# Práctica no. 7: Máquinas de Soporte Vectorial


Jair Antonio Bautista Loranca
a01365850@itesm.mx
Tecnológico de Monterrey
Monterrey, N.L., México

Maximiliano Zambada Camacho
a01570146@itesm.mx
Tecnológico de Monterrey
Monterrey, N.L., México



## RESUMEN

Al estar avanzando cada vez más dentro de los modelos de aprendizaje supervisado, nos hemos encontrado con muchas maneras de analizar y clasificar datos, unas más efectivas que otras. Uno de estos métodos es el de las máquinas de soporte vectorial, el cual es uno de los mejores modelos de clasificación en el área de aprendizaje automático.



**ACM Reference Format:**

Jair Antonio Bautista Loranca and Maximiliano Zambada Camacho. 2021. Práctica no. 7: Máquinas de Soporte Vectorial. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn


## 1. INTRODUCCIÓN

Para seguir profundizando en este ámbito de modelos de aprendizaje, continuamos con un modelo que es considerado como uno de los más efectivos en la clasificación de datos, las Máquinas de Soporte Vectorial. Se puede decir que desde un inicio es un método de clasificación binaria, pero su aplicación se ha extendido a problemas de clasificación múltiple y regresión.

## 2. CONCEPTOS PREVIOS

Las Máquinas de Soporte Vectorial se basan en el Maximal Margin Classifier, que al mismo tiempo, se basa en el concepto de hiperplano. El hiperplano es un subespacio plano con dimensiones $p - 1$, tomando en cuenta un espacio $p$-dimensional. Siguiendo el mismo patrón, en un espacio de dos dimensiones, el hiperplano es un subespacio de 1 dimensión, o sea una recta, y en un espacio tridimensional, es dos dimensiones, un plano normal. Para dimensiones mayores a 3 no es útil visualizar uno, pero el concepto de p1 dimensiones se mantiene.

. La definición matemática de un hiperplano no tiene mucha ciencia. Su longitud depende de las dimensiones a utilizar, pero en esencia, el hiperplano se describe como la ecuación de una recta:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p = 0$$

. Dados los parámetros $\beta_0$, $\beta_1$ y $\beta_2$, todos los pares de valores $x = (x_1, x_2, ..., x_p)$ para los que se cumple la igualdad son puntos del hiperplano.




*Conference'17, July 2017, Washington, DC, USA*





ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn


. Cuando $x$ no satisface la ecuación:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p < 0$$

. o bien,

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p > 0$$

. el punto $x$ cae a un lado o al otro del hiperplano. Entonces se puede ver que un hiperplano divide un espacio $p$-dimensional en dos mitades. Para saber en qué lado del hiperplano se encuentra un determinado punto $x$, solo hay que calcular el signo de la ecuación.

. En la imagen se muestra el hiperplano de un espacio bidimensional. La ecuación que describe el hiperplano es $1 + 2x_1 + 3x_2 = 0$. La región azul representa el espacio en el que se encuentran todos los puntos para los que $1 + 2x_1 + 3x_2 > 0$ y la región roja el de los puntos para los que $1 + 2x_1 + 3x_2 < 0$.

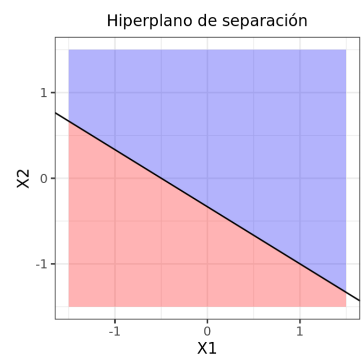



**Figura 1: Hiperplano de Separación `Ejemplo`**

. Si con la distribución de las observaciones se pueden separar linealmente de forma perfecta en las dos clases (+1 y $-1$), entonces, un hiperplano de separación cumple que:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p < 0$$

. si $y = -1$, y

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p > 0$$

. si $y = 1$

. Entonces, al identificar cada clase como $+1$ o $-1$, y dado que multiplicar dos valores negativos resultan en un valor positivo, las dos condiciones anteriores pueden simplificarse en una:

$$y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p) > 0$$

. para $i = 1...n$

## 3. METODOLOGÍA

. Durante el desarrollo de la práctica decidimos programar en parejas, de manera que desarrollamos la mayor parte del script entre los dos. Sin embargo decidimos dividir el trabajo para la generación de gráficas y adelantar el reporte.

. Debido a que los clasificadores de SKLearn tienen casi las mismas funciones, esto nos permitió trabajar de manera genérica para la creación de modelos y cálculos de precisiones. Sin embargo al momento de generar la comparación en el espacio ROC, fue necesario tener algunas condiciones para tratar algunos modelos diferentes. Aún así estos ajustes son mínimos y nos permite tener un script bastante extensible si es que en el futuro se requiere comparar otros modelos.

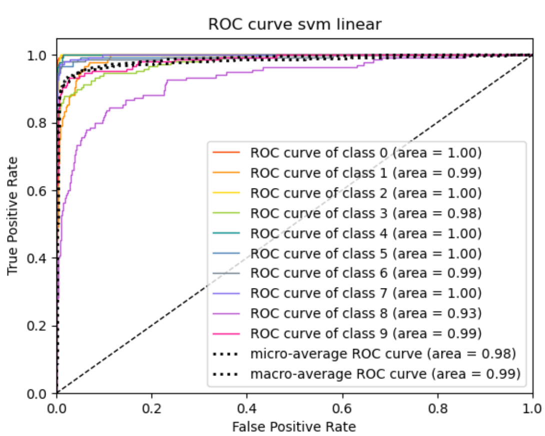
## 4. RESULTADOS

A continuación se muestran los resultados obtenidos por cada modelo. Principalmente nos enforcaremos en discutir sobre las gráficas en el espacio ROC, ya que estás nos dan mucho mejor entendimiento de cómo es el performance del clasificador en comparación de la precisión.

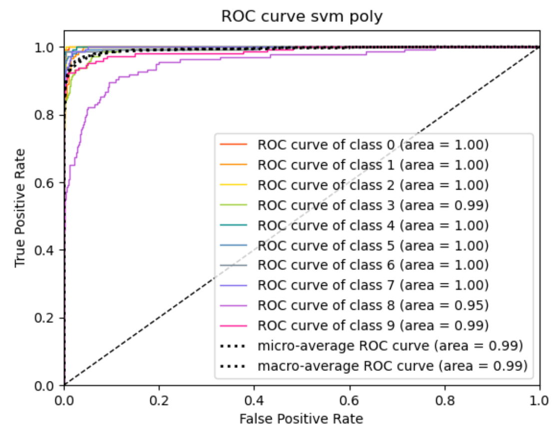
| model | precision |
|---|---|
| svm linear | 83.38% |
| svm poly | 87.62% |
| svm rbf | 0.00% |
| svm sigmoid | 71.91% |
| log reg | 84.63% |
| knn | 96.66% |
| bayes | 14.19% |

**Figura 2: Precisiones por modelo generado**

. La precisión de cada modelo generado parece bastante normal, aunque destaca que el modelo de *k-Nearest Neighbors* es el que tiene mejor calificación con esta métrica. Sin embargo el problema es que esta métrica suele ser demasiado dura y como está reducida a un sólo valor, no proporciona suficiente información.

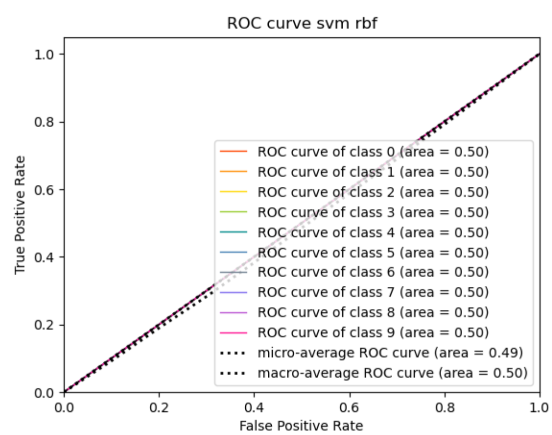



. Podemos ver que la SVM con kernel lineal tiene un área ROC macro de 0.99, esto nos indica que en general en un muy buen clasificador. Sin embargo le cuesta un poco de trabajo clasificar instancias que correspondientes al dígito 8. Esto resulta esperado ya que el 8 se puede sobreponer en la forma de los otros dígitos.

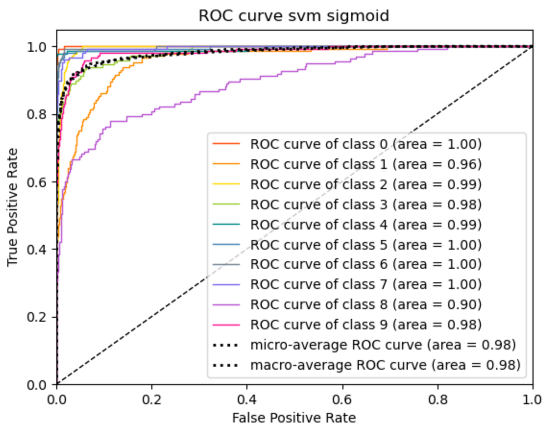



. Podemos ver que la SVM con kernel polinomial tiene un área ROC macro de 0.99, esto nos indica que en general en un muy buen clasificador. A diferencia de la SVM lineal, presenta una ligera ventaja cuando se trata de clasificar instancias correspondientes al dígito 8, así como también al clasificar otras clases.

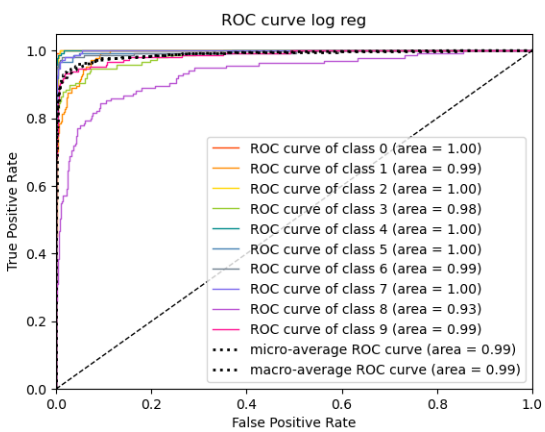
. Considerando esto y además de que tiene mayor precisión que la SVM lineal nos puede indicar que es una mejora, sin embargo tenemos nuestras dudas ya que puede ser un caso de overfiting. Aún hace falta ajustar más los parámetros del kernel para determinar si es el caso o no.




. Podemos ver que para la SVM con base radial tiene un área ROC maco de 0.5, lo cual nos indica que el clasificador realmente no está dando resultados correctos o mejor dicho no es diferente a un clasificador aleatoreo.

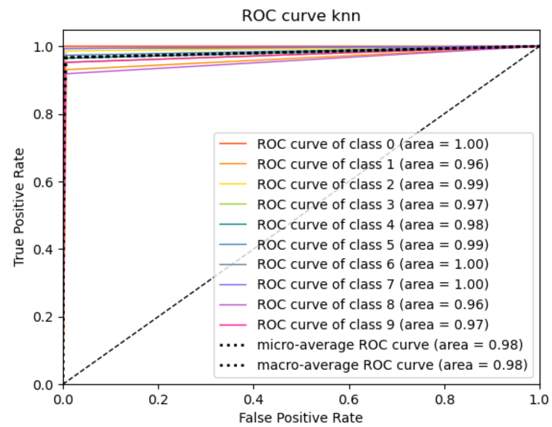



. Podemos ver que para la regresión logística tiene un área ROC macro de 0.99, esto nos indica que en general en un muy buen clasificador. Aunque de manera similar con todos clasificadores, el dígito 8 le cuesta mucho trabajo.



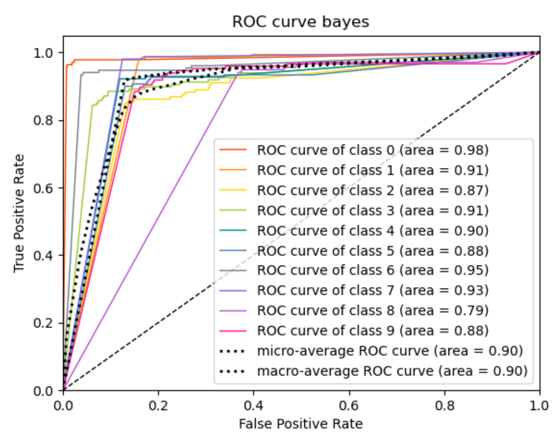

. Podemos ver que para la regresión logística tiene un área ROC macro de 0.99 esto nos indica que en general en un muy buen clasificador. Sin embargo le cuesta un poco de trabajo clasificar instancias que correspondientes al dígito 8.

. Esto es bastante curioso ya que se puede decir, de manera empírica, que la SVM lineal y la regresión lineal funcionan de manera similar, ya que ambas generan de alguna manera una frontera lineal en el dataset. Por lo que ver casi las mismas métricas para ambas no es de extrañar.




. Podemos ver que para la k-NN tiene un área ROC macro de 0.98, aunque es ligeremente más bajo que otros que hemos visto, este es el que tiene mejor precisión sobre los demás.

. Si observamos las curvas en el espacio ROC, parece que tiende a clasificar de manera positiva con mayor tendencia que otros clasificadores. Esto es lo que ocasiona que tenga más precisión que los otros modelos, ya que al tratarse de dígitos escritos a mano, hay una tendencia a que los datos se agrupen por lo que comparar con otras instacias cercanas resulta en una estrategia buena para clasificar en este dataset.




. Podemos ver que para la Bayes Ingenuo tiene un área ROC macro de 0.90, sin embargo considerando las curvas ROC por clase y su precisión, es el segundo peor clasificador.

. Esto es probablemente a que el modelo asume independencia en los atributos de una instancia. Sin embargo al tratarser de dígitos escritos a mano es claro que no son independientes debido a que cada dígito es un conjunto de trazos, entonces que ciertos pixeles adyacentes tengan ciertos valores no es por al azar ya que la mayor parte de las veces puede pertenecer a un trazo concereto.

## 5. CONCLUSIONES Y REFLEXIONES

Como podemos ver, casi todos los modelos de clasificación tienen un porcentaje de precisión alto, sin embargo, eso no determina cuál sea el mejor para utilizar. Así como se puede ver, la mayoría de las regresiones tienen un área macro de .99, arriba del área de kNN, sin embargo, el kNN tiene mayor precisión, y por ende mejor clasificación de valores. Sin embargo, como se mencionó, como está reducida a un sólo valor, no proporciona suficiente información. Entonces los modelos con buena precisión funcionan como clasificadores para los datos, sin embargo, se tienen que tomar en cuenta aspectos como los mencionados, para saber en qué te afecta o beneficia cada uno.

*Jair Antonio.* A partir de los resultados obtenidos puedo observar la utilidad y performance que tienen los modelos de Machine Learning. Considero que esta práctica ha sido de gran utilidad, no sólo para entender las máquinas de soporte vectorial sino también cómo analizar diferentes modelos de manera objetiva utilzando el espacio ROC y sus propiedades.

*Maximiliano Zambada.* Con esta práctica, pude aún más profundizar mis conocimientos de los modelos de aprendizaje y clasificación de datos. Asimismo, el ejercicio de utilizar todos los modelos vistos hasta el momento en un mismo dataset, y compararlos de acuerdo a su precisión, me dio un mejor entendimiento en su aplicación y cómo se puede aprovechar cada uno.