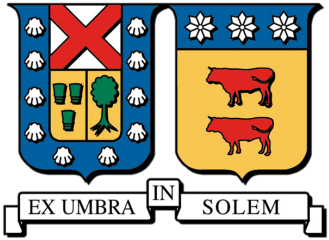


Predicción y clasificación de curse de beneficio bancario

Trabajo de Memoria para optar al título profesional Ingeniero Civil Matemático.

Juan Francisco Briceño Figueroa

Universidad Técnica Federico Santa María, Santiago, Chile.

Profesor Guía: Julio Deride (USM) Profesor Co-Guía: Francisco Alfaro (USM)

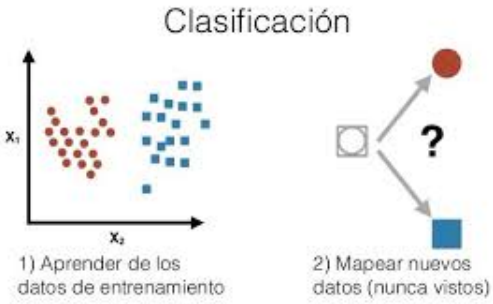
18 de enero de 2022

Outline

Motivación

- Toma de decisiones

- Teoría de clasificación

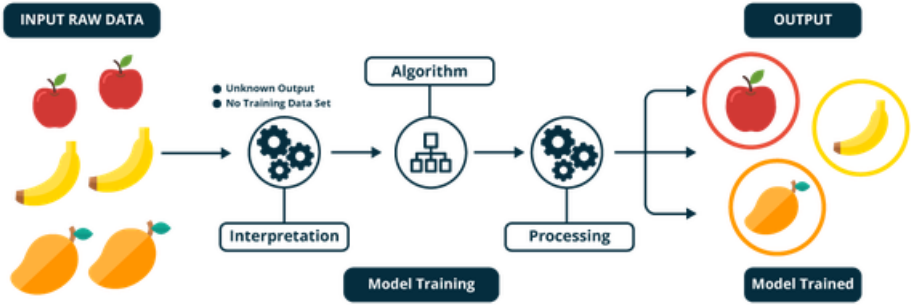


Modelos

Estamos interesados en generar dos tipos de modelos:

1. Modelo clásico de predicción
2. Modelo de soluciones robustas

Modelo clásico de predicción



Objetivo Modelo clásico de predicción:

Diseñar e implementar un método para predecir el comportamiento de cada uno de los clientes de la institución bancaria, basándose en algoritmos de clasificación supervisada.

Modelo de soluciones robustas

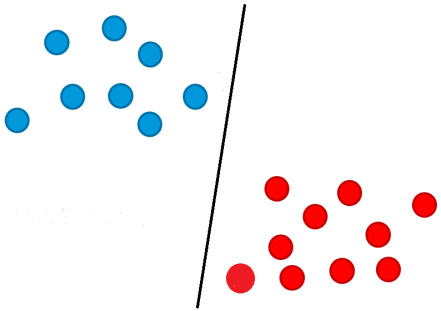


Figura: Modelo Clásico

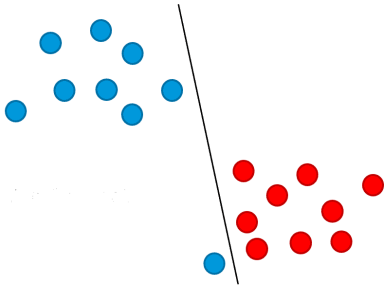


Figura: Modelo robusto

Encontrar un modelo que sea *estable* ante perturbaciones. Esto será generado mediante la técnica de **Diametrical Risk Minimization**, en donde la optimización se genera a partir de bolas con centro en el parámetro entregado.

Objetivo Modelo de soluciones robustas:
Diseñar una nueva técnica de clasificación la cual sea robusta en sus soluciones. Esta técnica será una combinación de Support Vector Machines en conjunto con Diametrical Risk Minimization.

Outline

Conjunto de datos

Datos obtenidos desde una entidad bancaria

Resumen de datos:

N.° de sujetos totales: 5.890.586

N.° de sujetos mensuales: 218.170 aproximadamente

N.° de variables: 96

Clasificación de variables

- Descripción del sujeto.
- Eventos de predicción
- Productos tarjeta de crédito
- Otros productos

Período de tiempo: 2019-10-01 - 2021-04-30

Outline

Problema bancario

Figura: Camino de deudas

Figura: Camino de avance

Problema bancario

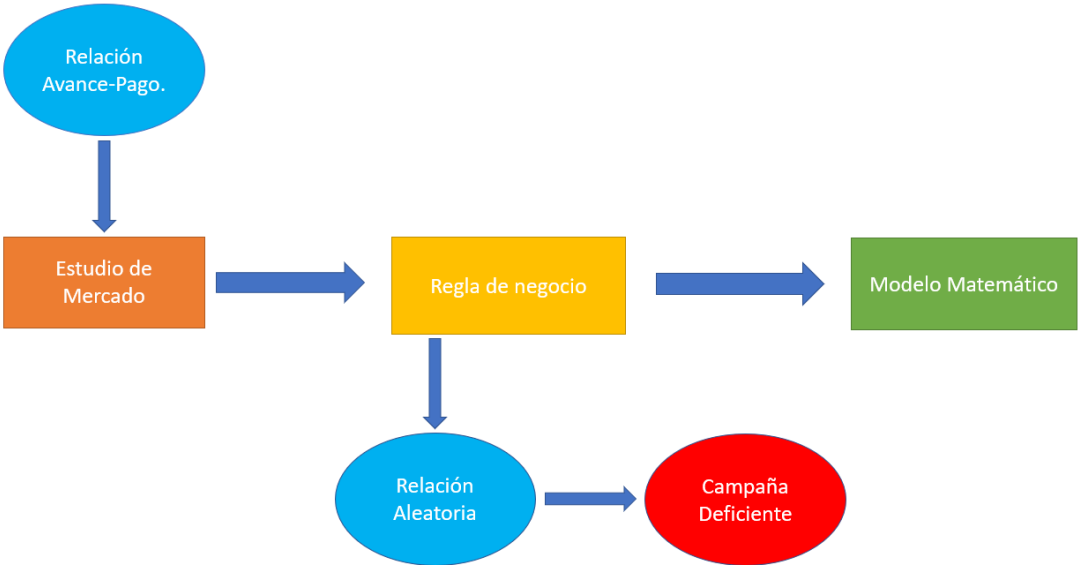


Figura: Línea de tiempo

Algoritmos de clasificación

Estudiamos algoritmos de clasificación matemática que tuvieran un gran rendimiento hoy en día. Los algoritmos seleccionados fueron:

- Support Vector Machines
- Random Forest
- Extreme Gradient Boosting

Support Vector Machines

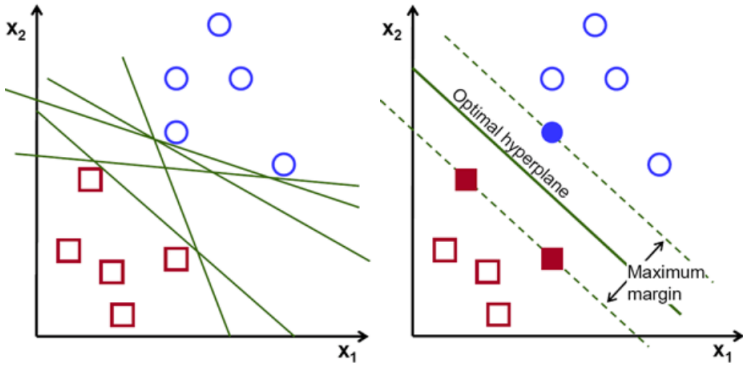


Figura: Support Vector Machines

Support Vector Machines

Sea una muestra de datos clasificados

$$S = \{(x_i, y_i) \in \mathbb{R}^n \times \mathbb{R} \quad \forall i = 0, .., n, \quad y_i \in \{-1, 1\}\},$$

entonces generamos un hiperplano separador definido por (w', b), tal que:

$$y_i(w'x_i + b) < 1\,, \forall i$$

Definiremos w' y b de manera que:

$$\min_i(y_i(w'x_i + b)) = 1$$

Sea el Margen:

$$\text{Margen} = \frac{\min_i(y_i(w'x_i + b)}{\parallel w' \parallel} = \frac{1}{\parallel w' \parallel}, \tag{1}$$

lo que deseamos es buscar el hiperplano que maximiza el margen. Para ello se debe resolver el problema convexo:

$$\min_{w' \in W} \frac{\parallel w' \parallel^2}{2} + C \cdot \sum_i^n \max(0, 1 - y_i \cdot (x_i' \cdot w' + b))$$

Random Forest

Definición

Un árbol de decisión es una estructura similar a un diagrama de flujo en la que cada nodo interno representa una "prueba" en un atributo, cada rama representa el resultado de la prueba y cada nodo hoja representa una etiqueta de clase.

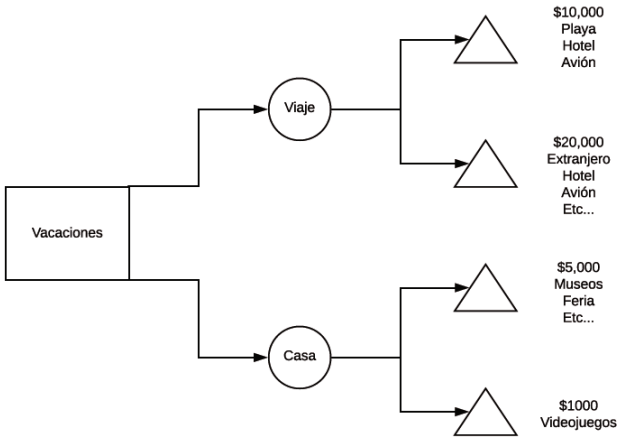


Figura: Árbol de decisión

Random Forest

La idea esencial del **bagging** es promediar muchos modelos ruidosos, pero un tanto imparciales, y así reducir la varianza de estos. Los árboles predictores son candidatos ideales para este proceso.

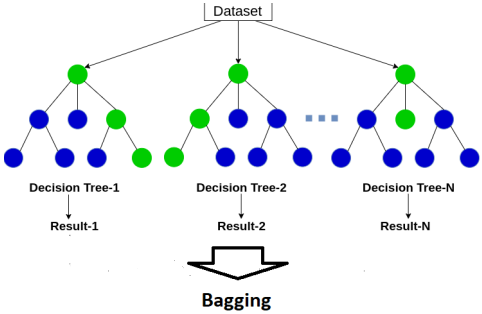


Figura: Bagging con árboles de decisión

Gracias al proceso de **bagging** se logra producir una diferencia en la varianza del proceso, esto concluye con una varianza promedio:

$$V_{bagg} = \rho\sigma^2 + \frac{1-\rho}{N}\sigma^2$$

Random Forest

Definición

Random Forest es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging que construye una larga colección de árboles no correlacionados.

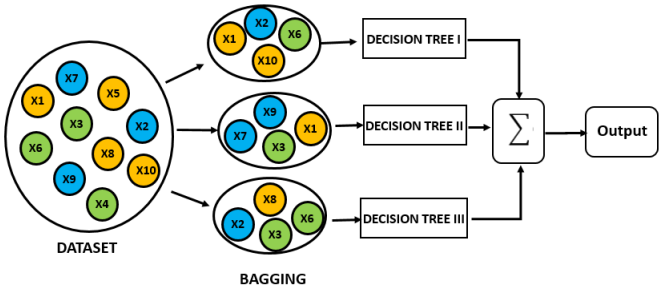


Figura: Random Forest

Random Forest

Medidas de varianza interna de Random Forest

Sea $\hat{p}_{mk}$ como la proporción de observaciones de entrenamiento en la m-esima región que se encuentra en la k-esima clase, entonces:

- Criterio Gini:

$$G = \sum_{k=1}^{N} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- Criterio Entropía:

$$E = -\sum_{k=1}^{N} \hat{p}_{mk} \log \hat{p}_{mk}$$

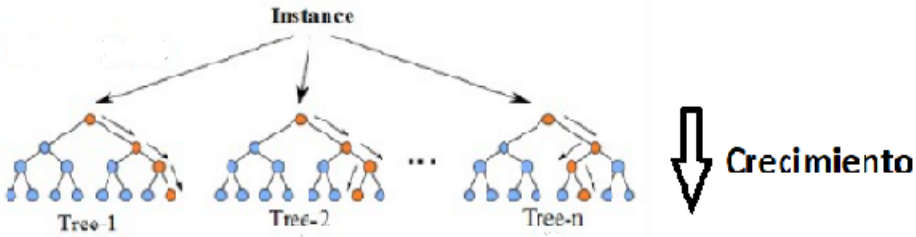


Figura: Crecimiento de los árboles

Extreme Gradient Boosting

El **boosting**, consiste en combinar los resultados de varios clasificadores débiles para obtener un clasificador robusto.

La idea general es, generar múltiples modelos débiles secuencialmente y que cada uno tome los resultados del modelo anterior, lo cual implicaría un modelo más fuerte y más estable.

Definición

Extreme Gradient Boosting (XGBoost) es una técnica que sigue el principio de árboles de decisión boosting potenciados por gradientes, pero este a su vez se especializa en su formulación.

$$Fun(\phi) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \tag{2}$$

donde:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f \in F \tag{3}$$

Extreme Gradient Boosting

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \| w \|^2 \tag{4}$$

donde:

- γ: peso asociado a la cantidad de hojas T.
- λ: factor asociado a la norma del vector w.
- w: vector con *scores* en cada hoja.
- T: cantidad de hojas.

Para una estructura fija es posible encontrar el peso óptimo dado por:

$$w_j^* = -\frac{\sum_{j \in I} g_j}{\sum_{j \in I} h_j + \lambda}$$

Dada nuestra función objetivo, el valor óptimo estará denotado por:

$$F^*(q) = \frac{-1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} + \gamma T$$

donde:

- $g_i = \partial_{\hat{y}^{t-1}} L(y_i, \hat{y}^{t-1})$.
- $h_i = \partial^2_{\hat{y}^{t-1}} L(y_i, \hat{y}^{t-1})$.

Extreme Gradient Boosting

Crecimiento de los árboles:

$$Ganancia = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

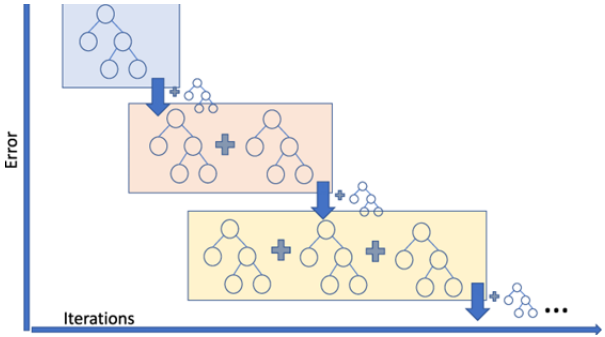


Figura: Extreme Gradient Boosting

Metricas de evaluación

Para evaluar el desempeño de los algoritmos usados en la predicción de la propensión a usar este beneficio bancario, usamos dos métricas diferentes. Entre las métricas que se utilizarán se encuentran:

- **F1-Score**
- **ROC-Score**

	Negativo Predicho	**Positivo Predicho**
Falso Real	Verdadero Negativo (TN)	Falso Positivos (FP)
Verdadero Real	Falso Negativo (FN)	Verdadero Positivo (TP)

Cuadro: Matriz de confusión para nuestro problema.

F1-Score

Precisión

Responde a la pregunta: ¿cuántos clientes contactados, estarán interesados?

$$Precisión = \frac{TP}{TP + FP}$$

Recall

Responde a la pregunta: ¿qué porcentaje de los clientes están interesados y somos capaces de identificar?

$$Recall = \frac{TP}{TP + FN}$$

F1-Score es el promedio harmónico entre *Precisión* y *Recall*:

$$F1 = \frac{2}{Recall^{-1} + Precisión^{-1}}$$

$$= \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Area bajo la curva (Roc-Score)

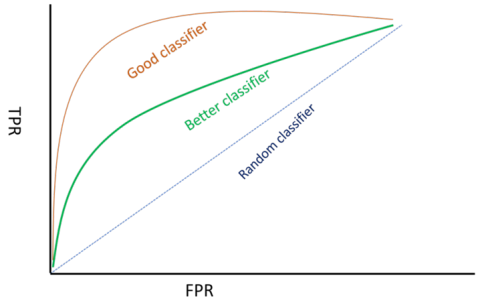


Figura: Curva ROC

Tasa de verdaderos positivos (TPR)	$TPR = \frac{TP}{TP+FN}$
Tasa de falsos positivos (FPR)	$FPR = \frac{TP}{TN+FP}$

Cuadro: Entradas que generan los ejes de la curva ROC.

El área bajo la curva Roc (*AUC* o *ROC-Score*) es equivalente a la probabilidad de que un evento positivo seleccionado aparezca sobre un evento negativo seleccionado al azar.

Resultado de todos los algoritmos: SVM, RF, XGB

Para cada uno de los sets de datos generados (SD), los cuales se diferencian por el "target" o suceso.

- Cada uno de los sets de datos (SD) posee un 50 % de valores positivos y un 50 % de valores negativos.
- Se utilizó un 50 % de la muestra como entrenamiento para el modelo y el 50 % restante para testear el modelo.

Resultados

Precisión	*Recall*	*F1-Score*	*ROC-Score*	*SD*	*Modelo*
0,80	0,89	0,84	0,84	target_2	Random Forest 1
0,79	0,86	0,82	0,81	target_3	Random Forest 1
0.79	**0.89**	**0.84**	**0.83**	**target_2**	**Random Forest 2**
0,80	0,90	0,84	0,84	target_2	Random Forest 3
0,81	0,84	0,82	0,82	target_2	Support Vector Machines 1
0.81	**0.83**	**0.82**	**0.82**	**target_2**	**Support Vector Machines 2**
0,81	0,84	0,82	0,82	target_2	Support Vector Machines 3
0,79	0,83	0,81	0,80	target_1	Extreme Gradient Boosting 1
0.80	**0.89**	**0.84**	**0.84**	**target_2**	**Extreme Gradient Boosting 1**
0,78	0,83	0,81	0,80	target_3	Extreme Gradient Boosting 1
0,79	0,89	0,84	0,83	target_2	Extreme Gradient Boosting 2
0,80	0,87	0,83	0,83	target_2	Extreme Gradient Boosting 3

Cuadro: Resultados para los conjuntos de entrenamiento de todos los sets de datos estudiados.

Extreme Gradient Boosting 1

Grupo Homogéneo	*Promedio probabilidad*	*Total de personas positivas*	*% por grupo*
1	95,6 %	695	19,4 %
2	91,5 %	657	18,4 %
3	87,2 %	640	17,9 %
4	80,9 %	586	16,4 %
5	70,1 %	507	14,2 %
6	50,6 %	329	9,2 %
7	23,0 %	128	3,6 %
8	4,6 %	29	0,8 %
9	1,6 %	7	0,2 %
10	1,0 %	1	0,0 %

Cuadro: Cuadro con grupos homogéneos, obtenido de la clasificación realizada por el modelo XGB 1 en el conjunto de testeo.

Extreme Gradient Boosting 1

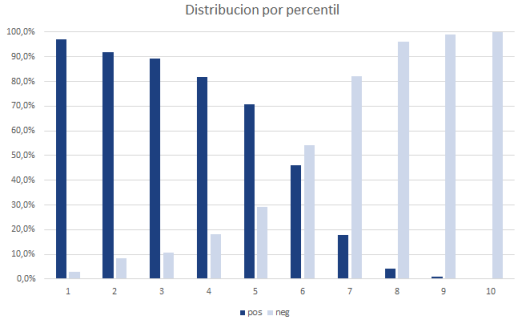


Figura: Distribución de personas positivas a lo largo de los grupos homogéneos.

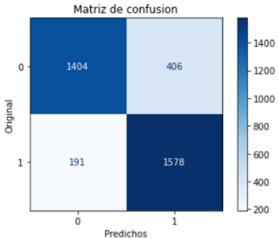


Figura: Matriz de confusión obtenida por el modelo XGB 1.

Random Forest 2

Grupo Homogéneo	*Promedio probabilidad*	*Total de personas positivas*	*% por grupo*
1	97,4 %	704	19,7 %
2	93,4 %	698	19,5 %
3	88,5 %	668	18,7 %
4	81,3 %	634	17,7 %
5	68,8 %	521	14,6 %
6	43,5 %	280	7,8 %
7	21,6 %	55	1,5 %
8	7,6 %	12	0,3 %
9	1,3 %	5	0,1 %
10	0,0 %	2	0,1 %

Cuadro: Cuadro con grupos homogéneos, obtenido de la clasificación realizada por el modelo RF 2 en el conjunto de testeo.

Random Forest 2

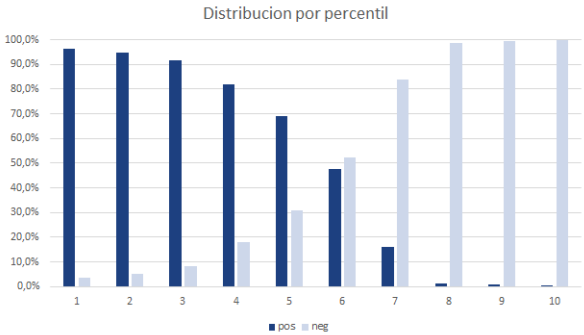


Figura: Distribución de personas positivas a lo largo de los grupos homogéneos.

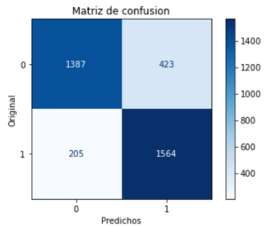


Figura: Matriz de confusión obtenida por el modelo RF 2.

Support Vector Machines 2

Grupo Homogéneo	*Promedio probabilidad*	*Total de personas positivas*	*% por grupo*
1	96,5 %	640	17,9 %
2	91,5 %	625	17,5 %
3	84,3 %	604	16,9 %
4	73,6 %	550	15,4 %
5	59,2 %	485	13,6 %
6	40,8 %	373	10,4 %
7	23,2 %	172	4,8 %
8	15,5 %	60	1,7 %
9	9,4 %	44	1,2 %
10	3,7 %	26	0,7 %

Cuadro: Cuadro con grupos homogéneos, obtenido de la clasificación realizada por el modelo SVM 2 en el conjunto de testeo.

Support Vector Machines 2

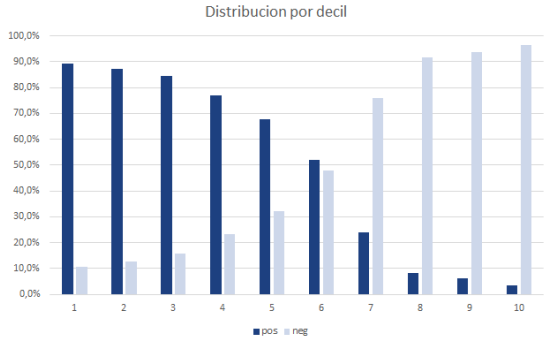


Figura: Distribución de personas positivas a lo largo de los grupos homogéneos.

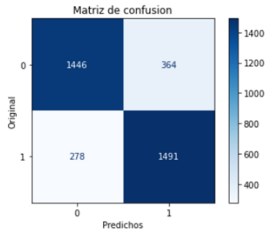


Figura: Matriz de confusión obtenida por el modelo SVM 2.

Resultados

Modelo	*Tiempo de ejecución*	*Memoria utilizada*
Extreme Gradient Boosting	$8,19$**s**	$577,69$**Mb**
Random Forest	$10,1$s	541Mb
Support Vector Machines	265**s**	$538,32$**Mb**

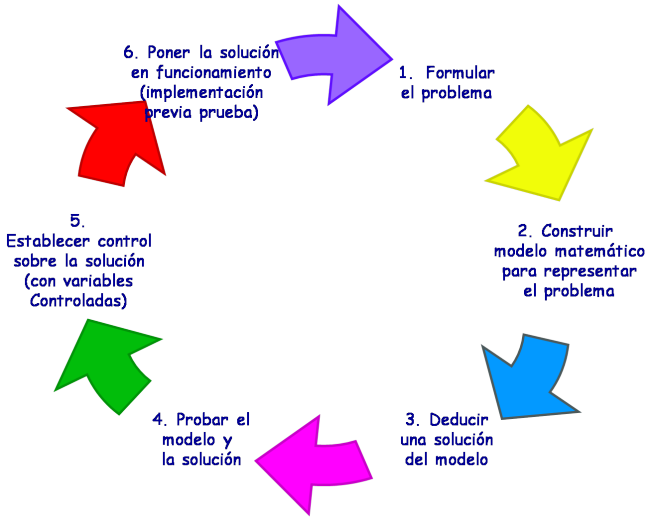
Cuadro: Cuadro con los valores obtenidos por cada modelo en su ejecución.

Outline

Teoría del aprendizaje

La búsqueda de una solución a un problema aplicado generalmente requiere los siguientes pasos:

1. Exprese el problema en términos matemáticos.
2. Formular un principio general para buscar una solución al problema.
3. Desarrollar un algoritmo basado en dicho principio general.



Teoría del aprendizaje

El proceso de aprendizaje se describe a través de tres componentes:

1. Un generador de **vectores aleatorios** $x \in X$, extraído independientemente de un X fijo pero desconocido con **distribución** $P(x)$.
2. Un **supervisor** que devuelve un vector de salida y a cada vector de entrada x, según una **función de distribución condicional** $P(y|x)$, también fija, sin embargo, desconocida.
3. Una **máquina de aprendizaje** capaz de implementar un conjunto de funciones $f(X, w)$, $w \in W$.

La formulación dada anteriormente implica que el aprendizaje es un problema de aproximación de funciones y para elegir la **mejor aproximación** se escoge según que tan cercano es a la solución real. Considere el valor esperado de la pérdida, dado por el funcional de riesgo:

$$R(w) = \int_{X \times Y} l(y, f(x, w)) dP(x, y)$$

¿Cómo estimaremos nuestra solución?

Principio de inducción de la minimización de riesgo (Vapnik, 1990)

La meta es minimizar el funcional de riesgo $R(w)$ sobre la clase de funciones $f(x, w)$, $w \in W$, es decir:

$$\min_{w \in W} R(w)$$

Como $P(x, y)$ en la fórmula es desconocido y la única forma de obtenerlo es con base en nuestro conjunto de datos.

Así para resolver el problema del **principio de minimización de riesgo (ERM)**, se cambia el funcional $R(w)$ por el funcional de riesgo empírico:

$$E(w) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i, w))$$

ERM asume que $E(w) \approx R(w)$

ERM $\longrightarrow$ menor sesgo que el riesgo verdadero $\Longrightarrow$ pobre en términos de su riesgo verdadero $R(w)$

Diametrical Risk Minimization

Definición

Para una función de pérdida $l : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ y muestra $S \subset \mathbb{R}^d$, el riesgo diametral de vector parámetro $w \in \mathbb{R}^n$, está dado por:

$$R_m^\gamma(w) = \sup_{\|v\| \leq \gamma} R_m(w+v) = \sup_{\|v\| \leq \gamma} \frac{1}{m} \sum_{i=1}^{m} l(w+v, z_i) \tag{5}$$

donde $\gamma \in [0, \infty)$ es un parámetro que representa el radio de riesgo diametral.

Entonces para un conjunto $W \subset \mathbb{R}^n$ de vectores parámetros permisibles, el **problema DRM** apunta a:

$$\min_{w \in W} R_m^\gamma(w) \tag{6}$$

lo cual tiene como resultado una solución w_m^γ

Diametrical Risk Minimization

DRM considera el riesgo diametral en un punto del espacio de parámetros, el cual es dado por el peor riesgo empírico en una vecindad de un punto.

Cota uniforme en R real:

$$R(w_m^\gamma) \leq R_m^\gamma(w_m^\gamma) + \epsilon, \text{ con una alta probabilidad}$$

para cualquier vector de parámetro w_m^γ producido por DRM.

Cambios

- Tiene menos limitaciones que ERM, pues ERM depende del módulo de Lipschitz. **Teorema (Cota de estabilidad ERM)[Charles y Papailiopoulos, 2017]:** Asuma que para todo S, f_S satisface condiciones de gradiente con constante μ, suponga que el mínimo es único, y la función f es L-Lipschitz. Entonces la técnica A con parámetro ϵ_A de convergencia tiene una estabilidad uniforme con parámetro ϵ_{est} satisfaciendo las siguientes condiciones:
  Si para todo S, $\| w_s - w_s^* \| \leq O(\epsilon_A)$ entonces:

$$\epsilon_{est} \leq O(L\epsilon_A) + \frac{2L^2}{\mu n}$$

- Perturba el vector de parámetros en vez de la distribución que gobierna sobre los datos mismos.

Mejora: El problema del sobreajuste es removido

Cambios

- Tiene menos limitaciones que ERM, pues ERM depende del módulo de Lipschitz. **Teorema (Cota de estabilidad ERM)[Charles y Papailiopoulos, 2017]:** Asuma que para todo S, f_S satisface condiciones de gradiente con constante μ, suponga que el mínimo es único, y la función f es L-Lipschitz. Entonces la técnica A con parámetro ϵ_A de convergencia tiene una estabilidad uniforme con parámetro ϵ_{est} satisfaciendo las siguientes condiciones:
  Si para todo S, $|f_S(w_S) - f_S(w_S^*)| \leq O(\epsilon_A)$ entonces

$$\epsilon_{est} \leq O(L\sqrt{\frac{\epsilon_A}{\mu}}) + \frac{2L^2}{\mu n}$$

- Perturba el vector de parámetros en vez de la distribución que gobierna sobre los datos mismos.

Mejora: El problema del sobreajuste es removido

Cambios

- Tiene menos limitaciones que ERM, pues ERM depende del módulo de Lipschitz. **Teorema (Cota de estabilidad ERM)[Charles y Papailiopoulos, 2017]:** Asuma que para todo S, f_S satisface condiciones de gradiente con constante μ, suponga que el mínimo es único, y la función f es L-Lipschitz. Entonces la técnica A con parámetro ϵ_A de convergencia tiene una estabilidad uniforme con parámetro ϵ_{est} satisfaciendo las siguientes condiciones:
  Si para todo S, $\| \nabla f_S(w_S) \| \leq O(\epsilon_A)$ entonces

$$\epsilon_{est} \leq O(\frac{L\epsilon_A}{\mu}) + \frac{2L^2}{\mu n}$$

- Perturba el vector de parámetros en vez de la distribución que gobierna sobre los datos mismos.

Mejora: El problema del sobreajuste es removido

Algoritmo DRM-SGD

Entrada: S: conjunto de datos a los cuales se le desea generar SVM-DRM.
γ: valor de la norma de perturbación (radio de perturbación).
r: número de perturbaciones distintas que deseamos.

Salida: Parámetros robustos según DRM.

Paso 1: Se generan una partición de T elementos tales que S=$\bigcup_{t=0}^{t=T} B_t$
Paso 2: Iniciar $w^0 \in W, t = 0$
Paso 3: Iniciar secuencia de subgrupos $B_t \subset S$ con una tasa de aprendizaje $\lambda_t > 0$
Paso 4: Generar r perturbaciones aleatorias $U = \{u_1, \ldots, u_r | \parallel u \parallel = \gamma\}$
Paso 5: Recorrer las perturbaciones de U, calcular $P = \frac{1}{|B_t|} \sum_{z \in B_t} l(w^t + u, z)$
Paso 6: Seleccionar $u^* \in \text{argmax}_{u \in U} P$
Paso 7: Calcular $w^{t+1} = \text{pry}_W(w^t - \lambda_t \nabla_w R_{B_t}(w^t + u^*))$
Paso 8: Si $t = T$ se termina el algoritmo si no se regresa al Paso 4 con $t = t + 1$

Algoritmo DRM-SGD memoria corta

Entrada: S: conjunto de datos a los cuales se le desea generar SVM-DRM.
γ: valor de la norma de perturbación (radio de perturbación).
r: número de perturbaciones distintas que deseamos.
q: número máximo de elementos en nuestro conjunto V_t.
Salida: Parámetros robustos según DRM.

Paso 1: Se generan una partición de T elementos tales que S=$\bigcup_{t=0}^{t=T} B_t$
Paso 2: Iniciar $w^0 \in W, t = 0$
Paso 3: Iniciar secuencia de subgrupos $B_t \subset S$ con una tasa de aprendizaje $\lambda_t > 0$
Paso 4: Generar r perturbaciones aleatorias $U = \{u_1, \ldots, u_r | \parallel u \parallel = \gamma\}$
Paso 5: Recorrer las perturbaciones de U, calcular $P = \frac{1}{|B_t|} \sum_{z \in B_t} l(w^t + u, z)$
Paso 6: Seleccionar $u^* \in \text{argmax}_{u \in U} P$
Paso 7: Añadir u^* a un conjunto V_t, si $|V_t| > q$ remover el elemento más antiguo
Paso 8: Seleccionar $v^* \in \text{argmax}_{u \in V_t} P$
Paso 9: Calcular $w^{t+1} = \text{pry}_W(w^t - \lambda_t \nabla_w R_{B_t}(w^t + v^*))$
Paso 10: Si $t = T$ se termina el algoritmo si no se regresa al Paso 4 con $t = t + 1$

Datos de Iris

Describiremos los efectos de SVM-DRM sobre la muestra de Iris.

Descripcion de datos Iris

- Número de sujetos: 150
- Número de atributos: 4, de los cuales 3 son atributos para predecir y el último es la clase.
- Información de atributo:
  - largo del sépalo en cm
  - ancho del sépalo en cm
  - largo del pétalo en cm
  - ancho del pétalo en cm
  - Clase: Iris-Setosa, Iris-Versicolor, Iris-Virginica.

Resumen estadístico:

Variable	Min	Max	Promedio	Desviacion std.	Correlación clase
largo del sépalo	4.3	7.9	5.84	0.83	0.7826
ancho del sépalo	2.0	4.4	3.05	0.43	-0.4194
largo del pétalo	1.0	6.9	3.76	1.76	0.9490
ancho del pétalo	0.1	2.5	1.20	0.76	0.9565

Cuadro: Tabla descriptiva conjunto Iris.

Modelo Iris

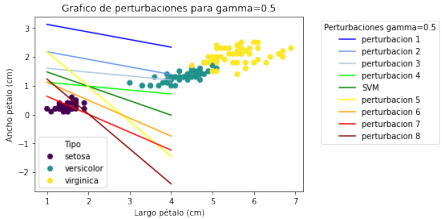


Figura: Toda perturbación cuando $\gamma = 0{,}5 \cdot \| w \|$

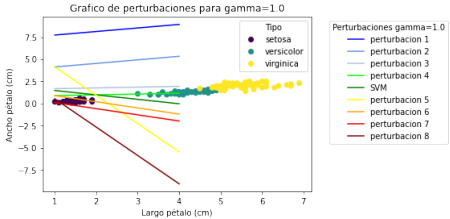


Figura: Toda perturbación cuando $\gamma = 1{,}0 \cdot \| w \|$

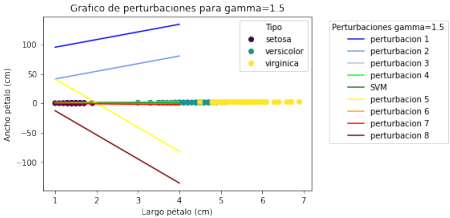


Figura: Toda perturbación cuando $\gamma = 1{,}5 \cdot \| w \|$

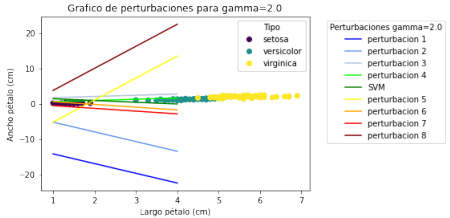


Figura: Toda perturbación cuando $\gamma = 2{,}0 \cdot \| w \|$

Solución modelo Iris SVM-DRM-SDG

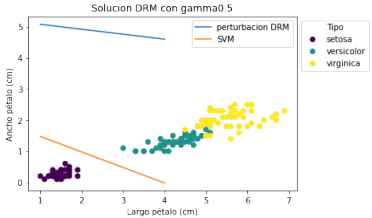


Figura: Solución SVM-DRM con $\gamma = 0{,}5 \cdot \| w \|$

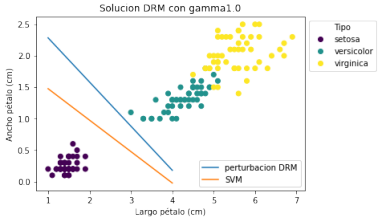


Figura: Solución SVM-DRM con $\gamma = 1{,}0 \cdot \| w \|$

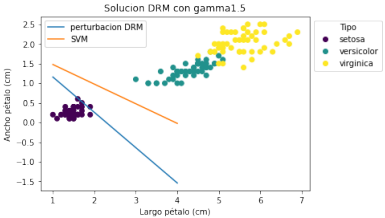


Figura: Solución SVM-DRM con $\gamma = 1{,}5 \cdot \| w \|$

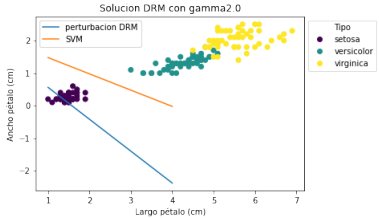


Figura: Solución SVM-DRM con $\gamma = 2{,}0 \cdot \| w \|$

Modelo Iris SVM-DRM-memoria corta

Para este modelo utilizaremos las mismas perturbaciones que el algoritmo anterior, pero debemos agregar la cardinalidad del conjunto V_t. Esta variará entre $[2, 5, 8]$

Solución modelo Iris SVM-DRM-memoria corta

Para un conjunto V_t con cardinalidad 2 se tiene que:

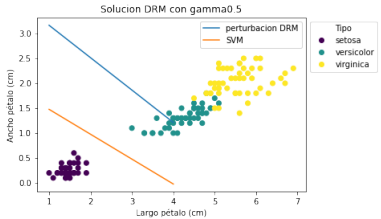


Figura: Solución SVM-DRM con $\gamma = 0{,}5 \cdot \| w \|$

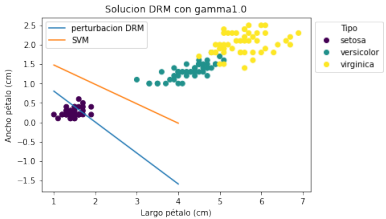


Figura: Solución SVM-DRM con $\gamma = 1{,}0 \cdot \| w \|$

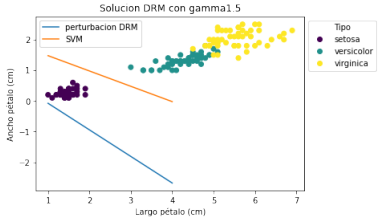


Figura: Solución SVM-DRM con $\gamma = 1{,}5 \cdot \| w \|$

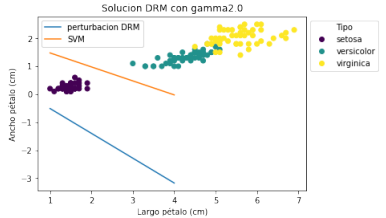


Figura: Solución SVM-DRM con $\gamma = 2{,}0 \cdot \| w \|$

Modelo Bancario SVM-DRM-SGD

Utilizaremos un conjunto de $[4400, 5000]$ prospectos, todos ellos equilibrados para utilizarlos como entrenamiento para nuestros modelos, y se usó 1432 prospectos como conjunto de prueba.

El estudio del algoritmo se basó en 3 aristas:

- Matriz de confusión
- Roc-Score
- N.º Positivos

La porción de perturbación de γ varía entre $10\,\% - 100\,\%$ del módulo de w.

El término λ_t descrito en los algoritmos varía de la siguiente forma;

- porción de datos entre él $0 - 25\,\%$ $\quad \lambda_t = 0{,}1$
- para la porción $25 - 75\,\%$ $\quad \lambda_t = 0{,}01$
- para la porción final $\lambda_t = 0{,}001$

Resultados modelos bancarios SVM-DRM-SDG

Gama γ	ROC-Score	N.º positivos	Función objetivo
10 %	0.38	1001	1
20 %	0.36	593	0.0013
30 %	0.41	688	0.0018
40 %	0.43	680	0.0012
50 %	0.33	539	0.0015
60 %	0.39	316	0.0181
70 %	0.44	854	0.0002
80 %	0.40	706	0.0001
90 %	0.32	920	0.0001
100 %	**0.51**	**1411**	**0.0911**

Cuadro: Cuadro resumen de resultados obtenidos en los distintos modelos para una población de 4400 personas. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.

Modelo Bancario SVM-DRM-SGD

Mejor modelo de generado con un conjunto de 4400

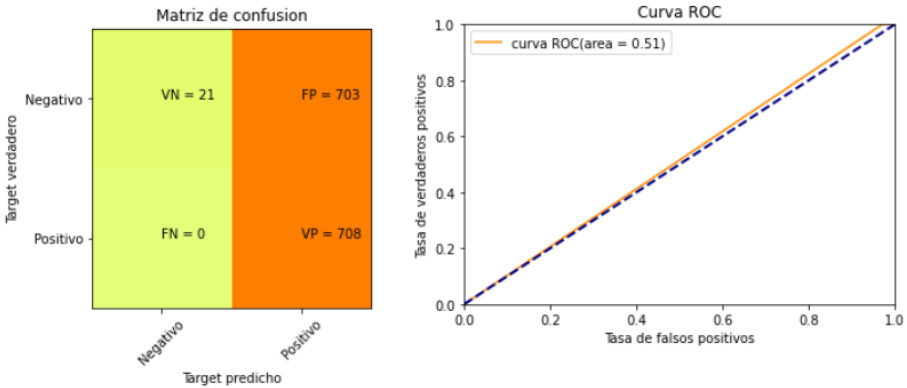


Figura: Resultados de algoritmo SVM-DRM para un $\gamma = 100\,\%$

Resultados modelos bancarios SVM-DRM-SDG

Gama γ	ROC-Score	N.º positivos	Función objetivo
10 %	0.37	375	0.0021
20 %	0.42	470	0.0002
30 %	**0.55**	**1072**	**0.2259**
40 %	0.35	596	0.0012
50 %	0.41	377	0.6777
60 %	0.32	650	0.0019
70 %	0.43	527	0.0014
80 %	0.32	445	1
90 %	0.28	615	0.00003
100 %	0.44	978	0.00001

Cuadro: Cuadro resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.

Modelo Bancario SVM-DRM-SGD

Mejor modelo de generado con un conjunto de 5000

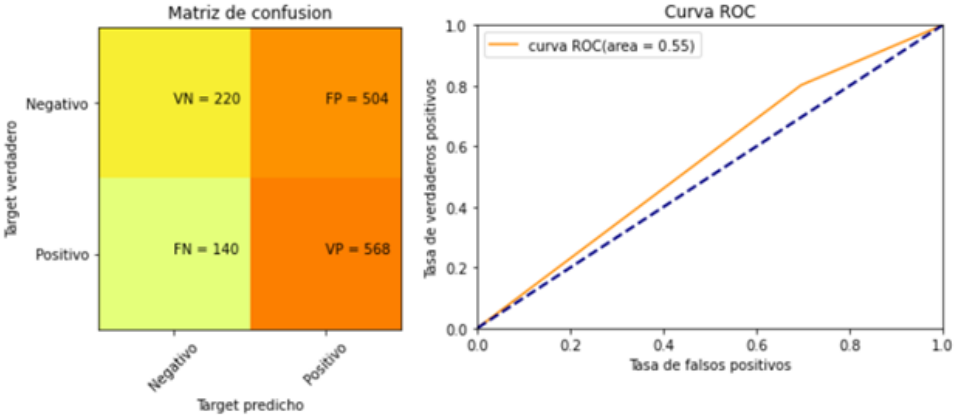


Figura: Resultados de algoritmo SVM-DRM para un $\gamma = 30\,\%$

Modelo Bancario SVM-DRM-memoria corta

Utilizaremos un conjunto de 5000 prospectos, todos ellos equilibrados para utilizarlos como entrenamiento para nuestros modelos, y se usó 1432 prospectos como conjunto de prueba. La cardinalidad del conjunto V_t variará entre $[2, 5, 8]$

El estudio del algoritmo se basó en 3 aristas:

- Matriz de confusión
- Roc-Score
- N.º Positivos

La porción de perturbación de γ varía entre $10\,\% - 100\,\%$ del módulo de w.

El término λ_t descrito en los algoritmos varía de la siguiente forma;

- porción de datos entre él $0 - 25\,\%$ $\quad \lambda_t = 0{,}1$
- para la porción $25 - 75\,\%$ $\quad \lambda_t = 0{,}01$
- para la porción final $\lambda_t = 0{,}001$

Resultados modelos bancarios SVM-DRM-memoria corta

Gama γ	ROC-Score	N.º positivos	Función objetivo
10 %	0.28	804	0.3118
20 %	0.25	590	0.3565
30 %	**0.47**	**1006**	**0.1553**
40 %	0.43	1089	0.1377
50 %	0.28	608	0.3285
60 %	0.33	491	0.3852
70 %	0.33	521	0.2491
80 %	0.32	542	1
90 %	0.33	729	0.1383
100 %	0.41	882	0.1549

Cuadro: Cuadro resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas, con un conjunto $|V_t| = 2$. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.

Modelo Bancario SVM-DRM-memoria corta

Mejor modelo de generado con un conjunto de 5000 y $|V_t| = 2$

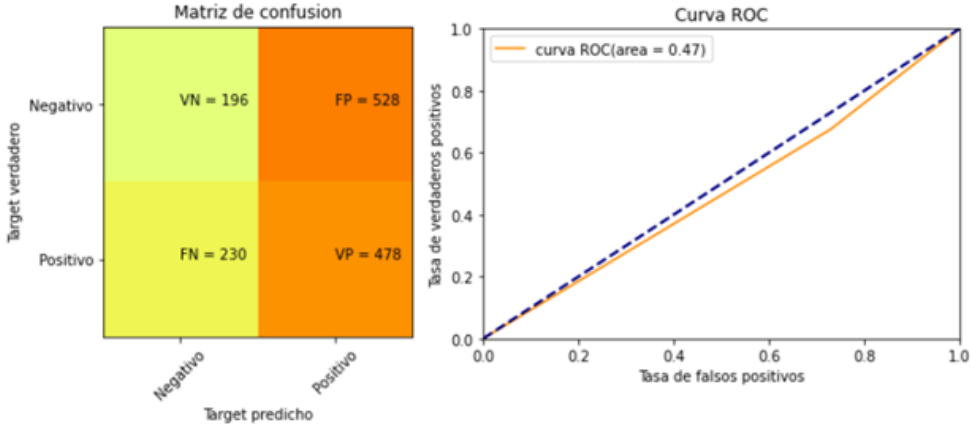


Figura: Resultados de algoritmo SVM-DRM para un $\gamma = 30\,\%$

Resultados modelos bancarios SVM-DRM-memoria corta

Gama γ	ROC-Score	N° positivos	Función objetivo
10 %	0.26	661	0.0406
20 %	0.25	624	0.5350
30 %	0.37	433	0.7625
40 %	0.30	736	0.1066
50 %	**0.51**	**859**	**0.1353**
60 %	0.27	713	0.2405
70 %	0.29	777	0.2865
80 %	0.29	768	0.2236
90 %	0.32	657	0.1404
100 %	0.34	518	1

Cuadro: Cuadro resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas, con $|V_t| = 5$. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.

Modelo Bancario SVM-DRM-memoria corta

Mejor modelo de generado con un conjunto de 5000 y $|V_t| = 5$

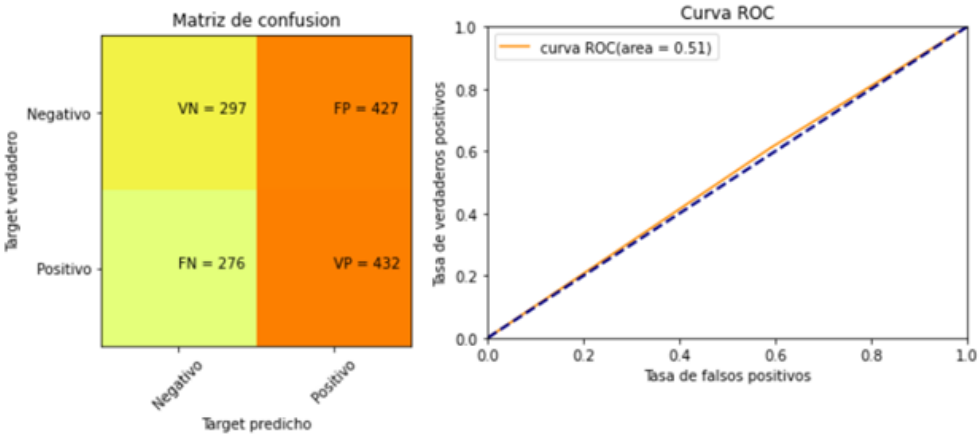


Figura: Resultados de algoritmo SVM-DRM para un $\gamma = 30\,\%$

Resultados modelos bancarios SVM-DRM-memoria corta

Gama γ	ROC-Score	N.º positivos	Función objetivo
10 %	0.33	554	0.3422
20 %	0.36	937	0.5271
30 %	0.32	411	0.6052
40 %	**0.42**	**478**	**0.2033**
50 %	0.32	571	0.8811
60 %	0.27	912	0.0965
70 %	0.31	799	0.2064
80 %	0.30	742	0.6485
90 %	0.37	855	0.3525
100 %	0.26	650	1

Cuadro: Cuadro resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas, con $|V_t| = 8$. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.

Modelo Bancario SVM-DRM-memoria corta

Mejor modelo de generado con un conjunto de 5000 y $|V_t| = 8$

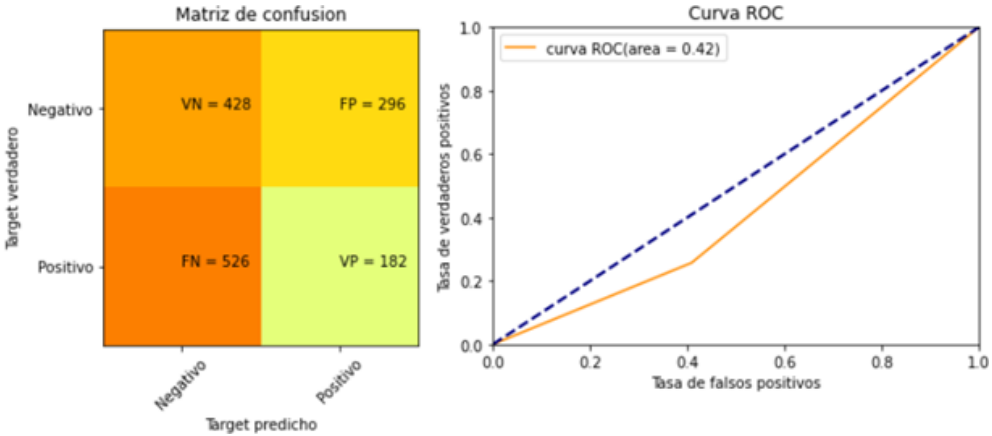


Figura: Resultados de algoritmo SVM-DRM para un $\gamma = 40\,\%$

Outline

Conclusiones

- El mejor de los modelos de clasificación probados fue XGBoost, para la base de datos bancaria. El cual quedo implementado en la entidad bancaria.
- Generamos una variante de Support Vectort Machines la cual es robusta ante perturbaciones con el algoritmo de Diametrical Risk Minimization.
- Los modelos robustos de SVM-DRM, pueden ser utilizados para entender de mejor forma el problema, desde una nueva perspectiva, en la cual podemos identificar mejor los comportamientos de algunos prospectos, con los resultados, se mejora la toma de decisión sobre este grupo de público.

Trabajo Futuro

- Utilizar otros algoritmos de clasificación supervisados tal como Catboost.
- Utilizar otro tipo de algoritmos, tales como redes neuronales para realizar la clasificación bancaria.
- Generar las variantes de DRM para los otros modelos expuestos.
- Probar nuevos sets de datos de la vida real con nuestra técnica SVM-DRM.
- Estudiar las métricas para evaluar el rendimiento de los modelos robustos.

¡Gracias por su atención!

Outline

Definiciones ERM

- Condición Polyak-Lojasiewicz. Fije X y sea f^* la que denota el mínimo valor de f en X. Diremos que una función f satisface la condición de Polyak-Lojasiewicz en X sí existe $\epsilon > 0$ tal que para todo $x \in X$ se tiene:

$$\frac{1}{2} \parallel \nabla f(x) \parallel^2 \leq \epsilon(f(x) - f^*)$$

- Condición Crecimiento Cuadrático. Diremos que una función f satisface la condición de crecimiento cuadrático en X, si existe un $\epsilon > 0$ tal que para todo $x \in X$ se tiene

$$f(x) - f^* \leq \frac{\epsilon}{2} \parallel x_p - x^* \parallel^2$$

donde x_p denota la proyección euclidiana de x dentro del conjunto de mínimos globales de f en X.

Outline

Convergencia DRM

Teorema (Ratio de Convergencia DRM): Suponga que $W \subset \mathbb{R}^n$ es compacto, $l : \mathbb{R}^n \times Z \longrightarrow \mathbb{R}$ es una función de Caratheodory localmente sup-integrable, y para todo $w \in W$, $l(w, \cdot) - R(w)$ es subgausina. Entonces para cualquier $\alpha \in (0, 1)$, $\gamma > 0$ y un m, existe un $\beta > 0$ (independiente de m) tal que:

$$\mathbb{P}^m \left(\sup_{w \in W} [R(w) - R_m^{\gamma}(w)] \leq \beta m^{\frac{-1}{2}} \right) \geq 1 - \alpha$$

Outline

Solución modelo Iris DRM-SVM-memoria corta

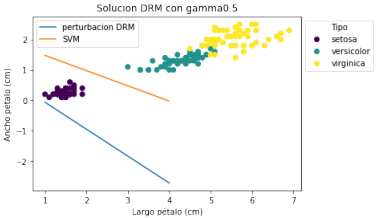


Figura: Solución SVM-DRM con $\gamma = 0{,}5 \cdot \| w \|$

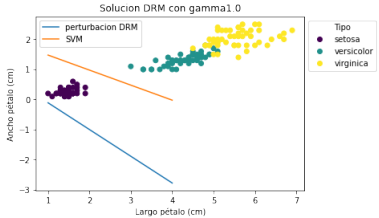


Figura: Solución SVM-DRM con $\gamma = 1{,}0 \cdot \| w \|$

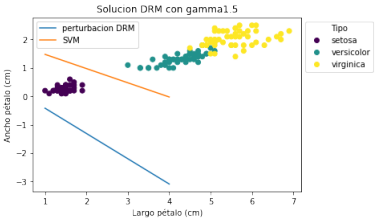


Figura: Solución SVM-DRM con $\gamma = 1{,}5 \cdot \| w \|$

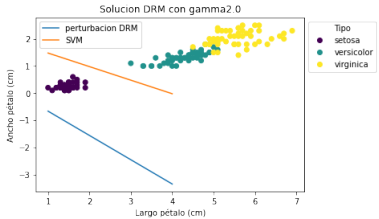


Figura: Solución SVM-DRM con $\gamma = 2{,}0 \cdot \| w \|$

Solución modelo Iris DRM-SVM-memoria corta

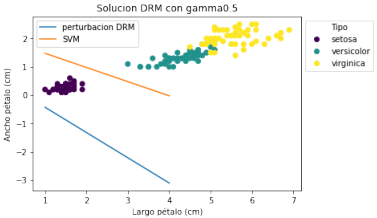


Figura: Solución SVM-DRM con $\gamma = 0{,}5 \cdot \| w \|$

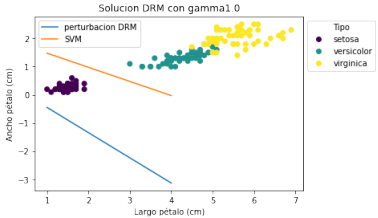


Figura: Solución SVM-DRM con $\gamma = 1{,}0 \cdot \| w \|$

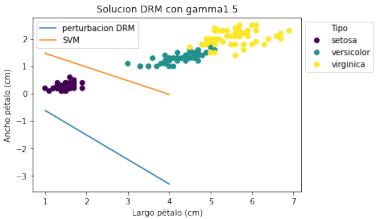


Figura: Solución SVM-DRM con $\gamma = 1{,}5 \cdot \| w \|$

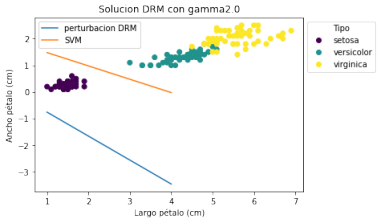


Figura: Solución SVM-DRM con $\gamma = 2{,}0 \cdot \| w \|$

Outline

Resultados para un grupo de 4400

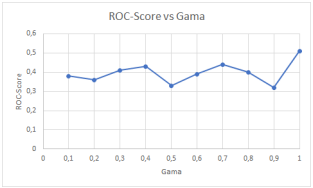


Figura: Gráfico ROC vs. γ

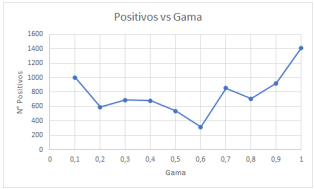


Figura: Gráfico N. Positivos vs. γ

Figura: Gráfico valor función objetivo vs. γ

Resultados para un grupo de 5000

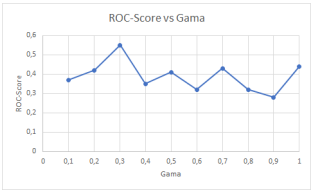


Figura: Gráfico ROC vs. γ

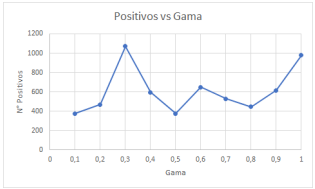


Figura: Gráfico N. Positivos vs. γ

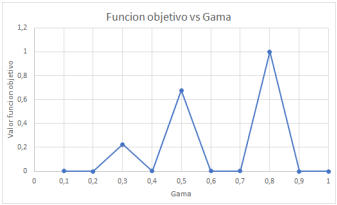


Figura: Gráfico valor función objetivo vs. γ

Resultados SVM-DRM-memoria corta $|v_t| = 2$

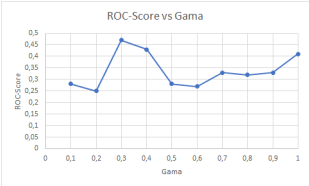


Figura: Gráfico ROC vs. γ

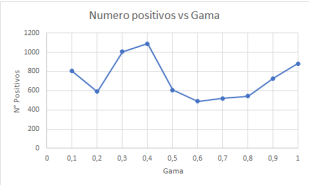


Figura: Gráfico N. Positivos vs. γ

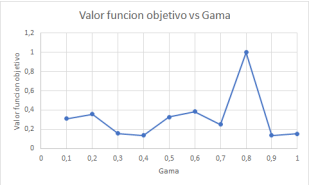


Figura: Gráfico valor función objetivo vs. γ

Resultados SVM-DRM-memoria corta $|v_t| = 5$

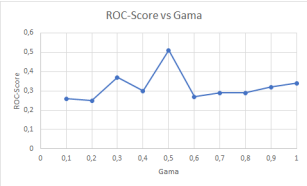


Figura: Gráfico ROC vs. γ

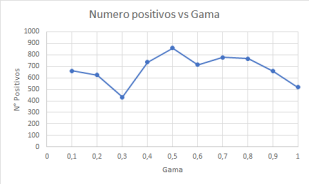


Figura: Gráfico N. Positivos vs. γ

Figura: Gráfico valor función objetivo vs. γ

Resultados SVM-DRM-memoria corta $|v_t| = 8$

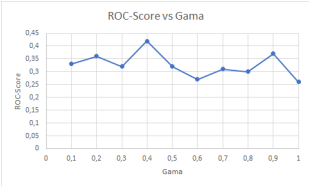


Figura: Gráfico ROC vs. γ

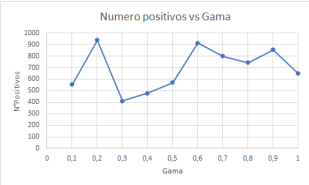


Figura: Gráfico N. Positivos vs. γ

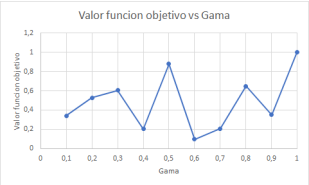


Figura: Gráfico valor función objetivo vs. γ