Proyecto de Minería de Datos

Proyecto Final

Índice

Contenido

Introducción

Describe los datos, es decir, que información contiene y de donde obtuviste la base de datos que vas a utilizar. Incluye una descripción del problema (preguntas que quieren resolver).

La base de datos utilizada fue descargada de la plataforma kaggle y los datos provienen originalmente del Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales de Estados Unidos. El objetivo es predecir en base a medidas diagnósticas si un paciente tiene diabetes, por lo que se incluyen 768 observaciones (con una usabilidad del 100%) constituidas por 8 características y dos etiquetas o posibles resultados. Adicionalmente, el proveedor de los datos puntualiza que se impusieron ciertas restricciones para la selección de las instancias de esta base de datos de una más grande. En particular, todos los pacientes son mujeres de al menos 21 años, de ascendencia indígena pima.

Características

Embarazos: Número de veces embarazada

Glucosa: Concentración de glucosa en plasma a las 2 horas en una prueba de tolerancia oral a la glucosa

Presión arterial: Presión arterial diastólica (mm Hg)

Grosor de la piel: Grosor del pliegue cutáneo del tríceps (mm)

Insulina: insulina sérica de 2 horas (mu U/ml)

IMC: Índice de masa corporal (peso en kg/(altura en m)^2)

DiabetesPedigreeFunction: Función de pedigrí de diabetes, puntuación basada en el historial familiar

Edad: Edad (años)

Etiquetas

Resultado: Variable de clase dependiendo si se tiene diabetes o no (0 o 1)

La diabetes mellitus constituye uno de los principales problemas de salud en el mundo, ya que hay cerca de 100 millones de diabéticos en el planeta. La prevalencia de esta enfermedad está incrementándose de forma importante en las poblaciones en vía de desarrollo debido al envejecimiento de la población, el cambio de hábitos dietéticos (mayor consumo de azúcares refinados) y un descenso de la actividad física, lo que también conlleva a un aumento de las personas obesas (González, 2011). Además, muchas de las personas que sufren la enfermedad desconocen su situación, como ocurre con muchas personas obesas.

En el presente trabajo se busca entrenar modelos de machine learning para poder ser aplicados en el campo de la medicina, específicamente facilitar el diagnóstico de personas con diabetes. Se busca llegar a un diagnóstico lo más preciso posible dadas las medidas diagnósticas de un paciente.

Créditos

Nombre y puesto de los integrantes. (Analista, programador, data scientist…)

Gerardo Moreno Zizumbo: Programador

Cristina López Ontiveros: Data Scientist

Diego Alejandro Flores Meza: Data Scientist

Sofía Ingigerth Cañas Urbina: Analista

Método 1: Red Neuronal Artificial

Explicación del método

Describe el funcionamiento del método visto en clase. Sus partes y cómo estarás usando los datos de entrada y el entrenamiento.

Una red neuronal artificial (ANN) tiene como objetivo en cada una de la una o múltiples capas transformar los datos por medio de multiplicación y suma de matrices con las matrices de pesos y de bias para finalizar con una función de activación y de esta forma regresar en la última capa como output el valor de predicción. Las redes neuronales se pueden configurar por medio de la última capa para resolver un problema de regresión o de clasificación. Para el caso actual, la última capa tiene como función de activación una sigmoidal, de esta forma se asegura que el resultado será un valor numérico entre el 0 y 1 que representa la probabilidad de que el paciente tenga la enfermedad. El "entrenar" a la ANN es influenciado por una función de costo y que por medio de retropropagación, se toma el resultado de esta función para ajustar las matrices, este proceso iterativo continúa por un número de veces especificadas por el programador.

Los datos de entrada son una matriz de tamaño [n, 8] donde n es el número de casos de entrada que le damos y el 8 representa el número de variables con las que se calculará la predicción.

Objetivo

Predecir si un cierto paciente presenta diabetes o no.

Implementación

La red neuronal se implementó con la librería keras.

En primer lugar se definió el modelo, este es una red neural "shallow" debido a que solo implementé una capa. La activación de dicha capa es Rectified Linear Unit (Relu) y como previamente se explicó, la capa de output tiene una función de activación sigmoidal para obtener la probabilidad buscada. Se especificó la función de pérdida de función de pérdida binaria con entropía y se compiló el modelo, como optimizador, se utilizó el optimizador de Adam. Para entrenar el modelo se dividió la base de datos en datos de entrenamiento y datos de prueba. A partir de la época número 100, las métricas no mejoran es por eso que no se continuó entrenando el modelo más allá de dicha época.

Evaluación

El reporte de clasificación es el siguiente:

	Precisión	Sensibilidad	Score F1
0	0.82	0.86	0.84
1	0.73	0.66	0.69
Exactitud			0.79

Método 2: Árbol de decisión con la librería Scikit-Learn

Explicación del método

Describe el funcionamiento del método que hayas visto en clase. Sus partes y cómo estarás usando los datos de entrada y el entrenamiento.

Este tipo de método es uno de los más fáciles y más usados con el aprendizaje automatizado supervisado, y este se entrena con una base de datos que tenga los datos de salida correctos, por lo que la máquina va a aprender de los patrones que se le den, por mismo, la base de datos debe de ser balanceada y amplia. Un Árbol de decisiones tiene 4 componentes principales, los cuales son;

- Root node: Es el primer nodo del árbol, este es el que da mayor información por lo que de ahí va a sacar sus primeras “raíces” las cuales son los Nodes.
- Nodes: Esta será la raíz que sería la respuesta al nodo con una respuesta de “Sí” o “No”.
- Leaf nodes: Esta sería la respuesta final, por ejemplo, en el caso de nuestra base de datos sería si tiene diabetes o no.

Objetivo

Predecir un resultado positivo o negativo a diabetes.

Implementación

Cómo codificaste y usaste el algoritmo. No tires todo el código aquí, platica las partes relevantes.

Para la elaboración de nuestro árbol de decisión nosotros usamos la librería de Scikit-Learn, pero para poder hacer empezar a realizar nuestro análisis los datos deben de ser numéricos por lo que en el caso de que sean strings se van a tener que encriptar con la función “preprocessing.LabelEncoder()”, eventualmente se crean las etiquetas de las características y de

ahí se entrena el algoritmo para después desplegar con un plot nuestro árbol y apreciar cómo nuestro método clasifica los nodos para tener la solución más óptima.

Evaluación

	Precisión	Sensibilidad	Score F1
0	0.80	0.84	0.82
1	0.62	0.56	0.59
Exactitud			0.75

Método 3: K-Nearest Neighbors

Explicación del método - Diego Flores

El modelo del vecino más cercano es un algoritmo de instancia supervisado el cual puede ser utilizado para clasificar muestras o predecir nuevos datos. El método se enfoca totalmente en las observaciones más cercanas de las cuales se busca predecir y clasificar el dato de interés en base a los datos que lo rodean. Uno de los principales contratiempos que se producen al utilizar este método es que es necesario hacer uso de todo el dataset para entrenar el modelo lo cual utiliza más memoria y capacidad de procesamiento.

Objetivo

Clasificar y predecir.

Implementación

Para poder realizar el código es necesario hacer unos cambios en la base de datos para que los resultados salgan con menor error, los principales cambios sería rellenar las celdas que no cuentan con ningún tipo de dato estos se reemplazarían con la mediana o con la media según sea el caso, haciendo tendríamos una base de datos que parecería normal. Después procedemos a usar KNeighborsClassifier de la librería sklearn.neighbors para implementar el modelo, tratamos con diferentes valores de k y el que daba el valor máximo es el de k=11.

Evaluación

	Precisión	Sensibilidad	Score F1
0	0.80	0.85	0.83
1	0.68	0.61	0.64
Exactitud			0.76

Método 4: Support Vector Machine

Explicación del método - Sofía Cañas

Describe el funcionamiento del método que hayas investigado. Sus partes y cómo estarás usando los datos de entrada y el entrenamiento.

Una máquina de vectores de soporte (support vector machine o SVM) es un tipo de algoritmo de clasificación basado en aprendizaje automático supervisado. Es muy preferido por muchos, ya que produce una precisión significativa en relación a una menor potencia computacional. El objetivo de SVM es encontrar un hiperplano en un espacio N-dimensional (donde N es el número de características) que clasifique claramente los puntos de datos. En dos dimensiones un hiperplano es una recta y en tres, un plano. Para mayores dimensiones es algo geométricamente abstracto, pero en general con una definición matemática a partir de vectores de soporte, que son los puntos más cercanos al separador.

Para separar dos clases de puntos de datos, se pueden elegir muchos hiperplanos posibles. Nuestro objetivo es encontrar un plano que tenga el margen máximo, es decir, la distancia máxima entre puntos de datos de ambas clases. Maximizar la distancia del margen proporciona cierto refuerzo para que los puntos de datos futuros se puedan clasificar con más confianza. A menudo, los datos que no son linealmente separables en el espacio de entrada original son fácilmente separables en un espacio de mayor dimensión, para lo cual se usa un truco llamado kernel. El separador de mayor dimensión es en realidad no lineal en el espacio original y esto significa que el espacio de hipótesis se expande mucho sobre los métodos que usan representaciones estrictamente lineales.

Los datos de entrada son una matriz de tamaño [n, 8] donde n es el tamaño del training set y 8 es el número de características o en este caso dimensiones de cada dato.

Objetivo

A partir de la manipulación matemática de los valores en el training set se determinará un hiperplano en 8 dimensiones (la cantidad de características de cada dato) que dividirá el espacio entre las dos etiquetas, en sí el objetivo del algoritmo es clasificar.

Implementación

Para el entrenamiento de este modelo se importa la librería Scikit-Learn. Tambien se usa seaborn para poder visualizar los datos con pairplot y darnos una idea de si los datos son linearmente separables o no y considerar el uso de kernel, en este caso es necesario.

Como preprocesamiento se dividen los datos en atributos (X) y etiquetas (Y) y luego en training y testing set con la función train_test_split en la que se especifica la proporción correspondiente al testing set, se maneja entre un 30% y 10% dependiendo de los resultados obtenidos, la exactitud se ve optimizada con 10% y esto puede ser porque hay un mayor entrenamiento.

Por último se entrena el modelo con las funciones SVC y fit, y se evalúa el algoritmo. Se probaron distintos kernels (polynomial a diferentes grados, gaussian, sigmoid) y el que arrojó una mejor evaluación fue el gaussiano. El algoritmo se prueba con el testing set, usando una matriz de confusión y un reporte de la clasificación. De ser requerido se puede usar para realizar predicciones específicas plasmando las características en vectores e ingresándolos al modelo.

Evaluación

	Precisión	Sensibilidad	Score F1
0	0.77	0.92	0.84
1	0.80	0.55	0.65
Exactitud			0.78

Comparación

Algoritmo Redes Neuronales - Resultado = 0.79, es un modelo muy fácil de entrenar ya que realizar tareas de entrenamiento inicial el cual es aprendizaje adaptativo, el principal contratiempo del algoritmo es que necesita mucho poder computacional y un base de datos grande y completa.

Algoritmo KNN - Resultado = 0.76, la principal ventaja es que es fácil de implementar pero es necesario hacer uso de todo el dataset para entrenar el modelo lo cual utiliza más memoria y capacidad de procesamiento.

Algoritmo Árbol de decision - Resultado = 0.75, este modelo es muy fácil de interpretar y no es necesario realizar tanta preparación de los datos ya que el modelo es capaz de manejar diferentes tipos de datos, pero suele ser inestable y no se garantiza un resultado óptimo.

Algoritmo Support Vector Machine - Resultado = 0.78, posee la ventaja de una precisión significativa en relación a una menor potencia computacional cuando la cantidad de datos es relativamente pequeña. Cuando los datos son linearmente separables se utiliza un modelo lineal, cuando no, es necesario usar un kernel que se integra en el mismo entrenamiento y otra ventaja es que existen diferentes kernels y se puede seleccionar el que arroje mejores resultados. Es muy eficaz para dimensiones grandes (varias características), pero no para enormes cantidades de datos porque implica una complejidad O(n^3). Ademas de que es un modelo propenso al exceso de ajuste, lo que resultaría en una desventaja crucial si quisiéramos mejorar la exactitud alimentando el aprendizaje con más datos.

Conclusiones

Posibles Mejoras

Las mejoras a implementar serían en primer lugar necesarias en la base de datos ya que esta cuenta con una cantidad de observaciones relativamente pequeña (~700), afortunadamente con una usabilidad de prácticamente el 100%, pero debemos recordar que a más observaciones, se obtienen resultados más apegados a la realidad (no necesariamente mejores pero en situaciones médicas es valioso apegarse a la certidumbre) porque el aprendizaje de la máquina se basa en estas observaciones.

La otra mejora posible reside en los modelos utilizados, hemos aplicado todos los vistos en clase y uno más investigado por cuenta propia, además de variaciones posibles de cada uno, siempre buscando la mejor exactitud al evaluar los algoritmos obtenidos. Una base de datos es un mundo particular, un acomodo específico en un espacio n-dimensional y cada modelo presenta sus ventajas y desventajas de acuerdo a las características particulares de los datos. Se puede considerar probar otros modelos o agregar mejoras a los actuales.

Algoritmo seleccionado

Método 1: Red Neuronal Artificial

Esta elección se debe a que fue el modelo que arrojó la exactitud más alta, además de que es congruente con la problemática específica analizada. Se puede adaptar a una base de datos de mayor tamaño y ofrece flexibilidad en el ajuste a una situación de la vida real como esta, donde los pacientes son individuos con casos muy particulares.

Reflexiones

Gerardo Moreno

Sin duda una demostración de cómo se pueden aplicar los conocimientos matemáticos y el poder computacional para optimizar un proceso tan importante para la sociedad como lo es el predecir la presencia de enfermedades. Una de las cosas que más me ha hecho reflexionar es la relativa rapidez y facilidad con la que se pueden implementar un modelo y hacerlo predecir de manera aceptable resultados. Una demostración más del potencial que tiene el campo de la inteligencia artificial, en este caso, especialmente el de aprendizaje de máquina y las oportunidades que se tienen para mejorar procesos y en este caso hasta posiblemente salvar vidas.

Cristina López

Actualmente el conocimiento y la información pueden considerarse como un sinónimo de poder, y esto se debe a que entre más datos tengas (como en este caso de el área médica) se puede usar tanto como para mejorar y optimizar los servicios como los horarios o porciones de medicamentos, sin embargo, este conocimiento también se puede para propósitos egoístas y consumistas.

Diego Flores

Cada día contamos con muchos más datos e información la cual nos permite ser capaces de implementar estos modelos de machine learning en diferentes ámbitos de la vida diaria. La implementación del machine learning en la medicina creo que es algo muy relevante ya que este ayuda a predecir enfermedades mucho antes con los doctores y realizar un diagnóstico precoz el cual ayudaría a salvar más vidas. En mi opinión se debe continuar la implementación del machine learning en la medicina siempre y cuando los datos estén protegidos de la manera correcta.

Sofía Cañas

Cada modelo nos presentó un acercamiento diferente a la problemática, enseñándonos distintas maneras de analizar los datos con planteamientos matemáticos y posibles ventajas y desventajas para situaciones específicas. El trabajo en equipo fue fluido y se pudieron comparar los diferentes resultados con éxito. Conocer el mecanismo que hay detrás de cada modelo de machine learning nos hace apreciar la rapidez con la que una computadora puede realizar operaciones en una base de datos para entrenar un modelo que resulta muy relevante, esto constituye la base de la inteligencia artificial. La máxima exactitud obtenida fue de 79%, en el caso de SVM se probaron varios kernels para poder maximizar la exactitud y en mi opinión para tratarse de una situación médica es una proporción aceptable pero debería buscarse una mayor. La ciencia de datos es multidisciplinaria pero cuando se tratan temas de la salud y la vida de las personas deben existir regulaciones éticas y cierto grado de confianza en los resultados. En esta situación por ejemplo, podría tratarse de una prueba complementaria que va por muy buen camino para determinar si es diabetes o no y conforme se tengan aún más datos y se ajusten los modelos se puede llegar a algo con mayor confianza.

Referencias

Incluyan los trabajos consultados en formato APA.

Borcan, M. (2020). *Decision Tree Classifier Tutorial in Python and Scikit-Learn*. Programmer Backpack. Recuperado de : https://web.archive.org/web/20210418080446/https://programmerbackpack.com/decision-trees-classifier/

Gandhi, R. (2018) *Support Vector Machine — Introduction to Machine Learning Algorithms*. Towards Data Science. Recuperado de: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

González, L. (2011) *Características, diagnóstico y tratamiento de la diabetes*. Offarm Farmacéutica. Vol. 20. Núm. 7 (72-80). Recuperado de: https://www.elsevier.es/es-revista-offarm-4-articulo-caracteristicas-diagnostico-tratamiento-diabetes-13018328

Malik, U. (2019) *Implementing SVM and Kernel SVM with Python's Scikit-Learn*. StackAbuse. Recuperado de: https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/

Russell, S. & Norving, P. (2010) *Artificial Intelligence: A Modern Approach*. Third ed. Prentice Hall.

A, M. (2022) *Descubra el algoritmo KNN : un algoritmo de aprendizaje supervisado.* Formación en ciencia de datos | DataScientest.com. Recuperado de: https://datascientest.com/es/que-es-el-algoritmo-knn

Robinson, S. (2021, 21 noviembre). K-Nearest Neighbors Algorithm in Python and Scikit-Learn. Stack Abuse. https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/

Anexos

Datos

Base de datos: Diabetes Dataset
https://www.kaggle.com/datasets/mathchi/diabetes-data-set

Código

Incluyan su código completo en python.

1. https://colab.research.google.com/drive/15bRMTmmHlqL6kXw4jvw1uBRjKyeh1sD5?usp=sharing
2. https://colab.research.google.com/drive/1HwU55tMZW5UUTT2QGzFr1o_lTQfpxyh6?usp=sharing
3. https://colab.research.google.com/drive/1mqxoZAcnVxaBp07mDcRDXJWGhhE-7WcM?usp=sharing
4. https://colab.research.google.com/drive/14GbZtJTAabKazpL4NkAvNdeoiveSb6lY?usp=sharing

Evidencias de trabajo en equipo

Incluyan capturas de pantalla que evidencien su trabajo en equipo.