Predicción de Ocurrencia de Accidentes Cerebrovasculares

Seminario

Entregable II

A. J. Espinal
Especialización en Analítica y Ciencia de Datos
Departamento de Ingeniería de Sistemas
Universidad de Antioquia, Colombia
https://github.com/AndresEspinal/monografia-stroke
ajulian.espinal@udea.edu.co

Resumen— En kaggle se encuentra la base de datos llamada Stroke Prediction Dataset [1] que busca predecir según un número de variables si el paciente puede sufrir o no de un accidente cerebrovascular. Aquí se utilizaron herramientas de machine learning con la intención de lograr predecir en un alto porcentaje si se puede sufrir o no de esta condición médica y pueda ser de ayuda para futuros casos médicos que cumplan con esta condición.

En este proyecto se abordaron varias técnicas referentes a modelos supervisados como la regresión logística, knn, máquinas de soporte de vectores, random forest, árboles de decisiones, entre otros. Se usaron metodologías de validación como la división de datos de entrenamiento y prueba y la validación cruzada, además se usó el balanceo de clases de la variable respuesta debido a una considerable diferencia de registros positivos y negativos. Los resultados obtenidos dieron como mejor medida por ahora al modelo KNN con un recall del 0.947 y precisión del 0.735.

Índice de términos— Aprendizaje supervisado, Condición médica (stroke), Machine learning, Matriz de confusión.

I. Recursos

La elaboración de este proyecto usó en gran parte los recursos de estudio proporcionados por las clases vistas en la especialización de Analítica y Ciencia de Datos de la Universidad de Antioquia, además de asesorías virtuales con el profesor Yony Fernando Ceballos el cual es el asesor de la monografía.

II. Entregable I

A. *Comprensión del problema de aprendizaje automático*

1) Descripción del problema:

Stroke Prediction Dataset es una base de datos que se encuentra en Kaggle, sitio de búsqueda de muchos científicos de datos y aspirantes a científicos de datos que busca crear un entorno en el cual se puedan resolver problemas allí publicados. La base de datos anteriormente nombrada busca predecir si es probable que un paciente sufra de un accidente cerebrovascular dado unas variables de salud como tabaquismo, índice de masa corporal, sexo, edad, informaciones sobre otras enfermedades, entre otras. Para darle una idea al lector acerca de esta condición, explicaremos brevemente de qué se trata.

Según la OMS, los accidentes cerebrovasculares ocurren cuando se impide que la sangre fluya hacia el cerebro debido a una obstrucción, ya sea por depósitos de grasa, coágulos de sangre y hemorragias de los vasos cerebrales y sanguíneos que irrigan el cerebro **[2].** Además de ello, este problema es relevante ya que estamos hablando de una condición médica que está ubicada en el segundo puesto de causas de defunciones a nivel mundial **[3]**.

Aunque solo hablamos de una base datos, está puede ser de gran utilidad al momento de saber si el estilo de vida de la persona, las condiciones médicas y datos generales como edad y sexo pueden generar una propensión a sufrir este padecimiento.

El dataset cuenta con una variable de salida conformada por ceros y unos, lo que indica que se está tratando un modelo de aprendizaje supervisado de clasificación donde se busca que las predicciones digan 1 si es susceptible de padecer esta condición o cero si no es propenso a padecerlo.

2) Comparaciones:

Para hacer el ejercicio un poco más riguroso, es importante analizar y conocer si la base datos ya ha sido trabajada anteriormente con el fin de comparar resultados y análisis diferentes. Esto ayuda mucho a enriquecer los conocimientos y las maneras de abordar un mismo tema, por consiguiente, se analizaron 4 proyectos referentes a la misma base de datos que podrán ser consultados en la bibliografía de este entregable y explicaremos brevemente sobre las técnicas de aprendizaje utilizadas, metodología de validación y resultados obtenidos.

Las siguientes muestras fueron comparadas en el mismo sitio de kaggle, más específicamente en la misma base de datos.

En primer lugar, abordaremos la solución que ofrece Thomas Konstantin **[4]**. Allí destacamos lo siguiente:

- Los modelos de aprendizaje usados fueron los árboles de decisiones, el Random Forest, las Máquinas de Soporte Vectorial y la Regresión Logística
- La metodología de validación usada fue la validación cruzada.
- Después de probar el modelo con la validación cruzada, el autor identifica que el modelo random forest alcanza un F1 del 95% aproximadamente, sin embargo, sugiere que se deben hacer análisis adicionales para evitar algún tipo de azar y problemas que hayan surgido en la etapa de modelado.

Siguiendo con la búsqueda de soluciones de la base de datos se encuentra la solución propuesta por Siddhesh Sawant **[5]** que enmarca lo siguiente:

- Los modelos usados por el autor corresponden a la Regresión Logística, la Máquina de soporte vectorial, los K vecinos cercanos, Naives Bayes Gaussiana, Naives Bayes Bernoulli, Arboles de decisiones, Random Forest, Redes Neuronales y XG Boost.
- La metodología de validación fue con datos de entrenamiento y validación en una proporción de 80/20 y adicional a ello usó la validación cruzada.
- Para el autor el mejor modelo fue el Random Forest con una exactitud del 91%.

La tercera solución planteada la realiza Dmitry Uarov **[6]** donde propone lo siguiente:

- Usa los modelos de máquinas de soporte vectorial, los K vecinos Cercanos, Random Forest, XGBC Classifier y LGBM Classifier.
- Usa dos metodologías de validación las cuales fueron con datos de entrenamiento y validación en una proporción de 80/20 y la validación cruzada.
- Encuentra que el mejor modelo que clasifica de mejor manera es el Random Forest con un recall del 71%.

Y por último se encuentra la clasificación realizada por Alexandre Petit **[7]** y propone el siguiente modelo:

- Usa los modelos de Regresión Logística, Arboles de Decisiones, Random Forest y Gradient Boosting.
- Este proyecto solo usa la validación por medio de datos de entrenamiento y validación en un porcentaje de 80/20.
- El resultado final predice de mejor manera el modelo de Regresión Logística con un recall del 83% prediciendo las salidas positivas.

B. Entrenamiento y evaluación de modelos

1) Experimentos:

A continuación, se hará un breve recuento de lo encontrado y analizado en la primera iteración de los modelos de machine learning usados en la base de datos Stroke Prediction Dataset pero antes se explicará muy brevemente lo relacionado con el análisis exploratorio:

La base de datos cuenta con 5110 registros y 12 variables con las siguientes características y conteos:

- **id**: Se refiere a un código único que tiene cada paciente. Su distribución es única en cada registro.
- **gender**: Aquí se encuentran 3 géneros los cuales son "*Male*", "*Female*" y "*Other*"
  Male = 2115, *Female*=2994 y *Other*=1
- **age**: Hace referencia a la edad del paciente. Es una escala numérica que va desde 0.08 a 82 años.
- **hypertension**: La base de datos clasifica con 0 si el paciente no tiene hipertensión y con 1 si el paciente sufre hipertensión.
  0= 4612 y *1*=498
- **heart_disease**: En esta variable se clasifica con 0 si el paciente no tiene ninguna enfermedad cardíaca y con 1 si el paciente padece una enfermedad cardíaca.
  0=4834 y *1*=276
- **ever_married**: Esta variable explica si el paciente está casado con "Yes" y con "No" si no lo está.
  Yes=3353 y *No*=1757
- **work_type**: Se divide en si trabajó con niños como "*children*", si obtuvo un trabajo en el gobierno "*Govt_jov*", si nunca trabajó "*Never_worked*", si trabajó en el sector privado "*Private*" o por el contrario trabajó como independiente "*Self-employed*".
  Private=2925, *Self-employed*=819, *Children*=687, *Govt_job*=657 y *Never_worked*=22
- **Residence_type**: Se divide en si la zona de residencia es rural "*Rural*" o urbana "*Urban*".
  Urban=2596 y *Rural*=2514
- **avg_glucose_level**: Dato numérico que mide el nivel promedio de glucosa en sangre. Es una escala numérica que va desde 55.12 a 271.74.
- **bmi**: Dato numérico que muestra el índice de masa corporal que va desde 10.3 hasta 97.6
- **smoking_status**: Columna que especifica si el paciente ya había fumado anteriormente "*formerly smoked*", si nunca fumó "*never smoked*", si en la actualidad fumaba "*smokes*" o si la información no está disponible para el paciente como "*Unknown*".
  never smoked=1892, *Unknown*=1544, *formerly smoked*=884 y *smokes*=789.
- **stroke**: Es la variable respuesta y nos indica con 1 si el paciente tuvo un accidente cerebrovascular o

con 0 si no lo tuvo. Su distribución se verá gráficamente a continuación:

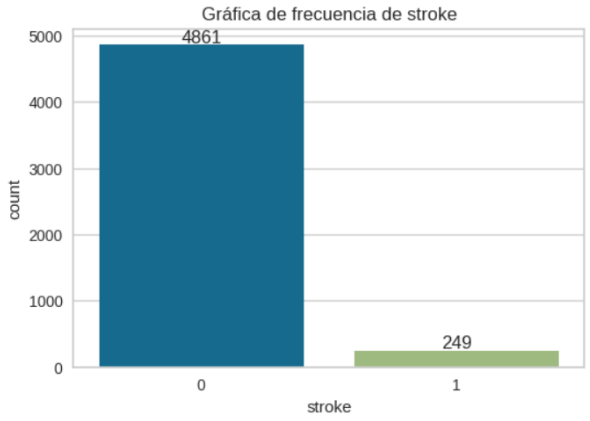


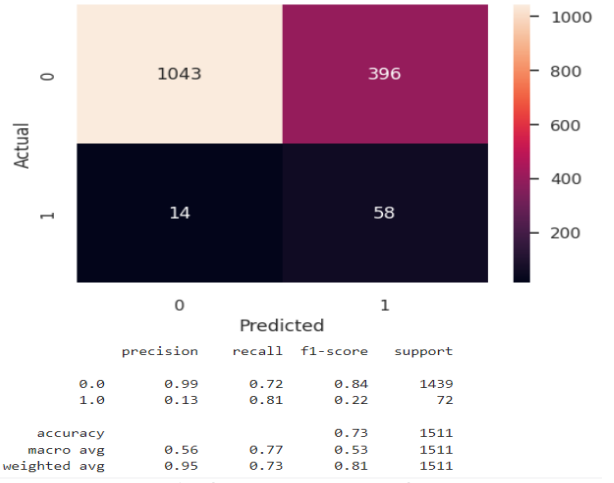
Fig. 1 Corresponde a la variable de salida llamada stroke.

Esta última variable vista gráficamente permite verificar que se encuentra desbalanceada en porcentaje de 4.87% para registros positivos contra un 95.13% para registros negativos. Por tal motivo se utiliza la técnica de RandomOverSampler para aumentar la cantidad de registros sintéticos positivos.

En el proceso de análisis exploratorio se elimina el registro other de la variable gender ya que solo es un registro, logrando así reducir una variable al momento de aplicar dummies para la transformación de variables categóricas a numéricas. Se aplica imputación por la mediana para evitar que los datos atípicos generen ruido en la medida.

Otro de los procesos realizados fue la eliminación de datos atípicos de forma controlada por el método LOF para evitar perder información importante en los casos positivos de stroke.

El siguiente paso es balancear la variable de salida pero antes de eso se realiza el ejercicio sin balancearla y usando el parámetro *class_weight=”balance”* para una regresión logística. Este parámetro lo que hace es que da más peso a la clase con menos registros ya que la utilidad del caso es conocer los casos positivos de los accidentes cerebrovasculares [8]. Así tendremos un punto de comparación con los modelos aplicando balanceo de datos. Seguido a ello, se usó la validación con el método de datos de entrenamiento y prueba en una proporción 70/30 y estos fueron los resultados de esa matriz de confusión.



	precision	recall	f1-score	support
0.0	0.99	0.72	0.84	1439
1.0	0.13	0.81	0.22	72
accuracy			0.73	1511
macro avg	0.56	0.77	0.53	1511
weighted avg	0.95	0.73	0.81	1511

Fig. 2 Matriz de confusión para primera iteración sin balanceo y usando la medida class_weight = ”balance”

Se puede observar que la medida recall logra un buen porcentaje para los registros positivos del 0.81 y esto es debido a lo anteriormente mencionada dado que se da prioridad y mayor peso a esta clase.

2) Medidas de desempeño

Las medidas de desempeño que se usaron fueron varias por practicidad, pero lo que se busca es aumentar la precisión y el recall debido a que es un caso médico con variable de salida desbalanceada. Esta última métrica nombrada será la medida principal ya que se desea que el modelo arroje buenas predicciones en el caso de la clase positiva debido a que es la que tiene menos registros y por lo tanto menos peso.

C. Resultados

Hasta el momento los modelos que han sido usados en el proyecto para predecir tanto los casos positivos como negativos de la variable stroke han sido la Regresión Logística, k Vecinos Cercanos, Máquinas de Soporte Vectoriales, Random Forest, Arboles de decisiones, Redes Neuronales y Naives Bayes. En esta primera iteración se usaron estos modelos sin ningún tipo o poca parametrización con el fin de observar cómo funcionaban en la base de datos. A continuación, en la siguiente figura Fig. 3 se encuentran los modelos usados en la primera iteración:

```
modelSVC       = svm.SVC(kernel='linear', probability = True)
modelLR        = LogisticRegression(random_state=0)
modelknn       = neighbors.KNeighborsClassifier(n_neighbors = 31)
modelComplNB   = ComplementNB()
modelTreeClas  = tree.DecisionTreeClassifier(random_state=0)
modelRanForest = RandomForestClassifier(n_estimators=50, max_depth=8, random_state=0)
ModelNN        = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(15, 2), random_state=0)
```

Fig. 3 Modelos usados por default sin ningún proceso de mejora por hiperparámetros.

Además, se validaron con el método de entrenamiento y prueba en una proporción 70/30 y se usó de igual manera la validación cruzada y la curva ROC y sus resultados se encuentran en la TABLA I. Es importante precisar que al

principio de la exploración de datos se sacaron 10 muestras por cada clase de la variable de salida para usarlas en la instancia de pruebas para conocer el impacto de cada modelo sobre muestras reales.

TABLA I
MEDIDAS DE LOS MODELOS SIN HIPERPARÁMETROS

Modelos	Recall		Precision		f1-score	
	70%/30%	CV	70%/30%	CV	70%/30%	CV
Support Vector Machines	0,828	0,815	0,752	0,747	0,788	0,781
Logistic Regression	0,816	0,815	0,761	0,756	0,787	0,784
K Nearest Neighbor	0,88	0,947	0,749	0,735	0,809	0,827
Naive Bayes	0,693	0,69	0,669	0,67	0,68	0,683
Decision Tree Classifier	1	1	0,949	0,958	0,973	0,978
Random Forest	0,969	0,96	0,843	0,84	0,902	0,896
Neural Network	0,932	0,935	0,759	0,75	0,837	0,832

Modelos	Accuracy		ROC	
	70%/30%	CV	70%/30%	CV
Support Vector Machines	0,78	0,7714	0,85	-
Logistic Regression	0,782	0,776	0,85	-
K Nearest Neighbor	0,795	0,802	0,88	-
Naive Bayes	0,679	0,675	0,75	-
Decision Tree Classifier	0,973	0,978	0,97	-
Random Forest	0,896	0,889	0,97	-
Neural Network	0,821	0,812	0,89	-

CV = Cross Validation

La curva ROC también fue una medida buscada en el planteamiento de este problema, los resultados de esta curva medida se encuentran también en la TABLA I en la columna de ROC y dejan ver un posible sobreajuste en los modelos Decision Tree Classifier y Random Forest.

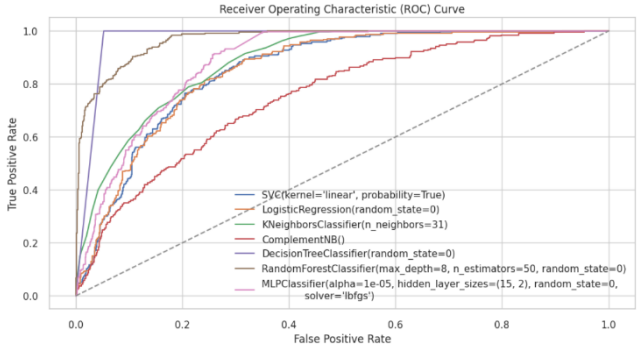


Fig. 4 Esta curva ROC muestra todos los modelos usados para la configuración de esta medida

Seguido de ello se usaron 3 modelos con algunos hiperparámetros basados en validación cruzada que se encuentran en la figura Fig. 5, con ellos se buscó mejorar el f1-score y así subir tanto la precisión y el recall.

```
modelLR2=LogisticRegression(multi_class = "ovr", solver='liblinear',  random_state=0, penalty='l1')
modelLR3=LogisticRegression(multi_class = "ovr", solver='lbfgs',  random_state=0)
modelLR4=LogisticRegression(multi_class = "multinomial", solver='lbfgs',  random_state=0)
modelsvmh1 = svm.SVC(kernel='linear', C= 5,  random_state = 0)
modelsvmh2 = svm.SVC(kernel='poly', C=10 , degree=3 , random_state = 0)
modelsvmh3 = svm.SVC(kernel='rbf', C=10 , gamma=7 , random_state = 0)
Knn1=KNeighborsClassifier(metric='manhattan', n_neighbors=37)
```

Fig. 5 Modelos usados con mejoras de hiperparámetros

Los resultados obtenidos de esta segunda iteración con modelos usando hiperparámetros se encuentran a continuación en la TABLA II.

TABLA II
MEDIDAS DE LOS MODELOS CON HIPERPARÁMETROS

Modelos	Recall (CV)	Precision (CV)	f1-score (CV)	Accuraxy (CV)
Logistic Regression 2	0,816	0,757	0,785	0,777
Logistic Regression 3	0,819	0,758	0,787	0,779
Logistic Regression 4	0,816	0,757	0,785	0,777
SMV 1	0,77	0,77	0,77	0,77
SMV 2	0,86	0,86	0,86	0,86
SMV 3	0,97	0,97	0,97	0,97
KNN1	0,81	0,84	0,81	0,81

CV = Cross Validation

Según los datos obtenidos, el mejor modelo hasta ahora hablando por la métrica de recall fue el modelo knn con un valor de 0.88 con el modelo inicial (sin hiperparámetros) y usando la validación cruzada se obtiene un valor de 0.947. Cabe mencionar que los árboles de decisiones y el random forest tuvieron métricas muy buenas pero se descartaron por ahora debido a que presentaron un sobreajuste al momento de probar con las 20 muestras removidas de la base de datos.

Ahora si se contrastan las métricas obtenidas en la sección de comparaciones contra las realizadas en este proyecto se puede resumir en lo siguiente:

El modelo randon forest fue el ganador en 3 de los cuatro proyectos contra uno de regresión logística. Para este proyecto el modelo ganador fue los k vecinos cercanos.

Ahora si comparamos las métricas principales por cada proyecto tenemos lo siguiente:
En el primer proyecto comparativo se obtiene como mejor modelo el Random Forest con un f1-score del 95%. Por parte de este trabajo el f1-score fue del 82% en el modelo KNN.
Para el segundo proyecto el Random Forest fue el mejor modelo con una exactitud del 91% en comparación con este trabajo que cuenta con un 80.2% en el modelo KNN.
En el tercer proyecto el modelo Random Forest obtuvo un recall del 71% y para este trabajo es del 94.7% con validación cruzada en el modelo KNN.
Y para finalizar el cuarto proyecto posee una recall del 83% para la Regresión Logística mientras que la validación cruzada en este trabajo otorga 94.7% en la medida recall para el modelo KNN.

Como conclusión final KNN es el modelo que mejor se desempeñó para nuestra base de datos pero hasta ahora el proyecto se encuentra en una etapa inicial y es posible que más adelante modelos diferentes arrojen mejores resultados.

BIBLIOGRAFIA

[1] F. Soriano Palacios, «kaggle.com,» 2021. [En línea]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/code. [Último acceso: 15 05 2023].

[2] OMS, «who.int,» 17 05 2017. [En línea]. Available: https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). [Último acceso: 08 06 2023].

[3] OMS, «who.int,» 09 12 2020. [En línea]. Available: https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death. [Último acceso: 08 06 2023].

[4] T. Konstantin, «kaggle.com,» 2021. [En línea]. Available: https://www.kaggle.com/code/thomaskonstantin/analyzing-and-modeling-stroke-data. [Último acceso: 08 06 2023].

[5] S. Sawant, «kaggle.com,» 2021. [En línea]. Available: https://www.kaggle.com/code/siddheshera/stroke-eda-smote-9-models-90-accuracy/notebook#Stroke-Prediction. [Último acceso: 08 06 2023].

[6] D. Uarov, «kaggle.com,» 2021. [En línea]. Available: https://www.kaggle.com/code/dmitryuarov/stroke-eda-prediction-with-6-models. [Último acceso: 08 06 2023].

[7] A. Petit, «kaggle.com,» 10 2022. [En línea]. Available: https://www.kaggle.com/code/alexandrepetit881234/stroke-prediction/notebook . [Último acceso: 08 06 2023].

[8] J. I. Baganato, «aprendemachinelearning.com,» 19 05 2019. [En línea]. Available: https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/. [Último acceso: 08 06 2023].