Universidad del Valle de Guatemala
Minería de datos
Sección: 20

Luis Quezada 18028
Jennifer Sandoval 18962
Esteban del Valle 18221

Avances del proyecto

Tema de proyecto:

Análisis de datos para la predicción de diabetes en una persona en base a sus antecedentes médicos y condiciones presentes. El dataset contiene 769 casos, con información sobre las condiciones físicas de cada paciente. La idea es hacer un una prueba con los algoritmos de machine learning seleccionados para construir un clasificador contundente que permita diagnosticar a una persona con diabetes, sin necesidad de que sea revisado por un médico.

Resultados de EDA

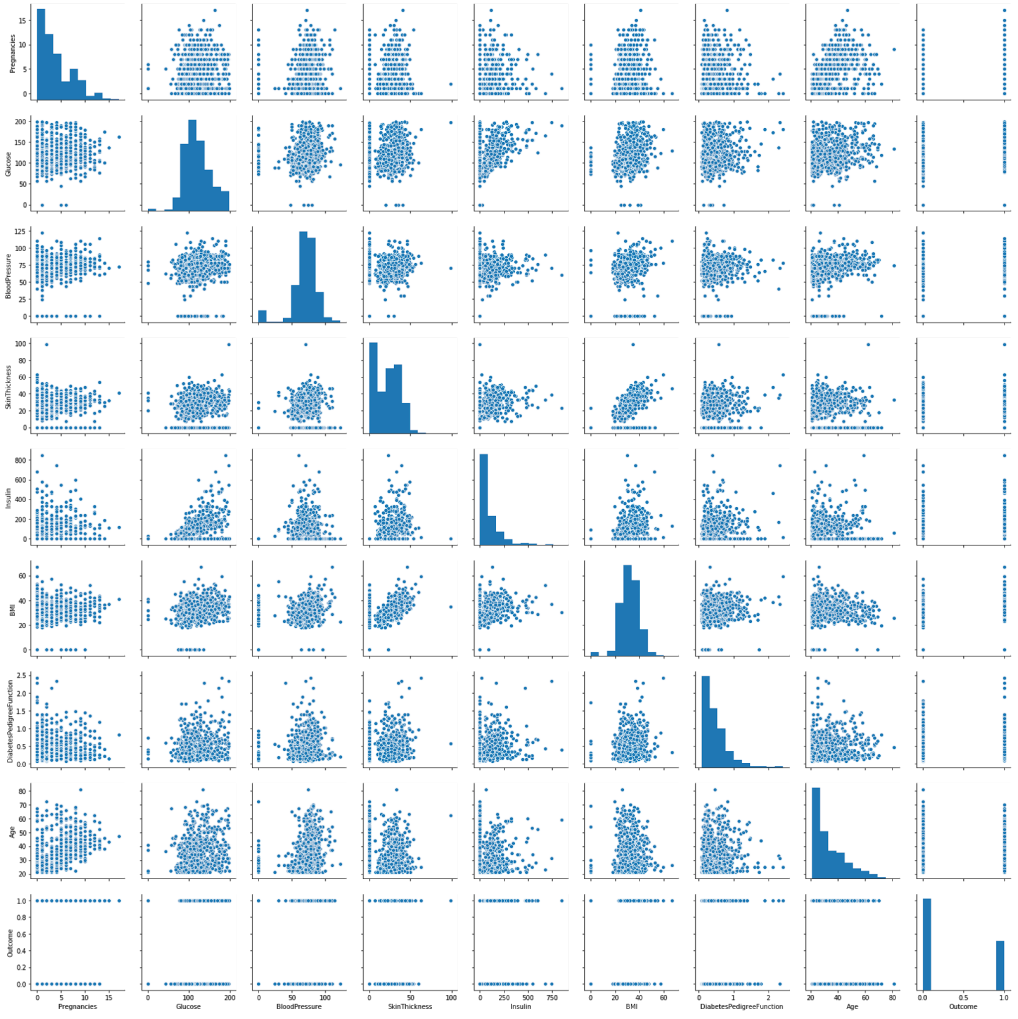

Figura 1. Pairplot de todas las variables involucradas en el set de datos utilizado

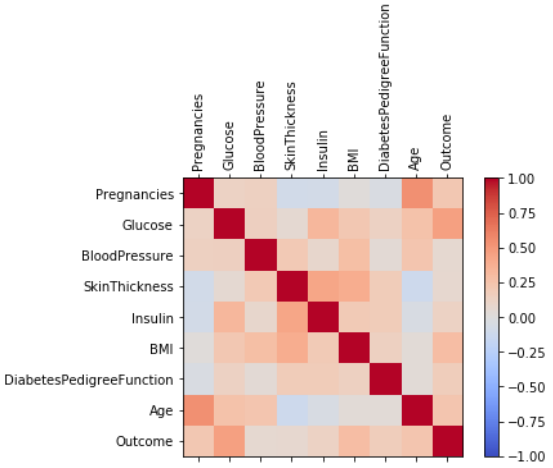

Figura 2. Diagrama de correlación de variables para el análisis de multicolinealidad

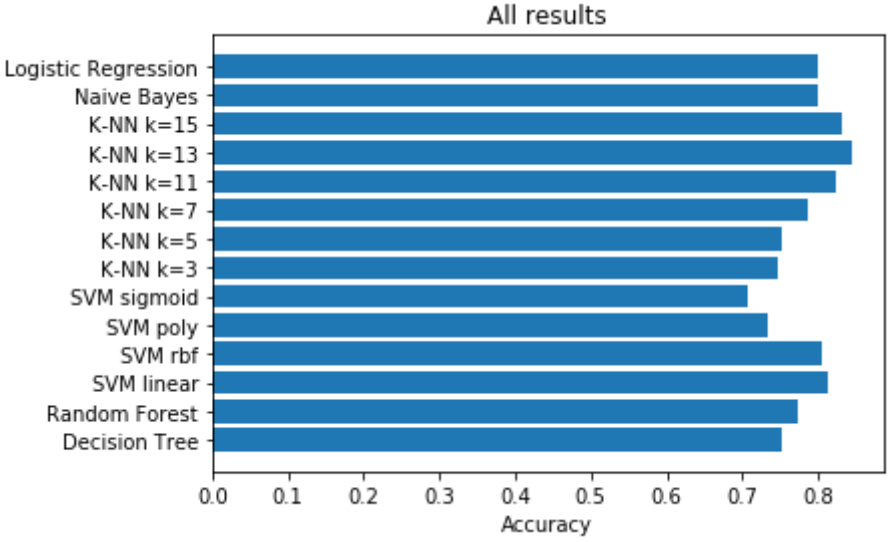

Figura 3. Exactitud de cada algoritmo de clasificación para el set de datos utilizado.

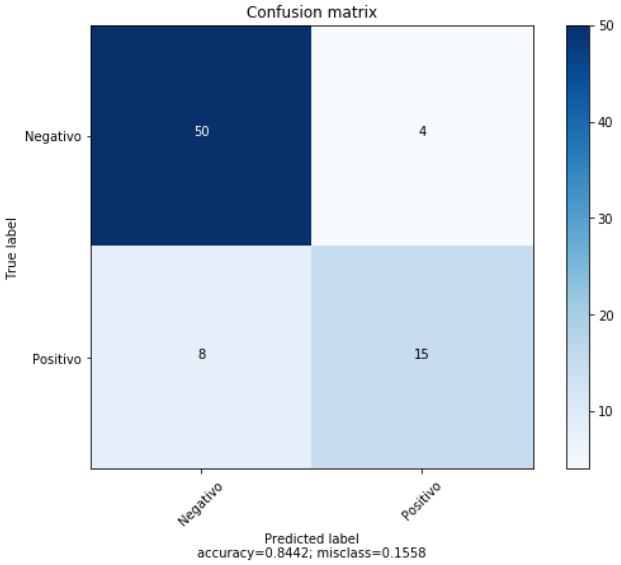

Figura 4. Exactitud de clasificación para el set de datos utilizado con RNA.

Algoritmos a utilizar

Se utilizó un módulo llamado ‘’Klas” (hecho por el grupo) que permite la comparación de los distintos algoritmos de clasificación y selecciona el algoritmo que muestre un mayor porcentaje de exactitud. Además, se realizaron varios modelos de RNA, de los cuales se obtuvo uno con un rendimiento alto por lo que también se decidió tomar en cuenta este acercamiento como parte de los algoritmos a utilizar. En base a esta comparación se decidieron utilizar los siguientes algoritmos para el set de datos:

<u>KNN</u>

Este algoritmo es un método de clasificación supervisado que permite calcular la probabilidad de que un elemento pertenezca a una clase, basándose en la información proporcionada por elementos asignado a una clase previamente. Así un elemento es asignado a la clase con mayor frecuencia de los ejemplos de entrenamientos que se encuentre más cercano al elemento. La distancia euclidiana es la que se utiliza para calcular las distancias entre cada elemento. Al utilizar un valor más grande de K se promueve la reducción del ruido en la clasificación (Srivastava, 2018).

- KNN con K=15
- KNN con K=13
- KNN con K=11

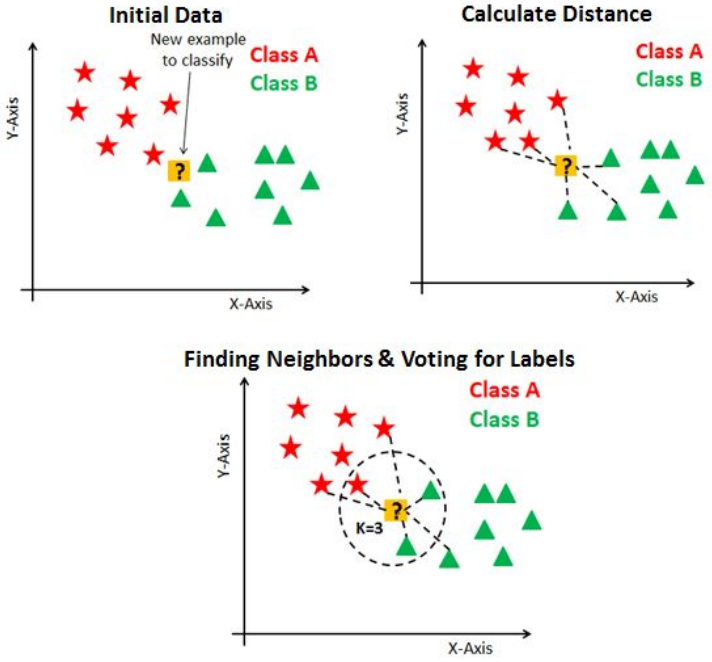

Figura 5. Ilustración de la clasificación de datos utilizando el algoritmo KNN

SVM

Es un algoritmo de clasificación que predice la clase de un elemento en base a clases previamente proporcionadas al modelo en la fase de entrenamiento. La forma en la que predice estas clasificaciones se basa en representar a los puntos de muestra en el espacio, separando las clases a espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los puntos, de las n clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una o la otra clase (Numerentur, 2019).

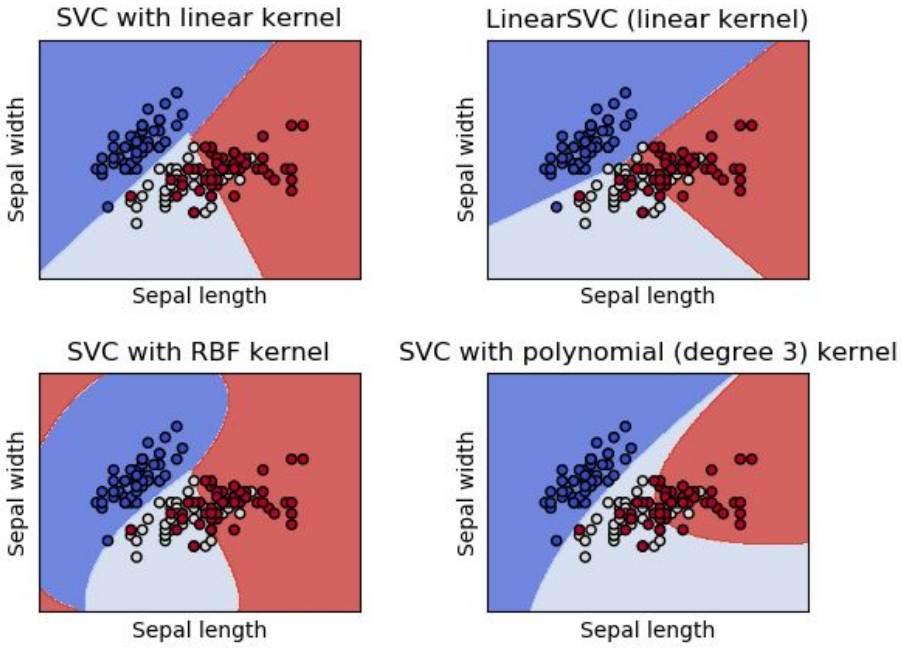

Figura 6. Ilustración de la clasificación de datos utilizando el algoritmo SVM

La diferencia entre el algoritmo SVM lineal y SVM RBF se basa en la forma de la delimitación del hiperplano. Como se observa en la figura 5, el algoritmo lineal utiliza limitaciones lineales para cada clase. Mientras que el SVM RBF utiliza limitaciones radiales para la separación de clases (Sckit-learn, 2020).

- SVM lineal
- SVM RBF

Redes Neuronales Artificiales

Las redes neuronales artificiales están basadas en cómo funcionan las redes de neuronas biológicas. Hay muchas variaciones de redes neuronales y diferentes arquitecturas. Una red neuronal está compuesta por neuronas, y cada una de ellas tiene un nivel de activación y un peso. El nivel de activación se puede representar como un número de 0 a 1, donde 0 es una neurona “apagada” y 1 es una neurona “activa” (Nielsen, 2019).

Las neuronas forman capas sobre capas, unas conectadas a otras. En la primera capa o input layer se da el ingreso de datos. Luego se encuentran las *hidden layers* que son las secciones internas de la red previas a la última capa; en estas se da la transformación de información. Pueden tener diferentes arquitecturas, formas, tamaños, etc. Finalmente está el *output layer* que es la capa final donde se obtiene el resultado. Cada capa puede tener una diferente función de activación, entre ellas Relu, Linear, Sigmoid, etc. Luego de crear el modelo, es necesario llevar a cabo un entrenamiento con información de prueba, donde los pesos de cada neurona se ajustan utilizando *back propagation*. El último paso es validar el modelo midiendo su precisión con información nueva.

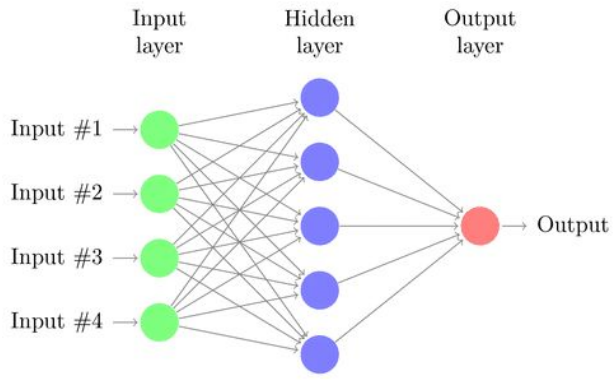

Figura 7. Ilustración de una red neuronal básica (Nielsen, 2019)

Referencias

Nielsen, M. (2019). Neural Networks and Deep Learning. 1 ed. MIT Press.

Numeretur. (2019). Máquina de soporte vectorial SVM. Extraído de: http://numerentur.org/svm/

Quezada, L. (2020). Klas Classifier . Extraído de https://github.com/Lfquezada/Klas-Classifier

Sckit-learn. (2020). Support Vector Machines. Extraído de: https://scikit-learn.org/stable/modules/svm.html

Srivastava. (26 de marzo del 2018). K nearest neighbor. Extraído de: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/