

ESTADÍSTICA MULTIVARIADA AVANZADA
PROYECTO FINAL

PRESENTADO POR:

Edgar Leandro Jiménez Jaimes

Jorge Luis Renteria Roa

DOCENTE:

Tomás Olarte Hernandez

UNIVERSIDAD EAFIT

MEDELLÍN

MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

MAYO DE 2020

INTRODUCCIÓN

Aunque la aplicación de técnicas analíticas sobre datos no son en todos los casos rutinas secuenciales de métodos, existen metodologías que permiten aproximar una manera lógica de abordar estos problemas, partiendo desde el conocimiento del negocio hasta la puesta en escena de una solución. Con base a esto, en este trabajo se utiliza como guía la metodología ASUM DM de IBM para dar una respuesta analítica a un problema real. Esta metodología fue propuesta y explicada al detalle en el anteproyecto. Por lo cual, se abordará cada etapa metodológica de la siguiente forma:

Etapas de ASUM DM	Etapas rubrica proyecto
Analizar	Pregunta de investigación y objetivos
Diseñar	Análisis de los datos
Configurar y construir	Análisis de los resultados
Despliegue	Análisis de los resultados
Operar	Conclusiones y trabajo futuro

1. Pregunta de investigación y objetivos: (5%)

Los negocios del sector gastronómico son similares en que la gran mayoría de ellos llevan sus procesos de manera convencional con una participación casi nula de los datos para la toma de decisiones, por ello, preguntas como *¿Cuánto productos X hemos elaborado?* suelen responderse con base a la sensación de los empleados. Utilizar métodos de visión computacional y algoritmos de aprendizaje a través de redes neuronales representan una alternativa como respuesta tecnológica a esas preguntas.

Durante el desarrollo de este trabajo se implementarán modelos de aprendizaje estadístico y aprendizaje profundo buscando clasificar de manera precisa dos clases (dos tipos de pizza) en un conjunto de imágenes. Todo esto apalancado en los conceptos recibidos en clase y con la motivación personal extra de disrumpir la manera tradicional de controlar los negocios a través de métodos inteligentes de bajo costo al alcance de pequeños emprendedores (quienes son los más afectados por el problema) y escalable a los de mayor tamaño.

En resumen, el objetivo del trabajo es identificar cuán factible es la aplicación de técnicas de aprendizaje automático convencional y aprendizaje profundo para el control de productos terminados en los negocios gastronómicos. Para esto, es indispensable implementar una regresión logística como algoritmo de aprendizaje base; entrenar redes neuronales convolucionales desde cero como ejercicio académico y utilizar transfer learning por medio de una red neuronal convolucional pre-entrenada, en este caso, resnet50 de keras; presentar un comparativo de los aportes de cada forma de aprendizaje para solucionar problemas de estudio, análisis de imágenes y entregar resultados base que permitan en un futuro escalar y estabilizar la solución en producción para ser implementada en los procesos rutinarios del negocio.

2. Revisión de la literatura, estado del arte y bibliografía (8%)

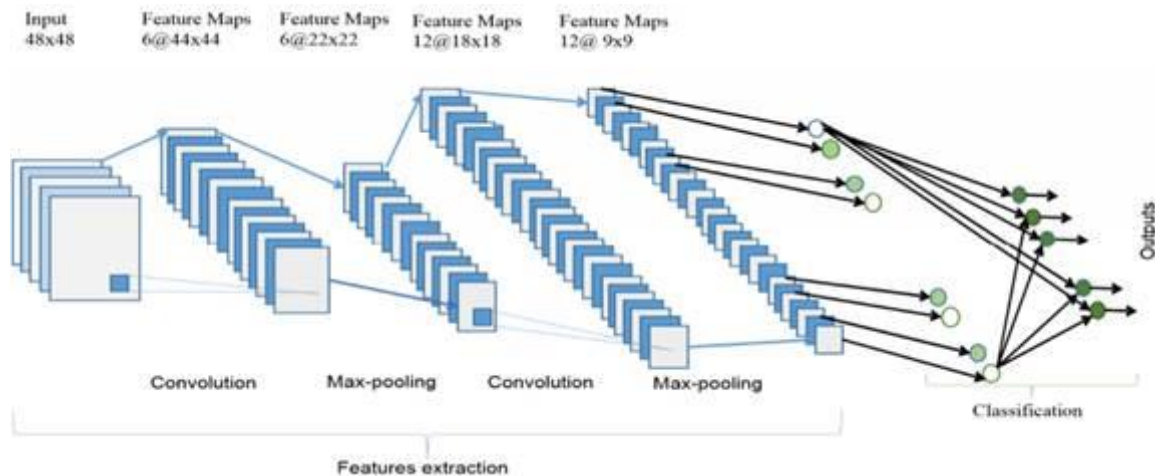
El análisis de imágenes como campo de estudio ha tenido diversas aproximaciones analíticas que han evolucionado de manera significativa en los últimos años. Estas aproximaciones comprendieron métodos de aprendizaje avanzado convencional (máquinas de soporte vectorial y regresiones logísticas) los cuales fueron migrando hacia redes neuronales profundas conforme las diversas condiciones de la misma imagen complejizaban su correcta interpretación.

La posibilidad de clasificar imágenes con alta precisión y de manera automática es importante para la organización y recuperación de este tipo de datos, sin embargo, lograr esa alta precisión en la clasificación utilizando métodos convencionales es bastante difícil pues las imágenes suelen representarse a través de un gran cúmulo de variables semánticas que no necesariamente son linealmente separables. (Goh, Chang, & Cheng, 2001) Pese a ello, las máquinas de soporte vectorial se comportan de manera relativamente buena en la clasificación de imágenes gracias a que en la capa de interpretación de datos que tienen incorporada (conocido como kernel), pueden utilizar la salida de otro clasificador. Además, las características comunes de las imágenes están representadas en los histogramas de los píxeles y las diferencias en estos histogramas permiten detectar los bordes. El verdadero reto en la aplicación de las máquinas de soporte para análisis de imágenes está en las transformaciones que se deben realizar sobre los datos para garantizar que sean linealmente separables, en caso de fallar en esta etapa el método no logrará generalizar suficientemente bien. (DataTurks: Data Annotations Made Super Easy, 2018)

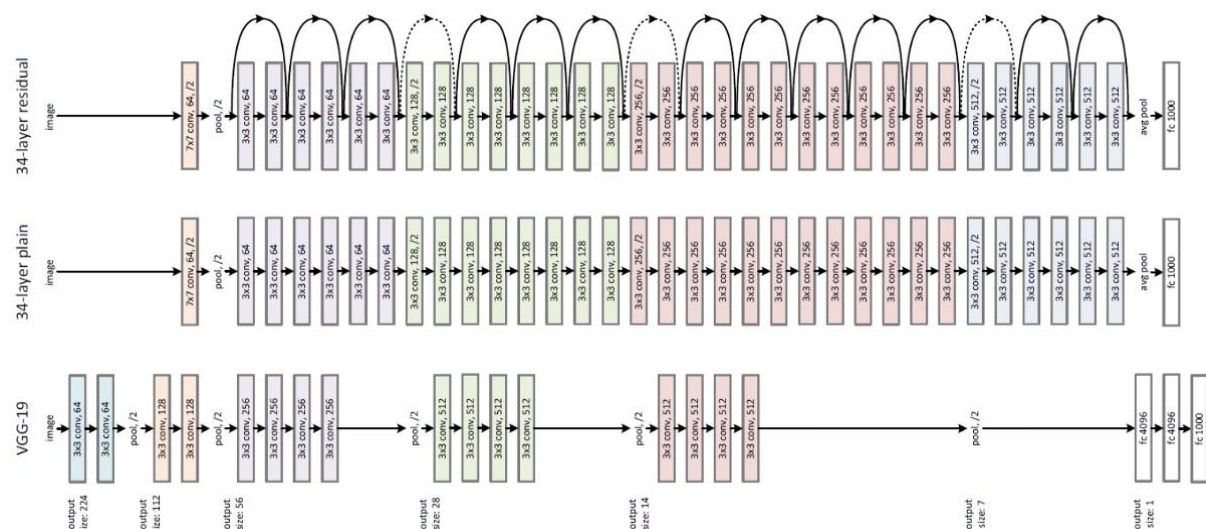
Considerando los *bullets* que hay que tener en cuenta para la aplicación de clasificadores lineales en el análisis de imágenes los métodos de aprendizaje profundo han tomado gran relevancia pues su facultad de “descubrir” automáticamente las características relevantes de una imagen a través de capas convolucionales facilitan la construcción de clasificadores. Conforme han evolucionado los métodos de aprendizaje también lo han hecho las herramientas computacionales para responder en igual medida. Un ejemplo de ello son las redes neuronales convolucionales (CNN), las cuales fueron propuestas hace más de 30 años por Fukushima pero dada la infraestructura computacional tan limitada no era posible entrenarlas en ese momento. En términos técnicos, los modelos de CNN utilizan cada una de las imágenes de entrada para pasarla a través de varias capas de convolución y aplicarles una función de activación. En detalle, utilizan pequeños píxeles de la imagen para extraer las principales características y aprender sus “rasgos clave” para reconocerlos en datos posteriores.

La arquitectura de la CNN consiste en una combinación de tres tipos de capas: Convolución, agrupación máxima y clasificación. Las capas numeradas pares son para las convoluciones y las capas numeradas impares son para las operaciones de agrupación máxima. Los nodos de salida de las capas de convolución y de maxpooling se agrupan en un mapa de características. Cada plano de una capa suele derivar de la combinación de uno o más planos de capas anteriores. Los nodos de un plano están conectados a una pequeña región de cada uno de los planos conectados de la capa anterior. Cada nodo de la capa de convolución

extrae los rasgos de las imágenes de entrada mediante operaciones de convolución en los nodos de entrada, tal como se muestra en la siguiente imagen.



De la misma forma en que se han transformado los métodos y herramientas, las arquitecturas de las mismas también han cambiado buscando como objetivo la alta precisión e imitar el procesamiento de imágenes humano. Las arquitecturas usualmente refieren a mayores capas convolucionales y de agrupamiento conectadas con una de tipo *fully connect*, sin embargo, existen otras arquitecturas avanzadas más eficientes que se diseñan para el análisis de datos a gran escala. Este es el caso de ResNet (Residual Network), esta arquitectura fue diseñada por Kaiming He buscando evitar los problemas con el gradiente presente en arquitecturas menos densas. Dentro de los diferentes subtipos, existe ResNet50 la cual contiene 49 capas dedicadas a la convolución y una última conectada al final de la red. En otras palabras, consiste en varios bloques residuales básicos cuya salida depende de la arquitectura empleada a la hora de entrenar la red. (Tsang, 2018)



Tomada de: <https://towardsdatascience.com/review-resnet-winner-of-ilsrvc-2015-image-classification-localization-detection-e39402bfa5d8>

Por otro lado, la arquitectura FractalNet es una alternativa que busca acortar los pasos en la propagación del gradiente durante el entrenamiento, con respecto a las redes residuales, las cuales suelen desempeñar de manera más eficiente en el diseño de modelos grandes.

Recientemente, las redes neuronales convolucionales han actualizado su arquitectura llevando a que cada una de las salidas en cada capa se conecten con las sucesoras en un bloque robusto, este tipo de redes fueron desarrolladas en 2017 por Gao, se conocen como redes densamente conectadas y tienen por objeto reducir los parámetros de manera drástica reutilizando eficientemente las características. (Das, 2017)

3. Análisis de datos (13%)

Siendo el actual un problema binario (dos clases de pizza) se utilizan 84 imágenes recolectadas en campo de la Pizzería Vulcano en Medellín donde 57 imágenes corresponden a la clase 1 (Pizza Pepperoni) y 27 imágenes a la clase 2 (Pizza Hawaiana). Dado el volumen de datos con el que se cuenta, se utiliza la técnica de aumentación de datos (Data augmentation) a través de ImageDataGenerator de Keras.

ImageDataGenerator es una función creada en Keras la cual genera pilas de imágenes con datos crecientes en tiempo real. Los códigos que se utilizaron en el trabajo para crear y configurar el data augmentation y para formar una red neural profunda con imágenes mejoradas son los siguientes:

Rescale → Iguala el tamaño de los datos a un nivel establecido

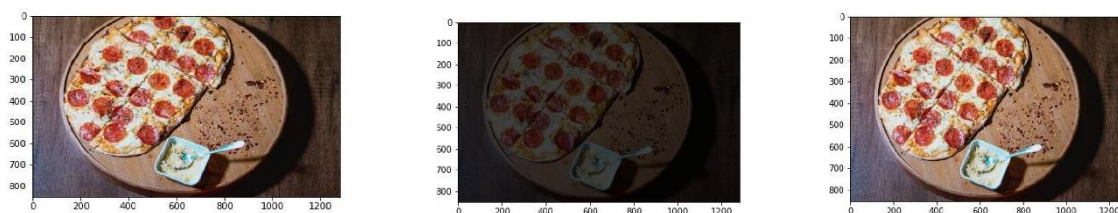
Horizontal flip → Ejecuta vueltas a las imágenes (efecto similar a un espejo)



Rotation range → Realiza giros sobre sí de cada imagen



Brightness range → Cambia los niveles de luminosidad de cada dato



Zoom Range → Realiza acercamientos o alejamientos en las imágenes.



Para mejorar el rendimiento del modelo, la función descrita se encuentra integrada en streaming sobre cada uno de los métodos aplicados, es decir, no hay generación ni almacenamiento de imágenes en batch pues en cada ejecución se genera de manera aleatoria un nuevo registro. Con la función e integración descrita, se logran generar aproximadamente 3,200 nuevas imágenes, representando esto una tasa de crecimiento cercana al 3.600%.

4. Uso de la metodología y herramientas de aprendizaje estadístico (10%)

	Preprocesamiento	Modelación	Evaluación
Funcional	Se utilizaron librerías de imágenes y recorrido entre directorios para aumentar las imágenes y analizar los datos iniciales.	Empleadas para generar una regresión logística, red convolucional básica y preentrenada desde las funciones de Keras/Tensorflow.	En esta etapa se visualizó el comportamiento de la función de pérdida de los modelos y el AUC durante cada época de las redes.
Librerías	Keras.ImageDataGenerator Keras.Image Glob Tensorflow	Keras.Conv2D Keras.MaxPooling2D Keras.Layers Keras.Models Tensorflow	Tensorflow.AUC Matplotlib Keras.evaluate_generator

Herramientas	Justificación
Colab	La creación de código y el trabajo colaborativo se facilitan
Drive	La vinculación con google Colab es nativa, permite que todos los miembros tengan la última y misma versión de datos.
Python	Conocimiento.
PIL	Uso de imágenes crudas en python, desde aquí se asignan los labels iniciales.
Keras.preprocessing	Incluye <i>imagedatagenerator</i> y permite ampliar el número de imágenes debido al pequeño volumen de registros iniciales.

Keras.Layers	Se utiliza para entrenar tres modelos, una regresión logística (para entender la aproximación analítica básica y gozar de interpretación), Red neuronal convolucional propia (como ejercicio académico y aplicación de conceptos de implementación) y Red neuronal convolucional preentrenada -Resnet50- (Para resolver el problema analítico y dimensionar las arquitecturas más robustas de cara a situaciones aplicables).
AUC	Mide la probabilidad de que dados dos puntos aleatorios, uno de la clase positiva y otro de la clase negativa, el clasificador clasifique el punto de la clase positiva más alto que el de la clase negativa. Permite medir el rendimiento del clasificador.

El archivo Libraries.txt presente en el repositorio de *github* contiene las librerías que se cargaron en el notebook de la modelación final y las versiones respectivas.

5. Entregables y su descripción (20%)

Los modelos que se usaron en definitiva en este trabajo fueron 3, regresión logística como modelo base (utilizando tensorflow y keras), una red neuronal convolucional creada y una red neuronal convolucional preentrenada: ResNet50. Se utilizó la función de pérdida o loss de `binary_crossentropy` que es la función de pérdida de cross entropía binaria y se utiliza en decisiones del tipo sí / no, 1 / 0, por ejemplo. La pérdida le dice cuán equivocadas son las predicciones del modelo; para la optimización se utilizó Adam que es un método de descenso de gradiente estocástico que se basa en la estimación adaptativa de los momentos de primer y segundo orden y como medida de desempeño se empleó el AUC (Área bajo la curva) el cual mide el “área o espacio” bajo la curva ROC y tiene por objeto medir la certeza de un modelo a la hora de realizar predicciones (toma valor de 1 cuándo se es 100% preciso) siendo invariable al umbral de clasificación.

Modelo 1: Regresión logística:

Se realiza la construcción de la regresión logística utilizando Keras, donde utilizando una sola neurona de salida la cual tendrá la función sigmoide.

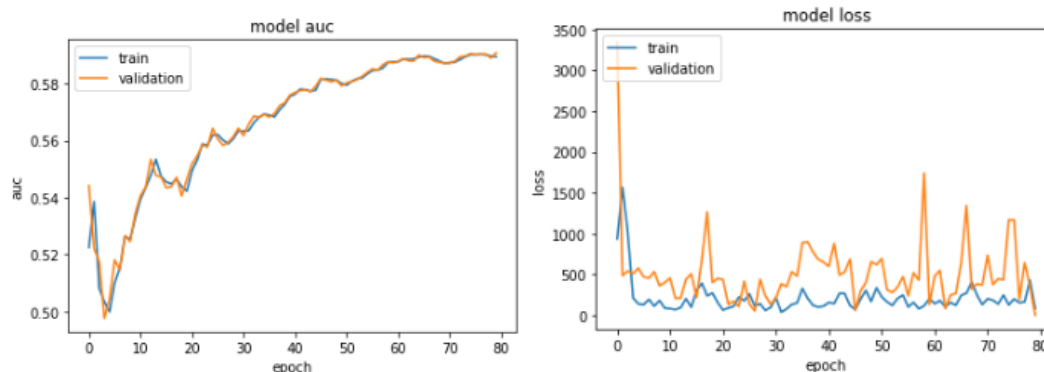
Una vez construida la arquitectura de la red, se compila el modelo, se define utilizar la función de pérdida `binary_crossentropy` debido a que es un problema de clasificación binario, se emplea ADAM como optimizador, donde se le indica una tasa de aprendizaje de 0.0001 debido a que la que trae por defecto luego de varias pruebas arrojaba variaciones altas en los desempeños de los modelos, por lo cual se decidió reducir la tasa de aprendizaje del optimizador y finalmente se le indica utilizar AUC como métrica de desempeño. Esta misma configuración se utilizó para los dos modelos siguientes

Los resultados en la métrica de AUC para etapa (train, validación y test) se resumen a continuación:

AUC en train	AUC en validación	AUC en test
0,59	0,59	0,59

Para evaluar cómo fue el desempeño del modelo se un par de gráficos que muestran cómo se comportó la métrica AUC, tanto del conjunto de entrenamiento como de validación, durante las épocas en que se entrenó el modelo, que para este caso se utilizaron un total de **80 épocas**, es decir, el número de veces que el modelo pasó por los datos de train para realizar el ajuste.

Como segunda gráfica se observa el comportamiento de la función de costo, al igual que el auc, para train y validación durante todas las épocas



Los resultados de este modelo son como se esperaban, no muy buenos, y esto es debido a que solo se está trabajando con los valores de los píxeles en la entrada y no tienen un procesamiento adecuado al tratarse de imágenes. En la curva de AUC se observa una leve mejora pero no es muy significativa en términos reales, tanto para train como para validación se observa un comportamiento estable, por su parte, aunque la función de pérdida disminuye al comienzo de las épocas, esta tiene valores altos y el comportamiento de train y test fluctúa bastante, siendo menos estable el conjunto de validación.

Para la entrega final de este proyecto solo se consideró utilizar un modelo de machine learning básico y apuntar los esfuerzos a modelo más recientes. Sin embargo, para la presentación del anteproyecto se habían planteado unos modelos clásicos iniciales los cuales en asesoría con el profesor se tomó su recomendación de solo tomar un modelo base y enfocar, como se mencionó anteriormente, esfuerzos en los modelos de deep learning

De acuerdo a lo anterior, se presenta a continuación la construcción y ejecución de dos modelos de deep learning, donde en primer lugar se realiza y ejecuta una arquitectura de una red neuronal convolucional desde cero y posteriormente se carga un modelo preentrenado (resnet50), se configura y entrena la capa de salida.

Modelo 2: Red Neuronal Convolucional con arquitectura propuesta:

Para la arquitectura de la CNN propuesta, se utilizó 5 capas convolucionales, donde la primera capa tiene 128 filtros, es decir la dimensionalidad del espacio de salida (o también se puede decir como filtros de salida en la convolución), la segunda capa 64, luego 32, hasta

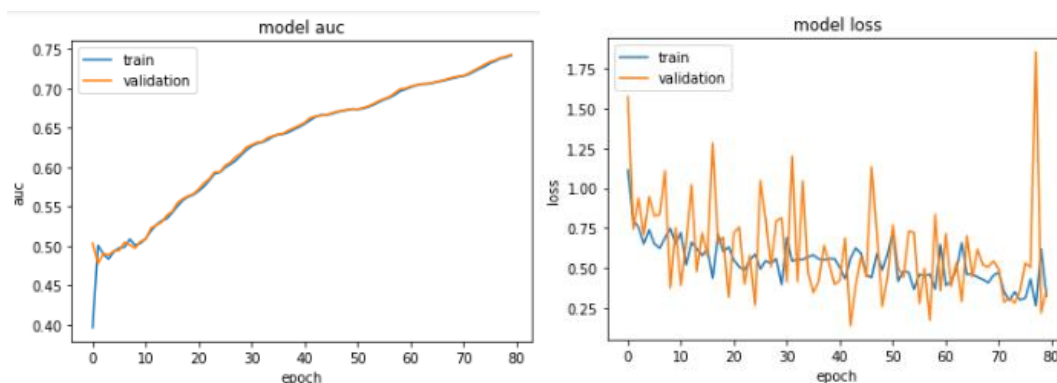
llegar a 8. Para cada capa convolucional se utiliza la función de activación relu, y finalmente se utiliza una capa densa de 64 neuronas con función de activación relu y una capa de salida con la función sigmoide debido al problema de clasificación binario.

Como se mencionó anteriormente, se compilo el modelo utilizando binary cross entropy como función de pérdida, ADAM como optimizador y AUC como métrica de desempeño.

Los resultados en la métrica de AUC para etapa (train, validación y test) se resumen a continuación:

AUC en train	AUC en validacion	AUC en test
0,741	0,743	0,743

Al igual que el caso del modelo de regresión logística, para este modelo de CNN con arquitectura propuesta se presentan el mismo par de gráficas de comportamientos del AUC y la función de costo para cada una de las épocas **(80 épocas)**



Los resultados de esta CNN son mejores que los que se presentó en la logística, al igual que la anterior, los resultados del auc son estables para los conjuntos de train y validación, por su parte el comportamiento de la función de pérdida fluctúa bastante, siendo más inestable en validación, sin embargo los valores son significativamente menores a los que se presentó en la logística.

Modelo 3: Red Neuronal Convolucional con arquitectura predefinida (Resnet de Keras):

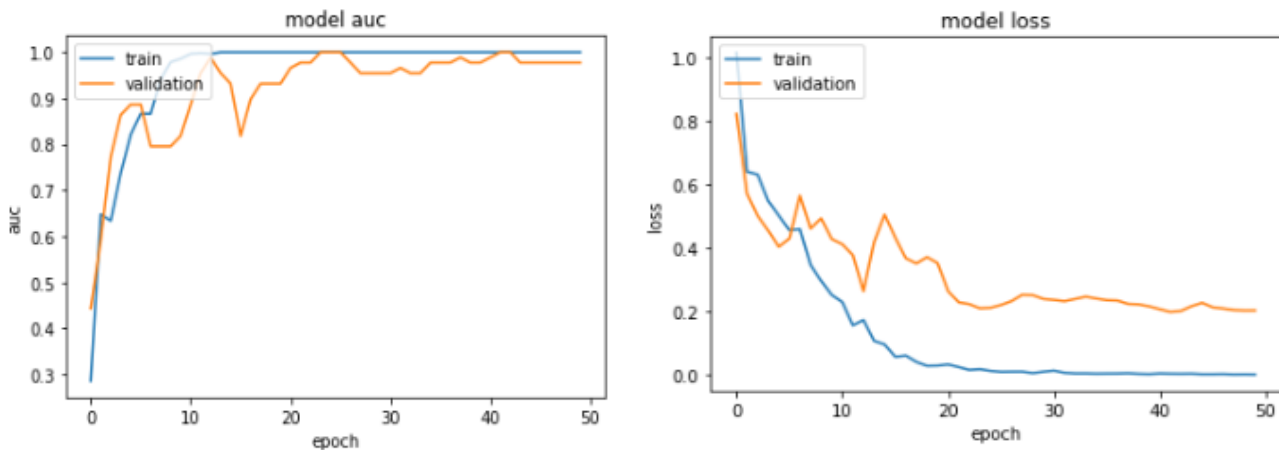
Para el uso de esta red pre-entrenada se importa desde keras resnet50 utilizando: `tf.keras.applications.resnet50.ResNet50` y se le indica dentro de la función no incluir la capa de salida. Esto principalmente a que se va a construir esta capa utilizando la función sigmoide, al igual que para los dos ejercicios anteriores. De esta manera, se importa resnet50, posteriormente se crea la capa de salida y se unen estas dos etapas (inputs – outputs) para posteriormente ser compilada y entrenada.

Como se mencionó anteriormente, se compilo el modelo utilizando binary cross entropy como función de pérdida, ADAM como optimizador y AUC como métrica de desempeño.

Los resultados en la métrica de AUC para etapa (train, validación y test) se resumen a continuación:

AUC en train	AUC en validacion	AUC en test
1	1	0,99

De igual manera que los casos anteriores, para este medir los resultados de Resnet50 se presentan el mismo par de gráficas de comportamientos del AUC y la función de costo para cada una de las épocas (**50 épocas**) , para esta caso se utilizaron 50 épocas luego de varias pruebas debido los resultados eran muy buenos desde muy pocas épocas.



El comportamiento del desempeño de ResNet50 es el mejor de los tres modelos, tal como se esperaba, la gráfica de desempeño de AUC muestra valores altos en pocas épocas, aunque tiene algunas fluctuaciones en validación con el pasar de las épocas este se estabiliza a partir de la 30 aproximadamente. Por otra parte la función de pérdida tiene valores ligeramente menores a los presentados en la CNN anterior, y para este caso se presenta una brecha de 0.2 unidades al finalizar la época 50, sin embargo presenta los mejores resultados en la función de pérdida.

6. Conclusiones y trabajo futuro (7%)

El análisis de imágenes utilizando métodos de aprendizaje avanzados (incluyendo redes neuronales) corresponde a un campo que ha sido foco de esfuerzos académicos muy marcados durante los últimos años y que gracias a la evolución en materia de fuerza computacional se han podido implementar muchas soluciones pensadas incluso desde hace más de tres décadas. Estas soluciones mencionadas poseen la característica que más allá de ser extensas (en cuanto a “filas de código”) son ejercicios reiterativos donde la mayoría de esfuerzos se concentran en la optimización de parámetros y selección de técnicas correctas.

Los métodos de aprendizaje convencionales (como lo son las regresiones logísticas, y modelos clásicos de machine learning) aportan interpretación a los datos, pese a ello, en muy pocas situaciones logran generalizar el problema lo suficientemente bien como para considerarse una solución factible al caso. Con base a esto, se recomienda que en los problemas de clasificación de imágenes se enfoquen los esfuerzos en modelos de aprendizaje profundo los cuáles a través de sus capas de convolución gozan de la capacidad de identificar de manera autónoma y automática cuáles son aquellas características determinantes del problema,

Así mismo, aumentar la capacidad de generalización de los métodos de aprendizaje profundo (evitar el overfit) es un paso “obligado” de cara a la escalabilidad de los modelos y su puesta en producción, por ello, utilizar técnicas de aumentación de datos permite robustecer el aprendizaje y crear registros artificiales. Además de esto, hacer uso de modelos preentrenados ayuda a aproximar la solución del problema con base a arquitecturas de aprendizaje previamente diseñadas (conocido esto como *transfer learning*).

La combinación de estas técnicas mencionadas (aumentación de datos y transfer learning) permitió determinar que es muy factible utilizar métodos de aprendizaje profundo y visión computacional para controlar los productos terminados en los negocios gastronómicos, esta conclusión existe gracias a la construcción de un modelo clasificador basado en Resnet50 capaz de identificar la clase de la pizza en un conjunto de imágenes con una precisión cercana al 100%.

Trabajos futuros

La operación de los negocios gastronómicos está sujeta a dinámicas particulares que llevan a que modelos como el construido tengan un alcance limitado y en muchas situaciones, no sean capaces de responder o controlar todas las preguntas de valor a las que están sujetos. Un ejemplo de este tipo de casos aplicado a una pizzería es el escenario donde un cliente pide una pizza con insumos de la pizza de clase 1 y la clase 2. Pensando en estas situaciones el modelo deberá evolucionar a determinar no la clase de la pizza producida sino cuáles son los ingredientes con los cuáles está hecha.

Desde el punto de vista técnico, el modelo futuro deberá pensarse como un sistema que como entrada recibe una cadena de video en streaming de una cámara desde la cocina del negocio y en tiempo real es capaz de determinar las cantidades utilizadas y producidas, además, de hacer mayor uso de capacidades computacionales avanzadas para brindar una continuidad en la operación.

7. Ejecución del plan (10%)

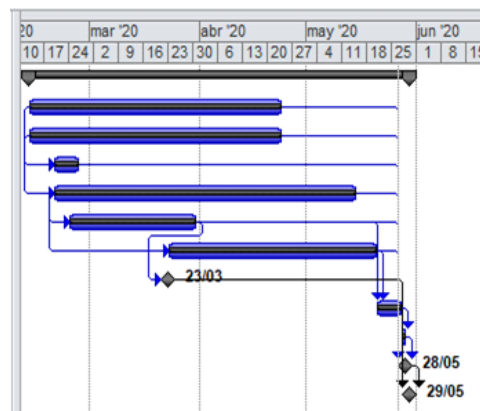
A continuación se presenta como fue la planeación inicial del proyecto, que tenía como fecha de entrega el 19 de Mayo vs la ejecución real con finalización del proyecto el día 29 de mayo (contando entrega de documento y visualización):

Task Name	Duración	Comienzo	Fin	% Línea Base	% completado	% Var	Predecesoras
Cronograma Proyecto Clasificación de Imágenes	69 días	13/02/2020 8:00	19/05/2020 17:00	100%	95%	5%	
Captura de imágenes	51 días	13/02/2020 8:00	23/04/2020 17:00	100%	100%	0%	
Asignación de label	51 días	13/02/2020 8:00	23/04/2020 17:00	100%	100%	0%	
Lectura y estructuración de pipeline	5 días	20/02/2020 8:00	26/02/2020 17:00	100%	100%	0%	2CC+5 días
Data preprocessing	56 días	20/02/2020 8:00	7/05/2020 17:00	100%	100%	0%	3CC+5 días
Modelación y calibración Machine Learning	26 días	24/02/2020 8:00	30/03/2020 17:00	100%	100%	0%	5CC+2 días
Modelación y calibración Deep Learning	33 días	23/03/2020 8:00	6/05/2020 17:00	100%	90%	10%	5CC+22 días
Hto: Entrega Anteproyecto	0 días	23/03/2020 8:00	23/03/2020 8:00	100%	100%	0%	6FC-6 días
Análisis de resultados	4 días	7/05/2020 8:00	12/05/2020 17:00	100%	0%	100%	6,7
Conclusiones	2 días	13/05/2020 8:00	14/05/2020 17:00	100%	0%	100%	9
Entrega final	3 días	15/05/2020 8:00	19/05/2020 17:00	100%	0%	100%	10;4;3;6;7;5;2
Entrega Documento Final	0 días	19/05/2020 17:00	19/05/2020 17:00	100%	0%	100%	11;8

El proyecto se planeó inicialmente del 13 de febrero al 19 de mayo. De acuerdo a este planeación inicial y tomando el avance al 19 de mayo, día que se debería de haber entregado, se tenía un avance del 95% del proyecto, esto quiere decir un atraso aproximado de 3 días,

sin embargo debido a la ampliación del plazo, se construyó otra línea base que es la que se presenta a continuación:

Task Name	Duración	Comienzo	Fin	% Línea Base	% completado	% Var	Predecesoras
Cronograma Proyecto Clasificación de Imágenes	77 días	13/02/2020 8:00	29/05/2020 17:00	100%	100%	0%	
Captura de imágenes	51 días	13/02/2020 8:00	23/04/2020 17:00	100%	100%	0%	
Asignación de label	51 días	13/02/2020 8:00	23/04/2020 17:00	100%	100%	0%	
Lectura y estructuración de pipeline	5 días	20/02/2020 8:00	26/02/2020 17:00	100%	100%	0%	2CC+5 días
Data preprocessing	61 días	20/02/2020 8:00	14/05/2020 17:00	100%	100%	0%	3CC+5 días
Modelación y calibración Machine Learning	26 días	24/02/2020 8:00	30/03/2020 17:00	100%	100%	0%	5CC+2 días
Modelación y calibración Deep Learning	43 días	23/03/2020 8:00	20/05/2020 17:00	100%	100%	0%	5CC+22 días
Hto: Entrega Anteproyecto	0 días	23/03/2020 8:00	23/03/2020 8:00	100%	100%	0%	6FC-6 días
Análisis de resultados	5 días	21/05/2020 8:00	27/05/2020 17:00	100%	100%	0%	6,7
Conclusiones	1 día	28/05/2020 8:00	28/05/2020 17:00	100%	100%	0%	9
Entrega final Documeto Técnico	0 días	28/05/2020 17:00	28/05/2020 17:00	100%	100%	0%	10;4;3;6;7;5;2
Entrega Visualización Final	0 días	29/05/2020 17:00	29/05/2020 17:00	100%	100%	0%	11FC+1 día;8FC+1 día



Aunque las actividades de procesamiento de datos y modelación hubo atrasos de acuerdo a lo inicialmente planeado, dicho tiempo se compensó casi en su totalidad gracias a la reducción de esfuerzo empleado en la elaboración del trabajo final. El retraso inicial en ambas actividades se dio con motivo de mayores investigaciones requeridas para implementar la aumentación de datos en tiempo real (es decir, sin utilizar almacenamiento en batch) y conocer el funcionamiento de los modelos de deep learning. La primera investigación permitió no colapsar los recursos de *google colab* al momento de cargar un poco más de 3.000 imágenes desde *google drive*, mientras que la segunda, apalancó el entendimiento de los métodos de aprendizaje profundo, los cuáles no habían sido abordados previamente por los investigadores.

Los aprendizajes obtenidos en el campo de análisis de imágenes están concentrados en dos frentes principalmente. El primero es controlar los datos desde la entrada de las imágenes buscando encontrar registros con mala calidad o duplicados en el repositorio, esto es de suma importancia pues podría significar grandes problemas en la implementación de métodos de aprendizaje convencionales e incluso, altos niveles de desempeño falsos. El segundo aprendizaje está alrededor de indagar desde el inicio en los métodos no conocidos (como lo eran los modelos de aprendizaje profundo) para evitar reprocesos continuos y errores en el diseño de sus parámetros.

8. Implicaciones éticas (2%)

La existencia de los controles está ligada al origen del hombre, atribuible principalmente a la ausencia de confianza, dichos controles han significado durante todo el tiempo la vulneración de la privacidad y en algunos casos, señales de “irrespeto” hacia los controlados. En este caso, se ha buscado cambiar la forma en la cual se revisan las ventas y producción de la pizzería utilizando técnicas de menor fricción, pese a ello, la mera existencia de una cámara allí implica ruptura de confianza por parte de los empleados y la sensación de poca privacidad dentro del negocio. Considerando una evolución avanzada del proyecto se tendrían impactos en el número de empleos (o al menos en el rol que se desempeña) dentro de la pizzería, pues ya no sería necesario contratar a una persona que lidere el consumo y abastecimiento de la materia prima.

6. Aspectos legales y comerciales (5%)

Uno de los grandes problemas que encuentran los restaurantes durante su fase de crecimiento es la pérdida de control sobre los inventarios y dada la naturaleza del negocio, constantemente se presentan situaciones que dan lugar a malos manejos de la materia prima o la contabilización de las ventas. Esta solución apunta directamente a disminuir dichas situaciones de manera muy económica (baja inversión) y basado en su gran componente tecnológico se presenta como un servicio capaz de ser replicado en cualquier negocio de alimentos sin importar su tamaño.

Desde el ámbito legal no existen problemas para la puesta en producción pues los datos son información propia que se recopila en cada uno de los establecimientos que requieran implementar la solución. Esto, sin dejar de lado que se precisa registrar legalmente la actividad comercial y proteger la propiedad intelectual.

7. Repositorio de Github.

Se trabajó con libros de jupyter en Google Colab y se crearon copias al siguiente repositorio de Github:

<https://github.com/eljimenezj/CM0868--Estadistica-Multivariada-Avanzada>

8. Bibliografía

(Tutorial) Understanding Logistic REGRESSION in PYTHON. (2019, December 16). DataCamp Community. <https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python>.

CS231n Convolutional Neural Networks for Visual Recognition. (2014). CS231n Convolutional Neural Networks for Visual Recognition. <https://cs231n.github.io/convolutional-networks/>

Deep Residual Learning for Image Recognition. (2015). arXiv.org. <https://arxiv.org/abs/1512.03385>

Gradient boosting. (2017, September 29). Wikipedia, la enciclopedia libre. Retrieved March 23, 2020, from https://es.wikipedia.org/wiki/Gradient_boosting

Koehrsen, W. (2017, December 27). *Random Forest Simple Explanation.* Medium. <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>

TensorFlow for Deep Learning. (2014.). O'Reilly Online Learning. <https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>

Towards an Improved ASUM-DM Process Methodology for Cross-Disciplinary Multi-organization Big Data & Analytics Projects. (2018). SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-95204-8_51

Das, S. (16 de Noviembre de 2017). *Medium.* Obtenido de CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more...: <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>

DataTurks: Data Annotations Made Super Easy. (10 de Agosto de 2018). *Medium.* Obtenido de Understanding SVMs': For Image Classification: <https://medium.com/@dataturks/understanding-svms-for-image-classification-cf4f01232700>

Goh, K.-S., Chang, E., & Cheng, K.-T. (Enero de 2001). SVM Binary Classifier Ensembles for Image Classification.

Tsang, S.-H. (15 de Septiembre de 2018). *Towards Data Science.* Obtenido de Review: ResNet — Winner of ILSVRC 2015 (Image Classification, Localization, Detection): <https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8>