informe SVM

Diego Hernández Jiménez

1. Introducción

En este último ptoyecto se vuelve a retomar el objetivo de clasificar una muestra de de sujetos en dos clases, "individuos que han sufrido ictus" (*Yes*) e "individuos que no han sufrido ictus" (*No*). Adicionalmente se persigue que los errores de clasificación de falsos negativos se vean reducidos al mínimo.

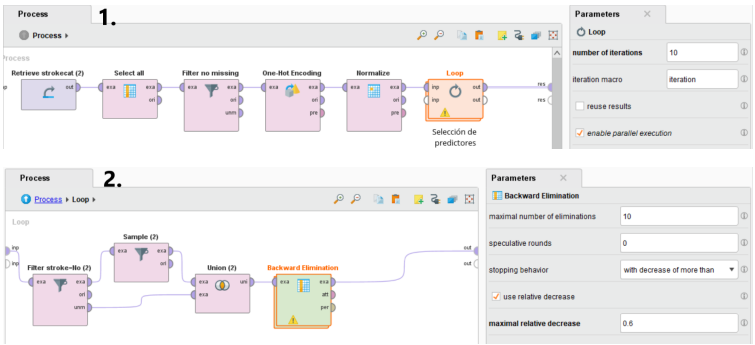
Este trabajo se diferencia de los previos en que tiene un carácter más sintético. En trabajos anteriores se han construido modelos que requerían predictores cuantitativos como modelos que incorporaban solo variables nominales. En el presente análisis se propone un modelo basado en máquinas de vector soporte que incluye ambos tipos de variables. Igualmente, mientras que en los proyectos de análisis discriminante y áboles de decisión se adoptaban estrategias distintas para enfrentar el problema de las clases no balanceadas, aquí se emplean ambas (*downsampling* y aprendizaje sensible a costes) para tratar de comparar los dos enfoques.

En cuanto al tipo de atributos a incluir, no se hace ninguna selección a priori, sino que se deja que el proceso de selección esté más guiado por los datos. Ello implica que incialmente se incluyen diez predictores: *age*, *avg_glucose_level*, *bmi*, *ever_married*, *gender*, *heart_disease*, *hypertension*, *Residence_type*, *work_type*, *smoking_status*. Resultan novedosas varias de ellas. *ever_married* es una variable binaria que indica si se ha estado casado. *Residence_type* es una variable dicotómica que hace referencia a la residencia y tiene las categorías *Urban* y *Rural*. Las categorías de *work_type* son: *Private*, *Self-employed*, *Govt_job*, *never worked* y *children*, que hace refrencia a menores .*smoking_status* tiene las categorías *never smoked*, *formerly smoked*, *smokes* y *unkown*. En el archivo original 13292 casos eran considerados *unkown*, pero al ser un número excesivo de casos desconocidos se realizó un análisis más exhaustivo. Con base en Barrington-Trimis et. al (2020) se ha considerado pertenecientes a la categoría *never smoked* a aquellos casos *unkown* con edad igual o inferior a 16, pues esa es la edad media aproximada de iniciación al consumo de tabaco en Estados Unidos (datos de 2018). De esta manera, el número de casos *unkown* se reduce a 7356.

Como parte del preprocesamiento también se han normalizado las variables cuantitativas (método de rango) y se han convertido en variables *dummy* las categóricas.

2. Métodos y herramientas

1. En la primera etapa se realiza la selección de variables. Para ello se sigue un proceso de eliminación hacia atrás mediante el operador del mismo nombre de Rapidminer. Se introducen todas las variables y en cada iteración se ajusta un modelo de máquina de vector soporte (SVM) con los parámetros por defecto y se obtiene una medida de precisión. Este proceso de ajuste se realiza mediante validación cruzada de orden 10 y tras haber realizado un submuestreo de la clase mayoritaria (*stroke=No*) y haber creado una nueva muestra con todos los casos *stroke= Yes*. Para evitar posibles sesgos de selección al haber creado esta muestra, se repite todo el proceso 10 veces. Como resultado se obtienen los 10 conjuntos de atributos que han resultado seleccionados en cada iteración. El criterio de parada en el procedimiento de eliminación es la reducción relativa de la precisión en más de 0.6.



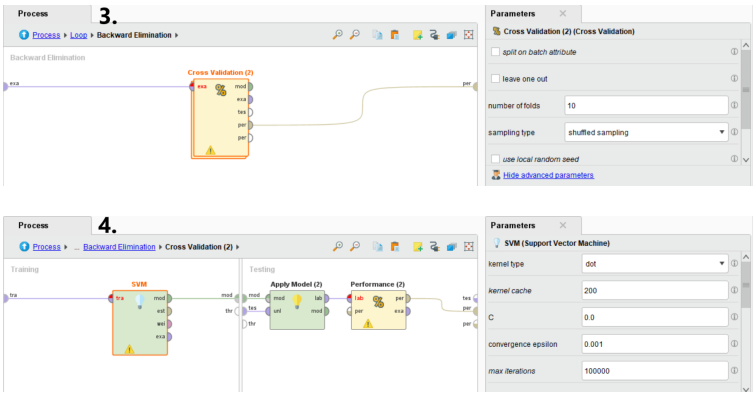


Figure 1: Proceso de selección de predictores

2. Una vez seleccionadas las variables se procede a ajustar el modelo. Se realiza mediante dos estrategias, que varían en función de la respuesta que se da al problema de desequilibrio de clases. En Rapidminer se lleva a cabo *downsampling* y partición en conjunto de calibración y conjunto de validación. Se repite el proceso con 100 muestras distintas y las medidas de rendimiento finales son un promedio de las obtenidas en todas las iteraciones. Por otra parte, en Matlab, se hace una partición en conjunto de calibración y conjunto de validación a partir de toda la muestra, pero se impone la matriz de costes [0 1;55 0] para tratar de reducir los falsos negativos. Estos primeros modelos se estiman con los parámetros por defecto en Rapidminer y Matlab y servirán como referencia. En Matlab, el código para construir los modelos y obtener medidas de rendimiento es el siguiente:

```
% stroke -> dataset
features=stroke(:,["age","avg_glucose_level","hypertension", ...
    "smoking_status","heart_disease"]);
cuanti=["age","avg_glucose_level"];
features(:,cuanti)=normalize(features(:,cuanti),"range");
group=stroke.stroke;

rng(13)
% split in train and test set
part=cvpartition(group,'HoldOut',.3, 'Stratify',true); % stratify to make sure same priors
ids_train=training(part);
ids_test=test(part);

svm1=fitcsvm(features(ids_train,:),group(ids_train), ...
    'BoxConstraint',1, ...
    'CategoricalPredictors',{'hypertension','smoking_status','heart_disease'}, ...
    'KernelFunction','linear', ...
    'Cost',[0 1;55 0]);

[preds1, postprob1]=predict(svm1,features(ids_test,:));

% assessment...........
confusionchart(group(ids_test,:),preds1);
c=confusionmatStats(group(ids_test,:),preds1);
accuracy=c.accuracy; sensitivity=c.sensitivity; specificity=c.specificity; Fscore=c.Fscore;
table(accuracy,sensitivity,specificity,Fscore,'RowNames',{'Class No','Class Yes'})

% ROC...........
```

```
[FPR1, TPR1, ~, AUC1]=perfcurve(group(ids_test),postprob1(:,2),'Yes');
area(FPR1,TPR1,'FaceColor',[0.8 0.89 0.95]);
grid on
text(.5,.6,['AUC = ' num2str(AUC1)],"FontSize",16);
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC Curve for SVM model 1')
```

3. En la tercera etapa se realiza un ajuste de los parámetros del modelo. Los más relevantes son el parámetro C y el tipo de kernel empleado. La selección de los parámetros también está guiada por los datos. Se utiliza el operador Optimize parameters en Rapidminer, y el argumento "OptimizeHyperparameters" en Matlab para encontrar la función kernel más adecuada (lineal, polinomial, gaussiana).En Rapidminer, debido a que se está siguiendo una estrategia basada en el submuestreo, este proceso de optimización se repite con 5 muestras distintas. El parámetro C, en Matlab se representa mediante el argumento `BoxConstraint`, y no es estrictamente igual, pues en este caso valores altos indican menor tolerancia, márgenes menos "suaves", al contrario que ocurre con C. Debido al gran coste computacional que supone construir múltiples niveles modelos con distintos valores, simplemente se comparan cinco en Rapidminer y diez valores aleatorios en Matlab.

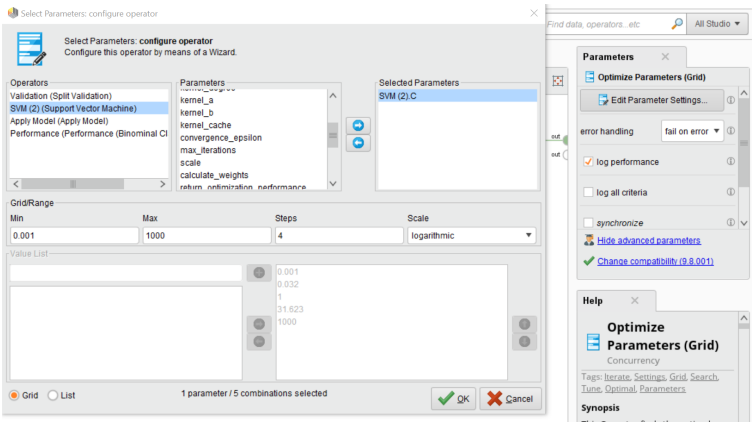


Figure 2: Especificaciones procedimiento grid search para el parámetro C

4. Finalmente, para tener mejores medidas de la capacidad de generalización del modelo y evitar en lo posible el sobreajuste, se ajusta el modelo óptimo mediante validación cruzada de orden 10.

3. Resultados y discusión

En el paso 1. Rapidminer devuelve 10 conjuntos de predictores. Se escogen los cinco más frecuentes, *age*, *heart_disease*, *avg_glucose_level*, *hypertension* y *smoking_status*. Técnicamente, al haberse convertido *smoking_status* en una variable *dummy*, se han obtenido tres variables dicotómicas distintas, pues la variable tiene 4 niveles. Y lo que se comprueba es que la variable *dummy never smoked* y *unkown* eran las que más aparecían. Sin embargo, dado que *formerly smoked* también aparece en dos de los diez modelos, se decide mantener todas las categorías, y por tanto, toda la variable.

Los modelos base de la segunda etapa son similares en Rapidminer y Matlab (ver table 1). Se reproducen los mismos patrones que en trabajos anteriores. Ambas estrategias producen tasas de acierto (global) similares.

Igualmente ambos consiguen equilibrar la sensibilidad y especificidad, dando más importancia a la sensibilidad. También son semejantes los valores AUC. La mayor diferencia se produce en la medida F_1, donde el modelo de Matlab (estrategia basada en aprendizaje sensible a costes) es muy inferior.

Table 1: resultados modelo base según la estrategia

	downsampling (Rapidminer)	aprendizaje sensible a costes (Matlab)
Precisión (%)	76.81	72.54
Sensibilidad (%)	88.36	81.28
Especificidad (%)	65.29	72.38
F (%)	79.2	9.65
AUC	0.844	0.845

En el siguiente paso se realiza una búsqueda para encontrar los parámetros óptimos del modelo. En Matlab la función kernel óptima es la lineal. Por otro lado, el parámetro `BoxConstraint` es aproximadamente 0.24. En Rapidminer el procedimiento de optimización fracasa. Si bien debían compararse únicamente tres modelos (cada uno con un kernel distinto) en cada una de las cinco iteraciones, tras más de dos horas de tiempo de ejecución se decide paralizar el proceso. También fracasa la optimización del parámetro C. Se toman, por tanto, como parámetros óptimos, los obtenidos en Matlab. A partir de estos resultados se propone un modelo en Rapidminer con kernel lineal ("dot") y con parámetro C igual a 2 (valor más laxo que el fijado por defecto). Al igual que en la etapa 1, se repite el proceso de ajuste 100 veces y se promedian los resultados. La comparación puede verse en la siguiente tabla. Puede apreciarse que los resultados para el caso de Matlab son prácticamente idénticos a los anteriores. Aunque no se ha podido identificar la causa, es posible que se deba a que los valores de `BoxConstraint` comparados, al ser solo diez, no variaban lo suficiente como para producir resultados diferentes (Table 3).

Table 2: resultados modelo base según la estrategia

	downsampling (Rapidminer)	aprendizaje sensible a costes (Matlab)
Precisión (%)	76.84	72.44
Sensibilidad (%)	88.01	81.28
Especificidad (%)	65.76	72.28
F (%)	79.13	9.62
AUC	0.842	0.8374

Table 3: valores de `BoxConstraint` comparados

Iter	Eval result	Objective runtime	Objective (observed)	BestSoFar	BoxConstraint
1	Best	0.22062	324.43	0.22062	14.072
2	Accept	0.22179	38.351	0.22062	0.017028
3	Accept	0.22629	43.901	0.22062	0.0014349
4	Accept	0.22084	47.806	0.22062	0.59913
5	Best	0.22034	47.951	0.22034	0.23966
6	Accept	0.23011	39.393	0.22034	0.0024968
7	Accept	0.22072	71.128	0.22034	0.93678
8	Accept	0.22479	45.958	0.22034	0.0011368
9	Accept	0.22571	41.075	0.22034	0.0012279
10	Accept	0.22197	2025	0.22034	552.51

El modelo ajustado mediante validación cruzada produce los siguientes resultados

Table 4: resultados modelo (validación cruzada) base según la estrategia

	downsampling (Rapidminer)	aprendizaje sensible a costes (Matlab)
Precisión (%)	76.77	72.16
Sensibilidad (%)	88.43	83.27
Especificidad (%)	65.15	71.95
F (%)	79.2	9.74
AUC	0.84	0.8507

Como se puede comprobar, los valores de los estadísticos no difieren mucho de los encontrados por el procedimiento de validación simple, lo que hace suponer que no ha habido sobreajuste.

4. Conclusiones

Los dos enfoques para tratar el problema de clases no balanceadas producen resultados similares, siendo generalmente más bajos los valores obtenidos mediante el enfoque de aprendizaje sensible a los costes. En lo que respecta al estadístico F_1, sin embargo, si hay grandes diferencias, siendo muy inferior en el segundo caso. A pesar de todo, hay que recordar que ambas estrategias no son directamente comparables, pues en un caso se emplean promedios, mientras que en el otro solo los resultados de una única muestra de validación. Además, en el primer caso (*downsampling*) se está empleando en todo momento una muestra que puede no estar reflejando apropiadamente las propiedades de la muestra de datos original, ya que para equilibrar los casos *stroke=Yes* y *stroke=No* se seleccionan siempre todos los casos *Yes*, pero solo una pequeña muestra de los casos *No*. Por eso, de ambos enfoques, resulta preferible el basado en costes.

Por último, dados los problemas encontrados y la dificultad de comparar estrategias utilizando distintos softwares, se llega a la conclusión de que resulta preferible ceñirse a un enfoque, y preferiblemente, también a un software. De esta manera se evitan errores de interpretación y se facilita el análisis (no hay que tener en cuenta, por ejemplo, cuál es la correspondencia entre el parámetro C de Rapidminer y el de `BoxConstraint` de Matlab).

5. Referencias bibliográficas

Barrington-Trimis, J. L., Braymiller, J. L., Unger, J. B., McConnell, R., Stokes, A., Leventhal, A. M., ... y Goodwin, R. D. (2020). Trends in the age of cigarette smoking initiation among young adults in the US From 2002 to 2018. *JAMA network open*, 3(10), e2019022-e2019022.