Machine Learning para predicción de enfermedades cardiovasculares

Castillo Almazán Johan Axel and Merida Santana Jesus

Licenciatura de Fisíca Biomedica, Facultad de Ciencias, UNAM.

Algoritmos Computacionales

6 de Junio del 2023

Abstract

En este trabajo se desarrollo un algoritmo de machine learning, con ayuda de la libreria "pandas", el cual es capaz de predecir enfermedades cardiacas a partir de una base de datos, que para este caso en particular contenia información sobre la población de una región especifica de la CDMX.

Motivación

La principal motivación para reaalizar este proyecto es que México es un de los paises con mayor indice de personas que sufren enfermedades cardiovasculares en Latino America, solo por debajo de Brasil y Argentina respectivamente. En México, el 19 % de mujeres y hombres de 30 a 69 años muere de enfermedades cardiovasculares, y se estima que el 70.3% de la población adulta vive con al menos un factor de riesgo cardiovascular, esto por una falta de control de los mismos, como lo son el tabaquismo, la presión arterial alta, el colesterol elevado, etc. Es por eso que con la ayuda de un algoritmo de machine learning, se podria ayudar a la población a saber que tan probable es que sufran de una enfermedad cardiovascular, y de esta manera comenzar un tratamiento para prevenir consecuencias fatales como la muerte.

Introduction

Antes de poder empezar a desarrollar nuestro codigo, se requiere del análisis de datos y por ello es necesario hacer uso de algunas herramientas de python. Estas herramientas son las siguientes librerias

- Libreria NUMPY

- Libreria PANDAS

NUMPY es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos, hace uso de elementos llamados arrays que permiten representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación, estados elementos son procesados de manera rápida permitiendo usar vectores o matrices de grandes dimensiones

PANDAS es una librería de Python especializada en el manejo y análisis de estructuras de datos, esta permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos SQL así como acceder a los datos mediante índices o nombres para filas y columnas, además de que cuenta con métodos que ayudan a ordenar, dividir y combinar datos.

Algoritmo K-NN

Es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Para el programa a realizar este algoritmo se utilizará para una clasificación. Se requiere de una base de datos que contenga valores discretos, en este caso se van a etiquetar dos grupos, uno de ellos será de personas enfermas y otro de personas no enfermas, posteriormente al ingresar nuevos datos el algoritmo calcula la distancia métrica entre el punto generado y los ya establecidos, finalmente asignará el que tenga una menor distancia generando una predicción de si la persona está enferma o no.

¿Cómo se calcula esta distancia?

Para determinar qué puntos de datos están más cerca de un punto de consulta determinado, será necesario calcular la distancia entre el punto de consulta y los otros puntos de datos. Distancia euclidiana (p=2): Mide una línea recta entre el punto de consulta y el otro punto que se mide usando la siguiente fórmula:

$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$

Regresion logistica

La regresión logística es un modelo estadístico que se utiliza para determinar la probabilidad de que ocurra un evento. Muestra la relación entre características y luego calcula la probabilidad de un resultado determinado. La regresión logística se utiliza en Machine Learning (ML) para ayudar a crear predicciones precisas. Es similar a la regresión lineal, excepto que en lugar de un resultado gráfico, la variable objetivo es binaria; el valor es 1 o 0. Existen dos tipos de variables medibles, las variables o características explicativas (elemento que se mide) y la variable de respuesta o variable binaria objetivo, que corresponde al resultado.

Existen tres tipos básicos de regresión logística:

1-Regresión logística binaria: aquí solo existen dos resultados posibles para la respuesta categórica.

2-Regresión logística multinomial: aquí es donde las variables de respuesta pueden incluir tres o más variables, que no estarán en ningún orden.

3-Regresión logística ordinal: al igual que la regresión multinomial, puede haber tres o más variables. Sin embargo, existe un orden en el que siguen las medidas.

Para este proyecto basta con comprender la regresion logistica binaria, sin embargo es importante mencionar que si bien la complejidad es diferentes para cada uno de los tipos de regresion, en ecia usan el mismo pricipio. Se hace uso de una funcion que permite transformar variables independientes de caracter continuo, en una clasificacion para la regresion logistica binaria obtendremos una clasificacion alta (1) y una baja(0). La funcion que permite hacer esto se conoce como sigmoide y esta dada por la siguiente formula:

$P(Z) = \frac{1}{1+e^{-z}}$

Donde z representa la combinacion lineal entre los coeficientes de regresion logistica y las variables independientes mas un error.

Coeficiente de correlacion de punto biserial

El coeficiente de correlación de punto biserial es una medida de asociación entre una variable continua y una variable binaria (dicotómica). Se calcula utilizando la siguiente fórmula:

$rpbis = \frac{M_1 - M_0}{Sn}\sqrt{\frac{n_1 n_0}{n^2}}$

-M1 = Media del puntaje global del instrumento del grupo que contestó de manera positiva a la variable binaria (alta, uno).

-M0 = Media del puntaje global del instrumento del grupo que contestó de manera negativa a la variable binaria (baja, cero).

-Sn = Desviación estándar del instrumento.

-n = Tamaño de la población que contestó el instrumento.

-n1 = Tamaño del grupo que contestó de manera positiva a la variable binaria.

-n0 = Tamaño del grupo que contestó de manera negativa a la variable binaria.

Desarrollo

Se programo un algoritmo de clasificacion que tratara de replicar lo hecho por el algoritmo KNN de paqueterias como numpy, para poder hacerlo primero fue necesario programar una funcion que nos calulara la distancia euclidiana entre dos puntos, esta funcion se creo siguiendo la formula que ya se menciono en la introduccion y que permite usarse en n dimensiones, con esta funcion definida se procedio a realizar una funcion a la cual se llamo como Algoritmo KNN, esta reuqiere de una base de datos que contiene variables independientes, ademas se le proporcionan las nuevas variables a considerar para la clasificacion, es importante mencionar que estos datos deberan ser del mismo caracter que los de la base de datos, es decir que si por ejemplo en la base de datos tenemos una variabe que indica el peso entonces para los datos de la prediccion tambien se tendra que proporcionar dicho valor, finalmente se le da el numero de puntos que determinaran la etiqueta de clasificacion. El algoritmo funciona calculando la distancia que hay entre la variable de nuestra base de datos con

la que le proporcionamos, esto lo hace para todas las variables una vez calculadas, usando el numero de puntos proporcionado se determinan las distancias mas cercanas, revisando la etiqueta que tiene dicho valor fianlmente se cuenta que etiqueta se repite mas y se asiga la clasificacion en funcion de esta. Se probo el funcionamiento de esta funcion para verificar que la clasifiacion fuera correcta, para ello se propuso una base de datos que contenia estatura, peso y la nacionalidad asociada a estos segun el banco mundial de estatura promedio, se le proporcionoaron datos de forma aleatoria para estas dos variable asi como el numero de puntos para la clasificacion, finalmente se corrio la funcion y se verifico que en funcion de la base datos el algoritmo otorgara la clasificacion , en este caso una nacionalidad.

Continuando con el proyecto y debido a que se busca una prediccion con una mayor efectividad se hizo uso de la regresion logistica, para poder programar este algoritmo fue necesario definir algunas funciones primero, en principio se definio una variable a la cual se le dio una aproximacion al numero de euler, esta permitio segun la formula para la regresion logistica programar la funcion sigmoide, una vez programada esta funcion se procedio a realizar el algoritmo completo de regresion logistica, se hizo en forma de una funcion a la cual se llamo como "Regresion logistica", a esta se le proporcionan coeficientes de gresion, variables y sesgo (error), en la funcion lo primero que se hizo fue calcular la combinacion lineal entre los coeficientes de regresion logistica y las variables proporcionadas, para ello se asume que el sesgo ya esta incluido en los coeficientes, esto se asumio debido a la complejidad que tiene calcular este sesgo para grandes bases de datos, una vez calculada esta combancion, se le aplica la funcion sigmoide y se fija un umbral para determinar las etiquetas de clasificacion, el umbral fue el siguiente:

-Si valor obtenido por la funcion sigmoide es mayor que 0.5 se toma como 1 y se asocia una etiqueta 1.

-Si valor obtenido por la funcion sigmoide es menor que 0.5 se toma como 0 y se asocia una etiqueta 2.

Para probar este funcion se propusieron valores al azar para coeficientes de regresion asi como variables que fueron edad, sexo y presion arterial, finlamnte al aplicar la funcion esta realizo la clasificacion de forma correcta.

Finalmente y una vez probados los algorimos para esos metodos de prediccion se prosiguio a trabjar con una base real de datos medicos, enfocados en variables independientes relacionados con enfermedades cardiacas, en esta base se contemplan aproximadamente 500 datos de personas a las cuales se les registraron las siguientes variables:

-Presion arterial.

-Concentracion de tabaco en el organismo.

-Concentracion de lipoproteína de muy baja densidad.

-Cantidad de adipositos

-Indice de masa corporal

-Edad

Es importante mencionar que las variables de esta base de datos estan relacionadas con la presencia de enfermdades cardiacas, pues para cada persona ademas de incluir los valores para las variables se incluyo si la persona esta enferma o no, esta ultima es la que nos permitio hacer la clasificacion

para ambos algoritmos.

Debido a la gran cantidad de datos, solo se consideraron los primero 30 esto debido a que tienen un equilibrio entre personas enfermas y no enfermas.

Para el algorimo KNN se genero la base de datos con las primeras 30 personas,se asigno un numero de puntos de 10. Finalmnte se definieron los valores a clasiificar y se aplico la funcion creada obteniendo una clasificacion que segun los parametros dados esta correspondio a una persona enferma o con riesgos de enfermar.

Por otro lado para el algoritmo de regresion, para una mayor precision se calculo el coefiente de correlacion biseral de cada una de las variables de la base datos con el diagnostico, una vez calculado se buscaron aquellas variables con un mejor coeficiente.

En funcion de estas variables previamente escogidas, se genero la base de datos, esta contenia valores para tabaco, adipositos y la edad, una vez generada se procedio a calular los coeficientes de regresion, debido a la complejidad de este calculo y la implementacion de un algoritmo para hacerlo se opto por usar la paqueteria numpy pues no se contaban con los conocimientos necesarios en elmachine learning, una vez calculados, se definieron como variables, se definieron las variables y se aplico la funcion,para esto se fijo una nueva clasificacion en la que la etiqueta 1 correspondiera a una persona enferma o con riesgo de enfermar mientras que para la etiqueta 2 se tendria una persona no enferma o sin riesgo de enfermar, lo que se obtuvo fue que segun los datos ingresados la persona pertence a la etiqueta 2.

Cabe mencionar que los datos ingresados en las vraiables independientes para la prueba que contempla la base de datos fueron escogidos al azar,sin embargo lo correcto seria usar valores reales pues esto permitiria acercarse mas al propisto de ser un modelo que permita predecir enfermedades cardiacas

Conclusión

Se desarrollaron de manera exitosa los algoritmos de machine learning, por lo que sería posible aumnetar la compplejidad del algoritmo para realizar una mejor prediccion, pues cabe mencionar que el algoritmo es basico. Aunado ha esto es que el algoritmo puede replicarse para cualquier parte del mundo, es decir que solo bastaria cargarle una base de datos diferente (dependiendo de la región que se quiera analizar), y podria predecir enfermedades. Por lo anterior se propone que como trabajo futuro, se podria programar para que fuera más preciso a la hora de detectar enfermedades cardiovasculasres, y que no solo funcionara para dichas enfermedades, si no para otras tambien.

Anexos

Enlaces

Codigo del reposotirio https://github.com/JohanCA38/Proyecto_Final.git

Enlace de la base de datos.

https://drive.google.com/file/d/1MzvF4aPqGAiHq83VZOJRTs9iLSPHq9pe/view?usp=sharing

Bibliografía

- ¿Qué es el algoritmo de k vecinos más cercanos? IBM. Obtenido de: https://www.ibm.com/mx-es/topics/knn
- Clasificación con Árboles de Decisión: el algoritmo CART. Codificando Bits. Obtenido de: https://www.codificandobits.com/blog/clasificacion-arboles-decision-algoritmo-cart/
- Regresión Lineal. Aprende Machine Learning. Obtenido de: www.aprendemachinelearning.com
- Regresion logistica.RPubs.Obtenido de : https://rpubs.com/jboscomendoza/correlacion.biserial.puntual.r