Entrega primera evaluación del curso aprendizaje automático

Liceth Cristina Mosquera Galvis

Maestría en Ciencias de los Datos y Analítica

Código: 201910046228

## Email: lcmosquerg@eafit.edu.co

31 de agosto de 2019

Descripción de los datos:

La base de datos con la que se va a trabajar fue obtenida de Bloomberg y son indicadores del mercado en resolución diaria desde el 01-ene-2000 a 01-ene-2019. La variable independiente y sobre la que se van a entrenar los modelos se llama “Class” es una variable categórica que toma 3 posibles valores -1 si la decisión debe ser vender alguna posición debido a que el mercado tiende a perder valor, 0 si la mejor decisión es esperar y no hacer movimientos y 1 si la decisión debe ser comprar.

Son 4,075 datos y se distribuyen así:

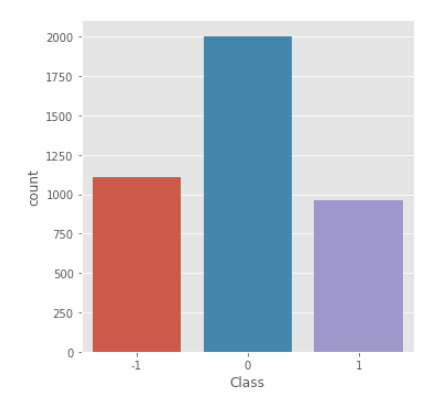


Ilustración 1 Distribución de variable independiente en los diferentes estados.

Por lo tanto, este será un problema de clasificación multiclases, tres para ser exactos. El conjunto de entrenamiento S consiste en puntos extraídos independiente al azar de D (distribución de probabilidad) sobre el espacio X y el objetivo es predecir bien los nuevos puntos que también son extraídos de D.

1. Error real deseado

es un subconjunto de X que corresponde a la clase positiva para una clasificación binaria deseada, en ingles *target concept*. Cada punto del conjunto de entrenamiento S está etiquetado de acuerdo con si pertenece o no a . El objetivo es producir un conjunto , llamado hipótesis, que esté cerca de con respecto a la distribución D.

Donde 𝚫 denota una diferencia simétrica y la masa de probabilidad es acorde a D.

El error verdadero de h es la probabilidad de que clasifique incorrectamente un punto de datos extraído al azar de D. El objetivo es producir una h de error verdadero bajo.

El error deseado es de **0.2**, lo ideal es tener un error menor a 0.5 para que el modelo sea mejor el tomar una decisión al azar.

1. Error de entrenamiento deseado

El error de entrenamiento deseado, lo vamos a iniciar en **0.1**, suponiendo que mi error de entrenamiento inicial va a ser mejor que una decisión al azar entre no hacer nada y tomar alguna decisión de vender o comprar.

El error de entrenamiento de h, denotado , es la fracción de puntos en S en los que h yno están de acuerdo. Es decir,

Hay que tener en cuenta que, aunque se supone que S consiste en puntos extraídos al azar de D, es posible que una hipótesis h tenga un error de entrenamiento bajo o incluso que esté completamente de acuerdo con sobre la muestra de entrenamiento, y sin embargo tenga un error verdadero alto.

1. Garantía probable de aprendizaje

La clase de hipótesis 𝓗 sobre X es una es una colección de subconjuntos de X, llamados hipótesis o conceptos. El objetivo es encontrar una hipótesis en 𝓗 que coincida estrechamente consobre S.

Si el conjunto de entrenamiento S es lo suficientemente grande comparado con algunas propiedades de 𝓗, entonces hay una alta probabilidad que todos los h ∈ 𝓗, tengan su error de entrenamiento cercano al error real. Si encontramos una hipótesis cuyo error de entrenamiento sea bajo, podemos estar seguros de que su error real también será bajo.

Para un conjunto de entrenamiento S suficientemente grande, cada h ∈ 𝓗 tiene una alta probabilidad que el error de entrenamiento esté dentro del error verdadero. Tal enunciado se llama convergencia uniforme porque estamos pidiendo que los errores del conjunto de entrenamiento converjan a sus errores verdaderos de manera uniforme en todos los conjuntos en 𝓗.

Sea 𝓗 una clase de hipótesis y mayores que cero, si el conjunto de entrenamiento S de tamaño

es extraído de una distribución D, entonces con una probabilidad más grande o igual a 1- δ, cada h en 𝓗 satisface .

Bajo este contexto para nuestros datos tenemos 68 variables y una de estas es la que deseamos modelar por lo tanto la dimensión VC que es la dimensión de nuestros datos más 1 será 68.

Modelo lineal y SMV:

Con 𝓗=2k, donde k es el numero de regresores que para nuestro caso es 68

PAC para regresión lineal y SVM lineal:

PAC para Árbol. Donde h es profundidad por ramas

1. Tamaño óptimo de la muestra

El problema de cuán grande se requiere una muestra surge en muchas aplicaciones además de la teoría de aprendizaje. Una de esas aplicaciones es el tamaño de una muestra de un conjunto de datos que necesitamos para asegurarnos que con esa muestra podemos responder preguntas con alta probabilidad que la respuesta sea confiable. La respuesta se basa en la complejidad de la clase de preguntas que en cierto sentido corresponde a lo sofisticado que es el algoritmo de aprendizaje. Para introducir el concepto de dimensión VC, consideraremos muestrear una base de datos en lugar de entrenar una red.

La dimensión VC para dividir espacios en 2 mediante un separador lineal en d dimensiones es d+1. Hay un conjunto de tamaño d+1 que puede ser dividido por espacios medios.

Es el número máximo de subconjuntos de cualquier conjunto A de tamaño n que puede expresarse como A∩h para h en 𝓗. La función es igual a 2n para n menores que o iguales que la dimensión VC de 𝓗 y d infinito y para n mayor que la dimensión VC crece polinómicamente con n, con el grado polinomial igual a la dimensión VC.

La probabilidad que exista un h ∈ 𝓗 con diferencias entre el error real y el error empírico es más grande que es menor o igual que .

Con probabilidad mayor o igual a 1-δ, cada h ∈ 𝓗 satisface

Se calcula el tamaño óptimo de la muestra para cada uno de los algoritmos a utilizar.

* Regresión lineal y SMV lineal
* Árbol de decisión

Los parámetros para el cálculo del tamaño optimo de la muestra de un árbol de decisión se calcula con los parámetros de profundidad, que para este caso vamos a tomar k=3, m que es el número de características que en este caso son 67, que es 0.2 y que es 0.1.

* Máquina de soporte vectorial

Para un espacio de entrada n-dimensional, la dimensión VC de la función indicadora orientada al hiperplano es igual a h=m+1, por lo tanto, se puede manejar la misma que para la regresión lineal. Los hiper parámetros son el kernel, el parámetro Gamma y el parámetro C, como el gamma y C dependen del kernel, entonces con considerar este se está modificando los demás.

* Random forest: por la cantidad de hiperparámetros que puede manejar este algoritmo la dimensión VC del mismo es infinita y por lo tanto no se tiene claridad sobre el tamaño óptimo de la muestra.

1. Dividir el conjunto de muestra en: entrenamiento, validación y prueba.

Se dividen los datos según lo recomendado 60% training, 20% test y 20% validation. El total de datos luego de sacar los que les falta información son 4,075 datos.

1. Visualizar los datos con los algoritmos de visualización por defecto de la suite, extraer características si hace falta para aumentar la dimensionalidad

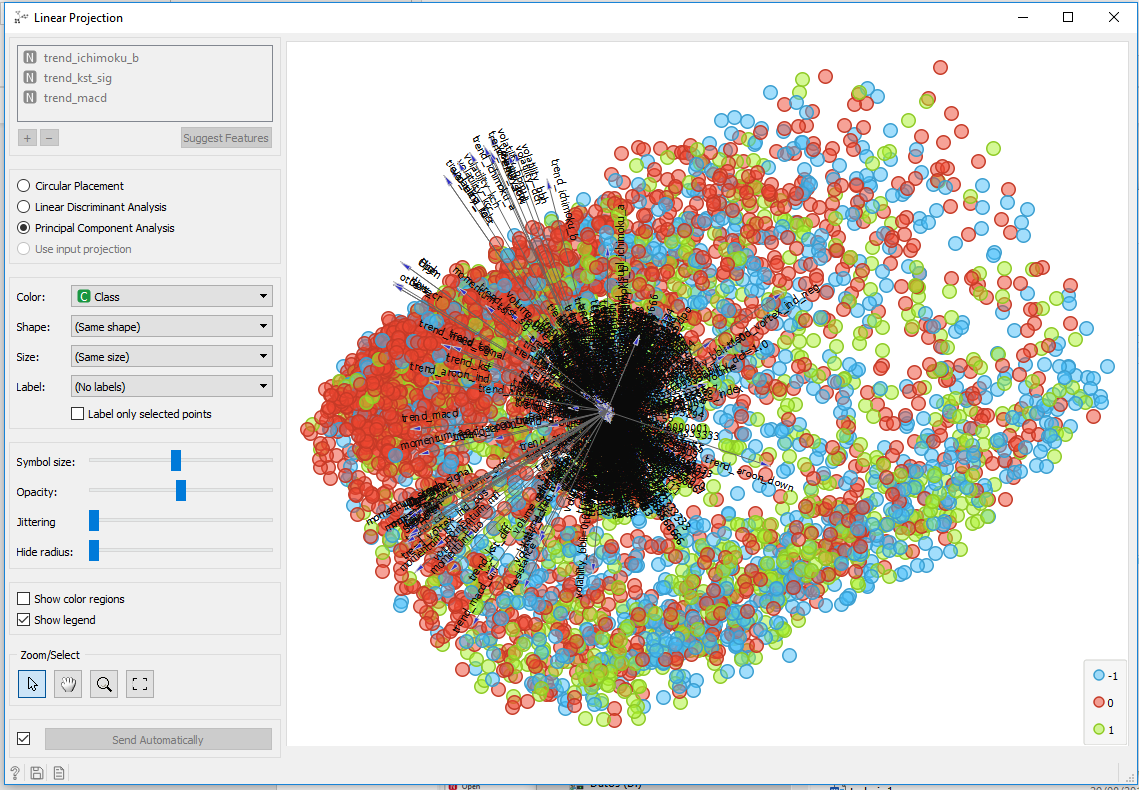


Ilustración Proyección de variables desde Orange por Análisis de componentes principales

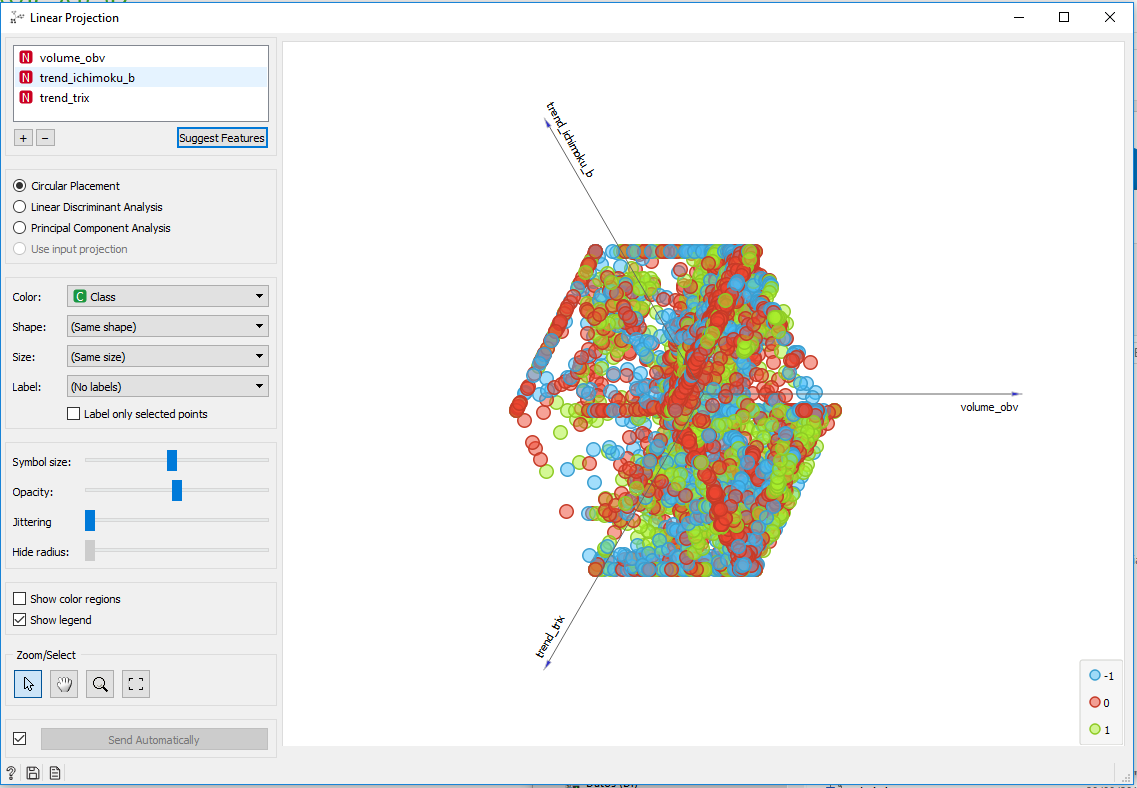


Ilustración Proyección de variables desde Orange por Placement circular

En los datos estudiados estas técnicas no mejoran la visualización.

1. Visualizar los datos con el embebimiento BH tsne <https://lvdmaaten.github.io/tsne/>

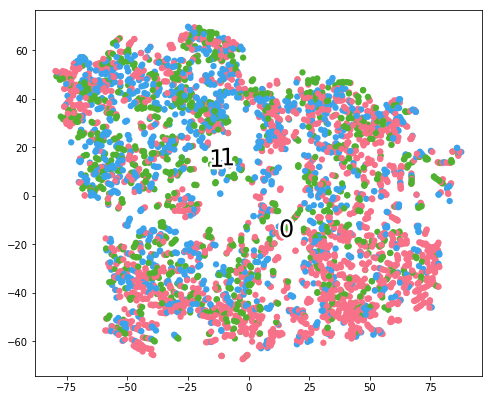


Ilustración Reducción de dimensionalidad con tsne

Aunque se redujo la dimensionalidad, la visualización no mejora y no se notan separaciones claras entre las clases de datos.

1. Entrenar el modelo con los datos en altas dimensiones (espacio original)
   1. Resultados desde Orange.

El mejor modelo según los resultados de Orange sería una random forest sin embargo son modelo muy regulares y dada la complejidad es mejor la regresión logística.

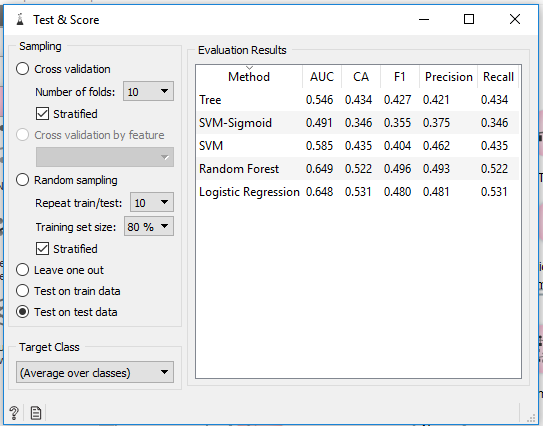


Ilustración Resultados extraídos desde Orange para los diferentes modelos.

Roc de la clase -1:

Imagen que contiene texto, mapa

Descripción generada automáticamenteImagen que contiene texto, mapa

Descripción generada automáticamenteImagen que contiene texto, mapa

Descripción generada automáticamente

Ilustración ROC para cada una de las clases. -1, 0 y 1 respectivamente

* 1. Box plot de todos los modelos

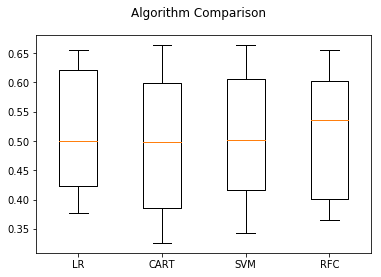
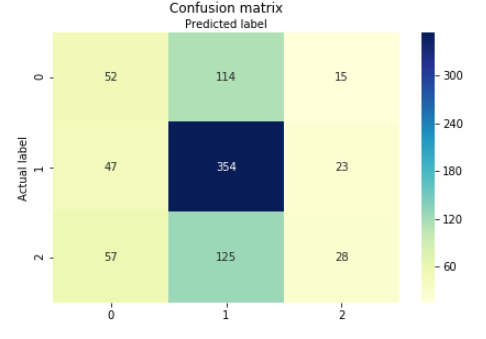
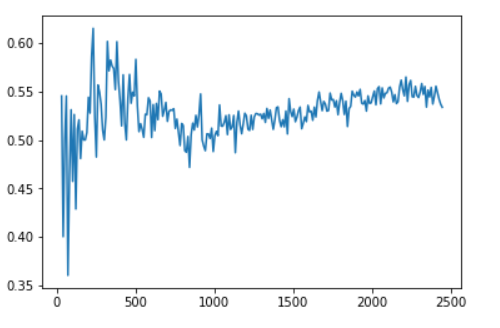
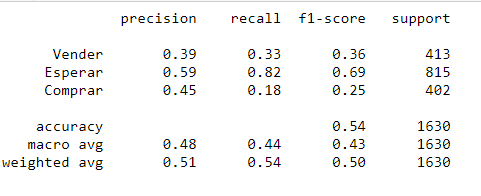
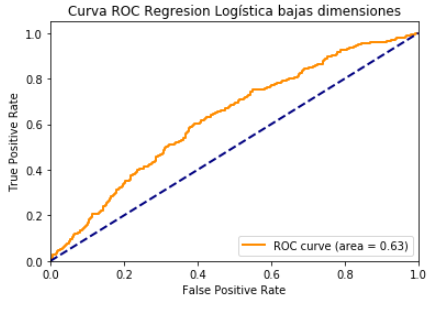
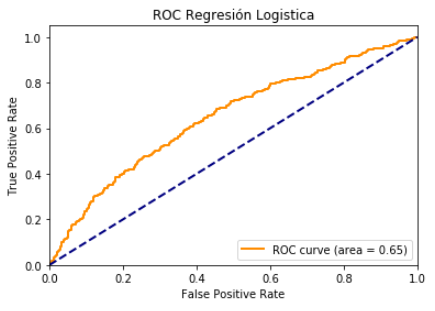


Ilustración Box plot de los modelos Regresión logística, árbol, SMV y Random Forest

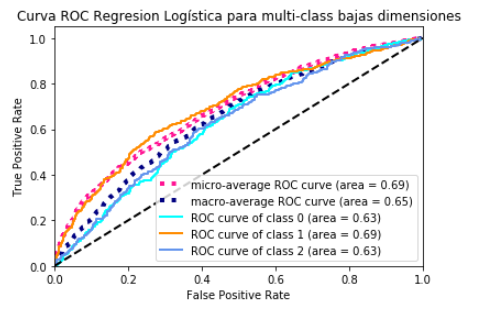
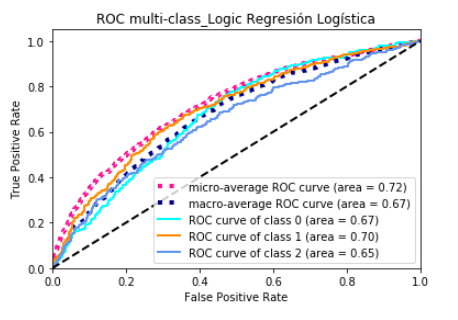
* 1. Resultados Regresión Logística normal y en bajas dimensiones







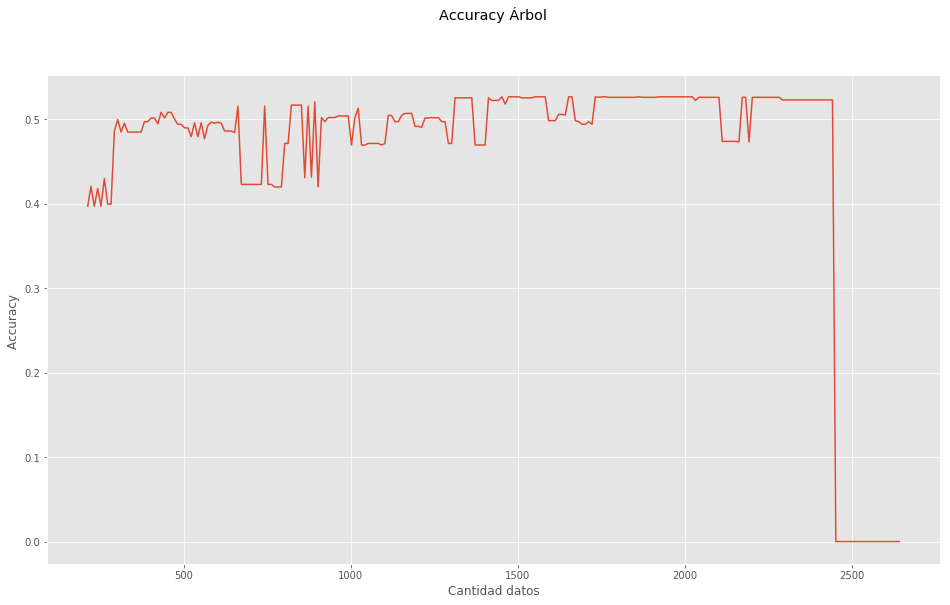
El ROC desmejoró al bajar las dimensiones.

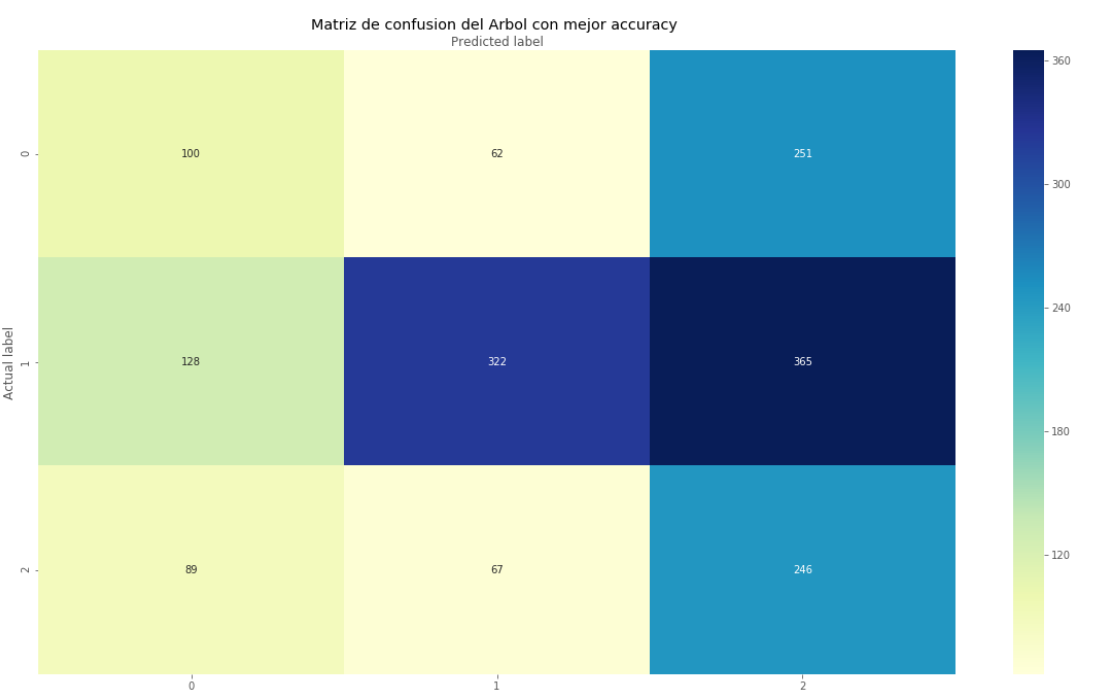


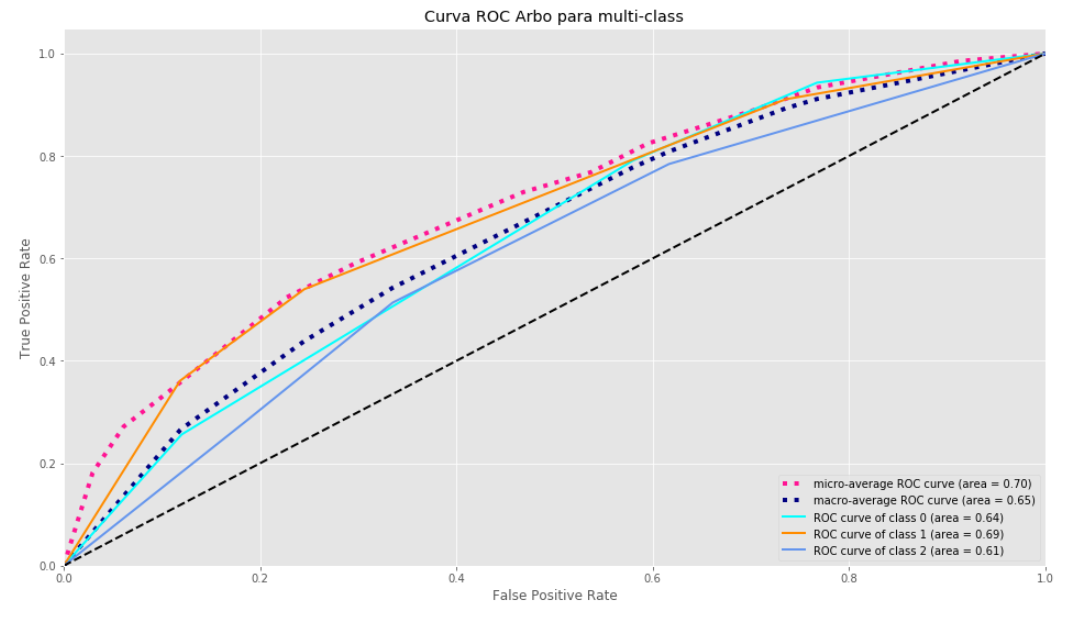
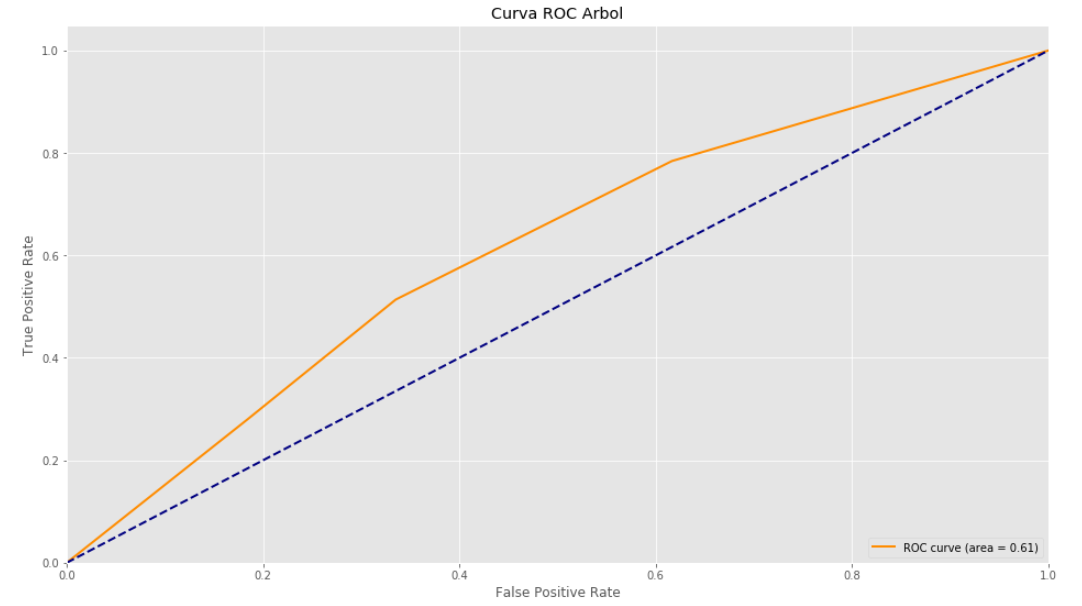
* 1. Resultados Árbol de decisión normal y en bajas dimensiones

La cantidad de datos son 4075, como se debe ir aumentando la muestra para el entrenamiento del modelo máximo esta puede llegar máximo a 2,852, que representa alrededor del 70% de los datos, de ahí en adelante no hay forma de seguir entrenando. El mejor accuracy está es para un tamaño de muestra de 2,227, llegando a 0.536 de ahí el modelo empieza a desmejorar.

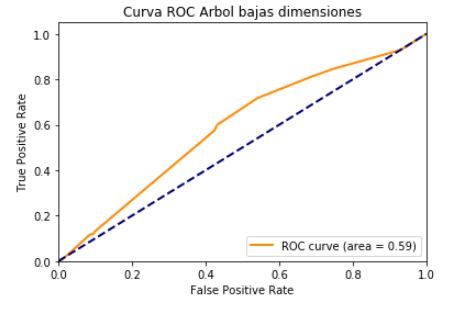
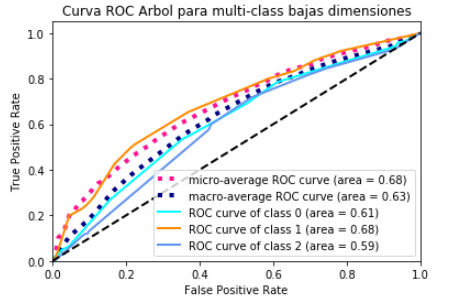
El mejor modelo tiene una profundidad de 2 y una cantidad de hojas 1.





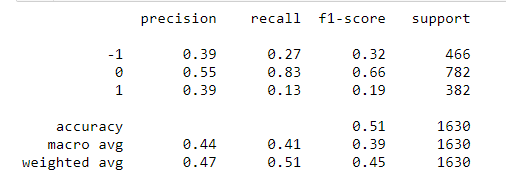


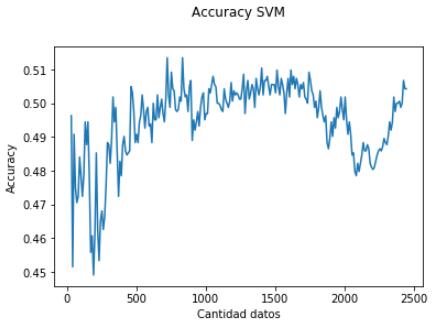
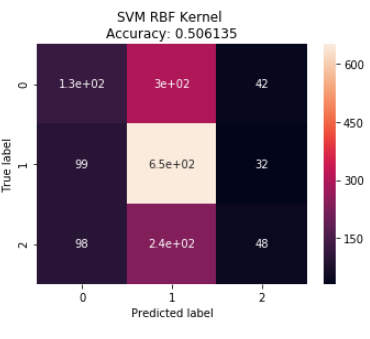
Para el modelo escogido el mejor accuracy se da con solo 1452 datos.

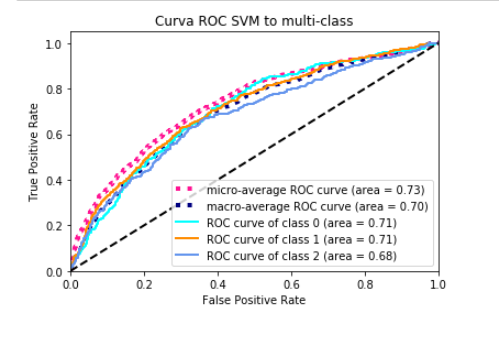
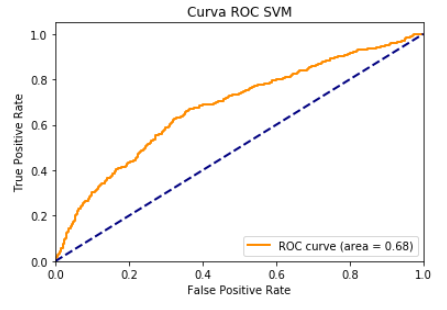
Los resultados no mejoran con la reducción de dimensiones.

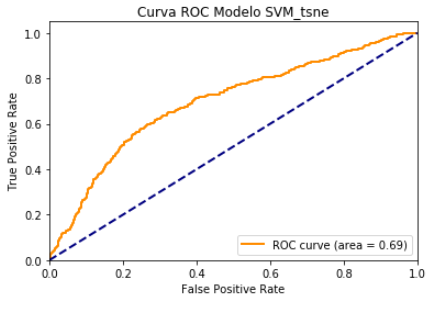
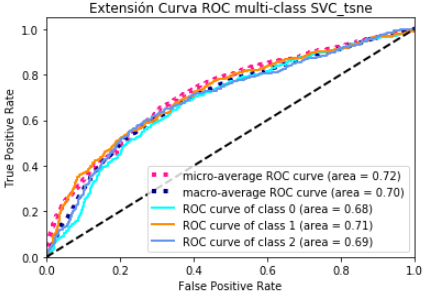
* 1. Resultados Máquina de Soporte Vectorial normal y en bajas dimensiones





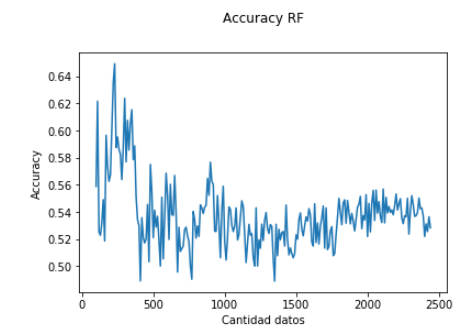
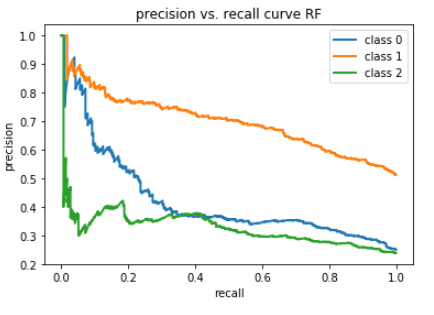
El tamaño de los datos para entrenar con el que se obtiene el mejor modelo es 723



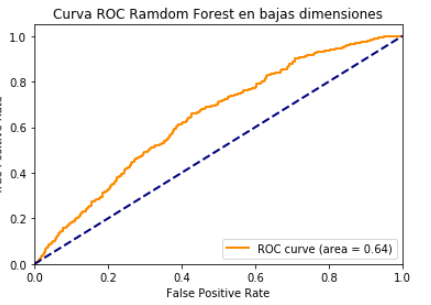
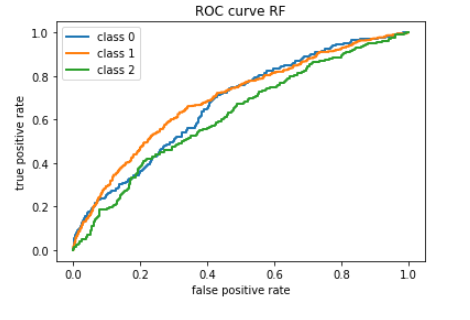
 

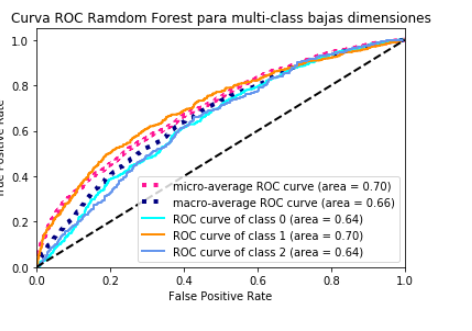
No hay mejoras considerables al aplicar reducción de dimensionalidad.

* 1. Resultados Random Forest normal y en bajas dimensiones

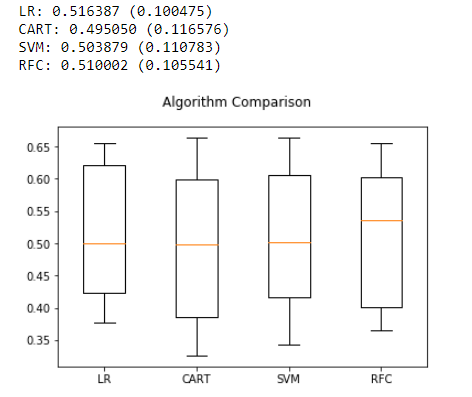
El modelo optimo se obtiene con una muestra de 230 datos.





1. Entrenar el modelo con los datos en dimensiones reducidas (luego del embebimiento del paso

7)



1. Realizar la comparación del modelo del paso 8 contra el modelo del paso

La evaluación de los modelos Tanto con todas las dimensiones como en dimensiones reducidas no obtiene modelos con alto