PRACTICA KAGGLE

MD: clasificación y preprocesamiento

Carlos Jesús Fernández Basso
karloos@correo.ugr.es

Contenido

Conjunto de datos

El conjunto de datos a examinar contiene 51 variables de ellas tendremos que predecir en el conjunto test el calor de la variable *class*. Este conjunto contiene varios problemas como son: ruido, valores perdidos y que las clases están desbalanceadas. Para mejorar esto he utilizado diferentes técnicas de preprocesamiento y clasificación obteniendo mejoras sobre la clasificación.

Preprocesamiento

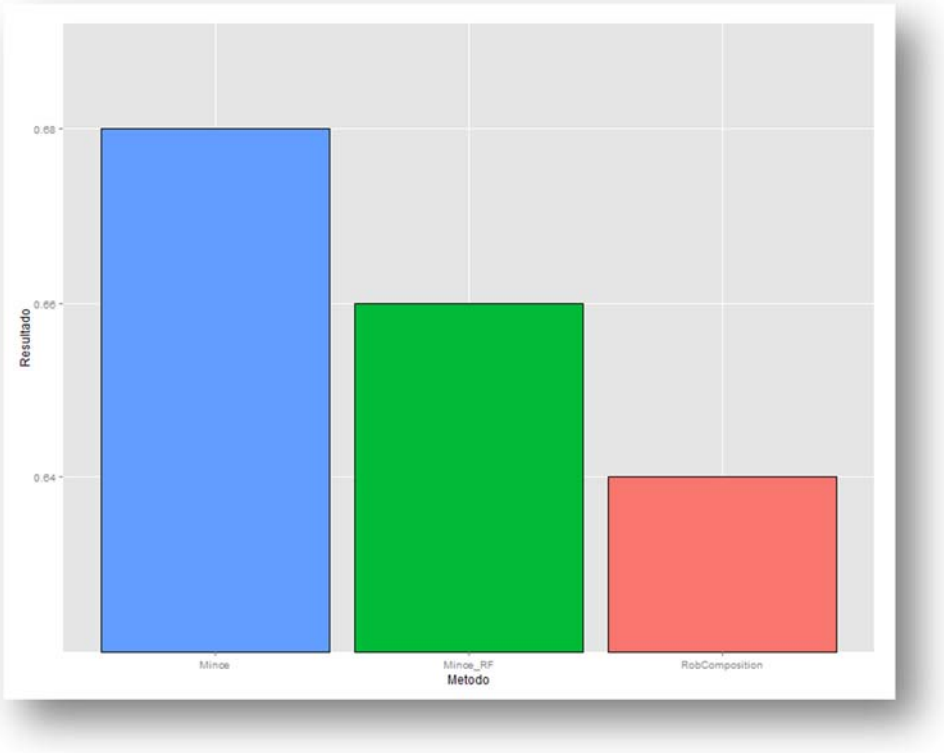
Antes de realizar el modelo para predecir la clase de los elementos del conjunto test, he realizado diferentes métodos de preprocesamiento de los datos. De ellos algunos me han mejorado los resultados mientras con otros no tuve mejora.

Imputación de datos

Lo primero que hice con estos datos fue la imputación de los valores perdidos, pues si omitía estos regisros los resultados de la clasificación mediante C4.5 eran pesimos.

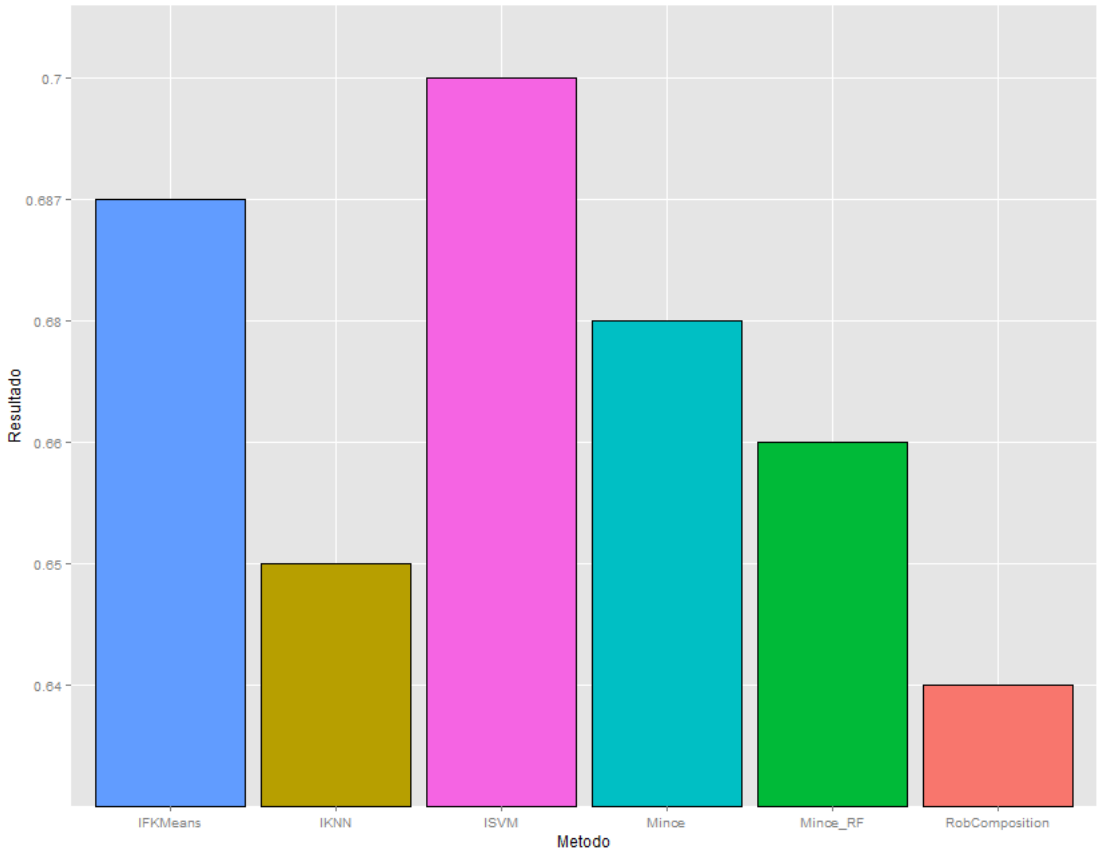
Para la imputación de los datos utilice varios métodos, en cuanto a R utilice la imputación mediante *RobComposition*, *Mice* y *Mice* mediante ramdonForest.

Utilizando estas imputaciones y el clasificador randomforest obtuve los siguientes resultados:



Como se puede observar de estos métodos de imputación, el que nos proporcionaba mejor resultado a la hora de clasificar mediante ramdomforest era *MInce*. Ademas de estos métodos utilice métodos que existían en la herramienta Keel. De ellos utilice varios para ver si

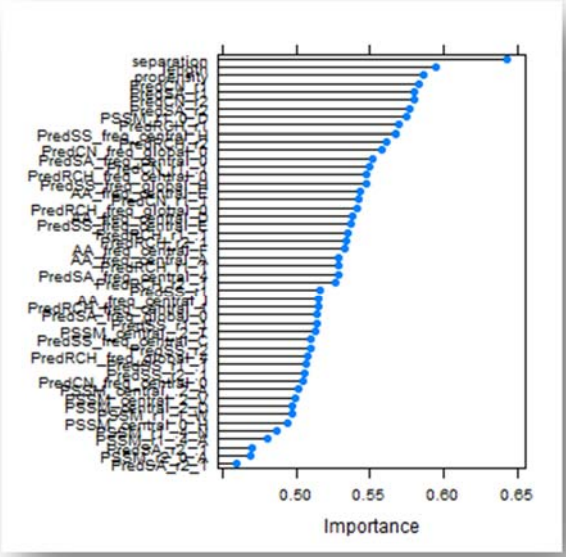
eran mejores que estos anteriores o no. De los métodos de Keel probé el *IFKMeans*, *ISVM*, *IKNN*.



Observando estos resultados vi que la imputación mediante SVM era buena para realizar las siguientes pruebas, en las que realizaría además de la imputación otros métodos de preprocesamiento.

Selección de características

Para la selección de las mejores características utilice los paquetes que vimos en clase. Los mejores resultados utilizando seleecion de características fueron con el paquete boruta, el paquete caret y el FSelector (RandomForest). Aquí acontinuacion vemos las caracterictas mas importantes con el paquetec caret:



Mediante esta información realice un randomforest con 40 variables (las mejores según caret) y el resultado no mejoro aunque tanpoco fue peor. En cambio al utilizar la información del paquete Boruta, utilice 43 variables y el resultado me mejoro un 0.014.

Balanceo

Para el balanceo en mis primeras pruebas realice una selección del cardinal de elementos en la clase minoritaria en la clase mayoritaria. Quedando así las clases balanceadas. Esto me produjo mejoras mediante el RamdomForest, pues pase de 0.701 a 0.715.

Además de este método, utilice el algoritmo *SMOTE* con el cual con los parámetros por defecto no tuve mejora sobre el método anterior pero al cambiarle algunos paramtros si obtuve 0.72.

Reducción de ruido

En la reducción de ruido utilice el algoritmo IPF implementado en clase y el IPF de la herramienta Keel. Utilizando el algoritmo implementado en keel , losresultados eran malos pues est algoritmo eliminaba muchos elementos de la clase minoritaria. Por ellos realice algunos cambios en este algormitmo.

- Cambie el método de aprendizaje a Bagging y Boosting (para liadian con el no balanceo)
- Cambie el parámetro de cuantas vaces debe fallar para eliminar el dato

Outliers

Realice una eliminación de datos anómalos. Esto lo realice mediante la función *cerioli2010,* mediante una función de clustering. El mejor resultado me ocurrió al utilizar *cerioli2010* justo antes de clasificar. El método iterativo de esta función me eliminaba 3400 instancias del conjunto.

Clasificadores

Se han probado varios clasificadores obteniendo mejores resultados en algunos de ellos. Los primeros intentos se probó con árboles simples

C4.5

Este clasificador solo lo utilice al principio para realizar las primeras pruebas.

GLM (regresión lineal logística)

Utilice este modelo de clasificación pues kaggle acepta probabilidades. Por ello al utilizar las probabilidades y calcular ROC mejoran significativamente los resultados.

RamdonForest

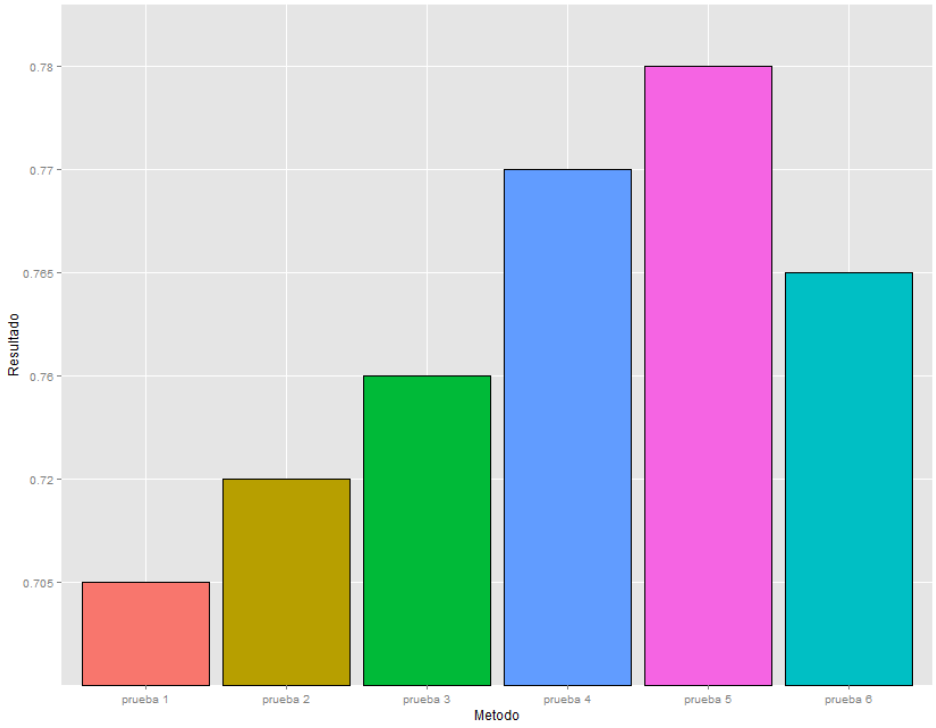
Es el primer método que utilice después del C4.5 , este mejoro los resultados , aunque al utilizar probabilidades el resto mejoro sobre él.

Boosting y Bagging

Utilice estos dos métodos antes de balancear las clases, pues son más robustos al no balanceo de las clases. De estos dos el que mejores resultados me promedio fue el Bagging.

Pruebas

Realice multitud de pruebas por el problema de que el cambio del orden en las técnicas cambiaba mucho el resultado por ejemplo:

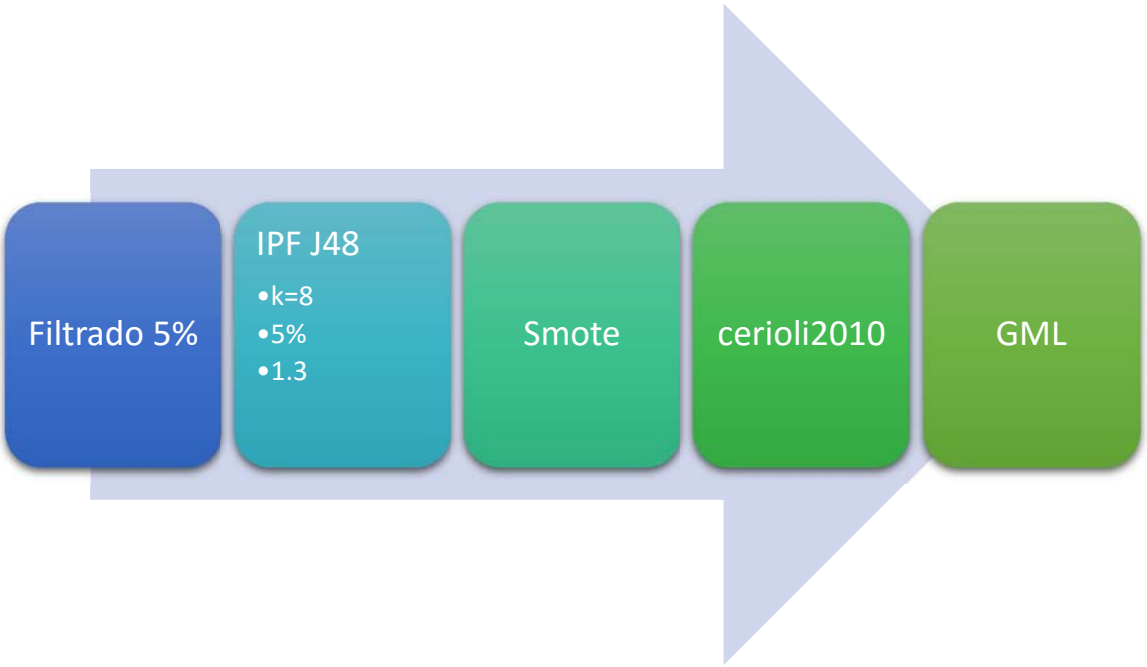


1. Prueba 1:En esta utilice ipf y luego impute los datos, luego smote
2. Prueba 2:En esta impute datos y luego Smote
3. Prueba 3:En esta utilice ipf y luego impute los datos, luego smote

4. Prueba 4:En esta impute los datos luego ipf después elimine ouliers
5. Prueba 5: En esta impute los datos luego ipf después, utilice smote y elimine ouliers
6. Prueba 6:En esta impute los datos,luego smote, ipf y outliers

Mejor solución

Mi mejor solución se realizó de la siguiente manera y orden:



Obteniendo como resultado 0.78133