Análisis de Datos III
Reporte Réplica de Artículo
Prof. Maikol Solís
Est. Jimmy Calvo Monge
II-2021

En el siguiente proyecto se estudia y se intenta replicar los resultados obtenidos en el artículo Liang et al. (2008). El objetivo del éste artículo es desarrollar y comprobar la eficacia de una nueva herramienta para la selección de variables, conocida como FSDD (Feature Selection with Distance Discriminant, por sus siglas en inglés). El algoritmo FSDD trata de calificar a las variables predictivas continuas de un problema de clasificación mediante una métrica denominada discriminante de distancias. Los autores formulan el problema teóricamente y obtienen la métrica a evaluar, en seguida comparan este método de filtro de variables con otros dos: reliefF y mrmrMID, en varios conjuntos de datos y utilizando varios modelos de clasificación. Los resultados presentados por el artículo son prometedores. Aquí replicamos estos experimentos mediante una implementación en python y discutimos los resultados y diferencias con los experimentos originales.

1. INTRODUCCIÓN

Los algoritmos de selección de variables tienen por objetivo recuperar una cantidad específica de atributos que puedan ser de utilidad predictiva para un conjunto de datos de un problema de clasificación o regresión. La cantidad de variables que se desea recuperar usualmente es menor que la cantidad total de variables, ya que una de las motivaciones de estos métodos también es conseguir una reducción de la dimensionalidad. De manera muy general estos algoritmos se pueden separar en dos grupos: los métodos **wrapper** y **filter**, a continuación explicamos de manera superficial de qué se tratan. De acuerdo a Kuhn and Johnson (2013), los métodos wrapper evalúan múltiples modelos usando procedimientos que agregan y/o remueven predictores para encontrar la combinación óptima que maximiza el desempeño del modelo. En esencia, los métodos wrapper son algoritmos de búsqueda que tratan los predictores como las entradas y utilizan el desempeño de un modelo como la salida a ser optimizada. Por otro lado los métodos filter evalúan la relevancia de los predictores sin considerar modelos predictivos.

Los métodos filter usualmente se encargan de dar un peso o **calificación** a cada variable, y luego seleccionan las primeras m variables con mayor calificación (donde m es un entero a especificar por el analista). Esta métrica o calificación se obtiene comparando la variable con otras y con la variable de respuesta. En este trabajo vamos a estudiar métodos de selección de variables que caen dentro de la categoría filter y que trabajan sobre problemas de clasificación.

2. MARCO TEÓRICO

Antes de estudiar la formulación de la métrica FSDD que es propuesta en el artículo de referencia, vamos a presentar un pequeño recuento teórico de dos métodos ya establecidos anteriormente para la selección de variables bajo el modo *filter*, éstos son los algoritmos **ReliefF** y **mrmr-MID**. La razón por la que presentamos estos es porque en la sección de experimentos elalgoritmo FSDD será comparado con éstos dos.

2.1. Algoritmo ReliefF. El algoritmo Relief fue desarrollado por Kira y Rendell en 1992 y su motivación en general consiste en evaluar los atributos viendo qué tan buenos son sus valores para distinguir instancias del conjunto de datos que están muy cerca entre sí. Luego de ésto el algoritmo se modificó para ser más robusto y poder lidiar mejor con valores faltantes y con problemas multiclase. A esta modificación se le denomina el algoritmo ReliefF. A continuación mostramos los pasos de este algoritmo, tomados de Robnik-Sikonja and Kononenko (2003). Un análisis completo tanto teórico como computacional de este método se puede encontrar en Urbanowicza et al. (2018). Los aciertos cercanos a X_i son observación cercanas X_i que están en una clase igual a la clase de X_i. El algoritmo involucra una serie de m repeticiones para mejorar su desempeño.

Algorithm 1 ReliefF

Input: conjunto de observaciones X_i de n atributos, y vector de etiquetas (clases). Cantidad de repeticiones m.
Output: vector W con pesos para cada atributo.
PROCESO:
Inicializar todos los pesos en cero: $W = [0, 0, 0, \cdots, 0]$
for $i = 1 : m$ **do**
    Seleccione al azar una observación X_i
    Encuentre k aciertos $H_\ell, \ell = 1, \cdots k$ más cercanos a X_i
    for Clase $C \neq$ clase(X_i) **do**
        De la clase C encontrar las k observaciones $M_\ell(C), \ell = 1, \cdots, k$ más cercanas a X_j
    for $j = 1 : n$ **do**
        $W[j] = W[j] - \sum_{\ell=1}^{k} \frac{\text{diff}(j,X_i,H_\ell)}{mk} + \sum_{C \neq \text{clase}(X_i)} \left[\frac{P(C)}{1-P(\text{clase}(X_i))} \sum_{\ell=1}^{k} \text{diff}(j, X_i, M_\ell(C)) \right] /(mk)$

Donde las funciones de diferencia son:

$$\text{diff}(j, X_i, X_k) = \frac{\left| X_i^j - X_k^j \right|}{\text{máx}_X(X^j) - \text{mín}_X(X^j)},$$

para dos observaciones $X_i = (X_i^1, \cdots, X_i^n), X_k = (X_k^1, \cdots, X_k^n)$

2.2. Algoritmo mrmrMID. Este algoritmo fue propuesto recientemente en los artículos C.Ding and H.Peng (2003) y Peng et al. (2005), en donde se hace uso de la teoría de información para la selección de variables. Hablaremos brevemente sobre algunos aspectos teóricos necesarios para comprender éste método, específicamente sobre la teoría de la información mutua. Una referencia para este tema se puede obtener en los siguientes apuntes: `http://www.math.ucsd.edu/~lrothsch/information.pdf`

La teoría de información mutua requiere algunas definiciones desde la teoría de la probabilidad.

Definición 2.1.

- Si X es una variable aleatoria con función de densidad de probabilidad $p(x)$, su **entropía** se define como

$$H(X) = -\int_{\mathbb{R}} p(x)\log p(x)dx$$

  Esta es una medida común de *incertidumbre* para una variable aleatoria.
- Si X, Y son dos variables aleatorias, las cuales tienen la función de densidad de probabilidad conjunta $p(x, y)$, entonces su **entropía conjunta** se define como

$$H(X,Y) = -\int\int_{\mathbb{R}^2} p(x,y)\log(p(x,y))dxdy.$$

Una propiedad importante que posee la entropía es la siguiente: dadas dos variables aleatorias X, Y, entonces $H(X,Y) = H(X) + H(Y) \Leftrightarrow X, Y$ son independientes . Esto motiva la siguiente definición.

Definición 2.2. La **Información Mutua** entre dos variables aleatorias X, Y se define como:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = \int\int_{\mathbb{R}^2} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)dxdy$$

Esta medida trata de cuantificar qué tanta información se necesita saber de una variable aleatoria para caracterizar a la otra. Naturalmente la motivación detrás de estas definiciones ha hecho que la técnica de la información mutua haya sido aplicada con frecuencia en algoritmos de selección de variables.

Un ejemplo de método que utiliza estas teorías es el algoritmo mrmrMID. Sus siglas vienen del acrónimo en inglés Minimum redundancy Maximum relevancy, consiste en realizar una búsqueda hacia adelante de variables de manera que en cada paso del proceso el atributo que se considera significativo minimiza la redundancia en la información que puede dar con respecto a

los atributos ya seleccionados y maximiza la relevancia posible con respecto a la variable de predicción. De nuestra interpretación del artículo C.Ding and H.Peng (2003), el proceso para realizar esta selección es el siguiente:

Algorithm 2 mrmrMID

Input: conjunto de variables X_i (columnas), vector de etiquetas Y (clases). Cantidad de variables m que se desea recuperar.
Output: Conjunto S de variables a seleccionar.
PROCESO:
while $|S| < m$ **do**
    if $|S| = 0$ **then**
        Agregar a S la variable X_i que maximiza $I(X_i; Y)$.
    else
        Sea Ω_S el conjunto de columnas que no están en S.
        Agregar a S la columna X_i de Ω_S para la que se alcance el máximo:
        $i = \text{argmax}_{i \in \Omega_S} \left[I(X_i, Y) - \frac{1}{|S|} \sum_{j \in S} I(X_i, X_j) \right]$

Esta diferencia minimiza la redundancia de la variable al agregarla al cojunto S (mediante la resta que involucra la suma sobre S) y maximiza la relevancia de la variable (mediante la adición del factor $I(X_i, Y)$).

2.3. Nuevo método propuesto por el artículo: algoritmo FSDD. Ahora vamos a presentar el método de selección de variables que se introduce en el artículo principal de referencia en este proyecto. Si bien el argumento detrás de los métodos anteriores representa una forma bastante plausible de calificar a las variables de acuerdo a su importancia predictiva, éstos métodos pueden ser computacionalmente caros y también su motivación teórica un poco complicada (en el caso de mrmrMID por ejemplo). El método FSDD tiene una motivación sencilla, y su desarrollo lo hará más liviano de implementar que los otros métodos.

Todo se basa en el **discriminante de distancias**. Suponemos que tenemos n observaciones Y_i para $i = 1, \cdots, n$, las cuales pertenecen a ciertas clases C_ℓ para $\ell = 1, \cdots, c$. El discriminante de distancias con parámetro de control $\beta \in \mathbb{R}_{>0}$ para este conjunto de datos se define como

$$d_b - \beta d_w,$$

donde d_b es una métrica que cuantifica la distancia entre observaciones de distintas clases y d_w intenta medir la distancia promedio entre observaciones de una misma clase. La motivación detrás de este discriminante es sencilla: se desea tener un conjunto de datos en el que la distancia entre diferentes clases se maximice a la vez que la distancia entre observaciones de una misma clase sea mínima.

Ahora introducimos cierta terminología para explicar estas distancias. Veremos que las definiciones de d_b y d_w son bastante naturales con su motivación.

Definición 2.3.

- La distancia entre dos observaciones $Y_i = (y_i^1, \cdots, y_i^n), Y_j = (y_j^1, \cdots, y_j^n) \in \mathbb{R}^n$ como
$$d(Y_i, Y_j) = \sum_{k=1}^{n} \frac{(y_i^k - y_j^k)^2}{\sigma_k^2},$$
donde σ_k^2 es la varianza de la k-ésima columna predictiva.
- La distancia intra-clase, para la clase C viene dada por
$$d(C) = \frac{2}{N(N-1)} \sum_{i<j} d(Y_i, Y_j), \quad Y_i, Y_j \in C,$$
y la **distancia intra-clase** general entonces se define como $d_w = \sum_{i=1}^{c} \rho_i d(C_i)$, donde ρ_i es la probabilidad anterior de la clase C_i.
- Por otro lado, la **distancia entre clases** se define como $d_b = \frac{1}{2} \sum_{i,j} \rho_i \rho_j d(m_i, m_j)$, donde m_i es el centroide de la clase C_i, es decir $m_i = \frac{1}{|Y_\ell \in C_i|} \sum_{Y_\ell \in C_i} Y_\ell$.

Para efectos del filtro de variables, el siguiente teorema es esencial. Su demostración se encuentra desarrollada en el artículo de referencia.

Teorema 2.4. *Se cumple la fórmula:*

$$d_b - \beta d_w = \sum_{k=1}^{n} \frac{1}{\sigma_k^2} \left[\sigma_k''^2 - \beta \sum_{i=1}^{c} \rho_i \sigma_k^2(i) \right] \quad (1)$$

Donde:

- σ_k^2 *es la varianza de la variable* k*-ésima, de todas las observaciones, esto es:* $\sigma_k^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i^k - \overline{y_k})^2$ *para la media* $\overline{y_k} = \frac{1}{N} \sum_{i=1}^{N} y_i^k$.

- $\sigma_k^2(i)$ *es la varianza de la variable* k*-ésima, de las observaciones en la clase* $i-$*ésima, esto es:*

$$\sigma_k^2(i) = \frac{1}{|Y \in C_i| - 1} \sum_{Y \in C_i} (y^k - \overline{y_k(i)})^2, \quad \overline{y_k(i)} = \frac{1}{|Y \in C_i|} \sum_{Y \in C_i} y^k.$$

- $\sigma_k''^2$ *es la varianza promediada del centroide de la clase* i*-ésima en la variable* k*-ésima. Esto es,* $\sigma_k''^2 = \sum_{i=1}^{c} \rho_i (m_i^k - m_k)^2$ *donde* $m_k = \sum_{i=1}^{c} \rho_i m_i^k$.

El método de ranqueo de variables en este caso es utilizando la fórmula (1). Como los términos en la sumatoria de la fórmula dependen de cada variable y el objetivo principal es maximizar

la métrica general $d_b - \beta d_w$, entonces los autores argumentan que, en cierto sentido, las mejores m variables son las m primeras variables con mayor valor en la métrica

$$\text{FSDD}(k) = \frac{1}{\sigma_k^2}\left[\sigma_k''^2 - \beta \sum_{i=1}^{c} \rho_i \sigma_k^2(i)\right]$$

Y esta es la métrica para ranquear las variables que se propone en este artículo. Es un indicador por variable y depende de la naturaleza del conjunto de datos, y no de ningún modelo en particular (filter).

3. METODOLOGÍA

Para comparar la eficacia del método de selección de variables FSDD el artículo propone el siguiente experimento.

- Se considera una colección de varios conjuntos de datos, todos con una variable de respuesta categórica y variables de predicción numéricas. Para cada conjunto de datos se obtiene un conjunto de datos filtrado en el que aparecen sólo las mejores m variables usando los tres métodos de selección estudiados: reliefF, mrmrMID y FSDD. Esto se hace para varios m, que van desde 1 hasta el número total de columnas en el conjunto de datos que se estudia. En este momento hay un conjunto de datos filtrado para cada combinación de método de selección de variables y cantidad de variables seleccionadas.
- En seguida se ajusta un clasificador (KNN, Naive Bayes, Árbol de Decisión o SVM lineal) sobre cada conjunto de datos filtrado. Esto se hace usando validación cruzada. En seguida se obtiene la precisión de este ajuste. Finalmente se comparan las precisiones obtenidas usando cada método de selección estudiado y cantidad de variables.
- Finalmente se comparan las precisiones obtenidas usando cada método de filtreo y cantidad de variables. Los resultados se muestran en una serie de gráficos de líneas para cada conjunto de datos y para cada clasificador, en los cuales el eje x corresponde a la cantidad de variables seleccionadas y el eje y corresponde a la precisión obtenida con el modelo de clasificación. Cada gráfico tiene 3 líneas que muestran la relación cantidad de variables - precisión usando cada uno de los tres métodos de selección de variables estudiados. Esta información también se presenta en tablas.

Los conjuntos de datos utilizados en el artículo de referencia sobre el valor FSDD son los siguientes: 1. MFeat (Multiple Features Dataset) 2. Satimage 3. Spambase 4. Spectrometer 5. Wine 6. Analcatdata 7. Iris 8. Vowel. Todos, excepto Analcatdata que está en Statlib, se encuentran en el repositorio UCI, de libre acceso y descarga. Para este trabajo se decidió utilizar sólo los que

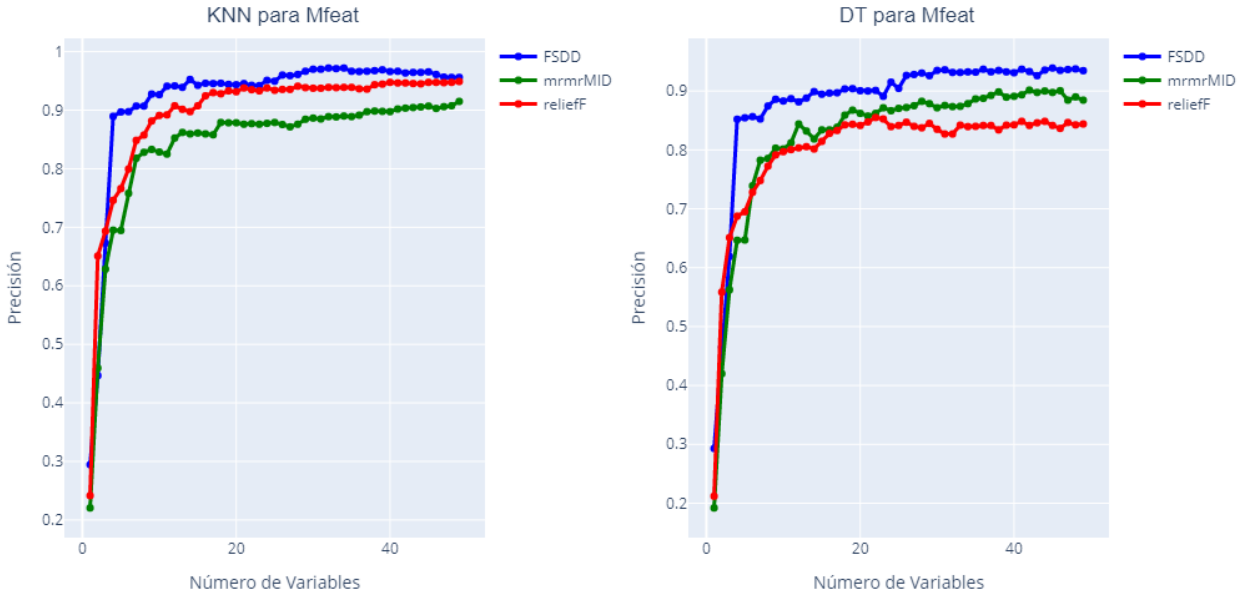
siguen, por cuestiones de tiempo y duración de algunos de los algoritmos. 1. Mfeat, 2. Satimage, 3. Spambase y 4. Wine. En la siguiente table presentamos la información sobre estos conjuntos de datos, dada por el artículo.

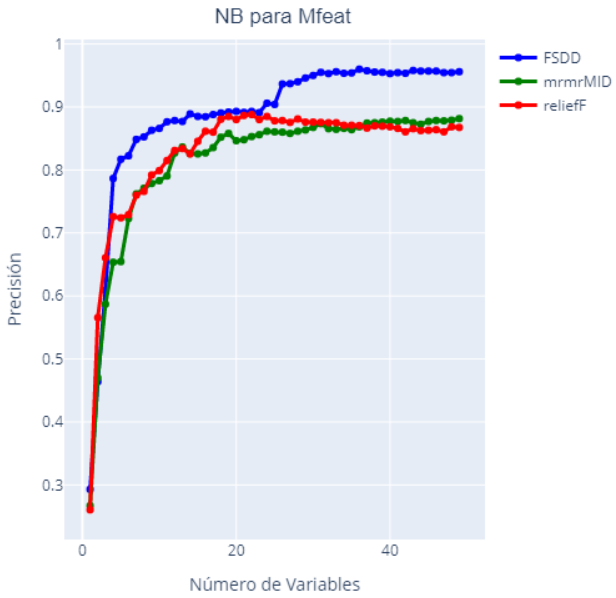
Conjunto de Datos	#Variables	#Instancias	CV Fold
Mfeat	649	2000	2-Fold CV
Satimage	36	6435	2-Fold CV
Spambase	57	4601	2-Fold CV
Wine	13	178	10-Fold CV

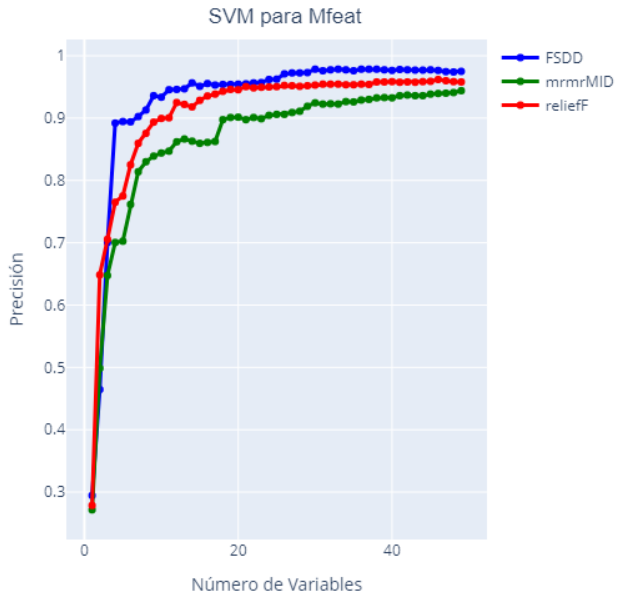
4. Reproducción de resultados

En esta sección mostramos algunos de los resultados obtenidos mediante la implementación que se hizo para simular los experimentos del artículo y discutimos sobre cómo se comparan con los resultados originales. Por razones de espacio no se han incluido los gráficos originales en esta sección sino en el apéndic. Los que siguen son los resultados obtenidos para el conjunto de datos Mfeat utilizando los cuatro clasificadores disponibles.

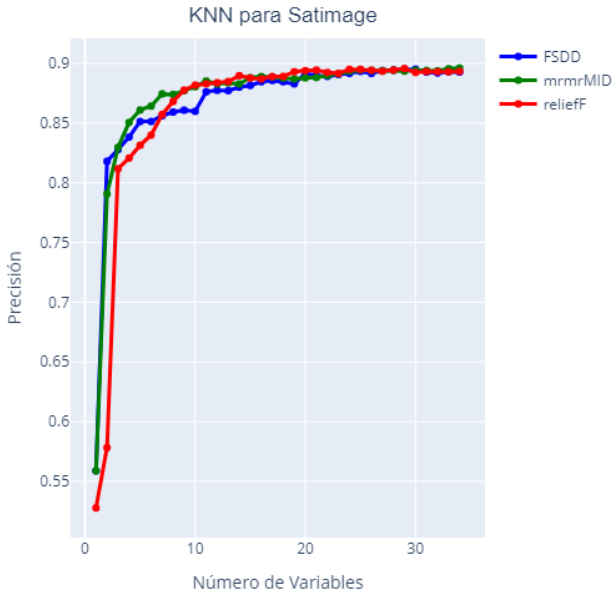
Como se puede observar en este caso se lograron obtener resultados de la misma naturaleza que los que se presentan en el artículo, en los que se observa que la precisión obtenida en los cuatro clasificadores usando las variables seleccionadas por el método FSDD supera a la precisión obtenida con ReliefF y mrmrMID para varios m.

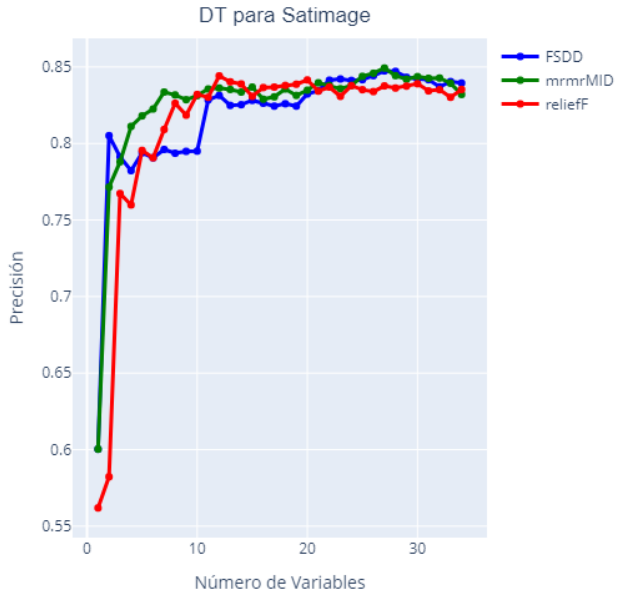


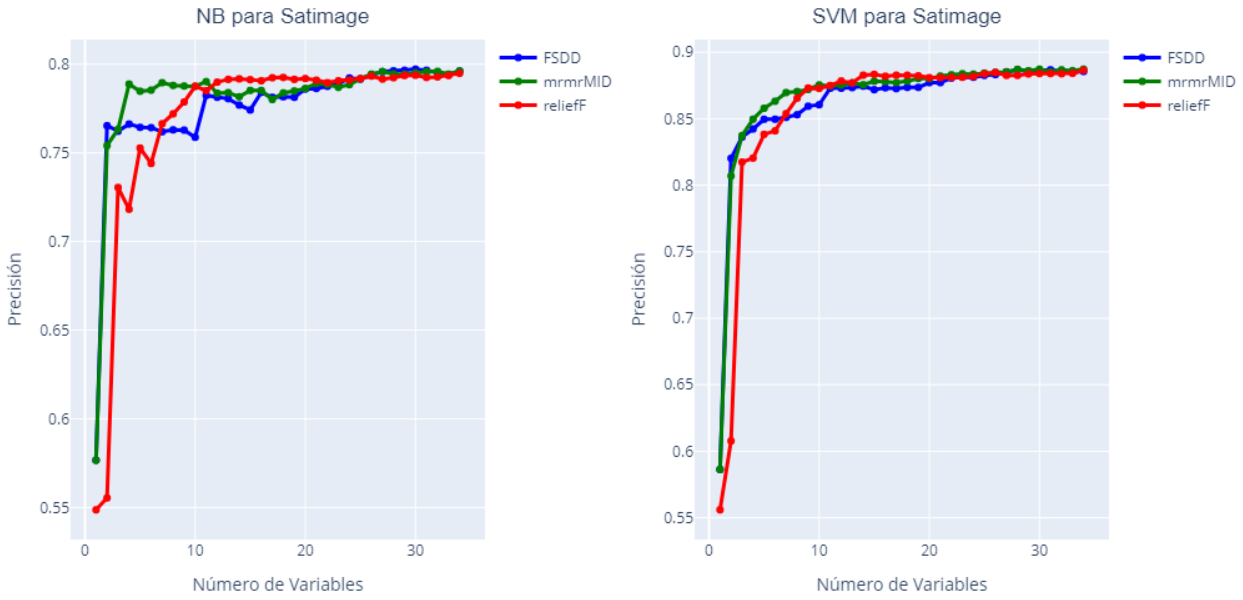




Por su parte éstos son los resultados obtenidos para el conjunto de datos Satimage, que tiene menos variables, pero más observaciones. En este caso tenemos la situación en la que el método FSDD no es significativamente superior a los otros, y más bien el desempeño de los tres métodos es similar a través de los distintos números de variables.







Por cuestiones de espacio, a continuación sólo presentamos dos de los gráficos para el conjunto de datos Spambase. Con éstos dos modelos también se logra obtener, como en el artículo, un desempeño superior por parte de la métrica FSDD, sin embargo no tan significativa en algunos casos a como es propuesta en el artículo de referencia.

Finalmente, para el conjunto de datos Wine se presenta a continuación la tabla que resume las estadísticas obtenidas con la implementación realizada en este proyecto. En el apéndice se

encuentra la tabla original presentada en el artículo, en caso de que el lector quiera compararlas directamente en este documento. En este caso, hemos obtenido análogamente un desempeño relativamente superior del método FSDD con respecto a los otros algoritmos.

Clasif.	Método	m												
		1	2	3	4	5	6	7	8	9	10	11	12	13
KNN	FSDD	0.798	0.899	0.933	0.944	0.955	0.961	0.955	0.944	0.944	0.949	0.961	0.944	0.938
	mrmrMID	0.798	0.719	0.736	0.803	0.809	0.803	0.787	0.803	0.809	0.904	0.927	0.944	0.938
	reliefF	0.528	0.781	0.803	0.949	0.966	0.944	0.921	0.916	0.933	0.91	0.927	0.91	0.938
NB	FSDD	0.753	0.848	0.815	0.876	0.949	0.933	0.933	0.938	0.938	0.916	0.933	0.927	0.921
	mrmrMID	0.753	0.719	0.736	0.86	0.854	0.82	0.843	0.831	0.809	0.888	0.899	0.91	0.921
	reliefF	0.511	0.719	0.764	0.938	0.949	0.944	0.944	0.949	0.938	0.916	0.921	0.916	0.921
DT	FSDD	0.787	0.899	0.91	0.938	0.944	0.966	0.961	0.966	0.972	0.972	0.966	0.972	0.978
	mrmrMID	0.787	0.775	0.831	0.854	0.893	0.876	0.871	0.893	0.916	0.938	0.972	0.983	0.978
	reliefF	0.567	0.775	0.826	0.944	0.961	0.944	0.933	0.927	0.938	0.938	0.949	0.949	0.978
SVM	FSDD	0.803	0.893	0.916	0.949	0.966	0.972	0.972	0.972	0.978	0.972	0.966	0.978	0.978
	mrmrMID	0.803	0.736	0.809	0.86	0.904	0.893	0.871	0.916	0.933	0.955	0.972	0.989	0.978
	reliefF	0.556	0.781	0.82	0.938	0.955	0.949	0.949	0.955	0.944	0.955	0.961	0.944	0.978

5. Problemas y Limitaciones

Los artículos no brindaban referencias para accesar al código original que fue usado por los autores. Se menciona que la implementación fue hecha en matlab. Por esta razón se recurrió a una implementación en python, que puede tener errores o aspectos que no he considerado todavía. Es muy probable que esto haya afectado los resultados. De hecho existen algunos hiperparámetros en el método ReliefF de python que no se pueden controlar, como el número de vecinos a seleccionar en cada iteración. Por falta de tiempo no se implementó este método y en su lugar se utilizó una biblioteca de python hecha para éste.

6. Conclusiones Generales

La motivación teórica detrás de la definición de la métrica FSDD es muy intuitiva y la fórmula desarrollada por los autores permite obtener una nueva métrica para calificar variables dentro del problema de clasificación planteado. A pesar de no contar con la implementación explícita realizada por los autores, se logró emular el código apropiado siguiendo las descripciones de los algoritmos hechas en la bibliografía. Por detalles del software utilizado en algunos casos los experimentos efectuados originalmente no se han podido emular con exactitud, aunque esto también es razonable debido a la diferencia desde el punto de vista del software empleado. A pesar de esta limitación, los resultados obtenidos en este proyecto muestran un desempeño relativamente superior de la métrica FSDD en muchas instancias, y en el peor de los casos las tres métricas terminan siendo equivalentes con respecto al objetivo de precisión.

7. Apéndice

7.1. Código. En el siguiente repositorio de github se encuentra todo el código desarrollado en python para emular los resultados del artículo de referencia.

7.2. Gráficos del artículo original. Se presentan algunos gráficos y tablas obtenidos en el artículo de referencia, con el fin de que el lector pueda comparar los resultados obtenidos en la presente réplica.

Gráficos obtenidos por los autores para Mfeat Tomados de Liang et al. (2008).

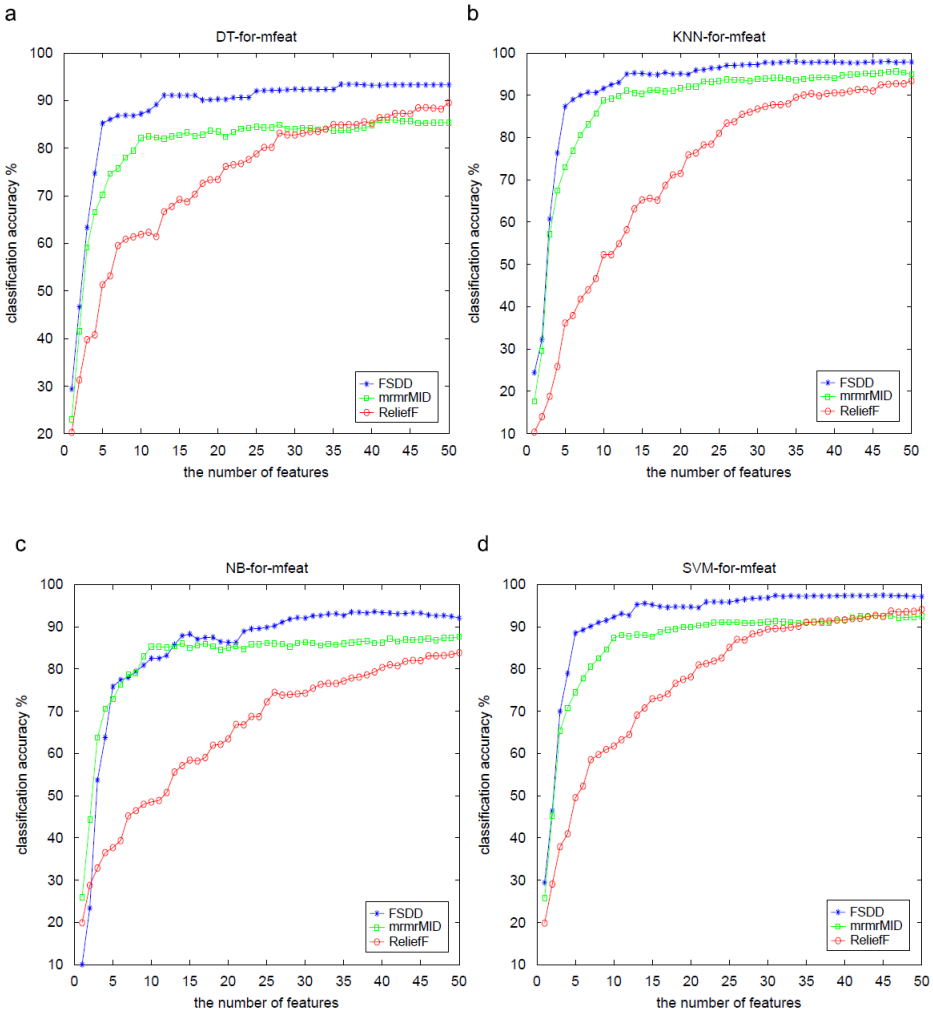


Fig. 1. Two-fold CV classification accuracy for Mfeat with four classifiers.

Gráficos obtenidos por los autores para Satimage Tomados de Liang et al. (2008).

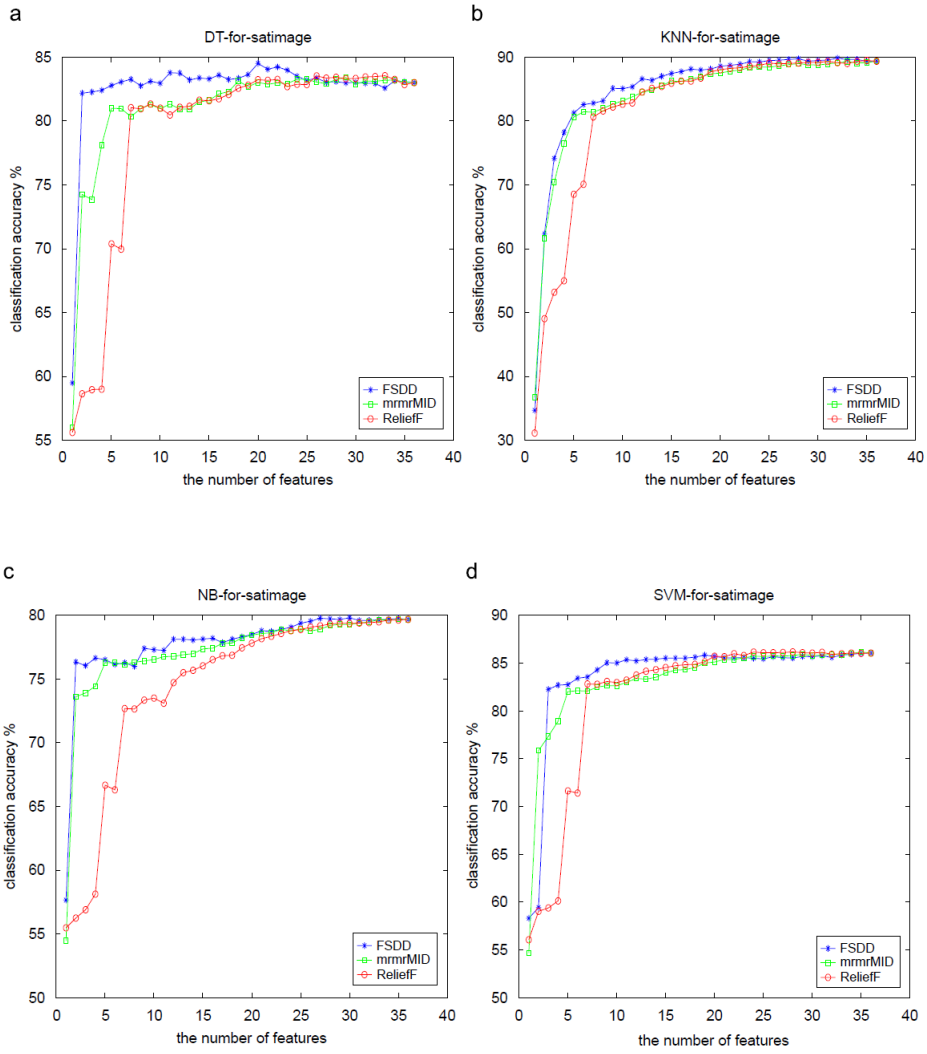
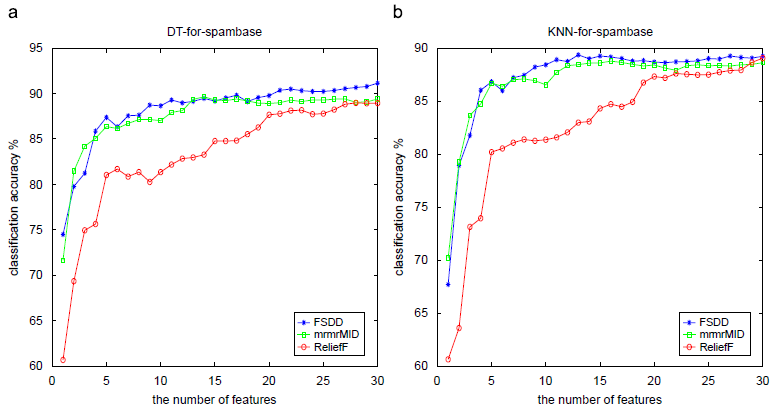


Fig. 2. Two-fold CV classification accuracy for Satimage with four classifiers.

Gráficos obtenidos por los autores para Spambase Tomados de Liang et al. (2008).



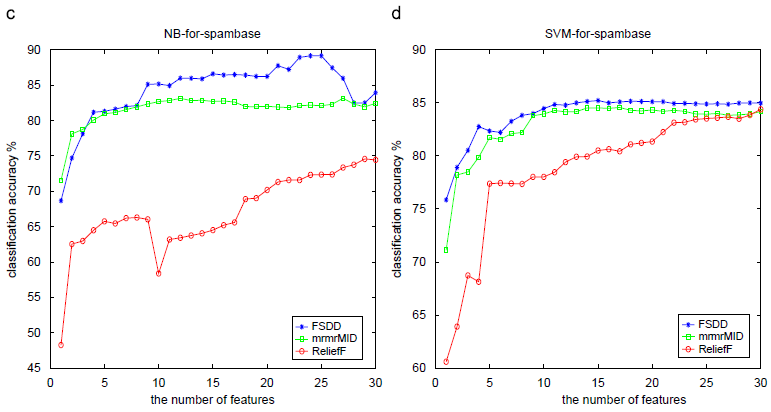


Tabla obtenida por los autores para Wine Tomado de Liang et al. (2008).

Table 3
Ten-fold classification accuracy for Wine

Classifier	mtds	*m*												
		1	2	3	4	5	6	7	8	9	10	11	12	13
KNN	FSDD	70.23	85.39	90.45	92.14	94.94	96.07	96.07	96.07	95.51	96.07	96.07	97.19	95.51
	mrmrMID	38.76	65.17	75.84	75.84	82.58	80.9	85.39	91.01	91.57	94.38	93.82	93.26	95.51
	ReliefF	69.1	82.58	91.57	92.7	93.26	94.94	95.51	95.51	95.51	95.51	94.38	94.94	95.51
NB	FSDD	79.21	88.76	91.57	94.94	94.38	97.19	97.19	96.07	96.63	96.07	96.07	96.07	97.75
	mrmrMID	57.3	73.6	78.09	79.21	84.27	86.52	88.2	94.94	94.38	94.94	95.51	95.51	97.75
	ReliefF	76.97	88.2	91.01	91.57	93.26	94.38	94.38	94.38	94.94	96.63	96.07	97.19	97.75
DT	FSDD	71.91	87.64	92.7	91.57	95.51	95.51	95.51	95.51	95.51	95.51	95.51	94.94	94.94
	mrmrMID	49.44	66.29	73.6	78.65	76.97	78.65	86.52	92.14	92.14	93.82	94.38	93.82	94.94
	ReliefF	70.79	81.46	92.14	92.7	95.51	95.51	95.51	95.51	95.51	95.51	95.51	94.94	94.94
SVM	FSDD	79.21	88.2	92.7	95.51	96.07	97.75	97.19	97.75	97.75	98.32	98.32	97.75	98.32
	mrmrMID	56.18	75.28	82.02	80.9	83.71	85.39	91.57	94.94	95.51	97.19	96.07	97.75	98.32
	ReliefF	75.84	87.64	92.7	92.7	95.51	96.63	97.19	97.19	98.32	98.32	98.32	97.75	98.32

m is the number of features selected. mtds is the abbreviation of the word 'methods'.

REFERENCIAS

C.Ding and H.Peng (2003). Minimum redundancy feature selection from microarray gene expression data. *Proceedings of the IEEE Computer Society, Conference on Bioinformatics*, page 523.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer, New York, 1 edition.

Liang, J., Yang, S., and Winstanley, A. (2008). Invariant optimal feature selection: a distance discriminant and feature ranking based solution. *Pattern Recognition*, 41:1429–1439.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and minredundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1226–1238.

Robnik-Sikonja, M. and Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Mach.Learn.*, 53:23–69.

Urbanowicza, R. J., Olsona, R. S., Schmitta, P., Meekerb, M., and Moorea, J. H. (2018). Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of Biomedical Informatics*, 85:168–188.