Los archivos de video, la presentación para SaludAlpes y el dashboard de PowerBI se encuentran en la carpeta llamada docs.

El archivo de preparación de datos se llama PrepDatos.ipynb

El archivo de los algoritmos KNN, Random Forest y Decision Tree son respectivamente KNN.ipynb, RandomForest.ipynb, dtc.ipynb

El archivo en el que probamos el mejor modelo está en PruebaMejorModelo.ipynb

Las carpetas KNN, RF, Tree contienen los mejores modelos serializados de cada uno de los algoritmos.

En data encuentra todos los archivos csv.

**Descripción y análisis del entendimiento de los datos**

En el csv encontramos 18 columnas entre las cuales encontramos:

HeartDisease: Puede ser Yes o No dependiendo si la persona reporta enfermedad coronaria.

BMI: Indice de masa corporal, debe estar entre 1 y 99.

Smoking: Respuestas a la pregunta ¿Ha fumado al menos 100 cigarillos en toda su vida? Sí o No.

AlcoholDrinking: Hombres adultos que tienen más de 14 bebidas a la semana y mujeres adultas que han tenido más de 7 bebidas a la semana (Sí/No).

Stroke: Respuestas a la pregunta ¿Alguna vez ha tenido un derrame cerebral? (Sí/No)

PhysicalHealth: Respuestas a la pregunta ¿Por cuantos días en los últimos 30 días se ha sentido enfermo? Debe estar entre 0 y 30.

MentalHealth: Respuestas a la pregunta ¿Por cuantos días en los últimos 30 días se ha sentido mal mentalmente? Debe estar entre 0 y 30.

DiffWalking: Respuestas a la pregunta ¿Ha tenido dificultades al caminar o al subir escaleras? (Sí/No).

Sex: Respuestas a la pregunta ¿Es hombre o mujer? (Mujer/hombre).

AgeCategory: Categoría de edad, mínimo su edad debe ser mayor a 50.

Race: Raza a la que pertenece (White/Hispanic/Black/Asian/Other).

Diabetic: Respuestas a la pregunta ¿Tiene diabetes? (Sí/No).

PhysicalActivity: Adultos que reportaron hacer actividad física o ejercicio durante los últimos 30 días más que en su trabajo regular. (Sí/No).

GenHealth: Respuestas a la pregunta ¿Como considar su salud general? (Very good/Good/Excellent/Fair/Poor).

SleepTime: En promedio cuantas horas durmió en las últimas 24 horas. Debe estar entre 0 y 24.

Asthma: Yes o No dependiendo de si la persona reporta asma.

KidneyDisease: Yes o No dependiendo de si la persona reporta enfermedades del riñón (No incluye cálculos en el riñón, infección de vejiga e incontinencia).

SkinCancer: Yes o No dependiendo si la persona reporta cáncer de piel.

**Descripción del preprocesamiento de los datos**

Primero se analizó la cantidad de nulos presente en cada columna, con el fin de decidir que columnas necesitaban completar datos y si se debían borrar filas. En este paso se decidió borrar las filas nulas de la columna HeartDisease ya que era la variable objetivo y si intentábamos completar podríamos traer complicaciones al modelo y generar gran cantidad de falsos positivos y negativos.

Para poder utilizar los datos en nuestros algoritmos, cambiamos todas las columnas a variables numéricos. Adjunto se puede encontrar el archivo PrepDatos.ipynb en donde se hizo el preparamiento.

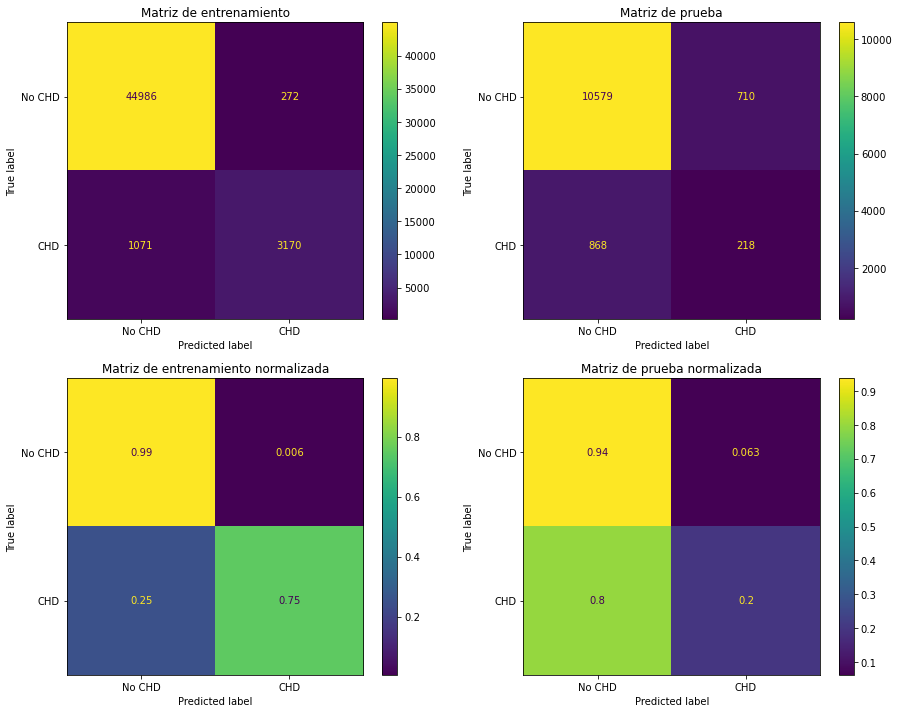
* **Cambios en SleepTime:** Existen ciertos datos que no están dentro del rango aceptado >0 y <=24, sacamos un promedio de los datos que están correctos y reemplazamos los datos que están afuera del rango con el promedio calculado. Además, la columna está en un tipo diferente al esperado.
* **Cambios en AgeCategory:** Existen algunos datos que están dentro de un rango inferior a 50, estos datos serán cambiados por la moda de los rangos que si pertenecen a la definición.
* **Cambios en BMI:** Existían datos incoherentes que tenían caracteres alfanuméricos, los cuales fueron cambiados a números, además también encontramos nulos y números fuera del rango, se hizo un promedio con los datos que eran correctos en principio y se utilizó este promedio para completar nulos y datos incoherentes.
* **Cambios en GenHealth:** Existían nulos y datos que no correspondían a las opciones dadas para esta columna, se limpiaron las palabras que tenían alguna opción dada y caracteres adicionales, además se completaron los nulos y los datos incoherentes con la moda.
* **Cambios en Sex:** Para esta columna se asigna 0 a las mujeres y 1 a los hombres. Además, los datos indefinidos se asignan como indefinido y se le pone el numero 2.
* **Cambios en Physical/ Mental health :** Como estas columnas constan de numeros decimales con caracteres basura y se deben interpretar como enteros en un rango de 0 a 30 se eliminan los caracteres basura y se asigna el numero entero correspondiente, tomando los valores negativos como 0 y los mayores a 30 como 30.
* **Cambios en Diabetic**: Para preparar esta columna se eliminan los caracteres basura, se asigna los valores vacios a sí y se asignan valores numericos de la forma:
* 0: No
* 1: Yes
* 2: No, borderline diabetes
* 3: Yes, (during pregnancy)
* **Corrección Race: Para esta columna se hace un proceso similar a la anterior, se eliminan los caracteres basura, pero los valores indefinidos se agregan a la categoría otros y se asignan los valores numéricos de la forma:**
* 0: White
* 1: Hispanic
* 2: Black
* 3: Asian
* 4: American Indian/Alaskan Native
* 5: Other

**Implementación de algoritmos**

* **Arboles de decisión – Juliana Andrea Galeano Caicedo:** Arboles de decisión es un algoritmo de aprendizaje supervisado utilizados para tareas de clasificación y regresión lineal. Este algoritmo crea una estructura de árbol donde cada nodo representa una característica, la rama representa una regla de decisión y cada nodo hoja es el resultado. En el algoritmo utilizamos todas variables ya procesadas presentes en el csv, utilizando como variable objetivo HeartDisease, se realizó un Grid para simular diferentes combinaciones de hiperparametros y con esto obtener las que mejores resultados arrojaba. Se obtuvo como mejor resultado la combinación de hiperparametros: 'min\_samples\_split': 20, 'min\_samples\_leaf': 2, 'max\_leaf\_nodes': 100, 'max\_features': 'sqrt', 'max\_depth': 12, 'criterion': 'gini'.
* **K-NN – María Camila Gómez Hernández:** K-nearest neighbor es un algoritmo de aprendizaje supervisado que puede ser usado para problemas de regresión y de clasificación. KNN funciona en el principio de asumir que cualquier dato cercano a otro cae en la misma clase, clasifica un nuevo dato basado en la similitud. Al estandarizar los datos vimos que nuestro modelo se comportaba mucho más optimo que al no estandarizarlos, utilizamos GridSearchCV para encontrar nuestro mejor modelo haciendo que el K sea variable entre el 0 y 10, debido a que evidenciamos que después de k = 10 el modelo empeoraba notablemente, hicimos que oscilara entre un p = 1 y p = 2 junto con los pesos del algoritmo, es decir entre ‘uniform’ y ‘distance’.
* **Algoritmo de libre elección (Random Forest) – William Felipe Mendez Ardila:** El algoritmo Random Forest es un algoritmo de machine learning supervisado para aprendizaje supervisado que se basa en realizar el algoritmo de árboles de decisión múltiples veces y unir sus resultados por medio de validación cruzada. Durante la implementación de este algoritmo se realizó un modelo de referencia con hiperparametros definidos y luego se ajustaron estos por medio de la clase RandomizedSearchCV de sklearn, la cual probó con varias combinaciones aleatorias de datos múltiples veces y llegó a que la mejor combinación de hiperparametros es: 'n\_estimators': 5, 'min\_samples\_split': 9, 'min\_samples\_leaf': 1, 'max\_features': 'sqrt', 'max\_depth': 100, 'criterion': 'gini'.

**Análisis de resultados obtenidos y justificación**

* **Arboles de decisión:** En el notebook dtc.ipynb puede encontrar la ejecución completa del algoritmo junto con su paso a paso. La configuración más optima luego de realizar un Grid y comprobar diferentes combinaciones de hiperparametros fue: 'min\_samples\_split': 20, 'min\_samples\_leaf': 2, 'max\_leaf\_nodes': 100, 'max\_features': 'sqrt', 'max\_depth': 12, 'criterion': 'gini'.





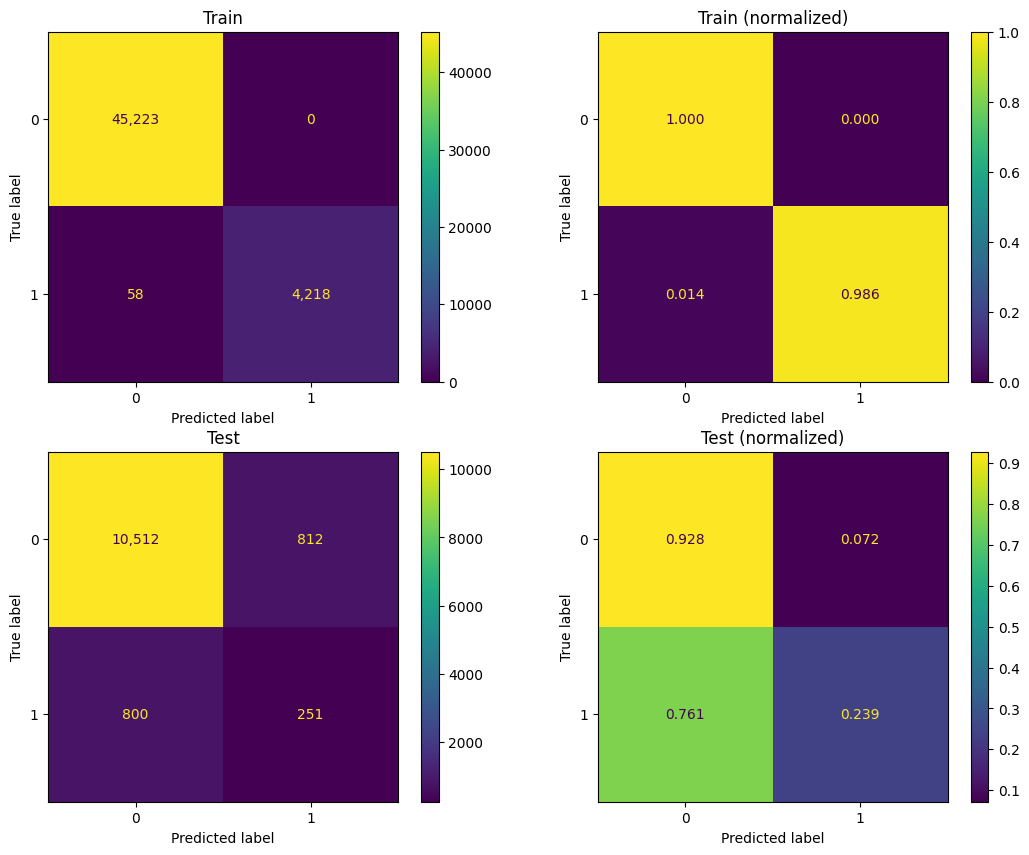
* **KNN:** Después de haber estandarizado los datos, encontramos que este modelo tiene uno de los f1 más altos de los modelos probados con otros algoritmos. A continuación, estarán las matrices de confusión de las pruebas de los datos sin estandarizar en nuestro primer modelo junto con las métricas de testeo y la matriz de confusión de nuestro mejor modelo junto con las métricas de testeo. En el notebook KNN.ipynb puede encontrar la ejecución completa del algoritmo junto con su paso a paso hasta la serialización. La configuración más optima es {'n\_neighbors': 2, 'p': 1, 'weights': 'distance'}

**Gráfico, Gráfico de rectángulos

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

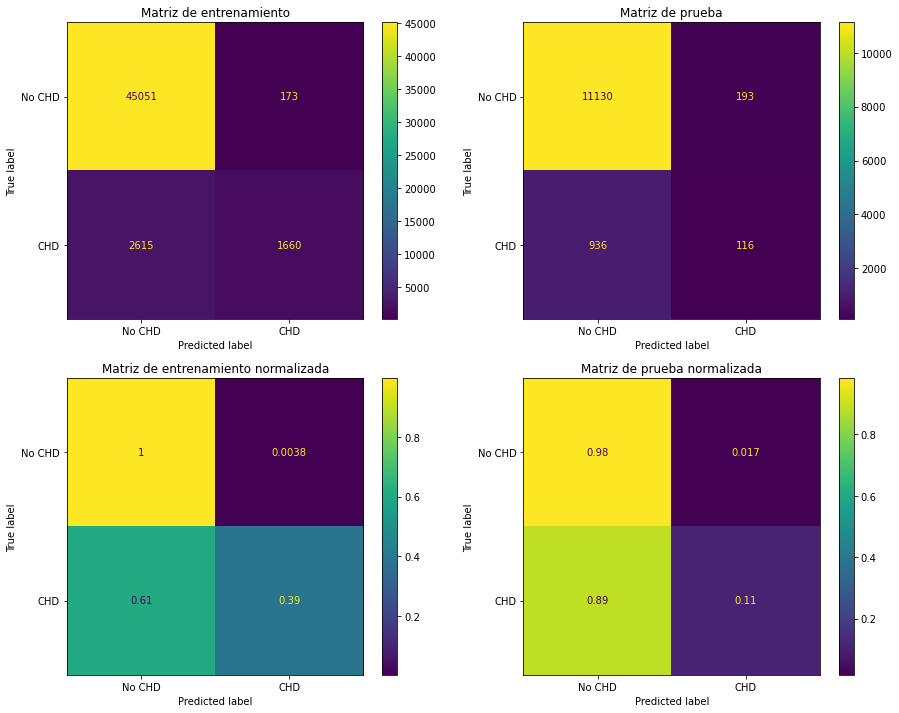


Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

Debido a esto decidimos utilizar este algoritmo pues nos dio un mejor f1 lo cual demuestra que es más preciso.

* **Random Forest**

Después de realizar la implementación del modelo con el notebook RandomForest.ipynt se llegó a los siguientes resultados para la matriz de confusión y las métricas de resultados.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Pantalla azul con letras blancas

Descripción generada automáticamente con confianza media