PRACTICA 3

Jaime Lorenzo Sánchez

5 de mayo de 2022

Índice general

Capítulo 1

Introducción

El objetivo de esta práctica es introducir los conceptos de clasificación en presencia de datos con desequilibrio de clases. En esta práctica se va a estudiar el funcionamiento de los diferentes métodos de gestión del desequilibrio de clases.

Debido a su sencillez es fácil la implementación, aunque también es posible usar métodos ya incluidos en scikit-learn.

Seleccione un número suficiente de problemas de los repositorios usuales (UCI MLR, Weka, etc.). Se deben escoger varios problemas de 2 clases con diferente nivel de desequilibrio de clases.

Elija al menos relaciones clase mayoritaria/minoritaria de 2:1

Los problemas a utilizar son los siguientes:

- diabetes.csv
- balance-scale.data
- abalone.data

1.1. Ejercicio 1

Elija tres métricas de las estudiadas en teoría para evaluar el rendimiento de los métodos en problemas con desequilibrio de clase. Elija, al menos, dos clasificadores diferentes. En primer lugar vamos a comparar el efecto del desequilibrio de clases en estos dos clasificadores. Estudie el error con las métricas escogidas y obtenga las conclusiones que observe sobre el efecto del desequilibrio.

Dado que se está realizando un estudio de 3 problemas con desequilibrio de clases, primero obtenemos el desequilibrio de clase de cada problema al aplicar el entrenamiento del dataset. Dicho desequilibrio se muestra en la tabla Ejercicio1:

Para la realización del ejercicio, hemos elegido los siguientes clasificadores:

- RandomForestClassifier
- SVC

Para la realización del estudio, se han utilizado las siguientes 3 métricas:

- Hamming Loss
- Accuracy Store
- f1_score
- Error

Al aplicar las métricas y cada clasificador a cada dataset, obtenemos los datos mostrados en las tablas Ejercicio1:SVC y Ejercicio1:RandomForestClassifier.

Si observamos la tabla Ejercicio1:SVC, comprobamos lo siguiente:

- Al aplicar la métrica hamming Loss, comprobamos que el valor obtenido por dicha métrica es muy pequeño con un desequilibrio de datos pequeño. Dicha métrica aumenta su valor al aumentar el desequilibrio de clases hasta un desequilibrio de 527 datos de clase, a partir del cual disminuye el valor del hamming loss.

- Al aplicar la métrica accuracy Store, comprobamos que cuanto menor es el desequilibrio de clases mayor es el valor de la métrica. Esto se produce hasta un desequilibrio de 527 datos de clase, a partir del cual aumenta el valor de la métrica utilizada.

- Al aplicar la métrica f1_score micro, comprobamos que cuanto menor es el desequilibrio de clases mayor es el valor de la métrica. Esto se mantiene hasta un desequilibrio de clase de 527 casos, a partir del cual aumenta el valor de la métrica utilizada.

- Al calcular el error cometido, comprobamos que cuanto mayor es el desequilibrio de clases menor es el error cometido. Esto se mantiene hasta un desequilibrio de clases de 527 casos, a partir del cuál disminuye el error cometido.

Si observamos la tabla Ejercicio1.RandomForestClassifier, comprobamos lo siguiente:

- Al aplicar la métrica hamming Loss, comprobamos que el valor obtenido por dicha métrica es muy pequeño con un desequilibrio de datos pequeño. Dicha métrica aumenta su valor al aumentar el desequilibrio de clases hasta un desequilibrio de 527 datos de clase, a partir del cual disminuye el valor del hamming loss.

- Al aplicar la métrica accuracy Store, comprobamos que cuanto menor es el desequilibrio de clases mayor es el valor de la métrica. Esto se produce hasta un desequilibrio de 527 datos de clase, a partir del cual aumenta el valor de la métrica utilizada.

- Al aplicar la métrica f1_score micro, comprobamos que cuanto menor es el desequilibrio de clases mayor es el valor de la métrica. Esto se mantiene hasta un desequilibrio de clase de 527 casos, a partir del cual aumenta el valor de la métrica utilizada.

- Al calcular el error cometido, comprobamos que cuanto mayor es el desequilibrio de clases menor es el error cometido. Esto se mantiene hasta un desequilibrio de clases de 527 casos, a partir del cuál disminuye el error cometido.

1.2. Ejercicio 2

Estudio de las dos técnicas básicas. Implemente, o use, las técnicas de over-sampling y under-sampling. Aplique estas dos técnicas a los conjuntos de datos seleccionados y estudie cómo se comportan para cada uno de los dos modelos

de clasificación. Indique las ventajas e inconvenientes que observa en cada una de ellas.

Tras aplicar oversampling, obtenemos el siguiente equilibrio de clases:

Al aplicar este método, se produce un equilibrio de clases utilizando el mismo número de datos de clase que el contenido por la clase mayoritaria.

Al aplicar este método, se produce un equilibrio de clases utilizando el mismo número de datos de clase que el contenido por la clase minoritaria.

Realizamos un estudio de las métricas obtenidas y representadas en la tabla Ejercicio2:SVC_OverSampling:

- Al aplicar la métrica Hamming Loss, observamos que cuanto mayor es el número de casos de la clase mayor es el valor obtenido por esta métrica.
- Al aplicar la métrica Accuracy Store y la métrica f1_score, observamos que cuanto mayor es el número de casos de la clase menor es el valor obtenido por dichas métricas.
- Si calculamos el error cometido, observamos que cuanto mayor es el número de casos por clase mayor es el error cometido

Si observamos la tabla Ejercicio2:SVC_UnderSampling, comprobamos lo siguiente:

- Si calculamos la métrica Hamming Loss, observamos que el valor de la métrica obtenida es muy alto cuando tenemos un tamaño de clases muy pequeño. Aunque dicho valor disminuye conforme aumenta el tamaño de las clases, a partir de un tamaño de clases de 268 casos el valor de la métrica aumenta.
- Al calcular las métricas accuracy Store y f1_score observamos que con número de casos de clase muy pequeños, dichas métricas tiene un valor pequeño. Aunque el valor de las métricas aumenta conforme aumenta el tamaño de las clases, a partir de un tamaño de clase de 268 el valor de dichas métricas aumenta.
- Si calculamos el error cometido, observamos que con un tamaño de clases pequeño el error cometido es muy elevado. Conforme aumenta el tamaño de las clases, el

error cometido disminuye, pero a partir de un tamaño de clases de 268 casos el error cometido aumenta.

Si observamos la tabla Ejercicio2:RandomForest_OverSampling, observamos lo siguiente:

- Si aplicamos la métrica Hamming Loss, observamos que el valor de la métrica es menor cuanto menor sea el tamaño de las clases. Sin embargo, a partir del tamaño de clases 576 el valor de la métrica obtenido disminuye.

- Si calculamos las métricas Accuracy Store y f1_score, observamos que con un tamaño de clases de 500 el valor de las métricas es muy elevado. Aunque a partir de este tamaño de clases el valor de las métricas disminuye, a partir de un tamaño de clases de 576 el valor de dichas métricas aumenta.

- Si calculamos el error cometido, observamos que con un tamaño de clases de 500 casos el error cometido es muy pequeño. Aunque a partir de dicho tamaño de clases el error cometido aumenta considerablemente, observamos que a partir de los 576 casos por clase el error cometido disminuye.

Si observamos la tabla Ejercicio2:RandomForest_UnderSampling, comprobamos lo siguiente:

- Si calculamos la métrica Hamming Loss, observamos que el valor de la métrica obtenida es muy alto cuando tenemos un tamaño de clases muy pequeño. Aunque dicho valor disminuye conforme aumenta el tamaño de las clases, a partir de un tamaño de clases de 268 casos el valor de la métrica aumenta.

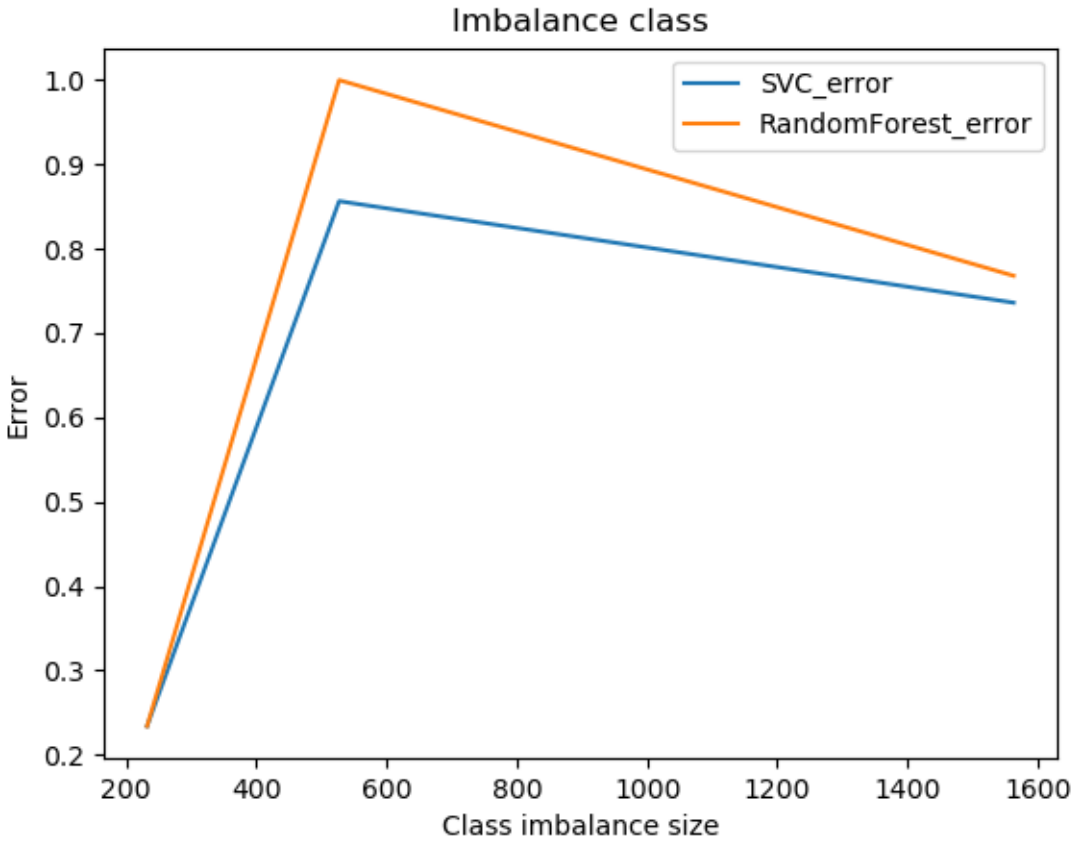
- Al calcular las métricas accuracy Store y f1_score observamos que con número de casos de clase muy pequeños, dichas métricas tiene un valor pequeño. Aunque el valor de las métricas aumenta conforme aumenta el tamaño de las clases, a partir de un tamaño de clase de 268 el valor de dichas métricas aumenta.

- Si calculamos el error cometido, observamos que con un tamaño de clases pequeño el error cometido es muy elevado. Conforme aumenta el tamaño de las clases, el error cometido disminuye, pero a partir de un tamaño de clases de 268 casos el error

cometido aumenta.

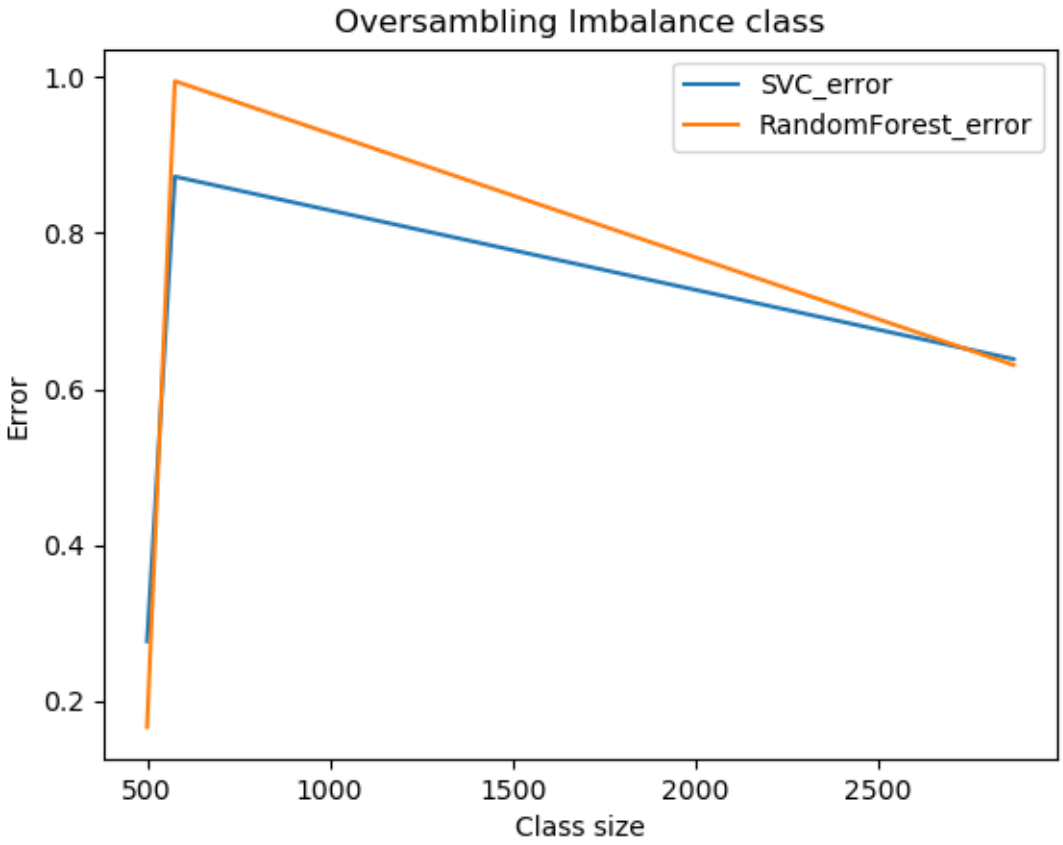
1.3. Ejercicio 4

Compare gráficamente los resultados más significativos de los ejercicios anteriores usando cualquiera de las representaciones gráficas que conozca



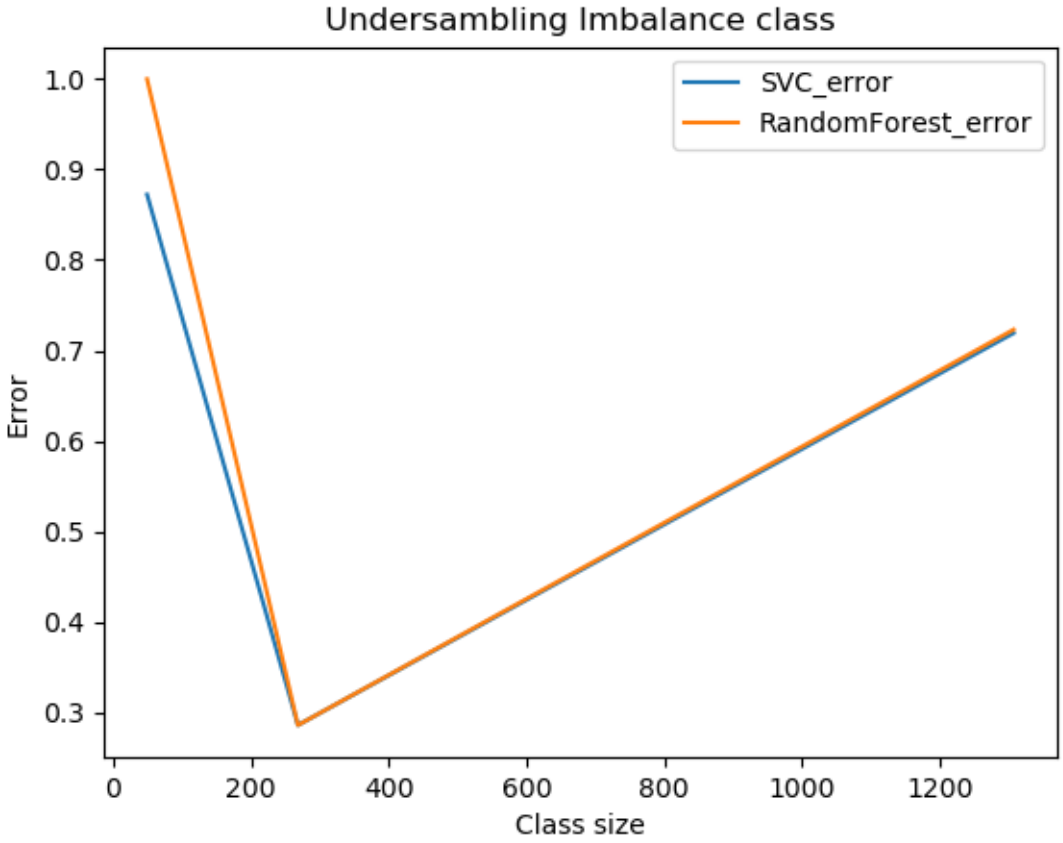
Si observamos el gráfico, comprobamos que al producirse un desequilibrio entre clases el error cometido es mayor cuanto mayor sea dicho desequilibrio.

Podemos comprobar que, en función de los dataset empleados, podemos comprobar que el clasificador SVC ha obtenido mejores resultados que el clasificador RandomForest.



Si observamos el gráfico, comprobamos que si aplicamos oversambling en dataset con clases en desequilibrio el error cometido muy elevado entre los 500 y 1000 de casos por clase. Sin embargo, dicho error disminuye conforme aumenta el tamaño de las clases.

En función de los datasets empleados, podemos comprobar que al aplicar oversambling es mejor utilizar el clasificador SVC para clases de tamaño inferior a 2500; sin embargo, a partir de un tamaño de clases de 2500 casos es mejor utilizar el clasificador RandomForest.



Si observamos el gráfico, comprobamos que si aplicamos undersambling en datasets con clases en desequilibrio el error cometido es muy bajo entre los 200 y 400 de casos de clase, pero a partir de los 400 casos por clase el error cometido aumenta.

En función de los datasets empleados, podemos comprobar que ambos clasificadores son igual de buenos.

Dataset	Clase mayoritaria Numero de casos	Clase minoritaria Numero de casos	Desequilibrio
diabetes.csv	0:500	1:268	232
balance-scale.data	0:576	1:49	527
abalone.data	0:2870	1:1307	1563

Cuadro 1.1: Ejercicio1

Dataset	Desequilibrio	Hamming Loss	Accuracy Store	f1_score	Error
diabetes.csv	232	0.2338	0.7662	0.7662	0.2338
balance-scale.data	527	0.8564	0.1436	0.1436	0.8564
abalone.data	1563	0.7360	0.2640	0.2640	0.7360

Cuadro 1.2: Ejercicio1:SVC

Dataset	Desequilibrio	Hamming Loss	Accuracy Store	f1_score	Error
diabetes.csv	232	0.2337	0.7662	0.7662	0.2338
balance-scale.data	527	1.0	0.0	0.0	1.0
abalone.data	1563	0.7679	0.2321	0.2321	0.7679

Cuadro 1.3: Ejercicio1:RandomForestClassifier

Dataset	Clase mayoritaria Numero de casos	Clase minoritaria Numero de casos	Desequilibrio
diabetes.csv	0:500	1:500	0
balance-scale.data	0:576	1:576	0
abalone.data	0:2870	1:2870	0

Cuadro 1.4: Ejercicio2.OverSambling

Dataset	Clase mayoritaria Numero de casos	Clase minoritaria Numero de casos	Desequilibrio
diabetes.csv	0:268	1:268	0
balance-scale.data	0:49	1:49	0
abalone.data	0:1307	1:1307	0

Cuadro 1.5: Ejercicio2.UnderSambling

Dataset	TamanoClases	Hamming Loss	Accuracy Store	f1_score	Error
diabetes.csv	500	0.2767	0.7233	0.7233	0.2767
balance-scale.data	576	0.8723	0.1277	0.1277	0.8723
abalone.data	2870	0.6384	0.3616	0.3616	0.6384

Cuadro 1.6: Ejercicio2:SVC_OverSampling

Dataset	TamanoClases	Hamming Loss	Accuracy Store	f1_score	Error
diabetes.csv	268	0.34783	0.6522	0.6522	0.2857
balance-scale.data	49	0.8723	0.1277	0.1277	0.8723
abalone.data	1307	0.7191	0.2808	0.2808	0.7192

Cuadro 1.7: Ejercicio2:SVC_UnderSampling

Dataset	TamañoClases	Hamming Loss	Accuracy Store	f1_score	Error
diabetes.csv	500	0.1667	0.8333	0.8333	0.1667
balance-scale.data	576	0.9947	0.0053	0.0053	0.9947
abalone.data	2870	0.6308	0.3692	0.3692	0.6308

Cuadro 1.8: Ejercicio2:RandomForest_OverSampling

Dataset	TamañoClases	Hamming Loss	Accuracy Store	f1_score	Error
diabetes.csv	268	0.2857	0.7143	0.7143	0.2857
balance-scale.data	49	1.0	0.0	0.0	1.0
abalone.data	1307	0.7191	0.2770	0.2770	0.7230

Cuadro 1.9: Ejercicio2:RandomForest_UnderSampling