COLEGIO DE CIENCIAS E INGENIERIA

INGENIERIA INDUSTRIAL

IIN-3007 Analítica de Datos

NRC: 2209

PROYECTO: ATAQUES CEREBROVASCULARES

SEMESTRE: Segundo Semestre 2023-2024 (202320)

NOMBRE(S) Y CÓDIGO DE ESTUDIANTE(S):

Emilia Pavón 00323300, Belén Carrasco 00323335, Emily Alta 00324764

PROFESOR(A): María Gabriela Baldeón Calisto

FECHA DE ENTREGA: 24 de marzo de 2024

Introducción

Los ACV, también llamados ataques cerebrovasculares, son una de las causas principales de muerte en todo el mundo. Estos eventos tienen un impacto significativo en la salud pública, representando alrededor del 11% de todas las muertes anuales a nivel mundial. En este escenario, es de suma importancia desarrollar herramientas predictivas precisas que puedan identificar a las personas con mayor probabilidad de padecer un ACV. Esto resulta crucial para prevenir y tratar tempranamente esta enfermedad. El enfoque de este proyecto es la aplicación de técnicas de Machine Learning a una base de datos médica con el fin de desarrollar un modelo que pueda predecir la probabilidad de que un paciente sufra un ataque cerebral. Incluye un conjunto de variables predictivas vinculadas al estado sociodemográfico y médico de los pacientes, junto con una variable de respuesta que indica si el paciente ha experimentado un ACV o no.

Explicar todas las operaciones de preprocesamiento, EDA, y visualizaciones de las variables predictivas y discusión

En esta sección, se describirán las operaciones de preprocesamiento de datos y el análisis exploratorio de datos realizadas en el conjunto de datos con el objetivo de prepararlos para el modelado predictivo.

El preprocesamiento comenzó con la verificación de los tipos de datos de cada variable para asegurar su correcta interpretación. Se identificaron variables numéricas y categóricas, y se asignaron los tipos de datos correspondientes. El conjunto de datos inicial consistió en 5112 filas y 13 columnas. Se identificaron valores atípicos y datos faltantes. Para abordar los valores atípicos, se realizó una inspección detallada de cada variable. Se observó un dato atípico en la columna "smoking", que se modificó para reflejar una de las categorías existentes ("smokes"), ya que el paciente fumaba (CDC, 2021). Además, en la variable "Heart_Disease", se encontró una entrada inconsistente ("no"), que se transformó en 0 para representar una variable binaria.

En cuanto a la variable "work", se optó por aplicar una eliminación listwise para eliminar las filas que no correspondían a ninguna categoría específica (Bengtsson & Lindblad , 2020), asegurando así la integridad de los datos restantes. Para abordar los datos faltantes, se implementaron diferentes estrategias. Para los valores nulos en la variable "income", se utilizó la técnica de "median substitution" para llenar los espacios vacíos con la mediana de los ingresos. Para los valores atípicos en la variable "BMI", se los trató como datos faltantes (Aristhophane, Doffou, & Edoété, 2020) y se les aplicó la técnica de "regression imputation" para estimar valores apropiados basados en el resto de la información disponible.

En cuanto a la visualización de datos pudimos apreciar la representación de las variables, en general se generaron gráficas de barras, gráficos de pie, histogramas, etc. De esta manera pudimos determinar diferentes patrones dentro de los datos (Graficas adjuntas en el Anexo 1). La grafica más relevante fue la de distribución de la variable de respuesta “Stroke” (imagen 1) la cual nos muestra que esta base de datos se encuentra desbalanceada.

Imagen 1: Distribución de datos con diagnóstico de Stroke.

Explicar la correlación

Para explicar la correlación entre las variables predictivas y la variable de respuesta en primer lugar realizamos una matriz de correlación (Anexo 2), de esta manera, pudimos visualizar como las diferentes variables afectaban nuestra variable de respuesta, de esta manera, realizamos el siguiente gráfico:

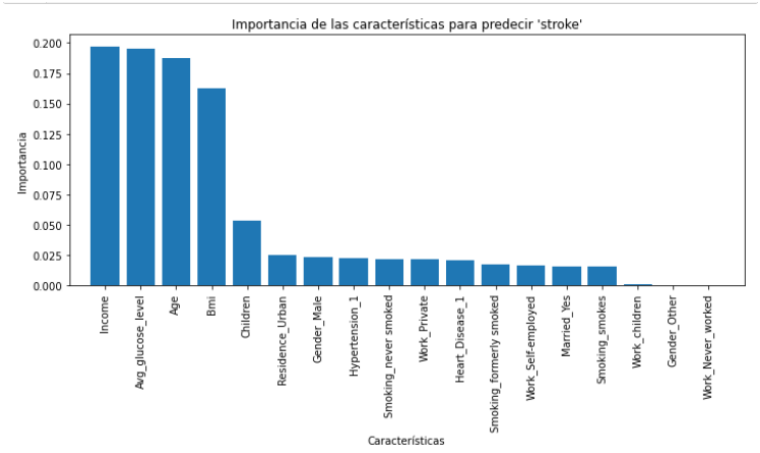


Imagen 2: Importancia de variables predictivas para predecir “Stroke”

Al analizar la gráfica pudimos observar que las variables que mayor efecto tienen en nuestra variable predictiva son: “income”, “avg_glucose_level”, “age” y “Bmi” estas están altamente relacionadas con la salud del paciente.

Investigación sobre bases de datos no-balanceadas. Incluir referencias pertinentes en el texto.

Los datos desbalanceados ocurren cuando la cantidad de muestras de una clase es mucho más grande que la de otra (Ramyachitra & Manikandan, 2014). En otras palabras, es cuando hay mucha información acerca de un hecho en específico y muy poca acerca de un hecho contrario. Esto se conoce como clase mayoritaria y clase minoritaria. Este balance se presenta mayormente en casos de la vida real, ya sea en el campo económico (fraudes en bancos), médico, comercial, entre otros.

En general, poseer una base de datos desbalanceada genera inconsistencias pues el modelo se entrena mejor para la clase con muchos datos, mientras que no es tan robusto para la clase minoritaria, lo que impide la generalización. Esto también se traduce en la disminución del desempeño del set de entrenamiento (Ramyachitra & Manikandan, 2014). De igual forma, este desbalance se puede ver aún más afectado por la “noisy data” dado que tiene un mayor impacto en la clase minoritaria (Ramyachitra & Manikandan, 2014). En este sentido, esta clase igual es más sensible a errores de clasificación singulares (Ramyachitra & Manikandan, 2014). Para

solucionar la imbalanced data existen varios algoritmos, pero en el presente proyecto se utilizó el undersampling y el oversampling.

Prueba estadística para escoger la mejor estrategia de entrenamiento que reduce los problemas de bases de datos no balanceados y el análisis de los resultados.

Se realizó una prueba de Wilcoxon pareada utilizando la accuracy como métrica de interés pues se encarga de medir lo datos predichos correctamente y eso es lo que busca el proyecto. Se seleccionó a esta prueba porque las métricas de rendimiento provienen de la misma muestra y no se requiere suposiciones sobre la distribución de los datos. Las hipótesis planteadas son:

- Ho: No hay diferencia significativa entre el oversampling y el undersampling en términos de la métrica de accuracy.
- Ha: Hay una diferencia significativa entre el oversampling y el undersampling en términos de la métrica de accuracy.

Se seleccionó un alfa de 0.05 ya que se utilizó como referencia el paper titulado "An application of machine learning to haematological diagnosis", donde igualmente se empleó este valor de alfa ya que se considera el más apropiado y utilizado en los problemas del sector médico (Gunčar et al., 2018). Se llegó a la conclusión de que no se rechaza Ho, por lo que no hay suficiente evidencia significativa de que los métodos de oversampling y undersampling son diferentes, es decir, no se puede determinar que uno es mejor que otro. Sin embargo, se seleccionó el undersampling ya que tiene un costo computacional menor.

División de la base de la base de datos con su respectiva referencia.

La partición de la base de datos se conoce por ser una práctica fundamental en el Machine Learning puesto que permite evaluar objetivamente el rendimiento del modelo para evitar el sobreajuste, desarrollar modelos robustos y generalizables (Bangert, 2021). Al revisar diferentes estudios se pudo determinar que la mejor opción es dividir la base de datos en 75% para el set de entrenamiento ya que permite una mayor cantidad de información para aprender, 10% para el set de validación puesto que tendrá una evaluación más precisa en su rendimiento y 15% para el set de prueba para una estimación más robusta del rendimiento final en datos no observados (Camacho et al., 2024) así mitigando los efectos negativos de un conjunto desequilibrado. Teniendo en cuenta diferentes aspectos como la cantidad de datos que están siendo entrenados, también al revisar otro estudio, no hay diferencia significativa al ocupar una partición mayor (Ashraf et al., 2023).

Mencionar y justificar los 4 algoritmos seleccionados. Explicar en qué modelos e hiperparámetros se utiliza un modelo de optimización. Debidamente justificar los valores de los otros hiperparámetros. Explicar que ensamble se utiliza y el nuevo modelo implementado. Incluir una descripción del nuevo modelo.

- **Regresion logística**

  Para el primer algoritmo, utilizamos hiper parámetro como penalty = "None" al igual que solver = "newton-cg", no obstante, no se realizó una optimización de hiper parámetros.

La elección de "penalty=None" se justifica debido a que se desea explorar el modelo sin imponer restricciones adicionales a los coeficientes, permitiendo así que el modelo se ajuste libremente a los datos. Esto puede ser útil debido a que se sospecha que la relación entre las variables predictoras y la variable objetivo es compleja y no lineal, y se prefiere dejar que el modelo aprenda directamente de los datos. (Wang, Xiong, Liu, Wang, & Li, 2024)

El solver "newton-cg" es apropiado cuando no se aplica penalización a los coeficientes. “utiliza el algoritmo de gradiente conjugado no lineal de Polak y Ribiere para calcular la dirección de búsqueda. Este método es adecuado para problemas de gran escala.” Por lo que el solver es apropiado bajo el contexto del problema. (Martin-Baos, Garcia-Rodenas, & Rodriguez-Benitez, 2021)

- **KNN**

Para el algoritmo de clasificación K-Nearest Neighbors (KNN) se realizó optimización de hiperparámetros utilizando GridSearchCV. Los hiperparámetros específicos y sus valores que se están explorando son (Kilicarslan, Diker, Kozkurt, & Donmez, 2024):

  - n_neighbors: Determina el número de vecinos más cercanos para la clasificación. Se está explorando un rango de valores de 1 a 84 para este parámetro.
  - weights: Determina cómo se ponderan los vecinos en función de su distancia. Los valores posibles son 'uniform', donde todos los vecinos tienen el mismo peso, y 'distance', donde los vecinos más cercanos tienen más peso que los vecinos más lejanos.
  - algorithm: Determina el algoritmo utilizado para calcular los vecinos más cercanos. Los valores posibles son 'auto', 'ball_tree', 'kd_tree' y 'brute'.
  - leaf_size: Controla el tamaño de las hojas en los árboles de búsqueda (ball_tree o kd_tree) o no tiene ningún efecto en los otros algoritmos. Se están explorando tres valores: 10, 20 y 30.
  - metric: Determina la métrica de distancia utilizada para calcular la distancia entre las observaciones. Los valores posibles son 'euclidean' y 'manhattan'.

La búsqueda de hiperparámetros se realiza utilizando validación cruzada con 5 folds y se utiliza la métrica de precisión para evaluar el rendimiento de cada combinación.

La elección de estos valores y parámetros es apropiada para explorar un amplio rango de configuraciones posibles del modelo KNN y encontrar la combinación óptima. La validación cruzada garantiza que el modelo generalice bien a datos no vistos. (Siva, Gupta, & Borah, 2023)

- **Random Forest**

Para el algoritmo de random forest estamos utilizando un método de ensamble al igual que realizamos una optimización de hiper parámetros mediante BayesianCV, para los mismos estamos optimizando los siguiente hiper parámetros (Saxena, 2023):

  - n_estimators: Determina el número de árboles en el bosque. Entre un rango de valores entre 10 y 200.
  - max_depth: Determina la profundidad máxima de cada árbol en el bosque. Se está buscando en un rango de valores entre 2 y 10.

- min_samples_split: Determina el número mínimo de muestras requeridas para dividir un nodo interno. Se está buscando en un rango de valores entre 2 y 20.
- min_samples_leaf: Determina el número mínimo de muestras requeridas para estar en un nodo hoja. Se está buscando en un rango de valores entre 1 y 10.

En este caso existen dos hiper parametros que no se estan optimizando "max_terminal_nodes" y "max_sample":

- max_terminal_nodes: Establece una condición para la división de los nodos del árbol y, por tanto, restringe el crecimiento del árbol. (Saxena, 2023). Establecer este límite puede ser útil para prevenir el sobreajuste. Sin embargo, en este caso, se está optimizando el hiperparámetro "max_depth", que controla la profundidad máxima de los árboles. Al controlar la profundidad máxima, se limita automáticamente el número de nodos terminales.
- max_samples: Este hiperparámetro determina el número máximo de muestras que se utilizarán para entrenar cada árbol individual en el bosque. Sin embargo, en este caso, no se está optimizando "max_samples" porque el modelo de Random Forest ya implementa una técnica de muestreo aleatorio llamada Bagging, donde cada árbol se entrena con una muestra bootstrap del conjunto de datos de entrenamiento original.

El número de iteraciones para la búsqueda se establece en 50. Se utiliza una validación cruzada estratificada con 5 splits. Este enfoque de optimización de hiper parámetros busca encontrar la combinación óptima que maximiza el rendimiento del modelo en los datos de entrenamiento y generaliza bien a datos no vistos.

- **Xgboost**

En este caso hemos decidido implementar el algoritmo Xgoboost, el cual no ha sido revisado en clases, para comprender de mejor manera este algoritmo "XGBoost es un algoritmo de impulso que utiliza "bagging", entrena múltiples árboles de decisión y luego combina los resultados. Permite que XGBoost aprenda más rápidamente, pero también le brinda una ventaja en situaciones con muchas características a considerar." (Simplilearn, 2023)

Para este caso se están utilizando los valores predeterminados de Xgoboost para entrenar al algoritmo los cuales son (Mendoza, 2020):

- n_estimators: 100 (número predeterminado de árboles en el ensamble.)
- max_depth: 6 (Profundidad máxima predeterminada de cada árbol.)
- learning_rate: 0.3 (controla la contribución de cada árbol al modelo final.)
- min_child_weight: 1 (El peso mínimo predeterminado requerido de las instancias hijo (nodos hoja).)
- gamma: 0 (controla cuándo se realizará una división en un nodo del árbol.)

Incluir la matriz de confusión de cada algoritmo y curva ROC. Resumir mediante una tabla las métricas de evaluación de los algoritmos en el set de entrenamiento, validación y

prueba. Explicar y justificar la selección de la métrica de evaluación y el mejor algoritmo. Analizar por cada algoritmo si está sobreajustado, ajustado deficiente balanceado.

Regresión Categórica:

Entrenamiento

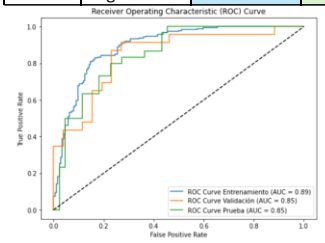
		Predicción	
		Positivos	Negativos
Obs	Positivos	137	39
	Negativos	30	163

Validación

		Predicción	
		Positivos	Negativos
Obs	Positivos	20	6
	Negativos	5	18

Test

		Predicción	
		Positivos	Negativos
Obs	Positivos	34	10
	Negativos	7	23



KNN:

Entrenamiento

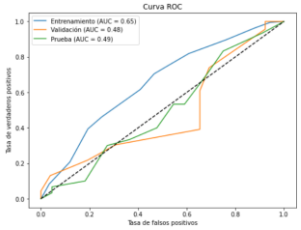
		Predicción	
		Positivos	Negativos
Obs	Positivos	94	182
	Negativos	57	136

Validación

		Predicción	
		Positivos	Negativos
Obs	Positivos	9	17
	Negativos	11	12

Test

		Predicción	
		Positivos	Negativos
Obs	Positivos	20	24
	Negativos	14	16



Random Forest:

Entrenamiento

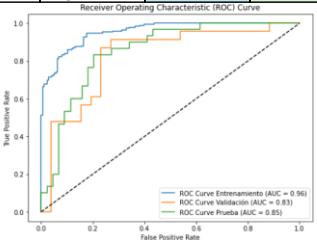
		Predicción	
		Positivos	Negativos
Obs	Positivos	146	30
	Negativos	14	179

Validación

		Predicción	
		Positivos	Negativos
Obs	Positivos	20	6
	Negativos	5	18

Test

		Predicción	
		Positivos	Negativos
Obs	Positivos	32	12
	Negativos	5	25



Xgboost:

Entrenamiento

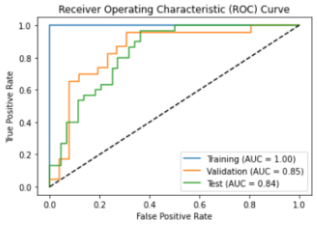
		Predicción	
		Positivos	Negativos
Obs	Positivos	176	0
	Negativos	0	193

Test

		Predicción	
		Positivos	Negativos
Obs	Positivos	32	12
	Negativos	6	24

Validación

		Predicción	
		Positivos	Negativos
Obs	Positivos	20	6
	Negativos	5	18



Tablas de métricas por cada set:

Exactitud

Modelo	Entrenamiento	Validación	Test
Regresión Categórica	0.813	0.776	0.77
KNN	0.623	0.429	0.486
Random Forest	0.88	0.776	0.77
Xgboost	1	0.776	0.757

Especificidad

Modelo	Entrenamiento	Validación	Test
Regresión Categórica	0.778	0.769	0.772
KNN	0.534	0.346	0.455
Random Forest	0.829	0.769	0.728
Xgboost	1	0.767	0.728

Precisión

Modelo	Entrenamiento	Validación	Test
Regresión Categórica	0.807	0.75	0.697
KNN	0.624	0.414	0.4
Random Forest	0.856	0.75	0.676
Xgboost	1	0.75	0.667

AUC

Modelo	Entrenamiento	Validación	Test
Regresión Categórica	0.894	0.846	0.855
KNN	0.652	0.479	0.494
Random Forest	0.956	0.826	0.53
Xgboost	1	0.51	0.836

Sensibilidad

Modelo	Entrenamiento	Validación	Test
Regresión Categórica	0.845	0.783	0.767
KNN	0.705	0.522	0.533
Random Forest	0.927	0.783	0.833
Xgboost	1	0.783	0.8

Para el contexto médico del proyecto la métrica de evaluación que se seleccionó es la de sensibilidad (recall) ya que su objetivo es minimizar los falsos negativos (EvidentlyAI, s. f). Es decir, se desea evitar clasificar a una persona como que no va a tener un ataque cerebro vascular cuando en realidad es muy probable que sí. El cometer este error de clasificación podría ser muy perjudicial para el paciente. El mejor algoritmo con esta métrica es el de Random Forest ya que posee los valores más altos en 2 de los 3 sets. Es importante notar que el set de entrenamiento, el Xgboost tiene valores mayores que el Random Forest, pero estos valores son 1, lo que indica que hay sobreajuste y esto genera un sesgo y errores al predecir con nuevos datos. De esta manera, no es el mejor algoritmo y por eso no fue seleccionado.

<u>Análisis de overfitting o underfitting por cada algoritmo:</u>

- Regresión categórica: está balanceado, los valores de las métricas para cada set no son ni muy altos ni muy bajos.

- KNN: subajustado porque los valores de la mayoría de métricas son bajos.
- Random Forest: está balanceado, los valores de las métricas para cada set no son ni muy altos ni muy bajos.
- Xgboost: este es el algoritmo que está más sobreajustado ya que los valores de todas sus métricas en el set de entrenamiento son 1 lo que quiere decir que aprendió muy bien los datos de este set.

Análisis de las variables predictivas más importantes para la predicción. Explicar el método basado en random forest utilizado para obtener estos resultados. Comparar los resultados con el análisis de correlación.

El método basado en Random Forest se forma mediante un conjunto de árboles de decisión individuales. Cada uno de estos árboles se entrena con una muestra aleatoria extraída de los datos de entrenamiento originales mediante el método de bootstrapping, lo que significa que cada árbol se entrena con un conjunto de datos ligeramente diferente (Ciencia de datos, s. f.). En cada árbol, las observaciones se distribuyen a través de nodos de bifurcación, configurando la estructura del árbol hasta llegar a un nodo terminal (Ciencia de datos, s. f.). La predicción de una nueva observación se logra combinando las predicciones de todos los árboles individuales que conforman el modelo. De esta manera, la variable de predicción que se encuentra en el nodo raíz es la más importante, y los nodos de decisión que parten de este son las siguientes variables más importantes. Mientras más cerca esté una variable del nodo de raíz, más importante es.

Utilizando este método en Python, se obtuvo que las variables predictivas más importantes son: Age, nivel promedio de glucosa, bmi e income. Estas variables coinciden con aquellas que se obtuvieron en la correlación, la única diferencia es el orden de importancia. Esto se debe a que la correlación se hizo con los datos desbalanceados, mientras que el Random Forest se hizo con los datos del undersampling.

Costos de implementación del modelo. Ventajas y desventajas de utilizar el modelo propuesto en el hospital.

Para calcular el costo de implementar un modelo se debe tener en cuenta aspectos necesarios dependiendo del alcance, la complejidad y el tamaño de la empresa, algunos son: infraestructura, recursos humanos, herramientas de preparación, los costos de mantenimiento y actualización (Eby, 2017).

En cuanto al personal especializado como son los analistas de datos o científicos de datos se estima un sueldo promedio estadounidense de $152.013 al año con un mínimo de $77.000 (Indeed, 2024), por otro lado, están las capacitaciones al personal que es de $2.000 a $10.000. Algunos precios en cuanto al software son gracias a Azure, en servidores y almacenamiento 1 TB/mes es de $17.013 a $50.000, actualizaciones, soporte y protección de datos es de $25.000 anual, análisis de datos y visualización $5.000 a $20.000, por otra parte, el hardware como son computadores y estaciones de trabajo es de $800 a $2.000.

En total implementar el modelo en un hospital es de $809.012 tomando todos los valores más altos, sin embargo, se realizó también una cotización en Azure con un resultado para todos los componentes de software en TB y el resultado fue de $20.458,62 mensual.

Algunas ventajas de utilizar el Random Forest en el ámbito médico es que tiene alta precisión, una buena robustez, una fácil interpretación, versatilidad para aplicarse de distintas maneras y eficiencia computacional al utilizar menor tiempo de ejecución (Khalilia et al., 2011). Igualmente existen desventajas que pueden ser contraproducentes al momento de predecir, como es la complejidad por lo que tendrán que invertir, el sobreajuste que puede sufrir el modelo, dependencia de datos, falta de estandarización.

Enlace al repositorio de GitHub.

https://github.com/em137942/AD-Proyecto-PredecirAC

Conclusiones

EDA nos permite resumir las características del conjunto de datos mediante visualizaciones, detectar patrones y anomalías para comprender y preparar los datos. También la limpieza de datos asegura que los resultados no se vean alterados y genere datos coherentes.

Tener una base de datos balanceada permite organizar y relacionar datos de manera congruente para tener un mejor rendimiento en el set de entrenamiento. Al dividir la base de datos se asegura que tenga un buen desempeño el modelo, sea robusto, confiable y aplicable en otras situaciones reales.

Realizar una prueba estadística en los métodos que se debe ocupar para balancear los datos es esencial para conocer si hay diferencia significativa y cual método escoger, se concluyó que la técnica de undersampling mejora la predicción puesto que tiene un mayor porcentaje en las métricas estadísticas de exactitud y precisión, esta nos ayuda para tener un enfoque en las respuestas de interés.

La elección del algoritmo depende de la base de datos, la complejidad del problema y los recursos computacionales. La regresión logística es buena para problemas de clasificación binaria, mientras que KNN es útil para datos estructurales no lineales. Por otra parte, Random Forest y XGBoost son opciones sólidas para conjuntos de datos grandes y complejos puesto que manejan de mejor manera el sobreajuste y la precisión.

Utilizar diferentes algoritmos ayuda a comparar los diferentes resultados y observar cual es el mejor o más optimo dependiendo de lo que se necesita, en cuanto al estudio como es en base a información clínica es importante que no haya fallas o equivocaciones ya que son vidas humanas, por lo que la sensibilidad es importante.

Al realizar un cálculo estimado de los costos de implementar modelos de analítica de datos en un hospital es de $809 mil, como se puede observar parece un valor alto, sin embargo, a largo plazo nos da más beneficios, por lo que se debe tomar en cuenta.

Referencias

Aristophane, K., Doffou, J., & Edoété, P. (2020). *Method for Automatically Processing Outliers of a Quantitative Variable* . Obtenido de International Journal of Advanced Computer Science and Applications,: https://thesai.org/Downloads/Volume11No7/Paper_53-Method_for_Automatically_Processing_Outliers.pdf

Ashraf, M., Abrar, M., Qadeer, N., Alshdadi, A. A., Sabbah, T., & Khan, M. A. (2023). A Convolutional Neural Network Model for Wheat Crop Disease Prediction. *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, *75*(2), 3867-3882. https://doi.org/10.32604/cmc.2023.035498

Bangert, P. (2021). Machine learning. En *Elsevier eBooks* (pp. 37-67). https://doi.org/10.1016/b978-0-12-820714-7.00003-0

Bengtsson, F., & Lindblad , K. (2020). *Methods for handling missing values*. Obtenido de UPPSALA UNIVERSITET: https://www.diva-portal.org/smash/get/diva2:1520218/FULLTEXT01.pdf

Camacho, M., Wilms, M., Almgren, H., Amador, K., Camicioli, R., Ismail, Z., ... & Alzheimer's Disease Neuroimaging Initiative. (2024). Exploiting macro-and micro-structural brain changes for improved Parkinson's disease classification from MRI data. *npj Parkinson's Disease*, *10*(1), 43.

CDC. (17 de Noviembre de 2021). *Salud de los pulmones*. Obtenido de Centros para el control y la prevencion de enfermedades: https://www.cdc.gov/marijuana/health-effects/es/lung-health.html

Ciencia de datos. (s. f.). *Random Forest python*. Recuperado 21 de abril de 2024, de https://cienciadedatos.net/documentos/py08_random_forest_python

Eby, K. (2017). Guía definitiva para estimar los costos del proyecto. *Smartsheet*. https://es.smartsheet.com/ultimate-guide-project-cost-estimating

EvidentlyAI. (s. f.). Accuracy vs. precision vs. recall in machine learning: what's the difference? Recuperado 31 de marzo de 2024, de https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~

Gunčar, G., Kukar, M., Notar, M., Brvar, M., Černelč, P., Notar, M., & Notar, M. (2018). An application of machine learning to haematological diagnosis. *Scientific reports*, *8*(1), 411.

Humphries, M. (2015). *Missing Data & How to Deal: An overview of missing data.* Obtenido de University of Texas: https://minio.la.utexas.edu/webeditor-files/prc/pdf/missing-data.pdf

Imbalanced Learn. (s. f.). Over-sampling. Recuperado 31 de marzo de 2024, de https://imbalanced-learn.org/stable/over_sampling.html

Imbalanced Learn. (s. f.). Under-sampling. Recuperado 31 de marzo de 2024, de https://imbalanced-learn.org/stable/under_sampling.html

Indeed. (2024). *Sueldos por año de Data scientist en Indeed en Estados Unidos*. https://www.indeed.com/cmp/Indeed/salaries/Data-scientist

InteractiveChaos. (03 de Diciembre de 2018). *Comprobación del tipo de las columnas de un dataframe*. Obtenido de Interactive Chaos: https://interactivechaos.com/es/python/scenario/comprobacion-del-tipo-de-las-columnas-de-un-dataframe

Interactivechaos. (16 de Enero de 2021). *numpy.nan*. Obtenido de Interactivechaos: https://interactivechaos.com/es/python/function/numpynan

Khalilia, M., Chakraborty, S. & Popescu, M. Predicting disease risks from highly imbalanced data using random forest. *BMC Med Inform Decis Mak* **11**, 51 (2011). https://doi.org/10.1186/1472-6947-11-51

Kocak, B., Kus, E. A., & Kilickesmez, O. (2021). How to read and review papers on machine learning and artificial intelligence in radiology: a survival guide to key methodological concepts. *European Radiology*, *31*, 1819-1830.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, *30*(1), 25-36.

Microsoft. (2024). *Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure*. Azure. https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-azure/

Mountassir, A., Benbrahim, H., & Berrada, I. (2012, October). An empirical study to address the problem of unbalanced data sets in sentiment classification. In *2012 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 3298-3303). IEEE.

Prashant. (2019, 23 diciembre). *Random Forest Classifier + Feature importance*. Kaggle. https://www.kaggle.com/code/prashant111/random-forest-classifier-feature-importance

PyPI. (2024, 19 febrero). *prettytable*. Recuperado 22 de abril de 2024, de https://pypi.org/project/prettytable/

Ramyachitra, D., & Manikandan, P. (2014). Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, *5*(4), 1-29.

Saini, M. (2023, 20 junio). Understanding Performance Metrics in Machine Learning: Precision, Recall, F1 Score, Confusion Matrix, and AUC ROC with an example. https://www.linkedin.com/pulse/understanding-performance-metrics-machine-learning-precision-saini#:~

Scipy. (s. f.). *Scipy.Stats.Wilcoxon*. Recuperado 22 de abril de 2024, de https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html

Valenzuela-Nunez, C. I., Espinosa, F. H. T., & Latorre-Núñez, G. (2023). Prediction of absenteeism in medical appointments using Machine Learning. *Universidad, Ciencia y Tecnología/Universidad, Ciencia y Tecnología*, *27*(120), 19-30. https://doi.org/10.47460/uct.v27i120.728