APLICACIÓN DE SUPPORT VECTOR MACHINE AL MERCADO COLOMBIANO

Trabajo de grado

Daniel Santiago Castañeda

Maestría en finanzas cuantitativas- Universidad del Rosario

Directores: Rafael Serrano – Hugo Ramirez

Julio 2019

Tabla de contenido

1. Introducción

Durante las últimas décadas y gracias al desarrollo de la tecnología en los informatica, los algoritmos de machine learning han evolucionado lo que ha permitido en la mayoría de los casos obtener mejores resultados que los algoritmos tradicionales.

Recientemente se ha empezado a estudiar la implementación de *machine learning* a la hora de analizar datos altamente volátiles y complejos como lo son aquellos del sector financiero. Algoritmos como Redes Neuronales, Support vector Machine (SVM) y análisis de sentimientos han sido estudiadas con el objetivo específico de obtener mejores estimaciones y encontrar tendencias en los mercados que se traduzcan en estrategias de trading más robustas y confiables.

El presente documento busca pronosticar los movimientos de algunos activos del mercado colombiano usando Support vector machine (SVM).

Es importante resaltar:

- Los algoritmos de machine learning son algoritmos muy poco explorados a nivel Colombia, en otras áreas de investigación como por ejemplo la industria bancaria han demostrado generalmente tener un mejor desempeño que los modelos tradicionales.
- Es un tema novedoso a nivel mundial poco explorado por el alto nivel computacional necesario
- Los algoritmos de machine learning permiten encontrar relaciones entre variables que escapan a la experticia del investigador
- Este ejercicio se ha realizado en países asiáticos con buenos resultados

Este documento comenzará con una mirada al estado del arte. Posteriormente se realizará el desarrollo del marco teórico en el que se detalla el algoritmo SVM, este marco finaliza con un ejemplo que permite ilustrar el funcionamiento del algoritmo. Por último, se realizará una aplicación con datos del mercado colombiano y por último una propuesta de estrategia de trading basada en SVM.

2. Objetivo:

General:

- Generar y evaluar los pronósticos generados con SVM de la serie COLCAP y generar una estrategia de trading

Específicos:

- Generar una herramienta de machine learning que obtenga un alto índice de acierto en los pronósticos de los movimientos del COLCAP
- Valorar si el SVM funciona en procesos de volatilidad inestable
- Encontrar los rezagos necesarios en las variables explicativas para generar una estrategia con los mejores índices posibles

3. Revisión literaria

- Los modelos tradicionales de pronósticos de finanzas con una metodología alternativa que introduce aplicaciones de machine learning en el sector financiero se remontan a principios del siglo XXI en el que se resalta el trabajo de Robert P Schumacher (2006) quien se considera el precursor de la implementación del machine learning en el pronóstico de activos financieros. En 2006 Schumaker realiza el texto *Textual Analysis of Stock Market Prediction Using Financial News Articles* (método AZFinText), en el cual propone comparar un análisis de texto de noticias financieras aplicando una regresión SVM contra una regresión lineal, este fue el primer artículo en retar modelos de *machine learning* encontrando una mejora sensible en los pronósticos debido a la inclusión de información no estructurada. Este estudio es un pionero en la aplicación de los modelos de *machine learning* a finanzas y, por lo tanto, un referente indispensable a la hora de realizar este tipo de investigaciones. El desarrollo de los algoritmos de machine learning continuó con Schumacher (2008) *Sentiment Analysis of Financial News Articles* donde se explora la posibilidad de usar el análisis de sentimientos en la exploración de datos no estructurados.

  Mientras tanto, en Asia se proponen los primeros documentos de análisis de datos estructurados Huang, Nakamori, Wang (2005) en el documento *Forecasting stock market movement direction with support vector machine* proponen el uso de SVM para pronosticar los movimientos del índice Nikkei 225 en función de un índice de mercado y una tasa de cambio. En el presente documento se retoma esta idea y se propone aplicarla al mercado colombiano, adicionando rezagos de la variable con lo cual se busca mejorar el 73% de aciertos obtenidos con el modelo aplicado a mercado asiático.

  Posterior a este trabajo Kara y Baykan (2011) en el documento *Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange* muestran el desempeño de los modelos SVM con los distintos Kernels mostrando que el mejor de los Kernels para pronosticar movimientos de series financieras es el Gaussiano, basada en estas conclusiones se usará el Kernel Gaussiana para el modelo mostrado en este documento.

  Durante los últimos años el desarrollo del machine learning aplicado a finanzas se ha trasladado ha desarrollado mayormente en Asia donde se ha buscado optimizar los tiempos de procesamiento tal es el caso del documento propuesto por Yeh, C. Y., Huang, C. W., & Lee, S. J (2011) (*A multiple-kernel support vector regression approach for stock market price forecasting. Expert Systems with Applications*) en el que se agrega el modelo de regresión al SVM y se compara contra un modelo ARIMA para el pronóstico de precios para pronosticar el índice TAIEX (Índice de valores de la bolsa de Taipei) es importante resaltar que los modelos *machine learning* han demostrado tener mejor capacidad de pronóstico con respecto a los modelos tradicionales. En 2014, un equipo interdisciplinar conformado por Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., ... & Deng, X. (Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications*) trata

de Mostrar la eficacia de los distintos métodos de machine learning para el pronóstico del precio de Hshre (criptomoneda). En el que muestran la necesidad de reducir los tiempos de computo en la Kernelización en los modelos de Extreme Machine learning (ELM). Este documento muestra que SVM no es solo el modelo que mejor pronostica, sino que, además con respecto a los métodos de machine learning más avanzados es el que menos tiempo de proceso toma. Este documento es un aporte a la implementación de estos mercados en Colombia que son poco o nada utilizados, se espera que este documento genere una línea de investigación probando por ejemplo con datos de alta frecuencia, regresiones de SVM medición de tiempos de proceso, entre otras posibles líneas de investigación.

4. Marco teórico

Support vector machine

Las máquinas soporte Vectorial (SVM, del inglés Support Vector Machines) tienen su origen en los trabajos sobre la teoría del aprendizaje estadístico y fueron introducidas en los años 90 por Vapnik y sus colaboradores (Support-vector networks. *Machine learning*, 1995). Aunque originariamente las SVMs fueron pensadas para resolver problemas de clasificación dirigida binaria, su aplicación se ha extendido a problemas de clasificación múltiple y regresión.

Las Máquinas de Vector Soporte se fundamentan en el Maximal Margin Classifier, que, a su vez, se basa en el concepto de hiperplano. A continuación, se explicarán los diferentes casos de SVM desde el caso en el que los datos son fácilmente separables en el espacio de los datos, hasta la complejidad de no ser separables.

Caso linealmente separables

Este es el ejemplo más sencillo de las posibles formas para separar un conjunto de datos. Intuitivamente es el caso en el que existe un plano que separa perfectamente un conjunto de datos. Para empezar, se definirá formalmente este caso.

Definición 1 Conjunto separable Un conjunto de datos será separable si dado un las parejas $\Omega = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$ con $x_i \epsilon \mathbb{R}^n$ y $y \epsilon \{-1,1\}$ (variable de clasificación) existe $W^t = (w_1, w_2, \dots, w_n)$ y b tal que $y_i(W^t X + b) \geq 0$.

En la anterior expresión cada y_i hace referencia la función de clasificación que, en la práctica indicará si la observación está o no por encima del hiperplano $(W^t X + b = 0)$.

De una manera un poco más formal dado $W^t = (w_1, w_2, \dots, w_n)$ un vector de constantes fijas, el conjunto de observaciones X'_i pertenecerá al Hiperplano si se cumple que $(W^t X'_i + b = 0)$. En caso que $(W^t X'_i + b > 0)$ el clasificador $y_i = 1$, en caso que $(W^t X'_i + b < 0)$ entonces $y_i = -1$.

Hiperplano Optimo de separación

El hiperplano $(W^t X + b = 0)$ que permite separar las dos clases no suele ser único. La selección de un hiperplano de entre todos los posibles hiperplanos de separación se realizará a partir del concepto de margen, que explicaremos más adelante y permitirá la optimización del proceso.

Definición 2: Vector de soporte. Un vector de soporte es aquel vector paralelo generado por la observación más cercana al hiperplano. La ecuación de dicho vector de soporte está dada por $(W^t X + b = \pm 1)$. Pues $y_i = \pm 1$.

Definición 3: Margen. Se define margen como la distancia máxima generada por los vectores de soporte de dos poblaciones. Se denotará por τ.

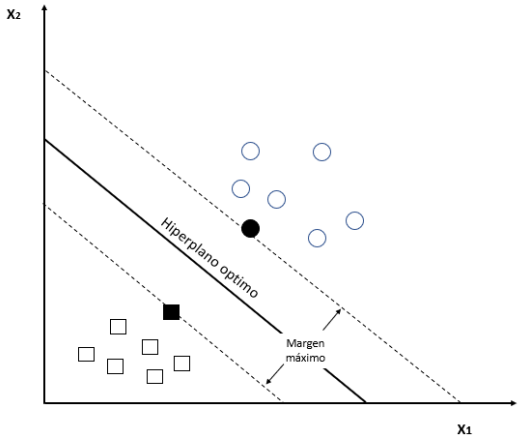

***Ilustración 1**: SVM casos separables*

El objetivo del método es maximizar la distancia entre los conjuntos de datos, lo que implica maximizar el Margen que viene dado por las ecuaciones de los vectores de soporte:

$$\begin{cases} W^t x_+ + b = 1 \\ W^t x_- + b = -1 \end{cases}$$

Donde x_+ corresponde a todos los elementos x_i que se encuentran por encima del hiperplano y x_- denota los elementos en la parte inferior del hiperplano.

Al restar las dos ecuaciones se obtiene la función a optimizar: $W^t(x_+ - x_-) = 2$.

Dividiendo por la norma de *W* entonces:

$$\frac{W^t}{||W||}(x_+ - x_-) = \frac{2}{||W||}$$

Donde $\left||W|\right| = \sqrt{W^t W}$

Con esto encontrar un máximo de $\frac{2}{||W||}$ que equivale a encontrar un mínimo de $\frac{||W||^2}{2}$ sujeto a $y_i(W^t X + b) \geq 1 \ \ \forall i$, que es la ecuación del hiperplano generado por los vectores de soporte.

A continuación, se ilustra el problema de optimización.

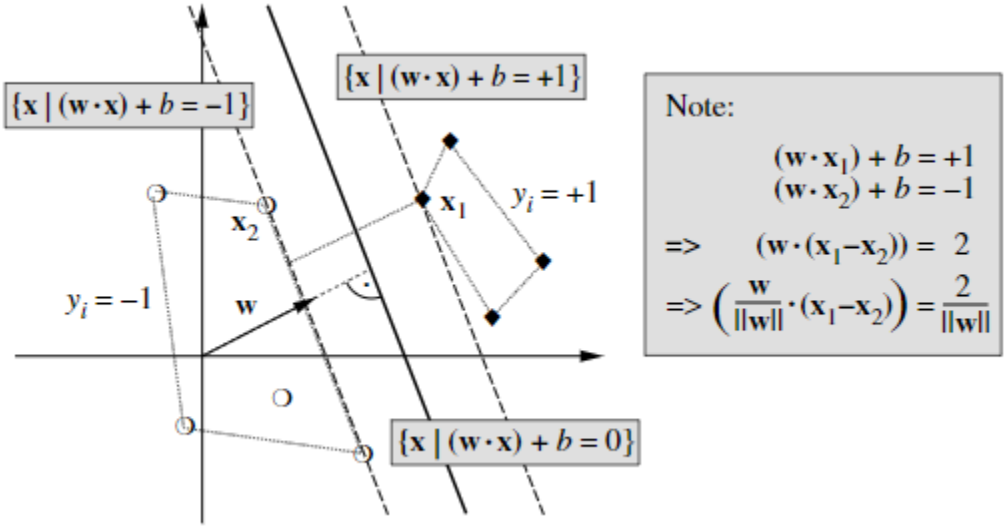

Ilustración 2: *Geometria del SVM*

Para mayor comodidad de notación se notará $x_{\pm} = \varphi(x_i)$ que no es más que la función de clasificación del algoritmo.

De tal manera que:

$$\begin{cases} \varphi(x_i) = x_+ \ si \ y_i = 1 \\ \varphi(x_i) = x_- \ si \ y_i = -1 \end{cases}$$

El problema de optimización

Para encontrar el hiperplano óptimo de separación, se debe minimizar la distancia entre el conjunto de datos y el hiperplano, es decir:

$$d(x, H) = \min_{i=1,2,\ldots,m} \frac{y_i(w^t \varphi(x_i) \ + b)}{||w||^2}$$

$$\begin{cases} \min_i \frac{1}{2}||w||^2 \\ \text{Sujeto a } y_i(w^t \varphi(x_i) + b) \geq 1 \end{cases}$$

Ecuación 1: *Función de optimización svm*

Condiciones de Karush, Kuhn y Tucker (KKT).

La condición de KKT d fueron publicadas por primera vez (1939) en la tesis de Maestría de William Karush (1917-1997) y son las condiciones necesarias que deben satisfacer los óptimos de problemas de optimización no lineal con restricciones de desigualdad.

El problema a considerar es:

$$\min f(x_1, x_2, \dots, x_n)$$

$$\text{s.a } g_1(x_1, x_2, \dots, x_n) \leq 0$$

$$g_2(x_1, x_2, \dots, x_n) \leq 0$$

$$\vdots$$

$$g_n(x_1, x_2, \dots, x_n) \leq 0$$

No existe un algoritmo general para resolver modelos no lineales debido al comportamiento irregular de dichas funciones. Es por ello que en contraste con la programación lineal no se puede reducir el campo de elección al conjunto de puntos extremos de la región factible.

Sin embargo, se han determinado condiciones que bajo ciertos requisitos se convierten en condiciones de primer orden o necesarias, e inclusive en condiciones necesarias y suficientes. Estas son las condiciones de Karush – Kuhn – Tucker , que se indicarán como condiciones de KKT, y fueron desarrolladas independientemente por Karush y por Kuhn – Tucker.

El método de solución procede de la siguiente manera. Cambiando $g_i \leq 0$ a una restricción de igualdad introduciendo una variable s_i de la siguiente manera:

$$g_i \leq 0 \ \rightarrow \ g_i + s_i^2 = 0$$

Para aplicar multiplicadores de Lagrange se constituye la función:

$$F(\boldsymbol{x}, \lambda, \boldsymbol{s}) = f(\boldsymbol{x}) + \sum_{i=1}^{n} \lambda_i \cdot \left(g_i + s_i^2\right)$$

Ecuación 2: *Forma funcional para condiciones KKT*

Los puntos que minimizan a *f* sujeta a las restricciones $g_i \leq 0$ están dentro de los puntos críticos de *F*:

- Que hacen cero las parciales con respecto a las variables x_i:

$$\frac{\partial F}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} = 0$$

- Que hacen cero las parciales con respecto a las variables λ:

$$\frac{\partial F}{\partial \lambda_i} = g_i + s_i^2 = 0 \leftrightarrow g_i \leq 0$$

- Que hacen cero las parciales con respecto a las variables s_i^2

$$\frac{\partial F}{\partial \lambda_i} = 2\lambda_i s_i = 0 \leftrightarrow s_i = 0 \leftrightarrow \lambda_i g_i = 0$$

Lo anterior se resume en el siguiente teorema que indica las condiciones que deben satisfacer los puntos que minimizan la función sujeta a las restricciones.

Teorema 1

Suponga una formulación para el problema anterior de minimización. Si $\boldsymbol{x}_0 \epsilon \mathbb{R}^n$ es un óptimo, entonces deben existir números reales llamados multiplicadores $\Lambda_i \epsilon \mathbb{R}^m \geq 0$ tales que, $(\boldsymbol{x}_0, \Lambda_i)$ es punto crítico para *F*. Es decir que se cumple:

Condición 1 (Estacionalidad):

$$\frac{\partial f(\boldsymbol{x_0})}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\boldsymbol{x_0})}{\partial x_j} = 0 \qquad j = 1,2 \dots, n$$

Condición 2 (Condición de holgura complementaria):

$$\lambda_i g_i(\boldsymbol{x_0}) = 0 \qquad\qquad i = 1,2, \dots, m$$

Condición 3:

$$g_i \leq 0 \qquad\qquad i = 1,2, \dots, m$$

Es importante ver que los valores de s_i se obtienen de la relación $g_i + s_i^2 = 0$ y de que $g_i \leq 0$.

En caso de tratarse de un problema de maximización:

$$\max f(x_1, x_2, \dots, x_n)$$

$$\text{s.a } g_1(x_1, x_2, \dots, x_n) \leq 0$$

$$g_2(x_1, x_2, \dots, x_n) \leq 0$$

$$\vdots$$

$$g_n(x_1, x_2, \dots, x_n) \leq 0$$

Basta con replantear el problema anterior cambiando $f(x)$ por $-f(x)$, es decir:

$$F(\boldsymbol{x}, \lambda, \boldsymbol{s}) = -f^*(\boldsymbol{x}) + \textstyle\sum_{i=1}^{n} \lambda_i \cdot \left(g_i^* + s_i^2\right) \text{ (*)}$$

Esta condición es conocida como la solución Dual del problema. El problema dual básicamente se hace referencia a dos funciones una cóncava y una convexa dentro de una región que comparten un punto crítico, es decir, si para el problema original se tiene un problema de maximización para la segunda función a la se notará como el problema dual tendrá un problema de minimización y viceversa. Las condiciones bajo las cuales se puede determinar esta ecuación dual están dadas por las condiciones KKT que son:

Condición 1 (Estacionalidad):

$$-\frac{\partial f(\boldsymbol{x_0})}{\partial x_j}+\sum_{i=1}^{m}\lambda_i\frac{\partial g_i(\boldsymbol{x_0})}{\partial x_j}=0 \qquad j=1,2\dots,n$$

Condición 2 (Condición de holgura complementaria):

$$\lambda_i g_i(\boldsymbol{x_0})=0 \qquad\qquad i=1,2,\dots,m$$

Condición 3:

$$g_i\leq 0 \qquad\qquad i=1,2,\dots,m$$

Condiciones KKT en SVM:

Volviendo al problema de optimización:

$$\begin{cases}\min\limits_{x}\dfrac{1}{2}||w||^2\\ \text{Sujeto a } y_i(w^t\varphi(x_i)+b)\geq 1\end{cases}$$

El objetivo será encontrar las condiciones necesarias para que el margen sea máximo, además de una forma computacional que permita encontrar la solución al problema.

Lo primero que se hará será encontrar el Lagraniano de la función:

$$\mathcal{L}(w,b,\alpha)=\frac{1}{2}||w||^2-\sum_{i=1}^{n}\alpha_i(y_i(w^t\varphi(x_i)+b)-1)$$

Ecuación 3: *Lagraniano SVM casos separable*

Los parámetros a encontrar son la pendiente *w* y el intercepto *b.* Derivando el Lagraniano con respecto a los parámetros:

- $\frac{\partial\mathcal{L}}{\partial W}=0 \Rightarrow \nabla_w\mathcal{L}(w,b,\alpha)=w-\sum_{i=1}^{n}\alpha_i y_i\varphi(x_i)=0$
- $\frac{\partial\mathcal{L}}{\partial b}=0 \Rightarrow \nabla_b\mathcal{L}(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i y_i=0$

Con esto validamos las condiciones KKT:

La forma funcional dada en la ecuación 2:

$$F(\boldsymbol{x}, \lambda, \boldsymbol{s}) = f(\boldsymbol{x}) + \sum_{i=1}^{n} \lambda_i \cdot \left(g_i + s_i^2\right)$$

Viene dada por:

$$f(w) = \frac{1}{2}||w||^2$$

$$g(w) = -y_i(w^t \varphi(x_i) + b)$$

$$s_i^2 = 1$$

Así la forma funcional para las condiciones del SVM será

$$F(\boldsymbol{x}, \lambda, \boldsymbol{s}) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \lambda_i (y_i(w^t \varphi(x_i) + b) - 1)$$

A partir de esta forma funcional se encontrarán las condiciones KKT que permitirán encontrar el optimo y la función dual.

Condición 1:

$$\frac{\partial f(\boldsymbol{x_0})}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\boldsymbol{x_0})}{\partial x_j} = 0$$

Dado lo anterior derivando con respecto a *w*

$$w - \textstyle\sum_{i=1}^{n} \lambda_i y_i \varphi(x_i) \ = 0 \ \Rightarrow \textstyle\sum_{i=1}^{n} \lambda_i y_i \boldsymbol{\varphi}(\boldsymbol{x_i}) = w$$

Derivando respecto a *b* se obtiene:

$$\sum_{i=1}^{n} \lambda_i y_i \ = 0$$

Ecuación 4:*Restricciones adicionales para los coeficientes* λ_i

Condición 2:

Aplicando la segunda condición se tiene:

$$\lambda_i g_i(\boldsymbol{x_0}) = 0$$

$$\lambda_i(y_i(w^t \varphi(x_i) + b)) = 0$$

Condición 3:

$$g_i \leq 0$$

Esta condición se tiene por la definición del problema

De $g_i + s_i^2 = 0$ se obtiene el óptimo para b de la siguiente manera:

$$(y_i(w^t\varphi(x_i) + b) - 1) = 0$$

$$b = \frac{1}{y_i} - w^t\varphi(x_i)$$

Dado que $y_i = \pm 1$ entonces

$$b = y_i - w^t\varphi(x_i)$$

El parámetro *w* optimo está dado por:

$$w = \sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)$$

Ecuación 5: *Parámetros óptimos del SVM*

Reemplazando en el Lagraniano tenemos:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \lambda_i (y_i(w^t\varphi(x_i) + b) - 1)$$

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \lambda_i (y_i w^t\varphi(x_i)) + (b\sum_{i=1}^{n} \lambda_i y_i) + \sum_{i=1}^{n} \lambda_i)$$

Aplicando la restricción de parámetros dado por la ecuación 4 y aplicando el parámetro optimo *w* se tiene:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}\left(\sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)\right)\left(\sum_{j=1}^{n} \lambda_j y_j \varphi(x_j)\right) - \sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)\left(\left(\sum_{j=1}^{n} \lambda_j y_j\right)^t \varphi(x_j)\right) + \sum_{i=1}^{n} \lambda_i$$

$$\mathcal{L}(w, b, \lambda) = -\frac{1}{2}\left(\sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)\right)\left(\sum_{j=1}^{n} \lambda_j y_j \varphi(x_j)\right) + \sum_{i=1}^{n} \lambda_i$$

$$\mathcal{L}(w, b, \lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\left(\sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)\right)\left(\sum_{j=1}^{n} \lambda_j y_j \varphi(x_j)\right)$$

$$\mathcal{L}(w, b, \lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \varphi(x_i)\varphi(x_j)$$

Expresando lo anterior como un producto interno se tiene:

$$\mathcal{L}(w,b,\lambda)=\sum_{i=1}^{n}\lambda_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\langle\varphi(x_i)\varphi(x_j)\rangle$$

Ecuación 6: *Problema Dual asociado a la optimización SVM*

Al igual que el problema primal, este problema es abordable mediante técnicas estándar de programación cuadrática. Sin embargo, como se puede comprobar, el tamaño del problema de optimización dual escala con el número de muestras, *n*, mientras que el problema primal lo hace con la dimensionalidad, *d*. Por tanto, aquí radica la ventaja del problema dual, es decir, el coste computacional asociado a su resolución es factible incluso para problemas con un número muy alto de dimensiones.

En la práctica, existe un *b* por cada vector de soporte por lo cual es habitual promediar todos los b para la selección final de los datos, con esto b es:

$$b=\frac{1}{N_s}\sum_{0<\alpha_i<C}[y_i-w^t\varphi(x_i)]$$

Así cada vez que llegue una nueva observación la función de clasificación está dada por

$$f(x)=Signo(w^t\varphi(x_i)+b)$$

Cambiando por los óptimos encontrados en pasos anteriores

$$f(x)=signo\left(\sum_{i=1}^{n}\alpha_i y_i\varphi(x_i)^t\varphi(x_i)+\frac{1}{N_s}\sum_{0<\alpha_i<C}\left[y_i-\sum_{i=1}^{n}\alpha_i y_i\varphi(x_i)^t\varphi(x_i)\right]\right)$$

Caso no separable lineal.

En los problemas reales encontrar un conjunto con dos clases totalmente separables es escasamente probable, entre otras cosas por la existencia de ruido en los datos, para tratar con este tipo de casos con ruido es introducir un conjunto de variables reales y positivas, variables artifíciales, ξ_i =1,2,…,n de forma que permitan algunos ejemplos no separables, es decir:

$$y_i(W^t\varphi(x_i)+b)\geq 1-\xi_i$$

Ecuación 7: ***ecuación de hiperplano permitiendo errores en clasificación***

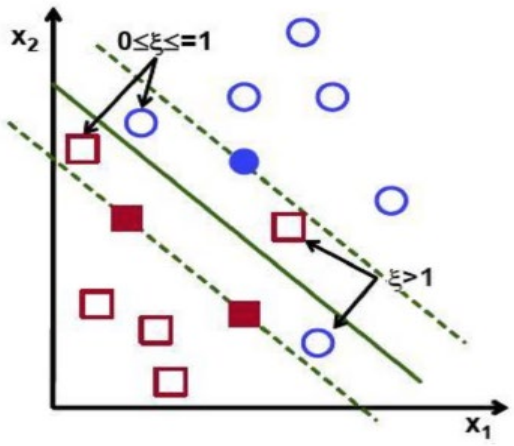

Ilustración 3: Hiperplano con errores de clasificación

De acuerdo con la expresión anterior los ejemplos con variable artificial nula corresponden a ejemplos separables, mientras que los que tengan asociada una variable artificial positiva son los llamados ruido, mal clasificados. La función a optimizar en este caso debe incluir estas variables artificiales de forma que controle el error en la clasificación permitiendo un margen ξ_i de equivocación. De esta forma el nuevo problema de optimización a resolver será:

$$min \frac{||w||^2}{2} + C \sum_{i=1}^{n} \xi_i$$

sujeto a $$y_i(w^t \varphi(x_i) + b) + \xi_i - 1 \geq 0$$

$$\xi_i \geq 0$$

Donde *C* es una constante positiva a determinar de la que puede depender el clasificador. Al mismo tiempo, vamos a penalizar ese error con un nuevo parámetro C, que es añadido a la función objetivo, de tal manera que, a mayor C, mayor es la penalización que damos a los errores y por tanto permitimos menos. Por el contrario, si C es pequeño, permitimos a nuestro modelo cometer más errores. Este hiperplano se conoce como *soft margin*.

El valor de C va ser importante en este sentido, ya que, si damos un valor demasiado grande, el modelo penalizara mucho los errores cometidos en el conjunto de entrenamiento y por tanto se producirá el overfitting o sobre aprendizaje, esto es, el modelo sobre aprender los datos de entrenamiento, ciñéndose a ellos, lo que produce que no haya una buena generalización y la clasificación en los nuevos datos de test no sea buena. Por otro lado, si C es muy pequeño, el modelo permitirá muchos errores y no será bueno, produciéndose el conocido underfitting

Condiciones KKT:

Lo primero es expresar la ecuación para que se cumpla el teorema de las restricciones KKT:

$$min\frac{||w||^2}{2} + C\sum_{i=1}^{n}\xi_i$$

sujeto a $$1 - (y_i(w^t\varphi(x_i) + b) + \xi_i) < 0$$

$$-\xi_i < 0$$

$$F(\boldsymbol{x}, \lambda, \boldsymbol{s}) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i(y_i(w^t\varphi(x_i) + b) - 1 + \xi_i) - \sum_{i=1}^{n}\alpha_i\xi_i$$

Aplicando las condiciones KKT:

Condición 1:

Derivando respecto a los parámetros se deriva:

$$\frac{\partial F}{\partial w} = w - \sum_{i=1}^{n}\lambda_i y_i\varphi(x_i) = 0 \Rightarrow w = \sum_{i=1}^{n}\lambda_i y_i\varphi(x_i)$$

$$\frac{\partial F}{\partial b} = \sum_{i=1}^{n}\lambda_i y_i = 0$$

$$\frac{\partial F}{\partial \xi_i} = C - \lambda_i - \alpha_i = 0 \Rightarrow C = \lambda_i + \alpha_i$$

Con esta condición se encuentra el óptimo de *w* en función de λ_i y las restricciones de λ_i y α_i. Nótese que dado que $\lambda_i \geq 0$ y $\alpha_i \geq 0$ se puede deducir fácilmente que $0 \leq \alpha_i \leq C$

.

Condición 2:

$$\lambda_i g_i(\boldsymbol{x_0}) = 0$$

$$\lambda_i(y_i(w^t\varphi(x_i) + b) - \xi_i) = 0$$

$$\lambda_i\alpha_i = 0$$

Condición 3:

$$-\xi_i \leq 0$$

$$-(y_i(w^t\varphi(x_i) + b) + \xi_i - 1) \leq 0$$

Bajo estas condiciones se establece el óptimo de *w* y las restricciones de las variables duales como:

$$\widehat{w} = \sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)$$

Ecuación 8: *Valor optimo encontrado en el caso lineal no separable para w*

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

$$C = \lambda_i + \alpha_i$$

Ecuación 9: *Restricciones de los parámetros en el caso lineal no separable*

De manera análoga al cálculo de la ecuación 6 se obtiene el dual de la función que está dado por:

$$\mathcal{L}(w, b, \lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \langle \varphi(x_i) \varphi(x_j) \rangle$$

Con esto el problema dual asociado será:

$$\max \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \langle \varphi(x_i) \varphi(x_j) \rangle$$

$$\text{sujeto a } \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$0 \leq \alpha_i \leq C$$

Ecuación 9: *Forma dual de SVM en el caso lineal no separable*

Caso no lineal

A pesar de la mejora realizada en el caso lineal no separable, este caso aún es muy difícil de encontrar en la práctica por lo menos en espacios pequeños, lo que hace necesario pensar en espacios más grandes y complicados, pues en la medida que se sube de espacio la separación lineal necesitará más parámetros, por lo cual hace que la optimización sea engorrosa aun cuando se trabaje con el dual. Por eso será de gran utilidad encontrar funciones que permitan ir a espacios más grandes. En esta sección se explorará el truco del Kernel que permitirá separaciones lineales en espacios grandes con la facilidad de la optimización en espacios más pequeños.

Sea $\Phi: \mathbb{X} \longrightarrow \mathcal{F}$ la función que hace corresponder a cada punto de entrada x un punto en el espacio de características $\mathcal{F}$. Lo que pretendemos es encontrar el hiperplano de separación en este nuevo espacio $\mathcal{F}$. Este hiperplano en el espacio de características se transforma en una función no lineal que separa nuestro conjunto en el espacio original de entradas.

Dado lo anterior si el hiperplano separador en el espacio $\mathcal{F}$ es de la forma $f(x) = w^t \varphi(x_i) + b$ en el espacio $\mathbb{X}$ se convertirá en $f(x) = \phi(w^t) \varphi(x_i) + b$ obteniéndose un separador lineal en el espacio proyectado para datos no linealmente separables en el espacio de entrada.

De esta forma el problema de optimización será:

$$\text{Min } \frac{||W||^2}{2} + C\sum_{i=1}^{n}\xi_i$$

sujeto a $$y_i(W^t\Phi(X) + b) + \xi_i - 1 \geq 0$$

$$\xi_i \geq 0$$

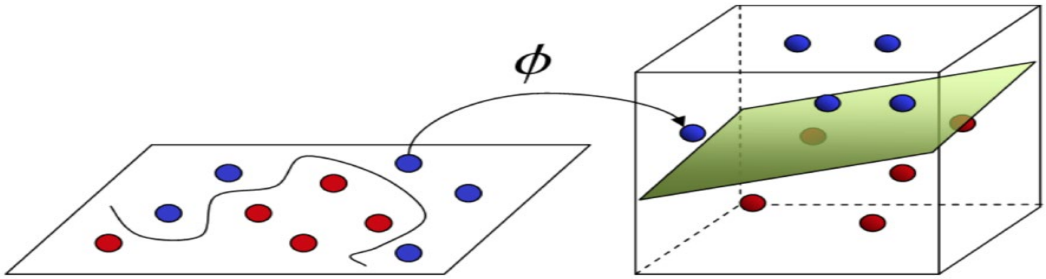

Ilustración 4: *Separación en casos no lineales a través de la función Φ*

Con esto el problema de optimización utilizando multiplicadores de Lagrange y la forma dual de la ecuación vista en la sección anterior aplicando la función Φ es:

$$\text{máx}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\lambda_j\lambda_i y_i y_j\langle\Phi(x_i),\Phi(x_j)\rangle$$

$$\text{Sujeto a } \sum_{i=1}^{n}\lambda_i y_i = 0$$

$$0 \leq \lambda_i \leq C$$

Ecuación 10: *Problema dual SVM no lineal.*

El problema que tiene esta técnica es la necesidad de conocer ϕ_i, para esto se propone usar la función Kernel que permite no conocer la función $\phi_i$simplemente evaluando el producto interno de las funciones lo que nos permite usar explícitamente la función de inversión Φ. Para empezar, se definirá la función Kernell.

Definición 4 (función Kernel): Sea X el espacio de entrada, *H* un espacio dotado con producto interno al que en adelante se nombrará como el espacio de características. Además sea *F* una función tal que $F: X \to H$ con *H* espacio inducido de Hilbert, se define la función Kernel $F: X \times X \to \mathbb{R}$ como:

$$K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$$

Ecuación 11: *Función Kernel*

El problema con esta técnica es que no cualquier función puede ser un Kernel, ya que algunas veces esa función no se puede descomponer como sugiere la definición 4. Sin embargo, el teorema de Mercer permite encontrar una condición suficiente para que se cumpla la definición 4.

Teorema 2 (Teorema de Mercer)

Si una función escalar $k(x_i, x_j)$ es semidefinida positiva, existe una función $\Phi: \mathbb{R}^d \longrightarrow H$, con *H* espacio de Hilbert, tal que $k(x_i, x_j)$ pude descomponerse como un producto interno

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

La anterior definición y el teorema de Mercer sirven para indicar que lo que se busca es construir un hiperplano de separación lineal en el espacio de características *H*.

Dado lo anterior la función Kernel aplicada a los datos será:

$$k(x_i, x_j) = \langle \phi(\boldsymbol{x}), \phi(x') \rangle = \phi_1(x)\phi_1(x') + \phi_2(x)\phi_2(x') + \cdots + \phi_m(x)\phi_m(x')$$

Donde $\Phi: X \longrightarrow H$, además la función Φ es *m*-dimensional, esto en la práctica indicará el tamaño del espacio de llegada en el cual las variables son separables linealmente, a veces este espacio es demasiado grande lo que genera demoras en los tiempos de proceso.

Con esto:

$$\widehat{w} = \sum_{i=1}^{n} \lambda_i y_i \varphi(x_i)$$

$$\text{s.a } \sum_{i=1}^{n} \lambda_i y_i = 0$$

Si se reemplaza $\varphi(x_i)$ por una función Kernel tenemos:

$$\widehat{w} = \sum_{i=1}^{n} \lambda_i y_i k(\boldsymbol{x}, x_j)$$

$$\text{s.a } \sum_{i=1}^{n} \lambda_i y_i = 0$$

Dado lo anterior $\widehat{w}$ termina siendo la función estimada del hiperplano en el espacio características *H*, por ende este también será la función de decisión del algoritmo. Es decir, cualquier observación que se encuentre por encima de este hiperplano tomará el valor de y=1 y cualquiera por debajo tomará el valor de y=-1.

Como se decía anteriormente no cualquier función es un Kernel a continuación listamos algunos de los más usados:

Kernel Lineal: $K(x, x') = < x, x' >$

Kernel polinómico de grado –p: $K(x, x') = [\gamma \langle x, x' \rangle + \tau]^p$

Kernel Gaussiano: $K(x, x') = exp(-\gamma\|x - x'\|^2), con\ \gamma > 0$

Kernel sigmoidal: $K(x, x') = \tanh(\gamma\langle x, x'\rangle + \tau)$

Ejemplo practico

En base a la teoría construida, esta sección busca mostrar con un pequeño ejemplo cómo funciona el algoritmo. El objetivo del ejercicio será separar sin error los movimientos de COLCAP en función al movimiento del cierre S&P500 y el movimiento de la TRM. Para el ejemplo se usarán los siguientes datos.

Fecha	S&P 500	Dólar/peso	COLCAP
23/04/2019	2945,36	3229,50	1595,02
24/04/2019	2951,27	3253,30	1600,17
25/04/2019	2923,48	3263,15	1599,81
26/04/2019	2904,37	3239,85	1600,75
29/04/2019	2914,79	3229,50	1590,38

Tabla 1: *Valores de S&P, Dólar y COLCAP desde el 23 de Abril de 2019 al 29 de Abril*

Para el ejercicio solo se mirarán los movimientos que serán 1 si el precio sube y -1 si baja.

Fecha	S&P 500	Dólar/peso	COLCAP
24/04/2019	1	1	1
25/04/2019	-1	1	-1
26/04/2019	-1	-1	1
29/04/2019	1	-1	-1

Tabla 2*: Base para clasificar*

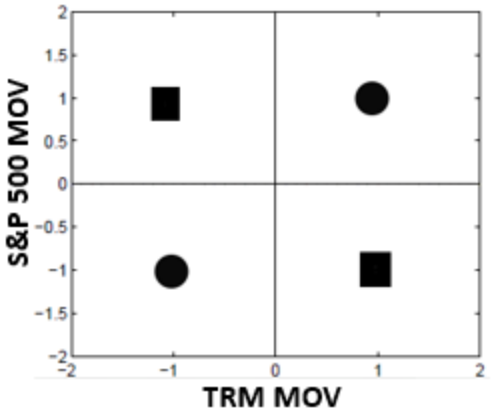

Ilustración 5: *Datos a clasificar*

Como se puede observar en la gráfica, la separación en $\mathbb{R}^2$ es imposible mediante un plano, por ende será necesario aplicar Kernel para subir de espacio y lograr una separación adecuada. Para esto se usará como ejemplo una función Kernel polinómica con parámetros $p = 2, \gamma = 1\ y\ \tau = 1$, la fórmula de este Kernel está dada por:

$$K_2(x, x') = [\langle x, x' \rangle + 1]^2$$

Aplicando la definición de Kernel y teniendo en cuenta que el problema es de dos variables la variable *x* hace referencia a un vector de dos componentes, basado en esto se tiene:

$$K(\boldsymbol{x}, x') = \langle \Phi(\boldsymbol{x}), \Phi(x') \rangle = [\langle x, x' \rangle + 1]^2$$

$$= [\langle (x_1, x_2), (x'_1, x'_2) \rangle + 1]^2$$

$$= x_1^2 x'^2_1 + x_2^2 x'^2_2 + 2x_1 x_2 x'_1 x'_2 + 2x_1 x'_1 + 2x_2 x'_2 + 1$$

Separando las componentes de la variable **x** de las de **x'** se encuentran las funciones $\Phi = \{\phi_1(x), \phi_2(x), \dots, \phi_6(x)\}$

$$\langle \left(1, \sqrt{2}x_1 \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2\right), \left(1, \sqrt{2}x'_1 \sqrt{2}x'_2, \sqrt{2}x'_1 x'_2, x'^2_1, x'^2_2\right) \rangle$$

Separando las componentes encontramos las funciones $\phi_i(\boldsymbol{x})$ donde $\boldsymbol{x} \in \mathbb{R}^2$:

$$\phi_1(x_1, x_2) = 1, \qquad \phi_2(x_1, x_2) = \sqrt{2}x_1\ , \quad \phi_3(x_1, x_2) = \sqrt{2}x_2$$

$$\phi_4(x_1, x_2) = \sqrt{2}x_1 x_2, \qquad \phi_5(x_1, x_2) = x_1^2\ , \quad \phi_6(x_1, x_2) = x_2^2$$

Ecuación 12: *funciones* $\phi_i(\boldsymbol{x})$

Con este kernel el siguiente paso será buscar el hiperplano de separación cuya ecuación está dada por:

$$\widehat{w} = \sum_{i=1}^{n} \lambda_i y_i k(\boldsymbol{x}, x_j)$$

Sin embargo es necesario encontrar los λ_i que solucionen el problema dual

$$\text{Máx} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_j \lambda_i y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle$$

$$\text{Sujeto a } \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$0 \leq \lambda_i \leq C$$

Dado que se busca una separación perfecta se hace $C = \infty$ y además reemplazando los y_i por los valores de los movimientos del COLCAP en la tabla 2, tenemos:

$$\sum_{i=1}^{n} \lambda_i y_i = 0 = \lambda_1 - \lambda_2 + \lambda_3 - \lambda_4$$

Se encuentra que $\lambda_i^* = \lambda = 0.125$

Con estos valores se procede a calcular a ecuación del hiperplano tomando los valores de la tabla 2:

$$\widehat{w} = \sum_{i=1}^{n} \lambda_i y_i k(\boldsymbol{x}, x_j) = 0.125 \sum_{i=1}^{4} y_i k(\boldsymbol{x}, x_j) = 0.125 \sum_{i=1}^{4} y_i \langle \Phi(\mathbf{x}), \Phi(x_j) \rangle$$

$$= 0.125 \sum_{j=1}^{4} y_j [\phi_1(x)\phi_1(x_j) + \phi_2(x)\phi_2(x_j) + \cdots + \phi_6(x)\phi_6(x_j)]$$

Evaluando las funciones $\phi_i(x_j)$ de la ecuación 1* con los valores de la tabla 2 para los valores $x_j = (x_1, x_2)$ se tiene:

$\widehat{w} = 0.125[\phi_1(x)\phi_1(x_1 = (x_1, x_2)) + \phi_2(x)\phi_2(x_1 = (x_1, x_2)) + \cdots + \phi_6(x)\phi_6(x_1 = (x_1, x_2)) + \phi_1(x)\phi_1(x_2 = (x_1, x_2)) + \phi_2(x)\phi_2(x_2 = (x_1, x_2)) + \cdots + \phi_6(x)\phi_6(x_2 = (x_1, x_2)) + \phi_1(x)\phi_1(x_3 = (x_1, x_2)) + \phi_2(x)\phi_2(x_3 = (x_1, x_2)) + \cdots + \phi_6(x)\phi_6(x_3 = (x_1, x_2)) + \phi_1(x)\phi_1(x_4 = (x_1, x_2)) + \phi_2(x)\phi_2(x_4 = (x_1, x_2)) + \cdots + \phi_6(x)\phi_6(x_4 = (x_1, x_2))$

$$\begin{aligned}
\widehat{w} = 0.125 \Big[& y_1 \left(\phi_1(x)(1) + \phi_2(x)\left(\sqrt{2}x_{11}\right) + \phi_3(x)\left(\sqrt{2}x_{12}\right) + \phi_4(x)\left(\sqrt{2}x_{11}x_{12}\right) \right. \\
& \left. + \phi_5(x)(x_{11}^2) + \phi_6(x)(x_{12}^2) \right) \\
& + y_2 \left(\phi_1(x)(1) + \phi_2(x)\left(\sqrt{2}x_{21}\right) + \phi_3(x)\left(\sqrt{2}x_{22}\right) + \phi_4(x)\left(\sqrt{2}x_{21}x_{22}\right) \right. \\
& \left. + \phi_5(x)(x_{21}^2) + \phi_6(x)(x_{22}^2) \right) \\
& + y_3 \left(\phi_1(x)(1) + \phi_2(x)\left(\sqrt{2}x_{31}\right) + \phi_3(x)\left(\sqrt{2}x_{32}\right) + \phi_4(x)\left(\sqrt{2}x_{31}x_{32}\right) \right. \\
& \left. + \phi_5(x)(x_{31}^2) + \phi_6(x)(x_{32}^2) \right) \\
& + y_4 \left(\phi_1(x)(1) + \phi_2(x)\left(\sqrt{2}x_{41}\right) + \phi_3(x)\left(\sqrt{2}x_{42}\right) + \phi_4(x)\left(\sqrt{2}x_{41}x_{42}\right) \right. \\
& \left. + \phi_5(x)(x_{41}^2) + \phi_6(x)(x_{42}^2) \right) \Big]
\end{aligned}$$

Reemplazando los datos de la tabla 2 haciendo los valores S&P500 y el Dólar/peso.

$$\begin{aligned}\widehat{w} = 0.125[1 \cdot \Big(&\phi_1(x)(1) + \sqrt{2}\phi_2(x)(1) + \sqrt{2}\phi_3(x)(1) + \sqrt{2}\phi_4(x)(1)(1) + \phi_5(x)(1) \\ &+ \phi_6(x)(1)\Big) \\ &+ (-1)\Big(\phi_1(x)(1) + \sqrt{2}\phi_2(x)(-1) + \sqrt{2}\phi_3(x)(1) + \sqrt{2}\phi_4(x)(-1)(1) \\ &+ \phi_5(x)(1) + \phi_6(x)(1)\Big) \\ &+ (1)\Big(\phi_1(x)(1) + \sqrt{2}\phi_2(x)(1) + \sqrt{2}\phi_3(x)(1) + \sqrt{2}\phi_4(x)(1)(1) \\ &+ \phi_5(x)(1) + \phi_6(x)(1)\Big) \\ &+ (-1)\Big(\phi_1(x)(-1) + \sqrt{2}\phi_2(x)(-1) + \sqrt{2}\phi_3(x)(-1) + \sqrt{2}\phi_4(x)(-1)(-1) \\ &+ \phi_5(x)(1) + \phi_6(x)(1)\Big)]\end{aligned}$$

$$\begin{aligned}\widehat{w} = 0.125\Big[&\phi_1(x) + \sqrt{2}\phi_2(x) + \sqrt{2}\phi_3(x) + \sqrt{2}\phi_4(x) + \phi_5(x) + \phi_6(x) - \phi_1(x) + \sqrt{2}\phi_2(x) \\ &- \sqrt{2}\phi_3(x) + \sqrt{2}\phi_4(x) - \phi_5(x) + \phi_6(x) + \phi_1(x) - \sqrt{2}\phi_2(x) + \sqrt{2}\phi_3(x) \\ &- \sqrt{2}\phi_4(x) + \phi_5(x) - \phi_6(x) - \phi_1(x) - \sqrt{2}\phi_2(x) - \sqrt{2}\phi_3(x) - \sqrt{2}\phi_4(x) \\ &- \phi_5(x) - \phi_6(x)\Big] = \frac{1}{\sqrt{2}}\phi_4(x)\end{aligned}$$

Con este resultado se puede afirmar que dado que se tienen 6 funciones ϕ el resultado encuentra un hiperplano en $\mathbb{R}^6$ con separación perfecta con ecuación.

$$\frac{1}{\sqrt{2}}\phi_4(x) = 0 \Rightarrow \phi_4(x) = 0$$

En $\mathbb{R}^6$ la ecuación del plano se genera a partir de del vector

$$\widehat{w} = (0,0,0,\frac{1}{\sqrt{2}},0,0)$$

Lo que en $\mathbb{R}$ se puede ver como la ecuación:

$$\phi_4(x) = 0$$

$$x_1 x_2 = 0$$

A continuación los datos proyectados en el espacio de características:

	Espacio entradas		Espacio de caracteristicas						y
Fecha	**S&P 500**	**Dólar/peso**	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	**COLCAP**
24/04/2019	1	1	1	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$	1	1	1
25/04/2019	-1	1	1	$-\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	1	1	-1
26/04/2019	-1	-1	1	$-\sqrt{2}$	$-\sqrt{2}$	$\sqrt{2}$	1	1	1
29/04/2019	1	-1	1	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$	1	1	-1

Tabla 3: *Tabla de proyección de datos en el espacio características.*

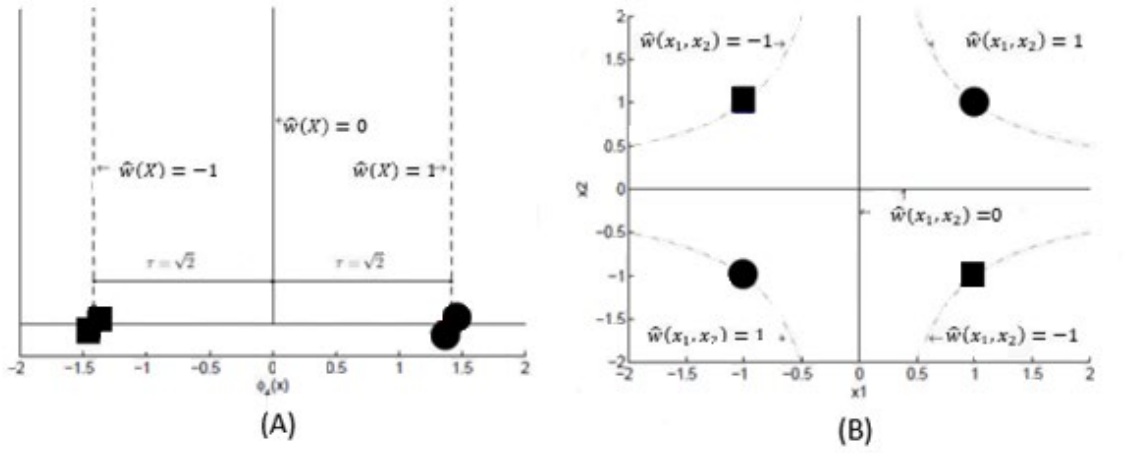

Ilustración 6 (A) Grafico de los datos proyectados en el espacio características (B) Función de decisión no lineal en el espacio de los datos.

En base a lo anterior se pudo encontrar una función que separa perfectamente dos conjuntos.

5. Modelo y estrategia de trading

Con el fin de identificar los movimientos del COLCAP este documento se basa en el documento de Wei Huang (2005) en el que se pronostica los movimientos de NIKKEI 225 que es un índice Japonés en función de un índice de mercado S & P 500 y una tasa de cambio en ese caso YEN/DÓLAR. Este documento fue escogido debido a que los resultados del modelo SVM son baste buenos (73% de acierto).

En este documento adicionamos los auto rezagos del COLCAP, con esto se espera mejorar los resultados obtenidos por Wei Huang (2005).

Datos

Los datos seleccionados se han tomado los precios de cierre desde el mes de junio de 2017 a Abril de 2019, es decir 2 años de ventana de pronóstico. El supuesto del modelo es que existe una relación entre los movimientos del COLCAP, el S & P 500, el USD/COL y los rezagos de los retornos del COLCAP, es decir:

$$Dirección_t = F\left(S_{t-k}^{COLCAP}, S_{t-k}^{S\&P500}, S_{t-k}^{USD/COP}\right)$$

Donde $S_{t-k}^{x} = \log\left(\frac{P_t^X}{P_{t-k}^X}\right)$ donde P_t^X indica el precio del activo *X* en el tiempo t.

Estructura de la base

La estructura de la base de datos vendrá dada con una fila de los diferentes valores de los movimientos de COLCAP medidos en el tiempo *t* y cada columna los rezagos de cada variable explicativa del modelo. Con la base completa se tomará una base de datos con el 80% de los datos para entrenar el modelo, y un 20% con datos que no hacen parte de la base de entrenamiento y con esta se realizará la validación del modelo.

La muestra total está dada por un total de 454 observaciones tomadas desde el 20 de junio de 2017 hasta el 30 de Abril de 2019. Se tomó esta ventana de observaciones de aproximadamente dos años pues se consideró suficientemente grande para cubrir la mayor cantidad de ciclos y situaciones de mercado posibles.

Como se ha comentado anteriormente la muestra de entrenamiento consta del 80% de los registros de la base de datos que consta de 377 registros y una muestra de validación que consta de 77 registros.

Dado que el SVM es un modelo que se basa en distancias, es recomendado estandarizar las variables para que la clasificación del modelo no se vea afectada por efectos de escala de alguna de las variables.

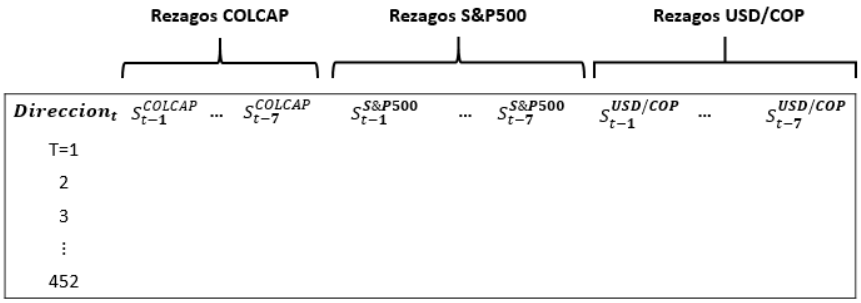

Ilustración 7:*Estructura de la base de datos*

Estimación de parámetros

La mejora que se propone es encontrar el número de rezagos que hace optimo el algoritmo para esto se prueba el modelo con diferentes ventanas de tiempo, en las que encuentra el valor donde se obtengan mayor número de aciertos usando el mismo Kernel, en este caso un Kernel Lineal y el parámetro C=1, parámetro establecido por defecto en R. Para el modelo de pronóstico se usó el Kernel Gausiano pues en la literatura mostró ser el Kernel optimo en los pronósticos de series con volatilidades altas como la de las series financieras.

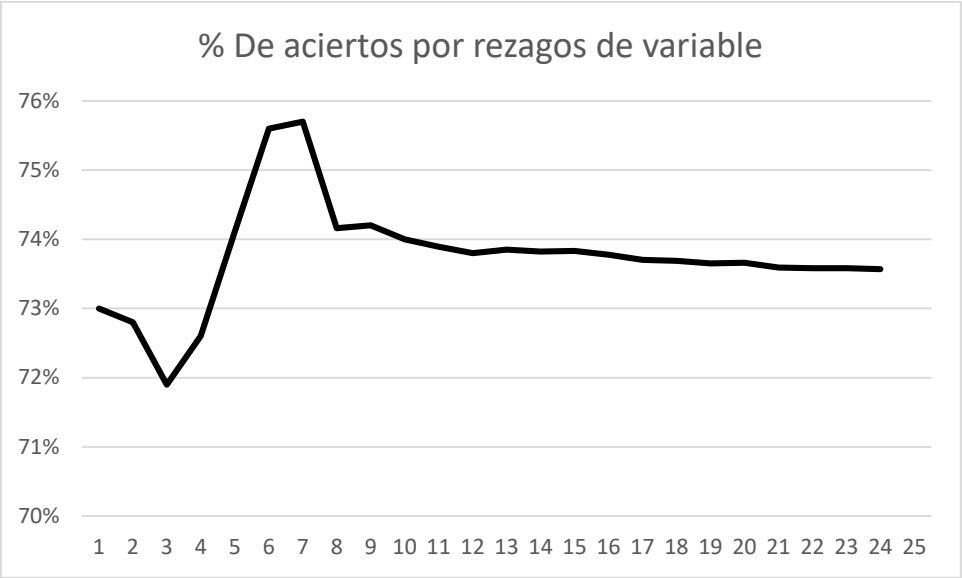

Ilustración 8: *Aciertos para los distintos rezagos de todas las variables*

Como se puede observar en la gráfica el método SVM muestra un buen desempeño con cualquier rezago, con rangos desde el 72% hasta el 75% de aciertos sin afinar el modelo, el número de rezagos que presento más aciertos en el pronóstico fue el rezago 7 por esto, se tomarán 7 rezagos de cada una de las variables explicativas.

El siguiente problema es encontrar el parámetro que hace menor el error de clasificación recordando que si $C \to \infty$ no se permiten errores de clasificación generando sobre-ajuste y que si C=0 se van a tener muchos errores en clasificación, por ende lo adecuado es encontrar un valor que no sea cero, pero dada la volatilidad de las series, se espera usar un parámetro de miss clasificación pequeño. El siguiente ejercicio permite ver como se mueven los errores en función del parámetro, usando los rangos más comúnmente usados en la literatura.

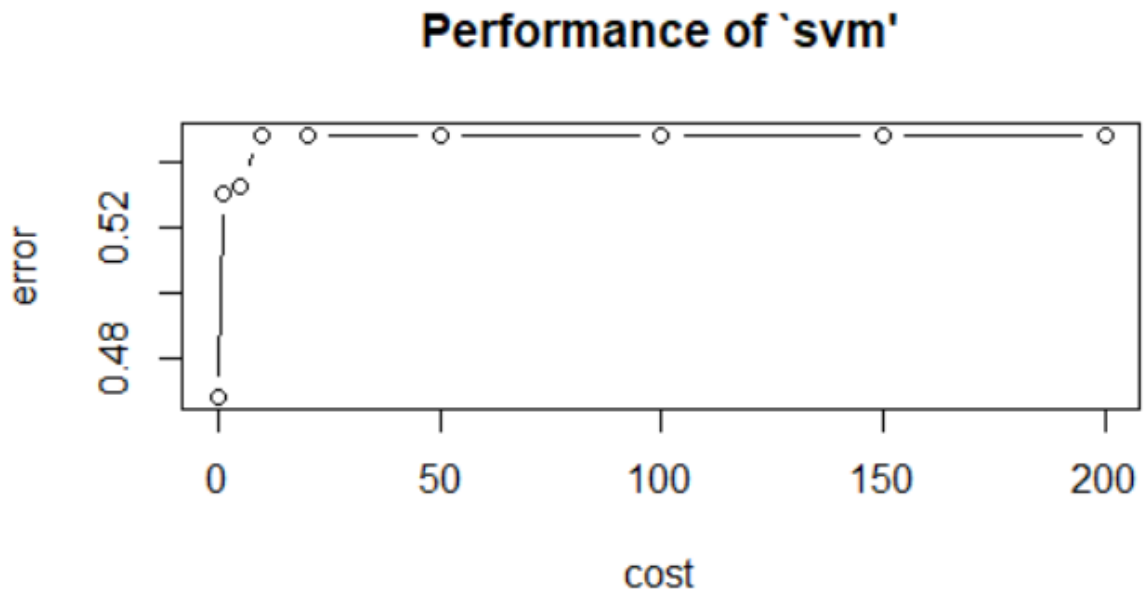

Ilustración 9: Variación de errores en función del parámetro C.

Como se puede observar en la gráfica en la medida en que el parámetro *C* se aumenta, el error de clasificación aumenta, basado en lo anterior se buscó el C donde el cambio de parámetro aumente lo menos posible el error de clasificación y que este no sea cero, por esto el valor seleccionado será de *C*=5.

Resultados

Al aplicar el modelo a la base de entrenamiento usando SVM con un Kernel Gaussiano y un parámetro de miss- clasificación C=5, se obtiene un total de 83% de aciertos discriminados de la siguiente manera:

Entrena_svm_rand$prediccion	Entrena_svm_rand$Delta_Colcap 0	1	Row Total
0	128	6	134
	0.955	0.045	0.355
	0.727	0.030	
	0.340	0.016	
1	48	195	243
	0.198	0.802	0.645
	0.273	0.970	
	0.127	0.517	
Column Total	176	201	377
	0.467	0.533	

Tabla 4 Resultados muestra de desarrollo

A priori el alto de índice de aciertos en la base de entrenamiento podría mostrar un índice de sobre-ajuste del modelo a la base de entrenamiento un ROC del 82% y un KS de 67%.

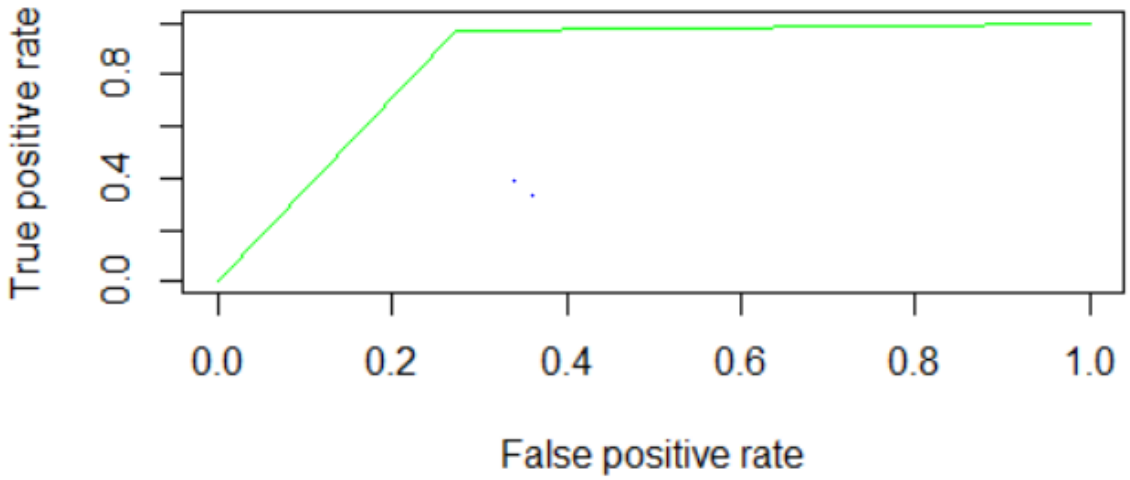

***Ilustración 10**: Curva ROC muestra de entrenamiento*

La muestra de validación muestra que se tiene un porcentaje de aciertos de 81.8%, lo que a priori muestra que el modelo a pesar de su alto grado de acierto es estable. Es importante resaltar que el ROC en la muestra de validación es de 82.9% y un KS de 65.9%

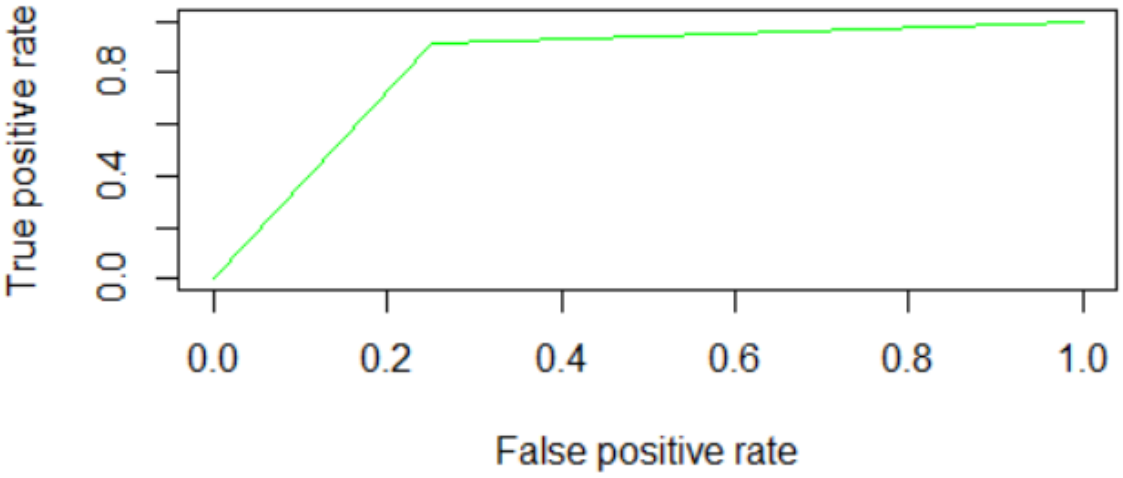

Ilustración 11: Curva ROC muestra de validación

Valida_svm_rand$prediccion	Valida_svm_rand$Delta_Colcap 0	1	Row Total
0	33	3	36
	0.917	0.083	0.468
	0.750	0.091	
	0.429	0.039	
1	11	30	41
	0.268	0.732	0.532
	0.250	0.909	
	0.143	0.390	
Column Total	44	33	77
	0.571	0.429	

Tabla 5: Resultados muestra de validación.

Es claro que el modelo pronostica adecuadamente fuera de muestra, sin embargo, se realizó una tercera base que será la base de la última semana antes de la fecha de realización de este trabajo que corresponde a las fechas del 20 al 24 de mayo de 2019, obteniendo como resultados que el modelo clasifica adecuadamente 4 de los cinco días.

Comparación contra un Benchmark

Para revisar la precisión del modelo se ha comparado el modelo contra un modelo ARIMA más precisamente un modelo AR(1) y un promedio móvil, se probó la bondad de ajuste de los datos en la muestra de validación. Los parámetros del modelo AR son:

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         34.25099   2.625287   2.019    0.044
as.numeric(Colcap_t) 0.977063   0.009846  99.235   <2e-16
```

Tabla 6: *Parámetros AR1*

El porcentaje de aciertos de los tres modelos se puede ver a continuación.

	SVM	**Pormedios moviles**	**AR 1**
Modelo	81.8%	54.5%	47.4%

Tabla 7*: Comparación número de aciertos del modelo*

Propuesta de estrategia de trading

Después de ajustado el modelo se propone generar una estrategia de trading basada en los pronósticos del modelo, esta estrategia se construirá a partir de la distancia entre las distintas observaciones y el hiperplano de probabilidad.

En los datos de entrenamiento el modelo generó puntuaciones entre 0.41 y 0.70, con estas puntuaciones se busca calcular la probabilidad que una observación nueva genere un valor por encima del hiperplano, que mirada desde el punto de vista financiero indicará una subida en el precio del COLCAP. Con estas puntuaciones se genera un score entre cero y uno estandarizando por la puntuación mínima y máxima, es importante resaltar que el hiperplano clasifica las observaciones como 1 si el score calculado es mayor a 49% y 0 en caso contrario.

***Ilustración 12**: Distribución de errores del modelo para score menor a 0.49*

Como se puede ver en la ilustración 12 el 66% de los errores están clasificados dentro del rango de score mayor a 40%. Se decidió tomar como punto de corte una probabilidad del 30% con lo cual se ira en largo y se espera que el modelo se equivoque en apenas un 17% de las veces que el modelo sugiere ir en corto.

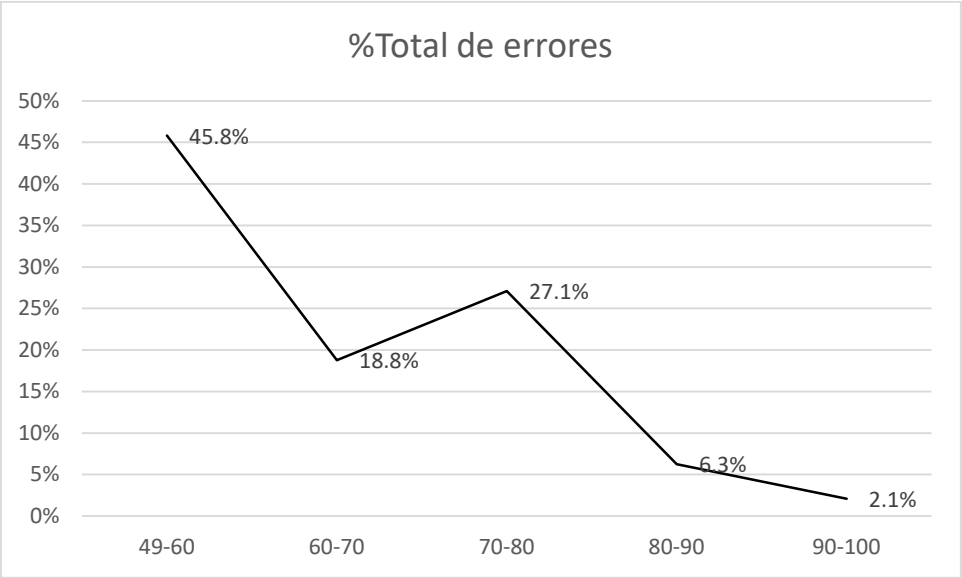

Ilustración 13: *Distribución de errores del modelo para score mayor a 0.49*

La ilustración13 muestra cómo se distribuyen los errores en los scores superiores al 49%, con esto se define que se ejercerá en largo en caso que el score supere 0.8, se espera un error cercano a 10%.

Posterior a la definición de los puntos de corte, se medirá el impacto de la estrategia en la muestra de validación, donde la ganancia será el cambio en COLCAP.

Score	No de casos	Mal clasif	%mal clasificación
<30%	7	2	28.6%
>80%	11	2	18.2%

Tabla 8: *Porcentaje de mala clasificación muestra de validación*

Conclusiones

Como principal conclusión de este documento se obtiene que el SVM es una herramienta con altos niveles de acierto en series financieras, además, se ha podido construir un modelo estable fuera de la muestra de entrenamiento igualmente con altos índices de acierto en los pronósticos. Además, se propone una nueva estrategia de trading que genera bajos índices de error en la toma de decisiones.

Es importante resaltar que a pesar que en la literatura se encuentran altos índices de acierto en el pronóstico del modelo, el modelo propuesto en este documento los supera haciendo de este una poderosa herramienta en la toma de decisiones. Se puede asumir que esto se puede deber a que

el mercado colombiano es más tranquilo que los mercados de primer mundo en los que se ha probado los pronósticos del SVM

A pesar de lo anterior se tienen los siguientes campos de mejora y sugerencias para futuras investigaciones:

- Evaluar la estabilidad del modelo periódicamente para encontrar el punto en el que el modelo debe ser calibrado, pues en la literatura se ha encontrado que estos modelos tienen ventanas de descalibración cortas y tienden a perder poder de pronostico en ventanas de un mes.
- Una de las mejoras a este trabajo pude darse en la posibilidad de generar pronósticos a distintas ventanas de tiempo no solo un periodo adelante usando la metodología desarrollada en este documento.
- Generar portafolios en función de un modelo SVM
- Comparar contra modelos tradicionales y evaluar todo el posible performance de los mismos.
- Comparar contra otros modelos de machine learning con datos de mercado colombiano para encontrar mejores modelos
- Realizar modelos de regresión de SVM que permita evaluar no solo el movimiento de un activo sino la magnitud del mismo

6. Bibliografía

- Schumacher, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)*, *27*(2), 12.

- Schumacher, R. P., Zhang, Y., & Huang, C. N. (2009, October). Sentiment analysis of financial news articles. In *20th Annual Conference of International Information Management Association*.

- Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513-2522.

- Theodoridis, S., & Mavroforakis, M. (2007). Reduced convex hulls: a geometric approach to support vector machines [lecture notes]. *IEEE Signal Processing Magazine*, *24*(3), 119-122.

- Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert systems with Applications*, *38*(5), 5311-5319.

- Gala García, Y. (2013). *Algoritmos SVM para problemas sobre big data* (Master's thesis).

- Martín Guareño, J. J. (2016). Support vector regression: propiedades y aplicaciones.

- Cocianu, C. L., & Grigoryan, H. (2016). MACHINE LEARNING TECHNIQUES FOR STOCK MARKET PREDICTION. A CASE STUDY OF OMV PETROM. *Economic Computation & Economic Cybernetics Studies & Research*, *50*(3).

- Pyo, S., Lee, J., Cha, M., & Jang, H. (2017). Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets. *PloS one*, *12*(11), e0188107.

- Xia, X. L. (2018). Training sparse least squares support vector machines by the QR decomposition. *Neural Networks*, *106*, 175-184