Trabajo 1

Juan Mauricio Cuscagua López - código: 201910017228
Programa: Maestría en Ciencia de los Datos y Analítica
Asignatura: Aprendizaje Automático
Docente: Olga Lucía Quintero
30 de agosto de 2019

I. Descripción del problema y los datos

Para efectos del presente trabajo se consideró explorar un problema de clasificación relacionado al mercado de valores. El objetivo consiste en clasificar la decisión de inversión en el SPY[1]. Las posibles decisiones a tomar son el de comprar el índice, venderlo o no participar del mercado en lo absoluto; dichas decisiones son representadas por los valores 1, -1 y 0 respectivamente en la variable objetivo de predicción. Dicha clasificación depende de si a una ventana al futuro de 3 días, el movimiento del precio del SPY ha cambiado en más del 1 %. Así, se clasifica un día particular como 1 si 3 días después habría tenido una valorización de mas del 1 %, se clasificará como -1 si 3 días después habría tenido una devaluación de mas del 1 % y se clasificará como 0 en cualquier otro caso.

Entre las variables que se usaron como características que aportan información sobre la decisión a tomar, se encuentran los siguientes:

- Desempeño de los sectores: consumo básico, consumo discrecional, energético, financiero, salud, industrial, tecnológico y de servicios públicos.
- Los índices de dolar mundial y volatilidad de Estados Unidos
- Los precios del Euro/Dolar, bonos de tesoros de Estados Unidos a 2, 5 y 10 años.
- Aproximadamente 50 indicadores técnicos sobre la serie de tiempo del SPY, incluyendo medias móviles, retornos del activo en los últimos 10 días, spreads y ratios entre los bonos.

Dichos datos fueron estandarizados bajo la siguiente expresión:

$$x_{estandar} = \frac{x - x_{minimo}}{x_{maximo} - x_{minimo}}$$

Dicha evaluación se hace bajo una ventana móvil de 60 días.

Una primera aproximación a la distribución de los datos se puede ver en el histograma (figura 1) de las 3 clases asociadas.

[1]Instrumento transable en la bolsa de Estados Unidos que replica el movimiento del índice del Standard and Poor's 500 de Estados Unidos

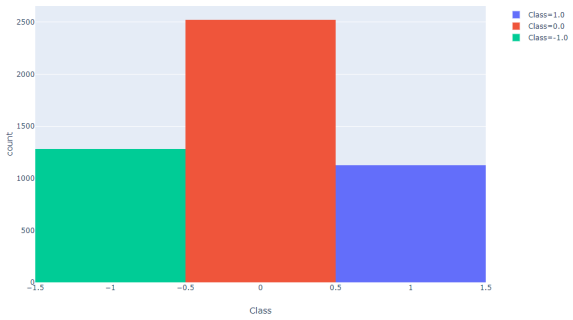

Figura 1: Distribución de las observaciones de cada clase

Se tienen un total de 65 variables explicativas y un total de 4931 observaciones.

II. Selección del error real deseado y el error de entrenamiento deseado

Por la misma naturaleza aleatoria del mercado de valores, se puede probar computacionalmente que la probabilidad de valorización o devaluación del mercado de un día para otro es aproximadamente 0.5, es decir, existe independencia. Por otro lado, el concepto de tendencia en una serie de tiempo permite tomar provecho de la dirección del mercado para sacar utilidades. Basado en esto, se puede asegurar que el error real deseado debería ser mayor a 0.5 para que la construcción del algoritmo tenga sentido (superar el random guessing), es por esto que se tomará el error real deseado como $\epsilon = 0{,}4$

Adicional a esto, se busca que algún algoritmo sea capaz de clasificar correctamente los días en los que se esperaría tener movimientos como se describió anteriormente con una tasa de acierto de un 85 %. De esta manera, el error de entrenamiento deseado se tomará como $\delta = 0{,}15$.

III. División del conjunto de datos

Usando el paquete sklearn de python, los datos se dividieron aproximadamente en las siguientes proporciones:

- conjunto de entrenamiento: 70 %
- conjunto de validación: 20 %
- conjunto de evaluación: 10 %

IV. Visualización con embebimiento BH TSNE

Se implementó la técnica Distributed Stochastic Neighbor Embedding, T-SNE. Para esto se implementó el módulo tsne que se puede encontrar en el siguiente repositorio[2] donde se puede profundizar más al respecto.

Luego de implementar el embebimiento, se observó que la metodología no pudo generar una visualización alternativa de los datos para identificar posibles clusters tal como se aprecia en la figura 2

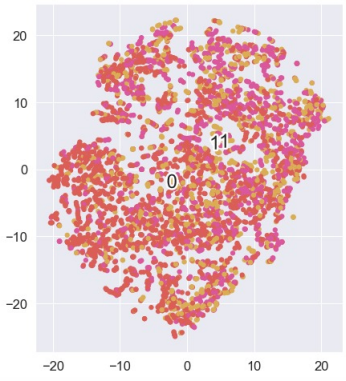

Figura 2: Reducción de dimensionalidad

El conjunto resultante del embebimiento fue utilizado como un conjunto de datos alternativo para entrenar los algoritmos objetos de estudio y sus resultados se mostrarán para cada uno de ellos.

V. Entrenamiento de modelos y resultados

Para efecto de este trabajo y al ser un problema de clasificación, se consideraron los siguientes algoritmos:

- árboles de clasificación
- bosque aleatorio
- Regresión logística
- Máquina de soporte vectorial

Se consideraron los hiper parámetros asociados a cada algoritmo para acotar la cardinalidad del espacio de modelos posibles para cada algoritmo. Dichos parámetros fueron explorados en una metodología de fuerza bruta para ver el desempeño del algoritmo/modelo.

Adicional a esto, la garantía probable de aprendizaje se calculó como

$$n \geq \epsilon^{-1}(ln(|h|) + ln(\delta^{-1})) \quad (1)$$

donde n es la cantidad de datos mínimos para tener una garantía probable de aprendizaje, ϵ es error real deseado, δ es el error de entrenamiento deseado y $|h|$ es la cardinalidad de las combinaciones posibles a considerar para la exploración de la estructura de los modelos para cada algoritmo.

El cálculo del n_{optimo} se especifica para cada algoritmo.

V-A. *Árbol de Clasificación*

Se tomaron los siguientes hiper parámetros con los valores especificados:

- máxima profundidad: $\{1, 2, 3, 4, 5\}$
- número mínimo de hojas: $\{1, 2, ..., 10\}$

La combinación de estos parámetros da una aproximación de $|h|$. Este será usado más adelante.

Garantía probable de aprendizaje y tamaño óptimo de la muestra

De acuerdo a la ecuación 1, la garantía probable de aprendizaje es

$$n \geq 0{,}4^{-1}(ln(50) + ln(0{,}15^{-1})) = 14{,}52$$

Para obtener el n_{optimo} se usó la siguiente expresión:

$$n_{optimo} \geq \frac{ln(2)}{2\epsilon^2} * ((2^k - 1)(1 + log_2 N) + 1 + ln(\delta^{-1}))$$

Siendo k la profundidad máxima de cada árbol. Al ser un parámetro que impacta el crecimiento de n_{optimo} de manera exponencial, se considerará el peor escenario tenido en cuenta en la definición de los hiper parámetros. Esto es, $k = 5$. De esta manera, el valor de n_{optimo} es

$$\frac{ln(2)}{2 * 0{,}4^2} * ((2^k - 1)(1 + log_2(4931)) + 1 + ln(0{,}15^{-1})) = 3397{,}81$$

Resultados de entrenamiento y análisis

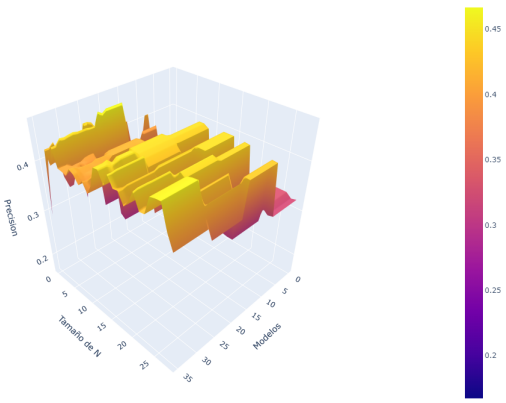

Figura 3: Precisión de los modelos de árboles de clasificación

[2] https://lvdmaaten.github.io/tsne/

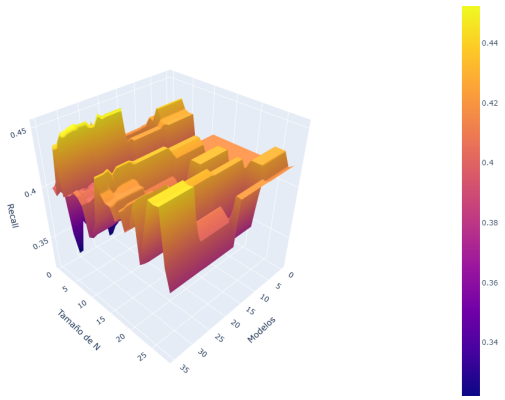

Figura 4: Recall de los modelos de árboles de clasificación

Una vez se explora de forma exhaustiva el desempeño del árbol de clasificación con diferentes configuraciones en sus hiper parámetros, se observa que si bien se mantienen cerca de una precisión y recall de 0.4, este nivel es demasiado malo (ver figuras 3 y 4

En general, no se esperaba que los árboles tuvieran un buen desempeño. Esto se puede dar porque las clases no estén bien balanceadas, lo que incumple uno de los supuestos para que este algoritmo funcione de la manera correcta.

En las figuras 5 y 6 se ve el desempeño del algoritmo al tratar de clasificar la muestra de prueba, y una vez mas evidencia un mal desempeño. No supera el umbral del 60 % deseado en el error real.

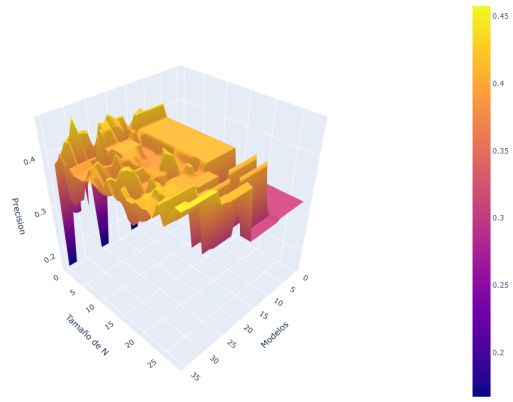

Figura 5: Precisión de los modelos de árboles de clasificación bajo reducción de dimensión

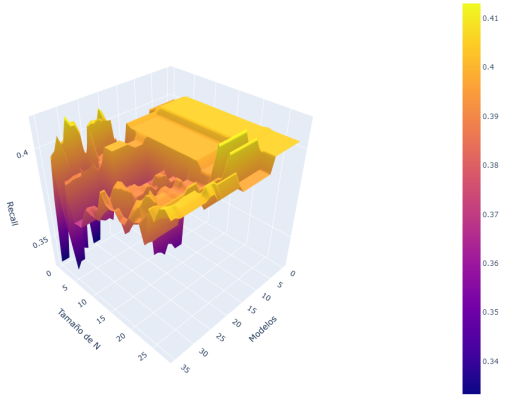

Figura 6: Recall de los modelos de árboles de clasificación bajo reducción de dimensión

V-B. Bosque Aleatorio

Para el bosque aleatorio se tomaron los siguientes hiper parámetros:

- número de árboles: $n_i = n_{i-1} + 50$ con $n_1 = 50$ para $i = 1 : 9$
- máxima profundidad de los árboles: $\{1, 2, 3, 4, 5\}$
- número mínimo de hojas: $\{1, 2, ..., 10\}$

Garantía probable de aprendizaje

De manera similar al árbol de clasificación, se obtiene que la garantía probable de aprendizaje corresponde a $n \geq 53{,}88$

En este caso, por la estructura del bosque aleatorio, el tamaño óptimo de la muestra no puede determinarse, por lo que se asume infinito. En este sentido, la exploración únicamente se hace partiendo de la garantía probable de aprendizaje.

Resultados de entrenamiento y análisis

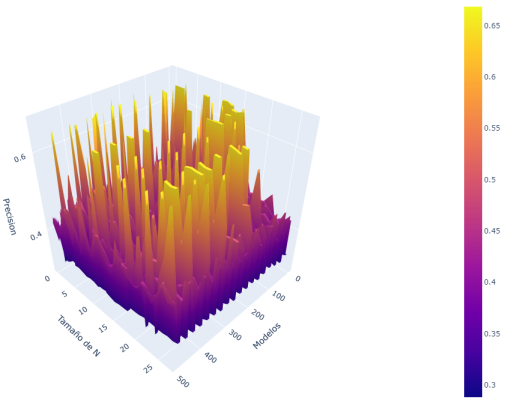

Figura 7: Precisión de los modelos de bosque aleatorio

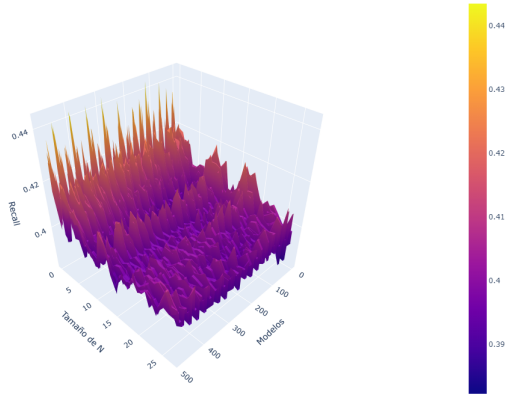

Figura 8: Recall de los modelos de bosque aleatorio

En el caso del bosque aleatorio, es curioso que aunque hay modelos que alcanzan niveles de precisión de 0.65, es bastante inestable e inconsistente (ver figura 7, siendo muy interesante que el incremento del tamaño de la muestra para entrenamiento no afecta la estabilidad en la capacidad de aprendizaje del bosque aleatorio. Incluso se aprecia como el incremento del tamaño de la muestra para los diferentes modelos hace que se recupere menos información relevante (ver figura 8

De la misma manera, la reducción de dimensión presenta el mismo comportamiento para el algoritmo. No hay un cambio significativo respecto al entrenamiento bajo la dimensión original del problema.

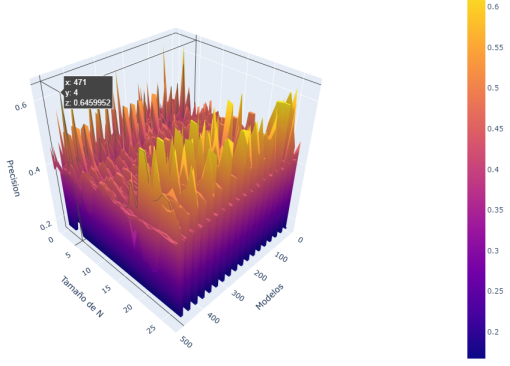

Figura 9: Precisión de los modelos de bosque aleatorio bajo reducción de dimensión

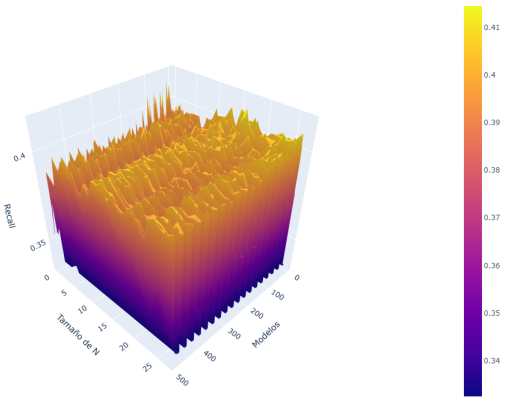

Figura 10: Recall de los modelos de bosque aleatorio bajo reducción de dimensión

V-C. Regresión Logística

Ya que el objetivo de este trabajo no es explorar la selección de variable se toman como hiper parámetros el número posible de regresores. Teniendo en cuenta que se tienen 65 regresores y que para clasificadores lineales su dimensión VC es igual $D + 1$, siendo D el número de regresores máximo para el modelo, se procede entonces con una búsqueda aleatoria del impacto en la precisión del modelo al variar el los regresores a implementar en el algoritmo.

Resultados de entrenamiento y análisis

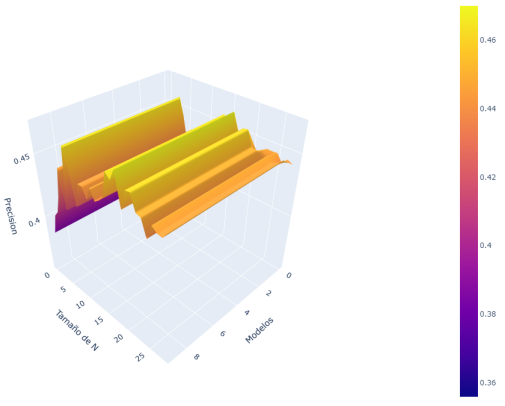

Figura 11: Precisión de los modelos de regresión logística

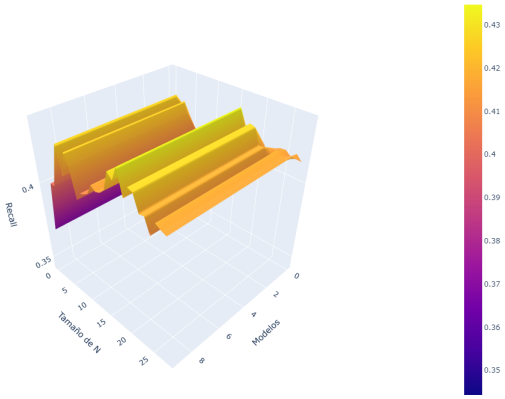

Figura 12: Recall de los modelos de regresión logística

El desempeño de la regresión logística no es muy diferente a los anteriores resultados. Si bien es más estable que los algoritmos anteriores, no supera un umbral deseado de precisión y recall (ver figuras 11 y 12. La gran diferencia que se aprecia en este caso, es el desempeño al reducir la dimensionalidad. En este caso, el algoritmo se ve altamente impactado en la reducción de dimensionalidad al medir el efecto del incremento en el tamaño de la muestra. Esto se aprecia fácilmente en la figura 13.

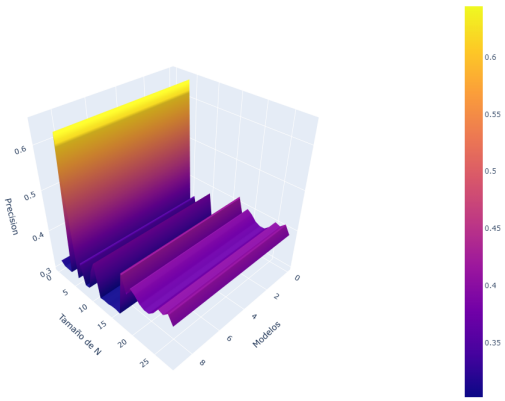

Figura 13: Precisión de los modelos de regresión logística bajo reducción de dimensión

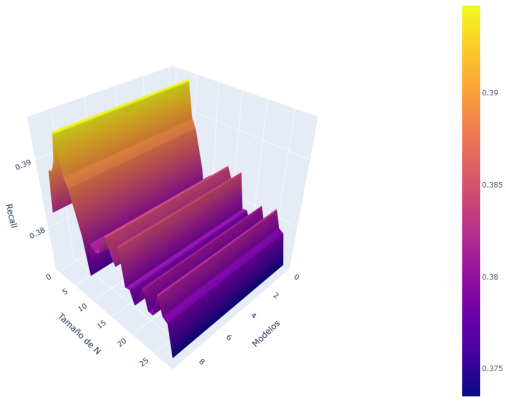

Figura 14: Recall de los modelos de regresión logística bajo reducción de dimensión

V-D. Máquina de Soporte Vectorial

Para la máquina de soporte vectorial únicamente se tienen en cuenta los diferentes kernels que se pueden implementar. Estos son linear, poly, rbf y sigmoid.

Garantía probable de aprendizaje y tamaño óptimo de la muestra

Usando la misma ecuación 1, es posible obtener que $n \geq 21{,}88$. Ahora bien, dado que uno de los kernels es el lineal, este será usado como referente para calcular el tamaño óptimo de la muestra incluso para los demás kernels. De esta manera, en términos del clasificador lieal, el tamaño óptimo de la muestra será $D + 1$, siendo D el número de variables explicativas.

Resultados de entrenamiento y análisis

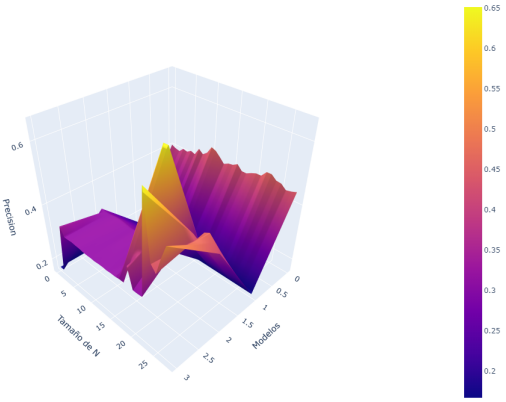

Figura 15: Precisión de los modelos de SVM

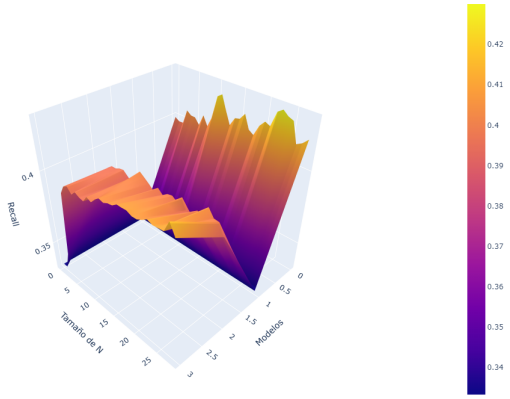

Figura 16: Recall de los modelos de SVM

Siendo el último de los algoritmos a revisar, tampoco se observa una diferencia significativa en el desempeño de la precisión y el recall de la máquina de soporte vectorial. En este caso, al igual que en bosque aleatorio, se observaron modelos que lograron superar el umbral del 60 % en la precisión, pero no fue así para el recall del algoritmo.

VI. Conclusiones

La principal conclusión es que el problema planteado no puede ser solucionado a partir de los 4 algoritmos presentados en el trabajo. Vale la pena explorar otras metodologías o revisar si es un problema que se puede aprender.

Al tratarse de un problema en el que se trata de clasificar un día particular como un día previo a un movimiento significativo del mercado al futuro, se podría pensar en un algoritmo de aprendizaje orientado a series de tiempo, donde los factores que se usen en términos de regresores correspondan con los descritos en la presentación del problema.