

**Ciencia de Datos Aplicada – MINE4101**  
**Maestría en Ingeniería de la información**

**Taller 2: Supermercado Inteligente**  
Informe de Insights

**Grupo:**

María Catalina Ibáñez Piñeres  
Lina María Gómez Mesa



Universidad de Los Andes

Noviembre 17, 2024

## **Contents**

1.	Entendimiento y preparación de los datos .....	3
2.	Entrenamiento del modelo de Machine Learning.....	4
3.	Análisis de resultados del modelo.....	5
4.	Generación de valor .....	6
5.	Insights.....	Error! Bookmark not defined.

## 1. Entendimiento y preparación de los datos

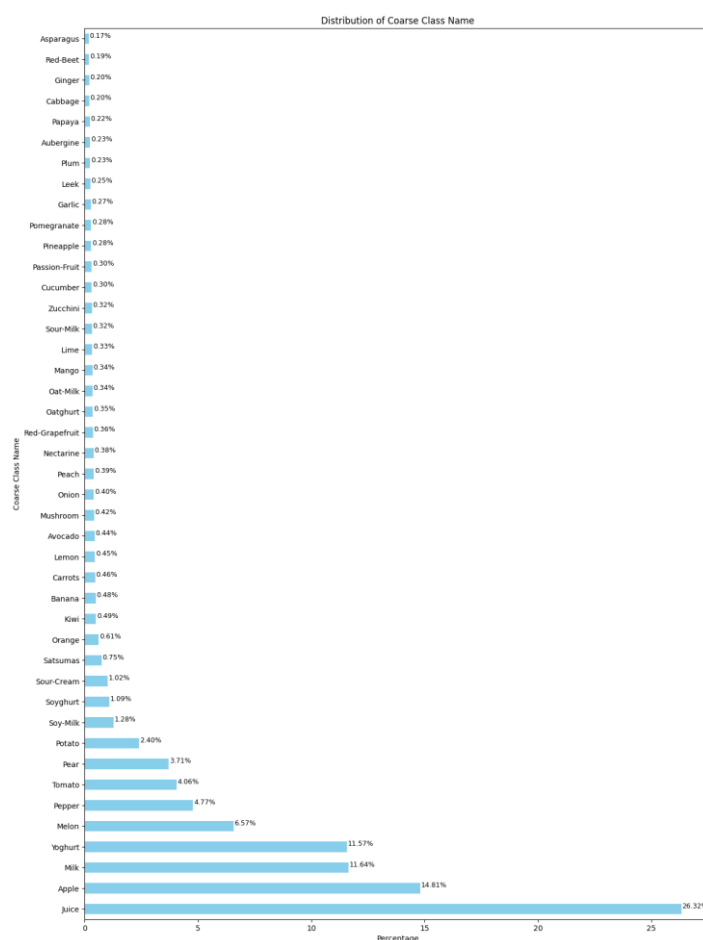
Con el fin de realizar el entendimiento de los datos inicial, se analizaron las imágenes proporcionadas por el supermercado. Los análisis realizados se realizaron por clases en el set de entrenamiento con la columna llamada *Coarse\_class\_id*. En primer lugar, con respecto al *dataset* en general y sus dimensiones en el set de entrenamiento hay 2640 imágenes, el de prueba tiene 2485 imágenes y el de validación 296. En total, el set de entrenamiento y test tienen 43 clases en la columna *Coarse\_class\_name* y el de validación tiene 37 clases de las cuáles le faltan Nectarine, Papaya, Plum, Arla-Sour-Milk, Alpro-Fresh-Soy-Milk y Garlic. De las imágenes se tenían dimensiones iniciales diferentes las cuales eran (348,348,3), (464, 348,3), (348,464,3).

Con respecto a indicadores importantes se consideró en general que estadísticas descriptivas de la intensidad de los pixeles pueden brindar información de las diferencias entre las clases. Estas se analizaron en el espacio de color RGB sin diferenciar por canal. Por lo tanto, se calculó para cada categoría la media, mediana, moda y rango de la intensidad de los pixeles. Estos valores se pueden observar como anexos en el [jupyter notebook](#) encontrado en el repositorio. A continuación, se observan estas métricas generales obtenidas para el dataset en el set de train en general.

**Tabla 1.** Estadísticas descriptivas para el set de entrenamiento en general

Media	Mediana	Moda	Std	Min.	Max.
102.727189	99.0	0	63.527691	0	255

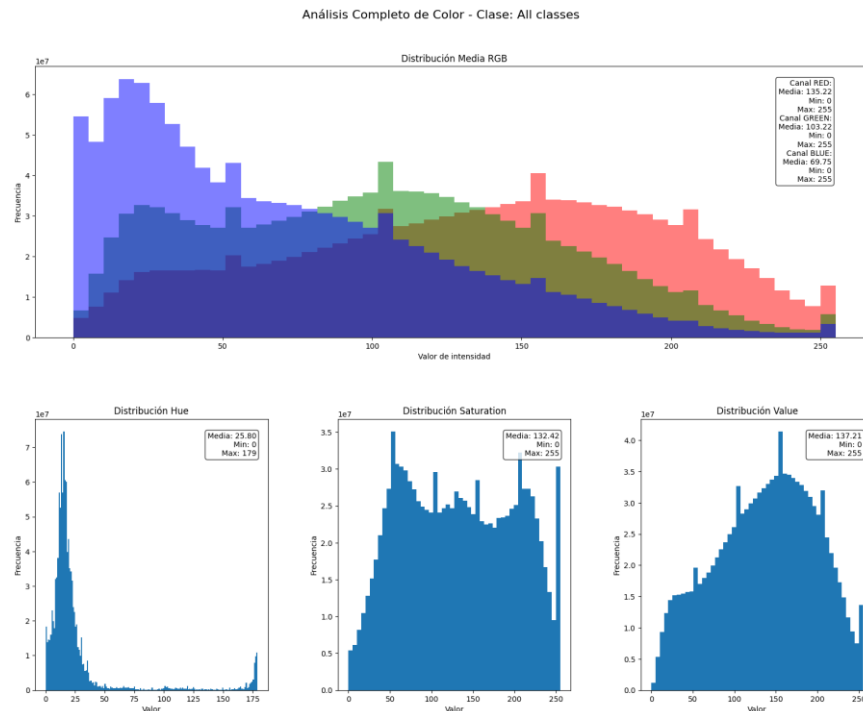
Además, se calculó la distribución de las clases basado de nuevo en la columna *Coarse\_class\_id*. Se observa que hay un alto desbalance entre clases dado que la clase mayoritaria llega a tener 26% y la clase que tiene menos tiene 0.17%. Una buena aproximación inicial es tener como *baseline* el modelo tal cual está sin aumentación. Pero un segundo modelo debería considerar la posibilidad de realizar aumentación para las clases minoritarias.



**Figura 1.** Estadísticas descriptivas sobre porcentaje total por clases basado en la columna *Coarse\_class\_id* [\[Imagen completa\]](#)

Esta aumentación se realizaría sobre las clases como Asparagus, Ginger, Papaya (...etc) para tener una cantidad de imágenes similar a la clase mayoritaria como el jugo. Por otro lado, se decidió comenzar con 43 clases y luego pasar a la cantidad total de la columna *class\_id* que son 82 (bono). Se realizó aumentación de datos utilizando el sampler de pytorch el cual les brinda prioridad a las clases desbalanceadas.

Otra estadística general que se calculó para cada clase de la columna *Coarse\_class\_id* fue la distribución de intensidad por cada canal tanto en el espacio RGB como HSV. A continuación, nuevamente, se presentan estadísticas en general del dataset en la figura 2.



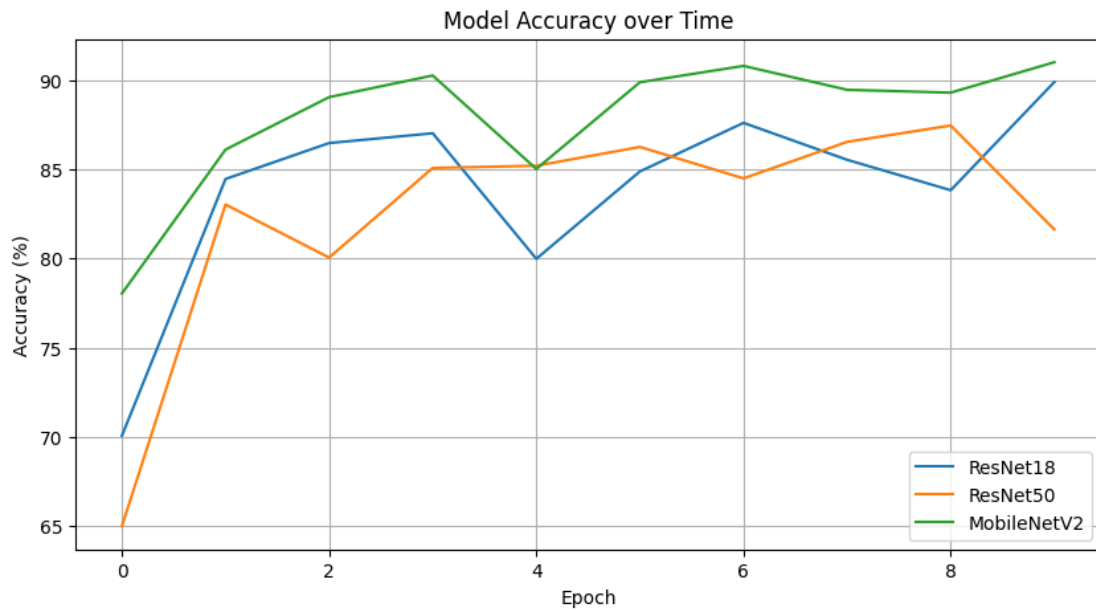
**Figura 2.** Estadísticas descriptivas sobre intensidad general de todo el dataset [\[Imagen completa\]](#)

Analizando las distribuciones de color del dataset completo como se muestra de la figura 2, podemos observar que predominan los píxeles con valores bajos en el canal azul (0-50), seguidos por una distribución más uniforme en el canal verde en valores medios (alrededor de 100) y una presencia significativa del canal rojo en valores más altos (150-250). En cuanto a las características HSV, el tono (Hue) muestra un pico pronunciado cerca de 0° que corresponde a tonos rojizos/anaranjados, lo cual es típico en productos alimenticios y frutas; la saturación (Saturation) presenta una distribución bastante uniforme con varios picos, sugiriendo una buena variedad en la intensidad de los colores; y el valor (Value) muestra una distribución en forma de campana centrada alrededor de 150, lo que indica que la mayoría de las imágenes tienen un brillo medio. No se tomaron decisiones adicionales de modificar, normalizar las imágenes.

## 2. Entrenamiento del modelo de Machine Learning

Se decidió realizar el entrenamiento del modelo de machine learning sobre modelos de la librería pytorch ya pre-entrenados con el fin de ver sus métricas en el set inicial de 42 clases. Se eligieron modelos como resnet18, resnet50 y mobilenetv2 para entrenar. Resnet18 y Resnet50 son modelos relativamente ligeros que podría llegar a optimizar los tiempos de inferencia en el supermercado, además de que tiene una arquitectura eficiente y probada que maneja bien el problema del desvanecimiento del gradiente gracias a sus conexiones residuales. Por otro lado, mobilenetv2 fue específicamente diseñado, como su nombre lo indica, para dispositivos móviles y aplicaciones en tiempo real, crucial para un entorno de supermercado también.

Por lo que este último modelo utiliza convoluciones separables en profundidad, reduciendo significativamente el número de parámetros y costo computacional. Los hiperparámetros utilizados fueron un batch size de 32, crossEntropyLoss y un learning rate de 0.001. A continuación, se muestran los resultados obtenidos por ambos modelos utilizando como métrica el accuracy en el set de prueba.



**Figura 3.** Accuracy obtenido sobre el conjunto test por los modelos [\[imagen completa\]](#)

De acuerdo con estos resultados, el mejor modelo es MobileNetV2 con un accuracy de 91.11%. Este fue el modelo seleccionado para posteriormente probar en el conjunto de validación. Por otro lado, como se indicó en la parte superior se utilizó asimismo un sampler para ver si al aumentar los datos de acuerdo con su porcentaje de participación en train los resultados mejoraban. Este se encuentra como mobilenetV2\_balanced. Se obtuvo un accuracy de 92% por lo que los resultados finales sí mejoraron en 1%.

### 3. Análisis de resultados del modelo

Se probaron los resultados en el conjunto de validación para lo cual el modelo obtuvo las siguientes métricas que se observan en la tabla 2 con mobilenetV2\_balanced.

**Tabla 2.** Estadísticas finales para el set de validación

Accuracy	F1-score	Precision	Recall
92.41%	92.17%	93.26%	92.41%

El accuracy indica que, de cada 100 productos, aproximadamente 92 son clasificados correctamente. Por otro lado, la precisión indica que cuando el modelo predice una clase, acierta el 93.26% de las veces. Esto es particularmente importante en el contexto del supermercado porque significa que hay una baja tasa de falsos positivos, reduciendo errores de cobro incorrectos. Por otro lado, el recall indica que, de todos los productos de una clase específica, el modelo identifica correctamente el 92.41%. En particular, el recall se vuelve importante en este contexto ya que un alto *recall* significa que el sistema es capaz de identificar correctamente la mayoría de los productos que un cliente toma. Lo cual minimiza las posibilidades de que un producto no sea registrado, lo que se traduce en pérdidas para el supermercado. Finalmente, el f-1 que es una mezcla de media armónica entre precisión y recall, indica un buen balance entre ambas métricas con 92.17%.

Algunos de los elementos que impactaron positivamente en las métricas obtenidas fueron el uso de modelos pre-entrenados (ResNet18/MobileNetV2) que ya tienen capacidad de extraer características relevantes.

Como mejora se puede incrementar el dataset de entrenamiento, especialmente para clases con menor rendimiento. En particular, en el conjunto de validación se obtuvo que las peores clases fueron:

- Clase 5 (Lime): No logró clasificar ninguna muestra correctamente
- Clase 16 (Pomegranate): No pudo identificar ningún caso
- Clase 36 (Brown-Cap-Mushroom): Falló en todas las clasificaciones

Se propone al negocio a futuro realizar más iteraciones, enfocadas especialmente en aquellas que mostraron los peores resultados. Para las clases 5,16,36 que obtuvieron un F1-score de 0.00, indicando un fallo completo en la clasificación, se debe aumentar significativamente el dataset de entrenamiento mediante la captura de nuevas imágenes en diferentes condiciones de iluminación, ángulos y variantes del producto. Esto se puede hacer con técnicas de aumentación de datos. A continuación, se muestran algunas muestras cualitativas de dichas clases. La lima la confunde con zucchini por su color dado que el zucchini puede ser también amarillo o verde, la pomegranada la confunde con manzanas por su color también y los mushrooms con papas también por su color. Intentar en el espacio HSV puede traer mejoras probablemente.

True: Lime  
Pred: Zucchini



Falsos Negativos para clase: Lime  
True: Lime  
Pred: Zucchini



True: Lime  
Pred: Pepper



True: Pomegranate  
Pred: Apple



Falsos Negativos para clase: Pomegranate  
True: Pomegranate  
Pred: Asparagus



True: Pomegranate  
Pred: Apple



True: Mushroom  
Pred: Potato



Falsos Negativos para clase: Mushroom  
True: Mushroom  
Pred: Potato



True: Mushroom  
Pred: Potato



#### 4. Generación de valor

Con respecto a la generación de valor, se dividió este escenario de acuerdo con: a. Los costos de tiempo asociado al registro de productos. b. El ahorro de tiempo teórico de su modelo. c. El costo asociado a los errores de su modelo. d. El ahorro real por predicción acertada. e. El ROI tomando en cuenta los costos de desarrollo y despliegue de su modelo.

#### A. Costos de tiempo asociado al registro de productos

Para este primer literal se tuvo en cuenta los minutos por registro manual que se demoraría un vendedor en registrar un producto en un supermercado, el salario de un trabajador por hora, el número de productos promedio que un trabajador registra al día y la cantidad de días laborales que hay en un año en Colombia. Se tomaron como valores base: 2 min por producto registrando y verificando la información, registro de productos por día (100) y [260 días laborales](#) al año para el 2024 en Colombia. Teniendo en cuenta los valores base se tiene:

$$Horas_{anuales} = \frac{2 \frac{\text{min}}{\text{producto}} * 100 \frac{\text{productos}}{\text{día}} * 260 \text{ días}}{60}$$
$$Horas_{anuales} = 866.66$$

Por tanto, un vendedor se gastaría 867 horas al año registrando 100 productos al día. Ahora bien, independientemente de cuantos productos registre al día el supermercado tiene la suposición de que por día en promedio vende 700 unidades de cualquier producto en general. Por tanto, necesitan 7 trabajadores trabajando los 260 días del año. Si se les paga el [salario mínimo](#) entonces el costo por mes es de:

$$Costo_{mesual} = \$1300000 * 7 = \$9,100,000$$
$$Costo_{anual} = \$9,100,000 * 12 = \$109.000.000$$

#### B. Ahorro de tiempo del modelo

Para el tiempo que se puede gastar el modelo, se tuvo en cuenta como supuesto el tiempo de captura de foto de la persona levantando el producto y el tiempo de inferencia del modelo. El tiempo de captura es de 1 segundo y tiempo de inferencia es de 4 segundos. Por tanto, si el supermercado tiene la suposición de que venden en promedio 700 productos al día, el modelo se demoraría.

$$Horas_{dia} = (1 \text{ segundo} + 4 \text{ segundos}) * 700 \text{ productos} = 0.97 \text{ horas}$$

Si en Colombia se trabaja 260 días al año entonces al año:

$$0.97 \text{ horas} * 260 = 252.2 \frac{\text{horas}}{\text{año}}$$

El modelo es mucho más eficiente que los trabajadores en alrededor del 70%.

#### C. Costo asociado a los errores del modelo

El modelo obtuvo un accuracy de 0.92 entonces eso significa que se equivoca el 8% de las veces que va a clasificar un producto. El supermercado dice que por cada error pierden en promedio \$4.500 COP. Por tanto, si al año venden 182000 productos el costo a pagar por error es de:

$$(182000 * 8\%) * 4500 = \$65.520.000$$

Se ve que el costo asociado al modelo errando a pesar de tener un accuracy de 92% es bastante alto.

Por otro lado, los costos de mantenimiento son si se despliega en GCP con **VertexAI** \$3.465 USD/hr por entrenar, \$ 1.375 USD/hr predecir y una máquina dedicada cuesta \$0.1 per node-hour. Si el supermercado está abierto 8 horas entonces:

$$Costo_{anual} = \left( (0.1 + 1.375) * 8 \text{ horas} * 21,66 \frac{\text{días laborales}}{\text{mes}} \right) * 12 + (12 * 3.465)$$
$$Costo_{anual} = 3058.56 + 41.58 = 3100.14 \text{ USD anuales}$$
$$Costo_{anual} = \$13,946,219.49 \text{ COP}$$

#### D. Ahorro real por predicción acertada

El ahorro real por predicción acertada se calcula como:

$$Ahorro_{total} = Costo_{actual} - (Costo_{modelo} + Costo_{error})$$

$$Ahorro_{total} = \$109.000.000 - (\$65.520.000 + 13,946,219) = \$29533781 \text{ COP}$$

Por lo tanto, el ahorro de predicción sería:

$$Ahorro_{predicción} = \frac{Ahorro_{total}}{predicciones_{acertadas}}$$

$$Ahorro_{predicción} = \frac{\$29533781}{(700 * 260) * 0.92} = \$176.38 \text{ pesos}$$

Se están ahorrando \$176 pesos colombianos por predicción.

#### E. ROI (Return over investments)

Finalmente, el ROI se calcula como  $\%ROI = \frac{(INGRESOS - INVERSION)}{INVERSION}$ . Los ingresos son \$29533781 y la inversión inicial incluye el tiempo de desarrollo, setup y capacitación. Al ser un proyecto de implementación en la nube, se requiere de 2 ingenieros de datos y 1 científico de datos. Por lo tanto, considerando un tiempo de implementación de 40 horas y un tiempo de entrenamiento inicial de 2 horas con GPU:

$$Inversión_{inicial} = 40 * \$100.000 \text{ COP} * 3 \text{ personas} + 3.465 \text{ USD} * 2 \text{ horas} * \$4500 \text{ COP}$$

$$Inversión_{inicial} = \$12.031.105 \text{ COP}$$

$$ROI = \frac{\$29.533.781 - \$12.031.105 \text{ COP}}{12.031.105 \text{ COP}} = 1.45 * 100 = 145\%$$

Esto significa que, por cada peso invertido, se obtiene un retorno de 1.45 pesos adicionales en el primer año. Desde el inicio el proyecto deja dividendos positivos.

### 5. Recomendación final

En conclusión, se recomienda implementar el sistema de clasificación automática de productos porque el proyecto tiene viabilidad financiera 145%, aumenta la eficiencia operacional del 70% y un accuracy obtenido para las 42 clases de 92.41% en la clasificación. A futuro, si bien el modelo da bien con 42 clases es importante seguir tomando fotos de las clases si se quiere llegar a un nivel de clasificación de las 80 categorías ya que el accuracy al duplicar las categorías cae a 25%. Por último, en las clases en las que se presentaron tan falsos positivos como negativos sería importante a futuro también intentar en otros espacios de color como HSV para ver si el modelo logra diferenciar mejor entre dichas clases.