Andrés Peña Diaz (201913766)

Isabela Galindo (202012613)

Juan José Córdoba (201922105)

Limpieza de datos:

Para cada columna de los datos hicimos una limpieza exhaustiva que reemplaza o elimina los valores nulos y los valores que no tienen sentido.

En la columna “BMI” empezamos reemplazando los valores nulos por 0. Luego limpiamos eliminamos las letras de los valores numéricos para que tengan sentido. En seguida convertimos todos los valores en flotantes y por último los valores que tienen un valor de 0 los cambiamos por el promedio de la columna.

Para las columnas que pueden tomar valores booleanos, como lo son “Smoking”, “AlcoholDrinking”, “Stroke”, “DiffWalking”, “PhysicalActivity”, “Asthma”, “KidneyDisease”, “SkinCancer”; primero reemplazamos los valores nulos por un “No”, ya que consideramos que al asumir que no han tenido estas enfermedades afectaría menos a nuestro modelo. Luego revisamos los valores y si en el valor contenían en algún lugar de la cadena la palabra “Yes”, reemplazamos el valor por “True”. En seguida hicimos lo mismo con la palabra “No”, pero lo reemplazamos con el booleano “False”.

Para “PhysicalHealth” y “MentalHealth” primero eliminamos los posibles caracteres que estén en la cadena. Luego convertimos los valores en enteros. Por último, revisamos si los valores eran mayores a 30, y si lo eran, son reemplazados por la mediana de la columna.

Para la columna “Sex” primero revisamos si la palabra “Female” estaba contenida en el valor, si lo estaba el valor era reemplazado por esta palabra. Lo mismo hicimos con la palabra “Male”. Para los valores que no tenían sentido los reemplazamos por nulos. Para finalizar, todas las filas que tienen valores nulos en esta columna eran eliminadas.

Para la columna “AgeCategory” convertimos los valores que tuvieran “or older 80” a “80 or older”. Luego eliminamos todas las filas que tuvieran rangos que sean inferiores a 50 y las demás eran eliminadas.

Para la columna “Race” primero reemplazamos los valores nulos con la palabra “Other”. Luego si en el valor estaba contenida la palabra “White”, “Hispanic”, “Black” o “Asian” eran reemplazados por estas mismas palabras respectivamente.

Para la columna “Diabetic” primero reemplazamos los valores nulos con la palabra “No”. Luego si en el valor estaba contenida la palabra “Yes (during pregnancy)”, “No, borderline diabetes”, “No” o “Yes” eran reemplazados por estas mismas palabras respectivamente.

Para la columna “GenHealth” primero reemplazamos los valores nulos con la palabra “Fair”. Luego si en el valor estaba contenida la palabra “Good”, “Very good”, “Excellent” , “Fair” o “Poor” eran reemplazados por estas mismas palabras respectivamente.

Para la columna “SleepTime” primero reemplazamos los valores nulos por el valor 0. Luego convertimos los valores en flotantes. En seguida, los valores que aparecen como negativos los

convertimos en positivos y si los valores tienen un valor mayor a 24 los reemplazamos por el promedio.

Para la columna “HeartDisease” reemplazamos los valores nulos por la palabra “Yes”, esto debido a que consideramos que es mejor tratar a una persona que no esté enfermos del corazón a no tratar a una persona que si está enferma del corazón. Luego si en el valor estaba contenida la palabra “No” o “Yes” eran reemplazados por estas mismas palabras respectivamente. Por último, si el valor no tenía sentido también era reemplazado por la palabra “Yes”.

Preparación de los datos para el modelo KNN y árbol de decisión

Tanto para el modelo de K-Nearest Neigbors y el árbol de decisión, se prepararon los datos de la misma manera. Antes de empezar a entrenar los modelos se tuvo que convertir las columnas categóricas en columnas numéricas. En la columna ***“Sex”*** si el valor es masculino se le asigna 1, si es femenino se le asigna un 0. En la columna ***“AgeCategory”*** se le asigna a cada categoría un número del 0 al 6 para poderlas diferenciar. En la columna ***“Race”*** se le asigna a cada raza especificada un número del 0 al 4. En la columna ***“GenHealth”*** se le asigna a cada categoría un número del 0 al 4. Por último, a la columna ***“Diabetic”*** se le asigna un número del 0 al 3 a cada categoría.

Esto se hace para que los algoritmos puedan ejecutarse de manera correcta y que no haya ningún error.

Para el árbol de decisión la razón por la cual el “random_state” se eligió poner el 1626 es debido a que se hicieron muchas pruebas y el que mejor F1 daba era este.

Preparación de los datos para el modelo Gaussian Naive Bayes

Para el modelo de Gaussian Naive Bayes se hizo una preparación extra de los datos que consistía en desplegar las variables categóricas de {“Sex”, “AgeCategory”, “Race”, “GenHealth”, “Diabetic”} en forma de columna utilizando la función de get_dummies que ofrece pandas sobre un dataframe.

Lo anterior, debido a que el modelo de Gaussian Naive Bayes requiere que las variables categóricas estén en ese formato para poder hacer el entrenamiento del mismo.

Adicionalmente, el modelo de GaussianNB no tiene unos hiperparametros los cuales haya podido establecer para obtener una mejora en los resultados del F1

Responsabilidades de algoritmos:

KNN Algorithm: Laura Isabela Martínez Galindo (202012613)

Arboles de decisión: Juan José Córdoba Vela (201922105)

Gaussian Naive Baye: Andrés Peña Diaz (201913766)