## PLANTEMIENTO DEL PROBLEMA

Hola Nicole, ¿Cómo van esas predicciones de ventas? El Departamento de Operaciones ya está pensando en aplicar tus modelos predictivos para uno de sus casos de uso más críticos: la reposición en tiendas. **El abastecimiento de stock a tiendas** se lleva a cabo de manera **semanal**, al principio de cada semana (aunque si es necesario, el stock de algunos productos se puede reponer diariamente). Probablemente ya estés visualizando la importancia de las predicciones de ventas para eso. **Minimizar el stock** remanente es el sueño de todo retailer, pero ese sueño es aún más fuerte **para los artículos de supermercado**.  ¿Podrías redactar una propuesta que detalle tu solución para aplicar los modelos de ventas al caso de uso de abastecimiento de tiendas? También tendríamos que especificar las **extensiones que los modelos puedan requerir**, así como los detalles de la **productivización del modelo** para el caso de uso. Martin solicitó el despliegue de una **API** para su ejecución. ¿Es algo con lo que te sentirías cómoda? No te preocupes por la implementación por ahora, centrémonos en ese documento con la solución.  Gracias Paul

Hola Nicole,  Tengo las expectativas muy altas para la aplicación de los modelos de predicción de ventas al problema de abastecimiento de las tiendas. Este es un caso de uso muy bien elegido para comenzar a demostrar los beneficios asociados a la aplicación de la ciencia de datos a todos los niveles de la empresa. No tengo ninguna duda de que nuestra solución basada en algoritmos de ML mejorará significativamente la forma en que DSMarket ha estado estimando el stock necesario para cada tienda durante todo este tiempo.  Antes de ir a lo grande en todo el país, **me gustaría que diseñaras una prueba piloto para demostrar con confianza las mejoras directamente asociadas con el nuevo enfoque** (¡es decir, dólares!). La creación de una cultura basada en datos debe ser una de nuestras prioridades en esta etapa, y es importante que todos comiencen a creer en el poder de este tipo de algoritmos. Estoy pensando en un estudio de casos y controles (¿tal vez por productos o tiendas? ¿tal vez por ambos?). ¡Tú eres la experta! ¿Qué harías y cómo la evaluarías? ¿Puedes enviarme una propuesta?  ¡Gracias por el trabajo! M

## CONSIDERACIONES PREVIAS

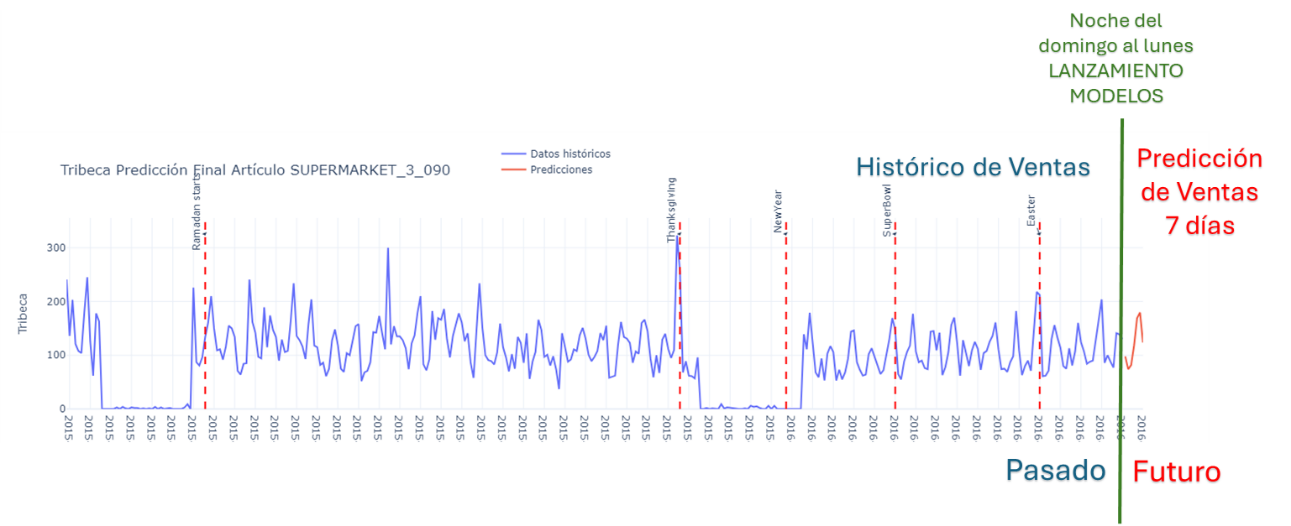
Aprovechando los modelos para la previsión de ventas esta vez los utilizamos para predecir unidades de venta en lugar de cifra de ventas.

Existen ciertos datos necesarios que nos tiene que proporcionar Operaciones y Logística para poder afinar bien las predicciones pero a priori partiremos de las siguientes premisas:

* Se pueden realizar pedidos a los proveedores todos los días de la semana con fecha de entrega los lunes a primera hora.
* El plazo de entrega es de 24 horas a todas las tiendas y todos los artículos,
* Gráfico, Gráfico de barras

  Descripción generada automáticamentePrácticamente el 50% de las ventas de la semana se hacen de viernes a Domingo, con especial incidencia los sábados y el domingo, el lunes conserva un poco la inercia de ventas y de martes a jueves son los días más tranquilos.

## OPERATIVA

1. **Noche del domingo al lunes:** Se lanzan las previsiones semanales y obtendremos las ventas previstas para el periodo del martes al lunes siguiente.
2. **El lunes a primera hora:** Una vez obtenida la demanda prevista por artículo, se realizan los pedidos automáticamente teniendo en cuenta la siguiente fórmula:
   1. ***Predicción Venta del Lunes:*** La incluimos en la fórmula ya que el pedido se realizará el lunes a primera hora y llegará el martes, por lo tanto hay que tener en cuenta la venta del lunes para restarla al stock disponible.
   2. ***Predicción Venta próximos 7 días:*** Hemos acordado que se haría un pedido semanal para abastecer los 7 días, y si fuera necesario se harían pedidos puntuales.
   3. ***Stock Disponible:*** Restamos la cantidad de stock que hay disponible en los lineales y en el almacén de la tienda.
   4. ***Pedidos Pendientes:*** Restamos las cantidades que ya están pedidas y que por algún motivo están pendientes de llegar a la tienda (este aspecto es algo a tratar por el jefe de sección periódicamente para anular todos los pedidos pendientes que por algún motivo no van a llegar)
   5. ***Stock Mínimo:*** Es el stock que el jefe de producto determina que debe tener un artículo. Este stock a veces responde a la necesidad de que haya presencia de producto en los ganchos para incentivar la venta o a veces sencillamente porque hay productos que se suelen vender de dos en dos o de cuatro en cuatro como en el caso de los neumáticos. Si con el pedido propuesto no se llega al stock mínimo, se suma la cantidad necesaria.
   6. ***Stock Máximo:*** Es el stock que se define como medida de seguridad ante predicciones desproporcionadas. Si queremos automatizar la confección del pedido y que esté mas o menos desatendida es un valor necesario que debería tener asignado cada artículo. Si el pedido supone sobrepasar el stock máximo se restarán las unidades necesarias para no hacerlo.
3. **Martes:** Llega la mercancía y se realiza la recepción ya que es un día valle y habrá más disponibilidad de los trabajadores para recepcionar, almacenar y reponer los lineales. En caso de diferencias entre lo pedido y lo recepcionado, los responsables de cada departamento tendrán que realizar pedidos de refuerzo de artículos sustitutos o devoluciones de aquello que sobre.
4. **La noche del Jueves al viernes** lanzaría un punto de control con el stock disponible por producto y las previsiones de venta ya calculadas para el fin de semana, para reponer puntualmente de cara a los siguientes días por si hubiera riesgo de rotura de stock debido a una demanda extraordinaria o posibles demarcas de producto (robo, deterioro, mermas)
5. **Viernes:** De esta forma el viernes se harían los pedidos extraordinarios y estarían disponible el sábado a primera hora.
6. **Se repite el ciclo semanal el domingo por la noche** con una nueva previsión de ventas incorporando ya al histórico la semana que acaba de terminar.
7. Gráfico, Diagrama, Gráfico de burbujas

   Descripción generada automáticamente**Caso Especial es el de los productos Frescos y perecederos de la categoría SUPERMERCADO** y que además pueden tener fechas de caducidad distintas. El ciclo de vida de algunos de ellos es diario (pan, leche fresca), mientras que otros pueden durar varios días (frutas y verduras) y otros semanas. La propuesta sería gestionarlos de forma distinta, o más bien, con una periodicidad distinta, de tal forma que todos los días y en función de las predicciones para el día siguiente que nos proporciona el API, se pueda hacer un pedido al cierre de la tienda, una vez descontado el stock de los productos caducados.
8. Diagrama, Gráfico de burbujas

   Descripción generada automáticamente

ARQUITECTURA Y MLOPS

Para montar un sistema en Google Cloud que permita a tu modelo de machine learning hacer predicciones de ventas para todas las tiendas geográficamente dispersas y mantener actualizada una base de datos con ventas diarias, aquí tienes una propuesta detallada que aprovecha varios servicios en la nube de Google. Esta solución considera la escalabilidad, automatización, procesamiento paralelo y almacenamiento seguro de los datos.

### 1. \*\*Google Cloud Services principales\*\*:

- \*\*Google Cloud Storage (GCS)\*\*: Para almacenar los datos de entrada y salida (como predicciones, registros y datos históricos).

- \*\*BigQuery\*\*: Para almacenar las ventas diarias y otros datos históricos que tu modelo utilizará para hacer predicciones.

- \*\*Vertex AI\*\*: Para gestionar, entrenar y desplegar el modelo de machine learning.

- \*\*Cloud Functions o Cloud Run\*\*: Para la ejecución del proceso de predicción, particularmente útil para orquestar la paralelización.

- \*\*Cloud Scheduler\*\*: Para automatizar las ejecuciones del modelo cada domingo por la noche.

- \*\*Pub/Sub\*\*: Para gestionar los eventos y coordinar la ejecución en paralelo.

- \*\*Cloud SQL o Firestore\*\*: Base de datos para el almacenamiento de las ventas diarias actualizadas.

### 2. \*\*Diagrama general del flujo\*\*:

1. \*\*Preparación de Datos\*\*:

- Las ventas diarias de cada tienda se almacenan automáticamente en \*\*BigQuery\*\* o \*\*Cloud SQL\*\*.

- Estos datos se pueden obtener a través de integraciones con el sistema de puntos de venta (POS) o mediante archivos CSV que se cargan en \*\*Cloud Storage\*\*.

- Puedes usar \*\*Dataflow\*\* o scripts programados para cargar los datos automáticamente en el sistema de almacenamiento.

2. \*\*Despliegue del Modelo de Machine Learning\*\*:

- Usa \*\*Vertex AI\*\* para gestionar el modelo de machine learning, ya que ofrece una solución de ML completa en la nube (entrenamiento, despliegue y predicción).

- Una vez que el modelo esté entrenado, puedes desplegarlo como un servicio de predicciones utilizando \*\*Vertex AI\*\*.

- El modelo se conecta a los datos almacenados en \*\*BigQuery\*\* o \*\*Cloud SQL\*\* para realizar predicciones basadas en las ventas más recientes.

3. \*\*Automatización de la Ejecución Semanal\*\*:

- Utiliza \*\*Cloud Scheduler\*\* para que cada domingo por la noche (o en cualquier momento programado), se desencadene una función que ejecute las predicciones del modelo para todas las tiendas.

- \*\*Cloud Functions\*\* o \*\*Cloud Run\*\* se pueden utilizar para orquestar la lógica de ejecución. Aquí se lanzan trabajos paralelos usando \*\*Pub/Sub\*\* para cada tienda o grupo de tiendas.

4. \*\*Paralelización de Predicciones\*\*:

- La clave para procesar las predicciones rápidamente es dividir el trabajo. Puedes usar \*\*Pub/Sub\*\* para enviar mensajes que desencadenen procesos en \*\*Cloud Run\*\* o \*\*Cloud Functions\*\*, cada uno haciendo predicciones para un subconjunto de las tiendas.

- Esto permite que las predicciones para las tiendas se realicen de manera concurrente, reduciendo significativamente el tiempo de procesamiento.

5. \*\*Almacenamiento de Resultados\*\*:

- Las predicciones generadas se almacenan en \*\*BigQuery\*\* o \*\*Cloud Storage\*\*, dependiendo de cómo quieras analizarlas posteriormente.

- \*\*BigQuery\*\* puede ser útil si necesitas realizar análisis o informes detallados sobre las predicciones y ventas.

6. \*\*Actualización Diaria de Ventas\*\*:

- Cada día, los datos de ventas se actualizan en \*\*BigQuery\*\* o \*\*Cloud SQL\*\*. Para esto, podrías usar una integración directa desde los sistemas de POS de las tiendas a \*\*BigQuery\*\*, o bien procesar los archivos que se cargan automáticamente en \*\*Cloud Storage\*\* mediante \*\*Dataflow\*\*.

7. \*\*Monitoreo y Alertas\*\*:

- Configura \*\*Google Cloud Monitoring\*\* para asegurarte de que todos los procesos funcionan correctamente y establecer alertas en caso de fallos.

- Usa \*\*Logging\*\* para registrar todas las ejecuciones y predicciones realizadas, lo que te permitirá rastrear el rendimiento y los errores.

### 3. \*\*Tecnologías clave y ventajas\*\*:

- \*\*Vertex AI\*\*: Permite entrenar, probar y desplegar modelos en producción de manera eficiente y escalable. Además, está completamente integrado con otros servicios de Google Cloud como BigQuery y Cloud Storage.

- \*\*BigQuery\*\*: Es ideal para manejar grandes volúmenes de datos, como los históricos de ventas y predicciones. Su capacidad para realizar análisis en tiempo real es útil para alimentar el modelo con los últimos datos.

- \*\*Cloud Functions o Cloud Run\*\*: Son servicios serverless que permiten ejecutar procesos sin preocuparse por la infraestructura. Ambos soportan escalabilidad automática, lo cual es perfecto para manejar cientos o miles de tiendas simultáneamente.

- \*\*Pub/Sub\*\*: Facilita la paralelización de tareas y la comunicación asincrónica entre diferentes servicios. Es clave para lanzar procesos concurrentes para cada tienda.

- \*\*Cloud Scheduler\*\*: Programa la ejecución automática de los procesos semanales, lo que elimina la necesidad de intervención manual.

### 4. \*\*Costos y Escalabilidad\*\*:

- \*\*Costo\*\*: Los servicios mencionados son pagos bajo demanda, es decir, solo pagas por el uso que realices. Por ejemplo, BigQuery cobra por consulta y almacenamiento, Vertex AI por el entrenamiento y las predicciones, y Cloud Functions por las ejecuciones.

- \*\*Escalabilidad\*\*: Al usar servicios como Pub/Sub, Cloud Run y Vertex AI, puedes escalar el procesamiento a cientos o miles de tiendas sin preocuparte por la capacidad de la infraestructura. Google Cloud ajusta automáticamente los recursos en función de la demanda.

### 5. \*\*Pasos para Implementar\*\*:

1. \*\*Configurar almacenamiento y bases de datos\*\*: Crear los buckets de \*\*Cloud Storage\*\* para datos temporales y la base de datos en \*\*BigQuery\*\* o \*\*Cloud SQL\*\*.

2. \*\*Entrenar y desplegar el modelo\*\*: Cargar los datos de entrenamiento en \*\*BigQuery\*\*, usar \*\*Vertex AI\*\* para entrenar el modelo y desplegarlo como un servicio.

3. \*\*Configurar las funciones serverless\*\*: Implementar \*\*Cloud Functions\*\* o \*\*Cloud Run\*\* para ejecutar predicciones paralelas por tienda usando \*\*Pub/Sub\*\*.

4. \*\*Automatización\*\*: Configurar \*\*Cloud Scheduler\*\* para desencadenar el proceso cada domingo por la noche.

5. \*\*Monitoreo\*\*: Activar \*\*Cloud Monitoring\*\* y definir alertas para asegurar el correcto funcionamiento del sistema.

Con esta propuesta, tendrás una solución escalable, automatizada y eficiente para predecir las ventas de todas tus tiendas usando servicios en la nube de Google. ¿Te gustaría más detalles sobre alguno de estos pasos o algún servicio en particular?

Las predicciones por artículo y por tienda suponen un proceso lento y tenemos un tiempo limitado para su ejecución, desde el cierre de cajas el domingo hasta la apertura el lunes por la mañana (aproximadamente 9 horas). Por lo tanto necesitaremos recursos para acelerar el proceso. La arquitectura que propongo es la siguiente:

* Subir la base de datos a la nube, donde se actualizará