Aprendizaje supervisado 1 - Proyecto

Adrián Robles Arques

Fase 1: Análisis descriptivo de las variables

El presente conjunto de datos cuenta con un gran conjunto de variables, 22 en total, de las cuales tomaremos “Target” como nuestra variable objetivo y las 21 restantes como las variables explicativas, aunque de estas aún podemos eliminar algunas que no aportan información relevante al conjunto a la hora de elaborar un modelo predictivo. En primer lugar, vamos a catalogar las variables:

- **Cualitativas**: ID, Tendency y Target, aunque estén categorizadas como valores numéricos, no tienen sentido matemático como tal, en el caso de ID es solo un identificador, un nombre valdría igual, y para el Target solo es una categorización numérica en sustitución de un valor booleano, True/False o Si/No. Tendency por su parte solo toma 3 valores, que pueden sustituirse por categorías según la tendencia sea creciente, decreciente o constante.
- **Cuantitativas discretas**: DL, DS, DP, DR, UC, Nmax y Nzeros. Aunque por definición sólo las dos últimas son estrictamente discretas, dado que las otras 5 variables se definen como el número de veces que se repite X evento por segundo, y esto por definición podría ser una variable continua si se toman valores reales, en este caso parece que se han aproximado siempre al valor natural más cercano, y por tanto podemos tratarlo como variables discretas, viendo que para más de 2000 entradas, todas tienen menos de 20 valores posibles diferentes. Incluso podría incluirse AC que tiene solo 22 casos distintos.
- **Continuas**: Por definición, a parte de las cinco anteriores mencionadas que podrían considerarse continuas aunque en este caso no se comporten como tal, tenemos las restantes: b, e, LBE, FM, ASTV, MSTV, ALTV, MLTV, Width, Min, Max, Mode, Mean, Variance y Median.

De las variables explicativas, podemos eliminar DR, dado que únicamente toma un valor para todos los casos, b y e que son datos temporales de inicio y fin de toma de la muestra, y el identificador ID, que es irrelevante para el modelado de datos, no aportando información relevante.

En el caso de nuestra variable objetivo, es importante tener en cuenta que la mayoría de casos son normales, los casos positivos o anómalos solo suceden algo más de ⅕ de las veces, algo que se hará notar en la matriz de confusión final.

Fase 2: Preparación del conjunto de datos

Para preparar el conjunto de datos de entrenamiento y test vamos a emplear la función de la librería sklearn conocida como model_selection.train_test_split. En este caso, tomaremos, como ya hemos dicho, las 17 variables explicativas por un lado y la variable Target por otro, y generaremos dos conjuntos divididos en el 60% para el set de entrenamiento y 40% para el de test, empleando el siguiente código:

```
#Importamos librerías para muestreo de datos
from sklearn.model_selection import train_test_split

#Separamos la variable objetivo de las variables explicativas
X_data = data.drop('Target', axis=1)
Y_data = data['Target']

#Creamos los subconjuntos
X_data_train, X_data_test, Y_data_train, Y_data_test =
train_test_split(X_data,Y_data, test_size=0.4, random_state=1)
```

La variable test_size = 0.4 nos permite controlar la distribución proporcional del subconjunto de test respecto al de entrenamiento. Con estos conjuntos creados, probaremos los siguientes modelos, tanto Naive Bayes como los Support Vector Machine.

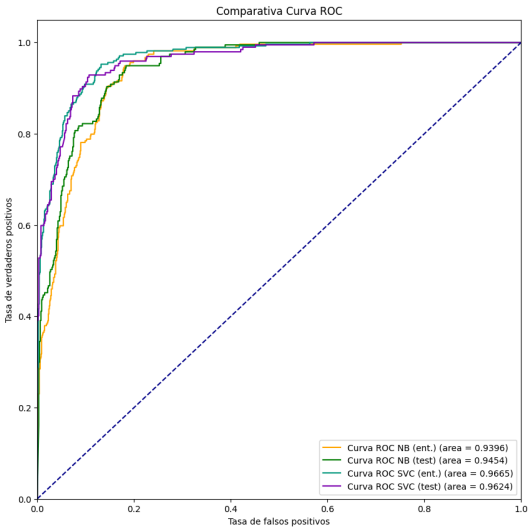
Fase 3: Generación del modelo y testeo

En este ejercicio vamos a probar dos tipos de modelos, por una parte en el caso del algoritmo de Support Vector Machines, al tener una variable objetivo binaria, emplearemos el SVC, o Support Vector Classifier, y por otra emplearemos el Naive Bayes.

Para el primer caso, vamos a emplear la función GridSearchCV para realizar pruebas con diferentes funciones de modelado, o kernels, así como sus respectivos hiperparámetros asociados, de forma que demos con la combinación que mejor sea capaz de ajustar el modelo. En este caso, vamos a probar con un kernel lineal, uno gaussiano y otro polinómico, con C entre 0.1 y 100. Para el caso gaussiano emplearemos una gamma de 0.001 o 0.0001, y en el caso polinómico, probaremos con polinomios de segundo, tercer y cuarto grado.

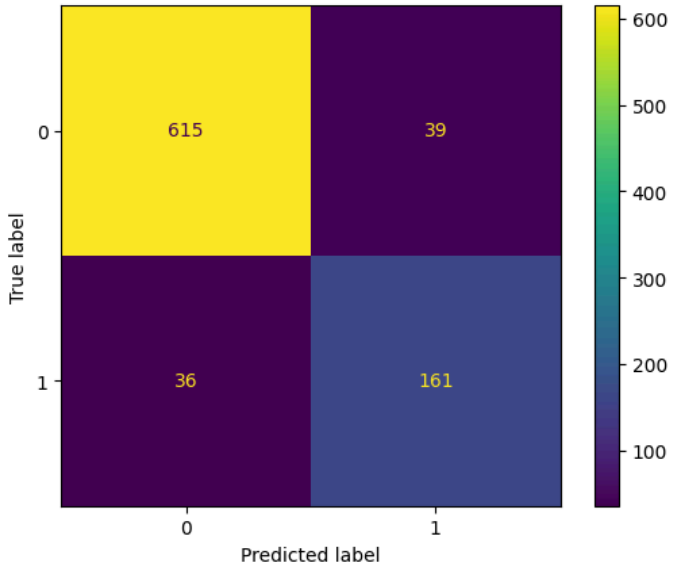
	mean_test_score	mean_train_score	params	rank_test_score
7	0.966992	0.992736	{'C': 100, 'gamma': 0.0001, 'kernel': 'rbf'}	1
5	0.961776	0.979311	{'C': 10, 'gamma': 0.0001, 'kernel': 'rbf'}	2
10	0.960807	0.967642	{'C': 10, 'kernel': 'linear'}	3
9	0.959996	0.968022	{'C': 1, 'kernel': 'linear'}	4

Vemos que el mejor resultado para el SVC que se ha probado empleando el grid es para el kernel gaussiano, con un área bajo la curva en el test de 0.967, sin embargo se aprecia un notable sobreajuste en el conjunto de test, dado que el área bajo la curva para este caso es de 0.993. Por ello tomamos como modelo válido el del kernel lineal con C = 10.



Por otra parte, para el ajuste del modelo de Naive Bayes solo hemos considerado el modelo Gaussiano, dado que los modelos alternativos, como el de Bernoulli o el multimodal, están preparados para trabajar con datos categóricos, no con variables continuas, como es el caso que nos ocupa. En este caso, el resultado del Naive Bayes es inferior al del SVC, obteniendo un área bajo la curva de 0.945, por lo que tomaremos por válido el modelo de SVC.

Fase 4: Validación del modelo



Tomando el modelo SVC como el mejor, podemos calcular que presenta una precisión de 0.911, frente a 0.888 del NB.

De la matriz de confusión que arroja el modelo somos capaces además de extraer las siguientes métricas:

- Sensibilidad: 81.7%
- Especificidad: 94.0%

Como vemos, la sensibilidad no es tan buena como podría ser, aunque la especificidad sí es bastante elevada. Si bien hemos tomado un modelo que reducía la métrica del área bajo la curva en pos de evitar el sobreentrenamiento, no creo que esto haya empeorado el modelo. Realmente este defecto en la sensibilidad puede explicarse con el hecho, como venía diciendo al principio, que la anomalía fetal se presenta en tan solo un 22% del total de los casos, por lo que sabiendo que tan solo contamos con 2126 entradas de datos, podrían ser demasiado pocas para elaborar un modelo lo suficientemente bueno. Tal vez con una muestra mayor la capacidad predictora mejorase.