PROYECTO FINAL – INTELIGENCIA ARTIFICIAL
ORANGES VS GRAPEFRUITS

AUTORES:
Ángela María Benítez López
Javier Andrés Bernal Castañeda
Juan Sebastián Barbosa Rivas

PRESENTADO A:
Francisco Carlos Calderón Bocanegra, Ing

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C. 2020

1. Introducción

La inteligencia artificial (IA) hace posible que las máquinas aprendan de la experiencia, se ajusten a nuevas aportaciones y realicen tareas como seres humanos. La mayoría de los ejemplos de inteligencia artificial que se escuchan hoy en día, desde computadoras que juegan ajedrez hasta automóviles de conducción autónoma, recurren al aprendizaje profundo y al procesamiento del lenguaje natural, pero también existen otra ramas de la inteligencia artificial tan importantes como las ya mencionadas, como es el Aprendizaje de Maquina y la Inteligencia Computacional, las cuales permiten que las computadoras pueden ser entrenadas para realizar tareas específicas procesando grandes cantidades de datos y reconociendo patrones en ellos [2].

Dentro del Aprendizaje de Maquina se destaca el método supervisado, el cual crea un modelo matemático que busca explicar unas etiquetas de entrada o de salida a partir de un conjunto de características de entrada, dentro de este se encuentran clasificadores como la Regresión Logística, Vecinos Más Cercanos (KNN) y Maquinas de Vectores de Soporte (SVM). Por parte de la Inteligencia Computacional se encuentran las Redes Neuronales, que también son capaces de realizar las tareas propias de la clasificación.

Ahora bien, se espera que mediante la implementación de los clasificadores mencionados anteriormente se pueda dar solución al problema de clasificación de frutas, en particular entre naranjas y pomelos, de acuerdo a los resultados obtenidos y de las métricas de evaluación escogidas para cada clasificador, se determinará cuál es la solución más eficiente al problema planteado.

Su principal aplicación está destinada para la industria de alimentos, ya que a simple vista la tarea de separación de naranjas y pomelos puede resultar bastante obvia para un ser humano, existen errores de observación, por esta razón, se busca los alcances futuros del proyecto permitan ahorrar tiempo y evitar errores durante el procesamiento de estas frutas.

2. Desarrollo y Resultados

Para el desarrollo de este proyecto se trabajó en Colab, el cual es un servicio cloud, basado en los Notebooks de Jupyter, que permite el uso gratuito de las GPUs y TPUs de Google [4]. Además, el dataset con el que se trabajó se obtuvo a través de la página Kaggle, siendo este de uso gratuito, este dataset contiene 6 columnas, la primera corresponde al nombre de la fruta, la segunda al diámetro, la tercera al peso, la cuarta al valor en RGB (0 a 255) del color rojo, la quinta al valor en RGB (0 a 255)

del color verde y finalmente, el valor en RGB (0 a 255) del color azul de la fruta. A continuación, se muestra gráficamente las características de peso y diámetro para naranjas y pomelos.

Figura 1. weight vs diameter - naranjas y pomelos

En la Figura 1 se hace notorio el problema de clasificación a realizar, ya que como se observa, existe una región en la que se podría clasificar erróneamente una naranja como un pomelo o viceversa. Es aquí donde toman importancia las características restantes del dataset, correspondientes al color de cada fruta en R-G-B.

Como primer paso se tomó la primera columna del nombre, la cual contenía las palabras orange y grapefruit, y se reemplazó por sus correspondientes etiquetas, en este caso 1 y 0 respectivamente. El siguiente paso fue el acondicionamiento de los datos, esto consistió en hacer una partición de estos, donde se tomó el 30% para el conjunto de validación (test) y el 70% para el conjunto de entrenamiento (train), seguido de esto se procedió a hacer la escalización de datos de entrenamiento.

Además, las métricas de evaluación de desempeño que se implementaron para cada clasificador fueron el Coeficiente de correlación de Matthews, la Exactitud (Accuracy) y la curva ROC.

Donde el Coeficiente de correlación de Matthews tiene en cuenta los positivos y los negativos verdaderos y falsos, por lo tanto, esta métrica se considera equilibrada y se puede utilizar si las clases tienen tamaños diferentes. Si su resultado es 1, es una perfecta predicción, si es cero no es mejor que una predicción aleatoria y si es -1 es una muy mala predicción [7].

La exactitud es la cercanía de las mediciones a un valor específico, por lo tanto, si su resultado es 1 o cercano a 1, significa que es una buena predicción, de lo contrario, si se acerca a cero, es una pésima predicción.

Donde la ROC corresponde a un diagrama gráfico, en específico una curva, la cual representa la sensibilidad, es decir la tasa de verdaderos positivos en función de la especificidad que significa la tasa de falsos positivos. En la figura 2 se puede observar que entre más cerca se encuentre la curva al punto superior izquierdo (0,1) más se acerca a un modelo perfecto, mientras que si se obtiene una diagonal positiva se considera un mal modelo.

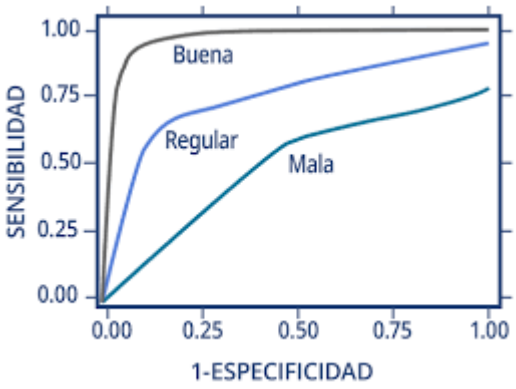

Figura 2 ROC [2]

2.1 Regresión Logística

Con los datos listos, el primer clasificador que se implementó fue el de Regresión Logística, con ayuda de la librería *sklearn* se utilizó la clase *LogisticRegression*, la cual nos da como resultado un valor que indica que tan probable es que sea una naranja o un pomelo, dependiendo de sus características. Para entender mejor el funcionamiento de la Regresión logística se presenta la gráfica de la función logística o función sigmoide, la cual se observa que esta acotada entre 0 y 1, por lo tanto, su valor mínimo es 0 y el máximo es 1, se supone que los valores menores de 0.5 corresponden a la clase 0 y los superiores a 0.5 a la clase 1.

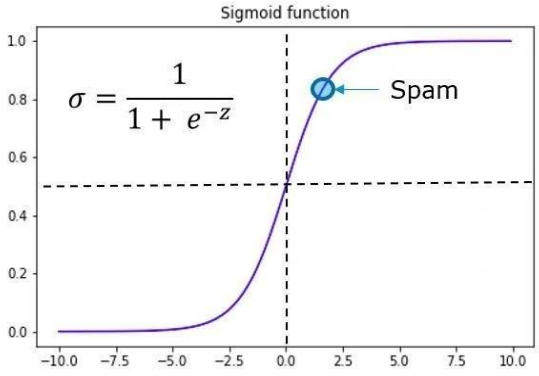

Figura 3 Función Sigmoide [5]

Teniendo en cuenta lo anterior, para los primeros datos del dataset se obtuvo un valor de 0.98942227 de que sea naranja y 0.01057773 de que sea pomelo, por lo tanto, se determina que la fruta es una naranja.

Los resultados que se obtuvieron con las métricas de desempeños son los siguientes.

- Se obtuvo un coeficiente de correlación de Matthews igual a 0.8350525556791627, el cual es un valor cercano a 1, representando así un buen desempeño.

- Se obtuvo una exactitud de 91.73333333333333, el cual es valor muy cercano a 1, igual que en la métrica anterior representa un buen desempeño.
- Se obtuvo la gráfica de la curva ROC, la cual se observa en la figura 4 que es muy cercana al punto (0,1) con un área de 0.97, esto indica que es un modelo de clasificación bastante bueno con un alto desempeño.

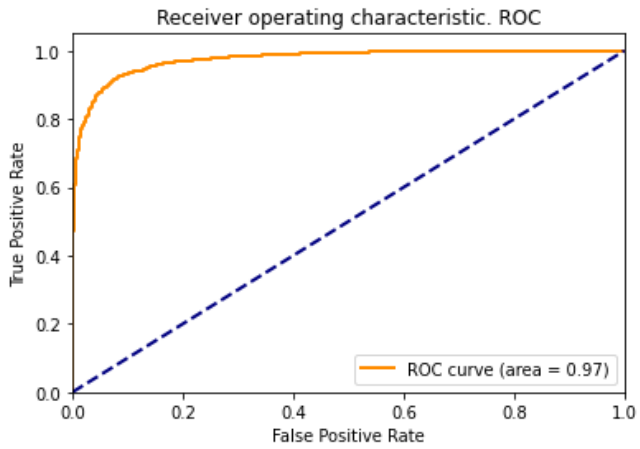

Figura 4 ROC Regresión Logística

2.2 KNN

Como segundo método de clasificación se empleó KNN haciendo uso de la clase *KNeighborsClassifier* de *sklearn.neighbors*. Este método se basa en hallar el grupo de k muestras en el conjunto de entrenamiento que estén más cercanas a una muestra desconocida haciendo uso por ejemplo de las funciones de distancia [8]. De esta manera se manera se puede llegar a clasificar una muestra basándose en las características de sus k vecinos más cercanos [9].

Los hiperparámetros a variar para hallar el modelo óptimo de clasificación fueron el *n_neighbors* que corresponde al número de vecinos más cercanos usados para clasificar una muestra, y *metric* que indica al modelo la función de distancia a usar para hallar los k vecinos más cercanos. Para esto se estableció un rango de k y un conjunto de métricas para entrenar el modelo con cada una de la combinación entre estos dos parámetros y hallar la combinación para la cual el modelo tuviera el máximo score.

Para establecer el rango de k se estableció como valor máximo la raíz cuadrada del número de muestras del conjunto de entrenamiento, para luego generar un arreglo que contenga los números naturales entre 1 y 83. Las métricas con las que se entrenó el modelo fueron: euclidean, manhattan, chebyshev, minkowski, seuclidean, mahalanobis y haming.

Luego de iterar el modelo para cada combinación posible se obtuvo en score máximo de 0.981, con k=1 y métrica de distancia mahalanobis. Con estos hiperparámetros hallados se entrenó nuevamente

el modelo de KNN del que se obtuvieron los siguientes resultados para las métricas de desempeño aplicadas a los resultados sobre el conjunto de validación

- Se obtuvo un coeficiente de correlación de Matthews (MCC) igual a 96.2096 %, el cual es un valor cercano a 1, representando así un buen desempeño.
- Se obtuvo una exactitud (ACC) de 98.1 %, el cual es valor muy cercano a 1, igual que en la métrica anterior representa un buen desempeño.
- Se obtuvo la gráfica de la curva ROC, la cual se observa en la figura 5 muy cercana al punto (0,1) con un área de 0.97, esto indica que es un modelo de clasificación bastante bueno con un alto desempeño

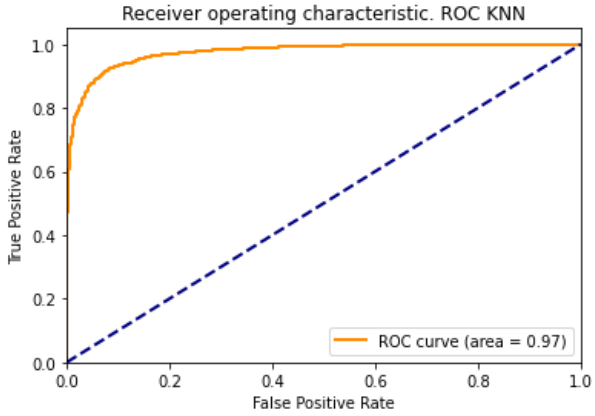

Figura 5.ROC KNN

Para este modelo se aplican dos maneras adicionales para la evaluación de sus resultados. Primero se hará uso de la matriz de confusión la cual indica el número de true positives, false negatives, false positives y true negatives a partir de los resultados sobre el conjunto de validación.

	P	**N**
P	1471	18
N	39	1472

Tabla 1 Matriz de confusion KNN

En la anterior tabla se observa como los valores de TP y TN son significativamente mayores en relación con los valore de FP y FN, lo cual tiene sentido si se contrastan con los valores obtenidos previamente para la ROC, MCC y ACC

Por último, se analiza el reporte de clasificación:

	precisión	recall	f1-score	support
0 (Grapefruits)	0.97	0.99	0.98	1489
1 (Oranges)	0.99	0.97	0.98	1511

Tabla 2 Reporte de clasificación KNN

Aquí se observa que la precisión en la clasificación fue mayor para las muestras correspondientes a pomelos, sin embargo, se observa con la métrica recall que el 99% de elementos de la clase pomelos se clasificaron como tal. La métrica support muestra que el número de elementos por clase son relativamente cercanos, lo cual garantiza un equilibrio entre las muestras del conjunto de validación

2.3 SVM

El tercer método de clasificación empleado es SVM (máquinas de vectores de soporte). Su funcionamiento se basa en hallar un hiperplano de separación a partir de los datos de entrenamiento que sea capaz de distinguir entre las clases a clasificar, en este caso Oranges y Grapefruits [8]. Para establecer los límites de separación entre clases el modelo utiliza funciones de kernel, los 4 tipos más utilizados son lineal, polinómico, RBF o sigmoide [9]. Lo anterior corresponde al hiperparámetro a manipular para encontrar el modelo de SVM óptimo para el problema de clasificación aquí planteado. Así pues, se probó este método con cada uno de los 4 kernels para hallar aquel cuyas predicciones, sobre el conjunto de validación, al ser evaluadas con las métricas MCC, ACC y MOC entreguen los mejores valores.

Kernel	**MOC**	**ACC**	**ROC**
Lineal	0.909	0.954	0.98
Polinomial - grado1	0.859	0.9296	0.98
Polinomial - grado 2	0.2939	0.624	0.68
Polinomial – gado 3	0.833	0.916	0.97
RBF	0.859	0.9293	0.97
Sigmoide	0.764	0.882	0.95

Tabla 3 Métricas para los diferentes tipos de kernel

A partir de los datos registrados en la tabla 3. Se establece que el kernel que mejor responde al problema de clasificación aquí tratado, es el lineal, lo cual tiene sentido si se considera que se están clasificando las muestras en 2 clases, lo cual hace de este un problema sencillo de clasificación para el método SVM

Se observa que con el kernel Lineal se tiene en cuanto a métricas:

- Un coeficiente de correlación de Matthews (MCC) igual a 90.96 %, el cual es un valor cercano a 1, representando así un buen desempeño.
- Se obtuvo una exactitud (ACC) de 95.4 %, el cual es valor muy cercano a 1, igual que en la métrica anterior representa un buen desempeño.
- Se obtuvo la gráfica de la curva ROC, la cual se observa en la figura 6 muy cercana al punto (0,1) con un área de 0.98, esto indica que es un modelo de clasificación bastante bueno con un bajo número de false positives y false negatives

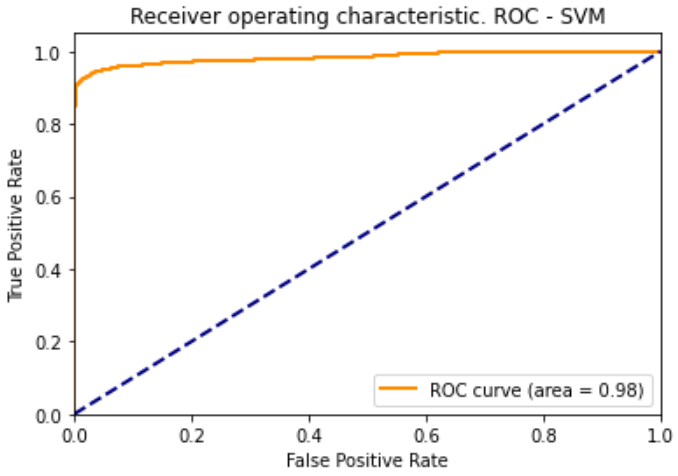

Figura 6 ROC SVM

2.4 Redes neuronales

El ultimo método empleado para clasificación son las redes neuronales como clasificador, las redes neuronales reciben su nombre por la similitud que existe entre este tipo de modelo y una neurona real, la idea básica de su funcionamiento es utilizar los valores de las entradas, inputs, y multiplicarlos por ciertos pesos para producir un valor de salida. Este valor se ira repitiendo por las capas ocultas de la neurona hasta llegar a la capa de salida en donde se producirá el resultado de si es una naranja o un pomelo. Para este caso uso de la librería MultilayerPerceptron de sklearn.

Como se ha mencionado anteriormente, se utilizó el 70% de los datos del dataset para el conjunto de entrenamiento y el 30% restante para el conjunto de validación, como se ha visto en clase se utilizó la función de activación "relu" para lograr una mayor velocidad en el proceso de entrenamiento. Para optimizar el número de neuronas de cada capa oculta de la red se utilizaron dos ciclos *for* anidados variando el número de neuronas presente en cada capa oculta hasta poder optimizar el modelo. El procedimiento anterior dio como resultado que cada capa oculta debe tener 8 neuronas para lograr un *Accuracy Score* de 99.63%.

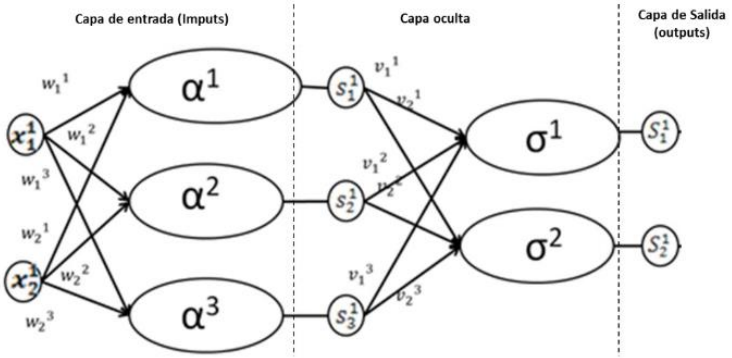

Figura 7 Red neuronal tomada de [10]

3. Conclusiones

- Como se pudo observar cada uno de los clasificadores implementados cuenta con medidas de desempeño bastante altas, llegando casi al 100%, sin embargo, si se considerase implementar alguno de estos algoritmos en la práctica, el clasificador KNN por su peso computacional podría ser descartado, mientras que la clasificación por Redes Neuronales se evitaría hacer, debido a un posible sobre entrenamiento del modelo. Por lo que la clasificación por regresión logística o SVM podrían ser implementadas en la vida real.
- Observando los resultados obtenidos en este trabajo, es importante resaltar el papel tan importante que juega la escalización de los datos, esto ocurre debido a que los rangos de la codificación RGB tienen valores muy grandes a comparación del diámetro o el peso (estos dos no llegan a pasar de las decenas en la mayoría de los casos) por lo que fue de suma importancia usar la escalización para que los datos se encontraran en el rango de valores que permitiera a los clasificadores hacer la separación
- Para cada clasificador se implementó varias métricas de desempeño, esto con el fin de comparar cual es el clasificador más adecuado para nuestro problema, al hacer dicha comparación se determinó que el mejor clasificador es el de Vecinos más Cercanos (KNN), ya que presento uno de los valores más altos en el coeficiente de correlación de Matthews y en la exactitud.

4. Referencias

[1] Dataset Oranges vs Grapefruit, [Online]. Available: https://www.kaggle.com/joshmcadams/oranges-vs-grapefruit

[2] Software y Soluciones de Analítica. ¿Qué es la Inteligencia Artificial? [Online]. Available: https://www.sas.com/es_co/insights/analytics/what-is-artificial-intelligence.html

[3] Métodos de Clasificación. [Online]. Available: https://bookdown.org/content/2274/metodos-de-clasificacion.html

[4] Datahack. Introducción a google colab para data science. (2019, junio 17), [Online]. Available: https://www.datahack.es/blog/big-data/google-colab-para-data-science/#:~:text=Colab%20es%20un%20servicio%20cloud,disponible%20para%20R%20y%20Scala.

[5] J, Martinez. Regresión Logística para Clasificación. (2020, septiembre 21). [Online]. Available: https://www.iartificial.net/regresion-logistica-para-clasificacion/

[6] Universidad Nacional Autónoma de México. Utilidad y Validez de las Pruebas Diagnósticas. [Online]. Available: http://132.248.48.64/repositorio/moodle/pluginfile.php/1833/mod_resource/content/7/contenido/index.html

[7] Coeficiente de correlación de Matthews [Online]. Available: https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

[8] Qian, Y., Zhou, W., Yan, J., Li, W., & Han, L. (2015). Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. *Remote Sensing*, *7*(1), 153-168.

[9] Xu, H., Zhou, J., G Asteris, P., Jahed Armaghani, D., & Tahir, M. M. (2019). Supervised machine learning techniques to the prediction of tunnel boring machine penetration rate. *Applied Sciences*, *9*(18), 3715.

[10] Matson Hernández, J. (2017).Facultad de Ciencias Económicas y Administrativas Maestría en Economía1Redes Neuronales para Clasificación:Una aplicación al caso de Riesgos Laborales en Colombia. Disponible en: https://repository.javeriana.edu.co/bitstream/handle/10554/37845/MatsonHernandezCamiloEduardo2017.pdf?sequence=1&isAllowed=y