

Pump it Up. Equipo CharlaTED

Ignacio Aguilera Martos (jusuario¿, KNN), Luis Balderas Ruiz (luisbalru, RIP-PER), Francisco Luque Sánchez (jusuario¿, jalgoritmo¿), Iván Sevillano García isega24, SVM

18 de febrero de 2020

Preprocesamiento y Clasificación

1. SVM
2. RIPPER
3. KNN
4. J48

SVM

- Eliminación de variables.
 - *region, recorded_by, num_private,...*
 - Variables categóricas con más de 100 valores.

Preprocesamiento para SVM

- Eliminación de variables.
 - *region*, *recorded_by*, *num_private*,...
 - Variables categóricas con más de 100 valores.
- Detección de valores perdidos(NA).
 - *population* = 0??
 - *construction_year* = 0??
 - Valores vacíos.

Preprocesamiento para SVM

- Eliminación de variables.
 - *region*, *recorded_by*, *num_private*,...
 - Variables categóricas con más de 100 valores.
- Detección de valores perdidos(NA).
 - *population* = 0??
 - *construction_year* = 0??
 - Valores vacíos.
- Cambiar tipos de dato.
 - *region_code*/*district_code* a factor.
 - *date* a numérico.

IPF(Iterative partitioning filter)

- Split the current training dataset E into n equal sized subsets.
- Build a classifier with the C4.5 algorithm over each of these n subsets and use them to evaluate the whole current training dataset E .
- Add to A the noisy examples identified in E using a voting scheme (consensus or majority).
- Remove the noisy examples.

Sáez, J. A., Luengo, J., Stefanowski, J., Herrera, F. (2015).

SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering.

Information Sciences, 291, 184-203.

Imputación de valores perdidos

- Amelia para valores perdidos numéricos(método iterativo de imputación).

Honaker, J., King, G., Blackwell, M. (2011). Amelia II: A program for missing data. Journal of statistical software, 45(7), 1-47.

- Imputación mediante KNN de valores categóricos(modas).

Zhang, S. (2012). Nearest neighbor selection for iteratively kNN imputation. Journal of Systems and Software, 85(11), 2541-2552.

PCA para variables dumificadas. Selección de subespacios relevantes

- One hot encoding. Para cada valor las variables categórica, creamos una variable que valdrá 1 si la instancia tiene este valor.
- Normalizamos las variables.
- Aplicamos PCA y nos quedamos con las variables cuya desviación típica sea mayor de un umbral(0.00001).
- Pasamos de 208 variables a 173.

Aplicación de SMOTE para balancear clases

El método SMOTE genera instancias de la clase minoritaria, sobrerrepresentandola.

Flujo de información

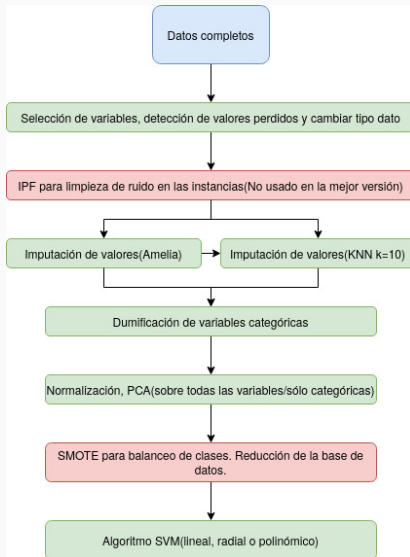


Figura 1: Verde: Utilizado en el mejor modelo

Puntuación a lo largo del tiempo de SVM

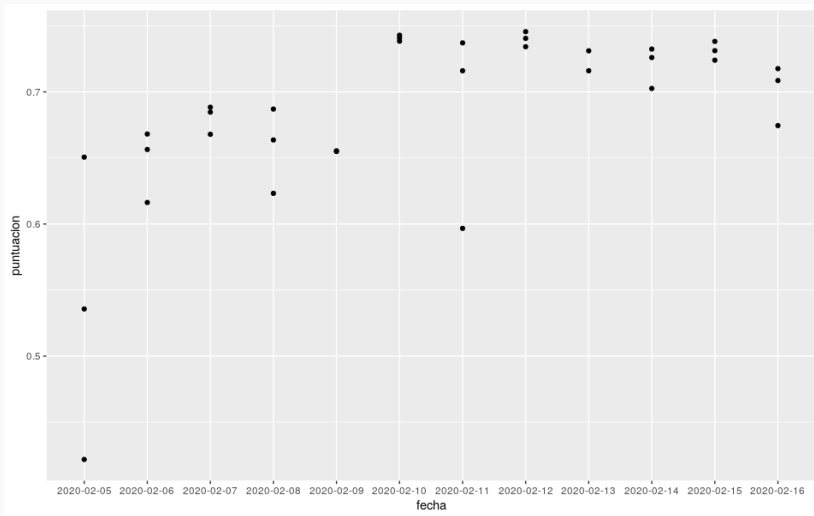


Figura 2: Máxima puntuación: 74.52 %, 12 de febrero

RIPPER

Algoritmos y técnicas utilizados

- **Ingeniería de características. Selección y creación de características**
 - Selección de variables semánticamente representativas.
 - LasVegas Wrapper.
 - Imputación de valores perdidos con media o mediana.
 - Creación de características.

Algoritmos y técnicas utilizados

- **Ingeniería de características. Selección y creación de características**
 - Selección de variables semánticamente representativas.
 - LasVegas Wrapper.
 - Imputación de valores perdidos con media o mediana.
 - Creación de características.
- **Técnicas basadas en instancias. Selección, eliminación de ruido, sobremuestreo**
 - ENN.
 - IPF.
 - SMOTE. Oversampling
 - Random Undersampling.

Algoritmos y técnicas utilizados

- **Ingeniería de características. Selección y creación de características**
 - Selección de variables semánticamente representativas.
 - LasVegas Wrapper.
 - Imputación de valores perdidos con media o mediana.
 - Creación de características.
- **Técnicas basadas en instancias. Selección, eliminación de ruido, sobremuestreo**
 - ENN.
 - IPF.
 - SMOTE. Oversampling
 - Random Undersampling.
- **Ajuste de hiperparámetros.**
 - Gridsearch de parámetros F (número de folds), N (mínimo peso de instancias) y O (número de ejecución para optimizar).
 - 5-CV.

LasVegas Wrapper

- Implementación en R (FSinR) que no consigue terminar una ejecución completa.

LasVegas Wrapper

- Implementación en R (FSinR) que no consigue terminar una ejecución completa.

ENN, IPF

- Limpieza de ruido empeora los resultados tanto en validación cruzada como test. Para ciertas configuraciones, incluso hacen desaparecer la clase minoritaria.

SMOTE, Oversampling. Random Undersampling

- Necesidad de 'numerizar' los datos. Creación de variables dummies.

SMOTE, Oversampling. Random Undersampling

- Necesidad de 'numerizar' los datos. Creación de variables dummies.
- Las técnicas de oversampling generan datos demasiado artificiales.
En general, tienen accuracy cercano al 58 %.

Imputación de valores perdidos o mal escritos

SMOTE, Oversampling. Random Undersampling

- Necesidad de 'numerizar' los datos. Creación de variables dummies.
- Las técnicas de oversampling generan datos demasiado artificiales. En general, tienen accuracy cercano al 58 %.

Imputación de valores perdidos o mal escritos

- *Funder*: factor de 2141 niveles.
- *Installer*: factor de 2411 niveles.
- Muchos de esos niveles son resultado de escribir mal o de distintas formas la misma palabra.
- Reducción de los niveles hasta aproximadamente 500 corrigiendo los nombres, con peor precisión como resultado.

Selección de características: Muchas columnas parecen inútiles

- Elimino las variables *wpt-name*, *subvillage*, *ward*, *recorded-by*, *scheme-name*, *num-private*, *region-code*, *quantity-group*, *source-type*, *waterpoint-type-group*, *payment-type* y *extraction-type-group*
- Imputo valores perdidos en *funder*, *installer*, *permit*, *scheme-management*, *public-meeting*, *gps-height*, *extraction-type*.
- Imputación de la variable *construction-year* con los resultados de la validación cruzada en entrenamiento del mejor algoritmo hasta el momento (KNN) para las instancias de test. Para training, para cada categoría, tomo la media de las instancias que pertenecen a dicha categoría y no tienen valor perdido ($\neq 0$).

- Ripper (JRip de RWeka) con variables *latitude, longitude, date-recorded, basin, lga, funder, population, construction-year, gps-height, public-meeting, scheme-name, permit, extraction-type-class, management, management-group, payment, quality-group, quantity, source, source-type, source-class* y *waterpoint-type*.
- Hiperparámetros $F = 2$, $N = 3$, $O = 29$.

Resultados

- 27 subidas. Mejor resultado: 0.7869.
- Mejor posición: 1752. Posición actual (17/02): 1834

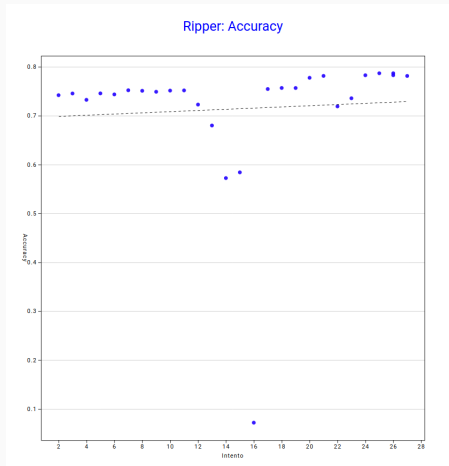
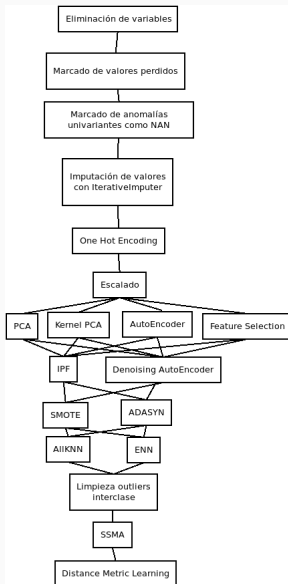


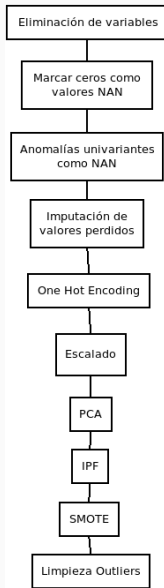
Figura 3: Gráfica de resultados

KNN

Pipeline empleado



Pipeline con mejor resultado



Eliminación de variables

Elimino las variables wpt_name, subvillage, scheme_name, funder, installer, ward, amount_tsh y num_private.

Explicación de las técnicas

Eliminación de variables

Elimino las variables wpt_name, subvillage, scheme_name, funder, installer, ward, amount_tsh y num_private.

Marcado de anomalías como valores perdidos

En cada columna se calcula la media y la desviación típica. Aquellos datos que se salgan del intervalo $[media - 5std, media + 5std]$ se marcan como NAN.

Explicación de las técnicas

Eliminación de variables

Elimino las variables wpt_name, subvillage, scheme_name, funder, installer, ward, amount_tsh y num_private.

Marcado de anomalías como valores perdidos

En cada columna se calcula la media y la desviación típica. Aquellos datos que se salgan del intervalo $[media - 5std, media + 5std]$ se marcan como NAN.

Imputación iterativa

Empleamos una imputación iterativa sobre los valores perdidos.

Explicación de las técnicas

Eliminación de variables

Elimino las variables `wpt_name`, `subvillage`, `scheme_name`, `funder`, `installer`, `ward`, `amount_tsh` y `num_private`.

Marcado de anomalías como valores perdidos

En cada columna se calcula la media y la desviación típica. Aquellos datos que se salgan del intervalo $[media - 5std, media + 5std]$ se marcan como NAN.

Imputación iterativa

Empleamos una imputación iterativa sobre los valores perdidos.

PCA

Aplicamos PCA pero sólo sobre las columnas categóricas. El objetivo es explicar las variables categóricas mejor que en su codificación original. Reducimos a 44 variables todas las categóricas.

IPF

Ejecutamos un IPF para limpiar el ruido con 4 iteraciones.

IPF

Ejecutamos un IPF para limpiar el ruido con 4 iteraciones.

SMOTE

Hacemos un oversampling de las clases "functional needs repair" y "non functional" a 7500 y 22000 con respecto a 23500 de la clase "functional" con $k = 7$.

Explicación de las técnicas

IPF

Ejecutamos un IPF para limpiar el ruido con 4 iteraciones.

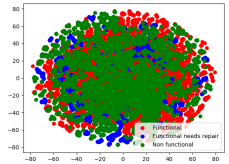
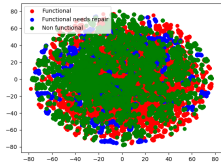
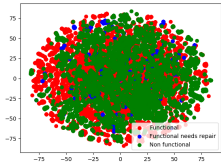
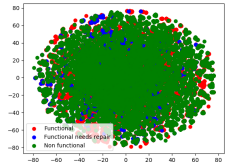
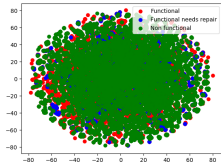
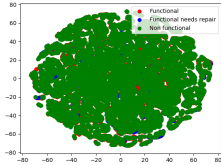
SMOTE

Hacemos un oversampling de las clases "functional needs repair" y "non functional" a 7500 y 22000 con respecto a 23500 de la clase "functional" con $k = 7$.

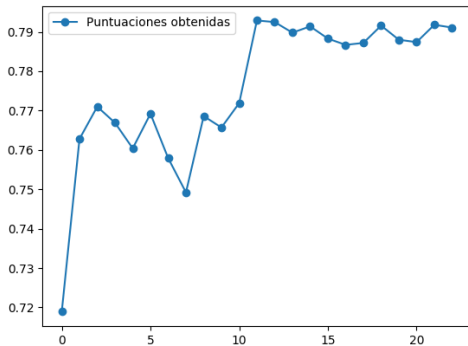
Limpieza Outliers

Hacemos una limpieza de anomalías por cada clase eliminando el 1% más anómalo según KNN con $k = 7$ y la métrica de la mayor distancia.

Visualización de las técnicas



Posición en DrivenData














Puntuación final obtenida: 79.29 %



Ranking final: 1729

Número de subidas: 23

J48

-  *An experiment with the edited nearest-neighbor rule*, IEEE Transactions on Systems, Man, and Cybernetics **SMC-6** (1976), no. 6, 448–452.
-  Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer, *Smote: Synthetic minority over-sampling technique*, J. Artif. Intell. Res. (JAIR) **16** (2002), 321–357.
-  Salvador García María J del Jesus Francisco Herrera David Charte, Francisco Charte, *A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines*.
-  Haibo He, Yang Bai, Edwardo Garcia, and Shutao Li, *Adasyn: Adaptive synthetic sampling approach for imbalanced learning*, 07 2008, pp. 1322 – 1328.

-  I.T. Jolliffe and Springer-Verlag, *Principal component analysis*, Springer Series in Statistics, Springer, 2002.
-  *Python distance metric learning*.
-  *Python outlier detection*.
-  Francisco Herrera Salvador García, José Ramón Cano, *A memetic algorithm for evolutionary prototype selection: A scaling up approach*.
-  José A. Sáez, Mikel Galar, Julián Luengo, and Francisco Herrera, *Inffc: An iterative class noise filter based on the fusion of classifiers with noise sensitivity control*, Information Fusion **27** (2015).
-  Kyuseok Shim Sridhar Ramaswamy, Rajeev Rastogi, *Efficient algorithms for mining outliers from large data sets*.
-  Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller, *Nonlinear component analysis as a kernel eigenvalue problem*, 1996.

-  Stef van Buuren and Karin Groothuis-Oudshoorn, *mice: Multivariate imputation by chained equations in r*, Journal of Statistical Software, Articles **45** (2011), no. 3, 1–67.
-  Dennis L. Wilson, *Asymptotic properties of nearest neighbor rules using edited data*, IEEE Trans. Systems, Man, and Cybernetics **2** (1972), 408–421.

¿Preguntas?