**INTELIGENCIA ARTIFICIAL - INGENIERÍA INFORMÁTICA**

**PECL3**

Adrián Martínez Díaz

Raúl López Llana

Elena Martín Naranjo

Luis Ángel Barderas San Segundo

**PREGUNTA 1:**

El problema consiste en una simulación de diagnóstico de una enfermedad como es la diabetes, lo cual es en esencia, la clasificación de los datos de una forma binaria, en función de los valores de sus atributos.

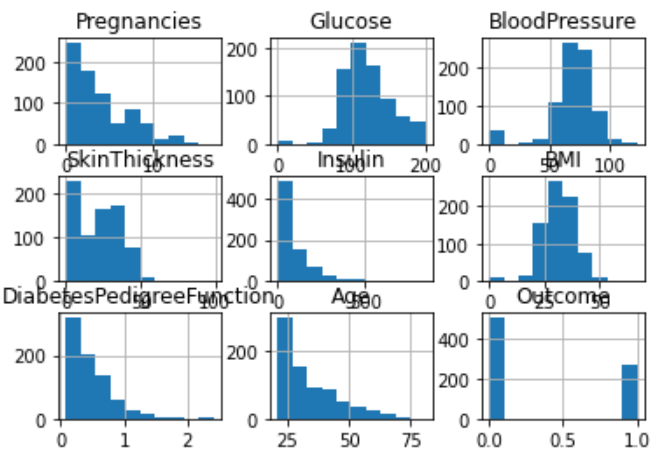
Para crear los modelos para el estudio y clasificación de los datos obtenidos se utilizarán técnicas de aprendizaje inductivo, analógico, bayesiano y redes artificiales; estas serían k-vecinos próximos, árboles de clasificación y Naive Bayes y redes neuronales artificiales. Se debe razonar la comparación de los resultados de los distintos modelos.

Se debe escoger razonadamente unos conjuntos de test, entrenamiento y validación, así como los valores de los hiper-parámetros para conseguir los mejores resultados posibles.

**PREGUNTA 2:**

Hay diferentes alternativas en cuanto al preprocesamiento de datos, en nuestro caso hemos elegido el escalado de datos porque en modelos con múltiples características, cuando los valores son numéricos, la escala en la que se miden, así como la magnitud de su varianza puede tener gran influencia en el modelo; de manera que aquellos datos que se midan en una escala mayor o que tengan más varianza dominarán el modelo aunque no tengan mayor relación con la variable resultado.

En nuestro caso, las características van desde escalas 0-1 a escalas de 0-200:



Por esto, hemos visto necesario que todas las características tomen escalas similares para que sean tenidas en cuenta de forma similar en el entrenamiento de los modelos. Por tanto, el valor de cada característica estará acotado entre 0 y 1.

**PREGUNTA 3:**

Hoy en día 90%/5%/5% es una distribución de los conjuntos de datos muy empleada pues actualmente los modelos trabajan con una gran cantidad de datos. Puesto que nuestro modelo trabajará con una cantidad menor, los porcentajes asignados serán 80%/10%/10%

El objetivo fundamental es reducir tanto como sea posible el error de entrenamiento, no obstante, si asignamos un porcentaje demasiado bajo al test y validación, los resultados arrojados por estos serán más sensibles a outliers y darán un error mayor que el de training, lo indica un sobreajuste del modelo al training data, perdiendo éste capacidad de adaptación. El error del conjunto de validación debe ser similar al error del conjunto de test y training.

Para evitar el problema de una selección adversa del conjunto de pruebas y obtener una estimación más fiable del error de validación, se utiliza el método de validación cruzada.

**PREGUNTA 4:**

Para hallar los hiper parámetros óptimos para cada modelo hemos evaluado el rendimiento de cada modelo mediante validación cruzada, dividiendo el grupo de datos de entrenamiento en 5 grupos, con varios posibles valores de dichos hiper-parámetros. Una vez evaluados todos los posibles valores, se seleccionará como hiper-parámetro óptimo el que haya causado la mayor precisión posible del modelo. Para cada modelo, se indican a continuación los valores evaluados y el valor óptimo de los hiper-parámetros de dicho modelo:

k-nn (número de vecinos):

valores evaluados -> 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

valor óptimo -> 11

árbol de clasificación (profundidad del árbol):

valores evaluados -> 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

valor óptimo -> 2

bayes naive ( parámetro de suavizado):

valores evaluados -> 0.1,0.2,0.3,0.4,0.5

valor óptimo -> 0.1

feed forward network (número de capas + número de neuronas por capa): Para encontrar los hiperparámetros óptimos hemos hecho uso de for anidados para hallar el número de neuronas de cada capa. Sin embargo, este algoritmo tarda al menos 1 día en analizar todos los posibles valores, así que una vez obtenidas las neuronas por capa que permitían un mejor rendimiento del modelo, hemos evaluado qué combinación de dichos valores más los distintos números de capas que genera un modelo más óptimo, reduciendo así el tiempo de ejecución.

valores evaluados -> n\_capas: 1,2,3 neu\_capa1: 31,25,16 neu\_capa2: 97, 91 neu\_capa3: 61

valor óptimo -> n\_capas: 3 neu\_capa1: 31 neu\_capa2: 97 neu\_capa3: 61

**PREGUNTA 5:**

Para elegir qué modelo es el mejor de todos hemos empleado la matriz de confusión para ver los aciertos y fallos de cada tipo (True negative, False negative, True positive, False positive) en cada uno de los modelos.

De esta forma, hemos podido comprobar los errores en los que el modelo decía que no había diabetes cuando sí que la había y viceversa.

Para tomar la decisión de cuál es el mejor modelo, hemos tenido en cuenta además el tipo de fallo ya que vemos más perjudicial que no se detecte un caso en el que sí que hay diabetes por las consecuencias que esto puede conllevar.

Los resultados de precisión obtenidos son:

KNN: True negative:47, False negative:15, True positive:9, False positive:6

arboles: True negative:49, False negative:16, True positive:8, False positive:4

Naive Bayes: True negative:45, False negative:11, True positive:13, False positive:8

Red neuronal: True negative:44, False negative:7, True positive:17, False positive:9

El modelo que ha dado un menor número de fallos es la red neuronal. Además, disminuye bastante los casos de False Negative en comparación con el resto de modelos, que es considerado un fallo más perjudicial que el False negative por lo dicho anteriormente.

Teniendo en cuenta estas premisas, vemos que el modelo que ha realizado las mejores predicciones es la red neuronal..

**PREGUNTA 6:**

Una vez halladas las predicciones sobre el conjunto de datos de test, pasamos a realizar las predicciones sobre el conjunto de datos de validación y obtener la matriz de confusión respectiva para comprobar que los resultados son similares al conjunto de test.

La matriz de confusión para el conjunto de validación es la siguiente:

True negative:40, False negative:9, True positive:22, False positive:6

Observamos por tanto que los resultados son similares a los obtenidos sobre el conjunto de test y que siguen siendo mejores predicciones que las realizadas por el resto de modelos anteriormente, concluyendo así que efectivamente el modelo más óptimo se trata de la red neuronal.

**PREGUNTA 7:**

Como modelo lineal hemos decidido emplear un perceptrón para la resolución de este problema, comparando los resultados obtenidos en este con los del mejor modelo anterior que era la red neuronal alimentada hacia delante multicapa.

Para hacer la comparación hemos empleado la matriz de confusión, tanto para el conjunto de datos de test como el de validación:

True negative:37 False negative:20 True positive:4 False positive:16 <- datos test

True negative: 38 False negative: 27 True positive: 4 False positive: 8 <- datos validación

Vemos que las predicciones obtenidas son bastante peores que las del modelo escogido, por lo que seguimos considerando la red neuronal como el modelo óptimo.