# INFORME CIENTÍFICO: COMPARACIÓN DE TRES TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Diego Hernández Jiménez

Máster en metodología de las ciencias del comportamiento y de la salud

Curso 2020/2021

**Resumen**

Desde un enfoque basado en el aprendizaje automático, se realiza la comparación de tres modelos para una tarea de clasificación de una muestra en dos clases, “individuos que han sufrido ictus” e “individuos que no han sufrido ictus”. Los clasificadores son un modelo de análisis cuadrático discriminante, un árbol de decisión y una máquina de vector soporte. Para evitar los problemas derivados del severo desequilibrio de clases, se propone una estrategia basada en el aprendizaje sensible a costes. Los modelos se evalúan mediante validación cruzada y se obtienen medidas de tasa de acierto, sensibilidad, especificidad, medida $F_1$ y AUC. Del análisis de los resultados se concluye que el modelo de árbol de decisión es el mejor de los tres. Finalmente se revisan algunas limitaciones.

---

Los accidentes cerebrovasculares o ictus constituyen la segunda causa de muerte en la población general española, siendo además la primera causa de muerte en mujeres, (Consejo Interterritorial del Sistema Nacional de Salud, 2009). En otros países desarrollados la situación es similar (en Estados Unidos, por ejemplo, es la quinta causa de muerte, Center of Disease Control and Prevention, 2018). Debido al enorme impacto que tienen a nivel social, resulta necesario investigar acerca de sus determinantes. Son muchos los factores de riesgo que han sido estudiados, pero desde un punto de vista aplicado resultan de mayor un subconjunto concreto de ellos. Siendo el objetivo el desarrollar planes de prevención a gran escala, son más relevantes aquellos fácilmente identificables (como la edad, frente a marcadores genéticos, por ejemplo), y preferiblemente modificables. Entre éstos se encuentra, por ejemplo, la hipertensión, que es el factor más estrechamente relacionado con el riesgo de sufrir ictus, el padecimiento de diabetes mellitus, o incluso la obesidad y el estilo de vida sedentario (Boehme et al., 2017).

Sin embargo, la investigación en este contexto plantea algunos problemas. No es posible, por ejemplo, realizar diseños experimentales, y tampoco resultan factibles en muchos casos los diseños cuasi-experimentales. Una alternativa son los estudios de riesgo y los diseños de casos y controles (Pardo y Ruiz, 2015). Otra alternativa dentro del planteamiento no experimental es adoptar un enfoque basado en aprendizaje automático. La mayor virtud de este enfoque es que permite explotar grandes bases de datos para realizar predicciones en tareas de clasificación y regresión. Es precisamente ésta la aproximación que va a guiar el presente estudio.

Específicamente, se van a proponer tres modelos distintos, que ya han sido evaluados de manera independiente en trabajos anteriores, para tratar de clasificar de manera precisa una muestra de sujetos en dos clases, "individuos que han sufrido ictus" e "individuos que no han sufrido ictus". El objetivo es seleccionar el modelo con el mejor rendimiento, de acuerdo con una serie de criterios: precisión en las predicciones, sensibilidad y parsimonia. En el apartado de método se describen de manera más formal.

**Método**

Los análisis se realizan con el lenguaje de programación Matlab, versión R2020b.

La muestra empleada para construir y evaluar los modelos fue publicada por McKinsey & Company (Nwosu et. al, 2019). Contiene 43.000 observaciones. Cada una se corresponde con una persona distinta que tiene puntuaciones en diez atributos distintos. Además, otro atributo hace referencia a la variable de clasificación, el haber sufrido o no ictus. Concretamente, la variable *Stroke* define la categoría *Yes* y la categoría *No*. Una característica relevante de la base de datos es que la distribución de casos *Yes* y casos *No* es muy asimétrica, tan solo el 1,8 % de la muestra pertenece a la categoría *Yes*. Este desequilibrio puede considerarse "intrínseco" (He y Garcia, 2009), es decir, está directamente relacionado con la naturaleza del fenómeno y supone un gran problema. Al utilizar directamente la muestra, los modelos producen resultados que, si

bien maximizan la precisión (porcentaje total de casos clasificados correctamente), son extremadamente sesgados, pues la precisión se consigue a costa de clasificar todos los ejemplos como pertenecientes a la clase mayoritaria, en este caso la clase *No.* Este problema puede abordarse de múltiples formas (He y Garcia, 2009), pero la estrategia que se va a emplear aquí es la que ya se ha utilizado en trabajos anteriores, y se basa en el aprendizaje sensible a costes (*cost sensitive learning*). De manera general, primero se define una matriz $C$ de costes de errores de clasificación:

| | Predicción = *No* | Predicción = *No* |
|---|---|---|
| Verdadero = *No* | $C_{00}$ | $C_{01}$ |
| Verdadero = *Yes* | $C_{10}$ | $C_{11}$ |

*Tabla 1: Matriz de costes de errores de clasificación*

Esta matriz permite calcular la función de coste esperado, cuya definición es similar a la del valor esperado (Ling y Sheng, 2008). Sea x un ejemplar cualquiera y sea i una de las clases, el coste esperado de clasificar el caso x como perteneciente a la clase i es:

$$E(i|x) = \sum_{j=0}^{1} C_{ij} P(j|x)$$

Los clasificadores que se van a evaluar pronostican probabilidades, que son las que se emplean para asignar las categorías. Realmente, la regla de decisión se basa en minimizar el coste esperado, de modo que se clasifica un caso x como de la clase 1 cuando $E(1|x) \leq E(0|x)$. En la matriz de costes, $C_{00}$ y $C_{11}$ valen cero, pues no resulta razonable penalizar los aciertos, de modo que la regla de decisión puede redefinirse como clasifica un caso x como de la clase 1 cuando:

$$C_{10} P(1|x) \leq C_{01} P(0|x)$$

Si no se penaliza de manera diferente unos errores frente a otros, es obvio que la regla de decisión únicamente se basa en las probabilidades pronosticadas. Sin embargo, en este caso se penalizan más gravemente los falsos negativos, casos para los que se predice que no se sufre ictus cuando realmente sí se sufre. Ello supone que $C_{10} > C_{01}$. No solo es coherente con los

criterios de decisión en contextos aplicados, sino que, además, de esta manera se logra compensar el desequilibrio de clases.

En cuanto a los algoritmos que se van a comparar, la primera técnica de clasificación es un modelo basado en análisis cuadrático discriminante. El único predictor es la edad (*age*). Este modelo difiere ligeramente del que se utilizó en el primer trabajo. En aquella ocasión no se empleó el aprendizaje sensible a costes, sino que se afrontó el problema del desequilibrio de clases haciendo uso del submuestreo repetido de la clase mayoritaria (*downsampling*, ver He y Garcia, 2009). Aun a riesgo de estar proponiendo un modelo subóptimo, en este trabajo se ajusta un modelo con el mismo predictor pero siguiendo la estrategia de sensibilidad a los costes. La matriz de costes, que es $\begin{bmatrix} 0 & 1 \\ 70 & 0 \end{bmatrix}$.

El segundo modelo propuesto es un árbol de decisión, concretamente el algoritmo ID3. Se utilizan como predictores el grupo de edad (*age_cat*), si se padece hipertensión (*hypertension*), si se padece alguna afección cardíaca (*heart_disease*) y la categoría de índice de masa corporal (*bmi_cat*). Una descripción más detallada de estas variables puede encontrarse en el anexo 1. Los parámetros más relevantes del modelo son el criterio de partición, que se basa en la reducción de la entropía, el número mínimo de casos por hoja, que se fija en 200, y la matriz de costes, que es $\begin{bmatrix} 0 & 1 \\ 70 & 0 \end{bmatrix}$.

El tercer y último modelo es una máquina de vector soporte. Las variables predictoras son *age*, el nivel de glucosa en sangre en mg/dl (*avg_glucose_level*), *hypertension*, *heart_disease* y condición de fumador (*smoking_status*). El kernel utilizado es el lineal y el parámetro "BoxConstraint" de Matlab, que se corresponde con el parámetro C, se fija en 0,24 (por defecto el valor es 1, y valores altos indican menor tolerancia a las violaciones del margen, y por tanto menor número de vectores soportes). La matriz de costes, que es $\begin{bmatrix} 0 & 1 \\ 55 & 0 \end{bmatrix}$.

Puede observarse que los modelos no son estrictamente equivalentes porque varían en las variables utilizadas y también en los costes asociados a los errores. A pesar de ello, es necesario recordar que los modelos que van a ser comparados son los que mejor rendimiento han obtenido al ser evaluados de manera individual.

El procedimiento seguido en el análisis es el siguiente (en el anexo 2 se facilita el código utilizado): Primero se normalizan las variables cuantitativas para que se encuentren en la misma escala [0;1] mediante la ecuación:

$$X^* = \frac{X - min_X}{max_X - min_X}$$

Donde X es una variable cuantitativa cualquiera y X* es la variable reescalada.

Como parte del preprocesamiento también se eliminan los valores perdidos. Se toma esta decisión porque solo existen para la variable *bmi_cat* y su número no es muy elevado. Por último, se transforman las variables cuantitativas en variables indicador (*dummy variables*) cuando se requiere (máquina de vector soporte). Esta transformación se realiza en Matlab automáticamente.

A continuación, se ajusta cada modelo con las variables y parámetros especificados mediante validación cruzada de orden 10. Esta estrategia es preferible a la validación simple con una muestra de entrenamiento y otra de test porque permite emplear toda la muestra, evitando así posibles sesgos de selección. Las medidas de rendimiento obtenidas son mejores representaciones de la capacidad de generalización de los modelos. El segundo paso consiste precisamente en obtener estos estadísticos de rendimiento. Finalmente se hace una comparación cuantitativa y cualitativa de los algoritmos y se selecciona el más apropiado.

La calidad del modelo viene descrita por:

Precisión: No hace referencia a una, sino a tres medidas que resumen la capacidad predictiva "global" del modelo. Sean VP, FP los verdaderos y falsos positivos y VN, FN los verdaderos y falsos negativos y N el tamaño de la muestra empleada

- Tasa de acierto: Proporción de casos correctamente clasificados.

$$\frac{VP + VN}{N}$$

- Medida $F_1$ : Media armónica de dos proporciones, la sensibilidad (ver más adelante) y la proporción de casos *Yes* correctamente identificados del total de casos pronosticados como *Yes.* Una medida más general es $F_\beta$ y permite ponderar de diferente manera la importancia de la sensibilidad, pero dado que ese ajuste se ha introducido directamente en el modelo mediante la matriz de costes, aquí se fija el valor de $\beta$ en 1. La ecuación que permite el cálculo de $F_1$ es

$$\frac{VP}{VP + \frac{1}{2}(FP + FN)}$$

- Área bajo la curva ROC (AUC): La curva ROC relaciona la sensibilidad y especificidad del modelo (ver más adelante) para un conjunto de criterios o umbrales de decisión (probabilidad a partir de la cual se clasificará un caso como *Yes*). El área bajo la curva resume esta relación para todos los umbrales de decisión.

Sensibilidad y especifidad: Esta información queda recogida en los estadísticos anteriores, pero dada su importancia en este contexto, se evalúan independientemente.

- Sensibilidad: Proporción de casos *Yes* correctamente clasificados del total de casos *Yes*

$$\frac{VP}{VP + FN}$$

- Especificidad: Proporción de casos *No* correctamente clasificados del total de casos *No*

$$\frac{VN}{VN + FP}$$

Parsimonia: Es un criterio más bien cualitativo. A igual o semejantes valores en los estadísticos anteriores, se considera más parsimonioso a aquel modelo que incluye menos variables. En este caso, también se considera más parsimonioso aquel modelo cuya interpretación resulta más sencilla.

**Resultados y discusión**

Los resultados de cada modelo quedan resumidos en la siguiente tabla

| | Tasa de acierto | Medida $F_1$ | AUC | Sensibilidad | Especificidad |
|---|---|---|---|---|---|
| QDA | 0,6331 | 0,0803 | 0,8396 | **0,8876** | 0,6284 |
| Árbol de decisión | **0,7482** | **0,1050** | 0,8118 | 0,8186 | **0,7469** |
| SVM | 0,7218 | 0,0974 | **0,8507** | 0,8314 | 0,7198 |

*Tabla 2. Medidas de rendimiento de los modelos. En negrita la puntuación más alta obtenida para cada estadístico*

Considerando todas las medidas, puede observarse que el modelo de árbol de decisión es el mejor de los tres, ya que obtiene las puntuaciones más altas en tres de las cinco medidas propuestas. No obstante, cabe destacar que no hay una clara superioridad, al menos con respecto al modelo de máquina de vector soporte. De hecho, la mayor diferencia se da en el caso de la especificidad, y es de 0,0271 puntos. Por otro lado, el AUC del modelo SVM es aproximadamente 0,04 puntos superior al del árbol de decisión. También el AUC del modelo QDA, a pesar de contar con un solo predictor, es superior. Sin embargo, aunque no se ha realizado ningún contraste de significación, puede apreciarse por la representación gráfica de las tres curvas ROC (Figura 1) y de las AUC (Figura 2) que los valores son bastante semejantes.

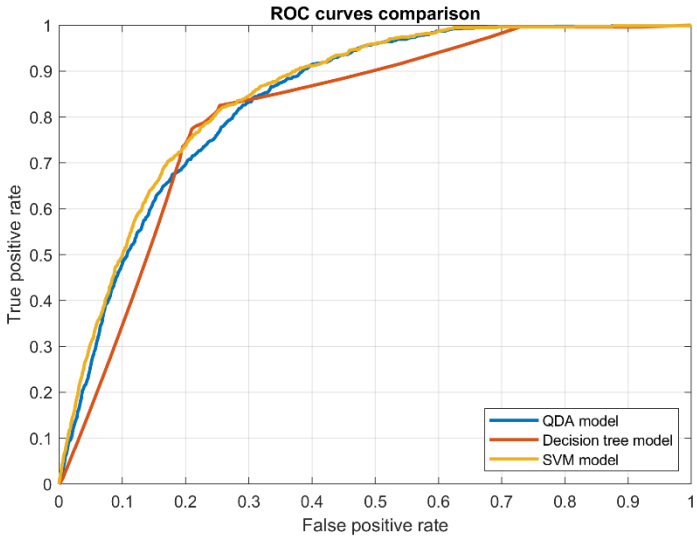


*Figura 1. Comparación de curvas ROC*

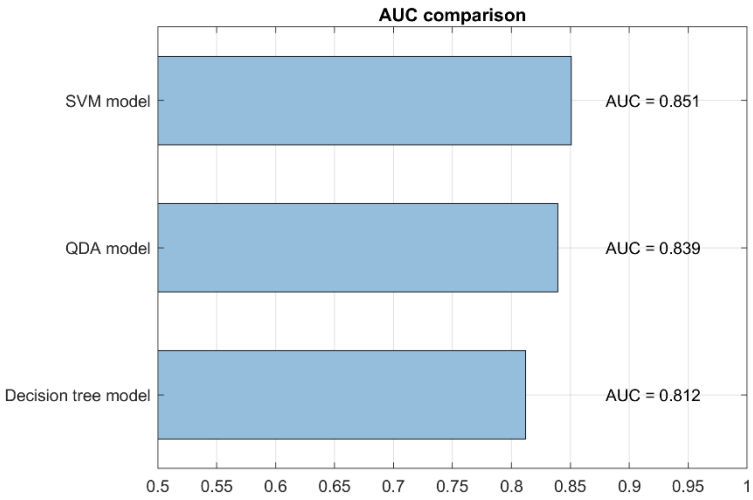


*Figura 2. Comparación de valores AUC*

Por otro lado, en términos de parsimonia, asumiendo que el árbol de decisión y el modelo SVM (los mejores a nivel general) son más o menos equivalentes en rendimiento, se considera más simple el modelo de árbol de decisión. Éste, contando con una variable menos, consigue resultados muy similares. Además, en lo referido a la interpretación resulta mucho más práctico el modelo de árbol, pues las reglas de decisión que genera (la "estructura" del árbol) es directamente visible.

Teniendo en cuenta todos estos motivos, se decide finalmente seleccionar como mejor modelo el árbol de decisión.

**Conclusiones**

El modelo finalmente propuesto obtiene unos resultados razonablemente buenos y que son congruentes con lo que otros investigadores han logrado en la misma base de datos (Nwosu et. al, 2019). A pesar de ello, existen algunas limitaciones que se deben tener en cuenta. En primer lugar, como se ha mencionado anteriormente, los modelos comparados no son equivalentes. El clasificador basado en funciones discriminantes, por ejemplo, solo incluía una variable predictora. Esto probablemente haya podido contribuir a obtener los resultados que se han obtenido, que eran inferiores a los hallados con los otros dos modelos. Por otra parte, a nivel práctico, si bien es cierto que el modelo puede resultar útil para clasificar sujetos con riesgo de sufrir ictus, esta capacidad predictiva se debe en gran parte a la variable edad (*age*). Quizá de mayor interés hubiera sido construir modelos que únicamente contasen con factores de riesgo modificables como predictores.

**Referencias**


Boehme, A. K., Esenwa, C., y Elkind, M. S. (2017). Stroke risk factors, genetics, and prevention. *Circulation research*, 120(3), 472-495.

Center of Disease Control and Prevention (2018). Stroke facts. Recuperado de https://www.cdc.gov/stroke/facts.htm

Consejo Interterritorial del Sistema Nacional de Salud (2009). *Estrategia en Ictus del Sistema Nacional de Salud*. Recuperado de https://www.mscbs.gob.es/organizacion/sns/planCalidadSNS/docs/EstrategiaIctusSNS.pdf

He, H., y Garcia, E. A. (2009). Learning from imbalanced data*. IEEE Transactions on knowledge and data engineering*, 21(9), 1263-1284

Ling, C. X., & Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. Encyclopedia of machine learning, 2011, 231-235.

Nwosu, C. S., Dev, S., Bhardwaj, P., Veeravalli, B., y John, D. (2019, July). Predicting stroke from electronic health records. En *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 5704-5707). IEEE.

Pardo, A., Ruiz, M.A. (2015). Inferencia con dos variables categóricas. En A. Pardo, M.A. Ruiz y R. San Martín (Eds.), Análisis de datos en ciencias sociales y de la salud II (pp.94-104). Editorial Síntesis.

## Anexo 1: Descripción de las variables

Edad (*age*): Edad cronológica de la persona.

Grupo de edad (*age_cat*): variable categórica creada a partir de *age* con los niveles

- *Kid* (edad menor o igual a 14 años)
- *Young* (edad comprendida entre 14 y 25)
- *Mature* (edad comprendida entre 25 y 64)
- *Elder* (edad mayor de 64)

Nivel de glucosa (*avg_glucose_level*): Nivel de glucosa en sangre en mg/dl.

Hipertensión (*hypertension*): Variable binaria. Indica si se padece o no hipertensión.

Afección cardíaca (*heart_disease*): Variable binaria. Indica si se padece o no alguna afección cardíaca.

Grupo de índice de masa corporal (*bmi_cat*): variable categórica creada a partir de la variable cuantitativa bmi. Los niveles son:

- *Obese* (bmi mayor o igual 30)
- *Overweight* (bmi entre 25 y 30)
- *Normal* (bmi entre 18.5 y 25)
- *Underweight* (bmi menor de 18.5)

Condición de fumador (*smoking_status*): Variable categórica con niveles:

- *never smoked* (no ha fumado nunca)
- *formerly smoked* (ha sido fumador, pero actualmente no fuma)
- *smokes* (fuma actualmente)
- *unknown* (no se ha obtenido una respuesta, se desconoce.)

**Anexo 2: código empleado para la realización del análisis**

Importar función `confusionmatStats` no contenida en los toolbox de Matlab (el directorio se ha omitido)

```
addpath('ruta');
```

Importar base de datos (el directorio se ha omitido)

```
stroke=readtable('ruta');
```

Normalizar variables cuantitativas relevantes

```
stroke(:,["age","avg_glucose_level"])=normalize(stroke(:,["age","avg_glu
cose_level"]),'range');
```

Definir los atributos de cada modelo y la variable de clasificación

```
featQDA=stroke.age;
feattree=stroke(:,["age_cat","hypertension","heart_disease","bmi_cat"]);
featSVM=stroke(:,["age","avg_glucose_level","hypertension", ...
    "smoking_status","heart_disease"]);

% target
group=stroke.stroke;
```

Construcción del modelo discriminante cuadrático y obtención de medidas de rendimiento

```
qda=fitcdiscr(featQDA,group, ...
    'DiscrimType','quadratic', ...
    'CrossVal','on', ...
    'Cost',[0 1;70 0]);

% predictions ............
[predsQDA, postprQDA]=kfoldPredict(qda);

% assessment ..............
confusionchart(group,predsQDA);
title('QDA confusion matrix')
c=confusionmatStats(group,predsQDA);
accuracy=c.accuracy; sensitivity=c.sensitivity;
specificity=c.specificity; Fscore=c.Fscore;
```

```
 table(accuracy,sensitivity,specificity,Fscore,'RowNames',{'Class
No','Class Yes'})

 % ROC .................
 [FPRqda, TPRqda, ~, AUCqda]=perfcurve(group,postprQDA(:,2),'Yes');
 AUCqda
```

Construcción del modelo de árbol de decisión y obtención de medidas de rendimiento

```
 tree=fitctree(feattree,group, ...
     'CategoricalPredictors','all', ...
     'SplitCriterion','deviance', ...
     'MinLeafSize',200, 'MinParentSize',230, ...
     'CrossVal','on', ...
     'Cost',[0 1;70 0]);

 % predictions ............
 [predstree, postprtree]=kfoldPredict(tree);

 % assessment ..............
 confusionchart(group,predstree);
 title('Decision tree confusion matrix')
 c=confusionmatStats(group,predstree);
 accuracy=c.accuracy; sensitivity=c.sensitivity;
specificity=c.specificity; Fscore=c.Fscore;
 table(accuracy,sensitivity,specificity,Fscore,'RowNames',{'Class
No','Class Yes'})

 % ROC .................
 [FPRtree, TPRtree, ~, AUCtree]=perfcurve(group,postprtree(:,2),'Yes');
 AUCtree
```

Construcción del modelo de máquina de vector soporte y obtención de medidas de rendimiento

```
 SVM=fitcsvm(featSVM,group, ...
'CategoricalPredictors',{'hypertension','smoking_status','heart_disease'}
, ...
     'BoxConstraint', 0.24, ...
     'KernelFunction','linear', ...
     'CrossVal','on', ...
     'Cost',[0 1;55 0]);

 % predictions ............
 [predsSVM, postprSVM]=kfoldPredict(SVM);

 % assessment ..............
 confusionchart(group,predsSVM);
```

```
 title('SVM confusion matrix')
 c=confusionmatStats(group,predsSVM);
 accuracy=c.accuracy; sensitivity=c.sensitivity;
specificity=c.specificity; Fscore=c.Fscore;
 table(accuracy,sensitivity,specificity,Fscore,'RowNames',{'Class
No','Class Yes'})

 % ROC ..................
 [FPRSVM, TPRSVM, ~, AUCSVM]=perfcurve(group,postprSVM(:,2),'Yes');
 AUCSVM
```

Comparación gráfica de los tres modelos

```
 % ROC comparison
 plot(FPRqda,TPRqda,'LineWidth',2);
 hold on
 grid on
 plot(FPRtree,TPRtree,'LineWidth',2);
 plot(FPRSVM,TPRSVM,'LineWidth',2);
 title('ROC curves comparison');
 xlabel('False positive rate');
 ylabel('True positive rate');
 legend({'QDA model' 'Decision tree model' 'SVM
model'},'Location','southeast');
 hold off

 % AUC comparison
 models=categorical({'QDA model' 'Decision tree model' 'SVM model'});

 barh(models,[AUCqda AUCtree AUCSVM],.6,'FaceColor','#95bddc')
 title('AUC comparison')
 grid on
 text(.88,3,['AUC = ' num2str(round(AUCSVM,3))]);
 text(.88,2,['AUC = ' num2str(round(AUCqda,3))]);
 text(.88,1,['AUC = ' num2str(round(AUCtree,3))]);
 xlim([0.5 1])
```