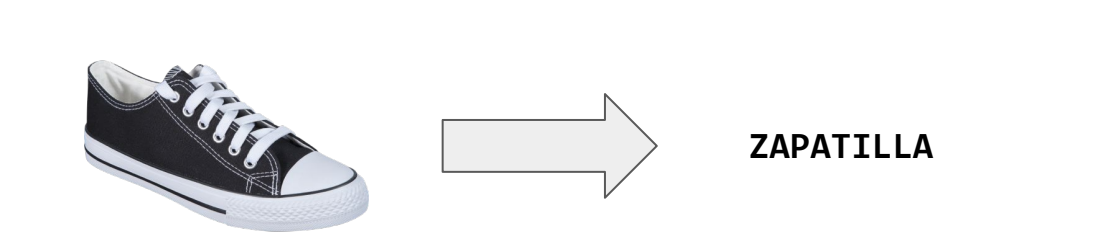




Introducción y motivación

En la actualidad, muchos sitios de comercio en línea dependen de trabajadores manuales para etiquetar y clasificar sus productos, lo que puede resultar lento y costoso. Una aplicación de clasificación de imágenes automatizada podría ayudar a agilizar este proceso y a mejorar la precisión de las clasificaciones, lo que podría beneficiar tanto a los sitios de comercio en línea como a sus clientes.



Como solución se han implementado los siguientes métodos:

1. SVM (Support Vector Machine) usando OVO y OVA
2. MLP (Multilayer Perceptron) de Scikit-learn
3. MLP (Multilayer Perceptron) de Tensorflow
4. Adaboost ensembler usando OVA
5. RandomForest

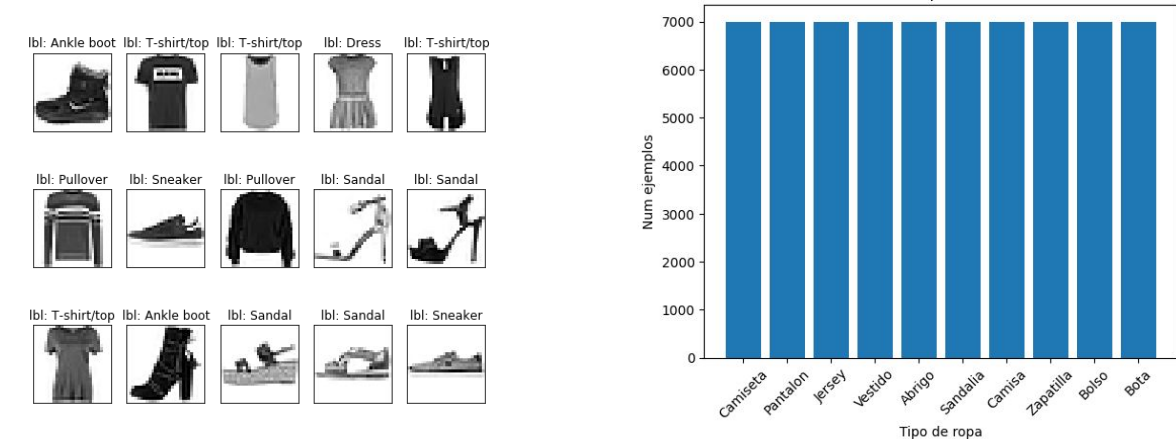
Conjunto de datos

Para este proyecto, hemos elegido utilizar el dataset MNIST fashion debido que cuenta con 10 clases diferentes, lo que lo hace adecuado para un problema de clasificación de imágenes. Además, las imágenes tienen un tamaño pequeño de 28x28, lo que significa que son fáciles de procesar y requieren menos cómputo que imágenes más grandes.

Es importante señalar que es posible utilizar otros datasets para este proyecto aunque es importante tener en cuenta que el cambio de dataset podría requerir más cómputo y afectaría al tiempo de entrenamiento y a la precisión del modelo.

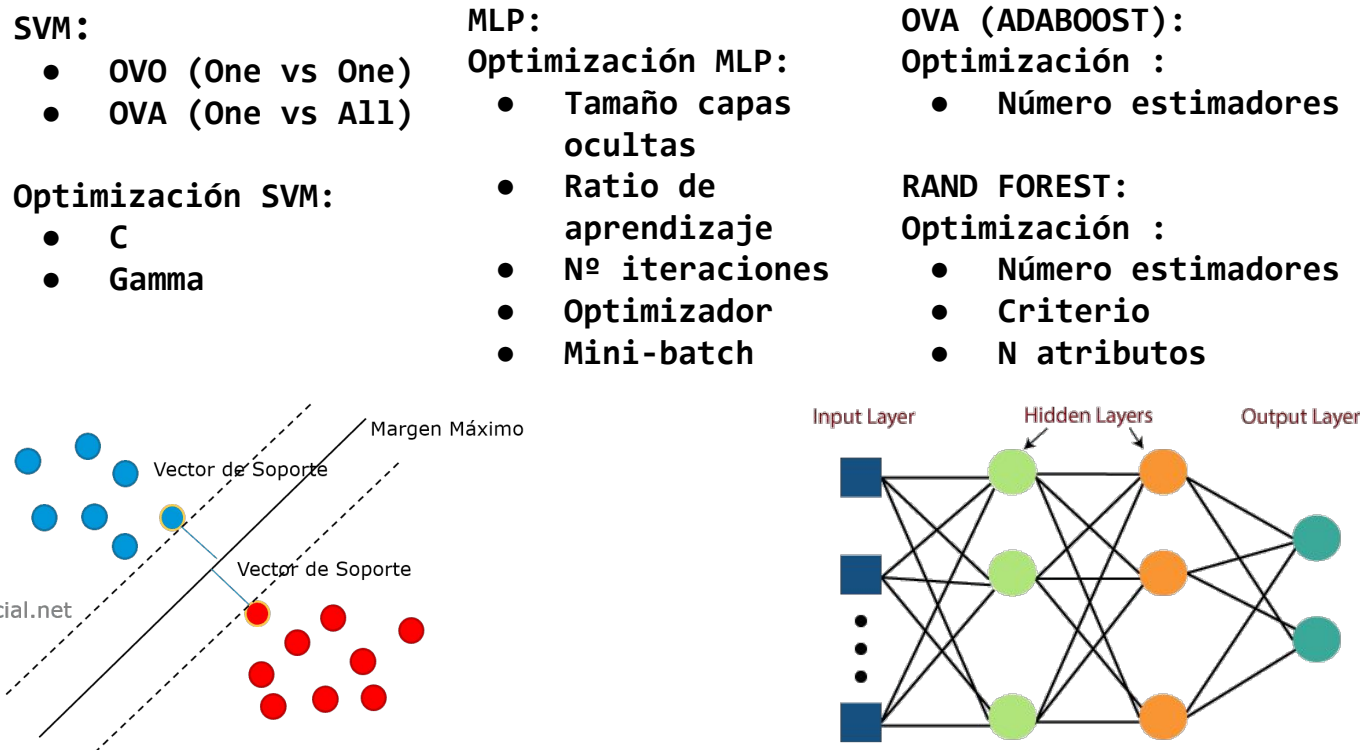
Número de ejemplos: 70000 (repartidos entre 10 clases)

- Train: 60000
- Test: 10000



Marco experimental

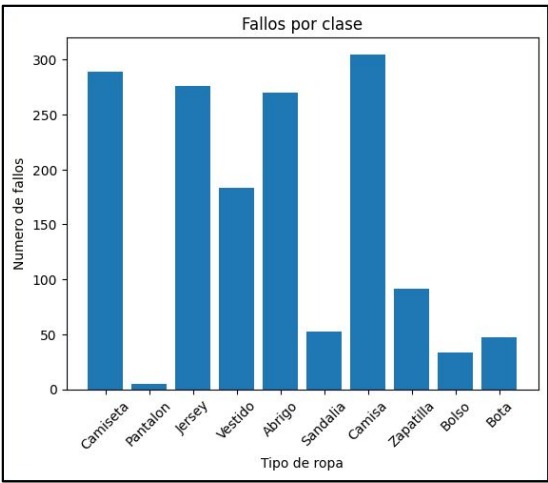
Para este proyecto, hemos utilizado tres modelos de aprendizaje automático diferentes: máquinas de vectores de soporte (SVM), redes neuronales multinivel (MLP) y un ensembler de adaboost. Además, hemos utilizado un modelo MLP implementado en TensorFlow, ya que al utilizar la GPU este modelo es capaz de entrenarse más rápidamente. No hemos usado Naive Bayes ya que se basa en que los atributos sean independientes entre sí, cosa que no sucede en nuestro caso ya que estamos tratando con píxeles.



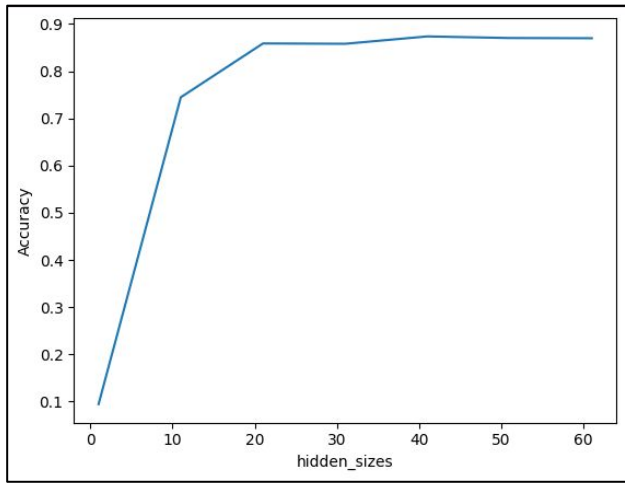
Estudio experimental

RESULTADOS POR MODELO		Acc	Tiempo
SVM	OVO --> Gamma 'scale' C = 1.0	0.889	6.12 min
	OVA --> Gamma 'scale' C = 1.0	0.889	5.8 min
MLP	Capas ocultas --> 121 Alpha --> 0.01 Iteraciones --> 250 Optimizador --> Adam Modo --> por Lotes	0.864	5.25min
	Capas ocultas --> (100,60,20) Alpha --> 0.01 Iteraciones --> 250 Optimizador --> Adam Modo --> mini-batch de 1000	0.872	37.5 seg (tensorflow)
RANDOM FOREST	Número de estimadores = 150 Criterio = geany N atributos = sqrt	0.885	15,9 seg
OVA(ADABOOST)	Número de estimadores = 150	0.853	39 min

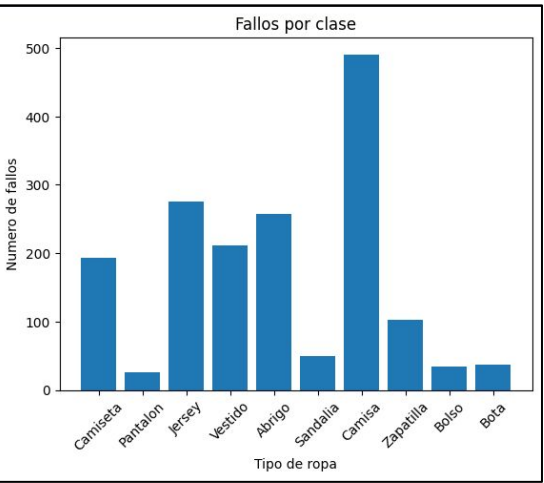
Fallos por clase SVM:



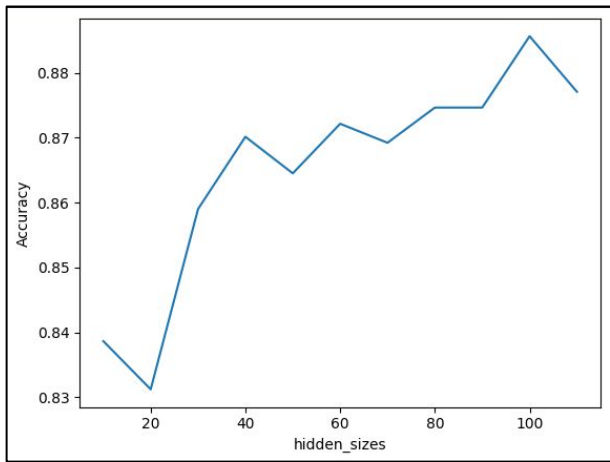
Accuracy respecto a una única capa oculta MLP:



Fallos por clase MLP:



Accuracy respecto a variación de la primera capa oculta MLP (X,30,20):



Como modelo final podríamos elegir tanto MLP, SVM o randomForest ya que son los que obtienen mejores resultados. Con ninguno logramos pasar del 90% de precisión ya que debido a la baja resolución de las imágenes y a que hay 4 clases bastante parecidas (camisa, camiseta, abrigo, jersey), cometemos bastantes fallos(los gráficos mostrados arriba). El SVM falla las 4 clases más o menos por igual, pero el MLP falla sobre todo en la clase camisa. Por lo tanto, puede que el MLP sea más interesante ya que con centrarse en las camisas se reducen en gran medida los errores de clasificación. Aunque si tenemos un dataset muy grande con muchos atributos randomForest funciona muy bien ya que da muy buenos resultados en muy poco tiempo. En general, ambos son modelos que generalizan bastante bien y que podrían usarse en un entorno real.

Conclusiones y líneas futuras

En conclusión, hemos desarrollado una aplicación de clasificación de imágenes utilizando cuatro modelos de aprendizaje automático diferentes (un modelo MLP implementado en TensorFlow). Hemos evaluado estos modelos utilizando el dataset MNIST fashion y hemos comparado sus resultados: un OVA de adaboost nos permite obtener resultados decentes, aunque peores que con MLP o SVM, eso si, es batante lento; los MLP y el SVM dan resultados muy parejos, ambos muy buenos, tanto en tiempo como en precisión; y RandomForest consigue una alta precisión en un tiempo muy reducido a comparación a los demás probados.

En un futuro, podría ser interesante explorar la posibilidad de utilizar videos en lugar de imágenes y modelos más complejos usando convoluciones para la clasificación de imágenes. Esto podría permitir una mayor precisión en la clasificación y podría ser útil en aplicaciones que involucren objetos en movimiento, como el reconocimiento de ropa en una tienda.