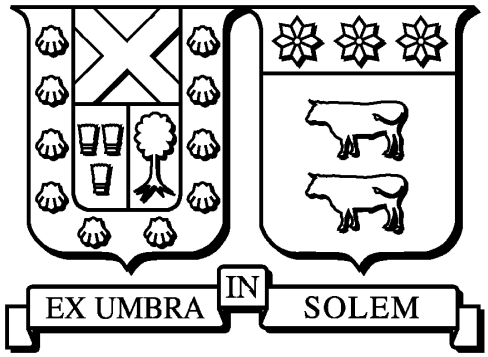
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Departamento de Matemática
Santiago-Chile



Predicción y clasificación de curse de un beneficio bancario

Memoria presentada por:
Juan Francisco Briceño Figueroa

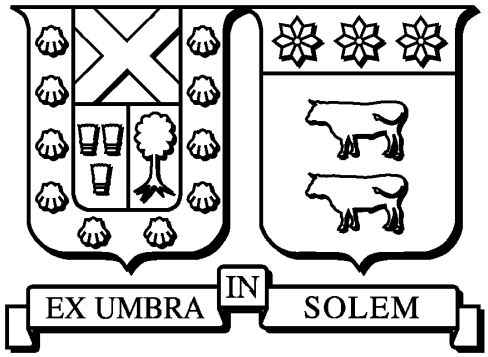
Como requisito parcial para optar al título profesional Ingeniero Civil Matemático

Profesor Guía:
Julio Deride

Profesor Co-Guía:
Francisco Alfaro

Diciembre 2021

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Departamento de Matemática
Santiago-Chile



Predicción y clasificación de curse de un beneficio bancario

Memoria presentada por:
Juan Francisco Briceño Figueroa

Como requisito parcial para optar al título profesional Ingeniero Civil Matemático

Profesor Guía:
Julio Deride

Profesor Co-Guía:
Francisco Alfaro

Examinadores:
Julio Deride
Francisco Alfaro

Diciembre, 2021

TÍTULO DE LA MEMORIA:
Predicción y clasificación de curse de un beneficio bancario.

AUTOR: Juan Briceño Figueroa.

TRABAJO DE MEMORIA, presentado como requisito parcial para optar al título profesional Ingeniero Civil Matemático de la Universidad Técnica Federico Santa María.

COMISIÓN EVALUADORA:

Integrantes	Firma
Julio Deride Universidad Técnica Federico Santa María, Santiago.	_____
Francisco Alfaro Universidad Técnica Federico Santa María, Santiago.	_____

Santiago, Diciembre 2021.

Resumen

En este trabajo estamos interesados en resolver y analizar el problema de predecir el comportamiento de un consumidor y su posterior clasificación. Dentro de este problema estudiaremos a cada uno de los prospectos y buscaremos el algoritmo que mejor pueda trabajar con nuestros datos, dichos algoritmos tienen raíz en herramientas entregadas por Machine Learning. Luego de encontrar la mejor solución procederemos a robustecer uno de los modelos, bajo el algoritmo de Diametrical Risk Minimization.

En la primera parte del problema, propondremos y evaluaremos 3 algoritmos que son utilizados hoy en día para clasificar prospectos con las métricas ROC-Score y F1-Score. Usaremos los algoritmos con los datos obtenidos de una empresa bancaria [1]. Según nuestros resultados empíricos, nuestro trabajo hay resultados más significativos cuando utilizamos algoritmos que tienen como base la optimización de una función objetivo (ver [25],[22],[2]). También adoptaremos una nueva perspectiva de uno de los algoritmos en base a la optimización de los parámetros. Proponemos la optimización desde el punto de vista que nos entrega Diametrical Risk Minimization para el algoritmo de Support Vector Machines, el cual nos entregara resultados robustos para nuestros parámetros, dentro de nuestro estudio nos entregaría una ayuda positiva en la distribución de las matrices de confusión entregadas por la clasificación. Finalmente, mostraremos que nuestro algoritmo evidencia una mejora en la distribución de las matrices de confusión, la cual ayudaría a entregar una mejora en la toma de decisión por parte de la empresa bancaria.

Palabras Clave: Clasificación de prospectos, Machine Learning, Predicción de prospectos, Diametrical Risk Minimization.

[1] Los datos con los que trabajamos son sacados de bases de datos reales de la identidad bancaria.

Agradecimientos

Me gustaría agradecer a Julio Deride, mi profesor guía en esta memoria, quien aceptó ayudarme durante este año en este trabajo. Gracias por todos sus comentarios, aportes, y conocimientos para con este trabajo, con lo cual he podido perfeccionar mis habilidades y renovó mi interés por el área que decidí estudiar. También me gustaría agradecer a Francisco Alfaro, mi profesor co-guía, quien me empezó a apoyar desde mi practica para crecer como profesional y como persona.

También agradecer al Departamento de Matemática de la Universidad Técnica Federico Santa María Campus San Joaquín, a todos sus profesores, quienes me aportaron de gran manera tanto dentro como fuera de la sala de clases. Agradecer además a los distintos departamentos en los cuales tuve clases, sus funcionarios, profesores, estudiantes.

A mi familia, a mis padres, mi hermano Andres Briceño y mis abuelos por todos estos años de apoyo incondicional y ayuda en todo lo que ha sido necesario para poder lograr pequeños y grandes objetivos personales y familiares.

Me gustaría agradecer a mis compañeros dentro de la universidad, saber que lo más importante son las relaciones que se generan, el compañerismo y la amistad entre nosotros. Un agradecimiento especial a Francisca Román, Gonzalo Gacitua, Gonzalo Arias, Fabian Rubilar, Javier Pizarro, Nathaly Corrales y Javier Díaz por el apoyo de cada uno tanto académico como personal en todo momento.

Índice general

Capítulo 1

Introducción

El área bancaria ha tenido un interés por entender los cambios de cada uno de los usuarios para poder entregar un servicio de forma correcta. Desde el inicio, el análisis estadístico y la predicción de comportamientos mediante el uso de modelos matemáticos se ha transformado en algo esencial dentro de cada área en cada banco.

En este trabajo nos centraremos en un problema de uso normal dentro de esta área, el cual intentaremos resolver desde tres perspectivas distintas. La primeras perspectiva es utilizar una técnica basada en la inferencia estadística y minimización de parámetros estadísticos para encontrar la solución del problema. Luego seguiremos con más técnicas basadas en la optimización de funciones para la solución de dicha problemática. Finalmente, mediante una nueva técnica optimizaremos los parámetros, para así poder obtener resultados más robustos dentro del abanico completo de soluciones.

El problema principal consiste en predecir el uso o no de un tipo de beneficio que entrega la institución bancaria[1]. En principio es posible pensar que la solución de este tipo de problemas es común, ya que la problemática es común, pero la solución es la que debe adecuarse según la situación que se enfrente. Dada las situaciones que hoy en día atraviesa Chile, es posible ver el nivel de importancia de tener un respaldo matemático para la toma de decisiones en el ámbito monetario.

Mejoraremos nuestros resultados en base a la idea de robustez generada por Ben-Tal[3]. La particularidad de las que buscamos es el entorno en donde optimizaremos. Lo común en estos problemas, es modificar la base de datos para optimizar el modelo, nuestro punto de vista es distinto, deseamos modificar el modelo para optimizar el problema con la base de datos que nos entrega la institución bancaria. Dicho método de resolución, entregara un modelo con otra perspectiva de clasificación, la cual beneficiaria a la institución, pues les dará una nueva solución a su problema, con la cual lleva a generar una nueva ruta ante la toma de una decisión.

1.1. Alcance de este trabajo

En este trabajo presentaremos un modelo de predicción al uso de un beneficio bancario y la mejora de las técnicas actuales mediante optimización de los modelos que presenten una estabilidad ante perturbaciones de la solución.

[1]Se reserva el nombre de la institución, por confidencialidad

En la primera parte de nuestro trabajo, investigaremos algunas de las técnicas que se tiene actualmente (gracias a machine learning), se adapta de una forma correcta a la solución de este problema. En particular trabajaremos con tres tecnicas de clasificacion de prospectos. La primera tecnica que nos ayudara a resolver el problema es *Random Forest*([13],[7],[6]). Luego seguiremos con un modelo de optimización básico como es *Suport Vector Machines* [23], luego procederemos con la ultima técnica llamada *Extreme Gradient Boosting* (la cual se abrevia XGBoost)[9]. En este trabajo, nosotros proponemos que las herramientas entregadas por los algoritmos Support Vector Machines y XGBoost tienen un mejor rendimiento que las herramientas entregadas por Random Forest.

Como segunda parte de este trabajo, investigaremos la posibilidad del cambio en la perspectiva de la clasificacion, generando una nueva solucion la cual servira de estudio para resolver el problema, dicha solucion a su vez sera robusta ante perturbaciones. En la literatura es posible encontrar técnicas que entregan dicha robustez al modelo (en el sentido que describio Ben Tal [3]), es así como encontramos la técnica *Diametrical Risk Minimization* (DRM)[24], la cual nos ayudaría en dicho cometido. El modelo que utilizaremos para generar la robustez de este mismo sera la tecnica de Support Vector Machines.

1.2. Trabajos relacionados

En la primera parte de nuestro trabajo describiremos el comportamiento en ventanas temporales de los usuarios de una identidad bancaria. Propondremos un modelo para el comportamiento de cada variable y como se ajusta a cada uno de los modelos predictivos seleccionados anteriormente, para así poder generar un buen modelo de clasificación predictivo.

En la literatura, el problema de "Data driven aproach to predict the success of use a bank benefit" es descrito como un caso en base a las herramientas dadas por distintos modelos clásicos tales como regresión logística y arboles de decisión. Estos modelos tienen una ventaja en interpretación visual de los resultados. Dado que en este trabajo se intentó utilizar modelos que tienen una naturaleza matemática distinta es esperable que los resultados difieran. Estos cambios inducirán un cambio en cómo resolver el problema (al ver la diferencia entre las variables que resulten de ser las más importantes) lo que nos entregara distintas perspectivas de la solución al problema. En los distintos trabajos que nos proponen Chiranjit Chakraborty y Andreas Joseph [8] es posible ver que dependiendo de las necesidades a resolver variaran las métricas a utilizar para medir el funcionamiento del algoritmo. Utilizaremos distintas métricas, en nuestro caso utilizaremos ROC-Score y F1-Score, estas métricas fueron seleccionadas pues, la métrica de ROC-Score nos entregara una medida del rendimiento de nuestro clasificador, en el sentido de si puede generar una buena discriminación de los datos. Por otro lado, la métrica del F1-Score nos entrega un valor que cuantifica la precisión del modelo.

En el trabajo de Chiranjit Chakraborty, Andreas Joseph [8], se proponen diversas estrategias para el estudio de problemas bancarios según predicción. Ellos presentan diversos puntos de vista para afrontar la predicción, tales como los modelos basados en árboles, Support Vector Machines, redes neuronales, etc, mostrando así que todas estas técnicas pueden ayudar en el día a día en el banco. Estas herramientas ayudan a mejorar la toma de decisiones, por ejemplo, el reconocimiento de patrones, puede ayudar a discernir a la institución, si el cliente es bueno o malo según los beneficios que estén buscando, como también estas técnicas pueden ayudar a generar predicciones sobre indicadores financieros futuros.

En la segunda parte de este trabajo, trataremos con el problema de robustez de soluciones. A partir de un macro análisis podemos ver que cada modelo del anterior problema se ira a justando de distinta forma según los datos que se le entregue y también los parámetros que se le den como input al modelo, esto nos hace pensar que cada una de estas soluciones será distinta y pueden tener una naturaleza distinta en el sentido de estabilidad. Es así como llegamos a pensar que, si optimizáramos la selección de los parámetros, los modelos encontrarían las mejores soluciones y además estas serían estables. Es por ello que se querrá estudiar el algoritmo de *Diametrical Risk Minimzation* para su implementación. Dicho algoritmo fue descrito por Matthew D. Norton, Johannes O. Royset, en el cual nos describen el trasfondo de la aplicación DRM[24], se basa en una optimización robusta de los parámetros entorno a bolas de radio definido para buscar soluciones estables en esta misma.

1.3. Contenido

En el capítulo 2 de datos y análisis descriptivo presentaremos la data que se utilizamos en todo el proyecto, se procedió a describir las variables las cuales según nuestro criterio forman parte en el comportamiento de cada uno de los prospectos. Presentaremos un análisis detallado de las variables que a nuestro criterio mostraban una variabilidad importante, para poder generar una limpieza de estas (variables con menos ruido).

En el capítulo 3, presentamos el modelo de propensión a la toma de producto bancario, se procedió a introducir el problema desde 2 puntos de vista, el primero es explicar el tipo de problema que se desea resolver y posteriormente se mostró el problema desde el punto de vista matemático. Se muestran los algoritmos que utilizamos para trabajar en la resolución de este, luego se procedió a presentar las métricas de evaluación que fueron utilizadas para mejorar el rendimiento de cada uno de los modelos. También presentaremos los resultados obtenidos para cada una de las técnicas anteriormente descritas.

Por último, en el capítulo 4 de robustez de las soluciones, procederemos a introducir el problema y una solución la cual será entregada por el algoritmo *Diametrical Risk Minimization* en conjunto con la técnica estadística de *Support Vector Machines*. Puesto que los datos que se utilizaron en este capítulo son los mismos que antes, procederemos a estudiar la solución propuesta por el equipo de investigadores, explicando matemáticamente la mejora que se deseamos implementar. Luego se procedió a mostrar el algoritmo y los resultados obtenidos en ellos.

Capítulo 2

Datos y Análisis descriptivo

En este capítulo presentaremos los datos que serán utilizados en los capítulos posteriores, también mostraremos el análisis descriptivo de los datos en general y el análisis que se pudo generar entorno a las direcciones que deseamos guiar el modelo.

2.1. Conjunto de datos

Utilizaremos los datos entregados por una institución bancaria. Dado que dicha identidad nos pidió anonimizar las bases de datos, las etiquetas de las variables fueron transformados a números, también los datos fueron desnaturalizados según reglas que nos propuso la misma identidad bancaria. Las siguientes categorías son necesarias para entender las variables a utilizar en el modelo:

- Descripción del sujeto: en esta sección se encontrarán los datos que hagan alusión a las características sociales del sujeto tales como, la edad, el sexo, el código identificador, el segmento de riesgo (añadido por la identidad), estado civil, si es un trabajador del banco, segmento homogéneo al cual pertenece, si es una persona deudora, clasificación de la tarjeta de crédito, tipo de tarjeta de crédito (si es gold, platinum, black, entre otros), la cantidad de productos que tiene contratados con el banco y periodo. Los indices asociados a esta categoría son:

$$1, 2, 3, 4, 7, 8, 9, 39, 40, 41, 42, 43, 44, 65, 76, 79, 80, 81, 89, 90$$

- Eventos a predecir: Estudiaremos 3 distintos tipos de eventos para poder predecir el suceso que esperamos:
  - Target 1: Este evento hace referencia a generar un nexo entre la gente que pidió un avance para pagar la deuda internacional, nacional o ambas en todos los periodos disponibles en los datos, las reglas que debían ocurrir son, el monto de avance debe ser mayor o igual a el pago de la tarjeta con hasta una diferencia de un 20 % entre ellos y el plazo máximo entre avance y pago debe ser de 5 días. El indice es 73.
  - Target 2: Si cursaron o no la oferta de este beneficio bancario (avance para pagar la deuda). El indice es 74.
  - Target 3: Combinación de el target 1 y target 2 sin intersección entre ellos. El indice es 75.

- Productos tarjeta de crédito: en esta categoría colocaremos todas las variables que dependan del uso de la tarjeta de crédito, tales como compras hechas con la tarjeta de crédito, avances pedidos con la tarjeta, cantidad de compras hechas en un mes, saldo adeudado en la tarjeta de crédito, cupo adeudado con la tarjeta de crédito (el cual puede encontrarse diferenciado por internacional, nacional o completo), cantidad de usos de tarjetas de créditos (compras, giros, pagos), cupo disponible para avances (internacional, nacional), deuda total (internacional, nacional), disponible para utilizar (internacional, nacional), numero de cuotas que tiene en el mes, número de días entre avance (con sus máximos entorno al mes, mínimos entorno al mes y total), monto máximo de avance solicitado por la persona, número máximo de cuotas solicitadas por avance. Los indices asociados a esta categoría son:

  $$5, 6, 23, 24, 25, 30, 31, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 66, 67, 68, 69, 70, 71, 72.$$

- Otros productos: en esta categoría podremos encontrar las variables que hacen referencia a otros productos utilizados por la persona en el banco, tales como, el uso de la cuenta corriente, depósitos a largo plazo, créditos hipotecarios, fondos mutuos, facturas, línea de crédito, giros solicitados por cajeros (con su respectivo monto), compras realizadas con el medio de pago débito, share of wallet, datos generales del usuario en forma bancaria (datos superintendencia de bancos e instituciones bancarias), créditos de consumo, datos de moras (cantidad , número de días, cantidad de monto, etc.), renta general y pagos automáticos (PAT, PAC y PAS). Los indices asociados a esta categoría son:

  $$10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 26, 27, 28, 29, 32, 33, 34, 35, 36, 37, 38, 59,$$

  $$60, 61, 62, 63, 64, 77, 78, 83, 84, 85, 86, 87, 88, 91, 92, 93, 94, 95, 96.$$

Estos datos fueron extraídos de las bases de datos de la identidad bancaria, los cuales fueron desnaturalizados para el uso en el exterior de la compañía. La data histórica se remonta desde octubre del año 2019 hasta abril del año 2021. La cantidad de variables fueron 96.

Luego de la extracción de datos y estudio de los potenciales casos de estudio se generaron 3 bases (target_1, target_2 y target_3) las cuales tienen volumetrías distintas, pero las 3 son equilibradas entre casos positivos y negativos, la base de datos target_1 tiene un tamaño de 11524 prospectos, la base de datos target_2 tiene un tamaño de 7158 y la base de datos target_3 tiene un tamaño de 11750.

Analizando el problema y la estructura de los datos, procedimos a generar un estudio de panel de sujetos en los distintos periodos. Haciendo un análisis de los sucesos demarcados como gatillo o target anteriormente descritos, fue posible producir un estudio estadístico en torno a los 6 meses, de la siguiente forma:

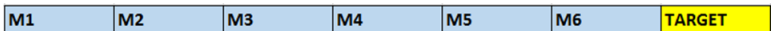


Figura 2.1: Análisis temporal

En conjunto con estos datos se generaron variables que estadísticamente tuvieran una relevancia a la hora de describir el suceso, los cuales se encuentran descritos en el cuadro2.1

Nombre	Tipo	Descripción
Prom_u3m	Numérico	Promedio de los últimos 3 meses
Prom_u6m	Numérico	Promedio de los últimos 6 meses
Min_u3m	Numérico	Mínimo en los últimos 3 meses
Min_u6m	Numérico	Mínimo en los últimos 6 meses
Max_u3m	Numérico	Máximo en los últimos 3 meses
Max_u6m	Numérico	Máximo en los últimos 6 meses
Dif_p_u3m	Numérico	Diferencia porcentual en los últimos 3 meses
Dif_p_u6m	Numérico	Diferencia porcentual en los últimos 6 meses
V_dism_u3m	Numérico	Variación de disminución en los últimos 3 meses
V_dism_u6m	Numérico	Variación de disminución en los últimos 6 meses
V_aum_u3m	Numérico	Variación de aumento en los últimos 3 meses
V_aum_u6m	Numérico	Variación de aumento en los últimos 6 meses
Tend_u3m	Numérico	Tendencia de los últimos 3 meses

Cuadro 2.1: Resumen de las variables. Puesto que las variables que utilizaremos son generadas de manera mensual, es necesario generar dicho resumen para poder comprender el comportamiento alrededor de los meses colindantes.

Para el estudio descriptivo del problema, utilizaremos las siguientes librerías de Python:

- Pandas[1]: Esta librería permite convertir datos en dataframes para facilitar el análisis consultando según los parámetros necesarios. Utilizaremos la version 1.3.4.
- Numpy[2]: Numpy es una potente librería para cálculo numérico y para trabajar con álgebra lineal, para la realización de procedimientos de forma optimizada. Utilizamos la verion 1.2.4.
- Matplotlib[3]: Matplotlib es una biblioteca completa para crear visualizaciones estáticas, animadas e interactivas en Python. Utilizamos la version 3.1.2.
- Seaborn[4]: Seaborn es una biblioteca de visualización de datos de Python basada en matplotlib. Proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos e informativos. Utilizamos la version 0.11.2.

2.2. Análisis descriptivo

En esta sección describiremos los datos, en la subsección 2.2.1 entregaremos un análisis estadístico básico junto con las correspondientes correlaciones entre las variables, en la subsección 2.2.2 entenderemos las distribuciones de las variables con mayor variabilidad, por último, en la sección 2.3 daremos nociones del comportamiento temporal de las variables más importantes. A partir de esta información es posible ver que modelo será más certero para resolver nuestro problema.

[1] http://pandas.pydata.org/
[2] http://www.numpy.org/
[3] https://matplotlib.org/
[4] https://seaborn.pydata.org//

2.2.1. Análisis Numérico

Para nuestros conjuntos de datos en la sección 2.1, se tiene un historial de 5890586 registros en total, con alrededor de 218170 clientes mensualmente. Ahora analizaremos de una forma atemporal los datos con estadística descriptiva lo cual nos entregara un primer acercamiento a estos.

Para nuestro primer grupo de variables se obtuvo el cuadro 2.2 con información sobre las distintas variables[5]:

Variable	Cuenta	Promedio	Desviación std.	Mínimo	Máximo
1	5890555.0	4.479463e+05	2.582329e+05	6.715760e+05	8.951840e+05
3	5890586.0	1.000000e+00	0.000000e+00	1.0	1.000000e+00
5	5890419.0	7.895771e-02	5.159651e-01	0.0	9.300000e+01
7	5890419.0	4.353681e+01	1.322796e+01	0.0	1.210000e+02
10	5890419.0	1.013053e+00	7.225024e-01	0.0	2.290000e+02
11	5887992.0	2.824837e+06	9.790619e+06	-39674044.0	3.046774e+08
12	5890419.0	2.808447e-01	1.306517e+00	0.0	1.330000e+02
13	691110.0	2.914375e+07	6.146954e+07	10023.0	1.275396e+09
14	5890419.0	2.747684e-01	5.872730e-01	0.0	3.700000e+01
15	1313815.0	6.311313e+07	4.330458e+07	1.0	2.570392e+08
16	5890419.0	1.643027e-01	7.228896e-01	0.0	8.000000e+01

Cuadro 2.2: Tabla descriptiva 1

Uno de los aspectos mas relevantes que se pueden observar dentro de la primera tabla es que existen diferencias importantes en la cantidad de categorías no nulas en las variables $13\,y\,15$, lo cual nos ayuda a verificar su importancia dentro del grupo de personas testeados. Otro aspecto a notar es la desviación estándar de algunas variables, esto nos indicaría que estas casillas tienen una importancia a nivel numérico, pues nos pueden describir cambios grandes en los datos, hasta incluso afectar en los modelos que hemos propuesto como solución.

Para nuestro segundo grupo de variables se obtuvo el cuadro 2.3 con información sobre las distintas variables. Dentro de este grupo de variables descritas en el cuadro 2.3 nuevamente es posible ver que existen variables con una cantidad muy menor a la promedio (var $19, 22\,y\,27$). Nuevamente, es posible percibir que existen variables con una alta variabilidad, lo cual nos puede afectar en el modelo que estemos utilizando para modelar posteriormente. Además, otra característica notoria de esta tabla es la gran diferencia entre los valores predeterminados como mínimo y máximo (y hasta incluso con el promedio), esto nos puede ya dar una alusión de que la gente que utiliza estas variables está concentrada en un grupo pequeño.

Para nuestro tercer grupo de variables se obtuvo el cuadro 2.4 con información sobre las distintas variables. Para el cuadro 2.4 si analizamos la entrada cuenta, nuevamente es posible ver que existen variables que tienen poca información (menos del promedio), lo cual nos puede dar un grado de importancia dentro de la base de datos extraída. Si proseguimos viendo el cuadro 2.4, también es notorio ver que existen variables con un grado de varianza alto en comparación a la mayoría, lo cual puede tener efectos en los modelos que utilizaremos posteriormente para resolver el problema

[5]Se omitieron algunas variables, porque hacen alusión a variables no numéricas y datos binarios los cuales no son relevantes al momentos de generar la tabla descriptiva.

Variable	Cuenta	Promedio	Desviación std.	Mínimo	Máximo
17	558161.0	1.712980e+07	4.046178e+07	0.51	393575000.0
18	5890419.0	2.465020e-04	2.878789e-02	0.00	8.0
19	630.0	1.482160e+08	1.563904e+08	196.00	839051151.0
20	5890419.0	9.588208e-01	2.651652e-01	0.00	4.0
21	5556382.0	2.920785e+06	3.368524e+06	1.00	56400000.0
22	1930607.0	1.658732e+06	2.784028e+06	1.00	57147485.0
23	5890419.0	1.199787e+00	4.724174e-01	0.00	42.0
24	5890419.0	5.612027e+06	6.119108e+06	0.00	500000000.0
25	3893271.0	1.814417e+06	2.323738e+06	0.08	12601879.0
26	5890419.0	1.258931e+00	2.587718e+00	0.00	99.0
27	2038759.0	2.305427e+05	2.738638e+05	1000.00	8730000.0
28	5890419.0	8.425582e+00	1.483178e+01	0.00	559.0
29	2702594.0	4.394588e+05	5.032284e+05	1.00	11479357.0
30	5890419.0	5.463511e+00	1.234585e+01	0.00	2739.0

Cuadro 2.3: Tabla descriptiva 2

Variable	Cuenta	Promedio	Desviación std.	Mínimo	Máximo
32	5890419.0	4.358139e-01	3.906521e-01	0.0	1.000000e+00
33	5890419.0	1.930937e-01	3.773475e-01	0.0	1.000000e+00
34	1047548.0	5.431761e+07	1.703815e+08	1000	5.632437e+09
35	5584979.0	1.322096e+07	1.760774e+07	1000	1.052710e+08
36	5890419.0	2.581780e+00	1.478375e+00	0.0	1.500000e+01
37	2761019.0	8.109616e+07	6.373238e+07	1000	3.473500e+08
38	5861004.0	2.280738e+07	2.962945e+07	1000	8.421770e+08
39	5889798.0	6.852867e-03	8.249791e-02	0.0	1.000000e+00
40	5889798.0	3.499188e-01	4.769441e-01	0.0	1.000000e+00
41	5767902.0	8.616098e-01	3.453091e-01	0.0	1.000000e+00
42	5767902.0	8.798502e-02	2.832731e-01	0.0	1.000000e+00
43	5767902.0	2.550511e-02	1.576534e-01	0.0	1.000000e+00
44	4676229.0	2.717029e+00	1.130219e+00	1.0	4.000000e+00
45	2416971.0	3.395724e+06	3.123773e+06	1.0	1.696250e+07
46	2433986.0	4.130452e+06	3.760981e+06	735.	2.249100e+07

Cuadro 2.4: Tabla descriptiva 3

propuesto. Ahora si vemos las diferencias entre máximo y mínimo, es posible ver que hay variables que destacan por su gran diferencia, estas variables también contienen un alto grado de varianza, lo cual nos puede ayudar a encontrar distintos prospectos dentro de la base de datos extraída.

Para nuestro cuarto grupo de variables se obtuvo el cuadro 2.5 con información sobre las distintas variables. Analizando el cuadro es posible denotar que la mayoría de las variables no contienen a todos los prospectos, lo cual nos ayuda a ver que no todas las personas dentro de nuestro espacio de estudio contienen características que nos parecía importante a analizar, notemos que todas las variables con una cuenta baja (menor al promedio) son variables que pertenecen al grupo de tarjeta

Variable	Cuenta	Promedio	Desviación std.	Mínimo	Máximo
47	2391863.0	3.742605e+06	3.105272e+06	1.00	1.734000e+07
48	2406416.0	4.093441e+06	3.728457e+06	735.00	2.205000e+07
49	2464880.0	1.047597e+06	2.277651e+06	-23097220	2.547316e+08
50	710701.00	8.034290e+04	1.793700e+05	7.35	1.655066e+06
51	2281652.0	2.884248e+06	2.650342e+06	1.00	1.471006e+07
52	2401850.0	4.063548e+06	3.709769e+06	7.35	2.199121e+07
53	1629500.0	1.310336e+06	1.520228e+06	1.00	7.027809e+06
54	180928.00	3.588015e+06	3.653433e+06	7.35	1.950176e+07
55	2283869.0	2.884568e+06	2.649925e+06	1.00	1.471222e+07
56	2403059.0	4.062425e+06	3.709106e+06	610.05	2.198679e+07
57	5890586.0	5.369597e+04	4.207116e+05	0.00	1.393577e+08
58	5890586.0	1.298677e+04	1.452706e+05	0.00	2.995007e+07
59	5890586.0	2.112365e+01	1.719335e+02	0.00	1.720600e+04
60	5890586.0	6.784123e-01	5.527446e+00	0.00	1.800000e+02
61	5890586.0	1.999584e+05	2.809441e+06	0.00	1.271612e+09

Cuadro 2.5: Tabla descriptiva 4

de crédito, lo cual nos lleva a pensar que en su mayoría la gente no tiene tarjeta de crédito activa, lo cual será nuestro primer filtro de entrada al estudio posterior el cual culminara con las bases de datos necesarias para generar el modelo. Si seguimos analizando el cuadro 2.5, es posible ver que nuevamente la mayoría de las variables tienen una desviación alta, la cual tendremos que estudiar desde otra perspectiva para que no entorpezcan a el modelo matemático. Una característica importante, es que dentro de esta tabla se encuentra una variable con un mínimo negativo, lo cual no es extraño, ya que nos encontramos dentro de las variables de la categoría *producto tarjeta de crédito*, en las cuales se puede dar ese evento.

Para nuestro quinto y último grupo de variables se obtuvo el cuadro 2.6 con información sobre las distintas variables[6]. Estudiando el cuadro, es posible ver principalmente que es la primera que contienen a todas las variables completas (es decir que nuestra base de datos aplica en todos estos datos), lo cual nos ayuda a ver que estas variables son normales en la mayoría de los usuarios de la identidad bancaria. Siguiendo con la tabla 2.6 es posible ver que la desviación estándar de todas las variables es bajo, lo cual nos ayuda a percibir que estas variables no nos agregarían ruido a nuestro modelo lo cual es bueno.

[6]Dentro del grupo de variables que se pueden analizar sin faltar a los acuerdos de confidencialidad.

Variable	Cuenta	Promedio	Desviación std.	Mínimo	Máximo
62	5890586.0	0.018996	0.137421	0.000	5.000
63	5890586.0	0.284878	1.298335	0.000	44.000
64	5890586.0	4.709881	22.131135	0.000	510.000
65	5890586.0	0.083237	0.276240	0.000	1.000
66	5890586.0	3.209127	25.539458	0.000	830.000
67	5890586.0	3.085804	25.284411	0.000	830.000
68	5890586.0	2.435162	23.076179	0.000	830.000
70	5890586.0	0.134492	1.822003	0.000	48.000
72	5890586.0	0.120215	1.712063	0.000	48.000
73	5890586.0	0.002646	0.051372	0.000	1.000
74	5890586.0	0.000710	0.026629	0.000	1.000
75	5890586.0	0.002668	0.051584	0.000	1.000
76	5890586.0	1.412497	0.492284	1.000	2.000
77	5890586.0	5.215053	10.293371	0.000	1437.000
79	5890586.0	1.484783	0.719523	1.000	3.000

Cuadro 2.6: Tabla descriptiva 5

2.2.2. Análisis de distribución

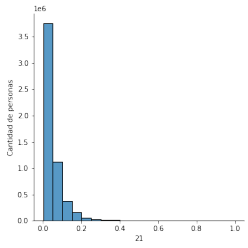
Ya con nuestro primer análisis, es posible ver la calidad de las variables que tenemos disponible para nuestro modelo, pero gracias a lo que pudimos notar anteriormente en cada una de las tablas, necesitaremos estudiar la distribución de un grupo de variables (las cuales tienen el mayor grado de varianza[7]), para poder entender si esta nos entorpecerá en nuestros modelos matemáticos posteriores.

Iniciaremos mostrando las primeras variables con su distribución, la cual se encuentra en la figura 2.2

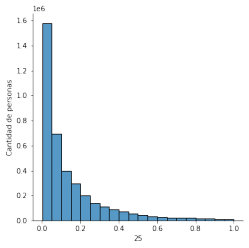
A partir de estas estructuras es posible ver que, las distribuciones asociadas a este grupo de variables con poca gente, están concentrada entorno a el valor 0, es así como podemos pensar que no son variables descriptivas en torno a la masa general, pero si pensamos en la variabilidad de estas mismas, es posible notar que estas variables pueden tener una distribución diferente en los valores mayores que 0 lo cual nos podría dar más información sobre estas mismas y así poder incluirlas dentro de nuestro set de variables a utilizar. Es así como llegamos a la conclusión de que lo importante seria entender la relación que describen los valores estrictamente positivos (pues denotarían un uso dentro de estas), para ello se filtró y se hizo una transformación no lineal de los datos, los resultados de esto se pueden ver en el siguiente grupo de variables transformadas (figura 2.3.

Si analizamos de forma global, las distribuciones de las variables adoptan las formas características de una distribución normal, con ello es posible también observar que la varianza de las variables disminuyo, a su vez esta distribución puede tener características que deseemos para nuestros modelos, pues la mayoría de los modelos les es fácil trabajar con este tipo de datos.

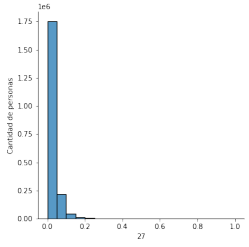
[7]El grupo de variables a estudiar en esta sección es reducida, pues intentaremos describir solo las más importantes para nuestros modelos.



(a) Variable 21.

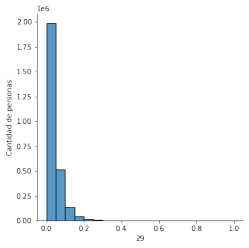


(b) Variable 25.

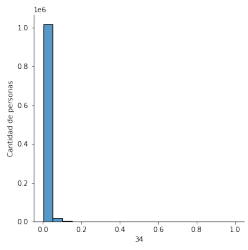


(c) Variable 27.

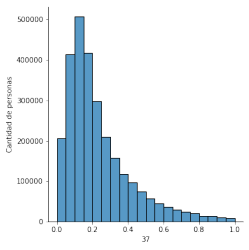
fig:f3



(d) Variable 29.

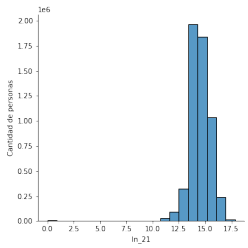


(e) Variable 34.

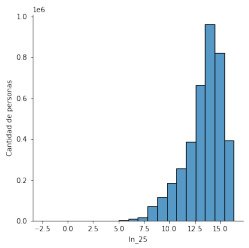


(f) Variable 37.

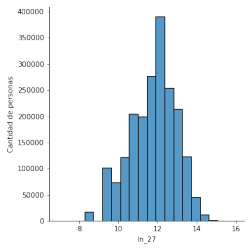
Figura 2.2: Primeras variables ruidosas.



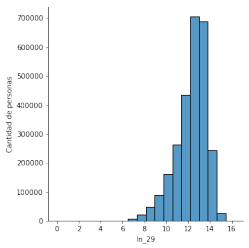
(a) Logaritmo natural variable 21.



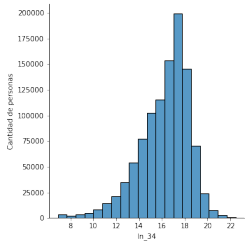
(b) Logaritmo natural variable 25.



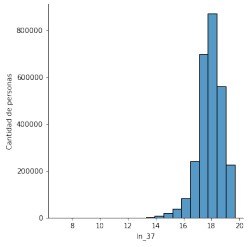
(c) Logaritmo natural variable 27.



(d) Logaritmo natural variable 29.



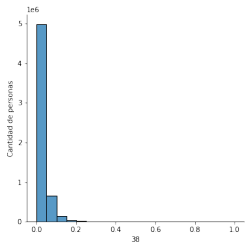
(e) Logaritmo natural variable 34.



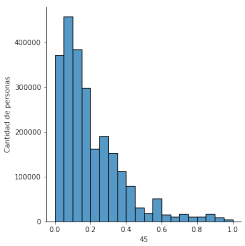
(f) Logaritmo natural variable 37.

Figura 2.3: Transformación logaritmo primeras variables ruidosas.

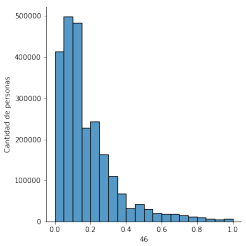
Si nuevamente las variables descritas en la figura 2.4 y en la figura 2.5 y analizamos de forma global las estructuras, es posible ver que existe nuevamente un suavizamiento en la estructura y una transformación en las distribuciones de las variables, pues nuevamente se adoptan distribuciones normales, con un suavizamiento de la varianza, lo cual nos ayudaría a modelar nuestros modelos.



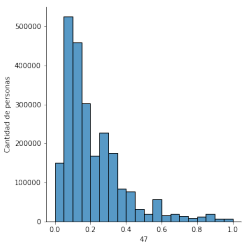
(a) Variable 38.



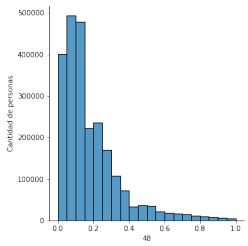
(b) Variable 45.



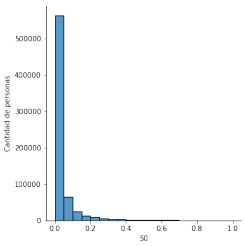
(c) Variable 46.



(d) Variable 47.

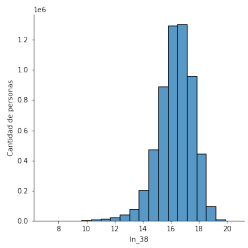


(e) Variable 48.

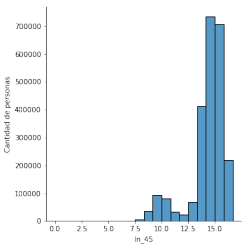


(f) Variable 50.

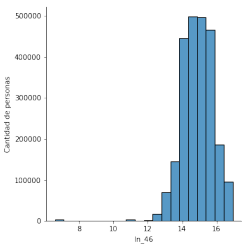
Figura 2.4: Segundas variables ruidosas.



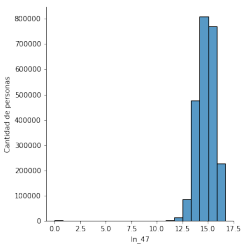
(a) Variable 38.



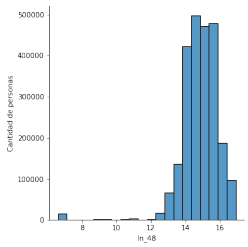
(b) Variable 45.



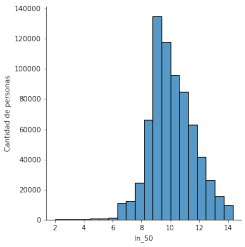
(c) Variable 46.



(d) Variable 47.



(e) Variable 48.



(f) Variable 50.

Figura 2.5: Transformación logaritmo segundas variables ruidosas.

2.3. Análisis histórico

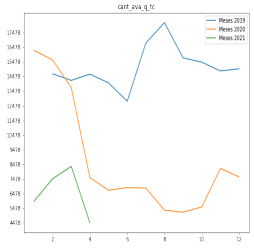
A partir de los conjuntos de datos descritos en la sección 2.1, estudiaremos un grupo de variables para ver el comportamiento a lo largo de los meses disponibles. Este estudio se realizó en base a la cantidad de personas que recurrió a la cantidad máxima mensual de cada uno de las variables, con esto podríamos analizar el comportamiento que mueve a la masa importante de nuestros prospectos.[8]

[8]Puesto que estas variables no hacen alusión a los valores reales referenciales, podremos analizar de una mejor forma identificando el beneficio con los valores utilizados.

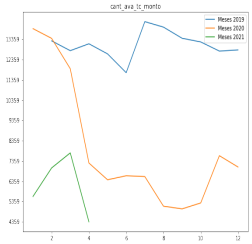
En cada imagen se pueden ver 3 líneas las cuales describen:

Color	Azul	Naranjo	Verde
Año	2019	2020	2021

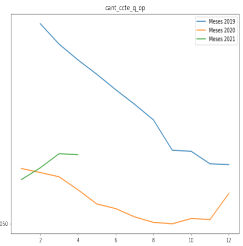
Cuadro 2.7: Colores y años



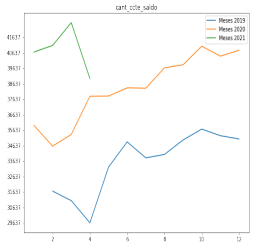
(a) Beneficio cantidad de avances.



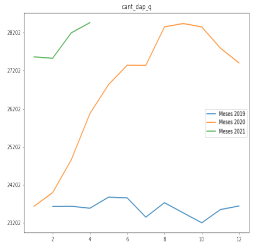
(b) Beneficio monto de avance solicitado.



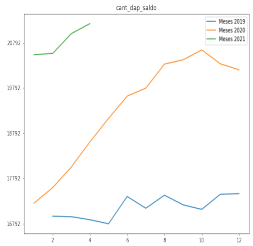
(c) Operaciones realizadas con cuenta corriente.



(d) Saldo cuenta corriente.



(e) Cantidad de depósitos a largo plazo (DAP).



(f) Saldo depósitos a largo plazo (DAP).

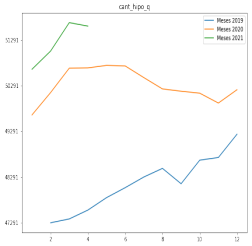
Figura 2.6: Primeras variables serie de tiempo.

Si analizamos las figuras 2.6a y 2.6b, según los colores y años descritos en el cuadro 2.7, es posible ver que el año 2019 es el año con mayor cantidad de avances (en cantidad y en cantidad monetaria), pero es posible ver que la transición de año $2019 - 2020$ empieza un descenso sin fin, esto suponemos que es un efecto generado por la pandemia, lo cual se puede dar porque no hay instancias de desorden monetario líquido. Es también importante mencionar que ambas figuras están fuertemente relacionadas pues, describen 2 variables que son directamente proporcionales.

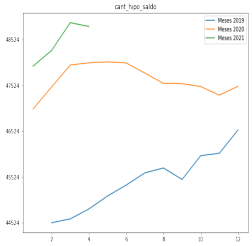
Si procedemos con las figuras 2.6c y 2.6d, y según los colores establecidos anteriormente, es posible ver que en la figura 2.6c el año 2019 sigue teniendo la mayor cantidad de prospectos con máximo lo cual nos muestra un año normal, la tendencia a la disminución a lo largo del año es normal entre los años $2019 - 2020$, teniendo los menores valores en el año 2020, lo cual nuevamente podemos identificar el hecho de la época de pandemia, pero tiende a la alza en los meses de diciembre 2020 hasta marzo de 2021, esto se debe a que en estas fechas las reglas de movilidad fueron menores que en el año 2020 y esto pudo generar un incremento en las operaciones realizadas físicamente, agregándose a las de base que ya se tenían. Si analizamos la figura 2.6d podemos ver que los saldos máximos en la cuenta corriente se dan en el año 2021, esto se debe a que a lo largo del año 2020 como no existió un incentivo a consumir, el prospecto de persona empezó a ahorrar (esto es claro también ya que la curva naranja tiende a la alza a lo largo del año 2020 después del inicio de las

cuarentenas).

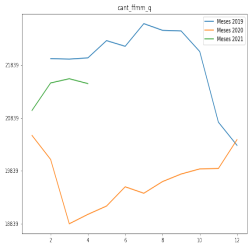
Estudiando las figuras 2.6e y 2.6f, que hacen referencia a la cantidad y monto de los depósitos a largo plazo (DAP) es posible percibir que ambas variables poseen gráficos similares, esto se debe a que estas variables son directamente proporcionales. Nuevamente es posible ver que el año 2019 es el año con menor cantidad de ambas variables, lo cual es posible pues el comportamiento de los usuarios promedio se describe como poco conservador dentro de la institución bancaria, luego es posible percibir que en el año 2020, el comportamiento de los usuarios tendió a ser más conservador, pues a fines del año 2019 existió un evento que genero este comportamiento, lo cual tiende a inestabilidad, tendió a ajustar el comportamiento de los usuarios.



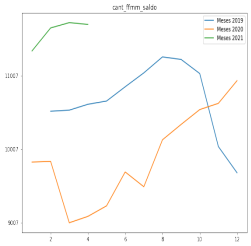
(a) Cantidad de créditos hipotecarios.



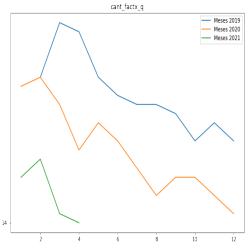
(b) Saldo de créditos hipotecarios.



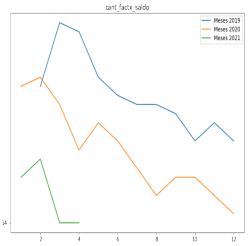
(c) Cantidad de fondos mutuos.



(d) Saldo fondos mutuos.



(e) Cantidad de facturas.



(f) Saldo de facturas.

Figura 2.7: Segundo grupo de variables serie de tiempo.

Analizando la figura 2.7, es posible ver que las variables descritas en 2.7a y 2.7b tienen gráficos similares, lo cual es efecto de que ambas variables son directamente proporcionales entre ellas. Es posible rescatar de estos gráficos, que ambas variables tienen una tendencia al alza desde al año 2019, estancándose en el segundo semestre del año 2020, luego es posible ver que los usuarios vuelven a tener un alza en ambas variables en el inicio del año 2021 lo cual es normal para este tipo de variables (por los periodos navidad, año nuevo y vacaciones, los cuales tienen una alza importante de fluidez monetaria).

Estudiando las variables descritas en las figuras 2.7c y 2.7d, como primer análisis es posible ver que tienden a desordenarse en los 3 periodos y no es posible denotar algún patrón, esto se debe a la alta inestabilidad económica que vive el país a fines del año 2019 (en gran parte se debe a el estallido social) y luego por la pandemia. Es importante mencionar que, en ambos casos, a fines del año 2021 tienden a tener un claro repunte y un incremento sustancial, lo cual nos entrega información sobre la estabilidad monetaria inicial de fines del periodo, lo cual es muy alentador pues está creciendo de forma positiva.

Examinando las ultimas 2 figuras 2.7e y 2.7f, podemos ver que ambos gráficos son parecidos,

esto se debe a que ambas variables son directamente proporcionales. De ambos gráficos es posible desprender que la cantidad de facturas (con su saldo correspondiente) fue a la baja desde el año 2019 hasta el año 2021, dentro de esta variable es posible ver que no fue golpeada por los efectos de la pandemia, ni tampoco por los efectos generados por el estallido social, pues en todo el espacio temporal tiende a tener las mismas características mensuales. De estas variables podemos también podemos apreciar que entorno a los 3 años es posible ver que en el inicio de los 2 primeros meses tiene un alza y luego tiende a decrecer a lo largo del año.

A partir de análisis realizados en las sección 2.2 y en la sección 2.3, fue posible evidenciar que el entorno temporal de cada una de estas variables era importante para poder entender los hechos que llevaran a un cliente a tomar la decisión que deseamos predecir, es por esto que se decidió generar las variables atemporales que nos pudieran entregar dichos resúmenes (ver cuadro 2.1).

Capítulo 3

Predicción Modelo Propensión

En este capítulo introduciremos los modelos de predicción, en donde describiremos los algoritmos y las métricas a utilizar. Como primer puntapié introduciremos los antecedentes del problema y como lo atacaremos. Luego mostraremos los algoritmos que serán utilizados para resolver nuestro problema junto con las métricas que nos ayudaran para todos los casos en la sección 3.3. Finalmente en la sección 3.4 mostraremos los resultados obtenidos para nuestros datos.

3.1. Antecedentes

Los modelos de propensión generan una predicción de que suceda un evento o eventos el cual podría ocurrir dentro de un tiempo dadas las condiciones que se tienen hoy. La propensión y la clasificación de prospectos tienen un uso importante dentro de la industria bancaria [8], puesto que es útil para mantener el nivel de servicio alto y a los distintos clientes contentos. Estos tipos de modelos no son utilizados solo dentro de la industria bancaria , por ejemplo, puede ser utilizado por un grupo de marketing para predecir la probabilidad de que un potencial cliente se transforme en un verdadero cliente, o que un cliente se retire de la institución en la cual se encuentra, o hasta incluso que un destinatario dé de baja el servicio de suscripción de correo electrónico, entre los más importantes. Esto nos ayuda a poder mantener a los clientes dentro de la institución y con un interés alto en los productos que se desea entregar. Nosotros modelaremos nuestro problema estudiando a cada uno de los clientes dentro de un marco temporal de 6 meses en los cuales se espera que entre en alguna de las categorías anteriormente descritas. La problemática se formó en base a ciertas conductas que fueron estudiadas por la institución, estos mismos se dieron cuenta que existían relaciones muy estrechas entre personas que recurrían a un beneficio bancario (avances por lo general) cuando ya estaban en problemas monetarios (endeudados o morosos) para pagar su deuda y así poder volver a estar en una categoría de no riesgoso para la institución, con lo cual podrían mantener los beneficios otorgados por esto mismo (tales como tarjeta de crédito por ejemplo). Esto llevo al banco a generar un producto que incluyera ambas necesidades y con tasas distintas a las que antes eran sometidos los clientes, dicha oferta es llamada 'Cuotaflex'. Luego de generar el producto, intentaron generar campañas que pudieran llegar a los distintos clientes, dichas campañas no tuvieron éxito. Esto genero una necesidad de parte del banco por generar un prospecto de persona a la cual se le va a ofertar y una predicción de quien sería el ideal para ofrecerle este beneficio. Con estos requerimientos fue necesario generar un modelo matemático que tuviera ciertas capacidades para discernir entre los distintos clientes.

Trabajaremos con 3 modelos distintos, los cuales son utilizados hoy en día, uno de ellos es una herramienta que se basa en un problema de optimización bajo restricciones (*Support Vector Machines*) y otros dos que dependen de árboles de decisión; uno de ellos se basa en la disminución de la varianza y disminución del sesgo en cada árbol (*Random Forest*), y por último un modelo de árboles que es potenciado por pasos de gradientes entre cada uno de los nodos de cada árbol (*Extreme Gradient Boosting*). Cada uno de estos modelos genera una propensión para la base de datos, luego de este proceso, compararemos los resultados obtenidos por cada uno de ellos según las métricas que fueron definidas.

- **Support Vector Machines**: Support Vector Machines Classifier (S.V.M) genera un hiperplano separador entre la muestra, de forma que cada uno de los prospectos queda en una de las categorías, posterior a esto le genera una puntuación a cada uno de los prospectos entorno a qué tan probable es que se encuentre en el lado fijado como positivo.
- **Random Forest**:Random Forest Classifier (RF) extrae distintos grupos dentro de la base de datos, con los cuales genera arboles de decisión de forma que cada uno de ellos sea independiente del otro. Posterior a generar este proceso, cada árbol de decisión genera una predicción para cada uno de los prospectos, luego todas estas predicciones se promedian entorno a la cantidad de árboles que se tienen. Finalmente, con este promedio se genera una predicción final de RF para cada uno de los sujetos entorno a la probabilidad de pertenecer al grupo demarcado como positivo (1).
- **Extreme Gradient Boosting**: Extreme Gradient Boosting (XGB) este también genera distintos extractos de una misma base de datos, con los que posteriormente genera arboles de decisión independientes unos de otros, pero estos árboles están controlados por pasos de gradiente y una función extra que le da mayor estabilidad al algoritmo para su funcionamiento. Posteriormente cada uno de los árboles genera su propia predicción, la cual en el paso final se promediará con los demás resultados obtenidos por el resto de árboles. Concluyendo con una predicción final para cada prospecto entorno a la probabilidad de pertenecer al grupo denotado como positivo.

3.2. Algoritmos

Tal como mencionamos en 3.1, utilizaremos 3 algoritmos distintos para resolver nuestro problema. Los cuales procederemos a describir a continuación.

3.2.1. Random Forest Classifier

Como primer punto, es necesario definir que es un arbol de decisión para entender la técnica de Random Forest. 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 (por ejemplo, si el lanzamiento de una moneda sale cara o cruz), cada rama representa el resultado de la prueba y cada nodo hoja representa una etiqueta de clase (decisión tomada después de calcular todos los atributos). Las rutas de la raíz a la hoja representan reglas de clasificación[15].

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[1] que construye una larga colección de árboles no correlacionados entre ellos y luego los promedia. Cada árbol es construido usando el algoritmo 1.

Algorithm 1 Árbol de decisión

1.-Sea N el número de casos para prueba, M el número de variables en el clasificador.

2-Sea m el número de variables de entrada a ser usado para determinar la decisión en un nodo dado; por lo tanto $m << M$.

3.-Elegir un conjunto de entrenamiento para este árbol y usar el resto de los casos de prueba para estimar error.

4.-Para cada nodo del árbol, se elige aleatoriamente m variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las m variables.

La idea esencial del bagging [6] es promediar muchos modelos ruidosos, pero un tanto imparciales, y así reducir la varianza de estos. Los árboles predictores son candidatos ideales para el proceso de bagging, dado que ellos pueden registrar estructuras con interacción compleja en los datos, y si estos crecen suficientemente, tienen relativamente una baja parcialidad. Producto de que los árboles son notariamente ruidosos [14], ellos se benefician enormemente al promediarlos. Además, dado que cada árbol generado con bagging es idénticamente distribuido, la esperanza del promedio de M arboles es la misma que en cada uno de estos. Esto significa que el sesgo de los arboles con baggin es el mismo para cada uno de ellos, y la única manera de mejorarlo es bajo la reducción de varianza de estos mismos. Esto es en contraste a el boosting, donde los arboles crecen de una forma adaptativa para remover el sesgo y por lo tanto no son i.i.d.

Un promedio de M variables aleatorias i.i.d, cada una con varianza σ^2, tiene varianza $\frac{1}{M}\sigma^2$. Si la varianza es simplemente idénticamente distribuida pero no necesariamente independiente con correlación positiva a pares ρ, la varianza del promedio es:

$$\rho\sigma^2 + \frac{1-\rho}{M}\sigma^2$$

Si M crece, el segundo termino desaparece, pero el primero se mantiene, y por lo tanto el tamaño de la correlación a pares de los arboles con bagging, limita los beneficios de promediar. La idea de los Random Forest es mejorar la reduccion de la varianza del bagging reduciendo la correlacion entre los arboles, sin incrementar mucho la varianza. Esto se logra con el proceso de crecimiento de arboles mediante la selección aleatoria de las variables de entrada.

Para la predicción, un nuevo caso es generado por el arbol para el crecimiento de este mismo. Luego se le asigna la etiqueta del nodo terminal donde finaliza. Este proceso es iterado por todos los árboles en el ensamblado, y la etiqueta que obtenga la mayor cantidad de incidencias es reportada

[1]Bagging: es una técnica usada para reducir la varianza de las predicciones a través de la combinación de los resultados de varios clasificadores, cada uno de ellos modelados con diferentes subconjuntos tomados de la misma población.

como la predicción, luego de obtener todos los árboles predictores, los resultados se promedian y se tiene la predicción final. Es así como la función objetivo asociada a Random Forest ([7],[13]) es:

$$Fun_{bag}(x) = \frac{1}{M}\sum_{b=1}^{M}\hat{f}_b(x) \tag{3.1}$$

donde $\hat{f}_b(x)$ es la estructura de cada árbol independiente. Dicha función está controlada por el error de clasificación y otros 2 tipos de medida de la varianza total a lo largo de las clases seleccionadas [15]:

- *Criterio de Gini*: El Criterio de Gini se calcula mediante la siguiente formula:

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

  la que denota la varianza total en M clases. No es difícil ver que el índice de Gini toma valores pequeños si ${\hat{p}_{mk}}^2$ son cercanas a cero o cercanas a uno. Es por esto por lo que se refieren al índice de Gini como una medida de pureza en los nodos, es decir, un valor pequeño indica que un nodo contiene predominantemente observaciones de una sola clase.

- *Entropía*: El criterio de Entropía esta dado por:

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

  dado que, $0 \leq \hat{p}_{mk} \leq 1$, se sigue que $0 \leq -\hat{p}_{mk}\log\hat{p}_{mk}$. Se puede mostrar que la Entropía toma valores cerca del cero si los $\hat{p}_{mk}$ se encuentran cerca del cero o cerca del uno. Por lo tanto, como el índice de Gini, la entropía tomara valores pequeños si el nodo mk es puro (en el mismo sentido que se definió la pureza de el Criterio de Gini).

Ambos índices son bastante similares entre sí. Cuando se crean los árboles de clasificación, el índice de Gini o el índice de Entropía son típicamente usados para evaluar la calidad de cada uno de los splits, ya que estos 2 son más sensibles ante la pureza de los nodos que el error de clasificación. Con esto es posible ver la distribución de varianza generada en cada uno de los árboles para cada una de las variables, lo cual nos ayudaría a clasificar los modelos o entender más aun los modelos y como genero esta clasificación.

3.2.2. Support Vector Machines Classifier

Esta técnica genera un hiperplano que pueda subdividir a la muestra en 2 regiones, una en donde se identifiquen los casos positivos y otra con los casos negativos. Los clasificadores S.V.M, tienen distintos kernels que utilizan para mejorar la clasificación dependiendo de la muestra con la que se desee entrenar el modelo, para esta sub-sección daremos las nociones básicas de un modelo de Support Vector Machines con un kernel lineal [5].

[2] $\hat{p}_{mk}$ representa la proporción de observaciones de entrenamiento en la m-esima región que se encuentra en la k-esima clase

Si disponemos de una muestra de datos clasificados

$$S = \{(x_i, y_i) \forall i = 0, .., n, y_i = -1, 1\},$$

entonces lo que se desea es genera un hiper plano separador que verifique:

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

Siempre es posible definir w' y b de manera que:

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

Si definimos el Margen como:

$$\text{marg} \, = \frac{\min_i(y_i(w' x_i + b)}{\parallel w \parallel} = \frac{1}{\parallel w \parallel} \tag{3.2}$$

Ahora lo que deseamos es buscar el hiperplano que maximiza el margen, para ello se debe resolver el siguiente problema convexo:

$$\min_{w \in W} \frac{\parallel w \parallel}{2} + C \cdot \sum_i^n \max(0, 1 - y_i * (x_i' * w + b)) \tag{3.3}$$

$$\text{s.a.} \, y_i(w' x_i + b) \geq 1$$

Dado que es casi imposible encontrar hiperplanos que sean perfectos, se generó una variante que es más robusta y tiene una mejor capacidad predicativa al aplicarlo a nuevas observaciones, los Soft Margin Support Vector Machines Classifiers[1]. Para lograr esta mejora, en vez de buscar el margen de clasificación más ancho posible que consigue que las observaciones estén en el lado correcto, se permite que ciertas observaciones estén en el lado incorrecto del margen o incluso del hiperplano. Dicha variante, solo reemplaza la restricción anterior por:

$$\text{s.a.} \, y_i(w' x_i + b) > 1,$$

para algún i en el conjunto. Por lo tanto, agrega una variable de holgura como penalización por cada clasificación incorrecta para cada punto de datos representado por β. Por lo tanto, que no exista penalización significa que el punto de datos está clasificado correctamente, $\beta = 0$, y en cualquier clasificación errónea $\beta > 1$, hay una penalización [23]. Para nuestro propósito, generaremos un método de Soft Margin SVM creado por nosotros, también utilizaremos los modelos entregados por Scikit-learn para generar una solución mas certera que la nuestra. Ambos métodos, están descritos en el anexo A con sus respectivas demostraciones de convergencia.

3.2.3. Extreme Gradient Boosting

Los métodos de árboles boosting son unas de las herramientas más efectivas y utilizadas hoy en día como métodos de machine learning[16].

El boosting[32] consiste en combinar los resultados de varios clasificadores débiles para obtener un clasificador robusto. Cuando se añaden estos clasificadores débiles, se lo hace de modo que estos

tengan diferente peso en función de la exactitud de sus predicciones. Luego de que se añade un clasificador débil, los datos cambian su estructura de pesos: los casos que son mal clasificados ganan peso y los que son clasificados correctamente pierden peso. Así, los clasificadores débiles se centran de mayor manera en los casos que fueron mal clasificados por los clasificadores débiles. La idea detrás del boosting es generar múltiples modelos de predicción "débiles" secuencialmente, y que cada uno de estos tome los resultados del modelo anterior, para generar un modelo más "fuerte", con mejor poder predictivo y mayor estabilidad en sus resultados.

Para conseguir un modelo más fuerte a partir de estos modelos débiles, se emplea un algoritmo de optimización, este caso Gradient Descent (descenso de gradiente). Durante el entrenamiento, los parámetros de cada modelo débil son ajustados iterativamente tratando de encontrar el mínimo de una función objetivo, que puede ser la proporción de error en la clasificación, el área bajo la curva (AUC), la raíz del error cuadrático medio (RMSE) o alguna otra. Cada modelo es comparado con el anterior. Si un nuevo modelo tiene mejores resultados, entonces se toma este como base para realizar modificaciones. Si, por el contrario, tiene peores resultados, se regresa al mejor modelo anterior y se modifica ese de una manera diferente. Qué tan grandes son los ajustes de un modelo a otro es uno de los hiper parámetros que debe definir el usuario. Este proceso se repite hasta llegar a un punto en el que la diferencia entre modelos consecutivos es insignificante, lo cual nos indica que hemos encontrado el mejor modelo posible, o cuando se llega al número de iteraciones máximas definido por el usuario.

Extreme Gradient Boosting (XGBoost) [9] sigue el principio de árboles de decision boosting potenciados por gradientes, pero este a su vez se especializa en su formulación, dado que agrega a la función de perdida un término regularizador para controlar el sobre ajuste, lo que induciría un mejor rendimiento [4]. La función objetivo a optimizar es:

$$Fun(\phi) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \tag{3.4}$$

donde:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)\,,\ f \in F \tag{3.5}$$

En la función 3.4, el término $L(\cdot, \cdot)$ indica la función de perdida que es generada por la predicción y el dato observado, y el termino Ω indica nuestra función regularizadora que disminuirá el sobre ajuste de los datos.

Entre las funciones de perdida más utilizadas están MSE y Logistic Loss, los cuales dependiendo del tipo de información y el tipo de problema podremos ajustar. Dicha función mide que tan predictivo es nuestro modelo con respecto a los datos de entrenamiento (datos observados).

Por otro lado, el termino regulador Ω controla la complejidad del modelo, lo cual nos ayuda a evitar el sobre ajuste. Dicha complejidad viene determinada por la siguiente formula:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \| \ w \ \|^2 \tag{3.6}$$

donde:

- 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:

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

Dicho valor encontrado nos dice que tan buena es nuestra estructura, siendo mientras menor sea el numero mejor es nuestra estructura.

3.3. Métricas de Evaluación

Para evaluar el desempeño de los algoritmos usados en la predicción de la propensión a usar este beneficio bancario, usamos 2 métricas diferentes, entre las métricas que se utilizarán están:

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

Los problemas de propensión se formulan comúnmente como problemas de categoría, donde el evento que desea predecir ocurre (positivo) o no ocurre (negativo). Si esto se representa en términos de un problema de machine learning, un caso positivo es donde el evento que desea predecir realmente ocurrió, por otro lado, un caso negativo es aquel en el que el evento no ocurre y no ocurre realmente. Dicho esto, se construyó una *matriz de confusión* para el problema de predicción de propensión como se puede ver en el siguiente cuadro 3.1.

	Positivo predicho	**Negativo predicho**
Positivo real	Verdadero positivo (TP)	Falso negativo (FN)
Negativo real	Falso positivo (FP)	Verdadero negativo (TN)

Cuadro 3.1: Matriz de confusión para los problemas de predicción

En el análisis estadístico de la clasificación binaria de *F1-Score* es una medida de la precisión de una prueba. Se calcula a partir de la prueba de precisión y recall. Es por esto que primero se definirá dichas pruebas para posteriormente presentar la formula de la métrica F1.

En la métrica *Precisión*, la cual cuantifica el número de predicciones positivas correctas realizadas. Se calcula como la proporción de éxitos predichos correctamente del número total de predicciones realizadas. Con el cuadro 3.1 se define como:

$$Precision = \frac{TP}{TP + FP} \tag{3.7}$$

En nuestro trabajo, calculamos la *Precisión* de todas las posibilidades en el grupo de prueba y las predicciones.

Recall es el número de resultados positivos verdaderos dividido por el número de todas las muestras que deberían haber sido identificadas como positivas. Con la tabla 3.1, se define como:

$$Recall = \frac{TN}{TN + FN} \tag{3.8}$$

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

$$F1\text{-}Score = \frac{2}{Recall^{-1} + Precisión^{-1}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3.9}$$

Para la segunda métrica, debemos definir la curva ROC. La *curva ROC* la cual ilustra la capacidad de un clasificador binario cuando se varía el umbral de discriminación[10].

La *curva ROC* se genera trazando la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FRP). Representa el nivel de separación entre dos distribuciones, una correspondiente a los verdaderos negativos y la otra correspondiente a los verdaderos positivos, dados los puntajes de un clasificador. La fórmula para TPR es: $TPR = \frac{TP}{TP+FN}$ y para FPR es $FPR = \frac{TP}{TN+FP}$ [10].

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. El *ROC-Score* tiene un valor máximo de 1, un puntaje alto corresponde a mejores resultados de clasificación. Un valor de 0,5 representa una opción sin capacidad de discriminación.

3.4. Resultados

En esta sección, compararemos las métricas definidas en 3.3 y generaremos .Todas estas medidas definidas como dijimos anteriormente, con 50 % del número total de entradas para cada subconjunto seleccionado como entrenamiento para nuestros modelos, y se usó el otro 50 % de estos subconjuntos como conjunto de prueba. Es importante recalcar que para cada uno de los set de datos (SD) se generó un modelo especializado, el cual se adaptó según los hiper parámetros que obtenían un mejor resultado para la métrica *F1-Score*. Posterior a el análisis de estos resultados, intentaremos estudiar cual de estos modelos es el que mejor discrimina la propensión a la toma del producto bancario mediante un análisis general de categorías (cada una de las categorías está compuesta de la misma cantidad de personas). Finalmente mostraremos el tiempo de ejecución de cada modelo y los recursos que utiliza su aplicación sobre el mejor set de datos.

Los mejores resultados de *F1-Score*, *Recall*, *Precisión* y *ROC-Score*, junto con el nombre de cada set de datos y el modelo con el cual se obtuvo dicho resultado, será mostrado en el cuadro 3.2.

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 3.2: Cuadro con los mejores resultados obtenidos, luego de testear todos los modelos generados con las bases de datos obtenidas para cada uno de los eventos asociados. Los mejores resultados del cuadro están remarcados con color rojo, esta selección se basó en, la selección de los mejores resultados dentro de las métricas y el estudio de la importancia de las variables asignado por cada uno de los modelos.

Luego de una revisión de cada uno de los modelos descritos en el cuadro 3.2, se procedió a seleccionar los mejores resultados por cada uno de los modelos. Esta selección se basó en el estudio de las variables seleccionadas como más influyentes dentro de cada modelo (dentro de los modelos que era posible hacer esto).

Nuestra meta es predecir de forma correcta el suceso de pedir el beneficio bancario en el mes futuro usando los algoritmos propuestos es sección 3.2. Podemos ver del cuadro 3.2, que los mejores modelos se basan en el set de datos 2, a partir de esto, se seleccionó la mejor variante de cada modelo, escogiendo según los mejores valores de *ROC-Score*, *F1-Score* y también según las variables más importantes descritas por cada variante, pues esto último nos ayudaría a entender si la clasificación se está guiando por los patrones que esperamos. Siguiendo este camino, los mejores modelos, en base a su rendimiento y las variables importantes descritas, fueron *Extreme Gradient Boosting 1*, seguido por el modelo *Support Vector Machines 2* y por ultimo *Random Forest 2*.

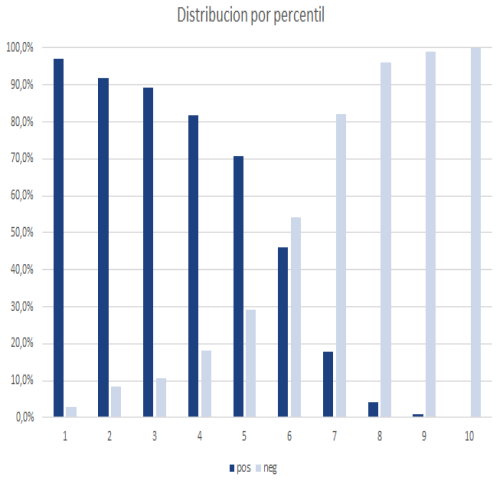
Para analizar los modelos, presentaremos un cuadro descriptivo en el cual podremos ver como es el comportamiento por cada categoría, es decir veremos la probabilidad promedio obtenida por el grupo, la cantidad de personas positivas en la categoría y el porcentaje de personas positivas por categoría.

Para nuestro modelo de *Extreme Gradient Boosting 1* el cuadro 3.3 muestra cómo se comportó el clasificador dentro de cada una de las categorías que se generaron, en la figura 3.1a podemos visualizar el comportamiento de las categorías entorno al porcentaje de positivos (resultados con el tag 1) y negativos (resultados con el tag 0) , por otro lado en la figura 3.1b la matriz de confusión obtenida por esta variante del modelo *Extreme Gradient Boosting*.

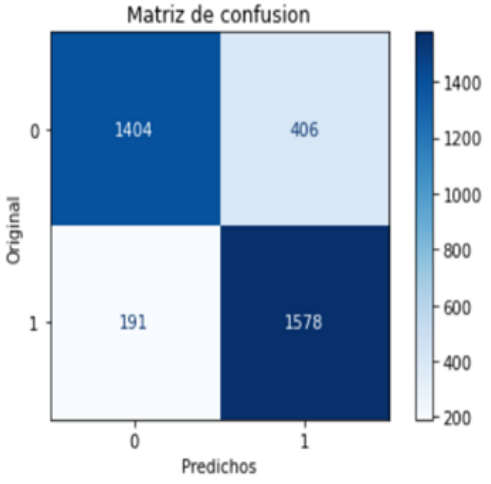
De acuerdo a la figura 3.1 es posible ver que existe una buena discriminación por parte del modelo *Extreme Gradient Boosting*, pues el total de personas es una forma decreciente (siendo el grupo homogéneo 1 el grupo con mayor tendencia a tomar el producto y el 10 el con menos

Grupo Homogéneo	*Promedio probabilidad*	*Total de personas positivas*	*Porcentaje 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 3.3: Cuadro con grupos homogéneos, obtenido de la clasificación realizada por el modelo XGB 1



(a) Distribución de personas positivas a lo largo de los grupos homogéneos.



(b) Matriz de confusión obtenida por el modelo XGB 1.

Figura 3.1: Resultados de clasificación obtenidos por Extreme Gradient Boosting 1.

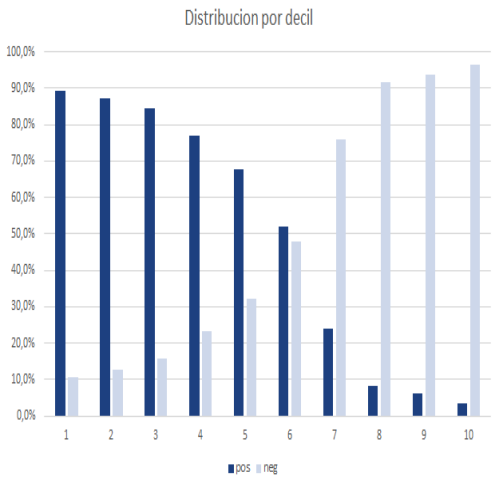
propensión) con respecto a la probabilidad y el total de personas positivas por grupo. Por otro lado, la matriz de confusión también nos dice que existe una buena discriminación por parte del modelo pues el mayor número de personas se distribuye entre los verdaderos positivos (TP) y los verdaderos negativos (TN).

Para nuestro modelo de *Support Vector Machines 2* el cuadro 3.4 muestra cómo se comportó el clasificador dentro de cada una de las categorías que se generaron, en la figura 3.2a podemos visualizar el comportamiento de las categorías entorno al porcentaje de positivos (resultados con el tag 1) y negativos (resultados con el tag 0) , por otro lado en la figura 3.2b la matriz de confusión obtenida por esta variante del modelo *Support Vector Machines*.

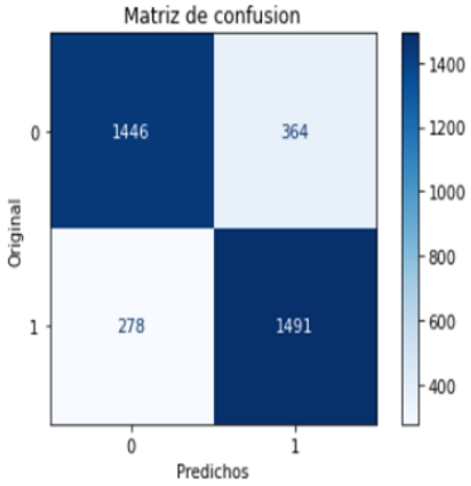
De acuerdo a la figura 3.2 es posible ver que existe una buena discriminación por parte del modelo *Support Vector Machines*, aunque la distribución del grupo homogéneo 6 la cual se ve en la figura

Grupo Homogéneo	*Promedio probabilidad*	*Total de personas positivas*	*Porcentaje 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 3.4: Cuadro con grupos homogéneos, obtenido de la clasificación realizada por el modelo svm 2



(a) Distribución de personas positivas a lo largo de los grupos homogéneos.



(b) Matriz de confusión obtenida por el modelo Support Vector Machines.

Figura 3.2: Resultados de clasificación obtenidos por Support Vector Machines 2.

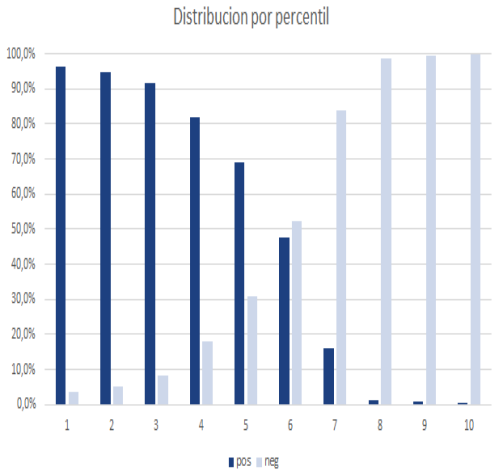
3.2a, tiene un comportamiento extraño pues en ese grupo ya debería existir un porcentaje mayor de casos negativos que de casos positivos. Por otro lado la matriz de confusión nos dice que existe una discriminación por parte del modelo pues el mayor número de personas se distribuye entre los verdaderos positivos (TP) y los verdaderos negativos (TN).

Para nuestro modelo de *Random Forest 2* el cuadro 3.5 muestra cómo se comportó el clasificador dentro de cada una de las categorías que se generaron, en la figura 3.3a podemos visualizar el comportamiento de las categorías entorno al porcentaje de positivos (resultados con el tag 1) y negativos (resultados con el tag 0) , por otro lado en la figura 3.3b la matriz de confusión obtenida por esta variante del modelo *Random Forest*.

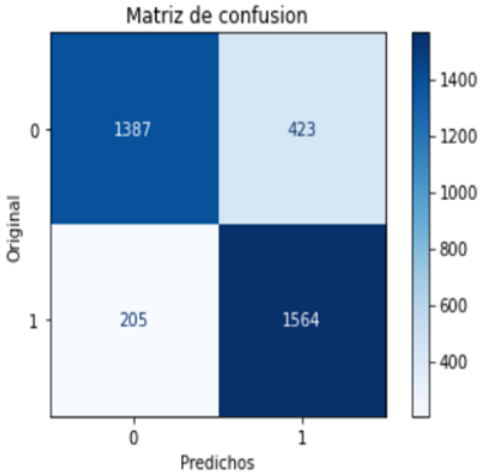
De acuerdo a la figura 3.3 es posible ver que existe una buena discriminación por parte del modelo *Random Forest*, pues no tiene ningún corte abrupto en la relación descendente de porcentajes a lo

Grupo Homogéneo	*Promedio probabilidad*	*Total de personas positivas*	*Porcentaje 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 3.5: Cuadro con grupos homogéneos, obtenido de la clasificación realizada por el modelo RF 2



(a) Distribución de personas positivas a lo largo de los grupos homogéneos.



(b) Matriz de confusión obtenida por el modelo Random Forest 2.

Figura 3.3: Resultados de clasificación obtenidos por Random Forest 2.

largo de las categorías de clasificación, lo cual se puede observar en la figura 3.3a. Por otro lado la matriz de confusión nos dice que existe un problema en la discriminación de falsos, pues tuvo una tendencia a aumentar la cifra de los falsos positivos por parte del modelo, pero la gran mayoría se distribuye entre los verdaderos positivos (TP) y los verdaderos negativos (TN).

Modelo	*Tiempo de ejecución*	*Memoria utilizada*
Extreme Gradient Boosting	8, 19s	577, 69Mb
Support Vector Machines	4min 25s	538, 32Mb
Random Forest	10, 1s	541Mb

Cuadro 3.6: Cuadro con los valores obtenidos por cada modelo en su ejecución.

Otro aspecto importante a mostrar, son los recursos que se utilizaron al generar estos modelos, para ello se tomó el tiempo de ejecución y los recursos utilizados de la máquina virtual que hizo el proceso, estos datos es posible observarlos en el cuadro 3.6.

Junto con el cuadro 3.6, es posible notar que el algoritmo *Extreme Gradient Boosting* es el que menos tiempo de ejecución tuvo, pero tuvo el mayor uso de recursos durante este tiempo, por otro lado el modelo de *Support Vector Machines* fue el que más demoro pero el que menos memoria utilizo dentro en su ejecución.

Capítulo 4

Robustez de las soluciones

4.1. Introducción

La estructura de la teoría del aprendizaje difiere de la de la mayoría de las otras teorías para problemas aplicados. La búsqueda de una solución a un problema aplicado generalmente requiere los siguientes pasos:

(I) Exprese el problema en términos matemáticos.

(II) Formular un principio general para buscar una solución al problema.

(III) Desarrollar un algoritmo basado en dicho principio general.

Los dos primeros pasos de este procedimiento no presentan en general mayores dificultades. El tercer paso requiere la mayor parte de los esfuerzos, en el desarrollo de algoritmos computacionales para resolver el problema que nos ocupa.

4.1.1. Problema de estimación de funciones

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

(I) Un generador de vectores aleatorios x, extraído independientemente de un fijo pero desconocido con distribución $P(x)$.

(II) 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 pero desconocida.

(III) Una máquina de aprendizaje capaz de implementar un conjunto de funciones $f(X, w)$, $w \in W$.

El problema del aprendizaje es el de elegir del conjunto dado de funciones el que se aproxima mejor a la respuesta del supervisor. La selección se basa en un conjunto de entrenamiento de n observaciones independientes:

$$S = \{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)\}$$

La formulación dada arriba implica que el aprendizaje corresponde al problema de aproximación de funciones. Para elegir la mejor aproximación disponible a la respuesta del supervisor, medimos

la pérdida o discrepancia $l(y, f(x, w))$ entre la respuesta y del supervisor a una entrada dada x y la respuesta $f(x, w)$ proporcionada por el aprendizaje. Considere el valor esperado de la pérdida, dado por el funcional de riesgo:

$$R(w) = \int l(y, f(x, w))dP(x, y)$$

En este capítulo estará dedicado a explicar el problema que deseamos resolver sobre la estimación de funciones y su solución mediante una nueva técnica llamada Diametrical Risk Minimization (DRM)[24]. Para entender mejor la técnica, comencemos explicando la técnica llamada Empirical Risk Minimization (ERM) [30].

4.1.2. Empirical Risk Minimization

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

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

Pero la distribución de probabilidad conjunta $P(x, y) = P(x|y)P(x)$ es desconocida y la única información disponible está contenida en el conjunto S anteriormente señalado.

Para resolver este problema, el siguiente principio de inducción es propuesto: el funcional de riesgo $R(w)$ es reemplazado por el funcional de riesgo empírico:

$$E(w) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i, w)), \tag{4.1}$$

el cual es construido en base al conjunto de entrenamiento. El principio de inducción de la minimización de riesgo (ERM)([30],[29]) asume que la función $f(x, w_l^*)$, la cual minimiza $E(w)$ sobre el conjunto $w \in W$, resulta en el riesgo $R(w_l^*)$, el cual es cercano a el mínimo.

El mínimo valor del riesgo empírico exhibe un sesgo bajo y por lo tanto, los minimizadores correspondientes son a menudo pobres en términos de su verdadero riesgo (poblacional) [29].

En lugar del riesgo empírico, 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. Esto le entrega a DRM una visión más amplia del panorama del riesgo empírico que ERM lo cual nos entrega como resultado una mejora en el rendimiento.

4.2. Diametrical Risk Minimization

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

$$R_m^\gamma(w) = \sup_{\|v\| \leq \gamma} R_m(w + v) = \sup_{\|v\| \leq \gamma} \frac{1}{m} \sum_{i=1}^{m} l(w + v, z_i) \tag{4.2}$$

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

Es posible ver que el riesgo diametral de un vector parámetro $w \in \mathbb{R}^n$ es el peor riesgo empírico posible en una vecindad de w. Cualquier norma puede usarse para definir la vecindad. Trivialmente, $R_m^0(w) = R_m(w)$ pero generalmente $R_m^\gamma \geq R_m(w)$. Entonces para un conjunto $W \subset \mathbb{R}^n$ de vectores parámetros permisibles, el *problema DRM* apunta a:

$$\min_{w \in W} R_m^\gamma(w) \tag{4.3}$$

lo cual tiene como resultado una solución w_m^γ, la cual puede ser un mínimo global, uno local, un punto estacionario o un vector parámetro con un bajo riesgo diametral.

El módulo de lipschitz [1] frecuentemente juega un rol importante en la teoría del aprendizaje estadístico. Por ejemplo, un paradigma de la teoría del aprendizaje es el análisis de la estabilidad del algoritmo. Sin embargo, la mayoría de estas cotas de aprendizaje requieren una noción de suavidad en términos de una función objetivo continua de Lipschitz y / o gradiente continuó de Lipschitz. El módulo de Lipschitz entra entonces en los límites de generalización e influye en la estabilidad (teórica) del algoritmo usado para generar ERM. La dependencia de esta y los otros límites de generalización en la magnitud del módulo de Lipschitz han generado diversas investigaciones. En el escrito de Diametrical Risk Minimization, entregan diversas cotas las cuales no depende del módulo de Lipschitz, lo cual aminoraría los errores de generalización.

El sesgo a la baja asociado al ERM, es conocido desde los primeros días de la optimización estocástica y los M-estimadores(ver las siguientes referencias para mayor entendimiento de optimización estocástica y M-estimadores respectivamente,[27][28]). Las soluciones tradicionales a este tipo de problemas, incluye unos variados esquemas de regularización, enfocados en la alteración de la función objetivo, o en la optimización del procedimiento mismo con, por ejemplo, paradas tempranas. Otro remedio es el reemplazo de ERM por el problema de minimizar la distribución del peor caso de riesgo empírico. Típicamente el peor caso es definido en términos de una bola en alguna métrica, en algún espacio de medidas de probabilidad centrada en la distribución empírica generada por los datos disponibles. El entrenamiento adversario es una estrecha relación al enfoque en el que el peor de los casos se calcula perturbando los datos directamente como en el caso del modelo robusto clásico de M-estimadores en estadística. DRM es distinto de estas soluciones, perturbando el vector de parámetros en vez de la distribución que gobierna sobre los datos mismos.

El [24, Teorema 3.2] en el escrito de Johannes Roysset y Matthew Norton, proporciona una cota uniforme en R, lo cual implica en particular que:

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

para cualquier vector de parámetro w_m^γ producido por DRM. Por lo tanto existe una fuerte justificación para minimizar R_m^γ : menores valores del riesgo diametral nos asegura mejores garantías sobre el verdadero riesgo. La meta ahora, se transforma en generar mejores métodos para producir w_m^γ con bajo R_m^γ. El problema del sobre ajuste es removido, en nuestro caso para SVM, pues

[1] Se define el módulo de lipschitz de F en $\bar{x}$ como

$$lipF(\bar{x}) = \limsup_{x,x' \to \bar{x}} \frac{|F(x') - F(x)|}{|x' - x|},$$

para nuestros propósitos pensaremos en una vecindad lo bastante grande para que contenga dichos puntos de continuidad.

es poco probable que un vector de parametros (w_m, b_m) con un riesgo diametral bajo es decir un $R_m(w_m, b_m)$ bajo , tenga un riesgo verdadero alto $R(w_m, b_m)$.

4.2.1. Algoritmo

En esta subsección detallaremos dos algoritmos descritos para Diametrical Risk Minimization, los cuales posteriormente se pondrán en práctica con una base de ejemplo para poder analizar las variaciones que realiza este mismo sobre los datos. En lo siguiente, $pry_W(w)$ denotará la proyección de w sobre el conjunto W y $R_{B_t}(w) = \frac{1}{|B_t|}\sum_{z\in B_t} l(w, z)$ denotará el riesgo empírico sobre el lote $B_t \subset S$.

Algorithm 2 Diametrical Risk Minimization SGD

Input: S: conjunto de datos a los cuales se le desea generar SVM-DRM.
γ: valor de la norma de perturbación (radio de perturbación).
num: número de perturbaciones distintas que deseamos.
Output: Parámetros robustos según DRM.

1 Se generan una partición de T elementos tales que S=$\bigcup_{t=0}^{t=T} B_t$
Iniciar $w^0 \in W, r \in \mathbb{N}, t = 0$
Iniciar secuencia de subgrupos $B_t \subset S$
con una tasa de aprendizaje $\lambda_t > 0$

2 Generar r perturbaciones aleatorias
$U = \{u_1, \ldots, u_r | \parallel u \parallel = \gamma\}$
3 **for** $B_t \in S$ **do**
4 **for** $u \in U$ **do**
5 *Calcular* $P = \frac{1}{|B_t|}\sum_{z\in B_t} l(w^t + u, z)$
Seleccionar $u^* \in \text{argmax}_{u\in U} P$
6 *Calcular* $w^{t+1} = pry_W(w^t - \lambda_t \nabla_w R_{B_t}(w^t + u^*))$

El algoritmo 2 de Diametrical Risk Minimizatio SGD (DRM SGD), en cada iteracion t, realiza una actualizacion SGD (Stochastic Gradient Descent) para minimizar la funcion objetivo

$$w \to \max_{u\in U} \frac{1}{|B_t|}\sum_{z\in B_t} l(w + u, z_i).$$

Como primer paso en el algoritmo , se genera una partición de el conjunto de datos S, en T subconjuntos disjuntos entre si, posteriormente se procede a generar r distintas perturbaciones las cuales tienen norma igual γ. Luego se determina el vector u que genere el mayor valor en la función $\frac{1}{|B_t|}\sum_{z\in B_t} l(w^t + u, z)$. Para finalizar se procede a actualizar el parámetro w^t con la proyección.

Dicho algoritmo tiene pasos que pueden mejorarse, por ejemplo, el generar los r distintos vectores aleatorios con norma γ, puede generar un gasto de recursos y tiempo de ejecución alto. En nuestro algoritmo se generaron vectores los cuales posteriormente fueron normalizados y amplificados por γ, mejorando el tiempo de ejecución. A su vez la gran cantidad de iteraciones generadas por los distintos γ obtenidos puede tener un costo computacional alto.

El siguiente algoritmo, el cual llamaremos DRM de memoria corta, incluye un paso que ayuda

a disminuir los recursos computacionales a utilizar pues, se genera un subconjuto acotado de las perturbaciones a estudiar.

Algorithm 3 Diametrical Risk Minimization SGD memoria corta

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

7 Se generan una partición de T elementos tales que S=$\bigcup_{t=0}^{t=T} B_t$
Iniciar $w^0 \in W, r \in \mathbb{N}, t = 0$
Iniciar secuencia de subgrupos $B_t \subset S$
con una tasa de aprendizaje $\lambda_t > 0$

8 Generar r perturbaciones aleatorias
$U = \{u_1, \ldots, u_r | \parallel u \parallel = \gamma\}$
9 **for** $B_t \in S$ **do**
10 **for** $u \in U$ **do**
11 *Calcular* $P = \frac{1}{|B_t|} \sum_{z \in B_t} l(w^t + u, z)$
Seleccionar $u^* \in \text{argmax}_{u \in U} P$
12 *Añadir* u^* *a un conjunto* V_t*, si* $|V_t| > q$ *remover el elemento mas antiguo*
Seleccionar $v^* \in \text{argmax}_{u \in V_t} P$
Calcular $w^{t+1} = pry_W(w^t - \lambda_t \nabla_w R_{B_t}(w^t + v^*))$

La gran diferencia entre los algoritmos 2 y 3, se encuentra en el paso 12 del algoritmo 3. El nuevo conjunto V_t permite mantener 1 o mas de vectores de las iteraciones pasadas. A medida que el algoritmo progresa el conjunto V_t actúa como una cola de de tamaño máximo q. En toda iteración V_t es igual al conjunto V_{t-1} con el elemento mas antiguo reemplazado por u^*.

4.3. Resultados

En esta sección analizaremos los resultados obtenidos por los algoritmos 2 y 3 DRM propuesto en la subsección 4.2.1 al implementarlo junto con la técnica de *Support Vector Machines*, como primer punto analizaremos un conjunto de prueba, con el cual podremos evidenciar los efectos de implementar DRM, posteriormente se mostraran los resultados obtenidos para la base de datos entregada por la identidad bancaria.

4.3.1. Resultados dataset iris

Como primer punto dentro de esta subsección, describiremos el conjunto de datos iris y posteriormente mostraremos los resultados que obtuvimos al implementar el algoritmo 2.

La famosa base de datos Iris, utilizada por primera vez por Sir R.A. Fisher. El conjunto de datos proviene del artículo de Fisher. Repositorio de aprendizaje automático, que tiene dos puntos de datos incorrectos.

Esta es quizás la base de datos más conocida que se puede encontrar en la literatura de reconocimiento de patrones. El artículo de Fisher[11] es un clásico en el campo y se hace referencia con frecuencia a este día. El conjunto de datos contiene 3 clases de 50 instancias cada una, donde cada clase se refiere a un tipo de planta de iris. Una clase es linealmente separable de las otras 2; los últimos no son linealmente separables entre sí.

Descripción de datos iris

Descripción de las variables:

- Numero de sujetos: 150
- Numero de atributos: 4, de los cuales 3 son atributos para predecir y el ultimo 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

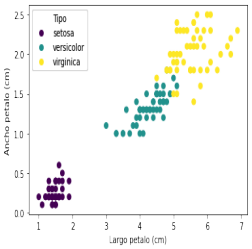
Cuadro 4.1: Tabla descriptiva conjunto Iris

Atributo	Valor
Missing:	0
Distribución de clases:	33 % para cada una de las clases.
Creador	R.A. Fisher
Donor	Michael Marshall

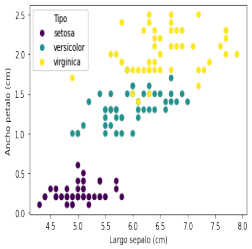
Cuadro 4.2: Descripción de los datos iris

De los datos anteriormente descritos fue posible generar los siguientes gráficos:

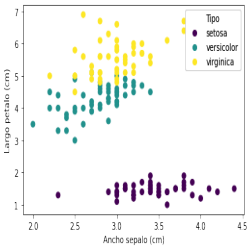
De lo cual es posible visualizar que existen 2 grupos que son separables ajustando una recta entre ellas, lo cual nos entregaría 2 categorías dentro del universo de datos, uno que pertenezca al grupo sétosa y otro al grupo de no sétosa (en este caso seria las plantas de categoría virginica y versicolor). Así, es posible prever que nuestro algoritmo de SVM podrá separarlos de una forma eficiente. Dado que el ancho y el largo del pétalo tienen una mayor correlación sobre las categorías, se seleccionarán



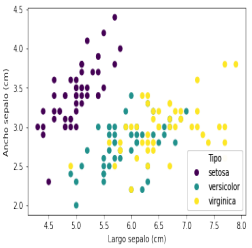
(a) Ancho del pétalo vs Largo del pétalo.



(b) Ancho del pétalo vs Largo del sépalo.



(c) Largo del pétalo vs Ancho del sépalo.



(d) Largo del sépalo vs Ancho del sépalo.

Figura 4.1: Relación observable entre las variables del conjunto de datos.

estas para poder estudiarlas con nuestro algoritmo de SVM y posteriormente con DRM en conjunto con SVM.

Los resultados para nuestro algoritmo de SVM con nuestro conjunto de datos nos entregó la siguiente figura:

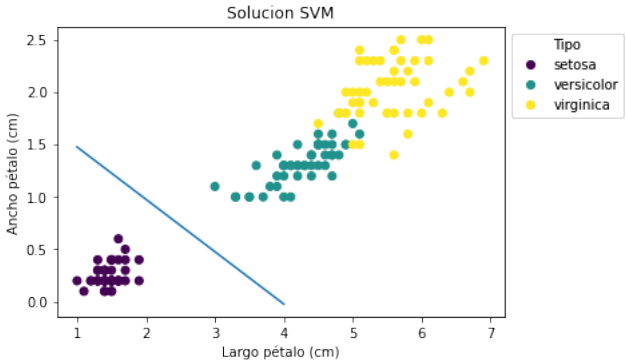
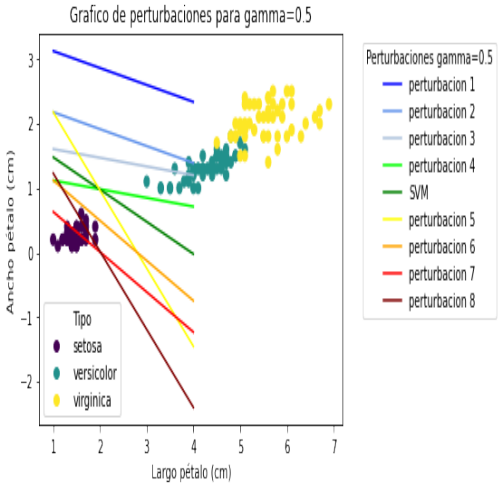
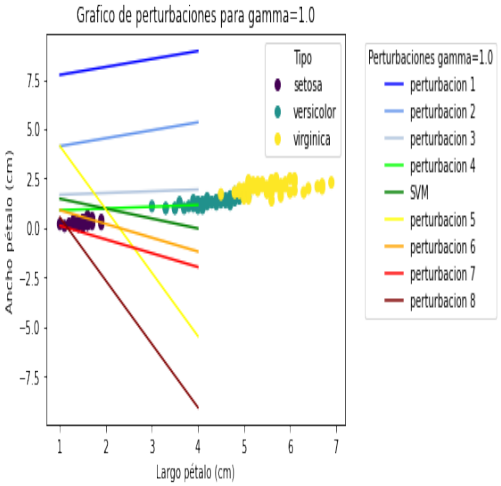


Figura 4.2: Solucion Support Vector Machines

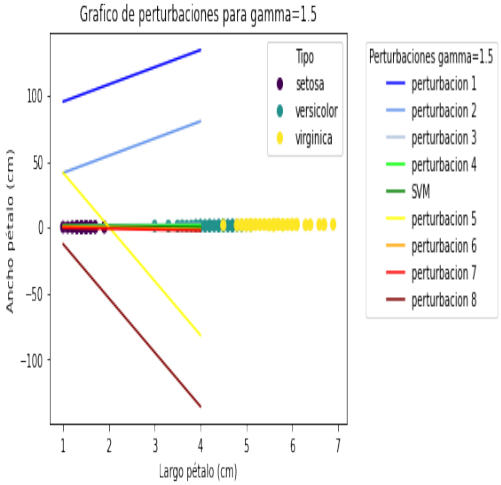
Lo cual nos hace ver que los resultados son perfectos para nuestro modelo de Support Vector Machines. Ahora procederemos a generar con el algoritmo de DRM distintas perturbaciones a nuestros parámetros e intentaremos visualizar las diferencias entre las respuestas y las soluciones. Como primer acercamiento mostraremos las distintas perturbaciones que pueden generarse con DRM a nuestra solución si es que le entregamos un valor de diámetro de riesgo que varía según la siguiente lista $\gamma = [0{,}5 - 1{,}0 - 1{,}5 - 2{,}0]$, para dichos diámetros obtenemos las siguientes figuras:



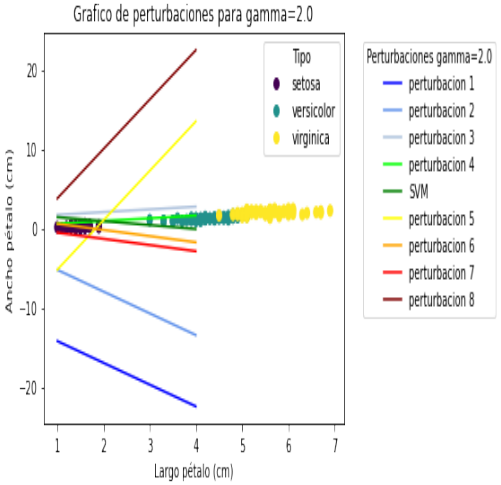
(a) Resultado de algoritmo DRM-SVM para un $\gamma = 0{,}5$



(b) Resultado de algoritmo DRM-SVM para un $\gamma = 1{,}0$



(c) Resultado de algoritmo DRM-SVM para un $\gamma = 1{,}5$

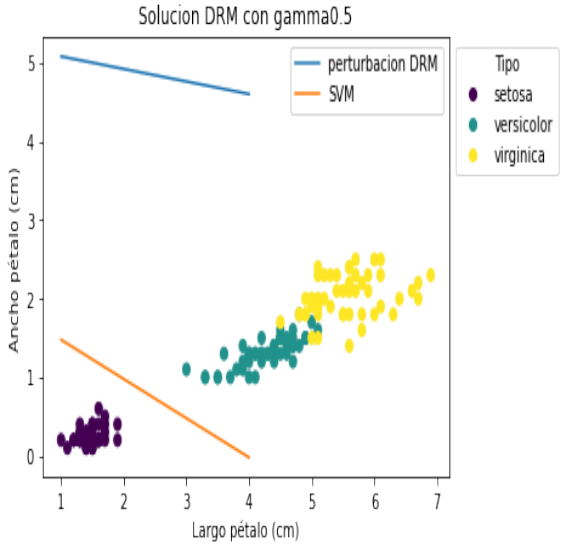


(d) Resultado de algoritmo DRM-SVM para un $\gamma = 2{,}0$

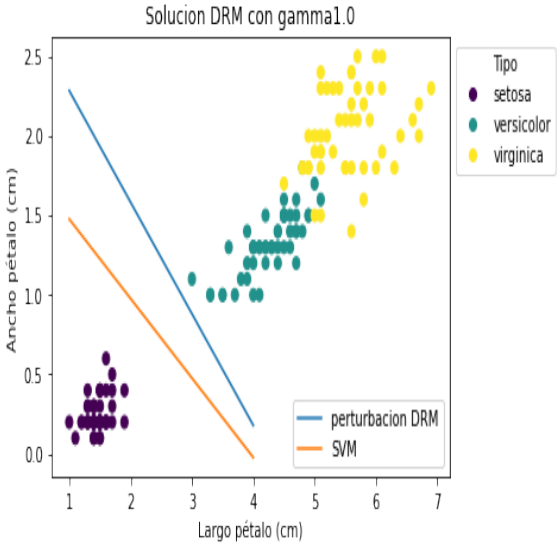
Figura 4.3: Posibles resultados de SVM-DRM para el conjunto de datos iris.

de la figura 4.3 es posible analizar que el algoritmo de DRM hace que cada una de las perturbaciones puedan ser una buena o una mala clasificación de nuestro conjunto de datos, dentro de dichas perturbaciones es posible observar que existen rectas que generan una clasificación tan perfecta como la que se describió en la figura 4.2. También es posible ver que para este problema, existen rangos en el cual el algoritmo de DRM degenera las soluciones por su alto valor, en este caso en particular fue con un $\gamma = 1{,}5$ el factor que nos indicaría el problema, a su vez es posible ver que en la figura 4.3d las soluciones empiezan a variar en el sentido de la recta (cambian la pendiente del margen de una negativa a una pendiente positiva), lo cual en este caso fue igual de favorable, pues pueden generar soluciones que identifican bien a los prospectos. Ya que analizamos las rectas resultantes para cada uno de estos valores de diámetro de riesgo, ahora resolveremos el problema con los mismos valores de diámetro de riesgo utilizados para generar las figuras anteriores

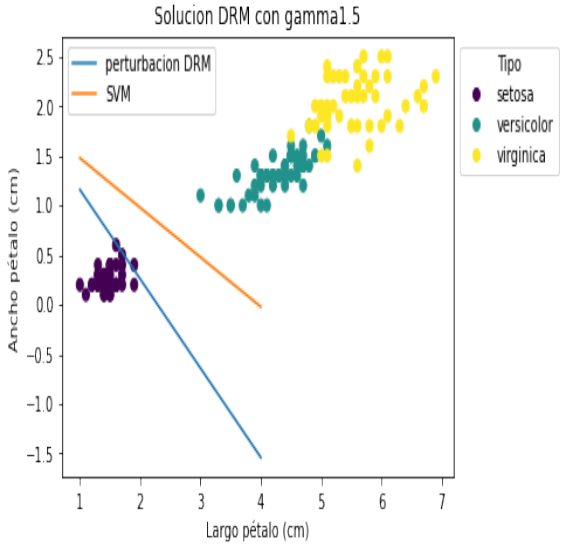
y analizaremos los resultados en comparación a la solución original entregada por Support Vector Machines. Las soluciones son las siguientes:



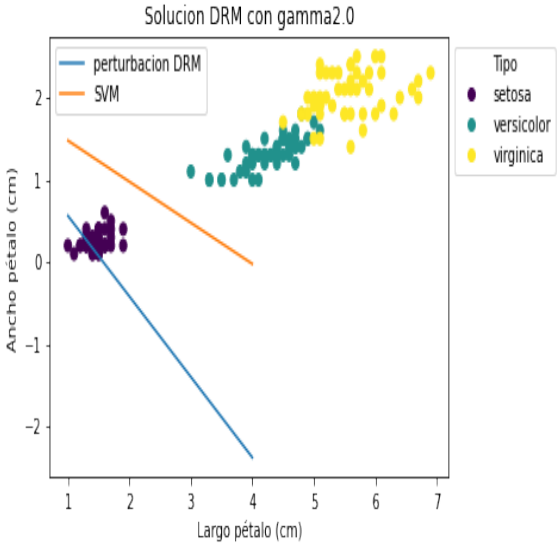
(a) Resultado de algoritmo DRM-SVM para un $\gamma = 0{,}5$



(b) Resultado de algoritmo DRM-SVM para un $\gamma = 1{,}0$



(c) Resultado de algoritmo DRM-SVM para un $\gamma = 1{,}5$



(d) Resultado de algoritmo DRM-SVM para un $\gamma = 2{,}0$

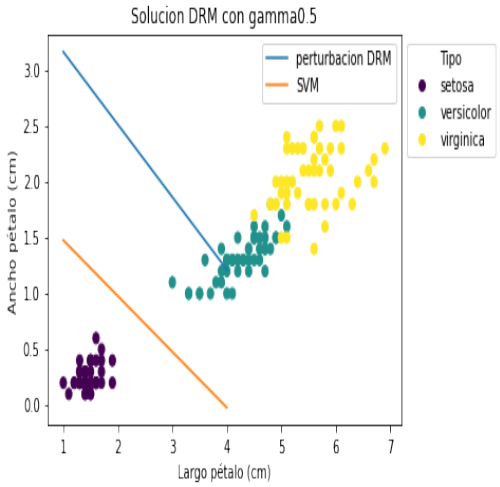
Figura 4.4: Resultados de SVM-DRM para el conjunto de datos iris.

A partir de estos resultados es posible ver que el algoritmo convergió a una buena recta clasificatoria solo en un caso, pues solo en la figura 4.4b es posible visualizar los diferentes grupos mostrando un límite separador entre ellos. La solución entregada por las figura 4.4a, nos indica que dentro de los primeros valores de perturbación la solución converge a una mala recta clasificatoria, pues no clasifica ambos grupos, posteriormente con las perturbaciones con $\gamma > 1{,}0$ es posible percibir que la

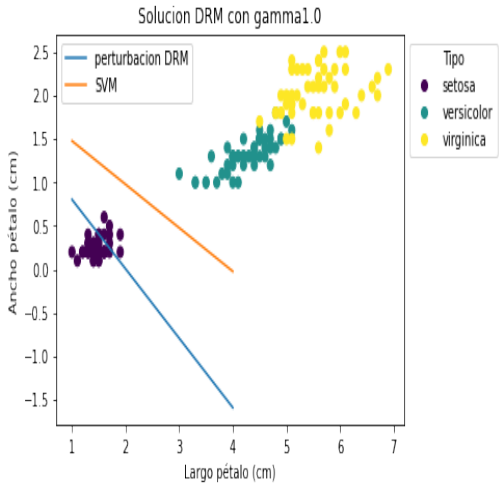
recta tiene una tendencia a generar falsos positivos como clasificador. La recta tiende a tener una inclinación por el grupo setosa, lo cual no deja de ser una solución aceptable para nuestro problema de clasificación.

Ahora analizaremos las soluciones obtenidas para el algoritmo 3, con una variación en la cardinalidad del grupo V_t, los valores variaran entre $[2, 5, 8]$. Los resultados de estas simulaciones son:

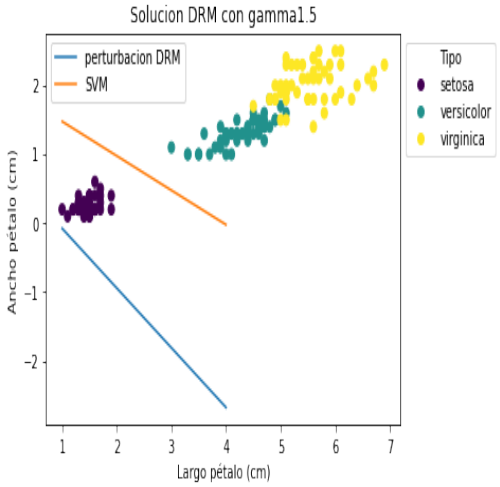
Para el grupo con cardinalidad $|V_t| = 2$, los resultados se encuentran en la figura 4.5:



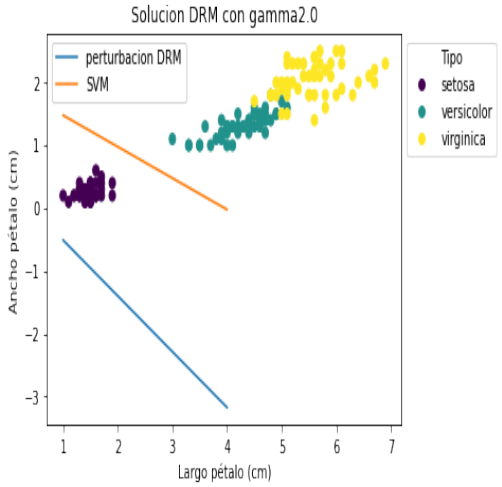
(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 50\,\%$



(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 100\,\%$



(c) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 150\,\%$



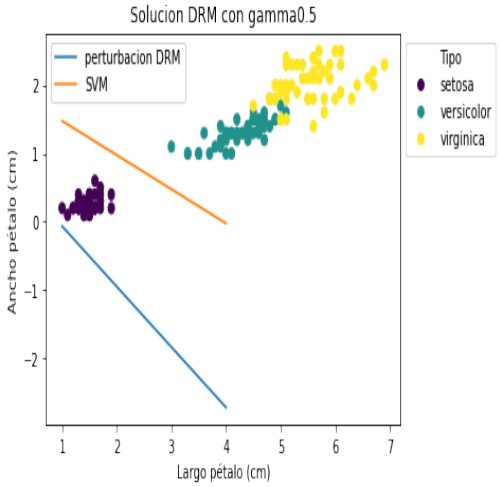
(d) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 200\,\%$

Figura 4.5: Posibles resultados de SVM-DRM memoria corta para el conjunto de datos iris con cardinalidad en el conjunto de cola $|V_t| = 2$.

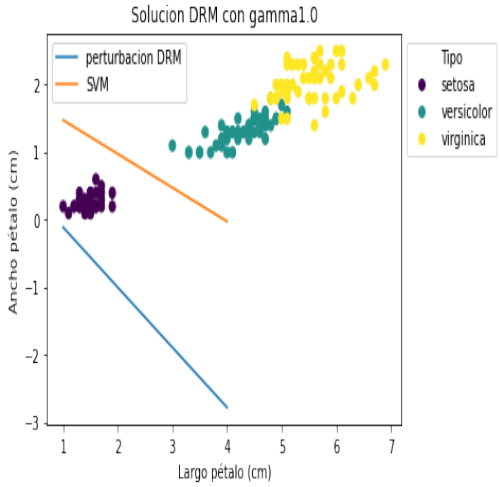
A partir de estos resultados en la figura 4.5 es posible ver que las perturbaciones entre $50\,\%$ y $100\,\%$, generan una clasificación con tendencias de falsos negativos o falsos positivods. Por otra parte las alteraciones mayores de $100\,\%$ tienden a generar un modelo no clasificatorio. El grupo

de cola con cardinalidad de $|V_t| = 2$ nos hace pensar que la memoria influirá en los resultados y como evoluciona el modelo, a partir de esto notamos que las alteraciones grandes (mayores a 100 %) tienden a caer en soluciones similares.

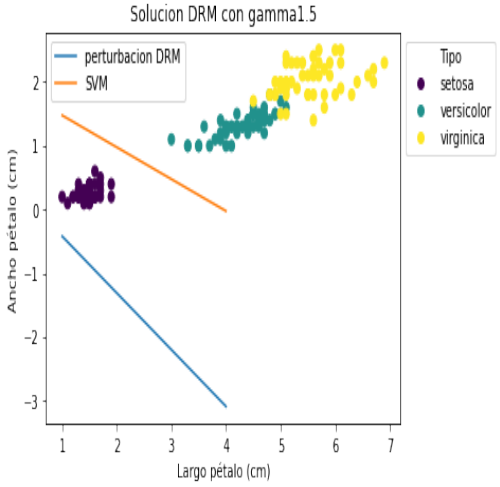
Para el grupo con cardinalidad $|V_t| = 5$, se obtuvieron los resultados mostrados por la figura 4.6.



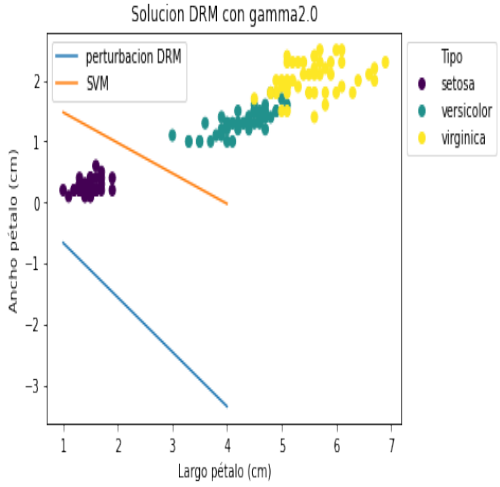
(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 50\,\%$



(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 100\,\%$



(c) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 150\,\%$

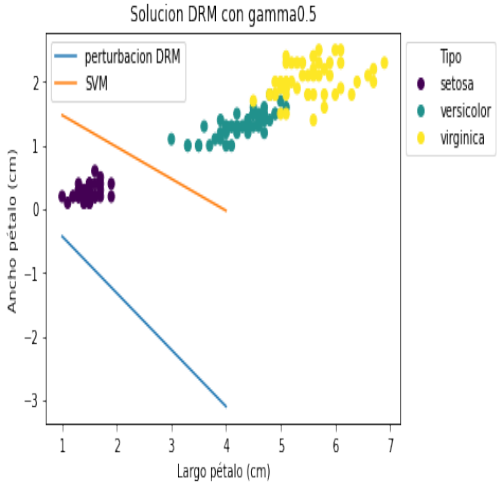


(d) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 200\,\%$

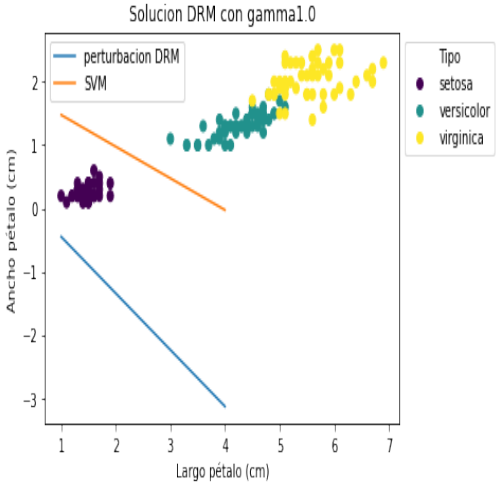
Figura 4.6: Posibles resultados de SVM-DRM memoria corta para el conjunto de datos iris con cardinalidad en el conjunto de cola $|V_t| = 5$.

A partir de los resultados mostrados por la figura 4.6 es posible ver que todas las perturbaciones, no generan una clasificación. Todos los resultados tienden a generar los mismos resultados dentro de la clasificación, mostrando una mínima variación en la recta delimitadora de conjuntos.

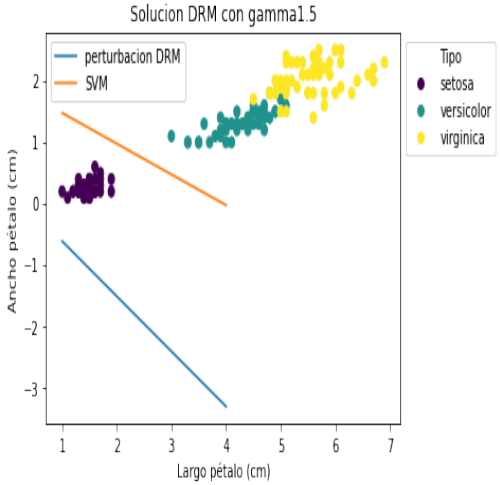
Los resultados para el conjunto con cardinalidad $|V_t| = 8$ se encuentran en la figura 4.7.



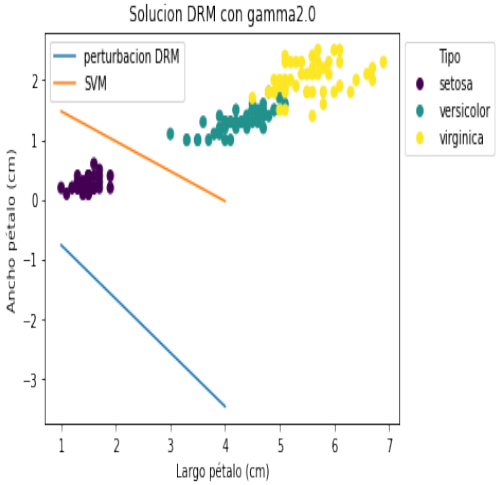
(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 50\,\%$



(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 100\,\%$



(c) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 150\,\%$



(d) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 200\,\%$

Figura 4.7: Posibles resultados de SVM-DRM memoria corta para el conjunto de datos iris con cardinalidad en el conjunto de cola $|V_t| = 8$.

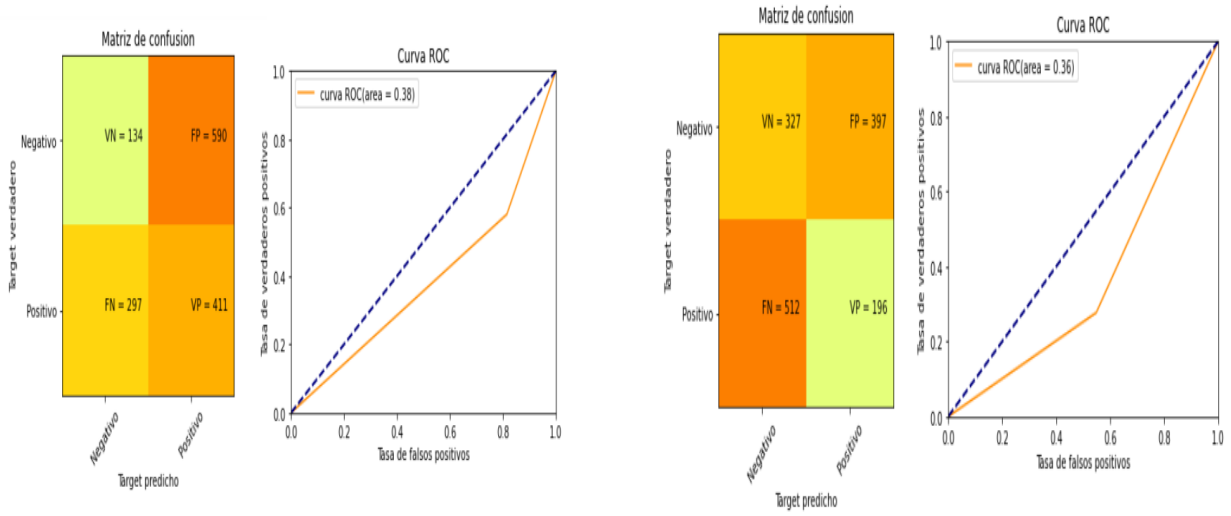
A partir de los resultados mostrados por la figura 4.7 es posible ver que todas las perturbaciones, no generan una clasificación. Todos los resultados tienden a generar los mismos resultados dentro de la clasificación, mostrando una mínima variación en la recta delimitadora de conjuntos.

A partir de los resultados obtenidos para ambos algoritmos es posible identificar que, el algoritmo 2 tuvo un modelo que si genero una buena clasificación delimitando ambos conjuntos. Por otro lado, en ambos algoritmos cuando se aumentaba el la perturbación, la recta separadora tiende a generar falsos negativos. También, si identificamos el tiempo de ejecución para esta base de datos , es posible ver que el algoritmo 3 tiene un mayor tiempo de ejecución, tardando en promedio 3 segundos.

4.3.2. Resultados datos bancarios

En esta sub-sección, estudiaremos los resultados entregados por DRM para los datos obtenidos de la institución bancaria y generaremos un análisis con base en 3 aristas, una de ellas es la matriz de confusión, otro el valor *Roc-Score* y por ultimo la cantidad de positivos generados por el modelo. Para el análisis, 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. Es importante recalcar que para cada uno de los set de datos (SD) se generó un modelo especializado con la técnica de DRM, el cual se adaptó según una escala de valores para la perturbación γ, el cual se adaptara a un porcentaje de los parámetros obtenidos como solución inicial del SVM. Posterior a el análisis de estos resultados, intentaremos comparar el algoritmo 2 con su contra parte de memoria corta el cual esta descrito en el algoritmo 3 y estudiar cual de estos tiene un mejor resultado, comparando los tiempos de ejecución y los resultados.

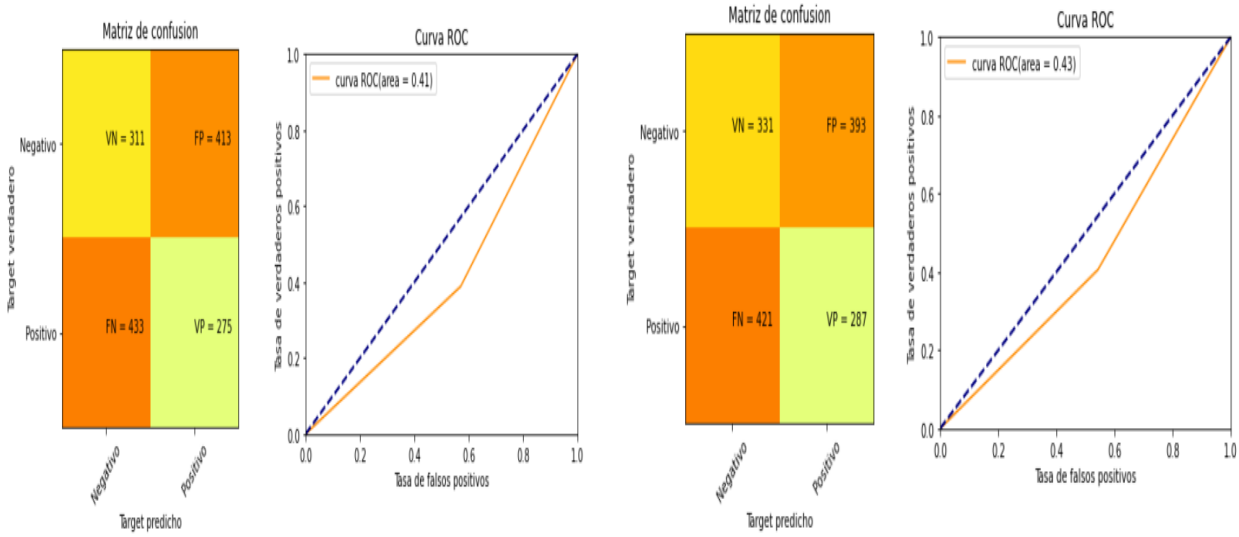
Para la muestra con 4400 datos para generar el proceso de DRM, con una escala de γ variando entre 10 % y 100 % de perturbación en los parámetros se obtuvieron las figuras 4.8, 4.9, 4.10, 4.11 y 4.12.



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 10\,\%$

(b) Resultado de algoritmo DRM-SVM para un $\gamma = 20\,\%$

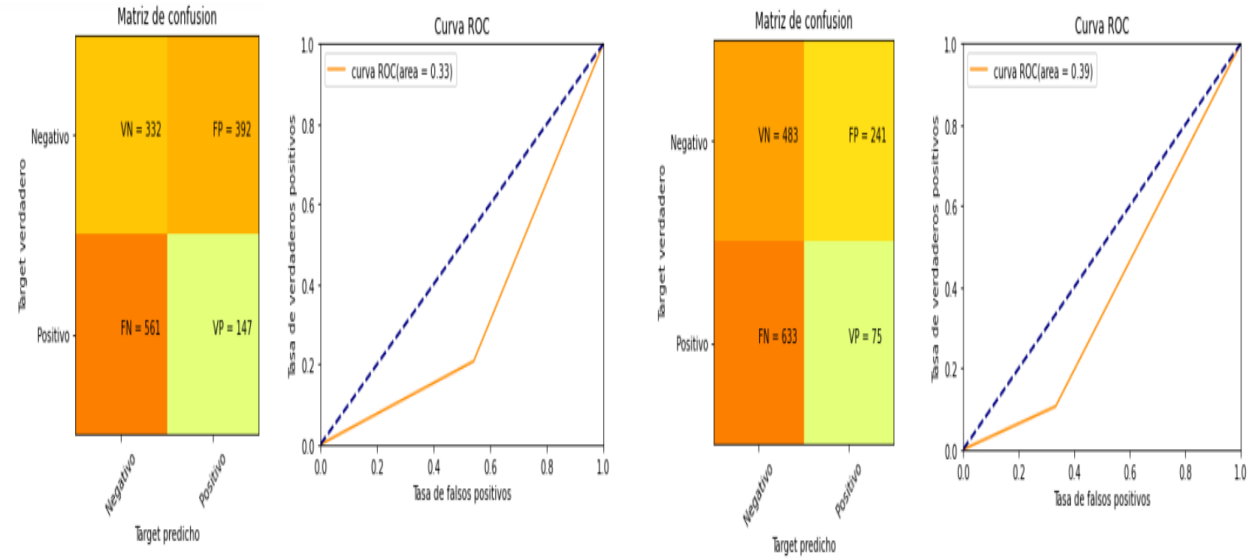
Figura 4.8: Resultados de SVM-DRM para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 30\,\%$

(b) Resultado de algoritmo DRM-SVM para un $\gamma = 40\,\%$

Figura 4.9: Resultados de SVM-DRM para el conjunto de datos bancarios.

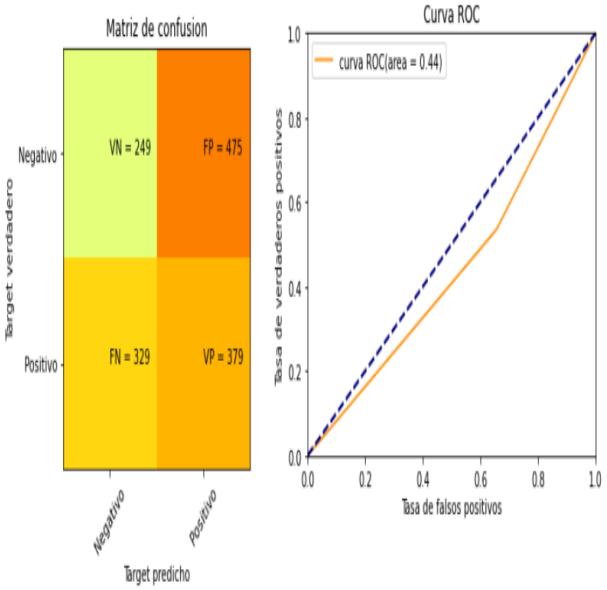


(a) Resultado de algoritmo DRM-SVM para un $\gamma = 50\,\%$

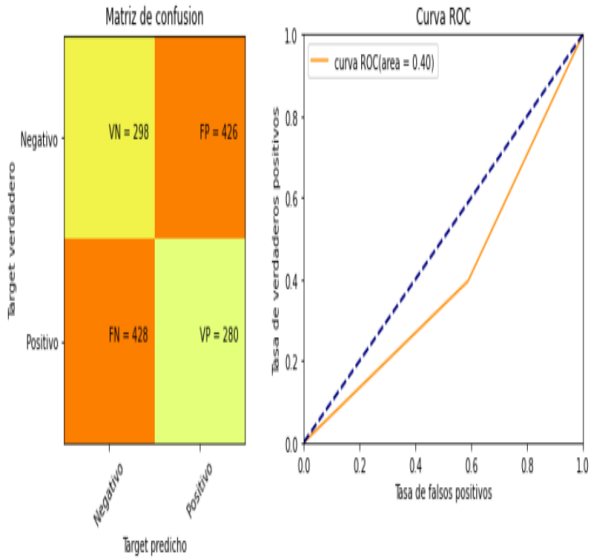
(b) Resultado de algoritmo DRM-SVM para un $\gamma = 60\,\%$

Figura 4.10: Resultados de SVM-DRM para el conjunto de datos bancarios.

De los resultados en las figuras 4.8, 4.9, 4.10, 4.11 y 4.12 es posible ver que, en su totalidad que, las matrices de confusión y los gráficos ROC nos muestran un claro predominio de los sectores falsos,

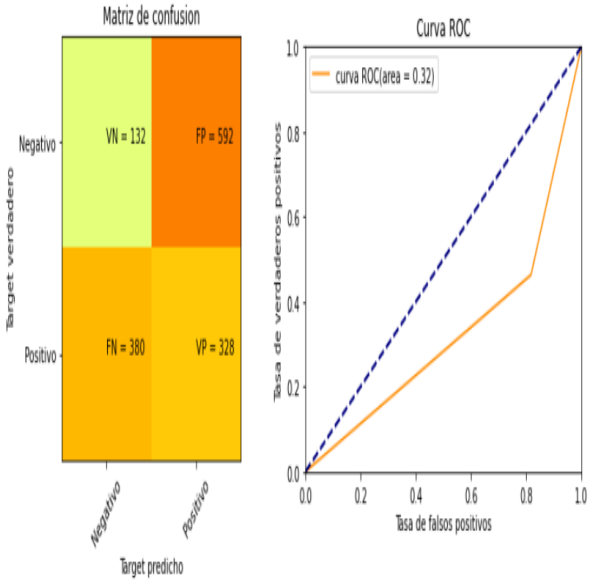


(a) Resultado de algoritmo DRM-SVM para un $\gamma = 70\,\%$

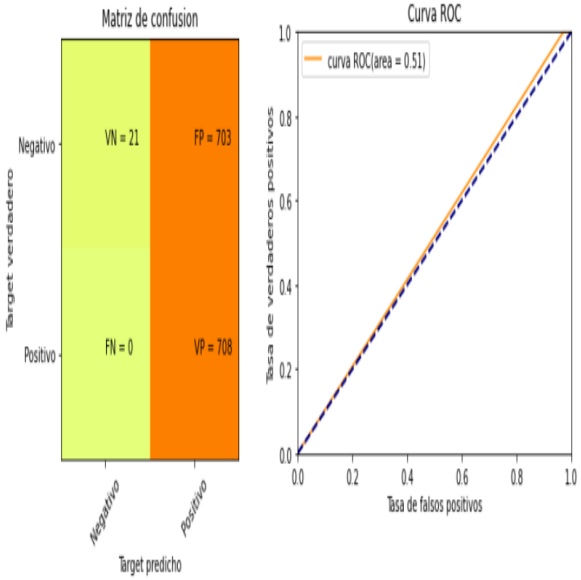


(b) Resultado de algoritmo DRM-SVM para un $\gamma = 80\,\%$

Figura 4.11: Resultados de SVM-DRM para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 90\,\%$



(b) Resultado de algoritmo DRM-SVM para un $\gamma = 100\,\%$

Figura 4.12: Resultados de SVM-DRM para el conjunto de datos bancarios.

en vez de los verdaderos tal como se esperaría de un buen clasificador. Junto a esto es necesario mencionar que el área bajo la curva o *ROC-Score* es siempre menor a un 0,6, lo cual nos indicaría

que estos modelos como clasificadores, son deficientes, pues no cubren de una manera correcta a los casos verdaderos.

En la figura 4.8 es posible ver que la transición de 10 % a 20 % hace que el modelo invierta la orientación de la matriz de confusión, manteniendo el orden del *ROC-Score*.

Siguiendo con nuestro análisis, en la figura 4.9 es posible ver que existe una tendencia similar en las matrices de confusión pertenecientes a 30 % y 40 % respectivamente manteniendo la predominancia de los resultados falsos por sobre los resultados verdaderos. Es importante mencionar, que los órdenes de *ROC-Score* tienden al aumento, lo cual nos indicaría que muestran un mejor discernimiento de los casos verdaderos.

Continuando con los resultados obtenidos para esta muestra, en la figura 4.10, es posible percibir que existe una nueva tendencia por los sectores falsos, lo cual empieza a demarcar un cambio en la orientación de las matrices, tienden a generar más falsos negativos y a mejorar el reconocimiento de los patrones negativos. Junto a esto, también es necesario mencionar que el *Roc-Score* tiende a disminuir, lo cual nuevamente nos muestra una deficiencia en la clasificación.

Prosiguiendo con el análisis de las figuras, en la figura 4.11, es posible percibir que existe una mejora en el gráfico de $\gamma = 70\,\%$ en el discernimiento de los prospectos verdaderos y cambia la tendencia hacia el sector positivo falso, en cambio en la siguiente figura nuevamente se tiene una tendencia hacia el sector negativo, equilibrando las entradas de los sectores falsos. Es importante mencionar que existe un aumento del *ROC-Score* en ambas figuras, lo cual resume lo que vimos en la matriz de confusión.

Finalmente en la figura 4.12 es posible ver que en la imagen referente a la perturbación de $\gamma = 90\,\%$ nuevamente el modelo retorna a el sector de falsos positivos y aumenta la clasificación de los prospectos positivos, disminuyendo el Roc-Score. Luego en la última imagen es posible ver que existe una clara tendencia hacia los positivos en general (falsos y verdaderos), aumentando el Roc-Score hasta el mejor resultado de esta base de datos, esto último no nos indica que es el mejor clasificador pues, la matriz nos indica que solo tiende a reconocer los resultados como verdaderos en su mayoría.

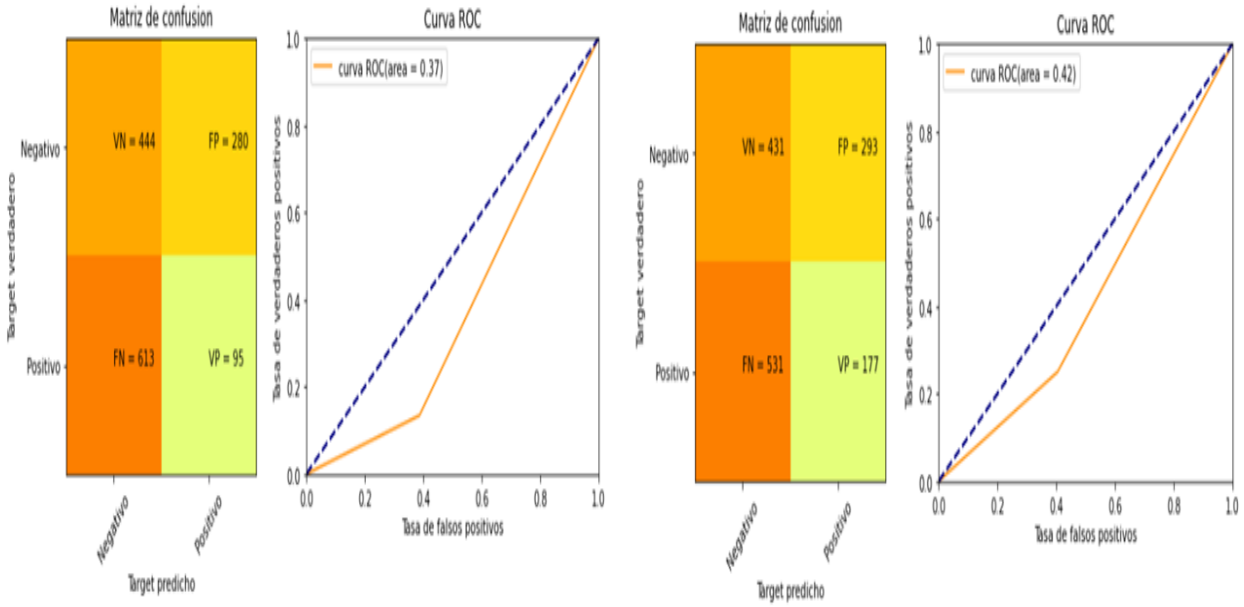
A manera de resumen dejaremos los resultados obtenidos para cada uno de los porcentajes en el cuadro 4.3, con los valores de ROC-Score, cantidad de positivos y el valor de la función objetivo[2].

Para la muestra con 5000 de datos para generar el proceso de DRM, con una escala de γ variando entre 10 % y 100 % de perturbación en los parámetros se obtuvieron los resultados presentados por las figuras 4.13, 4.14, 4.15, 4.16 y 4.17. A partir de estos resultados es posible ver que, en su mayoría, las matrices de confusión y los gráficos ROC nos muestran un claro predominio de los sectores falsos, en vez de los verdaderos tal como se esperaría de un buen clasificador. Con esto es necesario mencionar que el área bajo la curva o *ROC-Score* es siempre menor a un 0,6 , lo cual nos indicaría que estos modelos como clasificadores, son deficientes, pues no cubren de una manera correcta a los casos verdaderos.

[2]Los resultados de la función objetivo fueron normalizados para una mejor comprensión

Gama γ	ROC-Score	N° 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 4.3: 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.



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 10\,\%$

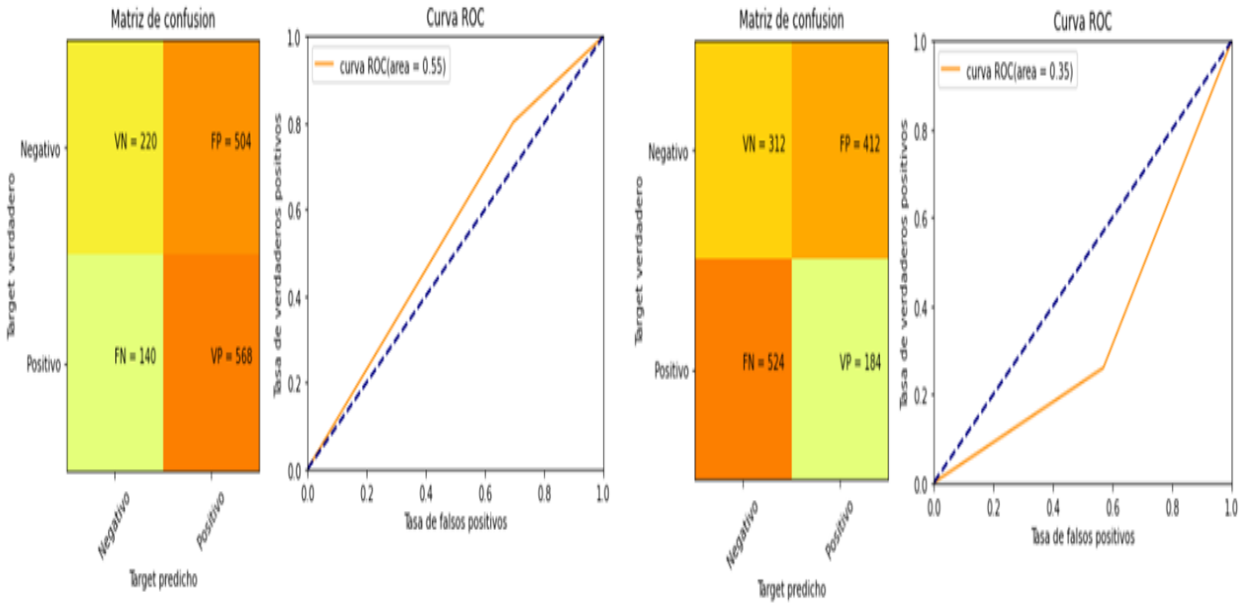
(b) Resultado de algoritmo DRM-SVM para un $\gamma = 20\,\%$

Figura 4.13: Resultados de SVM-DRM para el conjunto de datos bancarios.

Analizando las figuras 4.13, 4.14, 4.15, 4.16 y 4.17, se tiene que:

En la figura 4.13, es posible ver que existe una tendencia por ambos modelos a identificar más casos negativos que casos positivos, el caso de $\gamma = 10\,\%$ tiende a no reconocer los patrones positivos, por otro lado el modelo con perturbación $\gamma = 20\,\%$ tiende a reconocer más el sector antes aludido, pero aun manteniendo la tendencia por los casos falsos negativos. Por otra parte el valor del área bajo la curva o *Roc-Score* es creciente, lo cual reafirma lo mostrado por la matriz de confusión.

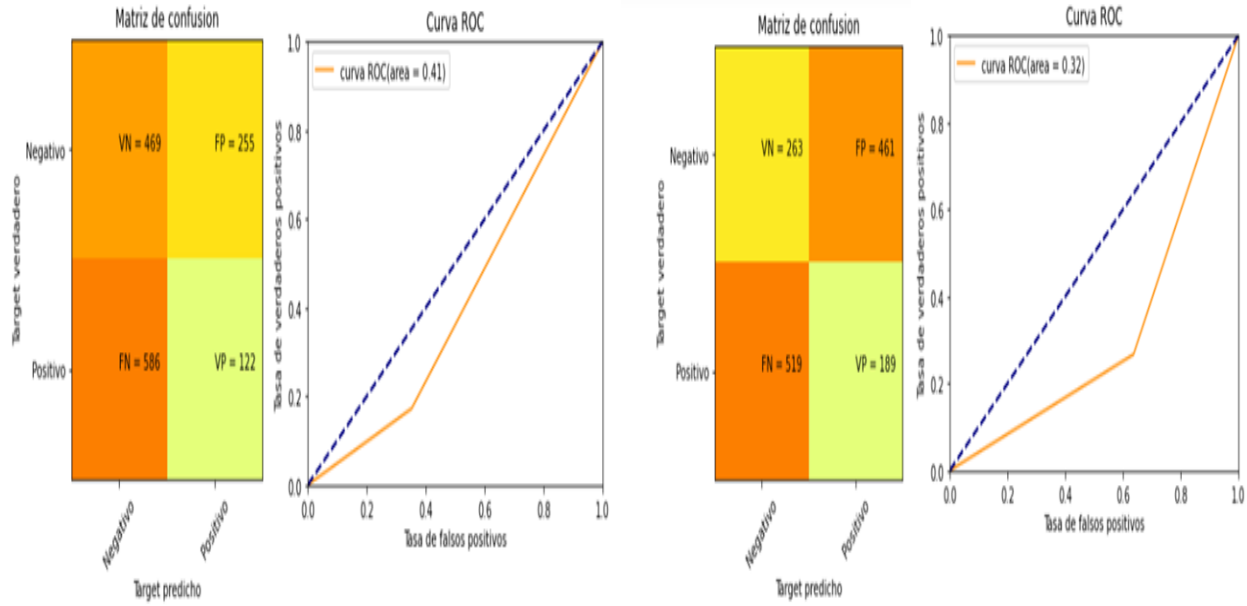
En la figura 4.14, es posible ver que existe un claro cambio en el discernimiento del modelo, pues



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 30\,\%$

(b) Resultado de algoritmo DRM-SVM para un $\gamma = 40\,\%$

Figura 4.14: Resultados de SVM-DRM para el conjunto de datos bancarios.

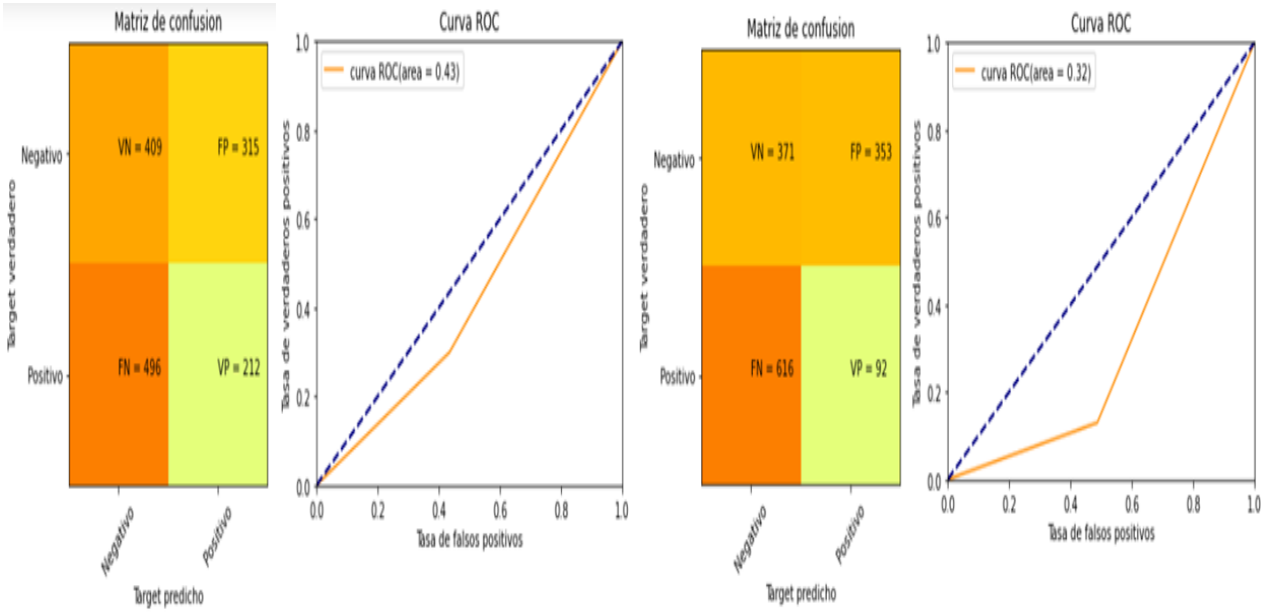


(a) Resultado de algoritmo DRM-SVM para un $\gamma = 50\,\%$

(b) Resultado de algoritmo DRM-SVM para un $\gamma = 60\,\%$

Figura 4.15: Resultados de SVM-DRM para el conjunto de datos bancarios.

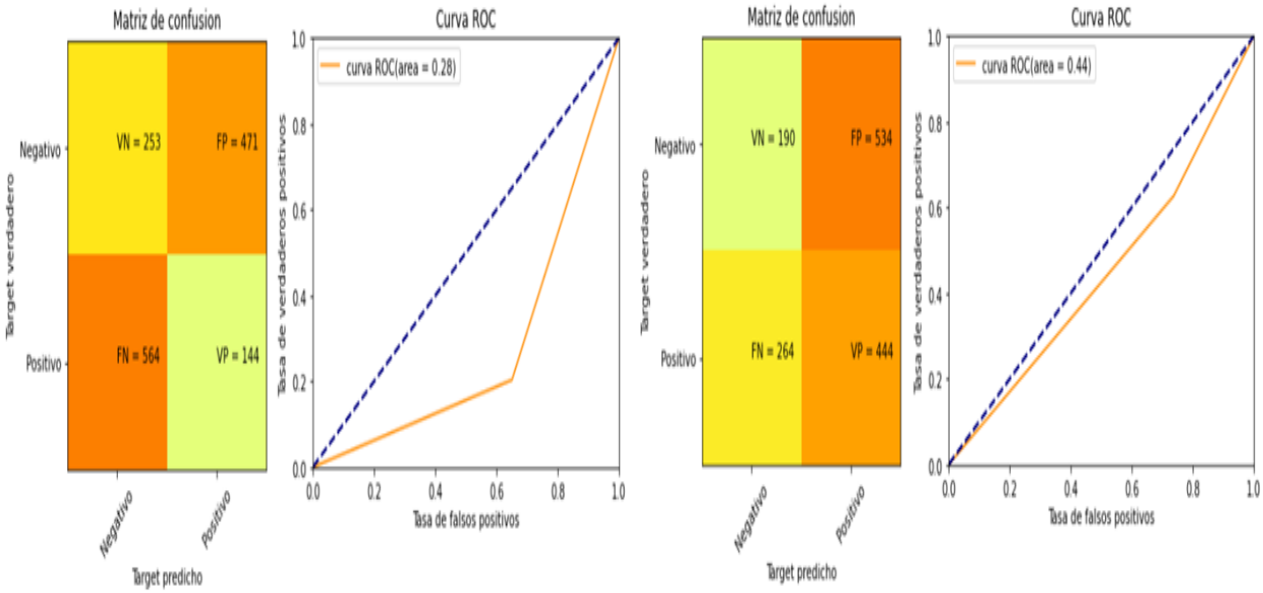
en la matriz de confusión generada para el modelo con perturbación $\gamma = 30\,\%$, denota un aumento en el reconocimiento verdadero (tanto negativo como positivo) y muestra que los números para



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 70\,\%$

(b) Resultado de algoritmo DRM-SVM para un $\gamma = 80\,\%$

Figura 4.16: Resultados de SVM-DRM para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM para un $\gamma = 90\,\%$

(b) Resultado de algoritmo DRM-SVM para un $\gamma = 100\,\%$

Figura 4.17: Resultados de SVM-DRM para el conjunto de datos bancarios.

estas casillas es mayor que los de las casillas falsas, con esto también es posible ver el aumento del valor del área bajo la curva. Por otro lado, la matriz para el modelo con perturbación $\gamma = 40\,\%$

nuevamente nos describe un modelo que tiene un bajo discernimiento y una clara tendencia por generar más falsos (positivos y negativos) que una clasificación correcta, con una tendencia por los falsos negativos.

En la figura 4.15, en la primera figura, la que hace representación a el modelo con la perturbación de $\gamma = 50\,\%$, dicho modelo tiende a generar una mayor cantidad de verdaderos (positivos como negativos) con una tendencia por generar más negativos en general (falsos y verdaderos), aun así la figura de la curva ROC, es posible ver que tiende a aumentar el área con respecto al valor anterior. Para el modelo con perturbación $\gamma = 60\,\%$, se tiene que existe un mal reconocimiento de los patrones positivos y negativos. Tiende a generar más falsos (positivos y negativos), esto se ve reflejado con la curva ROC, la cual nos muestra una disminución del área bajo la curva y una tendencia hacia los falsos negativos.

Prosiguiendo con el análisis de las figuras obtenidas, en la figura 4.16, para el caso en que la perturbación $\gamma = 70\,\%$, es posible ver que existe una mejora en el reconocimiento de las etiquetas verdaderas, pero aun así mantiene la tendencia del modelo predecesor sobre el lado negativo, con ello también tenemos un aumento del valor *ROC-Score*, lo cual nos indica una mejora en la clasificación tal como lo habíamos mencionado anteriormente. En el siguiente modelo, es posible ver que se mantiene la tendencia de generar más etiquetas falsas que las verdaderas, pero este modelo tiene la tendencia a disminuir el reconocimiento verdadero de la muestra, lo cual se ve reflejado en el valor de área bajo la curva.

Finalmente, en la figura 4.17 es posible ver que en el modelo que tiene una perturbación de $\gamma = 90\,\%$, tiende a generar más falsos que verdaderos en la clasificación, tratando de equilibrar ambas entradas, aun así mejora el reconocimiento positivo verdadero. En el modelo con $100\,\%$ de perturbación, es posible ver que tiende a cambiar la orientación que traían los modelos, pues tiende a generar más positivos que negativos (tanto falsos como verdaderos), manteniendo las cifras verdaderas menores que las anteriores, este aumento en los verdaderos hace que el área de la curva aumente.

A manera de resumen dejaremos los resultados obtenidos para cada uno de los porcentajes en el cuadro 4.4, el cual incluye los valores de ROC-Score, cantidad de positivos y el valor de la función objetivo[3].

A partir de los cuadros 4.3, 4.4, es posible buscar alguna tendencia en los datos. Para ello generaremos 3 gráficos distintos, todos ellos con respecto a el valor gama asociado al modelo, de esto intentaremos entender que es lo que se gana con el modelo de DRM y que es lo que se pierde a lo largo de las distintas restricciones.

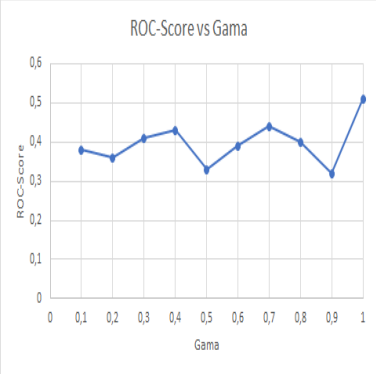
Los gráficos que se obtuvieron en torno a el cuadro 4.3 y a el cuadro 4.4, se pueden apreciar en las figuras 4.18 y 4.19.

Si comparamos los cuadros 4.3 y 4.4, es posible ver el máximo valor de ROC-Score se encontró en el modelo DRM-SVM de 5000 eventos, el cual conllevo a la mejor puntuación de positivos entre todos los resultantes. Este resultado fue el único entre estas veinte pruebas que genero una curva cóncava en el gráfico ROC. Por otra parte, la gran cantidad de positivos generados, provienen de la entrada para los positivos verdaderos, los cuales superaron en una mínima cantidad a los falsos positivos. También es importante mencionar que el promedio de ROC-Score y positivos en el modelo

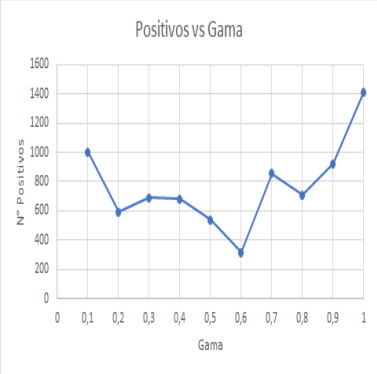
[3]Los resultados de la función objetivo fueron normalizados para una mejor comprensión

Gama γ	ROC-Score	N° 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	0.0
100 %	0.44	978	0.0

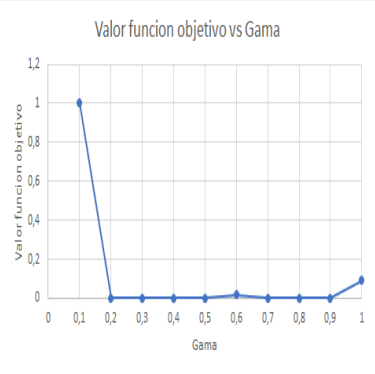
Cuadro 4.4: Tabla resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.



(a) Resumen ROC-Score vs Gama



(b) Resumen Numero de positivos vs Gama



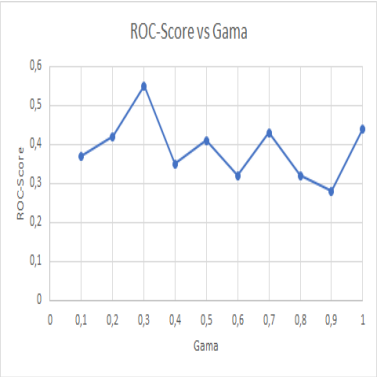
(c) Resumen valor de funcion objetivo vs Gama

Figura 4.18: Resultados de las relaciones encontradas en la tabla resumen 4.3

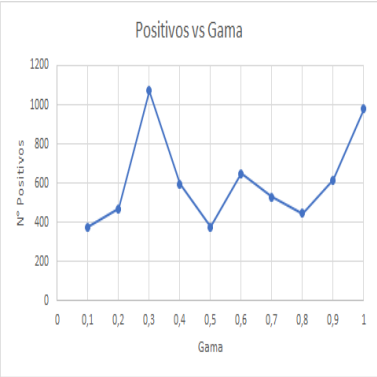
DRM-SVM de 4400 eventos es mayor que en el modelo DRM-SVM de 5000 eventos.

Ahora analizaremos el comportamiento del algoritmo 3, con respecto a la capacidad del conjunto V_t la cual variara entre $[2, 5, 8]$. Todas estas pruebas serán realizadas para un set de datos de 5000 prospectos, así podremos comparar los resultados obtenidos por ambos algoritmos, con respecto a las medidas ROC-Score, numero de positivos generados y el rendimiento ocupado por el computador.

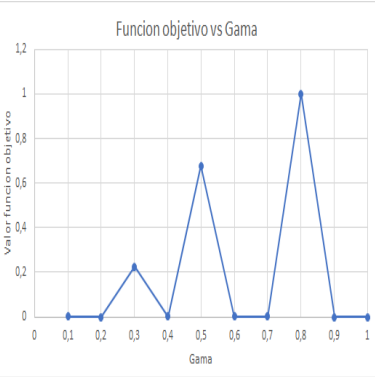
Los resultados para un set de datos 5000 y una capacidad de $V_t = 2$ son los que se muestran en las figuras 4.20, 4.21, 4.22, 4.23 y 4.24. De estos resultados, es posible ver que, nuevamente en la mayoría de los casos las matrices de confusión y los gráficos ROC nos muestran un claro predominio de los sectores falsos (exceptuando por algunos casos), en vez de los verdaderos tal como se esperaría de un buen clasificador. Con esto es necesario mencionar que el área bajo la curva o *ROC-Score* es siempre menor a un 0,6, lo cual nos indicaría que estos modelos como clasificadores, son deficientes, pues no cubren de una manera correcta a los casos verdaderos, pero pueden utilizarse para generar un modelo que nos ayude a visualizar mejor los falsos verdaderos.



(a) Resumen ROC-Score vs Gama

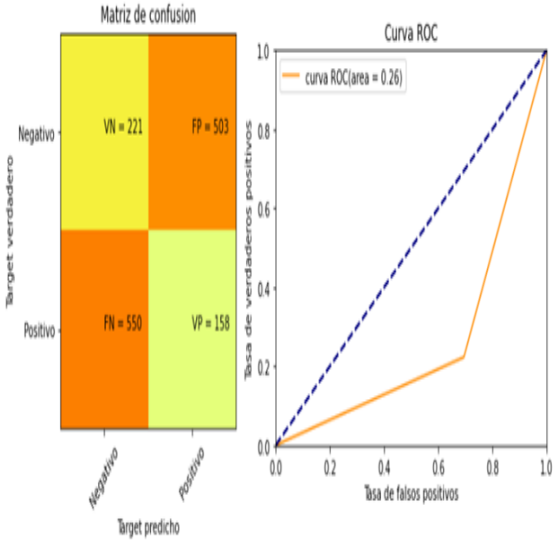


(b) Resumen Numero de positivos vs Gama

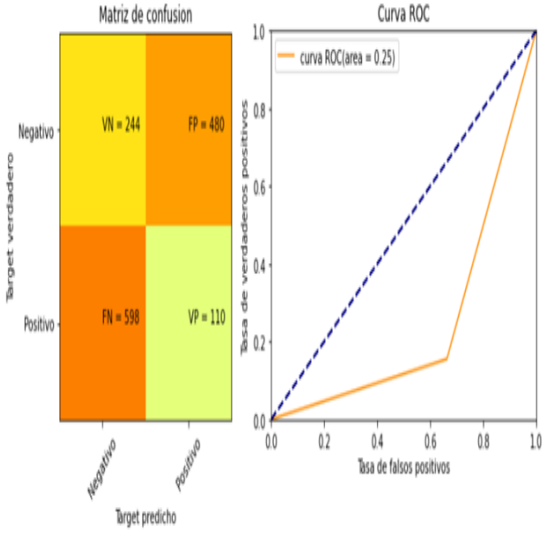


(c) Resumen valor de función objetivo vs Gama

Figura 4.19: Resultados de las relaciones encontradas en la tabla resumen 4.4



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 10\,\%$

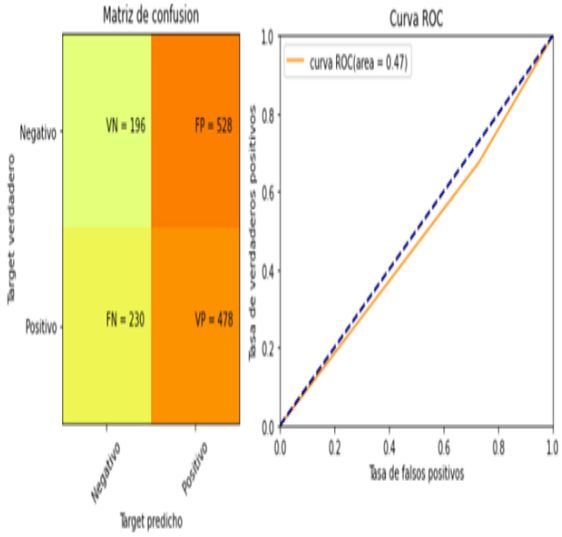


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 20\,\%$

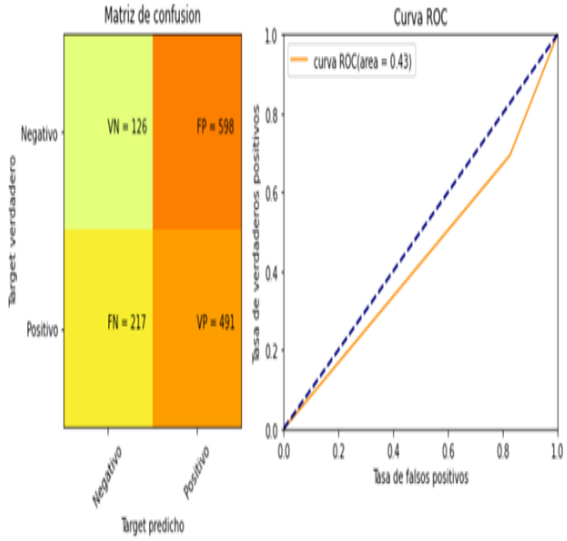
Figura 4.20: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

Ahora si analizamos las figuras 4.20, 4.21, 4.22, 4.23 y 4.24, obtenemos que:

En la figura 4.20, los gráficos pertenecientes a una perturbación de $\gamma = 10\,\%$ es posible observar que el modelo tiene una tendencia a generar más falsos que verdaderos, teniendo cifras muy parecidas entre ellas, el reconocimiento verdadero es muy reducido si comparamos las entradas de la matriz. Nuevamente si vemos el gráfico ROC, podremos observar que existe una tendencia del modelo por generar más falsos negativos que falsos positivos. Por otro lado, el valor del área bajo la curva nos indica que este modelo no es un buen clasificador. Prosiguiendo con el siguiente modelo, el cual tiene una alteración de $\gamma = 20\,\%$, podemos observar que no existe una diferencia drástica en las entradas, pero si podemos evidenciar que existe una mejora en la generación de negativos (tanto falsos como verdaderos). Por otro lado, el valor del área bajo la curva es muy parecido, lo que valida la mala

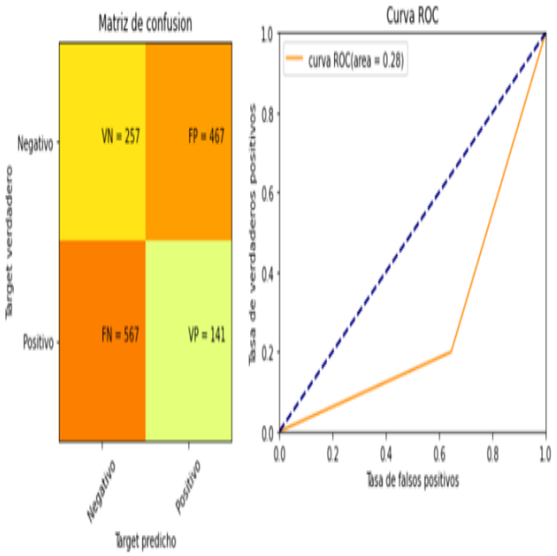


(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 30\,\%$

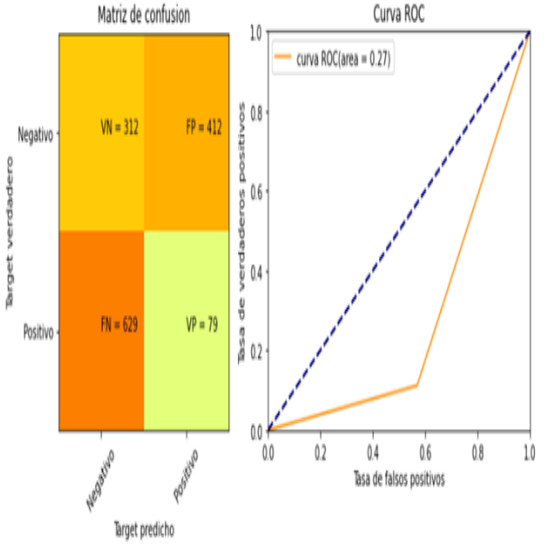


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 40\,\%$

Figura 4.21: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 50\,\%$

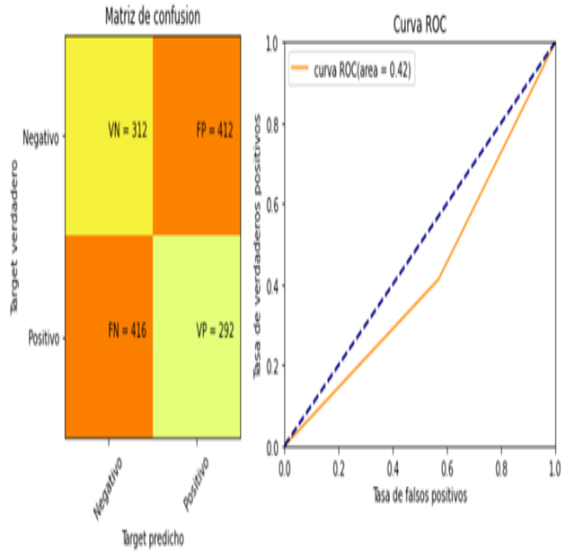


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 60\,\%$

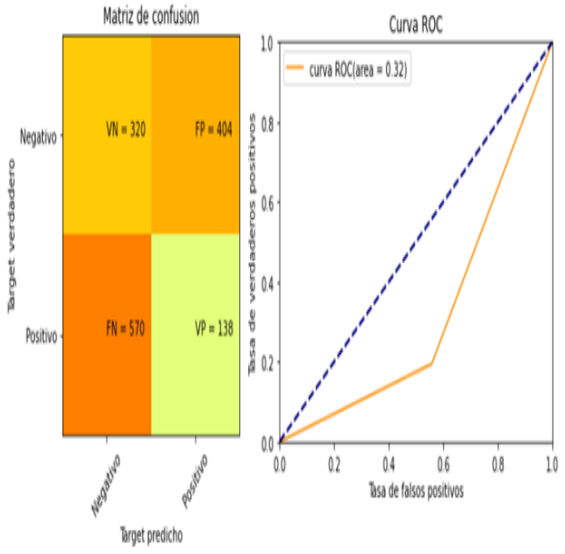
Figura 4.22: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

clasificación del modelo.

En la figura 4.21, a partir de los gráficos relacionados a la perturbación de $\gamma = 30\,\%$, se observa que existe un aumento de los números positivos tanto verdaderos como negativos, pero también se ve una disminución fuerte en los números pertenecientes a los negativos. En el gráfico ROC es posible observar un aumento del ROC-Score a casi el doble con respecto al modelo anterior, esto

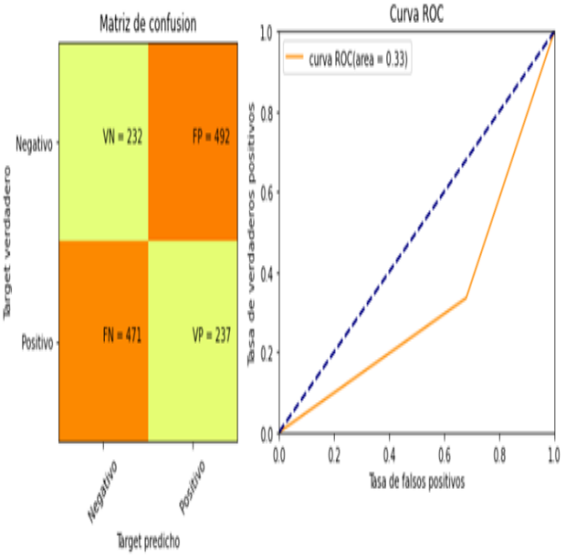


(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 70\,\%$

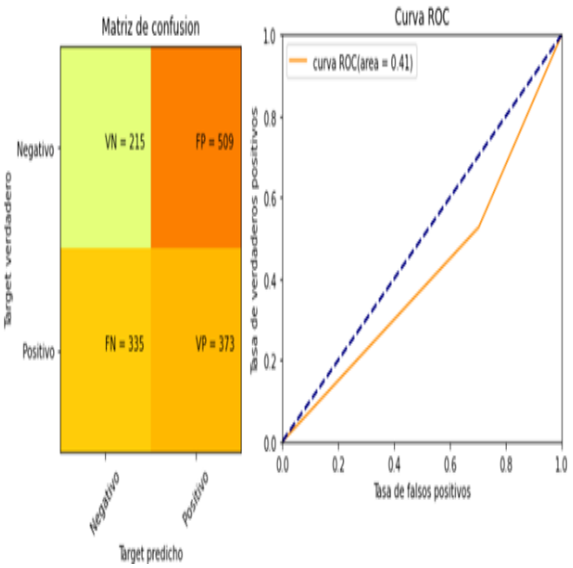


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 80\,\%$

Figura 4.23: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 90\,\%$



(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 100\,\%$

Figura 4.24: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

se debe a que se tiene una polarización de los resultados (generando mayormente más positivos que negativos, no discerniendo de una forma correcta). Prosiguiendo con el análisis de esta figura, ahora con los resultados obtenidos para un $\gamma = 40\,\%$, es posible observar que se mantiene la misma tendencia de generar más positivos que negativos (tanto falsos como verdaderos), los números tienden a incrementarse del lado positivo, disminuyendo claramente los del lado negativo, esto

tendría repercusión en el gráfico ROC, tanto en el área bajo la curva como en la tendencia del vértice a pertenecer a el sector falso positivo.

En la figura 4.22, en los gráficos pertenecientes a una perturbación de $\gamma = 50\,\%$, es posible observar que existe un retorno a los casos con gama $10\,\%$ y $20\,\%$, puesto que existe una tendencia más hacia los falsos negativos que a los casos positivos, con respecto a el modelo predecesor. Existe un aumento considerable en los casos negativos, manteniendo los falsos positivos, la curva ROC nos entrega esta información, con la clara tendencia del modelo a generar falsos negativos, con un ROC-Score bajo, lo que nos indicaría que el poder de discernimiento es bajo. Continuando con el análisis de los gráficos de esta figura, en los gráficos pertenecientes a un $\gamma = 60\,\%$, es posible observar que se mantiene la tendencia del modelo anterior, con un leve aumento de los casos negativos, obteniendo un modelo más polarizado.

En la figura 4.23, para el modelo con una perturbación $\gamma = 70\,\%$, es posible observar que existe una tendencia a equilibrar los casos falsos y aumentar el etiquetado correcto de la muestra, este equilibrio también se ve reflejado en el gráfico ROC, pues el vértice de la figura se encuentra central, es importante mencionar que existe un aumento considerable del área bajo la curva. Prosiguiendo con el análisis del siguiente modelo (modelo de perturbación $\gamma = 80\,\%$), es posible ver que existe una mejora en el discernimiento de los casos negativos verdaderos, pero un empeoramiento en el de positivo verdaderos, aumentando la clasificación negativa total (falsos y verdaderos), lo cual se ve reflejado en el gráfico ROC del cual también es posible desprender el área bajo la curva entregada para este modelo. Ambos modelos tienen ROC-Score bajo, pero el modelo de $70\,\%$ tiende a ser mejor clasificador.

En la figura 4.24, para el modelo con perturbación $\gamma = 90\,\%$ se tiene que, el modelo tiende a equilibrar nuevamente las entradas y por lo tanto no tener una orientación completa, el modelo tiene a reconocer patrones de una forma baja, aumentando el campo de los positivos a seleccionar. El área bajo la curva tiende a disminuir con respecto al modelo anterior. Finalmente, para el último modelo (modelo con perturbacion $\gamma = 100\,\%$), se tiene una polarización del modelo, pues tiende a reconocer más casos positivos que negativos, siendo los casos falsos negativos los más afectados por esta polarización, esto se ve evidenciado en el gráfico ROC, el cual nos entrega una inclinación sobre el lado falso verdadero, también se ve un aumento del ROC-Score.

A manera de resumen dejaremos los resultados en el cuadro 4.5 obtenidos para cada uno de los porcentajes, en el cual se pueden encontrar los valores para el ROC-Score, cantidad de positivos y el valor de la función objetivo[4]. Los gráficos asociados a la tabla resumen 4.6, pueden encontrase en la figura 4.31.

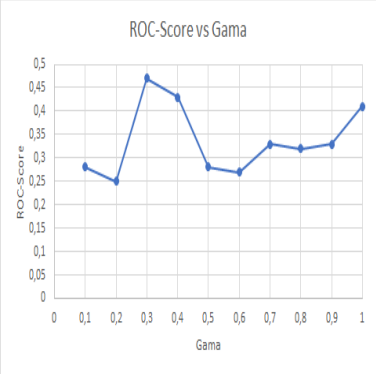
Los resultados para un set de datos 5000 y una capacidad de $|V_t| = 5$, se pueden encontrar en las figuras 4.26, 4.27, 4.28, 4.29 y 4.26. A partir de los resultados recientemente expuestos es posible visualizar que, en su mayoría, las matrices de confusión y los gráficos ROC muestran un claro predominio de los sectores falsos, exceptuando por algunos casos, en vez de los verdaderos tal como se esperaría de un buen clasificador. Con esto es necesario mencionar que el área bajo la curva o *ROC-Score* es siempre menor a un 0,6, lo cual nos indicaría que estos modelos como clasificadores, son deficientes, pues no cubren de una manera correcta a los casos verdaderos, pero pueden utilizarse para generar un modelo que nos ayude a visualizar mejor los falsos verdaderos.

En la figura 4.26, podemos apreciar que en el modelo con perturbación $\gamma = 10\,\%$ vemos que

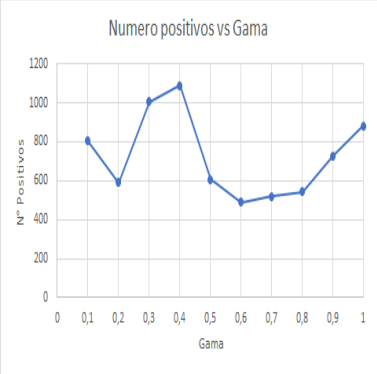
[4]Los resultados de la función objetivo fueron normalizados para una mejor comprensión

Gama γ	ROC-Score	N° 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 %	0.327	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

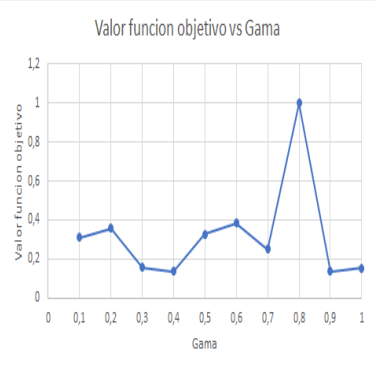
Cuadro 4.5: Tabla resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas, con un conjunto $|V_t| = 2$. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.



(a) Resumen ROC-Score vs Gama



(b) Resumen Numero de positivos vs Gama



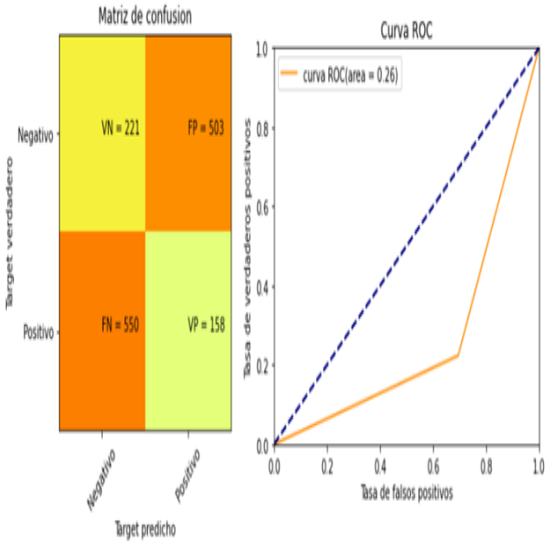
(c) Resumen valor de función objetivo vs Gama

Figura 4.25: Resultados de las relaciones encontradas en el cuadro resumen 4.5

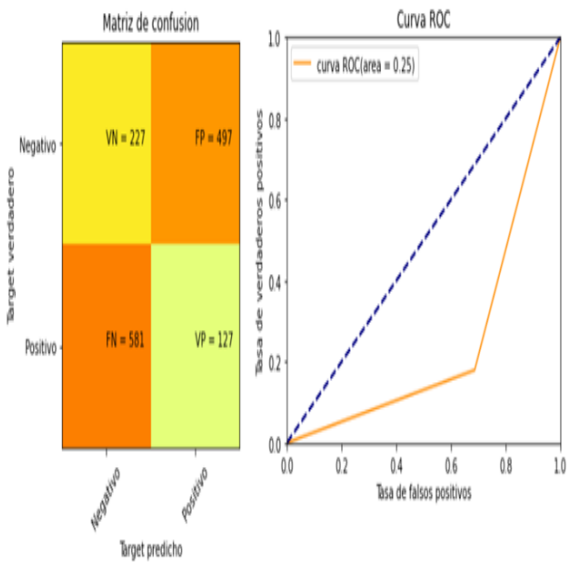
el sector negativo es un claro dominante en la matriz de confusión, por otra parte, las entradas falsas tienden a tener una similitud entre ellas. Si revisamos el gráfico ROC, es posible percibir que el poder de clasificación de este modelo es bajo pues, el área bajo la curva ROC es deficiente. Prosiguiendo con el siguiente modelo es posible ver que las características del modelo anterior se mantienen, variando en una pequeña cantidad el valor en las entradas de la matriz de confusión.

En la figura 4.27, en el modelo con una perturbación de $\gamma = 30\,\%$ es posible ver que tiene una tendencia a generar una mejor clasificación de los casos negativos, manteniendo la parte positiva sin mucha diferencia con respecto al modelo de $\gamma = 20\,\%$, estos cambios aumenta el área bajo la curva generada por la curva ROC, mejorando el modelo como clasificador. Prosiguiendo con el modelo de perturbación $\gamma = 40\,\%$ es posible ver que tiende a generar un cambio en la clasificación, pues genera más casos positivos falsos y mejora el discernimiento de los casos verdaderos positivos, esto con lleva a que el área bajo la curva se vea afectado de forma negativa.

En la figura 4.28, es posible ver que en el modelo que fue generado con una perturbación $\gamma = 50\,\%$,

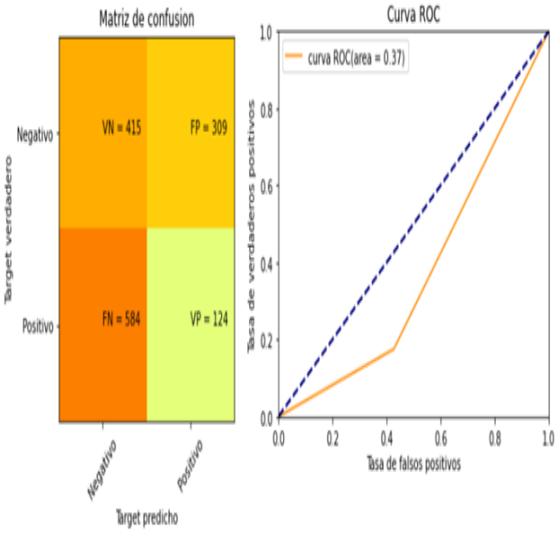


(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 10\,\%$

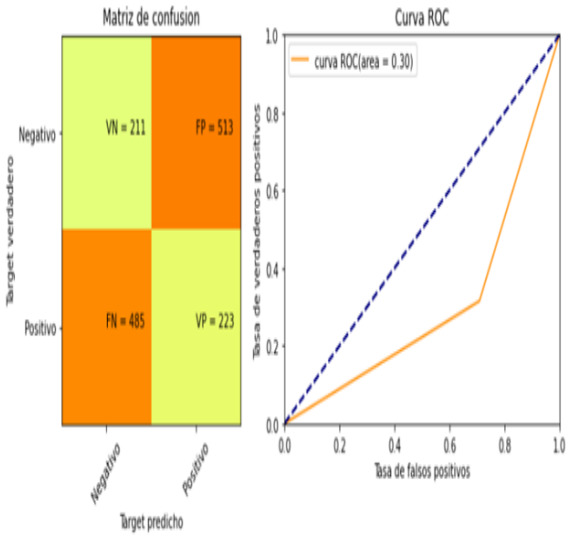


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 20\,\%$

Figura 4.26: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.



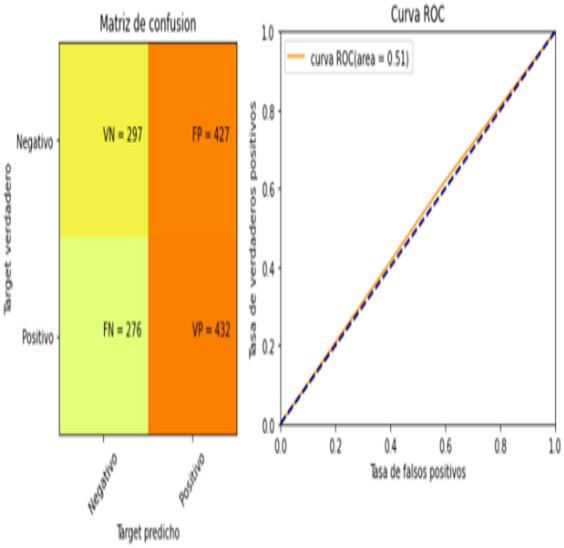
(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 30\,\%$



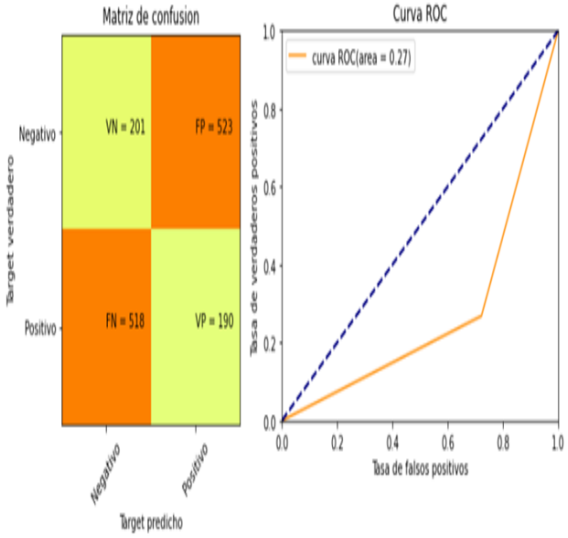
(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 40\,\%$

Figura 4.27: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

tiende a generar una clasificación mayoritariamente positiva, pues tiende a reconocer bien los casos verdaderos positivos. Mas aun, los falsos positivos se mantienen altos, lo que nos lleva a que el modelo sea polarizado hacia el sector positivo. También aumenta los casos negativos verdaderos, este aumento en las entradas verdaderas tiende a aumentar el área bajo la curva del gráfico ROC, tendiendo a una curva en la zona superior (muy pequeña). Por otro lado, el modelo con perturbación

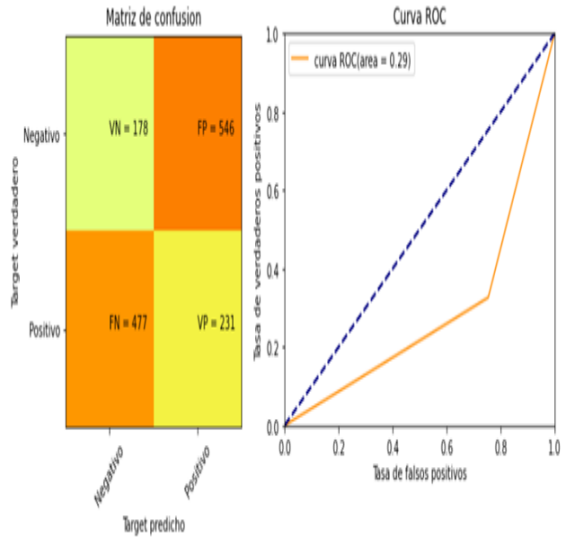


(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 50\,\%$

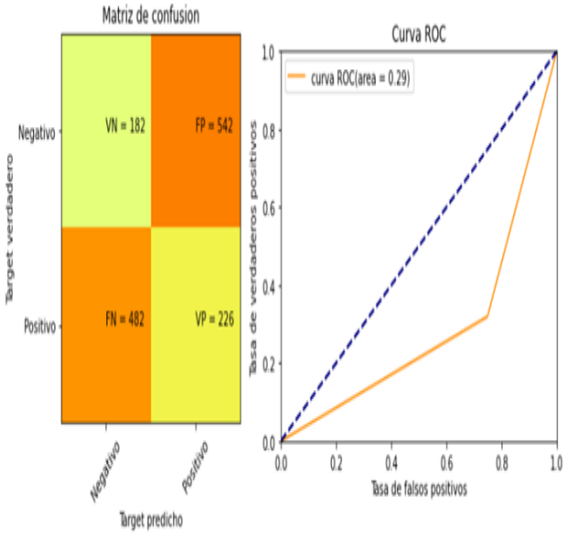


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 60\,\%$

Figura 4.28: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 70\,\%$

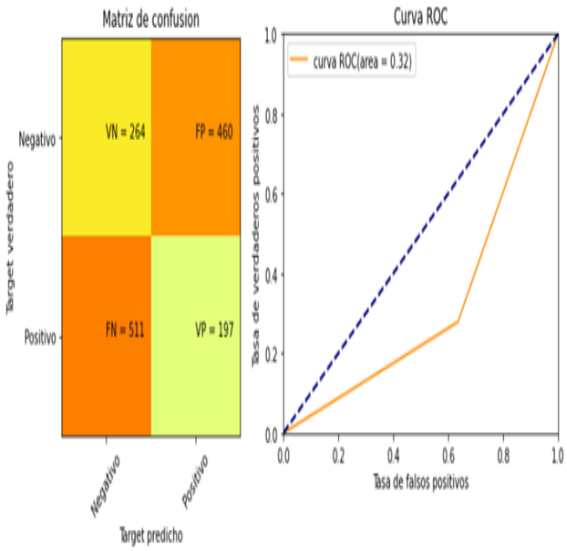


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 80\,\%$

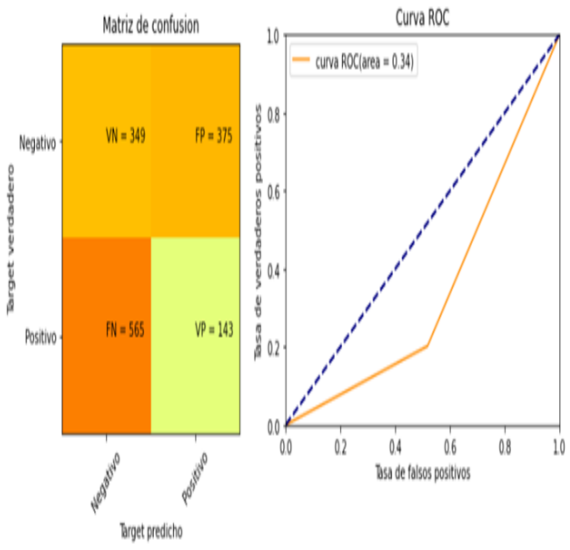
Figura 4.29: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

$\gamma = 60\,\%$, tiende nuevamente a disminuir el reconocimiento verdadero y a equilibrar las entradas diagonales (ambas diagonales), lo que lleva a una disminución del área bajo la curva en el gráfico ROC.

En la figura 4.29, es posible apreciar que el modelo con perturbación $\gamma = 70\,\%$, tiende a generar nuevamente una tendencia a clasificar de manera positiva, disminuyendo la cantidad reconocida



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 90\,\%$



(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 100\,\%$

Figura 4.30: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

como negativa, aunque esta disminución no es muy importante, tiende a generar un área bajo la curva mejor. Prosiguiendo con el modelo cuya perturbación fue de $\gamma = 80\,\%$, es posible ver que tiende a mantener los resultados anteriores, con pequeñas diferencias de clasificación (no más de 10 prospectos pasaron de verdaderos a falsos), esto hace que el área bajo la curva se mantenga en la misma cifra.

En la figura 4.30, es posible visualizar que en el modelo con una variación de $\gamma = 90\,\%$ tiende a mejorar la clasificación negativa por sobre la positiva, esto hace que las entradas de la diagonal secundaria se inviertan en magnitud. Como el aumento de la clasificación verdadera negativa es mayor que la disminución en la clasificación verdadera positiva, esto genera un aumento en el valor del área bajo la curva ROC. Por último, en el modelo con una variación de $\gamma = 100\,\%$, es posible percibir que existe un aumento considerable en la clasificación verdadera negativa y una disminución menor en la clasificación verdadera positiva, siguiendo la misma línea de discriminación que el modelo anterior, lo cual influye en el ROC-Score, aumentándolo en un 0,02.

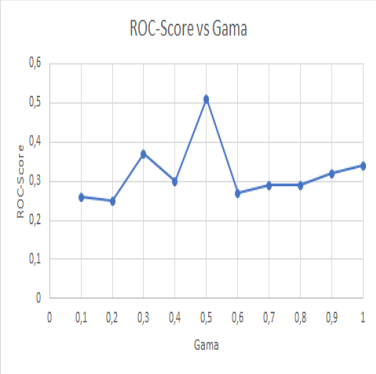
A manera de resumen dejaremos los resultados en el cuadro 4.6 obtenidos para cada uno de los porcentajes de γ, la cual contiene los resultados de ROC-Score, cantidad de positivos y el valor de la función objetivo[5]. Los gráficos asociados a la tabla resumen 4.6, pueden encontrase en la figura 4.31.

Por ultimo, los resultados asociados a un set de datos de 5000 y una capacidad de $|V_t| = 8$, se encuentran en las figuras 4.32, 4.33, 4.34, 4.35 y 4.36. De estos resultados, es posible ver que, en su totalidad, las matrices de confusión y los gráficos ROC nos muestran un claro predominio de los sectores falsos, exceptuando por algunos casos, en vez de los verdaderos tal como se esperaría de un buen clasificador. Con esto es necesario mencionar que el área bajo la curva o *ROC-Score* es siempre menor a un 0,6, lo cual nos indicaría que estos modelos como clasificadores, son deficientes,

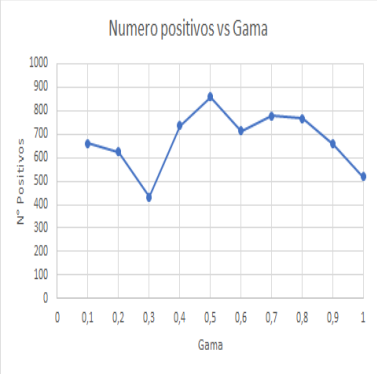
[5]Los resultados de la función objetivo fueron normalizados para una mejor comprensión

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

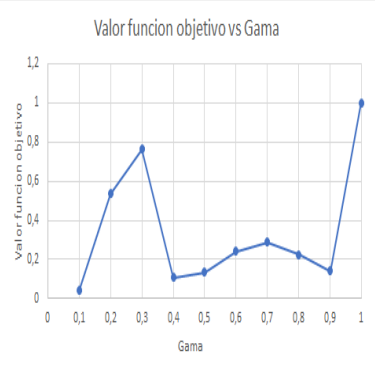
Cuadro 4.6: Tabla resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas, con $|V_t| = 5$. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.



(a) Resumen ROC-Score vs Gama



(b) Resumen Numero de positivos vs Gama



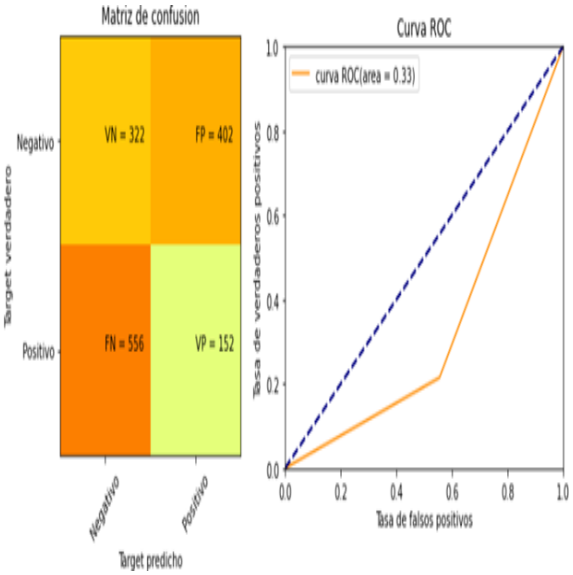
(c) Resumen valor de función objetivo vs Gama

Figura 4.31: Resultados de las relaciones encontradas en el cuadro resumen 4.6.

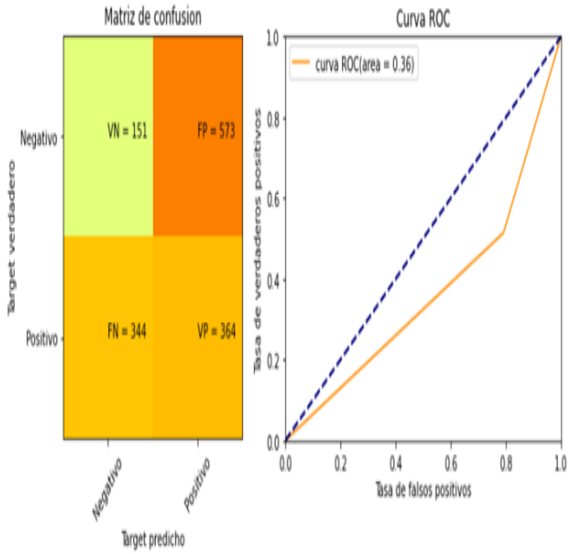
pues no cubren de una manera correcta a los casos verdaderos, pero pueden utilizarse para generar un modelo que nos ayude a visualizar mejor los falsos verdaderos

En la figura 4.32, es posible visualizar que en el modelo que se representa en 4.32a, existe un reconocimiento negativo y positivo, pero también genera una mayor cantidad de falsos negativos que falsos positivos, lo que nos hace pensar que el modelo esta polarizado hacia los casos negativos. El área bajo la curva hace alusión a la baja clasificación correcta. Por otro lado, el modelo con una variación porcentual de $\gamma = 20\,\%$ que se encuentra representado por la figura 4.32b tiende a invertir la matriz de confusión, cambiando la polarización a el sentido positivo, junto con este cambio existe una variación mínima de los resultados , lo cual es representado en las cantidades de la matriz de confusión, estas variaciones generan un aumento en el área bajo la curva con respecto al modelo anteriormente estudiado.

En la figura 4.33, es posible ver que el modelo con una variación porcentual de $\gamma = 30\,\%$ el cual esta representado por la figura 4.33a, nuevamente genera un cambio radical del modelo pues cambia

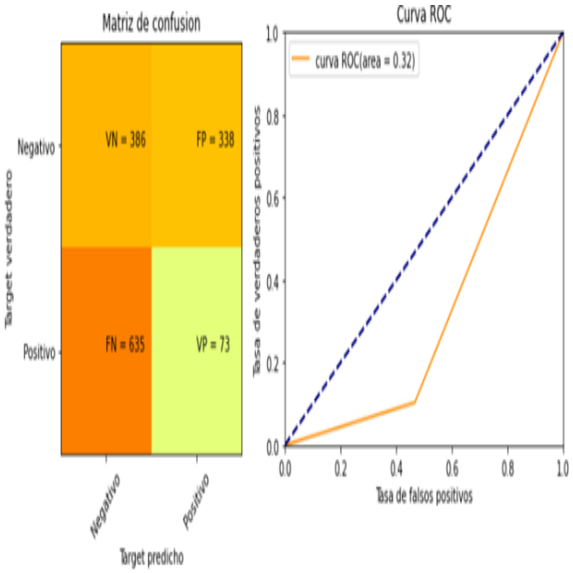


(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 10\,\%$

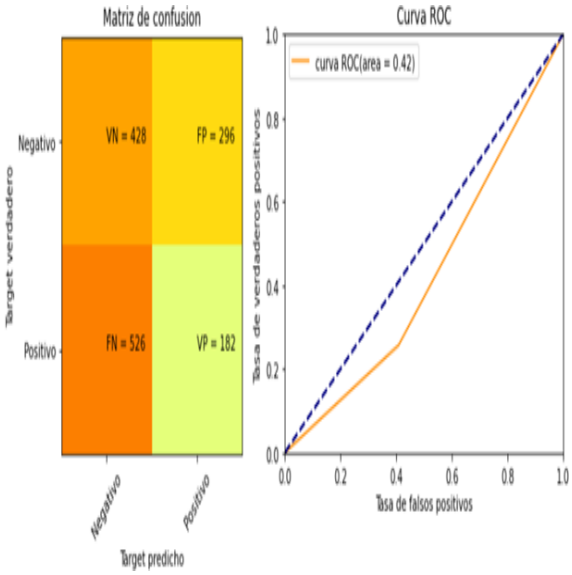


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 20\,\%$

Figura 4.32: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.



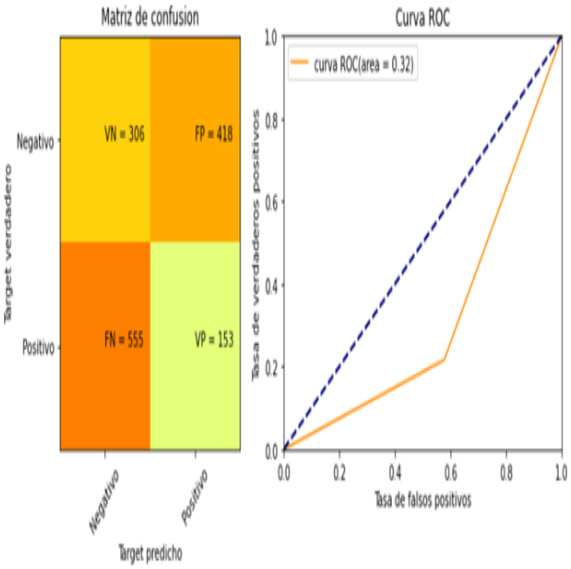
(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 30\,\%$



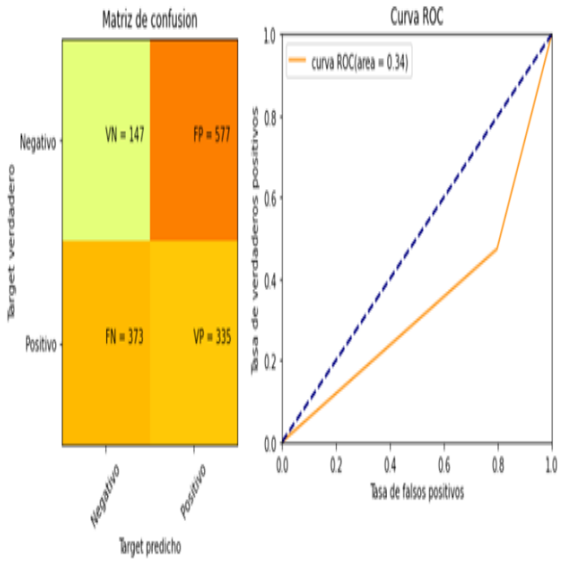
(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 40\,\%$

Figura 4.33: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

la orientación de la matriz de confusión, volviendo a generar una similitud en los valores negativos y un desequilibrio en los valores positivos. Puesto que estos cambios no generan una transformación en las cifras en general. El área bajo la curva sufre una variación de 0,04 con respecto al modelo anterior. Por otro lado, el modelo que está representado en la figura 4.33b evidencia un cambio en la zona de los casos negativos, inclinándose por los negativos verdaderos más que por los falsos

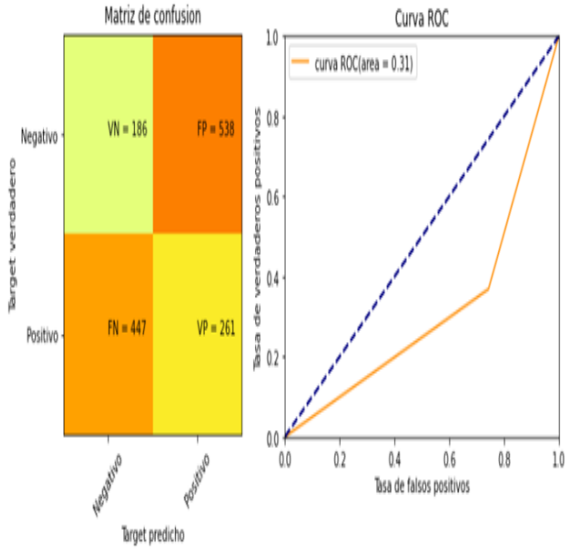


(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 50\,\%$

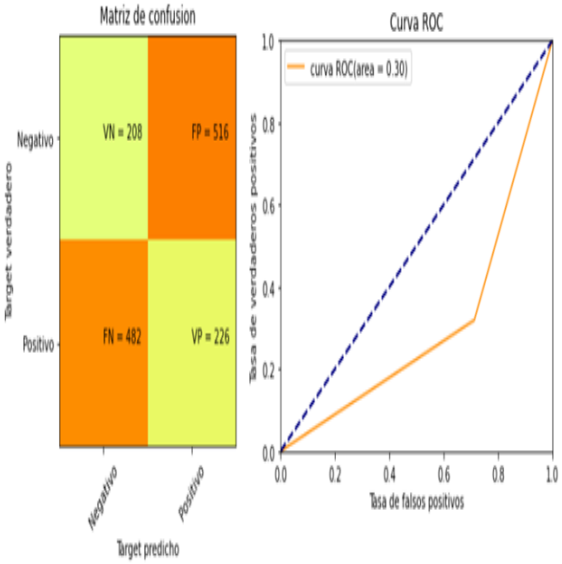


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 60\,\%$

Figura 4.34: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 70\,\%$

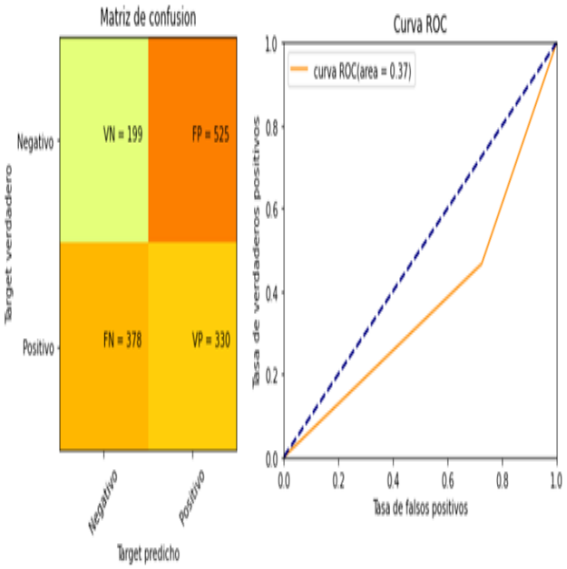


(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 80\,\%$

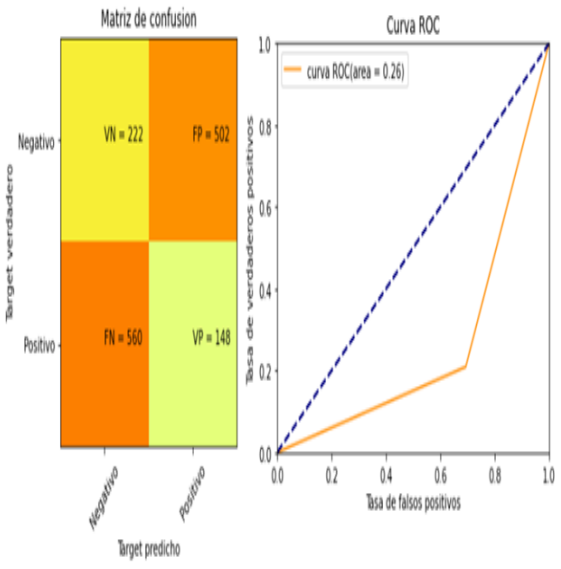
Figura 4.35: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

positivos. Aun así, el sector positivo verdadero aumenta, estos resultados influyen en el aumento del área bajo la curva ROC en un 0,10.

En la figura 4.34, es posible ver que en el modelo representado por la figura 4.34a, muestra un empeoramiento en el reconocimiento de prospectos negativos, tendiendo a generar más falsos positivos que verdaderos negativos. Por otro lado, el sector de positivos mantiene la distribución



(a) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 90\,\%$



(b) Resultado de algoritmo DRM-SVM memoria corta para un $\gamma = 100\,\%$

Figura 4.36: Resultados de SVM-DRM de memoria corta para el conjunto de datos bancarios.

de las cifras, con el sector de falsos negativos siendo el predominante. Prosiguiendo con el análisis de los resultados, el modelo que tuvo una alteración $\gamma = 60\,\%$ el cual esta representado por la figura 4.34b , muestra una inversión en la matriz, los resultados negativos tienden a acumularse entre los falsos positivos y los resultados positivos tienden a entrar en un equilibrio, puesto que los resultados variaron en cantidades similares, el área bajo la curva tiende a variar una cantidad mínima, manteniendo la calidad del clasificador.

En la figura 4.35, es posible ver que en el modelo cuya alteración fue de $\gamma = 70\,\%$ el cual esta representado por la figura 4.35a, que el sector negativo mantiene la distribución anterior, acumulando prospectos sobre los falsos negativos, en los casos positivos es posible ver que disminuye la cantidad de verdaderos positivos aumentando los falsos negativos. Dicha variación hará que el modelo tenga un decaimiento en el área bajo la curva ROC de un 0,03 con respecto al modelo anterior. Por otro lado, el modelo con alteración agregada de $\gamma = 80\,\%$ el cual esta representado por la figura 4.35b tiende a generar un equilibrio entre las partes que son verdaderas y los falsamente etiquetados. El valor que se ve más afectado es la etiqueta correctamente clasificada como positiva, esto tiene un efecto sobre la calidad del clasificador en el área bajo la curva, la cual disminuye en un 0,01 con respecto al modelo anteriormente analizado.

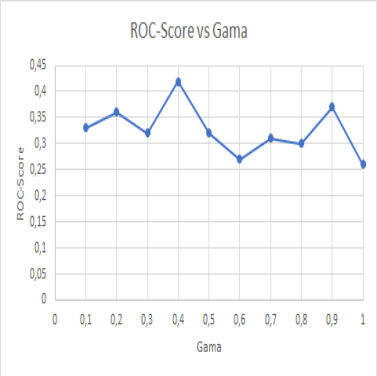
En la figura 4.36, es posible ver que en el modelo cuya alteración es $\gamma = 90\,\%$ (ver figura 4.36a), es posible visualizar que vuelve a generar un aumento en el reconocimiento positivo verdadero y una pequeña disminución en la clasificación de los prospectos negativos, lo que lleva a un aumento en el área bajo la curva ROC. Esto nos da a entender que tiende a ser un buen clasificador entre sus pares. Finalmente, el último modelo (ver 4.36b), nos muestra una tendencia a generar más falsos negativos rompiendo el esquema del último clasificador, aumentando la cantidad de verdaderos positivos en menor cantidad, lo que lleva a que el modelo disminuya el ROC-Score.

A manera de resumen dejaremos los resultados obtenidos en el cuadro 4.7 para cada uno de los

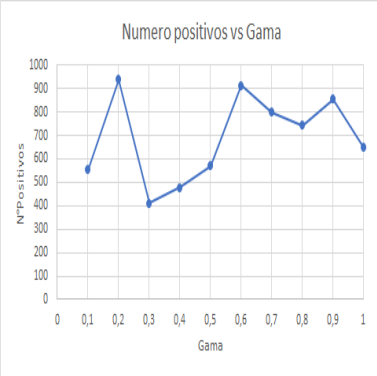
porcentajes de γ con los resultados de ROC-Score, cantidad de positivos y el valor de la función objetivo[6]. Los gráficos asociados a el cuadro resumen 4.7, pueden encontrase en la figura 4.37

Gama γ	**ROC-Score**	**N° 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

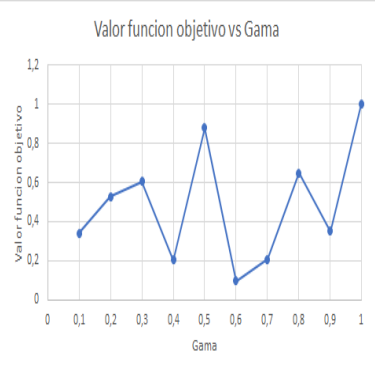
Cuadro 4.7: Tabla resumen de resultados obtenidos en los distintos modelos para una población de 5000 personas, con $|V_t| = 8$. El porcentaje en gama, hace alusión a la proporción utilizada del vector de parámetros para perturbar a este mismo.



(a) Resumen ROC-Score vs Gama



(b) Resumen Numero de positivos vs Gama



(c) Resumen valor de función objetivo vs Gama

Figura 4.37: Resultados de las relaciones encontradas en el cuadro resumen 4.7

A partir de las figuras 4.18, 4.19 , 4.25, 4.31 y 4.37, es posible evidenciar que no existe un comportamiento monótono de las variables que estábamos utilizando para contraponer el factor gama. Esto pudo deberse a que nuestro algoritmo no necesariamente genera una mejora estadística en los modelos, si no que genera parámetros del modelo mas robustos ante perturbaciones. Esto también con llevó a que no se pudiera medir el valor agregado de utilizar la técnica de Diametrical Risk Minimization sobre nuestro algoritmo de Support Vector Machines.

Si comparamos los cuadros 4.5, 4.6 y 4.7 es posible ver el máximo valor de ROC-Score se encontró en el modelo DRM-SVM con un grupo $|V_t| = 5$, el cual con llevo a la mejor puntuación de positivos entre todos los resultantes de sus pares. Este resultado fue el único entre estas treinta pruebas que genero una curva recta en el gráfico ROC. Por otra parte, la gran cantidad de positivos generados,

[6]Los resultados de la función objetivo fueron normalizados para una mejor comprensión

provienen de la entrada para los positivos, los cuales superaron en una mínima cantidad a los falsos positivos. También es importante mencionar que el promedio de ROC-Score y positivos en modelo DRM-SVM con un grupo $|V_t| = 2$ es mayor que en los otros 2 tipos de modelos.

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo, introducimos técnicas que ayudan en la predicción de un suceso bancario en los meses futuros a el estudio y posteriormente buscamos como mejorar el modelamiento matemático con un cambio de perspectiva en la forma de solucionar el problema.

En este trabajo analizamos datos provenientes de una institución bancaria. Realizamos un análisis descriptivo de la base de datos recolectada, en el cual estudiamos el comportamiento desde 3 puntos de vista distintos los cuales fueron, el descriptivo general, de distribución e histórico.

En la primera parte de este trabajo lidiamos con el problema de predicción del curse de un beneficio bancario y presentamos los algoritmos que utilizaríamos para trabajar en la resolución de dicho problema. Propusimos estos modelos de clasificación por ser los más utilizados en la industria y con el mejor rendimiento. Queríamos estudiar el rendimiento de los algoritmos en su clasificación y también el consumo de recursos que demanda cada uno de ellos. Según los resultados obtenidos, pudimos mostrar que le mejor algoritmo para predecir el curso del beneficio bancario fue XGBoost dentro de los algoritmos, utilizando las métricas ROC-Score y F1-Score para medir la eficiencia de este. A su vez también es importante mencionar que XGBoost fue el modelo que tuvo el menor tiempo de ejecución dentro de los modelos estudiados, pero en recursos usados fue el que más utilizo. También pudimos corroborar la buena precisión del modelo mediante la inspección de la matriz de confusión resultante de la discriminación. Por último, el gráfico de distribución por grupo se pudo percibir armoniosa a medida que cambiaban estos mismos, manteniendo la disminución de los casos positivos a medida que nos acercábamos a los grupos finales y aumentando los casos negativos. El resto de los modelos obtenidos, tienen un buen rendimiento en el discernimiento, catalogando a Suport Vector Machines y Random Forest como buenas técnicas para este tipo de trabajos.

Para la segunda parte de este trabajo, lidiamos con el problema de robustez de soluciones utilizando la técnica de Diametrical Risk Minimization, para lo cual presentamos 2 variantes de esta técnica. Dichos métodos fueron utilizados en conjunto con el modelo de Support Vector Machines. Propusimos la variante de este modelo pues, fue el modelo que más incertidumbre nos generó al momento de los resultados, ya que no encontramos como medir el cambio del modelo frente a alteraciones. Deseábamos estudiar el comportamiento la técnica de DRM en ambas variantes en conjunto a este modelo. Para dicho cometido, lo primero que hicimos fue, analizar las variantes de SVM con el set de datos de iris en la sub-sección 4.3.1. Según los resultados entregados por dichos modelos, el que tuvo mejor rendimiento fue DRM-SVM (el algoritmo 2), pues entrego al menos 3 variantes que

generan una clasificación dentro del conjunto de datos, uno de ellos fue una clasificación perfecta y el resto tendía a generar una clasificación deficiente con tendencia a generar falsos positivos. Por otro lado, el algoritmo DRM-SVM con memoria corta, tuvo un rendimiento deficiente sobre nuestra muestra, pues no genero ninguna variante de la clasificación que fuera aceptable. Con ello pudimos corroborar que es lo que hace el algoritmo de DRM sobre nuestro método de clasificación de SVM. Una vez que vimos el comportamiento de DRM con el conjunto de datos de iris, realizamos pruebas de los métodos sobre el conjunto de datos reales bancarios. Lo que deseábamos esta vez era generar una comparativa de dichas variantes de SVM sobre un problema de la vida real, con ello estudiamos el comportamiento de ambos algoritmos (de manera clasificatoria) y realizamos un estudio para entender los beneficios que nos entregan dichas variantes con respecto a 3 variables, las cuales fueron valor de ROC-Score para el modelo, numero de positivos generados por el modelo y el valor de la función objetivo generada por la solución. A partir de los resultados mostrados en la sub-sección 4.3.2, fue posible ver que los resultados para el algoritmo 2 fueron los mejores de estas pruebas, pues el modelo con el máximo de casos positivos y mayor ROC-Score fue generado por una variante de este modelo, generando así los mejores resultados desde el punto de vista clasificatorio, pero aun así no fue posible generar un modelo con un alto rendimiento. Estudiando los modelos generados por ambas variantes fue posible ver que para dichos datos, no generamos modelos que fueran eficiente desde un punto de vista estadístico, pero lo que si se pudo generar fueron modelos con una perspectiva distinta, ya que en dichos modelos generalmente predominaban los casos falsos (positivos y negativos) antes que los verdaderos. Esto nos entrega un margen de prospectos que pudieran tener un comportamiento distinto al de la etiqueta target anexada, dicha característica, puede ayudarnos a entender más la calidad de prospectos obtenidos en la muestra, entregándonos un margen de casos que puedan ser utilizados para campañas comerciales (tal como en este caso que estudiábamos un beneficio bancario que se deseaba publicitar). También a partir de los resultados obtenidos por ambas variantes de nuestro modelo, intentamos analizar la conducta de las variables que a nuestro parecer generarían un comportamiento monótono con respecto a la variación de perturbación γ. Según los resultados obtenidos de manera gráfica en los cuadros resumen de los resultados y en los gráficos de estos mismos en la sub-sección 4.3.2, fue posible ver que no existe tal comportamiento, pues las mediciones que utilizamos provenían de medidas estadísticas, las cuales no necesariamente aumentarían en un modelo robusto, es por esto que no pudimos medir de alguna forma el comportamiento de los modelos ante la variación de perturbación.

Como trabajo futuro que podemos continuar después de esta tarea es probar las variantes del DRM con respecto a los dos modelos que nuestro trabajo tuvo como objetivo estudiar en el capítulo 3.2 y así poder descubrir más sobre el comportamiento de nuestra técnica para robustecer modelos. También podemos utilizar nuestras técnicas para estudiar otros sets de datos y con los resultados generar campañas de marketing para testear estos mismos.

Así mismo para, para el problema de predicción al uso de un beneficio bancario, se puede continuar evaluando modelos de clasificación que tengan base en otro tipo de aprendizaje, tales como las redes neuronales.

Apéndice A

Anexo

A.0.1. Procedimiento SVM

En este apartado intentaremos describir los distintos algoritmos de SVM con los cuales se trabajo para realizar los distintos test. Es importante mencionar que para todos los experimentos llevados acabo con SVM, se utilizo un kernel de tipo lineal[1].

- Scikit-learn(LIBLINEAR): Dado un vector de entrenamiento $x_i \in \mathbb{R}^n$, y un vector $y \in \mathbb{R}^l$ tal que $y_i = \{1, -1\}$, el problema que resuelve este algoritmo son 2 tipos de problemas, los cuales son:

$$\min_{w \in W} \frac{1}{2} \parallel w \parallel + C \sum_{i=1}^{l} (\text{máx}(0, 1 - y_i w^T x_i)$$

$$\min_{w \in W} \frac{1}{2} \parallel w \parallel + C \sum_{i=1}^{l} (\text{máx}(0, 1 - y_i w^T x_i)^2$$

  De los cuales pueden ser resueltos mediante una transformación a su problema dual, el cual se ve de la siguiente forma:

$$\min_{\alpha} \frac{1}{2} \alpha^T \bar{Q} \alpha - e^T \alpha$$
$$\text{sujeto a,} 0 \leq \alpha_i \leq U$$

  Donde e es el vector de unos, $\bar{Q} = Q + D$, D es una matriz diagonal, y $Q_{ij} = y_i y_j x_i^T x_j$. Para una perdida lineal, $U = C$ y $D_{ii} = 0 \forall i$. Para una perdida lineal cuadrática, $U = \infty$ y $D_{ii} = \frac{1}{2C} \forall i$. Para solucionar este problema de optimización, el algoritmo recurre a un método llamado SDM (Sequential Dual Methods) para la formulación de Crammer-Singer. Crammer

[1]Kernel lineal: Para una salida prevista $t = 1$ e $y = w \cdot x + b$, la pérdida lineal de la predicción y se define como:

$$\ell(y) = \text{máx}(0, 1 - t)$$

Tenga en cuenta que y debe ser la salida "sin procesar"de la función de decisión del clasificador, no la etiqueta de clase predicha. Por ejemplo, en SVM lineales, $y = \mathbf{w} \cdot \mathbf{x} + b$, donde $(\mathbf{w}, b)$ son los parámetros del hiperplano y $\mathbf{x}$ es la (s) variable (s) de entrada.

and Singer proponen una solución al problema primal

$$\min_{\{w_m\},\{\xi_i\}} \frac{\lambda}{2} \sum_m \parallel w_m \parallel^2 + C \sum_i \xi_i$$

$$\text{s.t.} w_{y_i}^T x_i - w_m^T x_i \geq e_i^m - \xi_i \forall m, i \tag{A.1}$$

, donde $C > 0$ es el parámetro de regularización, w_m es el vector de pesos asociados a la clase m, $e_i^m = 1 - \delta_{y_i,m}$ y $\delta_{y_i,m} = 1$ si $y_i = m$, $\delta_{y_i,m} = 0$ si $y_i \neq m$. Note que, en la ecuación (6,1) la restricción para $m = y_i$ corresponde a una restricción no negativa, $\xi_i \geq 0$. La función de decisión es:

$$\text{argmax}_m \, w_m^T x. \tag{A.2}$$

El problema dual de (6.1), desarrollado en la línea de [17] y [18], involucra un vector α teniendo variables duales $\alpha_i^m \forall m, i$. El vector w_m se define vía α como:

$$w_m(\alpha) = \sum_i \alpha_i^m x_i \forall m \tag{A.3}$$

En muchos párrafos posteriores, simplemente escribiremos $w_m(\alpha)$ en vez de w_m. Sea $C_i^m = 0$ si $y_i \neq m$, $C_i^m = C$ si $y_i = m$. El problema dual es

$$\min_\alpha f(\alpha) = \frac{\lambda}{2} \sum_m \parallel w_m(\alpha) \parallel^2 + \sum_i \sum_m e_i^m \alpha_i^m$$

$$\text{s.a.} (\alpha_i^m \leq C_i^m \forall m, \sum_m \alpha_i^m = 0) \forall i, m. \tag{A.4}$$

El gradiente de f juega un rol importante y está dado por:

$$g_i^m = \frac{\partial f(\alpha)}{\partial \alpha_i^m} = w_m(\alpha)^T x_i + e_i^m \forall i, m. \tag{A.5}$$

Si $\bar{n}$ es el promedio de los elementos no ceros por entrenamiento, entonces en promedio la evaluación de cada g_i^m toma un esfuerzo de $O(\bar{n})$. Optimalidad de α para (6.4) puede ser chequeada usando la cantidad,

$$v_i = \max_m g_i^m - \min_{m:\alpha_i^m < C_i^m} g_i^m \forall i \tag{A.6}$$

Para un i dado, los valores de m que alcanzan el máximo y el mínimo en (6.6) juegan un rol importante y los denotaremos como:

$$M_i = \text{argmax}_m \, g_i^m \text{y} m_i = \text{argmin}_{m:\alpha_i^m < C_i^m} \, g_i^m \tag{A.7}$$

desde (6.6) es claro que v_i es no negativo. Optimalidad dual se tiene cuando:

$$v_i = 0 \forall i \tag{A.8}$$

Para una terminación práctica, podemos verificar aproximadamente la optimalidad usando el parámetro de tolerancia, $\epsilon > 0$:

$$v_i < \epsilon \forall i \tag{A.9}$$

Nos referiremos a esto como ϵ-optimalidad. El valor de $\epsilon = 0{,}1$ es una elección buena para una implementación del SDM. SDM consiste en elegir secuencialmente un i a la vez y resolver el problema restringido optimizando solamente $\alpha_i^m \forall m$, teniendo todas las otras variables fijas. Para hacer esto, nosotros dejamos que $\delta\alpha_i^m$ denote el cambio aditivo que se aplicará al mi actual, y optimizamos $\delta\alpha_i^m \forall m$. Sea α_i, $\delta\alpha_i$, g_i y C_i son vectores que reúnen respectivamente los elementos α_i^m, $\delta\alpha_i^m$, g_i^m y C_i^m sobre todo m. Con $A_i = \| x_i \|^2$ el subproblema de optimizar $\delta\alpha_i$ esta dado por:

$$\begin{aligned} &\min_{\delta\alpha_i} \frac{1}{2} A_i \| \delta\alpha_i \|^2 + g_i^T \delta\alpha_i \\ &\text{s.a} \delta\alpha_i \leq C_i - \alpha_i, 1^T \delta\alpha_i = 0 \end{aligned} \tag{A.10}$$

Esto se puede derivar teniendo en cuenta lo siguiente:

- cambiando α_i^m por $\alpha_i^m + \delta\alpha_i^m$ lo que causa que w_m cambie a $w_m + \delta\alpha_i^m x_i$
- $\| w_m + \delta\alpha_i^m x_i \|^2 = \| w_m \|^2 + A_i(\delta\alpha_i^m)^2 + 2(w_m^T x_i)\delta\alpha_i^m$
- $e_i^m \alpha_i^m$ cambia a $e_i^m \alpha_i^m + \delta e_i^m \alpha_i^m$
- Usando la definición de g_i^m en (6.5)
- Utilizando lo anterior en (6.4) y omitiendo todas las constantes que no dependen de $\delta\alpha_i$

El subproblema en (6.10) tiene una forma simple. Para resolverla se discutirán distintos métodos. Supongamos que $A_i > 0, \forall i$. Cramer y Singer ([17], [18]), sugiere 2 metodos para resolver (6.10):

- en un algoritmo $O(k \log(k))$
- en un algoritmo de aproximación iterativa de un punto fijo

Alternativamente, también se podría emplear un método de conjunto activo destinado a problemas de programación cuadrática convexa [12], iniciando con $\delta\alpha_i = 0$. Dado que la hessiana de la función objetivo (6.10) es A_i veces la matriz identidad y las restricciones de esta misma están en forma simple, varios pasos de resolución de ecuaciones lineales del método del conjunto activo se pueden realizar de forma analítica y económica. La implementación que utiliza LIBLINEAR se basa en este método.

Algorithm 4 SDM para Crammer-Singer

Iniciar α y el correspondiente w_m $\forall m$.
Hasta que (6.8) se mantenga en un ciclo completo sobre muestras, haga lo siguiente:

for *i = 1,...,l* **do**
(a) Calcular g_i^m $\forall m$ y obtener v_i
(b) Si $v_i \neq 0$, resolver (6.10) y actualizar:
$\alpha_i \longleftarrow \alpha_i + \delta\alpha_i$
$w_m \longleftarrow_m + \delta\alpha_i^m x_i$ $\forall m$

Una descripción del algoritmo que utiliza SDM está dado por 4. Si los valores semilla buenos no están disponibles, una forma simple de inicializar el método es iniciar con $\alpha = 0$, esto corresponde con $w_m = 0$ $\forall m$. El siguiente teorema de convergencia explica la convergencia del método SDM.

Teorema 1: Sea α^t el cual denota α en el fin de la t-esima vuelta en el algoritmo 4. Cualquier punto límite de una subsucesión convergente $\{\alpha^t\}$ es un mínimo global de (6.4).

Dem: La demostración es posible encontrarla en [21]

En [20] fue mostrado que SDM para un problema de clasificación binaria tiene un ratio de convergencia lineal, i.e, existe $0 \leq \mu < 1$ y un t_0 tal que:

$$f(\alpha^t) - f(\alpha^*) \leq \mu^{t-t_0}(f(\alpha^{t_0}) - f(\alpha^*)), \forall t \geq t_0,$$

donde f es la función objetivo dual, t es un contador de iteraciones en el algoritmo, α^t es α en el fin de la t-esima iteración, y α^* es la solución dual óptima. Por otro lado el teorema solo implica un convergencia débil del algoritmo 4.

- **Creado:** Para este proceso, se llevó a cabo un estudio del algoritmo y las premisas del problema, y para ello llegamos a que la mejor forma de solucionar el problema de optimizacion de SVM, era realizando un metodo SVM-Stochastic Gradient Descent([26],[31]), con un paso especialmente estudiado.

  El algoritmo es el siguiente:

Algorithm 5 SVM−SGD

Input: dataset,λ,lr,iters,C

Se inicializa:
$w_0 = 0$
$b_0 = 0$
for *i = 1,. . . ,iters* **do**
  Seleccionar $i_t \in \{1, \ldots, |dataset|\}$:
    Sel. un valor random $a \in \{1, \ldots, |dataset|\}$
    $\nu_t = lr * \frac{1}{\lambda*(a+1)}$

  Si $y_i\langle w_i, x_{i_t}\rangle < 1$, entonces:
    $w_{t+1} \leftarrow (1 - \nu_t * \lambda)w_i + C * \nu_t y_{i_t} x_{i_t}$
    $b_{t+1} \leftarrow (b_t - \nu_t * y_{i_t})$
    Si no:
  _$w_{t+1} \leftarrow (1 - \nu_t * \lambda)w_i$

Output: $w_{iters+1}, b_{iters+1}$

En cada iteracion nuestro procedimiento opera de la siguiente forma, se inicia con un vector w en un cero vector. En cada iteracion i del algoritmo, primero se selecciona un valor de nuestro conjunto de datos (x_{i_t}, y_{i_t}) tomando un indice i_t de la muestra. Luego reemplazamos la funcion objetivo con una aproximacion basada en nuestra muestra de entrenamiento (x_{i_t}, y_{i_t}) , lo que queda de la siguiente forma:

$$f(w, i_t) = \frac{\lambda}{2} \parallel w \parallel^2 + \ell(w, (x_{i_t}, y_{i_t})) \tag{A.11}$$

Consideremos el sub-gradiente de la funcion objetivo, el cual esta dado por:

$$\nabla_t = \lambda w_t - \mathbb{I}[y_{i_t}\langle x_{i_t}, y_{i_t}\rangle < 1] y_{i_t} x_{i_t} \tag{A.12}$$

Donde $\mathbb{I}[y_{i_t}\langle w_t, x_{i_t}\rangle < 1]$ es la funcion indicatriz la cual toma un valor de 1 en donde el argumento es verdad y 0 si no. Luego se actualiza $w_{t+1} \longleftarrow w_t - \nu_t \nabla_t$ usando el paso $\nu_t = \frac{1}{\lambda * t}$. Note que esta actualizacion puede escribirse de la siguiente forma:

$$w_{t+1} \longrightarrow (1 - \frac{1}{t}) w_t + \nu_t \mathbb{I}[y_{i_t}\langle w_t, x_{i_t}\rangle < 1] y_{i_t} x_{i_t} \tag{A.13}$$

Despues de predeterminado numero iters iteraciones, se obtiene el $w_{iters+1}$ final junto con el sesgo $b_{iters+1}$.

Ahora intentaremos estudiar las propiedades y convergencias que se obtienen con el algoritmo diseñado. A lo largo de esta sub-seccion denotaremos

$$w^* = \text{argmin}_w \, f(w)$$

Para entender mejor las demostraciones, intentaremos focalizarnos en funciones instantaneas del estilo $f(w; A_t)$, la cual denotaria una funcion objetivo momentanea con la pendiente

actualizada en la iteracion t. Se iniciara el estudio acotando la funcion objetivo momentanea del algoritmo relativa a el promedio instantaneo de la solucion optima. Necesitaremos primero un lema el cual se basa en nociones de funciones fuertemente convexas[2]

Lema 1: Sea $f_1, \ldots, f_T$ sucesión de funciones λ-fuertemente convexas. Sea B un conjunto cerrado convexo y definamos $\Pi_B(w) = \text{argmin}_{w' \in B} \parallel w - w^{'} \parallel$. Sea $w_1, \ldots, w_{T+1}$ una sucesión de vectores tal que $w_1 \in B$ y para $t \geq 1$, $w_{t+1} = \Pi_B(w_t - \nu_t \nabla_t)$ donde ∇_t pertenece a el subgradiente del conjunto de f_t en w_t y $\nu_t = \frac{1}{\lambda t}$. Asuma que para todo t, $\parallel \nabla_t \parallel \leq G$. Entonces, para todo $u \in B$ se tiene que

$$\frac{1}{T}\sum_{t=1}^{T} f_t(w_t) \leq \frac{1}{T}\sum_{t=1}^{T} f_t(u) + \frac{G^2(1+\ln(T))}{2\lambda T}$$

Dem: La demostración se puede ver en la pagina 8 de [26].

Ya con esta cota obtenida gracias al lema anterior podemos acotar el promedio instantaneo de la funcion objetivo.

Teorema 2: Asuma que para todo $(x, y) \in S$ la norma de x es a lo mas R. Sea w^* la solucion de nuestro problema tal como se dijo en un inicio y sea $c = (2R)^2$. Entonces para $T \leq 3$,

$$\frac{1}{T}\sum_{t=1}^{T} f(w_t; A_t) \leq \frac{1}{T}\sum_{t=1}^{T} f(w^*; A_t) + \frac{c(1+\ln(T))}{2\lambda T}$$

Dem: Para simplificar nuestra notacion utilizaremos la siguiente abreviacion $f_t(w) = f(w, A_t)$. La actualizacion de nuestro algoritmo puede ser esrita como $w_{t+1} = \Pi_B(w_t - \nu_t \nabla_t)$, donde ∇_t es el gradiente tal como se definio en la ecuacion (6.12) y $B = \mathbb{R}^n$. Por lo tanto, para probar el teorema es suficiente mostrar que la condicion del Lema 1 se tiene. Ya que f_t es la suma de funciones λ- fuertemente convexas ($\frac{\lambda}{2} \parallel w \parallel^2$) y una funcion convexa (promedio de nuestra funcion de perdida lineal sobre A_t), es claro que es λ-fuetemente convexa. Seguiremos con una cota para $\parallel \nabla_t \parallel$. Reescribiremos nuestro paso de actualizacion como:

$$w_{t+1} = (1 - \frac{1}{t})w_t - \frac{1}{\lambda t} v_t, \tag{A.14}$$

donde $v_t = \frac{1}{|A_t|}\sum_{i \in A_i} \mathbb{I}[y_i \langle w_t, x_t \rangle < 1] y_i x_i$. Por lo tanto el peso inicial para cada v_i es $\frac{1}{\lambda t}$ y luego para el resto de los j, se multiplicara por $(\frac{j-1}{j})$.Por lo tanto , el peso total de v_i en w_{t+1} es :

$$\frac{1}{\lambda t}\Pi_{j=i+1}^{t} \frac{j-1}{j} = \frac{1}{\lambda t},$$

lo cual implica que w_{t+1} puede ser reescrito como:

$$w_{t+1} = \frac{1}{\lambda t}\sum_{i=1}^{t} v_i \tag{A.15}$$

[2]Una funcion f es llamada λ-fuertemente convexa si $f(w) - \frac{\lambda}{2} \parallel w \parallel^2$ es una funcion convexa.

De lo anterior , inmediatamente es posible obtener que $\| w_{t+1} \| \leq \frac{R}{\lambda}$ y por lo tanto $\| \nabla_t \| \leq 2R$ con $w^* \in \mathbb{R}^n$. Luego la cota se obtiene siguiendo el Lema 1. □

Pasamos ahora a la obtención de un límite en el objetivo general $f(w_t)$ evaluada en un predictor w_t. La convexidad de f implica que:

$$f(\frac{1}{T}\sum_{t=1}^{T} w_t) \leq \frac{1}{T}\sum_{t=1}^{T} f(w_t) \tag{A.16}$$

Utilizando la desilguadad anterior y el teorema anterior, inmediatamente obtenemos el corolario 1, el cual proporciona un análisis de convergencia para el caso determinista cuando $k = m$ donde $f(w, A_t) = f(w)$.

Corolario 1: Asuma que se tienen las condiciones del Teorema anterior y que $A_t = S$ para todo t. Sea $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. Entonces

$$f(\bar{w}) \leq f(w^*) + \frac{c(1 + \ln(T))}{2\lambda T}$$

Cuando $A_t \subset S$, el Corolario 1 no se tiene. Sin embargo, Kakade y Tewari en [19] desmotraron que cotas similares se obtienen cuando A_t es una muestra o subconjunto de S

Lema 2 (Corolario en [19]) Asuma que las condiciones del teorema se tienen y que para todo t, cada elemento en A_t es toma de una forma aleatoria uniformemente de S (con o sin repeticion). Asuma tambien que $R \geq 1$ y que $\lambda \leq \frac{1}{4}$. Entonces con una probabilidad de al menos $1 - \delta$ se tiene:

$$\frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*) \leq \frac{21c\ln(\frac{T}{\delta})}{\lambda T}$$

Combinando el Lema 2 junto con la convexidad de la funcion, inmediatamente obtenemos el siguiente corolario.

Corolario 2 Asuma que las condiciones del Lema 2 se cumplen y sea $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. Entonces, con probabilidad de al menos $1 - \delta$ se tiene

$$f(\bar{w}) \leq f(w^*) + \frac{21c\ln(\frac{T}{\delta})}{\lambda T}$$

En los corolarios previos son validos para $\bar{w}$. En la practica la hipotesis final, w_{T+1}, suele entregar mejores resultados. Ahora disminuiremos esta brecha proporcionando una tasa de convergencia similar para un mecanismo diferente de elección del vector de salida. Para hacerlo, primero mostramos que al menos la mitad de las hipótesis son buenas.

Lema 3: Asuma que las condiciones del lema 2 se cumplen. Entonces, si t es un numero aleatoreamente seleccionado de $K = \{1, \ldots, T\}$, tenemos que con al menos una probabilidad de $\frac{1}{2}$ que

$$f(w_t) \leq f(w^*) + \frac{42c\ln(\frac{T}{\delta})}{\lambda T}$$

Dem: la demostracion se puede ver en la pagina 11 de [26].

Bibliografía

[1] Shigeo Abe. *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[2] Arvinder Pal Singh Bali, Mexson Fernandes, Sourabh Choubey, and Mahima Goel. Comparative performance of machine learning algorithms for fake news detection. In Mayank Singh, P.K. Gupta, Vipin Tyagi, Jan Flusser, Tuncer Ören, and Rekha Kashyap, editors, *Advances in Computing and Data Sciences*, pages 420–430, Singapore, 2019. Springer Singapore.

[3] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[4] Candice Bentéjac, Anna Csörgo, and Gonzalo Martínez-Muñoz. A comparative analysis of xgboost. *CoRR*, abs/1911.01914, 2019.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[6] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] Joseph Andreas Chakraborty, Chiranjit. Machine learning at central banks. Bank of England working papers 674, Bank of England, September 2017.

[9] Tianqi Chen and Carlos Guestrin. Xgboost. In *XGBoost: A Scalable Tree Boosting System*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[10] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[11] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

[12] Roger Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, NY, USA, second edition, 1987.

[13] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2nd edition, 2017.

[14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Random Forests*, pages 587–604. Springer New York, New York, NY, 2009.

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[16] Jason Brownlee. https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/, 17 de Agosto de 2016.

[17] Y. Singer. K. Crammer. On the learnability and design of output codes for multiclass problems. *COLT*, 2000.

[18] Y. Singer. K. Crammer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2001.

[19] Sham M Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009.

[20] C.J.Lin S.S.Keerthi y S.Sundararajan KC.J.Hsieh, K.W.Chang. A dual coordinate descent method for large-scale linear svm. In *ICLM*, 2008.

[21] S. Sathiya Keerthi, S. Sundararajan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A sequential dual method for large scale multi-class linear svms. In *KDD*, 2008.

[22] Aishwarya Mujumdar and V Vaidehi. Diabetes prediction using machine learning algorithms. *Procedia Computer Science*, 165:292–299, 2019. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.

[23] Rashmi Murty, M. N. y Raghava. *Linear Support Vector Machines*, pages 41–56. Springer International Publishing, Cham, 2016.

[24] Matthew Norton and Johannes O. Royset. Diametrical risk minimization: Theory and computations, 2021.

[25] Omar Ibrahim Obaid, Mazin Abed Mohammed, Mohd Khanapi Abd Ghani, Salama A. Mostafa, and Fahad Taha AL-Dhief. Evaluating the performance of machine learning techniques in the classification of wisconsin breast cancer. *International Journal of Engineering Technology*, 7(4.36):160–166, 2018.

[26] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm.

[27] James C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley Sons, Inc., USA, 1 edition, 2003.

[28] Jon A. van der Vaart, Aad W. y Wellner. *M-Estimators*, pages 284–308. Springer New York, New York, NY, 1996.

[29] V. Vapnik. Principles of risk minimization for learning theory. In J. Moody, S. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1992.

[30] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2nd edition, 2000.

[31] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent

algorithms. In *ICML 2004: Proceedings of the twenty-first international conference on machine learning. Omnipress*, pages 919–926, 2004.

[32] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms.*, volume 1. Chapman and Hall/CRC, 2012.