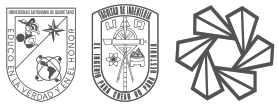


UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA
DIVISIÓN DE INVESTIGACIÓN Y POSGRADO

Machine Learning: Práctica 5. K Nearest Neighbours.

Alumno:
Ing. Enrique Mena Camilo

Profesor:
Dr. Marco Antonio Aceves Fernández

Junio 2023

Índice

1. Objetivos

El objetivo principal de esta práctica es implementar el algoritmo K Neares Neighbours, el cual será probado con un conjunto de datos sobre clasificación de accidente cerebrovascular, haciendo uso de la validación en línea y probando distintos métodos de selección de atributos.

2. Introducción

La detección temprana y precisa de un accidente cerebrovascular, también conocido como stroke, es de vital importancia para la atención médica de emergencia y el tratamiento adecuado de los pacientes. El stroke es una condición médica grave que ocurre cuando el suministro de sangre al cerebro se interrumpe o se reduce significativamente, lo que resulta en daño cerebral. Identificar rápidamente los signos de un stroke y tomar medidas inmediatas puede marcar la diferencia entre la vida y la muerte, así como también puede prevenir discapacidades graves y duraderas.

En este contexto, el desarrollo de algoritmos de clasificación para la detección de stroke ha demostrado ser una herramienta prometedora. Estos algoritmos están diseñados para analizar y procesar grandes cantidades de datos clínicos y de imagen, como resultados de pruebas médicas, imágenes de resonancia magnética y registros de síntomas. Al aplicar técnicas de aprendizaje automático y análisis de datos, estos algoritmos pueden identificar patrones y características específicas asociadas con la presencia de un stroke.

Los algoritmos de clasificación para la detección de stroke pueden ayudar a los profesionales de la salud a tomar decisiones más informadas y precisas en cuanto a la evaluación de pacientes. Pueden proporcionar una evaluación objetiva y cuantitativa de los riesgos de un individuo de sufrir un stroke, lo que facilita la toma de decisiones sobre los pasos a seguir en términos de diagnóstico y tratamiento. Además, estos algoritmos pueden ser utilizados para desarrollar sistemas de alerta temprana que notifiquen a los médicos sobre la posible presencia de un stroke en pacientes en riesgo, permitiendo una intervención médica más rápida y eficiente.

3. Marco teórico

3.1. K-Nearest Neighbors

El algoritmo K-Nearest Neighbors (KNN) es un algoritmo que pertenece a los algoritmos de aprendizaje supervisado. Como primera etapa se debe seleccionar un número k de vecinos, posteriormente se calcula la distancia de cada uno de los puntos hacia todos los demás. Se toman los k vecinos más cercanos y se le atribuye al punto que se está analizando la clase más frecuente en los vecinos que se analizan. Finalmente se selecciona el mejor número de vecinos.

3.2. Análisis de componentes principales

El Análisis de Componentes Principales (PCA) es una técnica estadística que busca reducir la dimensionalidad de los conjuntos de datos complejos. Se basa en encontrar las direcciones principales que capturan la mayor variabilidad en los datos. Al proyectar los datos en estas direcciones, se conserva la información más relevante mientras se reduce la dimensionalidad. El PCA es utilizado para eliminar correlaciones, detectar patrones y anomalías, y preparar datos para algoritmos de aprendizaje automático. Es una herramienta fundamental en el análisis exploratorio de datos.

3.3. Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson, es una medida estadística que evalúa la fuerza y dirección de la relación lineal entre dos variables continuas. Es ampliamente utilizado para medir la correlación entre dos variables, donde un valor de +1 indica una correlación perfectamente positiva, un valor de -1 indica una correlación perfectamente negativa, y un valor de 0 indica una falta de correlación.

El coeficiente de Pearson se calcula como la covarianza entre las dos variables dividida por el producto de las desviaciones estándar de las dos variables. La fórmula matemática del coeficiente de Pearson es:

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})_i}}$$

Donde r es el coeficiente de Pearson. x_i y y_i son los valores de las dos variables continuas en la i-ésima observación. $\bar{x}$ y $\bar{y}$ son las medias de las dos variables continuas.

3.4. Validación en línea

Este método consiste en la creación de subconjuntos, en los cuales el 70 % de los datos se considera para el entrenamiento, mientras que un 30 % es para las pruebas del modelo de clasificación. Sin embargo, estos porcentajes pueden variar según las necesidades de la base de datos y el proyecto.

4. Materiales y métodos

Para el desarrollo de esta práctica se utilizó el lenguaje de programación Python en su versión 3.10, con el que se diseñó un conjunto de scripts y una libreta de Jupyer para cumplir con los objetivos de la práctica.

4.1. Conjunto de datos utilizado

El conjunto de datos utilizado para esta práctica consta de una colección cuyo objetivo es la predicción de un derrame cerebral (stroke, en inglés). Dicho conjunto de datos fue obtenido de la plataforma Kaggle, y consta de 10 atributos de diversos tipos y 1 variable objetivo, teniendo un total de 40,910 instancias. El conjunto de datos contiene 2 clases en la variable objetivo, las cuales se encuentran balanceadas en porciones del 50 %. Los tipos de datos disponibles en el conjunto de datos empleado son:

- Ordinales
  - age
- Continuos
  - avg_glucose_level
  - bmi
- Categóricos
  - sex
  - hypertension
  - heart_disease
  - ever_married
  - work_type
  - residence_type
  - smoking_status
  - stroke (variable objetivo)

4.2. Pre-procesamiento de datos

4.2.1. Imputación de datos

El conjunto de datos contaba con 3 instancias con datos faltantes en el atributo *sex*. Dada la gran cantidad de instancias disponibles, se optó por rellenar estos valores faltantes usando el valor más frecuente de dicho atributo. En la Figura 1 se puede observar la distribución del atributo *sex* posterior al proceso de imputación.

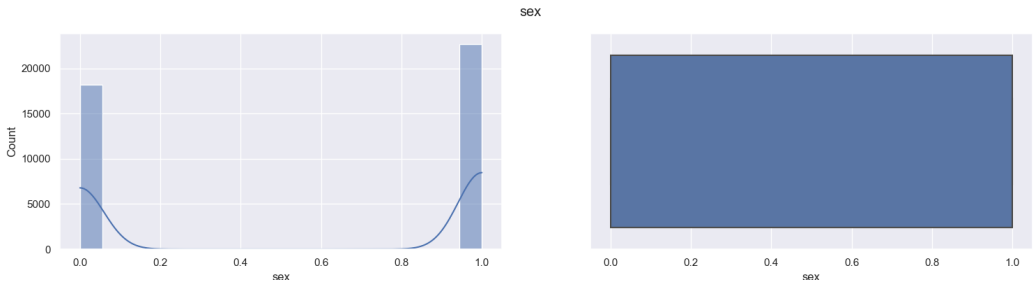
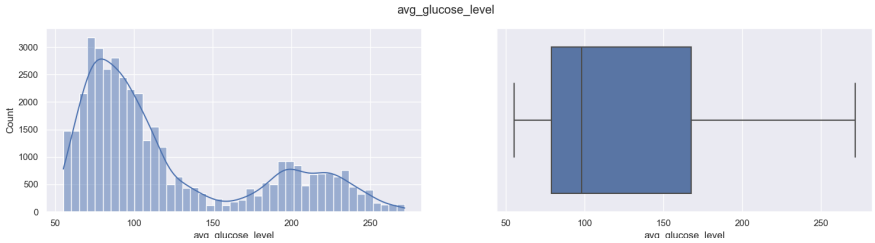


Figura 1: Distribución de los datos posterior a la imputación aplicada al atributo *sex*.

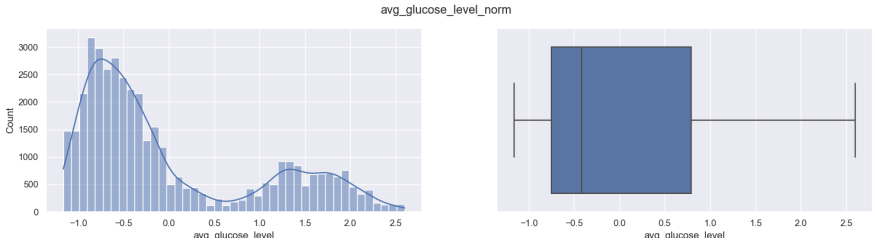
4.2.2. Normalización de datos

Se realizó un proceso de obtención de la distribución de los datos que conforman el conjunto de datos, usando histogramas y diagramas de cajas. Se pudo observar que para los atributos continuos se contaba con una distribución de tipo uniforme, por lo que se usó el método z-score para normalizar los datos. Para el resto de atributos, dada su naturaleza categórica y ordinal, se optó por no realizar algún tipo de normalización.

Las Figuras 2 y 3 muestran la una comparación entre la distribución de los atributos previa y posterior a normalizar.

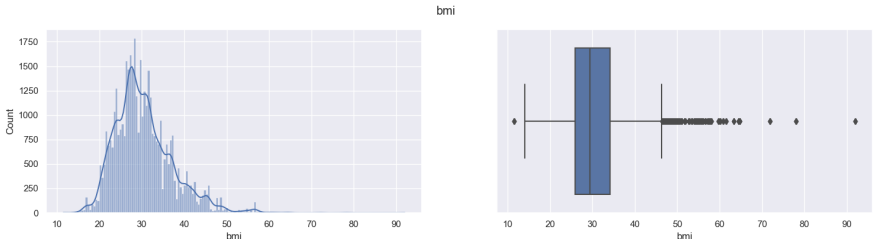


(a) Histograma de atributo *avg_glucose_level* sin normalizar.

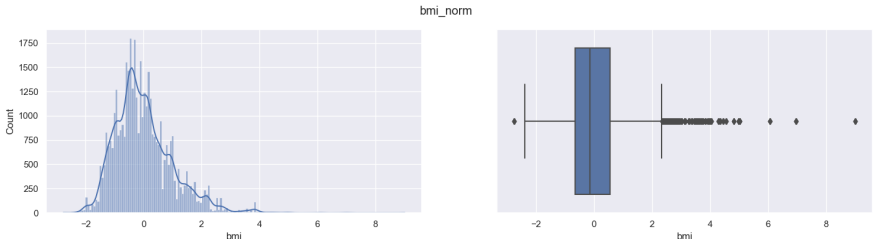


(b) Histograma de atributo *avg_glucose_level* posterior a normalización.

Figura 2: Normalización de atributo *avg_glucose_level*.



(a) Histograma de atributo *bmi* sin normalizar.



(b) Histograma de atributo *bmi* posterior a normalización.

Figura 3: Normalización de atributo *bmi*.

4.3. Selección de atributos

Con el fin de obtener una perspectiva general del desempeño del algoritmo KNN al usar diferentes tipos de datos, se optó por utilizar 3 métodos distintos para extracción de atributos.

4.3.1. Método Pearson

Utilizando la matriz de correlación de Pearson, se generó un mapa de calor para determinar los atributos más correlacionados con la variable objetivo. De este análisis se tomaron los 3 atributos con mayor índice de correlación.

En la Figura 4 se puede observar el mapa de calor obtenido tras en análisis de correlación de Pearson. Donde se puede concluir que los atributos con mayor correlación con la variable objetivo (*stroke*) son: *avg_glucose_level*, *hypertension* y *heart_disease*

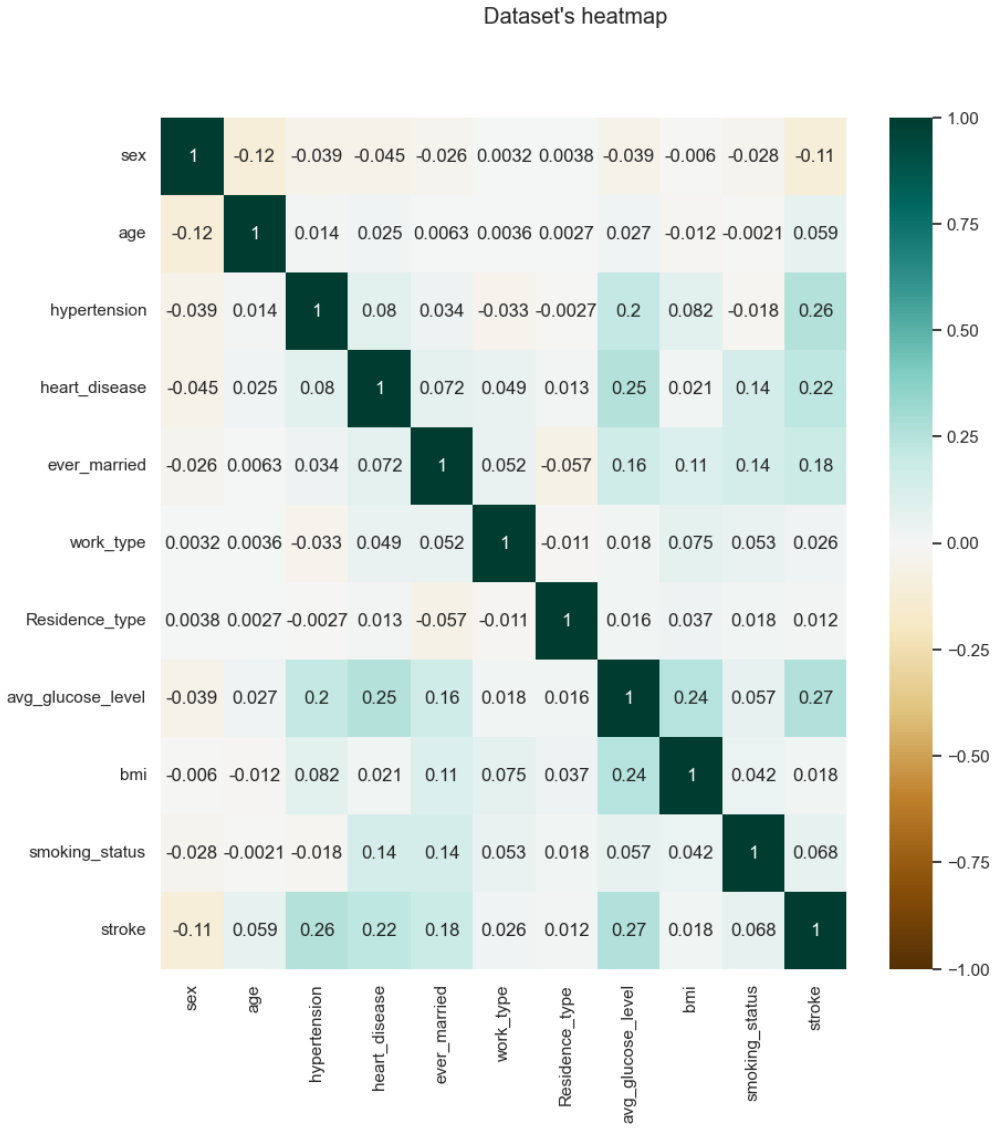


Figura 4: Mapa de calor obtenido tras el análisis de correlación de Pearson.

4.3.2. Método PCA

Utilizando el método de PCA para generar un subespacio que comprima la información global del conjunto de datos, se obtuvieron 3 componentes principales del conjunto de datos.

En la Tabla 1 se observa una muestra de 10 elementos de los 3 componentes principales resultantes del método PCA.

Tabla 1: Muestra de los 3 componentes principales generados por PCA.

PC1	PC2	PC3
-62.99	2.71	1.65
-41.99	8.22	2.87
-61.00	0.33	0.57
-40.99	0.64	0.80
-84.99	1.01	-0.03
-54.99	-0.46	1.00
-82.00	-0.82	0.43
-16.99	-0.12	1.12
-30.99	0.91	1.00
-55.00	-0.11	0.71

4.3.3. Selección experimental

Mediante el apoyo de una pairplot de la biblioteca *Seaborn*, se busca a los atributos que mejor logren separar visualmente a la variable objetivo. Dichos atributos se seleccionan para realizar el entrenamiento del modelo.

Para el caso de nuestro conjunto de datos, la Figura 5 muestra el pairplot resultante, donde se puede observar que los atributos que proporcionan una mejor separación de la variable objetivo son *age*, *avg_glucose_level* y *bmi*.

Figura 5: Pairplot obtenido del conjunto de datos.

4.4. Implementación de KNN

El Algoritmo 1 muestra el pseudocódigo utilizado para implementar el algoritmo KNN dentro del contexto de la práctica.

Algoritmo 1 Clasificador KNN

procedure KNN($X_{\text{train}}, y_{\text{train}}, X_{\text{test}}, k$)
    for instancia $\in X_{\text{test}}$ **do**
        Calcular la distancia entre la instancia actual y todas las instancias en X_{train}
        Ordenar las distancias de menor a mayor
        Tomar los k vecinos más cercanos
        Obtener las etiquetas correspondientes a los k vecinos más cercanos
        Realizar la votación mayoritaria para determinar la etiqueta predicha
        Asignar la etiqueta predicha a la instancia actual
    end for
    Retornar las etiquetas predichas para todas las instancias en X_{test}
end procedure

4.5. Evaluación de algoritmo

Se realizó un proceso de evaluación para cada uno de los conjuntos de datos generados por las técnicas de selección de características. Estas evaluaciones consistieron en la obtención de la matriz de confusión y las métricas exactitud, sensibilidad, precisión y puntaje F1. Dicho proceso de evaluación se realizó sobre el 30 % del conjunto de datos.

5. Resultados

5.1. Atributos seleccionados por Pearson

La Figura 6 muestra la nube de puntos resultante al graficar los atributos seleccionados mediante el algoritmo de Pearson. En esta representación gráfica se puede observar que los datos no presentan una segmentación notoria, y que, dado la naturaleza de los atributos *hypertension* y *heart_disease* se genera un conjunto de 4 pilares, los cuales corresponden a la combinación de los 2 atributos categóricos.

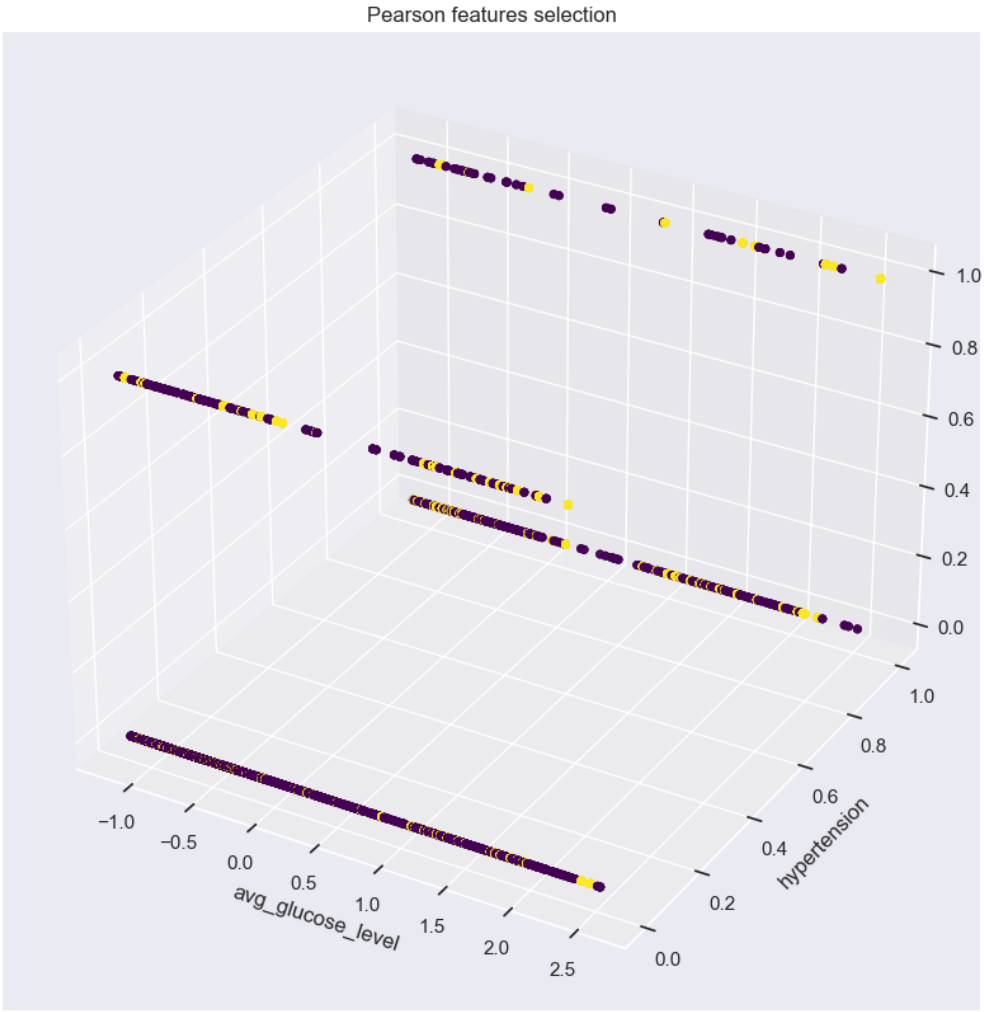


Figura 6: Atributos seleccionados por el algoritmo de Pearson.

5.2. Atributos seleccionados por PCA

La Figura 7 muestra la nube de puntos resultante de los 3 componentes principales obtenidos mediante PCA. Se puede observar un aislamiento de diversas regiones, sin embargo, también se logran observar puntos donde se mezclan las clases.

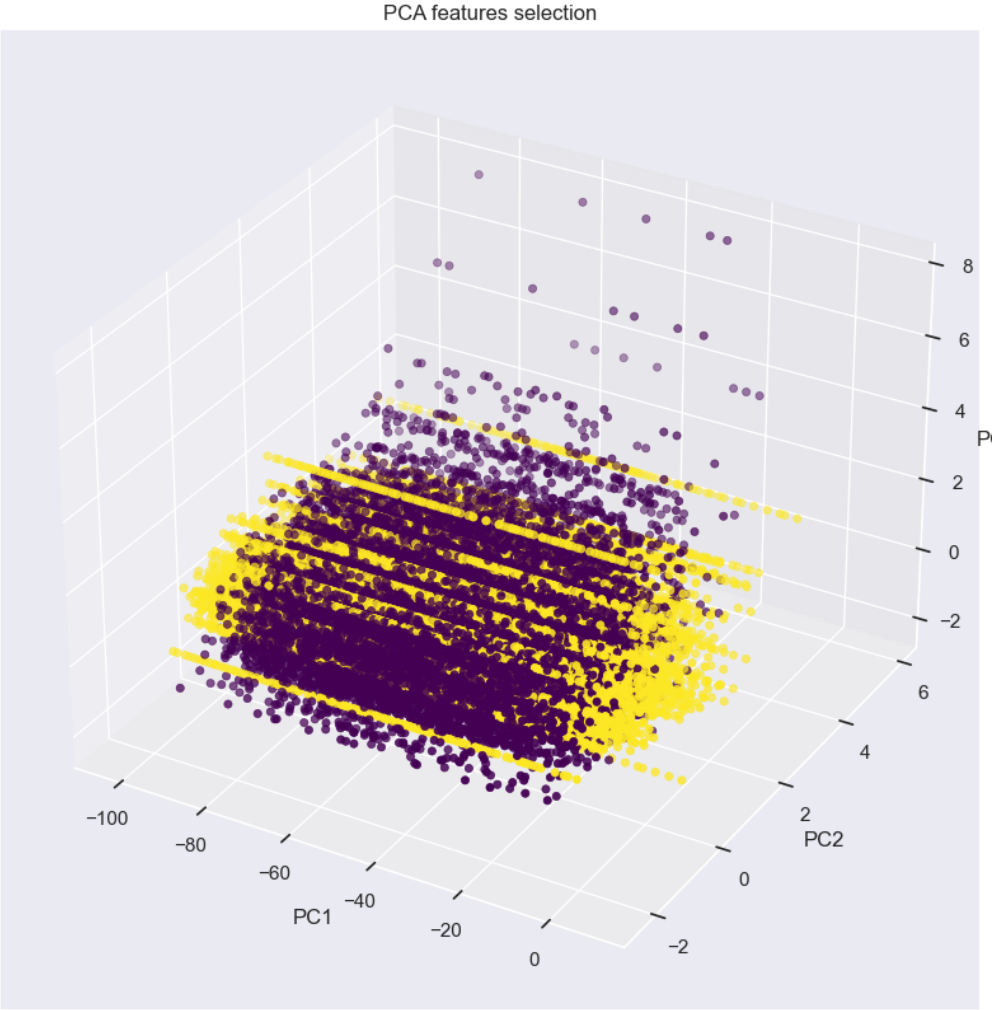


Figura 7: Atributos seleccionados por el algoritmo de PCA.

5.3. Atributos seleccionados por selección experimental

Relacionado al método experimental desarrollado para esta práctica, se determinó que los atributos *age*, *avg_glucose_level* y *bmi* serían los mejores. La Figura 8 muestra la nube de puntos resultante de dichos atributos, donde se puede observar que la naturaleza continua de los atributos genera una nube de puntos con áreas donde prevalece una determinada clase.

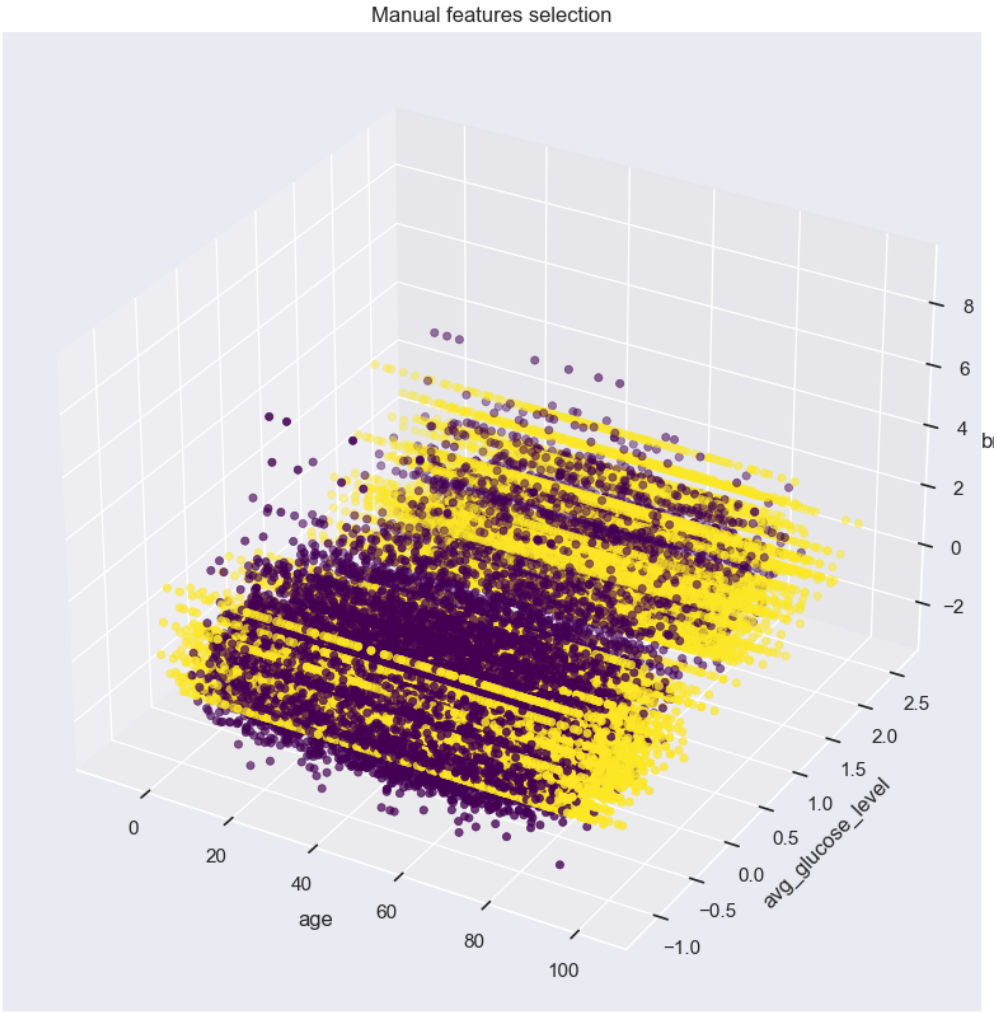


Figura 8: Atributos seleccionados por selección experimental.

5.4. KNN con atributos de Pearson

Los resultados obtenidos tras la evaluación del algoritmo utilizando los atributos seleccionados mediante el algoritmo de Pearson pueden observarse en la Figura 9 y en la Tabla 2

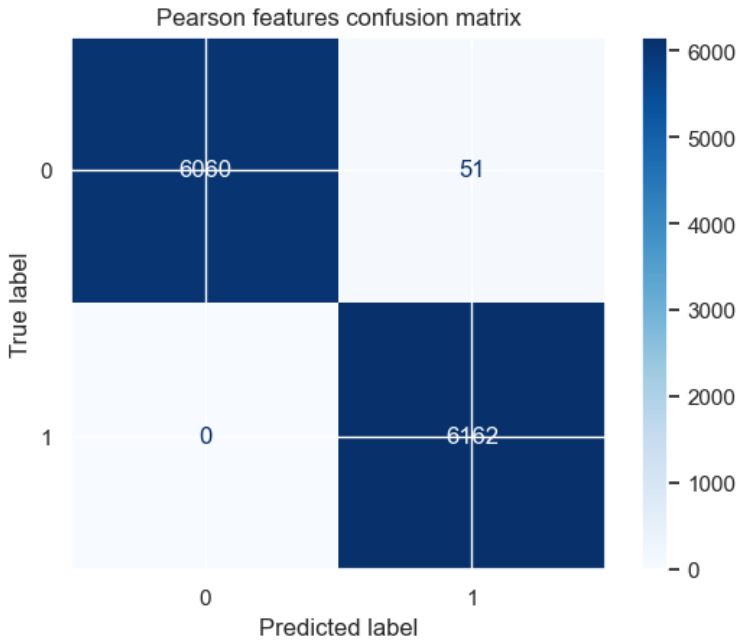


Figura 9: Matriz de confusión resultante tras la evaluación de KNN con atributos seleccionados por Pearson.

Tabla 2: Métricas de desempeño del algoritmo KNN con atributos seleccionados por Pearson.

Exactitud	Sensibilidad	Precisión	Puntaje F1
99.58 %	100 %	99.18 %	99.59 %

5.5. KNN con atributos de PCA

Los resultados obtenidos tras la evaluación del algoritmo utilizando los atributos seleccionados mediante el algoritmo de PCA pueden observarse en la Figura 10 y en la Tabla 3

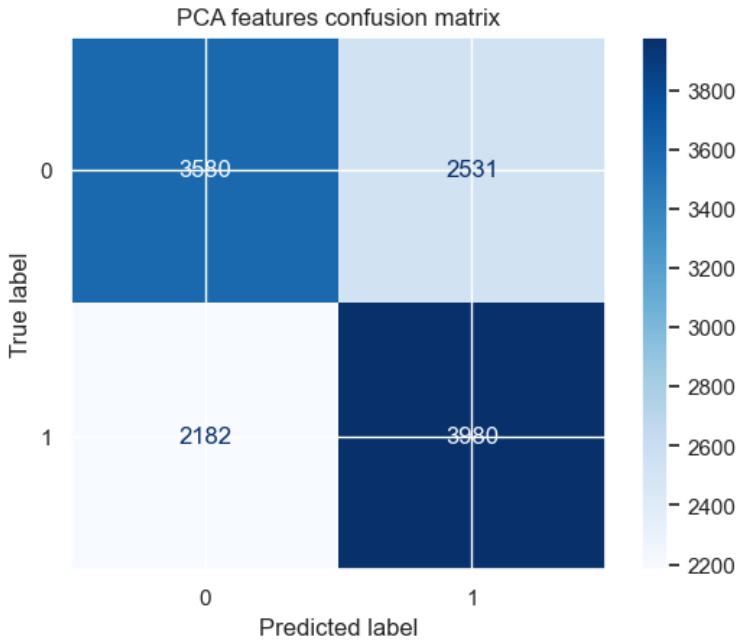


Figura 10: Matriz de confusión resultante tras la evaluación de KNN con atributos seleccionados por PCA.

Tabla 3: Métricas de desempeño del algoritmo KNN con atributos seleccionados por PCA.

Exactitud	Sensibilidad	Precisión	Puntaje F1
61.60 %	64.59 %	61.13 %	62.81 %

5.6. KNN con atributos de selección experimental

Los resultados obtenidos tras la evaluación del algoritmo utilizando los atributos seleccionados mediante la selección experimental pueden observarse en la Figura 11 y en la Tabla 4

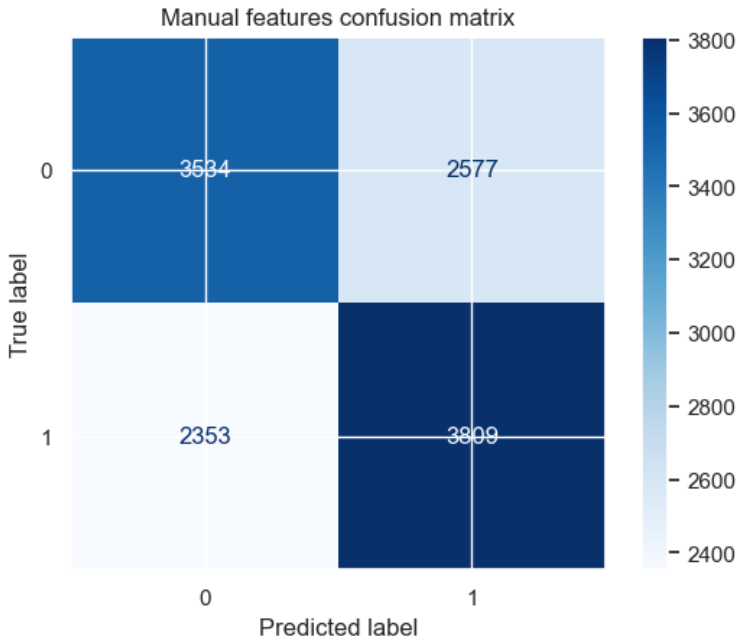


Figura 11: Matriz de confusión resultante tras la evaluación de KNN con atributos seleccionados mediante selección experimental.

Tabla 4: Métricas de desempeño del algoritmo KNN con atributos seleccionados de forma experimental.

Exactitud	Sensibilidad	Precisión	Puntaje F1
59.83 %	61.81 %	59.65 %	60.71 %

6. Conclusiones

Basado en el desempeño obtenido, se puede concluir que la técnica de selección de atributos utilizando el coeficiente de correlación de Pearson ha demostrado un rendimiento notablemente superior en comparación con las otras dos técnicas, PCA y la técnica experimental.

La técnica de Pearson logró una alta exactitud del 99.58 %, lo que indica que el modelo clasificador pudo predecir correctamente la clase de la mayoría de las instancias del conjunto de datos. Además, alcanzó una sensibilidad del 100 %, lo que significa que pudo identificar correctamente todos los casos positivos de stroke en el conjunto de datos. La precisión también fue alta, con un valor del 99.18 %, lo que implica que la mayoría de las instancias clasificadas como casos positivos fueron realmente positivos. El valor F1 de 99.59 % indica un buen equilibrio entre la precisión y la sensibilidad.

En contraste, las técnicas de PCA y la técnica experimental obtuvieron resultados inferiores. La técnica de PCA logró una exactitud del 61.60 %, lo que sugiere que su capacidad para clasificar las instancias correctamente fue significativamente menor. También tuvo una sensibilidad y precisión más bajas, lo que indica que el modelo fue menos efectivo para identificar los casos positivos de stroke y para clasificar correctamente las instancias. El valor F1 de 62.81 % también muestra un rendimiento moderado en términos de equilibrio entre precisión y sensibilidad.

Estos resultados sugieren que la técnica de selección de atributos basada en el coeficiente de correlación de Pearson puede ser una herramienta valiosa para la detección de stroke. La alta exactitud, sensibilidad y precisión alcanzadas por esta técnica indican que tiene el potencial de identificar de manera efectiva los casos de stroke y minimizar los falsos positivos y falsos negativos. Sin embargo, se necesita más análisis y validación con conjuntos de datos adicionales para confirmar la utilidad y generalidad de estos resultados.

A medida que la investigación en inteligencia artificial y aprendizaje automático avanza, es posible que se logren avances aún mayores en la detección temprana y precisa de stroke. Esto podría conducir a mejoras continuas en la atención médica de emergencia y a una mejora significativa en la calidad de vida de los pacientes que han sufrido un stroke.

Referencias bibliográficas

[1] M. A. Aceves Fernández, *Inteligencia artificial para programadores con prisa*. Universo de Letras, 2021, ISBN: 9788418854613.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, 2007, ISBN: 9780387310732.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2008, ISBN: 9780387848846.

[4] K. P. Murphy, *Machine learning : a probabilistic perspective*. MIT Press, 2013, ISBN: 9780262018029.

A. Código documentado

El código completo y funcional se puede encontrar anexo en el archivo *zip* compartido en conjunto con este reporte.