Clasificación desbalanceada

Usuario: ItaloGarleniRodriguez

Autor: Ítalo Garleni Rodríguez

Contenido

Descripción del dataset

El objetivo de este estudio es obtener el mejor modelo de clasificación mediante el uso de herramientas de preprocesamiento para datasets desbalanceados. Para ello, se posee de un conjunto de datos clasificados por una variable binaria, una de ellas con un número reducido de individuos. El resto de variables que se utilizarán para predecir la variable clase son todas numéricas. Por otra parte, se posee un total desconocimiento de estas variables, ya que no existe documento que las describa.

Volviendo al punto anterior, el desbalanceo de la variable clase se produce sobre el valor 0, el cual posee un individuo por cada 1.93 individuos de la clase 1. Será necesario aplicar técnicas de desbalanceo para poder obtener modelo clasificador más fiable.

Preprocesamiento y Clasificación

Selección de variables

Para la selección de variables se utilizará una combinación de métodos de la librería FSelector. Todos ellos arrojan un resultado, o una puntuación sobre cada variable, la cual es útil para indicar cuánto de fiable es dicha variable para predecir la clase. Basta con escoger aquellos métodos que se aplican sobre variables numéricas, para luego hacer una media de puntuaciones y obtener un resultado final. El resultado obtenido es el de la imagen situada a la derecha.

	attr_importance
MATHEFF	0.8084467
ESCS	0.5597442
ANXMAT	0.5579311
SCMAT	0.4792346
misced	0.4279869
SMATBEH	0.3597843
fisced	0.3201113
CLCUSE1	0.3102301
INTMAT	0.3067937
ST15Q01	0.2996359
FAILMAT	0.2818227
ST19Q01	0.2662365
MATINTFC	0.2617183
SUBNORM	0.2379558
ST04Q01	0.2359130
INSTMOT	0.2209660
STUDREL	0.2133856
FAMSTRUC	0.2000000
BELONG	0.2000000
ATTLNACT	0.2000000
ATSCHL	0.2000000
MATWKETH	0.2000000

Como se puede observar, existen variables que tienen mayor influencia que otras, por lo que será interesante comparar modelos con todas las variables y otros con un número de variables menor. De este modo, se podrá obtener modelos más sencillos con una tasa de acierto alta.

También cabe la posibilidad de añadir interacciones entre las variables, pero como este estudio se va a centrar en la gestión del desbalanceo, solo se introducirá dos variantes a la selección de variables: Una interacción simple y otra multiplicativa.

```
> formulaClassAll
PV1MATH ~ .
> formulaClassSelected
PV1MATH ~ MATHEFF + ESCS + ANXMAT + SCMAT + misced + SMATBEH +
    fisced + CLCUSE1 + INTMAT + FAILMAT
> formulaClassSelectedMultiplying
PV1MATH ~ MATHEFF * ESCS * ANXMAT * SCMAT * misced * SMATBEH *
    fisced * CLCUSE1 * INTMAT * FAILMAT * ST15Q01
```

Tabla de pruebas

Se procede a probar diferentes tipos de combinaciones, con el objetivo de mejorar el AUC obtenido en la plataforma KAGGLE. Decir que éstos no son todos los test subidos a la plataforma, ya que los test previos a éstos fueron realizados mediante predicciones definidas (1 y 0) y no mediante probabilidades. El AUC se obtiene a través de una media de los AUC de un 10 cross fold validation.

Con los resultados obtenidos en los test de predicciones exactas, se intuyó un AUC aproximado de los modelos testados con ACCURACY, por lo que en estos test se han repetido con el método de probabilidades, son los mejores test obtenidos previamente con ACCURACY (por falta de tiempo, tenía la semana planeada ya y no podía dedicarle más tiempo a la competición). Aun así, se puede observar que se realizaron numerosos test para obtener el mejor modelo.

Dicho esto, la tabla de resultados es la siguiente:

N.º Test	**Información del clasificador**	**AUC 10CFV**	**AUC KAGGLE (public)**	**AUC KAGGLE (private)**
16	SVM + OSS (IR 1.583333), All variables	0.9100755	0.82607	0.82499
17	SVM + Tomek (IR 1.586996), All variables	0.9186421	0.82342	0.81927
18	SVM + Tomek (IR 1.428571), All variables	0.9315692	0.81818	0.81701
19	SVM + OSS (IR 1.56685), All variables		0.82631	0.82522
20	SVM + Tomek (IR 1.586996), Simple variable selection	0.9023085	0.81833	0.80576
21	SVM + Tomek (IR 1.586996), Product variable selection	(takes too long)	0.76495	0.75927
22	RandomForest + Tomek (IR 1.586996), Product variable selection	0.8531477	0.72226	0.72389
23	SVM + Tomek (IR 1.586996), type = "C-svc" kernel = "rbfdot" C = 1,	0.9191841	0.79567	0.77592

	Simple variable selection			
24	Boosting (maxdepth = 2) (mfinal = 10) + Undersampling (perc = 38.69) (IR 1.584249), All variables	-	0.79754	0.80243
25	SVM + Undersampling (perc = 38.69) (IR 1.584249), type = "C-svc" kernel = "rbfdot" C = 1, All variables	-	0.82467	0.81895
26	SVM + Tomek (IR 1.232601), type = "C-svc" kernel = "rbfdot" C = 1, Simple variable selection	-	0.82467	0.81895
27	SVM + Tomek (IR 1.232601), type = "C-svc" kernel = "rbfdot" C = 1, All variables	-	0.80667	0.81140
28	RandomForest + Tomek (IR 1.586996), 100 trees, All variables	0.8703026	0.72000	0.73867
29	SVM + Undersampling (perc = 50) (IR 1.564103) and Oversampling (IR 1), Type = "C-svc" kernel = "rbfdot" c = 1, All variables	-	0.81617	0.81448
30	SVM + OSS (IR 1.586996), Type = "C-svc" Kernel = "polydot" c = 1, All variables	0.8714406	0.82745	0.82530

31	SVM + OSS (IR 1.586996), type = "C-svc" kernel = "polydot" c = 1.1, All variables	0.8715248	0.82745	0.82530
32	SVM + OSS (IR 1.586996), type = "C-svc" kernel = "polydot" c = 2, All variables	0.8718397	0.82744	0.82528
33	SVM + OSS (IR 1.586996), type = "C-svc" kernel = "polydot" c = 0.5, All variables	0.86971	0.82742	0.82522

Como se puede observar, el mejor modelo obtenido es SVM con el kernel "polydot" o polinómico. Sin embargo, los demás modelos tienen un resultado bastante aceptable.

También se ha comprobado que los mejores modelos de desbalanceo son OSS y Tomek, los cuales arrojan resultados muy similares. Para ello, se ha probado a realizar Undersampling con un Perc = 38.69 para obtener un ratio de desbalanceo similar al de los algoritmos mejores (IR = 1.586996). Sin embargo, no se ha mejorado el resultado.

Por otra parte, también se ha experimentado con la selección de variables, probando las tres posibles fórmulas nombradas en el apartado anterior. El AUC es bastante similar, pero más bajo con la selección de variable. Por tanto, optar por un modelo más simple puede ser una opción si lo que se busca es reducir las variables de entrada del modelo.

Funcionamiento de los R-Scripts

Los scripts incluidos en la entrega se dividen en varios elementos, los cuales cada uno tiene su función definida. El script principal se denomina "analyzer.R", el cual contiene el proceso de ejecución que se ha llevado a cabo para obtener los resultados, además de una batería de test individuales. También se incluyen de forma comentada las ejecuciones de los modelos anteriores a la modificación del tipo de medida de rendimiento (accuracy → AUC).

Los demás scripts contienen funciones de clasificación, en las que se realiza el 10cfv y la correspondiente generación de la predicción del test. Finalmente, en el script principal existe una sección para guardar dicha predicción en un fichero csv.