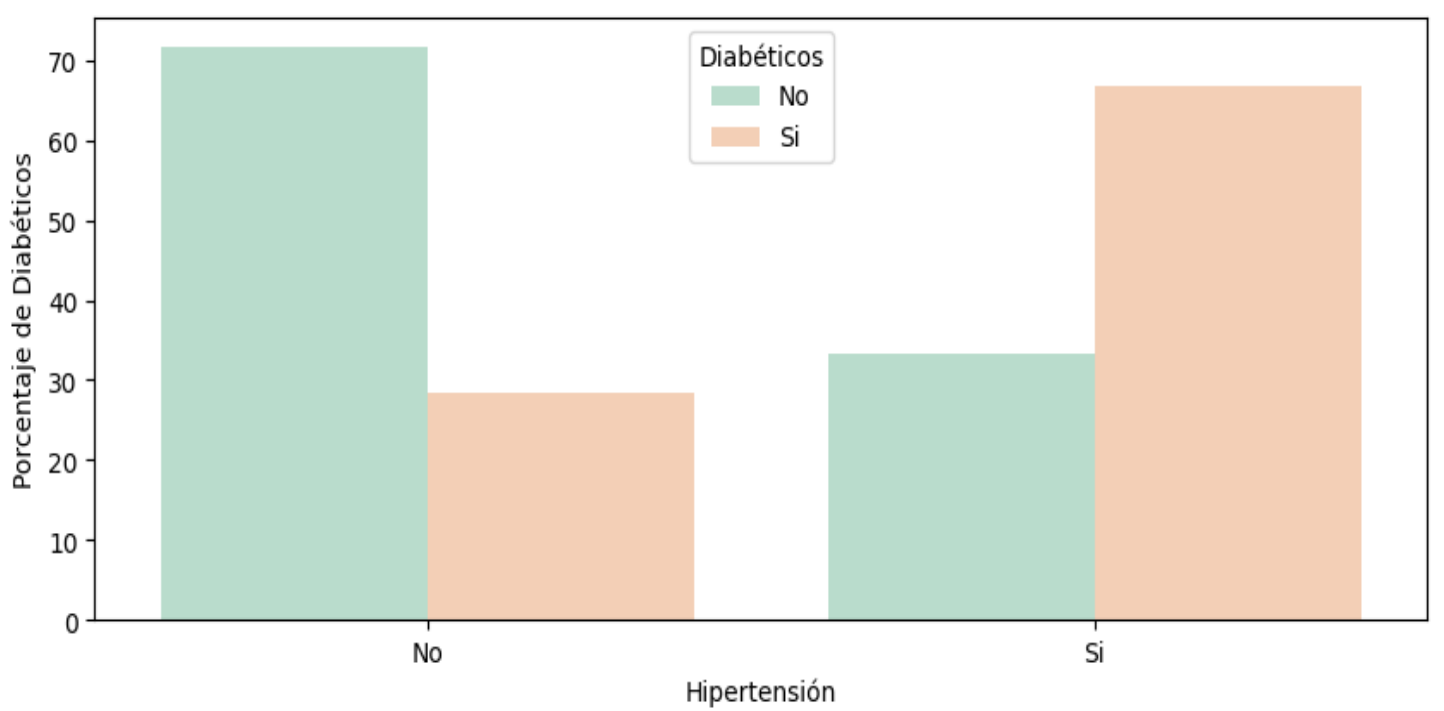


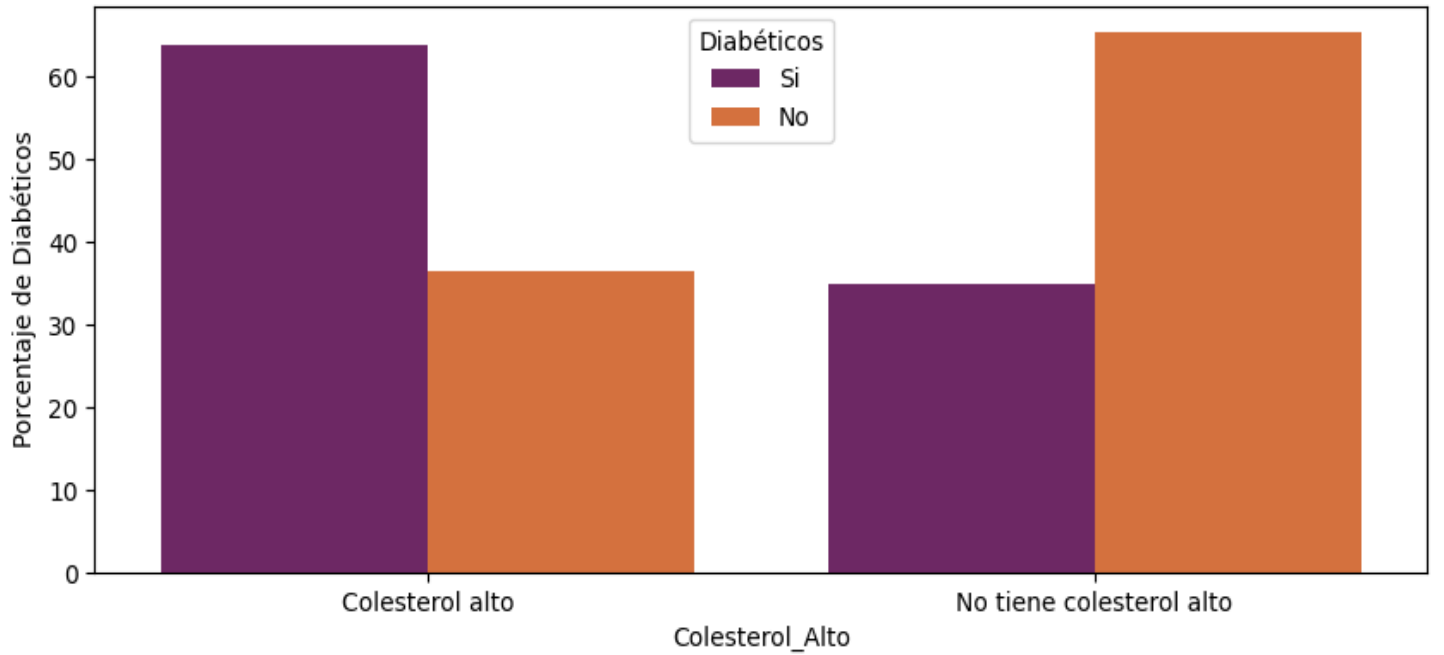
CODERHOUSE

También interfieren algunas patologías graves como:

"Colesterol Alto“ e ""Hipertensión"

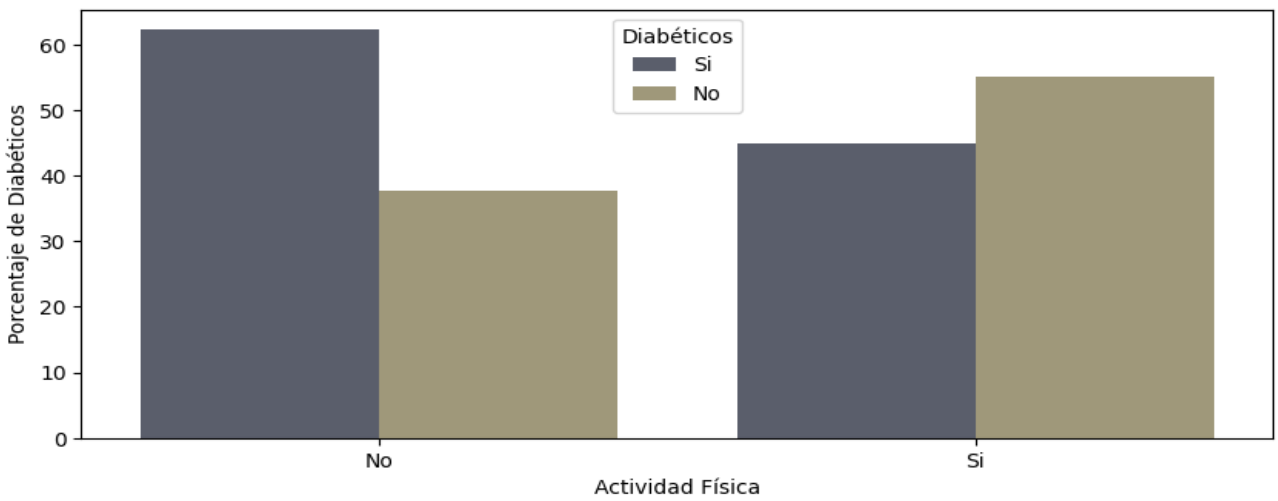
Que contribuyen a la posibilidad de que la persona tenga la enfermedad de Diabetes.

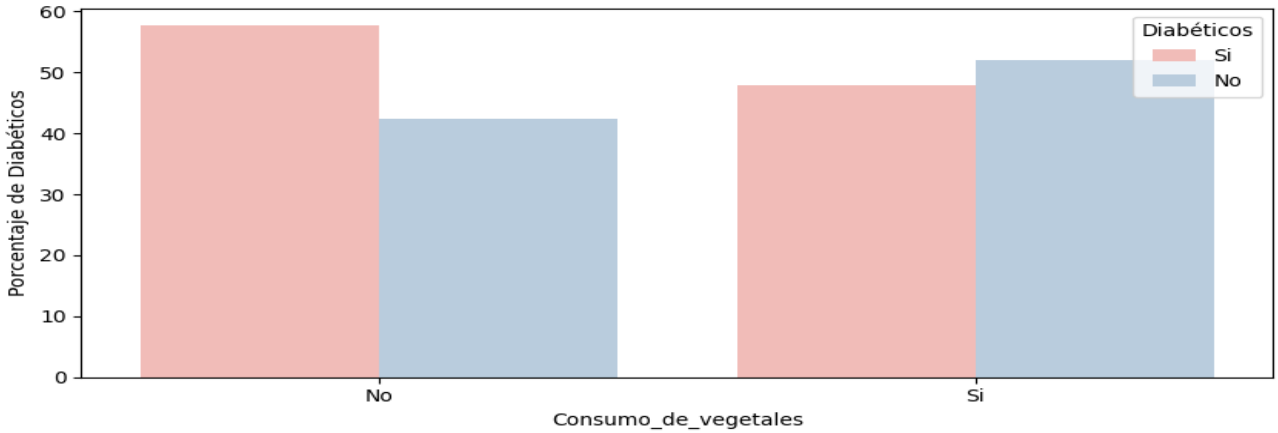


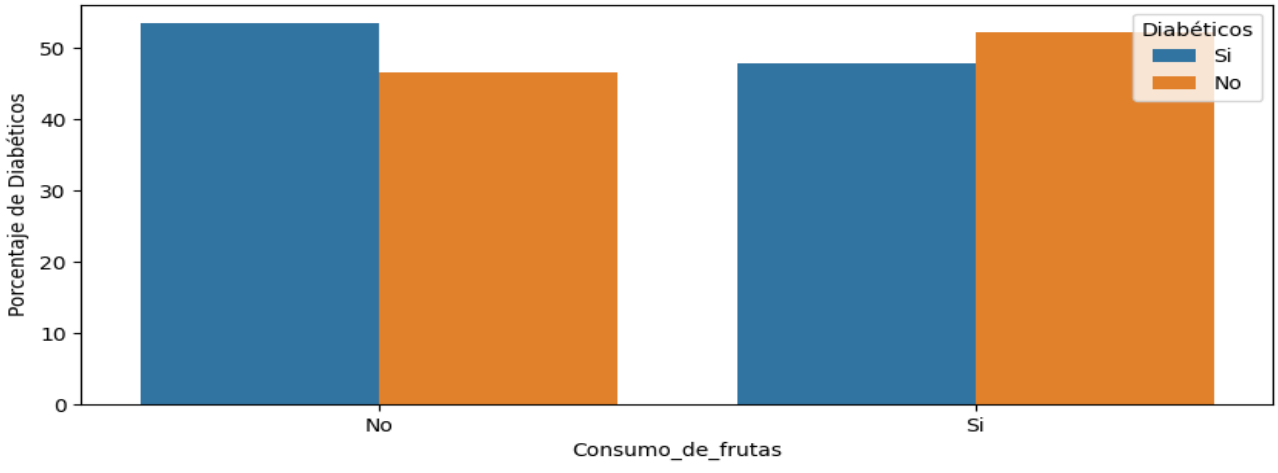


CODERHOUSE

Al ver los gráficos y todo nuestro análisis podemos deducir que la alimentación saludable **(frutas y verduras)** y la **actividad física** son factores que en nuestro dataset son indicador de alarma ya que interfieren notoriamente con la enfermedad.





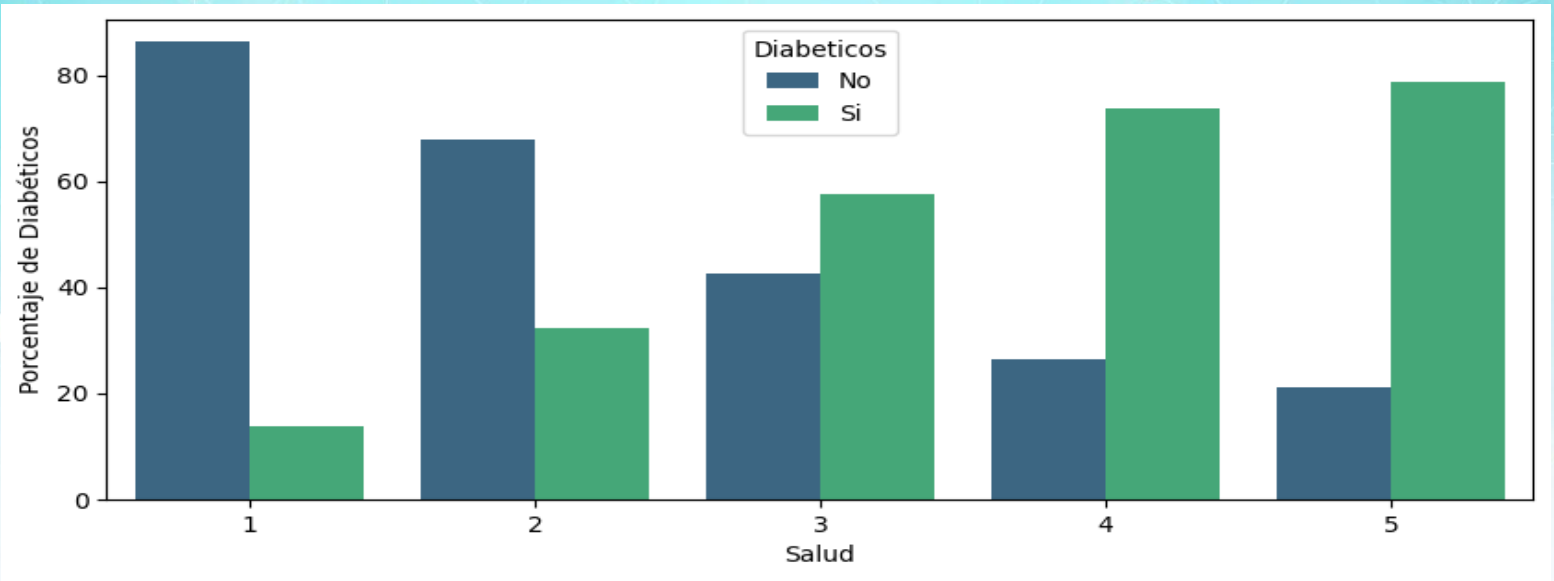


CODERHOUSE

Conclusión Final (Gráficos):

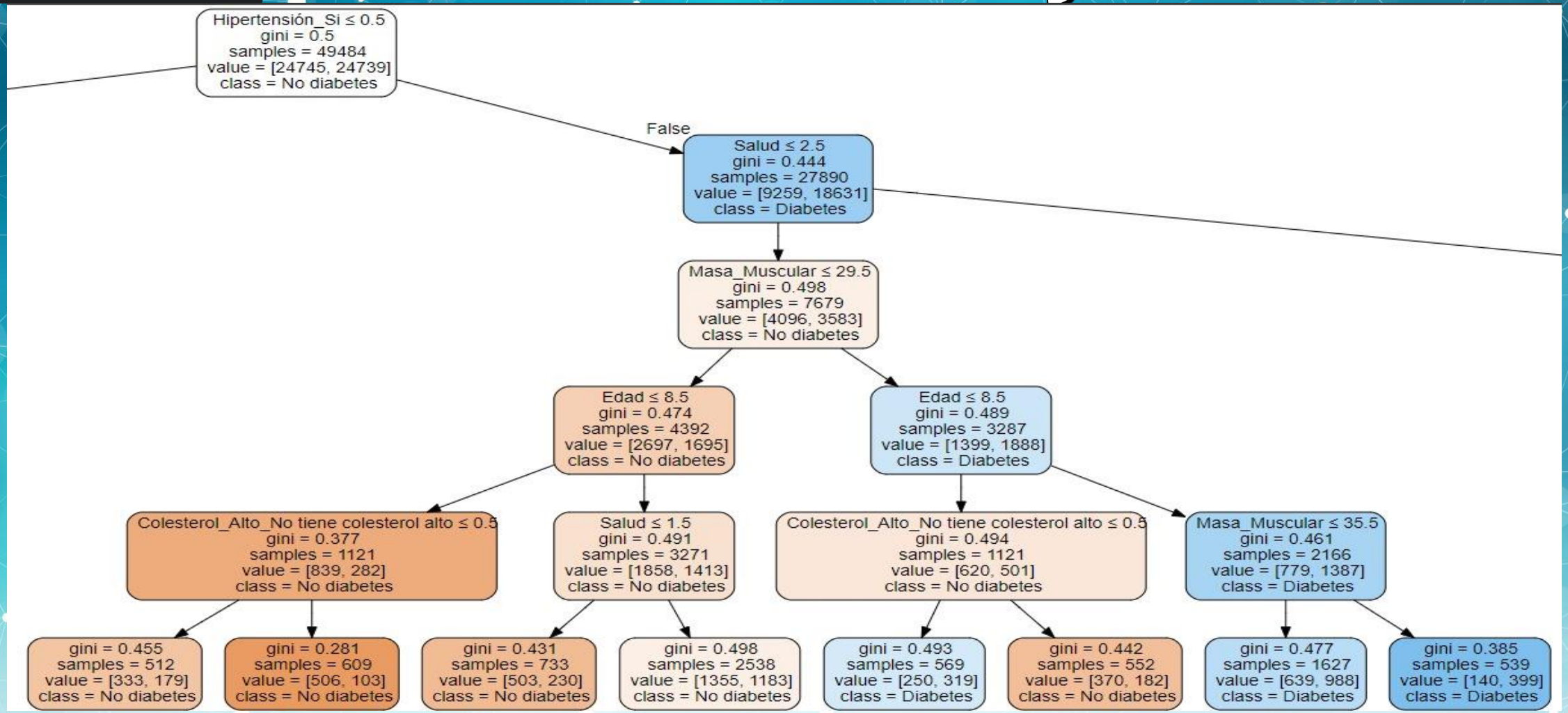
Podemos ver que según nuestro gráfico los pacientes que tienen la **enfermedad de diabetes** llevan una mala calidad de vida a comparación de aquellos pacientes que no poseen la enfermedad **(Siendo en salud: 1 el número mas bajo de malestar y 5 el mayor número).**

Es muy importante saber que hay que tener una buena alimentación saludable, realizar actividad física y no fumar para mantenernos más saludables y en caso de poseer alguna patología intentar tratarla lo antes posible para que no se agrave la situación del paciente.



CODERHOUSE

Machine Learning:



Este es un fragmento del gráfico que obtuve del “Árbol de decisión” que implemente en el trabajo.
(Podemos encontrar el grafico completo en el archivo titulado “Proyecto_Final_Florencia_Paz_Ponce.ipynb”)

CODERHOUSE

Machine Learning:

Podemos decir que gracias al árbol de decisión identificamos que la variable **hipertensión** es la más importante para el modelo porque es la primera que elige para tomar decisiones y abrirse en ramas para indicarnos como quedó conformado nuestro gráfico y como se ramifica el mismo de acuerdo a si el valor es verdadero o falso.

El **nodo raíz** del árbol es la variable "Hipertension_SI".

Segun "Hipertension_SI" el árbol se divide en dos ramas:

Una rama representa la condición **"salud = True"** y se etiqueta como **"NO DIABETES"**. O sea que es menos probable que tenga diabetes si el paciente tiene buena salud.

La otra rama representa la condición **"salud = False"** y se etiqueta como **"DIABETES".** Es mas probable que posea diabetes si el paciente no posee buena salud.

Esta división significa que si alguien tiene o no hipertensión, se sigue la rama correspondiente a su estado de salud y en base a ello se sigue subdividiendo con la misma lógica siguiendo las flechas y tomando en cuenta si el resultado de esa variable es **TRUE O FALSE Y SI POSEE DIABETES O NO.**

CODERHOUSE

Machine Learning:

En base al ejemplo del fragmento que podemos ver del gráfico:

Como la variable **"Salud" = falso** se subdivide en-:

Si **"Masa Muscular" es verdadero (Masa Muscular = True)**, la clasificación es "NO DIABETES". Significa es menos probable que el valor muscular alto de la persona este relacionado con diabetes

Si **"Masa Muscular" es falso (Masa Muscular = False),** la clasificación es "DIABETES". Significa es más probable que un valor muscular bajo este relacionado con la enfermedad de diabetes.

Luego podemos ver que el árbol sigue subdividiéndose y algunas de las métricas se repiten, en base a los resultados que vamos obteniendo según si da **TRUE O FALSE** es como debemos interpretar nuestro ML.

Podemos decir también que las variables más importantes de todo nuestro árbol fueron:

Hipertensión - Salud - Masa Muscular – Edad - Colesterol Alto

El árbol de decisión analizado clasifica a las personas en **"NO DIABETES" o "DIABETES"** basándose en la presencia de hipertensión, el estado de salud, el colesterol, la masa muscular y la edad.

CODERHOUSE

Métricas:

Podemos ver la precisión, exactitud, recall y F1- score para identificar con cual de ellos obtenemos una mayor precisión para nuestro ML.
(Árbol de decisión, Random Forest y XGBOOST)

Metricas (Arbol de Decisión)

```
[111] from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
```

```
[112] accuracy = accuracy_score(y_test, y_pred)
      precision = precision_score(y_test, y_pred, average='macro')
      recall = recall_score(y_test, y_pred, average='macro')
      f1 = f1_score(y_test, y_pred, average='macro')
```

```
print("Exactitud:", accuracy)
print("Precisión:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```

```
Exactitud: 0.737174651075066
Precisión: 0.7393280795672601
Recall: 0.7371612101215133
F1-score: 0.7365748339539835
```

CODERHOUSE

Métricas (Random Forest y XGBOOST):

Metricas RandomForest

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
f1 = f1_score(y_test, y_pred, average='macro')

print("Exactitud:", accuracy)
print("Precisión:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```

```
Exactitud: 0.7257939033877926
Precisión: 0.7270536796524608
Recall: 0.7258992712907691
F1-score: 0.7254722118436586
```

Metricas XGBOOST

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
f1 = f1_score(y_test, y_pred, average='macro')

print("Exactitud:", accuracy)
print("Precisión:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```

```
Exactitud: 0.7545088054317844
Precisión: 0.756626242248387
Recall: 0.7546386157265853
F1-score: 0.7540629649220296
```

CODERHOUSE

Conclusión Métricas:

Según podemos observar el modelo **más preciso** para nuestro análisis seria el de **XGBOOST** ya que es el que posee una mayor exactitud, precisión, recall y F1- score a comparación de los otros dos modelos.

Podemos ver que: El **árbol de decisión** tiene un acierto en el 73% de sus predicciones. Logra identificar dentro de los casos de diabetes al 73% de que realmente poseen la enfermedad.

El RandomForest tiene un acierto en el 72% de sus predicciones. Logra identificar dentro de los casos de diabetes al 72% de que realmente poseen la enfermedad.

Y por último el **Xgboost** tiene un acierto en el 75% de sus predicciones. Logra identificar dentro de los casos de diabetes al 75% de que realmente poseen la enfermedad.

CODERHOUSE

Cross Validation

Interpretación de los resultados para cada métrica:

Accuracy: Es la métrica más común para problemas de clasificación. Mide la proporción de predicciones correctas sobre el total de predicciones. En este caso, el resultado promedio de accuracy es aproximadamente 0.927, lo que significa que el modelo está acertando alrededor del 92.7% de las veces en las predicciones.

Precision_macro: Esta métrica mide el promedio de precisión para todas las clases en el problema de clasificación. La precisión se refiere a la proporción de predicciones positivas que son realmente correctas. El resultado promedio de precision_macro es aproximadamente 0.928, lo que indica que el modelo tiene una buena precisión en las predicciones.

Recall_macro: Mide el promedio de recall para todas las clases en el problema de clasificación. Recall se refiere a la proporción de ejemplos positivos que fueron correctamente identificados por el modelo. El resultado promedio de recall_macro es aproximadamente 0.925, lo que significa que el modelo tiene una buena capacidad para encontrar ejemplos positivos.

F1_macro: Esta métrica es el promedio armónico de precisión y recall (ponderando igualmente ambas métricas). El resultado promedio de f1_macro es aproximadamente 0.926, lo que indica que el modelo tiene un buen equilibrio entre precisión y recall.

En general, los resultados obtenidos indican que el modelo tiene un rendimiento sólido en términos de precisión, recall, F1-score y accuracy.

CODERHOUSE

Evaluación: Grilla de Hiperparámetros

Mejores hiperparámetros encontrados: Según los resultados obtenidos, los mejores hiperparámetros para el modelo de clasificación son {'C': 10, 'kernel': 'rbf'}. Esto significa que el valor óptimo para el hiperparámetro 'C' es 10, y el tipo óptimo de función kernel es 'rbf'.

Mejor resultado de precisión: El mejor resultado de precisión obtenido es aproximadamente 0.947. Esta precisión representa la proporción de predicciones correctas realizadas por el modelo en datos nuevos o en el conjunto de prueba.

En resumen, el modelo con hiperparámetros 'C'=10 y 'kernel'='rbf' logra una precisión promedio de 94.7%. Esto significa que el modelo está acertando en sus predicciones alrededor del 94.7% de las veces en el conjunto de prueba utilizado en la validación cruzada.