Predicción de Diabetes Usando Regresión Logística

Grant Nathaniel Keegan | A01700753

ABSTRACT

En este proyecto, voy a aplicar un algoritmo de regresión logística para predecir si un paciente tiene diabetes o no.

ÍNDICE

I. INTRODUCCIÓN

La diabetes es un problema global que ha crecido con el tiempo.[1] De acuerdo con la Organización Mundial de la Salud, los casos de diabetes han aumentado de 200 millones en 1990 a 830 millones en 2022. Los países más afectados son los de escasos recursos, y la prevención no es entendida del todo por gente que la padece.

Por esto es importante entender los síntomas de la diabetes, enfocado en prevención temprana, así como identificar quienes tienen más riesgo de sufrir de diabetes. El objetivo de este proyecto es generar un modelo de predicción efectivo para no solo prevenir diabetes, si no diagnosticar dentro de un rango de precisión (>90%) si una persona padece de ella.

Aquí voy a documentar cómo podemos predecir diabetes utilizando aprendizaje de máquina basado en procesamiento de datos. Analizando dos modelos efectivos utilizando un algoritmo de **regresión logística**, basada en un cuestionario a personas que están en riesgo, o ya padecen de diabetes.

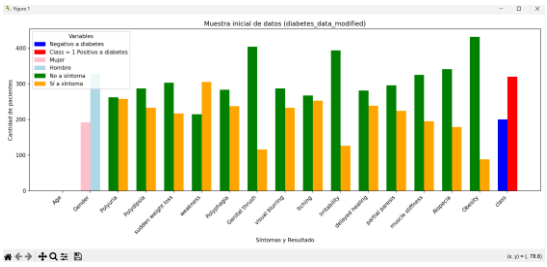
II. ANÁLISIS Y PROCESAMIENTO DE DATOS

Los datos que utilicé para este proyecto, fueron recolectados del artículo *Likelihood Prediction of Diabetes at Early Stage Using Data Mining Techniques.*[2] A través del recurso *UC Irvine Machine Learning Repository.* Los datos incluyen una muestra de 520 pacientes que tomaron una encuesta basada en síntomas. Los datos que incluyen son edad, sexo, una recolección de 14 síntomas, tales como alopecia, visión nublada, obesidad e irritabilidad. Finalmente, 'class', que indica si el paciente resultó positivo o negativo para la diabetes.

Para este proyecto, me voy a enfocar solamente en los datos binarios de los 14 síntomas, a los que los pacientes respondieron "yes" o "no". Y "class" a la cual el resultado dio positive o negative.

Para poder trabajar con los datos, es importante modificarlos usando procesos de ETL que sean fáciles de trabajar en los algoritmos implementados. Es importante saber si la tabla está completa, así que se implementa el comando `print(df.isnull().any())` para saber si hay espacios en blanco. Como no hay pasamos al siguiente paso, que fue convertir los valores de "yes" o "no" a 0 y 1 para cada uno de los síntomas. También "positive" o "negative" en class para el resultado de si una persona padece diabetes o no. Esto lo logré usando el comando df.replace().

Los resultados los podemos ver en la siguiente tabla de figura 1, donde muestra el género, cada síntoma de quienes tuvieron 1 o 0 (naranja o verde), y al final, cuantas personas padecen de diabetes después de la prueba.



Aquí podemos ver cómo los síntomas más raros fueron los de "Genital thrush", "Irritability" y "Obesity". Mientras que los más comunes fueron "weakness",

"polyuria" e "itching". Visualizar los datos de nuestra tabla antes de aplicar los algoritmos es importante, ya que nos da una representación gráfica de qué síntomas de diabetes son más comunes.

Es importante mencionar que en esta base de datos **no hay outliers** importantes que puedan afectar los resultados. Por lo que no aplique más procesamiento para eliminar o omitir datos o pacientes de esta base.

Además de eso, no apliqué más técnicas de procesamiento de datos, ya que la tabla original está completa y con interpretación efectiva para trabajar con mis modelos de regresión logística.

III. DISEÑO DEL MODELO SIN FRAMEWORK (REGRESIÓN LOGÍSTICA)

Mi objetivo para este proyecto es utilizar un modelo de regresión logística, basada en los datos de los 14 síntomas para predecir con alta exactitud si un paciente dará 0 o 1 en los resultados de su examen para detectar diabetes. Para este proyecto, me apoyé del recurso *How To Implement Logistic Regression From Scratch in Python*, de Jason Brownlee en Machine Learning Mastery.[3] Y de las notas de clase del módulo 2 y 3 del curso inteligencia artificial para la ciencia de datos I. Además de recursos de GeeksforGeeks y YouTube. (Referencias Adicionales).

III (a): Dividir datos entre entrenamiento (Train) y prueba (test).

El primer paso para poder entrenar mi modelo, es dividir los datos entre datos de entrenamiento, y datos de prueba.

IV. RESULTADOS DEL PRIMER MODELO SIN FRAMEWORK

Para correr el programa, es primero necesario generar la tabla adaptada para nuestro modelo de entrenamiento ejecutando el

V. DESARROLLO DEL MODELO MEJORADO

…

VI. RESULTADOS DEL MODELO MEJORADO

V. DESARROLLO DEL MODELO CON FRAMEWORKS

Para mi último modelo, voy a utilizar dos frameworks requeridos.

XGBOOST

VI. RESULTADOS CON EL FRAMEWORK

…

VII. CONCLUSIONES

…

NOTAS Y REFERENCIAS

[1] World Health Organization. (2023, April 5). *Diabetes*. World Health Organization. https://www.who.int/news-room/fact-sheets/detail/diabetes

[2] Islam, M., & Ferdousi, R. (2019). *Likelihood prediction of diabetes at early stage using data mining techniques*. Semantic Scholar. https://www.semanticscholar.org/paper/Likelihood-Prediction-of-Diabetes-at-Early-Stage-Islam-Ferdousi/9329dec57c5f13f195220ffa7077fd0029983f07

[2] Brownlee, J. (2019, June 17). *How to implement logistic regression with stochastic gradient descent from scratch with Python*. Machine Learning Mastery. https://machinelearningmastery.com/implement-logistic-regression-stochastic-gradient-descent-scratch-python

https://www.geeksforgeeks.org/machine-learning/xgboost/

https://www.geeksforgeeks.org/machine-learning/xgboost-parameters/

REFERENCIAS ADICIONALES

[1] GeeksforGeeks. (2024, January 19). Matplotlib tutorial. GeeksforGeeks.https://www.geeksforgeeks.org/python/matplotlib-tutorial/ **# Para modificar las tablas de MatPlotLib.**

[2] Hands-On Machine Learning: Logistic Regression with Python and Scikit-Learn. Ryan and Matt Data Science. https://www.youtube.com/watch?v=aL21Y-u0SRs